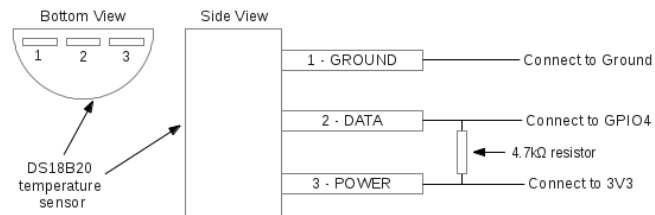


Raspberry Pi

DS18B20 1-wire Temperature Sensor:



We'll need to enable the One-Wire interface before the Pi can receive data from the sensor. Once you've connected the DS18B20, power up your Pi and log in, then follow these steps to enable the One-Wire interface:

At the command prompt, enter:

```
sudo nano /boot/config.txt, then add dtoverlay=w1-gpio to the bottom of the file, exit.  
sudo reboot
```

Log in to the Pi again, and at the command prompt enter:

```
sudo modprobe w1-gpio  
sudo modprobe w1-therm  
cd /sys/bus/w1/devices
```

list the devices: 28-000000xxxx (temp sensor)

```
cd 28-000000xxxx
```

```
cat w1_slave           t=xxxx the raw temperature reading output by the sensor.
```

Temperature Output to SSH Terminal

This is a basic Python program that will output the temperature readings in Fahrenheit and Celsius to your SSH terminal:

```
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)
```

Display Temperature on web site:

install the Apache web server software:

```
sudo apt-get install apache2 -y
sudo reboot
```

Once you Pi has booted back up, on another device or PC browse to the IP address of your Raspberry Pi. You should be greeted with the Apache default web page.

Modifying the Raspberry Pi Web Server Default Page:

If you'd have a little fun and edit the contents of the Raspberry Pi web server's default page you can do so by running the following command:

```
sudo nano /var/www/html/index.html
```

You can also copy this page to another file as follows to create an additional page:

```
sudo cp /var/www/html/index.html /var/www/html/mypage.html
```

Installing and using PHP on the Raspberry Pi Web Server:

If you want to do anything beyond basic web pages you'll want install PHP on your Raspberry Pi web server. This will allow you to "execute code" on your web page and build dynamic content.

To install PHP, follow these steps:

Step 1: Open a terminal window and type the following command to install PHP.

```
sudo apt-get install php7.0 libapache2-mod-php7.0 -y          (note removed 7.0)
```

Step 2: Create a PHP page in the html directory in Nano.

```
sudo nano /var/www/html/myphppage.php
```

Step 3: Enter the following code and then save the php file.

```
<?php phpinfo(); ?>          The PHP info page.
```

Step 4: In a browser on another computer, browse to the IP address of your Raspberry Pi web server and this page: <http://10.2.3.4/myphppage.php>

You will be presented with a webpage showing some basic stats and information on your server.

To display static temperature:

```
<?php
//File to read
$file = '/sys/devices/w1_bus_master1/28-0116258464ee/w1_slave';

//Read the file line by line
$lines = file($file);

//Get the temp from second line
$temp = explode('=', $lines[1]);

//Setup some nice formatting (i.e. 21,3)
$temp = number_format(($temp[1] / 1000) * 9.0 / 5.0 + 32.0 , 1, '.', '');

//And echo that temp
echo $temp . " °F";
?>
```