Design Assignment 6

Student Name: Moriah Wingrove
Student #: 5000156247
Student Email: Wingrove@unlv.nevada.edu
Primary Github address: https://github.com/windew/Tiny_Dragons.git
Directory:

 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

Atmega 328pb qty2
MPU6050

INITIAL/MODIFIED/DEVELOPED CODE OF TASK 6
C code

```c
/*
 * Design_assignment_6.c
 *
 * Created: 12/11/2019 4:25:28 PM
 * Author : Moriah Wingrove
 */


#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <inttypes.h>
#include <stdlib.h>
#include <stdio.h>
#include "MPU6050_res_define.h"
#include "I2C_Master_H_file.h"
#include "USART_RS232_H_file.h"

float Acc_x,Acc_y,Acc_z,Temperature,Gyro_x,Gyro_y,Gyro_z;

void MPU6050_Init()
{
	_delay_ms(150);
	/* Power up time >100ms                   */
	I2C_Start_Wait(0xD0);                                                /*
Start with device write address    */
	I2C_Write(SMPLRT_DIV);                                               /*
Write to sample rate register      */
	I2C_Write(0x07);                                                     /*
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                              */
```

```c
        I2C_Stop();

        I2C_Start_Wait(0xD0);
        I2C_Write(GYRO_CONFIG);                                         /*
Write to Gyro configuration 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                  */
}

void Read_RawValue()
{
        MPU_Start_Loc();                                               /*
Read Gyro values */
        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();
}

int main()
{
        char buffer[20], float_[10];
        float Xa,Ya,Za,t;
        float Xg=0,Yg=0,Zg=0;
        I2C_Init();
        MPU6050_Init();
        USART_Init(9600);

        while(1)
        {
                Read_RawValue();

                Xa = Acc_x/16384.0;                                    /*
Divide raw value by sensitivity scale factor to get real values */
                Ya = Acc_y/16384.0;
                Za = Acc_z/16384.0;

                Xg = Gyro_x/16.4;
```

```c
        Yg = Gyro_y/16.4;
        Zg = Gyro_z/16.4;

        dtostrf( Xa, 3, 2, float_ );                            /* Takes the
first parameter and formts is with size 3 precission 2 and puts it in float_ */
        sprintf(buffer," Ax = %s g\t",float_);          /* This is where the
value is being put as a string onto the buffer variable */
        USART_SendString(buffer);                          /* This is
where it is being displayed on the serial terminal  */

        dtostrf( Ya, 3, 2, float_ );                            /* Takes the
first parameter and formts is with size 3 precission 2 and puts it in float_v */
        sprintf(buffer," Ay = %s g\t",float_);          /* This is where the
value is being put as a string onto the buffer variable */
        USART_SendString(buffer);                          /* This is
where it is being displayed on the serial terminal  */

        dtostrf( Za, 3, 2, float_ );                            /* Takes the
first parameter and formts is with size 3 precission 2 and puts it in float_ */
        sprintf(buffer," Az = %s g\t",float_);          /* This is where the
value is being put as a string onto the buffer variable */
        USART_SendString(buffer);                          /* This is
where it is being displayed on the serial terminal  */

        dtostrf( Xg, 3, 2, float_ );                            /* Takes the
first parameter and formts is with size 3 precission 2 and puts it in float_ */
        sprintf(buffer," Gx = %s%c/s\t",float_,0xF8);    /* This is where the value
is being put as a string onto the buffer variable */
        USART_SendString(buffer);                          /* This is
where it is being displayed on the serial terminal  */

        dtostrf( Yg, 3, 2, float_ );                            /* Takes the
first parameter and formts is with size 3 precission 2 and puts it in float_ */
        sprintf(buffer," Gy = %s%c/s\t",float_,0xF8);    /* This is where the value
is being put as a string onto the buffer variable */
        USART_SendString(buffer);                          /* This is
where it is being displayed on the serial terminal  */

        dtostrf( Zg, 3, 2, float_ );                            /* Takes the
first parameter and formts is with size 3 precission 2 and puts it in float_ */
        sprintf(buffer," Gz = %s%c/s\r\n",float_,0xF8); /* This is where the value
is being put as a string onto the buffer variable */
        USART_SendString(buffer);                          /* This is
where it is being displayed on the serial terminal  */
        _delay_ms(1000);                                       /*
This just be a delaylay for the thingys */
    }
}
```
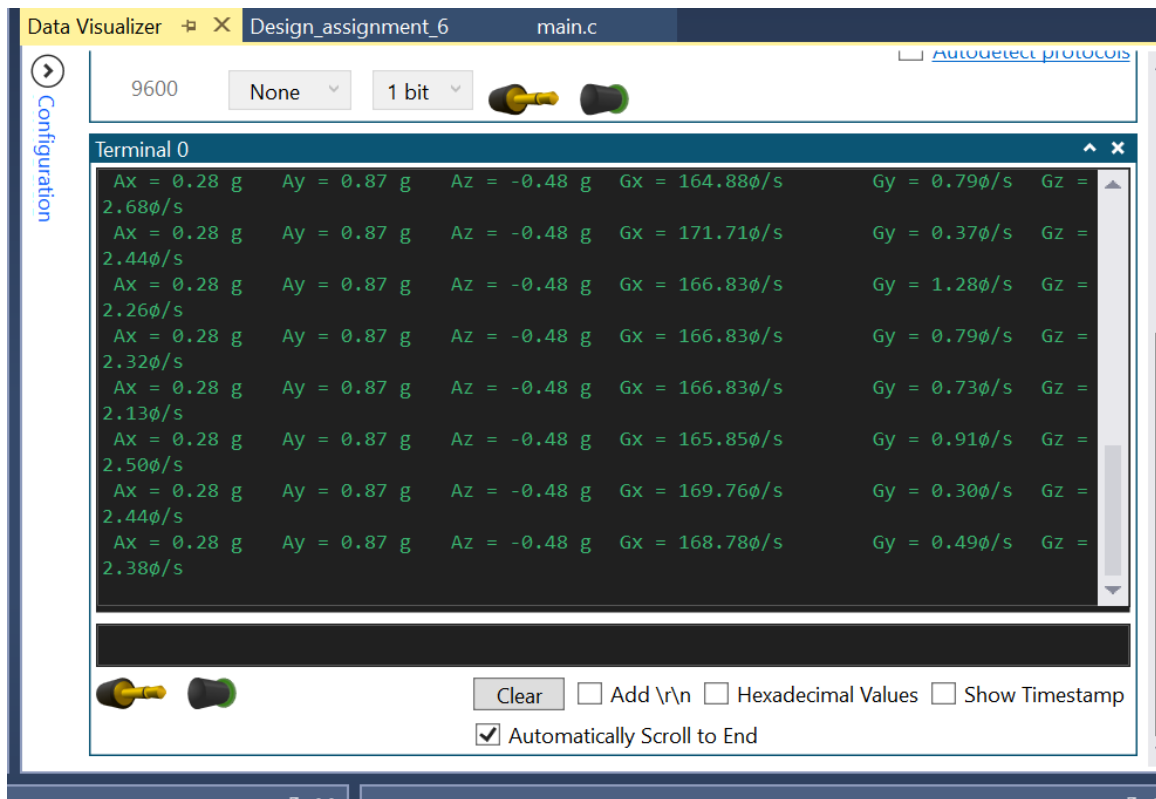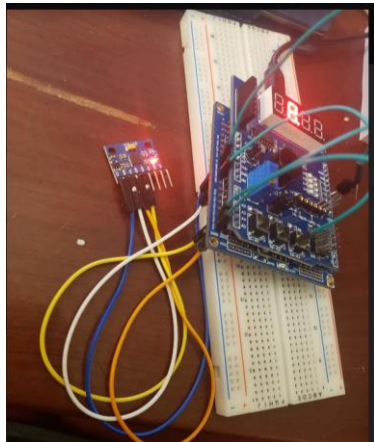
2.   SCHEMATICS

Use fritzing.org

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

Terminal 0 output:

```
 Ax = 0.28 g    Ay = 0.87 g    Az = -0.48 g   Gx = 164.88ø/s       Gy = 0.79ø/s   Gz =
2.68ø/s
 Ax = 0.28 g    Ay = 0.87 g    Az = -0.48 g   Gx = 171.71ø/s       Gy = 0.37ø/s   Gz =
2.44ø/s
 Ax = 0.28 g    Ay = 0.87 g    Az = -0.48 g   Gx = 166.83ø/s       Gy = 1.28ø/s   Gz =
2.26ø/s
 Ax = 0.28 g    Ay = 0.87 g    Az = -0.48 g   Gx = 166.83ø/s       Gy = 0.79ø/s   Gz =
2.32ø/s
 Ax = 0.28 g    Ay = 0.87 g    Az = -0.48 g   Gx = 166.83ø/s       Gy = 0.73ø/s   Gz =
2.13ø/s
 Ax = 0.28 g    Ay = 0.87 g    Az = -0.48 g   Gx = 165.85ø/s       Gy = 0.91ø/s   Gz =
2.50ø/s
 Ax = 0.28 g    Ay = 0.87 g    Az = -0.48 g   Gx = 169.76ø/s       Gy = 0.30ø/s   Gz =
2.44ø/s
 Ax = 0.28 g    Ay = 0.87 g    Az = -0.48 g   Gx = 168.78ø/s       Gy = 0.49ø/s   Gz =
2.38ø/s
```

4.   SCREENSHOT OF EACH DEMO (BOARD SETUP)

5.   VIDEO LINKS OF EACH DEMO

6.   GITHUB LINK OF THIS DA

https://github.com/windew/Tiny_Dragons

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

"This assignment submission is my own, original work".