

# How to Use Boot Assist Module (BAM) on TWR-PXR40 Board

by: **Antonio Pintor**  
**Automotive and Industrial Solutions Group**

## Contents

|   |  |    |
|---|--|----|
| 1 | Introduction.....                                    | 1  |
| 2 | Boot assist module (BAM).....                        | 1  |
| 3 | RAppID Boot Loader utility.....                      | 5  |
| 4 | Tower system setup.....                              | 6  |
| 5 | Firmware download using RAppID Boot Loader tool..... | 10 |
| 6 | Conclusion.....                                      | 13 |

## 1 Introduction

Boot loader is a very powerful tool to load new firmware to a microcontroller. Boot loaders can be used for factory programming and firmware updates without having access to the program/debug pins of the microcontroller. The PXR40 microcontroller has a special module called the boot assist module (BAM) that can be used to download code to the microcontroller just after coming out of the factory.

This application note will briefly explain how to use RAppID Boot Loader tool on the TWR-PXR40 board. This procedure can be duplicated on a custom board leaving the correct lines assigned to select between the normal boot and the BAM boot.

## 2 Boot assist module (BAM)

Boot loader is a small program running on the microcontroller that allows to get a binary file from the external world and store it into the flash memory of the microcontroller. It must be noted that the BAM is not a code boot loader but provides a method to download a small program into the RAM of the microcontroller and execute it. This code can be a firmware loader or anything that the user wants to store and execute from RAM after power-on sequence.



## Boot assist module (BAM)

The BAM is a 4-KB block of read-only memory, containing the boot program code for the MCU. The BAM program supports the following boot modes:

- Boot from internal
- Serial boot via SCI or CAN interface with optional baud-rate detection

The BAM program is executed by the MCU core just after the MCU resets. Depending on the boot mode, the program initializes the appropriate minimum MCU resources to start user application code. The features of the BAM are as follows:

- Initial MCU core MMU setup with minimum address translation for all internal MCU resources
- MMU configuration to boot user application, compiled as Classic PowerPC Book E code or as Freescale VLE code
- Passes control to user application code in the internal flash memory
- Automatic switch to Serial Boot mode if internal flash is blank or invalid
- Serial boot by loading user program via CAN bus or eSCI to the internal SRAM
  - User programmable 64-bit password protection
  - Optional automatic detection of the host SCI or CAN speed
- Controls MCU core Watchdog Timer and/or the Software Watchdog Timer (SWT)

The following figure depicts the workflow of the boot loader application using the BAM.

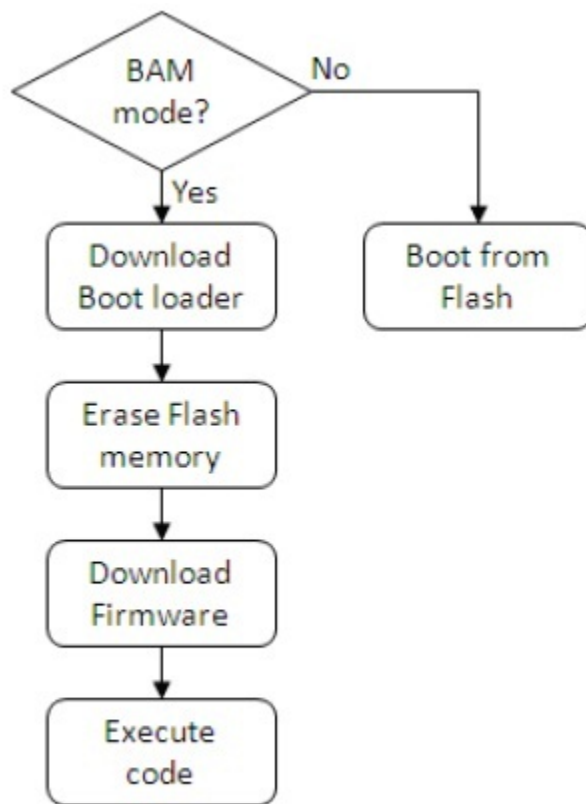


Figure 1. BAM boot loader application flow

## 2.1 Modes of operation

## 2.1.1 Normal mode

The BAM program is executed immediately following the negation of RESET.

## 2.1.2 Debug mode

The BAM program is not executed when the MCU exits reset in OnCE Debug mode. Before accessing the MCU resources, use the development tool to initialize the MCU.

## 2.1.3 Internal Boot mode

Use this operating mode to boot from internal flash memory, which holds all the codes and configuration data.

## 2.1.4 Serial Boot mode

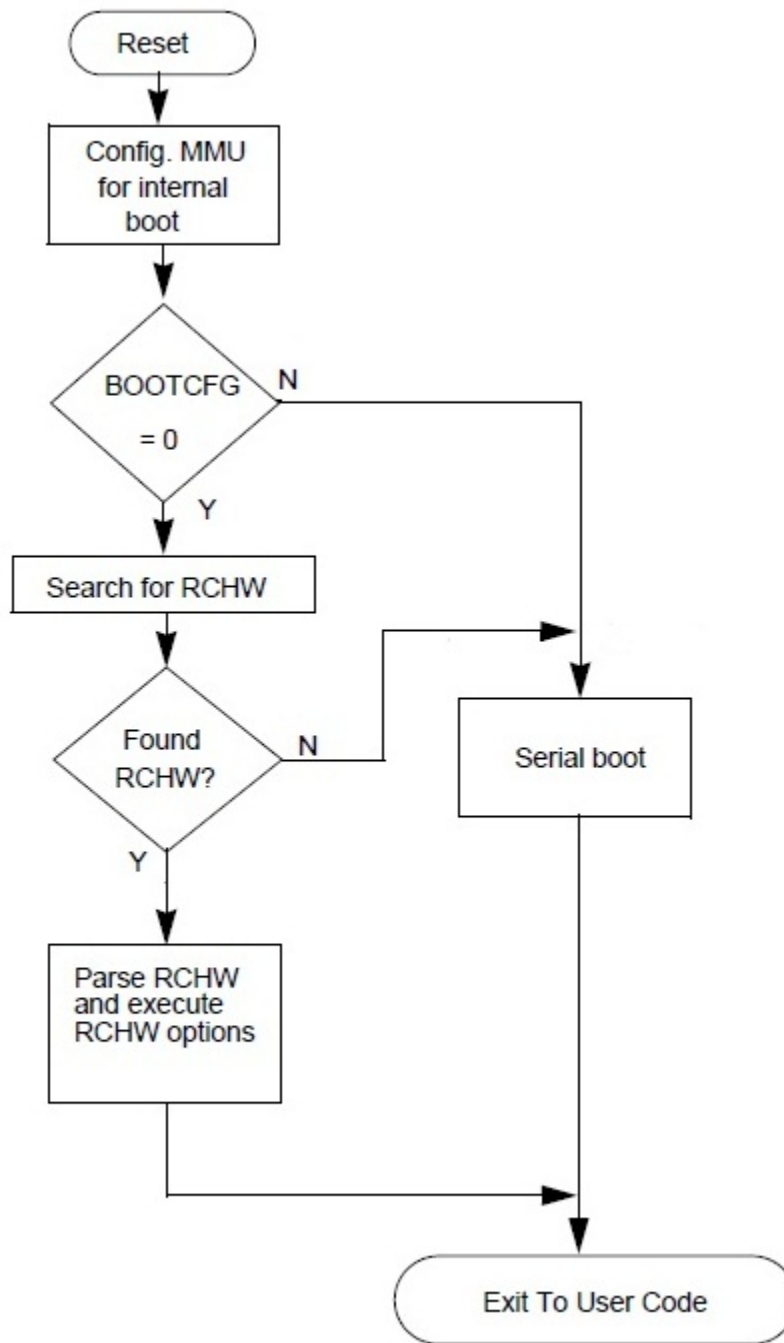
Use this mode to load a user program into internal SRAM, using the eSCI or CAN serial interface, then execute that program.

The program can then control

- the download of data, and
- erasing and programming of the internal flash memory

## 2.2 Executing BAM in Serial Boot mode

The following figure shows the BAM program flow.



**Figure 2. BAM program flow chart**

After negation of RESET, the MCU core accesses the BAM before user code starts. First, the BAM program configures the core Memory Management Unit (MMU) to allow access to all MCU internal resources and then the BAM program starts the boot sequence depending on the BOOTCFG value.

Serial Boot mode is executed if BOOTCFG is asserted (see [Table 1](#)) or when a valid RCHW word is not found in any of the expected locations in the flash.

The Serial Boot mode can run in two modes, depending on the state of the EVTO pin during reset:

- Standard Serial Boot mode — fixed baud rates derived from the MCU system frequency. EVTO pin is kept high during reset.
- Baud Rate Detection Serial Boot mode — allows communication with adaptable speed, based on measured baud rate of the host transmission. EVTO pin has to be driven low for this mode.

The EVTO pin is pulled up by an internal pull-up. Actively driven low EVTO pin selects the Baud Rate Detection mode; this mode is selected for using RAppID Boot Loader, see [Table 1](#).

**Table 1. Boot mode configuration**

| BOOTCFG | EVTO        | Boot mode  |
|---------|-------------|--|
| 0       | Do not care | Boot from internal flash memory                            |
| 1       | 0           | FlexCAN / eSCI boot (Standard Serial Boot mode)            |
| 1       | 1           | FlexCAN / eSCI boot (Baud Rate Detection Serial Boot mode) |

### 2.2.1 Reset Configuration Half Word (RCHW)

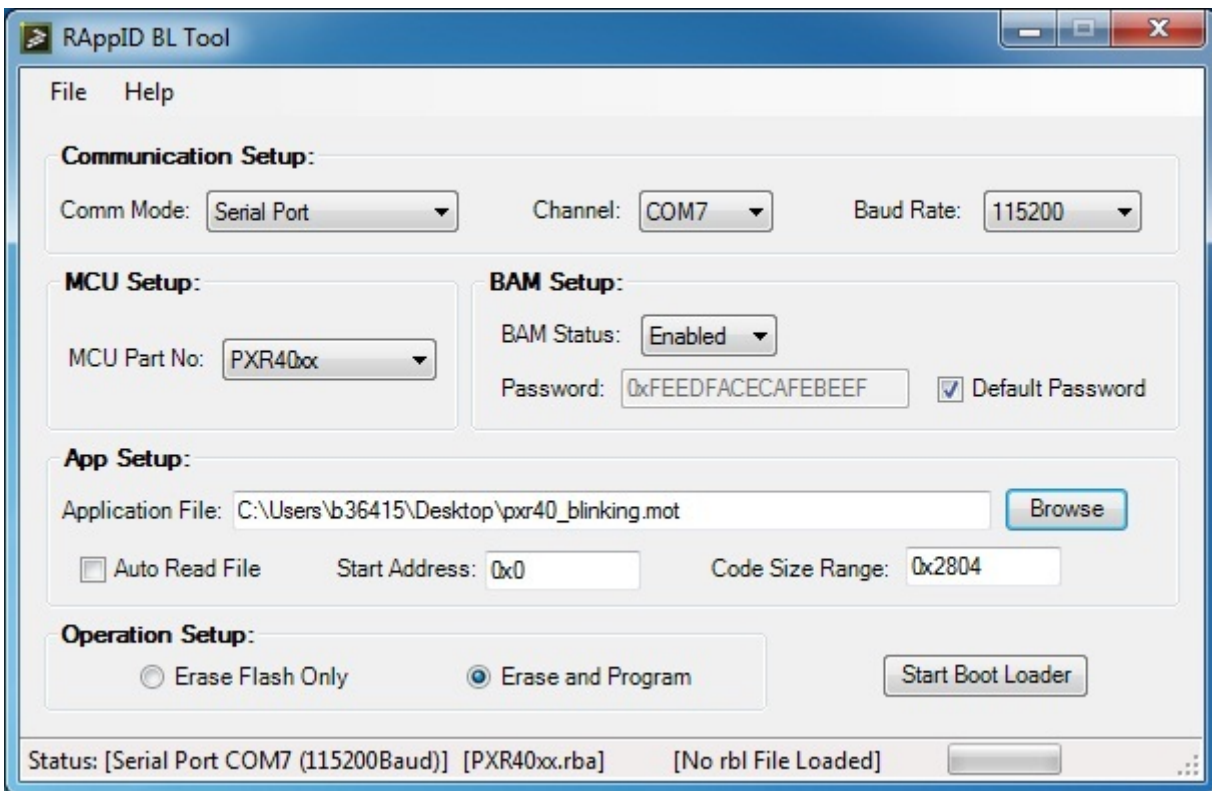
The RCHW defines boot options and must be programmed to predefined locations in the internal flash. The 32-bit word after the RCHW must be programmed with the user application's starting address. The BAM passes control to the user application at the starting address.

## 3 RAppID Boot Loader utility

The RAppID Boot Loader is a tool developed by Freescale to help software development for Freescale MCUs by allowing the customer a method to update software of an MCU through a serial link.

The RAppID Boot Loader works with the built-in BAM included in Freescale's Qorivva and PX series family of parts. The boot loader supports the operational modes of the different part family BAM silicon instantiations. The boot loader provides a streamlined method for programming code into FLASH or RAM on either target EVBs or custom boards. The boot loader allows to interface to the MCU with the communication interfaces supported by the BAM, which are Serial (UART) port and CAN interfaces.

The boot loader provides a GUI (see [Figure 3](#)) allowing the user to select the target communication mode, as well as set the target address, code size, and .s19/.mot file to be programmed. It also allows these settings to be brought in from a pre-set configuration file (.rbl) or allows to save a configuration file based on the current GUI settings. Once programming is complete, the application code automatically starts.



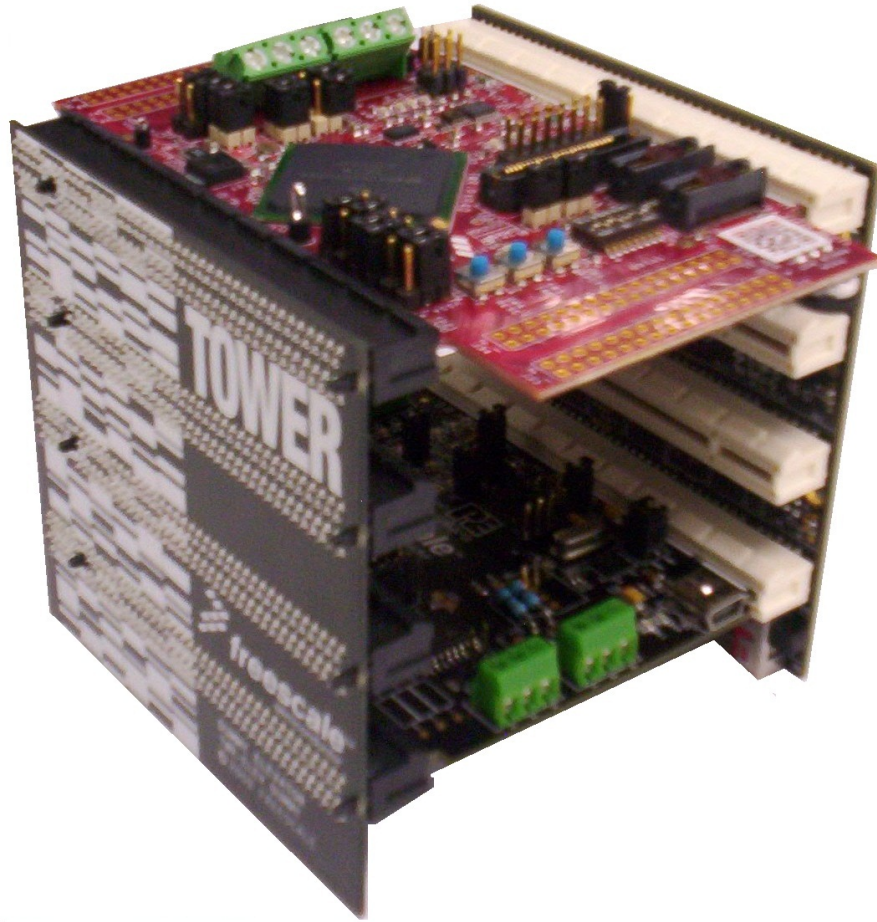
**Figure 3. RAppID boot loader utility**

RAppID Boot Loader utility can be used for application factory programming because it is an easy-to-use and free-of-charge tool. For more information regarding RAppID tools and downloading the RAppID Boot Loader utility, visit the RAppID web page, <http://www.freescale.com/RAPPID>.

## 4 Tower system setup

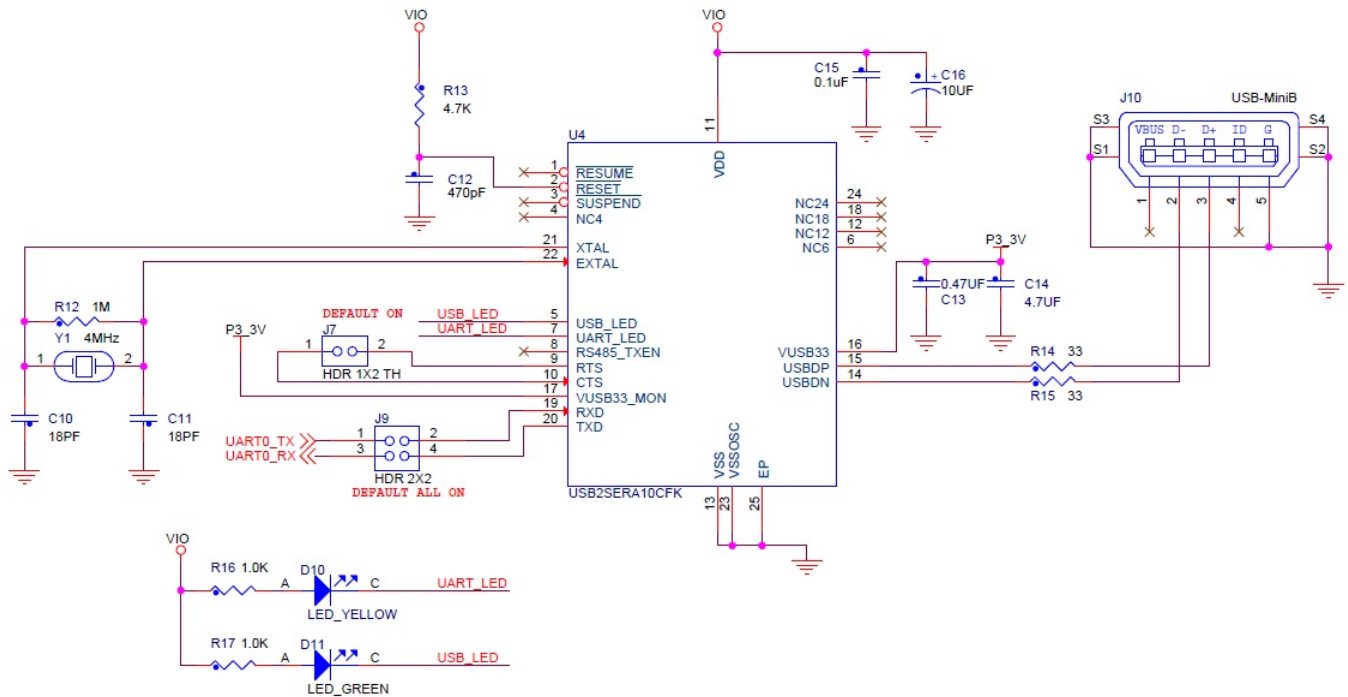
TWR-PXR40 board has the UART\_A mapped to the onboard RS485 transceiver and to the UART0 standard lines of the tower primary board. This will allow to use any tower peripheral module that has the UART0 enabled to download the code using the BAM.

This application note will use the TWR-IND-IO board that has a USB-to-serial (USB2SER) IC provided by Freescale. This USB2SER provides a USB-to-serial bidirectional bridge allowing the communication between the PXR40 microcontroller and the RAppID Boot Loader utility using a USB port.



**Figure 4. TWR-PXR40 with TWR-IND-IO**

The following figure shows USB2SER integrated on the TWR-IND-IO board.



**Figure 5. USB2SER implementation on TWR-IND-IO board**

**NOTE**

Check that jumpers on J9 connector on the TWR-IND-IO board are properly setup to allow the serial communication between the USB2SER and the microcontroller.

1. On TWR-PXR40 board, Dip-Switch No. 3 and No. 8 must be set “ON” to enable the BAM module of the PXR40 microcontroller.

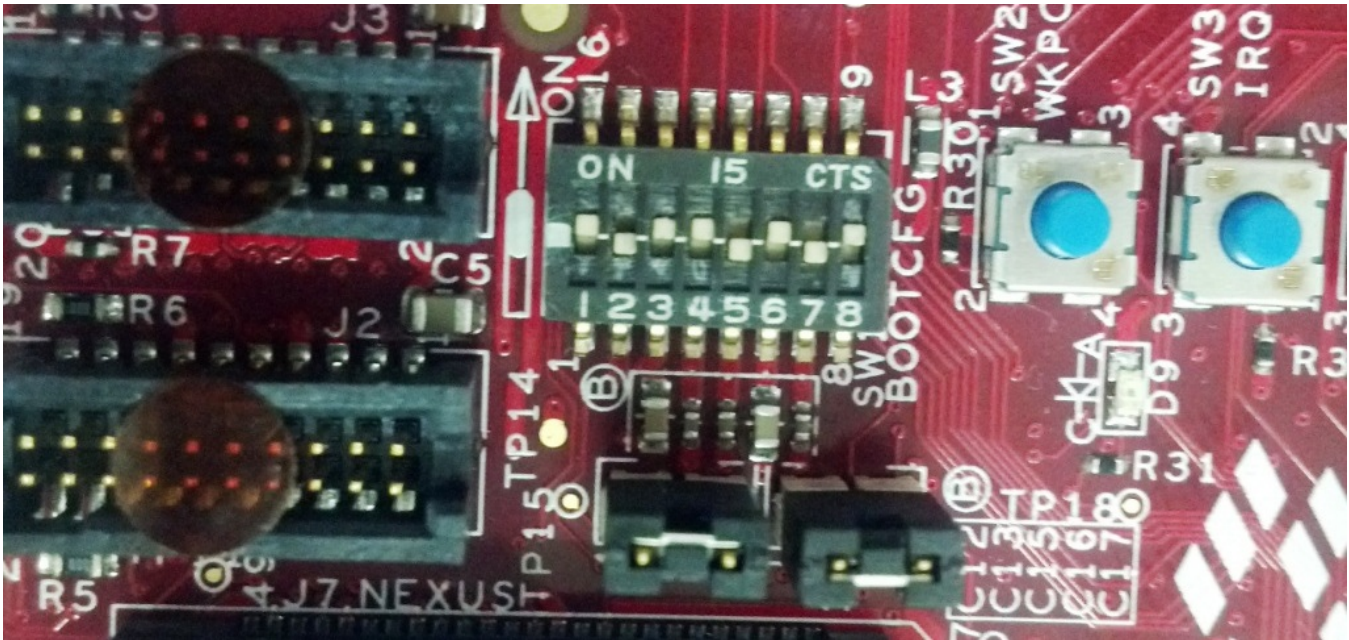
The following table shows the appropriate configuration for Dip-Switch No. 3 and No. 8 to allow PXR40 microcontroller to enter into BAM mode using auto-baud rate configuration.

**Table 2. Dip-Switch configuration on TWR-PXR40**

| Dip-Switch       | Position |
|------------------|----------|
| Dip-Switch No. 3 | ON       |
| Dip-Switch No. 8 | ON       |

The following figure shows how the Dip-Switch physically appears on the board.





**Figure 6. TWR-PXR40 Dip-Switch configuration for BAM**

2. Connect the USB cables, one on the TWR-IND-IO board and the other to the TWR-ELEV board to power-up the board.
3. Finally, turn on the tower board using the switch located at the bottom of the TWR-ELEV primary board.

**NOTE**

Make sure that USB2SER drivers are installed properly. If the USB2SER is installed correctly, one USB CDC device can be found under the Ports section on the Windows Device Manager, as shown in the figure below.

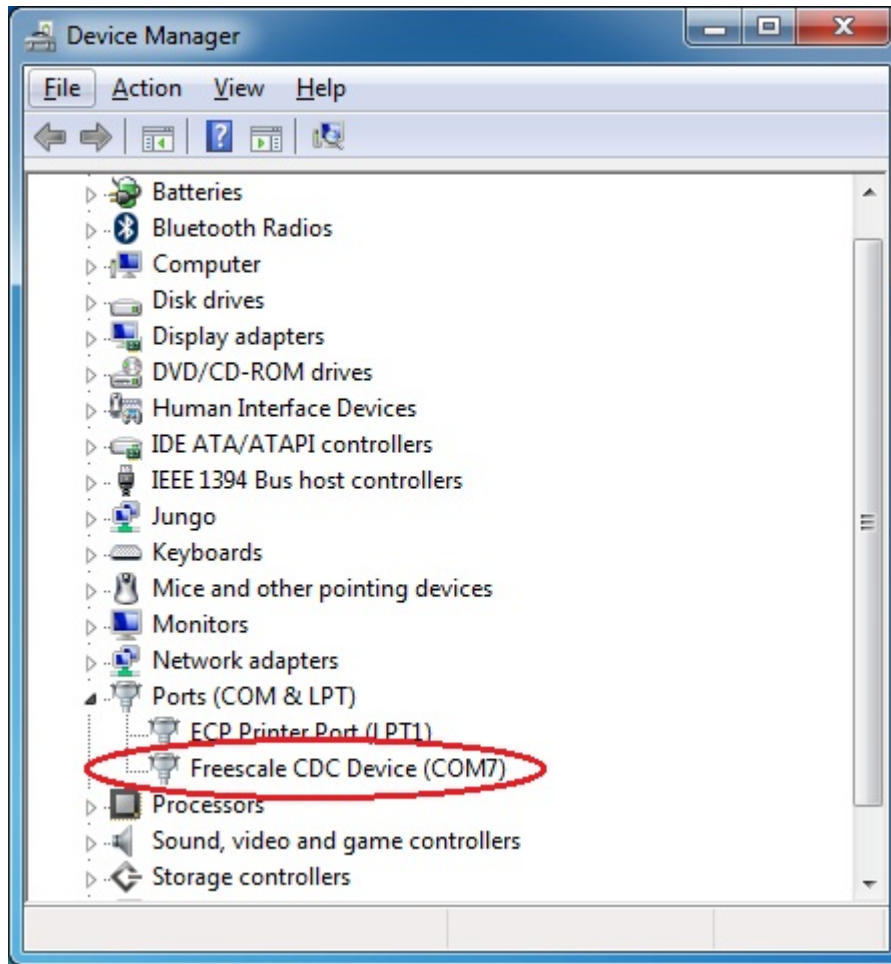


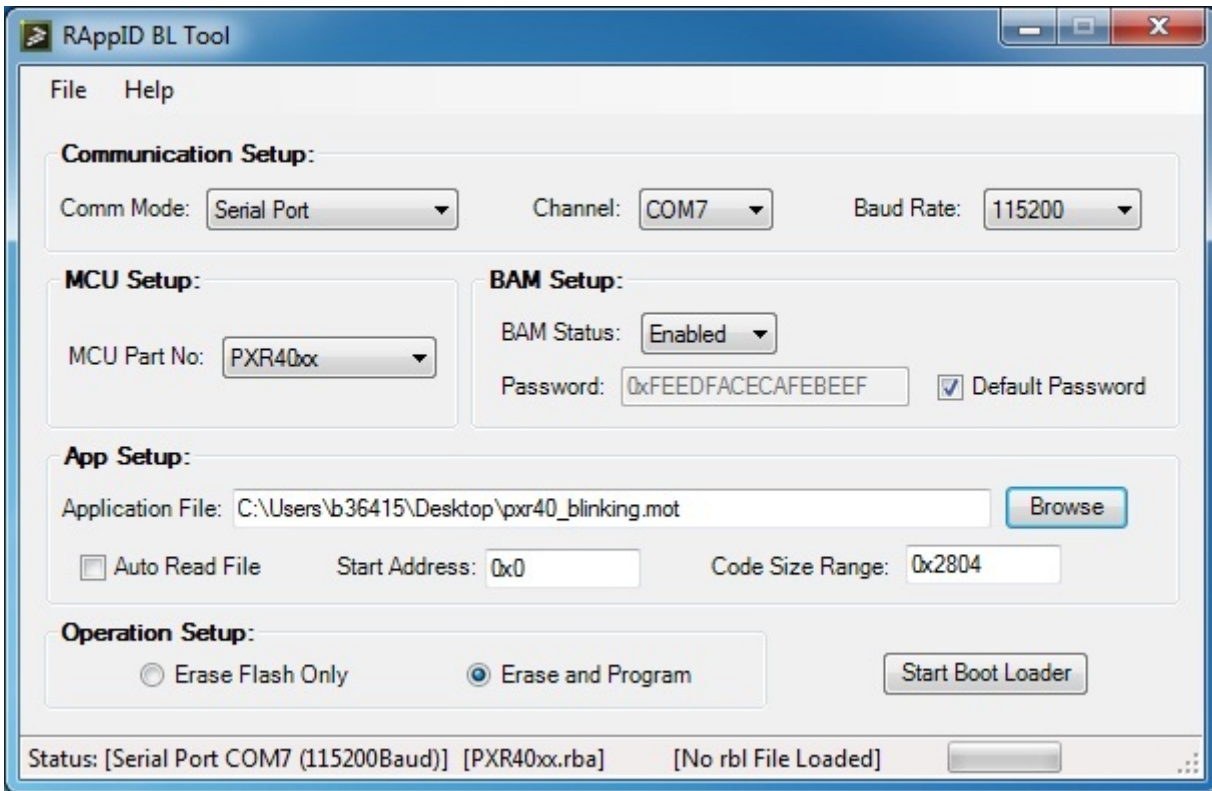
Figure 7. Windows Device Manager

## 5 Firmware download using RAppID Boot Loader tool

The RAppID Boot Loader tool is very simple to work with. The following steps will guide users to download firmware using RAppID.

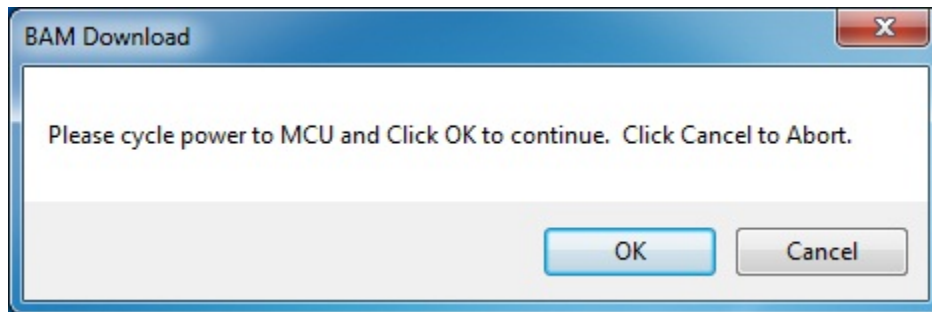
1. Select the appropriate configuration on each field on the graphical interface
  - Comm Mode > Serial Port (Communication interface for the boot loader)
  - Channel > COM7 (Serial Port assigned to the TWR-IND-IO board)
  - Baud Rate > Auto-baud rate is used, so leave the default value (115200)
  - MCU Part No > PXR40xx (Target MCU)
  - BAM Status > Enabled
  - Password > use the default password, so select the Default password checkbox
  - Application File > The S19 or MOT file to download to the microcontroller, use Browse to locate it

The following figure shows the complete configuration to download the Firmware to PXR40.



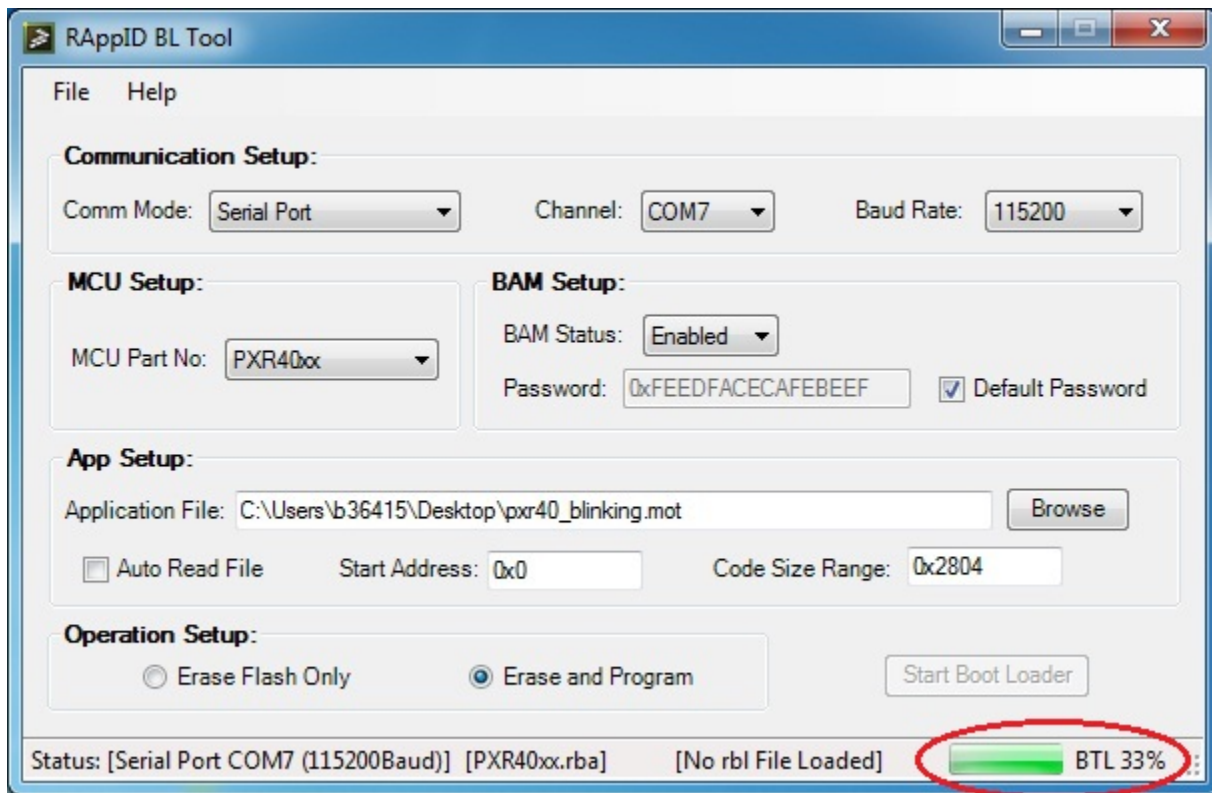
**Figure 8. RAppID Boot Loader tool interface**

- Once the configuration is complete, press the Start Boot Loader button and the following message will appear on the screen asking for a reset sequence of the microcontroller.



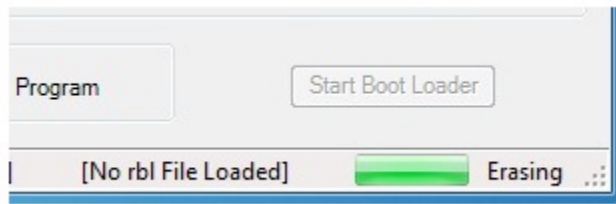
**Figure 9. BAM download**

- Press the RESET button of the TWR-PXR40 board and the OK button to start the firmware download process.
- If hardware setup and the GUI configuration are correct, the graphical interface at the bottom right will show the percentage of boot loader firmware downloaded.



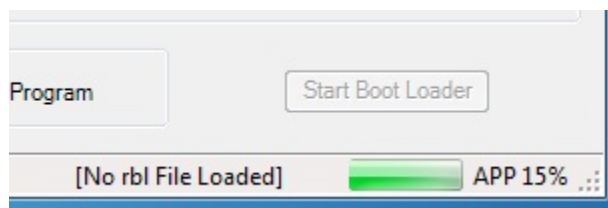
**Figure 10. Firmware download process**

- After it reaches 100%, the boot loader tool will show that the flash erase is in process.



**Figure 11. Flash erase process**

- Once the microcontroller flash has been erased, the MCU proceeds to download the application firmware and the graphical interface shows the percentage of the progress.



**Figure 12. Application download process**

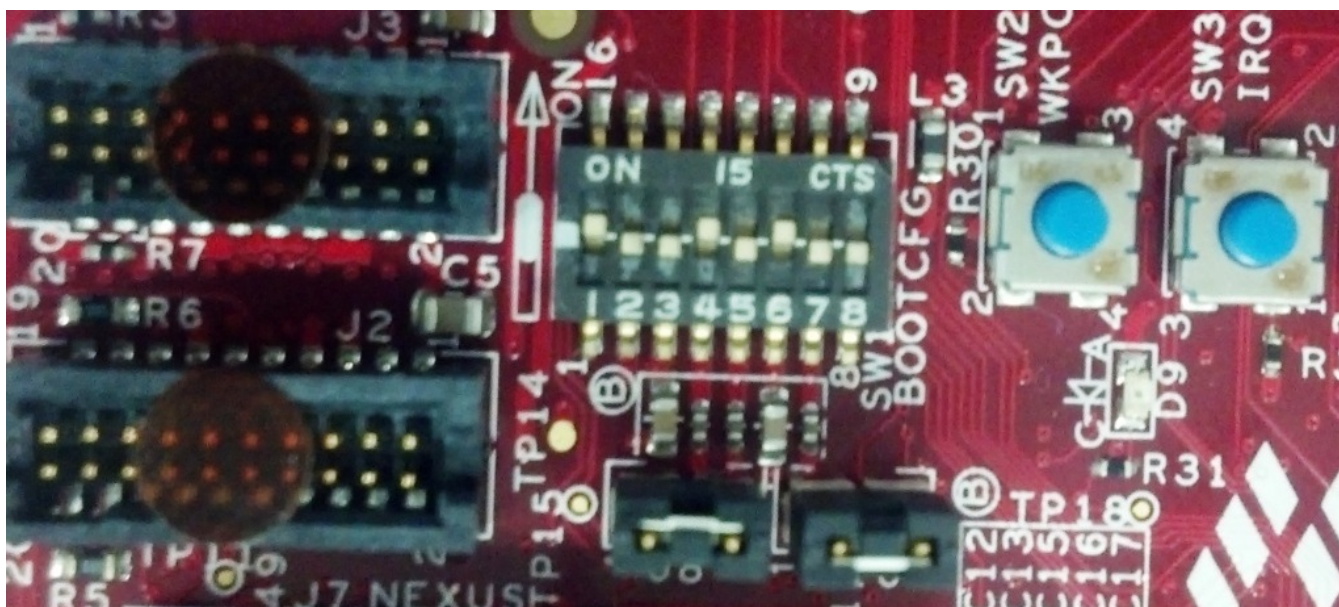
- Finally, the graphical interface informs that the new firmware has been downloaded successfully to the flash memory of the microcontroller.

To execute the application previously downloaded using the RAppID Boot Loader tool, return the Dip-Switch No. 3 and No. 8 to the default position, as shown in the table below.

**Table 3. Default Dip-Switch configuration on TWR-PXR40**

| Dip-Switch       | Position |
|------------------|----------|
| Dip-Switch No. 3 | OFF      |
| Dip-Switch No. 8 | OFF      |

The following figure shows the default Dip-Switch configuration that is required to run the application previously stored in the flash memory using the BAM tool.



**Figure 13. Default Dip-Switch configuration**

## 6 Conclusion

The BAM module of the PXR40 microcontroller together with the RAppID Boot Loader tool is a very good option for application factory programming because it provides an easy way to download code using a serial interface instead of a full JTAG interface.

See the PXR40RM: PXR40 Reference Manual available in <http://www.freescale.com/PXR40> for more details of the BAM module.

## **How to Reach Us:**

### **Home Page:**

[www.freescale.com](http://www.freescale.com)

### **Web Support:**

<http://www.freescale.com/support>

### **USA/Europe or Locations Not Listed:**

Freescale Semiconductor  
Technical Information Center, EL516  
2100 East Elliot Road  
Tempe, Arizona 85284  
+1-800-521-6274 or +1-480-768-2130  
[www.freescale.com/support](http://www.freescale.com/support)

### **Europe, Middle East, and Africa:**

Freescale Halbleiter Deutschland GmbH  
Technical Information Center  
Schatzbogen 7  
81829 Muenchen, Germany  
+44 1296 380 456 (English)  
+46 8 52200080 (English)  
+49 89 92103 559 (German)  
+33 1 69 35 48 48 (French)  
[www.freescale.com/support](http://www.freescale.com/support)

### **Japan:**

Freescale Semiconductor Japan Ltd.  
Headquarters  
ARCO Tower 15F  
1-8-1, Shimo-Meguro, Meguro-ku,  
Tokyo 153-0064  
Japan  
0120 191014 or +81 3 5437 9125  
[support.japan@freescale.com](mailto:support.japan@freescale.com)

### **Asia/Pacific:**

Freescale Semiconductor China Ltd.  
Exchange Building 23F  
No. 118 Jianguo Road  
Chaoyang District  
Beijing 100022  
China  
+86 10 5879 8000  
[support.asia@freescale.com](mailto:support.asia@freescale.com)

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductors products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claims alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-complaint and/or non-Pb-free counterparts. For further information, see <http://www.freescale.com> or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to <http://www.freescale.com/epp>.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© 2012 Freescale Semiconductor, Inc.