

# DTPML Series

## Communication Protocol

### V1.2

---

목 차	1. 개 요		1
	2. 회로 연결시 주의사항	SPI	1
		RS-485	1
	3. 연결 회로도	SPI	2
		RS-485	3
	4. 통신 프로토콜	SPI	4
		RS-485	8

---

## 1. 개요

- 본 문서는 (주)디웰전자 의 DTPML series 의 통신 프로토콜 통합 문서 입니다.
- 본 문서의 프로토콜은 DTPML 제품군에만 적용됩니다.
- DTPML 은 디지털 통신 방식 2 가지를 지원합니다. (SPI, RS485)
- 한 모델이 동시에 여러 통신방식을 지원하지 않습니다.
  - SPI 통신 모델은 SPI 로만 통신이 가능합니다.(IO level 3.3V)
  - RS-485 통신 모델은 RS-485로만 통신이 가능합니다.(MODBUS RTU 프로토콜 지원)
- 본 문서 전체를 볼 필요는 없습니다.  
목차를 참고하여 사용하는 모델의 통신 부분만 발췌독 하기 바랍니다.

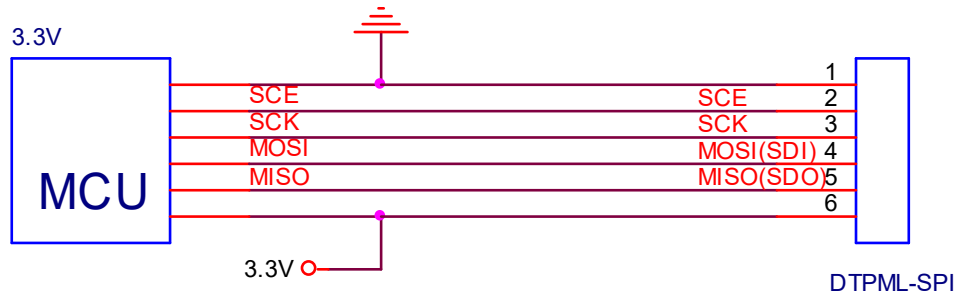
## 2. 회로 연결시 주의사항

- SPI
  - 센서 공급 전압은 3.3V 입니다.
  - 통신포트의 IO 레벨은 3.3V입니다. 5V tolerant 미지원.
  - 5V 로 동작하는 컨트롤러와 연결할 경우 레벨 시프터 또는 저항 분압 회로를 통해 전압을 맞춰 사용하십시오.
  - SPI 포트를 MCU 프로그램 다운로드 포트와 공용으로 사용하지 마십시오.  
예) SPI 포트가 한 개 있는 Atmega8 시리즈 주의. 그 외 SPI포트가 하나이면서 프로그램의 다운로드 역시 SPI 포트를 이용하는 MCU일 경우 특히 주의해야 합니다. 이럴 경우 반드시 프로그램 다운로드 할 경우에는 센서를 먼저 제거 한 후 다운로드 하십시오.
- RS-485
  - 센서 공급 전압은 5V 입니다.
  - 통신 포트(D+,D-)에는 Bias 및 종단저항이 적용돼 있으며, 제품 후면의 slide 스위치를 통해 종단저항의 적용 여부를 결정할 수 있습니다.

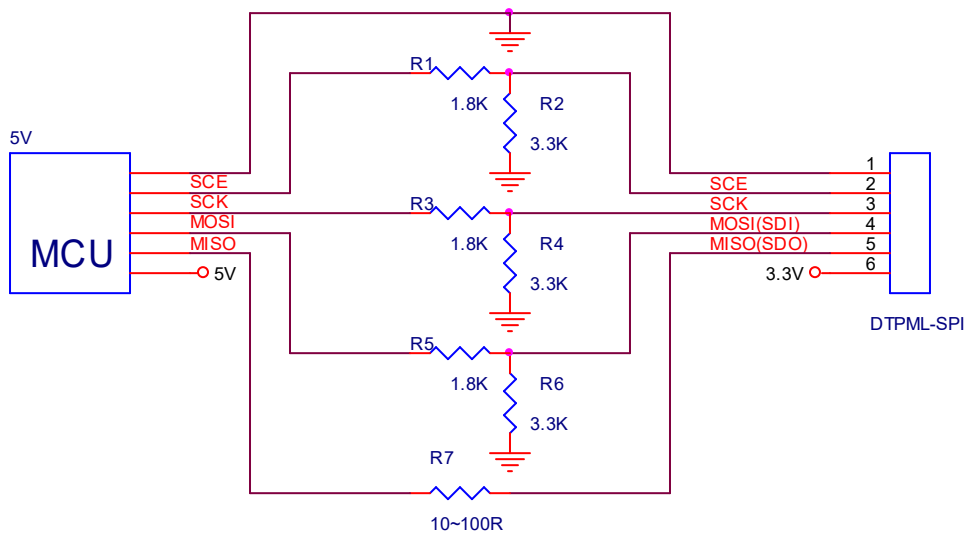
### 3. 연결 회로도

- SPI

➤ 3.3V 로 동작하는 컨트롤러와 연결시 회로도(직접 연결)



➤ 5V 로 동작하는 컨트롤러와 연결시 회로도(저항 분압 회로)

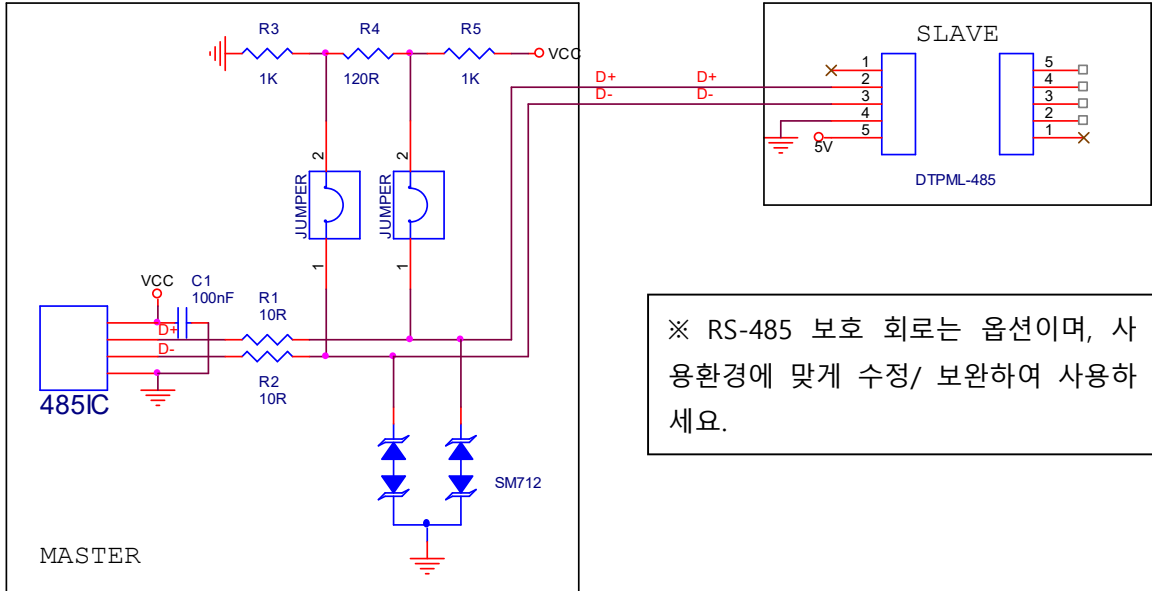


MISO 포트는 필요한 경우 댐핑저항(Damping Resistor)을 추가하셔도 됩니다.  
저항값은 상황에 맞추어 0~100 옴 정도로 사용하시면 됩니다.

Communication Protocol

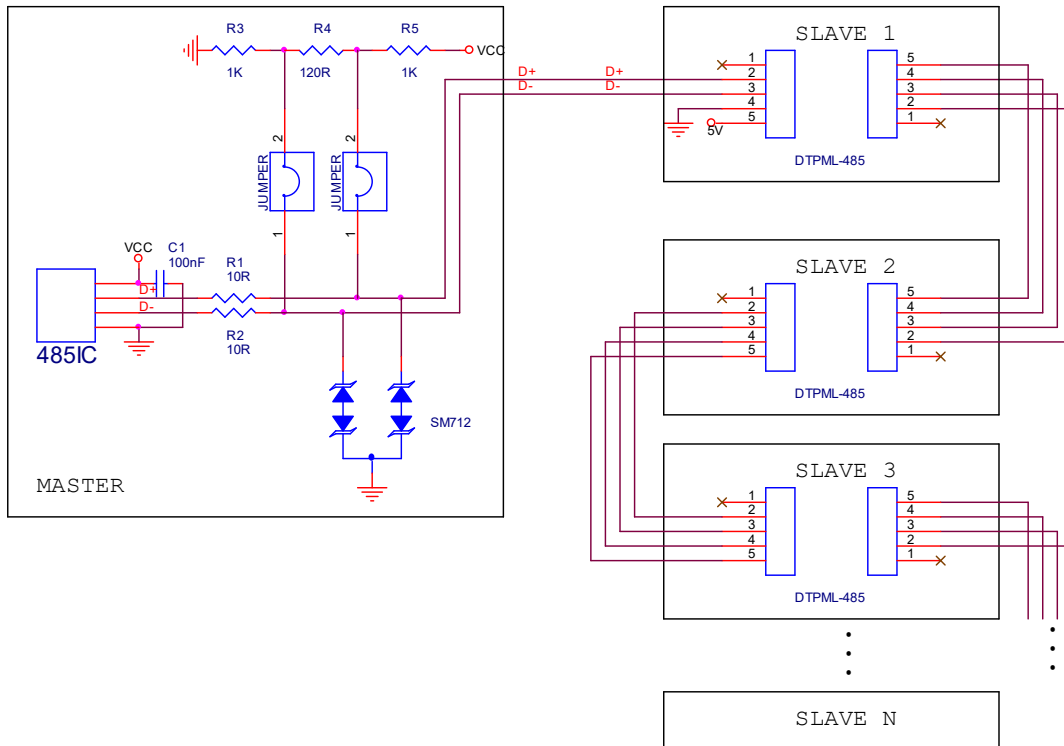
● RS-485

➤ 1:1 연결



※ DTPML-485 는 연결 커넥터가 2 개가 있습니다. 두 커넥터는 서로 1:1 로 연결 돼 있으며 Master 와 센서를 1:1 연결 할 경우 두 커넥터 중 한곳에 연결하면 됩니다.(위 회로도)

➤ 1:N 연결



※ DTPML-485 는 연결 커넥터가 2 개가 있습니다. 두 커넥터는 서로 1:1 로 연결 돼 있으며 Master 와 센서를 1:N 연결 할 경우 나머지 커넥터를 이용해 센서의 확장 연결이 쉽습니다.

## 4. 통신 프로토콜

### 4.1. SPI

#### 4.1.1. 통신 규격

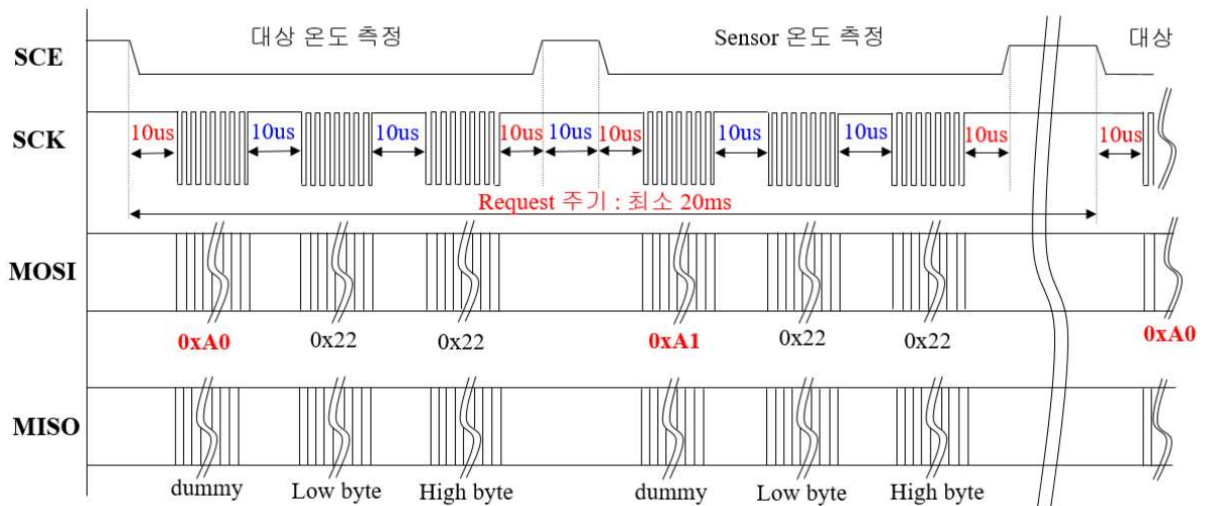
clock	SPI Mode	SCK idle	sampling	bit order
1MHz (Max)	MODE 3	high	rising edge	MSB first

※ SPI 통신은 설정에 따라 4가지로 나뉩니다. 그 중 Mode 3 을 사용하세요.

#### 4.1.2. 온도 데이터 읽기

- ※ Request 주기 : 반드시 최소 20ms 이상
- ※ 전원 공급 후 첫 Request 시간 : 최소 200 ms 이후

※ Object & Sensor 온도 읽기 timing



※ 오실로스코프 캡처 화면



### 4.1.3. SPI 통신 모델 온도 계산 방법(필독)

object	0x6D	0x01	sensor	0xFA	0x00
--------	------	------	--------	------	------

\* **object 온도 계산** : 상위 Byte(0x01) + 하위 Byte(0x6D) = 0x016D (Hex)  
=> 365(dec) 이며 이 값을 10으로 나누면 36.5 도입니다.

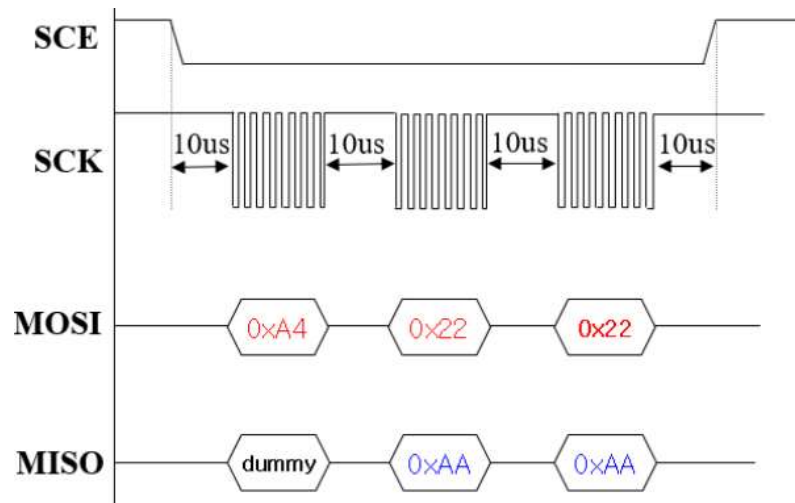
\* **sensor 온도 계산** : 상위 Byte(0x00) + 하위 Byte(0xFA) = 0x00FA  
=>250 (dec) 즉, 25.0 도입니다.

※ 만약 온도 데이터가 영하일 경우는 2의 보수 값으로 표현됩니다.

object	0xF1	0xFF
--------	------	------

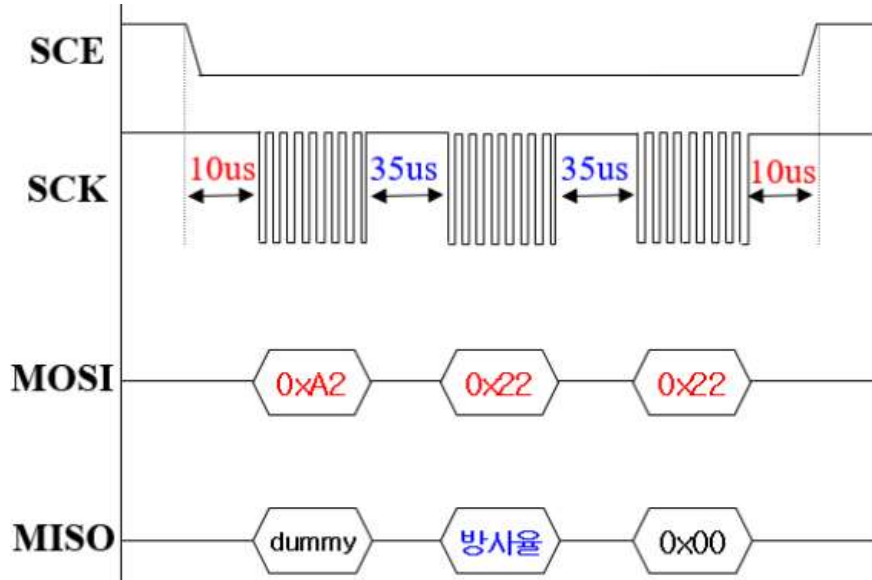
상위 Byte(0xFF) + 하위 Byte(0xF1) = 0xFFFF1 (Hex)  
0xFFFF1 의 2의 보수 = 0x000F = 15 즉, -1.5 도를 의미합니다.

### 4.1.4. 레이저 켜기



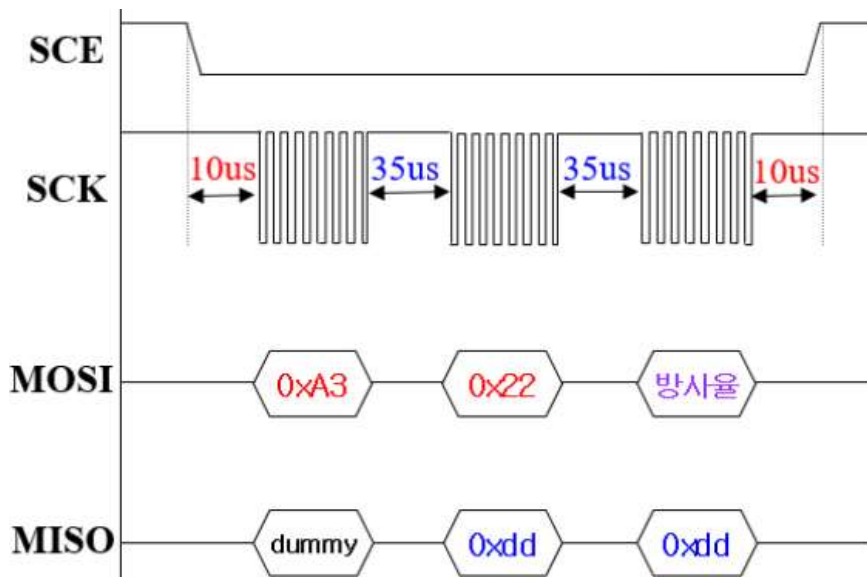
※ 레이저는 약 15-20초간 동작 후 자동으로 꺼집니다.

#### 4.1.5. 방사율 읽기



※ “방사율” 데이터가 97 인 경우 : 0.97 의 방사율을 의미합니다.

#### 4.1.6. 방사율 쓰기



※ 방사율(default 0.97) 변경은 0.1~ 1.0 까지만 가능합니다.

방사율 수치에 100을 곱한 값을 전송하면 됩니다.

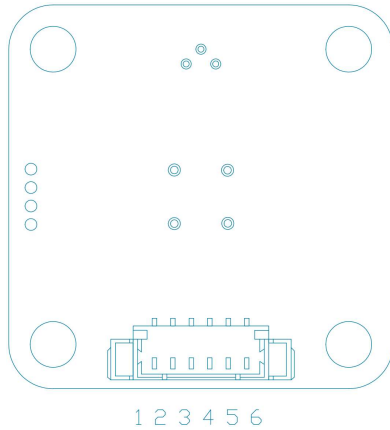
예) 0.97 로 변경하고자 하는 경우 :  $0.97 * 100 = 97$

※ 방사율은 한번 write 할 경우 전원이 리셋 돼도 유지가 됩니다.

※ 메인문의 반복문이나 전원 켜 때마다 명령을 수행할 필요가 전혀 없습니다.

※ 반드시 SCK의 byte 통신간 **최소 35us** 의 지연시간이 필요합니다.

### 4.1.7. Pin Description



Pin Number	통신 방식
	SPI
1	GND
2	SCE
3	SCK
4	MOSI(SDI)
5	MISO(SDO)
6	3.3V

- ※ 위 핀 번호는 SPI 통신 모델에만 적용됩니다.
- ※ Connector 정보  
 Header : molex 0533980671 (pcb 실장(實裝) 하우징)  
 Female : molex 51021-0600

### 4.1.8. 프로그래머 주의 사항.

- ※ 방사율은 변경 할 경우 전원이 리셋 돼도 값을 유지합니다. 따라서 컨트롤러 메인문의 while문 또는 초기화마다 매번 명령을 수행토록 코딩 하면 안됩니다.
- ※ 제품 전원 공급 후 최소 200ms 이후에 통신을 시작하십시오.



## 4.2. RS-485

### 4.2.1. 통신 규격

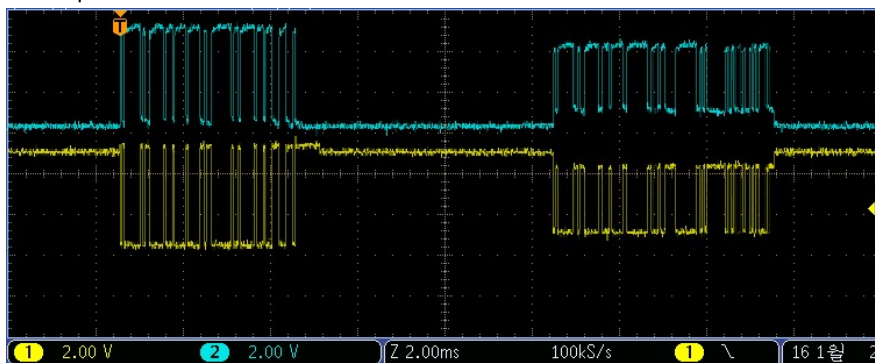
- ※ Half-duplex, RS485 Multi-Drop
- ※ Modbus 485-RTU Protocol 지원

통신속도(bps)	DATA bit	Stop bit
<b>19200</b>	8	1

### 4.2.2. 온도 데이터 읽기

#### 4.2.2.1. Modbus register(Modbus Function 03)

- ※ Request 주기 : 반드시 최소 50ms 이상
- ※ 전원 공급 후 첫 Request 시간 : 최소 200 ms 이후
- ※ Request 후 응답 지연 시간 : 약 5ms



※ 위 파형은 테스트키트와 DTPML-485 제품간의 통신 파형입니다.

Request ( Master → DTPML-485-XXX )			Response ( DTPML-485-XXX → Master )		
Field Name	Hex	DEC	Field Name	Hex	DEC
ID(default 0x01)	0x01~0xc8	1~200	ID(default 0x01)	0x01~0xc8	1~200
Function	0x03	3	Function	0x03	3
Starting Address Hi	0x04	1200	Byte Count	0x04	4
Starting Address Lo	0xB0		Data Value 1 Hi	대상온도	
No. of Data Hi	0x00	2	Data Value 1 Lo		
No. of Data Lo	0x02		Data Value 2 Hi	센서온도	
CRC (19p~20p 참고)	데이터에 따라 바뀜.		Data Value 2 Lo		
CRC			CRC	데이터에 따라 바뀜.	
			CRC		
8 Byte Request			9 Byte Response		

※ No. of Data는 항상 2로 요청해야 합니다.

### 4.2.2.2. 485 통신 모델 온도 계산 방법(필독)

응답한 데이터	HEX	DEC
Data Value 1 Hi	0x016D	365
Data Value 1 Lo		
Data Value 2 Hi	0x00FA	250
Data Value 2 Lo		

대상온도 : 0x016D = 365 --> 36.5도

센서온도 : 0x00FA = 250 --> 25.0도

※ 만약 온도 데이터가 영하일 경우는 2의 보수 값으로 표현됩니다.

응답한 데이터	HEX	DEC
Data Value 1 Hi	0xFFF1	
Data Value 1 Lo		

0xFFF1 -> (2의보수) -> 0x000F = 15 즉, -1.5도

### 4.2.3. ID 변경하기 ( 반드시 마스터와 1:1 로 연결한 상태에서만 진행해야 합니다. )

#### 4.2.3.1. Modbus register(Modbus Function 06)

Request ( Master → DTPML-485-XXX )			Response ( DTPML-485-XXX → Master )		
Field Name	Hex	DEC	Field Name	Hex	DEC
ID	0xFF	255	ID	0xFF	255
Function	0x06	6	Function	0x06	6
Register Address Hi	0x03	1000	Register Address Hi	0x03	1000
Register Address Lo	0xE8		Register Address Lo	0xE8	
Register Value Hi	0x00	1~200	Register Value Hi	0x00	1~200
Register Value Lo	변경할 ID		Register Value Lo	변경한 ID	
CRC			CRC		
CRC					
8 Byte Request			7 Byte Response		

변경할 ID 는 1~200 까지 설정이 가능합니다. 범위를 초과하는 ID 를 설정하면, ID 는 1 로 초기화 됩니다. Default ID 는 1 이기 때문에 1:1 사용시에는 변경할 필요가 없습니다.

#### 4.2.4. 방사율 읽기

##### 4.2.4.1. Modbus register(Modbus Function 04)

Request ( Master → DTPML-485-XXX )			Response (DTPML-485-XXX → Master )		
Field Name	Hex	DEC	Field Name	Hex	DEC
ID	0x01~0xc8	1~200	ID	0x01~0xc8	1~200
Function	0x04	4	Function	0x04	4
Starting Address Hi	0x03	800	Byte Count	0x02	2
Starting Address Lo	0x20		방사율 Hi	0x00	0
Quantity of Input Reg. Hi	0x00	1	방사율 Lo	0x0A~0x64	10~100
Quantity of Input Reg. Lo	0x01		CRC		
CRC			CRC		
CRC					
8 Byte Request			7 Byte Response		

방사율은 default 0.97이며 100을 곱한 값이 전송됩니다.

만약, 방사율 Lo값이 97이면 0.97을 의미합니다.

#### 4.2.5. 방사율 변경

##### 4.2.5.1. Modbus register(Modbus Function 06)

Request ( Master → DTPML-485-XXX )			Response (DTPML-485-XXX → Master )		
Field Name	Hex	DEC	Field Name	Hex	DEC
ID	0x01~0xc8	1~200	ID	0x01~0xc8	1~200
Function	0x06	6	Function	0x06	6
Register Address Hi	0x03	800	Register Address Hi	0x03	800
Register Address Lo	0x20		Register Address Lo	0x20	
Register Value Hi	0x00	10~100	Register Value Hi	0x00	10~100
Register Value Lo	방사율		Register Value Lo	방사율	
CRC			CRC		
CRC			CRC		
8 Byte Request			8 Byte Response		

"방사율" 은 10~100 까지 변경이 가능합니다. (실제 방사율 0.1 ~ 1.0 에 100을 곱한 값)

#### 4.2.6. Laser ON

##### 4.2.6.1. Modbus register(Modbus Function 06)

Request ( Master → DTPML-485-XXX )			Response (DTPML-485-XXX → Master )		
Field Name	Hex	DEC	Field Name	Hex	DEC
ID	0x01~0xc8	1~200	ID	0x01~0xc8	1~200
Function	0x06	6	Function	0x06	6
Register Address Hi	0x04	1100	Register Address Hi	0x04	1100
Register Address Lo	0x4C		Register Address Lo	0x4C	
Register Value Hi	0x00	1	Register Value Hi	0x00	1
Register Value Lo	0x01		Register Value Lo	0x01	
CRC			CRC		
CRC			CRC		
8 Byte Request			8 Byte Response		

※ 레이저를 눈으로 바라보지 마십시오.

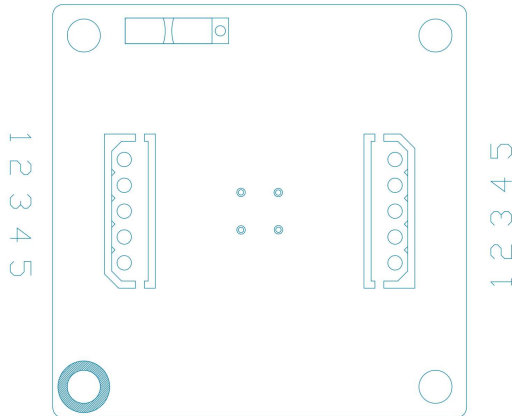
※ 레이저는 명령 후 약 15~20 초간 동작합니다. 레이저가 꺼지기 전에 반복 명령을 주면 마지막 Command 을 기준으로 15~20 초간 동작합니다.

※ 레이저 포인터는 측정 대상이 멀어질수록 포인트 오차가 더욱 커질 수 있습니다. 가리키는 방향을 참고하시기 바랍니다.

※ 온도 데이터를 반복적으로 Read 하는 경우, 레이저 명령을 보낼 시 타이밍에 유의 하십시오. 마스터에서 온도 데이터가 수신이 완료 된 후에 레이저 명령을 보내야 합니다.

## Communication Protocol

### 4.2.7. Pin Description



Pin Number	통신 방식
	RS-485
1	PE
2	D+
3	D-
4	GND
5	5V

※ 위 핀 번호는 485 통신 모델에만 적용됩니다.

### 4.2.8. 프로그래머 주의 사항.

- ※ 방사율 변경 메시지는 메인 함수의 반복문에 넣지 마십시오.  
방사율 변경이 필요할 때 한번만 실행하면 됩니다. 전원을 리셋 해도 값은 유지됩니다.
- ※ 제품 전원 공급 후 최소 200ms 이후에 통신을 시작하십시오.
- ※ 485 통신chip은 메시지를 요청할 경우를 제외하고는 항상 입력 상태로 유지하십시오.

### 4.2.9. 모드 버스(Modbus)란?

생산 현장에서는 여러 장비들이 존재합니다. 모든 장비들을 한 회사에서 만들었다면 문제가 없겠지만 현실은 그렇지 않습니다. 장비도 다르고 환경도 다른 제 각각인 상황이 발생합니다. 따라서 일련의 약속을 통해 장비들을 제어 하는 통신 방법이 요구되었으며, 이런 통신 방법을 필드버스(Fieldbus) 라고 정의합니다. 필드버스에는 여러 기술들이 있습니다. 익히 알려진 CAN, Profibus, EtherCAT 등이 있으며, 본 TB-485 에 적용된 모드버스(Modbus) 또한 표준으로 정의돼 있습니다.

RS-485 통신을 사용하며, 마스터/ 슬레이브로 구분되어 연결된 모든 장비의 정보를 모니터링 할 수 있습니다.(환경에 따라 동시 연결 장비 수는 제한됩니다) 마스터는 한대만 가능하며, 질의 응답 구조로 해당하는 장비만 응답을 하게 됩니다. 또한 오픈 프로토콜이기 때문에 무료로 사용이 가능합니다.

## Communication Protocol

### 4.2.10. CRC 설명 / 연산 과정 예제

CRC(Cyclic Redundancy Check) 는 데이터 통신시 전송 데이터에 오류가 있는지 확인 하기 위해 체크값을 결정하는 방식입니다. 모든 Request/ Response 메시지에 2byte 크기의 CRC 값이 붙어서 전송됩니다. 전송된 데이터 값으로 연산한 CRC 값과 수신된 CRC 값을 비교하여 오류 여부를 검증합니다. Polynomial 값은 0xA001 이지만, 아래 예제 코드는 비트단위 연산 부하를 줄이기 위해 CRC 테이블을 활용하는 예제 입니다. 요즘 시스템들은 메모리가 작지 않기 때문에 테이블 적용을 추천합니다.

=====

※ CRC 계산 예제 코드.

```
const uint16_t TableCRC16[256]={
    0x0000, 0xC0C1, 0xC181, 0x0140, 0xC301, 0x03C0, 0x0280, 0xC241, 0xC601, 0x06C0,
    0x0780, 0xC741, 0x0500, 0xC5C1, 0xC481, 0x0440, 0xCC01, 0x0CC0, 0x0D80, 0xCD41,
    0x0F00, 0xCFC1, 0xCE81, 0x0E40, 0x0A00, 0xCAC1, 0xCB81, 0x0B40, 0xC901, 0x09C0,
    0x0880, 0xC841, 0xD801, 0x18C0, 0x1980, 0xD941, 0x1B00, 0xDBC1, 0xDA81, 0x1A40,
    0x1E00, 0xDEC1, 0xDF81, 0x1F40, 0xDD01, 0x1DC0, 0x1C80, 0xDC41, 0x1400, 0xD4C1,
    0xD581, 0x1540, 0xD701, 0x17C0, 0x1680, 0xD641, 0xD201, 0x12C0, 0x1380, 0xD341,
    0x1100, 0xD1C1, 0xD081, 0x1040, 0xF001, 0x30C0, 0x3180, 0xF141, 0x3300, 0xF3C1,
    0xF281, 0x3240, 0x3600, 0xF6C1, 0xF781, 0x3740, 0xF501, 0x35C0, 0x3480, 0xF441,
    0x3C00, 0xFCC1, 0xFD81, 0x3D40, 0xFF01, 0x3FC0, 0x3E80, 0xFE41, 0xFA01, 0x3AC0,
    0x3B80, 0xFB41, 0x3900, 0xF9C1, 0xF881, 0x3840, 0x2800, 0xE8C1, 0xE981, 0x2940,
    0xEB01, 0x2BC0, 0x2A80, 0xEA41, 0xEE01, 0x2EC0, 0x2F80, 0xEF41, 0x2D00, 0xEDC1,
    0xEC81, 0x2C40, 0xE401, 0x24C0, 0x2580, 0xE541, 0x2700, 0xE7C1, 0xE681, 0x2640,
    0x2200, 0xE2C1, 0xE381, 0x2340, 0xE101, 0x21C0, 0x2080, 0xE041, 0xA001, 0x60C0,
    0x6180, 0xA141, 0x6300, 0xA3C1, 0xA281, 0x6240, 0x6600, 0xA6C1, 0xA781, 0x6740,
    0xA501, 0x65C0, 0x6480, 0xA441, 0x6C00, 0xACC1, 0xAD81, 0x6D40, 0xAF01, 0x6FC0,
    0x6E80, 0xAE41, 0xAA01, 0x6AC0, 0x6B80, 0xAB41, 0x6900, 0xA9C1, 0xA881, 0x6840,
    0x7800, 0xB8C1, 0xB981, 0x7940, 0xBB01, 0x7BC0, 0x7A80, 0xBA41, 0xBE01, 0x7EC0,
    0x7F80, 0xBF41, 0x7D00, 0xBDC1, 0xBC81, 0x7C40, 0xB401, 0x74C0, 0x7580, 0xB541,
    0x7700, 0xB7C1, 0xB681, 0x7640, 0x7200, 0xB2C1, 0xB381, 0x7340, 0xB101, 0x71C0,
    0x7080, 0xB041, 0x5000, 0x90C1, 0x9181, 0x5140, 0x9301, 0x53C0, 0x5280, 0x9241,
    0x9601, 0x56C0, 0x5780, 0x9741, 0x5500, 0x95C1, 0x9481, 0x5440, 0x9C01, 0x5CC0,
    0x5D80, 0x9D41, 0x5F00, 0x9FC1, 0x9E81, 0x5E40, 0x5A00, 0x9AC1, 0x9B81, 0x5B40,
    0x9901, 0x59C0, 0x5880, 0x9841, 0x8801, 0x48C0, 0x4980, 0x8941, 0x4B00, 0x8BC1,
    0x8A81, 0x4A40, 0x4E00, 0x8EC1, 0x8F81, 0x4F40, 0x8D01, 0x4DC0, 0x4C80, 0x8C41,
    0x4400, 0x84C1, 0x8581, 0x4540, 0x8701, 0x47C0, 0x4680, 0x8641, 0x8201, 0x42C0,
    0x4380, 0x8341, 0x4100, 0x81C1, 0x8081, 0x4040
};

uint16_t CalcCRC16(uint8_t *pdata, uint16_t DataLen)
{
    uint16_t AccumCRC16 = 0xFFFF;
    uint8_t i, j;
    for(j=0; j<DataLen; j++)
    {
        i = (AccumCRC16 ^ *(pdata++) ) & 0xFF;
        AccumCRC16 = ((AccumCRC16>>8) ^ TableCRC16[i]) & 0xFFFF;
    }
    return AccumCRC16;
}
```

### ※ 온도 읽는 명령 CRC 구하는 과정

온도 읽는 명령의 CRC 값을 구해보도록 하겠습니다.

먼저 데이터를 전송할 적당한 크기의 전역 변수를 선언합니다.

```
uint8_t RequestData[8];          // 전송할 배열
uint16_t crc;                    // 연산한 crc값 임시 저장 변수
```

“4.3.2 온도데이터 읽기”를 참고하여 메인 함수에서 배열 요소에 값을 할당합니다.

```
RequestData[0] = 0x01;
RequestData[1] = 0x03;
RequestData[2] = 0x04;
RequestData[3] = 0xB0;
RequestData[4] = 0x00;
RequestData[5] = 0x02;
```

여기까지가 보낼 데이터이며, 여기에 CRC 2byte를 계산하여 추가해야 합니다.

앞페이지에서의 CalcCRC16 함수를 사용합니다.

```
crc = CalcCRC16(RequestData, 6); // RequestData 배열 0~5 까지, 6 바이트 데이터를 연산합니다.
```

```
RequestData[6] = (unsigned char)((crc >> 0) & 0x00FF); // CRC 값의 하위 바이트
RequestData[7] = (unsigned char)((crc >> 8) & 0x00FF); // CRC 값의 상위 바이트
```

연산을 거친 crc 값은 아래와 같습니다.

```
RequestData[6] 변수에는 0xC4 값이 저장됩니다.
RequestData[7] 변수에는 0xDC 값이 저장됩니다.
```

CRC를 포함한 전송할 데이터 준비는 끝났으며, RequestData 배열 0~7 의 값을 전송하면 됩니다.

수신 측에서는 (RequestData[6], RequestData[7] 의 데이터) 와 (수신받은 데이터의 crc 연산값)을 비교한 후, 일치하면 응답, 일치하지 않으면 응답을 하지 않습니다.

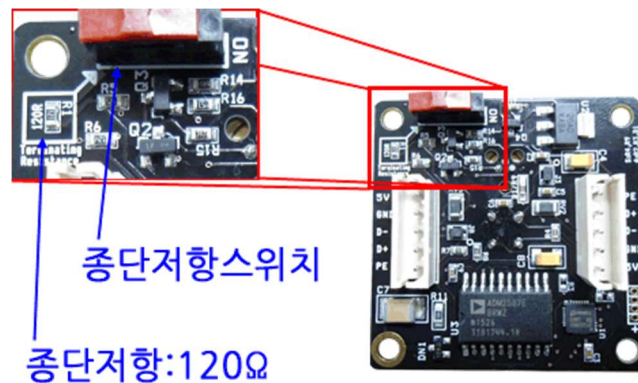
반대로 마스터는 센서로부터 수신된 데이터를 위와 같은 연산을 거쳐, 데이터에 오류가 없는지 검증한 후 온도 데이터를 활용하면 됩니다.

### ※ 온도 요청 데이터(CRC 포함) 예제

```
ID1번 : 0x01, 0x03, 0x04, 0xB0, 0x00, 0x02, (0xC4), (0xDC)
ID2번 : 0x02, 0x03, 0x04, 0xB0, 0x00, 0x02, (0xC4), (0xEF)
ID3번 : 0x03, 0x03, 0x04, 0xB0, 0x00, 0x02, (0xC5), (0x3E)
```

### 4.2.11. 종단 저항 스위치 (전원 스위치가 아닙니다)

RS485 의 통신에 있어서 지연 등에 의한 왜곡과 감쇠 방지를 목적으로 선로의 임피던스 매칭을 위해 저항을 삽입합니다. 이를 종단저항이라 합니다. DTPML-485 모듈에는 120옴의 기본 종단저항이 적용돼 있으며 간단히 스위치의 조작을 통해 종단저항 적용을 ON/OFF 시킬 수 있습니다.



< 종단 저항 미적용 >



<종단 저항 적용 >

#### ※ 주의 사항

1. 종단 저항은 항상 120옴은 아닙니다. 상황에 따라 저항을 바꿔줘야 할 상황이 있을 수 있습니다. 저항을 바꿀 때에는 위 그림에서 종단저항 120옴 부분의 저항을 바꿔 주시면 됩니다.
2. 종단 저항 교체시 **90옴 이하의 저항은 사용하지 마십시오.**
3. 통신선로 상에서 **2개 이상의 종단 저항을 ON** 하면 안됩니다.
4. 제품 출고시 기본 종단 저항 스위치가 OFF 상태로 출고 됩니다.



## ▶ Additional Information

- manufacturer : Diwell Electronics Co., Ltd. <(주)디웰전자>
- Homepage : [www.diwell.com](http://www.diwell.com)
- shopping mall : [www.diwellshop.com](http://www.diwellshop.com)
- Phone : +82-70-8235-0820
- Fax : +82-31-429-0821
- Technical support : [expoeb2@diwell.com](mailto:expoeb2@diwell.com), [dsjeong@diwell.com](mailto:dsjeong@diwell.com)

## ▶ Revision History

Version	Date	Description
1.0	2020-01-02	First version is released.
1.1	2020-04-23	RS-485 통신 모델 Laser Protocol 누락에 따른 추가.
1.2	2020-11-13	SPI 모델 커넥터 정보 추가.