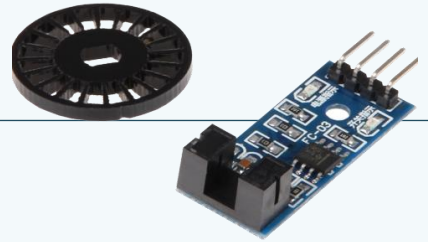
sea:me
DES_PROJECT_2
Instrument Cluster

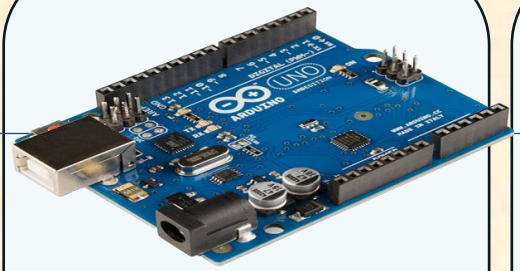
Team2 _ Hokyung-Park, Seungwoo-L , Ilgizar Khuzin

Index

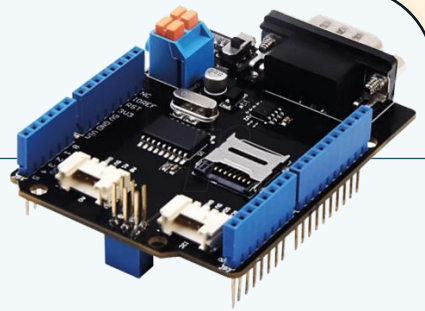
SpecIfIcatIon - Hardware / Software



- Speed Sensor LM 393
- Measure Speed

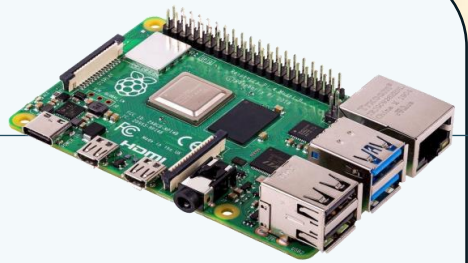


- Arduino Uno
- Receive data from sensor



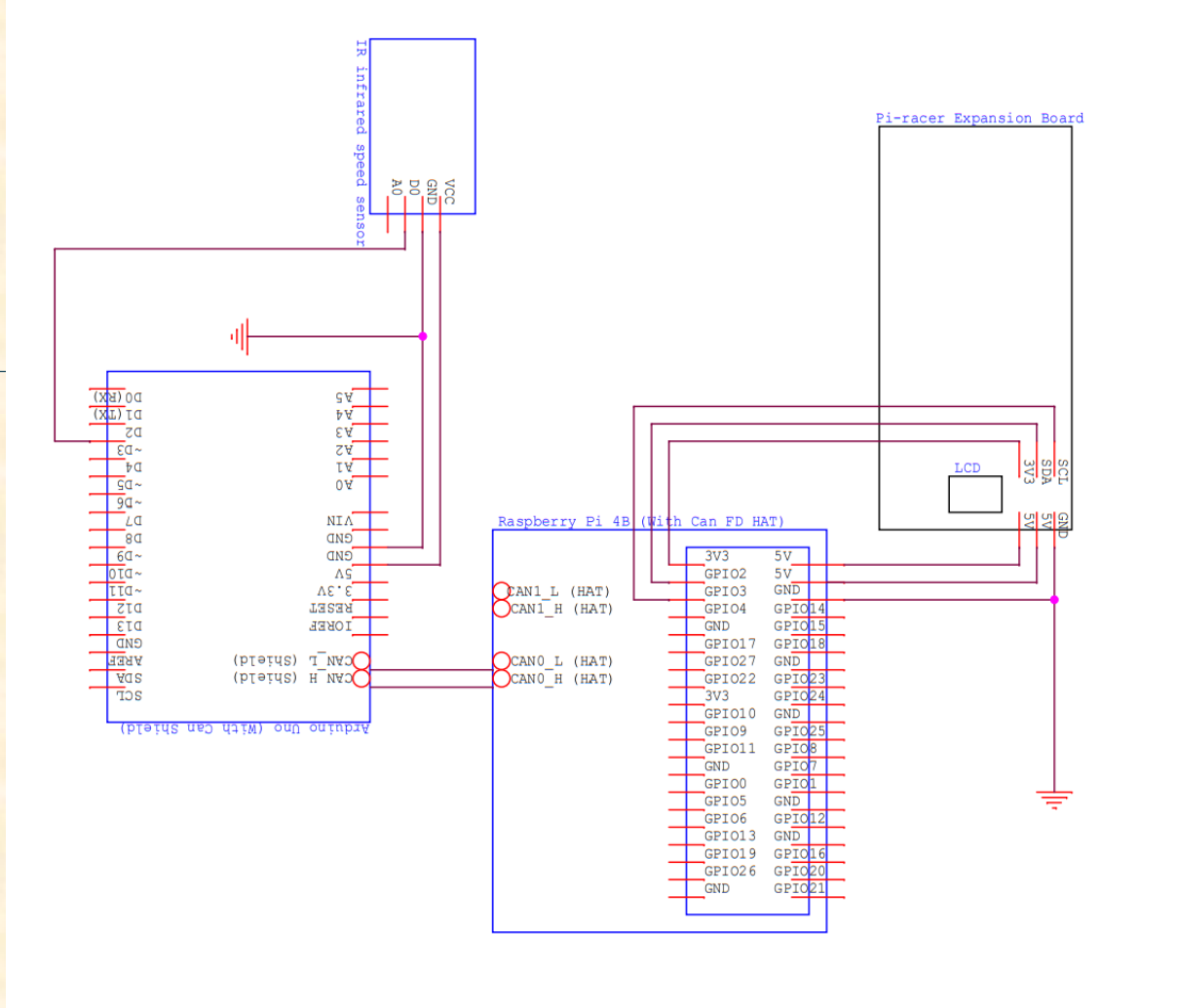
- CAN – BUS Shield V 2.0
- Send data via CAN

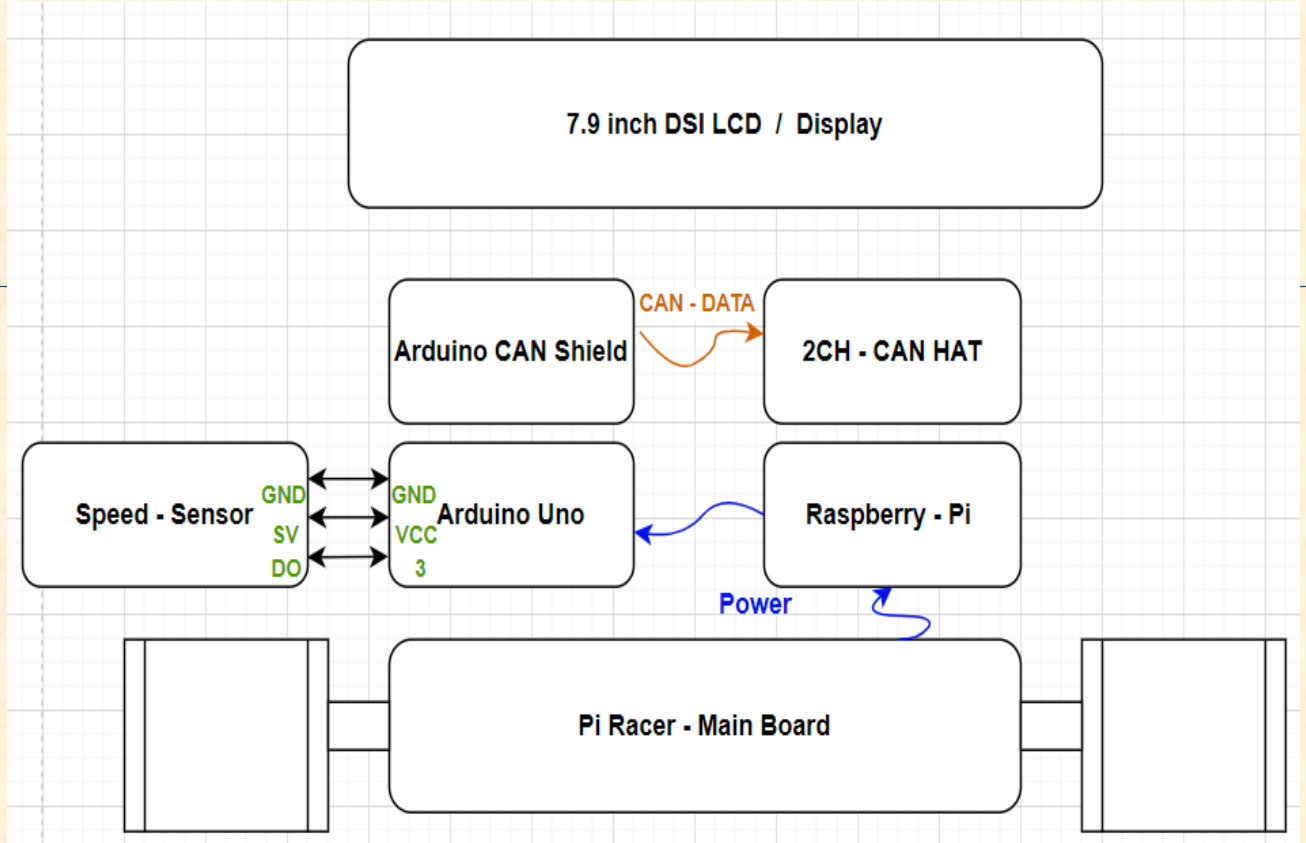
- 2CH – CAN HAT
- Receive data via CAN



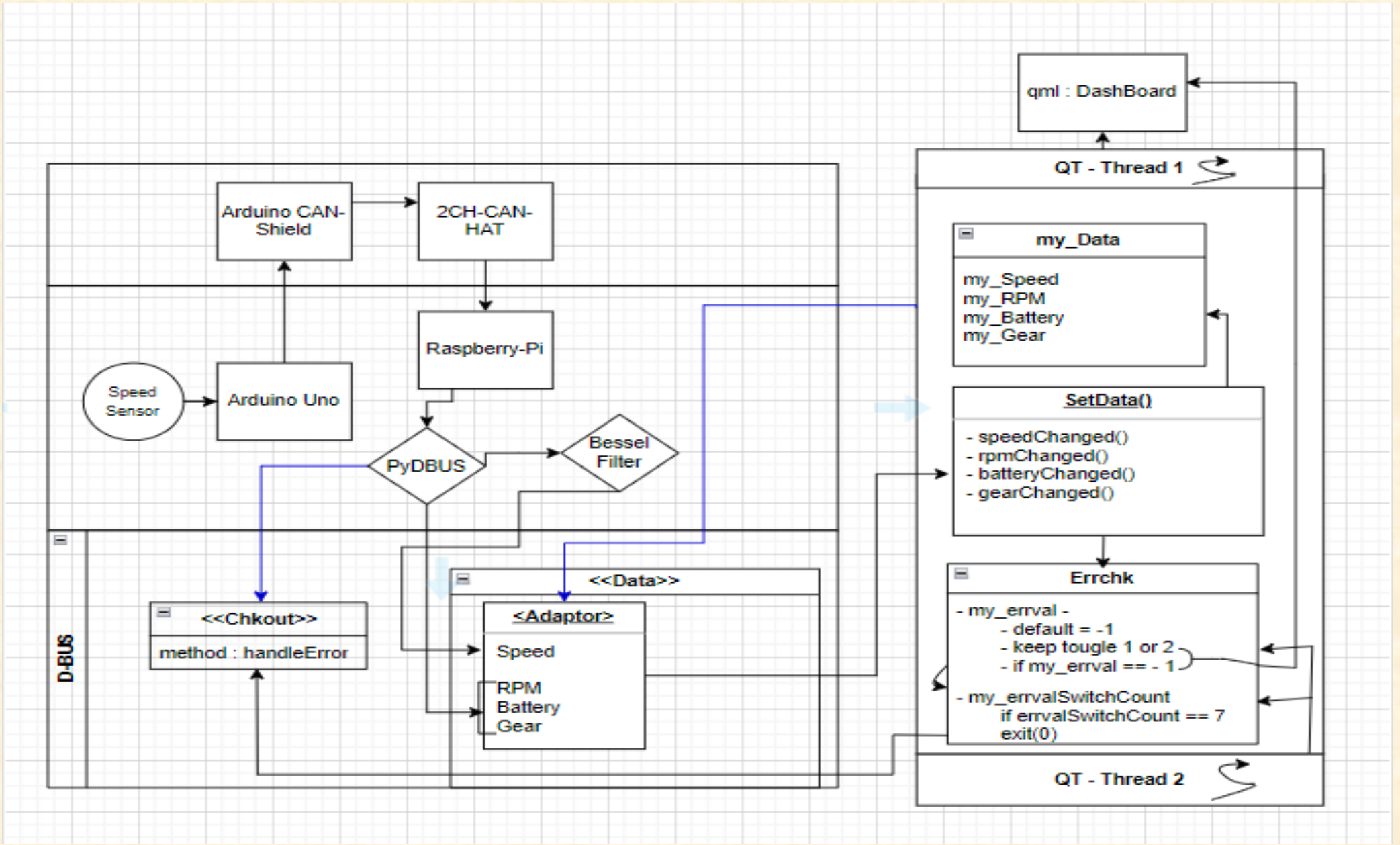
- Raspberry – Pi 4B
- Data Processing & Launch all exec( )

Hardware connection

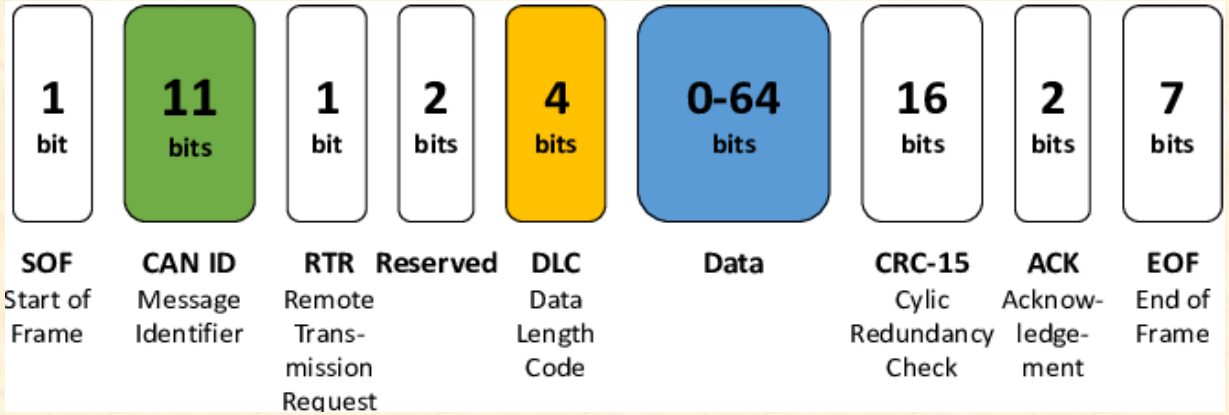


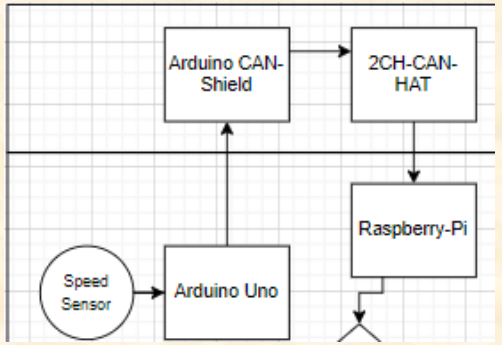


CodeFlow


qml : DashBoard
QT - Thread 1
Arduino CAN-Shield
2CH-CAN-HAT
Raspberry-Pi
Speed Sensor
Arduino Uno
PyDBUS
Bessel Filter
my_Data
my_Speed
my_RPM
my_Battery
my_Gear
SetData()
- speedChanged()
- rpmChanged()
- batteryChanged()
- gearChanged()
Errchk
- my_errval -
- default = -1
- keep tougle 1 or 2
- if my_errval == - 1
- my_errvalSwitchCount
if errvalSwitchCount == 7
exit(0)
D-BUS
<<Chkout>>
method : handleError
<<Data>>
<Adaptor>
Speed
RPM
Battery
Gear
QT - Thread 2

Part.2 - Get/Send Data In CAN



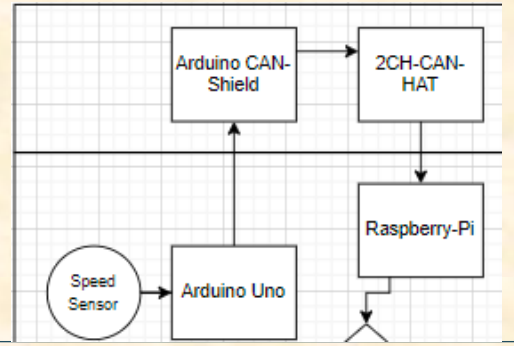


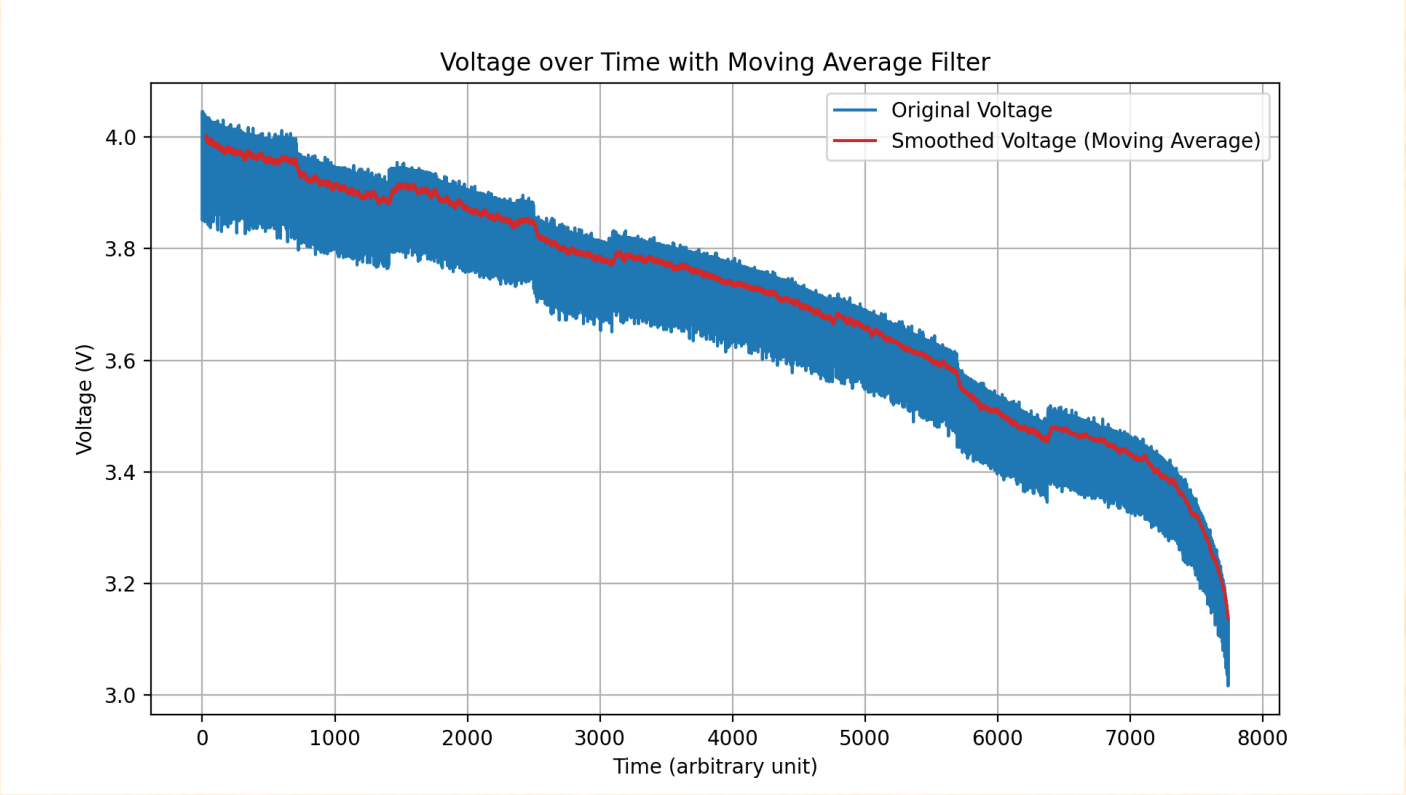

```
byte rpmData[8];
  float tempRPM = RPM_w;
  float tempSpeed = speed_mps;
  memcpy(rpmData, &tempRPM, 4);
  memcpy(rpmData + 4, &tempSpeed, 4);

  CAN0.sendMsgBuf(0x100, 0, 8, rpmData);
```

CAN0.sendMsgBuf(ID , RTR , Data_len , Data)

ID : 0x100
RTR : 0 (0 = send , 1 = receive)
Data_len = 8 bytes
Data = rpmdata

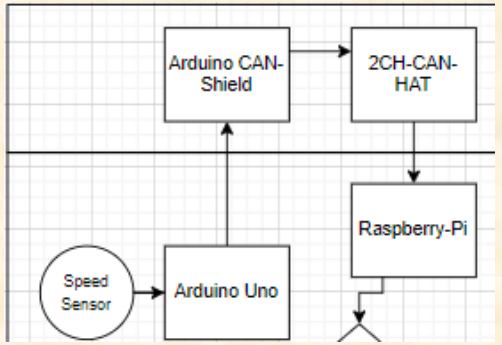
Part.2 - Pydbus -Get Voltage

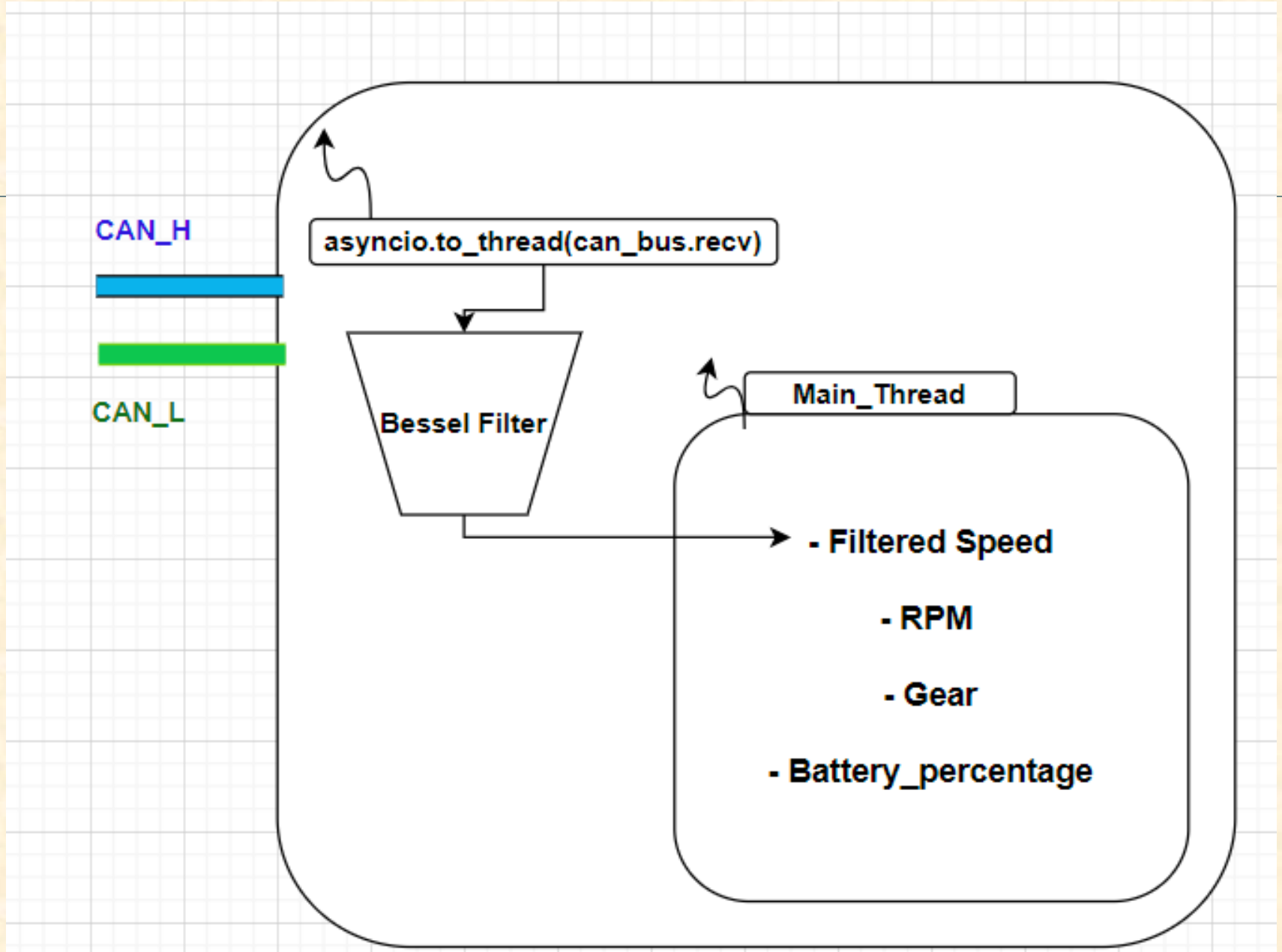




- Get Battery Voltage using i2c
- Check Official datasheet
- Experiment Minimum Voltage (3.1V)
- Convert to Equation and apply filter

Part.2 - Send Data to D–BUS In asynchronously



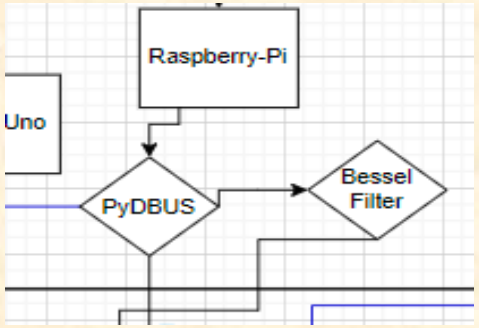


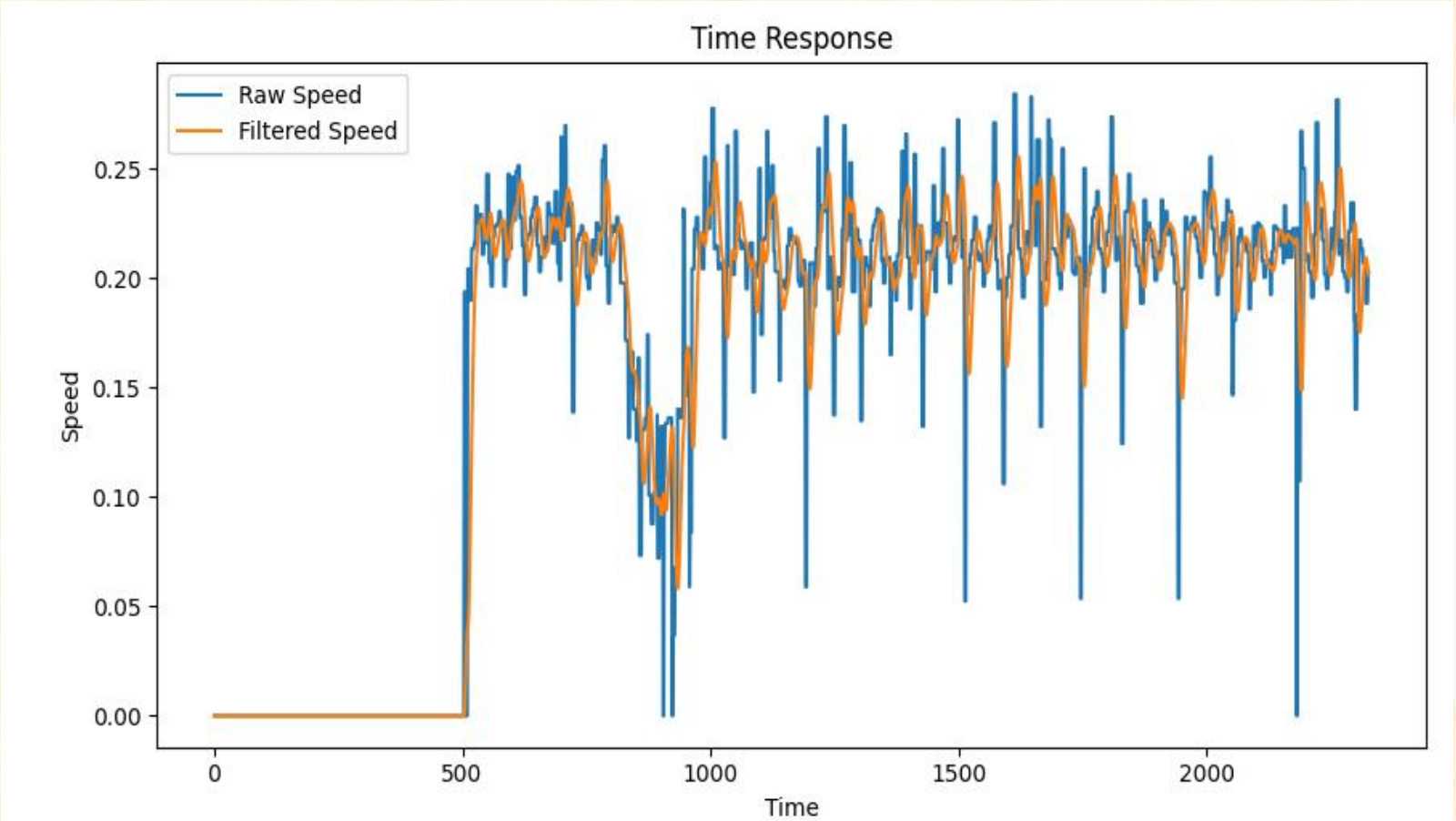
- Return Data Asynchronously

- Using Bessel Filter for reduce noise

- Update Data in D-bus with Main thread

Part.2–Data Processing - Bessel Filter





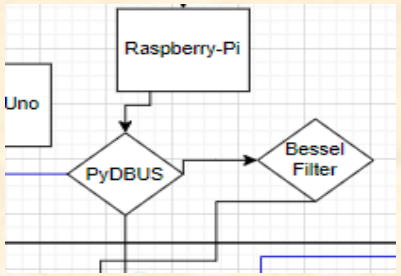
<Bessel Filter>

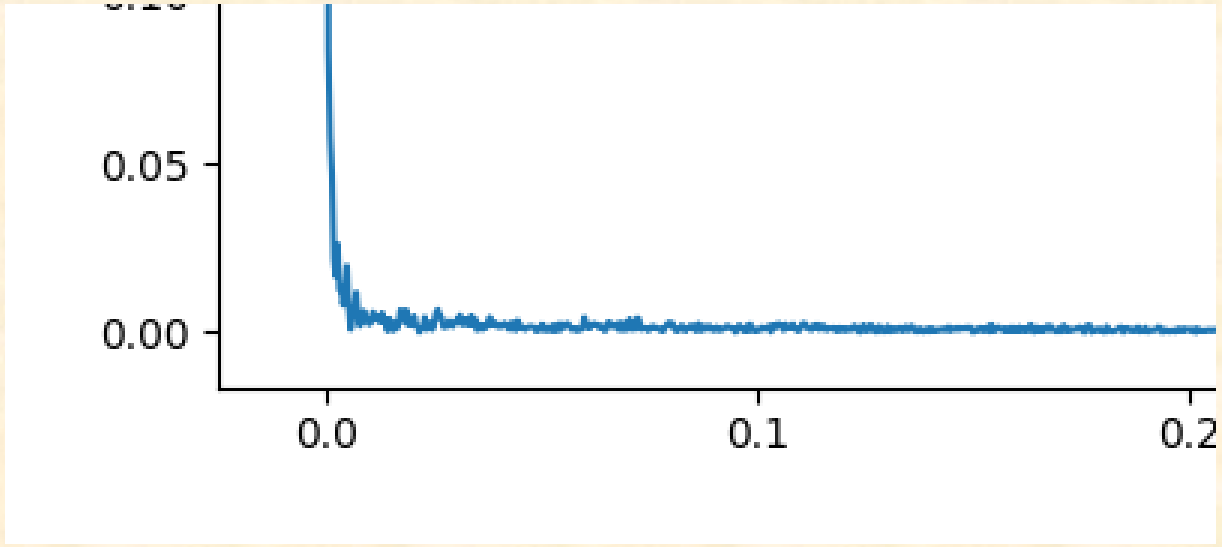
- It has nearly constant group delay, minimizing phase distortion

- Ideal for applications needing signal integrity like velocity measurements.

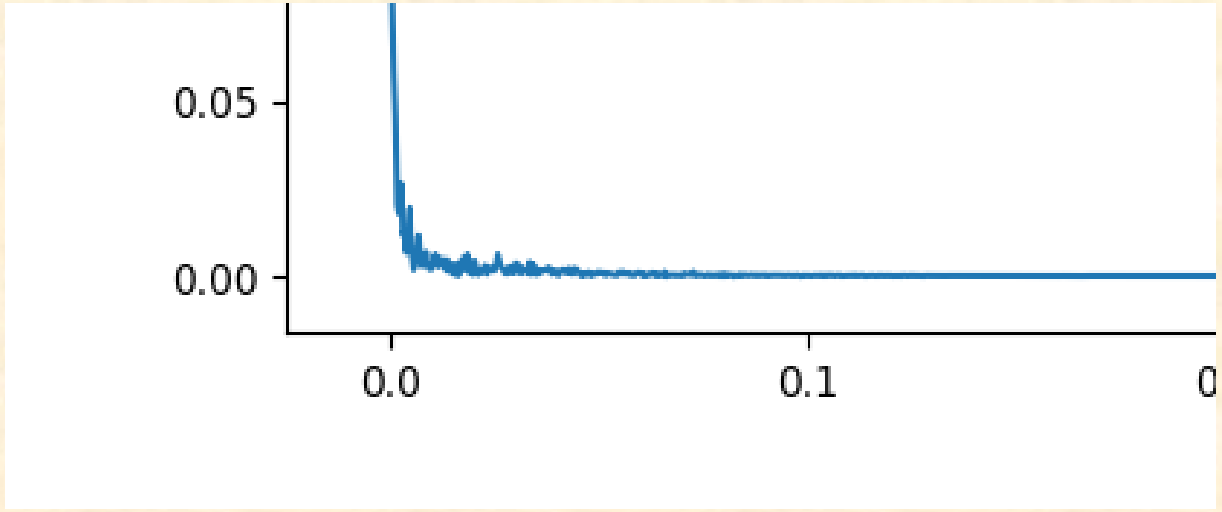
- Slower roll-off and less high-frequency noise attenuation compared to other filters.

Overall–ALL CodeFlow



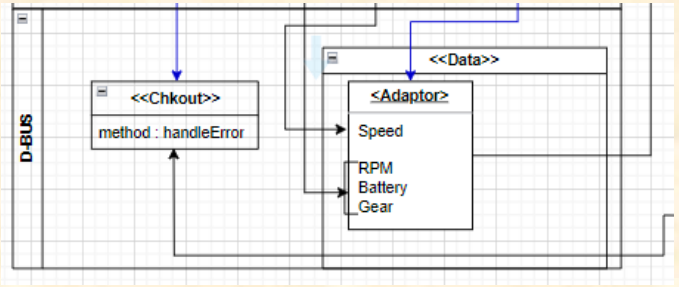


Frequency Response - Raw Data



Frequency Response – Filtered Data

Open D–BUS,Data In D–BUS



D - BUS : "com.example.CanData"

Object Interface
method

Pydbus

"/com/example/CanData/Chkout"

"com.example.Chkout"

"handleError"

"/com/example/CanData/Data"

"local.carDash"

"setData"

Speed

RPM

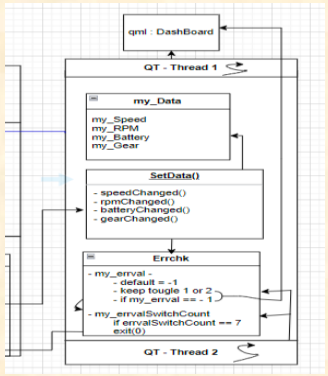
Battery

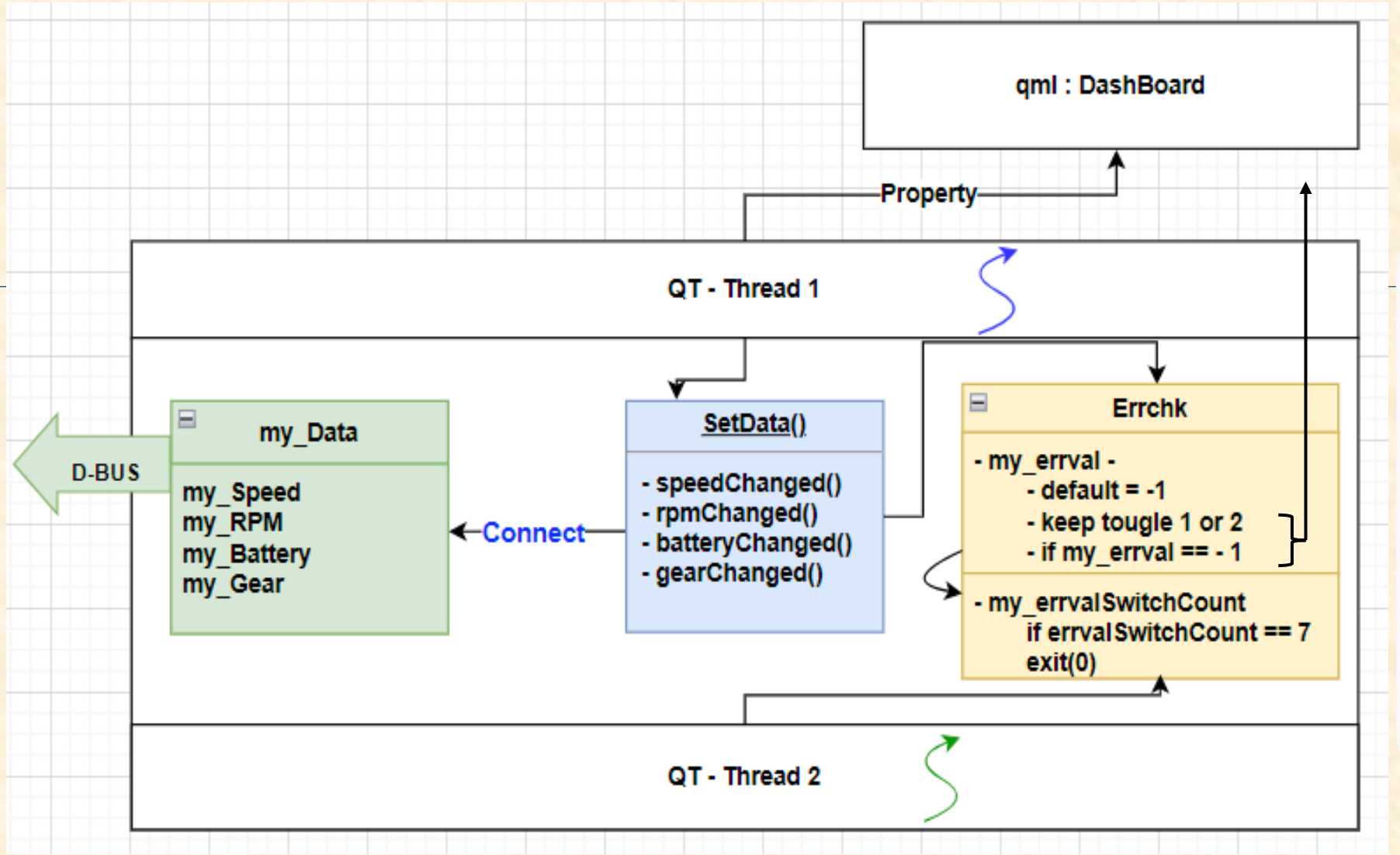
Gear

QT

- Use an Adapter in XML to enhance security and modularity by encapsulating the interface

MultI–Threads In QT

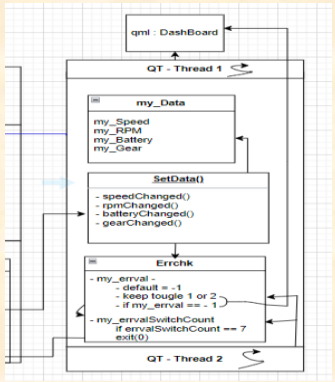


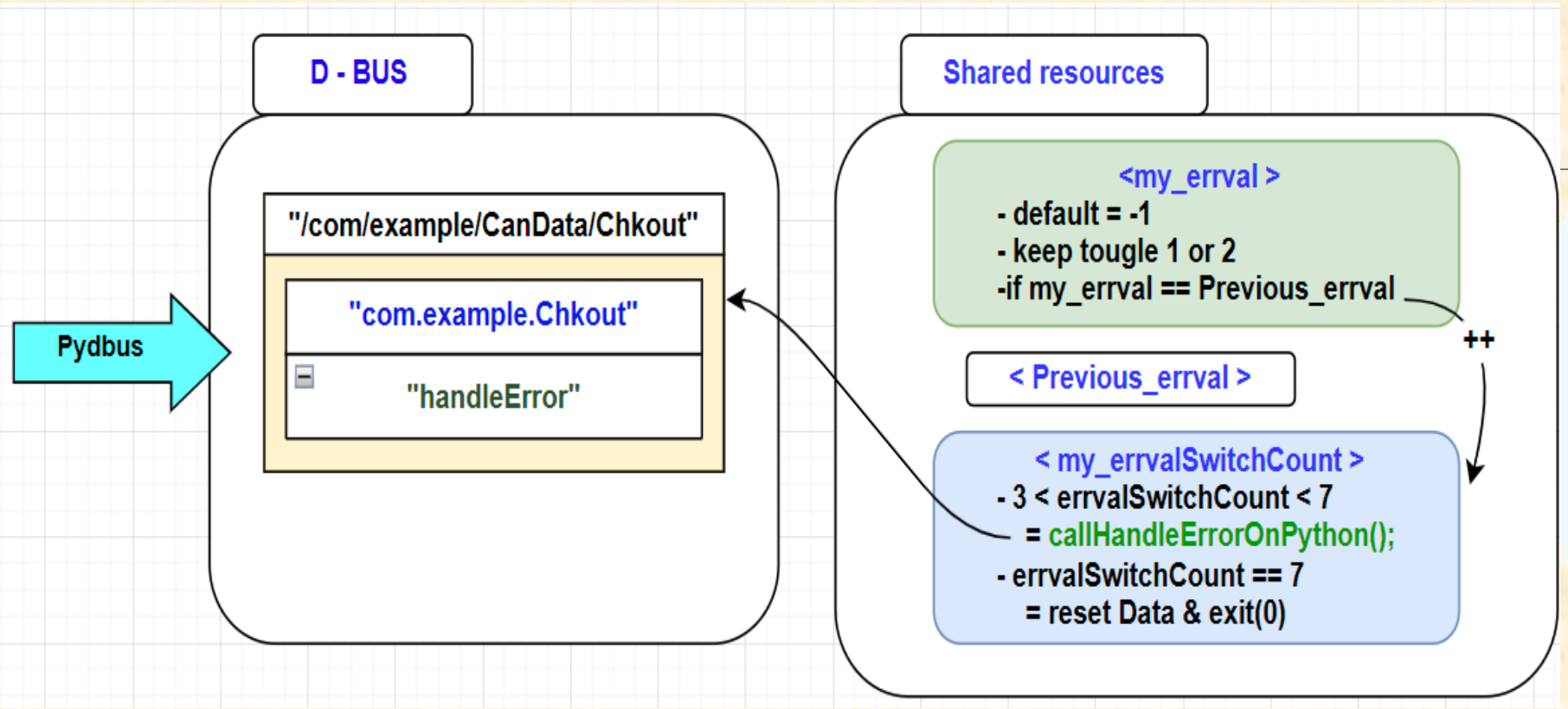


Main Thread:
- Initializes objects and registers D-Bus service.
- Manages value access and updates.

Second Thread:
- Monitors and manages error values in the background.

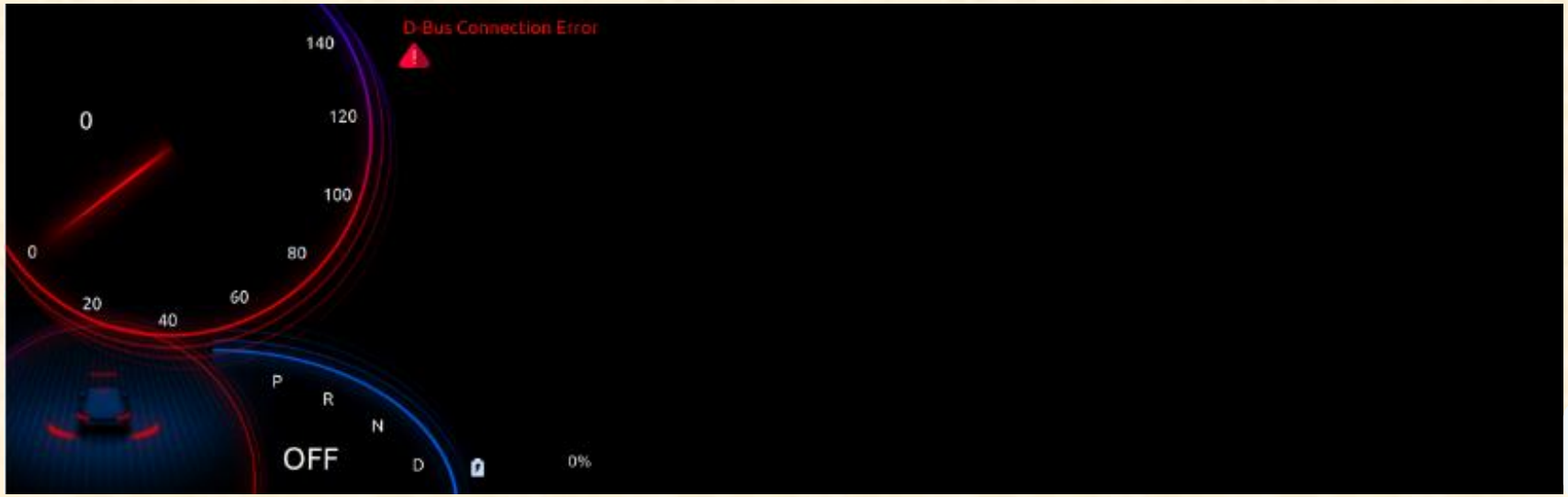
Error Check



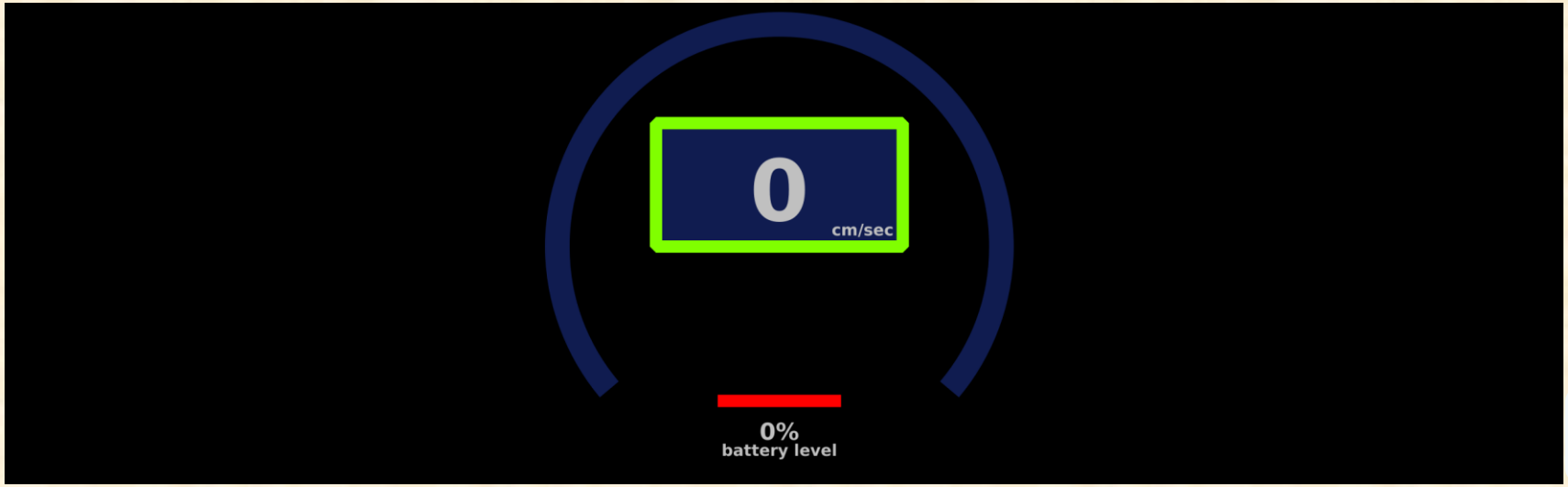


- The thread actively checks the shared resources every 0.2 seconds in the background.

DashBoards



Lee’s



Llgizar’s