Embedded Connectivity (SS21)

Prof. Dr. Ing. Andreas Grzemba

Ing. Johann Bretzendorfer

G2WS4
Rashed Al-Lahaseh - 00821573

Automotive Ethernet Industry

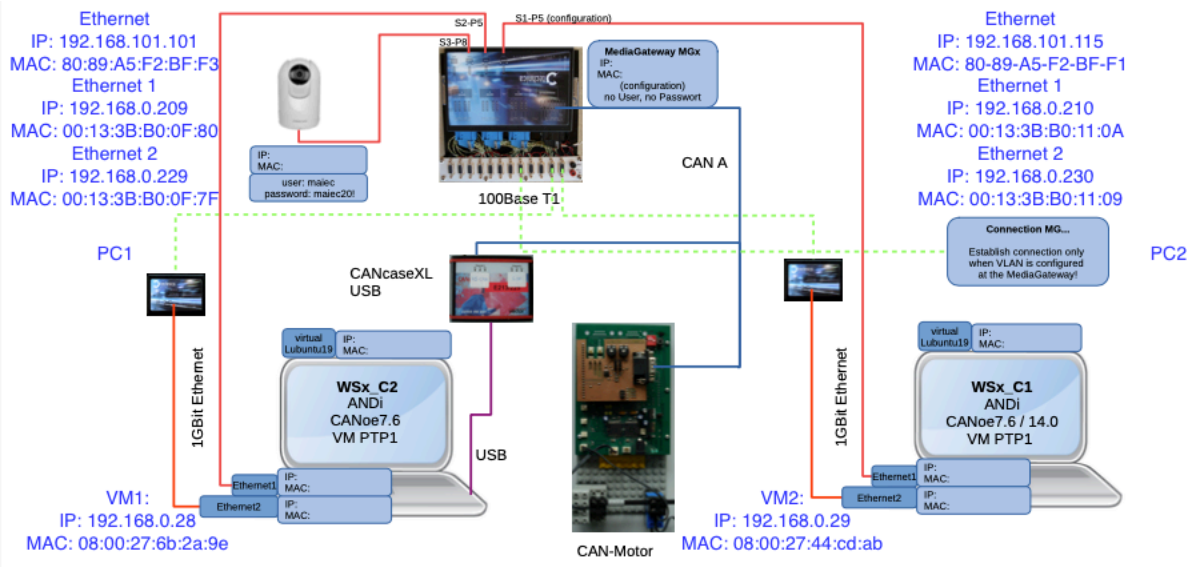
- Regarding Research & Market latest report, the automotive Ethernet market is projected to grow from USD 1.8 billion in 2020 to USD 5.6 billion by 2026, at a Compound Annual Growth Rate (CAGR) of 20.9% from 2020 to 2026.
- Which means major factors expected to drive the growth of the automotive Ethernet market include increasing demand for higher bandwidth, rise in deployment of (Advanced driver-assistance systems) ADAS and infotainment systems, rising vehicle production, and growing demand for passenger and safety and convenience.

Labs

- Now we will take a quick look for the tasks we took during our labs.

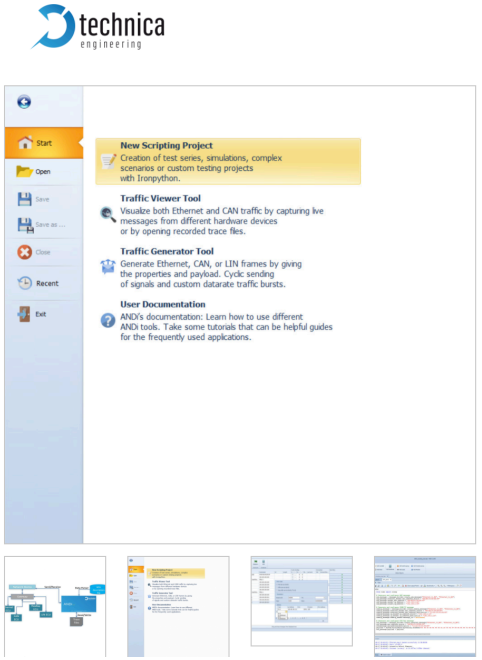
Lab 1

- During the first lab we had the chance to check our work station and we saw that workstation consists of 2 computers.
- The computers have 3 Ethernet ports, which are used for different tasks.



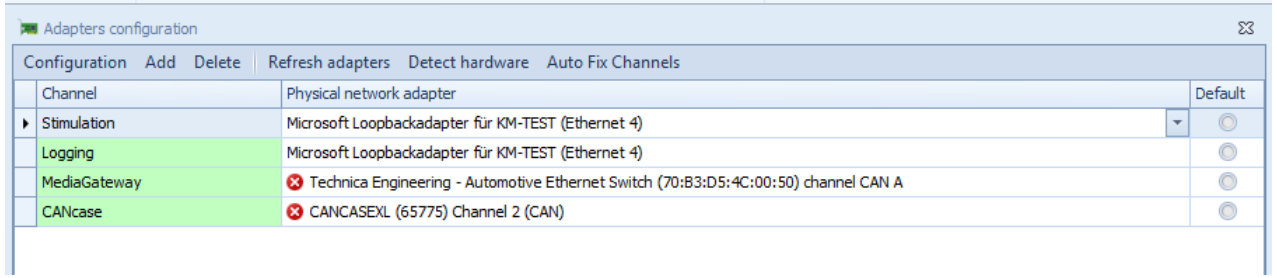
Lab 2

- On the second lab, a new tool where introduced ‘ANDi’ which is a software developed by Technica-Engineering.
- **The Automotive Network Diagnoser (ANDi)** is a test and simulation environment for electronic control units Automotive Ethernet (100BASE-T1 / 1000BASE-T1), CAN, CAN-FD, LIN and FlexRay bus systems.



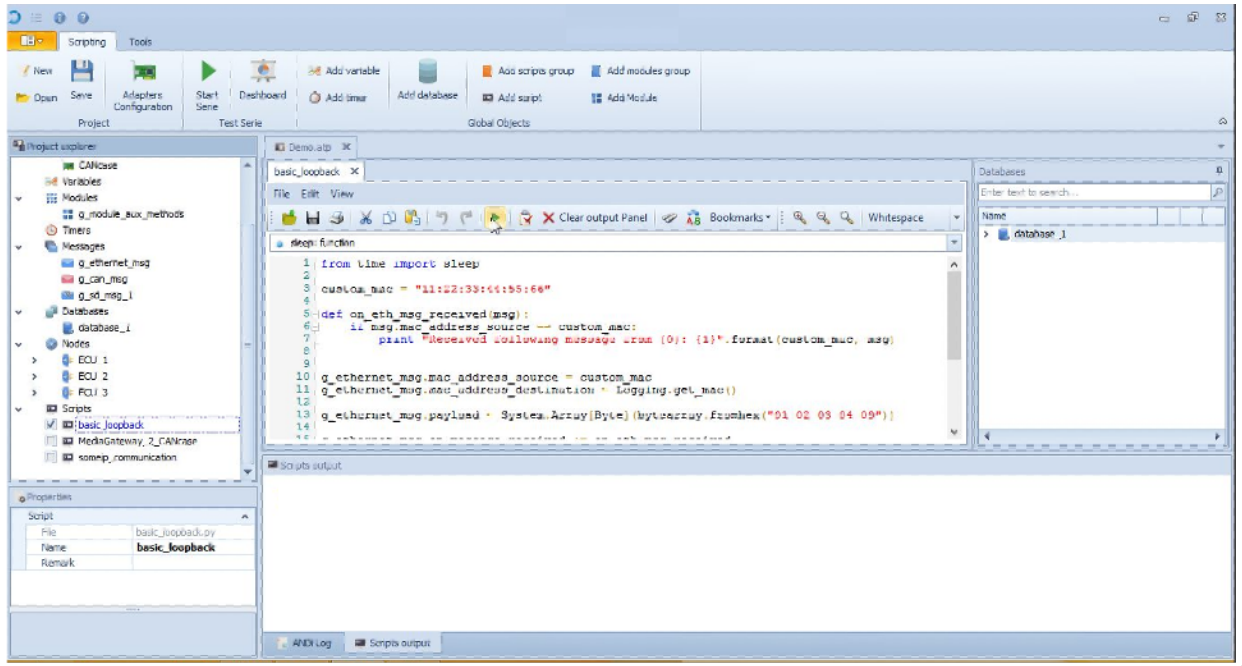
Lab 2

- First we had the chance to check the demo project, which helped us discover the main functionalities in the ANDi tool.
- Adapter configuration, where we set the channels responsbile for simulation and logging process, as the demo project was already setuped to Loopback adapters.



Lab 2

- Basic loopback script, the programing language used is Python and the script was provided using custom_mac address and it we saw that if we want to establish a bidirectionl data traffic, the sender address must be correct.



Lab 3

- On the third lab we had to establish a send/recive script based on the basic loopback script we tested.
- And from this task we saw that we can modify the adapter configuration and set the logging and simulation to the required enviorment.

```
[2021-04-30 11:45:21]: Start 'receive'
[2021-04-30 11:45:45]: Start 'send'
[2021-04-30 11:45:45]: Sender: Frame 01 has been sent.
[2021-04-30 11:45:45]: Receiver: Received message: [Mac src: 02:00:4C:4F:4F:50 Mac dest: 02:00:4C:4F:4F:50 EtherType: Unknown [2021-
04- 30 11:45:45]: Payload Data : {1123}]
[2021-04-30 11:45:46]: Sender: Frame 02 has been sent.
[2021-04-30 11:45:46]: Receiver: Received message: [Mac src: 02:00:4C:4F:4F:50 Mac dest: 02:00:4C:4F:4F:50 EtherType: Unknown [2021-
04- 30 11:45:46]: Payload Data : {2123}]
[2021-04-30 11:45:47]: Sender: Frame 03 has been sent.
```

```
[2021-04-30 11:45:47]: Receiver: Received message: [Mac src: 02:00:4C:4F:4F:50 Mac dest: 02:00:4C:4F:4F:50 EtherType: Unknown [2021-
04- 30 11:45:47]: Payload Data : {3123}]
...
[2021-04-30 11:46:04]: Sender: Frame 20 has been sent.
[2021-04-30 11:46:04]: Receiver: Received message: [Mac src: 02:00:4C:4F:4F:50 Mac dest: 02:00:4C:4F:4F:50 EtherType: Unknown [2021-
04- 30 11:46:04]: Payload Data : {20123}]
[2021-04-30 11:46:05]: Sender: Frame 21 has been sent.
[2021-04-30 11:46:05]: Sender: Sending End of Transfer message to receiver...
[2021-04-30 11:46:05]: Receiver: Received message: [Mac src: 02:00:4C:4F:4F:50 Mac dest: 02:00:4C:4F:4F:50 EtherType: Unknown [2021-
04- 30 11:46:05]: Payload Data : {21123}]
[2021-04-30 11:46:05]: Receiver: Received message: [Mac src: 02:00:4C:4F:4F:50 Mac dest: 02:00:4C:4F:4F:50 EtherType: Unknown [2021-
04- 30 11:46:05]: Payload Data : {0}]
[2021-04-30 11:46:05]: finished 'send' in 20258.7164ms (+193ms cleanup)
[2021-04-30 11:46:06]: Receiver: All frames received successfully. Ending receiver script...
[2021-04-30 11:46:06]: finished 'receive' in 44937.036ms (+145ms cleanup)
```

Lab 3

- And modify the source code to initialize the IP for the source and destination instead of using the static custom mac address that was used in the demo script.
- Where PC-1 was the sender and my partner PC-2 was the receiver.

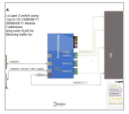
send.py

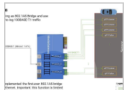
```python
#include libraries
from time import sleep

#used to add a sequence number to frames
count = 0

#set source and destination MAC addresses
g_ethernet_msg.mac_address_source = "00-13-3B-B0-0F-7F"
g_ethernet_msg.mac_address_destination = "00-13-3B-B0-11-09"

#timer function
def on_elapsed(source, event_args):
    #inc counter
    global count
    count += 1
    #make message using value of count as it's first byte and then send it
    g_ethernet_msg.payload = System.Array[Byte](bytearray.fromhex('%02X'%count + "01 02 03"))
    g_ethernet_msg.send()
    #display a status message
    print("Sender: Frame" + '%02d'%count + " has been sent.")

try:
    #timer setup
    timer = andi.create_timer()
    timer.interval = 1000
    timer.on_time_elapsed += on_elapsed
    timer.start()
    #wait 20 seconds
    sleep(20)

finally:
    #send 'end of transfer' message : {0} this message ends receiver script
    print("Sender: Sending ""End of Transfer"" message to receiver...")
    g_ethernet_msg.payload = System.Array[Byte](bytearray.fromhex("00"))
    g_ethernet_msg.send()
    #different event
    timer.on_time_elapsed -= on_elapsed
    timer.stop()
```

Lab 3

- On the second part from the lab we had to establich a VLAN connecting through MediaGateway.
- **Media Gateway** from Technica-Engineering is a development tool for testing and analyzing on-board vehicle networks.

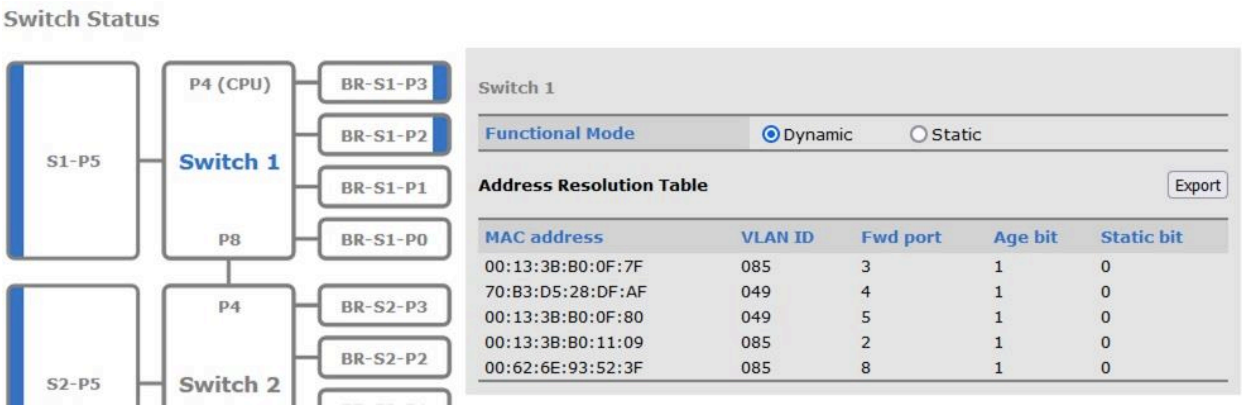
technica engineering





Lab 3

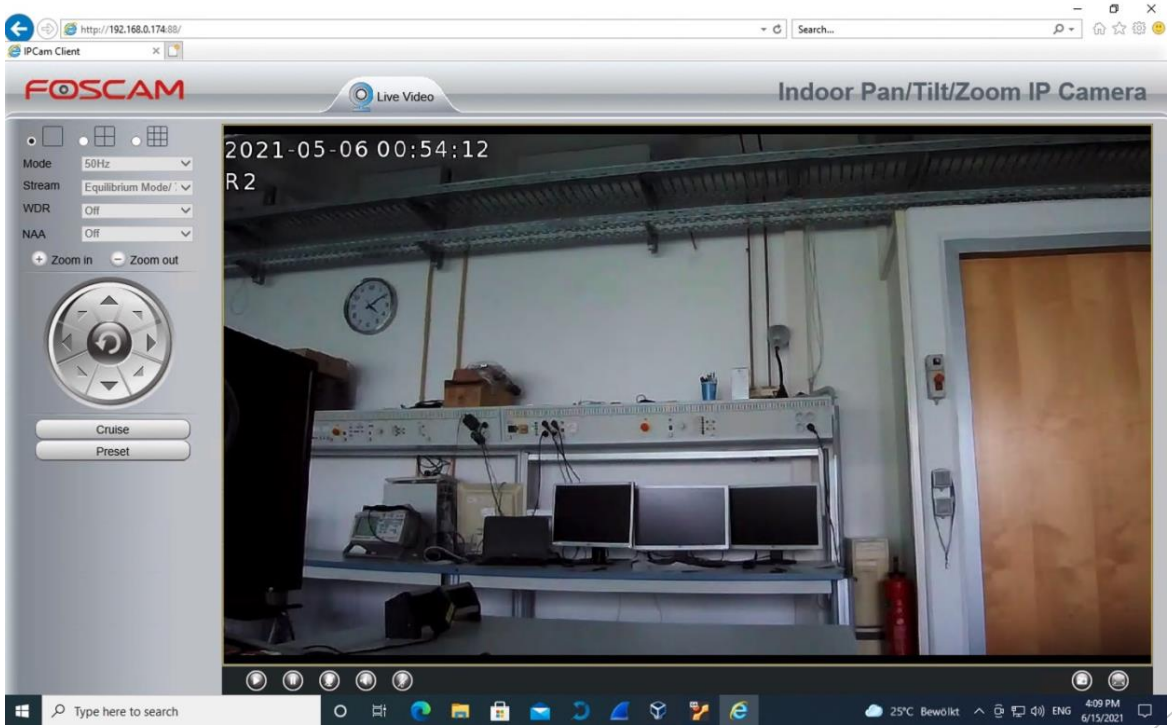
- We had the chance in this part to understand how does MediaGateway inerface works and what should we do to establish a VLAN connection, access to another gateway.
- Also how to connect the 2 PCs to the local webcam.
- I were responsible for the MediaGateway configuration on PC-1 and my partner for analyzing the connection through wireshark for PC-2.



In order to gain access to the local webcam, ports S3-P8, S3-P4, S2-P8, S2-P4 and S1-P8 are configured to be a member of VLAN 85.

Lab 3

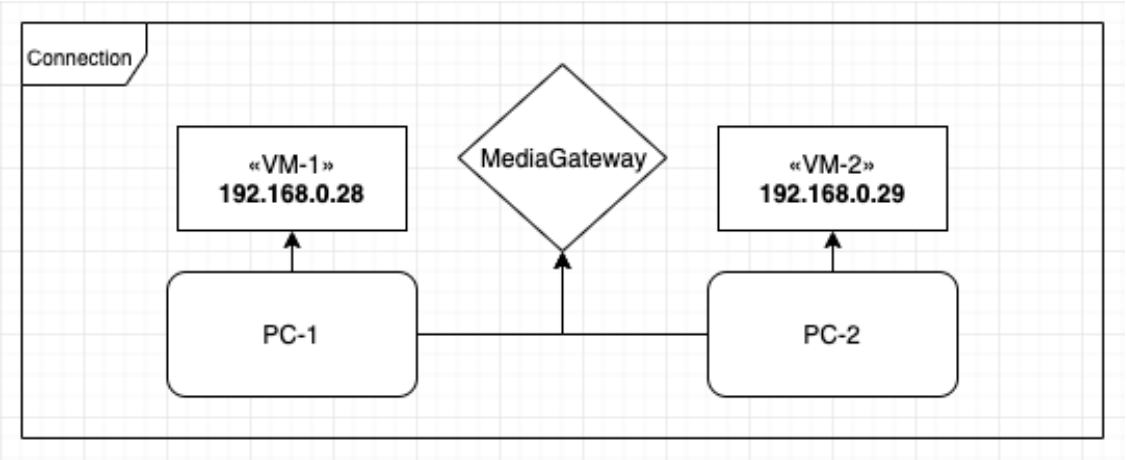
- On the last task we had to establsh a connection through MediaGateway and centeral MediaGateway.
- Where we need to setup PC1: Central Webcam – MediaGateway Configuration and PC2: Central Webcam – Local Webcam.



PC1 access to the Central Webcam

Project

- We have choosen the VSomieIP project, where we had to establish a connection between 2 virtual machines through MediaGateway using SomeIP protocol.
- Both of us were responsible for the work on the porject part.



SOME/IP

- SOME/IP is an automotive middleware solution that can be used for control messages.
- Provides service oriented communication over a network.
- It is based on service definitions that list the functionality that the service provides.
- A service can consist of combinations of zero or multiple events, methods and fields.

SOME/IP

- Serialization describes the way data is represented in protocol data units (PDUs) as payload of either UDP or TCP messages, transported over an IP-based automotive in-vehicle network.
- SOME/IP Header Format

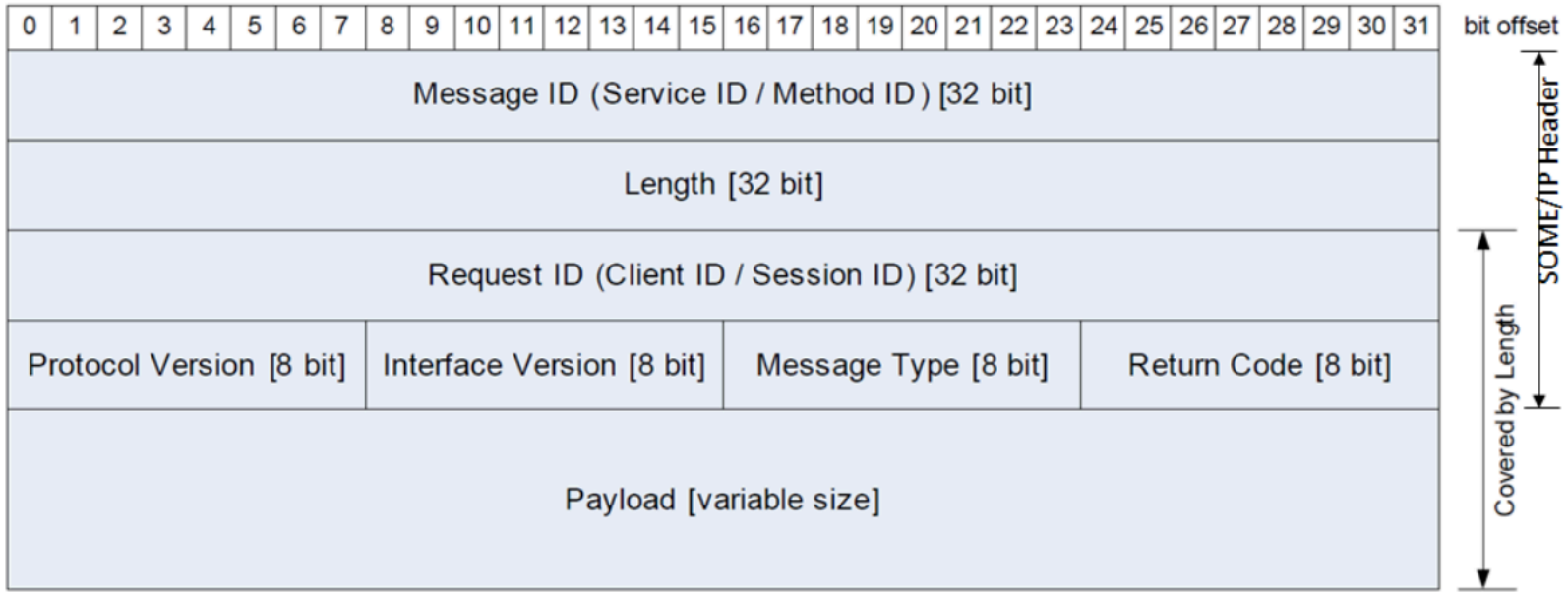
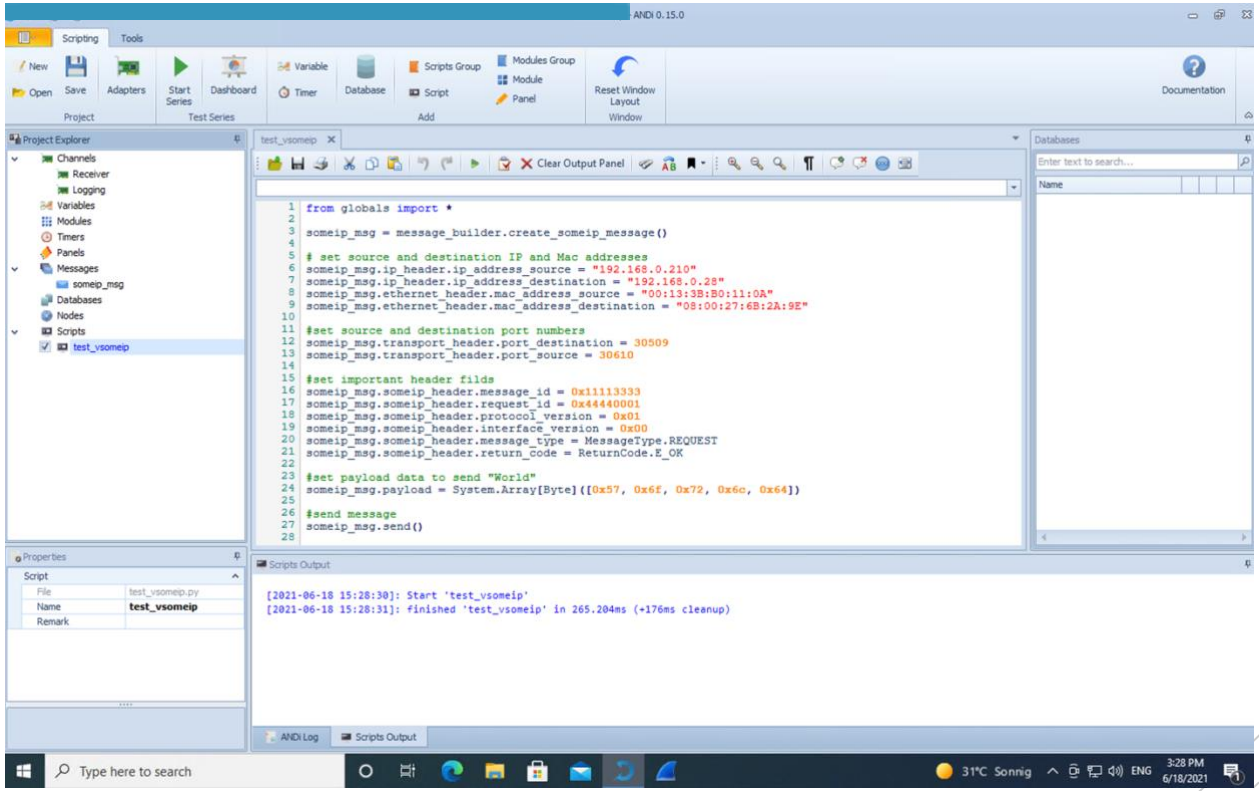


Figure taken from: *SOME/IP Service Discovery by Dr. Lars Völker*

Project

- And the second task was to model one of the nodes using ANDi.
- We need to be able to communicate between Linux computers and Andi on Windows.



Project Conclusion

- In order to establish service communication:
- The services must be made known in the network where this is done using the service discovery protocol.
- First, a server announces its services where the IP and port of the service are also transmitted.
- If a client needs data from a service, it will subscribe to this at the server.