SnS CAN API Driver

User Guide

Rev 1.4

August 2020

This document refers to Xilinx Vivado and SDK Release Version 2018.2.

Table of Contents

1 Introduction

1.1 Scope

This document provides a walkthrough for CAN API StandAlone 2018.2 AXI UltraScale+ driver development.

1.2 Audience

This document assumes basic familiarity with Vivado and SDK 2018.2 provided by Xilinx.
The data and procedures described in this document cover UltraScale+, Xilinx SDK Release Version 2018.2.

1.3 Support

If you have any question or require further assistance, use any of the following methods to contact Sital customer support:

- By Email: support@sitaltech.com
- By Phone: +972-9-7633300
- By Fax: +972-9-7663394

1.4 About this User Guide

This document is the user's manual for the CAN API software for Sital's AXI ARINC-825-4 IP Core.

This core incorporates an AXI Slave core coupled with one ARINC-825-4 IP core. This manual is intended to serve as a user's manual for aspects of the AXI ARINC-825-4 core's operation that are not covered by the ARINC825 IPs manual. Please refer to the ARINC-825-4IP core user's manual for a detailed description of the ARINC-825-4 IP.

2 Architecture and Software Layers

2.1 Components

The software architecture is composed of the following components / blocks:

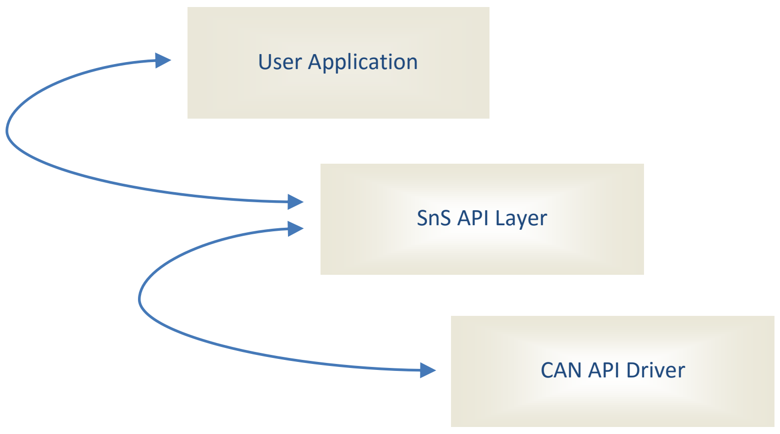


Table 1 below provides descriptions of the various software layers.

Table 1. Software Layers

Layer	Name	Purpose & Functionality
User Application	Code example	- Entry point for the application - Get Number of CAN devices in AXI / PCI - Initialize all devices - Create frame, messages, receive and transmit messages
SnS API layer		- Apply SnS logic by user application calls - Write and Read using driver layer to specific SnS device - Support DataBase arrays of SnS layer
Driver layer	Layer2 – SitalSDK	- Negotiate with layer1 to verify devices communication - Translate RW from Application layer to AXI address RW
	Layer1 – SitalAPI	- RW via AXI to and from IP core

The software image deployed to the target platform is based on target-specific StandAlone BSP.
The flow of creation is:

- Export Vivado hardware project AXI Master, AXI Slave ARINC825 (and PMOD socket Configuration for transceiver connection), targeting for the platform you are working on

For the software components; User application and driver layers

- Create Xilinx SDK StandAlone Application and Library project
  Note: User application uses Driver Layer library.

NOTE – for settings details of the various **projects described, see chapter 3 of this document.**

2.2 Delivery Package

The delivery Package contains the following projects' settings and the complete source code

- Layer2 – (Driver Layer) Library project
- Layer2 - Driver documentation
- Vivado project for Avnet UltraZed-EG SOM with UltraZed_EG IO Carrier Card
- Test – User application

3 Environment and Settings

Notes: –

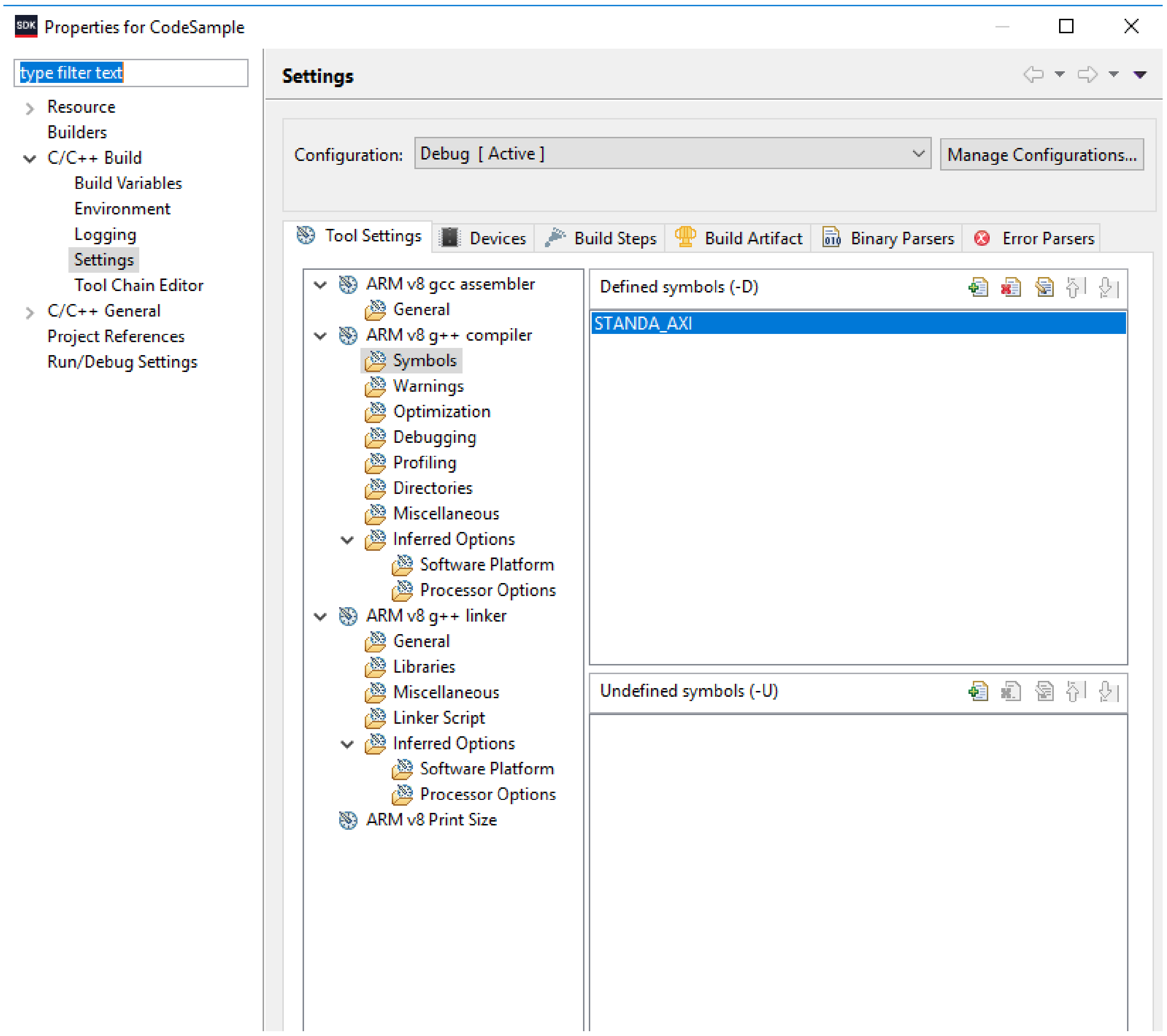
1. Current StandAlone AXI version was developed and tested on a Xilinx ZYNQ evaluation board consisting of an Avnet UltraZed-EG SOM System on Module and an UltraZed_EG IO Carrier Card. This used StandAlone Xilinx SDK Release Version 2018.2.
2. All projects source code and environments are delivered as a reference; See section 2.1.

3.1 Required Settings and Files by Application and Library projects

General Note: For all Test Application and Library projects, add the Symbol

- STANDA_AXI - for AXI ARINC825 Driver

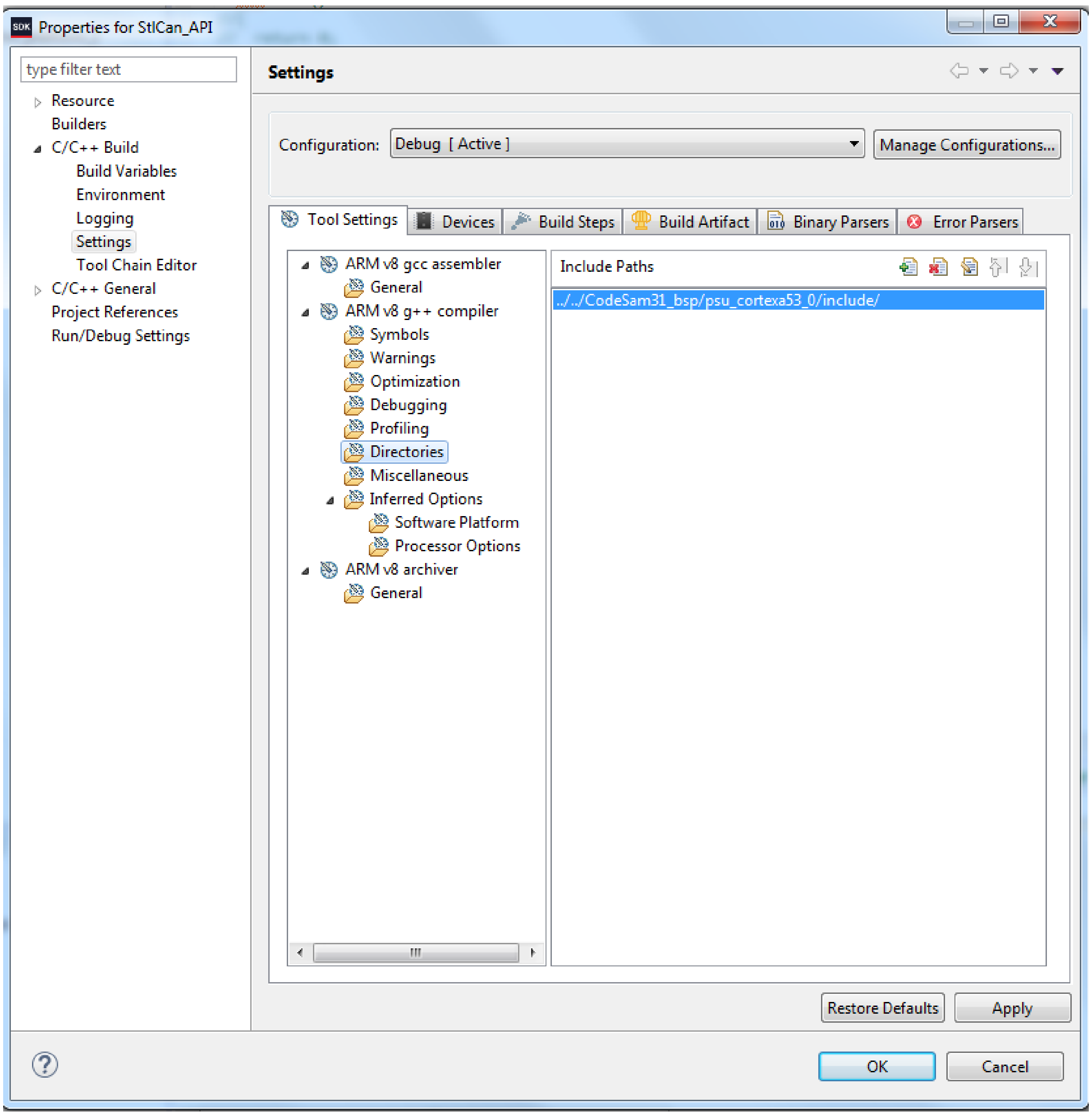
To do this, go to project properties: Project → Properties -> C/C++ Build -> Settings -> Symbols -> "STANDA_AXI" (-D is added automatically by SDK).



To use BSP .h files in a library project, one needs to go to project properties: Project → Properties -> C/C++ Build -> Settings -> Directories and to add the following path –

../../CodeSam31_bsp/psu_cortexa53_0/include/

Where CodeSam31_bsp – BSP name (CodeSam207_bsp – in latest delivery).


Properties for StlCan_API

Settings

Configuration: Debug [Active]

Include Paths

../../CodeSam31_bsp/psu_cortexa53_0/include/

3.1.1 Driver – Layer2 and Layer1 files

For the driver layer project, import the following files:

can4linux.h
CommonTypes.h
ReturnCodes.h
SitalAPI.h
SitalCan_API.cpp
SitalCan_API.h
SitalCan_SDK.cpp

3.1.2 SnS API Layer files

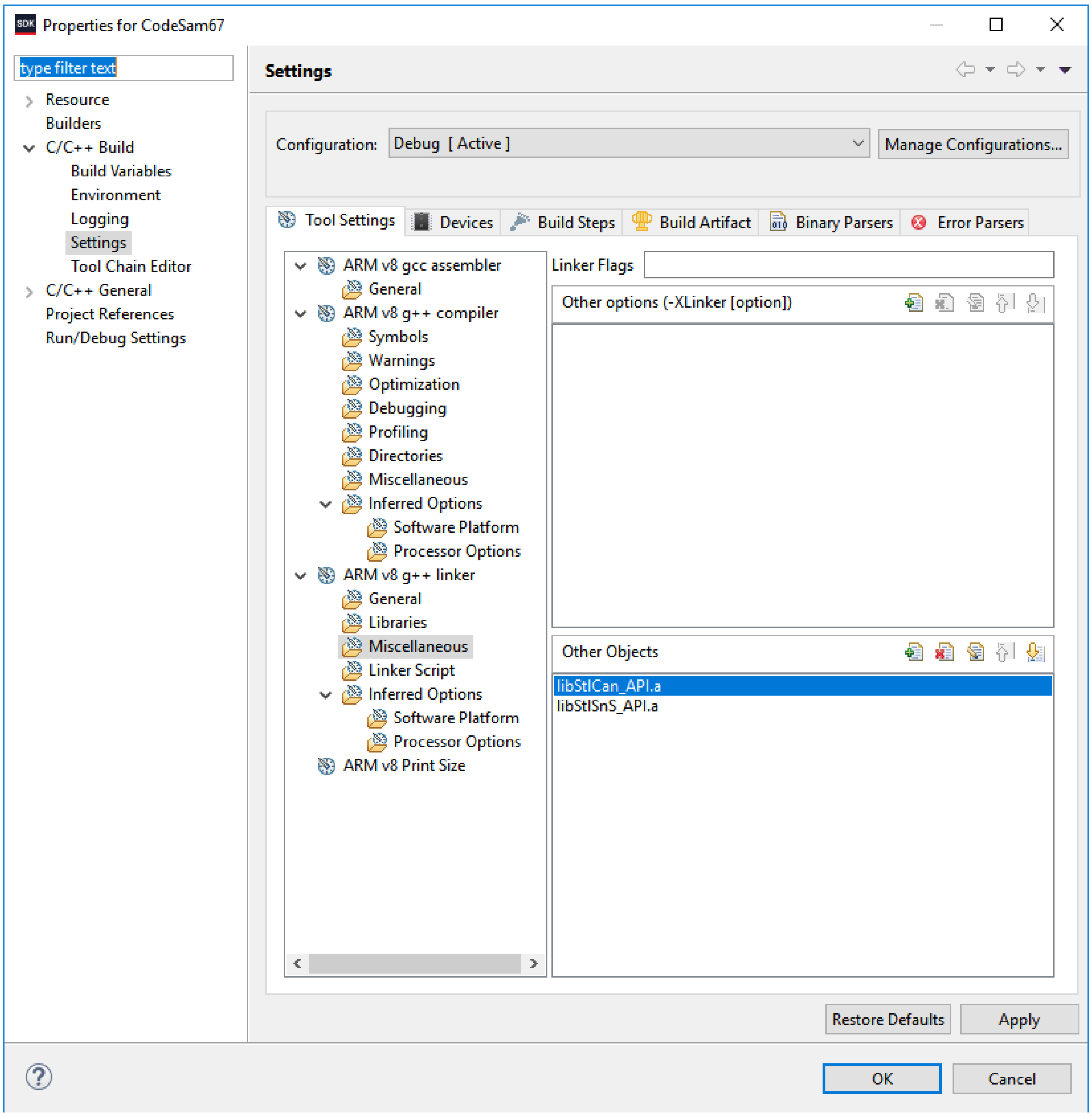
For the SnS layer project, import the following files:

can4linux.h
CommonTypes.h
ReturnCodes.h
SitalCan_API.h
SitalSnS_API.cpp

3.1.3 User Application – test

-

- Make Linker to use Driver Layer Library Project and SnS Layer Library project


Properties for CodeSam67

Settings

Configuration: Debug [Active]

Tool Settings | Devices | Build Steps | Build Artifact | Binary Parsers | Error Parsers

ARM v8 g++ linker > Miscellaneous

Linker Flags:

Other options (-XLinker [option])

Other Objects:

libStICan_API.a
libStISnS_API.a

4 Driver Layer – Layer2 & Layer1

4.1 CAN API Driver

Xilinx SDK 2018.2 StandAlone AXI driver is used as low level driver in accordance with Vivado Hardware project.

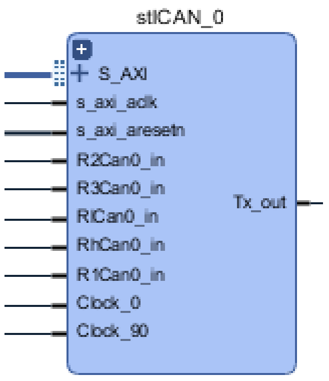
A handle is created and any call to the driver layer is translated to this handle internally.

A general function ‘`sitalDevice_AccessMemory(..)`’ is being used widely by the CAN API to access the driver layer’s read and write to registers and memory sections.

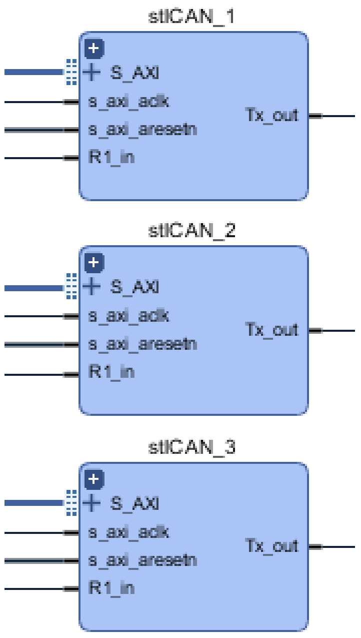
5 Vivado Project

5.1 Vivado project Top modules

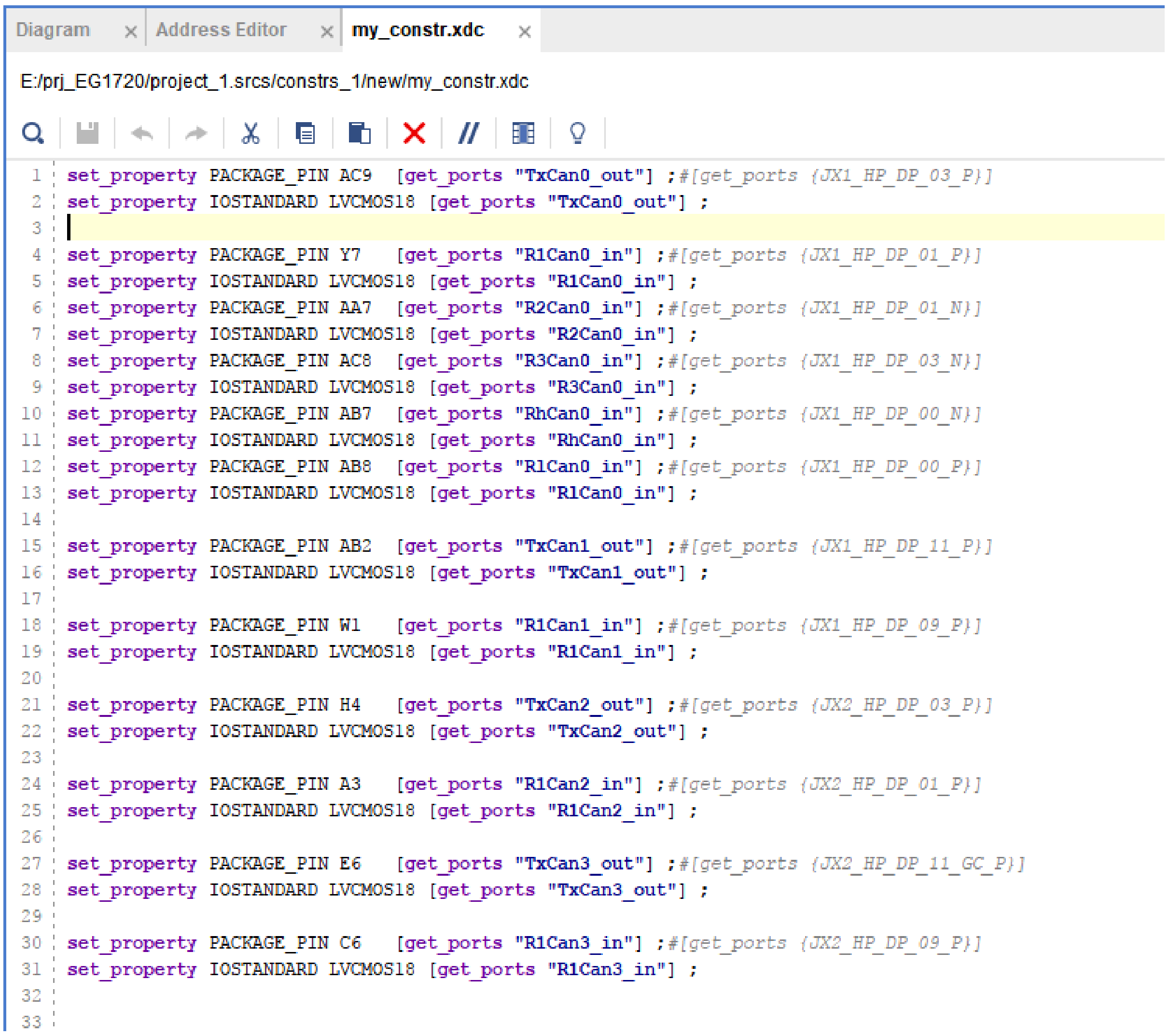
1. SnS CAN Module – SnS Sital Transceiver is used in Testing System as SnS Receiver



2. Three CAN Modules CAN/CAN FD Sital Transceivers are used in Testing System as CAN/CAN FD Transmitters (Sequencers are configured)



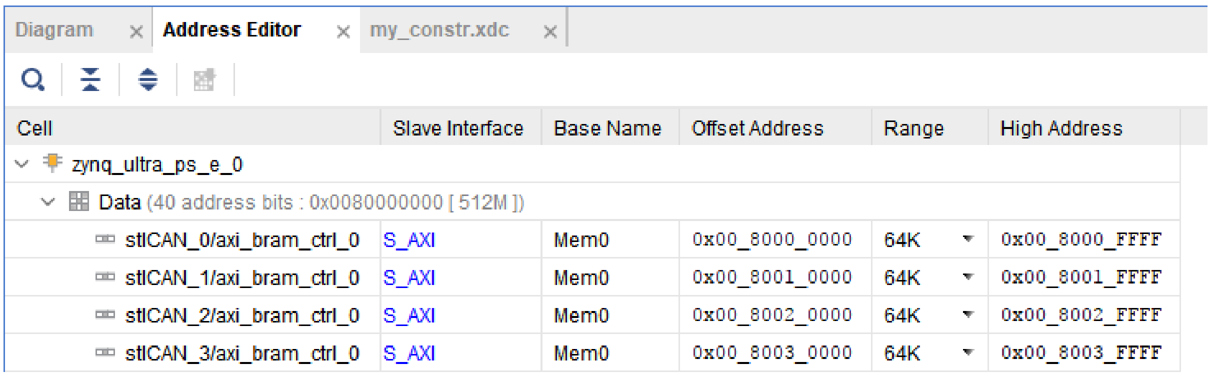
5.2 Vivado project Output/Input Ports



E:/prj_EG1720/project_1.srcs/constrs_1/new/my_constr.xdc

```
set_property PACKAGE_PIN AC9  [get_ports "TxCan0_out"] ;#[get_ports {JX1_HP_DP_03_P}]
set_property IOSTANDARD LVCMOS18 [get_ports "TxCan0_out"] ;

set_property PACKAGE_PIN Y7   [get_ports "R1Can0_in"] ;#[get_ports {JX1_HP_DP_01_P}]
set_property IOSTANDARD LVCMOS18 [get_ports "R1Can0_in"] ;
set_property PACKAGE_PIN AA7  [get_ports "R2Can0_in"] ;#[get_ports {JX1_HP_DP_01_N}]
set_property IOSTANDARD LVCMOS18 [get_ports "R2Can0_in"] ;
set_property PACKAGE_PIN AC8  [get_ports "R3Can0_in"] ;#[get_ports {JX1_HP_DP_03_N}]
set_property IOSTANDARD LVCMOS18 [get_ports "R3Can0_in"] ;
set_property PACKAGE_PIN AB7  [get_ports "RhCan0_in"] ;#[get_ports {JX1_HP_DP_00_N}]
set_property IOSTANDARD LVCMOS18 [get_ports "RhCan0_in"] ;
set_property PACKAGE_PIN AB8  [get_ports "R1Can0_in"] ;#[get_ports {JX1_HP_DP_00_P}]
set_property IOSTANDARD LVCMOS18 [get_ports "R1Can0_in"] ;

set_property PACKAGE_PIN AB2  [get_ports "TxCan1_out"] ;#[get_ports {JX1_HP_DP_11_P}]
set_property IOSTANDARD LVCMOS18 [get_ports "TxCan1_out"] ;

set_property PACKAGE_PIN W1   [get_ports "R1Can1_in"] ;#[get_ports {JX1_HP_DP_09_P}]
set_property IOSTANDARD LVCMOS18 [get_ports "R1Can1_in"] ;

set_property PACKAGE_PIN H4   [get_ports "TxCan2_out"] ;#[get_ports {JX2_HP_DP_03_P}]
set_property IOSTANDARD LVCMOS18 [get_ports "TxCan2_out"] ;

set_property PACKAGE_PIN A3   [get_ports "R1Can2_in"] ;#[get_ports {JX2_HP_DP_01_P}]
set_property IOSTANDARD LVCMOS18 [get_ports "R1Can2_in"] ;

set_property PACKAGE_PIN E6   [get_ports "TxCan3_out"] ;#[get_ports {JX2_HP_DP_11_GC_P}]
set_property IOSTANDARD LVCMOS18 [get_ports "TxCan3_out"] ;

set_property PACKAGE_PIN C6   [get_ports "R1Can3_in"] ;#[get_ports {JX2_HP_DP_09_P}]
set_property IOSTANDARD LVCMOS18 [get_ports "R1Can3_in"] ;
```

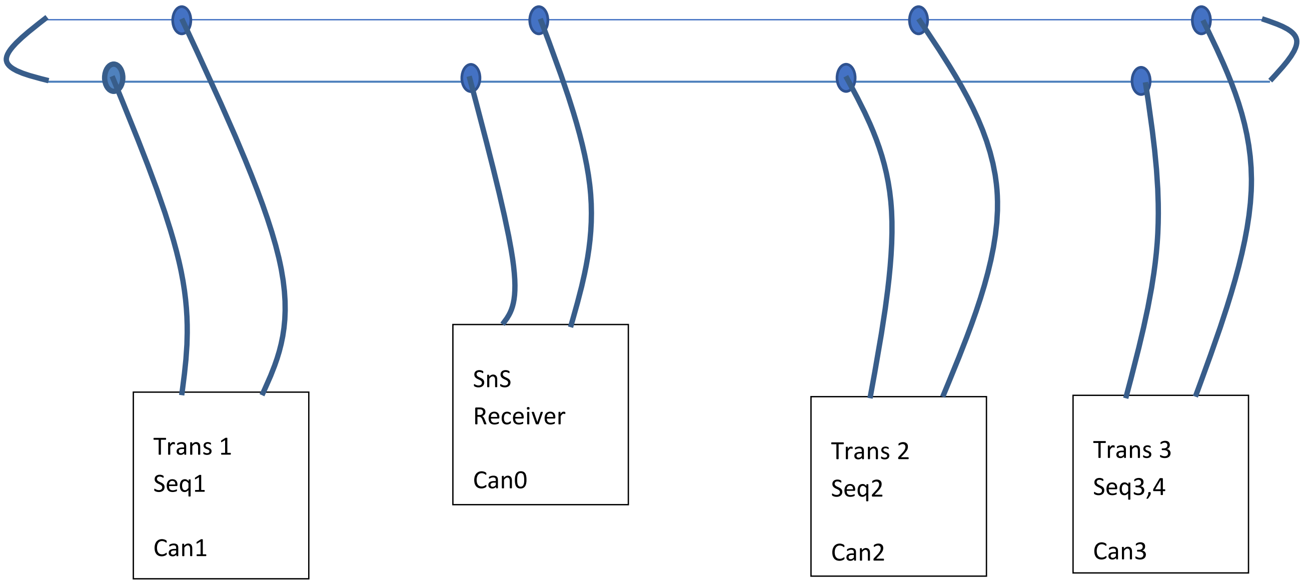
5.3 Vivado project Top modules Addresses



Cell	Slave Interface	Base Name	Offset Address	Range	High Address
zynq_ultra_ps_e_0					
Data (40 address bits : 0x0080000000 [512M])					
stlCAN_0/axi_bram_ctrl_0	S_AXI	Mem0	0x00_8000_0000	64K	0x00_8000_FFFF
stlCAN_1/axi_bram_ctrl_0	S_AXI	Mem0	0x00_8001_0000	64K	0x00_8001_FFFF
stlCAN_2/axi_bram_ctrl_0	S_AXI	Mem0	0x00_8002_0000	64K	0x00_8002_FFFF
stlCAN_3/axi_bram_ctrl_0	S_AXI	Mem0	0x00_8003_0000	64K	0x00_8003_FFFF

6 Testing Schem and Environment

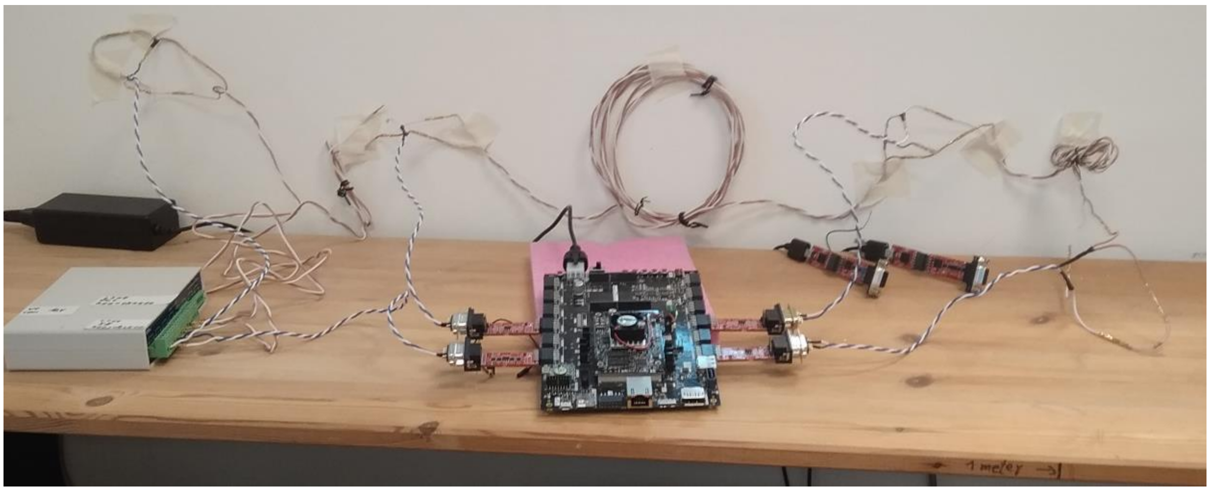
6.1 Network Schem



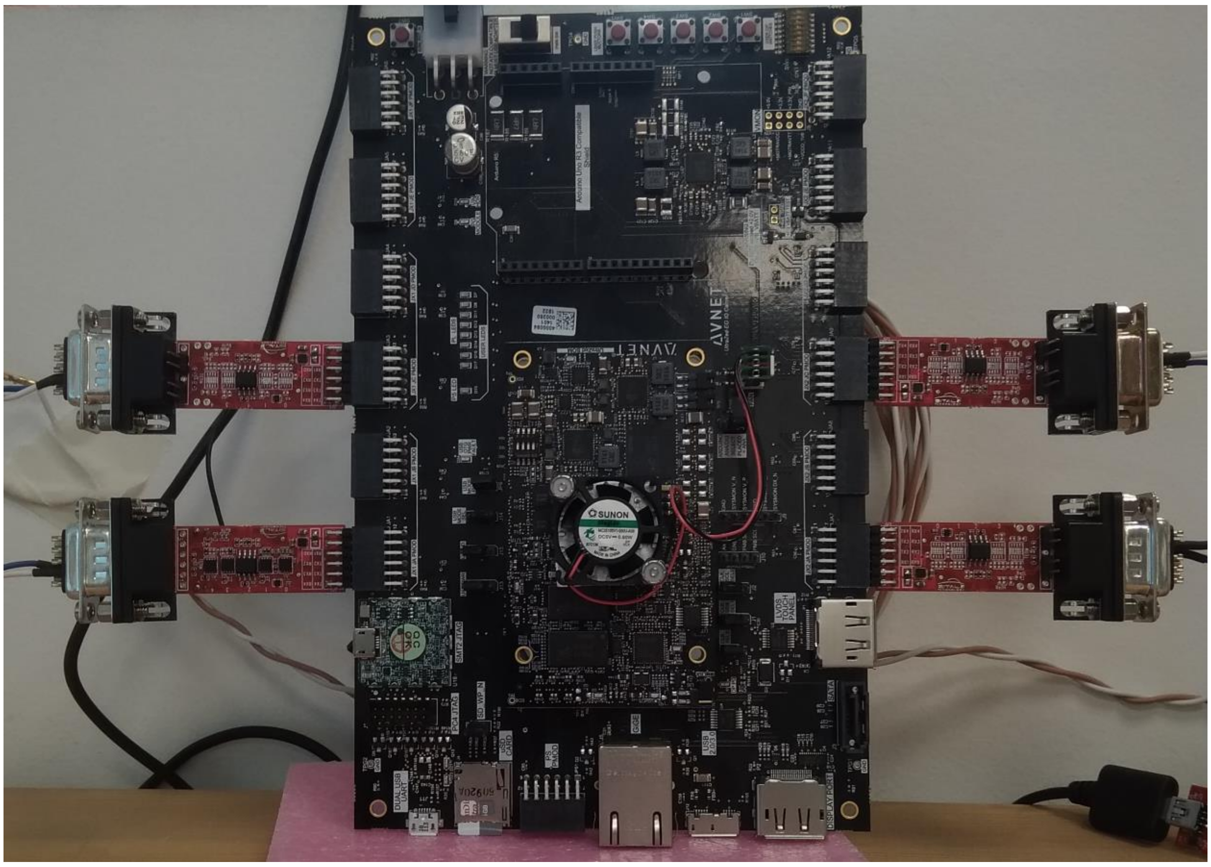
6.2 Board, Transceivers, Network

Testing system consists of

- Evaluation board - Avnet UltraZed-EG SOM with UltraZed_EG IO Carrier Card
- One Sital PMOD SnS CAN/CAN FD Transceiver (used as receiver only)
- Three Sital PMOD CAN/CAN FD Transceivers (used as transmitters only)
- CAN/CAN FD Network (changeable configuration)
- Relays switch for disconnections simulation

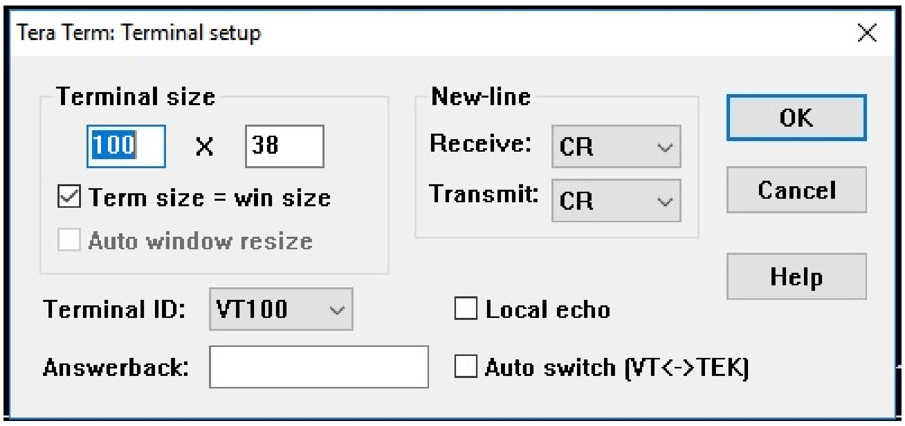


Please, use the following picture for right placement of SnS Transceiver – on the left bottom.



6.3 Terminal Program

Tera Term is used as a Terminal program for Xilinx SDK environment. Terminal ID is VT100 and Terminal size must be more than 80x24.



Tera Term: Terminal setup

Terminal size: 100 x 38

☑ Term size = win size

☐ Auto window resize

New-line

Receive: CR

Transmit: CR

Terminal ID: VT100

Answerback:

☐ Local echo

☐ Auto switch (VT<->TEK)

7 Test Application

7.1 Program Description

First, `sitalCan_DeviceInit()` function should be called for every device (CAN IP Core) to be used with baseAddress, Low - baseRate and High – dataRate parameters.

Then, `sitalCan_OpenDevices()` function should be called only once for all devices

The following fragment sets Sequencer for CAN FD Message from deviceId1 -

```
 msgId1 = 0x11111111;

tx_msg.MsgExtended = TRUE;
tx_msg.MsgId = msgId1;
tx_msg.MsgLength = 64;

for (int i=0; i<CAN_MSG_LENGTH; i++)
{
        tx_msg.MsgData[i] = 1 + i;
}
sitalCan_SetSeqEntryMsg(deviceId1, CAN_FD_PROT, seqInd1, 100, 1, &tx_msg);   // 1000 - 1 sec
```

sitalCan_EnableSequencers() function should be called for certain device to Enable(Disable) all the sequencers configured for this device.

One of the initialized CAN devices can be determine as SnS Receiver device using th following function – `sitalSnS_Initialize(deviceId0).`

The following fragment (simplified) means the core is continuously receiving LEARNING_MSG_NUMBER messages – for the Learning Phase and TRACKING_MSG_NUMBER messages – for the Tracking Phase.

Following every 1,000 messages, a dot '.' is printed.

```
for (int count=0; count< LEARNING_MSG_NUMBER; count++)
{
        while(1)
        {
                swResult = sitalCan_GetNextMsg(deviceId0, &msgReceived, &RecMsg);
                if (sitalReturnCode_SUCCESS != swResult)
                {
                        xil_printf("\r\n  get swResult = %d \r\n", swResult);
                        return swResult;
                }
                if (msgReceived) break;
        }
        if (count%1000 == 0)
                xil_printf(".");

        sitalSnS_MsgLearning (deviceId0, &RecMsg, &msgIdIndex,      &calculatedDeviation,
                                        &bParticipation);
}
xil_printf(" \n\r Learning phase is completed");
xil_printf(" \n\r Please, run script of disconnects simulation. Sleep for %d seconds", sleepSeconds);
sleep(sleepSeconds);

for (int count=0; count< TRACKING_MSG_NUMBER; count++)
{
        while(1)
        {
                swResult = sitalCan_GetNextMsg(deviceId0, &msgReceived, &RecMsg);
                if (sitalReturnCode_SUCCESS != swResult)
                {
                        xil_printf("\r\n  get swResult = %d \r\n", swResult);
                        return swResult;
                }
                if (msgReceived) break;
        }
        if (count%1000 == 0)
                xil_printf(".");

        sitalSnS_MsgTracking (deviceId0, &RecMsg, &msgIdIndex,
                                        &apprSnSEntryIndex, &calculatedDeviation);
}
xil_printf(" \n\r Tracking phase is completed");

sitalSnS_FaultIsolation (deviceId0, &msgNonProblemNumber, MsgNonProblemArray,
                                        &msgProblemNumber, MsgProblemArray);
```

7.2 Test Results

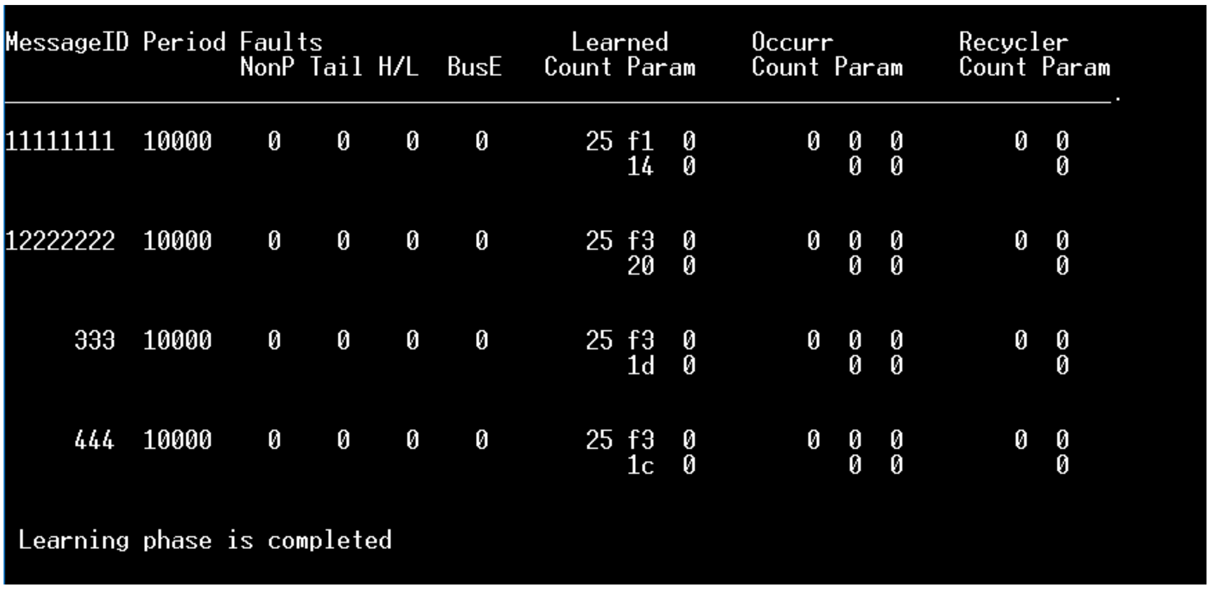
The following results are also copied to **TerminalPrint.txt** file wich is included in the Delivery package.

1. **Filter functionality check –**

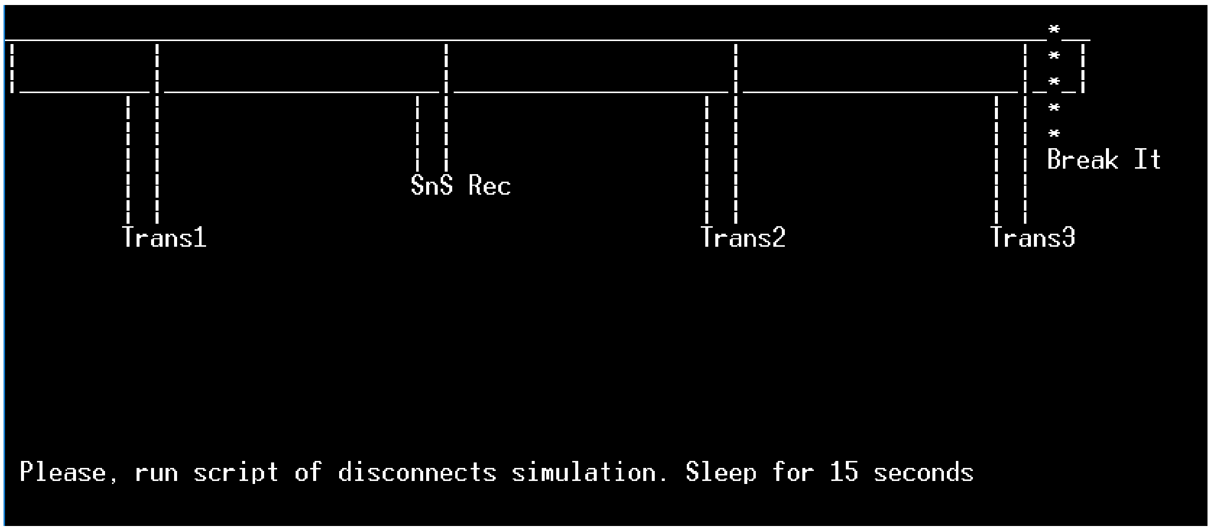
```
     133   293275   8 12 23 34 45 56 67 78 89
11111444   303276   1 cc
     133   313277   8 12 23 34 45 56 67 78 89
 No Message  11111444
```

2. **Wiring Fault functionality check –**

After Learning phase -

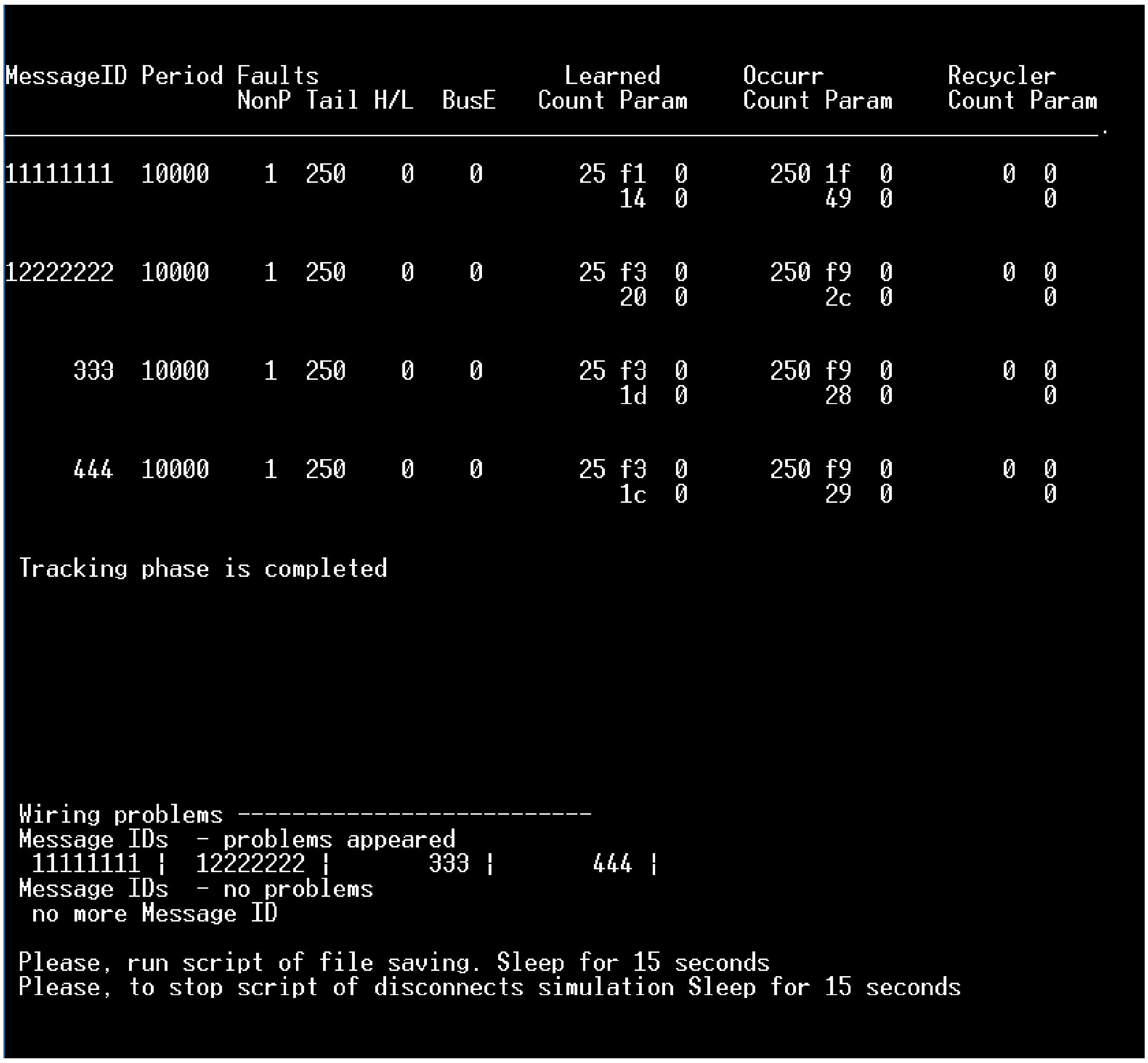


```
MessageID Period Faults                      Learned        Occurr         Recycler
                 NonP Tail H/L   BusE      Count Param    Count Param    Count Param
________________________________________________________________________________.

11111111  10000    0    0    0    0          25 f1  0         0  0  0        0  0
                                                14  0            0  0           0

12222222  10000    0    0    0    0          25 f3  0         0  0  0        0  0
                                                20  0            0  0           0

     333  10000    0    0    0    0          25 f3  0         0  0  0        0  0
                                                1d  0            0  0           0

     444  10000    0    0    0    0          25 f3  0         0  0  0        0  0
                                                1c  0            0  0           0

 Learning phase is completed
```



```
 Please, run script of disconnects simulation. Sleep for 15 seconds
```

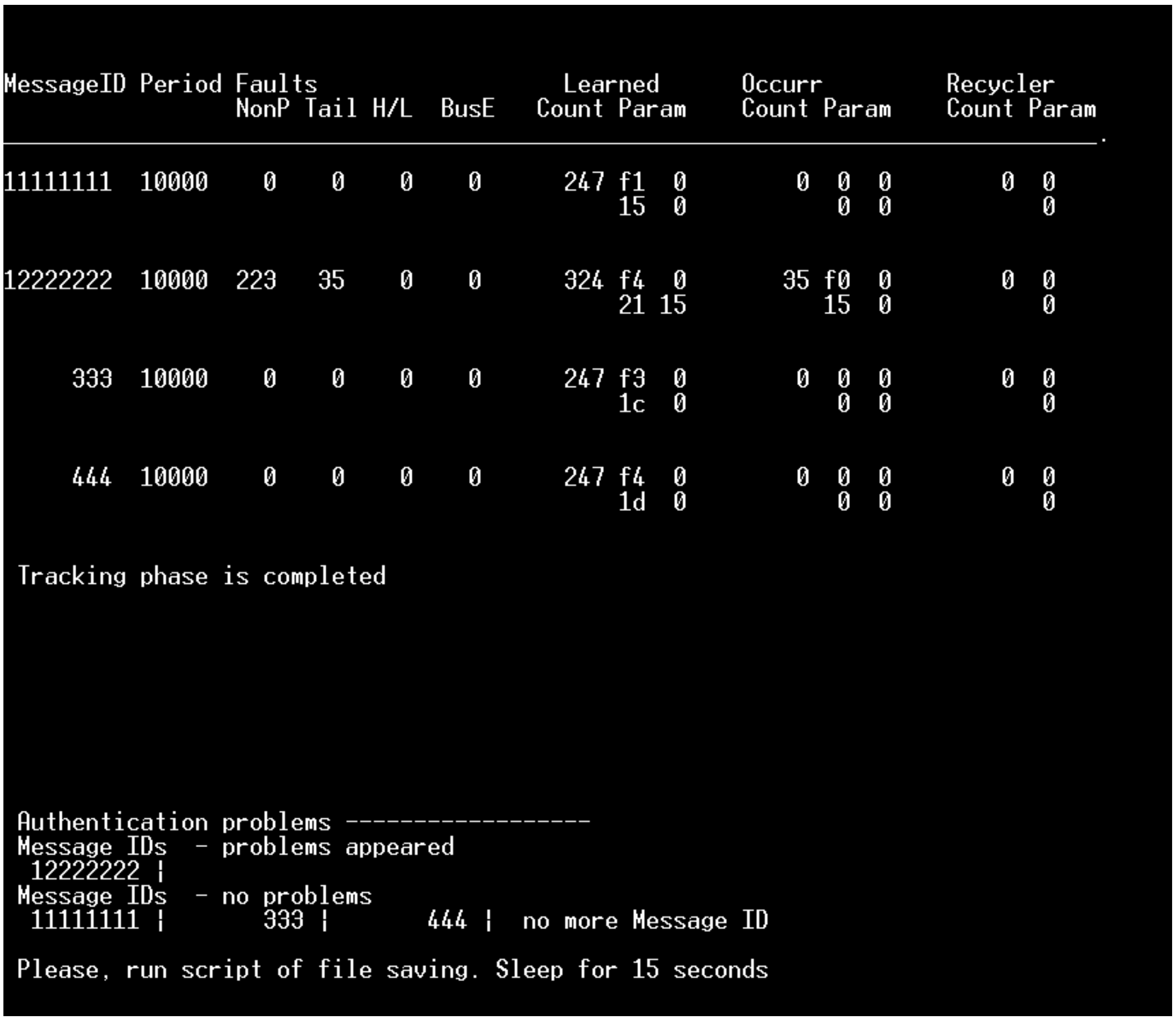
After Tracking phase -



```
MessageID Period Faults                    Learned       Occurr         Recycler
                 NonP Tail H/L   BusE     Count  Param   Count  Param   Count  Param
_____________________________________________________________________________________.

11111111  10000    1  250    0     0       25 f1   0      250 1f   0       0   0
                                              14   0          49   0           0


12222222  10000    1  250    0     0       25 f3   0      250 f9   0       0   0
                                              20   0          2c   0           0


     333  10000    1  250    0     0       25 f3   0      250 f9   0       0   0
                                              1d   0          28   0           0


     444  10000    1  250    0     0       25 f3   0      250 f9   0       0   0
                                              1c   0          29   0           0


 Tracking phase is completed









 Wiring problems --------------------------
 Message IDs  - problems appeared
  11111111 |  12222222 |       333 |       444 |
 Message IDs  - no problems
  no more Message ID

 Please, run script of file saving. Sleep for 15 seconds
 Please, to stop script of disconnects simulation Sleep for 15 seconds
```

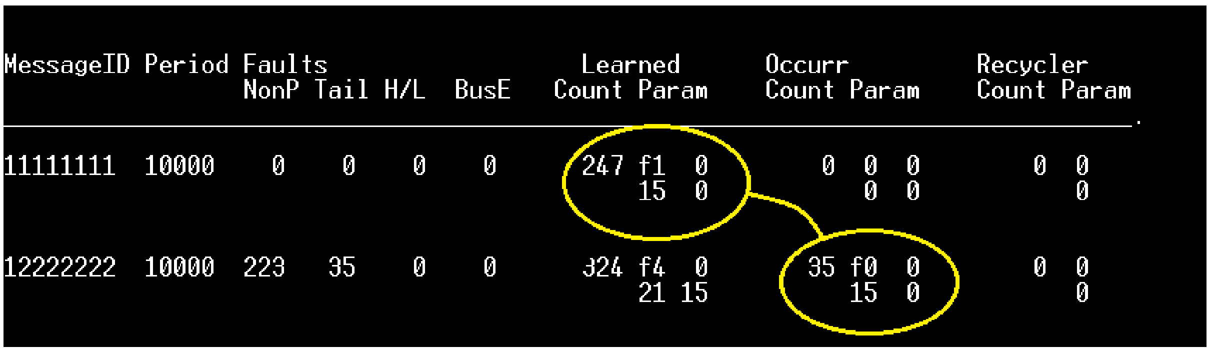
3. Authentication fault functionality check

The Learning phase has exactly the same result.
After Tracking phase -



```
MessageID Period Faults                    Learned       Occurr        Recycler
                 NonP Tail H/L   BusE    Count Param   Count Param   Count Param
_______________________________________________________________________________.
11111111  10000    0    0    0    0        247 f1  0       0  0  0       0  0
                                                15  0          0  0          0

12222222  10000  223   35    0    0        324 f4  0      35 f0  0       0  0
                                                21 15         15  0          0

     333  10000    0    0    0    0        247 f3  0       0  0  0       0  0
                                                1c  0          0  0          0

     444  10000    0    0    0    0        247 f4  0       0  0  0       0  0
                                                1d  0          0  0          0

 Tracking phase is completed




 Authentication problems ------------------
 Message IDs   - problems appeared
  12222222 ¦
 Message IDs   - no problems
  11111111 ¦        333 ¦        444 ¦   no more Message ID

 Please, run script of file saving. Sleep for 15 seconds
```

Some Message of the origin, where Message ID 0x11111111 was sent from, is sending now by the origin of the Message with ID 0x12222222 –



```
MessageID Period Faults                    Learned       Occurr        Recycler
                 NonP Tail H/L   BusE    Count Param   Count Param   Count Param
_______________________________________________________________________________.
11111111  10000    0    0    0    0        247 f1  0       0  0  0       0  0
                                                15  0          0  0          0

12222222  10000  223   35    0    0        324 f4  0      35 f0  0       0  0
                                                21 15         15  0          0
```

17 Atir Yeda St., Kfar-Saba, ISRAEL 44643

Email: info@sitaltech.com

Website: http://www.sitaltech.com