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)

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

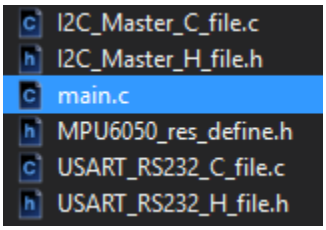
My main code:

```
#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):

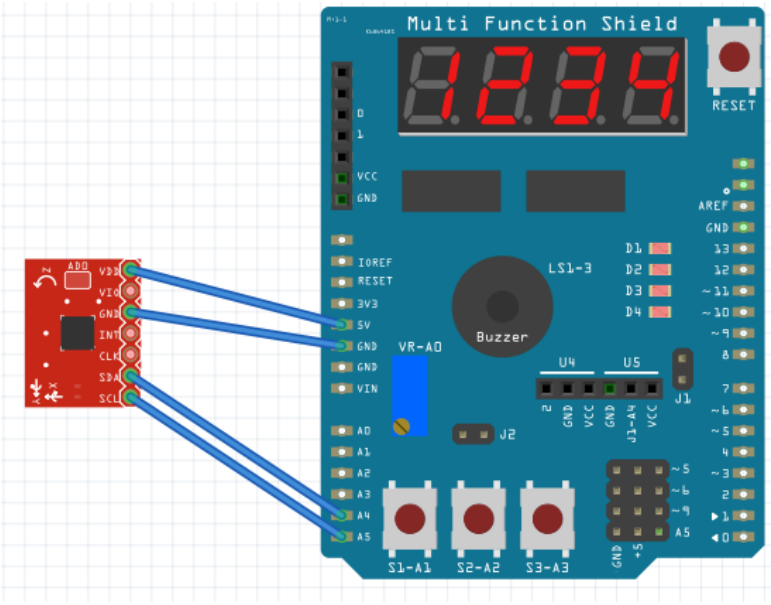


3. DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A

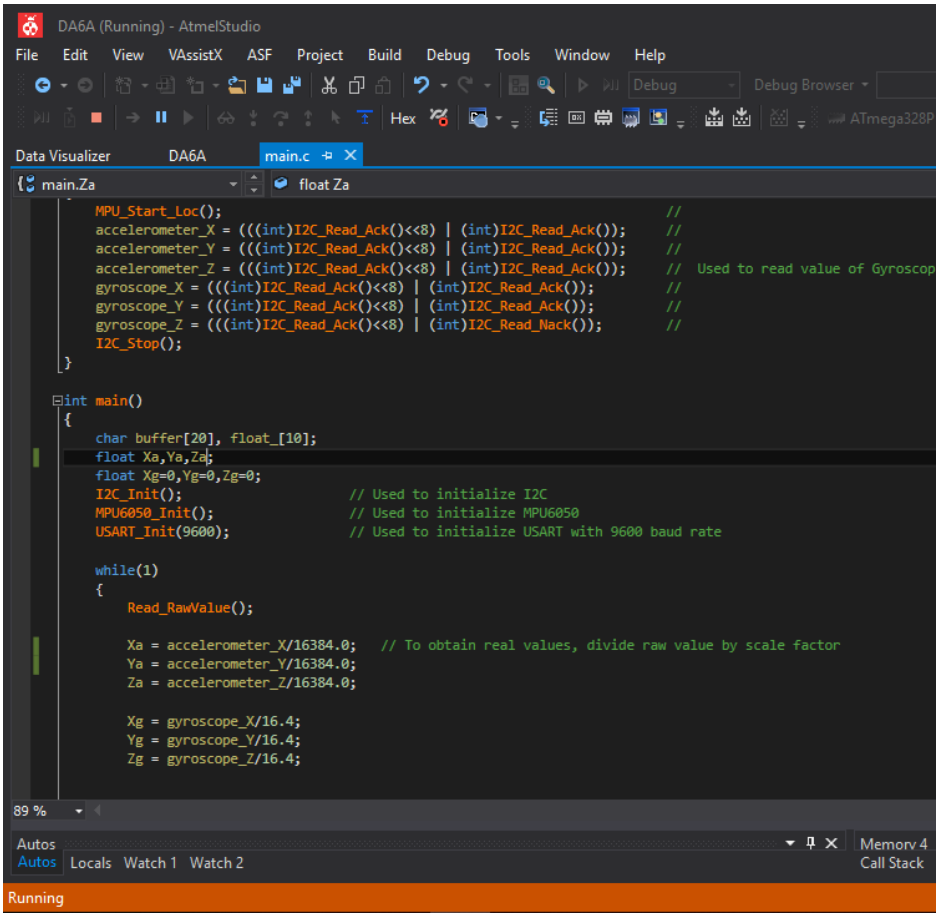
My main code, and libraries I've used are in number 1.

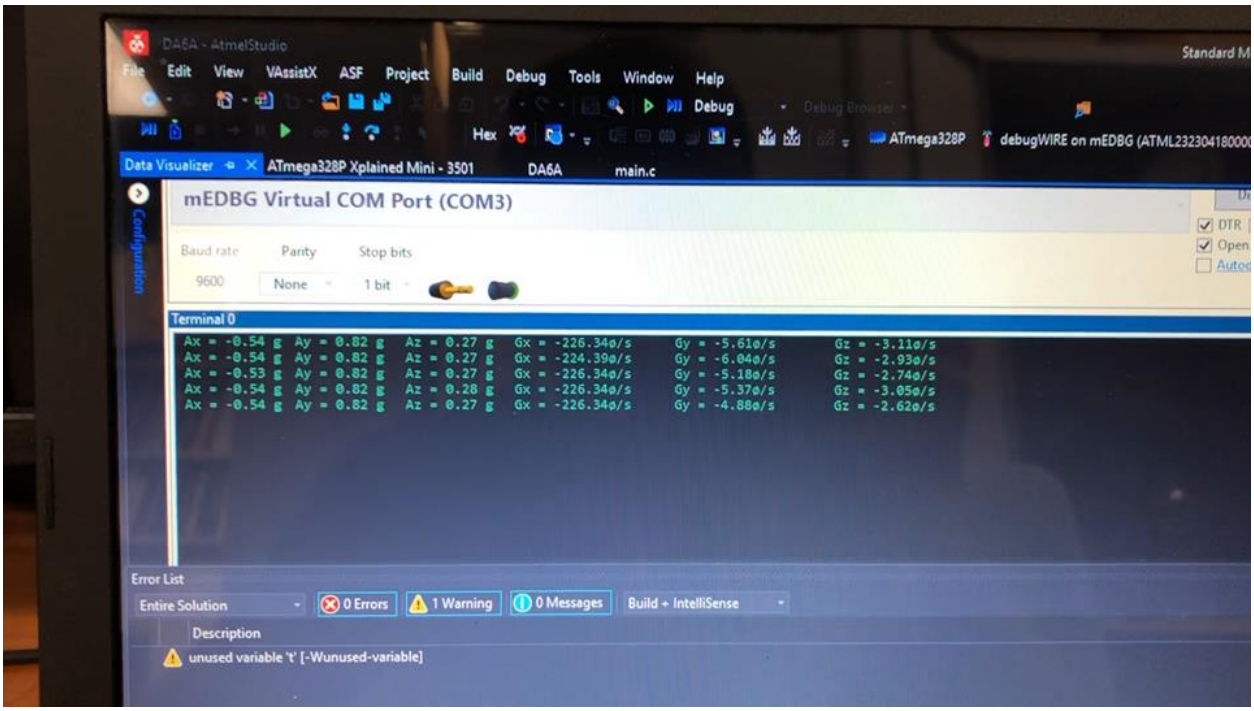
NEXT PAGE =>

4. SCHEMATICS

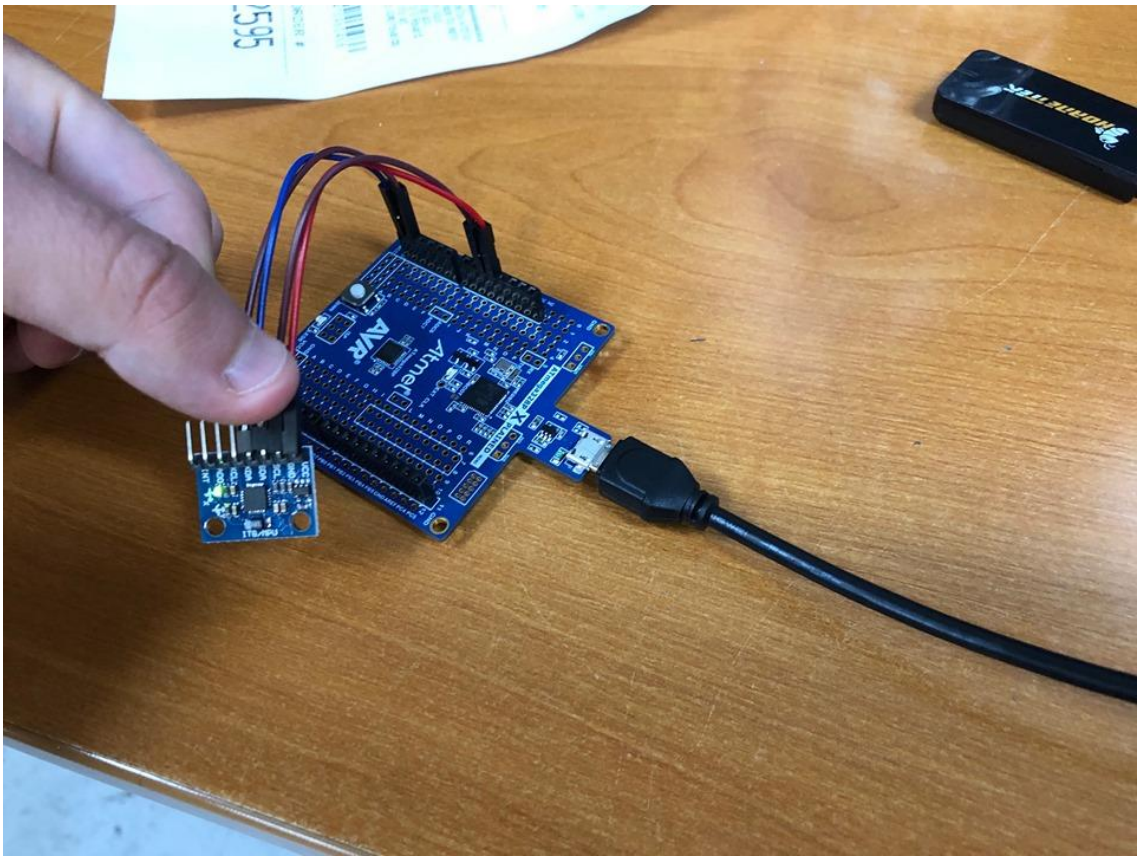


5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)





6. SCREENSHOT OF EACH DEMO (BOARD SETUP)



7. VIDEO LINKS OF EACH DEMO

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

8. GITHUB LINK OF THIS DA

Link: https://github.com/MohamedJundi1994/Submission_DA.git

This assignment submission is my own, original work.
MOHAMAD JUNDI