**Termometro ONE WIRE DS18B20**

Il sensore DS18B20 è un sensore di temperatura digitale. È dotato di tre pin, due per l’alimentazione ed uno come sensore.

Caratteristiche del sensore DS18B20:

Tensione di alimentazione da 3.0V a 5.5V

±0.5°C 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

Ogni DS18B20 contiene una memoria ROM di 64 bit. I primi 8 bit sono per la famiglia one-wire; 48 bit sono il codice unico di ogni dispositivo; gli ultimi 8 sono per il codice CRC e sono gli ultimi di 56 bit del codice

CRC=X8+x5+x4+1





**Progetto con Arduino**

*#include <OneWire.h>*

*#include <DallasTemperature.h>*

*#define ONE\_WIRE\_BUS 14*

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

*DallasTemperature sensors(&ourWire);*

*void setup()*

*{*

*delay(1000);*

*Serial.begin(9600);*

*Serial.println("Misura la temperatura con sensore DS18B20");*

*delay(1000);*

*sensors.begin();*

*}*

*void loop()*

*{*

*Serial.println();*

*sensors.requestTemperatures();*

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

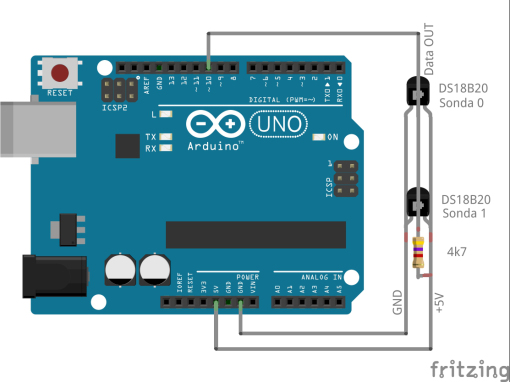
*Serial.print(sensors.getTempCByIndex(0));*

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

*}*

**Misure di temperatura tramite più sensori**

L’importanza di questo sensore è di permettere di determinare la temperatura di più zone attraverso più dispositivi collegati in parallelo. Questi sensori hanno la caratteristica di trasmettere la temperatura su un piedino in forma seriale e di poter essere collegati tutti su uno stesso cavo, “one wire” appunto. 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.

[](https://cfpmanfredini.files.wordpress.com/2014/04/ds18b20doppio.jpg)

Layout dei componenti, i dati viaggiano su un filo solo (ONE WIRE) e arrivano al piedino 10 di Arduino

Per determinare la quantità di sonde collegate e la temperatura di ognuna abbiamo utilizzato questo programma

*#include <OneWire.h>*

*#include <DallasTemperature.h>*

*#define ONE\_WIRE\_BUS 10*

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

*DallasTemperature sensors(&oneWire);*

*DeviceAddress tmp\_address;*

*int numberOfDevices;*

*void* ***setup****(void)*

*{*

***Serial****.begin(9600);*

***Serial****.println("Sto cercando i sensori...");*

*sensors.begin();*

*delay(6000);*

*numberOfDevices = sensors.getDeviceCount();*

***Serial****.print("Trovati ");*

***Serial****.print(numberOfDevices);*

***Serial****.println(" sensori!");*

***Serial****.println("Inizio la misurazione...");*

***Serial****.println();*

*}*

*void* ***loop****(void)*

*{*

*sensors.requestTemperatures(); // Comando //per misurare la temp.*

*for(int i=0;i<numberOfDevices; i++)*

*{*

***Serial****.print("Sensore ");*

***Serial****.print(i);*

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

***Serial****.print(sensors.getTempCByIndex(i));*

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

*delay(100);*

***Serial****.println();*

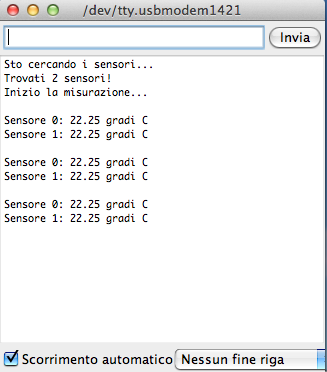
*}*

***Serial****.println();*

*delay(1000);*

*}*

L’output del programma è sul monitor seriale di Arduino e si presenta come nell’immagine sottostante. Sono indicati i sensori (in questo caso 2) e la relativa temperatura. Qui i 2 sensori misuravano lo stesso valore perché erano collegati molto vicini.

[](https://cfpmanfredini.files.wordpress.com/2014/04/schermata-2014-04-24-alle-16-06-32.png)

L’uscita sul monitor seriale

Il programma seguente è identico al precedente ma la visualizzazione viene fatta su un display a cristalli liquidi tipo Hitachi.

*// Misura la temperatura con 2 sensori DS18B20*

*// e la visulaizza su un display LCD*

*#include <OneWire.h>*

*#include <DallasTemperature.h>*

*#include <LiquidCrystal.h>*

*#define Backlight 10*

*LiquidCrystal lcd(8, 9, 4, 5, 6, 7);*

*#define ONE\_WIRE\_BUS 11*

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

*DallasTemperature sensors(&oneWire);*

*DeviceAddress tmp\_address;*

*int temp0=0;*

*int temp1=0;*

*int numberOfDevices;*

*void* ***setup****(void){*

*lcd.begin(16, 2);*

*numberOfDevices = 1;*

*sensors.begin();*

*delay(1000);*

*pinMode(Backlight, OUTPUT);*

*digitalWrite(Backlight, HIGH); // Accende retroilluminazione*

*}*

*void* ***loop****(void){*

*sensors.requestTemperatures(); // Comando per misurare la temp.*

*temp0 = sensors.getTempCByIndex(0);*

*lcd.setCursor(0,0); // Colonna 0, Riga 0*

*lcd.print(temp0);*

*lcd.print(char( 223)); // Scrive il pallino dei gradi*

*lcd.print("C");// Scrive la C di Celsius*

*temp1 = sensors.getTempCByIndex(1);*

*lcd.setCursor (0,1); //Colonna 0, Riga 1*

*lcd.print(temp1);*

*lcd.print(char( 223)); // Scrive il pallino dei gradi*

*lcd.print("C");// Scrive la C di Celsius*

*}*

Esistono due versioni del sensore DS18B20: uno per effettuare misure in aria, all’asciutto e un altro per effettuare misure in acqua. Quest’ultimo, può essere utilizzato anche per effettuare misure fuori dall’acqua.

L’immagine seguente mostra il sensore ad immersione:



Anche questo sensore è dotato di tre pin: due per l’alimentazione e uno per interfaccia su un microcontrollore. Lo schema circuitale è lo stesso per il sensore fuori dall’acqua e anche il programma di arduino è lo stesso. Se si vuole fare un datalogger con registrazione su scheda microsd, basta caricare il seguente programma sulla scheda di arduino:

*/\**

*\* analog sensors on analog ins 0, 1, and 2*

*\* SD card attached to SPI bus as follows:*

*\*\* MOSI - pin 11*

*\*\* MISO - pin 12*

*\*\* CLK - pin 13*

*\*\* CS - pin 4 (for MKRZero SD: SDCARD\_SS\_PIN)*

*\*/*

*#include <SPI.h>*

*#include <SD.h>*

*#include <OneWire.h>*

*#include <DallasTemperature.h>*

*#define ONE\_WIRE\_BUS 14 //A0*

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

*DallasTemperature sensors(&ourWire);*

*#define analogPin A0*

*const int chipSelect = 4;*

*void setup() {*

*Serial.begin(9600);*

*Serial.println("Misura la temperatura con sensore DS18B20");*

*Serial.println("Mariangela");*

*delay(1000);*

*sensors.begin();*

*while (!Serial) {*

*;*

*}*

*Serial.print("Initializing SD card...");*

*// vede se la card è inizializzata*

*if (!SD.begin(chipSelect)) {*

*Serial.println("Card failed, or not present");*

*// don't do anything more:*

*return;*

*}*

*Serial.println("card initialized.");*

*}*

*void loop() {*

*String dataString = "";*

*//inizializza una stringa*

*sensors.requestTemperatures();*

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

*Serial.print(sensors.getTempCByIndex(0));*

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

*dataString += String(sensors.getTempCByIndex(0));*

*dataString += ",";//scrive i dati del sensore in una stringa*

*File dataFile = SD.open("datalog.txt", FILE\_WRITE);*

*if (dataFile) {*

*dataFile.println(dataString);*

*dataFile.close();*

*// print to the serial port too:*

*Serial.println(dataString);*

*}*

*else {*

*Serial.println("error opening datalog.txt");}delay(3000);}*

Non è stato necessario riportare lo schema della scheda SD su arduino poiché tutte le informazioni sono scritte nel file e il numero di pin è riportato sulla scheda stessa. La comunicazione tra la scheda ed Adruino è di tipo SPI dove sono coinvolti i pin MISO, MOSI, clk e chipselect.