

راه اندازی سنسور دمای دیجیتال DS18B20 با برد رزبری پای | درس دوازدهم

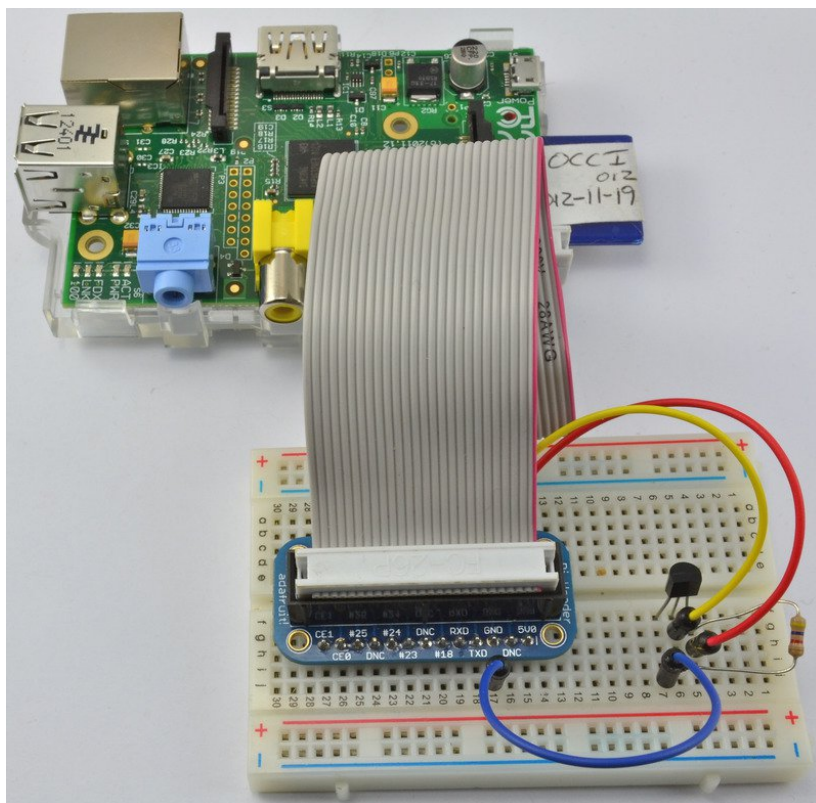


قسمت دوازدهم

سری آموزش رزبری پای: درس دوازدهم

این سنسور کوچک سه پایه، ظاهری مشابه ترانزیستور داشته و یک دستگاه دیجیتال دقیق می باشد. سیستم عامل لینوکس Occidentalis برای برد رسپیری پای (و Raspbian از دسامبر 2012) از سنسور دمای دیجیتال تک سیمه DS18B20 پشتیبانی می کند.

[کنترل استیر موتور با برد رزبری پای | درس یازدهم](#)



در این درس نحوه راه اندازی سنسور دمای دیجیتال DS18B20 توسط برد رزبری پای، برای اندازه گیری دما را یاد خواهید گرفت.

از آنجایی که برد رزبری پای مبدل آنالوگ به دیجیتال (ADC) ندارد، نمی تواند به طور مستقیم از سنسور دمای آنالوگ مانند TMP36 استفاده کند، لذا سنسور دمای دیجیتال DS18B20 گزینه ی مناسبی برای اندازه گیری دما می باشد.

کد های کتابخانه:

اگر از پایتون استفاده می کنید، می توانید از [کتابخانه سنسور دمای دیجیتال DS18B20](#) استفاده کنید، که به خوبی با برد رزبری پای کار می کند.

قطعات مورد نیاز:

برای انجام این پروژه شما نیاز به قطعات زیر دارید که می توانید آنها را از فروشگاه صنعت بازار تهیه کنید:

برد رزبری پای	1 عدد	
برد بورد	1 عدد	
سیم اتصال برد بورد	1 بسته	
برد GPIO تی شکل + کابل	1 عدد	
سنسور دمای دیجیتال DS18B20	1 عدد	
مقاومت 4.7k تا 10k	1 عدد	

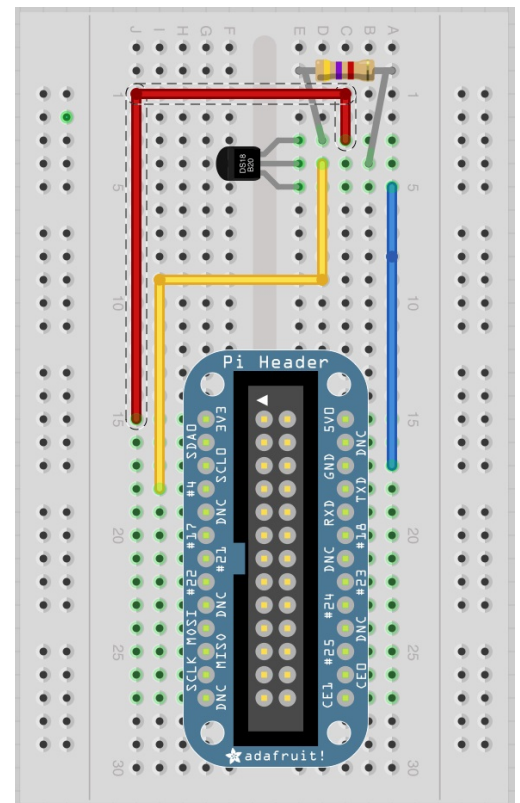
سخت افزار:

در شکل زیر نمایشی ساده از اتصالات DS18B20 نشان داده شده است.

سنسور های تک سیمه DS18B20 برخلاف سایر سنسورها می توانند به صورت موازی متصل شوند. تمامی این سنسورهای موازی باید از یک پایه مشترک استفاده کنند، اما فقط یک مقاومت 4.7k برای همه آنها لازم است.

این مقاومت به عنوان یک pullup (بالا برنده) برای خط های داده عمل می کند و وجود آن برای پایدار بودن داده ها ضروری است.

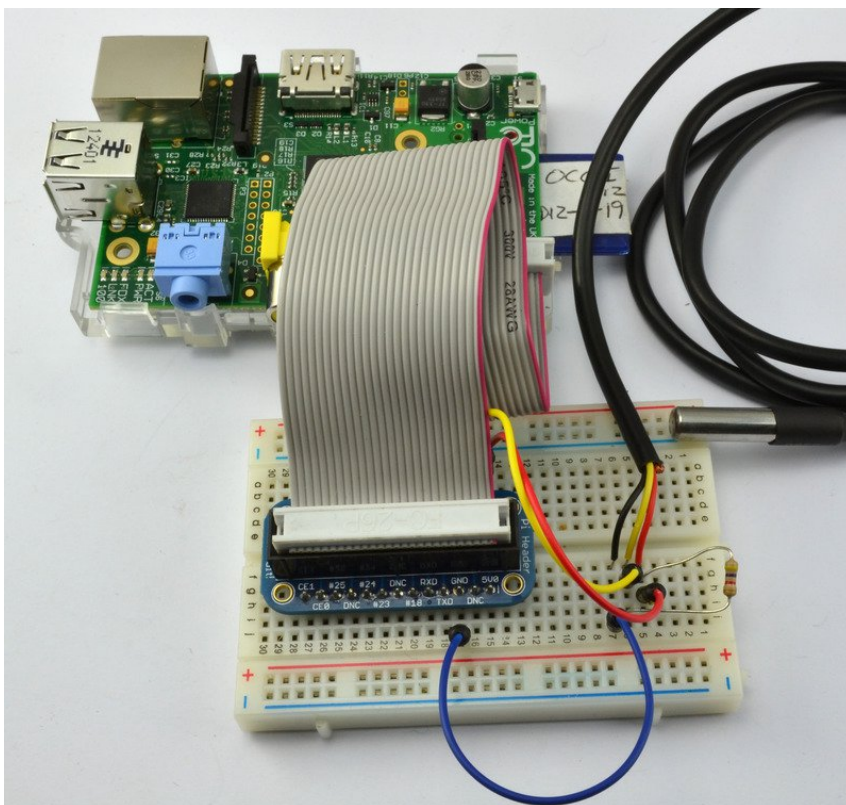
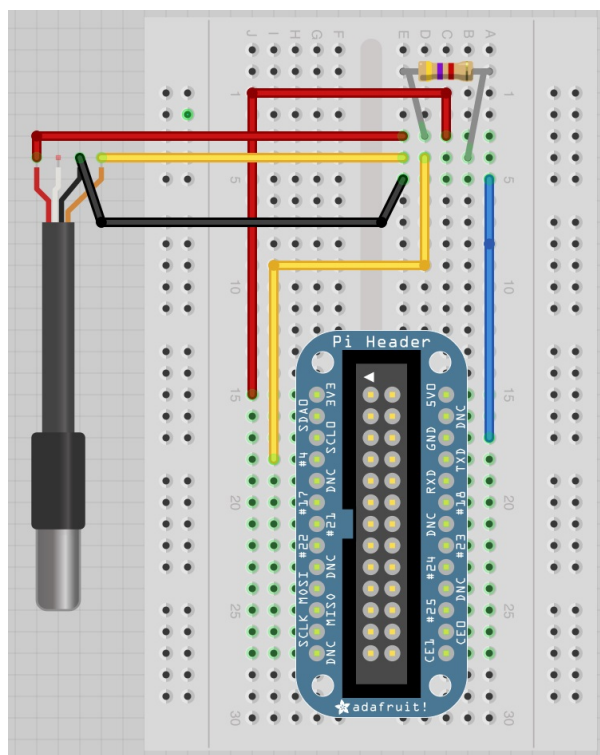
در راه اندازی DS18B20 دقت کنید . لبه ی خمیده و منحنی آن باید به سمت چپ باشد. اگر آنرا به صورت اشتباه وصل کنید سنسور داغ و سپس آسیب می بیند.



نسخه ضد آب DS18B20 ، دارای 3 سیم با رنگهای قرمز و سیاه و زرد می باشد. سر سیم مسی نیازی به اتصال ندارد.

در صورتی که از نسخه "high temperature" می کنید، سیم نارنجی را به منبع 3.3 ولت، سیم سفید را به زمین و سیم آبی نیز برای داده ها می باشد.

در اینجا نیز نیاز به یک مقاومت 4.7 الی 10 کیلو اهمی میان سیم داده و منبع 3.3 ولتی دارید.



سنسور دمای دیجیتال DS18B20:

اگرچه ظاهر DS18B20 مشابه ترانزیستور های رایج است، اما در واقع داخل آن اتفاقات زیادی می افتد.

این تراشه داده های خروجی را از طریق یک سیم در اختیار کاربر قرار می دهد و همچنین دارای کنترلر منطقی و سنسور دما در درون خود است.

پیام های دیجیتال از طریق پایه های خروجی ارسال می شود، و سیستم عامل های Raspbian/Occidentalis دارای رابطی برای خواندن این پیام ها هستند. می توان قبل از اجرای کامل برنامه، از طریق خط دستور و یا در SSH که درس ششم بیان گردید، قطعه را آزمایش کرد.

اضافه کردن پشتیبان تک سیمه:

با اضافه کردن خط زیر در فایل boot/config.txt/ شروع کنید.

می توانید این فایل را از طریق اجرای دستور `sudo nano /boot/config.txt` ویرایش کنید، سپس به انتهای آن رفته و خط زیر را وارد کنید:

```
dtoverlay=w1-gpio
```

سپس سیستم را با دستور `sudo reboot` راه اندازی مجدد کنید. با بالا آمدن رسیبری پای و وارد شدن مجدد کاربر، دستورات زیر را در terminal window وارد کنید، زمانی که در مسیر 'devices' قرار دارید، مسیر که با '28-' آغاز می شود، ممکن است نام های متفاوتی داشته باشد، بنابراین به نام مسیری مورد نظر دقت کنید.

```
sudo modprobe w1-gpio
sudo modprobe w1-therm
cd /sys/bus/w1/devices
ls
(cd 28-xxxx (change this to match what serial number pops up
cat w1_slave
```

```
pi@raspberrypi: /sys/bus/w1/devices/28-000003cee4ca
File Edit Tabs Help
pi@raspberrypi /sys/bus/w1/devices $ sudo modprobe w1-gpio
pi@raspberrypi /sys/bus/w1/devices $ sudo modprobe w1-therm
pi@raspberrypi /sys/bus/w1/devices $ cd /sys/bus/w1/devices/
pi@raspberrypi /sys/bus/w1/devices $ ls
28-000003cee4ca w1 bus master
pi@raspberrypi /sys/bus/w1/devices $ cd 28-000003cee4ca
pi@raspberrypi /sys/bus/w1/devices/28-000003cee4ca $ cat w1_slave
4b 01 4b 46 7f ff 05 10 e1 : crc=e1 YES
4b 01 4b 46 7f ff 05 10 e1 t=20687
pi@raspberrypi /sys/bus/w1/devices/28-000003cee4ca $ cat w1_slave
a2 01 4b 46 7f ff 0e 10 d8 : crc=d8 YES
a2 01 4b 46 7f ff 0e 10 d8 t=26125
pi@raspberrypi /sys/bus/w1/devices/28-000003cee4ca $
```

رابط کاربری که در بالا مشاهده می کنید، کمی غیر قابل اعتماد است، اما خوشبختانه اگر دمای معتبری وجود داشته باشد، در اختیار قرار می دهد. مقدار نمایش داده شده شبیه یک فایل است، لذا تنها کاری که لازم، خوانده آن است.

Yes یا No پاسخی است که در انتهای خط اول نمایش داده می شود. اگر yes باشد، دما (ضرب در 1000 درجه سانتیگراد) در انتهای خط دوم نمایش داده می شود. در مثال بالا دما در واقع 20.687 و سپس 26.125 درجه سانتیگراد است.

اگر شما از بیش از یک سنسور استفاده می کنید، چند فایل xxx-28 خواهید دید، که هرکدام یک شماره سریال منحصر به فرد دارند. ممکن است در هر زمان نیاز به متصل کردن یکی از سنسور ها باشد. کافیسٹ برچسب (label) سنسور و فایل ایجاد شده را بررسی کرد.

نرم افزار:

نرم افزار پایتون با هر پیام خطا تعامل را ایجاد کرده و دما را برحسب سانتی گراد و فارنهایت گزارش می دهد.

```
import os
import glob
import time

('os.system('modprobe w1-gpio
('os.system('modprobe w1-therm

'/base_dir = '/sys/bus/w1/devices
[device_folder = glob.glob(base_dir + '28*')[0
'device_file = device_folder + '/w1_slave

:()def read_temp_raw
('f = open(device_file, 'r
()lines = f.readlines
()f.close
return lines

:()def read_temp
()lines = read_temp_raw
:'while lines[0].strip()[-3:] != 'YES
(time.sleep(0.2
()lines = read_temp_raw
('=equals_pos = lines[1].find('t
:if equals_pos != -1
[:temp_string = lines[1][equals_pos+2
temp_c = float(temp_string) / 1000.0
temp_f = temp_c * 9.0 / 5.0 + 32.0
return temp_c, temp_f

:while True
    (()print(read_temp
```

```
(time.sleep(1
```

برنامه با صدور دستور modprobe که برای شروع کار رابط کاربری مورد نیاز می باشد، آغاز می گردد.

3 خط بعدی، فایلی که خواندن پیام از آن امکان پذیر است را پیدا می کند.

ممکن است گاه گاهی در هنگام کار با Raspbian برای خواندن دما، مشکلاتی گزارش شود. اگر مشکلات مشابه آنچه که ذکر شد، پیش آمد. به جای تابع `function read_temp_raw` کد زیر را وارد کنید. برای این کار نیاز به اضافه کردن خط `import subprocess` در بالای فایل می باشد.

```
def read_temp_raw:()
catdata = subprocess.Popen(['cat',device_file], stdout=subprocess.PIPE,
(stderr=subprocess.PIPE
)out,err = catdata.communicate
('out_decode = out.decode('utf-8
('lines = out_decode.split('\n
return lines
```

خواندن دما توسط دو تابع انجام می گیرد. تابع `read_temp_raw`، دو خط از رابط کاربری را بازخوانی می کند. تابع `read_temp` پیام های بد را چک می کند و این کار را آنقدر ادامه می دهد تا به پیامی که در انتهای خطش `yes` دارد، برسد. تابع دو مقدار را به عنوان خروجی بر می گرداند، اولی دما بر حسب سانتیگراد و دیگری دما بر حسب فارنهایت می باشد.

باتوجه به مثال زیر می توان این دو عدد (خروجی) را متمایز نشان داد.

```
()deg_c, deg_f = read_temp
```

حلقه اصلی برنامه، ساده است. قبل از یک ثانیه استراحت، دما را خوانده و نمایش می دهد. برای بارگذاری برنامه در برد رسیبری پای، می توانید از `ssh` برای اتصال به برد رسیبری پای استفاده کنید. یک پنجره ویرایشگر را با استفاده از دستور زیر اجرا کنید:

```
nano thermometer.py
```

سپس کد زیر را وارد کرده و در نهایت با استفاده از `CTRL-x` و سپس `Y` آن را ذخیره کنید.

```

def read_temp_raw():
    f = open(device_file, 'r')
    lines = f.readlines()
    f.close()
    return lines

def read_temp():
    lines = read_temp_raw()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = read_temp_raw()
    equals_pos = lines[1].find('t=')
    if equals_pos != -1:
        temp_string = lines[1][equals_pos+2:]
        temp_c = float(temp_string) / 1000.0
        temp_f = temp_c * 9.0 / 5.0 + 32.0
        return temp_c, temp_f

```

تست و نکات نهایی:

برنامه باید به صورت super user اجرا گردد، لذا لازم است برای شروع کد زیر وارد شود:

```
sudo python thermometer.py
```

اگر همه چیز خوب پیش برود متنی مانند شکل زیر نمایش داده می شود:

```

pi@raspberrypi /media/65EE-17E2 $ sudo python thermometer.py
(21.375, 70.475)
(22.0, 71.6)
(23.5, 74.3)
(24.375, 75.875)
(24.75, 76.55)
(25.062, 77.11160000000001)
(25.625, 78.125)
(25.75, 78.35)
(25.875, 78.575)
(25.812, 78.4616)
(25.25, 77.45)
(24.812, 76.6616)
(24.375, 75.875)
(23.687, 74.6366)
(23.375, 74.075)
(23.062, 73.5116)
(22.75, 72.95)

```


با گذاشتن انگشت خود روی سنسور آنرا گرم تر کنید و نتیجه را مشاهده کنید.

اضافه کردن سنسور های بیشتر:

میتوان سنسور های DS18B20 بیشتری را به صورت اتصال- موازی اضافه کرد. باید تمامی پین های VCC، data و ground را به یکدیگر متصل کنید. و از یک مقاومت 4.7 کیلو اهمی استفاده کنید. شما مسیرهای متفاوت (/sys/bus/w1/devices/28- nnnnn) را مشاهده خواهید کرد که هرکدام دارای یک شماره ی سریال منحصر بفرد به عنوان directionery name می باشند.

کد پایتون بالا فقط برای یک سنسور اجرا می شود، لذا باید آنرا برای خواندن از سنسور های مختلف در یک زمان تغییر داد.

راه اندازی سنسور تشخیص حرکت با برد رزبری پای | درس سیزدهم

ترجمه شده و تکمیل توسط تیم الکترونیک صنعت بازار | منبع: سایت adafruit