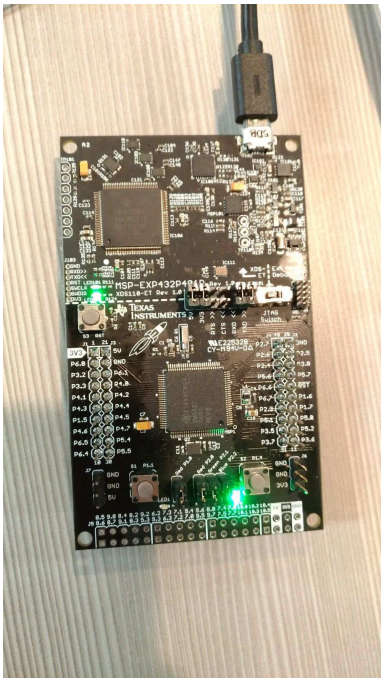
Colleen Lau
Brian
4/9/2018
A2 Blinking LEDS

1.1



https://youtu.be/h9ZabkehBmg

Objective:
Demonstrate one pulse blinking leds at varied frequencies

1.2

```
//CPE 329 Colleen Lau, Brian Buchanan
#include "msp.h"
void delay_ms(int ms, int freq);
void delay_us(int us, int freq1);
void set_DCO(int frequency);

int main(void) {

  P2->SEL1 &= ~BIT1;      // set P2.1 as GPIO
  P2->SEL0 &= ~BIT1;      // set P2.5 as GPIO
  P2->DIR  |= BIT1;       // set P2.1 as output

  while (1) {
    P2->OUT |= BIT1;    // P2.1 on
    delay_ms(500, 1000);
    P2->OUT &= ~BIT1;   // P2.1 off
    delay_ms(500);

    P2->OUT |= BIT1;    // P2.1 on
```

```
      delay_us(500, 1000);
      P2->OUT &= ~BIT1;    // P2.1 off
      delay_us(500);
    }
}

// Delay milliseconds function
void delay_ms(int ms, int freq) {
    int i, j;
    i = freq*0.0001  ;              // convert to ms
    for (j = 0; j < ms; j++)
        for (i = 750; i > 0; i--);   // delay 1 ms (approx)
}

// Delay microseconds function
void delay_us(int us, int freq1) {
    int i, j;
    i = freq*0.0000001;             // convert to us
    for (j = 0; j < us; j++)
        for (i = 300; i > 0; i--);  // delay 1us (approx)
}

// Set  MSP432 Frequency
void set_DCO(int frequency)
{
  if (frequency == 1500000)
  {
    // Changing DCO of default 3MHz to 1.5MHz
    CS -> KEY = CS_KEY_VAL;
    CS -> CTL0 = 0;
    CS -> CTL0 = CS_CTL0_DCORSEL_0;

    // select clock sources
    CS -> CTL1 = CS_CTL1_SELA_2 | CS_CTL1_SELS_3 | CS_CTL1_SELM_3;
    CS -> KEY = 0;
  }
  else if (frequency == 6000000)
  {
    // Changing DCO of default 3MHz to 6MHz
    CS -> KEY = CS_KEY_VAL;
    CS -> CTL0 = 0;
    CS -> CTL0 = CS_CTL0_DCORSEL_2;
```

```
  // select clock sources
  CS -> CTL1 = CS_CTL1_SELA_2 | CS_CTL1_SELS_3 | CS_CTL1_SELM_3;
  CS -> KEY = 0;
 }
 else if (frequency == 12000000)
 {
  // Changing DCO of default 3MHz to 12MHz
  CS -> KEY = CS_KEY_VAL;
  CS -> CTL0 = 0;
  CS -> CTL0 = CS_CTL0_DCORSEL_3;

  // select clock sources
  CS -> CTL1 = CS_CTL1_SELA_2 | CS_CTL1_SELS_3 | CS_CTL1_SELM_3;
  CS -> KEY = 0;
 }
 else if (frequency == 24000000)
 {
  // Changing DCO of default 3MHz to 1.5MHz
  CS -> KEY = CS_KEY_VAL;
  CS -> CTL0 = 0;
  CS -> CTL0 = CS_CTL0_DCORSEL_4;

  // select clock sources
  CS -> CTL1 = CS_CTL1_SELA_2 | CS_CTL1_SELS_3 | CS_CTL1_SELM_3;
  CS -> KEY = 0;
 }
 else if (frequency == 48000000)
 {
  // Transition to VCOre Level 1: AM0_LD0 --> AM1_LD0
  while ((PCM -> CTL1 & PCM_CTL1_PMR_BUSY));
  PCM -> CTL0 = PCM_CTL0_KEY_VAL | PCM_CTL0_AMR_1;
  while ((PCM -> CTL1 & PCM_CTL1_PMR_BUSY));

  // Configure Flash wait-state to 1 for banks 0 & 1
  FLCTL -> BANK0_RDCTL = (FLCTL -> BANK0_RDCTL &
                  ~(FLCTL_BANK0_RDCTL_WAIT_MASK)) |
FLCTL_BANK0_RDCTL_WAIT_1;
  FLCTL -> BANK0_RDCTL = (FLCTL -> BANK0_RDCTL &
                  ~(FLCTL_BANK1_RDCTL_WAIT_MASK)) |
FLCTL_BANK1_RDCTL_WAIT_1;

  // Configure DCO to 48MHz
  CS -> KEY = CS_KEY_VAL;
```

```
    CS -> CTL0 = 0;
    CS -> CTL0 = CS_CTL0_DCORSEL_5;

    // Select MCLK = DCO
    CS -> CTL1 = CS -> CTL1 & ~(CS_CTL1_SELM_MASK | CS_CTL1_DIVM_MASK) |
                CS_CTL1_SELM_3;
    CS ->KEY = 0;
  }
  else
  {
    // Default Frequency
    CS -> KEY = CS_KEY_VAL;
    CS -> CTL0 = 0;
    CS -> CTL0 = CS_CTL0_DCORSEL_1;

    // select clock sources
    CS -> CTL1 = CS_CTL1_SELA_2 | CS_CTL1_SELS_3 | CS_CTL1_SELM_3;
    CS -> KEY = 0;
  }

}
```

Index of comments

1.1 In the future, take an extra hour or two to do the documentation correctly (at no cost to your grade). As long as it is turned in withing that time frame you won't be marked down. Make sure you read the lab manual carefully and include all deliverables!

1.2 - No screenshots for 1s, 100us, or smallest pulse (-20)
- No documentation of accuracy of the 1s pulses. This should include a simple calculation to determine accuracy, and is important for verification (-5)
- No confirmation of accuracy for the 100us pulse (-5)
- No confirmation of accuracy for the 1us pulse (-5)
- Files should be split into header and source files! (-3)