CMPE-110
Intro to Computer Engineering
Fall
Laboratory Seven
Introduction to the TI Robot Kit

This laboratory exercise covers the process of using the Energia Integrated Development Environment (IDE) to create, build, load, and run a program on the Microcontroller. The Microcontroller is the printed circuit board (PCB) that sits on the robot chassis. Once familiar with the Energia IDE, this exercise will cover how to program your robot to do simple movements.

Energia Configuration

Energia software is already installed on the lab computers (PCs). Before starting, write down your robot number on the worksheet. The robot number is R-XX, and can be found on a sticker on the top/back of the robot.

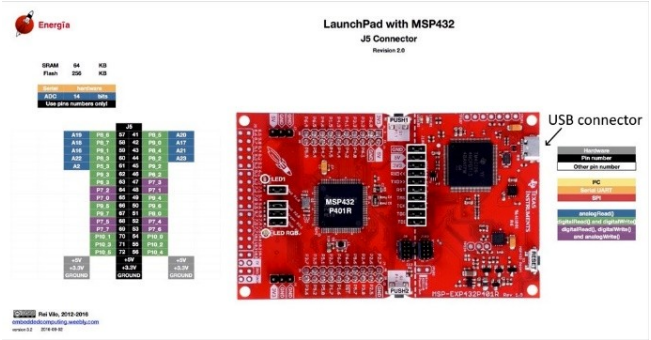
Connection

The Energia IDE communicates with the Microcontroller via a USB port on the PC. The USB cable is already connected to the PC and will always remain connected to the PC. Find this cable and then check the corresponding box on the worksheet checklist.

The microcontroller (rectangular board on top) can be powered from the battery pack on the robot or via the USB port. The robot itself cannot be powered over USB and must be supplied with batteries. Batteries can be inserted underneath the cover on the bottom of the robot. Pressing the power button on the bottom left corner of the robot will toggle the battery power supply.

The program that is already on the microcontroller will run as soon as the robot is powered. If you do not know what code loaded on the board, you want to be sure you are holding the robot (securely and off any surface) when you power on the robot.

Locate the USB programming port on the microcontroller. The cable is meant to be plugged in one way. Care must be taken when plugging in the cable to make sure it is in the proper orientation. If you are not sure, ask the instructor/TA for assistance. Plug the USB cable into the programming port.



Energia IDE Settings and Communication Verification

Start the Energia program. We should find in in the Window's Program list, or search for it. Follow these steps to configure the Energia IDE settings:

1. The IDE needs to be configured so that is knows it is communicating with the MSP432 board that is on the robot. Select menu item: Tools→Board→MSP432. Check the corresponding blank on the worksheet checklist.
2. The IDE needs to know how the board is connected to the PC. While it connects using a USB cable, the PC will see the robot on a Serial Port (i.e. COM***x***, where ***x*** is the Serial Port number, for example COM2). Your PC may have multiple serial ports, in which case you will need to use the highest COM port number; never select COM1. The COM ports are found within menu item Tools→Serial Port. If the Serial Port is grayed out, or if you only see COM1, check to make sure the microcontroller is connected to the PC via the USB cable. Once you can select a COM port higher than COM1 under Tools→Serial Port, check the corresponding blank on the worksheet checklist.

NOTE: You will want to double-check these selections each time you run the IDE.

Creating a Robotics Software Project

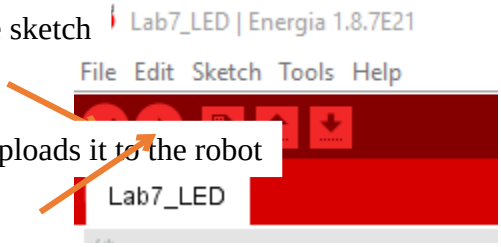
The Energia IDE uses project files to make it easy to switch between different projects. Programs in Energia are called a **sketch**. Use the following procedure to create a sketch called Lab7_LED from the already provided Blink example:

1. Select menu item File→Examples→Basics→Blink. A new window will appear.
3. Select menu item File→Save As…. Navigate to a folder on your thumb drive in which you want to create the new sketch and select that folder.
4. Enter Lab7_LED for the File name and then click Save.
5. At this point the window title and the name on the file tab will say Lab7_LED, as shown below.

Building, Loading, and Running a Program/Sketch

The Lab7_LED sketch you just created will blink the built-in LED on and off every second. Here are the common buttons used in the IDE:

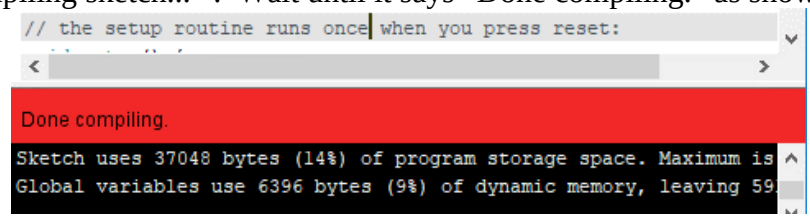
Verify - this will just compile the sketch



Upload - this compiles the sketch and then uploads it to the robot

Follow these steps to compile, load, and run the Lab7_LED sketch:

1. Click the **Verify** button to compile the sketch. You will notice the status bar below the code will say: "Compiling sketch... ". Wait until it says "Done compiling." as shown below:



   NOTE: if there were any errors, they would appear in the black area.

2. Since you do not know what code is currently in the robot, pick up the robot and hold it secure in your hand. Turn on the robot by press the power button. The blue LED will turn on indicating the robot is powered.
3. Click the **Upload** button to load the sketch onto the robot. Wait until the status bar says "Done uploading".

   NOTE: You will get errors if the USB cable is not connected to the robot and PC, or the selected Serial Port is not correct, or the selected Board type is not correct.

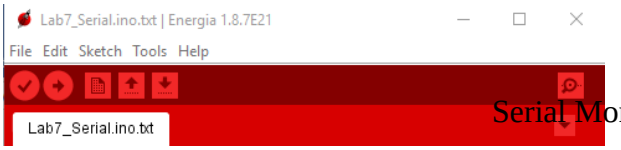
4. Confirm the LED at pin 13 blinks on for one second and off for one second, over and over. Please see the lab instructor/TA if you have a problem.

5. In the sketch, modify the second delay, in the loop() function, from 1000 (1 sec) to 3000 (3 sec).
6. Click the Verify button, then the Upload button to load the sketch onto the robot.
7. Confirm the LED at pin 13 blinks on for one second and off for three seconds, over and over.
8. Show your instructor/TA the LED to get your 'LED' program worksheet line initialed.

Serial Port Text Output

We will now create a new sketch to output text to the serial monitor, a built-in feature to the Energia IDE.

1. Select menu item File→Save As…
2. Navigate to a folder on your thumb drive in which you want to create the new sketch and select that folder.
3. Enter Lab7_Serial for the File name and then click Save.
4. Replace all the default code that appears in this new sketch with the code from the Lab7_Serial.ino.txt file on myCourses (Ctrl-A will select all the code, then hit Delete).
5. Save the sketch by using menu item File→Save
6. Click the Verify button, then the Upload button to load the sketch onto the robot.
7. We will now verify we are transferring serial data over the USB cable. Click the **Serial Monitor** button, location within the IDE is shown below, to open the serial monitor. As you should each time the monitor is run, make sure the baud rate is set to 19200, using pull-down in lower right corner of the serial monitor window and select *19200 baud*.



Serial Monitor - this will open another window

8. Confirm the serial monitor is displaying the word “HIGH” for two seconds and then the word “LOW” for two second intervals, over and over.
9. Show your instructor/TA the window to get your ‘Serial’ program worksheet line initialed.

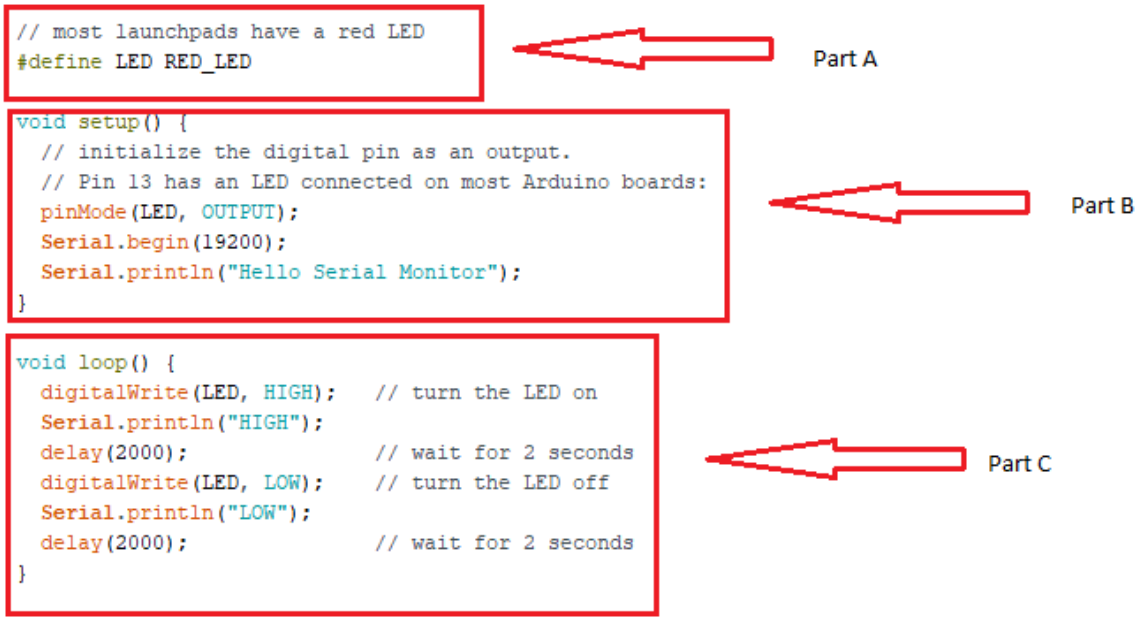
NOTE: If the Serial Monitor is open when then the USB cable is unplugged, the next upload may fail (see below for the type of error you may get). If that happens, just close the Serial Monitor window and try again.

java.io.IOException: jssc.SerialPortException: Port name - COM3; Method name - setEventsMask(); Exception type - Can't set mask

The Energia Sketch

So far Energia IDE has been used to perform several basic steps: create a software sketch; build and download the sketch to the Microcontroller; and run the sketch. Complete the following steps to further investigate Energia programming.

1. The Energia sketch we created has three components



Part A is where #defines and global variables are declared. Global variables are variables that can be used throughout the sketch.

Part B is the setup() function where we specify initialization code. This code executes only once. This happens when the code is first uploaded to the Energia, and whenever the board is reset. In the Blink example, it initializes the I/O pin 'LED' to OUTPUT. I/O pins can be initialized to INPUT or OUTPUT. 'LED' is defined as a global variable set in Part A.

Part C is the main code (i.e. loop() function). Each line of code is executed sequentially, one line at a time. This loop() function is called over and over again, until the board loses power or is reset. You may notice that **loop()** is similar to a **while(1)** loop in C/C++. In the Blink example, it makes use of the functions **`digitalWrite()`**, which sets an IO pin to HIGH/1/+5V or LOW/0/0V, and **delay()**, which pauses the sketch for the specified number or milliseconds. The **`digitalWrite(LED, HIGH)`** line is executed first, then the **delay(1000)** line of code is executed, followed by **digitalWrite(LED, LOW)**, and then **delay(1000)**. After the last line of code is executed, execution goes back to the first line of the loop code (i.e. **digitalWrite(LED, HIGH)**) and repeats the entire loop

2. On your worksheet, fill in the value of the variable 'led' in the commands digitalWrite(led, HIGH) and digitalWrite(led, LOW) in the Blink example sketch?
3. On your worksheet fill the descriptions for each argument for the **`pinMode()`** function. To find the information:

a. Navigate to the Arduino (NOTE: Energia is compatible with Arduino IDE) website: https://www.arduino.cc/
b. Click on Reference under the Resources menu item, to see all the commands.
c. Click on the pinMode() Digital I/O function.

Making my robot move!

NOTE: While you are writing code (i.e. the robot is sitting on the workbench), please turn the battery power off by pressing the power button so the robot does not drain the batteries.

We will now create a sketch to make our robot drive on the floor.

1. Select menu item File→New.
2. Replace all the default code that appears in this new sketch with the code from the Lab7_Robot.ino.txt file on myCourses (Ctrl-A will select all the code, then hit Delete).
3. Select menu item File→Save As… menu item. Navigate to a folder on your thumb drive in which you want to create the new sketch and select that folder.
4. Enter Lab7_Robot for the File name and then click Save.
5. Click the Verify button. If there are any compile problems, fix them before continuing.
6. Open the Serial Monitor and make sure the Baud rate it is set to 19200.
7. Hold the robot securely and with obstructing the wheels from moving.
8. Press the power button to turn battery power on
9. Press the 'Start' button **(button on the side of the microcontroller)** and watch the serial port and wheels. This will give a rough idea that things are working.
10. Carefully disconnect the USB cable from the Microcontroller. Please be careful plugging and unplugging the USB programming cable, only push it straight in and straight out. If you push on an angle, you will eventually break the USB port.
11. Place the robot on the ground and then hit the 'Start' button on the Microcontroller.
12. Confirm the robot: drives forward for 2 seconds; stops for 1.5 seconds; turns left for 1/2 second; stops for 1 second; turns right for 1 second; stops for 1 second; turns left for 1/2 second; stops for 1 second; drives forward for 1 second; stops for 1 second; drives backwards for 1 second; and then waits for the 'Start' button again.
13. Pick up the robot and turn the battery power off
14. Place the robot on the workbench and connect the USB cable. Please be careful plugging and unplugging the USB programming cable, only push it straight in and straight out. If you push on an angle, you will eventually break the USB port.
15. Study the commands that are in the loop() function. They are described in the file comment block at the top of the code. The commands available to you are:

```
fwd(speed)
rev(speed)
turnRight(speed)
turnLeft(speed)
stopMotors()
wait(duration)
```

16. By only modifying the commands in the loop() function, change the sketch so the robot does the following:
    a) Goes straight at a speed of 30 for 1 second
    b) Turns left at a speed of 40 for 3 seconds
    c) Does nothing for 1 second
    d) Goes backwards at a speed of 60 for 1/2 second
    e) Does nothing for 750msec
    f) Turns right at a speed of 50 for 2 seconds
    g) Goes straight at a speed of 50 for 1 second

17. Test your code to make sure it is working (i.e. Verify, Compile, disconnect USB, switch to the BAT position, place robot on floor, hit reset button on robot).

18. Demonstrate your working robot code to the instructor/TA, to get your 'Robot moves' program worksheet line initialed.

19. Pick up the robot and turn the power off.

20. Add an appropriate header at the top of the code, do NOT add any other comments. Save this code for the next lab.

Submission Checklist

One per team —*make sure team member names are written on each item*:

___ A completed laboratory worksheet.
___ A printed copy of portion of the 'Lab7_Robot' Energia sketch that you changed (e.g. the file comment block and loop() function; no need to print the code that was provided to you); also, do not forget to comment your code.
___ Make sure the robot's Power switch is in the OFF (left) position
___ Robot was put away.