Laboratory 1: Getting Started with Energia and MSP430 Experimenter's Board

Equipment Required:

1 × MSP430FR5739 Experimenter Board Curtin's Kit (USB programming cable, breadboard, wires, Servo Motor)
1 × PC with Energia software installed

1. Objectives

The purpose of this laboratory is to get started with the Energia integrated development environment (IDE) and use it to program an MSP430 supported development board, the MSP430FR5739 Experimenter Board by Texas Instrument.

The objectives of this laboratory session can be summarised as follows:
Learn how to use Energia IDE.

Get familiar with the MSP430FR5739 experimenter's board: Programming, compiling and loading.

2. Pre-laboratory

Before coming to the laboratory:

- Get familiar with the MSP430FR5739 experimenter's board, which will be referred to as the MSP430 development board or MSP430 board in this unit, by **reading the user's manual** available from Blackboard (Bb). Write any useful notes in your logbook.
- Review on Pulse Width Modulation (PWM) for the MSP430 and study the code from https://energia.nu/guide/tutorials/basics/tutorial_fade/
- Read this laboratory script and make sure you understand what is expected.

3. General Recommendations

Some general recommendations to follow:

- Follow your lab supervisor's safety instructions.
- This lab is an introduction to the hardware and software and is not marked. So, please take your time understanding the software and hardware and feel free to ask your lab supervisors for advice. Keep in mind the lab supervisors will only give you general directions and any debugging of the code will be on the students.
- You can download the Energia software and device driver from: Energia download link
- Please download version 16 because the later versions have some bugs specially for people who are using Windows 10.
- IMPORTANT: If you want to keep a copy of your work projects, compress the whole directory into a ZIP file and copy it to your thumb or I:\ drive.

4. Introduction

The MSP430 board is packed full of features. In this unit, we will be exploring some of the capabilities of the board by performing certain tasks which will make use of some inbuilt interfaces to the external world.

Working with the MSP430 board requires some prior knowledge of the external pin functions for program setup and hardware interfacing. The pinout is shown in Figure 1 is particularly useful.

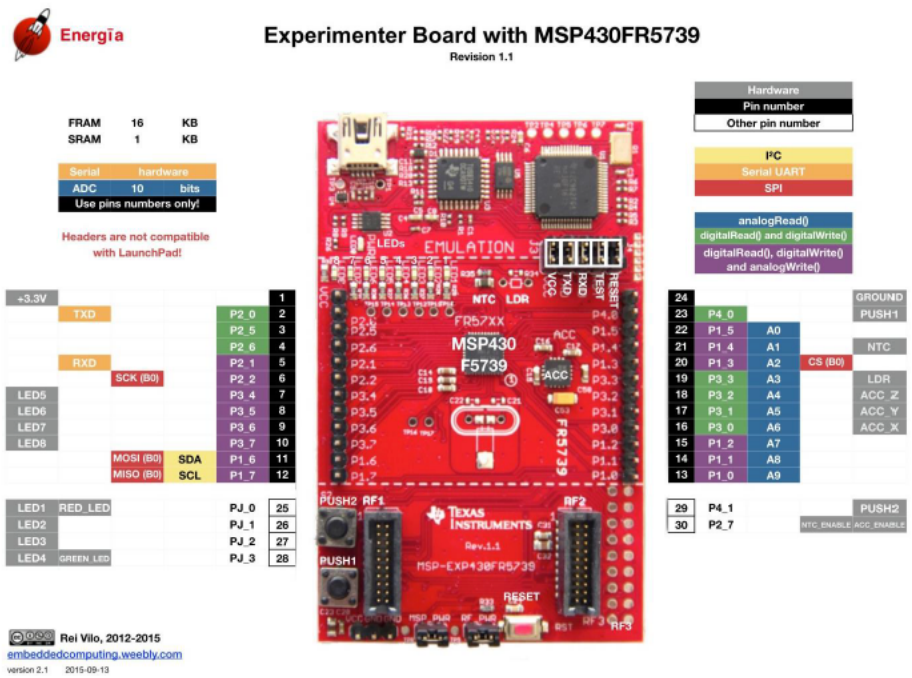

Figure 1 MSP430FR5737 pinout.

5. Tasks

1. After connecting your MSP430 to your computer, open Energia and go to Tools →Board and choose FraunchPad w/MSP430FR5739.

2. Go to Tools →Serial Port and choose the COM port connected to your MSP430. If you are working with the computers in the lab, it will be any COM port other than COM 1.

3. Turning a **blue LED** ON or OFF (Source Code: Blink from Energia)

The example sketch provided by the Energia IDE under the File →Examples (Blink) will allow you to pulse an LED on the experimenter's board using a software timer. Compile and upload to the MSP430 board. Show the programs running to your lab supervisor to get the marks for this task. Make sure that you record down important notes in your logbook that will help you achieve this task.

For this task, you will have to study the Blink sketch and get any hints from the comments to setup the correct blue LED for blinking. Define the correct BLUE_LED definition in your code by attaching to the correct pin by using the pin mapping shown in Figure 1. Also structure your code to make easier to modify for future changes. It is also a good practice to make use of the Help menu. For example, use Help if you do not know how to use a particular function by looking for the Language Reference.

4. Flash LEDs in Round-robin Sequence (Source Code: Round Robin Example from blackboard)

Modify your Blink sketch to toggle in a round-robin sequence, the four LEDs (use the pin map in Figure 1 for help). Remember to save as a different sketch if you want to keep a record of all your programs. Experiment between the delay times and use a delay that is not too short for the sequence to be visible. Show this task to your lab demonstrator upon completion.

5. Working with an Interrupt (Source Code: Push Button Example from blackboard)

Change your Task 4 sketch to round-robin sequence the four LEDs when an on-board pushbutton is pressed. In order to detect key presses, you will have to detect on interrupt. Setup your sketch so that the MSP430 operates in the low power mode. Make sure you understand what is happening with your sketch when you upload to the MSP430 board. Record any notes in your logbook.

For this task, toggle between any two chosen on-board LEDs when a button is pressed. Show this task to your laboratory supervisor upon completion.

An example on what your sketch may contain is shown below. This example is taken from the Energia Reference Help. You may look up for attachInterrupt and study the function and record down any important notes. Pay attention to the Note stated in the attachInterrupt reference and understand what it means. The example shows the use of the Serial function for printing results in the Serial Monitor (Ctrl+Shift+M). Using the serial monitor is useful to display status of your sketch and also assist in code debugging purposes.

Example

```
volatile int state = HIGH;
volatile int flag = HIGH;
int count = 0;
void setup()
{
	Serial.begin(9600);
	pinMode(GREEN_LED, OUTPUT);
	digitalWrite(GREEN_LED, state);
	/* Enable internal pullup.
	* Without it, the pin will float and the example will not work. */
	pinMode(PUSH2, INPUT_PULLUP);
	attachInterrupt(PUSH2, blink, FALLING); // Interrupt is fired whenever button is pressed
}

void loop()
{
	digitalWrite(GREEN_LED, state); //LED starts ON
	if(flag) {
		count++;
		Serial.println(count);
		flag = LOW;
	}
}

void blink()
{
	state = !state;
	flag = HIGH;
}
```

6. Connecting a Servo Motor to MSP430 (Source Code: Servo Sweep from Energia)

In this example you will be connecting a servo motor to your MSP430. Servo motor have 3 pins, VCC, GND, Signal. Connect the VCC pin and GND pin on the respective pins on your MSP430. Connect the signal pin to one of the generic GPIO pins of your microcontroller. Under the built in Energia examples, find the servo sweep example. This code generates a PWM signal to sweep the servo motor. Modify the pin in the source code to the one you have connected the signal pin into. Observe the behavior of the servo motor and try modifying the code so the servo motor goes to different angles.

7. Working with the onboard Temperature sensor (Source Code: NTC_FRAM from blackboard)

Download the NTC_FRAM source code from blackboard and put the program on the board. Note that the header file and library has been written in C++. This program read the data from the temperature sensor on the board and displays it on the serial monitor which can be seen on the top right of your Energia window with a magnifying glass icon.