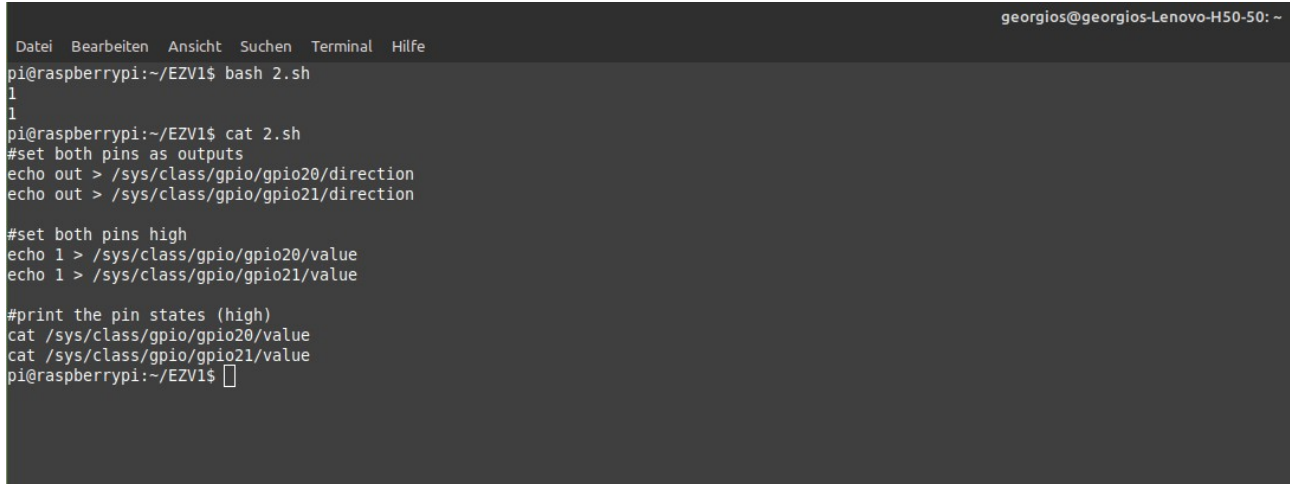


Echtzeitverarbeitung 1 Georgios Markou

1)

3)

A terminal window titled 'georgios@georgios-Lenovo-H50-50: ~' with a menu bar (Datei, Bearbeiten, Ansicht, Suchen, Terminal, Hilfe). The prompt is 'pi@raspberrypi:~/EZV1\$'. The user enters 'bash 2.sh', which outputs '1' on two lines. Then the user enters 'cat 2.sh', which displays the following script content:

```
#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
```

The terminal ends with the prompt 'pi@raspberrypi:~/EZV1\$' and a cursor.

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$
```

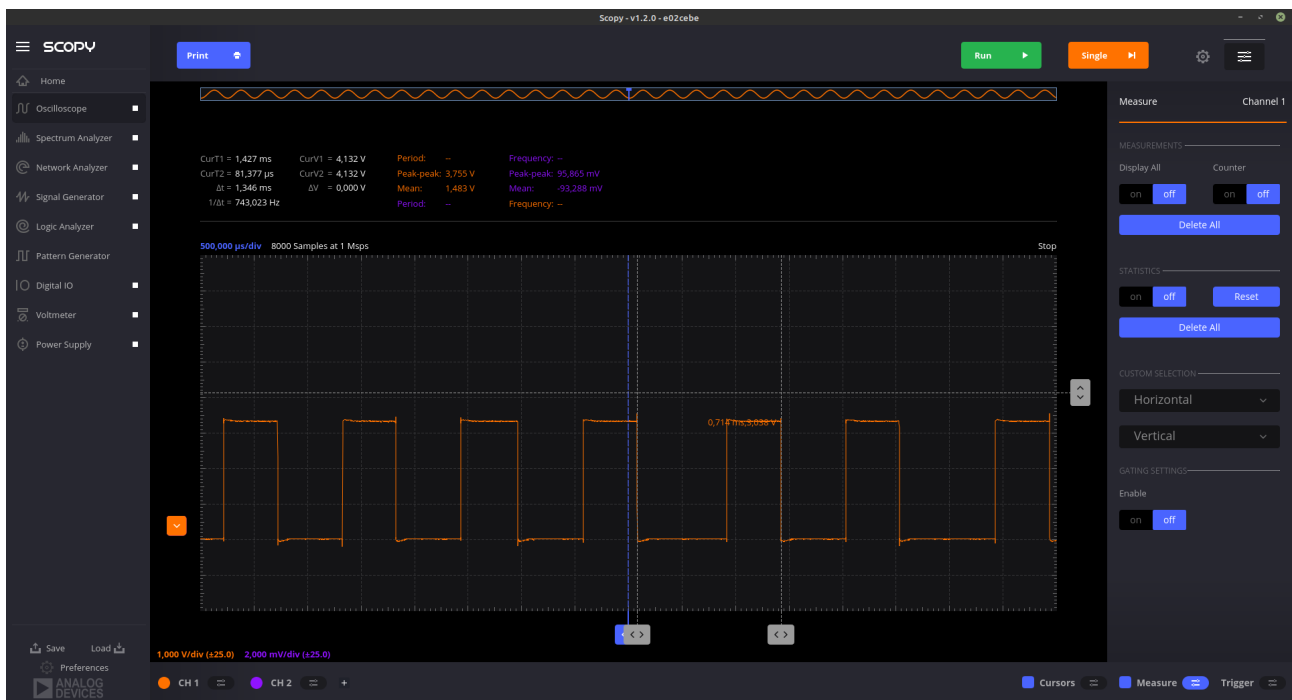
Wie man erkennt habe ich das Skript "3.sh" ausgeführt. Es zeigt den Zustand der beiden Taster an (1 = nicht-gedrückt). Man wird dann aufgefordert die beiden Taster zu drücken und sieht wenn sie gedrückt werden ist die Ausgabe "low" (0 = Taster gedrückt). Mit dem nächsten Befehl zeige ich den Inhalt meines Skripts. Die 2 Zeilen unter dem Kommentar sorgen dafür, dass die Pins "HIGH" sind wenn die Taster nicht gedrückt werden.

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

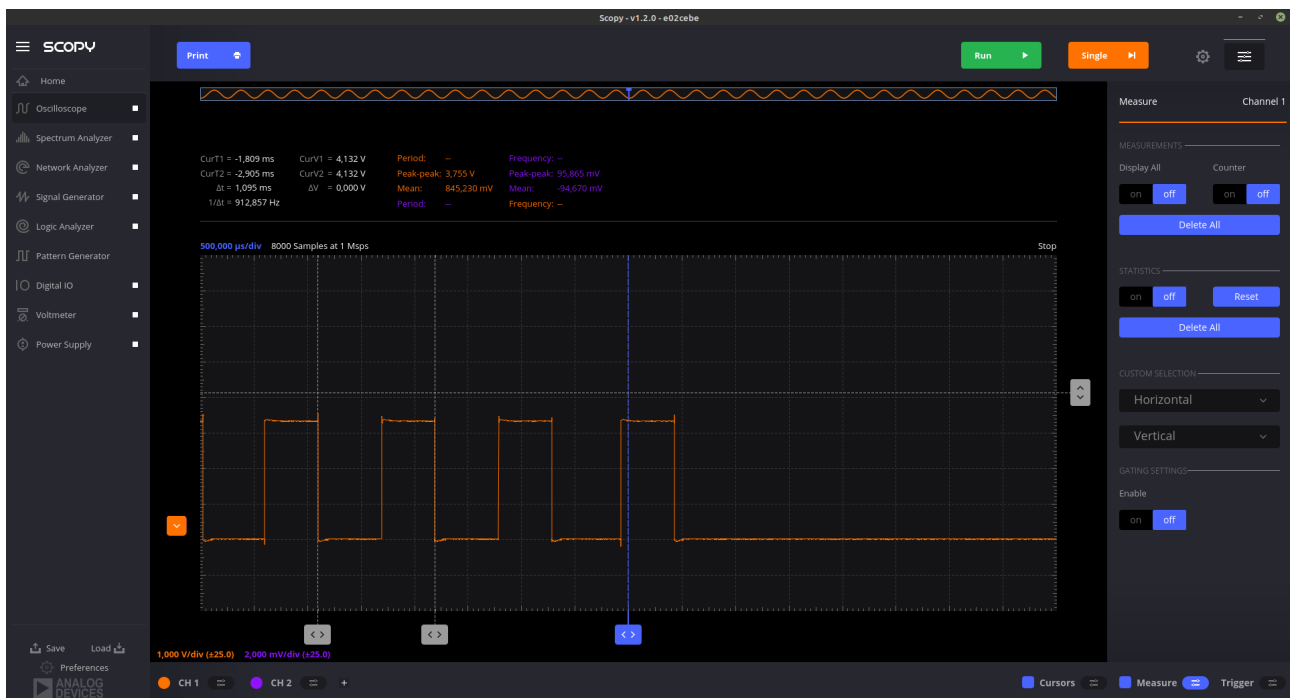
while true
do
    echo 1 > /sys/class/gpio/gpio21/value
    echo 0 > /sys/class/gpio/gpio21/value
done
pi@raspberrypi:~/EZV1$
```

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



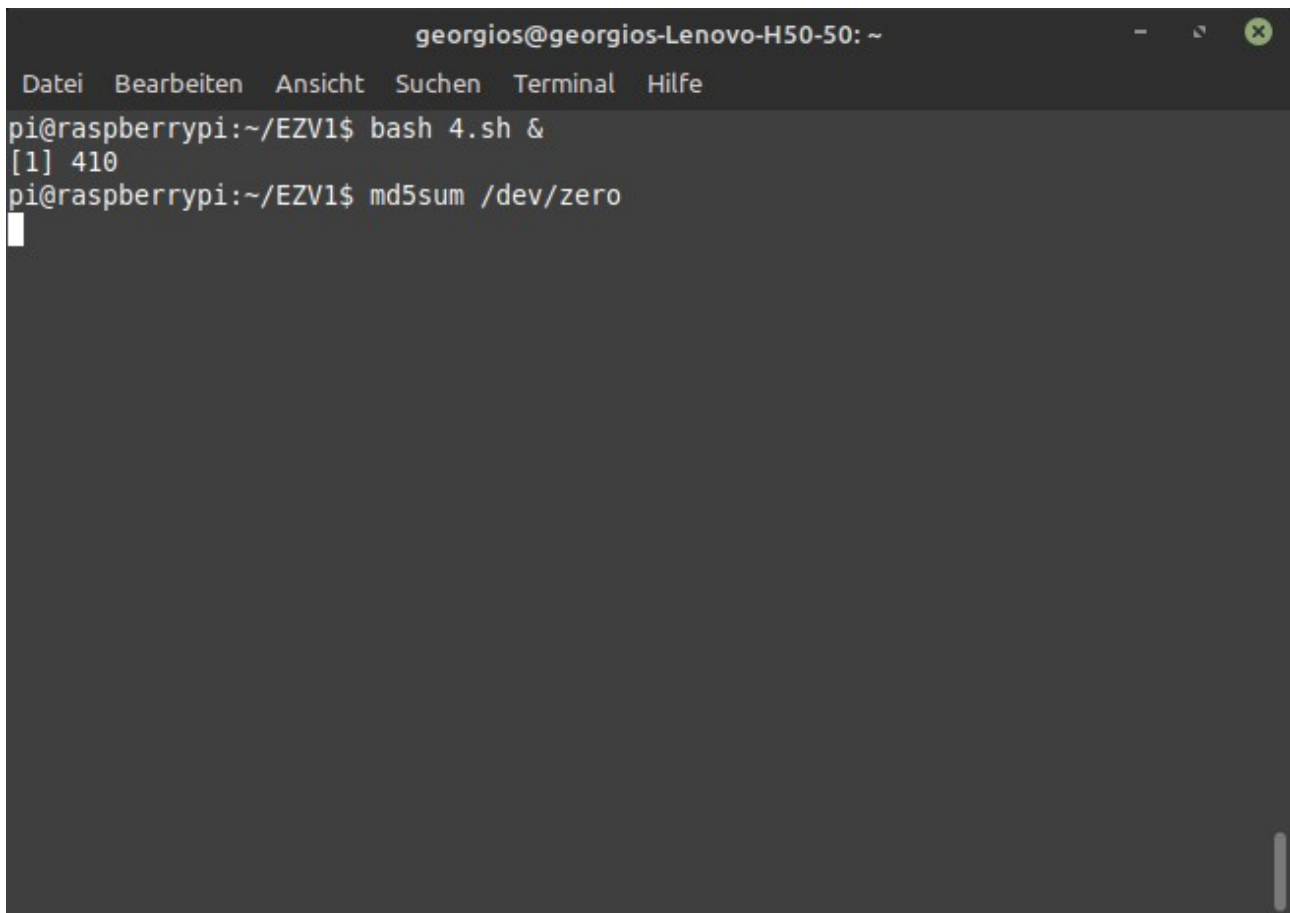


Diese 2 Schnapschüsse von Scopy zeigen, dass die Frequenz schwankt (1. Bild ca. 743 Hz; 2. Bild 934 Hz). Bei mir gab es selten Aussetzer (keine anderen Programme wurden ausgeführt). Die Aussetzer kommen vom Betriebssystem (Scheduler), weil mehrere Ressourcen gleichzeitig verwaltet werden müssen.



In diesem Schnapschuss sieht man, dass es Aussetzer gab. Die gemessene Frequenz war ca. 913 Hz jedoch sieht man auf der rechten Seite das es keine steigenden Flanken gibt. Das System hat im Hintergrund das Bash-Skript ausgeführt, während es im Vordergrund ein anderes Programm

ausgeführt hat. Man sieht, dass das Betriebssystem (Scheduler) Zeit braucht und eine gewisse Latenz erzeugt.

A terminal window titled 'georgios@georgios-Lenovo-H50-50: ~' with a menu bar containing 'Datei', 'Bearbeiten', 'Ansicht', 'Suchen', 'Terminal', and 'Hilfe'. The terminal shows the user 'pi' at 'raspberrypi' in the directory '~/EZV1'. The first command is 'bash 4.sh &', which returns '[1] 410'. The second command is 'md5sum /dev/zero', followed by a blank line.

```
georgios@georgios-Lenovo-H50-50: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
pi@raspberrypi:~/EZV1$ bash 4.sh &
[1] 410
pi@raspberrypi:~/EZV1$ md5sum /dev/zero

```

Mit dem ersten Befehl wurde ein neuer Prozess gestartet der “4.sh“ ausführt. Gleichzeitig wurde auch “md5sum /dev/zero“ ausgeführt.