CPE301 – SPRING 2019

Design Assignment DA6

Student Name: Cody McDonald
Student #: 5000382538
Student Email: mcdonc4@unlv.nevada.edu
Primary Github address: https://github.com/elev8rProcrastinator/submission_da.git
Directory: https://github.com/elev8rProcrastinator/submission_da/tree/master/DA6

Submit the following for all Labs:

1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.

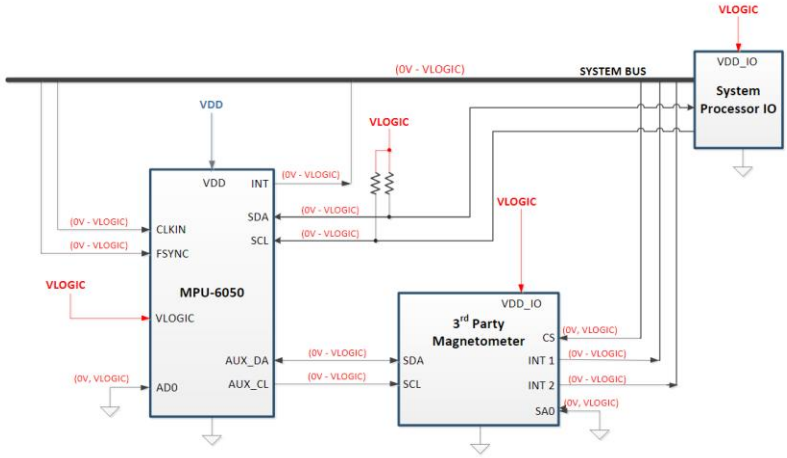
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder ESD301/DA, sub-folder named LABXX, with one document and one video link file for each lab, place modified asm/c files named as LabXX-TYY.asm/c.

3. If multiple asm/c files or other libraries are used, create a folder LabXX-TYY and place these files inside the folder.

4. The folder should have a) Word document (see template), b) source code file(s) and other include files, c) text file with youtube video links (see template).

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

Atmini Xplained
LM35
FTDI USB to serial converter
MPU-6050:



2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A

```
#ifndef F_CPU
#define F_CPU 16000000UL
#endif

//include standard libraries
#include <avr/io.h>
#include <stdlib.h>
#include <stdio.h>
#include <util/delay.h>
#include <math.h>


//include custom libraries
#include "MPU6050_def.h"
#include "i2c_master.h"
#include "uart.h"

//set bits for reading and writing with the mpu6050
#define MPU6050_WRITE 0xD0
#define MPU6050_READ 0xD1

//Global variables
float Acc_x, Acc_y, Acc_z;
float Gyro_x, Gyro_y, Gyro_z;
float temp;
```

```
//Function declarations
void getValues(void);
void uart_char(unsigned char c);
void uart_string(char *s);

int main(void){
	char buffer[20];
	char floatVal[10];
	float Ax, Ay, Az;
	float Gx, Gy, Gz;

	//initialize our communication modes and gyroscope
	i2c_init();
	init_MPU6050();
	init_uart(9600);


	while(1){
		getValues(); //get raw values

		//covert each raw value with their correct divisor
		Ax = Acc_x/16384.0;
		Ay = Acc_y/16384.0;
		Az = Acc_z/16384.0;
		Gx = Gyro_x/16.4;
		Gy = Gyro_y/16.4;
		Gz = Gyro_z/16.4;

		USART_SendString("\n----------------------\n");//Spacer line

		//output Ax value
		dtostrf( Ax, 3, 2, floatVal );
		sprintf(buffer,"Ax = %s g, ",floatVal);
		USART_SendString(buffer);

		//output Ay value
		dtostrf( Ay, 3, 2, floatVal );
		sprintf(buffer,"Ay = %s g, ",floatVal);
		USART_SendString(buffer);

		//output Az value
		dtostrf( Az, 3, 2, floatVal );
		sprintf(buffer,"Az = %s g\n\n",floatVal);
		USART_SendString(buffer);

		//output Gx value
		dtostrf( Gx, 3, 2, floatVal );
		sprintf(buffer,"Gx = %s degrees/s, ",floatVal);
		USART_SendString(buffer);

		//output Gy value
		dtostrf( Gy, 3, 2, floatVal );
		sprintf(buffer,"Gy = %s degrees/s, ",floatVal);
		USART_SendString(buffer);

		//output Gz value
		dtostrf( Gz, 3, 2, floatVal );
		sprintf(buffer,"Gz = %s degrees/s",floatVal);
		USART_SendString(buffer);

		USART_SendString("\n----------------------\n"); //end line break
		_delay_ms(1000);
	}
```

```
	return 0;
}


void init_uart(uint16_t baudrate){

	uint16_t UBRR_val = (F_CPU/16)/(baudrate-1);

	UBRR0H = UBRR_val >> 8;
	UBRR0L = UBRR_val;

	UCSR0B |= (1<<TXEN0) | (1<<RXEN0) | (1<<RXCIE0);
	UCSR0C |= (1<<USBS0) | (3<<UCSZ00);
}

void uart_char(unsigned char c){

	while(!(UCSR0A & (1<<UDRE0))); // wait
	UDR0 = c; // output character
}

void uart_string(char *s){
	while(*s){
		uart_char(*s);
		s++;
	}
}

void init_MPU6050(void){
	_delay_ms(150);
	i2c_start(MPU6050_WRITE); /* Start with device write address */
	i2c_write(SMPLRT_DIV);  /* Write to sample rate register */
	i2c_write(0x07);           /* 1KHz sample rate */
	i2c_stop();

	i2c_start(MPU6050_WRITE);
	i2c_write(PWR_MGMT_1); /* Write to power management register */
	i2c_write(0x01); /* X axis gyroscope reference frequency */
	i2c_stop();

	i2c_start(MPU6050_WRITE);
	i2c_write(CONFIG); /* Write to Configuration register */
	i2c_write(0x00);     /* Fs = 8KHz */
	i2c_stop();

	i2c_start(MPU6050_WRITE);
	i2c_write(GYRO_CONFIG); /* Write to Gyro configuration register */
	i2c_write(0x18);           /* Full scale range +/- 2000 degree/C */
	i2c_stop();

	i2c_start(MPU6050_WRITE);
	i2c_write(INT_ENABLE);  /* Write to interrupt enable register */
	i2c_write(0x01);
	i2c_stop();
}

void getValues(void){

	//Start system by setting cursor
	i2c_start(MPU6050_WRITE);
	i2c_write(ACCEL_XOUT_H);
	i2c_stop();
```

```
        //Start reading process for each value
        i2c_start(MPU6050_READ);
        Acc_x = (((int)i2c_read_ack()<<8) | (int)i2c_read_ack());

        Acc_y = (((int)i2c_read_ack()<<8) | (int)i2c_read_ack());

        Acc_z = (((int)i2c_read_ack()<<8) | (int)i2c_read_ack());

        temp = (((int)i2c_read_ack()<<8) | (int)i2c_read_ack());

        Gyro_x = (((int)i2c_read_ack()<<8) | (int)i2c_read_ack());

        Gyro_y = (((int)i2c_read_ack()<<8) | (int)i2c_read_ack());

        Gyro_z = (((int)i2c_read_ack()<<8) | (int)i2c_read_ack());

        //Stop reading
        i2c_stop();
}
```

3. DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A

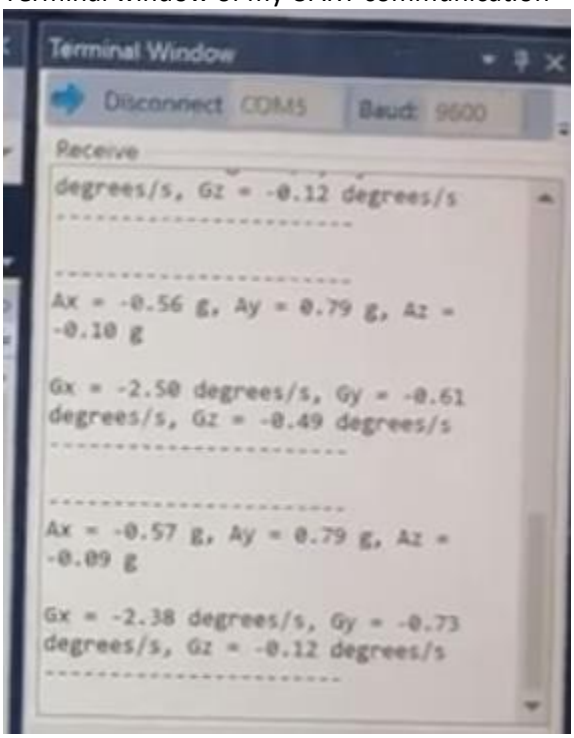
N/A

4. SCHEMATICS

Refer to board demos for connections of the motors and drivers to the pins on the microcontroller

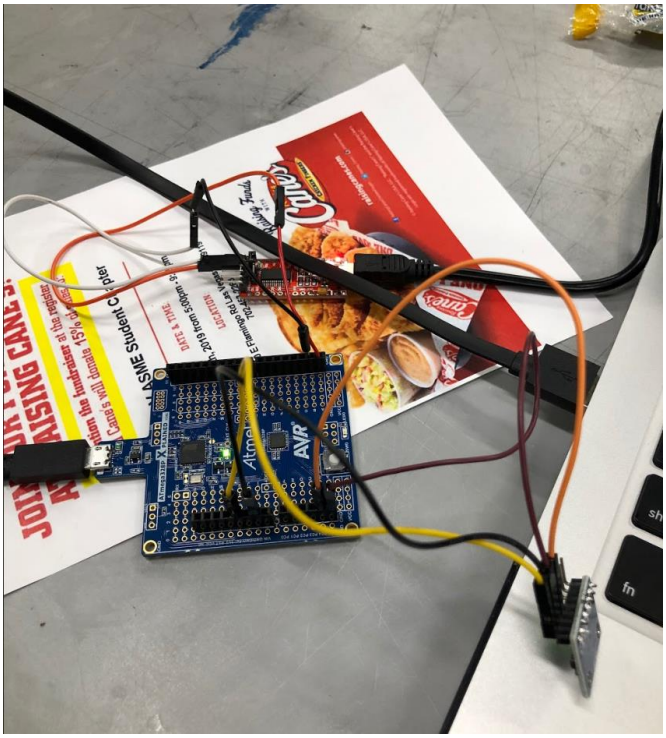
5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

Terminal window of my UART communication



6. SCREENSHOT OF EACH DEMO (BOARD SETUP)

This is the set-up of my controller setup



7. VIDEO LINKS OF EACH DEMO
https://youtu.be/YTTyRZ8qZz4

8. GITHUB LINK OF THIS DA
https://github.com/elev8rProcrastinator/submission_da/tree/master/DA6

Student Academic Misconduct Policy
http://studentconduct.unlv.edu/misconduct/policy.html

"*This assignment submission is my own, original work*".
Cody McDonald