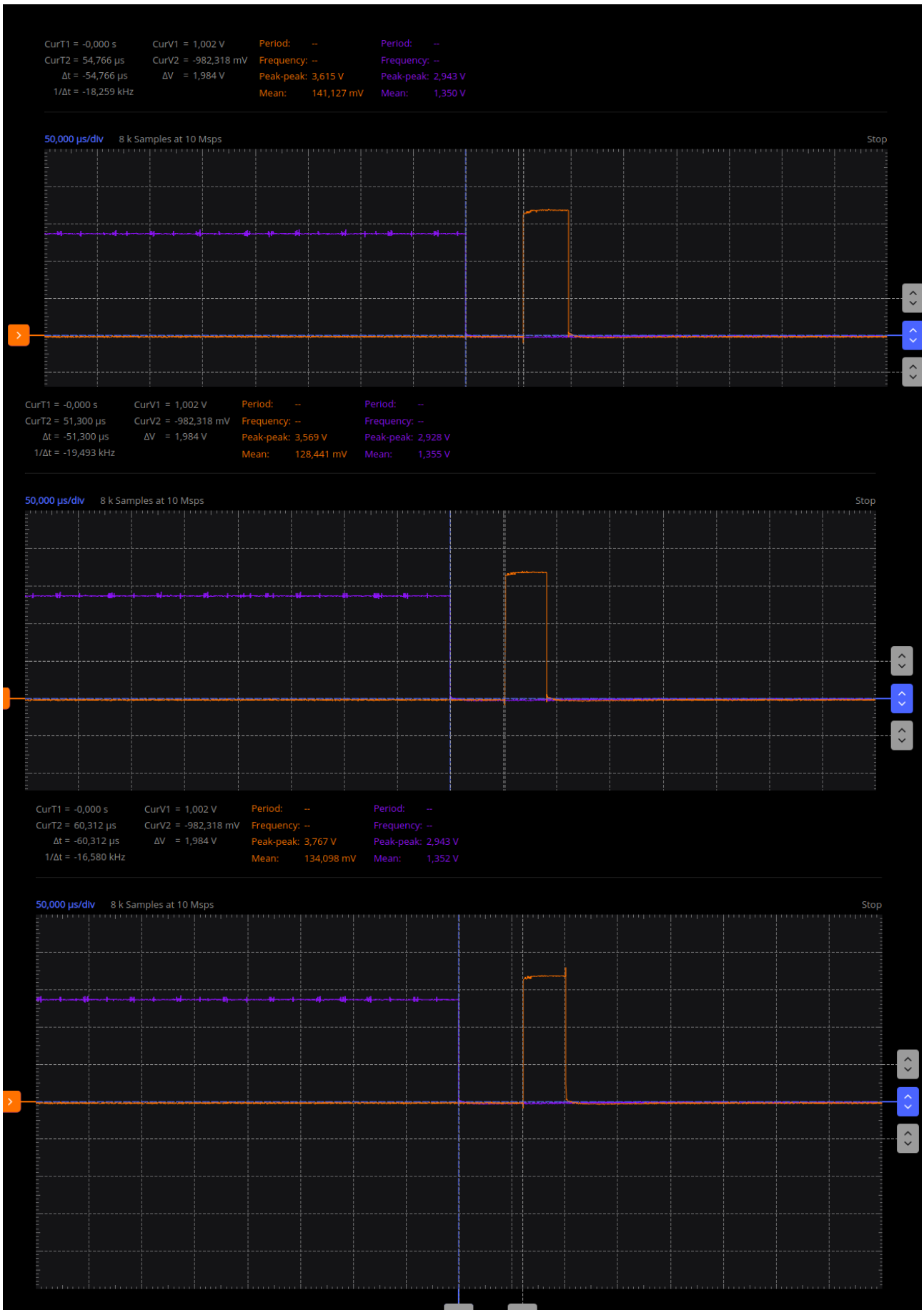
Georgios Markou EZV 3
1)

```
#kernel=kernel-4.19.img
kernel=kernel-4.19-rt.img
#kernel=kernel.img
#initramfs initrd.gz

# For more options and information see
# http://rpf.io/configtxt
# Some settings may impact device functionality. See link above for details

# uncomment if you get no picture on HDMI for a default "safe" mode
#hdmi_safe=1

# uncomment this if your display has a black border of unused pixels visible
# and your display can output without overscan
disable_overscan=1

# uncomment the following to adjust overscan. Use positive numbers if console
# goes off screen, and negative if there is too much border
#overscan_left=16
#overscan_right=16
#overscan_top=16
#overscan_bottom=16

"/boot/config.txt" 92L, 2169B                                 1,1           Anfang
```

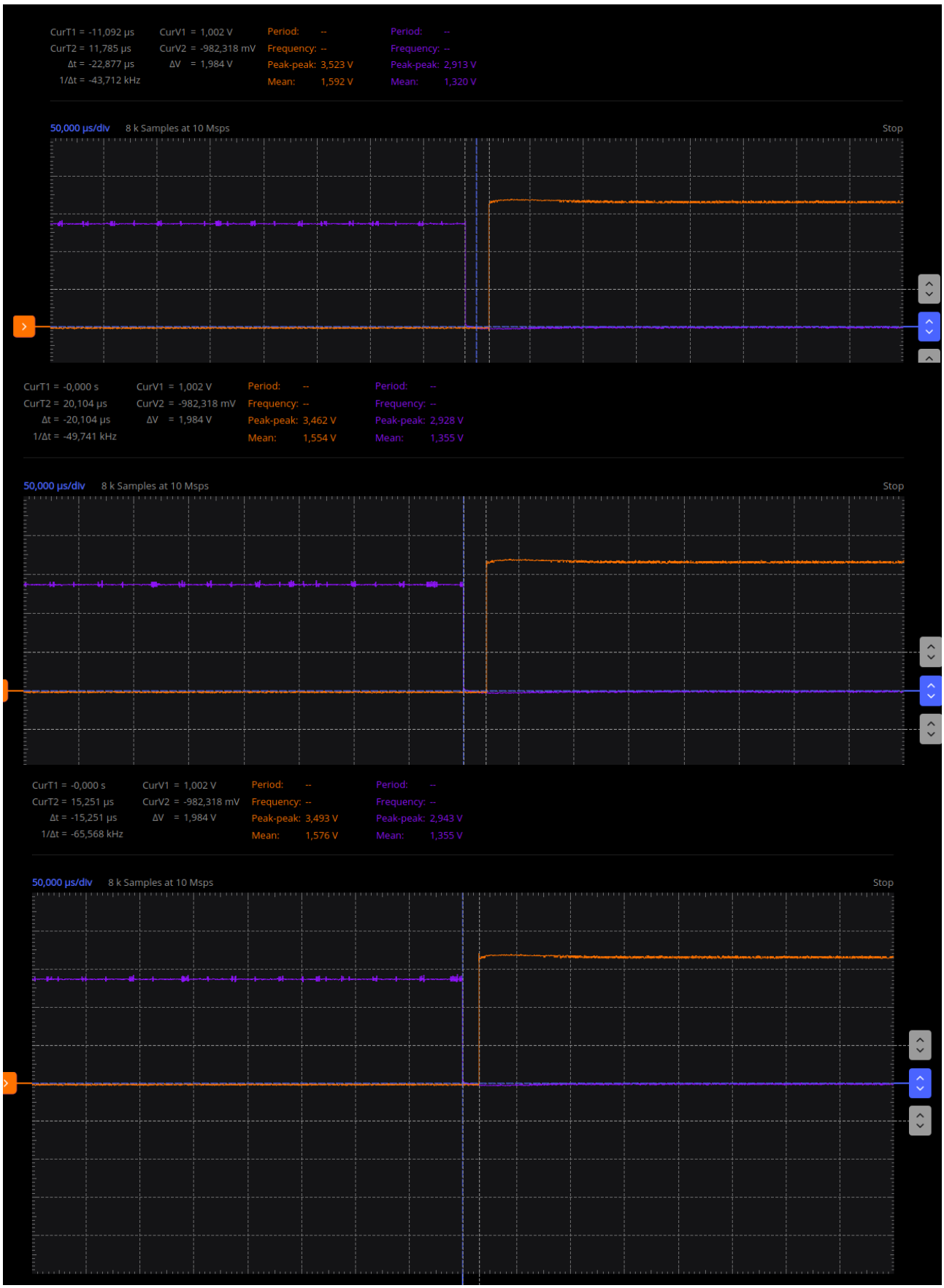
Um das Realtime-Modul zu laden muss die zweite Zeile der Datei „/boot/config.txt“ auskommentiert werden (s. Bild). Für das Laden des Non-Realtime-Moduls muss die erste Zeile auskommentiert werden. Im nächsten Bild sieht man wie nach einem Neustart das Realtime-Modul geladen wird.

```
pi@raspberrypi:~$ sudo insmod ezv_rt.ko
[sudo] Passwort für pi:
[   72.965586] ezv: loading out-of-tree module taints kernel.
[   72.986278] EZV: Init EZV kernel module
[   72.998988] EZV: The button (GPIO 27) state is currently: 1
[   73.019621] EZV: Button is mapped to IRQ #161
[   73.041884] EZV: Button interrupt request is: 0
pi@raspberrypi:~$
```

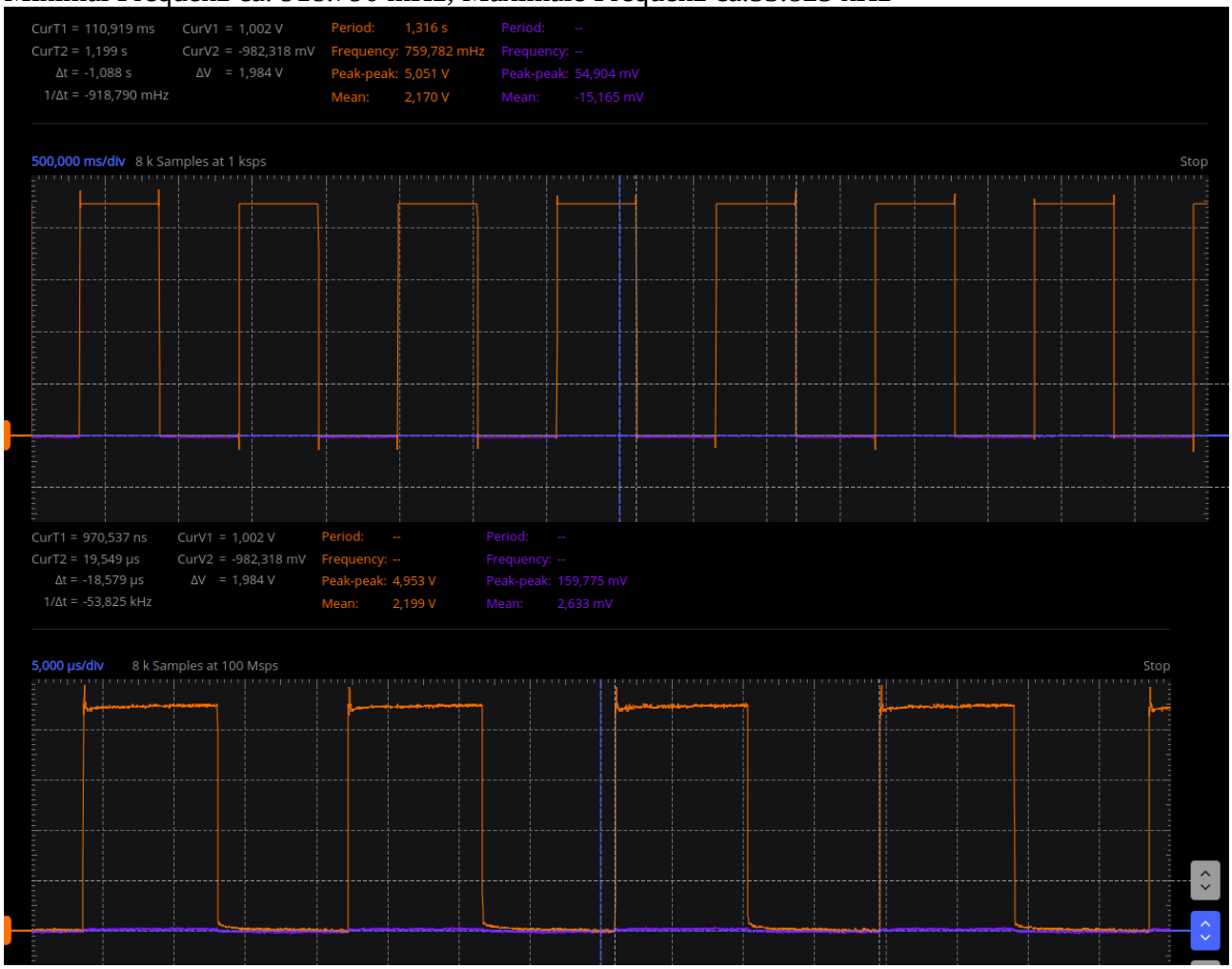
Die nächsten 3 Bilder zeigen die Messungen mit dem Realtime-Modul. Die durchschnittliche Antwortzeit beträgt ca. 55.4 µS.



Die nächsten 3 Bilder zeigen die Messungen mit dem Non-Realtime-Modul. Die durchschnittliche Antwortzeit beträgt ca. 19.4 µS.


CurT1 = -11,092 µs
CurT2 = 11,785 µs
Δt = -22,877 µs
1/Δt = -43,712 kHz
CurV1 = 1,002 V
CurV2 = -982,318 mV
ΔV = 1,984 V
Period: --
Frequency: --
Peak-peak: 3,523 V
Mean: 1,592 V
Period: --
Frequency: --
Peak-peak: 2,913 V
Mean: 1,320 V
50,000 µs/div
8 k Samples at 10 Msps
Stop
CurT1 = -0,000 s
CurT2 = 20,104 µs
Δt = -20,104 µs
1/Δt = -49,741 kHz
CurV1 = 1,002 V
CurV2 = -982,318 mV
ΔV = 1,984 V
Period: --
Frequency: --
Peak-peak: 3,462 V
Mean: 1,554 V
Period: --
Frequency: --
Peak-peak: 2,928 V
Mean: 1,355 V
50,000 µs/div
8 k Samples at 10 Msps
Stop
CurT1 = -0,000 s
CurT2 = 15,251 µs
Δt = -15,251 µs
1/Δt = -65,568 kHz
CurV1 = 1,002 V
CurV2 = -982,318 mV
ΔV = 1,984 V
Period: --
Frequency: --
Peak-peak: 3,493 V
Mean: 1,576 V
Period: --
Frequency: --
Peak-peak: 2,943 V
Mean: 1,355 V
50,000 µs/div
8 k Samples at 10 Msps
Stop

2)
Ich habe erneut die Frequenz am OSC Pin gemessen.
Minimal Frequenz ca. 918.790 mHz; Maximale Frequenz ca.53.825 kHz

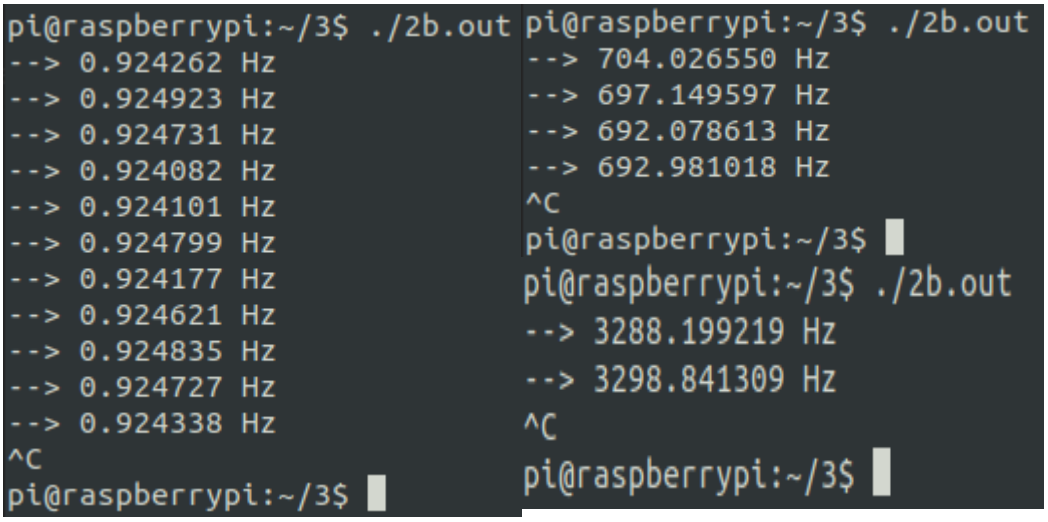


Die Polling-Version (2a.c) hat folgende Frequenzen gemessen (in 5 Sek. Abständen).

```
pi@raspberrypi:~/3$ ./2a.out
--> 0.923441 Hz
--> 0.923515 Hz
--> 0.923548 Hz
--> 0.923306 Hz
--> 0.924172 Hz
--> 0.923714 Hz
--> 0.923756 Hz
--> 0.924197 Hz
--> 0.923436 Hz
--> 0.922916 Hz
--> 0.923248 Hz
--> 0.923245 Hz
--> 0.923199 Hz
--> 0.923006 Hz
--> 0.923494 Hz
--> 0.923600 Hz
--> 0.923104 Hz
--> 0.923134 Hz
--> 0.924074 Hz
^C
pi@raspberrypi:~/3$
```

```
pi@raspberrypi:~/3$ ./2a.out
--> 52886.628906 Hz
--> 52920.238281 Hz
--> 52823.421875 Hz
--> 52877.976562 Hz
--> 52855.925781 Hz
--> 52845.363281 Hz
--> 52930.578125 Hz
--> 52887.265625 Hz
--> 52727.128906 Hz
--> 52849.121094 Hz
--> 52890.039062 Hz
--> 52849.988281 Hz
--> 52806.152344 Hz
--> 52832.656250 Hz
--> 52869.574219 Hz
--> 52869.273438 Hz
--> 52813.644531 Hz
--> 52861.886719 Hz
--> 52825.761719 Hz
^C
pi@raspberrypi:~/3$
```

Die nächsten drei Bilder wurden mit der Interrupt-Version (2b.c) gemacht.
Das linke Bild zeigt die Messungen der kleinsten Frequenz.
Das obere rechte Bild zeigt die (falsche) Messung der höchsten Frequenz.
Das untere rechte Bild zeigt die (falsche) Messung ab der Frequenz die nicht mehr richtig gemessen wird (5 kHz).



Wenn das Signal mit einer Frequenz von 100 Hz abgetastet wird kann man theoretisch maximal 50 Hz korrekt messen. Ich habe mit einer Abtastfrequenz f1 = 100 Hz ein Signal mit der Frequenz f2 = 100 Hz versucht zu messen. Einmal wurde die Frequenz 49 Hz gemessen, alle anderen Ergebnisse lagen weit drunter. Mithilfe eines FPGAs (der Funktionsgenerator des ADALM 2000 hat nicht immer funktioniert), habe ich ein 50 Hz Signal erzeugt. Dieses Signal wurde mit der Frequenz f1 = 100 Hz abgetastet und es wurde die Frequenz f2 = 49 Hz +- (0.2).

Es handelt sich um das Nyquist–Shannon sampling theorem (dt. Nyquist-Shannon-Abtasttheorem). Dieses beschreibt das um ein beliebig geformtes kontinuierliches Signal mit Frequenz f1 korrekt zu erfassen muss man es mindestens mit einer Frequenz f2 = f1 * 2 abtasten. Nur wenn die Bedingung „f2 >= f1 * 2“ gilt, kann man aus dem Signal mit der Frequenz f2, das Signal mit der Frequenz f1 rekonstruieren.