CPE301 – SPRING 2019

Design Assignment 6

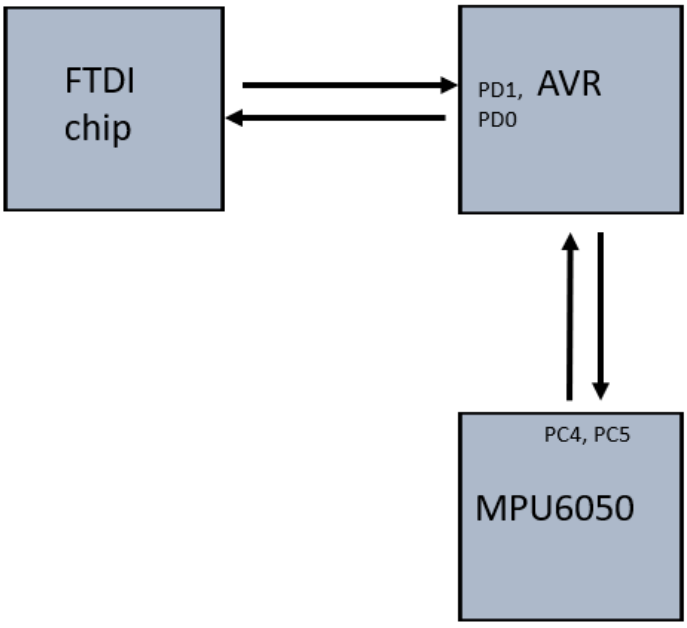
Student Name: Francisco Mata Carlos
Student #: 1012593607
Student Email: matacarl@unlv.nevada.edu
Primary Github address: https://github.com/chicosisco/da_sub.git
Directory: repository/cpe301/DesignAssignments/DA6

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

The components used for this assignment are the next:

a. Atmega328p Xplained Mini
b. Multi-functional Shield
c. Atmel Studio 7
d. FTDI chip
e. MPU6050

Block diagram with pins used



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

1. Task 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.

```
#define F_CPU 16000000UL                                                    // Define CPU
clock Frequency

// headers program operation
#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"

// initializing variables
float Acc_x,Acc_y,Acc_z;
float Gyro_x,Gyro_y,Gyro_z;

int main()
{
	float X_a,Y_a,Z_a;
	float X_g=0,Y_g=0,Z_g=0;
	char buffer[20], float_[10];
	// I2C init function call
	I2C_Init();
	// Initialize MPU6050 initialization

	Gyro_MPU6050_Init();
	// Initialize USART, BAUD RATE = 9600
	USART_Init(9600);

	while(1)
	{
		Read_Raw_Value();

		// Values divided by scale factor
		X_a = Acc_x/16384.0;
		Y_a = Acc_y/16384.0;
		Z_a = Acc_z/16384.0;

		// Values divided by scale factor
		X_g = Gyro_x/16.4;
		Y_g = Gyro_y/16.4;
		Z_g = Gyro_z/16.4;
```

```
                // send values to USART
                dtostrf( X_a, 3, 2, float_ );
                sprintf(buffer," acce_x = %s g\t",float_);
                USART_SendString(buffer);

                dtostrf( Y_a, 3, 2, float_ );
                sprintf(buffer," acce_y = %s g\t",float_);
                USART_SendString(buffer);

                dtostrf( Z_a, 3, 2, float_ );
                sprintf(buffer," acce_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);
        }
}

void Read_Raw_Value()       // Read Raw values from gyro, and wait for acknowledgment
{
        MPU_Start_Loc();
        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_Nack());
        I2C_Stop();
}

void Gyro_MPU6050_Init()                                                // Gyro
initialization function
{
        _delay_ms(150);                                                 // Power up time
>100ms
        I2C_Start_Wait(0xD0);                                   // Start at device that
will be written to address
        I2C_Write(SMPLRT_DIV);                                  // Write to sample rate
register
        I2C_Write(0x07);                                        // set 1KHz sample rate
        I2C_Stop();
        I2C_Start_Wait(0xD0);
        I2C_Write(PWR_MGMT_1);                                  // Write to power
management register
```

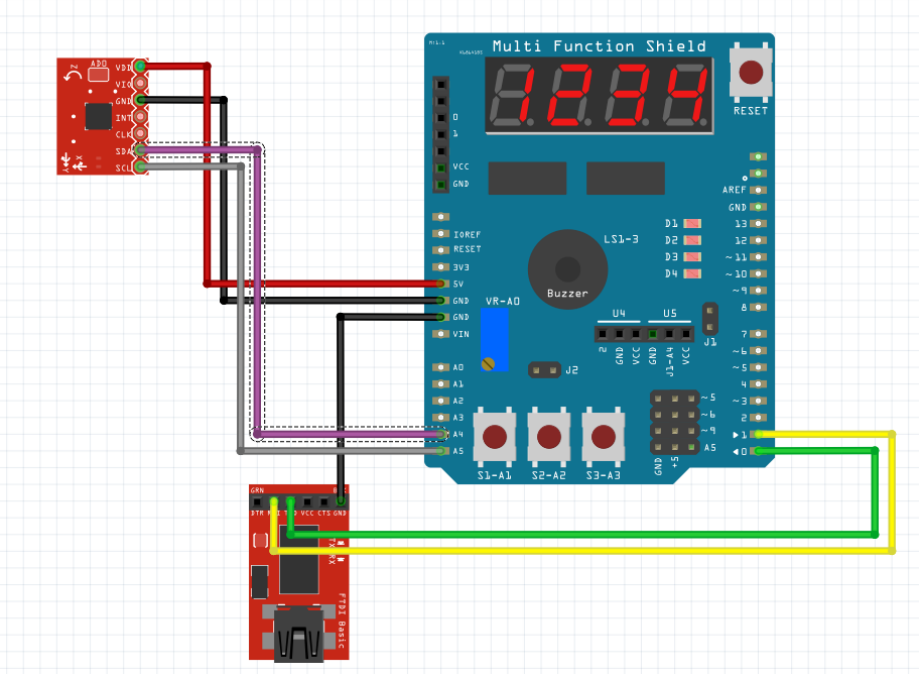
```
	I2C_Write(0x01);                                        // X axis gyroscope
reference frequency
	I2C_Stop();
	I2C_Start_Wait(0xD0);
	I2C_Write(CONFIG);                                      // Write to Configuration
register
	I2C_Write(0x00);                                        // Fs = 8KHz
	I2C_Stop();
	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 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
}
```

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

Same as above

4. SCHEMATICS

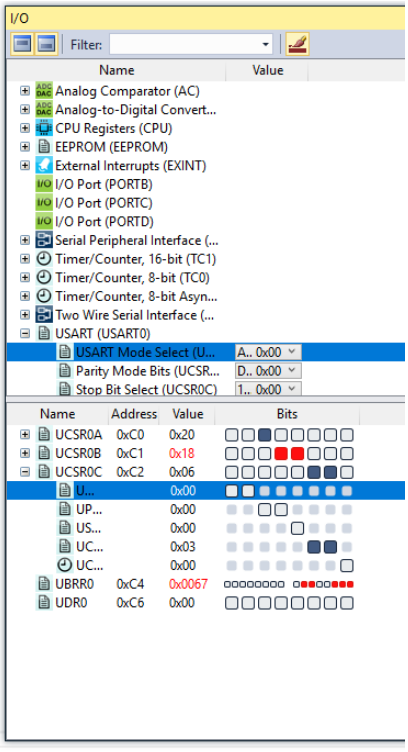
Task 1



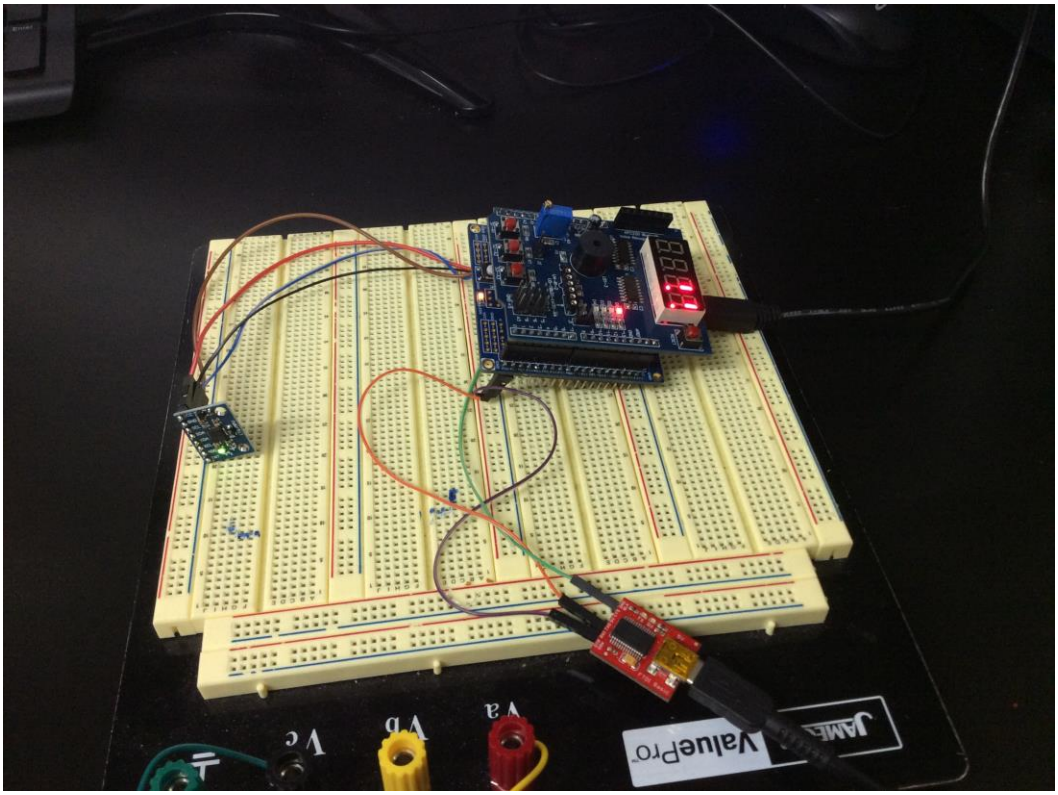
5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

TASK 1

```c
#define F_CPU 16000000UL                        // Define CPU clock Frequency

// headers program operation
#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"

// initializing variables
float Acc_x,Acc_y,Acc_z;
float Gyro_x,Gyro_y,Gyro_z;

int main()
{
    float X_a,Y_a,Z_a;
    float X_g=0,Y_g=0,Z_g=0;
    char buffer[20], float_[10];
    // I2C init function call
    I2C_Init();
    // Initialize MPU6050 initialization
    Gyro_MPU6050_Init();
    // Initialize USART, BAUD RATE = 9600
    USART_Init(9600);

    while(1)
    {
        Read_Raw_Value();

        // Values divided by scale factor
        X_a = Acc_x/16384.0;
        Y_a = Acc_y/16384.0;
        Z_a = Acc_z/16384.0;

        // Values divided by scale factor
        X_g = Gyro_x/16.4;
        Y_g = Gyro_y/16.4;
        Z_g = Gyro_z/16.4;

        // send values to USART
        dtostrf( X_a, 3, 2, float_ );
        sprintf(buffer," acce_x = %s g\t",float_);
```



6. SCREENSHOT OF EACH DEMO (BOARD SETUP)



7. VIDEO LINKS OF EACH DEMO

Task 1 video:
https://youtu.be/vOMQIEACitk

8. GITHUB LINK OF THIS DA

https://github.com/chicosisco/da_sub.git

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

"*This assignment submission is my own, original work*".
Francisco Mata Carlos