Notebook 0

ELEC 3040/3050 Spring 2021

By Jacob Howard, Undergraduate Computer Engineering

Pre-Lab

Prior to the lab, our only objectives were to familiarize ourselves with everything we will be using in the Lab. We first read the user manual of the NUCLEO-32 board to learn about the features and pin locations.

Once we learned about the NUCLEO-32 board, we studied the features and uses of the Digilent Analog Discovery Studio. This is basically a docking station for our breadboard. We can wire up and give power to the board and also use it with software to send certain signals to the board or microprocessor.

Lastly, we were to familiarize ourselves with the µVision5 software for debugging, installing, and testing code on the microprocessor.

In-Lab

Our Lab started with creating a new project in uVision and downloading the given code to our NUCLEO board. The code can be viewed at the bottom of the page in *Code 1*.

Once the code was downloaded to the board with no errors, we were then required to install and wire up the board correctly to the Breadboard and set up the WaveForms software. We opened the WaveForms software to test if the button was wired up correctly and working properly. The schematic of the wiring can be shown in *Figure 1* at the bottom of the page.

The objective was to hold the button for a certain period and then an LED would start blinking. Once we hold the button again, the LED would stop. The reason for a hold instead of a button press is because of a delay() in the code.

Once everything was set up correctly and working properly, we went into the debug setting in uVision5. This was to get us familiar with how the debugger worked. We were asked to look at the assembly code within the debugger to see the firmware code on the board and how it worked. Specifically how the *delay()* code worked in assembly.

We were then asked to look at the changes in i and j within the assembly which had binary values assigned to R0 and R1. The changes I recorded are shown at the bottom in *Table 1.*

At the end of the lab, we had a *challenge step* to modify the code to perform a different task. I did not get to finish the challenge for this lab, so I cannot provide data for this. This was the end of this Lab project.

Summary

This lab was a good introduction to all the software and hardware we will be using in the Lab. The goal was to get us familiar with everything so other labs will go smoothly. The lab was easy to follow and taught us about everything well.

```
/*====================================================*//* Victor
P. Nelson *//* Modified by: Joey Hines, Spring 2021 *//* Edited
by: Soham Roy, 13Jan2021 *//*   Fixed errors about PB3-PB4 bit
pattern and comments *//* ELEC 3040/3050 - Lab 1, Program 1 *//*
Toggle LED1 while button pressed, with short delay inserted
*//*====================================================*/#include
"stm32l4xx.h" /* Microcontroller information *//* Define global
variables */int toggles; /* number of times LED state toggled
*//*---------------------------------------------------*//*
Initialize GPIO pins used in the program *//* PA11 = push button
*//* PB4 = LDR, PB5 = green LED *//*------------------------------
---------------------*/void PinSetup () { /* Configure PA0 as
input pin to read push button */RCC->AHB2ENR |= 0x01; /* Enable
GPIOA clock (bit 0) */GPIOA->MODER &= ~(0x00C00000); /* General
purpose input mode */ /* Configure PB4,PB3 as output pins to drive
LEDs */RCC->AHB2ENR |= 0x02; /* Enable GPIOB clock (bit 1)
*/GPIOB->MODER &= ~(0x000003C0); /* Clear PB4-PB3 mode bits
*/GPIOB->MODER |=  (0x00000140); /* General purpose output
mode*/}/*---------------------------------------------------------
-*//* Delay function - do nothing for about 1 second *//*---------
-----------------------------------------------*/void delay ()
{int i,j,n;for (i=0; i<20; i++) { //outer loopfor (j=0; j<20000;
j++) { //inner loopn = j; //dummy operation for single-step test}
//do nothing}}/*------------------------------------------------
*//* Main program *//*--------------------------------------------
----*/int main(void) { unsigned int sw1; //state of SW1 unsigned
char led1; //state of LED1 PinSetup(); //Configure GPIO pins led1
= 0; //Initial LED state toggles = 0; //#times LED state changed
/* Endless loop */while (1) { //Can also use: for(;;) {if (led1 ==
0) { //LED off?GPIOB->BSRR = 0x0010 << 16; //Reset PB4=0 to turn
OFF LED (in BSRRupper half)}else { //LED onGPIOB->BSRR = 0x0010;
//Set PB4=1 to turn ON LED (in BSRR
lowhalf)}sw1 = GPIOA->IDR & 0x0800; //Read GPIOA and mask all but
bit 11 /* Wait in loop until SW1 pressed */while (sw1 == 0) {
//Wait for SW1 = 1 on PA11sw1 = GPIOA->IDR & 0x0800; //Read GPIOA
and mask all but bit 11}delay(); //Time delay for button
releaseled1 = ~led1; //Complement LED1 statetoggles++; //Increment
#times LED toggled} /* repeat forever */}lowhalf)}sw1 = GPIOA->IDR
& 0x0800; //Read GPIOA and mask all but bit 11 /* Wait in loop
until SW1 pressed */while (sw1 == 0) { //Wait for SW1 = 1 on
PA11sw1 = GPIOA->IDR & 0x0800; //Read GPIOA and mask all but bit
11}delay(); //Time delay for button releaseled1 = ~led1;
//Complement LED1 statetoggles++; //Increment #times LED toggled}
/* repeat forever */}
```

Code 1

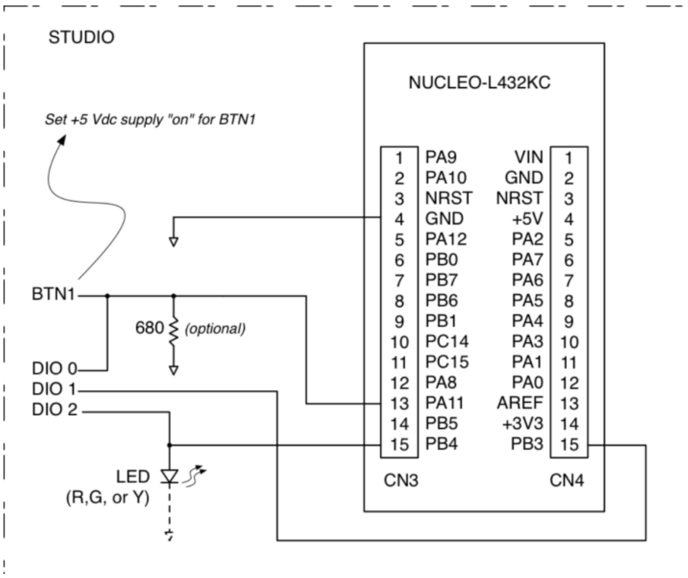

Figure 1

R0	R1
0x00000000	0x00000000
0x00F00000	0x00000000
0x00F00000	0xE000ED88

Table 1