



# MtConnect08

## JTAG Quick Guide

---

GUIDELINE

25<sup>th</sup> August, 2017

## REVISION HISTORY

Revision	Date	Author	Reason
V0.0.1	2017/03/016	Ray Tsou	New Create (draf)

# CONTENTS

Revision HISTORY.....	1
CONTENTS .....	3
Introduction .....	4
1. LoRa Developer Kit (DK) JTAG Interface.....	4
1.1. Prerequisites.....	4
1.2. Setup JLink Steps .....	5
Step 1: Connect the LoRa DK to host PC.....	5
Step 2: Setup JLink USB driver .....	6

# Introduction

This guide help user to setup LoRa DK JTAG interface in HW and SW.

## 1. LoRa Developer Kit (DK) JTAG Interface

Intel Qurak C1000 Developer Kit uses “FTDI USB to JTAG” connect to JTAG interface of C1000. And Intel System Studio for Microcontrollers (ISSM) use OpenOCD JTAG to programming and debugging. So, if want use another USB to JTAG device with ISSM, user need configure “openocd configure file” to setup a new USB to JTAG interface. ISSM default setting is FTDI. In this section, use a JLink to replace FTDI.

### 1.1. Prerequisites

Host machine:

PC running Microsoft Windows 7 or later with an available USB port.

Hardware :

- (1). An LoRa DK.
- (2). SEGGER Debug Probes - JLink device.

Software :

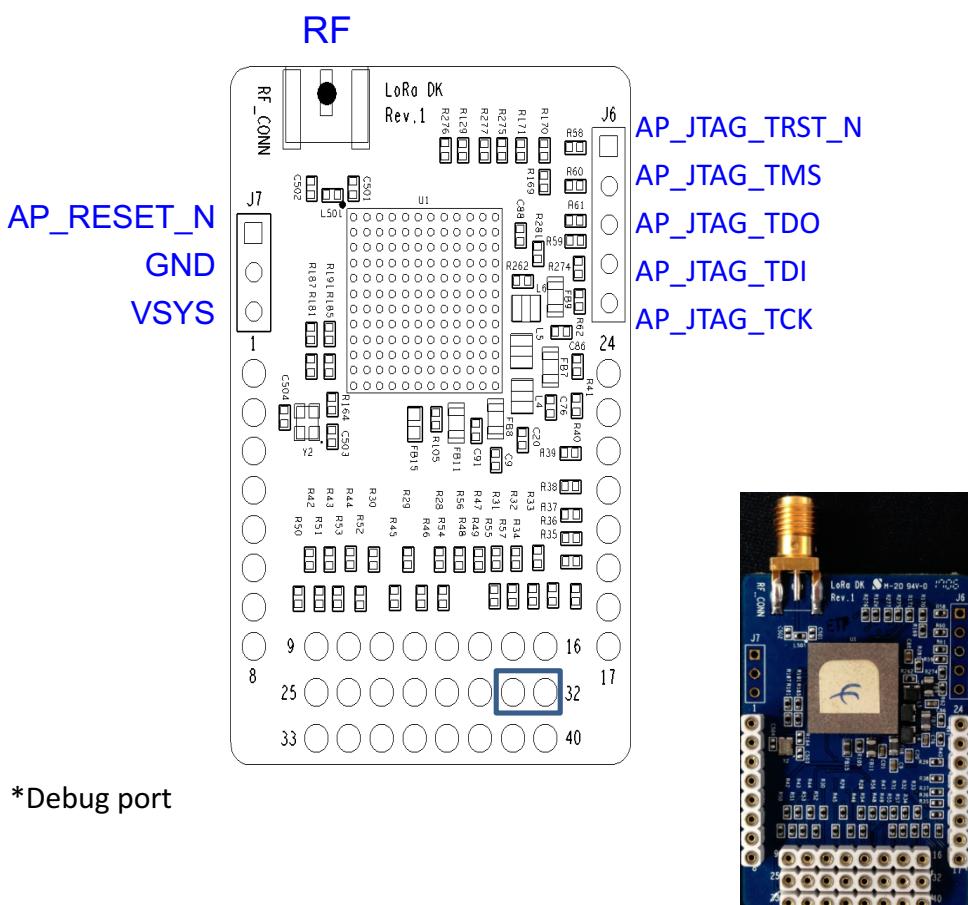
- (1). Intel IDE : Intel System Studio for Microcontrollers.
- (2). Zadig : USB driver tool

## 1.2. Setup JLink Steps

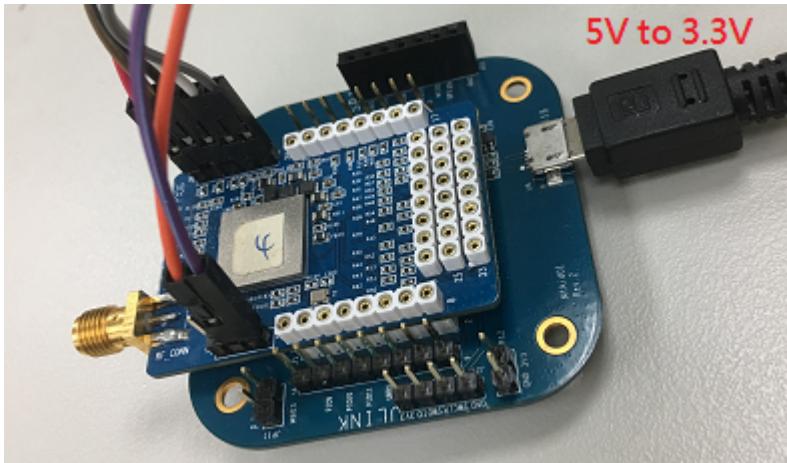
## **Step 1: Connect the LoRa DK to host PC.**

Intel Quark C1000 support JTAG interface for programming and debugging. The JTAG interface of LoRa DK is on the top left and right (block pin headers), list as below figure. Connect JTAG interface to those pins.

Board No.		
Series Pin No.	Functions	Intel LoRa
1	AVDD	VCC_IO
2	DEC2	-
3	GND	GND
4	GND	GND
5	VDD	VSYS
6	SWDIO	-
7	SWDCLK	-
8	GND	GND
9	P0_27	-
10	P0_26	-
11	P0_16	AP_UART[0]_RXD_ADC[18]
12	P0_15	AP_UART[0]_TXD_GPIO[31]
13	P0_14	AP_I2C[0]_SS_SDA
14	P0_13	AP_I2C[0]_SS_SCL
15	P0_09	AP_GPIO_SS[2]_ADC[10]
16	P0_08	AP_GPIO_AON[0]
17	P0_07	AP_GPIO[0]_ADC[0]
18	P0_06	AP_GPIO[24]_SPI[0]_M_CS[0]
19	P0_05	AP_GPIO[23]_SPI[0]_M_MOSI
20	P0_04	AP_GPIO[22]_SPI[0]_M_MISO
21	P0_03	AP_I2C[1]_SS_SDA
22	P0_02	AP_I2C[1]_SS_SCL
23	P0_01	AP_GPIO[21]_SPI[0]_M_SCK
24	P0_00	AP_GPIO_SS[3]_ADC[11]
25	P0_22	AP_GPIO_SS[10]_PWM[0]
26	P0_23	AP_GPIO_SS[11]_PWM[1]
27	P0_24	AP_GPIO_SS[12]_PWM[2]
28	P0_25	AP_GPIO_SS[13]_PWM[3]
29	P0_28	AP_GPIO_SS[0]_ADC[8]_UART[1]_CTS
30	P0_29	AP_GPIO_SS[1]_ADC[9]_UART[1]_RTS
31	P0_30	AP_GPIO_SS[8]_ADC[16]_UART[11]_TXD
32	P0_31	AP_GPIO_SS[9]_ADC[17]_UART[11]_RXD
33	P0_21	AP_SPI[0]_SS_MISO
34	P0_20	AP_SPI[0]_SS_MOSI
35	P0_19	AP_SPI[0]_SS_SCK
36	P0_18	AP_SPI[0]_SS_CS[0]
37	P0_17	AP_GPIO[1]_ADC[1]
38	P0_12	AP_GPIO_SS[4]_ADC[12]
39	P0_11	AP_GPIO_AON[1]
40	P0_10	AP_GPIO_AON[2]



\*Inter Quark C1000 use UART(1) as debug port. When user use “QM\_PRINT” function, the output will at UART(1).

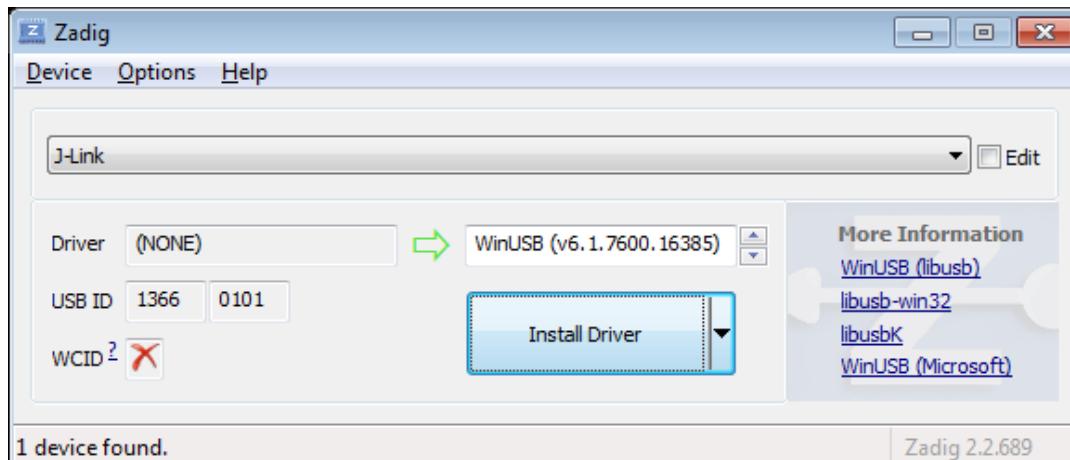


## Step 2: Setup JLink USB driver

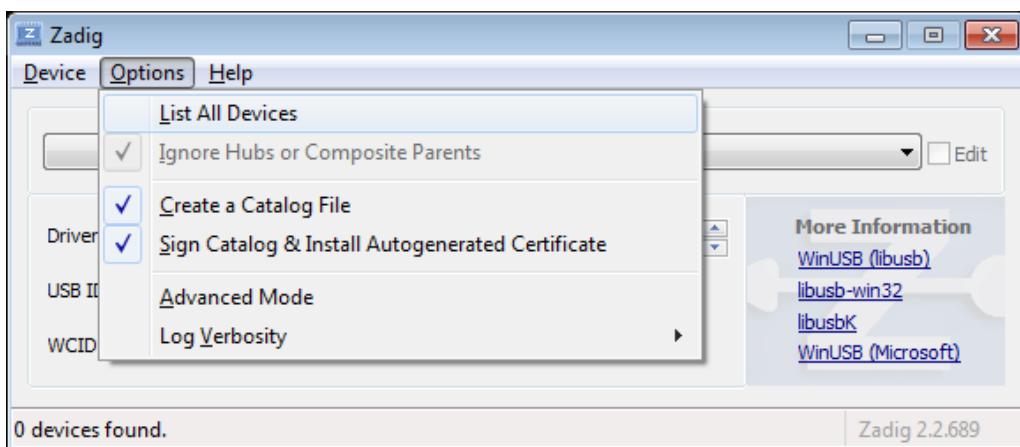
After step1, host PC should be install JLink usb driver, but this driver can't use with ISSM. We need change it via Zadig tool. And configure "openocd configure file".

### (Step 2-1) Changing the USB driver

To use the J-Link as a debug adapter for OpenOCD, at first the USB driver needs to be changed from the J-Link driver to WinUSB. This can be easily done by using the tool [Zadig](http://zadig.akeo.ie/) (<http://zadig.akeo.ie/>). Chose the J-Link from the device list, select "WinUSB" as driver and then click "Install Driver" to install the WinUSB driver.



If the J-Link's USB driver is already installed, it will not be shown in the device list by default. You need to enable "List All Devices" in the Options menu, then it will show up.



## (Step 2-2)Configure OpenOCD

Once the USB driver is changed (maybe a reboot is required), the J-Link can be used as a debug adapter for OpenOCD. User need configure “openocd configure file” which locate in ***\$ISSM install directory)/ISSM\_2016.1.0xx/tools/debugger/openocd/scripts/board***. And the file name is ***“quark\_se\_onboard.cfg”***. We use Jlink to replace FTDI. It need switch interface to Jlink. Add new line [ interface jlink ], and mark all ftdi setting like below :

```
<quark_se_onboard.cfg>

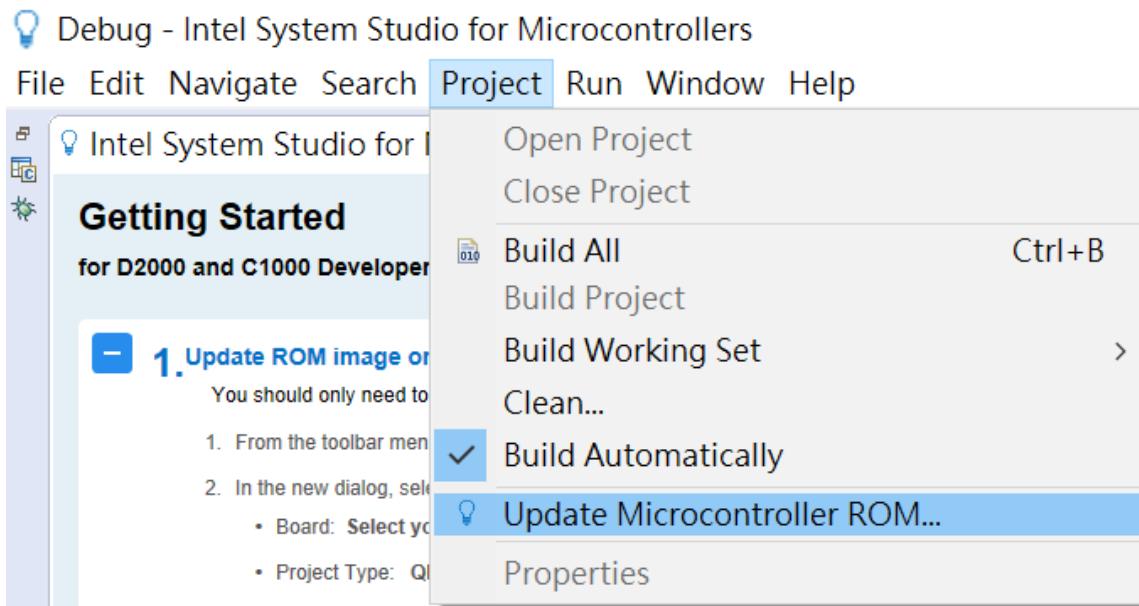
## Onboard FT2232D Configuration
interface jlink
#interface ftdi
#ftdi_vid_pid 0x0403 0x6010
#ftdi_channel 0
#ftdi_layout_init 0x0110 0xffff
#ftdi_layout_signal nTRST -data 0x0100

source [find target/quark_se.cfg]
# default frequency but this can be adjusted at runtime
adapter_khz 1000
reset_config trst_only
# need a delay before the initial discovery scan can succeed
jtag_ntrst_delay 300
tcl_port 4226
```

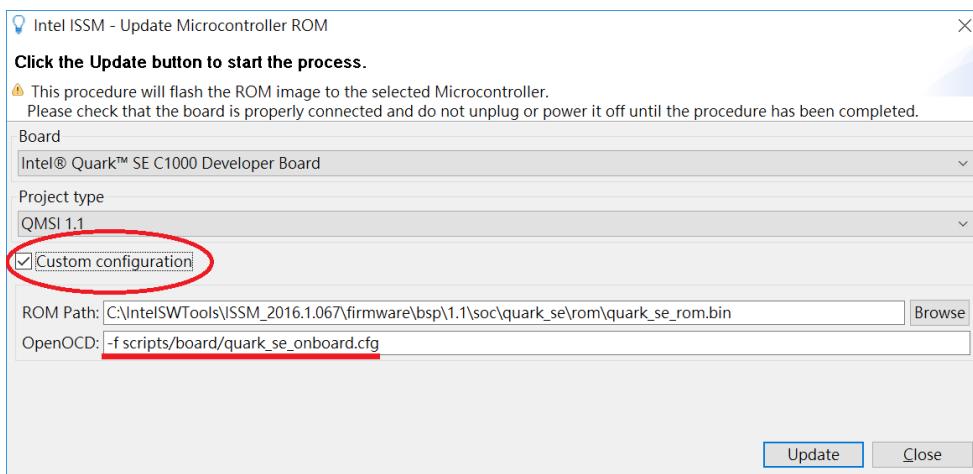
## (Step 2-3) Testing with Update target ROM

Note: Updating the boot ROM only needs to be performed once. Check with your lab instructor to determine if this step has already been done for your board

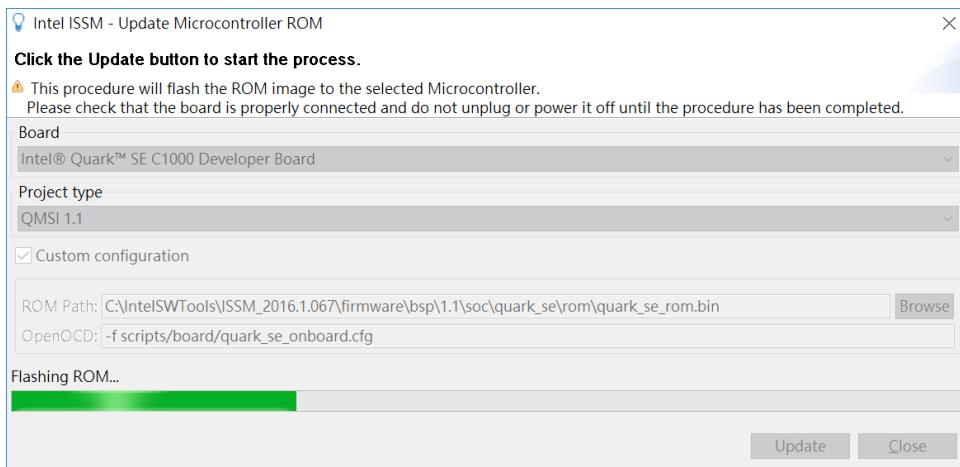
From the Project menu select the [Update Microcontroller ROM...] menu item.

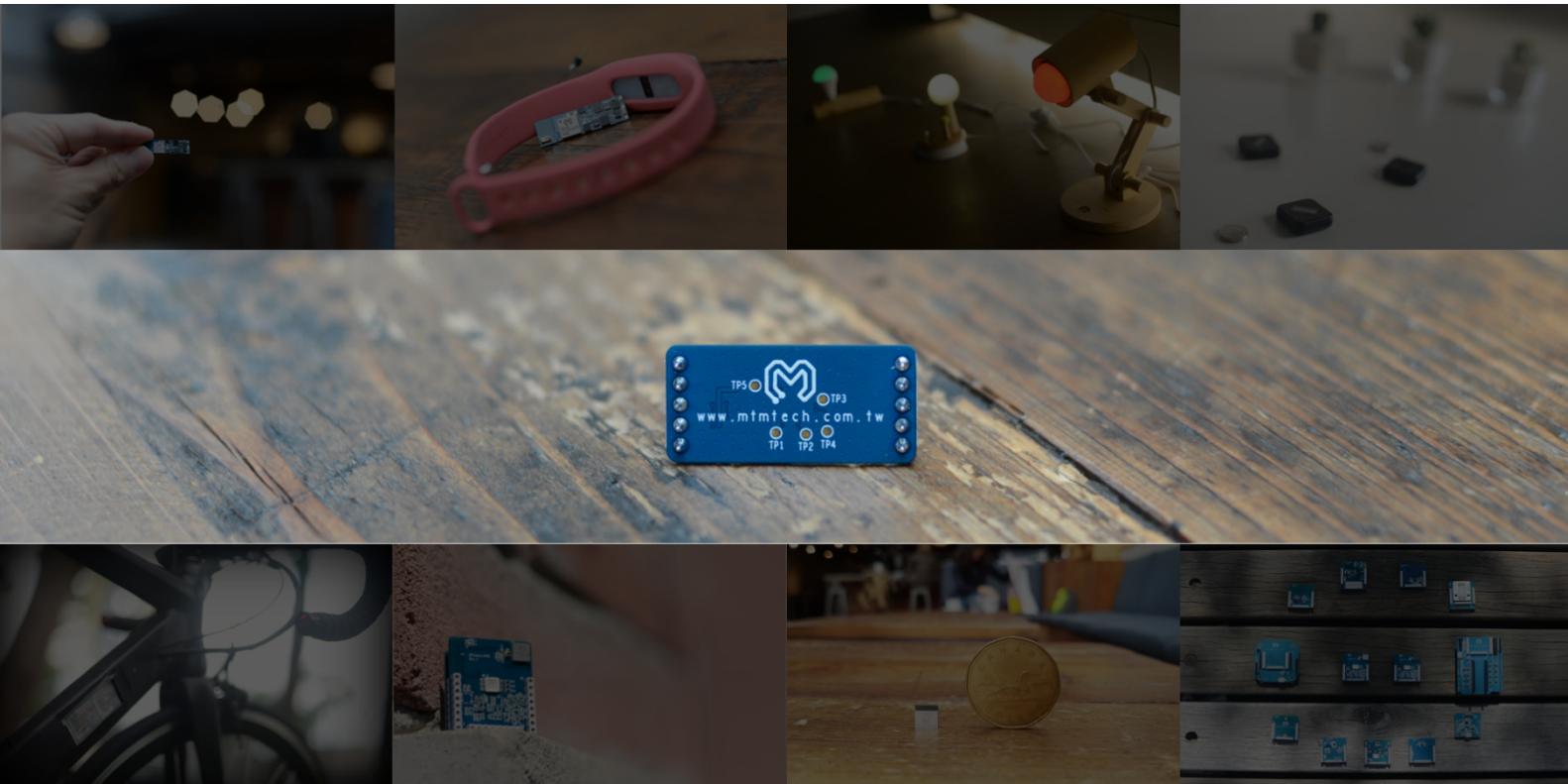


The Update Microcontroller ROM dialog box will appear. Make sure the Board is the **Intel® Quark™ SE C1000 Developer Board**, and the Project type is **QMSI 1.1**. Select **[Custom configuration]** check box , check OpenOCD configuration file path is correct. Click the **Update** button to re-flash the microcontroller's ROM.



If new configuration workable, the status bar will show "Flashing ROM ...", else it will pop-up an error message window.





## MtM+ Technology Corporation

[sales@mtmtech.com.tw](mailto:sales@mtmtech.com.tw)

7F, 178, Sec. 3, Minquan East Rd.  
Songshan District, Taipei,  
Taiwan (R.O.C.)

+886-2-7736-7386



<https://www.mtmtech.com.tw/>



<https://blog.mtmtech.com.tw/>



<https://www.facebook.com/MtMTechnologyCorporation>



<https://www.instagram.com/mtmtech>

© 2017 MtM+ Technology Corporation

DISCLAIMER: The information in this document is provided in connection with MtM+ Technology products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of MtM+ Technology products. MTM+ TECHNOLOGY ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL MTM+ TECHNOLOGY BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF MTM+ TECHNOLOGY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. MtM+ Technology makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. MtM+ Technology does not make any commitment to update the information contained herein. MtM+ Technology products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

SAFETY-CRITICAL, MILITARY, AND AUTOMOTIVE APPLICATIONS DISCLAIMER: MtM+ Technology products are not designed for and will not be used in connection with any applications where the failure of such products would reasonably be expected to result in significant personal injury or death ("Safety-Critical Applications") without an MtM+ officer's specific written consent. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. MtM+ Technology products are not designed nor intended for use in military or aerospace applications or environments unless specifically designated by MtM+ Technology as military-grade.