hCPE301 – SPRING 2019

Design Assignment 6

Student Name: Billy Maddex

Student #: 2000928390

Student Email: [maddex@unlv.nevada.edu](mailto:maddex@unlv.nevada.edu)

Primary Github address: <https://github.com/billymaddex/fluffy-chainsaw>

Directory: 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**

Atmega328PB-XMINI

Breadboard Arduino Module

GY-521 MPU6050

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

/\*

\* DA6.c

\*

\* Created: 12/8/2019 10:37:43 PM

\* Author : Billy

\*/

// 16MHz CPU clock

#define F\_CPU 16000000UL

// 115200 baud rate setting

#define URRR\_115200 8

// 9600 baud rate setting

#define UBRR\_9600 103

#include <avr/io.h>

#include <util/atomic.h>

#include <util/delay.h>

#include <stdio.h>

#include "i2cmaster.h"

#include "MPU6050\_def.h"

// MPU6050 read and write addresses

#define MPU6050\_WRITE 0xD0

#define MPU6050\_READ 0xD1

// terminal output function declarations

void uart\_init(unsigned int ubrr);

void uart\_send\_string(char \*data);

// MPU6050 function declarations

void init\_MPU6050(void);

float get\_reading(unsigned char data);

int main(void)

{

// configure terminal

uart\_init(URRR\_115200);

// initialize i2c

i2c\_init();

// initialize MPU6050

init\_MPU6050();

// variables to store data from MPU6050

float acl\_x, acl\_y, acl\_z;

float gyr\_x, gyr\_y, gyr\_z;

// output string for terminal display

char out\_string[200];

// loop forever collecting and formatting the data, output to terminal

while (1)

{

// output every 1 second

\_delay\_ms(1000);

// update all variables with readings from the peripheral

acl\_x = get\_reading(ACCEL\_XOUT\_H);

\_delay\_ms(200);

acl\_y = get\_reading(ACCEL\_YOUT\_H);

\_delay\_ms(200);

acl\_z = get\_reading(ACCEL\_ZOUT\_H);

\_delay\_ms(200);

gyr\_x = get\_reading(GYRO\_XOUT\_H);

\_delay\_ms(200);

gyr\_y = get\_reading(GYRO\_YOUT\_H);

\_delay\_ms(200);

gyr\_z = get\_reading(GYRO\_ZOUT\_H);

// generate output string for acl\_x and send to terminal

snprintf(out\_string, sizeof(out\_string), "Accel X:\t%.0f\r\nAccel Y:\t%.0f\r\nAccel Z:\t%.0f\r\nGyro X:\t\t%.0f\r\nGyro Y:\t\t%.0f\r\nGyro Z:\t\t%.0f\r\n\r\n", acl\_x, acl\_y, acl\_z, gyr\_x, gyr\_y, gyr\_z);

// send output string to terminal

ATOMIC\_BLOCK(ATOMIC\_FORCEON)

{

uart\_send\_string(out\_string);

}

}

}

// configure terminal output

void uart\_init(unsigned int ubrr)

{

// set buad rate to 9600

UBRR0H = (unsigned char)(ubrr>>8);

UBRR0L = (unsigned char)ubrr;

// set terminal to output

UCSR0B = (1 << TXEN0);

// set output mode

UCSR0C = (1 << UCSZ01) | (1 << UCSZ00);

}

void uart\_send\_string(char \*data)

{

while((\*data != '\0'))

{

while (!(UCSR0A & (1 <<UDRE0)));

UDR0 = \*data;

data++;

}

}

// get reading for MPU6050 register specified by pointer 'data'

float get\_reading(unsigned char data)

{

float value;

i2c\_start(MPU6050\_WRITE);

i2c\_write(data);

i2c\_stop();

i2c\_start(MPU6050\_READ);

//value = (((int)i2c\_readAck()<<8) | (int)i2c\_readAck());

int high = ((int)i2c\_readAck()<<8);

value = high | (int)i2c\_readNak();

i2c\_stop();

return value;

}

// initialize MPU6050 peripheral

void init\_MPU6050(void)

{

// power up delay

\_delay\_ms(150);

// reset all registers in the peripheral

i2c\_start(MPU6050\_WRITE);

i2c\_write(PWR\_MGMT\_1);

i2c\_write(0x80);

i2c\_stop();

// reset delay

\_delay\_ms(900);

// set sample rate divider to 8, sample rate = 8kHz

i2c\_start(MPU6050\_WRITE);

i2c\_write(SMPLRT\_DIV);

i2c\_write(0x07);

i2c\_stop();

// set clock source to internal 8MHz and disable temp sensor

i2c\_start(MPU6050\_WRITE);

i2c\_write(PWR\_MGMT\_1);

i2c\_write(0x08);

i2c\_stop();

// DLPF configured for 8kHz

i2c\_start(MPU6050\_WRITE);

i2c\_write(CONFIG);

i2c\_write(0x00);

i2c\_stop();

// gyro configuration, set to 500 degrees per second

i2c\_start(MPU6050\_WRITE);

i2c\_write(GYRO\_CONFIG);

i2c\_write(0x08);

i2c\_stop();

// accelerometer configuration, set to 8g

i2c\_start(MPU6050\_WRITE);

i2c\_write(ACCEL\_CONFIG);

i2c\_write(0x10);

i2c\_stop();

// enable the data ready interrupt

//i2c\_start(MPU6050\_WRITE);

//i2c\_write(INT\_ENABLE);

//i2c\_write(0x01);

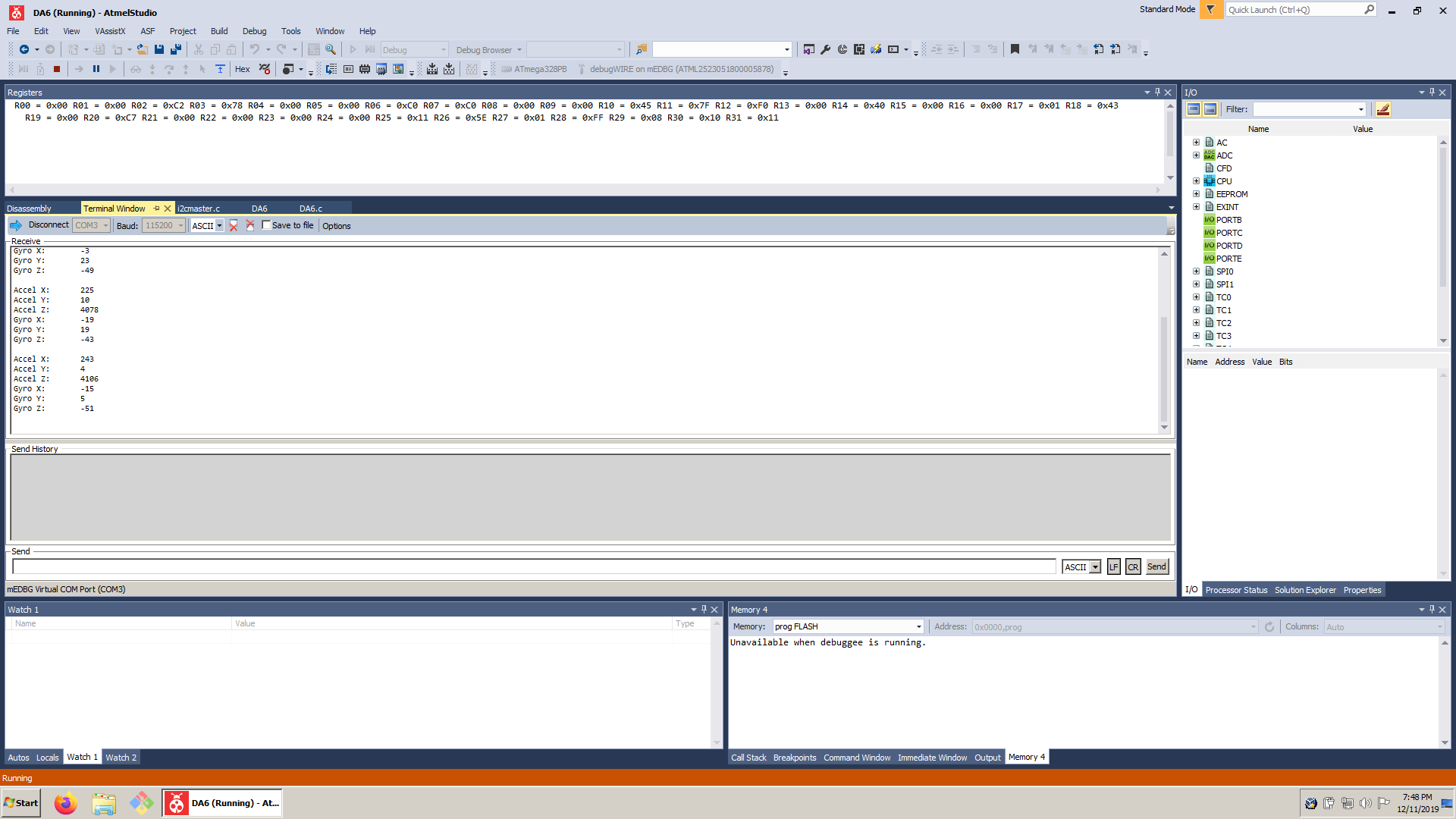
//i2c\_stop();

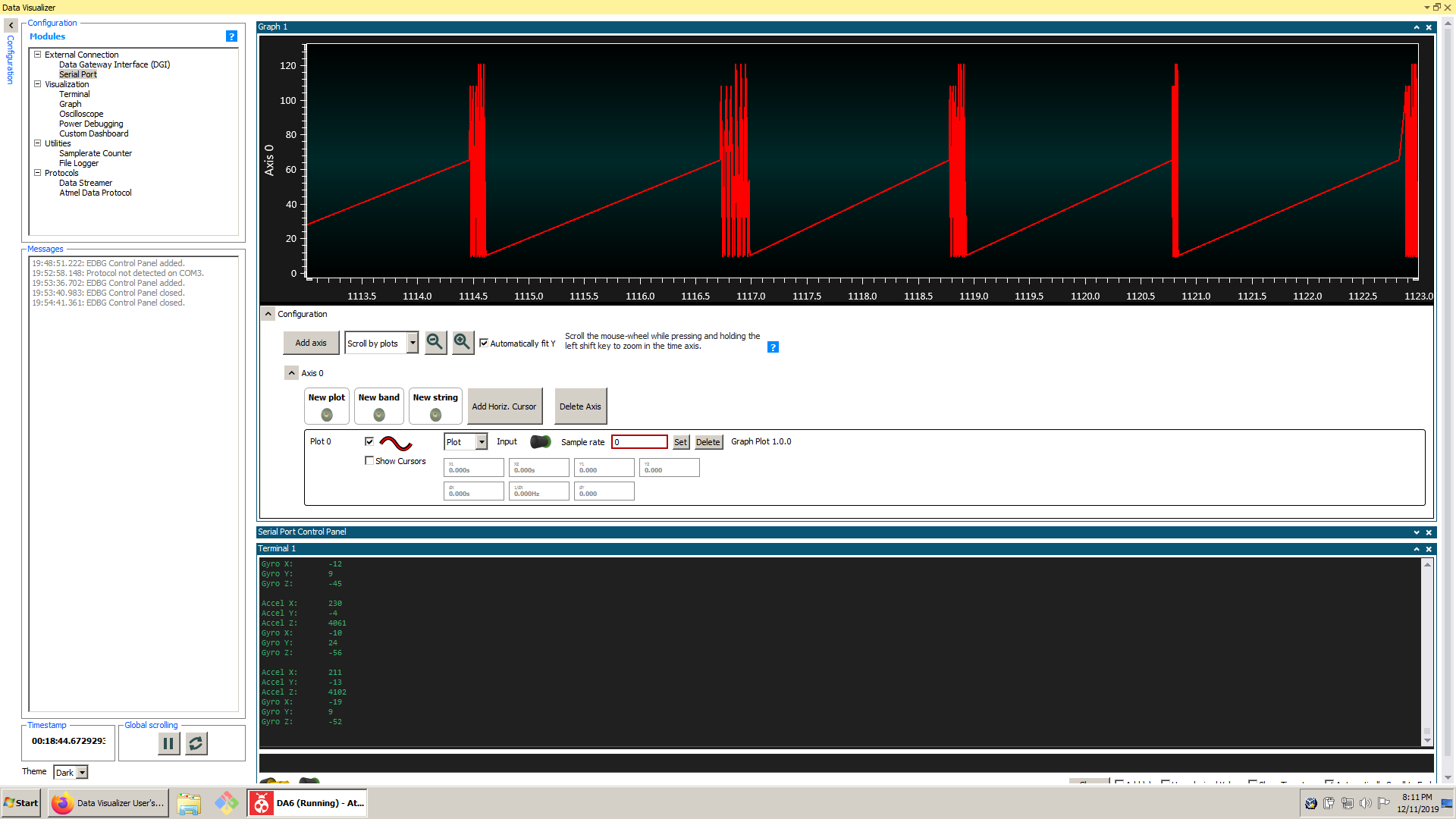
}

1. **DEVELOPED MODIFIED CODE OF TASK 2**
2. **SCHEMATICS**

Use fritzing.org

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/8ptLOUYCFF4

1. **GITHUB LINK OF THIS DA**

<https://github.com/billymaddex/fluffy-chainsaw/tree/master/DesignAssignments/DA6>

**Student Academic Misconduct Policy**

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

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

Billy Maddex