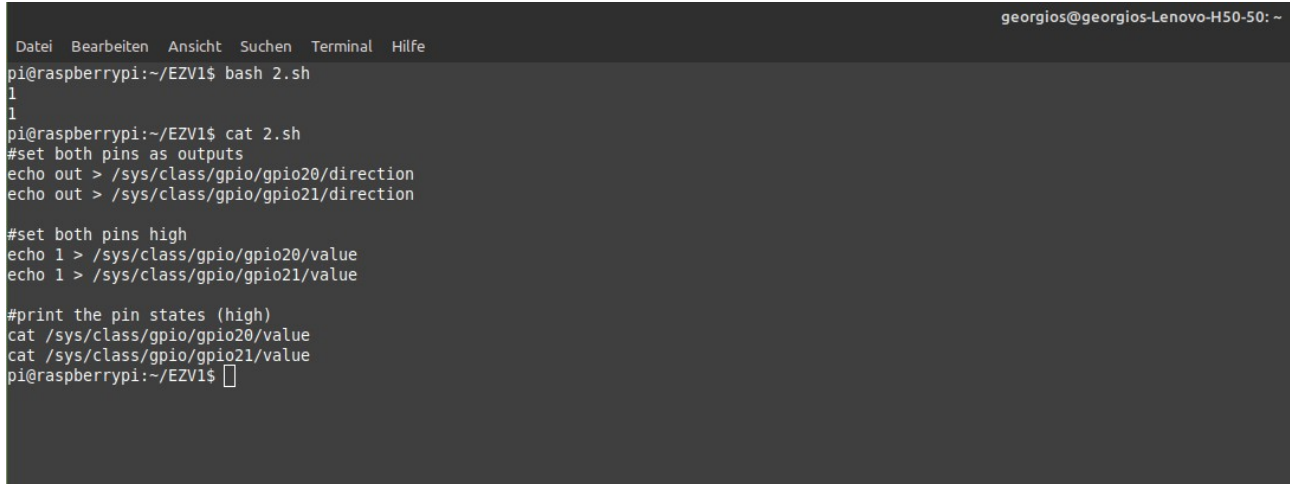


Echtzeitverarbeitung 1 Georgios Markou

1)

3)

A terminal window titled 'georgios@georgios-Lenovo-H50-50: ~' with a menu bar (Datei, Bearbeiten, Ansicht, Suchen, Terminal, Hilfe). The terminal shows the execution of a script '2.sh' on a Raspberry Pi. The script sets GPIO pins 20 and 21 as outputs, sets them to high, and prints their states. The output shows two '1's, indicating both pins are high. The user then runs 'cat 2.sh' to display the script's content.

```
georgios@georgios-Lenovo-H50-50: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
pi@raspberrypi:~/EZV1$ bash 2.sh
1
1
pi@raspberrypi:~/EZV1$ cat 2.sh
#set both pins as outputs
echo out > /sys/class/gpio/gpio20/direction
echo out > /sys/class/gpio/gpio21/direction

#set both pins high
echo 1 > /sys/class/gpio/gpio20/value
echo 1 > /sys/class/gpio/gpio21/value

#print the pin states (high)
cat /sys/class/gpio/gpio20/value
cat /sys/class/gpio/gpio21/value
pi@raspberrypi:~/EZV1$
```

Wie man erkennt habe ich mich eingeloggt und das Skript “2.sh“ mit dem Kommando “bash 2.sh“ ausgeführt. Das Skript gibt zwei mal “1“ aus was bedeutet, dass beide Pin „HIGH“ sind. Der nächste Befehl “cat 2.sh“ zeigt mein Skript.

4)

```
Datei  Bearbeiten  Ansicht  Suchen  Terminal  Hilfe
pi@raspberrypi:~/EZV1$ bash 3.sh
1
1
press the buttons!
0
0
pi@raspberrypi:~/EZV1$ cat 3.sh
#set both pins as outputs
echo in > /sys/class/gpio/gpio22/direction
echo in > /sys/class/gpio/gpio27/direction

#invert the logic of both pins
echo 0 > /sys/class/gpio/gpio22/active_low
echo 0 > /sys/class/gpio/gpio27/active_low

#print the states
cat /sys/class/gpio/gpio22/value
cat /sys/class/gpio/gpio27/value

echo "press the buttons!"

#wait for user to press button
sleep 5

#print the states again
cat /sys/class/gpio/gpio22/value
cat /sys/class/gpio/gpio27/value
pi@raspberrypi:~/EZV1$
```

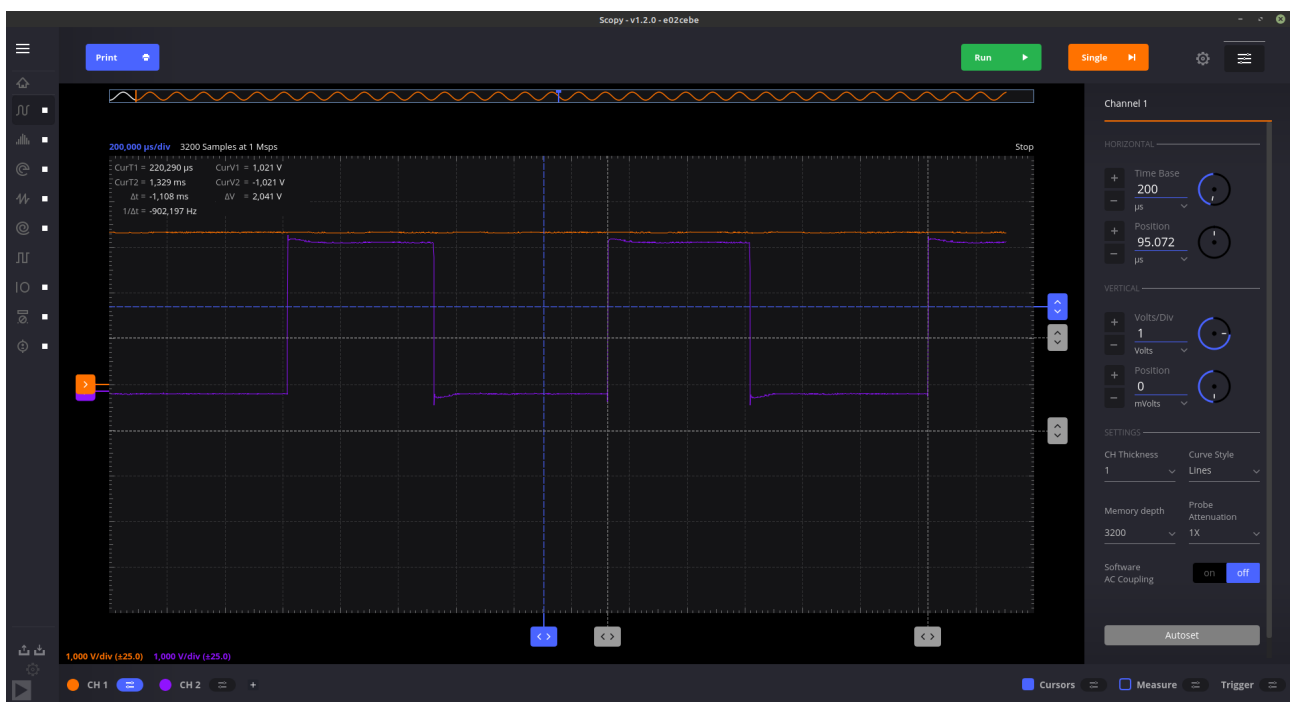
Wie man erkennt habe ich das Skript "3.sh" ausgeführt. Es zeigt den Zustand der beiden Taster an (1 = nicht-gedrückt). Man wird dann aufgefordert die beiden Taster zu drücken und sieht wenn sie gedrückt werden ist die Ausgabe "low" (0 = Taster gedrückt). Mit dem nächsten Befehl zeige ich den Inhalt meines Skripts.

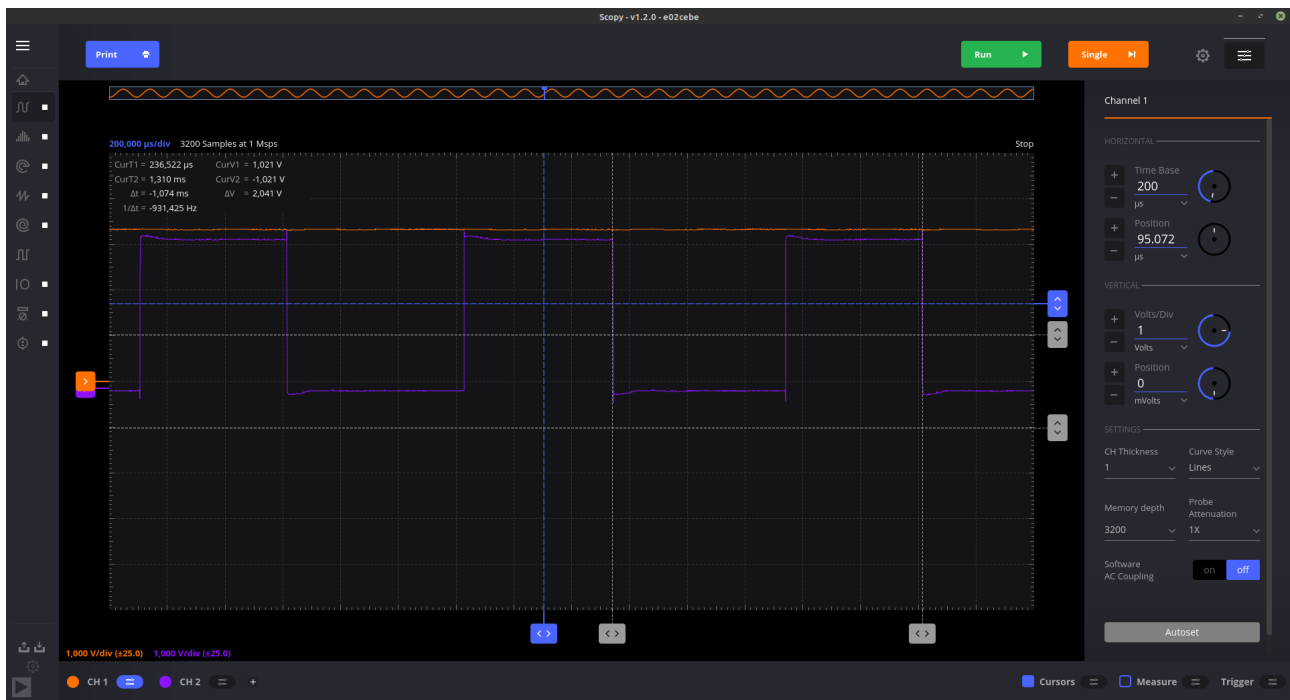
5)

```
Datei  Bearbeiten  Ansicht  Suchen  Terminal  Hilfe
pi@raspberrypi:~/EZV1$ bash 4.sh
^C
pi@raspberrypi:~/EZV1$ cat 4.sh
#set pin as outputs
echo out > /sys/class/gpio/gpio21/direction

while true
do
    echo 1 > /sys/class/gpio/gpio21/value
    echo 0 > /sys/class/gpio/gpio21/value
done
pi@raspberrypi:~/EZV1$
```

Das Vorgehen ist gleich wie in 3) und 4).





Diese 2 Schapschüsse von Scopy zeigen, dass die Frequenz schwankt (1. Bild ca. 902Hz; 2. Bild 931 Hz). Bei mir gab es selten Aussetzer (keine anderen Programme wurden ausgeführt). Die Aussetzer kommen vom Betriebssystem (Scheduler), weil mehrere Ressourcen gleichzeitig verwaltet werden müssen.