

デジタル温度センサ ADT7310 DIP 化モジュール [MDK001]

特徴

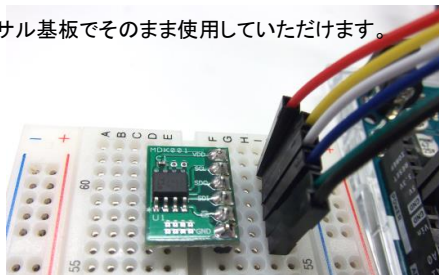
- Analog Devices 社製 **高性能温度センサ** 使用
- 広い温度測定範囲 (**-55°C ~ +150°C**)
- 高精度な温度測定 (**±0.5°C**)
- Arduino やマイコン等と相性のよい **SPI インターフェース**
- **2.54 ピッチ** で、ブレッドボードでそのまま使えます



販売元 **marutsu**
<http://www.marutsu.co.jp>

本モジュールは、Analog Devices 社の高精度温度センサ ADT7310 を搭載した、デジタル温度センサモジュールです。

1.27mm ピッチの IC を専用基板上で 2.54mm ピッチに変換していますので、ブレッドボードやユニバーサル基板上でそのまま使用いただけます。



●回路図

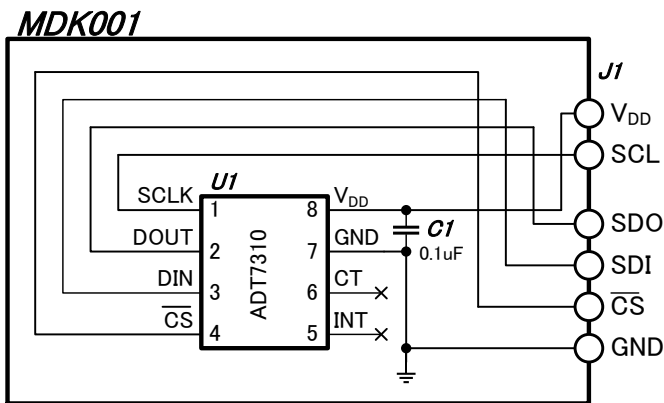


Fig.1 回路図

●部品表

List.1 部品表

No.	Reference	Part number	Characteristic	Description	Qty.
1	U1	ADT7310	SOIC-R8	Digital SPI Temperature Sensor	1
2	C1	-	1005 (0402")	0.1μF/16V	1
3	J1	-	2.54mm pitch	6-pin (2.54mm) pitch header	1

●ADT7310 の主な仕様

- | | |
|---|--|
| <ul style="list-style-type: none"> - 温度精度
±0.5°C @-40°C~+150°C (2.7~3.6V) ±0.4°C @-40°C~+105°C (3.0V) - 温度分解能 0.0078°C (16bit 設定時) 0.0625°C (13bit 設定時) - SPI 互換インターフェース | <ul style="list-style-type: none"> - 動作範囲
温度範囲 : -55°C~+150°C 電圧範囲 : 2.7V~5.5V - プログラマブル割り込み
温度加熱割り込み、温度上昇/低下割り込み - 温度校正/補正、直線性補正 不要 |
|---|--|

ADT7310 の詳細の仕様は、Analog Devices 社ウェブサイトのデータシートをご覧ください。

●使用例

ArduinoUNO R3 と接続して

温度を取得する例

MDK001 で取得した AD 変換結果を Arduino で温度に換算し、シリアルで出力します。

List.3
MDK001 と Arduino の接続

MDK001	接続	Arduino UNO R3
VDD	⇔	5V
SCL	⇔	D13 (SPI:SCK)
SDO	⇔	D12 (SPI:MISO)
SDI	⇔	D11 (SPI:MOSI)
CSx	⇔	D9
GND	⇔	GND

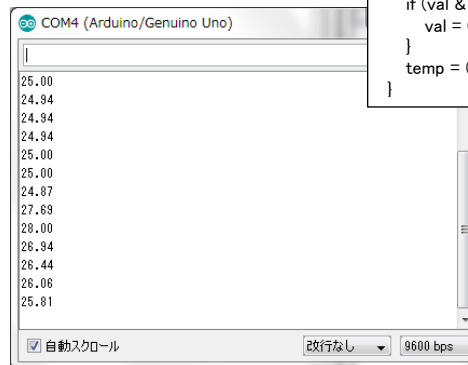


Fig.2 シリアル出力の例

List.2 サンプルプログラム

```
#include <SPI.h>
int CS_x = 9; // Pin9:CSx 信号とする

void setup() {
  pinMode(CS_x, OUTPUT); // CS_x を OUTPUT に設定
  digitalWrite(CS_x, HIGH); // CS_x をディセーブル

  SPI.begin(); // SPI の初期化
  SPI.setDataMode(SPI_MODE3);
  SPI.setBitOrder(MSBFIRST);
  SPI.setClockDivider(SPI_CLOCK_DIV8);

  Serial.begin(9600); // モニタ用シリアル通信の初期化

  digitalWrite(CS_x, LOW); // CS_x をイネーブル
  SPI.transfer(0x54); // CONTINUOUS READ MODE に設定
  delay(240); // 240ms 待つ
}

void loop() {
  uint16_t val; // AD 変換結果
  float temp; // 温度測定結果

  val = (uint16_t)SPI.transfer(0) << 8; // AD 結果上位 8bit 読み出しとシフト
  val = val | SPI.transfer(0); // AD 結果下位 8bit 読み出し
  val = val >> 3; // AD 変換結果の下位 3bit を捨てる

  if (val & 0x1000) { // AD 結果が負の数の場合の処理
    val = (uint32_t)val - 8192;
  }

  temp = (float)val / 16.0; // AD 結果を温度に変換する
}
```

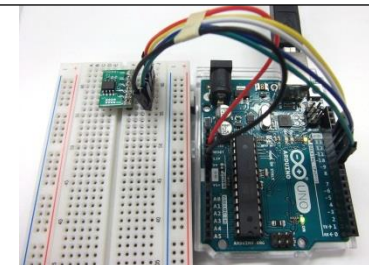


Fig.3 接続の例