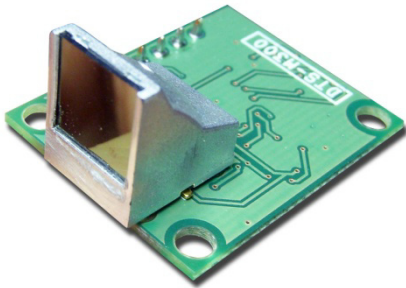


DTS-M300 Specification



- 비접촉 온도 측정
- 근거리 온도 측정
- High Accuracy
- 저전류 동작(typ. 600uA)
- Digital Interface : SPI
- Arduino UNO 예제 코드 제공

▶ 제품 설명

- DTS-M300는 접촉을 하지 않고 원하는 물체 표면에 온도를 500ms 이내에 정확하게 측정할 수 있는 온도센서모듈입니다.
- DTS-M300는 온도계산 프로세서를 내장하고 있어 정확한 온도 값을 출력합니다.
(Master Controller에 온도계산 알고리즘이 필요하지 않습니다.)
- DTS-M300는 디지털 통신(SPI)으로 온도값을 출력합니다.
- 센서 온도와 대상 온도를 동시에 측정합니다.

▶ 특징

- 측정 온도 구간 : -30 °C ~ 300°C
- 동작 온도 구간 : -20 °C ~ 70°C
- 분해능 : 0.01°C
- DS ratio : 15:1
- 정확도 : ±2%
- 입력 전압 : 3.3V
- 1mA 이하의 저전류
- 통신 인터페이스 : SPI

▶ 응용분야

- 과열방지 시스템
- 산업용 온도 측정 장치
- 체온 측정을 통한 인체 감지
- 가전기기
- 전자레인지 음식 온도 감지

DTS-M300 Specification

▶ Absolute Maximum Ratings

- Supply voltage : 3.5V
- Operating Temperature Range : -20°C ~ 70°C
- Storage Temperature Range : -40°C ~ 85°C

위 조건을 넘어서게 되면 제품의 수명을 보장할 수 없습니다.

반드시 아래 Electrical Requirements 를 지켜주세요.

▶ Electrical Requirements

Parameter	Symbol	Conditions	min	Typ	Max	Unit
공급전압	Vcc	Measured versus GND	3.1	3.3	3.5	V
방사율(Emission Coefficient)	ϵ			0.95		
소비 전류 (3.3V 기준)		Full ambient temp. range, Typical value, no output load		0.6	1	mA
SPI Clock					1	MHz
INPUT High Level			3.1		3.5	V
INPUT Low Level					0.9	V
OUTPUT High Level			Vcc-0.3		Vcc	V
OUTPUT LOW Level			Vss		Vss+0.3	V

▶ Operational Characteristics

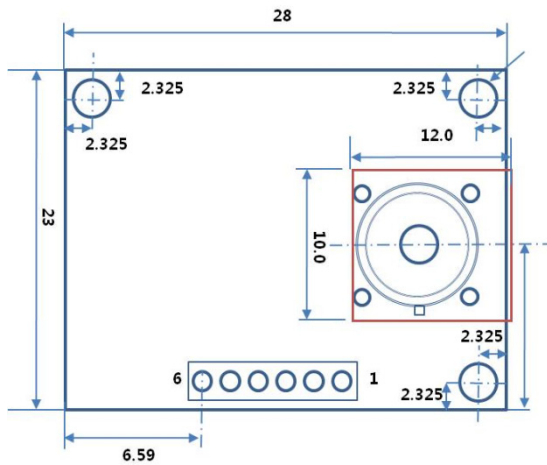
- if not otherwise noted, 25°C ambient temperature, 3.3V supply voltage and object with $\epsilon = 0.95$ were applied

Parameter	Symbol	Conditions	min	Typ	Max	Unit
DS ratio				8:1		
온도측정범위(타겟온도범위)	Tobj		-30		300	°C
동작온도(센서온도)	Tamb		-20		70	°C
온도측정 시간	Fout			0.5		sec
정확도	AccT			±2		%
Resolution Digital				0.01		°C
Standard Start-UP Time	tStart			3		sec
Stabilization Time	tStab			1		min

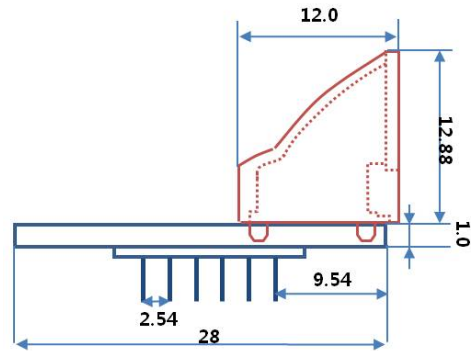
DTS-M300 Specification

▶ Mechanical Dimensions

단위 : mm



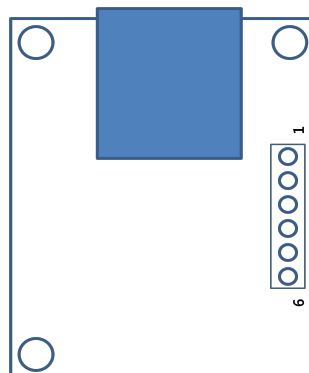
< Top View >



< Side View >

▶ Pin Assignment

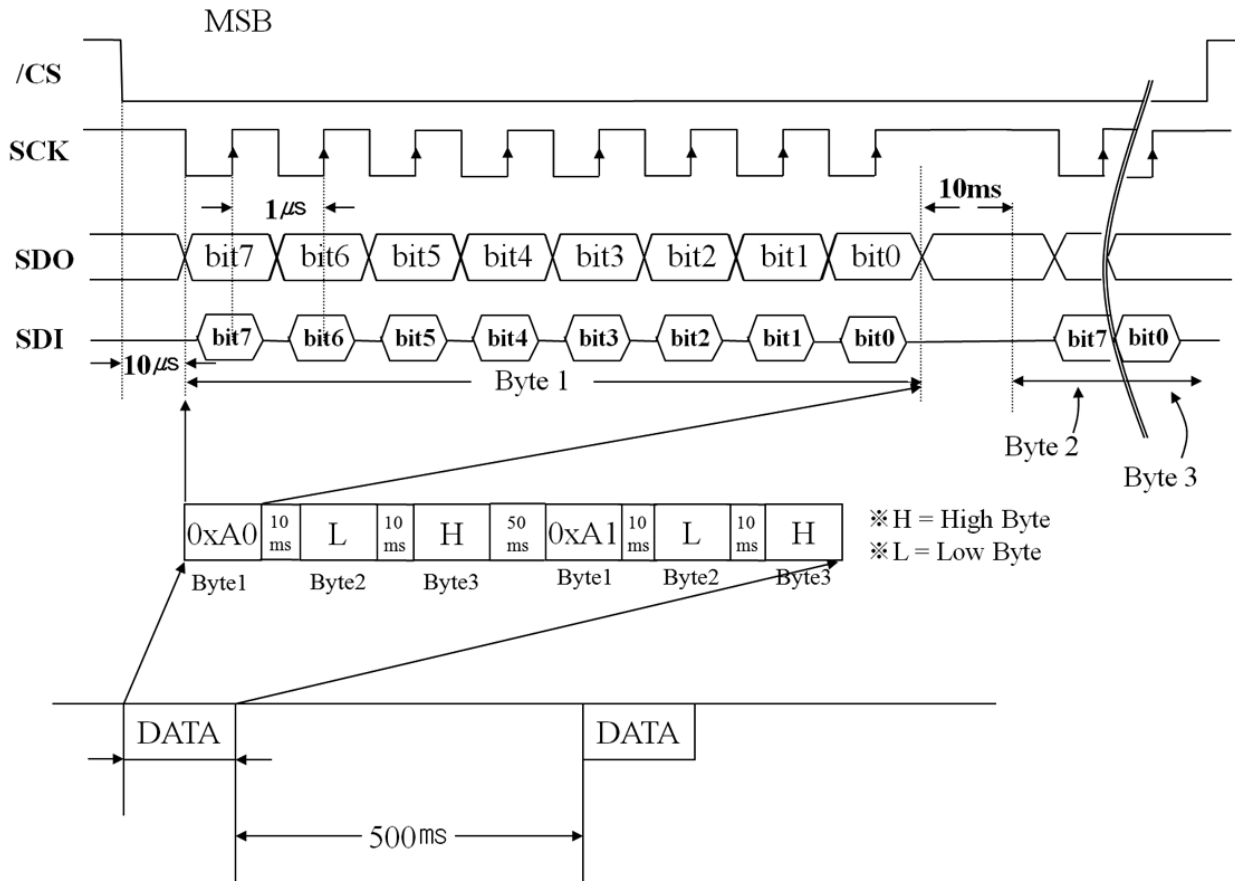
number	Name	Description	Type
1	GND	Ground	Ground
2	/CS or SCE	ENABLE	Input
3	SCK	CLOCK	Input
4	SDI	Signal Input	Input
5	SDO	Signal Output	Output
6	VCC	Supply Voltage	Supply



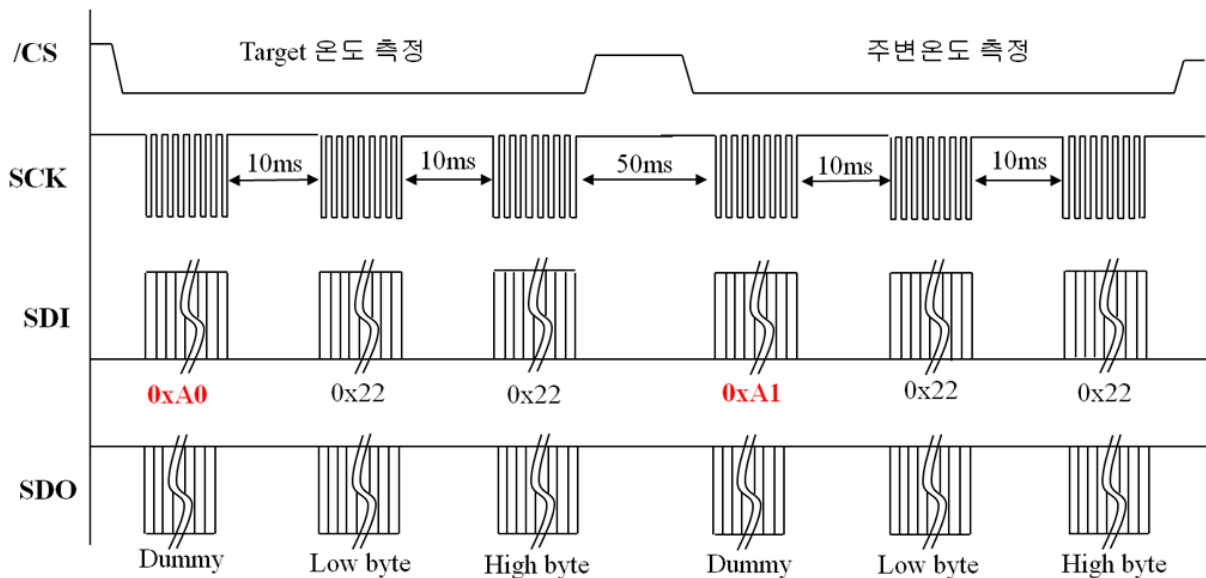
< Top View >

DTS-M300 Specification

▶ **SPI - interface(모듈은 SPI Slave Mode로 동작합니다.)**



- 온도 측정 프로토콜



DTS-M300 Specification

▶ 온도 계산 방법

- 영상온도 계산

0xA0	0x42	0x0E	0xA1	0xC4	0x09
------	------	------	------	------	------

* 타겟 온도 계산 : 상위 Byte(0x0E) + 하위 Byte(0x42) = 0x0E42
 => 3650(HEX → 10진수) 즉 36.50 도입니다.

* 주변 온도 계산 : 상위 Byte(0x09) + 하위 Byte(0xC4) = 0x09C4
 => 2500 (HEX → 10진수) 즉 25.00 도입니다.

- 영하온도 계산(영하(0도 미만)일 때는 2의 보수 값으로 전송됩니다.)

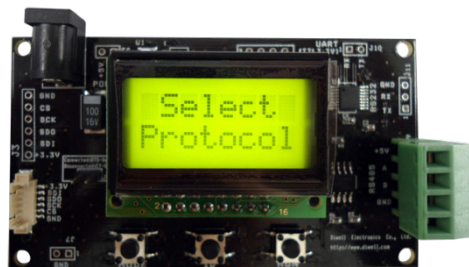
0xA0	0x6A	0xFF	0xA1	0x7A	0xFF
------	------	------	------	------	------

*타겟 온도 계산 : 상위 Byte(0xFF) + 하위 Byte(0x6A) = 0xFF6A = 350
 0xFF6A = 1111 1111 0110 1010 (1의 보수 값 + 1 의 연산을 합니다)
 0000 0000 1001 0101 □ 1의 보수값
 0000 0000 1001 0110 □ +1 = 0x0096
 0x0096 = 150 즉, -1.50도 입니다.

*주변 온도 계산 : 상위 Byte(0xFF) + 하위 Byte(0x7A) = 0xFF7A = -1.34도입니다.

▶ CT-Testboard-Plus 테스트 보드(별도 구매)

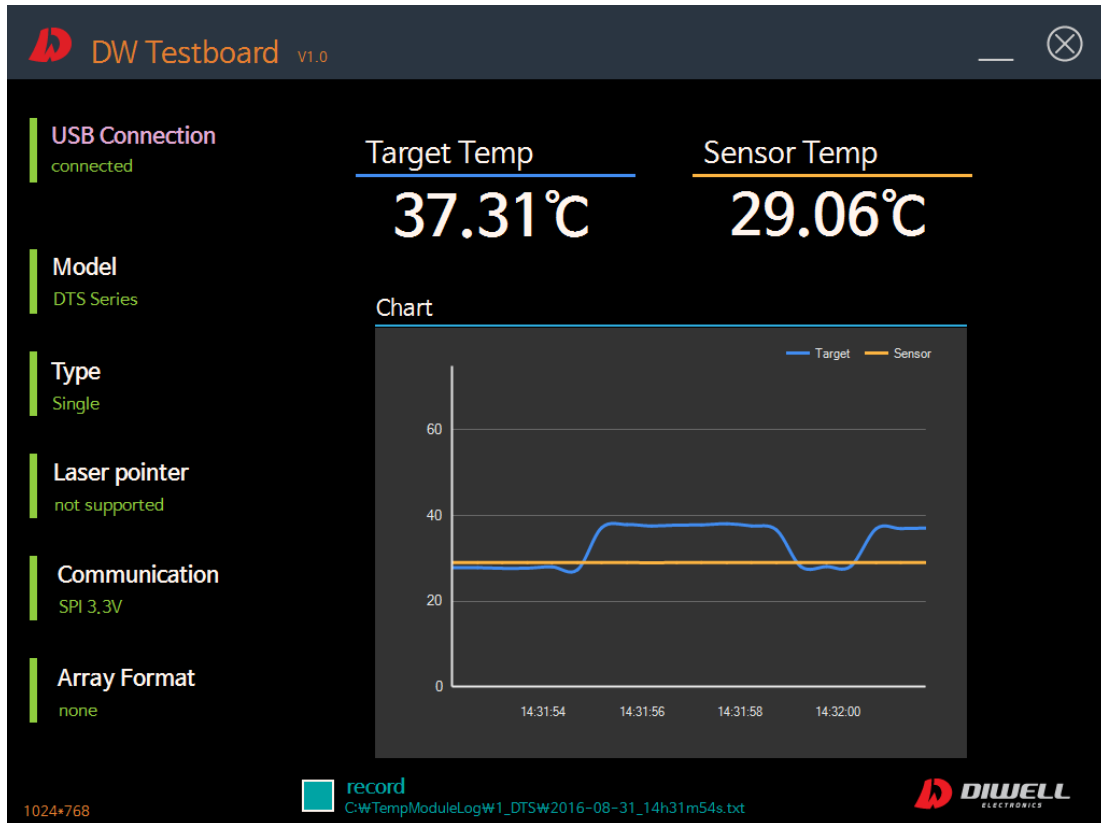
- 손쉽게 측정하여 온도값을 다른 device(MCU, PC, embedded등) 로 전송(RS-232)할 수 있습니다.



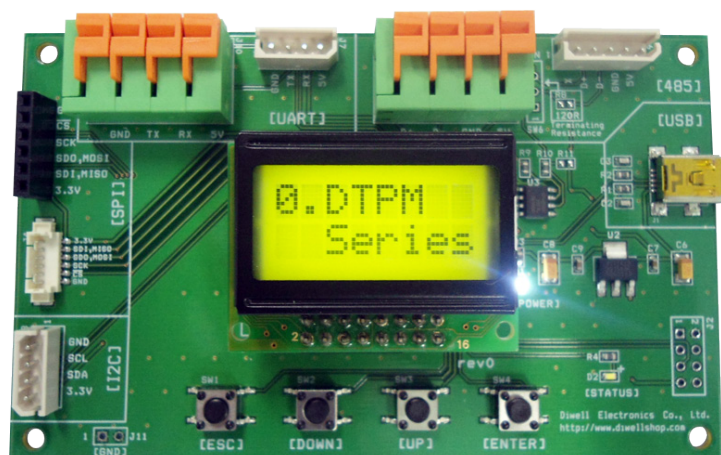
DTS-M300 Specification

▶ DW Testboard (신제품 - 별도 구매)

- PC와 연결하여 온도 측정/ 실시간 기록이 가능한 테스트보드 신제품이 출시됐습니다.



< PC 화면 >



< DW Testboard - 신제품 >

DTS-M300 Specification

▶ Additional Information

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

▶ Appendix - A (Example Code - Sonix SPI 레지스터 이용)

아래 소스코드는 DTS-M300 통신을 위한 참고용 소스 코드입니다.

SPI 초기값 세팅

- Clock 주파수 최대 1Mhz
- Internal SPI Clock(Master Mode)
- SCK data transfer edge : Rising Edge
- MSB first data transfer
- SCK idle status : High

사용하시는 환경에 따라 다르므로 내용을 이해 하신 후 적용하고자 하는 컴파일러에서 레지스터 설정해 주시면 됩니다. 레지스터 설정이 힘들다면 Appendix - B 의 IO 제어 코드와 Appendix - C 아두이노 코드를 참고하시길 바랍니다.

```
long Check_Temp(unsigned char datum) // Sonix 컴파일러에는 Long 이 2byte 입니다.
```

```
{  
    long temp_bank=0;  
    SIOB = datum; // Buffer 레지스터에 저장  
    NOP(1);  
    EN_LOW; // Enable Low  
    delay_us(10); // 10us delay  
    FSTART = 1; // SPI 전송 시작  
    while(FSTART==1); // 전송 완료 flag check  
  
    buffer2=SIOB; // dummy  
    SIOB = 0x22; // 0x22 or 0x00 가능합니다.  
    delay_ms(10); // 10ms delay  
    FSTART = 1; // SPI 전송 시작
```

DTS-M300 Specification

```
while(FSTART==1);    // 전송 완료 flag check

buffer=SIOB;        // 하위 Byte 저장
SIOB = 0x22;        // 0x22 or 0x00 가능합니다.
delay_ms(10);       // 10ms delay
FSTART = 1;         // SPI 전송 시작
while(FSTART==1);   // 전송 완료 flag check

buffer1=SIOB;       // 상위 Byte 저장
EN_HIGH;           // Enable High

temp_bank=(buffer1*256)+buffer; // 상위, 하위 Byte 온도 계산식.
// temp_bank : 최종 온도

return temp_bank;
}

void main(void)
{
    Long Target_Value, Ambient_Value; //부호 있는 2byte 선언
    Port_init();    // PORT 초기화
    LCD_init();     // LCD 초기화
    EN_HIGH;       // CS idle High
    SPI_init();    // SPI 초기화

    while(1)
    {
        Target_Value = Check_Temp(0xa0); // SPI 통신(타겟온도)
        delay_ms(50);
        Ambient_Value = Check_Temp(0xa1); // SPI 통신(주변온도)
        LCD_view();           // 온도값 LCD 표시
        delay_ms(500);       // 500 delay(필수)
        // 온도값 read 명령 후 다음 명령까지 500ms delay
    }
}
```


DTS-M300 Specification

▶ Appendix - B (Example Code - I/O)

하단 코드는 컨트롤러에서 SPI 레지스터 설정이 아닌 I/O 포트 제어를 통해 통신하는 예제 코드입니다. I/O 포트 전압 레벨이 3.3V 인지를 꼭 확인하세요.

- **SPI.H**

```
#define SCK_HIGH    FP16=1
#define SCK_LOW     FP16=0
#define SDO_HIGH   FP14=1
#define SDO_LOW    FP14=0
#define EN_HIGH    FP17=1
#define EN_LOW     FP17=0
long CHECK(unsigned char datum);
```

- **Main.C**

```
#include "SN8F27E65.h"
#include "delay.h"
#include "SPI.H"
Long Target_Value, Ambient_Value;
// sonix 컴파일러는 long이 2byte 입니다. 해당하는 컴파일러에 맞게 2byte변수 선언하세요
void main(void)
{
    While(1)
    {
        Target_Value = CHECK(0xa0); // 대상온도
        delay_ms(50);
        Ambient_Value = CHECK(0xa1) // 주변온도
        delay_ms(500);
        // LCD View CODE here
    }
}
```

● SPI.C

```

unsigned char buffer_Lo, buffer_Hi, p02; //1byte 선언

long CHECK(unsigned char datum) //2 byte return 함수
{
    unsigned char i=0;
    buffer_Lo=0;
    buffer_Hi = 0;
    EN_LOW;
    delay_us(10);
    for(i=0; i<8; i++)
    {
        if(((0x80 >>i)&datum)==0){SDO_LOW;}
        else {SDO_HIGH;}
        SCK_LOW;
        delay_us(1);
        SCK_HIGH;
        delay_us(1);
    }
    SDO_LOW; // 0x22 or 0x00 전부 가능합니다만 코드간격을 위해 0x00 사용
    delay_ms(10);

    //Low byte read
    for(i=0; i<8; i++)
    {
        buffer_Lo = buffer_Lo <<1;
        SCK_LOW;
        delay_us(1);
        SCK_HIGH;
        delay_us(1);
        p02=FP02; // 포트의 상태 읽는 문장
        if(p02 == 1){buffer_Lo = buffer_Lo|0x01;}
        else{buffer_Lo = buffer_Lo&0xFE;}
    }

    SDO_LOW;
    delay_ms(10);

    //High byte read
    for(i=0; i<8; i++)
    {
        buffer_Hi = buffer_Hi <<1;
        SCK_LOW;
        delay_us(1);
        SCK_HIGH;
        delay_us(1);
        p02=FP02; // 포트의 상태 읽는 문장
        if(p02 == 1){buffer_Hi = buffer_Hi|0x01;}
        else{buffer_Hi = buffer_Hi&0xFE;}
    }
    EN_HIGH;
    return (buffer_Hi*256+buffer_Lo);
}

```

▶ Appendix - C (Example Code - Arduino UNO)

하단 코드는 아두이노 UNO 코드 예제 입니다.

```

/*****
*
* Copyright (C) 2016 Diwell Electronics Co.,Ltd.
* Project Name : (DTS 시리즈) SPI Master Code
* Version : 1.0 (2016.05.04)
* SYSTEM CLOCK : 16Mhz
* BOARD : Arduino UNO. 5V operation

PORT Description

1. ChipSelectPin : 10
2. MOSI(Master Output) : 11
3. MISO(Master Input) : 12
4. SCK : 13
온도센서모듈 입력전원은 3.3V 로 하셔야 하며 포트 연결 방법은 회로도를 참고하십시오.
온도센서 통신포트의 논리 레벨은 3.3V 이기 때문에 반드시 회로도를 참고하시기 바랍니다.

Revision history.

1. 2016.5.4 : First version is released.
*****/

#include<SPI.h>

#define TARGET_CMD      0xA0           // 대상 온도 커맨드
#define SENSOR_CMD      0xA1           // 센서 온도 커맨드

const int chipSelectPin = 10;
unsigned char T_high_byte;
unsigned char T_low_byte;
int iTARGET, iSENSOR;    // 부호 2byte 온도 저장 변수

```

DTS-M300 Specification

```
void setup() {
  /* Initalize PORT */
  pinMode(MISO, INPUT);
  pinMode(chipSelectPin , OUTPUT);
  pinMode(MOSI, OUTPUT);
  pinMode(SCK, OUTPUT);

  Serial.begin(9600);

  /* Setting CS & SPI */
  digitalWrite(chipSelectPin , HIGH); // CS High Level
  SPI.setDataMode(SPI_MODE3); // Setting SPI Mode
  SPI.setClockDivider(SPI_CLOCK_DIV16); // 16MHz/16 = 1MHz
  SPI.setBitOrder(MSBFIRST); // MSB First
  SPI.begin(); // Initialize SPI
  delay(500); // waiting for DTS setup time
}

int SEND_COMMAND(unsigned char cCMD)
{
  digitalWrite(chipSelectPin , LOW); // CS Low Level
  delayMicroseconds(10); // delay(10us)
  SPI.transfer(cCMD); // Send 1st Byte
  delay(10); // delay(10ms)
  T_low_byte = SPI.transfer(0x22); // Send 2nd Byte
  delay(10); //delay(10ms)
  T_high_byte = SPI.transfer(0x22); // Send 3rd Byte
  digitalWrite(chipSelectPin , HIGH); // CS High Level

  return (T_high_byte<<8 | T_low_byte); // 상위, 하위 바이트 연산
}

void loop() {
  while(1) {

    iTARGET = SEND_COMMAND(TARGET_CMD); // 대상 온도 Read
    delay(50); // 50ms : 이 라인을 지우지 마세요

    iSENSOR = SEND_COMMAND(SENSOR_CMD); // 센서 온도 Read
    delay(500); // 500ms : 이 라인을 지우지 마세요.
    Serial.print("Target Temp : ");
```

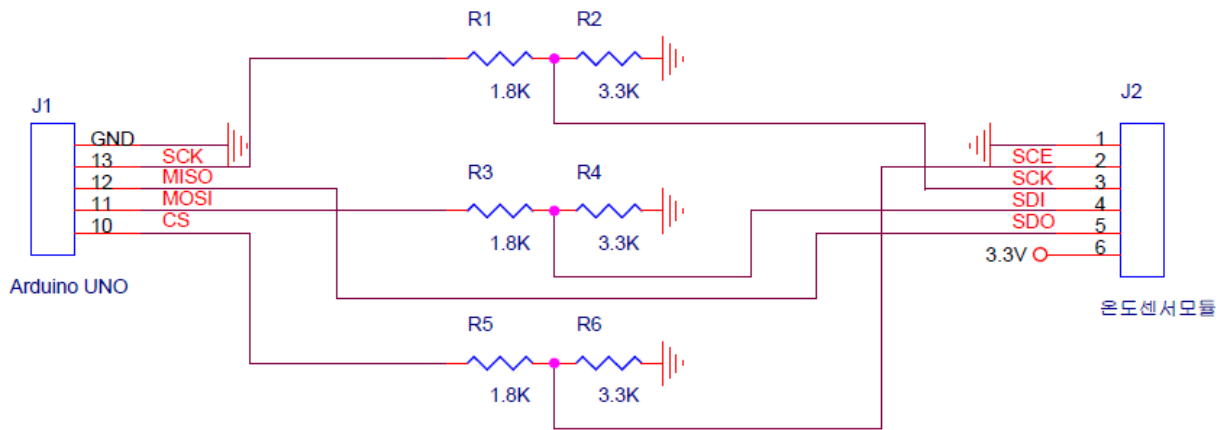
DTS-M300 Specification

```

Serial.print(float(iTARGET)/100);
Serial.print("    Sensor Temp : ");
Serial.println(float(iSENSOR)/100);

}
}
    
```

아두이노 UNO 와 DTS-M300 연결 회로도.



▶ DTS-M300 Revision History

Version	Date	Description
1.0	2013-10-02	First version is released.
1.5	2016-08-31	입력 전압 3.3V 고정 Arduino UNO 코드/ 회로도 추가 DW-Testboard 내용 추가