CPE301 – SPRING 2019

Design Assignment 6

Student Name: James Skelly

Student #: 2000945485

Student Email: skellj1@unlv.nevada.edu

Primary Github address: <https://github.com/skellj1/submission_da>

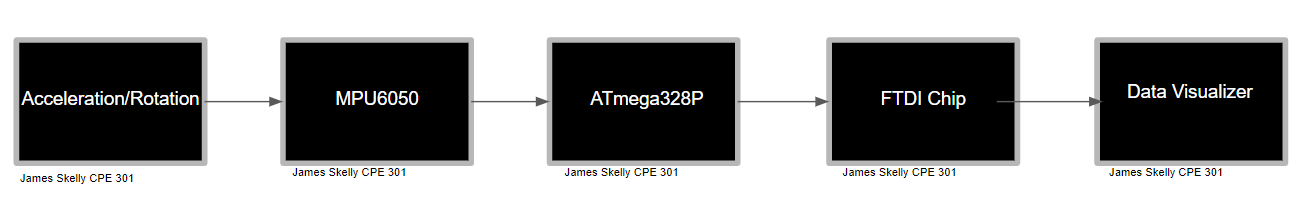
Directory: skellj1/submission\_da

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).
5. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

Components used for this DA include breadboard, Xplained Mini, jumper wire, MPU6050 Accelerometer/Gyroscope module, FTDI Chip (USART), iPhone for recording, USB cable, Atmel Studio 7.

A block diagram is shown below describing the flow of data in this DA.



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

/\*

\* DA6.c

\*

\* Created: 5/4/2019 8:06:22 PM

\* Author : J. Skelly

\*/

#define F\_CPU 16000000UL // Define CPU clock Frequency

// include necessary headers for operation of program

#include <avr/io.h>

#include <util/delay.h>

#include <inttypes.h>

#include <stdlib.h>

#include <stdio.h>

#include "MPU6050\_res\_define.h"

#include "I2C\_MasterH.h"

#include "USART\_RS232\_H.h"

// Function prototyping

void scale\_values(void);

void send\_values(void);

// initializing variables

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

float Ax,Ay,Az,t;

float Gx=0,Gy=0,Gz=0;

float Acc\_x,Acc\_y,Acc\_z;

float Temperature;

float Gyro\_x,Gyro\_y,Gyro\_z;

int main()

{

I2C\_Init(); // I2C init function call

MPU6050\_Init(); // Initialize MPU6050

USART\_Init(9600); // Initialize USART, BAUD RATE = 9600

while(1)

{

Read\_RawValue(); // call function to read raw values

scale\_values(); // call function to scale values into proper units

send\_values(); // send values to USART

*\_delay\_ms*(1000);

}

}

void Read\_RawValue() // Read values from gyro, wait for acknowledgement

{

MPU\_Start\_Loc();

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();

}

void scale\_values()

{

// Scale accelerometer values into G force units

Ax = Acc\_x/16384.0;

Ay = Acc\_y/16384.0;

Az = Acc\_z/16384.0;

// Scale gyroscope values into degrees/second units

Gx = Gyro\_x/16.4;

Gy = Gyro\_y/16.4;

Gz = Gyro\_z/16.4;

}

void send\_values()

{

// Send values to UART, formatting, unit outputs

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

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

USART\_SendString(buffer);

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

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

USART\_SendString(buffer);

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

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

USART\_SendString(buffer);

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

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

USART\_SendString(buffer);

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

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

USART\_SendString(buffer);

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

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

USART\_SendString(buffer);

}

void MPU6050\_Init() // Gyro initialization function

{

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

I2C\_Start\_Wait(0xD0); // Start at device that will be written to address

I2C\_Write(SMPLRT\_DIV); // Write to sample rate register

I2C\_Write(0x07); // set 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

I2C\_Stop();

I2C\_Start\_Wait(0xD0);

I2C\_Write(GYRO\_CONFIG); // Write to Gyroscope config. 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

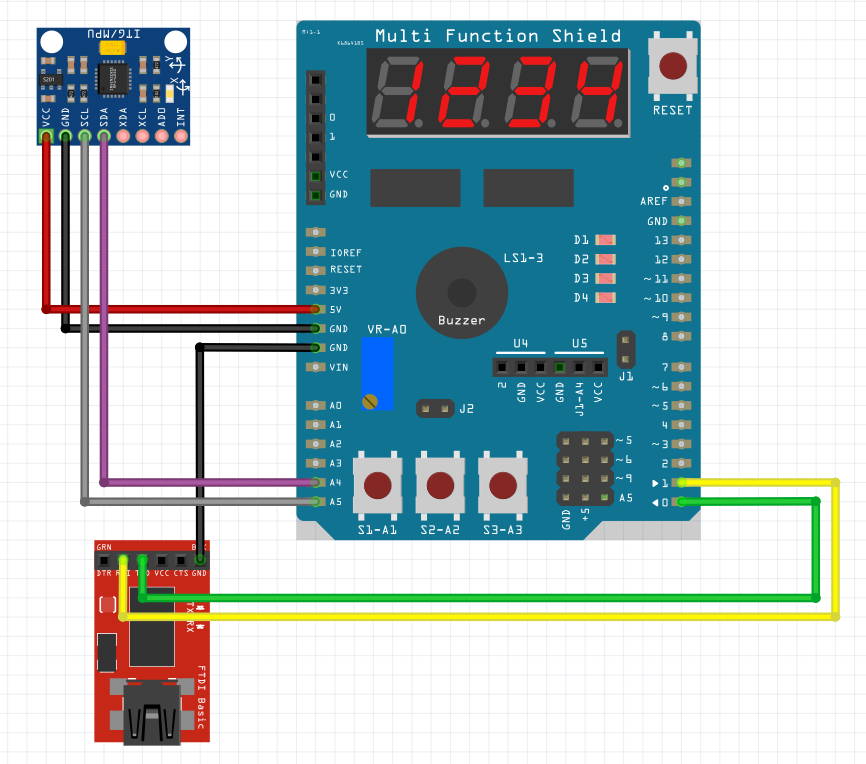
}

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

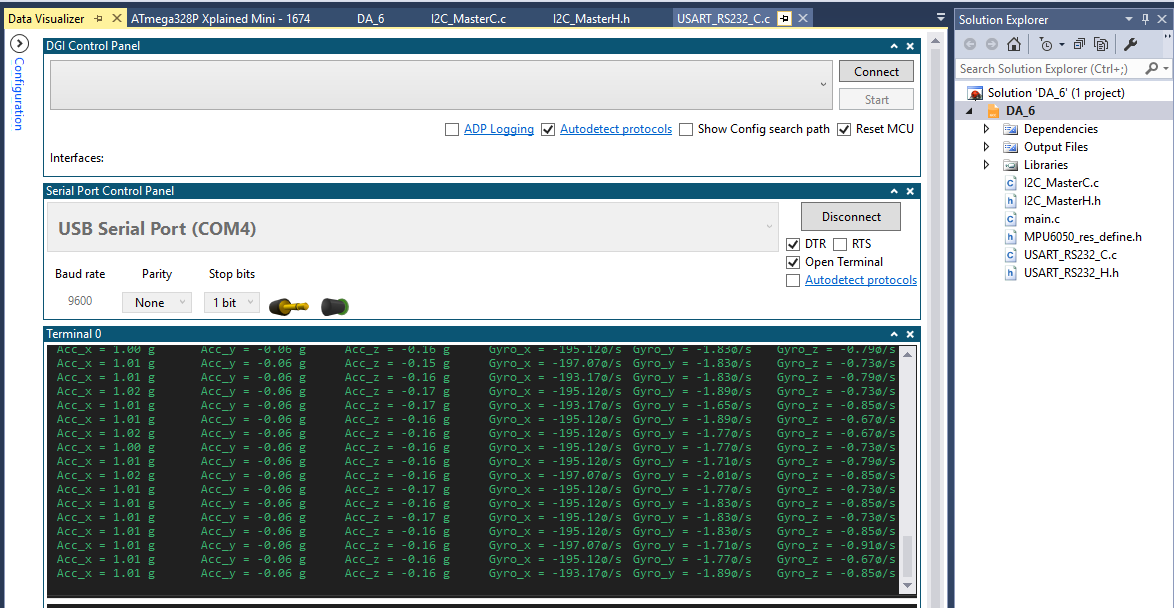
Not applicable for this assignment.

1. **SCHEMATICS**

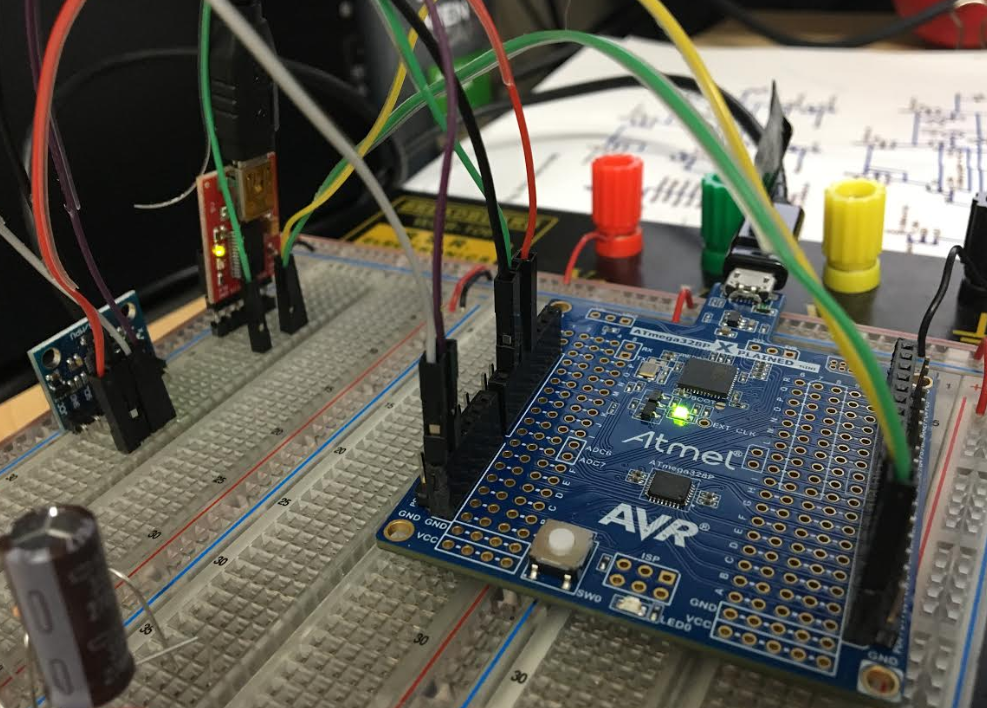
The below schematic was generated using Fritzing’s Breadboard Schematic Creator.



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



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



1. **VIDEO LINKS OF EACH DEMO**

<https://www.youtube.com/watch?v=ROYeCyrbeLc>

1. **GITHUB LINK OF THIS DA**

<https://github.com/skellj1/submission_da>

**Student Academic Misconduct Policy**

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

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

James W. Skelly