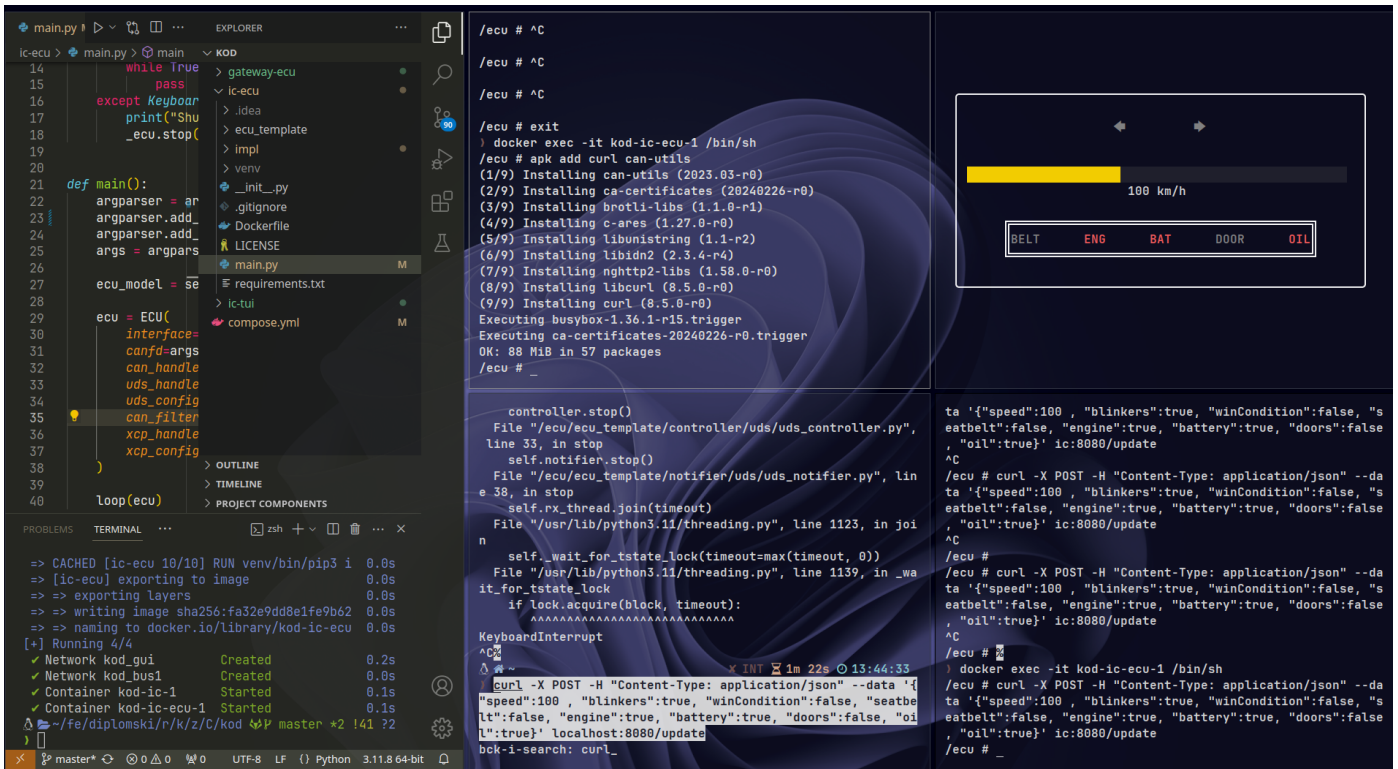


Preostalo

- driver
 - ☐ perzistentnost
 - ☐ upakirat za linux distribucije
 - ☐ github actions za deploy na dockerhub
- zadaci
 - napraviti repo koji builda zadatke za dockerhub automatski s github actionsima
 - ☐ XCP - dump memory caring caribou
 - ☒ XCP predložak pripremljen
 - ☐ treba dodati parsiranje XCP poruka
 - ~~UDS Authentication~~ - ne vidim potencijalni zadatak
 - ☐ CAN zadatak s dva ECU-a i GUI-jem
 - uds routine control bi se mogao koristiti za ovo
 - upalit zmigavce
 - postići veću brzinu nego maksimalnu
 - kad uspije, postaviti flag na sabirnicu
 - ☐ UDS security access MITM između dvije sabirnice
 - ~~UDS zadatak s GUI-jem, routine control~~ duplikat gore
- ecu_template
 - ☐ dodati template za UDS SA
 - ☐ dodati glavni program za slanje repetitivnih poruka umjesto CAN_BCM-a?
 - ☐ README
 - čemu svaka od impl datoteka služi
 - kompatibilnost s caring caribouom
 - primjeri kako neke stvari implementirati
 - ☐ tagovi za razne varijante templateova
 - ☐ XCP
 - ☐ DoIP
 - ☐ parsiranje CAN signala po DBC-u
- dodatni programi containeri:
 - ☐ gw program za povezivanje više dockercan mreža?
 - ☐ program za cannelloni udaljeni pristup
 - ☐ program za generiranje "smeca" na sabirnici
- ☐ skripta za generiranje docker composeova

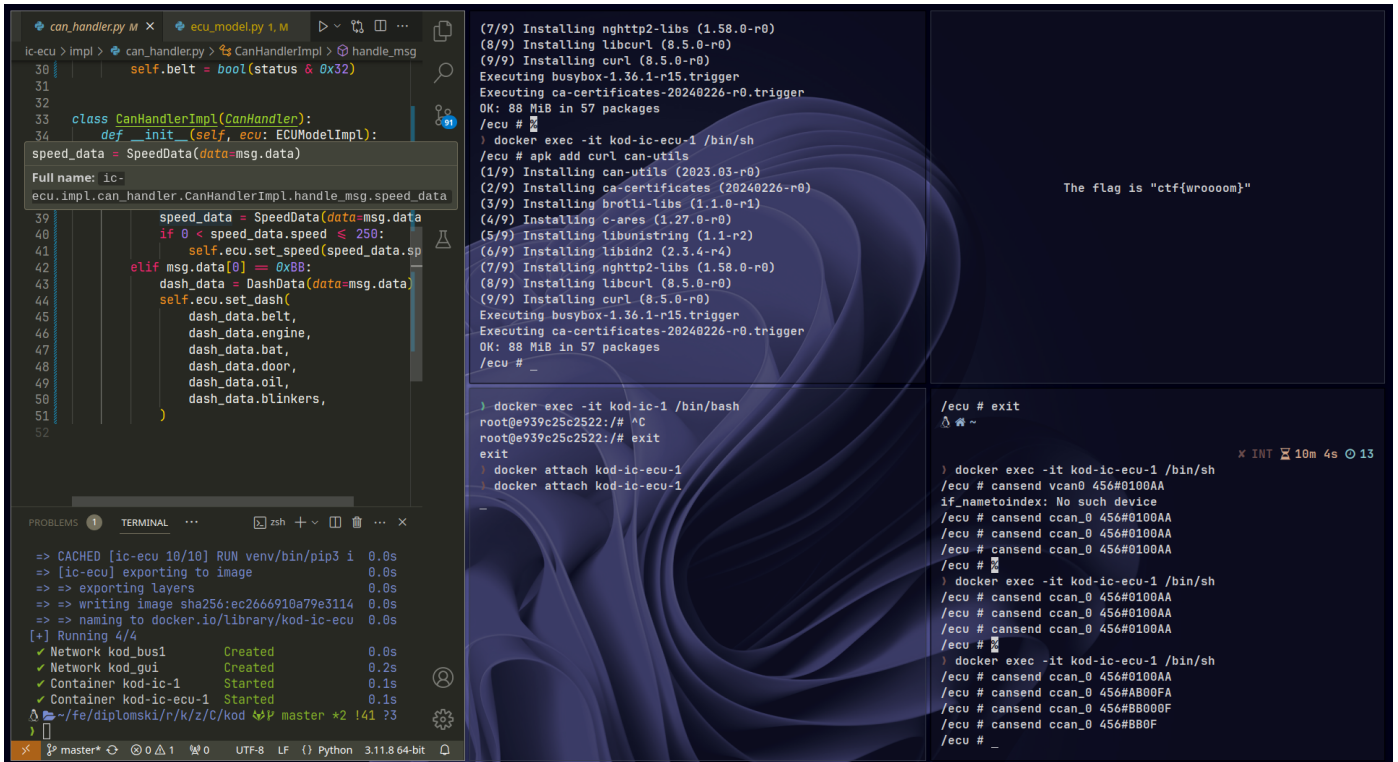
2.6.

<https://forums.docker.com/t/compose-how-to-connect-container-to-both-custom-bridge-and-host-network-basler-cameras-in-docker-containers/139391>



Treba ugasiti listener na 8080 kad se dogodi ssh disconnect, kako bi iduca instanca mogla ponovno slusati na istom portu

Uspjesno rjesen zadatak:



```

> candump vcan0
vcan0 100 [5] 04 31 00 00 00
vcan0 101 [5] 04 71 00 00 00
vcan0 456 [3] AB 00 00
vcan0 456 [3] AB 00 19
vcan0 456 [3] AB 00 32
vcan0 456 [3] AB 00 48
vcan0 456 [3] AB 00 64
vcan0 456 [3] AB 00 70
vcan0 456 [3] AB 00 96
vcan0 456 [3] AB 00 AF
vcan0 456 [3] AB 00 C8
vcan0 456 [3] AB 00 E1
vcan0 456 [3] AB 00 FA
vcan0 456 [2] BB 01
vcan0 456 [2] BB 02
vcan0 456 [2] BB 04
vcan0 456 [2] BB 08
vcan0 456 [2] BB 10
vcan0 456 [2] BB 20

>>> s.sr1(UDS()/UDS_RC())
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
<UDS service=RoutineControlPositiveResponse |<UDS_RCPR routineControlType=ISOSAEReserved r
outineIdentifier=0x0 |>>
>>> -

```

Drugi ECU koji služi za generiranje primjera prometa koji treba reverzati, a pokreće se pomoću UDS_RC paketa

```

load_contrib("isotp")
conf.contribs['ISOTP'] = {'use-can-isotp-kernel-module': True}
load_contrib("automotive.uds")
s = ISOTPNativeSocket(iface="ctf_can", tx_id=0x100, rx_id=0x101, basecls=UDS)

```