DS18B20

 Sensori digitali della Dallas Semiconductors; forniscono temperature in gradi Celsius tramite un segnale digitale in uscita a 9-12 bit; esistono in due versioni differenti: sonde per le misure in acqua e sonde per la misura fuori dall’acqua. Naturalmente, quelle in acqua possono funzionare anche fuori dall’acqua. Le caratteristiche principali sono le seguenti:

* Tensione di alimentazione da 3.0V a 5.5V
* ±0.5°C di precisione da -10°C a +85°C
* Range di temperatura misurabile: -55 a 125°C (-67°F a +257°F)
* da 9 a 12 bit di risoluzione
* Un unico cavo di connessione digitale
* Tempo di risposta inferiore a 750ms

|  |  |
| --- | --- |
|  |  |



* Questo sensore è caratterizzato dal protocollo one wire che permette la trasmissione seriale di più dispositivi collegati su uno stesso cavo che comunica su un pin del microcontrollore. Il numero massimo di sensori collegati su un solo cavo è 100; la lunghezza massima del cavo è 300 m. I dati sono trasmessi solo quando il sensore viene interrogato in questo modo si evitano trasmissioni in contemporanea di più sonde che causerebbero errori di lettura. A tal fine, ogni dispositivo è munito di un indirizzo univoco memorizzato su una ROM interna ad esso.
* La memoria ROM di ciascun ds18b20 è di 64 bit di cui:
	+ I primi 8 bit sono per il codice CRC=X8+x5+x4+1;
	+ 48 bit intermedi per il serial number;
	+ Gli ultimi 8 bit per il family code;





Collegamento elettrico

**Struttura interna del ds18b20**



**Registri di temperatura**

Il ds18b20 è fornito di un registro di temperatura (come si vede dalla figura in alto) su una memoria non volatile, una EEPROM. La memoria fa da tampone e riporta l’ultima temperatura letta; ha una capacità di 2 byte così ripartiti:



I primi 11 bit esprimono il valore della temperatura (26 25 24 23 22 21 20 2-12-22-3 2-4 ); di questi valori, i 7 bit più significativi, danno valore intero fino ad un massimo di 125 °C fino ad un minimo di -55 °C, gli ultimi quattro bit meno significativi, danno un valore decimale. Gli altri bit sono il segno che è espresso con zero se positivo, con uno se il segno è negativo.

L’interrogazione del dispositivo avviene tramite il comando “Convert T” che corrisponde in esadecimale con 44h; la sonda risponde con 0 mentre esegue l’operazione e poi con 1 ad operazione terminata. Il valore della temperatura viene messo nella memoria EEPROM detta anche SCRATCHPAD; per leggere tale valore, si invia al dispositivo il comando di lettura corrispondente al codice esadecimale beh.

**Progetto Arduino con un solo sensore**

****

*#include <OneWire.h>*

*#include <DallasTemperature.h>*

*#define onewire 10*

*OneWire ourWire(onewire);*

*DallasTemperature sensors(&ourWire);*

*void setup()*

*{Serial.begin(9600);*

*sensors.begin();}*

*void loop() {Serial.println();*

*sensors.requestTemperatures();*

*float t=sensors.getTempCByIndex(0);*

*Serial.print("Temperatura = ");*

*Serial.print(t);*

*Serial.println(" gradi C");*

*}*

**Progetto Arduino con più sensori**

****

*#include <OneWire.h>*

*#include <DallasTemperature.h>*

*#define ONE\_WIRE\_BUS 2 //sul pin2*

*#define TEMPERATURE\_PRECISION 9*

*OneWire oneWire(ONE\_WIRE\_BUS);*

*DallasTemperature sensors(&oneWire);*

*DeviceAddress insideThermometer, outsideThermometer;*

*void setup(void)*

*{ Serial.begin(9600);*

 *Serial.println("Dallas Temperature IC Control Library Demo");*

 *sensors.begin();*

 *Serial.print("Locating devices...");*

 *Serial.print("Found ");*

 *Serial.print(sensors.getDeviceCount(), DEC);*

 *Serial.println(" devices.");*

 *Serial.print("Parasite power is: ");*

 *if (sensors.isParasitePowerMode()) Serial.println("ON");*

 *else Serial.println("OFF");*

 *if (!sensors.getAddress(insideThermometer, 0)) Serial.println("Unable to find address for Device 0");*

 *if (!sensors.getAddress(outsideThermometer, 1)) Serial.println("Unable to find address for Device 1");*

 *// show the addresses we found on the bus*

 *Serial.print("Device 0 Address: ");*

 *printAddress(insideThermometer);*

 *Serial.println();*

 *Serial.print("Device 1 Address: ");*

 *printAddress(outsideThermometer);*

 *Serial.println();*

 *// set the resolution to 9 bit*

 *sensors.setResolution(insideThermometer, TEMPERATURE\_PRECISION);*

 *sensors.setResolution(outsideThermometer, TEMPERATURE\_PRECISION);*

 *Serial.print("Device 0 Resolution: ");*

 *Serial.print(sensors.getResolution(insideThermometer), DEC); Serial.println();*

 *Serial.print("Device 1 Resolution: ");*

 *Serial.print(sensors.getResolution(outsideThermometer), DEC);*

 *Serial.println();*

*}*

*void printAddress(DeviceAddress deviceAddress)*

*{ for (uint8\_t i = 0; i < 8; i++)*

 *{*

 *if (deviceAddress[i] < 16) Serial.print("0");*

 *Serial.print(deviceAddress[i], HEX);*

 *}*

*}*

*void printTemperature(DeviceAddress deviceAddress)*

*{*

 *float tempC = sensors.getTempC(deviceAddress);*

 *Serial.print("Temp C: ");*

 *Serial.print(tempC);*

 *Serial.print(" Temp F: ");*

 *Serial.print(DallasTemperature::toFahrenheit(tempC));*

*}*

*void printResolution(DeviceAddress deviceAddress)*

*{*

 *Serial.print("Resolution: ");*

 *Serial.print(sensors.getResolution(deviceAddress));*

 *Serial.println();*

*}*

*void printData(DeviceAddress deviceAddress)*

*{*

 *Serial.print("Device Address: ");*

 *printAddress(deviceAddress);*

 *Serial.print(" ");*

 *printTemperature(deviceAddress);*

 *Serial.println();*

*}*

*void loop(void)*

*{*

 *Serial.print("Requesting temperatures...");*

 *sensors.requestTemperatures();*

 *Serial.println("DONE");*

 *printData(insideThermometer);*

 *printData(outsideThermometer);}*

**Programma per STM32**

#include "mbed.h"

#include "DS1820.h"

Serial pc(SERIAL\_TX, SERIAL\_RX);

const int MAX\_PROBES = 1;

DS1820 probe[1] = {A0};

int main()

{

 int i;

 int devices\_found=0;

 //pc.attach(&pcRx, pc.RxIrq);

 pc.printf("Started\r\n");

 pc.printf("search\_ROM\_setup\r\n");

 probe[0].search\_ROM\_setup();

 pc.printf("search\_ROM\r\n");

 while (probe[devices\_found].search\_ROM() and devices\_found<MAX\_PROBES-1)

 devices\_found++;

 // If maximum number of probes are found,

 // bump the counter to include the last array entry

 if (probe[devices\_found].ROM[0] != 0xFF)

 devices\_found++;

 pc.printf("devices found:%d\r\n", devices\_found);

 while(1) {

 probe[0].convert\_temperature(DS1820::all\_devices);

 for (i=0; i<devices\_found; i++) {

 // pc.printf("Device[%d]: %3.1f \r\n",i, probe[i].temperature('f'));

 float t=probe[i].temperature('f');

 float tc=(t-32)/1.8; //da Farenheit a Celsius

 pc.printf("Device[%d]: %3.1f \r\n",i, t);

 pc.printf("Device[%d]: %3.1f \r\n",i, tc);

 }

 wait(1); // 1 sec

 }}