

# CAN We Talk Python?

*Talking to your vehicle over the CAN bus  
with Python*

@shnewto

**CAN?**



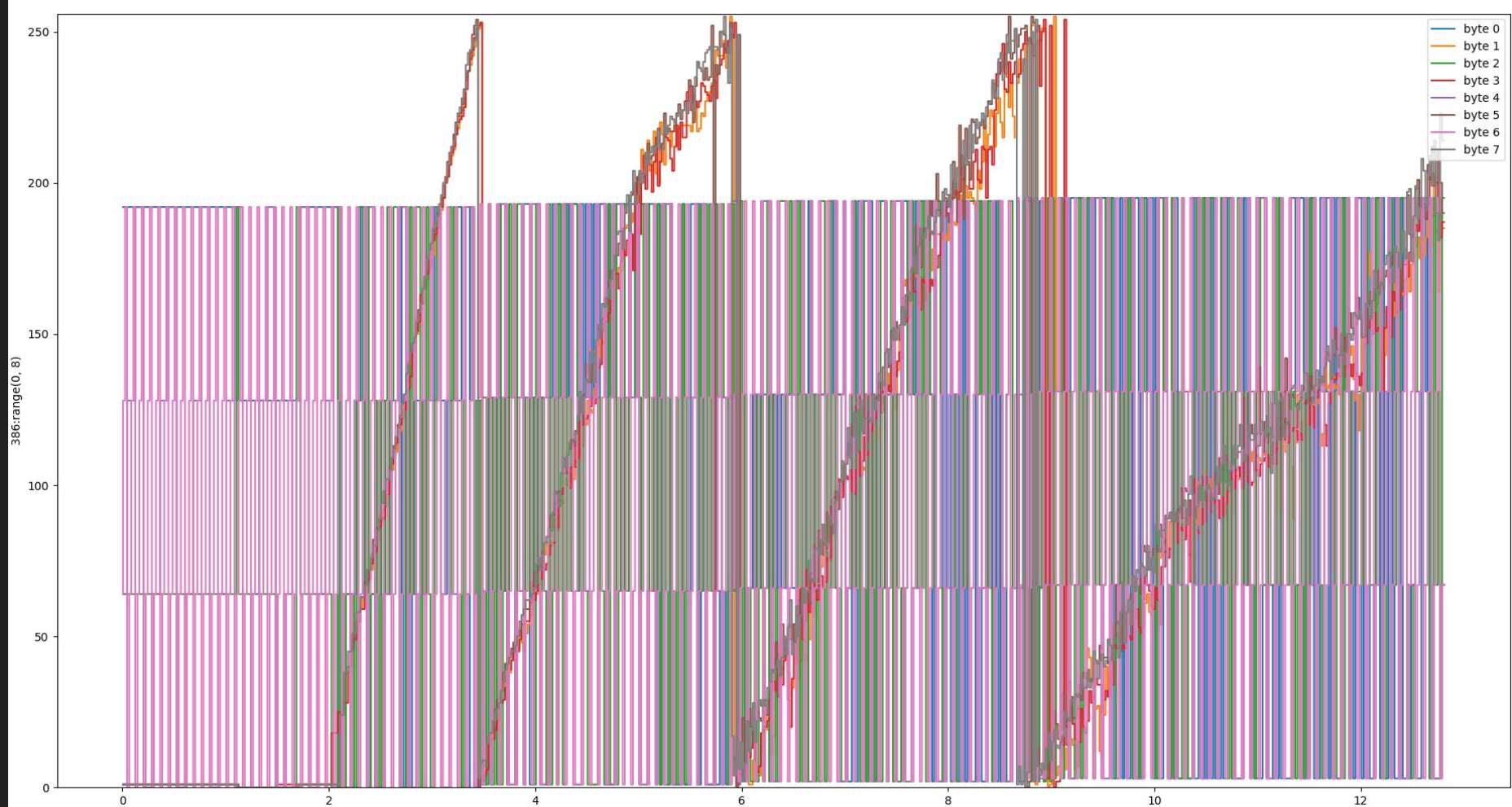
# The standard CAN frame (mostly)

ID: 0x123

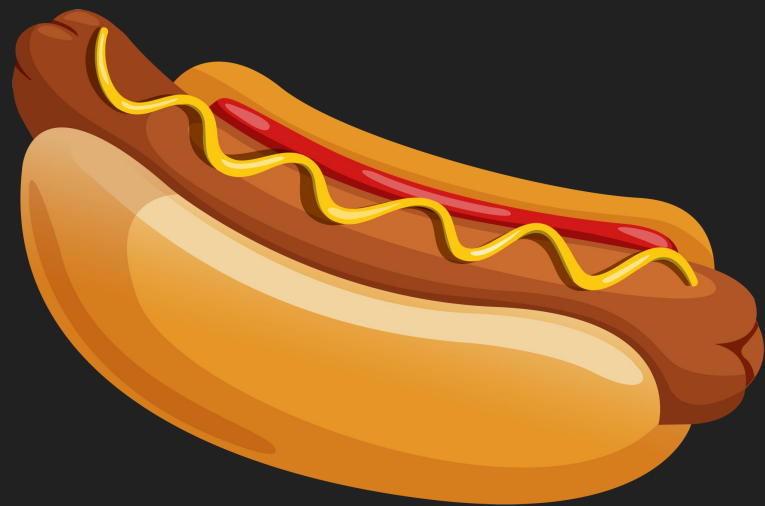
DLC: 8

Data: 8 bytes / 64 bits

[0]	0x02	0x04	0x06	0x08
[1]	0x0A	0x0C	0x0E	0x10
[2]	0x12	0x14	0x16	0x18
[3]	0x1A	0x1C	0x1E	0x20
[4]	0x22	0x24	0x26	0x28
[5]	0x2A	0x2C	0x2E	0x30
[6]	0x32	0x34	0x36	0x38
[7]	0x3A	0x3C	0x3E	0x40



```
byte1 = (data[offset + 1] & 0x0F) << 8  
byte0 = data[offset]  
value = byte1 | byte0  
wheel_speed = value / 10.0
```



# The OBD-II Port





(My 2006 Hyundai Elantra)

**A LOT of chatter!**

**can\_we\_talk.py**

[github.com/shnewto/can-we-talk](https://github.com/shnewto/can-we-talk)

```
1 # pip install python-can
2 import can
3
4 bus = can.interface.Bus(
5     bustype='socketcan_native', channel='can0', bitrate=500000)
6
7 pid = 0x2F # fuel tank level
8
9 query_frame = [0x02, 0x01, pid, 0x55, 0x55, 0x55, 0x55, 0x55]
10
11 query_arbitration_id = 0x7DF # Defined by OBD-II standard
12
13 msg = can.Message(
14     arbitration_id=query_arbitration_id, data=query_frame, extended_id=False)
15
16 bus.send(msg)
17
18 reply_arbitration_id = 0x7E8 # Defined by OBD-II standard
19
20 reply = bus.recv(timeout)
21
22 if (reply is not None) and (reply.arbitration_id == reply_arbitration_id):
23     print(reply.data)
```

**oscc-check.py**

[github.com/PolySync/oscc-check](https://github.com/PolySync/oscc-check)

```
import osccan
```

```
bus = osccan.CanBus( )
```

```
report = bus.check_wheel_speed( )
```

```
print(report.value)
```

# The CAN bus and your car

**Questions?**



# Resources

[sheas.blog/talks](https://sheas.blog/talks) (slide deck)

Code:

- [github.com/shnewto/can-we-talk](https://github.com/shnewto/can-we-talk) OBD-II reader
- [github.com/PolySync/oscc-check](https://github.com/PolySync/oscc-check) Vehicle actuation and status reports

Twitter: @shnewto

Blog: sheas.blog

GitHub: shnewto

Email: shnewto@gmail.com