

실험제목 : SPI(Serial Peripheral Interface)

실험목적

모토로라에서 개발한 직렬 통신의 한 방법인 SPI(Serial Peripheral Interface)의 개념을 이해하고 ATmega328PB와 SPI 방식의 3축 가속도 센서 소자인 ADXL345 간의 데이터 통신에 대해 이해한다.

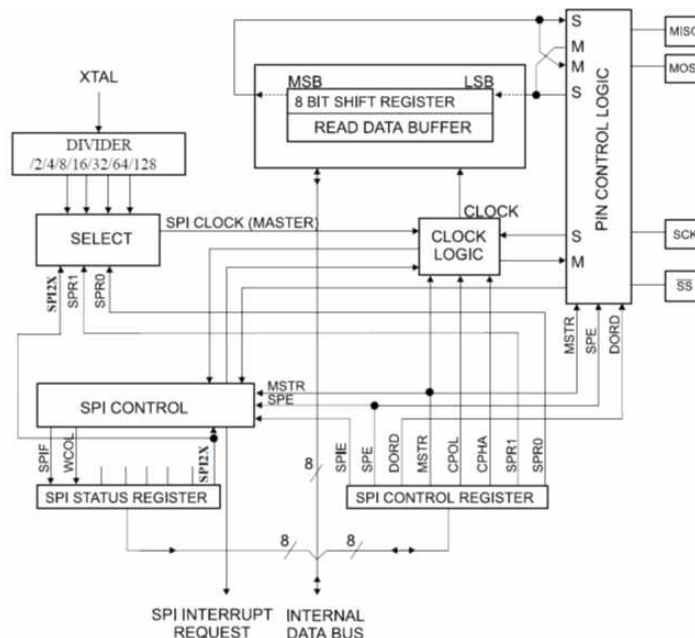
실험 준비물

- Microchip Studio 7
- RealTerm / TeraTerm
- Atmega328PB Xplained Mini
- ADXL345 3-Axis Accelerometer Module (GY-291)

실험에 필요한 예비지식

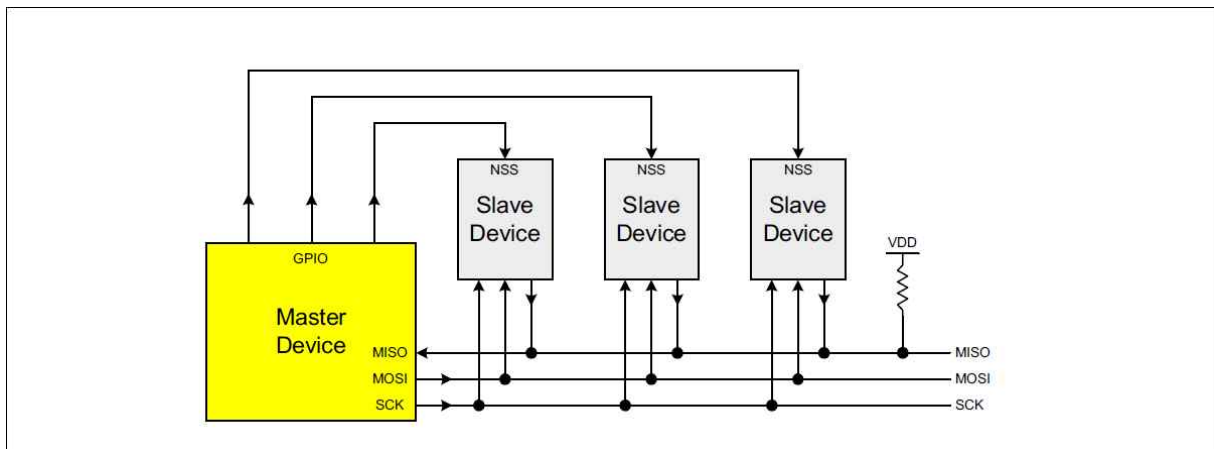
1. SPI 정의

SPI(Serial Peripheral Interface)는 모토로라에서 개발한 인터페이스 방식으로 동기식 통신 방식이다. Data의 송수신에는 MISO, MOSI, SCK의 3개의 선이 필요하며 전 이중방식 통신(Full Duplex)이 가능하다. 하나의 Master와 여러 개의 Slave를 가질 수 있는데 이때는 Slave 선택을 위한 선이 하나 더 필요하다. 주로 근거리 통신, 즉 마이크로컨트롤러와 주변장치간의 고속 직렬 통신에 사용된다.



2. SPI 연결

- ① MOSI (Master Out, Slave In) : Master Device로부터 출력되어 Slave Device로 입력되는 신호.
- ② MISO(Master In, Slave Out) : Slave Device로부터 출력되어 Master Device로 입력되는 신호.
ATmega328PB가 Master Device로 동작할 경우 MISO 핀은 High Impedance 상태가 됨.
- ③ SCK(Serial Clock) : Master Device와 Slave Device에서 MOSI와 MISO의 데이터 전송 동기에 사용됨.
항상 Master Device에서 출력됨.
- ④ NSS(Slave Select) : 1:N 통신 시, Master Device에서 해당 Slave Device를 선택하는데 필요한 신호. 데이터 전송을 시작할 때 NSS신호는 Low상태여야 하며 High가 되면 Slave Mode로부터 빠져나옴.

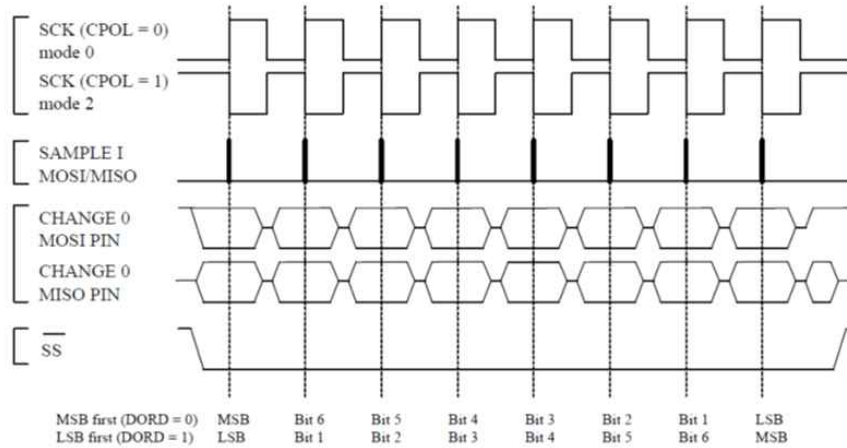


전형적 SPI 연결 구성

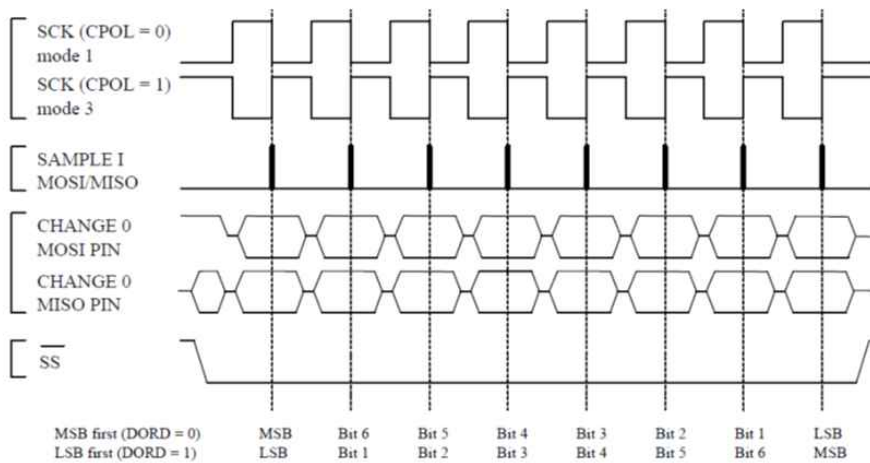
3. SPI 동작 타이밍

아래의 그림에 나타난 것처럼 데이터를 동기화시키는 방법에는 4가지 경우가 있다.

SPI Modes	Conditions	Leading Edge	Trailing Edge
0	CPOL=0, CPHA=0	Sample on Rising Edge	Setup on Falling Edge
2	CPOL=1, CPHA=0	Sample on Falling Edge	Setup on Rising Edge

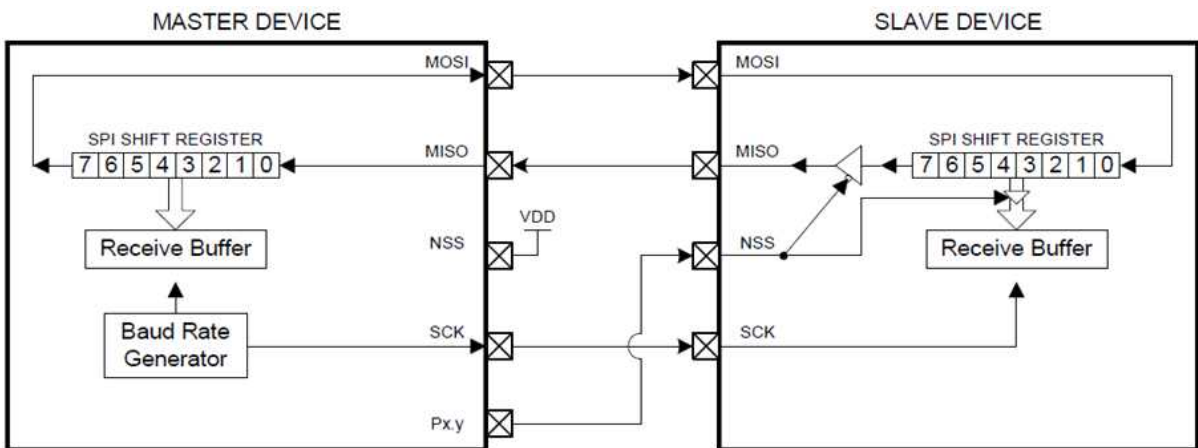
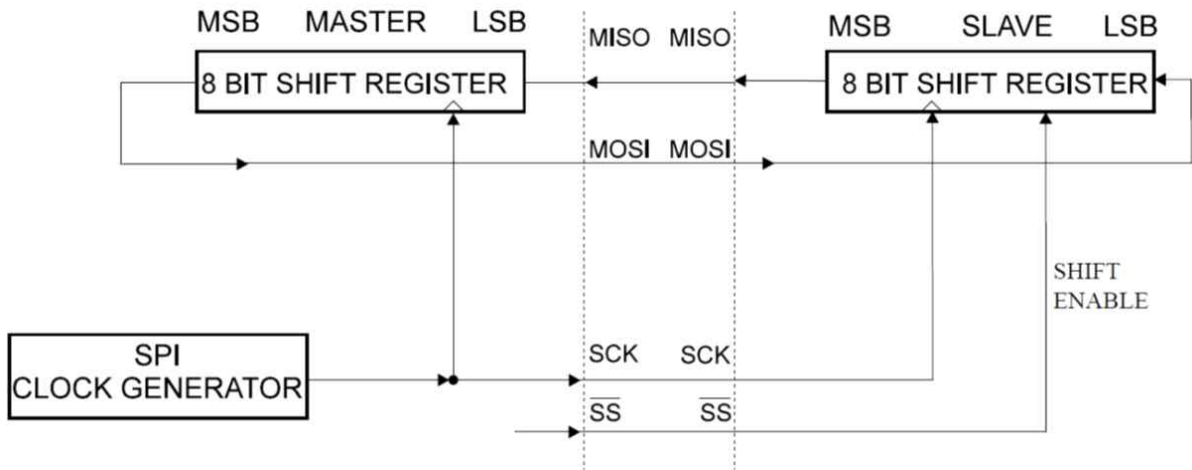


SPI Modes	Conditions	Leading Edge	Trailing Edge
1	CPOL=0, CPHA=1	Setup on Rising Edge	Sample on Falling Edge
3	CPOL=1, CPHA=1	Setup on Falling Edge	Sample on Rising Edge



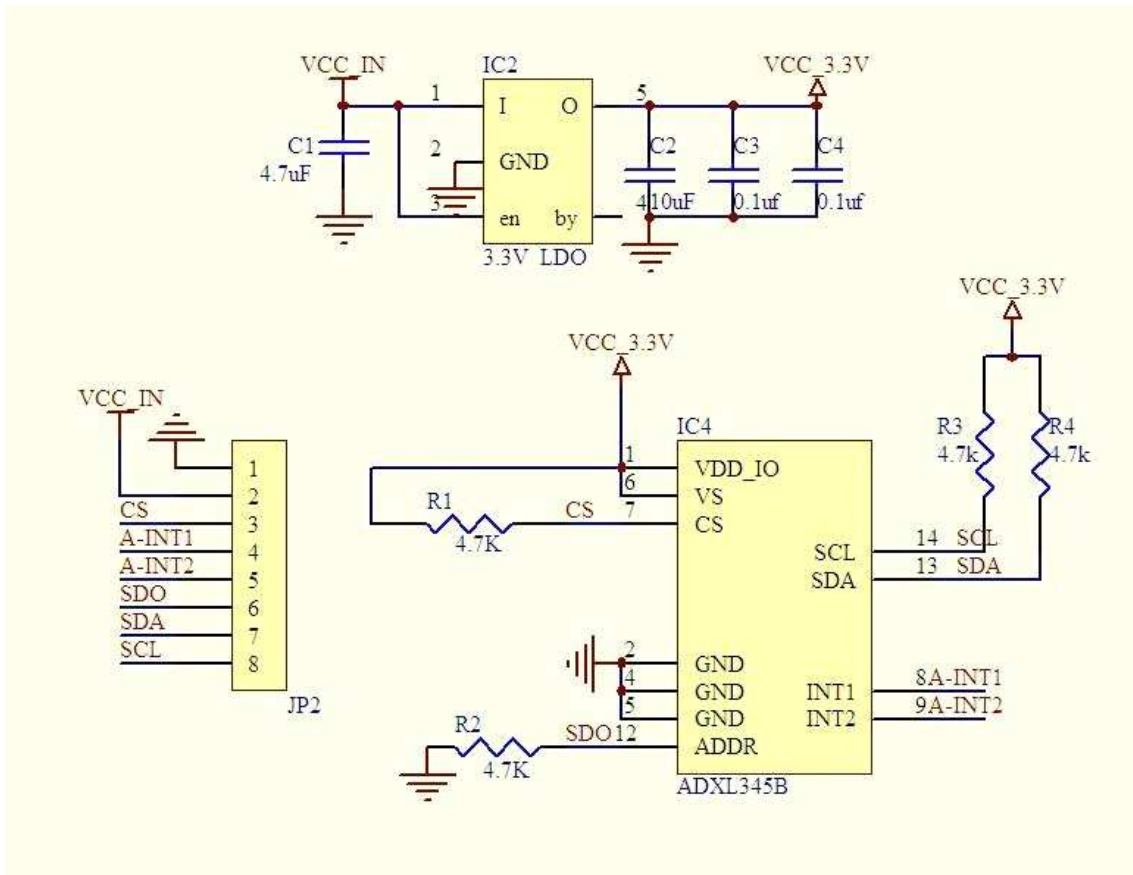
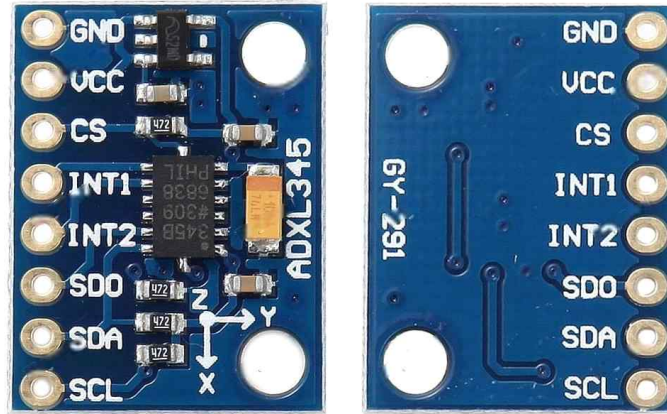
4. SPI 동작

- ① Master Device에서 클록을 생성하여 Slave Device와 동기화 함.
- ② 데이터 전송시, NSS신호는 항상 Low 상태여야 함.
- ④ 전송이 완료되면 쉬프트 레지스터의 데이터가 SPDR Register로 복사되고 SPIF 비트가 1로 세트됨.

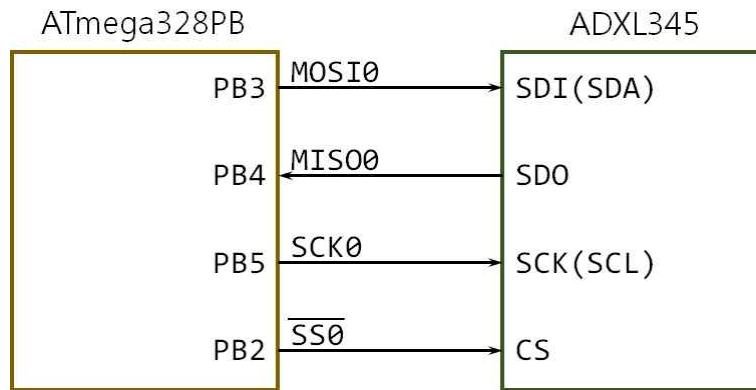


실험내용

GY-291 Board에는 Analog Device사의 3축 가속도 센서인 ADXL345가 아래의 그림과 같이 연결되어 있으며, 3축의 가속도 값을 SPI 혹은 IIC 방식을 통해 출력한다.



1. ATmega328PB에 아래와 같이 3축 가속도 센서 모듈인 GY-291 Board를 연결하고, SPI Interface를 사용하여 3축의 가속도 값을 읽어내어 UART0에 연결된 터미널 프로그램의 화면에 출력하는 프로그램을 작성하시오. UART0 polling 방법을 사용하고, baud rate는 500,000 bps로 설정하시오.



가. SPI0와 관련된 다음과 같은 함수를 만들어서 사용하시오.

- 1) `void spi0_init(void)`
 - (1) SPICR0 레지스터의 초기화
 - (2) 해당 GPIO port pin의 초기화
- 2) `void write_adxl345(uint8_t reg_addr, uint8_t data)`

ADXL345의 해당 레지스터(`reg_addr`)에 데이터(`data`)를 출력
Status Register(SPSR0)의 SPIF0 비트를 읽어 쓰기 동작이 완료된 것을 확인한 후에 리턴.
- 3) `uint8_t read_adxl345(uint8_t reg_addr)`

ADXL345의 해당 레지스터(`reg_addr`)로부터 데이터(`data`)를 읽은 후에 이 값을 반환.
- 4) `void read_adxl345_reg_multi(uint8_t num, uint8_t start_addr, uint8_t *buff)`

ADXL345의 시작 레지스터(`start_addr`)부터 연속해서 `num` 번지의 레지스터 값을 읽어 `buff`로 지정된 메모리에 저장한 후 리턴.
(DATA0, DATA1, DATAY0, DATAY1, DATAZ0, DATAZ1의 데이터를 읽는데 사용)

나. (H/W 연결 상태 확인)

`read_adxl345()` 함수를 이용하여 ADXL345 소자의 Device ID를 읽어 USART0로 출력하고, 그 값을 기록한다.

다. (ADXL345 설정)

`write_adxl345()` 함수를 이용하여 아래에 표에 나타난 값을 ADXL345 소자의 해당 레지스터에 써 넣는다.

Register Address	Name	Type	Set Value	Description
0x2C	BW_RATE	R/W	00001010	BW=50Hz (Output Data Rate = 100Hz)
0x31	DATA_FORMAT	R/W	00001000	Full Resolution +/-2g, 4mg/LSB
0x2D	POWER_CTL	R/W	00001000	Enter Measurement Mode

라. (ADXL345로부터 3축 가속도 값 읽기)

`read_adxl345_reg_multi()` 함수를 이용하여 아래에 표에 나타난 ADXL345 소자의 해당 레지스터 값을 읽어 내어 USART0를 통해 출력한다. 먼저 INT_SOURCE (Address 0x30) 레지스터의 Bit 7을 읽어 새로운 가속도 데이터가 준비된 것을 확인한 후에 3축의 가속도 센서의 출력 데이터(6 바이트)를 차례대로 모두 읽어 내어 출력한다.

Register Address	Name	Type	Reset Value	Description
0x32	DATA0	R	00000000	Low Byte of X-Axis Data
0x33	DATA1	R	00000000	High Byte of X-Axis Data
0x34	DATAY0	R	00000000	Low Byte of Y-Axis Data
0x35	DATAY1	R	00000000	High Byte of Y-Axis Data
0x36	DATAZ0	R	00000000	Low Byte of Z-Axis Data
0x37	DATAZ1	R	00000000	High Byte of Z-Axis Data

```
#include <avr/io.h>
#include <stdio.h>

int main(void)
{
    // Init pins for SPI0 (DDRB)

    // Set ADXL CS signal to High (PB2)

    // Init SPI0 as a MASTER (SPCR)

    // Read ADXL345 ID and display it

    // Set ADXL345 BW_RATE

    // Set ADXL345 DATA_FORMAT
    // Full Resolution, +/-2g (4mg/LSB)

    // Enter MEASUREMENT mode

    while (1)
    {
        // Wait until new data is available

        // Read 3-axis data with multiple-byte read format

        // Display 3-axis data

    }
}

// Refer to Fig. 37 of ADXL345 Datasheet
void write_adxl345(uint8_t reg_addr, uint8_t data)
{
}

uint8_t read_adxl345_reg(uint8_t reg_addr)
{
}

void read_adxl345_reg_multi(uint8_t num, uint8_t start_addr, uint8_t *buff)
{
}
```

숙제

1. ADXL345 소자에 대해 조사
2. 위 실험 1의 프로그램에서 얻은 세 축의 중력가속도 데이터 가운데 두 축의 중력가속도 값을 이용하여 현재의 기울기 각도를 계산한 후, 이를 UART0에 연결된 터미널 프로그램의 화면에 출력하는 프로그램을 작성하시오. UART0 polling 방법을 사용하고, baud rate는 9,600 bps로 설정하시오.