CPE301 – SPRING 2019

Design Assignment 6

Student Name: James Skelly
Student #: 2000945485
Student Email: skellj1@unlv.nevada.edu
Primary Github address: https://github.com/skellj1/submission_da
Directory: skellj1/submission_da

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

Components used for this DA include breadboard, Xplained Mini, jumper wire, MPU6050 Accelerometer/Gyroscope module, FTDI Chip (USART), iPhone for recording, USB cable, Atmel Studio 7.

A block diagram is shown below describing the flow of data in this DA.

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

```
/*
 * DA6.c
 *
 * Created: 5/4/2019 8:06:22 PM
 * Author : J. Skelly
 */

#define F_CPU 16000000UL         // Define CPU clock Frequency

// include necessary headers for operation of program
#include <avr/io.h>
#include <util/delay.h>
#include <inttypes.h>
#include <stdlib.h>
#include <stdio.h>
#include "MPU6050_res_define.h"
#include "I2C_MasterH.h"
#include "USART_RS232_H.h"

// Function prototyping
void scale_values(void);
void send_values(void);

// initializing variables
char buffer[20], float_[10];
float Ax,Ay,Az,t;
float Gx=0,Gy=0,Gz=0;
float Acc_x,Acc_y,Acc_z;
float Temperature;
float Gyro_x,Gyro_y,Gyro_z;

int main()
{
        I2C_Init();                     // I2C init function call
        MPU6050_Init();         // Initialize MPU6050
        USART_Init(9600);               // Initialize USART, BAUD RATE = 9600

        while(1)
        {
                Read_RawValue();// call function to read raw values
                scale_values();  // call function to scale values into proper units
                send_values();   // send values to USART
                _delay_ms(1000);
        }
}

void Read_RawValue()                    // Read values from gyro, wait for acknowledgement
{
        MPU_Start_Loc();
        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());
        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_Nack());
        I2C_Stop();
}


void scale_values()
{
        // Scale accelerometer values into G force units
        Ax = Acc_x/16384.0;
        Ay = Acc_y/16384.0;
        Az = Acc_z/16384.0;
```

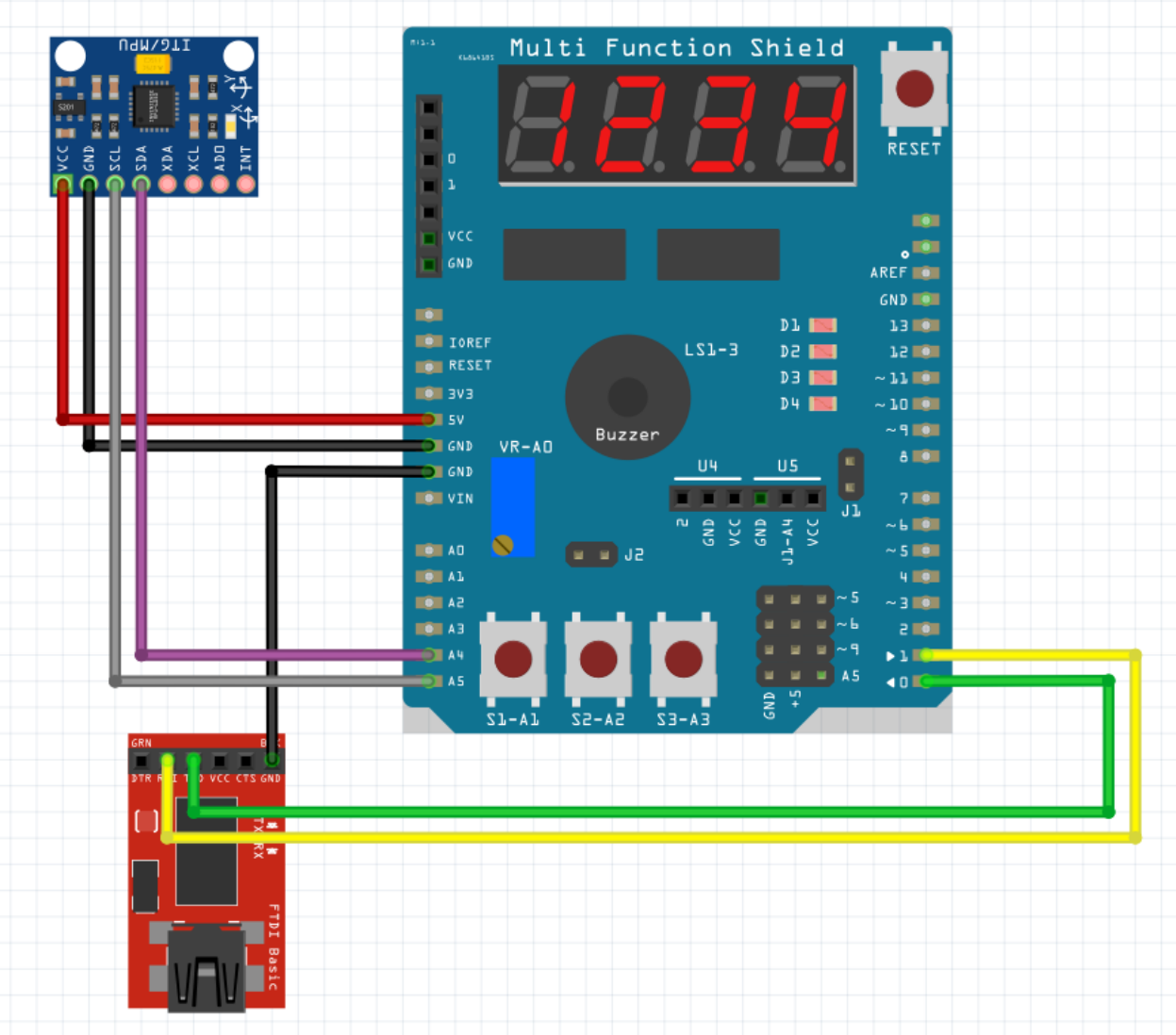
```
        // Scale gyroscope values into degrees/second units
        Gx = Gyro_x/16.4;
        Gy = Gyro_y/16.4;
        Gz = Gyro_z/16.4;
}

void send_values()
{
                        // Send values to UART, formatting, unit outputs
                        dtostrf( Ax, 3, 2, float_ );
                        sprintf(buffer," Acc_x = %s g\t",float_);
                        USART_SendString(buffer);

                        dtostrf( Ay, 3, 2, float_ );
                        sprintf(buffer," Acc_y = %s g\t",float_);
                        USART_SendString(buffer);

                        dtostrf( Az, 3, 2, float_ );
                        sprintf(buffer," Acc_z = %s g\t",float_);
                        USART_SendString(buffer);

                        dtostrf( Gx, 3, 2, float_ );
                        sprintf(buffer," Gyro_x = %s%c/s\t",float_,0xF8);
                        USART_SendString(buffer);

                        dtostrf( Gy, 3, 2, float_ );
                        sprintf(buffer," Gyro_y = %s%c/s\t",float_,0xF8);
                        USART_SendString(buffer);

                        dtostrf( Gz, 3, 2, float_ );
                        sprintf(buffer," Gyro_z = %s%c/s\r\n",float_,0xF8);
                        USART_SendString(buffer);
}

void MPU6050_Init()                     // Gyro initialization function
{
        _delay_ms(150);         // Power up time >100ms
        I2C_Start_Wait(0xD0);   // Start at device that will be written to address
        I2C_Write(SMPLRT_DIV);  // Write to sample rate register
        I2C_Write(0x07);        // set 1KHz sample rate
        I2C_Stop();
        I2C_Start_Wait(0xD0);
        I2C_Write(PWR_MGMT_1);  // Write to power management register
        I2C_Write(0x01);        // X axis gyroscope reference frequency
        I2C_Stop();
        I2C_Start_Wait(0xD0);
        I2C_Write(CONFIG);              // Write to Configuration register
        I2C_Write(0x00);        // Fs = 8KHz
        I2C_Stop();
        I2C_Start_Wait(0xD0);
        I2C_Write(GYRO_CONFIG); // Write to Gyroscope config. register
        I2C_Write(0x18);        // Full scale range +/- 2000 degree/C
        I2C_Stop();
        I2C_Start_Wait(0xD0);
        I2C_Write(INT_ENABLE);  // Write to interrupt enable register
        I2C_Write(0x01);
        I2C_Stop();
}

void MPU_Start_Loc()
{
        I2C_Start_Wait(0xD0);   // I2C start with device write address
        I2C_Write(ACCEL_XOUT_H); // Write start location address from where to read
        I2C_Repeated_Start(0xD1);       // I2C start with device read address
}
```

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

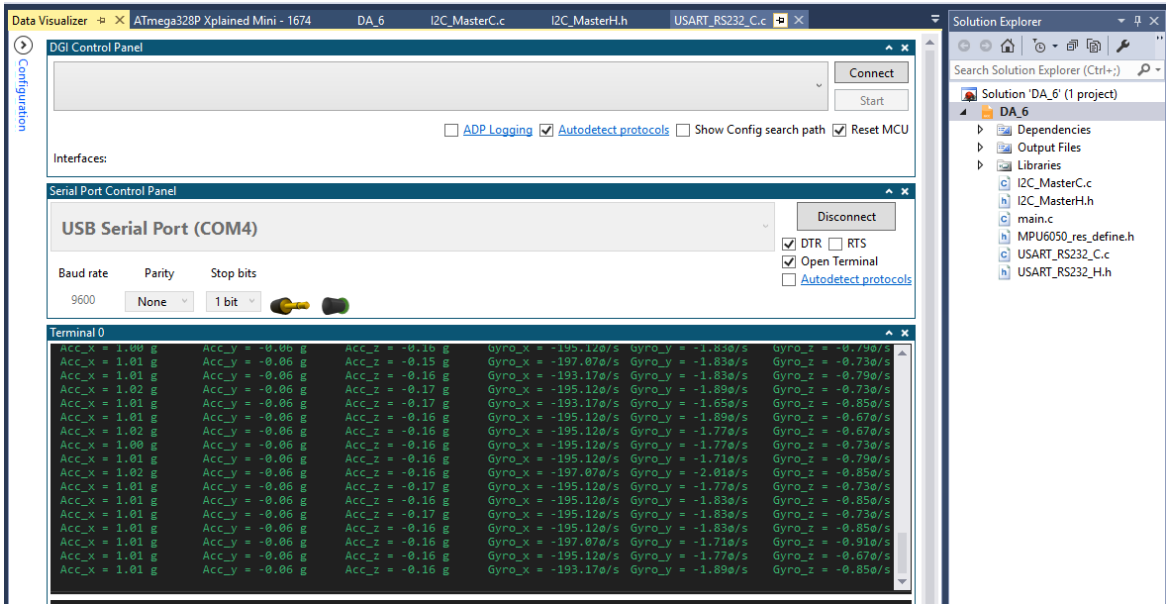
Not applicable for this assignment.

4. SCHEMATICS

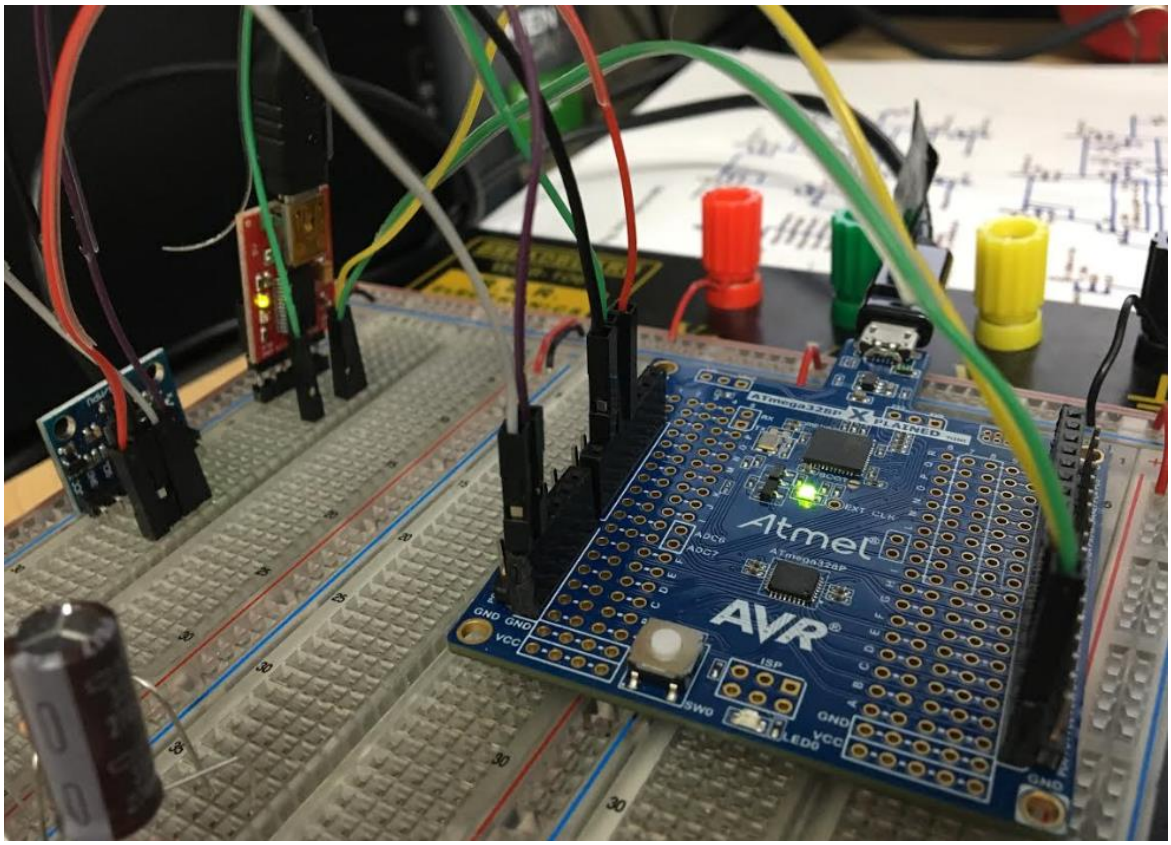
The below schematic was generated using Fritzing's Breadboard Schematic Creator.



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



6. SCREENSHOT OF EACH DEMO (BOARD SETUP)



7. VIDEO LINKS OF EACH DEMO

https://www.youtube.com/watch?v=ROYeCyrbeLc

8. GITHUB LINK OF THIS DA

https://github.com/skellj1/submission_da

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

"*This assignment submission is my own, original work*".
James W. Skelly