MINI PROJECT REPORT
ON

"Temperature System With Ubidot Using Raspberry-Pi"

Submitted by

1. Gayatri Asodekar(5)
2. Vijay Chaudhari(15)

Under the guidance of

Prof. Santosh Khorne

DATTA MEGHE COLLEGE OF ENGINEERING, AIROLI.

MUMBAI UNIVERSITY

Year 2017-2018

Date: -------------------

CERTIFICATE

This is to certify that the project report entitled Temperature System With Ubidot Using Raspberry-Pi being submitted by Gayatri Asodekar and Vijay Chaudhari in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer engineering to the Mumbai University is a record of bonafied work carried out by him under my guidance and supervision.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

Guide Department
Prof. Santosh Khorne

HoD. Computer Engineering
Prof.A.P.Pande

ACKNOWLEDGEMENT

Those who walk the difficult path to success never rest at this destiny they walk ahead towards greater success. I consider myself lucky to work under guidance of such talented and experienced people during the preparation my mini project who guided me all through it.

I am indebted to **HOD Prof. A.P.Pande** for his support at various stages during the formation of this piece of work.

A special mention must go to my guide **Prof. Santosh.W.Khorne** who supported me with her vast knowledge, experience and suggestion. Only their inspiration has made this mini project easy and interesting.

I would also like to thank our principle **Dr. Dr. S.D.Sawarkar** for his warm support and providing all necessary facilities to us, the student.

Last but not the least I am thankful to all the **Teachers and Staff** members of Computer Engineering Department for their expert guidance and continuous encouragement throughout to see that the maximum benefit is taken out of his experience.

INDEX

ABSTRACT

A temperature monitoring system provides valuable insights in both commercial and industrial environments to reduce inefficiencies or maintain quality of products and their quality. What if I told you that you can monitor the temp of your self-built wine-cellar or your family's aquarium at home using the same device. Further, what if I told you that the same device could be used to monitor air and liquid temperatures of fluids at your factory too? The makers of our world have made this possible and this guide is here to help kickstart your own initiatives at home or on the shop floor.

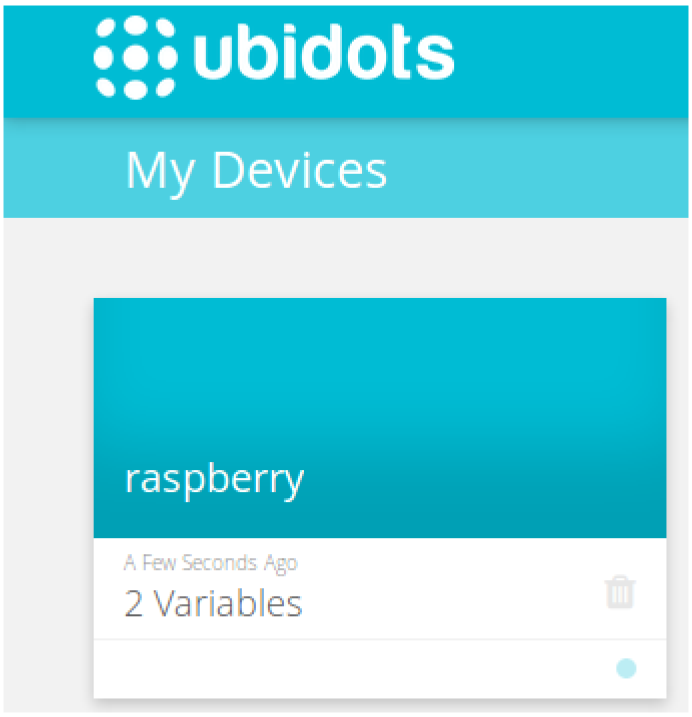
This guide will be your tutorial for a simple DIY temperature monitoring system that is also waterproof to boot. Using a **Raspberry Pi** and **Ubidots** we'll show you how to connect your Pi and display in real-time your temperature system's metrics. Using Ubidots, you can also create emails or SMS events to ensure your "variable" (in this case, the temperature) remains within a set of defined limits assigned by you to ensure quality and efficiency of your system's conditions.

For this project we are going to use a **1-wire pre-wired** and **waterproof** version of the **DS18B20** sensor. What is **1-wire?** It's a communication protocol that makes connecting your IoT sensors simpler by aggregating all cabling into is a single wire (...well actually it's three, two are ground and power connections for energy, the third being the 1-wire for data transmission).

REQUIREMENT

- Raspberry Pi Model
- Raspbian Stretch Lite(OS)
- OneWire Temperature Sensor - DS18B20
- Ubidots account -or- Educational License
- Resistor - 4.7KΩ
- Jumpers

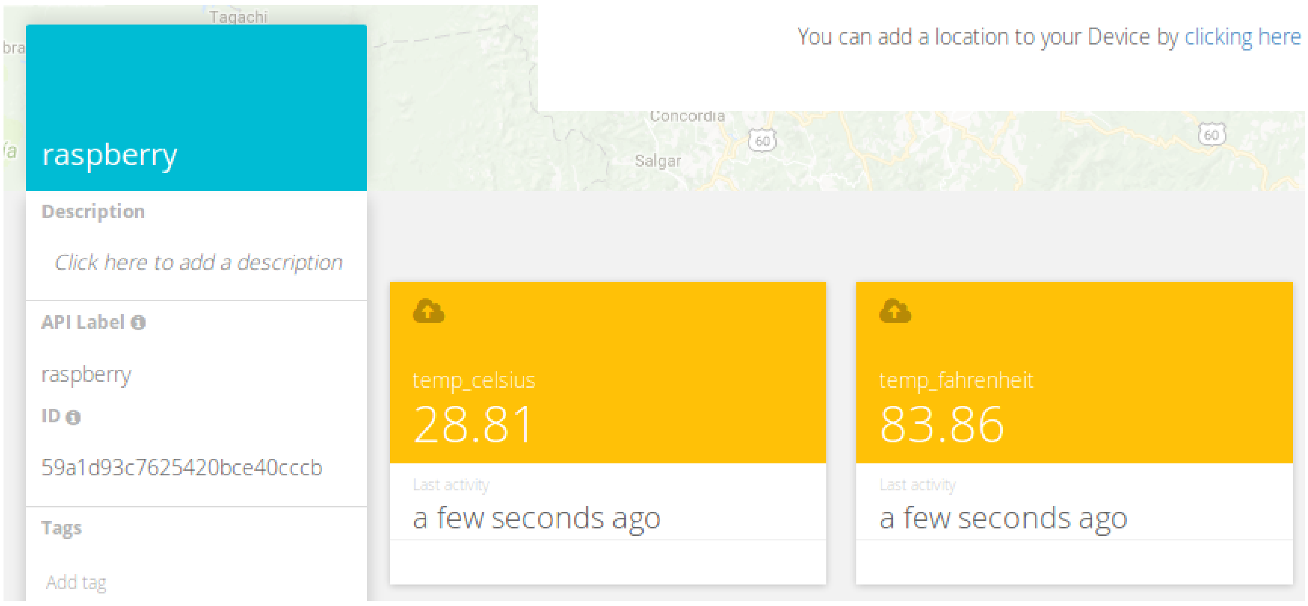
Screen Shots



raspberry

A Few Seconds Ago

2 Variables



You can add a location to your Device by clicking here

raspberry

Description

Click here to add a description

API Label

raspberry

ID

59a1d93c7625420bce40cccb

Tags

Add tag

temp_celsius

28.81

Last activity

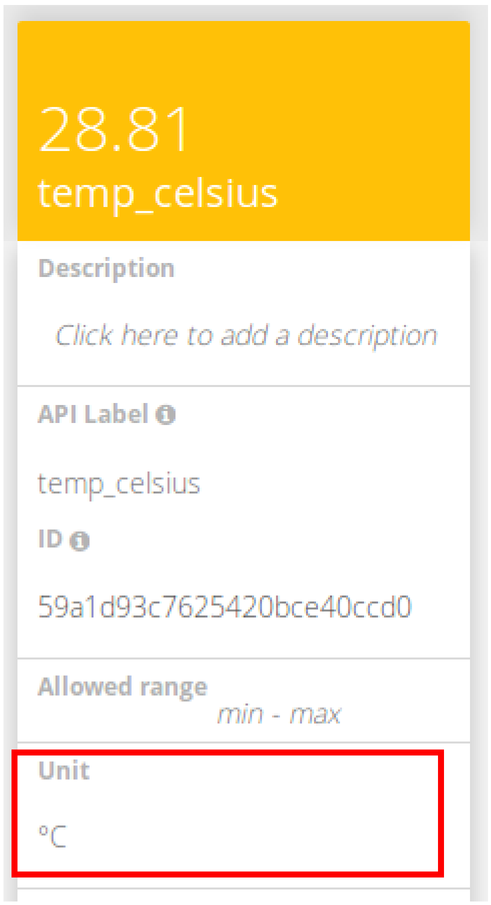
a few seconds ago

temp_fahrenheit

83.86

Last activity

a few seconds ago


28.81
temp_celsius

Description

Click here to add a description

API Label

temp_celsius

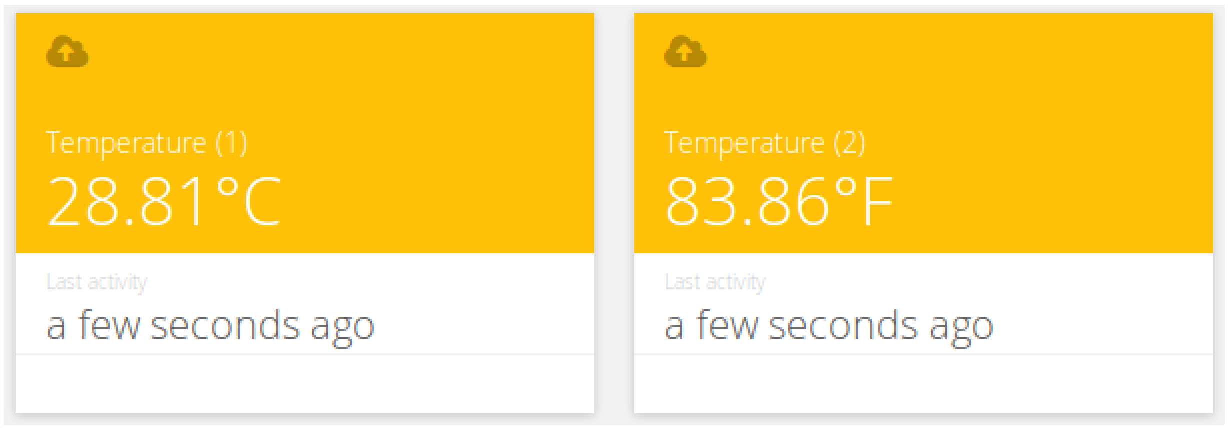
ID

59a1d93c7625420bce40ccd0

Allowed range
min - max

Unit

°C


Temperature (1)
28.81°C
Last activity
a few seconds ago

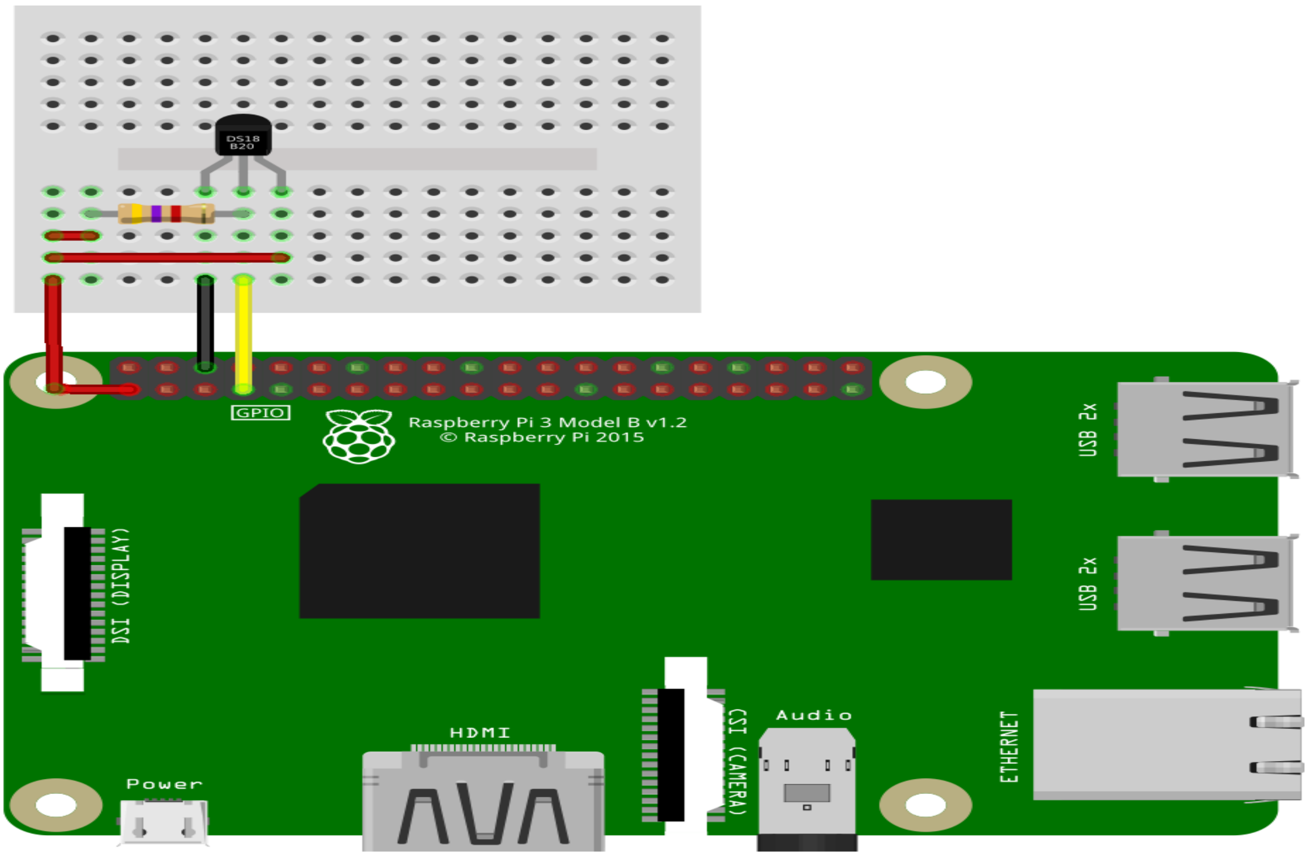
Temperature (2)
83.86°F
Last activity
a few seconds ago

HARDWARE SETUP

Wiring Setup

The resistor in this setup is used as a pull-up for the data-line, and should be connected between the data wire and the power wire. This ensures that the data line is at a defined logic level, and limits interference from electrical noise if our pin was left floating.
Using a 4.7kΩ (or 10kΩ) resistor and following the diagram below correct connections are done. Note that the pins connected in the Raspberry Pi are the same used in the table below:

1-Wire Temperature Sensor	Raspberry Pi 3
Black Wire	GND
Red Wire	3.3 V
Yellow Wire	GPIO4

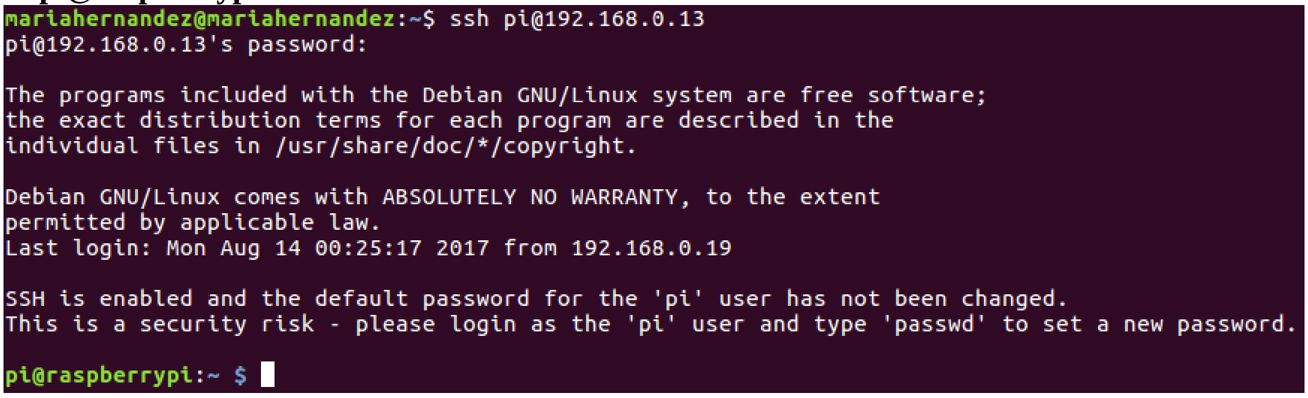


Sensor Setup

1. With your Raspberry Pi connected to the internet, verify the IP address assigned to the board access using ssh in your computer's terminal:
ssh pi@{IP_Address_assigned}
If you haven't already configured the credentials of your Raspberry Pi, note that you will have to use the default credentials provided:

- **User Name**: pi
- **Password**: raspberry

When your pi is configured and connected correctly, the user of your terminal becomes listed as:**pi@raspberrypi**



```
mariahernandez@mariahernandez:~$ ssh pi@192.168.0.13
pi@192.168.0.13's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Aug 14 00:25:17 2017 from 192.168.0.19

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@raspberrypi:~ $ 
```

2. Now let's upgrade some packages and install **pip,** Python's packet manager. Copy and paste the below commands into your terminal and press "enter" after each to run the commands.
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install python-pip python-dev build-essential

3. Then, install **Request** library, which is a popular Python library that simplifies making HTTP requests. Copy and paste the below commands into your terminal and press "enter" run the command.
$ pip install requests

4. The Raspberry Pi comes equipped with a range of drivers for interfacing. In this case, to be able to load the 1-Wire sensor's driver on the GPIO pins, we have to use these below two drivers. These drivers are therefore stored as loadable modules and the command ***modprobe*** is employed to boot them into the Linux kernel when required.
Run the commands below:
$ sudo modprobe w1-gpio
$ sudo modprobe w1-therm

5. Now, we need to change the directory to our 1-Wire device folder and list the devices in order to ensure that our sensor has loaded correctly. Copy and paste the below commands into your terminal and press "enter" after each to run the commands.
$ cd /sys/bus/w1/devices/
$ ls
At this moment you sensor has already been assembled and connected and should be listed as a series of numbers and letters. In our case, the device is registered as 28-00000830fa90 , but your case will be

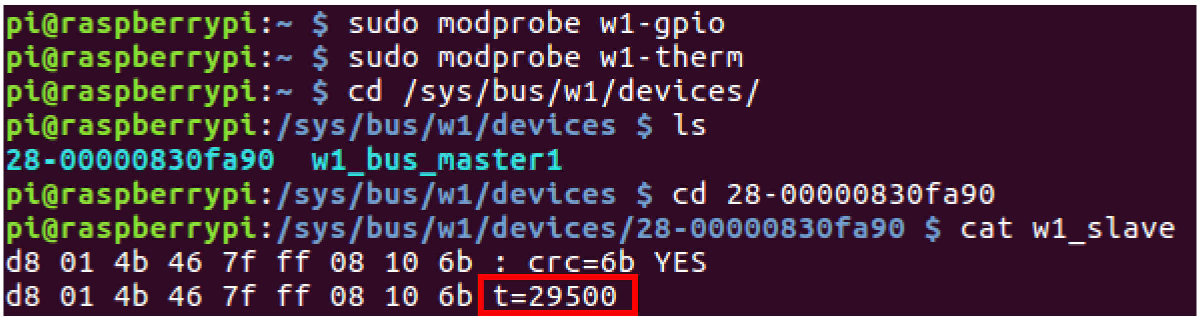
a different series of letters and numbers, so replace our serial number with your own and run the command.

$ cd 28-00000830fa90

The sensor periodically writes to the w1_slave file, to read your temp sensor, please run the command below:

$ cat w1_slave

This command will show you two lines of text with the output t= showing the temperature in degrees Celsius. Please note that a decimal point should be placed after the the first two digits (this is provided in the final code- do not worry); for example, the temperature reading we've received is 29.500 degrees Celsius.



```
pi@raspberrypi:~ $ sudo modprobe w1-gpio
pi@raspberrypi:~ $ sudo modprobe w1-therm
pi@raspberrypi:~ $ cd /sys/bus/w1/devices/
pi@raspberrypi:/sys/bus/w1/devices $ ls
28-00000830fa90  w1_bus_master1
pi@raspberrypi:/sys/bus/w1/devices $ cd 28-00000830fa90
pi@raspberrypi:/sys/bus/w1/devices/28-00000830fa90 $ cat w1_slave
d8 01 4b 46 7f ff 08 10 6b : crc=6b YES
d8 01 4b 46 7f ff 08 10 6b t=29500
```

Sending data to Ubidots for visualization

Create and run a Python script in your computer's terminal:

$ nano onewire_temp_ubidots.py

```
import os
import time
import requests

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

temp_sensor = '/sys/bus/w1/devices/28-000008be5880/w1_slave'

def temp_raw():
   f = open(temp_sensor, 'r')
   lines = f.readlines()
   f.close()
   return lines

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

while True:
        try:
r=requests.post('http://things.ubidots.com/api/v1.6/devices/temperature/?token=A1E-yUes0YXv0jZPP3TfMUVtnsL8WPOZU3', data=read_temp())
            print('Posting temperatures in Ubidots')
            print(read_temp())
        except:
            pass
        time.sleep(10)
```

Now run the above program using following command:
python onewire_temp_ubidots.py

Event Setup

An event (or alert) is any action triggered when data fulfills or exceeds a design rule. For example, an email or SMS message can be sent anytime a sensor stops sending data or a temperature exceeds a maximum or minimum threshold.
To create the event, please reference the article below:

- Events: Creating a Text Message Event (SMS, Email, and Telegram)

To create an Event:

- Select **Events** (from the Device Management dropdown).
- Select **Add Event**
- Select a device: **Machine 1** (the name of our device)
- Select a variable: **Power**
- If the value is greater than **34°C**, then send an **email**.

CONCLUSION

So we have developed a system to get the temperature of the room wirelessly using **‘Raspberry-Pi’** over internet. Also this system will not only receive the temperature but also be smart enough to read the temperature and process according to it. That is, if the temperature rises above **‘34°C’**, then it will send an email to a predefined email address which will notify the user about the risen temperature. Which makes the system able to read temperature of inaccessible areas remotely.