ARA Smart Power Supply Microcontroller Guide

Introduction

This document details how to set up programming environments for the ARA Smart Power System microcontrollers.

In general, the ASPS microcontrollers are programmed in the Energia framework. The ASPS-Power and main ASPS-DAQ microcontroller are programmed entirely from the Energia environment. The *heater* microcontroller on the ASPS-DAQ is *written* in the Energia framework, however some regarding optimization resulted in the Energia code being *imported* into Code Composer Studio, and compiled from there. This situation is evolving, and this document will be updated with information on recompiling the heater microcontroller.

Tools required

- Energia 17. Note that this is *not* the most recent version of Energia. However Energia 18 was a major shift, and while the improvements in build infrastructure were significant, at the moment it is easier to just stick with Energia 17.
- Git
- LM Flash Programmer
- (optional for heater microcontroller) Code Composer Studio

Step 1: Download and extract Energia 17

Energia 17 does not “install” in the sense of running an executable under Windows or installing a package under Linux: it’s just a zip/tar-gz archive which is extracted and is immediately runnable.

Therefore the first step is to download the archive and extract it.

Note that there will be a few changes to the Energia 17 framework specifically for the ARA microcontrollers, so if you already have Energia setup, it might make sense to download a new copy of Energia specifically for the ARA microcontrollers.

Step 2: Update the hardware/msp430 and hardware/lm4f directories

Inside the energia-0101E017/ directory is a ‘hardware’ subdirectory which contains the specific Energia implementations for those microcontrollers. Under the hardware subdirectory, the msp430 and lm4f directories need to be changed.

1. Go to the hardware subdirectory.
2. Delete the ‘msp430’ and ‘lm4f’ subdirectories.
3. git clone https://github.com/ara-daq-hw/energia-hardware-msp430 msp430
4. git clone https://github.com/ara-daq-hw/energia-hardware-lm4f lm4f

Step 3: Run Energia to create the directory structure for Energia sketches

Energia stores sketches and libraries in a separate directory (under Windows, this is under Documents/Energia). To create those directories, simply run Energia. The executable should be in the energia-0101E017/ directory.

Energia will warn that there is a newer version of Energia when it runs. Click "no."

Step 4: Clone the repository containing the microcontroller code

From the Energia sketch/library directory, clone the desired repository. For example, the ASPS-DAQ repository would be

```
git clone https://github.com/ara-daq-hw/asps_daq_basic asps_daq_basic
```

Step 5: Copy the libraries from the repository to the Energia library directory

Under the sketch directory there will be some number of libraries used in the microcontroller code in a libraries/ subdirectory. Copy those to the Energia/libraries directory.

Step 6: Open the sketch and select the microcontroller

Run Energia, and then open the microcontroller sketch. Next you will have to select the microcontroller, under Tools->Board.

- For the ASPS-DAQ, select "LaunchPad (Tiva C) w/tm4c129"
- For the ASPS-Power, select "ASPS-POWER w/msp430f2274"

Step 7: Compile and load.

In Energia, the "Verify" button actually compiles the code.

At this point, if you have a programmer directly hooked up to the microcontroller that Energia can use, select the serial port that it connects to, and use the "Upload" button and it will program it directly.

For the ASPS-DAQ, to program the system via the Ethernet bootloader, use the LM Flash Programmer. In Energia, click 'Verify' to compile, then Sketch->Open Compilation Folder. In the compilation folder, copy "asps_daq_basic.cpp.bin" to someplace else. This is the compiled binary.

In LM Flash Programmer, change the "Interface" to Ethernet. Enter in the IP Address and MAC address of the board to reprogram (from the ASPS-DAQ command line these can be obtained via getmac and ip).

Note: you *must be on the same physical network* as the device. You can figure this out from your IP address and netmask – go here http://jodies.de/ipcalc and enter your IP address and netmask, and write down what it gives as your network. Then enter the IP address of the device, and compare that network to yours. If it's the same, you're on the same physical network.

Then click on the “Program” tab, and select the bin file wherever you copied it to. Finally, click on the “Program” button, and it will reprogram the ASPS-DAQ over Ethernet.

Note: the ASPS-DAQ uses the “Ethernet Bootloader” class for reprogrammability. You can see that in the “EthernetBootloader boot;” line, as well as the “boot.begin();” which enables the bootloader. *If you mess with the Ethernet bootloader, this will not work,* and you’ll have to reprogram it via the serial port procedure detailed on GitHub.