In-Car Trigger
User Guide

V1.01
2017/01/16

Prepare by Frost Lin

COBAN Asia Technologies, Inc. | Firmware Engineer
Office: 02-2760-1269 #23 | Fax: 02-2760-1279
Email: frostl@cobantech.com| www.cobantech.com |

Revision History

Data	Description
2016/10/12	➢ Add chapter 11 to descript firmware flushing process with firmware flushing fixture.
2016/12/06	➢ Chapter 2. Add this chapter to highlight the version requirement on SW, FW, and HW for this demo guide. ➢ Section 7.3. Update the procedure description on programming the BWC BT MAC address into NFC tag. ➢ Section 8.2.1. Add this section to descript BWC service discovery procedure and the connection bonding between BWC and In-Car Trigger. ➢ Section 8.2.4. Add this section to implement a simple Bluetooth sniffer with smartphone make diagnose to the Bluetooth connection problem between In-Car Trigger and BWC.
2016/12/28	➢ Chapter 7.1. Update the data context and format programmed in the NFC tag. ➢ Chapter 7.3. Update the procedure description on programming the BWC Bluetooth Mac address and User defined data into NFC tag.
2017/01/11	➢ Chapter 5, Update appearance of Test UI program. ➢ Section 6.2, Update all digital output function definition based on V2 In-Car Trigger ➢ Chapter 10, Update push button function definition based on V2 In-Car Trigger. ➢ Chapter 11, Update LEDs indicators behavior based on V2 In-Car Trigger. ➢ Chapter 12, Update context definition of registers table. ➢ Chapter 13, Update firmware flushing process with serial bootloader application.

Table of contents

1 Abstraction

1.1 The Purpose

This document descript the functions specification of Coban In-Car Trigger and demonstrate how to operate these provided functions.

Primary functions of In-Car Trigger could be summarized below:

- Monitor and control the connected digital input/output of vehicle (EX. Ignition, Light bar…).
- Establish wireless connection to Coban Body-Wore Camera (BWC) with Bluetooth LE technology. Periodically collect the status information of BWC and perform remote control according to pre-defined scenario after the connection setup.

1.2 Document organization

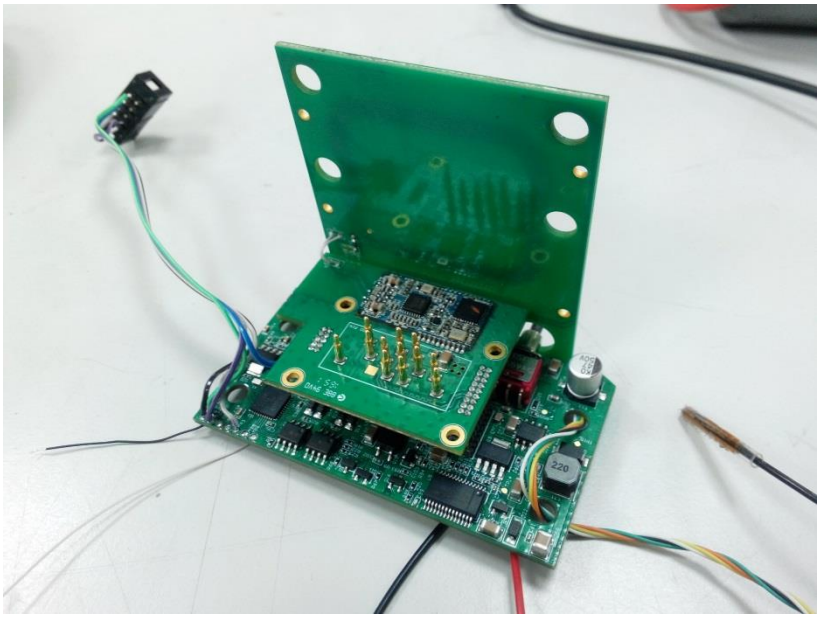
Beside this chapter, this document contains others for the listed topic:

- Chapter 2, Highlight the version requirement on HW, FW, and SW for this In-Car Trigger function demo.
- Chapter 3, overviewing the appearance of In-Car Trigger.
- Chapter 4, introduce how to setup and power on the In-Car Trigger
- Chapter 5, introduce the In-Car Trigger Test UI. A utility program provides an easy way to configure or demonstrate In-Car Trigger functions.
- Chapter 6, introduce connected digital input/output.
- Chapter 7, introduce NFC sensing, includes the information could be retrieved from BWC NFC tag and the action that will be triggered by NFC sensing.
- Chapter 8, introduce the Bluetooth pairing, connection setup, between In-Car Trigger and BWC. Also introduce the BWC remote polling and control function provided on In-Car Trigger.
- Chapter 9, summarize all buzzer notifying event.
- Chapter 10, introduce all push buttons function.
- Chapter 11, summarize all LEDs notifying event.
- Chapter 12, descript communication UART protocol between In-Car Trigger and connected host system.
- Chapter 13, descript the firmware flushing/update process.

2 Version requirement

2.1 Hardware requirement

The version of In-Car Trigger PCBA is **V2.0** for this function demo guide.



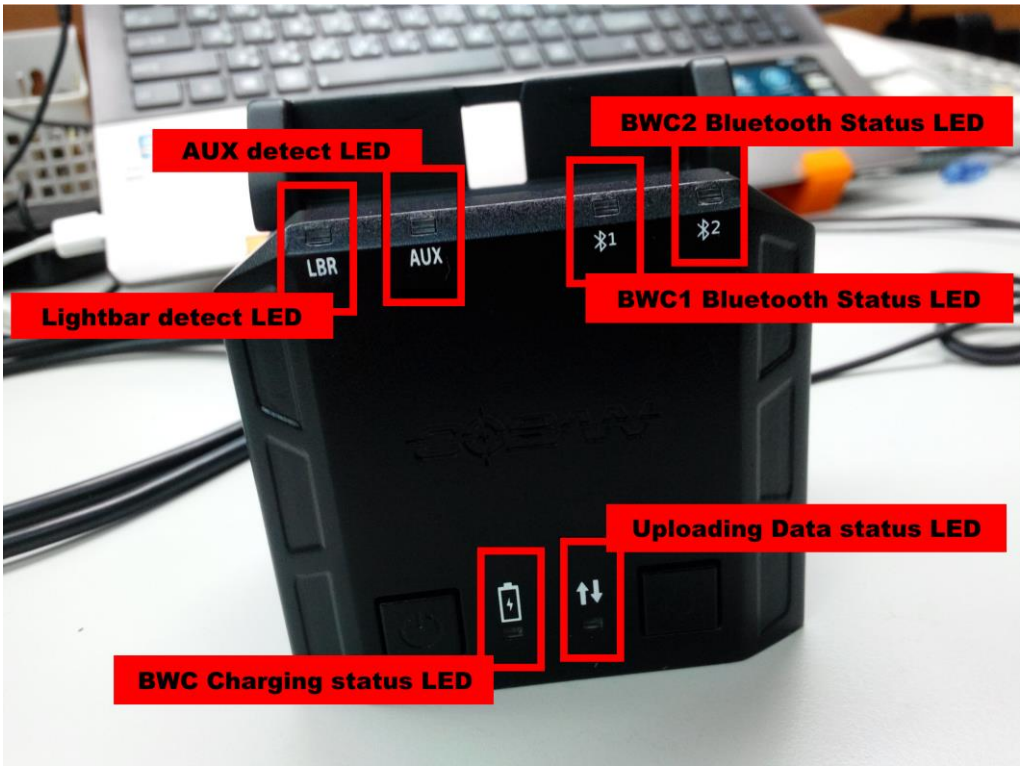
2.2 Software requirement

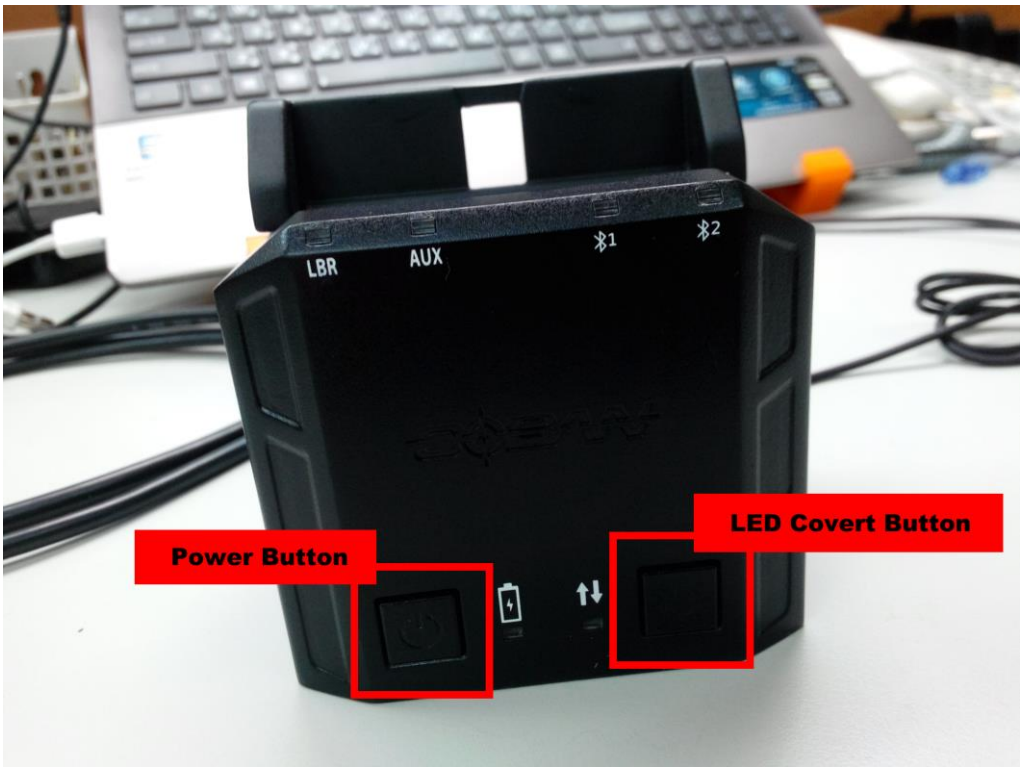
The require version of the In-Car Trigger Test UI software utility (introduced in Chapter 5) must equal or greater than **V01.01**.

2.3 Firmware requirement

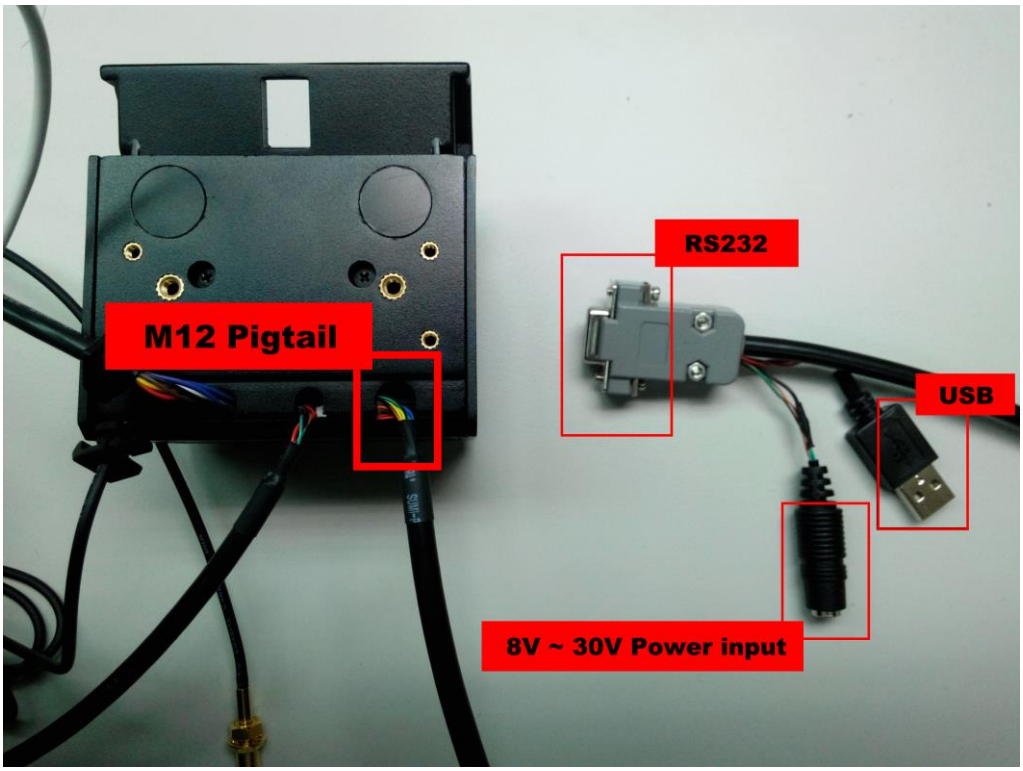
The require version of the In-Car Trigger application firmware must equal or greater than **V01.01**. The way to retrieve In-Car Trigger firmware version is introduced in section 5.2

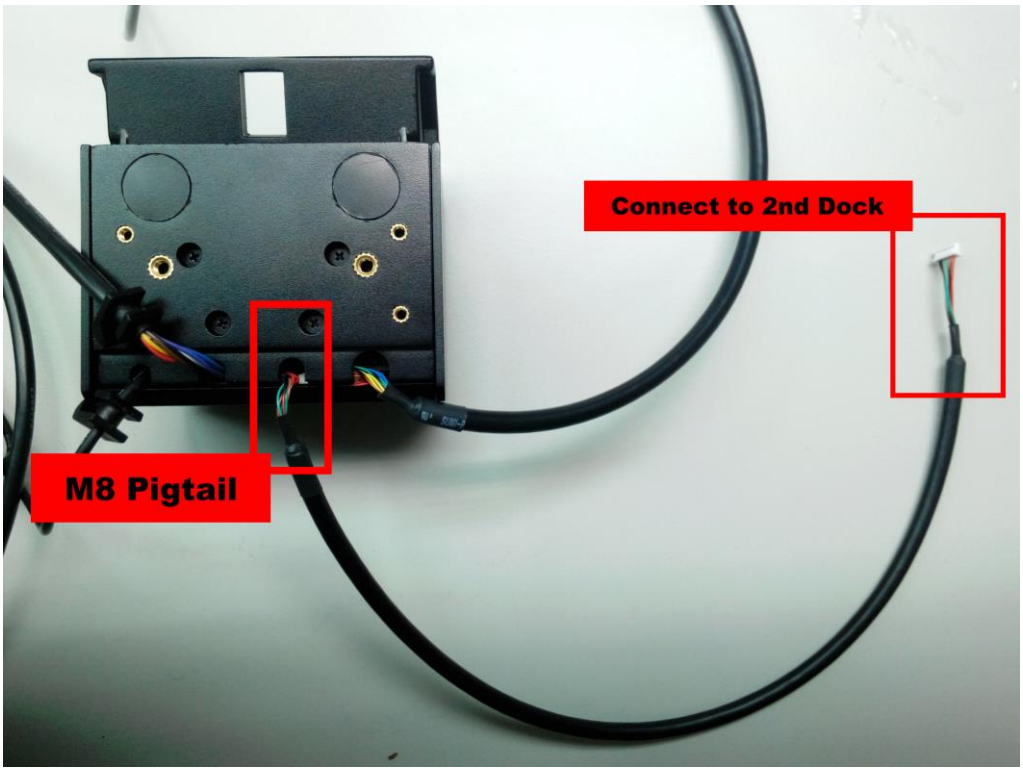
3 Appearance

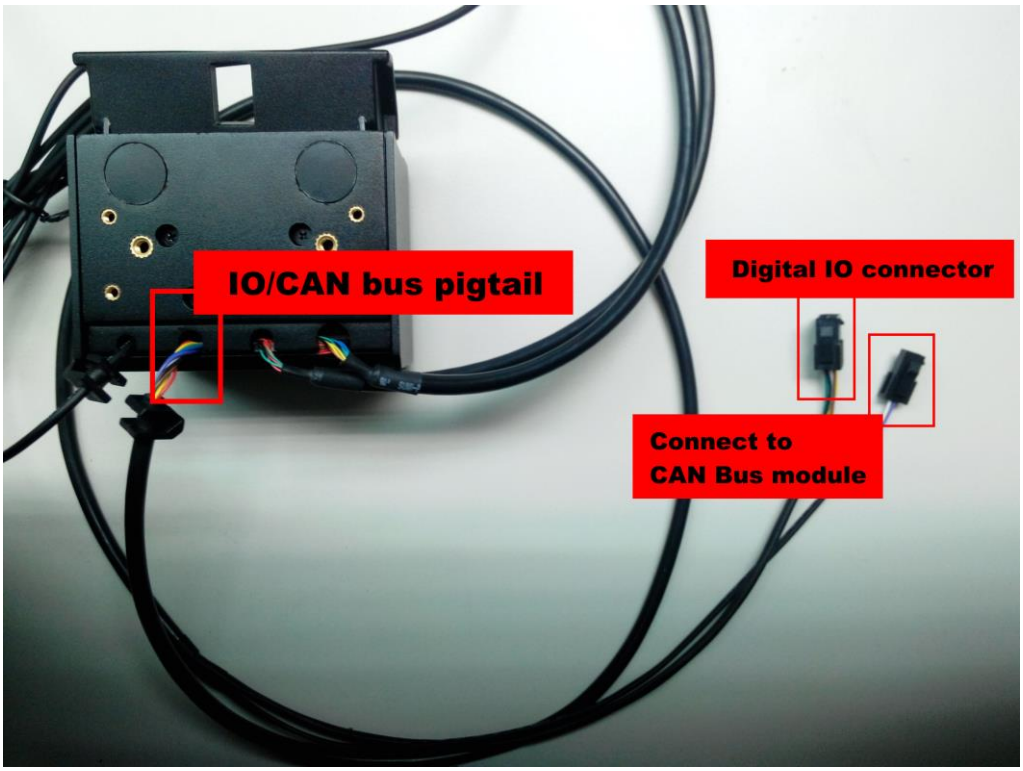



LBR
AUX
Power Button
LED Covert Button

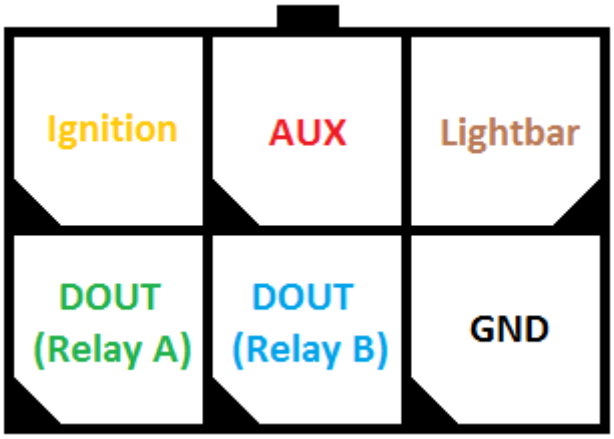

BWC Slot

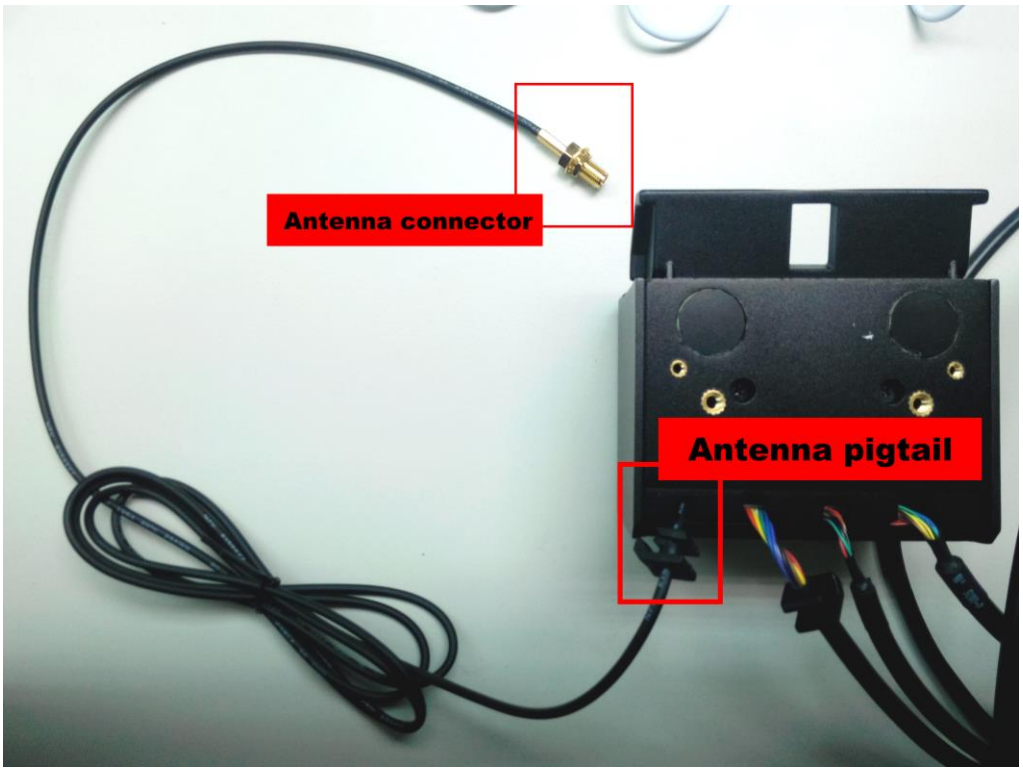

RS232
M12 Pigtail
USB
8V ~ 30V Power input


Connect to 2nd Dock
M8 Pigtail



Digital IO connector pinout:

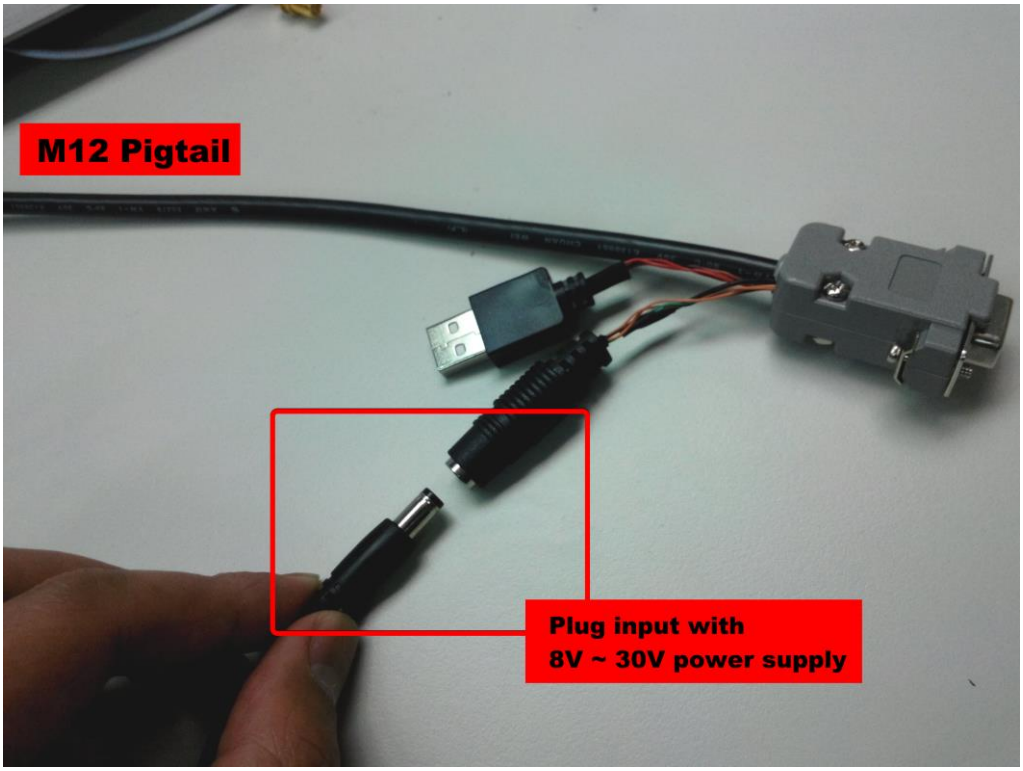



Antenna connector
Antenna pigtail

4 Power On Initialization

4.1 Hardware setup

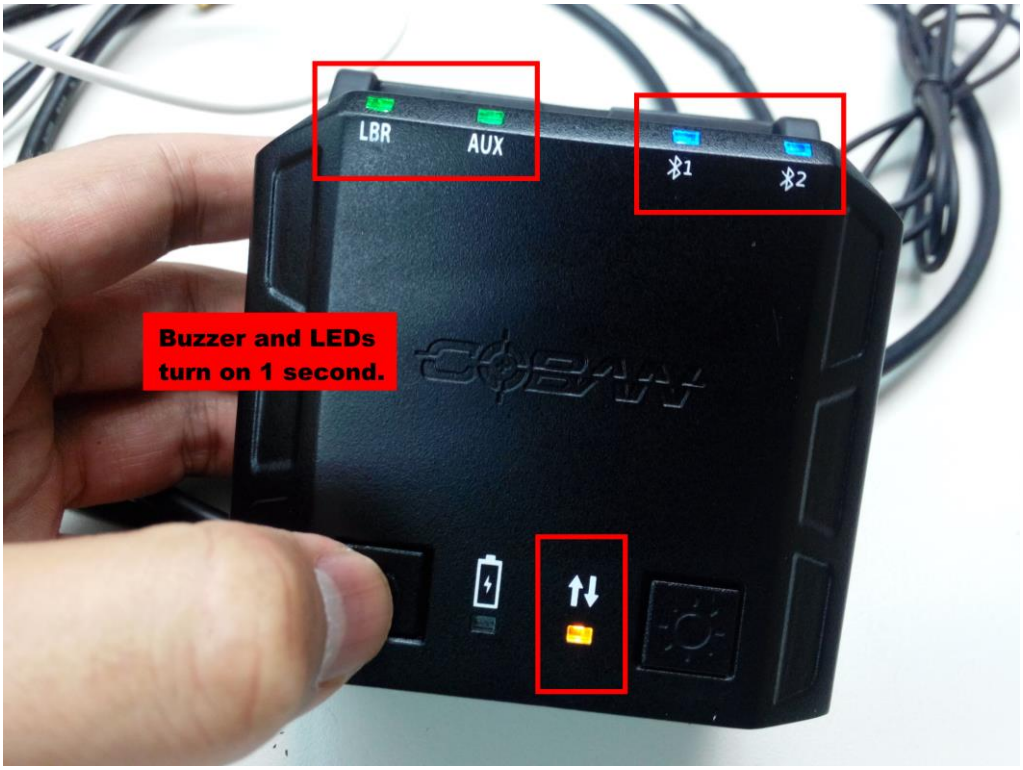
Connect the power input to 8V ~ 30V power supply and click power button to power on the In-Car Trigger.



4.2 Power on notification

When the In-Car Trigger is successfully powered on, it would generate power-on notification:

- Buzzer turn on for 1 sec
- All LEDs turn on for 1 secs, except the BWC Charging status LED remain off.

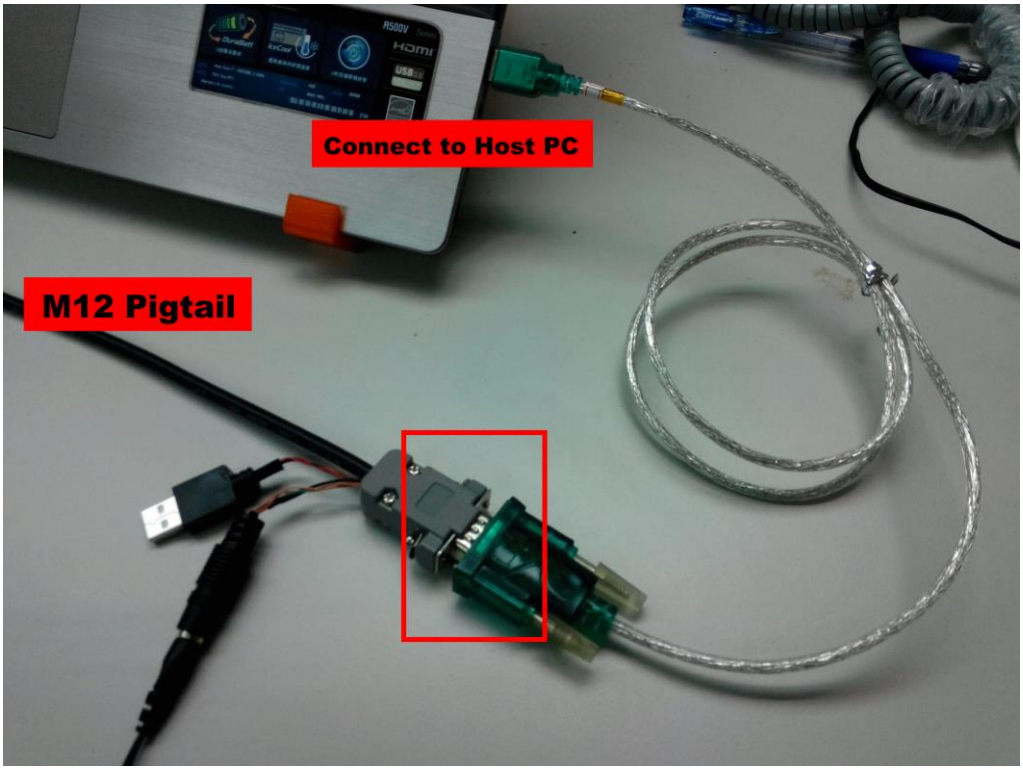


After the notification, buzzer would stop beeping and together with all LEDs start to be controlled based on defined scenario.

5 In-Car Trigger Test UI

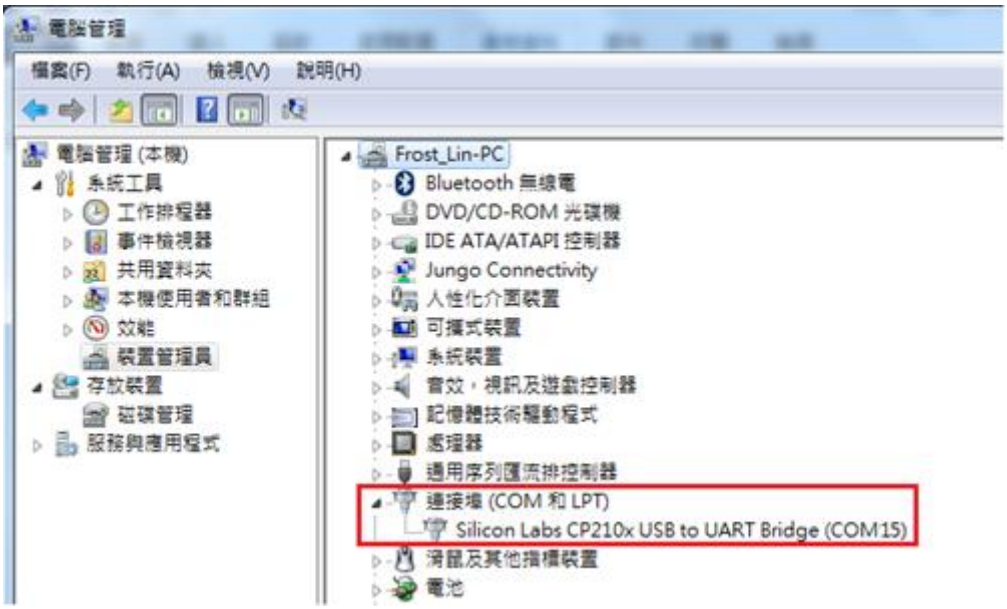
A Win32 software program, In-Car Trigger Test UI, is designed to dump the status of In-Car Trigger via UART interface at run-time.

5.1 Hardware setup

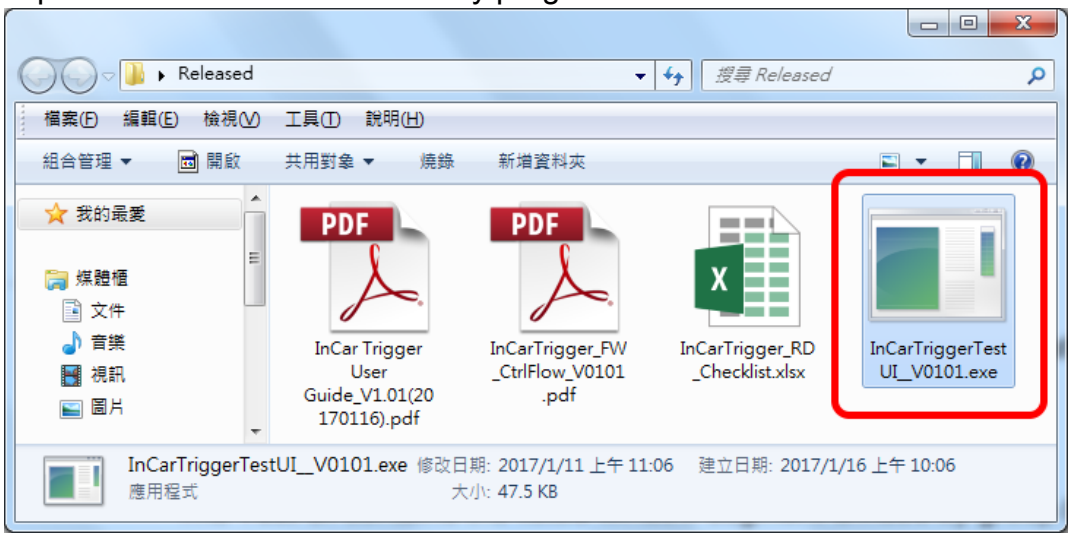


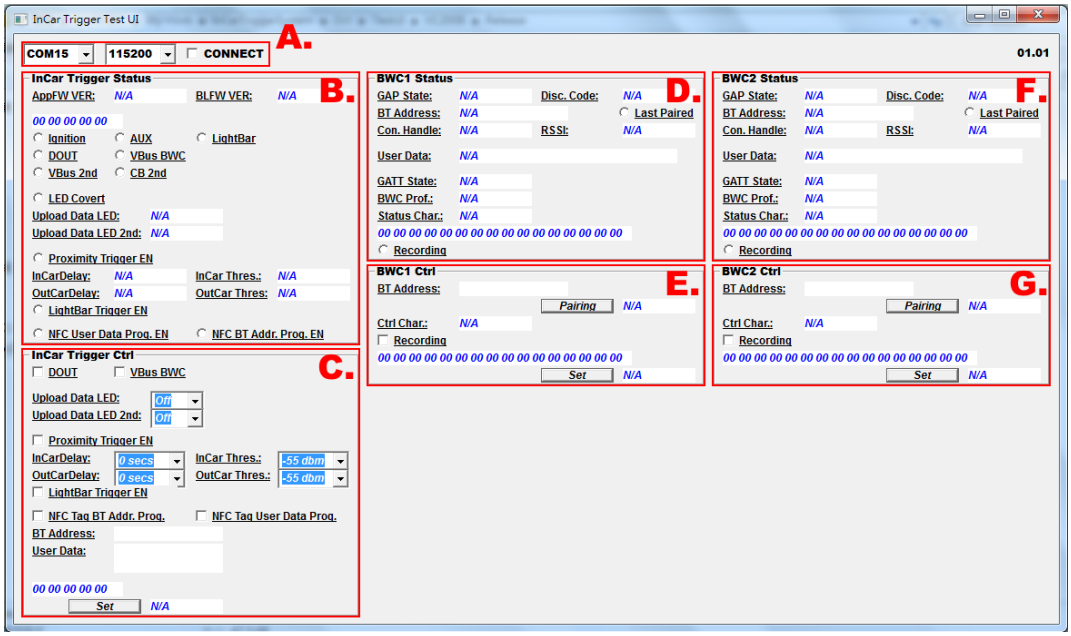
5.2 Software setup

Make sure the COM interface is successfully reorganized in device management of the connected Host PC and identify the its COM port index after setting up the connection descript in section 5.1



Launch the executable of In-Car Trigger Test UI (InCarTriggerTestUI.exe) to bring up the user interface of this utility program.





The Test UI contains the listed window segment divided by group box:

A. **Connection window**
   Includes the UART interface configuration box to setup the serial connection to In-Car Trigger.

B. **In-Car Trigger Status window**
   Display the current status of In-Car Trigger.

C. **In-Car Trigger Control window**
   Includes the icon to configure and control the In-Car Trigger

D. **BWC1 Status window**
   Display the current status of the first connected BWC.

E. **BWC1 Control window**
   Includes the icon to control the first connected BWC.

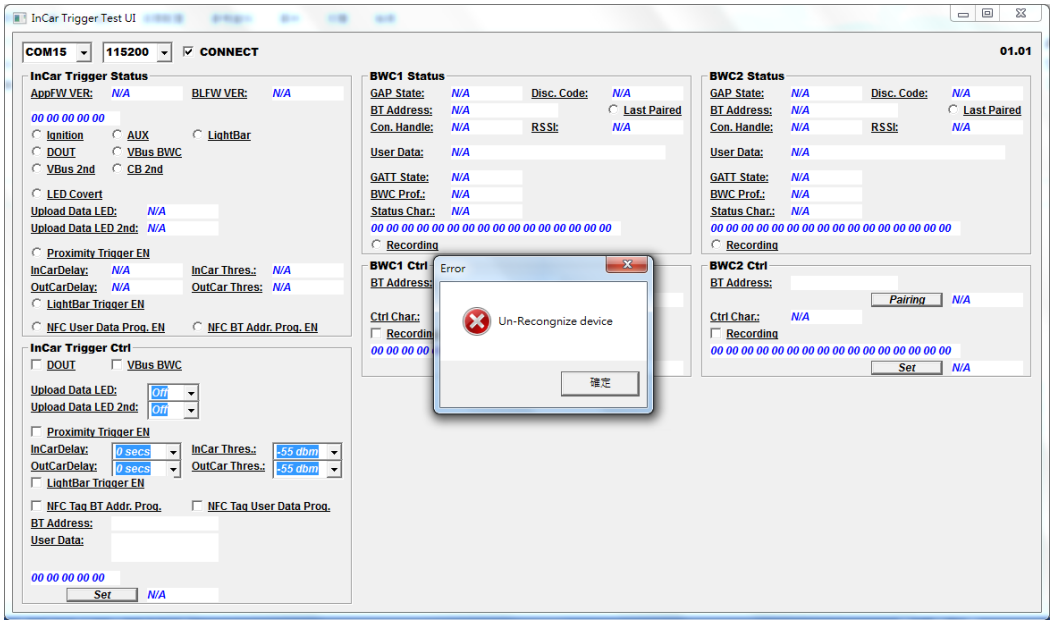
F. **BWC2 Status window**
Display the current status of the second connected BWC.

G. **BWC2 Control window**
Includes the icon to control the second connected BWC.

In-Car Trigger Test UI must establish the UART interface connection to In-Car Trigger before it could control or gather the status from In-Car Trigger. In connection window, select the COM index corresponding to the In-Car Trigger and select the baudrate to 115200 then click connect icon.

Once successfully setup the connection to In-Car Trigger, Test UI will first update the firmware version information of In-Car Trigger in In-Car Trigger Status window.

If the connection to In-Car Trigger is fail to set up, an error window would pop-up to notify the connection failure.



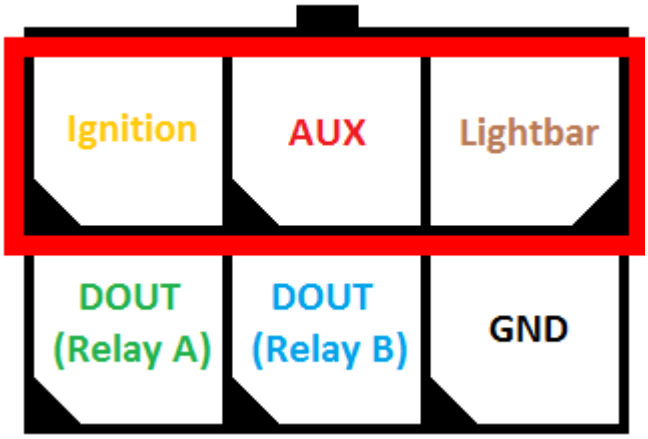
The rest functions provided with In-Car Trigger Test UI would be introduced separately in following chapters together with related In-Car Trigger functionality.

6 Digital IO

6.1 Input

In-Car Trigger provide 3 digital input on Digital IO connector of IO/CAN pigtail to connect and monitor the listed digital signals on patrol car:

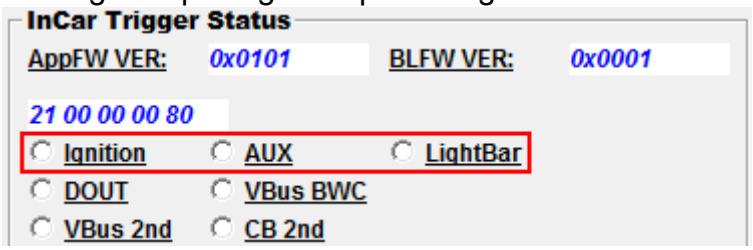
- Ignition
- Auxiliary
- Light bar



All these three digital input could sustain 8V ~ 30V input voltage and controller of In-Car Trigger would determine the digital input as logic high level while apply voltage with this range.

AUX LED and Light bar LED of In-Car Trigger will be turned on if the input voltage level on corresponded digital input is pulled high.

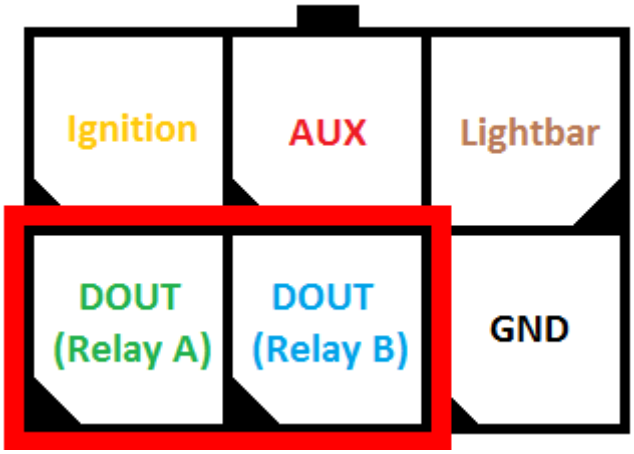
Also, the In-Car Status window on Test UI will keep updating the digital input status on correspond radio icons. Icon will be set if the voltage level on the corresponded digital input signal is pulled high.



6.2 Output

In-Car Trigger provide the listed digital output, which is connected to the BWC on BWC slot, 2nd BWC dock, or peripheral device on patrol car.

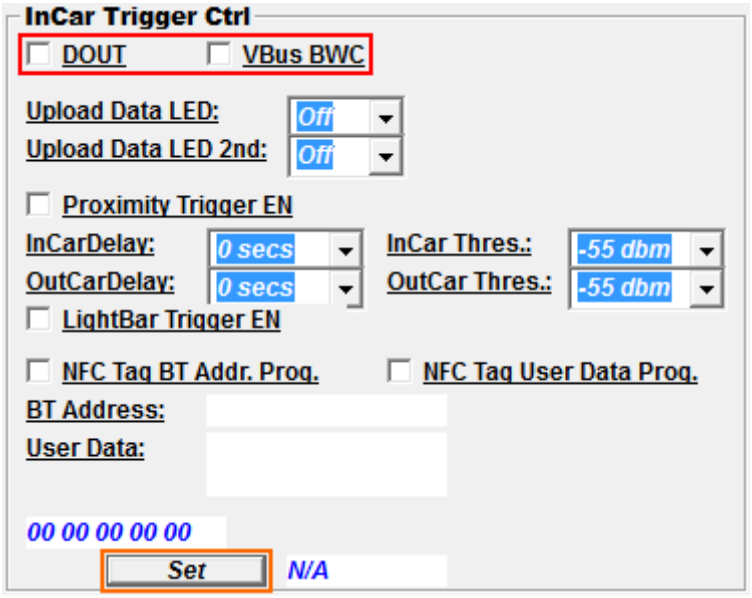
- **DOUT**: Turn on/off (short/open) the connection between the relay's A/B pin on I/O Pigtail



- **VBUS_BWC:** Turn on/off the 5V USB VBus voltage to enable/disable the USB interface implemented on pogo pins inside BWC slot.

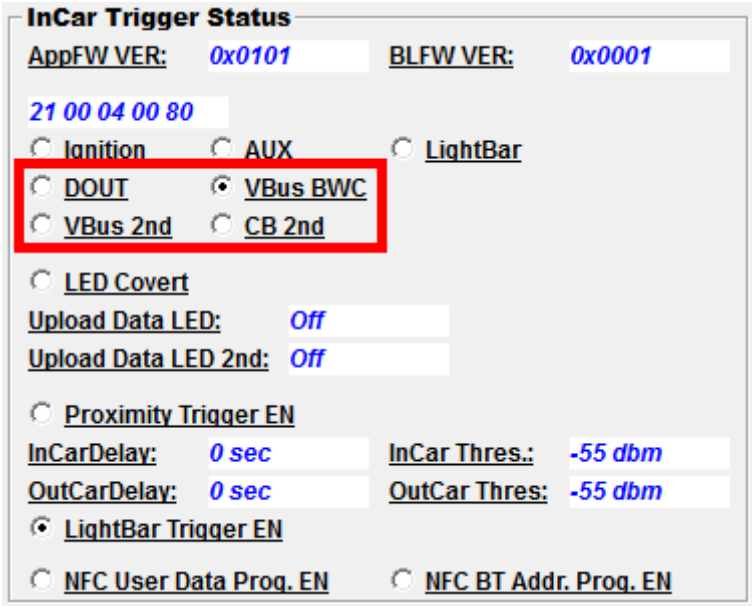
- **Upload Data LED 2nd:** Turn on off the Upload Data LED on connected 2nd Dock, controlling of this output will be descript in section 11.1.5
- **VBUS_2nd:** Turn on/off the 5V USB VBus voltage to enable/disable the USB interface implemented on M8 pigtail connected to 2nd Dock.
- **CB_2nd:** Turn on/off the CB voltage output on M8 pigtail connected to 2nd Dock.

DOUT and **VBus_BWC** is could be directly controlled by the UART connected host device. Test UI also provides checkbox in the In-Car Trigger Ctrl window, Clicking/unclicking these checkbox of corresponded digital output signal and clicking “set” icon to control the output state.



VBus_2nd and **CB_2nd** output is controlled based on the status of ignition input. These two output signal will be automatically pull high if state of ignition input is considered as logic high.

All Digital output status will be periodically updated on In-Car Trigger Status window of In-Car Trigger Test UI.



7 NFC

NFC reader is implemented within In-Car Trigger to retrieve the Bluetooth MAC address and user defined data from the NFC tag implemented within BWC for triggering the Bluetooth pairing between In-Car Trigger and BWC (refer to chapter 8 for detail description about Bluetooth pairing)

The NFC read/write communication between In-Car Trigger and BWC follows **ISO 14443A** protocol.

7.1 NFC Tag information

Storage of Each BWC's NFC tag contains the listed data which is programmed individually at manufacturing:

- **Bluetooth MAC address**
  A 6 bytes length unique MAC address of the Bluetooth module within the BWC. When performing NFC sensing with In-Car Trigger, it will be retrieved by In-Car Trigger to initiate Bluetooth pairing.

  This 6 bytes address is programmed in specific offset within the storage of NFC tag and is not wrapped with standard NFC Data Exchange Format (NDEF). NFC reader other than In-Car Trigger might unable to retrieve or parsing this data successfully out from NFC tag of BWC.

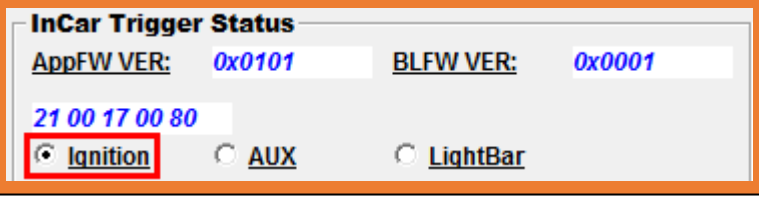
- **User Data**
  A data space is preserved in NFC tag to store a user self-defined max 32 bytes length string. When performing NFC sensing, In-Car Trigger follow the **Plain Text** record format with "**en**" IANA language code specified defined within **NDEF** to parse out this 32 bytes user data from the retrieved raw data and buffer it for the connected host device's usage.

  By following the same NDEF Plain Text record format, NFC reader other than In-Car Trigger also could read/write this user data on BWC's NFC tag.

7.2 NFC sensing

[Note]

The internal power supply of NFC reader is controlled by **Ignition** digital input. Make sure 8V ~ 30V is applied to the Ignition on digital input connector of IO/CAN pigtail before using NFC sensing function.



Approach the sensing area of NFC tag of BWC to the sensing area NFC reader sensing area of In-Car. This behavior will make NFC reader start to retrieve information programmed in NFC tag.

If NFC reader successfully receive the programmed information on NFC tag, pairing notification on buzzer will be triggered and Bluetooth pairing procedure will start which will be introduced in section 8.1. The retrieved tag information will be updated to the BWC Status window of Test UI.

If NFC reader fail to parse the retrieve data from NFC tag (EX. the programmed user data doesn’t following NDEF Plain Text format), The Buzzer would generate 1 long beeping to notify user sensing fail.

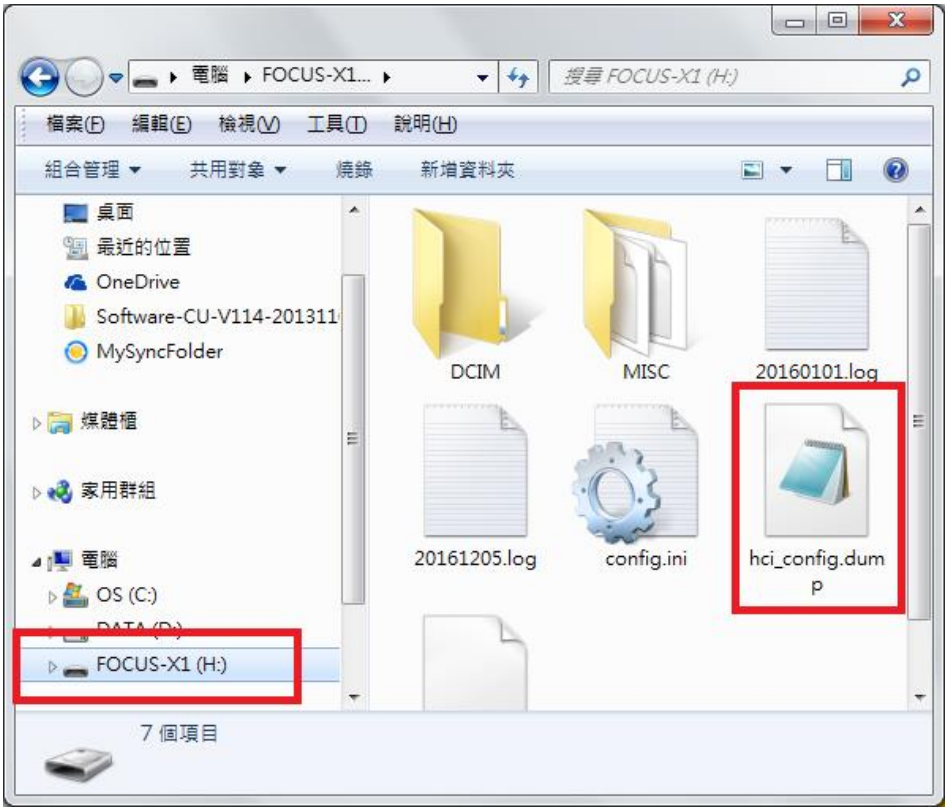
7.3 NFC tag programming

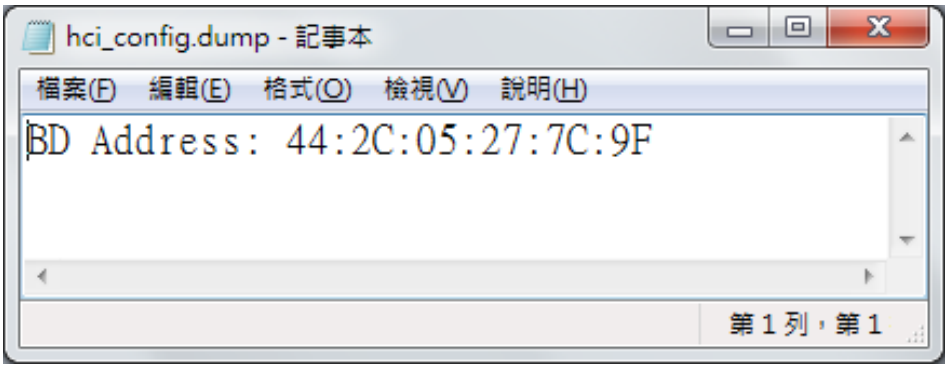
Besides retrieve the data from NFC tag, the NFC reader of In-Car Trigger could be used to program the Bluetooth Mac address and user data into the BWC's NFC tag.

7.3.1 Bluetooth Mac address programming

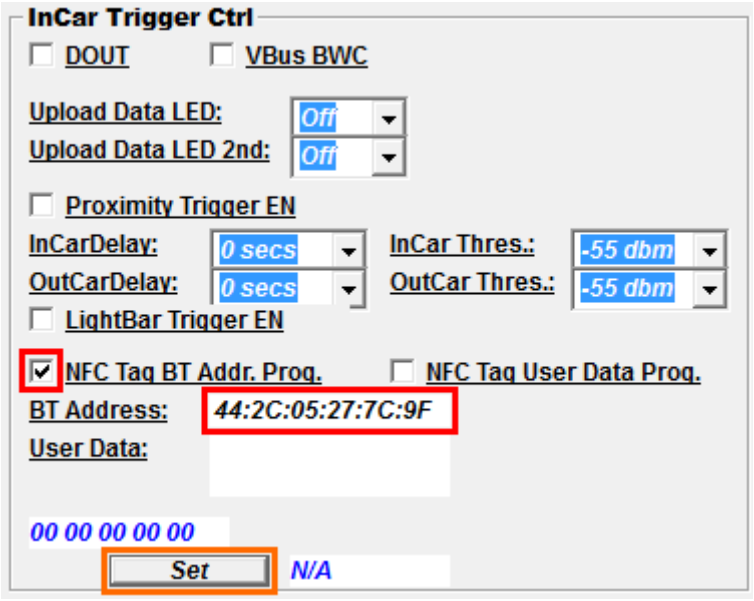
The unique BT address of each BWC could be looked up in the TF card storage of BWC. Connect the BWC to a PC with USB cable and wait for the PC identifying the **FOUCS-X1** storage in files manager.

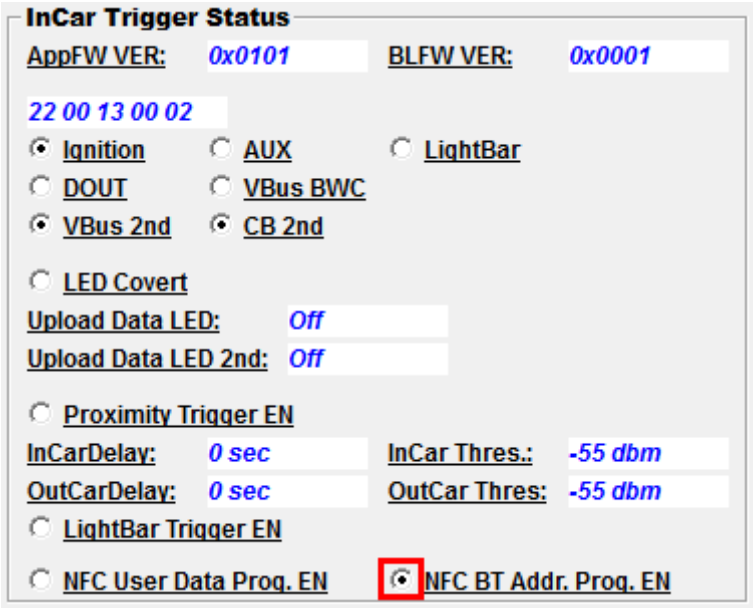
Opening the **hci_config.dump** with text editor under FOUCS-X1 storage could get the unique BT address of this BWC.





In BWC Ctrl window, click the **NFC Tag BT Addr. Prog.** icon, assign value into **BT Address** column, and click "***Set***" button. Wait for the radio button indicator **NFC Tag Prog. EN** and **BT addr. Prog. EN** being set by In-Car Trigger fixture itself to switch the NFC reader to programming mode. The following diagram example set the NFC reader the program the NFC tag with the listed info:

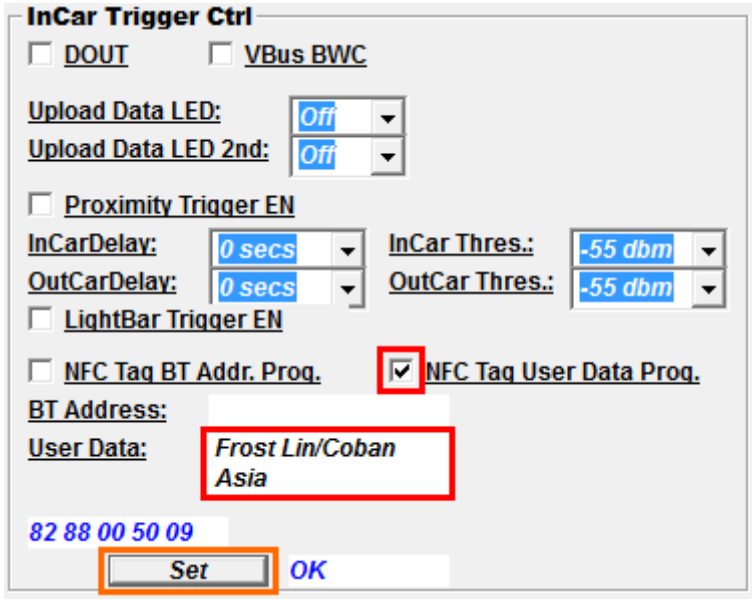


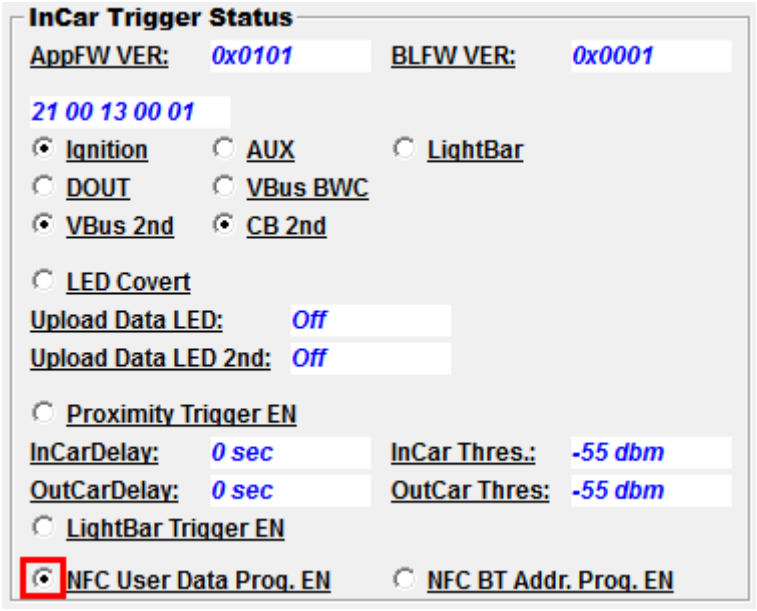


Approach the BWC NFC tag to the sensing area of the NFC reader on In-Car Trigger fixture like section 7.2. The buzzer on fixture would generate 3 short beeps to notify programming complete.

7.3.2 User Data programming

In BWC Ctrl window, click the **NFC Tag User Data Prog.**, fill max 32 characters in **User Data** column, and click "***Set***" button. Wait for the radio button indicator **NFC User Data Prog. EN** being set itself to switch the NFC reader to programming mode.

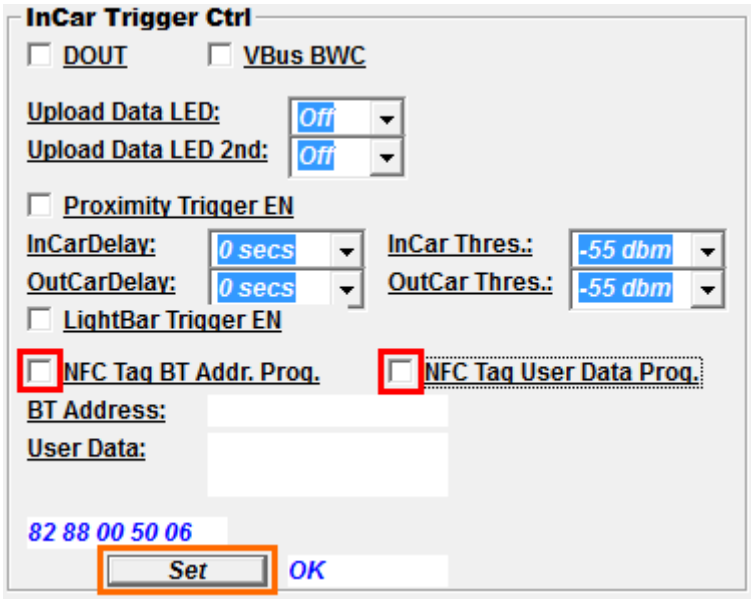


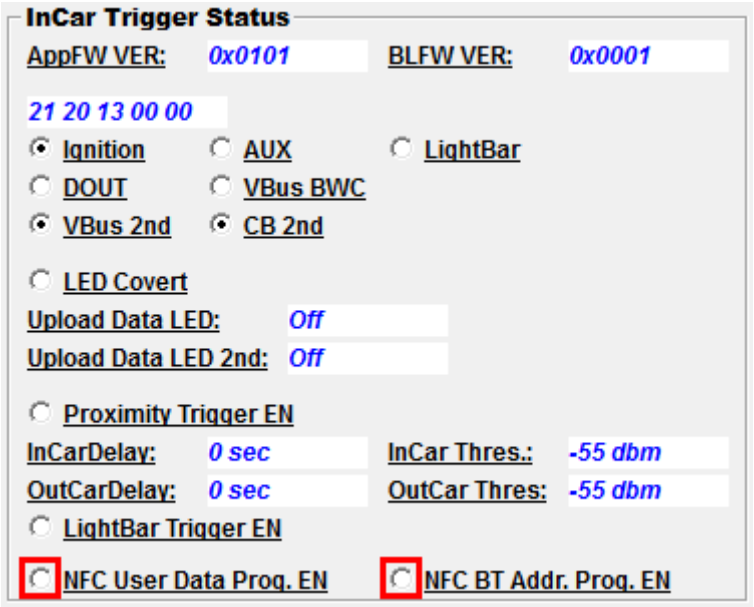


Approach the BWC NFC tag to the sensing area of the NFC reader on In-Car Trigger fixture like section 7.2. The buzzer on fixture would generate 3 short beeps to notify programming complete.

7.3.3 Resume NFC reader to reading mode

Un-click the **NFC Tag User Data Prog.** and **NFC Tag BT Addr. Prog.**, click Set icon and wait for **NFC User Data Prog. EN** and **NFC BT Addr. Prog. EN** being un-set to bring the NFC reader out of programming mode.





8 Bluetooth

In-Car Trigger and BWC adopt Bluetooth Low Energy (BLE or Bluetooth 4.0) as the wireless communication interface between each other. Comparing with Bluetooth 2.1 technology, BLE has the listed features:

- Ultra-low power consumption
  The BLE maintain the connection between two paired device via a periodically handshake mechanism instead of sustaining the link continuously, it save the power to diving the RF component enormously to raise the operating time of BWC.
- Low data rate
  As a side effect to the periodically handshaking, data transmission between 2 paired devices is not streaming but only small amount periodically. BLE is only suitable for transferring self-defined remote command but not used to streaming the video or audio.

8.1 Pairing decision

In-Car Trigger and BWC must perform Bluetooth pairing to setup the Bluetooth connection, which is trigged by NFC sensing introduced in section 7.2

In Bluetooth pairing procedure, In-Car Trigger first retrieve BWC's Bluetooth Mac address via its NFC reader from the NFC tag of BWC, and then it decide to setup Bluetooth connection to this BWC base on the defined pairing decision.

In-Car Trigger max support pairing to 2 BWCs with Bluetooth simultaneously, the pairing decision is designed to which one of these Bluetooth channels will be allocated for the new-paired BWC, and which BWC will be dis-paired if required base on the following description of scenario:

A. The Bluetooth mac address retrieved from NFC tag of target BWC will be compared with the mac address currently paired with both of two Bluetooth channels. If the address is match, this corresponded Bluetooth channel will be allocated to pair with the target BWC.

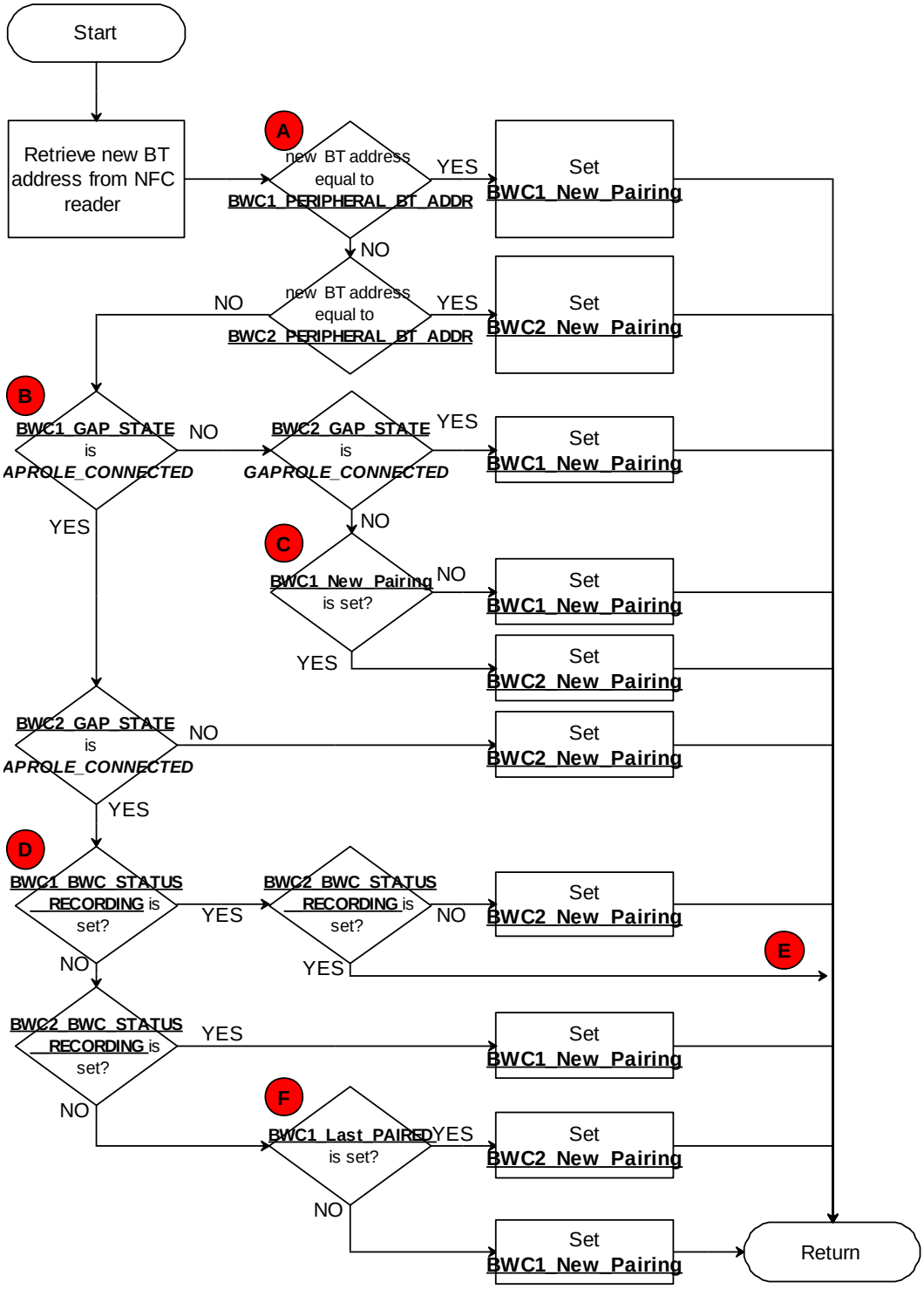
B. Continues from (A). If the mac address mismatch to both of the two Bluetooth channels, the channel which status currently is not connected to its paired device will be allocated to pair with the target BWC.

C. Continues from (B). If both Bluetooth channels' status currently are not connected, the Bluetooth channel which is the last allocated for pairing will be reserved, and the other one will be allocated to pair with the target BWC.

D. Continues from (B). If both Bluetooth channels' status currently are connected, the one that its paired BWC currently is not recoding will be allocated to pair with the target BWC.

E. Continues from (C). If both BWCs currently paired with In-Car Trigger is recoding, the target BWC is not allowed to pair with In-Car Trigger.

F. Continues from (C). If both BWCs currently paired with In-Car Trigger is not recoding, the Bluetooth channel which is the last allocated for pairing will be reserved, and the other one will be allocated to pair with the target BWC.



In the scenario (B), (C), (D), and (F) above, The buffered BWC mac address paired with allocated Bluetooth channel will be overwrite by the one of target BWC, and the original BWC should be re-paired to In-Car Trigger if it want to setup the Bluetooth connection to In-Car Trigger again.

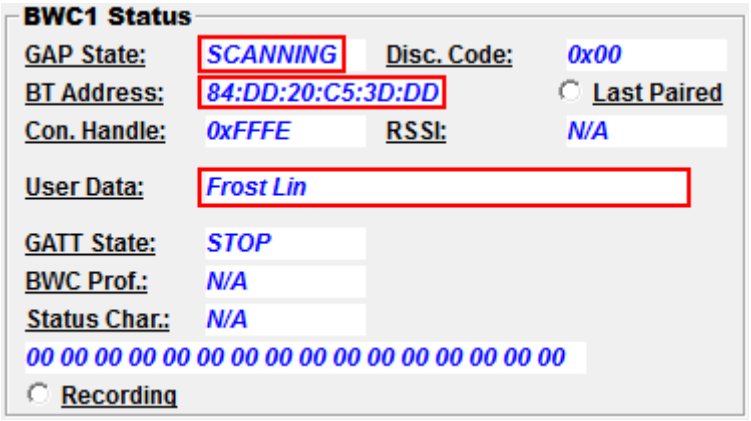
8.1.1 Pairing / Forbidden pairing notification

Once the Bluetooth channel is chosen to be allocated to pairing with target BWC, In-Car Trigger will generate a pairing notification to notify user it is starting to setup the Bluetooth connection to the target BWC.

Pairing notification includes:

- **Buzzer beeping.** Depend on it is Bluetooth channel#1 or #2 allocated to pair with target BWC. Buzzer will generate one or two *short beeping*.
- **BWC Bluetooth Status LED flashing.** Depend on it is Bluetooth channel#1 or #2 allocated to pair with target BWC. BWC1 or BWC2 LED will start flashing until the Bluetooth connection is successfully setup.

After generating the pairing notification, **BT Address** and **User Data** in BWC Status window of Test UI would be updated to the value received form NFC during the sensing. Value of **GAP Status** would switch between “STARTED” and “SCANNING” periodically.



Regarding of case (E) of pairing decision, if status of both Bluetooth channels of In-Car Trigger is currently connected and the both paired BWCs are recording, In-Car Trigger will generate forbidden pairing notification to notify user pairing is not allowed currently.

Forbidden pairing notification includes:

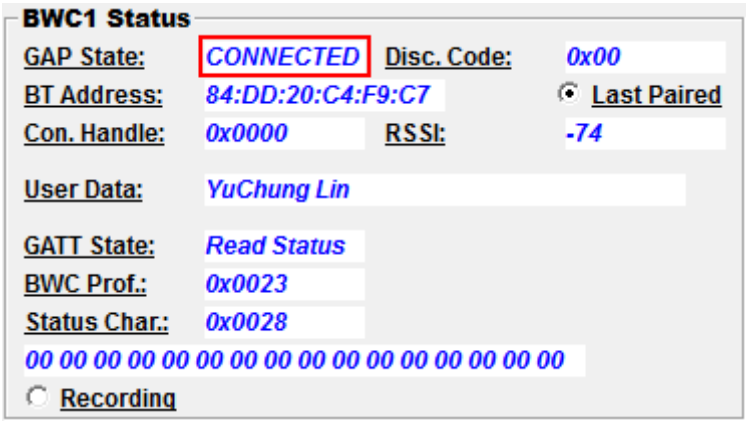
- **Buzzer beeping.** Buzzer will generate one *long beeping*.
- **BWC Bluetooth Status LED un-change.** BWC LEDs will remain the original status.

8.2 Bluetooth link establishing

After In-Car Trigger generate pairing notification, normally it would take less than 4 seconds to set up the Bluetooth link to the target BWC within 2m light-of-sign distance.

Depend on it is Bluetooth channel#1 or #2 allocated to pair with target BWC. BWC1 or BWC2 LED will stop blinking and turn solid on when link is successfully established.

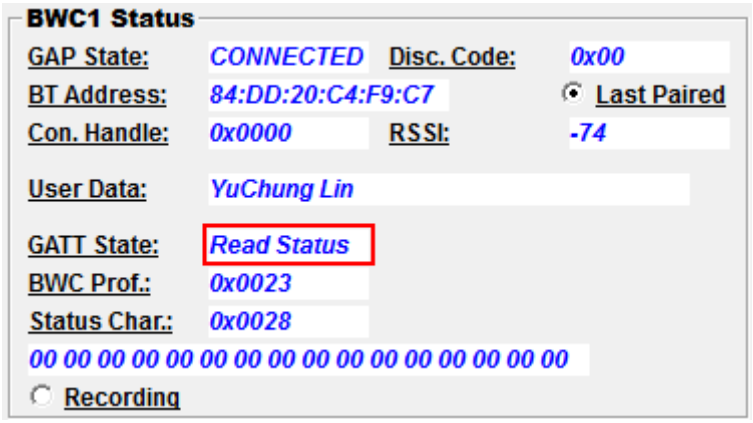
Value of GAP State in BWC Status window of Test UI would be switch to "CONNECTED" when Bluetooth link is established.



8.2.1 Bluetooth service discovery and bonding

After establishing the Bluetooth link, In-Car Trigger would discover the service of connected device and make sure it provide "BWC Service" for identifying it is actually a BWC device.

During the discovery, value of GATT State in BWC Status window of Test UI would update and display the service discovery state until the displayed value keep switch between *"Read Status"* and *"Write Ctrl"* which indicate the service discovery complete.



Once complete service discovery, In-Car Trigger would make bonding to this

BWC by storing its connection info into EEPROM and automatically try to resume the connection at connection lost or power reset until it is paired to another BWC.

8.2.2 Bluetooth transmission distance

Transmission distance between BWC V4 Hardware….(TBD)

8.2.3 Bluetooth link disconnecting

The Bluetooth link between In-Car Trigger and BWCs would be disconnected due to the following reason:

1. Low signal or signal blocking

   Pull the BWC faraway than defined transmission distance or block the radiation signal with metal case would decay the Receive Signal Strength Indicator (RSSI), which will make the Bluetooth link un-stable or even disconnected.

   The link would be re-established when the RSSI raise again due to the distance between In-Car Trigger and BWC getting closer (within TBD m) or signal is un-blocked.

2. New pairing

   Perform Bluetooth pairing on the target BWC with other In-Car Trigger or pairing other BWC to the In-Car Trigger which the target BWC currently paired with would make the original Bluetooth link between target BWC and In-Car Trigger be replaced and destroyed.

   Refer to pairing decision descript in section 8.1 for all possible condition which the link would be replaced by a new pairing. Once the link is replaced, it won't recover itself unless perform Bluetooth pairing on the BWC and In-Car Trigger again.

Depend on it is BWC1 or BWC2 disconnected from In-Car Trigger, BWC1 or BWC2 LED would turn off.

Value of GAP Status would switch from "*CONNECTED*" to between "*STARTED*" and "*SCANNING*" periodically. And Value of Disc. Code would be updated with the code of reason of the last disconnecting.

8.2.4 Connection diagnose

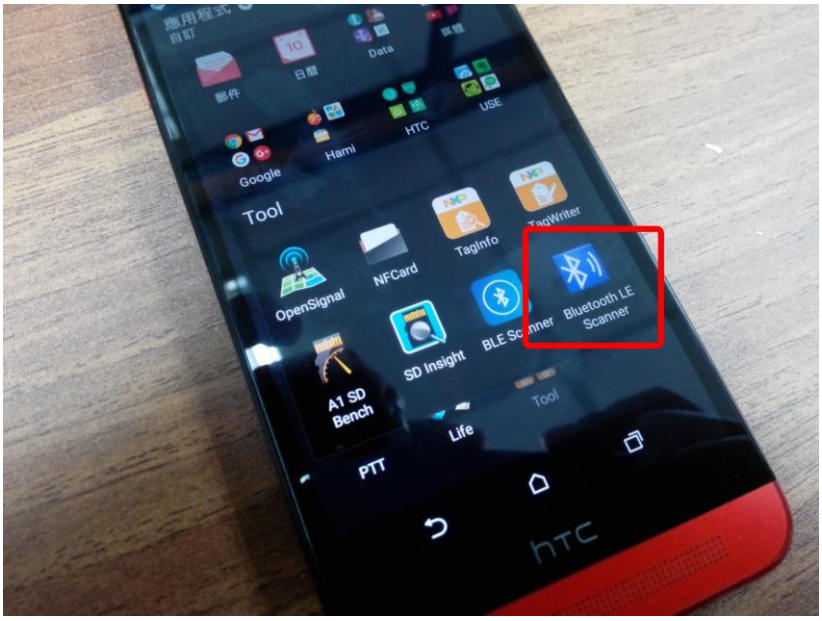
This section descript some methods to diagnose the situation of failing to establish Bluetooth connection between In-Car Trigger and BWC.

1. Scan BWC's advertising signal

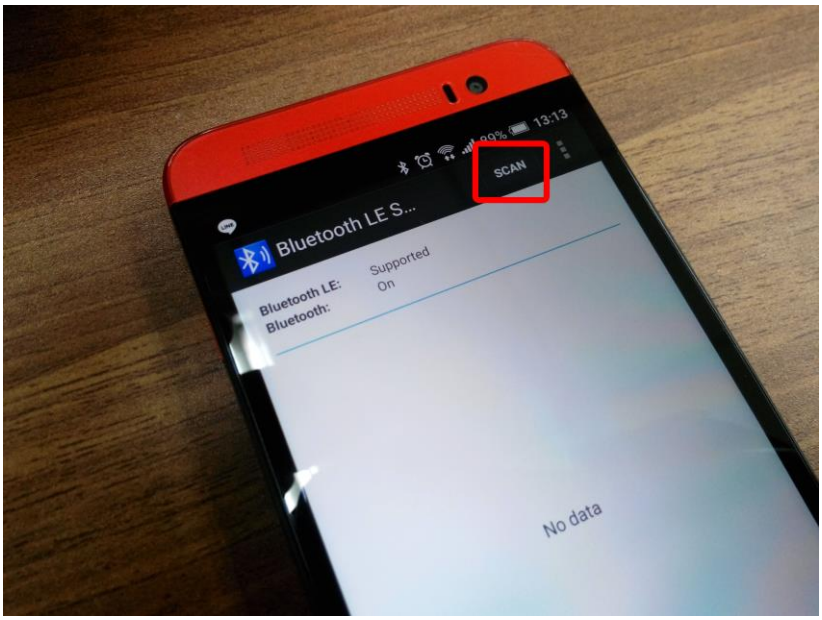
   The first step to make diagnose on failing to establish Bluetooth connection is to check whether the BWC (serve as the Bluetooth Peripheral device) could generate advertising signal for the In-Car Trigger (serve as Bluetooth Central device) to make connection request.

   A **Smartphone** support Bluetooth 4.0, with a free Android App **Bluetooth LE Scanner** or other Bluetooth 4.0 specific App could be served as a simple Bluetooth sniffer.

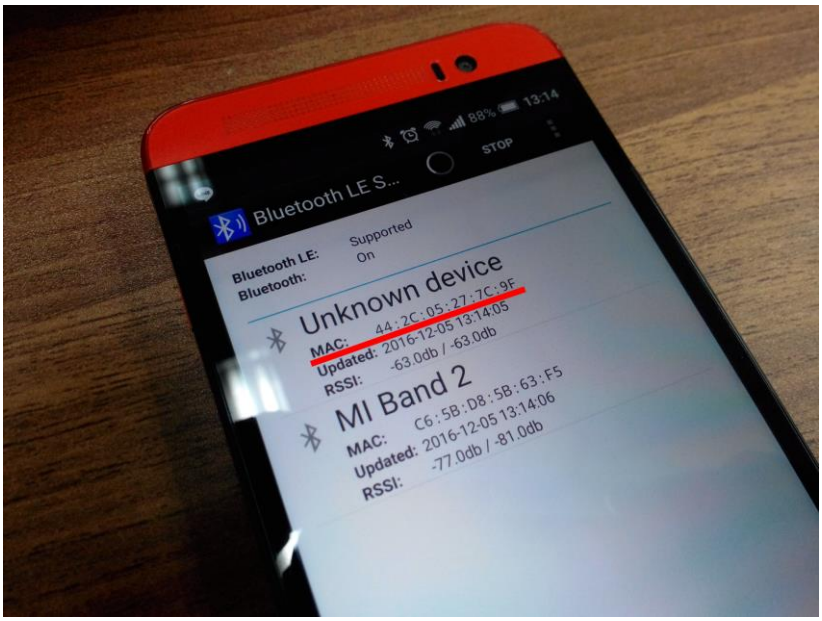
   Launch the Bluetooth LE scanner to bring up the user interface.



   Press "Scan" icon to start scanning all advertising signal generated from Bluetooth peripheral devices nearby.



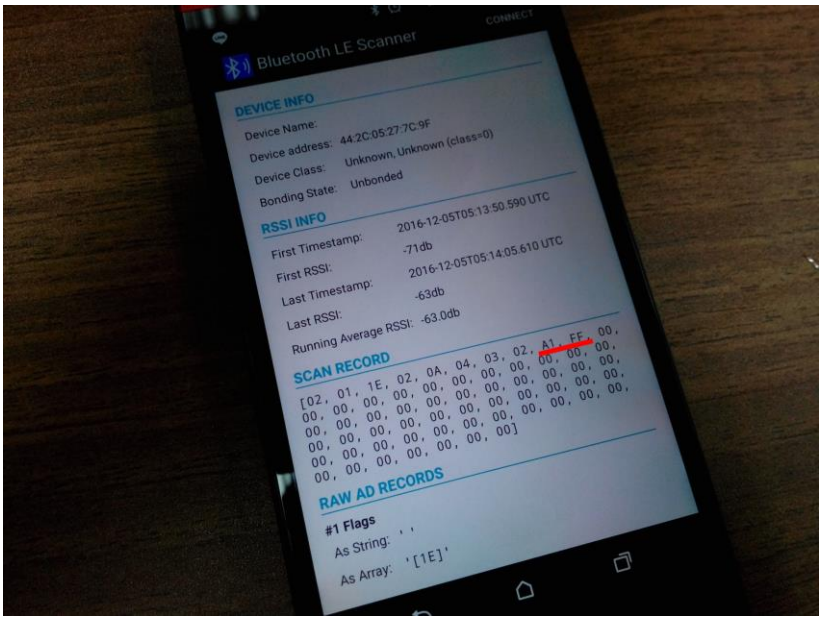
On the display list of found devices, check if the BT address of the target BWC (refer to section 7.3), EX. 44:2C:05:27:7C:9F, is discover or not.



2. **Check the context of advertising**

   If the target BWC could generate advertising signal, the next step is to check the advertising data contain the correct **Service UUID** for In-Car Trigger to make connection request.

   Click the device with correct BT address to display its advertising data, and make sure the service UUID, **FFA1**, is contained in the advertising data.


Bluetooth LE Scanner
CONNECT
DEVICE INFO
Device Name:
Device address: 44:2C:05:27:7C:9F
Device Class: Unknown, Unknown (class=0)
Bonding State: Unbonded
RSSI INFO
First Timestamp: 2016-12-05T05:13:50.590 UTC
First RSSI: -71db
Last Timestamp: 2016-12-05T05:14:05.610 UTC
Last RSSI: -63db
Running Average RSSI: -63.0db
SCAN RECORD
[02, 01, 1E, 02, 0A, 04, 03, 02, A1, FF, 00,
00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,
00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,
00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,
00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,
00, 00, 00, 00, 00, 00, 00]
RAW AD RECORDS
#1 Flags
As String: ' '
As Array: '[1E]'

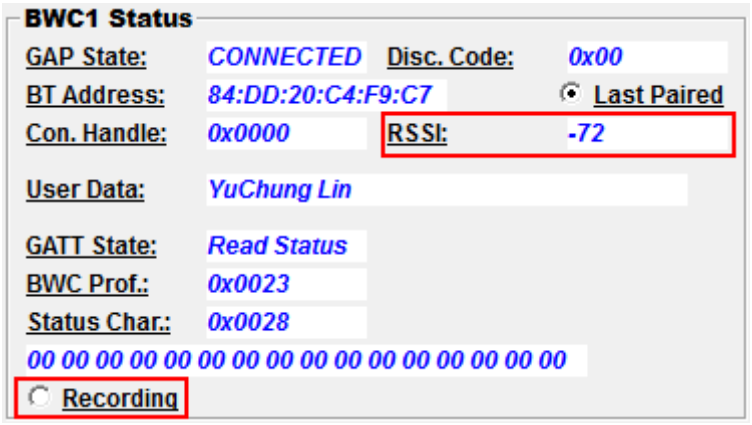
3. (TBD)

8.3 BWC remote polling

Once the Bluetooth link is established between In-Car Trigger and BWC, In-Car Trigger will periodically poll and retrieve the listed BWC status information via Bluetooth:

- **RSSI**: The receive signal strength from paired BWC.
- **Operating status** Indicate the paired BWC is recording or not.

Base on the raw data of retrieved BWC status, Test UI will periodically update the value of BWC status information listed above in BWC Status window. Detail raw data format to parsing the retrieved raw data of BWC status information would be introduced at section 12.2.

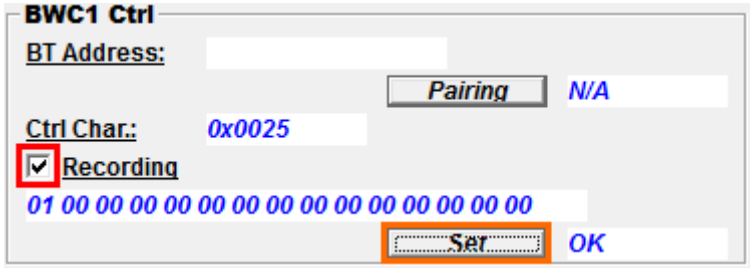


8.4 BWC remote control

Once the Bluetooth link is established between In-Car Trigger and BWC, In-Car Trigger could the listed BWC status via defined UART command through UART interface:

- **Operating status**: Includes turn recording on/off remotely.

Test UI provide checkbox in BWC Control window for user to select the configuration to control BWC's operating status. It will generate raw data UART command and dispatch to In-Car Trigger via UART interface to control BWC remotely. Detail raw data of UART command format to remotely control the BWC would be introduced at section 12.2



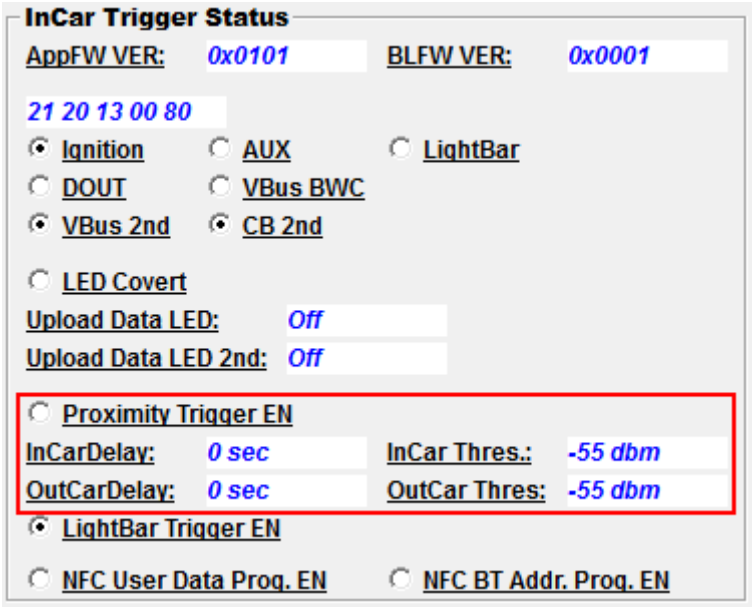
8.4.1 Proximity trigger recording

Proximity trigger recording is an automatic BWC recording remote control

function base on the RSSI, which is related to different distance between the paired BWC and In-Car Trigger, to turn on/off BWC's recording remotely:

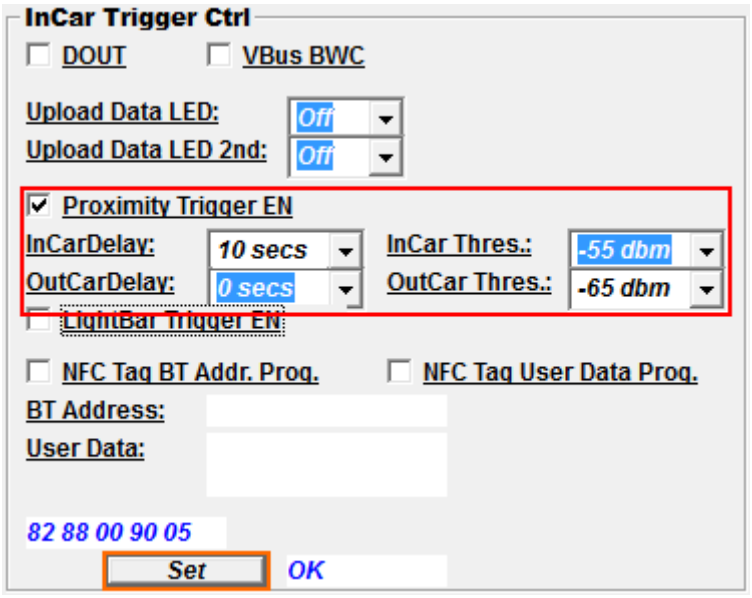
- If the RSSI is lower than a threshold which is considered that the officer leave the patrol car (EX. for examining the crime scene), In-Car Trigger will remotely turn on BWC's recording automatically.
- If the RSSI is higher than a threshold which is considered that the officer sit back to the driver seat of patrol car, In-Car Trigger will remotely turn off BWC's recording automatically.

In Test UI, In-Car Trigger Status window display the listed configuration of proximity trigger function:



- Proximity Trigger EN: A radio box that indicate the proximity trigger is enabled or not. It will be set if proximity trigger function is enabled.
- InCarDelay: A time delay parameter for considering the BWC is located in the patrol car. Refer to the following description on InCar Thres parameter.
- InCar Thres: The upper RSSI threshold, BWC would be considered located in the patrol car if the RSSI is higher than this value for defined InCarDelay secs.
- OutCarDelay: A time delay parameter for considering the BWC is located in the patrol car. Refer to the following description on OutCar Thres parameter.
- OutCar Thres: The lower RSSI threshold, BWC would be considered located out of the patrol car if the RSSI is lower than this value for defined OutCarDelay secs.

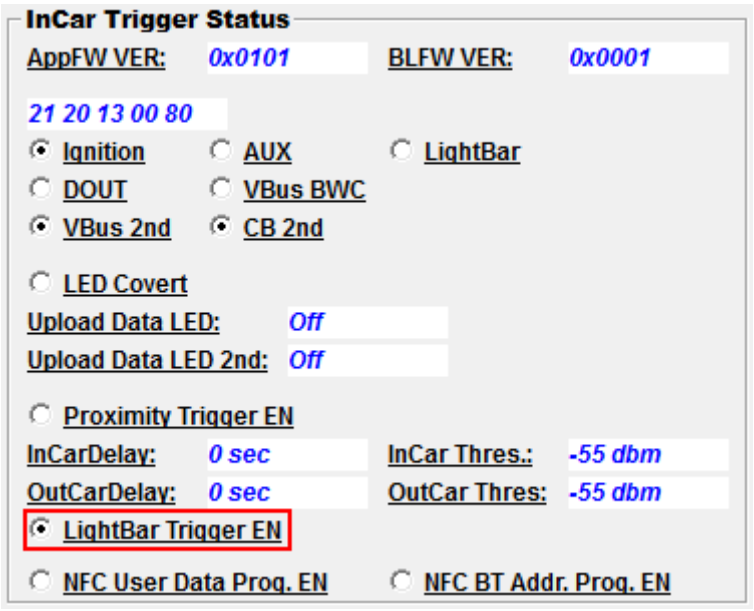
All the parameters above for proximity trigger configuration could also be configured under the In-Car Trigger Control window of Test UI:



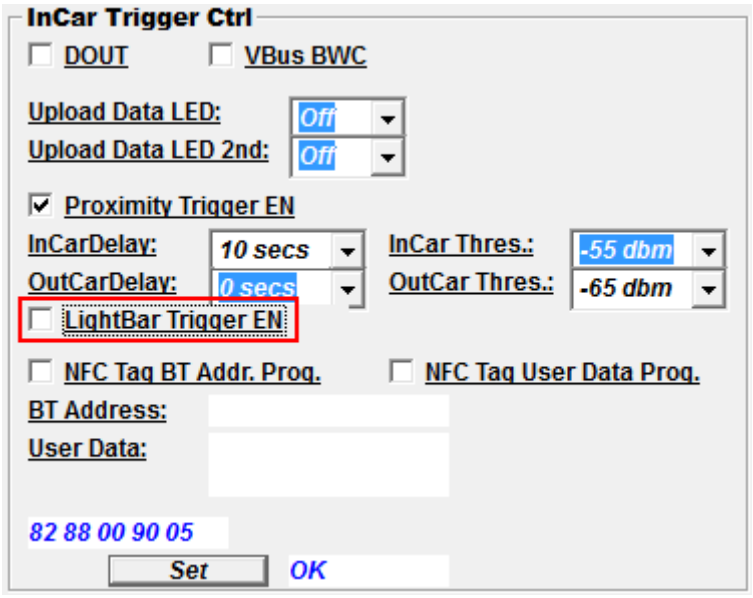
8.4.2 Light bar trigger recording

Light bar trigger recording is another automatic BWC recording remote control function base on the patrol car's light bar status. A status transition from turned-off to turned-on of light bar will make In-Car Trigger remotely turn on the both paired BWCs' recording.

In Test UI, In-Car Trigger Status window display light bar trigger function is enabled or not:

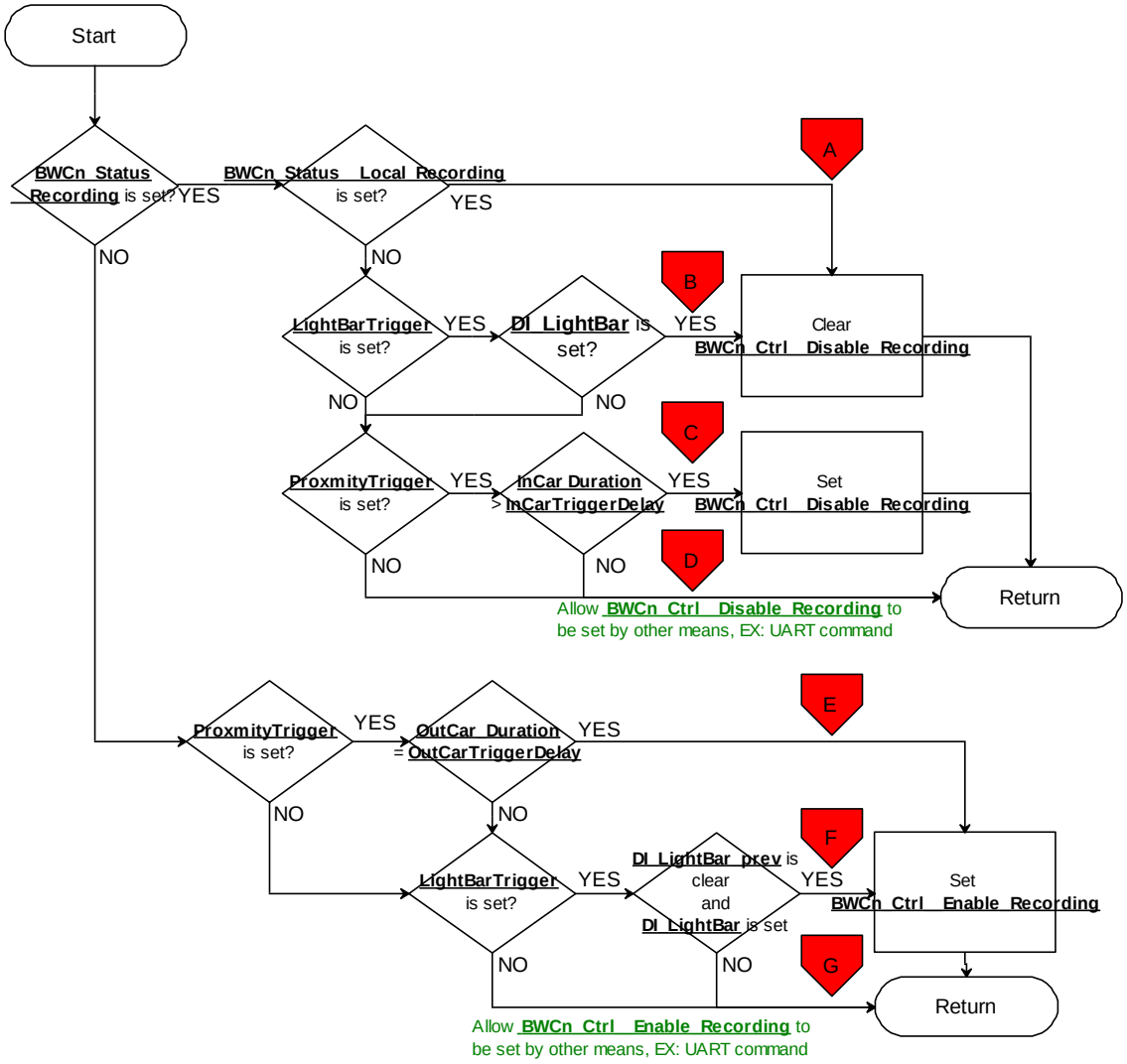


Also it could be configured in In-Car Trigger control window:


InCar Trigger Ctrl
DOUT
VBus BWC
Upload Data LED:
Off
Upload Data LED 2nd:
Off
Proximity Trigger EN
InCarDelay:
10 secs
InCar Thres.:
-55 dbm
OutCarDelay:
0 secs
OutCar Thres.:
-65 dbm
LightBar Trigger EN
NFC Tag BT Addr. Prog.
NFC Tag User Data Prog.
BT Address:
User Data:
82 88 00 90 05
Set
OK

8.4.3 Remote recording control integration

Through the introduction of section 8.4 to 8.4.2, multiple scenario would make In-Car Trigger remotely turn on/off the recording if it's paired BWCs separately. This section summarize the all these scenario to show how they work together.



Base on the recording is turned on or off in received status of paired BWC, the remote recording control scenarios could be listed in the following cases tagged in the control flow above:

A. **Forbidden remotely turn off recording due to local triggering**
   If BWC currently is recording, which is triggered by click the REC button of BWC locally, then it is not allowed to turn off the BWC recording remotely via wireless.

B. **Forbidden remotely turn off recording due to light bar triggering**
   Continue from (A). If BWC currently is recording, which is triggered by light bar triggering and the light bar is still turned on, then it is not allowed to turn off the BWC recording remotely via wireless.

C. Remote turn off recording due to proximity triggering.
Continue from (B). If BWC currently is recording, proximity triggering is enable and BWC is located in patrol car for defined duration. In-Car Trigger would remotely turn off the BWC's recording.

D. Remote turn off recording due to user's demand.
Continue from (C). User could still dispatch remote command to In-Car Trigger to turn off the recording of BWC.

E. Turn on recording due to proximity triggering
If BWC currently is not recording, proximity triggering is enabled, and BWC is located out of patrol car for defined duration. In-Car Trigger will automatically turn on this BWC's recording remotely.

F. Turn on recording due to Light bar triggering
Continue from (E). If BWC currently is not recording, proximity triggering is enabled, and BWC is located out of patrol car for defined duration. In-Car Trigger will automatically turn on this BWC's recording remotely.

G. Turn on recording due to user's demand
Continue from (F). User could still dispatch remote command to In-Car Trigger to turn on the recording of BWC.

9 Buzzer

This chapter summarize all buzzer notification generated from In-Car Trigger as the listed form below:

Notification event	Event description	Action of notification
Power on	Section 4.2	Continuously beep for 1 sec
NFC tag programming complete	Section 7.3	Shortly beep 3 times
Bluetooth pairing	Section 8.1.1	Short beeping 1 time: BWC1 start pairing Short beeping 2 times: BWC2 start pairing Continuously beep for 1 sec: Forbidden pairing or invalid data format on NFC tag.
Initiate firmware update process.	Section 13.2.2	Continuously beep for 1 sec 2 times: ready to receive controller firmware image to update.

10 Push buttons

This chapter summarize all user cases triggered by push buttons.

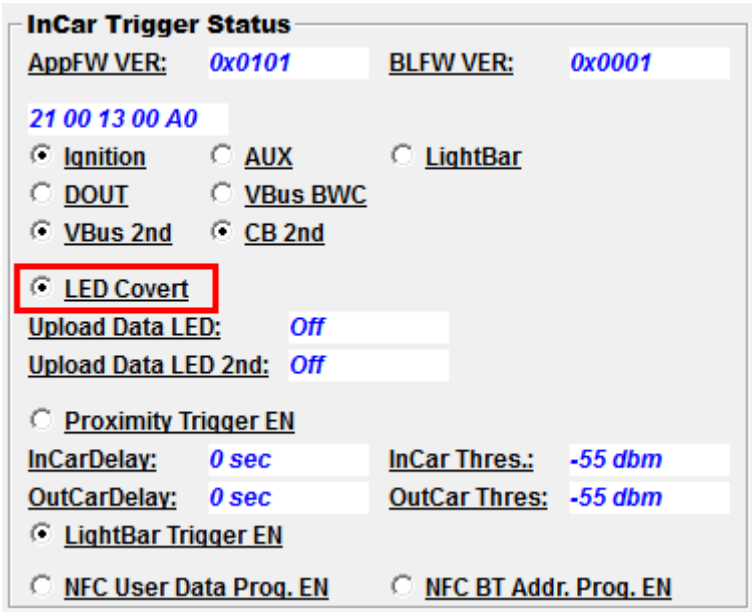
10.1 Power button

Press the power button will power on/shutdown the In-Car Trigger. refer to chapter 4 about the power on initialization.

10.2 Covert button

Press the Covert button will switch In-Car Trigger in/out of LEDs covert mode. When In-Car Trigger is in LEDs covert mode, all LEDs will be force turned off regardless the LED notification defined in previous chapters.

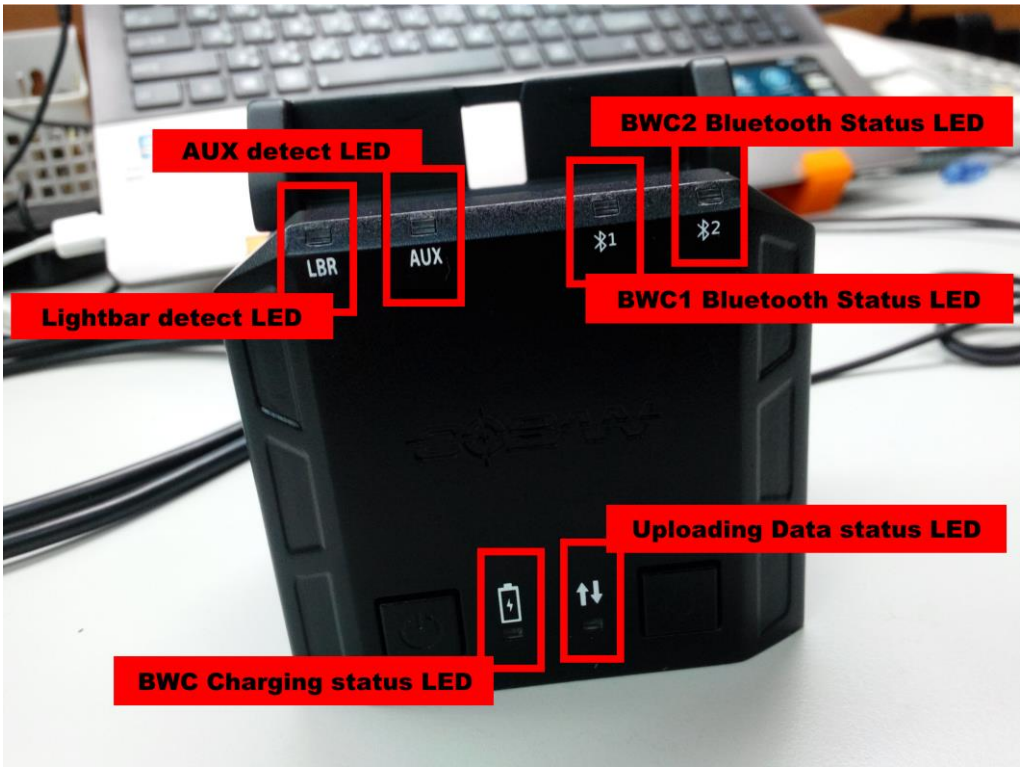
In Test UI, the current LED covert mode setting will be periodically updated on In-Car Trigger status window.



11 LEDs

This chapter summarize all LEDs notification descript in previous chapters.

11.1 Trigger Status LEDs



11.1.1 AUX LED

Notification event	Event description	Action of notification
Power on notification	Section 4.2	Continuously turned on for 1 sec
Spare on	Section 6.1	Turned on
Spare off	Section 6.1	Turned off

11.1.2 Light Bar LED

Notification event	Event description	Action of notification
Power on notification	Section 4.2	Continuously turned on for 1 sec
Light Bar on	Section 6.1	Turned on
Light Bat off	Section 6.1	Turned off

11.1.3 BWC1 Bluetooth Status LED

Notification event	Event description	Action of notification
Power on notification	Section 4.2	Continuously turned on for 1 sec
BWC1 pairing	Section 8.1.1	Blinking
BWC1 connected	Section 8.2	Turned on
BWC1 disconnected	Section 8.2.2	Turned off

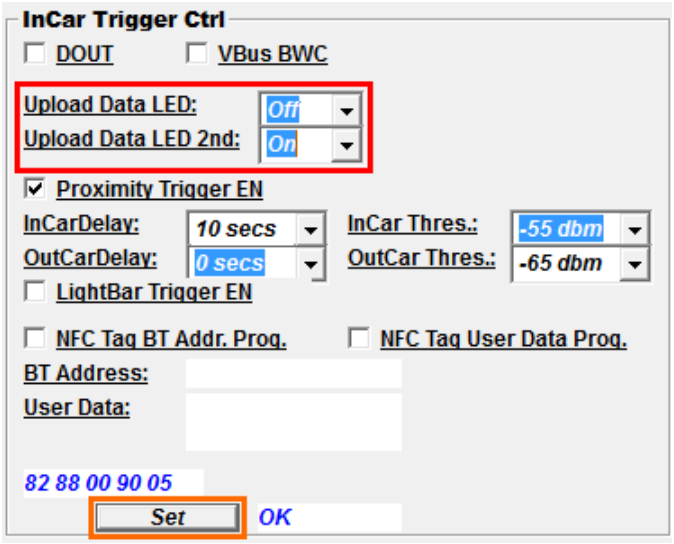
11.1.4 BWC2 Bluetooth Status LED

Notification event	Event description	Action of notification
Power on notification	Section 4.2	Continuously turned on for 1 sec
BWC2 pairing	Section 8.1.1	Blinking
BWC2 connected	Section 8.2	Turned on
BWC2 disconnected	Section 8.2.2	Turned off

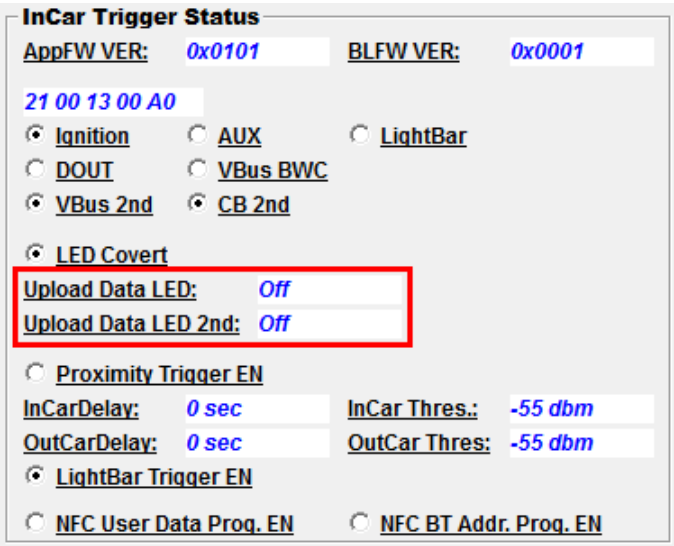
11.1.5 Upload Data status LED, Upload Data status LED 2nd

Upload data LED is directly controlled by the UART interface connected host device with defined UART command descript in section 12.2

In Test UI, In-Car Trigger Ctrl Window provide checkboxes to control these two Upload data status LEDs.



Also, current configuration of Upload data status LEDs would be periodically updated in In-Car Trigger status window.

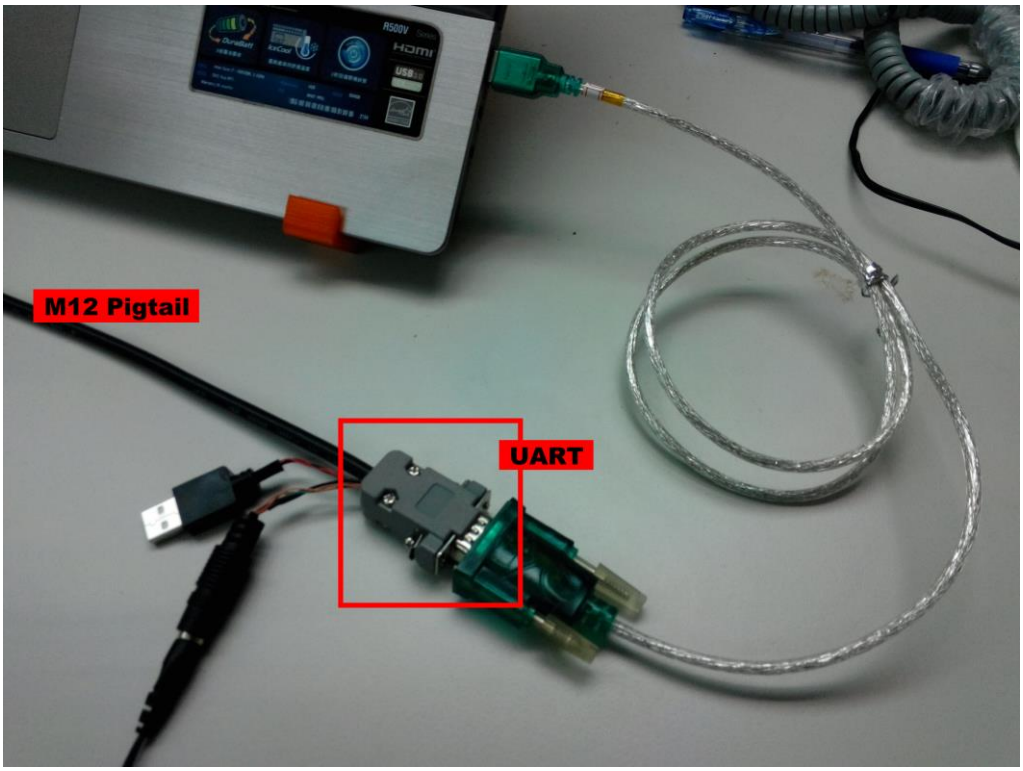


11.2 LEDs Covert mode

Refer to chapter 10.2, press Covert button will make In-Car Trigger switch in/out of LEDs Covert mode, which make all LEDs turned off regardless the definition of LEDs notification.

12 UART

In-Car Trigger implement 1 UART interface to provide serial communication to connected host device to access its internal register table for retrieving or controlling the internal status.



12.1 UART configuration and command format

The UART interface connected host device must following the listed configuration before dispatching UART command to In-Car Trigger:

- Baudrate: 115200
- Data Bits: 8
- Parity: None
- Stop Bits: 1
- Flow Type: RTS/CTS

The form below define the format of UART command which In-Car Trigger accept to access the context of its internal register table.

- **Registers read**

  The UART command format to retrieve ***N*** bytes data started from address ***Addr*** from register table would be the listed:

Byte Index	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Field Name	PID	Op. Code	Base Address (High Byte)	Base Address (Low Byte)	Access Length	Checksum
Value	*0xA1*	*0x00*	*Addr >> 8*	*Addr & 0x00FF*	*N (Note #1)*	*(Note #2)*

  (Note #1) The maximum access data length could be read/write in one command is 16 bytes.

  (Note #2) Checksum is evaluated by taking the 2's complement of the sum of

all value of previous bytes.

The format of register read command return echoed from In-Car Trigger would be the listed:

Byte Index	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Field Name	PID	Op. Code	Base Address (High Byte)	Base Address (Low Byte)	Access Length	Result
Value	*0xA1*	*0x00*	*Addr >> 8*	*Addr & 0x00FF*	*N (Note #1)*	*(Note #3)*

Byte Index	Byte 6 ~ Byte (6 + (N - 1))	Byte (6 + N)
Field Name	Data	Checksum
Value	*Context of Registers*	*(Note #2)*

(Note #3) Value of Result must be 0x00 to indicate command handling without error.

- **Registers write**

  The UART command format to write ***N*** bytes data started from address ***Addr*** into register table.

Byte Index	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Field Name	PID	Op. Code	Base Address (High Byte)	Base Address (Low Byte)	Access Length
Value	*0xA1*	*0x01*	*Addr >> 8*	*Addr & 0x00FF*	*N*

Byte Index	Byte 5 ~ Byte (5 + (N - 1))	Byte (5 + N)
Field Name	Data	Checksum
Value	*Context of Register*	*(Note #2)*

The format of register read command return echoed from In-Car Trigger would be the listed:

Byte Index	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Field Name	PID	Op. Code	Base Address (High Byte)	Base Address (Low Byte)	Access Length	Result
Value	*0xA1*	*0x01*	*Addr >> 8*	*Addr & 0x00FF*	*N*	*(Note #3)*

Byte Index	Byte 6
Field Name	Checksum
Value	*(Note #1)*

12.2Register table

Address	Register Name	Privilege	Description	
0x0000	Product ID	Read	In-Car Trigger Product ID. The value of its context should always be **0xA1**	
0x0001 ~0x0002	Bootloader firmware version		Bootloader firmware version (MSB at 0x0001)	
0x0003 ~0x0004	Application firmware version		Application firmware version (MSB at 0x0003)	
0x0005	Reserve			
0x0006	DIO Status 0	Read	Bit7	b0: Light bar LED is off. b1: Light bar LED is on.
			Bit6	b0: AUX LED is off. b1: AUX LED is on.
			Bit5	b0: BWC1 Bluetooth Status LED is off. b1: BWC1 Bluetooth Status LED is on.
			Bit4	b0: BWC2 Bluetooth Status LED is off. b1: BWC2 Bluetooth Status LED is on.
			Bit3	Reserve
			Bit2	b0: Light bar input is pull low b1: Light bar input is pull high
			Bit1	b0: AUX input is pull low b1: AUX input is pull high
			Bit0	Reserve
0x0007	DIO Status 1		Bit7	b0: DOUT output is off b1: DOUT output is on
			Bit6	b0: Upload Data status LED is off. b1: Upload Data status LED is on
			Bit5	b0: Upload Data status 2nd LED is off. b1: Upload Data status 2nd LED is on
			Bit4	b0: Ignition input is pull low. b1: Ignition input is pull high.

			Bit3	Reserve
			Bit2	b0: VBus BWC output is off b1: VBus BWC output is on
			Bit1	b0: VBus 2nd output is off b1: VBus 2nd output is on
			Bit0	b0: CB 2nd output is off b1: CB 2nd output is on
0x0008	Proximity trigger configuration	Read	Bit7 ~ Bit6	b00: InCarDelay is set to 0 sec. b01: InCarDelay is set to 10 secs. b10: InCarDelay is set to 30 secs. b11: InCarDelay is set to 60 secs.
			Bit5 ~ Bit4	b00: OutCarDelay is set to 0 sec. b01: OutCarDelay is set to 3 secs.
			Bit3 ~ Bit2	b00: InCarThreshold is set to 55 dBm. b01: InCarThreshold is set to 60 dBm. b10: InCarThreshold is set to 65 dBm. b11: InCarThreshold is set to 70 dBm.
			Bit1 ~ Bit0	b00: OutCarThreshold is set to 55 dBm. b01: OutCarThreshold is set to 60 dBm. b10: OutCarThreshold is set to 65 dBm. b11: OutCarThreshold is set to 70 dBm.
0x0009	Auto Trigger/ NFC / LED Convert configuration	Read	Bit7	b0: Light bar trigger recording is off. b1: Light bar trigger recording is on.
			Bit6	b0: Proximity trigger recording is off. b1: Proximity trigger recording is on.

			Bit5	b0: LEDs Covert mode is off. b1: LEDs Covert mode is on.
			Bit4 ~ Bit2	Reserve
			Bit1	b0: NFC tag Bluetooth Mac address programming is off. b1: NFC tag Bluetooth Mac address programming is off.
			Bit0	b0: NFC tag User Data programming is off. b1: NFC tag User Data programming is on.
0x000A ~ 0x0011	Reserve			
0x0012	BWC1 Bluetooth GAP status	Read	Bit7	Reserve
			Bit6	b0: Don't Care. b1: the last time Bluetooth pairing is with BWC1.
			Bit5	Reserve
			Bit4 ~ Bit0	Bluetooth GAP Status. b00000: INIT, b00001: STARTED, b00010: SCANNING, b00011: CONNECTING, b00100: CONNECTED, b00101: IDLE
0x0013	BWC1 Disconnection Code		Reason of the last Bluetooth disconnection.	
0x0014	BWC1 RSSI		BWC1 RSSI = - (Register value) dBm	
0x0015	BWC1 In-Car Duration		secs	
0x0016	BWC1 Out-Car Duration		secs	
0x0017	BWC1 Bluetooth GATT status	Read	Bit7	Reserve
			Bit6 ~ Bit0	GATT Status 0000000: Service discovery stop, 0000001: Service discovery init,

				0000010: Discovery BWC service, 0000011: Discovery BWC Status characteristic, 0000100: Discovery BWC Ctrl characteristic, 0001000: Updating BWC RSSI. 0001001: Read BWC Status characteristic. 0001010: Write BWC Ctrl characteristic.
0x0018	Reserve			
0x0019 ~0x001E	BWC1 Bluetooth Mac address	Read		6 Bytes Mac address retrieve from BWC1 NFC tag. (MSB at 0x0018)
0x001F ~0x003E	BWC1 User Data			32 Bytes User Data retrieve from BWC1 NFC tag. (MSB at 0x0018)
0x003F ~0x005F	Reserve			
0x0060	BWC1 Status 0	Read	Bit7 ~ Bit2	Reserve
			Bit1	0: BWC1 is recording 1: BWC1 is not recording.
			Bit0	Reserve
0x0061	BWC1 Status 1	Read	Reserve	
0x0062	BWC1 Status 2			
0x0063	BWC1 Status 3			
0x0064	BWC1 Status 4			
0x0065	BWC1 Status 5			
0x0066	BWC1 Status 6			
0x0067	BWC1 Status 7			
0x0068	BWC1 Status 8			
0x0069	BWC1 Status 9			
0x006A	BWC1 Status 10	Read	Reserve	
0x006B	BWC1 Status 11			
0x006C	BWC1 Status 12			
0x006D	BWC1 Status 13			
0x006E	BWC1 Status 14			
0x006F	BWC1 Status 15			

0x0070 ~0x0071	Reserve			
0x0072	BWC2 Bluetooth GAP status	Read	Bit7	Reserve
			Bit6	b0: Don’t Care. B1: the last time Bluetooth pairing is with BWC2.
			Bit5	Reserve
			Bit4 ~ Bit0	Bluetooth GAP Status. B00000: INIT, b00001: STARTED, b00010: SCANNING, b00011: CONNECTING, b00100: CONNECTED, b00101: IDLE
0x0073	BWC2 Disconnection Code		Reason of the last Bluetooth disconnection.	
0x0074	BWC2 RSSI		BWC2 RSSI = - (Register value) dBm	
0x0075	BWC2 In-Car Duration		secs	
0x0076	BWC2 Out-Car Duration		secs	
0x0077	BWC2 Bluetooth GATT status		Bit7	Reserve
			Bit6 ~ Bit0	GATT Status b0000000: Service discovery stop, b0000001: Service discovery init, b0000010: Discovery BWC service, b0000011: Discovery BWC Status characteristic, b0000100: Discovery BWC Ctrl characteristic, b0001000: Updating BWC RSSI. b0001001: Read BWC Status characteristic. b0001010: Write BWC Ctrl characteristic.
0x0078	Reserve			

0x0079 ~0x007E	BWC2 Bluetooth Mac address	Read	6 Bytes Mac address retrieve from BWC2 NFC tag. (MSB at 0x0018)	
0x007F ~0x009E	BWC2 User Data		32 Bytes User Data retrieve from BWC2 NFC tag. (MSB at 0x0018)	
0x009F ~0x00BF	Reserve			
0x00C0	BWC2 Status 0	Read	Bit7 ~ Bit2	Reserve
			Bit1	b0: BWC2 is recording b1: BWC2 is not recording.
			Bit0	Reserve
0x00C1	BWC2 Status 1		Reserve	
0x00C2	BWC2 Status 2			
0x00C3	BWC2 Status 3			
0x00C4	BWC2 Status 4			
0x00C5	BWC2 Status 5			
0x00C6	BWC2 Status 6			
0x00C7	BWC2 Status 7			
0x00C8	BWC2 Status 8			
0x00C9	BWC2 Status 9			
0x00CA	BWC2 Status 10			
0x00CB	BWC2 Status 11			
0x00CC	BWC2 Status 12			
0x00CD	BWC2 Status 13			
0x00CE	BWC2 Status 14			
0x00CF	BWC2 Status 15			
0x00D0	Digital output control 0	Write	Bit7	0: Don’t care 1: Turn off VBus BWC
			Bit6	0: Don’t care 1: Turn on VBus BWC
			Bit5	0: Don’t care 1: Turn off VBus 2nd
			Bit4	0: Don’t care 1: Turn on VBus 2nd
			Bit3	0: Don’t care 1: Turn off DOUT 2nd
			Bit2	0: Don’t care 1: Turn on DOUT 2nd
			Bit1	0: Don’t care

				1: Turn off DOUT
			Bit0	0: Don't care 1: Turn on DOUT
0x00D1	Digital output control 1		Bit7 ~ Bit4	Reserve
			Bit3	0: Don't care 1: Turn off Upload Data LED
			Bit2	0: Don't care 1: Turn on Upload Data LED
			Bit1	0: Don't care 1: Turn off CB 2nd
			Bit0	0: Don't care 1: Turn on CB 2nd
0x00D2	Proximity Trigger control	Write	Bit7 ~ Bit6	b00: Set InCarDelay is set to 0 sec. b01: Set InCarDelay is set to 10 secs. b10: Set InCarDelay is set to 30 secs. b11: Set InCarDelay is set to 60 secs.
			Bit5 ~ Bit4	b00: Set OutCarDelay is set to 0 sec. b01: Set OutCarDelay is set to 3 secs.
			Bit3 ~ Bit2	b00: Set InCarThreshold is set to 55 dBm. b01: Set InCarThreshold is set to 60 dBm. b10: Set InCarThreshold is set to 65 dBm. b11: Set InCarThreshold is set to 70 dBm.
			Bit1 ~ Bit0	b00: Set OutCarThreshold is set to 55 dBm. b01: Set OutCarThreshold is set to 60 dBm. b10: Set OutCarThreshold is set to 65 dBm. b11: Set OutCarThreshold is set to 70 dBm.
0x00D3	Auto Trigger	Write	Bit7	0: Don't Care

	control			1: Enable light bar trigger recording
			Bit6	0: Don't Care 1: Disable light bar trigger recording
			Bit5	0: Don't Care 1: Enable proximity trigger recording
			Bit4	0: Don't Care 1: Disable proximity trigger recording
			Bit3 ~ Bit0	Reserve
0x00D4	NFC Reader control	Write	Bit7 ~ Bit4	Reserve
			Bit3	Enable NFC tag Bluetooth Mac address programming.
			Bit2	Disable NFC tag Bluetooth Mac address programming.
			Bit1	Enable NFC tag User Data programming.
			Bit0	Disable NFC tag User Data programming.
0x00D5 ~ 0x00DA	6 bytes programmed NFC tag Bluetooth Mac address.	Write	The Bluetooth Mac address to be programmed into NFC tag. (MSB at 0x00D3)	
0x00DB ~ 0x00FA	32 bytes programmed NFC tag User Data.	Write	The Bluetooth Mac address to be programmed into NFC tag. (MSB at 0x00D9)	
0x00FB	Bootloader	Write	Bit7 ~ Bit1	Reserve
			Bit0	Firmware update triggering (Refer to Chapter 13…TBD)
0x00FC	BWC1 Control 0	Write	Bit7 ~ Bit2	Reserve
			Bit1	Remote disable BWC1 recording
			Bit0	Remote enable BWC1

				recording
0x00FD	BWC1 Control 1	Write	Reserve	
0x00FE	BWC1 Control 2			
0x00FF	BWC1 Control 3			
0x0100	BWC1 Control 4			
0x0101	BWC1 Control 5			
0x0102	BWC1 Control 6			
0x0103	BWC1 Control 7			
0x0104	BWC1 Control 8			
0x0105	BWC1 Control 9			
0x0106	BWC1 Control 10			
0x0107	BWC1 Control 11			
0x0108	BWC1 Control 12			
0x0109	BWC1 Control 13			
0x010A	BWC1 Control 14			
0x010B	BWC1 Control 15			
0x010C ~ 0x0112	Reserve			
0x0113	BWC2 Control 0	Write	Bit7 ~ Bit2	Reserve
			Bit1	Remote disable BWC2 recording
			Bit0	Remote enable BWC2 recording
0x0114	BWC2 Control 1	Write	Reserve	
0x0115	BWC2 Control 2			
0x0116	BWC2 Control 3			
0x0117	BWC2 Control 4			
0x0118	BWC2 Control 5			
0x0119	BWC2 Control 6			
0x011A	BWC2 Control 7			
0x011B	BWC2 Control 8			
0x011C	BWC2 Control 9			
0x011D	BWC2 Control 10			
0x011E	BWC2 Control 11			
0x011F	BWC2 Control 12			
0x0120	BWC2 Control 13			
0x0121	BWC2 Control 14			
0x0122	BWC2 Control 15			
0x0123	Reserve			

~ 0x0129	

13 Controller firmware update

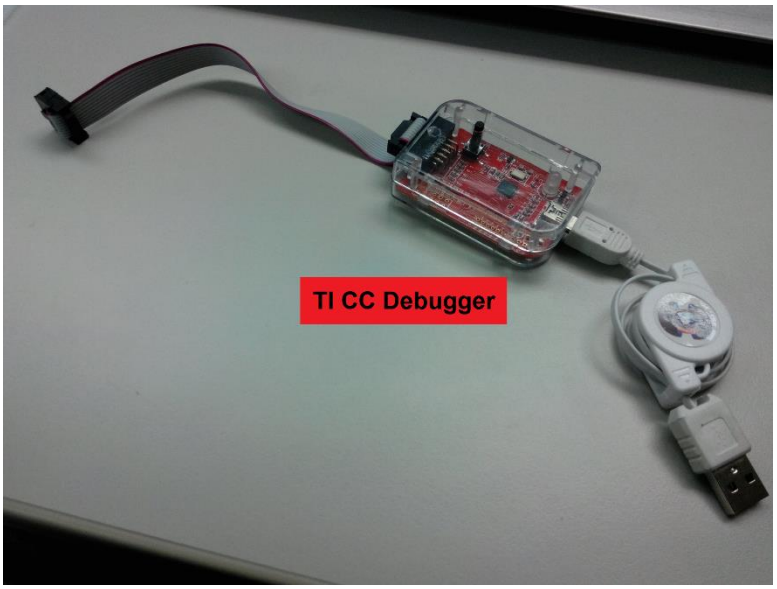
In-Car Trigger provide 2 ways to flush/update the firmware of controller: Hardware flushing and Serial bootloader.

13.1 Hardware flushing

Firmware hardware flushing is the way that completely erase the In-Car Trigger controller and replace it with a new provided firmware image. The cover case of In-Car Trigger have to be removed and a controller vendor specific fixture is required for this flushing process.

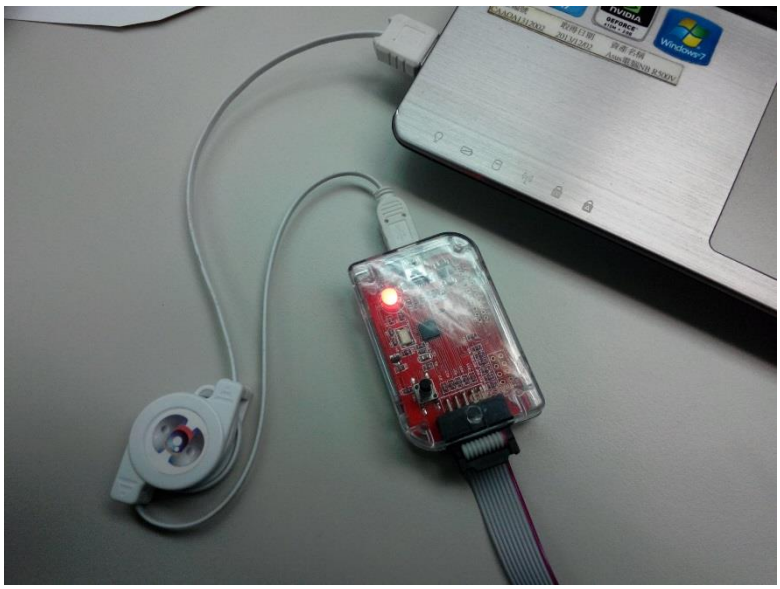
13.1.1 Environment setup

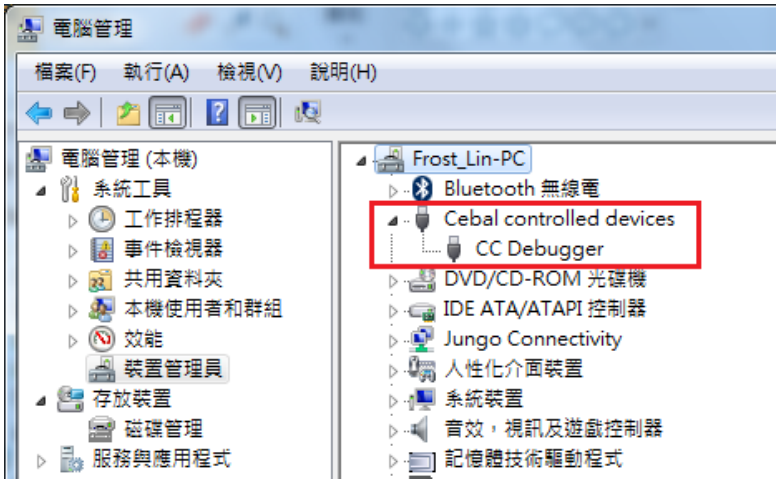
Texas Instruments CC Debugger is the fixture need to connect to the In-Car Trigger to erase the In-Car Trigger controller.



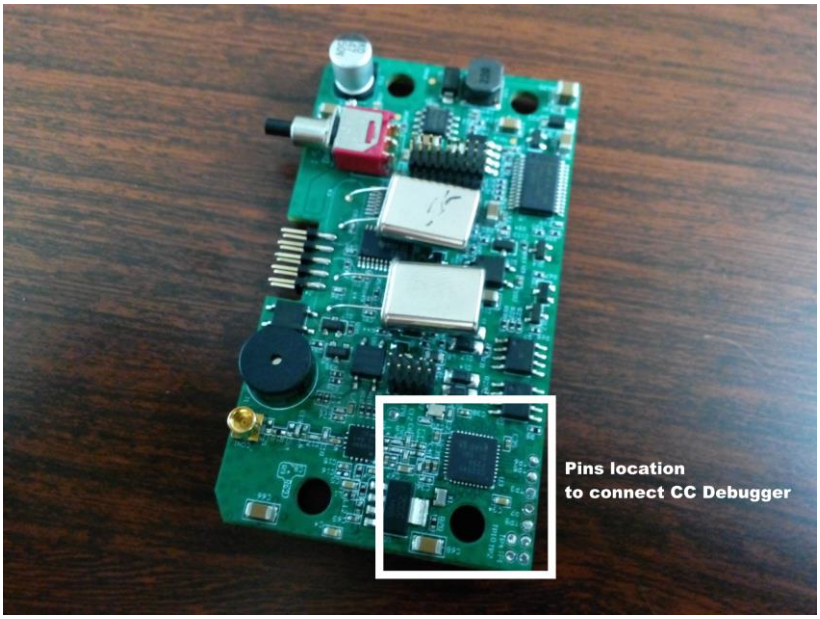
Texas Instruments Smart RF Flash Programmer software package contains the driver of CC debugger to let the connected host system to identify this device. Download and install the Smart RF Flash Programmer from Texas Instruments official Website http://www.ti.com/tool/flash-programmer.

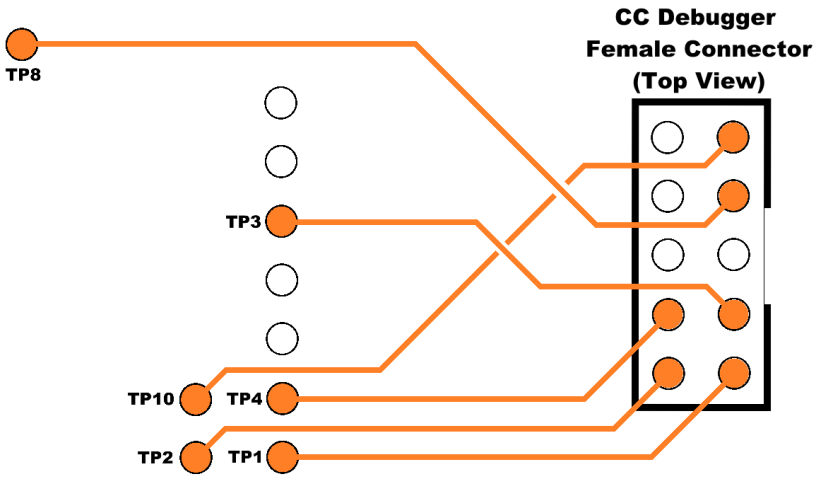
After successfully install the Smart RF Flash Programmer on the host system, plug the USB connector of CC debugger into the USB port of host system. CC debugger should be identified under the device manager of the connected host system.



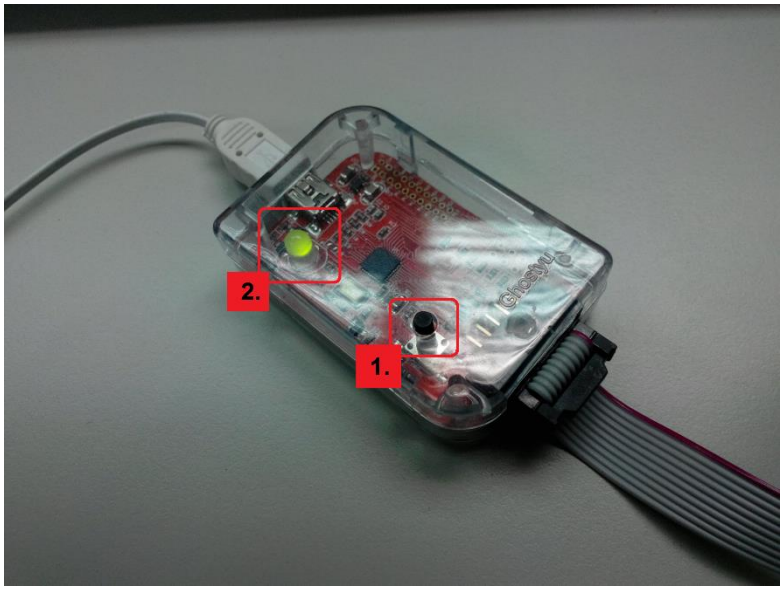


Connect the 2x5 connector of CC Debugger to the In-Car Trigger PCB as the pin map below.



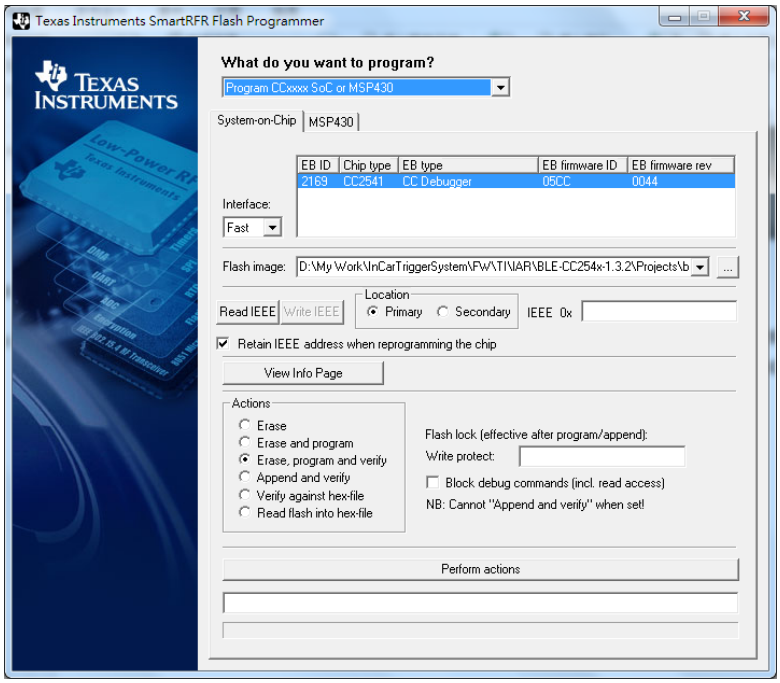


Click the reset button of CC Debugger, the connection status LED should turn to green if the debugger successfully connect to the In-Car Trigger controller.

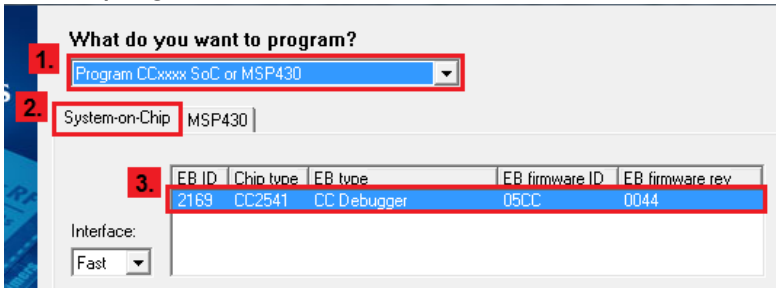


13.1.2 Firmware Flushing

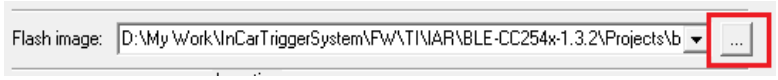
Once installing the Smart RF Flash Programmer, Launch Smart RF Flash Programmer on host system to bring up the In-Car Trigger controller flash programming user interface.



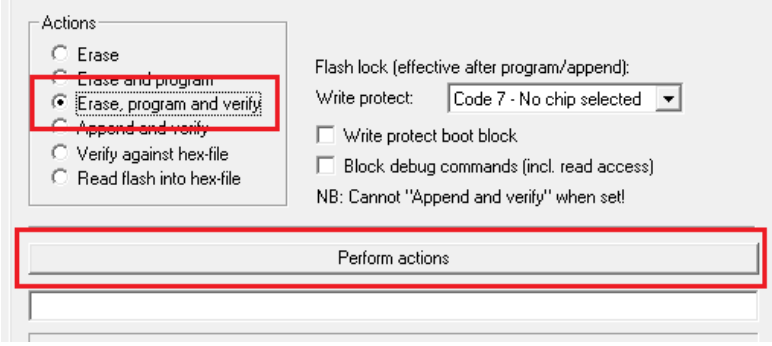
Correctly select the target device and make sure CC debugger is recognized by the flash programmer



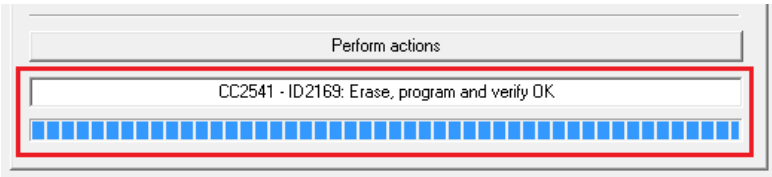
Select the provided firmware image *.hex file for flushing the In-Car Trigger controller.



Configure Flushing procedure and click “Perform action” to start firmware flushing.



Unplug the CC Debugger from Inn-Car Trigger when flash programmer generate programming successfully message.



13.2Serial Bootloader

Serial Bootloader provide a pure software process to flush the firmware of controller with provided firmware image file. The process requires no specific fixture or removing the cover case.

[Note]

Serial Bootloader is staring phasing-in In-Car Trigger controller firmware since **V1.01** to provide ability to update the controller to the **LATER** version firmware. If the current controller firmware version of In-Car Trigger is less than V1.01, it still require to hardware flushing the firmware to V1.01 with provided firmware image first to support serial bootloader.

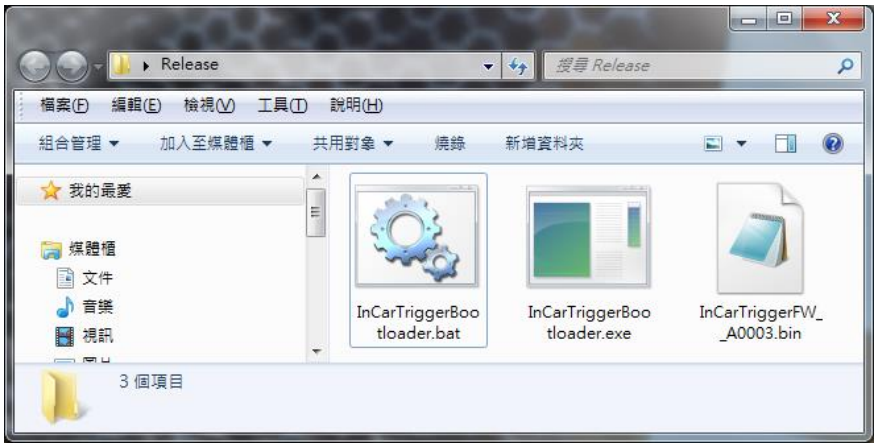
13.2.1 Environment setup

The M12 pigtail of In-Car Trigger have to be connected to a 8~30V power supply and Windows system host device. Make sure In-Car Trigger is powered on, the connected Windows system successfully identify the UART interface on M12 pigtail, and DO NOT let any application (includes the In-Car Trigger Test UI) on Windows system occupy the UART port.

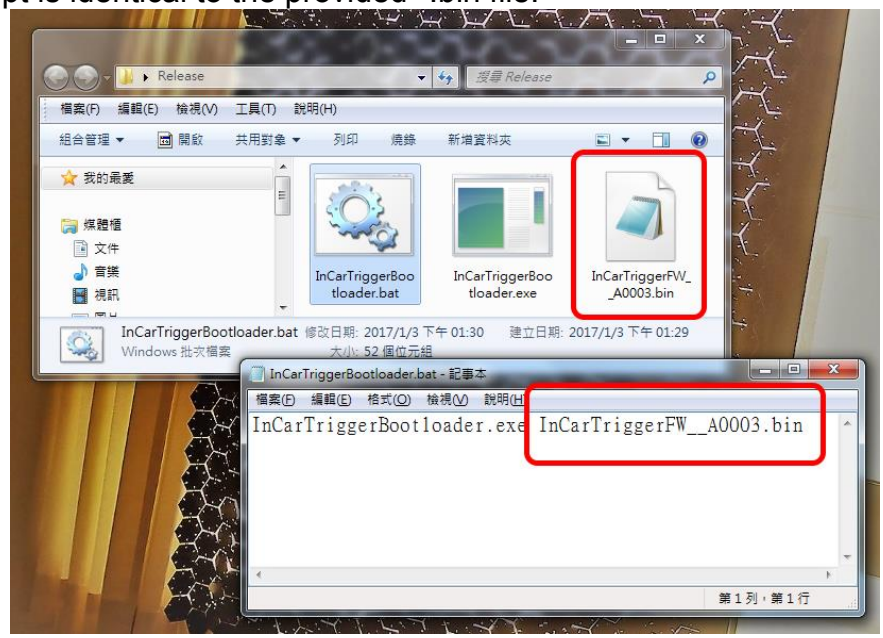
13.2.2 Firmware flushing

The listed file will be provided for the usage of serial bootloader function.

- **InCarTriggerBootloader.exe**: An executable of serial bootloader application.
- **InCarTriggerBootloader.bat**: Batch script.
- **InCarTriggerFW_A****.bin:** In-Car Trigger controller firmware.



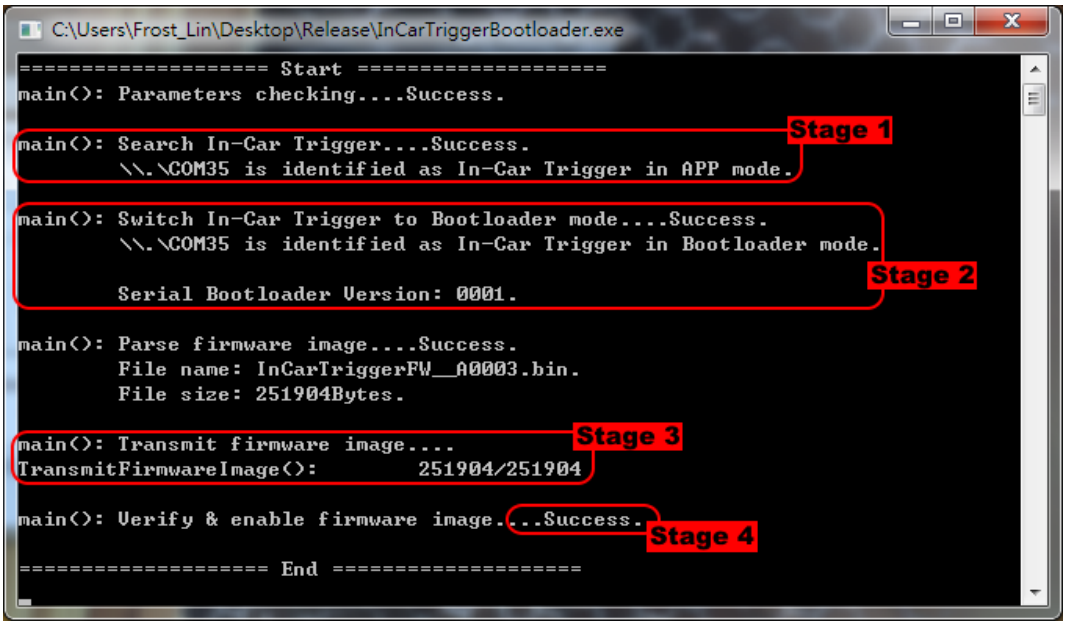
Before launching the firmware flushing process, edit the provided InCarTriggerBootloader.bat script to make sure the filename specified in batch script is identical to the provided *.bin file.



Save and execute InCarTriggerBootloader.bat script to initiate the firmware flushing process and pop up command line window below to notify user the progress of whole process:

- **Stage 1:** Bootloader application will poll all serial ports of system to search out the one connected to the In-Car Trigger.
- **Stage 2**: Bootloader application will generate UART command to switch the In-Car Trigger to handle firmware flushing commands. If In-Car Trigger successfully receive this command, it would generate 2 long beeping to notify ready to receive the controller firmware image.
- **Stage 3**: Bootloader application will keep displaying how many bytes are already sent versus how many bytes should be sent as the progress notification on firmware transmission.

- **Stage 4**: Bootloader application will poll the return status sent from In-Car Trigger after finish the firmware transmission.



14 TBD

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.

FCC Caution: To assure continued compliance, any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment. (Example - use only shielded interface cables when connecting to computer or peripheral devices).

FCC Radiation Exposure Statement

This equipment complies with FCC RF radiation exposure limits set forth for an uncontrolled environment.

This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

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.