

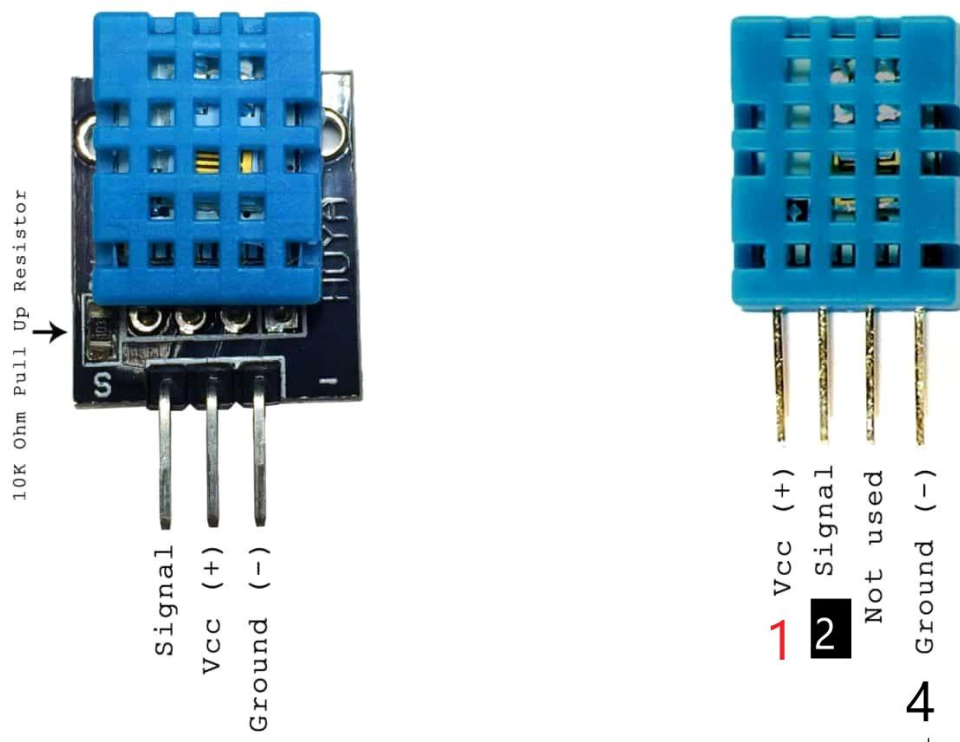
Note di rilascio della mini stazione meteo, versione 1.0

Il programma e' stato realizzato in linguaggio Python con l'ausilio di due librerie per controllare i due componenti.Nella cartella principale "source" si trova il codice sorgente "Meteo.py" e le due librerie citate nelle note di installazione.Il software e' stato collaudato su Raspberry Pi 4.

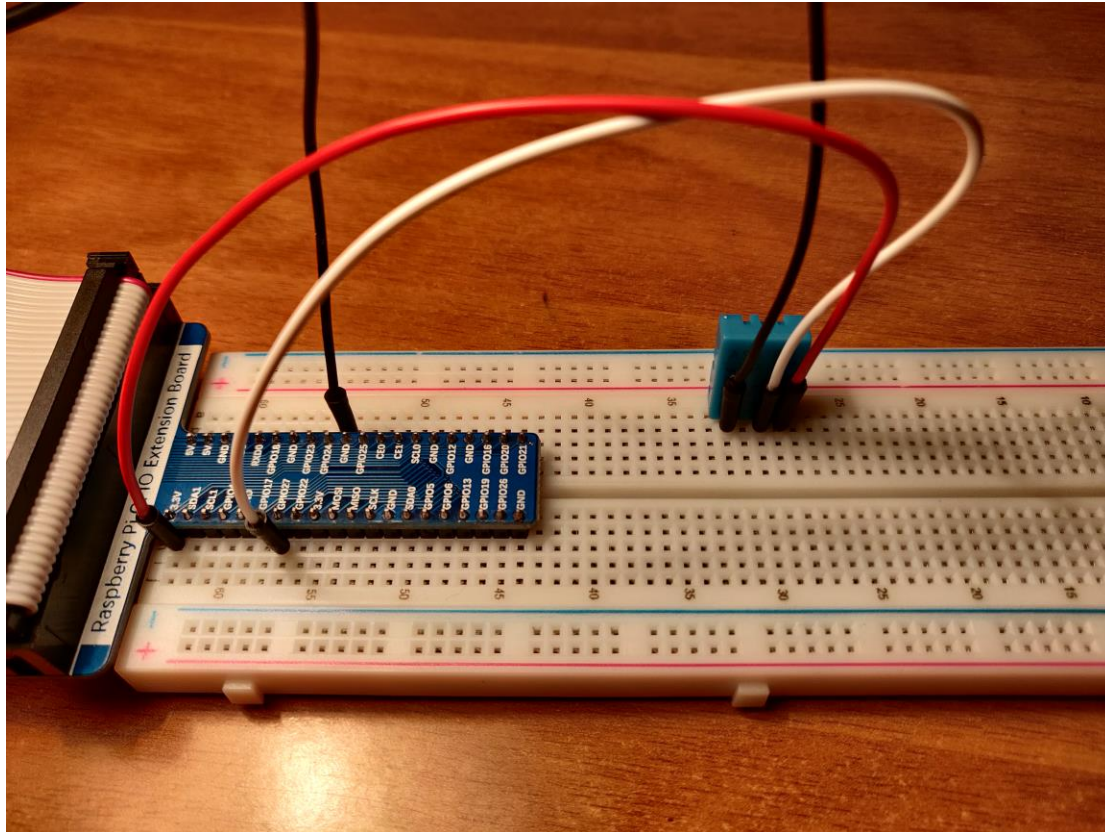
1.0)REALIZZAZIONE DEL DISPOSITIVO

1.1) COLLEGARE DHT-11 AL RASPBERRY PI

Esistono due varianti.Uno con 4 pin e l'altro con 3 pin.



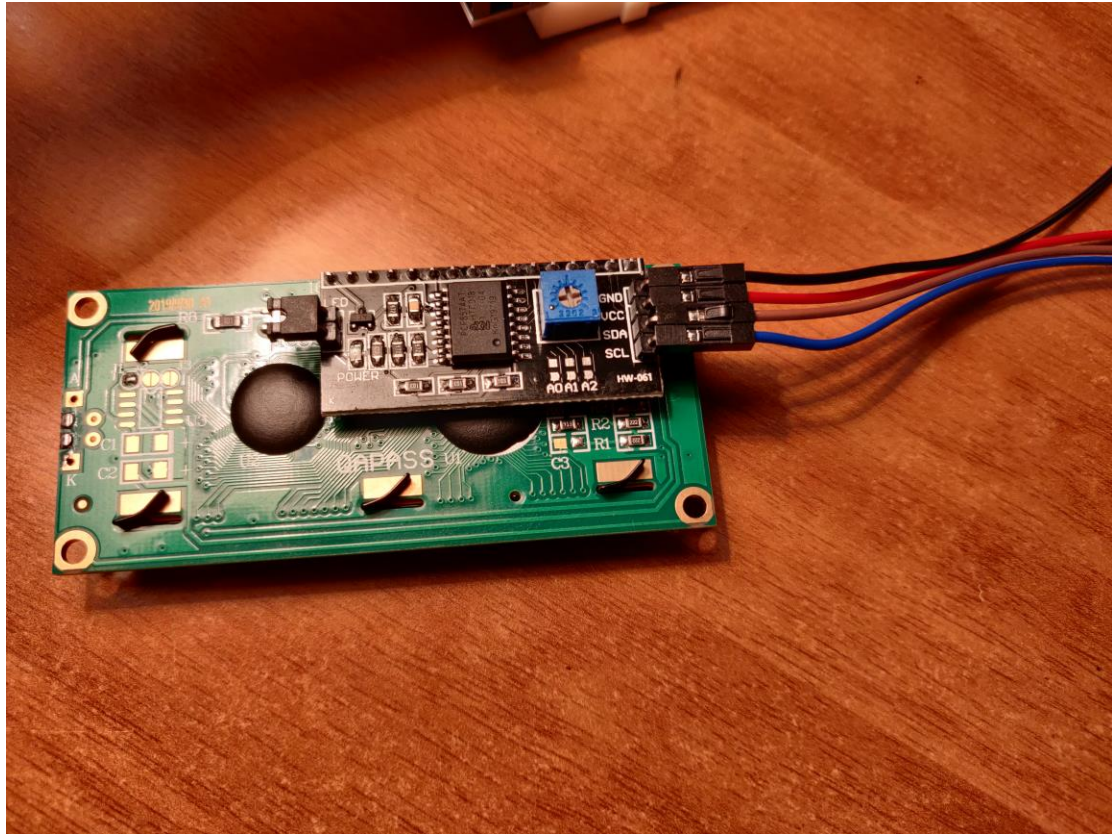
Se si dispone di un DHT11 a quattro pin e si desidera trasmettere umidità e temperatura al nostro Raspberry PI bisogna collegarlo in questo modo:



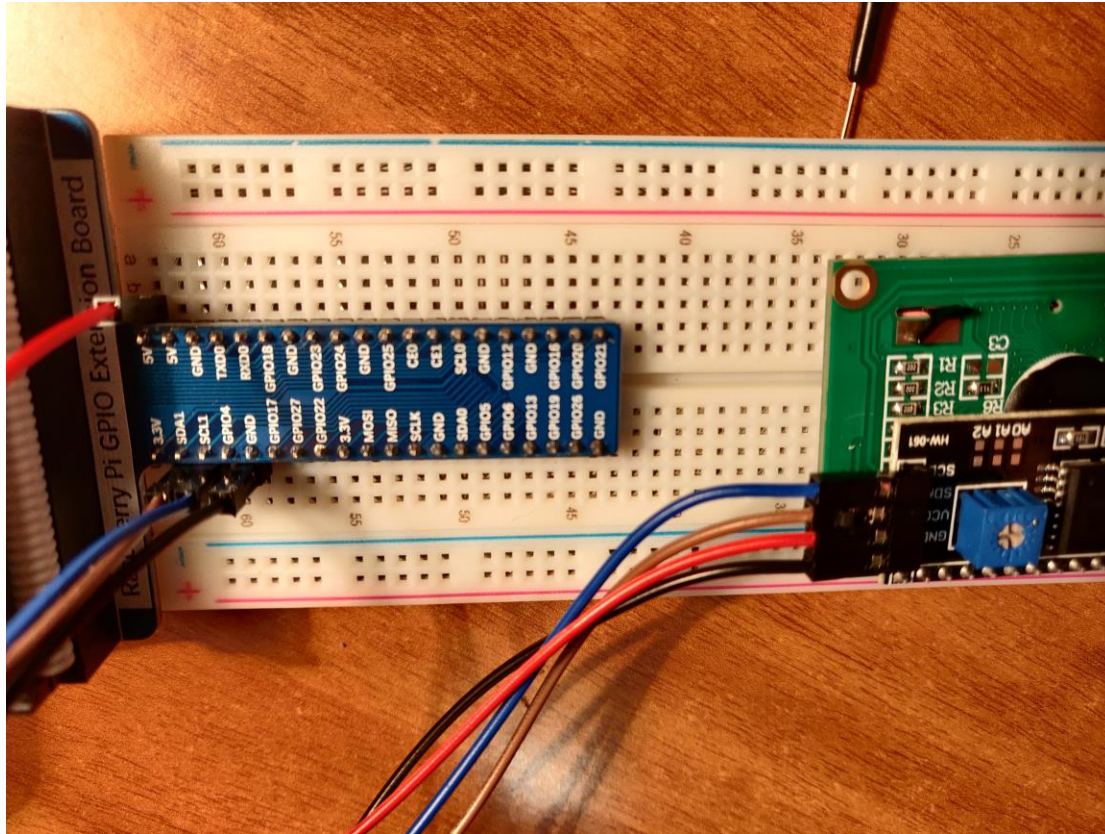
DHT-11	COLLEGAMENTO	RASBERRY
PIN 1(VCC)	Cavo rosso	3.3V
PIN 2(signal)	Cavo bianco	GPIO27
PIN 3	non usato	////////
PIN 4(GND)	Cavo nero	GND

1.2) COLLEGARE I2C LCD AL RASBERRY PI

I2C (circuito integrato) è anche noto come interfaccia a due fili poiché utilizza solo due fili per inviare e ricevere dati. In realtà ne servono quattro se si contano i cavi Vcc e di terra.

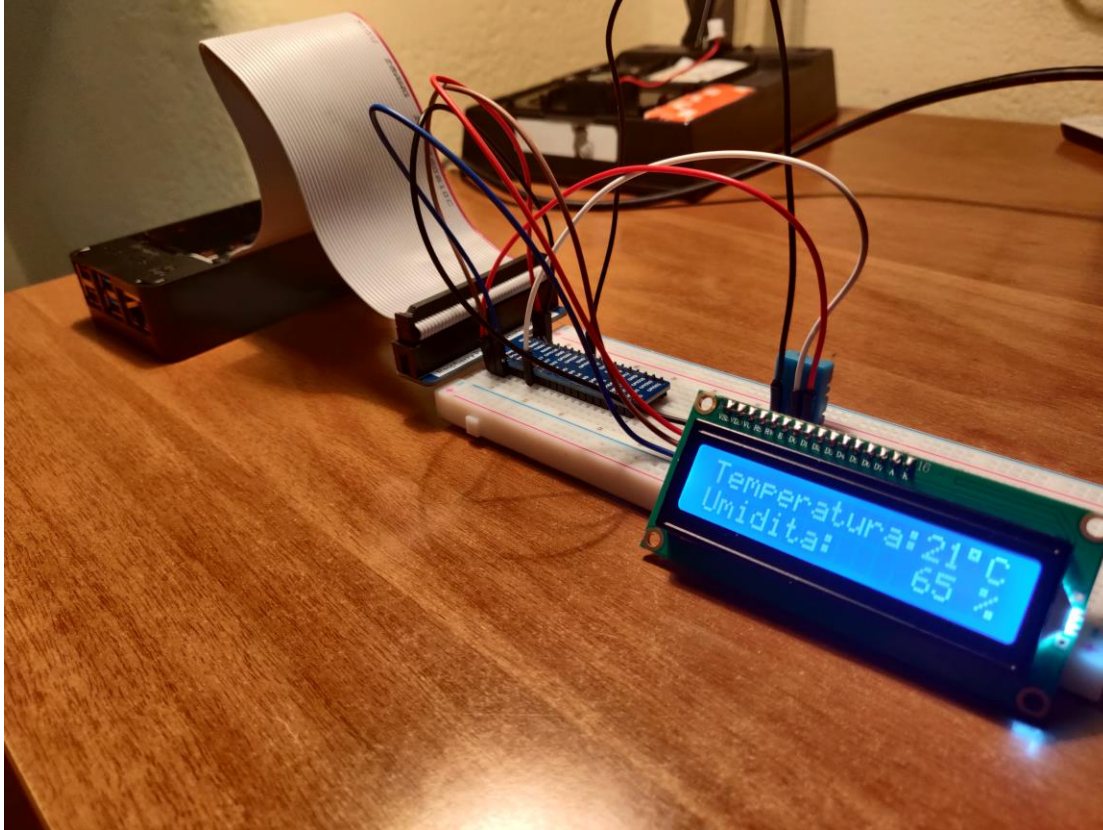


Bisogna collegarlo in questo modo:



I2C LCD	COLLEGAMENTO	RASBERRY
PIN 1(GND)	Cavo nero	GND
PIN 2(VCC)	Cavo rosso	5V
PIN 3(SDA)	Cavo marrone	SDA1
PIN 4(SCL)	Cavo blue	SLC1

1.3)RISULTATO FINALE



2.0) GUIDA AL SOFTWARE

2.1) ABILITARE L'I2C SU Raspberry PI

Per prima cosa bisogna abilitare l'I2C sul Raspberry PI eseguendo dal terminale il comando:

```
sudo raspi-config
```

Una volta eseguito il comando comparirà il menù di configurazione del Raspberry PI dove noi andremo a selezionare "**impostazioni avanzate**".

Ora selezioniamo "**I2C attivo/disabilitato caricato automaticamente**" e premiamo invio, usciamo dal menu e riavviamo il Raspberry per applicare la modifica.

2.2) INSTALLIAMO I2C-TOOLS E SMBUS

Ora dobbiamo installare un programma chiamato **I2C-tools**, che ci dirà l'indirizzo I2C del display LCD quando è collegato al Pi. Quindi apriamo il terminale ed eseguiamo il comando:

```
sudo apt-get install i2c-tools
```

Terminata l'installazione del programma, scarichiamo e installiamo **SMBUS**, che fornirà una libreria per python che ci permetterà di avere l'accesso al bus I2C sul Raspberry. Apriamo il terminale ed eseguiamo

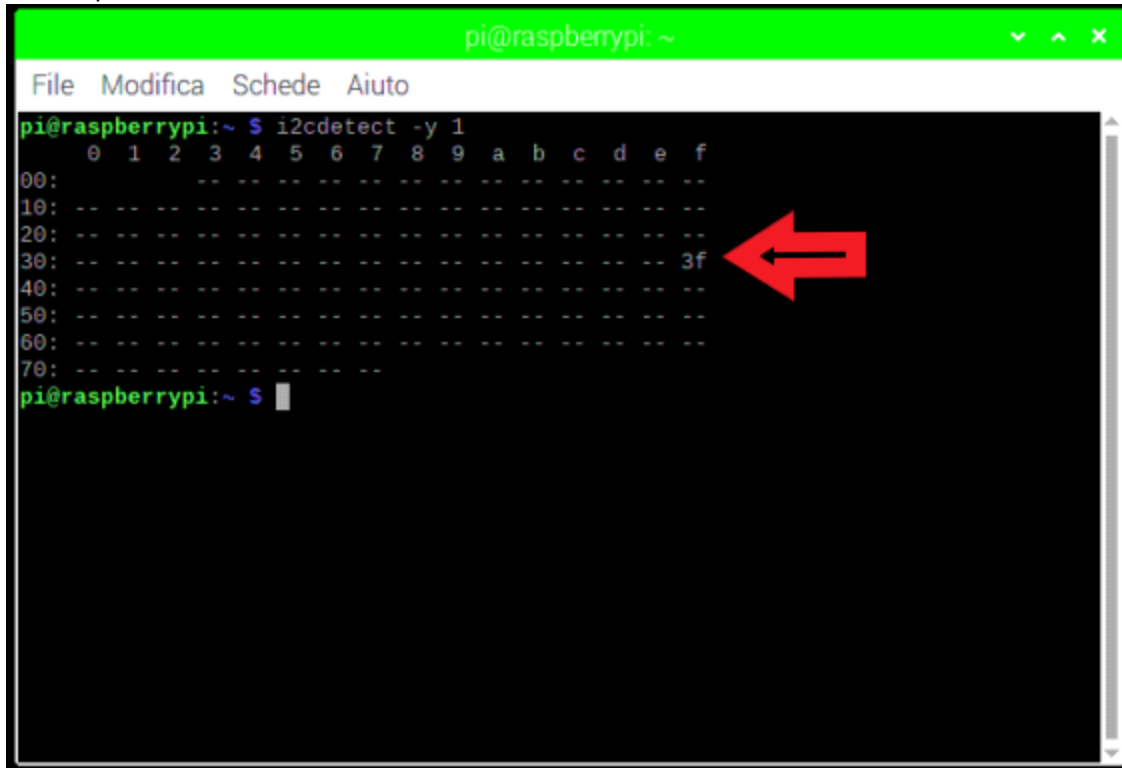
il comando:

```
sudo apt-get install python-smbus
```

Riavviamo il Raspberry un'altra volta. Con il nostro LCD collegato inseriamo al terminale il comando:

```
i2cdetect -y 1
```

così da poter identificare l'indirizzo I2C del nostro LCD.



```
pi@raspberrypi: ~  
File Modifica Schede Aiuto  
pi@raspberrypi:~ $ i2cdetect -y 1  
 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  3f  --  
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
pi@raspberrypi:~ $
```

In questo caso l'indirizzo I2C del mio LCD è 3f.

2.3) COMMENTO DEL CODICE SORGENTE "Meteo.py"

```
import RPi.GPIO as GPIO          ##importiamo la libreria GPIO(General Purpose Input/OutPut)  
import dht11                    ##importiamo la libreria dht11(sensore di temp e umidità)  
import I2C_LCD_driver          ## importiamo la libreria I2C LCD  
  
from time import *  
  
lcd = I2C_LCD_driver.lcd()      ##assegniamo a lcd la classe lcd presente in I2C_LCD_driver  
  
GPIO.setwarnings(False)        ## serve a disabilitare gli avvisi  
GPIO.setmode(GPIO.BCM)        ##L'opzione GPIO.BCM specifica che si fa riferimento ai pin in base alla  
##numerazione dei pin del Raspberry  
  
GPIO.cleanup()                 ## Ripristina tutte le porte di input utilizzate in questo programma
```

```

lcd.lcd_display_string("mini stazione", 1)

while True:

    istanza = dht11.DHT11(pin = 27)    ##(pin = pin GPIO) in questo caso il pin del segnale
                                       ##del DHT-11 è collegato al GPIO27

    risultato = istanza.read() ## Se è presente un segnale dal dht11

    # Fahrenheit:
    # risultato.temperature = (risultato.temperature * 1.8) + 32

    if risultato.is_valid():           ## se il segnale è valido stampa sul lcd temp e umidità
        lcd.lcd_display_string("Temperatura:%d%C" % (risultato.temperature, chr(223)), 1)
        lcd.lcd_display_string("Umidita:  %d %%%" % risultato.humidity, 2)

```

2.4)AVVIO AUTOMATICO DEL CODICE

Aggiungiamo il percorso del nostro codice meteo.py al crontab per far sì che si avvii all'accensione. Avviamo il terminale e digitiamo:

crontab -e

si aprirà l'editor di testo(nano). Alla fine del testo aggiungiamo la posizione del nostro codice:

@reboot python /home/pi/source/Meteo.py

Per salvare le modifiche fare clic su "CTRL-O poi "Y" e infine "CTRL-X".

IDEE DI SVILUPPO FUTURO

Aggiungere un L'ANEMOMETRO per il controllo della velocità del vento.

Creare un Database.