CPE301 – SPRING 2019

Design Assignment 6

Student Name: yeeunLee

Student #:2001614222

Student Email: leey81@nevada.unlv.com

Primary Github address: https://github.com/yeeun219/submission\_da.git

Directory: cpe301\DesignAssignments\DA6

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

void Gyro\_Init() /\* Gyro initialization function \*/

void MPU\_Start\_Loc()

void Read\_RawValue()

void USART\_tx\_string( char \*data ) /\* SEND A STRING TO THE RS-232 \*/

void USART\_init( unsigned int ubrr ) /\* INIT USART (RS-232) \*/

while(1) read accelerometer and gyro value, then display on uart repeatly.

1. **DEVELOPED CODE OF**

/\*

\* DA6.c

\*

\* Created: 5/10/2019 11:15:17 AM

\* Author : llje2

\*/

/\*

\* ATmega16 Interface with MPU6050

\* http://www.electronicwings.com

\*

\*/

#define F\_CPU 16000000UL

#include <avr/io.h> /\* Include AVR std. library file \*/

#include <util/delay.h> /\* Include delay header file \*/

#include <inttypes.h> /\* Include integer type header file \*/

#include <stdlib.h> /\* Include standard library file \*/

#include <stdio.h> /\* Include standard I/O library file \*/

#include "MPU6050\_res\_define.h" /\* Include MPU6050 register define file \*/

#include "I2C\_Master\_H\_file.h" /\* Include I2C Master header file \*/

#include "uart.h" /\* Include USART header file \*/

#define BAUDRATE 9600

#define BAUD\_PRESCALLER (((F\_CPU / (BAUDRATE \* 16UL))) - 1)

float Acc\_x,Acc\_y,Acc\_z,Temperature,Gyro\_x,Gyro\_y,Gyro\_z;

void Gyro\_Init() /\* Gyro initialization function \*/

{

//\_delay\_ms(150); /\* Power up time >100ms \*/

I2C\_Start\_Wait(0xD0); /\* Start with device write address \*/

I2C\_Write(SMPLRT\_DIV); /\* Write to sample rate register \*/

I2C\_Write(0x07); /\* 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 Gyro configuration 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 \*/

}

void Read\_RawValue()

{

MPU\_Start\_Loc(); /\* Read Gyro 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());

Temperature = (((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();

}

/\* SEND A STRING TO THE RS-232 \*/

void USART\_tx\_string( char \*data )

{

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

{

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

UDR0 = \*data;

data++;

}

}

/\* INIT USART (RS-232) \*/

void USART\_init( unsigned int ubrr )

{

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

UBRR0L = (unsigned char)ubrr;

UCSR0B = (1 << TXEN0); // Enable RX, TX & RX interrupt

UCSR0C = (3 << UCSZ00); //asynchronous 8 N 1

}

int main()

{

char buffer[20], float\_[10];

float Xa,Ya,Za,t;

float Xg=0,Yg=0,Zg=0;

I2C\_Init(); /\* Initialize I2C \*/

Gyro\_Init(); /\* Initialize Gyro \*/

USART\_init(BAUD\_PRESCALLER); // Initialize the USART

USART\_tx\_string("Connected!\r\n"); // we're alive!

while(1)

{

Read\_RawValue();

/\* Divide raw value by sensitivity scale factor \*/

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;

/\* Convert temperature in /c using formula \*/

t = (Temperature/340.00)+36.53;

/\* Take values in buffer to send all parameters over USART \*/

*dtostrf*( Xa, 3, 2, float\_ );

*sprintf*(buffer," Ax = %s g\t",float\_);

USART\_tx\_string(buffer);

*dtostrf*( Ya, 3, 2, float\_ );

*sprintf*(buffer, " Ay = %s g\t",float\_);

USART\_tx\_string(buffer);

*dtostrf*( Za, 3, 2, float\_ );

*sprintf*(buffer," Az = %s g\t",float\_);

USART\_tx\_string(buffer);

*dtostrf*( t, 3, 2, float\_ );

/\* 0xF8 Ascii value of degree on serial \*/

*sprintf*(buffer," T = %s%cC\t",float\_,0xF8);

USART\_tx\_string(buffer);

*dtostrf*( Xg, 3, 2, float\_ );

*sprintf*(buffer," Gx = %s%c/s\t",float\_,0xF8);

USART\_tx\_string(buffer);

*dtostrf*( Yg, 3, 2, float\_ );

*sprintf*(buffer," Gy = %s%c/s\t",float\_,0xF8);

USART\_tx\_string(buffer);

*dtostrf*( Zg, 3, 2, float\_ );

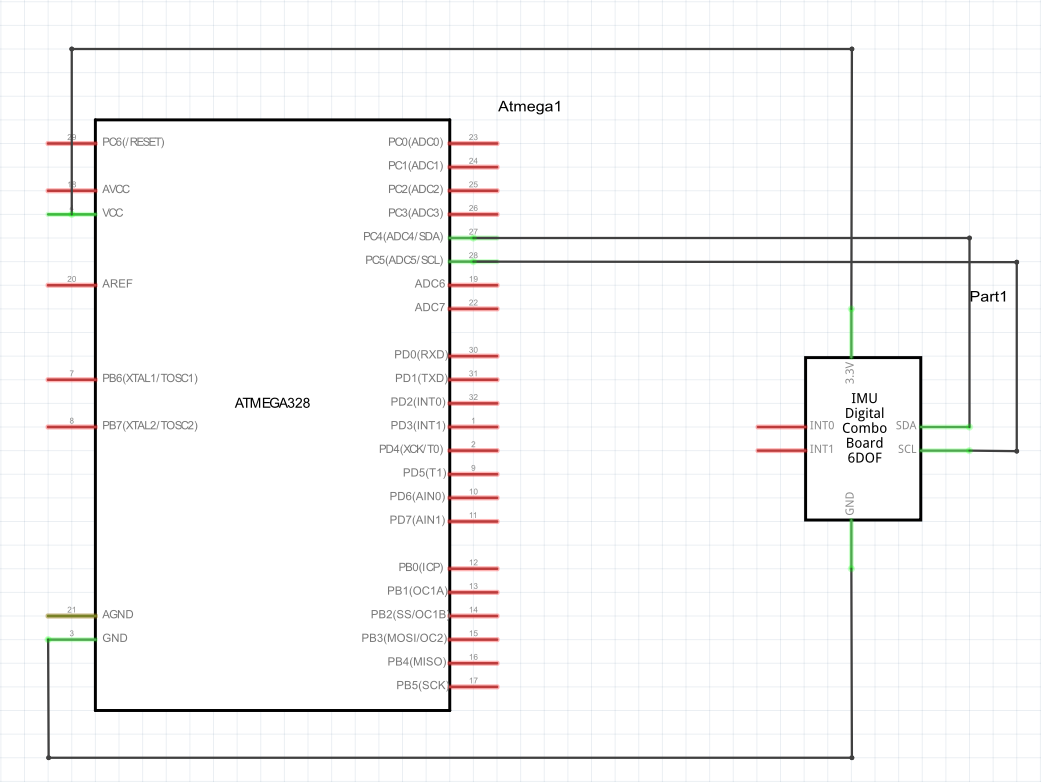
*sprintf*(buffer," Gz = %s%c/s\r\n",float\_,0xF8);

USART\_tx\_string(buffer);

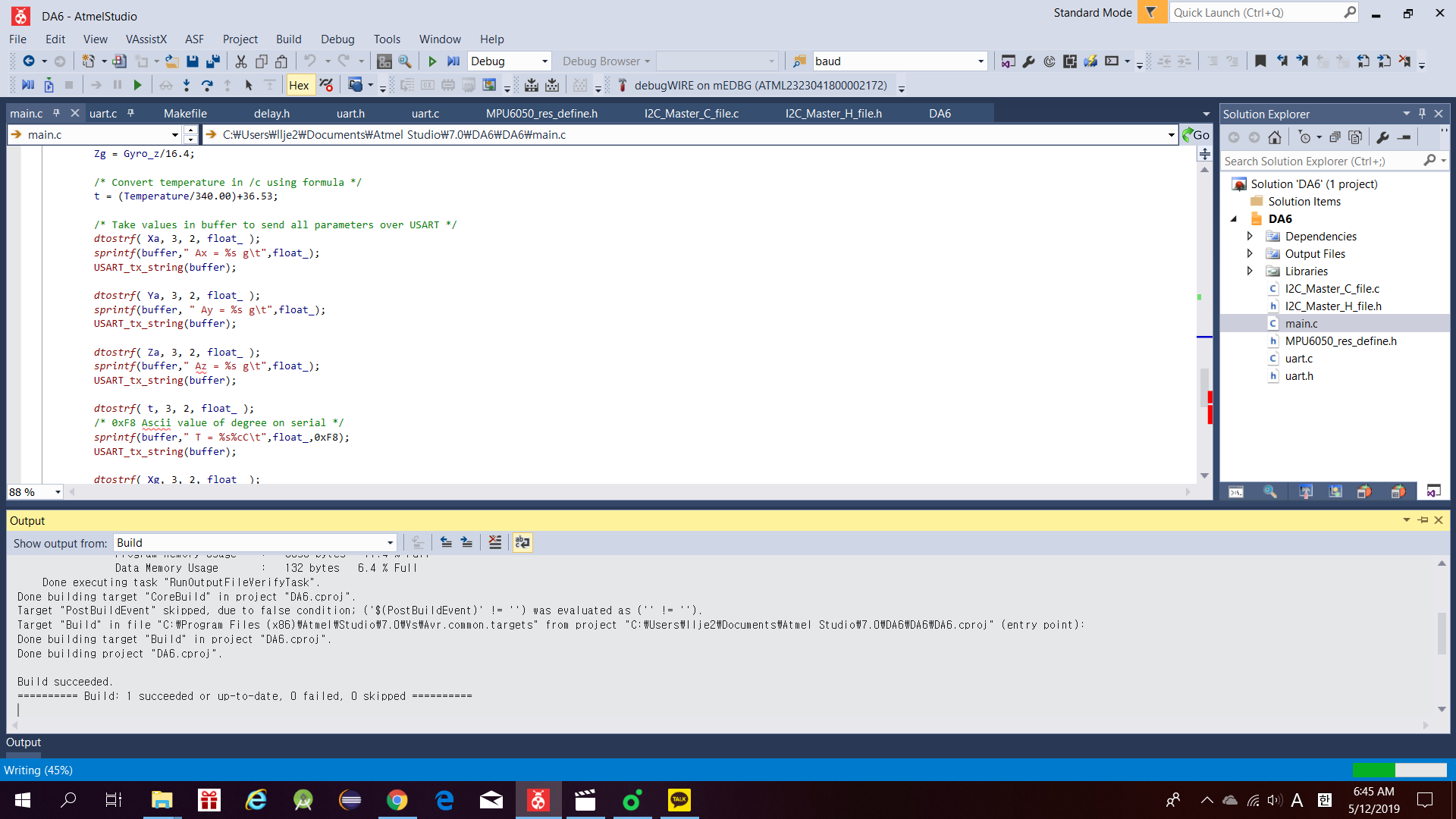
}

}

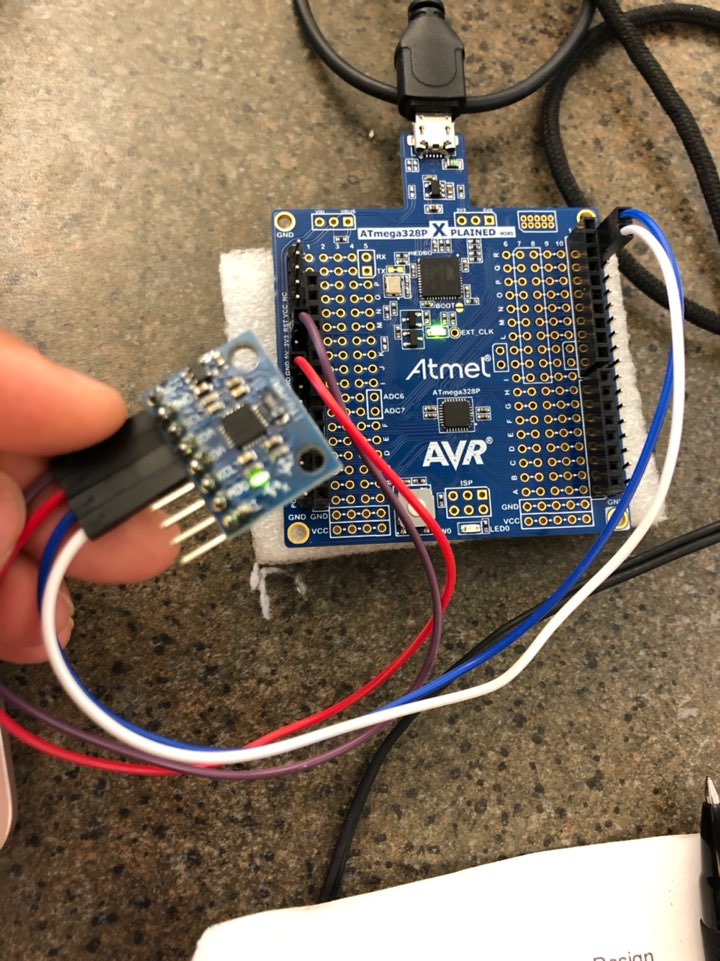
1. **SCHEMATICS**



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/Puc4eGRTJQI>

1. **GITHUB LINK OF THIS DA**

Primary Github address: https://github.com/yeeun219/submission\_da.git

Directory: cpe301\DesignAssignments\ DA6

**Student Academic Misconduct Policy**

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

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

YEEUNLEE