CPE301 – SPRING 2021

Design Assignment 5

Student Name: Brian Wolak

Student #: 2000509437

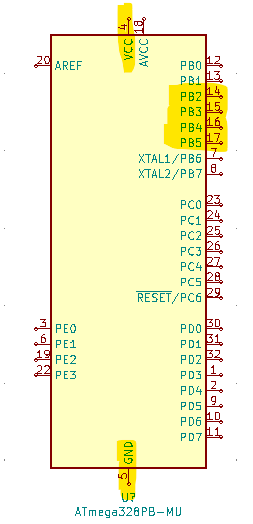
Student Email: wolak@unlv.nevada.edu

Primary Github address: <https://github.com/brianwolak/submission_da.git>

Directory: <https://github.com/brianwolak/submission_da/tree/main/DA_5>

**Task 1:**

In this design assignment a C code is written to interface a 9-DOF ICM-20948 sensor with the ATmega328pb using SPI. The goal was to gather information from the accelerometer and gyro for each axis (x,y,z). Below you can see the code, demonstration video link, and outputs verifying the proper operation.



*Design Assignment 5 Ports Used*

**Video Link:**

<https://youtu.be/sRIJNMjxx3k>

**C Code:**

#define *F\_CPU* 16000000UL //CPU clock speed

#define BAUD 9600 //9600 baud rate

#ifndef \_\_SMALL\_SPI\_H\_\_

#define \_\_SMALL\_SPI\_H\_\_

#include <avr/common.h>

void spi\_master\_init(void);

void spi\_bulk\_send(*uint8\_t* \*send\_buffer, *uint8\_t* count);

void spi\_send(*uint8\_t* send\_data);

void spi\_bulk\_exchange(*uint8\_t* \*send\_buffer, *uint8\_t* \*receive\_buffer, *uint8\_t* count);

*uint8\_t* spi\_exchange(*uint8\_t* send\_data);

#endif

#include <avr/common.h>

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/setbaud.h>

#include <stdio.h>

#include <util/delay.h>

#define DDR\_SPI DDRB //define SPI port

#define DD\_MOSI DDB3 //MOSI pin3

#define DD\_MISO DDB4 //MISO pin4

#define DD\_SCK DDB5 //SCK pin5

char display[20]; //uart string display

char X = 'X';

char Y = 'Y';

char Z = 'Z';

*uint8\_t* who\_am\_i = 0; //whoami value

//retrived data storage variables

*uint16\_t* Xacc, Yacc, Zacc, Xgyr, Ygyr, Zgyr;

//SPI initialize

void spi\_master\_init(void)

{

SPCR0 = (0 << SPIE) | (1 << SPE) | (0 << DORD) | (1 << MSTR) |

(0 << CPOL) | (0 << CPHA) | (0 << SPR1) | (1 << SPR0);

}

//bulk send function

void spi\_bulk\_send(*uint8\_t* \*send\_buffer, *uint8\_t* count)

{

while(count--)

{

SPDR0 = \*send\_buffer++;

loop\_until\_bit\_is\_set(SPSR0, SPIF);

}

}

//spi command function

void spi\_send(*uint8\_t* send\_data)

{

SPDR0 = send\_data;

while(!(SPSR0 & (1<<SPIF)));

}

//bulk send/receive function

void spi\_bulk\_exchange(*uint8\_t* \*send\_buffer, *uint8\_t* \*receive\_buffer, *uint8\_t* count)

{

while(count--)

{

SPDR0 = \*send\_buffer++;

loop\_until\_bit\_is\_set(SPSR0, SPIF);

\*receive\_buffer++ = SPDR0;

}

}

//8-bit exchange function

*uint8\_t* spi\_exchange(*uint8\_t* send\_data)

{

SPDR0 = send\_data;

while(!(SPSR0 & (1<<SPIF)));

return SPDR0;

}

//UART initialize

void USART\_init(void)

{

UBRR0H = *UBRRH\_VALUE*;

UBRR0L = *UBRRL\_VALUE*;

UCSR0C = \_BV(UCSZ01) | \_BV(UCSZ00);

UCSR0B = \_BV(RXEN0) | \_BV(TXEN0);

}

//terminal print function

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

{

while((\*data != '\0')) //loop while string not empty

{

while(!(UCSR0A & (1 << UDRE0))); //loop while data register not empty

UDR0 = \*data; //save data to UDR0

data++; //increment data

}

}

//who\_am\_i function

*uint16\_t* who\_am\_i\_func(void){

*uint16\_t* data;

PORTB &= ~(1 << 2); //spi enable

spi\_send(0x80); //send who\_am\_i address

data = spi\_exchange(0x00); //get who\_am\_i value

PORTB |= (1 << 2); //spi disable

return data;

}

//get 16-bit data function

int get\_data(*uint16\_t* addressH, *uint16\_t* addressL){

int data;

PORTB &= ~(1 << 2); //spi enable

spi\_send(addressH); //send address high

data = spi\_exchange(0x00); //get data

PORTB |= (1 << 2); //spi disable

PORTB &= ~(1 << 2); //spi enable

spi\_send(addressL); //send address low

data = ((*uint16\_t*)(8<<data)) | (*uint16\_t*)(spi\_exchange(0x00)); //combining L and H

PORTB |= (1 << 2); //spi disable

return data;

}

int main(void)

{

DDRB |= (1 << 2)|(1<<3)|(1<<5); //set output ports

DDRB &= ~(1<<4); //set input

USART\_init(); //initialize UART

USART\_tx\_string("Connection Established\r\n"); //confirm UART connection

spi\_master\_init(); //spi master initialize

PORTB &= ~(1 << 2); //spi enable

spi\_send(0x06); //address sensor clock

spi\_send(0x05); //send sensor clock speed

PORTB |= (1 << 2); //toggle SPI enable

//who\_am\_i

who\_am\_i = who\_am\_i\_func();

//print who\_am\_i info

*sprintf*(display, "%d", who\_am\_i);

USART\_tx\_string("Slave address is: ");

USART\_tx\_string(display);

USART\_tx\_string("\r\n");

while (1)

{

who\_am\_i = who\_am\_i\_func(); //who\_am\_i check

if (who\_am\_i == 234) //if true, device connected

USART\_tx\_string("ICM-20948 is still Connected\r\n");

else //if false, device not connected

USART\_tx\_string("ICM-20948 is NOT Connected\r\n");

//gyro X

Xgyr = get\_data(0xB3, 0xB4);

//print gyro X info

*sprintf*(display, "%d", Xgyr);

USART\_tx\_string("Gyro X Value: ");

USART\_tx\_string(display);

USART\_tx\_string("\r\n");

//gyro Y

Ygyr = get\_data(0xB5, 0xB6);

//print gyro Y info

*sprintf*(display, "%d", Ygyr);

USART\_tx\_string("Gyro Y Value: ");

USART\_tx\_string(display);

USART\_tx\_string("\r\n");

//gyro Z

Zgyr = get\_data(0xB7, 0xB8);

//print gyro Z info

*sprintf*(display, "%d", Zgyr);

USART\_tx\_string("Gyro Z Value: ");

USART\_tx\_string(display);

USART\_tx\_string("\r\n");

//accelerometer X

Xacc = get\_data(0xAD, 0xAE);

//print accel X info

*sprintf*(display, "%d", Xacc);

USART\_tx\_string("Accelerometer X Value: ");

USART\_tx\_string(display);

USART\_tx\_string("\r\n");

//accelerometer Y

Yacc = get\_data(0xAF, 0xB0);

//print accel Y info

*sprintf*(display, "%d", Yacc);

USART\_tx\_string("Accelerometer Y Value: ");

USART\_tx\_string(display);

USART\_tx\_string("\r\n");

//Accelerometer Z

Zacc = get\_data(0xB1, 0xB2);

//print accel Z info

*sprintf*(display, "%d", Zacc);

USART\_tx\_string("Accelerometer Z Value: ");

USART\_tx\_string(display);

USART\_tx\_string("\r\n");

USART\_tx\_string("\r\n");

*\_delay\_ms*(500); //terminal delay

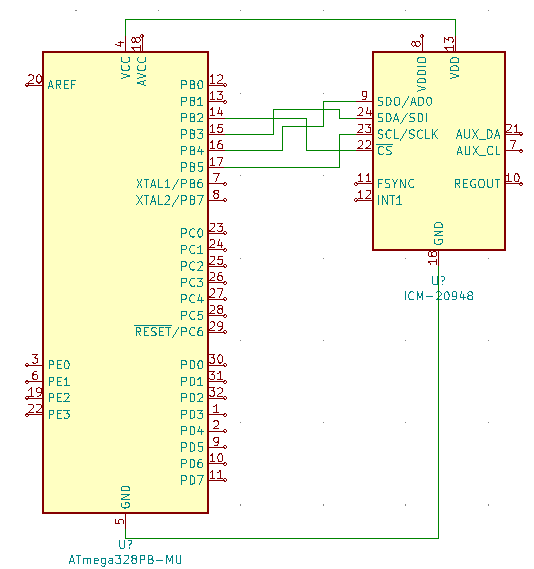
}

}

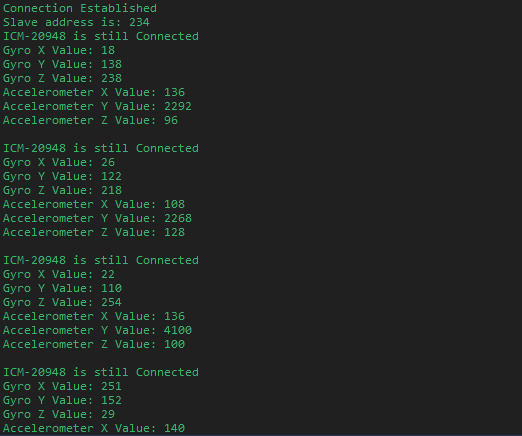
A picture containing electronics, circuit

Description automatically generated

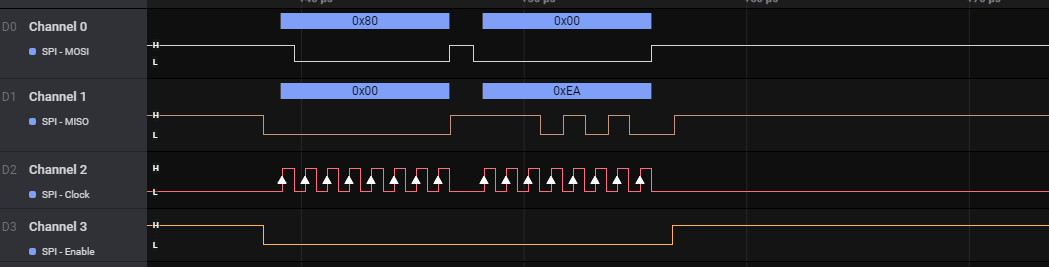
*Design Assignment 5 Board Setup*



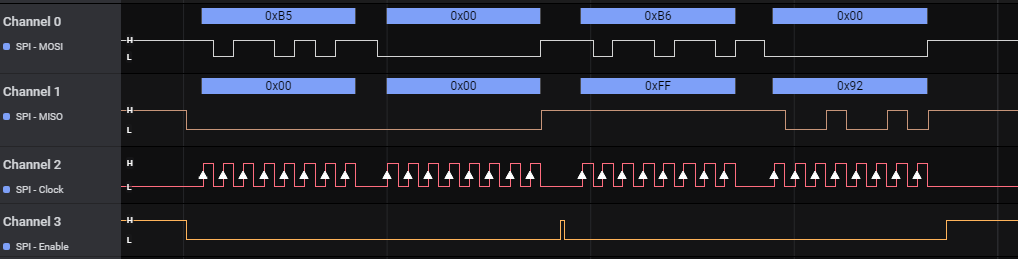
*Design Assignment 5 Circuit*

**

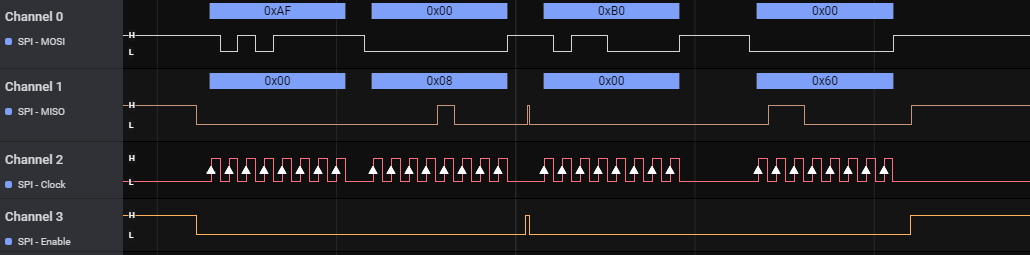
*Terminal Output Showing Initial Connection and Confirmation of Connection*

**

*Logic Output Showing who\_am\_i Reading*

**

*Logic Output Showing Y Gyro Scope H/L Read*

**

*Logic Output Showing Y Accelerometer H/L Read*

**GitHub Link:**

<https://github.com/brianwolak/submission_da/tree/main/DA_5>

**Video Link:**

<https://youtu.be/sRIJNMjxx3k>