

11.4.

<https://docs.python.org/3/library/typing.html>

<https://refactoring.guru/design-patterns/command>

- obrazac naredba za implementiranje interakcije s GUI-jem
 - u ovom slucaju "klijent" je dio koda koji se bavi primanjem poruka i stanjem ECU-a, a "receiver" je GUI kod
- ECU neka drzi svoje stanje i daje ga na modifikaciju u Callbackove
 - provjerava svoje stanje i prema tome mijenja gui

12.4.

37C3 - Breaking "DRM" in Polish trains

<https://www.youtube.com/watch?v=XrlrbfGZo2k>

- ideja za ctf zadatak

```
class Notifier:
    def __init__(
        self, bus: bus,
        timeout: timeout,
        loop: loop,

        #: Exception raised in thread
        exception: Optional[Exception] = None,

        self._running = True,
        self._lock = threading.Lock(),

        self._readers: List[Union[int, threading.Thread]] = []
        buses = self.bus if isinstance(self.bus, list) else [self.bus]
        for each_bus in buses:
            self.add_bus(each_bus)
```

can.Notifier vec koristi lock, callbackovi se ne mogu izvorsavat istovremeno

13.4.

ISO-TP

<https://can-isotp.readthedocs.io/en/latest/isotp/implementation.html#transport-layer>

Ne podrzava asinkrono citanje poruka, rjesenja:

- napisati svoj wrapper oko isotp socketa koji podrzava asinkrono citanje poruka
- forkati python-can, dodati podrsku za isotp sockete
 - korisiti can-isotp kao wrapper za socket API ?
- napraviti Notifier za can-isotp

repetitivne poruke

- dodati CAN_BCM
- dodati glavnu petlju za ECU?

14.4.

Izrada svog Notifiera za can-isotp

- cini se da je isotp library za python previse spor da bi radio sa socketcanom
- radi povrh can notifiera iz pythoncan librarija sto ga vjerojatno dodatno usporava
- stvara errore kod normalnih paketa poslanih isotpsendom, koje isotprecv inace normalno moze primiti:

```
class Notifier:
    def __init__(self, transport_layer: isotp.TransportLayer):
        self._rx_thread = None
        self._running = False
        if transport_layer is None:
            raise TypeError
        self.transport_layer = transport_layer
        self.listeners = []

    def _rx_thread(self):
        while self._running:
            if data := self.transport_layer.recv(lock=False, timeout=5):
                for listener in self.listeners:
                    listener.on_message_received(data)

    def start(self):
        self._running = True
        self._rx_thread = threading.Thread(target=self._rx_thread, name="isotp_notifier")
        self._rx_thread.daemon = True
        self._rx_thread.start()
```

```
> echo "09 02 05 06 07 08 08 07 07 09 02 05 06 07 08 08 07 07 09 02 05 06 07 08 08 07 07 09 02 05 06 07 08 08 07 07" | isotpsend -s 100 -d 101 vcan0
```

```
Received isotp data: 09020506070808
Received isotp data: 09020506070808
Received isotp data: 09020506070808
Reception of CONSECUTIVE_FRAME timed out.
Received a ConsecutiveFrame with wrong SequenceNumber. Expecting 0x01, Received 0x02
Received a ConsecutiveFrame while reception was idle. Ignoring
Reception of CONSECUTIVE_FRAME timed out.
Received a ConsecutiveFrame with wrong SequenceNumber. Expecting 0x02, Received 0x04
```

```
> echo "09 02 05 06 07 08 08 07 07 09 02 05 06 07 08 08 07 07 09 02 05 06 07 08 08 07 07 09 02 05 06 07 08 08 07 07" | isotpsend -s 100 -d 101 vcan0
```

```
Reception of CONSECUTIVE_FRAME timed out.
Received a ConsecutiveFrame with wrong SequenceNumber. Expecting 0x01, Received 0x02
Received a ConsecutiveFrame while reception was idle. Ignoring
Reception of CONSECUTIVE_FRAME timed out.
Received a ConsecutiveFrame with wrong SequenceNumber. Expecting 0x02, Received 0x04
Received a ConsecutiveFrame with wrong SequenceNumber. Expecting 0x02, Received 0x05
```

```
> isotprecv -d 100 -s 101 vcan0
09 02 05 06 07 08 08 07 07 09 02 05 06 07 08 08 07 07 09 02 05 06 07 08 08 07 07 09 02 05 06 07 08 08 07 07
```

Umjesto python-can sloja, prepravio sam notifier da polla isotp sockete izravno sto je rjesilo problem