CPE301 – SPRING 2019

Design Assignment 6

Student Name: Allis Hierholzer

Student #: 2000160356

Student Email: hierholz@unlv.nevada.edu

Primary Github address: https://github.com/acexhp/submission\_da.git

Directory: Repository/cpe301/DesignAssignment/DA6

Task:

This DA is to be implemented in groups of two students. The goal of the assignment is to develop the above code to do the following:

1. Interface the provided MPU-6050 6-DOF IMU Sensor to the ATmega328p using the I2C interface. Using the earlier developed code for UART, display the accelerometer and gyro data to the UART Terminal. Extra credits for 1) visualizing the accelerometer and gyro values (10 points), and 2) Apply Kalman Filtering on at least one sensor data and display the filtered value.

Submission:

The following are required for successful completion of the design assignment:

a. AVR C code that has been compiled and working.

b. The C code should be well documented with explanation of every instruction.

c. A word document that contains the flow chart of the assembly code along with the snapshots of the schematics, components connected on the breadboard and screenshots.

1. **COMPONENTS LIST**

* MPU6050
* FTDI Chip
* Multi-functional Shield
* ATMEGA328P XPLAINED MINI
* ATMEL STUDIO 7.0

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

/\*

\* DA6.c

\*/

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

//Headers

#include <avr/io.h>

#include <util/delay.h>

#include <inttypes.h>

#include <stdlib.h>

#include <stdio.h>

#include "MPU6050\_define.h"

#include "I2C\_MasterH.h"

#include "USART\_RS232\_H.h"

//Variables

float acceleration\_x,acceleration\_y,acceleration\_z;

float gyro\_x,gyro\_y,gyro\_z;

int main()

{ //Initialize I2C, MPU6050, USART, Baud rate = 9600

float a\_x,a\_y,a\_z;

float X\_g=0,Y\_g=0,Z\_g=0;

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

I2C\_Init();

MPU6050\_init();

USART\_Init(9600);

while(1)

{

Read\_Raw\_Value();

//Divide values by scale factor

a\_x = acceleration\_x/16384.0;

a\_y = acceleration\_y/16384.0;

a\_z = acceleration\_z/16384.0;

X\_g = gyro\_x/16.4;

Y\_g = gyro\_y/16.4;

Z\_g = gyro\_z/16.4;

//Values sent to USART

*dtostrf*( a\_x, 3, 2, float\_ );

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

USART\_SendString(buffer);

*dtostrf*( a\_y, 3, 2, float\_ );

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

USART\_SendString(buffer);

*dtostrf*( a\_z, 3, 2, float\_ );

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

USART\_SendString(buffer);

*dtostrf*( X\_g, 3, 2, float\_ );

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

USART\_SendString(buffer);

*dtostrf*( Y\_g, 3, 2, float\_ );

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

USART\_SendString(buffer);

*dtostrf*( Z\_g, 3, 2, float\_ );

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

USART\_SendString(buffer);

*\_delay\_ms*(1000);

}

}

//Read values from gyro

void Read\_Raw\_Value()

{

MPU\_Start\_Loc();

acceleration\_x = (((int)I2C\_Read\_Ack()<<8) | (int)I2C\_Read\_Ack());

acceleration\_y = (((int)I2C\_Read\_Ack()<<8) | (int)I2C\_Read\_Ack());

acceleration\_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 MPU6050\_init() //Initialize gyroscope

{

*\_delay\_ms*(150); //Delay time

I2C\_Start\_Wait(0xD0);

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

I2C\_Write(0x07); //Set 1KHz

I2C\_Stop();

I2C\_Start\_Wait(0xD0);

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

I2C\_Write(0x01); //X-axis ref freq

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 interrupt enable register

I2C\_Write(0x01);

I2C\_Stop();

}

void MPU\_Start\_Loc()

{

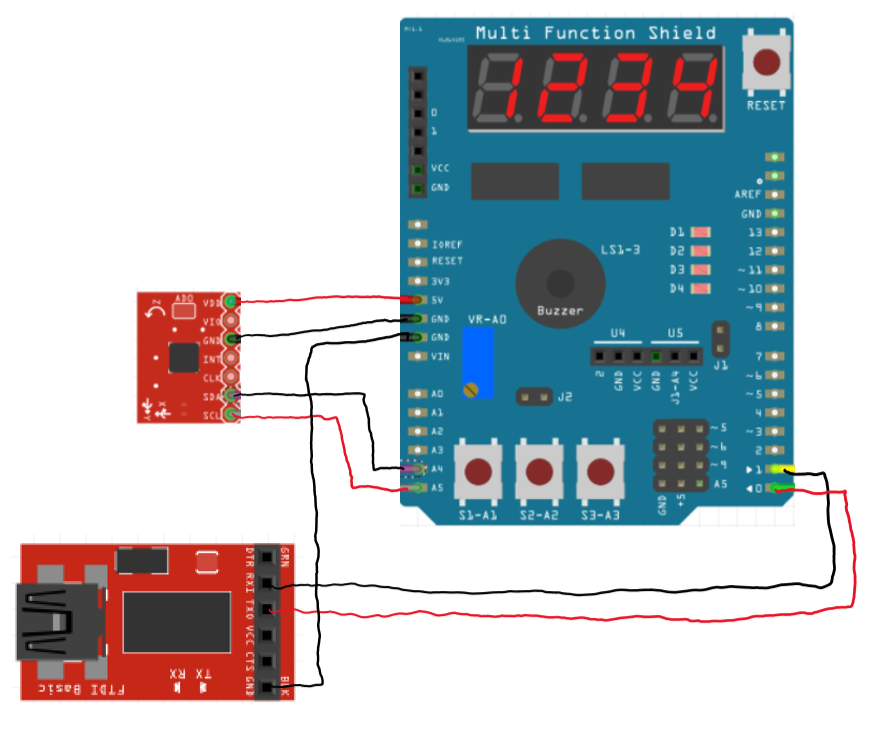
I2C\_Start\_Wait(0xD0); //I2C start with device write address

I2C\_Write(ACCEL\_XOUT\_H); //Write start location address from where to read

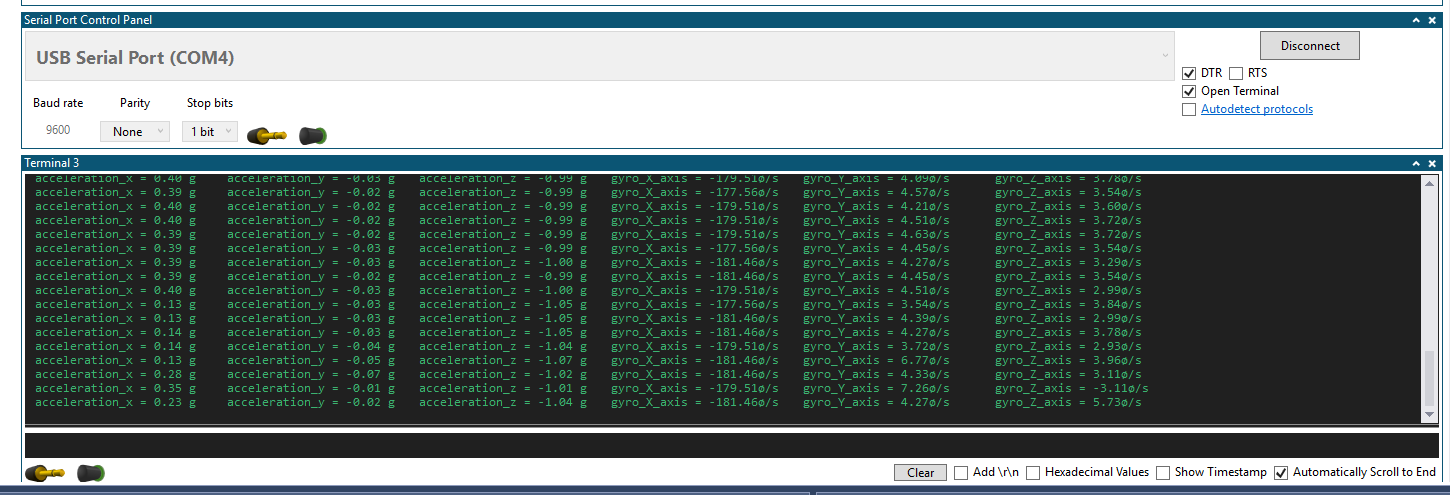
I2C\_Repeated\_Start(0xD1); //I2C start with device read address

}

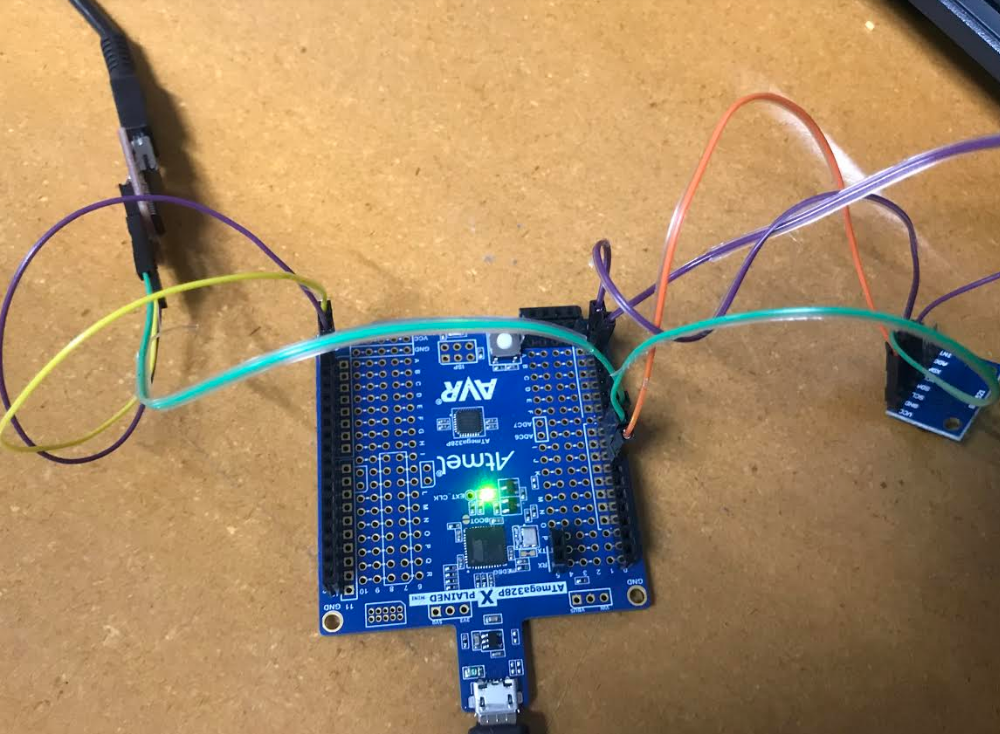
1. **SCHEMATICS**



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



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



1. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/aQQ-GUgWv4c>

1. **GITHUB LINK OF THIS DA**

<https://github.com/acexhp/submission_da.git>

**Student Academic Misconduct Policy**

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

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

Allis Hierholzer