Keywords: MAXQ, MAXQ7665

APPLICATION NOTE 4100

# MAXQ7665 Evaluation Kit Quick Start Guide

Sep 27, 2007

*Abstract: This application note is a Quick Start Guide. It explains how to install and configure the IAR Embedded Workbench®. In addition, it supplies a brief guide to compile, link, and debug several example projects included in the MAXQ7665 microcontroller EV kit. A troubleshooting section is provided to determine the status of the EV kit.*

## Introduction

The MAXQ7665 Microcontroller Evaluation (EV) Kit contains a serial-to-JTAG interface for programming and debug. The IAR Embedded Workbench interfaces to the MAXQ7665 through JTAG and is used during debugging. This *Quick Start Guide* supplements the EV kit data sheet, and documents the use of the IAR toolset.

## IAR Workbench Overview

The IAR Embedded Workbench is an integrated development environment for the MAXQ7665 microcontroller and other MAXQ® microcontrollers. The IAR toolset includes a text editor, a C compiler, serial download, and a powerful debugger.

## IAR Workbench Installation

Install the IAR Embedded Workbench on the PC from the IAR website. Select the "Chip manufacturer" box on the IAR homepage. Then select Maxim/Dallas Semiconductor from the "Please select chip manufacturer" drop-down box and click the OK button. A new page appears with hyperlinks. Click on the "IAR Embedded Workbench for Maxim MAXQ" and another page appears with a highlighted Download box. Select the "30-day evaluation version." A page for product registration appears and must be completed to begin downloading the IAR Embedded Workbench for Maxim MAXQ. Once downloaded, the evaluation version is valid for 30 days before you must purchase a permanent license.

The IAR toolset is now installed. The MAXQ7665's personality files included in the installation are:

a) Config Folder:

```
maxq7665_A64K.ddf
maxq7665_A128K.ddf
lnkmaxq7665_A64K.xcl
lnkmaxq7665_A128K.xcl
lnkmaxq7665_cbl_A64K.xcl
maxq7665.hex
maxq7665.sfr
```

b) Config/Devices Folder:

```
maxq7665_A64K.menu
maxq7665_A128K.menu
maxq7665_cbl_A64K.menu
```

c) Include Folder

```
iomaxq766x.h
```

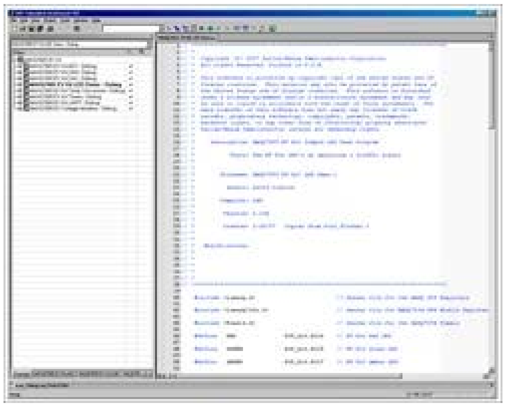
Copy the MAXQ7665 EV Kit folder from the `src` directory on the EV kit's Quick Start CD either to `C:\Program Files\IAR Systems\Embedded Workbench 4.0 Evaluation Version\MAXQ\src` in the default directory, or to the path where the IAR tools were installed.

## Hardware Installation

Connect the MAXQ7665 EV kit to the AC-power adapter. Connect the RS-232 serial-port cable from the PC to the EV kit's DB-9 connector labeled PC RS232 JTAG.

## IAR Workspace

The IAR structure for file organization requires a Workspace for projects. This space can then include C and assembly source files for specific tasks. Each project can be compiled, debugged, and run independently. **Figure 1** below shows the MAXQ7665 EV Kit workspace and the projects contained there. Please see the *IAR Embedded Workbench User's Guide* under the Help menu in the IAR Embedded Workbench application for additional details.

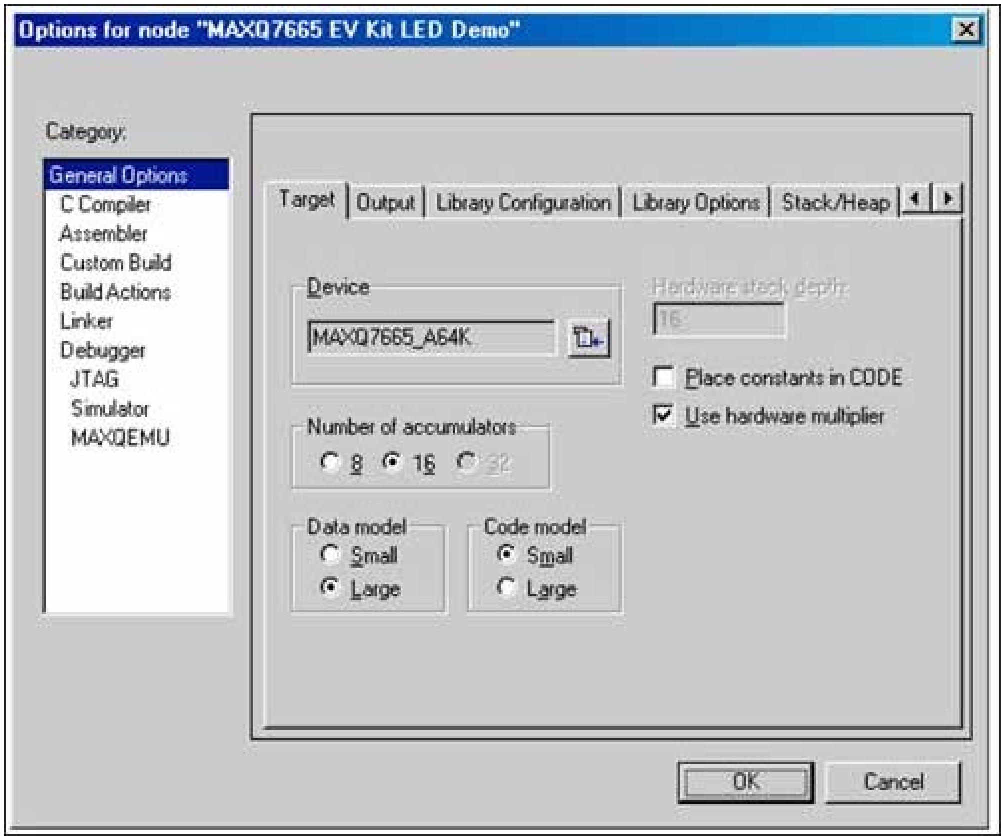


More detailed image

*Figure 1. This workspace for the MAXQ7665 lists the projects contained in the software.*

## Setting IAR Project Options

The project options must be set for each project in the workspace. If a new project is added, its options must be configured. The sample projects included with the EV kit already have their options configured. To view or change the project options, make the project "active" by either right clicking on the highlighted project in the Overview screen or by clicking the project tab at the bottom of the screen. The project name will appear in bold as in Figure 1. Next, go to the Project menu and select Options so the following screenshot will appear. The option for each category can be set, as shown in **Figure 2**.



Device
MAXQ7665_A64K

Hardware stack depth:
16

Number of accumulators
○ 8 ● 16 ○ 32

Data model
○ Small
● Large

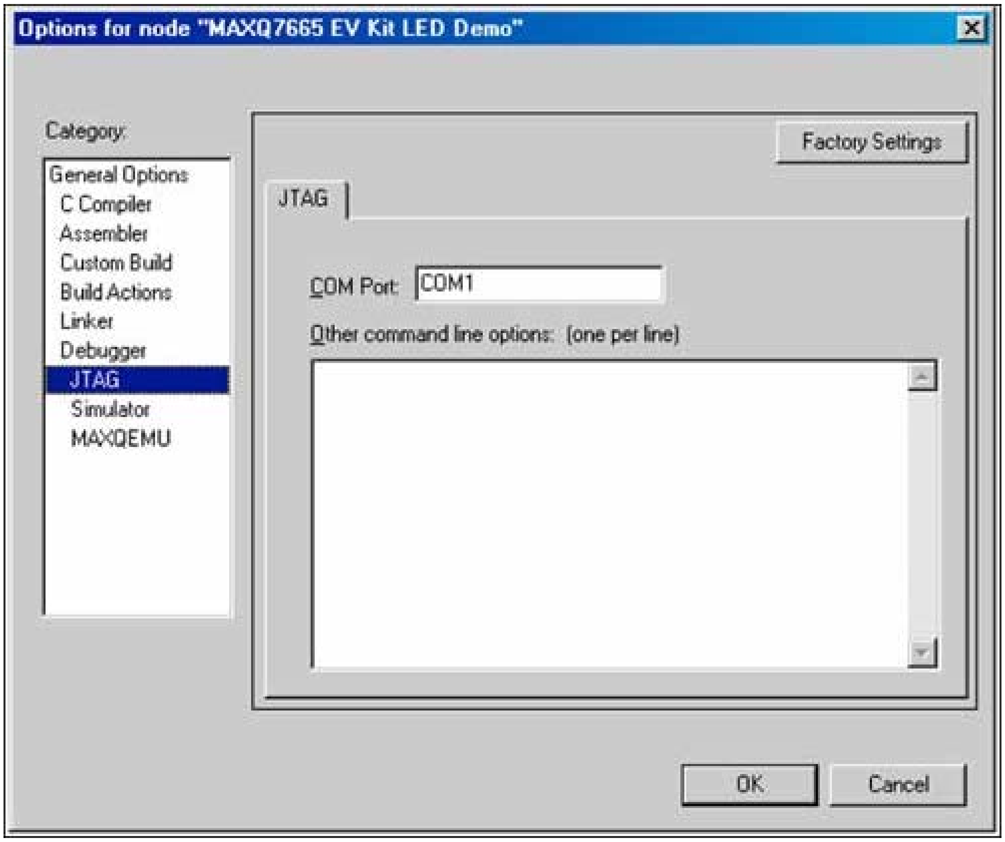
Code model
● Small
○ Large

□ Place constants in CODE

☑ Use hardware multiplier

*Figure 2. The project options menu lets you configure settings for each project.*

The JTAG serial port is the only option that might need to be set for the sample projects. If you select JTAG, the screen in **Figure 3** will appear. In this example we used COM1 as the communication port from the PC to the MAXQ7665 EV kit. To use a different port, change the COM designation to the appropriate serial port.



Options for node "MAXQ7665 EV Kit LED Demo"

COM Port: COM1

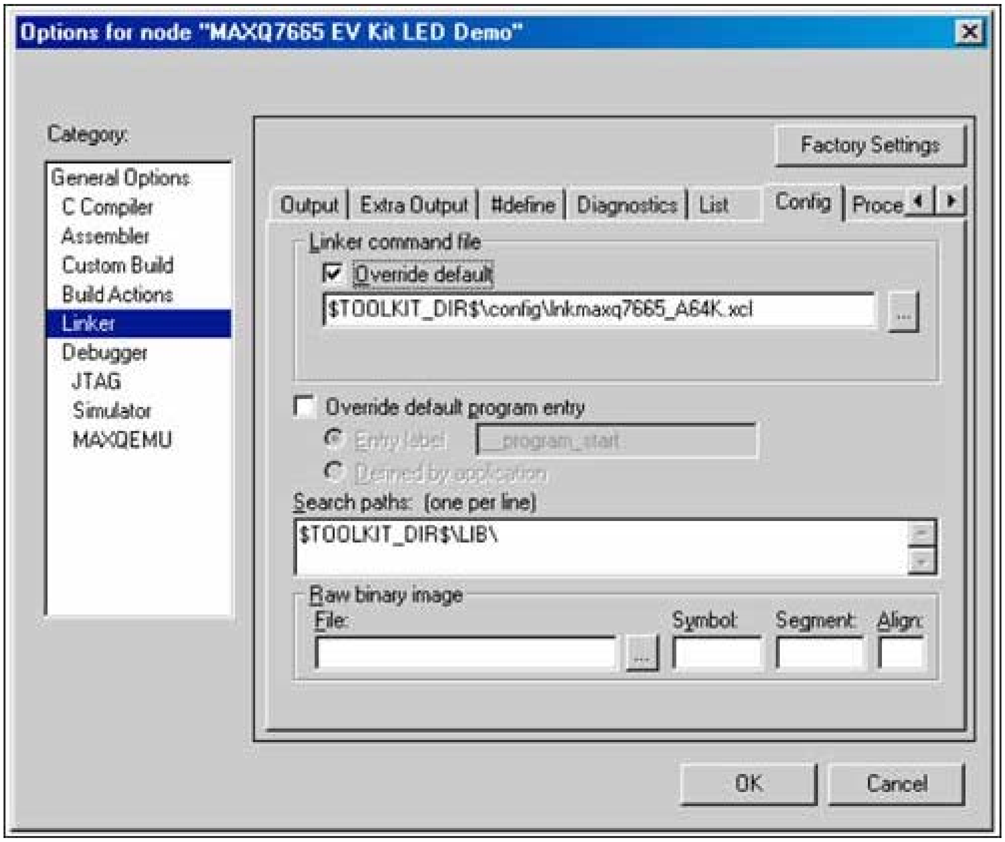
Other command line options: (one per line)

*Figure 3. Communication between the IAR toolset and the MAXQ7665 is set up through the JTAG interface.*

To create new projects for the MAXQ7665 EV kit or the target application, some important options must be set. The optional linker and debugger configuration for the MAXQ7665 are set up using specific files.

## IAR Linker Options

In the Options window select the Linker category and the Config tab, as in **Figure 4**. Select the "Override default" box in the `XCL` file name section, and locate the `lnkmaxq7665_A64K.xcl` linker file in the config directory.



Linker command file

☑ Override default

$TOOLKIT_DIR$\config\lnkmaxq7665_A64K.xcl

☐ Override default program entry

● Entry label __program_start

○ Defined by application

Search paths: (one per line)

$TOOLKIT_DIR$\LIB\

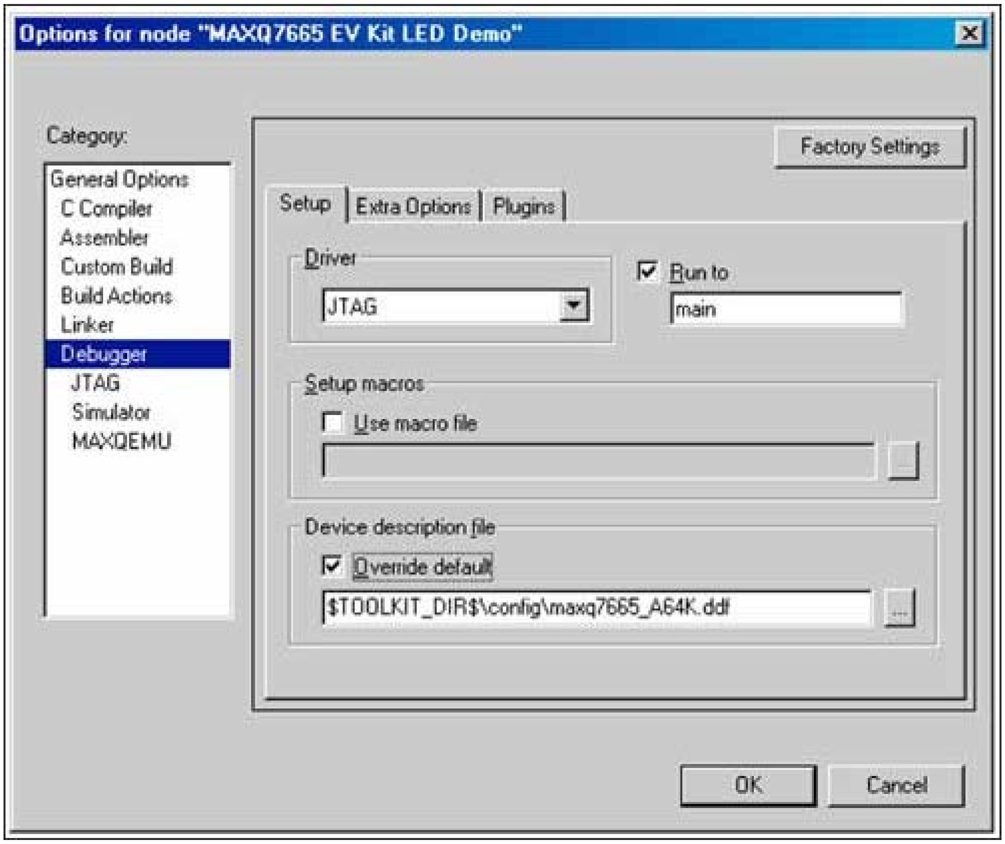
Raw binary image

File: Symbol: Segment: Align:

*Figure 4. This screen lets you configure the linker options for a project.*

## IAR Debugger Options

To set the debugger options, select the Debugger category. Select the "Use device description file" under "Device description file" (**Figure 5**) and locate the `maxq7665_A64K.ddf` file in the config directory.



Driver: JTAG

☑ Run to: main

☐ Use macro file

☑ Override default: $TOOLKIT_DIR$\config\maxq7665_A64K.ddf

*Figure 5. This screen lets you configure debugging options.*

## Sample Projects

The EV kit's Quick Start CD contains several sample projects. Example code demonstrates the features of the MAXQ7665 and the EV kit. The sample projects are programs that can be compiled, linked, and run. The user can create a workspace and project, and then copy code sections from the sample projects to build an application. A brief description of the sample projects follows.

### ADC

This project contains software to set up the MAXQ7665's analog-to-digital converter (ADC). The ADC parameters can easily be changed and passed to the `ADC_Convert_Int` function if interrupts are used. The ADC data will be returned in the interrupt service routine when the conversion is complete and the data ready. If ADC polling is used, uncomment the `ADC_Convert_Poll` function. The value of the ADC conversion will be returned to the function when the ADC has finished conversion and the data is ready.

### CAN

The CAN project contains software to set up the CAN controller. The sample project sets the global CAN parameters and then sets up receive and transmit message centers. To verify the operation, the CAN controller can be set in autobaud (loopback) mode, which transfers the CAN data from the transmit message center to the receive message center.

### DAC

This project contains functions to configure the digital-to-analog converters (DACs) and to generate a continuous sine-wave output from each DAC.

### LED Demo

The LED Demo project sets and clears the MAXQ7665's port bits that drive the LEDs. The result is a simple traffic signal emulation using the LEDs.

### Temp Conversion

The MAXQ7665 has an internal temperature sensor and the EV kit has two external temperature-sensing devices. The temperature conversion project can measure the internal or external temperature.

### Timers

The Timers project demonstrates the setup of timers for 8-/16-bit pulse-width modulators with variable frequency and duty cycle. There is also a delay timer with 1ms resolution using Timer 0.

### UART

The UART project allows the EV kit to interface to an RS-232 serial port. The MAXQ7665 sets the UART parameters and controls the serial data exchange.

### Voltage Monitors

The Brownout Monitors project contains software to demonstrate the functionality and use of the brownout-monitor detection circuitry. The program sets the voltage thresholds for the brownout interrupts; the interrupt service routine breakpoint can be moved to the desired breakpoint for the voltage under test. The user then presses switches on the EV kit PC board to cause the desired brownout interrupt.

## Start the IAR Embedded Workbench

Start the IAR toolset. At the File menu select Open Workspace and go either to the `C:\Program Files\IAR Systems\Embedded Workbench 4.0\MAXQ\src\MAXQ7665 EV Kit` directory or to the directory where the source files were installed. Select the workspace file `MAXQ7665 EV Kit.eww` and click open.

This selection opens a workspace for the MAXQ7665 EV kit that contains several projects. Find the tab labeled MAXQ7665 EV Kit LED Demo (bottom of window; click right), and highlight MAXQ7665 EV Kit Led Demo–Debug. Right click to select Set as Active Project, if it is not already set (bolded on screen).

Right click again and select Options. The project settings are stored here and will be saved. **Note** that the serial port must be set. Select JTAG. Under the COM Port enter the serial port that you are using, if it is not COM1. **Note** that you must use COM#, not just the number of the port. Also some laptops do not come with an RS-232 port, but USB-to-RS-232 converters are available.

On the Project menu select Rebuild All. If all the files were located correctly, the message box should show Errors: None and Warnings: None.

After the step above is completed with no errors, go to the Project menu and select Debug. This will download the application code to the MAXQ7665 EV kit.

Once the application file has downloaded, the debugger starts at the "main" statement. On the Debug menu press the Go command and the LEDs will emulate a traffic signal.

## Basic Troubleshooting

If you get an error message when trying to communicate with the MAXQ7665 EV kit, it will be necessary to close and reopen the IAR Embedded Workbench. If restarting IAR does not fix the problem, try removing the RS-232 cable from the EV kit and reset the EV kit by cycling the power. Then reconnect the RS-232 cable, apply power, and open the IAR toolset. If IAR still cannot communicate with the MAXQ7665 EV kit, review the *Advanced Troubleshooting* section below.

## Advanced Troubleshooting

This section describes the steps necessary to install and use Maxim's Microcontroller Tool Kit (MTK) for debugging the MAXQ7665 EV kit over an RS-232 connection. Debugging may be necessary if the IAR Embedded Workbench cannot download to the MAXQ7665 EV kit.

## Required Hardware and Software

1. PC with Windows® and an RS-232 port
2. MAXQ7665 EV Kit
3. MAXQ7665 EV Kit Install CD
4. AC-to-DC converter with 12V output
5. DB-9 to DB-9 RS-232 cable

## Software Installation

The MTK utility is provided on the MAXQ7665 EV kit's Install CD in the MTK directory. Run the `SETUPEX.exe` file to install the MTK application.

## Hardware Installation

Connect the MAXQ7665 EV kit to the AC-power adapter. Connect the RS-232 serial port cable from the PC to the EV kit's DB-9 connector labeled PC RS232.

## Start the MTK Utility

Once the application is installed, open it and select the DS89C430 from the drop-down menu. This is the device used to communicate between the PC's RS-232 port and the MAXQ7665's JTAG connections.

## Set the MTK Options

Select Options and then Configure Serial Port. Enter the serial port that you are using, and set the speed for 115,200 Baud. Also select "Toggle DTR on connect/disconnect," "Save working directory on exit," and "Save COM Port Settings on Exit," if these settings are not checked already.

## Connect to the MAXQ7665 EV Kit

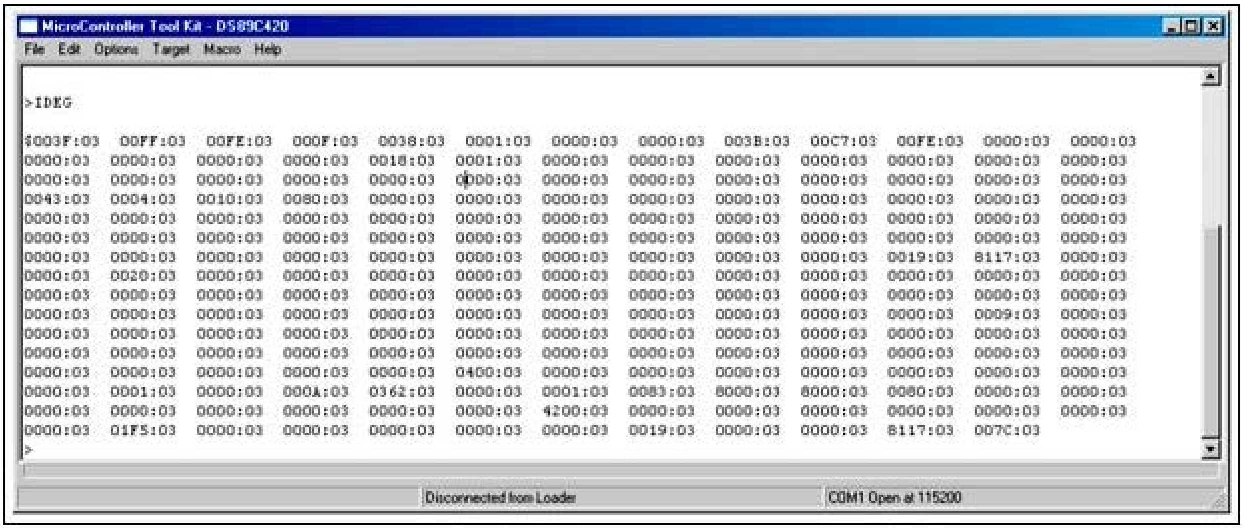
Select the Target and "Open COMx at 115200 baud," where x is the serial port configured from the Options above. Test the RS-232 interface by pressing Enter or Return at the keyboard. The MAXQ7665 EV kit should respond with the prompt character ">" each time that Enter is pressed. If this step is working, it indicates that the microcontroller on the EV kit is recognizing the Enter and is ready for input. If this step does not work, see the *No Prompt* section below.

# Connect to the MAXQ7665 Debug Engine

There is a simple method to determine the state of the MAXQ7665: "Get" all the internal registers. The Get command sends a few basic commands to the debug engine and has the MAXQ7665 send all its internal registers. The commands used are included in the file provided on the Quick Start CD in the MTK directory named `protocol.txt`. Use the following commands to Get all the registers. Make sure that you are using capitals (CAPs), as MTK is case-sensitive. The comments are in parentheses.

>I (Initialize the debug engine and press Enter.)
>D (Enter debug background mode and press Enter.)
>E (Enter debug mode and press Enter.)
>G (Get all registers and press Enter.)

If the MAXQ7665 is working correctly it should reply with the register map shown in **Figure 6**. Note that the registers start with Module 0 Register 0, followed by Module 0 Register 1, etc. Note too that the commands can be combined onto a single line. For example, >IDEG will yield the same result.



MicroController Tool Kit - DS89C420
File Edit Options Target Macro Help
Disconnected from Loader
COM1 Open at 115200

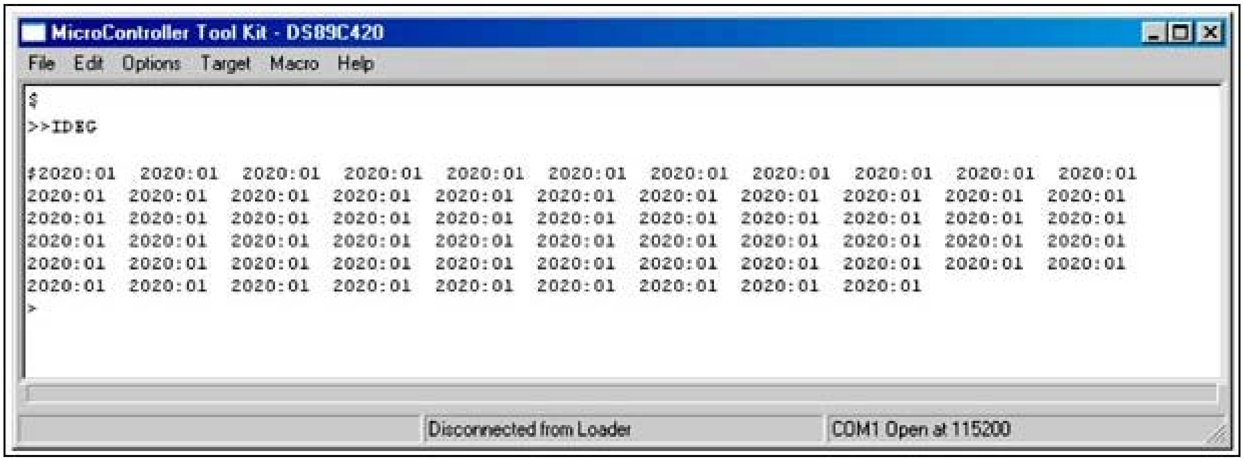
```
>IDEG

$003F:03  00FF:03  00FE:03  000F:03  0038:03  0001:03  0000:03  0000:03  003B:03  00C7:03  00FE:03  0000:03  0000:03
0000:03  0000:03  0000:03  0000:03  0018:03  0001:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03
0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03
0043:03  0004:03  0010:03  0080:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03
0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03
0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03
0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0019:03  8117:03  0000:03
0000:03  0020:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03
0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03
0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0009:03  0000:03
0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03
0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03
0000:03  0000:03  0000:03  0000:03  0000:03  0400:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03
0000:03  0001:03  0000:03  000A:03  0362:03  0000:03  0001:03  0083:03  8000:03  8000:03  0080:03  0000:03  0000:03
0000:03  0000:03  0000:03  0000:03  0000:03  0000:03  4200:03  0000:03  0000:03  0000:03  0000:03  0000:03  0000:03
0000:03  01F5:03  0000:03  0000:03  0000:03  0000:03  0000:03  0019:03  0000:03  0000:03  8117:03  007C:03
>
```

*Figure 6. If the MAXQ7665 is configured correctly, the register map should start with Module 0, Register 0.*

From the register map above the first register from Module 0 and Index 0 M[00,00] is the Port 0 Output Register (PO0). The power-on reset (POR) value of this register is 0x3F, which is correct. Also the next two registers are the Port 1 M[01,00] and Port 2 M[02,00] output registers whose POR values are 0xFF, which is also correct.

If you do not get the register map shown in Figure 6 and instead get the register map in **Figure 7**, reset the MAXQ7665 by pressing and releasing SW10. Once this is done, repeat the IDEG commands and you should get the register map in Figure 6. Note that the register map in Figure 7 is also an acceptable state for the debug engine.



```
$
>>IDEG

#2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01
2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01
2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01
2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01
2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01
2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01  2020:01
>
```

*Figure 7. Alternate Get register map.*

If the previous steps are successful, the communication link from the PC to the JTAG on the microcontroller to the MAXQ7665 debug engine is working. You should be able to close the communication port in MTK and close the MTK application. The IAR Embedded Workbench should download code and debug the application code in the MAXQ7665 EV kit.

## Debug Engine State

The debug engine also returns its state along with the data above. For example, the first register M[00,00] is 003F:03. The four left characters are the register's hex value. The data returned is always 16-bit wide, regardless of the register size. There is a colon and 03 in this example. The 03 is the handshaking between the host and the debug engine. In this case the 03 indicates that the debug is valid. The possible states are shown in the table below.

| Status | Condition |
|---|---|
| 00 | Default Condition. Background mode or debug engine inactive (MAXQ7665 in reset) |
| 01 | Debug Idle. Ready to receive data from the host (ready for download and debug) |
| 02 | Debug Busy. Debug engine is busy without valid data (Password Lock State) |
| 03 | Debug Valid. Debug engine is busy with valid data (ready for download and debug) |

## Troubleshooting

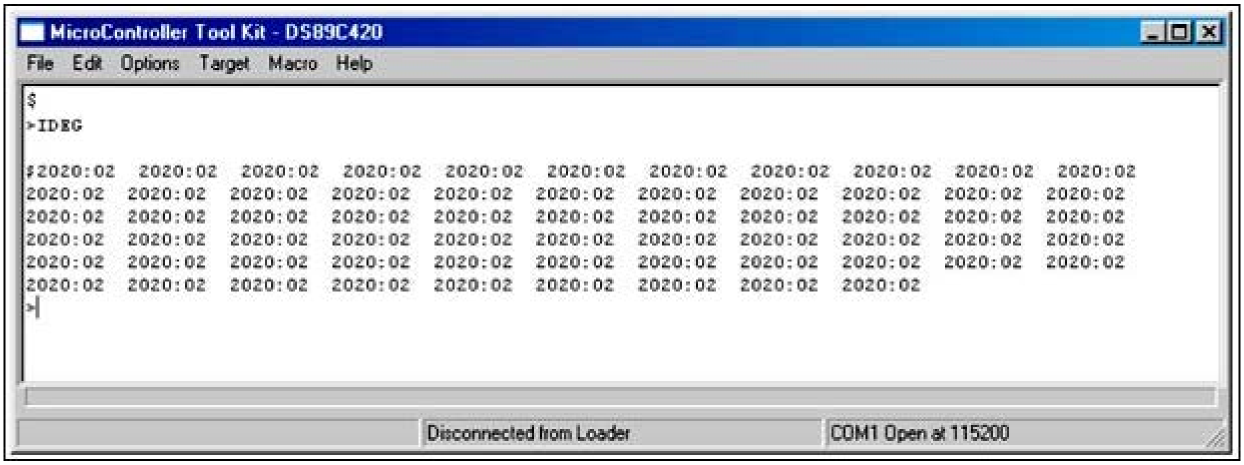
If the action above does not work or you get results other than those shown here, there are some things to try.

### No Prompt

If you cannot get a prompt back from the MAXQ7665 EV kit, disconnect the RS-232 cable from the EV kit and remove the DC power plug. Then reconnect the RS-232 cable and DC power plug. Repeat the procedure if necessary. Also make sure that the serial-port selection in the Options menu is correct for the port that you are using. To see the available serial ports, run the following application. Open the control-panel System application then select the Hardware tab. Select the Device Manager and Ports (COM & LPT) to see which serial ports are available.

### Invalid Data–Debug Busy

Sometimes the IDEG returns data similar to the data shown below, i.e., the data contains :02 for the debug engine state. This value usually indicates that the password is locked for the MAXQ7665, thus preventing the flash from being erased and preventing any communication from the IAR Embedded Workbench to the MAXQ7665. See **Figure 8**.


MicroController Tool Kit - DS89C420
File Edit Options Target Macro Help
Disconnected from Loader
COM1 Open at 115200

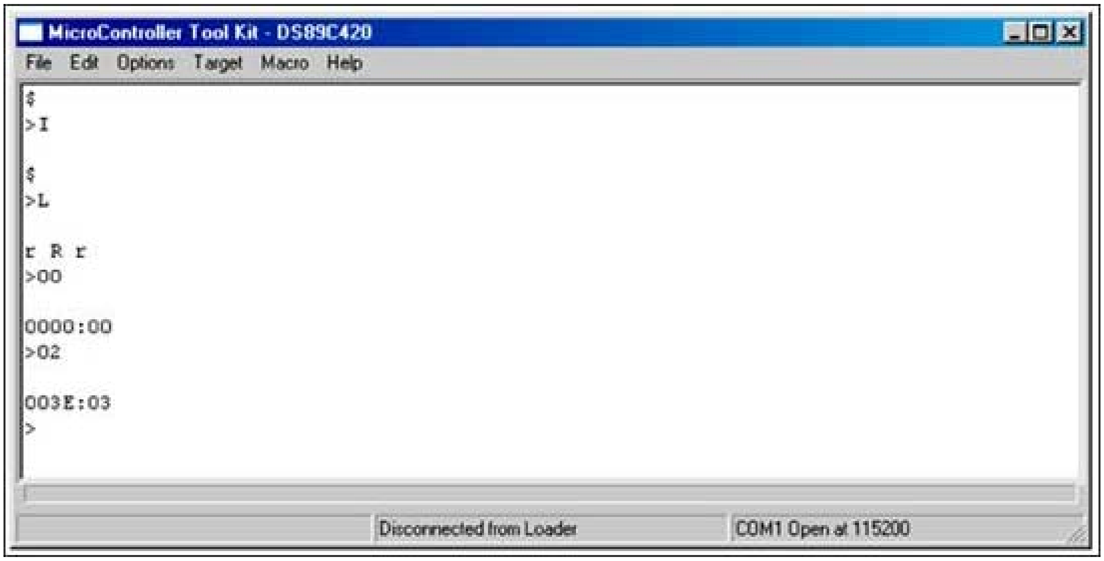
```
$
>IDEG

$2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02
2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02
2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02
2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02
2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02
2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02  2020:02
>
```

*Figure 8. An :02 on the IDEG screen indicates a locked password for the MAXQ7665.*

## Erase MAXQ7665 Flash

The MTK interface can be used to erase the MAXQ7665 flash. The steps necessary for erasure are detailed below in **Figure 9**.


MicroController Tool Kit - DS89C420
File Edit Options Target Macro Help
Disconnected from Loader
COM1 Open at 115200

```
$
>I

$
>L

r R r
>00

0000:00
>02

003E:03
>
```

*Figure 9. Screen shows the steps to erase the MAXQ7665's flash memory.*

| | |
|---|---|
| >I | (Type I. Initialize the debug engine and press Enter.) |
| $ | (Debug engine replies.) |
| >L | (Type L. Enter the bootstrap loader and press Enter.) |
| r R r | (Loader replies.) |
| >00 | (Type 00. Press Enter.) |
| | |

| | |
|---|---|
| 0000:00 | (Loader replies.) |
| >02 | (Type 02. Erase the flash and press Enter.) |
| 0000:02 | (Debug engine replies still busy if unsuccessful.) |
| 003E:03 | (Debug engine replies debug valid. Flash is erased.) |

If the steps above were not successful in erasing the flash memory, repeat the whole process. Remove DC power, hold the reset switch, and enter the commands above. If the erase was successful, you should be able to "Get" all the registers successfully by using the IDEG commands described above. The IAR Embedded Workbench should download application code.

## Support

An application engineer is available for support of the MAXQ7665 EV Kit. For email support, contact: Maxim Support Center. In North America Maxim, call toll-free: (800) 998-9872

IAR Embedded Workbench is a registered trademark of IAR Systems AB.
MAXQ is a registered trademark of Maxim Integrated Products, Inc.
Windows is a registered trademark and registered service mark of Microsoft Corporation.

| Related Parts | |
|---|---|
| MAXQ7665 | 16-Bit RISC Microcontroller-Based Smart Data-Acquisition Systems |

**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 4100: http://www.maximintegrated.com/an4100
APPLICATION NOTE 4100, AN4100, AN 4100, APP4100, Appnote4100, Appnote 4100