Basics: Introduction to Programming

The LaunchPad is a programmable device. Let’s observe its versatility by learning how to program it.

Setup and Preparation

Item List
- TI MSP430G2 LaunchPad

Connecting to the LaunchPad
1. Connect the LaunchPad to the computer by USB.
2. Open the program Energia (Figure 1).
3. Go to Tools → Board
   a. Select “LaunchPad w/ msp430G2553”
4. Go to Tools → Serial Port
   a. Select COM port with the highest value.

Test the connection to the LaunchPad
1. Go to File → New
   a. This will open up a blank program.
2. Verify the program. (Click the Verify button).
3. Upload the program to the LaunchPad. (Click the Upload button).
   a. Check the messages at the bottom to ensure there were no errors.
   b. If it says “Done, [number] bytes total,” then it was successful. You are ready to program the LaunchPad.

Figure 1: Energia Program
Program 1: Creating Output

Create the following program (Figure 2) and upload it to the LaunchPad. Be sure to Verify your code before uploading it to the LaunchPad.

```c
void setup() {
  pinMode(RED_LED, OUTPUT);
}

void loop() {
  digitalWrite(RED_LED, HIGH);
  delay(1000);
  digitalWrite(RED_LED, LOW);
  delay(1000);
}
```

Figure 2: Blink Program

Things to consider

- You can change the speed of the blinking LED, or make a new pattern.
- There is also a GREEN_LED that you can use.
- Information about the functions in this program can be found at the following link:
  - [http://energia.nu/Reference_Index.html](http://energia.nu/Reference_Index.html)
Program 2: Detecting Input

Create the following program (Figure 3) and upload it to the LaunchPad.

```c
void setup() {
    Serial.begin(9600);
    pinMode(PUSH2, INPUT_PULLUP);
}

void loop() {
    boolean buttonState = digitalRead(PUSH2);
    Serial.println(buttonState);
    delay(500);
}
```

**Figure 3: Button Program**

**Things to consider**

- You can see what is happening by clicking on the Serial Monitor button in Energia after the program is uploaded.
- There are 2 pushbuttons on the LaunchPad: RESET and PUSH2. Be sure to use PUSH2 (labeled on the LaunchPad as P1.3).
- Information about the functions in this program can be found at the following link:
  - [http://energia.nu/Reference_Index.html](http://energia.nu/Reference_Index.html)

**Troubleshooting**

- “`usbutil: unable to find a device matching xxxx:xxxx`”
  - The serial port is blocking further uploads. Simply unplug the LaunchPad from the computer and plug it back in.
- You get weird looking text on the Serial Monitor
  - Make sure your baud rate at the bottom right of the screen is set to 9600 baud
Program 3: Arithmetic Computations

Create the following program (Figure 4) and upload it to the LaunchPad.

```csharp
void setup() {
    Serial.begin(9600);

    byte result1 = 26 + 10;
    byte result2 = 26 - 10;
    byte result3 = 26 * 10;
    byte result4 = 26 / 10;
    byte result5 = 26 % 10;
    byte result6 = result1 + result2;

    Serial.println();
    Serial.println(result1);
    Serial.println(result2);
    Serial.println(result3);
    Serial.println(result4);
    Serial.println(result5);
    Serial.println(result6);
    Serial.println();
}

void loop() {
}
```

Figure 4: Arithmetic Computation Program

Things to consider

- You can observe the output from this program using the Serial Monitor.
- The RESET button on the LaunchPad will re-run the program (useful if the output in the Serial Monitor is garbled).
- You should get strange result for result3 and result4. This is because are limitations associated with each data type.
- Information about arithmetic operators and data types can be found at the following link:
Program 4: Boolean Computations

Create the following program (Figure 5) and upload it to the LaunchPad.

```java
void setup() {
    Serial.begin(9600);

    boolean result1 = 5 < 3;
    boolean result2 = 4 == 4;
    boolean result3 = 8 >= 2;
    boolean result4 = !result3;
    boolean result5 = result1 && result2;
    boolean result6 = result1 || result2;

    Serial.println();
    Serial.println(result1);
    Serial.println(result2);
    Serial.println(result3);
    Serial.println(result4);
    Serial.println(result5);
    Serial.println(result6);
    Serial.println();
}

void loop() {
}
```

**Figure 5: Boolean Computation Program**

**Things to consider**

- You can observe the output from this program using the **Serial Monitor**.
- The RESET button on the LaunchPad will re-run the program (useful if the output in the **Serial Monitor** is garbled).
- This program demonstrates **Boolean logic**, using **comparison operators** and **Boolean Operators**.
- Information about **comparison operators** and **Boolean operators** can be found here:
Program 5: Incrementing Numbers

Create the following program (Figure 6) and upload it to the LaunchPad.

```c
void setup() {
    Serial.begin(9600);
}

byte number = 0;

void loop() {
    number = number + 1;
    Serial.println(number);
    delay(100);
}
```

Figure 6: Increment Program

Things to consider

- You can observe output from this program using the Serial Monitor.
- Variable scope determines where you can use a variable in your program.
- There is another way to increment numbers using compound operators.
- Information about variable scope and compound operators can be found at the following link:
  - http://energia.nu/reference/
Review

After this lab, you should have a good understanding of the following topics. If you’re not sure about some of them, go back through the lab and try to find a good place to explore the topic.

You should be able to do the following

- Program the LaunchPad to blink its LEDs in a desired pattern.
- Receive data from the LaunchPad using the Serial Monitor.
- Explain the purpose and usage of the following functions:
  - setup(), loop(), pinMode(), digitalWrite(), digitalRead(), delay(), and Serial.println()
- Perform mathematical computations and store the result in a variable.
- Perform Boolean computations and store the result in a variable.
- Explain the limitations of each data type when declaring variables.
- Explain how to expand or shrink the range for which a variable can be used (variable scope).
- Increment or decrement a number using the assignment operator or a compound operator.

Application

Try applying the skills you learned from this lab. You can come up with your own project idea or try one from the flowchart.