CPE301 – SPRING 2019

Design Assignment 6

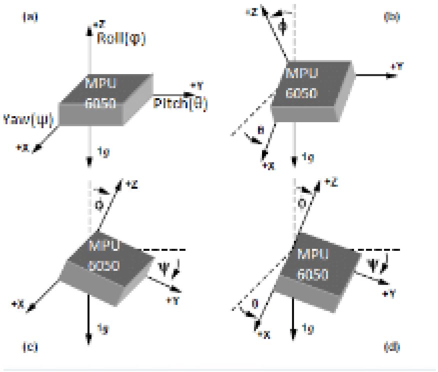
Student Name: Tyler Gardenhire
Student #: 8000450294
Student Email: gardenhi@unlv.nevada.edu
Primary Github address: gardenhi@unlv.nevada.edu
Directory: https://github.com/tylergardenhire/submission_projects.git

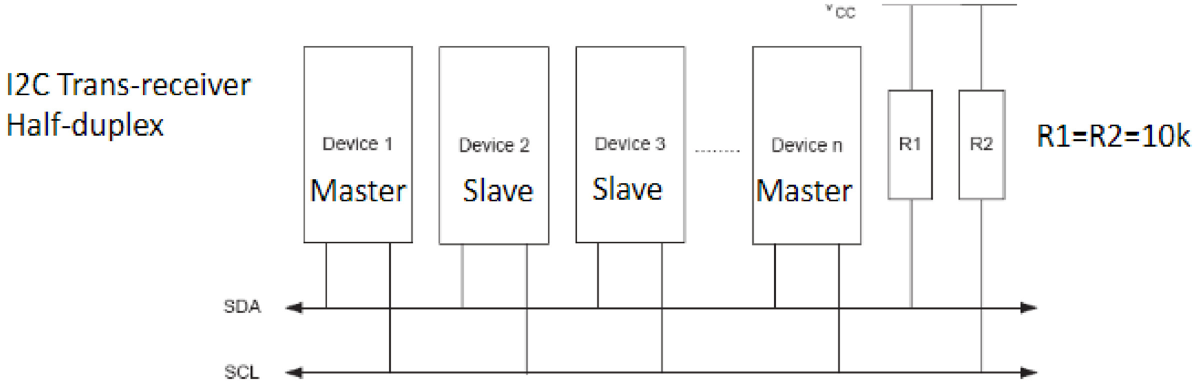
Submit the following for all Labs:

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

Atmel Studio 7 w/ AVR assembly and simulator, Atmega328p board.





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

Task 1 C code:

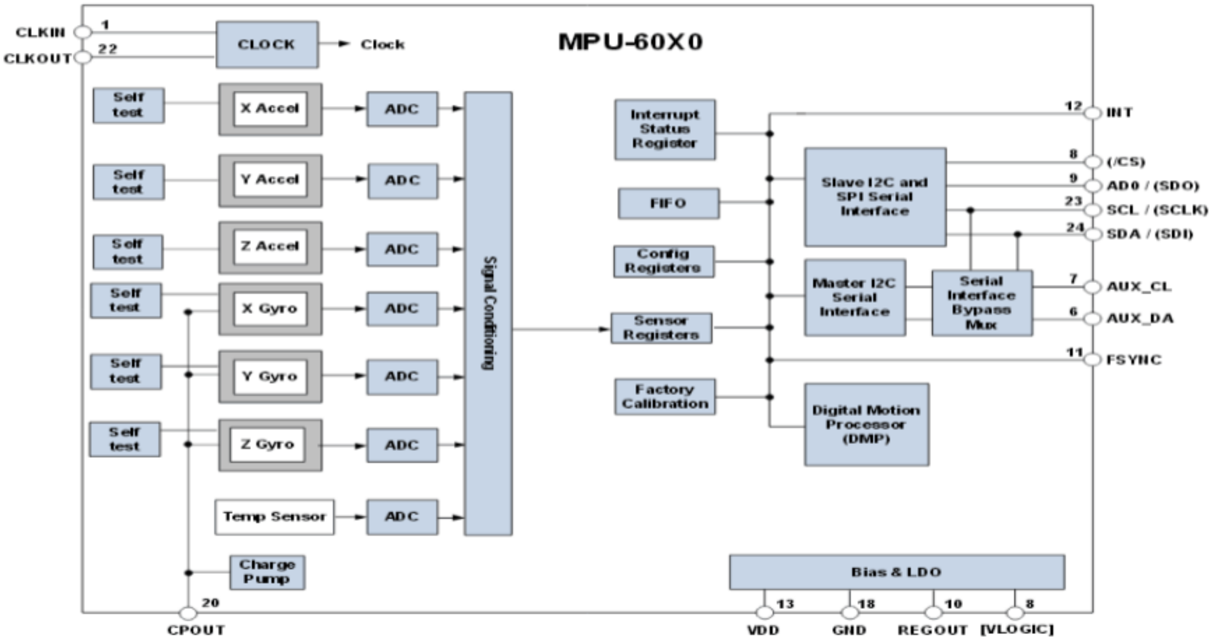
```
#define F_CPU 16000000UL

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

#include "MPU6050_def.h"    //include library files
#include "i2c_master.h"
#include "uart.h"

#define MPU6050_WRITE 0xD0
#define MPU6050_READ 0xD1

float Acc_x;
float Acc_z;
float Acc_y;

float Gyro_x;
float Gyro_z;
float Gyro_y;

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_putc(unsigned char c){

        while(!(UCSR0A & (1<<UDRE0))); //wait until sending is possible
        UDR0 = c; //output character saved in c
}

void uart_puts(char *s){
        while(*s){
                uart_putc(*s);
                s++;
        }
}

void init_MPU6050(void){
        _delay_ms(150);
        i2c_start(MPU6050_WRITE); //set Gyroscope Sample Rate = 1 KHz
        i2c_write(SMPLRT_DIV); //sample rate = gyroscope output rate divided by SMPLRT_DIV
        i2c_write(0x07); //gyroscope Output Rate = 8kHz, Sample Rate = Gyroscope Output
Rate / (1 + SMPLRT_DIV)
        i2c_stop();
        i2c_start(MPU6050_WRITE);
        i2c_write(PWR_MGMT_1);
        i2c_write(0x01); //PLL with X axis gyroscope reference
        i2c_stop();

        i2c_start(MPU6050_WRITE);
        i2c_write(CONFIG); //DLPF setting
        i2c_write(0x00);
        i2c_stop();

        i2c_start(MPU6050_WRITE);
        i2c_write(GYRO_CONFIG); //gyroscope scale range
        i2c_write(0x18);            //accelerometer range
        i2c_stop();

        i2c_start(MPU6050_WRITE);
        i2c_write(INT_ENABLE); //DATA_RDY_EN = 1
        i2c_write(0x01);
        i2c_stop();
}

void getreading(void){

        i2c_start(MPU6050_WRITE);
        i2c_write(ACCEL_XOUT_H); //set pointer
        i2c_write(ACCEL_YOUT_H); //set pointer
        i2c_write(ACCEL_ZOUT_H); //set pointer
        i2c_stop();

        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());

        i2c_stop();

        i2c_start(MPU6050_WRITE);
        i2c_write(GYRO_XOUT_H);
        i2c_write(GYRO_YOUT_H);
        i2c_write(GYRO_ZOUT_H);
        i2c_stop();

        i2c_start(MPU6050_READ);

        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(void){
        char buffer[20], float_[10];
        float Xa, Ya, Za;
        float Xg, Yg, Zg;
        init_uart(9600);
```

```
    i2c_init();
    init_MPU6050();

    while(1){
        getreading();
        Xa = Acc_x/16384.0;
        Ya = Acc_y/16384.0;
        Za = Acc_x/16384.0;

        Xg = Gyro_x/16.4;
        Yg = Gyro_y/16.4;
        Zg = Gyro_z/16.4;

        dtostrf( Xa, 3, 2, float_ );
        sprintf(buffer,"Ax: %s, ",float_);
        USART_SendString(buffer);
        dtostrf( Ya, 3, 2, float_ );
        sprintf(buffer,"Ay: %s, ",float_);
        USART_SendString(buffer);
        dtostrf( Za, 3, 2, float_ );
        sprintf(buffer,"Az: %s, \n",float_);
        USART_SendString(buffer);

        dtostrf( Xg, 3, 2, float_ );
        sprintf(buffer,"Gx: %s, ",float_);
        USART_SendString(buffer);
        dtostrf( Yg, 3, 2, float_ );
        sprintf(buffer,"Gy: %s, ",float_);
        USART_SendString(buffer);
        dtostrf( Zg, 3, 2, float_ );
        sprintf(buffer,"Gz: %s, \n\n",float_);
        USART_SendString(buffer);

        _delay_ms(1000);
    }

    return 0;
}
```

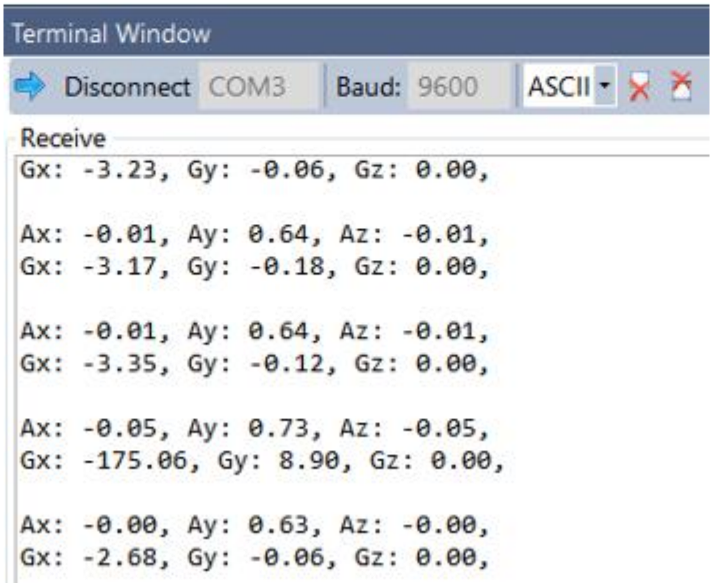
3. DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A

N/A

4. SCHEMATICS

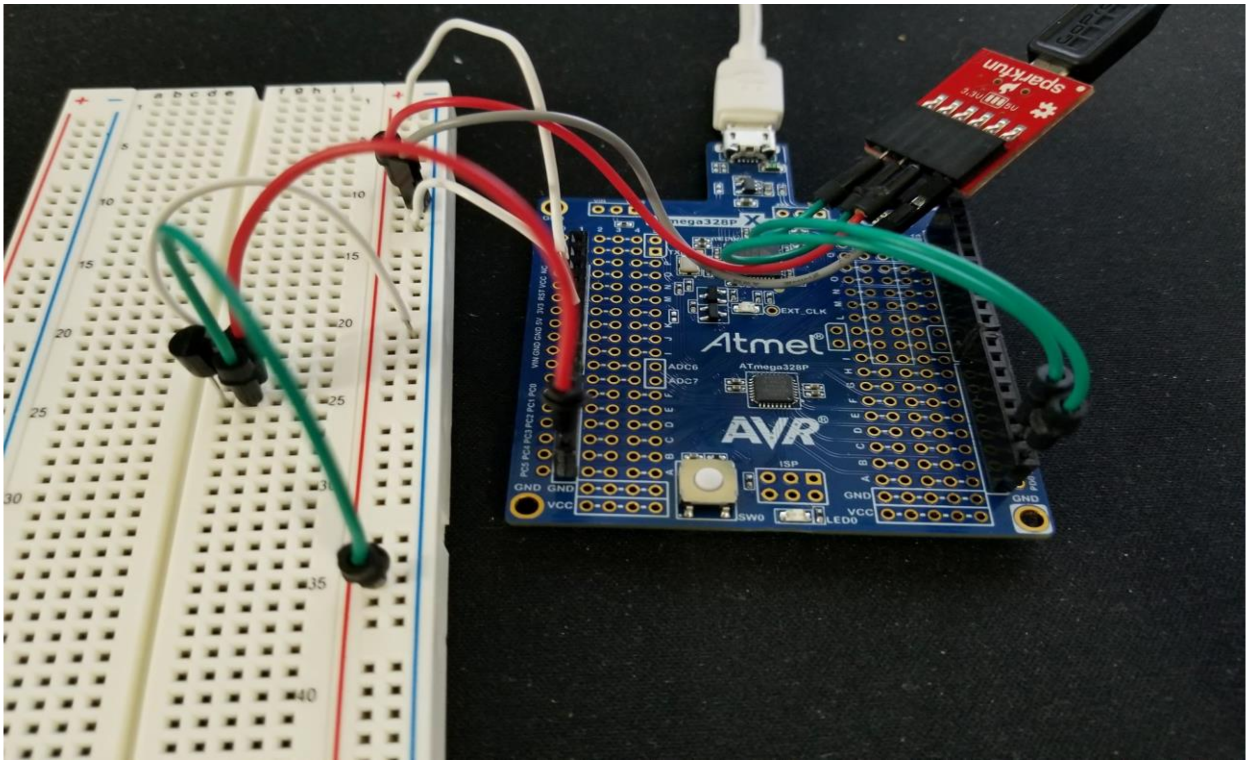


5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)



```
Gx: -3.23, Gy: -0.06, Gz: 0.00,

Ax: -0.01, Ay: 0.64, Az: -0.01,
Gx: -3.17, Gy: -0.18, Gz: 0.00,

Ax: -0.01, Ay: 0.64, Az: -0.01,
Gx: -3.35, Gy: -0.12, Gz: 0.00,

Ax: -0.05, Ay: 0.73, Az: -0.05,
Gx: -175.06, Gy: 8.90, Gz: 0.00,

Ax: -0.00, Ay: 0.63, Az: -0.00,
Gx: -2.68, Gy: -0.06, Gz: 0.00,
```

6. SCREENSHOT OF EACH DEMO (BOARD SETUP)



7. VIDEO LINKS OF EACH DEMO

N/A

8. GITHUB LINK OF THIS DA

https://github.com/tylergardenhire/submission_projects.git

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

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