Quick Start Guide

QUICK START GUIDE FOR IMX93

# PROPRIETARY NOTICE:

This document contains proprietary material for the sole use of the intended recipient(s). Do not read this document if you are not the intended recipient. Any review, use, distribution or disclosurenot the intended recipient (or authorized to receive for the recipient), you are herebynotified that any disclosure, copying distribution or use of any of the information contained within this document is STRICTLY PROHIBITED.

Thank you.

# Table of Contents

[INTRODUCTION 3](#_bookmark0)

[Building the](#_bookmark0) Yocto source [....................................................................................... 4](#_bookmark0)

Guide to flashing the Yocto Source7

Build & Flash

Quick Start Guide

# INTRODUCTION

**Purpose and Scope**

This document is intended to assist the engineer in programming the compiled binaries onto the IMX93 EVK board and in validating the peripherals and functionalities associated with the IMX93 board.

#### Acronyms

The following acronyms will be used throughout the document

|  |  |
| --- | --- |
| Acronym | Abbreviation |
| USB | Universal Serial Bus |
| UUU | Universal Update Utility |
| GB | GigaByte |
| OS | Operating System |
| SOM | System On Module |
| eMMC | embedded Multi-Media Card |
| RAM | Random Access Memory |
| CAN | Controller Area Network |

**Peripherals/Functionality Validation**

The Peripheral/Functionality which are used and tested are:

**1. CAN2**

**Note:**

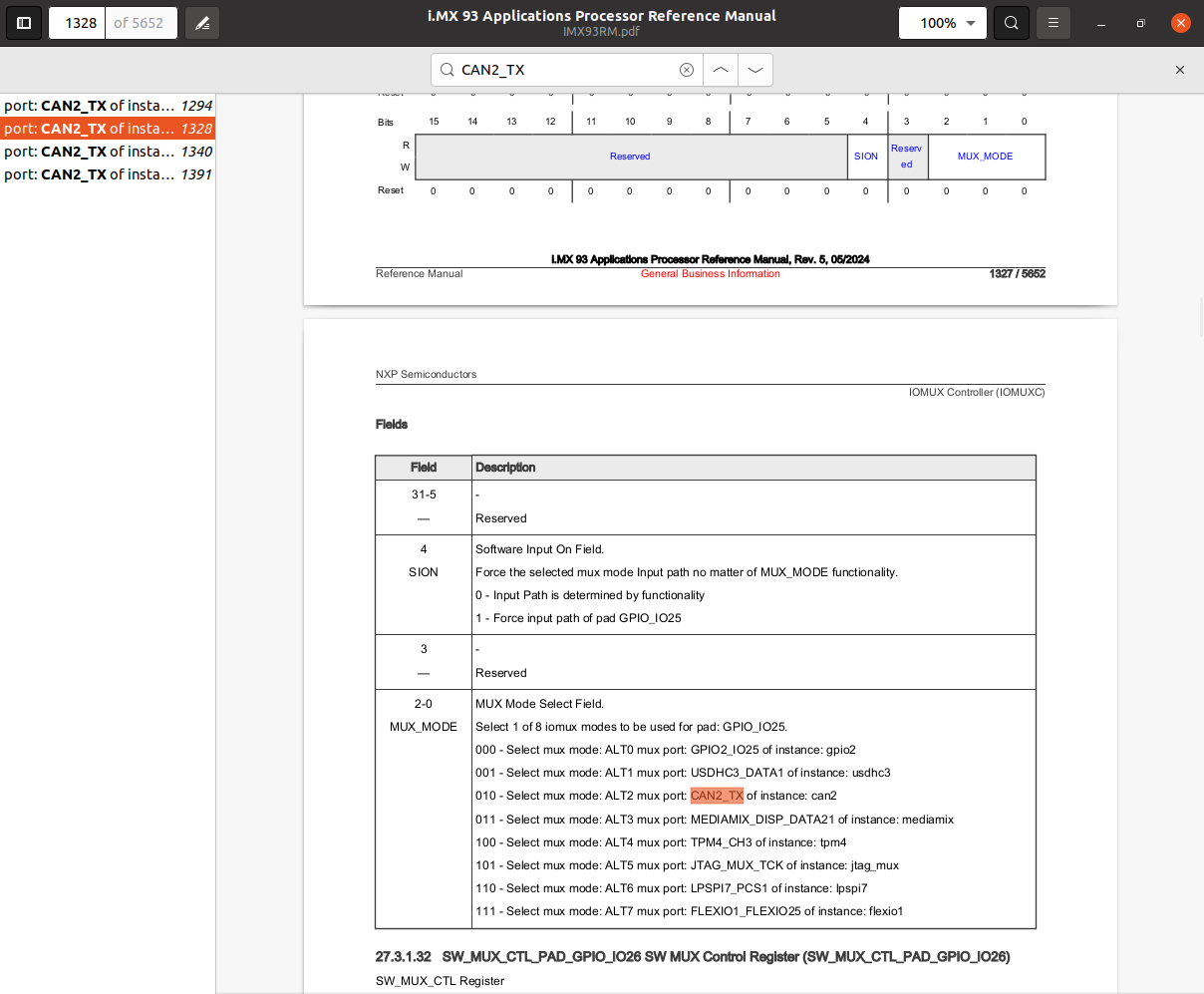
Board Login: root

# **FLEXCAN on IMX93**

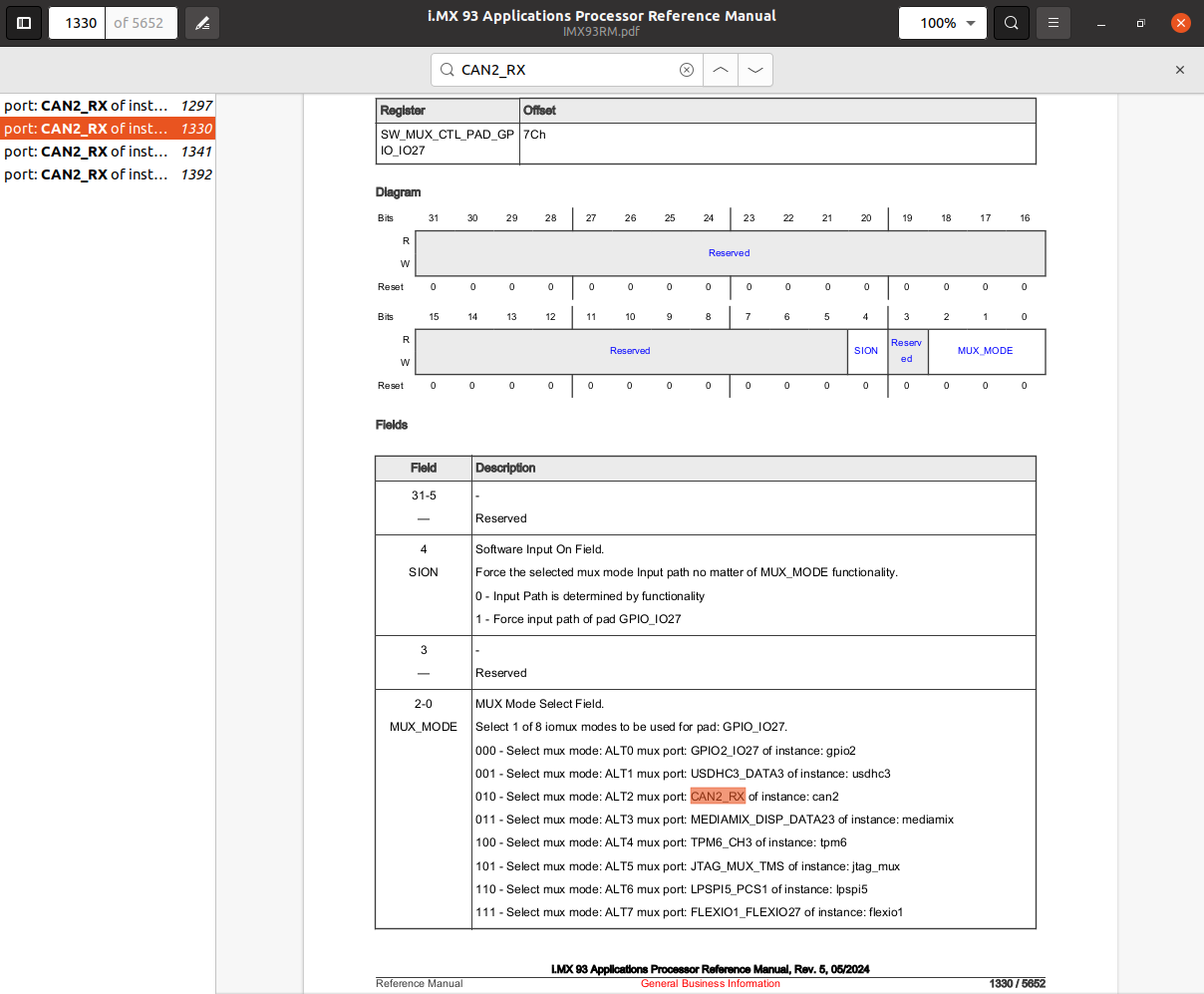
There are two pins to be configured for CAN2 interface:

* CAN2\_TX -GPIO\_IO25.
* CAN2\_RX -GPIO\_I027.

SW\_MUX\_CTL\_PAD\_GPIO\_IO25 SW MUX Control Register (SW\_MUX\_CTL\_PAD\_GPIO\_IO25)



SW\_MUX\_CTL\_PAD\_GPIO\_IO25 SW MUX Control Register (SW\_MUX\_CTL\_PAD\_GPIO\_IO25)



# STEP1:**Changes to Device Tree File for FLEXCAN2**

* IMX93 EVK does not have an onboard CAN transceiver.Add the transceiver configuration for **flexcan2** to enable the transceiver.
* **Access the DTS File:** Navigate to the device tree source file for the IMX93 EVK board.

**$ cd ~/IMX93/build-imx93-evk/tmp/work/imx93evk-poky-linux/linux-imx/6.6.23+git/git/arch/arm64/boot/dts/freescale**

**$ vi imx93-11x11-evk.dts**

* **Enable the CAN Transceiver:** By default the flexcan2 status is **“okay”.**

+ transceiver1: can-phy0 {

+ compatible = "ti,tcan1043";

+ #phy-cells = <0>;

+ max-bitrate = <5000000>;

+ };

&flexcan2 {

pinctrl-names = "default", "sleep";

pinctrl-0 = <&pinctrl\_flexcan2>;

pinctrl-1 = <&pinctrl\_flexcan2\_sleep>;

xceiver-supply = <&reg\_can2\_stby>;

status = "okay";

+ phys = <&transceiver1>;

};

* **Remove the Standby GPIO Pin Configuration:**

reg\_can2\_stby: regulator-can2-stby {

compatible = "regulator-fixed";

regulator-name = "can2-stby";

regulator-min-microvolt = <3300000>;

regulator-max-microvolt = <3300000>;

- gpio = <&adp5585gpio 5 GPIO\_ACTIVE\_LOW>;

enable-active-low;

};

* **Update the** local.conf **File:** Add the following line to the local.conf file to install iproute2 package.

**$ cd ~/IMX93/build-imx93-evk/conf**

**$ IMAGE\_INSTALL:append = " iproute2"**

**STEP2: Install and Set Up CANdo**

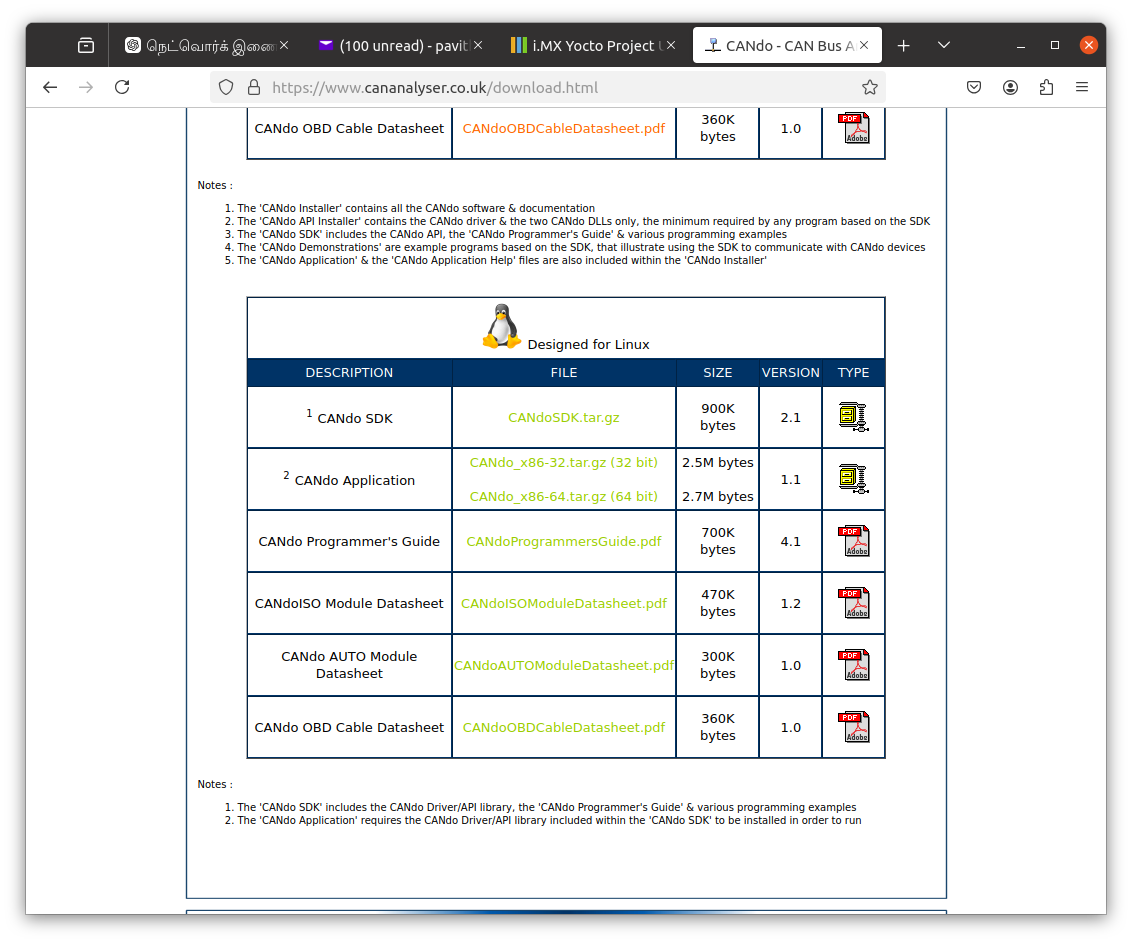
**Download the CANdo Application & CANdo SDK:**

Go to the following link to download the **CANdo** application & CANdo SDK

<https://www.cananalyser.co.uk/download.html>

**CANdo Application** (My system, e.g., CANdo\_x86-64.tar.gz)

**CANdo SDK** (for compiling drivers and using the SDK)



**Install the CANdo Application:**

* **Extract the** CANdo\_x86-64.tar.gz **file.**

**$ cd ~/Downloads**

**$ tar -xvzf CANdo\_x86-64.tar.gz**

* **Make the** CANdo **application executable.**

**$ chmod +x CANdo**

* **Install the CANdo driver:Extract the** CANdoSDK.tar.gz **file.**

**$ tar -xvzf CANdoSDK.tar.gz**

**$ cd Driver/Source/**

**$ make**

**$ sudo cp libCANdo.so /usr/lib/**

* **Attach the CAN device:** Now, attach the CAN device (connected via USB) to the system:

**$ cd ~/Downloads/**

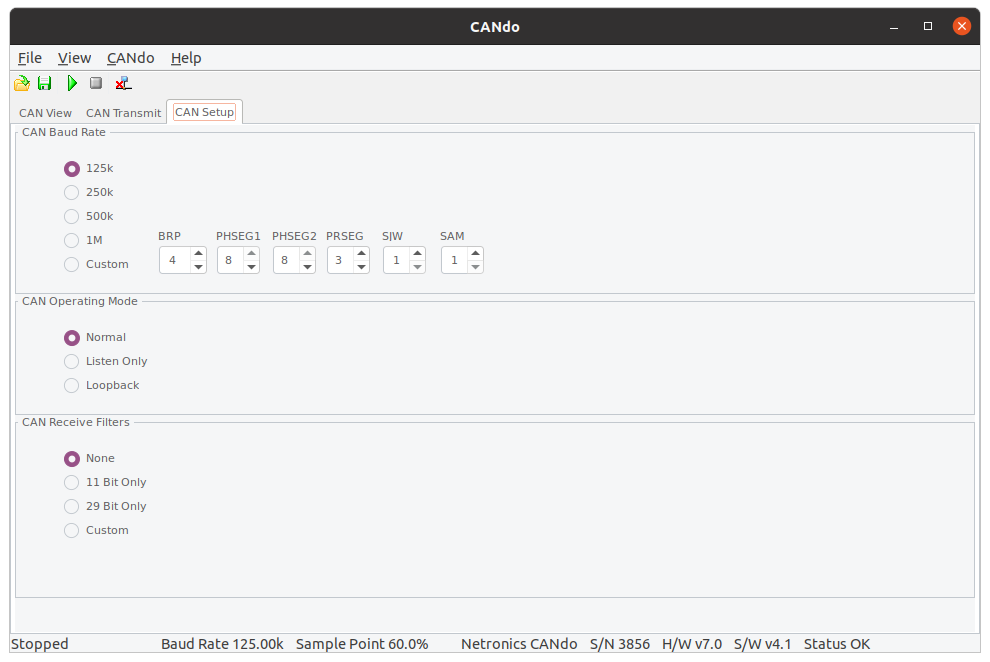
**$ sudo slcan\_attach -o /dev/ttyUSB0**

**attached tty /dev/ttyUSB0 to netdevice slcan0**

Open the Application by running the following command:

**$ sudo ./CANdo**

Once the application is open, go to “CAN setup” set the baud rate to “125k” as shown in the following image.



**STEP 3:Test the CAN Interface**

**Monitor Receive Data Using CANdo Application:**

* Once the baud rate is set, click on “**CAN View” to see the receive data& transmitted data .**To test the CAN interface on your i.MX93 board, follow these steps in the **board terminal**:
* Set the CAN interface to the correct bitrate (125000)

**$ ip link set can0 up type can bitrate 125000**

* Bring the CAN interface up

**$ ip link set can0 up**

* Send a test message on the CAN bus.

**$ cansend can0 123#DEADBEEF**

Eg:

root@imx93evk:~# cansend can0 123#DEADBEEF

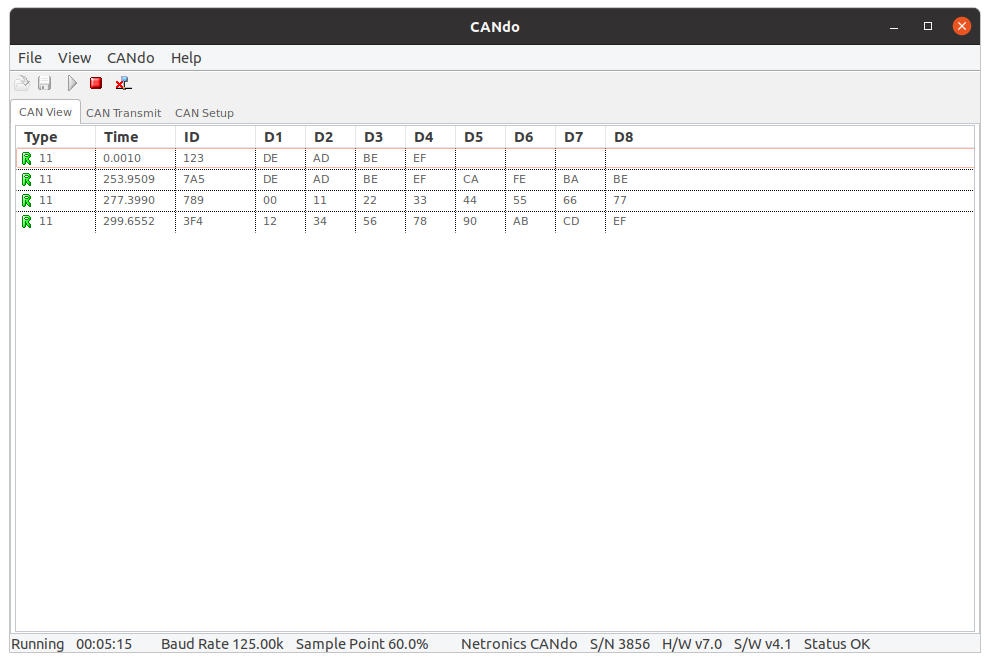
root@imx93evk:~# cansend can0 7A5#DEADBEEFCAFEBABE

root@imx93evk:~# cansend can0 789#0011223344556677

root@imx93evk:~# cansend can0 3F4#1234567890ABCDEF

* Clicking the start button (green symbol) in the CANdo application, the CAN message will begin transmitting, and the received data will be displayed in the application.

Below is a screenshot showing the verified receive data on the CANdo application terminal:



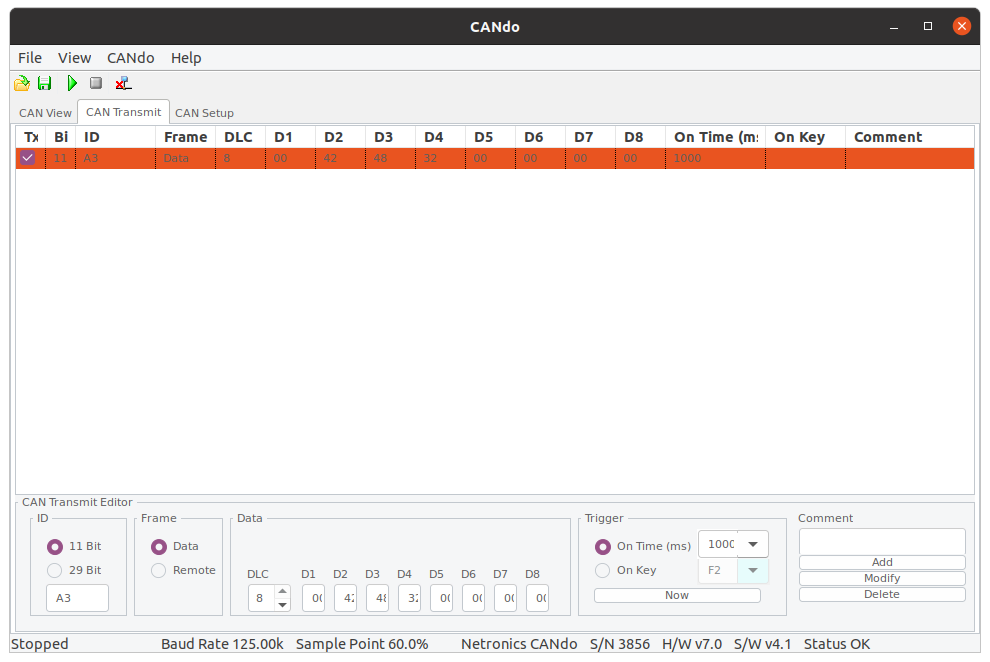
* **Stop the Data Transmission in CANdo:** Once the data is received and displayed, click the **stop button (red symbol)** to halt the transmission.

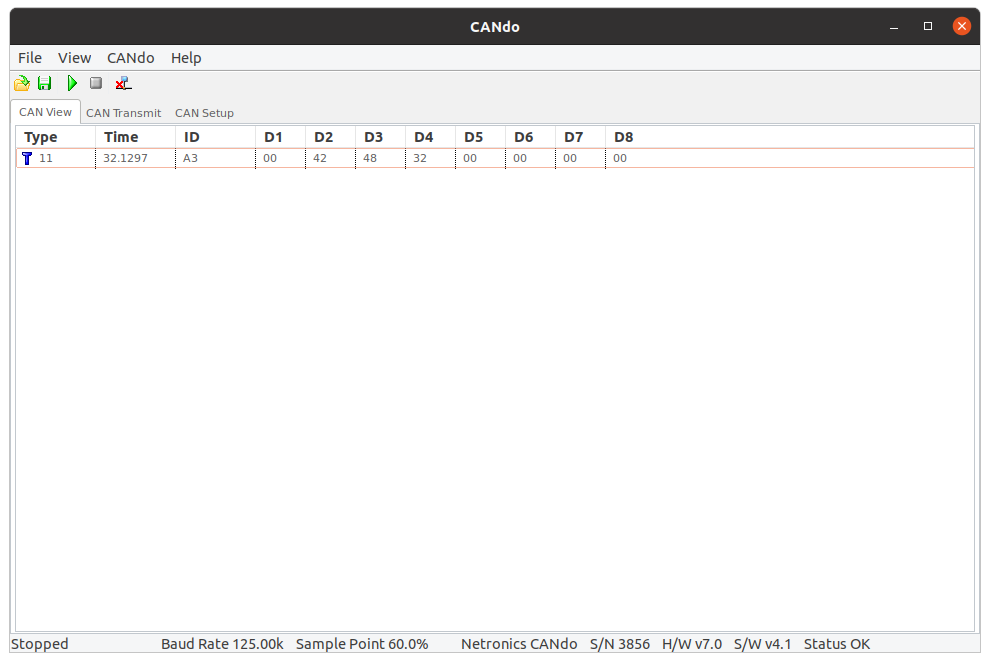
**Monitor the CAN Bus data on the board terminal:**

* **Run the** candump **Command:** To see the data received by the board

**$ candump can0**

* **Transmitting Data with CANdo Application:** In the **CANdo application**:
* Select the **ID as "11-bit"**.
* Enter the data to send.
* Click the **"Add"** option.
* Then click the **start button (green symbol)** to begin transmission.





* Same data we are able to see in Board terminal:

root@imx93evk:~# candump can0

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

can0 0A3 [8] 00 42 48 32 00 00 00 00

* Once the data is successfully received on the board terminal, stop the data transmission in CANdo by clicking the stop button (red symbol).
* bring down the CAN interface on the board using the following command:

**$ ip link set can0 down**