

DTS Series Specification



< DTS-L300-V2 >



<DTS-SIL300-B>

- 비접촉 온도 측정
- IR refresh rate : 2Hz
- High Accuracy
- Digital Resolution : 0.01°C
- Digital Interface : SPI
- 아두이노, 라즈베리파이 예제 코드 제공

▶ 특징

- 대상 온도와 센서 온도를 동시 측정
- 복잡한 온도 계산 수식이 필요 없습니다.
- SPI 디지털 통신으로 온도 데이터가 출력됩니다.

▶ 응용분야

- 과열방지 시스템
- 산업용 온도 측정 장치
- 체온 측정을 통한 인체 감지
- 가전기기
- 지능형 온도 제어 시스템

▶ Operational Characteristics

- if not otherwise noted, 25°C ambient temperature, 3.3V supply voltage and object with $\epsilon = 0.97$ were applied

Parameter		min	Typ	Max	Unit
측정각도(FOV)	DTS-SIL300-B		7.16		°
	DTS-L300-V2		3.814		
온도 측정 범위		-30		300	°C
측정 파장 대역		5.5		14	μm
동작온도(주변온도)		-20		70	°C
온도측정 시간			0.5		sec
정확도			± 2		%
Digital Resolution			0.01		°C
Standard Start-UP Time			3		sec
Stabilization Time			1		min

▶ 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 Electrical Requirements 를 지켜주세요.

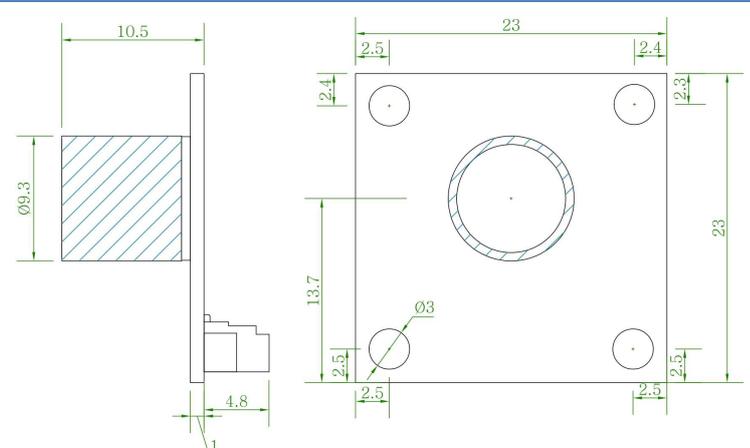
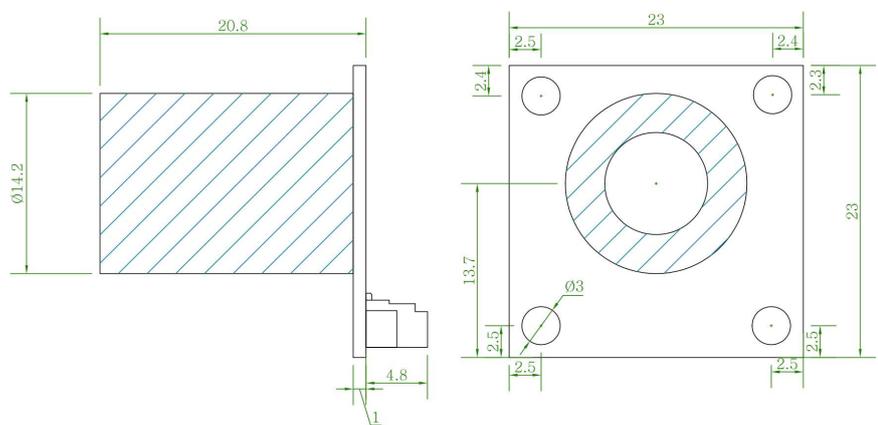
DTS Series Specification

▶ 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			0.97		
소비 전류		Full ambient temp. range, Typical value, no output load		10.96	12	mA
SPI	Clock		0.1		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

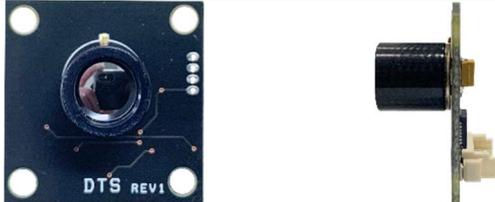
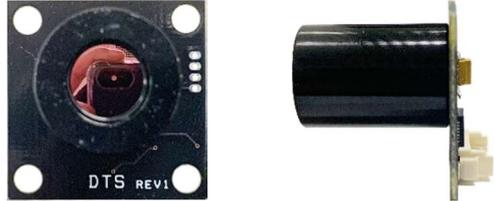
▶ Mechanical Dimensions

unit : mm

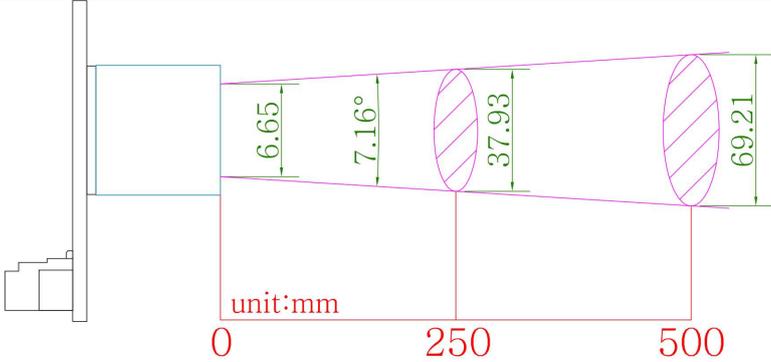
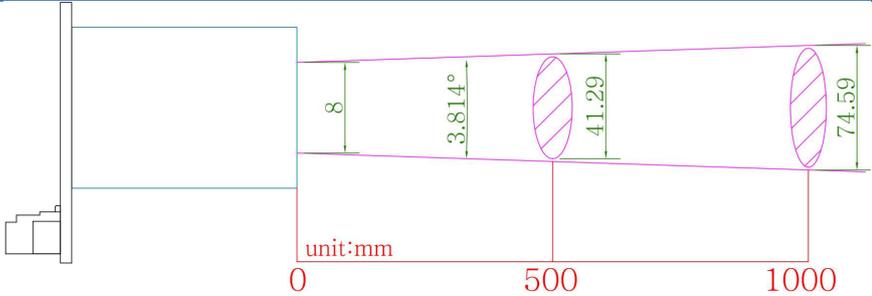
모델명	도면
DTS-SIL300-B	
DTS-L300-V2	

DTS Series Specification

▶ 제품 사진

모델명	사진
DTS-SIL300-B	
DTS-L300-V2	

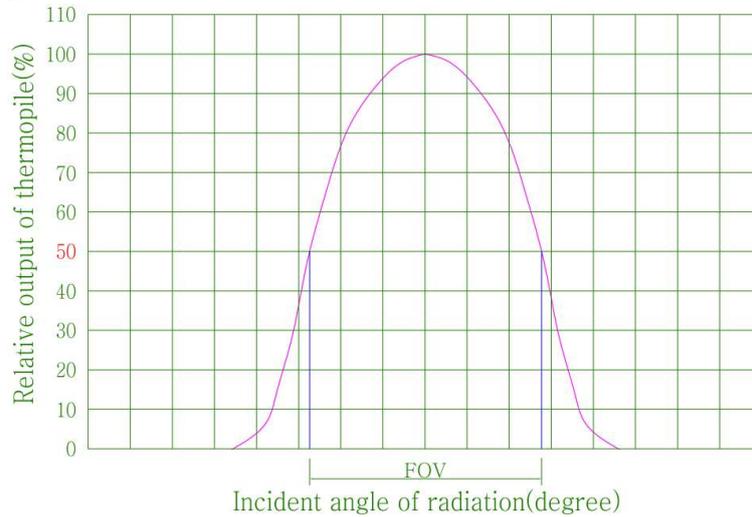
▶ Optical field of view (FOV)

모델명	측정각도	FOV
DTS-SIL300-B	7.16°	 <p>unit:mm</p> <p>0 250 500</p> <p>※ 측정직경 계산식 : $2 \times \tan(7.16^\circ / 2) \times \text{거리(mm)} + 6.65 \text{ (mm)}$</p>
DTS-L300-V2	3.814°	 <p>unit:mm</p> <p>0 500 1000</p> <p>※ 측정직경 계산식 : $2 \times \tan(3.814^\circ / 2) \times \text{거리(mm)} + 8 \text{ (mm)}$</p>

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

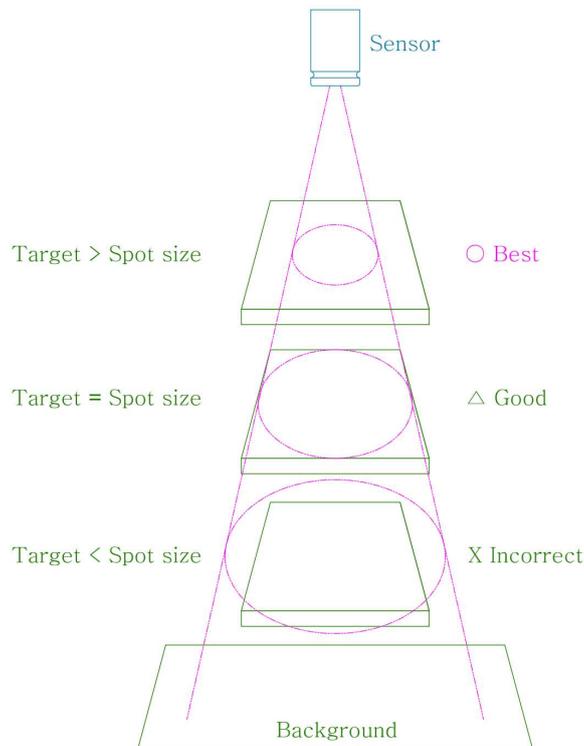
DTS Series Specification

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



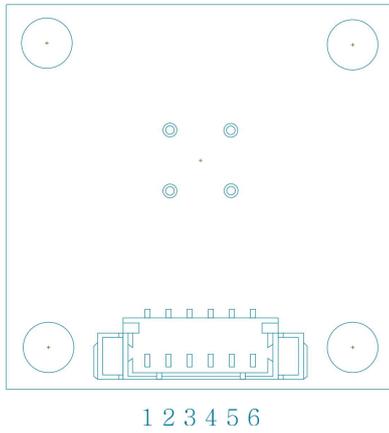
▶ DISTANCE AND SPOT SIZE

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



DTS Series Specification

▶ Pin Assignment



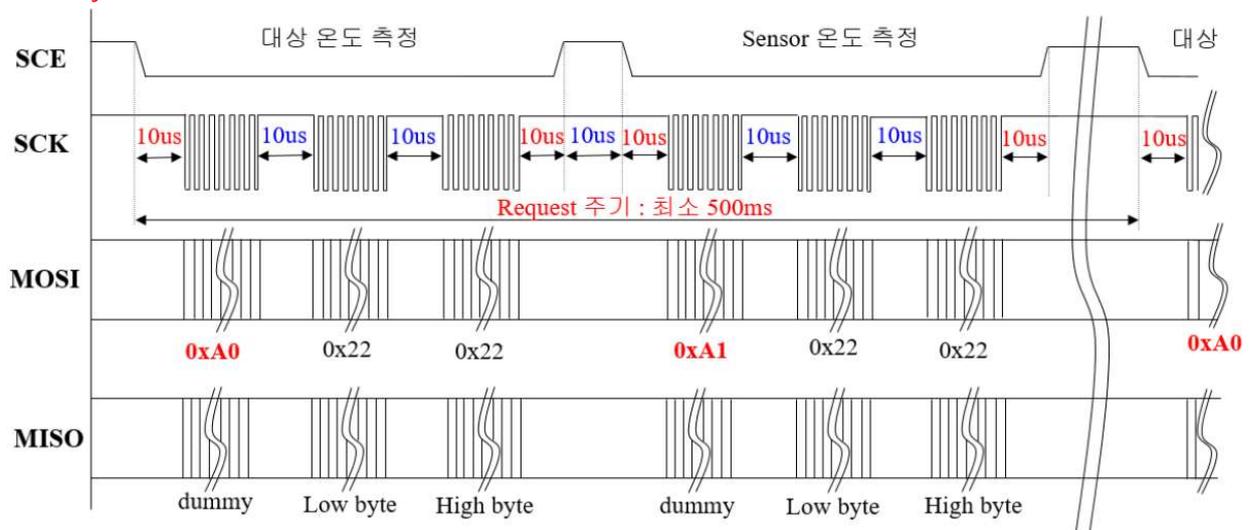
Pin Number	통신 방식
	SPI
1	GND
2	SCE
3	SCK
4	MOSI(SDI)
5	MISO(SDO)
6	3.3V

- Connector : Molex 053398-0671(반대편 Connector : Molex 51021-0600)

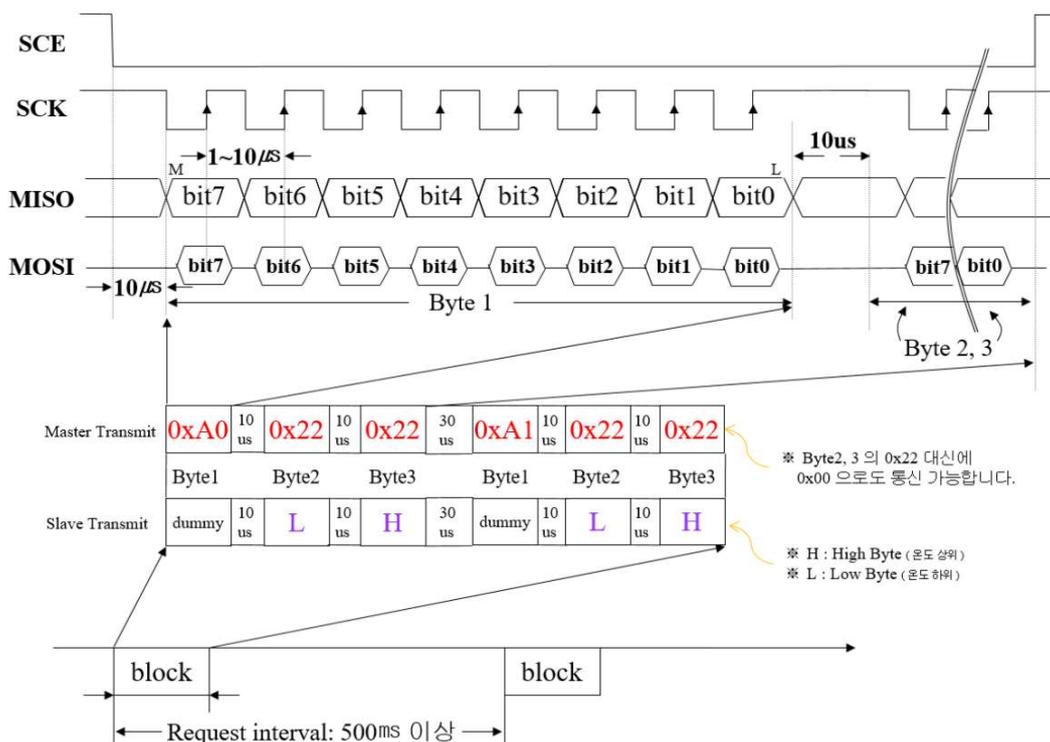
DTS Series Specification

SPI Communication and Timings

- SPI Data Mode : **Mode3** (SCK rising edge data sampling, SCK idle High)
- SPI Clock : Max **1MHz**
- SPI bit order : **MSB First**
- **Object(대상) & Sensor(센서) 온도 읽기 통신 파형**



- 1byte 확대 그림



DTS Series Specification

▶ 온도 계산 방법

- 영상온도 계산

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

* **object 온도 계산** : 상위 Byte(0x0E) + 하위 Byte(0x42) = 0x0E42 (Hex)
=> 3650 (dec) 이며 이 값을 100으로 나누면 36.50 도입니다.

* **sensor 온도 계산** : 상위 Byte(0x09) + 하위 Byte(0xC4) = 0x09C4 (Hex)
=>2500 (dec) 즉, 25.00 도입니다.

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

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

* **object 온도 계산** : 상위 Byte(0xFF) + 하위 Byte(0x6A) = 0xFF6A
0xFF6A의 2의 보수 = 0x0096 = 150 즉, -1.50도 입니다.

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

▶ Master 컨트롤러에 SPI 포트가 한 개인 경우 주의 사항.

SPI 포트는 하나 이면서 플래시 다운로드 포트 역시 SPI포트를 이용하는 경우(예, Atmega8 시리즈) 특히 주의 해야 합니다. 프로그램을 다운로드 할 경우, 반드시 센서의 연결을 해제 한 후 다운로드 하시기 바랍니다. 그렇지 않을 경우 의도치 않게 제품의 온도 보정수치가 변경되어 측정치에 오류가 생길 수 있습니다.

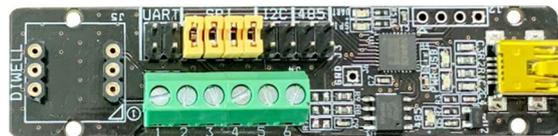
DTS Series Specification

▶ 윈도우 실행 프로그램 제공

온도 센서는 사용환경에서 측정해 보는 것이 가장 중요합니다.
 하지만 응용분야에 적용 가능 여부를 판단하기 위해 개발에 소요되는 시간/ 비용은 무시 못합니다.
 이런 경우 DW-LINK 통신 보드를 구매하시면 간단히 PC와 연결하여 온도 측정 및 기록이 가능합니다.
 DW-LINK 관련 자세한 사항은 별도의 문서로 제공됩니다. Windows 10 전용입니다.
 쇼핑몰 상세 페이지에서 다운받으세요.



< PC 화면 >



< DW-LINK >

DTS Series Specification

▶ Appendix - A (Example Code - Arduino UNO)

하단 코드는 아두이노 UNO 코드 예제 입니다. 예제 코드는 쇼핑몰에서 다운로드 가능합니다

```
/*  
*  
* Copyright (C) 2016 Diwell Electronics Co.,Ltd.  
* Project Name : (DTS 시리즈) SPI Master Code  
* Version : 1.2  
* SYSTEM CLOCK : 16Mhz  
* BOARD : Arduino UNO. 5V operation  
* 지원 모델명 : DTS-SIL300-B, DTS-L300-V2  
  
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.  
2. 2020.1.2 : 데이터 시트 통신 지연시간 변경에 따른 코드 수정.  
*/  
  
#include<SPI.h>  
  
#define OBJECT 0xA0 // 대상 온도 커맨드  
#define SENSOR 0xA1 // 센서 온도 커맨드  
  
const int chipSelectPin = 10;  
unsigned char Timer1_Flag = 0;;  
int iOBJECT, iSENSOR; // 부호 2byte 온도 저장 변수
```

DTS Series Specification

```
void setup() {
  /* Setting SCE & SPI */
  digitalWrite(chipSelectPin , HIGH);      // SCE High Level
  pinMode(chipSelectPin , OUTPUT);        // SCE OUTPUT Mode
  SPI.setDataMode(SPI_MODE3);            // SPI Mode
  SPI.setClockDivider(SPI_CLOCK_DIV16);   // 16MHz/16 = 1MHz
  SPI.setBitOrder(MSBFIRST);             // MSB First
  SPI.begin();                            // Initialize SPI

  delay(500);                             // Sensor initialization time
  Timer1_Init();                          // Timer1 setup : 500ms(2Hz) interval
  Serial.begin(9600);
  interrupts();                           // enable all interrupts
}

int SPI_COMMAND (unsigned char cCMD){
  unsigned char T_high_byte, T_low_byte;

  digitalWrite(chipSelectPin , LOW);      // SCE Low Level
  delayMicroseconds(10);                 // delay(10us)
  SPI.transfer(cCMD);                    // transfer 1st Byte
  delayMicroseconds(10);                 // delay(10us)
  T_low_byte = SPI.transfer(0x22);        // transfer 2nd Byte
  delayMicroseconds(10);                 // delay(10us)
  T_high_byte = SPI.transfer(0x22);       // transfer 3rd Byte
  delayMicroseconds(10);                 // delay(10us)
  digitalWrite(chipSelectPin , HIGH);    // SCE High Level

  return (T_high_byte << 8 | T_low_byte); // 온도값 return
}

ISR(TIMER1_OVF_vect) {                   // interrupt service routine (Timer1 overflow)
  TCNT1 = 34286;                        // preload timer : 이 값을 바꾸지 마세요.
  Timer1_Flag = 1;                      // Timer 1 Set Flag
}
```

DTS Series Specification

```

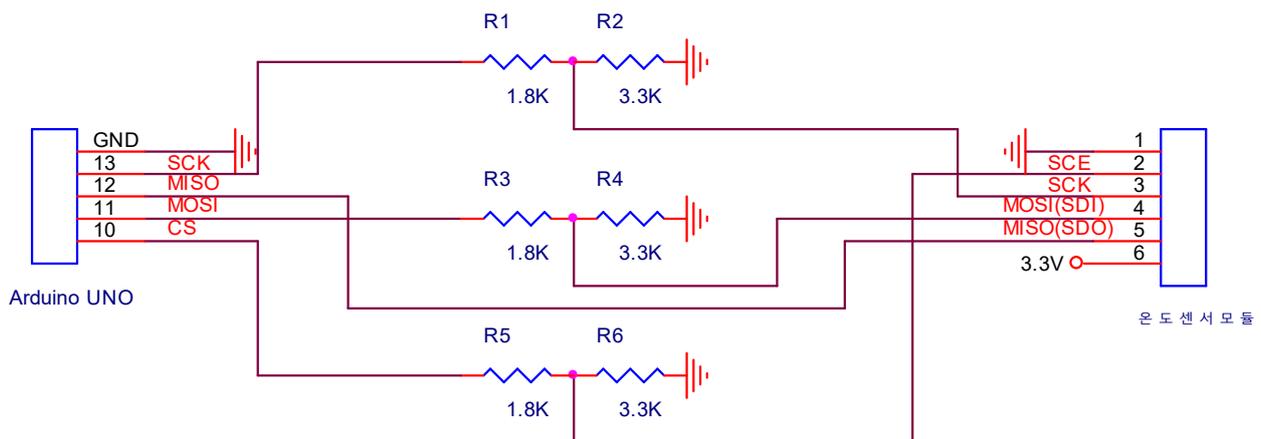
void Timer1_Init(void){
    TCCR1A = 0;
    TCCR1B = 0;
    TCNT1 = 34286;                // preload timer 65536-16MHz/256/2Hz
    TCCR1B |= (1 << CS12);        // 256 prescaler
    TIMSK1 |= (1 << TOIE1);       // enable timer overflow interrupt
}

void loop() {
    if(Timer1_Flag){              // 500ms 마다 반복 실행(Timer 1 Flag check)
        iOBJECT= SPI_COMMAND(OBJECT); // 대상 온도 Read
        delayMicroseconds(10);       // delay(10us)
        iSENSOR = SPI_COMMAND(SENSOR); // 센서 온도 Read

        Serial.print("Object Temp : "); // 하이퍼터미널 출력
        Serial.print(float(iOBJECT)/100, 2);
        Serial.print("   Sensor Temp : ");
        Serial.println(float(iSENSOR)/100, 2);
    }
}
    
```

※ 아두이노 UNO 와 DTS series 연결 회로도.

MISO 포트를 제외한 나머지 포트는 5V 로 동작하는 아두이노와 연결시 반드시 아래 회로도 대로 연결해야 합니다. 아두이노가 아닌 3.3V 로 동작하는 그 외 MCU 와 연결할 경우는 저항 없이 바로 연결하면 됩니다.



DTS Series Specification

▶ Appendix - B (Example Code - 라즈베리파이)

하단 코드는 라즈베리파이 코드 예제 입니다. 예제 코드는 쇼핑몰에서 다운로드 가능합니다.

```
/*
 *   description : DTS series Example Code.
 *   Version : 1.0
 *   Board : Raspberry 2 Model B V1.1
 *   Support Module : DTS-SIL300-B, DTS-L300-V2
 *   Revision history.
 *   1.0 2020.01.02 : First version is released.
 */

#include <stdint.h>
#include <stdio.h>
#include <string.h>
#include "wiringPi.h"
#include "wiringPiSPI.h"

#define LED           21           // LED PORT NUM 21
#define SCE           22           // SCE PORT NUM 22
#define spi_chn0      0           // SPI Channel 0
#define SPEED_1MHz    1000000    // SPI Speed : 1MHz
#define SPI_MODE3     3           // SPI MODE
#define OBJECT        0xA0        // COMMAND(Read Object Temp.)
#define SENSOR        0xA1        // COMMAND(Read Sensor Temp.)

int16_t iSensor, iObject;

int16_t SPI_COMMAND(uint8_t ADR){
    uint8_t Data_Buf[3];
    Data_Buf[0] = ADR;
    Data_Buf[1] = 0x22;
    Data_Buf[2] = 0x22;
}
```

DTS Series Specification

```
digitalWrite(SCE, 0); // SCE LOW
delayMicroseconds(10); // delay 10us

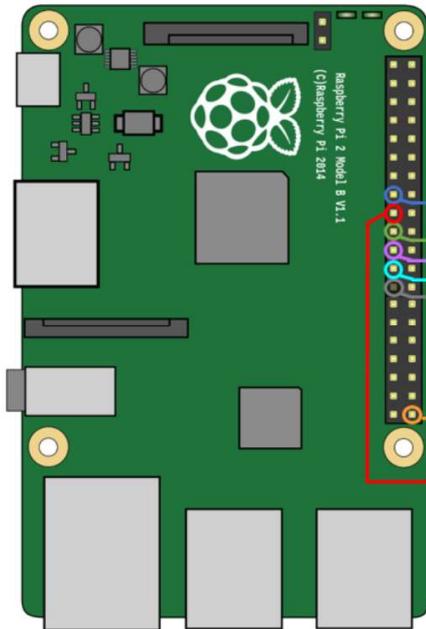
wiringPiSPIDataRW (spi_chn0, Data_Buf, 1); // transfer 1st byte.
delayMicroseconds(10); // delay 10us
wiringPiSPIDataRW (spi_chn0, Data_Buf+1, 1); // transfer 2nd byte
delayMicroseconds(10); // delay 10us
wiringPiSPIDataRW (spi_chn0, Data_Buf+2, 1); // transfer 3rd byte
delayMicroseconds(10); // delay 10us

digitalWrite(SCE, 1); // SCE HIGH
return (Data_Buf[2]*256+Data_Buf[1]); // High + Lo byte
}

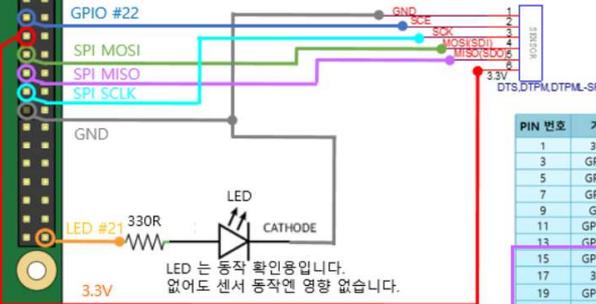
int main(void){
    wiringPiSetup(); // Wiring Pi setup
    if(wiringPiSetupGpio() == -1)
        return 1;
    pinMode(LED, OUTPUT); // LED Port Output (not necessary)
    pinMode(SCE, OUTPUT); // SCE Port Output
    digitalWrite(SCE,1); // SCE high

    wiringPiSPISetupMode(spi_chn0, SPEED_1MHz, SPI_MODE3); //SPI0, 1MHz, SPI Mode3 Setting
    delay(500); // wait 500ms
    while(1){
        iSensor = SPI_COMMAND(SENSOR); // Read Sensor temp.
        digitalWrite(LED, 1); // LED ON(not necessary)
        delayMicroseconds(10); // delay 10us
        iObject = SPI_COMMAND(OBJECT); // Read Object temp.
        digitalWrite(LED,0); // LED OFF(not necessary)
        delay(500); // Wait 500ms
        printf("Sensor : %5.1f , Object : %5.1f Wn", (double)iSensor/10, (double)iObject/10);
    }
    return 0;
}
```

DTS Series Specification



※ 라즈베리파이는 버전별로 GPIO 포트 핀배치가 다를 수 있습니다. 동작 테스트는 라즈베리파이2 Model B V1.1 를 사용하였습니다. 따라서 반드시 본인의 파이 보드 버전을 확인한 후에, 포트 위치가 맞는지 정확히 확인 후 연결하십시오.



PIN 번호	기능	비고	PIN 번호	기능	비고
1	3.3V		2	5V	
3	GPIO2	SDA1	4	5V	
5	GPIO3	SCL1	6	GND	
7	GPIO4	GPIO_GCLK	8	GPIO14	TXD0
9	GND	RXD0	10	GPIO15	
11	GPIO17	GPIO_GEN0	12	GPIO18	GPIO_GEN1
13	GPIO27	GPIO_GEN2	14	GND	
15	GPIO22	GPIO_GEN3	16	GPIO23	GPIO_GEN4
17	3.3V		18	GPIO24	GPIO_GEN5
19	GPIO10	SPI_MOSI	20	GND	
21	GPIO9	SPI_MISO	22	GPIO25	GPIO_GEN6
23	GPIO11	SPI_SCLK	24	GPIO8	SPI_CE0_N
25	GND		26	GPIO7	SPI_CE1_N
27	ID_SD		28	ID_SC	
29	GPIO5		30	GND	
31	GPIO6		32	GPIO12	
33	GPIO13		34	GND	
35	GPIO19		36	GPIO16	
37	GPIO26		38	GPIO20	
39	GND		40	GPIO21	

DTS Series Specification

▶ Additional Information

- manufacturer : Diwell Electronics Co., Ltd. <(주)디웰전자>
- Homepage : www.diwell.com
- shopping mall : www.diwellshop.com
- Phone : +82-70-8235-0820
- Fax : +82-31-429-0821
- Technical support : expoeb2@diwell.com, dsjeong@diwell.com
- 본 문서의 내용은 사전 통보 없이 변경될 수 있습니다.
- 쇼핑몰 내 제품 상세 페이지에서 최신 데이터시트가 제공됩니다.

▶ Revision History

Version	Date	Description
1.0	2013-10-02	First version is released.
1.1	2013-12-10	Update Physical Interface Parameters.(page 2)
1.2	2014-03-17	Sleep mode 삭제, 방사율 수정
1.3	2016-08-31	입력 전압 3.3V 고정 소비 전류 오류 수정, 제품 사진 update Arduino UNO 코드/ 회로도 추가 DW-Testboard 내용 추가
2.0	2020-01-02	DW-Testboard 단종. DW-LINK 내용 추가 표기 오류 정정 SPI 통신 스펙 변경(byte간 지연시간 축소 10ms→ 10us) DTS series 내용 통합.