**CPE301 – SPRING 2019**

Design Assignment 6

Student Name: Prachi Patel

Student #: 5002380222

Student Email: patelp3@unlv.nevada.edu

Primary Github address: <https://github.com/prachi173>

Directory: <https://github.com/prachi173/da_sp18>

Submit the following for all Labs:

* In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
* 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.
* If multiple asm/c files or other libraries are used, create a folder LabXX-TYY and place these files inside the folder.
* 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).
* **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

Atmega328p Xplained Mini

MPU6050



* **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A**

#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 "libraries\MPU6050\_def.h" /\* Include MPU6050 register define file \*/

#include "libraries\i2c\_master.h" /\* Include i2c Master header file \*/

#include "libraries\i2c\_master.c"

#include "libraries\uart.h" /\* Include USART header file \*/

#include "libraries\uart.c"

#define MPU6050\_WRITE 0xD0

#define MPU6050\_read 0xD1

float Acc\_x, Acc\_y, Acc\_z, Gyro\_x, Gyro\_y, Gyro\_z; // Values for acceleration and position

int main(void){

char buffer[20], float\_[10];

float Xa, Ya, Za;

float Xg, Yg, Zg;

init\_MPU6050(); //initialize MPU6050

init\_uart(9600); //initialize UART

i2c\_init(); //initialize i2c

while(1){

start\_read();

//divide raw value by sensitivity scale to get real values

Xa = Acc\_x/16384.0;

Ya = Acc\_y/16384.0;

Za = Acc\_z/16384.0;

Xg = Gyro\_x/16.4;

Yg = Gyro\_y/16.4;

Zg = Gyro\_z/16.4;

// take values in buffer to send all parameters over USART

*dtostrf*( Xa, 3, 2, float\_ );

*sprintf*(buffer,"Xa: %s ",float\_);

USART\_SendString(buffer);

*dtostrf*( Ya, 3, 2, float\_ );

*sprintf*(buffer,"Ya: %s ",float\_);

USART\_SendString(buffer);

*dtostrf*( Za, 3, 2, float\_ );

*sprintf*(buffer,"Za: %s ",float\_);

USART\_SendString(buffer);

*dtostrf*( Xg, 3, 2, float\_ );

*sprintf*(buffer,"Xg: %s ",float\_);

USART\_SendString(buffer);

*dtostrf*( Yg, 3, 2, float\_ );

*sprintf*(buffer,"Yg: %s ",float\_);

USART\_SendString(buffer);

*dtostrf*( Zg, 3, 2, float\_ );

*sprintf*(buffer,"Zg: %s \r\n ",float\_);

USART\_SendString(buffer);

*\_delay\_ms*(1000);

}

return 0;

}

void init\_uart(*uint16\_t* baudrate){

//uart register setup function

*uint16\_t* UBRR\_val = (F\_CPU/16)/(baudrate-1);

UBRR0H = UBRR\_val >> 8;

UBRR0L = UBRR\_val;

UCSR0B |= (1<<TXEN0) | (1<<RXEN0);

UCSR0C |= (1<<USBS0) | (3<<UCSZ00);

}

void uart\_outchar(unsigned char c){

//output uart data

while(!(UCSR0A & (1<<UDRE0))); // wait until sending is possible

UDR0 = c; // output character saved in c

}

void uart\_outstring(char \*s){

while(\*s){

uart\_outchar(\*s);

s++;

}

}

void init\_MPU6050(void){

//gyro function

*\_delay\_ms*(150); //power up time is > 100ms

i2c\_start(MPU6050\_WRITE);

i2c\_write(SMPLRT\_DIV); // write sample rate of 1khz

i2c\_write(0x07); // sample rate = gyroscope output rate / (1 + SMPLRT\_DIV)

i2c\_stop();

i2c\_start(MPU6050\_WRITE);

i2c\_write(PWR\_MGMT\_1);

i2c\_write(0x01); //X axis gyroscope reference freq

i2c\_stop();

i2c\_start(MPU6050\_WRITE);

i2c\_write(CONFIG); //write to configuration register

i2c\_write(0x00); //fs is 8kHz

i2c\_stop();

i2c\_start(MPU6050\_WRITE);

i2c\_write(GYRO\_CONFIG);

i2c\_write(0x18); //gyro configuration refister set range to +- 2000 deg C

i2c\_stop();

i2c\_start(MPU6050\_WRITE);

i2c\_write(INT\_ENABLE); //write enable register

i2c\_write(0x01);

i2c\_stop();

}

void start\_read(void){

//reading function

i2c\_start(MPU6050\_WRITE);

i2c\_write(ACCEL\_XOUT\_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());

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();

}

* **SCHEMATICS**

Schematic of MPU6050 given in lecture.



* **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

Task output plus extra credit.



* **SCREENSHOT OF EACH DEMO (BOARD SETUP)**

A circuit board

Description automatically generated

* **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/O3aRyoaPeIY>

* **GITHUB LINK OF THIS DA**

<https://github.com/prachi173/da_sp18/tree/master/Design%20Assignments/DA6>

**Student Academic Misconduct Policy**

<http://studentconduct.unlv.edu/misconduct/policy.html>

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

PRACHI PATEL