



특장점

- 온도, 습도센서를 하나로 구성
- Photo-Acoustic방식의 소형 CO2센서,
- ±0.09°C의 정확한 온도센서 탑재,
- ±2%의 정확한 습도센서 탑재
- 데이터 로깅 프로그램(SR-100) 연동 가능(무료 배포)

빌딩공조, 실내 공기질관리, 농업현장의 환경감시 및 제어 시스템에 최적화 된 소형 복합센서.

습도센서는 일반적인 화학물질에 내성이 강하고, 안정성과 내구성이 우수한 정전 용량방식을 채택하였고, 온도센서는 정확한 반도체방식의 센서를 채택.

센서는 무료로 배포하는 데이터 로깅 프로그램(SR-100)과 연동하여 이용하실 수 있습니다.

프로그램(SR100) 설치환경은 Microsoft WINDOWS 7 / 8 / 10 입니다.
(자세한 사항은 홈페이지 참조)



Technical data

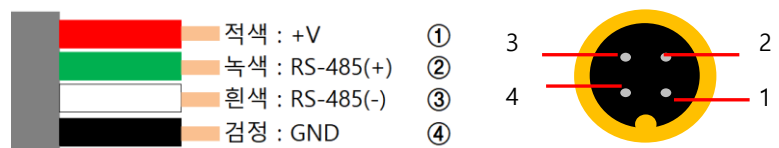
Measurement	검출방식	습도 : 정전용량 방식 온도 : 반도체 방식
	검출범위	습도 : ~ 95%RH 온도 : -40 ~ 60°C
	불확도 (@25°C)	습도 : ±2% (10~70%RH) 온도 : ±0.09°C (20~40°C)
Electrical	사용전원	5~24VDC
	소모전력	< 0.2W
	소모전류 (@ DC12V)	평균 : 약 10mA 최대 : 약40mA
	통신출력	RS-485 (Modbus RTU 2선식) Baudrate 115,200s
Dimensions	외형	Φ27.6mm x 106.1mm (케이블 제외)
	케이블	M12-4P, 3m
	무게	센서 30g, 케이블 85g,
	커넥터	M12-4p (센서: SCN-12-04A, 케이블: SC-12-04A)

- 본 자료 내용은 품질 향상을 위해 예고 없이 변경 될 수 있습니다.

외형크기



케이블 커넥터 (SC-12-04A)



■ 통신 Default COM Setting (RS485 MODBUS RTU / 2선식)

내용	Default	비 고
ADDRESS (ID,국번)	49(0x31)	
통신속도	115200bps	
Data bit	8	
Parity	None	
Stop bit	1	

■ 모드버스 통신

	Register Address (Hex)	Register Type	Description	Range	비 고
제품 정보	0x10	Input(0x04) / r	고유 식별 코드 (0x400A)		
	0x11	Input(0x04) / r	제품 구분 (0x02)		0x01 : KSH-7330 0x02 : KSH-7360
	0x12	Input(0x04) / r	F/W Version (0x01)		
측정	0x40	Input(0x04) / r	Humidity Value * 10	[0~1000] 0~100.0%RH	2byte signed형[short]
	0x41	Input(0x04) / r	Temperature Value * 10	[-400~1200] -40.0~120.0°C	2byte signed형[short]
	0x42	Input(0x04) / r	CO2 Value	[0~4000] 0~4000ppm	2byte signed형[short]
설정	0xFF81	Holding(0x03) / rw	Modbus ID(0x31)	1~128	
	0xFF82	Holding(0x03) / rw	Baudrate	0~6	0 : 2400 bps 1 : 4800 bps 2 : 9600 bps 3 : 19200 bps 4 : 38400 bps 5 : 57600 bps 6 : 115200 bps

■ Connection Information

Register Access

Function Code (Decimal)	Function Code (Hexadecimal)	Register type	Read / Write
03	03 hex	Read Holding Register	R / W
04	04 hex	Read Input Register	Only Read
06	06 hex	Single Write Coil Register	-
16	10 hex	Write Holding Register	-

Register Table

Modbus Measurement Data Registers (Read-Only)

Address (Decimal)	Address (Hexadecimal)	Register Description	Data Format	Unit
16	0010 hex	Application Type	16-bit unsigned integer	400A hex
17	0011 hex	Hardware Version	16-bit unsigned integer	KSH7310 (0001 hex) KSH7360 (0002 hex)
18	0012 hex	Firmware Version	16-bit unsigned integer	0001 hex

Modbus Measurement Data Registers (Read-Only)

Address (Decimal)	Address (Hexadecimal)	Register Description	Data Format	Unit
64	0040 hex	Measured Temperature*10	16-bit signed integer	°C
65	0041 hex	Measured Humidity*10	16-bit signed integer	%RH
66	0042 hex	Measured CO2 value	16-bit signed integer	ppm

[§] The measured temperature and humidity value register (address 64, 65) is scaled and multiplied by 10.

Modbus Configuration Data Registers (Writable / Available, but not functional.)

Address (Decimal)	Address (Hexadecimal)	Register Description	Data Format	Unit
65409	FF81 hex	Modbus address (Available)	16-bit integer	Valid range 1...128 (default:49 (31hex))
65410	FF82 hex	Serial speed (Available)	enum 16-bit integer	Valid range 2400...115200 0 = 2400 1 = 4800 2 = 9600 3 = 19200 4 = 38400 5 = 57600 6 = 115200 (default)

Calibration Protocol

	전송 패킷	응답 패킷
센서 측정값 읽기	{CMD=20,CHS=494}	
센서 보정	{CMD=3000,INT=1000,CHS=1124} (측정센서를 1000ppm으로 보정 예)	RS485통신으로 다중 센서(1:N) 보정을 위한 것임. 보정은 응답 없이 센서내부 연산 후 재 부팅.

Checksum 계산

전송 패킷과 응답 패킷을 포함하는 모든 패킷은 Checksum 정보를 추가하여 통신 오류에 대한 처리를 수행할 수 있습니다. Checksum의 값은 패킷 구성에서 CHS 접두사로 시작됩니다.

Checksum 값을 구하는 방법은 패킷 시작 기호 ({)에서CHK 접두사 앞의 속성 구분자 (,) 이전까지, 모든 문자를 표현하는 Hex값을 더하여 구한 값을 십진수 ASCII 값으로 구합니다.

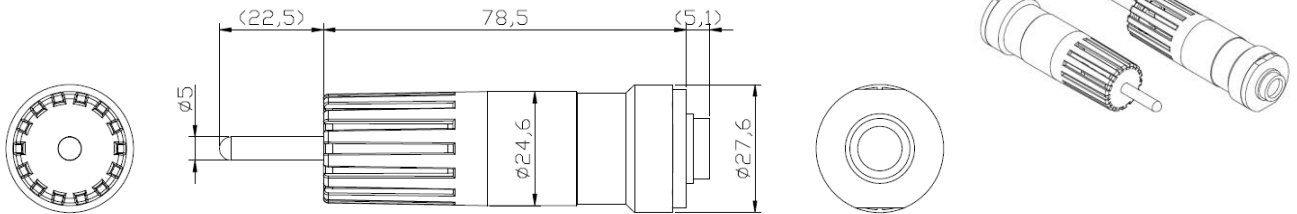
Checksum 값을 구하는 방법에 대한 C언어 예제는 다음 내용을 참조하여 주십시오.

[Checksum 연산 예제]

```
//간단한 패킷 만들기( {CMD=xxx,CHK=xxx})
static char MsgBuffer[COMM_BUFFERSIZE];
char *CreateMessage(MSG_COMMAND command)
{
    char tempstrbuf[0x20];
    sprintf((char*) MsgBuffer, "%s=%d", "CMD", command);
    {
        int chksum = compute_checksum(MsgBuffer, strlen(MsgBuffer));
        sprintf((char*) tempstrbuf, "%s=%d\r\n", "CHS", chksum);
        strcat(MsgBuffer, tempstrbuf);
    }
    return MsgBuffer;
}
```

```
//Checksum 구하기
int compute_checksum(char *string, int size)
{
    uint8_t ch;
    int chksum = 0;
    for (int i = 0; i < size; i++)
    {
        ch = (uint8_t) string[i];
        chksum += ch;
    }
    return chksum;
}
```

■ 외형도



■ 기타

함께 사용 가능한 간이 백엽상, 센서 부착용 브리켓트 등, 부속자재가 있습니다. 검토 중 필요하신 사항 문의하여 주십시오.

