1 Installing the Educational Kit

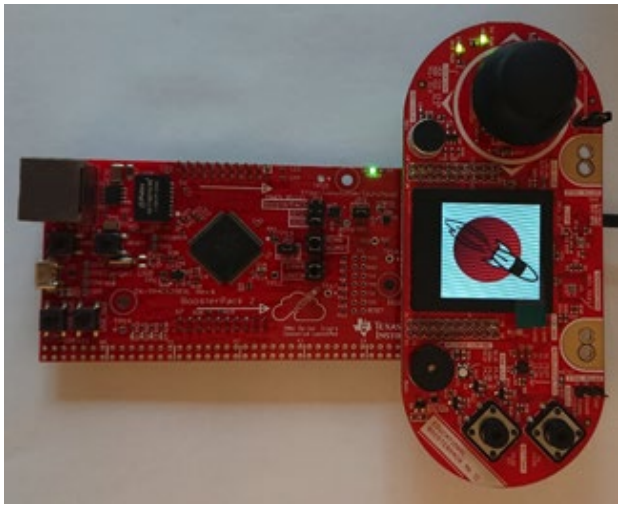
Programming the EK-TM4C129EXL chip from scratch is time consuming and complex. To make it faster and easier to develop software there is a Software Framework which contain drivers for the board. The Framework provide good functionality at the expense of less flexibility of the more exotic options. A good enough trade-off in many cases.

In this lab, you will investigate how to use the analog to digital converter driver (ADC) and Pulse Width Modulation (PWM) modules.

Note: Energia IDE presented below is an open-source IDE. Due to security reasons, installation of Energia IDE in the Lab computers is not supported by the IT helpdesk of MDH. The assignments described in Section 3 can also be solved in two ways: (i) using code composer studio or (ii) Energia IDE. We leave the choice of solving the assignments to you. For the students who would like to use the option of Energia IDE by installing it to their personal computers, we provide a guideline on how to install the software. **You are only allowed to use Energia APIs in this lab, not in the other ones.**

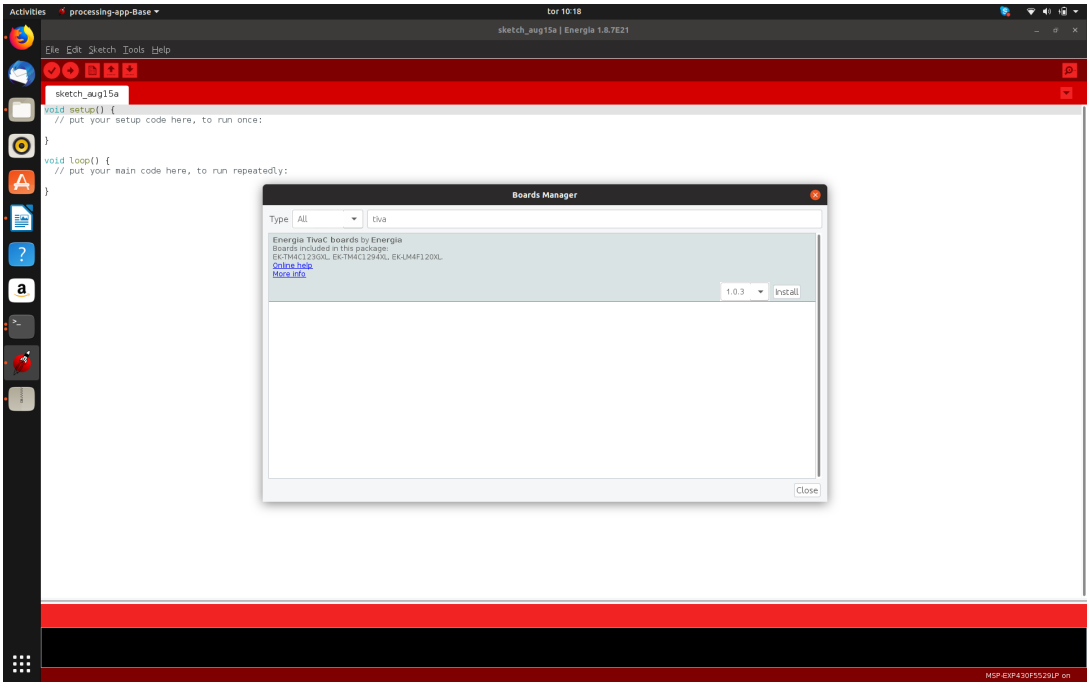
1.1 Hardware Setup

In this lab, you will use the booster pack provided to you, to generate a potentiometer signal. Start to connect the Educational Kit to the booster pack board as shown in the following figure.



1.2 Install Driver

Once the booster pack is connected, you should unzip the file “energia_drivers” and double click "Install Energia Drivers for Windows 32 bit" for Windows 32 bit or "Install Energia Drivers for Windows 64 bit" for Windows 64 bit. This installs the energia drivers for your computer, which provides a complete framework for the booster pack.



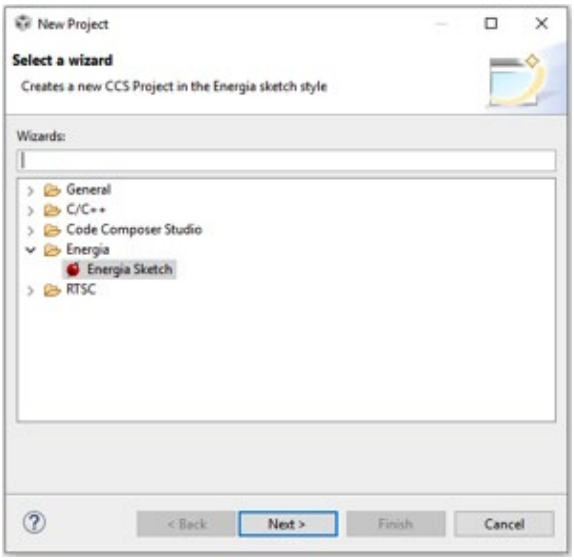
1.3 Install Software

The second step is to install the energia software, unzip the file “energia-1.6.10E18-windows”. The zip archive contains only an executable and does not require installation. Before you can start working with energia integrated in CCS, you also need to install the board support package (BSP) for the EK-TM4C129EXL. Start the energia IDE and go to tools → board → board manager. From the board manager, you need to install the BSP for TivaC boards, so search for TivaC and then press install.

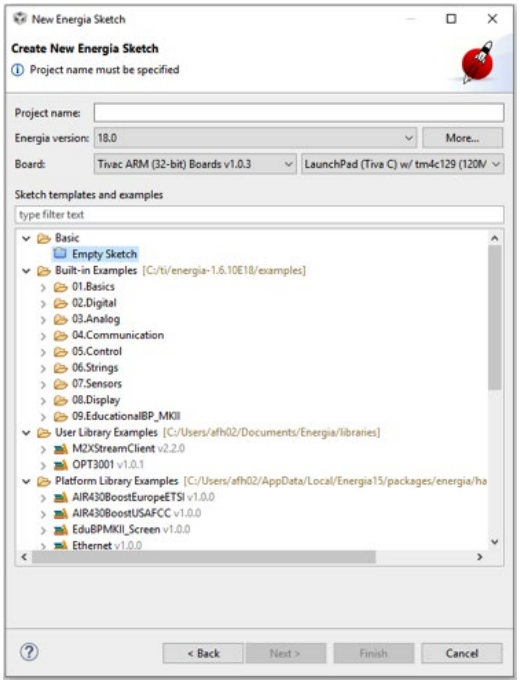
1.4 Create a Project

To create a compilable project, use the following steps:

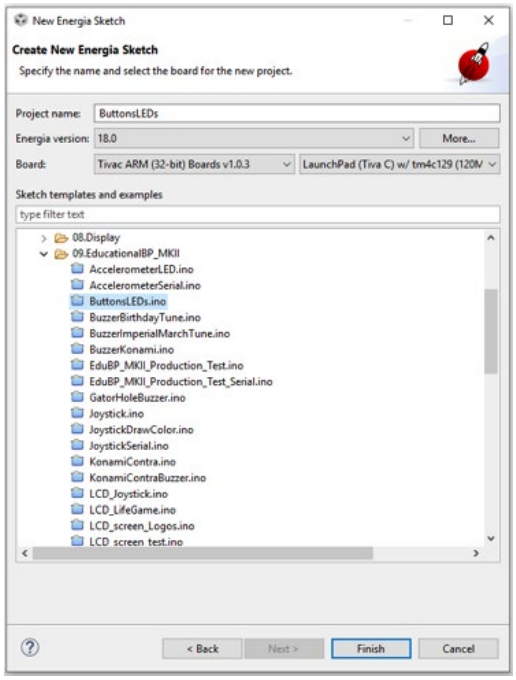
1. Start Code Composer Studio.
2. Select File -> New -> Project -> Energia -> Energia Sketch



3. Specify the path to energia-1.6.10E18-windows.
4. Make the selections shown in the following figure.



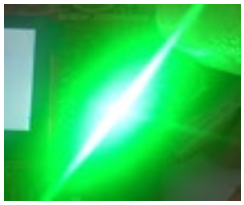
Instruction step 4.



Instruction step 5.

5. Click on 09.EducationalBP_MKII and select the ButtonsLEDs.ino project shown in the above figure.

6. Compile and flash the project. The LED will be blue when pressing the lower button and green when you press the upper button.



2 ADC / PWM

Once the previous steps are completed, you should be able to upload code to the board using the energia framework. The following subsections explain how you create projects, specific to the peripherals you are going to use in this lab, i.e., the ADC and the PWM frameworks.

2.1 Analog to Digital Converter (ADC): Joystick -> LCD Screen

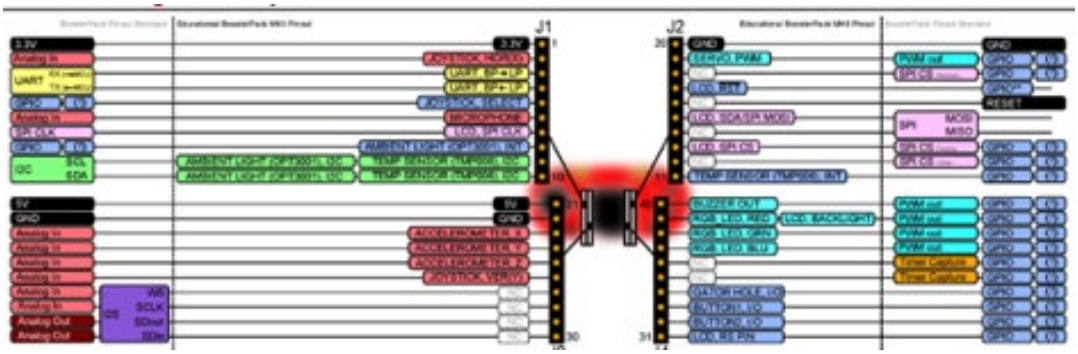
1. Create a project as explained in the previous section.
2. Select the LCD_Joystick project. This project shows *how to read analog inputs and how to write on the LCD screen*.
3. Compile and flash the project. The LCD screen will display the position of the joystick (x,y) with different colors as a function to (x,y).



2.2. Pulse-Width Modulation (PWM): PWM-> LED Brightness

Pulse-Width Modulation (PWM) is a modulation technique to allow control of the power supply of electrical devices. This can be for example the brightness of a LED or the speed of an engine. The voltage level is controlled by switching the supply voltage on and off. If the switching happens fast enough the sink experiences a voltage equivalent to the mean voltage value. However, switching needs to happen at a frequency much faster than what would be noticed by the sink.

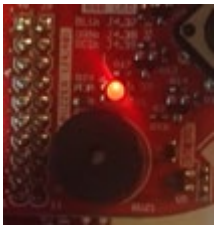
In this example, we will control the brightness of RGB_LED_RED which is connected to PG0 (this signal is controlled by Module 0 PWM Generator 2).



1. In Project Explorer, copy the project *Controlled* and paste it with the name *pwm_led*.
2. Copy the content of *pwm_led.c* and paste it in the file *main.c*
3. Build the project and load it.
4. By changing the PWMPulseWidthSet (line 59), the brightness of the red LED will be

changed.

PWMPulseWidthSet – 1000



PWMPulseWidthSet – 100

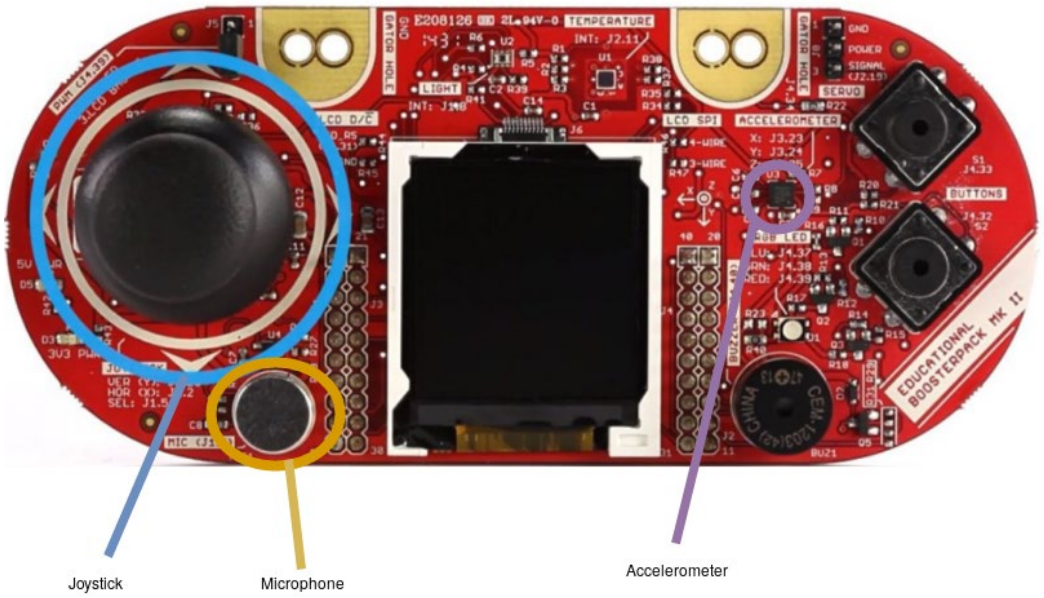
PWMPulseWidthSet – 10

3 Assignments (1.5 bonus points if completed together with the report before the deadline, which is specified in Canvas for each assignment)

1.1 Write a program which lets the user decide on the brightness of a LED using the PWM library for the embedded system. The expressed values should be on a percentage basis (between 0% and 100%), and should represent the LED intensity. The program should be able to set any percentage of light. The program must handle 3 different cases. Two special cases of 0% and 100% should turn off and on the LED using UART (functions for LEDs used in lab 1), respectively. Other values more than 0% and less than 100% should be expressed using PWM library.

1.2 Add another function, which uses the potentiometer values as the basis of input instead of the UART.

The peripherals are encircled in the Figures below, refer to pages 11, 12 and 13 for more info regarding these peripherals. Examples for all peripherals are provided by Energia examples.



Hint 1: To utilize the UART, you have two options. Either use the predefined serial functions from energia directly, or use the predefined UART functions in code composer studio (**Energia IDE is not allowed to be used in any labs**). The code composer studio UART functions require a bit more setup, but will be used in Lab 3, and is therefore recommended to use.

4 Report

Explain the mechanism of changing the LED intensity through discussing what happens in the source code. Explain the key functionalities such as duty cycle and others.

5 Optional Assignments (4 bonus points if you can implement 4 of the specified game and 2 bonus point for using LCD if completed before the deadline, which is specified in Canvas for each assignment)

5.1 Write a simple game on the platform. Do not feel limited by the small display area. Scroll the screen if you need more space. Implement 4 games to get 4 bonus points.

Examples:

(a) Snake

(b) Pong

(c) Space invaders

(d) Breakout

(e) Pacman

(f) Maze

You are welcome to make any other game that is not included in the above list. However, the game should be of equal complexity as compared to the games in the above list. Therefore, you must discuss the new game with the lab assistant and get approval before implementing it. **Note that you can get bonus points for a maximum of 4 games.**

5.2 Create a function which displays the values of the microphone, accelerometer, and the potentiometer (joystick) on the LCD screen using the Tivaware LCD boosterpack. Since the values are jittery, you must take average readings, e.g., moving average filter. **You get 2 bonus points if you complete this assignment.**