Raspberry PI 'How-To' Series

Zabbix Agent
User Parameters

Written by: Sopwith
Revision 1.0
April 4, 2019
sopwith@ismellsmoke.net

Introduction

Zabbix is a popular open-source platform used by IT professionals all over the world to monitor their security infrastructure. The list of companies that use the platform is impressive.

Figure 1: Zabbix dashboard

In a recent 'How-To', I provided detailed information to get a Zabbix agent up and running on a Raspberry Pi. In this document, I will show how to customize a Zabbix agent so you can place your Pi sensor data on a Zabbix dashboard.

Specifically, we will create five (5) Zabbix dashboard Widgets.

1. A Widget to display the CPU temperature of a Raspberry Pi
2. A Widget to display the GPU temperature of a Raspberry Pi
3. A Widget to display the relative humidity from an AM2315 sensor
4. A Widget to display the temperature in C/F from an AM2315 sensor
5. A Widget to display the calculated dew-point from AM2315 sensor data

To complete the steps needed to create these custom Widgets, you must complete the steps in the previous two 'How-To' documents, 'Zabbix Server on a Raspberry PI', and 'Zabbix Agent on a Raspberry Pi'. Once you have completed the above steps, let's move on to customizing a Zabbix agent.

Step-by-Step

There are 4 steps to creating custom Zabbix Widgets.

1. Create a shell or Python script to gather the data you want to display on the Zabbix dashboard.
2. Modify the Zabbix agent configuration file so it is aware of the scripts to run.
3. Configure the Zabbix server for the data items.
4. Add widgets to the dashboard.

Step-1 - Creating a shell or Python script to collect sensor data

Thanks to the flexibility of the Zabbix, platform, you can program the agent to perform nearly any task. In our case, we need to write scripts to collect the data we want on our Zabbix dashboard. Then we tell the Zabbix agent to run those scripts. I have created four bash shell scripts and one Python script.

The four bash scripts collect a Pi CPU and GPU temperatures. Two collect the CPU temperature in C and F degrees, and the other two collect the GPU temperature in C and F degrees. To access the temperatures you have to use different methods.

The Pi CPU temperature is available by reading a value named '*temp*' from the path: */sys/class/thermal/thermal_zone0/.*

```
pi@am2315:~ $ cat /sys/class/thermal/thermal_zone0/temp
45622
pi@am2315:~ $
```

Figure 2: CPU temperature value

In Figure 2 above if you cat the '*temp*' variable, you will see it holds a large value. To get the Celsius value, you simply divide the number by 1,000.

Figure 3 below lists the simple bash script I wrote to print out the Pi CPU temperature in C.

```
pi@am2315:/usr/local/bin $ cat cpu_temp_c.sh
t=$(cat /sys/class/thermal/thermal_zone0/temp | awk -F "[\n]" '{print $1}')
c=$(awk -v tmp=$t 'BEGIN { printf("%.2f\n", tmp/1000) }')
echo $c
```

Figure 3: CPU temperature bash script - Celsius

I also wrote an enhanced version of this script to convert the Celsius value to Fahrenheit. This is shown if Figure 4.

```
pi@am2315:/usr/local/bin $ cat cpu_temp_f.sh
t=$(cat /sys/class/thermal/thermal_zone0/temp | awk -F "[\n]" '{print $1}')
f=$(awk -v tmp=$t 'BEGIN { printf("%.2f\n", (((tmp/1000)*9)/5)+32) }')
echo $f
```

Figure 4: CPU temperature script - Fahrenheit

The output of the Celsius script is shown in Figure 5.

```
pi@am2315:/usr/local/bin $ cpu_temp_c.sh
45.62
```

Figure 5: CPU script output - Celsius

To obtain the temperature of the Pi GPU you have to take a different approach. There is a binary program that provides this information named *vcgencmd*. It is located in the */opt/vc/bin* folder. Figure 6 shows the output when the program is executed.

```
pi@am2315:/usr/local/bin $ vcgencmd measure_temp
temp=45.1'C
pi@am2315:/usr/local/bin $
```

Figure 6: vcgencmd output

The bash script to collect the GPU temperature in Celsius is shown below in Figure 7.

```
pi@am2315:/usr/local/bin $ cat gpu_temp_c.sh
t=$(/opt/vc/bin/vcgencmd measure_temp | awk -F "[=:']" '{print $2}')
c=$(awk -v tmp=$t 'BEGIN { printf("%.2f\n", tmp) }')
echo $c
```

Figure 7: GPU temperature script - Celsius

The output of the script is shown in Figure 8.

```
pi@am2315:/usr/local/bin $ ./gpu_temp_c.sh
45.10
```
Figure 8: GPU script output - Celsius

I also wrote a script that converts the GPU temperature value to Fahrenheit (gpu_temp_f.sh).

NOTE: In order for you to execute the *vcgencmd,* you must be a member of the video group. To add the Pi user to the video group, execute the below command:

$ *sudo usermod -aG video pi*

If you are not a member of the video group, the *vcgencmd* will silently fail and return no results. Trust me when I say this – I lost several hours trying to figure out why my script did not work.

In order to collect temperature, humidity, and dew-point calculation data from a high-precision AM2315 temperature sensor, I wrote a Python script named *zabbix_AM2315.py.* The script has a help menu shown below in Figure 9.

```
pi@am2315:/usr/local/bin $ zabbix_AM2315.py -h
usage: zabbix_AM2315.py [-h] [-f] [-d | -r | -t]

optional arguments:
  -h, --help        show this help message and exit
  -f, --fahrenheit  temp in fahrenheit
  -d, --dewpoint    return dewpoint
  -r, --humidity    return humidity
  -t, --temp        return temperature
```
Figure 9: zabbix_AM2315.py help screen

When the script is run without any parameters it prints out all the values (Figure 10).

```
pi@am2315:/usr/local/bin $ zabbix_AM2315.py
Temp C: 20.3
Temp F: 68.54
Humidity: 51.6
Dewpoint C: 10.0 - Dewpoint_F: 50.0
```
Figure 10: zabbix_AM2315.py default output

NOTE: To learn how to get an AM2315 temperature sensor working on a Raspberry PI, see this document.

At this point, we now have the five scripts that collect the data we need to put on our Zabbix dashboard:

1. Pi CPU temperature in Celsius – cpu_temp_c.sh
2. Pi CPU temperature in Fahrenheit – cpu_temp_f.sh
3. Pi GPU temperature in Celsius – gpu_temp_c.sh
4. Pi GPU temperature in Fahrenheit – gpu_temp_f.sh
5. AM2315 Temperature sensor data – zabbix_AM2315.py

The scripts are placed in the folder */usr/local/bin*. Figure 11 below show a listing of the scripts in that folder:

```
pi@am2315:/usr/local/bin $ ls -l
total 24
-rwxr-xr-x 1 root zabbix  144 Mar  1 12:37 cpu_temp_c.sh
-rwxr-xr-x 1 root zabbix  156 Mar  1 12:40 cpu_temp_f.sh
-rwxr-xr-x 1 root zabbix  130 Mar  1 12:55 gpu_temp_c.sh
-rwxr-xr-x 1 root zabbix  142 Mar  1 12:41 gpu_temp_f.sh
-rwxr-xr-x 1 root zabbix 7131 Mar 22 21:00 zabbix_AM2315.py
```
Figure 11: Script listing

Also, note all of the scripts print their output to *stdout*. This provides the data to the Zabbix agent when the relevant data is requested by the Zabbix server. The agent simply sends the output of the user parameter script back to the Zabbix server. The server then places the data on the appropriate dashboard.

Step-2 – Modify the Zabbix agent configuration file so it is aware of the scripts to run

Now that the scripts are working correctly and placed in folder */usr/local/bin,* the next step it to tell the Zabbix agent about them. To do so, we have to modify the agent configuration file (*/etc/zabbix/zabbix_agent.conf*).

Open the file with nano or your favorite text editor:

$ sudo nano /et/zabbix/zabbix_agentd.conf

Scroll down to the bottom of the file and add the below lines:

```
# AM2315 data.
UserParameter=am2315.temp_c, /usr/local/bin/zabbix_AM2315.py -t
UserParameter=am2315.temp_f, /usr/local/bin/zabbix_AM2315.py -f
UserParameter=am2315.humid, /usr/local/bin/zabbix_AM2315.py -r
UserParameter=am2315.dewpoint_c, /usr/local/bin/zabbix_AM2315.py -d
UserParameter=am2315.dewpoint_f, /usr/local/bin/zabbix_AM2315.py -d -f

# Pi CPU/GPU temperatures.
UserParameter=pi.cputemp_c, /usr/local/bin/cpu_temp_c.sh
UserParameter=pi.cputemp_f, /usr/local/bin/cpu_temp_f.sh
UserParameter=pi.gputemp_c, /usr/local/bin/gpu_temp_c.sh
UserParameter=pi.gputemp_f, /usr/local/bin/gpu_temp_f.sh
```

A Zabbix agent *UserParameter* contains two values: a parameter key, and the command to execute. The parameter key must be unique and follow the equal sign with no spaces. Following the key is a comma, space, and the name of the command to run.

The AM2315 temperature sensor data commands all use the same script (*zabbix_AM2315.py*), but vary in the command line parameter(s) passed to the script.

Figure 12 below shows the user parameter contents of my agent configuration file.

```
# 2019-02-16 Sopwith - Added UserParameters for AM2315.
# AM2315 data.
UserParameter=am2315.temp_c, /usr/local/bin/zabbix_AM2315.py -t
UserParameter=am2315.temp_f, /usr/local/bin/zabbix_AM2315.py -f
UserParameter=am2315.humid, /usr/local/bin/zabbix_AM2315.py -r
UserParameter=am2315.dewpoint_c, /usr/local/bin/zabbix_AM2315.py -d
UserParameter=am2315.dewpoint_f, /usr/local/bin/zabbix_AM2315.py -d -f

# Pi CPU/GPU temperatures.
UserParameter=pi.cputemp_c, /usr/local/bin/cpu_temp_c.sh
UserParameter=pi.cputemp_f, /usr/local/bin/cpu_temp_f.sh
UserParameter=pi.gputemp_c, /usr/local/bin/gpu_temp_c.sh
UserParameter=pi.gputemp_f, /usr/local/bin/gpu_temp_f.sh
```

Figure 12: Zabbix agent user parameters

When done with the required changes to the agent configuration file, restart the agent daemon with the below command (Figure 13):

$ *sudo service zabbix-agent restart*

```
pi@am2315:/etc/zabbix $ sudo service zabbix-agent restart
pi@am2315:/etc/zabbix $
```

Figure 13: Zabbix agent restart

Step-3 – Configure the Zabbix server for the data items

Now that the agent running on your Pi is properly configured, it is time to configure the Zabbix server. The first task it to tell the server about the new user-parameters you just created. Login to your Zabbix server and click on *Configuration* | *Hosts* and select your Pi from the list of hosts (Figure 14).

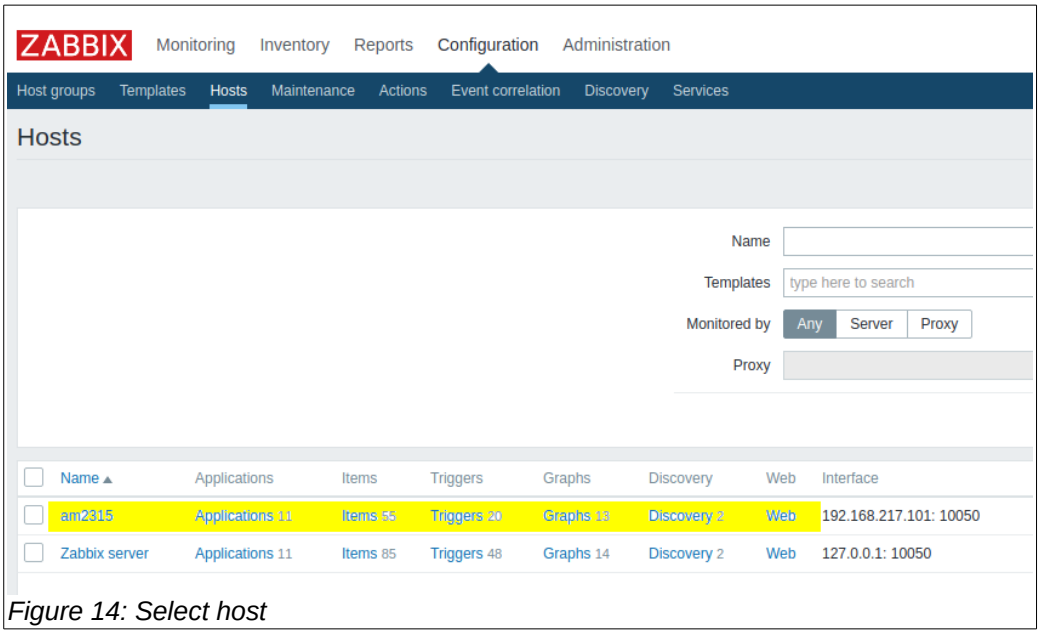

Figure 14: Select host

When you double-click on your host select '*Items'.* This will provide you list of all the items associated with your host. You can see the items I created for the AM2315 temperature sensor and Pi temperature items in Figure 15 and Figure 16.

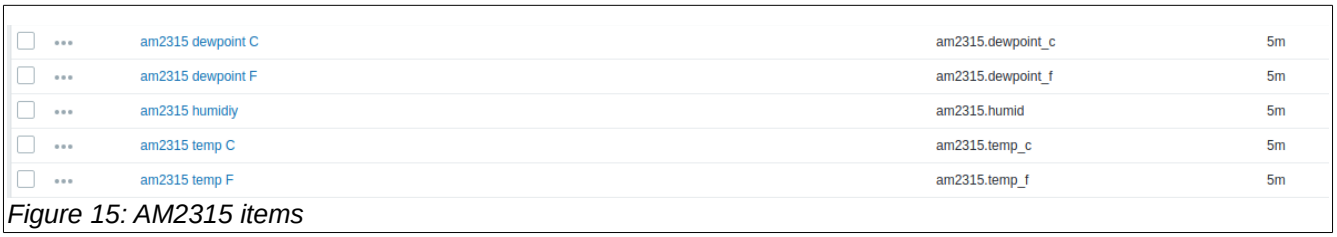

Figure 15: AM2315 items

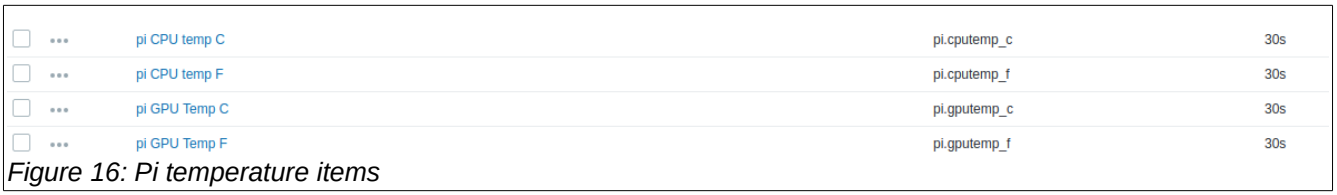

Figure 16: Pi temperature items

Configuring an item is easy. In the upper right corner of the screen click on '*Create Item'.* Next, fill out the required fields identified with a red asterisk (*).

Name: This field contains a description of your item.
Key: This field must match the name of the key in your Zabbix agent configuration file on the host.
Host interface: This contains the IP address of you host and the listening port (Default value is fine).
Type of information: This specifies the data type. All data types in this example are Numeric (float).
Update interval: This is the dashboard update interval (e.g how often to poll the data from the host).
History storage period: How long you want to store the raw data.
Trend storage period: How long you want to store the trend data.

Figure 17 shows the setting for my '*Pi CPU temp C*' item.

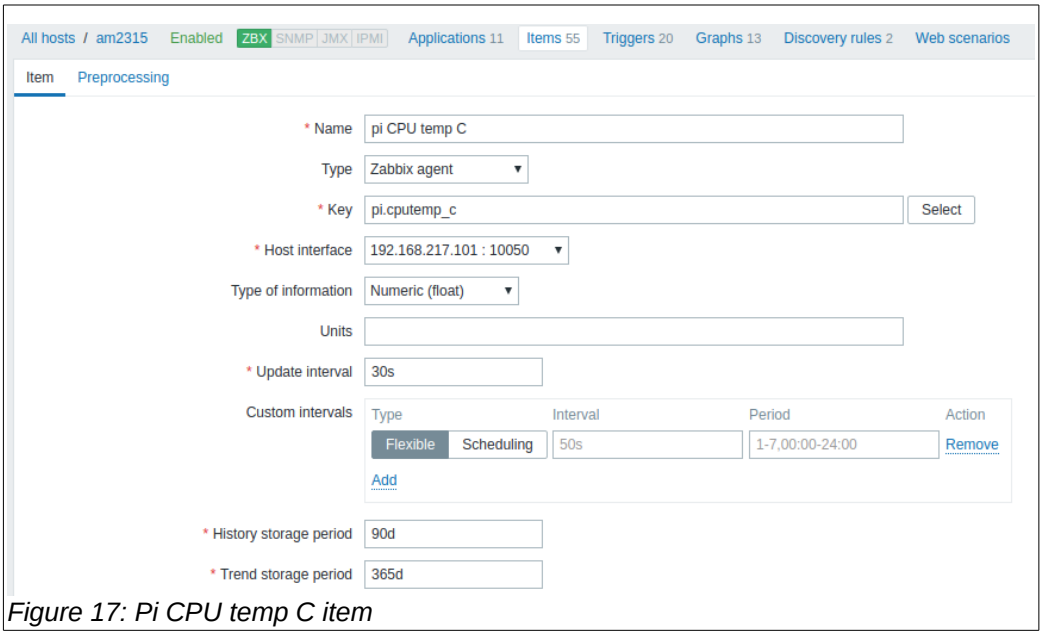

Figure 17: Pi CPU temp C item

NOTE: Be sure the '*Enabled*' checkbox is checked at the bottom of this panel.

You will need to create the additional items for the other datum in the same manner. Be sure that the '*Key*' and '*Host Interface*' items are correct for each item.

Step-4 – Add Widgets to the Zabbix dashboard

Now that the Zabbix server knows about the data items, all we need to do is plot them on a dashboard. To add a Widget, click on *Monitoring* | *Dashboard* | + *Add Widget.* Click *Graph* in the drop-down menu.

Figure 18 shows the settings for the Widget that tracks my Pi CPU temperature. Notice the values I have set for this Widget.

Type: Graph
Name: Pi CPU Temp
Refresh interval: 30 seconds (Your choice)
Data Set: Select your host from the list
Item: Select your item from the list (pi CPU temp C)
Base Colour: Your choice
Missing Data: Connected
Y-axis: Left is for Celsius.

Figure 18: Pi CPU temp C data set

I added a second data set to this Widget. This allows me to show the Celsius temperature on the left axis and the Fahrenheit temperature on the right axis (Figure 19).

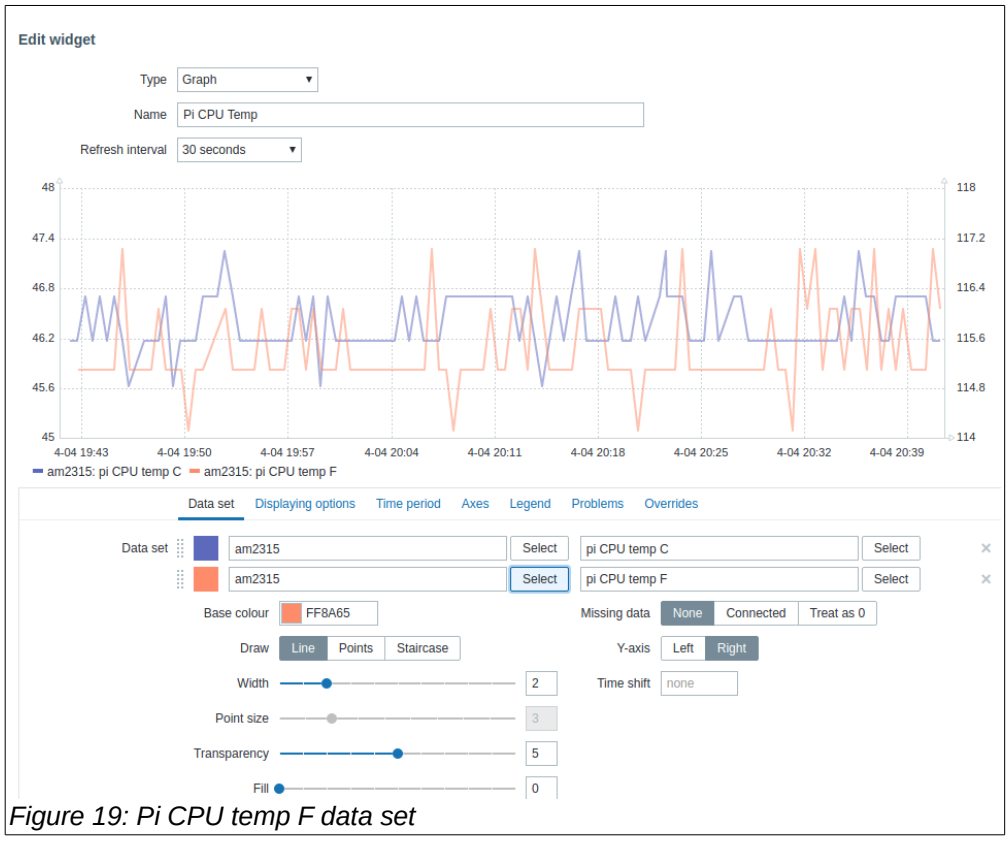

Figure 19: Pi CPU temp F data set

When the Widget is created, it will begin to poll the required data from the Pi host at the requested interval. Figure 20 below shows my Pi CPU temperature over the last 12 hours.

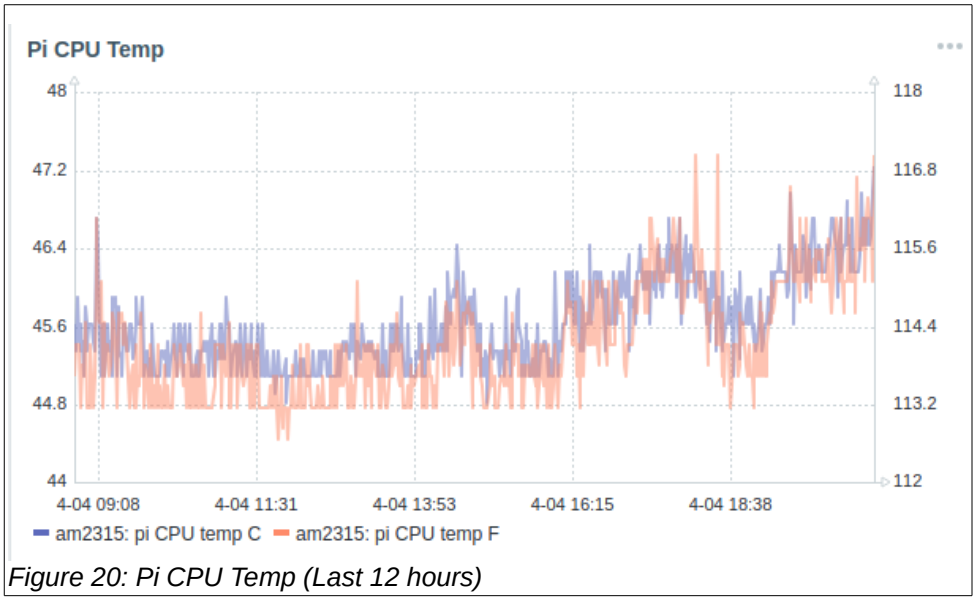

Figure 20: Pi CPU Temp (Last 12 hours)

You will need to create the additional Widgets in the same way to populate your dashboard. The process is exactly the same. Add a Widget to your dashboard and select the host and data item you want to display.

Below (Figure 21) shows my dashboard data for the Pi CPU/GPU temperatures and the AM2315 temperature sensor for the last week.

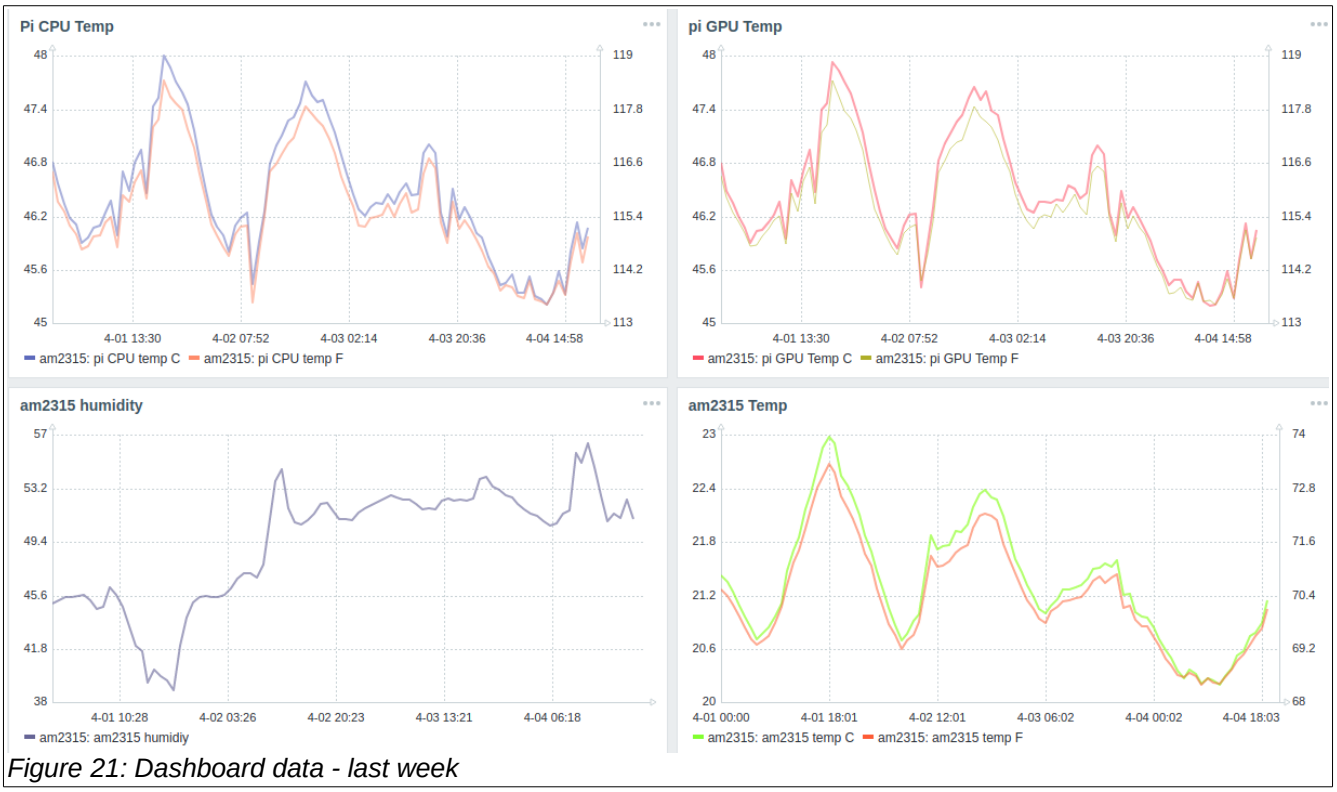

Figure 21: Dashboard data - last week

Congratulations! You now know how to add custom user-data parameters to your Zabbix server to track any kind of data from your Raspberry Pi's.

Summary

This 'How-To' document explains the steps required to plot Raspberry Pi temperature and sensor data on a Zabbix dashboard. The amazing flexibility of the Zabbix platform allows Makers to do all kinds of cool things with sensor data.

Having the ability to monitor a Pi's CPU/GPU temperature is really useful for those of you that overclock or stretch the limits of a Pi's performance capabilities.

Send corrections, comments, complaints, ideas, or any other feedback to: *sopwith@ismellsmoke.net*.