

Keywords: uC, evaluation kit, AFE, smart card, microcontroller, Rowley, IAR embedded workbench

## APPLICATION NOTE 5388

# Getting Started with the DS8005 Evaluation Kit

Apr 01, 2013

*Abstract: This application note describes how to build, debug, and run applications on the on-board MAXQ622 microcontroller to interface with the DS8005 dual smart card interface. This is demonstrated in both IAR Embedded Workbench and the Rowley CrossWorks IDE, using sample code provided with the kit.*

## Introduction

The **DS8005** dual smart card interface is a low-cost, dual analog front-end (AFE) for an IC card reader interface that needs to communicate with two smart cards in a mutually exclusive fashion. The analog interface is designed for use in ISO 7816, EMV®, and B-CAS applications. Additionally, the device is designed for applications where the C4/C8 (AUX1/AUX2) contacts are not required on either card interface. The DS8005 is designed to be used with microcontrollers that contain an ISO 7816 UART, or have the bandwidth to run this protocol in software by bit-banging IO ports.

In the DS8005 evaluation kit (**DS8005-KIT**), the DS8005 is provided in a 28-pin SOIC package. A smaller 28-pin TSSOP package is also available.

## Setting Up the DS8005 Evaluation Kit

A photo of the DS8005 evaluation kit board is provided in **Figure 1**. The following hardware components are contained in the evaluation kit and are used for implementing and verifying the demonstrated program:

1. DS8005 evaluation kit board
2. On-board **MAXQ622** microcontroller for developing smart card applications
3. USB-to-JTAG dongle for programming the MAXQ622
4. One ACOS3 smart card
5. Two card sockets (one smart card and one SIM/SAM card)
6. 2-line liquid-crystal display (LCD) screen for the user interface
7. Level-shifted RS-232 interface for debugging and the user interface
8. Pushbuttons for reset and user input
9. Regulated power supply (5V, ±5%, 300mA, center positive)

**Table 1** shows all the jumpers and includes a description of what each setting does. Settings highlighted in blue indicate the default settings. See **Figure 2** for jumper locations.



**Table 1. Jumper Settings for the DS8005 Evaluation Kit (JU1–JU7)**

Jumper	Setting	Description
JU1	Open	JTAG dongle is able to supply power to the board; an external power source (i.e., power jack) is needed to program the MAXQ622.
	Closed	JTAG dongle supplies power to the board and no external power source (i.e., power jack) is needed.
JU2	Closed (V50)	Ties the 5.0V supply directly to the VBUS pin on the MAXQ622
	Closed (VBUS)	Ties VBUS to the VBUS pin on the MAXQ622
JU3	Closed (V33)	3.3V is supplied to the smart card interface supply (VDDA).
	Closed (V50)	5.0V is supplied to the smart card interface supply (VDDA).
JU4	Open	Disconnects power to the DS8005; allows for current testing
	Closed	3.3V is supplied to the DS8005.
JU5	Open	Breaks the smart card interface power supply connection (VDDA). See JU3 setting.
	Closed	Closes the smart card interface power supply connection (VDDA). See JU3 setting.
JU6	Open	The interface A card presence indicator (SIM/SAM card) is pulled up to 3.3V. This pin is active high.
	Closed	The interface A card presence indicator (SIM/SAM card) is connected to ground. This pin is active high.
JU7	Open	The MAXQ622 is not able to supply a clock signal to the DS8005; an external clock is needed in socket XY1.
	Closed	Either the MAXQ622 or external clock can supply a clock signal to the DS8005. If the MAXQ622 is used, pin 6 of port 0 (TBB0) needs to be used to provide the clock signal. Otherwise, an external clock is needed in socket XY1.

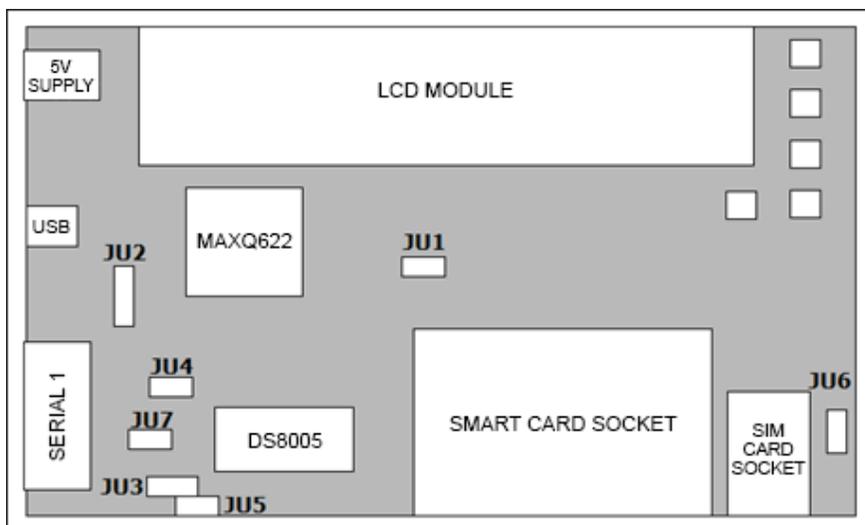


Figure 2. The location of all the jumpers on the DS8005 evaluation kit.

## Setting Up the FTDI Drivers

The USB-to-JTAG dongle is designed to connect to the host PC over a USB interface, which allows software running on the host (such as a program loading utility or application debugger) to communicate with the JTAG loader/debug interface on the MAXQ622 microcontroller. It uses the FTDI USB-to-serial IC to communicate with the USB interface on the PC. Before the FTDI interface can be used, the proper drivers must first be installed.

Note: The most up-to-date versions of all FTDI drivers and utilities can always be downloaded directly from FTDI's website at: [www.ftdichip.com](http://www.ftdichip.com). The DS8005 evaluation kit CD contains the latest versions of these utilities and drivers at the time the CD was last updated.

## Installing FTDI Drivers for the FT232RL

All required drivers for the FTDI USB-to-serial IC can be installed by running the FTDI driver setup executable. To run this executable, navigate to the **install/FTDI\_Drivers** directory of the CD and run the executable: **CDM20814\_Setup.exe**.

Once this setup executable has been run, you will be able to connect the USB-to-JTAG dongle to the PC using the USB interface cable.

When the USB cable is connected with USB-to-JTAG dongle, you will see notices in the Windows notification area that new hardware has been found, followed by a notice that the hardware installed correctly.

If you have difficulty installing the FTDI drivers correctly using the driver setup executable provided on the DS8005 evaluation kit CD, you can download and unpack the standard driver package, available on FTDI's website at [www.ftdichip.com/Drivers/VCP.htm](http://www.ftdichip.com/Drivers/VCP.htm). The driver package can be downloaded from this page and will be named similar to "CDM20814\_WHQL\_Certified.zip", where the five-digit number depends on the exact version of the driver that is currently available.

There are detailed instructions for installing the drivers provided by FTDI for Windows XP®, Windows Vista®, and Windows® 7; these are available in application note form: [www.ftdichip.com/Support/Documents/InstallGuides.htm](http://www.ftdichip.com/Support/Documents/InstallGuides.htm)

See also [www.ftdichip.com/Support/Documents/AppNotes.htm](http://www.ftdichip.com/Support/Documents/AppNotes.htm) for additional information from FTDI.

## Determining the COM Port Location of the USB-to-Serial Interface

The USB-to-serial interface provided by the FT232RL chip on the USB-to-JTAG dongle uses the virtual COM port (VCP) device model on Windows. This means that the USB-to-serial bridge appears as a standard COM port and can be used by any Windows application that has the ability to communicate over a COM port.

However, for an application (such as MAX-IDE or IAR Embedded Workbench®) to communicate with the VCP, the application must be configured by the user to use the correct COM port that has been assigned to the USB-to-serial interface. This COM port number may vary depending on previous devices that have been used and previous USB-to-serial drivers that have been installed.

To determine the current COM port number that has been assigned to the VCP, open the *Device Manager* utility in Windows.

## Opening Device Manager in Windows 7

**Method 1:** Click the Windows *Start* menu, followed by *Control Panel*. Next, click *System and Security*, followed by *Device Manager*.

**Method 2:** Click the Windows *Start* menu. In the search box, type “devmgmt.msc”, and hit *Enter*.

In the *Device Manager* window, open the section of the tree display labeled *Ports (COM & LPT)*. You will see a listing for *USB Serial Port (COMX)*, where *X* is the number of the COM port that has been assigned.

## Removing and Reinstalling the FTDI Drivers

If you have connected a large number of different FTDI-based USB-to-serial-adapter devices to your system, or if you simply have many USB-to-serial-based devices installed, you may find that the VCP assigned to the USB-to-JTAG dongle has a very high COM port number (such as COM23). Some tools (such as the IAR Embedded Workbench) may have difficulty working properly if the VCP number you are using is too high.

FTDI provides a graphical utility that can be used to remove all installations of the FTDI driver set automatically. This should allow future installations of the FTDI driver to start at a more reasonable VCP number (usually COM3, depending on the configuration of your system).

The FTDI uninstallation utility is located in the *install/FTDI\_Uninstall* directory on the CD. In this directory, run the *CDMuninstallerGUI.exe* program. Note: Before proceeding, disconnect all USB-to-serial devices from your PC.

In the *CDM Uninstaller* window, the *Vendor ID* field should read **0403**, and the *Product ID* field should read **6001**. Click **Add**, and these values will be copied into the center window. At this point, you can click the **Remove Devices** button, and the utility will uninstall all instances of the FTDI driver set.

After running the CDM Uninstaller tool, you will have to reinstall the FTDI driver set by rerunning the *CDM20814\_Setup.exe* executable in the *install/FTDI\_Drivers* directory, as described above.

If you have difficulty getting the drivers to reinstall correctly after running the CDM Uninstaller tool, try adding the following step: After running the CDM Uninstaller, but before running the *CDM20814\_Setup.exe*

executable, check the **DRVSTORE** directory, usually found in the Windows directory under the **System32** subdirectory. If you see directories named **ftdibus\_<xxxxx>** or **ftdiport\_<xxxxx>**, where <xxxxx> is a random-appearing string of hex digits, delete these directories before running the **CDM20814\_Setup.exe** executable. This will remove the preinstalled copies of the drivers and allow the drivers to reinstall cleanly.

## Configuring the FTDI Driver Settings

After installing the FTDI drivers and connecting the USB-to-JTAG adapter, there is an optional change you can make to the FTDI driver settings. Generally, this setting's change should improve the overall communications speed of the USB-to-JTAG interface, which is used to load and debug code on the MAXQ622.

To make this change, open **Device Manager** as described above. Find the entry **USB Serial Port (COMx)**, which represents the VCP for the USB-to-JTAG dongle you have connected. Double-click on this entry, or right-click it and select **Properties** from the pop-up menu (**Figure 3**).

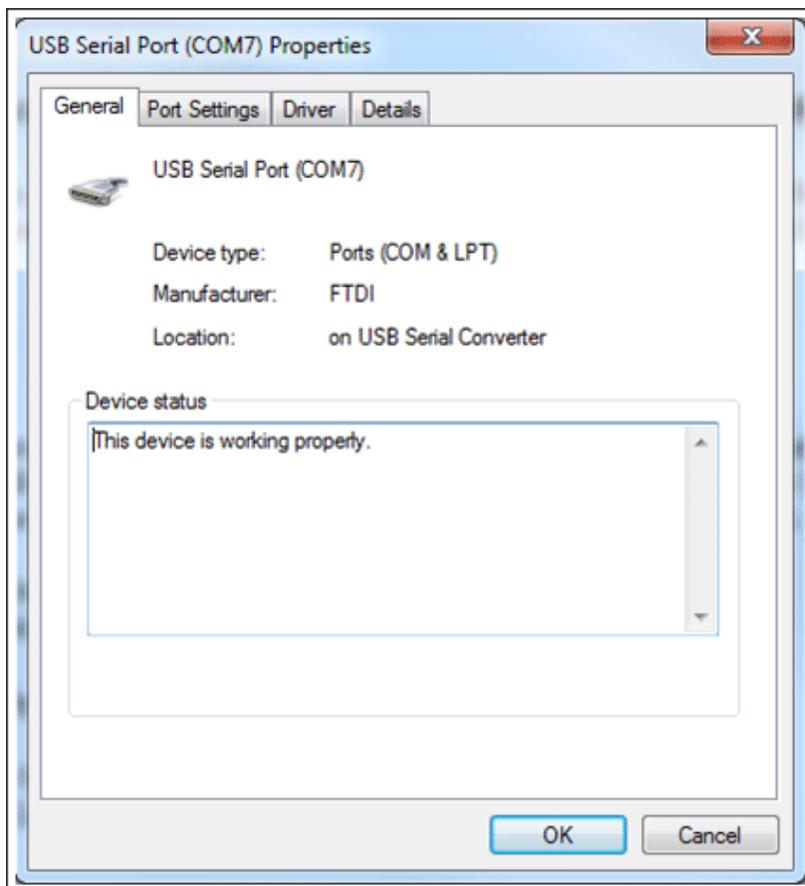


Figure 3. VCP driver Properties window.

In the **Properties** window, select the **Port Settings** tab, and click **Advanced**. This will display a secondary dialog box with advanced settings (**Figure 4**).

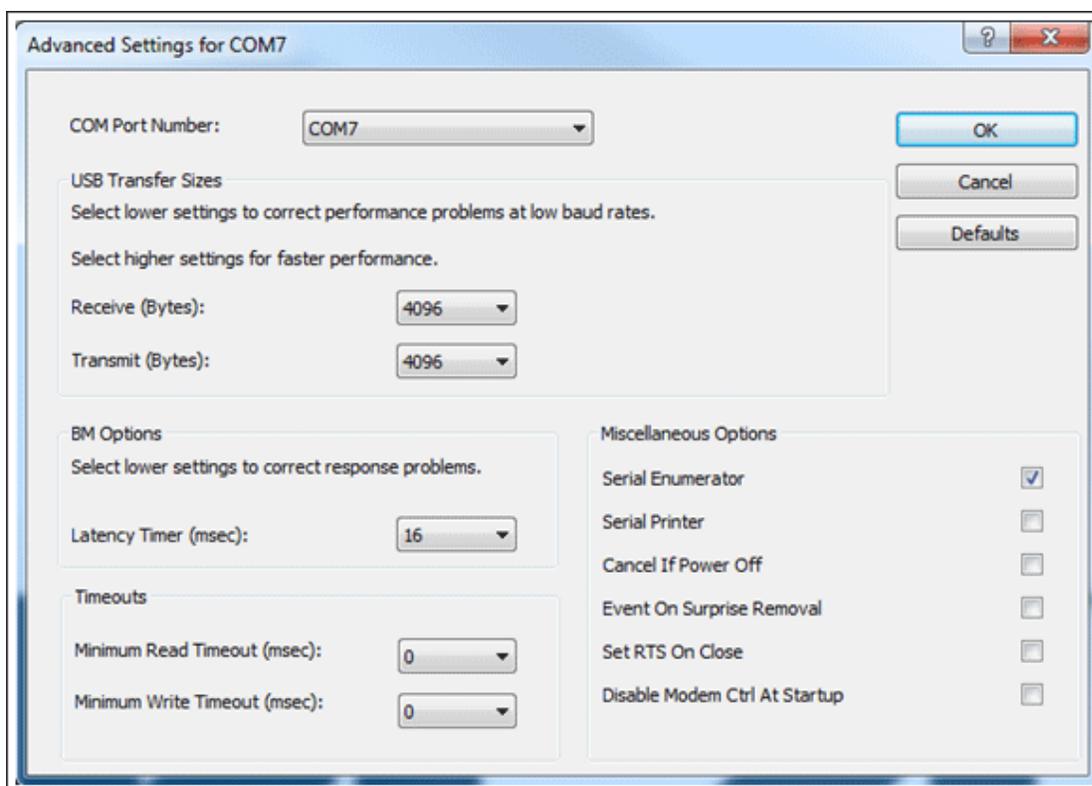


Figure 4. Advanced settings for the VCP driver.

In this dialog box, find the **Latency Timer** field shown above in the **BM Options** section. Change this field from its existing value (likely 16 or something similar) to 1 (which is the lowest possible value). Click **OK**, and then click **OK** again in the previous **Properties** window to complete the change.

Note that this is a change specific to one instance of the VCP, not a global settings change. If you uninstall and reinstall drivers, you must make this change again the first time a new kit board is connected. However, the change is made, it will be retained even if the kit is disconnected and reconnected; the setting stays with a particular instance of the FTDI driver installation.

## Installing and Using the Microcontroller Tool Kit (MTK)

The CD provides a sample transaction application. The binary hex file, **DS8005\_EK.hex**, is located under the **QuickStart\_Demo** folder. To load the code directly with a .hex file, the Maxim-provided Microcontroller Tool Kit 2 (MTK2) utility can be used. It is included in the CD, but can also be downloaded for [free](#) under “Loader Software.” Additionally, MTK2 can act as a terminal emulator, which can be used to view the example code’s data output through the RS-232 connector.

To install MTK2, run **MTK2\_Install\_2.4.14.exe**, which can be found in the **install/MTK** folder on the install CD, or on [Maxim’s website](#).

After installing MTK2, run it from the **Start** menu. In the initial **Select device** dialog box, select the MaxQ option and click **OK** to communicate with the MAXQ622 loader (**Figure 5**).

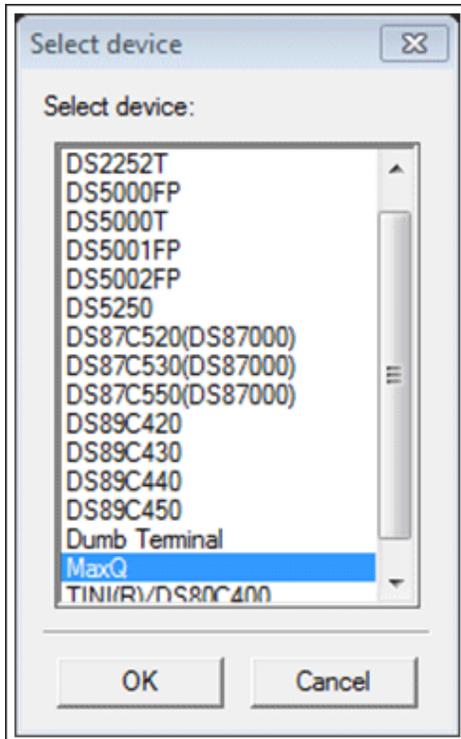


Figure 5. Selecting the MaxQ option from the initial device selection dialog box in MTK2.

Next, configure and open the serial port. Make sure the USB-to-serial adapter is connected; use DS1 to confirm the adapter is powered and check that the device has enumerated properly (**Devices and Printers** from the **Start** menu) before continuing. From the MTK menu, select **Options** → **Configure Serial Port**. In the dialog box (Figure 6), select the VCP that the USB-to-serial adapter is connected to (choose from the drop-down list, or enter it directly in the **Port** field) and click **Ok**.

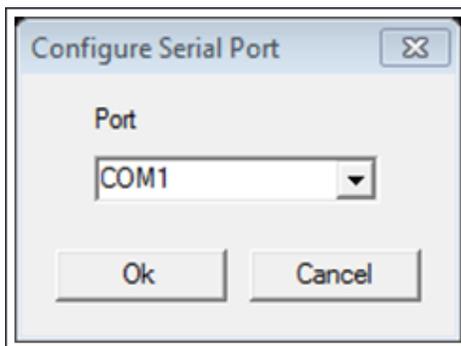


Figure 6. Configuring the serial port to load code in MTK2.

Now, connect the USB-to-serial adapter to the DS8005 evaluation kit board with the provided 10-pin ribbon cable. Use the red indicator to make sure pin 1 of the adapter's JTAG header is connected to the TCK pin of the JTAG header on the DS8005 (Figure 7). Ensure that the jumper settings on the board are on their default settings, as specified in the [Setting Up the DS8005 Evaluation Kit](#) section. Then from the MTK menu again, select **Target** → **Connect to Loader** to connect to the MAXQ622's loader. MTK should output a message showing that it was successful in communicating with the MaxQ Bootloader (Figure 8).

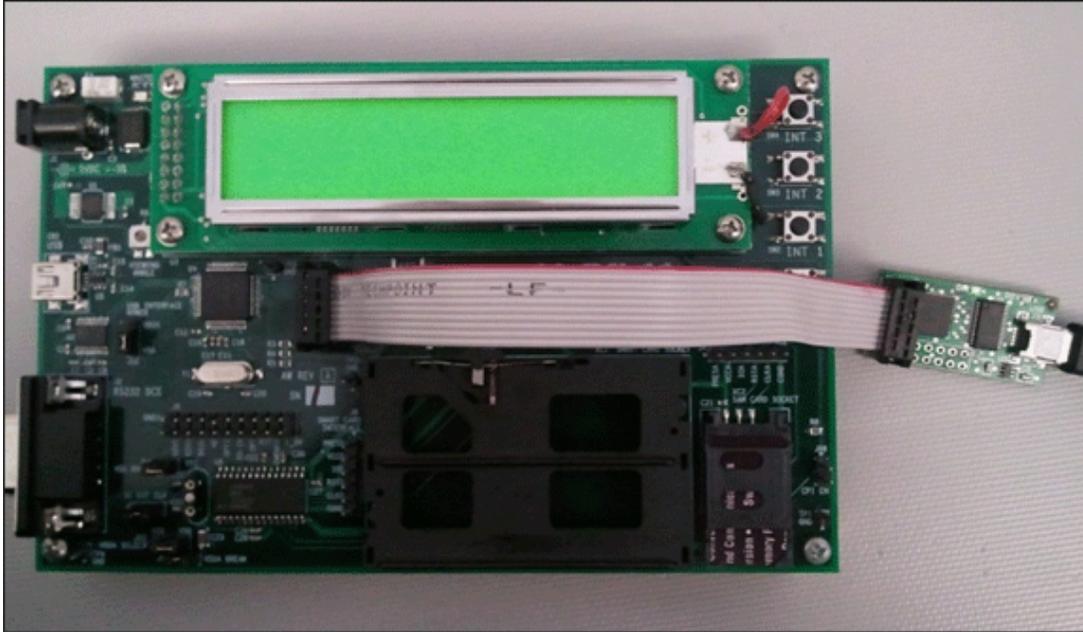


Figure 7. The USB-to-serial adapter connected to the JTAG header on the DS8005 with a standard 10-pin ribbon cable.

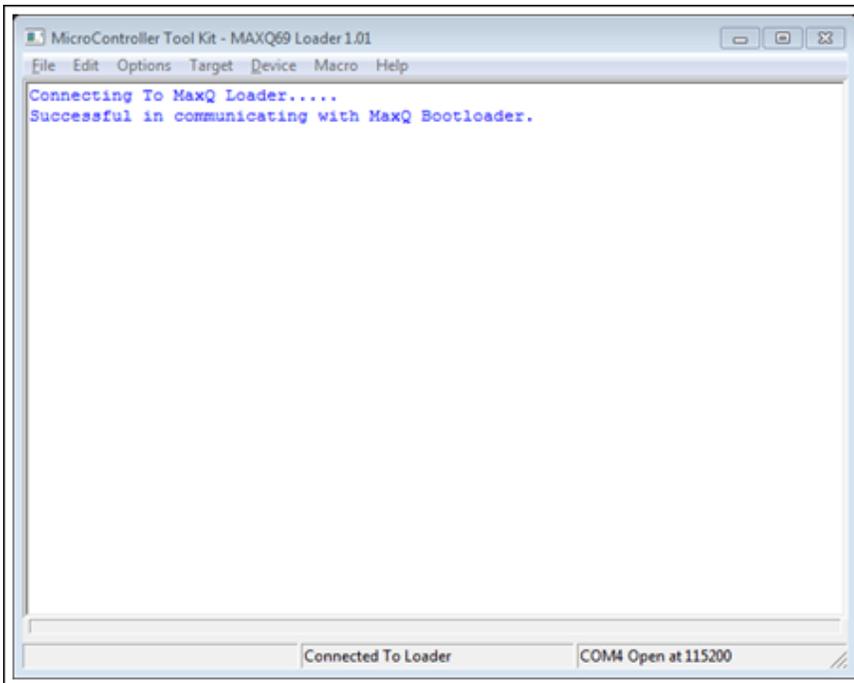


Figure 8. MTK2 after successfully connecting to the MaxQ Bootloader.

Finally, to load the example code, select **File** → **Master Erase** if the option is not already selected (indicated by a check mark), then select **File** → **Load Code** and locate and open the DS8005\_EK.hex file. This will take several seconds as the code size is relatively large. Once MTK is done, it will output a message showing that the load completed successfully (Figure 9). Select **Target** → **Disconnect from loader** to release the MAXQ622 from reset and begin code execution. Removal of the ribbon cable is optional. If the ribbon cable is removed, power will be removed as well, and the regulated power supply will need to be plugged in to provide



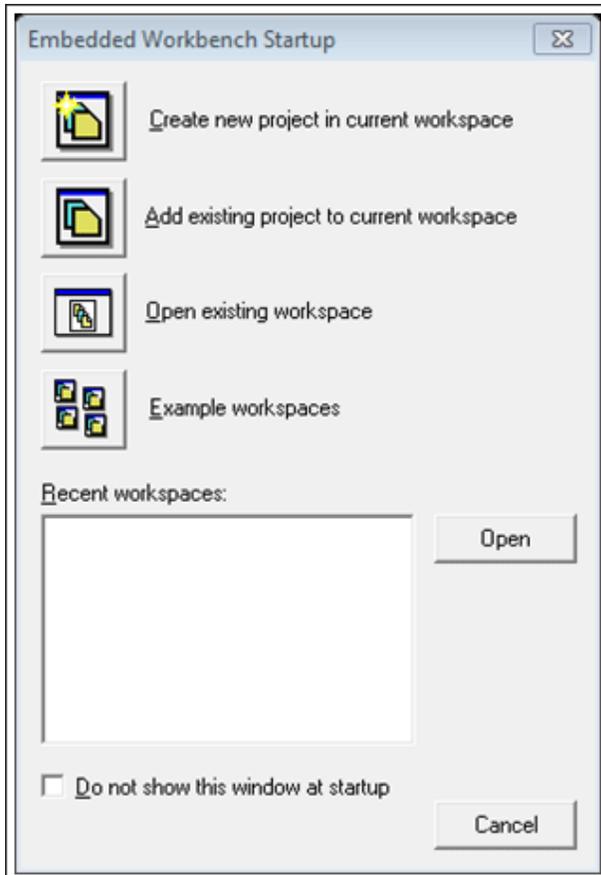


Figure 10. Opening an existing workspace on IAR startup.

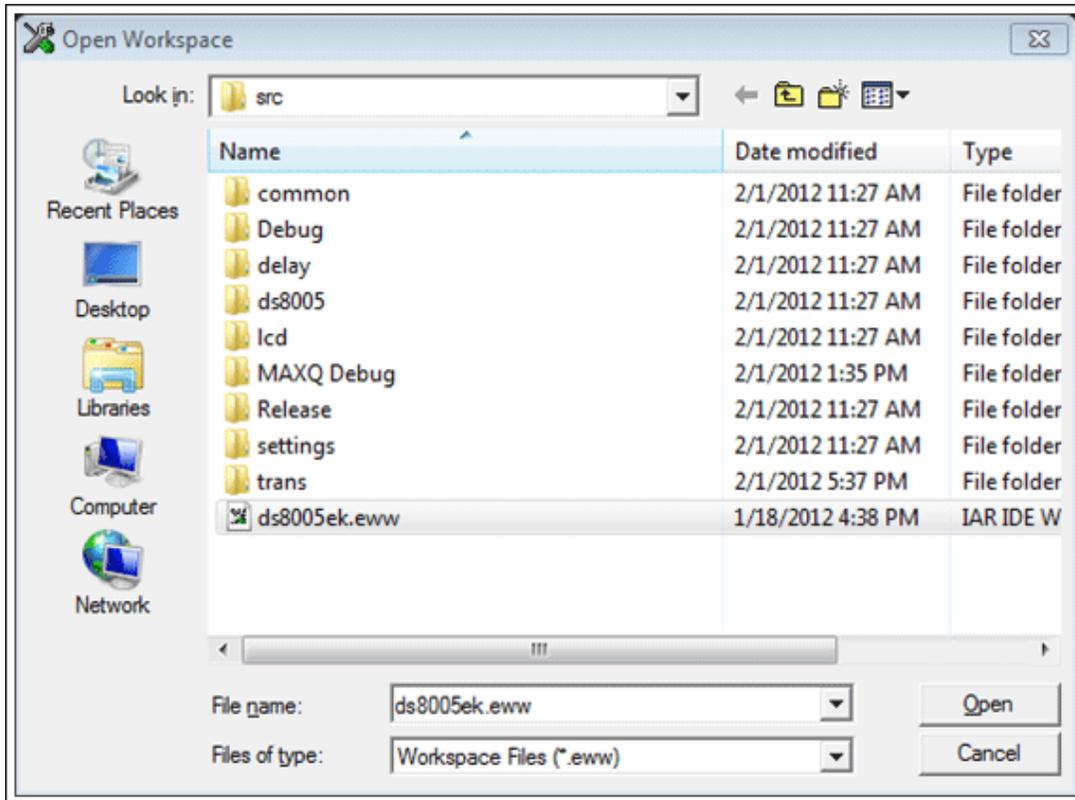


Figure 11. Opening up the sample code workspace in IAR.

All the project options should be correctly set. However, the COM port will need to be changed to match the VCP of the USB-to-serial adapter. Right-click on the project and select **Options** (Figure 12).

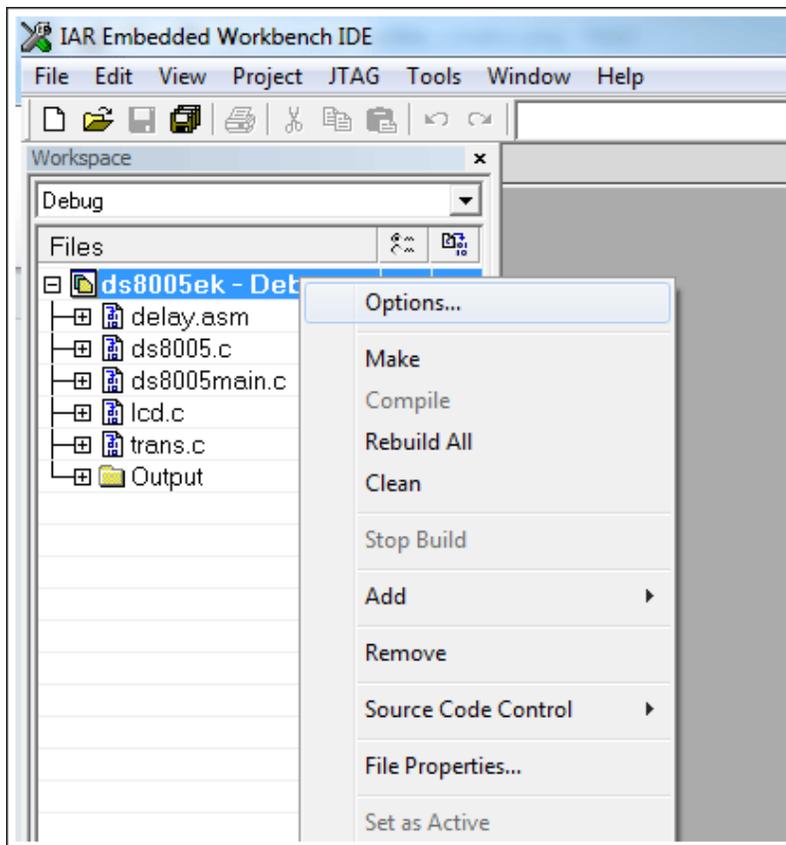


Figure 12. Modifying the project options in IAR.

In the Options dialog box, select the Debugger tab (**Figure 13**). Select JTAG from the Driver drop-down menu. Now select the JTAG subtab just below the Debugger tab (**Figure 14**). Change the COM port accordingly and click OK.

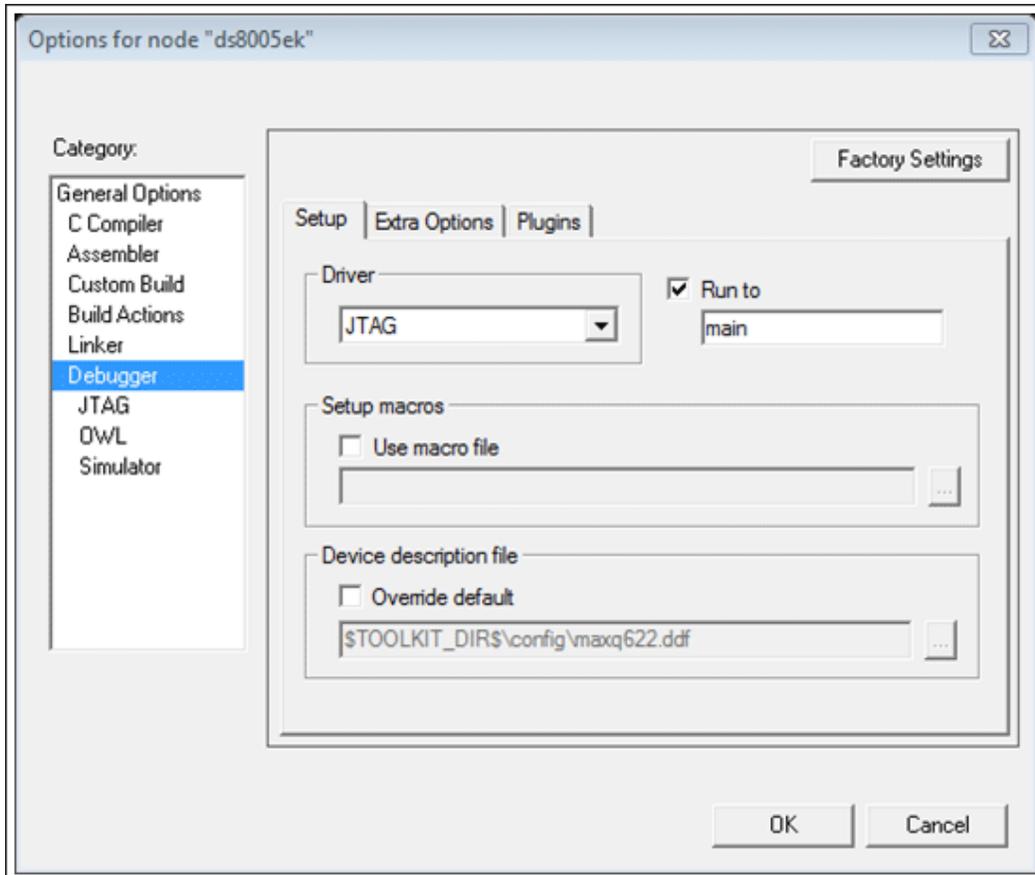


Figure 13. Selecting the JTAG driver in IAR.

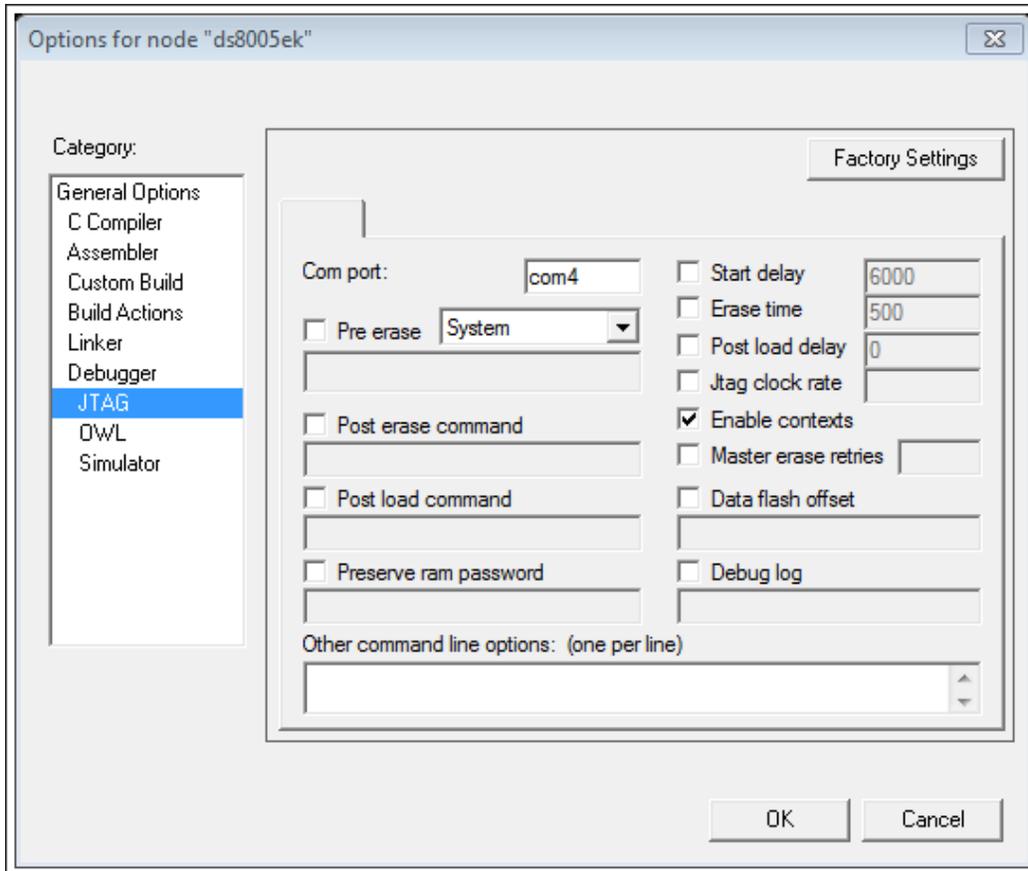


Figure 14. Selecting the COM port to be used for programming the MAXQ622 in IAR.

To rebuild the program, select **Project** → **Rebuild All** from the menu. The **Build Log** at the bottom should show that the program was successfully rebuilt by showing that the number of errors is zero. To run and debug the program, select **Project** → **Debug** from the menu. A small window named **Downloading Application** will appear with a status bar while the code is being loaded onto the MAXQ622. Once done, the window will disappear and a green arrow will point to the start of code since IAR automatically sets a break point at the start of main. To debug the code (Step Over, Step Into, Step Out, Next Statement, Go, Reset, etc.), select **Debug** from the menu and select one of the debug functions. These can also be selected via the shortcut buttons in the toolbar below the **IAR** menu (Figure 15).

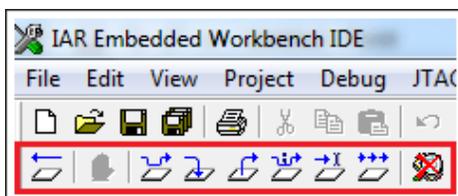


Figure 15. Debugging shortcut buttons in IAR.

To stop debugging, select **Debug** → **Stop Debugging** from the menu. To then run the code, push the reset button on the DS8005 evaluation kit board.

## Compiling, Debugging, and Running Code in Rowley CrossWorks

The example code in this evaluation kit was written and tested with version 2.1.0.2011101107.12397 of Rowley CrossWorks. This section describes how to compile, run, and debug the example code with Rowley. It may be necessary to reconnect the USB-to-serial adapter to avoid problems while trying to connect to the device.

Start CrossStudio for MAXQ 2.1 and select **File** → **Open...** from the menu. Locate and open the example code's project file DS8005\_EK.hzp (**Figure 16**).

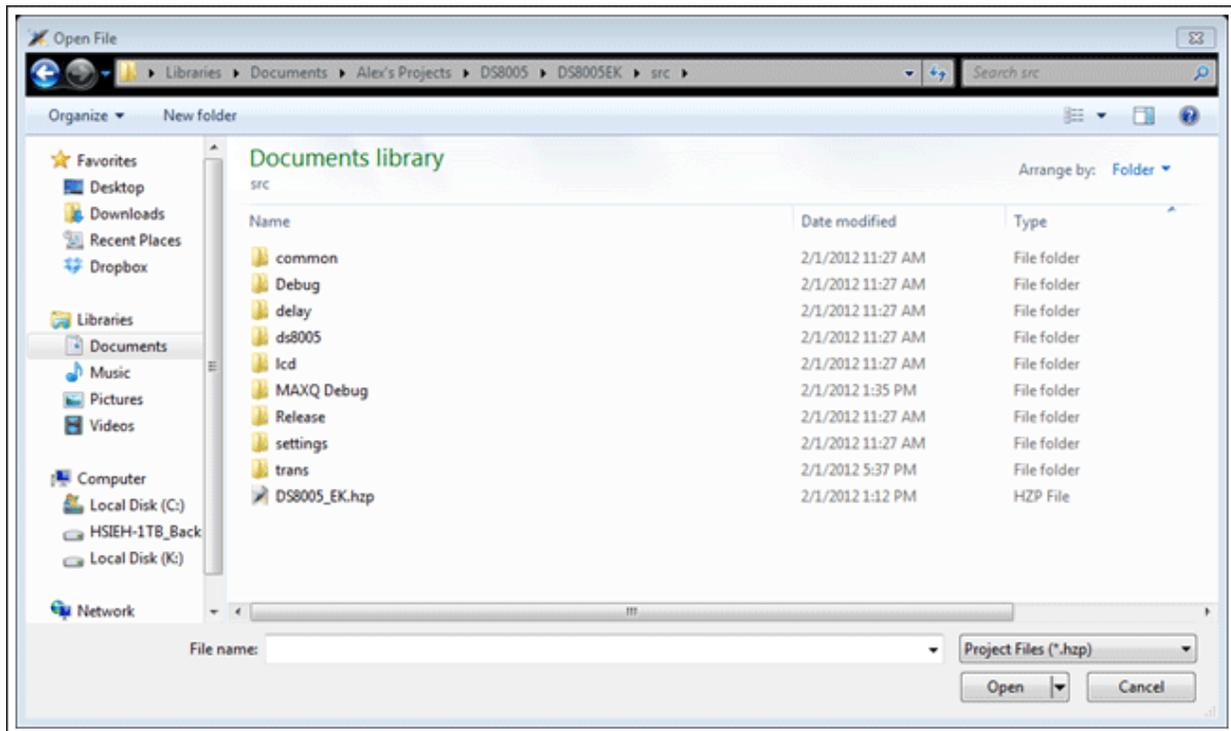


Figure 16. Opening the sample code project file in Rowley.

All the project options should be correctly set. From the menu, select **Build** → **Rebuild DS8005\_EK** to rebuild the example code (**Figure 17**). The output message below will show that the build completed successfully, as well as a summary of flash and RAM usage.

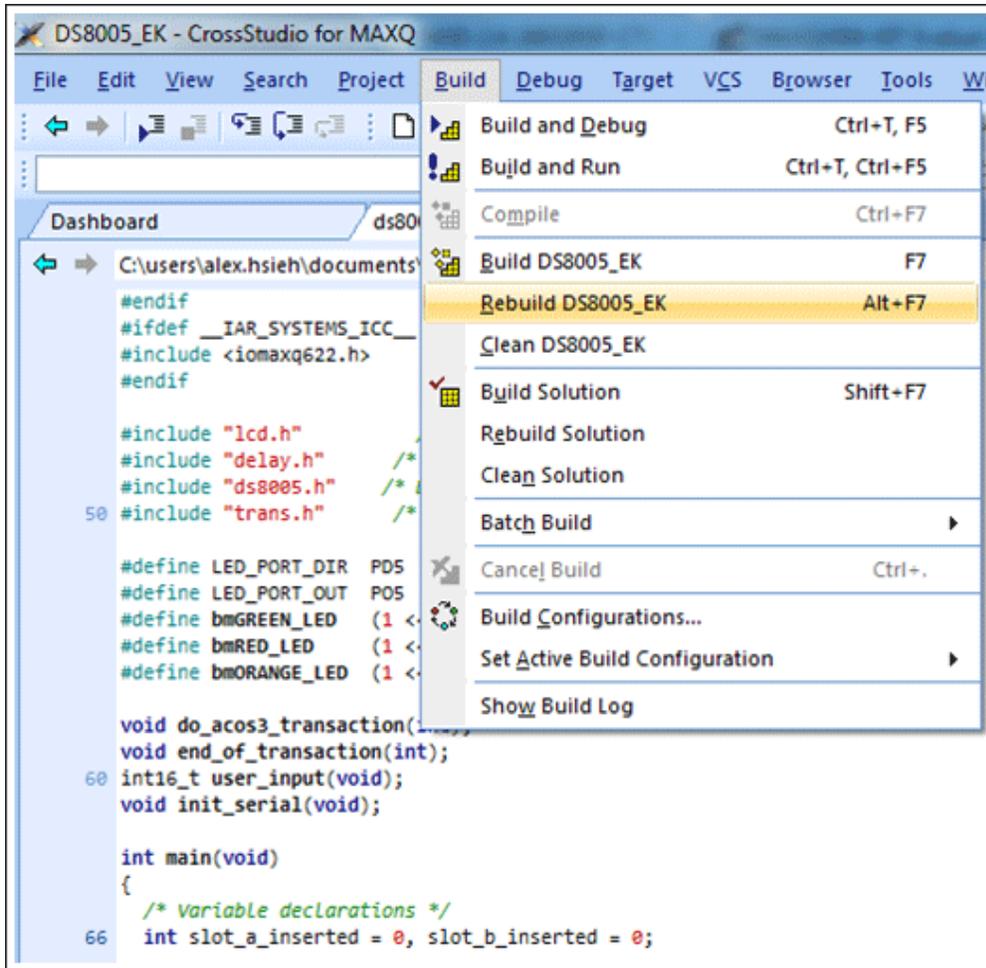


Figure 17. Rebuilding the sample code in Rowley.

Next, connect to the USB-to-serial adapter by first selecting **View** → **Targets**, then selecting **View** → **Property Window** to view the targets and target properties. These windows will appear on the right. Inside the **Targets** window, select **Maxim Serial JTAG Adapter** (Figure 18). The target's properties will appear in the **Properties Window**. Change the port name accordingly (Figure 19).

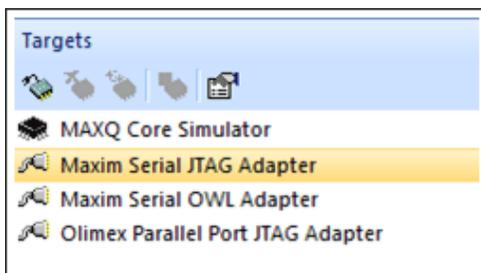


Figure 18. Selecting the Serial JTAG Adapter option in the Targets window in Rowley.

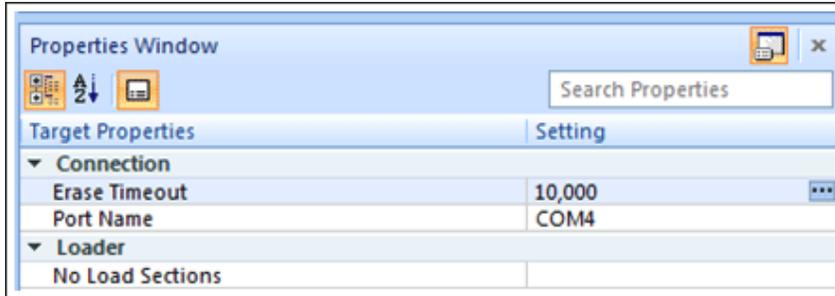


Figure 19. Selecting the COM port for the Serial JTAG Adapter in Rowley.

Next, select **Target** → **Connect** → **Maxim Serial JTAG Adapter** to connect to the USB-to-serial adapter (Figure 20). If successful, a message on the bottom left of the window will confirm the connection and a status indicator at the bottom right of the window will show a yellow circle and “Maxim Serial JTAG adapter” (Figure 21).

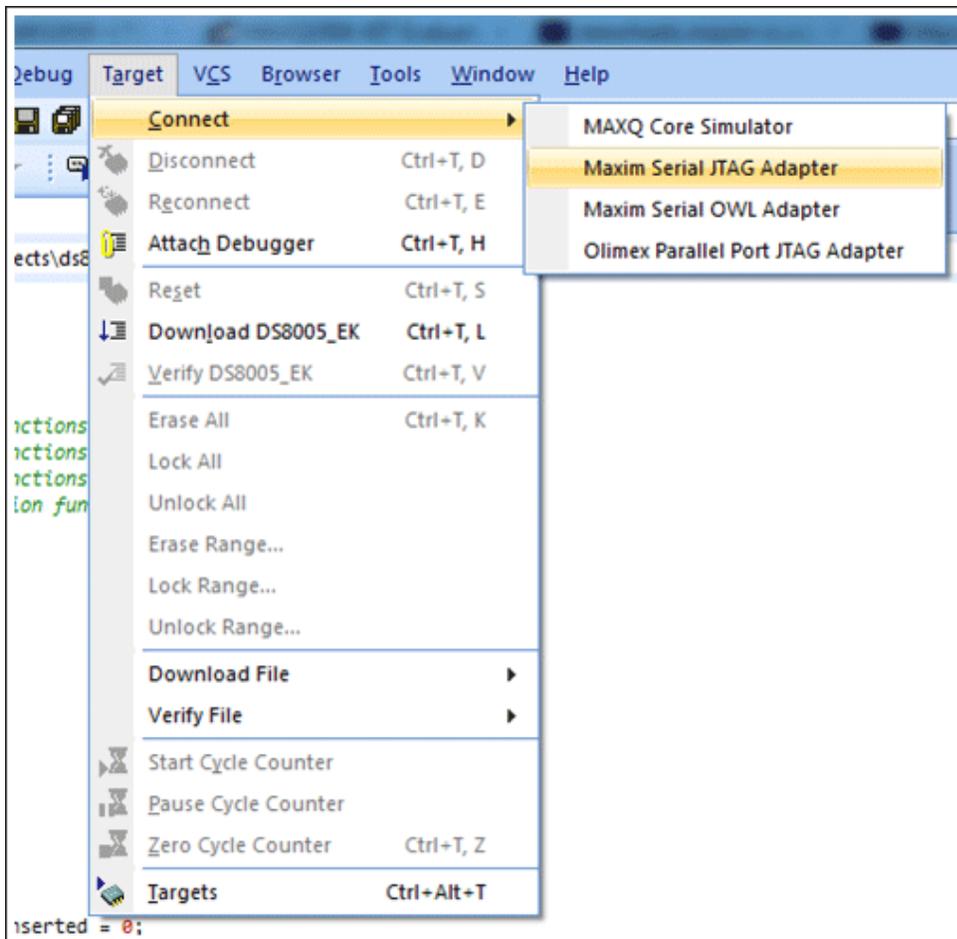


Figure 20. Connecting to the Serial JTAG Adapter in Rowley.

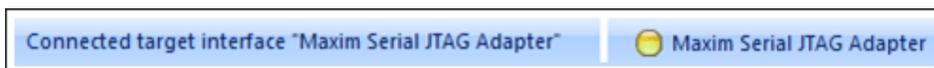
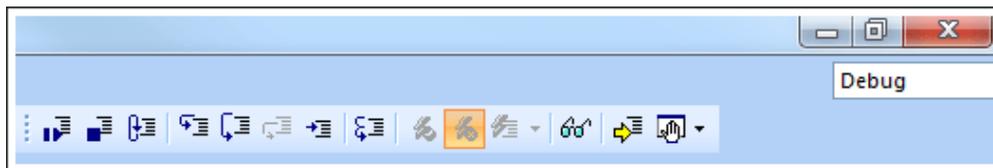


Figure 21. Serial JTAG adapter connection status. The left message shows feedback on the connection status, and the message on the right shows that the serial JTAG adapter is connected and idle.

To run and debug the program, select **Debug → Go**. Rowley will download the code onto the MAXQ622, with the progress displayed in the output window at the bottom. Once done, a disassembly window will appear on the left and a yellow arrow will point to the start of code, since Rowley also automatically places a breakpoint at the start of main. To debug the code (Step Over, Step Into, Step Out, Next Statement, Go, Reset, etc.), select Debug from the menu and select one of the debug functions. These can also be selected via the shortcut buttons in the toolbar in the Disassembly window (**Figure 22a**) or on the far right corner of the main toolbar (**Figure 22b**).



(a)



(b)

Figure 22. Shortcut buttons for debugging in Rowley. The top shows the buttons available in the “Disassembly” window. The bottom shows the buttons available on the far right corner of the main toolbar.

To directly program the code onto the MAXQ622 (without debugging), first exit debug mode by selecting **Debug → Stop**. Then select **Target → Download DS8005\_EK**.

## Using MTK as a Terminal Emulator

The sample code sends data over the serial port to provide more details about what is going on. MTK provides an easy-to-use terminal emulator (on Windows) for convenience. To use MTK as a terminal emulator, restart MTK by closing it and reopening it. In the initial **Select device** dialog box (**Figure 23**), select **Dumb Terminal**, and click **OK**.

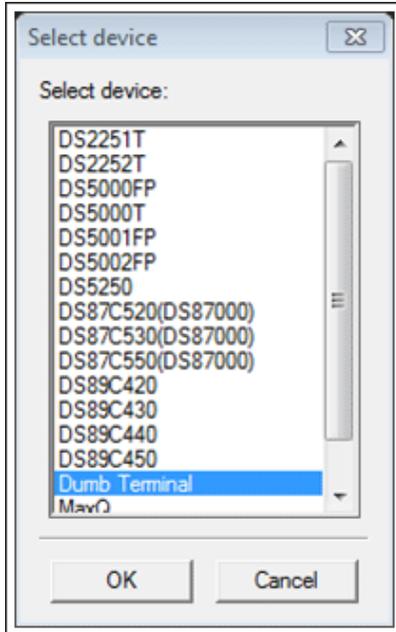


Figure 23. Selecting the Dumb Terminal option from the initial Select device dialog box in MTK2.

The serial port will need to be reconfigured before opening it, as it will still be configured to the VCP of the USB-to-serial adapter. From the MTK menu, select **Options** → **Configure Serial Port**. This time, the dialog box (Figure 24) will provide an additional speed (baud rate) setting. Select the physical COM port the serial cable is connected to and select “115200” for the Speed setting. This is the baud rate used in the example code.

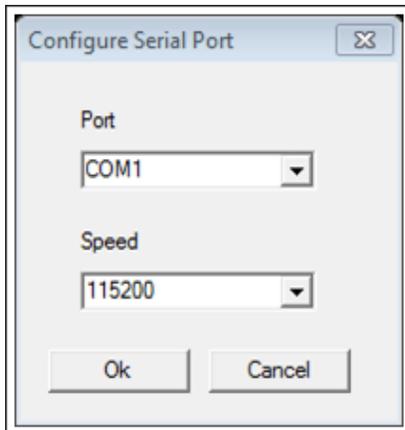


Figure 24. Configuring the serial port for terminal emulation in MTK2.

Lastly, select **Target** → **Open COMx at 115200 baud** from the menu. The main section of the MTK window will turn white to indicate that the port has successfully been opened.

## Running the Example Code

Before running the code, we recommended opening a terminal emulator like MTK and connecting a serial cable from the PC to J2 on the EV kit board to view additional output. To run the sample code, load the hex file through MTK, directly download the code in Rowley, or use the debugger (in IAR or Rowley).

The example application performs transactions involving two ACOS3 smart cards (a SAM and SIM card), and demonstrates the DS8005's ability to communicate with two cards in a mutually exclusive fashion. The application simulates a situation where a SIM card remains in place once inserted, while it waits for a SAM card to be inserted.

Initially, the application waits until either pushbutton SW4 is pressed (for communication with the SIM card) or a SAM card is inserted into the smart-card socket. The first time SW4 is pressed, the application will power up the card and issue an Answer to Reset (ATR) to the SIM card. If this is successful, the application will perform basic credit and debit transactions, along with a smart card initialization function based on user input from the pushbuttons. When a SAM card is inserted, the application halts communication with the SIM card (but retains power to the card) and performs an ATR on the SAM card. Again, if the ATR is successful, it performs basic credit and debit transactions and a smart-card initialization function until the SAM card is removed. Once the SAM card is removed, the application goes back to waiting until either SW4 is pressed or another SAM card is inserted. If SW4 is not pressed for the first time (i.e., the application has already performed an ATR on the SIM card), it will assume the SIM card is still present and resume communication with it, bypassing the ATR step as it is only necessary at card power-up. If the SIM card is removed after it has already been properly powered up, the next time the application tries to communicate with it, an error will occur, and it will power down the SIM card and return to waiting for SW4 or SAM card insertion. Now, if SW4 is pressed, and the SIM card is properly inserted back into the SIM card socket, the application will power-up the SIM card and issue another ATR.

The LCD screen acts as the main user interface, providing instructions on how to proceed and perform basic functions on the two smart cards, while the serial port gives additional information as to what is going on behind the scenes.

The three LEDs act as visual indicators of what is going on in the application as well. The green LED indicates that the application is running. The orange (top) LED indicates that the application is currently communicating with the inserted SAM card. The red (middle) LED indicates that the application is currently communicating with the SIM card.

## Smart Card Transactions

The smart card reader ([MAXQ622](#)) properly manages the sending and receiving of application protocol data units (APDUs), as defined in the EMV specifications, to the ACOS3 smart cards supplied in the DS8005-KIT. This is used to demonstrate typical payment transactions. The four transactions implemented in the example code are:

1. A debit transaction, where \$10 is deducted from the card's balance
2. A credit transaction, where \$10 is added to the card's balance
3. A card-initialization transaction, where the card's balance and transaction count are set to \$100 and 0, respectively
4. A check-balance transaction, where the card's balance and transaction count are read from the card

EMV is a registered certification mark owned by EMVCo, LLC. (See [EMVCo Disclaimer](#).)

IAR Embedded Workbench is a registered trademark of IAR Systems AB.

Windows is a registered trademark and registered service mark of Microsoft Corporation.

Windows Vista is a registered trademark and registered service mark of Microsoft Corporation.

Windows XP is a registered trademark and registered service mark of Microsoft Corporation.

## Related Parts

<a href="#">DS8005</a>	Smart Card Interface	<a href="#">Free Samples</a>
<a href="#">DS8005-KIT</a>	Evaluation Kit for the DS8005	
<a href="#">MAXQ622</a>	16-Bit Microcontrollers with Infrared Module and Optional USB	<a href="#">Free Samples</a>

---

### More Information

For Technical Support: <http://www.maximintegrated.com/support>

For Samples: <http://www.maximintegrated.com/samples>

Other Questions and Comments: <http://www.maximintegrated.com/contact>

---

Application Note 5388: <http://www.maximintegrated.com/an5388>

APPLICATION NOTE 5388, AN5388, AN 5388, APP5388, Appnote5388, Appnote 5388

© 2013 Maxim Integrated Products, Inc.

Additional Legal Notices: <http://www.maximintegrated.com/legal>