Raspberry PI si senzorul SHT11

Senzorul SHT11 ofera posibilitatea de a masura temperatura si umiditatea din mediul inconjurator cu o precizie ridicata. Se conecteaza la placa Raspberry PI prin intermediul a 2 pini digitali. Consumul senzorului este foarte redus, rezolutia temperaturii masurata de catre senzor este de 0.01 °C si 0.03 % pentru umiditatea relativa. In cel mai rau caz temperatura poate avea o acuratete de ± 2 °C si ± 3.5 % pentru umiditate.

Senzorul se alimenteaza cu o tensiune cuprinsa intre 2.4 si 5.5V, comunica printr-un protocol serial (Two-Wire Serial) si iesirea digitala este deja calibrata din fabrica.

In prima parte a tutorialului, vei conecta senzorul la placa Raspberry si vei afisa 3 valori distincte in terminal. In cea de-a doua parte, vei conecta un shield LCD pe care vei afisa 2 din cele 3 valori.

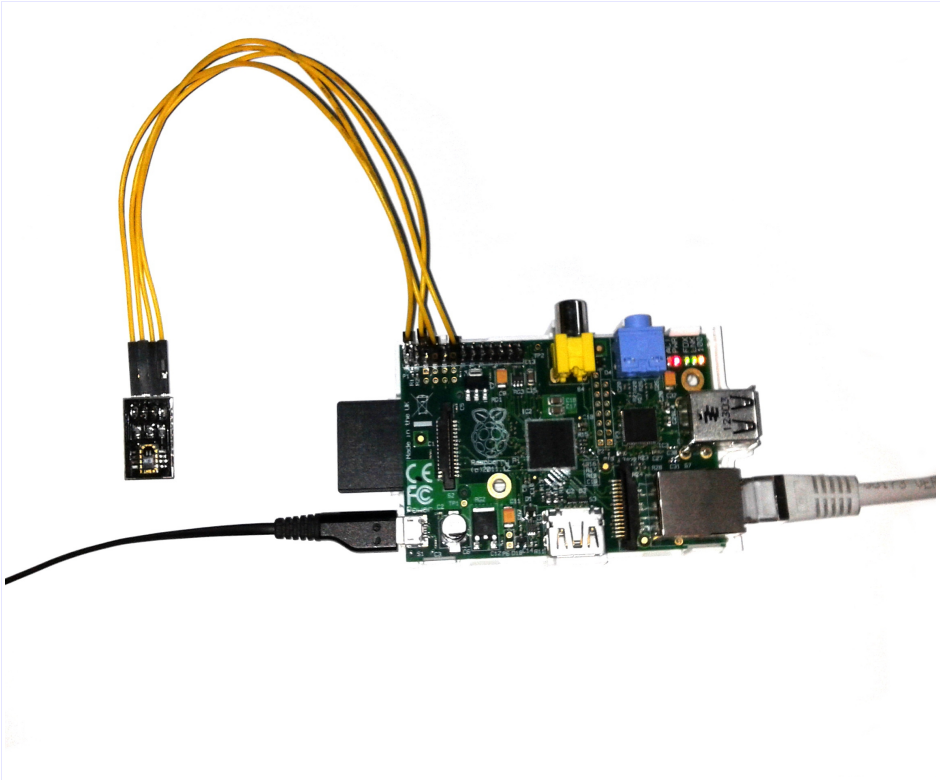
Cum conectez senzorul ?

Senzorul se conecteaza foarte simplu la placa Raspberry PI. Foloseste tabelul si imaginea de mai jos.

Raspberry PI 5V	SHT11 VCC
Raspberry PI GND	SHT11 GND
Raspberry PI pin 13	SHT11 pin DAT
Raspberry PI pin 7	SHT11 pin SCK

3.3V	1	2	5V
I2C0 SDA	3	4	DNC
I2C0 SCL	5	6	GROUND
GPIO4	7	8	UART TXD
DNC	9	10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 21	13	14	DNC
GPIO 22	15	16	GPIO 23
DNC	17	18	GPIO 24
SP10 MOSI	19	20	DNC
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
DNC	25	26	SP10 CE1 N

Asa arata senzorul conectat la placa Raspberry, conform tabelului.



Programul senzorului.

Exista un pachet Python special construit pentru acest senzor. Instaleaza pachetul in modul urmator:

1. Logheaza-te prin SSH in consola Raspbian. Poti folosi Putty daca vrei sa te conectezi de pe Windows sau ssh <ip> de pe Ubuntu.
2. Pachetul se afla aici: https://pypi.python.org/pypi/rpiSht1x si din documentatie el depinde de un alt modul.
3. Descarca modulul prin comanda:

sudo wget https://pypi.python.org/packages/source/R/RPi.GPIO/RPi.GPIO-0.4.1a.tar.gz

```
pi@raspberrypi ~/senzor_sht11 $ sudo wget https://pypi.python.org/packages/source/R/RPi.GPIO/RPi.GPIO-0.4.1a.tar.gz
--2013-06-30 00:28:37--  https://pypi.python.org/packages/source/R/RPi.GPIO/RPi.GPIO-0.4.1a.tar.gz
Resolving pypi.python.org (pypi.python.org)... 199.27.76.192, 199.27.76.129
Connecting to pypi.python.org (pypi.python.org)|199.27.76.192|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 15287 (15K) [application/x-gzip]
Saving to: `RPi.GPIO-0.4.1a.tar.gz.1'

100%[=====================================================================================================>] 15,287

2013-06-30 00:28:42 (454 KB/s) - `RPi.GPIO-0.4.1a.tar.gz.1' saved [15287/15287]
```

4. Dupa ce l-ai descarcat, acum il dezarhivezi prin comanda:

sudo tar -xvf Rpi.GPIO-0.4.1a.tar.gz

```
pi@raspberrypi ~/senzor_sht11 $ sudo tar -xvf RPi.GPIO-0.4.1a.tar.gz
RPi.GPIO-0.4.1a/
RPi.GPIO-0.4.1a/distribute_setup.py
RPi.GPIO-0.4.1a/INSTALL.txt
RPi.GPIO-0.4.1a/setup.cfg
RPi.GPIO-0.4.1a/setup.py
RPi.GPIO-0.4.1a/PKG-INFO
RPi.GPIO-0.4.1a/README.txt
RPi.GPIO-0.4.1a/RPi.GPIO.egg-info/
RPi.GPIO-0.4.1a/RPi.GPIO.egg-info/top_level.txt
RPi.GPIO-0.4.1a/RPi.GPIO.egg-info/PKG-INFO
RPi.GPIO-0.4.1a/RPi.GPIO.egg-info/SOURCES.txt
RPi.GPIO-0.4.1a/RPi.GPIO.egg-info/dependency_links.txt
RPi.GPIO-0.4.1a/RPi/
RPi.GPIO-0.4.1a/RPi/__init__.py
RPi.GPIO-0.4.1a/LICENCE.txt
RPi.GPIO-0.4.1a/.hg/
RPi.GPIO-0.4.1a/.hg/last-message.txt
RPi.GPIO-0.4.1a/MANIFEST.in
RPi.GPIO-0.4.1a/source/
RPi.GPIO-0.4.1a/source/cpuinfo.h
RPi.GPIO-0.4.1a/source/cpuinfo.c
RPi.GPIO-0.4.1a/source/c_gpio.c
RPi.GPIO-0.4.1a/source/c_gpio.h
RPi.GPIO-0.4.1a/source/py_gpio.c
RPi.GPIO-0.4.1a/test/
RPi.GPIO-0.4.1a/test/test.py
RPi.GPIO-0.4.1a/CHANGELOG.txt
pi@raspberrypi ~/senzor_sht11 $
```

5. Schima locatia in noul fisier:

cd Rpi.GPIO-0.4.1a

6. Instaleaza modulul prin comanda:

sudo python setup.py install

```
Installed /usr/local/lib/python2.7/dist-packages/RPi.GPIO-0.4.1a-py2.7-linux-armv6l.egg
Processing dependencies for RPi.GPIO==0.4.1a
Finished processing dependencies for RPi.GPIO==0.4.1a
pi@raspberrypi ~/senzor_sht11/RPi.GPIO-0.4.1a $
```

7. Pentru pachetul senzorului vei proceda la fel dar mai intai trebuie sa iesi din fisierul actual:

cd ..

sudo wget https://pypi.python.org/packages/source/r/rpiSht1x/rpiSht1x-1.2.tar.gz

```
pi@raspberrypi ~/senzor_sht11 $ sudo wget https://pypi.python.org/packages/source/r/rpiSht1x/rpiSht1x-1.2.tar.gz
--2013-06-30 00:46:31--  https://pypi.python.org/packages/source/r/rpiSht1x/rpiSht1x-1.2.tar.gz
Resolving pypi.python.org (pypi.python.org)... 199.27.76.192, 199.27.76.129
Connecting to pypi.python.org (pypi.python.org)|199.27.76.192|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8648 (8.4K) [application/x-gzip]
Saving to: `rpiSht1x-1.2.tar.gz'

100%[=====================================================================================================>] 8,648

2013-06-30 00:46:37 (12.5 KB/s) - `rpiSht1x-1.2.tar.gz' saved [8648/8648]

pi@raspberrypi ~/senzor_sht11 $
```

8. Dezarhiveaza pachetul:

sudo tar -xvf rpiSht1x-1.2.tar.gz

9. Schimba locatia:

cd rpiSht1x-1.2

10. Instaleaza pachetul prin comanda:

sudo python setup.py install

11. Codul sursa il vei scrie in nano dar mai intai schimba locatia:

cd ..

sudo nano senzorSHT11.py

12. Copiaza in nano codul sursa:

```
from sht1x.Sht1x import Sht1x as SHT1x
dataPin = 13
clkPin = 7
sht1x = SHT1x(dataPin, clkPin, SHT1x.GPIO_BOARD)

temperature = sht1x.read_temperature_C()
humidity = sht1x.read_humidity()
dewPoint = sht1x.calculate_dew_point(temperature, humidity)

print("Temperature: {} Humidity: {} Dew Point:
{}".format(temperature, humidity, dewPoint))
```

```
  GNU nano 2.2.6                                    File: senzorSHT11.py

from sht1x.Sht1x import Sht1x as SHT1x
dataPin = 11
clkPin = 7
sht1x = SHT1x(dataPin, clkPin, SHT1x.GPIO_BOARD)

temperature = sht1x.read_temperature_C()
humidity = sht1x.read_humidity()
dewPoint = sht1x.calculate_dew_point(temperature, humidity)

print("Temperature: {} Humidity: {} Dew Point: {}".format(temperature, humidity, dewPoint))
```

13. Salveaza-l cu CTRL X si Y si executa-l cu:

```
sudo python senzorSHT11.py
```

```
pi@raspberrypi ~/senzor_sht11 $ sudo python senzorSHT11.py
Temperature: 27.51 Humidity: 54.7550730205 Dew Point: 17.5898163705
pi@raspberrypi ~/senzor_sht11 $
```

14. Daca vrei sa obtii citiri la fiecare 5 secunde atunci modifica fisierul py:

```
import time
from sht1x.Sht1x import Sht1x as SHT1x
dataPin = 13
clkPin = 7
sht1x = SHT1x(dataPin, clkPin, SHT1x.GPIO_BOARD)

while 1:
    temperature = sht1x.read_temperature_C()
    humidity = sht1x.read_humidity()
    dewPoint = sht1x.calculate_dew_point(temperature, humidity)
    print("Temperature: {} Humidity: {} Dew Point:
{}".format(temperature, humidity, dewPoint))
    time.sleep(5)
```

```
  GNU nano 2.2.6                                    File: senzorSHT11.py

import time
from sht1x.Sht1x import Sht1x as SHT1x
dataPin = 11
clkPin = 7
sht1x = SHT1x(dataPin, clkPin, SHT1x.GPIO_BOARD)

while 1:
 temperature = sht1x.read_temperature_C()
 humidity = sht1x.read_humidity()
 dewPoint = sht1x.calculate_dew_point(temperature, humidity)
 print("Temperature: {} Humidity: {} Dew Point: {}".format(temperature, humidity, dewPoint))
 time.sleep(5)
```

```
Temperature: 27.44 Humidity: 55.276329442 Dew Point: 17.6751989319
Temperature: 27.4 Humidity: 55.1802058845 Dew Point: 17.6104108866
Temperature: 27.43 Humidity: 55.2433319805 Dew Point: 17.656427966
Temperature: 27.44 Humidity: 55.214645728 Dew Point: 17.6574726838
Temperature: 27.43 Humidity: 55.1830794845 Dew Point: 17.6391053553
```

Cum afisez temperatura si umiditatea pe un LCD?

Daca vrei sa afisezi temperatura si umiditatea pe un LCD, atunci iti propun shield-ul LCD 16x2. Exista 2 tipuri de shield: primul tip se conecteaza la Raspberry PI prin cablu cobbler iar cel de-al doilea tip se infige direct in portul GPIO, fara nici un alt cablu.

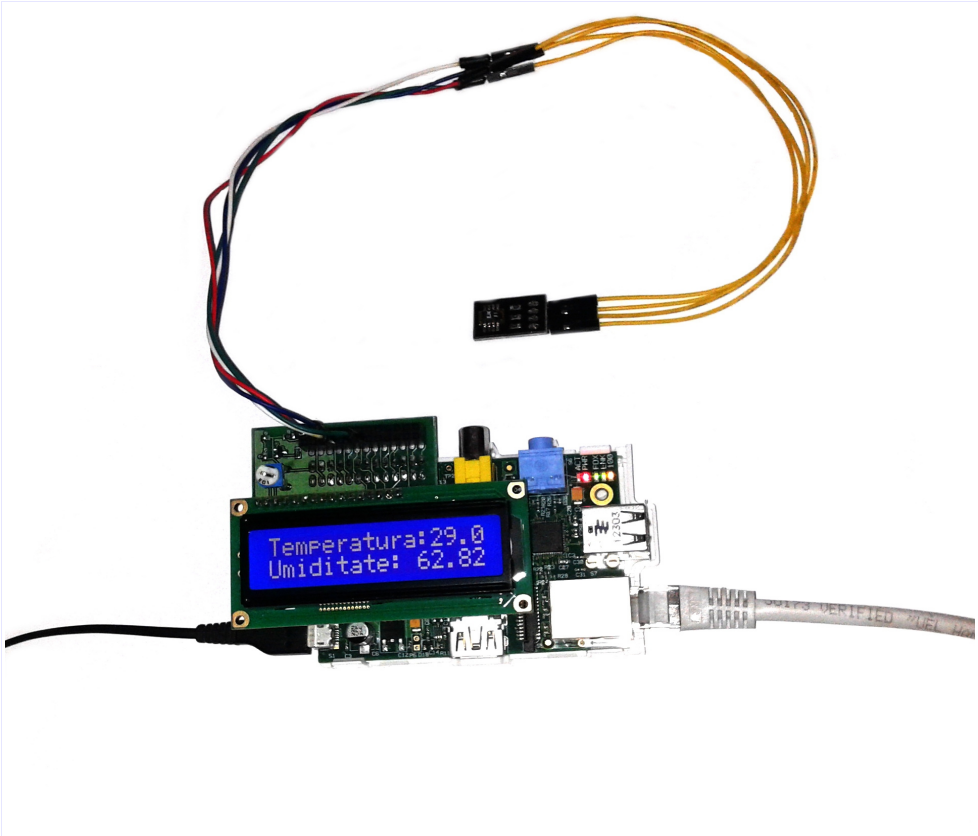
Placa expune in acelasi timp si toti pinii GPIO ai Raspberry PI (conectorul 2X13, nepopulat, de langa LCD). Daca vrei sa ai acces la orice pin al portului GPIO dar in acelasi timp sa folosesti si LCD-ul, atunci este necesara lipirea conectorului mama 2x13.

In tutorialul de fata este necesara lipirea acestui conector, altfel nu vei putea conecta senzorul SHT11.

Cum procedez?

1. Lipeste bareta mama 2x13 pe shield. Daca vrei sa sari peste aceasta operatiune, atunci la achizitionarea shield-ului opteaza pentru bareta lipita.
2. Infige shield-ul in portul placii Raspberry PI.
3. Conecteaza senzorul SHT11, prin fire, urmand acelasi tabel de mai sus.
4. Copiaza codul listat mai jos si executa-l.

Cum arata shield-ul si senzorul conectat ?



```
import RPi.GPIO as GPIO
from sht1x.Sht1x import Sht1x as SHT1x
import time

LCD_RS = 22
LCD_E  = 18
LCD_D4 = 16
LCD_D5 = 11
LCD_D6 = 12
LCD_D7 = 15
LED_ON = 15

LCD_WIDTH = 16
LCD_CHR = True
LCD_CMD = False

LCD_LINE_1 = 0x80
LCD_LINE_2 = 0xC0

E_PULSE = 0.00005
E_DELAY = 0.00005

dataPin = 13
clkPin = 7

def main():

  lcd_init()
  lcd_byte(LCD_LINE_1, LCD_CMD)
  lcd_string("ROBOFUN.RO",2)
  lcd_byte(LCD_LINE_2, LCD_CMD)
  lcd_string("Raspberry PI",2)
  time.sleep(5)
  while 1:
        sht1x = SHT1x(dataPin, clkPin, SHT1x.GPIO_BOARD)
        temperature = sht1x.read_temperature_C()
        humidity = sht1x.read_humidity()
        dewPoint = sht1x.calculate_dew_point(temperature, humidity)

        print("Temperatura: {} Umiditate: {} Punct de roua:
{}".format(temperature, humidity, dewPoint))

        textLCDOne = str("%.1f" % temperature)
        textLCDOne = "Temperatura:" + textLCDOne
        textLCDTwo = str("%.2f" % humidity)
        textLCDTwo = "Umiditate: " + textLCDTwo

        lcd_init()

        lcd_byte(LCD_LINE_1, LCD_CMD)
        lcd_string(textLCDOne,1)
        lcd_byte(LCD_LINE_2, LCD_CMD)
        lcd_string(textLCDTwo,1)
```

```
def lcd_init():
  GPIO.setwarnings(False)
  GPIO.setmode(GPIO.BOARD)
  GPIO.setup(LCD_E, GPIO.OUT)
  GPIO.setup(LCD_RS, GPIO.OUT)
  GPIO.setup(LCD_D4, GPIO.OUT)
  GPIO.setup(LCD_D5, GPIO.OUT)
  GPIO.setup(LCD_D6, GPIO.OUT)
  GPIO.setup(LCD_D7, GPIO.OUT)
  GPIO.setup(LED_ON, GPIO.OUT)

  lcd_byte(0x33,LCD_CMD)
  lcd_byte(0x32,LCD_CMD)
  lcd_byte(0x28,LCD_CMD)
  lcd_byte(0x0C,LCD_CMD)
  lcd_byte(0x06,LCD_CMD)
  lcd_byte(0x01,LCD_CMD)

def lcd_string(message,style):
  # style=1 Left justified
  # style=2 Centred
  # style=3 Right justified

  if style==1:
    message = message.ljust(LCD_WIDTH," ")
  elif style==2:
    message = message.center(LCD_WIDTH," ")
  elif style==3:
    message = message.rjust(LCD_WIDTH," ")

  for i in range(LCD_WIDTH):
    lcd_byte(ord(message[i]),LCD_CHR)

def lcd_byte(bits, mode):
  # Send byte to data pins
  # bits = data
  # mode = True  for character
  #        False for command

  GPIO.output(LCD_RS, mode) # RS

  # High bits
  GPIO.output(LCD_D4, False)
  GPIO.output(LCD_D5, False)
  GPIO.output(LCD_D6, False)
  GPIO.output(LCD_D7, False)
  if bits&0x10==0x10:
    GPIO.output(LCD_D4, True)
  if bits&0x20==0x20:
    GPIO.output(LCD_D5, True)
  if bits&0x40==0x40:
    GPIO.output(LCD_D6, True)
  if bits&0x80==0x80:
```

```
    GPIO.output(LCD_D7, True)

  # Toggle 'Enable' pin
  time.sleep(E_DELAY)
  GPIO.output(LCD_E, True)
  time.sleep(E_PULSE)
  GPIO.output(LCD_E, False)
  time.sleep(E_DELAY)

  # Low bits
  GPIO.output(LCD_D4, False)
  GPIO.output(LCD_D5, False)
  GPIO.output(LCD_D6, False)
  GPIO.output(LCD_D7, False)
  if bits&0x01==0x01:
    GPIO.output(LCD_D4, True)
  if bits&0x02==0x02:
    GPIO.output(LCD_D5, True)
  if bits&0x04==0x04:
    GPIO.output(LCD_D6, True)
  if bits&0x08==0x08:
    GPIO.output(LCD_D7, True)

  # Toggle 'Enable' pin
  time.sleep(E_DELAY)
  GPIO.output(LCD_E, True)
  time.sleep(E_PULSE)
  GPIO.output(LCD_E, False)
  time.sleep(E_DELAY)

if __name__ == '__main__':
  main()
```