CPE301 – SPRING 2019

Design Assignment 6

Student Name: Ron Joshua Recrio
Student #: 5003825419
Student Email: recrio@unlv.nevada.edu
Primary Github address: https://github.com/recrio/submissions
Directory: /DesignAssignments/DA6

Submit the following for all Labs:

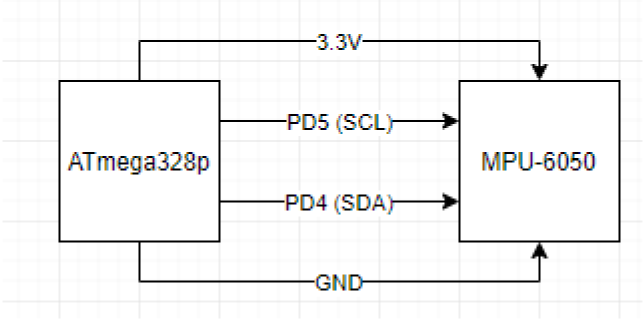
1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder 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

List of Components used:

- ATmega328p Xplained Mini
- MPU-6050

Block diagram with pins used in the Atmega328P



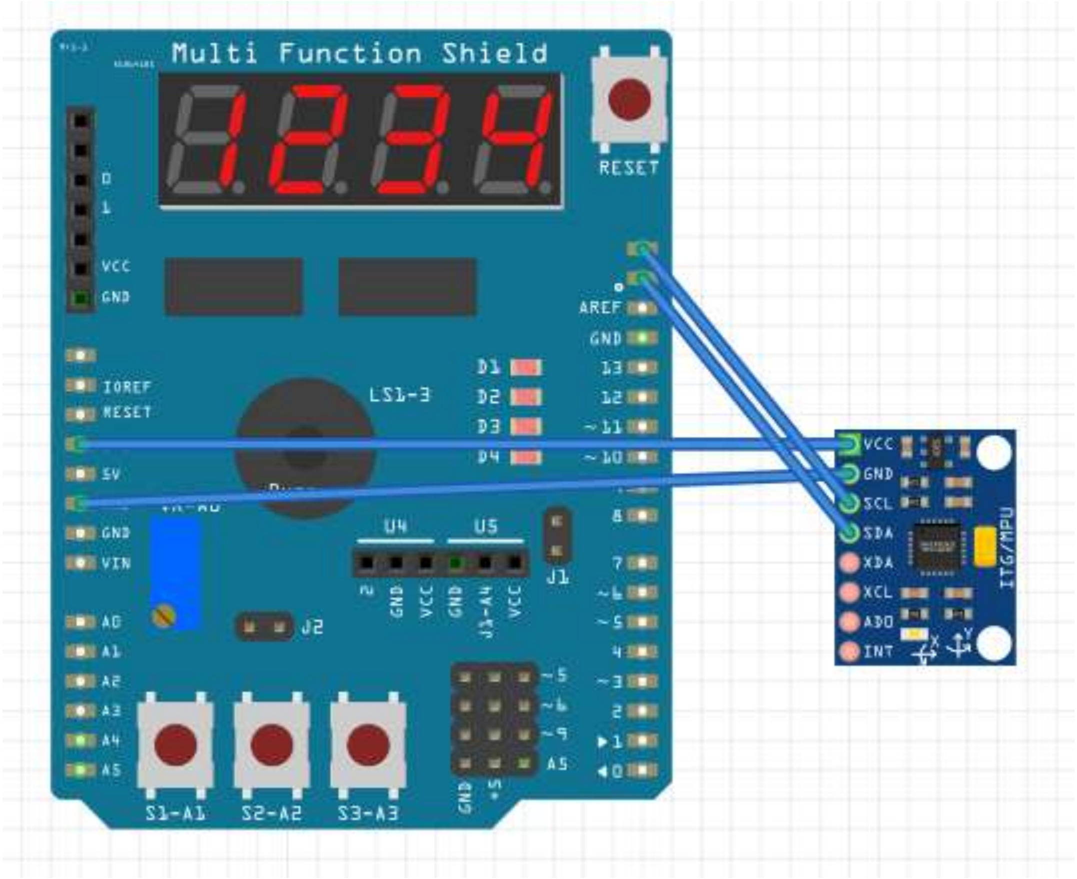
2. INITIAL CODE OF TASK 1

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

#include <avr/io.h>
#include <util/delay.h>
#include <math.h>
#include <stdlib.h>                                                                   /*
Include standard library file */
#include <stdio.h>                                                                    /*
Include standard library file */
#include "MPU6050_def.h"                                                /* Include MPU6050
register define file */
#include "i2c_master.h"                                                 /* Include I2C
Master header file */
#include "uart.h"                                                /* Include USART header
file */

#define MPU6050_WRITE 0xD0
#define MPU6050_READ 0xD1

// stores raw values
float Acc_x, Acc_y, Acc_z, Gy_x, Gy_y, Gy_z;

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); // UART TX (Transmit - senden)
einschalten
       UCSR0C |= (1<<USBS0) | (3<<UCSZ00); //Modus Asynchron 8N1 (8 Datenbits, No Parity,
1 Stopbit)
}

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);
       /* Power up time >100ms */
       i2c_start(MPU6050_WRITE); // Set Gyroscope Sample Rate = 1 KHz, Accelerometer
Sample Rate = 1 KHz (default)
```

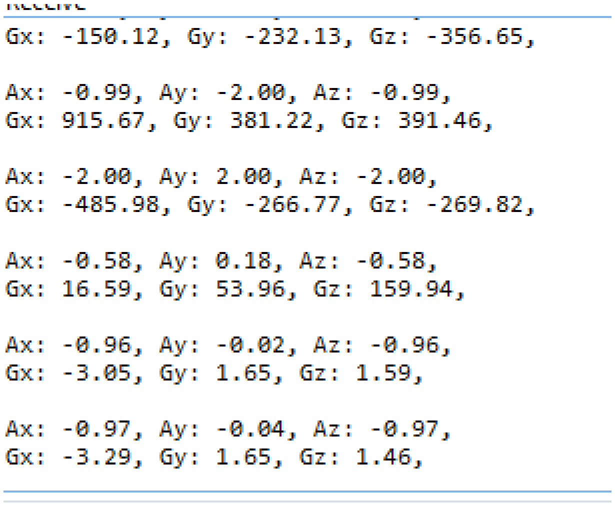
```
	i2c_write(SMPLRT_DIV); // Sample Rate is generated by dividing the gyroscope
output rate by SMPLRT_DIV
	i2c_write(0x07); // Gyroscope Output Rate = 8kHz, Sample Rate = Gyroscope Output
Rate / (1 + SMPLRT_DIV)
	i2c_stop();

	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); //Frame Synchronization & Digital Low Pass Filter (DLPF)
setting
	i2c_write(0x00);
	i2c_stop();

	i2c_start(MPU6050_WRITE);
	i2c_write(GYRO_CONFIG); //gyroscopes’ scale range = FS_SEL selects = 11 = ± 2000
°/s
	i2c_write(0x18);		// accelerometer range = ± 2g (default)
	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); // point to this addr
	i2c_stop();

	i2c_start(MPU6050_READ);

	// take in high byte and combine with low byte
	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); // point to gyro addr
	i2c_stop();

	i2c_start(MPU6050_READ);

	// take in high byte and combine with low byte
	Gy_x = (((int)i2c_read_ack()<<8) | (int)i2c_read_ack());
	Gy_y = (((int)i2c_read_ack()<<8) | (int)i2c_read_ack());
	Gy_z = (((int)i2c_read_ack()<<8) | (int)i2c_read_ack());

	i2c_stop();
}
```

```
int main(void){
       char buffer[20], float_[10];
       float Xa, Ya, Za, Xg, Yg, Zg;
       init_uart(9600);
       i2c_init();
       init_MPU6050();

       while(1){
              getreading();
              Xa = Acc_x/16384.0;                                                  /*
Divide raw value by sensitivity scale factor to get real values */
              Ya = Acc_y/16384.0;
              Za = Acc_x/16384.0;

              Xg = Gy_x/16.4;
              Yg = Gy_y/16.4;
              Zg = Gy_z/16.4;

              dtostrf( Xa, 3, 2, float_ );                                  /* Take
values in buffer to send all parameters over USART */
              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. SCHEMATICS

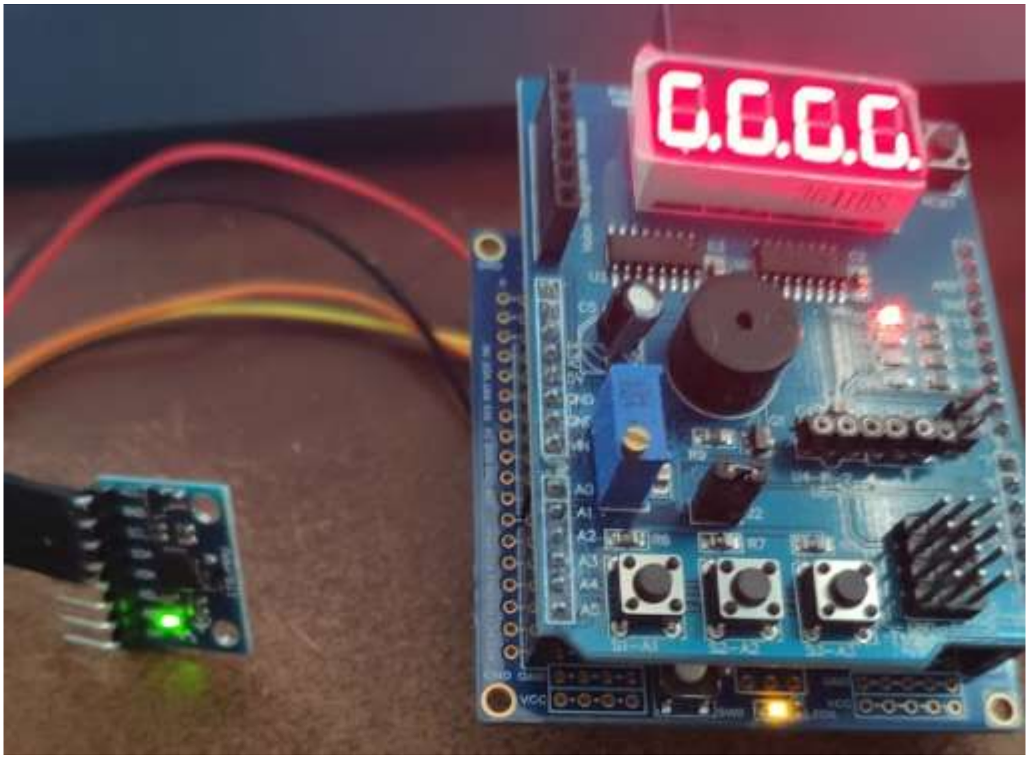


4. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)



```
Gx: -150.12, Gy: -232.13, Gz: -356.65,

Ax: -0.99, Ay: -2.00, Az: -0.99,
Gx: 915.67, Gy: 381.22, Gz: 391.46,

Ax: -2.00, Ay: 2.00, Az: -2.00,
Gx: -485.98, Gy: -266.77, Gz: -269.82,

Ax: -0.58, Ay: 0.18, Az: -0.58,
Gx: 16.59, Gy: 53.96, Gz: 159.94,

Ax: -0.96, Ay: -0.02, Az: -0.96,
Gx: -3.05, Gy: 1.65, Gz: 1.59,

Ax: -0.97, Ay: -0.04, Az: -0.97,
Gx: -3.29, Gy: 1.65, Gz: 1.46,
```

5. SCREENSHOT OF EACH DEMO (BOARD SETUP)



6. VIDEO LINKS OF EACH DEMO

https://youtu.be/94g8JURskBE

7. GITHUB LINK OF THIS DA

https://github.com/recrio/submissions/tree/master/DesignAssignments/DA6

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

“*This assignment submission is my own, original work*”.
Ron Joshua Recrio