



Powerful Processors – Easy to Use™

# RTA-FoUSB-MON

Flash Programmer/In-Circuit Debugger

---

# User's Manual

Rev. 1.8  
August 2003

## Table of Contents

<b>1. Kit Overview.....</b>	<b>2</b>
<b>2. Software Installation .....</b>	<b>3</b>
<b>3. Driver Installation.....</b>	<b>4</b>
<b>4. Running KD30 Debugger .....</b>	<b>6</b>
<b>5. Running FoUSB Programmer .....</b>	<b>7</b>
<b>6. Target Hardware Connectivity for In-System Programming.....</b>	<b>10</b>
<b>7. Target Hardware Connectivity for In-Circuit Debugging.....</b>	<b>14</b>
<b>8. Target MCU Resources Used by the ICD for In-Circuit Debugging.....</b>	<b>18</b>
<b>Appendix A. ICD Specifications.....</b>	<b>20</b>
<b>Appendix B. Technical Notes &amp; Limitations .....</b>	<b>21</b>
<b>Appendix C. Troubleshooting .....</b>	<b>22</b>
<b>Appendix D. Reference Manuals.....</b>	<b>27</b>
<b>Appendix E. Connecting to Legacy RS-232C Port .....</b>	<b>28</b>
<b>Appendix F. USB Board Schematics .....</b>	<b>29</b>

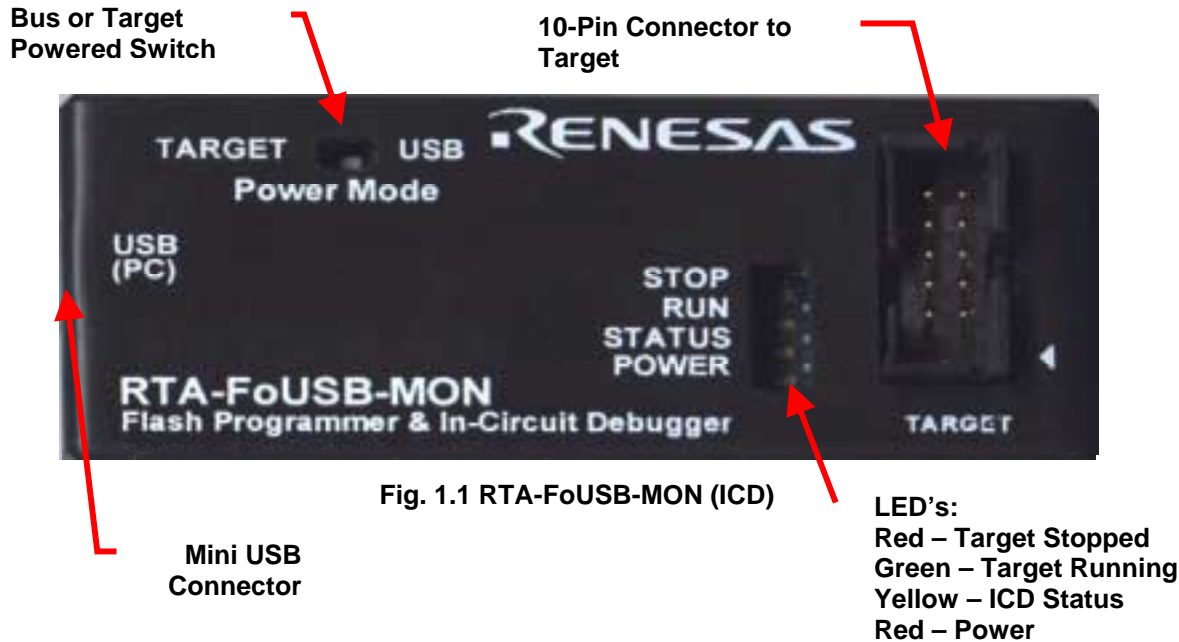


Fig. 1.1 RTA-FoUSB-MON (ICD)

## 1. Kit Overview

The RTA-FoUSB-MON (ICD) is a low cost compact device that can function as an in-circuit debugger (ICD) tool or as an FoUSB (Flash-over-USB™) Programmer for Renesas M16C Flash microcontrollers (MCU). KD30 is the software application for debugging M16C firmware and FoUSB Programmer is the software application for programming the Renesas M16C Flash MCU's.

Here are some of the ICD highlights:

- M37641F8HP (8-bit USB Renesas MCU) with 3.9KB RAM and 32KB Flash ROM.
- Bus-powered (powered by USB) or target-powered selectable. No external power supply needed when connecting to Renesas' SKP's (Starter Kits).
- Target connection through a 10-pin header with matching 6 in. cable.
- 4-pin port available for Renesas' MSV-RS232 cable connectivity.
- 8-pin port available for external interrupts and timers, which can be use for debug and development.
- Mini-USB connector with matching 6ft cable to connect to host PC.
- Semtech USB upstream port filter, EMI, and ESD protection circuitry. All in one device reduces target system part count and complexity while provides internal circuit protection.
- Works on Windows XP, 2000, Windows ME, and Windows 98.

### Package Contents:

- 1 – ICD
- 1 – 6ft USB Cable
- 1 – 6 in. 10-pin Flat Cable
- 1 – Software CD (Documentation, FoUSB Programmer, KD30 Debugger)

## 2. Software Installation

Before using the ICD, we need to install the required software files and applications.

- 2.1 Insert the enclosed CD ROM in PC's CD Rom drive.
- 2.2 The CD will auto start and the Renesas Starter Kit Plus welcome screen is displayed. If the screen does not come up, browse the root folder of the CD using Windows Explorer and double-click on '**SKP\_Installer.exe**'.
- 2.3 Click on 'RTA-FOUSB-MON' to start software installation. The install will create a new directory **C:\MTOOL\RTA-FOUSB-MON** on your PC where it will copy the necessary files for the ICD. During the install process, development tools installation will come up, as these have specific install programs, to setup software necessary for the ICD.
- 2.4 After installation, please review the Quick Start Guide, which may contain information about the ICD not yet available when this user's manual was published.
- 2.5 Click on "Finish" to exit install program.

If you experience problems with the install software, please see *C.1 Installing Required Files Manually* in Appendix C. Troubleshooting.

### 3. Driver Installation – Powering up the first time

Remove the ICD from the anti-static bag, place it device side up with the USB connector (J1) on your left hand side. The silk-screened writing will appear normal. Note the location of several key features on the board such as:

- 10/9 Connector (J3) – Provides connection to target board
- D1 LED – Power LED
- D2, D3, D4 – Displays status of the ICD and target MCU.

Before connecting the ICD to your PC, we need to understand the following terminologies, which are used when connecting the board the first time.

- Boot Mode – Shorting JP1 jumper in Fig. 1.1. Mode when the M37641F8HP is executing coded from boot ROM area.
- User Mode – Open JP1 jumper in Fig. 1.1. Mode when the M37641F8HP is running program from User ROM area.
- Drivers – Windows device drivers located in the 'drivers' folder on the accompanying disk
- Software – Windows applications supplied on the accompanying disk

Getting the board up and running is easy. Simply plug the board into an available USB port of a PC (running Windows XP, 2000, Windows ME, or Windows 98), install the drivers and Windows software, and that's it. The following is a step-by-step process on how this is accomplished using Windows 2000 Professional. Windows ME or Windows 98 will work similarly.

Plug the small end of the supplied USB cable into connector J1 located in the left side of the board (board laid out as seen on Fig.1.1). Note that the USB symbol of the cable will be facing up. Next, plug the other end of the cable into the target PC. A dialog box will appear like the following:



**Fig. 3.1 PC Detected the ICD**

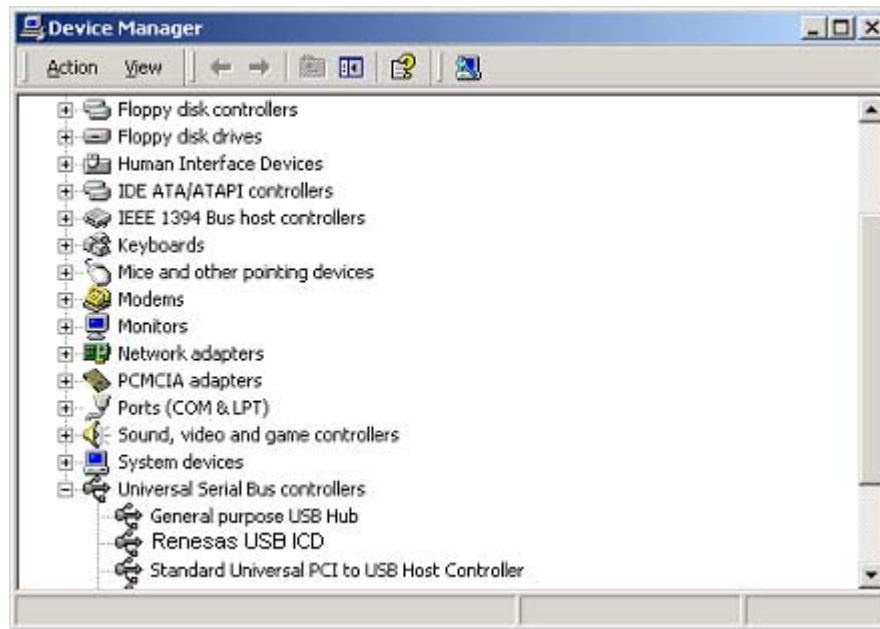
The newly attached device, Renesas USB ICD, has been discovered by the operating system. Drivers need to be loaded so Windows applications can use the device. The dialog box similar to the one shown in Fig. 3.2 will now appear. At this time, if you look at the ICD's Status LED (Yellow), it should be blinking at about 2-3 times a second (indicating that Windows has not recognized the device yet).



**Fig. 3.2 Driver Installation Dialog**

From this window, click on the 'Browse' button and locate **C:\WTOOL\FOUSB\USB Drivers** and then click the 'OK' button to complete driver installation. At this time, if you look at the ICD's Status LED, it should be blinking every second (indicating that the Windows successfully recognized the ICD).

To verify that the driver was installed properly, right click on the 'My Computer' icon on the Windows desktop. Click on the 'Hardware' tab and then click on 'Device Manager' button. Open the 'Universal Serial Bus Controllers' list and see that the ICD was loaded into the hardware table.



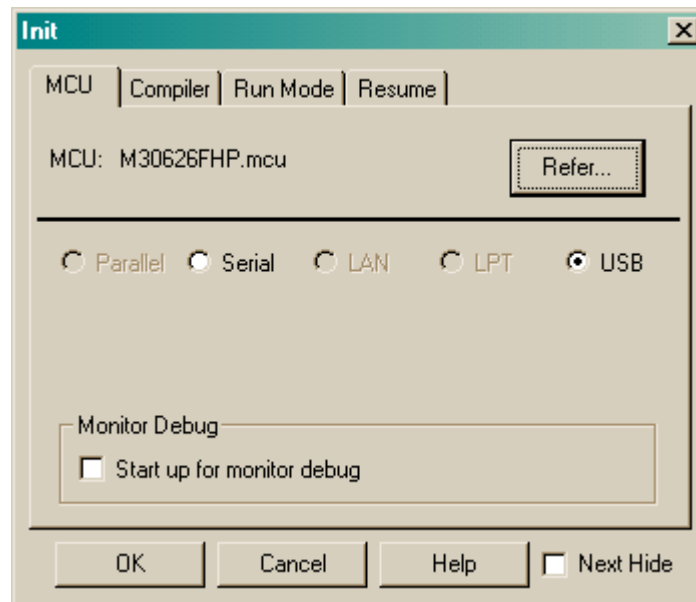
**Fig. 3.3 ICD Driver Loaded in PC Memory**

## 4. Running KD30 Debugger

Please note the hardware setup and software controls discussed here uses the RTA-M16C62P-SKP (RTA-M16C62P Board and ICD kit).

Before running the debugger, connect the target board to the ICD using the 10-pin flat ribbon cable. Connect one end of the mini-USB cable to the ICD and the other end to the PC. If you are using a Renesas SKP starter kit such as RTA-M16C62P-SKP, the ICD should be on bus-powered mode. If not, select target-powered mode and supply power on the target board.

To run the KD30 debugger, double click on kd30.exe shortcut in your desktop. An initial window, similar to one shown below, will be displayed.



**Fig. 4.1 KD30 Debugger Initialization Screen**

Select 'USB' option and click on 'OK' button. The KD30 screen should appear. Please see the KD30 Help on how to use the debugger (Start > Programs > Renesas-Tools > KD30 V.3.20 Release 1B >KD30 Help).

To close KD30, click on 'File' menu and select 'Exit'; or, click on the close button on the upper right corner of the Window. Please do not unplug or detach the ICD from your PC without closing KD30 first. Detaching the ICD while KD30 is still up may cause problems when you try to use KD30 again.

## 5. Running FoUSB (Flash-over-USB™) Programmer

Renasant Technology America, Inc. has created an easy to use Windows application for programming Renasant's M16C Flash MCU's. This application can be used along with the ICD to program the M16C MCU on a target board. To open the application, click on Start > Programs > Renasant-Tools > Flash-over-USB Ver. 1.x > FoUSB Programmer and a screen similar to Fig. 5.1 will be displayed. **Please ensure that you close KD30 first before running FoUSB Programmer.**

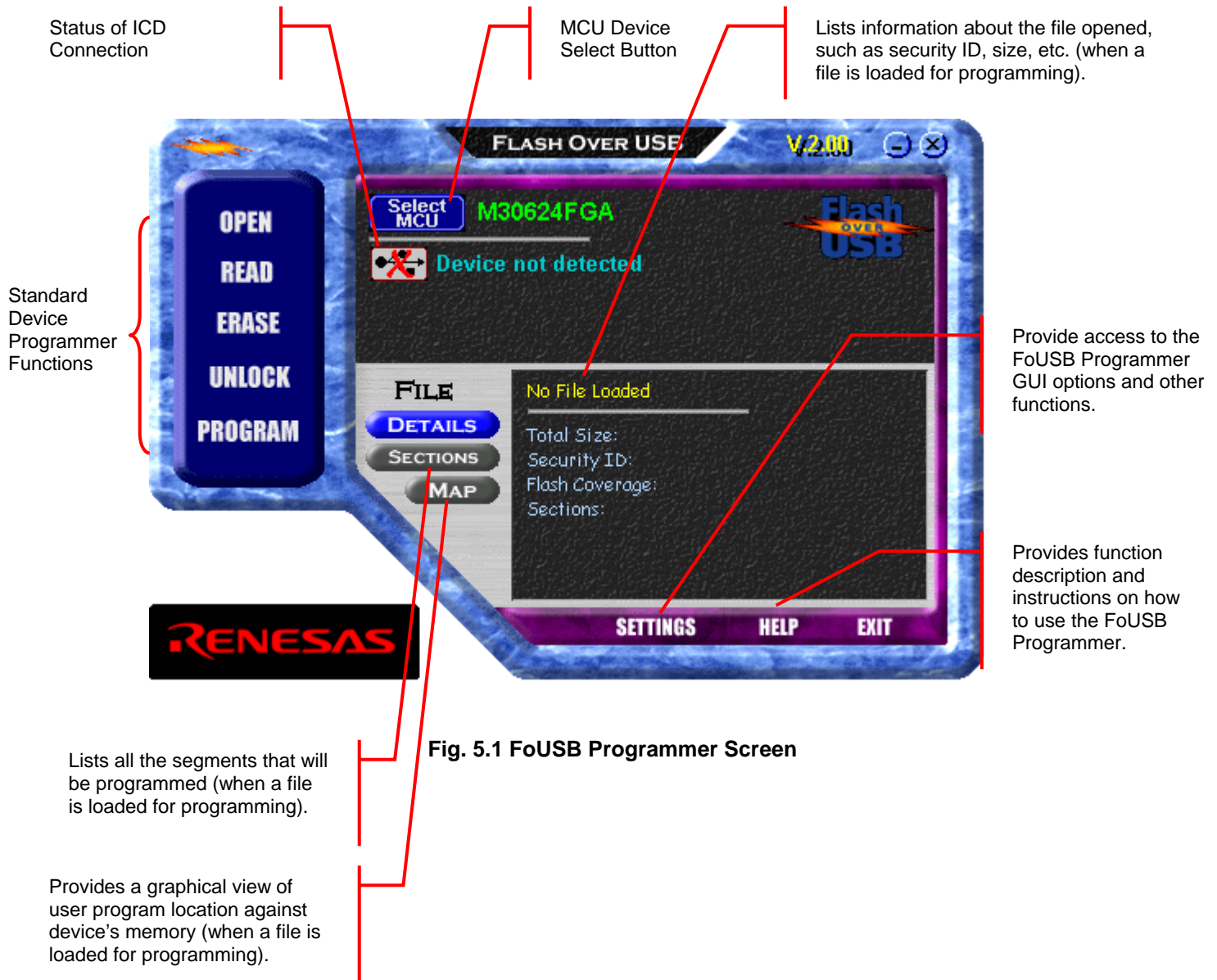


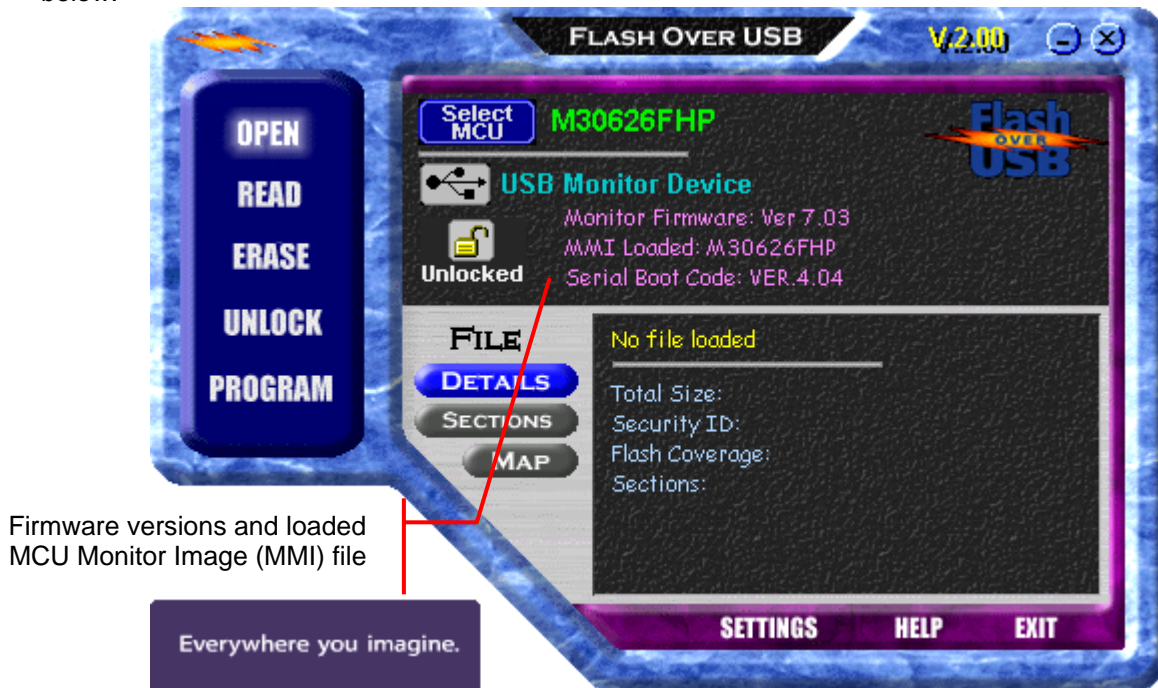
Fig. 5.1 FoUSB Programmer Screen

**NOTE:**

1. For additional descriptions and operations, please click on Help button of the FoUSB Programmer.
2. Renasant continues to update the application and the version number may be different from what is on this document.



An example of the FoUSB Programmer application with the RTA-M16C62P board connected is shown below.



**Fig. 5.2 Example of FoUSB Programmer Screen with Target Connected**

### **Basic FoUSB Programmer Operations**

The **Standard Device Programmer Functions** follow the basic file open, erase, program, ROM memory dump, and RAM memory read/write like other programmers.

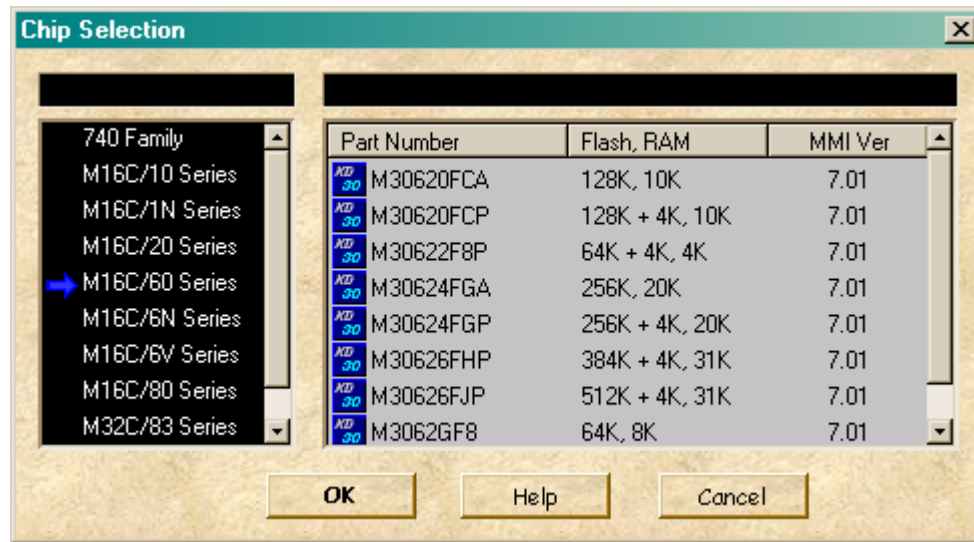
- **Open** allows you to select a firmware file you want to program to the target MCU.
- **Read** allows you to dump Flash memory and display it on a screen or write to a text file.
- **Erase** allows you to erase a block or the entire Flash memory.
- **Memory** allows you to read or write internal RAM memory including MCU SFRs.
- **Program** allows you to program or erase, program, and verify the target MCU.

The **Unlock** function used to “unlock” Renesas MCU’s for memory access. Renesas M16C MCU’s are equipped with a security feature, a 7-byte ID, to prevent firmware program disassembly. The user must enter the security ID information to be able to access Flash memory access. In a ‘Locked’ state, the only function you can use is the Unlock function.

An exception of the Unlock function is when the security ID is all zeroes (0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00) or all ones (0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF). For these two ID’s, the application automatically unlocks the device.

The **Command** button is used to send boot commands to the target MCU (when in boot mode). This button is used to debug monitor firmware and by device programmer manufacturers.

The **Select MCU** button allows the user to change FoUSB Programmer configuration to match the target MCU. When you click on this button, a list of currently supported devices will be displayed.



**Fig. 5.3 MCU Selection Screen**

If the selected MCU is different from the MCU on the ICD, the user will be prompted before re-programming the ICD.

The **Settings** button allows the user to configure the FoUSB Programmer application such as colors, etc.

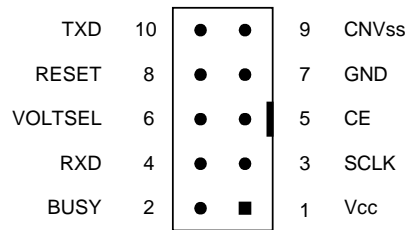
The firmware version and loaded MMI (MCU Monitor Image) file shows the following information:

- a. **Monitor Firmware:** displays the version of ICD (M37641) firmware that the board was programmed with.
- b. **MMI Loaded:** displays the MMI (MCU Monitor Image) file that the ICD firmware is using. This image file contains the information of the type of MCU the ICD is connected to. Different MCU devices use different image files (similar functionality to MCU files of the emulator).
- c. **Serial Boot Code:** displays the version of boot code of the target MCU.

## 6. Target Hardware Connectivity for In-System Programming

This section describes the hardware connections to the ICD for in-system programming purposes. Examples for different M16C MCU's are shown at the end of the section.

The ICD connects to an M16C target board through a polarized, .100" (2.54mm), 10-pin connection using a flat ribbon cable. It communicates with the M16C MCU using the boot UART/SIO in a clocked SIO configuration. The signals used and how these signals are assigned is described below.



**Figure 6.1 Target Board 10-Pin Header/Connector Signal Assignment**

**NOTE:**

All target interface lines are in high impedance state until instructed to do otherwise by the FoUSB Programmer application.

**a. Vcc – Power Supply Pin**

This provides the common connection point for power between ICD and the target board. This must be connected to the target board's Vcc line/plane.

In Target Power mode, the ICD requires at least 3.3V (5.0V max) and 45mA to operate properly.

**b. Busy – Target MCU SIO Busy Signal**

This line indicates whether the target is busy (and cannot receive data) or not. This must be connected to the target MCU's boot UART/SIO BUSY (CTS/RTS) pin.

This pin may be used for other applications as long as the user ensures that the ICD can drive the pin to valid levels with 5mA current.

**c. SCLK – Target MCU Serial Clock Signal**

This clock is used for synchronous serial communication between ICD and target board. This must be connected to the target MCU's boot UART/SIO SCLK pin.

This pin may be used for other applications as long as the user ensures that the ICD can drive the pin to valid levels with 5mA current.

**d. RXD –Target MCU Receive Data Input Signal (ICD Transmit Data Output Signal)**

This is the ICD SIO transmit data output. This must be connected to the target MCU's boot UART/SIO receive data input (RXD) pin.

This pin may be used for other applications as long as the user ensures that the ICD can drive the pin to valid levels with 5mA current.

## **e. CE Pin**

This pin is used to enter boot mode for programming of the target M16C MCU. Not all M16C MCU's have a CE pin. If a CE pin exists on the target MCU, it should be connected to this pin (refer to serial mode flash programming in the target MCU's datasheet). If not, leave pin unconnected.

If the target MCU's CE pin is used as input, isolate the driving circuit from the header connection with a 1Kohm resistor.

If to be left unconnected (if users decides to), care must be taken to ensure that the target MCU's CE pin has a high signal level using a pull-up resistor during any programming operations.

## **f. EPM Pin (in Target Power Mode)**

In Target Power Mode, the pin is used to control target MCU's EPM pin (if it exists on the target MCU). EPM pin is used during programming of the target MCU. If this pin exists on the target MCU, it should be connected to the header (refer to serial mode flash programming of the target MCU's datasheet). If not, leave unconnected.

If the target MCU's EPM pin is used as input, isolate the driving circuit from the header connection with a 1Kohm resistor.

If the EPM pin of the M16C MCU is left unconnected (no external circuit and not connected to 10-pin header), care must be taken to ensure that the target MCU's EPM pin has a high signal level by a pull-up resistor during any programming operations.

## **g. GND – Ground Pin**

This provides the common point where the ICD ground and target board ground planes can be connected. This must be connected to target board's GND line/plane.

## **h. RESET – Target MCU Reset Signal**

This active low signal is used by the ICD to reset the target MCU. This must be connected to the target board's reset circuit or target MCU's reset pin.

The user must ensure that the ICD can drive the pin to a valid low level with 5mA current.

## **i. CNVss – Target MCU CNVss Pin**

This active high signal is used by the ICD to control the target MCU's CNVss pin for programming purposes. This must be connected to the target MCU's CNVss pin. During normal operation, this pin must have a low signal. To accomplish this, connect a 10K (nominal) pull-down resistor. See M16C MCU datasheet on CNVss pin description for details.

## **j. TXD – Target MCU Transmit Data Output Signal (ICD Receive Data Input Signal)**

This is the ICD SIO receive data input. This must be connected to the target M16C MCU's boot UART/SIO transmit data output (TXD) pin.

This pin may be used for other applications as long as the user ensures that the ICD can drive the pin to valid levels with 5mA current.

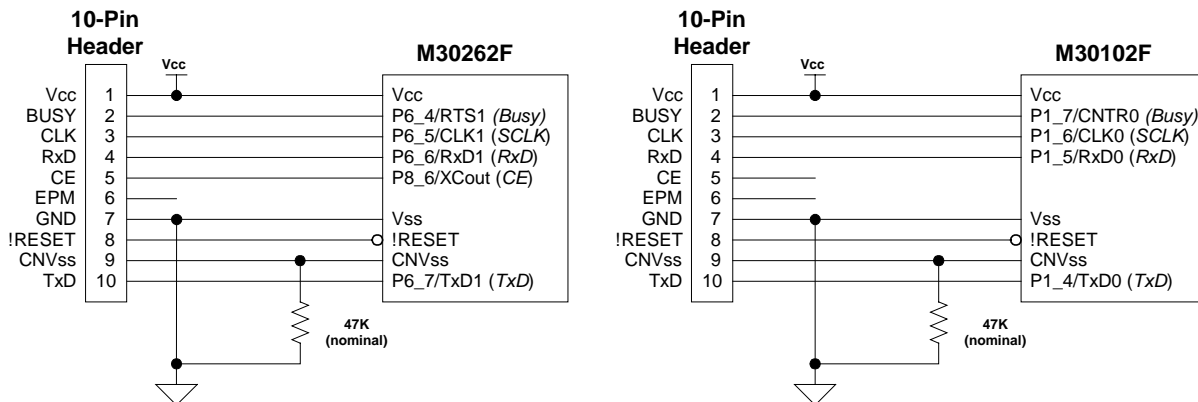
### k. Hardware Summary and Precautions

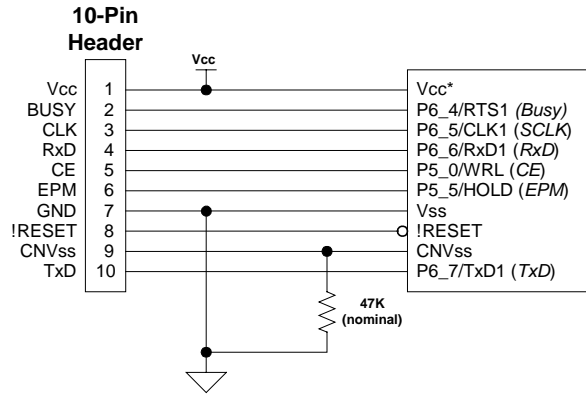
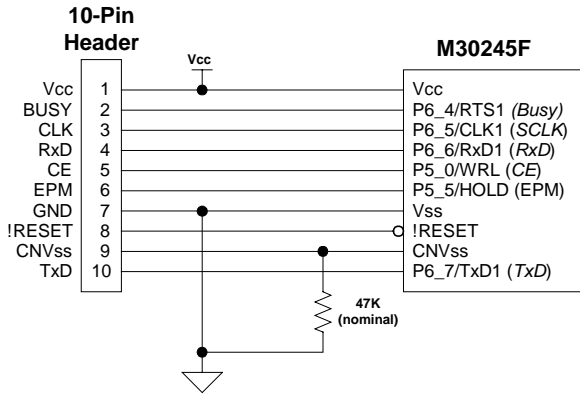
- Connect pull-down resistors to the CNVss line.
- Connect Reset line to the target MCU Reset pin or the target board's reset circuit.
- If EPM or CE lines are used as input pins, isolate the driving circuit from the 10-pin header using a 1Kohm resistor.
- If possible, avoid using the boot SIO/UART pins for other purposes. If not, ensure the lines can be driven in boot mode.
- Table 6.1 lists the M16C Flash MCU's supported and corresponding CE or EPM pin support and the boot UART/SIO.

**Table 6.1 M16C MCU Support Summary**

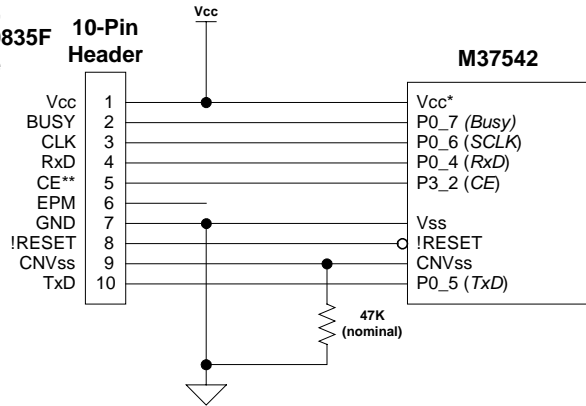
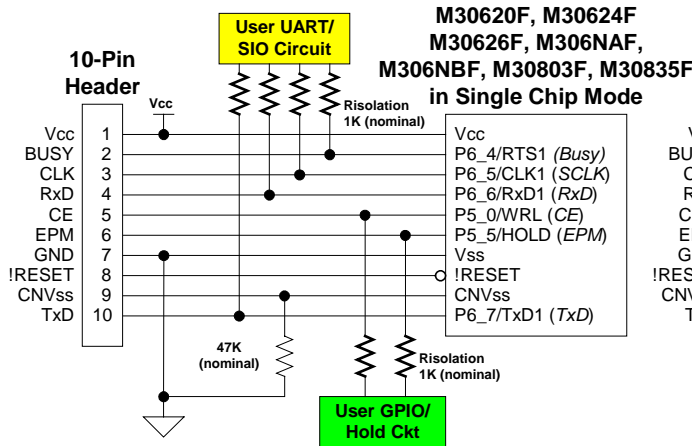
M16C MCU	CE/EPM Pin	Boot UART/SIO
M30100F3	None	UART0
M30245FC	Both	UART1
M30262F8	CE	UART1
M3062GF8N	Both	UART1
M30620FCA	Both	UART1
M30624FGA	Both	UART1
M30626FGP	Both	UART1
M306V7FG	Both	UART0

Examples of target board hardware connection to the 10-pin header (ICD), when used for in-system programming only, are shown below. If you cannot find your M16C Flash MCU in the examples shown, please contact Renesas representative for assistance.





\* For M30626F (M16C/62P), connect to Vcc1.



When only used in programming (i.e. no debug operation) and user uses boot UART, CE, and EPM pins, connect 1K (nominal) isolation resistors between user UART, CE, & EPM circuit and M16C MCU - 10-pin header connection.

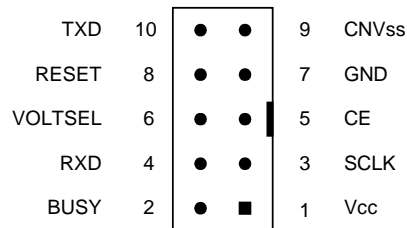
**NOTE:**

If you cannot find your M16C Flash MCU in the examples shown, please contact Renesas representative for assistance.

## 7. Target Hardware Connectivity for In-Circuit Debugging

This section describes the hardware connections to the ICD for in-circuit debugging operations. Examples for different M16C MCU's are shown at the end of the section. For target MCU resources used during in-circuit debugging, see reference section for details.

The ICD connects to an M16C target board through a polarized, .100" (2.54mm), 10-pin connection using a flat ribbon cable. It communicates with the M16C MCU using the boot UART/SIO in a clocked SIO configuration. The signals used and how these signals are assigned is described below.



**Figure 7.1 Target Board 10-Pin Header/Connector Signal Assignment**

**NOTE:**

All target interface lines are in high impedance state until instructed to do otherwise by the KD30 Debugger application.

**a. Vcc – Power Supply Pin**

This provides the common connection point for power between ICD and the target board. This must be connected to the target board's Vcc line/plane.

In Target Power mode, the ICD requires at least 3.3V and 45mA to operate properly. The ICD requires 5V when downloading new MCU Monitor Image (MMI).

**b. Busy – Target MCU SIO Busy Signal**

This line indicates whether the target is busy (and cannot receive data) or not. This must be connected to the target MCU's boot UART/SIO BUSY (CTS/RTS) pin.

This pin cannot be used while the Kernel program is running (during in-circuit debug operations).

**c. SCLK – Target MCU Serial Clock Signal**

This clock is used for synchronous serial communication between ICD and target board. This must be connected to the target MCU's boot UART/SIO SCLK pin.

This pin cannot be used while the Kernel program is running (during in-circuit debug operations).

**d. RXD –Target MCU Receive Data Input Signal (ICD Transmit Data Output Signal)**

This is the ICD SIO transmit data output. This must be connected to the target MCU's boot UART/SIO receive data input (RXD) pin.

This pin cannot be used while the Kernel program is running (during in-circuit debug operations).

## **e. CE Pin**

This pin is used to enter boot mode for programming of the target M16C MCU. Not all M16C MCU's have a CE pin. If a CE pin exists on the target MCU, it should be connected to this pin (refer to serial mode flash programming in the target MCU's datasheet). If not, leave pin unconnected.

This pin is in high impedance when in user mode (during debug operations).

If the target MCU's CE pin is used as input, isolate the driving circuit from the header connection with a 1Kohm resistor.

If to be left unconnected (if users decides to), care must be taken to ensure that the target MCU's CE pin has a high signal level using a pull-up resistor during any programming operations.

## **f. EPM Pin**

The pin is used to control target MCU's EPM pin (if it exists on the target MCU). EPM pin is used during programming of the target MCU. If this pin exists on the target MCU (and in Target Powered mode), it should be connected to the header (refer to serial mode flash programming of the target MCU's datasheet). If not, leave pin unconnected.

If the target MCU's EPM pin is used as input, isolate the driving circuit from the header connection using a 1Kohm resistor.

If the EPM pin of the M16C MCU is left unconnected (no external circuit and not connected to 10-pin header), care must be taken to ensure that the target MCU's EPM pin has a low signal level using a pull-down resistor during any programming operations.

## **g. GND – Ground Pin**

This provides the common point where the ICD ground and target board ground planes can be connected. This must be connected to target board's GND line/plane.

## **h. RESET – Target MCU Reset Signal**

This active low signal is used by the ICD to reset the target MCU. This must be connected to the target board's reset circuit or target MCU's reset pin.

The user must ensure that the ICD can drive the pin to a valid low level with 5mA current.

## **i. CNVss – Target MCU CNVss Pin**

This active high signal is used by the ICD to control the target MCU's CNVss pin for programming purposes. This must be connected to the target MCU's CNVss pin. During normal operation, this pin must have a low signal. To accomplish this, connect a 10K (nominal) pull-down resistor. See M16C MCU datasheet on CNVss pin description for details.

## **j. RXD – Target MCU Transmit Data Output Signal (ICD Receive Data Input Signal)**

This is the ICD SIO receive data input. This must be connected to the target M16C MCU's boot UART/SIO transmit data output (TXD) pin.

This pin must be pulled down with a resistor connected to GND. The target MCU's Kernel program polls this pin for the presence of the ICD before running the user program. Without the pull-down resistor, the target MCU will only run the Kernel program program.

This pin cannot be used while the Kernel program is running (during in-circuit debug operations).



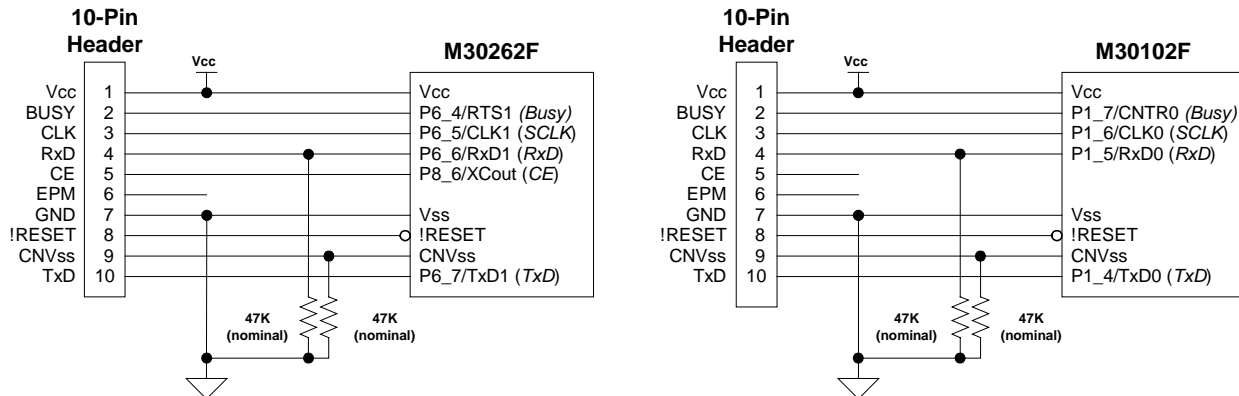
### k. Hardware Summary and Precautions

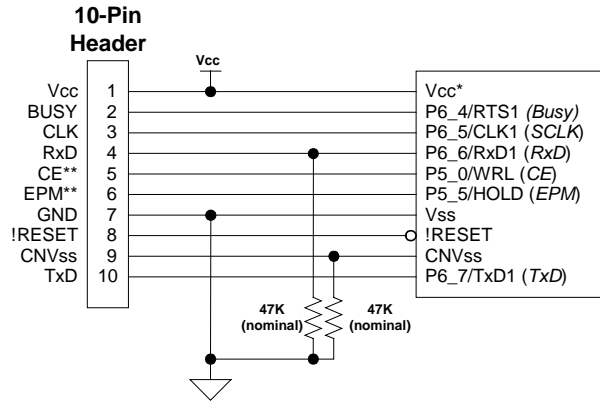
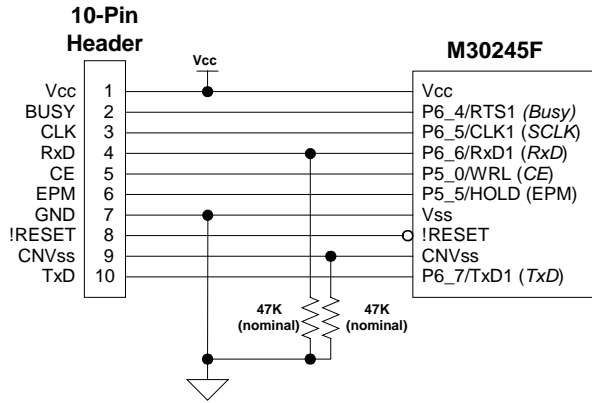
- Connect pull-down resistors to RXD and CNVss lines.
- Connect Reset line to the target MCU Reset pin or the target board's reset circuit.
- If EPM or CE lines are used as input pins, isolate the driving circuit from the 10-pin header using a 1Kohm resistor.
- If possible, avoid using the boot SIO/UART pins for other purposes.
- Table 7.1 lists the M16C Flash MCU's supported, CE or EPM pin support, the boot UART/SIO, DBC interrupt support, and the no. of breakpoints that can be set in KD30.

**Table 7.1 M16C MCU Support Summary**

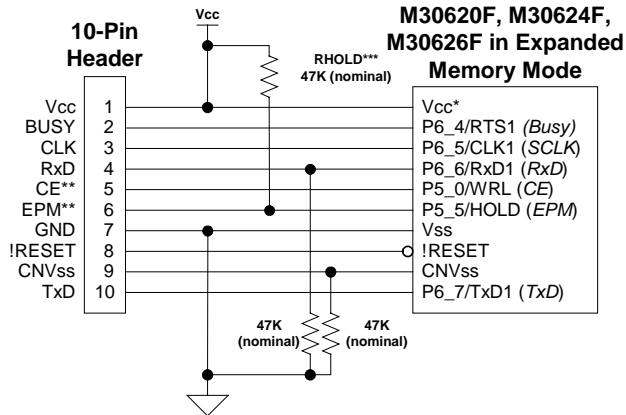
M16C MCU	CE/EPM Pin	Boot UART/SIO	DBC Irq Support	Breakpoints
M30100F3	None	UART0	No	2
M30245FC	Both	UART1	No	2
M30262F8	CE	UART1	Yes	6
M3062GF8N	Both	UART1	No	2
M30620FCA	Both	UART1	No	2
M30624FGA	Both	UART1	No	2
M30626FGP	Both	UART1	Yes	8
M306V7FG	Both	UART0	No	2

Examples of target hardware connection to the 10-pin header (ICD) when used for debugging purposes. If you cannot find your M16C Flash MCU in the examples shown, please contact Renesas representative for assistance.

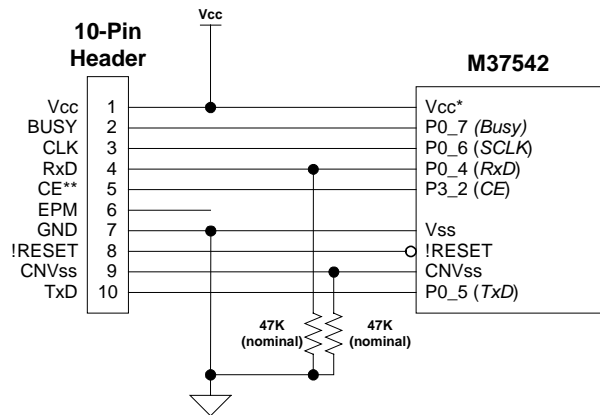




\* For M30626F (M16C/62P), connect to Vcc1.  
\*\* CE and EPM pins are high impedance in user mode (during debug operations).



\* For M30626F (M16C/62P), connect to Vcc1.  
\*\* CE and EPM pin are high impedance in user mode (during debug operations).  
\*\*\* HOLD/EPM pin must be pulled high in memory expansion mode.



**NOTE:**

If you cannot find your M16C Flash MCU in the examples shown, please contact Renesas representative for assistance.

## 8. Target MCU Resources Used by ICD for In-Circuit Debugging

This section describes the M16C MCU resources used by the ICD for in-circuit debugging (with the KD30 Debugger). These resources are NOT used when the ICD is used for programming (with the FoUSB Programmer). See previous sections for details on how to connect the hardware for in-circuit debugging or in-system programming support.

When used as debug interface, the ICD downloads a small program, called a kernel program (or ROM Monitor), to the target M16C MCU to establish communications used for program debugging operations. This kernel program is downloaded to the target MCU when KD30 application is started.

It is important to note that the operation of the ICD is transparent to the user. There is no special software processing required to be able to use it as long as the user understands the exceptions and follows the rules described below.

### NOTE:

**These resources are ONLY used when the ICD is used for in-circuit debugging. Only the MCU pin resources are used when the ICD is used for in-system programming.**

#### a. ROM

The kernel program code is located in the M16C memory space starts at address 0FF900H to 0FFE80H. KD30 will display a message on any attempt to overwrite specified ROM location. Rules on ROM usage are described below.

- User memory allocated to the Kernel program should not be used. Any attempt to use this area by the user code will be ignored.
- The special page vector table is mapped from 0FFE00H to 0FFFDBH. It is suggested that, if special page jumps are used, the special page numbers start with the minimum value of 20 and increases to a maximum of 127 to accommodate future monitor size increases.

#### b. RAM

The kernel program utilizes 128 bytes of RAM memory located at the top of MCU RAM area. For example, on an M30624 MCU, there is 20Kbytes of RAM located from 00400H to 053FFH. In this case, the monitor will be using the addresses from 05380H to 053FFH. In addition, the monitor also uses 4 bytes (max) of RAM on the interrupt stack.

#### c. Interrupts

For all MCU's that do NOT support DBC interrupt, the kernel program requires that the receive interrupt vector for the monitor/boot UART **MUST** point to the communications entry point of the monitor, which is 0FF900H. In addition, the monitor reserves the following hardware interrupt vectors for its own use:

- BREAK vector (0FFFE4H)
- ADDRESS MATCH vector (0FFFE8H)
- SINGLE STEP vector (0FFFECH)
- DBC vector (0FFFF4H)

The user code download process will properly load values into these locations. User code that globally disables interrupts for long periods of time (>1s), may interfere with the proper operation of the monitor.

### NOTE:

**For MCU's that do NOT have DBC interrupts, always ensure that the global interrupt is enabled in the startup files. Communications between ICD and the kernel program will fail when the global interrupt is not set or disabled for long periods (>1s). This is the primary cause of communication errors while using the KD30 Debugger.**

**Table 8.1 M16C MCU Support Summary**

M16C MCU	Boot UART/SIO	DBC Irq Support	Breakpoints
M30100F3	UART0	No	2
M30245FC	UART1	No	2
M30262F8	UART1	Yes	6
M3062GF8N	UART1	No	2
M30620FCA	UART1	No	2
M30624FGA	UART1	No	2
M30626FGP	UART1	Yes	8
M306V7FG	UART0	No	2

#### d. Peripherals

The kernel program uses the serial port associated with the boot mode of the MCU for communication with the ICD. Connect the SCLK, TxD, RxD signals from the 10-pin header to this serial port. For example, in the case of M30624 MCU, the monitor uses UART1 as the boot UART.

For MCU's that do NOT support DBC, ensure that the interrupt vector for the boot UART is set to FF900h.

Avoid byte-writes to the (boot UART/SIO) port, including direction register, associated with the SIO when the other half of the port is used as General Purpose I/O (GPIO) pins. Byte-writes to the port and direction register may prevent the monitor program from functioning properly. Use read-modify-write instructions instead.

If possible, avoid using this serial port for other applications in the firmware when the ICD will be used for development.

#### e. Real-Time Capability

Please be aware that while the kernel program is in its "STOP"ed state, the hardware peripherals will continue to run. Therefore, interrupts may not be serviced properly. Also, the watchdog timer will not be serviced and will likely timeout if active.

#### NOTE:

*While the kernel program is in its "RUN"ning state, there is no overhead on the application code, UNLESS a RAM monitor window is open in KD30. This window requires periodic communication with the target MCU, which suspends normal application operation while servicing the request (approximately 2000 BCLK cycles for each 16 bytes of data displayed in the window are used per window update). The user must determine whether or not, this behavior is acceptable.*

#### f. Software Summary and Precautions

- Ensure that global interrupt is set in the startup file and not disabled for long periods (>1s).
- Do NOT use the RAM or ROM memory as specified in sections 8.a and 8.b.
- Avoid byte-writes to the port and direction register to avoid interfering with monitor operation associated with the boot SIO/UART. Use bit manipulations.
- Do NOT change boot UART configuration.
- Ensure that the boot SIO/UART vector is properly set-up to point to FF900H for those M16C MCU's that do NOT support/have DBC interrupt.
- Do NOT attempt to debug watchdog, WAIT mode, or STOP mode.
- Do NOT execute a 'STEP' operation after a write to PRC2 SFR.

## Appendix A. ICD Board Specifications

The following table lists the specifications for the ICD Board.

Table A.1 ICD Specification

Item	Description
<b>USB MCU</b>	Renesas M37641F8HP (8-bit USB Flash MCU)
<b>Interface</b>	PC host and target connectivity.
a. USB (J1)	Full speed USB connection and USB 2.0 compliant. Primary connection to debug PC.
b. UART Interface (J2)	4-pin header for RS-232C PC connection (TTL Level -RS-232C transceiver/circuit is mounted externally).
c. Target Interface (J3)	10-pin connection with starter kit board or target board. Please read 'Target Board Connectivity' on how to design the target hardware to connect the ICD.
<b>LED</b>	4 LED's: 2 red, 1 green, 1 yellow
a. Red (D1)	Power LED.
b. Red (D2)	Lights up when target MCU is stopped.
c. Green (D3)	Lights up when target MCU is running.
d. Yellow (D4)	ICD ready status – blinking
<b>Switch, Jumpers</b>	Provide additional debug pins, microprocessor mode and power supply selection.
a. Bus Power (S1)	Power selection: Bus powered or target powered.
b. MCU Mode (JP1)	CNVss pin for boot mode (shunted) or user mode (open).
c. Debug (JP3)	Provide debug pins for external interrupts (INT0, INT1), monitor status, counters 0 & 1 output pins.
<b>Power Configuration</b>	The ICD will operate either at 5V or 3.3V and may be bus powered (from USB) or target powered. 5V/3.3V selection using pin 6 of J3 connection to target.
<b>Current Requirements</b>	The maximum current draw by the ICD is 50mA.
<b>Board Size</b>	2.75" x 1.15" (70mm x 29 mm)

**Appendix B. Technical Notes & Limitation****A. MSV1632-SKP Board**

When using the ICD with the MSV1632-SKP board, connect a 10KOhm pull-down resistor on JP9's (UART1) RXD line to be able to run user code. The monitor program polls the RXD line for the presence of the ICD before running the user program. Without the pull-down resistor, the M16C/62 MCU will only run the monitor program.

## Appendix C. Troubleshooting

This section discusses possible problems you may encounter while installing the software (and drivers) and while running the KD30 or FoUSB Programmer applications. This section also discusses the countermeasures and solutions to resolve these problems.

If, for any reason, you cannot resolve the problem, please contact your Renesas representative for assistance.

### C.1 Installing Required Files Manually

Before connecting the ICD to your PC, the driver files (inf and sys) files and executables must be copied to the C:\MTOOL\FOUSB directory.

To do this, run FoUSB\_Vx.xx.exe under \Tools\FoUSB directory on the SKP CD. After the FoUSB Programmer install, if the default directory was used during install, a C:\MTOOL\FoUSB subfolder should have been created. The Windows USB drivers can be found under the USB Drivers folder, i.e. fousb.inf, fousb.sys (driver files to run FoUSB.exe), usbmon.inf, and usbmon.sys (driver files to run KD30).

The C:\MTOOL\RTA-FOUSB-MON folder contains all latest documentation about the ICD from Quick Start Guide to schematics. The development tools can be found under *tools* folder from the CD root. The *tools\kd30* folder contains the KD30 installer executable.

*NOTE: If you are using Windows 2000 or XP, you will need Administrator privileges to be able to install the drivers.*

- (1) Windows 2000
  - (a). Install FoUSB Programmer by double-clicking on fousb.exe from the \Tools\FOUSB folder of the CD.
  - (b). Copy the fousb.inf and usbmon.inf files from C:\MTOOL\FOUSB\USB Drivers folder to \WINNT\INF folder.
  - (c). Copy the fousb.sys and usbmon.sys files from C:\MTOOL\FOUSB\USB Drivers folder to \WINNT\SYSTEM32\drivers folder.
  - (d). Install KD30 by double-clicking on KD30V3R1\_e.exe in *tools\kd30* folder, which can be found on the CD root. It will create \Mtool\KD30 directory on your PC.
  - (e). Copy communi.dll and kd30dll.def from \Tools\ICD\KD30V3R1 folder of the CD to the C:\Mtool\KD30 directory.
  - (f). Right-click on kd30.exe file on C:\Mtool\KD30 directory and select 'Send to Desktop (create a desktop shortcut).
  
- (2) Windows 98
  - (a). Install FoUSB programmer by double-clicking on fousb.exe from the \Tools\FOUSB folder of the CD.
  - (b). Copy the fousb.inf and usbmon.inf files from C:\MTOOL\FOUSB\USB Drivers folder to \WINDOWS\INF folder.
  - (c). Copy the fousb.sys and usbmon.sys files from C:\MTOOL\FOUSB\USB Drivers folder to \WINDOWS\SYSTEM32\drivers folder.
  - (d). Install KD30 by double-clicking on KD30V3R1\_e.exe in *tools\kd30* folder, which can be found on the CD root. It will create C:\Mtool\KD30 directory on your PC.
  - (e). Copy communi.dll and kd30dll.def from \Tools\ICD\KD30V3R1 folder of the CD to the C:\Mtool\KD30 directory.
  - (f). Right-click on kd30.exe file on Mtool\KD30 directory and select 'Send to Desktop (create a desktop shortcut).

## C.2 Driver Problems

This part discusses the usual problems with the driver installation and how to fix it. The most common problem encountered is, Windows did not properly install the driver and so the ICD is not recognized. This may also cause the device status to indicate that the device is not working properly. An indication of this problem is the ICD's Status LED (yellow), it should be blinking at about 2-3 times a second. When the driver is installed properly, the Status LED only blinks every second.

Before trying the following steps, try re-starting your PC to see if this resolves the problem. You can check the status using the Device Manager. If the ICD appears under the Universal Serial Bus Controllers with NO red X or yellow exclamation point, the driver was installed properly.

For cases where the 'Device Status' states the device is not working properly, please try the following:

1. Double-click on 'Renesas USB ICD' and a Renesas USB ICD Properties dialog box appears.
2. Click on 'Driver' tab and click on 'Update Driver' button.
3. Select 'Display a list...' and click on 'Have Disk' button.
4. Locate the C:\MTOOL\FOUSB\USB Drivers folder and install 'usbmon.sys' driver.
5. If this process does not work, please follow instructions below.

For cases where in the driver was not installed properly by Windows (Windows 98, Windows 2000) or not listed in the Device Manager > Universal Serial Bus Controllers, please try the following:

1. Unplug the USB Cable so Windows removes driver from memory.
2. Delete the driver 'usbmon.sys' from '\WINNT\SYSTEM32\DRIVERS\' folder in Windows 2000 or \WINDOWS\SYSTEM32\DRIVERS folder in Windows 98.
3. Plug in ICD + target and try installing the driver as written above and use the driver from the C:\MTOOL\FOUSB\USB Drivers.



## C.3 KD30 Problems

The common problems encountered with KD30 are: USB cannot be selected on the initialization screen, cannot connect to target, and KD30 already exist. This part discusses the cause of the problem and the countermeasures to resolve it.

When USB cannot be selected, you have an old version of KD30. Un-install the current version and delete the KD30 folder under C:\MTOOL. Run the latest KD30 install from the SKP CD or download the file from the Renesas Tools website.

When the message 'Can't connect with the target' is displayed, there are several reasons that may cause this message to appear. We'll discuss each cause and the corresponding countermeasure.

- (1) The ICD is not connected to the PC. Please connect the ICD to your PC.

*NOTE:*

*Regardless of whether the ICD is bus or target powered, please connect target board to ICD first before plugging the USB cable to your PC.*

- (2) The ICD to target cable is not connected. Please connect the ICD to the target board using the 10-pin flat ribbon cable.
- (3) There is insufficient bus power to run the ICD and target board. Bus power can supply up to 100mA to both ICD and SKP boards. Renesas SKP boards were designed so that the combination of ICD and SKP board will be under 100mA limit. If your target board is a non-Renesas SKP board, change S1 to TRGTPWRD and connect an external power supply to your target board.
- (4) The target MCU programmed on the ICD during KD30 initialization does not match the actual target MCU. Close the error message by clicking on 'OK' button and then click on 'Cancel' button of the KD30 Init window to close KD30. Select the correct MCU from the Refer button of the KD30 Init screen.
- (5) Hardware connection on the target board does not follow the convention on how to connect the ICD and target board. Please review the connection and check your circuit whether there are signals that have been wired to the wrong pin.
- (6) The target MCU is damaged. Try a different target board and see if KD30 will come up as you may have a damaged MCU.

When a message 'KD30 already exists' is displayed, the usual reason for this problem is that KD30 application was not properly closed. Please check if KD30 is already running by looking at your task bar. If KD30 cannot be found there, bring up Task Manager (for Windows 2000, right-click on Task Bar and select 'Task Manager'; for Windows 98 or ME, press CTRL-ALT-DEL once and to display the 'Task Manager'). Select KD30.exe on the 'Processes' list and click on 'End Process' to terminate KD30.

## C.4 FoUSB Programmer

This section describes the problems, the causes, and countermeasures when working with the FoUSB Programmer. The common problems with FoUSB Programmer are: FoUSB Programmer does not come up, device is not detected, or target is not connected.

When FoUSB Programmer does not come up properly, the main reason is that the MCU text file list folder does not exist on the folder where the FoUSB Programmer executable is. The MCU text file is a list of all MCU's that the FoUSB Programmer supports. Please read 'Installing Required Files Manually' on running fousb.exe to fix the problem.

If you encounter a 'device is not detected', the possible causes and countermeasures are listed below.

- (1) The ICD and target board is not connected to the PC. Please plug one end of the USB cable to the ICD and the other end to the PC. Make sure you connect the ICD to the target board first before plugging the USB cable to the PC.
- (2) The FoUSB Programmer driver (fousb.inf and fousb.sys) was not installed properly. Please install the drivers manually, please read 'Installing Required Files Manually'.
- (3) The ICD has no user ROM firmware. Please read 'Using the ICD Boot Mode' to program the user ROM firmware.

If you encounter a 'target is not connected', the possible causes and countermeasures are listed below.

- (1) The target board is not connected to the ICD. Please unplug the USB cable first and then connect the target board to the ICD with the 10-pin flat ribbon cable before plugging back the USB cable to your PC.
- (2) The target MCU is not a flash version. The ICD can only talk to M16C flash MCU's as these MCU's have boot program that the ICD communicates with. Please verify if the target MCU is a flash part.
- (3) Hardware connection on the target board does not follow the convention on how to connect the ICD and target board. Please review the connection and check your circuit whether there are signals that have been wired to the wrong pin.
- (4) The target MCU is damaged. Try a different target board and see if a target connection can be established. You may have a damaged MCU on your target board.

### C.5 Using the ICD's Boot Mode

This section discusses how to re-program the ICD if for some reason the user ROM firmware was erased or damaged.

The ICD has a boot mode that can be used to program the user ROM area. When the ICD was shipped, the user ROM firmware has been pre-programmed with a firmware that can communicate with the FoUSB Programmer. However, in the event that this firmware got erased or damaged, you can use this boot mode to recover the ROM firmware.

- (1) Unplug the USB cable and disconnect the target board from the ICD.
- (2) Open the ICD. Renesas recommend that you open from the USB connector side.
- (3) Shunt JP1 with a 2.54mm (.100 mil) jumper and shunt or a bent paper clip. This will then configure the ICD to run in boot mode when it is powered up.
- (4) Plug the USB cable to the PC. Ensure that the target board is not connected at this time. A new hardware found window would come up showing 'Flash-over-USB' and installs the necessary drivers. If Windows requests for the drivers, select 'Specify a location', locate C:\MTOOL\FoUSB\USB Drivers folder, and follow Windows instruction on driver installation. If the drivers were already installed before, this will not appear.
- (5) Open FoUSB Programmer and it should come up as M37641F8 FOUSB Device. If it does not come up, please see 'Troubleshooting, FoUSB Programmer'.
- (6) Open FoUSB Programmer and it should come up as M37641F8 FOUSB Device. If it does not come up, please see 'Troubleshooting, FoUSB Programmer'.
- (7) Click on 'Load MMI' button and select the MCU from the Chip Selection screen.
- (8) After a successful download, a message is displayed. Unplug the ICD from your PC and remove the JP1 shunt. This completes the recovery process.
- (9) If you plug the ICD to the PC, the FoUSB Programmer should show the target MCU device name you selected earlier. If for some reason, it still does not come up, your ICD may have been damaged. Please contact your Renesas representative to get a replacement.

## Appendix D. Reference Manuals

Item	Title	Description
1.	<b>RTA-FOUSB-MON Quick Start Guide</b>	Document that will help you get started on using the ICD.
2.	<b>RTA-FOUSB-MON User's Manual</b>	Installation and operation guide for the ICD.
3.	<b>ICD Board Schematic</b>	Schematic diagram for the ICD board.
4.	<b>ICD Board BOM</b>	Bill of materials for the ICD board.
5.	<b>M16C/20/60 Series C-Language Programming Manual</b>	ANSI C-language programming guide for the M16C/20/60 series MCU.
6.	<b>M16C/20/60 Series Assembler Language Programming Manual</b>	Assembler language programming guide for the M16C/20/60 series MCUs.
7.	<b>M16C/20/60 Series Software Manual</b>	This document details the instruction set and timing information for the M16C/20/60 series CPU cores.
8.	<b>M16C/20/60 Series Sample Programs</b>	This document details some sample programs and application notes for the M16C/20/60 series MCUs.
9.	<b>Tool Manager User's Manual</b>	This document describes installation and operation of this Integrated Development Environment for Renesas' Tools.
10.	<b>KD30 User's Manual</b>	This document describes the operation of this easy-to-use source and assembly debugger.
11.	<b>AS30 User's Manual</b>	Assembler Language Guide for AS30 assembly language compiler.
12.	<b>NC30 User's Manual</b>	ANSI C-Compiler Guide for KNC30WA C-language compiler.

**NOTE:**

The RTA-FOUSB-MON installer will copy all these manuals and can be accessed using the Document List web page by clicking on the Start > Programs > Renesas-Tools > RTA-FOUSB-MON > Document List shortcut.

## Appendix E. Connecting to Legacy RS-232C Serial Port

This section shows how to connect the ICD to a computer's RS-232C ports using Renesas' MSV-RS232 cable. This connection will only allow you to use the ICD as a debugger but not a programmer.

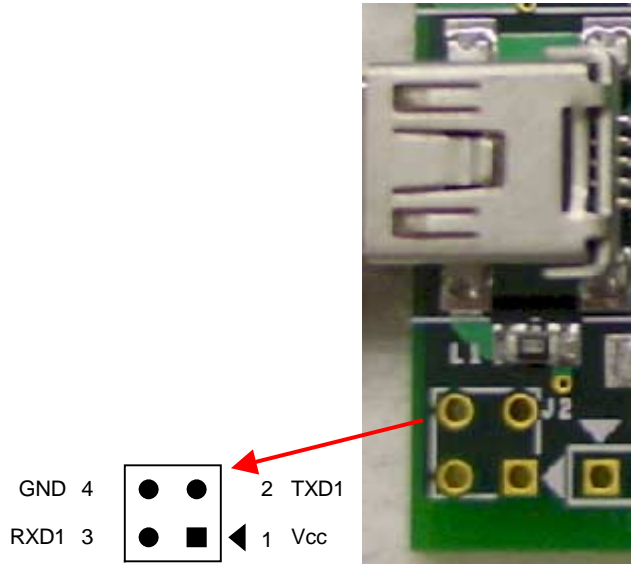


Fig. E.1 J2 Pin Assignment

MSV-RS232  
10-Pin Header

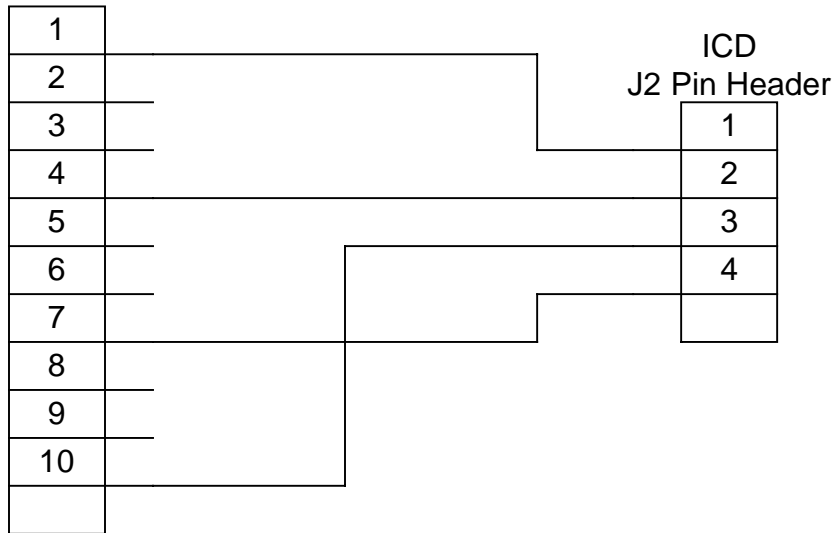


Fig. E.2 Connecting the ICD to an MSV-RS232 Cable

**NOTES:**

An external power supply is required to power the ICD (and MSV-RS232 cable) when connecting the ICD to an RS-232C port.

On KD30, the Baud Rate must be set to 38400 to be able to communicate with the ICD.

## Appendix F. USB Monitor Schematics

