CPE301 – SPRING 2019

Design Assignment 6

Student Name: Tian Chen

Student #: 2001086575

Student Email: chent5@unlv.nevada.edu

Primary Github address: <https://github.com/Chentian12138/AAAABBBBB>

Directory: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**

Atmega 328p and mpu 6050

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

/\*

\* DA 6.c

\*

\* Created: 5/5/2019 11:46:25 PM

\* Author : chent

\*/

#define *F\_CPU* 16000000UL // Board running at 16MHz

#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 necessary libraries

#include "USART\_RS232\_H\_file.h" //

double acc\_X; //

double acc\_Y; //

double acc\_Z; // Declare all variables for Accelerometer and Gyroscope

double gyro\_X; //

double gyro\_Y; //

double gyro\_Z; //

void MPU6050\_Init() // Gyro initialization function

{

*\_delay\_ms*(150); // power up time

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); // start with device write address

I2C\_Write(PWR\_MGMT\_1); // write to power management register

I2C\_Write(0x01); // Reference frequency of X axis gyroscope

I2C\_Stop();

I2C\_Start\_Wait(0xD0); // start with device write address

I2C\_Write(CONFIG); // write to configuration register

I2C\_Write(0x00); // Fs = 8KHz

I2C\_Stop();

I2C\_Start\_Wait(0xD0); // start with device write address

I2C\_Write(GYRO\_CONFIG); // write to Gyro configuration register

I2C\_Write(0x18); // Celsius +/- 2000 degrees

I2C\_Stop();

I2C\_Start\_Wait(0xD0); // start with device write address

I2C\_Write(INT\_ENABLE); // rite 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(); //

accelerometer\_X = (((int)I2C\_Read\_Ack()<<8) | (int)I2C\_Read\_Ack()); //

accelerometer\_Y = (((int)I2C\_Read\_Ack()<<8) | (int)I2C\_Read\_Ack()); //

accelerometer\_Z = (((int)I2C\_Read\_Ack()<<8) | (int)I2C\_Read\_Ack()); // read value of Gyroscope

gyroscope\_X = (((int)I2C\_Read\_Ack()<<8) | (int)I2C\_Read\_Ack()); //

gyroscope\_Y = (((int)I2C\_Read\_Ack()<<8) | (int)I2C\_Read\_Ack()); //

gyroscope\_Z = (((int)I2C\_Read\_Ack()<<8) | (int)I2C\_Read\_Nack()); //

I2C\_Stop();

}

int main()

{

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

float Xa,Ya,Za;

float Xg=0,Yg=0,Zg=0;

I2C\_Init(); // initialize I2C

MPU6050\_Init(); // initialize MPU6050

USART\_Init(9600); // initialize USART with 9600 baud rate

while(1)

{

Read\_RawValue();

Xa = acc\_X/16384.0; // To obtain real values, divide raw value by scale factor

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;

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

*sprintf*(buffer," Ax = %s g\t",float\_);

USART\_SendString(buffer);

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

*sprintf*(buffer," Ay = %s g\t",float\_);

USART\_SendString(buffer);

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

*sprintf*(buffer," Az = %s g\t",float\_);

USART\_SendString(buffer);

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

*sprintf*(buffer," Gx = %s%c/s\t",float\_,0xF8);

USART\_SendString(buffer);

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

*sprintf*(buffer," Gy = %s%c/s\t",float\_,0xF8);

USART\_SendString(buffer);

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

*sprintf*(buffer," Gz = %s%c/s\r\n",float\_,0xF8);

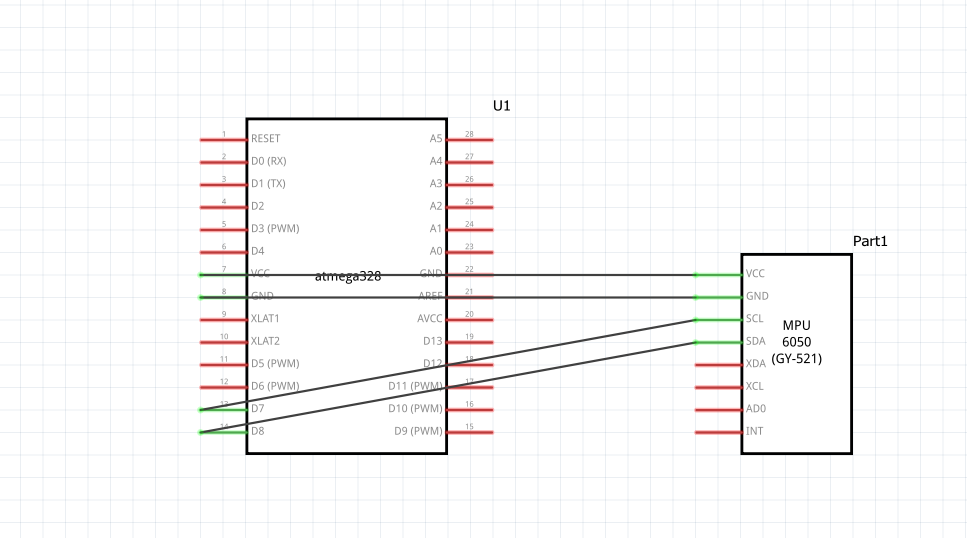
USART\_SendString(buffer);

*\_delay\_ms*(1000);

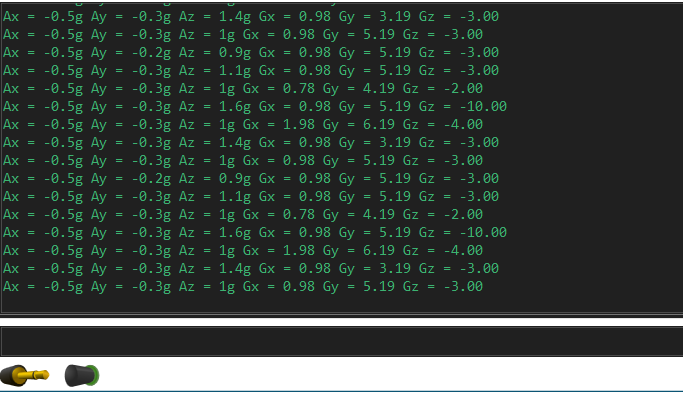
}

}

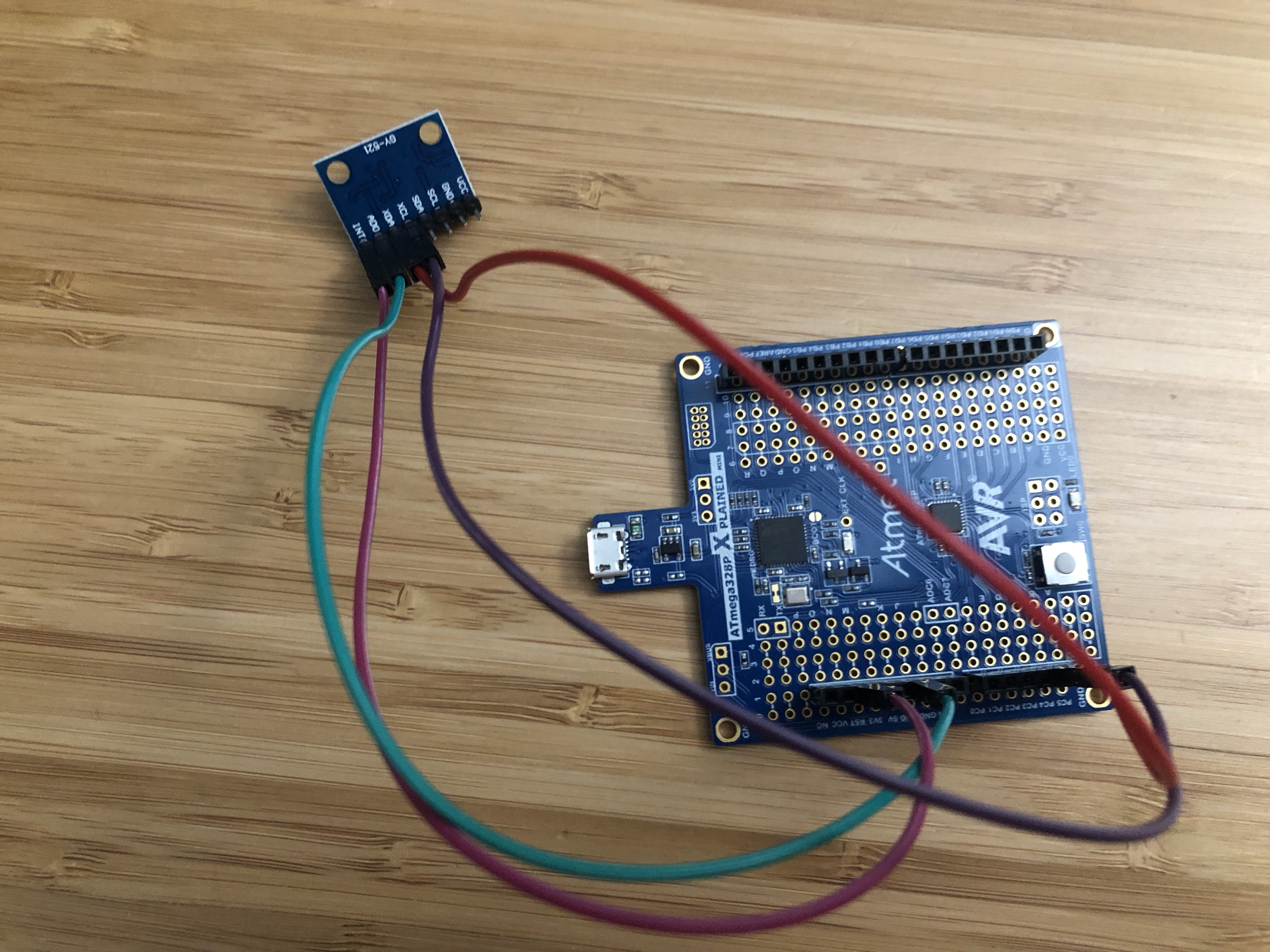
1. **SCHEMATICS**



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



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



1. **GITHUB LINK OF THIS DA**

<https://github.com/Chentian12138/AAAABBBBB>

**Student Academic Misconduct Policy**

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

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

NAME OF THE STUDENT