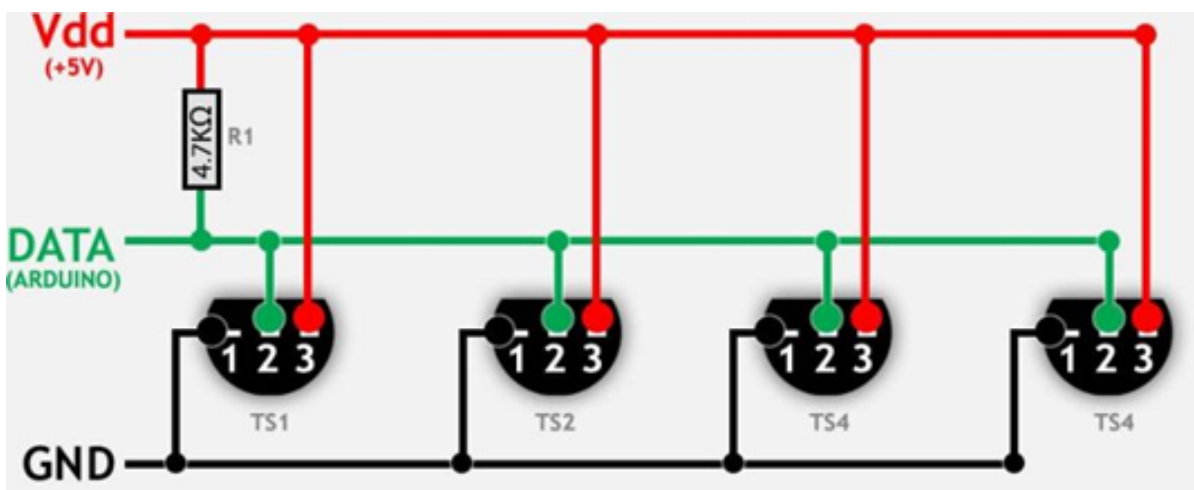
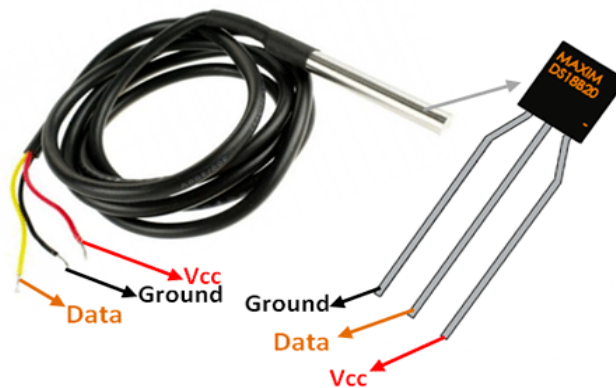


# TRASDUTTORE DI TEMPERATURA DIGITALE DALLAS/Maxim - DS18B20

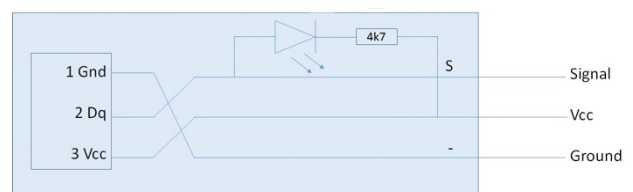
In questa prova, impareremo ad usare il trasduttore di temperatura DS18B20, questo trasduttore è di tipo digitale ed ha una versatilità di funzionamento superiore a quella del "cugino analogico" LM35D pur mantenendo la stessa estetica ovvero utilizza lo stesso contenitore (TO-92). Ogni singolo dispositivo ha un suo "codice identificativo" così è possibile collegarne diversi in parallelo utilizzando un solo pin per la lettura della temperatura alla quale ognuno di essi è sottoposto (one wire). Per renderlo operativo e compatibile con Arduino, dobbiamo utilizzare le seguenti librerie: DallasTemperature.h & OneWire.h. Lo sketch risulta molto semplice poiché utilizza le sopracitate librerie che si occupano di tutta la gestione del trasduttore.

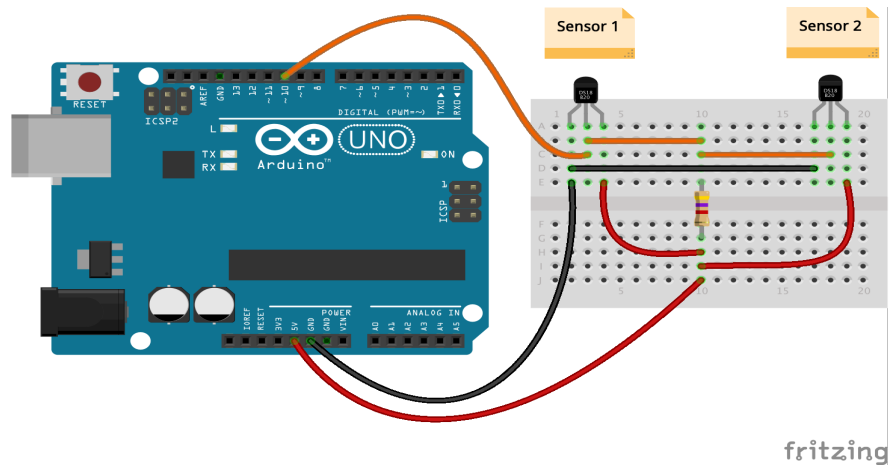


Per applicazioni professionali è disponibile la versione water proof.



Versione su pcb con led signal





### Codice di esempio per n°2 sensori su Serial Monitor

```

#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 10 //Impostiamo sul pin 10 l'ingresso dati dei sensori
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);
}

```

**Codice di esempio per l'utilizzo autonomo con display LCD 16X2**

```

/*
Il programma visualizza su di un display LCD 16X2
la temperatura misurata con il trasduttore DS18B20.

Vengono utilizzati i seguenti pin:
Pin +5V      -> Alimentazione VCC SENSORE E VDD LCD
Pin GND      -> Alimentazione GND SENSORE E VSS LCD
Non dimenticare la resistenza da 4K7 tra +5V e pin Data del DS18B20
Pin Digital 7 -> Linea dati sensore DS18B20
Pin Digital 5 -> Bus D4 dati LCD
Pin Digital 4 -> Bus D5 dati LCD
Pin Digital 3 -> Bus D6 dati LCD
Pin Digital 2 -> Bus D7 dati LCD
Pin Digital 11 -> Terminale EN display LCD
Pin Digital 12 -> Terminale RS display LCD
Pin GND      -> Terminale RW display LCD

Revisione del 02.03.2022
This example code is in the public domain.
*/

#include <OneWire.h>
#include <DallasTemperature.h>
#include <LiquidCrystal.h>

//Impostiamo il terminale Data del DS18B20 al Digital Pin 7 di Arduino
#define ONE_WIRE_BUS 7

// Impostiamo la comunicazione oneWire con un dispositivo compatibile
OneWire oneWire(ONE_WIRE_BUS);

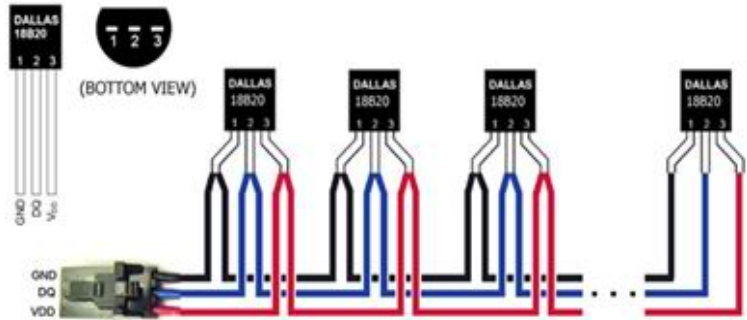
// Passaggio oneWire reference alla libreria Dallas Temperature.
DallasTemperature sensors(&oneWire);

// RS EN D4 D5 D6 D7 (RW to GND of Arduino)- pin collegamento LCD
LiquidCrystal lcd(2, 3, 4, 5, 11, 12);

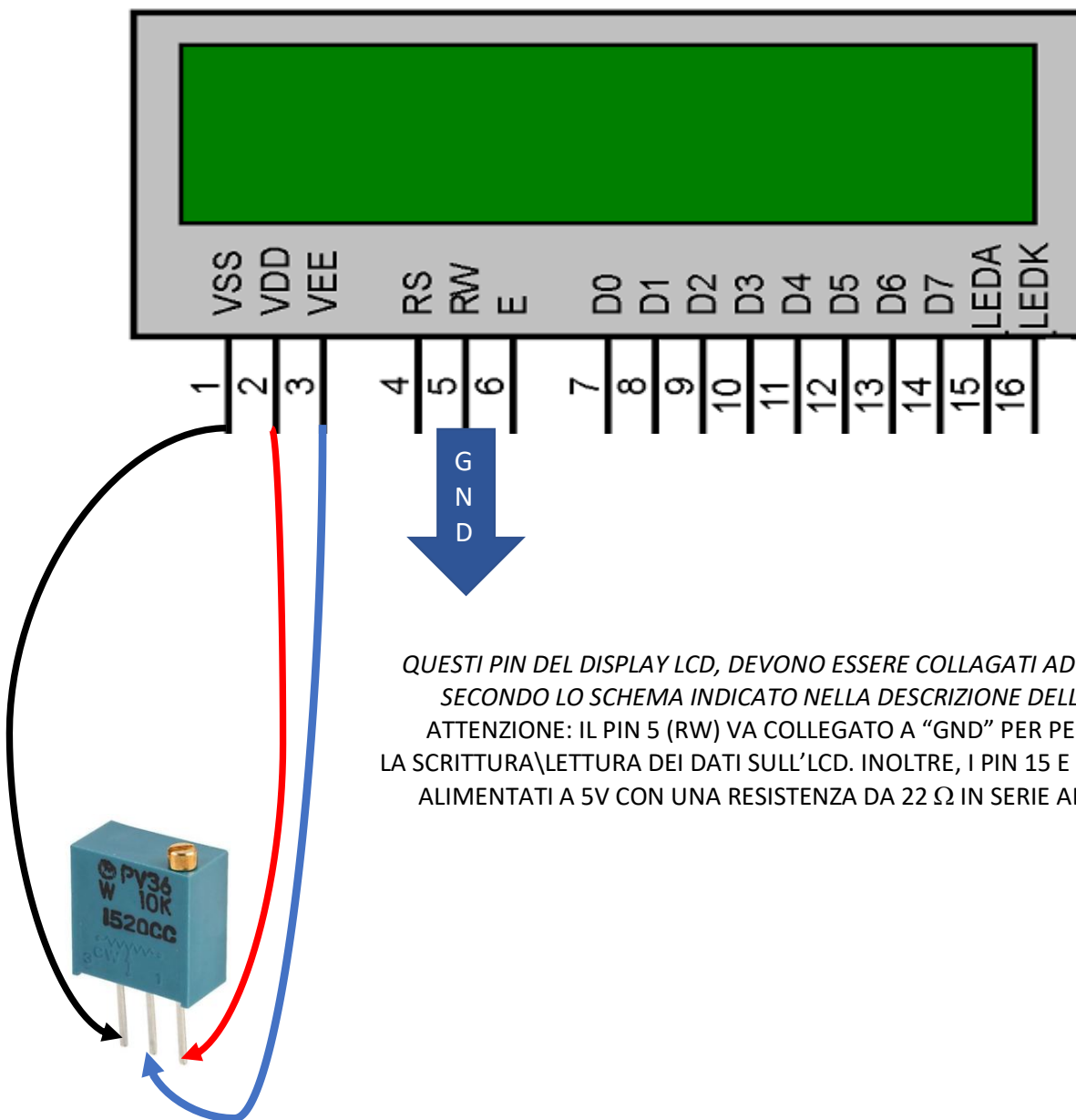
void setup(void)
{
  // Start up the library
  sensors.begin();
  // Impostiamo il valore di righe e colonne del display LCD
  lcd.begin(16, 2);
}

void loop(void)
{
  sensors.requestTemperatures(); //Invia il comando di lettura delle Temp.
  lcd.clear();
  lcd.setCursor(0, 0); //Abilitiamo la scrittura sulla prima riga dell'LCD
  lcd.print("Temperatura di: ");
  lcd.setCursor(0, 1); //Abilitiamo la scrittura sulla seconda riga dell'LCD
  lcd.print (sensors.getTempCByIndex(0)); //scriviamo sull'LCD i gradi
  lcd.print (" C");
}

```

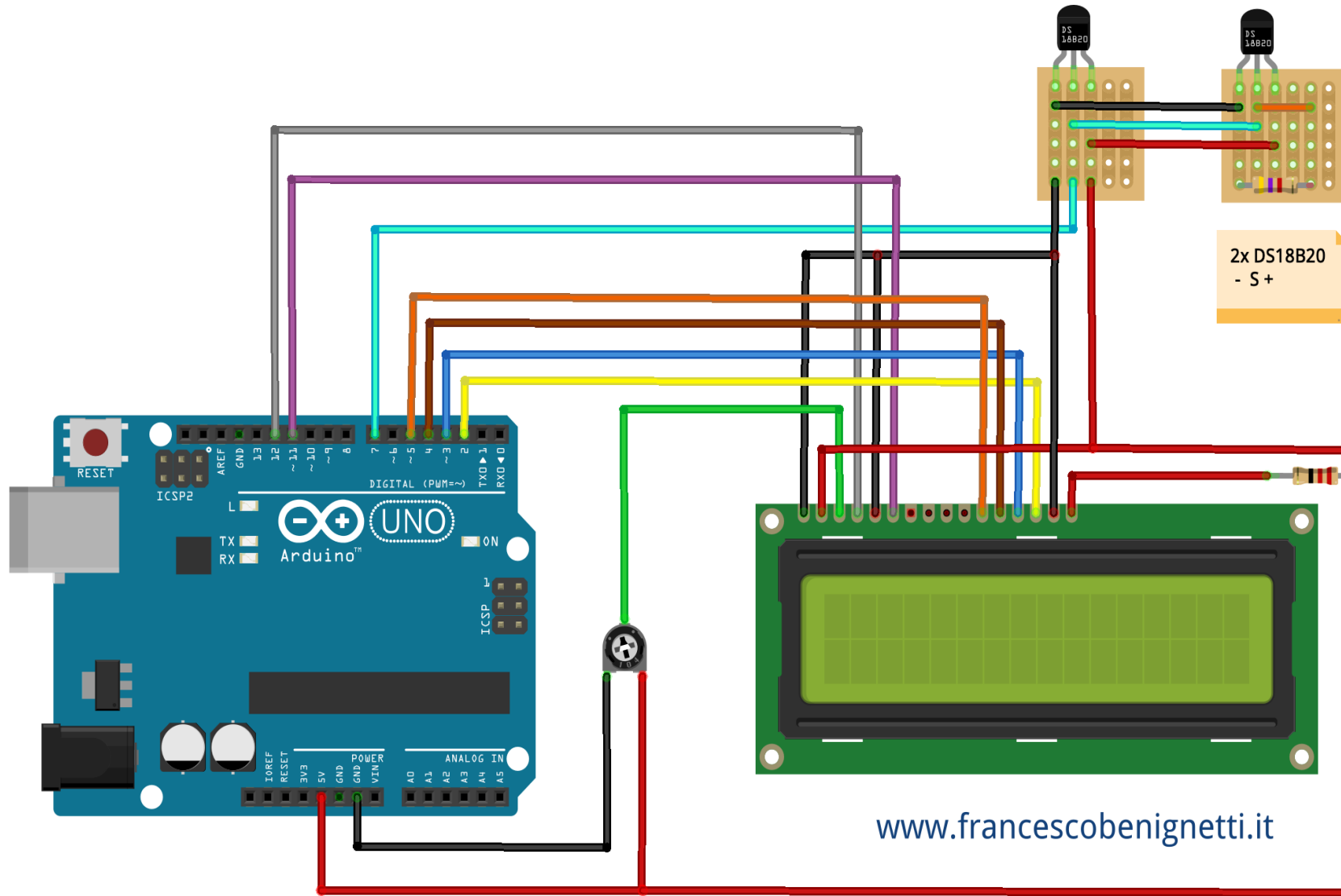


Connessioni Display LCD 16X2 - standard HD 44780



QUESTI PIN DEL DISPLAY LCD, DEVONO ESSERE COLLAGATI AD ARDUINO SECONDO LO SCHEMA INDICATO NELLA DESCRIZIONE DELLO SKETCH. ATTENZIONE: IL PIN 5 (RW) VA COLLEGATO A "GND" PER PERMETTERE LA SCRITTURA\LETTURA DEI DATI SULL'LCD. INOLTRE, I PIN 15 E 16 VANNO ALIMENTATI A 5V CON UNA RESISTENZA DA 22 Ω IN SERIE ALL'ANODO.

Schema di montaggio con LCD standard HD44780



[www.francescobenignetti.it](http://www.francescobenignetti.it)

fritzing

**Esempio Misura temperatura di due sensori DS18B20 + LCD 20x4 I2C**

```

/*
Il programma visualizza su di un display LCD 20X4
la temperatura misurata con due trasduttori DS18B20.
Vengono utilizzati i seguenti pin:
Pin +5V -> Alimentazione VCC SENSORE E VDD LCD
Pin GND -> Alimentazione GND SENSORE E VSS LCD
Pin A4 -> SDA
Pin A5 -> SCL
Pin 7 -> Ingresso Segnale One Wire dei DS18B20
Revisione del 03.03.2022 per LCD 20X4 @ Prof. Benignetti Francesco
This example code is in the public domain.
*/
#include <Wire.h> // libreria wire presente, di default, nell'IDE
#include <OneWire.h> // dichiarazione di utilizzo della libreria OneWire.h
#include <DallasTemperature.h> // dichiarazione di utilizzo della libreria DallasTemperature.h
#include <LiquidCrystal_I2C.h> // libreria di gestione dell'interfaccia I2C per display LCD
LiquidCrystal_I2C lcd(0x27, 20, 4); //Utilizziamo Display LCD 20X4 con una interfaccia I2C
#define ONE_WIRE_BUS 7 // utilizziamo il pin 7 e gli diamo il nome ONE_WIRE_BUS
OneWire oneWire(ONE_WIRE_BUS); // associa al Pin 7 il valore di ONE_WIRE_BUS ovvero
// il canale DATI dei sensori (Pin 7 Arduino)
DallasTemperature sensors(&oneWire); // trasferisce alle routine della libreria DallasTemperature
//il controllo delle funzioni OneWire

void setup(void)
{
Serial.begin(9600); // inizializza la porta seriale per l'utilizzo e il controllo delle temperature nel monitor seriale
lcd.begin(); //inizializzo il display LCD
sensors.begin(); // inizializza i sensori
}
void loop(void)
{
Serial.print("Richiedi temperatura...");

sensors.requestTemperatures(); // invia ai sensori la richiesta di misurare la temperatura
delay (1000); // attende un secondo per dare tempo ai sensori di effettuare la misurazione
Serial.println("fatto");

Serial.print("Temperatura 1 (indice 0) e': ");
Serial.print("Temperatura 2 (indice 1) e': ");

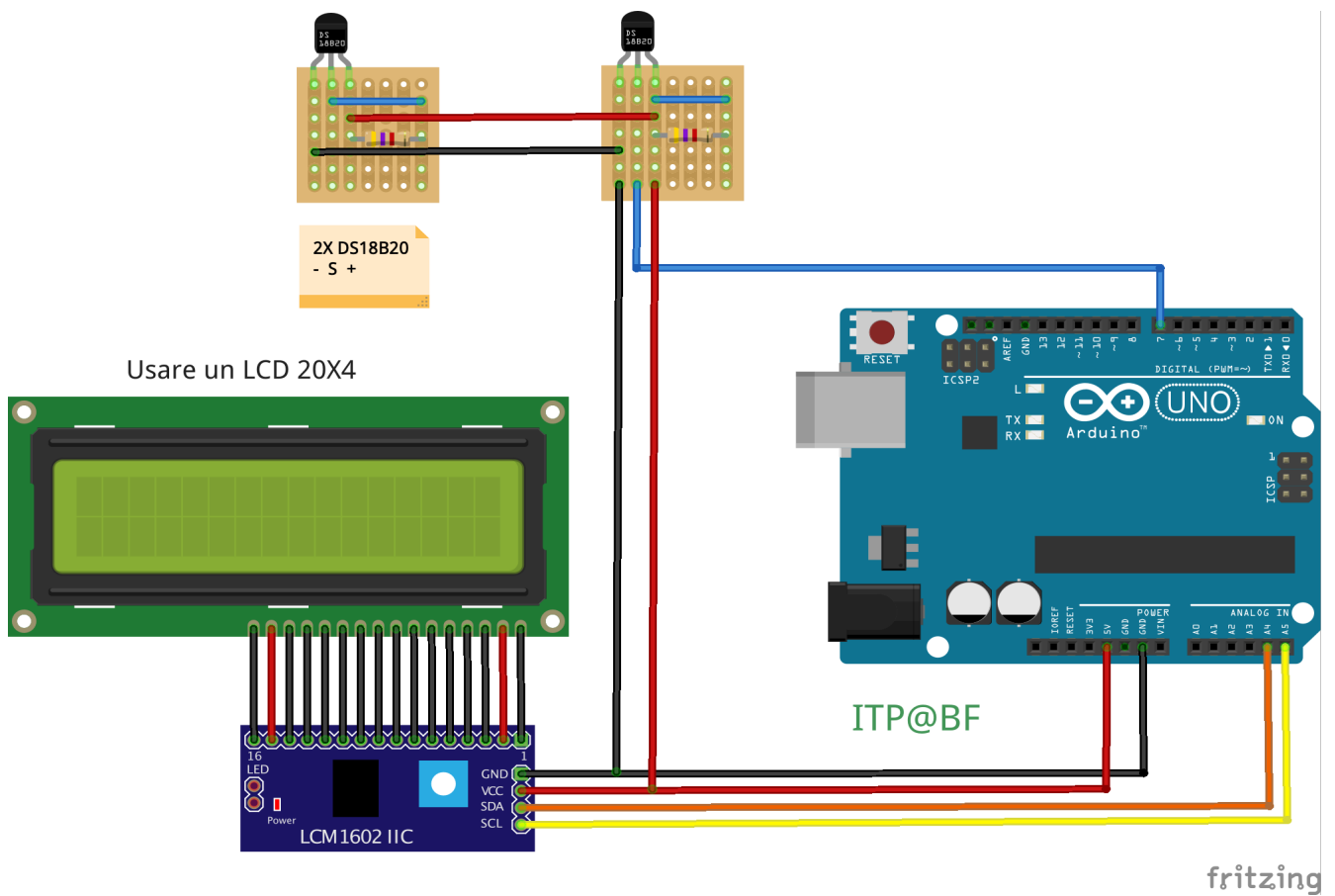
Serial.println(sensors.getTempCByIndex(0)); // invia la monitor seriale la temperatura rilevata
Serial.println(sensors.getTempCByIndex(1)); // invia la monitor seriale la temperatura rilevata
delay (2000); // attende 3 secondi prima di verificare nuovamente la temperatura

lcd.backlight(); // abilito la retroilluminazione del LCD
lcd.setCursor(0,0); // posiziona il cursore all'inizio della prima riga
lcd.print("Cerulli_Soap Machine"); //scrive sulla prima riga LCD
lcd.setCursor(0, 1); // posiziona il cursore all'inizio della seconda riga
lcd.print ("Soda_Temp: "); // scrive sulla seconda riga LCD Soda_Temp:
lcd.setCursor(12, 1); //stampo il valore della temperatura dalla posizione 12 della seconda riga
lcd.print (sensors.getTempCByIndex(0)); // espone il livello temp 1 sul display LCD

lcd.setCursor(0, 2); // posiziona il cursore all'inizio della terza riga
lcd.print ("Oil_Temp: ");
lcd.setCursor(12, 2); //stampo il valore della temperatura dalla posizione 12 della terza riga
lcd.print (sensors.getTempCByIndex(1)); // espone il livello temp 2 sul display LCD
lcd.setCursor(0,3); // posiziona il cursore all'inizio della quarta riga
lcd.print("Work in Progress!!!"); //Scrive sulla quarta riga dell'LCD
}

```

Schema di montaggio con LCD 20x4 equipaggiato con interfaccia I2C



Note: