

- 비접촉 온도 측정
- 원거리 온도 측정
- 16 x 4 Thermopile Array
- High Accuracy
- Digital Interface :



RS485 MODBUS RTU ( Isolation Interface )

※ 본 제품은 DTPAL-485-1604를 대체합니다. 통신 프로토콜은 기존 제품과 완벽히 호환됩니다.  
주요 변경 사항 : FOV 변경, 센서 길이 변경, 레이저 삭제

### ▶ 제품 설명

- DTPA-485-1604N는 16 x 4 개의 Array 센서를 내장하여 동시에 64개 픽셀의 대상 온도를 얻을 수 있습니다.
- DTPA-485-1604N를 이용하여 열화상 이미지 구현이 가능합니다.
- DTPA-485-1604N는 측정 대상에서 고열 분포 위치를 쉽게 알 수 있습니다.
- DTPA-485-1604N는 접촉을 하지 않고 원하는 물체 표면의 온도를 250ms 이내에 정확하게 측정할 수 있는 온도센서모듈입니다.
- DTPA-485-1604N는 온도계산 프로세서를 내장하고 있어 정확한 온도 값을 출력합니다. (Master Controller에 온도계산 알고리즘이 필요하지 않습니다.)
- DTPA-485-1604N는 디지털 통신(RS485)으로 온도를 출력합니다.
- 레이저 포인터를 장착하여 측정 방향을 쉽게 알 수 있습니다.
- 센서 온도와 대상 온도를 동시에 측정합니다.

### ▶ 특징

- 측정 온도 구간 : -30°C ~ 300°C
- 동작 온도 구간 : -20°C ~ 70°C
- 분해능 : 0.1°C
- FOV (가로:53°, 세로:15°)
- 정확도 : ±2% (동작온도 : 25°C)
- 입력 전압 : 5V
- 통신 인터페이스 : RS485 MODBUS RTU

### ▶ 응용분야

- 열화상 과열방지 시스템
- 산업용 온도 측정 장치
- 배전반, 분전반 과열 감지용
- 사람의 위치 판별이 필요한 환경
- 가전기기
- PCB 내의 발열 지점 확인

## ▶ Absolute Maximum Ratings

- Absolute Maximum Rating 값을 초과하는 조건에서 제품을 동작시킬 경우 치명적인 손상을 가할 수 있습니다.

Parameter	Symbol	Conditions	min	Typ	Max	Unit
Supply Voltage	Vcc	Measured Versus GND	-0.2		6.0	V
Storage temperature	Tstor		-40		85	°C
Operating temperature	Top		-20		70	°C

## ▶ Electrical Requirements

Parameter	Symbol	Conditions	min	Typ	Max	Unit
공급전압	Vcc	Measured versus GND	4.8	5	5.2	V
방사율(Emission Coefficient)	$\epsilon$			0.97		
소비 전류 (5V 기준)		Full ambient temp. range, Typical value, no output load		49.20		mA
		On Laser		60.20		mA

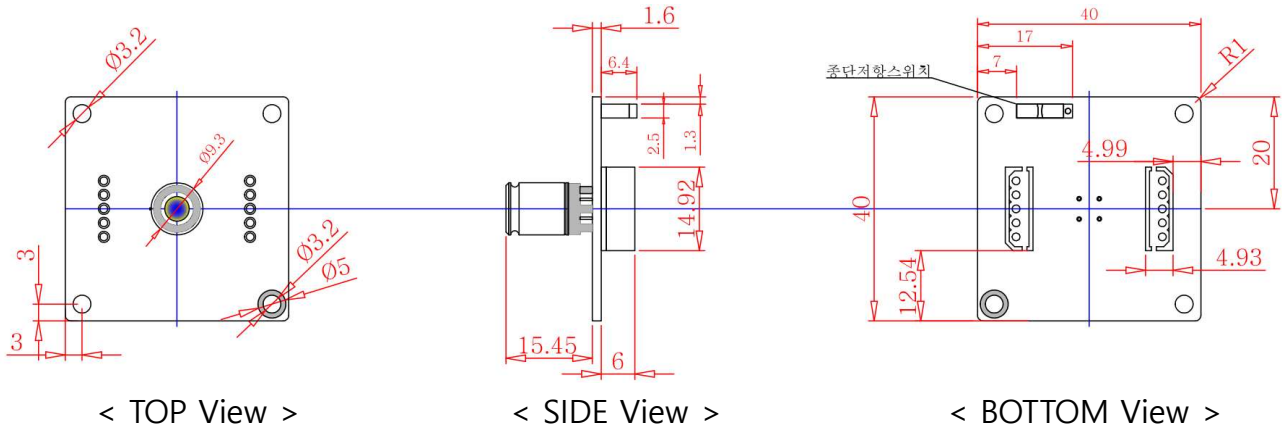
## ▶ Operational Characteristics

- if not otherwise noted, 25°C ambient temperature, 5V supply voltage and object with  $\epsilon = 0.97$  were applied

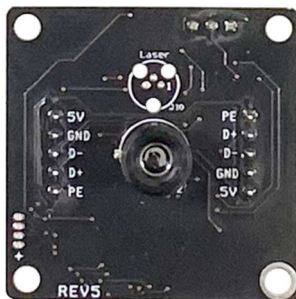
Parameter	Symbol	Conditions	min	Typ	Max	Unit
Field of View	FOV	16 Sensors(가로)		53		degree
		4 Sensors(세로)		15		
온도측정범위	Tobj		-30		300	°C
동작온도(주변온도)	Tamb		-20		70	°C
온도측정 시간	Fout			250		msec
정확도	AccT			±2		%
Resolution Digital				0.1		°C
Standard Start-UP Time	tStart			3		sec
Stabilization Time	tStab			1		min

▶ Mechanical Dimensions

단위 : mm



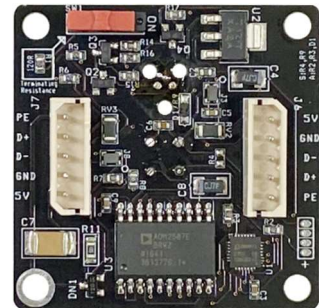
▶ 제품 사진



< TOP View >



< SIDE View >



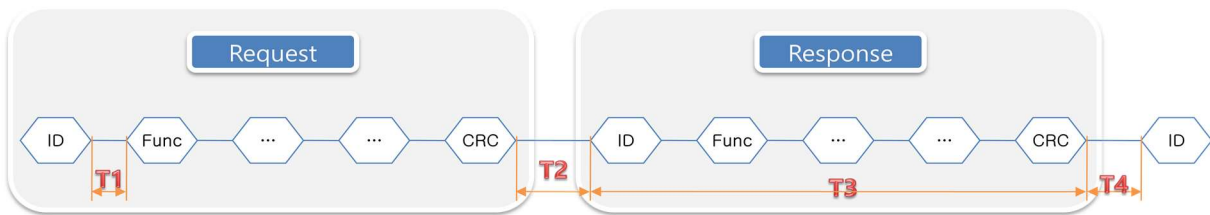
< BOTTOM View >

## ▶ MODBUS 485-RTU Protocol

### 1. 통신 규격

- 통신버스 : RS485 Multi-Drop, Half duplex
- 통신속도 : 19200 bps
- DATA : 8 bit
- Parity : NONE
- STOP : 1 bit

### 2. 송수신 시퀀스(필독!)



T1 : 캐릭터간의 지연시간

T2 : 요구 텔레그램과 응답 텔레그램 간의 지연시간

T3 : 응답 프레임 전체시간

T4 : 한 Frame(Request->Response) 완료 후 다음 Frame 시작간의 지연시간

Baud rate	T1	T2	T3	T4
19200	<1ms	5 ms <b>(Note. 1)</b>	76ms	지연시간 <b>(Note.2)</b>

**Note 1.** 응답지연시간이 약 5ms로 항상 일정합니다.

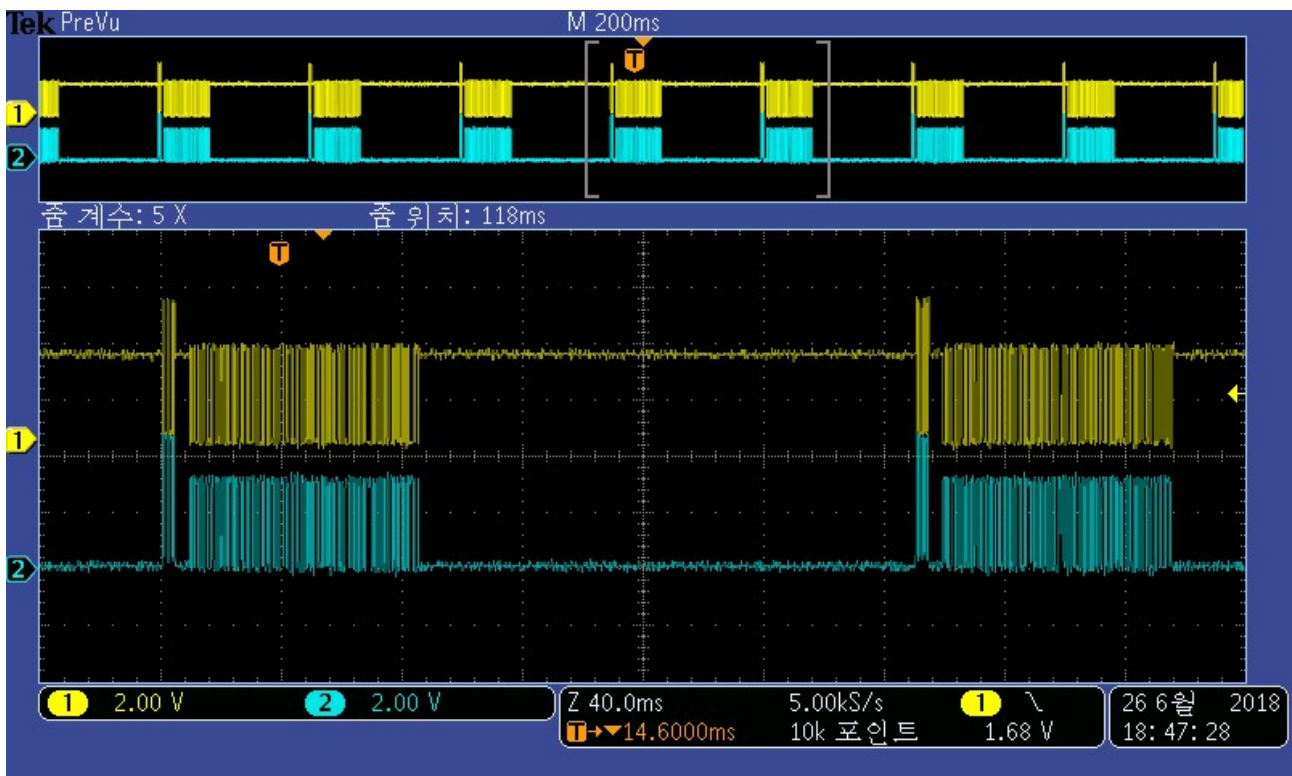
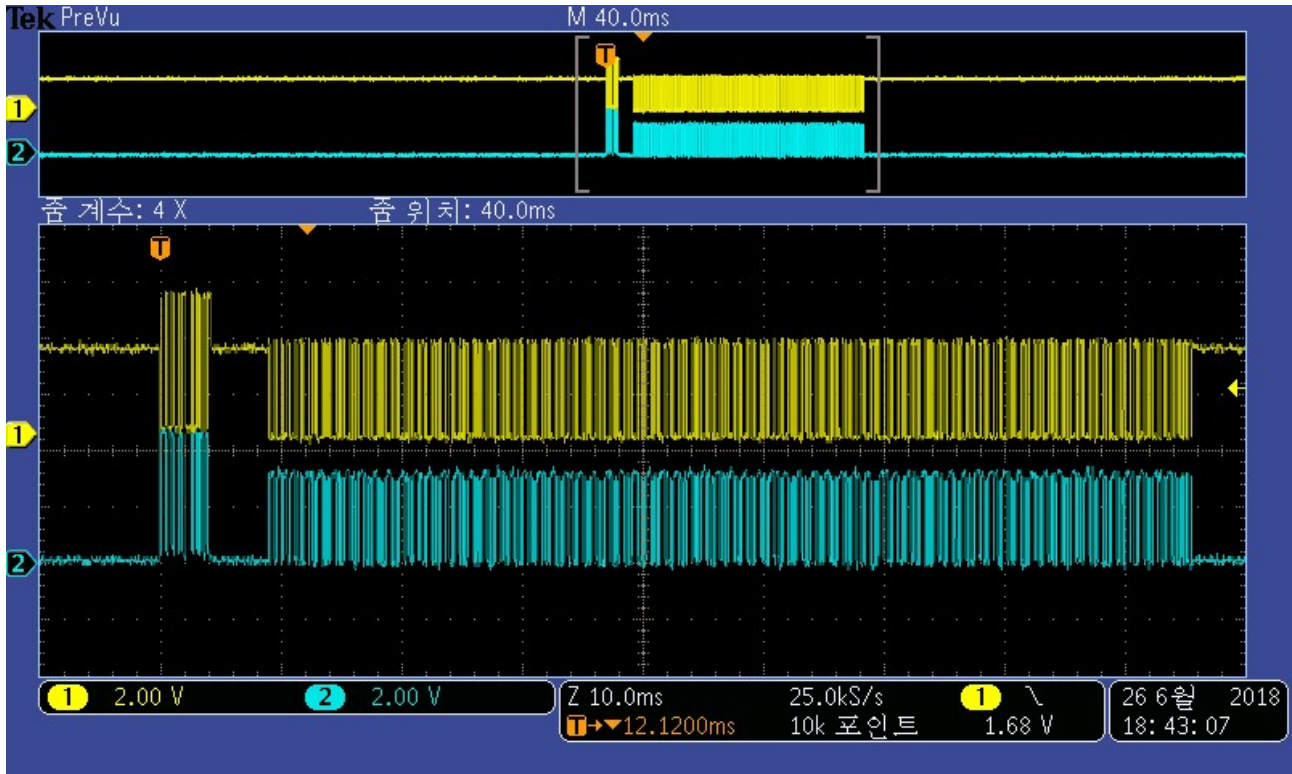
**Note 2.**  $T4 = 250ms - (request\ time + T2 + T3)$  : 4Hz 기준

예1)  $250 - (5 + 5 + 76) =$  최소 164ms 이상

※ T3 구간에 Request 명령이 겹치면 통신 칩이 손상될 수 있으니 주의바랍니다.

※ 반드시 T4 시간을 지켜야 안정적인 통신이 가능합니다.

통신 파형 캡처 화면



### 3. Request/ Response (Function 0x03) - 온도값 Read 예제

Request ( Master → DTPA-485-1604N)			Response ( DTPA-485-1604N → Master )		
Field Name	Hex	DEC	Field Name	Hex	DEC
① ID	0x01~0xc8	1~200	① ID	0x01~0xc8	1~200
Function	0x03	3	Function	0x03	3
Starting Address Hi	0x01	500	Byte Count	0x82	130
Starting Address Lo	0xF4		Data Value 1 Hi	센서온도	
No. of Data Hi	0x00	65	Data Value 1 Lo		
No. of Data Lo	0x41		Data Value 2 Hi	대상픽셀(1,0)	
CRC			Data Value 2 Lo		
CRC			Data Value 3 Hi	대상픽셀(3,15)	
			Data Value 3 Lo		
			...	...	
			Data Value 64 Hi		
			Data Value 64 Lo	대상픽셀(3,15)	
			Data Value 65 Hi		
			Data Value 65 Lo	CRC	
			CRC		
			CRC	8 Byte Request	135 Byte Response

① ID는 출고시 0x01로 설정 돼 있습니다. ID를 바꾸려면 다음 페이지의 ID Write 를 참고하세요.  
ID값은 1~200 까지 가능합니다.

※ 주소값 500 과 데이터수량 65는 고정값입니다.

※ 응답 Data Value의 첫 데이터는 센서온도가 나오며, 이어서 픽셀별 대상온도가 출력됩니다.

#### 4. Request/ Response (Function 0x06) - ID Write 예제

Request ( Master → DTPA-485-1604N)			Response ( DTPA-485-1604N → Master )		
Field Name	Hex	DEC	Field Name	Hex	DEC
② 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	③ ID	Register Value Hi	0x00	ID
Register Value Lo	③ ID		Register Value Lo	ID	
CRC			CRC		
CRC			CRC		
8 Byte Request			8 Byte Response		

② 번 ID 위치에는 0xff값을 써야 ID write 를 인식합니다. ③ ID에는 쓰고자 하는 ID를 넣으면 됩니다.

※ ID 변경시 반드시 회로에 모듈 1대만 연결해야 합니다.

#### 5. 온도 계산법

##### 예제1) 영상 온도

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

##### 예제2) 영하 온도

응답한 데이터	HEX	DEC
Data Value 1 Hi	0xFFF1	
Data Value 1 Lo		
Data Value 2 Hi	0xFF9C	
Data Value 2 Lo		

대상온도 : 0xFFF1 -> (2의보수연산) -> 0x000F = 15 즉, -1.5도

센서온도 : 0xFF9C -> (2의보수연산) -> 0x0064 = 100 즉, -10.0도

## 6. 통신 예제

### 6-1. 온도 읽기

ID 1번의 500(0x01F4)번지부터 65개의 레지스터값을 얻으려면 다음과 같습니다.

Request Command : 0x01,0x03,0x01,0xF4,0x00,0x41,(0xC5),(0xF4)

Response : 0x01, 0x03, 0x82, 주변온도 Hi, 주변온도 Lo, 픽셀(0,0) Hi, ... CRC,CRC

※ 레지스터 수량은 65개(0x41)로 고정입니다.

※ 마지막 2byte(CRC) 부분은 7.CRC 파트를 참고하세요.

### 6-2. ID Write

ID 1 쓰기 : 0xFF, 0x06, 0x03, 0xE8, 0x00, 0x01, (0xDD),(0xA4)

ID 2 쓰기 : 0xFF, 0x06, 0x03, 0xE8, 0x00, 0x02, (0x9D),(0xA5)

ID 3 쓰기 : 0xFF, 0x06, 0x03, 0xE8, 0x00, 0x03, (0x5C),(0x65)

ID 4 쓰기 : 0xFF, 0x06, 0x03, 0xE8, 0x00, 0x04, (0x1D),(0xA7)

※ PLC와 연결했는데, 통신 응답이 없을 경우 확인할 사항.

1. 이 방법은 회로 연결에는 문제가 없을 경우에만 유효합니다.
2. Modbus 의 FUNCTION 이 03(Read Holding Register)일 경우는 Starting Address 는 40,000부터 시작을 하는 경우가 많습니다. 하지만 본 제품의 Starting Address는 500 입니다. 사용하시는 장비에 따라 500이라고 설정을 해도 40,000을 옵션으로 가져가서 40,500 으로 설정되는 경우가 있습니다. 그러면 센서에서는 응답을 하지 않습니다.
3. 해당 PLC 제조사에 FUNCTION 03 에 대한 default 주소를 40,000 으로 하지 않도록 하는 방법을 요청하시기 바랍니다. 즉, 실제 전송되는 주소는 500이 되어야 합니다.



### 7. CRC란? (개발자 분은 반드시 아래 내용을 확인해서 CRC 값을 적용해야 합니다.)

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. 각 전송/ 수신 데이터에 따라 CRC값이 맞는지 반드시 검증 한 후 데이터를 활용해야 합니다.

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 예제.

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
```

본문의 온도 읽는 프로토콜을 참고하여 메인 함수의 배열 요소에 값을 할당합니다.

```
RequestData[0] = 0x01;  
RequestData[1] = 0x03;  
RequestData[2] = 0x01;  
RequestData[3] = 0xF4;  
RequestData[4] = 0x00;  
RequestData[5] = 0x41;
```

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

이전 페이지의 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.

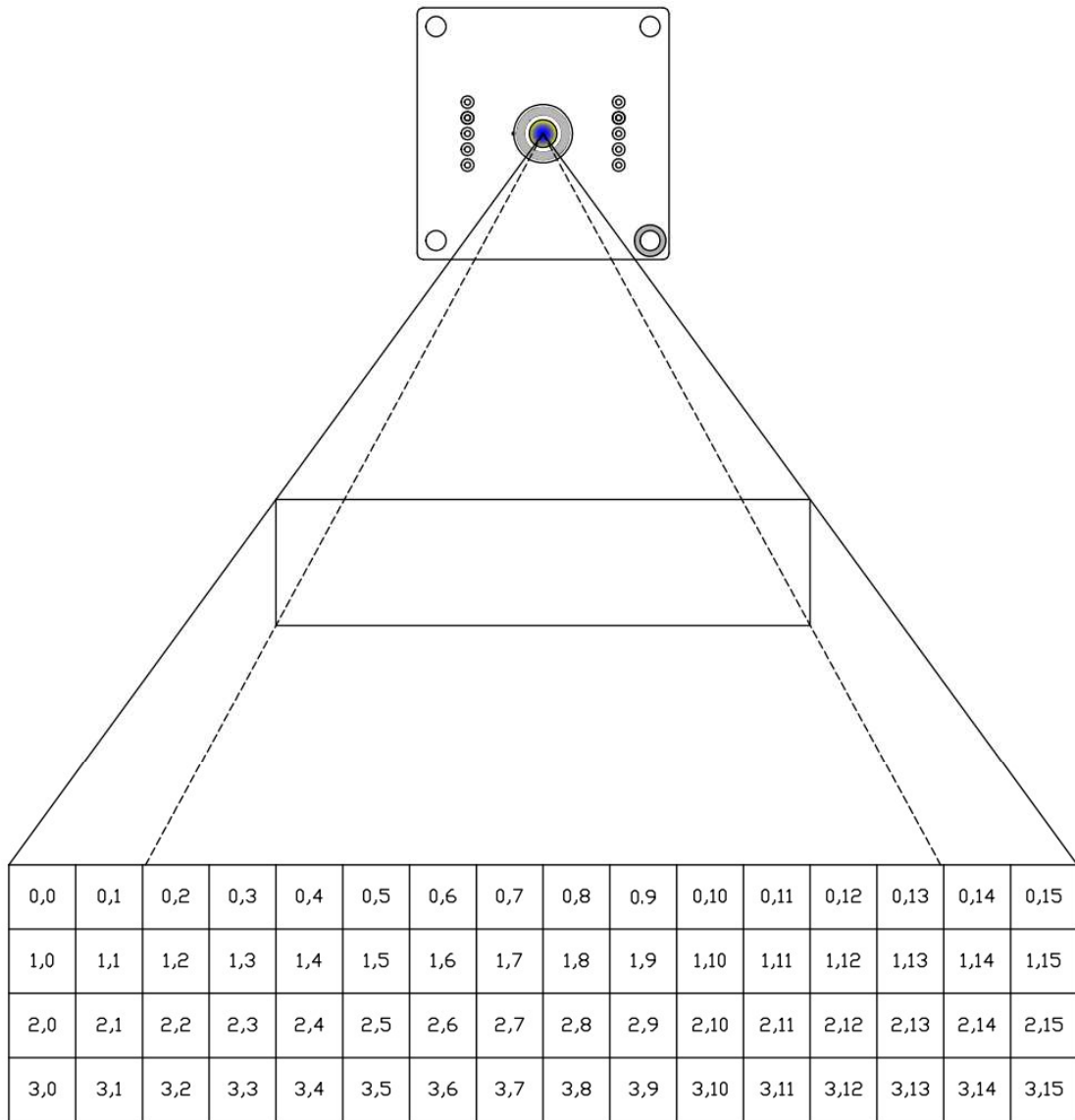
```
crc = 0xF4C5 (dec : 62,661)  
RequestData[6] 변수에는 0xC5 값이 저장됩니다. (CRC 값의 하위 바이트)  
RequestData[7] 변수에는 0xF4 값이 저장됩니다. (CRC 값의 상위 바이트)
```

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

수신 측(모듈)에서는 (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 반드시 검증한 후 온도 데이터를 활용해야 합니다.

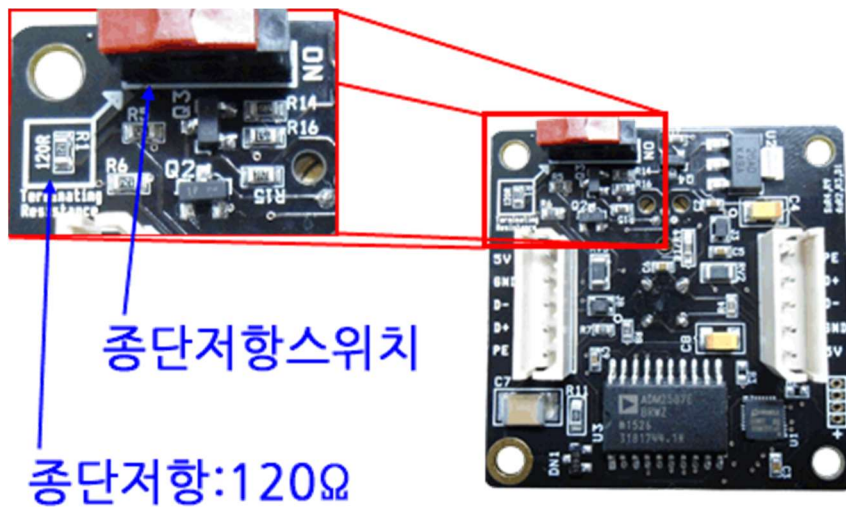
▶ 16 x 4 Array layout



	Col0	Col1	Col2	Col3	Col4	Col5	Col6	Col7	Col8	Col9	Col10	Col11	Col12	Col13	Col14	Col15
Row0	0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60
Row1	1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61
Row2	2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62
Row3	3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63

## ▶ 종단저항

RS485 의 통신에 있어서 지연 등에 의한 왜곡과 감쇠 방지를 목적으로 선로의 임피던스 매칭을 위해 저항을 삽입합니다. 이를 종단저항이라 합니다. DTPA-485-1604N 모듈에는 120옴의 기본 종단저항이 적용돼 있으며 간단히 스위치의 조작을 통해 종단저항 적용을 ON/OFF 시킬 수 있습니다.



< 주 의 : 스위치는 전원 스위치가 아닙니다!! >



< 종단 저항 미적용 >



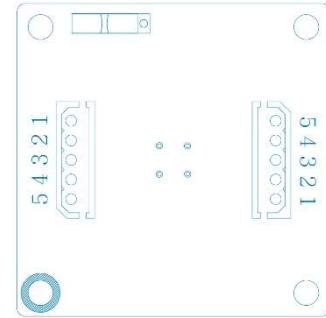
<종단 저항 적용 >

### ※ 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.

## ▶ Pin Assignment

number	Name	Description
1	PE	Protective earth
2	D+	485 D+
3	D-	485 D-
4	GND	Ground
5	5V	Supply Voltage



※ 제품 본체 헤더 정보 : molex 5267-05

※ 결합용 하우징 정보 : molex 5264-05

(주의) 제품 본체만 구매시 케이블은 미포함입니다.

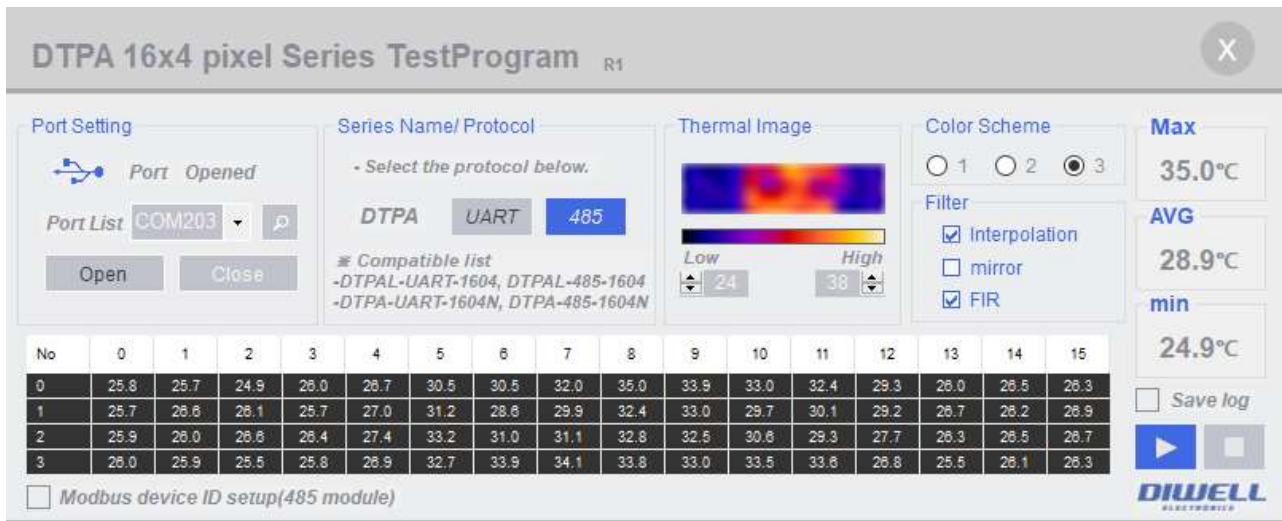
TestKit 구매시에만 케이블이 같이 출고됩니다.

## ▶ 주의 사항

1. 진동이 심한 모터 근처 또는 전원 노이즈가 유발되는 환경은 사용에 주의를 요합니다.
2. 열충격이 발생되지 않는 환경에서 사용하십시오. 급격한 온도 변화는 센서의 정확도에 영향을 줍니다.
3. 태양을 바라보면서 측정하면 안됩니다. 또한 제품이 직사광선에 노출되지 않도록 하세요.
4. 이 제품은 방수가 되지 않습니다.
5. 유증기가 발생하는 곳에서의 사용은 추천하지 않습니다.
5. 독성 화학 물질이 있는 곳에서의 사용을 금합니다. 렌즈 코팅 및 제품 부식 우려가 있습니다.
6. 외력에 의한 제품 파손, 임의 개조 사용, 주의 사항을 지키지 않아 발생한 문제에 대하여 AS는 불가능 하오니 유의 바랍니다.

## ▶ DTPA-485-1604N-TestKit 판매

회로 연결에 어려움이 있으신 분들은 DTPA-485-1604N-TestKit 을 구매하시기 바랍니다. Mini USB B type(별도 구매) cable을 통해 간편하게 PC와 연결이 가능합니다. 그리고 무료 배포되는 윈도우용 프로그램을 실행하면 손쉽게 온도 측정/ 기록이 가능합니다. 별도의 회로 연결이나 전원 구성이 필요 없으며, 응용 분야에 적용이 가능한지 간편하게 확인이 가능합니다.



※ 제공 프로그램은 윈도우 10 전용입니다. 이 외 OS는 지원하지 않습니다.

※ DTPA-485-1604N-TestKit 제품 링크

- [http://www.diwellshop.com/product/detail.html?product\\_no=962](http://www.diwellshop.com/product/detail.html?product_no=962)

## ▶ Ordering Guide

DTPA - △△△ - ◇◇◇◇◇◇

Protocol		Pixel		측정 온도 범위	FOV
485	Modbus 485 RTU	1604N	16 * 4 = 64pixel	-30°C ~ 300°C	가로:53°, 세로:15°
UART	UART 3.3V TTL				

## ▶ Additional Information

- manufacturer : Diwell Electronics Co., Ltd. <(주)디웰전자>
- Homepage : [www.diwell.com](http://www.diwell.com)
- shopping mall : <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)
- 본 문서의 내용은 별도 공지 없이 변경될 수 있습니다.
- 최신 데이터시트는 자사 쇼핑몰을 통해 확인 바랍니다.

## ▶ Revision History

Version	Date	Description
1.0	2020-09-17	First version is released.