CAN BOARD PRACTICE – CAN COMMUNICATION

# Benefit

After this practice, the students will have experience how to implement CAN communication protocol to receive and transmit data.

The student will understand how CAN nodes are connected in a physical layer.

Also, the students will understand how requirement for CAN protocol looks like.

Moreover, the students understand the application checksum.

# Practice Board information

[Open405R-C | STM32 development board | Cortex M4 (waveshare.com)](https://www.waveshare.com/open405r-c-package-a.htm)

Board: Core405R, STM32F4 Core Board ([Core405R | STM32 development board | Cortex M4 (waveshare.com)](https://www.waveshare.com/core405r.htm))

Development Resources Related software (KEIL etc.)

* Examples in C
* Schematic (PDF)
* Development documentations

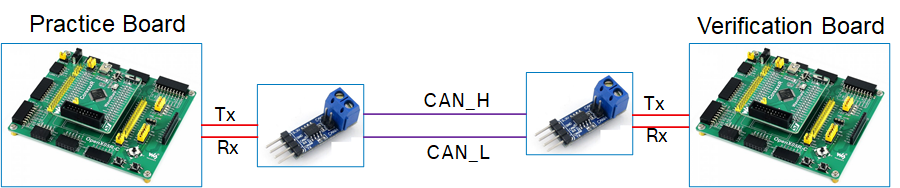
Wiki: [www.waveshare.com/wiki/Core405R](http://www.waveshare.com/wiki/Core405R)

Mother board: OpenX05R-C

Schematic: [File:OpenX05R-C-Schematic.pdf - Waveshare Wiki](https://www.waveshare.com/wiki/File:OpenX05R-C-Schematic.pdf)

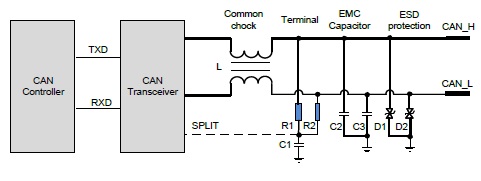
# Overview

CAN node 1 (Practice Board) <-> CAN node 2 (Verification board)



The student has to implement CAN read and CAN write method in CAN node 1 to communication with CAN node 2 on Data frame as BOSCH CAN spec 2.0

Assuming that CAN controller and CAN Transceiver are available in CAN node 1 and CAN node 2



System requirement:

* CAN communication should use the 11-bit ID format standard frame

## Bit timing parameters:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Parameter | Symbol | Minimum value | Nominal value. | Maximum value | Unit | Remarks |
| Bit timing | tBit | 1992 | 2000 | 2008 | ns | fHSCAN = 500kbps (± 0.4 %) |
| Tq quantity | NBT | 10 | 16 | 20 | Tq |  |
| Sampling position | tSP | 75 | 75 | 82 | % |  |
| Synchronisation jump width | SJW | 2 | 2 | 3 | Tq |  |
| Sampling amount | NSP | - | 1 | - |  |  |

Byte order of signal value should be transmitted and received in the Motorola format (big endian format). The start bit of the signal starts from the least significant bit (lsb) of the least significant byte (LSB).

Example, as shown in Figure 1, a data field contains a 10-bit-signal message in the Motorola format, whose start bit is 28 bit.



Figure 1 10-bit-signal message configuration (Motorola format)

The unused bits in the message must be set to 0, and the unused bytes must be set to 0x00.

**CAN node 1:**

TX message: 0x012

RX message: 0x0A2

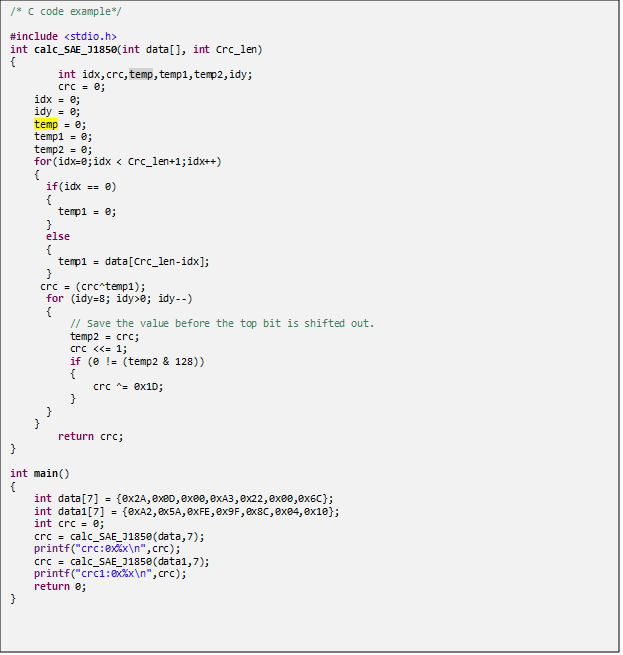
**CAN node 2:**

TX message: 0x0A2

RX message: 0x012

CAN node 2 (Verification board) will send out message with ID 0x0A2 with data in byte 0 and byte 1 (0x0A2), the byte 6 will be message counter from (0x0 ~ 0xF), the byte 7 will be the checksum of data from byte 0 to byte 6

CAN node 1 needs to receive message with ID 0x0A2 data, extract the latest data from 0x0A2 (excluding the message counter information), copy the value to byte 0 and byte 1 and sum (byte 0 + byte 1) then set the result to byte 2 of 0x012. The byte 6 will be message counter from (0x0 ~ 0xF), byte 7 of message 0x012 will be the checksum byte of data from byte 0 – byte 6. Checksum algorithm will follow CRC 8 SAE J1850:



Node 2 will send out 0x012 with cycle 50ms.

Summary:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Frame | Type | DLC(Bytes) | Cycle | Sender | Receiver | Byte0 | Byte1 | Byte2 | .. | Byte6 | Byte7 |
| 0x12 | 11bit ID, CAN standard | 8 | 4s | Node1 | Node2 | Value1 | Value2 |  |  | Mes counter | Checksum |
| 0xA2 | 11bit ID, CAN standard | 8 | 4s | Node2 | Node1 | Value1 | Value2 | Value1+Value2 |  | Mes counter | Checksum |

Additional requirement:

Each time when note 1(or note 2) send or receive CAN messages the board will send the data of these message to Computer via UART and display it in the Hercules tool

Push one button to make the CRC in of message 0x12 wrong.

Incase node 2 do not review data from message 0x12(cause of wrong CRC or message 0x12 is not sent out), node 2 will send message 0xA2 with data byte 0 to 5 are 0x00.

Assessment:

Node 1 can send 0x012 (2 point)

Node 1 can send 0x012 on time (+- 1 millisecond) (2 point)

Node 1 can send correct data in byte 0, 1 (1 point)

Node 1 can send correct data in byte 2 (1 point)

Node 1 can send correct checksum in byte 7 (2 point)

Node 1 can display the value Via UART and display in the computer