

1)

```
pi@raspberrypi:~/2$ cat 1.c
#include <wiringPi.h>
#include <softPwm.h>
#include <stdint.h>
#include <assert.h>

void mySoftPwm(uint32_t us, uint8_t dutyCycle, int pin)
{
    assert(dutyCycle >= 0 && dutyCycle <= 100);
    softPwmCreate(pin, 0, 100);
    softPwmWrite(pin, dutyCycle);
    delayMicroseconds(us);
}

int main()
{
    int pin = 1;

    wiringPiSetup();

    while (1)
    {
        mySoftPwm(100, 10, pin);
    }

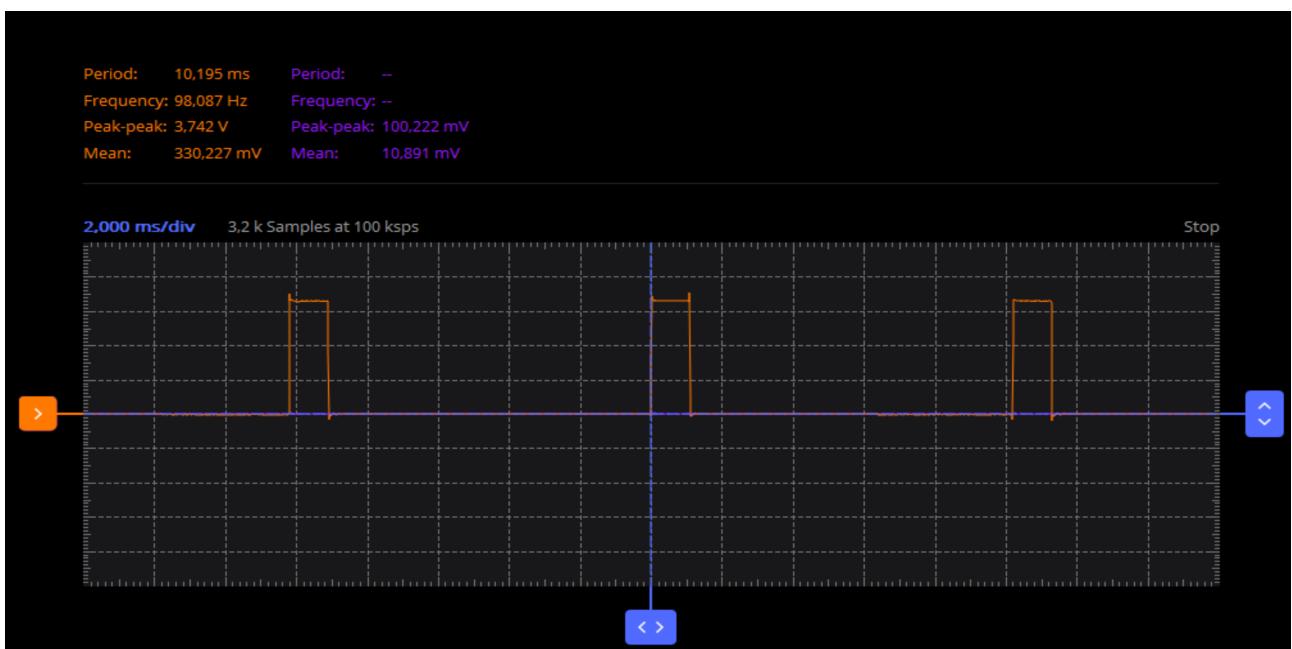
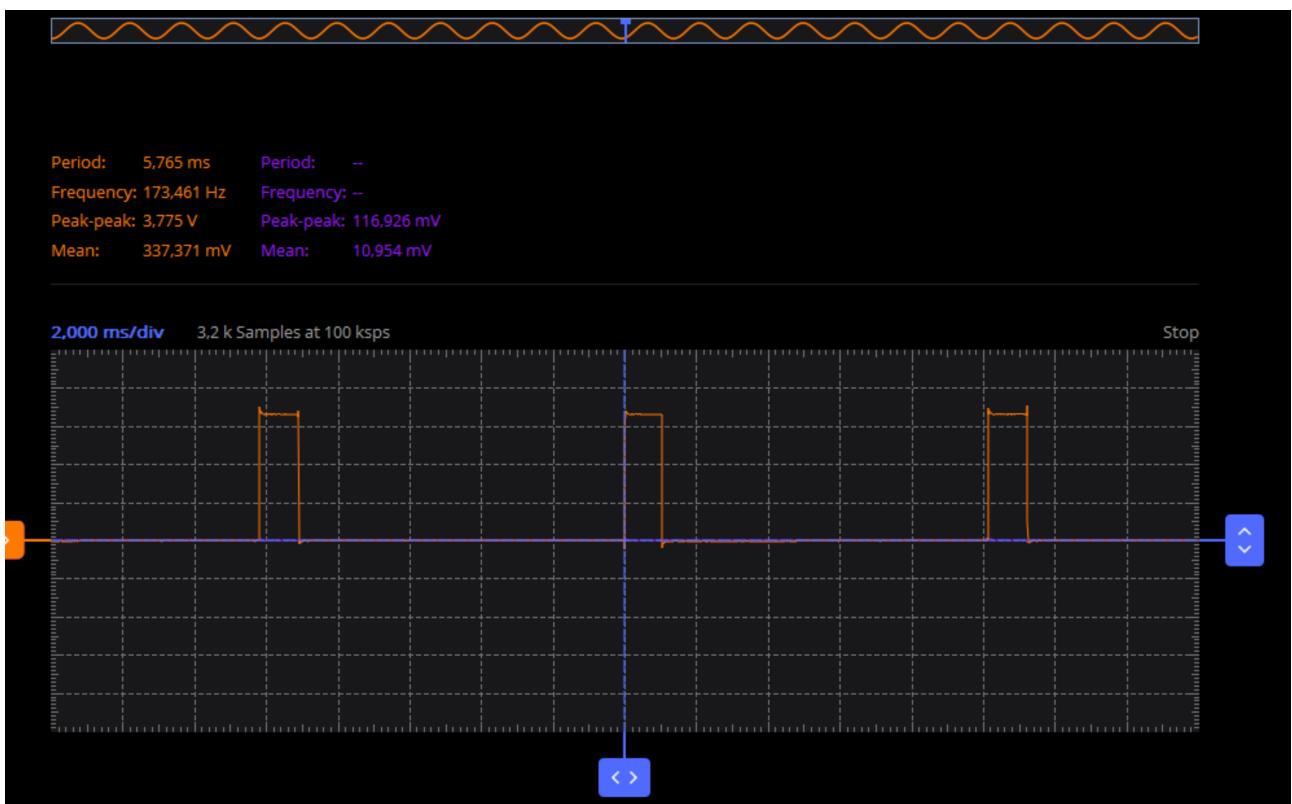
    return 0;
}
pi@raspberrypi:~/2$ ./1.out
```

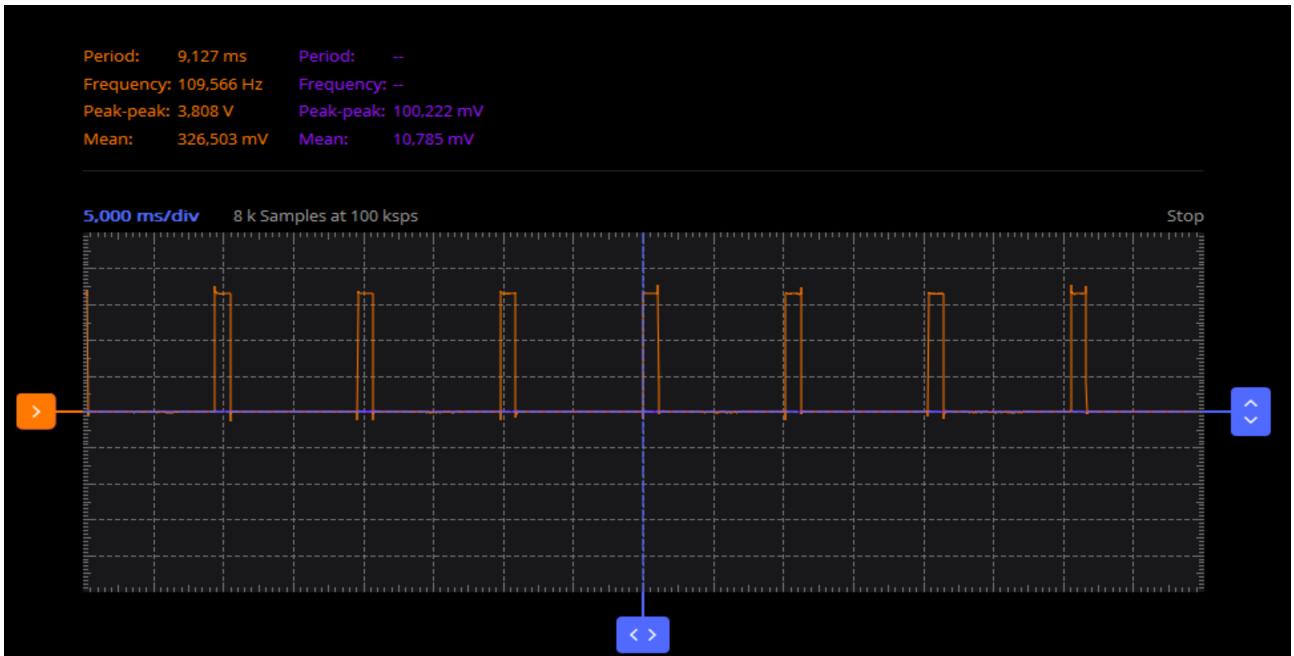
Code von 1.c

Das erste Bild (unten) hat eine Zykluszeit von 1000 us.

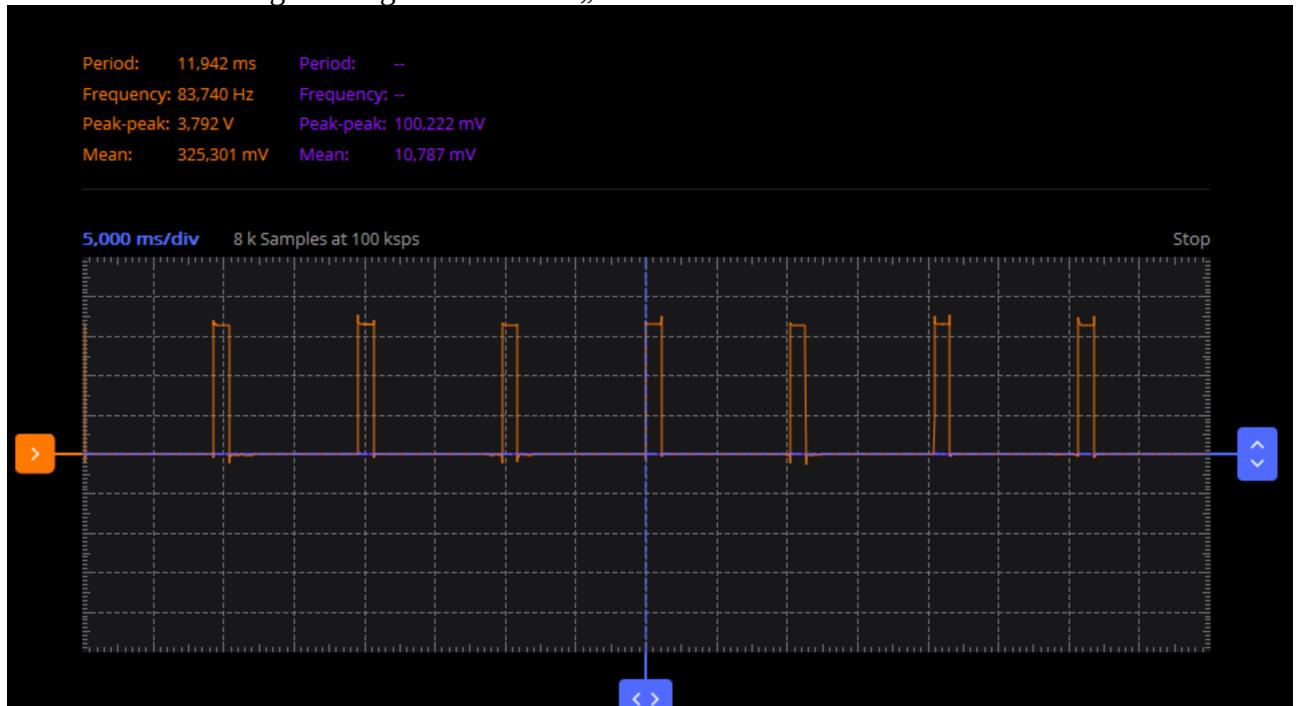
Das nächsten zwei Bilder haben eine Zykluszeit von 100 us.

Man sieht das die Frequenz schwankt.

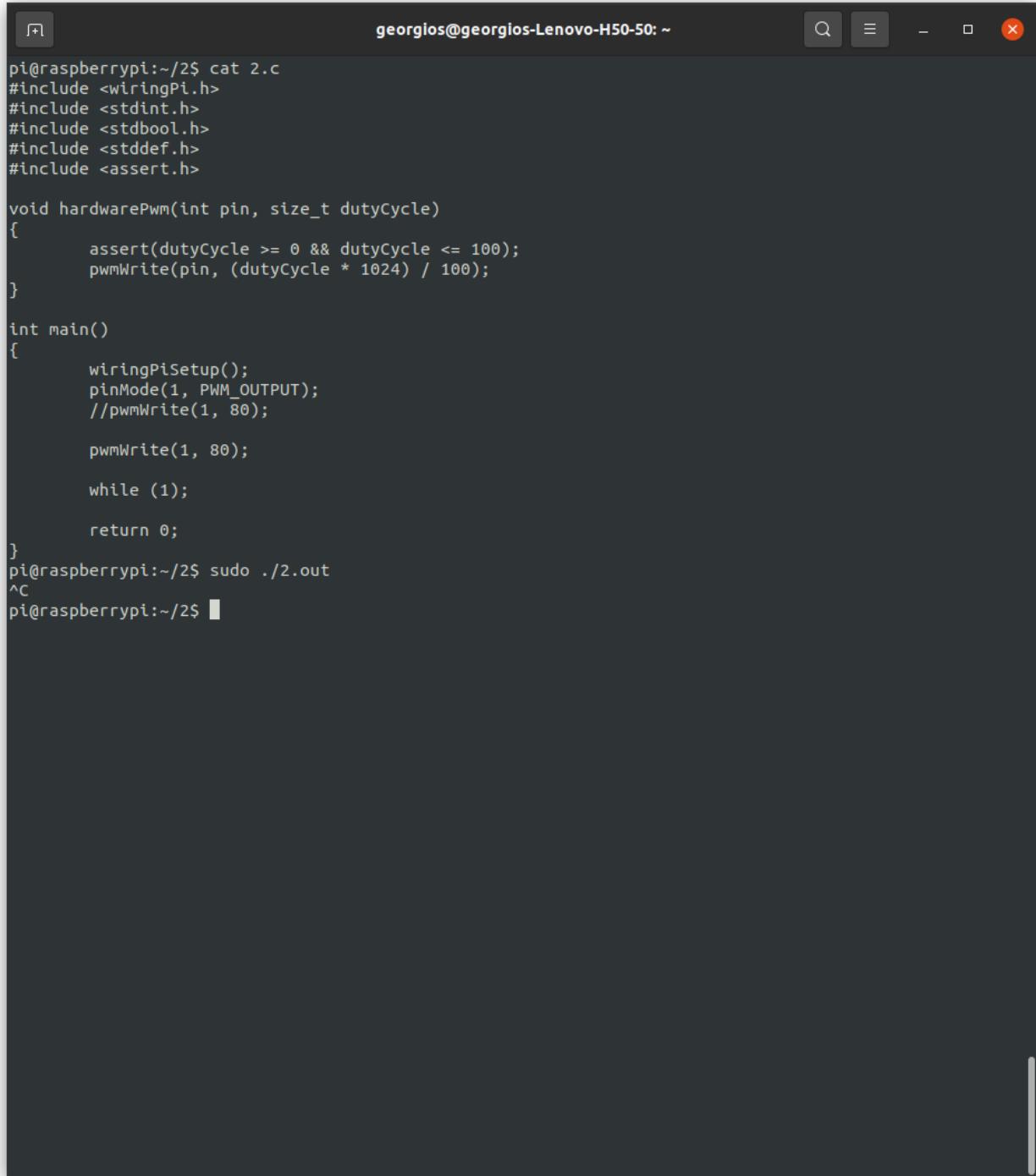




Das nächste Bild zeigt das Signal unter Last „md5sum /dev/zero“.



2)



The screenshot shows a terminal window titled "georgios@georgios-Lenovo-H50-50: ~". The window contains the following C code:

```
pi@raspberrypi:~/2$ cat 2.c
#include <wiringPi.h>
#include <stdint.h>
#include <stdbool.h>
#include <stddef.h>
#include <assert.h>

void hardwarePwm(int pin, size_t dutyCycle)
{
    assert(dutyCycle >= 0 && dutyCycle <= 100);
    pwmWrite(pin, (dutyCycle * 1024) / 100);
}

int main()
{
    wiringPiSetup();
    pinMode(1, PWM_OUTPUT);
    //pwmWrite(1, 80);

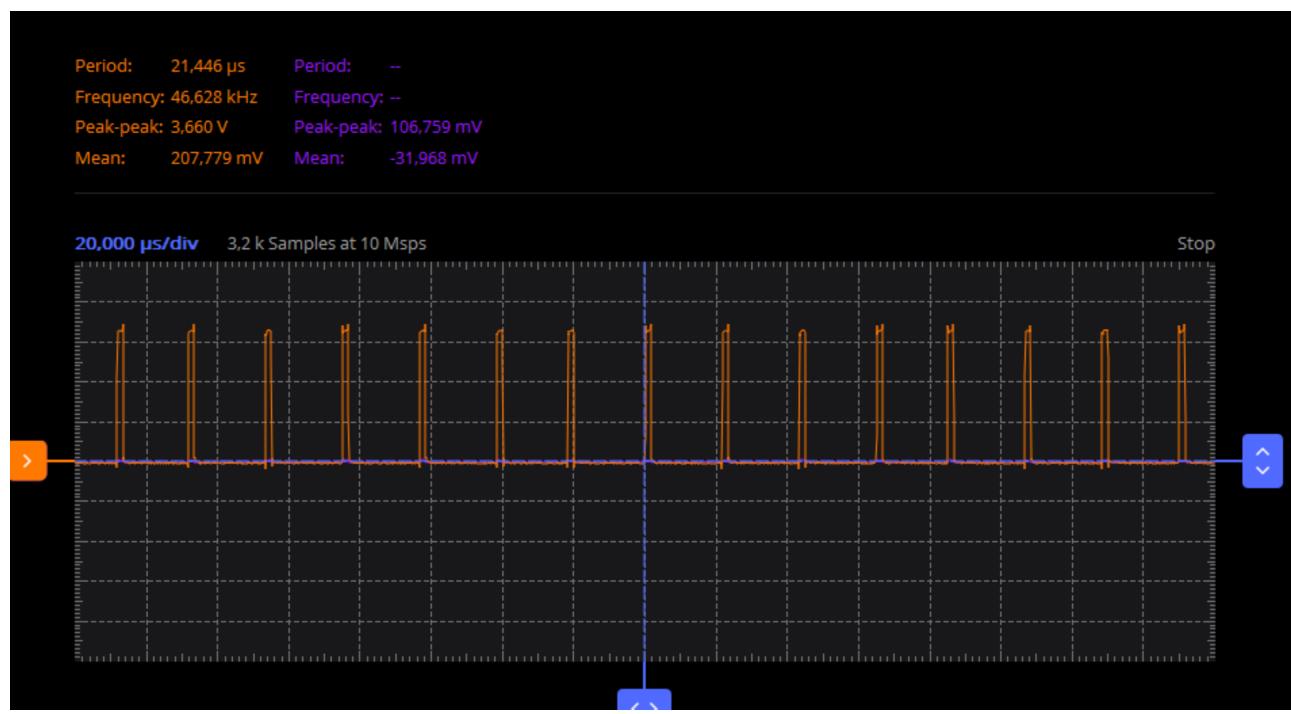
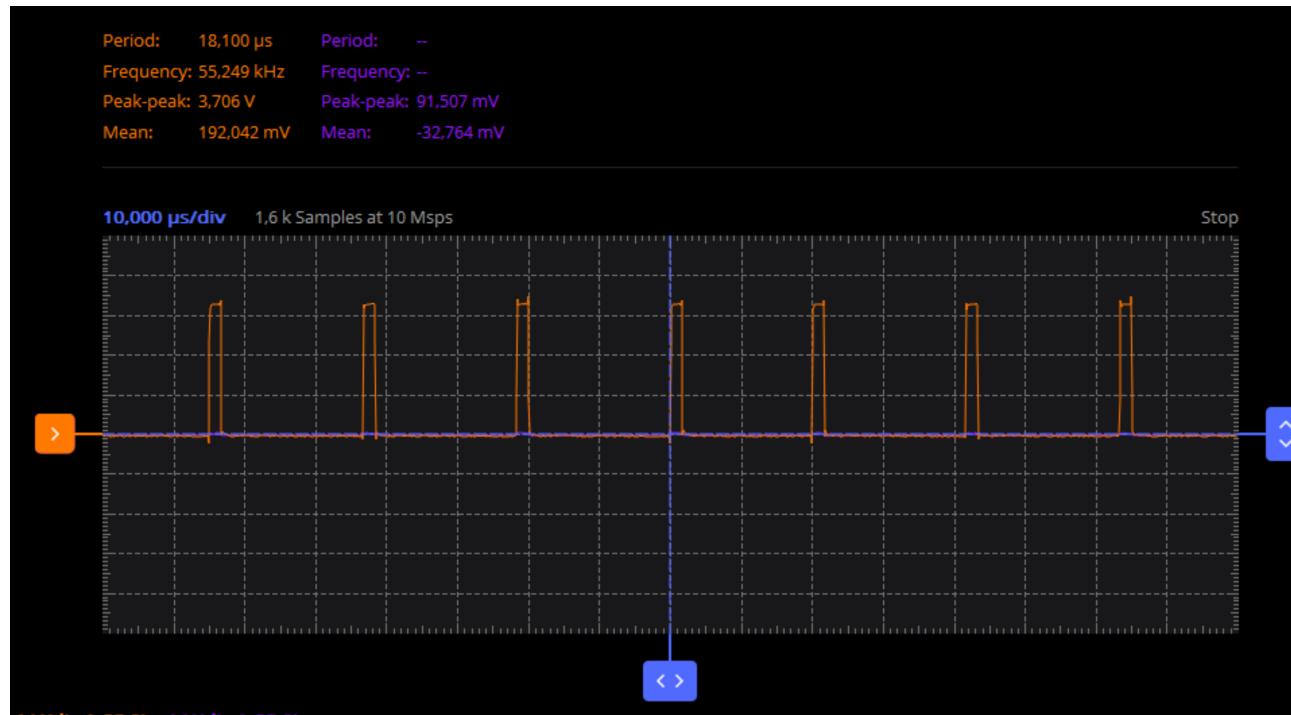
    pwmWrite(1, 80);

    while (1);

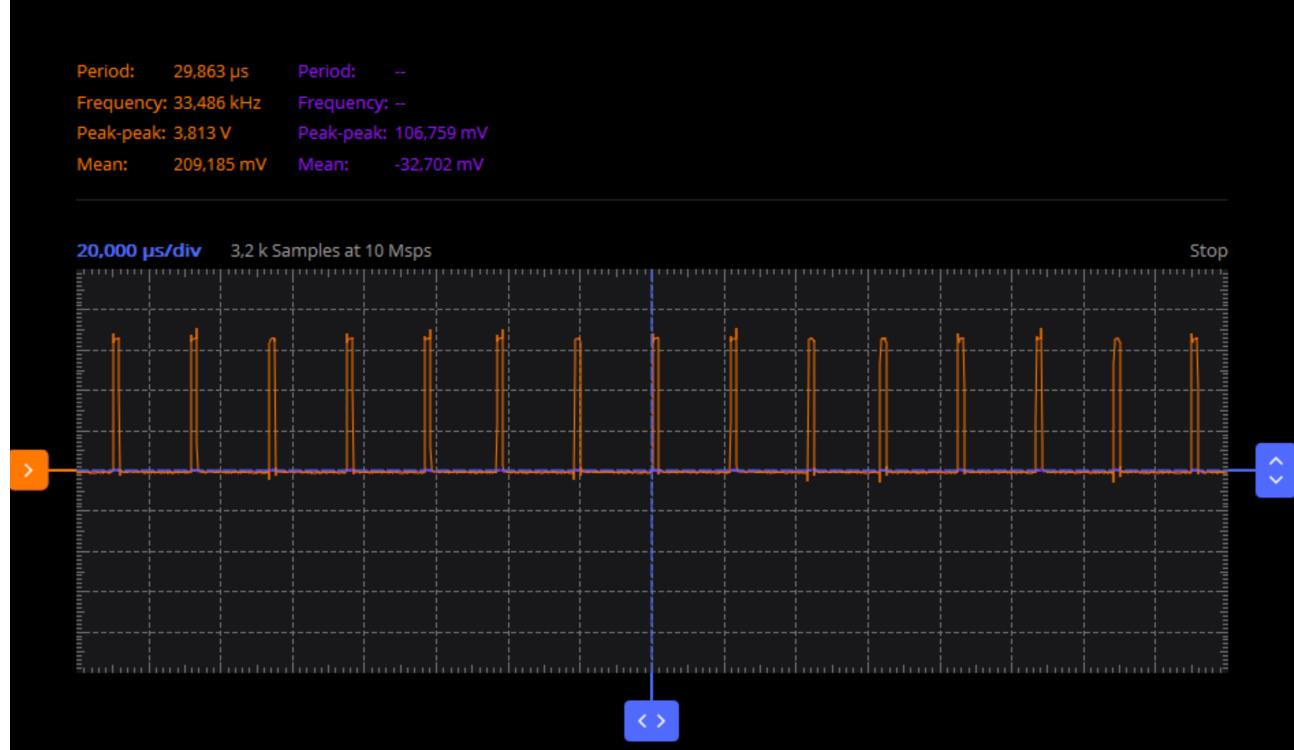
    return 0;
}
pi@raspberrypi:~/2$ sudo ./2.out
^C
pi@raspberrypi:~/2$
```

Code für Aufgabe 2.

Die folgenden zwei Bilder wurde ohne Hintergrundlast gemacht. Die Frequenz schwankt (1. Bild f=55.249 kHz 2. Bild 46.628 kHz). Es gibt aber keine Aussetzer.



Das nächste Bild wurde mit Hintergrundlast (md5sum wie bei Aufgabe 1) gemacht. Die Frequenz ist deutlich niedriger.



Die Hardware-PWM erreicht höhere Frequenzen mit weniger Schwankungen/Aussetzern.

4)

```
pi@raspberrypi:~/2$ cat 4.c
#include <wiringPi.h>
#include <stdint.h>
#include <stdlib.h>

int main(int argc, char* argv[])
{
pi@raspberrypi:~/2$ cat 4.c
#include <wiringPi.h>
#include <stdint.h>
#include <stdlib.h>

int main(int argc, char* argv[])
{
    int pinSwitch = 2;
    int pinGpio = 1;

    wiringPiSetup();

    pinMode(pinSwitch, INPUT);
    pinMode(pinGpio, OUTPUT);
    digitalWrite(pinGpio, LOW);

    while (1)
    {
        if (digitalRead(pinSwitch) == LOW)
        {
            digitalWrite(pinGpio, HIGH);
            delayMicroseconds(1);
            digitalWrite(pinGpio, LOW);
        }
    }

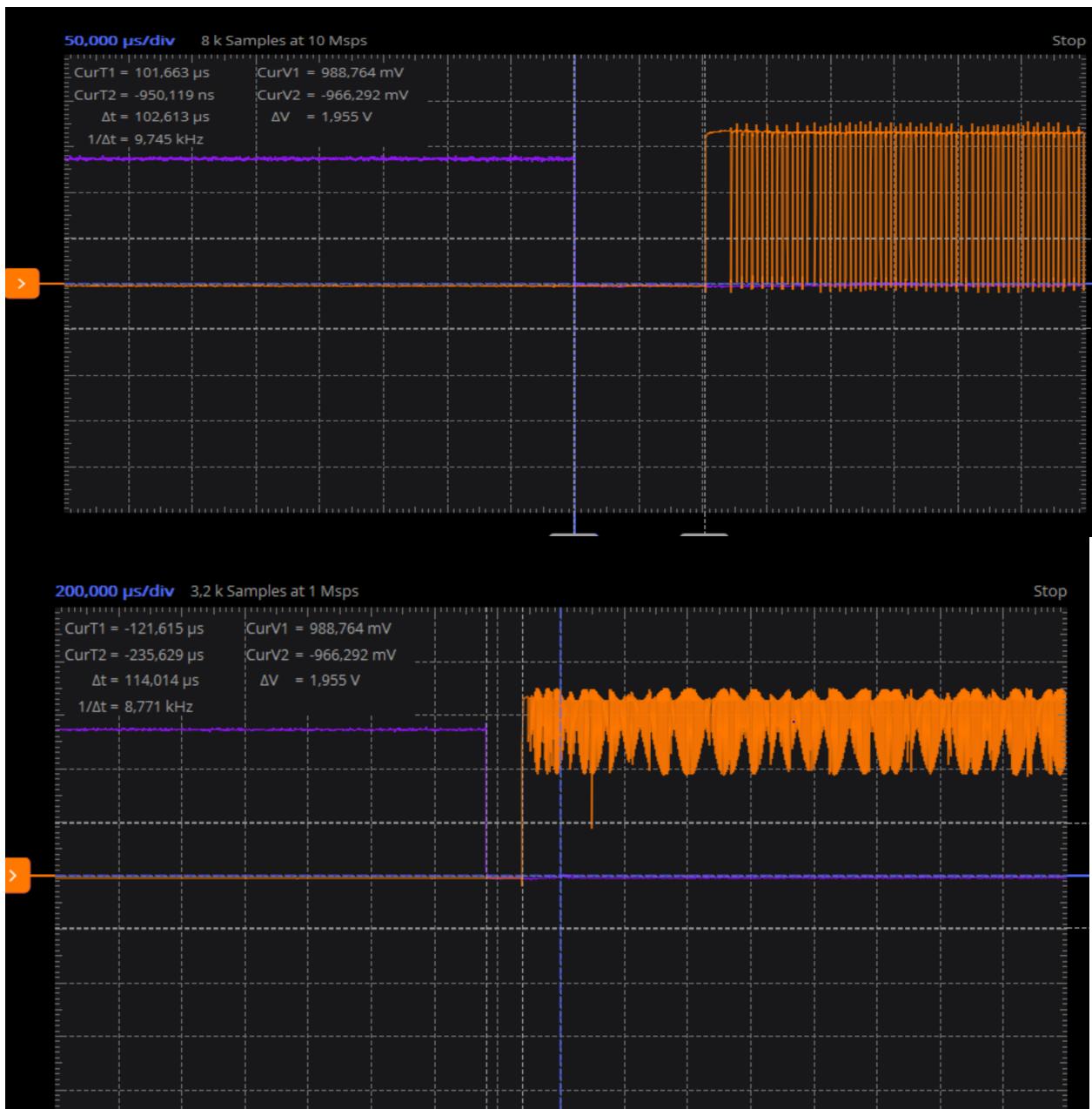
    return 0;
}
pi@raspberrypi:~/2$
```

Code für Aufgabe 4.

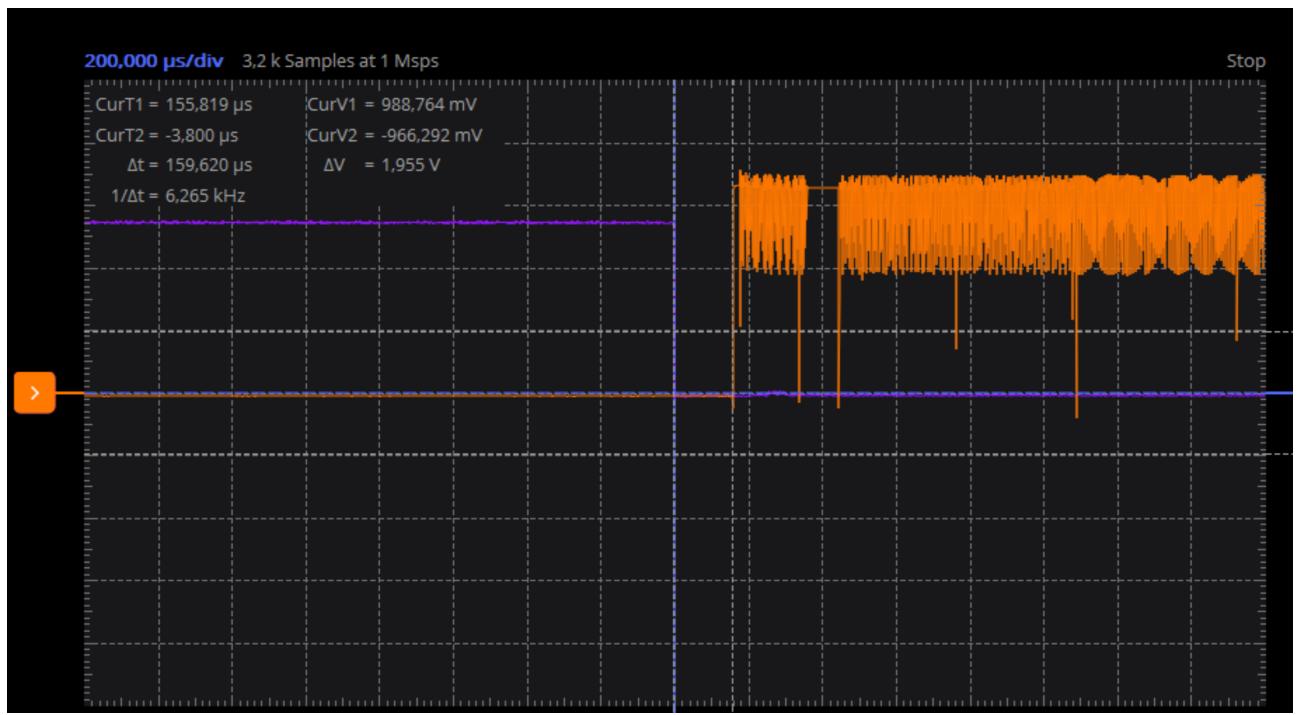
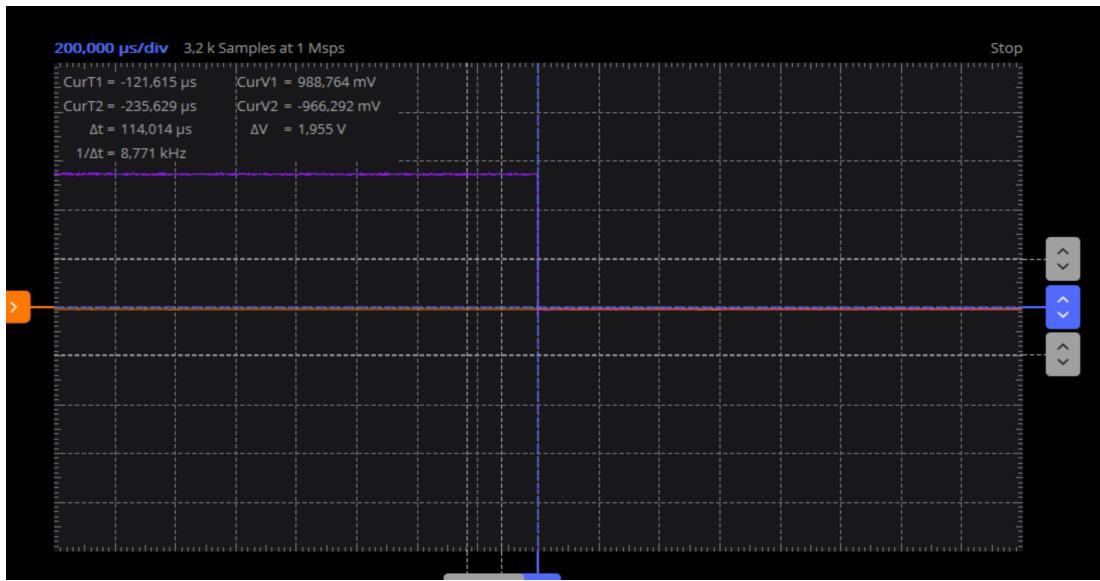
Die nächsten zwei Bilder zeigen das Signal ohne Hintergrundlast.

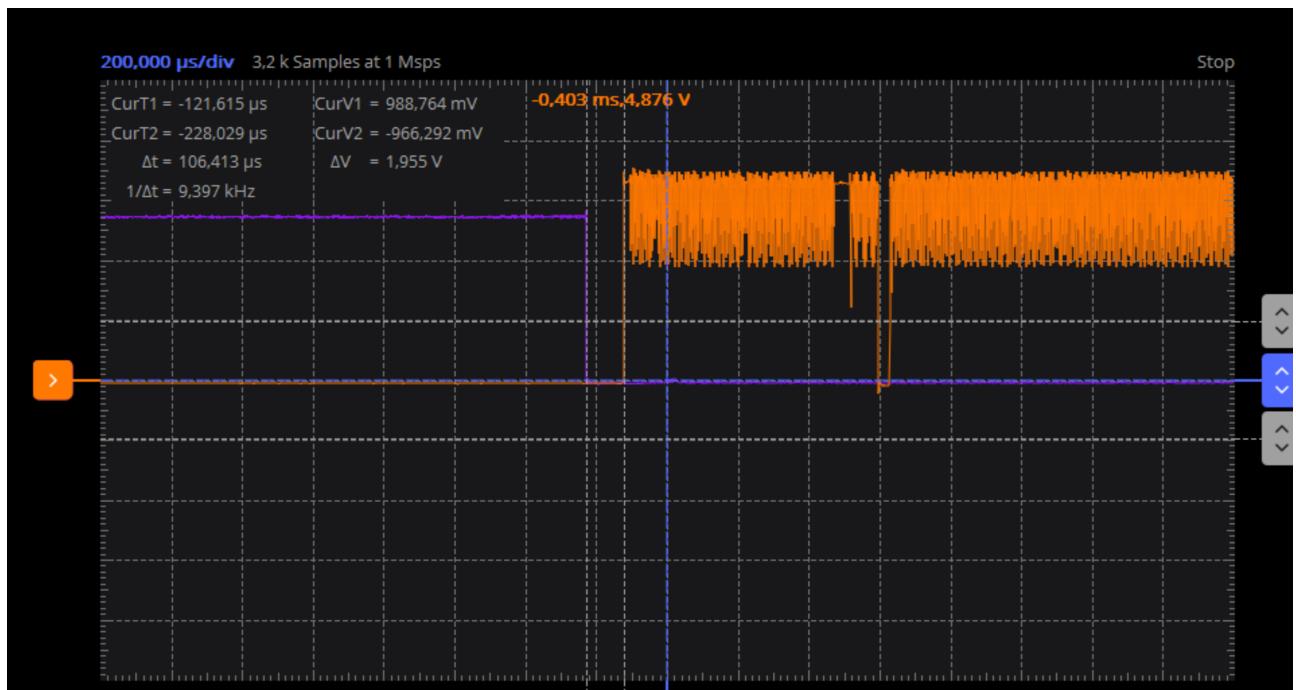
Durchschnittlich ist Periode = 106 µS.

Delay = 1 µS



Die nächsten drei Bilder wurden bei gleichen Einstellung unter Last (wieder md5sum) gemacht.

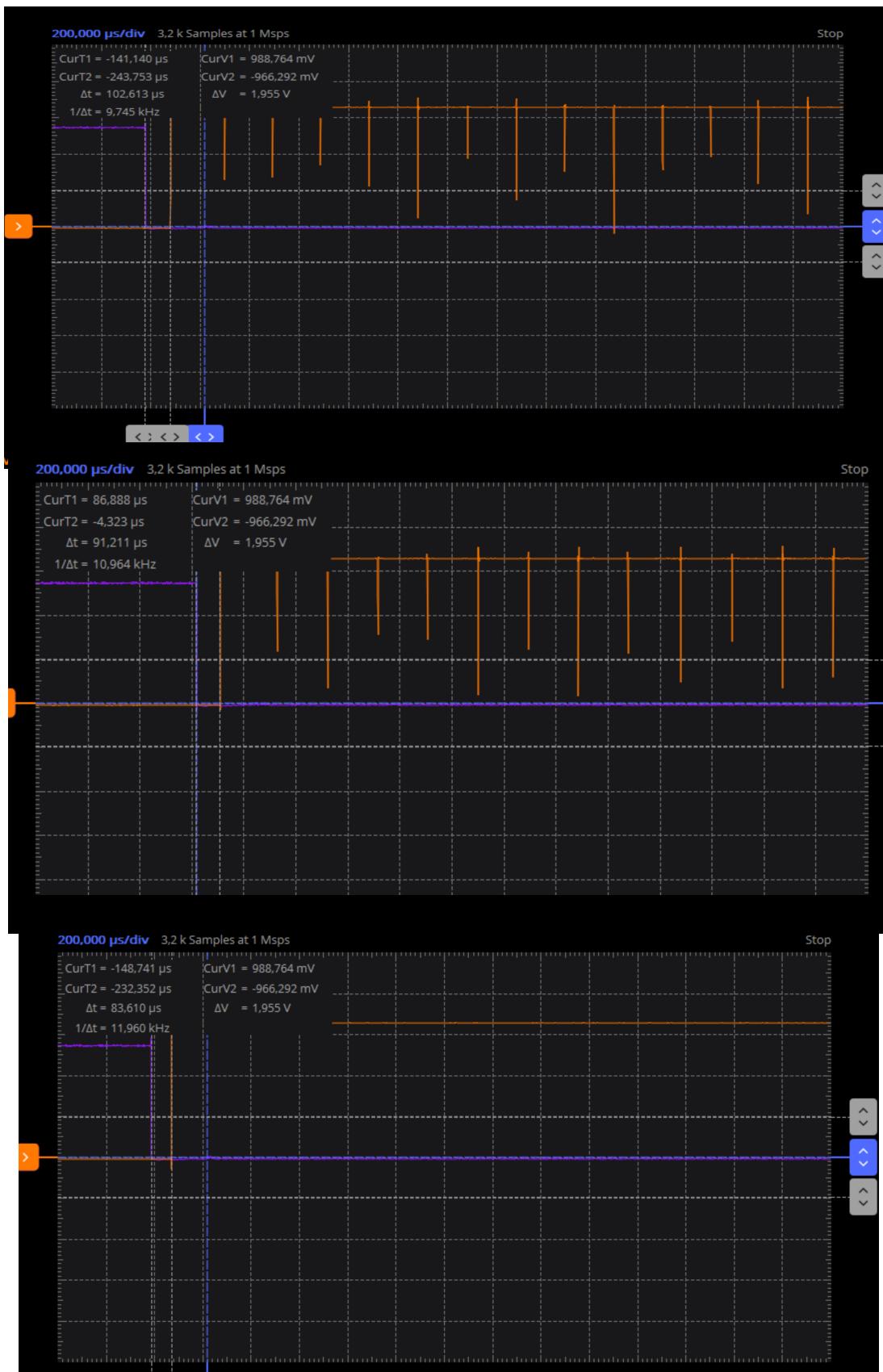




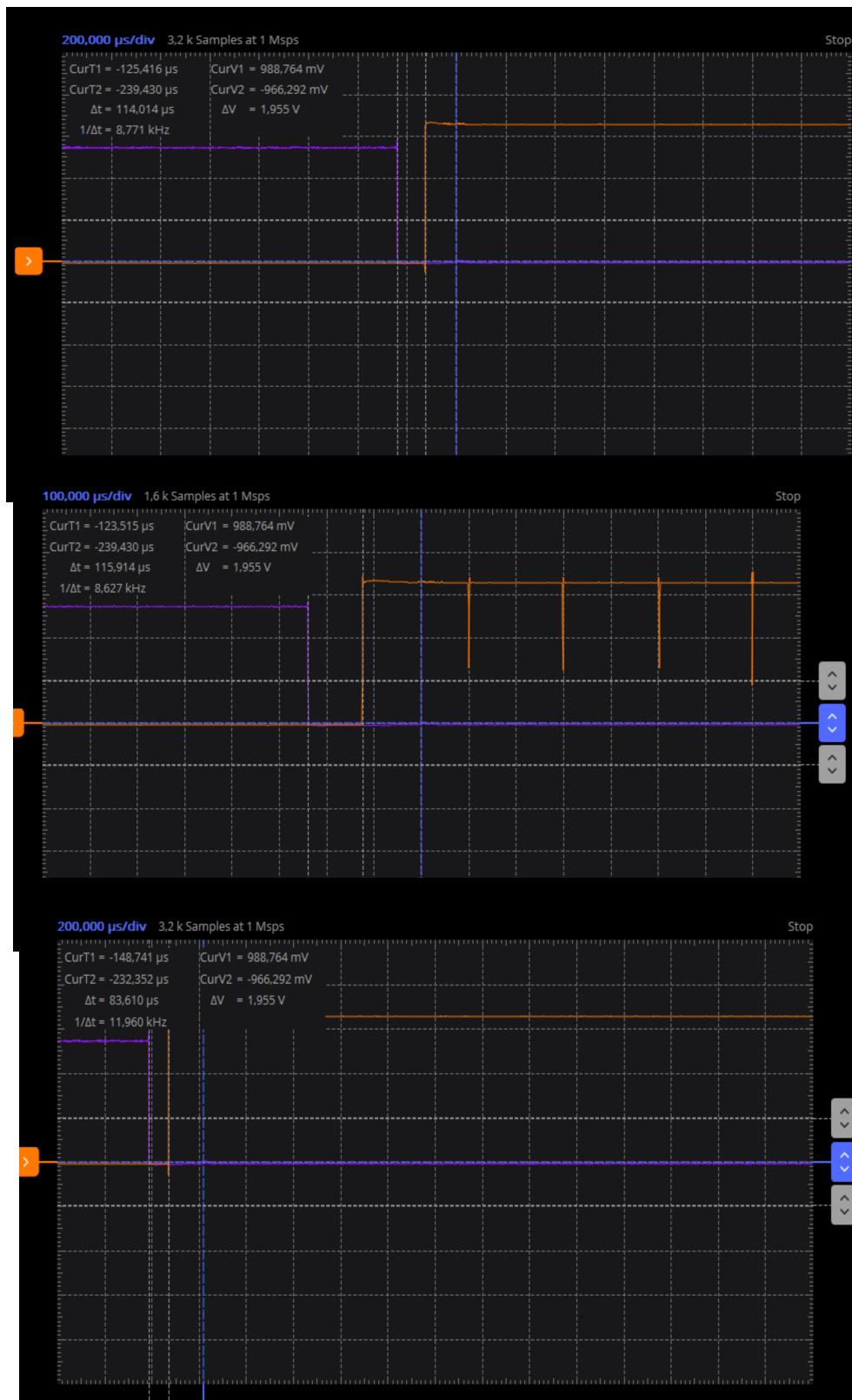
Die Periode beträgt durchschnittlich ca 127 μ S.

Für die nächsten Bilder ist Delay=100 μ S.

Die ersten drei Bilder wurden ohne Last gemacht.



Die Periode beträgt durchschnittlich ca. 92 μ S



Diese drei Bilder wurden unter Last gemacht. Die durchschnittliche Reaktionszeit beträgt ca. 104 μ S. Um die theoretische ideale Reaktionszeit auszurechnen müsste man sich das Datenblatt anschauen und ermitteln wie viele Befehle ausgeführt werden müssen um einen Interrupt zu bearbeiten.

5)

```
pi@raspberrypi:~/2$ cat 5.c
#include <wiringPi.h>
#include <stdint.h>

static const int Pin = 1;

void ISR()
{
    digitalWrite(Pin, HIGH);
    delayMicroseconds(10000);
    digitalWrite(Pin, LOW);
}

int main()
{
    int pinSwitch = 2;

    wiringPiSetup();
    pinMode(pinSwitch, INPUT);
    pinMode(Pin, OUTPUT);

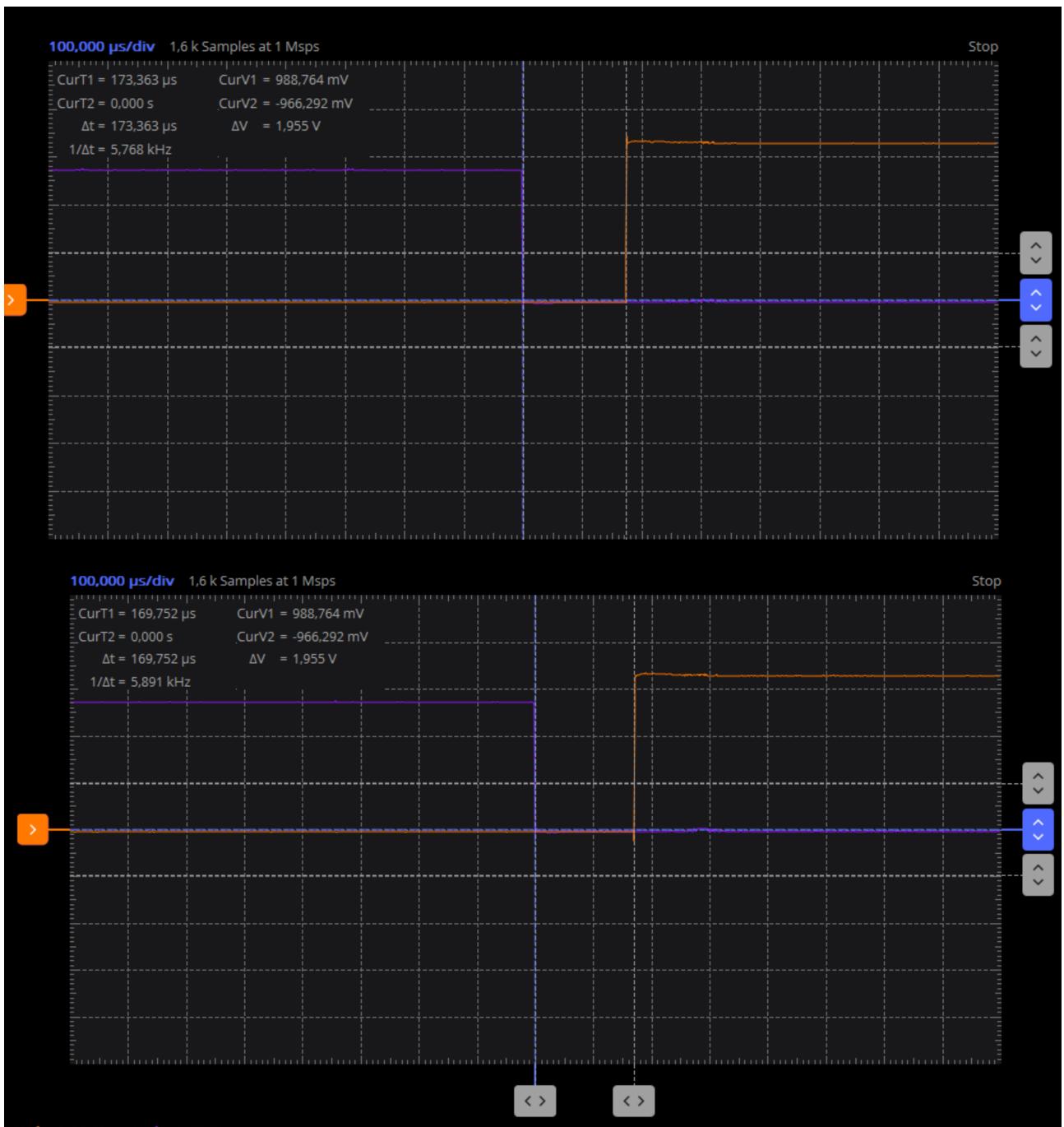
    wiringPiISR(pinSwitch, INT_EDGE_FALLING, ISR);

    while (1);

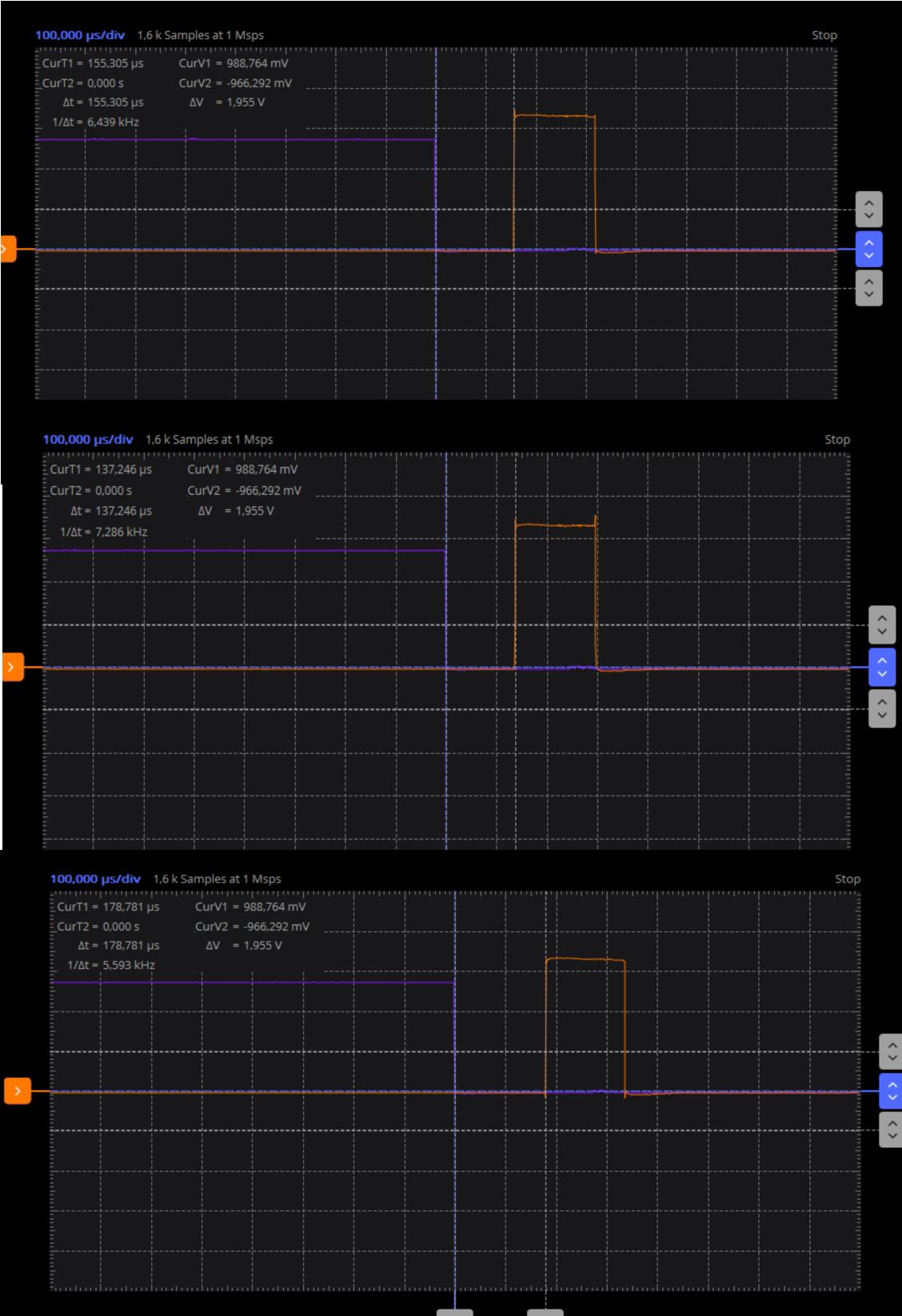
    return 0;
}
pi@raspberrypi:~/2$ sudo ./5.out
```

Code für Aufgabe 5

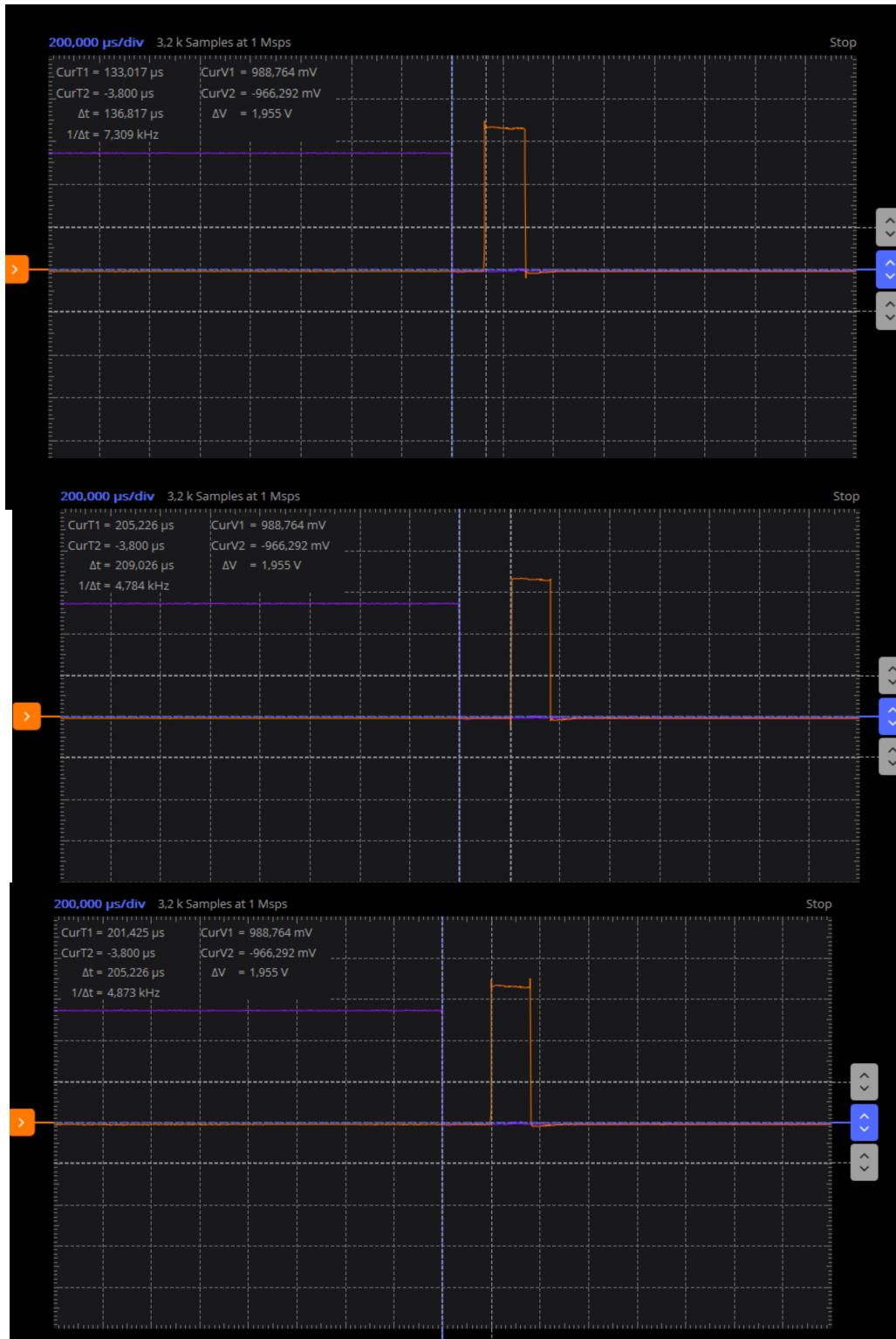
Die nächsten zwei Bilder zeigen die Reaktionszeit ohne Last.
Das Delay ist 1000 μ S.



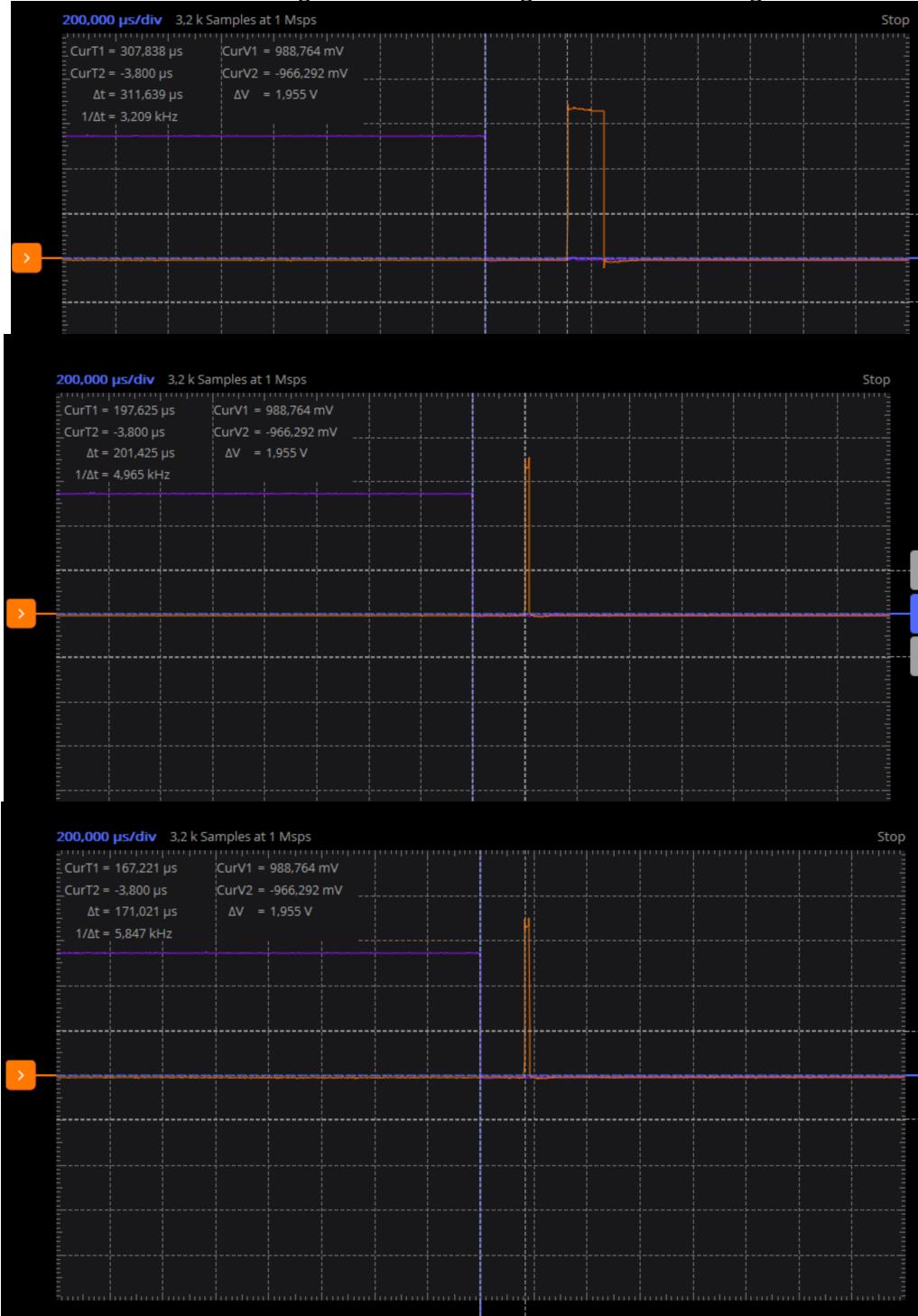
Die nächsten drei Bilder wurden bei gleichen Einstellungen mit Last gemacht.
Die Reaktionszeiten sind nicht schlechter als ohne Last.



Die nächsten drei Bilder wurden wieder ohne Last gemacht. Das Delay beträgt 100 μ s.
Die Antwortzeit ist wesentlich schlechter.

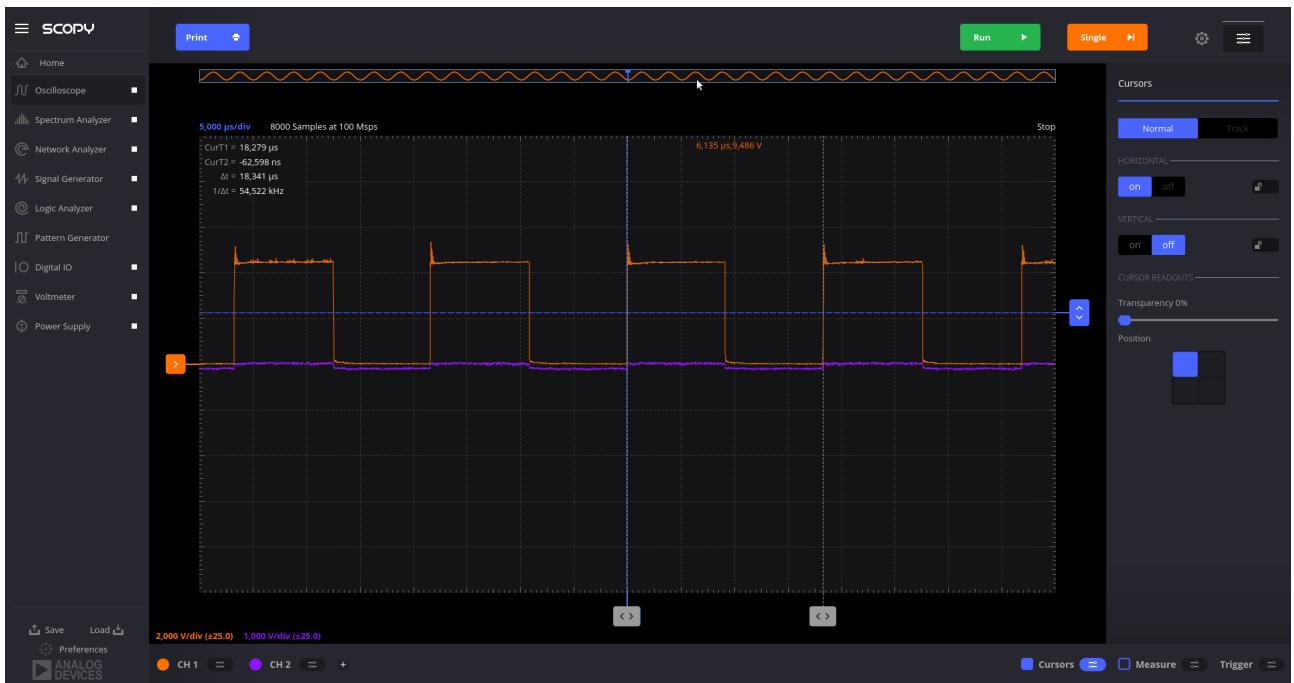
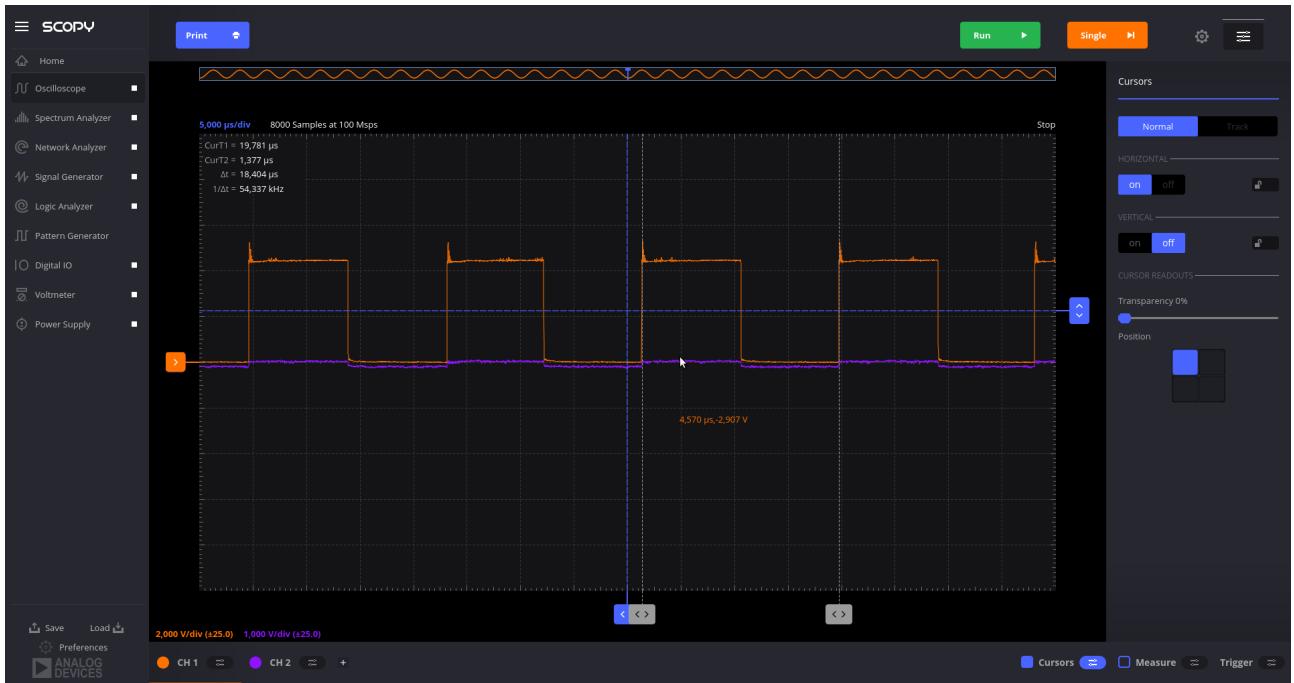


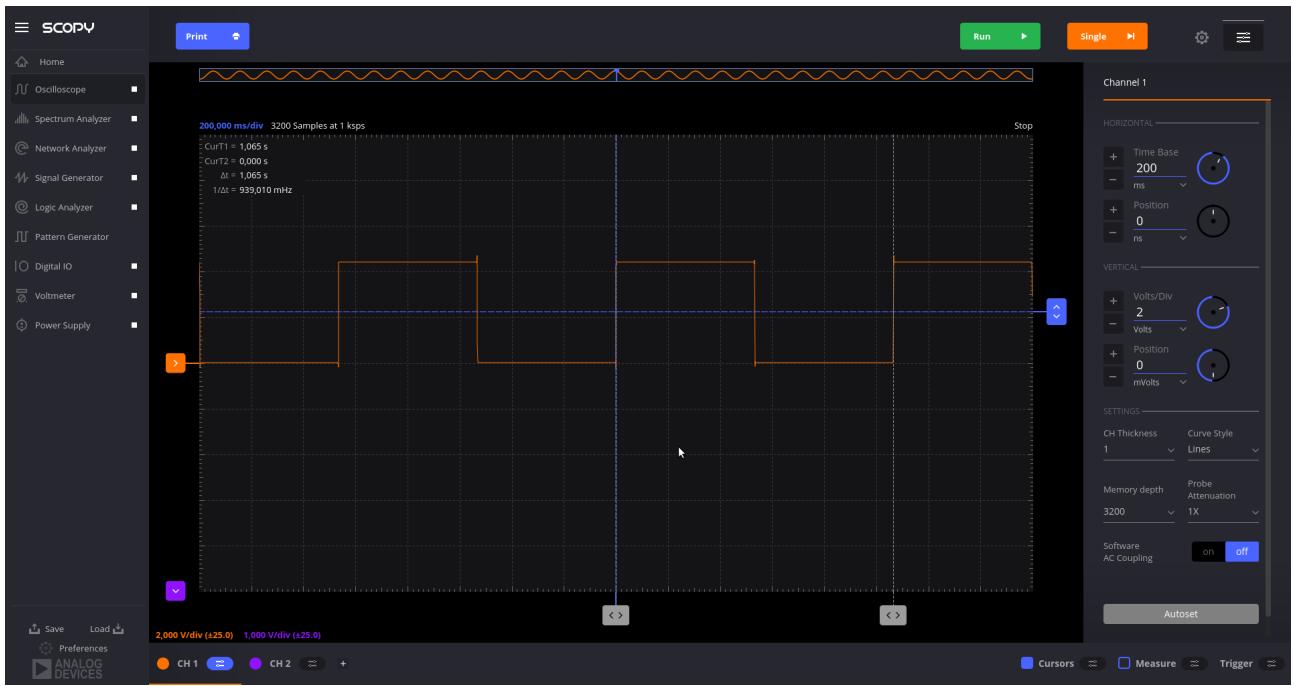
Die nächsten drei Bilder wurde bei gleichen Einstellungen aber unter Last gemacht.



Die Reaktionszeiten sind schlechter. Im ersten Bild sieht man die schlechteste Antwortzeit (311 μS).

3) (alte Abgabe, weil die Sample-Rate dieses mal nicht einstellbar ist und ich die kleinstmögliche Frequenz nicht messen kann).





Wie man sieht lag die höchste gemessene Frequenz bei 54.522 kHz und die niedrigste bei 939 mHz.
Bei der hohen Frequenz gab es Schwankungen im Signal, aber keine Aussetzer