CPE301 – SPRING 2019

Design Assignment 6A

Student Name: Mohamad Jundi

Student #: 8000321867

Student Email: jundi@unlv.nevada.edu

Primary Github address: https://github.com/MohamedJundi1994/Submission\_DA.git

Directory: Documents\School\CPE 301\Repository\CPE\_301\DesignAssignments\DA6A

1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

USB port => Xplained Mini => 5V => VCC (MPU 6050)

=> GND => GND (MPU 6050)

=> PC4 => SDA (MPU 6050)

=> PC5 => SCL (MPU 6050)

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

My main code:

#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 accelerometer\_X; //

double accelerometer\_Y; //

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

double gyroscope\_X; //

double gyroscope\_Y; //

double gyroscope\_Z; //

void MPU6050\_Init() // Gyro initialization function

{

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

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

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

I2C\_Write(0x07); // Write 1KHz sample rate

I2C\_Stop();

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

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

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

I2C\_Stop();

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

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

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

I2C\_Stop();

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

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

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

I2C\_Stop();

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

I2C\_Write(INT\_ENABLE); // Used to 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()); // Used to 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(); // Used to initialize I2C

MPU6050\_Init(); // Used to initialize MPU6050

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

while(1)

{

Read\_RawValue();

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

Ya = accelerometer\_Y/16384.0;

Za = accelerometer\_Z/16384.0;

Xg = gyroscope\_X/16.4;

Yg = gyroscope\_Y/16.4;

Zg = gyroscope\_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);

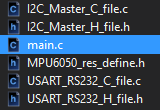
USART\_SendString(buffer);

*\_delay\_ms*(1000);

}

}

Libraries I used (included in the Github file):

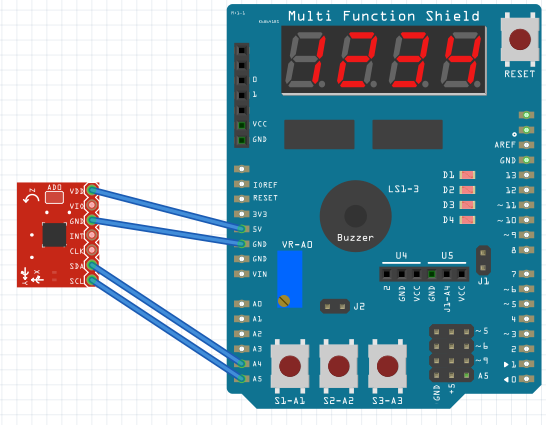


1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

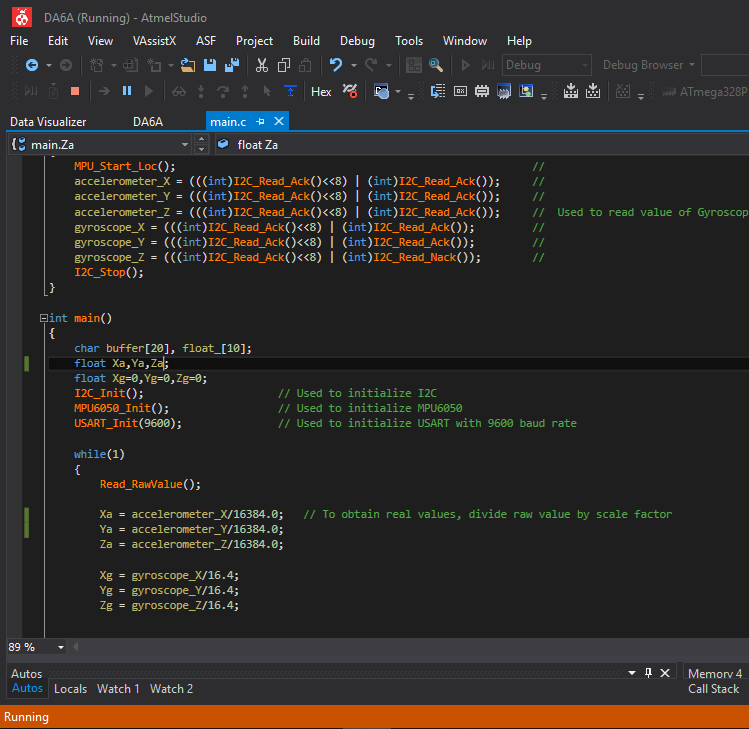
My main code, and libraries I’ve used are in number 1.

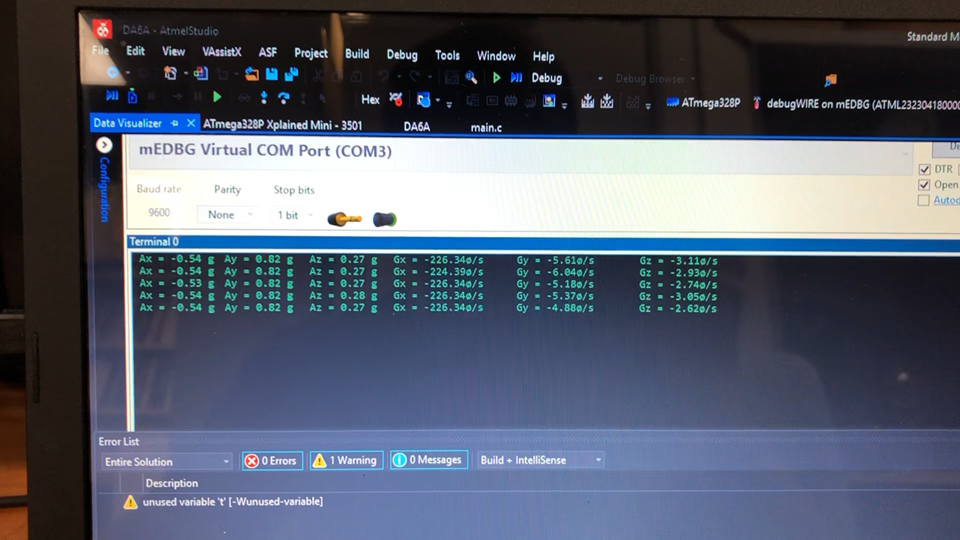
**NEXT PAGE =>**

1. **SCHEMATICS**

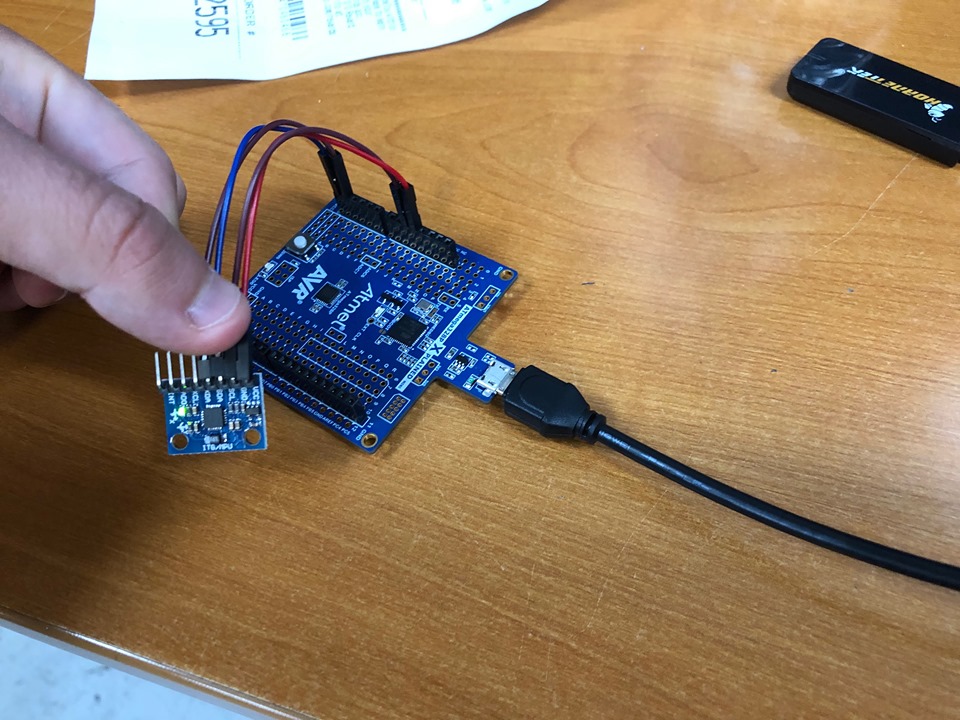


1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**





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



1. **VIDEO LINKS OF EACH DEMO**

Link: <https://www.youtube.com/watch?v=VhkeNOQqWY0&feature=share&fbclid=IwAR07Uq8vSecJnPbMjDDbKm7_mcWSBZrwz0VR0lDFYbZv_EmYQ86LxakFfs4>

1. **GITHUB LINK OF THIS DA**

Link: https://github.com/MohamedJundi1994/Submission\_DA.git

This assignment submission is my own, original work.

MOHAMAD JUNDI