

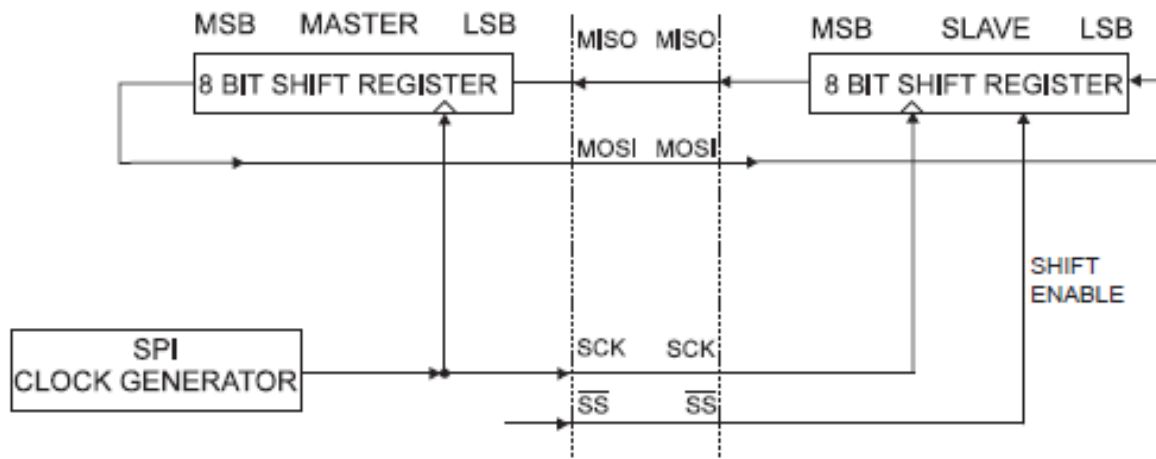
Project #13(Pjt13_SPI_TC77)

◆ SPI(Serial Peripheral Interface) Device

- 3-wire full-duplex synchronous communication.
- 설정에 따라
 - . master 혹은 slave로 동작
 - . LSB first 혹은 MSB first transfer
 - . 전송속도 선택(7 가지) 및 double speed 가능(현재 $f_{osc}/4$ 등 속도무관 동작!)



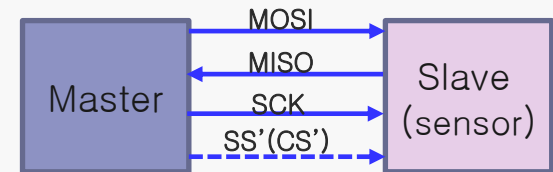
Figure 80. SPI Master-slave Interconnection



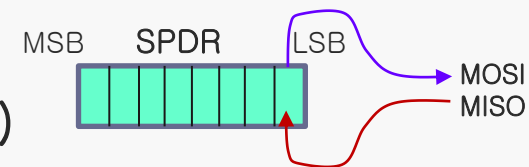
Project #13(Pjt13_SPI_TC77)

◆ SPI Control

- SS(CS) 신호(핀)을 low(0)로 설정하면 SPI 동작개시 (active low)



- SPI 동작상태에서 SPDR(SPI Data Register)에 값을 넣으면(write) 1bit씩 송수신 자동 시작(총 8bit)
- 수신 비트는 SPDR의 보낸 비트를 덮어씀(반드시 주고 받음)
- 1byte 전송이 완료되면 인터럽트 발생



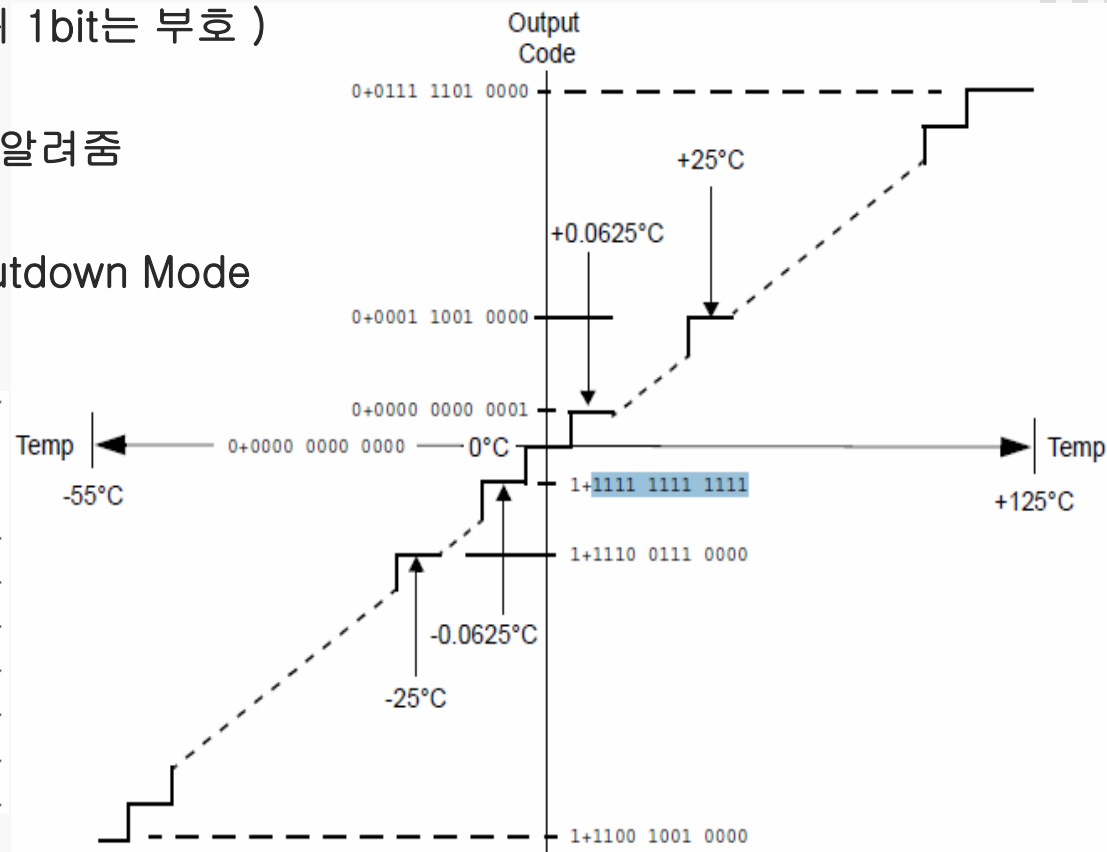
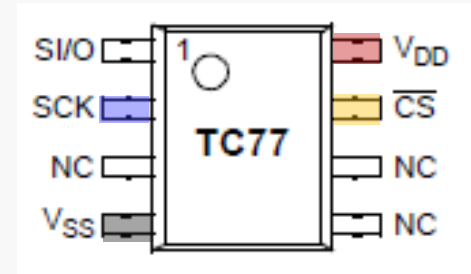
- . ISR에서 SPDR을 읽으면 수신 데이터가 됨
- . 여러 바이트를 송수신할 경우 수신 데이터를 읽은 후, 다음 데이터를 씀
- 모든 데이터의 송수신이 완료되면 SS신호를 high(1)로 설정하여 SPI 동작 중지

Project #13(Pjt13_SPI_TC77)

◆ TC77 센서

- 온도 값을 2 바이트 16bit 중 13bit value로 알려줌
 - . MSB first로 송신
 - . 16bit 중 하위 3bit는 상태정보 표시
 - . 13bit를 signed int로 해석함(최상위 1bit는 부호)
 - . $\text{degree} = (\text{value} \gg 3) * 0.0625$;
- Chip ID를 2 바이트 16bit 중 14bit로 알려줌
 - . 16bit 중 하위 2bit는 무시됨(0x54)
- Continuous Conversion Mode와 Shutdown Mode
두 가지 모드 설정 가능

Temperature	Binary MSB / LSB	Hex
+125°C	0011 1110 1000 0111	3E 87h
+25°C	0000 1100 1000 0111	0B 87h
+0.0625°C	0000 0000 0000 1111	00 0Fh
0°C	0000 0000 0000 0111	00 07h
-0.0625°C	1111 1111 1111 1111	FF FFh
-25°C	1111 0011 1000 0111	F3 87h
-55°C	1110 0100 1000 0111	E4 87h



Project #13(Pjt13_SPI_TC77)

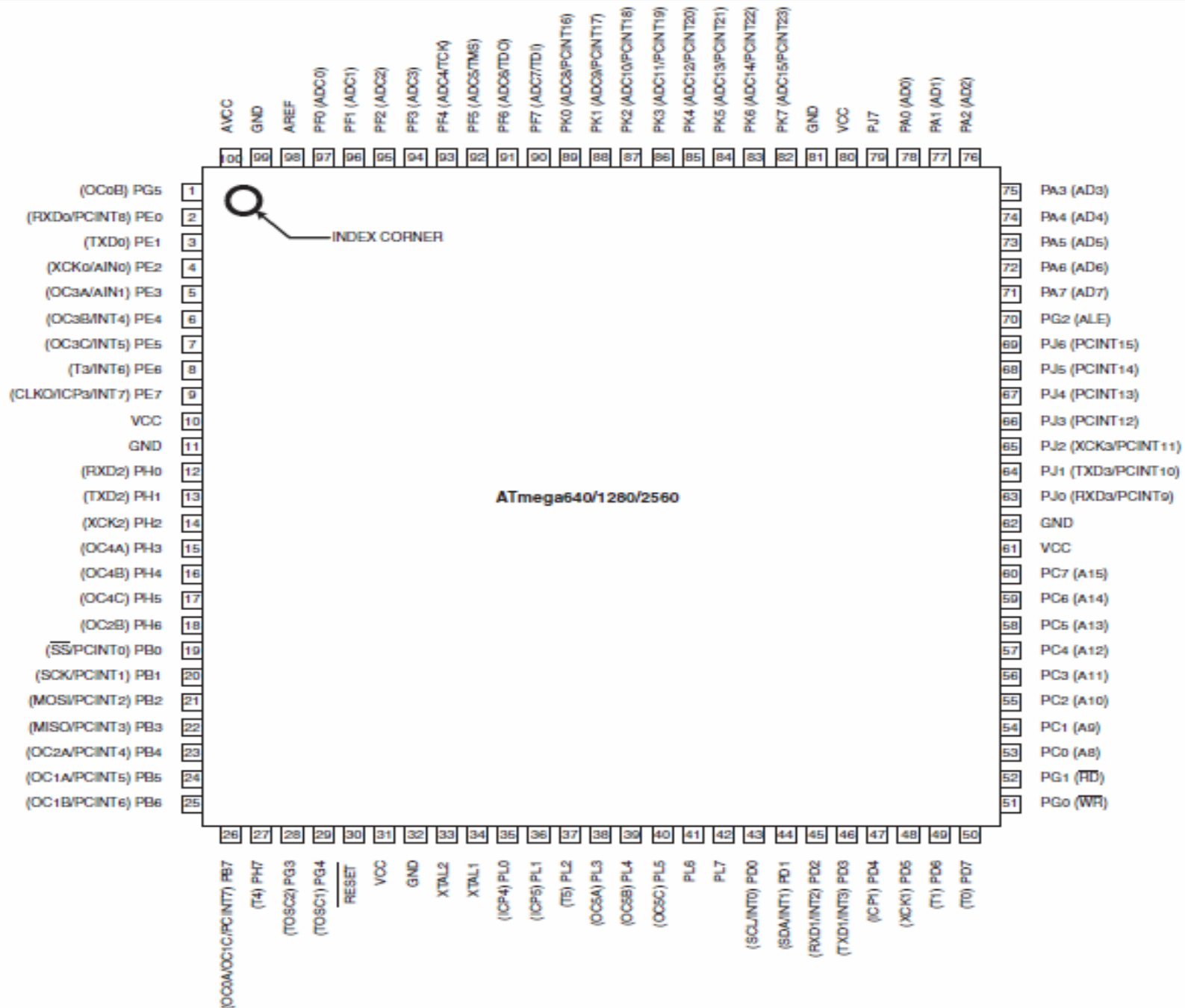
◆ TC77 센서 레지스터 및 Protocol

– SPI를 동작시킨 후

- . 0x00, 0x00을 보내면, **Config Register**를 Continuous Conversion Mode로 설정하고, **Temp Register**(측정 완료된 온도 데이터) 값을 보내옴(하위 3bit 중 bit 2는 정상 여부를, bit 0~1은 무시됨)
- . 0xff, 0xff를 보내면, Config Register를 Shutdown Mode로 설정하고, **M_ID Register**(Chip ID) 값을 보내옴(상위 8bit에 0x54, 하위 2bit는 무시, 나머지는 0).
- . 현재 write 값에 관계없이 온도 값이 수신됨 ← MOSI pin과 관계 있는 듯 !!!

Name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value at Powerup/Reset
CONFIG	C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0	XXXX/XXXX 0000/0000
TEMP	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0	*	x	x	1111/1111 0000/0*XX
M_ID	0	1	0	1	0	1	0	0	0	0	0	0	0	0	x	x	0101/0100 0000/00XX

* Bit 2 = 0 during power-up; otherwise, bit 2 =1



Project #13(Pjt13_SPI_TC77)

◆ SPI(Serial Peripheral Interface) Application : TC77 Driver

```
void spi_init()
{
    sbi(DDRB, 0); // sbi(DDRB, nSS,nCS); // output
    sbi(DDRB, 1); // sbi(DDRB, SCK); // output
    sbi(DDRB, 2); // sbi(DDRB, MOSI); // output
    cbi(DDRB, 3); // cbi(DDRB, MISO); // input

    sbi(PSR, SPI2X); // double speed

    SPCR = ( (1 << MSTR) | (1 << SPE) | (1 << SPIE) ); // MASTER, MSB first, Fosc/4
}

void spi_select() { cbi(PORTB, 0); } // cbi(PORTB, nSS

void spi_release() { sbi(PORTB, 0); } // sbi(PORTB, nS

void spi_write(uint8_t data) { SPDR = data; }
```

```
ISR(SPI_STC_vect)
{
    struct task task;
    uint8_t data;

    data = SPDR;

    task.fun = task_tc77;
    sprintf(task.arg, "%d", data);

    task_insert(&task);
}
```

Project #13(Pjt13_SPI_TC77)

◆ SPI Sensor Application : task_tc77

```
void task_tc77(void *arg) #define TC77_WAIT_HI 1
{                          #define TC77_WAIT_LO 2
    static uint8_t  state;
    static uint16_t value;

    if (!strcmp(arg, "")) {
        state = TC77_WAIT_HI;
        spi_select();
        spi_write(0x00);
    }
    else {
        switch(state) {
            case TC77_WAIT_HI :
                value = atoi(arg) << 8;  state = TC77_WAIT_LO; spi_write(0x00);
                break;
            case TC77_WAIT_LO:
                value |= atoi(arg);  value = (value >> 3) * 0.0625;
                spi_release();
                printf("task_tc77() : current_temperature ➡ %d degree.Wn", value);
                break;
            default:
                spi_release();
                printf("task_tc77() : unexpectd state in task_tc77...Wn");
        }
    }
}
```

```
void task_cmd(void *arg)
{
    ...
    ...
    else if (!strcmp(cp0, "tc77"))
        task_tc77("");
    ...
    ...
}
```

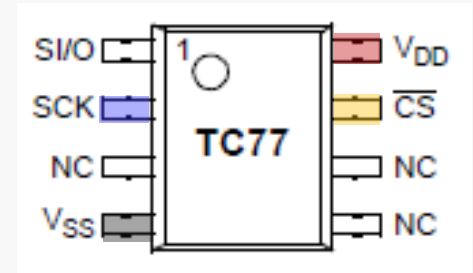
Project #13(Pjt13_SPI_TC77)

◇ TC77 pin description

Vdd ↔ Power : 빨강 } 보드 밑면 소켓
Vss ↔ Ground : 검정 }

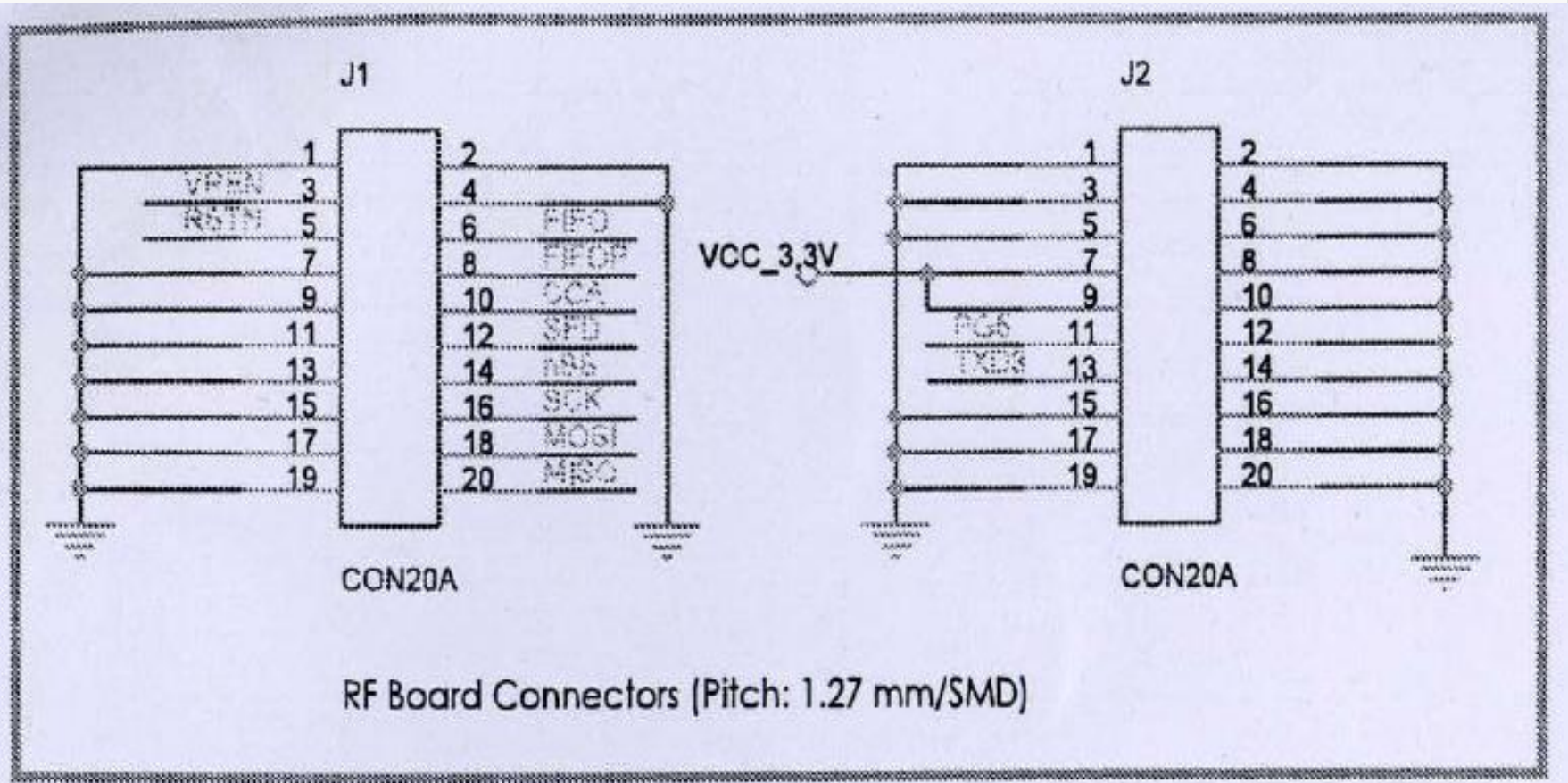
CS' ↔ nSS : 노랑(select) } 보드 윗면 좌측 소켓
SCK ↔ SCK : 청색(clock) }
SI/O ↔ MISO : 흰색(data) }

* MOSI 연결하지 않음(MCU가 Slave로 동작할 때 사용)



Project #13(Pjt13_SPI_TC77)

◆ PIN Layout



Project #13(Pjt13_SPI_TC77)

◆ PIN Layout

