Eclipse OpenXilEnv

About the speakers

Ricardo Gonzalez Ramos

- Born 1973
- Joined ZF Summer 2003 (Master Thesis Mechanical Engineering)
- From 2004 onwards Embedded SW Developer Passenger Car Transmission systems
- Senior Manager Infrastructure and Tooling

Eric Bieber

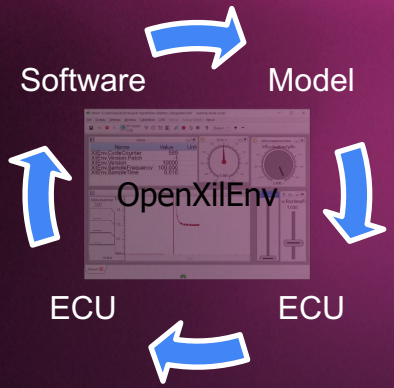
- Born 1968
- 1990 Studies Electrical Systems (Telecommunications) HTWG Konstanz
- 1995 ZF Friedrichshafen AG Embedded Software developer
- 2001 ZF Friedrichshafen AG Infrastructure and tooling

AGENDA SLIDE

- What is OpenXilEnv and where it comes from
- Usecases for OpenXilEnv
- What will be part of open source
- Main Working principle of OpenXilEnv
- Demonstration based on Electric car

SDV
Eclipse Software Defined Vehicle

What is OpenXilEnv



- Eclipse OpenXilEnv provides an environment for creating Software In the Loop (SIL) systems for the Software Defined Vehicle ecosystem. OpenXilEnv is primary developed for setup a SIL (digital twin), through its versatile nature it is also possible to use it in a HIL environment.
- OpenXilEnv provides also an interface to Matlab/Simulink where it can be used for co-simulation between Code and functional models in a Model in the Loop environment (MIL)
- OpenXilEnv also provides a lightweight Hardware in the Loop (HIL) system option for MiniHils (CAN,CN-FD)

Where comes the the Name from?

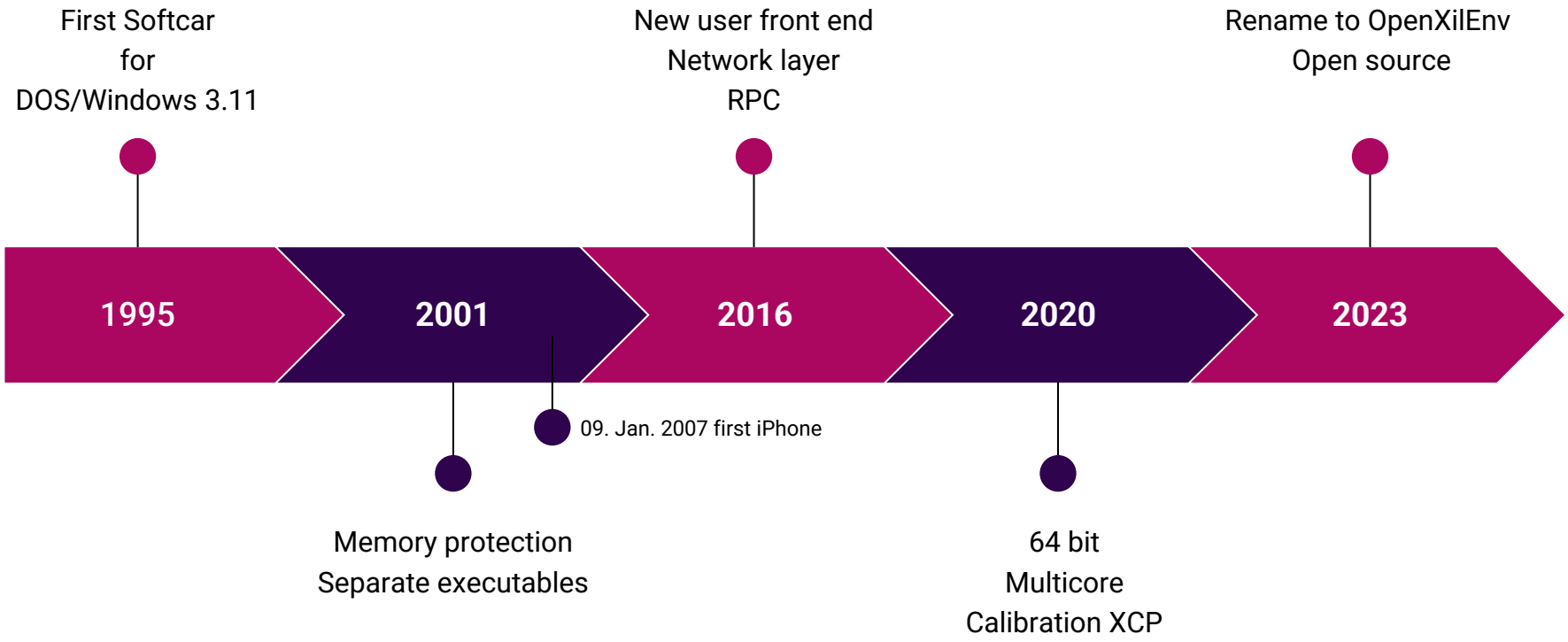
- Open Source - > Open
- Environment -> Env
- Capabilities - > HIL,SIL,MIL
- Heritage Name - > Softcar

```
#include<stdio.h>

int sum(int summand1, int summand2, summand3)
{
        return (summand1 + summand2 + summand3 );
}
int main() {
        int result = sum(„MIL“,“SIL“,“HIL“);
        printf("Summ of MIL and SIL and HIL is %d\n", sum);
        return 0;
}
```

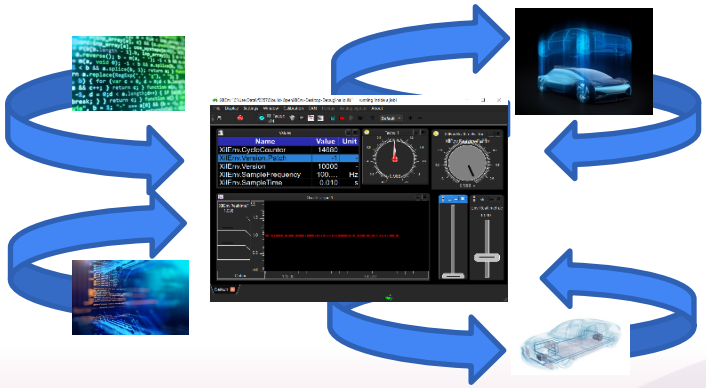
OpenXilEnv ← `Summ of MIL and SIL and HIL is Xil`

Softcar -> OpenXilEnv



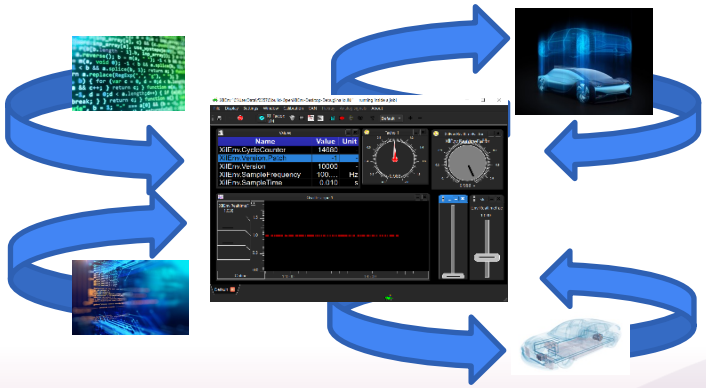
Usecases and features of OpenXilEnv

- SIL **S**oftware **I**n the **L**oop
  - No Target hardware, compiler , … needed
- All parts separated in own executables
  - Communication over a network layer
- Distributed digital twin, the control units/ models must not run on the same location
- Residual bus simulation for emulating ECUs not present
  - Fault injection
- FMI interface for FMUs
- XilEnv (without GUI)
  - No installation is needed.
  - Docker container, or/and in the cloud.
  - Automated Simulations incl. result evaluation
- XilEnvGui (with a Qt GUI)
  - Expansive configurable sheets



Usecases and features of OpenXilEnv

- Parallel execution schedulers/barriers (configuration must done by the user)
- Cosimulation interface to Matlab/Simulik
- Recording
- Stimulation through measurement data or script
- Pre calibration with XCP over ethernet to interact with an external calibration system
  - Or a small internal Calibration system
- RPC (Remote Procedure Call Interface) for automation



Target platform is Windows or Linux.
Mixed 32/64bit Windows/Linux executable.

What will be part of open source

What is our goal

To promote open source digital twin environment and Simplify interaction between digital twin participants

What will be part of OpenXilEnv

The sources of OpenXilEnv to build:

XilEnv

XilEnvGui

XilEnvRpc

XilEnvExtProc

And some small example

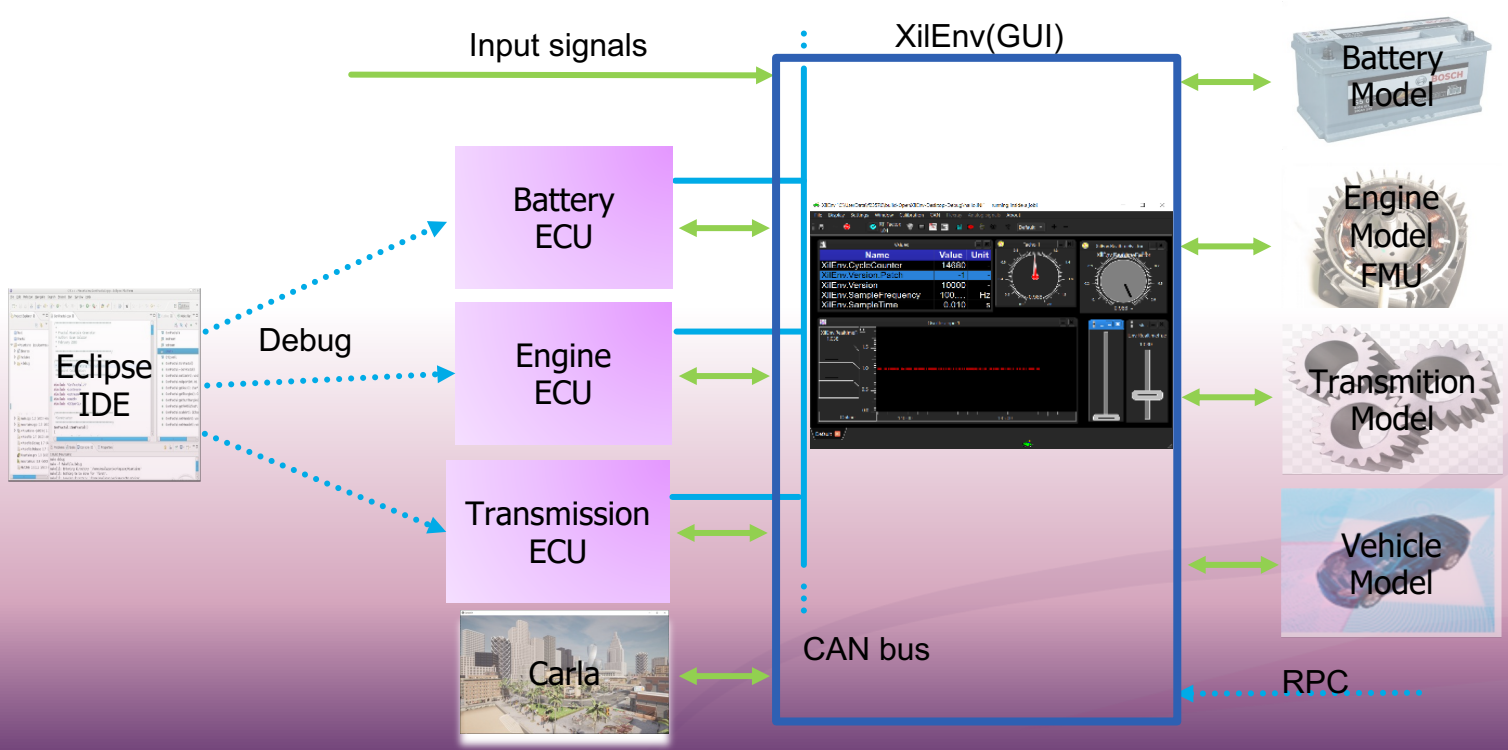
You have to build executables your own.

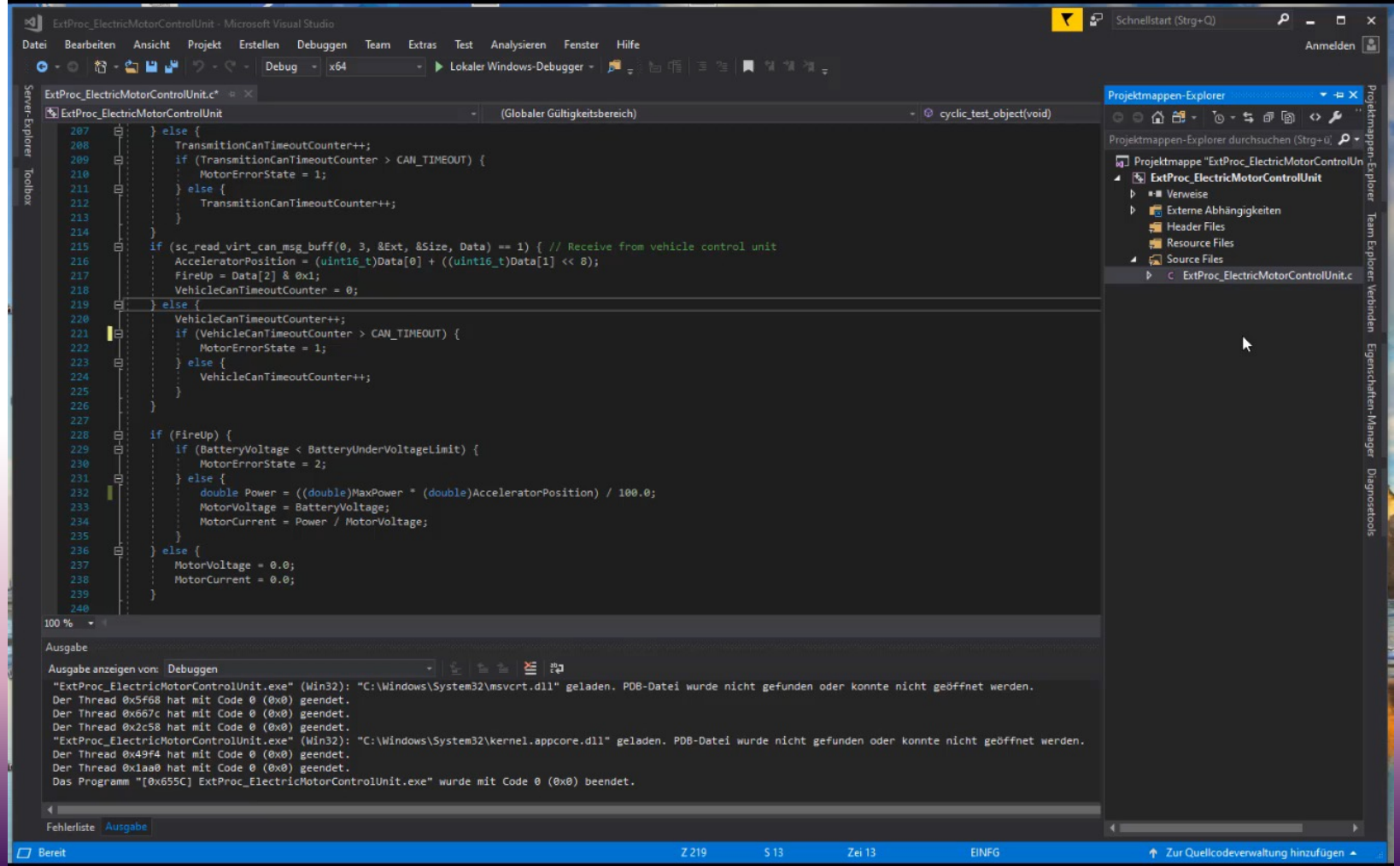
What you can

Use it as it is, change it if you need, give back changes if you want. Contribution will be welcome

Eclipse OpenXilEnv

Working principles




ExtProc_ElectricMotorControlUnit - Microsoft Visual Studio
Datei Bearbeiten Ansicht Projekt Erstellen Debuggen Team Extras Test Analysieren Fenster Hilfe
Debug x64 Lokaler Windows-Debugger
ExtProc_ElectricMotorControlUnit.c*
ExtProc_ElectricMotorControlUnit (Globaler Gültigkeitsbereich) cyclic_test_object(void)
} else {
TransmitionCanTimeoutCounter++;
if (TransmitionCanTimeoutCounter > CAN_TIMEOUT) {
MotorErrorState = 1;
} else {
TransmitionCanTimeoutCounter++;
}
}
if (sc_read_virt_can_msg_buff(0, 3, &Ext, &Size, Data) == 1) { // Receive from vehicle control unit
AcceleratorPosition = (uint16_t)Data[0] + ((uint16_t)Data[1] << 8);
FireUp = Data[2] & 0x1;
VehicleCanTimeoutCounter = 0;
} else {
VehicleCanTimeoutCounter++;
if (VehicleCanTimeoutCounter > CAN_TIMEOUT) {
MotorErrorState = 1;
} else {
VehicleCanTimeoutCounter++;
}
}
if (FireUp) {
if (BatteryVoltage < BatteryUnderVoltageLimit) {
MotorErrorState = 2;
} else {
double Power = ((double)MaxPower * (double)AcceleratorPosition) / 100.0;
MotorVoltage = BatteryVoltage;
MotorCurrent = Power / MotorVoltage;
}
} else {
MotorVoltage = 0.0;
MotorCurrent = 0.0;
}
100 %
Ausgabe
Ausgabe anzeigen von: Debuggen
"ExtProc_ElectricMotorControlUnit.exe" (Win32): "C:\Windows\System32\msvcrt.dll" geladen. PDB-Datei wurde nicht gefunden oder konnte nicht geöffnet werden.
Der Thread 0x5f68 hat mit Code 0 (0x0) geendet.
Der Thread 0x667c hat mit Code 0 (0x0) geendet.
Der Thread 0x2c58 hat mit Code 0 (0x0) geendet.
"ExtProc_ElectricMotorControlUnit.exe" (Win32): "C:\Windows\System32\kernel.appcore.dll" geladen. PDB-Datei wurde nicht gefunden oder konnte nicht geöffnet werden.
Der Thread 0x49f4 hat mit Code 0 (0x0) geendet.
Der Thread 0x1aa0 hat mit Code 0 (0x0) geendet.
Das Programm "[0x655C] ExtProc_ElectricMotorControlUnit.exe" wurde mit Code 0 (0x0) beendet.
Fehlerliste Ausgabe
Projektmappen-Explorer
Projektmappe "ExtProc_ElectricMotorControlUn
ExtProc_ElectricMotorControlUnit
Verweise
Externe Abhängigkeiten
Header Files
Resource Files
Source Files
ExtProc_ElectricMotorControlUnit.c
Bereit Z 219 S 13 Zei 13 EINFG Zur Quellcodeverwaltung hinzufügen

THANK YOU!