# User Manual

# SmartBond Production Line Tool

## UM-B-041


## Abstract

*This document describes the SmartBond Production Line Tool (PLT). The various software applications, as well as the PLT hardware are explained in detail. The purpose of this document is to help users to become familiar with the tool and help them use it in*


DRAFT

# Contents

**Abstract** ..... **1**

**Contents** ..... **2**

**Figures** ..... **6**

**Tables** ..... **10**

**1 Terms and Definitions** ..... **13**

**2 References** ..... **15**

**3 New version features** ..... **16**

**4 Introduction** ..... **17**

**5 Hardware** ..... **18**

5.1 Hardware Block Diagram ..... 18
5.2 Printed Circuit Board Layout ..... 19
5.3 PLT Power Supply ..... 20
5.4 DUT Connector ..... 20
5.5 DA14531 Single Wire UART Connections ..... 21
5.6 Data Streaming ..... 22
5.7 Golden Unit ..... 23
5.7.1 GU Reset ..... 24
5.8 Current Measurements ..... 24
5.9 Jumper Settings ..... 25
5.9.1 J26 - Current Measurements ..... 25
5.9.2 J42 - DA1458x OTP Burning Voltage ..... 25
5.9.3 J47, J46 - GU Reset ..... 26
5.9.4 J37 - GU Programming ..... 27
5.10 PLT Functional Blocks ..... 28

**6 Software** ..... **29**

6.1 Introduction ..... 29
6.2 Software Package Contents ..... 30
6.3 Prerequisites ..... 33
6.4 System Requirements ..... 34
6.5 Limitations ..... 34
6.6 Building the Code ..... 35
6.7 Executing the Applications ..... 37
6.8 Test Sequence ..... 42
6.8.1 DA1458x Test Sequence ..... 42
6.8.2 DA1469x Test Sequence ..... 46
6.9 VBAT/Reset Signals Operation ..... 50
6.9.1 VBAT Only ..... 50
6.9.2 VBAT On with Reset ..... 51
6.10 Custom Memory Data ..... 51
6.10.1 Homekit Hash Setup Code ..... 52
6.10.2 Custom data CSV file format ..... 54
6.11 Golden Unit Scan Test ..... 55
6.12 Creating PLT Firmware Files ..... 56

**7 Applications** ..... **59**

 © 2020 Dialog Semiconductor

7.1 Introduction ... 59
7.2 CFG PLT Application ... 59
7.2.1 XML and XSD Files ... 61
7.2.2 Hardware Setup ... 63
7.2.2.1 Test Station ... 63
7.2.2.2 Device IC ... 63
7.2.2.3 Active DUTs ... 63
7.2.2.4 DUT COM Ports ... 64
7.2.2.5 Golden Unit ... 64
7.2.2.6 VBAT/Reset Mode ... 65
7.2.3 General ... 66
7.2.3.1 Statistics ... 66
7.2.3.2 Test Options ... 66
7.2.4 BD Addresses ... 69
7.2.4.1 BD Address Assignment ... 69
7.2.5 DUT Hardware Setup (DA14531) ... 73
7.2.5.1 UART Boot Pins Setup ... 73
7.2.5.2 UART Baud Rate ... 73
7.2.5.3 SPI Flash Configuration ... 74
7.2.5.4 I2C EEPROM Configuration ... 75
7.2.6 Test Settings (DA14531) ... 76
7.2.6.1 GPIO Watchdog Operation ... 76
7.2.6.2 VBAT Level Log ... 76
7.2.6.3 OTP Timestamp Read ... 76
7.2.6.4 DC-DC Converter Level Test ... 77
7.2.6.5 XTAL Trim ... 77
7.2.6.6 Scan DUT Advertise Test ... 78
7.2.6.7 RF Tests ... 78
7.2.6.8 GPIO/LED Test ... 85
7.2.6.9 GPIO Connection Test ... 86
7.2.6.10 Sensor Test ... 87
7.2.6.11 Custom Test ... 88
7.2.6.12 External 32kHz Test ... 88
7.2.6.13 Current Measurement Test ... 89
7.2.6.14 Temperature Measurement Test ... 92
7.2.6.15 Scan Test ... 93
7.2.7 Memory Functions (DA14531) ... 94
7.2.7.1 OTP Memory ... 94
7.2.7.2 SPI Flash Memory ... 95
7.2.7.3 I2C EEPROM Memory ... 97
7.2.7.4 Memory read ... 98
7.2.8 Memory Header (DA14531) ... 99
7.2.8.1 General ... 99
7.2.8.2 BD Address ... 100
7.2.8.3 Custom Memory Data ... 101
7.2.8.4 OTP Configuration Script ... 103
7.2.9 DUT Hardware Setup (DA1469x) ... 105

 © 2020 Dialog Semiconductor

7.2.9.1 UART Baud Rate ... 105
7.2.10 Test Settings (DA1469x) ... 105
7.2.10.1 VBAT Level Log ... 105
7.2.10.2 OTP Timestamp Read ... 105
7.2.10.3 XTAL Trim ... 106
7.2.10.4 GPIO Watchdog operation ... 106
7.2.10.5 Scan DUT Advertise Test ... 107
7.2.10.6 RF Tests ... 107
7.2.10.7 GPIO/LED Test ... 113
7.2.10.8 GPIO Connection Test ... 114
7.2.10.9 Sensor Test ... 115
7.2.10.10 Custom Test ... 116
7.2.10.11 External 32kHz Test ... 116
7.2.10.12 Current Measurement Test ... 117
7.2.10.13 Temperature Measurement Test ... 120
7.2.10.14 Scan Test ... 121
7.2.11 Memory Functions (DA1469x) ... 122
7.2.11.1 OTP Memory ... 122
7.2.11.2 QSPI Flash Memory ... 123
7.2.11.3 Memory read ... 124
7.2.12 Memory Header (DA1469x) ... 125
7.2.12.1 QSPI Header - BD Address ... 125
7.2.12.2 Custom Memory Data ... 125
7.2.13 Debug Settings ... 127
7.2.14 Security ... 128
7.3 GUI PLT Application ... 129
7.3.1 GUI PLT Settings ... 131
7.3.2 Barcode Scanner Mode ... 132
7.3.2.1 Homekit Setup Code Scan Example ... 134
7.3.3 Running the GUI PLT and Executing Tests ... 138
7.3.4 Debug Console ... 141
7.3.5 DUT Log File ... 142
7.3.6 CSV File ... 143
7.4 CLI PLT Application ... 144
7.4.1 CLI Commands ... 144
7.4.2 Running the CLI and Executing Tests ... 146
7.4.3 Using CLI Commands with Arguments ... 149
7.5 GU Upgrade Application ... 150
**8 Example Usage ... 156**
**Appendix A Top View of PLT PCB Version D ... 162**
**Appendix B Electrical Schematics ... 163**
**Appendix C Hardware Modifications PLT Version D ... 167**
**Appendix D Application Hardware Design Considerations ... 168**
**Appendix E Suggestions about Hardware and Cabling ... 169**
**Appendix F RF Path Losses Calibration ... 171**
F.1 Prerequisites ... 171

 © 2020 Dialog Semiconductor

F.2 Setup ... 172
F.3 Procedure ... 174
**Appendix G Hex2Bin ... 180**
**Appendix H Bin2Image ... 181**
**Appendix I Automatic GU COM Port Find ... 182**
**Appendix J Improving Cabling between PLT and DUTs ... 184**
**Appendix K DA14583 Internal SPI Flash Memory ... 186**
**Appendix L DA14586 Internal SPI Flash Memory ... 187**
**Appendix M Honeywell Xenon 1900 Barcode Scanner Setup ... 188**
**Appendix N Golden Unit Upgrade Using Smart Snippets Toolbox ... 189**
**Appendix O Connecting a Speaker to the Golden Unit for Audio Test ... 190**
**Appendix P FTDI Driver Removal and Installation ... 191**
**Appendix Q DA1458x DK Pro Motherboard Connection ... 192**
**Appendix R DA1468x DK Pro Motherboard Connection ... 193**
**Appendix S Connecting DA1468x DK Pro Motherboard for Current Measurements ... 194**
**Appendix T Connecting DUT with Battery Supply ... 195**
**Appendix U User Interfaces Shortcut Keys ... 196**
**Appendix V DA1458x Supported SPI Flash\EEPROM Memories ... 197**
**Appendix W DA14531 Supported SPI Flash Memories ... 198**
**Appendix X DA1468x Supported QSPI Flash Memories ... 199**
**Appendix Y DA1469x Supported QSPI Flash Memories ... 199**
**Appendix Z BLE Tester Measurement Results ... 200**
**Appendix AA Memory programming ... 204**
AA.1 DA14580/1/2/3/5/6 Memory Programming Tests ... 204
AA.2 DA1468x Memory Programming Operation ... 207
**Appendix BB CSV Log File Contents ... 209**
**Appendix CC DUT Status Codes ... 222**
**Appendix DD Golden Unit Status Codes ... 233**
**Revision History ... 236**

 © 2020 Dialog Semiconductor

# Figures

Figure 1: Production Line Tool Hardware ... 17
Figure 2: Production Line Tool Hardware Board Block Diagram ... 18
Figure 3: Top View of the PLT Hardware Board ... 19
Figure 4: PLT Hardware Power Connections ... 20
Figure 5: Production Line Tool DUT Connections ... 20
Figure 6: DA14531 Single Wire UART Connections ... 21
Figure 7: CPLD UART Data Streams ... 22
Figure 8: CPLD XTAL Trim Pulse Data Stream ... 22
Figure 9: CPLD UART Loopback Data Stream ... 23
Figure 10: Golden Unit ... 23
Figure 11: GU Reset Circuit ... 24
Figure 12: VBAT DUT Current Measurement Setup ... 25
Figure 13: Connections for ‘Floating Current’ Measurements ... 25
Figure 14: VPP Control Circuit Schematic ... 26
Figure 15: Location of the VPP Jumper J42 ... 26
Figure 16: Location of J46 Jumper ... 26
Figure 17: Location of J47 Jumper ... 27
Figure 18: J37 - GU Programming Jumper Schematics ... 27
Figure 19: Location of J37 Jumper ... 27
Figure 20: PLT Functional Blocks ... 28
Figure 21: Production Line Tool Software Block Diagram ... 29
Figure 22: SmartBond PLT Software Package Contents ... 30
Figure 23: DA14531 Test Sequence ... 45
Figure 24: DA1469x Test Sequence ... 49
Figure 25: VBAT only ... 50
Figure 26: VBAT On with Reset ... 51
Figure 27: Custom Memory Data CSV File Example ... 54
Figure 28: Golden Unit Scan Test ... 55
Figure 29: Golden Unit Scan Test Example Parameters ... 55
Figure 30: Binaries ... 56
Figure 31: Folder Contents of ‘fw_files’ ... 56
Figure 32: CFG PLT Startup Screen ... 59
Figure 33: CFG PLT with Erroneous Configuration Parameter ... 60
Figure 34: XSD Schema File Example ... 62
Figure 35: Station Identification ... 63
Figure 36: Device IC ... 63
Figure 37: Active DUTs ... 63
Figure 38: DUT COM Ports ... 64
Figure 39: Golden Unit COM Port ... 64
Figure 40: VBAT/Reset Mode Selection ... 65
Figure 41: Statistics ... 66
Figure 42: Test Options ... 66
Figure 43: BD Address Assignment ... 69
Figure 44: Example for Load from File Mode ... 70
Figure 45: UART Boot Pins Setup - DA14531 ... 73
Figure 46: UART Baud Rate - DA14531 ... 73
Figure 47: SPI Flash Configuration - DA14531 ... 74
Figure 48: I2C EEPROM Configuration - DA14531 ... 75
Figure 49: GPIO Watchdog Operation – DA14531 ... 76
Figure 50: VBAT Level Log - DA14531 ... 76
Figure 51: OTP Timestamp Read - DA14531 ... 76
Figure 52: DC-DC Converter Level Test - DA14531 ... 77
Figure 53: XTAL Trim - DA1458x ... 77
Figure 54: Scan DUT Advertise Test - DA14531 ... 78
Figure 55: Golden Unit RF Tests - DA14531 ... 79
Figure 56: BLE Tester General Settings - DA1458x ... 80
Figure 57: BLE Tester TX Power - DA14531 ... 81
Figure 58: BLE Tester Frequency Offset - DA14531 ... 82

 © 2020 Dialog Semiconductor

Figure 59: BLE Tester Modulation Index - DA14531 ... 83
Figure 60: BLE Tester RX Sensitivity - DA14531 ... 84
Figure 61: Path Losses per DUT - DA14531 ... 85
Figure 62: GPIO/LED Tests - DA14531 ... 85
Figure 63: GPIO Connection Test - DA14531 ... 86
Figure 64: Sensor Test - DA14531 ... 87
Figure 65: Custom Test - DA14531 ... 88
Figure 66: External 32kHz Test - DA14531 ... 88
Figure 67: Current Measurement Test - DA14531 ... 89
Figure 68: Temperature Measurement Test - DA14531 ... 92
Figure 69: Scan Test - DA14531 ... 93
Figure 70: OTP Memory - DA14531 ... 94
Figure 71: Different Image per DUT Folder Example ... 95
Figure 72: SPI Flash Memory - DA14531 ... 95
Figure 73: I2C EEPROM Memory - DA1458x ... 97
Figure 74: Memory Read Test - DA14531 ... 98
Figure 75: Memory Read Test Example Log File - DA14531 ... 98
Figure 76: OTP Header - DA14531 ... 99
Figure 77: BD Address - DA14531 ... 100
Figure 78: Custom Memory Data - DA14531 ... 101
Figure 79: OTP Configuration Script - DA14531 ... 103
Figure 80: UART Baud Rate - DA1469x ... 105
Figure 81: VBAT Level Log - DA1469x ... 105
Figure 82: OTP Timestamp Read - DA1469x ... 105
Figure 83: XTAL Trim - DA1469x ... 106
Figure 84: GPIO Watchdog operation - DA1469x ... 106
Figure 85: Scan DUT Advertise Test - DA1469x ... 107
Figure 86: Golden Unit RF Tests - DA1469x ... 107
Figure 87: BLE Tester General Settings - DA1469x ... 108
Figure 88: BLE Tester TX Power - DA1469x ... 109
Figure 89: BLE Tester Frequency Offset - DA1469x ... 110
Figure 90: BLE Tester Modulation Index - DA1469x ... 111
Figure 91: BLE Tester RX Sensitivity - DA1469x ... 112
Figure 92: Path Losses per DUT - DA1469x ... 113
Figure 93: GPIO/LED Tests - DA1469x ... 113
Figure 94: GPIO Connection Test - DA1469x ... 114
Figure 95: Sensor Test - DA1469x ... 115
Figure 96: Custom Test - DA1469x ... 116
Figure 97: External 32kHz Test - DA1469x ... 116
Figure 98: Current Measurement Tests - DA1469x ... 117
Figure 99: Temperature Measurement Test - DA1469x ... 120
Figure 100: Scan Test - DA1469x ... 121
Figure 101: OTP Memory - DA1469x ... 122
Figure 102: QSPI Flash - DA1469x ... 123
Figure 103: Memory Read Test - DA1469x ... 124
Figure 104: Memory Read Test Example Log File - DA1469x ... 124
Figure 105: QSPI Header BD Address - DA1469x ... 125
Figure 107: Custom Memory Data - DA1469x ... 125
Figure 108: Debug Settings ... 127
Figure 109: Security ... 128
Figure 110: GUI PLT Main Screen ... 129
Figure 111: GUI PLT Settings ... 131
Figure 112: Barcode Scan Option in GUI PLT ... 133
Figure 113: Barcode Scanner Controls ... 133
Figure 114: Barcode Scan - BD Address Assignment ... 133
Figure 115: GUI PLT - Erroneous Messages in “run scripts before testing starts” ... 138
Figure 116: GUI PLT OTP Burn Warning Message ... 139
Figure 117: GUI PLT Testing (1 of 2) ... 139
Figure 118: GUI PLT Testing (2 of 2) ... 140

 © 2020 Dialog Semiconductor

Figure 119: GUI PLT Tests Finished ... 140
Figure 120: GUI PLT Retry Failed DUTs Message ... 141
Figure 121: Debug Console ... 141
Figure 122: DUT Log File ... 142
Figure 123: CSV File ... 143
Figure 124: CLI Software Start Screen ... 144
Figure 125: CLI PLT Print Settings (x Command) ... 146
Figure 126: CLI PLT DUT COM Port Enumeration ('p' Command) ... 147
Figure 127: CLI PLT Read and Compare BD Address in QSPI ('v' Command) ... 147
Figure 128: CLI PLT Testing ... 148
Figure 129: CLI PLT Testing Finished ... 148
Figure 130: CLI with Commands as Arguments ... 149
Figure 131: GU Upgrade - Introduction page ... 150
Figure 132: GU Upgrade - Exit Message ... 150
Figure 133: GU Upgrade - Hardware Version ... 151
Figure 134: GU Upgrade - Hardware Version Compatibility ... 151
Figure 135: GU Upgrade - Power Supply ... 152
Figure 136: GU Upgrade - Power Supply Pop-Up ... 152
Figure 137: GU Upgrade - Golden Unit Reset ... 153
Figure 138: GU Upgrade - Golden Unit Reset Message for Manual Mode ... 153
Figure 139: GU Upgrade - GU COM Port ... 154
Figure 140: GU Upgrade - Burn Firmware ... 154
Figure 141: GU Upgrade - Burn Firmware Pop-Up Message ... 155
Figure 142: GU Upgrade - Burn Firmware Success ... 155
Figure 143: Top View of PLT PCB Version D ... 162
Figure 144: VBAT and VPP Control from CPLD ... 163
Figure 145: CPLD DUT UART Connections ... 163
Figure 146: CPLD FTDI and GU Control Connections ... 164
Figure 147: FTDI Chip for USB UART to DUTs 1, 2, 3 and 4 ... 164
Figure 148: Quad USB HUB ... 165
Figure 149: Golden Unit - Dedicated USB Port and FTDI Chip ... 165
Figure 150: Golden Unit - GU LED and SPI Flash Memory ... 166
Figure 151: VBAT_DUT and VDDIO Supplies ... 166
Figure 152: GU Supply and VPP Generation ... 166
Figure 153: DA14580_RD_tester Version D ... 167
Figure 154: Jumper J47 Added Next to Golden Unit Socket ... 167
Figure 155: R365 (10 kΩ) Added Next to Reset Button ... 167
Figure 156: R365, J47 and RESET Shown in Electrical Schematic ... 167
Figure 157: Possible Solution of Antenna on Cable and Fixed Radius of DUTs to Antenna ... 169
Figure 158: Possible Solution of Antenna on Cable and DUTs Put in Line ... 169
Figure 159: Example Locations and RSSI Readouts of Horizontal Antenna ... 170
Figure 160: Setup diagram ... 172
Figure 161: Hex2Bin Example Directory with Files ... 180
Figure 162: Hex2Bin.exe Example ... 180
Figure 163: Bin2Image Example Directory with Files ... 181
Figure 164: Bin2Image Example ... 181
Figure 165: Example of Twisted Pair Cable with 4 Pairs and Ferrite ... 184
Figure 166: Location of Pull-Down Resistors ... 184
Figure 167: Anti-Ringing Solution ... 185
Figure 168: Pin Assignment of DA14583 – QFN40 ... 186
Figure 169: Pin Assignment of DA14586 - QFN40 ... 187
Figure 170: Speaker Connection for Audio Test. ... 190
Figure 171: DA14580/5 Pro Motherboard DK Wiring. ... 192
Figure 172: DA14680/1/2/3 Pro Motherboard DK Wiring. ... 193
Figure 173: DA1468x PRO DK Ammeter Connection with PLT ... 194
Figure 174: DA1468x DK PRO Current Measurement Settings ... 194
Figure 175: CSV File Entries (1/4) ... 214
Figure 176: CSV File Entries (2/4) ... 215
Figure 177: CSV File Entries (3/4) ... 219

 © 2020 Dialog Semiconductor

Figure 178: CSV File Entries (4/4) ..... 221

 © 2020 Dialog Semiconductor

# Tables

Table 1: SmartBond_PLT_v4.4 added features ... 16
Table 2: Power Supply Requirements ... 20
Table 3: PLT Connections to Applications ... 21
Table 4: Jumpers ... 25
Table 5: PLT User Interface Application Executables ... 30
Table 6: Executables Folder Description ... 30
Table 7: Production Line Tool Prerequisites ... 33
Table 8: Minimum System Requirements ... 34
Table 9: Opening the PLT Visual Studio 2017 Express Source Code Solution ... 35
Table 10: SmartBond_CFG_PLT.exe Application Execution ... 37
Table 11: SmartBond_GUI_PLT.exe Application Execution ... 39
Table 12: SmartBond_CLI_PLT.exe Application Execution ... 40
Table 13: GU_fw_upgrade.exe Application Execution ... 41
Table 14: DA14531 Test Sequence ... 42
Table 15: DA1469x Test Sequence ... 46
Table 16: Custom Memory Data Input Modes ... 52
Table 17: Homekit Setup Code Format ... 52
Table 18: Homekit Setup Code Checksum Algorithm ... 53
Table 19: CFG PLT Main Menu Options ... 60
Table 20: CFG PLT Bottom Strip Options ... 60
Table 21: XML File Parts ... 61
Table 22: Station Identification ... 63
Table 23: Device IC ... 63
Table 24: Active DUTs ... 64
Table 25: DUT COM Ports ... 64
Table 26: Set the GU COM Port ... 65
Table 27: Golden Unit Firmware Version Upgrade ... 65
Table 28: VBAT/Reset Mode ... 65
Table 29: Statistics ... 66
Table 30: BD Address Assignment - Standard Mode ... 70
Table 31: BD Address Assignment Options - Range Mode ... 70
Table 32: BD Address Assignment Options - Load from File Mode ... 71
Table 33: BD Address Assignment Options - Scan Mode ... 72
Table 34: UART TX-RX Pins - DA1458x ... 73
Table 35: UART Baud Rate - DA1458x ... 73
Table 36: SPI Pin Setup - DA14531 ... 74
Table 37: SPI Flash Configuration - DA14531 ... 74
Table 38: I2C Pin Setup - DA14531 ... 75
Table 39: I2C EEPROM Configuration – EEPROM Memory Options - DA14531 ... 76
Table 40: GPIO Watchdog Configuration Options – DA14531 ... 76
Table 41: DC-DC Converter Level Test Options - DA14531 ... 77
Table 42: XTAL Trim - DA1458x ... 77
Table 43: Scan DUT Advertise Test - DA1458x ... 78
Table 44: Golden Unit RF Tests - DA14531 ... 79
Table 45: BLE Tester General Settings - DA1458x ... 80
Table 46: BLE Tester TX Power - DA14531 ... 81
Table 47: BLE Tester Frequency Offset - DA14531 ... 82
Table 48: BLE Tester Modulation Index - DA14531 ... 83
Table 49: BLE Tester RX Sensitivity - DA14531 ... 84
Table 50: Path Losses per DUT from RF Tests DA14531 Options ... 85
Table 51: GPIO/LED Tests - DA14531 ... 85
Table 52: GPIO Connection Test - DA14531 ... 86
Table 53: Sensor Tests - DA14531 ... 87
Table 54: Custom Tests - DA14531 ... 88
Table 55: External 32kHz Test - DA14531 ... 88
Table 56: Current Measurement Test – General Settings - DA14531 ... 90
Table 57: Peripheral Current Measurement - DA14531 ... 90
Table 58: Current Measurement Test - Sleep Current Measurement - DA14531 ... 91

 © 2020 Dialog Semiconductor

Table 59: Temperature Measurement Test - DA14531 ... 92
Table 60: Scan Test DA14531 Options ... 93
Table 61: OTP Memory - DA14531 ... 94
Table 62: SPI Flash Erase - DA14531 ... 96
Table 63: SPI Flash Image Write - DA14531 ... 96
Table 64: I2C EEPROM Image Write - DA1458x ... 97
Table 65: Memory Read Test - DA14531 ... 98
Table 66: OTP Header - DA14531 ... 99
Table 67: BD Address - DA14531 ... 100
Table 68: Custom Memory Data - DA14531 ... 102
Table 69: OTP Configuration Script - DA14531 ... 103
Table 70: UART Baud Rate - DA1469x ... 105
Table 71: XTAL Trim - DA1469x ... 106
Table 72: GPIO Watchdog operation - DA1469x ... 106
Table 73: Scan DUT Advertise Test - DA1469x ... 107
Table 74: Golden Unit RF Tests - DA1469x ... 108
Table 75: BLE Tester General Settings - DA1469x ... 108
Table 76: BLE Tester TX Power - DA1469x ... 109
Table 77: BLE Tester Frequency Offset - DA1469x ... 110
Table 78: BLE Tester Modulation Index - DA1469x ... 111
Table 79: BLE Tester RX Sensitivity - DA1469x ... 112
Table 80: Path Losses per DUT from RF Tests DA1469x Options ... 113
Table 81: GPIO/LED Tests - DA1469x ... 113
Table 82: GPIO Connection Test - DA1469x ... 114
Table 83: Sensor Tests - DA1469x ... 115
Table 84: Custom Tests DA1469x Options ... 116
Table 85: External 32kHz Tests DA1469x Options ... 116
Table 86: Current Measurement Tests - DA1469x ... 118
Table 87: Current Measurement Test – Peripheral Current Measurement - DA1469x ... 118
Table 88: Current Measurement Test - Sleep Current Measurement - DA1469x ... 119
Table 89: Temperature Measurement Test - DA1469x ... 120
Table 90: Scan Test DA1469x Options ... 121
Table 91: OTP Memory - DA1469x ... 122
Table 92: QSPI Flash Erase - DA1469x ... 123
Table 93: QSPI Flash Image Write - DA1469x ... 123
Table 94: Memory Read Test - DA1469x ... 124
Table 95: QSPI Header BD Address - DA1469x ... 125
Table 97: Custom Memory Data - DA1469x ... 126
Table 98: Debug Settings ... 127
Table 99: Security Options ... 128
Table 100: GUI PLT Main Screen Description ... 129
Table 101: GUI PLT Settings ... 132
Table 102: Homekit Setup Code Scan Example ... 134
Table 103: CLI Commands ... 144
Table 104: DA14580 PLT Example Usage ... 156
Table 105: RF Test RSSI Results ... 170
Table 106: Prerequisites ... 171
Table 107: Calibration Setup ... 172
Table 108: Procedure steps ... 174
Table 109: FTDI "DialogSemi" Serial Number ... 182
Table 110: DA14583 Internal SPI Flash Connections ... 186
Table 111: DA14586 Internal SPI Flash Connections ... 187
Table 112: User Interface Shortcut Keys ... 196
Table 113: DA1458x Supported SPI Flash Memories ... 197
Table 114: DA14531 Supported SPI Flash Memories ... 198
Table 113: DA1468x Supported QSPI Flash Memories ... 199
Table 114: DA1469x Supported QSPI Flash Memories ... 199
Table 115: MT8852B Supported Command Codes ... 200
Table 116: MT8852B – BLE TX Output Power Test Results ... 200

 © 2020 Dialog Semiconductor

Table 117: MT8852B – BLE Carrier Frequency Offset and Drift Test Results ..... 201
Table 118: MT8852B – BLE Modulation Characteristics Test Results ..... 201
Table 119: MT8852B – Error List ..... 202
Table 120: DA1458x Memory Programming – OTP Memory ..... 209
Table 121: DA1458x Memory Programming - SPI Flash ..... 209
Table 122: DA1458x Memory Programming – EEPROM Memory ..... 209
Table 123: DA1468x Memory Programming – OTP Memory ..... 209
Table 124: Memory Programming – QSPI Memory ..... 209
Table 125: CSV File Contents ..... 209
Table 126: CSV File Entries (1/4). ..... 214
Table 127: CSV File Entries (2/4). ..... 215
Table 128: CSV File Entries (3/4). ..... 219
Table 129: CSV File Entries (4/4). ..... 221
Table 130: DUT Status Codes. ..... 222
Table 131: Golden Unit Status Codes ..... 233

 © 2020 Dialog Semiconductor

# 1 Terms and Definitions

| | |
|---|---|
| API | Application Programming Interface |
| BD | Bluetooth Device |
| .bin | Firmware files in binary format |
| BLE | Bluetooth low energy |
| CFG | Configuration |
| CLI | Command Line Interface |
| COM | Communication port |
| CPLD | Complex Programmable Logic Device |
| CRC | Cyclic redundancy check |
| CS | Configuration Script |
| CSV | Comma Separated Values |
| DLL | Dynamic Link Library |
| DMA | Direct Memory Access |
| DMM | Digital Multi-meter |
| DTM | Direct Test Mode (as specified by the BLE Core standard) |
| DUT | Device Under Test |
| DVM | Digital Voltage Meter |
| EEPROM | Electrically Erasable Programmable Read-Only Memory |
| .exe | Executable file |
| FTDI | Future Technology Devices International Ltd. |
| GPIO | General Purpose Input-Output |
| GU | Golden Unit |
| GUI | Graphical User Interface |
| Hex | Firmware file in ASCII format |
| HW | hardware |
| IC | Integrated Circuit |
| IDE | Integrated Development Environment |
| I2C | Inter-Integrated Circuit |
| JTAG | Joint Test Action Group |
| OS | Operating System |
| OTP | One Time Programmable (memory) |
| PC | Personal Computer |
| PCB | Printed circuit board |
| PER | Packet Error Rate |
| PLT | Production Line Tool |
| PLTD | Production Line Tool DLL |
| POR | Power-On Reset |
| RAM | Random Access Memory |
| RCX | Resistor Crystal Oscillator |
| RF | Radio Frequency |
| RX | Receive |
| SCPI | Standard Commands for Programmable Instruments |
| SoC | System on Chip |
| SDK | Software Development Kit |
| SPI | Serial Peripheral Interface |
| SW | Software |

 © 2020 Dialog Semiconductor

| | |
|---|---|
| TCS | Trim and Calibration Section |
| TX | Transmit |
| UART | Universal Asynchronous Receiver/Transmitter |
| UI | User Interface |
| USB | Universal Serial Bus |
| VISA | Virtual Instrument Software Architecture |
| VPP | Programming supply voltage (pin) |
| XML | Extensible Markup Language |
| XTAL | Crystal |
| XSD | XML Schema Definition |

 © 2020 Dialog Semiconductor

# 2 References

[1] UM-B-040, DA1458x/DA1468x Production Line Tool Libraries v4.2s, User manual, Dialog Semiconductor.

[2] UM-B-014, DA1458x Bluetooth Smart Development Kit - Expert, User Manual, Dialog Semiconductor

[3] FT4232H – Hi-Speed Quad USB UART IC, FTDI Chip.

[4] FT232 – USB UART IC, FTDI Chip

[5] Anritsu MT8852B, https://www.anritsu.com/en-US/test-measurement/products/mt8852b

[6] Keysight 34401A, http://www.keysight.com/en/pd-1000001295%3Aepsg%3Apro-pn-34401A/digital-multimeter-6-digit?cc=US&lc=eng

[7] Keithley 2000, http://www.tek.com/tektronix-and-keithley-digital-multimeter/keithley-2000-series-6%C2%BD-digit-multimeter-scanning

[8] Papouch TMU USB thermometer, https://www.papouch.com/en/shop/product/tmu-usb-thermometer/

[9] NI USB TC-01, http://sine.ni.com/nips/cds/view/p/lang/en/nid/208177

[10] Honeywell Xenon 1900, https://www.honeywellaidc.com/products/barcode-scanners/general-duty/xenon-1900g-1902g

[11] Zebra/Motorola LS2208, https://www.zebra.com/us/en/products/scanners/general-purpose-scanners/handheld/ls2208.html

[12] AN-B-020, DA14580 End product testing and programming guidelines, Application Note, Dialog Semiconductor

[13] Litepoint IQXel-M, http://www.litepoint.com/test-solutions-for-manufacturing/iqxel-m/

[14] NI USB-6009 DAQ, http://sine.ni.com/nips/cds/view/p/lang/en/nid/201987

[15] Keysight 34461A, http://www.keysight.com/en/pd-2270273-pn-34461A/digital-multimeter-6-digit-34401a-replacement-truevolt-dmm?cc=GR&lc=eng

 © 2020 Dialog Semiconductor

# 3 New version features

This manual explains the usage of the 16 channels SmartBond™ Production Line Tool (PLT). It refers to the SmartBond_PLT_v4.4 software release, which compared to DA1458x_DA1468x_PLT_v4.3 has the added features illustrated in Table 1.

**Table 1: SmartBond_PLT_v4.4 added features**

| # | Features | Description |
|---|---|---|
| 1 | Application rename | Applications were renamed from DA1458x_DA1468x_PLT_vX.X to SmartBond_PLT_vX.X. |
| 2 | DA14531-AE/AF support | New chipset support. |
| 3 | DA1469x support | New chipset support. |
| 4 | Tester-ID | Added Tester ID. Tester ID is shown in the SmartBondTM Production Line Tool GUI, in the DUT logs and in the CSV log file. |
| 5 | Reset duration | Reset duration can now be more than 50 ms. In previous versions the reset duration was fixed to 50 ms. Now, this can be adjusted between 10ms and 1000 ms. |
| 6 | Single wire UART | Single wire UART support for DA14531 devices, at either P03 or P05 GPIOs. |
| 7 | VBAT measure | Measure VBAT and log it, using internal ADC. |
| 8 | OTP timestamp | Read IC specific OTP timestamp and log it. |
| 9 | DC-DC converter level | DA14531 DC-DC converter level test. |
| 10 | BLE scan test at all advertisement channels | BLE scan test at all advertisement channels. If 'All channels' is selected, three different tests are performed, at CH37, CH38 and at CH39. Before, if 'All channels' was selected, the Bluetooth LE stack was selecting the advertisement channel according to the Bluetooth® specification. |
| 11 | 'No short' GPIO connection test. | Added 'No short' GPIO connection test. If 'No short' checkbox is selected, the tool will return an error if the two GPIOs are found to be shorted. |
| 12 | TX power control for DA14531 devices. | Added TX power control for DA14531 devices. The TX power control can be adjusted in the 'Scan DUT Advertise Test' and in all TX Bluetooth LE tester tests. |
| 13 | UART RX pin select as XTAL trim GPIO input. | The UART RX pin can now be selected as XTAL trim GPIO input pulse pin. Before user had to select the specific GPIO (e.g. P05). |
| 14 | 'Single Device' current measurement test. | Added 'Single Device' current measurement test. This is to be used during PLT production setup and not in the actual production line, in order to find the average current measurement limits, by first measuring multiple devices. |
| 15 | 'Skip if written' in all OTP writes. | Added 'Skip if written' in all OTP writes. If this option is selected the tool will first read the OTP area to be written. If the area contains data, it will not write new data and proceed to the next operation without error. |
| 16 | Extend memory read size to 64 Mbytes. | The memory read size has been extended by more than 256 bytes to 64 Mbytes. If the size to be read is more than 256 bytes, the read data will be saved on a file under '`mem_read_test`' folder. |
| 17 | OTP configuration script | OTP configuration script support for DA14531 and DA1469x. |

 © 2020 Dialog Semiconductor

# 4 Introduction

By using the PLT, it is possible to test, calibrate and load firmware for 16 different devices under test (DUTs) in parallel.

The following are deliverable parts of the tool.

- Hardware
  - Main board (Figure 1**Error! Reference source not found.**) together with a DA14580-QFN48 Golden Unit.
  - Electrical schematics of the main board.
  - Gerber files of the main board.
  - Bill of Materials of the main board.
- Software
  - Source code files organized in a Microsoft® Visual Studio Express 2017 solution.
  - Application executables and required DLLs.
- Documents.

An example of a sequence of actions the tool performs is given below. All actions are performed in parallel for up to 16 devices.

1. Download the production test firmware (e.g. `prod_test_531.bin`).
2. Perform automatic crystal (XTAL) trimming.
3. Perform RF RSSI test.
4. Download and burn the customer firmware (into OTP, SPI flash, QSPI flash or I2C EEPROM)
5. Burn the OTP header.
6. Perform Scan test. Reset the DUTs and set the GU to scan for the DUT BLE advertisements.

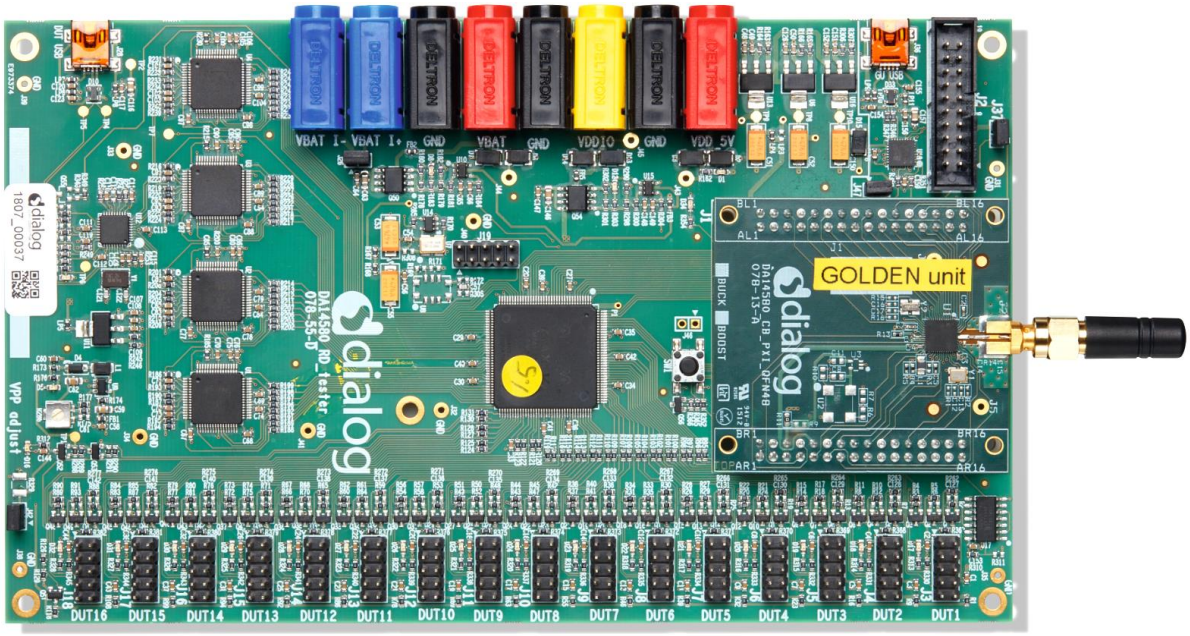



**Figure 1: Production Line Tool Hardware**


© 2020 Dialog Semiconductor

# 5 Hardware

## 5.1 Hardware Block Diagram

The Production Line Tool hardware consists of various blocks, as illustrated in Figure 2. These blocks are explained below.

- **Blue blocks:** USB-to-UART interfaces.
  - Four **FT4232 FTDI** QUAD USB-to-UART interfaces are used for a 16-channel USB-to-UART conversion.
  - The GU is connected to the PC via an **FT232 FTDI** USB-to-UART interface.
- **Red block:** A **CPLD** that has the following purpose.
  - Switch UART signals between the PC USB-UART and DUTs.
  - Switch DUTs VBAT signal
  - Switch DUTs VPP signal (only when VBAT is enabled).
  - Produce Reset signal to the DUTs.
  - Produce 500ms XTAL calibration pulse.
- **Orange block:** A **Golden Unit** (GU) is mounted, which has the following functionality:
  - CPLD control using custom commands.
  - Transceiver for Bluetooth RF signals to and from the DUTs.
  - Produce an audio tone using PWM, used for audio testing.
  - Scan for device BLE advertisements, after the customer firmware has been programmed.
- **Purple blocks:** Sixteen (16) device connectors.

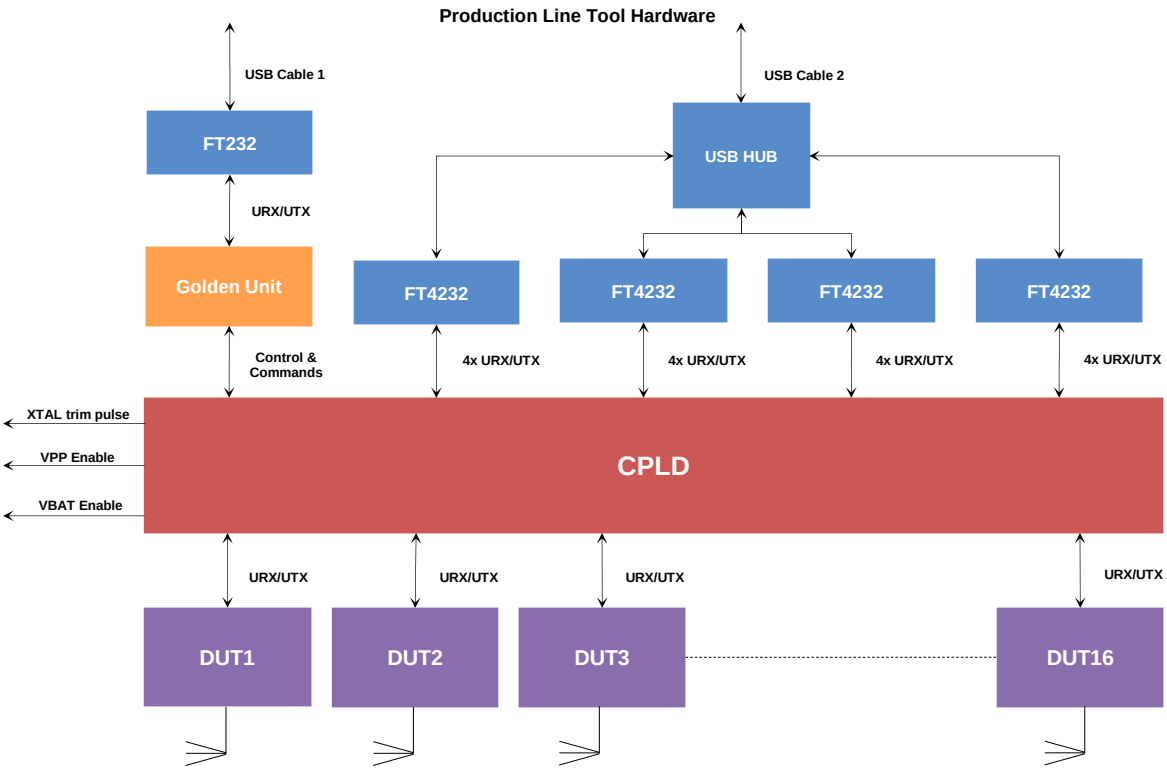



**Figure 2: Production Line Tool Hardware Board Block Diagram**


© 2020 Dialog Semiconductor

## 5.2 Printed Circuit Board Layout

In Figure 3, the top view of the PLT board is illustrated. The important parts are pointed by the orange boxes. The *VPP jumper* and the *Current jumper* are colored in blue.

The Golden Unit has a DA14580 QFN48-die soldered. Most of the 48 pins are basically used to connect to the CPLD. The CPLD is programmed during the production of the PLT board via the CPLD socket. No need for the users to use the CPLD socket.

The black banana sockets are all connected to the same ground (GND) plane.

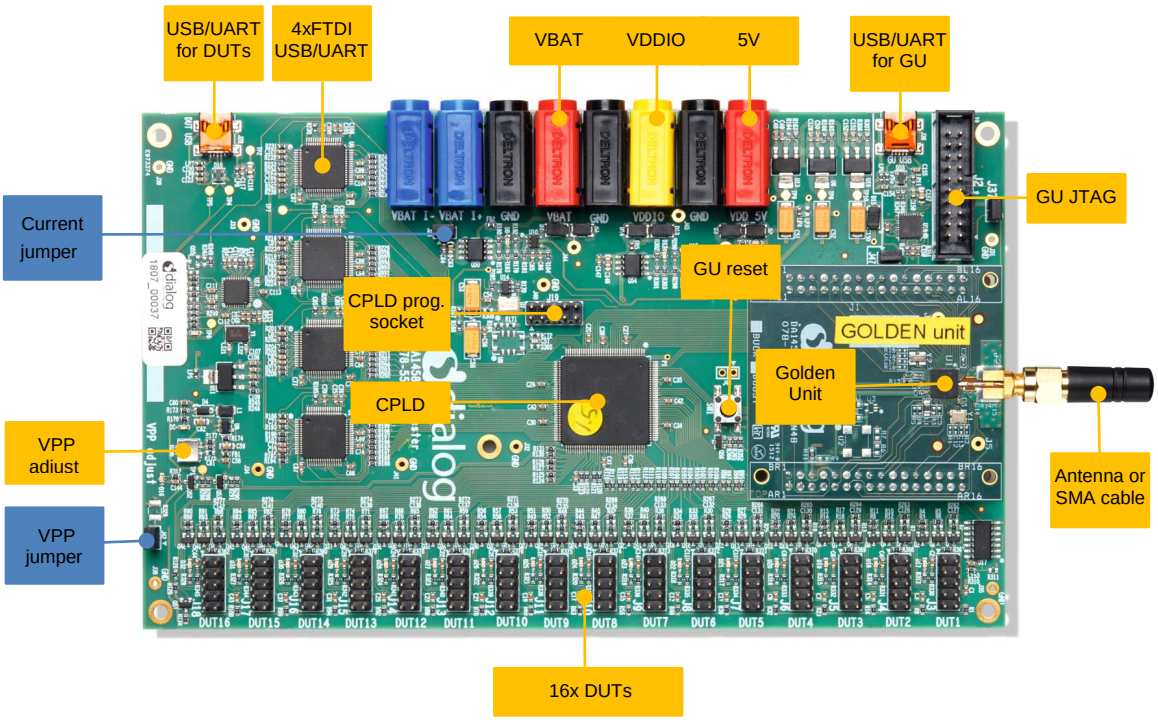



**Figure 3: Top View of the PLT Hardware Board**


© 2020 Dialog Semiconductor

## 5.3 PLT Power Supply

External power supply is needed for the PLT to run. This should be connected to the banana sockets as shown in Figure 4.

Table 2 shows the voltage and current requirements for each power supply. The blue banana sockets can be used for device current measurements.

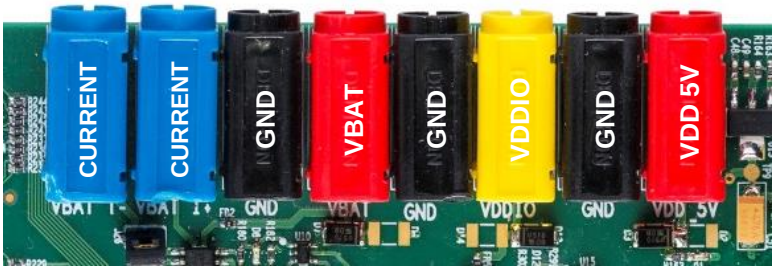



**Figure 4: PLT Hardware Power Connections**

**Table 2: Power Supply Requirements**

| Power Supply | Voltage (V) | Current (mA) | |
|---|---|---|---|
| | | Buck Mode | Boost Mode |
| VBAT (Buck mode) | 2.4 … 3.3 | 16 x 20 | |
| VBAT (Boost mode) | 1.5 … 3.3 | | 16 x 20 |
| VDDIO | 2.4 … 3.3 | 70 | 70 |
| VDD 5V | 4.75 … 5.25 | ~335 | ~335 |
| VPP | 6.6 … 6.8 | 16 x 2 | 16 x 2 |

## 5.4 DUT Connector

The BLE devices are connected to the PLT using the DUT1-16 connectors at the edge of the PLT board. Figure 5 shows the pin-header connections from the Production Line Tool hardware board to the DUTs. Table 3 describes the purpose of each pin.

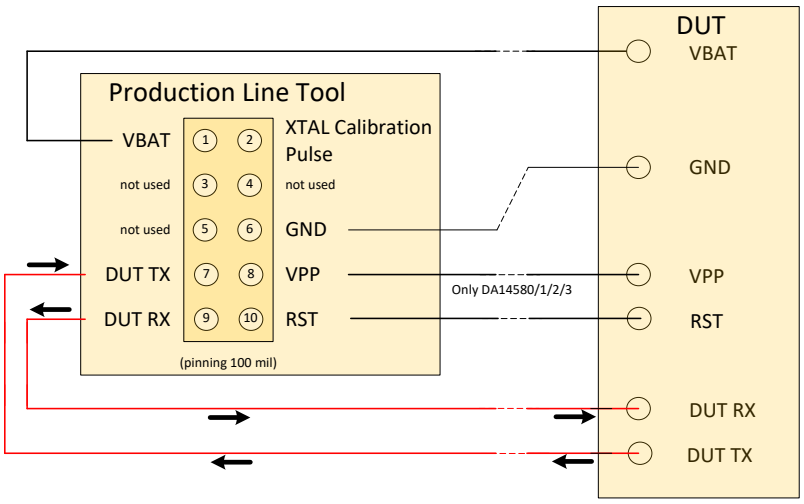



**Figure 5: Production Line Tool DUT Connections**


© 2020 Dialog Semiconductor

**Table 3: PLT Connections to Applications**

| Header Pin | Name | Description |
|---|---|---|
| 1 | VBAT | Depending on the VBAT/Reset Signals Operation mode this can be used as Voltage supply for the DUT or as Reset signal. Due to this connection, no external power supply is needed for the DUTs. This pin must be connected if there is no other power supply (e.g. battery). |
| 2 | XTAL Calibration Pulse | This pin can be used as a reference pulse during the automatic crystal calibration. More details are given in 7.2.6.5 for DA14531 devices and in 7.2.10.3 for DA1469x devices. The crystal trim pulse can also be supplied in the UART RX device pin. This is the most common scenario. However, there may be hardware limitations where the UART RX pin cannot be used. In such cases, this particular PLT header pin is used. |
| 6 | GND | Ground pin. This pin must be connected. |
| 7 | DUT TX | This is connected to the device UART TX pin. This pin must be connected. |
| 8 | VPP | This pin provides the 6.8V required to program the OTP in the DA14580/1/2/3 devices.<br>**Note:** This option is not available with the  mode. |
| 9 | DUT RX | This is connected to the device UART RX pin. This pin can also provide the crystal calibration reference pulse for the automatic crystal (XTAL) trim procedure, as described in 7.2.6.5 for DA1458x devices and in 7.2.10.3 for DA1468x devices. This pin must be connected. |
| 10 | RST | The reset signal must be connected if battery powered devices are used.<br>A power cycle of VBAT will produce a Power on Reset (POR), so a RESET is given to the DUT. In that case the RST-wire is not needed.<br>In summary, when no battery is used, the POR will RESET the DUT. |

## 5.5 DA14531 Single Wire UART Connections

DA14531 supports single wire UART. For this purpose, DA14531 GPIO P05 or P03 can be used. The UART pin connections to PLT are as shown in Figure 6. The 100Ohm resistor in the device is optional. The PLT hardware does not need any modifications. The only actual requirement is to short circuit the PLT DUT RX and DUT TX pins as close to PLT as possible. The resistors shown in the PLT block already exist.

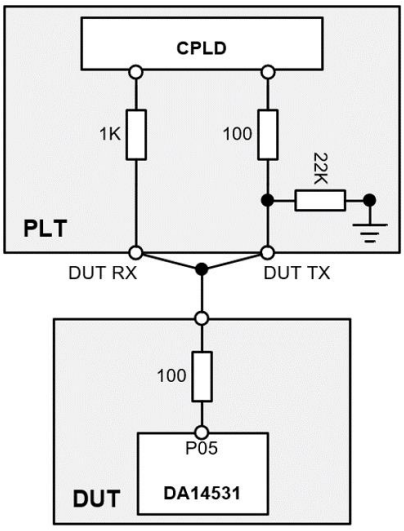



**Figure 6: DA14531 Single Wire UART Connections**

**Note 1** The 100Ohm resistor in the device is optional.


© 2020 Dialog Semiconductor

## 5.6 Data Streaming

Figure 7, Figure 8 and Figure 9 illustrate the three possible data streams through the CPLD. The CPLD switches S1, S2, S3 and S4 are controlled by the software via the Golden Unit.

### Normal Operation (Figure 7):

**UART-RxD data is transported via the RED arrows (AA):**
PC ➔ USB ➔ USB HUB ➔ Quad UART ➔ CPLD signal 'AA' ➔ DUT RxD (programmed as RxD).

**UART-TxD data is transported via the BLUE arrows (BB):**
PC ➔ USB ➔ USB HUB ➔ Quad UART ➔ CPLD signal 'BB' ➔ DUT TxD.

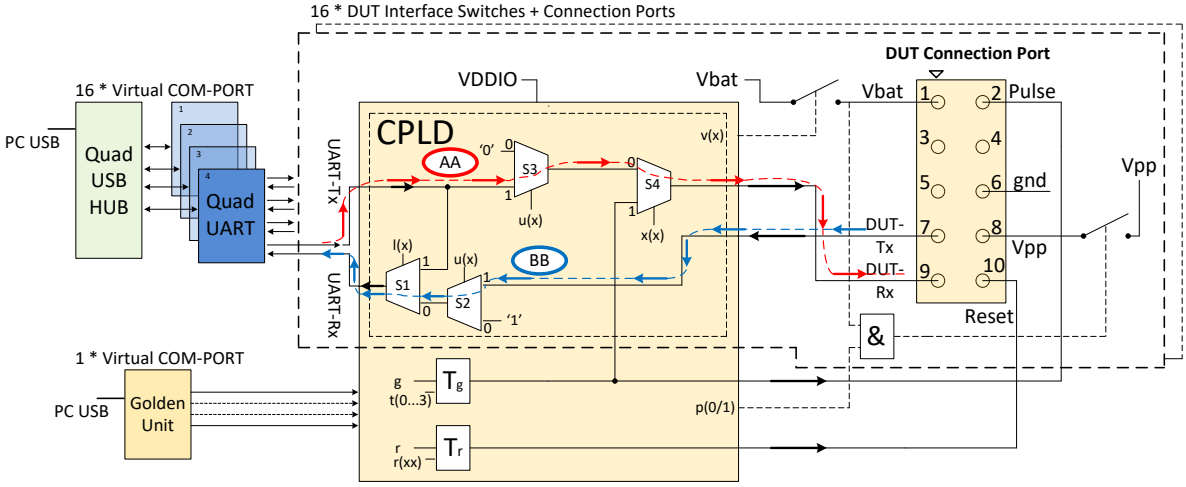



**Figure 7: CPLD UART Data Streams**

### Crystal Trimming (Figure 8):

**The XTAL calibration pulse (500ms) is transported via the PURPLE arrows (CC):**
CPLD TIMER Tg ➔ CPLD S4 ➔ DUT RxD (programmed as GPIO).

**UART-TxD data is transported via the BLUE arrows (BB):**
PC ➔ USB ➔ USB HUB ➔ Quad UART ➔ CPLD signal 'BB' ➔ DUT TxD.

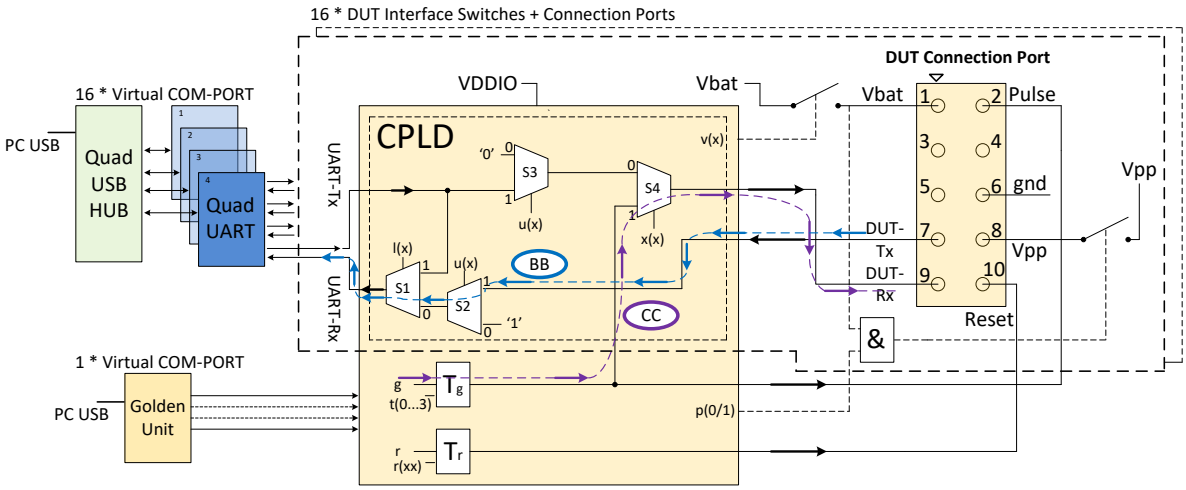



**Figure 8: CPLD XTAL Trim Pulse Data Stream**


© 2020 Dialog Semiconductor

### Loopback Operation (Figure 9):

Loopback operation is used during the start of the tests. The PC PLT software uses this feature to automatically find the numbers of the Virtual COM ports in the Windows PC.

**The UART loopback data is transported via the GREEN arrows (DD):**
PC ➔ USB ➔ USB HUB ➔ Quad UART ➔ CPLD signal 'DD' SW1 ➔ Quad UART ➔ USB HUB ➔ USB ➔ PC.

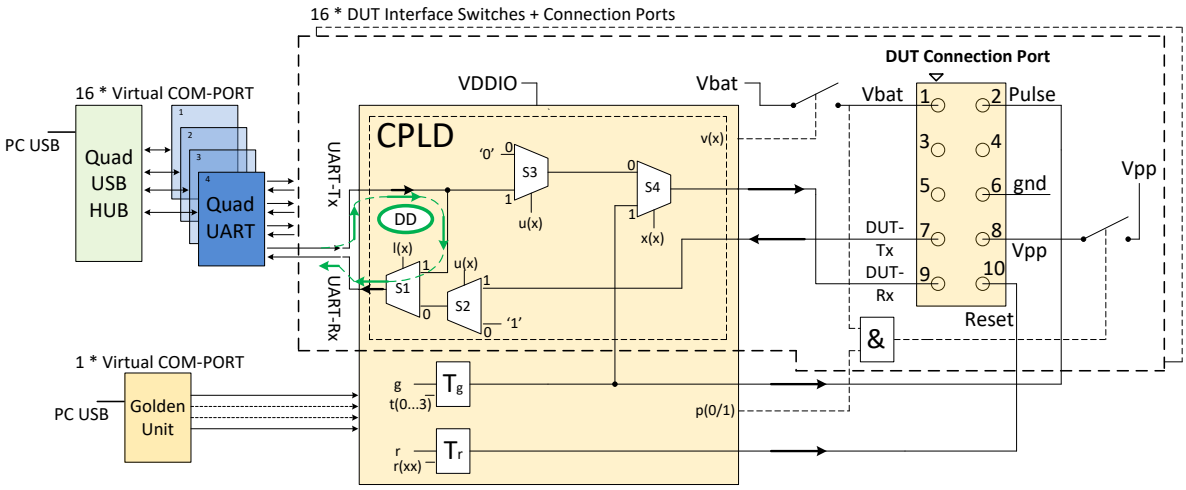



**Figure 9: CPLD UART Loopback Data Stream**

**Note:** The CPLD is also used to switch the UART signals between the QUAD FTDIs and the DUTs. When the VBAT is switched off and the UART wires are not disconnected, a 'rest voltage' may be present on the product. This could cause problems with the power-on reset (POR) and the product might not boot correctly. The CPLD will switch off the UART signals when the VBAT is not present.

## 5.7 Golden Unit

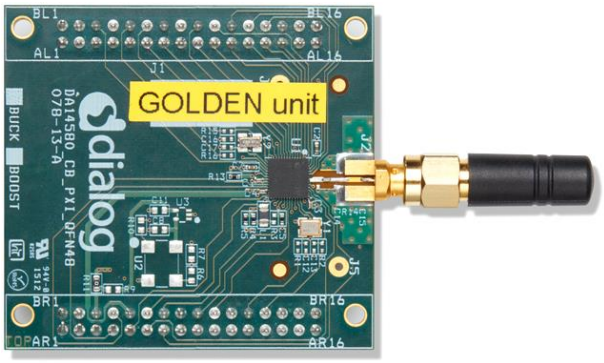



**Figure 10: Golden Unit**

The Golden Unit (GU) is a 'daughter' board mainly used in the Expert Development Kit [2]. In the PLT, the GU is used for various purposes:

- RF transmitter for the RF RSSI DUT test.
- RF Receiver for the device BLE advertisement scan test.
- Audio tone generator for the audio test.
- Controlling the CPLD.

The GU uses an SPI Flash memory mounted on the PLT board. The SPI Flash is pre-programmed with a specific production test firmware. If required, there are several ways to upgrade the GU firmware, either via the PLT's GU JTAG connector, via the UART or using a new GUI application


© 2020 Dialog Semiconductor

executable (`GU_fw_upgrade.exe`) as explained in 7.5. The latest GU firmware can be found inside the latest PLT software release, under the `executables\binaries\GU` folder.
**Note:** PLT v4.3 and onwards requires the latest firmware version of the Golden Unit. If the Golden Unit firmware is not updated, then the PLT applications will not run.

**Note:** The Golden Unit is calibrated during PLT production. It is delivered with a calibration characterization document.

### 5.7.1 GU Reset

The Golden Unit includes a hardware reset circuit. The GU reset signal is connected to an FTDI FT232 GPIO pin.

Figure 11 illustrates the electrical schematics of the GU reset circuit. Section 5.9.3 illustrates the jumper positions on the PLT PCB.

The red line is the connection between the FTDI IC GPIO pin (DTR) and the GU reset signal on the PLT GU connector header. The PLT software controls this pin via the FTDI DLL driver `ftd2xx.dll`. Making pin DTR low for a short period of time will reset the GU. Every time the PLT tests start, a hardware reset is issued to the Golden Unit. Jumper J47 should be ON and J46 OFF for this reset method to operate.

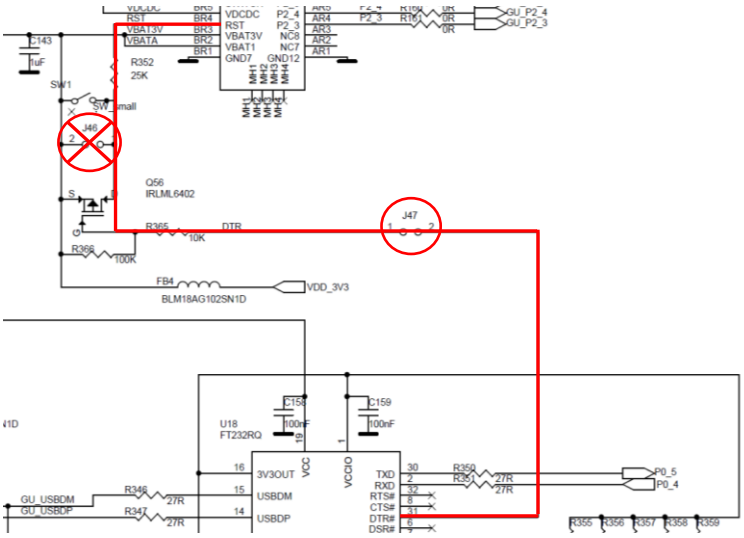



**Figure 11: GU Reset Circuit**

## 5.8 Current Measurements

The PLT board provides connections to perform DUT current measurements (see Figure 12). By connecting a current meter to the blue banana sockets, the combined VBAT current of all DUTs can be measured. Jumper J26 should be removed when a current meter is connected. If no current meter is used, jumper J26 should be mounted. See also section 5.9.

The connection shown in Figure 12 can only be used with the VBAT Only and VBAT On with Reset (when the VBAT lines are used to power the DUTs) modes. If the DUTs are powered using a single external power supply, then the multi-meter should be connected on that power supply in a similar way as described before with the PLT. If the DUTs are powered independently (e.g. each one with its own battery) the current measurement procedure cannot be used.

 © 2020 Dialog Semiconductor

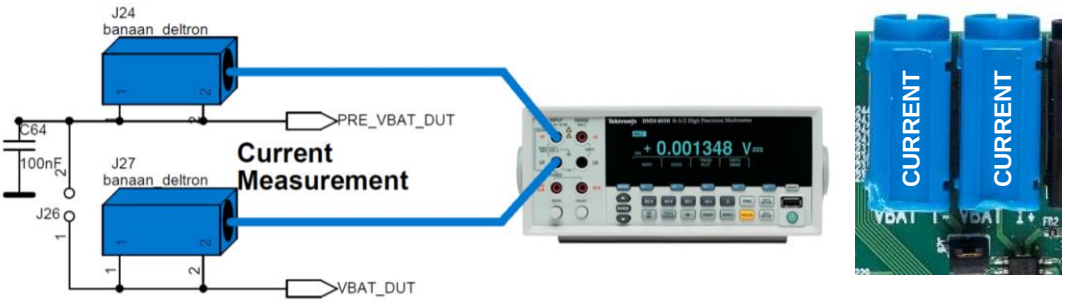



**Figure 12: VBAT DUT Current Measurement Setup**

## 5.9 Jumper Settings

This section describes the PLT hardware jumper settings.

**Table 4: Jumpers**

| Jumper | PLT HW version | Description |
|---|---|---|
| J26 | A, B, C, D | Connects the VBAT line from the PLT power supply to the DUTs. This jumper can be used when there is no multi-meter instrument connected for current measurement. |
| J37 | B, C, D | This jumper sets the Golden Unit's SPI Flash chip select (CS) pin to high. This jumper is needed placed when the Golden Unit should **NOT** boot from the SPI flash. |
| J42 | B, C, D | Feeds the VPP lines of the DUT connectors with VPP voltage used for OTP burning in DA14580/1/2/3 DUTs. |
| J46 | C, D | This jumper can be used to reset the Golden Unit. The two pins on the jumper are the same as the ones in the GU reset switch next to the jumper. |
| J47 | D | This jumper connects the Golden Unit's FTDI DTR line to the Golden Unit's reset pin. With this jumper on the PLT, software can reset the Golden Unit on-demand. |

### 5.9.1 J26 - Current Measurements

As shown in Figure 13, jumper J26 should be mounted when no external current meter is attached. Otherwise, when a current meter is connected via the blue banana sockets to measure the device current, the J26 jumper should be removed.

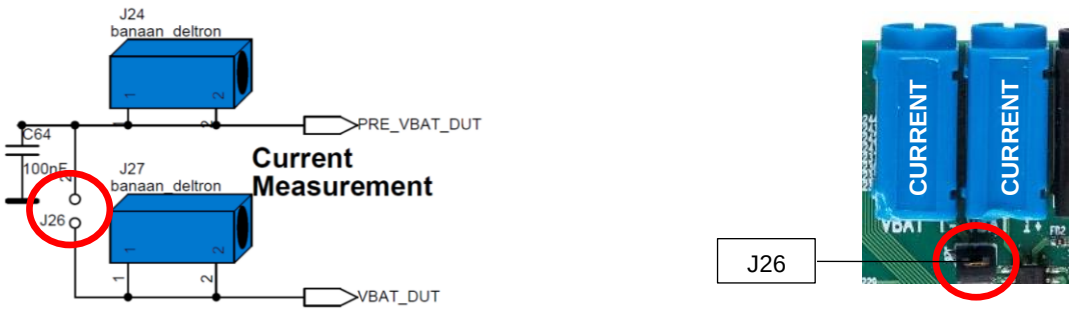



**Figure 13: Connections for 'Floating Current' Measurements**

### 5.9.2 J42 - DA1458x OTP Burning Voltage

If DA14580/1/2/3 OTP programming is required, the VPP line should be connected between the PLT DUT connector and the actual DUT (Figure 5). Jumper J42 on the PLT board should also be


© 2020 Dialog Semiconductor

mounted. Figure 15 shows the jumper position on the PLT hardware board. Figure 14 illustrates the electrical schematics of the VPP and the location of the J42 jumper.

DA14585/6 and DA1468x devices do not need an external voltage for the OTP to be burned. Therefore, the VPP line from the PLT DUT connector (Figure 5) should not be connected to the DA1468x DUT.

**Note:** The VPP line feature **cannot** be used with the  mode, because of the fact that the VPP line will not operate when VBAT is not switched on. It is secured in the hardware.

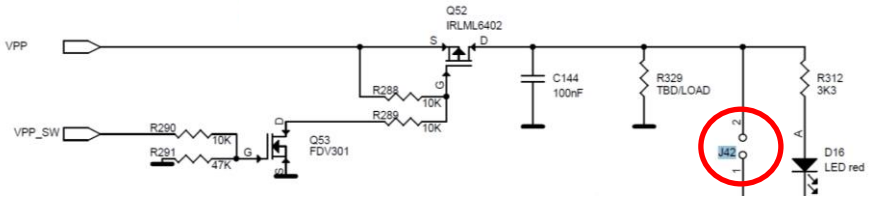



**Figure 14: VPP Control Circuit Schematic**

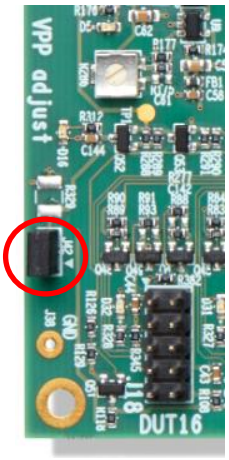



**Figure 15: Location of the VPP Jumper J42**

### 5.9.3 J47, J46 - GU Reset

For a GU hardware reset, jumper J47 should be mounted and jumper J46 should be removed. These two jumpers are involved in the circuit illustrated in Figure 11. In this way, the PLT software will control the GU hardware reset. Figure 16 shows the jumper placement on the actual PCB.

**Figure 16: Location of J46 Jumper**


© 2020 Dialog Semiconductor

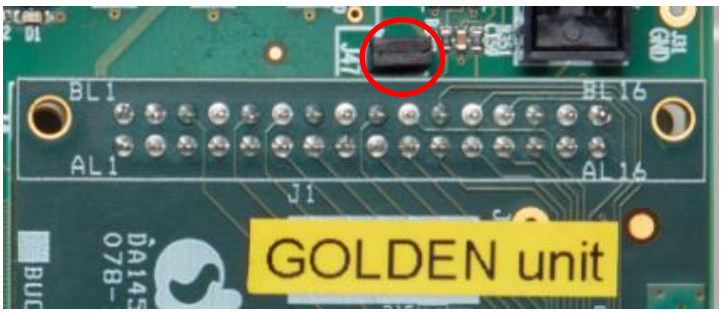



**Figure 17: Location of J47 Jumper**

### 5.9.4 J37 - GU Programming

Jumper J37 connects the Chip Select of the GU SPI Flash to a logic high level. This causes the GU not to boot from the already programmed SPI Flash, allowing the GU to load different code into its System-RAM via the JTAG connector or via UART. Figure 18 shows the circuit schematic and Figure 19 shows the location of jumper J37 on the PLT PCB.

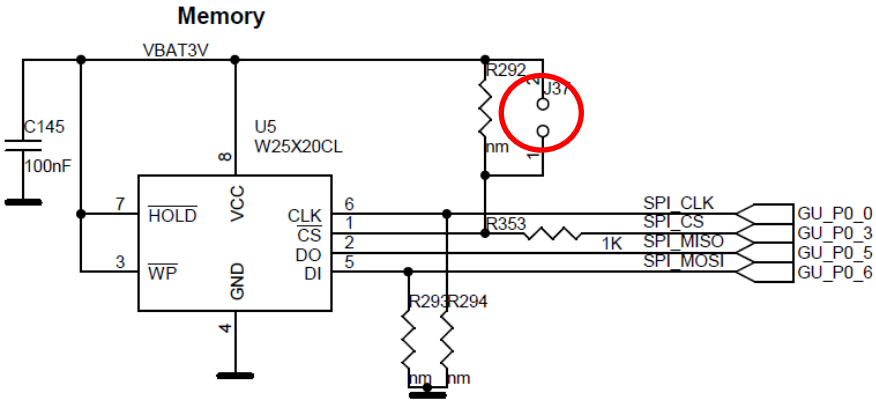



**Figure 18: J37 - GU Programming Jumper Schematics**

**Figure 19: Location of J37 Jumper**


© 2020 Dialog Semiconductor

## 5.10 PLT Functional Blocks

Figure 20 shows an overview of the PLT hardware functions. For detailed electrical schematics, see Appendix B.

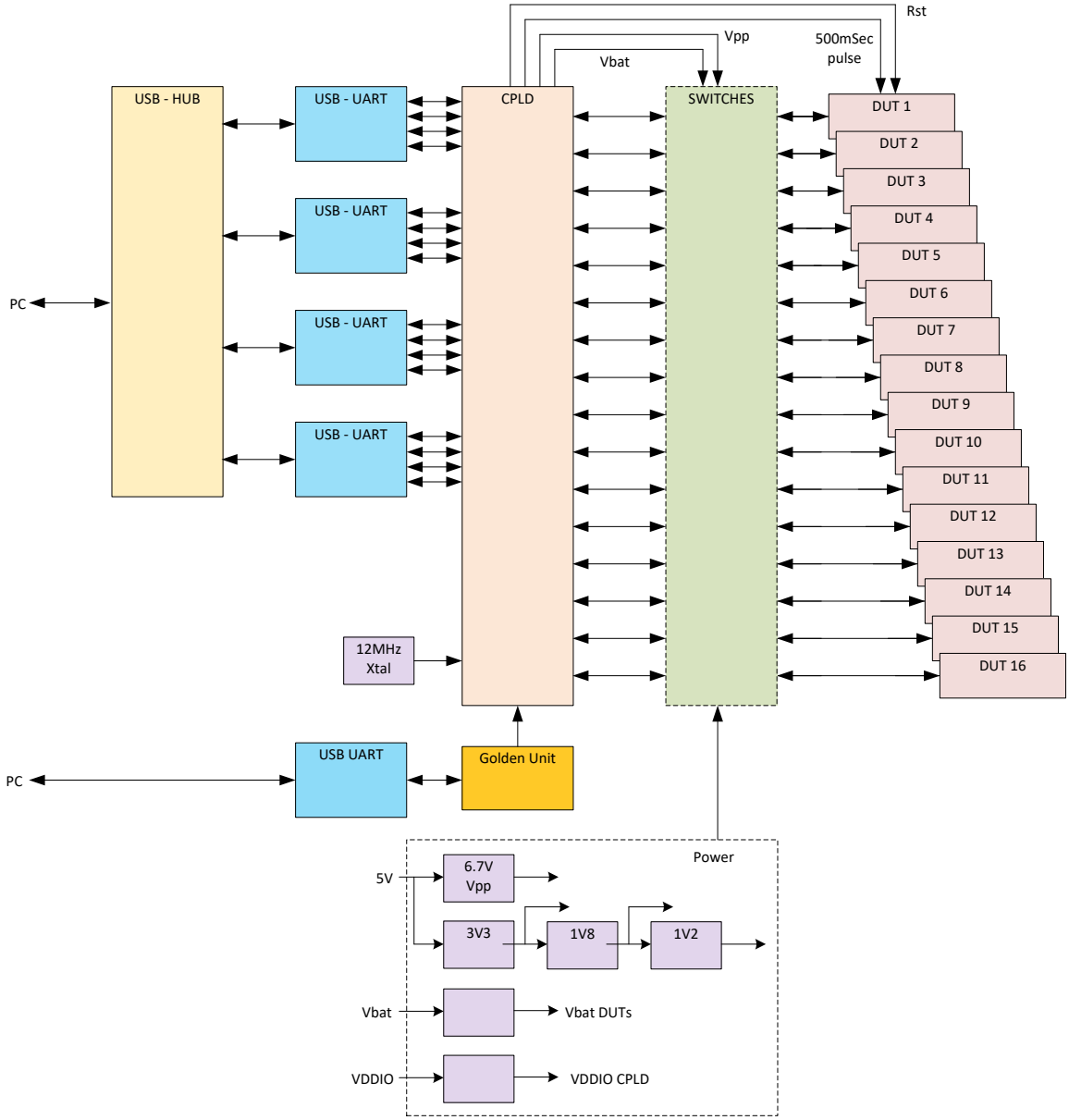



**Figure 20: PLT Functional Blocks**


© 2020 Dialog Semiconductor

# 6 Software

## 6.1 Introduction

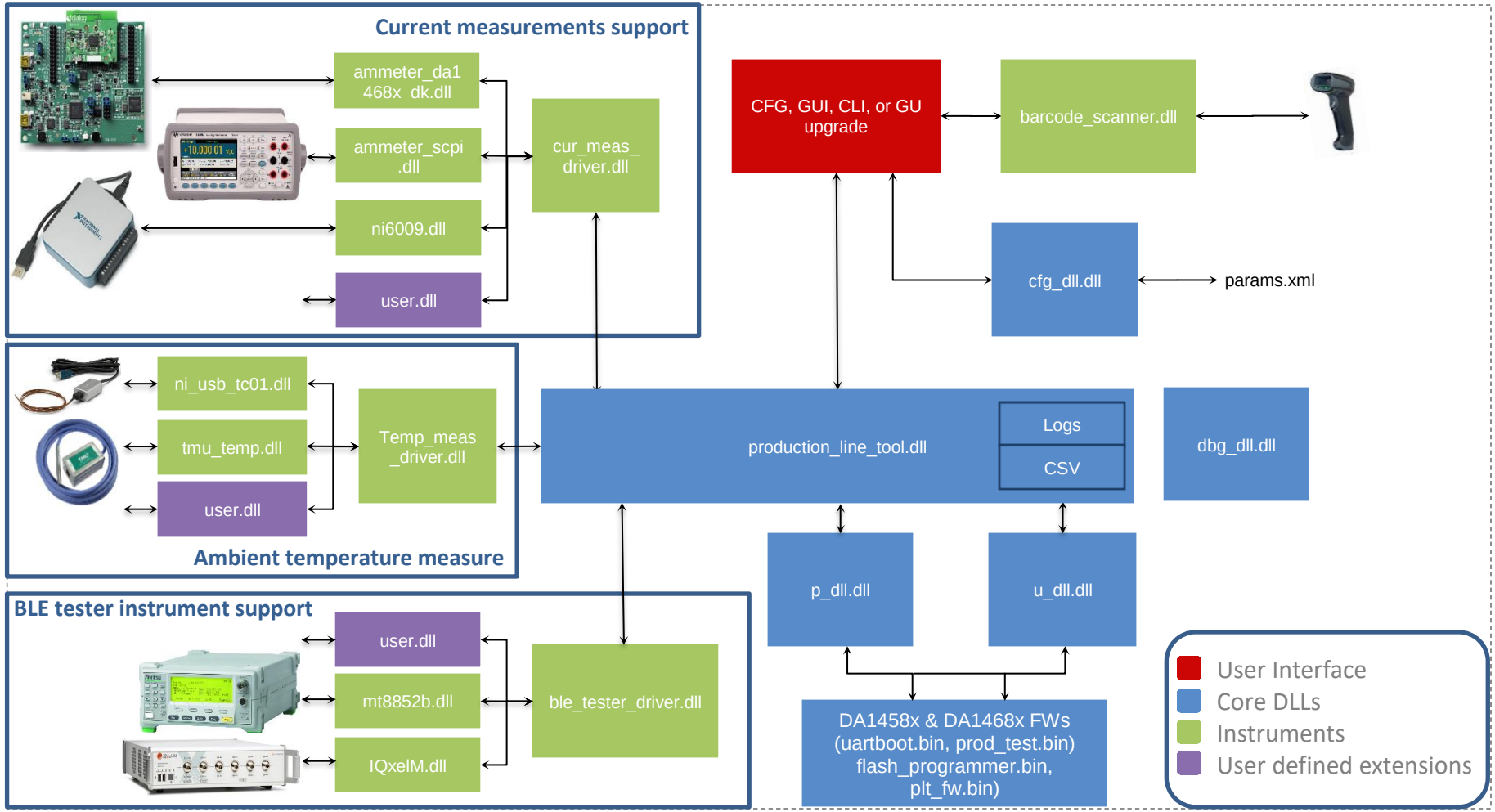



**Figure 21: Production Line Tool Software Block Diagram**


© 2020 Dialog Semiconductor

The Production Line Tool software is a collection of software blocks that interact with each other, as shown in Figure 21. Its main purpose is to communicate with the PLT hardware and the DUTs in order to run the production tests and perform memory operations. The software blocks can be arranged in four main groups:

- **Red blocks:** User Interface (UI) applications.
- **Blue blocks:** Core libraries.
- **Green blocks:** Instrument interface libraries.
- **Purple blocks:** User defined extensions.

Core libraries, instrument interface libraries and user-defined extension APIs can be found in the HTML help inside the source PLT directory. The User Interface applications block consists of four application executables. For details, see section Applications.

**Table 5: PLT User Interface Application Executables**

| Short Name | File Name | Description |
|---|---|---|
| CFG PLT | SmartBond_CFG_PLT.exe | Configuration application. Load, edit and save the test parameters and the memory actions to be performed during device testing. |
| GUI PLT | SmartBond_GUI_PLT.exe | Graphical User Interface (GUI) application. Performs the actual device validation and memory programming. Provides a visual indication of the test results and access to the result logs. |
| CLI PLT | SmartBond_CLI_PLT.exe | The same as the GUI PLT but console based. |
| GU Upgrade | GU_fw_upgrade.exe | A Graphical User Interface (GUI) application, which is used to easily upgrade the firmware of the Golden Unit. |

## 6.2 Software Package Contents

The PLT software release package comes in a compressed folder `SmartBond_PLT_v_X.X.zip`, where '`X`' represents the version number of the current PLT release.

Figure 22 illustrates the main folders of the PLT software package. Folder `executables` holds all the executables and libraries needed for the PLT to run on a Windows 7/8/8.1/10 machine. Folder `source` contains the entire source code of the PLT, organized in a Visual Studio Express 2015 solution.

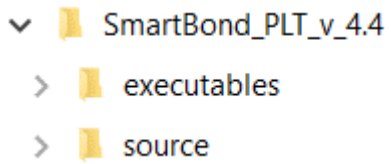



**Figure 22: SmartBond PLT Software Package Contents**

Table 6 gives a short description of the files and folders contained in the `executables` directories.

**Table 6: Executables Folder Description**

| File or Folder | Description |
|---|---|
| `ammeter_instr_plugins/` | Contains the current measurement instrument DLLs, used during the current measurement tests. |
| `ammeter_instr_plugins/ni6009.dll` | This is the DLL for the NI-6009 DAQ [14] that could be used in the current measurements. The usage of this instrument for measuring the current requires an external shunt resistor and things complicate when the measurement switches from many DUTs to one DUT. We only recommend using this instrument if one DUT per run is tested. |


© 2020 Dialog Semiconductor

| File or Folder | Description |
|---|---|
| ammeter_instr_plugins/ammeter_scpi.dll | This is the DLL for taking current measurements using a DMM that supports the standard SCPI commands. NI-VISA is also used for this purpose. Example DMM instruments are the Keysight 34401A [6], the Keithely 2000 [7] or the Keysight 34461A [15]. The PLT has been tested with all three instruments. |
| ammeter_instr_plugins/ammeter_da1468x_dk.dll | This is the DLL for taking current measurements using the current measurement module placed on the DA1469x pro motherboard. |
| binaries/ | Contains the necessary firmware binaries used during testing. Additionally, the SetupCode_Generator_680.exe application used in homekit products to create a specific hash key is included. |
| binaries/GU/prod_test_GU.bin | Contains the Golden Unit latest firmware binary. Users should better upgrade their PLT hardware with the GU firmware contained in this folder. |
| ble_tester_instr_plugins/ | Contains the BLE tester instrument DLLs. |
| ble_tester_instr_plugins/mt8852b.dll | This is the DLL that performs the Direct Test Mode RF tests using the Anritsu MT8852B instrument [5].<br>**Note:** There is an issue in Anritsu MT8852B firmware version 4.20.000 and should be upgraded to the latest one. Latest MT8852B instrument firmware can be downloaded from the following link:<br>https://www.anritsu.com/en-US/test-measurement/support/downloads?model=MT8852B |
| ble_tester_instr_plugins/IQxelM.dll | This is the DLL that performs the Direct Test Mode RF tests using the Litepoint IQxel-M instrument [13]. |
| icons/ | Contains pictures used by the PLT applications. |
| IQmeasure_3.1.2/ | Contains specific Litepoint IQxel-M DLLs as released by Litepoint. |
| params/ | Contains the configuration params.xml file, the XML schema params.xsd and a sample of BD address file named bd_address.ini. |
| params/custom_mem_data.csv | This is a sample CSV file to be used in the custom memory burn action. Users could edit this file and add their own specific memory data to be burned by the PLT. The PLT will match the entries in the CSV file using the BD addresses. The format of the file is explained later. |
| scripts/ | Contains sample batch script files. User can select batch script files to be executed by the PLT before and after each test. |
| scripts/run_before_tests.cmd | An example script that copies and renames binaries from a directory to a folder required by the PLT when 'Different image per DUT' is selected. This folder is accessed by the PLT to read and burn different binary per DUT. |
| scripts/run_after_tests.cmd | An example script that moves all logs files, except the ones with the current date, to a specific folder. |
| temp_meas_instr_plugins/ | Contains the temperature measurement instrument DLLs. |
| temp_meas_instr_plugins/ni_usb_tc01.dll | The ni_usb_tc01.dll is the DLL used to interface a NI USB TC01 [9] temperature sensor for temperature measurements. |

 © 2020 Dialog Semiconductor

| File or Folder | Description |
|---|---|
| temp_meas_instr_plugins/tmu_temp_sens.dll | The tmu_temp_sens.dll is the DLL used to interface a Papouch TMU sensor [7] for temperature measurements. |
| volt_meter_instr_plugins/ | Contains the voltage meter instrument DLLs. These are used only in DA14681-00 silicon for ADC calibration purposes. |
| volt_meter_instr_plugins /volt_meter_scpi.dll | The volt_meter_scpi.dll is a DLL that implements basic interface with a DVM using SCPI commands through NI-VISA libraries and GPIB interface. Has been tested with Keithley 2000 [7] and Keysight 34401A [6]. |
| SmartBond_CFG_PLT.exe | This is the configuration application. It is a graphical user interface application used to edit the PLT test configuration parameters, saved in an XML file, params.xml. |
| SmartBond_CLI_PLT.exe | This is the command line interface tool. It performs the production tests and memory programming through a console. |
| SmartBond_GUI_PLT.exe | This is the graphical user interface tool. It performs the production tests and memory programming through a graphical user interface. |
| GU_fw_upgrade.exe | This is the Golden Unit firmware upgrade application. |
| ammeter_driver.dll/.lib | This DLL loads and accesses all DMM instrument DLLs from inside the ammeter_instr_plugins. It acts as an intermediate layer between the prod_line_tool_dll and the instrument DLLs. |
| barcode_scanner.dll/.lib | This DLL receives BD addresses from a barcode scanner with USB to serial interface. Has been tested with Honeywell Xenon 1900 and the Motorola LS2208 barcode scan readers [10] [11]. |
| ble_tester_driver.dll/.lib | This DLL loads and accesses all BLE tester instrument DLLs from inside ble_tester_instr_plugins folder. |
| cfg_dll.dll/.lib | This is the configuration parameter handling DLL. It can validate, load and save parameters from a given XML file. |
| dbg_dll.dll/.lib | The dbg_dll.dll file is a DLL used to print debug messages to a file or to a debug console. |
| ftd2xx.dll | This is the FTDI DLL. Used to hard reset the Golden Unit from the application whenever needed through an FTDI GPIO pin. |
| p_dll.dll/.lib | This is the production test DLL that performs device functional tests. |
| prod_line_tool_dll.dll/.lib | This is the core DLL. The heart of the system that performs the state machines for all tests and memory actions to be executed. It is responsible to log the results and notify the user interfaces about the current device test status. |
| temp_meas_driver.dll/.lib | This is the temperature measurement driver DLL. It loads and accesses all temperature measurement DLLs from inside the temp_meas_instr_plugins folder. |
| u_dll.dll/.lib | This is the DLL that performs the memory actions, like the memory programming, erasing, etc. |


© 2020 Dialog Semiconductor

| File or Folder | Description |
|---|---|
| `vc_redist.x86.exe/vc_redist.x64.exe` | These are the Visual Studio 2017 Express redistributable packages for 32 and 64-bit machines. For installing these, users should agree to the license requirements described during the installation of any of these packages and also found here: https://www.visualstudio.com/license-terms/mt171551/. |
| `volt_meter_driver.dll/.lib` | This is the voltage meter driver DLL. It loads and accesses all voltage meter DLLs from inside the `volt_meter_instr_plugins` folder. |

## 6.3 Prerequisites

Before building and running the code, the packages indicated in Table 7 should be installed on the PC. Some are required and others are optional depending on the tests or actions needed.

**Table 7: Production Line Tool Prerequisites**

| Item | Optional | Description |
|---|---|---|
| Visual Studio 2017 Express | Yes | The IDE used to edit and debug the Production Line Tool. This is only required if users would like to edit the software. |
| vc_redist.x86.exe | No | Already described in Table 6. Users should agree to the license requirements described during the installation of any of these packages and also found here: https://www.visualstudio.com/license-terms/mt171551/. |
| MSXML6 | No | Installed by default in Win 7/8/8.1/10. |
| .NET framework 4.5 | No | Needed for the graphical user interface applications. |
| Latest FTDI drivers | No | Tested with FTDI v2.12.24, v2.12.26 and v2.12.28 drivers. |
| Honeywell Xenon 1900 drivers | Yes | Needed if the barcode scanner is going to be used for scanning the devices BD addresses and/or custom memory data. Other types of barcode scanners could also be used. |
| Motorola LS2208 drivers | Yes | Used if a barcode scanner is going to be used for scanning the device BD addresses and/or custom memory data. |
| NI-VISA 15.5 | Yes | Used for optional instrument control, like BLE tester and voltage meter.<br>NI-VISA 15.5 can be downloaded from http://www.ni.com/download/ni-visa-15.5/5846/en/ |
| NI-488.2 15.5 | Yes | Used for instrument control, like BLE tester and DMM.<br>NI-488.2 15.5 can be downloaded from http://www.ni.com/download/ni-488.2-15.5/5859/en/ |
| NI-DAQmx | Yes | Used for optional instrument control like temperature measurements using the NI USB TC01 sensor. |

 © 2020 Dialog Semiconductor

## 6.4 System Requirements

Table 8 contains the minimum system requirements for the PLT to operate.

**Table 8: Minimum System Requirements**

| Item | Minimum Requirements | |
|---|---|---|
| Operating system | Windows 7/8/8.1/10 | |
| CPU | Quad Core CPU | |
| Memory | 4 GB RAM or larger. Each device log can reach up to 40 kB. | |
| Hard drive | For 100000 devices, at least **4 GB** of available hard disk is required. | |
| Monitor resolution | 1280 x 768 or higher | |
| Monitor DPI | Smaller - 100% = 96 DPI | Supported |
| | Medium - 125% = 120 DPI | Supported |
| | Larger - 150% = 144 DPI | Not supported |

## 6.5 Limitations

**Parallel control** of multiple PLT hardware boards on the same PC is **not supported**.

However, by correctly setting up the system, two or more PLT hardware boards could be connected and controlled by multiple GUI PLT application instances on the same PC, but the tests should only be executed **sequentially**. The main reasons for this limitation are indicated below:

- The GU FT232 FTDI IC is programmed to have a special serial string, "`DialogSemi`" (see Table 108). This is used in the '`GU COM port find`' PLT operation. This operation searches all PC connected FTDIs to find the serial string "`DialogSemi`". When found, it saves it as the GU COM port number to be used by the PLT. The '`GU COM port find`' operation will open and lock, for a short period of time, all Windows COM ports, one by one, even the ones used by the other PLT hardware. If the second GUI PLT application instance is performing test operations at the same time and wants to open its DUT COM ports, the operation may fail.
- When the GUI PLT application starts the test operations, it performs a DUT COM port enumeration. During this process, the GU sets the CPLD in UART loopback mode. It opens all PC COM ports one by one and sends a specific word, while trying to see if it receives it back. During this process, other PLTs may need to work with 'their' DUT COM ports, which may happen to be currently used by the 'DUT COM port enumeration' process of the first PLT.
- GU hardware reset. In every PLT test run a GU HW reset is issued from the PLT software using a specific GU FTDI GPIO pin. In order to access the GU FTDI, the FTDI API is used from `ftd2xx.dll`. To access the FTDI hardware and read the serial number through the FTDI `ftd2xx.dll` the `FT_Open` API is used on all PC COM ports, one by one. Since `FT_Open` is used in all PC COM ports, conflicts could arise if other PLTs would also like to use these COM ports.
- BD addresses handling. Usually, the PLT automatically sets the DUT BD addresses by increasing them one by one. Special care should be taken to work with multiple PLT hardware and software. Most probably, two different BD address files should be used for each PLT hardware.

 © 2020 Dialog Semiconductor

## 6.6 Building the Code

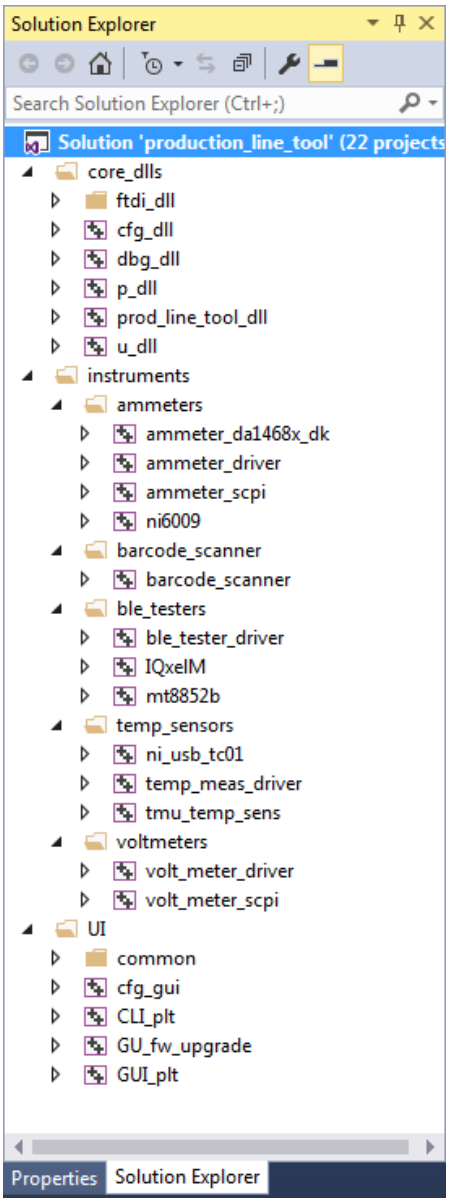
The PLT software release package contains not only application executables for directly performing the tests out of the box, but also the entire source code of the tools. This is organized in a Visual Studio 2017 Express solution.

To open the Visual Studio 2017 Express PLT source code solution the following steps should be executed (see Table 9).

**Table 9: Opening the PLT Visual Studio 2017 Express Source Code Solution**

| Step | Description |
|---|---|
| 1 | Download the latest PLT software package (e.g. `SmartBond_PLT_v_4.x.zip`) |
| 2 | Extract the software package. The following two folders should exist.<br>› SmartBond_PLT_v_4.4<br>Name<br>executables<br>source<br>licensing.txt |
| 3 | Go to folder '`source\production_line_tool`'. The following files and folders should exist.<br>› SmartBond_PLT_v_4.4 › source › production_line_tool ›<br>Name<br>.vs<br>core_dlls<br>DA14531_DA1469x_Release<br>Debug<br>fw_files<br>help<br>instruments<br>UI<br>VS2017_redist<br>production_line_tool.sln |


© 2020 Dialog Semiconductor

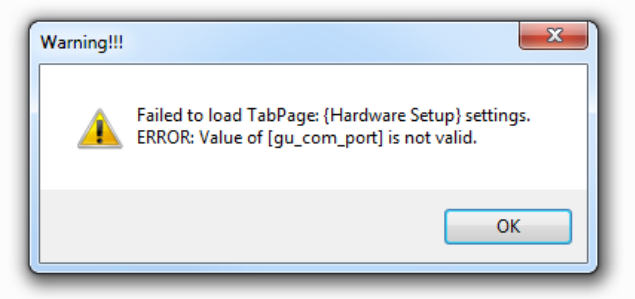
| Step | Description |
|---|---|
| 4 | Double click the `production_line_tool.sln` Visual Studio 2017 Express solution file. The Visual Studio 2017 Express application will start and the PLT Solution Explorer should be shown.<br>  <br> |
| 5 | To build the code follow the process described in the *UM-B-040* document [1].<br>**Note:** There is no need to unload any projects for the solution to be build. This was fixed in PLT v4.4 by dynamically linking DLLs. |

 © 2020 Dialog Semiconductor

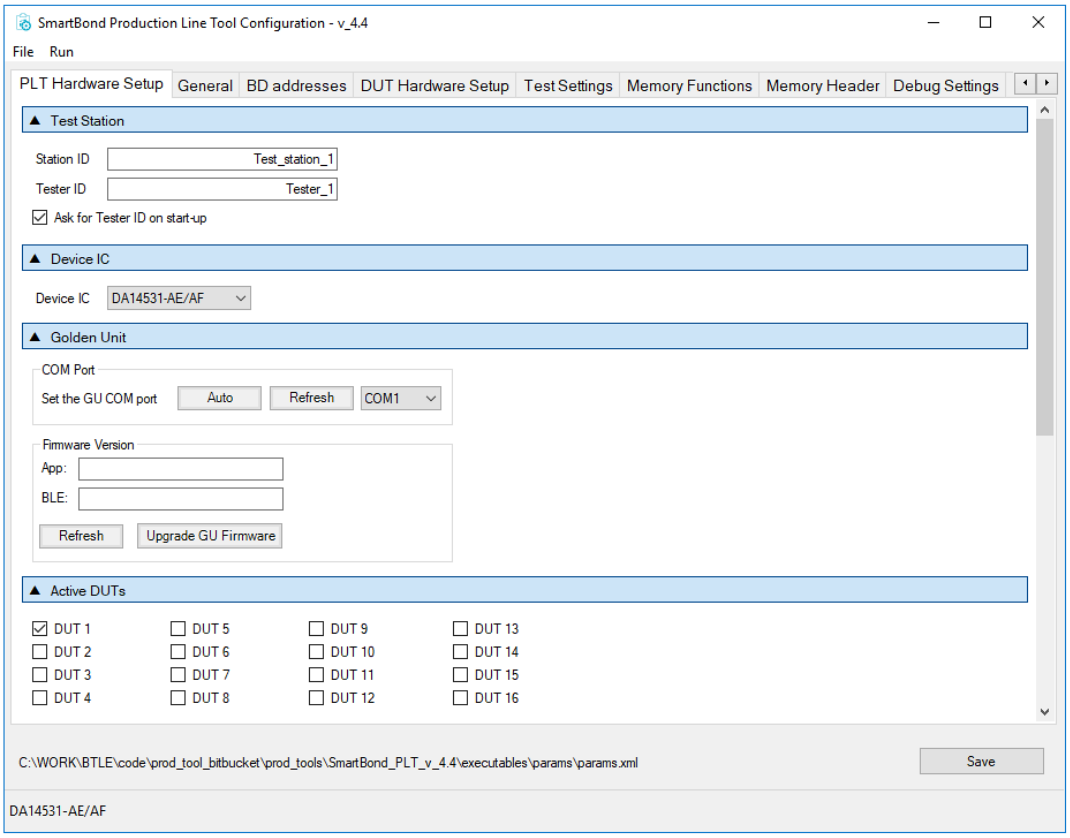
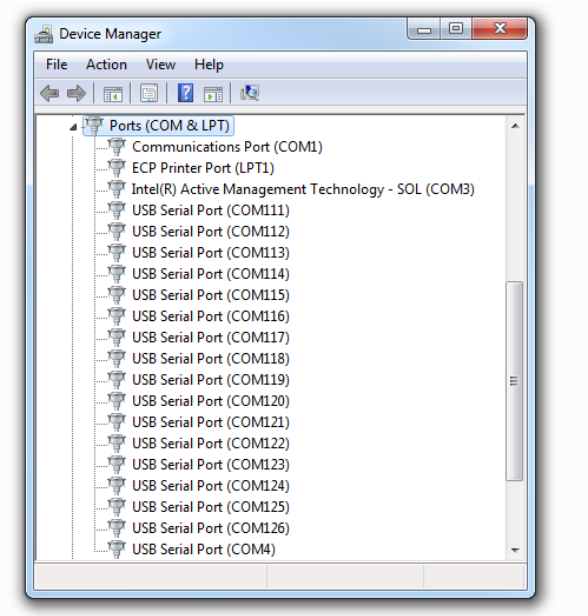
## 6.7 Executing the Applications

To execute the Production Line Tool applications, the process described in the following tables should be followed.

**Table 10: SmartBond_CFG_PLT.exe Application Execution**

| Step | Description |
|---|---|
| 1 | Download the latest PLT software package (e.g. `SmartBond_PLT_v_x.x.zip`). |
| 2 | Extract the software package. The following two folders should exist.<br>› SmartBond_PLT_v_4.4<br>Name<br>executables<br>source<br>licensing.txt |
| 3 | Go to folder '`executables`'. This folder should contain the following files and sub-folders.<br>› SmartBond_PLT_v_4.4 › executables<br>ammeter_instr_plugins<br>binaries<br>ble_tester_instr_plugins<br>icons<br>IQmeasure_3.1.2<br>params<br>scripts<br>temp_meas_instr_plugins<br>volt_meter_instr_plugins<br>GU_fw_upgrade.exe<br>SmartBond_CFG_PLT.exe<br>SmartBond_CLI_PLT.exe<br>SmartBond_GUI_PLT.exe<br>vc_redist.x86.exe<br>ammeter_driver.dll<br>barcode_scanner.dll<br>ble_tester_driver.dll<br>cfg_dll.dll<br>dbg_dll.dll<br>ftd2xx.dll<br>ni_daqmx_shim.dll<br>ni_visa_shim.dll<br>p_dll.dll<br>prod_line_tool_dll.dll<br>temp_meas_driver.dll<br>u_dll.dll<br>volt_meter_driver.dll<br>ammeter_driver.lib<br>barcode_scanner.lib<br>ble_tester_driver.lib<br>cfg_dll.lib<br>dbg_dll.lib<br>ftd2xx.lib<br>ni_daqmx_shim.lib<br>ni_visa_shim.lib<br>p_dll.lib<br>prod_line_tool_dll.lib<br>temp_meas_driver.lib<br>u_dll.lib<br>volt_meter_driver.lib |
| 4 | Double click the `SmartBond_CFG_PLT.exe` application executable. Most probably, the following warning will be shown.<br>  <br><br>During start-up, the `SmartBond_CFG_PLT.exe` application loads the `Hardware` configuration parameters from the `params.xml` file. These parameters also contain the GU COM port. The default `params.xml` file has the GU COM port set to 4. If this COM port number does not exist in the PC, then this warning message will be shown. Therefore, this warning message indicates that the default GU COM port set in the `params.xml` file is not valid or that the GU USB cable is not connected to the PC.<br>Press 'OK' if the warning message appears. |


© 2020 Dialog Semiconductor

| Step | Description |
|---|---|
| 5 | The application will start, and the initial *Hardware Setup* screen will be shown.   |
| 6 | Connect the PLT HW to the PC. Connect the GU and the DUT USB cables to the PC. Check the Windows Device Manager that 17 new COM ports were found, 16 for the DUTs and 1 for the GU. The following picture is an example of a Device Manager COM ports for a PC that has the PLT connected.   |

 © 2020 Dialog Semiconductor

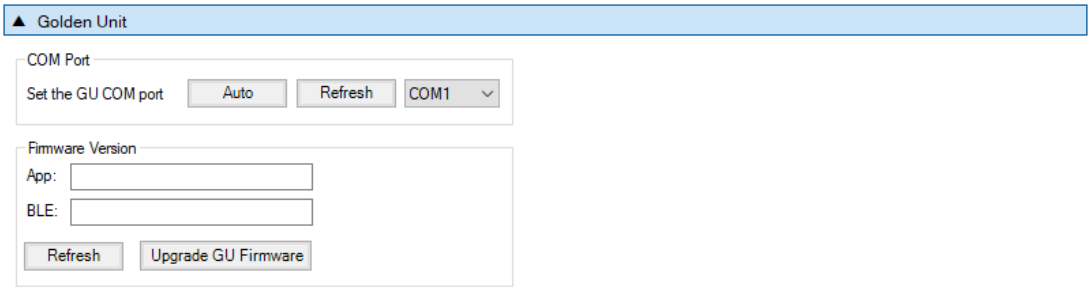
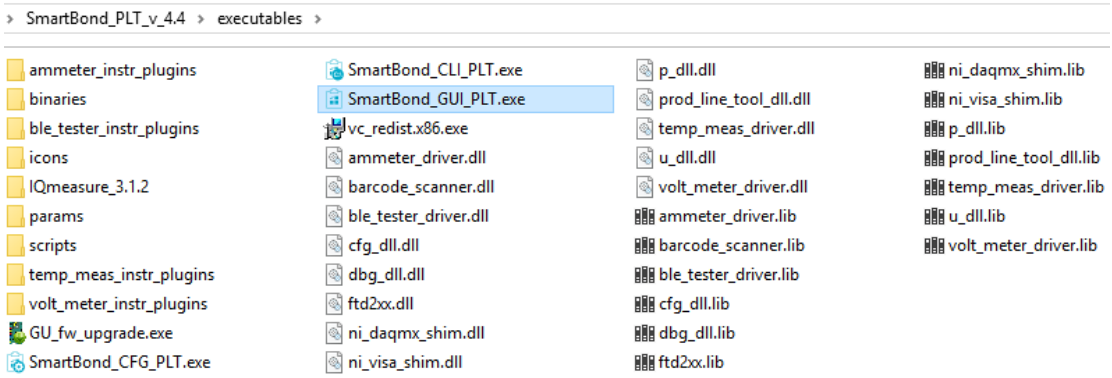
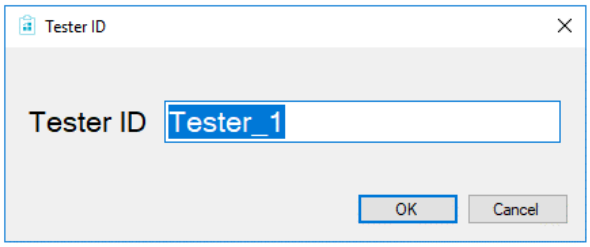
| Step | Description |
|---|---|
| 7 | On the `SmartBond_CFG_PLT.exe` Hardware Setup initial screen press, **Auto** to automatically find the GU COM port among the 17 Windows enumerated COM ports. The **Auto** button will turn green if successful. Press **Save*** to save the new GU COM port in the `params.xml` file.   |

**Table 11: SmartBond_GUI_PLT.exe Application Execution**

| Step | Description |
|---|---|
| 1 | To successfully start the `SmartBond_GUI_PLT.exe` application, the `SmartBond_CFG_PLT.exe` should be executed first in order to set up the system and perform the required tests. See Table 10. |
| 2 | Go to folder '`executables`'. This folder should contain the following files and sub-folders.<br>Double click on `SmartBond_GUI_PLT.exe`.   |
| 3 | A new window will ask for the Tested ID name. Enter the name and press OK.   |

 © 2020 Dialog Semiconductor

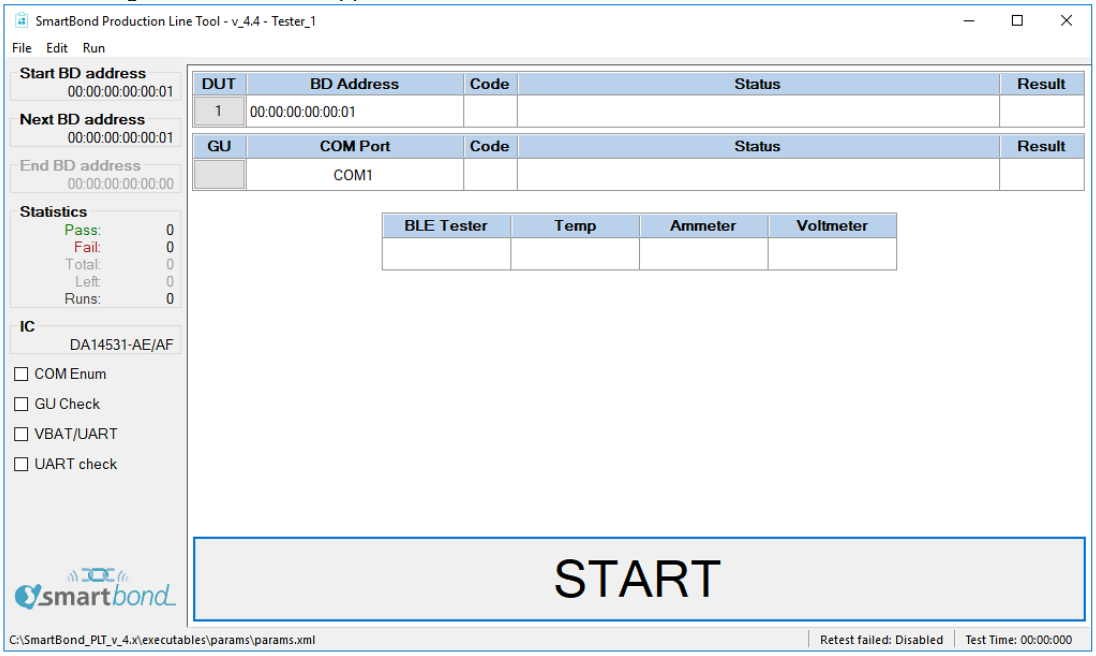
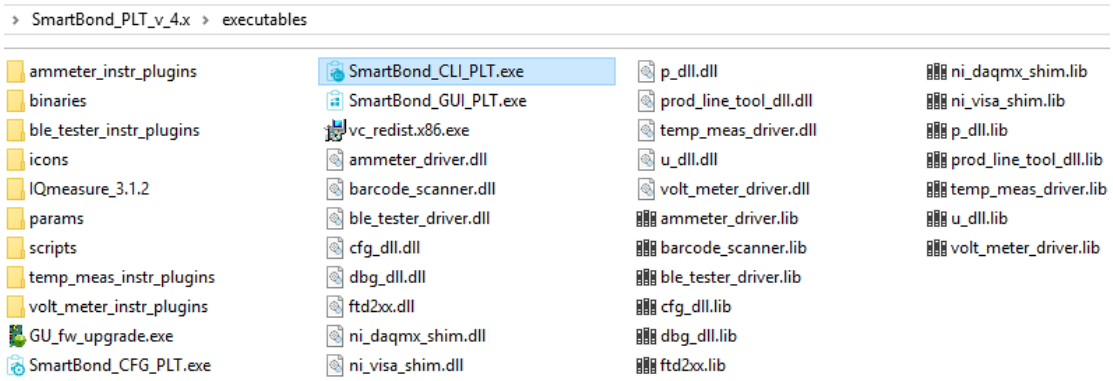
| Step | Description |
|---|---|
| 3 | The following initial screen will appear.<br>  <br> |
| 4 | By clicking the keyboard spacebar, the **START** button will be pressed and the preconfigured tests and memory actions will start to be executed. |

**Table 12: SmartBond_CLI_PLT.exe Application Execution**

| Step | Description |
|---|---|
| 1 | To successfully start the `SmartBond_CLI_PLT.exe` application, the `SmartBond_CFG_PLT.exe` should be executed first, in order to set up the system and perform the required tests. See Table 10. |
| 2 | Go to folder '`executables`'. This folder should contain the following files and sub-folders.<br>  <br> |

 © 2020 Dialog Semiconductor

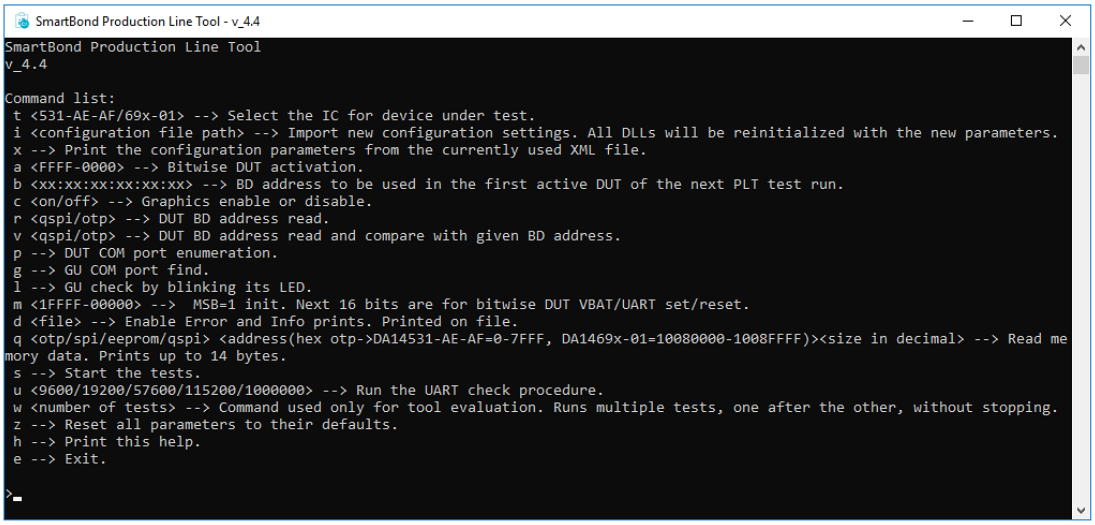
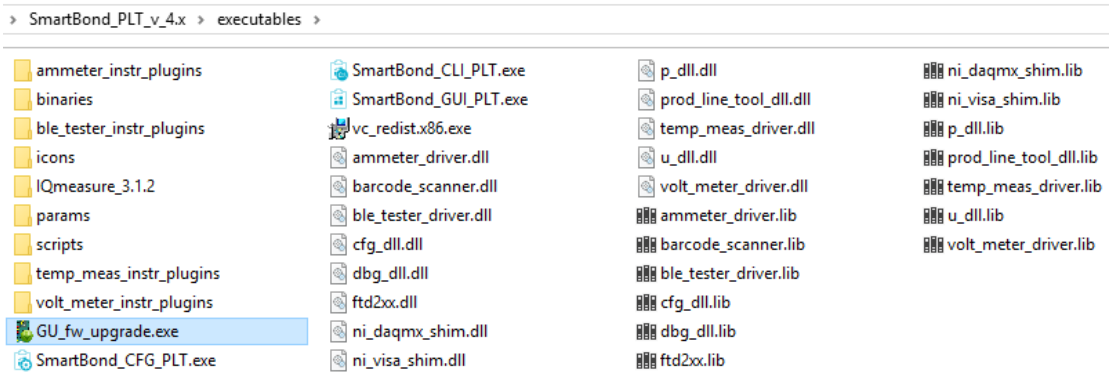
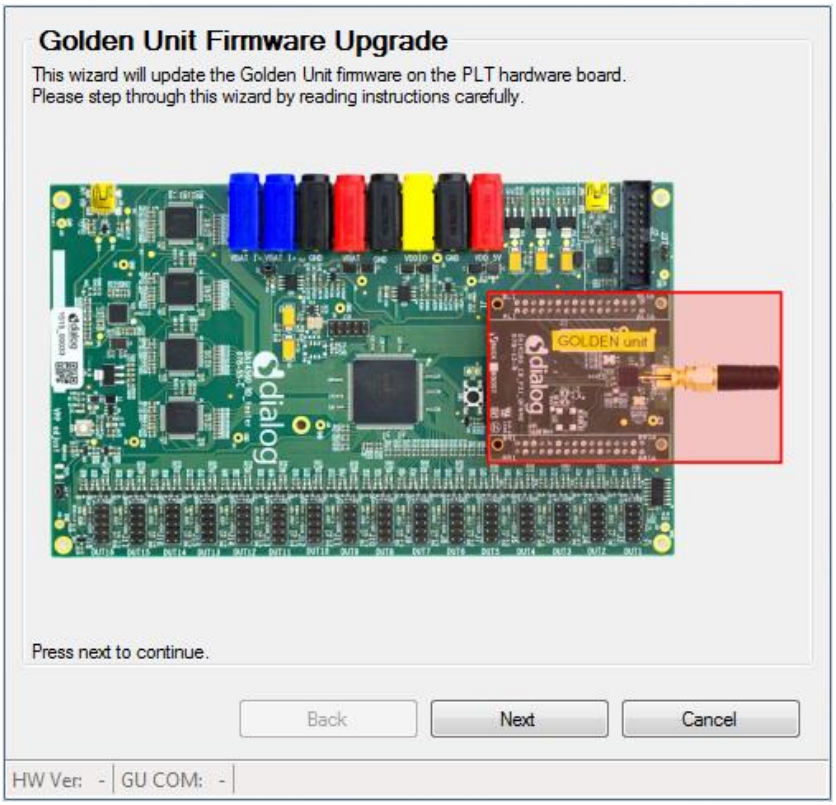
| Step | Description |
|---|---|
| 3 | Double click the `SmartBond_CLI_PLT.exe` application executable. The following initial screen will appear.<br>  <br> |
| 4 | Type '`s`' and then press Enter. The preconfigured tests and memory actions will start to be executed. |

**Table 13: GU_fw_upgrade.exe Application Execution**

| Step | Description |
|---|---|
| 1 | `GU_fw_upgrade.exe` can be started by either opening the application from the executables folder or by pressing the '`Upgrade GU Firmware`' button in the `PLT Hardware Setup` tab in `SmartBond_CFG_PLT.exe`. |
| 2 | Go to folder '`executables`'. This folder should contain the following files and sub-folders.<br>  <br> |

 © 2020 Dialog Semiconductor

| Step | Description |
|---|---|
| 3 | Double click the `GU_fw_upgrade.exe` application executable. The following initial screen will appear.   |
| 4 | Follow the instructions to guide you to successfully configure and select the Golden Unit. A detailed procedure is given in Chapter 7.5. |

## 6.8 Test Sequence

This section describes the sequence of steps involved for the DA14531 and DA1469x device testing. It outlines all the steps the PLT follows to successfully test a device.

### 6.8.1 DA1458x Test Sequence

Table 14 describes each step the PLT undertakes for DA1453x devices. Some of the steps are optional and will only be executed if the equivalent actions are enabled in the configuration parameters. Additionally, some of the steps are supported only for specific DA1458x IC versions.

The entire test sequence for the DA1453x DUTs is shown in Figure 23.

**Table 14: DA14531 Test Sequence**

| Step | Action | Opt | Description |
|---|---|---|---|
| 1 | Statistics update | No | Update the total tests executed. |
| 2 | BD addresses | No | Update the BD addresses for all DUTs. |
| 3 | Configuration parameters | No | Configuration parameters are passed from the CLI or GUI to the `prod_line_tool_dll`. If any of the parameters is not valid, an error will occur. |
| 4 | Reset GU | No | Golden Unit hardware reset by controlling an FT232 pin. |
| 5 | Initialize CPLD | No | Set the CPLD to an initial known state. |
| 6 | Check temperature sensor instrument | Yes | Check whether the temperature measurement instrument is online, only if the temperature measurement test is active. |

 © 2020 Dialog Semiconductor

| Step | Action | Opt | Description |
|---|---|---|---|
| 7 | Check BLE tester instrument | Yes | Check whether the BLE tester instrument is online, only if any of the BLE tester test operations is active. |
| 8 | Check ammeter instrument | Yes | Check whether the ammeter instrument is online, only if any of the current measurement tests is active. |
| 9 | Toggle GU LED | No | Toggle the GU red LED on the PLT hardware to indicate that the GU is alive. |
| 10 | Check DUT COM ports | No | Check whether the PLT has identified the DUT COM ports and if not run the automatic DUT COM port identification. |
| 11 | Temperature measure | Yes | If the temperature measurement test is active, take a measurement and log it to all DUT logs and in the CSV file. |
| 12 | Download `prod_test_531.bin` | Yes | If any of the production tests is active (e.g. RF tests, XTAL trim, etc.) download the `prod_test_531.bin` to the devices. |
| 13 | Open the devices COM ports and get the `prod_test_531.bin` firmware version. | Yes | After `prod_test_531.bin` has been downloaded to the DUTs, test commands can be sent to it. First, the Windows DUTs COM ports are opened. Then a command to get the `prod_test_531.bin` firmware version is sent to the devices. If there is a problem in the firmware or in the device, then this is the first failure to happen. The FW version get action will fail. |
| 14 | GPIO Watchdog | Yes | If the GPIO watchdog option is enabled, the firmware will begin a periodic GPIO toggling during the whole production test procedure. |
| 15 | VBAT level measure | Yes | PLT will send a command to each DUT to measure VBAT for each one, using the internal ADC. VBAT level will be logged for debugging purposes. |
| 16 | OTP Timestamp read | Yes | PLT will send a command to measure the DA14531 IC production date and time from the OTP. It will log it for testing purposes. |
| 17 | DC-DC Converter level test | Yes | The first test is to measure the DC-DC converter level using the internal ADC. If the level is outside the user defined limits a possible HW error exists. |
| 18 | XTAL trim | Yes | Perform the XTAL trim procedure, if this is active. |
| 19 | XTAL trim OTP burn | Yes | If the 'Burn to OTP' option is selected in the CFG PLT, then the calculated XTAL trim value will be burned to the OTP Header. |
| 20 | UART resync | Yes | If the XTAL trim procedure was performed in the UART RX pin, then a special UART resync procedure takes place to resynchronize the device's UART RX path, as it may have entered into a baud rate error state due to the 500ms XTAL trim pulse received. |
| 21 | Scan advertise test | Yes | If the Scan DUT Advertise test is active, a BLE scan test using HCI triggered advertisements will be performed. This is to measure DUTs TX power level. |
| 22 | BLE tester TX power | Yes | If the BLE tester TX Power test is active, then perform the test using the external BLE tester instrument. |
| 23 | BLE tester TX carrier offset | Yes | If the BLE tester TX carrier offset test is active, then perform the test using the external BLE tester instrument. |
| 24 | BLE tester TX modulation index | Yes | If the BLE tester TX modulation index test is active, then perform the test using the external BLE tester instrument. |
| 25 | BLE tester RSSI | Yes | If the BLE tester RSSI test is active, then perform the test using the external BLE tester instrument. |
| 26 | GU RSSI test | Yes | If the RSSI test using the GU as transmitter is active, then perform the test. |
| 27 | GPIO/LED | Yes | Perform the GPIO/LED test, if the test is active. |
| 28 | GPIO connection test | Yes | Perform a GPIO continuity or voltage level test, if the test is active. |

 © 2020 Dialog Semiconductor

| Step | Action | Opt | Description |
|---|---|---|---|
| 29 | Sensor test | Yes | Perform the sensor tests only if these are enabled. |
| 30 | Custom test | Yes | Perform any active custom test. |
| 31 | External 32kHz | Yes | Check whether the external 32kHz crystal operates correctly. |
| 32 | Current measure peripheral | Yes | Perform any active current measurement test for peripherals. |
| 33 | Current measure sleep | Yes | Perform the sleep current measurement. |
| 34 | Open COM port and perform firmware download | Yes | If any memory action is active (e.g. SPI Flash burn, erase etc.), download the `flash_programmer_531.bin` to the devices. |
| 35 | Get `flash_programmer.bin` version. | Yes | After `flash_programmer_531.bin` has been downloaded, commands can be sent. A command to get the `flash_programmer_531.bin` firmware version is sent to the devices. |
| 36 | GPIO watchdog | Yes | If the GPIO watchdog option is enabled, the firmware will begin a periodic GPIO toggling during the whole memory programming procedure. |
| 37 | Initialize SPI Flash memory | Yes | If any SPI flash operation is enabled, initialize memory. |
| 38 | SPI erase | Yes | Erase the SPI Flash, either entirely or part of it depending on the configuration. |
| 39 | SPI check empty | Yes | Depending on the configuration, check whether the SPI Flash is empty to verify the Flash erase procedure. |
| 40 | SPI image write | Yes | If enabled, write the SPI Flash with the customer image. If verify is enabled, the contents of the Flash will be read back and compared to the original image downloaded. |
| 41 | Initialize I2C EEPROM memory | Yes | If any EEPROM operation is enabled, initialize memory. |
| 42 | I2C EEPROM write | Yes | Write the I2C EEPROM with the customer image. If verify is enabled, the contents of the EEPROM will be read back and compared to the original image downloaded. |
| 43 | Custom memory data | Yes | Write custom memory data, taken from a barcode scanner, entered manually or through a CVS file. |
| 44 | OTP image write | Yes | Write the OTP image with the customer image. If verify is enabled, the contents of the OTP memory will be read back and compared to the original image downloaded. |
| 45 | OTP CS write | Yes | If enabled, the OTP configuration script will be burned. |
| 46 | OTP header write | Yes | If enabled, the OTP header fields will be burned. |
| 47 | Memory read | Yes | Up to ten memory read tests can be performed. |
| 48 | Scan test | Yes | If enabled, the GU will scan for device BLE advertisements. For the DUTs to be scanned a valid firmware must be burned into the OTP, SPI Flash or EEPROM that sends BLE advertisements after power up. Additionally, the BD address should be burned into the OTP by the PLT. PLT expects to find devices in the air with the BD addresses programmed by the same tool, so it can match the BD addresses returned by the GU. |

 © 2020 Dialog Semiconductor

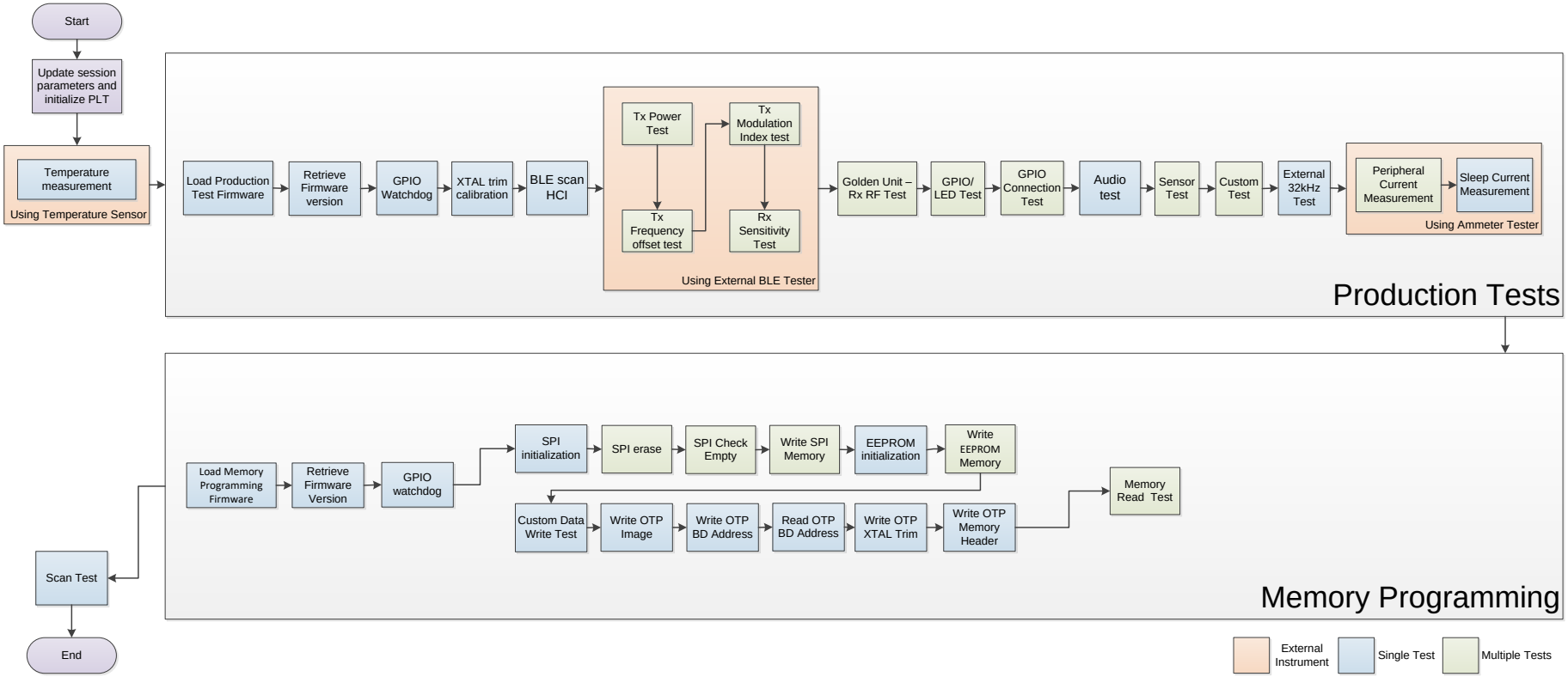



**Figure 23: DA14531 Test Sequence**


© 2020 Dialog Semiconductor

### 6.8.2 DA1469x Test Sequence

Table 15 describes each step that the PLT undertakes to validate and program DA1469x based devices. Some of the steps are optional and will only be executed if the equivalent actions are enabled in the configuration parameters. Additionally, some of the steps are supported only for specific DA1469x IC versions.

The entire test sequence for the DA1469x DUTs is shown in Figure 24.

**Table 15: DA1469x Test Sequence**

| Step | Action | Opt | Description |
|---|---|---|---|
| 1 | Statistics update | No | Update the total tests executed. |
| 2 | BD addresses | No | Update the BD addresses for all DUTs. |
| 3 | Configuration parameters | No | Configuration parameters are passed from the CLI or GUI to the `prod_line_tool_dll`. If any of the parameters is not valid, an error will occur. |
| 4 | Reset GU | No | GU hardware reset by controlling an FT232 pin. |
| 5 | Initialize CPLD | No | The GU will set the CPLD to an initial known state. |
| 6 | Check temperature sensor instrument | Yes | Check whether the temperature measurement instrument is online, only if the temperature measurement test is active. |
| 7 | Check BLE tester instrument | Yes | Check whether the BLE tester instrument is online, only if any of the BLE tester test operations is active. |
| 9 | Check ammeter instrument | Yes | Check whether the ammeter instrument is online, only if any of the current measurement tests is active. |
| 10 | Toggle GU LED | No | Toggle the GU red LED on the PLT hardware to indicate that the GU is alive. |
| 11 | Check DUT COM ports | No | Check whether the PLT has identified the DUT COM ports and if not run the automatic DUT COM port identification. |
| 12 | Temperature measure | Yes | If the temperature measurement test is active, take a measurement and log it to all DUT logs and in the CSV file. |
| 13 | Download `uartboot_69x.bin` | Yes | If any of the production tests is active the `uartboot_69x.bin` will be downloaded and then the production test firmware. In addition, if the GPIO watchdog option is enabled, it will start toggling after the `uartboot_69x.bin` is loaded and right before the production test download. |
| 14 | GPIO Watchdog | Yes | If the GPIO watchdog option is enabled, then firmware will start the toggling after the `uartboot_69x.bin` is loaded and right before the production test download. |
| 15 | Download `prod_test_69x.bin` | Yes | If any of the production tests is active (e.g. RF tests, XTAL trim, etc.) download the `prod_test_69x.bin` to the devices. |
| 16 | Open the devices COM ports and get the `prod_test_69x.bin` firmware version | Yes | After `prod_test_69x.bin` has been downloaded, commands can be sent to it. First, the Windows DUTs COM ports are opened. Then, a command to get the `prod_test_69x.bin` firmware version is sent to the devices. If there is a problem in the firmware or in the device, then this is the first failure to happen. The FW version get action will fail. |
| 17 | GPIO watchdog | Yes | If the GPIO watchdog option is enabled, the firmware will begin a periodic GPIO toggling during the whole production test procedure. |
| 18 | OTP timestamp | Yes | PLT will read the IC production timestamp and log it. |

 © 2020 Dialog Semiconductor

| Step | Action | Opt | Description |
|---|---|---|---|
| 19 | VBAT level measure | Yes | PLT will send a command to each DUT to measure VBAT for each one, using the internal ADC. VBAT level will be logged for debugging purposes. |
| 20 | XTAL trim | Yes | Perform the XTAL trim procedure, if this is active. |
| 21 | UART resync | Yes | If the XTAL trim procedure was performed in the UART RX pin, then a special UART resync procedure takes place to resynchronize the device's UART RX path as it may have entered in a baud rate error state due to the 500ms received XTAL trim pulse. |
| 22 | BLE scan HCI | Yes | If the Scan DUT Advertise test is active, then perform a BLE scan test using HCI triggered advertisements. |
| 23 | BLE tester TX power | Yes | If the BLE tester TX Power test is active, then perform the test using the external BLE tester instrument. |
| 24 | BLE tester TX carrier offset | Yes | If the BLE tester TX carrier offset test is active, then perform the test using the external BLE tester instrument. |
| 25 | BLE tester TX modulation index | Yes | If the BLE tester TX modulation index test is active, then perform the test using the external BLE tester instrument. |
| 26 | BLE tester RSSI | Yes | If the BLE tester RSSI test is active, then perform the test using the external BLE tester instrument. |
| 27 | GU RSSI test | Yes | If the RSSI test using the GU as transmitter is active, then perform the test. |
| 28 | GPIO/LED | Yes | Perform the GPIO/LED test, if the test is active. |
| 29 | GPIO Connection test | Yes | Perform a GPIO continuity or voltage level test, if the test is active. |
| 30 | Sensor test | Yes | Perform the sensor tests only if these are enabled. |
| 31 | Custom test | Yes | Perform any active custom test. |
| 32 | External 32kHz | Yes | Check whether the external 32kHz crystal operates correctly. |
| 33 | Current measure peripheral | Yes | Perform any active current measurement test for peripherals. |
| 34 | Current measure sleep | Yes | Perform the sleep current measurement. |
| 35 | Open COM port and download `uartboot_69x.bin` | Yes | If any of the memory actions is active (e.g. QSPI burn, QSPI erase, etc.) download the `uartboot_69x.bin` to the devices. |
| 36 | Get `uartboot_69x.bin` version. | Yes | After `uartboot_69x.bin` has been downloaded, commands can be sent to it. A command to get the `uartboot_69x.bin` firmware version is sent to the devices. |
| 37 | GPIO watchdog | Yes | If the GPIO watchdog option is enabled, the firmware will begin a periodic GPIO toggling during the whole memory programming procedure. |
| 38 | QSPI memory initialization | Yes | If any QSPI operation is enabled, initialize memory. |
| 39 | QSPI erase | Yes | Erase the QSPI, either the entire or part of it depending on the configuration. |
| 40 | QSPI check empty | Yes | Depending on the configuration, check whether the QSPI is empty to verify the QSPI erase procedure. |
| 41 | QSPI image write | Yes | If enabled, write the QSPI with the customer image. If verify is enabled, the contents of the QSPI will be read back and compared to the original image downloaded. |
| 42 | QSPI BD address write | Yes | If enabled, the device BD address is programmed to a specific QSPI flash address. |

 © 2020 Dialog Semiconductor

| Step | Action | Opt | Description |
|---|---|---|---|
| 43 | QSPI BD address read | Yes | If enabled, the PLT will read the BD address field from the QSPI. This will be printed in the GUI, CLI screen and in the device logs. An additional test can be enabled to compare the read BD address to the one supplied by the tool. |
| 44 | Custom memory data | Yes | Write custom memory data, taken from a barcode scanner, entered manually or through a CVS file. |
| 45 | OTP image write | Yes | Write the OTP image with the customer image. If verify is enabled, the contents of the OTP memory will be read back and compared to the original image downloaded. |
| 46 | OTP CS write | Yes | If enabled, PLT will program the OTP configuration script area |
| 47 | Memory read | Yes | Up to 10 memory read tests can be performed with up to 256 bytes in length. |
| 48 | Scan test | Yes | If enabled, the GU will scan for device BLE advertisements. For the DUTs to be scanned a valid firmware must be burned into the OTP or QSPI flash that sends BLE advertisements after power up. Additionally, the BD address should be burned into the OTP or the QSPI by the PLT.<br>The PLT expects to find devices in the air with the BD addresses programmed by the same tool, so it can match the BD addresses returned by the GU. |

 © 2020 Dialog Semiconductor

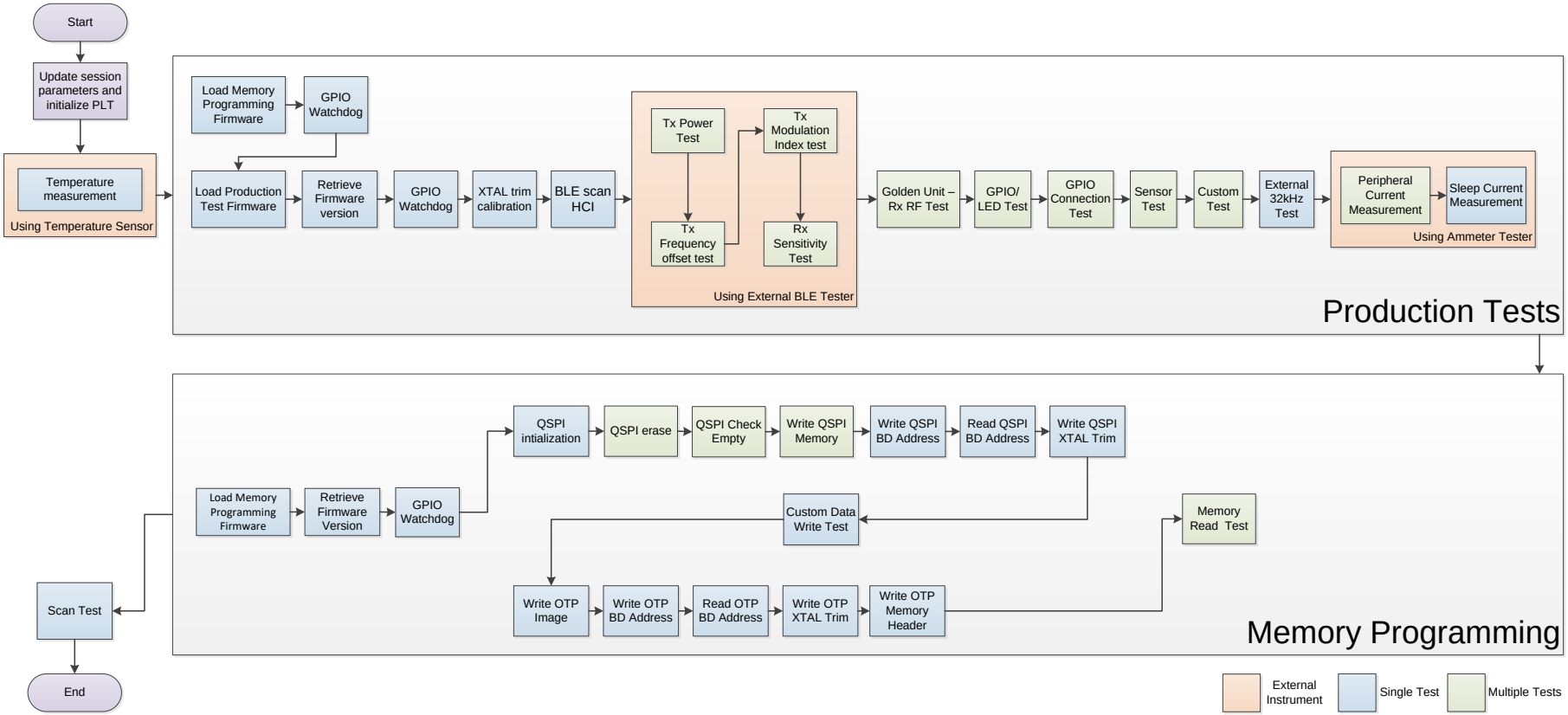



**Figure 24: DA1469x Test Sequence**


© 2020 Dialog Semiconductor

## 6.9 VBAT/Reset Signals Operation

The following chapter describes the PLT hardware VBAT and Reset signal operation during the DUT Test Sequence.

There are two different modes available to power and reset the DUTs using a combination of the PLT VBAT and Reset lines. These are described next.

### 6.9.1 VBAT Only

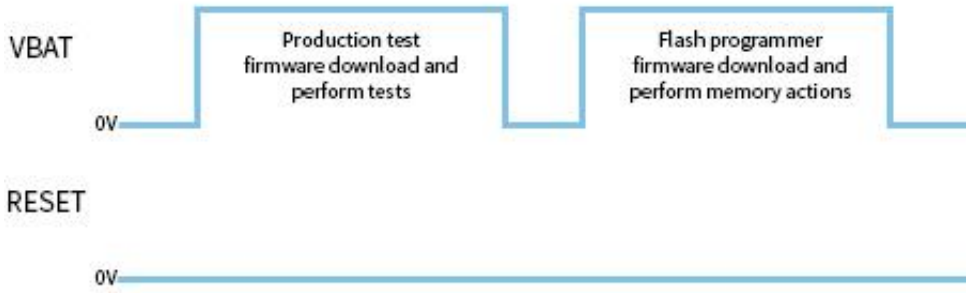



**Figure 25: VBAT only**

In this mode only the VBAT line is used, as shown in Figure 25. Only the VBAT signal from the PLT hardware board to the DUT should be connected. The Reset signal is not driven. The DUTs are powered independently from their VBAT lines connected to the PLT HW and when reset is needed, the PLT software toggles the VBAT line low in order to perform a POR to each device.

Battery powered DUTs or DUTs with an external power supply are not supported in this mode. PLT to DUT VBAT line connection is mandatory. PLT Reset line connection is not required.

#### Firmware download

When the firmware download procedure begins, the PLT VBAT line will power the DUTs and the UART connections will open. This will result to a POR for all active devices. The POR will activate the DUTs UART booting procedure and the PLT software will be able to download the test firmware.

If there are devices that failed the test firmware download procedure, PLT will perform a VBAT POR to retry the firmware download procedure only for those that failed. During the extra attempts to download firmware to the failed devices, the VBAT lines of the devices that succeeded will remain active. After maximum of three retry attempts, the PLT VBAT lines will remain active only for the devices that have succeeded. The retry operation and the amount of retries can be configured by the user. Check 7.2.3.2 for more details.

When the production testing has finished the above procedure will be repeated for the memory programming, as a different firmware needs to be downloaded to the DUTs.

#### Current measurement

Since the DUTs will be powered through the PLT HW using the VBAT line, the Current Measurement Test for DA14531 and the

Current Measurement Test for the DA1469x are supported as described in the Current Measurements chapter.

 © 2020 Dialog Semiconductor

### 6.9.2 VBAT On with Reset

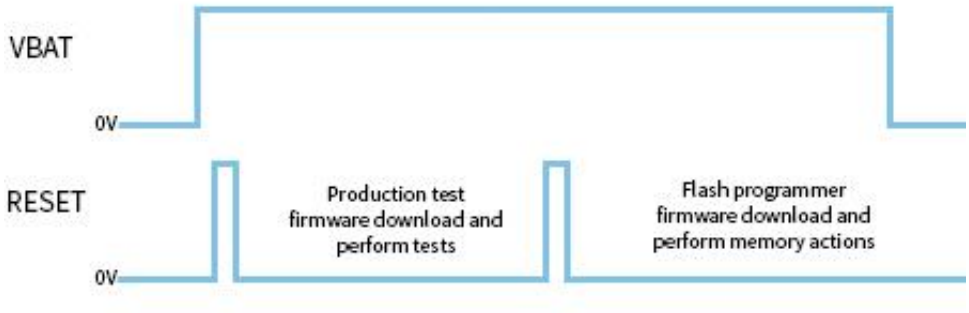



**Figure 26: VBAT On with Reset**

In this mode the reset of the DUTs will be performed by the PLT Reset line as shown in Figure 26. During this mode, the PLT VBAT line continuously provides power to the DUTs and the DUTs are reset using the PLT Reset line.

Power supply can be provided to the DUTs if the PLT VBAT line is connected to the DUTs. However, for battery powered DUTs or for DUTs with an external power supply VBAT should not be connected. For such devices, only the connection to the PLT Reset line is mandatory.

#### Firmware download

When the firmware download procedure begins, PLT will reset the DUTs using the PLT Reset line. The VBAT line is already active and remains active for the entire PLT test and memory programming procedure. If there are devices that failed to download firmware, the PLT will reset all the DUTs again and retry to download firmware to all of them even if these have succeeded. This is different approach from the VBAT Only procedure, since the Reset line is a single hardware line that cannot be differently controlled for each DUT, as opposed to the VBAT lines. The retry operation and the amount of retries can be configured by the user. Check 7.2.3.2 for more details.

When the production testing has finished, the above procedure will be repeated for the memory programming, as a different firmware needs to be downloaded to the DUTs.

#### Current measurement

If the DUTs are powered through the PLT HW using the VBAT line, or if they are powered using a single common line from an external power supply, the Current Measurement Test for DA14531 and the

Current Measurement Test for the DA1469x are supported as described in the Current Measurements chapter. If the DUTs are powered independently or have their own power supply (e.g. battery) then the current measurement tests are not supported.

## 6.10 Custom Memory Data

The following chapter describes the PLT '`Custom Memory Data`' configuration and programming procedure.

The PLT supports programming custom user data of any size up to 256 bytes, to any memory and from any start address. Custom data can be entered to the PLT by the three input methods described in Table 16.

 © 2020 Dialog Semiconductor

**Table 16: Custom Memory Data Input Modes**

| Input Modes | Description |
|---|---|
| Barcode scanner | Prior of starting the PLT tests, before pressing the START button in the PLT GUI, users can use a barcode scanner to enter custom memory data, different for each DUT. A new GUI screen is used to scan DUT barcodes and save the barcode scanned data to the PLT. The PLT will then burn these data to the user specified memory and address. Duplicate scan data protection can be enabled to protect scanning same data for different DUTs in the same test. |
| CSV file | Users can provide a path to a CSV file that will contain the custom memory data for each DUT. The format of the CSV file is specific and is provided in Custom data CSV file format. |
| Manual | Users can manually edit the custom memory data prior of each PLT test run. The edit can be done in the PLT GUI or in the params.xml file using an external application or script. If different data per DUT is required, then the update of the custom memory data should be done before every PLT test run. |

Chapter Custom Memory Data explains in detail the various configuration parameters of the '`Custom Memory Data`' programming PLT feature.

### 6.10.1 Homekit Hash Setup Code

The PLT supports the hashing and programming of homekit setup codes. This feature is only supported for DA1469x DUTs and can be enabled in the Custom Memory Data test using the `Homekit binary generator` option. For this to work users should input to the PLT a specific format of the setup code and device serial number. The format should be as described in Table 17.

**Table 17: Homekit Setup Code Format**

| Homekit Setup Code format | Description | | |
|---|---|---|---|
| XXXXXXXXYYYYAAZZZZZZZZZZZZC | Part | Length | Description |
| | XXXXXXXX | 8 digits | Homekit setup code without the dashes. |
| | YYYY | 4 characters | Setup ID |
| | AA | 1 byte | Accessory category |
| | ZZZZZZZZZZZZ | 12 characters | Serial number |
| | C | 1 character | Checksum of the previous characters |
| | Example:<br>Consider a device with setup code `50867478`, setup id `7OSX`, accessory category `03` and serial number `112233445566`. The input to the PLT should be given as `508674787OSX03112233445566R`.<br>The final character '`R`' is the checksum of the previous characters. The algorithm of the checksum is given in Table 18. | | |

The homekit setup code characters can be taken from either input mode described in Table 16. The PLT will read the data and will apply it as input argument to the `SetupCode_Generator_680.exe` application found under the `binaries` folder. The `SetupCode_Generator_680.exe` application will create, in the same `binaries` folder, a binary image with the name `XXXXXXXXYYYYAAZZZZZZZZZZZZ.bin`. The PLT will burn this file to the DUT memory and start address that the user has configured.

 © 2020 Dialog Semiconductor

**Table 18: Homekit Setup Code Checksum Algorithm**

| Homekit Setup Code Checksum Algorithm |
|---|

```
const char checkCharList[] = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
/**
 *
 * @brief Homekit specific setup code and serial number checksum check function.
 * This checksum algorithm is similar to EAN, ILN and NVE but with alphanumeric instead of numbers
 *
 * @param[in] *inString                    The string to check its checksum.
 * @param[in] *inLength                    The length of the string.
 *
 * @return The checksum character.
 *
 **/
char hmkt_setupcode_chksm(const char *inString, int inLength)
{
    char check = 0;
    int index = 0;
    unsigned int sum = 0;
    const char *characters = inString;

    for (index=0; index<inLength && characters[index]!=0; index++) {
        if ((index % 2) == 0)
            sum += characters[index];
        else sum += characters[index] * 3;
    }

    sum = sum % 36;

    if (sum != 0)
        sum = 36 - sum;

    check = checkCharList[sum];

    return check;
}
```

 © 2020 Dialog Semiconductor

### 6.10.2 Custom data CSV file format

This section describes the format of the CSV file used in *CSV file input mode* of the Custom Memory Data test.

| | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 80:EA:CA:80:00:01 | OTP | 0 | 5 | 5566778801 | SPI | 8000 | 10 | 112233445566778899A1 | | | | |
| 2 | 80:EA:CA:80:00:02 | OTP | 0 | 5 | 5566778802 | SPI | 8000 | 10 | 112233445566778899A2 | | | | |
| 3 | 80:EA:CA:80:00:03 | OTP | 0 | 5 | 5566778803 | SPI | 8000 | 10 | 112233445566778899A3 | | | | |
| 4 | 80:EA:CA:80:00:04 | OTP | 0 | 5 | 5566778804 | SPI | 8000 | 10 | 112233445566778899A4 | | | | |
| 5 | 80:EA:CA:80:00:05 | OTP | 0 | 5 | 5566778805 | SPI | 8000 | 10 | 112233445566778899A5 | SPI | 2 | 9000 | 5501 |
| 6 | 80:EA:CA:80:00:06 | OTP | 0 | 5 | 5566778806 | SPI | 8000 | 10 | 112233445566778899A6 | SPI | 2 | 9000 | 5502 |
| 7 | 80:EA:CA:80:00:07 | OTP | 0 | 5 | 5566778807 | SPI | 8000 | 10 | 112233445566778899A7 | | | | |
| 8 | 80:EA:CA:80:00:08 | OTP | 0 | 5 | 5566778808 | SPI | 8000 | 10 | 112233445566778899A8 | | | | |
| 9 | 80:EA:CA:80:00:09 | OTP | 0 | 5 | 5566778809 | SPI | 8000 | 10 | 112233445566778899A9 | | | | |
| 10 | 80:EA:CA:80:00:0A | OTP | 0 | 5 | 556677880A | SPI | 8000 | 10 | 112233445566778899AA | | | | |
| 11 | 80:EA:CA:80:00:0B | OTP | 0 | 5 | 556677880B | SPI | 8000 | 10 | 112233445566778899AB | | | | |
| 12 | 80:EA:CA:80:00:0C | OTP | 0 | 5 | 556677880C | SPI | 8000 | 10 | 112233445566778899AC | | | | |
| 13 | 80:EA:CA:80:00:0D | OTP | 0 | 5 | 556677880D | SPI | 8000 | 10 | 112233445566778899AD | | | | |
| 14 | 80:EA:CA:80:00:0E | OTP | 0 | 5 | 556677880E | SPI | 8000 | 10 | 112233445566778899AE | | | | |
| 15 | 80:EA:CA:80:00:0F | OTP | 0 | 5 | 556677880F | SPI | 8000 | 10 | 112233445566778899AF | | | | |

**Figure 27: Custom Memory Data CSV File Example**

Each line in the CSV file corresponds to a specific DUT, which is bound to a BD address. The BD address is written in the first column of the CSV file. After the DUT BD address, up to five memory operations can exist.

Each of these operations must have the following columns in the correct order as described below:

- Memory type (DA14531 can have OTP, SPI, EEPROM and DA1469x can have OTP and QSPI),
- Start address
- Size of data in bytes
- Data to be written.

Figure 27 shows an example of a CSV file targeted for DA14531 DUTs. In this particular example the CSV file contains information for DUTs with BD addresses `80:EA:CA:80:00:01` to `80:EA:CA:80:00:13`. For BD addresses `80:EA:CA:80:00:05-06` there are three tests and two for the rest.

- The first operation, which is similar for all BD addresses with only the Data field to be different, is to write in the OTP Header memory of the DA14531 DUTs five bytes, in OTP address 0x0000 (OTP image area).
- The second operation is configured to write into the SPI flash address 0x8000 10 bytes (0x112233445566778899AA1-AF).
- Finally, the third operation only applies for BD addresses `80:EA:CA:80:00:05` and `80:EA:CA:80:00:06`. This will write 2 bytes of data in address 0x9000 of the SPI flash memory.

 © 2020 Dialog Semiconductor

## 6.11 Golden Unit Scan Test

This section describes the PLT scan test procedure using the Golden Unit as scanner device.

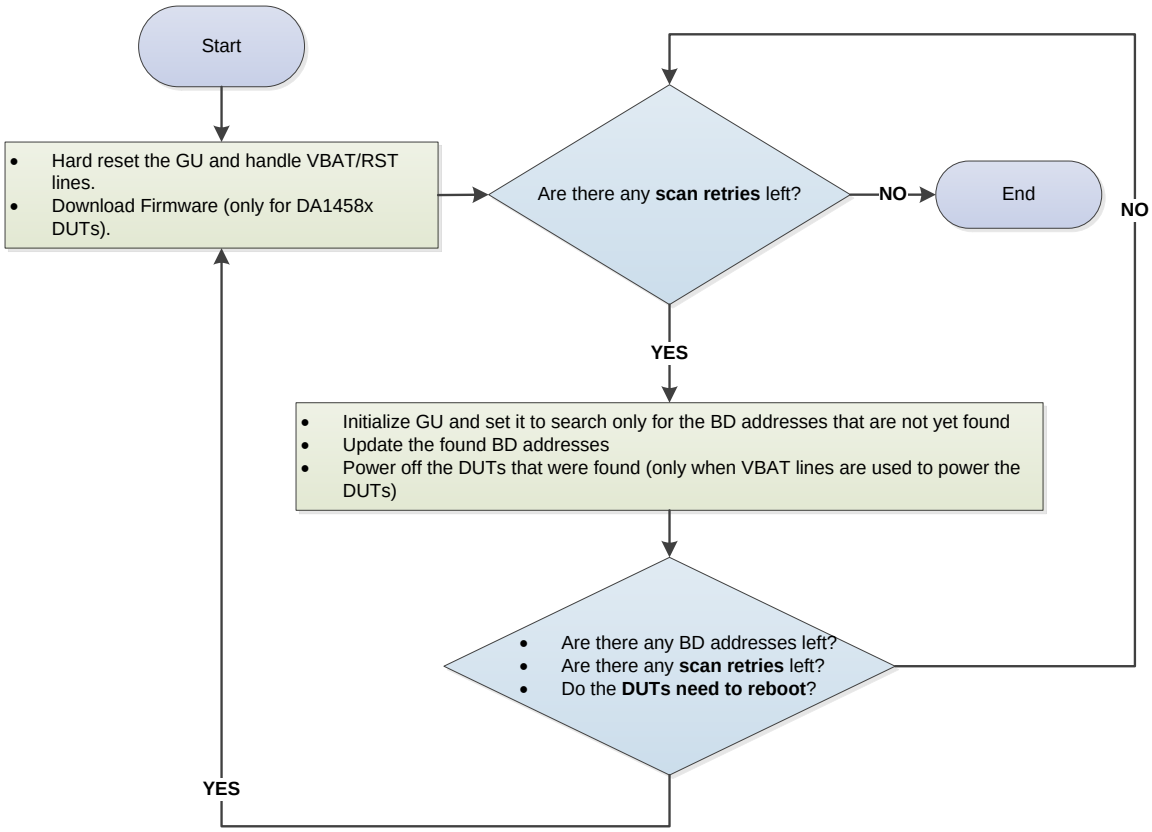



**Figure 28: Golden Unit Scan Test**

User can set various scan properties to adjust the Scan test procedure. The available properties that apply to the DA14531 devices are described Table 60 and for the DA1469x devices in Table 90.

Figure 28 shows the scan sequence. First, the Golden Unit and the DUTs are reset. At this stage, if the *Firmware load enable* is active (option is available only in DA14531 DUTs) the PLT will download the selected firmware. Then, the GU will begin scanning for the BD addresses of all active DUTs. After each scan cycle, the already found BD addresses are removed from the search list of the GU and the appropriate DUTs will be powered off. This procedure will continue until the retries have reached the *Scan retries* set by the user. The PLT will reset the GU after a specific number or retries, given in *DUT reboot* option. Finally, the parameters *DUT reboot time* and *DUT reboot difference* set the DUT time needed to perform a POR with a small delay between the DUTs if needed.

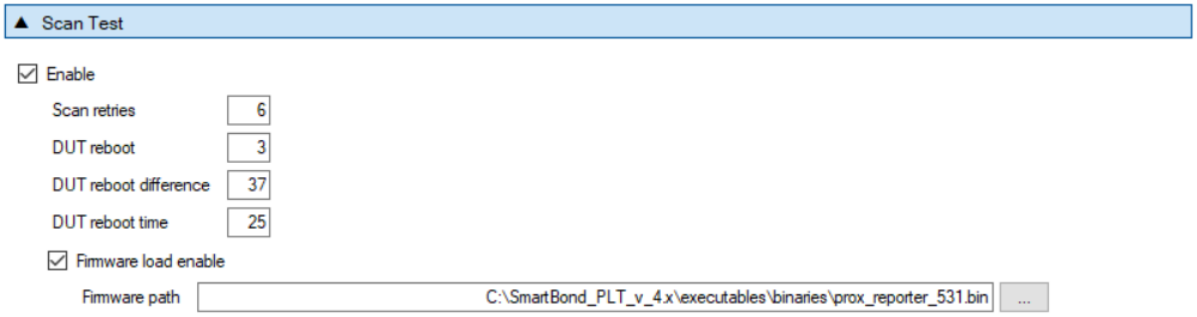



**Figure 29: Golden Unit Scan Test Example Parameters**

Figure 29 shows an example for DA14531 DUT connected with *VBAT only* mode as described in VBAT/Reset Signals Operation. For this example, the following steps will be executed.


© 2020 Dialog Semiconductor

- Reset the GU in order to be in a clear state, power off the DUTs and wait for 2500ms (DUT reboot time). Power on and load `prox_reporter_531.bin` firmware to each DUT with a 37ms time difference between them.
- Execute three GU scan procedures. After each scan procedure is finished, power off the found DUTs.
- Again, reset the GU, power off the DUTs and wait for 2500ms (DUT reboot time). Power on and load prox_reporter_531.bin firmware to each DUT with a 37ms time difference between them.
- Continue with another three GU scan procedures and after each scan procedure power off the found DUTs.

## 6.12 Creating PLT Firmware Files

For the PLT to successfully operate, various firmware files are used based on the device type (GU or DUT), the chipset flavor (DA14531or DA1469x) or the purpose of the firmware (different firmware for production tests and for memory programming).

All these firmware files are kept under the `binaries` folder in the PLT software package, as shown in Figure 30. In order to create these firmware files, the SDK packages should be downloaded from the customer portal and apply the source code patches located under the `fw_files` folder in the PLT software package as shown in Figure 31.

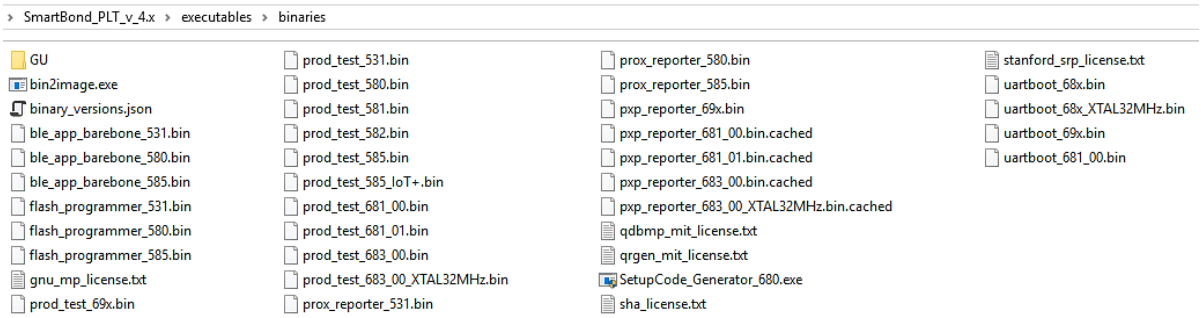



**Figure 30: Binaries**

The source code patches maintain the folder structure of the SDK they are targeting, in order to apply the source code patch using a simple copy and replace to the files needed. After patching, the projects contain all the necessary changes and the same firmware files can be built as those in the `binaries` folder of the PLT software package.

The '`fw_files`' folder has two main categories. Firmware targeted for the GU and for the DUTs. Under each category there is a folder indicating the IC target and the SDK used.

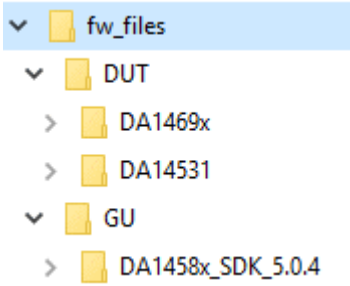



**Figure 31: Folder Contents of 'fw_files'**

Applying a source code patch for each one of the binaries is described below.

### Golden Unit Firmware

The Golden Unit is a DA14580 device. A modified version of the `prod_test_580.bin` firmware is used.


© 2020 Dialog Semiconductor

This patch contains all the changes needed to re-create the following firmware.

- prod_test_GU.bin

In order to re-create the exact source code of the prod_test_GU.bin firmware:

1. Use a clean copy of the DA1458x_SDK_5.0.4 SDK from the customer portal.
2. Copy the contents of the '...\fw_files\GU\DA1458x_SDK_5.0.4\DA1458x_SDK\5.0.4\' folder to the default SDK.
3. The Keil v5 project file of the prod_test_GU.bin is the 'prod_test.uvprojx' under the folder '\5.0.4\projects\target_apps\prod_test\prod_test\Keil_5\'.

### DA14531 Firmware

This patch contains all the changes needed to re-create the following firmware:

- prod_test_531.bin
- flash_programmer_531.bin
- prox_reporter_531.bin
- ble_app_barebone_531.bin

In order to re-create the exact source code of the above firmware:

1. Use a clean copy of the SDK_ 6.0.14.1114 from the Dialog Semiconductor customer portal.
2. Copy the contents of the 'fw_files\DUT\DA14531\6.0.14.1114' folder to the SDK SDK_6.0.14.1114\DA145xx_SDK\6.0.14.1114 folder.
3. The Keil v5 project file of the prod_test_531.bin is the 'prod_test.uvprojx' under 'SDK_6.0.14.1114\DA145xx_SDK\6.0.14.1114\projects\target_apps\prod_test\prod_test\Keil_5' folder.
4. The Keil v5 project file of the flash_programmer_531.bin is the 'programmer.uvprojx' under the 'SDK_6.0.14.1114\DA145xx_SDK\6.0.14.1114\utilities\flash_programmer' folder. Make sure that 'programmer_uart' option is selected under "Select Target".
5. The Keil v5 project file of the prox_reporter_531.bin is the 'prox_reporter.uvprojx' under the 'SDK_6.0.14.1114\DA145xx_SDK\6.0.14.1114\projects\target_apps\ble_examples\prox_reporter\Keil_5' folder.
6. The Keil v5 project file of the ble_app_barebone_531.bin is the 'ble_app_barebone.uvprojx' under the 'SDK_6.0.14.1114\DA145xx_SDK\6.0.14.1114\projects\target_apps\ble_examples\ble_app_barebone\Keil_5' folder.

### DA14681/2/3 Firmware

This patch contains all the changes needed to re-create the following firmware:

- prod_test_69x.bin
- uartboot_69x.bin
- pxp_reporter_69x.bin

In order to re-create the exact source code of the above firmware:

1. Use a clean copy of the SDK_10.0.8.105 from the customer portal.
2. Copy the contents from 'fw_files\DUT\DA1469x\SDK_10.0.8.105' folder to the default SDK.

 © 2020 Dialog Semiconductor

3. The Smart Snippets Studio project file of the `prod_test_69x.bin` is the '`plt_fw`' project under ' `SDK_10.0.8.105\projects\dk_apps\reference_designs\plt_fw`' folder. To create each binary, select from the drop-down menu the "`Release RAM`" option for each chip.
4. The Smart Snippets Studio project file of the `uartboot_69x.bin` is the '`uartboot`' under the folder '`SDK_10.0.8.105\sdk\bsp\system\loaders\uartboot`'. To create the binary, select from the drop-down menu the "`Release`" option.
5. The Smart Snippets Studio project file of the `pxp_reporter_69x.bin` is the '`pxp_reporter`' under the folder '`SDK_10.0.8.105\projects\dk_apps\demos\pxp_reporter`'. To create each binary, select from the drop-down menu the '`QSPI_Release`' option for each chip.

Each binary will be created under the project folder in a folder having the same name as the selected option.

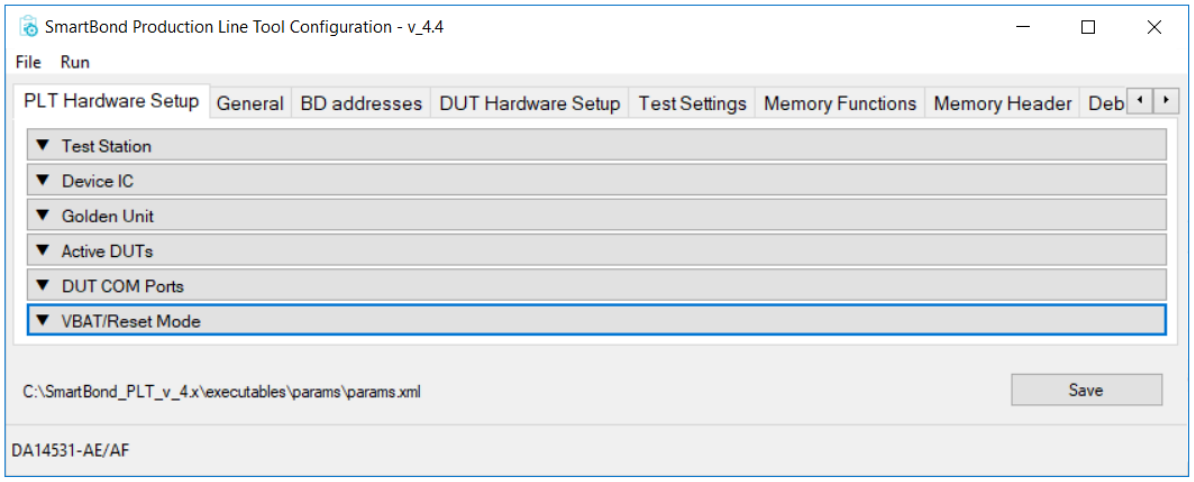
 © 2020 Dialog Semiconductor

# 7 Applications

## 7.1 Introduction

The PLT software includes four different applications (Table 5). The CFG PLT is used to setup the system according to the device hardware options and select the required tests and memory actions to be performed. The GU Upgrade is used to update the Golden Unit firmware. The GUI and the CLI PLT applications are used to perform the tests, monitor their progress in real-time and view the test results.

## 7.2 CFG PLT Application




**Note 1** Each field can be minimized by clicking on it, but it will not be disabled. The tests will run if they are enabled, even when the test field is minimized and not shown.

**Figure 32: CFG PLT Startup Screen**

The CFG PLT application (`SmartBond_CFG_PLT.exe`) is a GUI application tool, which is mainly used to appropriately configure the tests and memory operations the tool will perform. Depending on the selected device chipset and the enabled actions, only appropriate options are enabled and shown. Any change made by the user is validated before being saved to the XML file, with the use of a schema XSD file. This prevents erroneous values to be stored in the XML file that would harm the production procedure.

Figure 32 shows the initial CFG PLT screen. The `Main Menu` options are described in Table 19 and the bottom strip information is described in Table 20. The application begins with the `Hardware Setup` tab (see section 7.2.2). Users can navigate to the other PLT configurable options by selecting the different tabs.

When a tab is selected, the settings of this tab are reloaded from the XML file. If there is an error in the configuration XML file, a warning message will be shown indicating which of the parameters has error. Additionally, the related graphic entry in the CFG application for the erroneous configuration parameter will be highlighted in red.

An example is given in Figure 33. Configuration parameter `dut_num_1` has wrong value (`error` instead of either `false` or `true`) in the `params.xml` file. When the `Hardware Setup` CFG tab is selected the warning message will be displayed. If `OK` is pressed, the `Hardware Setup` tab will be loaded with the `DUT 1` checkbox in red. The displayed value will be the default value taken from the XML schema document (`params.xsd`).


© 2020 Dialog Semiconductor

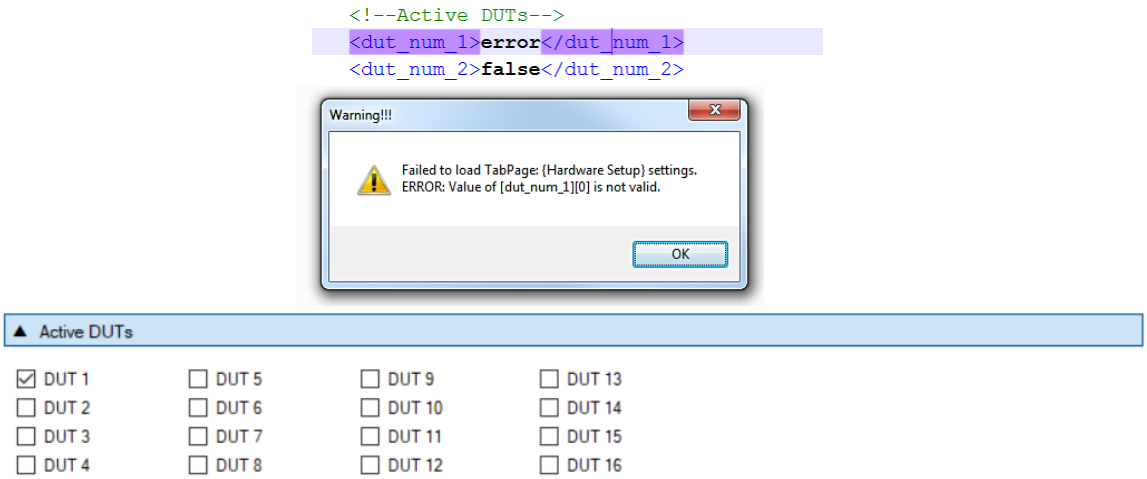



**Figure 33: CFG PLT with Erroneous Configuration Parameter**

When the user makes a change, the `Save` button will become `Save*` to indicate that a save is required.

In case of a configuration parameter error, pressing `Save` will save the default parameter value, overwriting the erroneous value.

**Table 19: CFG PLT Main Menu Options**

| Region | Option | Description |
|---|---|---|
| File | Open XML file | Opens a new XML file and loads the settings. The full path of the new XML file is shown at the bottom end of the screen. |
| | View XML file | Opens the XML file in notepad. |
| | Save as… | Exports all settings to a new XML file. The full path of the new XML file is shown at the bottom end of the screen. |
| | Reset to defaults | Overwrites all parameters options in the XML file with their default values taken from the XSD file. |
| | Exit | Exits the CFG PLT application. |
| Run | Run GUI PLT | Opens the GUI PLT application. |
| | Run CLI PLT | Opens the CLI PLT application. |

**Table 20: CFG PLT Bottom Strip Options**

| Option | Description |
|---|---|
| | |
| C:\SmartBond_PLT_v_4.x\executables\ params\params.xml | Shows the full path of the XML file currently used. |
| DA14531-AE/AF | Shows the selected device IC. |
| Save | Saves the options of the selected tab. E.g. If General settings tab is selected, then only the settings for this tab will be saved.<br>**Note:** A shortcut for this button is the `Ctrl+S` key combination. |


© 2020 Dialog Semiconductor

### 7.2.1 XML and XSD Files

The CFG PLT application is a front-end user interface for the `cfg_dll.dll` library (Figure 21). The `cfg_dll.dll` library, explained in detail in [1], is an XML parser, editor and parameter validator. It has an easy-to-use API for reading and manipulating the `params.xml` file. File `params.xsd` is the XML schema used for parameter validation.

In the CFG PLT application, all user selectable options are loaded and saved inside the XML file, by effectively using the `cfg_dll.dll` library API. The XSD schema file `params.xsd` is not edited in any way but only read by the `cfg_dll.dll` library API, whenever a parameter validation is needed.

The `params.xml` file is separated into three main parts as explained in Table 21.

**Table 21: XML File Parts**

| Part Name | Example | Description |
|---|---|---|
| Common part | `<programming_enable>true</programming_enable>`<br>`<tests_enable>true</tests_enable>`<br>`<retest_failed>false</retest_failed>`<br>`<!--  -->`<br>`<!--  BD Addresses -->`<br>`<!--  -->`<br>`<bd_addr_mode>0</bd_addr_mode>`<br>`<bd_addr_file_path>params\\bd_address.ini</bd_addr_file_path>`<br>`<start_bd_addr>00:00:00:00:00:01</start_bd_addr>`<br>`<next_bd_addr>00:00:00:00:00:01</next_bd_addr>`<br>`<end_bd_addr>00:00:00:00:00:01</end_bd_addr>`<br>`<scanner_iface>COM21</scanner_iface>`<br>`<scan_mode_auto>1</scan_mode_auto>` | The main top part of the XML file contains parameters common to any DUT, like the BD address mode, the COM ports and which device is enabled or disabled. It also holds the debug parameters, the test statistics and the test station name used in the logs. |
| DA14531 | `<!-- DA1458x -->`<br>`<config_params_da1458x>`<br>`<!--  -->`<br>`<!--  UART  -->`<br>`<!--  -->`<br>`<uart_boot_pins>4</uart_boot_pins>`<br>`<uart_change_pins>false</uart_change_pins>`<br>`<uart_pin_tx>P0_4</uart_pin_tx>`<br>`<uart_pin_rx>P0_5</uart_pin_rx>`<br>`<uart_baud_rate>1000000</uart_baud_rate>` | The second XML part, with the element name `config_params_da1458x`, holds parameters used for DA14531 devices. Under this part, the entire test and memory action settings are stored. |
| DA1469x | `<!--  DA1468x  -->`<br>`<config_params_da1468x>`<br>`<!--  -->`<br>`<!--  UART  -->`<br>`<!--  -->`<br>`<uart_boot_pins>4</uart_boot_pins>`<br>`<uart_baud_rate>1000000</uart_baud_rate>`<br>`<!--  -->` | The third and final XML part, with the element name `config_params_da1468x`, holds parameters used for DA1469x devices. Under this part, the entire test and memory action settings are stored. |

The XSD schema file, `params.xsd`, holds information about the overall structure of the `params.xml` file, the default and valid values a parameter can take and useful help information about the purpose of each parameter. An example part of the XSD file is given in Figure 34.

 © 2020 Dialog Semiconductor

```
<xs:element name="next_bd_addr"
            type="x:cfg_hex_array_6_bytes"
            x:use="required"
            x:default="00:00:00:00:00:01"
            x:info="The BD address of the first active DUT that will be used in the next test run. "/>
<!--cfg_hex_array_6_bytes-->
<xs:simpleType name="cfg_hex_array_6_bytes">
    <xs:restriction base="xs:string">
        <xs:pattern value="([0-9A-Fa-f]|[0-9A-Fa-f][0-9A-Fa-f])((:([0-9A-Fa-f]|[0-9A-Fa-f][0-9A-Fa-f])){5})"/>
    </xs:restriction>
</xs:simpleType>

<xs:element name="RF_path_loss_DUT_1"
            type="x:cfg_dut_path_losses"
            x:use="required"
            x:default="0"
            x:info="Set the RF path losses in dB between the device and the GU or the BLE tester instrument."/>
<!--cfg_dut_path_losses-->
<xs:simpleType name="cfg_dut_path_losses">
    <xs:restriction base="xs:float">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="40"/>
    </xs:restriction>
</xs:simpleType>
```

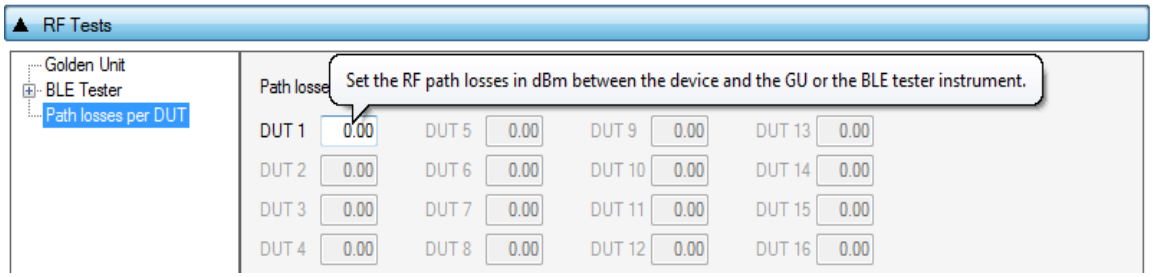



**Figure 34: XSD Schema File Example**

Element `next_bd_addr` holds the Next BD address, as described in section 7.2.4.1 and Table 30. It has a default value of `x:default="00:00:00:00:00:01"`. This default value will be returned by the `cfg_dll.dll` API if the XML file has an error entry in the equivalent `next_bd_addr` element, since the validation of the parameter will fail.

The `x:info="The BD address …"` value will be loaded by the `cfg_dll.dll` API and be used in the CFG PLT tooltips. The `type="x:cfg:hex_array_6_bytes"` defines the parameter type. This is the actual XSD entry that is used for the parameter validation. The `cfg:hex_array_6_bytes` type is defined later in the file and has a rather complicated pattern defined with `<xs:pattern value ="([0-9A-Fa-f]…"/>`. If the `next_bd_addr` element in the XML file has a value that does not match this pattern, the validation of the parameter will fail and the `cfg_dll.dll` API will return the default value (00:00:00:00:01). In the CFG PLT, the default value will be shown in red, indicating that an error exists in the `params.xml` file for this parameter. It will not change the erroneous value in the `params.xml` file until the user presses the **Save** button, in which case the default value will overwrite the erroneous value.

In the second example of Figure 34, the `RF_path_loss_DUT_1` XSD element is shown. This element is used in the Path Losses per DUT as shown in Figure 62. This element has a default value of 0 and the allowed values are floats, between `<xs:minInclusive value="0"/>` and `<xs:maxInclusive value="40"/>`, as shown in the `cfg_dut_path_losses` type description. The `x:info="Set the RF path .."/>` will be loaded by the `cfg_dll.dll` API and used in the CFG PLT tooltips as shown in the bottom part of Figure 34.


© 2020 Dialog Semiconductor

### 7.2.2 Hardware Setup

This section describes the Hardware Setup settings available for the PLT hardware board, as shown in Figure 32.

#### 7.2.2.1 Test Station

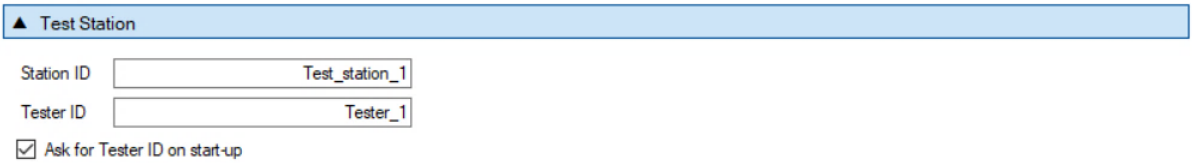



**Figure 35: Station Identification**

These fields hold the station ID and tester ID names to distinguish between different test stations and users. The values of these fields are written into the DUT logs and CSV files. Table 22 describes the available options for the *Station Identification*.

**Table 22: Station Identification**

| Option | Description |
|---|---|
| Station ID | The name of the PLT test station. |
| Tester ID | The PLT tester ID name. |
| Ask for Tester ID on start-up | When `SmartBond_PLT_GUI.exe` starts it will ask for the tester ID name. |

#### 7.2.2.2 Device IC

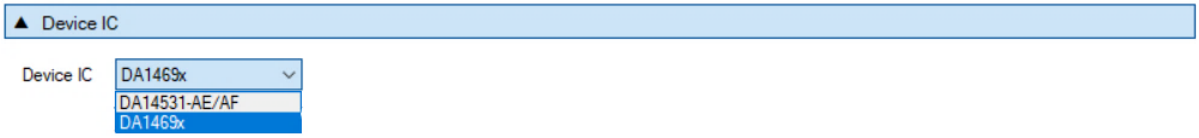



**Figure 36: Device IC**

Users can select the device IC type. This option will also change any IC related graphics, such as selectable tabs and tests. Table 23 describes the available options for the *Device IC*.

**Table 23: Device IC**

| Option | Description |
|---|---|
| Device IC | The Dialog BLE chipset used in the device under test. |

#### 7.2.2.3 Active DUTs

**Figure 37: Active DUTs**

Enables or disables the testing for each DUT. Table 24 describes the available options for the *Active DUT*.

 © 2020 Dialog Semiconductor

**Table 24: Active DUTs**

| Option | Description |
|---|---|
| DUT1-16 | Enables the specific DUT device placed on connector DUT1-DUT16. |

### 7.2.2.4 DUT COM Ports

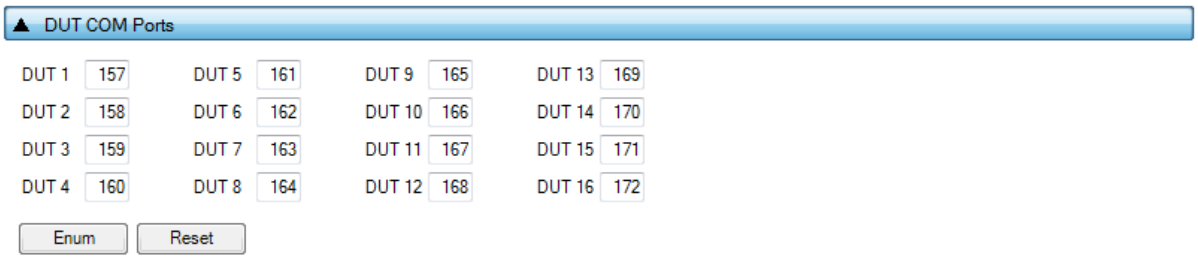



**Figure 38: DUT COM Ports**

This field shows the Windows COM port assigned to each DUT. The table is filled only when the '`COM Enum`' action has been performed by the CFG or the GUI PLT applications, or when non-zero entries exist in the `com_port_x` params.xml options. When the '`COM Enum`' action is performed, the tools will automatically find the DUT COM ports and save them in the `params.xml` file. These values will be read by the CFG PLT application and be displayed here. When a '`COM Enum`' action has not been performed, the GUI PLT will automatically run it once in every first test execution.

**Note:** Great care must be taken when the `params.xml` file is shared across different stations, where different DUT COM Ports will probably exist. The '`COM Enum`' action should then be performed again, so the new COM ports of the new PC system are identified and updated in the XML file.

Table 25 describes the available options for the *DUT COM Ports*.

**Table 25: DUT COM Ports**

| Option | Description |
|---|---|
| DUT1-16 | Shows the Widows COM port assigned to a specific DUT. |
| Reset | Sets all values to zero. |
| Enum | Executes the COM port enumeration procedure. The found COM ports are shown before being saved. |

### 7.2.2.5 Golden Unit

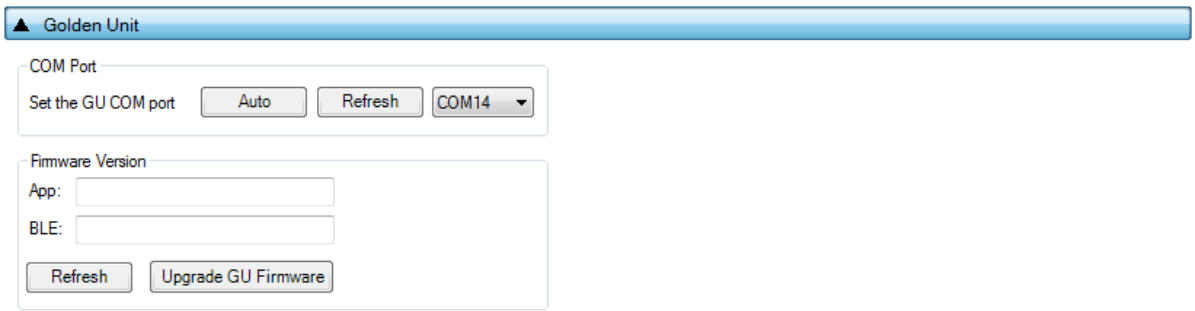



**Figure 39: Golden Unit COM Port**

This field holds the Golden Unit COM port. Manual or automatic COM port find can be selected.

The Golden Unit COM port can be manually selected from the list with all the available COM ports existing in the system. Additionally, it can automatically be found by pressing the `Auto` button. The

 © 2020 Dialog Semiconductor

automatic procedure searches the serial number of all system COM ports to find the "`DialogSemi`" string. Details on how to program the serial number in the GU FTDI can be found in Appendix I.

**Table 26: Set the GU COM Port**

| Option | Description |
|---|---|
| Auto | Initiates the automatic Golden Unit COM port find procedure. |
| Refresh | Refreshes the dropdown menu with all the available system COM ports. |
| Dropdown Menu | Manually select the Golden Unit COM port from all the available system COM ports. |

**Table 27: Golden Unit Firmware Version Upgrade**

| Option | Description |
|---|---|
| Refresh | Retrieves the current BLE and application versions of the connected Golden Unit. |
| Upgrade GU Firmware | Opens the GU Upgrade application, which is used to update the GU firmware. |

### 7.2.2.6 VBAT/Reset Mode

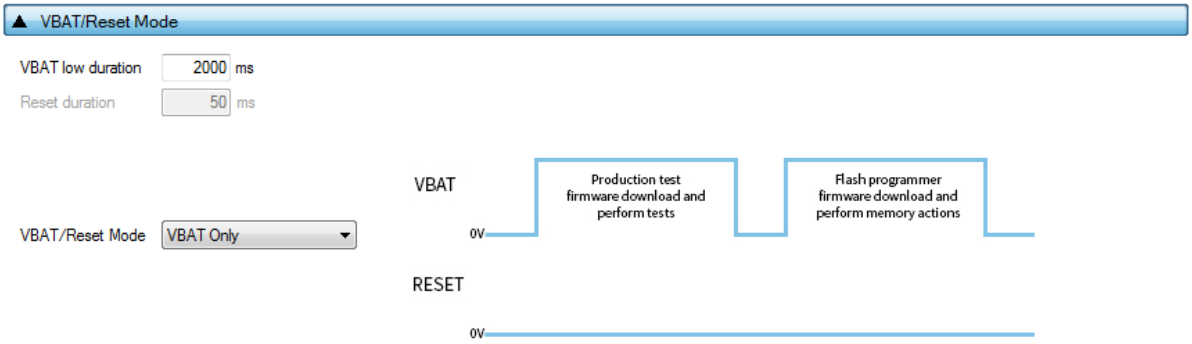



**Figure 40: VBAT/Reset Mode Selection**

This field holds the VBAT/Reset mode selections. This option sets the PLT VBAT and PLT Reset line modes for the DUT power supply and reset during the PLT test sequence. Table 28 describes the available selections.

**Table 28: VBAT/Reset Mode**

| Option | Description |
|---|---|
| VBAT/Reset Mode | Select the operation for VBAT/Reset signals. Available options are:<br>● VBAT Only<br>● VBAT On with Reset<br>VBAT/Reset Signals Operation chapter describes each mode in detail. Default setting is *VBAT only*. |


© 2020 Dialog Semiconductor

### 7.2.3 General

#### 7.2.3.1 Statistics

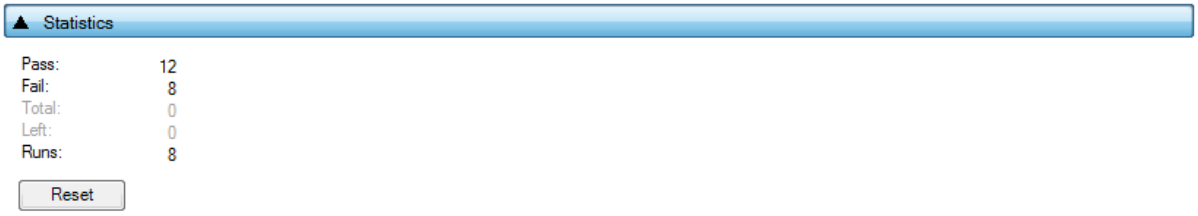



**Figure 41: Statistics**

This field holds the test result statistics. Table 29 describes the *Statistics* field.

**Table 29: Statistics**

| Option | Description |
|---|---|
| Pass | Shows the number of DUTs that have successfully passed all the tests. |
| Fail | Shows the number of DUTs that have failed the tests. |
| Total | Shows the number of DUTs that will be tested. This option is available only when *Range* mode is enabled in the BD Address Assignment. |
| Left | Shows how many DUTs are still to be tested. This option is available only when *Range* mode is enabled in the BD Address Assignment. |
| Runs | Shows the number of test runs the PLT has performed. |
| Reset | Pressing the `Reset` button clears all statistics values to their defaults. Values *Pass*, *Fail* and *Runs* will be set to zero. If *Range* mode is enabled in the BD Address Assignment, the *Total* and *Left* values will be set as the difference of *Next* and *End BD address*, otherwise will be set to zero. |

#### 7.2.3.2 Test Options

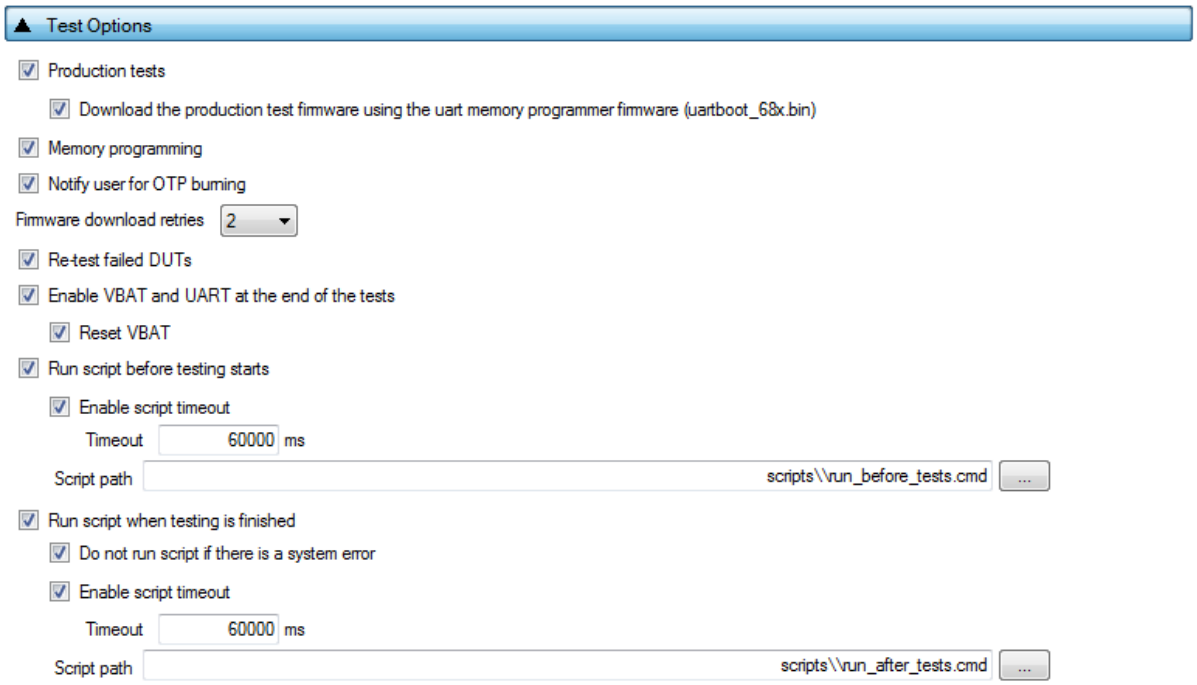



**Figure 42: Test Options**

 © 2020 Dialog Semiconductor

## SmartBond Production Line Tool

This field holds generic PLT test procedure options. The PLT procedure is split into two main parts: *Production tests* and *Memory programming*.

*Production tests* include all the tests under Test Settings (DA14531) or Test Settings (DA1469x). *Memory Programming* includes all the tests under Memory Functions (DA14531) or Memory Functions (DA1469x) and

 © 2020 Dialog Semiconductor

Memory Header (DA14531) or

Memory Header (DA1469x).

describes the available settings for the *Test Options*.

| Option | Description |
|---|---|
| Production tests | This option enables the production test operations. |
| Memory programming | This option enables the memory programming operations. |
| Notify user for OTP burning | When this option is enabled, PLT will inform the user with all the OTP burning tests that are enabled. A pop-up message will appear prompting the user whether to proceed with the tests. |
| Firmware download retries | Configures the firmware download retries in case of an error during firmware download. |
| Re-test failed DUTs | When this option is enabled, any DUT that failed will immediately be retested with the exact same options, including the *BD address*. This option is the same to the *Retest failed DUTs - Enable* under GUI PLT Settings. |
| Enable VBAT and UART at the end of the tests | Enables the VBAT lines and UART communication between the PC and the devices after all the tests have finished. If enabled, DUTs will remain powered after the end of the tests. |
| Reset VBAT | If this option is enabled the VBAT line will be toggled. If not selected, the DUTs will keep in their system RAM the last test firmware downloaded by the PLT. |
| Run script before testing starts | This option enables the execution of a batch or an executable before the device testing procedure starts. As described in Running the GUI PLT and Executing Tests, the success return code should be a value between 0 and 100 for the tool not to report an error. Any other value will be taken as error and prevent the tool from running the tests. |
| Enable script timeout | Enables a wait timeout for the script to finish. The time to wait is set in the `timeout` field below. If this option is disabled PLT will wait until the script ends. |
| Timeout | The time to wait for the script to finish. |
| Script path | The path of file to execute when the *Run script before testing starts* option is enabled. |
| Run script when testing is finished | This option enables the execution of a batch or an executable after the device testing procedure has finished. The success return code should be 0 for the tool not to report an error. |
| Enable script timeout | Enables a wait timeout for the script to finish. The time to wait is set in the `timeout` field below. If this option is disabled, PLT will wait until the script ends. |
| Timeout | The time to wait for the script to finish. |
| Script path | The path of file to execute when the *Run script when testing is finished* option is enabled. |

 © 2020 Dialog Semiconductor

### 7.2.4 BD Addresses

#### 7.2.4.1 BD Address Assignment

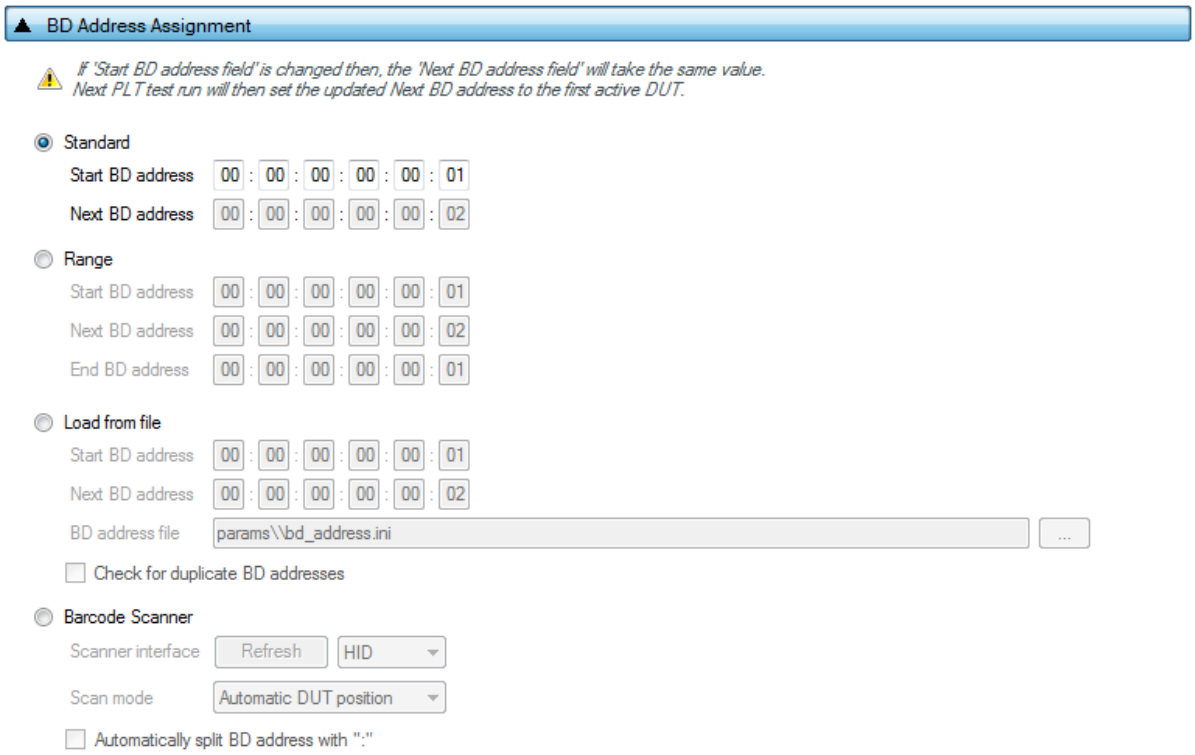



**Figure 43: BD Address Assignment**

The *BD Address Assignment* field defines different ways the PLT can handle the device BD address. The available modes are *Standard*, *Range*, *Load from file* and *Scan* mode.

The *Standard*, *Range*, and *Load from file* modes are similar. These have a *Start BD address,* which is the initial address that the PLT session begins. The *Next BD address* field holds the BD address that will be used on the next PLT run, so the BD address assignment can be continued even after the GUI PLT is closed. For that reason, the user cannot alter the *Next BD address*. The *Next BD address* initial value is the same as the *Start BD address* when the PLT session begins.

For *Scan* mode, an external barcode scanner is needed to assign the device BD addresses.

**Note:** In CFG PLT only the Start BD address is given. The assignment of the actual device BD addresses occurs in the GUI PLT at the beginning of each test run.

**Note:** The only invalid BD address is `00:00:00:00:00:00`.

#### Standard Mode

Table 30 describes the available options for the *Standard* mode. In this mode, the first active DUT takes the *Next BD address*. This BD address is incremented by one and assigned to the next active DUT until all active DUTs have a BD address assigned to them.

This assignment mode never runs out of BD addresses and it will continue assigning addresses until the *Next BD address* reaches `FF:FF:FF:FF:FF:FF`.


© 2020 Dialog Semiconductor

**Table 30: BD Address Assignment - Standard Mode**

| Option | Description |
|---|---|
| Start BD address | The BD address that the PLT session has started with. |
| Next BD address | The BD address that will be used in the first active DUT of the next PLT run. |

### Range Mode

Table 31 describes the available options for the *Range* mode. This mode is the same as *Standard* mode except for the additional *End BD address*.

Sine a 'Start BD address' and an 'End BD address' exist, the total amount of devices to be tested can be calculated. Therefore, this mode enables the *Total* and *Left* fields in the Statistics, where *Total* is the number of the BD addresses to be used from *Start BD address* to *End BD address* and *Left* is the number of BD addresses remaining.

**Note:** The *End BD address* must always be greater than the *Start BD address*. In addition, when *Left* BD addresses are not enough for the remaining active DUTs, the PLT will not run.

**Table 31: BD Address Assignment Options - Range Mode**

| Option | Description |
|---|---|
| Start BD address | The BD address that the PLT session has started with. |
| Next BD address | The BD address that will be used in the first active DUT of the next PLT run. |
| End BD address | The BD address that the PLT session will end with. |

### Load from File Mode

Table 32 describes the available options for the *Load from file* mode. In this mode, the *Start BD address* and the *Next BD address* have the same roles as before. The difference in this mode is that the BD addresses are loaded from a file in the order as they are written in that file, not using the automatic incremental method of the previous modes. In every test run, the PLT will search for the first occurrence of the *Next BD address* in the file and will load it along with the BD addresses that follow, until all active DUTs have a BD address.

```
1  00:00:00:44:33:0a
2  00:00:00:44:33:09
3  00:00:00:44:33:08
4  00:00:00:11:22:08
5  00:00:00:11:22:06
6  00:00:00:11:22:05
7  00:00:00:11:22:04
8  00:00:00:11:22:03
9  00:00:00:11:22:02
```

**Figure 44: Example for Load from File Mode**

For example, consider three active DUTs: DUT3, DUT6, and DUT 9 and the *Next BD address* to be `00:00:00:11:22:08`. Figure 44 shows the beginning of the BD address file used in this example. The PLT will search for the *Next BD address* in the file and load it to the first active DUT: DUT3. It will then continue with `00:00:00:11:22:06` for DUT6 and `00:00:00:11:22:05` for DUT9. It will also return 00:00:00:11:22:04 as the *Next BD address* to be used in the next PLT test run.

**Note:** The BD address file should always end with a zero BD address (`00:00:00:00:00:00`) and a new line at the end.

 © 2020 Dialog Semiconductor

**Table 32: BD Address Assignment Options - Load from File Mode**

| Option | Description |
|---|---|
| Start BD address | The BD address that the PLT session has started with. |
| Next BD address | The BD address from file that will be used in the first active DUT of the next PLT run. |
| BD address file | Path to the file that contains the BD addresses. Use button [...] on the right to navigate and select a file. |
| Check for duplicate BD addresses | Before any BD address is assignment happens, there will be a check to find double BD addresses in the selected BD address file. |

### Scan Mode

Table 33 describes the available options for the *Scan* mode. For this option a USB-to-Serial barcode scanner should be used to scan for BD address barcodes with '`XX:XX:XX:XX:XX:XX`' format.

The barcode scanner options are the same as those used for the barcode scanner mode in Custom Memory Data for the DA1458x devices and in Custom Memory Data for the DA1468x devices.

**Note:** Barcode scanner mode is only available with

 © 2020 Dialog Semiconductor

GUI PLT Application. CLI PLT Application does **NOT** support this feature.

**Table 33: BD Address Assignment Options - Scan Mode**

| Option | Description |
|---|---|
| Scanner Interface | Selection of the Barcode scanner input. Both HID and COM port interfaces are supported. This dropdown list provides an HID and all the available system COM ports as input options.<br>For the HID interface, any HID device is supported that includes newline (CR-LF) characters at the end of the scanned data.<br>For the COM port interface, a common USB to UART barcode scanner is supported. PLT has been tested with Honeywell Xenon 1900. Appendix M describes the setup procedure.<br>This option is the exact same option as for the DA14531 devices in Custom Memory Data and the DA1469x devices in Custom Memory Data**.** |
| Scan mode | ● Scan DUT position: In this mode the users must first scan the DUT position number and then the BD address. The string used for the position of each DUT is "TEST POSITION 0xx" where "xx" is the DUT position number.<br>● Automatic DUT position: Scanned BD address will be assigned to the selected DUT. The DUT selection is automatically been made, starting from the first active DUT and selecting the next one after a successful BD address scan. Users can change the selected DUT using the controls on the GUI PLT screen shown in Figure 112.<br>This option is the exact same option as for the DA14531 devices in Custom Memory Data and the DA1469x devices in Custom Memory Data**.** |
| Automatically split BD address with ':' | When this option is enabled, the input data will be automatically delimited with colon marks. E.g. To enter the '`11:22:33:44:55:66`' BD address the input string should be '`112233445566`'. |

 © 2020 Dialog Semiconductor

### 7.2.5 DUT Hardware Setup (DA14531)

#### 7.2.5.1 UART Boot Pins Setup

**Figure 45: UART Boot Pins Setup - DA14531**

Table 34 describes the available options for the *TX-RX pins* of the *UART Boot Pins Setup* DA1458x options. *The TX-RX pins* selection defines the UART pins and the baud rate that will be used for firmware downloading to the DA1458x during boot.

**Table 34: UART TX-RX Pins - DA1458x**

| Option | Description |
|---|---|
| TX: P0_0, RX: P0_1 | Sets UART TX pin to P0_0, UART RX pin to P0_1 and Baud rate to 115200 bit/s. |
| TX\RX: P0_3 (single wire) | Sets UART TX and RX pins to P0_3. Boot baud rate will be 115200 bit/s. |
| TX\RX: P0_5 (single wire) | Sets UART TX and RX pins to P0_3. Boot baud rate will be 115200 bit/s. |

**Note:** The baud rate is fixed during device boot, since it is controlled by the device ROM bootloader.

#### 7.2.5.2 UART Baud Rate

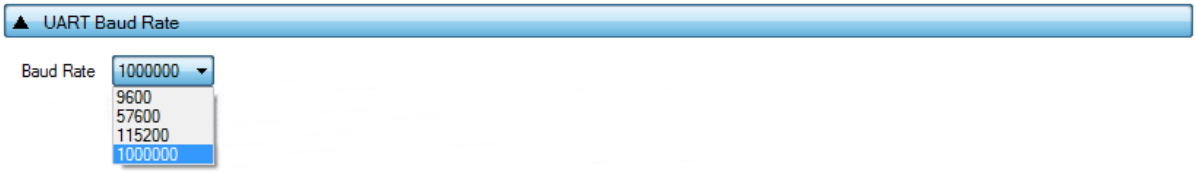



**Figure 46: UART Baud Rate - DA14531**

Table 35 shows the available options for the *UART baud rate.*

The *Baud Rate* selected here is used after the initial firmware (`flash_programmer_531.bin`) has been downloaded to the DUT. The software will send a command to the DUT to change the UART baud rate to the one selected. All following UART communications with the DUT will be performed using the new baud rate. Please note that this happening only during memory programming where the `flash_programmer_531.bin` is used. During tests (`RF tests, XTAL trimming`, etc.), where the production test firmware is used (`prod_test_531.bin`), the baud rate is fixed at 115200 bit/s.

**Table 35: UART Baud Rate - DA1458x**

| Option | Description |
|---|---|
| Baud Rate | • 9600 [bit/s]<br>• 57600 [bit/s]<br>• 115200 [bit/s]<br>• 1000000 [bit/s]<br>**Note:** 1 Mbit/s is the fastest and safest with 0% baud rate error. |

 © 2020 Dialog Semiconductor

### 7.2.5.3 SPI Flash Configuration

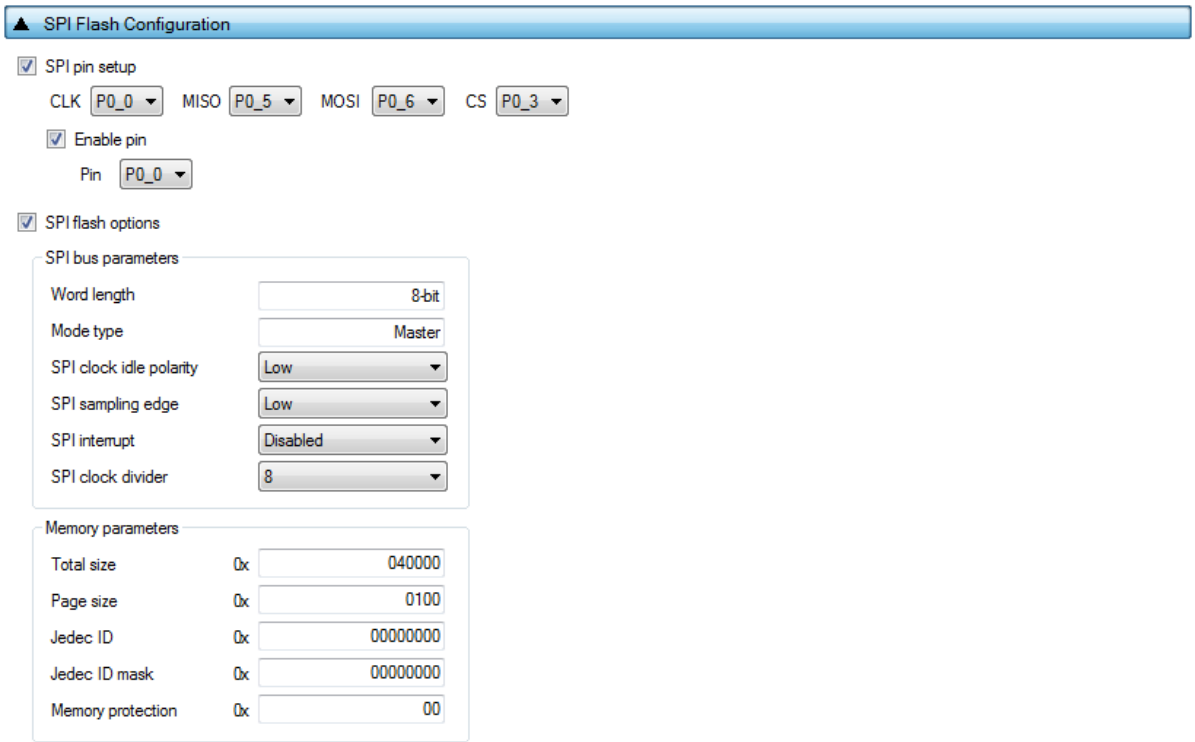



**Figure 47: SPI Flash Configuration - DA14531**

Table 36 describes the available options for the *SPI Pin Setup*.

**Table 36: SPI Pin Setup - DA14531**

| Option | Description |
|---|---|
| SPI pin setup | This option enables the SPI flash memory pin selections. |
| CLK | Sets the GPIO for the CLK pin of the SPI bus. Default GPIO pin is P0_0. |
| MISO | Sets the GPIO for the MISO pin of the SPI bus. Default GPIO pin is P0_5. |
| MOSI | Sets the GPIO for the MOSI pin of the SPI bus. Default GPIO pin is P0_6. |
| CS | Sets the GPIO for the CS pin of the SPI bus. Default GPIO pin is P0_3. |
| Enable pin | Sets a specific GPIO to high state during any SPI flash operation. |
| Pin | Sets the GPIO to be used as the enable pin. |

Table 37 describes the available options for the DA14531 *SPI Flash Configuration*. In order to properly setup the SPI flash configuration, refer to the datasheet of the memory to be used.

**Note:** If the memory to be used is listed in the supported memories (Appendix U), then the SPI pin setup configuration option can be disabled.

**Table 37: SPI Flash Configuration - DA14531**

| Option | Description |
|---|---|
| SPI flash options | Enables the on-demand SPI flash configuration. |
| Word length | Shows the length of each word in the SPI-bus. |
| Mode type | Shows the SPI-bus role of the chip. |

 © 2020 Dialog Semiconductor

| Option | Description |
|---|---|
| SPI clock idle polarity | Sets the level of the idle state of the clock. |
| SPI sampling edge | Sets the SPI-bus sampling edge. |
| SPI interrupt | Enables the SPI interrupt.<br>**Note:** This interrupt may be shared with other interrupts. |
| SPI clock divider | Sets the SPI-bus clock frequency. |
| Total size | Sets the SPI Flash size in bytes. |
| Page size | Sets the size of each page of the SPI Flash memory. |
| Jedec ID | Sets the SPI Flash Jedec ID. |
| Jedec ID mask | Sets the bitmask of the Jedec ID. |
| Memory protection | Sets the SPI Flash protection value. |

#### 7.2.5.4 I2C EEPROM Configuration

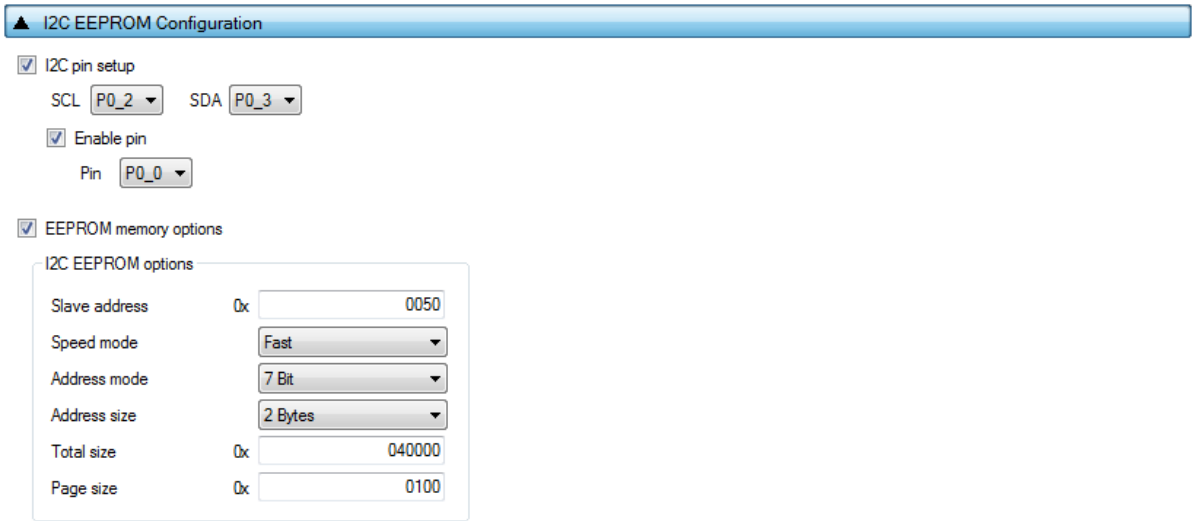



**Figure 48: I2C EEPROM Configuration - DA14531**

Table 38 describes the available options for the *I2C Pin Setup*.

**Table 38: I2C Pin Setup - DA14531**

| Option | Description |
|---|---|
| I2C pin setup | This option enables the I2C pin selections. If this option is disabled, the default pin configuration will be used. |
| SCL | Sets the GPIO for the SCL pin of the I2C bus. Default GPIO pin is P0_2. |
| SDA | Sets the GPIO for the SDA pin of the I2C bus. Default GPIO pin is P0_3. |
| Enable pin | Sets a specific GPIO to high state during any EEPROM operation. |
| Pin | Sets the GPIO to be used as the enable pin. |

Table 39 describes the available options for the DA14531 *I2C EEPROM Configuration*. In order to properly setup the I2C EEPROM configuration, refer to the datasheet of the memory to be used.

**Note:** If the memory to be used is listed in the supported memories (Appendix U), then the on demand I2C EEPROM configuration option is not needed.

 © 2020 Dialog Semiconductor

**Table 39: I2C EEPROM Configuration – EEPROM Memory Options - DA14531**

| Option | Description |
|---|---|
| EEPROM memory options | Enables the on demand EEPROM memory configuration. |
| Slave address | Sets the I2C-bus slave address of the EEPROM memory to be used. |
| Speed mode | Sets the I2C-bus speed. |
| Address mode | Sets the I2C-bus addressing mode. |
| Address size | Sets the I2C-bus number of bytes used for address. |
| Total size | Sets the EEPROM size in bytes. |
| Page size | Sets the size of each page of the EEPROM memory. |

## 7.2.6
## 7.2.6 Test Settings (DA14531)

### 7.2.6.1 GPIO Watchdog Operation

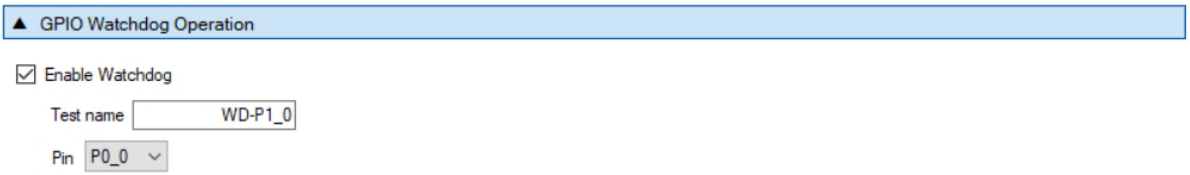



**Figure 49: GPIO Watchdog Operation – DA14531**

Table 40 describes the GPIO watchdog configuration options. When this feature is enabled, firmware will continuously toggle the selected GPIO at specific intervals, less than 4seconds.

**Table 40: GPIO Watchdog Configuration Options – DA14531**

| Option | Description |
|---|---|
| Enable Watchdog | The checkbox enables the operation. It enables the continuous toggling of a GPIO during the whole production testing and memory programming procedure, except during firmware download. The pulse on the GPIO has approximately 1.5% duty cycle and 0.48Hz frequency. |
| Test name | The test name to be used for logging purposes |
| Pin | The GPIO to toggle |

### 7.2.6.2 VBAT Level Log

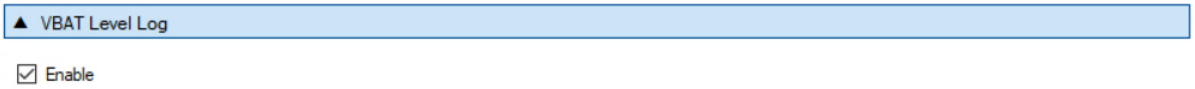



**Figure 50: VBAT Level Log - DA14531**

When this feature is enabled, PLT will send a command to the device to measure VBAT using its internal ADC. The VBAT level will then be logged. No, pass or fail limits exist for this test. It is only used for logging purposes.

### 7.2.6.3 OTP Timestamp Read

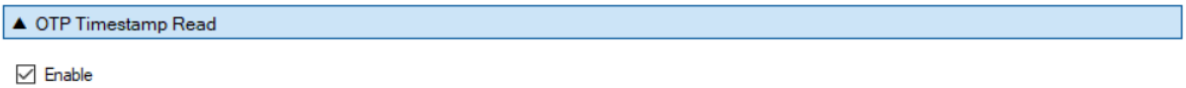



**Figure 51: OTP Timestamp Read - DA14531**


© 2020 Dialog Semiconductor

If this option is enabled, PLT will read the device timestamp from the OTP memory and log it. This operation is mainly used for logging purposes.

#### 7.2.6.4 DC-DC Converter Level Test

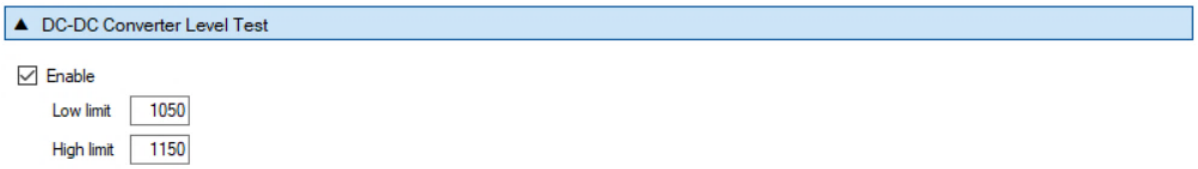



**Figure 52: DC-DC Converter Level Test - DA14531**

When this test is enabled, PLT will send a command to the device to measure the DC-DC converter level. It will then compare it to the limits given by the user. Table 41 describes the user configurable options.

**Table 41: DC-DC Converter Level Test Options - DA14531**

| Option | Description |
|---|---|
| Enable | The checkbox enables the test |
| Low limit | The low limit of the DC-DC converter level test. If result is lower test will fail. |
| High limit | The high limit of the DC-DC converter level test. If the result is higher test will fail. |

#### 7.2.6.5 XTAL Trim

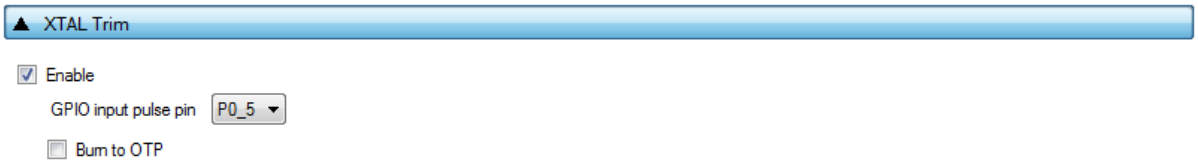



**Figure 53: XTAL Trim - DA1458x**

Table 42 describes the available options for the DA1458x *XTAL Trim* operation.

**Table 42: XTAL Trim - DA1458x**

| Option | Description |
|---|---|
| Enable | This option enables the automatic crystal oscillator frequency calibration procedure. |
| GPIO input pulse pin | The GPIO on which the DUT will receive the reference pulse during calibration. The UART RX pin can be used for this purpose without any additional connection from the PLT hardware to the DUT. |
| Burn to OTP | When this option is selected, the XTAL trim value calculated from the automated calibration process will be written into the OTP XTAL trim header field and the OTP XTAL calibration flag will be set. |

 © 2020 Dialog Semiconductor

#### 7.2.6.6 Scan DUT Advertise Test

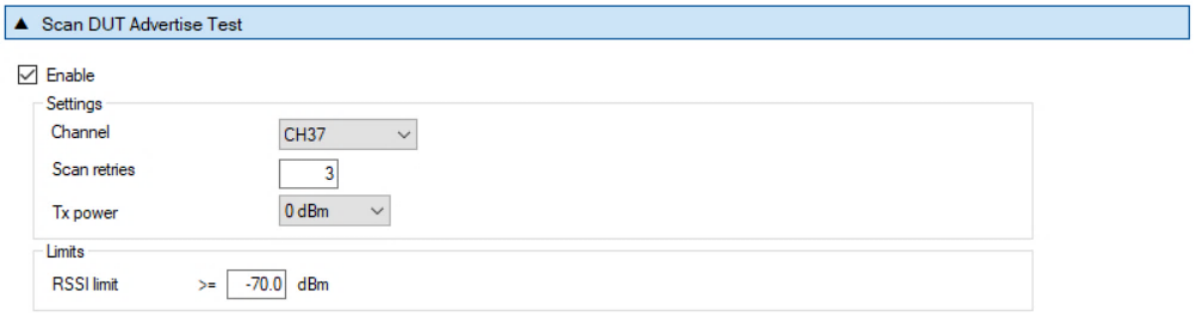



**Figure 54: Scan DUT Advertise Test - DA14531**

Table 43 describes the available options for the DA14531 *Scan DUT Advertise Test* operation. In this test, the Golden Unit acts as a scanner and the DUTs start advertising using HCI commands.

**Table 43: Scan DUT Advertise Test - DA1458x**

| Option | Description |
|---|---|
| Enable | This option enables the `Scan DUT Advertise Test` operation. |
| Channel | The BLE channel frequency used in the RF RX test using the Golden Unit. |
| Scan retries | The number of retries to perform the test. |
| TX power | Set the device output TX power. |
| RSSI limit | The RSSI limit for pass/fail criteria in the RF RX test using the Golden Unit. If the average RSSI of the device after it has received the packets transmitted from the Golden Unit is less than that the test will be considered as failed. |

#### 7.2.6.7 RF Tests

This section refers to various RF tests conducted between the DUTs and the Golden Unit or an external BLE tester.

These tests can have multiple instances with different settings. Tests can be added and removed using the two buttons (e.g. - and + in Figure 55) at the bottom right side of each panel.

**Note:** When adding or removing a test, all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

 © 2020 Dialog Semiconductor

### Golden Unit

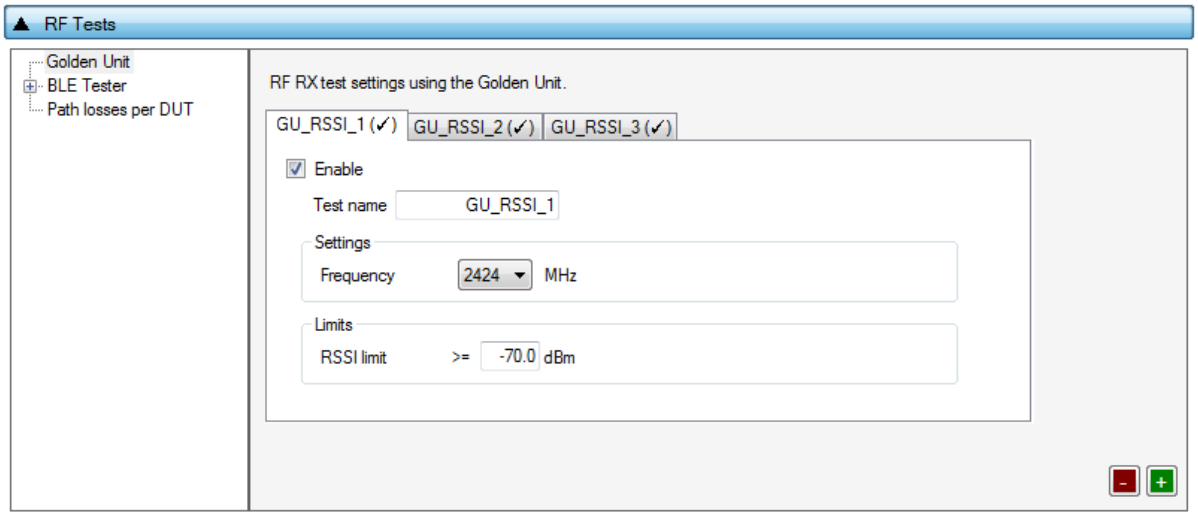



**Figure 55: Golden Unit RF Tests - DA14531**

Table 44 describes the available options for the *RF RX* test using the *Golden Unit* as a transmitter.

In the RF RX test, the Golden Unit sends 500 packets on the selected BLE channel. The DUTs are set in receive mode and the RSSI is measured. If the RSSI measured by the DUT reception is less than the specified *RSSI limit* value, the device will fail and the tests will stop for that particular device.

**Table 44: Golden Unit RF Tests - DA14531**

| Option | Description |
|---|---|
| Enable | This option enables the specific RF RX test using the Golden Unit as a transmitter. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Frequency | The BLE channel frequency used in the RF RX test using the Golden Unit. |
| RSSI limit | The RSSI limit for pass/fail criteria in the RF RX test using the Golden Unit. If the average RSSI of the device, after it has received the packets transmitted from the Golden Unit, is less than the value entered here the test will fail. |


© 2020 Dialog Semiconductor

### BLE Tester

In the *BLE Tester* panel, several tests can be enabled that require an external BLE tester instrument.

### BLE Tester – General Settings

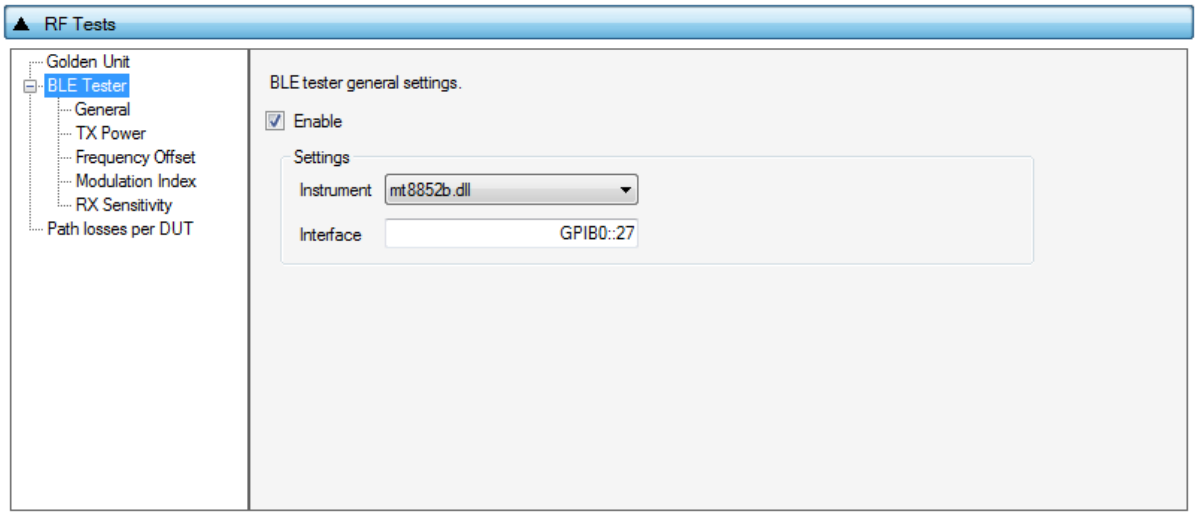



**Figure 56: BLE Tester General Settings - DA1458x**

Table 45 describes the general settings for the *BLE Tester* supported tests. Any available external instrument found by the `ble_tester_driver` DLL and their interfaces can be selected.

**Table 45: BLE Tester General Settings - DA1458x**

| Option | Description |
|---|---|
| Enable | This option enables all the *BLE Tester* tests, which include:<br>● BLE Tester TX Power<br>● Frequency Offset<br>● Modulation Index<br>● RX Sensitivity |
| Instrument | Select the BLE tester DLL name. Names are shown only if a BLE tester instrument DLL exists in the project `ble_tester_instr_plugins` folder. |
| Interface | The interface of the instrument to be used by the driver. |


© 2020 Dialog Semiconductor

### BLE Tester - TX Power

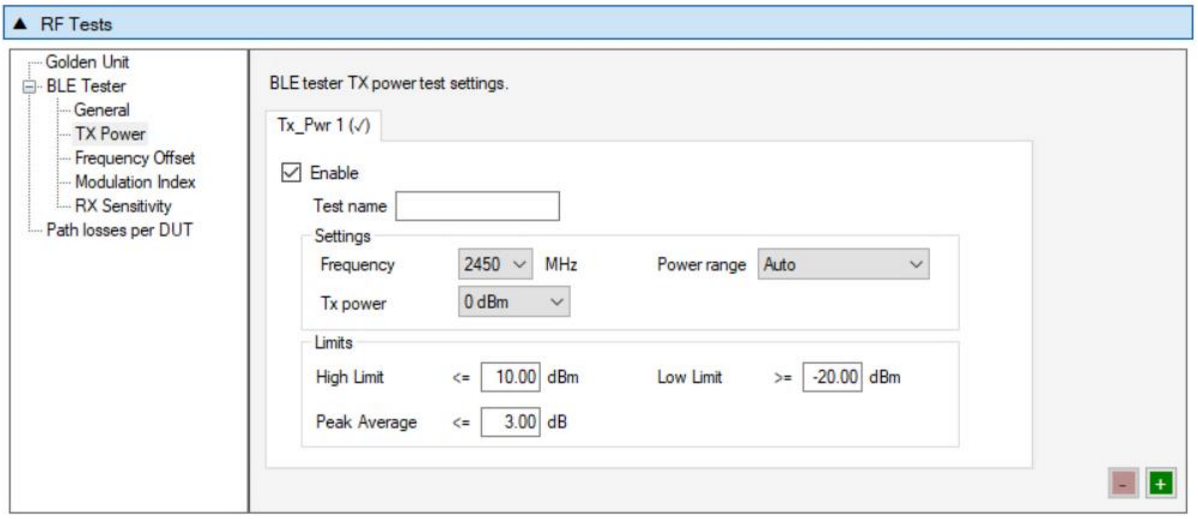



**Figure 57: BLE Tester TX Power - DA14531**

Table 46 describes the available options for the *TX Power* test using a BLE Tester instrument.

**Table 46: BLE Tester TX Power - DA14531**

| Option | Description |
|---|---|
| Enable | This option enables the specific TX power test using a BLE tester instrument. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Frequency | The BLE channel frequency used in the BLE TX power test. |
| Power range | Set the device TX output power range. Available options are:<br>● Auto (No auto option for Litepoint IQxel-M. Sets the instrument to trigger at -25dBm)<br>● +22 dBm to +7 dBm<br>● +9 dBm to -3 dBm<br>● +5 dBm to -7 dBm<br>● -4 dBm to -16 dBm<br>● -12 dBm to -26 dBm<br>● -24 dBm to -35 dBm<br>Default value is *Auto.* |
| TX power | Set the device output TX power as supported by DA14531. |
| High limit | Set the average high-power limit for the BLE TX output power pass/fail test criteria. |
| Low limit | Set the average low power limit for the BLE TX output power pass/fail test criteria. |
| Peak average | Set the peak-to-average power limit for the BLE TX output power pass/fail test criteria. |


© 2020 Dialog Semiconductor

### BLE Tester - Frequency Offset

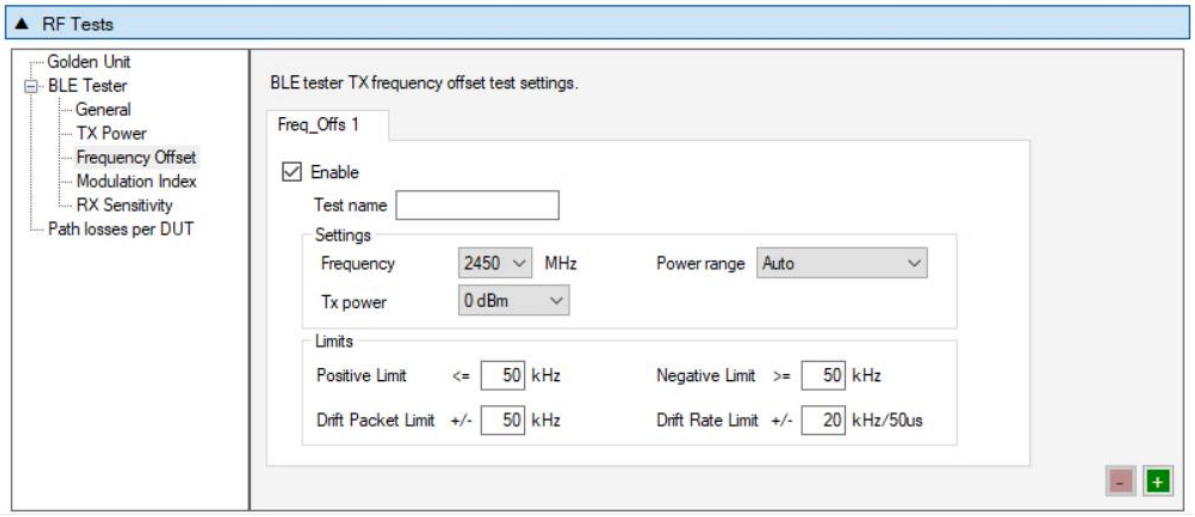



**Figure 58: BLE Tester Frequency Offset - DA14531**

Table 47 describes the available options for the *Frequency Offset* test using a BLE Tester instrument.

**Table 47: BLE Tester Frequency Offset - DA14531**

| Option | Description |
|---|---|
| Enable | This option enables the specific TX frequency offset test using a BLE tester instrument. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Frequency | The BLE channel frequency used in the BLE TX frequency offset test. |
| Power range | Set the device TX output power range. Available options are:<br>● Auto<br>● +22 dBm to +7 dBm<br>● +9 dBm to -3 dBm<br>● +5 dBm to -7 dBm<br>● -4 dBm to -16 dBm<br>● -12 dBm to -26 dBm<br>● -24 dBm to -35 dBm<br>Default value is *Auto.* |
| TX power | Set the device output TX power as supported by DA14531. |
| Positive limit | Set the maximum positive offset limit in kHz for the TX carrier frequency offset pass/fail test criteria. |
| Negative limit | Set the maximum negative offset limit in kHz for the TX carrier frequency offset pass/fail test criteria. |
| Drift packet limit | Set the overall packet drift in kHz for the TX drift pass/fail test criteria. |
| Drift rate limit | Set the drift rate limit in kHz/50 µs for the TX drift pass/fail test criteria. |


© 2020 Dialog Semiconductor

### BLE Tester - Modulation Index

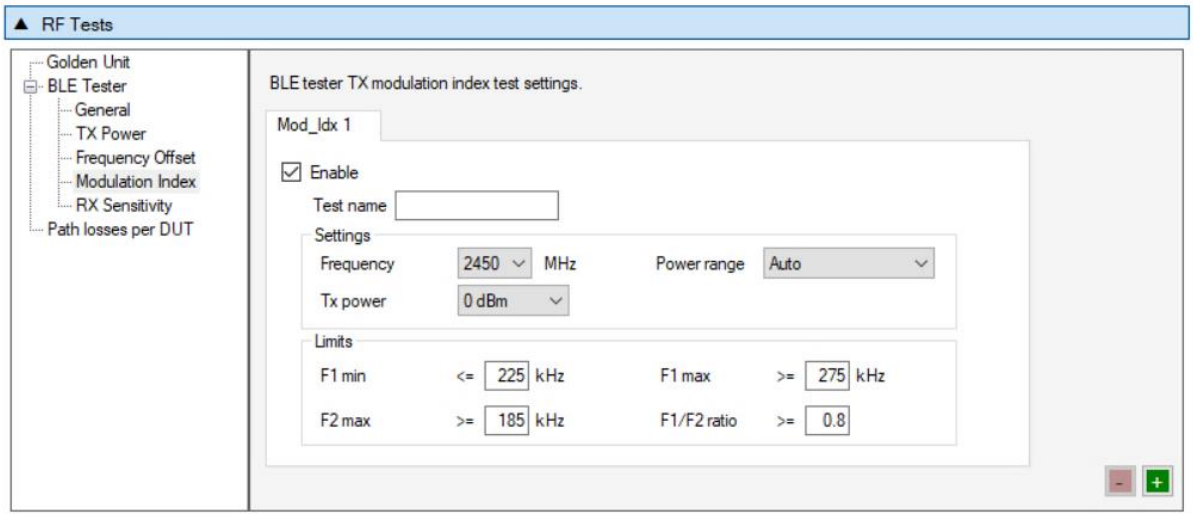



**Figure 59: BLE Tester Modulation Index - DA14531**

Table 48 describes the available options for the *Modulation Index* test using a BLE Tester instrument.

**Table 48: BLE Tester Modulation Index - DA14531**

| Option | Description |
|---|---|
| Enable | This option enables the specific TX modulation index test using a BLE tester instrument. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Frequency | The BLE channel frequency used in the BLE TX modulation index offset test. |
| Power range | Set the device TX output power range. Available options are:<br>● Auto<br>● +22 dBm to +7 dBm<br>● +9 dBm to -3 dBm<br>● +5 dBm to -7 dBm<br>● -4 dBm to -16 dBm<br>● -12 dBm to -26 dBm<br>● -24 dBm to -35 dBm<br>Default value is *Auto.* |
| TX power | Set the device output TX power as supported by DA14531. |
| F1 min | Set the F1 minimum average limit in kHz for the TX modulation index pass/fail test criteria. |
| F1 max | Set the F1 maximum average limit in kHz for the TX modulation index pass/fail test criteria. |
| F2 max | Set the F2 maximum limit in kHz for the TX modulation index pass/fail test criteria. |
| F1/F2 ratio | Set the F1/F2 maximum average ratio limit for the TX modulation index pass/fail test criteria. |


© 2020 Dialog Semiconductor

### BLE Tester - RX Sensitivity

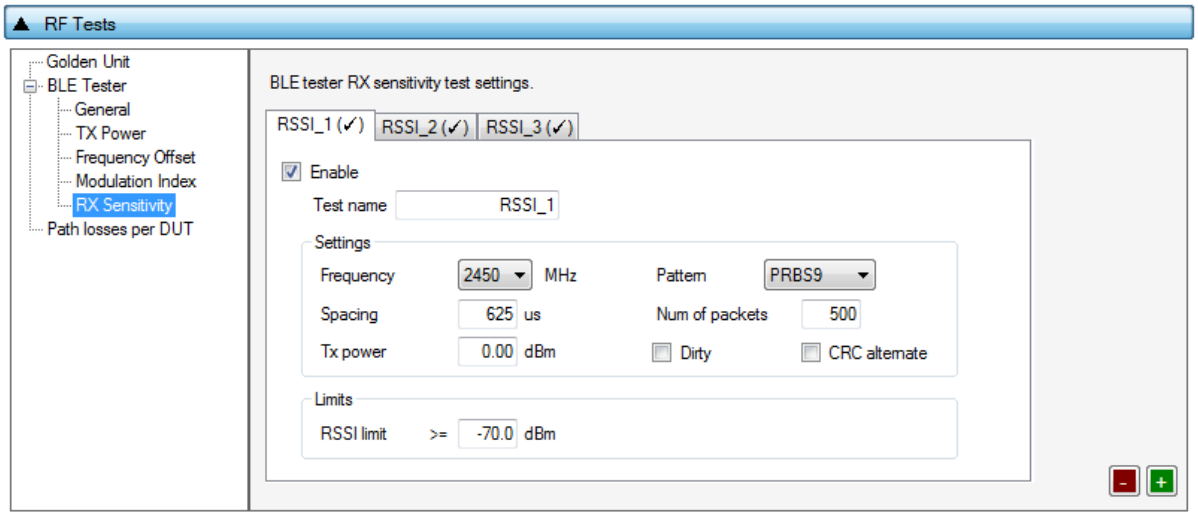



**Figure 60: BLE Tester RX Sensitivity - DA14531**

Table 49 describes the available options for the *RX Sensitivity* test using a BLE Tester instrument.

**Table 49: BLE Tester RX Sensitivity - DA14531**

| Option | Description |
|---|---|
| Enable | This option enables the specific RX sensitivity test using a BLE tester instrument. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Frequency | The BLE channel frequency used in the BLE RX sensitivity test. |
| Pattern | The bit pattern of the TX data. Available options are:<br>● PRBS9<br>● 10101010<br>● 11110000 |
| Spacing | The packet spacing in µs. |
| Num of packets | The number of packets the BLE tester instrument to transmit. |
| Tx power | The TX output power of the BLE tester instrument. Suggested values are 0 to -10 dBm. |
| Dirty | When enabled, the BLE tester packet generator can use a dirty table to transmit. |
| CRC alternate | When enabled, the BLE tester will alternatingly send packets with CRC correct and CRC incorrect. |
| RSSI limit | The RSSI limit for pass/fail criteria in the RF RX sensitivity test. If the average RSSI of the device after it has received the transmitted packets is less than this value, the test will be considered as failed. |


© 2020 Dialog Semiconductor

### Path Losses per DUT

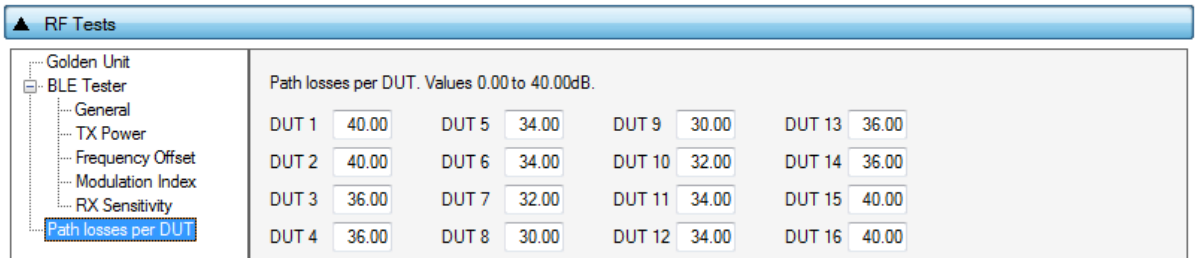



**Figure 61: Path Losses per DUT - DA14531**

Table 50 describes the available options for the *Path losses per DUT.*

Based on the relative position of each DUT during RF tests and since the RF tests are performed over the air, values can be used to correct for any path losses. These values are added to the limits of the TX Power and RSSI tests. Additional information can be found in Appendix D and Appendix F.

**Table 50: Path Losses per DUT from RF Tests DA14531 Options**

| Option | Description |
|---|---|
| DUT1-16 | Set the calibrated path loss value for each DUT. These will be added as corrections to the limits of the TX Power and RF RX RSSI tests. |

### 7.2.6.8 GPIO/LED Test

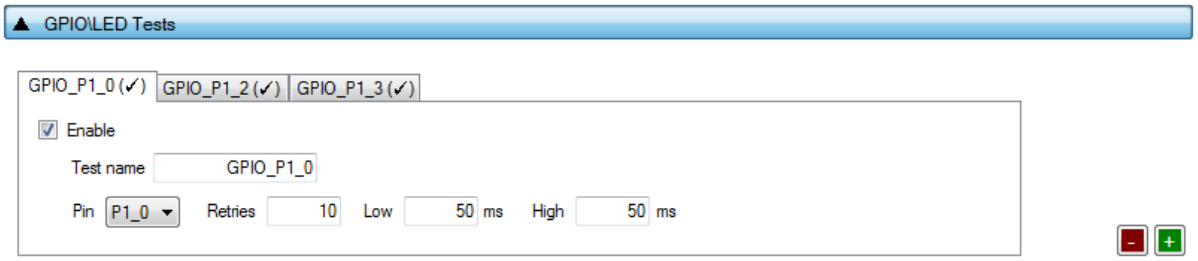



**Figure 62: GPIO/LED Tests - DA14531**

*GPIO/LED Tests* can have multiple instances with different settings. Tests can be added or removed using the two buttons (e.g. - and + in Figure 62) at the bottom right side of each panel.

**Note:** When adding or removing a test, all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

Table 51 describes the available options for the *GPIO/LED Tests* DA14531 Options.

In these tests, selected GPIO can be toggled and any LED connected to it can be visually tested.

**Table 51: GPIO/LED Tests - DA14531**

| Option | Description |
|---|---|
| Enable | This option enables the GPIO/LED toggling. Can be used for visual LED testing. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Pin | The GPIO that will be used for the specific test. |
| Retries | Number of pulses to be generated for the specific test. |
| Low | Sets the amount of the OFF time of the pulse in ms for the specific test. |
| High | Sets the amount of the ON time of the pulse in ms for the specific test. |


© 2020 Dialog Semiconductor

#### 7.2.6.9 GPIO Connection Test

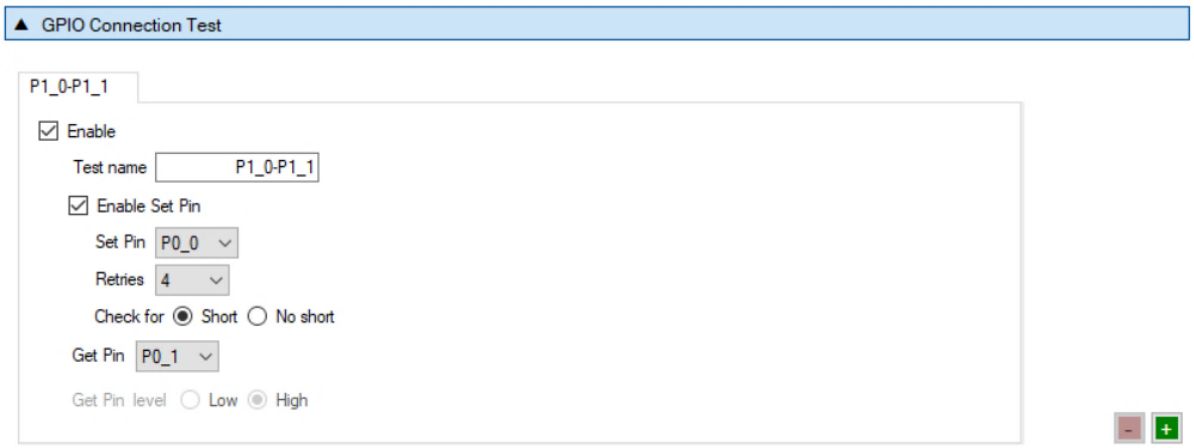



**Figure 63: GPIO Connection Test - DA14531**

*GPIO Connection Tests* can have multiple instances with different settings. Tests can be added and removed using the two buttons (e.g. - and + in Figure 63) at the bottom right side of each panel.

**Note:** When adding or removing a test all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

Table 52 describes the available options for the DA14531 *GPIO Connection Test*.

When enabled, the PLT software will check the connection of the specified GPIO (`Get Pin`) by either checking its state or the connection with another pin (`Set Pin`). In the latter case, the user gives the `Set Pin` and the state to check. It will also check for shorts between given GPIOs.

**Table 52: GPIO Connection Test - DA14531**

| Option | Description |
|---|---|
| Enable | This option enables the specific custom test. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Enable Set Pin | Enables the use of the secondary GPIO to drive the GPIO under test. When this option is set, the `Get Pin level` option will be disabled. |
| Set Pin | Select the GPIO to be tested |
| Retries | How many times the software will check for GPIO connection or short. In every retry it will change the `Set Pin` level and check the `Get Pin` level. |
| Check for Short/No short | If Short is selected, PLT will check whether the `Set Pin` has the same level with the `Get Pin` for all `Retries` tested. If `No short` is selected, PLT will check whether `Get Pin` is always low no matter what the `Set Pin` level is. |
| Get Pin | Select the GPIO to be tested. |
| Get Pin level | Sets the GPIO state the test awaits to see in the `Get Pin`. This option is disabled if the Set Pin mode is enabled. |


© 2020 Dialog Semiconductor

### 7.2.6.10 Sensor Test

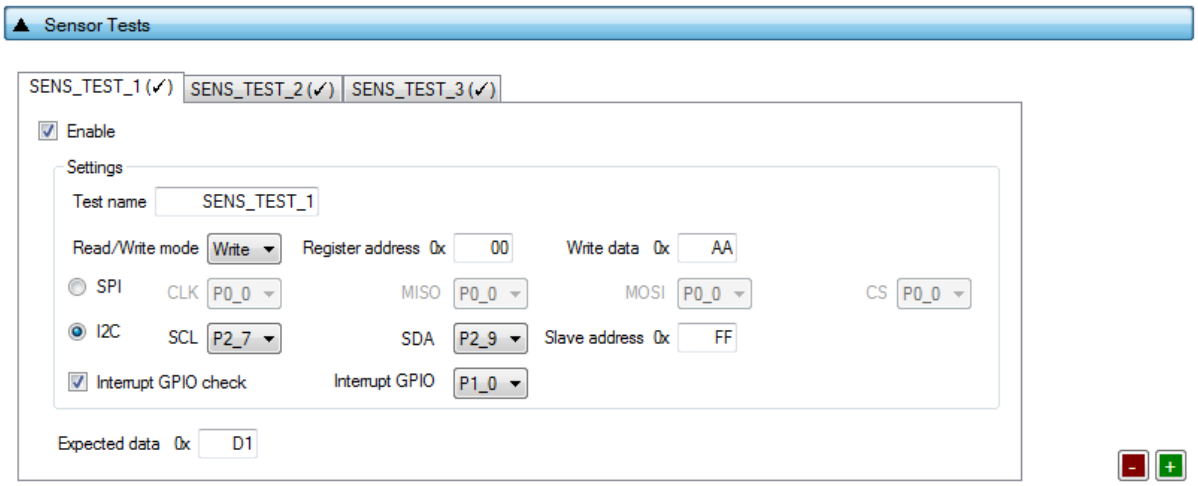



**Figure 64: Sensor Test - DA14531**

*Sensor Tests* can have multiple instances with different settings. Tests can be added and removed using the two buttons (e.g. and in Figure 64) at the bottom right side of each panel.

**Note:** When adding or removing a test all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

Table 53 describes the available options for the *Sensor Tests* DA14531 Options.

**Table 53: Sensor Tests - DA14531**

| Option | Description |
|---|---|
| Enable | This option enables the specific sensor test. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Read / Write mode | Select the sensor test procedure, to read or write. |
| Register address | The sensors register address to read or write data. |
| Write data | The byte to be written at the sensor register. |
| SPI / I2C | Select the interface that the sensor is connected to. |
| SPI - CLK | Select the GPIO for the sensor SPI CLK. |
| SPI - MISO | Select the GPIO for the sensor SPI MISO. |
| SPI - MOSI | Select the GPIO for the sensor SPI MOSI. |
| SPI - CS | Select the GPIO for the sensor SPI CS. |
| I2C - SCL | Select the GPIO for the sensor I2C SCL. |
| I2C - SDA | Select the GPIO for the sensor I2C SDA. |
| Slave address | The sensor I2C bus slave address. |
| Interrupt GPIO check | Enables the sensor interrupt signal test via GPIO. |
| Interrupt GPIO | Select the GPIO to be used as a sensor interrupt. |
| Expected data | The received sensor byte that will be expected on a successful operation. |


© 2020 Dialog Semiconductor

#### 7.2.6.11 Custom Test

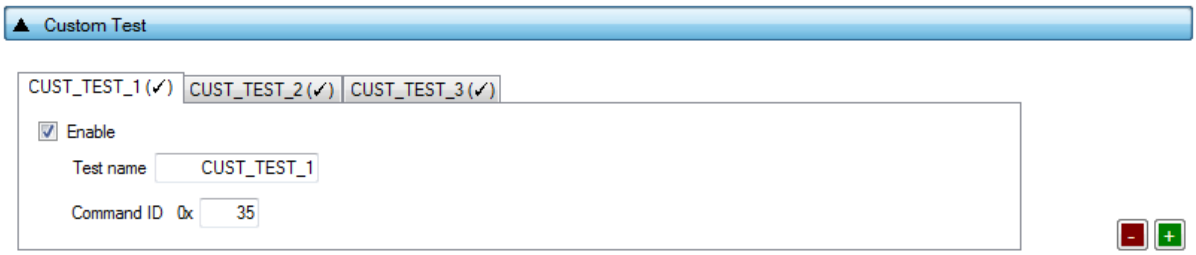



**Figure 65: Custom Test - DA14531**

*Custom Tests* can have multiple instances with different settings. Tests can be added and removed using the two buttons (e.g. - and + in Figure 65) at the bottom right side of each panel.

**Note:** When adding or removing a test all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

Table 54 describes the available options for the DA14531 *Custom Tests*.

When enabled, the PLT software will send an HCI command over UART to activate a customer defined test that will be executed on the DUTs. The HCI custom test command will contain a single byte as data (the *Command ID* byte), to be used mainly as identification for the specific test in the customized firmware. The default functionality of the production test firmware is to respond with the same `Command ID`. Otherwise, PLT will consider the test as failed.

**Table 54: Custom Tests - DA14531**

| Option | Description |
|---|---|
| Enable | This option enables the specific custom test. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Command ID | The byte that will be sent to the device running the production test firmware. |

#### 7.2.6.12 External 32kHz Test

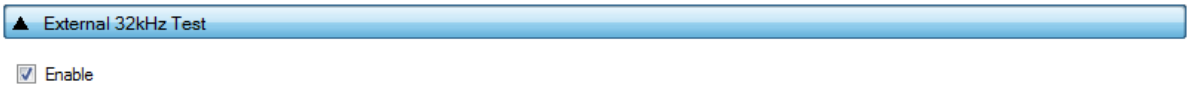



**Figure 66: External 32kHz Test - DA14531**

Table 55 describes the available options for the DA1458x *External 32kHz Test*.

When enabled, the PLT software will verify the correct operation of the External 32kHz crystal on each DUT.

**Table 55: External 32kHz Test - DA14531**

| Option | Description |
|---|---|
| Enable | This option enables the External 32kHz test. |


© 2020 Dialog Semiconductor

### 7.2.6.13 Current Measurement Test

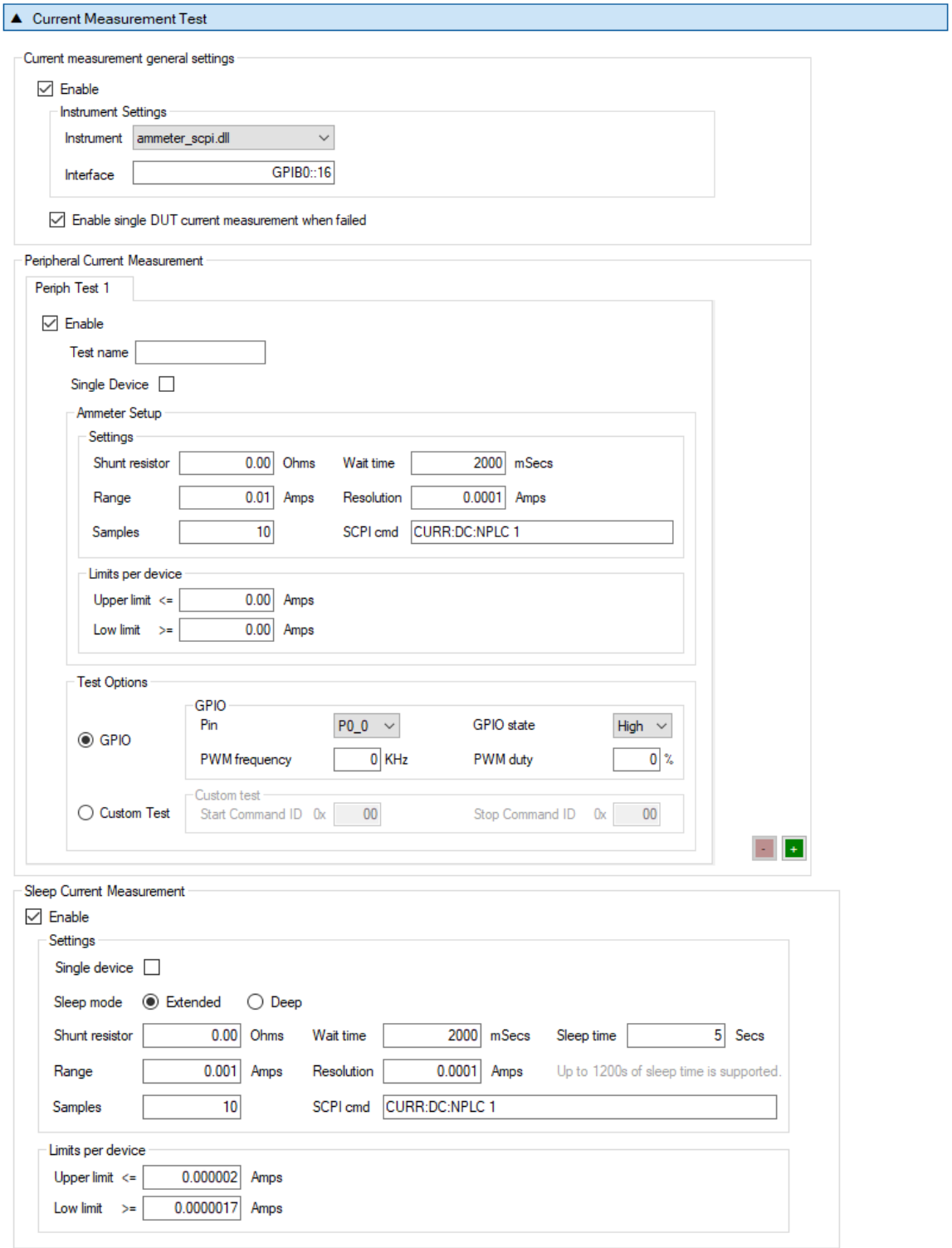



**Figure 67: Current Measurement Test - DA14531**

In this test, an external ammeter instrument can be used to calculate the total current consumption of all the active DUTs at the time of the sampling. The ammeter instrument can be connected in the blue banana plugs as described in Current Measurements or to an external power supply (if present) depending the selected VBAT/Reset Mode.

During measurement, PLT will control the instrument using the `ammeter_driver` DLL [1]. Table 56 describes the instrument selection settings found by the `ammeter_driver` DLL, Table 57 describes

 © 2020 Dialog Semiconductor

the settings used for each of the peripheral current measurement tests and Table 58 describes the current measurement options for each sleep state.

**Note:** Modifications in the production test firmware are mandatory in order to achieve the correct current consumption of a specific hardware design. Running the default firmware without any modifications may result in increased current consumption.

*Peripheral Current Measurement Tests* can have multiple instances with different settings. Tests can be added and removed using the two buttons (e.g. and in Figure 67) at the bottom right side of each panel.

**Note:** When adding or removing a test all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

**Table 56: Current Measurement Test – General Settings - DA14531**

| Option | Description |
|---|---|
| Enable | This option enables all Current Measurement tests, which include:<br>● Idle Current Measurement<br>● Extended Sleep Current Measurement<br>● Deep Sleep Current Measurement<br>Only one of the Extended/Deep sleep current measurements can be selected, meaning that the other one will be disabled. |
| Instrument | Select the Ammeter instrument DLL name. Names are shown only if an ammeter instrument DLL exists in the project `ammeter_instr_plugins` folder. |
| Interface | The interface of the instrument to be used by the driver. |
| Enable single DUT current measurement when failed | If this option is enabled and if the measurement taken is outside of the limits, PTL will reset all devices and begin a firmware download and measure the current to each device separately, in order to identify which exact device failed. |

**Table 57: Peripheral Current Measurement - DA14531**

| Option | Description |
|---|---|
| Enable | This option enables the specific peripheral current measurement test. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Single Device | If this option is enabled, PLT will measure the current consumption one device at the time. Initially, it will power off all DUTs. It will then power on one by one, download firmware and measure the current individually. This procedure takes a lot of time. It should only be used for production setup purposes, to identify the correct limits by measuring multiple DUTs and taking the average. |
| Shunt resistor | The value of the shunt resistor used for peripheral measurement. Applicable only if a voltmeter or a NI-DAQ is used to measure current instead of a DMM. Values from 0 to 9999 are supported. |
| Wait time | The time in ms the PLT will wait before taking a current measurement, after it has sent an instruction to the DUTs to go into sleep state. Supported values are 1 to 500000ms. |
| Range | The range value in Ampere units that the ammeter instrument will operate. Supported values are 0 to 9999 and default value is 0.001A. When the `ammeter_scpi.dll` is used, if this value is set to zero, the instrument will use automatic range functionality. |
| Resolution | The ammeter instrument resolution value in Ampere units. |
| Samples | The number of samples that the ammeter instrument will read and average. 1 to 1000 is supported. |

 © 2020 Dialog Semiconductor

| Option | Description | |
|---|---|---|
| SCPI cmd | A SCPI command to be passed to the ammeter instrument just before the measurement is taken. Supports multiple commands separated with a column. Up to 256 characters are supported. | |
| Upper limit | The upper limit value for a single DUT.<br>Next to this input field the total upper limit current consumption for all the enabled DUTs is shown, which is the value to be used during testing.<br>**Note:** Some DUTs may fail before the current measurement test start. PLT will automatically re-calculate the total upper limit by using the value given for a single DUT multiplied with the number of the remaining DUTs at the time that the test will run. | |
| Low limit | The low limit value for a single DUT.<br>Next to this input field the total low limit current consumption for all enabled DUTs is shown, which is the actual value to be used during testing.<br>**Note:** Some DUTs may fail before the current measurement test starts. PLT will automatically re-calculate the total low limit by using the value given for a single DUT multiplied with the number of the remaining DUTs at the time that the test will run. | |
| Test Options | Select between a PWM GPIO test and custom test.<br>**Note:** For the custom tests to work, a modified production test firmware must be created with tests that set the DUTs to specific states before the current measurement test. Each test must be assigned to a specific opcode. The custom tests are the exact same as in Custom Test. | |
| Test Options - GPIO | Pin | Sets the GPIO to toggle with the PWM pulse. |
| | GPIO state | Sets the active state of the GPIO. |
| | PWM frequency | Sets the PWM frequency. |
| | PWM duty | Sets the PWM duty cycle. |
| Test Options – Custom Test | Start Command ID | The opcode of the custom test that sets the state of the DUT. |
| | Stop Command ID | The opcode of the custom test that restores the DUT to its original state. |

**Table 58: Current Measurement Test - Sleep Current Measurement - DA14531**

| Option | Description |
|---|---|
| Enable | This option enables the specific current measurement using the ammeter instrument provided in the *Instrument* section. |
| Single Device | If this option is enabled, PLT will measure the current consumption one device at the time. Initially, it will power off all DUTs. It will then power on one by one, download firmware and measure the current individually. This procedure takes a lot of time. It should only be used for production setup purposes, to identify the correct limits by measuring multiple DUTs and taking the average. |
| Sleep mode | User can select either the Extended or the Deep sleep mode. |
| Shunt resistor | The value of the shunt resistor used for sleep current measurement. Applicable only if a voltmeter or a NI-DAQ is used to measure current instead of a DMM. Values from 0 to 9999 are supported. |
| Wait time | The time in ms the PLT will wait before taking a current measurement, after it has sent an instruction to the DUTs to go into a specific sleep state. Supported values are 1 to 500000ms. |
| Sleep time | The time in seconds that the DUTs will remain in sleep mode. A timer in the production test firmware will wake up the devices. Supported values are 1 to 9sec for DA14580/1/2/3 and up to 1200sec for the rest. |
| Range | The range value in Ampere units that the ammeter instrument will operate. Supported values are 0 to 9999 and default value is 0.001A. When the `ammeter_scpi.dll` is used, if this value is set to zero, the instrument will use the automatic range functionality. |

 © 2020 Dialog Semiconductor

| Option | Description |
|---|---|
| Resolution | The ammeter instrument resolution value in Ampere units. |
| Samples | The number of samples that the ammeter instrument will read and average. 1 to 1000 is supported. |
| SCPI cmd | An SCPI command to be passed to the ammeter instrument just before the measurement is taken. Supports multiple commands separated with a column. Up to 256 characters are supported. |
| Upper limit | The upper limit value for a single DUT.<br>Next to this input field the total upper limit current consumption for all the enabled DUTs is shown, which is the value to be used during testing.<br>**Note:** Some DUTs may fail before the current measurement test starts. PLT will automatically re-calculate the total low limit by using the value given for a single DUT multiplied with the number of the remaining DUTs at the time that the test will run. |
| Low limit | The low limit value for a single DUT.<br>Next to this input field the total upper limit current consumption for all the enabled DUTs is shown, which is the value to be used during testing.<br>**Note:** Some DUTs may fail before the current measurement test starts. PLT will automatically re-calculate the total low limit by using the value given for a single DUT multiplied with the number of the remaining DUTs at the time that the test will run. |

#### 7.2.6.14 Temperature Measurement Test

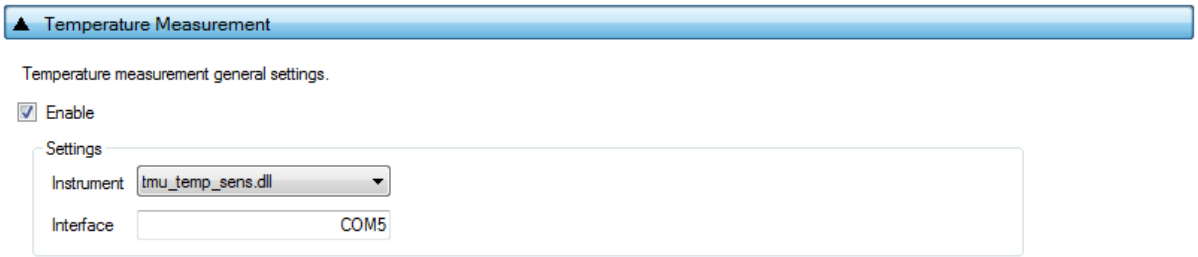



**Figure 68: Temperature Measurement Test - DA14531**

Table 59 describes the available options of the DA14531 *Temperature Measurement Test*.

**Table 59: Temperature Measurement Test - DA14531**

| Option | Description |
|---|---|
| Enable | This option enables the temperature measurement test. |
| Instrument | Selects the temperature measurement DLL. Names are shown only if a temperature measurement instrument DLL exists in the project folder `temp_meas_instr_plugins`. |
| Interface | The interface of the instrument to be used by the driver. |


© 2020 Dialog Semiconductor

#### 7.2.6.15 Scan Test

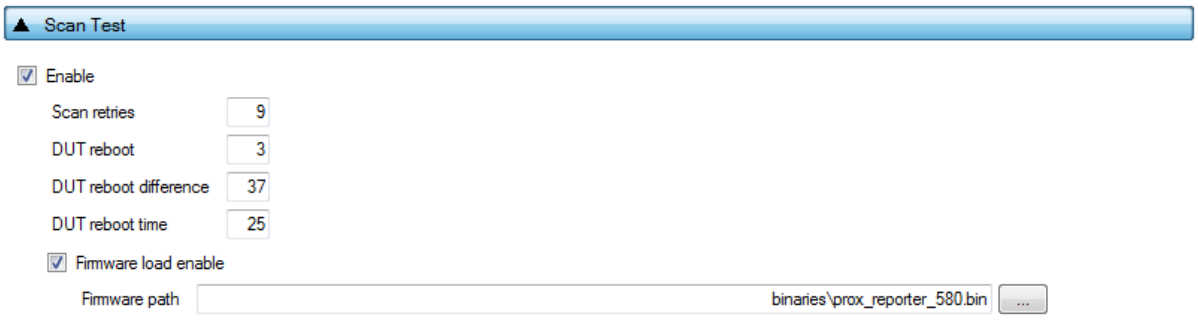



**Figure 69: Scan Test - DA14531**

Table 60 describes the available options for the DA14531 *Scan Test*.

By enabling this test, the Golden Unit will scan for the DUT's BD addresses advertised after the customer firmware has been burned. For this test to work, a bootable firmware with the ability to advertise with the BD address given by the PLT must be burned into each DUT. Additionally, the BD addresses provided by the PLT should be burned into OTP memory, such that the devices advertise with the BD addresses that the tool uses.

**Table 60: Scan Test DA14531 Options**

| Option | Description |
|---|---|
| Enable | This option enables the Scan test. |
| Scan retries | The total number of BLE advertising scans the Golden Unit will perform. |
| DUT reboot | Define after how many retries the PLT will reboot the DUTs. |
| DUT reboot difference | Set the time difference between each DUT when the PLT reboots the devices, in order to avoid air collisions. |
| DUT reboot time | The time the VBAT will remain low during the device reboot. This value is time in ms*100 (e.g. 15 is 1500ms). |
| Firmware load enable | By enabling this option, a new image will be downloaded to all active DUTs before scanning for BLE advertising devices. |
| Firmware path | The path of the binary file to download to the devices for the scan test. |


© 2020 Dialog Semiconductor

### 7.2.7 Memory Functions (DA14531)

This section describes the Memory Functions settings available when using DA14531 devices. Memory functions include OTP, SPI Flash, and I2C EEPROM memory programming.

#### 7.2.7.1 OTP Memory

This test enables the OTP memory programming. Table 61 describes the available options for the *OTP Memory* image write operation.

**Figure 70: OTP Memory - DA14531**

**Note:** If the binary is larger than the available OTP image area (OTP memory excluding the header area), the PLT software will split the binary into two parts. The first part will contain only the OTP image area. The second part will contain the OTP header fields, split in OTP words. PLT will burn the non-zero words one by one, as single OTP entries in the OTP header area. The check empty feature will handle the first part as an OTP image binary. The second part will be checked word by word.

**Table 61: OTP Memory - DA14531**

| Option | Description |
|---|---|
| Write enable | This option enables the OTP image write operation. |
| ● No check<br>● Check empty<br>● Check if data match<br>● Skip if written | Memory protection options:<br>**No check:** No protection is enabled. PLT will attempt to burn the OTP memory without running any check.<br>**Check empty:** PLT will first check if the OTP memory is empty.<br>**Check if data match:** PLT will first check if the memory to be burned is empty. If it is not, it will compare the contents with the data to be burned. If the data are not the same the test will fail without making any changes to the memory. If the data are the same, PLT will not burn the memory to prevent using the OTP repair memory and continue with success.<br>**Skip if written:** PLT will read the contents of OTP memory. If it contains any data, it will skip writing without producing any error. |
| Verify image | If this option is enabled, PLT will read back the contents of the OTP memory and compare them to the original image file. If these do not match it will fail. |
| Burn image length to OTP header (OTP DMA length) | If this option is selected, PLT software will calculate the length in OTP words and burn it to the Memory Header (DA14531) – OTP DMA length. When selected, it will disable the OTP DMA length option in Memory header tab. |
| Different image per DUT | If this option is selected, a different image per DUT can be burned into the OTP. The image name must be specific for each DUT, as described below. |


© 2020 Dialog Semiconductor

| Option | Description |
|---|---|
| Image path | Via this field, the user specifies the image file to be burned into the OTP. A `.bin` binary file of any name can be selected.<br>Depending on the size of the selected binary, PLT will inform the user if the binary contains both the image and the header part or if it exceeds the maximum supported size.<br>If *Different image per DUT* is selected, the user only selects the directory of the images. In that case, the binary file names must have the following format: `img_xx.bin`, where 'X' denotes the DUT number. For example, if the user has activated DUTs 1, 5 and 10 then `img_01.bin`, `img_05.bin` and `img_10.bin` binary files should exist in the selected OTP image path as shown in Figure 71.<br>Range of numbers is `img_01.bin` … `img_16.bin`. |

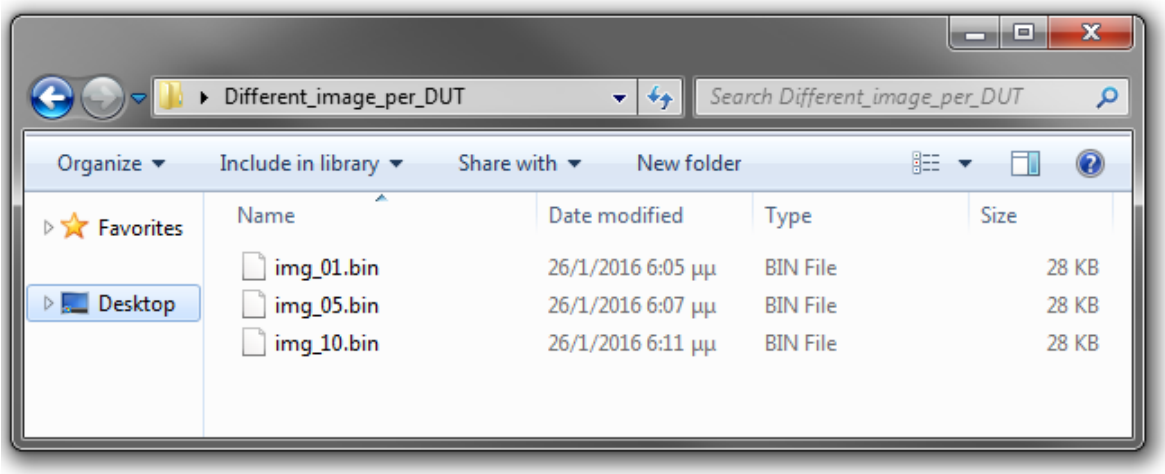



**Figure 71: Different Image per DUT Folder Example**

### 7.2.7.2 SPI Flash Memory

This section explains the settings of the SPI Flash Memory operations.

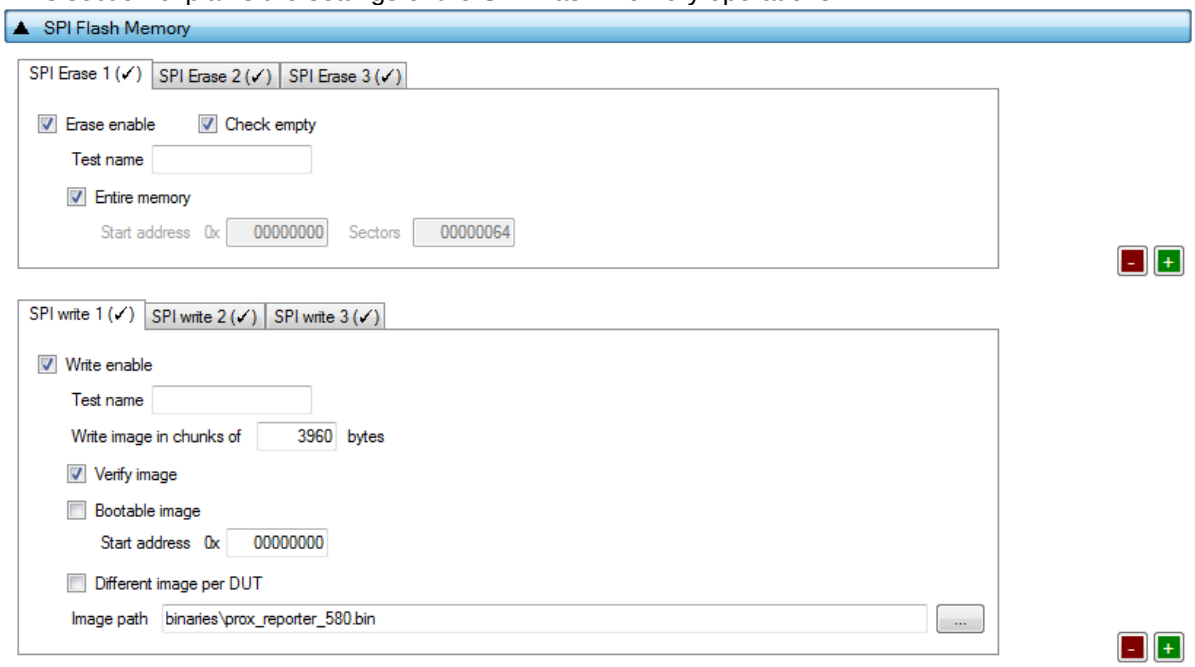



**Figure 72: SPI Flash Memory - DA14531**


© 2020 Dialog Semiconductor

Both erase and write tests can have multiple instances with different settings. Tests can be added and removed using the two buttons (e.g. and in Figure 72) at the bottom right side of each panel.

**Note:** When adding or removing a test all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

The SPI Flash memory should be erased before any image is written to it. Table 62 describes the available options for the *SPI Flash Erase* operation.

**Table 62: SPI Flash Erase - DA14531**

| Option | Description |
|---|---|
| Erase enable | This option will enable the SPI flash erase operation. |
| Check empty | After flash erasure, the PLT software can verify the erasure result by sending a specific command to the `flash_programmer_531.bin` firmware running in the DUT. The firmware will read the SPI flash memory and check if it is empty. The result will be returned to the PLT software. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Entire memory | If selected, the entire memory will be erased. Otherwise, the user can give a start address and a specific number of sectors to be erased. |
| Start address | The user can enter a specific start address for the SPI flash erasure operation. |
| Sectors | The number of sectors to erase, starting from the *Start address* as explained above. |

After all the SPI flash erase operations have finished, the SPI image write tests will begin. Table 63 describes the available options for the *SPI Flash Image Write* operation.

**Table 63: SPI Flash Image Write - DA14531**

| Option | Description |
|---|---|
| Write enable | This will enable the specific SPI flash image programming operation. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Write image in chunks of "`user_input`" bytes. | During memory programming, PLT will split the image into chunks of size defined in this field. Values from 1 byte to 28664 bytes are supported. |
| Verify image | By selecting this option, the PLT software will read back the contents of the SPI flash memory, after an image was burned. It will compare them to the original image file. If these do not match the SPI memory programming will fail. |
| Bootable image | Is this is enabled PLT will write a boot header at SPI address 0 and burn the image at SPI address 0x8. |
| Start address | Users can configure the SPI flash start address image write operation. If *Bootable image* is selected, this option is disabled. |
| Different image per DUT | If this option is selected, a different image per DUT will be burned into the SPI flash. The image name must be specific for each DUT, as described below. |
| Image path | Via this field, the user specifies the image file to be burned into the SPI Flash memory. A `.bin` binary file of any name can be selected.<br>If option *Different image per DUT* is selected the user only selects the directory of the images. In that case, the binary file names must have the following format: `img_0X.bin`, where 'X' denotes the DUT number. For example, if the user has activated DUTs 1, 5 and 10 then `img_01.img`, `img_05.img` and `img_10.img` binary files should exist in the selected SPI image path, as shown in Figure 71. |

 © 2020 Dialog Semiconductor

### 7.2.7.3 I2C EEPROM Memory

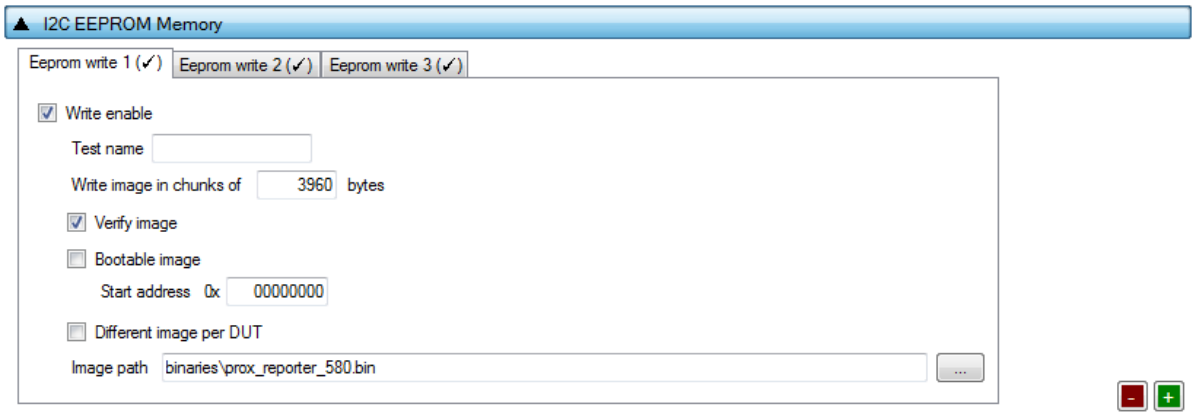



**Figure 73: I2C EEPROM Memory - DA1458x**

In this section, an I2C EEPROM memory can be programmed. The I2C EEPROM image write tests can be performed multiple times with different settings each time. Tests can be added and removed using the two buttons (e.g. and in Figure 73) at the bottom right side of each panel.

**Note:** When adding or removing a test all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

Table 64 describes the available options for the *I2C EEPROM Image Write* operation.

**Table 64: I2C EEPROM Image Write - DA1458x**

| Option | Description |
|---|---|
| Write enable | This will enable the specific I2C/EEPROM image programming test. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Write image in chunks of "`user_input`" bytes. | During memory programming, PLT will split the image into chunks of size defined in this field. Values from 1 byte to 32760 bytes are supported. |
| Verify image | By selecting this option, the PLT software will read back the contents of the EEPROM and compare them to the original image file. If these do not match the EEPROM memory programming will fail. |
| Bootable image | Is this is enabled PLT will write a boot header at EEPROM address 0 and burn the image at the EERPOM address 0x20. |
| Start address | Users can configure the EEPROM start address image write operation. If *Bootable image* is selected, this option is disabled. |
| Different image per DUT | If this option is selected, a different image per DUT will be burned into the EEPROM memory. The image name must be specific for each DUT, as described below. |
| Image path | This field specifies the image file to be burned into the EEPROM memory. A `.bin` binary file of any name can be selected.<br>If option *Different image per DUT* is selected, the user only selects the directory of the images. In that case, the binary file names must have the following format: `img_0X.bin`, where 'X' denotes the DUT number. For example, if the user has activated DUTs 1, 5 and 10 then `img_01.img`, `img_05.img` and `img_10.img` binary files should exist in the selected image path, as shown in Figure 71. |


© 2020 Dialog Semiconductor

#### 7.2.7.4 Memory read

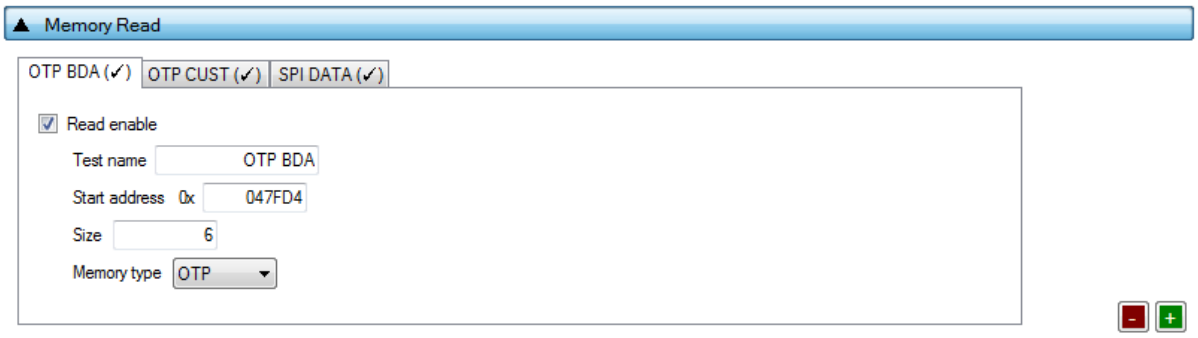



**Figure 74: Memory Read Test - DA14531**

*Memory Read Tests* can have multiple instances with different settings. Tests can be added or removed using the two buttons (e.g. - and + in Figure 74) at the bottom right side of each panel.

**Note:** When adding or removing a test, all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

Table 65 describes the memory read test options. With this test, the user can read up to 64MBytes of data from any address and any available memory, such as OTP, SPI Flash and EEPROM. An example of how the data appears in the log file is shown in Figure 75. If data to be read are greater than 256 bytes, then a file will be created to store the data under folder `mem_read_test` in the PLT execution path.

```
Memory read operation initialized. Memory read test name=[OTP BDA].
Memory read operation started. Memory read test name=[OTP BDA].
Memory read operation ended OK. Test name [OTP BDA]. Memory=[OTP]. Addr=[0x47fd4]. Size=[6]. Data=[0a0000808080].
```

**Figure 75: Memory Read Test Example Log File - DA14531**

**Table 65: Memory Read Test - DA14531**

| Option | Description |
|---|---|
| Read enable | This will enable the memory read test. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Start address | Configures the start address for the read test. DA14580/1/2/3 OTP memory 0x40000 offset should be used (e.g. BD address is written in 0x47FD4). DA14585/6 OTP memory offset is at 0x0 (e.g. BD address is written in 0xFFA8). DA14580/1/2/3 OTP valid address is 0x40000 to 0x47FFF and DA14585/6 OTP address 0x0000-0x10000. |
| Size | Number of bytes to read, up to 64MBytes. If data to be read are greater than 256 bytes, then a file will be created to store the data under folder `mem_read_test` in the PLT execution path. |
| Memory type | The type of memory to read the data from. Available options are OTP, SPI FLASH, and I2C EEPROM.<br>**Note:** For the SPI FLASH and EEPROM memories, the pin configurations are taken from the SPI Flash Configuration and I2C EEPROM Configuration sections. |


© 2020 Dialog Semiconductor

### 7.2.8 Memory Header (DA14531)

This section describes the OTP header programming settings.

#### 7.2.8.1 General

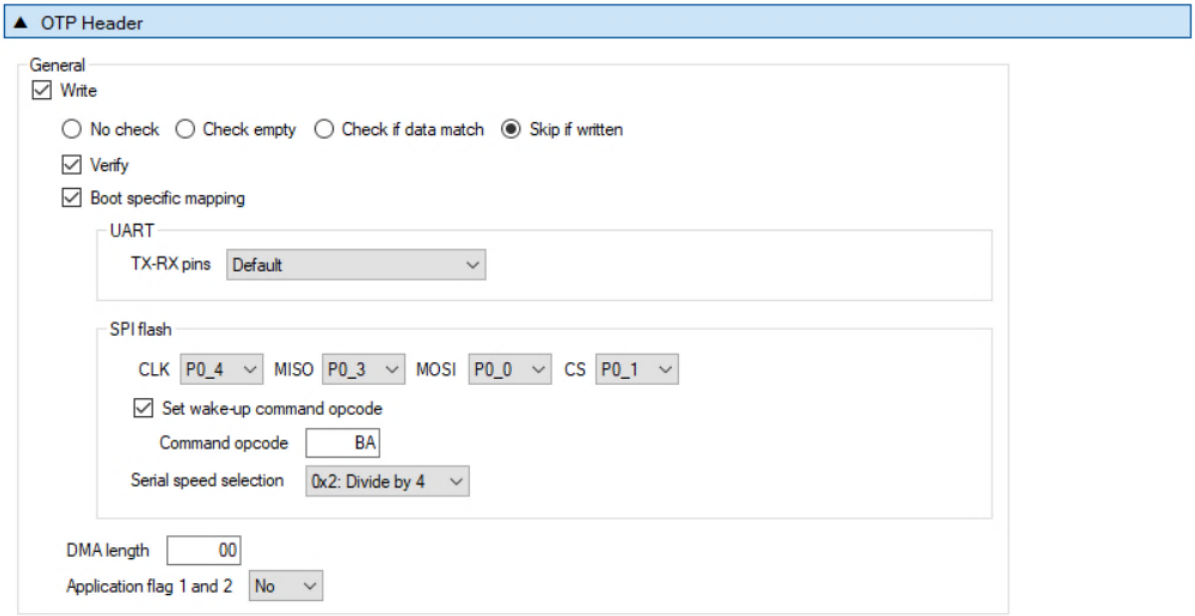



**Figure 76: OTP Header - DA14531**

Table 66 describes the available options for DA1458x *OTP Header* programming.

**Table 66: OTP Header - DA14531**

| Option | Description |
|---|---|
| Write | This option enables the OTP header programming. |
| • No check<br>• Check empty<br>• Check if data match<br>• Skip if written | Memory protection options:<br>**No check:** No protection is enabled. PLT will attempt to burn the OTP memory without running any check.<br>**Check empty:** PLT will first check if the OTP memory to be burned is empty. If it is empty will burn it.<br>**Check if data match:** PLT will first check if the memory to be burned is empty. If it is not, it will compare the contents with the data to be burned. If the data are not the same the test will fail without making any changes to the memory. If the data are the same, PLT will not burn the memory to prevent using the OTP repair memory and continue with success.<br>**Skip if written:** PLT will read the contents of OTP memory. If it contains any data, it will skip writing without producing any error. |
| Verify | Read back each OTP header value and compared with the original one to verify a successful write. |
| Boot specific mapping | Enables external booting from a specific SPI interface or UART pin configuration. |
| UART TX-RX pins | Select which pins will be used to boot from an external device through UART |
| SPI CLK | Sets the GPIO for the CLK pin of the SPI bus. |
| SPI MISO | Sets the GPIO for the MISO pin of the SPI bus. |
| SPI MOSI | Sets the GPIO for the MOSI pin of the SPI bus. |
| SPI CS | Sets the GPIO for the CS pin of the SPI bus. |

 © 2020 Dialog Semiconductor

| Option | Description |
|---|---|
| Set wake-up command opcode | Default wake-up command opcode is “AB”. If a different one is needed to be used, it can be set using this flag and the following field. |
| Command opcode | The command opcode to be used for the wake-up. |
| Serial speed selection | Division factor for SPI. |
| 32 kHz source | Selects the low power 32 kHz clock source. |
| DMA length | The size (in words) for the DMA controller to copy from OTP to system RAM during boot. Should match the OTP image size. Max value for the DA14531 devices is 0x1FF0.<br>**Note:** This option will be disabled if `Burn image length to OTP header` option in OTP Memory**Error! Reference source not found.** is enabled. |
| Application flag 1 and 2 | If this option is set, the device will boot only from the OTP memory. Used for a production ready device. There is no other means to access the device apart from JTAG, if this is still enabled in the OTP header. |

#### 7.2.8.2 BD Address

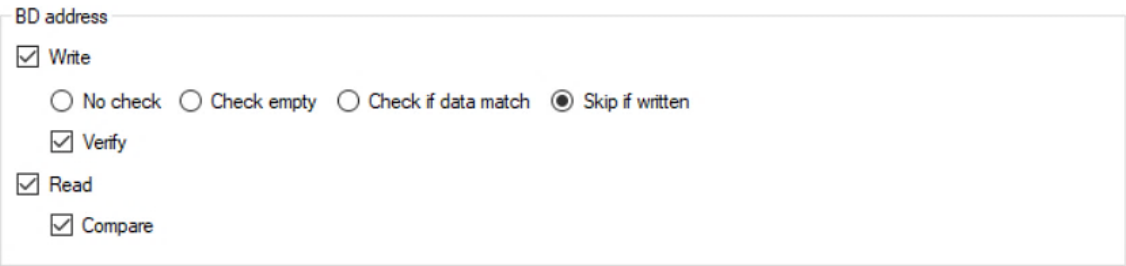



**Figure 77: BD Address - DA14531**

The BD address can be written independently from the rest of the OTP header fields described before. Table 67 describes the available options for the *BD Address* programming.

**Table 67: BD Address - DA14531**

| Option | Description |
|---|---|
| Write | When selected, the BD address will be written in the OTP Header. |
| ● No check<br>● Check empty<br>● Check if data match<br>● Skip if written | Memory protection options:<br>**No check:** No protection is enabled. PLT will attempt to burn the OTP memory without running any check.<br>**Check empty:** PLT will first check if the OTP memory to be burned is empty. If it is empty will burn it.<br>**Check if data match:** PLT will first check if the memory to be burned is empty. If it is not, it will compare the contents with the data to be burned. If the data are not the same the test will fail without making any changes to the memory. If the data are the same, PLT will not burn the memory to prevent using the OTP repair memory.<br>**Skip if written:** PLT will read the contents of OTP memory. If it contains any data, it will skip writing without producing any error. |
| Verify | When selected, the BD address will be read back from the OTP Header and will be compared to the original. |
| Read | This option will read the BD address written in the OTP Header field. It is a standalone memory operation. It does not depend on the previous tests to run, but it is necessary for the following *Compare* test. |
| Compare | If the *Read* option is enabled, a comparison will be performed between the read BD address and the BD address entered in the DUT by the PLT, as described in the BD address DUT assignment method. |


© 2020 Dialog Semiconductor

#### 7.2.8.3 Custom Memory Data

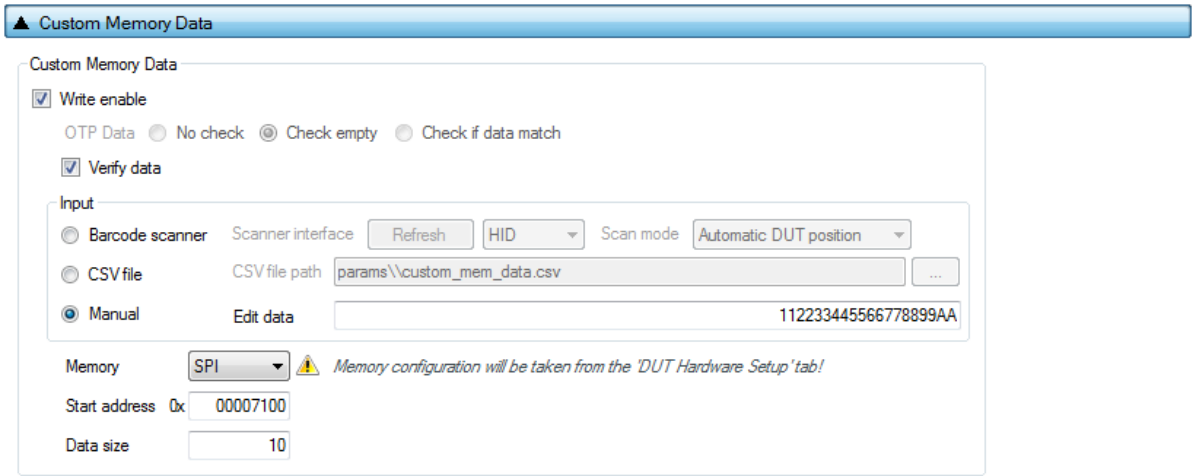



**Figure 78: Custom Memory Data - DA14531**

Table 68 describes the *Custom Memory* data test options. With this test, the user can write any data to any address to any available memory for DA14531 devices, such as OTP, SPI flash and EEPROM. Data input modes can be a Barcode Scanner, a CSV file or data entered manually.


© 2020 Dialog Semiconductor

**Table 68: Custom Memory Data - DA14531**

| Option | Description | |
|---|---|---|
| Write enable | This option enables the custom data programming. | |
| Verify data | When selected, the data written will be read back and compared to the original. | |
| • No check<br>• Check empty<br>• Check if data match<br>• Skip if written<br>(Only available when OTP memory or CSV file as input is selected) | Memory protection options:<br>**No check:** No protection is enabled. PLT will attempt to burn the OTP memory without running any check.<br>**Check empty:** PLT will first check if the OTP memory to be burned is empty. If it is empty will burn it.<br>**Check if data match:** PLT will first check if the memory to be burned is empty. If it is not, it will compare the contents with the data to be burned. If the data are not the same the test will fail without making any changes to the memory. If the data are the same, PLT will not burn the memory to prevent using the OTP repair memory.<br>**Skip if written:** PLT will read the contents of OTP memory. If it contains any data, it will skip writing without producing any error. | |
| • Barcode scanner<br>• CSV file<br>• Manual data | **Note:** Barcode scanner mode is only available with GUI PLT Application. CLI PLT Application does **NOT** support this feature. | |
| | Scanner interface (Barcode scanner) | Selection of the Barcode scanner input. Both HID and COM port interfaces are supported. This list provides an HID and all available COM ports as selectable options.<br>Any HID device is supported that includes newline (CR-LF) characters at the end of the scanned data.<br>For the COM port interface, a common USB to UART barcode scanner is supported. PLT has been tested with Honeywell Xenon 1900. Appendix M describes the setup procedure.<br>This option is the exact same option as in Scan Mode and the DA1469x devices in Custom Memory Data. |
| | Scan mode (Barcode scanner) | **Scan DUT position:** In this mode, users must first scan the DUT position number and then the BD address. The string for the position of each DUT is "TEST POSITION 0xx". "xx" is the position number.<br>**Automatic DUT position:** Scanned BD address will be assigned to the selected DUT. The DUT selection is automatically been made, starting from the first active DUT and selecting the next one after a successful BD address scan. Users can change the selected DUT using the controls on the GUI PLT screen shown in Figure 112.<br>This option is the exact same option as in Scan Mode and the DA1469x devices in Custom Memory Data. |
| | CSV file path (CSV file) | Path to the CSV file containing data for each device discriminated using BD addresses. The CSV file format is described in Custom data CSV file format. |
| | Edit data (Manual data) | Hexadecimal data input of up to 256 bytes. These data will be burned to all active DUTs. |
| Memory | Memory type selection to burn the data. Available options are OTP, SPI and EEPROM.<br>**Note:** For the FLASH and EEPROM memories, the pin configurations are taken from the SPI Flash Configuration and I2C EEPROM Configuration sections. These options must be enabled in order for the *Memory Read* test to operate successfully. | |
| Start address | Memory address offset to begin burning the data. DA14531 OTP valid address is 0x0000 to 0x7FFF. | |
| Data size | The size of the memory data to burn. In barcode scanner, the data size is the number of scanned ASCII characters. In manual data, data size is the number of bytes. | |

 © 2020 Dialog Semiconductor

### 7.2.8.4 OTP Configuration Script

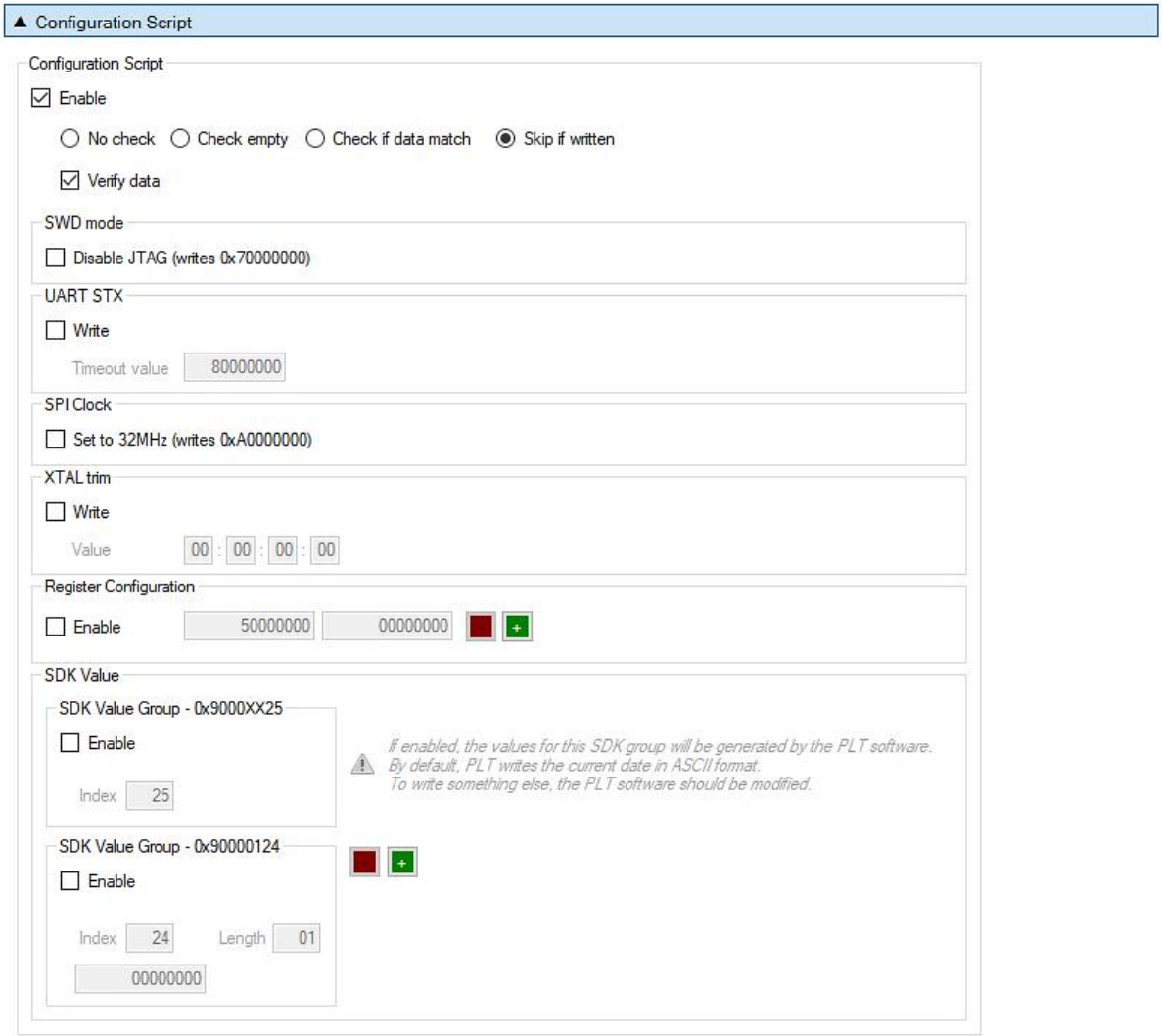



**Figure 79: OTP Configuration Script - DA14531**

Table 69 describes the *OTP Configuration Script* test options. With this option is enabled user has the option to program the Configuration Script (CS) with appropriate data.

**Table 69: OTP Configuration Script - DA14531**

| Option | Description |
|---|---|
| Enable | When selected, Configuration Script options will be enabled. |
| • No check<br>• Check empty<br>• Check if data match<br>• Skip if written | OTP memory protection options:<br>**No check:** No protection is enabled. PLT will attempt to burn the OTP memory without running any check.<br>**Check empty:** PLT will first check if the OTP memory to be burned is empty. If it is empty will burn it.<br>**Check if data match:** PLT will first check if the memory to be burned is empty. If it is not, it will compare the contents with the data to be burned. If the data are not the same the test will fail without making any changes to the memory. If the data are the same, PLT will not burn the memory to prevent using the OTP repair memory.<br>**Skip if written:** PLT will read the contents of OTP memory. If it contains any data, it will skip writing without producing any error. |


© 2020 Dialog Semiconductor

| Option | Description |
|---|---|
| Verify | When selected, the BD address will be read back from the OTP Header and will be compared to the original. |
| SWD mode | If selected JTAG will be disabled. Word 0x70000000 will be written in OTP CS. |
| UART STX | If this option is enabled user can program the STM timeout in multiple of 100ms. So, for example, value 0x80000028 is 40x100 us = 4 ms. |
| SPI Clock | If this option is enabled, it will overwrite the default 2-MHz clock speed of the SPI boot path and set it to 32 MHz |
| XTAL trim | User can manually set an XTAL trim calibration value. This value will be applied to all DUTs. It is suggested to have this disabled and use the automatic XTAL trim operation that finds the best trim value for each DUT. |
| Register Configuration | If this option is enabled, user can program in OTP CS a value that will be set to a specific register during boot. It contains:<br>• A 32-bit word containing an address of an existing register<br>• A 32-bit word containing the data value of the register<br>These are always in pairs with the address sitting in even memory addresses. |
| SDK Value Group - 0x9000YYXX | These are mainly used for device specific calibration values, used by the SDK. It contains one 32-bit word, equal to 0x9000YYXX.<br>● 9: Indicates that the following word(s) are not to be stored to registers but will be used by the SDK software.<br>● YY: Length - Indicates that YY amount of words follow<br>● XX: Index - An increasing value, used for indexing by the SW application. If YY > 1, XX will not be increased for the words that belong to the same value. |


© 2020 Dialog Semiconductor

### 7.2.9 DUT Hardware Setup (DA1469x)

#### 7.2.9.1
#### 7.2.9.1 UART Baud Rate

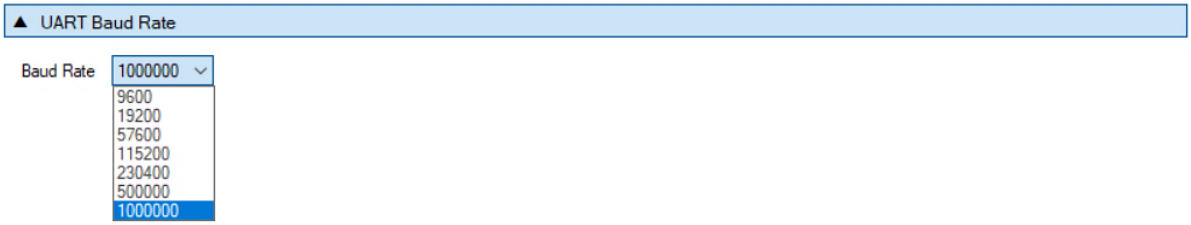



**Figure 80: UART Baud Rate - DA1469x**

Table 70 shows the available options for the DA1469x *UART Baud Rate* used during memory programming only.

The *Baud Rate* selected here is used after the firmware (`uartboot_69x.bin`) has been downloaded to the DUT. The software will send a command to the DUT to change the UART baud rate to the one selected. All following UART communications with the DUT will be performed using the new baud rate. Note that this is happening only during memory programming where `uartboot_69x.bin` is used. During tests (`RF tests, XTAL trimming`, etc.), where the production test firmware is used, the baud rate is fixed to 115200 bit/s.

**Table 70: UART Baud Rate - DA1469x**

| Option | Description |
|---|---|
| Baud Rate | • 9600 (bit/s)<br>• 19200 (bit/s)<br>• 57600 (bit/s)<br>• 115200 (bit/s)<br>• 230400 (bit/s)<br>• 1000000 (bit/s)<br>**Note:** 1 Mbit/s is the fastest and safest with 0% baud rate error. |

#### 7.2.10
#### 7.2.10 Test Settings (DA1469x)

#### 7.2.10.1 VBAT Level Log

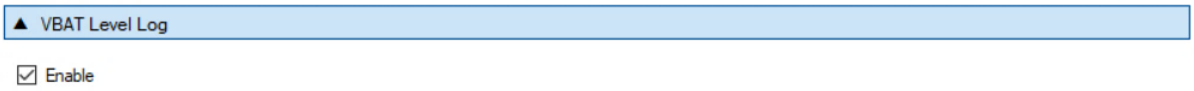



**Figure 81: VBAT Level Log - DA1469x**

When this feature is enabled, PLT will send a command to the device to measure VBAT using its internal ADC. The VBAT level will then be logged. No, pass or fail limits exist for this test. It is only used for logging purposes.

#### 7.2.10.2 OTP Timestamp Read

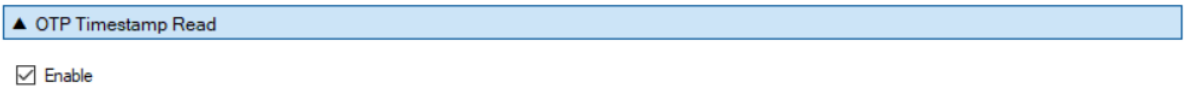



**Figure 82: OTP Timestamp Read - DA1469x**

If this option is enabled, PLT will read the device timestamp from the OTP memory and log it. This operation is mainly used for logging purposes.


© 2020 Dialog Semiconductor

#### 7.2.10.3 XTAL Trim

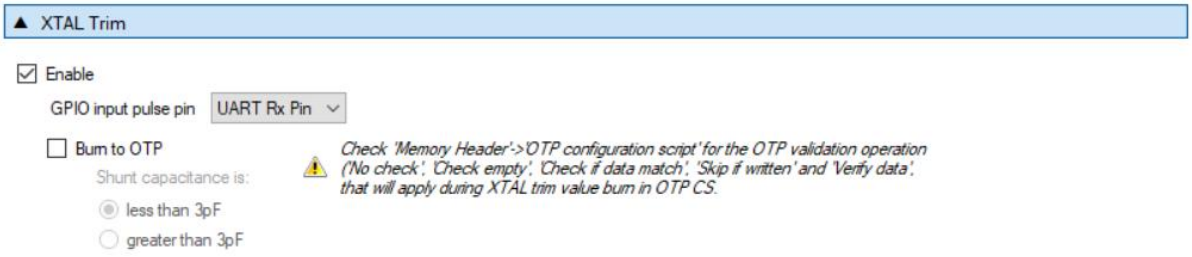



**Figure 83: XTAL Trim - DA1469x**

Table 71 describes the available options for the DA1469x *XTAL Trim* operation.

**Table 71: XTAL Trim - DA1469x**

| Option | Description |
|---|---|
| Enable | This option enables the automatic crystal oscillator frequency calibration procedure. |
| GPIO input pulse pin | The DUT GPIO to receive the reference pulse during calibration. UART RX pin can be used without any additional connection from the PLT hardware to the DUT. |
| Burn to OTP | If *Burn to OTP* option is selected, the XTAL trim value calculated from the automated calibration process will be written in the OTP XTAL trim header field. |
| Shunt capacitance is:<br>• Less than 3pF<br>• Greater than 3pF | Depending on the 32 MHz crystal oscillator C0 capacitance, additional settings will be burned to the OTP to highly improve boot time. The value of the C0 capacitance can be found from the crystal oscillator datasheet. |

#### 7.2.10.4 GPIO Watchdog operation

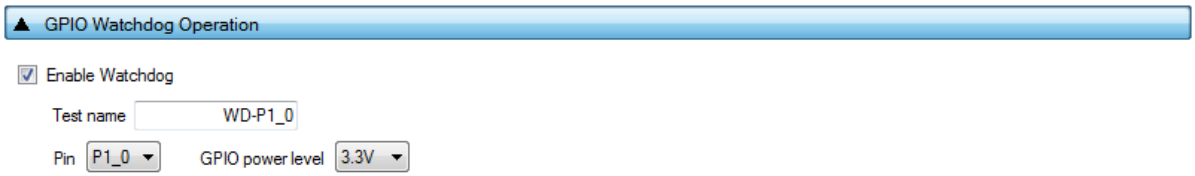



**Figure 84: GPIO Watchdog operation - DA1469x**

Table 72 describes the available options for the DA1469x *GPIO Watchdog operation*.

**Table 72: GPIO Watchdog operation - DA1469x**

| Option | Description |
|---|---|
| Enable Watchdog | This option enables the continuous toggling of a GPIO during the whole production testing and memory programming procedure, except during firmware download.<br>The pulse on the GPIO has approximately 0.75% duty cycle and 0.5Hz frequency.<br>**Note:** Production test firmware is downloaded through uartboot_69x firmware. After the uartboot_69x firmware is downloaded, the watchdog pin will be pulsed. |
| Test name | The name assigned for this test. |
| Pin | Select the GPIO that will be toggled. |
| GPIO power level | Sets the power level of the GPIOs. |

 © 2020 Dialog Semiconductor

#### 7.2.10.5 Scan DUT Advertise Test

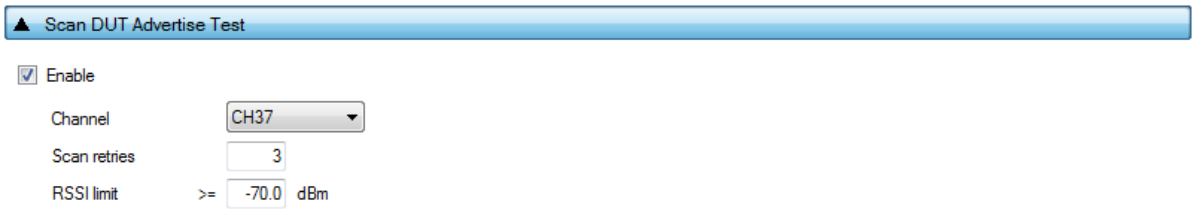



**Figure 85: Scan DUT Advertise Test - DA1469x**

Table 73 describes the available options for the DA1469x *Scan DUT Advertise Test* operation.

**Table 73: Scan DUT Advertise Test - DA1469x**

| Option | Description |
|---|---|
| Enable | This option enables the `Scan DUT Advertise Test` operation. |
| Channel | The BLE channel frequency used in the RF RX test using the Golden Unit. |
| Scan retries | The number of retries to perform the test. |
| RSSI limit | The RSSI limit for pass/fail criteria in the RF RX test using the Golden Unit. If the average RSSI of the device, after it has received the packets transmitted from the Golden Unit is less than that the test will be considered as failed. |

#### 7.2.10.6 RF Tests

This section refers to various RF tests conducted between the DUTs and the Golden Unit or an external BLE tester.

The following tests can have multiple instances with different settings. Tests can be added and removed using the two buttons (e.g. - and + in Figure 86) at the bottom right side of each panel.

**Note:** When adding or removing a test all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

##### Golden Unit

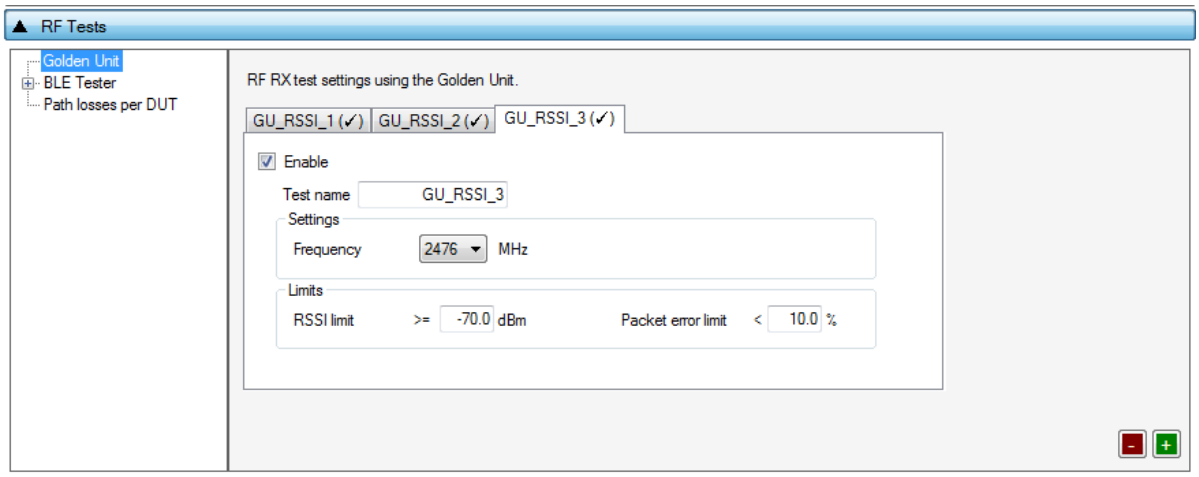



**Figure 86: Golden Unit RF Tests - DA1469x**

Table 74 describes the available options for the DA1469x *RF RX* test using the *Golden Unit* as a transmitter.

 © 2020 Dialog Semiconductor

In the RF RX test, the Golden Unit sends 500 packets. The DUTs are set in receive mode and the RSSI is measured. If the RSSI measured by the DUT reception is less than the specified *RSSI limit*, the device will fail and the tests will stop for that particular device.

**Table 74: Golden Unit RF Tests - DA1469x**

| Option | Description |
|---|---|
| Enable | This option enables the specific RF RX test using the Golden Unit as a transmitter. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Frequency | The BLE channel frequency used in the RF RX test using the Golden Unit. |
| RSSI limit | The RSSI limit for pass/fail criteria in the RF RX test using the Golden Unit. If the average RSSI of the device after it has received the packets transmitted from the Golden Unit is less than this value, the test will be considered as failed. |
| Packet error limit | This configures the PER limit for pass/fail criteria. If the percentage of the correct packets received is less than the value entered here, the test will fail. |

## BLE Tester

In the *BLE Tester* panels, a number of tests can be enabled that require an external BLE tester instrument. More detailed information about the BLE tester can be found in [1].

## BLE Tester - General

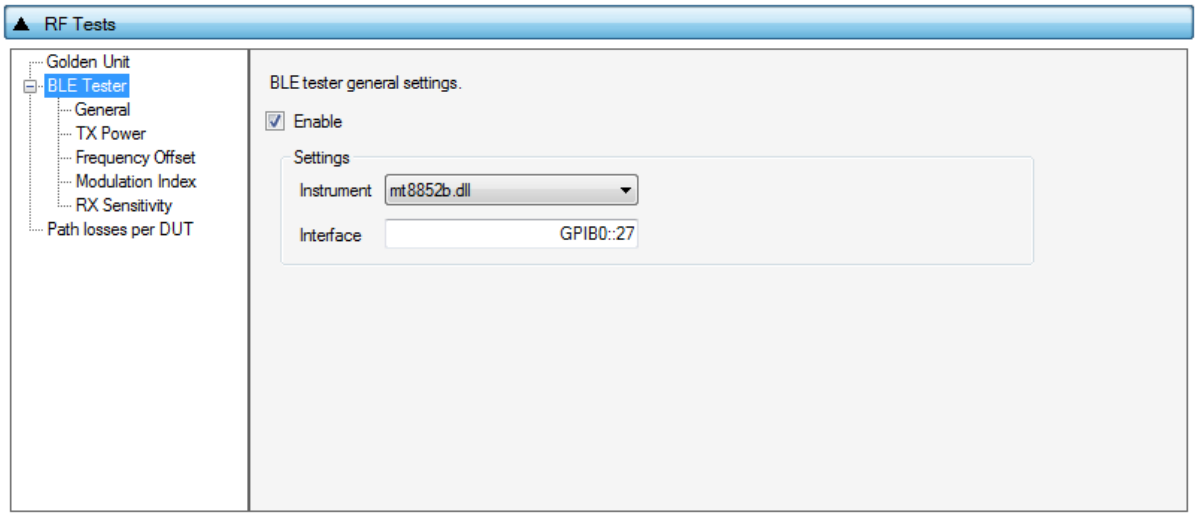



**Figure 87: BLE Tester General Settings - DA1469x**

Table 75 describes the *General* settings for the *BLE Tester* supported tests. Any available external instrument found by the `ble_tester_driver` DLL and their interfaces can be selected.

**Table 75: BLE Tester General Settings - DA1469x**

| Option | Description |
|---|---|
| Enable | This option enables all of the *BLE Tester* tests, which include:<br>● BLE Tester TX Power<br>● Frequency Offset<br>● Modulation Index<br>● RX Sensitivity |


© 2020 Dialog Semiconductor

| Option | Description |
|---|---|
| Instrument | Selects the BLE tester DLL. Names are shown only if a BLE tester instrument DLL exists in the project folder `ble_tester_instr_plugins`. |
| Interface | The interface of the instrument to be used by the driver. |

### BLE Tester - TX Power

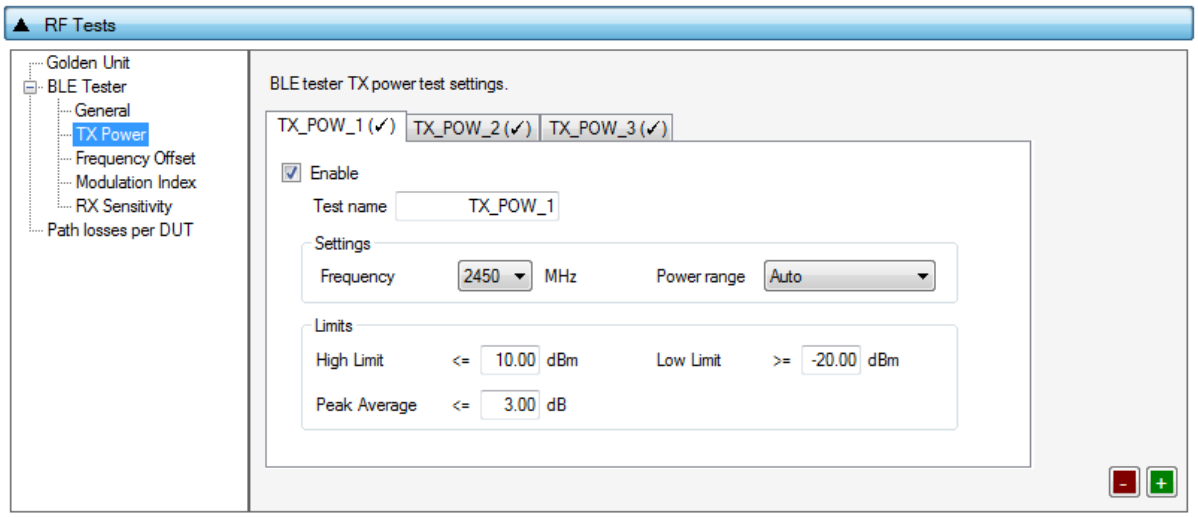



**Figure 88: BLE Tester TX Power - DA1469x**

Table 76 describes the available options for the DA1469x *TX Power* test using a BLE Tester instrument.

**Table 76: BLE Tester TX Power - DA1469x**

| Option | Description |
|---|---|
| Enable | This option enables the specific TX power test using a BLE tester instrument. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Frequency | The BLE channel frequency used in the BLE TX power test. |
| Power range | Set the device TX output power range. Available options are:<br>● Auto (No auto option for Litepoint IQxel-M. Sets the instrument to trigger at -25dBm)<br>● +22 dBm to +7 dBm<br>● +9 dBm to -3 dBm<br>● +5 dBm to -7 dBm<br>● -4 dBm to -16 dBm<br>● -12 dBm to -26 dBm<br>● -24 dBm to -35 dBm<br>Default value is *Auto.* |
| High limit | Set the average high power limit for the BLE TX output power pass/fail test criteria. |
| Low limit | Set the average low power limit for the BLE TX output power pass/fail test criteria. |
| Peak average | Set the peak-to-average power limit for the BLE TX output power pass/fail test criteria. |


© 2020 Dialog Semiconductor

### BLE Tester - Frequency Offset

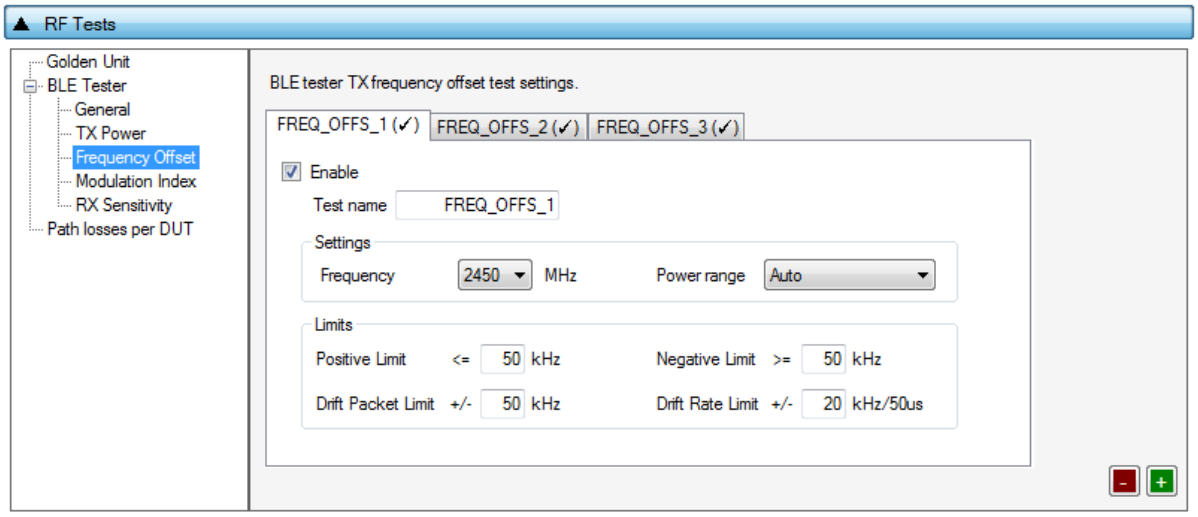



**Figure 89: BLE Tester Frequency Offset - DA1469x**

Table 77 describes the available options for the *Frequency Offset* test using a BLE Tester instrument.

**Table 77: BLE Tester Frequency Offset - DA1469x**

| Option | Description |
|---|---|
| Enable | This option enables the specific TX frequency offset test using a BLE tester instrument. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Frequency | The BLE channel frequency used in the BLE TX frequency offset test. |
| Power range | Set the device TX output power range. Available options are:<br>● Auto<br>● +22 dBm to +7 dBm<br>● +9 dBm to -3 dBm<br>● +5 dBm to -7 dBm<br>● -4 dBm to -16 dBm<br>● -12 dBm to -26 dBm<br>● -24 dBm to -35 dBm<br>Default value is *Auto.* |
| Positive limit | Set the maximum positive offset limit in kHz for the TX carrier frequency offset pass/fail test criteria. |
| Negative limit | Set the maximum negative offset limit in kHz for the TX carrier frequency offset pass/fail test criteria. |
| Drift packet limit | Set the overall packet drift in kHz for the TX drift pass/fail test criteria. |
| Drift rate limit | Set the drift rate limit in kHz/50 μs for the TX drift pass/fail test criteria. |

 © 2020 Dialog Semiconductor

### BLE Tester - Modulation Index

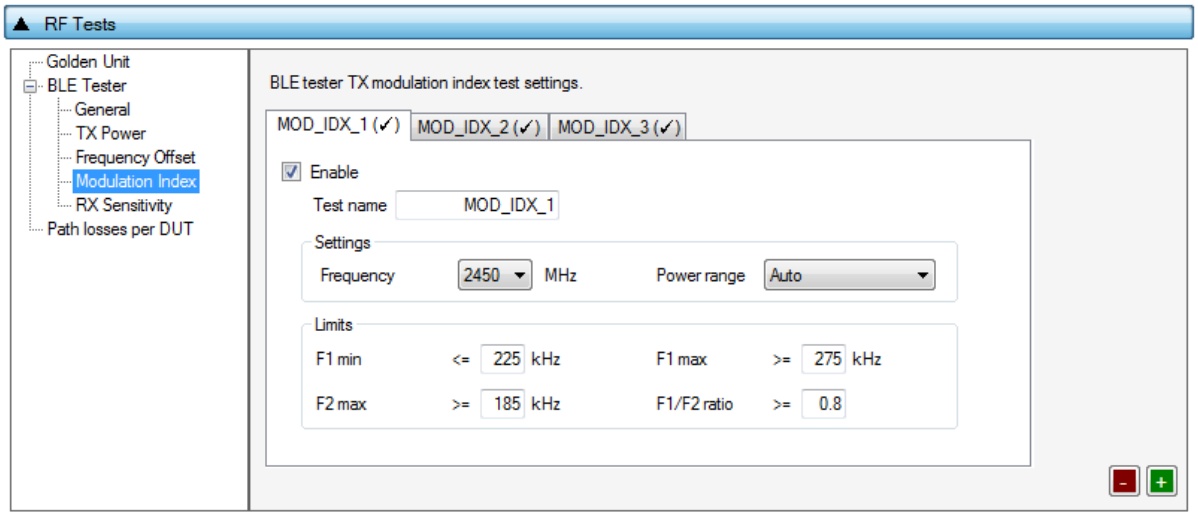



**Figure 90: BLE Tester Modulation Index - DA1469x**

Table 78 describes the available options for the *Modulation Index* test using a BLE Tester instrument.

**Table 78: BLE Tester Modulation Index - DA1469x**

| Option | Description |
|---|---|
| Enable | This option enables the specific TX modulation index test using a BLE tester instrument. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Frequency | The BLE channel frequency used in the BLE TX modulation index offset test. |
| Power range | Set the device TX output power range. Available options are:<br>● Auto<br>● +22 dBm to +7 dBm<br>● +9 dBm to -3 dBm<br>● +5 dBm to -7 dBm<br>● -4 dBm to -16 dBm<br>● -12 dBm to -26 dBm<br>● -24 dBm to -35 dBm<br>Default value is *Auto.* |
| F1 min | Set the F1 minimum average limit in kHz for the TX modulation index pass/fail test criteria. |
| F1 max | Set the F1 maximum average limit in kHz for the TX modulation index pass/fail test criteria. |
| F2 max | Set the F2 maximum limit in kHz for the TX modulation index pass/fail test criteria. |
| F1/F2 ratio | Set the F1/F2 maximum average ratio limit for the TX modulation index pass/fail test criteria. |


© 2020 Dialog Semiconductor

### BLE Tester - RX Sensitivity

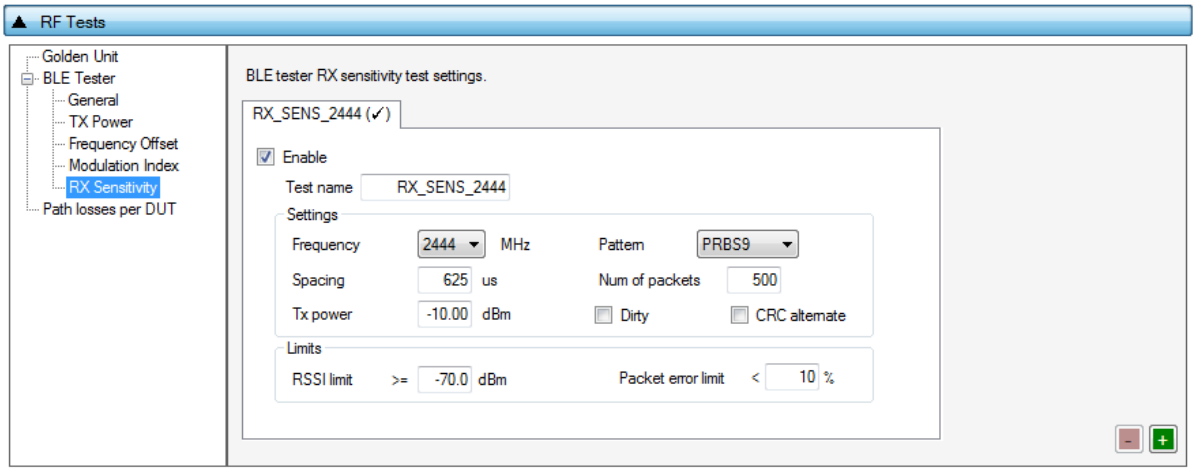



**Figure 91: BLE Tester RX Sensitivity - DA1469x**

Table 79 describes the available options for the *RX Sensitivity* test using a BLE Tester instrument.

**Table 79: BLE Tester RX Sensitivity - DA1469x**

| Option | Description |
|---|---|
| Enable | This option enables the specific RX sensitivity test using a BLE tester instrument. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Frequency | The BLE channel frequency used in the BLE RX sensitivity test. |
| Pattern | The bit pattern of the TX data. Available options are:<br>● PRBS9<br>● 10101010<br>● 11110000 |
| Spacing | The packet spacing in µs. |
| Num of packets | The number of packets the BLE tester instrument to transmit. |
| Tx power | The TX output power of the BLE tester instrument. Suggested values are 0 to -10 dBm. |
| Dirty | When enabled, the BLE tester packet generator can use a dirty table to transmit. |
| CRC alternate | When enabled, the BLE tester will alternatingly send packets with CRC correct and CRC incorrect. |
| RSSI limit | The RSSI limit for pass/fail criteria in the RF RX sensitivity test. If the average RSSI of the device after it has received the transmitted packets is less than this value, the test will be considered as failed. |
| Packet error limit | This configures the PER limit for pass/fail criteria. If the percentage of the correct packets received is less than the value entered here, the test will fail. |


© 2020 Dialog Semiconductor

### Path Losses per DUT

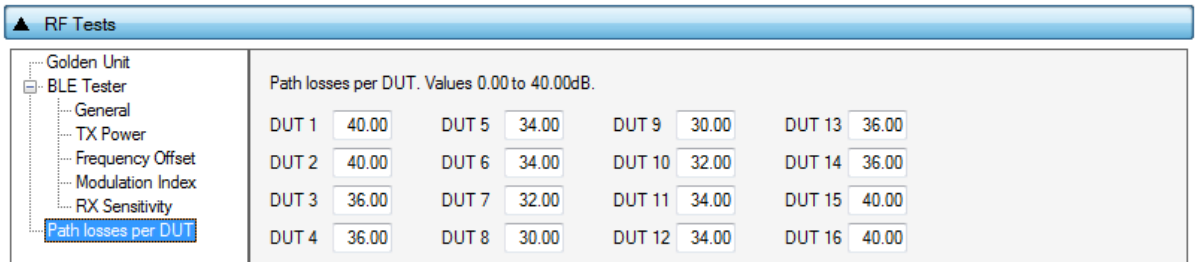



**Figure 92: Path Losses per DUT - DA1469x**

Table 80 describes the available options for the *Path losses per DUT.*

Based on the relative position of each DUT during RF tests and since the RF tests are performed over the air, values can be used to correct for any path losses. These values are added to the limits of the TX Power and RF RX RSSI tests. Additional information can be found in Appendix D and Appendix F.

**Table 80: Path Losses per DUT from RF Tests DA1469x Options**

| Option | Description |
|---|---|
| DUT1-16 | Set the path loss value for each DUT. These will be added as corrections to the limits of the TX Power and RF RX RSSI tests |

### 7.2.10.7 GPIO/LED Test

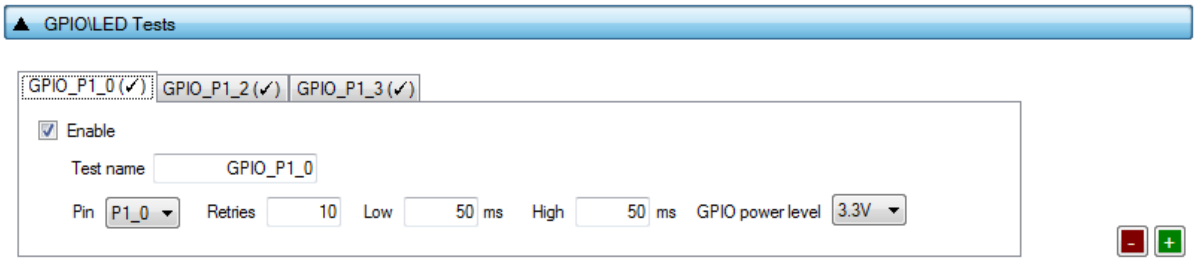



**Figure 93: GPIO/LED Tests - DA1469x**

*GPIO/LED Tests* can have multiple instances with different settings. Tests can be added or removed using the two buttons (e.g. - and + in Figure 93) at the bottom right side of each panel.

**Note:** When adding or removing a test, all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

Table 81 describes the available options for the *GPIO/LED Tests* DA1469x Options.

In these tests, a specific pulse can be given to a GPIO and any LED connected to it can be visually tested. The *Pin* option sets the GPIO to be used, *Low* and *High* define the duty cycle and the *Retries* the number of pulses.

**Table 81: GPIO/LED Tests - DA1469x**

| Option | Description |
|---|---|
| Enable | This option enables the GPIO/LED toggling. Can be used for visual LED testing. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Pin | The GPIO that will be used for the specific test. |
| Retries | Number of pulses to be generated for the specific test. |


© 2020 Dialog Semiconductor

| Option | Description |
|---|---|
| Low | Sets the amount of the OFF time of the pulse in ms for the specific test. |
| High | Sets the amount of the ON time of the pulse in ms for the specific test. |
| GPIO power level | Sets the power level of the GPIOs. |

#### 7.2.10.8 GPIO Connection Test

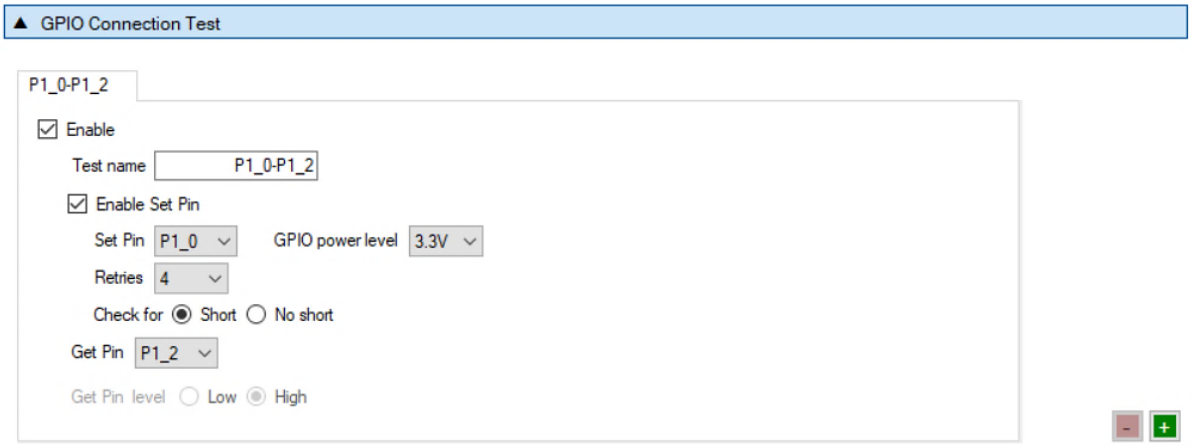



**Figure 94: GPIO Connection Test - DA1469x**

*GPIO Connection Tests* can have multiple instances with different settings. Tests can be added and removed using the two buttons (e.g. - and + in Figure 63) at the bottom right side of each panel.

**Note:** When adding or removing a test all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

Table 82 describes the available options for the DA1469x *GPIO Connection Test*.

When enabled, the PLT software will check the connection of the specified GPIO (`Get Pin`) by either checking its state or the connection with another pin (`Set Pin`). In the latter case, the user gives the `Set Pin` and the state to check. It will also check for shorts between given GPIOs.

**Table 82: GPIO Connection Test - DA1469x**

| Option | Description |
|---|---|
| Enable | This option enables the specific custom test. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Enable Set Pin | Enables the use of the secondary GPIO to drive the GPIO under test. When this option is set, the `Get Pin level` option will be disabled. |
| Set Pin | Select the GPIO to be tested |
| GPIO power level | The output GPIO power level, 3.3V or 1.8V. |
| Retries | How many times the software will check for GPIO connection or short. In every retry it will change the `Set Pin` level and check the `Get Pin` level. |
| Check for Short/No short | If Short is selected, PLT will check whether the `Set Pin` has the same level with the `Get Pin` for all `Retries` tested. If `No short` is selected, PLT will check whether `Get Pin` is always low no matter what the `Set Pin` level is. |
| Get Pin | Select the GPIO to be tested. |
| Get Pin level | Sets the GPIO state the test awaits to see in the `Get Pin`. This option is disabled if the Set Pin mode is enabled. |


© 2020 Dialog Semiconductor

### 7.2.10.9 Sensor Test

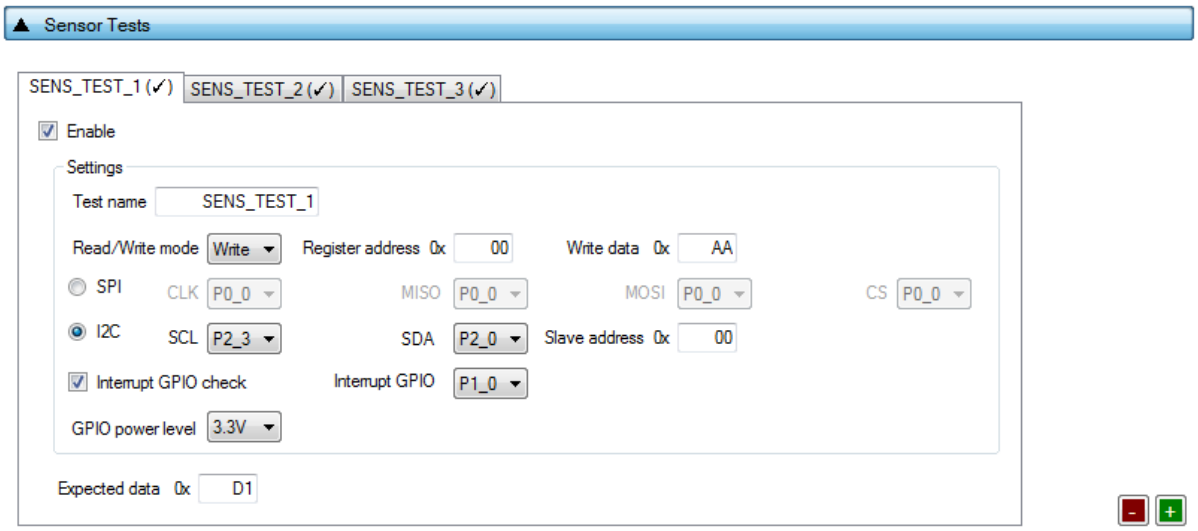



**Figure 95: Sensor Test - DA1469x**

*Sensor Tests* can have multiple instances with different settings. Tests can be added and removed using the two buttons (e.g. and in Figure 95) at the bottom right side of each panel.

**Note:** When adding or removing a test all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

Table 83 describes the available options for the *Sensor Tests* DA1469x Options.

**Table 83: Sensor Tests - DA1469x**

| Option | Description |
|---|---|
| Enable | This option enables the specific sensor test. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Read / Write mode | Select the sensor test procedure, to read or write. |
| Register address | The sensors register address to read or write data. |
| Write data | The byte to be written at the sensor register. |
| SPI / I2C | Select the interface that the sensor is connected to. |
| SPI - CLK | Select the GPIO for the sensor SPI CLK. |
| SPI - MISO | Select the GPIO for the sensor SPI MISO. |
| SPI - MOSI | Select the GPIO for the sensor SPI MOSI. |
| SPI - CS | Select the GPIO for the sensor SPI CS. |
| I2C - SCL | Select the GPIO for the sensor I2C SCL. |
| I2C - SDA | Select the GPIO for the sensor I2C SDA. |
| Slave address | The sensor I2C bus slave address. |
| Interrupt GPIO check | Enables the sensor interrupt signal test via GPIO. |
| Interrupt GPIO | Select the GPIO to be used as a sensor interrupt. |
| GPIO power level | Sets the power level of the GPIOs. |
| Expected data | The received sensor byte that will be expected on a successful operation. |


© 2020 Dialog Semiconductor

#### 7.2.10.10 Custom Test

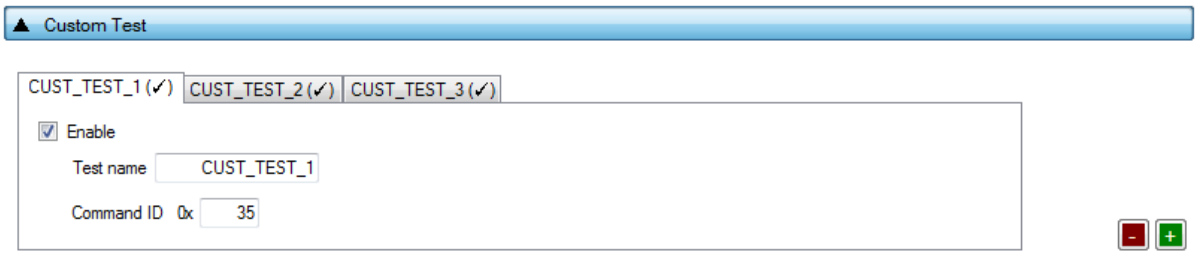



**Figure 96: Custom Test - DA1469x**

*Custom tests* can have multiple instances with different settings. Tests can be added and removed using the two buttons (e.g. - and + in Figure 96) at the bottom right side of each panel.

**Note:** When adding or removing a test all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

Table 84 describes the available options for the DA1469x *Custom Tests*.

When enabled, the PLT software will send an HCI command through UART to activate a customer-defined test that will run on the DUTs. The HCI custom test command will contain a single byte as data (the *Command ID* byte), to be used mainly as identification for a specific test in the firmware. Default functionality of the production test firmware is to respond with the same `Command ID`. Otherwise, the test will be considered as failed.

**Table 84: Custom Tests DA1469x Options**

| Option | Description |
|---|---|
| Enable | This option enables the specific custom test. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Command ID | The byte that will be sent to the device running the production test firmware. |

#### 7.2.10.11 External 32kHz Test

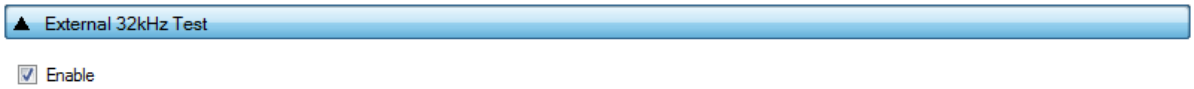



**Figure 97: External 32kHz Test - DA1469x**

Table 85 describes the available options for the DA1469x External 32kHz *Tests*.

**Table 85: External 32kHz Tests DA1469x Options**

| Option | Description |
|---|---|
| Enable | This option enables the external 32kHz low power clock test. |

 © 2020 Dialog Semiconductor

### 7.2.10.12 Current Measurement Test

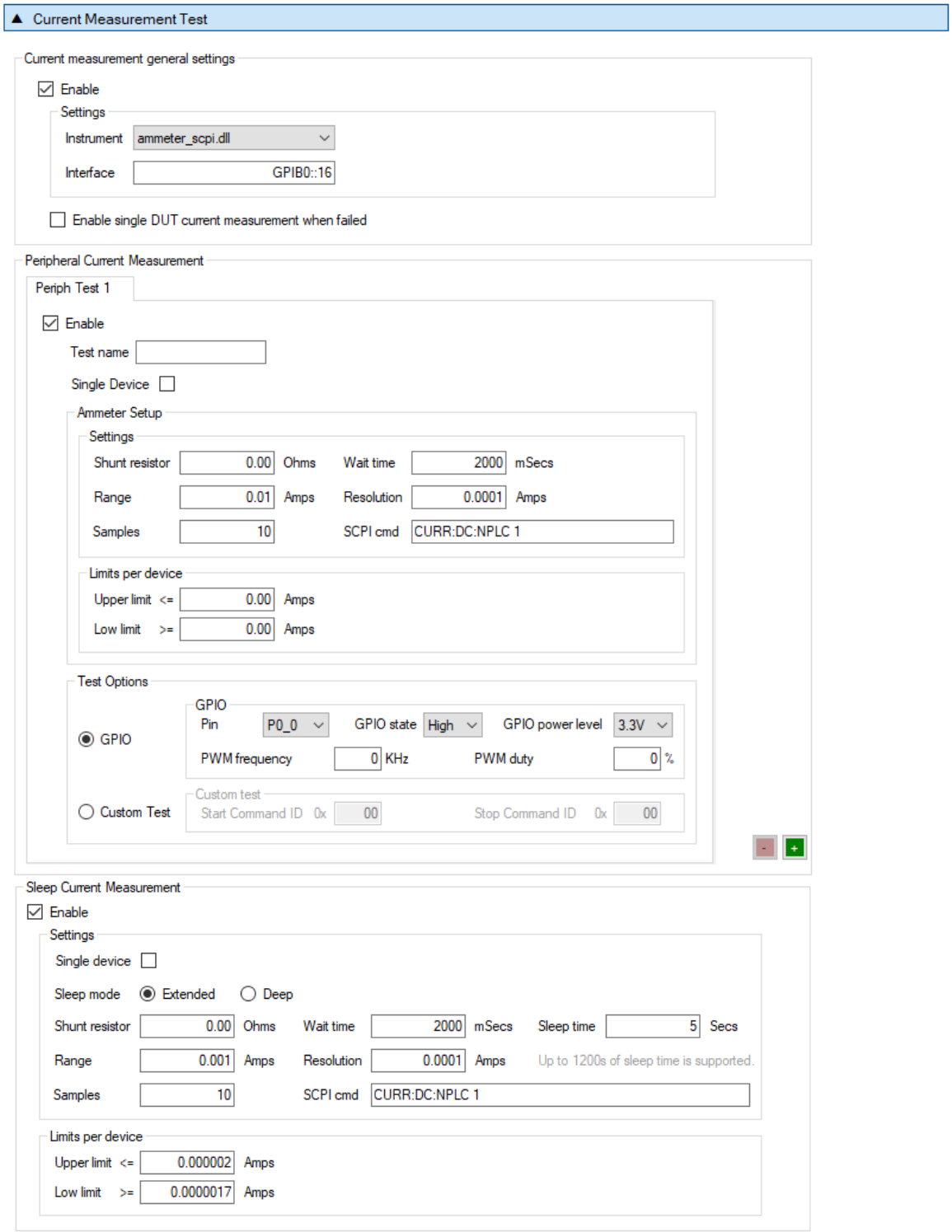



**Figure 98: Current Measurement Tests - DA1469x**

In this test, an external ammeter instrument can be used to measure the total current consumption of all active DUTs. The ammeter instrument can be connected in the blue banana plugs as described in Current Measurements or to an external power supply (if present) depending the selected VBAT/Reset Mode.


© 2020 Dialog Semiconductor

During measurement, PLT will control the instrument using the `ammeter_driver` DLL [1]. Table 86 describes the instrument selection settings found by the `ammeter_driver` DLL, Table 87 describes the settings used for each of the peripheral current measurement tests and Table 88 describes the current measurement options for each sleep state.

**Note:** Modifications in the production test firmware are mandatory in order to achieve the correct current consumption of a specific hardware design (IC and peripherals) for each sleep state. Running the default firmware without any modifications specific for the hardware design, may cause increased current consumption.

*Peripheral Current Measurement Tests* can have multiple instances with different settings. Tests can be added and removed using the two buttons (e.g. - and + in Figure 98) at the bottom right side of each panel.

**Note:** When adding or removing a test, all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

**Table 86: Current Measurement Tests - DA1469x**

| Option | Description |
|---|---|
| Enable | This option enables all Current Measurement tests, which include:<br>● Peripheral current measurements<br>● Extended sleep current measurement<br>● Deep sleep current measurement<br>Only one of the Extended/Deep sleep current measurements can be selected. |
| Instrument | Select the Ammeter instrument DLL name. Names are shown only if an ammeter instrument DLL exists in the project `ammeter_instr_plugins` folder. |
| Interface | The interface of the instrument to be used by the driver. |
| Enable single DUT current measurement when failed | If this option is enabled and if the measurement taken is outside of the limits, PTL will reset all devices and begin a firmware download and measure the current to each device separately, in order to identify which exact device failed. |

**Table 87: Current Measurement Test – Peripheral Current Measurement - DA1469x**

| Option | Description |
|---|---|
| Enable | This option enables the specific peripheral current measurement test. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Single Device | If this option is enabled, PLT will measure the current consumption one device at the time. Initially, it will power off all DUTs. It will then power on one by one, download firmware and measure the current individually. This procedure takes a lot of time. It should only be used for production setup purposes, to identify the correct limits by measuring multiple DUTs and taking the average. |
| Shunt resistor | The value of the shunt resistor used for peripheral measurement. Applicable only if a voltmeter or a NI-DAQ is used to measure current instead of a DMM. Values from 0 to 9999 are supported. |
| Wait time | The time in ms the PLT will wait before taking a current measurement, after it has sent an instruction to the DUTs to go into a specific sleep state. Supported values are 1 to 500000ms. |
| Range | The range value in Ampere units that the ammeter instrument will operate. Supported values are 0 to 9999 and default value is 0.001A. When the `ammeter_scpi.dll` is used, if this value is set to zero, the instrument will use the automatic range functionality. |
| Resolution | The ammeter instrument resolution value in Ampere units. |
| Samples | The number of samples that the ammeter instrument will read and average. 1 to 1000 is supported. |

 © 2020 Dialog Semiconductor

| Option | Description | |
|---|---|---|
| SCPI cmd | An SCPI command to be passed to the ammeter instrument just before the measurement is taken. Supports multiple commands separated with a column. Up to 256 characters are supported. | |
| Upper limit | The upper limit value of the peripheral current measurement test procedure, for a single DUT.<br>Next to this input field the total upper limit current consumption for all the enabled DUTs is shown, which is the value to be used during testing.<br>**Note:** During testing, some DUTs may fail until the current measurement test takes place. In that case, the PLT will automatically re-calculate the total upper limit by using the value given for a single DUT multiplied with the number of the active DUTs at the time that the test will run. | |
| Low limit | The low limit value of the peripheral current measurement test, for a single DUT.<br>Next to this input field the total upper limit current consumption for all the enabled DUTs is shown, which is the value to be used during testing.<br>**Note:** During testing, some DUTs may fail until the current measurement test takes place. In that case, the PLT will automatically re-calculate the total low limit by using the value given for a single DUT multiplied with the number of the active DUTs at the time that the test will run. | |
| Test Options | Select between a PWM GPIO test and custom test.<br>**Note:** For the custom tests to work, a modified production test firmware must be created with tests that set the DUTs to specific states before the current measurement test. Each test must be assigned to a specific opcode. The custom tests are the exact same as in Custom Test. | |
| Test Options - GPIO | Pin | Sets the GPIO to toggle with the PWM pulse. |
| | GPIO state | Sets the active state of the GPIO. |
| | GPIO power level | Sets the power level of the GPIOs. |
| | PWM frequency | Sets the PWM frequency. |
| | PWM duty | Sets the PWM duty cycle. |
| Test Options – Custom Test | Start Command ID | The opcode of the custom test that sets the state of the DUT. |
| | Stop Command ID | The opcode of the custom test that restores the DUT to its original state. |

**Table 88: Current Measurement Test - Sleep Current Measurement - DA1469x**

| Option | Description |
|---|---|
| Enable | This option enables the sleep current measurement using the ammeter instrument provided in the *Instrument* section. |
| Single Device | If this option is enabled, PLT will measure the current consumption, one device at the time. Initially, it will power off all DUTs. It will then power on one by one, download firmware and measure the current individually. This procedure takes a lot of time. It should only be used for production setup purposes, to identify the correct limits by measuring multiple DUTs and taking the average. |
| Sleep mode | User can select either the Extended or the Deep sleep mode. |
| Shunt resistor | The value of the shunt resistor used for sleep current measurement. Applicable only if a voltmeter or a NI-DAQ is used to measure current instead of a DMM. Values from 0 to 9999 are supported. |
| Wait time | The time in ms the PLT will wait before taking a current measurement, after it has sent an instruction to the DUTs to go into a specific sleep state. Supported values are 1 to 500000ms. |
| Sleep time (only for extended sleep) | The time in seconds that the DUTs will remain in extended sleep mode. A timer in the production test firmware will wake up the devices and set them to idle mode. Supported values are 1 to 9sec for DA14580/1/2/3 and up to 1200sec for the rest. |

 © 2020 Dialog Semiconductor

| Option | Description |
|---|---|
| Range | The range value in Ampere units that the ammeter instrument will operate. Supported values are 0 to 9999 and default value is 0.001A. When the `ammeter_scpi.dll` is used, if this value is set to zero, the instrument will use the automatic range functionality. |
| Resolution | The ammeter instrument resolution value in Ampere units. |
| Samples | The number of samples that the ammeter instrument will read and average. 1 to 1000 is supported. |
| SCPI cmd | An SCPI command to be passed to the ammeter instrument just before the measurement is taken. Supports multiple commands separated with a column. Up to 256 characters are supported. |
| Upper limit | The upper limit value for the sleep current measurement test procedure, for a single DUT.<br>Next to this input field the total upper limit current consumption for all the enabled DUTs is shown, which is the value to be used during testing.<br>**Note:** During testing, some DUTs may fail until the current measurement test takes place; in that case, the PLT will automatically re-calculate the total upper limit by using the value given for a single DUT multiplied with the number of the active DUTs at the time that the test will run. |
| Low limit | The low limit value for the sleep current measurement test procedure, for a single DUT.<br>Next to this input field the total upper limit current consumption for all the enabled DUTs is shown, which is the value to be used during testing.<br>**Note:** During testing, some DUTs may fail until the current measurement test takes place; in that case, the PLT will automatically re-calculate the total low limit by using the value given for a single DUT multiplied with the number of the active DUTs at the time that the test will run. |

#### 7.2.10.13 Temperature Measurement Test

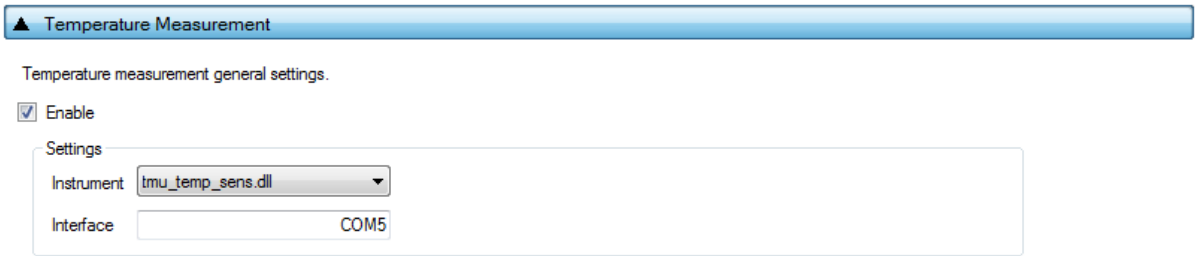



**Figure 99: Temperature Measurement Test - DA1469x**

Table 89 describes the available options for the DA1469x *Temperature Measurement Test*.

**Table 89: Temperature Measurement Test - DA1469x**

| Option | Description |
|---|---|
| Enable | This option enables the Temperature measurement test. |
| Instrument | Select the Temperature measurement DLL. Names are shown only if a Temperature measurement instrument DLL exists in the project folder `temp_meas_instr_plugins`. |
| Interface | The interface of the instrument to be used by the driver. |


© 2020 Dialog Semiconductor

#### 7.2.10.14 Scan Test

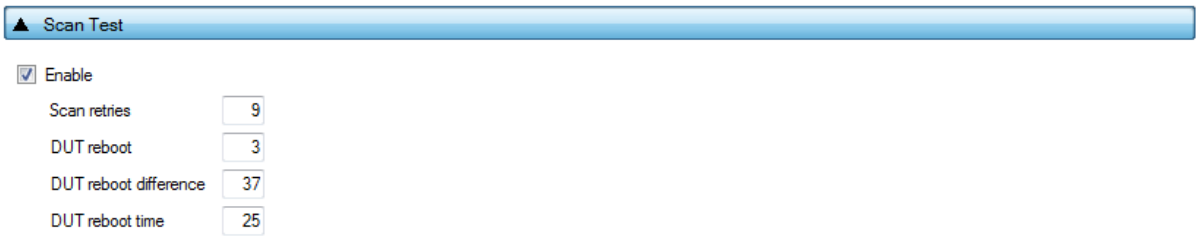



**Figure 100: Scan Test - DA1469x**

Table 90 describes the available options for the DA1469x *Scan Test*.

By enabling this test, the Golden Unit will scan for the DUT's BD addresses advertised after the customer firmware has been burned. For this test to work, a bootable firmware with the ability to advertise with the BD address given by the PLT must be burned into each DUT. Additionally, the BD addresses provided by the PLT should be burned into OTP memory or QSPI memory such that the devices advertise with the BD addresses the tool uses.

**Table 90: Scan Test DA1469x Options**

| Option | Description |
|---|---|
| Enable | This option enables the Scan test. |
| Scan retries | The total number of BLE advertising scans the Golden Unit will perform. |
| DUT reboot | Define after how many retries the PLT will reboot the DUTs. |
| DUT reboot difference | Set the time difference between each DUT when the PLT reboots the devices, in order to avoid air collisions. |
| DUT reboot time | The time the VBAT will remain low during the device reboot. This value is time in ms*100 (e.g. 15 is 1500ms). |

 © 2020 Dialog Semiconductor

### 7.2.11 Memory Functions (DA1469x)

This section describes the Memory Functions settings available when using DA1469x devices. Memory functions include OTP and QSPI Flash memory programming.

#### 7.2.11.1 OTP Memory

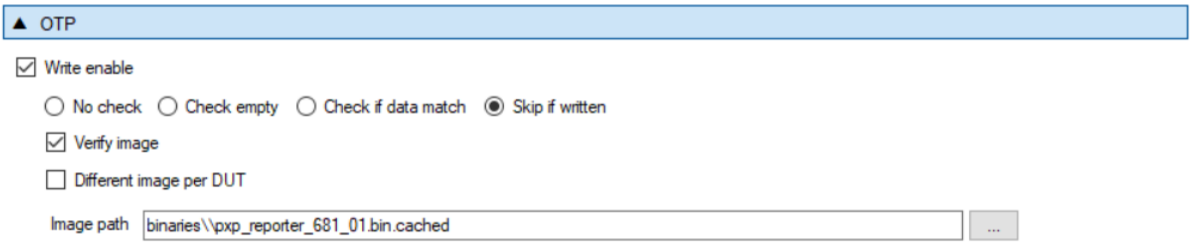



**Figure 101: OTP Memory - DA1469x**

This test enables the OTP memory programming. Table 91 describes the available options for the *OTP Memory* image write operation.

**Note:** If the binary is larger than the available OTP image area (OTP memory excluding the header area), the PLT software will split the binary into two parts. The first part will contain only the OTP image area. The second part will contain the OTP header fields, split in OTP words. PLT will burn the non-zero words one by one, as single OTP entries in the OTP header area. The check empty feature will handle the first part as an OTP image binary. The second part will be checked word by word.

**Table 91: OTP Memory - DA1469x**

| Option | Description |
|---|---|
| Write enable | This option enables the OTP image write operation. |
| • No check<br>• Check empty<br>• Check if data match<br>• Skip if written | Memory protection options:<br>**No check:** No protection is enabled. PLT will attempt to burn the OTP memory without running any check.<br>**Check empty:** PLT will first check if the OTP memory to be burned is empty. If it is empty will burn it.<br>**Check if data match:** PLT will first check if the memory to be burned is empty. If it is not, it will compare the contents with the data to be burned. If the data are not the same the test will fail without making any changes to the memory. If the data are the same, PLT will not burn the memory to prevent using the OTP repair memory.<br>**Skip if written:** PLT will read the contents of OTP memory. If it contains any data, it will skip writing without producing any error. |
| Verify image | If this option is enabled, PLT will read back the contents of the OTP memory and compare them to the original image file. If these do not match it will fail. |
| Different image per DUT | If this option is selected, a different image per DUT can be burned into the OTP. The image name must be specific for each DUT, as described below. |
| Image path | Via this field, the user specifies the image file to be burned into the OTP. A `.bin` binary file of any name can be selected.<br>Depending on the size of the selected binary, PLT will inform the user if the binary contains both the image and the header part or if it exceeds the maximum supported size.<br>If option *Different image per DUT* is selected, the user only selects the directory of the images. In that case, the binary file names must have the following format: `img_0X.bin`, where 'X' denotes the DUT number. For example, if the user has activated DUTs 1, 5 and 10 then `img_01.img`, `img_05.img` and `img_10.img` binary files should exist in the selected OTP image path, as shown in Figure 71. |

 © 2020 Dialog Semiconductor

### 7.2.11.2 QSPI Flash Memory

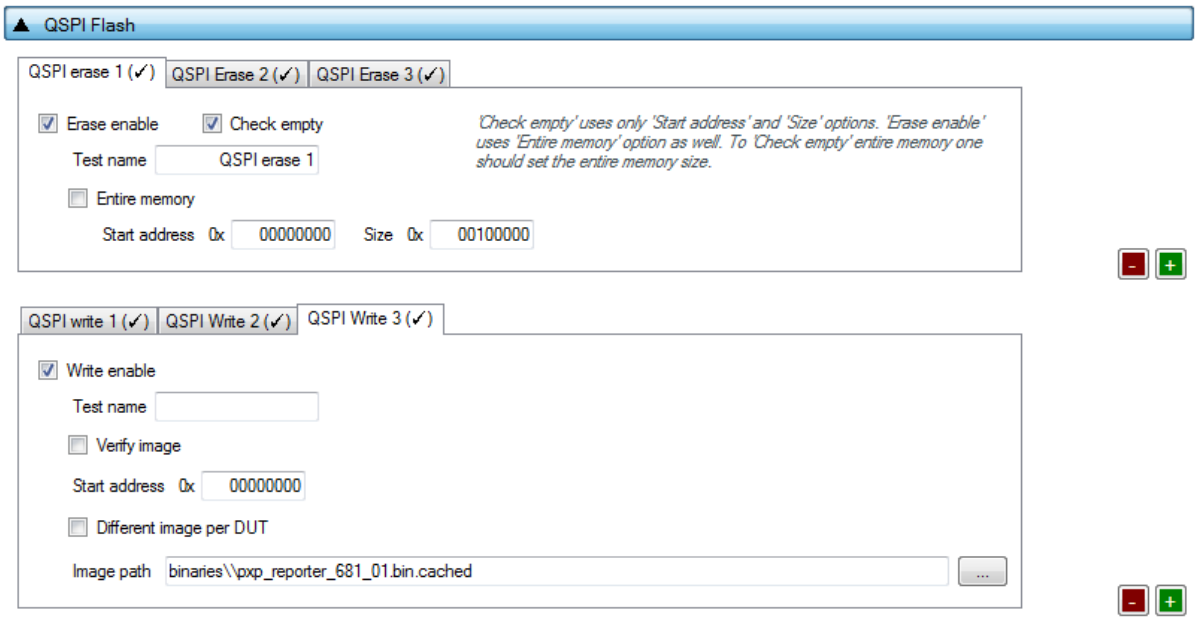



**Figure 102: QSPI Flash - DA1469x**

This section describes how the QSPI Flash memory can be erased and programmed.

Both erase and write operations can have multiple instances with different settings. Tests can be added and removed using the two buttons (e.g. - and + in Figure 102) at the bottom right side of each panel.

**Note:** When adding or removing a test all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

The QSPI flash memory should be erased before any image is written to it. Table 92 describes the available options for the *QSPI Flash Erase* operation.

**Table 92: QSPI Flash Erase - DA1469x**

| Option | Description |
|---|---|
| Erase enable | This will enable the specific QSPI flash erase test. |
| Check empty | After QSPI flash erasure, the PLT software can verify the result by sending a specific command to the `uartboot_69x.bin` firmware running in the DUT. The firmware will read the QSPI flash and check if it is empty. The result will be returned to the PLT software. |
| Entire memory | This option is only available for the *Erase enable* option. When this checkbox is selected, the entire memory can be erased. Otherwise, the user can give a start address and a specific number of bytes to be erased. |
| Start address | The user can enter a specific start address for the QSPI erasure to start. |
| Size | The size in bytes to erase, starting from the *Start address* as explained above. |

After every *QSPI Flash erase* test has finished, the *QSPI image write* tests will begin. Table 93 describes the available options for the *QSPI Flash Image Write* operation.

**Table 93: QSPI Flash Image Write - DA1469x**

| Option | Description |
|---|---|
| Write enable | This will enable the specific QSPI flash image programming test. |

 © 2020 Dialog Semiconductor

| Option | Description |
|---|---|
| Verify image | By selecting this option, the PLT software will read back the contents of the QSPI flash memory and compare them to the original image file. If these do not match, the QSPI memory programming will fail. |
| Start address | The user can configure the QSPI flash start address where the image will be written. |
| Different image per DUT | If this option is selected, ten a different image per DUT can be burned into the QSPI flash. The image name must be specific for each DUT, as described below. |
| Image path | Via this field, the user specifies the image file to be burned into the QSPI flash memory. A `.bin` binary file of any name can be selected.<br>If option *Different image per DUT* is selected, the user only specifies the directory of the images. In that case, the binary file names must have the following format: `img_0X.bin`, where 'X' denotes the DUT number. For example, if the user has activated DUTs 1, 5 and 10 then `img_01.img`, `img_05.img` and `img_10.img` binary files should exist in the selected QSPI image path, as shown in Figure 71. |

### 7.2.11.3 Memory read

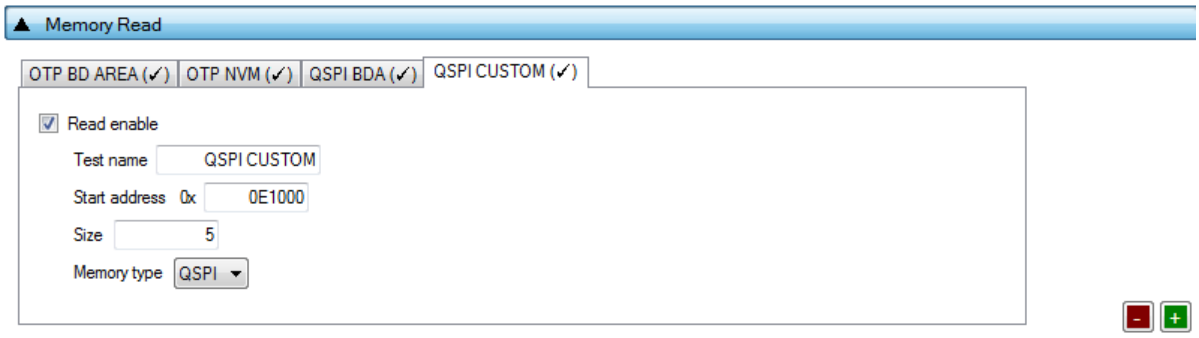



**Figure 103: Memory Read Test - DA1469x**

*Memory Read Tests* can have multiple instances with different settings. Tests can be added or removed using the two buttons (e.g. - and + in Figure 103) at the bottom right side of each panel.

**Note:** When adding or removing a test, all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

Table 94 describes the memory read test options. With this test, the user can read up to 64MBytes of data from any address from any available memory for the DA1469x devices, such as OTP and QSPI. An example of how the data appears on the log file is shown in Figure 104.

```
Memory read operation initialized. Memory read test name=[QSPI CUSTOM].
Memory read operation started. Memory read test name=[QSPI CUSTOM].
Memory read operation ended OK. Test name [QSPI CUSTOM]. Memory=[QSPI]. Addr=[0xE1000]. Size=[5]. Data=[1122334455].
```

**Figure 104: Memory Read Test Example Log File - DA1469x**

**Table 94: Memory Read Test - DA1469x**

| Option | Description |
|---|---|
| Read enable | This will enable the specific memory reading test. |
| Test name | The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not. |
| Start address | Configures the start address . Valid addresses for OTP are 0x10080000 – 0x1008FFFF. |
| Size | Number of bytes to read, up to 64MBytes. If data to be read are greater than 256 bytes, then a file will be created to store the data under folder `mem_read_test` in the PLT execution path. |
| Memory type | The type of memory to read the data from. Available options are OTP and QSPI FLASH. |


© 2020 Dialog Semiconductor

### 7.2.12 Memory Header (DA1469x)

This section describes the Memory Header programming settings (OTP and QSPI), available in DA1469x devices.

#### 7.2.12.1
7.2.12.1 QSPI Header - BD Address

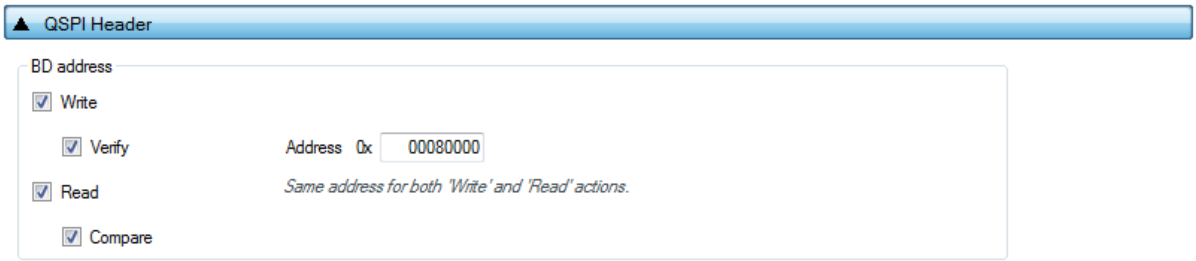



**Figure 105: QSPI Header BD Address - DA1469x**

Table 95 describes the available options for the DA1469x *BD Address* programming operation into the QSPI Header.

**Table 95: QSPI Header BD Address - DA1469x**

| Option | Description |
|---|---|
| Write | When selected the BD address will be written in the QSPI Header. |
| Verify | If selected, the BD address will be read back from the QSPI Header and compared to the original. |
| Read | This option will read the BD address written in the QSPI Header field. It does not depend on the previous tests to run, but it is necessary for the following *Compare* test. |
| Compare | When the *Read* option is enabled, a comparison will be performed between the read BD address and the BD address entered in the DUT by the PLT, as described in the BD address DUT assignment method. |
| Address | The QSPI Flash address where the BD address will be written. This field is the same for all of the above actions. Default value is 0x080000. |

#### 7.2.12.2
7.2.12.2 Custom Memory Data

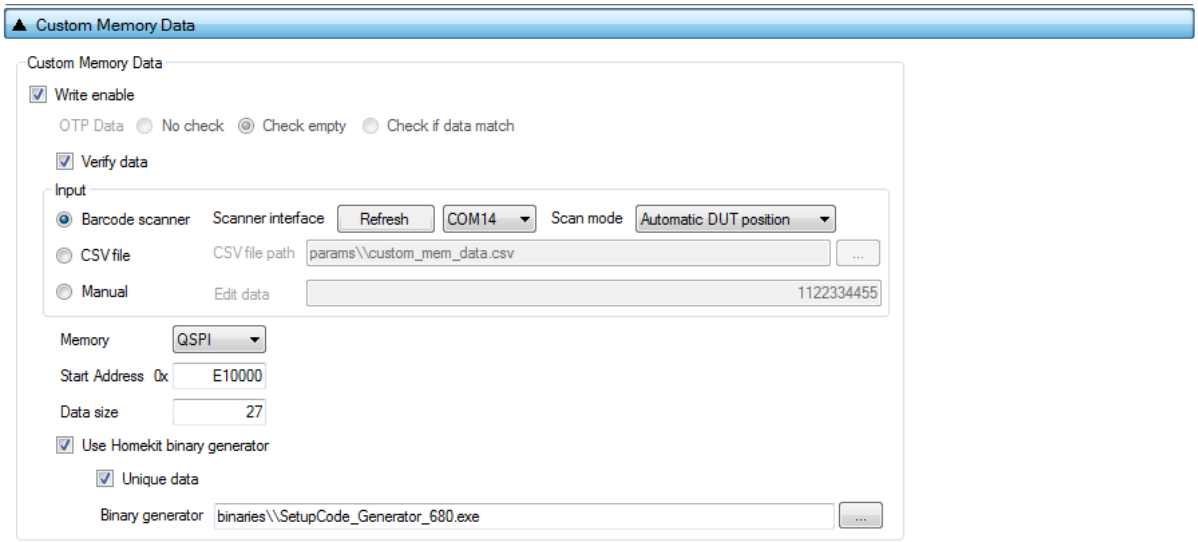



**Figure 106: Custom Memory Data - DA1469x**

Table 96 describes the *Custom Memory* data test options. With this test, the user can write any data to any address to any available memory for the DA1469x devices, such as OTP, register initialization

 © 2020 Dialog Semiconductor

at OTP-TCS section and QSPI. Data input modes can be a Barcode Scanner, a CSV file or data entered manually. For the DA1469x devices, the data entry can be used as input to the Homekit binary generator to create a binary and automatically write it to a memory.

**Table 96: Custom Memory Data - DA1469x**

<table>
<tr><th>Option</th><th colspan="2">Description</th></tr>
<tr><td>Write enable</td><td colspan="2">This option enables the Custom data programming.</td></tr>
<tr><td>Verify data</td><td colspan="2">When selected, the data written will be read back from the memory and will be compared to the original.</td></tr>
<tr><td>● No check<br>● Check empty<br>● Check if data match<br>● Skip if empty<br>(Only available when OTP memory, OTP TCS field or CSV file as input is selected)</td><td colspan="2">Memory protection options:<br>**No check:** No protection is enabled. PLT will attempt to burn the OTP memory without running any check.<br>**Check empty:** PLT will first check if the OTP memory to be burned is empty. If it is empty will burn it.<br>**Check if data match:** PLT will first check if the memory to be burned is empty. If it is not, it will compare the contents with the data to be burned. If the data are not the same the test will fail without making any changes to the memory. If the data are the same, PLT will not burn the memory to prevent using the OTP repair memory.<br>**Skip if written:** PLT will read the contents of OTP memory. If it contains any data, it will skip writing without producing any error.</td></tr>
<tr><td rowspan="5">● Barcode scanner<br>● CSV file<br>● Manual data</td><td colspan="2">**Note:** Barcode scanner mode is only available with GUI PLT Application. CLI PLT Application does **NOT** support this feature.<br>Custom data input options:</td></tr>
<tr><td>Scanner interface (Barcode scanner)</td><td>Selection of the Barcode scanner input. Both HID and COM port interfaces are supported. This list provides an HID and all available COM ports as selectable options.<br>Any HID device is supported that includes newline (CR-LF) characters at the end of the scanned data.<br>For the COM port interface, a common USB to UART barcode scanner is supported. PLT has been tested with Honeywell Xenon 1900. Appendix M describes the setup procedure.<br>This option is the exact same option as in Scan Mode and the DA1469x devices in Custom Memory Data.</td></tr>
<tr><td>Scan mode (Barcode scanner)</td><td>**Scan DUT position:** In this mode, the users must first scan the DUT position number and then the BD address. The string used for the position of each DUT is "TEST POSITION 0xx" where "xx" is the DUT position number.<br>**Automatic DUT position:** Scanned BD address will be assigned to the selected DUT. The DUT selection is automatically been made, starting from the first active DUT and selecting the next one after a successful BD address scan. Users can change the selected DUT using the controls on the GUI PLT screen shown in Figure 112.<br>This option is the exact same option as in Scan Mode and the DA1458x devices in Custom Memory Data.</td></tr>
<tr><td>CSV file path (CSV file)</td><td>Path to the CSV file containing data for each device discriminated using BD addresses. The CSV file format is described in Custom data CSV file format.</td></tr>
<tr><td>Edit data (Manual data)</td><td>Hexadecimal data input of up to 256 bytes to burn. These data will be burned to all active DUTs.</td></tr>
</table>

 © 2020 Dialog Semiconductor

| Option | Description |
|---|---|
| Memory (available with Barcode scanner and Manual data modes) | Memory type selection to burn the data. Available options are OTP, QSPI and OTP TCS (only with `Manual data`).<br>**Note:** When `Manual Data` input mode is selected, writing a register in `OTP TCS` section is also supported. In this mode, the address field changes to register address and the data to register value. PLT will automatically find the TCS slot to burn the value, and the mirrored equivalent. |
| Start address (available with Barcode scanner and Manual data modes) | Memory address offset to begin burning the data. OTP valid address is 0x10080000 – 0x1008FFFF. |
| Data size (available with Barcode scanner and Manual data modes) | The size of the memory data to burn. In barcode scanner, the size is the number of scanned characters. In manual data, the size is number of bytes.<br>**Note:** If the `Homekit binary generator` is used, the data size entered here is the size of the setup code and serial number as explained in Table 17. In the example the data to be scanned are "`5086747870SX03112233445566R`", meaning that the data size should be 27 since ASCII character will be scanned The data size to be actually burned into the devices will be 574 bytes, which is the binary size created by the `SetupCode_Generator_680.exe` application. |
| Use Homekit binary generator (available with Barcode scanner mode) | If enabled the input memory data from the barcode scanner will be applied as input to the Dialog Homekit setup code binary generator. PLT will automatically call the setup code binary generator and burn the files created. |
| Unique data (available with Barcode scanner mode) | If enabled the input memory data will be compared to each other and if same data are found an error will be issued. Comparison can only be performed per current PLT test run and not for previous tested devices. |
| Binary generator (available with Barcode scanner mode) | The path to the Homekit setup code binary generator executable. PLT will automatically call this application and burn the files created. |

### 7.2.13 Debug Settings

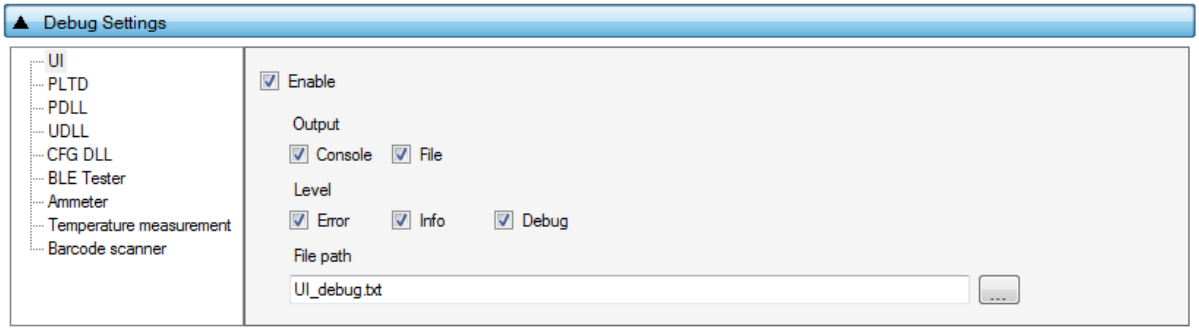



**Figure 107: Debug Settings**

Table 97 describes the available options for the *Debug Settings*. Debug messages are available in all PLT software blocks shown in Figure 21.

**Note:** Printing debug information may introduce system delay and thus some tests may fail due to time out expirations. We suggest having debug information disabled in all software blocks and only partially enable when there is a real need for it. From PLT v4.0 and onwards, this system delay has been almost eliminated as debug print messages are printed from a lower priority queue. It is safer, but it is still suggested to have the debug prints disabled.

**Table 97: Debug Settings**

| Option | Description |
|---|---|
| Enable | Enable debug message prints for the selected library or UI. |


© 2020 Dialog Semiconductor

| Option | Description |
|---|---|
| Output - Console | Sends the debug messages to the `stdio` output. The PLT CLI does not support this option. If enabled, debug messages will be redirected to the equivalent files. |
| Output - File | Save the debug messages to a file. |
| Level - Error | Enable error debug level messages. All debug print messages marked as error will be printed. |
| Level - Info | Enable info debug level messages. All debug print messages marked as info will be printed. |
| Level - Debug | Enable low level debug level messages. All low level debug print messages will be printed. |
| File path | Select the file that the debug messages will be saved. The file should exist; otherwise, it should be created manually. Used only when the option Output - File is selected. |

### 7.2.14 Security

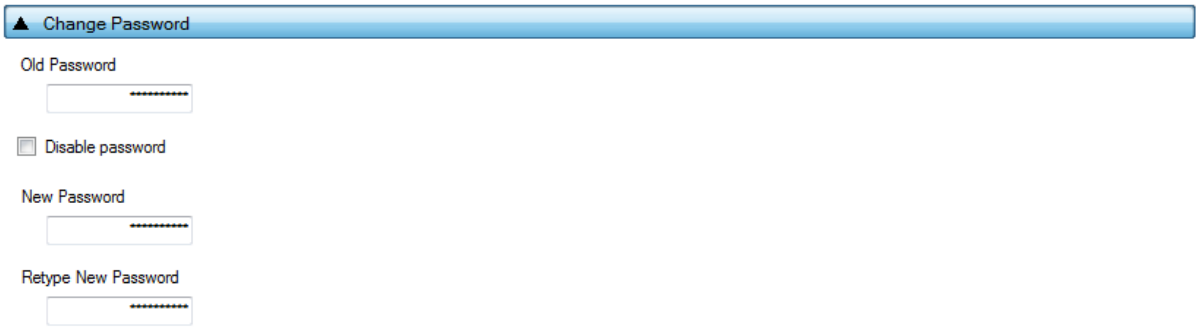



**Figure 108: Security**

In this field, a password can be set to protect specific tool actions, such as:

- Opening the CFG PLT or the GUI PLT application.
- Closing the CFG PLT or the GUI PLT application.
- Opening or refreshing configuration settings in the GUI PLT application.
- Opening the settings menu in the GUI PLT application.

Table 98 describes the available options for the *Security* Options.

**Table 98: Security Options**

| Option | Description |
|---|---|
| Old Password | Type the current password to enable changing of the following fields. |
| Disable Password | This option will disable the password usage. |
| New Password | Type a new password. |
| Retype New Password | Verify the new password. |


© 2020 Dialog Semiconductor

## 7.3 GUI PLT Application

The GUI PLT (`SmartBond_GUI_PLT.exe`) is a Graphical User Interface application that performs the device validation and programming process. At the same time, it allows the users to monitor the entire procedure in detail. The GUI PLT uses the same XML file configured from CFG PLT as described in section 7.2.

**Note:** If a change is made to the XML file from the CFG PLT, then the GUI PLT settings should be refreshed as described in Table 99.

Figure 109 shows the initial screen of the GUI PLT, which is described in Table 99.

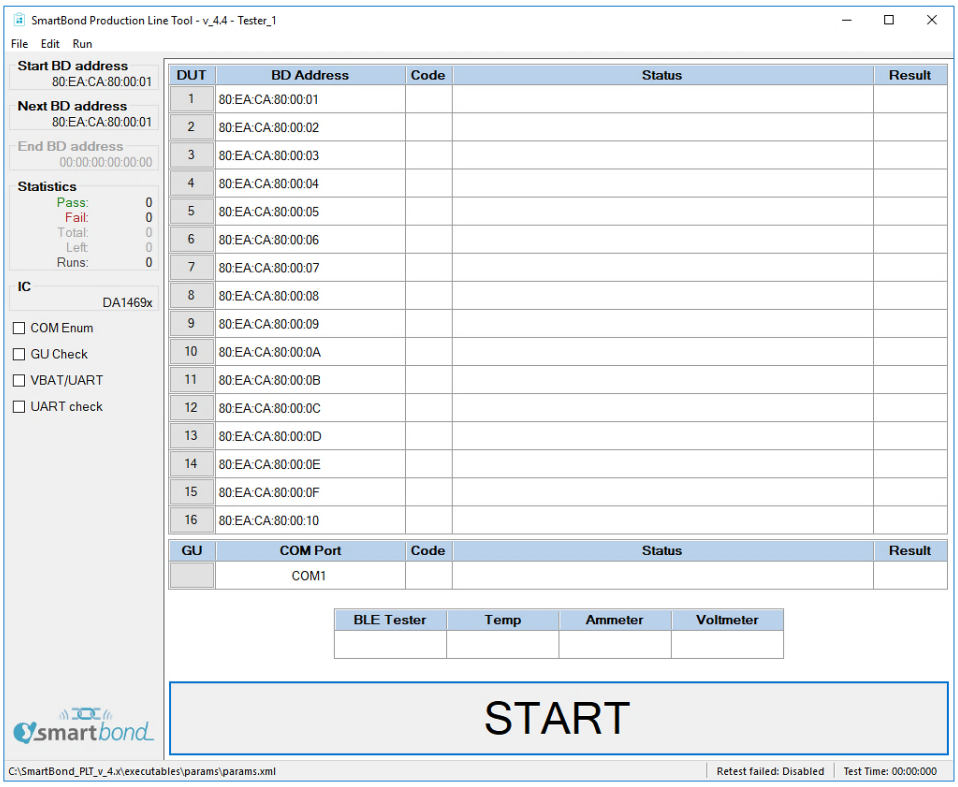



**Figure 109: GUI PLT Main Screen**

**Table 99: GUI PLT Main Screen Description**

| Options | Description |
|---|---|
| **File options** | |
| File > Open XML file | Opens a new XML file and loads its settings. The full path of the new XML file is shown at the bottom end of the screen. |
| File > Refresh XML file | Reloads the settings from the XML file and initializes itself with the new settings. |
| File > Open CSV file | Contains a list with all the available CSV files to open. |
| File > Exit | Exits the GUI PLT application. |
| **Edit options** | |
| Edit > Settings | Opens the GUI PLT Settings window. |
| **Run options** | |
| Run > Run Configuration PLT | Opens the CFG PLT application. |


© 2020 Dialog Semiconductor

| Options | Description |
|---|---|
| **Left Column options** | |
| Start BD Address | The BD address the PLT session started with, as described in section 7.2.4. |
| Next BD Address | The BD address that will be used on the BD address assignment for the next run as described in section 7.2.4. |
| End BD Address | The BD address the PLT session ends with as described in section 7.2.4. This option is available only when *Range mode* is enabled. |
| Statistics | This field holds statistics for each PLT session. Table 29 describes the *Statistics* field. |
| IC | The selected IC of the PLT. |
| COM Enum | If this checkbox is enabled then the START button initiates the automatic Window COM port enumeration for the DUT. |
| GU Check | If this checkbox is enabled then the START button initiates the automatic Window COM port enumeration for the Golden Unit. |
| VBAT/UART | If this checkbox is enabled then the START button will enable the VBAT and UART for the DUTs selected under *VBAT/UART* in Table 100. |
| UART check | If this checkbox is enabled then the START button initiates the UART check procedure for the DUTs with a specified Baud rate set from the user through the GUI PLT Settings. During this test, 1000 packets will be sent, received back and checked for errors. For the DA14531 and DA1469x DUTs the packets contain 252 bytes.<br>**Note:** Before any UART transfer begins, PLT ill download the production test firmware to the active DUTs. |
| **Center screen options** | |
| DUT panel | Shows the following fields for each DUT:<br>● **DUT:** DUT connector number on the PLT hardware. This field is also a button that opens the Log file for the specific DUT.<br>● **BD Address:** BD address assigned to the DUT.<br>● **Code:** Real-time status as a PLTD DLL special code described in [1].<br>● **Description:** A brief description of the status code.<br>● **Result:** Simplified color-coded status showing the progress per DUT. |
| GU panel | Shows the following fields for the Golden Unit:<br>● **GU:** A button that opens the Golden Unit Log file.<br>● **COM Port:** The COM port assigned to the Golden Unit.<br>● **Code:** Real-time status as a PLTD DLL special code described in [1].<br>● **Status:** A brief description of the status code.<br>● **Result:** Simplified color-coded status showing the progress of the GU. |

 © 2020 Dialog Semiconductor

| Options | Description |
|---|---|
| Instrument panel | This field shows a simplified color-coded status is shown for each of the instruments (**BLE Tester**, **Temp, Ammeter** and **Voltmeter**), if they are enabled. |
| START button | If one of the options *COM Enum*, *GU Check, VBAT/UART* or *UART check* is enabled, then selecting the *START* button will initiate the chosen test. If no option is selected, selecting the *START* button initiates the production procedure.<br>**Note:** To select and press the `Start Button` press the `space-bar` key. The `Start Button` can only be pressed with the mouse (or use the `'f'` key as a shortcut), after the selected procedure is finished, in order to return to main screen. This is to avoid pressing the `Start Button` and starting a new test procedure, by mistake. |
| **Bottom of the main screen** | |
| Left panel:<br>C:\SmartBond_PLT_v_4.x\executables\params\params.xml | Shows the full path of the XML file that is currently used. |
| Center panel:<br>Retest failed: Disabled | Shows if the Re-test option in GUI PLT Settings is enabled. |
| Right panel:<br>Test Time: 00:00:000 | This timer starts counting when the START button is pressed and runs until the PLT returns to its idle state, showing the approximate duration of the tests. |

### 7.3.1 GUI PLT Settings

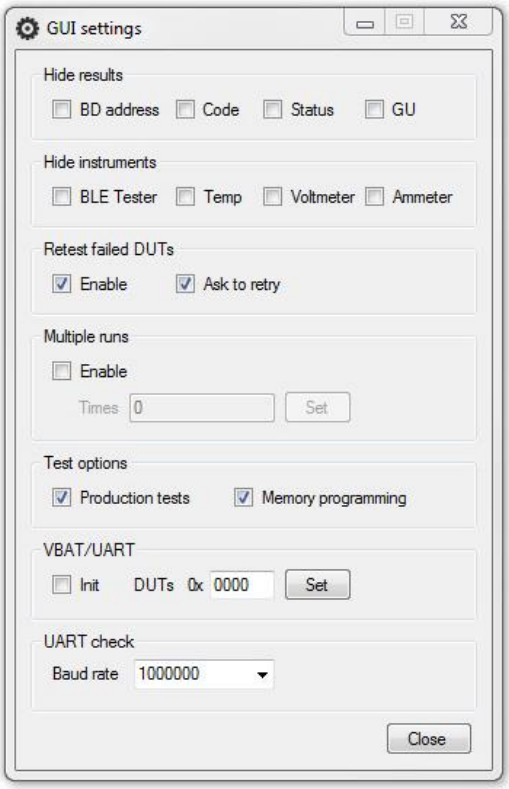



**Figure 110: GUI PLT Settings**


© 2020 Dialog Semiconductor

Figure 110 shows the *GUI PLT settings* window. In this window, various graphic options and features can be set as described in Table 100.

**Table 100: GUI PLT Settings**

| Field | Option | Description |
|---|---|---|
| Hide results | BD address | This option will hide the *BD address* column in the DUT panel of the GUI PLT. |
| | Code | This option will hide the *Code* column in the DUT panel of the GUI PLT. |
| | Status | This option will hide the *Status* column in the DUT panel of the GUI PLT. |
| | GU | This option will hide the *GU* column in the DUT panel of the GUI PLT. |
| Hide instruments | BLE Tester | This option will hide the *BLE Tester* column in the GU panel of the GUI PLT. |
| | Temp | This option will hide the *Temp* column in the GU panel of the GUI PLT. |
| | Voltmeter | This option will hide the *Voltmeter* column in the GU panel of the GUI PLT. |
| | Ammeter | This option will hide the *Ammeter* column in the GU panel of the GUI PLT. |
| Retest failed DUTs | Enable | If this option is enabled, any DUT that failed during the main procedure will immediately re-run the tests having the exact same options including the *BD address* assigned to it. This option is the exact same option as *Re-test failed DUTs* in section 7.2.3.2. |
| | Ask to retry | This option will show a message asking to do a re-test in case any DUT failed. If this option is disabled, the re-testing will be done automatically. |
| Multiple Runs | Enable | By enabling this option, the GUI PLT will perform multiple procedures without any delay between them. This is used for only for evaluation. |
| | Times | The number of times to run. |
| Test Options | Production tests | Enables /Disables the production test procedure. This is the same option as *Production tests* in section 7.2.3.2. |
| | Memory programming | Enables /Disables the production test procedure. This is the same option as *Memory programming* in section 7.2.3.2. |
| VBAT/UART | Init | If this option is enabled, the PLT hardware will be reset before enabling the DUTs. This option is enabled only when *VBAT/UART* in the main screen is enabled. |
| | DUTs | Bitwise DUT set/reset for each of the 16 DUTs using a 16-bit hexadecimal value. 089<br>**Example:** To enable only DUTs 1, 2, 15 and 16 use "C003" (1100 0000 0000 0011 = 0xC003). |
| UART check | Baud rate | Sets the Baud rate for the UART check test. |

### 7.3.2 Barcode Scanner Mode

A barcode scanner can be used for two purposes. It can be used to scan DUT BD addresses and/or Custom Memory Data. If any these options have the Barcode scanner option enabled then the Barcode Scan option will appear in the GUI PLT as shown in Figure 111. If both options are enabled then the GUI PLT will first use the Barcode scanner for the BD address Scan Mode assignment and then for the Custom Memory Data.

In all cases described in section 7.2.4.1 except Scan Mode, the GUI PLT assigns BD addresses right before the PLT starts the production test run. Device BD addresses should be scanned before the start of a production test run. If the *START* button is pressed without any BD address being assigned to the device, the PLT will **not** run the tests.

 © 2020 Dialog Semiconductor

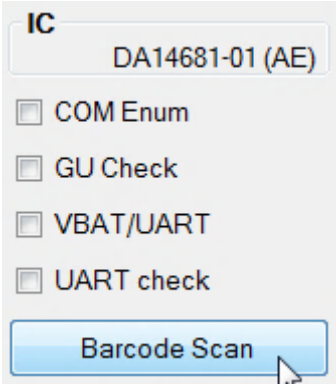



**Figure 111: Barcode Scan Option in GUI PLT**

Two different device BD address scanning procedures are supported. If the same BD address is used twice an error message appears in the DUT panel. It then waits for a unique BD address. An example is shown in Figure 113.

1. *Scan DUT position*. In this mode, the user must first scan the DUT position and then the BD address. The string used for the position of each DUT is "TEST POSITION 0xx" where "xx" is the DUT number 1 to 16.
2. *Automatic DUT position*. The scanned BD address will be assigned to the selected DUT. DUT selection is done automatically. The PLT starts from the first active DUT and goes to the next after a successful BD address scan. The user can change the selected DUT via the controls shown in Figure 112. If the scanned BD address was successfully assigned, the PLT will automatically select the next active DUT and wait for a new BD address to be scanned.

**Figure 112: Barcode Scanner Controls**

If the Custom Memory Data test requires data to be scanned then the user must scan the Custom data after the BD address for each DUT. Homekit Setup Code Scan Example provides detailed steps showing how to scan both BD addresses and data for the Custom Memory data test.

After all active DUTs have BD addresses assigned; the user should press the *END* button in the controls to return to the main screen. Pressing the *START* button will then start the test execution.

**Note:** If the *Barcode Scan* button is pressed again, all BD addresses will be reset and the BD address assignment procedure will begin again.

| DUT | BD Address | Code | Status | Result |
|---|---|---|---|---|
| 1 | 11:22:33:44:55:06 | | BARCODE SCANNER BD ADDRESS OK | PASS |
| 2 | 00:00:00:00:00:00 | | BD ADDRESS 11:22:33:44:55:06 ALREADY USED. RETRY | CHECK |
| 3 | 11:22:33:44:55:08 | | BARCODE SCANNER BD ADDRESS OK | PASS |
| 4 | 11:22:33:44:55:09 | | BARCODE SCANNER BD ADDRESS OK | PASS |
| 9 | 11:22:33:44:55:13 | | BARCODE SCANNER BD ADDRESS OK | PASS |
| 10 | 00:00:00:00:00:00 | | | |
| 11 | 00:00:00:00:00:00 | | | |
| 12 | 00:00:00:00:00:00 | | | |

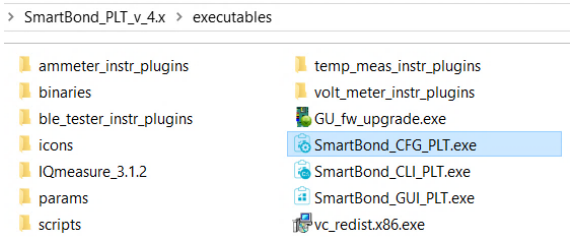
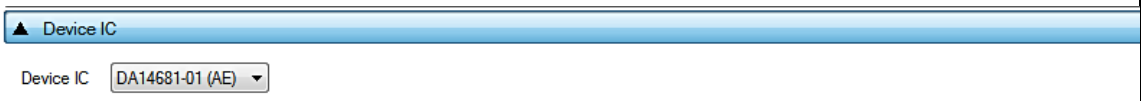
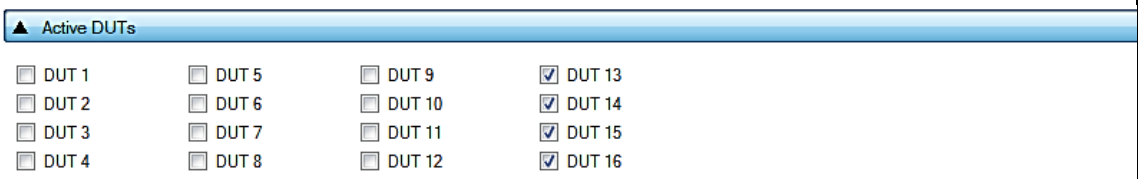
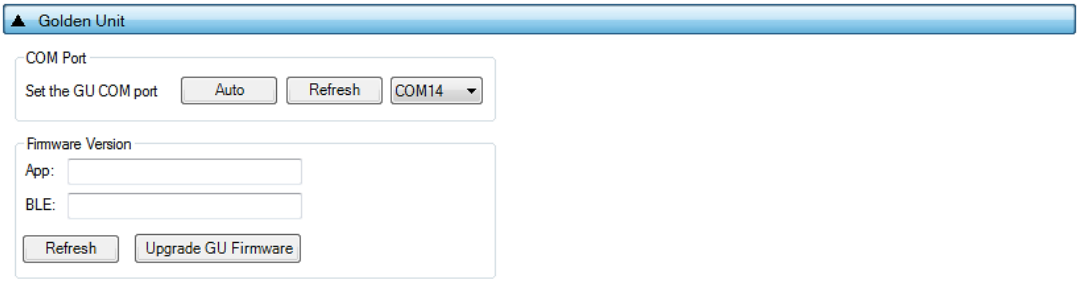
**Figure 113: Barcode Scan - BD Address Assignment**


© 2020 Dialog Semiconductor

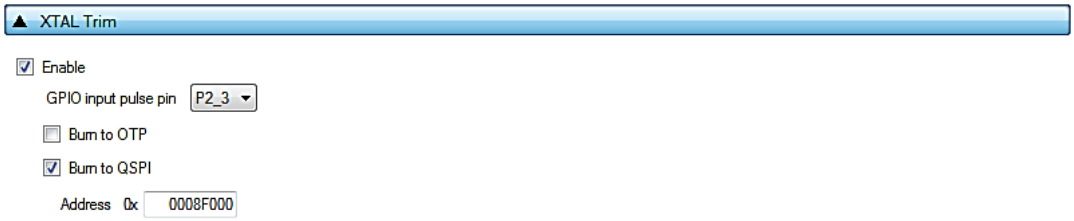
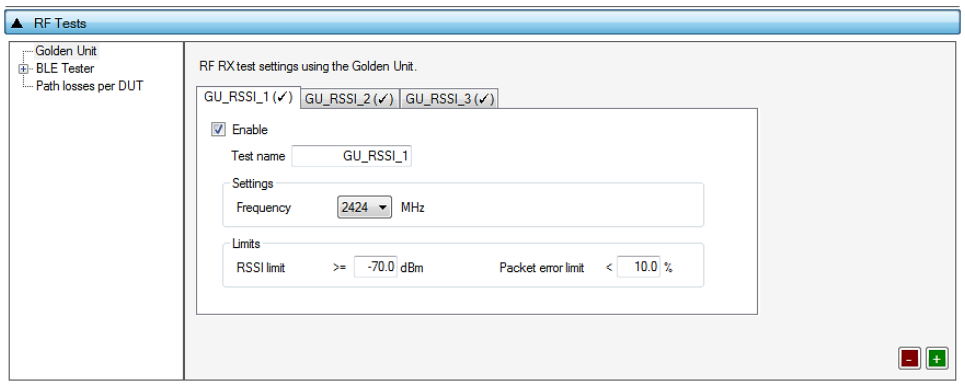
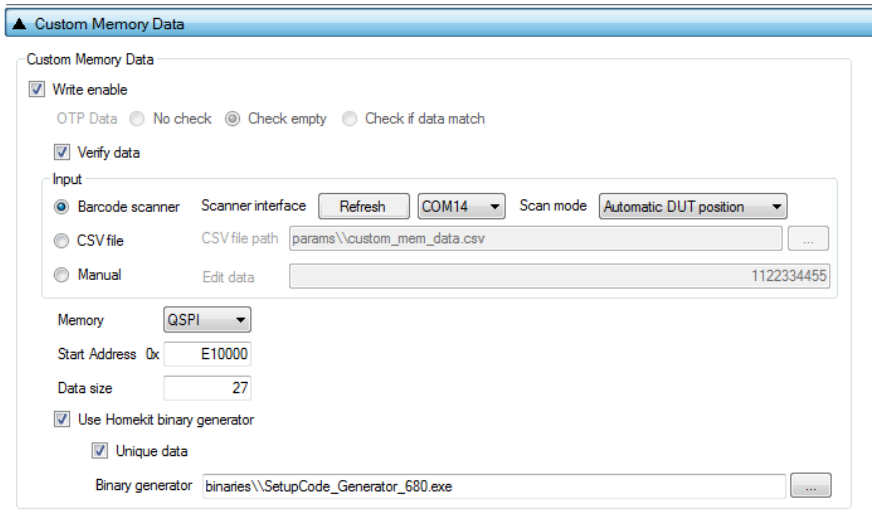
#### 7.3.2.1 Homekit Setup Code Scan Example

An example of using the barcode scanner and Custom Memory Data will be given. A barcode scanner will be used to scan different Homekit setup codes. PLT will call the *SetupCode_Generator_680.exe* application to create the binaries that contains the DUTs serial numbers and the hashed version of the setup codes. Finally, it will program the binaries to the DUTs. The process will be described in Table 101. The example will use DA14681-01 DUTs and configure the PLT such that to perform XTAL trim test, RF test and homekit setup code scanning and programming.

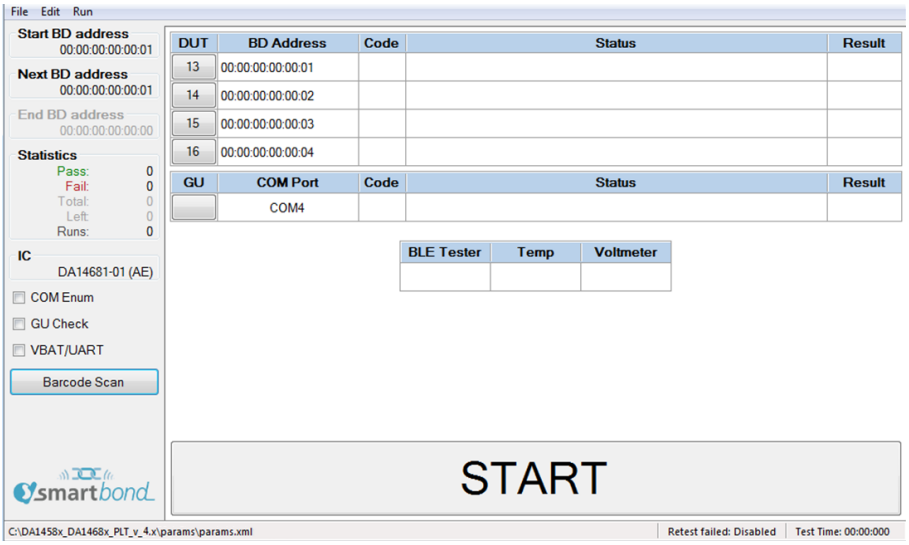
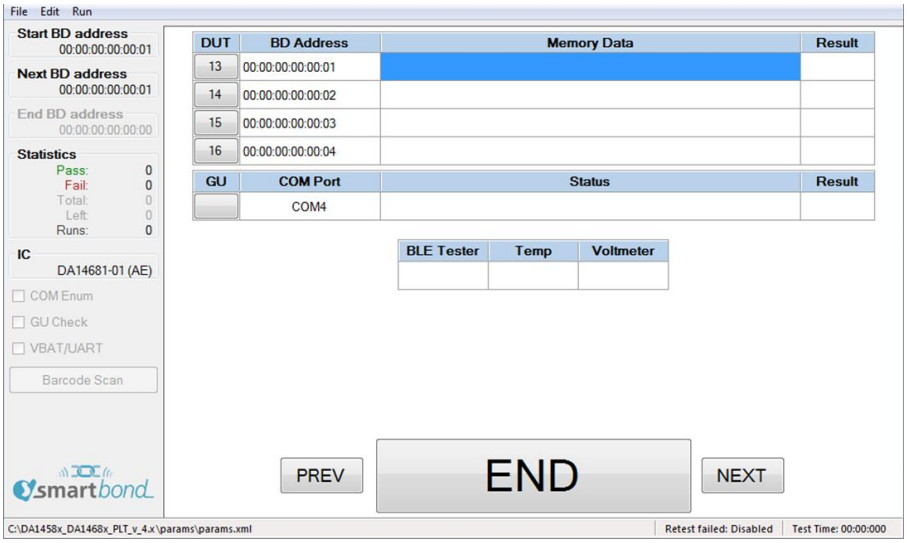
**Table 101: Homekit Setup Code Scan Example**

| # | Action |
|---|---|
| 1 | Copy PLT software `SmartBond_PLT_v_4.x.x.x` under `C:\` directory. |
| 2 | Open *SmartBond_CFG_PLT.exe* from executables folder.<br>  |
| 3 | Go to *Hardware Setup-> Device IC* and select *DA14681-01 (AE).* Press the *Save** button.<br>  |
| 4 | Go to *Hardware Setup-> Active DUTs* and select *DUT13, DUT14, DUT15* and *DUT16.* Press the *Save** button.<br>  |
| 5 | Go to *Hardware Setup-> Golden Unit COM Port* and auto-detect the COM port. Press the *Auto* button. Press the *Save** button.<br>  |

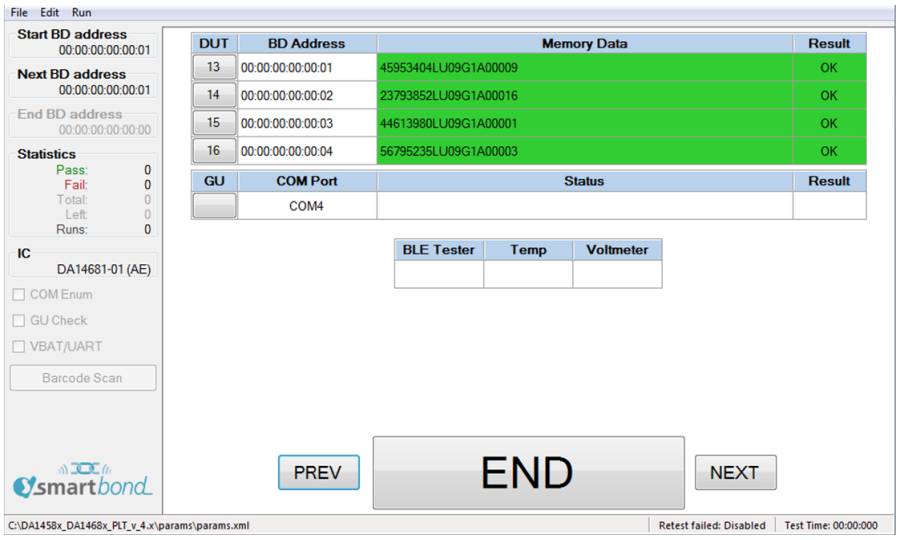
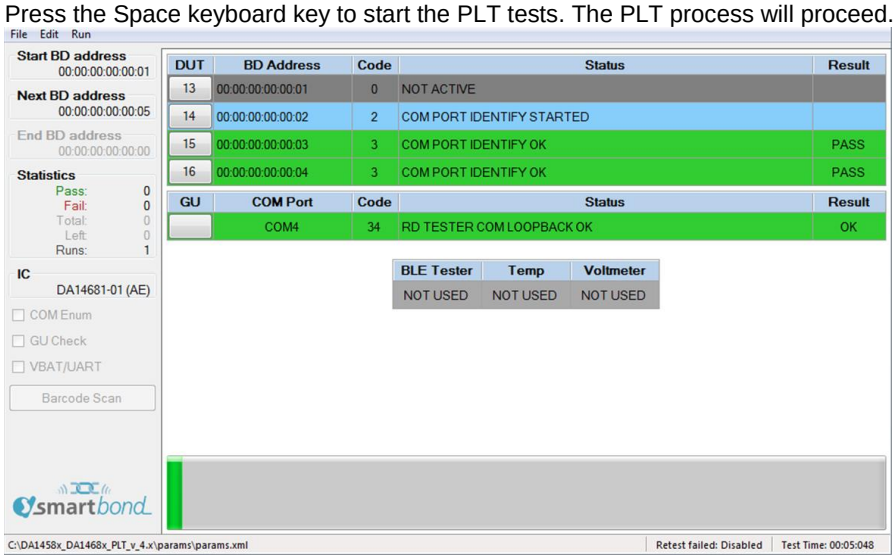
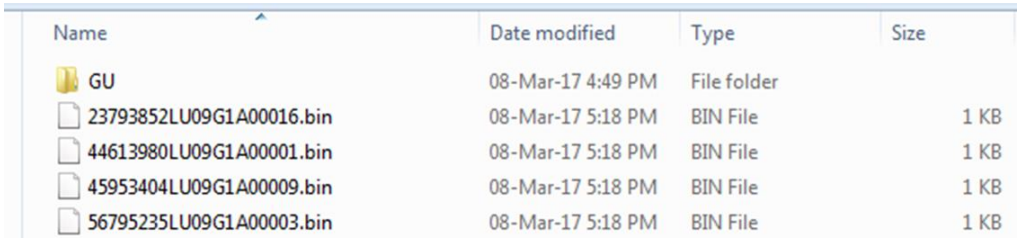
 © 2020 Dialog Semiconductor

| # | Action |
|---|---|
| 6 | Go to *Test Settings->XTAL Trim* and enable the settings as shown in the following picture. Press the *Save** button.<br>These settings will enable the XTAL trim calibration test. The result of the XTAL trim calibration will be saved into QSPI flash. Dialog SDK firmware is able to read the value from this specific QSPI address (0x8F000) and apply it to the appropriate chipset XTAL trim register.<br>  |
| 7 | Go to *Test Settings->RF Tests* and enable the settings shown in the following picture. Three RF tests at channels 2424MHz, 2450MHz and 2476MHz are already enabled in the PLT by default. Check that all settings are correct. Press the *Save** button.<br>  |
| 8 | Deselect everything in *Memory Functions* tab. No need to burn any image for this example. Press the *Save** button. |
| 9 | Connect to the PC a USB to Serial (COM) barcode scanner instrument. Keep the Windows COM port assigned to the instrument. |
| 10 | Go to *Memory Header->Custom Memory Data* and enable the settings shown in the following picture. Change the *Scanner Interface* COM port according to the COM port assigned to the barcode scanner in step 9. Press the *Save** button.<br>  |
| 11 | Close *SmartBond_PLT.exe* and open *SmartBond_GUI_PLT.exe.* |

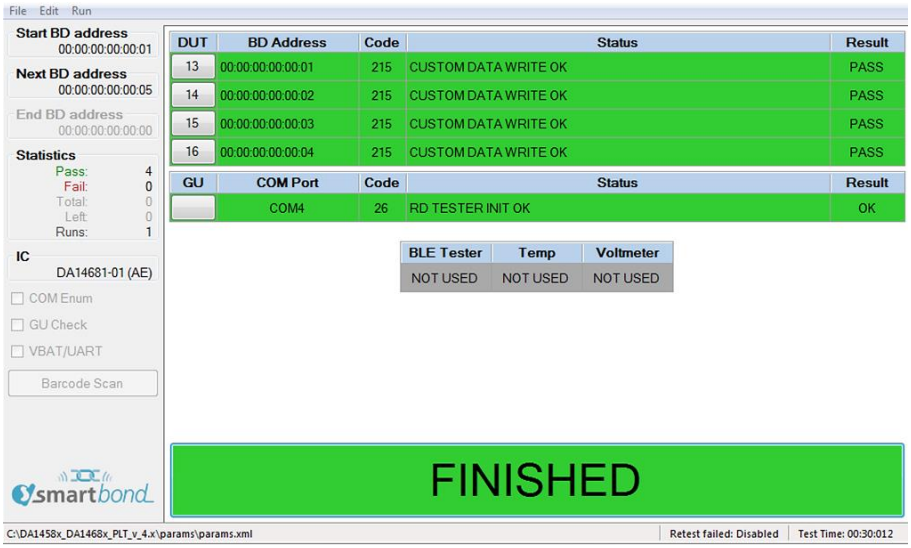
 © 2020 Dialog Semiconductor

| # | Action |
|---|---|
| 12 | The initial *SmartBond_GUI_PLT.exe* screen will appear.<br>  <br> |
| 13 | Press the *Barcode Scan* button on the bottom left corner. The following screen will appear.<br>  <br> |
| 14 | Use the barcode scanner instrument to scan four different homekit setup codes with specific format as this was described in Table 17. Four different example codes are given next.<br>45953404LU09G1A000096<br>23793852LU09G1A000161<br>44613980LU09G1A000011<br>56795235LU09G1A000038 |

 © 2020 Dialog Semiconductor

| # | Action |
|---|---|
| 15 | When the barcodes, given in step 14, are scanned using the barcode scanner instrument, the GUI PLT will be as shown next.<br>  <br> |
| 16 | If something goes wrong, either navigate to the different DUT Memory Data cells using the *PREV/NEXT* buttons or press the *END* button and then the *Barcode Scan* button again to restart the process. The previous scanned data will be erased.<br>If everything went well press the *END* button. The tool will return to its initial screen as shown in step 12. |
| 17 | Press the Space keyboard key to start the PLT tests. The PLT process will proceed.<br>  <br> |
| 18 | PLT will get the barcode scanned data for each DUT and call the *SetupCode_Generator_680.exe* to create four different binaries. The binaries will reside inside *SmartBond_PLT_v_4.x.x.x\executables\binaries*.<br>  <br> |


© 2020 Dialog Semiconductor

| # | Action |
|---|---|
| 19 | If everything finished with no error the following screen will be shown.   |
| 20 | Press the FINISHED button to return to the main screen. |

### 7.3.3 Running the GUI PLT and Executing Tests

The GUI PLT starts the test procedure when users press the *START* button. Before initiating the test procedure, the GUI PLT will assign BD addresses to the active DUTs and check for any wrong configuration parameters.

If `Run scripts before testing starts` is enabled, PLT will execute the selected script/executable, and wait until it finishes or times out, depending on the selections made in Test Options. If the script/executable has returned on time, PLT will check the return code. Values from 0 to 100 indicate a successful completion. Negative values or values larger than 100 indicate an error. In the case of an error (either time out or error returned result), a pop-up message will appear indicating the return code and the test procedure will not start.

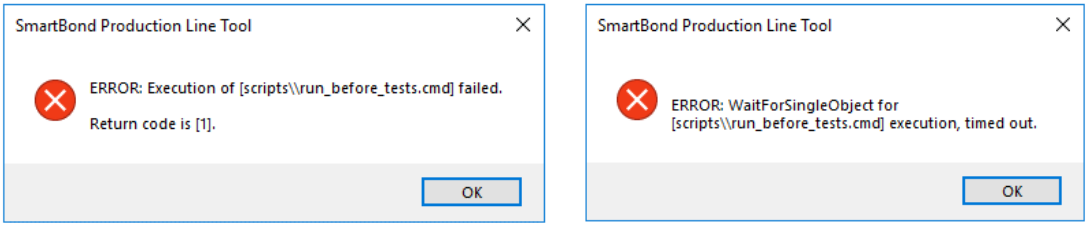



**Figure 114: GUI PLT - Erroneous Messages in "run scripts before testing starts"**

If any OTP burning test is scheduled, a pop-up message will inform the user and prompt for continuing (Figure 115).

 © 2020 Dialog Semiconductor

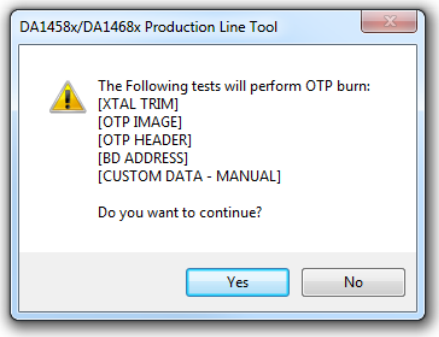



**Figure 115: GUI PLT OTP Burn Warning Message**

Immediately thereafter, the testing procedure will begin. PLT will update the status of the procedure for each DUT and the Golden Unit (Figure 116). The *START* button is replaced by a progress bar indicating the progress of the tests.

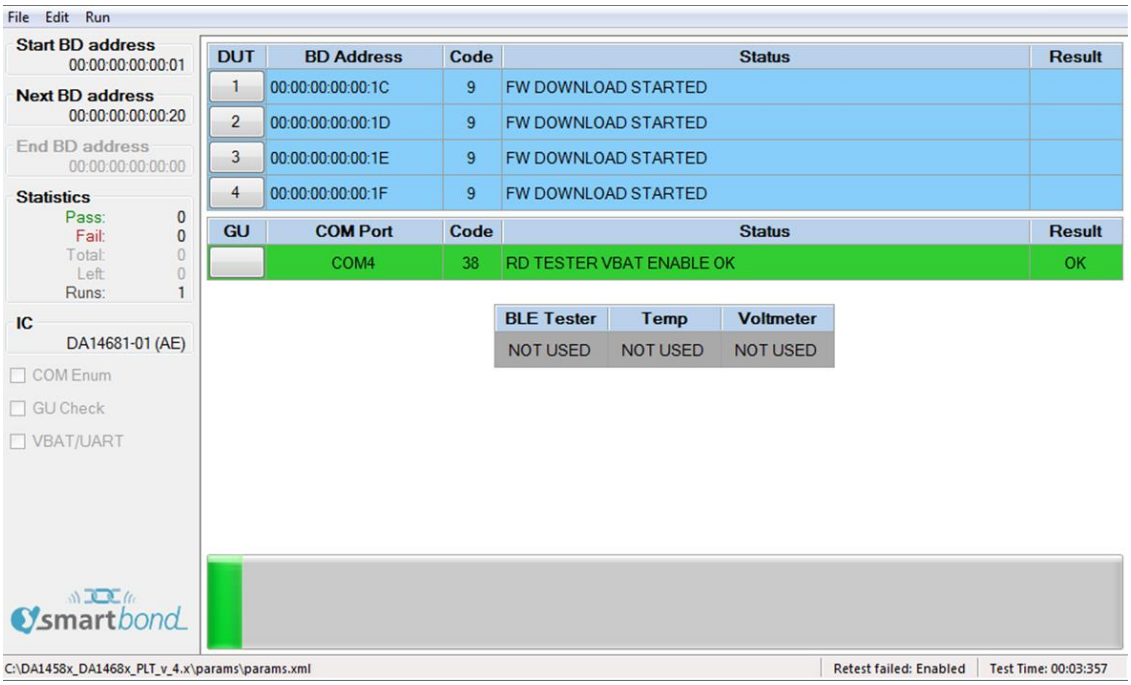



**Figure 116: GUI PLT Testing (1 of 2)**

If an error in a DUT is found (Figure 117), PLT will show the status code, a brief description of the error and the color of the DUT's status line will turn red. At all times the DUT number button can be pressed to access the DUT Log File to get more details about the parameters used, calculated values and the reason of failure in the case of an error.


© 2020 Dialog Semiconductor

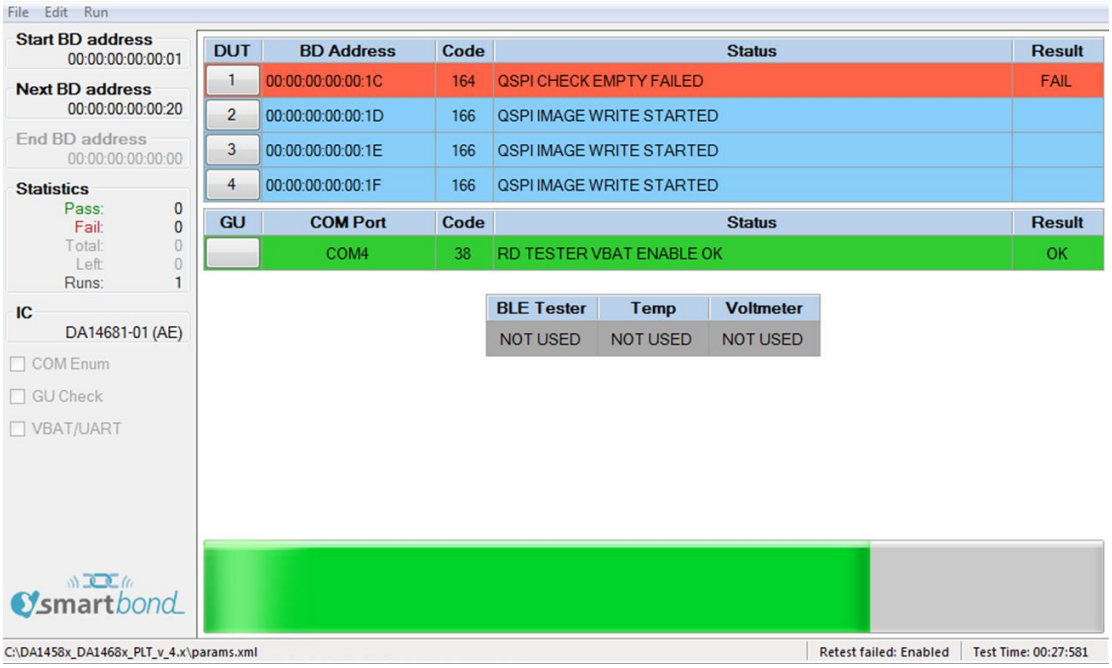



**Figure 117: GUI PLT Testing (2 of 2)**

After the testing procedure is completed (Figure 118), the progress bar shows *FINISHED* and the color will be red if any DUT has failed, otherwise it will be green. If there is an error and the *Retest failed DUTs* and *Ask to retry* options are enabled, a message will appear asking if the user would like to retest the failed DUTs, as shown in Figure 119. When the GUI PLT performs a retest run, all options (including the BD addresses) remain the same and only the tests that failed are retested. At this time, the CSV File and all the DUT Log Files will be updated.

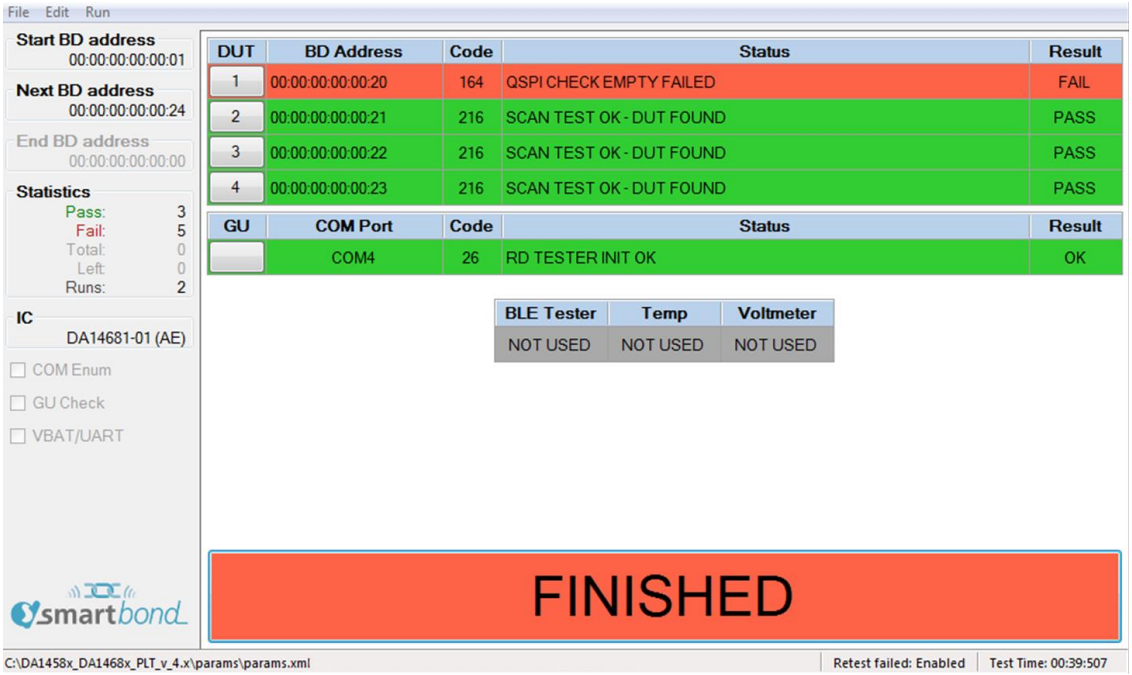



**Figure 118: GUI PLT Tests Finished**


© 2020 Dialog Semiconductor

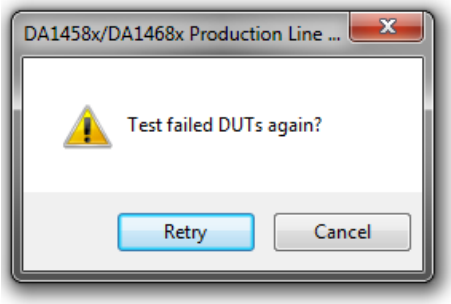



**Figure 119: GUI PLT Retry Failed DUTs Message**

If the DUT fails again, after the retest has finished the GUI PLT will remain in the *FINISHED* screen (Figure 118) with the *FINISHED* button shown in red.

If `Run scripts when testing is finished` is enabled, pressing the finished button will execute the selected script/executable. As with `Run scripts before testing starts`, PLT will wait until it finishes or times out, depending on the selections made in Test Options. If the script/executable has returned on time, PLT will check the return code. Zero value indicates a successful completion. Any other value is considered an error. In the case of an error (either time out or error result), a pop-up message will appear indicating the return code.

### 7.3.4 Debug Console

Section 7.2.12.2 shows the debug settings for all PLT applications including the GUI PLT. If at least one debug session is enabled with the output set to *Console,* the GUI PLT will open a new console window showing the desired debug information.

Figure 120 shows an example of the *Debug Console*. Depending on the type of the message, a different color is used: *DEBUG* messages are light blue, *INFO* messages are white and *ERROR* messages are red.

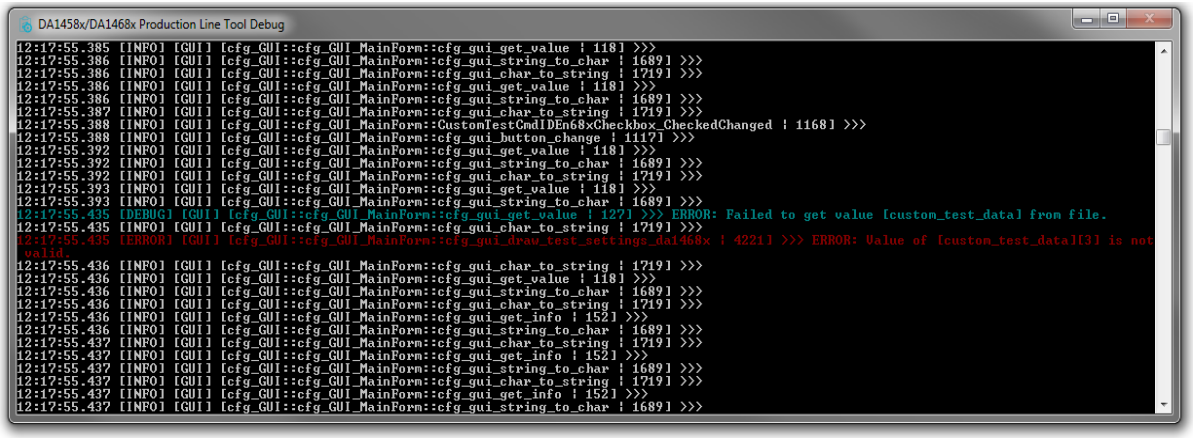
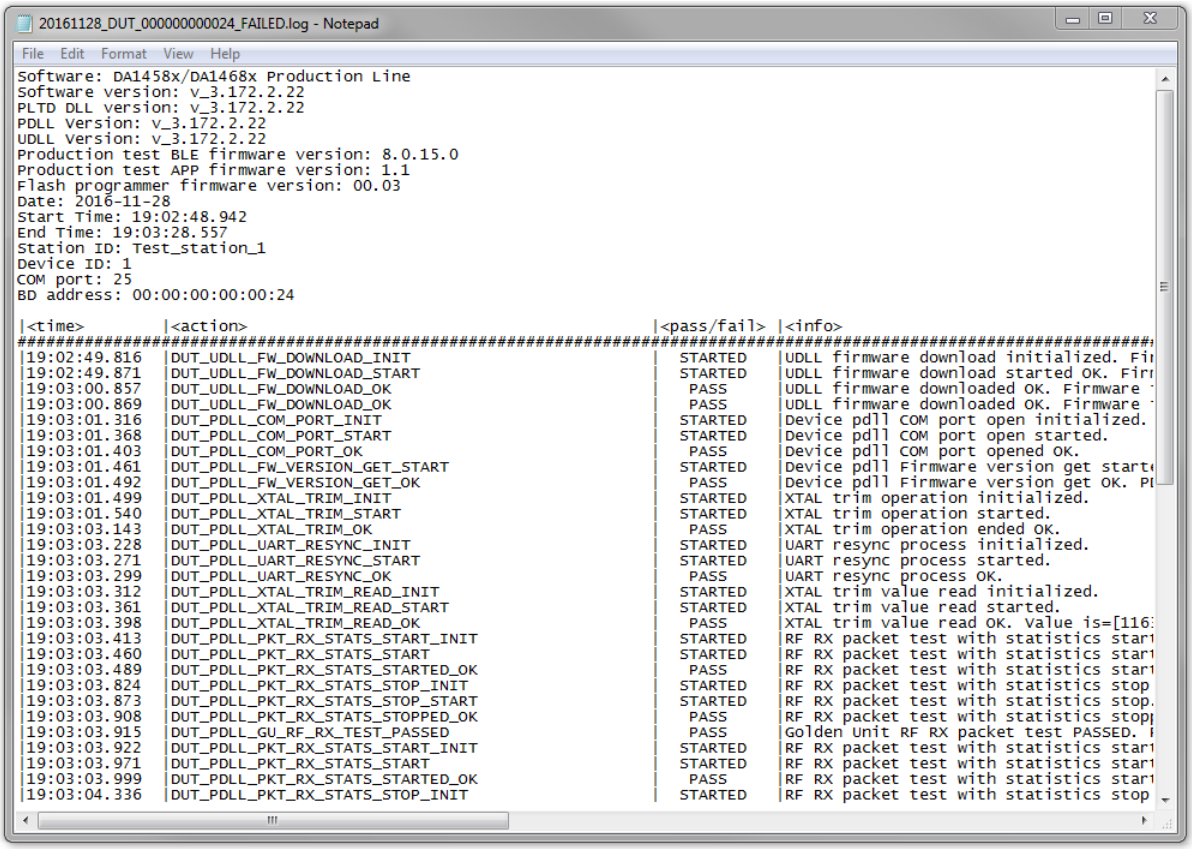



**Figure 120: Debug Console**


© 2020 Dialog Semiconductor

### 7.3.5 DUT Log File



```
20161128_DUT_000000000024_FAILED.log - Notepad
File  Edit  Format  View  Help
Software: DA1458x/DA1468x Production Line
Software version: v_3.172.2.22
PLTD DLL version: v_3.172.2.22
PDLL Version: v_3.172.2.22
UDLL Version: v_3.172.2.22
Production test BLE firmware version: 8.0.15.0
Production test APP firmware version: 1.1
Flash programmer firmware version: 00.03
Date: 2016-11-28
Start Time: 19:02:48.942
End Time: 19:03:28.557
Station ID: Test_station_1
Device ID: 1
COM port: 25
BD address: 00:00:00:00:00:24

|<time>        |<action>                                          |<pass/fail> |<info>
##############################################################################################################
|19:02:49.816  |DUT_UDLL_FW_DOWNLOAD_INIT                         |  STARTED   |UDLL firmware download initialized. Fi
|19:02:49.871  |DUT_UDLL_FW_DOWNLOAD_START                        |  STARTED   |UDLL firmware download started OK. Fir
|19:03:00.857  |DUT_UDLL_FW_DOWNLOAD_OK                           |   PASS     |UDLL firmware downloaded OK. Firmware
|19:03:00.869  |DUT_UDLL_FW_DOWNLOAD_OK                           |   PASS     |UDLL firmware downloaded OK. Firmware
|19:03:01.316  |DUT_PDLL_COM_PORT_INIT                            |  STARTED   |Device pdll COM port open initialized.
|19:03:01.368  |DUT_PDLL_COM_PORT_START                           |  STARTED   |Device pdll COM port open started.
|19:03:01.403  |DUT_PDLL_COM_PORT_OK                              |   PASS     |Device pdll COM port opened OK.
|19:03:01.461  |DUT_PDLL_FW_VERSION_GET_START                     |  STARTED   |Device pdll Firmware version get start
|19:03:01.492  |DUT_PDLL_FW_VERSION_GET_OK                        |   PASS     |Device pdll Firmware version get OK. P
|19:03:01.499  |DUT_PDLL_XTAL_TRIM_INIT                           |  STARTED   |XTAL trim operation initialized.
|19:03:01.540  |DUT_PDLL_XTAL_TRIM_START                          |  STARTED   |XTAL trim operation started.
|19:03:03.143  |DUT_PDLL_XTAL_TRIM_OK                             |   PASS     |XTAL trim operation ended OK.
|19:03:03.228  |DUT_PDLL_UART_RESYNC_INIT                         |  STARTED   |UART resync process initialized.
|19:03:03.271  |DUT_PDLL_UART_RESYNC_START                        |  STARTED   |UART resync process started.
|19:03:03.299  |DUT_PDLL_UART_RESYNC_OK                           |   PASS     |UART resync process OK.
|19:03:03.312  |DUT_PDLL_XTAL_TRIM_READ_INIT                      |  STARTED   |XTAL trim value read initialized.
|19:03:03.361  |DUT_PDLL_XTAL_TRIM_READ_START                     |  STARTED   |XTAL trim value read started.
|19:03:03.398  |DUT_PDLL_XTAL_TRIM_READ_OK                        |   PASS     |XTAL trim value read OK. Value is=[116
|19:03:03.413  |DUT_PDLL_PKT_RX_STATS_START_INIT                  |  STARTED   |RF RX packet test with statistics star
|19:03:03.460  |DUT_PDLL_PKT_RX_STATS_START                       |  STARTED   |RF RX packet test with statistics star
|19:03:03.489  |DUT_PDLL_PKT_RX_STATS_STARTED_OK                  |   PASS     |RF RX packet test with statistics star
|19:03:03.824  |DUT_PDLL_PKT_RX_STATS_STOP_INIT                   |  STARTED   |RF RX packet test with statistics stop
|19:03:03.873  |DUT_PDLL_PKT_RX_STATS_STOP_START                  |  STARTED   |RF RX packet test with statistics stop.
|19:03:03.908  |DUT_PDLL_PKT_RX_STATS_STOPPED_OK                  |   PASS     |RF RX packet test with statistics stop
|19:03:03.915  |DUT_PDLL_GU_RF_RX_TEST_PASSED                     |   PASS     |Golden Unit RF RX packet test PASSED.
|19:03:03.922  |DUT_PDLL_PKT_RX_STATS_START_INIT                  |  STARTED   |RF RX packet test with statistics star
|19:03:03.971  |DUT_PDLL_PKT_RX_STATS_START                       |  STARTED   |RF RX packet test with statistics star
|19:03:03.999  |DUT_PDLL_PKT_RX_STATS_STARTED_OK                  |   PASS     |RF RX packet test with statistics star
|19:03:04.336  |DUT_PDLL_PKT_RX_STATS_STOP_INIT                   |  STARTED   |RF RX packet test with statistics stop
```

**Figure 121: DUT Log File**

Figure 121 shows a Log file generated for DUT1 during testing.

In the first few lines of the log, a header is created giving vital information about the PLT hardware and the software. It includes the firmware and software version, the station name and test dates and times. It also holds information about the DUT, such as the connector number in the PLT hardware, the BD address assigned to it and the Windows COM port. For the DUTs that have failed, the log file is renamed with the word "_FAILED" at the end for easier retrieval.

The Log file is created at the beginning of each test, containing only the header and all information available at the time of creation. As the device testing progresses, the status of each test is written at the end of the log file, including information about the DUT and a timestamp of the event. After the tests finish the header is updated with the end time of the test and the firmware versions, which were retrieved during testing.

 © 2020 Dialog Semiconductor

### 7.3.6 CSV File

| Start time | End time | DUT | BD addres | Overall st | COM port | FW downl | FW path 2 | FW versio | FW versio | QSPI eras | QSPI chec | QSPI burn | QSPI imag | FW do |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 17:08:53 | 17:10:43 | 1 | 00:00:00:0 | FAIL | 25 | PASS | C:\Users\ | PASS | 00.03 | PASS | FAIL | | | |
| 17:08:53 | 17:10:43 | 2 | 00:00:00:0 | FAIL | 26 | PASS | C:\Users\ | PASS | 00.03 | PASS | FAIL | | | |
| 17:10:48 | 17:13:24 | 1 | 00:00:00:0 | FAIL | 25 | PASS | C:\Users\ | PASS | 00.03 | PASS | FAIL | | | |
| 17:10:48 | 17:13:24 | 2 | 00:00:00:0 | FAIL | 26 | PASS | C:\Users\ | PASS | 00.03 | PASS | PASS | FAIL | binaries\\pxp_r | |
| 17:13:30 | 17:16:39 | 1 | 00:00:00:0 | FAIL | 25 | PASS | C:\Users\ | PASS | 00.03 | PASS | FAIL | | | |
| 17:13:30 | 17:16:39 | 2 | 00:00:00:0 | FAIL | 26 | PASS | C:\Users\ | PASS | 00.03 | PASS | PASS | FAIL | binaries\\pxp_r | |
| 17:17:00 | 17:17:10 | 1 | 00:00:00:0 | FAIL | 25 | FAIL | C:\Users\pdimopou\Desktop\DA1458x_DA1468x_PLT_v_4.0\source\productior | | | | | | | |
| 17:17:00 | 17:17:10 | 2 | 00:00:00:0 | FAIL | 26 | PASS | C:\Users\ | FAIL | | | | | | |
| 17:17:55 | 17:18:13 | 1 | 00:00:00:0 | FAIL | 25 | FAIL | C:\Users\pdimopou\Desktop\DA1458x_DA1468x_PLT_v_4.0\source\productior | | | | | | | |
| 17:17:55 | 17:18:13 | 2 | 00:00:00:0 | FAIL | 26 | PASS | C:\Users\ | FAIL | | | | | | |
| 17:19:56 | 17:20:22 | 1 | 00:00:00:0 | FAIL | 25 | PASS | C:\Users\ | PASS | 00.03 | PASS | FAIL | | | |
| 17:19:56 | 17:20:22 | 2 | 00:00:00:0 | PASS | 26 | PASS | C:\Users\ | PASS | 00.03 | PASS | PASS | PASS | binaries\\pxp_r | |
| 17:21:39 | 17:22:16 | 1 | 00:00:00:0 | FAIL | 25 | PASS | C:\Users\ | PASS | 00.03 | PASS | FAIL | | | |
| 17:21:39 | 17:22:16 | 2 | 00:00:00:0 | FAIL | 26 | PASS | C:\Users\ | PASS | 00.03 | PASS | FAIL | | | |
| 17:22:18 | 17:25:41 | 1 | 00:00:00:0 | FAIL | 25 | PASS | C:\Users\ | PASS | 00.03 | PASS | FAIL | | | |
| 17:22:18 | 17:25:41 | 2 | 00:00:00:0 | FAIL | 26 | PASS | C:\Users\ | PASS | 00.03 | PASS | FAIL | | | |
| 17:25:47 | 17:26:08 | 1 | 00:00:00:0 | FAIL | 25 | PASS | C:\Users\ | PASS | 00.03 | PASS | FAIL | | | |
| 17:25:47 | 17:26:08 | 2 | 00:00:00:0 | PASS | 26 | PASS | C:\Users\ | PASS | 00.03 | PASS | PASS | PASS | binaries\\pxp_r | |
| 18:57:45 | 18:58:48 | 1 | 00:00:00:0 | FAIL | 25 | PASS | C:\Users\ | PASS | 00.03 | PASS | FAIL | | | PASS |
| 18:57:45 | 18:58:48 | 2 | 00:00:00:0 | PASS | 26 | PASS | C:\Users\ | PASS | 00.03 | PASS | PASS | PASS | binaries\\ | PASS |
| 18:57:45 | 18:58:48 | 3 | 00:00:00:0 | PASS | 27 | PASS | C:\Users\ | PASS | 00.03 | PASS | PASS | PASS | binaries\\ | PASS |
| 18:57:45 | 18:58:48 | 4 | 00:00:00:0 | PASS | 28 | PASS | C:\Users\ | PASS | 00.03 | PASS | PASS | PASS | binaries\\ | PASS |
| 18:59:40 | 19:00:20 | 1 | 00:00:00:0 | FAIL | 25 | PASS | C:\Users\ | PASS | 00.03 | PASS | FAIL | | | PASS |
| 18:59:40 | 19:00:20 | 2 | 00:00:00:0 | PASS | 26 | PASS | C:\Users\ | PASS | 00.03 | PASS | PASS | PASS | binaries\\ | PASS |

**Figure 122: CSV File**

Figure 122 shows an example of a generated CSV file. As with the DUT Log File, the PLT software and hardware information are shown along with valuable DUT information. The CSV file keeps information about all the production tests of a single day. A new CSV file will be created every day.


© 2020 Dialog Semiconductor

## 7.4 CLI PLT Application

The CLI PLT (`SmartBond_CLI_PLT.exe`) is a Command Line Interface application with similar functionality and features as the GUI PLT. It performs the device testing and memory programming. At the same time, it allows users to monitor the test procedure in detail. It supports the same configuration file created from the CFG PLT and can run the same tests as the GUI PLT.

**Note:** Barcode scanner mode (described in Scan Mode, Custom Memory Data, and Custom Memory Data) is only available with GUI PLT Application. CLI PLT Application does **NOT** support this feature.

Figure 123 shows the initial screen of the CLI PLT software. The CLI PLT can be executed from a command line prompt, passing arguments externally and initiating the tests immediately. This is useful for scripting/batch files as shown in section 7.4.3.

Parameters are automatically loaded from the `params/params.xml` file when the CLI PLT starts. If there is a parameter error, a warning will be shown. It is recommended to run the '`x`' command or start the CFG PLT before running the tests and check the configuration parameters. If a change is made to the `params.xml` configuration file while CLI is open, the file should be reloaded using the '`i`' command. If any OTP burning test is scheduled, a message will inform the user and prompt for continuing.

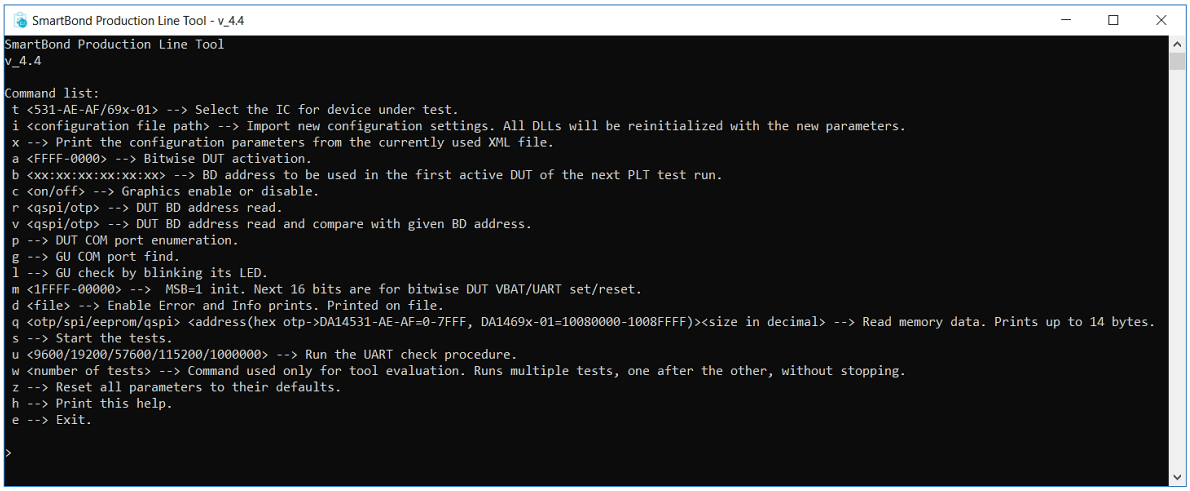



**Figure 123: CLI Software Start Screen**

### 7.4.1 CLI Commands

Table 102 lists all available CLI commands. A list with brief description of these commands can be printed using the '`h`' command.

**Table 102: CLI Commands**

| Cmd | Arguments | Description |
|---|---|---|
| `t` | • 531-AE-AF<br>• 69x-01 | Selects the type of IC that the DUT uses.<br>This option will change the DUT IC setting in the configuration file and reload all the settings if there is a switch from any DA14531 IC to a DA1469x IC and vice versa.<br>**Example:** "`t 69x-01`". |
| `i` | Path to XML configuration file. | Initializes the PLT with the parameters found in the `params.xml` file.<br>**Example:** "`i params\params.xml`". |
| `x` | *none* | Print the configuration parameters from the currently used XML file. |


© 2020 Dialog Semiconductor

| Cmd | Arguments | Description |
|---|---|---|
| a | Hex values from "FFFF" to "0000". | Bitwise DUT activation. Sets the active DUTs to be tested.<br>**Examples:**<br>● "a 1": Only DUT 1 will be activated and tested.<br>● "a 9": DUTs 1 and DUT 4 will be activated and tested. |
| b | xx:xx:xx:xx:xx:xx | Sets the start BD address of the first active DUT.<br>**Example:** "b FE:00:11:22:33:44" |
| c | on/off | Enables or disables the graphics debug output of the CLI. Useful in the read BD address command *r*, in order to see only the DUT BD address returned and not the entire process.<br>**Example:** "c on" |
| r | ● qspi<br>● otp | Reads the BD addresses of the active DUTs. It is better to disable the graphics beforehand by running the command "c off".<br>**Example:** "r qspi" |
| v | ● qspi<br>● otp | Verify the BD addresses of the active DUTs. It is better to disable the graphics beforehand by running the command "c off".<br>**Example:** "v qspi" |
| p | *none* | Execute the automatic DUT Window COM port enumeration. |
| g | *none* | Execute the automatic GU Window COM port enumeration. |
| l | *none* | Run the GU sanity check. The Golden Unit will start blinking its red LED. |
| m | First character: "1" or "0". Then hex value from "1FFFF" to "0000". | MSB character should be '1' the first time this command is executed. Consecutive 'm' commands should have the MSB character set to '0'. The next 16 bits are used for bitwise DUT VBAT/UART enable/disable.<br>**Example:** "m 1FFFF" |
| d | ● console<br>● file | Use this option to enable error and info prints. Choose file output or console output. Only the *file* option is currently supported.<br>**Example:** "d file" |
| q | First argument<br>● otp<br>● spi<br>● eeprom<br>● qspi<br>Second argument<br>● The address in hex<br>Third argument<br>● The size in bytes | This option can read from any memory, for any address offset up to 256 bytes of data. |
| s | *none* | Starts the tests. |
| u | ● 9600<br>● 19200<br>● 57600<br>● 115200<br>● 1000000 | This command performs a UART check connection for all the active DUTs for a specific Baud rate. |
| w | Number of tests to run. | This command is used only for PLT evaluation. It starts multiple tests. These are executed one after the other without user intervention. |
| z | *none* | Resets all the XML parameters to their defaults. |
| h | *none* | Help print out. Prints the list of the CLI commands that are available. |

 © 2020 Dialog Semiconductor

| Cmd | Arguments | Description |
|---|---|---|
| e | *none* | Exits the application. |

### 7.4.2 Running the CLI and Executing Tests

There are a number of options to be called in order to make sure that the CLI PLT is set up correctly. Each of following commands is explained in Table 102.

- Using the help command ('h') the entire CLI command list will be shown.

  Example: `>h`

#### Set Console Options

- To redirect the debugging messages to the file use command 'd'. This option is going to replace the UI debug values in the configuration file.

  Example: `>d file`

- To show or hide any prints in the Console window use command 'c' with on/off argument.

  Example: `>c on`

#### Check, Reset, Reload and Change Settings

- Because the configuration file is automatically loaded, use the 'x' command (Figure 124) to see the loaded settings. Errors will be shown in red.

  Example: `>x`

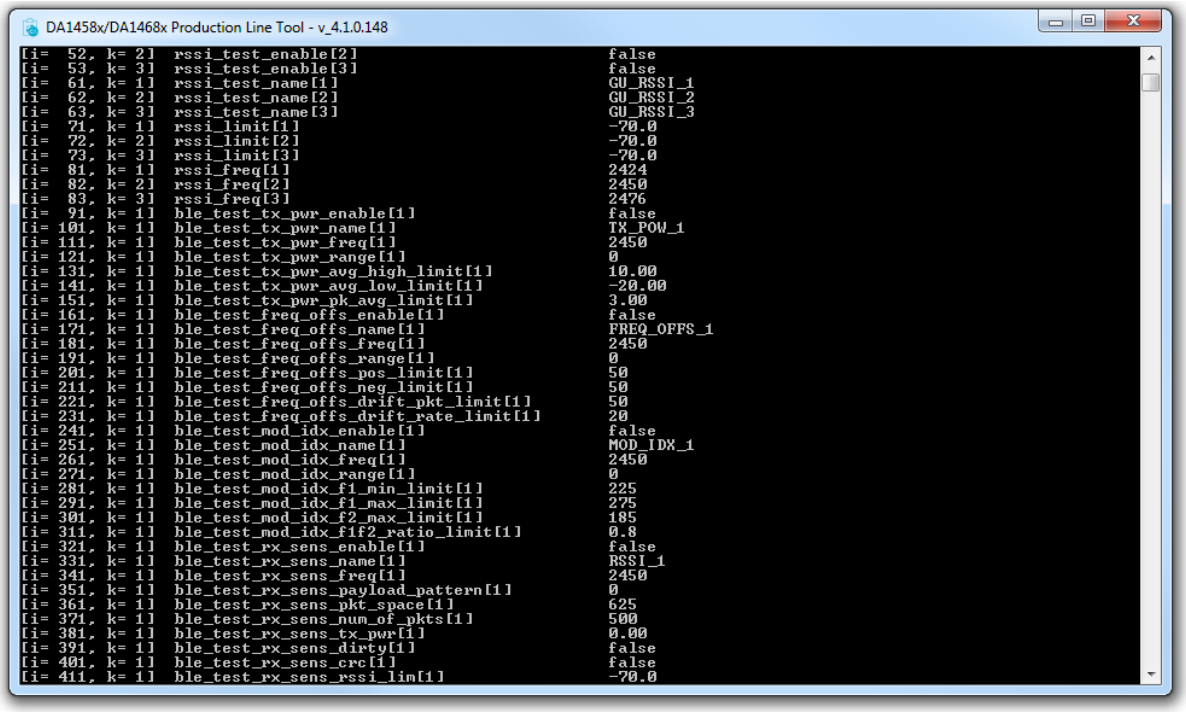



**Figure 124: CLI PLT Print Settings (x Command)**

- To reset the configuration parameters to their defaults values the 'z' command should be used.
- To reload the configuration file or to load another one, 'i' command can be used.

  Example: `>i params/params.xml`


© 2020 Dialog Semiconductor

- To change only the selected device IC, use the ‘`t xxx`’ command, where ‘`xxx`’ is the desired IC selection. If a change from DA14531 to DA1469x (and vice versa) is made, all the settings will be reloaded.

  Example: `>t 69x-01`

- To change the active DUTs use the '`a`' command. As an argument a 16-bit hexadecimal value is used, which is the bitwise representation of the active DUTs with DUT 1 being the LSB. This command will replace the `dut_num_x` values in the configuration file.

  The following example will enable only DUT1, DUT2, DUT15 and DUT 16.

  Example: `>a C003`

- To change the BD address that will be used in the next run use the '`b`' command.

  This option is going to replace the *BD address* and *Statistics* values in the configuration file.

  Example: `>b 00:00:00:00:00:01`

### Hardware Specific Tests

- To automatically find the Windows COM Port assigned to the Golden Unit, use the '`g`' command. This command will replace the `gu_com_port` value in the configuration file.

  Example: `>g`

- To verify that the Golden Unit COM port is found correctly and to check if the Golden Unit is ready run the '`l`' command.

  Example: `>l`

- To automatically find the Windows COM Port assigned for each DUT, use the '`p`' command. This command will replace the `com_port_dut_x` values in the configuration file.

  Example: `>p`

```
DUT  BD ADDRESS          CODE  STATUS                                RESULT
2    00:00:55:00:00:01   0     NOT ACTIVE
3    00:00:55:00:00:02   2     COM PORT IDENTIFY STARTED
4    00:00:55:00:00:03   3     COM PORT IDENTIFY OK                  PASS
GU   COM4                34    RD TESTER COM LOOPBACK OK             OK
```

**Figure 125: CLI PLT DUT COM Port Enumeration ('p' Command)**

- To run a UART error check use the '`u`' command followed with a specific Baud rate.

  Example: `>u 1000000`

### PLT Production Tests

- To check if there is a BD address written in the active DUTs use the '`r`' command. To read the BD addresses and verify that they are the same as the ones currently assigned use the '`v`' command. Both commands use '`qspi`' or '`otp`' as argument to define the memory header.

  The following example will read the BD address from the QSPI memory header and compare it with the one currently assigned to the DUTs.

  Example: `>v qspi`

```
DUT  BD ADDRESS          CODE  STATUS                                RESULT
2    00:00:55:00:00:01   177   QSPI BD ADDRESS COMPARED OK           PASS
3    00:00:55:00:00:02   177   QSPI BD ADDRESS COMPARED OK           PASS
4    00:00:55:00:00:03   177   QSPI BD ADDRESS COMPARED OK           PASS
GU   COM4                26    RD TESTER INIT OK                     OK

DUT  QSPI BD ADDRESS             GIVEN BD ADDRESS                    STATUS
02   00:00:55:00:00:01           00:00:55:00:00:01                   MATCH
03   00:00:55:00:00:02           00:00:55:00:00:02                   MATCH
04   00:00:55:00:00:03           00:00:55:00:00:03                   MATCH

return status = 0xFFF1
```

**Figure 126: CLI PLT Read and Compare BD Address in QSPI ('v' Command)**

 © 2020 Dialog Semiconductor

- Use the 's' command to begin testing with the current configuration. Figure 127 shows the CLI during the testing. After all the tests have finished, the result remains on the screen as shown in Figure 128.

```
DUT  BD ADDRESS          CODE  STATUS                                RESULT
2    00:00:55:00:00:01   9     FW DOWNLOAD STARTED
3    00:00:55:00:00:02   9     FW DOWNLOAD STARTED
4    00:00:55:00:00:03   9     FW DOWNLOAD STARTED
GU   COM4                38    RD TESTER VBAT ENABLE OK              OK

     BLE TESTER   TEMP      VOLT
      NOT USED   NOT USED  NOT USED
```

**Figure 127: CLI PLT Testing**

```
DUT  BD ADDRESS          CODE  STATUS                                RESULT
2    00:00:55:00:00:01   216   SCAN TEST OK - DUT FOUND              PASS
3    00:00:55:00:00:02   216   SCAN TEST OK - DUT FOUND              PASS
4    00:00:55:00:00:03   216   SCAN TEST OK - DUT FOUND              PASS
GU   COM4                26    RD TESTER INIT OK                     OK

     BLE TESTER   TEMP      VOLT
      NOT USED   NOT USED  NOT USED

DUT  QSPI BD ADDRESS          GIVEN BD ADDRESS                       STATUS
02   00:00:55:00:00:01        00:00:55:00:00:01                      MATCH
03   00:00:55:00:00:02        00:00:55:00:00:02                      MATCH
04   00:00:55:00:00:03        00:00:55:00:00:03                      MATCH

return status = 0xFFF1
```

**Figure 128: CLI PLT Testing Finished**

- Use the 'q' command to read from any memory up to 156 bytes of data. The following example will read the BD address (6 bytes from offset 0xFF0) of a DA14531s DUT.
  Example: `>q otp ff0 6`

### Other Test Commands

- Use the 'm' command to power on and access the DUTs to perform further testing. This will open the VBAT and the COM ports from the PLT hardware to the DUTs. As an argument, a '0' or '1' character is used to reset the PLT hardware. This is followed by a 16-bit hexadecimal number, which is the bitwise representation of the DUTs to use, with DUT 1 being the LSB.

  In the following example, the PLT hardware will be reset and DUTs 1, 2, 15 and 16 will be used.

  Example: `>m 1C003`

 © 2020 Dialog Semiconductor

### 7.4.3 Using CLI Commands with Arguments

It is possible to start the CLI program with the commands described in section 7.4.2 as arguments. This is useful for scripting/batch files.

```
C:\executables>DA1458x_DA1468x_CLI_PLT.exe -a 0001 -b 00:00:55:00:00:01 -s -b 00:00:55:00:00:01 -v qspi
```

**Figure 129: CLI with Commands as Arguments**

The example shown in Figure 129 will perform the following commands:

3. '`-a 0001`': Set the DUT1 as the only active DUT.
4. '`-b 00:00:55:00:00:01`'. Assign as the first BD address to be assigned the 00:00:55:00:00:01. This BD address will be used in DUT1, as it is the only active DUT.
5. '`-s`': Perform the tests. BD address write in QSPI header should be enabled for the following test to pass.
6. '`-b 00:00:55:00:00:01`': After the tests finish the BD address will be incremented; this command will reset it to 00:00:55:00:00:01.
7. '`-v qspi`': Read only the BD address written in the QSPI header and compare it with the one assigned to the DUT1. These are the same.

 © 2020 Dialog Semiconductor

## 7.5 GU Upgrade Application

The GU Upgrade (`GU_fw_upgrade.exe`) is a Graphical User Interface application, which can be used to upgrade the firmware of the Golden Unit automatically, in contrast with Golden Unit that describes a manual way to upgrade the firmware of the Golden Unit. It guides the user to configure, detect the PLT hardware and finally reprogram the SPI Flash memory onboard the PLT hardware with the Golden Unit firmware.

**Note:** Quick access to GU upgrade tool is provided by pressing the `Upgrade GU firmware` button, under `Firmware Version` – Golden Unit section in CFG PLT. Current version of the GU firmware can be seen using the `Refresh` button on the same section.

**Note:** This tool cannot upgrade PLT hardware version A. To update PLT hardware version A follow the instructions in Golden Unit .

### Introduction Page

The first page of the tool is an introduction page with the purpose of the tool. User can exit the tool at any step by pressing the `cancel` button or close the application using the `X` button from the windows bar at the top.

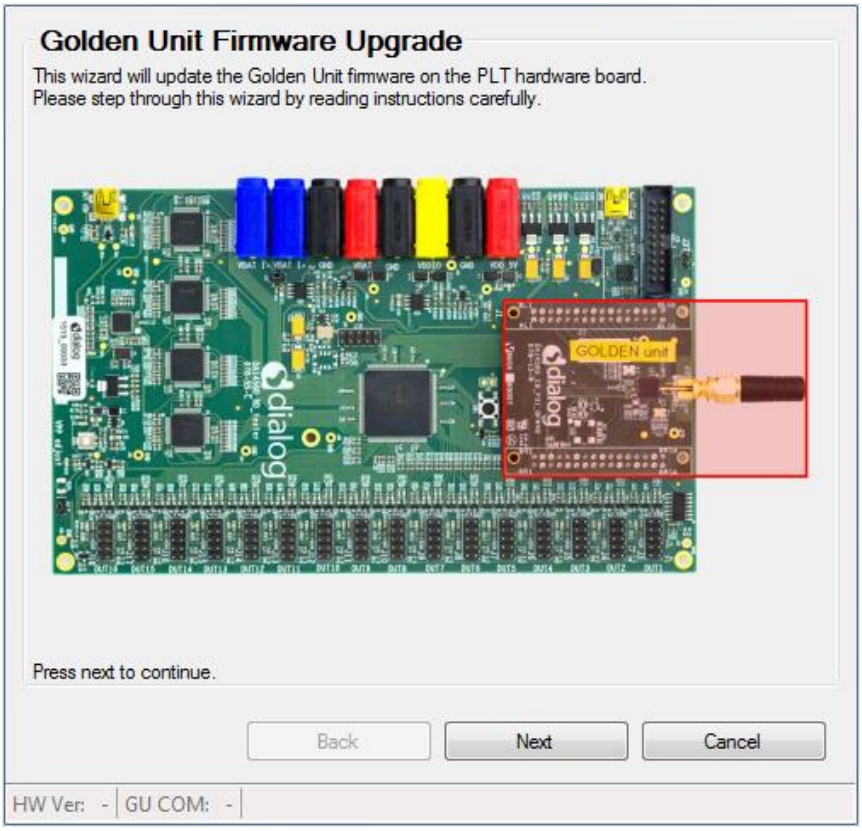



**Figure 130: GU Upgrade - Introduction page**

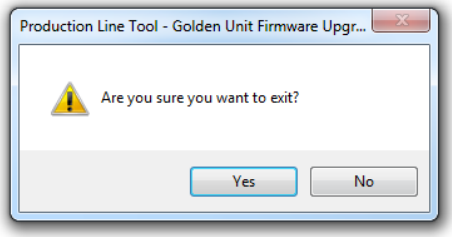



**Figure 131: GU Upgrade - Exit Message**


© 2020 Dialog Semiconductor

### Hardware Version

Select the PLT hardware version. Depending on the version, some options may be missing, or the tool may not support it. Selected hardware version will be shown on the left bottom corner of the tool at any of the following steps. As noted before PLT hardware `version A` is not supported by this tool.

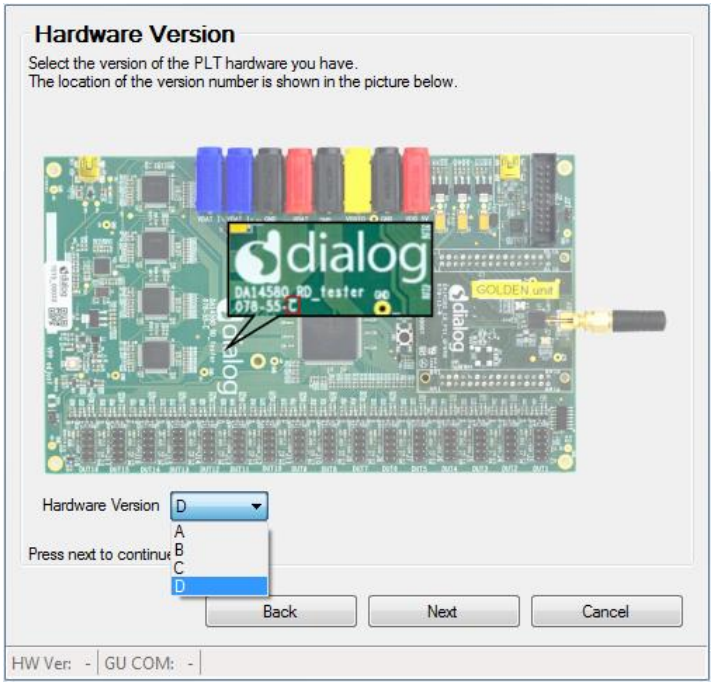



**Figure 132: GU Upgrade - Hardware Version**

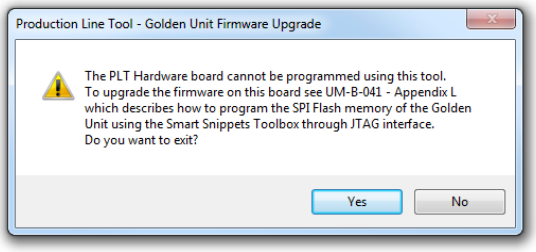



**Figure 133: GU Upgrade - Hardware Version Compatibility**

### Power Supply

Connect the power supply of the PLT, as described in PLT Power Supply and connect the jumpers as shown in Figure 135 where applicable. Adjust the jumpers as proposed by the tool.


© 2020 Dialog Semiconductor

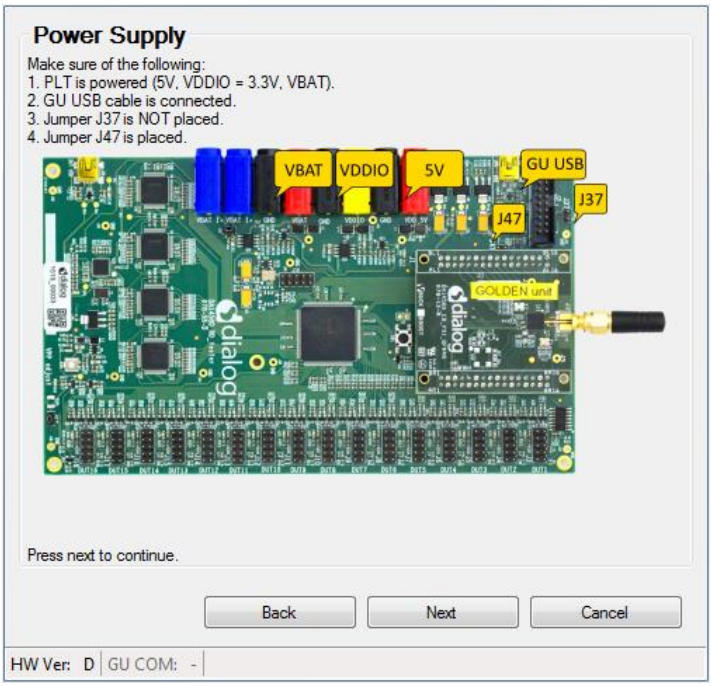



**Figure 134: GU Upgrade - Power Supply**

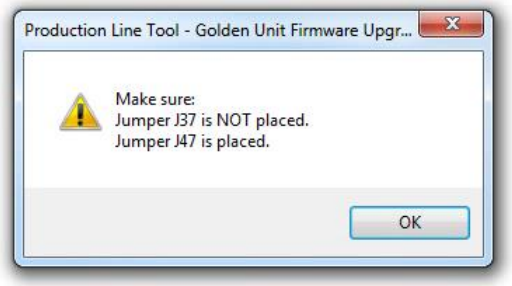



Figure **135**: GU Upgrade - Power Supply Pop-Up

### Golden Unit Reset

Select the way the GU will be reset. User can manually reset the GU, but PLT can do it automatically, which is the default selection. If the manual mode is selected, the user will be prompt any time the Golden Unit must be reset, to press the reset button located next to the Golden Unit on top of the PLT hardware.


© 2020 Dialog Semiconductor

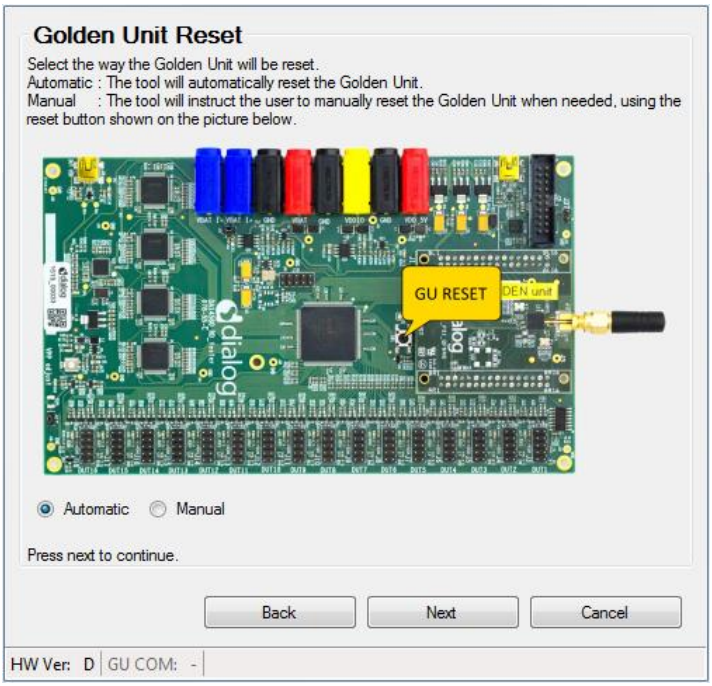



**Figure 136: GU Upgrade - Golden Unit Reset**

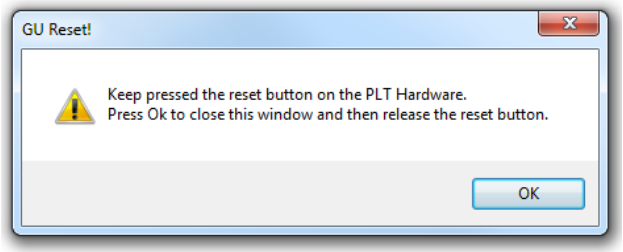



**Figure 137: GU Upgrade - Golden Unit Reset Message for Manual Mode**

### GU COM Port

Find the windows assigned GU COM port. User can either select it from the dropdown list or use the *Auto* button to find it using the serial number as described in Automatic GU COM Port Find. GU COM port can be also verified using the *Check* button. Selected GU COM port will be shown on the left bottom corner of the tool.


© 2020 Dialog Semiconductor

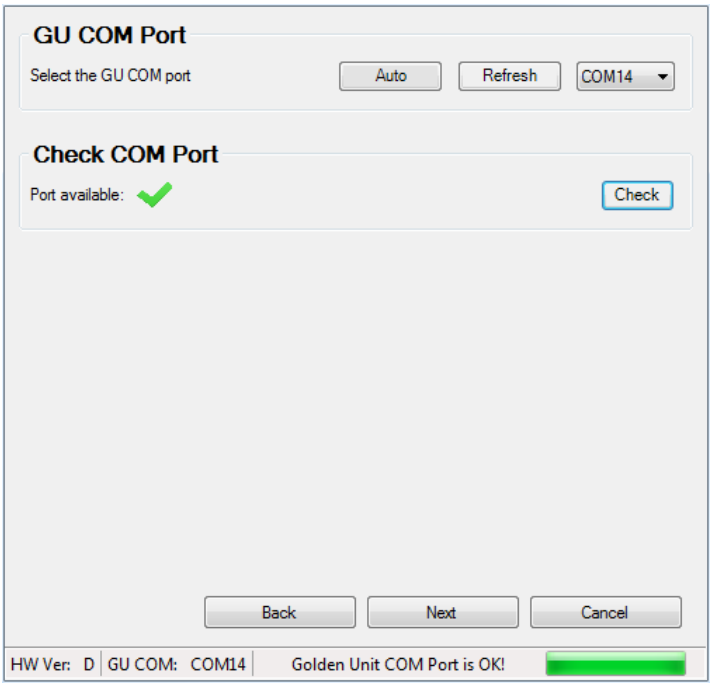



**Figure 138: GU Upgrade - GU COM Port**

### Burn Firmware

This is the final step. Select the binary to burn.

Pressing the **Burn** button will erase the SPI Flash on the PLT hardware, program it with the new firmware selected before, and then read it back to verify that the contents written are the same as those in the binary.

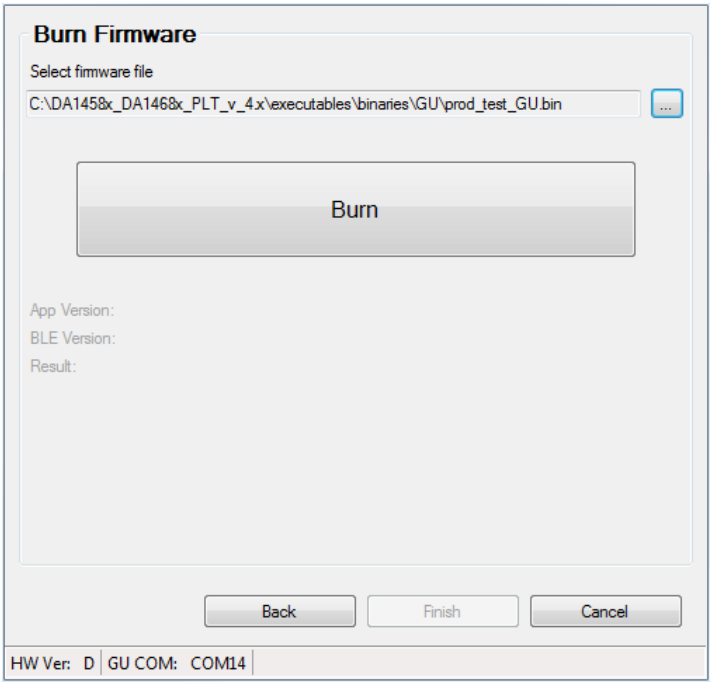



**Figure 139: GU Upgrade - Burn Firmware**


© 2020 Dialog Semiconductor

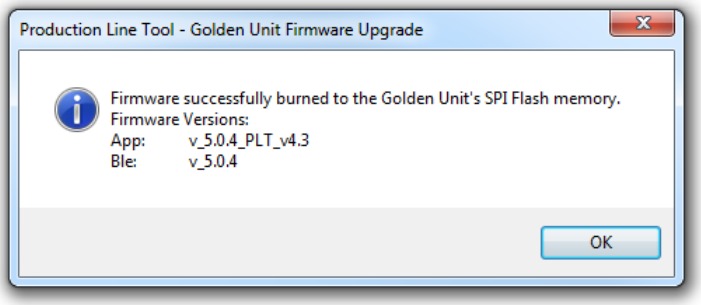



**Figure 140: GU Upgrade - Burn Firmware Pop-Up Message**

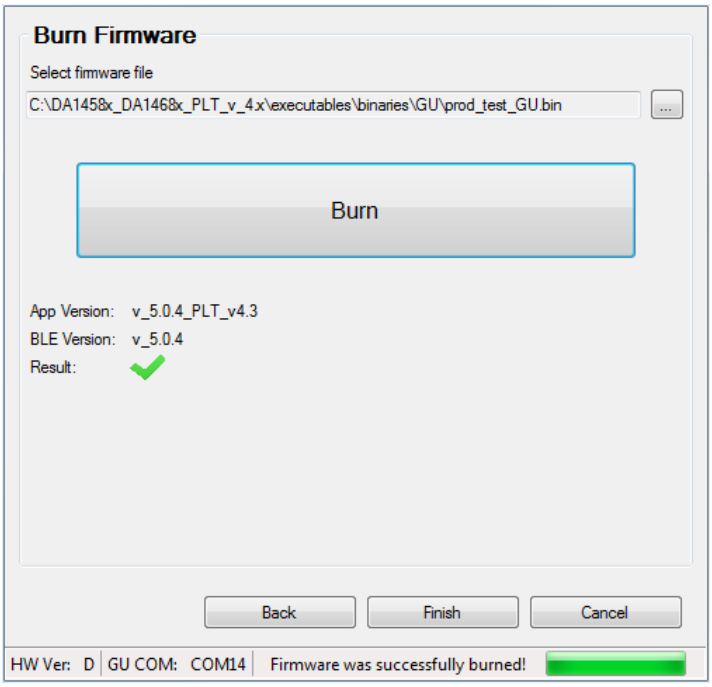



**Figure 141: GU Upgrade - Burn Firmware Success**

After the SPI flash program procedure is finished, a pop-up message will appear with the result of the programming procedure. If the SPI flash was programmed successfully, the pop-up message will also show the version of the new Golden Unit firmware (Figure 140).


© 2020 Dialog Semiconductor

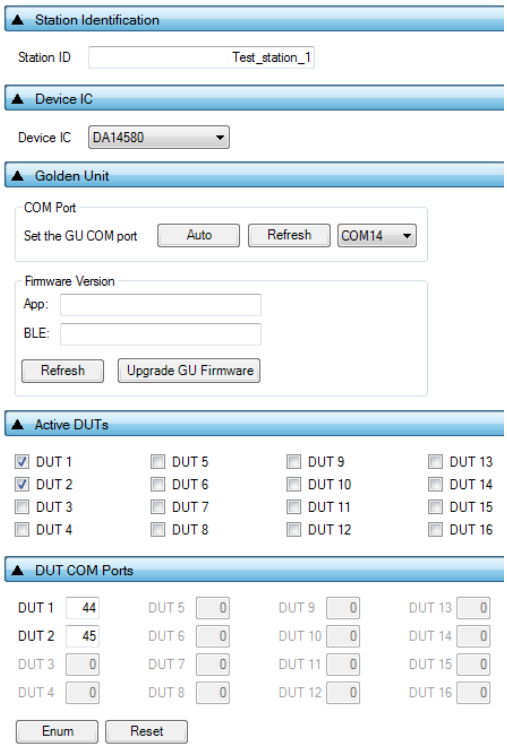
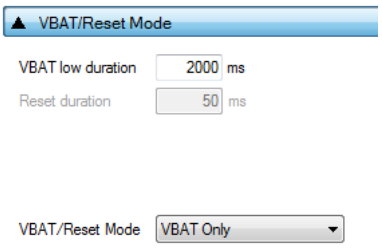
# 8 Example Usage

In this section, a simple example of the PLT will be described using two DA14580 devices. The example test procedure is explained step-by-step in Table 103.

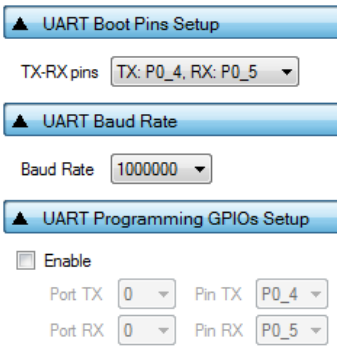
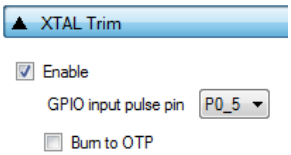
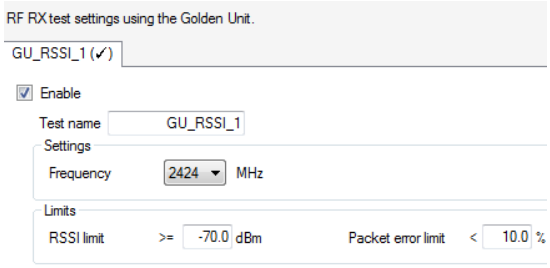
The tests to run in this example are the XTAL trim, RF RSSI test, SPI erase, SPI check empty and SPI image write.

Two DUTs will be used at PLT DUT connector positions DUT1 and DUT2.

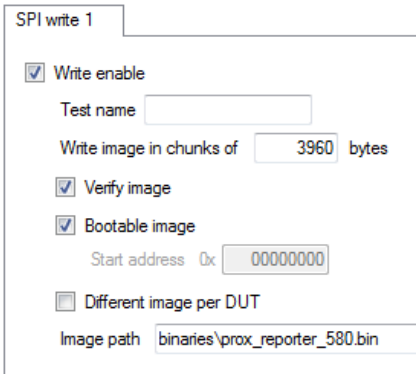
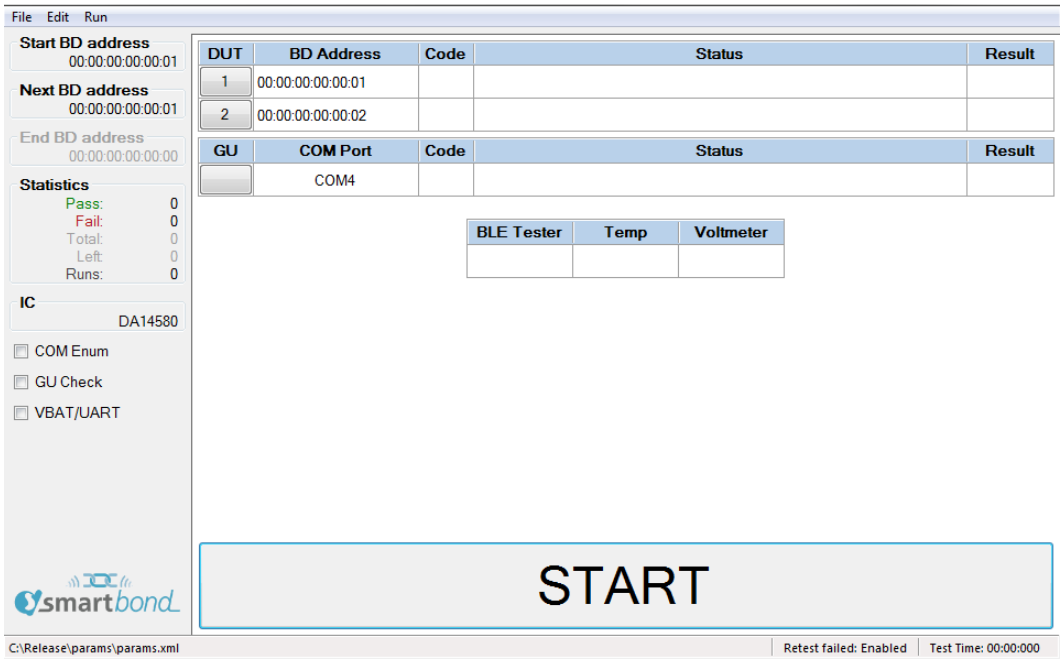
**Table 103: DA14580 PLT Example Usage**

| Step | Description | | |
|---|---|---|---|
| **Hardware Connections** | | | |
| 1 | Connect both DUTs to the PLT hardware as shown in DA1458x DK Pro Motherboard Connection. Four cables are needed. In order to use the SPI Flash memory a custom triple-jumper must be used. | | |
| **CFG GUI Settings** | | | |
| 2 | Open CFG PLT and make the following selections. | | |
| 3 | Hardware Setup | | |
| |  | Device IC | Select DA14580 and then Save. |
| | | Golden Unit COM Port | Press the Auto Button and then Save. |
| | | Active DUTs | Enable only DUT1 and DUT2 and Save. |
| | | DUT COM Ports | Press **Enum** button and then Save. |
| 4 | VBAT/Reset Mode | | |
| |  | VBAT low duration | Set a time for the POR that is sufficient enough for the DUTs to discharge their capacitors during the POR. |
| | | VBAT/Reset Mode | Since the DUTs will be powered from the PLT hardware and the Reset signal will not be used, use the `VBAT only` option. |
| 5 | General | | |
| | | Statistics | Press the Reset Button and then Save. |

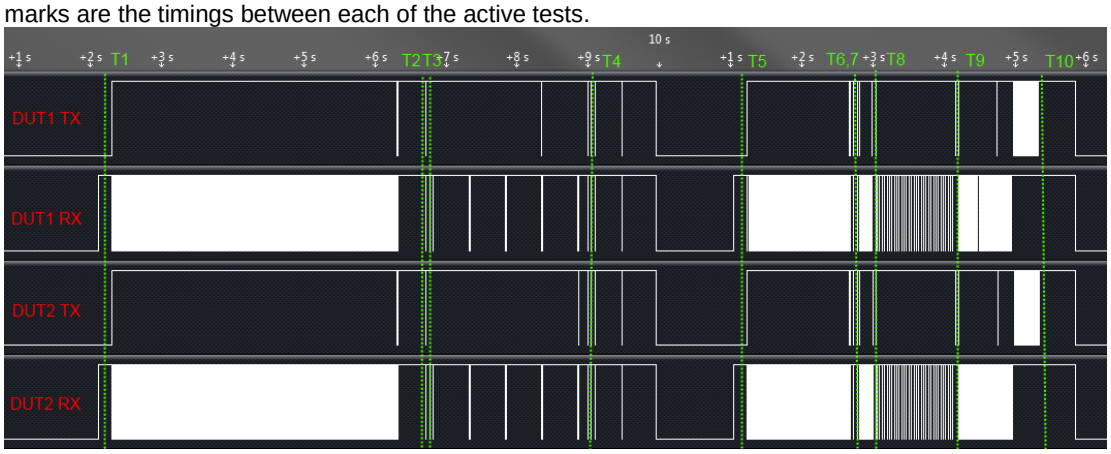
 © 2020 Dialog Semiconductor

| Step | Description | | |
|---|---|---|---|
| | | Test Options | Enable all options and then press Save.<br>Production tests are required to be enabled for the XTAL Trim and the RF tests to run.<br>Memory programming is required for the QSPI erase and check empty functions.<br>Disable the rest of the options. |
| 6 | UART | | |
| |  | UART Boot Pin Setup | TX: P0_4, RX: P0_5 |
| | | UART Baud Rate | 1000000 |
| 7 | Test Settings | | |
| |  | XTAL Trim | Check *Enable* in XTAL Trim. Select the same pin as the *UART Rx*, *P0_5*.<br>*Burn to OTP* is disabled |
| |  | RF Tests - Golden Unit | Only one test is enabled for this example.<br>In Golden Unit:<br>• Check *Enable*<br>• Select 2424 MHz as *frequency*<br>• Set the *RSSI limit* to -70 dBm<br>• Set *Packet error limit* to 10% |

 © 2020 Dialog Semiconductor

| Step | Description | | |
|---|---|---|---|
| 8 | Memory Functions | | |
| |  | Write 1 test | Check Write enable<br>Check Verify image<br>Check Bootable image<br>Make sure `prox_reporter_580.bin` is selected. |
| 9 | OTP Header | | |
| | General | Disable all options | |
| | BD address | Disable all options | |
| **GUI PLT** | | | |
| 10 | Open GUI PLT. | | |
| |  | | |

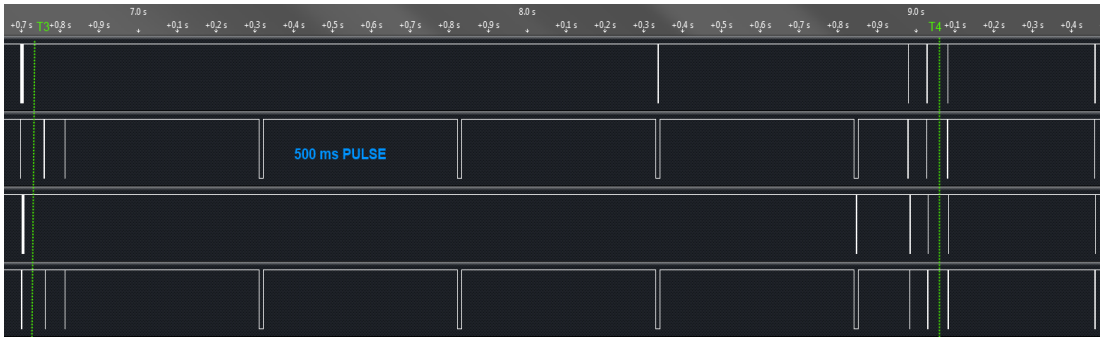
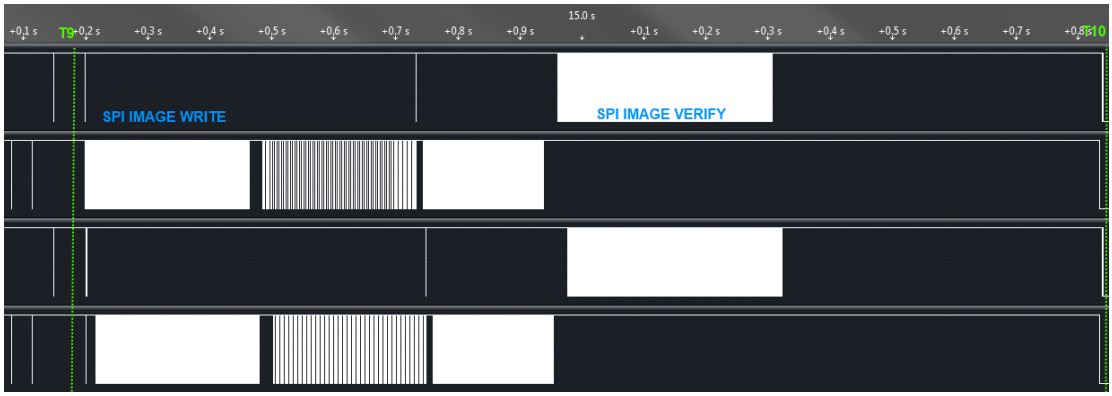
 © 2020 Dialog Semiconductor

| Step | Description |
|---|---|
| 11 | Press Space Bar to begin testing |
| | The following picture shows the UART channels (in red) for both DUTs for the entire PLT run. The green marks are the timings between each of the active tests.   |
| | Log files for both the DUTs are created. The following picture shows the test steps from the log file of DUT1 with the timing marks from the logic analyzer capture. These marks are described below. From the log file, the total time of each test can be calculated. |

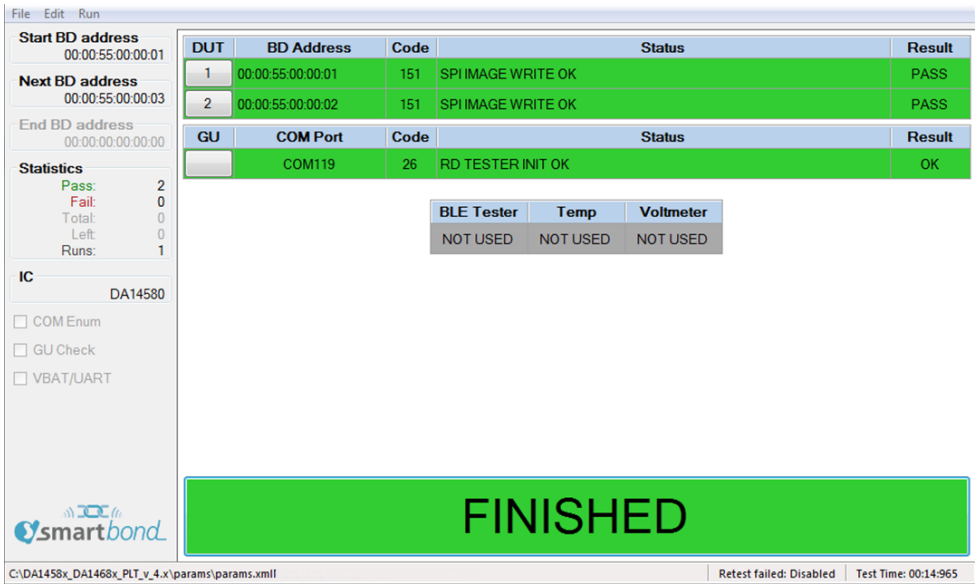
```
|<time>        |<action>                                  |<pass/fail> |<info>
##################################################################################################################
|17:51:11.321  |DUT_UDLL_FW_DOWNLOAD_INIT                 |  STARTED   |UDLL firmware download initialized. Firmwa
|17:51:11.355  |DUT_UDLL_FW_DOWNLOAD_START          T1    |  STARTED   |UDLL firmware download started OK. Firmware
|17:51:15.646  |DUT_UDLL_FW_DOWNLOAD_OK                   |   PASS     |UDLL firmware downloaded OK. Firmware is=[(
|17:51:15.652  |DUT_UDLL_FW_DOWNLOAD_OK                   |   PASS     |UDLL firmware downloaded OK. Firmware is=[(
|17:51:15.873  |DUT_PDLL_COM_PORT_INIT                    |  STARTED   |Device pdll COM port open initialized.
|17:51:15.899  |DUT_PDLL_COM_PORT_START             T2    |  STARTED   |Device pdll COM port open started.
|17:51:15.928  |DUT_PDLL_COM_PORT_OK                      |   PASS     |Device pdll COM port opened OK.
|17:51:15.982  |DUT_PDLL_FW_VERSION_GET_START             |  STARTED   |Device pdll Firmware version get started.
|17:51:16.003  |DUT_PDLL_FW_VERSION_GET_OK                |   PASS     |Device pdll Firmware version get OK. PDLL v
|17:51:16.008  |DUT_PDLL_XTAL_TRIM_INIT                   |  STARTED   |XTAL trim operation initialized.
|17:51:16.028  |DUT_PDLL_XTAL_TRIM_START            T3    |  STARTED   |XTAL trim operation started.
|17:51:18.141  |DUT_PDLL_XTAL_TRIM_OK                     |   PASS     |XTAL trim operation ended OK.
|17:51:18.207  |DUT_PDLL_UART_RESYNC_INIT                 |  STARTED   |UART resync process initialized.
|17:51:18.234  |DUT_PDLL_UART_RESYNC_START                |  STARTED   |UART resync process started.
|17:51:18.249  |DUT_PDLL_UART_RESYNC_OK                   |   PASS     |UART resync process OK.
|17:51:18.259  |DUT_PDLL_XTAL_TRIM_READ_INIT              |  STARTED   |XTAL trim value read initialized.
|17:51:18.283  |DUT_PDLL_XTAL_TRIM_READ_START             |  STARTED   |XTAL trim value read started.
|17:51:18.309  |DUT_PDLL_XTAL_TRIM_READ_OK                |   PASS     |XTAL trim value read OK. Value is=[1428].
|17:51:18.316  |DUT_PDLL_PKT_RX_STATS_START_INIT          |  STARTED   |RF RX packet test with statistics start ini
|17:51:18.342  |DUT_PDLL_PKT_RX_STATS_START         T4    |  STARTED   |RF RX packet test with statistics start.
|17:51:18.367  |DUT_PDLL_PKT_RX_STATS_STARTED_OK          |   PASS     |RF RX packet test with statistics started (
|17:51:18.686  |DUT_PDLL_PKT_RX_STATS_STOP_INIT           |  STARTED   |RF RX packet test with statistics stop init
|17:51:18.712  |DUT_PDLL_PKT_RX_STATS_STOP_START          |  STARTED   |RF RX packet test with statistics stop.
|17:51:18.729  |DUT_PDLL_PKT_RX_STATS_STOPPED_OK          |   PASS     |RF RX packet test with statistics stopped (
|17:51:18.739  |DUT_PDLL_GU_RF_RX_TEST_PASSED             |   PASS     |Golden Unit RF RX packet test PASSED. Freq
|17:51:20.280  |DUT_UDLL_FW_DOWNLOAD_INIT                 |  STARTED   |UDLL firmware download initialized. Firmwa
|17:51:20.319  |DUT_UDLL_FW_DOWNLOAD_START          T5    |  STARTED   |UDLL firmware download started OK. Firmware
|17:51:22.056  |DUT_UDLL_FW_DOWNLOAD_OK                   |   PASS     |UDLL firmware downloaded OK. Firmware is=[(
|17:51:22.065  |DUT_UDLL_FW_DOWNLOAD_OK                   |   PASS     |UDLL firmware downloaded OK. Firmware is=[(
|17:51:22.072  |DUT_UDLL_FW_VER_GET_INIT                  |  STARTED   |UDLL 'firmware version get' operation initi
|17:51:22.092  |DUT_UDLL_FW_VER_GET_STARTED         T6    |  STARTED   |UDLL 'firmware version get' operation start
|17:51:22.106  |DUT_UDLL_FW_VER_GET_OK                    |   PASS     |UDLL 'firmware version get' operation endec
|17:51:22.113  |DUT_UDLL_SPI_ERASE_INIT                   |  STARTED   |SPI erase operation initialized.
|17:51:22.131  |DUT_UDLL_SPI_ERASE_STARTED          T7    |  STARTED   |SPI erase operation started. Erase all SPI
|17:51:22.318  |DUT_UDLL_SPI_ERASE_OK                     |   PASS     |SPI erase operation ended OK. Erase all SP
|17:51:22.335  |DUT_UDLL_SPI_CHECK_EMPTY_INIT             |  STARTED   |SPI check empty operation initialized.
|17:51:22.360  |DUT_UDLL_SPI_CHECK_EMPTY_STARTED    T8    |  STARTED   |SPI check empty operation started, all SPI
|17:51:23.496  |DUT_UDLL_SPI_CHECK_EMPTY_OK               |   PASS     |SPI check empty operation ended OK, all SP
|17:51:23.500  |DUT_UDLL_SPI_IMG_WR_INIT                  |  STARTED   |SPI image write operation initialized. Imag
|17:51:23.524  |DUT_UDLL_SPI_IMG_WR_STARTED         T9    |  STARTED   |SPI image write operation started. Image to
|17:51:24.663  |DUT_UDLL_SPI_IMG_WR_OK                    |   PASS     |SPI image write operation ended OK. Image i
                                                   T10
```

| | |
|---|---|
| T1 | The firmware download (`prod_test_580.bin`) begins. From the log file, the test lasted for about 4.3 s, which can be verified from the logic analyzer capture. |
| T2 | PLT begins the operation to get the version of the `prod_test_580.bin` firmware. |


© 2020 Dialog Semiconductor

| Step | Description | |
|---|---|---|
| 11 cont. |  | |
| | T3 | After the firmware version was acquired the XTAL Trim test begins. The 500 ms reference pulse in the DUT RX channel is shown with blue letters. This test lasted for 2.3 s including the UART resync. |
| | T4 | The RF RX RSSI test begins. |
| | | |
| | T5 | The firmware download (`flash_programmer_58x.bin`) begins. Checking the log file, the test lasted about 1.8 s. This can also be verified by looking the timing difference between T5-T4 at the first oscilloscope capture. |
| | T6 | PLT begins the operation to get the version of the `flash_programmer.bin` firmware. |
| | T7 | The SPI erase action begins. As with any SPI operation, first, a command to set the SPI bus pins is used and then the erase command. The blocks of data after the SPI erase and before the SPI check empty are SPI traffic on P0_5, which is the UART RX and the SPI MISO pin at the same time. |
| | T8 | The SPI check empty action begins. As with any SPI operation, first, a command to set the SPI bus pins is used and then the check empty command. The blocks of data after the SPI check empty and before the SPI image write are SPI traffic on P0_5, which is the UART RX and the SPI MISO pin at the same time. |
| |  | |
| 11 cont. | T9 | The SPI image write operation begins. First, the image will be written, as shown in the picture above, and then the contents are read back for verification. The blocks of data after the SPI image write and before the SPI image verification are SPI traffic on P0_5, which is the UART RX and the SPI MISO pin at the same time. |


© 2020 Dialog Semiconductor

| Step | Description |
|---|---|
| T10 | The PLT run has finished.  |
| | After the tests have finished, the log and CSV files are fully updated. The GUI PLT test counter on the bottom right of the screen indicates that the test took approximately 15 s. This is verified from both the log files and the CSV file. Both files are shown below. |
| | Software: DA1458x/DA1468x Production Line<br>Software version: v_4.1.0.132<br>PLTD DLL version: v_4.1.0.132<br>PDLL Version: v_4.1.0.132<br>UDLL Version: v_4.1.0.132<br>Production test BLE firmware version: v_5.0.4<br>Production test APP firmware version: v_5.0.4_PLT_v4.1<br>Flash programmer firmware version: v_5.0.4_PLT_v4.1<br>Date: 2017-08-11<br>Start Time: 17:32:31.403<br>End Time: 17:32:49.038<br>Station ID: Test_station_1<br>Device ID: 1<br>COM port: 184<br>BD address: 00:00:55:00:00:01 |
| 11 cont. | |

| | A | B | C | D | E | F T1 | G |
|---|---|---|---|---|---|---|---|
| 1 | Start time | End time | DUT | BD address | Overall status | COM port | FW download 1 |
| 2 | 18:54:14 | 18:54:29 | 1 | 00:00:55:00:00:01 | PASS | 25 | PASS |
| 3 | 18:54:14 | 18:54:29 | 2 | 00:00:55:00:00:02 | PASS | 26 | PASS |

| H T2 | I | J T3 | K | L T4 | M |
|---|---|---|---|---|---|
| FW path 1 | FW version get 1 | FW version 1 | XTAL trim test | XTAL trim | GU RX test 1 |
| C:\Users\pd | PASS | v_5.0.4 | PASS | 1426 | PASS |
| C:\Users\pd | PASS | v_5.0.4 | PASS | 1346 | PASS |

| N T5 | O | P T6 | Q |
|---|---|---|---|
| GU RX RSSI 1 | FW download 2 | FW path 2 | FW version get 2 |
| -31.82 | PASS | C:\Users\pd | PASS |
| -21.87 | PASS | C:\Users\pd | PASS |

| R T7 | S T8 | T T9 | U | V T10 |
|---|---|---|---|---|
| FW version 2 | SPI erase 1 | SPI empty 1 | SPI burn 1 | SPI image 1 |
| v_3.0.11.554 | PASS | PASS | PASS | binaries\prox_reporte |
| v_3.0.11.554 | PASS | PASS | PASS | binaries\prox_reporte |


© 2020 Dialog Semiconductor

# Appendix A Top View of PLT PCB Version D

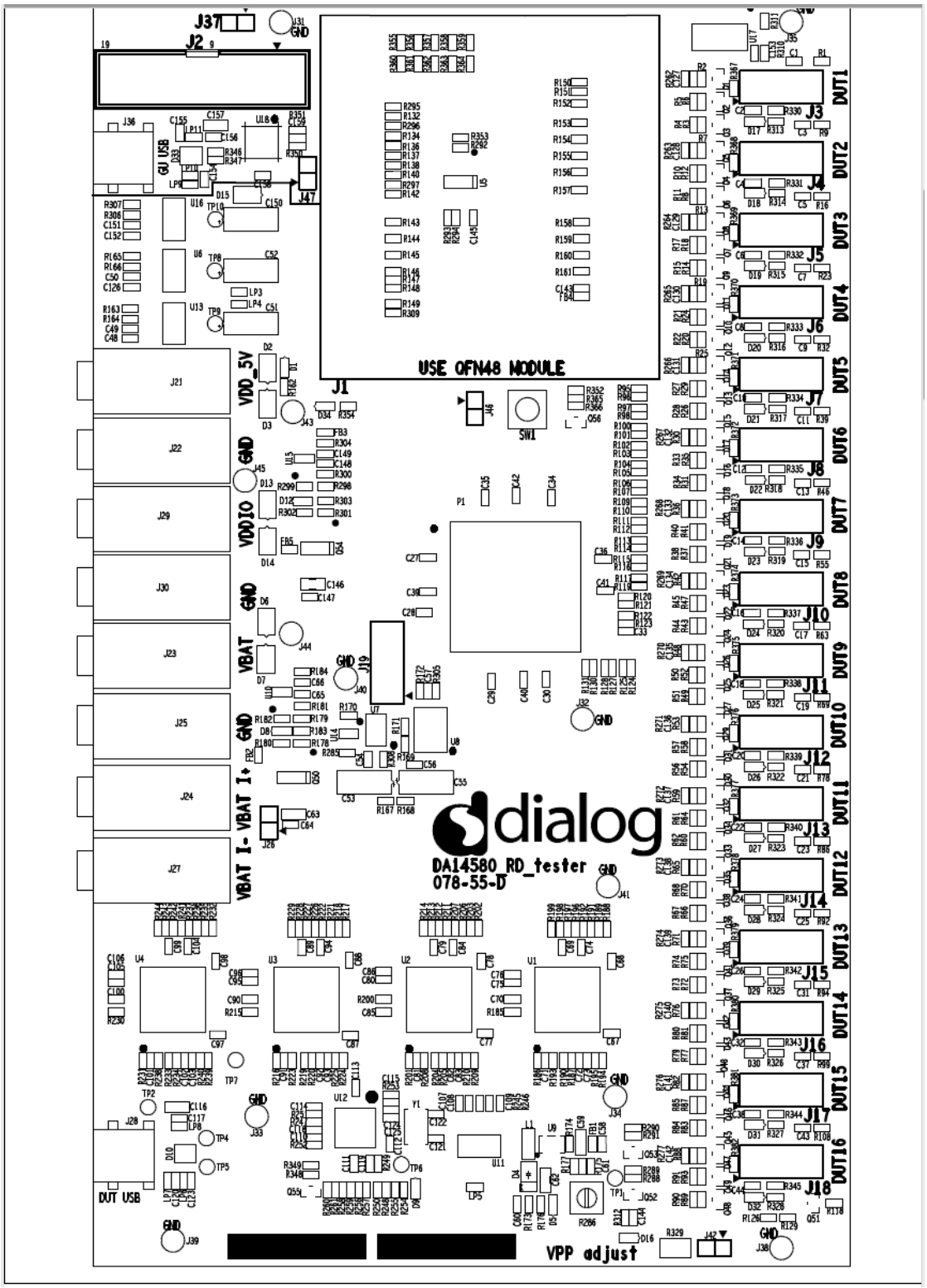



**Figure 142: Top View of PLT PCB Version D**


© 2020 Dialog Semiconductor

# Appendix B Electrical Schematics

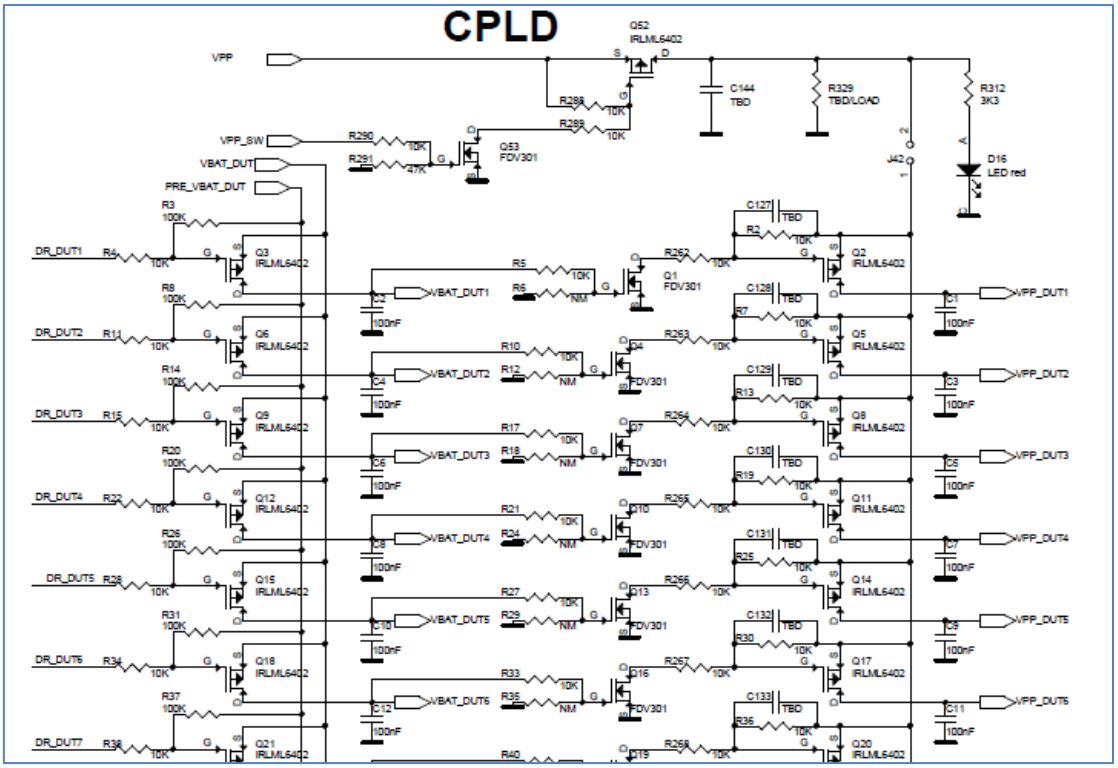



**Figure 143: VBAT and VPP Control from CPLD**

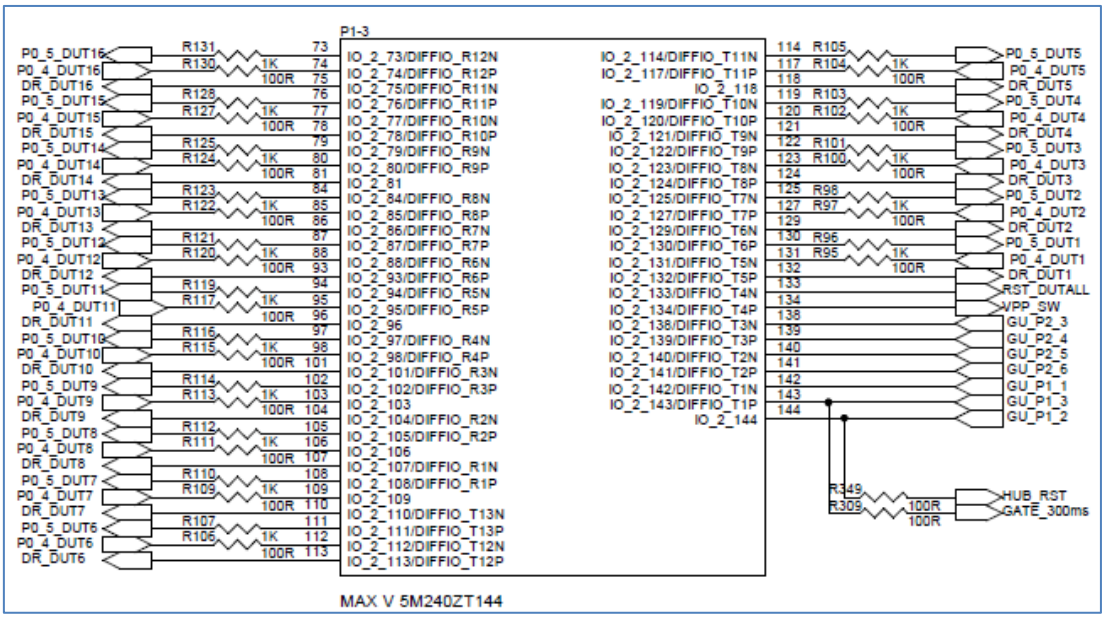



**Figure 144: CPLD DUT UART Connections**


© 2020 Dialog Semiconductor

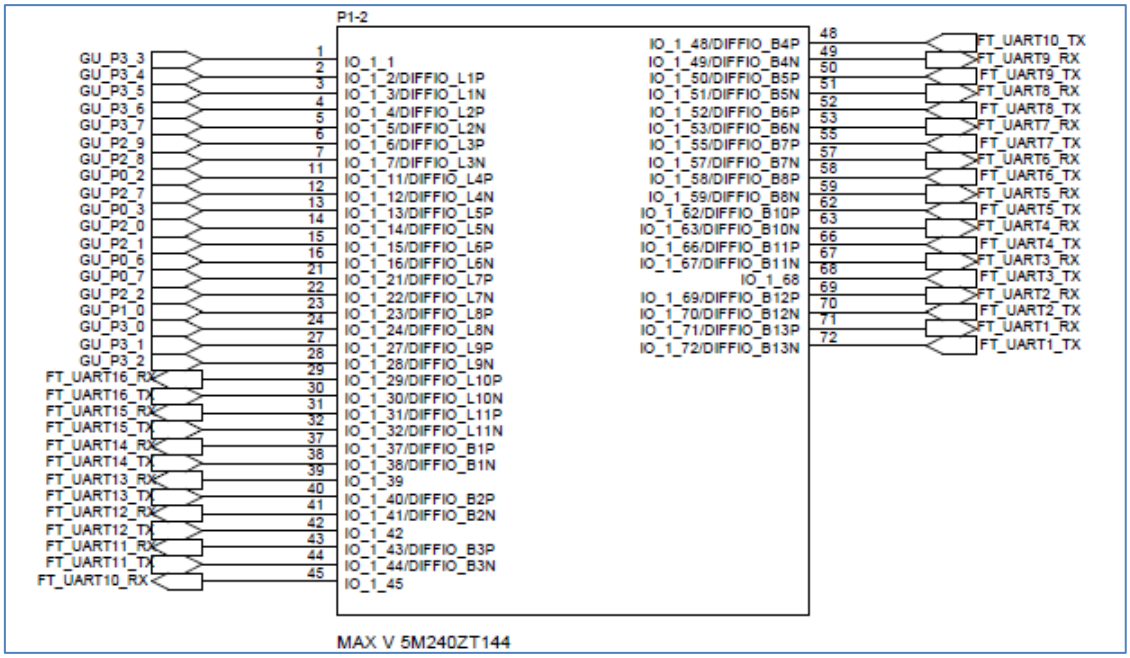



**Figure 145: CPLD FTDI and GU Control Connections**

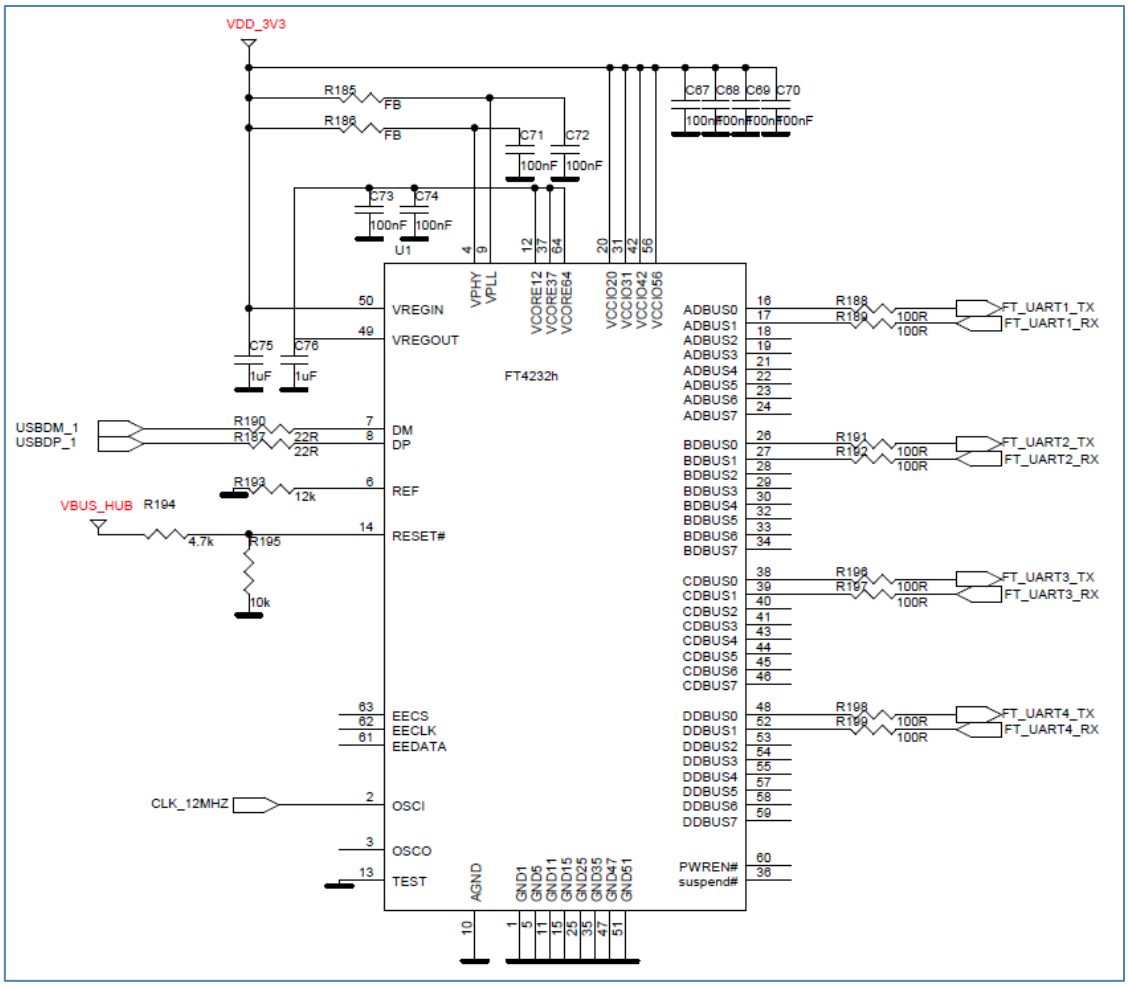



**Figure 146: FTDI Chip for USB UART to DUTs 1, 2, 3 and 4**


© 2020 Dialog Semiconductor

## SmartBond Production Line Tool

USB HUB: provides 5V input for the 3.3V LDO and USB input-signals to the four Quad FTDI chips.

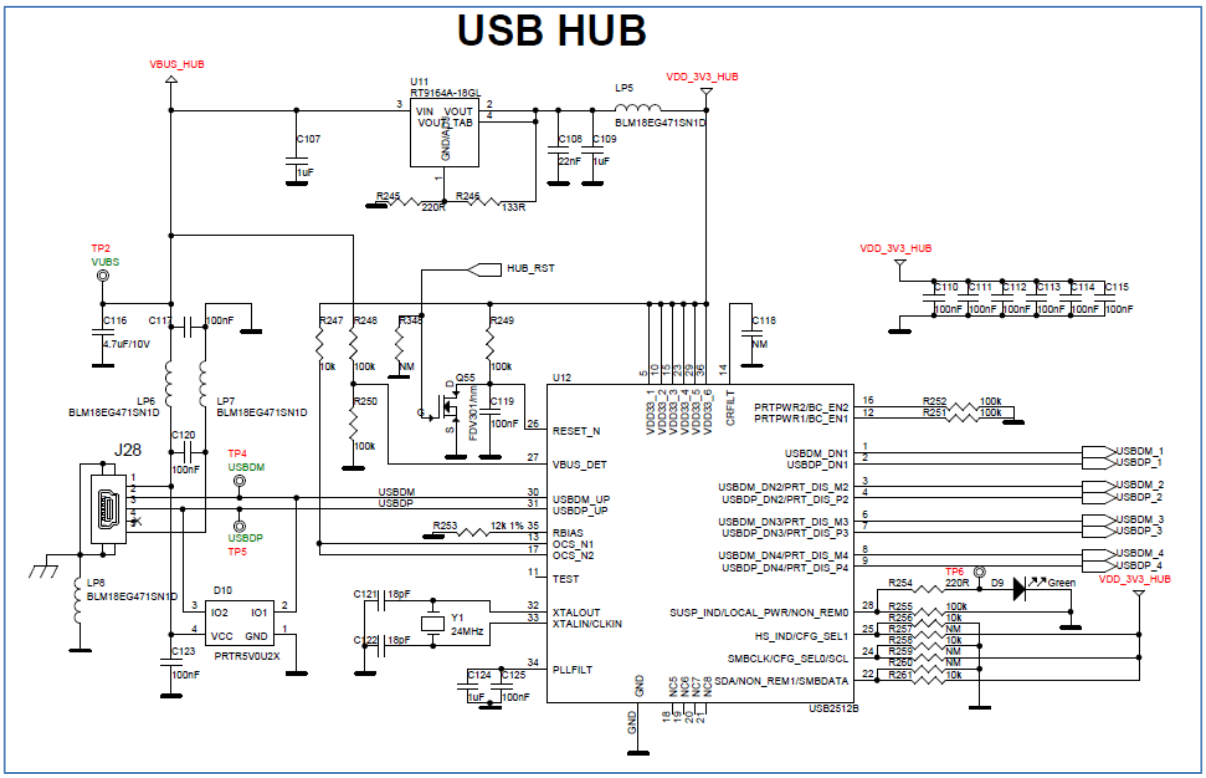



**Figure 147: Quad USB HUB**

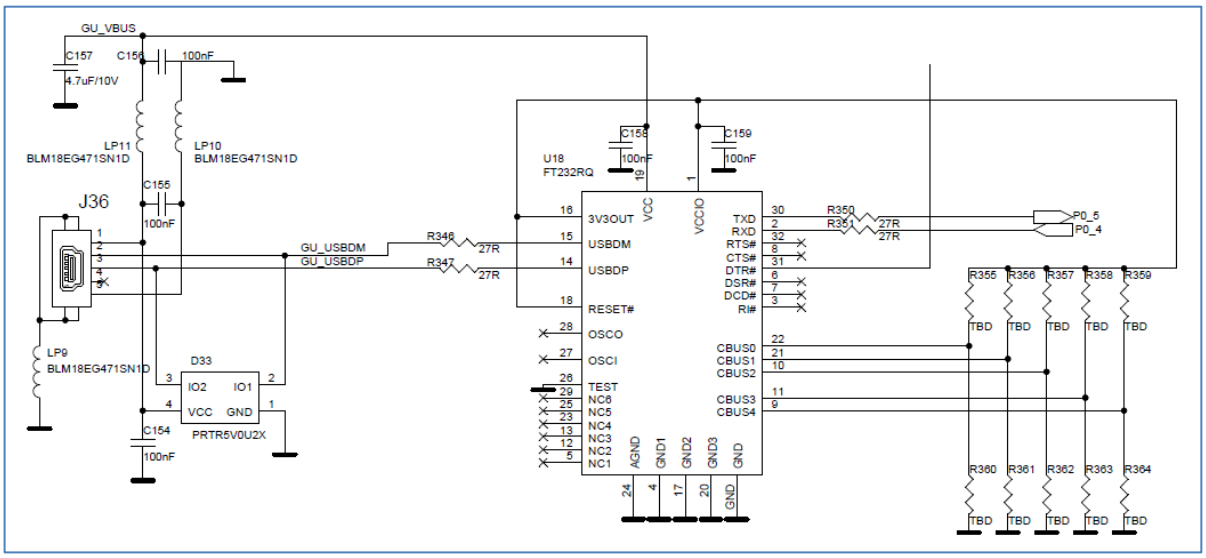



**Figure 148: Golden Unit - Dedicated USB Port and FTDI Chip**


© 2020 Dialog Semiconductor

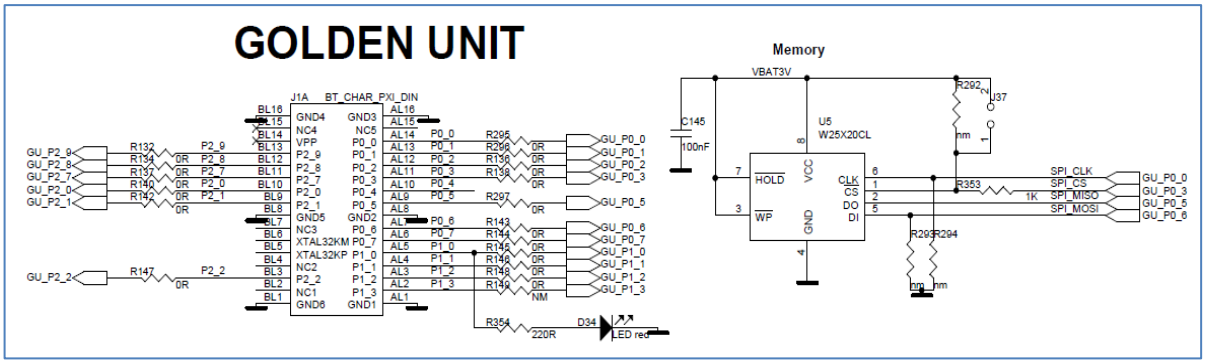



**Figure 149: Golden Unit - GU LED and SPI Flash Memory**

The Golden Unit SW (`prod_test_GU.bin`) is located in the SPI Flash memory mounted on the PLT hardware and is loaded into the GU's system RAM when powered on.

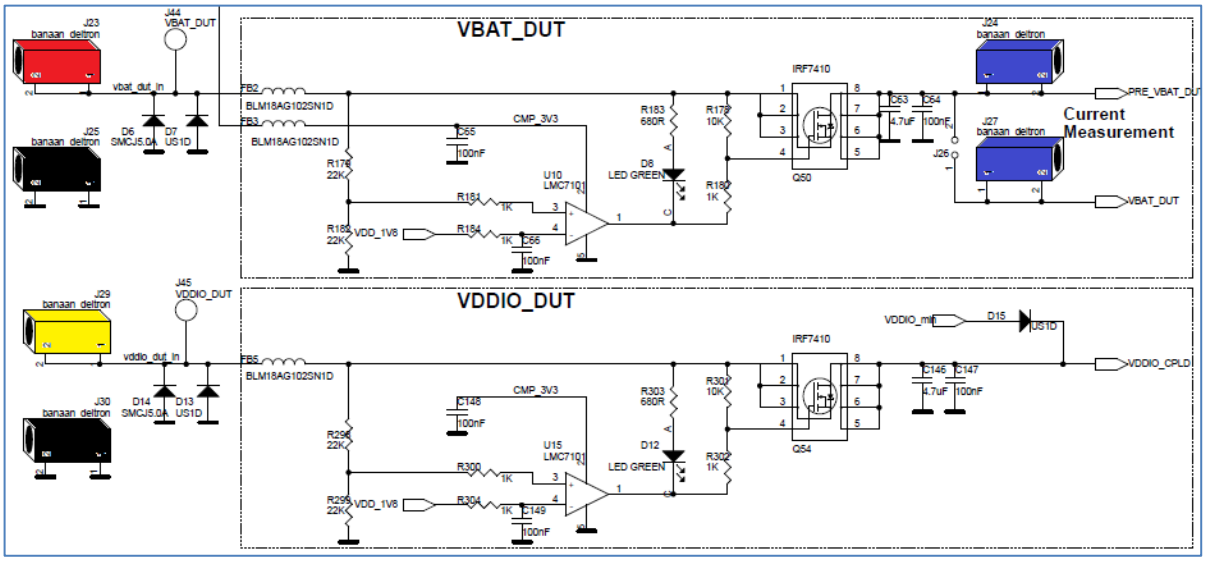



**Figure 150: VBAT_DUT and VDDIO Supplies**

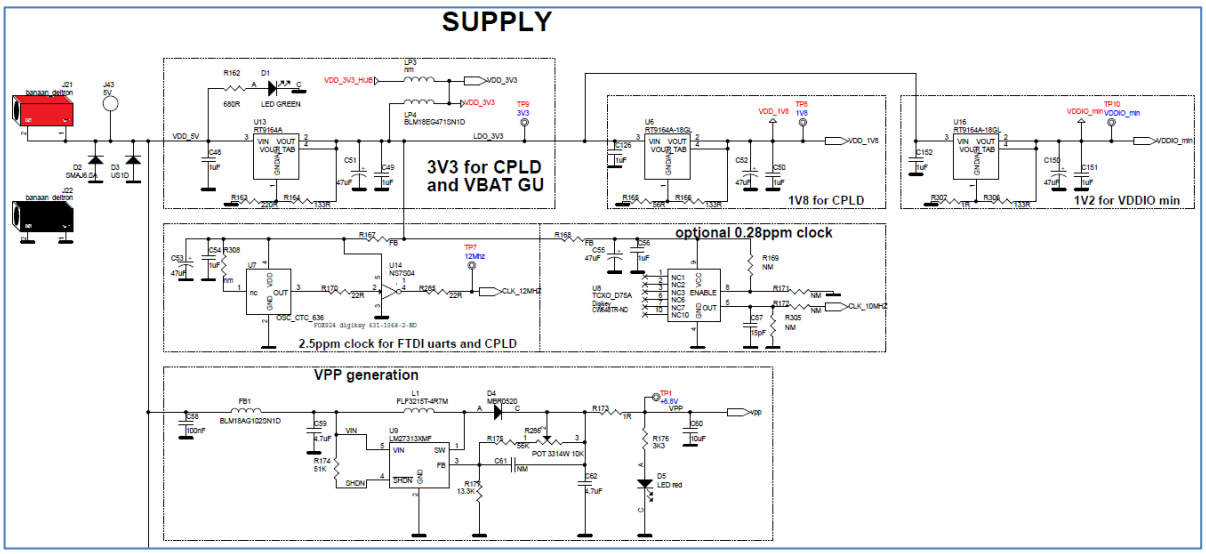



**Figure 151: GU Supply and VPP Generation**


© 2020 Dialog Semiconductor

# Appendix C Hardware Modifications PLT Version D

In the PLT hardware version D, a small modification was made. Resistor R365 (10 kΩ) and jumper J47 were added in series to the GU reset circuit.

**Figure 152: DA14580_RD_tester Version D**

**Figure 153: Jumper J47 Added Next to Golden Unit Socket**

**Figure 154: R365 (10 kΩ) Added Next to Reset Button**

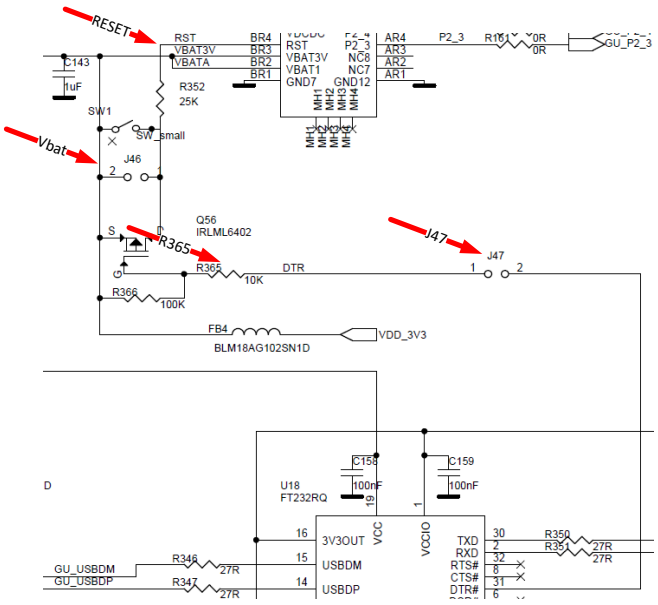



Figure 155: R365, J47 and RESET Shown **in Electrical Schematic**


© 2020 Dialog Semiconductor

# Appendix D Application Hardware Design Considerations

When the Production Line Tool (PLT) is used, one should be aware of the following items:

- Vpp PLT connection signal is used only for DA14580/1/2/3 devices.
- 1-Wire UART needs only one UART pin connection to the device. See Figure 6. It is only used for DA1453x devices.
- One could consider adding additional pads to their design for future debugging, not related to PLT, like pins for SWD.
- Pads are, in most cases, placed on the backside of the circuit board. They should be gold plated. Dimensions of these pads are crucial and have to do with the stability and accuracy of the pogo-pins that connect to the PLT HW. They should not be designed too critical. Long pogo pins might bend during production testing.
- Optionally, holes can be added for guiding-pins that fit on the test jig used for the PCB or panel
- Orientation of the antenna used on the application board will impact the RSSI-value.
  When panels are used, this RSSI will vary, dependent on the distance to the GU antenna on the PLT. In the PLT software an RSSI-offset can be added for each DUT location to compensate for these differences.

Other documentation can be found on the Dialog website:

- AN-B-054, DA1458x/DA1468x Application Hardware Design Guidelines, Application Note, Dialog Semiconductor.

  https://www.dialog-semiconductor.com/sites/default/files/an-b-054_da14585_da14586_application_hardware_design_guidelines_v1.2.pdf
- AN-B-061, DA1468x Application Hardware Design Guidelines, Application Note, Dialog Semiconductor.

  https://www.dialog-semiconductor.com/sites/default/files/an-b-061_da1468x_application_hardware_design_guidelines_v1.9.pdf
- AN-B-066, DA1469x Hardware Guidelines, Application Note, Dialog Semiconductor.

  https://www.dialog-semiconductor.com/sites/default/files/an-b-066_da1469x_application_hardware_design_guidelines_rev1.4.pdf
- AN-B-075, DA14531 Hardware Guidelines, Application Note, Dialog Semiconductor.

  https://www.dialog-semiconductor.com/sites/default/files/an-b-075_da14531_hardware_guidelines_1v3_0.pdf

 © 2020 Dialog Semiconductor

# Appendix E Suggestions about Hardware and Cabling

When connecting the PLT to the DUTs, special care should be taken with regard to the cabling.

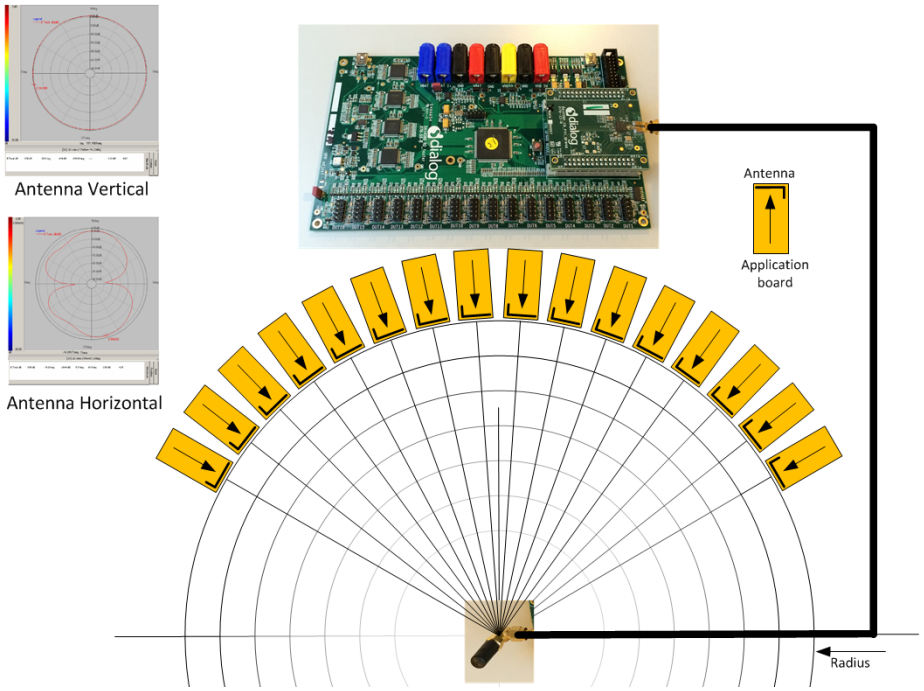



**Figure 156: Possible Solution of Antenna on Cable and Fixed Radius of DUTs to Antenna**

The user should realize that the PLT system is equipped with RF transmitters and receivers. These parts may induce noise on hardware and cables. The following aspects should be kept in mind:

- The direction of the GU antenna to the DUT antenna will influence the RSSI value.
- The distance of the DUT antenna to the GU antenna (radius) will influence the RSSI value.
- The control lines from the PLT to the DUTs must be kept as short as possible.
- A vertical GU antenna has different characteristics from a horizontal one. See Figure 156.

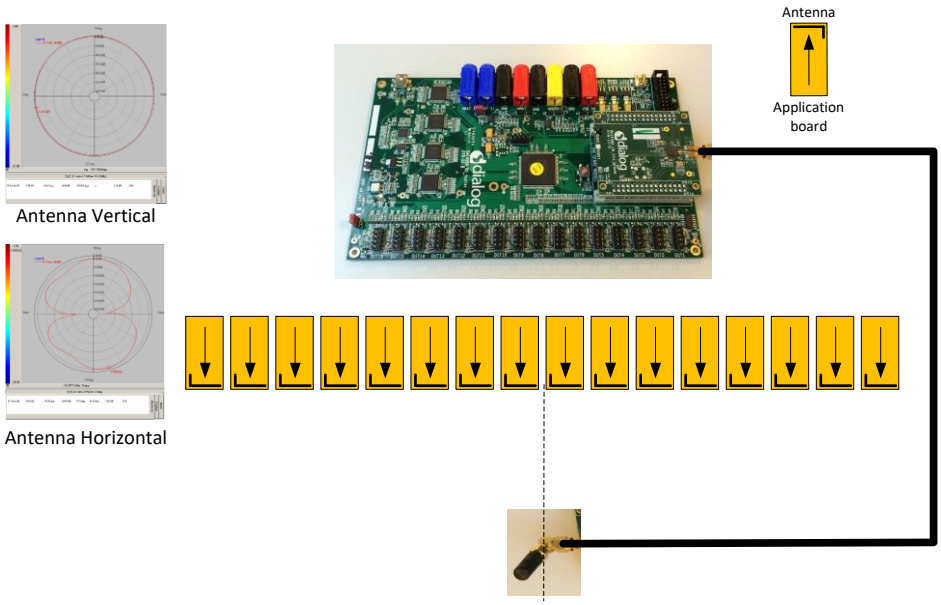



**Figure 157: Possible Solution of Antenna on Cable and DUTs Put in Line**


© 2020 Dialog Semiconductor

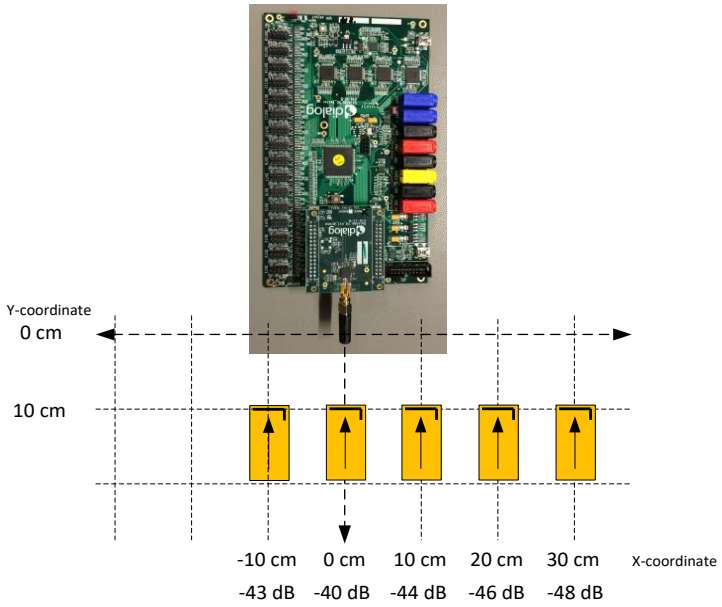



**Figure 158: Example Locations and RSSI Readouts of Horizontal Antenna**

Figure 158 shows the measured values from Table 104.

**Table 104: RF Test RSSI Results**

| Test | Distance (cm) | Offset (cm) | RSSI (dBm) | Description |
|---|---|---|---|---|
| 1 | 10 | 0 | -40 | DUT and GU boards are inline. |
| 2 | 10 | -10 | -43 | DUT moved 10 cm to the left relative to the GU. |
| 3 | 10 | 10 | -44 | DUT moved 10 cm to the right relative to the GU. |
| 4 | 10 | 20 | -46 | DUT moved 20 cm to the right relative to the GU. |
| 5 | 10 | 30 | -48 | DUT moved 30 cm to the right relative to the GU. |
| 6 | 10 | normal | -40 | DUT and GU boards are inline, functioning normally. |
| 7 | 10 | defect 1 | -60 ~ -70 | Coupling capacitor not soldered well, missing or damaged. |
| 8 | 10 | defect 2 | ~ -60 | Short circuited shunt matching inductor (e.g. solder bridge) |
| 9 | 10 | defect 3 | < -100 | 16 MHz crystal oscillator not working well. Received packets ~ 0. |
| Golden Unit output power = 0 dBm | | | | |

For more details on the RF setup, refer to [12] and Appendix F.

 © 2020 Dialog Semiconductor

# Appendix F RF Path Losses Calibration

To accurately perform radiated tests for 16 DUTs using the Golden Unit or a BLE tester, one should calibrate the setup in order to know what RSSI value can be expected for non-problematic devices. Because the distance and the position of each of the 16 DUTs to the GU RF antenna are different, the calibration process calculates the different path losses to compensate for these differences. The calculated values are applied to the Production Line Tool configuration as RF path losses (DA14531 - Path Losses per DUT, DA1469x - Path Losses per DUT), which are actually added in the RSSI result.

This chapter describes the process to calculate the RF path losses for each different DUT position.

## F.1 Prerequisites

Table 105 illustrates the prerequisites needed for performing the RF path loss calibration procedure.

**Table 105: Prerequisites**

| # | Requirements | Description |
|---|---|---|
| 1 | 1 PLT board | 1 PLT board with Golden Unit.<br>Power supply for the PLT.<br>USB cables for the PLT to the PC. |
| 2 | <10 PCBA with 16 DUTs each | At least 10 PCBAs. The more PCBAs used the better.<br>The PCBAs selected should all work as good as possible.<br>If a fault device is identified on a PCBA that PCBA should be replaced. |
| 3 | 1 shielded box | It must be big enough to fit the PCBA and the fixture.<br>It should have one SMA female to female connector and a small hole to pass the DUT to PLT cable connections. |
| 4 | 2 RF cables (1 optional) | One cable to be used from the PLT GU to the shielded box.<br>One more cable to be used from the shielded box to the RF antenna. (This is optional since the RF antenna can be directly mounted into the shielded box RF SMA connector).<br>The cables should have low attenuation at 2.5Ghz range (<2dB) and high shielding effectiveness (>60dB).<br>Proposed cables are from Radiall. Cables datasheet:<br>https://www.radiall.com/media/files/RFCableAssemblies%20D1C004XEe.pdf<br>● Flexible cable 2.6/50 D (RD316) P/N: C291 185 067<br>● Flexible cable 2/50 D (124416 type) P/N: C291 146 087<br>● Flexible cable 2.6/50 D (ECO316D: alternative to RD316) P/N: C291 999 905<br>● Flexible cable 5/50 D (ECO142: alternative to RG142) P/N: C291 325 290<br>● Flexible cable 5/50 D (Power 142: alternative to RG142) P/N: C291 325 270<br>● Flexible cable 6/50 D (ECO230) P/N: C291 326 490 |
| 5 | DUT fixture | A fixture to be placed inside the shielded box to easily connect the PCBAs to the PLT. |

 © 2020 Dialog Semiconductor

## F.2 Setup

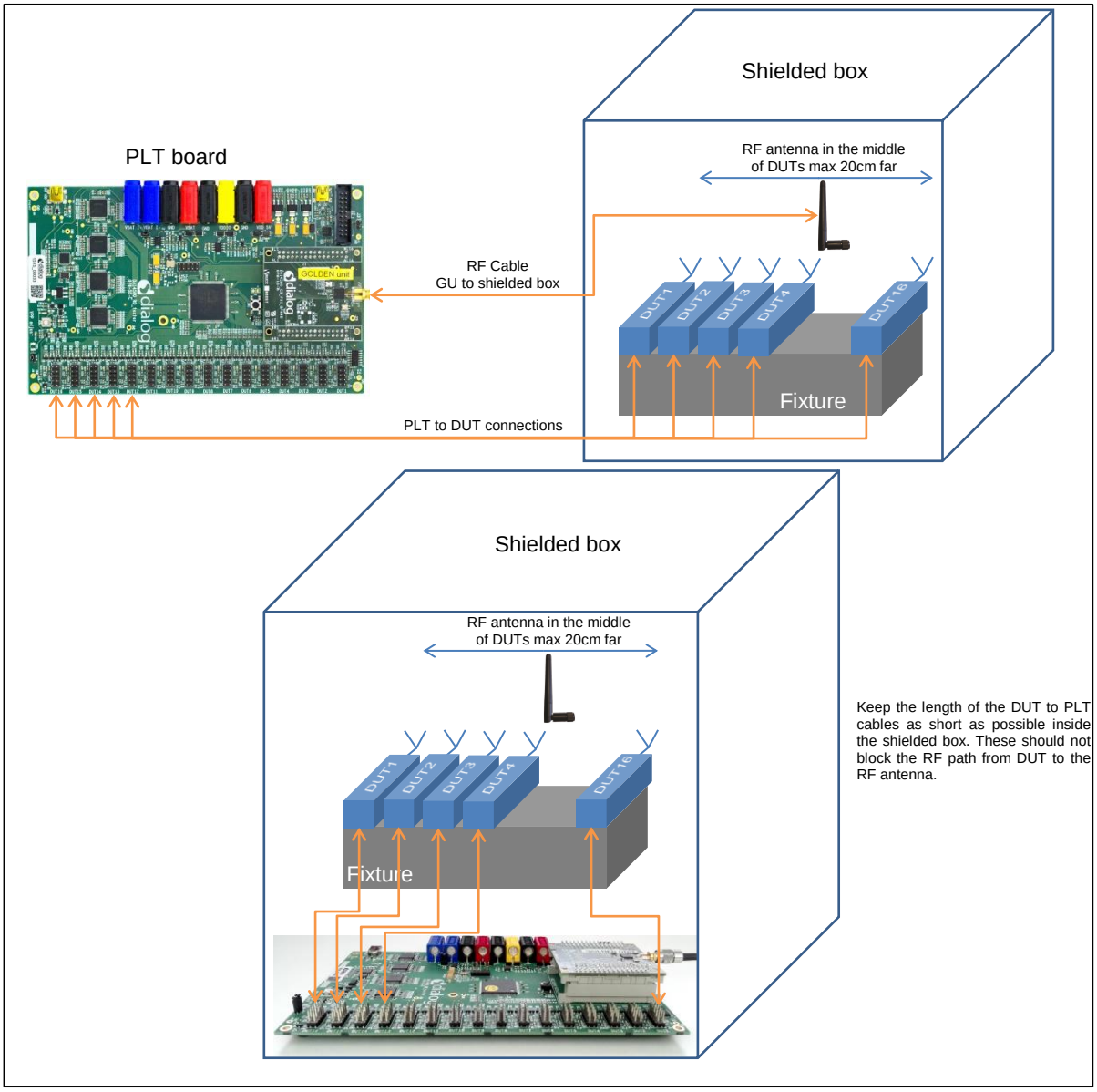



**Figure 159: Setup diagram**

**Table 106: Calibration Setup**

| Item | Description |
|---|---|
| PLT board | The PLT board could either be outside or inside the shielded box, as shown in Figure 159, depending on the fixture setup. |
| DUTs | The DUT antennas should point the RF antenna. |
| PLT to DUT connections | The PLT to DUT cable connections should be as short as possible. Also, the cables should not block the RF path from DUT to the RF antenna. |
| Shielded box | The shielded box must be big enough to fit the PCBA and the fixture and the PLT if it is inside. |

 © 2020 Dialog Semiconductor

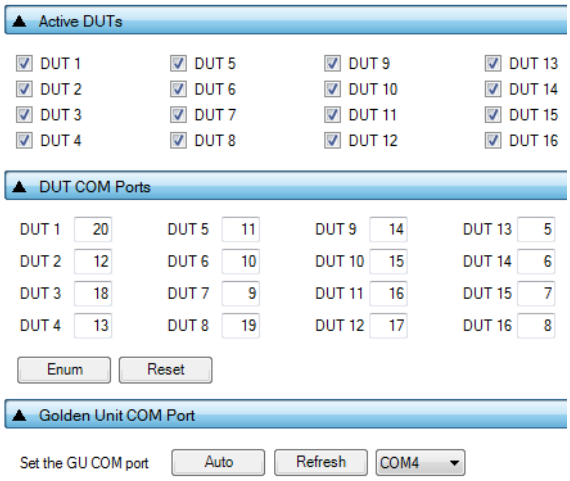
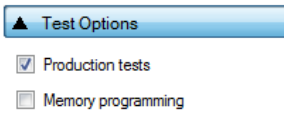
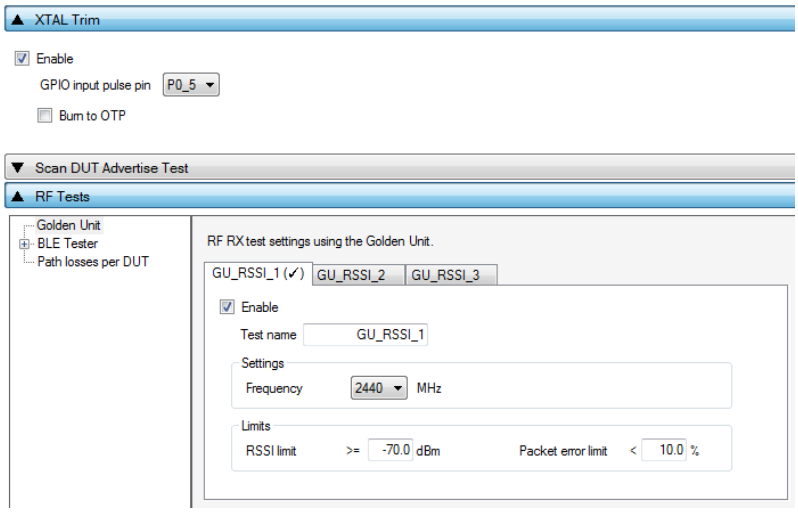
| Item | Description |
|---|---|
| RF cable - PLT to shielded box. | This type of cable must have good shielded effectiveness and low attenuation.<br>Its shielded effectiveness must be >60dB. Check Table 105 for the best proposals.<br>If multiple production lines are used the shielding of this RF cable is crucial not to have disturbance from other PLTs that are close. However, if other PLTs are far away (>3m) other cables could also be used.<br>● FLEXIBLE CABLE 2.6/50 S (RG316 - KX22A) P/N: C291 170 007 |
| RF cable – Shielded box to RF antenna (optional). | Due to the close distance between the DUT antennas and the RF antenna cable, the shielding of this cable should be very good. It must be at least 60dB. Check Table 105 for proposals.<br>This cable can be optional. The RF antenna could be mounted directly to the SMA RF connector inside the shielded box. In that case the fixture and the DUTs should be placed appropriately (less than 20cm). |
| RF antenna | The RF antenna can be any good Wi-Fi antenna that operates at 2.5GHz. It should be placed in a vertical position as shown in Figure 159. The distance to the DUTs should not be larger than 20cm. It should be placed in the middle of the DUTs (in front of DUT 8).<br>Bear in mind that the Anritsu MT8852B BLE tester cannot perform TX measurements if the received signal at its antenna is less than -50dBm. Having said that, for a good measurement, the signal reaching its antenna should be more than -40dBm.<br>Therefore, the distance and the placement between the DUTs and the RF antenna are very crucial. A lot of trial and errors could be tried out until the optimal antenna position is found.<br>The RF antenna placement should be very stable. After the optimal position is found, it should be fixed into position and not able to move in any way. |
| DUT fixture | The fixture position should be fixed compared to the RF antenna. The fixture should not move in any way in order to keep the distance between the DUTs and the RF antenna fixed. |

 © 2020 Dialog Semiconductor

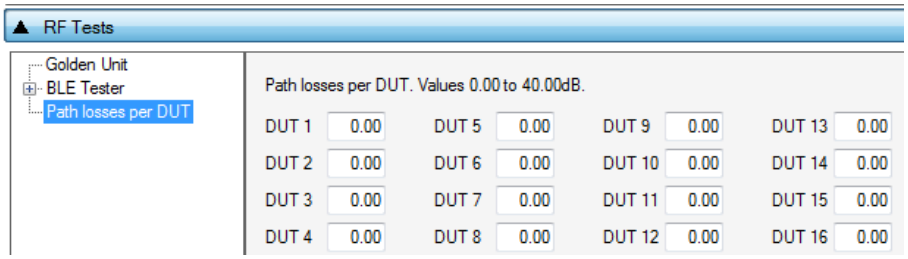
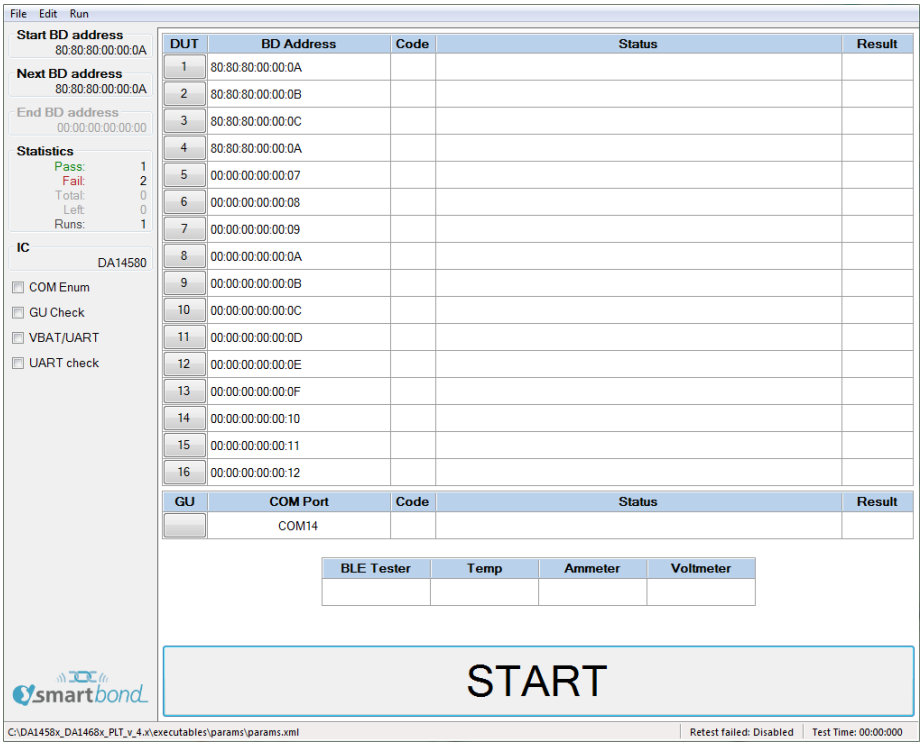
## F.3 Procedure

Table 107 describes the steps to be followed in order to calculate the RF path losses for each different DUT position.

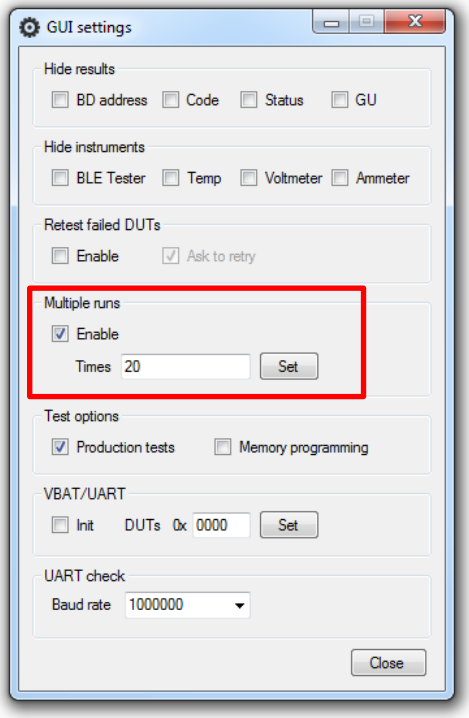
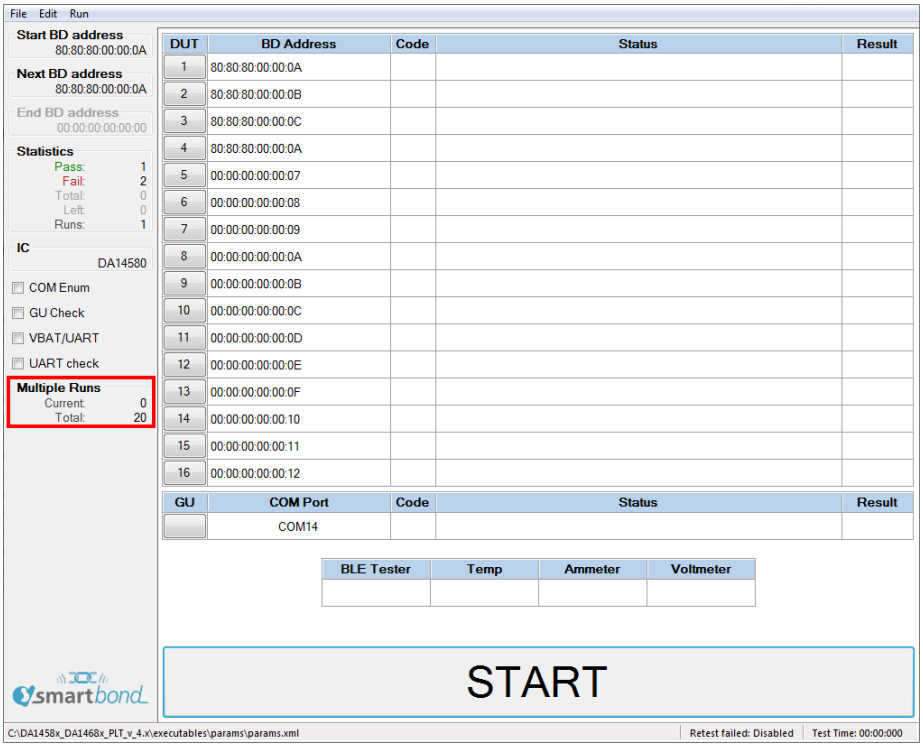
**Table 107: Procedure steps**

| # | Step | Description |
|---|---|---|
| 1 | Open the SmartBond_CFG_PLT.exe PLT configuration executable. | User should configure the PLT so only XTAL trim (without OTP burn) and one Golden Unit RF test is enabled. |
| 2 | Enable all 16 DUTs | Select the GU COM port and Enumerate the DUT COM port numbers.<br>  <br> |
| 3 | Enable only the Production tests | Under the General tab, enable only the Production tests and disable the Memory programming.<br>  <br> |
| 4 | Enable XTAL trim (no OTP burn) and one Golden Unit RF test at middle band at 2440MHz. | RSSI and PER limits do not matter. These should be set to a small value (less than -70dBm) as shown below. Ideally, we want the limits to be set to a value that all RF tests PASS.<br>  <br> |

 © 2020 Dialog Semiconductor

| # | Step | Description |
|---|---|---|
| 5 | Set all DUT path losses to 0dB. |  |
| 6 | Backup CSV log file. | Go to SmartBond_PLT_v_4.x\executables\logs and back up the today's CSV file. For example, if the today's CSV file is Test_station_1_20180306_csv_results.csv rename it to Test_station_1_20180306_csv_results_BackUp.csv.<br>Doing so, a new CSV file will be created at the next PLT test run. |
| 7 | Place the 1st PCBA | Place the 1st PCBA into the fixture inside the shielded box. |
| 8 | Open SmartBond_GUI_PLT.exe |  |

 © 2020 Dialog Semiconductor

| # | Step | Description |
|---|---|---|
| 9 | Go to<br>Edit->Settings | Select the Multiple runs. Set the Times to 20. Press Set and Close.<br>  <br> |
| 10 | In the SmartBond_GUI_PLT.exe the Multiple Runs should be shown on the left panel |  <br> |
| 11 | Press START | Press START and wait for the 20 tests to be performed. |
| 12 | Backup CSV log file | Go to SmartBond_PLT_v_4.2.3.198\executables\logs and back up the CSV file for the 1st PCBA. For example if the today's CSV file is Test_station_1_20180306_csv_results.csv<br>rename it to<br>Test_station_1_20180306_csv_results_PCBA_1.csv. |

 © 2020 Dialog Semiconductor

| # | Step | Description |
|---|---|---|
| 13 | Place the 2nd PCBA | Place the 2nd PCBA into the fixture inside the shielded box. |
| 14 | Repeat steps | Repeat the procedure from step 9 to step 13 for all 10 PCBAs. |
| 15 | Check CSV results | At the end, 10 CSV logs files should exist. E.g.<br>Test_station_1_20180306_csv_results_PCBA_1.csv.<br>Test_station_1_20180306_csv_results_PCBA_2.csv.<br>…<br>Test_station_1_20180306_csv_results_PCBA_10.csv.<br>Each CSV file should have 16 DUTs * 20 Tests = 320 lines + 1 CSV header = 321 lines. Example: |

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Start time | End time | DUT | BD addres | Overal | CON | FW do | FW path 1 | FW ve | FW versio | XTAL tri | XTAL t | GU RX | GU RX RSS | GU RX F |
| 2 | 11:20:33 | 11:20:47 | 1 | 00:00:00:0 | PASS | 20 | PASS | C:\Users\ | PASS | v_5.0.4_P | PASS | 1172 | PASS | -28.5 | 0 |
| 3 | 11:20:33 | 11:20:47 | 2 | 00:00:00:0 | PASS | 12 | PASS | C:\Users\ | PASS | v_5.0.4_P | PASS | 1207 | PASS | -26.61 | 0 |
| 4 | 11:20:33 | 11:20:47 | 3 | 00:00:00:0 | PASS | 18 | PASS | C:\Users\ | PASS | v_5.0.4_P | PASS | 1272 | PASS | -26.61 | 0 |
| 5 | 11:20:33 | 11:20:47 | 4 | 00:00:00:0 | PASS | 13 | PASS | C:\Users\ | PASS | v_5.0.4_P | PASS | 1222 | PASS | -28.5 | 0 |
| 6 | 11:20:33 | 11:20:47 | 5 | 00:00:00:0 | PASS | 11 | PASS | C:\Users\ | PASS | v_5.0.4_P | PASS | 1181 | PASS | -26.13 | 0 |
| 7 | 11:20:33 | 11:20:47 | 6 | 00:00:00:0 | PASS | 10 | PASS | C:\Users\ | PASS | v_5.0.4_P | PASS | 1218 | PASS | -22.34 | 0 |
| 8 | 11:20:33 | 11:20:47 | 7 | 00:00:00:0 | PASS | 9 | PASS | C:\Users\ | PASS | v_5.0.4_P | PASS | 1237 | PASS | -28.98 | 0 |
| 9 | 11:20:33 | 11:20:47 | 8 | 00:00:00:0 | PASS | 19 | PASS | C:\Users\ | PASS | v_5.0.4_P | PASS | 1180 | PASS | -38.93 | 0 |
| 10 | 11:20:33 | 11:20:47 | 9 | 00:00:00:0 | PASS | 14 | PASS | C:\Users\ | PASS | v_5.0.4_P | PASS | 1207 | PASS | -27.08 | 0 |
| 11 | 11:20:33 | 11:20:47 | 10 | 00:00:00:0 | PASS | 15 | PASS | C:\Users\ | PASS | v_5.0.4_P | PASS | 1244 | PASS | -36.09 | 0 |
| 12 | 11:20:33 | 11:20:47 | 11 | 00:00:00:0 | PASS | 16 | PASS | C:\Users\ | PASS | v_5.0.4_P | PASS | 1180 | PASS | -34.19 | 0 |
| 13 | 11:20:33 | 11:20:47 | 12 | 00:00:00:0 | PASS | 17 | PASS | C:\Users\ | PASS | v_5.0.4_P | PASS | 1168 | PASS | -27.08 | 0 |
| 14 | 11:20:33 | 11:20:47 | 13 | 00:00:00:0 | PASS | 5 | PASS | C:\Users\ | PASS | v_5.0.4_P | PASS | 1255 | PASS | -32.29 | 0 |
| 15 | 11:20:33 | 11:20:47 | 14 | 00:00:00:0 | PASS | 6 | PASS | C:\Users\ | PASS | v_5.0.4_P | PASS | 1247 | PASS | -23.76 | 0 |
| 16 | 11:20:33 | 11:20:47 | 15 | 00:00:00:0 | PASS | 7 | PASS | C:\Users\ | PASS | v_5.0.4_P | PASS | 1180 | PASS | -24.71 | 0 |
| 17 | 11:20:33 | 11:20:47 | 16 | 00:00:00:0 | PASS | 8 | PASS | C:\Users\ | PASS | v_5.0.4_P | PASS | 1247 | PASS | -28.98 | 0 |

| | | |
|---|---|---|
| 16 | Get the average for each DUT in each CSV file | In each CSV file, get the average GU RX RSSI value for each DUT. Apply the following formulas to each of the 10 CSV files. |

| DUT # | Formula | Example Result |
|---|---|---|
| DUT 1 | =SUMIF($C$2:$C$321,**1**, $N$2:$N$321)/20 | -28.5015 |
| DUT 2 | =SUMIF($C$2:$C$321,**2**, $N$2:$N$321)/20 | -26.609 |
| DUT 3 | =SUMIF($C$2:$C$321,**3**, $N$2:$N$321)/20 | -26.8685 |
| DUT 4 | =SUMIF($C$2:$C$321,**4**, $N$2:$N$321)/20 | -28.406 |
| DUT 5 | =SUMIF($C$2:$C$321,**5**, $N$2:$N$321)/20 | -26.346 |
| DUT 6 | =SUMIF($C$2:$C$321,**6**, $N$2:$N$321)/20 | -22.5515 |
| DUT 7 | =SUMIF($C$2:$C$321,**7**, $N$2:$N$321)/20 | -28.98 |
| DUT 8 | =SUMIF($C$2:$C$321,**8**, $N$2:$N$321)/20 | -39.283 |
| DUT 9 | =SUMIF($C$2:$C$321,**9**, $N$2:$N$321)/20 | -27.08 |
| DUT 10 | =SUMIF($C$2:$C$321,**10**,$N$2:$N$321)/20 | -36.607 |
| DUT 11 | =SUMIF($C$2:$C$321,**11**,$N$2:$N$321)/20 | -34.19 |
| DUT 12 | =SUMIF($C$2:$C$321,**12**,$N$2:$N$321)/20 | -26.7745 |
| DUT 13 | =SUMIF($C$2:$C$321,**13**,$N$2:$N$321)/20 | -32.0095 |
| DUT 14 | =SUMIF($C$2:$C$321,**14**,$N$2:$N$321)/20 | -23.784 |
| DUT 15 | =SUMIF($C$2:$C$321,**15**,$N$2:$N$321)/20 | -24.71 |
| DUT 16 | =SUMIF($C$2:$C$321,**16**,$N$2:$N$321)/20 | -29.168 |


© 2020 Dialog Semiconductor

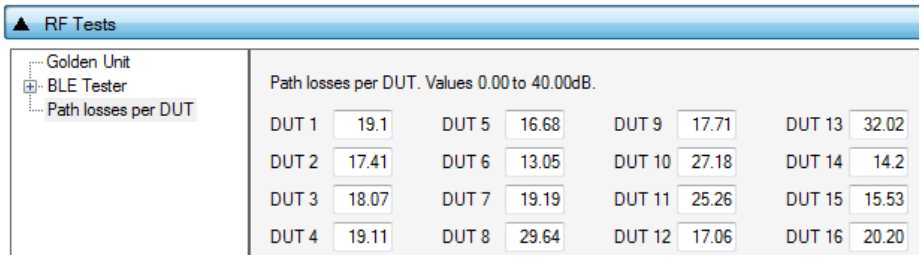
| # | Step | Description |
|---|---|---|
| 17 | Get all values to a new excel sheet. | Create a new excel sheet. Copy all values created at step 16 from the 10 CSV files to this new excel sheet. An example of all DUT average values from all 10 CSV files is given below. Only 2 decimal digits are shown. |

| DUT | PCBA 1 | PCBA 2 | PCBA 3 | PCBA 4 | PCBA 5 | PCBA 6 | PCBA 7 | PCBA 8 | PCBA 9 | PCBA 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -28.50 | -28.30 | -28.51 | -27.56 | -29.48 | -28.59 | -30.09 | -29.33 | -30.50 | -30.13 |
| 2 | -26.61 | -26.07 | -27.09 | -26.56 | -27.99 | -27.38 | -28.31 | -27.64 | -28.36 | -28.11 |
| 3 | -26.87 | -26.71 | -27.65 | -26.71 | -28.64 | -27.84 | -29.14 | -28.43 | -29.54 | -29.14 |
| 4 | -28.41 | -27.98 | -28.82 | -27.87 | -29.49 | -29.24 | -30.10 | -29.61 | -30.22 | -29.36 |
| 5 | -26.35 | -25.69 | -26.72 | -26.58 | -26.74 | -26.02 | -27.37 | -26.75 | -27.42 | -27.20 |
| 6 | -22.55 | -21.60 | -22.68 | -21.81 | -23.32 | -22.33 | -23.82 | -23.77 | -24.45 | -24.19 |
| 7 | -28.98 | -28.59 | -29.31 | -28.49 | -29.42 | -28.63 | -29.84 | -28.95 | -29.98 | -29.73 |
| 8 | -39.28 | -39.06 | -39.50 | -38.95 | -39.53 | -39.38 | -40.32 | -40.11 | -40.39 | -39.90 |
| 9 | -27.08 | -26.64 | -27.68 | -27.07 | -27.76 | -27.28 | -28.40 | -27.51 | -28.89 | -28.75 |
| 10 | -36.61 | -35.74 | -36.85 | -36.05 | -37.07 | -37.04 | -37.96 | -37.59 | -38.84 | -38.03 |
| 11 | -34.19 | -33.57 | -35.15 | -34.94 | -36.09 | -35.10 | -36.11 | -35.23 | -36.21 | -35.97 |
| 12 | -26.77 | -26.51 | -26.90 | -26.65 | -27.33 | -27.11 | -27.40 | -26.67 | -28.03 | -27.26 |
| 13 | -32.01 | -31.14 | -32.47 | -31.73 | -33.29 | -32.96 | -34.01 | -33.61 | -34.85 | -34.17 |
| 14 | -23.78 | -23.31 | -24.50 | -23.56 | -24.51 | -24.46 | -24.55 | -23.60 | -25.03 | -24.67 |
| 15 | -24.71 | -23.89 | -25.39 | -24.82 | -25.46 | -25.37 | -26.24 | -25.42 | -27.09 | -26.94 |
| 16 | -29.17 | -28.37 | -30.09 | -30.07 | -30.78 | -29.83 | -31.27 | -30.51 | -31.31 | -30.56 |

| # | Step | Description |
|---|---|---|
| 18 | Get the average of each DUT for all PCBAs | Average each DUTs results. |

| DUT | =AVERAGE(B1:K1) |
|---|---|
| DUT 1 | -29.10 |
| DUT 2 | -27.41 |
| DUT 3 | -28.07 |
| DUT 4 | -29.11 |
| DUT 5 | -26.68 |
| DUT 6 | -23.05 |
| DUT 7 | -29.19 |
| DUT 8 | -39.64 |
| DUT 9 | -27.71 |
| DUT 10 | -37.18 |
| DUT 11 | -35.26 |
| DUT 12 | -27.06 |
| DUT 13 | -33.02 |
| DUT 14 | -24.20 |
| DUT 15 | -25.53 |
| DUT 16 | -30.20 |


© 2020 Dialog Semiconductor

<table>
<tr><th>#</th><th>Step</th><th>Description</th></tr>
<tr><td>19</td><td>Calculate the RF path loss</td><td>To calibrate the RF result to -10dBm we should apply the formula shown in the third column below.
<table>
<tr><th>DUT</th><th>=AVERAGE(B1:K1)<br>Row L</th><th>Path Loss<br>=-10-L1</th></tr>
<tr><td>DUT 1</td><td>-29.10</td><td>19.10</td></tr>
<tr><td>DUT 2</td><td>-27.41</td><td>17.41</td></tr>
<tr><td>DUT 3</td><td>-28.07</td><td>18.07</td></tr>
<tr><td>DUT 4</td><td>-29.11</td><td>19.11</td></tr>
<tr><td>DUT 5</td><td>-26.68</td><td>16.68</td></tr>
<tr><td>DUT 6</td><td>-23.05</td><td>13.05</td></tr>
<tr><td>DUT 7</td><td>-29.19</td><td>19.19</td></tr>
<tr><td>DUT 8</td><td>-39.64</td><td>29.64</td></tr>
<tr><td>DUT 9</td><td>-27.71</td><td>17.71</td></tr>
<tr><td>DUT 10</td><td>-37.18</td><td>27.18</td></tr>
<tr><td>DUT 11</td><td>-35.26</td><td>25.26</td></tr>
<tr><td>DUT 12</td><td>-27.06</td><td>17.06</td></tr>
<tr><td>DUT 13</td><td>-33.02</td><td>23.02</td></tr>
<tr><td>DUT 14</td><td>-24.20</td><td>14.20</td></tr>
<tr><td>DUT 15</td><td>-25.53</td><td>15.53</td></tr>
<tr><td>DUT 16</td><td>-30.20</td><td>20.20</td></tr>
</table></td></tr>
<tr><td>20</td><td>Apply the calculated path losses to the SmartBond _CFG_PLT.exe PLT configuration executable.</td><td> 
</td></tr>
<tr><td>21</td><td>Verify</td><td>Repeat steps 10 to 13 with the 10 PCBAs. Check the GU RX RSSI results in the 10 CSV files. The results should be very close to -10dBm.</td></tr>
</table>

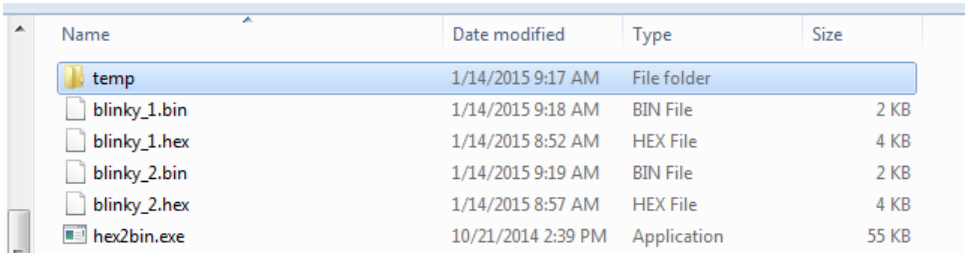

© 2020 Dialog Semiconductor

# Appendix G Hex2Bin

This section gives a step-by-step example of using the `hex2bin.exe` utility, which converts Intel HEX files into binary format. See Figure 161.

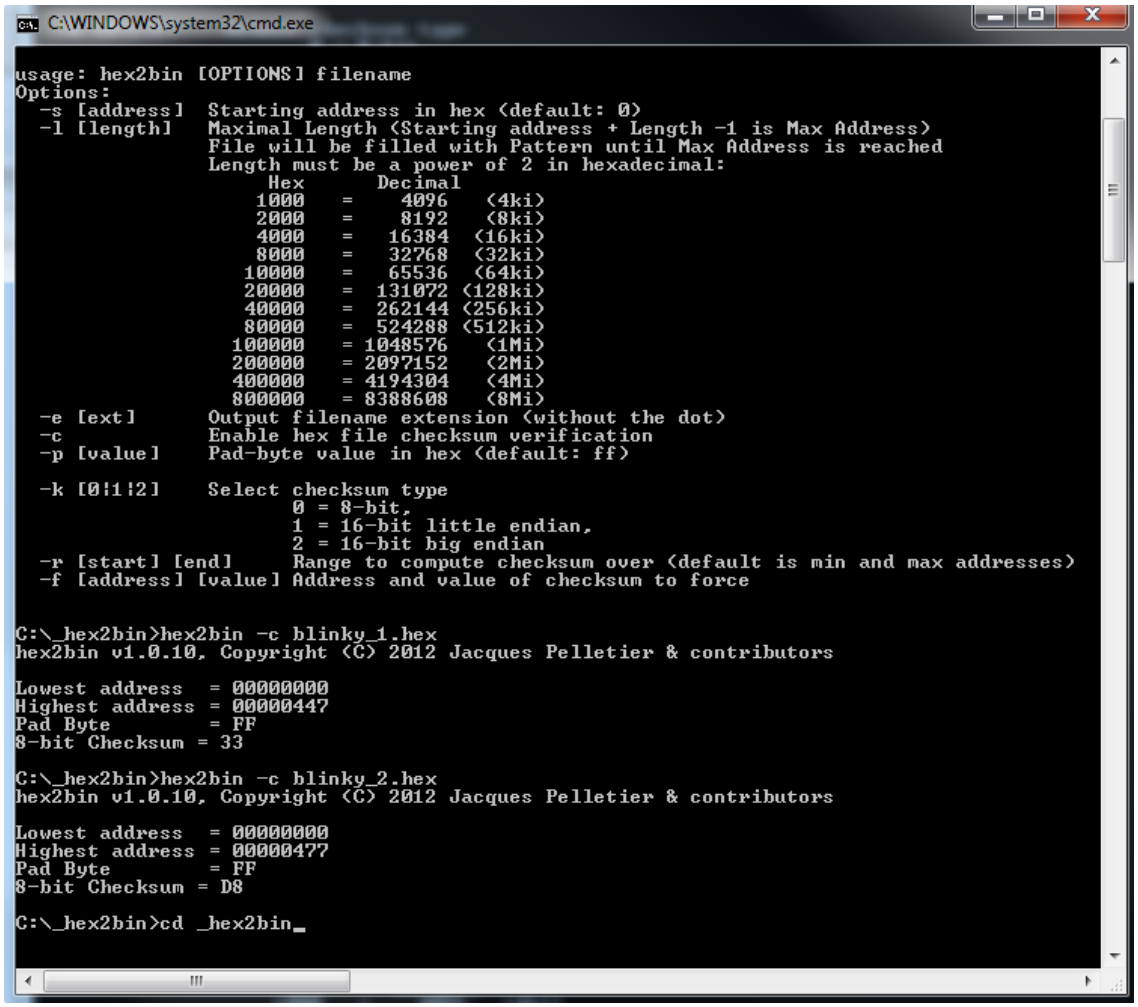
1. Put the `hex2bin.exe` file in same directory as the HEX files to be converted.
2. Open a Command Line Interface (CLI) in the same directory, e.g. by using <Shift>+<Right Click> and selecting 'Open command window here'.
3. Enter "`hex2bin -c blinky_1.hex`".
4. The binary file (`blinky_1.bin`) will be produced in the same directory.

Figure 160 shows the directory and the files used in this example.



| Name | Date modified | Type | Size |
|---|---|---|---|
| temp | 1/14/2015 9:17 AM | File folder | |
| blinky_1.bin | 1/14/2015 9:18 AM | BIN File | 2 KB |
| blinky_1.hex | 1/14/2015 8:52 AM | HEX File | 4 KB |
| blinky_2.bin | 1/14/2015 9:19 AM | BIN File | 2 KB |
| blinky_2.hex | 1/14/2015 8:57 AM | HEX File | 4 KB |
| hex2bin.exe | 10/21/2014 2:39 PM | Application | 55 KB |

**Figure 160: Hex2Bin Example Directory with Files**



```
C:\WINDOWS\system32\cmd.exe

usage: hex2bin [OPTIONS] filename
Options:
  -s [address]  Starting address in hex (default: 0)
  -l [length]   Maximal Length (Starting address + Length -1 is Max Address)
                File will be filled with Pattern until Max Address is reached
                Length must be a power of 2 in hexadecimal:
                     Hex       Decimal
                    1000   =    4096   (4ki)
                    2000   =    8192   (8ki)
                    4000   =   16384  (16ki)
                    8000   =   32768  (32ki)
                   10000   =   65536  (64ki)
                   20000   =  131072 (128ki)
                   40000   =  262144 (256ki)
                   80000   =  524288 (512ki)
                  100000   = 1048576   (1Mi)
                  200000   = 2097152   (2Mi)
                  400000   = 4194304   (4Mi)
                  800000   = 8388608   (8Mi)
  -e [ext]      Output filename extension (without the dot)
  -c            Enable hex file checksum verification
  -p [value]    Pad-byte value in hex (default: ff)

  -k [0|1|2]    Select checksum type
                       0 = 8-bit,
                       1 = 16-bit little endian,
                       2 = 16-bit big endian
  -r [start] [end]     Range to compute checksum over (default is min and max addresses)
  -f [address] [value] Address and value of checksum to force


C:\_hex2bin>hex2bin -c blinky_1.hex
hex2bin v1.0.10, Copyright (C) 2012 Jacques Pelletier & contributors

Lowest address   = 00000000
Highest address  = 00000447
Pad Byte         = FF
8-bit Checksum = 33

C:\_hex2bin>hex2bin -c blinky_2.hex
hex2bin v1.0.10, Copyright (C) 2012 Jacques Pelletier & contributors

Lowest address   = 00000000
Highest address  = 00000477
Pad Byte         = FF
8-bit Checksum = D8

C:\_hex2bin>cd _hex2bin_
```

**Figure 161: Hex2Bin.exe Example**

 © 2020 Dialog Semiconductor

# Appendix H Bin2Image

Figure 163 shows an example of using the `bin2image.exe` utility, which creates a bootable-cached image for DA1468x devices.

The file `bin2image.exe` must be put in the same directory as the file to be converted. Figure 162 shows the directory and the files used in this example.

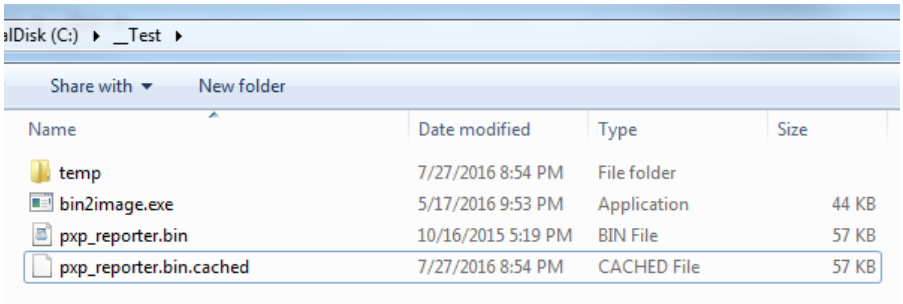



**Figure 162: Bin2Image Example Directory with Files**

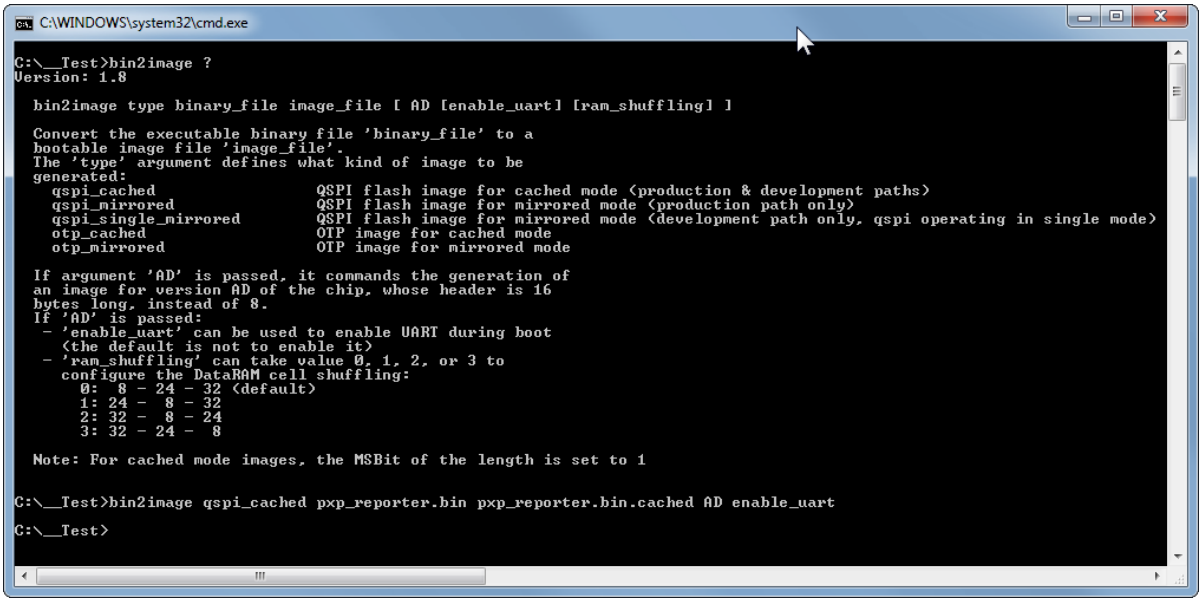



**Figure 163: Bin2Image Example**

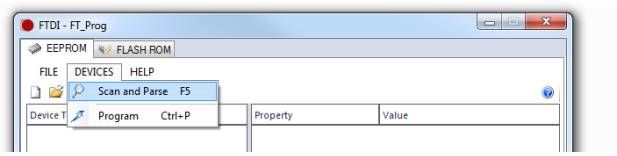
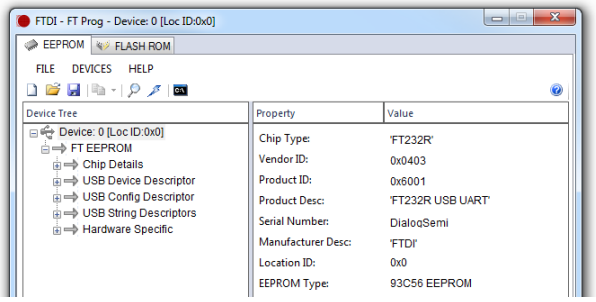
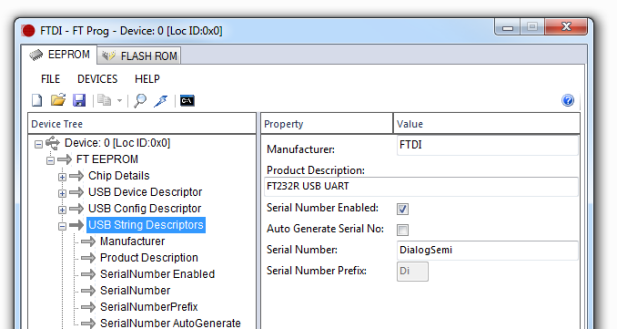

© 2020 Dialog Semiconductor

# Appendix I Automatic GU COM Port Find

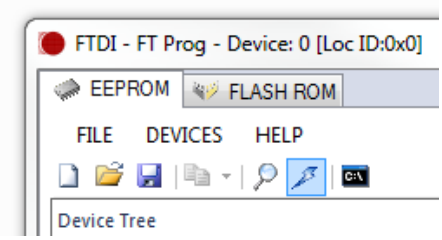
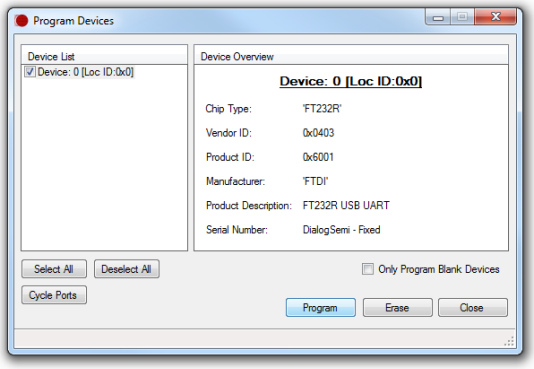
For the GU COM port automatic recognition to operate, a special serial number should exist in the GU FTDI IC. Usually this serial number is programmed during PLT PCB manufacturing, but it may not exist in some older versions.

If the 'GU COM port find' operation does not work, then the steps described in should be followed.

**Table 108: FTDI "DialogSemi" Serial Number**

| Step | Description |
|---|---|
| 1 | Download the FTDI `FT_Prog` tool from http://www.ftdichip.com/Support/Utilities.htm#FT_PROG. |
| 2 | Put power on the PLT board. |
| 3 | Remove any other USB FTDI connection to the PC. |
| 4 | Connect the USB cable of the GU to the PC. |
| 5 | Check the Device Manager that the GU COM port has been found. |
| 6 | Run the `FT_Prog.exe`.<br> <br> |
| 7 | Select 'Devices > Scan and Parse.<br>  <br> |
| 8 | A single 'FT232' device should be found.<br>  <br> |
| 9 | Select 'USB String Descriptors'. |
| 10 | Uncheck the 'Auto Generate Serial No:'. |
| 11 | Edit 'Serial Number' to "`DialogSemi`" as shown below.<br>  <br> |

 © 2020 Dialog Semiconductor

| Step | Description |
|---|---|
| 12 | Press the ‘Flash’ button to program the change to the FTDI IC.<br>  <br> |
| 13 | Press 'Program' in the new window.<br>  <br> |
| 14 | Press ‘Close’. |
| 15 | Unplug and reconnect the GU USB cable to the PC. |
| 16 | Verify the Serial Number change by running `FT_Prog.exe` again and reading the Serial Number value. |

 © 2020 Dialog Semiconductor

# Appendix J Improving Cabling between PLT and DUTs

The following recommendations can be used to improve the connections between PLT and DUTs:

- Keep the lengths of the cables as short as possible.
- When possible use twisted pair cables instead of separate cables for:
  - GND/VBAT
  - GND/TxD
  - GND/RxD
  - GND/VPP
- Use ferrite beads for noise reduction in cables.

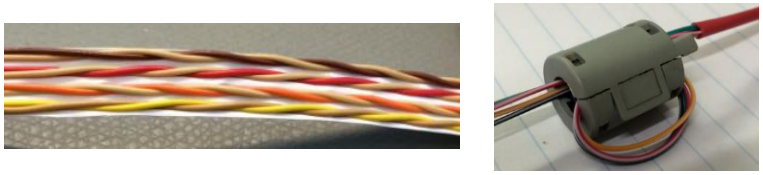


**Figure 164: Example of Twisted Pair Cable with 4 Pairs and Ferrite**

- Connect pull-down resistors at the end of the PLT TX signal lines. Use a 4.7 kΩ resistor at PLT DUT Connector Pin 7 (DUT TX) with the other end connected to ground. In total 16 resistors must be mounted, one for each PLT DUT connector.
- Connect a pull-down resistor as close as possible to the UART RX signal connector on the DUT. The value should be approximately 4.7 kΩ. Connect the other end of the resistor to ground.
- Use gold plated contacts in the connections between the PLT and the DUTs.
- Use extra drivers in the UART lines.
- Use series resistors of approximately 100 Ω in the UART lines, one mounted at the beginning and one at the end of the signal lines.

**Note:** Start with the simple solutions first by testing them one-by-one for stability.

Figure 165 and Figure 166 show examples of some of the above proposals.

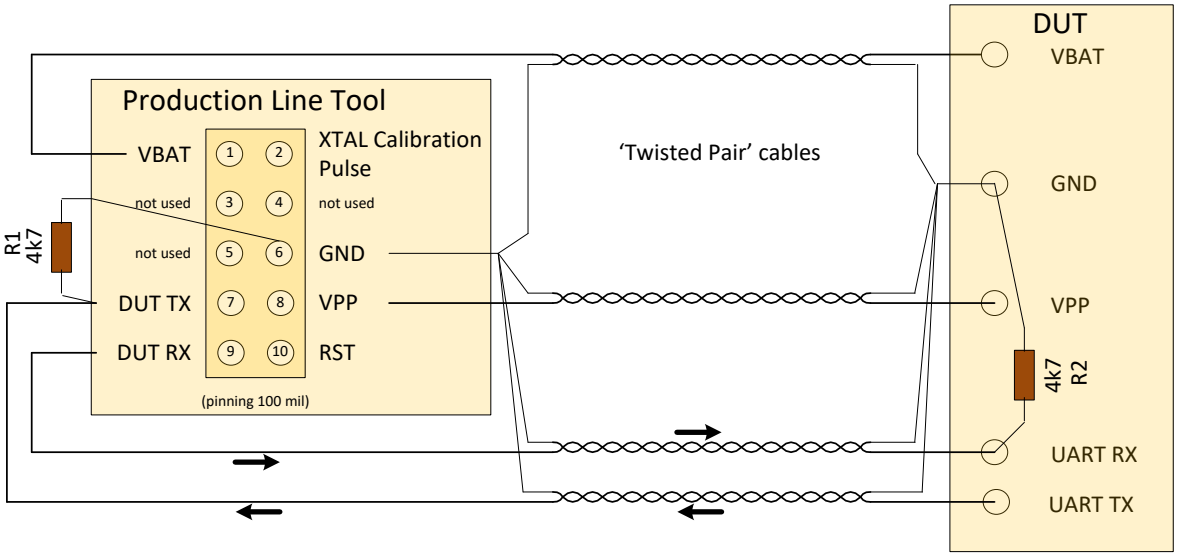



**Figure 165: Location of Pull-Down Resistors**


© 2020 Dialog Semiconductor

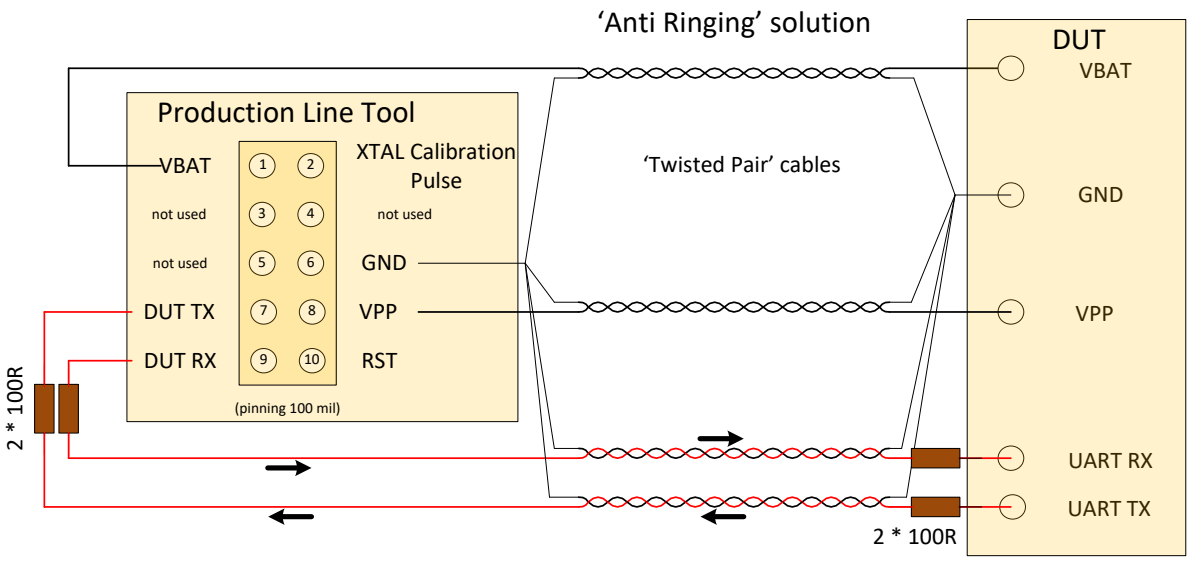



**Figure 166: Anti-Ringing Solution**


© 2020 Dialog Semiconductor

# Appendix K DA14583 Internal SPI Flash Memory

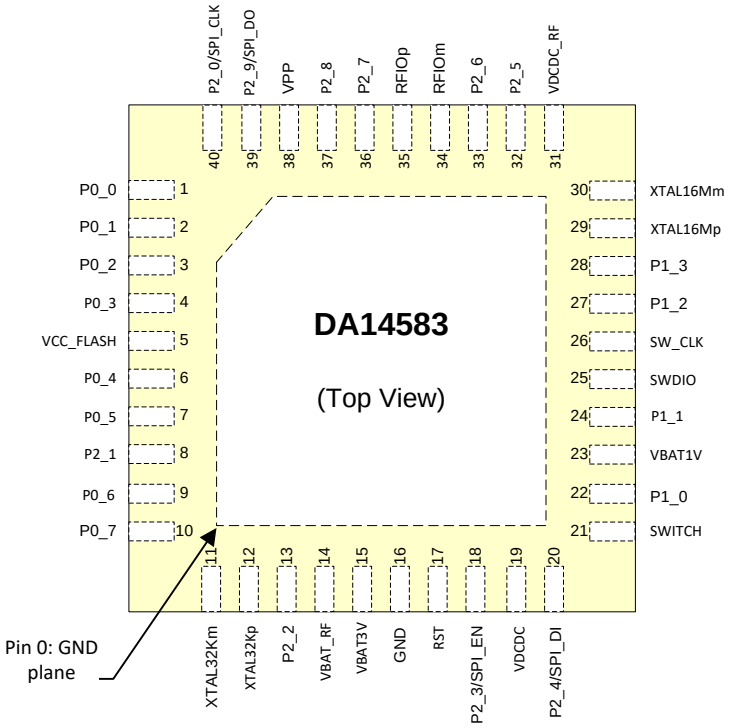



**Figure 167: Pin Assignment of DA14583 – QFN40**

DA14583 is the SPI master in regard to the internal SPI flash memory.

**Table 109: DA14583 Internal SPI Flash Connections**

| DA14583 Pin | SPI Function | Description |
|---|---|---|
| P2_0 | SPI_CLK | SPI Clock |
| P2_4 | SPI_DI (in) | MISO |
| P2_9 | SPI_DO (out) | MOSI |
| P2_3 | SPI_EN | SPI Chip Select |


© 2020 Dialog Semiconductor

# Appendix L DA14586 Internal SPI Flash Memory

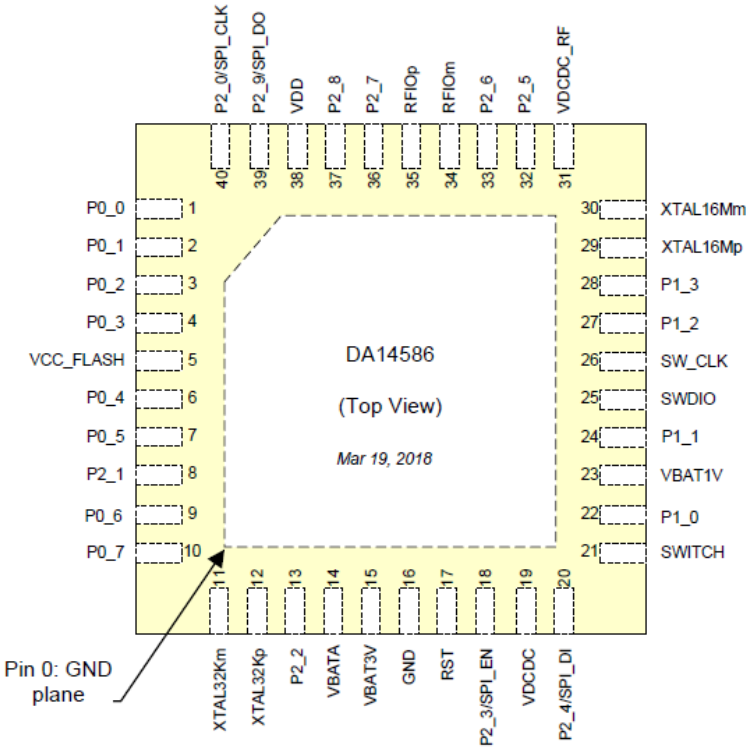



**Figure 168: Pin Assignment of DA14586 - QFN40**

DA14586 is the SPI master in regard to the internal SPI flash memory.

**Table 110: DA14586 Internal SPI Flash Connections**

| DA14586 Pin | SPI Function | Description |
|---|---|---|
| P2_0 | SPI_CLK | SPI Clock |
| P2_4 | SPI_DI (in) | MISO |
| P2_9 | SPI_DO (out) | MOSI |
| P2_3 | SPI_EN | SPI Chip Select |


© 2020 Dialog Semiconductor

# Appendix M Honeywell Xenon 1900 Barcode Scanner Setup

In order to use the Honeywell Xenon 1900 then please follow the steps below to appropriate set it up for PLT usage.

1. Download Xenon-UG.pdf User Guide.
2. Scan Restore factory defaults at page 198 (Resetting the Factory Defaults)
3. Program the USB to Serial Interface. Scan code at Page 32 (TRMUSB130)
4. Download Xenon USB to Serial drivers HSM USB Serial Driver version 3.5.5.zip

 © 2020 Dialog Semiconductor

# Appendix N Golden Unit Upgrade Using Smart Snippets Toolbox

The SPI Flash memory of the Golden Unit can be programmed as any DA14580 device, using the JTAG connector next to the Golden Unit and the Smart Snippets Toolbox application.

1. Connect the power supply to the PLT hardware as described in PLT Power Supply.
2. Connect the USB cable of the Golden Unit.
3. Connect the JTAG (J2) of the Golden Unit.
4. Open the Smart Snippets Toolbox and select the JTAG method and the DA14580 chip.
5. Under the "`Layout`" category, on the "`Booter & Board Setup page`", the GPIOs for the Flash memory should be the following: CLK: P0_0, CS: P0_3, MISO: P0_5, MOSI: P0_6.
6. Under the "`Tools`" category, on the "`SPI Flash Programmer`" tab, using the "`Browse`" button, select the "`prod_test_GU.bin`" binary which is under the SmartBond `SmartBond_PLT_v4.x/binaries/GU/ prod_test_GU.bin` folder on the PLT software package.
7. After the binary is loaded on the Smart Snippets Toolbox, press "`Connect`" at the bottom. This will download a firmware on the Golden Unit and set the SPI Flash GPIOs to program the memory.
8. Select "`Erase`" to completely erase the Flash memory before burning the new firmware.
9. Select "`Burn and Verify`". On the pop-up message, select to make the firmware `bootable`.
10. Remove the JTAG from the PLT hardware and then manually reset the Golden Unit using the reset button next to it. The Golden Unit has now booted with the new firmware.

 © 2020 Dialog Semiconductor

# Appendix O Connecting a Speaker to the Golden Unit for Audio Test

PLT is able to perform audio test for DA14582, DA14585 and DA14586 devices. The audio test settings are described in . A speaker can be connected to the Golden Unit using GPIOs P1_0 (AL4) and P1_1 (AL5) as shown in Figure 169 to generate the 4 kHz tone.

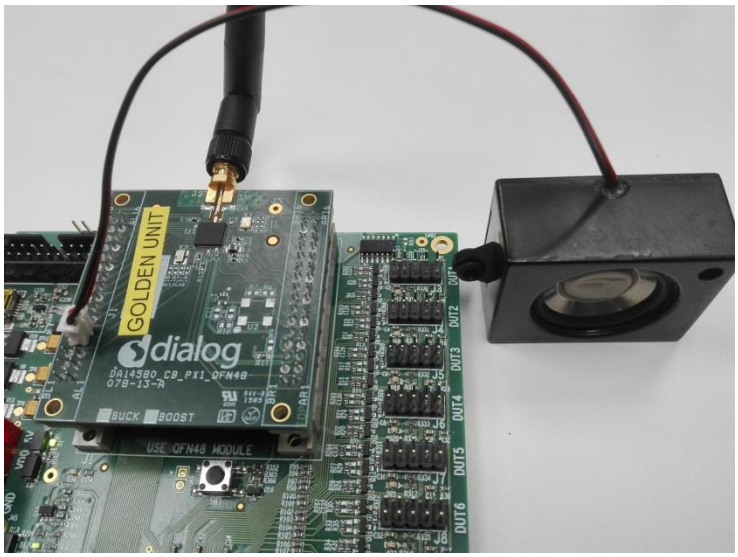



**Figure 169: Speaker Connection for Audio Test.**


© 2020 Dialog Semiconductor

# Appendix P FTDI Driver Removal and Installation

In order to re-install the latest FTDI drivers, the previous should be uninstalled.

FTDI driver removal:

1. Download CDM uninstaller from http://www.ftdichip.com/Support/Utilities.htm#CDMUninstaller.
2. Run `CDMuninstallerGUI.exe`
3. The VID/PID of the PLT FTDIs are VID=0403/PID=6011 for the DUTs and VID=0403/PID=6001 for the GU.
   Enter these VIDs and PIDs in the CDM Uninstaller and press add for each one.
4. Then press `Remove Devices` to uninstall the FTDI drivers.
5. Un-plug both USB cables.

More information can be found in the following link:

http://www.ftdichip.com/Support/Utilities/CDM_Uninst_GUI_Readme.html

FTDI driver installation:

1. Download the latest drivers from http://www.ftdichip.com/Drivers/VCP.htm and install them using the executable.
2. After uninstalling the drivers, plug in both USB cables. Windows will automatically assign the new drivers. Do not remove the cables during driver installation. A driver installation error may occur and the removal-installation will have to be repeated.
3. Check in the Windows Device manager that the driver versions of the 17 PLT COM Ports->USB Serial Ports are the latest.

FTDI driver versions v2.12.24, v2.12.26 and 2.12.28 have been tested.

 © 2020 Dialog Semiconductor

# Appendix Q DA1458x DK Pro Motherboard Connection

Figure 170 shows the wiring to a DA14580/1/2/3/5/6 Pro DK motherboard.

As described in DUT Connector the following connections are needed to connect a DUT to the PLT.

1. Ground. DUT connector pin6 <-> Pro DK J5 pin2.
2. VBAT. DUT connector pin1 <-> Pro DK J5 pin1.
3. UART Tx. DUT connector pin7 <-> Pro DK J5 pin11.
4. UART Rx. DUT connector pin9 <-> Pro DK J5 pin13.
5. Reset. DUT connector pin10 <-> Pro DK J5 pin3 (Optional).

If no power supply is provided through the USB cable (J12), the reset circuit will drive the reset pin of the DUT host board (connector J4) high, keeping the DUT at a reset state. To overcome this either the R84 resistor should be removed or the USB cable should be connected.

Additionally, J11 jumper should be removed. Power supply to the board will be provided from the PLT HW (VBAT line).

To use the onboard SPI flash memory J5 jumper configuration should be as shown on the silkscreen PCB print (left of J5). A special jumper must be used to connect the `P0_5` (J5 – pin13) with the `SPI MISO` (J6 – pin2) and the `PLT UART-Rx pin` (on DUT connector – pin9).

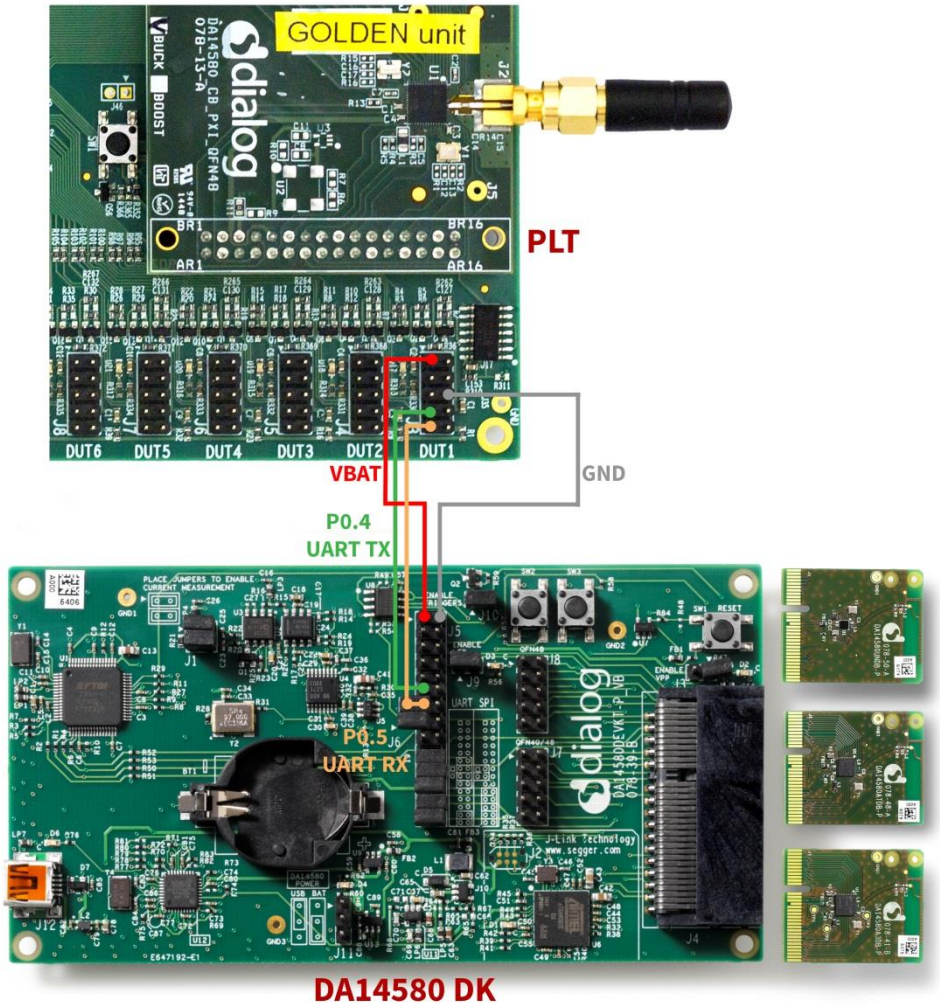



**Figure 170: DA14580/5 Pro Motherboard DK Wiring.**


© 2020 Dialog Semiconductor

# Appendix R DA1468x DK Pro Motherboard Connection

Figure 171 shows the wiring to a Pro DK motherboard for the DA14680/1/2/3 DUTs.

As described in DUT Connector the following connections are needed to connect a DUT to the PLT:

1. Ground. DUT connector pin6 <-> Pro DK J3 pin22.
2. VBAT. DUT connector pin1 <-> Pro DK J3 pin24.
3. UART Tx. DUT connector pin7 <-> Pro DK J4 pin8.
4. UART Rx. DUT connector pin9 <-> Pro DK J3 pin19.
5. Reset. DUT connector pin10 <-> Solder to Reset button (K2) (Optional).

All board jumpers must be removed. Power supply will be provided from the PLT HW (VBAT line).

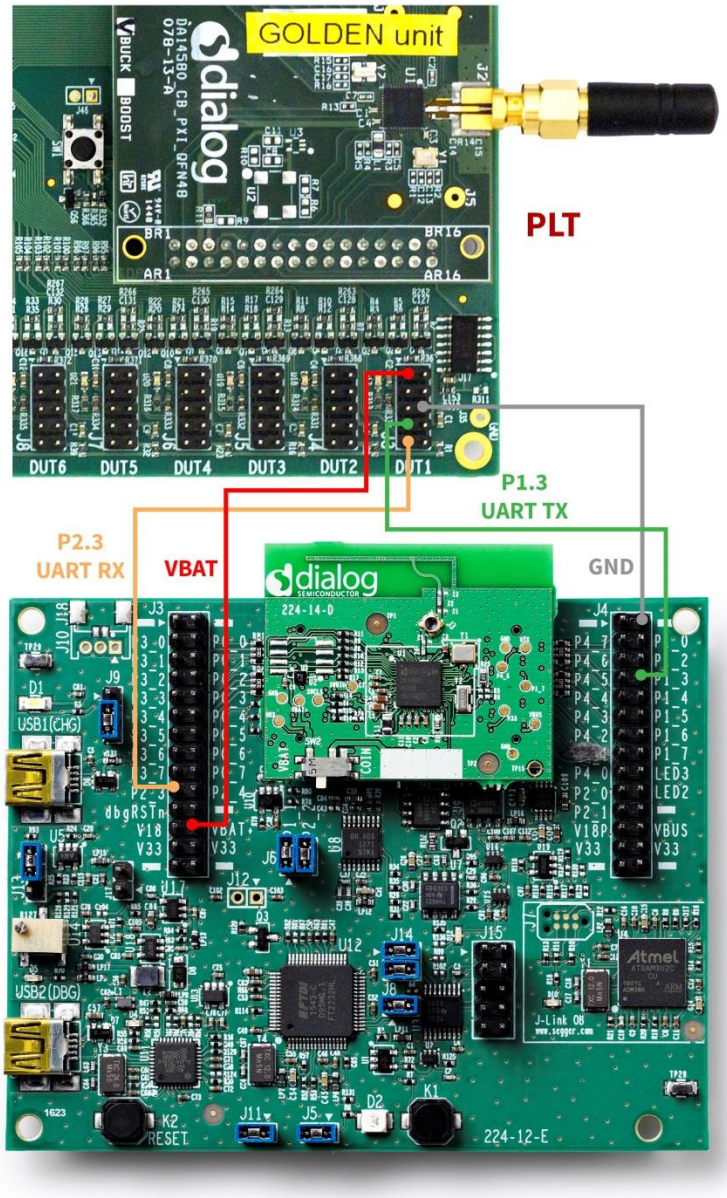



**Figure 171: DA14680/1/2/3 Pro Motherboard DK Wiring.**


© 2020 Dialog Semiconductor

# Appendix S Connecting DA1468x DK Pro Motherboard for Current Measurements

DA1468x DK Pro motherboards are equipped with current measurement modules. Therefore, they can operate as external ammeter instruments with the PLT.

The DA1468x DK Pro motherboard should be connected to J26 as shown in Figure 172. A three-wire connection must be made between the PLT board and the motherboard. Jumpers J9 should be removed from the DA1468x DK Pro motherboard. It is mandatory to have a common ground between the two boards. One possible ground connection is shown with the purple line.

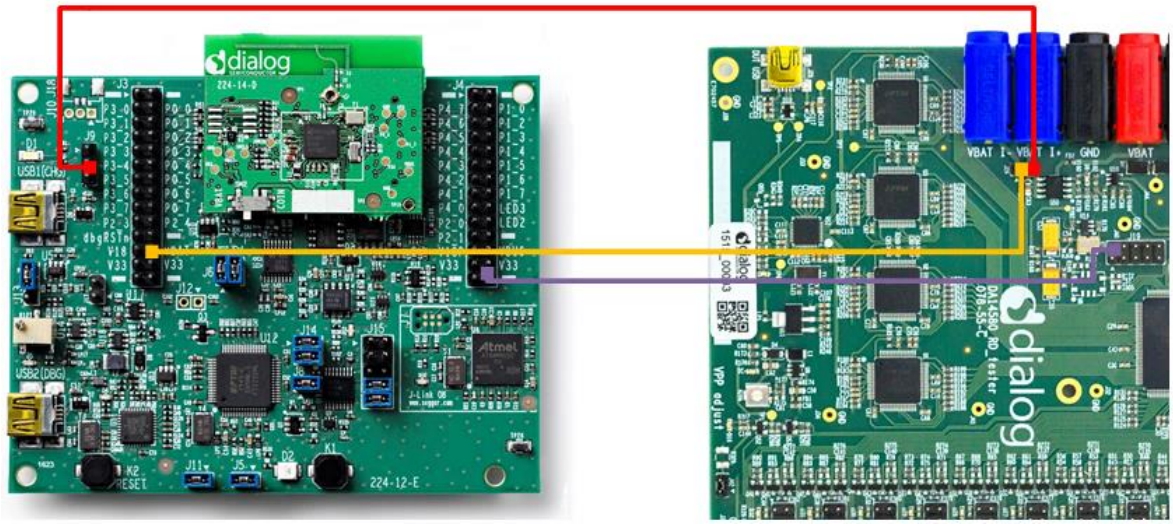



**Figure 172: DA1468x PRO DK Ammeter Connection with PLT**

To use the DA1468x Pro motherboard as current measurement instrument in PLT, the `ammeter_da1468x_dk.dll` should be selected in Current Measurement Test (DA1458x) or

Current Measurement Test (DA1469x) test settings panel. The interface should be set to the second FTDI COM port enumerated in Windows, as shown in Figure 173.

**Figure 173: DA1468x DK PRO Current Measurement Settings**


© 2020 Dialog Semiconductor

# Appendix T Connecting DUT with Battery Supply

Wiring connections to a battery powered DUT is described in DUT Connector. Example connections can be found in chapters DA1458x DK Pro Motherboard Connection and DA1468x DK Pro Motherboard Connection.

1. Four wires are mandatory for the connection:
   - Common Ground
   - UART Tx
   - UART Rx
   - Reset line
2. mode is the only mode supporting battery powered DUTs since POR cannot be performed. In order for the PLT to perform a reset on the DUTs, the VBAT line of each DUT connector must be connected to the reset line of the DUT as described in mode.
3. Current measurement is not supported, since there is no way to measure the current of the DUTs.
4. In order to program the OTP for the DA14580/1/2/3 DUTs an external VPP voltage must be supplied. VPP lines on the DUT connectors will not provide any voltage in mode so they cannot be used.

In order to have the least possible wiring connections, UART Rx line can also be used as input GPIO for the pulse used during the XTAL Trim procedure.

 © 2020 Dialog Semiconductor

# Appendix U User Interfaces Shortcut Keys

**Table 111: User Interface Shortcut Keys**

| Application | Shortcut | Description |
|---|---|---|
| CFG PLT Application | `Ctrl + S` | This shortcut is the equivalent of pressing the save button at the bottom of the screen. |
| GUI PLT Application | `Space-Bar` | It is used to select the 'Start' button to start testing. |
| | `F` | It is the equivalent of pressing the 'Finished' button with the left mouse-click. |

 © 2020 Dialog Semiconductor

# Appendix V DA1458x Supported SPI Flash\EEPROM Memories

Table 112 describes the supported SPI Flash and EEPROM memories for DA1458x devices. To use a memory not shown in the list, SPI Flash Configuration or I2C EEPROM Configuration can be used.

**Table 112: DA1458x Supported SPI Flash Memories**

| Memory type | Memory vendor | Product number |
|---|---|---|
| SPI Flash memory | Windbond | W25X10 |
| | | W25X20 |
| | | W25X40 |
| | Adesto | AT25Dx011 |
| | | AT25XE021 |
| | | AT25XE041 |
| | Macronix | MX25V1006E |
| | | MX25R1035F |
| | | MX25R2035F |
| | | MX25R4035F |
| | | MX25R8035F |
| | | MX25R1635F |
| | | MX25V1035F |
| | | MX25V2035F |
| | | MX25V4035F |
| | | MX25V8035F |
| | | MX25V1635F |
| I2C EEPROM | ST | M24M02 |

 © 2020 Dialog Semiconductor

# Appendix W DA14531 Supported SPI Flash Memories

Table 113 describes the supported SPI Flash memories for DA14531 devices. To use a memory not shown in the list, SPI Flash Configuration can be used.

**Table 113: DA14531 Supported SPI Flash Memories**

| Memory type | Memory vendor | Product number |
|---|---|---|
| SPI Flash memory | Windbond | W25X10 |
| | | W25X20 |
| | Adesto | AT25Dx011 |
| | Macronix | MX25V1006E |
| | | MX25R1035F |
| | | MX25R2035F |
| | | MX25R4035F |
| | | MX25R8035F |
| | | MX25R1635F |
| | | MX25V1035F |
| | | MX25V2035F |
| | | MX25V4035F |
| | | MX25V8035F |
| | | MX25V1635F |
| | Puya | P25Q10U |
| | | P25Q40U |
| | GigaDevice | GD25WD10 |
| | | GD25WD20 |

 © 2020 Dialog Semiconductor

# Appendix X DA1468x Supported QSPI Flash Memories

Table 112 describes the supported QSPI flash memories for DA1468x devices. In order to support a QSPI flash memory that is not in the list, follow the `DA1468x QSPI Flash Support {#flash_support}` instructions in `readme.md` file located under `DA1468x_DA15xxx_SDK_1.0.12.1078\DA1468x_DA15xxx_SDK_1.0.12.1078\sdk\bsp\memory\` folder to manually add it inside the `uartboot.bin` firmware.

**Table 114: DA1468x Supported QSPI Flash Memories**

| Memory vendor | Product number |
|---|---|
| Windbond | W25Q80EW |
| Gigadevice | GD25LQ80B |
| Macronix | MX25U51245 |
| ISSI | IS25LP128 |

# Appendix Y DA1469x Supported QSPI Flash Memories

describes the supported QSPI flash memories for DA1469x devices. In order to support a QSPI flash memory that is not in the list, follow the `DA1469x QSPI Flash Support {#flash_support}` instructions in `readme.md` file located under `SDK_10.0.8.105\sdk\bsp\memory\` folder to manually add it inside the `uartboot.bin` firmware.

**Table 115: DA1469x Supported QSPI Flash Memories**

| Memory vendor | Product number |
|---|---|
| Windbond | W25Q32FW |
| Gigadevice | GD25LE32 |
| Macronix | MX25U3235 |

 © 2020 Dialog Semiconductor

# Appendix Z BLE Tester Measurement Results

When using an MT8852B as an external BLE tester instrument for the DA14531 RF Tests and DA1469x RF Tests, PLT will instruct the MT8852B to perform specific tests and then wait for its reply. MT8852B replies with a string, containing the command code of the test performed followed by the results of the test or an Error Response string.

Table 116 shows the result command codes.

The PDF in the following link describes the format of the result string for each command code, under chapter 15 – 7.

https://dl.cdn-anritsu.com/en-au/test-measurement/files/Manuals/Programming-Manual/MT8852B/MT8852B-Bluetooth%20tester-Programming%20Manual%20Rev%20X.pdf

**Table 116: MT8852B Supported Command Codes**

| Code | Test |
|---|---|
| LEOP0 | TX power |
| LEICD0 | Carrier frequency and Drift |
| LEMI | Modulation index |
| ERRLST | Error response |

As an example, the test results of DUT1 in the example of CSV Log File Contents will be used.

## Tx Power

The result of DUT1 for the TX-Power test is:

```
TRUE;-14.25;-14.25;-14.25;0.10;0;2;PASS
```

**Table 117: MT8852B – BLE TX Output Power Test Results**

| Description | Format | Example |
|---|---|---|
| Results valid | TRUE \| FALSE | TRUE |
| Packet average power in dBm | floating point | -14.25 |
| Test avg max in dBm | floating point | -14.25 |
| Test avg min in dBm | floating point | -14.25 |
| Test peak to average power in dBm | floating point | 0.10 |
| Number of failed packets | integer | 0 |
| Number of tested packets | integer | 2 |
| Pass/fail result | PASS \| FAIL | PASS |

 © 2020 Dialog Semiconductor

### Frequency Offset

The result of DUT1 for the Frequency Offset test is:

```
TRUE;-2.500e+003;2.800e+003;-7.800e+003;-7646;-9.0e+003;-9.0e+003;0;2;PASS;-7646
```

**Table 118: MT8852B – BLE Carrier Frequency Offset and Drift Test Results**

| Description | Format | Example |
|---|---|---|
| Drift rate valid | TRUE \| FALSE | TRUE |
| Average Fn | Integer | -2.500e+003 |
| Maximum Positive Fn | Integer | 2.800e+003 |
| Minimum Negative Fn | integer | -7.800e+003 |
| Drift rate | integer | -7646 |
| Average drift | integer | -9.0e+003 |
| Maximum drift | integer | -9.0e+003 |
| Packets Failed | integer | 0 |
| Packets Tested | integer | 2 |
| Pass/fail result | PASS \| FAIL | PASS |
| Initial drift rate | integer | -7646 |

### Modulation Index

The result of DUT1 for the Modulation Index test is:

```
TRUE;282100.00;249100.00;200700.00;248700.00;1.00;0;576;1;1;FAIL;100.00%
```

The tester responded with FAIL because two tests with different patterns were needed. The overall result of the Modulation Index test will be concluded in a second step, after the second payload is tested.

**Table 119: MT8852B – BLE Modulation Characteristics Test Results**

| Description | Format | Example |
|---|---|---|
| Results valid | TRUE \| FALSE | TRUE |
| Delta f1 max in Hz | floating point | 282100.00 |
| Delta f1 average in Hz | floating point | 249100.00 |
| Delta f2 max in Hz<br>(Delta f1 max lowest for BLR8) | floating point | 200700.00 |
| Delta f2 average in Hz<br>(omitted for BLR8) | floating point | 248700.00 |
| Delta f2 avg / delta f1 avg<br>(Omitted for BLR8) | floating point | 1.00 |
| Delta f2 max Failed limit<br>(Delta f1 max Failed limit for BLR8) | integer | 0 |
| Delta f2 max count<br>(Delta f1 max count for BLR8) | integer | 576 |
| Packets failed | integer | 1 |
| Packets tested | integer | 1 |
| Pass/fail result | PASS \| FAIL | FAIL |

 © 2020 Dialog Semiconductor

| Description | Format | Example |
|---|---|---|
| Delta f2 max % pass rate<br>(Delta f1max % pass rate for BLR8) | floating point | 100.00% |

### Error Response

If the BLE Tester fails to perform the tests it will respond with an error. Table 120 describes the parts of the error message. More details can be found in chapter 4.3 of the following link:

https://dl.cdn-anritsu.com/en-au/test-measurement/files/Manuals/Programming-Manual/MT8852B/MT8852B-Bluetooth%20tester-Programming%20Manual%20Rev%20X.pdf

The format of the response message is

`ABCCDDEFGHIIJKK!LLLLLLL!MMMMMMM!NNNNNNN!OOOOOOO!`

A common error response, which is not an actual error, is the message below:

`000000000000000!NO ERRORS!NO ERRORS!!!`

This is due to the output received signal may be less than -50dB.

**Table 120: MT8852B – Error List**

| Alias | Error | Status | Description |
|---|---|---|---|
| A | CONNECTION ALREADY EXISTS | 0 | No previous connection |
| | | 1 | Connection already exists |
| B | EUT TEST MODE STATE | 0 | EUT Test Mode enabled |
| | | 1 | EUT Test Mode not enabled |
| CC | EUT HCI ERROR | 00 | OK |
| | | XX | 2-digit hex error code (EUT controlled via RS232) |
| DD | INTERNAL HCI ERROR | 00 | OK |
| | | XX | 2-digit hexadecimal error code |
| E | INTERNAL SYNC ERROR | 0 | OK |
| | | 1 | Internal HCI synchronization error |
| F | EUT SYNC ERROR | 0 | OK |
| | | 1 | EUT HCI synchronization error (control via RS232) |
| G | EUT HARDWARE ERROR | 0 | OK |
| | | 1 | EUT Reported HCI Hardware error message |
| H | REQUEST FAILED | 0 | OK |
| | | 1 | Request failed (system busy) |
| II | DSP STATUS<br>**Note:** Setting of the DSP status code will not set the DDE bit of the event register | 00 | OK |
| | | 01 | Searching channel |
| | | 02 | Searching sync word |
| | | 03 | Incorrect packet length |
| | | 04 | No payload |
| | | 05 | Auto ranging |
| | | 06 | Incorrect packet |
| | | 07 | Incorrect packet type |

 © 2020 Dialog Semiconductor

| Alias | Error | Status | Description |
|---|---|---|---|
| | | 08 | Over range |
| | | 09 | Under range |
| | | 10 | Invalid payload |
| | | 11 | Error finding start of packet using power profile |
| | | 12 | Error locating P0/GFSK sync word |
| | | 13 | Location of P0/GFSK sync word exceeds allowed limits |
| | | 14 | Error locating EDR sync word |
| | | 15 | Location of EDR sync word exceeds allowed limits |
| | | 16 | Error decoding the packet type field |
| | | 17 | Modulation mode of PI/4-DQPSK or 8DPSK not specified |
| | | 18 | pi/4-DQPSK modulation does not match with detected packet type |
| | | 19 | 8DPSK modulation does not match with packet type |
| | | 20 | Invalid packet type decoded |
| | | 21 | Unknown packet type decoded |
| | | 22 | Expected and measured packet lengths do not match |
| | | 23 | Insufficient blocks in packet for measurement |
| J | EUT BT ADDRESS | 0 | OK |
| | | 1 | EUT Bluetooth Address set (in Manual mode) |
| KK | HCI COMM STATUS | 00 | OK |
| | | 01 | Unknown HCI command |
| | | 02 | No connection |
| | | 03 | Hardware failure |
| | | 04 | Paging timeout |
| | | 05 | Connection timeout |
| | | 06 | Unsupported feature parameter |
| | | 07 | Connection ended by user |
| | | 08 | Low resource connection ended |
| | | 09 | Power Off connection ended |
| | | 10 | Local host connection ended |
| | | 11 | Unsupported remote feature |
| | | 12 | Role change not allowed |
| | | 13 | LMP response timeout |
| | | 14 | IQ modem DAC saturation |
| LLLLLLL | | | Internal core error text (variable length) |
| MMMMMMM | | | EUT core error text (variable length) |
| NNNNNNN | | | Last GPIB command that caused a Command error (variable length) |
| OOOOOOO | | | Last GPIB command that caused an Execution error (variable length) |

 © 2020 Dialog Semiconductor

# Appendix AA Memory programming

This appendix lists all possible memory programming operations. The memory operations are grouped per chipset and memory type.

 © 2020 Dialog Semiconductor

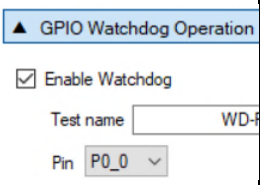
## AA.1 DA14580/1/2/3/5/6 Memory Programming Tests

### OTP

**Table 121: DA1458x Memory Programming – OTP Memory**

| Test Name | Operation | No. | Description |
|---|---|---|---|


© 2020 Dialog Semiconductor

| Test Name | Operation | No. | Description |
|---|---|---|---|
| GPIO Watchdog Operation<br>  <br><br>**Figure 49: GPIO Watchdog Operation – DA14531**<br><br>Table 40 describes the GPIO watchdog configuration options. When this feature is enabled, firmware will continuously toggle the selected GPIO at specific intervals, less than 4seconds.<br><br>**Table 40: GPIO Watchdog Configuration Options – DA14531**<br><br>**Option** / **Description**<br>Enable Watchdog: The checkbox enables the operation. It enables the continuous toggling of a GPIO during the whole production testing and memory programming procedure, except during firmware download. The pulse on the GPIO has approximately 1.5% duty cycle and 0.48Hz frequency.<br>Test name: The test name to be used for logging purposes<br>Pin: The GPIO to toggle | | | |
| 8.1.1.1 VBAT Level Log<br>CFR0012 | | | |



© 2020 Dialog Semiconductor

| Test Name | | Operation | No. | Description |
|---|---|---|---|---|
| **Error! Reference source not found.** | | Write\ Verify | 1 | This operation writes the user selected binary into the OTP memory. The start address will always be address 0.<br>If `Verify` is selected, PLT will read the OTP contents for the size of the binary burned and compare it with the actual binary.<br>If the binary size is larger than the OTP image area, PLT considers that the binary contains the header as well.<br>PLT can also program the following during OTP image burn.<br>● DMA length |
| OTP Header | | Write\ Verify | 1 | The OTP header write operation will write any non-zero header field one by one. If the `Verify` option is selected it will read the OTP header fields and compare them to the ones written before. |
| BD Address | | Write\ Verify | 1 | The BD address write procedure will write the BD address field in the OTP header area. If the `Verify` option is selected, PLT will read the OTP BD address and compare it to the one written before. |
| | | Read\ Compare | 1 | This will read the BD address field from the OTP header and save it to the DUT logs. If the `Compare` option is selected it will compare the address read with the one PLT uses for the particular device. |
| Memory read | | Read | 10[1] | The memory read procedure can read any OTP memory area and save the results in the DUT logs. |
| Custom Memory Data | Barcode scanner | Write\ Verify | 1 | This procedure will write data with a given size at a given address offset. A barcode scanner is used as data input. If the `Verify` option is selected, PLT will read the contents and compare it to the one written before. |
| | CSV file | | 5[1] | This procedure will read the memory programming parameters from a CSV file. Up to 5 memory writes can be set.<br>If the `Verify` option is selected, PLT will read the memory contents and compare them to the ones written before. |
| | Manual | | 1 | The manual procedure will write data with a given size at a given address offset. If the `Verify` option is selected, PLT will read the memory contents and compare them to the ones written before. |

**Note 2** Applies for all available memories simultaneously.

**Note 3** `Check Empty` is performed by reading the memory contents and checking whether these are all zeros.

**Note 4** `Check if data match` is performed by reading the memory contents and checking whether these are all zeros or the data to be written are the same.

**Note 5** If the OTP part to be written contains all-zero values, then the write operation will not be performed. This also applies for writing default values in the OTP header.

### SPI Flash

**Table 122: DA1458x Memory Programming - SPI Flash**

| Test Name | Operation | No. | Description |
|---|---|---|---|
| SPI Flash Configuration | Initialize | 1 | The memory programming firmware can auto-detect the memories listed in Table 112.<br>However, PLT can also use a user-defined SPI flash configuration. PLT will send a command containing the SPI configuration set by the user. For the DA14583 and DA1586 devices, this option is disabled because PLT initializes the firmware with the internal SPI flash characteristics. |
| SPI Flash Memory | Erase | 10 | The erase operation can perform up to 10 erase tests, which can erase either the entire memory or specific sections. |
| | Check Empty | 10 | This can perform up to 10 different check empty operations, which will verify that specific sections or the entire memory is empty. |

 © 2020 Dialog Semiconductor

| Test Name | | Operation | No. | Description |
|---|---|---|---|---|
| | | Write\ Verify | 10 | This operation will write a user defined binary at a given offset or at offset 0 if it is a bootable image. If the `Verify` option is selected, PLT will read the contents and compare it to the one written before. |
| Memory read | | Read | 10[1] | The memory read procedure can read any field of the SPI memory and save the data into the DUT logs. |
| Custom Memory Data | Barcode scanner | Write\ Verify | 1 | This procedure will write data with a given size at a given address offset. A barcode scanner is used as data input. If the `Verify` option is selected, PLT will read the contents and compare it to the one written before. |
| | CSV file | | 5[1] | This procedure will read the memory programming parameters from a CSV file. Up to 5 memory writes can be set.<br>If the `Verify` option is selected, PLT will read the memory contents and compare them to the ones written before. |
| | Manual | | 1 | The manual procedure will write data with a given size at a given address offset. If the `Verify` option is selected, PLT will read the memory contents and compare them to the ones written before. |

**Note 6** Applies for all available memories simultaneously.

 © 2020 Dialog Semiconductor

### EEPROM

**Table 123: DA1458x Memory Programming – EEPROM Memory**

| Test Name | | Operation | No. | Description |
|---|---|---|---|---|
| I2C EEPROM Configuration | | Initialize | 1 | The memory programming firmware can auto-detect the memories listed in Table 112.<br>However, PLT can also use a user-defined EEPROM configuration. PLT will send a command containing the EEPROM configuration set by the user. |
| I2C EEPROM Memory | | Write\ Verify | 10 | This operation will write a user defined binary at a given offset or at offset 0 if it is a bootable image. If the `Verify` option is selected, PLT will read the contents and compare it to the one written before. |
| Memory read | | Read | 10[1] | The memory read procedure can read any field of the SPI memory and save the data into the DUT logs. |
| Custom Memory Data | Barcode scanner | Write\ Verify | 1 | This procedure will write data with a given size at a given address offset. A barcode scanner is used as data input. If the `Verify` option is selected, PLT will read the contents and compare it to the one written before. |
| | CSV file | | 5[1] | This procedure will read the memory programming parameters from a CSV file. Up to 5 memory writes can be set.<br>If the `Verify` option is selected, PLT will read the memory contents and compare them to the ones written before. |
| | Manual | | 1 | The manual procedure will write data with a given size at a given address offset. If the `Verify` option is selected, PLT will read the memory contents and compare them to the ones written before. |

**Note 7** Applies for all available memories simultaneously.

 © 2020 Dialog Semiconductor

## AA.2 DA1468x Memory Programming Operation

### OTP

**Table 124: DA1468x Memory Programming – OTP Memory**

| Test Name | | Operation | No. | Description |
|---|---|---|---|---|
| XTAL Trim | | Write\ Verify | 1 | The XTAL trim procedure will first find an available OTP TCS field to write the value even if another XTAL trim value is programmed at a previous TCS field. If the `Verify` option is selected it will read the OTP TCS section field and compare them to the one written before. |
| OTP Memory | | Write\ Verify | 1 | This operation writes the user selected binary into the OTP memory. The start address will always be address 0.<br>If `Verify` is selected, PLT will read the OTP contents for the size of the binary burned and compare it with the actual binary.<br>If the binary size is larger than the OTP image area, PLT considers that the binary contains the header as well.<br>PLT can also program the following during OTP image burn.<br>● DMA length<br>● Image CRC |
| **Error! Reference source not found.** | | Write\ Verify | 1 | The OTP header write operation will write any non-zero header field one by one. If the `Verify` option is selected it will read the OTP header fields and compare them to the ones written before. |
| | | Write\ Verify | 1 | The BD address write procedure will write the BD address field in the OTP header area. If the `Verify` option is selected, PLT will read the OTP BD address and compare it to the one written before. |
| | | Read\ Compare | 1 | This will read the BD address field from the OTP header and save it to the DUT logs. If the `Compare` option is selected it will compare the address read with the one PLT uses for the particular device. |
| Memory read | | Read | 10[1] | The memory read procedure can read any OTP memory area and save the results in the DUT logs. |
| Custom Memory Data | Barcode scanner | Write\ Verify | 1 | This procedure will write data with a given size at a given address offset. A barcode scanner is used as data input. If the `Verify` option is selected, PLT will read the contents and compare it to the one written before. |
| | CSV file | | 5[1] | This procedure will read the memory programming parameters from a CSV file. Up to 5 memory writes can be set.<br>If the `Verify` option is selected, PLT will read the memory contents and compare them to the ones written before. |
| | Manual | | 1 | The manual procedure will write data with a given size at a given address offset. If the `Verify` option is selected, PLT will read the memory contents and compare them to the ones written before. |

**Note 8** Applies for all available memories simultaneously.

**Note 9** `Check Empty` is performed by reading the memory contents and checking whether these are all zeros.

**Note 10** `Check if data match` is performed by reading the memory contents and checking whether these are all zeros or the data to be written are the same.

**Note 11** If the OTP part to be written contains all-zero values, then the write operation will be omitted. This also applies for writing default values in the OTP header.

 © 2020 Dialog Semiconductor

### QSPI

**Table 125: Memory Programming – QSPI Memory**

| Test Name | | Operation | No. | Description |
|---|---|---|---|---|
| XTAL Trim | | Write\ Verify | 1 | The XTAL trim QSPI write procedure will write the XTAL trim value found during the calibration process to the user defined QSPI address. If the `Verify` option is selected, PLT will read the QSPI user defined address contents and compare them to the ones written before. |
| QSPI Flash Memory | | Erase | 10 | The erase operation can perform up to 10 erase tests, which can erase either the entire memory or specific sections. |
| | | Check Empty | 10 | This can perform up to 10 different check empty operations, which will verify that specific sections or the entire memory is empty. |
| | | Write\ Verify | 10 | This operation will write a user defined binary at a given offset or at offset 0 if it is a bootable image. If the `Verify` option is selected, PLT will read the contents and compare it to the one written before. |
| QSPI Header - BD Address | | Write\ Verify | 1 | The BD address write procedure will write the BD address field in the QSPI user defined address. If the `Verify` option is selected, PLT will read the data from QSPI user defined address and compare them with the ones written before. |
| | | Read\ Compare | 1 | This will read the BD address field from the QSPI user defined address and save it to the DUT logs. If the `Compare` option is selected it will compare the address read with the one PLT uses for the particular device. |
| Memory read | | Read | 10[1] | The memory read procedure can read any field of the SPI memory and save the data into the DUT logs. |
| Custom Memory Data | Barcode scanner | Write\ Verify | 1 | This procedure will write data with a given size at a given address offset. A barcode scanner is used as data input. If the `Verify` option is selected, PLT will read the contents and compare it to the one written before. |
| | CSV file | | 5[1] | This procedure will read the memory programming parameters from a CSV file. Up to 5 memory writes can be set.<br>If the `Verify` option is selected, PLT will read the memory contents and compare them to the ones written before. |
| | Manual | | 1 | The manual procedure will write data with a given size at a given address offset. If the `Verify` option is selected, PLT will read the memory contents and compare them to the ones written before. |

**Note 12** Applies for all available memories simultaneously.


© 2020 Dialog Semiconductor

# Appendix BB CSV Log File Contents

Table 126 describes the CSV File columns generated during PLT testing. In general, not all CSV file columns shown in Table 126 will be printed, but only those that relate to the enabled tests and memory operations user has selected. An example is given in CSV Log File Example.

**Table 126: CSV File Contents**

| Header | Value | Description |
|---|---|---|
| Start time | hh:mm:ss | Shows the actual time the test procedure has started |
| End time | hh:mm:ss | Shows the actual time the test procedure has ended |
| Tester ID | ID | The tester ID name. |
| DUT | 1-16 | The PLT connector for this device.<br>Values 1-16. |
| BD address | XX:XX:XX:XX:XX:XX | The BD address assigned for this device. |
| Overall status | PASS\FAIL | The overall status of the test procedure for this device. |
| COM Port | XX | Windows assigned COM Port |
| Temperature test | PASS\FAIL | Shows the temperature measured during temperature test.<br>The first column shows the result of the test.<br>The second column shows the temperature measured. |
| Temperature | xx.xx | |
| FW download 1 | PASS\FAIL | Production test FW download.<br>The first column shows the result of the production test firmware download procedure.<br>The second column shows the path to the firmware. |
| FW path 1 | C:\folder\to\bin | |
| RAM FW download | PASS\FAIL | Production test FW download through memory programming FW.<br>The first column shows the result of the production test firmware download procedure.<br>The second column shows the path to the firmware. |
| RAM FW path | C:\path\to\bin | |
| FW version get 1 | PASS\FAIL | Production test FW version.<br>The first column shows the result of the test.<br>The second column shows the production test FW version read back from each device. |
| FW version 1 | e.g. “v_5.0.4_PLT_v4.3” | |
| GPIO Watchdog [‘Test Name’] | PASS\FAIL | GPIO watchdog toggling test for production testing FW |
| GPIO Watchdog mem [‘Test Name’] | PASS\FAIL | GPIO watchdog toggling test for memory programming FW |
| ADC VBAT | PASS\FAIL | Whether VBAT was successfully measured. |
| VBAT level | VBAT level | The level of the VBAT as measured by the internal DUT ADC |
| DC-DC level test | PASS\FAIL | Whether the DA14531 DC-DC level is inside the user defined limits |
| DC-DC level | DC-DC level | The DC-DC level as measured from the internal DUT ADC. |

 © 2020 Dialog Semiconductor

| Header | Value | Description |
|---|---|---|
| OTP timestamp | PASS\FAIL | Whether the OTP timestamp read succeeded. |
| OTP timestamp value | Timestamp | The actual timestamp of the DUT IC. |
| Scan HCI Adv [CH37-9\All] | PASS\FAIL | Scan test using Advertising through HCI.<br>The first column shows the result of the test.<br>The second column shows the calculated value in decimal. |
| Scan HCI Adv RSSI [CH37-9\All] | The RSSI value measured for this device. | |
| Extended\Deep sleep current test | PASS\FAIL | Current measurement test – sleep tests.<br>The first column shows the result of the test.<br>The second column shows the calculated value, and the high and low limits used for this test. |
| Extended\Deep sleep current | RES=[xxxxA].<br>LL=xxxxA].<br>HL=[xxxxA]. | |
| Sleep clock select | PASS\FAIL | Sleep clock selection |
| XTAL trim test | PASS\FAIL | Automated XTAL Trim value calculation.<br>The first column shows the result of the test.<br>The second column shows the calculated value in decimal. |
| XTAL trim | e.g. “1155” | |
| ADC calibration | PASS\FAIL | ADC calibration test.<br>The first column shows the result of the test.<br>The second column shows the calculated value. |
| ADC calibration value | xx | |
| BLE TX power test ‘X’ [Test Name] | PASS\FAIL | Tx Power tests using external BLE Tester |
| BLE TX power ‘X’ [Test Name] | TRUE;-13.16;-13.16;-13.16;0.08;0;2;PASS | |
| BLE TX offset test ‘X’ [‘Test Name’] | PASS\FAIL | Tx Frequency offset tests using external BLE Tester |
| BLE TX offset ‘X’ [Test Name] | TRUE;-1.100e+003;1.400e+003;-4.000e+003;2902;-3.0e+003;-3.0e+003;0;2;PASS;2503 | |
| BLE TX modulation test ‘X’ [‘Test Name’] | PASS\FAIL | Tx Modulation Index tests using external BLE Tester.<br>The first column shows the result of the test.<br>The second column shows the calculated value. |
| BLE TX modulation ‘X’ [Test Name] | TRUE;260600.00;253300.00;216800.00;248000.00;0.98;0;576;1;1;FAIL;100.00% | |
| BLE RX RSSI test ‘X’ [‘Test Name’] | PASS\FAIL | Rx sensitivity tests using external BLE Tester.<br>The first column shows the result of the test.<br>The second column shows the RSSI value measured for this device.<br>The third column shows the Packet Error Rate measured for this device. |
| BLE RX RSSI [‘Test Name’] | The RSSI value measured for this device. | |
| BLE RX PER [‘Test Name’] | The Packet Error Rate measured for this device. | |
| GU RX RSSI test ‘X’ [‘Test Name’] | PASS\FAIL | Rx sensitivity tests using Golden Unit as Tester.<br>The first column shows the result of the test.<br>The second column shows the RSSI value measured for this device.<br>The third column shows the Packet Error Rate measured for this device. |
| GU RX RSSI [‘Test Name’] | The RSSI value measured for this device. | |
| GU RX PER [‘Test Name’] | The Packet Error Rate measured for this device. | |

 © 2020 Dialog Semiconductor

| Header | Value | Description |
|---|---|---|
| GPIO/LED test 'X' ['Test Name'] | PASS\FAIL | GPIO\LED tests |
| GPIO connection test 'X' [Test Name] | PASS\FAIL | GPIO connection tests |
| Audio test | PASS\FAIL | Audio test.<br>The first column shows the result of the test.<br>The second column shows the power level measured of each device. |
| Audio level | xx.xx | |
| Sensor test 'X' ['Test Name'] | PASS\FAIL | Sensor tests. |
| Custom test 'X' ['Test Name'] | PASS\FAIL | Custom tests |
| 32KHz Test | PASS\FAIL | External 32kHz crystal test |
| Range extender | PASS\FAIL | Range extender test |
| Peripheral test 'X' ['Test Name'] | PASS\FAIL | Current measurement tests for peripherals.<br>The first column shows the result of the test.<br>The second column shows the calculated value, and the high and low limits used for this test. |
| Peripheral test current | RES=[xxxxA].<br>LL=xxxxA].<br>HL=[xxxxA]. | |
| FW download 2 | PASS\FAIL | Memory programming FW download.<br>The first column shows the result of the production test firmware download procedure.<br>The second columns shows the path to the firmware. |
| FW path 2 | C:\path\to\bin | |
| FW version get 2 | PASS\FAIL | Memory programming FW version.<br>The first column shows the result of the test.<br>The second column shows the memory programming FW version read back from each device. |
| FW version 2 | e.g. "v_5.0.4_PLT_v4.3" | |
| QSPI init | PASS\FAIL | Initialize QSPI Flash memory.<br>The first column shows the result of the test.<br>The second columns shows the Jedec ID read back from the device. |
| QSPI jedec | xxxxxx | |
| QSPI erase 'X' ['Test Name'] | PASS\FAIL | Erase the QSPI Flash memory test. |
| QSPI check empty 'X' ['Test Name'] | PASS\FAIL | Verify that the QSPI Flash memory has been erased. |
| QSPI burn 'X' ['Test Name'] | PASS\FAIL | QSPI image write test.<br>The first column shows the result of the test.<br>The second columns shows the path to the firmware burnt. |
| QSPI image | C:\path\to\bin | |
| QSPI BDA burn | PASS\FAIL | Write BD address value to QSPI Header. |
| QSPI BDA rd/cmp | PASS\FAIL | Read BD address written in QSPI Header and compare it with the one the device is currently using.<br>The first column shows the result of the test.<br>The second columns shows the BD address written in the QSPI header. |
| QSPI BDA read | XX:XX:XX:XX:XX:XX | |
| QSPI XTAL trim burn | PASS\FAIL | Write XTAL Trim value to QSPI Header. |
| QSPI write ADC calibration | PASS\FAIL | Write ADC calibration value to QSPI memory. |
| Custom memory burn | PASS\FAIL | Write on any available memory test. |

 © 2020 Dialog Semiconductor

| Header | Value | Description |
|---|---|---|
| Custom memory data | “Data from CSV file” or “xx…” | The first column shows the result of the test.<br>The second columns shows whether the data are given from a CSV file or the contents written. |
| ‘SPI\OTP\EEPROM\QSPI’ Memory read ‘X’ [‘Test Name’] | PASS\FAIL | Read any part of any available memory.<br>The first column shows the result of the test.<br>The second columns shows the contents read back. |
| Memory read data | xx… | |
| SPI init | PASS\FAIL | Initialize SPI Flash memory. |
| SPI erase ‘X’ [‘Test Name’] | PASS\FAIL | Erase the SPI Flash memory test. |
| SPI empty ‘X’ [‘Test Name’] | PASS\FAIL | Verify that the SPI Flash memory has been erased. |
| SPI burn ‘X’ [‘Test Name’] | PASS\FAIL | SPI image write tests.<br>The first column shows the result of the test.<br>The second columns shows the path to the firmware burnt. |
| SPI image | C:\path\to\bin | |
| EEPROM init | PASS\FAIL | Initialize EEPROM memory. |
| EEPROM burn ‘X’ [‘Test Name’] | PASS\FAIL | EEPROM image write tests.<br>The first column shows the result of the test.<br>The second columns shows the path to the firmware burnt. |
| EEPROM image | C:\path\to\bin | |
| OTP burn | PASS\FAIL | OTP image write test.<br>The first column shows the result of the test.<br>The second columns shows the path to the firmware burnt. |
| OTP image | C:\path\to\bin | |
| OTP BDA burn | PASS\FAIL | Write BD address value to OTP Header. |
| OTP BDA rd/cmp | PASS\FAIL | Read BD address written in OTP Header and compare it with the one the device is currently using.<br>The first column shows the result of the test.<br>The second columns shows the BD address written in the OTP header. |
| OTP BDA read | XX:XX:XX:XX:XX:XX | |
| OTP XTAL trim burn | PASS\FAIL | Write XTAL Trim value to OTP Header. |
| OTP header burn | PASS\FAIL | OTP Header area burn. |
| OTP write ADC calibration | PASS\FAIL | Write ADC calibration value to OTP. |
| Scan | PASS\FAIL | Scan test with the DUTs booting and advertising. |

 © 2020 Dialog Semiconductor

### CSV Log File Example

This example uses three DA14583 devices. The following tests and memory operations are enabled:

- XTAL Trim
- GPIO Watchdog Operation
- Scan DUT Advertise Test
- Golden Unit - RF Test 1
- Golden Unit - RF Test 2
- BLE Tester - Tx Power 1
- BLE Tester - Frequency Offset 1
- BLE Tester - Modulation Index 1
- BLE Tester - Rx Sensitivity 1
- GPIO\LED Test 1
- Custom Test 1
- External 32kHz Test
- Current Measurement Test - Peripheral Test 1
- Current Measurement Test - Extended Sleep Test
- SPI Erase 1
- SPI Check Empty 1
- SPI Write 1
- Custom Memory Data - Manual
- OTP BD Address Read
- Memory Read Test 1
- Scan Test

The first device successfully completed all tests, the second failed to be found by the Golden Unit during the scan test and the third failed the external 32kHz test.

The CSV results of the tests are split into four main categories explained in detailed in the following chapters.

 © 2020 Dialog Semiconductor

### CSV Log File Entries 1/4

The first part of the CSV file contains general device and test information, as shown in Figure 174 and explained in Table 127.

| Start time | End time | DUT | BD address | Overall status | COM port |
|---|---|---|---|---|---|
| 15:05:55 | 15:06:54 | 1 | 80:80:80:00:00:0A | PASS | 44 |
| 15:05:55 | 15:06:54 | 2 | 80:80:80:00:00:0B | FAIL | 45 |
| 15:05:55 | 15:06:54 | 3 | 80:80:80:00:00:0C | FAIL | 46 |

**Figure 174: CSV File Entries (1/4)**

**Table 127: CSV File Entries (1/4).**

| Header | Value | Description |
|---|---|---|
| Start time | 15:05:55 | The time the tests started. |
| End time | 15:06:54 | The time the tests finished. |
| DUT | 1<br>2<br>3 | The PLT position each device is connected. |
| BD address | 80:80:80:00:00:0A<br>80:80:80:00:00:0B<br>80:80:80:00:00:0C | The BD address assigned to each device. |
| Overall status | PASS<br>FAIL<br>FAIL | The overall final result for each device. |
| COM port | 44<br>45<br>46 | The Windows assigned COM port to each device. |


© 2020 Dialog Semiconductor

### CSV Log File Entries 2/4

The second part of the CSV Log file contains the production tests entries, as shown in Figure 175 and explained in Table 128.

| FW download 1 | FW path 1 | FW version get 1 | FW version 1 | GPIO Watchdog [WD-P1_0] | XTAL trim test | XTAL trim | Scan HCI Adv [CH37] | Scan HCI Adv RSSI [CH37] |
|---|---|---|---|---|---|---|---|---|
| PASS | C:\DA1458x_DA1468x_PLT_v_4.x\executables\binaries\prod_test_580.bin | PASS | v_5.0.4_PLT_v4.3 | PASS | PASS | 1266 | PASS | -11.39 |
| PASS | C:\DA1458x_DA1468x_PLT_v_4.x\executables\binaries\prod_test_580.bin | PASS | v_5.0.4_PLT_v4.3 | PASS | PASS | 1237 | PASS | -14.71 |
| PASS | C:\DA1458x_DA1468x_PLT_v_4.x\executables\binaries\prod_test_580.bin | PASS | v_5.0.4_PLT_v4.3 | PASS | PASS | 1155 | PASS | -16.61 |

| BLE TX power test 1 [Tx Pwr 1] | BLE TX power 1 [Tx Pwr 1] | BLE TX offset test 1 [Freq Offs 1] | BLE TX offset 1 [Freq Offs 1] |
|---|---|---|---|
| PASS | TRUE;-17.91;-17.91;-17.91;0.10;0;2;PASS | PASS | TRUE;-2.500e+003;2.800e+003;-7.800e+003;-7646;-9.0e+003;-9.0e+003;0;2;PASS;-7646 |
| PASS | TRUE;-9.63;-9.63;-9.63;0.09;0;2;PASS | PASS | TRUE;-1.500e+003;3.200e+003;-6.800e+003;5945;-6.0e+003;-6.0e+003;0;2;PASS;-1054 |
| PASS | TRUE;-12.27;-12.27;-12.27;0.09;0;2;PASS | PASS | TRUE;4.000e+002;5.900e+003;-3.700e+003;-6009;-8.0e+003;-8.0e+003;0;2;PASS;1756 |

| BLE TX modulation test 1 [Mod Idx 1] | BLE TX modulation 1 [Mod Idx 1] | BLE RX RSSI test 1 [Rx sens 1] | BLE RX RSSI 1 [Rx sens 1] | BLE RX PER 1 [Rx sens 1] |
|---|---|---|---|---|
| PASS | TRUE;282100.00;249100.00;200700.00;248700.00;1.00;0;576;1;1;FAIL;100.00% | PASS | -18.03 | 4 |
| PASS | TRUE;276800.00;248800.00;190600.00;238200.00;0.96;0;576;1;1;FAIL;100.00% | PASS | -11.87 | 5.6 |
| PASS | TRUE;261800.00;246400.00;199400.00;238700.00;0.97;0;576;1;1;FAIL;100.00% | PASS | -13.29 | 0.4 |

| GU RX RSSI test 1 [GU_RSSI_1] | GU RX RSSI 1 [GU_RSSI_1] | GU RX PER 1 [GU_RSSI_1] | GU RX RSSI test 2 [GU_RSSI_2] | GU RX RSSI 2 [GU_RSSI_2] | GU RX PER 2 [GU_RSSI_2] | GPIO/LED test 1 [GPIO_P1_0] |
|---|---|---|---|---|---|---|
| PASS | -11.87 | 3.2 | PASS | -11.87 | 7.6 | PASS |
| PASS | -15.18 | 0 | PASS | -15.66 | 1.2 | PASS |
| PASS | -15.66 | 0.4 | PASS | -15.66 | 0.8 | PASS |

| Custom test 1 | 32KHz Test | Peripheral test 1 | Peripheral test current 1 | Extended sleep current test | Extended sleep current |
|---|---|---|---|---|---|
| PASS | PASS | PASS | RES=[0.007038353A]. LL=[0.0000020000A]. HL=[0.2000000000A]. | PASS | RES=[0.001338430A]. LL=[0.0000340000A]. HL=[0.0400000000A]. |
| PASS | PASS | PASS | RES=[0.007038353A]. LL=[0.0000020000A]. HL=[0.2000000000A]. | PASS | RES=[0.001338430A]. LL=[0.0000340000A]. HL=[0.0400000000A]. |
| PASS | FAIL | | | | |

**Figure 175: CSV File Entries (2/4)**

**Table 128: CSV File Entries (2/4).**

| Header | Value | Description |
|---|---|---|
| FW download 1 | PASS | Production test firmware downloaded successfully to all devices. Selected firmware is `C:\DA1458x_DA1468x_PLT_v_4.x\executables\binaries\prod_test_580.bin`. |
| FW path 1 | C:\SmartBond_PLT_v_4.x\executables\binaries\prod_test_580.bin | |


© 2020 Dialog Semiconductor

| Header | Value | Description |
|---|---|---|
| FW version get 1 | PASS | All devices responded to firmware version request. All of them have version `v_5.0.4_PLT_v4.3`. |
| FW version 1 | v_5.0.4_PLT_v4.3 | |
| GPIO watchdog [WD-P1_0] | PASS | GPIO toggling for watchdog has started. In addition, header has the name assigned to the test. |
| XTAL Trim test | PASS | XTAL Trim test finished successfully. The calculated value (in decimal) is shown for each device. |
| XTAL Trim | 1266<br>1237<br>1155 | |
| Scan HCI Adv [CH37] | PASS | Scan test using advertising through HCI commands test finished successfully. Channel 37 is selected, which is also shown on the header. RSSI values, for each device are also shown. |
| Scan HCI Adv RSSI [CH37] | -11.39<br>-14.71<br>-16.61 | |
| BLE TX power test 1 [Tx Pwr 1] | PASS | Transmission power test using external BLE Tester finished successfully.<br>In addition, header has the name assigned to the test and the BLE tester values are shown exactly as they were retrieved. |
| BLE TX power 1 [Tx Pwr 1] | TRUE;-14.25;-14.25;-14.25;0.10;0;2;PASS<br>TRUE;-19.89;-19.89;-19.89;0.13;0;2;PASS<br>TRUE;-9.76;-9.76;-9.76;0.08;0;2;PASS | |
| BLE TX offset test 1 [Freq Offs 1] | PASS | Transmission offset test using external BLE Tester finished successfully.<br>In addition, header has the name assigned to the test and the BLE tester values are shown exactly as they were retrieved. |
| BLE TX offset 1 [Freq Offs 1] | TRUE;-2.500e+003;2.800e+003;-7.800e+003;-7646;-9.0e+003;-9.0e+003;0;2;PASS;-7646<br>TRUE;-1.500e+003;3.200e+003;-6.800e+003;5945;-6.0e+003;-6.0e+003;0;2;PASS;-1054<br>TRUE;4.000e+002;5.900e+003;-3.700e+003;-6009;-8.0e+003;-8.0e+003;0;2;PASS;1756 | |
| BLE TX modulation test 1 [Mod Idx 1] | PASS | Transmission modulation index test using external BLE Tester finished successfully.<br>In addition, header has the name assigned to the test and the BLE tester values are |


© 2020 Dialog Semiconductor

| Header | Value | Description |
|---|---|---|
| BLE TX modulation 1 [Mod Idx 1] | TRUE;282100.00;249100.00;200700.00;248700.00;1.00;0;576;1;1;FAIL;100.00%<br>TRUE;276800.00;248800.00;190600.00;238200.00;0.96;0;576;1;1;FAIL;100.00%<br>TRUE;261800.00;246400.00;199400.00;238700.00;0.97;0;576;1;1;FAIL;100.00% | shown exactly as they were retrieved. |
| BLE RX RSSI test 1 [Rx sens 1] | PASS | Reception test using external BLE Tester finished successfully.<br>In addition, header has the name assigned to the test and the RSSI and packet error rate for each device are shown. |
| BLE RX RSSI 1 [Rx sens 1] | -18.03<br>-11.87<br>-13.29 | |
| BLE RX PER 1 [Rx sens 1] | 4<br>5.6<br>0.4 | |
| GU RX RSSI test 1 [GU_RSSI_1] | PASS | Reception test 1 using the Golden Unit finished successfully.<br>In addition, header has the name assigned to the test and the RSSI and packet error rate for each device are shown. |
| GU RX RSSI 1 [GU_RSSI_1] | -11.87<br>-15.18<br>-15.66 | |
| GU RX PER 1 [GU_RSSI_1] | 3.2<br>0<br>0.4 | |
| GU RX RSSI test 2 [GU_RSSI_2] | PASS | Reception test 2 using the Golden Unit finished successfully.<br>In addition, header has the name assigned to the test and the RSSI and packet error rate for each device are shown. |
| GU RX RSSI 2 [GU_RSSI_2] | -11.87<br>-15.66<br>-15.66 | |
| GU RX PER 2 [GU_RSSI_2] | 7.6<br>1.2<br>0.8 | |


© 2020 Dialog Semiconductor

| Header | Value | Description |
|---|---|---|
| GPIO/LED test 1 [GPIO_P1_0] | PASS | GPIO\LED toggling test finished successfully. |
| Custom test 1 | PASS | Custom test finished successfully. |
| External 32kHz test | Devices 1-2: PASS<br>Device 3: FAIL | External 32kHz test finished successfully for devices 1 and 2. In this test, device #3 **failed**. Since device 3 failed on this test, it will not continue with the remaining tests, meaning that the entries for device 3 will be blank. |
| Peripheral test 1 | PASS | Current measurement for peripherals test finished successfully. Only one test was active. The limits used and the value measured are shown. These values are for all active devices, in this case for two devices, devices 1 and 2. |
| Peripheral test current 1 | RES=[0.007038353A]. LL=[0.0000020000A]. HL=[0.2000000000A].<br>RES=[0.007038353A]. LL=[0.0000020000A]. HL=[0.2000000000A]. | |
| Extended sleep current test | PASS | Current measurement during extended sleep test finished successfully. The limits used and the value measured are shown. These values are for all active devices, in this case for two devices, devices 1 and 2. |
| Extended sleep current | RES=[0.001338430A]. LL=[0.0000340000A]. HL=[0.0400000000A].<br>RES=[0.001338430A]. LL=[0.0000340000A]. HL=[0.0400000000A]. | |


© 2020 Dialog Semiconductor

### CSV Log File Entries 3/4

The third part of the CSV Log file contains the memory programming entries, as shown in Figure 176 and explained in Table 129.

| FW download 2 | FW path 2 | FW version get 2 | FW version 2 | GPIO Watchdog mem [WD-P1_0] | SPI init | SPI erase 1 [SPI ER 1] | SPI empty 1 [SPI ER 1] |
|---|---|---|---|---|---|---|---|
| PASS | C:\DA1458x_DA1468x_PLT_v_4.x\executables\binaries\flash_programmer_580.bin | PASS | v_5.0.4_PLT_v4.3 | PASS | PASS | PASS | PASS |
| PASS | C:\DA1458x_DA1468x_PLT_v_4.x\executables\binaries\flash_programmer_580.bin | PASS | v_5.0.4_PLT_v4.3 | PASS | PASS | PASS | PASS |

| SPI burn 1 [SPI WR 1] | SPI image 1 [SPI WR 1] | Custom memory burn | Custom memory data | OTP BDA rd/cmp | OTP BDA read | SPI Memory read 1 [SPI @8000] | SPI Memory read data 1 [SPI @8000] |
|---|---|---|---|---|---|---|---|
| PASS | binaries\prox_reporter_580.bin | PASS | 1122334455 | PASS | 80:80:80:00:00:0A | PASS | 1122334455 |
| PASS | binaries\prox_reporter_580.bin | PASS | 1122334455 | FAIL | 00:00:00:00:00:00 | | |

**Figure 176: CSV File Entries (3/4)**

**Table 129: CSV File Entries (3/4).**

| Header | Value | Description |
|---|---|---|
| FW download 2 | PASS | Production test firmware downloaded successfully to all devices. Selected firmware is `C:\DA1458x_DA1468x_PLT_v_4.x\executables\binaries\prod_test_580.bin`. |
| FW path 2 | C:\DA1458x_DA1468x_PLT_v_4.x\executables\binaries\prod_test_580.bin | |
| FW version get 2 | PASS | All devices responded to firmware version request. All of them have version `v_5.0.4_PLT_v4.3`. |
| FW version 2 | v_5.0.4_PLT_v4.3 | |
| GPIO watchdog [WD-P1_0] | PASS | GPIO toggling for watchdog has started. In addition, header has the name assigned to the test. |
| SPI init | PASS | SPI flash initialization procedure completed successfully. |
| SPI erase 1 [SPI ER 1] | PASS | SPI flash erase procedure completed successfully. Only one test is active. In addition, header has the name assigned to the test. |
| SPI empty 1 [SPI ER 1] | PASS | SPI flash check for empty contents procedure completed successfully. Only one test is active. In addition, header has the name assigned to the test. |
| SPI burn 1 [SPI WR 1] | PASS | Production test firmware downloaded successfully to all devices. Selected firmware is |


© 2020 Dialog Semiconductor

## SmartBond Production Line Tool

| Header | Value | Description |
|---|---|---|
| SPI image 1 [SPI WR 1] | binaries\prox_reporter_580.bin<br>binaries\prox_reporter_580.bin | `C:\DA1458x_DA1468x_PLT_v_4.x\executables\binaries\prod_test_580.bin`. |
| Custom memory burn | PASS | Custom memory data write procedure completed successfully. Data written were `1122334455`. |
| Custom memory data | 1122334455<br>1122334455 | |
| OTP BDA rd/cmp | Device 1: PASS<br>Device 2: FAIL | OTP read and compared test completed successfully for both devices, but the contents of device 2 did not match the assigned BD address, resulting to **failure** for device 2. Assigned BD address for device 2 was `80:80:80:00:00:0B` and the contents of the OTP header were `00:00:00:00:00:00`. Device1 BD address assigned and OTP header contents were both `80:80:80:00:00:0A`. Since device 2 failed on this test, it will not continue with the remaining tests, meaning that the entries for device 2 will be blank. |
| OTP BDA read | 80:80:80:00:00:0A<br>00:00:00:00:00:00 | |
| SPI Memory read 1 [SPI @8000] | PASS | Read data from SPI Flash memory completed successfully. Only one test was active. Data read back were `1122334455`. The memory and address are the as those in `Custom memory write`, resulting to same contents. In addition, header has the name assigned to the test. |
| SPI Memory read data 1 [SPI @8000] | 1122334455 | |


© 2020 Dialog Semiconductor

### CSV Log File Entries 4/4

The last part of the CSV Log file contains the scan test entry, as shown in Figure 177 and explained in Table 130.

| |
|---|
| Scan |
| PASS |
| |
| |

**Figure 177: CSV File Entries (4/4)**

**Table 130: CSV File Entries (4/4).**

| Header | Value | Description |
|---|---|---|
| Scan | PASS | Scan test for device 1 has finished successfully. |


© 2020 Dialog Semiconductor

# Appendix CC DUT Status Codes

Table 131 contains all the possible status codes a DUT can have, followed by a brief description. The table categorizes the status based on the various states of the DUT during testing and programming.

**Table 131: DUT Status Codes**

| Status | Description |
|---|---|
| **Generic** | |
| DUT_NOT_ACTIVE | Device is not active. |
| DUT_INTERNAL_SYSTEM_ERROR | Internal system error. |
| DUT_COM_PORT_IDENTIFY_STARTED | COM port identification started. |
| DUT_COM_PORT_IDENTIFY_OK | COM port identified successfully. |
| DUT_COM_PORT_IDENTIFY_FAILED | COM port identification failed. |
| DUT_GU_ERROR | Error occurred due to a Golden Unit failure. Check the Golden unit status for more information. |
| **COM port enumeration** | |
| DUT_PDLL_UART_LOOP_INIT | UART loop test initialized. |
| DUT_PDLL_UART_LOOP_START | UART loop test start. |
| DUT_PDLL_UART_LOOP_OK | UART loop test ended successfully. |
| DUT_PDLL_UART_LOOP_FAILED | UART loop test failed. |
| **Temperature measurement** | |
| DUT_TEMPERATURE_MEASUREMENT_INIT | Temperature measurement initialized. |
| DUT_TEMPERATURE_MEASUREMENT_OK | Temperature measurement finished successfully. |
| DUT_TEMPERATURE_MEASUREMENT_ERROR | Temperature measurement error. |
| **Production test - Generic errors** | |
| DUT_PDLL_NO_ERROR | PDLL returned success. |
| DUT_PDLL_PARAMS_ERROR | PDLL Device parameters contain errors. |
| DUT_PDLL_RX_TIMEOUT | Device did not reply on a PDLL message request. |
| DUT_PDLL_TX_TIMEOUT | Sending a message to the device failed due to Tx timeout. |
| DUT_PDLL_UNEXPECTED_EVENT | Received an unexpected message from the device. |
| DUT_PDLL_CANNOT_ALLOCATE_MEMORY | PDLL cannot allocate memory. |
| DUT_PDLL_INTERNAL_ERROR | PDLL internal system error. |
| DUT_PDLL_THREAD_CREATION_ERROR | PDLL thread creation error. |
| DUT_PDLL_INVALID_DBG_PARAMS | PDLL debug library (dbg_dll.dll) access error. |
| DUT_PDLL_DBG_DLL_ERROR | PDLL invalid debug library (dbg_dll.dll) parameters. |
| DUT_PDLL_HCI_STANDARD_ERROR | HCI error. |
| **Production test - COM port** | |
| DUT_PDLL_COM_PORT_INIT | PDLL Device COM port open initialized. |
| DUT_PDLL_COM_PORT_START | PDLL Device COM port open started. |
| DUT_PDLL_COM_PORT_OK | PDLL Device COM port opened successfully. |
| DUT_PDLL_COM_PORT_FAILED | PDLL Device COM port failed. |


© 2017 Dialog Semiconductor

| Status | Description |
|---|---|
| **Production test - UART resync** | |
| DUT_PDLL_UART_RESYNC_INIT | UART resync process initialized. |
| DUT_PDLL_UART_RESYNC_START | UART resync process started. |
| DUT_PDLL_UART_RESYNC_OK | UART resync process completed successfully. |
| DUT_PDLL_UART_RESYNC_FAILED | UART resync process failed. |
| **Production test - Firmware version** | |
| DUT_PDLL_FW_VERSION_GET_START | PDLL Device Firmware version acquisition started. |
| DUT_PDLL_FW_VERSION_GET_OK | PDLL Device Firmware version acquisition completed successfully. |
| DUT_PDLL_FW_VERSION_GET_FAILED | PDLL Device Firmware version acquisition failed. |
| **Production test – GPIO watchdog** | |
| DUT_PDLL_GPIO_WD_INIT | GPIO watchdog operation initialized. |
| DUT_PDLL_GPIO_WD_START | GPIO watchdog operation started. |
| DUT_PDLL_GPIO_WD_OK | GPIO watchdog operation ended successfully. |
| DUT_PDLL_GPIO_WD_FAILED | GPIO watchdog operation failed. |
| **Production test – OTP timestamp read** | |
| DUT_PDLL_OTP_TIMESTAMP_RD_INIT | Initialize timestamp read from the OTP. |
| DUT_PDLL_OTP_TIMESTAMP_RD_START | OTP timestamp read operation started. |
| DUT_PDLL_OTP_TIMESTAMP_RD_OK | OTP timestamp read operation ended successfully. |
| DUT_PDLL_OTP_TIMESTAMP_RD_FAILED | OTP timestamp read operation failed. |
| **Production test – VBAT level read** | |
| DUT_PDLL_ADC_VBAT_INIT | Initialize VBAT level read using internal ADC. |
| DUT_PDLL_ADC_VBAT_START | VBAT level read operation started. |
| DUT_PDLL_ADC_VBAT_OK | VBAT level read operation ended successfully. |
| DUT_PDLL_ADC_VBAT_FAILED | VBAT level read operation failed. |
| **Production test – DC-DC level test** | |
| DUT_PDLL_DC_LEVEL_INIT | Initialize DC-DC level test, by reading voltage using internal ADC. |
| DUT_PDLL_DC_LEVEL_START | DC-DC level test operation started. |
| DUT_PDLL_DC_LEVEL_OK | DC-DC level test operation ended successfully. |
| DUT_PDLL_DC_LEVEL_FAILED | DC-DC level test operation failed. Cannot read voltage |
| DUT_PDLL_DC_LEVEL_LIMITS_PASSED | DC-DC level is outside user defined limits. Test failed. |
| DUT_PDLL_DC_LEVEL_LIMITS_FAILED | DC-DC level is inside user defined limits. Test passed. |
| **Production test – TX power level set** | |
| DUT_PDLL_SET_TX_PWR_INIT | Initialize TX power set operation. |
| DUT_PDLL_SET_TX_PWR_START | TX power set operation started. |
| DUT_PDLL_SET_TX_PWR_OK | TX power set operation ended successfully. |
| DUT_PDLL_SET_TX_PWR_FAILED | TX power set operation failed. |
| **Production test – Reset mode selection (DA14531)** | |
| DUT_PDLL_RESET_MODE_INIT | Reset mode set operation initialized. |


© 2017 Dialog Semiconductor

| Status | Description |
|---|---|
| DUT_PDLL_RESET_MODE_START | Reset mode set operation started. |
| DUT_PDLL_RESET_MODE_OK | Reset mode set operation ended successfully. |
| DUT_PDLL_RESET_MODE_FAILED | Reset mode set operation failed. |
| **Production test - Current measurement test** | |
| DUT_SLEEP_CURRENT_MEASURE_INIT | Sleep current measurement test initialized. |
| DUT_SLEEP_CURRENT_MEASURE_START | Sleep current measurement test start. |
| DUT_SLEEP_DEVICE_SLEPT_OK | Sleep current measurement device mode set successfully. |
| DUT_SLEEP_CURRENT_MEASURE_ERROR | Sleep current measurement test error. |
| DUT_SLEEP_CURRENT_MEASURE_PASSED | Sleep current measurement test passed. |
| DUT_SLEEP_CURRENT_MEASURE_FAILED | Sleep current measurement test failed. |
| DUT_PDLL_PERIPH_AMMETER_TEST_INIT | Peripheral current measurement test initialized. |
| DUT_PDLL_PERIPH_AMMETER_TEST_START | Peripheral current measurement test start. |
| DUT_PDLL_PERIPH_AMMETER_TEST_ERROR | Peripheral current measurement test error. |
| DUT_PDLL_PERIPH_AMMETER_TEST_PASSED | Peripheral current measurement test passed. |
| DUT_PDLL_PERIPH_AMMETER_TEST_FAILED | Peripheral current measurement test failed. |
| **Production test – Sleep Clock Source** | |
| DUT_PDLL_SLEEP_CLK_SRC_INIT | Sleep Clock source test operation initialized. |
| DUT_PDLL_SLEEP_CLK_SRC_START | Sleep Clock source test operation started. |
| DUT_PDLL_SLEEP_CLK_SRC_OK | Sleep Clock source test operation ended successfully. |
| DUT_PDLL_SLEEP_CLK_SRC_FAILED | Sleep Clock source test operation failed. |
| **Production test – External 32 kHz** | |
| DUT_PDLL_EXT32KHz_TEST_INIT | External 32kHz test operation initialized. |
| DUT_PDLL_EXT32KHz_TEST_START | External 32kHz test operation started. |
| DUT_PDLL_EXT32KHz_TEST_OK | External 32kHz test operation ended successfully. |
| DUT_PDLL_EXT32KHz_TEST_FAILED | External 32kHz test operation failed. |
| **Production test - XTAL trim** | |
| DUT_PDLL_XTAL_TRIM_INIT | XTAL trim operation initialized. |
| DUT_PDLL_XTAL_TRIM_START | XTAL trim operation started. |
| DUT_PDLL_XTAL_TRIM_OK | XTAL trim operation ended successfully. |
| DUT_PDLL_XTAL_TRIM_OUT_OF_RANGE | XTAL trim failed. Input frequency is out of range. |
| DUT_PDLL_XTAL_TRIM_FREQ_CAL_NOT_CONNECTED | XTAL trim could not be performed. Could not detect external input frequency. |
| DUT_PDLL_XTAL_TRIM_OTP_WRITE_FAILED | XTAL trim failed. Could not write the calculated value to the OTP header. |
| DUT_PDLL_XTAL_TRIM_FAILED | XTAL trim failed. |
| **Read value written in OTP** | |
| DUT_PDLL_OTP_XTAL_TRIM_READ_INIT | OTP XTAL trim read operation initialized. |
| DUT_PDLL_OTP_XTAL_TRIM_READ_START | OTP XTAL trim read operation started. |
| DUT_PDLL_OTP_XTAL_TRIM_READ_OK | OTP XTAL trim read operation ended successfully. |
| DUT_PDLL_OTP_XTAL_TRIM_READ_FAILED | OTP XTAL trim read operation failed. |


© 2017 Dialog Semiconductor

| Status | Description |
|---|---|
| **Read register value** | |
| DUT_PDLL_XTAL_TRIM_READ_INIT | XTAL trim value read initialized. |
| DUT_PDLL_XTAL_TRIM_READ_START | XTAL trim value read started. |
| DUT_PDLL_XTAL_TRIM_READ_OK | XTAL trim value read success. |
| DUT_PDLL_XTAL_TRIM_READ_FAILED | XTAL trim value read failed. |
| **Production test - Golden Unit RSSI** | |
| DUT_PDLL_GU_RF_RX_TEST_PASSED | Golden Unit RF RX packet test passed. |
| DUT_PDLL_GU_RF_RX_TEST_FAILED | Golden Unit RF RX packet test failed. |
| **DUT start packet RX** | |
| DUT_PDLL_PKT_RX_STATS_START_INIT | RF RX packet test with statistics start initialized. |
| DUT_PDLL_PKT_RX_STATS_START | RF RX packet test with statistics start. |
| DUT_PDLL_PKT_RX_STATS_STARTED_OK | RF RX packet test with statistics started successfully. |
| DUT_PDLL_PKT_RX_STATS_START_FAILED | RF RX packet test with statistics started failed. |
| **DUT stop packet RX** | |
| DUT_PDLL_PKT_RX_STATS_STOP_INIT | RF RX packet test with statistics stop initialized. |
| DUT_PDLL_PKT_RX_STATS_STOP_START | RF RX packet test with statistics stop. |
| DUT_PDLL_PKT_RX_STATS_STOPPED_OK | RF RX packet test with statistics stopped successfully. |
| DUT_PDLL_PKT_RX_STATS_STOP_FAILED | RF RX packet test with statistics stop failed. |
| **TX power measurement** | |
| DUT_BLE_TESTER_TX_PWR_PASSED | BLE tester TX power test passed. |
| DUT_BLE_TESTER_TX_PWR_FAILED | BLE tester TX power test failed. |
| **TX carrier offset measure** | |
| DUT_BLE_TESTER_TX_OFFS_PASSED | BLE tester TX frequency offset test passed. |
| DUT_BLE_TESTER_TX_OFFS_FAILED | BLE tester TX frequency offset test failed. |
| **TX modulation index measure** | |
| DUT_BLE_TESTER_TX_MOD_IDX_PASSED | BLE tester TX modulation index test passed. |
| DUT_BLE_TESTER_TX_MOD_IDX_FAILED | BLE tester TX modulation index test failed. |
| **RX sensitivity test** | |
| DUT_BLE_TESTER_RX_TEST_PASSED | BLE tester RX sensitivity test passed. |
| DUT_BLE_TESTER_RX_TEST_FAILED | BLE tester RX sensitivity test failed. |
| **DUT packet transaction** | |
| DUT_PDLL_PKT_TX_START_INIT | RF packet TX initialized. |
| DUT_PDLL_PKT_TX_START | RF packet TX start. |
| DUT_PDLL_PKT_TX_STARTED_OK | RF packet TX started successfully. |
| DUT_PDLL_PKT_TX_STARTED_FAILED | RF packet TX failed to start. |
| DUT_PDLL_PKT_TX_ENDED_START | RF packet TX ended successfully. |
| DUT_PDLL_PKT_TX_ENDED_OK | RF packet TX end initiated. |
| DUT_PDLL_PKT_TX_ENDED_FAILED | RF packet TX failed to end. |


© 2017 Dialog Semiconductor

| Status | Description |
|---|---|
| **Production test - Scan with HCI BLE advertisements test** | |
| DUT_PDLL_BLE_HCI_ADV_START_INIT | BLE HCI advertise start initialized. |
| DUT_PDLL_BLE_HCI_ADV_START | BLE HCI advertise start started. |
| DUT_PDLL_BLE_HCI_ADV_START_OK | BLE HCI advertise start success. |
| DUT_PDLL_BLE_HCI_ADV_START_FAILED | BLE HCI advertise start failed. |
| DUT_PDLL_BLE_HCI_ADV_STOP_INIT | BLE HCI advertise stop initialized. |
| DUT_PDLL_BLE_HCI_ADV_STOP_START | BLE HCI advertise stop started. |
| DUT_PDLL_BLE_HCI_ADV_STOPPED_OK | BLE HCI advertise stop success. |
| DUT_PDLL_BLE_HCI_ADV_STOP_FAILED | BLE HCI advertise stop failed. |
| DUT_PDLL_BLE_HCI_ADV_SCAN_START | BLE HCI advertise scan started. |
| DUT_PDLL_BLE_HCI_ADV_NOT_YET_FOUND | BLE HCI advertise not yet found. |
| DUT_PDLL_BLE_HCI_ADV_FOUND | BLE HCI advertise found. |
| DUT_PDLL_BLE_HCI_ADV_RSSI_FAILED | BLE HCI advertise RSSI failed. |
| DUT_PDLL_BLE_HCI_ADV_FAILED | BLE HCI advertise failed. |
| **Production test – Range extender test** | |
| DUT_PDLL_RANGE_EXT_EN_INIT | Range extender test enable initialization. |
| DUT_PDLL_RANGE_EXT_EN_START | Range extender test enable start. |
| DUT_PDLL_RANGE_EXT_EN_ERROR | Range extender test enable error. |
| DUT_PDLL_RANGE_EXT_EN_OK | Range extender test enable ended successfully. |
| DUT_PDLL_RANGE_EXT_EN_FAILED | Range extender test enable failed. |
| **Production test - GPIO/LED test** | |
| DUT_PDLL_GPIO_TOGGLE_INIT | GPIO-LED test operation initialized. |
| DUT_PDLL_GPIO_TOGGLE_START | GPIO-LED test operation start. |
| DUT_PDLL_GPIO_TOGGLE_FINISHED_OK | GPIO-LED test operation completed successfully. |
| DUT_PDLL_GPIO_TOGGLE_ERROR | GPIO-LED test operation error. |
| DUT_PDLL_GPIO_TOGGLE_FAILED | GPIO-LED test operation failed. |
| DUT_PDLL_GPIO_TOGGLE_PASSED | GPIO-LED test operation passed. |
| **Production test – GPIO connection test** | |
| DUT_PDLL_GPIO_CONNECTION_INIT | GPIO Connection test operation initialized. |
| DUT_PDLL_GPIO_SET_START | GPIO Connection test set operation start. |
| DUT_PDLL_GPIO_SET_ERROR | GPIO Connection test set operation error. |
| DUT_PDLL_GPIO_SET_FINISHED_OK | GPIO Connection test set operation success. |
| DUT_PDLL_GPIO_GET_START | GPIO Connection test get operation start. |
| DUT_PDLL_GPIO_GET_ERROR | GPIO Connection test get operation passed. |
| DUT_PDLL_GPIO_GET_FINISHED_OK | GPIO Connection test get operation success. |
| DUT_PDLL_GPIO_CONNECTION_ERROR | GPIO Connection test operation error. |
| DUT_PDLL_GPIO_CONNECTION_FAILED | GPIO Connection test operation failed. |
| DUT_PDLL_GPIO_CONNECTION_PASSED | GPIO Connection test operation completed successfully. |


© 2017 Dialog Semiconductor

| Status | Description |
|---|---|
| **Production test - Audio test** | |
| DUT_PDLL_AUDIO_TEST_START_INIT | Audio test start action initialized. |
| DUT_PDLL_AUDIO_TEST_START | Audio test action start. |
| DUT_PDLL_AUDIO_TEST_ALREADY_ACTIVE | Audio test is already active. |
| DUT_PDLL_AUDIO_TEST_STARTED_OK | Audio test action started successfully. |
| DUT_PDLL_AUDIO_TEST_START_FAILED | Audio test start action failed. |
| DUT_PDLL_AUDIO_TEST_STOP_INIT | Audio test stop action initialized. |
| DUT_PDLL_AUDIO_TEST_STOP | Audio test stop action started. |
| DUT_PDLL_AUDIO_TEST_STOPPED_OK | Audio test stop action completed successfully. |
| DUT_PDLL_AUDIO_TEST_STOP_FAILED | Audio test stop action failed. |
| DUT_PDLL_AUDIO_TEST_PASSED | Audio test passed. |
| DUT_PDLL_AUDIO_TEST_FAILED | Audio test failed. |
| DUT_PDLL_AUDIO_TEST_INVALID_COMMAND | Audio test invalid command. |
| **Production test - Sensor test** | |
| DUT_PDLL_SENSOR_TEST_INIT | Sensor test action initialized. |
| DUT_PDLL_SENSOR_TEST_START | Sensor test action start. |
| DUT_PDLL_SENSOR_TEST_OK | Sensor test action ended successfully. |
| DUT_PDLL_SENSOR_TEST_FAILED | Sensor test action failed. |
| DUT_PDLL_SENSOR_TEST_DATA_MATCH_OK | Sensor test action data matched. |
| DUT_PDLL_SENSOR_TEST_DATA_MATCH_FAILED | Sensor test action data match failure. |
| **Production test - Custom action test** | |
| DUT_PDLL_CUSTOM_ACTION_INIT | Custom test action initialized. |
| DUT_PDLL_CUSTOM_ACTION_START | Custom test action start. |
| DUT_PDLL_CUSTOM_ACTION_OK | Custom test action ended successfully. |
| DUT_PDLL_CUSTOM_ACTION_FAILED | Custom test action failed. |
| DUT_PDLL_CUSTOM_ACTION_DATA_MATCH_OK | Custom test action data matched. |
| DUT_PDLL_CUSTOM_ACTION_DATA_MATCH_FAILED | Custom test action data match failure. |
| **Production test - ADC calibration test** | |
| DUT_PDLL_ADC_CALIB_INIT | ADC calibration process. Initializing process. |
| DUT_PDLL_ADC_CALIB_VBAT_RD_START | ADC calibration process. Start reading VBAT voltage using the external voltage meter instrument. |
| DUT_PDLL_ADC_CALIB_VBAT_RD_OK | ADC calibration process. VBAT voltage read ended successfully. |
| DUT_PDLL_ADC_CALIB_VBAT_RD_FAILED | ADC calibration process. VBAT voltage read failed. |
| DUT_PDLL_ADC_CALIB_DUT_RD_INIT | ADC calibration process. Initialize reading device ADC samples. |
| DUT_PDLL_ADC_CALIB_DUT_RD_START | ADC calibration process. Start reading device ADC samples. |
| DUT_PDLL_ADC_CALIB_DUT_RD_OK | ADC calibration process. Device ADC samples read success. |
| DUT_PDLL_ADC_CALIB_DUT_RD_FAILED | ADC calibration process. Device ADC samples read failed. |
| DUT_PDLL_ADC_CALIB_OK | ADC calibration process ended successfully. |
| DUT_PDLL_ADC_CALIB_FAILED | ADC calibration process failed. |


© 2017 Dialog Semiconductor

| Status | Description |
|---|---|
| **Production test - Scan test** | |
| DUT_PDLL_BLE_SCAN_INIT | Scan operation initialized. |
| DUT_PDLL_BLE_SCAN_START | Scan operation start. |
| DUT_PDLL_BLE_SCAN_NOT_YET_FOUND | Scan operation. DUT has not been found yet. |
| DUT_PDLL_BLE_SCAN_FOUND | Scan operation completed successfully. DUT was found. |
| DUT_PDLL_BLE_SCAN_FAILED | Scan operation failed. DUT was not found. |
| DUT_PDLL_BLE_SCAN_RSSI_FAILED | Scan operation failed. RSSI level is outside user defined limits. |
| **Memory programming - Generic errors** | |
| DUT_UDLL_SUCCESS | UDLL returned success. |
| DUT_UDLL_ACTION_RESPONSE_ERROR | UDLL device responded with error. |
| DUT_UDLL_UART_RX_TIMEOUT_ERROR | UDLL UART RX timeout. Cannot communicate with the DUT or DUT is not present. |
| DUT_UDLL_NO_CRC_MATCH_ERROR | UDLL CRC match error. |
| DUT_UDLL_PROG_PARAMS_ERROR | UDLL programming parameter error. |
| DUT_UDLL_DEVICE_PARAMS_ERROR | UDLL device parameter error. |
| DUT_UDLL_UART_WRITE_ERROR | UDLL UART write returned error. |
| DUT_UDLL_UART_READ_ERROR | UDLL UART read returned error. |
| DUT_UDLL_INTERNAL_ERROR | UDLL internal error. |
| DUT_UDLL_COM_PORT_INIT_ERROR | UDLL COM port initialization error. |
| DUT_UDLL_COM_PORT_ERROR | UDLL COM port error. |
| DUT_UDLL_CANNOT_ALLOCATE_MEMORY | UDLL cannot allocate memory. |
| DUT_UDLL_READ_FILE_SIZE_ERROR | UDLL read file size error. |
| DUT_UDLL_CANNOT_OPEN_FW_FILE | UDLL cannot open firmware file. |
| DUT_UDLL_CANNOT_OPEN_IMAGE_FILE | UDLL cannot open image file. |
| DUT_UDLL_UART_PINS_PATCH_ERROR | UDLL cannot patch the UART pins into the firmware file. |
| DUT_UDLL_INVALID_DBG_PARAMS | UDLL invalid debug library (dbg_dll.dll) parameters. |
| DUT_UDLL_DBG_DLL_ERROR | UDLL debug library (dbg_dll.dll) access error. |
| **Firmware download** | |
| DUT_UDLL_FW_DOWNLOAD_INIT | UDLL firmware download initialized. |
| DUT_UDLL_FW_DOWNLOAD_STARTED | UDLL firmware download started successfully. |
| DUT_UDLL_FW_DOWNLOAD_RETRY | UDLL firmware download retry. |
| DUT_UDLL_FW_DOWNLOAD_OK | UDLL firmware downloaded successfully. |
| DUT_UDLL_FW_DOWNLOAD_FAILED | UDLL firmware download failed. |
| **Memory programming - Firmware version** | |
| DUT_UDLL_FW_VER_GET_INIT | UDLL device firmware version acquisition initialized |
| DUT_UDLL_FW_VER_GET_STARTED | UDLL device firmware version acquisition started. |
| DUT_UDLL_FW_VER_GET_OK | UDLL device firmware version acquisition completed successfully. |
| DUT_UDLL_FW_VER_GET_FAILED | UDLL device Firmware version acquisition failed. |


© 2017 Dialog Semiconductor

| Status | Description |
|---|---|
| **Memory programming – GPIO watchdog** | |
| DUT_UDLL_GPIO_WD_INIT | UDLL GPIO watchdog operation initialized. |
| DUT_UDLL_GPIO_WD_STARTEDs | UDLL GPIO watchdog operation started. |
| DUT_UDLL_GPIO_WD_OK | UDLL GPIO watchdog operation ended successfully. |
| DUT_UDLL_GPIO_WD_FAILED | UDLL GPIO watchdog operation failed. |
| **Memory programming – RAM firmware download** | |
| DUT_UDLL_RAM_FW_DOWNLOAD_INIT | UDLL firmware download to RAM initialized |
| DUT_UDLL_RAM_FW_DOWNLOAD_STARTED | UDLL firmware download to RAM started. |
| DUT_UDLL_RAM_FW_DOWNLOAD_OK | UDLL firmware download to RAM completed successfully. |
| DUT_UDLL_RAM_FW_DOWNLOAD_FAILED | UDLL firmware download to RAM failed. |
| **Memory programming - OTP image write** | |
| DUT_UDLL_OTP_IMG_WR_INIT | OTP image write operation initialized. |
| DUT_UDLL_OTP_IMG_WR_STARTED | OTP image write operation started. |
| DUT_UDLL_OTP_IMG_WR_OK | OTP image write operation ended successfully. |
| DUT_UDLL_OTP_IMG_WR_FAILED | OTP image write operation failed. |
| **Memory programming - BD address write to OTP memory** | |
| DUT_UDLL_OTP_BDA_WR_INIT | OTP BD address write operation initialized. |
| DUT_UDLL_OTP_BDA_WR_STARTED | OTP BD address write operation started. |
| DUT_UDLL_OTP_BDA_WR_OK | OTP BD address write operation ended successfully. |
| DUT_UDLL_OTP_BDA_WR_FAILED | OTP BD address write operation failed. |
| **Memory programming - BD address read/compare to OTP memory** | |
| DUT_UDLL_OTP_BDA_RD_INIT | OTP BD address read operation initialized. |
| DUT_UDLL_OTP_BDA_RD_STARTED | OTP BD address read operation started. |
| DUT_UDLL_OTP_BDA_RD_OK | OTP BD address read operation ended successfully. |
| DUT_UDLL_OTP_BDA_RD_FAILED | OTP BD address read operation failed. |
| DUT_UDLL_OTP_BDA_CMP_OK | OTP BD address comparison success. |
| DUT_UDLL_OTP_BDA_CMP_FAILED | OTP BD address comparison failed. No match. |
| **Memory programming - XTAL trim value write to OTP memory** | |
| DUT_UDLL_OTP_XTAL_TRIM_WR_INIT | OTP XTAL trim value write operation initialized. |
| DUT_UDLL_OTP_XTAL_TRIM_WR_STARTED | OTP XTAL trim value write operation started. |
| DUT_UDLL_OTP_XTAL_TRIM_WR_OK | OTP XTAL trim value write operation ended successfully. |
| DUT_UDLL_OTP_XTAL_TRIM_WR_FAILED | OTP XTAL trim value write operation failed. |
| **Memory programming - ADC calibration value write to OTP memory** | |
| DUT_UDLL_OTP_ADC_CALIB_WR_INIT | OTP ADC calibration value write operation initialized. |
| DUT_UDLL_OTP_ADC_CALIB_WR_STARTED | OTP ADC calibration value write operation started. |
| DUT_UDLL_OTP_ADC_CALIB_WR_OK | OTP ADC calibration value write operation ended successfully. |
| DUT_UDLL_OTP_ADC_CALIB_WR_FAILED | OTP ADC calibration value write operation failed. |
| **Memory programming - OTP header write** | |
| DUT_UDLL_OTP_HDR_WR_INIT | OTP header write operation initialized. |


© 2017 Dialog Semiconductor

| Status | Description |
|---|---|
| DUT_UDLL_OTP_HDR_WR_STARTED | OTP header write operation started. |
| DUT_UDLL_OTP_HDR_WR_OK | OTP header write operation ended successfully. |
| DUT_UDLL_OTP_HDR_WR_FAILED | OTP header write operation failed. |
| **Memory programming - OTP customer field write** | |
| DUT_UDLL_OTP_CUSTOMER_FIELD_WR_INIT | OTP customer field write operation initialized. |
| DUT_UDLL_OTP_CUSTOMER_FIELD_WR_STARTED | OTP customer field write operation started. |
| DUT_UDLL_OTP_CUSTOMER_FIELD_WR_OK | OTP customer field write operation ended successfully. |
| DUT_UDLL_OTP_CUSTOMER_FIELD_WR_FAILED | OTP customer field write operation failed. |
| **Memory programming - OTP configuration script write** | |
| DUT_UDLL_OTP_CFG_SCRIPT_INIT | OTP configuration script field write operation initialized. |
| DUT_UDLL_OTP_CFG_SCRIPT_STARTED | OTP configuration script field write operation started. |
| DUT_UDLL_OTP_CFG_SCRIPT_STATUS | OTP configuration script field write status update of progress. |
| DUT_UDLL_OTP_CFG_SCRIPT_OK | OTP configuration script field write operation ended successfully. |
| DUT_UDLL_OTP_CFG_SCRIPT_FAILED | OTP configuration script field write operation failed. |
| **Memory programming - OTP memory check-empty operation** | |
| DUT_UDLL_OTP_CHECK_EMPTY_INIT | Operation to check whether the OTP field to burn is empty initialized. |
| DUT_UDLL_OTP_CHECK_EMPTY_STARTED | Operation to check whether the OTP field to burn is empty started. |
| DUT_UDLL_OTP_CHECK_EMPTY_OK | The OTP field to burn is empty. |
| DUT_UDLL_OTP_CHECK_SAME_DATA_OK | The OTP field contains the same data as the ones to burn. |
| DUT_UDLL_OTP_CHECK_SKIP_WRITE | The OTP field is already written with data. No new data will be written. It will continue without errors. |
| DUT_UDLL_OTP_CHECK_EMPTY_FAILED | The OTP field is already burned with data. |
| **Memory programming - SPI initialization** | |
| DUT_UDLL_SPI_INIT_INIT | SPI initialization operation initialized. |
| DUT_UDLL_SPI_INIT_STARTED | SPI initialization operation started. |
| DUT_UDLL_SPI_INIT_OK | SPI initialization operation ended successfully. |
| DUT_UDLL_SPI_INIT_FAILED | SPI initialization operation failed. |
| **Memory programming - SPI memory erase** | |
| DUT_UDLL_SPI_ERASE_INIT | SPI erase operation initialized. |
| DUT_UDLL_SPI_ERASE_STARTED | SPI erase operation started. |
| DUT_UDLL_SPI_ERASE_OK | SPI erase operation ended successfully. |
| DUT_UDLL_SPI_ERASE_FAILED | SPI erase operation failed. |
| **Memory programming - SPI memory check empty** | |
| DUT_UDLL_SPI_CHECK_EMPTY_INIT | SPI check if empty operation initialized. |
| DUT_UDLL_SPI_CHECK_EMPTY_STARTED | SPI check if empty operation started, |
| DUT_UDLL_SPI_CHECK_EMPTY_OK | SPI check if empty operation ended successfully, |
| DUT_UDLL_SPI_CHECK_EMPTY_FAILED | SPI check if empty operation failed, |
| **Memory programming - SPI image write** | |
| DUT_UDLL_SPI_IMG_WR_INIT | SPI image write operation initialized. |


© 2017 Dialog Semiconductor

| Status | Description |
|---|---|
| DUT_UDLL_SPI_IMG_WR_STARTED | SPI image write operation started. |
| DUT_UDLL_SPI_IMG_WR_OK | SPI image write operation ended successfully. |
| DUT_UDLL_SPI_IMG_WR_FAILED | SPI image write operation failed. |
| **Memory programming - EEPROM initialization** | |
| DUT_UDLL_EEPROM_INIT_INIT | EEPROM initialization operation initialized. |
| DUT_UDLL_EEPROM_INIT_STARTED | EEPROM initialization operation started. |
| DUT_UDLL_EEPROM_INIT_OK | EEPROM initialization operation ended successfully. |
| DUT_UDLL_EEPROM_INIT_FAILED | EEPROM initialization operation failed. |
| **Memory programming - EEPROM image write** | |
| DUT_UDLL_EEPROM_IMG_WR_INIT | EEPROM image write operation initialized. |
| DUT_UDLL_EEPROM_IMG_WR_STARTED | EEPROM image write operation started. |
| DUT_UDLL_EEPROM_IMG_WR_OK | EEPROM image write operation ended successfully. |
| DUT_UDLL_EEPROM_IMG_WR_FAILED | EEPROM image write operation failed. |
| **Memory programming - QSPI memory initialization** | |
| DUT_UDLL_QSPI_INIT_INIT | QSPI initialization operation initialized. |
| DUT_UDLL_QSPI_INIT_STARTED | QSPI initialization operation started. |
| DUT_UDLL_QSPI_INIT_OK | QSPI initialization operation ended successfully. |
| DUT_UDLL_QSPI_INIT_FAILED | QSPI initialization operation failed. |
| **Memory programming - QSPI memory erase** | |
| DUT_UDLL_QSPI_ERASE_INIT | QSPI erase operation initialized. |
| DUT_UDLL_QSPI_ERASE_STARTED | QSPI erase operation started. |
| DUT_UDLL_QSPI_ERASE_OK | QSPI erase operation ended successfully. |
| DUT_UDLL_QSPI_ERASE_FAILED | QSPI erase operation failed. |
| **Memory programming - QSPI memory check empty** | |
| DUT_UDLL_QSPI_CHECK_EMPTY_INIT | QSPI check if empty operation initialized. |
| DUT_UDLL_QSPI_CHECK_EMPTY_STARTED | QSPI check if empty operation started. |
| DUT_UDLL_QSPI_CHECK_EMPTY_OK | QSPI check if empty operation ended successfully. |
| DUT_UDLL_QSPI_CHECK_EMPTY_FAILED | QSPI check if empty operation failed. |
| **Memory programming - QSPI image write** | |
| DUT_UDLL_QSPI_IMG_WR_INIT | QSPI image write operation initialized. |
| DUT_UDLL_QSPI_IMG_WR_STARTED | QSPI image write operation started. |
| DUT_UDLL_QSPI_IMG_WR_OK | QSPI image write operation ended successfully. |
| DUT_UDLL_QSPI_IMG_WR_FAILED | QSPI image write operation failed. |
| **Memory programming - BD address write to QSPI memory** | |
| DUT_UDLL_QSPI_BDA_WR_INIT | QSPI BD address write operation initialized. |
| DUT_UDLL_QSPI_BDA_WR_STARTED | QSPI BD address write operation started. |
| DUT_UDLL_QSPI_BDA_WR_OK | QSPI BD address write operation ended successfully. |
| DUT_UDLL_QSPI_BDA_WR_FAILED | QSPI BD address write operation failed. |


© 2017 Dialog Semiconductor

| Status | Description |
|---|---|
| **Memory programming - BD address read/compare to QSPI memory** | |
| DUT_UDLL_QSPI_BDA_RD_INIT | QSPI BD address read operation initialized. |
| DUT_UDLL_QSPI_BDA_RD_STARTED | QSPI BD address read operation started. |
| DUT_UDLL_QSPI_BDA_RD_OK | QSPI BD address read operation ended successfully. |
| DUT_UDLL_QSPI_BDA_RD_FAILED | QSPI BD address read operation failed. |
| DUT_UDLL_QSPI_BDA_CMP_OK | QSPI BD address comparison success. |
| DUT_UDLL_QSPI_BDA_CMP_FAILED | QSPI BD address comparison failed. No match. |
| **Memory programming - XTAL trim value write to QSPI memory** | |
| DUT_UDLL_QSPI_XTAL_TRIM_WR_INIT | QSPI XTAL trim value write operation initialized. |
| DUT_UDLL_QSPI_XTAL_TRIM_WR_STARTED | QSPI XTAL trim value write operation started. |
| DUT_UDLL_QSPI_XTAL_TRIM_WR_OK | QSPI XTAL trim value write operation ended successfully. |
| DUT_UDLL_QSPI_XTAL_TRIM_WR_FAILED | QSPI XTAL trim value write operation failed. |
| **Memory programming - ADC calibration value write to QSPI memory** | |
| DUT_UDLL_QSPI_ADC_CALIB_WR_INIT | QSPI ADC calibration value write operation initialized. |
| DUT_UDLL_QSPI_ADC_CALIB_WR_STARTED | QSPI ADC calibration value write operation started. |
| DUT_UDLL_QSPI_ADC_CALIB_WR_OK | QSPI ADC calibration value write operation ended successfully. |
| DUT_UDLL_QSPI_ADC_CALIB_WR_FAILED | QSPI ADC calibration value write operation failed. |
| **Memory programming - Custom data memory write** | |
| DUT_UDLL_MEM_DATA_WR_INIT | Custom memory data burn operation initialized. |
| DUT_UDLL_MEM_DATA_WR_STARTED | Custom memory data burn operation started. |
| DUT_UDLL_MEM_DATA_WR_STATUS | Custom memory data burn update interim status. |
| DUT_UDLL_MEM_DATA_WR_OK | Custom memory data burn operation ended successfully. |
| DUT_UDLL_MEM_DATA_WR_FAILED | Custom memory data burn operation failed. |
| **Memory programming - memory read operation** | |
| DUT_UDLL_MEM_RD_INIT | Memory read operation initialized. |
| DUT_UDLL_MEM_RD_STARTED | Memory read operation started. |
| DUT_UDLL_MEM_RD_OK | Memory read operation ended successfully. |
| DUT_UDLL_MEM_RD_FAILED | Memory read operation failed. |


© 2017 Dialog Semiconductor

# Appendix DD Golden Unit Status Codes

Table 132 contains all the possible status codes the Golden Unit can have, followed by a brief description. The table categorizes the status based on the various states the Golden Unit may be during testing and programming the DUTs.

**Table 132: Golden Unit Status Codes**

| Status | Description |
|---|---|
| **Generic** | |
| GU_NOT_ACTIVE | Golden Unit is not active. |
| GU_INTERNAL_SYSTEM_ERROR | Internal system error. |
| GU_COM_OPEN_OK | COM port opened successfully. |
| GU_COM_OPEN_FAILED | COM port failed to open |
| GU_PDLL_NO_ERROR | PDLL returned success. |
| GU_PDLL_PARAMS_ERROR | Golden Unit PDLL parameters have errors. |
| GU_PDLL_RX_TIMEOUT | Golden Unit did not reply on a PDLL message request. GU COM port may not be correct or it may need manual RESET. |
| GU_PDLL_TX_TIMEOUT | Golden Unit Tx timeout when sending a message to the device. |
| GU_PDLL_UNEXPECTED_EVENT | Received an unexpected message from the Golden Unit. |
| GU_PDLL_CANNOT_ALLOCATE_MEMORY | PDLL cannot allocate memory. |
| GU_PDLL_INTERNAL_ERROR | PDLL internal system error. |
| GU_PDLL_THREAD_CREATION_ERROR | PDLL thread creation error. |
| GU_PDLL_DBG_DLL_ERROR | PDLL debug library (dbg_dll.dll) access error. |
| GU_PDLL_INVALID_DBG_PARAMS | PDLL invalid debug library (dbg_dll.dll) parameters. |
| GU_PDLL_HCI_STANDARD_ERROR | Golden Unit HCI error. |
| **Golden Unit reset operation** | |
| GU_RESET_START | Golden Unit HW reset started. |
| GU_RESET_OK | Golden Unit HW reset OK. |
| GU_RESET_FAILED | Golden Unit HW reset FAILED. |
| **Golden Unit COM port handling** | |
| GU_PDLL_COM_PORT_INIT | Golden Unit COM port open initialized. |
| GU_PDLL_COM_PORT_START | Golden Unit COM port open started. |
| GU_PDLL_COM_PORT_OK | Golden Unit COM port opened OK. |
| GU_PDLL_COM_PORT_FAILED | Golden Unit COM port FAILED. |
| **Golden Unit firmware version** | |
| GU_PDLL_FW_VERSION_GET_START | Golden Unit PDLL firmware version acquisition started. |
| GU_PDLL_FW_VERSION_GET_OK | Golden Unit PDLL firmware version acquisition OK. |
| GU_PDLL_FW_VERSION_GET_FAILED | Golden Unit PDLL firmware version acquisition FAILED. |
| GU_PDLL_FW_VERSION_VALID | The Golden Unit firmware version is valid. |
| GU_PDLL_FW_VERSION_NOT_VALID | The Golden Unit firmware version is not valid. An upgrade |


© 2017 Dialog Semiconductor

| Status | Description |
|---|---|
| | may be needed. |
| **Golden Unit CPLD control** | |
| GU_PDLL_RDTESTER_INIT | PLT HW tester initializing. |
| GU_PDLL_RDTESTER_INIT_START | PLT HW tester initialize started. |
| GU_PDLL_RDTESTER_INIT_OK | PLT HW tester initialized successful. |
| GU_PDLL_RDTESTER_INIT_FAILED | PLT HW tester initialization failed. |
| GU_PDLL_RDTESTER_UART_CONNECT_INIT | PLT HW tester UART connection initialized. |
| GU_PDLL_RDTESTER_UART_CONNECT_START | PLT HW tester UART connection started. |
| GU_PDLL_RDTESTER_UART_CONNECT_OK | PLT HW tester UART connected successfully. |
| GU_PDLL_RDTESTER_UART_CONNECT_FAILED | PLT HW tester UART connection failed. |
| GU_PDLL_RDTESTER_UART_LOOPBACK_INIT | PLT HW tester UART loopback process initialized. |
| GU_PDLL_RDTESTER_UART_LOOPBACK_START | PLT HW tester UART loopback process started. |
| GU_PDLL_RDTESTER_UART_LOOPBACK_OK | PLT HW tester UART loopback process success. |
| GU_PDLL_RDTESTER_UART_LOOPBACK_FAILED | PLT HW tester UART loopback process failed. |
| GU_PDLL_RDTESTER_VBAT_UART_CNTRL_INIT | PLT HW tester VBAT/UART control initialized. |
| GU_PDLL_RDTESTER_VBAT_UART_CNTRL_START | PLT HW tester VBAT/UART control started. |
| GU_PDLL_RDTESTER_VBAT_UART_CNTRL_OK | PLT HW tester VBAT/UART control success. |
| GU_PDLL_RDTESTER_VBAT_UART_CNTRL_FAILED | PLT HW tester VBAT/UART control failed. |
| GU_PDLL_RDTESTER_VBAT_UART_RST_CNTRL_INIT | PLT HW tester VBAT/UART/Reset control initialized. |
| GU_PDLL_RDTESTER_VBAT_UART_RST_CNTRL_START | PLT HW tester VBAT/UART/Reset control started. |
| GU_PDLL_RDTESTER_VBAT_UART_RST_CNTRL_OK | PLT HW tester VBAT/UART/Reset control success. |
| GU_PDLL_RDTESTER_VBAT_UART_RST_CNTRL_FAILED | PLT HW tester VBAT/UART/Reset control failed. |
| GU_PDLL_RDTESTER_VPP_CNTRL_INIT | PLT HW tester VPP control initialized. |
| GU_PDLL_RDTESTER_VPP_CNTRL_START | PLT HW tester VPP control started. |
| GU_PDLL_RDTESTER_VPP_CNTRL_OK | PLT HW tester VPP control success. |
| GU_PDLL_RDTESTER_VPP_CNTRL_FAILED | PLT HW tester VPP control failed. |
| GU_PDLL_RDTESTER_RST_PULSE_INIT | PLT HW tester Reset pulse control initialized. |
| GU_PDLL_RDTESTER_RST_PULSE_START | PLT HW tester Reset pulse control started. |
| GU_PDLL_RDTESTER_RST_PULSE_OK | PLT HW tester Reset pulse control success. |
| GU_PDLL_RDTESTER_RST_PULSE_FAILED | PLT HW tester Reset pulse control failed. |
| GU_PDLL_RDTESTER_UART_PULSE_INIT | PLT HW tester XTAL trim pulse in UART TX pin initialized. |
| GU_PDLL_RDTESTER_UART_PULSE_START | PLT HW tester XTAL trim pulse in UART TX pin started. |
| GU_PDLL_RDTESTER_UART_PULSE_OK | PLT HW tester XTAL trim pulse in UART TX pin success. |
| GU_PDLL_RDTESTER_UART_PULSE_FAILED | PLT HW tester XTAL trim pulse in UART TX pin failed. |
| GU_PDLL_RDTESTER_XTAL_PULSE_INIT | PLT HW tester XTAL trim pulse in GATE pin initialized. |
| GU_PDLL_RDTESTER_XTAL_PULSE_START | PLT HW tester XTAL trim pulse in GATE pin started. |
| GU_PDLL_RDTESTER_XTAL_PULSE_OK | PLT HW tester XTAL trim pulse in GATE pin success. |
| GU_PDLL_RDTESTER_XTAL_PULSE_FAILED | PLT HW tester XTAL trim pulse in GATE pin failed. |
| GU_PDLL_RDTESTER_PULSE_WIDTH_INIT | PLT HW tester pulse width initialized. |


© 2017 Dialog Semiconductor

| Status | Description |
|---|---|
| GU_PDLL_RDTESTER_PULSE_WIDTH_START | PLT HW tester pulse width started. |
| GU_PDLL_RDTESTER_PULSE_WIDTH_OK | PLT HW tester pulse width success. |
| GU_PDLL_RDTESTER_PULSE_WIDTH_FAILED | PLT HW tester pulse width failed. |
| GU_PDLL_RDTESTER_VBAT_CNTRL_INIT | PLT HW tester VBAT control initialized. |
| GU_PDLL_RDTESTER_VBAT_CNTRL_START | PLT HW tester VBAT control started. |
| GU_PDLL_RDTESTER_VBAT_CNTRL_OK | PLT HW tester VBAT control success. |
| GU_PDLL_RDTESTER_VBAT_CNTRL_FAILED | PLT HW tester VBAT control failed. |
| GU_PDLL_RDTESTER_INVALID_COMMAND | PLT HW tester unknown command. |
| **Golden Unit RF packet transmission for DUT RSSI RF test** | |
| GU_PDLL_PKT_TX_START_INIT | Golden Unit RF packet TX initialized. |
| GU_PDLL_PKT_TX_START | Golden Unit RF packet TX started. |
| GU_PDLL_PKT_TX_STARTED_OK | Golden Unit RF packet TX success. |
| GU_PDLL_PKT_TX_STARTED_FAILED | Golden Unit RF packet TX failed. |
| GU_PDLL_PKT_TX_ENDED_OK | Golden Unit RF packet TX ended successfully. |
| GU_PDLL_PKT_TX_STARTED_FAILED | Golden Unit RF packet TX ended failed. |
| **Golden Unit audio tone generation for audio testing** | |
| GU_PDLL_AUDIO_TONE_START_INIT | Golden Unit audio tone start initialized. |
| GU_PDLL_AUDIO_TONE_START | Golden Unit audio tone start. |
| GU_PDLL_AUDIO_TONE_STARTED_OK | Golden Unit audio tone started successfully. |
| GU_PDLL_AUDIO_TONE_START_FAILED | Golden Unit audio tone start failed. |
| GU_PDLL_AUDIO_TONE_STOP_INIT | Golden Unit audio tone stop initialized. |
| GU_PDLL_AUDIO_TONE_STOP | Golden Unit audio tone stop. |
| GU_PDLL_AUDIO_TONE_STOPPED_OK | Golden Unit audio tone stopped successfully. |
| GU_PDLL_AUDIO_TONE_STOP_FAILED | Golden Unit audio tone stop failed. |
| **Golden Unit GPIO toggling for sanity test** | |
| GU_PDLL_GPIO_TOGGLE_INIT | Golden Unit GPIO toggle operation initialized. |
| GU_PDLL_GPIO_TOGGLE_START | Golden Unit GPIO toggle operation start. |
| GU_PDLL_GPIO_TOGGLE_FINISHED_OK | Golden Unit GPIO toggle operation completed successfully. |
| GU_PDLL_GPIO_TOGGLE_FAILED | Golden Unit GPIO toggle operation failed. |
| **Golden Unit BLE advertising scan test** | |
| GU_PDLL_BLE_SCAN_INIT | Golden Unit scan operation initialized. |
| GU_PDLL_BLE_SCAN_START | Golden Unit scan operation started. |
| GU_PDLL_BLE_SCAN_OK | Golden Unit scan operation completed successfully. |
| GU_PDLL_BLE_SCAN_FAILED | Golden Unit scan operation failed. |


© 2017 Dialog Semiconductor

# Revision History

| Revision | Date | Description |
|---|---|---|
| 1.0 | 08-Jul-2015 | Initial version. |
| 1.1 | 18-Jan-2016 | CLI part is added and Rev D. PLT Hardware |
| 2.0 | 04-May-2016 | Adding text and drawings |
| 3.0 | 22-May-2016 | Adding changes for the DA1468x |
| 4.0 | 22-Dec-2016 | Updated for DA1458x_DA1468x_PLT_v4.0 software release |
| 4.1 | 06-Oct-2017 | Updated for DA1458x_DA1468x_PLT_v4.1 software release |
| 4.2 | 10-Oct-2017 | Updated for DA1458x_DA1468x_PLT_v4.2 software release |
| 4.3 | 17-Jul-2018 | Updated for DA1458x_DA1468x_PLT_v4.3 software release |
| 4.4 | 08-May-2020 | Updated for SmartBond_PLT_v4.4 software release |


© 2017 Dialog Semiconductor

# SmartBond Production Line Tool

## Status Definitions

| Status | Definition |
|---|---|
| DRAFT | The content of this document is under review and subject to formal approval, which may result in modifications or additions. |
| APPROVED or unmarked | The content of this document has been approved for publication. |

## Disclaimer

Information in this document is believed to be accurate and reliable. However, Dialog Semiconductor does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information. Dialog Semiconductor furthermore takes no responsibility whatsoever for the content in this document if provided by any information source outside of Dialog Semiconductor.

Dialog Semiconductor reserves the right to change without notice the information published in this document, including without limitation the specification and the design of the related semiconductor products, software and applications.

Applications, software, and semiconductor products described in this document are for illustrative purposes only. Dialog Semiconductor makes no representation or warranty that such applications, software and semiconductor products will be suitable for the specified use without further testing or modification. Unless otherwise agreed in writing, such testing or modification is the sole responsibility of the customer and Dialog Semiconductor excludes all liability in this respect.

Customer notes that nothing in this document may be construed as a license for customer to use the Dialog Semiconductor products, software and applications referred to in this document. Such license must be separately sought by customer with Dialog Semiconductor.

All use of Dialog Semiconductor products, software and applications referred to in this document are subject to Dialog Semiconductor's Standard Terms and Conditions of Sale, available on the company website (www.dialog-semiconductor.com) unless otherwise stated.

Dialog and the Dialog logo are trademarks of Dialog Semiconductor plc or its subsidiaries. All other product or service names are the property of their respective owners.

© 2020 Dialog Semiconductor. All rights reserved.

# Contacting Dialog Semiconductor

United Kingdom (Headquarters)
*Dialog Semiconductor (UK) LTD*
Phone: +44 1793 757700

Germany
*Dialog Semiconductor GmbH*
Phone: +49 7021 805-0

The Netherlands
*Dialog Semiconductor B.V.*
Phone: +31 73 640 8822

Email:
enquiry@diasemi.com

North America
*Dialog Semiconductor Inc.*
Phone: +1 408 845 8500

Japan
*Dialog Semiconductor K. K.*
Phone: +81 3 5769 5100

Taiwan
*Dialog Semiconductor Taiwan*
Phone: +886 281 786 222

Web site:
www.dialog-semiconductor.com

Hong Kong
*Dialog Semiconductor Hong Kong*
Phone: +852 2607 4271

Korea
*Dialog Semiconductor Korea*
Phone: +82 2 3469 8200

China (Shenzhen)
*Dialog Semiconductor China*
Phone: +86 755 2981 3669

China (Shanghai)
*Dialog Semiconductor China*
Phone: +86 21 5424 9058