Notebook 0

ELEC 3040/3050 Spring 2021

By Jacob Howard, Undergraduate Computer Engineering

Objective

The objective of this lab review processes for creating, executing and debugging application programs for the STM32L432KCU6 microcontroller. We were to designs a C program, containing a "main" program and two subroutines (or more if needed), to exercise various general purpose I/O (GPIO) ports and elements of the microcontroller. We should be familiar with writing in C, so the main purpose of this lab was to get us used to seeting up the microcontroller ins and ports within our program. This was a difficult task, as for most of us, this was our first time programing like this.

Pre-Lab

Before Lab, we were supposed to write the code that we would be using. The code concisted of two “decase: counters (counters that count up to 9 and then reset). Counter A, which I had written has 2 functions, was to have the ability of counting up to 9, and resetting, but also would switch direction and cound down from 9 if a switch was flipped on. Counter B was just supposed to count up to 9 and reset; it did not change directions. I wrote my entire code before the Lab to ensure time for testing and other tasks the lab required.

In-Lab

In lab, we were supposed to set up two switches with our code. I had set up the pins in my code in the Pre-Lab. Switch 1 (microcontroller I/O port pin PA1) was supposed to turn on and off the entire code. I tried to accomplish this by having a while loop that always looped and checked if Switch 1 was 0 or 1 (0 being off and 1 being on). This is where my code first started

to fail in the debugger. It did not wait for Switch 1 to be off or on. The code just continued to the next step outside of that first loop. I had trouble fixing this part and even the TA could not figure out the problem. Switch 2 was supposed to flip the first counter from counting up to 9 to counting down to 9. Since Switch 1 was not working properly, I do not believe Switch 2 was either.

Once we had our Switches set up, we were supposed to write the outputs to 8 LEDs. I felt that this part was especially tricky in setting up. I tried setting up the output pins to PA[12:9] and PA[8:5} like the lab suggested and even tried PB ports. I had no way of testing and providing data as my code was not functioning correctly in lab, and adjustments I mad out of Lab are not possible to be tested since we do not have all the lab equipment. Basically, PA[12:9] would be 4 LEDs that displayed a binary number from 1-9 (although its possible to display up to 15). Same with PA[8:5]. We were also asked to make a *delay()* function to ensure stability of our code.

After everything was set up properly, we were to wire up our board and test the code. We were asked to us the WaveForms software and using Studio digital DIO0 and DIO1 for Switch 1 and Switch 2 respectivly. Then to display the counter values on the virtual LEDS, we connected port pins PA[8:5] to Studio digital I/O lines DIO7-DIO4, and port pins PA[12:9] to Studio digital I/O lines DIO11-DIO8. Then, we configure DIO11-DIO4 as LEDs in WaveForms. I do not have any data to provide on this part was my code was not functioning properly.

Once everything was set up, we were asked to compile and debug the code. We were supposed to demonstrate our use of breakpoints, watch-windows, single-step debugging, andy any other debug features. I demonstraited use of the debugger to my substitute TA while trying to debug and fix my code.

In Part b of the lab, if everything was working functionally, we were asked to use the digital oscilloscope and connect it to the circuit. I did not get to this part and cannot provide any data, but I can explain what we were required to do. First, we were asked to use wires to connect the indicated microcontroller signals to the Digital I/O (DIO) pins on the Studio. Then configure WaveForms to assign the signals listed in Table A. Table A can be seen below in *Figure 1*. Once we had the physical wiring set up, we were supposed to go into the oscilliscope application and set up all the settings required in the lab manuel. Once we ran the program, we were supposed to accire data related to our circuit. Again, I did not get to this part as my code was not functioning correctly in Part A, so I cannot provide anny data.

Summary

In conclusion, I felt that this lab was difficult comparted to our first lab. The lab manuel did try to help describe how to do some things, but many of us could not get out code working correctly. I think the most challenging part of this lab was properly coding the pin setp on our code and impleminitng them in the counters. I did work on my code more after lab, but was unsure of how it worked and had no way of testing it outside of the lab. Hopfilly I can find out my mistakes in where I went wrong in my code and prepare for next lab with better results. The code I wrote for this lab can be seen below in *Code 1*.

Analyzer signal #	*Microcontroller signal*	*Signal function*
DIO 0	PA1	switch S1 (start/stop)
DIO 1	PA2	switch S2 (up/down)
DIO 4	PA5	Counter A bit 0
DIO 5	PA6	Counter A bit 1
DIO 6	PA7	Counter A bit 2
DIO 7	PA8	Counter A bit 3
DIO 8	PA9	Counter B bit 0
DIO 9	PA10	Counter B bit 1
DIO 10	PA11	Counter B bit 2
DIO 11	PA12	Counter B bit 3

Figure 1

```
/*=========================================================*/
/* Jacob Howard */
/* 0-9 counters */
/*=========================================================*/

#include "stm32l4xx.h" /* Microcontroller information */

/* Define global variables */
static int i;
static int j;
static int counter1; //hex counter for first counter
static int counter2; //hex counter for second counter

static uint16_t sw1; //declare 16-bit variable that matches IDR size (Switch 1 to start and stop counters)
static uint16_t sw2; //Switch 2 to reverse counter 1
static uint16_t out1; //output 1
static uint16_t out2; //output 2


/*---------------------------------------------------------*/
/* Initialize GPIO pins used in the program */
/*---------------------------------------------------------*/
void PinSetup () {
 /* Configure PA0 as input pin to read push button */
		//RCC->AHB2ENR |= 0x01; /* Enable GPIOA clock (bit 0) */
		GPIOA->MODER &= (0xFD5557C3); /* General purpose input mode */		//THIS may be wrong. Might be ~0x0000003C.
		//RCC->AHB2ENR |= 0x02; /* Enable GPIOB clock (bit 1) */
		GPIOB->MODER &= ~(0x003FFFC0); /* Clear PB3-PB10 mode bits */ //tried GPIOB ports as well to see if that would help
		GPIOB->MODER |=  (0x00155540); /* General purpose output mode*/
}

/*---------------------------------------------------------------*/
/* Function to Count up to 9 and then back to zero
/*---------------------------------------------------------------*/

void countUp9 () {
		if (i<=9) {
		i++;
			counter1++;
		}
		else {
		 i = 0;
		}
}
//////////////////////////////////////////
void countDown9 () {
		if (i>=0) {
		i--;
		counter1--;
		}
		else {
			i = 9;
		}
}

/*---------------------------------------------------------------*/
/* Function to Count up indefinately
/*---------------------------------------------------------------*/

		void count () {
			if (j<=0) {
				j++;
				counter2++;
			}
		}
		//delay/////////////////////////////////////////////
```

```
	void delay () {
	int a,b,c;
	for (a=0; a<20; a++) { //outer loop
		for (b=0; b<20000; b++) { //inner loop
			c = j; //dummy operation for single-step test
		} //do nothing
	}
}

/*----------------------------------------------*/
/* Main program */
/*----------------------------------------------*/
int main(void) {
	i = 0;
	j = 0;

	counter1 = 0x0000; //hex counter for first counter
	counter2 = 0x0000; //hex counter for second counter

	PinSetup(); //Configure GPIO pins

	while (1) {
		delay(); //delay

	sw1 = GPIOA->IDR & 0x0002;
	while (sw1 == 0) {
                                                        //Edit: using PA not PB. Maybe could try this instead of all 8 code lines. (GPIOA->ODR
&= ~0x1FE0; // Clear PA[12:5])
	GPIOB->ODR = 0xFFF7; //reset PB3=0
	GPIOB->ODR = 0xFFEF; //reset PB4=0
	GPIOB->ODR = 0xFFDF; //reset PB5=0
	GPIOB->ODR = 0xFFBF; //reset PB6=0
	GPIOB->ODR = 0xFF7F; //reset PB7=0
	GPIOB->ODR = 0xFEFF; //reset PB8=0
	GPIOB->ODR = 0xFDFF; //reset PB9=0
	GPIOB->ODR = 0xFBFF; //reset PB10=0

	sw1 = GPIOA->IDR & 0x0002; //checks to see if PA_0 is on. If not, continues while loop.
	}
		sw2 = GPIOA->IDR & 0x0004;
	if (sw2 == 0) { //checks to see if switch 2 is on or off
	countUp9();
		out1 = counter1;
		GPIOB->ODR = out1 << 5;			//try 5 and the other 9. (was 3 and 7 at first)

		count();
		out2 = counter2;
		GPIOB->ODR = out2 << 9;

		sw1 = GPIOA->IDR & 0x0002;
	} //if sw2 is on this will continue count but will start cound down 9
	else {
		countDown9();
		out1 = counter1;
		GPIOB->ODR = out1 << 3;

	count();
	out2 = counter2;
	GPIOB->ODR = out2 << 7;
	}
}

}
```

Code 1