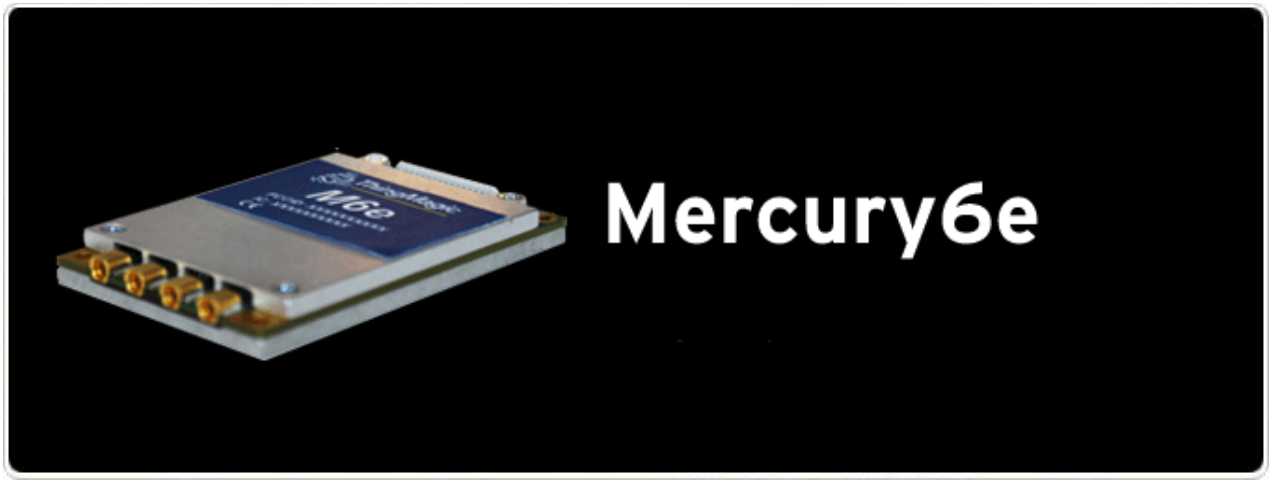
ThingMagic
A DIVISION OF TRIMBLE

M6e-31.5dBm Hardware Guide



For: M6e (Firmware Ver. 1.7 and later)

Government Limited Rights Notice:

All documentation and manuals were developed at private expense and no part of it was developed using Government funds.

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose the technical data contained herein are restricted by paragraph (b)(3) of the Rights in Technical Data — Noncommercial Items clause (DFARS 252.227-7013(b)(3)), as amended from time-to-time. Any reproduction of technical data or portions thereof marked with this legend must also reproduce the markings. Any person, other than the U.S. Government, who has been provided access to such data must promptly notify ThingMagic, Inc.

ThingMagic, Mercury, Reads Any Tag, and the ThingMagic logo are trademarks or registered trademarks of ThingMagic, A Division of Trimble.

Other product names mentioned herein may be trademarks or registered trademarks of Trimble or other companies.

©2011 ThingMagic – a division of Trimble Navigation Limited. ThingMagic and The Engine in RFID are registered trademarks of Trimble Navigation Limited. Other marks may be protected by their respective owners. All Rights Reserved.d

ThingMagic, A Division of Trimble

One Cambridge Center, 11th floor

Cambridge, MA 02142

866-833-4069

02 Revision 2
February, 2011

Revision Table

Date	Version	Description
4/2010	01 RevA	First Draft for Beta release
8/2010	01 RevB	• Updated GPIO content • Added FCC regulation info section
2/2011	02 Rev2	• updated content to meet regulatory requirements

Contents

Communication Regulation Information **9**
Federal Communication Commission Interference Statement 9
Industry Canada 11
Industrie Canada 12

Mercury6e Introduction **13**

Hardware Overview **15**

Hardware Interfaces **16**
Antenna Connections 16
Antenna Requirements 16
Digital/Power Connector 16
Control Signal Specification 17
General Purpose Input/Output (GPIO) 19
Reset Line 20

Power Requirements **21**
RF Power Output 21
Power Settings for Authorized Antennas and Cables 21
Power Supply Ripple 22
Power Consumption 22

Environmental Specifications **24**
Operating Temperature 24
Electro-Static Discharge (ESD) Specification 24

Assembly Information **25**
Cables and Connectors 25
Digital Interface 25
Antennas 25
M6e Mechanical Drawing 26

Authorized Antennas **27**

Authorized Cables **28**

Firmware Overview 29
Boot Loader 30
Application Firmware 31
Programming the M6e 31
Upgrading the M6e 31
Verifying Application Firmware Image 31
Custom On-Reader Applications 32
Communication Protocol 33
Serial Communication Protocol 34
Host-to-Reader Communication 34
Reader-to-Host Communication 35
CCITT CRC-16 Calculation 35
User Programming Interface 36
Functionality of the Mercury6e 37
Regulatory Support 38
Supported Regions 38
Frequency Setting 38
Frequency Units 39
Frequency Hop Table 39
Protocol Support 40
ISO 18000-6C (Gen2) 40
Protocol Configuration Options 40
Protocol Specific Functionality 41
I-PX 41
Protocol Configuration Options 41
ISO 18000-6B 41
Protocol Configuration Options 41
Antenna Ports 42
Using a Multiplexer 42
Port Power and Settling Time 44
Tag Handling 46
Tag Buffer 46
Tag Streaming 46
Tag Read Meta Data 47
Power Management 48
Power Modes 48

Transmit Modes. 48
DRM Compliant Mode 49
Power Save Mode (non-DRM Compliant) 49
Performance Characteristics. 50
Event Response Times. 50
Save and Restore Configuration 51

Appendix A: Error Messages 53

Common Error Messages 53
FAULT_MSG_WRONG_NUMBER_OF_DATA – (100h) 53
FAULT_INVALID_OPCODE – (101h) 54
FAULT_UNIMPLEMENTED_OPCODE – 102h 54
FAULT_MSG_POWER_TOO_HIGH – 103h 54
FAULT_MSG_INVALID_FREQ_RECEIVED (104h) 55
FAULT_MSG_INVALID_PARAMETER_VALUE - (105h) 55
FAULT_MSG_POWER_TOO_LOW - (106h) 55
FAULT_UNIMPLEMENTED_FEATURE - (109h) 56
FAULT_INVALID_BAUD_RATE - (10Ah) 56
Bootloader Faults. 57
FAULT_BL_INVALID_IMAGE_CRC – 200h 57
FAULT_BL_INVALID_APP_END_ADDR – 201h 57
Flash Faults 58
FAULT_FLASH_BAD_ERASE_PASSWORD – 300h 58
FAULT_FLASH_BAD_WRITE_PASSWORD – 301h 58
FAULT_FLASH_UNDEFINED_ERROR – 302h 59
FAULT_FLASH_ILLEGAL_SECTOR – 303h 59
FAULT_FLASH_WRITE_TO_NON_ERASED_AREA – 304h 59
FAULT_FLASH_WRITE_TO_ILLEGAL_SECTOR – 305h 59
FAULT_FLASH_VERIFY_FAILED – 306h 60
Protocol Faults 61
FAULT_NO_TAGS_FOUND – (400h) 62
FAULT_NO_PROTOCOL_DEFINED – 401h 62
FAULT_INVALID_PROTOCOL_SPECIFIED – 402h 62
FAULT_WRITE_PASSED_LOCK_FAILED – 403h 63
FAULT_PROTOCOL_NO_DATA_READ – 404h 63
FAULT_AFE_NOT_ON – 405h 63
FAULT_PROTOCOL_WRITE_FAILED – 406h 64
FAULT_NOT_IMPLEMENTED_FOR_THIS_PROTOCOL – 407h 64
FAULT_PROTOCOL_INVALID_WRITE_DATA – 408h 64

FAULT_PROTOCOL_INVALID_ADDRESS – 409h 64
FAULT_GENERAL_TAG_ERROR – 40Ah 65
FAULT_DATA_TOO_LARGE – 40Bh 65
FAULT_PROTOCOL_INVALID_KILL_PASSWORD – 40Ch 65
FAULT_PROTOCOL_KILL_FAILED - 40Eh 65
FAULT_PROTOCOL_BIT_DECODING_FAILED - 40Fh 66
FAULT_PROTOCOL_INVALID_EPC – 410h 66
FAULT_PROTOCOL_INVALID_NUM_DATA – 411h 66
FAULT_GEN2 PROTOCOL_OTHER_ERROR - 420h 66
FAULT_GEN2_PROTOCOL_MEMORY_OVERRUN_BAD_PC - 423h 67
FAULT_GEN2 PROTOCOL_MEMORY_LOCKED - 424h 67
FAULT_GEN2 PROTOCOL_INSUFFICIENT_POWER - 42Bh 67
FAULT_GEN2 PROTOCOL_NON_SPECIFIC_ERROR - 42Fh 68
FAULT_GEN2 PROTOCOL_UNKNOWN_ERROR - 430h 68

Analog Hardware Abstraction Layer Faults 69
FAULT_AHAL_INVALID_FREQ – 500h 69
FAULT_AHAL_CHANNEL_OCCUPIED – 501h 69
FAULT_AHAL_TRANSMITTER_ON – 502h 69
FAULT_ANTENNA_NOT_CONNECTED – 503h 69
FAULT_TEMPERATURE_EXCEED_LIMITS – 504h 70
FAULT_POOR_RETURN_LOSS – 505h 70
FAULT_AHAL_INVALID_ANTENA_CONFIG – 507h 70

Tag ID Buffer Faults 72
FAULT_TAG_ID_BUFFER_NOT_ENOUGH_TAGS_AVAILABLE – 600h 72
FAULT_TAG_ID_BUFFER_FULL – 601h 72
FAULT_TAG_ID_BUFFER_REPEATED_TAG_ID – 602h 72
FAULT_TAG_ID_BUFFER_NUM_TAG_TOO_LARGE – 603h 73

System Errors 74
FAULT_SYSTEM_UNKNOWN_ERROR – 7F00h 74
FAULT_TM_ASSERT_FAILED – 7F01h 74

Appendix B: Getting Started - Devkit 75

Devkit USB Interfaces 75
USB/RS232 75
Native USB 75

Demo Application 77
Demo Tool Notes 77

Communication Regulation Information

W A R N I N G !

Operation of the M6e module requires professional installation to correctly set the TX power for the RF cable and antenna selected.

EMC FCC 47 CFR, Part 15
Industrie Canada RSS-210

Federal Communication Commission Interference Statement

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

FCC Caution: Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment.

This transmitter module is authorized to be used in other devices only by OEM integrators under the following conditions:

1. The antenna(s) must be installed such that a minimum separation distance of 25cm is maintained between the radiator (antenna) & user's/nearby people's body at all times.

2. The transmitter module must not be co-located with any other antenna or transmitter.

As long as the two conditions above are met, further transmitter testing will not be required. However, the OEM integrator is still responsible for testing their end-product for any additional compliance requirements required with this module installed (for example, digital device emissions, PC peripheral requirements, etc.).

Note

In the event that these conditions can not be met (for certain configurations or co-location with another transmitter), then the FCC authorization is no longer considered valid and the FCC ID can not be used on the final product. In these circumstances, the OEM integrator will be responsible for re-evaluating the end product (including the transmitter) and obtaining a separate FCC authorization.

The OEM integrator has to be aware not to provide information to the end user regarding how to install or remove this RF module in the user manual of the end product.

User Manual Requirement

The user manual for the end product must include the following information in a prominent location;

"To comply with FCC's RF radiation exposure requirements, the antenna(s) used for this transmitter must be installed such that a minimum separation distance of 25cm is maintained between the radiator (antenna) & user's/nearby people's body at all times and must not be co-located or operating in conjunction with any other antenna or transmitter."

AND

"The transmitting portion of this device carries with it the following two warnings:

"This device complies with Part 15...."

AND

"Any changes or modifications to the transmitting module not expressly approved by ThingMagic Inc. could void the user's authority to operate this equipment" "

End Product Labeling

The final end product must be labeled in a visible area with the following:

"Contains Transmitter Module FCC ID: QV5MERCURY6E-A"

or

“*Contains FCC ID: QV5MERCURY6E-A.”*

Industry Canada

Under Industry Canada regulations, this radio transmitter may only operate using an antenna of a type and maximum (or lesser) gain approved for the transmitter by Industry Canada. To reduce potential radio interference to other users, the antenna type and its gain should be so chosen that the equivalent isotropically radiated power (e.i.r.p.) is not more than that necessary for successful communication.

This radio transmitter (identify the device by certification number, or model number if Category II) has been approved by Industry Canada to operate with the antenna types listed below with the maximum permissible gain and required antenna impedance for each antenna type indicated. Antenna types not included in this list, having a gain greater than the maximum gain indicated for that type, are strictly prohibited for use with this device

Operation is subject to the following two conditions: (1) this device may not cause interference, and (2) this device must accept any interference, including interference that may cause undesired operation of the device.

To reduce potential radio interference to other users, the antenna type and its gain should be so chosen that the equivalent isotropically radiated power (e.i.r.p.) is not more than that permitted for successful communication.

This device has been designed to operate with the antennas and cables listed in Authorized Antennas and Authorized Cables tables. Antennas or cables not included in these lists are strictly prohibited for use with this device.

To comply with IC RF exposure limits for general population/uncontrolled exposure, the antenna(s) used for this transmitter must be installed to provide a separation distance of at least 25 cm from all persons and must not be collocated or operating in conjunction with any other antenna or transmitter.

End Product Labeling

The final end product must be labeled in a visible area with the following:

“Contains ThingMagic Inc. Mercury6e (or appropriate model number you’re filing with IC) *transmitting module FCC ID: QV5MERCURY6E-A (IC: 5407A-MERCURY6EA)”*

Industrie Canada

Conformément à la réglementation d'Industrie Canada, le présent émetteur radio peut fonctionner avec une antenne d'un type et d'un gain maximal (ou inférieur) approuvé pour l'émetteur par Industrie Canada. Dans le but de réduire les risques de brouillage radioélectrique à l'intention des autres utilisateurs, il faut choisir le type d'antenne et son gain de sorte que la puissance isotrope rayonnée équivalente (p.i.r.e.) ne dépasse pas l'intensité nécessaire à l'établissement d'une communication satisfaisante.

Le présent émetteur radio (identifier le dispositif par son numéro de certification ou son numéro de modèle s'il fait partie du matériel de catégorie I) a été approuvé par Industrie Canada pour fonctionner avec les types d'antenne énumérés ci-dessous et ayant un gain admissible maximal et l'impédance requise pour chaque type d'antenne. Les types d'antenne non inclus dans cette liste, ou dont le gain est supérieur au gain maximal indiqué, sont strictement interdits pour l'exploitation de l'émetteur

Le fonctionnement de l’ appareil est soumis aux deux conditions suivantes:

1. Cet appareil ne doit pas perturber les communications radio, et
2. cet appareil doit supporter toute perturbation, y compris les perturbations qui pourraient provoquer son dysfonctionnement.

Pour réduire le risque d'interférence aux autres utilisateurs, le type d'antenne et son gain doivent être choisis de façon que la puissance isotrope rayonnée équivalente (PIRE) ne dépasse pas celle nécessaire pour une communication réussie.

L’ appareil a été conçu pour fonctionner avec les antennes et les câbles énumérés dans les tables Antennes Autorisées et Câbles Autorisés. Il est strictement interdit de l’ utiliser l’ appareil avec des antennes ou câbles qui ne sont pas inclus dans ces listes.

Au but de conformer aux limites d'exposition RF pour la population générale (exposition non-contrôlée), les antennes utilisés doivent être installés à une distance d'au moins 25 cm de toute personne et ne doivent pas être installé en proximité ou utilisé en conjonction avec un autre antenne ou transmetteur.

Marquage sur l’ étiquette du produit complet dans un endroit visible: "Contient ThingMagic transmetteur, FCC ID: QV5MERCURY6E-A (IC:5407A-MERCURY6EA)"

Mercury6e Introduction

The ThingMagic® Mercury6e® (M6e) embedded module is an RFID engines that you can integrate with other systems to create RFID-enabled products.

Applications to control the M6e modules and derivative products can be written using the high level MercuryAPI. The MercuryAPI supports Java, .NET and C programming environments. The MercuryAPI Software Development Kit (SDK) contains sample applications and source code to help developers get started demoing and developing functionality. For more information on the MercuryAPI see the *MercuryAPI Programmers Guide* and the *MercuryAPI SDK*, available on the ThingMagic website.

This document is for hardware designers and software developers. It describes the hardware specifications and firmware functionality and provides guidance on how to incorporate the M6e module within a third-party host system. The rest of the document is broken down into the following sections:

- Hardware Overview - This section provides detailed specifications of the M6e hardware. This section should be read in its entirety before designing hardware or attempting to operate the M6e module in hardware other than the ThingMagic DevKit.
- Firmware Overview - This section describes provides a detailed description of the M6e firmware components including the bootloader and application firmware.
- Communication Protocol - This section provides an overview of the low level serial communications protocol used by the M6e.
- Functionality of the Mercury6e - This section provides detailed descriptions of the M6e features and functionality that are supported through the use of the MercuryAPI.
- Appendix A: Error Messages - This appendix lists and provides causes and suggested solutions for M6e Error Codes.
- Appendix B: Getting Started - Devkit - QuickStart guide to getting connected to the M6e Developer's Kit and using the Demo Applications included with the MercuryAPI SDK.

Hardware Overview

The following section provides detailed specifications of the M6e hardware including:

- Hardware Interfaces
- Power Requirements
- Environmental Specifications
- Assembly Information

Hardware Interfaces

Antenna Connections

The M6e supports four monostatic bidirectional RF antennas through four MMCX connectors: labeled J1 through J4 on the module. See Cables and Connectors for more information on antenna connector parts.

The maximum RF power that can be delivered to a 50 ohm load from each port is 1.4 Watts, or +31.5 dBm (operation above 30dBm requires a professional installer).

Note

The RF ports can only be energized one at a time.

Antenna Requirements

The performance of the M6e is affected by antenna quality. Antennas that provide good 50 ohm match at the operating frequency band perform best. Specified sensitivity performance is achieved with antennas providing 17 dB return loss or better across the operating band. Damage to the module will not occur for any return loss of 1 dB or greater. Damage may occur if antennas are disconnected during operation or if the module sees an open or short circuit at its antenna port.

Antenna Detection

To minimize the chance of damage due to antenna disconnection, the M6e supports antenna detection. Detection can be done automatically or manually, the choice of which is configured through API calls. Regardless of how it's used it is generally recommend that antenna detection be enabled as it helps protect the module from possible damage due to return losses less than 1 dB.

Digital/Power Connector

The digital connector provides power, serial communications signals, shutdown and reset signals to the M6e module, and access to the GPIO inputs and outputs. These signals are provided through connector part number: *Molex 53261-1571* - 1.25mm pin centers, 1

amp per pin rating. which mates with Molex housing *p/n 51021-1500* with crimps *p/n 63811-0300*. See Cables and Connectors for more information on typical cable parts.

M6e Digital Connector Signal Definition

Molex 53261-1571 Pin Number	Signal	Signal Direction (In/Out of M6e)	Notes
1	GND	P/S Return	Must connect both GND pins to ground
2	GND	P/S Return	
3	+5VDC	P/S Input	Must connect both 5V supplies
4	+5 VDC	P/S Input	
5	GPIO1	Bi-directional	Input 5VDC tolerant, 16mA Source/Sink
6	GPIO2	Bi-directional	
7	GPIO3	Bi-directional	
8	GPIO4	Bi-directional	
9	UART_RX_TTL	In	In + 10k Ohm to Ground
10	UART_TX_TTL	Out	Out
11	USB_DM	Bi-directional	USB Data (D-) signal
12	USB_DP	Bi-directional	USB Data (D+) signal
13	USB_5VSENSE	In	Input 5V to tell module to talk on USB
14	SHUTDOWN	In	Disable all 5V Inputs
15	RESET	Bi-directional	HIGH output indicates Boot Loader is running LOW output indicates Application Firmware is running Note: Not 5V tolerant.

Control Signal Specification

TTL Level UART Interface

The module communicates to a host processor via a TTL logic level UART serial port or via a USB port. Both ports are accessed on the 15-pin Digital/Power Connector. The TTL logic level UART supports complete functionality. The USB port supports complete functionality except the lowest power operational mode.

Note

Power Consumption specifications apply to control via the TTL UART.

TTL Level TX

V-Low: Max 0.4 VDC
V-High: 2.1 to 3.3 VDC
8 mA max

TTL Level RX

V-Low: -0.3 to 0.6 VDC
V-High: 2.2 to 5 VDC
(Tied to ground through a 10kOhm pull-up resistor. It is not harmful, but not recommended to drive the input above 3.3 V.)

A level converter could be necessary to interface to other devices that use standard 12V RS232. Only three pins are required for serial communication (TX, RX, and GND). Hardware handshaking is not supported. The M6e serial port has an interrupt-driven FIFO that empties into a circular buffer.

The connected host processor's receiver must have the capability to receive up to 256 bytes of data at a time without overflowing.

Baud rates supported:

- 9600
- 19200
- 38400
- 115200
- 230400
- 460800
- 921600

Note

The baudrate in the Boot Loader mode depends on whether the module entered the bootloader mode after a power-up or through an assert or "boot bootloader" user command. Upon power up if the Reset Line is LOW then the default baud rate of 9600 will be used. If the module returns to the bootloader from Application Firmware mode, then the current state and baudrate will be retained.

USB Interface

Supports USB 2.0 full speed device port (12 Megabits per second) using the two USB pins (USB_DM and USB_DP).

General Purpose Input/Output (GPIO)

The four GPIO connections, provided through the M6e Digital Connector Signal Definition, may be configured as inputs or outputs using the MercuryAPI. The GPIO pins connect through 100 ohm resistors to the high current PA0 to PA3 pins of the AT91SAM7X processor. The processor data sheet can be consulted for additional details.

Pins configured as inputs must not have input voltages that exceed voltage range of -0.3 volts to +5.5 volts. In addition, during reset the input voltages should not exceed 3.3V.

Outputs may source and sink 16 mA. Voltage drop in the series 100 ohm resistor will reduce the delivered voltage swing for output loads that draw significant current.

Input Mode

- TTL compatible inputs,
- Logic low < 0.8 V,
- Logic high > 2.0V.
- 5V tolerant

Output Mode

- 3.3 Volt CMOS Logic Output with 100 ohms in series.
- Greater than 1.9 Volts when sourcing 8 mA.
- Greater than 2.9 Volts when sourcing 0.3 mA.
- Less than 1.2 Volts when sinking 8 mA.
- Less than 0.2 Volts when sinking 0.3 mA.

Module power consumption can be adversely affected by incorrect GPIO configuration. Similarly, the power consumption of external equipment connected to the GPIOs can also be adversely affected. The following instructions will yield specification compliant operation.

On power up, the M6E module configures its GPIOs as outputs to avoid contention from user equipment that may be driving those lines. The input configuration is as a 3.3 volt logic CMOS input and will have a leakage current not in excess of 400 nA. The input is in an undetermined logic level unless pulled externally to a logic high or low. **Module power consumption for floating inputs is unspecified**. With the GPIOs configured as inputs and individually pulled externally to either high or low logic level, module power consumption is as listed in the M6e Power Consumption table.

GPIOs may be reconfigured individually after power up to become outputs. This configuration takes effect either at API execution or a few tens of milliseconds after power up if the configuration is stored in nonvolatile memory. The configuration to outputs is

defeated if the module is held in the boot loader by Reset Line being held low. Lines configured as outputs consume no excess power if the output is left open. Specified module power consumption is achieved for one or more GPIO lines set as output and left open. Users who are not able to provide external pull ups or pull downs on any given input, and who do not need that GPIO line, may configure it as an output and leave it open to achieve specified module power consumption.

Configuring GPIO Settings

The GPIO lines are configured as inputs or outputs through the MercuryAPI by setting the reader configuration parameters /reader/gpio/inputList and /reader/gpio/outputList. Once configured as inputs or outputs the state of the lines can be Get or Set using the gpiGet() and gpoSet() methods, respectively. See the language specific reference guide for more details.

Reset Line

Upon power up the RESET (pin 15) line is configured as an input. The input value will determine whether the Boot Loader will wait for user commands or immediately load the Application Firmware image and enter application mode. After that action is completed, this line is configured as an output line. While the unit continues to be in bootloader the line is driven high.

Once in application mode, the RESET line is driven low. if the module returns to the bootloader mode, either due to an assert or “boot bootloader”, the RESET line will again be driven high.

To minimize power consumption in the application, the RESET line should be either left open or pulled weakly low (10k to ground).

Power Requirements

RF Power Output

The M6e supports separate read and write power level which are command adjustable via the MercuryAPI. Power levels must be between:

- Minimum RF Power = +5 dBm
- Maximum RF Power = +31.5 dBm *(+0.0/- 0.5 dB accuracy above +15 dBm)*

Note

Maximum power may have to be reduced to meet regulatory limits, which specify the combined effect of the module, antenna, cable and enclosure shielding of the integrated product.

W A R N I N G !

Operation requires professional installation to correctly set the TX power for the RF cable and antenna selected.

Power Settings for Authorized Antennas and Cables

This device has been designed to operate with the antennas listed in Authorized Antennas list using the cables in the Authorized Cables list. For any combination of antenna and cable the maximum RF power is determined from antenna gain (Max Linear Gain value from antenna list) and antenna cable loss (Insertion Loss value from cable list) using the formula:

Pmax = 36 dBm - Antenna Gain + Cable Loss

For example, for the Laird S8658WPL and the ThingMagic CBL-P6 6ft cable the following calculation can be performed:

Max linear antenna gain = 6 dBiL

Minimum cable insertion loss = 0.8 dB

Pmax = 36 - 6 + 0.8 = 30.8 dBm

The maximum RF power that may be set using this configuration is 30.8 dBm (see Warning above).

Power Supply Ripple

The following are the minimum requirements to avoid module damage and to insure performance and regulatory specifications are met. Certain local regulatory specifications may require tighter specifications.

- 5 Volt +/- 5%,
- Less than 25 mV pk-pk ripple all frequencies,
- Less than 11 mV pk-pk ripple for frequencies less than 100 kHz,
- No spectral spike greater than 5 mV pk-pk in any 1 kHz band.

Power Consumption

The following table defines the power/transmit mode settings and power consumption specifications for the M6e. Additional details about Power/Transmit Modes can be found in the Power Management section.

M6e Power Consumption

Operation **Power/Transmit Mode**	**Max Power**[1] **(Watts)**	**Voltage (Volts)**	**Current (mA)**	**RF Transmit Power Setting (dBm)**
Transmit CW Transmit Mode = DRM	7.5[2]	5.0 +/- 5%	1400	+31.5
Tag Reading Transmit Mode = DRM	7.5[2]	5.0 +/- 5%	1400	+31.5
Tag Reading Transmit Mode = Low Power	5.8	5.0 +/- 5%	1060	+30
Tag Reading Transmit Mode = DRM + PreDistortion	6.2	5.0 +/- 5%	1200	+30
Tag Reading Transmit Mode = DRM	2.5	5.0 +/- 5%	490	+5
No Tag Reading (M6e idle) Power Mode = 0	0.35	5.0 +/- 5%	60	N/A
No Tag Reading (M6e idle) Power Mode = 1	0.12	5.0 +/- 5%	20	N/A
No Tag Reading (M6e idle) Power Mode = 2	0.005	5.0 +/- 5%	1.0	N/A
Boot	0.12	5.0 +/- 5%	20	N/A

M6e Power Consumption

Operation **Power/Transmit Mode**	**Max Power**[1] **(Watts)**	**Voltage (Volts)**	**Current (mA)**	**RF Transmit Power Setting (dBm)**
Shut Down	< 0.001	5.0 +/- 5%	< 200uA	N/A
In Rush Current and Power, M6e Power up and/or any state change	7.5	5.0 +/- 5%	1500 Max	N/A

Note: 1 - Power consumption is defined for TTL RS232 operation. Power consumption may vary if the USB interface is connected.

Note: 2 - Power consumption is defined for operation into a 17dB return loss load or better. Power consumption may increase beyond this specification during operation into return losses worse than 17dB.

Environmental Specifications

Operating Temperature

Clamshell temperature must not exceed 70 degrees C. Heat sinking will be required for high duty cycle applications.

Electro-Static Discharge (ESD) Specification

Specifications to be determined.

Assembly Information

Cables and Connectors

The following are the cables and connectors used in the M6e Developer's Kit interface board:

Digital Interface

The cable assembly used consists of the following parts:

- 2 Connector Shells [Molex 51021-1500] with 15 Crimp Contacts each [Molex 50079-8100]
- 1 Wire (#28 AWG 7x36 - Black, Teflon) for Pin 1 connection [Alpha 284/7-2]
- 14 Wires (#28 AWG 7x36 - White, Teflon) for other connections [Alpha 284/7-1]

Note

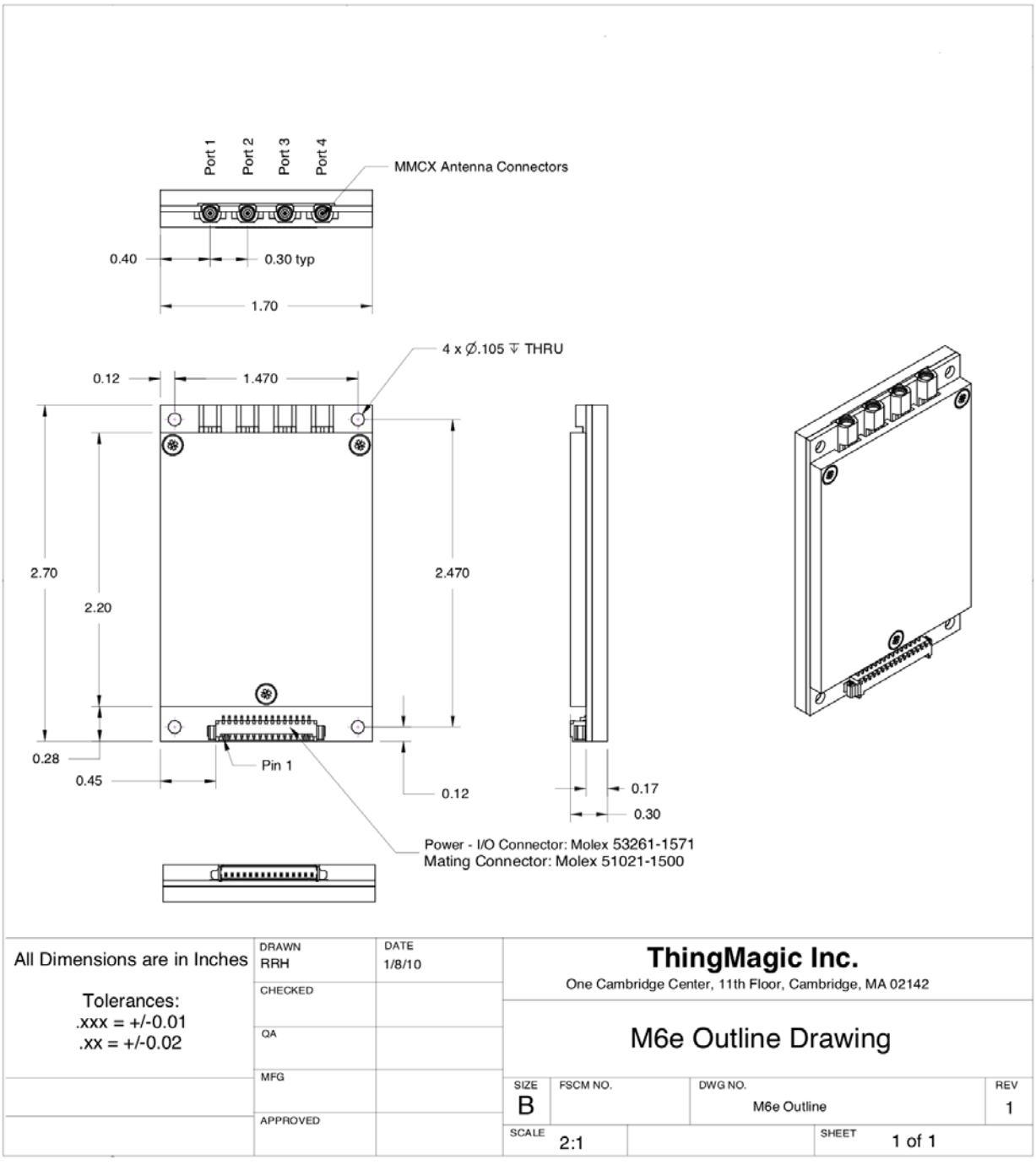
Pin numbers and assignments are shown in the M6e Digital Connector Signal Definition table.

Antennas

The cable assembly used to connect the "external" RP-TNC connectors on the M6e Devkit to the M6e MMCX connectors consists of the following parts:

- 1 Reverse TNC Bulkhead Jack Connector
- 1 LMR-100A Coaxial Cable
- 1 MMCX Right Angle Plug Connector

M6e Mechanical Drawing



Authorized Antennas

This device has been designed to operate with the antennas listed below, and having a maximum gain of 6 dBiL. Antennas not included in this list or having a gain greater than 6 dBiL are strictly prohibited for use with this device. The required antenna impedance is 50 ohms

.

Manufacturer	Type	Manufacturer Part Number	Max. Linear Gain (dBiL)
Laird	Patch	S9025P	4.3
Laird	Patch	S8658WPL	6.0
Laird	Patch	DCE8658WPR	6.0 dBiL
Laird	Patch	PEL90206	4.7 dBiL
Laird	Patch	S9026X	4.5 dBiL
MTI	Patch	MT-262013	6.0 dBiL
MTI	Patch	MT-242043	6.0 dBiL
MTI	Patch	MT-242042	4.1 dBiL
MTI	Patch	MT-241026	0.2 dBiL
Poynting	Patch	A-PATCH-0025	5 dBiL
Mobile Mark	Patch	PN8-915	6.0 dBiL
Mobile Mark	Patch	PN7-915	4.5 dBiL
Mobile Mark	Patch	PN6-915	4.0 dBiL

Authorized Cables

The following table contains the cable loss values for authorized shielded coaxial cables provided by ThingMagic

Cable Description	ThingMagic Part Number	Insertion Loss
6' RTNC to RTNC Cable	CBL-P6	0.8 dB
12' RTNC to RTNC Cable	CBL-P12	1.5 dB
20' RTNC to RTNC Cable	CBL-P20	2.4 dB
20' RTNC to RTNC Plenum Cable	CBL-P20-PL	2.4 dB
25' RTNC to RTNC Cable	CBL-P25	3.0 dB

Firmware Overview

The following section provides detailed description of the M6e firmware components:

- Boot Loader
- Application Firmware
- Custom On-Reader Applications

Boot Loader

The boot loader provides low-level functionality. This program provides the low level hardware support for configuring communication settings, loading Application Firmware and storing and retrieving data to/from flash.

When a module is powered up or reset, the boot loader code is automatically loaded and executed.

Note

Unlike previous ThingMagic modules (M4e and M5e) the M6e bootloader should effectively be invisible to the user. The M6e is by default configured to auto-boot into application firmware and for any operations that require the module be in bootloader mode the MercuryAPI will handle the switching automatically.

Application Firmware

The application firmware contains the tag protocol code along with all the command interfaces to set and get system parameters and perform tag operations. The application firmware is, by default, started automatically upon power up.

Programming the M6e

Applications to control the M6e module and derivative products are written using the high level MercuryAPI. The MercuryAPI supports Java, .NET and C programming environments. The MercuryAPI Software Development Kit (SDK) contains sample applications and source code to help developers get started demoing and developing functionality. For more information on the MercuryAPI see the *MercuryAPI Programmers Guide* and the *MercuryAPI SDK*, available on the ThingMagic website.

Upgrading the M6e

New features developed for the M6e are made available to existing modules through an Application Firmware upgrade, along with corresponding updates to the MercuryAPI to make use of the new features. Firmware upgrades can be applied using the MercuryAPI to build the functionality into custom applications or using the MercuryAPI SDK demo utilities.

Verifying Application Firmware Image

The application firmware has an image level Cyclic Redundancy Check (CRC) embedded in it to protect against corrupted firmware during an upgrade process. (If the upgrade is unsuccessful, the CRC will not match the contents in flash.) When the boot loader starts the application FW, it first verifies that the image CRC is correct. If this check fails, then the boot loader does not start the application firmware and an error is returned.

Custom On-Reader Applications

The M6e does not support installing customer applications on the reader. Continuous reader, tag streaming, scripting and other methods of configuring the module to operate in an autonomous or semi-autonomous reading modes maybe supported through the MercuryAPI but custom application **cannot** be installed on the module.

Communication Protocol

The following section provides an overview of the low level serial communications protocol used by the M6e.

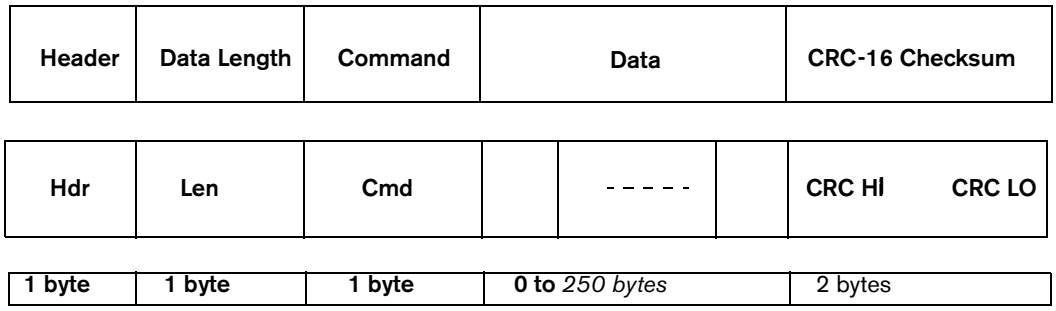
Serial Communication Protocol

The serial communication between a computer (host) and the M6e is based on a synchronized command-response/master-slave mechanism. Whenever the host sends a message to the reader, it cannot send another message until after it receives a response. The reader never initiates a communication session; only the host initiates a communication session.

This protocol allows for each command to have its own timeout because some commands require more time to execute than others. The host must manage retries, if necessary. The host must keep track of the state of the intended reader if it reissues a command.

Host-to-Reader Communication

Host-to-reader communication is packetized according to the following diagram. The reader can only accept one command at a time, and commands are executed serially, so the host waits for a reader-to-host response before issuing another host-to-reader command packet.



Reader-to-Host Communication

The following diagram defines the format of the generic Response Packet sent from the reader to the host. The Response Packet is different in format from the Request Packet.

Header	**Data Length**	**Command**	Status Word	**Data**	**CRC-16 Checksum**
Hdr	**Len**	**Cmd**	Status Word	- - - - -	**CRC HI** **CRC LO**
1 byte	**1 byte**	**1 byte**	*2 bytes*	0 to *248* bytes	2 bytes

CCITT CRC-16 Calculation

The same CRC calculation is performed on all serial communications between the host and the reader. The CRC is calculated on the Data Length, Command, Status Word, and Data bytes. The header is not included in the CRC.

User Programming Interface

The M6e does not support programming to the serial protocol directly. All user interaction with the M6e must be performed using the MercuryAPI.

The MercuryAPI supports Java, .NET and C programming environments. The MercuryAPI Software Development Kit (SDK) contains sample applications and source code to help developers get started demoing and developing functionality. For more information on the MercuryAPI see the *MercuryAPI Programmers Guide* and the *MercuryAPI SDK*, available on the ThingMagic website.

Functionality of the Mercury6e

The following section provides detailed descriptions of the M6e features and functionality that are supported through the use of the MercuryAPI.

Regulatory Support

Supported Regions

The M6e has differing levels of support for operation and use under the laws and guidelines of several regions. The regional support is shown in the following table.

Supported Regions

Region	Regulatory Support	Notes
North America (NA)	FCC 47 CFG Ch. 1 Part 15 Industrie Canada RSS-210	

The regional functionality is set using the MercuryAPI. Setting the region of operation configures the regional default settings including:

- Loads the Frequency Hop Table with the appropriate table for the selected region.
- Sets the PLL Frequency Setting to the first entry in the hop table, even if the RF is off.
- Selects the transmit filter, if applicable.

Frequency Setting

The modules have a PLL synthesizer that sets the modulation frequency to the desired value. Whenever the frequency is changed, the module must first power off the modulation, change the frequency, and then turn on the modulation again. Since this can take several milliseconds, it is possible that tags are powered off during a frequency hop. In addition to setting the default regional settings, the M6e has commands that allow the transmit frequency to be set manually.

C A U T I O N !

Use these commands with extreme caution. It is possible to change the module's compliance with the regional regulations.

Frequency Units

All frequencies in the M6e are expressed in kHz using unsigned 32-bit integers. For instance, a carrier frequency of 915 MHz is expressed as 915000 kHz.

The PLL is set automatically to the closest frequency - based on the minimum frequency quantization for the current region - that matches the specified value. The M6e has an absolute minimum quantization of 50 kHz. Each region also has a minimum quantization based on regulatory specifications, which may be greater. The following table details the frequency quantization in kHz for each region setting.

Regional Frequency Quantization

Region	Frequency Quantization	Minimum Frequency	Maximum Frequency
NA	250 kHz	902,000 kHz	928,000 kHz

When manually setting frequencies the module will round down for any value that is not an even multiple of the supported frequency quantization.

For example: In the NA region, setting a frequency of 902,999 kHz results in a setting of 902,750 kHz.

When setting the frequency of the module, any frequencies outside of the valid range for the specified region are rejected.

Frequency Hop Table

The frequency hop table determines the frequencies used by the M6e when transmitting. The hop table characteristics are:

- Contains up to 62 slots.
- Valid frequencies for the region currently selected.
- Changes not stored in flash, thus changes made are not retained after a power cycle or a restart of the boot loader.
- Inability to change individual entries after uploading without reloading the entire table.
- Frequencies used in the order of entries in the table.

If necessary for a region, the hop table can be randomized to create a pseudo-random sequence of frequencies to use. This is done automatically using the default hop tables provided for each region.

Protocol Support

The M6e has the ability to support many different tag protocols. Using the MercuryAPI ReadPlan classes the M6e can be configured to single or multi-protocol Read operations. The current protocols supported are (some may require a license to enable):

- ISO 18000-6C (Gen2)
- I-PX
- ISO 18000-6B

ISO 18000-6C (Gen2)

Protocol Configuration Options

The M6e supports multiple ISO-18000-6C profiles including the ability to specify the Link Frequency, encoding schemes, Tari value and modulation scheme. The protocol options are set in the MercuryAPI Reader Configuration Parameters (/reader/gen2/*). The following table shows the supported combinations:

ISO-18000-6C Protocol Options

Backscatter Link Frequency (kHz)	Encoding	Tari (usec)	Modulation Scheme	Notes
250	Miller (M=8)	12.5	PR-ASK	
250	Miller (M=4)	12.5	PR-ASK	
250	Miller (M=2)	12.5	PR-ASK	
250	FM0	12.5	PR-ASK	
250	Miller (M=8)	25	PR-ASK	
250	Miller (M=4)	25	PR-ASK	
250	Miller (M=2)	25	PR-ASK	
250	FM0	25	PR-ASK	
250	Miller (M=8)	25	PR-ASK	
640	FM0	6.25	PR-ASK	Not supported in PRC Region

Protocol Specific Functionality

See the *MercuryAPI Programmers Guide* and language specific reference guides for details on supported Gen2 command functionality.

I-PX

Protocol Configuration Options

The M6e supports multiple I-PX profiles including the ability to specify the Return Link Frequency, encoding and modulation scheme. The two profiles are treated as distinct protocols, the individual parameters are not configurable as with the other protocols. The following table shows the supported combinations:

ISO-18000-6B Protocol Options

Return Link Freq (kHz)	Modulation Scheme	Notes
64	PWM	Protocol ID = TagProtocol.IPX64
256	PWM	Protocol ID = TagProtocol.IPX256

Note

The two link rates are effectively two different protocols and treated as such. I-PX tags are fixed to one of the two frequencies and cannot communicate on the other, unlike ISO 18000-6B/C tags which can operate under multiple profiles.

ISO 18000-6B

Protocol Configuration Options

The M6e supports multiple ISO-18000-6B profiles including the ability to specify the Return Link Frequency, encoding, Forward Link Rate and modulation scheme. The protocol options are set in the MercuryAPI Reader Configuration Parameters (/reader/iso18000-6b/*). The following table shows the supported combinations:

ISO-18000-6B Protocol Options

Return Link Freq (kHz)	Return Encoding	Forward Link Freq (kHz)	Forward Encoding	Notes
40	FM0	10	Manchester	1
160	FM0	40	Manchester	2

Antenna Ports

The M6e has four monostatic antenna ports. Each port is capable of both transmitting and receiving. The modules also support Using a Multiplexer, allowing up to 16 total logical antenna ports, controlled using two GPIO lines and the internal physical port J1/J2/J3/J4 switching.

Note

The M6e does not support bistatic operation.

Using a Multiplexer

Multiplexer switching is controlled through the use of the internal module physical port J1/J2/J3/J4 switch along with the use of one or more of the General Purpose Input/Output (GPIO) lines. In order to enable automatic multiplexer port switching the module must be configured to use *Use GPIO as Antenna Switch* in `/reader/antenna/portSwitchGpos`.

Once the GPIO line(s) usage has been enabled the following control line states are applied when the different Logical Antenna settings are used. The tables below show the mapping that results using GPIO 1 and 2 for multiplexer control (as is used by the ThingMagic 1 to 4 multiplexer) allowing for 16 logical antenna ports.

Note

The Logical Antenna values are static labels indicating the available control line states. The specific physical antenna port they map to depends on the control line to antenna port map of the multiplexer in use. The translation from Logical Antenna label to physical port must be maintained by the control software.

GPIO 1 & 2 Used for Antenna Switching

Logical Antenna Setting	GPIO Output 1 State	GPIO Output 2 State	Active M6e Physical Port
1	Low	Low	J1
2	Low	Low	J2
3	Low	Low	J3
4	Low	Low	J4
5	Low	High	J1
6	Low	High	J2
7	Low	High	J3
8	Low	High	J4
9	High	Low	J1
10	High	Low	J2
11	High	Low	J3
12	High	Low	J4
13	High	High	J1
14	High	High	J2
15	High	High	J3
16	High	High	J4

If only one GPIO Output line is used for antenna control, the combinations of the available output control line states (the GPIO line in use and the module port) result in a subset of logical antenna settings which can be used.

ONLY GPIO 1 Used for Antenna Switching

Logical Antenna Setting	GPIO Output 1 State	Active M6e Physical Port
1	Low	J1
2	Low	J2

Logical Antenna Setting	GPIO Output 1 State	Active M6e Physical Port
3	Low	J3
4	Low	J4
9	High	J1
10	High	J2
11	High	J3
12	High	J4

Note

The “missing” logical antenna settings are still usable when only one GPIO line is used for antenna control and simply results in redundant logical antenna settings. For example, using only GPIO 1, logical setting 4 and 8 both result in GPIO1=Low and M6e port J4 active.

ONLY GPIO 2 Used for Antenna Switching

Logical Antenna Setting	GPIO Output 2 State	Active M6e Physical Port
1	Low	J1
2	Low	J2
3	Low	J3
4	Low	J4
5	High	J1
6	High	J2
7	High	J3
8	High	J4

Port Power and Settling Time

The M6e allows the power and settling time for each logical antenna to be set using the reader configuration parameters `/reader/radio/portReadPowerList` and `/`

`reader/antenna/settlingTimeList`, respectively. The order the antennas settings are defined does not affect search order.

Note

Settling time is the time between the control lines switching to the next antenna setting and RF turning on for operations on that port. This allows time for external multiplexer's to fully switch to the new port before a signal is sent, if necessary. Default value is 0.

Tag Handling

When the M6e performs inventory operations (MercuryAPI Read commands) data is stored in a Tag Buffer until retrieved by the client application, or streamed directly to the client if operating in Streaming mode [Not Yet Implemented].

Tag Buffer

The M6e uses a dynamic buffer that depends on EPC length and quantity of data read. As a rule of thumb it can store a maximum of 1024 96-bit EPC tags in the TagBuffer at a time. Since the M6e supports streaming of read results the buffer limit is, typically, not an issue. Each tag entry consists of a variable number of bytes and consists of the following fields:

Tag Buffer Entry

Total Entry Size	Field	Size	Description
68 bytes (Max EPC Length = 496bits)	EPC Length	2 bytes	Indicates the actual EPC length of the tag read. Cannot exceed the Max EPC length setting.
	PC Word	2 bytes	Contains the Protocol Control bits for the tag.
	EPC	62 bytes	Contains the tag's EPC value padded with trailing zeros if the size is less than the Max EPC Length size.
	Tag CRC	2 bytes	The tag's CRC.
Tag Read Meta Data			

The Tag buffer acts as a First In First Out (FIFO) — the first Tag found by the reader is the first one to be read out.

Tag Streaming

When reading tags during inventory operations (MercuryAPI `Reader.Read()` and `Reader.StartReading()`) by default the M6e "streams" the tag results back to the host processor. This means that tags are pushed out of the buffer as soon as they are processed by the M6e and put into the buffer. The buffer is put into a circular mode that keeps the buffer from filling. This allows for the M6e to perform continuous search operations without the need to periodically stop reading and fetch the contents of the buffer. Aside from not seeing "down time" when performing a read operation this behavior is essentially invisible to the user as all tag handling is done by the MercuryAPI.

Tag Read Meta Data

In addition to the tag EPC ID resulting from M6e inventory operation each `TagReadData` (see MercuryAPI for code details) contains meta data about how, where and when the tag was read. The specific meta data available for each tag read is as follows:

Tag Read Meta Data

Meta Data Field	Description
Antenna ID	The antenna on with the tag was read. If the same tag is read on more than one antenna there will be a tag buffer entry for each antenna on which the tag was read. When Using a Multiplexer, if appropriately configured, the Antenna ID entry will contain the logical antenna port of the tag read.
Read Count	The number of times the tag was read on [Antenna ID].
Timestamp	The time the tag was read, relative to the time the command to read was issued, in milliseconds. If the Tag Read Meta Data is not retrieved from the Tag Buffer between read commands there will be no way to distinguish order of tags read with different read command invocations.
Tag Data	When reading an embedded `TagOp` is specified for a `ReadPlan` the TagReadData will contain the first 4 bytes of data returned for each tag. Note: Tags with the same TagID but different Tag Data can be considered unique and each get a Tag Buffer entry if set in the reader configuration parameter `/reader/tagReadData/uniqueByData`. By default it is not.
Frequency	The frequency on which the tag was read
Tag Phase	Average phase of tag response in degrees
LQI/RSSI	The receive signal strength of the tag response in dBm.
GPIO Status	The signal status (High or Low) of all GPIO pins when tag was read.

Power Management

The M6e is designed for power efficiency and offers several different power management modes. The following power management modes affect the power consumption during different periods of M6e usage and impact performance in different ways. The available power management modes are:

- Power Modes - set in /reader/powerMode - Controls the power savings when the M6e is idle.
- Transmit Modes - set in

Power Modes

The Power Mode setting (set in `/reader/powerMode`) allows the user to trade off increased RF operation startup time for additional power savings. The details of the amount of power consumed in each mode is shown in the table under Power Consumption. The behavior of each mode and impact on RF command latency is as follows:

- **Power Mode 0** – In this mode, the unit operates at full power to attain the best performance possible. This mode is only intended for use in cases where power consumption is not an issue. This is the default Power Mode at startup.
- **Power Mode 1** – This mode may add up to 50 ms of delay from idle to RF on when initiating an RF operation. It performs more aggressive power savings, such as automatically shutting down the analog section between commands, and then restarting it whenever a tag command is issued.
- **Power Mode 2** – This mode essentially shuts down the digital and analog boards, except to power the bare minimum logic required to wake the processor. This mode may add up to 100 ms of delay from idle to RF on when initiating an RF operation. **Power Mode 2 is not supported when using the USB interface**.

Note

See additional latency specifications under Event Response Times.

Transmit Modes

The Transmit Mode setting (set in `/reader/radio/enablePowerSave`) allows the user to trade off RF spectral compliance with the Gen2 DRM Mask for increased power savings while transmitting. The details of the amount of power consumed in each mode is shown in the table under Power Consumption. The behavior of each mode is as follows:

DRM Compliant Mode

This mode maximizes performance in dense reader environments, minimizing interference when used with other M6e or similar DRM-compliant readers, and is fully compliant with the Gen2 DRM spectral mask.

Power Save Mode (non-DRM Compliant)

This mode reduces the power consumption during RF operations but is not 100% compliant with the DRM spectral mask. This can result increased interference with other readers and reduce overall systems performance.

Performance Characteristics

Event Response Times

The following table provides some metrics on how long common M6e operations take. An event response time is defined as the maximum time from the end of a command (end of the last bit in the serial stream) or event (e.g. power up) to the response event the command or event causes.

Event Response Times

Start Command/ Event	End Event	Time (msecs)	Notes
Power Up	Application Active (with CRC check)	800	This longer power up period should only occur for the first boot with new firmware.
Power Up	Application Active	100	Once the firmware CRC has been verified subsequent power ups do not require the CRC check be performed, saving time.
Tag Read	RF On	20	When in Power Mode = 0
Tag Read	RF On	50	When in Power Mode = 1
Tag Read	RF On	120	When in Power Mode = 2
Change to Mode 1	Power Mode 1	5	From Power Mode =0
Change to Mode 2	Power Mode 2	5	From Power Mode =0

Save and Restore Configuration

The M6e supports saving module and protocol configuration parameters to the module flash to provide configuration persistence across boots. See the *MercuryAPI Programmers Guide* and sample applications for details on saving and restoring reader configuration.

Appendix A: Error Messages

Common Error Messages

The following table lists the common faults discussed in this section.

Fault Message	Code
FAULT_MSG_WRONG_NUMBER_OF_DATA – (100h)	100h
FAULT_INVALID_OPCODE – (101h)	101h
FAULT_UNIMPLEMENTED_OPCODE – 102h	102h
FAULT_MSG_POWER_TOO_HIGH – 103h	103h
FAULT_MSG_INVALID_FREQ_RECEIVED (104h)	104h
FAULT_MSG_INVALID_PARAMETER_VALUE - (105h)	105h
FAULT_MSG_POWER_TOO_LOW - (106h)	106h
FAULT_UNIMPLEMENTED_FEATURE - (109h)	109h
FAULT_INVALID_BAUD_RATE - (10Ah)	10Ah

FAULT_MSG_WRONG_NUMBER_OF_DATA – (100h)

Cause

If the data length in any of the Host-to-M5e/M5e-Compact messages is less than or more than the number of arguments in the message, the reader returns this message.

Solution

Make sure the number of arguments matches the data length.

FAULT_INVALID_OPCODE – (101h)

Cause

The opCode received is invalid or not supported in the currently running program (bootloader or main application) or is not supported in the current version of code.

Solution

Check the following:

- Make sure the command is supported in the currently running program.
- Check the documentation for the opCode the host sent and make sure it is correct and supported.
- Check the previous module responses for an assert (0x7F0X) which will reset the module into the bootloader.

FAULT_UNIMPLEMENTED_OPCODE – 102h

Cause

Some of the reserved commands might return this error code.

This does not mean that they always will do this since ThingMagic reserves the right to modify those commands at anytime.

Solution

Check the documentation for the opCode the host sent to the reader and make sure it is supported.

FAULT_MSG_POWER_TOO_HIGH – 103h

Cause

A message was sent to set the read or write power to a level that is higher than the current HW supports.

Solution

Check the HW specifications for the supported powers and insure that the level is not exceeded.

The M5e 1 Watt units support power from 5 dBm to 30 dBm.

The M5e-Compact units support power from 10 dBm to 23 dBm.

FAULT_MSG_INVALID_FREQ_RECEIVED (104h)

Cause

A message was received by the reader to set the frequency outside the supported range

Solution

Make sure the host does not set the frequency outside this range or any other locally supported ranges.

FAULT_MSG_INVALID_PARAMETER_VALUE - (105h)

Cause

The reader received a valid command with an unsupported or invalid value within this command.

For example, currently the module supports four antennas. If the module receives a message with an antenna value other than 1 to 4, it returns this error.

Solution

Make sure the host sets all the values in a command according to the values published in this document.

FAULT_MSG_POWER_TOO_LOW - (106h)

Cause

A message was received to set the read or write power to a level that is lower than the current HW supports.

Solution

Check the HW specifications for the supported powers and insure that level is not exceeded. The M6e supports powers between 5 and 31.5 dBm.

FAULT_UNIMPLEMENTED_FEATURE - (109h)

Cause

Attempting to invoke a command not supported on this firmware or hardware.

Solution

Check the command being invoked against the documentation.

FAULT_INVALID_BAUD_RATE - (10Ah)

Cause

When the baud rate is set to a rate that is not specified in the Baud Rate table, this error message is returned.

Solution

Check the table of specific baud rates and select a baud rate.

Bootloader Faults

The following table lists the common faults discussed in this section.

Fault Message	Code
FAULT_BL_INVALID_IMAGE_CRC	200h
FAULT_BL_INVALID_APP_END_ADDR	201h

FAULT_BL_INVALID_IMAGE_CRC – 200h

Cause

When the application firmware is loaded the reader checks the image stored in flash and returns this error if the calculated CRC is different than the one stored in flash.

Solution

The exact reason for the corruption could be that the image loaded in flash was corrupted during the transfer or corrupted for some other reason.

To fix this problem, reload the application code in flash.

FAULT_BL_INVALID_APP_END_ADDR – 201h

Cause

When the application firmware is loaded the reader checks the image stored in flash and returns this error if the last word stored in flash does not have the correct address value.

Solution

The exact reason for the corruption could be that the image loaded in flash got corrupted during the transfer or, corrupted for some other reason.

To fix this problem, reload the application code in flash.

Flash Faults

The following table lists the common faults discussed in this section.

Fault Message	Code
FAULT_FLASH_BAD_ERASE_PASSWORD – 300h	300h
FAULT_FLASH_BAD_WRITE_PASSWORD – 301h	301h
FAULT_FLASH_UNDEFINED_ERROR – 302h	302h
FAULT_FLASH_ILLEGAL_SECTOR – 303h	303h
FAULT_FLASH_WRITE_TO_NON_ERASED_AREA – 304h	304h
FAULT_FLASH_WRITE_TO_ILLEGAL_SECTOR – 305h	305h
FAULT_FLASH_VERIFY_FAILED – 306h	306h

FAULT_FLASH_BAD_ERASE_PASSWORD – 300h

Cause

A command was received to erase some part of the flash but the password supplied with the command was incorrect.

Solution

When this occurs make note of the operations you were executing, save FULL error response and send a testcase reproducing the behavior to support@thingmagic.com.

FAULT_FLASH_BAD_WRITE_PASSWORD – 301h

Cause

A command was received to write some part of the flash but the password supplied with the command was not correct.

Solution

When this occurs make note of the operations you were executing, save FULL error response and send a testcase reproducing the behavior to support@thingmagic.com.

FAULT_FLASH_UNDEFINED_ERROR – 302h

Cause

This is an internal error and it is caused by a software problem in module.

Solution

When this occurs make note of the operations you were executing, save FULL error response and send a testcase reproducing the behavior to support@thingmagic.com.

FAULT_FLASH_ILLEGAL_SECTOR – 303h

Cause

An erase or write flash command was received with the sector value and password not matching.

Solution

When this occurs make note of the operations you were executing, save FULL error response and send a testcase reproducing the behavior to support@thingmagic.com.

FAULT_FLASH_WRITE_TO_NON_ERASED_AREA – 304h

Cause

The module received a write flash command to an area of flash that was not previously erased.

Solution

When this occurs make note of the operations you were executing, save FULL error response and send a testcase reproducing the behavior to support@thingmagic.com.

FAULT_FLASH_WRITE_TO_ILLEGAL_SECTOR – 305h

Cause

The module received a write flash command to write across a sector boundary that is prohibited.

Solution

When this occurs make note of the operations you were executing, save FULL error response and send a testcase reproducing the behavior to support@thingmagic.com.

FAULT_FLASH_VERIFY_FAILED – 306h

Cause

The module received a write flash command that was unsuccessful because data being written to flash contained an uneven number of bytes.

Solution

When this occurs make note of the operations you were executing, save FULL error response and send a testcase reproducing the behavior to support@thingmagic.com.

Protocol Faults

The following table lists the common faults discussed in this section.

Fault Message	Code
FAULT_NO_TAGS_FOUND – (400h)	400h
FAULT_NO_PROTOCOL_DEFINED – 401h	401h
FAULT_INVALID_PROTOCOL_SPECIFIED – 402h	402h
FAULT_WRITE_PASSED_LOCK_FAILED – 403h	403h
FAULT_PROTOCOL_NO_DATA_READ – 404h	404h
FAULT_AFE_NOT_ON – 405h	405h
FAULT_PROTOCOL_WRITE_FAILED – 406h	406h
FAULT_NOT_IMPLEMENTED_FOR_THIS_PROTOCOL – 407h	407h
FAULT_PROTOCOL_INVALID_WRITE_DATA – 408h	408h
FAULT_PROTOCOL_INVALID_ADDRESS – 409h	409h
FAULT_GENERAL_TAG_ERROR – 40Ah	40Ah
FAULT_DATA_TOO_LARGE – 40Bh	40Bh
FAULT_PROTOCOL_INVALID_KILL_PASSWORD – 40Ch	40Ch
FAULT_PROTOCOL_KILL_FAILED - 40Eh	40Eh
FAULT_PROTOCOL_BIT_DECODING_FAILED - 40Fh	40Fh
FAULT_PROTOCOL_INVALID_EPC – 410h	410h
FAULT_PROTOCOL_INVALID_NUM_DATA – 411h	411h
FAULT_GEN2 PROTOCOL_OTHER_ERROR - 420h	420h
FAULT_GEN2 PROTOCOL_MEMORY_OVERRUN_BAD_PC - 423h	423h
FAULT_GEN2 PROTOCOL_MEMORY_LOCKED - 424h	424h
FAULT_GEN2 PROTOCOL_INSUFFICIENT_POWER - 42Bh	42Bh
FAULT_GEN2 PROTOCOL_NON_SPECIFIC_ERROR - 42Fh	42Fh
FAULT_GEN2 PROTOCOL_UNKNOWN_ERROR - 430h	430h

FAULT_NO_TAGS_FOUND – (400h)

Cause

A command was received (such as like read, write, or lock) but the operation failed. There are many reasons that can cause this error to occur.

Here is a list of possible reasons that could be causing this error:

- No tag in the RF field
- Read/write power too low
- Antenna not connected
- Tag is weak or dead

Solution

Make sure there is a good tag in the field and all parameters are set up correctly. The best way to check this is to try few tags of the same type to rule out a weak tag. If none passed, then it could be SW configuration such as protocol value, antenna, and so forth, or a placement configuration like a tag location.

FAULT_NO_PROTOCOL_DEFINED – 401h

Cause

A command was received to perform a protocol command but no protocol was initially set. The reader powers up with no protocols set.

Solution

A protocol must be set before the reader can begin RF operations.

FAULT_INVALID_PROTOCOL_SPECIFIED – 402h

Cause

The protocol value was set to a protocol that is not supported with the current version of SW.

Solution

This value is invalid or this version of SW does not support the protocol value. Check the documentation for the correct values for the protocols in use and that you are licensed for it.

FAULT_WRITE_PASSED_LOCK_FAILED – 403h

Cause

During a Write Tag Data for ISO18000-6B or UCODE, if the lock fails, this error is returned. The write command passed but the lock did not. This could be a bad tag.

Solution

Try to write a few other tags and make sure that they are placed in the RF field.

FAULT_PROTOCOL_NO_DATA_READ – 404h

Cause

A command was sent but did not succeed.

Solution

The tag used has failed or does not have the correct CRC. Try to read a few other tags to check the HW/SW configuration.

FAULT_AFE_NOT_ON – 405h

Cause

A command was received for an operation, like read or write, but the AFE was in the off state.

Solution

Make sure the region and tag protocol have been set to supported values.

FAULT_PROTOCOL_WRITE_FAILED – 406h

Cause

An attempt to modify the contents of a tag failed. There are many reasons for failure.

Solution

Check that the tag is good and try another operation on a few more tags.

FAULT_NOT_IMPLEMENTED_FOR_THIS_PROTOCOL – 407h

Cause

A command was received which is not supported by a protocol.

Solution

Check the documentation for the supported commands and protocols.

FAULT_PROTOCOL_INVALID_WRITE_DATA – 408h

Cause

An ID write was attempted with an unsupported/incorrect ID length.

Solution

Verify the Tag ID length being written.

FAULT_PROTOCOL_INVALID_ADDRESS – 409h

Cause

A command was received attempting to access an invalid address in the tag data address space.

Solution

Make sure that the address specified is within the scope of the tag data address space and available for the specific operation. The protocol specifications contain information about the supported addresses.

FAULT_GENERAL_TAG_ERROR – 40Ah

Cause

This error is used by the GEN2 module. This fault can occur if the read, write, lock, or kill command fails. This error can be internal or functional.

Solution

Make a note of the operations you were performing and contact ThingMagic at http://support.thingmagic.com

FAULT_DATA_TOO_LARGE – 40Bh

Cause

A command was received to Read Tag Data with a data value larger than expected or it is not the correct size.

Solution

Check the size of the data value in the message sent to the reader.

FAULT_PROTOCOL_INVALID_KILL_PASSWORD – 40Ch

Cause

An incorrect kill password was received as part of the Kill command.

Solution

Check the password.

FAULT_PROTOCOL_KILL_FAILED - 40Eh

Cause

Attempt to kill a tag failed for an unknown reason

Solution

Check tag is in RF field and the kill password.

FAULT_PROTOCOL_BIT_DECODING_FAILED - 40Fh

Cause

Attempt to operate on a tag with an EPC length greater than the Maximum EPC length setting.

Solution

Check the EPC length being written.

FAULT_PROTOCOL_INVALID_EPC – 410h

Cause

This error is used by the GEN2 module indicating an invalid EPC value has been specified for an operation. This fault can occur if the read, write, lock, or kill command fails.

Solution

Check the EPC value that is being passed in the command resulting in this error.

FAULT_PROTOCOL_INVALID_NUM_DATA – 411h

Cause

This error is used by the GEN2 module indicating invalid data has been specified for an operation. This fault can occur if the read, write, lock, or kill command fails.

Solution

Check the data that is being passed in the command resulting in this error.

FAULT_GEN2 PROTOCOL_OTHER_ERROR - 420h

Cause

This is an error returned by Gen2 tags. Its a catch-all for error not covered by other codes.

Solution

Check the data that is being passed in the command resulting in this error. Try with a different tag.

FAULT_GEN2_PROTOCOL_MEMORY_OVERRUN_BAD_PC - 423h

Cause

This is an error returned by Gen2 tags. The specified memory location does not exist or the PC value is not supported by the Tag.

Solution

Check the data that is being written and where its being written to in the command resulting in this error.

FAULT_GEN2 PROTOCOL_MEMORY_LOCKED - 424h

Cause

This is an error returned by Gen2 tags.The specified memory location is locked and/or permalocked and is either not writable or not readable.

Solution

Check the data that is being written and where its being written to in the command resulting in this error. Check the access password being sent.

FAULT_GEN2 PROTOCOL_INSUFFICIENT_POWER - 42Bh

Cause

This is an error returned by Gen2 tags. The tag has insufficient power to perform the memory-write operation.

Solution

Try moving the tag closer to the antenna. Try with a different tag.

FAULT_GEN2 PROTOCOL_NON_SPECIFIC_ERROR - 42Fh

Cause

This is an error returned by Gen2 tags. The tag does not support error specific codes.

Solution

Check the data that is being written and where its being written to in the command resulting in this error. Try with a different tag.

FAULT_GEN2 PROTOCOL_UNKNOWN_ERROR - 430h

Cause

This is an error returned by M6e when no more error information is available about why the operation failed.

Solution

Check the data that is being written and where its being written to in the command resulting in this error. Try with a different tag.

Analog Hardware Abstraction Layer Faults

FAULT_AHAL_INVALID_FREQ – 500h

Cause

A command was received to set a frequency outside the specified range.

Solution

Check the values you are trying to set and be sure that they fall within the range of the set region of operation.

FAULT_AHAL_CHANNEL_OCCUPIED – 501h

Cause

With LBT enabled an attempt was made to set the frequency to an occupied channel.

Solution

Try a different channel. If supported by the region of operation turn LBT off.

FAULT_AHAL_TRANSMITTER_ON – 502h

Cause

Checking antenna status while CW is on is not allowed.

Solution

Do not perform antenna checking when CW is turned on.

FAULT_ANTENNA_NOT_CONNECTED – 503h

Cause

An attempt was made to transmit on an antenna which did not pass the antenna detection when antenna detection was turned on.

Solution

Connect a detectable antenna (antenna must have some DC resistance).

FAULT_TEMPERATURE_EXCEED_LIMITS – 504h

Cause

The module has exceeded the maximum or minimum operating temperature and will not allow an RF operation until it is back in range.

Solution

Take steps to resolve thermal issues with module:

- Reduce duty cycle
- Add heat sink
- Use Power Save Mode (non-DRM Compliant)

FAULT_POOR_RETURN_LOSS – 505h

Cause

The module has detected a poor return loss and has ended RF operation to avoid module damage.

Solution

Take steps to resolve high return loss on receiver:

- Make sure antenna VSWR is within module specifications
- Make sure antennas are correctly attached before transmitting
- Check environment to ensure no occurrences of high signal reflection back at antennas.

FAULT_AHAL_INVALID_ANTENA_CONFIG – 507h

Cause

An attempt to set an antenna configuration that is not valid.

Solution

Use the correct antenna setting or change the reader configuration.

Tag ID Buffer Faults

The following table lists the common faults discussed in this section.

Fault Message	Code
FAULT_TAG_ID_BUFFER_NOT_ENOUGH_TAGS_AVAILABLE – 600h	600h
FAULT_TAG_ID_BUFFER_FULL – 601h	601h
FAULT_TAG_ID_BUFFER_REPEATED_TAG_ID – 602h	602h
FAULT_TAG_ID_BUFFER_NUM_TAG_TOO_LARGE – 603h	603h

FAULT_TAG_ID_BUFFER_NOT_ENOUGH_TAGS_AVAILABLE – 600h

Cause

A command was received to get a certain number of tag ids from the tag id buffer. The reader contains less tag ids stored in its tag id buffer than the number the host is sending.

Solution

Send a testcase reproducing the behavior to support@thingmagic.com.

FAULT_TAG_ID_BUFFER_FULL – 601h

Cause

The tag id buffer is full.

Solution

Send a testcase reproducing the behavior to support@thingmagic.com.

FAULT_TAG_ID_BUFFER_REPEATED_TAG_ID – 602h

Cause

The module has an internal error. One of the protocols is trying to add an existing TagID to the buffer.

Solution

Send a testcase reproducing the behavior to support@thingmagic.com.

FAULT_TAG_ID_BUFFER_NUM_TAG_TOO_LARGE – 603h

Cause

The module received a request to retrieve more tags than is supported by the current version of the software.

Solution

Send a testcase reproducing the behavior to support@thingmagic.com.

System Errors

FAULT_SYSTEM_UNKNOWN_ERROR – 7F00h

Cause

The error is internal.

Solution

Send a testcase reproducing the behavior to support@thingmagic.com.

FAULT_TM_ASSERT_FAILED – 7F01h

Cause

An unexpected Internal Error has occurred.

Solution

The error will cause the module to switch back to Bootloader mode. When this occurs make note of the operations you were executing, save FULL error response and send a testcase reproducing the behavior to support@thingmagic.com.

Appendix B: Getting Started - Devkit

Devkit USB Interfaces

USB/RS232

The USB interface (connector labeled USB/ RS232) closest to the power plug is to the RS232 interface of the M6e through an FTDI USB to serial converter. The drivers for it are available at

http://www.ftdichip.com/Drivers/VCP.htm

Please follow the instructions in the installation guide appropriate for your operating system.

Native USB

To use the M6e native USB interface (connector labeled USB), if on Windows, a few installation steps are required for Windows to recognize the M6e and properly configure the communications protocol. In order to use the USB interface with Windows you must have the m6eultra.inf file (included in the M6e alpha package sent). The installation steps are:

1. Plug in the USB cable to the M6e (devkit) and PC.
2. Windows should report is has “Found New Hardware - Mercury6eUltra” and open the Hardware Installation Wizard.
3. Select the Install from a list or specific location (Advanced) option, click Next.
4. Select Don’t search..., click Next, then Next again.
5. Click Have Disk and navigate to where the m6ultra.inf file is stored and select it, click Open, then OK.

6. Mercury6eUltra” should now be shown under the Model list. Select it and click Next then Finished.

Note

The M6e driver file has not been Microsoft certified so compatibility warnings will be displayed. These can be ignored and clicked through.

7. A COM port should now be assigned to the M6e. If you aren’t sure what COM port is assigned you can find it using the Windows Device Manager:

   a. Open the Device Manager (located in Control Panel l System).

   b. Select the Hardware tab and click Device Manager.

   c. Select View l Devices by Type l Ports (COM & LPT) The device appears as Mercury6eUltra (COM#).

Demo Application

A demo application which supports multi-protocol reading and writing is provided in the MercuryAPI SDK package. The source code for this example is included in the MercuryAPI SDK package under /cs/samples/M6e-Read-Write-Demo-Tool. See the *MercuryAPI Programming Guide* for details on using the MercuryAPI.

Demo Tool Notes

- The region is only changed upon initialization. You must disconnect the reader, change the region, and then “Initialize Reader” to change this value.
- The protocol search display is only updated when one of the “Read” buttons is pressed, not when the choice is made via the pull-down menu
- “Read on all connected antennas” automatically activates antenna detection and will ignore ports with an undetectable antenna. To read on these ports, the antenna port must be explicitly selected.
- The “Total tags read in x seconds” display only works for the “Read Once” function, not Start/Stop Reads.
- When “Start Reads” is clicked the M6e will read for the specified timeout with a 2 second delay (RF off) between reads. It is not a continuous read.