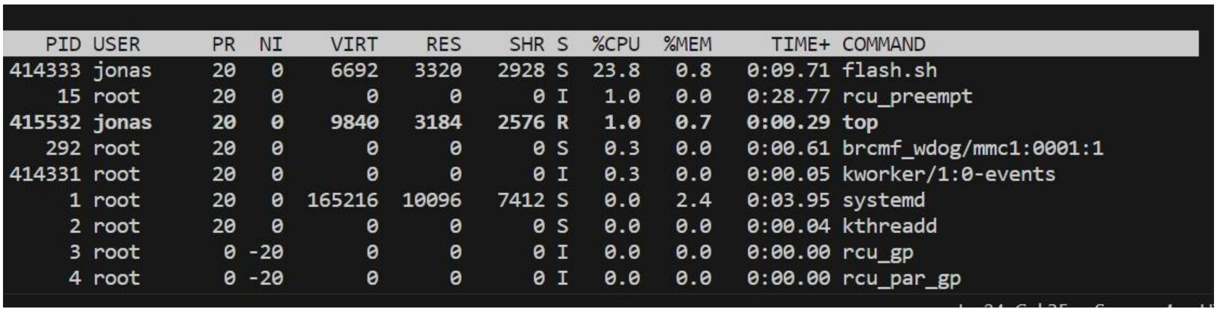
Project 4: Kernel space encoder driver

Part 1:

Compare response time and CPU load:

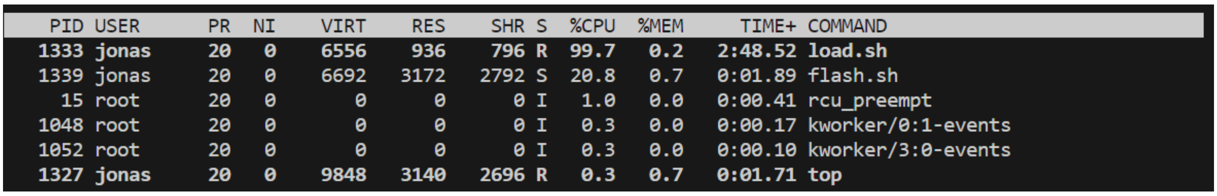
1. **Sysfs from shell script**
   Response Time: Ø 4.28 ms
   CPU Load: 23.8 %
   Results change if CPU fully loaded: Ø 1,97 ms

Top:



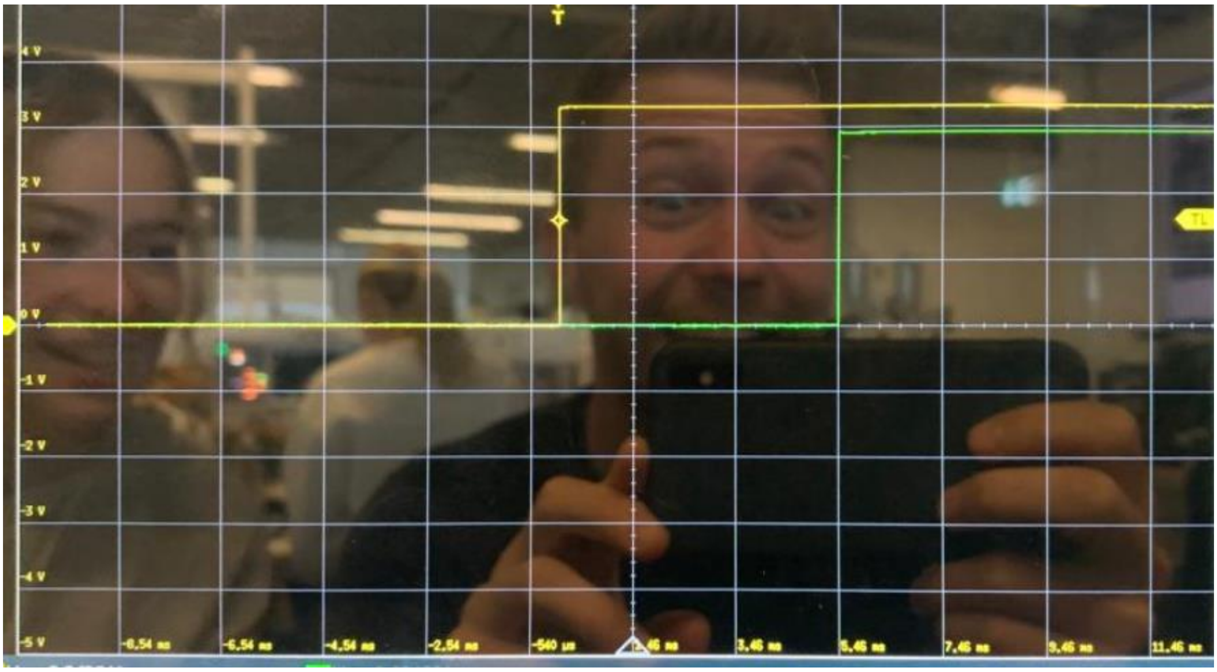
```
    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 414333 jonas     20   0    6692   3320   2928 S  23.8   0.8   0:09.71 flash.sh
     15 root      20   0       0      0      0 I   1.0   0.0   0:28.77 rcu_preempt
 415532 jonas     20   0    9840   3184   2576 R   1.0   0.7   0:00.29 top
    292 root      20   0       0      0      0 S   0.3   0.0   0:00.61 brcmf_wdog/mmc1:0001:1
 414331 root      20   0       0      0      0 I   0.3   0.0   0:00.05 kworker/1:0-events
      1 root      20   0  165216  10096   7412 S   0.0   2.4   0:03.95 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.04 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
```

Top with load:



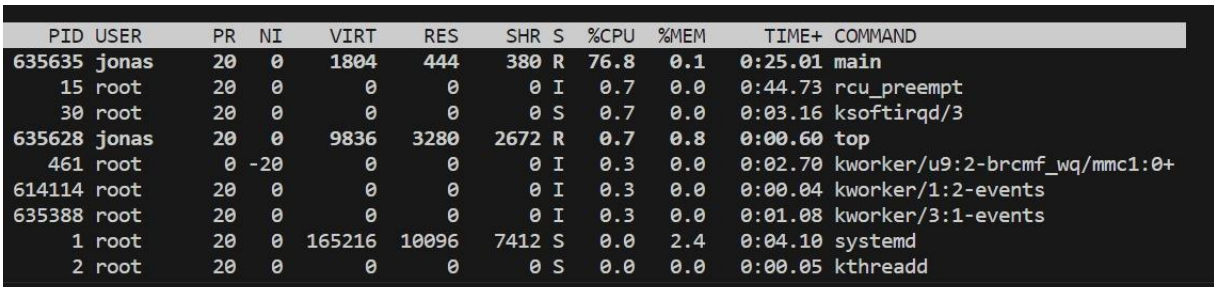
```
    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1333 jonas     20   0    6556    936    796 R  99.7   0.2   2:48.52 load.sh
   1339 jonas     20   0    6692   3172   2792 S  20.8   0.7   0:01.89 flash.sh
     15 root      20   0       0      0      0 I   1.0   0.0   0:00.41 rcu_preempt
   1048 root      20   0       0      0      0 I   0.3   0.0   0:00.17 kworker/0:1-events
   1052 root      20   0       0      0      0 I   0.3   0.0   0:00.10 kworker/3:0-events
   1327 jonas     20   0    9848   3140   2696 R   0.3   0.7   0:01.71 top
```

Oscilloscope picture:



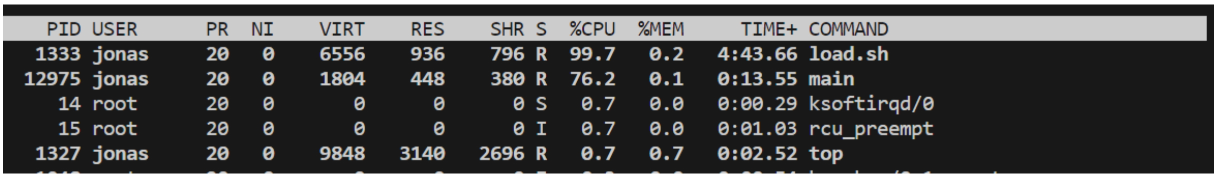
2. **Sysfs from C++ application polling pin status using a timed read()**
   Response Time: Ø 213.9 µs
   CPU Load: 76.8 %
   Results change if CPU fully loaded: Ø 205.3 µs

Top:



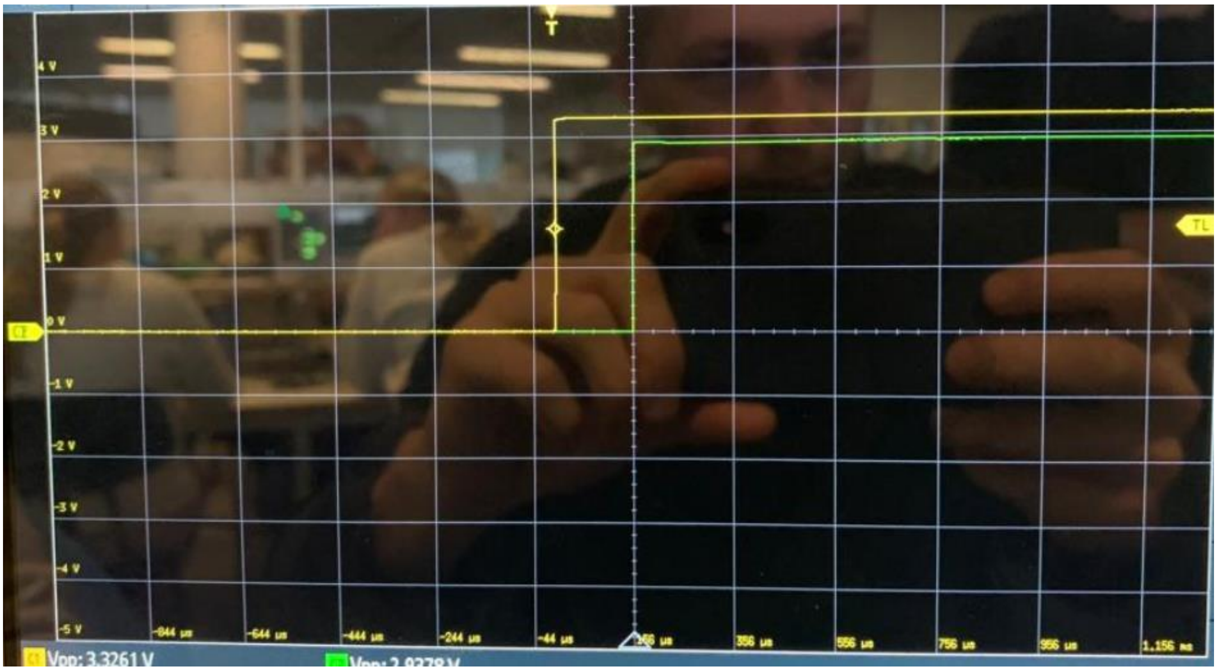
```
    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 635635 jonas     20   0    1804    444    380 R  76.8   0.1   0:25.01 main
     15 root      20   0       0      0      0 I   0.7   0.0   0:44.73 rcu_preempt
     30 root      20   0       0      0      0 S   0.7   0.0   0:03.16 ksoftirqd/3
 635628 jonas     20   0    9836   3280   2672 R   0.7   0.8   0:00.60 top
    461 root       0 -20       0      0      0 I   0.3   0.0   0:02.70 kworker/u9:2-brcmf_wq/mmc1:0+
 614114 root      20   0       0      0      0 I   0.3   0.0   0:00.04 kworker/1:2-events
 635388 root      20   0       0      0      0 I   0.3   0.0   0:01.08 kworker/3:1-events
      1 root      20   0  165216  10096   7412 S   0.0   2.4   0:04.10 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.05 kthreadd
```

Top with load:



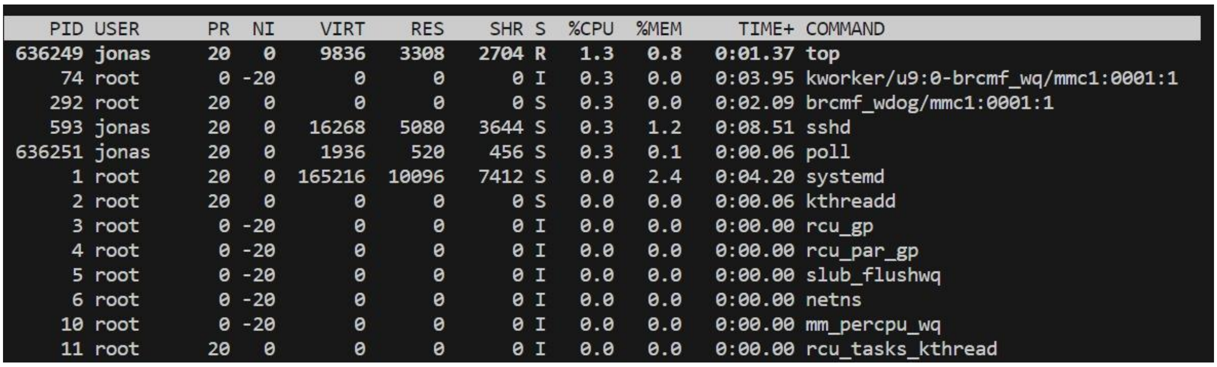
```
    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1333 jonas     20   0    6556    936    796 R  99.7   0.2   4:43.66 load.sh
  12975 jonas     20   0    1804    448    380 R  76.2   0.1   0:13.55 main
     14 root      20   0       0      0      0 S   0.7   0.0   0:00.29 ksoftirqd/0
     15 root      20   0       0      0      0 I   0.7   0.0   0:01.03 rcu_preempt
   1327 jonas     20   0    9848   3140   2696 R   0.7   0.7   0:02.52 top
```

Oscilloscope picture:



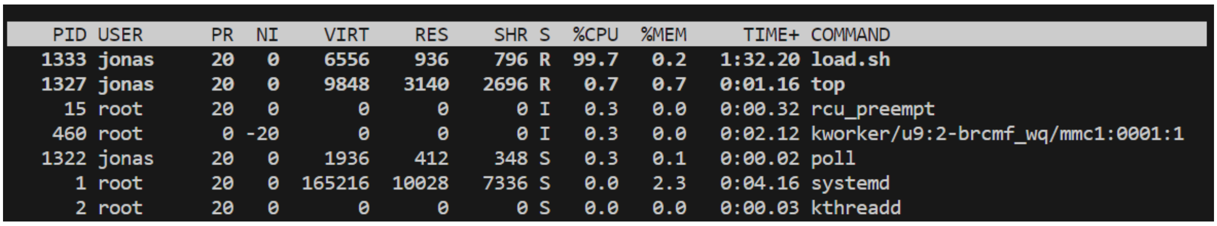
3. **Sysfs from C++ application using poll() function provided by kernel**
   Response Time: Ø 397.1 µs
   CPU Load: 0.3 %
   Results change if CPU fully loaded: Ø 262.7 µs

Top:



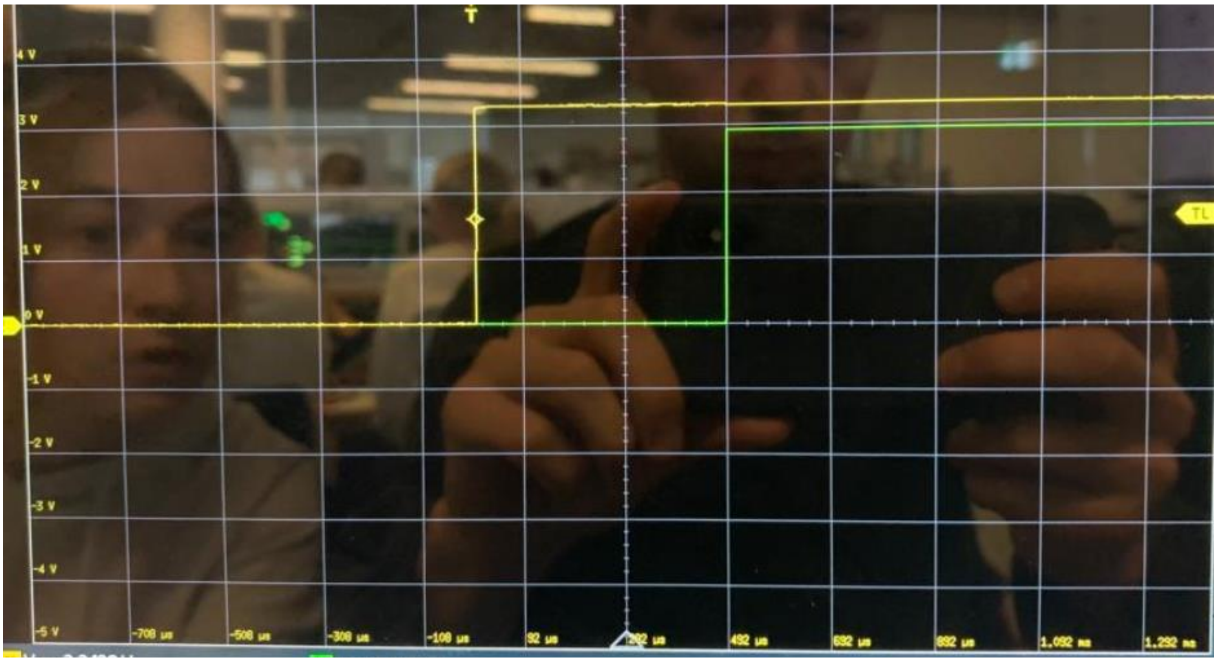
```
    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 636249 jonas     20   0    9836   3308   2704 R   1.3   0.8   0:01.37 top
     74 root       0 -20       0      0      0 I   0.3   0.0   0:03.95 kworker/u9:0-brcmf_wq/mmc1:0001:1
    292 root      20   0       0      0      0 S   0.3   0.0   0:02.09 brcmf_wdog/mmc1:0001:1
    593 jonas     20   0   16268   5080   3644 S   0.3   1.2   0:08.51 sshd
 636251 jonas     20   0    1936    520    456 S   0.3   0.1   0:00.06 poll
      1 root      20   0  165216  10096   7412 S   0.0   2.4   0:04.20 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.06 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
     11 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_kthread
```

Top with Load:



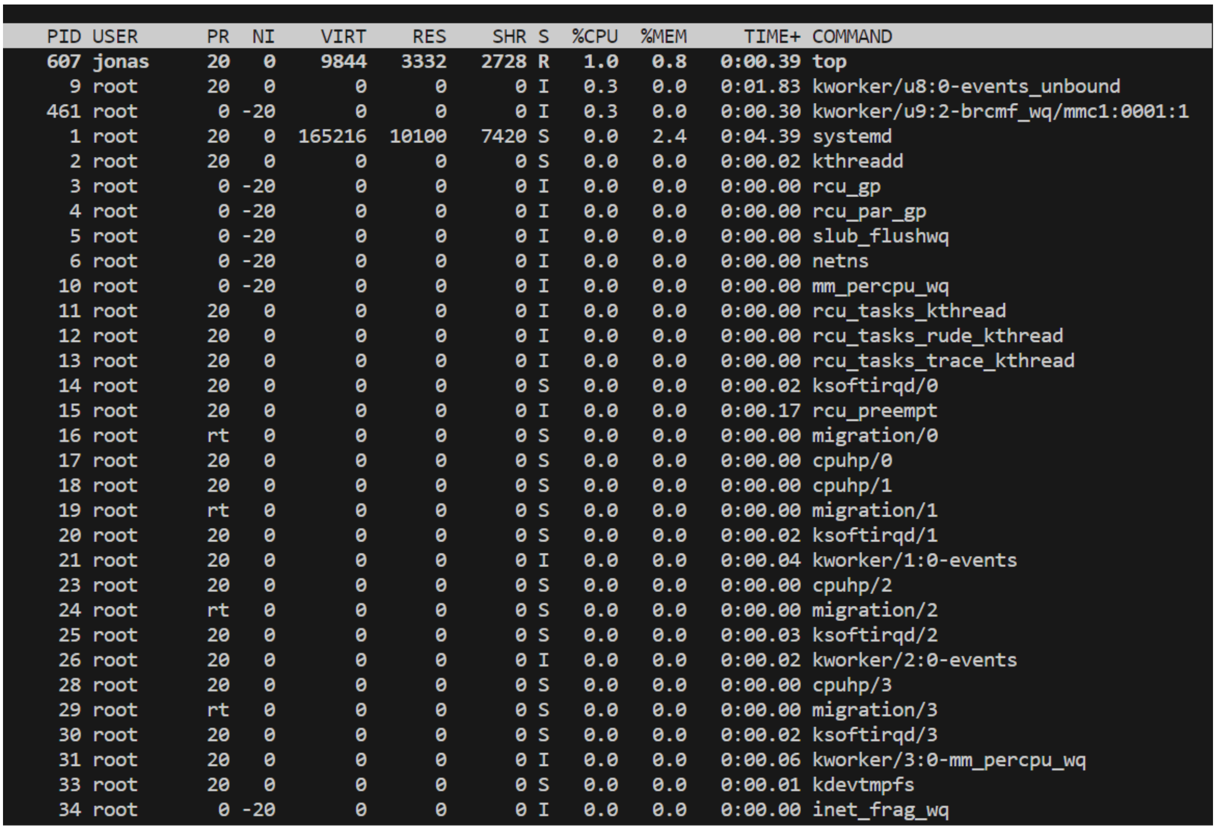
```
    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1333 jonas     20   0    6556    936    796 R  99.7   0.2   1:32.20 load.sh
   1327 jonas     20   0    9848   3140   2696 R   0.7   0.7   0:01.16 top
     15 root      20   0       0      0      0 I   0.3   0.0   0:00.32 rcu_preempt
    460 root       0 -20       0      0      0 I   0.3   0.0   0:02.12 kworker/u9:2-brcmf_wq/mmc1:0001:1
   1322 jonas     20   0    1936    412    348 S   0.3   0.1   0:00.02 poll
      1 root      20   0  165216  10028   7336 S   0.0   2.3   0:04.16 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.03 kthreadd
```

Oscilloscope picture:



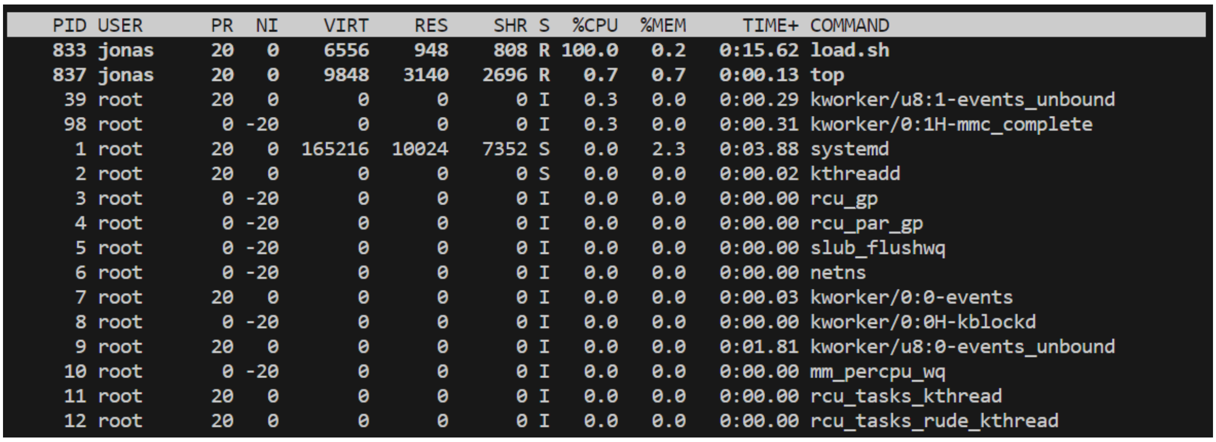
4. **Kernel modul using interrupts**
   Response Time: Ø 18.91 µs
   CPU Load: 0%
   Results change if CPU fully loaded: Ø 11.03 µs

   Top:



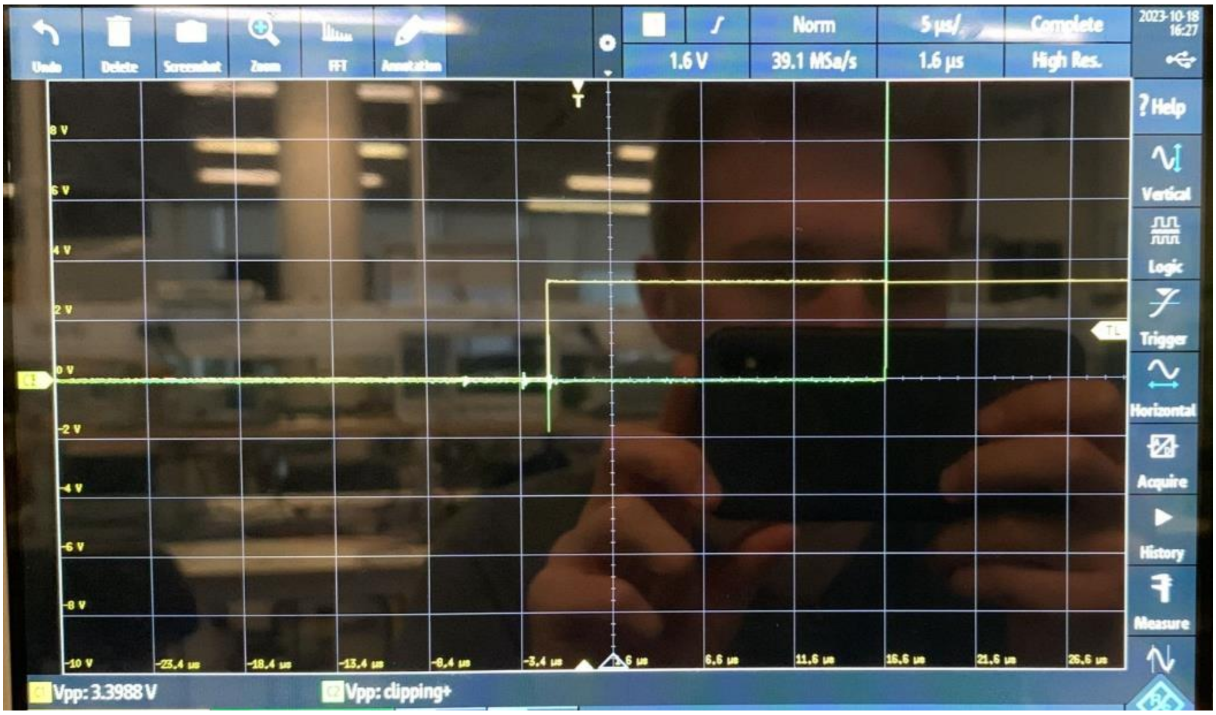
```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  607 jonas     20   0    9844   3332   2728 R   1.0   0.8   0:00.39 top
    9 root      20   0       0      0      0 I   0.3   0.0   0:01.83 kworker/u8:0-events_unbound
  461 root       0 -20       0      0      0 I   0.3   0.0   0:00.30 kworker/u9:2-brcmf_wq/mmc1:0001:1
    1 root      20   0  165216  10100   7420 S   0.0   2.4   0:04.39 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.02 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
    6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
   10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
   11 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_kthread
   12 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_rude_kthread
   13 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_trace_kthread
   14 root      20   0       0      0      0 S   0.0   0.0   0:00.02 ksoftirqd/0
   15 root      20   0       0      0      0 I   0.0   0.0   0:00.17 rcu_preempt
   16 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migration/0
   17 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   18 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
   19 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migration/1
   20 root      20   0       0      0      0 S   0.0   0.0   0:00.02 ksoftirqd/1
   21 root      20   0       0      0      0 I   0.0   0.0   0:00.04 kworker/1:0-events
   23 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/2
   24 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migration/2
   25 root      20   0       0      0      0 S   0.0   0.0   0:00.03 ksoftirqd/2
   26 root      20   0       0      0      0 I   0.0   0.0   0:00.02 kworker/2:0-events
   28 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/3
   29 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migration/3
   30 root      20   0       0      0      0 S   0.0   0.0   0:00.02 ksoftirqd/3
   31 root      20   0       0      0      0 I   0.0   0.0   0:00.06 kworker/3:0-mm_percpu_wq
   33 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kdevtmpfs
   34 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 inet_frag_wq
```

   Top with load:



```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  833 jonas     20   0    6556    948    808 R 100.0   0.2   0:15.62 load.sh
  837 jonas     20   0    9848   3140   2696 R   0.7   0.7   0:00.13 top
   39 root      20   0       0      0      0 I   0.3   0.0   0:00.29 kworker/u8:1-events_unbound
   98 root       0 -20       0      0      0 I   0.3   0.0   0:00.31 kworker/0:1H-mmc_complete
    1 root      20   0  165216  10024   7352 S   0.0   2.3   0:03.88 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.02 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
    6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
    7 root      20   0       0      0      0 I   0.0   0.0   0:00.03 kworker/0:0-events
    8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-kblockd
    9 root      20   0       0      0      0 I   0.0   0.0   0:01.81 kworker/u8:0-events_unbound
   10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
   11 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_kthread
   12 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_rude_kthread
```

Oscilloscope picture:



Vpp: 3.3988 V

Vpp: clipping+

- **Which mechanism would suffice for counting encoder pulses?**
  The encoder period is 664.45 µs, so every mechanism except from the *Sysfs from shell script* is valid for the counting.

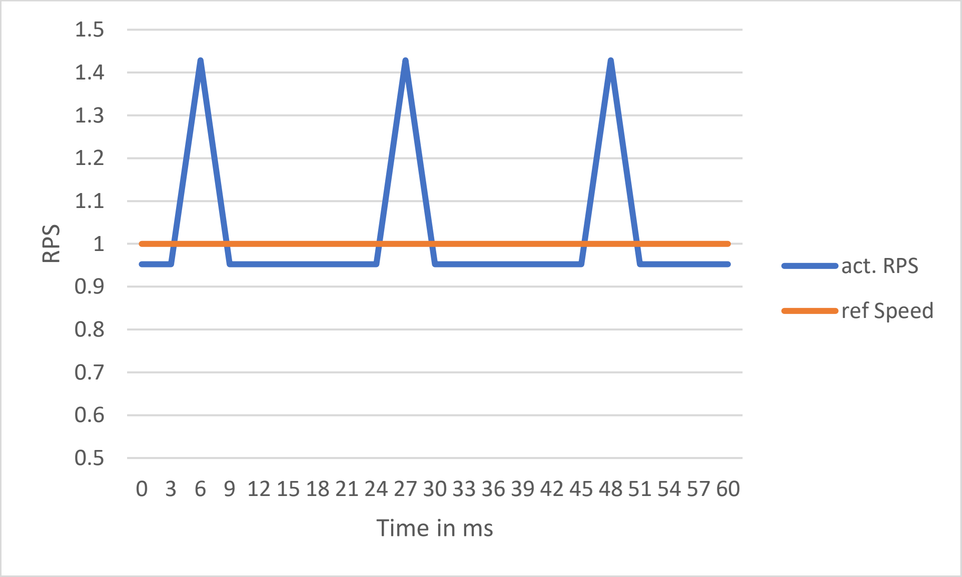
Project 4: Kernel space encoder driver

Part 2:

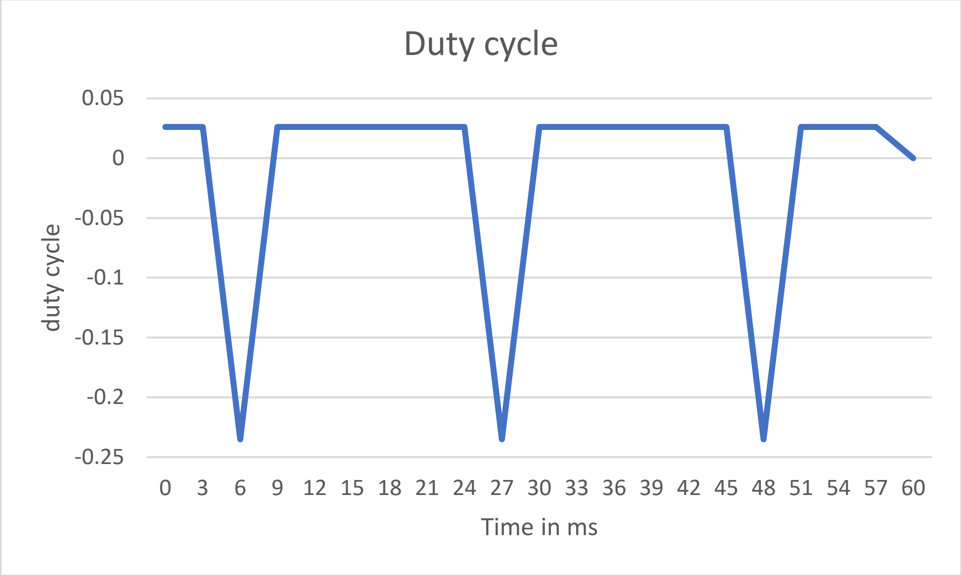
Speed print:

```
ref. rps: 1.000000, act. rps: 0.952381, duty: 0.026125
ref. rps: 1.000000, act. rps: 0.952381, duty: 0.026125
ref. rps: 1.000000, act. rps: 1.428571, duty: -0.235129
ref. rps: 1.000000, act. rps: 0.952381, duty: 0.026125
ref. rps: 1.000000, act. rps: 0.952381, duty: 0.026125
ref. rps: 1.000000, act. rps: 0.952381, duty: 0.026125
ref. rps: 1.000000, act. rps: 0.952381, duty: 0.026125
ref. rps: 1.000000, act. rps: 0.952381, duty: 0.026125
ref. rps: 1.000000, act. rps: 0.952381, duty: 0.026125
ref. rps: 1.000000, act. rps: 1.428571, duty: -0.235129
ref. rps: 1.000000, act. rps: 0.952381, duty: 0.026125
ref. rps: 1.000000, act. rps: 0.952381, duty: 0.026125
ref. rps: 1.000000, act. rps: 0.952381, duty: 0.026125
ref. rps: 1.000000, act. rps: 0.952381, duty: 0.026125
ref. rps: 1.000000, act. rps: 0.952381, duty: 0.026125
ref. rps: 1.000000, act. rps: 0.952381, duty: 0.026125
ref. rps: 1.000000, act. rps: 1.428571, duty: -0.235129
ref. rps: 1.000000, act. rps: 0.952381, duty: 0.026125
ref. rps: 1.000000, act. rps: 0.952381, duty: 0.026125
ref. rps: 1.000000, act. rps: 0.952381, duty: 0.026125
ref. rps: 1.000000, act. rps: 0.952381, duty: 0.026125
```

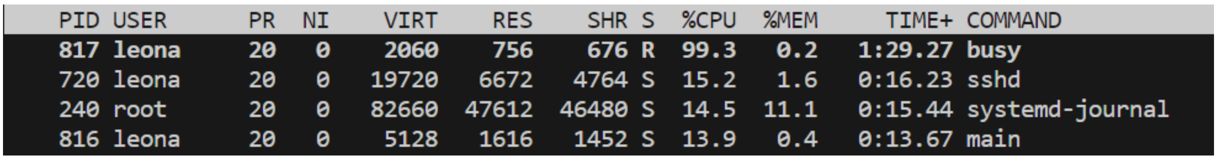
Control rate:



Duty Cycle:



In the diagram above the jitter (peaks downwards) are visible when using the usleep() function for timing of the control rate.



```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  817 leona     20   0    2060    756    676 R  99.3   0.2   1:29.27 busy
  720 leona     20   0   19720   6672   4764 S  15.2   1.6   0:16.23 sshd
  240 root      20   0   82660  47612  46480 S  14.5  11.1   0:15.44 systemd-journal
  816 leona     20   0    5128   1616   1452 S  13.9   0.4   0:13.67 main
```

The results weren`t affected in our case. In the table it is visible that the injected busy task takes up 99.3% CPU load. However, the controller worked fine.

Injecting the busy task before starting the main task to control the motor would result in the motor not spinning at all.

GitHub: https://github.com/Marlenexyz/EMBE-Group

YouTube: https://www.youtube.com/watch?v=buABB3gQtAQ