

Programming the Mavrk System For Wireless Temperature and Humidity Sensing

Kevin Yeh

11/09/11

Introduction

The uMavrk and Mavrk was designed to be a multipurpose embedded wireless system. Although there is a library of sample code, there are no clear instructions on how to use these resources, and more importantly, where to start in customizing them to your specific needs. This Application note will take you from unboxing the Mavrk boards to setting up a working wireless sensing system using their version of the sensor boards. It will also steer you away from common pitfalls in setting up this system. Then it will teach you how to customize it to your needs. Although we are using Temperature/humidity sensing module, we are essentially going to be able to take a voltage applied to the input pin of the uMavrk, send it over to the Mavrk wirelessly, and display it on a PC. This is a pretty general-purpose functionality that can be found in many other applications, not just sensors.

Steps

Estimated time to complete: 20 hours.

What you will need:

- uMavrk
- Mavrk Motherboard
- MCU-430F55438A Board
- AFE-Breakout-MVK Board
- RF-CC1101 Board
- uMavrk Debug Board
- uMavrk Sensor Board
- USB cable to connect TUSB and PC
- MSP430 USB Debugging Interface

These boards should be in the starter kit that you have purchased.

I)

Download and Install Code Composer 5. This is an important step that should be read in detail. If you are not paying for an IDE, **do not** use IAR like instructed in TI's written guide. The free version of IAR will not work past 8k of code, which means 80 percent of TI's modules will not compile. On top of that only Code Composer 5.1.0.0900 or above will work. *Previous version of Code Composer 5 will not work in building your modules.* Also, if you receive an error for licensing upon a build, download the correct .lic file from

<http://the-keygen.com/search/code-composer-studio-5-licence-file>

Click on help->About Code Composer Studio Licensing Information->Manage tab->Add And find your .lic file to add.

II)

Before you can do anything, you need to download TI's available pre-written software modules. TI uses a GIT repository where they update changes to these modules. You need to clone a repository so you can download their modules and receive any changes they have made. This process is actually written in good detail in this the software installation guide written by TI at this url:

http://processors.wiki.ti.com/index.php/MAVRK_Software_Installation_Guide

Note: After the completion of this guide, you SHOULD have a folder that contains two folders that you have downloaded: mavrk_embedded and mavrk_qt_tool.

The rest of this App note will work with these two folders.

III)

Install the QT application. TI's tutorial is very straightforward:

http://processors.wiki.ti.com/index.php/MAVRK_Qt_GUI_SDK_Installation_Guide

Configure your QT programs for correct build. You can follow TI's tutorial here:

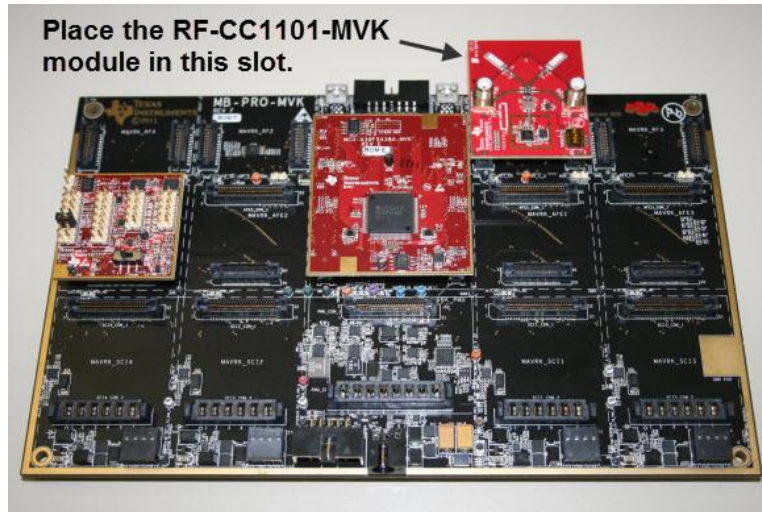
http://processors.wiki.ti.com/index.php/MAVRK_Qt_GUI_Build_Guide

However, there is one thing to note. Near the end of this tutorial, the qexternalportd1.dll should appear in the same folder as your executable. If it doesn't, find this file, copy, and paste it into this folder. It will work afterwards.

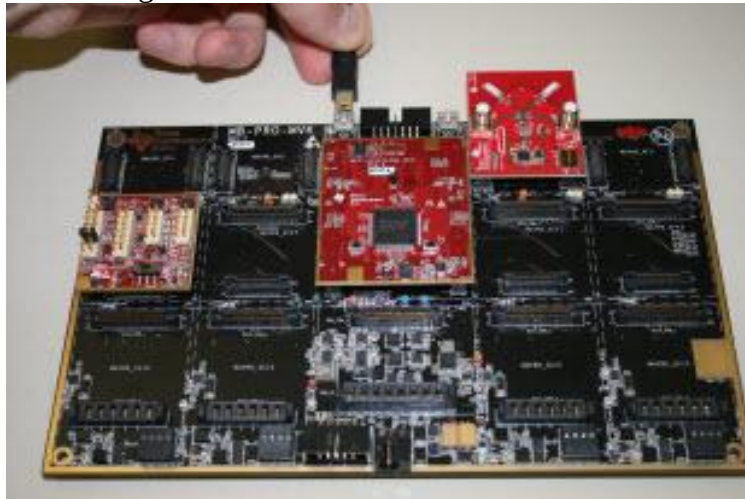
IV)

By now you should have the correct version of Code Composer 5 and the cloned software repository. This step will set up the connections for programming the uMavrk and Mavrk and load the modules. Do not follow TI's guide for loading the modules. It is outdated.

The left is the AFE break out board, the middle is the MCU, and the right is the RF-CC1101-MVK. Attach these as shown.



Connect the usb cable from your pc to the *left* slot in the mother board. This will power your entire board including the modules attached to it.



Connect the Debug Interface between the motherboard and to another usb slot in your pc as shown. This will be used to load actual program code to the MSP.



Open up CCS5. Go to Project->Import existing CCS/CCE Eclipse Project.
Browse for the project at this location:
Your Cloned Repository/mavrk_embedded/Component_Demo_Projects/uMavrk-TMP112_Sensor_demo_Project.

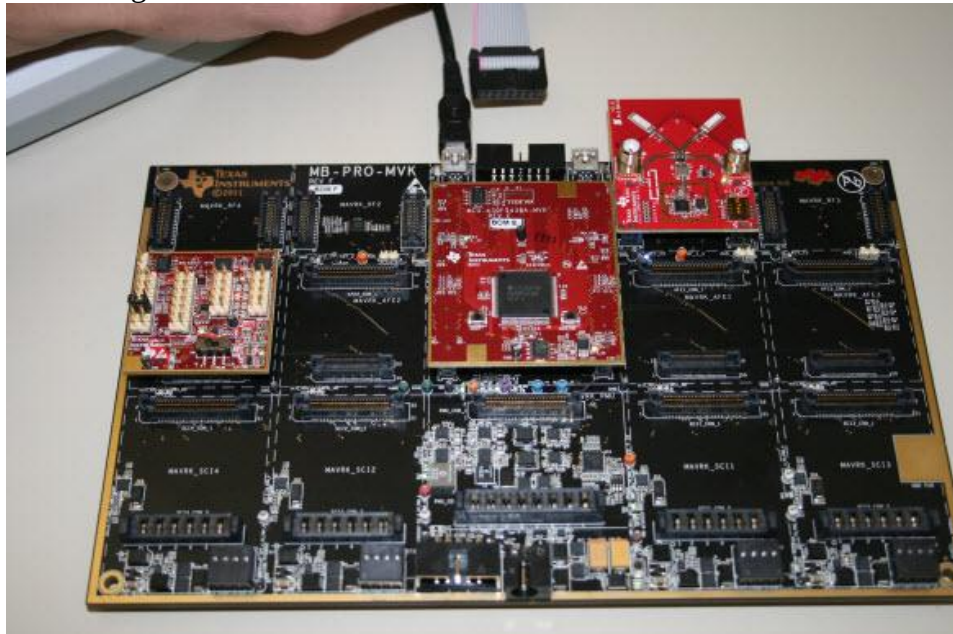
Click Finish.

On the Project Explorer on the left, Right click ->SetActive->Radio Receiver.
The module is now in Receiver mode.

Project->Build Project.

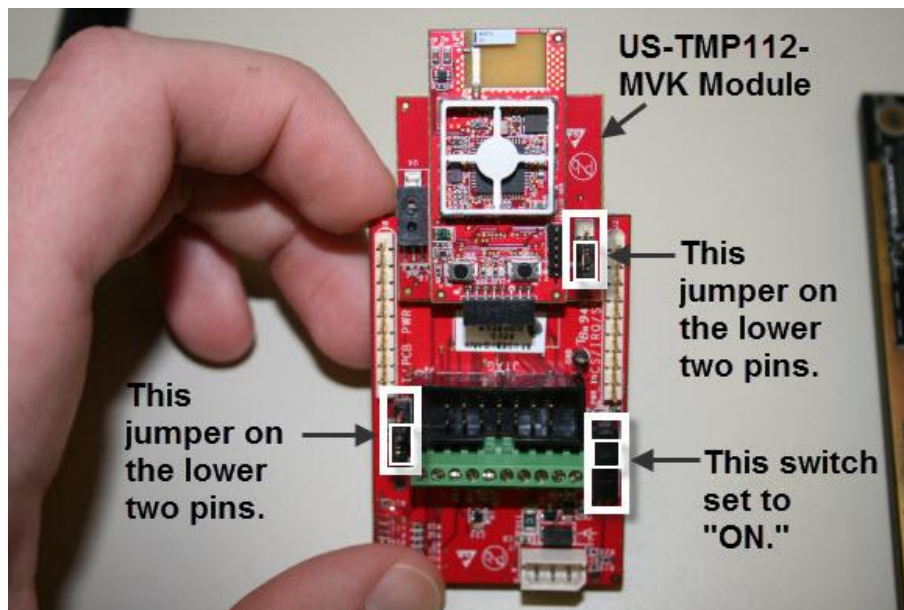
After a successful build, click Project->Debug.

Remove the debug interface from the Motherboard.



Press the stop sign to stop the debug session.

Using the uMavrk, sensor board, and the debug board, follow the picture as shown.



Attach the Debug Interface to the debug board like so:



Once this is setup, go to the same project you have opened.

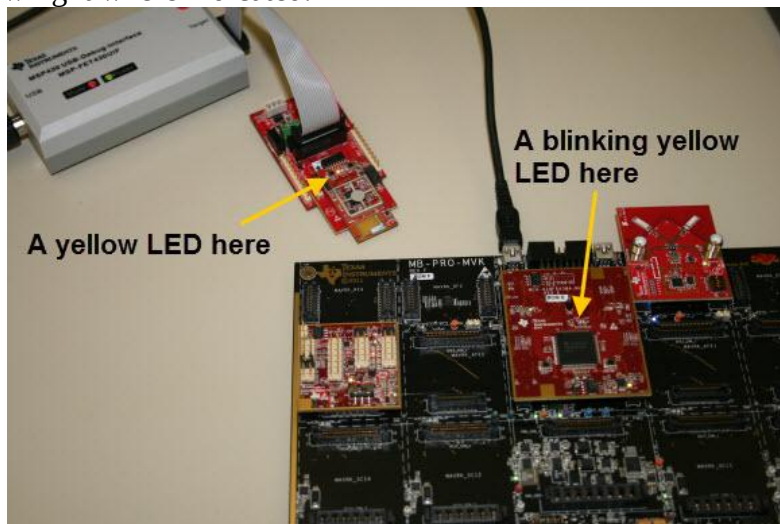
Right click->Build Configurations->Set Active->Sensor Transmitter. This sets the code to Transmitter Mode.

Right click-> Clean Project

Project->Build Project

Run-> Debug

After completion of these steps, look at the uMavrk and Mavrk. There should be a blinking yellow light where indicated.



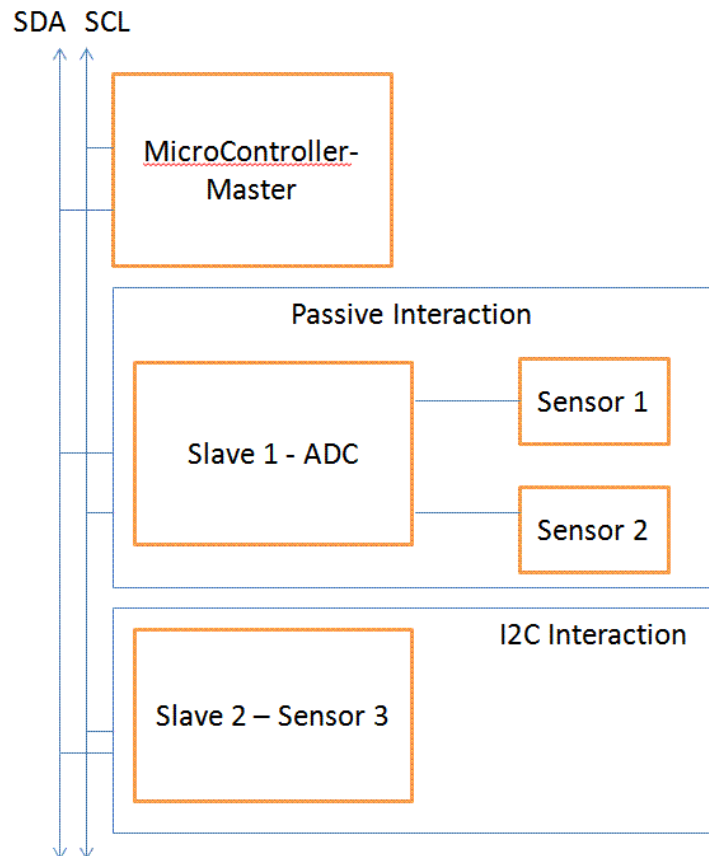
Launch the GUI for this demo. You should already have the build executable ready to be run from step III.

You should see a GUI that displays Temperature, Light, and Humidity data. This is good verification that your system is connecting properly over the RF UART.

V)

Lastly I will describe key information about the code which will help users customize this code to their purposes. Remember, although this is code for temperature/humidity sensing, we are interested in making this a general purpose code for transmitting nothing more than a voltage value.

The uMavrk uses an I2C connection to receive and transmit information to its peripherals. In our case, the peripherals are the atd converter, and the temperature sensor. We want to change the Temperature sensor interaction from i2c to a *passive* interaction. For an i2c interaction, the cc430 will address the sensor using the SDA bus directly. Once it is successfully addressed, it will receive its digital value (an integer value in 7 bit binary that represents temperature). Changing it to a passive interaction will mean that the code will look for a voltage value from the atd converters instead of addressing the digital value from the sensor.



Replace the sensor board with your customized pcb board. Remember, that the signal that you output has to be between 0 to 2.4 volts.

Set the code to Transmitter mode.

Delete line 231 for initializing of TMP112.

Delete line 261.

Replace line 298 with :

A function call to a function that you write: It will take in two parameters

&temp_radio_handle

ADC_Conversion_data_user_defined_array

This function has the purpose of taking in two voltage values.

Conclusion

After this tutorial, you should have a working wireless network between the uMavrk and the Mavrk. You should also have the ability to transmit a general purpose voltage across the RF connection and interface with the PC.