

SPI (SERIAL PERIPHERAL INTERFACE)

Overview

SPI 는 Motorola 사에서 처음 고안한 통신 방식으로 MCU 와 주변장치간의 serial 통신을 위한 규약이다. 이는 외부 주변장치와 clock 을 통하여 동기화 하는 동기식 통신 방식이며, 하나의 Master 와 하나 또는 다수의 Slave Device 간의 통신이 가능하다. 통신방식이 비교적 간단하여 IIC 와 함께 대표적으로 사용되는 통신 방식이다.

SPI 는 기본적으로 다음과 같이 4개의 Signal Pin 을 제공한다.

MOSI (Master Out Slave Input)

: Device 가 Master 로 설정되어 있는 경우 이 Pin 은 출력을 위한 Output pin으로 사용되며 Slave 로 설정되어 있다면 수신을 위한 Input Pin으로 사용된다..

MISO (Master Input Slave Out)

: 마찬가지로 Device 가 Master 로 설정되어 있는 경우 수신을 위한 Input Pin 으로 사용되며, Slave 일 경우 출력을 위한 Output Pin 으로 사용된다.

SS (Slave Select)

: SPI 는 여러 Device 간 통신이 가능하다. 이때, 현재 통신을 하고 있는 Device 를 구분하기 위한 Pin 이다.

SCK (Serial Clock)

: SPI 는 동기식 방식이다. 외부 Device 간의 Serial Data 의 동기화를 위한 Clock Pin 이다.

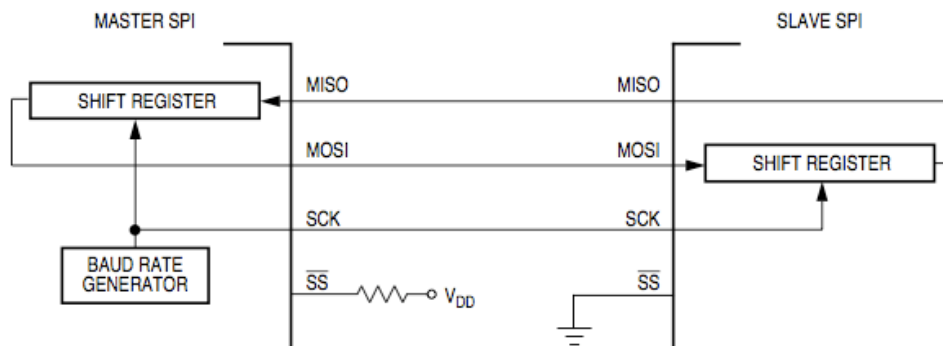


Figure 4-1 Master/Slave Transfer Block Diagram

위 그림은 Master 와 Slave 간의 Data 의 전송을 나타낸 Diagram 이다. 위에서 보는 바와 같이 Master 의 input pin 은 Slave 의 output pin 과 연결되어 있고, Slave 의 input pin 은 Master 의 output pin 과 연결되어 있다. 그대로 생각하면 Master 에서 내보내는 output data 는 Slave 의 input data 가 되고 Slave 에서 내보내는 output data 는 Master 의 input data 가 된다.

SPI 는 동기화 방식이라고 이야기 했다. Device 간에는 서로 clock 을 통해 연결되어 있는데 이를 바탕으로 Master 와 Slave 간의 signal data 의 동기성을 유지한다. Data 를 전송하는 쪽이든, 받는 쪽이든 이 clock 을 이용하여 signal data 를 sampling 하게 된다.

주목할 것은, SPI 전송의 시작과 끝은 Master 쪽에서 주관한다는 것이다. 즉, Master 에서 data 를 보내지 않으면 Slave 는 아무런 일도 하지 않는다. Master 만이 Data 전송을 시작하고 끝낼 수 있다. Baud rate generator 가 Master 에 있는 것도 이러한 이유에서 일 것이다.

Data 는 Shift Register 를 통해서 말 그대로 Shifting 된다. Shifting 된다는 것은 하나 씩 밀린다는 의미로 하나의 bit 가 교환되는 방식이다 Master 에서 clock 에 동기화 하여 Slave 로 하나의 bit 를 보내면 Slave 역시 clock 에 하나의 bit 를 실어 Master 로 전송한다. 마치 톱니바퀴 돌아가듯 data 가 교환된다.

Register Description

: SPI 통신은 다음과 같이 크게 4 가지의 Register 를 통하여 제어된다

SPISR (SPI Status Register)

: SPI 의 상태 레지스터로 현재 SPI 의 송신 및 수신 상태를 나타낸다.

SPIDR (SPI Data Register)

: SPI 의 Data 레지스터로 송수신을 위한 Data Buffer 역할을 한다. 송신과 수신 모두 하나의 Register 를 공유한다.

SPICR (SPI Control Register)

: SPI 의 송신 또는 수신과 관련된 제어를 하는 레지스터이다.

SPIBR (SPI Baud-rate Register)

: SPI 의 clock 을 제어한다.

I. SPICR 1 (SPI Control Register)

Register Address: \$__0

	Bit 7	6	5	4	3	2	1	Bit 0
R	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
W								
Reset:	0	0	0	0	0	1	0	0

Figure 3-1 SPI Control Register 1 (SPICR1)

SPE : SPI 통신의 가능 불가능을 설정. 1로 설정되어 있을 경우 SPI pin을 SPI 통신 전용으로 사용한다.

MSTR : Mater 와 Slave 를 결정하는 register

1 - SPI 를 Master 모드로 설정

0 - SPI 를 Slave 모드로 설정

LSBFE (Least Significant Bit First Enable)

: SPI 전송 시 어느 bit 부터 전송을 할 것인지를 설정한다. 이 register 의 설정은 단지 어떠한 bit 부터 전송할 지를 결정 하는 것 뿐이며, 실제 data 에 영향을 주지는 않는다. Data 는 항상 MSB(bit.7) 부터 읽고 쓰는 것을 원칙으로 한다.

1 - 가장 최 하위 bit 부터 전송

0 - 가장 최 상의 bit 부터 전송

I. SPICR 2 (SPI Control Register)

Register Address: \$__1

	Bit 7	6	5	4	3	2	1	Bit 0
R	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
W								
Reset:	0	0	0	0	0	0	0	0



= Reserved

Figure 3-2 SPI Control Register 2 (SPICR2)

SPISWAI (SPI Stop in Wait Mode Bit)

: Wait 상태에서 전력 소모를 최소화 하기 위해 clock 을 제어

1 - 전력 소모 방지를 위해 clock 을 정지한다.

0 - 정지 되어 있는 clock 을 정상화 한다.

2. SPIBR (SPI Baud rate Register)**Register Address: \$__2**

	Bit 7	6	5	4	3	2	1	Bit 0
R	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
W								
Reset:	0	0	0	0	0	0	0	0

= Reserved

Figure 3-3 SPI Baud Rate Register (SPIBR)

SPPR2 - SPPR0 : SPI Baud Rate Preselection Bits

SPR2 - SPR0 ; SPI Baud Rate Selection Bits

$$\text{BaudRateDivisor} = (\text{SPPR} + 1) \cdot 2^{(\text{SPR} + 1)}$$

The baud rate can be calculated with the following equation:

$$\text{Baud Rate} = \text{BusClock} / \text{BaudRateDivisor}$$

SPI 의 전송 속도 즉, baud rate 를 설정하는 bit 이다. 위에서 보는 바와 같이 SPPR 과 SPR 두 개의 register 를 적절히 설정하여 전송하고자 하는 baud rate 를 만들어 낸다. 만들어 내는 공식은 위에 나와 있는 공식을 이용한다. 다음 도표는 25 MHz clock 을 사용할 때 baud rate 를 만들어 내는 register 값이다.

Table 3-4 Example SPI Baud Rate Selection (25 MHz Bus Clock)

SPPR2	SPPR1	SPPR0	SPR2	SPR1	SPR0	BaudRate Divisor	Baud Rate
0	0	0	0	0	1	4	6.25 MHz
0	0	0	0	1	0	8	3.125 MHz
0	0	0	0	1	1	16	1.5625 MHz
0	0	0	1	0	0	32	781.25 kHz
0	0	0	1	0	1	64	390.63 kHz
0	0	0	1	1	0	128	195.31 kHz
0	0	0	1	1	1	256	97.66 kHz
0	0	1	0	0	0	4	6.25 MHz
0	0	1	0	0	1	8	3.125 MHz

공식에 대입 해 보면 어렵지 않게 계산이 가능하다. 예를 들어 SPPR 값이 0 SPR 값이 1 이라면,

BaudRateDivisor 값은 $(0+1) \cdot 2^{(1+1)} = 4$ 가 나오며 Baud Rate 값은 $25\text{MHz} / 4 = 6.25$ 가 나온다. 이는 6.25 MHz 의 속도로 clock 을 생성하여 SPI device 간의 동기 통신을 가능하게 해 준다.

3. SPIR (SPI Status Register)

Register Address: \$__3

	Bit 7	6	5	4	3	2	1	Bit 0
R	SPIF	0	SPTEF	MODF	0	0	0	0
W								
Reset:	0	0	1	0	0	0	0	0

= Reserved

Figure 3-4 SPI Status Register (SPIR)

SPIF (SPI Interrupt Register)

: Data 의 전송상태를 나타내는 register 로 SPIDR 에 data 가 전송이 완료되면 1로 설정된다. 전송이 완료 되었다는 것은 송신이나 수신이 완료되었음을 의미한다. 이 register 는 SPIR 를 읽을 때 다시금 0으로 설정된다.

SPTEF (SPI Transmit Empty Interrupt Flag)

: SPI Data Register 가 비어있는지를 나타내는 flag 로, 비어있다면 1, data 가 있다면 0 으로 설정된다.

4. SPIDR (SPI Data Register)

Register Address: \$__5

	Bit 7	6	5	4	3	2	1	Bit 0
R	Bit 7	6	5	4	3	2	2	Bit 0
W								
Reset:	0	0	0	0	0	0	0	0

Figure 3-5 SPI Data Register (SPIDR)

(SPIF 가 1 로 설정되어 있을 경우에만 read only 이며, 0 으로 설정되어 있는 경우 read / write 모두 가능하다.) 정

SPI Data Register 는 기본적으로 송신과 수신을 하나의 register 를 공유한다. 하나의 byte 를 register 에 써넣으면 queue 에 넣어 송신을 시작하며 만약, SPI 가 Master 로 설정되어 있다면 이전 byte 가 송신되는 동시에 다음 byte 가 바로 송신 하게 된다. 말했듯이, SPTEF 가 1 로 설정되어 있다면 data 가 새로운 data 를 받을 준비가 되어있음을 의미한다... SPIF 가 1로 설정된 시점부터 다음 전송이 끝날 때 까지 data 는 언제든지 읽을 수 있다.