Verifying Hardware Functionality via Qt Sensor Demo GUI

Jonathan Sangregario
March 30th, 2012
ECE480 Design Team 2

Keywords: MAVRK, uMAVRK, GUI, Hardware, Sensor, Code Composer, Microcontroller, MSP430.

Summary
This application note is intended to guide the user through the process of using Code Composer Studio v5 to program both the MAVRK and uMAVRK boards with the necessary software to run the Qt Sensor Demo GUI.
Table of Contents

Introduction ........................................................................................................................................... 3
Objective ............................................................................................................................................... 3
Getting Started .................................................................................................................................... 4
Obtaining the Code Repository ........................................................................................................... 4
Assembling the MAVRK Motherboard ................................................................................................. 6
Connecting the Debugger ..................................................................................................................... 9
Connecting the Motherboard to a PC .................................................................................................... 10
Programming the MAVRK MCU .......................................................................................................... 11
Assembling the uMAVRK for Programming ....................................................................................... 17
Programming the uMAVRK .................................................................................................................. 19
Running the Demo GUI ....................................................................................................................... 19
**Introduction**

Beginning with working hardware is crucial in the software development stage. Nearly every user of the MAVRK hardware will at some point desire to modify the software, or develop code from scratch. This is after all, the purpose of the MAVRK development hardware. Should the hardware be faulty, the user will not be able to determine if any errors encountered are a result of their software modifications. Upon verifying the functionality of the hardware, the user can begin the process of modifying the software as needed in order to apply the system to their specific task.

The Qt Sensor Demo Graphical User Interface (GUI) also serves the purpose of familiarizing the user with the process of programming the MAVRK and uMAVRK processors. The user will gain an understanding of how the embedded software library interacts with the hardware components. This demo also reveals the onboard sensing, and RF communication capabilities the MAVRK and uMAVRK process. The end result is a graphical display on a personal computer (PC) which will change according to variations in light and temperature conditions in the surrounding environment.

**Objective**

This application note will instruct the user how to verify that the MAVRK hardware and its various components are functioning as intended. The user will be guided through the steps necessary to assemble MAVRK hardware components in order to establish a Radio Frequency (RF) communication link between the MAVRK processor and the uMAVRK microprocessor. This RF communication link will relay light and temperature data gathered by the uMAVRK to the MAVRK, and ultimately output the data on the user friendly Qt Sensor Demo GUI.
Getting Started

An electronically safe working environment is mandatory when working with and exposed electronics. A safe environment is one that is completely dry, dust free and void of all static. These hardware components do contain elements that are highly sensitive to any static. Thus, it is crucial recommended that the user have a grounding wrist strap attached at all times – see Figure 1.

Prior to initiating this tutorial, it will be necessary that the user has read and understand background information about the MAVRK ([http://processors.wiki.ti.com/index.php/MAVRK_Introduction][1]) and also installed the necessary software ([http://processors.wiki.ti.com/index.php/MAVRK_Software_Installation_Guide][2]). The software installation guide will walk the user through the steps of installing Code Composer Studio v5, and the necessary utilities to pull and push an example code repository from Texas Instrument’s server using TortiseGit. The guide for installing the Qt GUI should be completed before proceeding ([http://processors.wiki.ti.com/index.php/MAVRK_Qt_GUI_SDK_Installation_Guide][3]).

Obtaining the Code Repository

First, we will attempt to pull the example code repository mavrk_embedded.

Begin by creating a new empty folder which will be the designated working area – see Figure 2. This tutorial creates the folder with the path “C:\Project”. Any easily accessible location will suffice.

Next, right click anywhere in this newly created folder and click on the “Git Clone...” option in the menu – see Figure 3.

The window of Figure 4 should appear. Be sure that the “Url:” field matches what is shown and that the “Load PuTTY Key” field is directed to the path of the PrivateKey.ppk file created during the PuTTY installation process (see MAVRK Software Installation Guide).

After verifying these settings, press “OK”. If everything worked accordingly, you should see the following “Success” window – see Figure 5. If an error is returned, review “2.1.5 Cloning a MAVRK Project” of the MAVRK Software Installation Guide.
You should now have a folder named “mavrk_embedded” in your working project folder – see Figure 6. This folder contains libraries of code and various code projects necessary to various demos and basic default functions the MAVRK hardware can perform. Amongst these code projects is the uMAVRK-Generic-Sensor-Demo which this tutorial aims to employ.

Assembling the MAVRK Motherboard

Begin with the motherboard as shown below – see Figure 7.
Attach the MSP430 board as shown – see Figure 8.

![Figure 8: MSP430 Placement](image)

Next, attach the AFE Breakout Board to the motherboard as shown in Figure 9 below.

![Figure 9: AFE Breakout Placement](image)
Be sure that there is a jumper on the 3rd set of pins as shown below. If it’s not present, add it now – see Figure 10.

Place the RF Module in any of the available RF slots – see Figure 11.
Connecting the Debugger

To program the MSP430, first attach the MSP430 USB-Debugging Interface to the JTAG serial port on the MAVRK motherboard – see Figure 12.

![Figure 12: Attaching MSP430 USB-Debugging Device](image1)

Connect one end of the USB cable to the MSP430 Debugger (Figure 13) and the other end to an open USB port on the PC (Figure 14).

![Figure 13: Connecting USB to the Debugger](image2) ![Figure 14: Connecting USB to the PC](image3)
Connecting the Motherboard to a PC

Attach the power USB to the right hand USB port on the MAVRK motherboard – see Figure 15.

![Figure 15: Connecting USB to the Motherboard](image)

After, attach the other end of the USB cable to another open USB port on your PC. You should notice various Light-Emitting Diodes (LEDs) illuminating on the MAVRK motherboard, AFE breakout board, and RF module.
Programming the MAVRK MCU

Open Code Composer Studio v5 (CCS). A window similar to the one below should appear – see Figure 16.

![Code Composer Interface](image)

To import the uMAVRK-Generic-Sensor-Demo project, click “Project” from the menu bar, then select “Import Existing CCS/CCE Eclipse Project” – see Figure 17. The window pictured in Figure 18 should appear.

![Import Project Interface](image)
Click on “Browse…” and navigate to the project folder you created earlier that now contains the mavrk_embedded code library. Once in this folder, open the “mavrk_embedded” folder. Expand the folder tree named “Modular_EVM_Projects”. Continue by opening the “Component_Demo_Projects” folder. Select (by clicking once) on the “uMAVRK_Generic_Sensor_Demo_Project” and press “OK” – see Figure 19. For our demo purposes the desired path is:

C:/Project/mavrk_embedded/Modular_EVM_Projects/Component_Demo_Projects/uMAVRK_Generic_Sensor_Demo_Project
Your window should now match the image as shown in Figure 20. Click on “Finish” to proceed.

In the “Project Explorer” pane, right-click on the project labeled “uMAVRK_Generic_Sensor_Demo” and navigate to “Build Configurations/Set Active”. Click on the “1 Radio Reciever (F5438 Main Receiver)” option – see Figure 21.
From the menu bar, click on “Project/Build Project” – Figure 22.

Upon completion, the Console pane should read “**** Build Finished ****” – see Figure 23.

Program the MCU by pressing the “Debug” button – Figure 24.
After the MCU is programmed, run the program by pressing the “Resume” button – see Figure 25.

You may now disconnect the JTAG connector from the MAVRK motherboard – see Figure 26.
Assembling the uMAVRK for Programming

Locate the RF-UDebug board. Be sure that there is a jumper as shown in Figure 27. If the jumper is absent, place one on the appropriate pins before proceeding.

Attach the uMAVRK via the 40-pin connector – see Figure 28.
Next, attach the MSP430 USB-Debug-Interface to the JTAG connector on the RF-UDebug board as shown in Figure 29. A yellow LED on the uMAVRK should illuminate – see Figure 30.

![Figure 29: Attaching the uMAVRK](image1)

![Figure 30: uMAVRK LED Illuminating](image2)

### Programming the uMAVRK

Go back into the CCS window and end the debug session (which was started when programming the MAVRK) by pressing the “Terminate” button – see Figure 31.

![Figure 31: Terminating the Existing Debug Session](image3)
Set the build configuration as before, only this time select “2 Sensor Transmitter (FS137 uMAVRK Sensor” – see Figure 32.

![Figure 32: Terminating the Existing Debug Session](image)

Program the uMAVRK by pressing the “Debug” button – see Figure 33.

![Figure 33: Terminating the Existing Debug Session](image)
Finally, press the “Resume” button to initiate the communication link between the MAVRK motherboard and the uMAVRK – see Figure 34.

Upon successful link creation, both the MCU and uMAVRK boards should exhibit blinking yellow LEDs.

Running the Demo GUI

Next, find the location where you’ve installed the Qt Demo GUI and run the MAVRK App Monitor application.

Upon opening the Demo GUI, a popup should appear to register the uMAVRK light sensor. Simply press “OK” – see Figure 35.
A second popup should appear attempting to register the temperature sensor. Press “OK” once again.

The resultant display should include a light meter, temperature meter and numerical temperature readout – see Figure 36.

Verify the functionality of the light meter by covering the uMAVRK device. The temperature sensor can be verified by applying heat to the uMAVRK (ex. breathing on the device).

Now that the hardware has been confirmed to be functioning properly, you may begin modifying any of the example code, or creating code of your own to suit the needs of your specific objective.