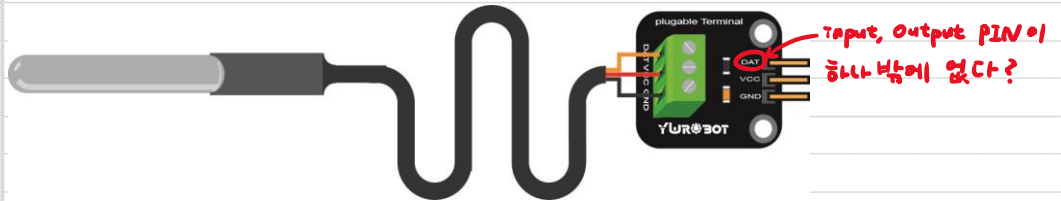


- 배선도



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 &lt;&lt;= 1)

↑ i가 0이 될 때 끝난다.

└ int8\_t == char

uint8\_t == unsigned char

```
uint8_t OneWire::read(){
    uint8_t bitMask;
    uint8_t r = 0; r = 0000 0000.
```

```
    for (bitMask = 0x01; bitMask; bitMask <= 1) {
```

```
        if ( OneWire::read_bit()) r |= bitMask;
```

```
    }
```

```
    return r;
```

```
}
```

*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); 읽은 값을 r에 저장.
```

```
    interrupts();
```

```
    delayMicroseconds(53);
```

```
    return r;
```

```
}
```

*r을 반환.*

*(ex)*

<i>bitMask</i>	<i>r</i>	<i>r  = bitMask.</i>	
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 return 이 0인 경우) == (읽은 값이 0인 경우.)

• *if (read\_bit()) == 0* 이면, *r |= bitMask* 연산을 하지 않는다.

⇒ 최종적으로 *read()* 함수의 *r*에 8bit의 INPUT 값이 쓰여진다.