CPE301 – SPRING 2019

Design Assignment 6

Student Name: John Duriman
Student #: 5002373995
Student Email: duriman@unlv.nevada.edu
Primary Github address: https://github.com/johnduriman/pirahnaplant.git
Directory: pirahnaplant/Design Assignments/DA6/

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

List of Components used:
Atmega328P
Wires
MPU6050

Block diagram with pins used in the Atmega328P

Atmega328P	MPU6050
VCC	VCC
Ground	Ground
PC4	SDA
PC5	SCL

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


```
#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                          //Address of device were sending to
#define MPU6050_READ 0xD1                           //Address of device were reading from

float Acc_x, Acc_y, Acc_z, Gyro_x, Gyro_y, Gyro_z;      //Store acceleration and gyro values

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 - send)
    UCSR0C |= (1<<USBS0) | (3<<UCSZ00);                 //Modus Asynchron 8N1 (8 Databits, 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++;
    }
}

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){
```

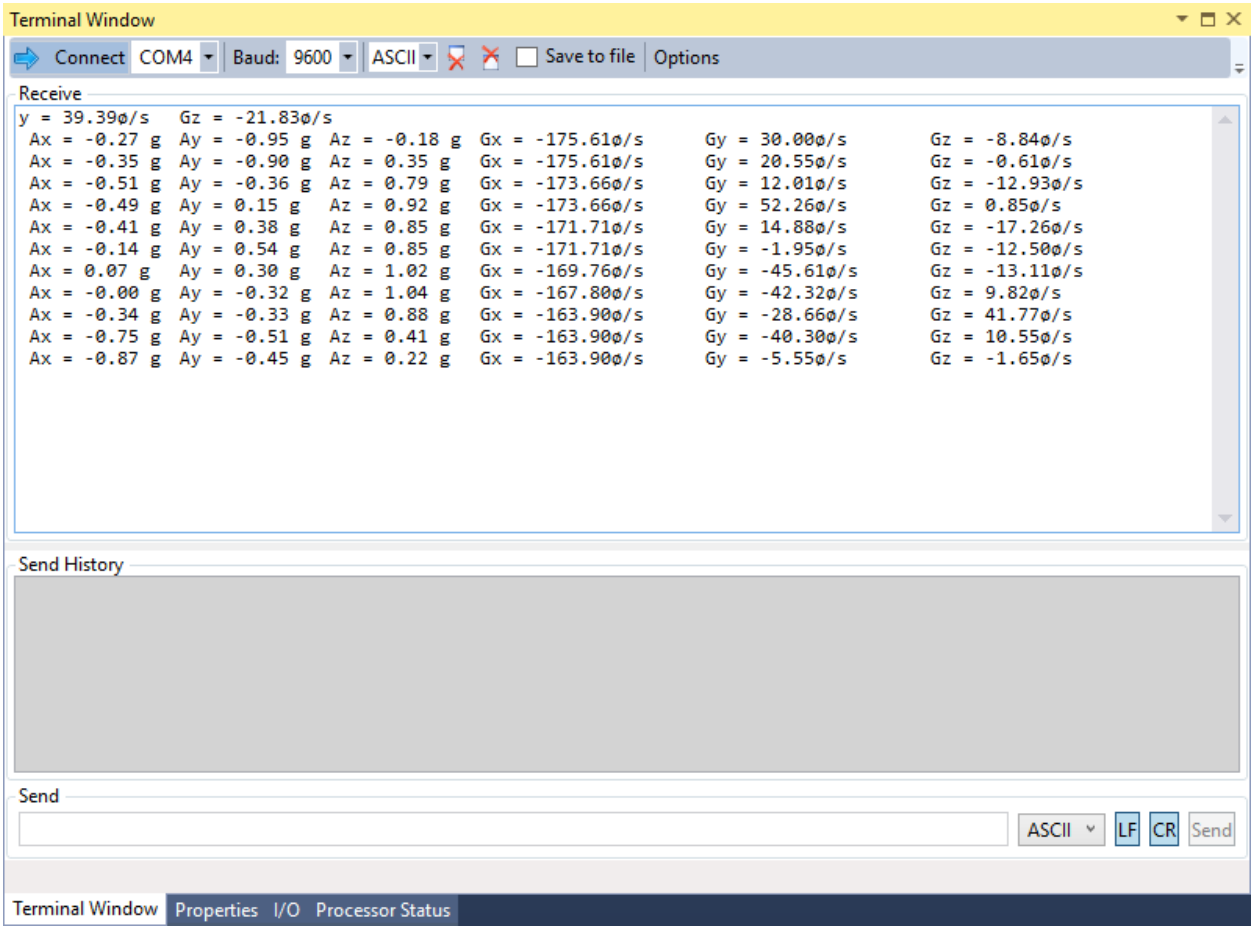
Solution2

main.c

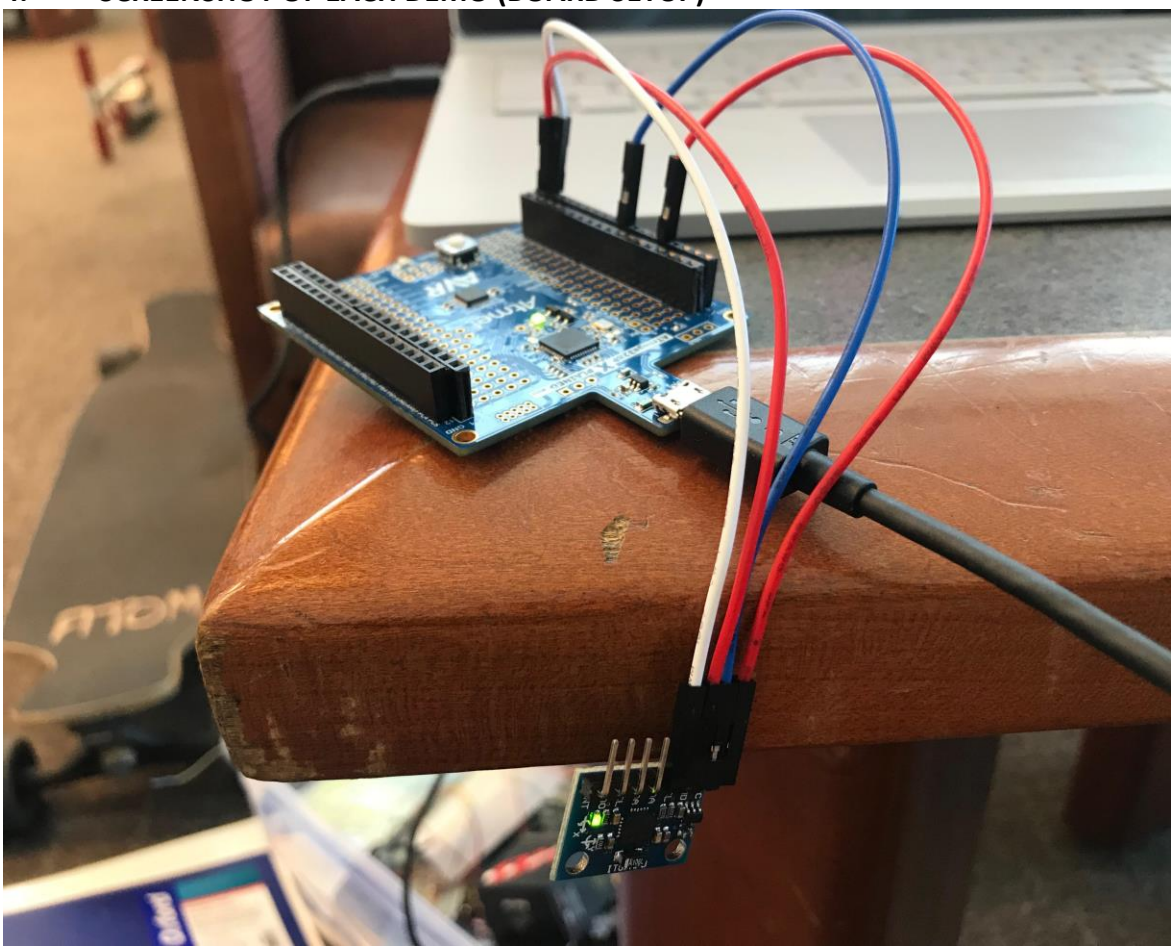
init_MPU6050 | void init_MPU6050 (void)

```
void getreading(void){

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

    i2c_start(MPU6050_READ);
    /*Store the values */
    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, Ya, Za, Xg, Yg, Zg;

    init_uart(9600);                                    //Initialize UART
    i2c_init();                                         //Initialize I2C
    init_MPU6050();                                     //Initialize

    while(1){
        getreading();

        /* Divide raw value by sensitivity scale factor to get real values */
        Xa = Acc_x/16384.0;
        Ya = Acc_y/16384.0;
        Za = Acc_z/16384.0;

        Xg = Gyro_x/16.4;
        Yg = Gyro_y/16.4;
        Zg = Gyro_z/16.4;

        /* Take values in buffer to send all parameters over USART */
        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);
    }

    return 0;
}
```

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



4. SCREENSHOT OF EACH DEMO (BOARD SETUP)



5. VIDEO LINKS OF EACH DEMO

N/A

6. GITHUB LINK OF THIS DA

pirahnaplant/Design Assignments/DA6/

Student Academic Misconduct Policy

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

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

John Duriman