

Hardwarepraktikum Internet-Technologien

Task 2: Network setup and temperature sensor in the Raspberry Pi



Julius-Maximilians-Universität Würzburg

Chair of Computer Science III

A project report submitted by **Group 11**

Enrique Mesonero Ronco

Manuel Calvo Martín

Pablo E. Ortega Ureña

Summer Term 2022

May, 2022

Contents

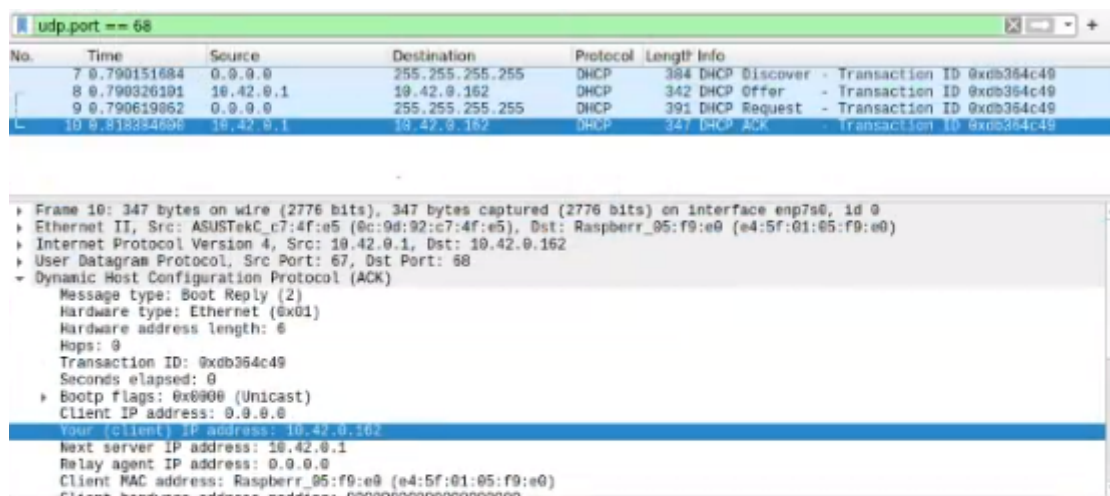
2. Network setup and temperature sensor in the Raspberry Pi	3
2.2. First connection: DHCP service on your end device	3
2.3. Assignment of static IP addresses	4
2.4. Temperature sensor setup	6

2. Network setup and temperature sensor in the Raspberry Pi

2.2. First connection: DHCP service on your end device

After connecting the Raspberry directly to the PC, we needed to discover its IP address. First, on our computer, we used the function to share the Internet connection, in order for the DHCP service to start. Our IP address now is 10.42.0.1.

We open Wireshark after this and check the Ethernet interface. There is a lot of traffic, but after filtering for UDP port 68, we see a DHCP Ack, sent from our IP address to another IP address, 10.42.0.162. This address is the Raspberry Pi one.

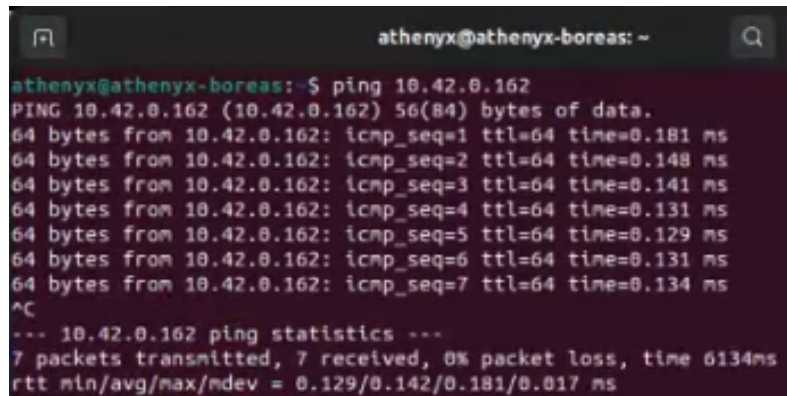


No.	Time	Source	Destination	Protocol	Length	Info
7	0.790151684	0.0.0.0	255.255.255.255	DHCP	384	DHCP Discover - Transaction ID 0xdb364c49
8	0.790326191	10.42.0.1	10.42.0.162	DHCP	342	DHCP Offer - Transaction ID 0xdb364c49
9	0.790619862	0.0.0.0	255.255.255.255	DHCP	391	DHCP Request - Transaction ID 0xdb364c49
10	0.818384696	10.42.0.1	10.42.0.162	DHCP	347	DHCP Ack - Transaction ID 0xdb364c49

Frame 10: 347 bytes on wire (2776 bits), 347 bytes captured (2776 bits) on interface enp7s0, id 0
Ethernet II, Src: ASUSTekC_c7:4f:e5 (0c:9d:92:c7:4f:e5), Dst: Raspberr_05:f9:e0 (e4:5f:01:05:f9:e0)
Internet Protocol Version 4, Src: 10.42.0.1, Dst: 10.42.0.162
User Datagram Protocol, Src Port: 67, Dst Port: 68
Dynamic Host Configuration Protocol (ACK)
Message type: Boot Reply (2)
Hardware type: Ethernet (0x01)
Hardware address length: 6
Hops: 0
Transaction ID: 0xdb364c49
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0
Your (client) IP address: 10.42.0.162
Next server IP address: 10.42.0.1
Relay agent IP address: 0.0.0.0
Client MAC address: Raspberr_05:f9:e0 (e4:5f:01:05:f9:e0)
Client hardware address address: 0000000000000000

Figure 2.1: DHCP Messages on Wireshark

We check pingg this address, receiving a response.

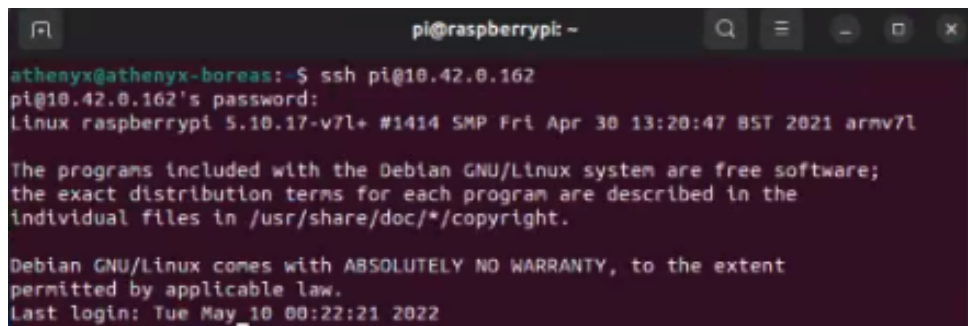


```
athenyx@athenyx-boreas: ~  
athenyx@athenyx-boreas:~$ ping 10.42.0.162  
PING 10.42.0.162 (10.42.0.162) 56(84) bytes of data.  
64 bytes from 10.42.0.162: icmp_seq=1 ttl=64 time=0.181 ms  
64 bytes from 10.42.0.162: icmp_seq=2 ttl=64 time=0.148 ms  
64 bytes from 10.42.0.162: icmp_seq=3 ttl=64 time=0.141 ms  
64 bytes from 10.42.0.162: icmp_seq=4 ttl=64 time=0.131 ms  
64 bytes from 10.42.0.162: icmp_seq=5 ttl=64 time=0.129 ms  
64 bytes from 10.42.0.162: icmp_seq=6 ttl=64 time=0.131 ms  
64 bytes from 10.42.0.162: icmp_seq=7 ttl=64 time=0.134 ms  
^C  
--- 10.42.0.162 ping statistics ---  
7 packets transmitted, 7 received, 0% packet loss, time 6134ms  
rtt min/avg/max/mdev = 0.129/0.142/0.181/0.017 ms
```

Figure 2.2: Pinging the Raspberry

2.3. Assignment of static IP addresses

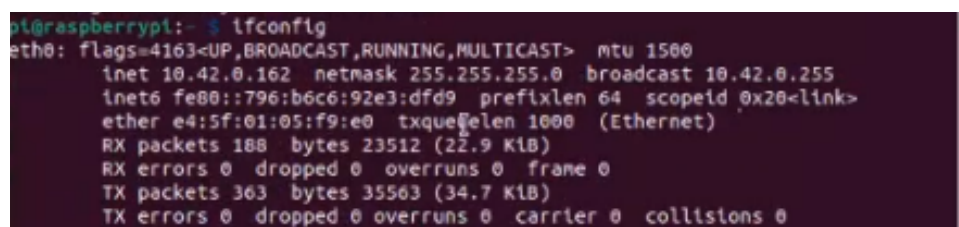
Since we want to use fixed IP addresses on our end devices instead of having dynamic ones via DHCP, we need to change the Raspberry configuration. To do so, we connect to the Raspberry using SSH.



```
pi@raspberrypi: ~  
athenyx@athenyx-boreas:~$ ssh pi@10.42.0.162  
pi@10.42.0.162's password:  
Linux raspberrypi 5.10.17-v7l+ #1414 SMP Fri Apr 30 13:20:47 BST 2021 armv7l  
  
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: Tue May 10 00:22:21 2022
```

Figure 2.3: Connecting to the Raspberry using SSH

After that, we run `ifconfig` to find out the name of our Ethernet interface. We see that the name of our Raspberry's Ethernet interface is `eth0` and its MAC address is `e4:5f:01:05:f9:e0`.



```
pi@raspberrypi:~$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 10.42.0.162 netmask 255.255.255.0 broadcast 10.42.0.255  
    inet6 fe80::796:b6c6:92e3:dfd9 prefixlen 64 scopeid 0x20<link>  
    ether e4:5f:01:05:f9:e0 txqueuelen 1000 (Ethernet)  
    RX packets 188 bytes 23512 (22.9 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 363 bytes 35563 (34.7 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figure 2.4: Checking the Ethernet interface of the Raspberry

In order to change the Raspberry IP address to one belonging to our group, we have to edit the `dhcpcd.conf` file, which manages the DHCP client daemon. We change the `eth0` interface, using the 10.11.1.1 address for the Raspberry, and setting the gateway under routers to the 10.11.1.2 address, which will later be our PC's IP address. After saving, we reboot the Raspberry to apply the changes, using `sudo reboot`.

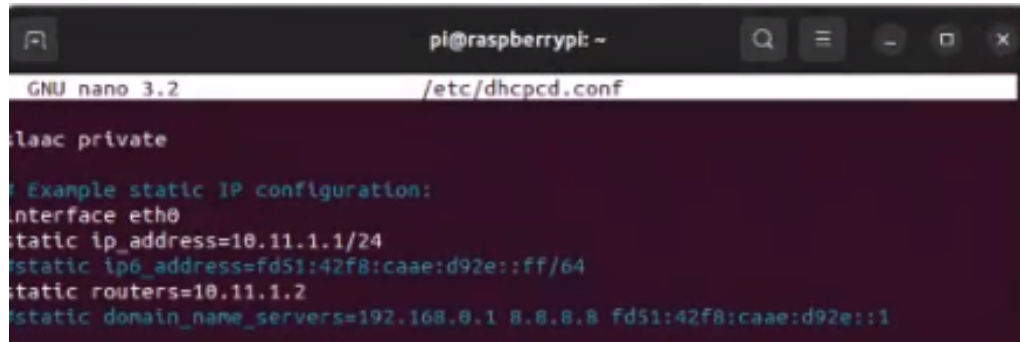


Figure 2.5: Editing the DHCP configuration file

Of course, after rebooting, we lose connection on our PC. To establish a connection again, we need to change the IP address of our PC to be in the same subnet as the Raspberry. We use the 10.11.1.2 address. We check if the Raspberry is accessible using ping, which it is, and then, we connect to it using SSH.

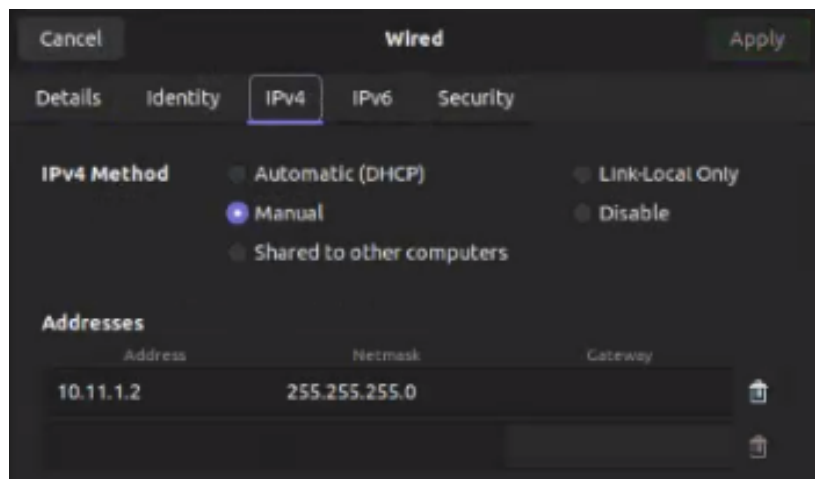


Figure 2.6: Changing the IP address of our PC


```
pi@raspberrypi: ~$ ping 10.11.1.1
64 bytes from 10.11.1.1: icmp_seq=7 ttl=64 time=0.135 ms
64 bytes from 10.11.1.1: icmp_seq=8 ttl=64 time=0.139 ms
^C
--- 10.11.1.1 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7150ms
rtt min/avg/max/mdev = 0.130/0.143/0.163/0.011 ms
athenyx@athenyx-boreas: ~$ ssh pi@10.11.1.1
The authenticity of host '10.11.1.1 (10.11.1.1)' can't be established.
ED25519 key fingerprint is SHA256:bTo3GQzJ9A3ZbdC29xuHkCcjsYMCh/qgan9rYklIbwI.
This host key is known by the following other names/addresses:
-/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.11.1.1' (ED25519) to the list of known hosts.
pi@10.11.1.1's password:
Linux raspberrypi 5.10.17-v7l+ #1414 SMP Fri Apr 30 13:20:47 BST 2021 armv7l

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 May 16 15:46:23 2022
pi@raspberrypi: ~$
```

Figure 2.7: Connecting to the Raspberry with static IP

2.4. Temperature sensor setup

Next up, we turned our attention to the breadboard. There, we had 3 resistors and a temperature sensor. We connected the temperature sensor to the Raspberry using one of the 3.3V pins, one of the ground pins and pin 4. In the breadboard, the voltage is connected to the 4.7 k Ω resistor and the voltage pin of the temperature sensor. The ground is connected to the ground pin of the temperature sensor and the 4th pin to the data pin of the sensor.

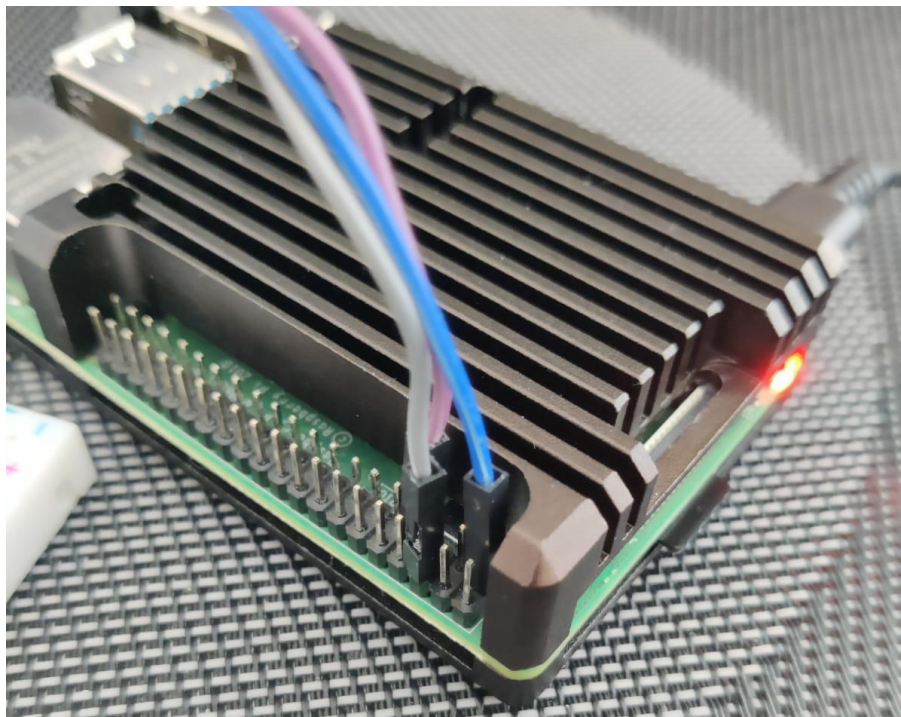


Figure 2.8: Physical connections on the Raspberry

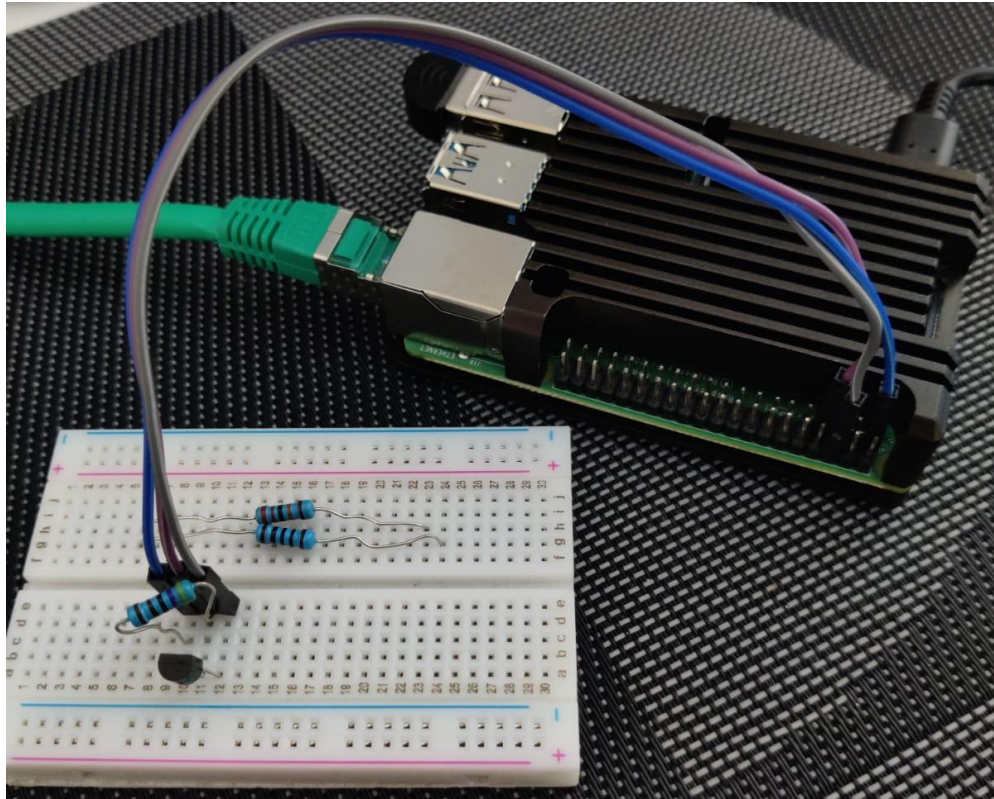


Figure 2.9: Full picture of the physical connections

Since to connect to the temperature data, we use the *One-Wire Interface* or W1-GPIO, we need to enable its kernel. For that, we add the `dtoverlay=w1-gpio` line to the `config.txt` file the Raspberry reads on boot. Now that the kernel is implemented after a restart, we need to load both `w1-gpio` and `w1-therm` modules.

```

pi@raspberrypi: ~
pi@raspberrypi:~$ nano /boot/config.txt
pi@raspberrypi:~$ nano /boot/config.txt
pi@raspberrypi:~$ sudo nano /boot/config.txt
pi@raspberrypi:~$ sudo reboot
Connection to 10.11.1.1 closed by remote host.
Connection to 10.11.1.1 closed.
athenyx@athenyx-boreas:~$ ssh pi@10.11.1.1
ssh: connect to host 10.11.1.1 port 22: No route to host
athenyx@athenyx-boreas:~$ ssh pi@10.11.1.1
pi@10.11.1.1's password:
Linux raspberrypi 5.10.17-v7l+ #1414 SMP Fri Apr 30 13:20:47 BST 2021 armv7l

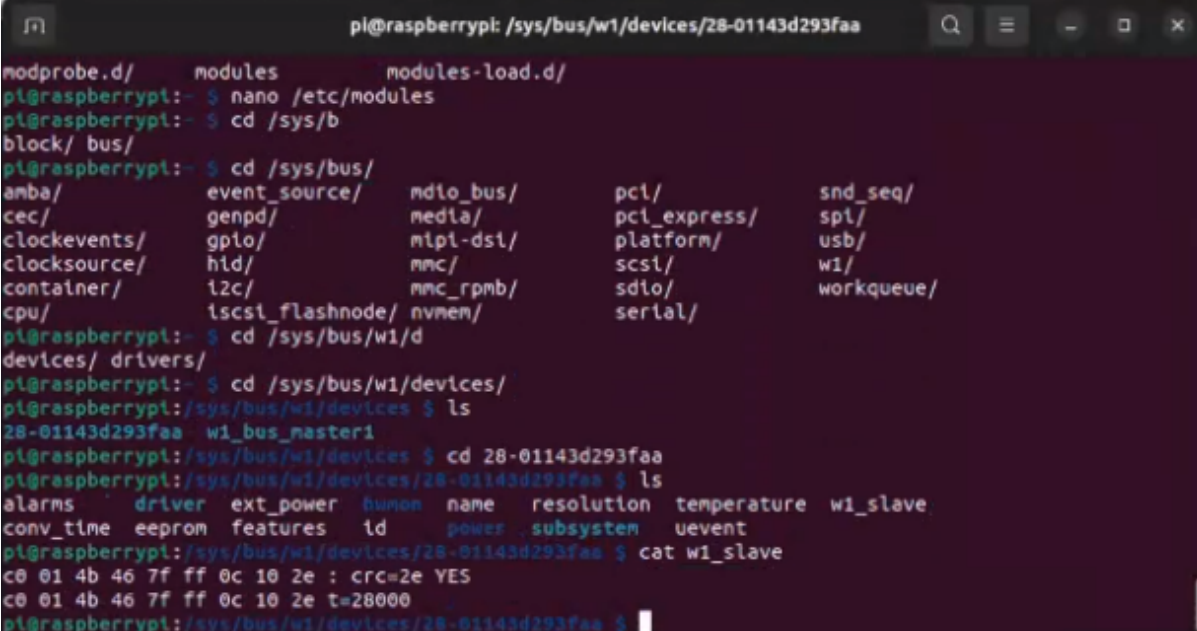
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon May 16 16:23:04 2022
pi@raspberrypi:~$ sudo mod
modeline2fb modinfo modprobe
pi@raspberrypi:~$ sudo modprobe w1-gpio
pi@raspberrypi:~$ sudo modprobe w1-therm
pi@raspberrypi:~$

```

Figure 2.10: One-Wire Interface kernel configuration

Finally, we check that the w1 folder appears under /sys/bus. Inside it, there is a folder called 28-01143d293faa, which represents our temperature sensor. If we read the w1_slave file contained inside that folder, we can check that the temperature sensor was measuring 28.000°C.



```

pi@raspberrypi: /sys/bus/w1/devices/28-01143d293faa
modprobe.d/  modules  modules-load.d/
pi@raspberrypi:~$ nano /etc/modules
pi@raspberrypi:~$ cd /sys/b
block/ bus/
pi@raspberrypi:~$ cd /sys/bus/
amba/      event_source/  mdio_bus/      pci/           snd_seq/
cec/       genpd/         media/         pci_express/   spt/
clockevents/  gpio/         mipi-dsi/     platform/      usb/
clocksource/  hid/          mmc/          scsi/          w1/
container/    i2c/          mmc_rpmb/     sdio/          workqueue/
cpu/          iscsi_flashnode/  nvme/         serial/
pi@raspberrypi:~$ cd /sys/bus/w1/d
devices/ drivers/
pi@raspberrypi:~$ cd /sys/bus/w1/devices/
pi@raspberrypi:/sys/bus/w1/devices$ ls
28-01143d293faa  w1_bus_master1
pi@raspberrypi:/sys/bus/w1/devices$ cd 28-01143d293faa
pi@raspberrypi:/sys/bus/w1/devices/28-01143d293faa$ ls
alarms      driver  ext_power  humnon  name  resolution  temperature  w1_slave
conv_time  eepron  features  id      power  subsystem  uevent
pi@raspberrypi:/sys/bus/w1/devices/28-01143d293faa$ cat w1_slave
c0 01 4b 46 7f ff 0c 10 2e : crc=2e YES
c0 01 4b 46 7f ff 0c 10 2e t=28000
pi@raspberrypi:/sys/bus/w1/devices/28-01143d293faa$

```

Figure 2.11: Temperature sensor detection and output