TMP100 Temperature Sensor (SKU:TOY0045)

Contents

Introduction

This temperature sensor used TMP100 chip is ideal for extended temperature measurement in a variety of communication,consumer, environmental,and instrumentation applications,etc...
You can change the address or working voltage of the device depend on the environment.5V or 3.3V regular voltage input change by one tiny switch.I2C digital output communication ,easy to get temperature datas of different address,the resolution can be changed also.
This sensor isable to be powered by 5V or 3.3V which make it compatile with both Gadgeteer, Arduino DUE 3.3V system and standard Arduino UNO/Mega 5V system.

Applications

- Power-supply Temperature Monitoring
- Computer Peripheral Thermal Protection
- Thermostat Controls project,etc...

Specification

- Input Voltage:2.7-5.5V
- Working current:45uA(0.1uA standby)
- Resolution:9-12 Bits
- Address:B1001000-B1001111
- Opterating frequence:100KHZ/400KHZ/3.4MHZ
- Accuracy:±2℃@-25℃~85℃,±3℃@-55℃~125℃
- Interface:I2C Digital Output x1
  .NET gadgeteer connector (IDC10)Socket x2
- Size:32 x 27mm

Diagram
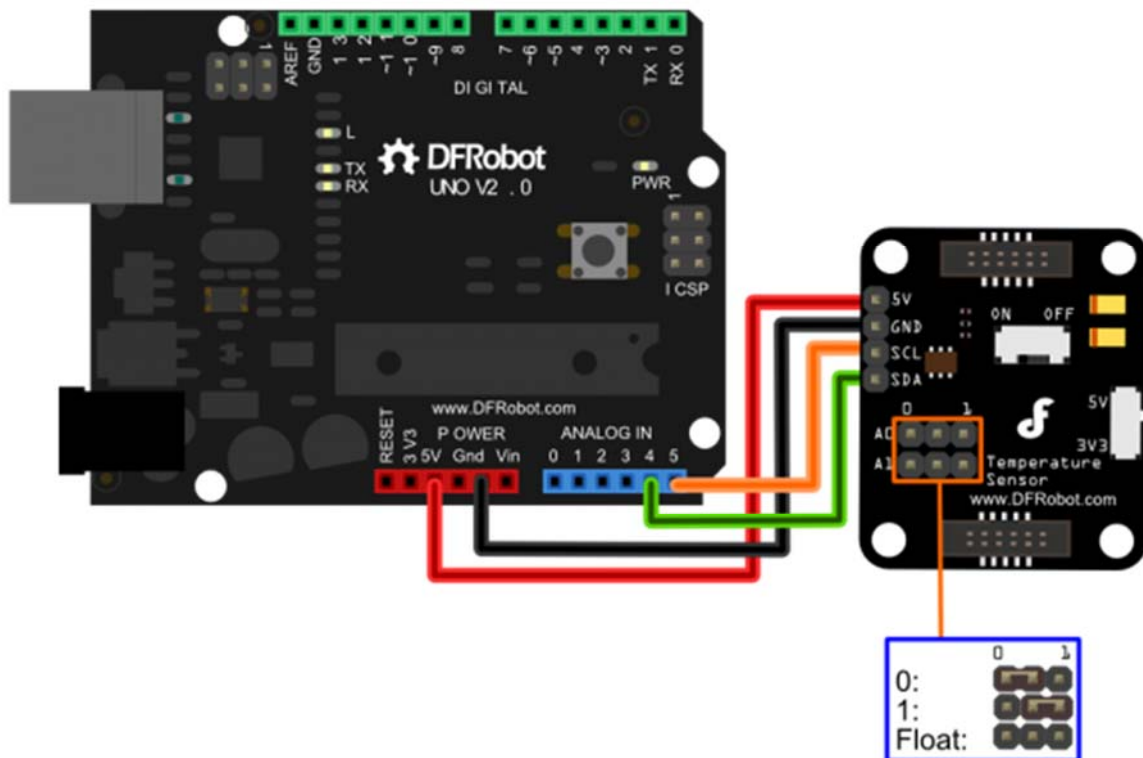
TOY0045 diagram

More details

- **ON/OFF Switch:**ON for Enable the module & OFF for Disable the module

- **5V/3V3 Switch:**5V or 3V3 power selection,make it compatile with both Gadgeteer, Arduino DUE 3.3V system and standard Arduino UNO/Mega 5V system.

TMP100 Rsgister Introduction

Address Pins and Slave Addresses

Steps:

1. Change the address in the code(it should be matched with the board)
2. Change the short current cap on the board according to the table
3. Power the whole system again once change the address

ADD1	ADD0	SLAVE ADDRESS
0	0	1001000
0	Float	1001001
0	1	1001010
1	0	1001100
1	Float	1001101
1	1	1001110
Float	0	1001011
Float	1	1001111

Address Pins and Slave Addresses

Pointer Register introduction

Pointer Register Type

P7	P6	P5	P4	P3	P2	P1	P0
0	0	0	0	0	0	Register Bits	

Pointer Addresses of the TMP100 Registers

P1	P0	REGISTER
0	0	Temperature Register(Read Only)
0	1	Configuration Register(Read/Write)
1	0	T_{LOW} Register(Read/Write)
1	1	T_{HIGH} Register(Read/Write)

Pointer Register Type

Resolution of the TMP100

R1	R0	RESOLUTION	CONVERSION TIME (typical)
0	0	9 Bits(0.5 °C)	40ms
0	1	10 Bits(0.25°C)	80ms
1	0	11 Bits(0.125°C)	160ms
1	1	12 Bits(0.0625°C)	320ms

Resolution of the TMP100

Configuration Register Format

D7	D6	D5	D4	D3	D2	D1	D0
OS/ALERT	R1	R0	F1	F0	POL	TM	SD

Configuration Register Format

For the TMP100,Shutdown Mode is enabled when the SD bit is 1. The device will shutdown once the current conversion is completed. For SD equal to 0, the device will maintain continuous conversion.

The power-up/reset value of the Configuration Register is all bits equal to 0. The OS/ALERT bit will read as 1 after power-up/reset.

Byte of Temperature Register

Byte1-High

D7	D6	D5	D4	D3	D2	D1	D0
T11	T10	T9	T8	T7	T6	T5	T4

Byte2-Low

D7	D6	D5	D4	D3	D2	D1	D0
T3	T2	T1	T0	0	0	0	0

Byte of Temperature Register

Sample Code

```
/*
 Sample code for the TMP100 Temperature sensor
 website:www.DFRobot.com

 Connection:

 VCC-5V
 GND-GND
 SDA-Analog pin 4
 SCL-Analog pin 5
*/

#include <Wire.h>
int tmpAddress = B1001011; //Slave Addresses set
int ResolutionBits = 10;    //Resolution set
void setup()
{
  Wire.begin();        // join i2c bus (address optional for master)
```

```
  Serial.begin(9600);  // start serial for output

  SetResolution();

}


void loop()

{

  getTemperature();

  delay(200);

}


float getTemperature()

{

  Wire.requestFrom(tmpAddress,2);

  byte MSB = Wire.read();

  byte LSB = Wire.read();

  int TemperatureSum = ((MSB << 8) | LSB) >> 4;

  float celsius = TemperatureSum*0.0625;

  Serial.print("Celsius: ");

  Serial.println(celsius);

}


void SetResolution(){

  if (ResolutionBits < 9 || ResolutionBits > 12) exit;

  Wire.beginTransmission(tmpAddress);

  Wire.write(B00000001); //addresses the configuration register

  Wire.write((ResolutionBits-9) << 5); //writes the resolution bits

  Wire.endTransmission();

  Wire.beginTransmission(tmpAddress); //resets to reading the temperature

  Wire.write((byte)0x00);
```

```
  Wire.endTransmission();
```