STM32에서는 SPI 기능을 제공한다구!

Date

목차

1) 프로젝트 복사

2) STM32 칩에서 제공하는 SPI 기능.

(1) 프로젝트를 단순 ctrl+C, V를 하면 위험하다.

손(프로젝트 내부의 파일들이 기존의 폴더를 가르킨다.)

⇒ 해결책: 강의 12분 ~

복사가 필요한 이유 (테스트를 하기 위해)

(2)

기존	⇒	변경
FND_SCLK : PB15	⇒	SPI2_MOSI
FND_DIO : PB14	⇒	SPI2_MISO
FND_RCLK : PB13	⇒	SPI2_SCLK

STM에서 제공하는 SPI 기능의 핀배열 때문에 세팅을 바꿔야 한다.

ex) PB13 에서는 MOSI, MISO 기능을 하지 않는다.

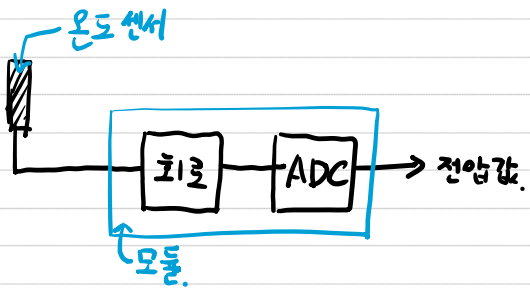
PB15 ⇒ PB13

PB14 ⇒ PB15

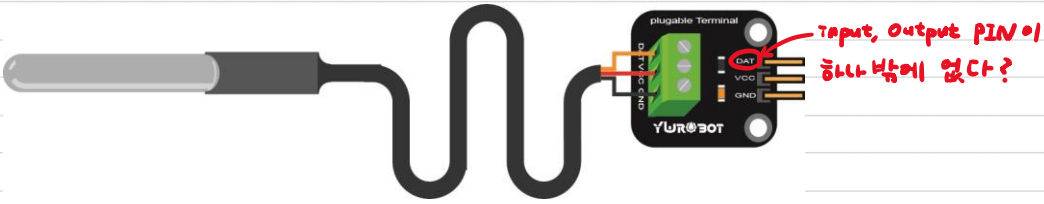
PB13 ⇒ PB14

온도 센서를 붙여보자!!

Date



- 배선도



OneWire.cpp — read()

└ read_bit()

OneWire.h

- DIRECT_MODE_OUTPUT()
- DIRECT_WRITE_LOW()
- DIRECT_MODE_INPUT()
- DIRECT_READ()

Temperature.pde - OneWire ds(PIN)

```
for (bitmask = 0x01; i ; i <<= 1)
```

└ i가 0이 될 때 끝난다.

┌ int8_t == char

uint8_t == unsigned char

Date

```
uint8_t OneWire::read() {
    uint8_t bitMask;
    uint8_t r = 0;

    for (bitMask = 0x01; bitMask; bitMask <<= 1) {
        if ( OneWire::read_bit()) r |= bitMask;
    }
    return r;
}
```

r = 0000 0000.

읽은 값이 있으면, if()

없으면 r==0 반환.

```
// Read a bit. Port and bit is used to cut lookup time and provide
// more certain timing.
//
uint8_t OneWire::read_bit(void)
{
IO_REG_TYPE mask=bitmask;
volatile IO_REG_TYPE *reg IO_REG_ASM = baseReg;
uint8_t r;

noInterrupts();
DIRECT_MODE_OUTPUT(reg, mask);
DIRECT_WRITE_LOW(reg, mask);
delayMicroseconds(3);
DIRECT_MODE_INPUT(reg, mask);// let pin float, pull up will raise
delayMicroseconds(10);
r = DIRECT_READ(reg, mask);
interrupts();
delayMicroseconds(53);
return r;
}
```

읽은 값을 r에 저장.

r을 반환.

(ex)

bitMask	r	r |= bitMask.	
0000 0001	0000 0000	0000 0001	
0000 0010	0000 0001	0000 0011	
0000 0100	0000 0011	X	← if() 문에
0000 1000	0000 0011	0000 1011	들어가지
0001 0000	0000 1011	0001 1011.	않는 경우.

- read_bit()함수로 1bit씩 값을 읽어온다.
- (read_bit()함수의 return이 0인 경우) == (읽은 값이 0인 경우.)
- if (read_bit( )) == 0 이면, r |= bitMask 연산을 하지 않는다.

⇒ 최종적으로 read()함수의 r에 8bit의 INPUT 값이 쓰여진다.