

Monitoring and alertings | Raspberry Pi Power States

This is based on <https://harlemsquirrel.github.io/shell/2019/01/05/monitoring-raspberry-pi-power-and-thermal-issues.html>

Make vcgencmd available

```
apt install libraspberrypi-bin
```

/opt/collectd_plugins/pwr_states.py

```
...  
Returns the throttled state of the system. This is a bit pattern - a bit being set indicates  
the following meanings:  
0x50000 = 0101 0000 0000 0000 0000  
Adding the bit numbers along the top we get:  
  
19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0  
 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
From this we can see that bits 18 and 16 are set, indicating that the Pi has previously been  
throttled due to under-voltage, but is not currently throttled for any reason.  
...  
import collectd  
import subprocess  
  
def read_func():  
    undervoltage = 0  
    armfreq_capped = 0  
    throttled = 0  
    soft_temp_limit = 0  
  
    GET_THROTTLED_CMD = 'vcgencmd get_throttled'  
    MESSAGES = {  
        0: 'Under-voltage!',
```

```

1: 'ARM frequency capped!',
2: 'Currently throttled!',
3: 'Soft temperature limit active',
16: 'Under-voltage has occurred since last reboot.',
17: 'Throttling has occurred since last reboot.',
18: 'ARM frequency capped has occurred since last reboot.',
19: 'Soft temperature limit has occurred'
}

throttled_output = subprocess.check_output(GET_THROTTLED_CMD, shell=True)
throttled_binary = bin(int(throttled_output.decode('utf8').split('=')[1], 0))

# print general information
#for position, message in MESSAGES.items():
    # Check for the binary digits to be "on" for each warning message
    #if len(throttled_binary) > position and throttled_binary[0 - position - 1] == '1':
        #print(message)

# relevant for monitoring are only the current possible states, which are 0,1,2,3
(16,17,18,19 are past values)
for position, message in MESSAGES.items():
    if len(throttled_binary) > position and throttled_binary[0 - position - 1] == '1':
        if position == 0:
            undervoltage = 1
        if position == 1:
            armfreq_capped = 1
        if position == 2:
            throttled = 1
        if position == 3:
            soft_temp_limit = 1

    collectd.Values(plugin='pwr_states', type='gauge', type_instance='undervoltage',
values=[undervoltage]).dispatch()
    collectd.Values(plugin='pwr_states', type='gauge', type_instance='armfreq_capped',
values=[armfreq_capped]).dispatch()
    collectd.Values(plugin='pwr_states', type='gauge', type_instance='throttled',
values=[throttled]).dispatch()
    collectd.Values(plugin='pwr_states', type='gauge', type_instance='soft_temp_limit',
values=[soft_temp_limit]).dispatch()

```

```
# debug print
#collectd.info("undervoltage = %s" % undervoltage)
#collectd.info("armfreq_capped = %s" % armfreq_capped)
#collectd.info("throttled = %s" % throttled)
#collectd.info("soft_temp_limit = %s" % soft_temp_limit)

#print("undervoltage = ", undervoltage)
#print("armfreq_capped = ", armfreq_capped)
#print("throttled = ", throttled)
#print("soft_temp_limit = ", soft_temp_limit)

collectd.register_read(read_func,1) # read every 1 seconds
#read_func()
```

Configure collectd

```
<Plugin python>
  ModulePath "/opt/collectd_plugins"
  Import "pwr_states"
  <Module pwr_states>
    </Module>
</Plugin>
```

Restart collectd

```
service collectd restart
journalctl -f -u collectd.service
```

The following warning can be safely ignored

```
python plugin: Found a configuration for the "pwr_states" plugin, but the plugin isn't loaded
or didn't register a configuration callback.
```

Grafana query from InfluxDB

```
select last(*) from "pwr_states_value" WHERE "type_instance" = 'throttled' AND "host" =~  
/^$host$/ AND $timeFilter GROUP BY time($interval)  
select last(*) from "pwr_states_value" WHERE "type_instance" = 'armfreq_capped' AND "host" =~  
/^$host$/ AND $timeFilter GROUP BY time($interval)  
select last(*) from "pwr_states_value" WHERE "type_instance" = 'undervoltage' AND "host" =~  
/^$host$/ AND $timeFilter GROUP BY time($interval)  
select last(*) from "pwr_states_value" WHERE "type_instance" = 'soft_temp_limit' AND "host" =~  
/^$host$/ AND $timeFilter GROUP BY time($interval)
```

The power states are also monitored by [Repetier Server integrated monitoring](#)

Version #1

Erstellt: 2026-06-08 15:37:22 CEST von Mario Voigt

Zuletzt aktualisiert: 2026-06-08 15:38:38 CEST von Mario Voigt