RoboSlam: Mac OS X MSP430 LaunchPad Setup

Please follow the next 9 steps, for a successful installation of the MSP430 LaunchPad development environment, and the examples of how to use the MSP430, for the purposes of building a robot.

1. INSTALL MSP LAUNCHPAD SOFTWARE

Install the following two components of software onto your Mac:

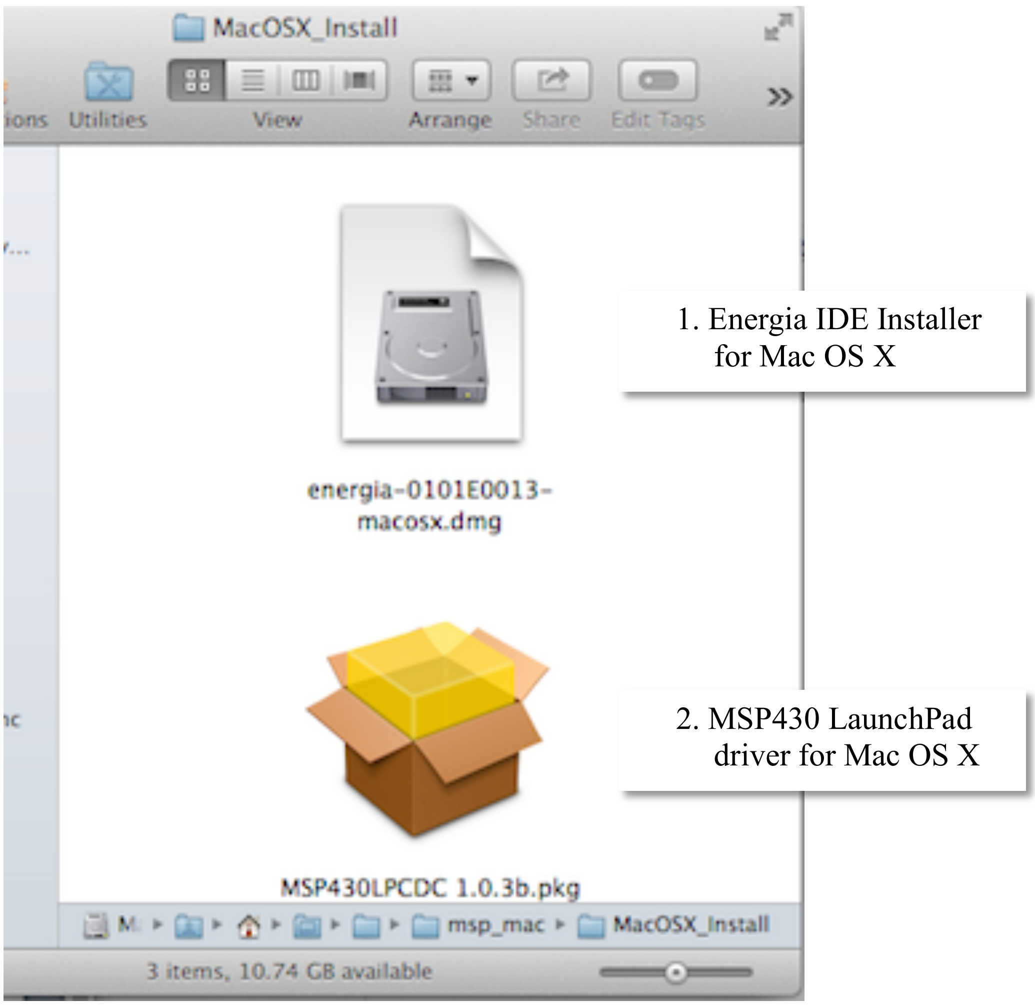

MacOSX_Install
Utilities
View
Arrange
Share
Edit Tags
energia-0101E0013-macosx.dmg
1. Energia IDE Installer for Mac OS X
MSP430LPCDC 1.0.3b.pkg
2. MSP430 LaunchPad driver for Mac OS X
msp_mac
MacOSX_Install
3 items, 10.74 GB available

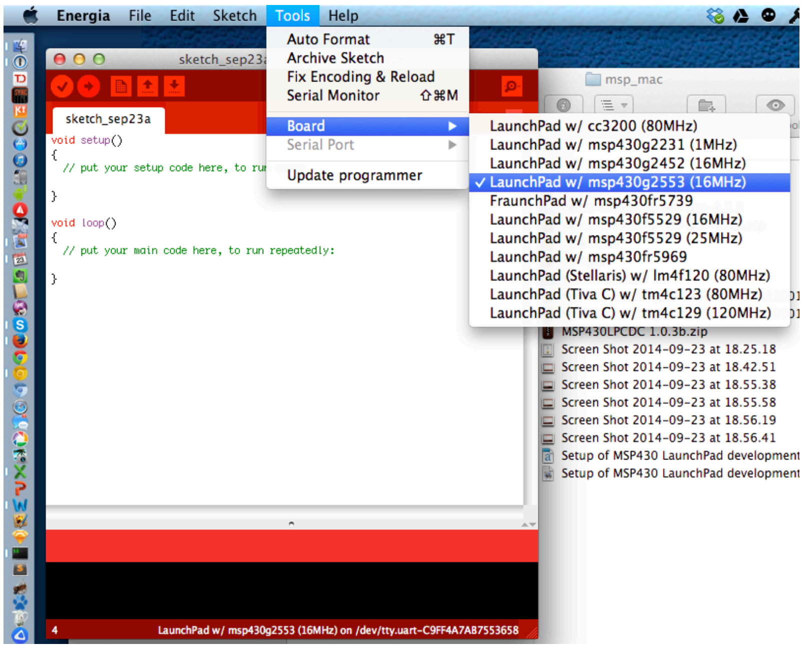
Figure 1 - MSP430 Software Installers for Mac OS X

After installing the Energia IDE and the MSP430 driver, you'll be asked to "Restart" your Mac (if you are not, then please do it anyway).

When your Mac has rebooted, you can launch the "Energia" application from your Applications folder.   (the Energia IDE is used for programming the TI MSP microcontroller chipsets, and it is almost identical to the "Arduino software" IDE, that is used to program the ATmega microcontrollers)

2. SELECT MSP BOARD AND PORT FOR SETUP

Figure 2 shows the setup of the MSP version. Each MSP version has specific features and capabilities, so that this setup is important to ensure that Engeria is configured to setup the MSP properly.


Energia
File
Edit
Sketch
Tools
Help
Auto Format ⌘T
Archive Sketch
Fix Encoding & Reload
Serial Monitor ⇧⌘M
Board
Serial Port
Update programmer
LaunchPad w/ cc3200 (80MHz)
LaunchPad w/ msp430g2231 (1MHz)
LaunchPad w/ msp430g2452 (16MHz)
✓ LaunchPad w/ msp430g2553 (16MHz)
FraunchPad w/ msp430fr5739
LaunchPad w/ msp430f5529 (16MHz)
LaunchPad w/ msp430f5529 (25MHz)
LaunchPad w/ msp430fr5969
LaunchPad (Stellaris) w/ lm4f120 (80MHz)
LaunchPad (Tiva C) w/ tm4c123 (80MHz)
LaunchPad (Tiva C) w/ tm4c129 (120MHz)
sketch_sep23a
msp_mac

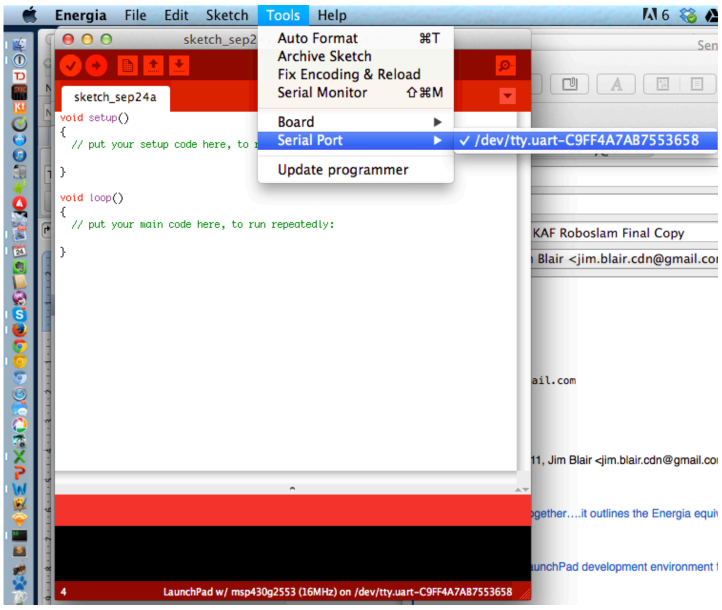
```
void setup()
{
  // put your setup code here, to ru
}

void loop()
{
  // put your main code here, to run repeatedly:

}
```

MSP430LPCDC 1.0.3b.zip
Screen Shot 2014-09-23 at 18.25.18
Screen Shot 2014-09-23 at 18.42.51
Screen Shot 2014-09-23 at 18.55.38
Screen Shot 2014-09-23 at 18.55.58
Screen Shot 2014-09-23 at 18.56.19
Screen Shot 2014-09-23 at 18.56.41
Setup of MSP430 LaunchPad development
Setup of MSP430 LaunchPad development

```
4    LaunchPad w/ msp430g2553 (16MHz) on /dev/tty.uart-C9FF4A7AB7553658
```

Figure 2 - Select MSP Version for Programming

Figure 3 illustrates the “Port” selection process, to enable communications between the Engergia application and the MSP430 LaunchPad board.



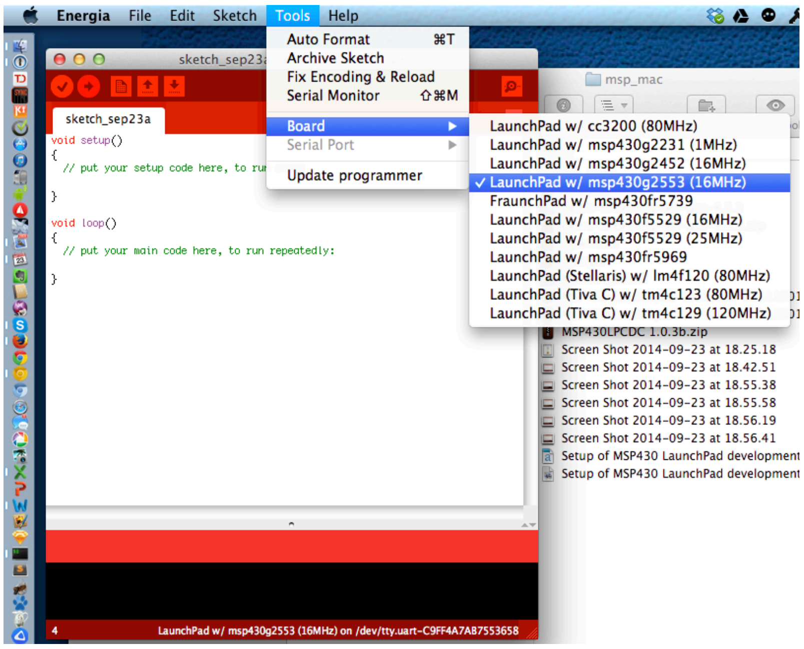
✓ /dev/tty.uart-C9FF4A7AB7553658

```
void setup()
{
  // put your setup code here, to r
}

void loop()
{
  // put your main code here, to run repeatedly:
}
```

```
4    LaunchPad w/ msp430g2553 (16MHz) on /dev/tty.uart-C9FF4A7AB7553658
```

Figure 3 - Select Port for LaunchPad Communications

Figure 4 shows the setup of the MSP version. Each MSP version has specific features and capabilities, so that this setup is important to ensure that Engeria is configured to setup the MSP properly.


Energia File Edit Sketch Tools Help
sketch_sep23a
Auto Format ⌘T
Archive Sketch
Fix Encoding & Reload
Serial Monitor ⇧⌘M
Board
Serial Port
Update programmer
msp_mac
MSP430LPCDC 1.0.3b.zip
Screen Shot 2014-09-23 at 18.25.18
Screen Shot 2014-09-23 at 18.42.51
Screen Shot 2014-09-23 at 18.55.38
Screen Shot 2014-09-23 at 18.55.58
Screen Shot 2014-09-23 at 18.56.19
Screen Shot 2014-09-23 at 18.56.41
Setup of MSP430 LaunchPad development
Setup of MSP430 LaunchPad development

LaunchPad w/ cc3200 (80MHz)
LaunchPad w/ msp430g2231 (1MHz)
LaunchPad w/ msp430g2452 (16MHz)
✓ LaunchPad w/ msp430g2553 (16MHz)
FraunchPad w/ msp430fr5739
LaunchPad w/ msp430f5529 (16MHz)
LaunchPad w/ msp430f5529 (25MHz)
LaunchPad w/ msp430fr5969
LaunchPad (Stellaris) w/ lm4f120 (80MHz)
LaunchPad (Tiva C) w/ tm4c123 (80MHz)
LaunchPad (Tiva C) w/ tm4c129 (120MHz)

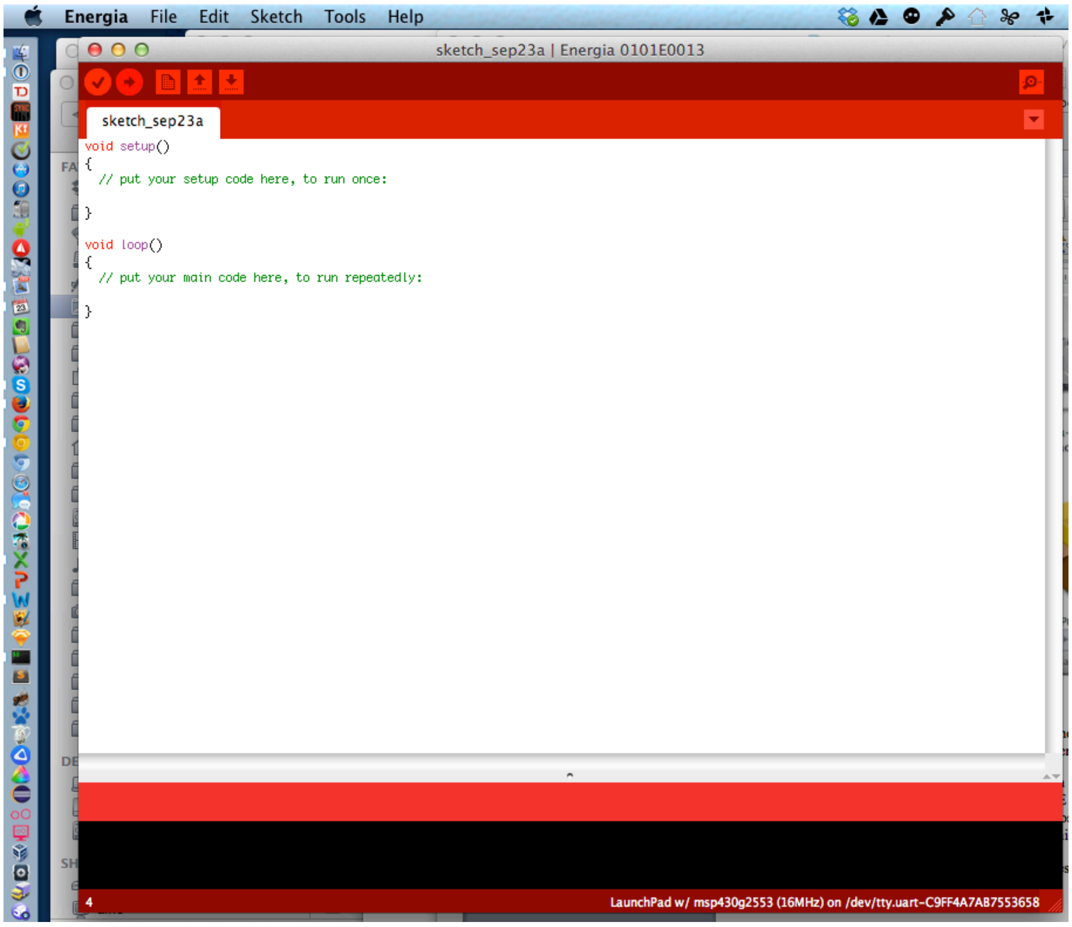
```
void setup()
{
  // put your setup code here, to ru
}

void loop()
{
  // put your main code here, to run repeatedly:
}
```

```
4    LaunchPad w/ msp430g2553 (16MHz) on /dev/tty.uart-C9FF4A7AB7553658
```

Figure 4 - Select Mac Port that the MSP LaunchPad is connected to.

3. EDITING CODE

When you launch Energia, you are presented with this window….


Energia File Edit Sketch Tools Help
sketch_sep23a | Energia 0101E0013
sketch_sep23a

```
void setup()
{
  // put your setup code here, to run once:

}

void loop()
{
  // put your main code here, to run repeatedly:

}
```

```
4                                   LaunchPad w/ msp430g2553 (16MHz) on /dev/tty.uart-C9FF4A7AB7553658
```

Figure 5 - Energia IDE Sketch Window

As is the case with the Arduino IDE, the Energia IDE uses the notion of "sketches", which are programs that are run on the MSP430 microcontroller.

When writing a microcontroller program, the developer writes code that is run during the MSP430 initialisation, and this code is placed in the "setup()" function, shown in Figure 5. The developer then writes the main microcontroller code by filling in the "loop()" function, shown in Figure 5. The loop() function runs forever, unless the code "break"s out of the loop(), or the user turns off the power to the microcontroller.

4. COMPILING CODE AND DOWNLOADING TO MSP LAUNCHPAD

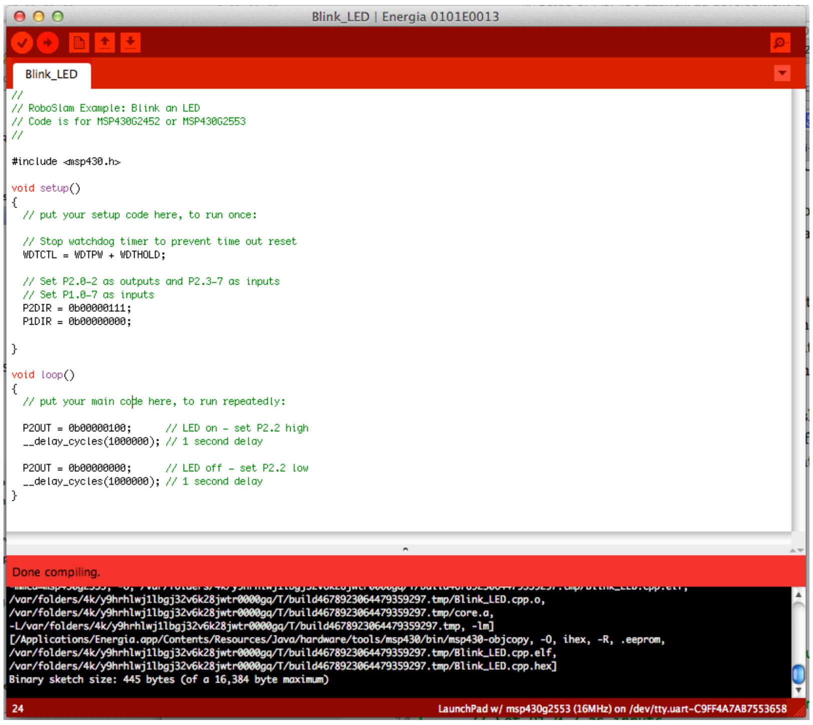
When writing code/software for the MSP430 Launchpad, you should frequently select the ✓ "Compile" icon in Energia's IDE window. Once you are finished writing the code, or you want to test new code you've added and compiled, you select the ➜ "Download" icon in Energia's IDE window. If you've written some new code, and you want to quickly test it on the MSP430, simply select the "Download" icon, because Energia is smart enough to know whether it needs to compile new code, before it tries to download it to the MSP430.

The following examples illustrate how to interact with the MSP430, as building blocks that can be used to build a full robot. Figure 6 thru Figure 9 illustrate those examples.

5. LED EXAMPLE

The RoboSlam examples are replicated, as "Energia sketches" below, in Figure 6 thru Figure 9:

Figure 6 - Blink LED Sketch/Example illustrates how to control the on/off state of the robot LED.



```
//
// RoboSlam Example: Blink an LED
// Code is for MSP430G2452 or MSP430G2553
//

#include <msp430.h>

void setup()
{
  // put your setup code here, to run once:

  // Stop watchdog timer to prevent time out reset
  WDTCTL = WDTPW + WDTHOLD;

  // Set P2.0-2 as outputs and P2.3-7 as inputs
  // Set P1.0-7 as inputs
  P2DIR = 0b00000111;
  P1DIR = 0b00000000;

}

void loop()
{
  // put your main code here, to run repeatedly:

  P2OUT = 0b00000100;       // LED on - set P2.2 high
  __delay_cycles(1000000); // 1 second delay

  P2OUT = 0b00000000;       // LED off - set P2.2 low
  __delay_cycles(1000000); // 1 second delay
}
```

Done compiling.

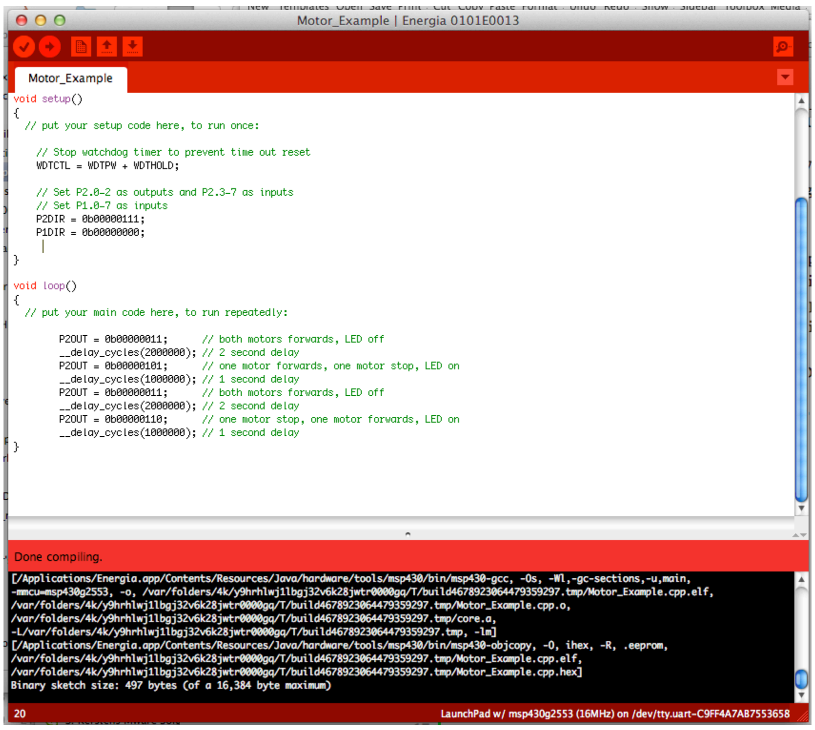
```
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/Blink_LED.cpp.o,
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/core.a,
-L/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp, -lm]
[/Applications/Energia.app/Contents/Resources/Java/hardware/tools/msp430/bin/msp430-objcopy, -O, ihex, -R, .eeprom,
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/Blink_LED.cpp.elf,
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/Blink_LED.cpp.hex]
Binary sketch size: 445 bytes (of a 16,384 byte maximum)
```

24   LaunchPad w/ msp430g2553 (16MHz) on /dev/tty.uart-C9FF4A7AB7553658

Figure 6 - Blink LED Sketch/Example

6. MOTOR CONTROL EXAMPLE

Figure 7 - Control Motors Sketch/Example illustrates how to control the operation of your robot wheel motors. The more effective you can control the robot wheel operation, the better your robot should perform.



```
void setup()
{
  // put your setup code here, to run once:

    // Stop watchdog timer to prevent time out reset
    WDTCTL = WDTPW + WDTHOLD;

    // Set P2.0-2 as outputs and P2.3-7 as inputs
    // Set P1.0-7 as inputs
    P2DIR = 0b00000111;
    P1DIR = 0b00000000;

}

void loop()
{
  // put your main code here, to run repeatedly:

        P2OUT = 0b00000011;      // both motors forwards, LED off
        __delay_cycles(2000000); // 2 second delay
        P2OUT = 0b00000101;      // one motor forwards, one motor stop, LED on
        __delay_cycles(1000000); // 1 second delay
        P2OUT = 0b00000011;      // both motors forwards, LED off
        __delay_cycles(2000000); // 2 second delay
        P2OUT = 0b00000110;      // one motor stop, one motor forwards, LED on
        __delay_cycles(1000000); // 1 second delay
}
```

Done compiling.

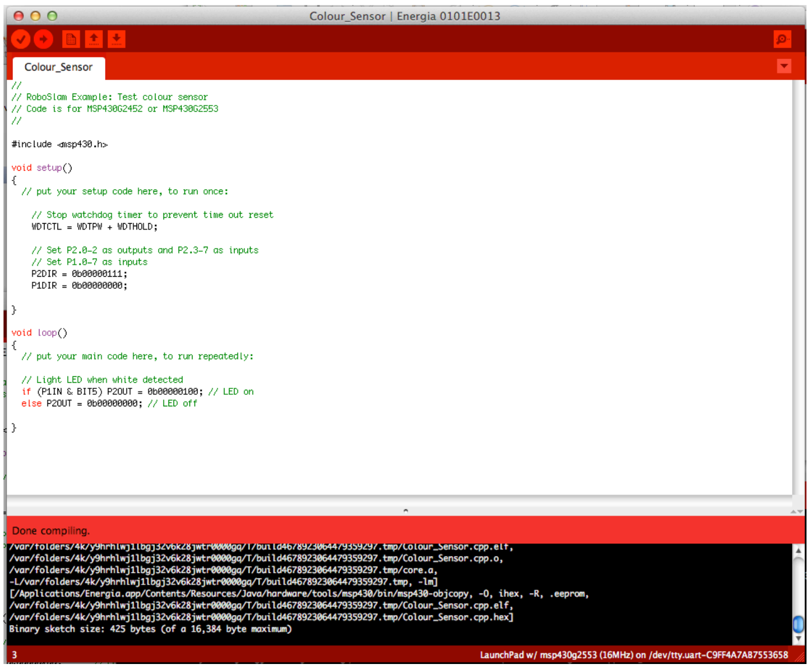
```
[/Applications/Energia.app/Contents/Resources/Java/hardware/tools/msp430/bin/msp430-gcc, -Os, -Wl,-gc-sections,-u,main,
-mmcu=msp430g2553, -o, /var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/Motor_Example.cpp.elf,
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/Motor_Example.cpp.o,
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/core.a,
-L/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp, -lm]
[/Applications/Energia.app/Contents/Resources/Java/hardware/tools/msp430/bin/msp430-objcopy, -O, ihex, -R, .eeprom,
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/Motor_Example.cpp.elf,
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/Motor_Example.cpp.hex]
Binary sketch size: 497 bytes (of a 16,384 byte maximum)
```

20   LaunchPad w/ msp430g2553 (16MHz) on /dev/tty.uart-C9FF4A7AB7553658

Figure 7 - Control Motors Sketch/Example

7. READING COLOUR SENSOR EXAMPLE

Figure 8 illustrates how to read the input from your robot's visual sensor circuit. Once you successful enable visual sensing, you should be able to turn the sensing input into robot actions, such as navigation, as per the next section, "NAVIGATE USING COLOUR SENSOR EXAMPLE."



```
//
// RoboSlam Example: Test colour sensor
// Code is for MSP430G2452 or MSP430G2553
//

#include <msp430.h>

void setup()
{
  // put your setup code here, to run once:

    // Stop watchdog timer to prevent time out reset
    WDTCTL = WDTPW + WDTHOLD;

    // Set P2.0-2 as outputs and P2.3-7 as inputs
    // Set P1.0-7 as inputs
    P2DIR = 0b00000111;
    P1DIR = 0b00000000;

}

void loop()
{
  // put your main code here, to run repeatedly:

  // Light LED when white detected
  if (P1IN & BIT5) P2OUT = 0b00000100; // LED on
  else P2OUT = 0b00000000; // LED off

}
```

Done compiling.

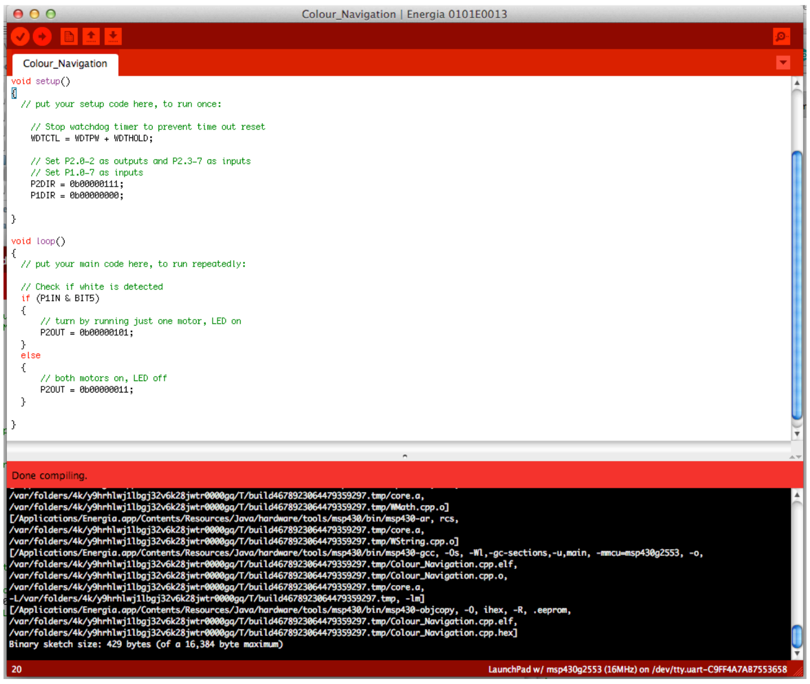
```
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/Colour_Sensor.cpp.elf,
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/Colour_Sensor.cpp.o,
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/core.a,
-L/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp, -lm]
[/Applications/Energia.app/Contents/Resources/Java/hardware/tools/msp430/bin/msp430-objcopy, -O, ihex, -R, .eeprom,
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/Colour_Sensor.cpp.elf,
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/Colour_Sensor.cpp.hex]
Binary sketch size: 425 bytes (of a 16,384 byte maximum)
```

3 LaunchPad w/ msp430g2553 (16MHz) on /dev/tty.uart-C9FF4A7AB7553658

Figure 8 - Read Colour Sensor Sketch/Example

8. NAVIGATE USING COLOUR SENSOR EXAMPLE

Figure 9 illustrates how to turn the visual sensing input into navigation control of your robot. The more effective your robot's visual sensing, the better your robot should perform.



```
void setup()
{
  // put your setup code here, to run once:

    // Stop watchdog timer to prevent time out reset
    WDTCTL = WDTPW + WDTHOLD;

    // Set P2.0-2 as outputs and P2.3-7 as inputs
    // Set P1.0-7 as inputs
    P2DIR = 0b00000111;
    P1DIR = 0b00000000;

}

void loop()
{
  // put your main code here, to run repeatedly:

  // Check if white is detected
  if (P1IN & BIT5)
  {
      // turn by running just one motor, LED on
      P2OUT = 0b00000101;
  }
  else
  {
      // both motors on, LED off
      P2OUT = 0b00000011;
  }

}
```

Done compiling.

```
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/core.a,
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/WMath.cpp.o]
[/Applications/Energia.app/Contents/Resources/Java/hardware/tools/msp430/bin/msp430-ar, rcs,
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/core.a,
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/WString.cpp.o]
[/Applications/Energia.app/Contents/Resources/Java/hardware/tools/msp430/bin/msp430-gcc, -Os, -Wl,-gc-sections,-u,main, -mmcu=msp430g2553, -o,
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/Colour_Navigation.cpp.elf,
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/Colour_Navigation.cpp.o,
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/core.a,
-L/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp, -lm]
[/Applications/Energia.app/Contents/Resources/Java/hardware/tools/msp430/bin/msp430-objcopy, -O, ihex, -R, .eeprom,
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/Colour_Navigation.cpp.elf,
/var/folders/4k/y9hrhlwj1lbgj32v6k28jwtr0000gq/T/build4678923064479359297.tmp/Colour_Navigation.cpp.hex]
Binary sketch size: 429 bytes (of a 16,384 byte maximum)
```

20 LaunchPad w/ msp430g2553 (16MHz) on /dev/tty.uart-C9FF4A7AB7553658

Figure 9 - Navigate Robot using Colour Sensing Sketch/Example

9. SUMMARY

To build a competitive robot for the "RoboSlam", developers should leverage the preceding examples, for flashing the LED, for controlling the robot motors, and for sensing and navigating through sight. The better you can integrate the robot senses, and turn that into robot control, the better your robot should perform. Best of luck!