# CPE301 – SPRING 2019

# DESIGN ASSIGNMENT 6

Student Name: Serak Gebremedhin

Student #: 2000862766

Student Email: gebremed@unlv.nevada.edu

Primary Github address:  https://github.com/Ber-geb/effective-octo-reaction

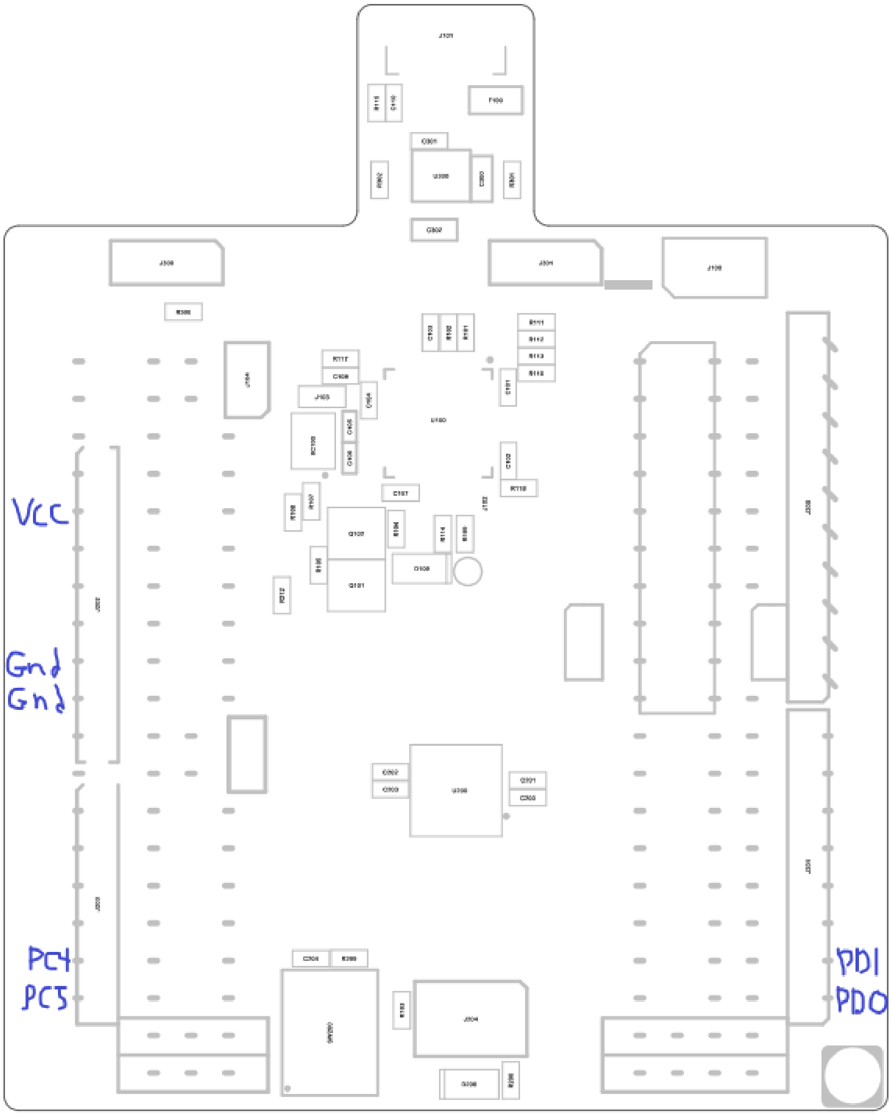
Directory: effective-octo-reaction/DesignAssignments/DA6/

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).

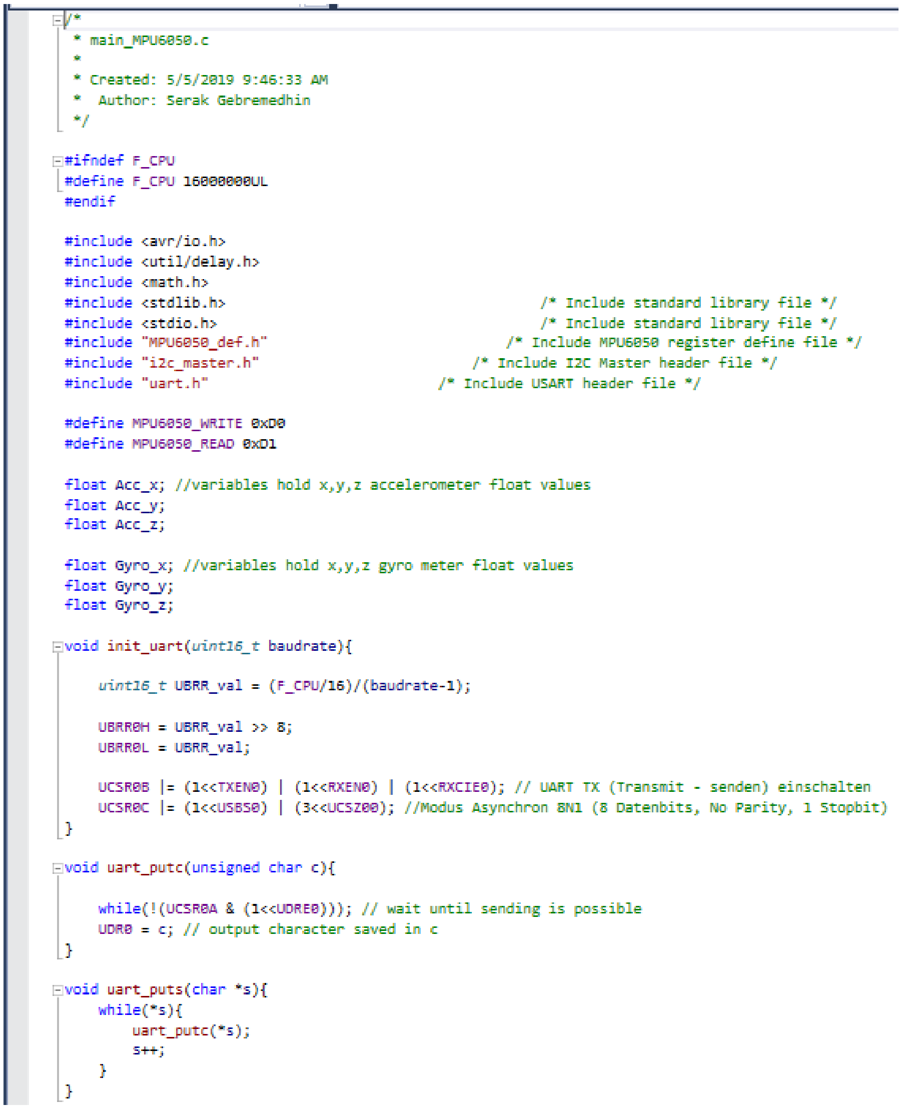
## 1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

List of Components used:
Breadboard
Atmega328P Xplained MiniBoard
MPU6050 Sensor
FTDI Chip
Block diagram with pins used in the Atmega328P:

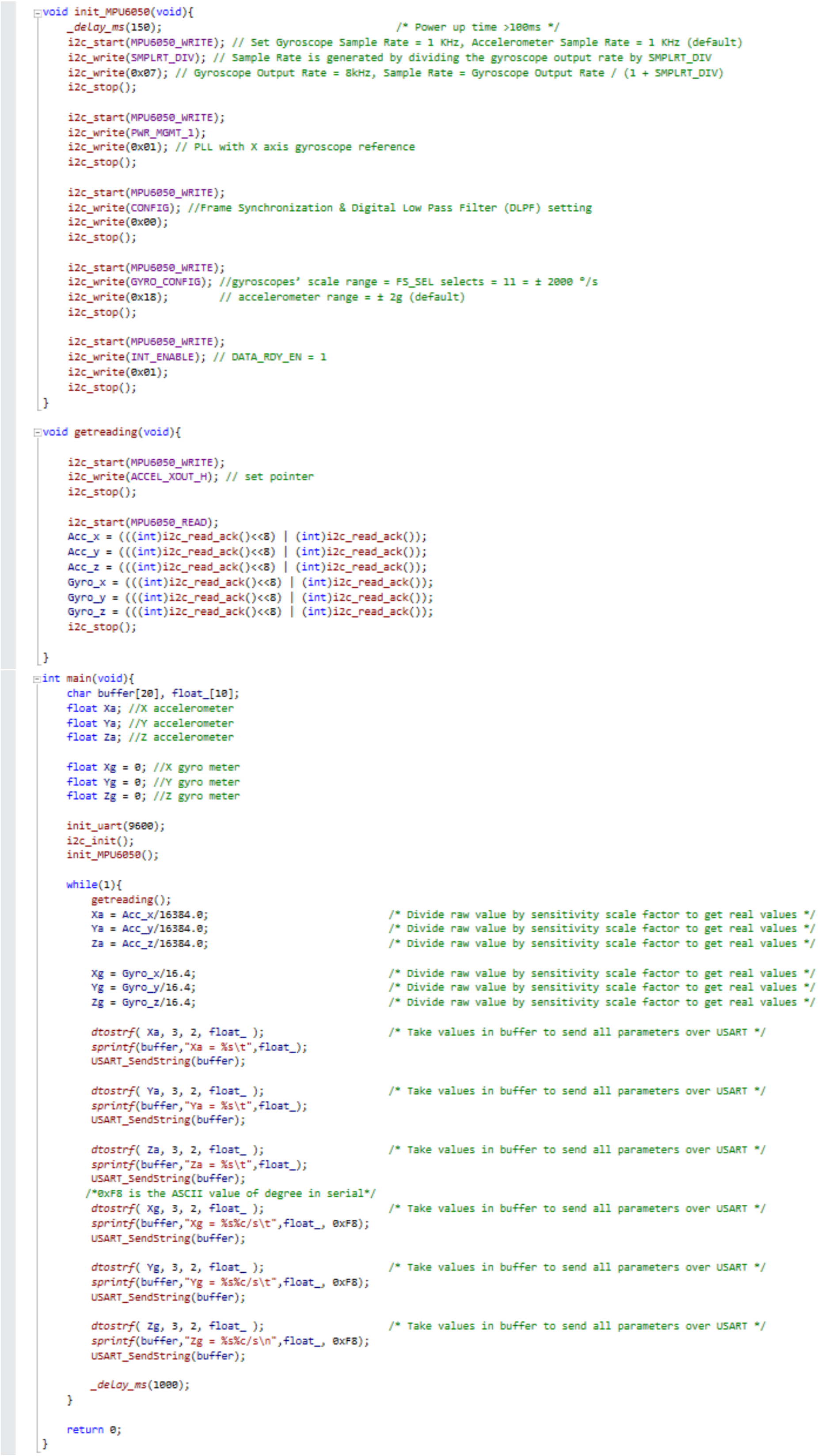


This shows the Xplained Mini Assembly Drawing. The areas of the drawing drawn in blue indicate which pins were used.

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



```c
/*
 * main_MPU6050.c
 *
 * Created: 5/5/2019 9:46:33 AM
 *  Author: Serak Gebremedhin
 */

#ifndef F_CPU
#define F_CPU 16000000UL
#endif

#include <avr/io.h>
#include <util/delay.h>
#include <math.h>
#include <stdlib.h>                                   /* Include standard library file */
#include <stdio.h>                                    /* Include standard library file */
#include "MPU6050_def.h"                          /* Include MPU6050 register define file */
#include "i2c_master.h"                       /* Include I2C Master header file */
#include "uart.h"                         /* Include USART header file */

#define MPU6050_WRITE 0xD0
#define MPU6050_READ 0xD1

float Acc_x; //variables hold x,y,z accelerometer float values
float Acc_y;
float Acc_z;

float Gyro_x; //variables hold x,y,z gyro meter float values
float Gyro_y;
float Gyro_z;

void init_uart(uint16_t baudrate){

    uint16_t UBRR_val = (F_CPU/16)/(baudrate-1);

    UBRR0H = UBRR_val >> 8;
    UBRR0L = UBRR_val;

    UCSR0B |= (1<<TXEN0) | (1<<RXEN0) | (1<<RXCIE0); // UART TX (Transmit - senden) einschalten
    UCSR0C |= (1<<USBS0) | (3<<UCSZ00); //Modus Asynchron 8N1 (8 Datenbits, No Parity, 1 Stopbit)
}

void uart_putc(unsigned char c){

    while(!(UCSR0A & (1<<UDRE0))); // wait until sending is possible
    UDR0 = c; // output character saved in c
}

void uart_puts(char *s){
    while(*s){
        uart_putc(*s);
        s++;
    }
}
```



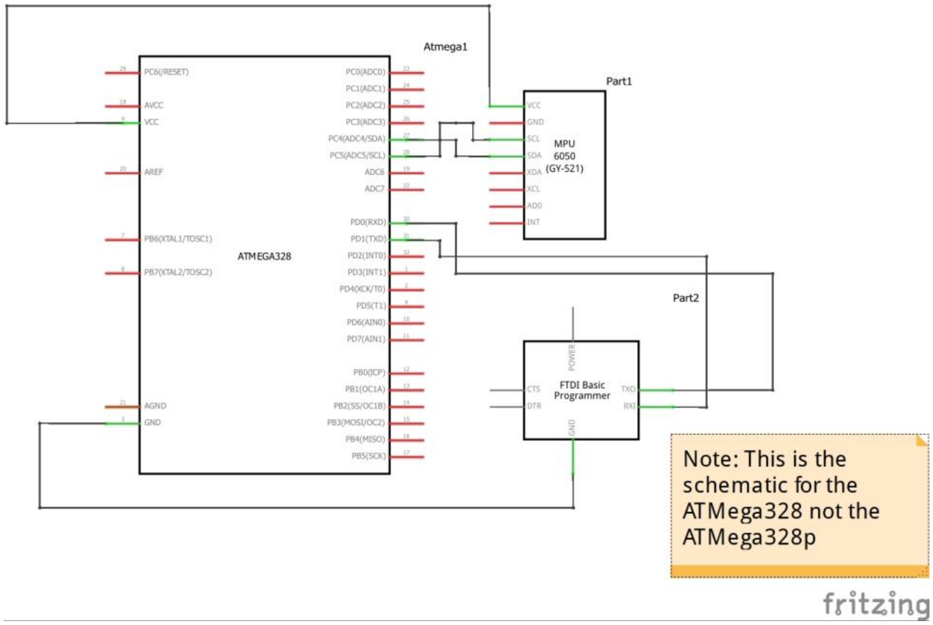
```c
void init_MPU6050(void){
    _delay_ms(150);                                     /* Power up time >100ms */
    i2c_start(MPU6050_WRITE); // Set Gyroscope Sample Rate = 1 KHz, Accelerometer Sample Rate = 1 KHz (default)
    i2c_write(SMPLRT_DIV); // Sample Rate is generated by dividing the gyroscope output rate by SMPLRT_DIV
    i2c_write(0x07); // Gyroscope Output Rate = 8kHz, Sample Rate = Gyroscope Output Rate / (1 + SMPLRT_DIV)
    i2c_stop();

    i2c_start(MPU6050_WRITE);
    i2c_write(PWR_MGMT_1);
    i2c_write(0x01); // PLL with X axis gyroscope reference
    i2c_stop();

    i2c_start(MPU6050_WRITE);
    i2c_write(CONFIG); //Frame Synchronization & Digital Low Pass Filter (DLPF) setting
    i2c_write(0x00);
    i2c_stop();

    i2c_start(MPU6050_WRITE);
    i2c_write(GYRO_CONFIG); //gyroscopes' scale range = FS_SEL selects = 11 = ± 2000 °/s
    i2c_write(0x18);        // accelerometer range = ± 2g (default)
    i2c_stop();

    i2c_start(MPU6050_WRITE);
    i2c_write(INT_ENABLE); // DATA_RDY_EN = 1
    i2c_write(0x01);
    i2c_stop();
}

void getreading(void){

    i2c_start(MPU6050_WRITE);
    i2c_write(ACCEL_XOUT_H); // set pointer
    i2c_stop();

    i2c_start(MPU6050_READ);
    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_ack());
    i2c_stop();

}
int main(void){
    char buffer[20], float_[10];
    float Xa; //X accelerometer
    float Ya; //Y accelerometer
    float Za; //Z accelerometer

    float Xg = 0; //X gyro meter
    float Yg = 0; //Y gyro meter
    float Zg = 0; //Z gyro meter

    init_uart(9600);
    i2c_init();
    init_MPU6050();

    while(1){
        getreading();
        Xa = Acc_x/16384.0;                         /* Divide raw value by sensitivity scale factor to get real values */
        Ya = Acc_y/16384.0;                         /* Divide raw value by sensitivity scale factor to get real values */
        Za = Acc_z/16384.0;                         /* Divide raw value by sensitivity scale factor to get real values */

        Xg = Gyro_x/16.4;                           /* Divide raw value by sensitivity scale factor to get real values */
        Yg = Gyro_y/16.4;                           /* Divide raw value by sensitivity scale factor to get real values */
        Zg = Gyro_z/16.4;                           /* Divide raw value by sensitivity scale factor to get real values */

        dtostrf( Xa, 3, 2, float_ );                /* Take values in buffer to send all parameters over USART */
        sprintf(buffer,"Xa = %s\t",float_);
        USART_SendString(buffer);

        dtostrf( Ya, 3, 2, float_ );                /* Take values in buffer to send all parameters over USART */
        sprintf(buffer,"Ya = %s\t",float_);
        USART_SendString(buffer);

        dtostrf( Za, 3, 2, float_ );                /* Take values in buffer to send all parameters over USART */
        sprintf(buffer,"Za = %s\t",float_);
        USART_SendString(buffer);
       /*0xF8 is the ASCII value of degree in serial*/
        dtostrf( Xg, 3, 2, float_ );                /* Take values in buffer to send all parameters over USART */
        sprintf(buffer,"Xg = %s%c/s\t",float_, 0xF8);
        USART_SendString(buffer);

        dtostrf( Yg, 3, 2, float_ );                /* Take values in buffer to send all parameters over USART */
        sprintf(buffer,"Yg = %s%c/s\t",float_, 0xF8);
        USART_SendString(buffer);

        dtostrf( Zg, 3, 2, float_ );                /* Take values in buffer to send all parameters over USART */
        sprintf(buffer,"Zg = %s%c/s\n",float_, 0xF8);
        USART_SendString(buffer);

         _delay_ms(1000);
    }

    return 0;
}
```

This shows the main_MPU6050 C code. This is the only file that was modified with the given files.

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

There is no task 2 for this design assignment.

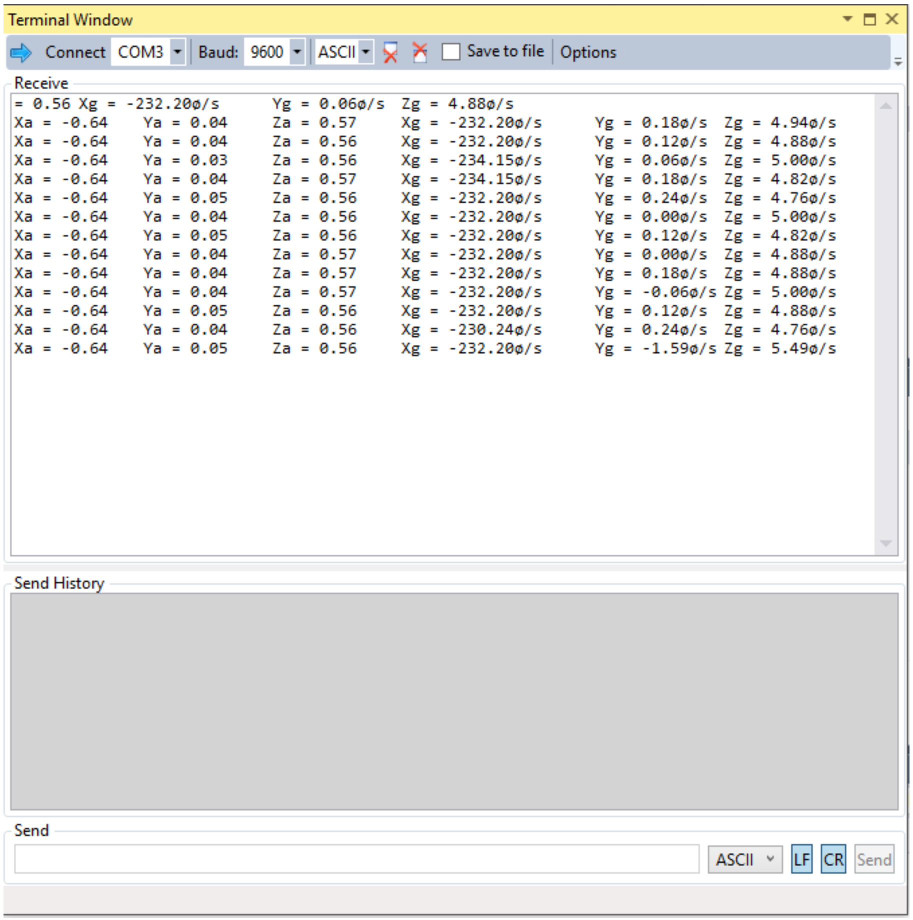
## 4. SCHEMATICS



Note: This is the schematic for the ATMega328 not the ATMega328p

This shows the schematic for this design assignment.

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

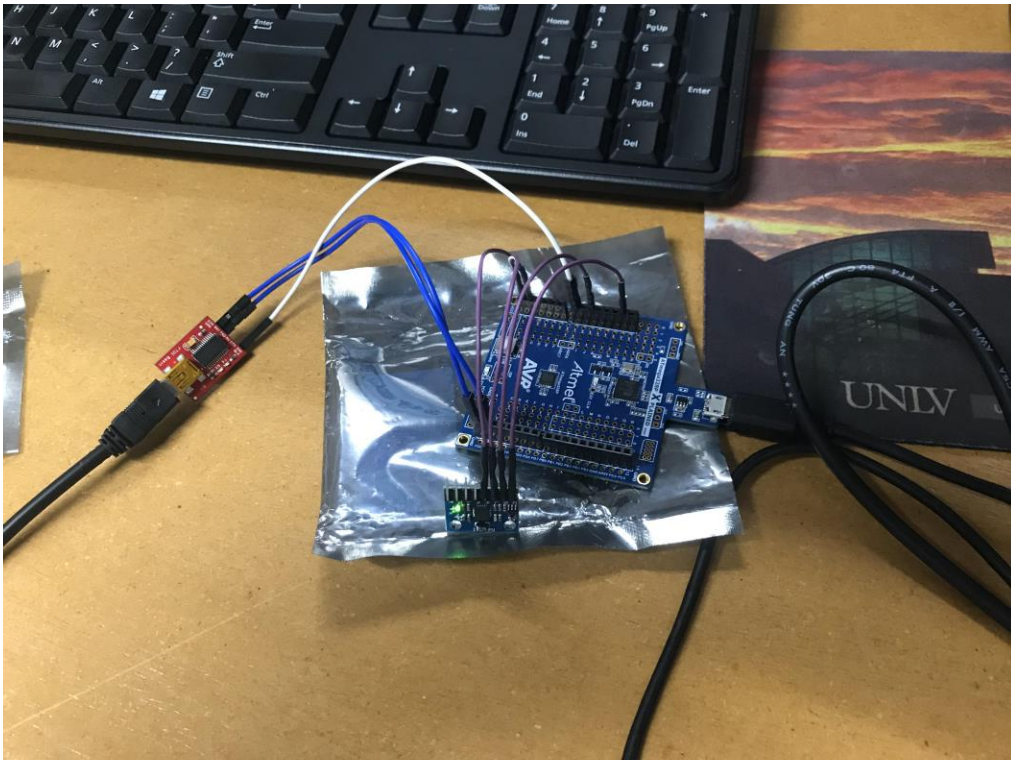


Connect COM3 Baud: 9600 ASCII □ Save to file Options

```
= 0.56 Xg = -232.20ø/s     Yg = 0.06ø/s  Zg = 4.88ø/s
Xa = -0.64    Ya = 0.04    Za = 0.57     Xg = -232.20ø/s     Yg = 0.18ø/s  Zg = 4.94ø/s
Xa = -0.64    Ya = 0.04    Za = 0.56     Xg = -232.20ø/s     Yg = 0.12ø/s  Zg = 4.88ø/s
Xa = -0.64    Ya = 0.03    Za = 0.56     Xg = -234.15ø/s     Yg = 0.06ø/s  Zg = 5.00ø/s
Xa = -0.64    Ya = 0.04    Za = 0.57     Xg = -234.15ø/s     Yg = 0.18ø/s  Zg = 4.82ø/s
Xa = -0.64    Ya = 0.05    Za = 0.56     Xg = -232.20ø/s     Yg = 0.24ø/s  Zg = 4.76ø/s
Xa = -0.64    Ya = 0.04    Za = 0.56     Xg = -232.20ø/s     Yg = 0.00ø/s  Zg = 5.00ø/s
Xa = -0.64    Ya = 0.05    Za = 0.56     Xg = -232.20ø/s     Yg = 0.12ø/s  Zg = 4.82ø/s
Xa = -0.64    Ya = 0.04    Za = 0.57     Xg = -232.20ø/s     Yg = 0.00ø/s  Zg = 4.88ø/s
Xa = -0.64    Ya = 0.04    Za = 0.57     Xg = -232.20ø/s     Yg = 0.18ø/s  Zg = 4.88ø/s
Xa = -0.64    Ya = 0.04    Za = 0.57     Xg = -232.20ø/s     Yg = -0.06ø/s Zg = 5.00ø/s
Xa = -0.64    Ya = 0.05    Za = 0.56     Xg = -232.20ø/s     Yg = 0.12ø/s  Zg = 4.88ø/s
Xa = -0.64    Ya = 0.04    Za = 0.56     Xg = -230.24ø/s     Yg = 0.24ø/s  Zg = 4.76ø/s
Xa = -0.64    Ya = 0.05    Za = 0.56     Xg = -232.20ø/s     Yg = -1.59ø/s Zg = 5.49ø/s
```

This shows the terminal during the debugging process of the program.

## 6. SCREENSHOT OF EACH DEMO (BOARD SETUP)



This shows the board setup for this design assignment. There is the MPU6050 sensor, the FTDI basic chip, and the ATMega328p Xplained Mini board.

## 7. VIDEO LINKS OF EACH DEMO

https://youtu.be/UxQbC09AzgQ

## 8. GITHUB LINK OF THIS DA

https://github.com/Ber-geb/effective-octo-reaction

**Student Academic Misconduct Policy**
http://studentconduct.unlv.edu/misconduct/policy.html

"*This assignment submission is my own, original work*".
Serak Gebremedhin