Test the CAN Controller and Transceiver Journal

Mikkel Wohlert Jensen

2024-12-23

Contents

1 Purpose

Experimentation is needed to verify the functionality of the application software for the Controller Area Network (CAN). This journal describes the following tests performed using the CAN driver developed in the Zynq-Based Master Controller bachelor's thesis:

- Decoding a CAN frame from the bus between the X90 Master Controller and Dashboard.
- Transmitting a 40 Kb/s CAN frame onto an inactive bus.
- Transmitting a 1 Mb/s CAN frame onto an inactive bus.
- Snooping on the bus between the X90 Master Controller and Dashboard.
- Integrating the Zynq-Based Master Controller onto the bus between the X90 Master Controller and Dashboard.

2 Materials used

- Zynq-Based Master Controller,
- Testbench for Zynq-Based Master Controller,
- 4-pin cable (3 m) which mates to the connector on the X90 Interface PCB,
- EA-PS 5080-20 A Power Supply from *Elektro-Automatik,*
- RTM 3004 Oscilloscope from *Rohde & Schwarz* with probe heads.

3 General Procedure for All Experiments

The materials are assembled as depicted in the figures below.

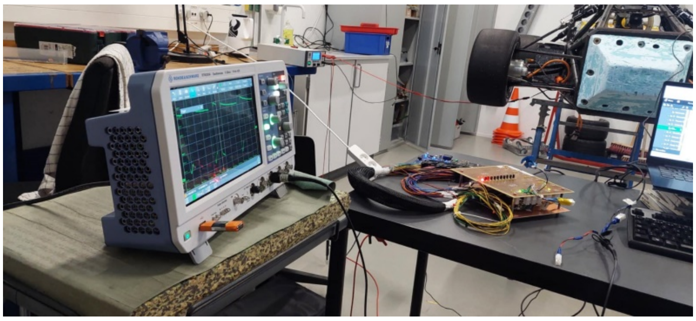

Figure 1: General setup

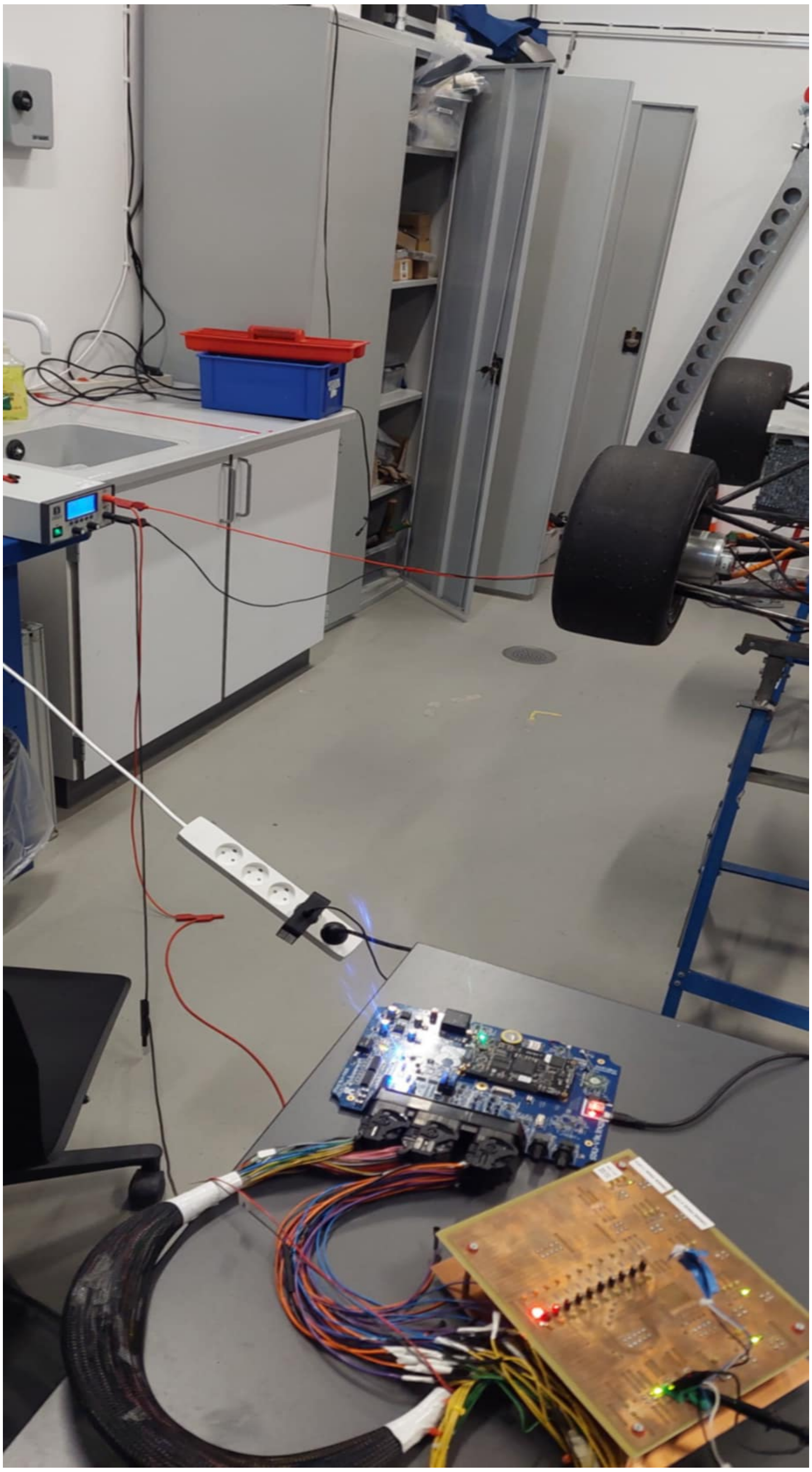

Figure 2: General setup

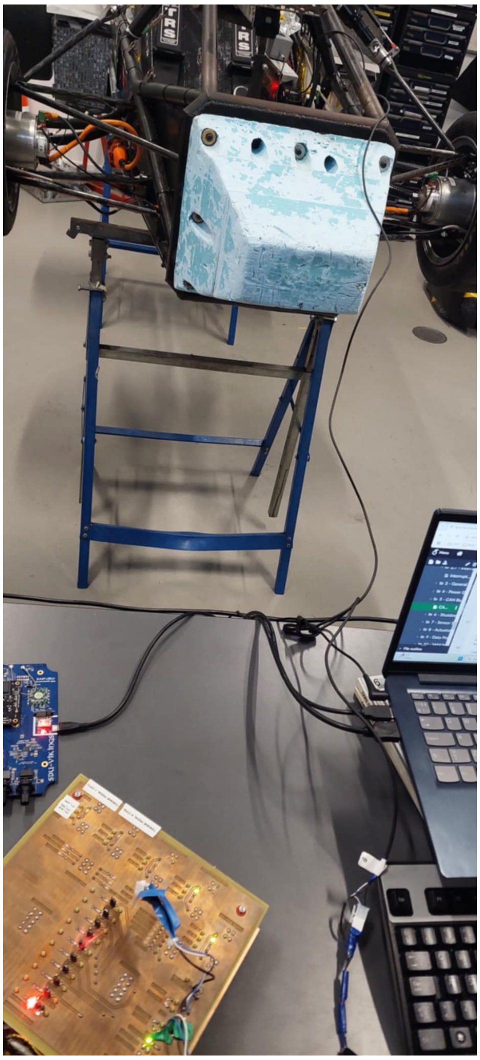

Figure 3: Connection the X90 Interface PCB via phoenix cable

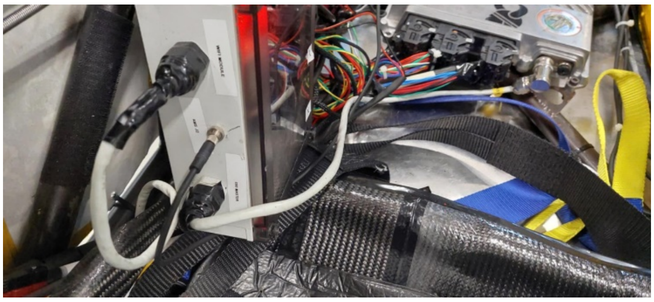

Figure 4: Connection the X90 Interface PCB via phoenix cable

Figure 5: Connection the X90 Interface PCB via phoenix cable

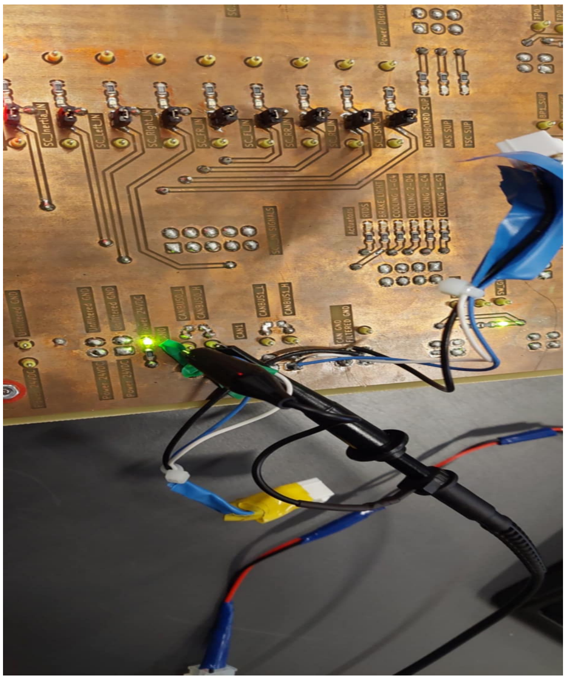

Figure 6: Probe connected to the CANH and CANL lines

4 Decoding a Message from the bus between the X90 Master Controller and Dashboard - Procedure

The power supply is connected to the Viking XI using banana cables with a 1-way battery connector attached that mate with the ones on the car.

The car's key (switch) is turned, thus enabling current from the power supply to flow into the LV system of the car, powering the Dashboard and starting the communication between the X90 Master Controller and the Dashboard.

The bus is then measured with the oscilloscope.

5 Results of Decoding Experiment

Snapshots from the oscilloscope can be found below.

Figure 7: The length of the entire frame is approximately 45 µs

The decoded message is discussed in the report.

Figure 8: The decoded message is discussed in the report

Figure 9: The length of a single bit is approximately 1 µs or ~1 Mb/s

6 Transmission of a 40 Kb/s Frame - Procedure

The power supply is disconnected from the Viking XI and instead connected to the power input of the Zynq-Based Master Controller.

The bit timing logic is configured to:

- BRPR = 29,
- BTR_FIRST_SEG = 15,
- BTR _SECOND_SEG = 2,
- BTR_SJW = 3.

The following is to be transmitted:

- u32 CAN_ID = 0b00111;
- u8 Data[] = {0b111}; // 1 byte
- CAN_Send_Data_Frame(&CAN0_PS_inst, CAN_ID, Data);

7 Results of 40 Kb/s Transmission Experiment

The 40 Kb/s experiment was done on two different occasions with a 7-10 day period in between. Unfortunately, the results from the first experiment, that yielded the expected 40 Kb/s frame have been lost.

Below are the results from the second attempt done on the same day as the 1 Mb/s experiment.

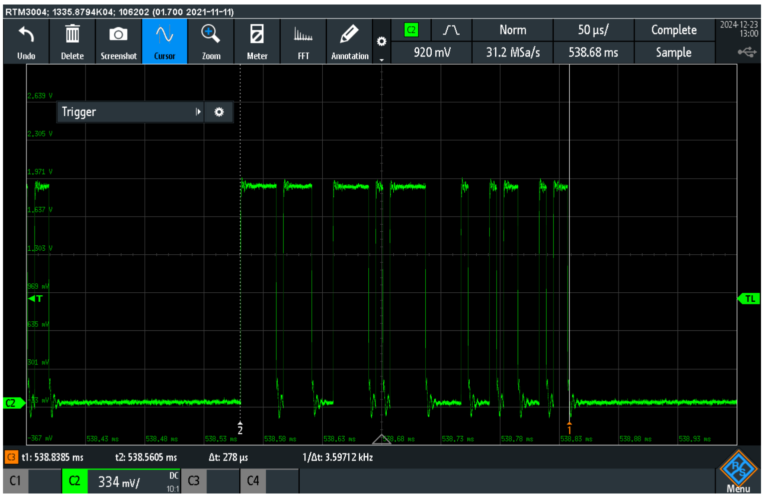

Figure 10: The complete CAN frame is roughly 275 µs long

The output of the transceiver matches the intended message (see CAN implementation section in report).

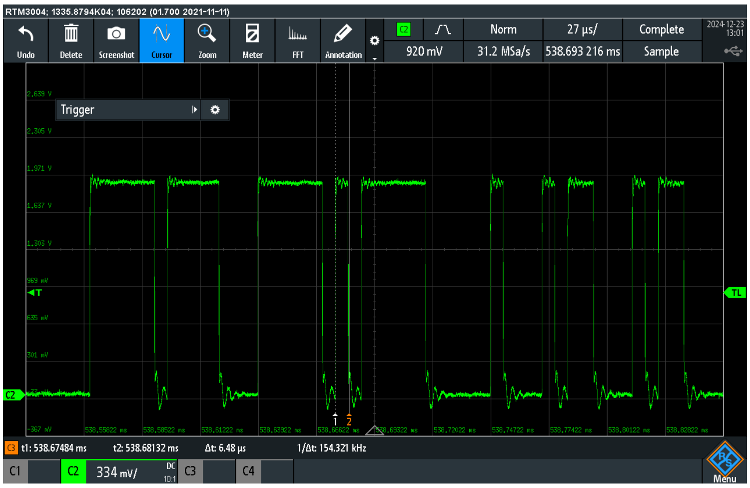

Figure 11: The output of the transceiver matches the intended message

Figure 12: A single bit is approximately 6 µs long or ˜160 Kb/s, 4 times the expected 40 Kb/s

8 Transmission of a 1 Mb/s Frame - Procedure

Using the same setup as the 40 Kb/s experiment, with the only difference being the bit timing logic. It is instead configured to:

- BRPR = 2,
- BTR_FIRST_SEG = 4,
- BTR_SECOND_SEG = 1,
- BTR_SJW = 3.

9 Transmission of a 1 Mb/s Frame - Procedure

Using the same setup as the 40 Kb/s experiment, with the only difference being the bit timing logic. It is instead configured to:

- BRPR = 2,
- BTR_FIRST_SEG = 4,
- BTR_SECOND_SEG = 1,
- BTR_SJW = 3.

10 Results of 1 Mb/s Transmission Experiment

Snapshots from the oscilloscope can be seen below:

Figure 13: A complete frame is roughly 10.5 µs long

The output of the transceiver matches the intended message.

Figure 14: The output of the transceiver matches the intended message

Figure 15: A single bit is roughly 250 nS, or ~4 Mb/s, also 4 times the expected value of 1 Mb/s

11 Transmission of a 1 Mb/s frame Using Previous Values - Procedure

Same setup, but different bit timing logic:

- BRPR = 2,
- BTR_FIRST_SEG = 3,
- BTR_SECOND_SEG = 2,
- BTR_SJW = 3.

12 Results of 1 Mb/s Transmission Using Previous Values Experiment

The transceiver was confirmed not transmit any message onto the bus, nor did the CAN controller output anything into the TxD input of the transceiver.

13 Snooping on Existing CAN Bus and Integrating the Zynq Onto Existing Can Bus - Results

The Zynq-Based Master Controller was unable to snoop on the CAN bus between the X90 Master Controller and the Dashboard due to the results found in the previous experiments. Therefore, no additional testing was able to be done in regards to integration.

14 Conclusion

- The Bit timing logic can achieve 40 Kb/s and 1 Mb/s, but for some reason a 4x factor has been included since the first tests, where a 40 Kb/s was achieved.

- Given these results, it seems to indicate that the bit timing logic works as intended, as the factor is constant across all transmissions, with it also scaling the 40 Kb/s transmission, which previously was 40 Kb/s. This ultimately means that no tests of snooping on the car / integration into the car's CAN bus can be done as of this writing.
- The source of error has yet to be determined, but the following has been confirmed to not be the cause:
  - Vivado setup says 23.8 MHz CAN clock.
  - There is no hardware issue (at least in regards to transceivers and wiring between testbench and CAR.
  - The 1 MHz and 40 Mhz signal was measured with and without connection to the car.