

# USER'S GUIDE

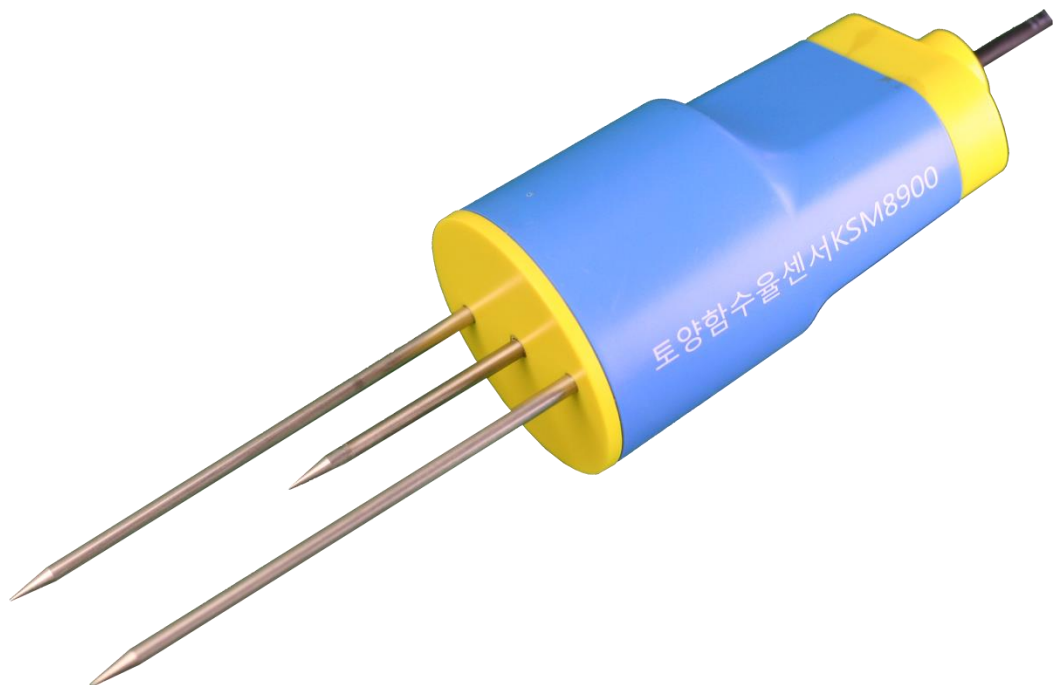
101101001101010100  
0100100101011010101  
01101010101010110  
0101010101010110110



## Soil Moisture / EC / Temperature

### Sensor

### KSM8900



## 공급자

코리아디지털(주)

서울시 구로구 디지털로 273, 에이스트원타워 2차 804호

Tel. 02)2109-8877 Fax. 02)2109-8878

[www.koreadigital.com](http://www.koreadigital.com)

이 설명서의 어떠한 부분도 사전 서면 허가 없이 제 3자에게 수정, 번역, 개조, 판매 또는 공개 할 수 없으며, 저작권자의 허가 없이 어떠한 형식이나 수단으로도 복제하거나 공개적으로 사용할 수 없습니다.

이 설명서의 내용은 사전 통지 없이 변경 될 수 있습니다.

# Table of Contents

## 제 1 장

<b>일반 정보</b> .....	<b>3</b>
<b>매뉴얼에 관하여</b> .....	<b>3</b>
매뉴얼의 내용.....	3
버전 정보.....	3
관련 매뉴얼.....	3
문서 규칙.....	4
<b>안전</b> .....	<b>5</b>
<b>보증</b> .....	<b>5</b>

## 제 2 장

<b>제품 개요</b> .....	<b>6</b>
<b>KSM8900 토양 수분 센서 소개</b> .....	<b>6</b>
기본 기능 및 옵션.....	7
KSM8900의 구조.....	8

## CHAPTER 3

<b>설치</b> .....	<b>9</b>
<b>단독 설치</b> .....	<b>9</b>
<b>다중 설치</b> .....	<b>11</b>
<b>연결</b> .....	<b>13</b>
M12 Connector.....	13
Strip Wire.....	13

## CHAPTER 4

<b>운용</b> .....	<b>14</b>
<b>시작하기</b> .....	<b>14</b>
<b>RS-485 통신 연결</b> .....	<b>14</b>
<b>RS-485 통신 설정</b> .....	<b>16</b>
<b>통신 프로토콜</b> .....	<b>17</b>

패킷 구성.....	17
전송 명령.....	18
응답 명령.....	19
전송 명령 속성.....	20
응답 명령 속성.....	22
다중 센서 사용시 센서 선택.....	23
Checksum 계산.....	24
RS-485 통신 송수신 패킷 예제.....	25
RS-485 통신 송신 예제 코드.....	27
RS-485 통신 수신 예제 코드.....	28
<b>MODBUS RTU 통신 프로토콜.....</b>	<b>31</b>
마스터 전송 패킷 구성.....	31
마스터 수신 패킷 구성 - 기능코드 04(Input Resister Read).....	33
마스터 수신 패킷 구성 - 기능코드 06(Preset Single Resister).....	34
예외 패킷 구성.....	34
MODBUS RTU 통신 송수신 패킷 예제.....	35
 CHAPTER 5	
<b>유지 보수.....</b>	<b>37</b>
청소.....	37
보관.....	37
기술지원.....	37
 CHAPTER 6	
<b>기술 데이터.....</b>	<b>38</b>

## CHAPTER 1

### 일반 정보

이 장에서는 매뉴얼 및 제품에 대한 일반적인 참고 사항을 제공합니다.

#### 매뉴얼에 관하여

이 매뉴얼은 KOREADIGITAL(주) KSM8900 토양 수분 센서의 설치, 작동 및 유지 보수에 대한 정보를 제공합니다.

#### 매뉴얼의 내용

이 매뉴얼은 다음과 같이 구성되어 있습니다.

- 제 1 장 : 일반 정보, 매뉴얼 및 KOREADIGITAL(주) KSM8900 토양 수분 센서에 대한 일반적인 정보
- 제 2 장 : 제품 개요, KSM8900 토양 수분 센서의 기능, 장점 및 제품 명칭
- 제 3 장 : 설치, KSM8900 토양 수분 센서를 설치를 돕기 위한 정보
- 제 4 장 : 운영, KSM8900 토양 수분 센서의 운용에 필요한 정보
- 제 5 장 : 유지 보수, KSM8900 토양 수분 센서의 기본적인 유지 보수 정보
- 제 6 장 : 기술 데이터, KSM8900 토양 수분 센서의 기술 데이터

#### 버전 정보

버전	설명
Rev 1.0	초기 배포용 매뉴얼
Rev 1.1	MODBUS RTU Protocol 추가
Rev 1.2	MODBUS RTU Unsigned Int형 센서값 읽기 지원 기능 추가

#### 관련 매뉴얼

버전	설명
Rev 1.0	KOREADIGITAL(주) KSM8900 토양 수분 센서 Quick Reference Guide

## 문서 규칙

이 매뉴얼 전반에 걸쳐 중요한 안전 관련 정보는 다음과 같이 강조 표시됩니다.

### **WARNING**

심각한 위험을 경고합니다.  
지시 사항을 숙지하고 주의 깊게 따르지 않으면 상해  
나 사망의 위험이 있습니다.

### **CAUTION**

잠재적 위험을 경고합니다.  
지시 사항을 숙지하고 주의 깊게 따르지 않으면 제품  
이 손상되거나 중요한 데이터가  
손실 될 수 있습니다.

### **NOTE**

제품 사용에 대한 중요한 정보를 표시합니다.

## 안전

### **WARNING**

제품의 전극은 매우 날카로우므로, 취급에 주의해 주시고, 센서를 들고 절대 장난을 치거나 다른 행동을 하지 마십시오.

### **NOTE**

장치를 개조하지 마십시오. 부적절한 개조는 제품을 손상 시키거나 오작동을 일으킬 수 있습니다.

## 보증

본 제품은 정상적인 설치 및 사용시 2년의 보증 기간을 가집니다.

단, 정상적인 마모로 인한 손상, 예외적인 작동 조건, 부주의한 취급 또는 설치, 무단 개조로 인한 손상이나 고장은 보증이 유효하지 않을 수 있습니다.

자세한 내용은 해당 공급에 대한 계약이나 판매 조건을 참조 하십시오.

## CHAPTER 2

**제품 개요**

이 장에서는 KOREADIGITAL(주) KSM8900 토양수분센서의 기능, 장점 및 제품의 명칭을 설명합니다.

**KSM8900 토양 수분 센서 소개**

KSM8900 토양 수분 센서는 토양의 함수율과 전기 전도도, 지중 온도 측정을 모두 제공합니다.

토양 함수율은 FDR (Frequency Domain Reflectometry) 방식을 응용하여 용적 수분 함량 법으로 측정하고, 전기 전도도(EC) 는 임피던스 측정 기술을 응용하여 토양이 함유하고 있는 수분의 전기 전도도를 측정하며, 지중 온도는 MEMS 기술을 이용하여 측정합니다.

센서의 끝부분에 해당하는 3개의 스테인레스 금속 전극으로 토양의 함수율과 전기 전도도를 측정하고, 전극 내부에 MEMS 온도 센서를 내장하여 온도를 측정합니다.

KSM8900 토양 수분 센서는 RS-485 통신 방식으로 토양의 함수율과 전기 전도도, 지중 온도 측정값을 출력합니다.

KSM8900 토양 수분 센서에서 제공하는 토양 함수율, 전기 전도도, 지중 온도 측정값은 아래의 측정 단위로 출력합니다.

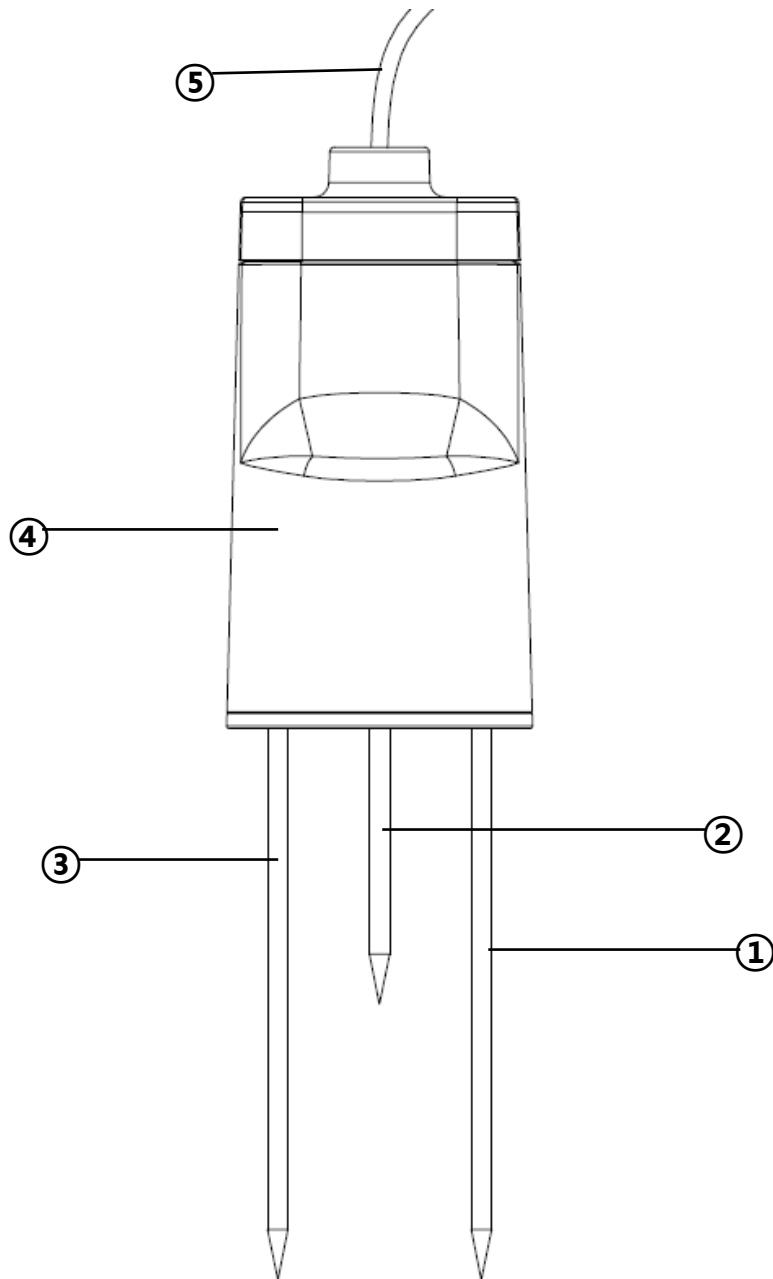
측정 항목	단위
토양 함수율	%VWC
전기 전도도	dS/m
지중 온도	°C



## 기본 기능 및 옵션

- 전통적인 토양 함수율 측정 방식처럼 토양을 직접 채취하고 복잡한 절차를 거치지 않아도 현장에서 간단히 토양의 수분을 측정하고 모니터링 할 수 있습니다.
- 토양의 함수율을 측정함으로써 작물의 수분 부족이나 수분 과다에 따른 피해를 미리 예방하고 관리할 수 있습니다.
- 토양의 전기 전도도를 측정함으로써 적정 토양 염분을 유지하도록 관리할 수 있습니다.
- 토양의 온도를 측정함으로써 작물이 적합한 온도에서 자랄 수 있도록 관리할 수 있습니다.
- KOREADIGITAL(주) 스마트팜 컨트롤러에 연결시 원격지에서 스마트폰을 활용하여 토양의 함수율, 전기 전도도 및 지중 온도를 확인할 수 있으며, 간단한사용자 설정으로 일정한 토양 함수율을 유지하도록 관수 장치와 연동할 수 있습니다.

## KSM8900의 구조



- ①, ②, ③ = 토양 함수율, 전기 전도도 측정 전극 (Stainless Steel)
- ② (내부) = 지중 온도 측정 소자 (MEMS)
- ④ = 몸체 (PC & ABS)
- ⑤ = 케이블 (AWG26 \* 4Core Shield Cable)

## CHAPTER 3

### 설치

이 장에서는 다양한 상황에서 KSM8900 토양 수분 센서를 설치하는 방법에 대한 정보를 제공합니다.

#### 단독 설치

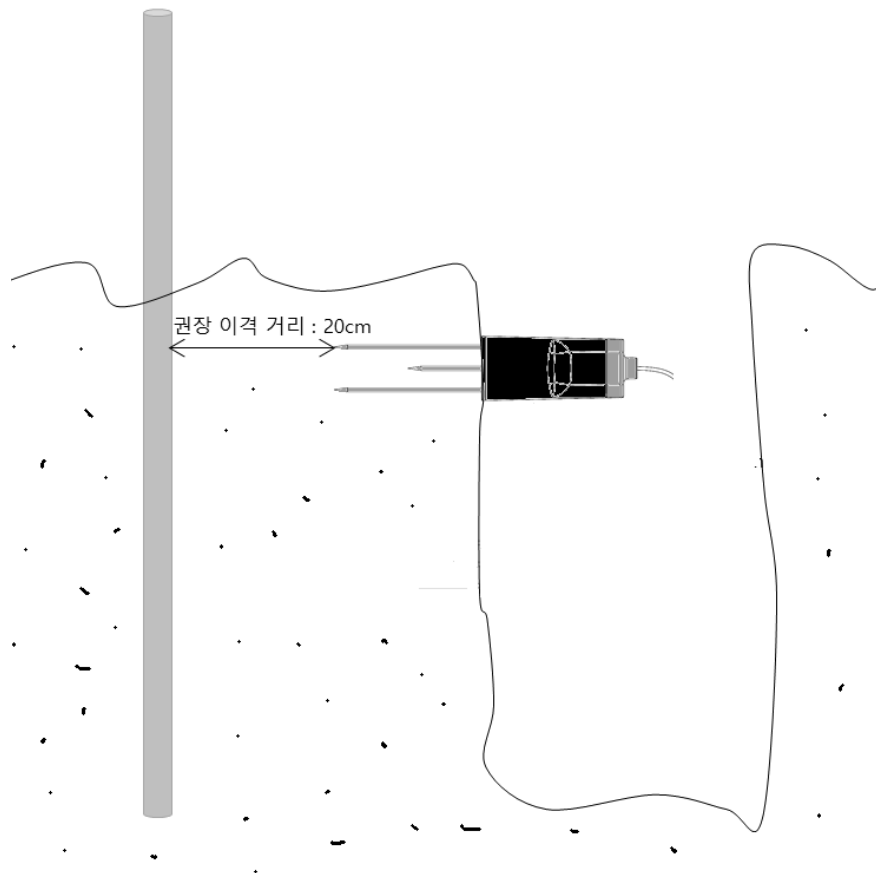
- 센서를 삽입할 때에는 센서가 흔들리지 않도록 삽입하여야 합니다. 센서가 흔들리면서 발생하는 공기 공간은 측정값의 오차를 발생 시킵니다.
- 완전히 마르고 단단한 토양에 센서를 삽입하면, 삽입 중 센서의 전극 주변에 토양이 깎이면서 공기 공간이 발생하여 오차를 발생 시킬 수 있으니, 토양에 물을 공급하여 충분히 적신 후 센서를 삽입하십시오.
- 센서의 주변에 금속성 물체가 있을 경우 금속성 물체가 전극 역할을 하게 되므로 측정값의 오차가 발생할 수 있습니다.

센서를 설치하는 위치는 가능한 금속성 물체를 멀리 하십시오.

(권장 이격 거리 : 20cm이상)



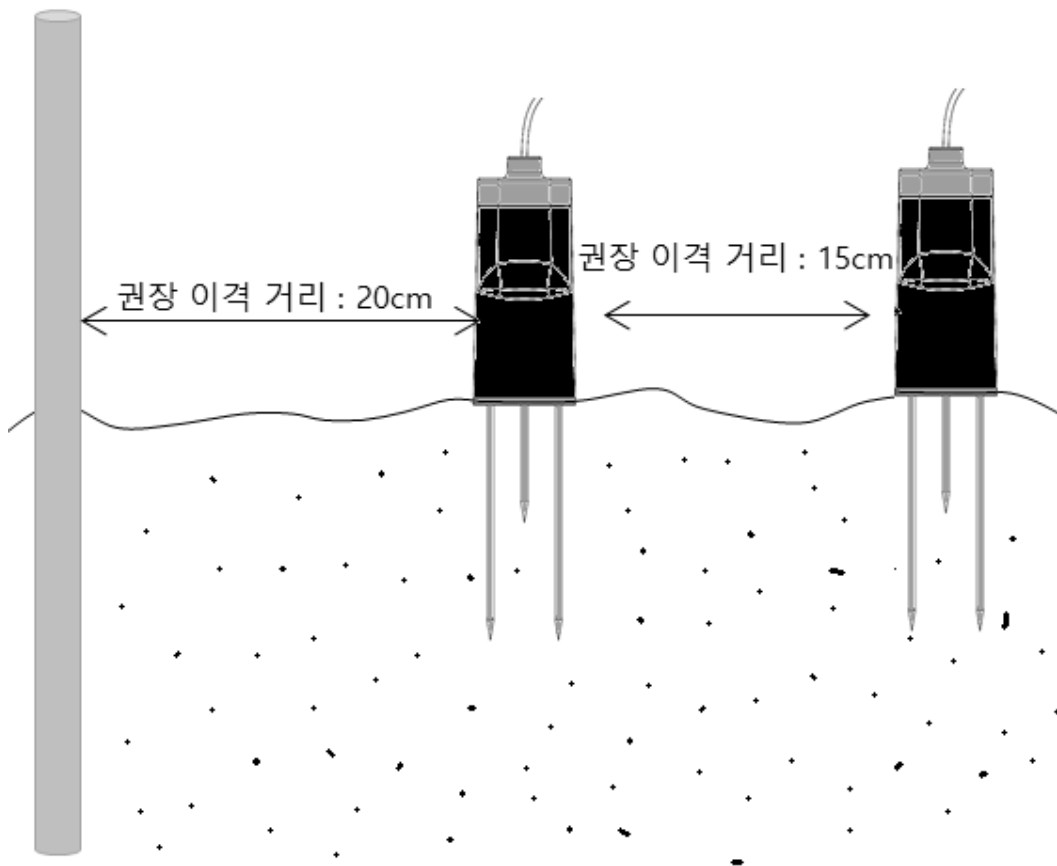
[설치 예제 1.]



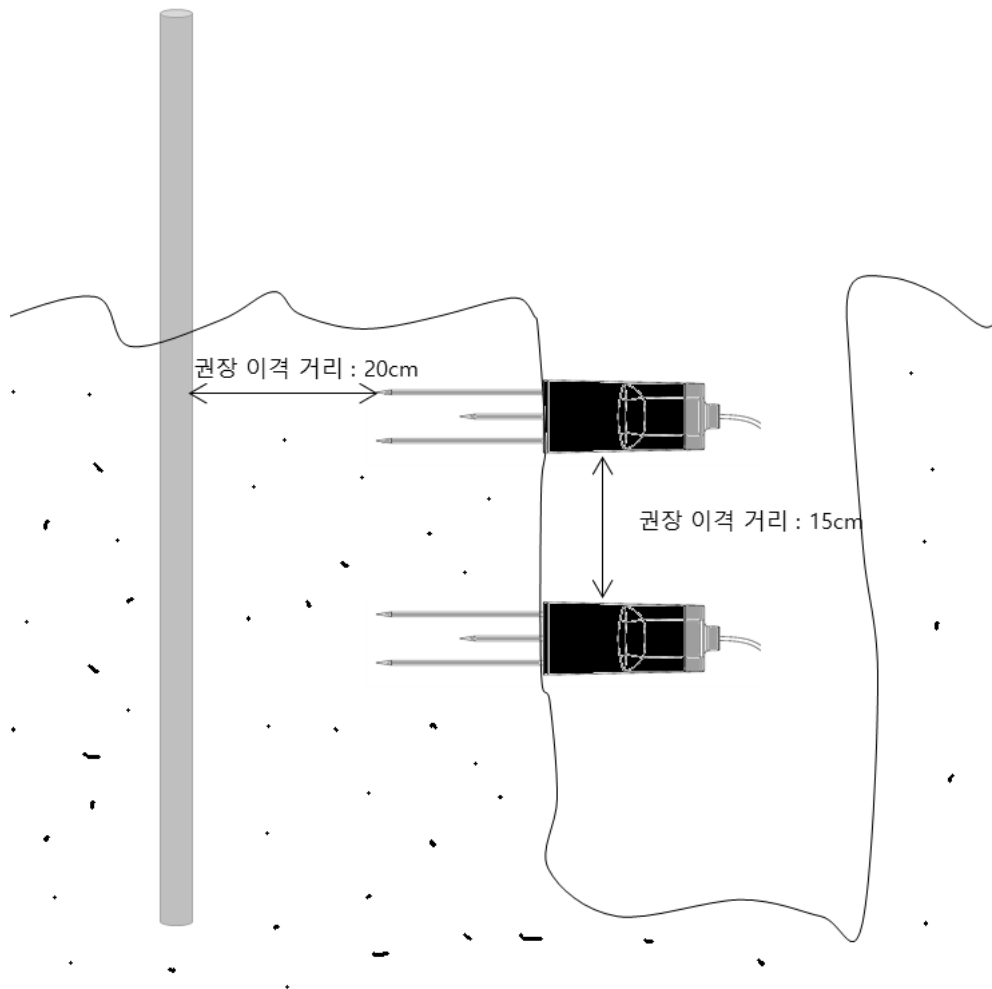
[설치 예제 2.]

## 다중 설치

- 다중 설치의 경우도 단독 설치와 같이 오차를 줄이기 위해 센서의 삽입 시 공기 공간이 발생하지 않도록 주의하여 토양에 삽입하고, 센서 이외의 금속성 물체는 권장 이격 거리인 20cm 이상 떨어뜨려 놓습니다.
- 센서간 신호 간섭을 줄이기 위해 센서 간 이격 거리는 권장 이격 거리인 15cm 이상 떨어뜨려서 설치하도록 합니다.



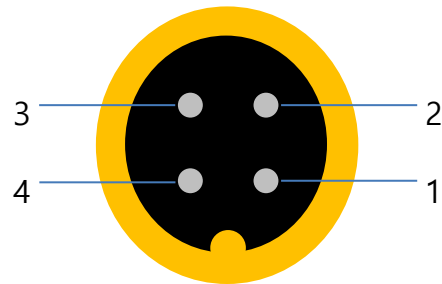
[설치 예제 3.]



[설치 예제 4.]

## 연결

### M12 Connector Type



1 = 5Vdc

2 = RS-485 +

3 = RS-485 -

4 = GND

### Strip Wire Type



적색 = 5Vdc

녹색 = RS-485 +

흰색 = RS-485 -

흑색 = GND

## CHAPTER 4

### 운용

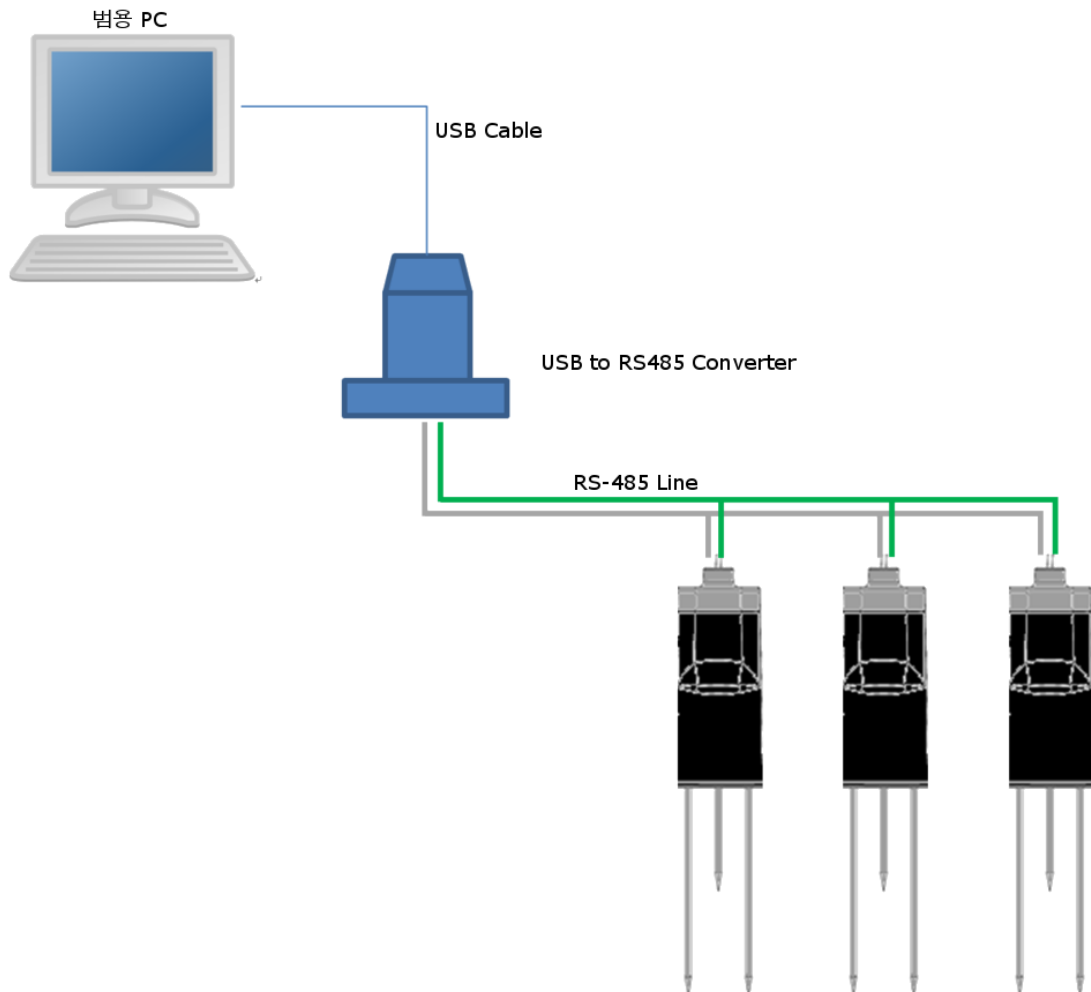
이 장에서는 KOREADIGITAL KSM8900 토양 수분 센서를 작동시키는데 필요한 정보를 제공합니다.

### 시작하기

KSM8900 토양 수분 센서에 5Vdc의 전원이 연결되면, RS-485 통신 데이터 라인을 통해 센서와 통신을 할 수 있습니다.

### RS-485 통신 연결

RS-485 전용 Port가 없는 범용 PC에 사용할 경우 USB to RS485 Converter 를 통해 PC와 KSM8900 센서를 연결해야 합니다.





**NOTE**

KOREADIGITAL(주) 스마트팜 컨트롤러를 사용하거나 별도의 RS-485 지원 컨트롤러를 사용할 경우 USB to RS485 Converter를 사용할 필요가 없습니다.

**NOTE**

다수의 KSM8900 센서를 연결하거나, 전기적 잡음이 심한 환경, KSM8900 센서와 HOST 시스템 간 거리가 먼 경우 120Ω의 종단 저항을 연결 하십시오.

**NOTE**

다수의 KSM8900 센서를 연결하거나, KSM8900 센서와 HOST 시스템 간 거리가 먼 경우 RS-485 드라이빙 전류가 부족하지 않도록 HOST 시스템을 구성하십시오.

## RS-485 통신 설정

### NOTE

KOREADIGITAL(주) 스마트팜 컨트롤러를 이용하실 경우 RS-485 통신 설정에 관한 부분은 컨트롤러에 이미 설정되어 있으므로 숙지할 필요가 없습니다.

RS-485 통신을 범용 PC에 연결하거나, 별도의 RS-485 지원 컨트롤러를 사용할 경우 아래와 같이 통신을 위한 설정이 필요합니다.

설정	값
<b>Bauds</b>	115,200 bps
<b>Parity</b>	None
<b>Data bits</b>	8bit
<b>Stop bits</b>	1bit
<b>Flow control</b>	None

## 통신 프로토콜

### NOTE

KOREADIGITAL(주) 스마트팜 컨트롤러를 이용하실 경우 RS-485 통신 프로토콜에 관한 부분은 컨트롤러에 이미 프로그램 되어 있으므로 반드시 숙지할 필요가 없습니다.

- KSM8900 센서의 측정값을 읽기 위해서는 전용 프로토콜을 이용하여 값을 읽을 수 있습니다.
- 전용 프로토콜은 ASCII 기반의 패킷 단위로 구성되어 있으며, 요청->응답의 구조로 통신을 할 수 있습니다.

### 패킷 구성

- 모든 패킷은 패킷 시작/종료 기호와 명령 접두사, CHECKSUM 접두사를 가지고 있습니다.
- 속성 접두사는 선택한 명령에 따라 다른 속성값을 선택적으로 사용합니다.
- 전송 패킷과 수신 패킷 모두 동일한 구조를 가집니다.

예시) {CMD=21,SMO= 0.0,STP= 25.0,SEC= 0.00,CHK=2204}

구성 기호	설명
{	패킷 시작
}	패킷 종료
=	등호
,	속성 구분자
CMD	명령 접두사
SMO,STP,SEC,ADDR,DADDR,FLOAT,STR,INT	속성 접두사
CHK	CHECKSUM 접두사

## 전송 명령

명령 접두사와 등호에 이어서 구성되는 십진수 값이 전송하고자 하는 명령이며, 명령 목록은 아래와 같습니다.

CMD	설명
20	KSM8900 센서 측정값 읽기
312	센서 이름 읽기
314	RS-485 통신 ID 읽기
316	RS-485 통신 ID 쓰기
508	레지스터 값 읽기
514	레지스터 값 쓰기
390	설정 값 저장

### NOTE

센서 이름 읽기 명령의 경우 모든 KSM8900 토양 수분 센서는 "SoilMoistureSensor" 로 동일한 이름을 가지고 있습니다.

### NOTE

RS-485 통신 ID 쓰기 명령이나 레지스터 값 쓰기 명령을 전송한 후 반드시 설정 값 저장 명령을 전송해야만 설정 값이 KSM8900 내부 메모리에 저장됩니다.

## 응답 명령

KSM8900 센서에 전송되는 모든 명령은 정상적으로 수신되고 처리되었을 때 그에 대한 응답 명령을 전송합니다. 전송 명령에 따른 응답 명령 목록은 아래와 같습니다.

전송 CMD	응답 CMD	설명
20	21	센서 측정값 읽기
312	313	센서 이름 읽기
314	315	RS-485 통신 ID 읽기
316	10	RS-485 통신 ID 쓰기
508	509	레지스터 값 읽기
514	10	레지스터 값 쓰기
390	10	설정 값 저장

## 전송 명령 속성

패킷의 속성 부분은 전송되는 명령에 따라 다르게 구성될 수 있습니다.  
전송 명령에 따른 전송 패킷의 속성은 아래와 같습니다.

전송 CMD	속성	설명
20	-	센서 측정값 읽기
312	-	센서 이름 읽기
314	-	RS-485 통신 ID 읽기
316	DADDR	RS-485 통신 ID 쓰기
508	ADDR	레지스터 값 읽기
514	ADDR,FLOAT	레지스터 값 쓰기
390	-	설정 값 저장

DADDR : 변경하고자 하는 Sensor ID (Default=1)

ADDR : 센서 레지스터 주소.

FLOAT : 센서 레지스터에 저장할 값.

### NOTE

속성값이 없는 명령의 경우 명령(CMD)과 Checksum(CHK)만 전송합니다.

### NOTE

레지스터 값 쓰기 명령과 RS-485 통신 ID 쓰기 명령을 전송한 후 설정 값 저장 명령을 전송하여야, 센서의 전원이 켜다 꺼지더라도 값을 유지할 수 있습니다.

- 센서 내부에는 특정한 값을 저장할 수 있는 레지스터가 있습니다.
- 각 레지스터는 특정한 목적이 이미 정해져 있으며, 각 레지스터의 목록과 설명은 아래와 같습니다.

레지스터 주소	설명
1	Soil Moisture 오프셋
2	Bulk EC 오프셋
3	Solution EC 오프셋
4	지중 온도 오프셋
5	EC 선택 0 : Solution EC 선택 1 : Bulk EC 선택

## CAUTION

지정된 레지스터가 아닌 다른 레지스터를 지정할 경우 센서가 오작동 할 수 있으니 레지스터 지정 시 주의하고, 다른 명령 전송으로 통신 확인이 된 후 레지스터 값 변경 명령을 수행해 주십시오.

## NOTE

EC의 경우 Bulk EC와 Solution EC 를 선택 측정 할 수 있습니다.  
Bulk EC : 토양에서 측정되는 실제 EC값  
Solution EC : Bulk EC 값에 지중 온도와 토양 수분값을 적용하여 침출수 EC값으로 환산한 값

## 응답 명령 속성

전송 패킷에 따라 KSM8900 센서의 응답 패킷의 속성은 다르게 구성될 수 있습니다. 전송 명령에 따른 응답 패킷의 속성 구성은 아래와 같습니다.

응답 CMD	속성	설명
21	SMO,STP,SEC	센서 측정값 읽기
313	STR	센서 이름 읽기
315	-	RS-485 통신 ID 읽기
509	FLOAT	레지스터 값 읽기
10	DADDR	RS-485 통신 ID 쓰기
	FLOAT	레지스터 값 쓰기
	INT	설정 값 저장

SMO : 토양 수분 값 (%VWC)

STP : 지중 온도 값 (°C)

SEC : EC값 (dS/m)



## 다중 센서 사용시 센서 선택

- 단독 센서 사용시에는 RS-485 통신 ID를 지정하지 않아도 통신이 가능합니다.
- 다중 센서 사용시에는 센서마다 각각 다른 RS-485 통신 ID를 지정하여야 합니다.
- RS-485 통신 ID는 Default 값으로 "1"의 값을 가지고 있으며, "316" 명령으로 값을 변경할 수 있습니다.
- KSM8900 센서는 자신의 RS-485 통신 ID를 호출하거나, 0번 ID를 호출할 때에만 응답 패킷을 전송합니다.
- RS-485 통신 ID를 지정해서 호출할 때, ID를 지정하는 방법은 아래 예시와 같이 CMD 속성 뒤에 DADDR 속성을 통해 지정할 수 있습니다.

예시) {CMD=21,DADDR=4,SMO= 0.0,STP= 25.0,SEC= 0.00,CHK=2204}

- RS-485 통신 ID 읽기와 RS-485 통신 ID 쓰기 명령 패킷에서는 RS-485 통신 ID 지정할 수 없습니다. 만약 수행할 경우 연결된 모든 센서의 RS-485 통신 ID 가 변경 됩니다.

## Checksum 계산

- 전송 패킷과 응답 패킷을 포함하는 모든 패킷은 Checksum 정보를 추가하여 통신 오류에 대한 처리를 수행할 수 있습니다.
- Checksum의 값은 패킷 구성에서 CHK 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 아래와 같습니다.

*[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", "CHK", 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;
}
```

## RS-485통신 송수신 패킷 예제

- 센서 측정값 읽기(1)

전송 패킷	{CMD=20,DADDR=1,CHS=999}
	단독 센서 사용 또는 다중 센서 사용에서 RS-485 통신 ID 1번 센서의 측정값 요청
응답 패킷	{CMD=21,SMO= 42.8,STP= 21.6,SEC= 0.00,CHS=2236}
	토양 수분 42.8 %VWC, 지중 온도 21.6°C, EC 0.00dS/m

- 센서 측정값 읽기(2)

전송 패킷	{CMD=20, CHS=494}
	단독 센서 사용시 측정값 요청
응답 패킷	{CMD=21,SMO= 42.8,STP= 21.6,SEC= 0.00,CHS=2236}
	토양 수분 42.8 %VWC, 지중 온도 21.6°C, EC 0.00dS/m

- 센서 이름 읽기

전송 패킷	{CMD=312,CHS=546}
	센서 이름 요청
응답 패킷	{CMD=313,STR=SoilMoistureSensor,CHS=2798}
	센서 이름은 SoilMoistureSensor

- RS-485 통신 ID 읽기

전송 패킷	{CMD=314,CHS=548}
	센서의 RS-485 통신 ID 요청
응답 패킷	{CMD=315,DADDR=1,CHS=1054}
	센서의 RS-485 통신 ID는 1번

- RS-485 통신 ID 쓰기

전송 패킷	{CMD=316,DADDR=4,CHS=1058}
	센서의 RS-485 통신 ID를 4번으로 변경
응답 패킷	{CMD=10,DADDR=4,CHS=1001}
	센서의 RS-485 통신 ID 4번으로 변경 완료

- 토양 수분 오프셋 레지스터 값 변경

전송 패킷	{CMD=514,ADDR=1,FLOAT=-10.5,CHS=1707}
	토양 수분 오프셋 값을 -10.5로 변경
응답 패킷	{CMD=10,FLOAT=-10.500000,CHS=1453}
	토양 수분 오프셋 값이 -10.5로 변경 완료

- Bulk EC 오프셋 레지스터 값 변경

전송 패킷	{CMD=514,ADDR=2,FLOAT=0.1,CHS=1610}
	Bulk EC 오프셋 값을 0.1로 변경
응답 패킷	{CMD=10,FLOAT=0.100000,CHS=1355}
	Bulk EC 오프셋 값이 0.1로 변경 완료

- Solution EC 오프셋 레지스터 값 변경

전송 패킷	{CMD=514,ADDR=3,FLOAT=1.1,CHS=1612}
	Solution EC 오프셋 1.1로 변경
응답 패킷	{CMD=10,FLOAT=1.100000,CHS=1356}
	Solution EC 오프셋 1.1로 변경 완료

- 지중 온도 오프셋 레지스터 값 변경

전송 패킷	{CMD=514,ADDR=4,FLOAT=-0.3,CHS=1659}
	지중 온도 오프셋 -0.3으로 변경
응답 패킷	{CMD=10,FLOAT=-0.300000,CHS=1402}
	지중 온도 오프셋 -0.3으로 변경 완료

- EC 측정 선택 레지스터 값 변경

전송 패킷	{CMD=514,ADDR=5,FLOAT=0,CHS=1518}
	EC 측정 모드를 Solution EC로 선택
응답 패킷	{CMD=10,FLOAT=0.000000,CHS=1354}
	EC 측정 모드를 Solution EC로 선택 완료

- 레지스터 변경 값 저장

전송 패킷	{CMD=390,CHS=552}
	레지스터 변경 값 저장
응답 패킷	{CMD=10,INT=390,CHS=989}
	레지스터 변경 값 저장 완료

## RS-485통신 송신 예제 코드

```

//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}WrWn", "CHK", 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;
}

//단독 센서 측정값 읽기 전송 패킷 만들기
char *msgbuf = CreateMessage(20);
//단독 센서 측정값 읽기 패킷 전송
RS485_send(SENSOR485_UART1, msgbuf);

```

## RS-485통신 수신 예제 코드

```
//응답 패킷 Checksum 확인하기
int VerifyAndParsing(uint8_t *ReceiveBuffer)
{
    char chksumstring[0x10];
    int receviechecksum = 0;
    int computechecksum = 0;
    int sindex = findstr_index((char*) ReceiveBuffer, "CHS", 0);
    if (sindex > 0)
    {
        computechecksum = compute_checksum((char *) ReceiveBuffer, (sindex - 1));
        if (getMsgparsing_pair_value((char *) ReceiveBuffer, "CHS", chksumstring,
            0x10, '}') == TRUE)
        {
            receviechecksum = atoi(chksumstring);
        }
        else
        {
            return FALSE;
        }
    }
    else
    {
        return FALSE;
    }
    if (computechecksum == receviechecksum)
    {
        return TRUE;
    }
    else
    {
        return FALSE;
    }
    return FALSE;
}
```

**//응답 패킷에서 특정 속성 접두사나 명령 접두사의 값 찾기**

```
int getMsgparsing_pair_value(char *msgstr, char *prefixstr, char *valuestr, int
valuestr_size, char terminalchar)
{
    int msglength = strlen(msgstr);
    int sindex = findstr_index(msgstr, prefixstr, 0);
    int si = 0;
    if (sindex > 0)
    {
        sindex = sindex + strlen(prefixstr) + 1;
        si = 0;
        for (int i = sindex; i < msglength; i++)
        {
            if (msgstr[i] == terminalchar)
            {
                valuestr[si++] = 'W0';
                break;
            }
            else
            {
                valuestr[si++] = msgstr[i];
            }
            if (si >= valuestr_size)
            {
                break;
            }
        }
    }
    else
    {
        return FALSE;
    }
    return TRUE;
}
```

```
//응답 패킷에서 특정 속성 접두사나 명령 접두사의 위치 찾기
int findstr_index(char *m_string, char *findstring, int startp)
{
    int mlength = strlen(m_string);
    int flength = strlen(findstring);
    for (int i = startp; i < (mlength - flength); i++)
    {
        int scout = 0;
        for (int j = 0; j < flength; j++)
            if (m_string[i + j] == findstring[j])    scout++;
            if (scout >= flength)    return i;
    }
    return -1;
}

//응답 패킷 파싱하기
int SensorCommParsingAndProcessing(char *rmsgBuffer, int rmsgCount)
{
    char tempstr[0x40];
    if(VerifyAndParsing(rmsgBuffer)==TRUE)
    {
        if(getMsgparsing_pair_value(rmsgBuffer, "CMD", tempstr, 0x40, ',') == TRUE)
        {
            int cmd = atoi(tempstr);
            if(cmd==21)
            {
                if(getMsgparsing_pair_value(rmsgBuffer, "SMO", tempstr, 0x40, ',') ==
                    TRUE)
                    float soilmoisture = atof(tmpstr);
                if(getMsgparsing_pair_value(rmsgBuffer, "SEC", tempstr, 0x40, ',') ==
                    TRUE)
                    float EC = atof(tmpstr);
                if(getMsgparsing_pair_value(rmsgBuffer, "STP", tempstr, 0x40, ',') ==
                    TRUE)
                    float Temperature = atof(tmpstr);
            }
        }
    }
}
```



## MODBUS RTU 통신 프로토콜

### NOTE

KOREADIGITAL(주) 스마트팜 컨트롤러를 이용하실 경우 RS-485 통신 프로토콜에 관한 부분은 컨트롤러에 이미 프로그램 되어 있으므로 반드시 숙지할 필요가 없습니다.

- 이전 단원에 설명된 통신 프로토콜 또는 MODBUS RTU 통신 프로토콜, 둘 중 하나만 구현하여도 KSM-8900을 이용하실 수 있습니다.
- MODBUS RTU는 KSM-8900을 다른 마스터 장치에 편리하게 연결하기 위한 일반화된 데이터 상호 교환 포맷을 제공하기 위해 구현되었습니다..

### 마스터 전송 패킷 구성

공백	장치ID	기능코드	시작어드레스	데이터	CRC
>3ms	1~99	04 or 06	X	X	CRC-16
	1byte	1byte	2byte	2 byte	2byte

-공백 : MODBUS RTU 통신은 3.5byte 전송 시간을 공백으로 유지하여 패킷을 구분합니다. KSM-8900은 3ms이상의 공백을 패킷 구분으로 인식합니다.

-장치ID : MODBUS RTU 통신은 장치ID를 통해 선택된 슬레이브 장치만이 응답할 수 있으며, KSM-8900은 1~100까지 장치ID를 변경할 수 있습니다. 장치의 기본 ID는 1로 설정되어 있습니다.

KSM-8900은 BROADCASTING ADDRESS(0)은 지원하지 않습니다.

-기능코드 : KSM-8900은 마스터의 요구 동작을 파악하기 위한 표준 MODBUS RTU 프로토콜의 기능코드 중 04(Input Resistors Read)와 06(Preset Single Resister) 두 가지 만을 지원합니다.

-시작어드레스 : KSM-8900에서 값을 읽을 레지스터 또는 값을 쓸 레지스터를 지정합니다.(16bit Integer)

04(Input Resisters Read) 기능 코드 사용시 Float Type Data의 Address는 1~6까지 유효 Address이나, 하나의 파라미터가 상위2byte, 하위 2byte로 구분되어 있으므로, 반드시 Address는 홀수(1,3,5)로 지정하도록 합니다. Unsigned Int Type Data의 Address는 7~9까지 유효 Address입니다. Unsigned Int Type Data는 Float Type Data에 100을 곱한 값이므로, 값을 읽은 뒤 100을 나누어 주어야 합니다.

06(Preset Single Resister) 기능 코드 사용시 Address는 10,11까지 유효 Address입니다.

Resister Address(Decimal)	Description
1	토양 수분 (부동 소수점 형, 상위 2byte)
2	토양 수분 (부동 소수점 형, 하위 2byte)
3	토양 온도 (부동 소수점 형, 상위 2byte)
4	토양 온도 (부동 소수점 형, 하위 2byte)
5	토양 EC (부동 소수점 형, 상위 2byte)
6	토양 EC (부동 소수점 형, 하위 2byte)
7	토양 수분 * 100 (Unsigned int형)
8	토양 온도 * 100 (Unsigned int형)
9	토양 EC * 100 (Unsigned int형)
10	장치 ID
11	통신 Baud rate (기본 115200bps)

-데이터 : 마스터에서 KSM-8900으로 전송되는 패킷에서 데이터 영역은 모두 2Byte(16bit Integer)입니다.

06(Preset Single Resister) 기능 코드를 사용하여 장치ID를 변경하고자 할 경우 변경하고자 하는 장치ID의 값이 기록되는 곳입니다.

06(Preset Single Resister) 기능 코드를 사용하여 통신 Baud rate를 변경하고자 할 경우 변경하고자 하는 Baud rate의 인덱스 값이 기록되는 곳입니다.

Index	Baud rate
1	115200bps
2	57600bps
3	38400bps
4	19200bps
5	9600bps

변경된 Baud rate와 장치ID는 KSM-8900의 전원이 제거되어도 변경되지 않습니다.

04(Input Resister Read) 기능 코드를 사용하여 Float Type Data형으로 KSM-8900의 측정값을 읽고자 할 경우 시작어드레스의 값과 데이터에 들어가는 값의 합이 7을 넘지 않도록 값을 지정하며, 반드시 데이터에 들어가는 값은 짝수만을 지정합니다.

시작 어드레스	데이터	응답 데이터
1	6	토양 수분, 토양 온도, 토양 EC 모두 응답
1	4	토양 수분, 토양 온도 응답
1	2	토양 수분 응답
3	4	토양 온도, 토양 EC 응답
3	2	토양 온도 응답
5	2	토양 EC 응답

-CRC : 장치ID에서부터 데이터까지 모든 값에 대해 CRC-16 에 따라 에러 검출을 하여 통신 에러 검사에 사용됩니다..

**NOTE**

장치ID나 통신 Baud rate를 변경한 경우 정확한 통신을 위해 KSM-8900의 전원을 제거했다가 다시 연결해 주십시오.,

## 마스터 수신 패킷 구성 - 기능코드 04(Input Resisters Read)

기능코드 04(Input Resister Read) 명령에 대한 슬레이브 응답 패킷은 KSM-8900 측정값으로 구성됩니다.

장치ID	기능코드	바이트 수	데이터	CRC
1~99	04	X	X	CRC-16
1byte	1byte	1byte	N byte	2byte

-장치ID : 마스터에서 호출한 KSM-8900 자신의 ID입니다.

- 기능코드 : 마스터에서 전송한 기능코드입니다.

- 바이트 수 : 데이터 영역의 바이트 개수입니다.

KSM-8900에서는 측정값이 모두 4byte 부동 소수점 형이므로 마스터 전송 패킷의 데이터 영역에 있는 값\*2의 값이 지정됩니다.

- 데이터 : 마스터에서 전송한 패킷에 따라 지정된 레지스터에서 측정값을 읽어서 지정합니다. 마스터 전송 패킷의 데이터 영역에 있는 값에 따라 구성되는 바이트 수는 유동적입니다.

-CRC : 장치ID에서부터 데이터까지 모든 값에 대해 CRC-16 에 따라 에러 검출을 하여 통신 에러 검사에 사용됩니다.

## 마스터 수신 패킷 구성 - 기능코드 06(Preset Single Resister)

KSM-8900의 장치ID변경이나, 통신 Baud rate 변경시 사용되는 기능코드 06(Preset Single Resister) 명령에 대한 슬레이브 응답 패킷은 마스터 전송 패킷과 동일합니다.

## 예외 패킷 구성

KSM-8900은 Modbus 통신시 몇 가지 잘못된 통신 상황에 대해 예외 응답 패킷을 마스터로 전송합니다.

패킷의 구성은 아래와 같습니다.

장치ID	기능코드	예외코드	CRC
1~99	0x80   기능코드	1 or 2 or 3	CRC-16
1byte	1byte	1byte	2byte

-장치ID : 마스터에서 호출한 KSM-8900 자신의 ID입니다.

- 기능코드 : (예외 발생을 표기하는 0x80 + 마스터에서 전송한 기능코드)를 지정합니다.

- 예외 코드

예외 코드	상황
0x01	마스터에서 전송한 패킷의 기능코드 영역이 잘못된 경우
0x02	마스터에서 전송한 패킷의 시작어드레스 영역이 잘못된 경우
0x03	마스터에서 전송한 패킷의 데이터 영역이 잘못된 경우

-CRC : 장치ID에서부터 데이터까지 모든 값에 대해 CRC-16 에 따라 에러 검출을 하여 통신 에러 검사에 사용됩니다.

아래의 경우는 예외 패킷을 포함한 어떠한 응답도 전송하지 않는 경우입니다.

- 장치ID가 KSM-8900 자신의 장치ID와 다를 경우
- CRC-16의 값이 잘못되어 통신 오류로 판단된 경우

## MODBUS RTU 통신 송수신 패킷 예제

- 장치ID 1을 가지는 KSM-8900 센서에 토양 수분, 토양 온도, 토양 EC값 요청 (Float Type Data)

전송 패킷	01 04 00 01 00 06 21 C8
	장치ID 1에 1번 어드레스부터 6개의 연속된 레지스터 읽기
응답 패킷	01 04 0C 41 F7 CB 24 41 C0 00 00 3F 18 41 74 16 70
	12bytes 데이터로 구성된 응답
	토양수분 30.97419 %VWC
	토양온도 24 °C
	토양EC 0.59 dS/m

- 장치ID 1을 가지는 KSM-8900 센서에 토양 수분, 토양 온도, 토양 EC값 요청 (Unsigned Int Data)

전송 패킷	01 04 00 07 00 03 01 CA
	장치ID 1에 1번 어드레스부터 3개의 연속된 레지스터 읽기
응답 패킷	01 04 06 13 41 0D 28 00 0F 9C 9F
	6bytes 데이터로 구성된 응답
	토양수분 4929 => 49.29 %VWC
	토양온도 3368 => 33.68 °C
	토양EC 15 => 0.15 dS/m

- 장치ID 1을 가지는 KSM-8900 센서의 ID를 2로 변경

전송 패킷	01 06 00 0A 00 02 28 09
	장치ID 1에 10번 레지스터에 2를 쓰기
응답 패킷	01 06 00 0A 00 02 28 09
	장치ID가 1에서 2로 변경됨(전송 패킷 반송)

- 장치ID 1을 가지는 KSM-8900 센서의 통신 Baud rate를 9600bps로 변경

전송 패킷	01 06 00 0B 00 05 38 0B
	장치ID 1에 11번 레지스터에 5를 쓰기
응답 패킷	01 06 00 0B 00 05 38 0B
	통신 Baud rate 가 9600으로 변경됨(전송 패킷 반송)

## CHAPTER 5

### 유지 보수

이 장에서는 제품의 기본적인 유지 보수에 필요한 정보를 제공합니다.

#### 청소

- 중성 세제를 적신 부드럽고, 보푸라기가 없는 천으로 닦아 주십시오.
- 센서의 전극은 강한 힘을 주지 않도록 주의해서 청소해 주십시오.

#### 보관

- 사용한 센서는 보관하기 전 깨끗이 청소해 주십시오.
- 전극은 포장재의 스티로폼을 이용해 보호해 주십시오.

#### 기술 지원

추가적인 기술적 지원은 [info@koreadigital.com](mailto:info@koreadigital.com) 으로 다음의 정보와 함께 연락을 해 주십시오.

- 모델명과 증상
- 센서를 설치한 위치에 대한 정보
- 제품 구매 시기
- 제품을 구매한 업체나 대리점 정보
- 연락 가능한 연락처

CHAPTER 6

기술 데이터

측정 범위	토양 수분	0 ~ 50 %VWC
	전기 전도도	0 ~ 10 dS/m
	지중 온도	-40 ~ 60 °C
Accuracy (@25°C)	토양 수분	±3 %VWC
	전기 전도도	±0.1 d/Sm (0~1dS/m) ±10% (1~10dS/m)
	지중 온도	±1 °C
온도 보정 계수	EC Automatic $\beta=2\%/^{\circ}\text{C}$	
분해능	토양 수분	0.1 %VWC
	전기 전도도	0.01 dS/m
	지중 온도	0.1 °C
보관 환경	-20 ~ 80 °C, Max 95%RH	
운영 환경	-40 ~ 60 °C, Max 95%RH	
측정 방식	토양 수분	Frequency Domain Reflectometry
	전기 전도도	Impedance
	지중 온도	MEMS
정격	전원	5Vdc
	소비 전력	70mA @5Vdc
	통신	RS-485
치수	외형	226mm * Ø 60 (전극 길이 : 110mm)
	무게	360g (표준 케이블 포함)
	케이블	표준 3Meter
보증 기간	2년	