Lab 3 – BlinkLED

The following C code makes the LED blink using nested for loops. The outer loop controls how many times a blink of a certain frequency is performed. The inner for loop determines the delay/frequency of the individual blinks. It would be reasoned that the code, instead of copy and paste, could be put inside a method, the two parameters taken in as the number of times to blink, and the frequency of the blinks. It should also be noted that in order to “blink” the LED, it requires two iterations run of the for loop, once to toggle the LED on and once again to toggle the LED off. Also note, that in the comments, it shows the original code, as well as the modified version to increment instead of decrement, as well as a Morse Code SOS pattern of the LED blinking. The actual code, however, runs the LED for 4 fast blinks, 2 slow blinks, and then a pause.

////////////////////////////////////////////////////////

//Willard Wider

//Lab 03 - BlinkLED

//05/15/18

//Modified blink LED example program

////////////////////////////////////////////////////////

**#include** <ti/devices/msp432p4xx/driverlib/driverlib.h>

**#define** ORIG\_DELAY 100000

**#define** SHORT\_DELAY 10

**#define** LONG\_DELAY

**#define** S\_DELAY 50000

**#define** O\_DELAY 100000

**#define** S\_BLINKS 6

**#define** O\_BLINKS 6

**#define** FULL\_DELAY 500000

**int** **main**(**void**)

{

**volatile** uint32\_t i, j;

// Stop watchdog timer

WDT\_A\_hold(WDT\_A\_BASE);

// Set P1.0 to output direction

**GPIO\_setAsOutputPin**(GPIO\_PORT\_P1,GPIO\_PIN0);

**while**(1)

{

// Toggle P1.0 output

//GPIO\_toggleOutputOnPin(GPIO\_PORT\_P1,GPIO\_PIN0);

//toggle implies switcing it one way or other, a full on of cycle takes 2 toggles

/\* original code

// Delay

//for(i=100000; i>0; i--);

//makeing it incriment instead of decriment

for(i=0; i < 100000; i++)

{

}

\*/

//4 fast blinks

//needs to be 8 cause a blink it a toggle on/off (two code hits)

**for**(i = 0; i < 8; i++)

{

**GPIO\_toggleOutputOnPin**(GPIO\_PORT\_P1,GPIO\_PIN0);

**for**(j = 0; j < S\_DELAY; j++)

{

}

}

//2 slow blinks

**for**(i = 0; i < 4; i++)

{

**GPIO\_toggleOutputOnPin**(GPIO\_PORT\_P1,GPIO\_PIN0);

**for**(j = 0; j < O\_DELAY; j++)

{

}

}

//and a reset delay

//delay of pause

//turn it back off

**for**(i = 0; i < FULL\_DELAY; i++)

{

}

/\*delay for SOS

//first S

for(i = 0; i < S\_BLINKS; i++)

{

GPIO\_toggleOutputOnPin(GPIO\_PORT\_P1,GPIO\_PIN0);

for(j = 0; j < S\_DELAY; j++)

{

}

}

//O

for(i = 0; i < O\_BLINKS; i++)

{

GPIO\_toggleOutputOnPin(GPIO\_PORT\_P1,GPIO\_PIN0);

for(j = 0; j < O\_DELAY; j++)

{

}

}

//second s

for(i = 0; i < S\_BLINKS; i++)

{

GPIO\_toggleOutputOnPin(GPIO\_PORT\_P1,GPIO\_PIN0);

for(j = 0; j < S\_DELAY; j++)

{

}

}

//delay of pause

for(i = 0; i < FULL\_DELAY; i++)

{

}

\*/

}

}