Lab 1

1.1 Understanding the CCS default Code Structure

Programming for Blinking LED using C and Register:

MSP432 Launchpad comes with Onboard LEDs, which is interfaced with Port-1 pin 1.0.

Also TI Launchpad MSP432p401R is featured with RED LED at PIN P2.0, GREEN LED at PIN P2.1 and BLUE LED at PIN P2.2.

Step 1: Open CSS and create new project with empty c file (main.c) as discussed in Lab 0

Step 2: Write code C code for blinking LED in main.c

Code :

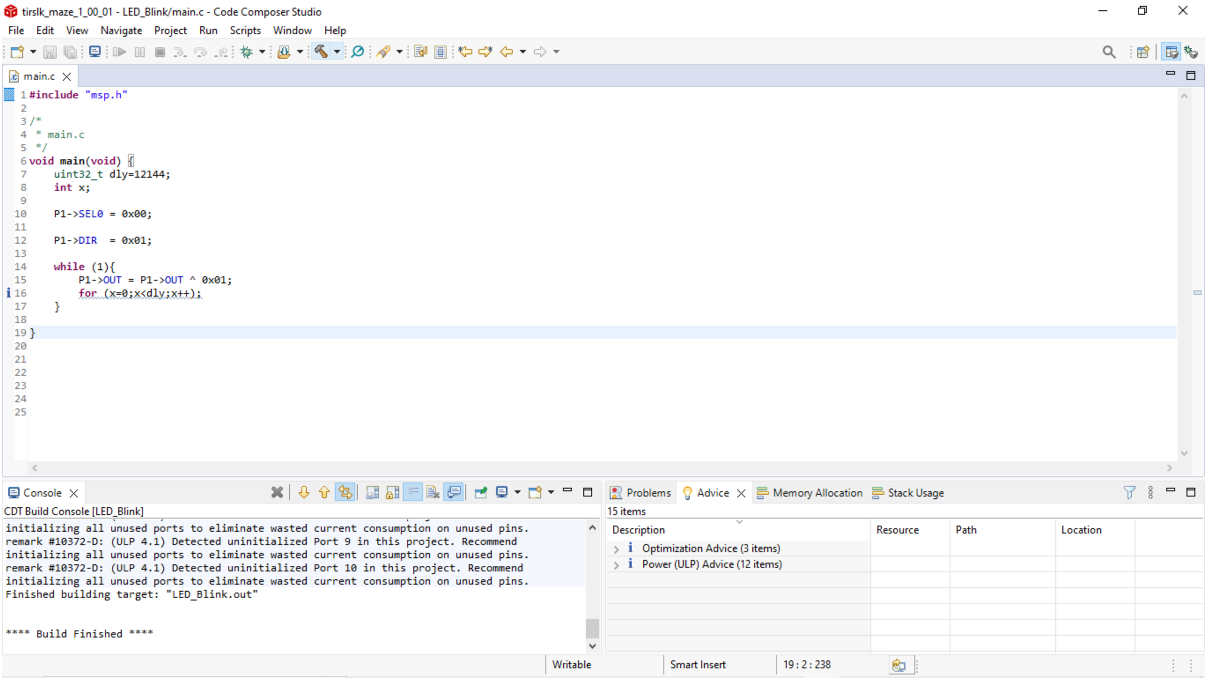
```
#include "msp.h"

/*
 * main.c
 */
void main(void) {
    uint32_t dly=12144;    // 4 byte unsigned integer //
    int x;

    P1->SEL0 = 0x00;      // Initialize P1.0 GPIO pin //

    P1->DIR  = 0x01;      // Set P1.0 (LED) to output direction //


    while (1){
        P1->OUT = P1->OUT ^ 0x01; // Set P1.0 output high //
        for (x=0;x<dly;x++);
    }
```

Step 3: Upload the code into MSP432.

To upload the code to MSP432 controller first we need to build the project and have to see is the any errors.

To Build the project Click> Project>> Build projects

If Projects build successful you will get below response in console window


tirslk_maze_1_00_01 - LED_Blink/main.c - Code Composer Studio
File Edit View Navigate Project Run Scripts Window Help
main.c
Console
CDT Build Console [LED_Blink]
Problems
Advice
Memory Allocation
Stack Usage
15 items
Writable
Smart Insert
19 : 2 : 238

```c
#include "msp.h"

/*
 * main.c
 */
void main(void) {
    uint32_t dly=12144;
    int x;

    P1->SEL0 = 0x00;

    P1->DIR  = 0x01;

    while (1){
        P1->OUT = P1->OUT ^ 0x01;
        for (x=0;x<dly;x++);
    }

}
```

```
initializing all unused ports to eliminate wasted current consumption on unused pins.
remark #10372-D: (ULP 4.1) Detected uninitialized Port 9 in this project. Recommend
initializing all unused ports to eliminate wasted current consumption on unused pins.
remark #10372-D: (ULP 4.1) Detected uninitialized Port 10 in this project. Recommend
initializing all unused ports to eliminate wasted current consumption on unused pins.
Finished building target: "LED_Blink.out"


**** Build Finished ****
```

Description	Resource	Path	Location
Optimization Advice (3 items)			
Power (ULP) Advice (12 items)			

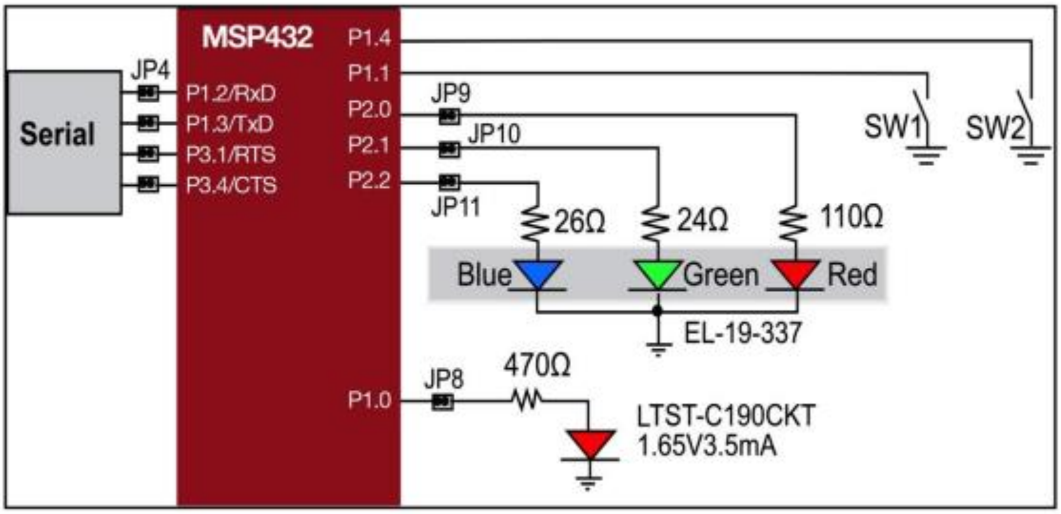
Now we need to upload the program. Click on Run -> Debug. After clicking on debug, you will get a window related to power saving, just click on proceed.

Step 4: Run the code.

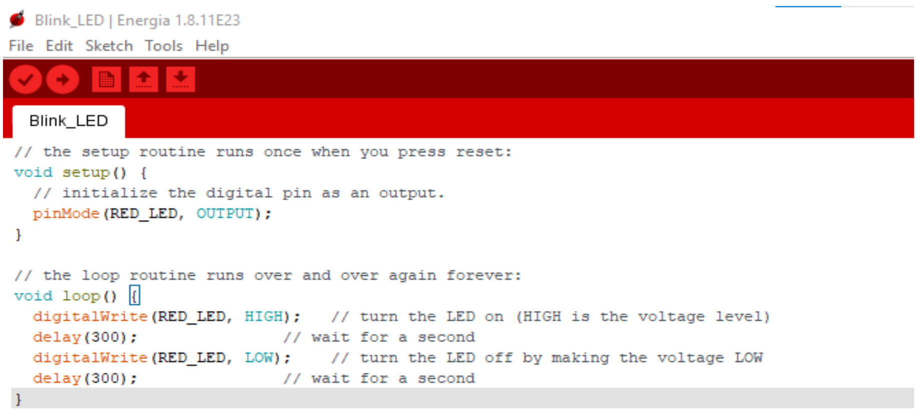
In debug option go to Run->Resume

1.2 How to program MSP432 to blink its LED using Energia.

MSP432 Launchpad has RED LED attached to pin P1.0 and RED, GREEN and BLUE LED’s on pin P2.0, P2.1 and P2.2 respectively on the board itself as shown below in the image.



Step 1: Open Energia IDE and Create New Project Blink_LED.

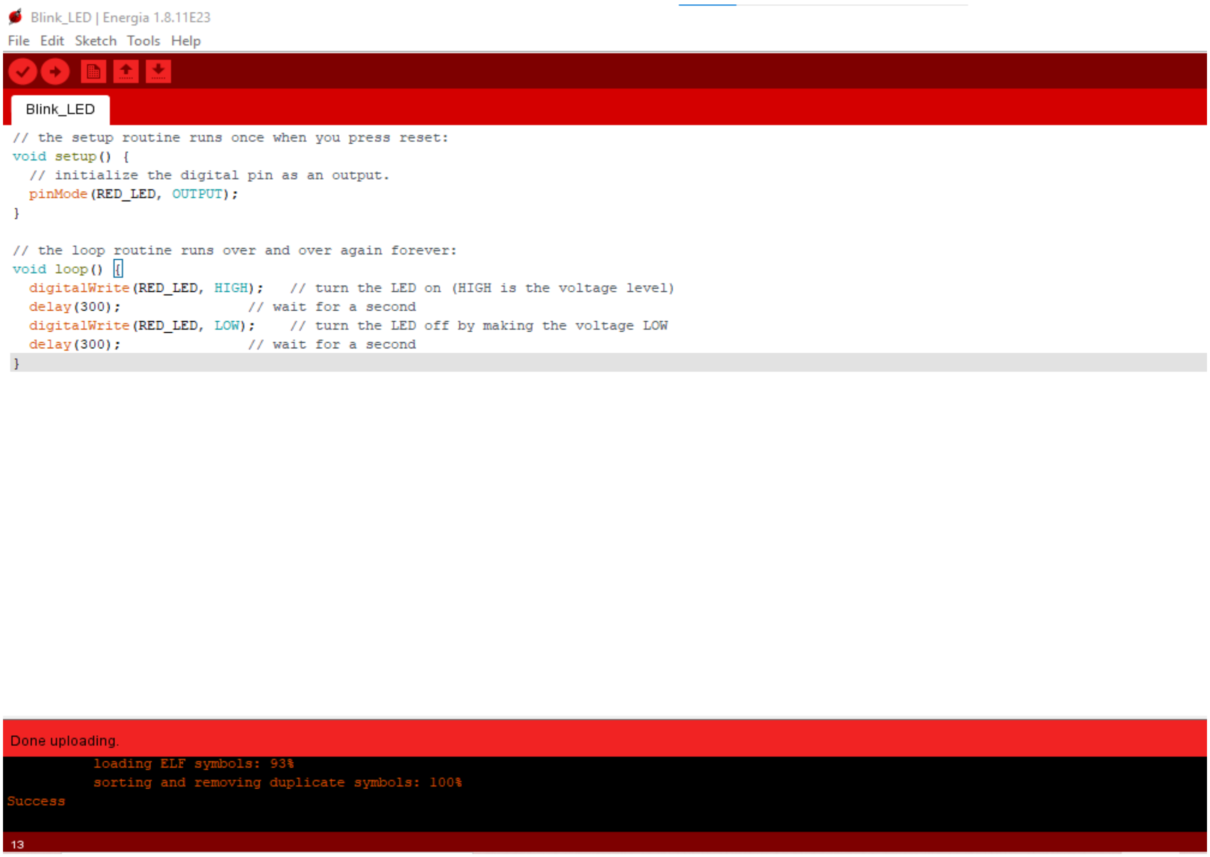


```
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(RED_LED, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(RED_LED, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(300);               // wait for a second
  digitalWrite(RED_LED, LOW);    // turn the LED off by making the voltage LOW
  delay(300);               // wait for a second
}
```

Step 2: Write Below Code.

```
/*
  Blink
  The basic Energia example.
  Turns on an LED on for one second, then off for one second,
repeatedly.

*/

// the setup routine runs once when you press reset:
void setup() {

  pinMode(RED_LED, OUTPUT); // initialize the digital pin as an output.
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(RED_LED, HIGH);   // turn the LED on (HIGH is the
voltage level)
  delay(1000);                 // wait for a second
  digitalWrite(RED_LED, LOW);    // turn the LED off by making the
voltage LOW
  delay(1000);                 // wait for a second
}
```

Step 2: Verify and Upload the code.



```
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(RED_LED, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(RED_LED, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(300);                  // wait for a second
  digitalWrite(RED_LED, LOW);    // turn the LED off by making the voltage LOW
  delay(300);                  // wait for a second
}
```

```
Done uploading.
        loading ELF symbols: 93%
        sorting and removing duplicate symbols: 100%
Success
```

Code Explanation

In the program, the first thing you do is to initialize pin 2 [RED_LED] as an output pin with the line:

```
pinMode(RED_LED, OUTPUT);
```

In the main loop, you turn the LED on with the line:

```
digitalWrite(RED_LED, HIGH);
```

This supplies 3 volts to pin 2. That creates a voltage difference across the pins of the LED, and lights it up. Then you turn it off with the line:

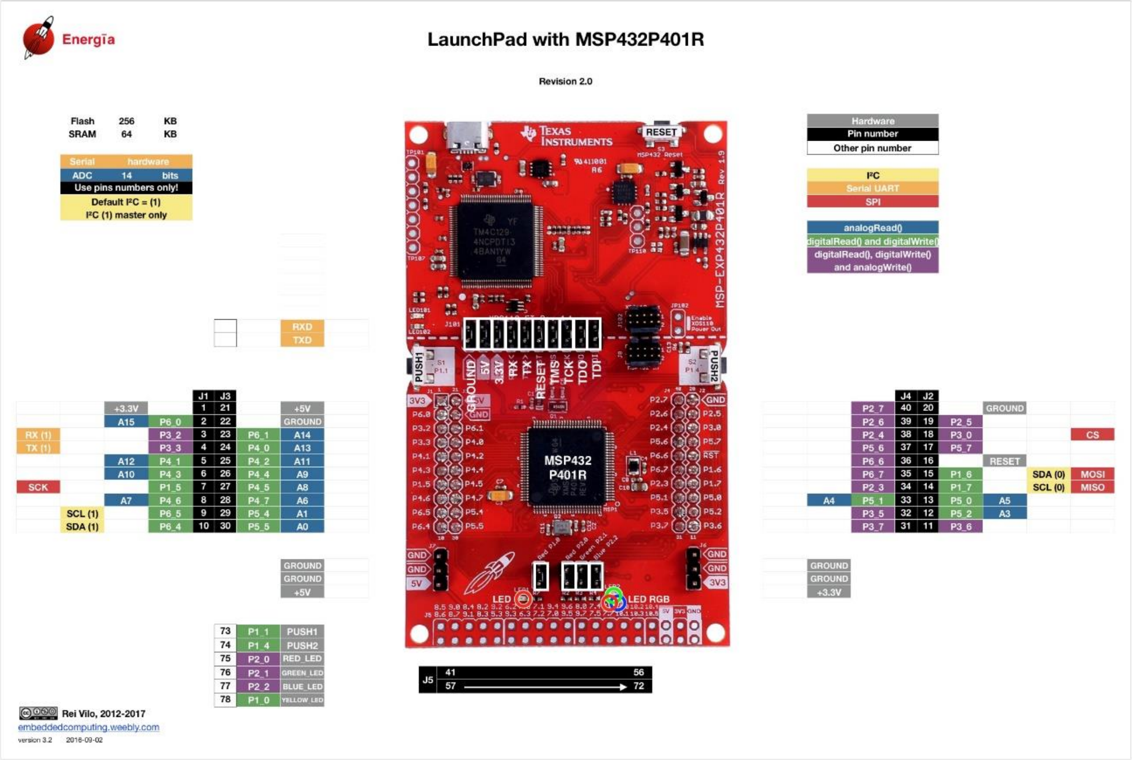
```
digitalWrite(RED_LINE, LOW);
```

That takes pin 2 back to 0 volts, and turns the LED off. In between the on and the off, you want enough time for a person to see the change, so the `delay`() commands tell the MSP430 to do nothing for 1000 milliseconds, or one second. When you use the `delay()` command, nothing else happens for that amount of time.

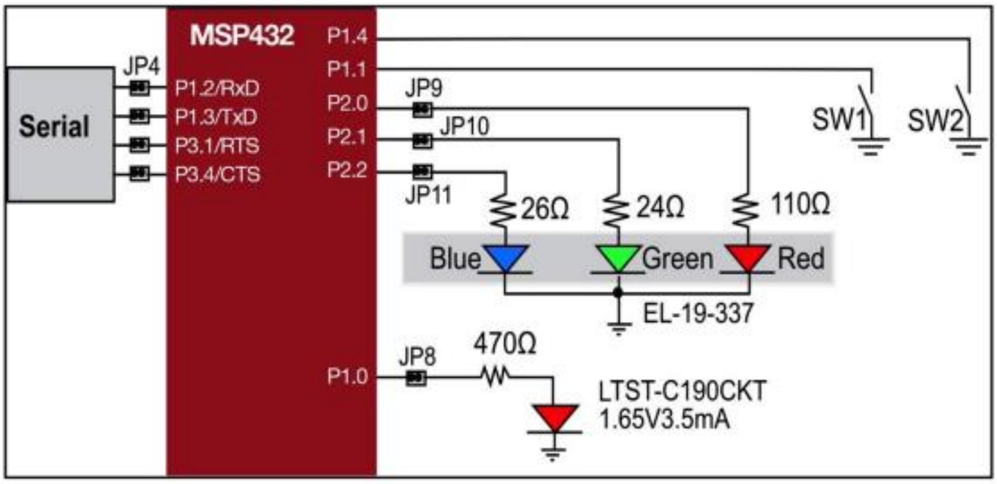
1.3 RGB LED

With the help of information in **1.1** control the RGB LED's on MSP432 board.

Pin Configuration for MSP432 board.



MSP432 Launchpad has RED LED attached to pin P1.0 and RED, GREEN and BLUE LED's on pin P2.0, P2.1 and P2.2 respectively on the board itself as shown below in the image.



Pin Configuration for RGB LED's on MSP432 board.

Step 1: Use code in 1.1 and assign pins P2.0, P2.1, P2.2 as an output pins

Step 2: Try to control each LED (RGB) individually.

```
        P2->SEL0 &= ~0x07;
        P2->SEL1 &= ~0x07;  // 1) configure P2.2-P2.0 as GPIO (All ‘RGB’ pins are
selected)


        P2->DIR |= 0x07;     // 2) make P2.2-P2.0 out (All are output)
```

Step 3: Make some blinking pattern using RGB LED’s on MSP432.

Hint – Use functions or Define each LED and pass it to blink function.