



- 비접촉 산업용 온도 측정
- 측정 범위 : -20°C ~ 100°C
- 정확도 : ±0.3°C (20~41°C) , ±1°C (else)
- 응답 시간 : 1초, 방사율 조정 가능
- Digital Interface : Modbus 485 RTU
- KC 인증: R-R-DwL-CTM-7



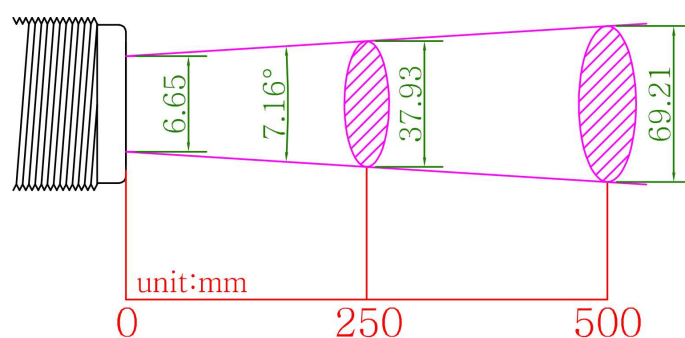
※ 응용 분야 : 식물잎 온도 측정, 배전반/분전반 화재 감시, AI 온도 제어시스템 등
비접촉 온도 측정이 필요한 각종 산업 분야 활용 가능.

▶ General Specifications

- if not otherwise noted, 25°C ambient temperature, 5V supply voltage were applied ($\epsilon = 0.99$)

Parameter	min	Typ	Max	Unit
공급전압	4.75	5	5.5	V
소비 전류(no output load)		15		mA
측정각도(FOV)		7.16		°
온도 측정 범위(Object)	-20	-	100	°C
측정 파장 대역	8	-	14	μm
동작 온도(operating temp.)	-20		70	°C
보관 온도(storage temp.)	-40		85	°C
응답 시간		1		Hz
IR Refresh rate		10		Hz
정확도(Accuracy)	Object 20~41°C 구간		±0.3	°C
	else(0~20, 41~100°C)		±1	°C
표시 단위		0.01		°C
방사율(Emission Coefficient)	0.1	0.99	1.0	ϵ
Standard Start-UP Time		1	2	sec
Stabilization Time	1			min

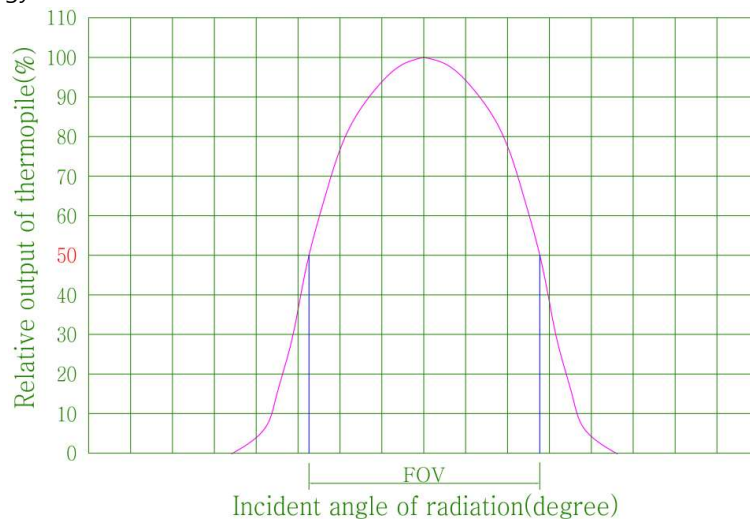
▶ Optical field of view (FOV)

모델명	측정각도	FOV
CT-100G	7.16°	 <p>측정 직경 계산식 : $2 \times \tan(7.16^\circ / 2) \times \text{거리(mm)} + 6.65 \text{ (mm)}$</p>

※ (필독!) 측정 거리 표시는 거리에 따른 측정 면적을 표시하는 것으로 정확도가 보장되는 거리를 의미하는 것이 아닙니다. 거리가 멀어질수록 측정값은 실제 온도와 차이가 발생합니다.

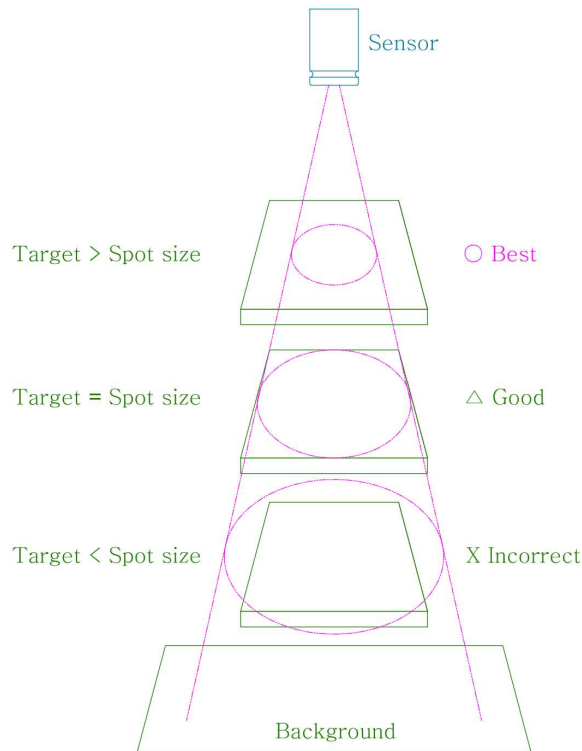
※ 측정하고자 하는 물체의 크기는 위 계산식의 spot size보다 충분히 더 커야 측정이 용이합니다.
다음 페이지 Distance and spot size 그림을 참고하십시오.

The optical chart below indicates the nominal target spot diameter at any given distance from the sensing head and assumes 50% energy.



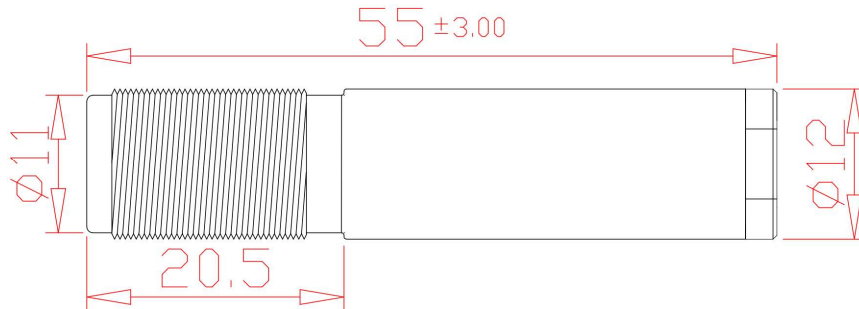
▶ DISTANCE AND SPOT SIZE

Spot Size는 아래 그림에서와 같이 측정하고자 하는 대상의 면적보다 반드시 작아야 합니다.

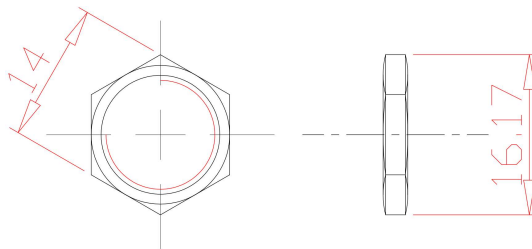


▶ Mechanical Dimensions

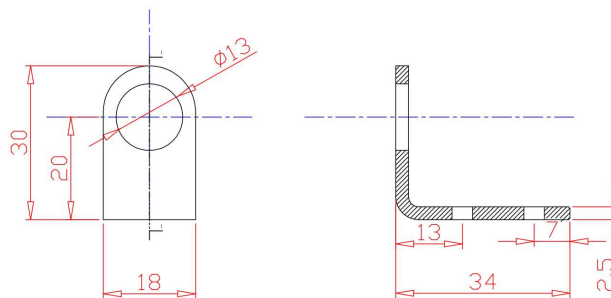
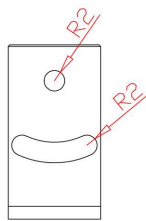
unit : mm



< body case >



< nut >



< bracket - 옵션(별도 구매)>

▶ Pin Assignment(Modbus 485 RTU)

Pin Color	Name	Description	Else
Red	VCC	Supply Voltage (5V)	
Black or Yellow	GND	Ground	
Blue or Green	A	485 D+	
White	B	485 D-	



커넥터 정보 : molex 5264-05

※ 기본 케이블의 길이는 1m 입니다.

※ 케이블 길이 변경 요청시 별도 제작에 따른 납기변경 및 추가비용이 발생 할 수 있습니다.
자세한 사항은 당사로 문의 바랍니다.(070-8235-0820)

▶ 기본 구성품



※ 온도 측정시 제거하세요.

본체 - 1EA

너트 - 2EA

센서 보호 캡 - 1EA

▶ Accessories (옵션 : 별도 구매)



Fixed mounting bracket

▶ 온도 측정 결과표

■ 측정 참고표 (※ 주 의 : 모든 제품이 동일한 오차를 가진 않습니다.)

센서 온도 기준(°C)	1		2		3		4		5	
	측정결과	오차	측정결과	오차	측정결과	오차	측정결과	오차	측정결과	오차
9.875	9.61	-0.265	9.79	-0.085	9.67	-0.205	9.63	-0.245	9.53	-0.345
19.912	19.87	-0.042	20.03	0.118	19.85	-0.062	19.97	0.058	19.83	-0.082
29.97	30.11	0.14	29.91	-0.06	29.97	0	29.97	0	29.99	0.02
40.986	41.03	0.044	41.01	0.024	40.91	-0.076	41.03	0.044	40.95	-0.036
49.926	50.03	0.104	49.95	0.024	50.03	0.104	50.07	0.144	50.19	0.264
59.925	59.99	0.065	59.77	-0.155	59.95	0.025	59.95	0.025	60.13	0.205
69.878	69.93	0.052	69.79	-0.088	69.91	0.032	69.87	-0.008	70.05	0.172
79.87	79.93	0.06	79.79	-0.08	79.91	0.04	79.99	0.12	80.19	0.32
89.88	89.95	0.07	89.59	-0.29	89.77	-0.11	89.71	-0.17	90.21	0.33
99.87	99.93	0.06	99.57	-0.3	99.79	-0.08	99.85	-0.02	100.23	0.36

실험 조건 [환경 온도 : 25°C, 측정 거리 : 6cm, 습도 : <30%, 흑체 방사율 : 0.99]

센서 흑체 접근 후 5초후에 측정 1회, 측정할 경우를 제외 센서와 흑체는 이격

각 온도 구간 안정화 시간 30분 후 측정

※ 온도 정확도는 아래의 조건 등에 영향을 받아 변화될 수 있습니다.

- ▷ 센서의 열평형이 이루어져 있을 것.
- ▷ 센서 패키지에 열이 가해지거나, 온도 차이가 없을 것.
- ▷ 측정 물체가 센서의 측정 영역에 꼭 채워질 것.
- ▷ 측정 영역에서 대상 표면의 온도가 균일하게 분포할 것.
- ▷ 측정 영역에서 대상 표면 재질의 방사율이 일정할 것.
- ▷ 측정할 때마다 거리가 변화되지 않을 것.

※ 정확도에 관한 이해(필독)

온도 정확도는 모든 환경에서 항상 만족하는 것은 아닙니다. 센서의 열평형 상태가 유지되지 않는 환경에서는 정확도 유지가 어렵다는 것을 가장 중요하게 이해하고 있어야 합니다. 예를 들어, 센서 패키지에 순간적으로 온도 변화를 일으킬 수 있는 외부 요소(히터, 에어컨, 센서 근처의 발열 IC, 손으로 잡는 행위 등)가 미친다면, 센서는 열충격을 받기 때문에 측정 오차가 발생할 수 있습니다.

이러한 온도 차이를 temperature gradient 라 부르며, 본 제품은 내부적으로 열충격을 active하게 보정하는 기능이 추가되었습니다. 따라서 이러한 보정 기능이 없는 센서들과 비교해 좀 더 안정적인 측정이 가능합니다. 하지만 완벽하게 제거되는 것은 아니기 때문에, 애초에 이런 온도차이가 발생되지 않도록 외부 요소를 피하거나 센서를 보호해야 합니다.

▶ MODBUS OVER SERIAL LINE(RS485)

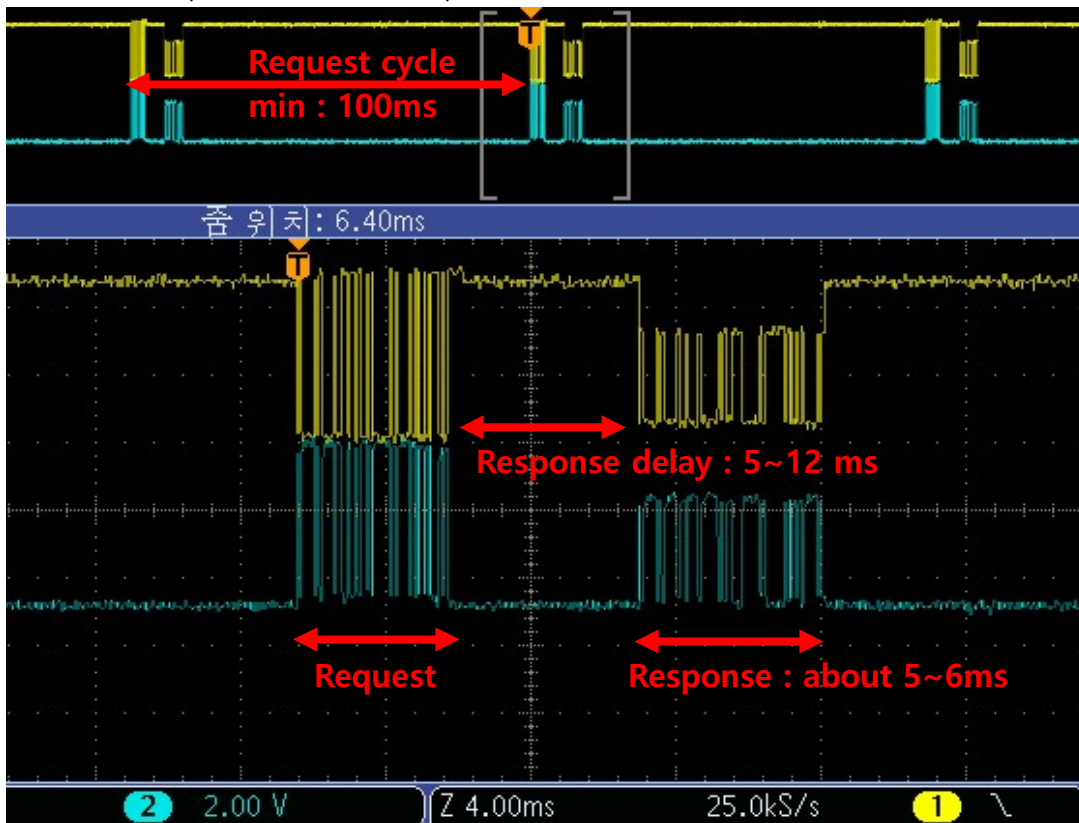
1. 통신 규격

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

통신속도(bps)	DATA bit	Stop bit
19,200	8	1

2. 송수신 시퀀스

- ※ Request 주기 : 반드시 최소 100ms 이상
- ※ 전원 공급 후 첫 Request Timing : 최소 1초 이후
(센서의 Start-up 시간보다 빠른 Request 는 응답이 없거나 데이터 수치 0 이 전송됩니다.)



※ 위 파형은 Object, PCB 온도 데이터를 읽어오는 파형입니다.

3. Supported Functions

Read register	0x03
Write single register	0x06

4. Address Table

- R = Read - W = Write (single write)

Address		Length (short)	Description	R/W
Dec	Hex			
40,000	0x9C40	1	Device ID (1 ~ 200)	R/W
40,001	0x9C41	1	Emissivity (방사율: 10~100. default : 99)	R/W
40,002	0x9C42	1	Object temperature (측정 물체의 온도)	R
40,003	0x9C43	1	PCB temperature (제품 내부 pcb 의 온도, 동작온도)	R

- ※ 위 Address Table 의 Address 에만 접근/ 수정이 가능합니다.
- ※ 잘못된 Request Command 에는 응답하지 않습니다.
- ※ Read Only 에 해당하는 Address 에는 Write 할 수 없습니다.
- ※ Modbus Broadcast 는 지원하지 않습니다.

5. Communication Protocol Table.

5-1. Device ID 변경 하기 (FUNCTION 06)

※ ID 1 인 제품을 원하는 ID 로 변경하기 예제.

Request (Main → Sensor)			Response (Sensor → Main)		
Field Name	Hex	DEC	Field Name	Hex	DEC
Device ID	0x01(현재ID)	1	Device ID	0x01	1
Function	0x06	6	Function	0x06	6
Register Address Hi	0x9C	40,000	Register Address Hi	0x9C	40,000
Register Address Lo	0x40		Register Address Lo	0x40	
Register Value Hi	0x00	1~200	Register Value Hi	0x00	1~200
Register Value Lo	변경할 ID		Register Value Lo	ID	
CRC	6.CRC란? 참고	※ Note 1	CRC		※ Note 2
CRC			CRC		
8 Byte Request			8 Byte Response		

- ※ default ID 는 1 번입니다. 1~200 번까지 변경이 가능합니다.
- ※ 동일한 Device ID 의 제품 여러 개를 동시에 연결하면 절대 안됩니다.
- ※ Device ID 변경 시 관리에 주의를 요합니다. 변경한 ID 를 잊었을 경우 ID 1~200 까지 데이터 응답이 있는지 Scan 하여 찾아야 합니다. TestKit 에 ID scan 기능이 구현되어 있습니다.
- ※ 규칙에 맞지 않는 데이터 요청시 센서의 응답은 없습니다.
- ※ 전원을 reset 해도 값이 초기화 되지 않습니다.
- ※ Note 1, 2 : CRC 는 통신과정에서 오류가 없는지 검증하는 데이터 입니다. 임의 값 부여(X).

5-2. 온도 데이터 읽기 (FUNCTION 03)

Request (Main → Sensor)			Response (Sensor → Main)		
Field Name	Hex	DEC	Field Name	Hex	DEC
Device ID	0x01~0xc8	1~200	Device ID	0x01~0xc8	1~200
Function	0x03	3	Function	0x03	3
Starting Address Hi	0x9C	40,002	Byte Count	0x04	4
Starting Address Lo	0x42		Data Value 1 Hi	Object temp. (측정 물체 온도)	
No. of Data Hi	0x00	2	Data Value 1 Lo		
No. of Data Lo	0x02		Data Value 2 Hi	PCB temp (동작 온도)	
CRC			Data Value 2 Lo		
CRC			CRC		
			CRC		
8 Byte Request			9 Byte Response		

※ Address Table을 참고하여, Starting Address 및 No. Of Data 를 변경하면 원하는 데이터만 수신할 수 있습니다.

5-2-1. 온도 계산 법

uTemp : unsigned 2 바이트 크기 변수 선언

fTemp : float 또는 double 형 선언

uTemp = (Data Hi byte) * 256 + (Data Lo byte)

fTemp = (double)uTemp * 0.02 - 273.15

※ 계산 예제 1

Data Value Hi = 0x3B,

Data Value Lo = 0xFB

$0x3B * 256 + 0xFB = 15,355$

$(15,355 * 0.02) - 273.15 = 33.95$

즉, 33.95°C 가 됩니다.

※ 계산 예제 2

Data Value Hi = 0x33

Data Value Lo = 0x65

$0x33 * 256 + 0x65 = 13,157$

$(13,157 * 0.02) - 273.15 = -10.01$

즉, -10.01°C 가 됩니다.

5-2-2. Output Data Limit(Object temp)

데이터의 출력 범위는 **-19.99°C ~ 105.01°C** 입니다. 온도 범위를 초과할 경우 제한이 걸립니다.

→ 센서에서 출력하는 최소 수치(min) : **12,658(dec) = -19.99°C**

→ 센서에서 출력하는 최대 수치(Max) : **18,908(dec) = 105.01°C**

예) 120°C 를 측정하여도 105.01°C 로 출력합니다.

PCB temp는 별도 출력 수치에 제한이 없습니다.

5-3. 방사율(Emissivity) 변경 하기 (FUNCTION 06)

방사율이란 물체가 외부 적외선 에너지를 흡수, 투과 및 반사하는 비율을 말하는데, 이론적으로 외부에 에너지를 흡수만 하고 반사하지 않는 물체를 흑체라 하여 이때의 방사율은 "1"입니다. 하지만, 일반적으로 물체의 표면상태(광택, 거칠, 산화여부 등)에 따라 흡수, 반사하는 에너지량이 변합니다. 재질에 따른 방사율 변경은 본문 "5-5. 방사율표" 및 변경 프로토콜을 참고하십시오. 단, "방사율표"의 값은 절대적인 값이 아니며 표면 상태와 그 외 복합적인 환경 요인에 따라 변할 수 있으니, 이점 참고하십시오. 특정 물체만이 아닌 두루 대상물체를 측정하고자 하는 경우에는 변경에 신중을 기해야 합니다.

※ 측정하고자 하는 대상의 재질이 코팅이 돼 있거나 반짝이는 재질(동판, 알루미늄, 스테인리스 등)이라면 방사율을 수정 적용해도 온도 변화가 적을 수 있습니다. 이 때에는 방사율 조절을 하지 마시고, 측정하고자 하는 측정 부위에 "흑체 테이프" 또는 "흑색 무광 스프레이"를 이용하여 반짝임을 없애면, 보다 정확한 온도를 측정할 수 있습니다. 단, 테이프나 스프레이가 적합한 내열성을 가지는지 확인하십시오.

Request (Main → Sensor)			Response (Sensor → Main)		
Field Name	Hex	DEC	Field Name	Hex	DEC
Device ID	0x01~0xc8	1~200	Device ID	0x01~0xc8	1~200
Function	0x06	6	Function	0x06	6
Register Address Hi	0x9C	40,001	Register Address Hi	0x9C	40,001
Register Address Lo	0x41		Register Address Lo	0x41	
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. 아래 표를 참고하세요.

Emissivity	전송할 데이터	
	HEX	DEC (방사율*100)
0.1	0x0A	0.1*100 = 10
0.5	0x32	0.5*100 = 50
0.97	0x61	0.97*100 = 97
0.99	0x63	0.99*100 = 99

만약 방사율을 0.99로 변경할 경우 "방사율"에 정수값 99를 입력하면 됩니다.

※ 변경할 제품의 Device ID를 반드시 확인 후 진행해야 합니다.

※ 전원을 reset 해도 값이 초기화 되지 않습니다.

5-4. 방사율(Emissivity) 읽기 (FUNCTION 03)

Request (Main → Sensor)			Response (Sensor → Main)		
Field Name	Hex	DEC	Field Name	Hex	DEC
Device ID	0x01~0xc8	1~200	Device ID	0x01~0xc8	1~200
Function	0x03	3	Function	0x03	3
Starting Address Hi	0x9C	40,001	Byte Count	0x02	2
Starting Address Lo	0x41		방사율 Hi (항상 0)	0x00	0
No. of Data Hi	0x00	1	방사율 Lo	0x0A~0x64	10~100
No. of Data Lo	0x01		CRC		
CRC			CRC		
CRC					
8 Byte Request			7 Byte Response		

5-5. 방사율표

대상	방사율	대상	방사율	대상	방사율
산화아연	0.1	에나멜	0.9	구리(연마된)	0.5
아연도금철	0.3	페인트	0.95	구리(산화된)	0.8
주석도금철	0.1	라 카	0.9	니켈(순수)	0.1
금(연마된)	0.1	고무(smooth)	0.9	니켈(산화된)	0.4~0.5
은(연마된)	0.1	고무(Rough)	0.98	니켈크롬	0.7
크롬(연마된)	0.1	플라스틱	0.8~0.95	니켈크롬(산화된)	0.95
붉은 벽돌	0.75~0.9	플라스틱필름	0.5~0.95	직물	0.9
흙	0.92~0.96	주철(연마)	0.2	피부	0.98
석면	0.95	Steel	0.6	가죽	0.75~0.8
콘크리트	0.7	산화 Steel	0.9	얼음	0.96~0.98
대리석	0.9	목재	0.8~0.9	모래	0.9
모르타르	0.89~0.91	스테인레스(연마된)	0.1	아스팔트	0.9~0.98
석고	0.85	스테인레스(기타)	0.2~0.6	유리	0.8~0.9
시멘트	0.96	알루미늄(연마된)	0.1	물	0.8~0.9
규토(정제된)	0.4	알루미늄(합금)	0.1~0.25	종이	0.9
세라믹	0.90~0.94	황동(연마된)	0.1	실리콘	0.7
석 영	0.9	황동(거친)	0.2	주철(부식된)	0.95
석 탄	0.75	황동(산화된)	0.6	Mild Steel	0.3~0.5
Fe(부식된)	0.7~0.85				

6. CRC 란?

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, 0xCF41, 0xCE81, 0x0E40, 0x0A00, 0xCAC1, 0xCB81, 0x0B40, 0xC901, 0x09C0,
    0x0880, 0xC841, 0xD801, 0x18C0, 0x1980, 0xD941, 0x1B00, 0xDB41, 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 명령(FUNCTION3)의 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
```

본문 (5-2. 온도 데이터 읽기) 를 참고하여 메인 함수에서 배열 요소에 값을 할당합니다.

```
RequestData[0] = 0x01;
RequestData[1] = 0x03;
RequestData[2] = 0x9C;
RequestData[3] = 0x42;
RequestData[4] = 0x00;
RequestData[5] = 0x02;
```

여기까지가 보낼 데이터이며, RequestData[6], RequestData[7]에 CRC 연산 후 값을 넣어야 합니다.
이전 페이지의 CalcCRC16 함수를 사용합니다.

```
crc = CalcCRC16(RequestData, 6); // RequestData 배열 0~5 까지, 총 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 = 0x4F4A (dec : 20,298)
RequestData[6] 변수에는 0x4A 값을 저장합니다. (CRC 값의 하위 바이트)
RequestData[7] 변수에는 0x4F 값을 저장합니다. (CRC 값의 상위 바이트)
```

CRC를 포함한 전송할 데이터 준비는 끝났으며, RequestData 배열 0~7 의 값을 전송하면 됩니다.
수신 측(센서)에서는 (RequestData[6], RequestData[7] 의 데이터) 와 (수신 데이터의 crc 연산값)을 비교한 후, 오류가 없는 요청에 한하여 응답을 합니다.

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

※ 온도 요청 데이터 예제(CRC 포함)

ID1번 : 0x01, 0x03, 0x9C, 0x42, 0x00, 0x02, (0x4A), (0x4F)
ID2번 : 0x02, 0x03, 0x9C, 0x42, 0x00, 0x02, (0x4A), (0x7C)

▶ 청소 및 관리

※ 제품의 렌즈가 오염되면 오염원에 따라 측정값이 부정확 할 수 있습니다.

※ 어떠한 경우에도 렌즈에 압력이 가해져서는 절대 안됩니다. 반드시 지켜주세요.

※ 렌즈 오염시 청소(일반 먼지)

- 부드러운 천(안경 천)으로 파손되지 않게 주의하여 이물질을 제거하세요.
- 청소 후에 바로 온도 측정을 하면 온도가 부정확할 수 있습니다.
- 청소 후에는 상온에서 최소 30분 정도 보관 후 측정을 시작하세요.
- 가벼운 생활 먼지 정도는 카메라 먼지 제거용 에어브러시 등을 활용하세요.
- 고압의 에어브러시는 사용을 금지합니다. 렌즈 파손의 우려가 있습니다.

※ 렌즈 오염시 청소(기름때)

- 유증기가 발생하는 곳에서의 사용은 추천하지 않습니다.
- 기름때의 정도에 따라 온도가 부정확해 질 수 있습니다.
- 오염이 심하다면 광학 렌즈 전용 세척액을 사용하는 것을 추천합니다만, 힘주어서 닦아 내면 안됩니다. 렌즈 코팅이 손상될 수 있습니다.
- 이소프로필 알코올(IPA) 이 광학 렌즈의 세정에 많이 쓰이니 참고하십시오.
- 에탄올, 메탄올, 아세톤, 그 외 렌즈 코팅을 손상시킬 수 있는 공업용 물질은 사용을 금합니다.
- 전용 세척액을 사용하더라도 오랜 기간 동안 반복 세척시 코팅이 벗겨질 수 있습니다. 렌즈 코팅이 손상되어 온도 차이가 심해지면, 신규 구매를 하셔야 합니다.

※ 특정 조건(거리에 따른 온도 보상)으로 설정하여 사용할 경우, 항상 같은 조건이 유지 돼야 합니다.

※ 독성 화학 물질이 있는 곳에서의 사용을 금합니다. 렌즈 코팅 및 제품이 부식될 수 있습니다.

※ 태양을 바라보면서 측정하면 안됩니다. 직사광선에 제품이 노출되지 않도록 하세요.

※ 방수가 되지 않으니 물이 튀는 환경 또는 비를 맞는 곳에서의 사용을 금합니다.

※ 전자파 간섭을 최소화 하기 위해서 가급적 모터나 generator 와 가까운 곳에 설치하지 마십시오.

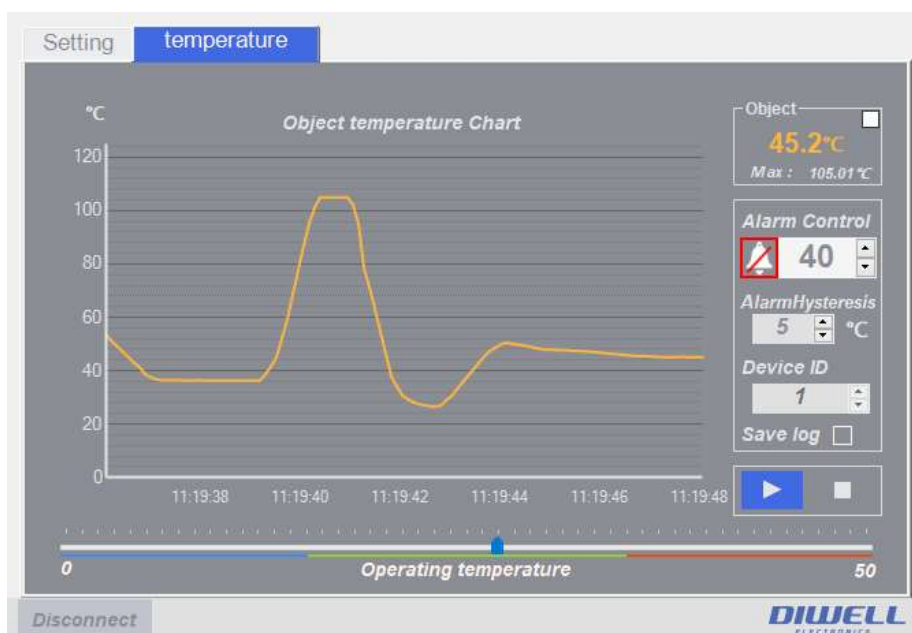
※ 외력에 의한 제품 파손, 임의 개조 사용, 잘못된 회로 연결로 인한 내부 보드 파손 등 주의사항을 지키지 않아 발생한 문제에 대하여 AS 는 불가능 하오니 유의 바랍니다.

▶ CT-100G TestKit

본 제품이 응용분야에 적합한지 측정해 보기 위해 회로나 프로토콜 분석을 해야 한다면, 구현에 따른 시간/ 비용이 소요 됩니다. CT-100G TestKit 은 USB 연결 만으로 PC에서 응용 분야에 적용이 가능 할지를 빠르게 판단할 수 있도록 도와줍니다. 자세한 사항은 TestKit 매뉴얼을 참고하십시오.

TestKit 은 Window 10 환경에서만 지원합니다.

http://www.diwellshop.com/product/detail.html?product_no=972



< Window PC 측정 화면 >

▶ Additional Information

- manufacturer : Diwell Electronics Co., Ltd. <(주)디웰전자>
- Homepage: <http://www.diwell.com>
- Shopping mall : <http://www.diwellshop.com>
- Phone : 070-8235-0820
- Fax : 031-429-0821
- Technical support : expoeb2@diwell.com, dsjeong@diwell.com
- 본 데이터시트 내용은 별도 공지 없이 변경될 수 있습니다. 최신 버전의 데이터 시트는 디웰전자 쇼핑몰 상세 페이지에서 확인 바랍니다.

▶ Revision History

Version	Date	Description
1.0	2021-3-2	First version is released.