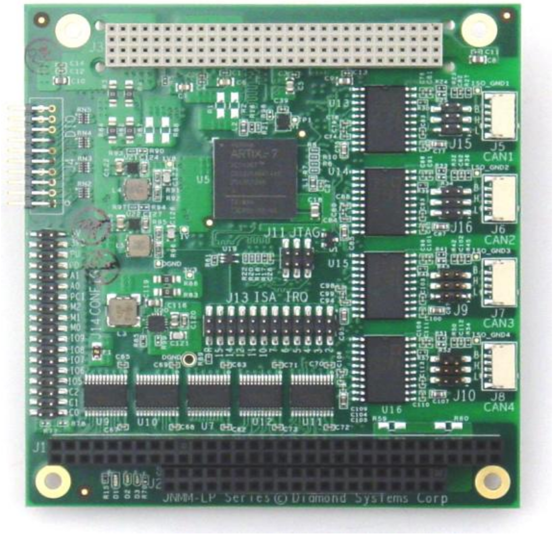
Janus-MM-4LP Linux Software User Manual

PC/104-*Plus* Dual or Quad CAN Port Module



Revision	Date	Comment
A.00	10/30/2015	Initial release

FOR TECHNICAL SUPPORT PLEASE CONTACT:

support@diamondsystems.com

Diamond Systems Corporation
555 Ellis Street
Mountain View, CA 94043 USA
Tel 1-650-810-2500
Fax 1-650-810-2525
www.diamondsystems.com

CONTENTS

1. IMPORTANT SAFE HANDLING INFORMATION

WARNING!

ESD-Sensitive Electronic Equipment

Observe ESD-safe handling procedures when working with this product.

Always use this product in a properly grounded work area and wear appropriate ESD-preventive clothing and/or accessories.

Always store this product in ESD-protective packaging when not in use.

Safe Handling Precautions

This board contains a high density connector with many connections to sensitive electronic components. This creates many opportunities for accidental damage during handling, installation and connection to other equipment. The list here describes common causes of failure found on boards returned to Diamond Systems for repair. This information is provided as a source of advice to help you prevent damaging your Diamond (or any vendor's) embedded computer boards.

ESD damage – This type of damage is usually almost impossible to detect, because there is no visual sign of failure or damage. The symptom is that the board eventually simply stops working, because some component becomes defective. Usually the failure can be identified and the chip can be replaced. To prevent ESD damage, always follow proper ESD-prevention practices when handling computer boards.

Damage during handling or storage – On some boards we have noticed physical damage from mishandling. A common observation is that a screwdriver slipped while installing the board, causing a gouge in the PCB surface and cutting signal traces or damaging components.

Another common observation is damaged board corners, indicating the board was dropped. This may or may not cause damage to the circuitry, depending on what is near the corner. Most of our boards are designed with at least 25 mils clearance between the board edge and any component pad, and ground / power planes are at least 20 mils from the edge to avoid possible shorting from this type of damage. However these design rules are not sufficient to prevent damage in all situations.

A third cause of failure is when a metal screwdriver tip slips, or a screw drops onto the board while it is powered on, causing a short between a power pin and a signal pin on a component. This can cause overvoltage / power supply problems described below. To avoid this type of failure, only perform assembly operations when the system is powered off.

Sometimes boards are stored in racks with slots that grip the edge of the board. This is a common practice for board manufacturers. However our boards are generally very dense, and if the board has components very close to the board edge, they can be damaged or even knocked off the board when the board tilts back in the rack. Diamond recommends that all our boards be stored only in individual ESD-safe packaging. If multiple boards are stored together, they should be contained in bins with dividers between boards. Do not pile boards on top of each other or cram too many boards into a small location. This can cause damage to connector pins or fragile components.

Power supply wired backwards – Our power supplies and boards are not designed to withstand a reverse power supply connection. This will destroy each IC that is connected to the power supply (i.e. almost all ICs). In this case the board will most likely will be unrepairable and must be replaced. A chip destroyed by reverse power or by excessive power will often have a visible hole on the top or show some deformation on the top surface due to vaporization inside the package. **Check twice before applying power!**

Overvoltage on digital I/O line – If a digital I/O signal is connected to a voltage above the maximum specified voltage, the digital circuitry can be damaged. On most of our boards the acceptable range of voltages connected to digital I/O signals is 0-5V, and they can withstand about 0.5V beyond that (-0.5 to 5.5V) before being damaged. However logic signals at 12V and even 24V are common, and if one of these is connected to a 5V logic chip, the chip will be damaged, and the damage could even extend past that chip to others in the circuit

2. INTRODUCTION

2.1 Description

The Janus-MM-4LP-XT family of I/O modules offers two or four opto-isolated CANbus 2.0B ports plus 16 digital I/O lines. Models are available in both the PC/104-*Plus* and PC/104 form factors. Janus-MM-4LP is based on Xilinx Artix-7 FPGA. This core houses the CAN controller logic and digital I/O logic providing data rates up to 1Mbps. Each CAN port supports standard and extended frames as well as expanded TX and RX message queues for enhanced performance. Each port has its own combination isolator and transceiver chip. The 16 digital I/O lines have a selectable voltage level of +3.3V or +5V.

2.2 Features

- 2 or 4 CAN 2.0B compatible ports
- Data rates up to 1Mbps
- Supports standard 11-bit identifier and extended 29-bit identifier frames
- Extended TX and RX message queues for enhanced performance
- 16 8-byte transmit message queues
- 31 8-byte receive message queues
- 16 receive filters
- Galvanically isolated transceivers
- 500V port-to-host and port-to-port isolation
- Jumper selectable biased split termination for improved noise reduction
- 16 digital I/O lines
- Latching I/O connectors for increased ruggedness
- PCI and ISA bus interfaces

2.3 Operating System Support

Windows Embedded 7 and Linux Ubuntu 12.04LTS

Basic CAN driver included with APIs and monitor program

2.4 Mechanical, Electrical, Environmental

- PC/104-*Plus* form factor compliant, 3.55” x 3.775” (90mm x 96mm) without wings
- -40°C to +85°C ambient operating temperature
- Power input requirements: +5VDC +/- 5%
- PCI (3.3V) and ISA (5V) host interfaces
- MIL-STD-202G shock and vibration compatible

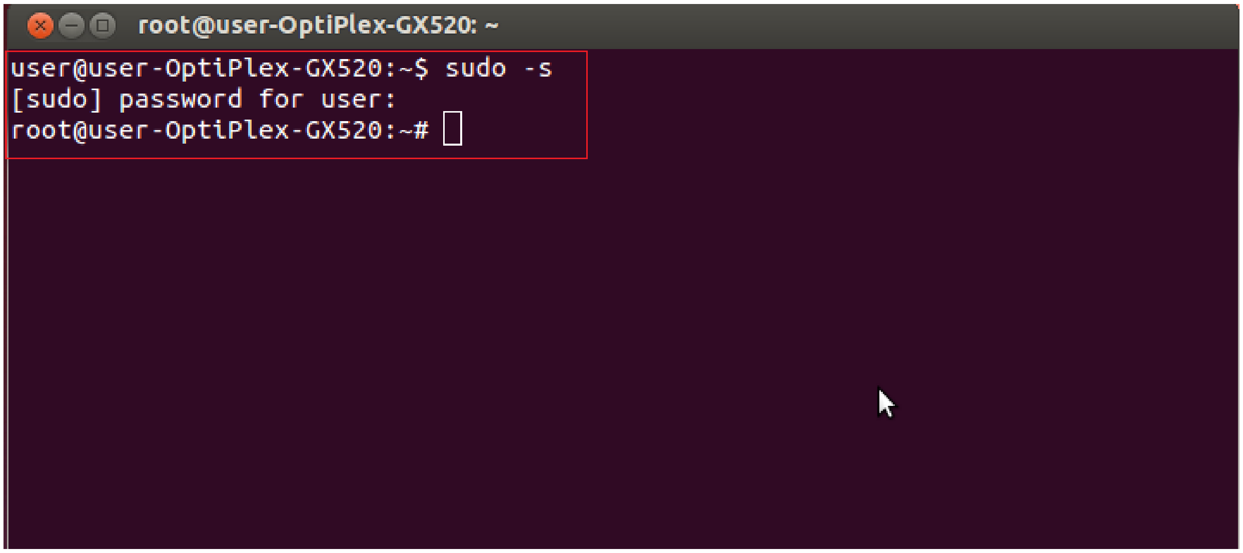
3. OPERATING SYSTEM

Linux Ubuntu-12.04

4. PCI Shared Library installation

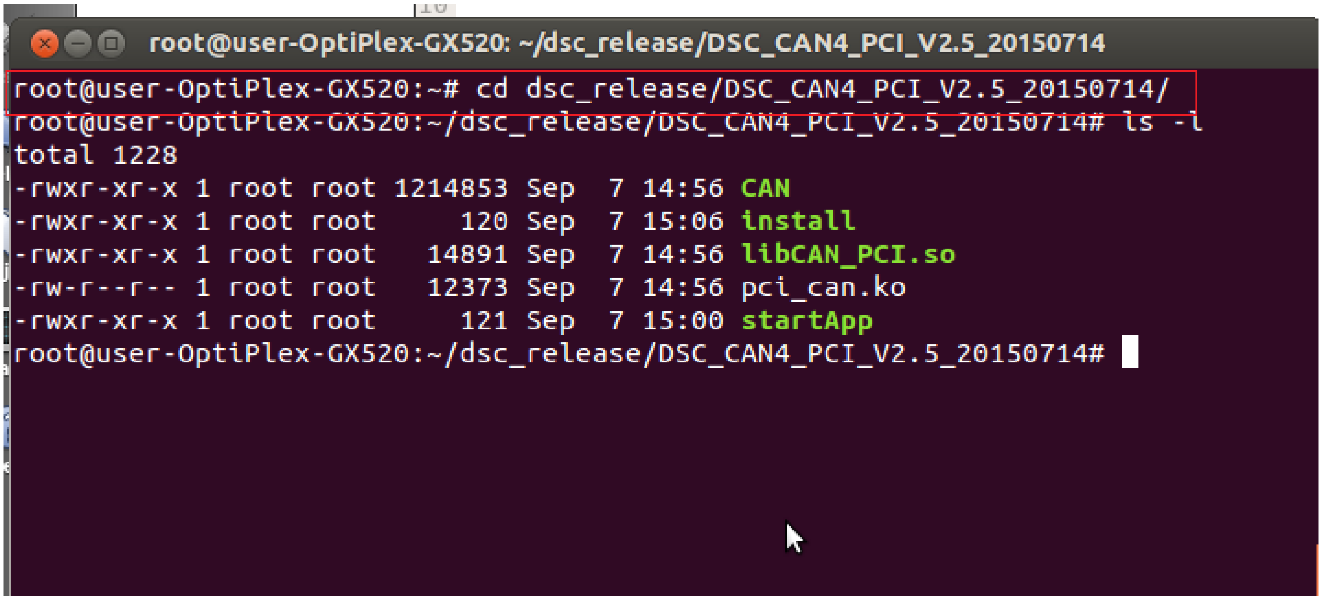
Note : The shared library installation only needs to be set up once.

Step-1: Login as the “root” user by using “sudo –s” command as shown in the below screen,
$ sudo –s
[sudo[password for user :



```
user@user-OptiPlex-GX520:~$ sudo -s
[sudo] password for user:
root@user-OptiPlex-GX520:~#
```

Step-2 : Change the working directory to the release directory where the release content is copied as shown in the below screen.
cd dsc_release/DSC_CAN4_PCI_V2.5_20150714

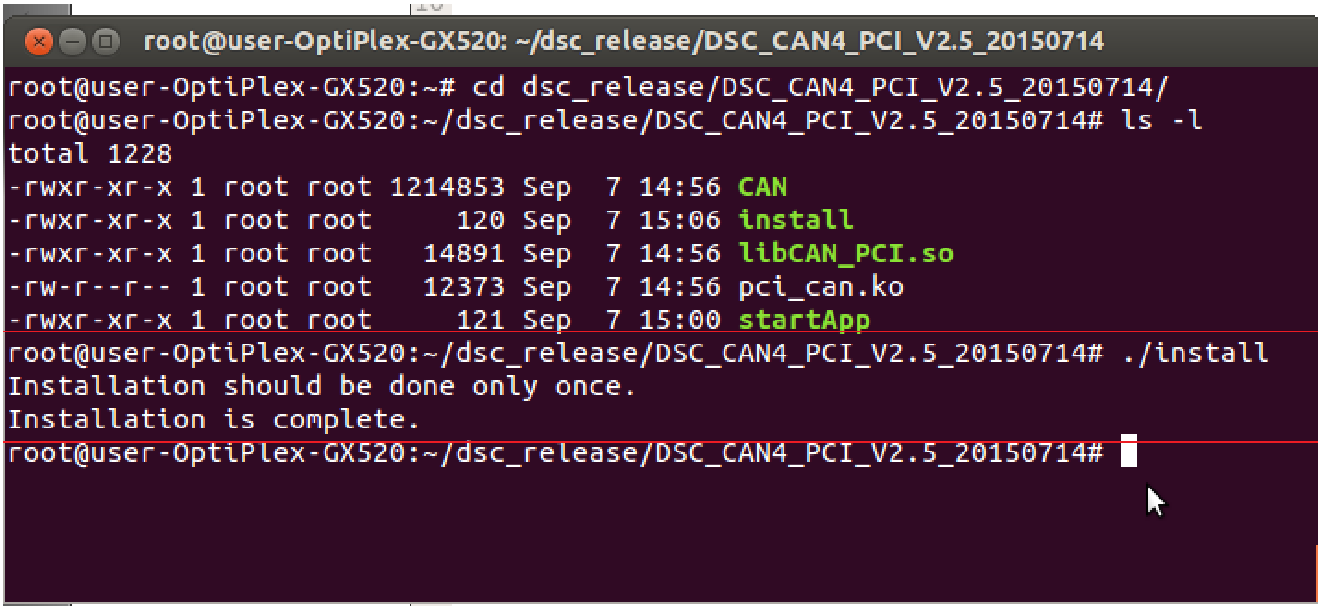

root@user-OptiPlex-GX520: ~/dsc_release/DSC_CAN4_PCI_V2.5_20150714

```
root@user-OptiPlex-GX520:~# cd dsc_release/DSC_CAN4_PCI_V2.5_20150714/
root@user-OptiPlex-GX520:~/dsc_release/DSC_CAN4_PCI_V2.5_20150714# ls -l
total 1228
-rwxr-xr-x 1 root root 1214853 Sep  7 14:56 CAN
-rwxr-xr-x 1 root root     120 Sep  7 15:06 install
-rwxr-xr-x 1 root root   14891 Sep  7 14:56 libCAN_PCI.so
-rw-r--r-- 1 root root   12373 Sep  7 14:56 pci_can.ko
-rwxr-xr-x 1 root root     121 Sep  7 15:00 startApp
root@user-OptiPlex-GX520:~/dsc_release/DSC_CAN4_PCI_V2.5_20150714#
```

Step-3: Execute the “**install**” script which will install the required shared libraries as shown in the below screen.

./install

After complete, install script it will display the message **“Installation is complete”**



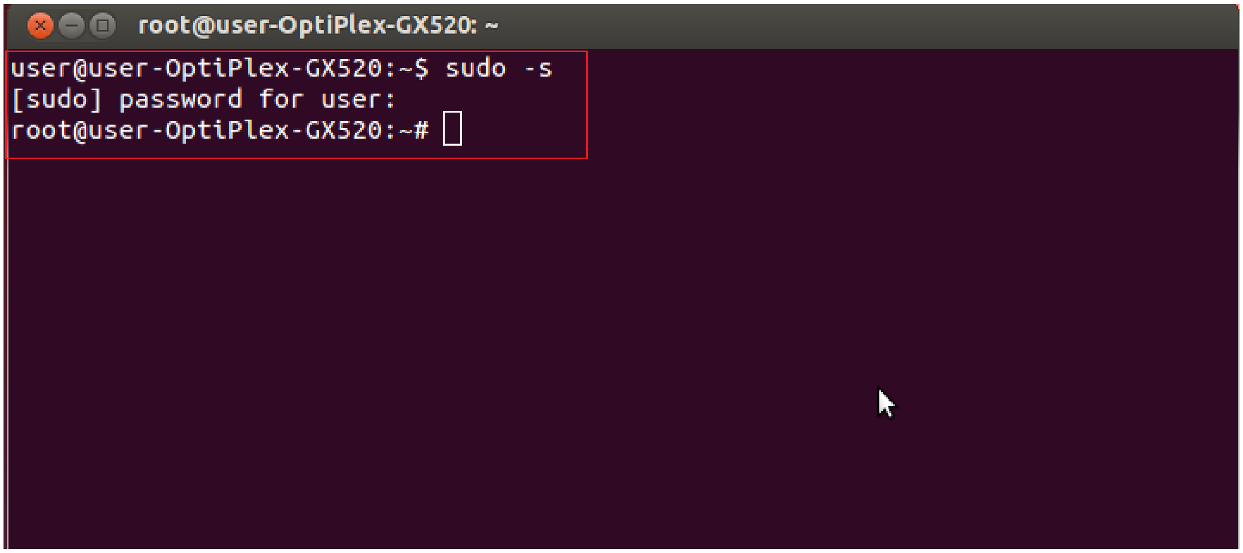
root@user-OptiPlex-GX520: ~/dsc_release/DSC_CAN4_PCI_V2.5_20150714

```
root@user-OptiPlex-GX520:~# cd dsc_release/DSC_CAN4_PCI_V2.5_20150714/
root@user-OptiPlex-GX520:~/dsc_release/DSC_CAN4_PCI_V2.5_20150714# ls -l
total 1228
-rwxr-xr-x 1 root root 1214853 Sep  7 14:56 CAN
-rwxr-xr-x 1 root root     120 Sep  7 15:06 install
-rwxr-xr-x 1 root root   14891 Sep  7 14:56 libCAN_PCI.so
-rw-r--r-- 1 root root   12373 Sep  7 14:56 pci_can.ko
-rwxr-xr-x 1 root root     121 Sep  7 15:00 startApp
root@user-OptiPlex-GX520:~/dsc_release/DSC_CAN4_PCI_V2.5_20150714# ./install
Installation should be done only once.
Installation is complete.
root@user-OptiPlex-GX520:~/dsc_release/DSC_CAN4_PCI_V2.5_20150714# 
```

5. Starting The Demo Application

Starting the demo application includes loading the driver and starting the application. It must be done with “root” privileges.

Step-1: Login as the “root” user by using “sudo –s” command as shown in the below screen,
$ sudo –s
[sudo[password for user :

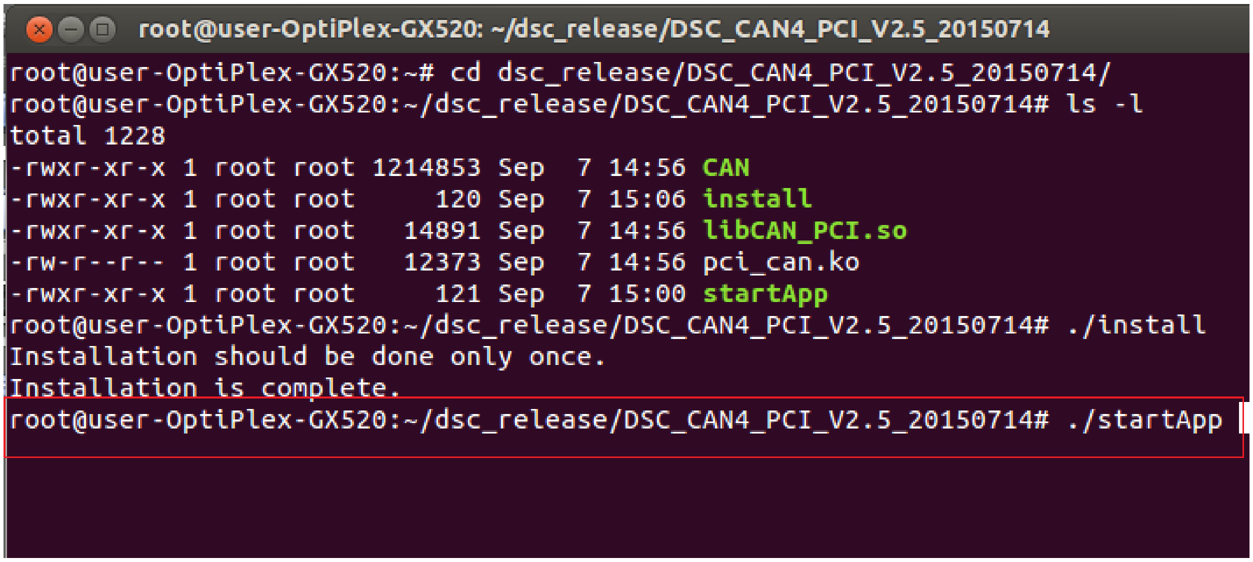


root@user-OptiPlex-GX520: ~

```
user@user-OptiPlex-GX520:~$ sudo -s
[sudo] password for user:
root@user-OptiPlex-GX520:~#
```

Step-2: Start the application by executing the “startApp” script as shown in the below screen.

./startApp



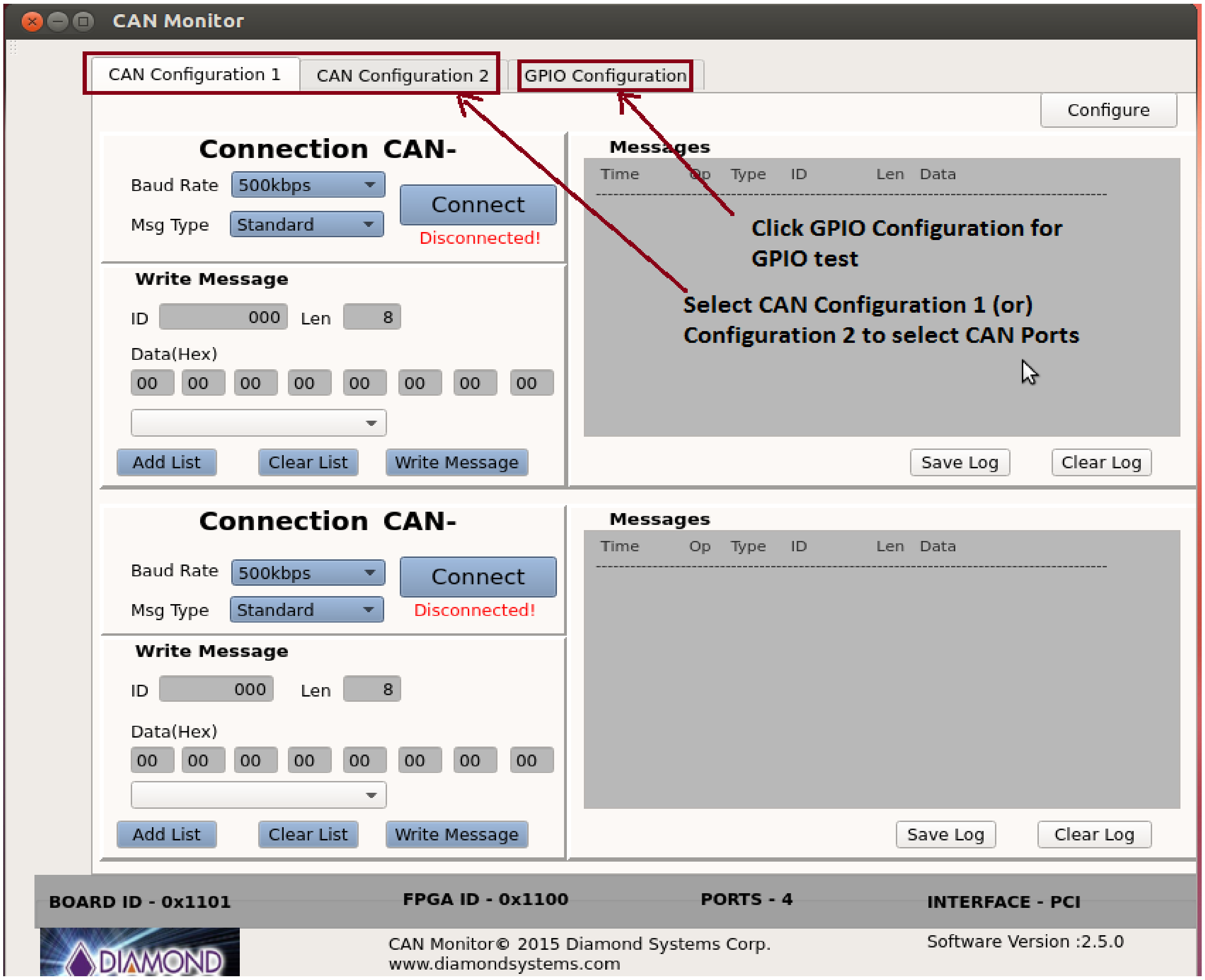
root@user-OptiPlex-GX520: ~/dsc_release/DSC_CAN4_PCI_V2.5_20150714

```
root@user-OptiPlex-GX520:~# cd dsc_release/DSC_CAN4_PCI_V2.5_20150714/
root@user-OptiPlex-GX520:~/dsc_release/DSC_CAN4_PCI_V2.5_20150714# ls -l
total 1228
-rwxr-xr-x 1 root root 1214853 Sep  7 14:56 CAN
-rwxr-xr-x 1 root root     120 Sep  7 15:06 install
-rwxr-xr-x 1 root root   14891 Sep  7 14:56 libCAN_PCI.so
-rw-r--r-- 1 root root   12373 Sep  7 14:56 pci_can.ko
-rwxr-xr-x 1 root root     121 Sep  7 15:00 startApp
root@user-OptiPlex-GX520:~/dsc_release/DSC_CAN4_PCI_V2.5_20150714# ./install
Installation should be done only once.
Installation is complete.
root@user-OptiPlex-GX520:~/dsc_release/DSC_CAN4_PCI_V2.5_20150714# ./startApp
```

The above step will start the CAN demo application.

6. Configuring CAN ports

The CAN Monitor demo application contains three tabs. Click on CAN Configuration 1 or 2 tab to configure the CAN ports, or click on the GPIO Configuration tab to configure the GPIO lines. In first two tabs, any two CAN ports can be configured using the “Configure” button provided on each tab.



Baud Rate: 500kbps
Msg Type: Standard
Disconnected!
ID: 000 Len: 8
Data(Hex): 00 00 00 00 00 00 00 00

Click GPIO Configuration for GPIO test

Select CAN Configuration 1 (or) Configuration 2 to select CAN Ports

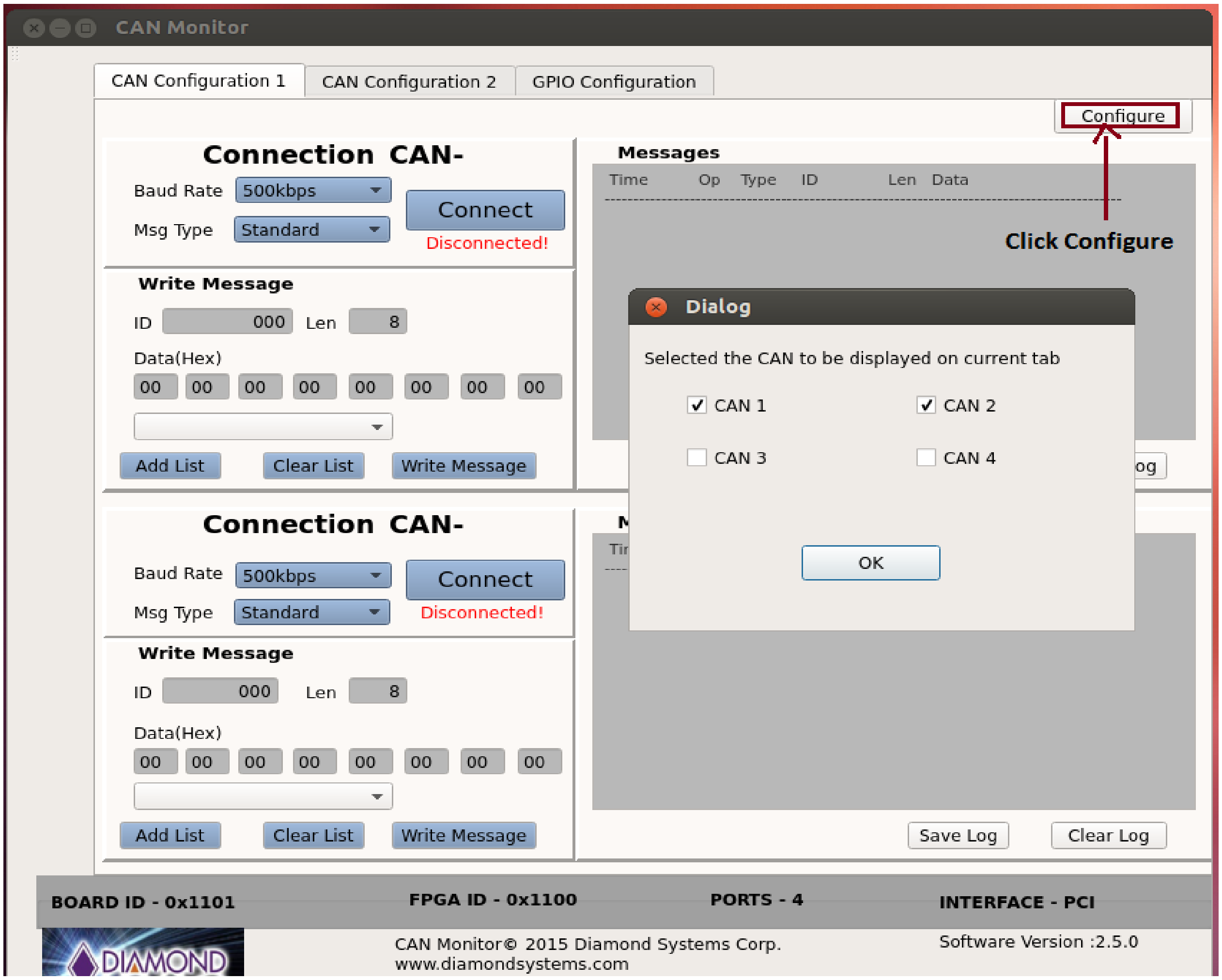
Baud Rate: 500kbps
Msg Type: Standard
Disconnected!
ID: 000 Len: 8
Data(Hex): 00 00 00 00 00 00 00 00

BOARD ID - 0x1101   FPGA ID - 0x1100   PORTS - 4   INTERFACE - PCI

CAN Monitor© 2015 Diamond Systems Corp.
www.diamondsystems.com

Software Version :2.5.0

A Pop-up window will appear by clicking on the **“Configure”** button when selecting any two CAN ports for each tab. After selecting, click on OK.



Click Configure

Connection CAN-

Baud Rate 500kbps

Msg Type Standard

Disconnected!

Write Message

ID 000 Len 8

Data(Hex)

00 00 00 00 00 00 00 00

Messages

```
Time      Op   Type   ID        Len  Data
--------------------------------------------------------------
```

Dialog

Selected the CAN to be displayed on current tab

☑ CAN 1 ☑ CAN 2

☐ CAN 3 ☐ CAN 4

Connection CAN-

Baud Rate 500kbps

Msg Type Standard

Disconnected!

Write Message

ID 000 Len 8

Data(Hex)

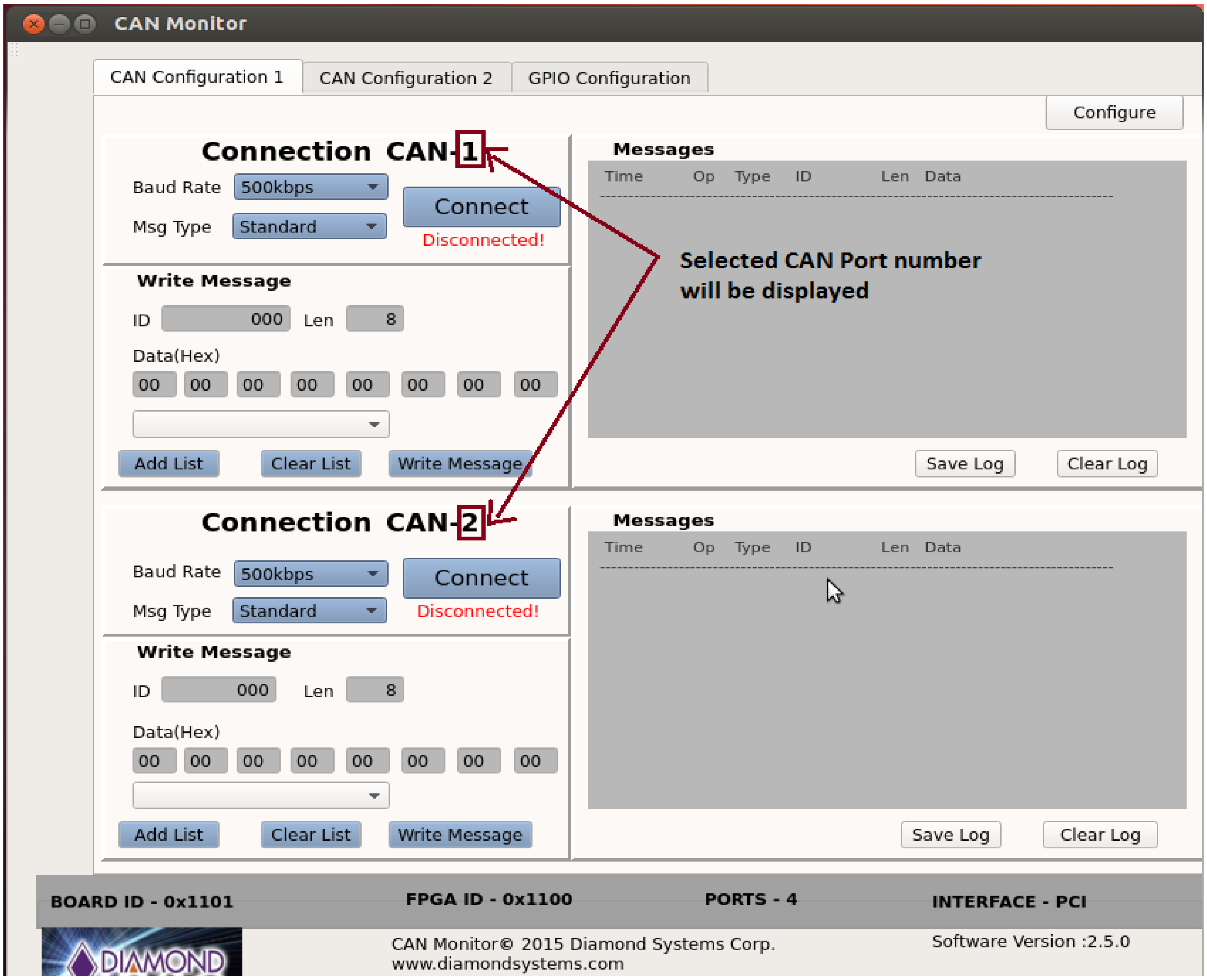
00 00 00 00 00 00 00 00

BOARD ID - 0x1101 FPGA ID - 0x1100 PORTS - 4 INTERFACE - PCI

CAN Monitor© 2015 Diamond Systems Corp.
www.diamondsystems.com

Software Version :2.5.0

Initially CAN numbers will be empty on each tab. After configuring the CAN ports, the selected CAN # number will be displayed.



Selected CAN Port number will be displayed

Connection CAN-1

Baud Rate 500kbps

Msg Type Standard

Disconnected!

Write Message

ID 000 Len 8

Data(Hex) 00 00 00 00 00 00 00 00

Connection CAN-2

Baud Rate 500kbps

Msg Type Standard

Disconnected!

Write Message

ID 000 Len 8

Data(Hex) 00 00 00 00 00 00 00 00

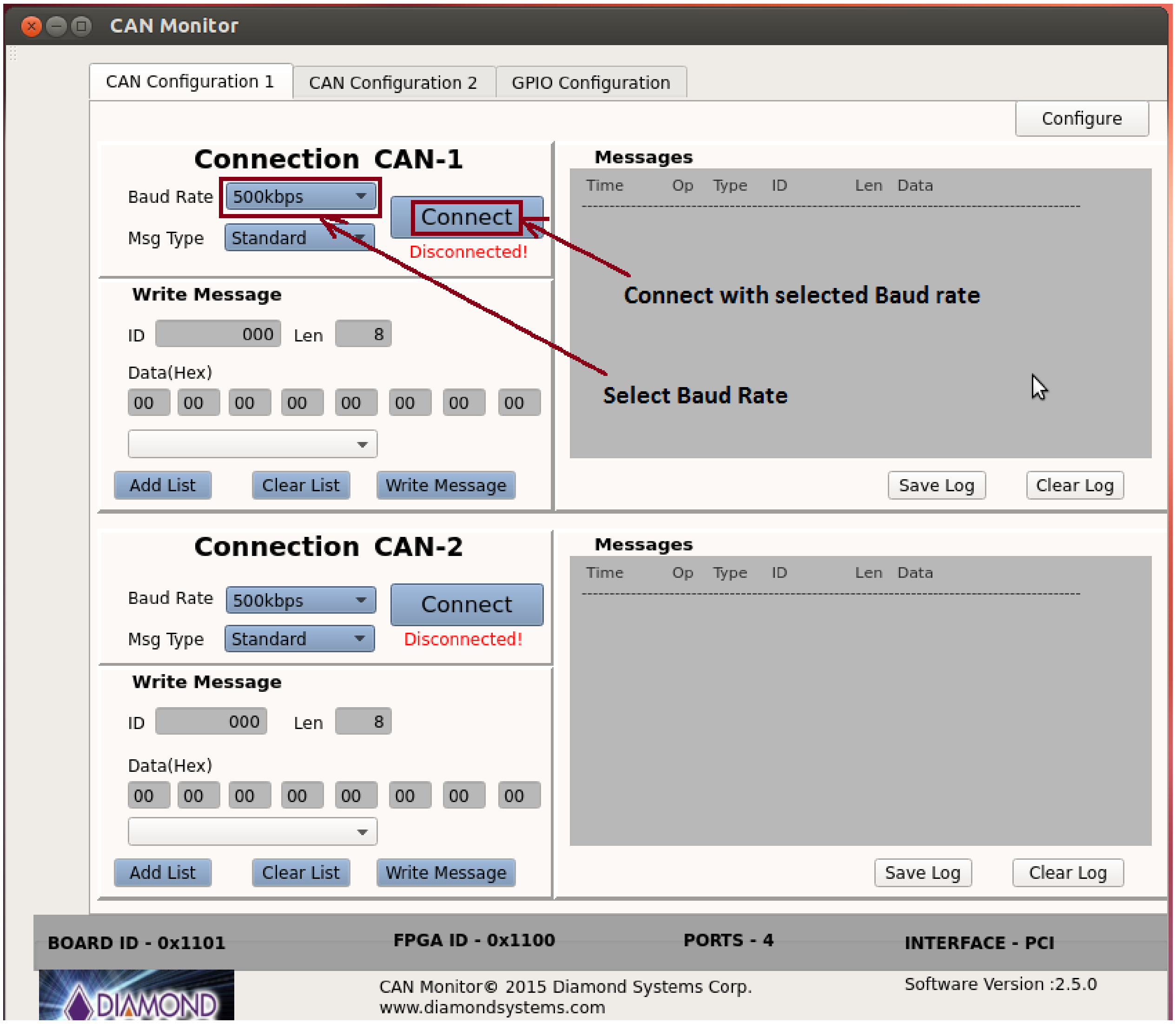
BOARD ID - 0x1101 FPGA ID - 0x1100 PORTS - 4 INTERFACE - PCI

CAN Monitor© 2015 Diamond Systems Corp.
www.diamondsystems.com

Software Version :2.5.0

7. Setting CAN Baud Rate

The baud rate for each port can be configured using the drop-down menu for the particular CAN port. After selecting the desired baud rate, pressing “**Connect**” will configure the specified baud rate for that particular port.



Connection CAN-1

Baud Rate 500kbps

Msg Type Standard

Disconnected!

Write Message

ID 000 Len 8

Data(Hex)

00 00 00 00 00 00 00 00

Messages

Time Op Type ID Len Data

Connect with selected Baud rate

Select Baud Rate

Connection CAN-2

Baud Rate 500kbps

Msg Type Standard

Disconnected!

Write Message

ID 000 Len 8

Data(Hex)

00 00 00 00 00 00 00 00

Messages

Time Op Type ID Len Data

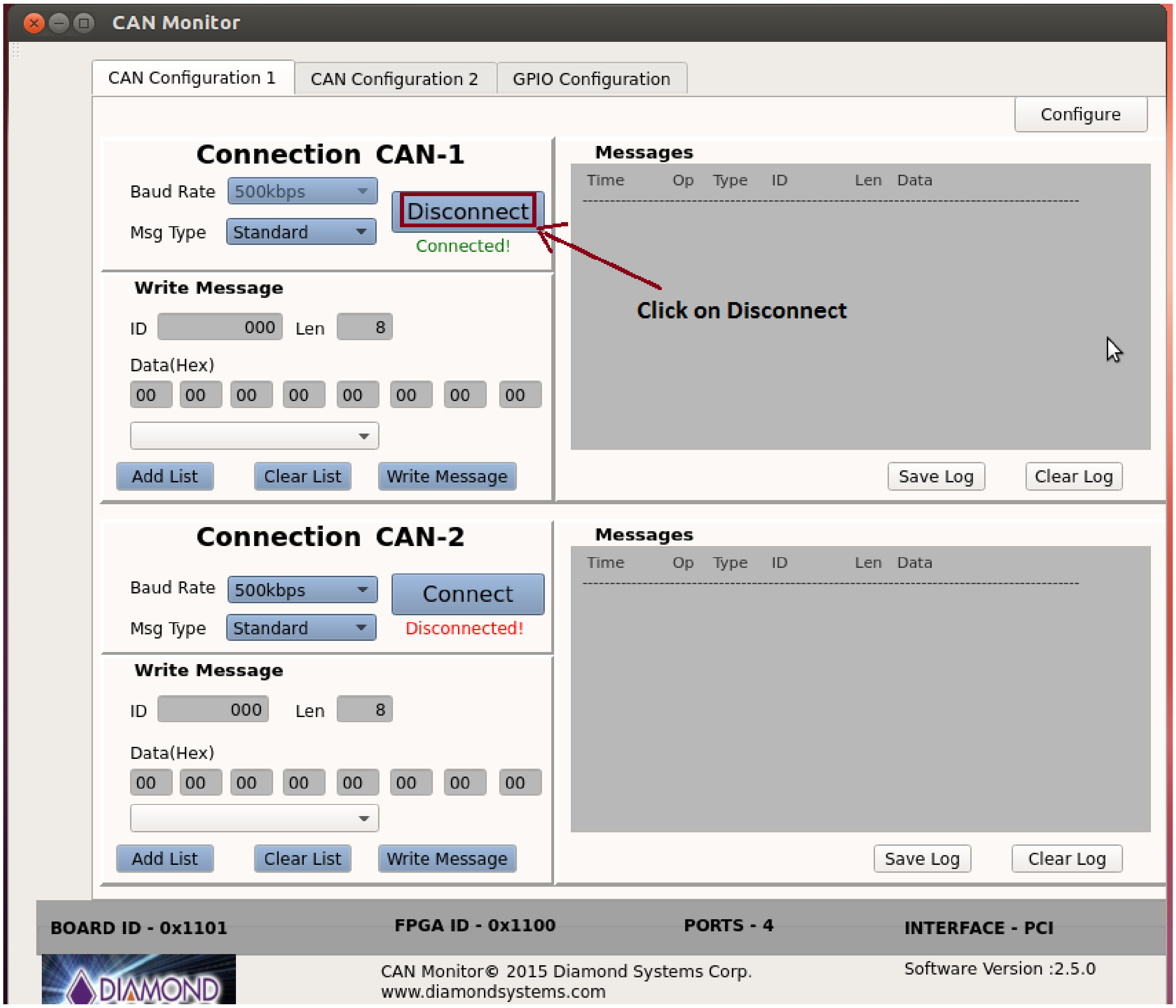
BOARD ID - 0x1101 FPGA ID - 0x1100 PORTS - 4 INTERFACE - PCI

CAN Monitor© 2015 Diamond Systems Corp.
www.diamondsystems.com

Software Version :2.5.0

8. Changing CAN Baud Rate

To change the baud rate, click on "**Disconnect**" and select a new desired baud rate. Click on the "Connect" button.



CAN Monitor

Connection CAN-1

Baud Rate 500kbps

Msg Type Standard

Connected!

Write Message

ID 000 Len 8

Data(Hex) 00 00 00 00 00 00 00 00

Connection CAN-2

Baud Rate 500kbps

Msg Type Standard

Disconnected!

Write Message

ID 000 Len 8

Data(Hex) 00 00 00 00 00 00 00 00

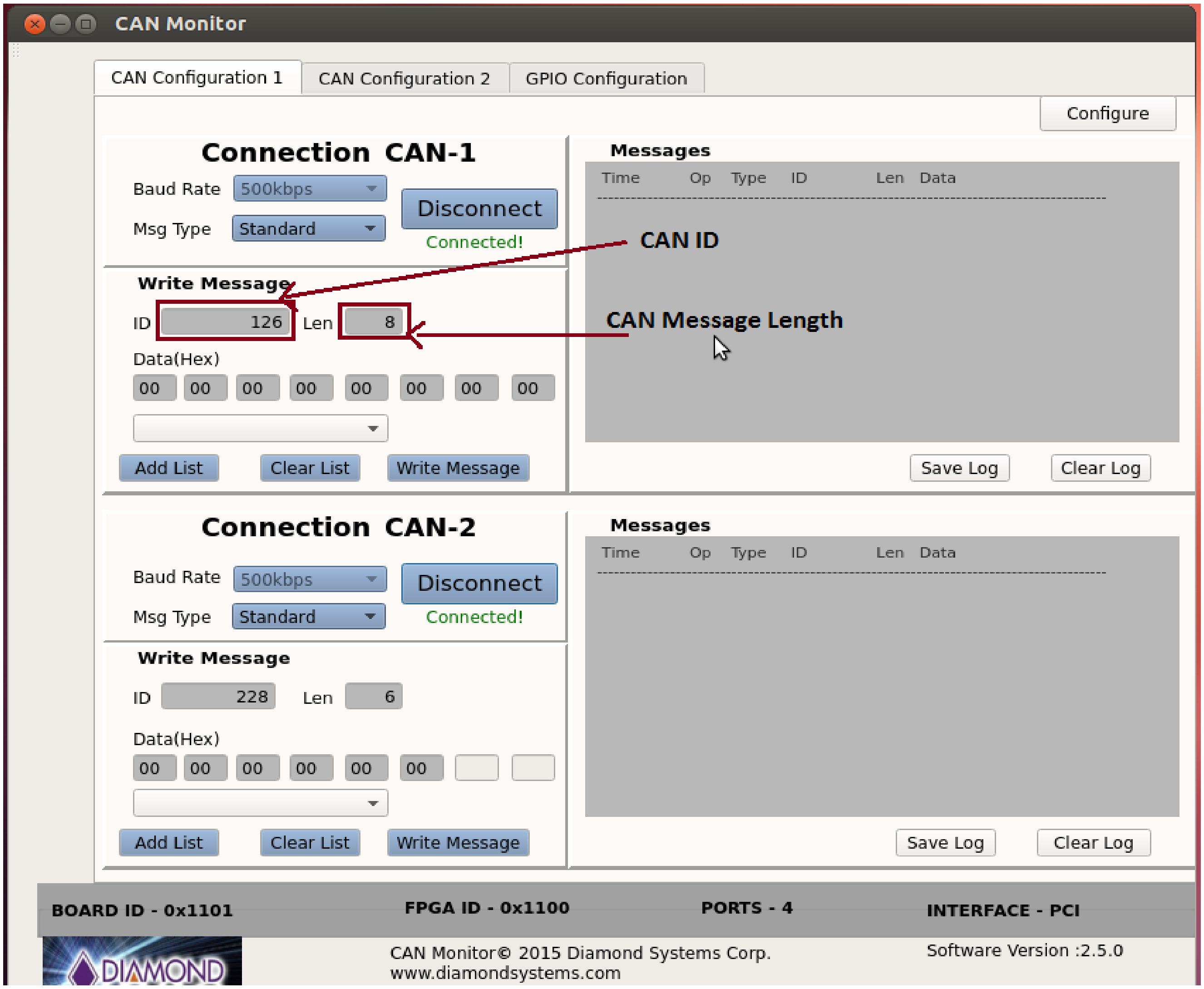
BOARD ID - 0x1101 FPGA ID - 0x1100 PORTS - 4 INTERFACE - PCI

CAN Monitor© 2015 Diamond Systems Corp.
www.diamondsystems.com

Software Version :2.5.0

9. Setting CAN ID and Message Length

The CAN ID and message length can be configured by entering the desired values into the respective fields for that particular port.



CAN ID

CAN Message Length

Connection CAN-1

Baud Rate: 500kbps

Msg Type: Standard

Connected!

Write Message

ID: 126 Len: 8

Data(Hex): 00 00 00 00 00 00 00 00

Messages

Time Op Type ID Len Data

Connection CAN-2

Baud Rate: 500kbps

Msg Type: Standard

Connected!

Write Message

ID: 228 Len: 6

Data(Hex): 00 00 00 00 00 00

Messages

Time Op Type ID Len Data

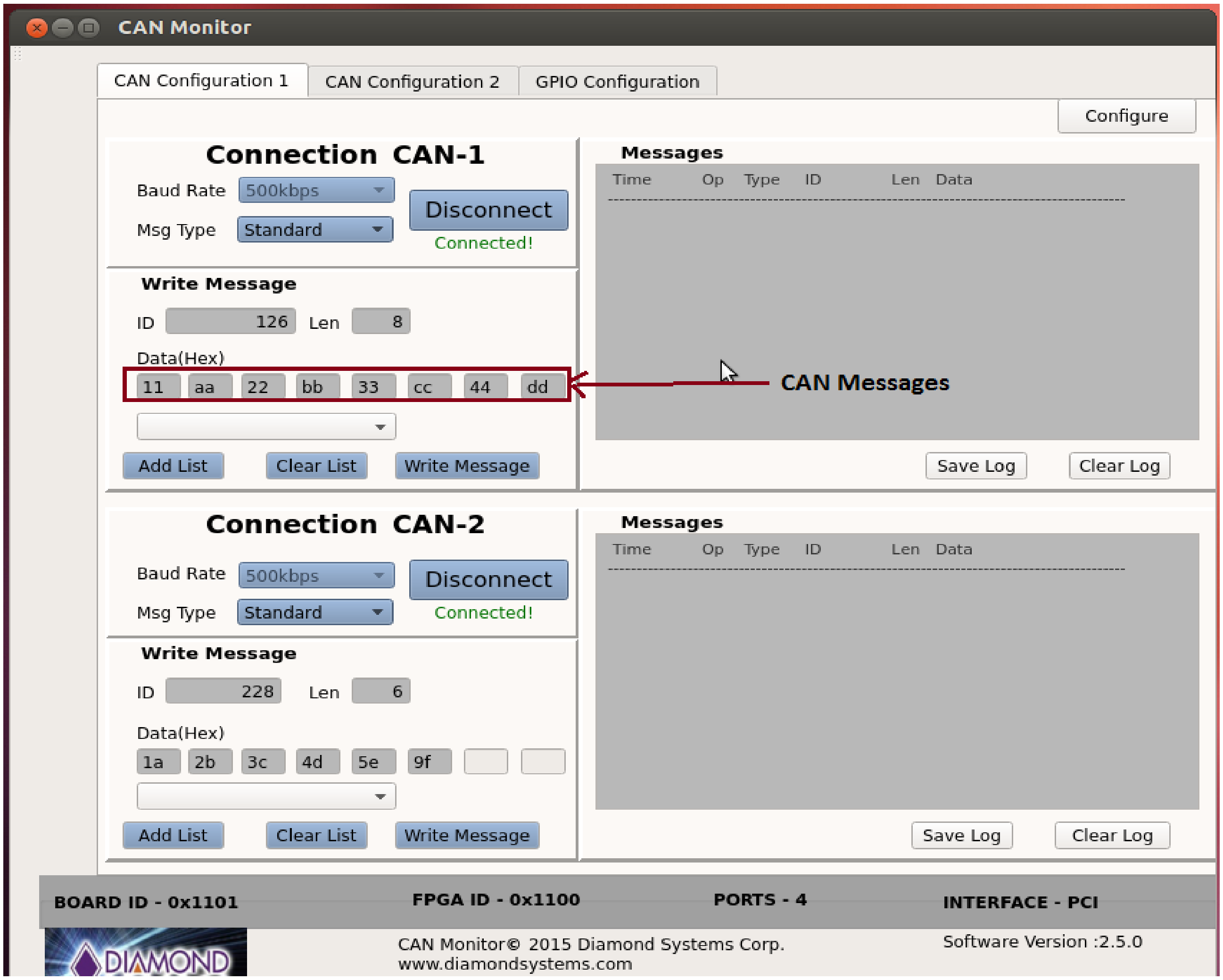
BOARD ID - 0x1101 FPGA ID - 0x1100 PORTS - 4 INTERFACE - PCI

CAN Monitor© 2015 Diamond Systems Corp.
www.diamondsystems.com

Software Version :2.5.0

10. Writing CAN Message

The CAN message data can be set by entering the desired CAN data into the Data (Hex) fields.



CAN Monitor

Connection CAN-1

Baud Rate: 500kbps

Msg Type: Standard

Connected!

Write Message

ID: 126 Len: 8

Data(Hex): 11 aa 22 bb 33 cc 44 dd

Messages

```
Time        Op   Type   ID          Len   Data
--------------------------------------------------------------------------------
```

CAN Messages

Connection CAN-2

Baud Rate: 500kbps

Msg Type: Standard

Connected!

Write Message

ID: 228 Len: 6

Data(Hex): 1a 2b 3c 4d 5e 9f

Messages

```
Time        Op   Type   ID          Len   Data
--------------------------------------------------------------------------------
```

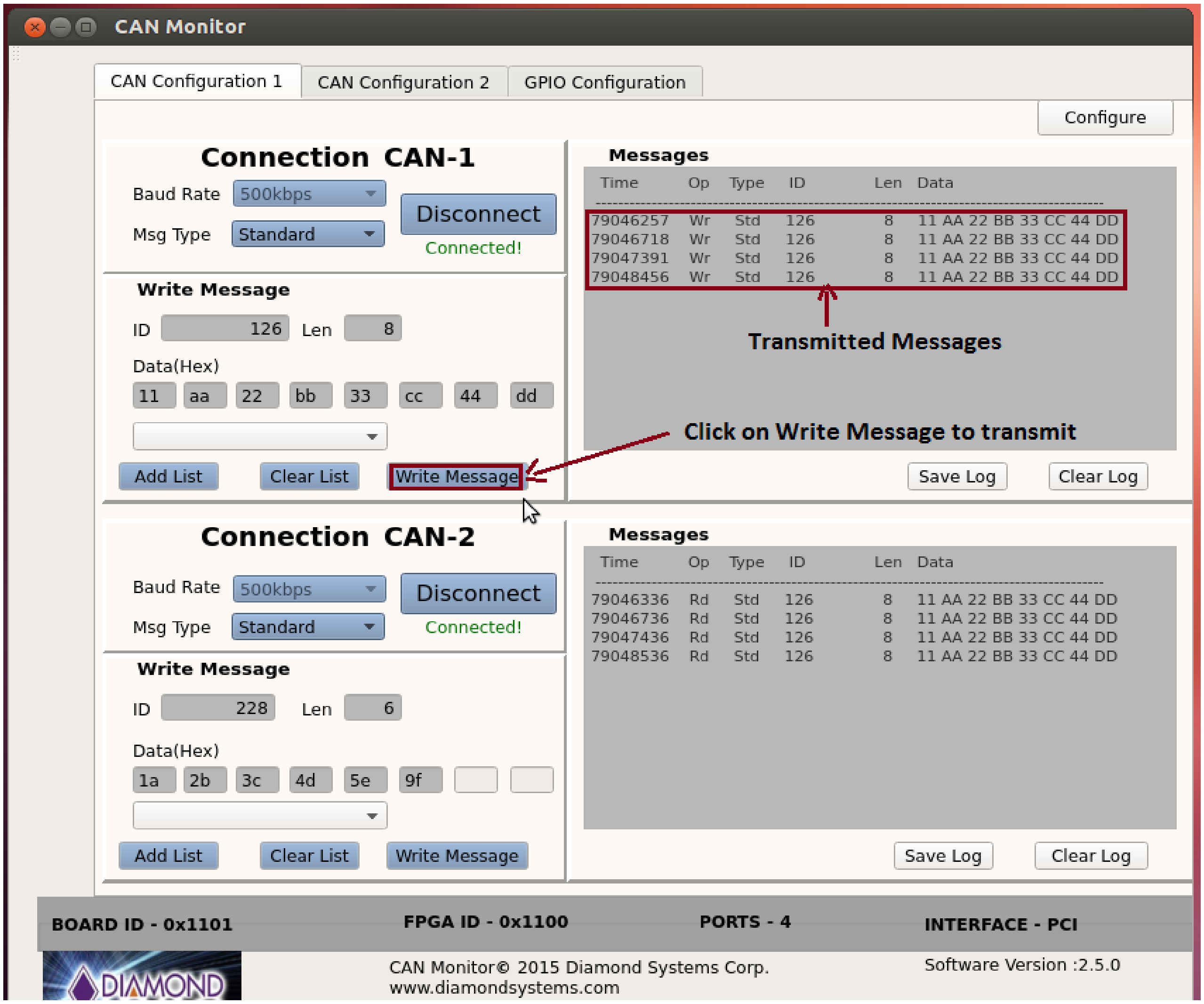
BOARD ID - 0x1101 FPGA ID - 0x1100 PORTS - 4 INTERFACE - PCI

CAN Monitor© 2015 Diamond Systems Corp.
www.diamondsystems.com

Software Version :2.5.0

11. Transmitting CAN Message

The configured CAN-ID, Len, and CAN message data can be transmitted using the **“Write Message”** button. Transmitted messages will be displayed in the CAN message box for that particular CAN port.



Connection CAN-1

Baud Rate: 500kbps
Msg Type: Standard
Connected!

Write Message

ID: 126 Len: 8

Data(Hex): 11 aa 22 bb 33 cc 44 dd

Messages

```
Time       Op   Type   ID       Len   Data
-----------------------------------------------------------------------------------
79046257   Wr   Std    126        8   11 AA 22 BB 33 CC 44 DD
79046718   Wr   Std    126        8   11 AA 22 BB 33 CC 44 DD
79047391   Wr   Std    126        8   11 AA 22 BB 33 CC 44 DD
79048456   Wr   Std    126        8   11 AA 22 BB 33 CC 44 DD
```

Transmitted Messages

Click on Write Message to transmit

Connection CAN-2

Baud Rate: 500kbps
Msg Type: Standard
Connected!

Write Message

ID: 228 Len: 6

Data(Hex): 1a 2b 3c 4d 5e 9f

Messages

```
Time       Op   Type   ID       Len   Data
-----------------------------------------------------------------------------------
79046336   Rd   Std    126        8   11 AA 22 BB 33 CC 44 DD
79046736   Rd   Std    126        8   11 AA 22 BB 33 CC 44 DD
79047436   Rd   Std    126        8   11 AA 22 BB 33 CC 44 DD
79048536   Rd   Std    126        8   11 AA 22 BB 33 CC 44 DD
```

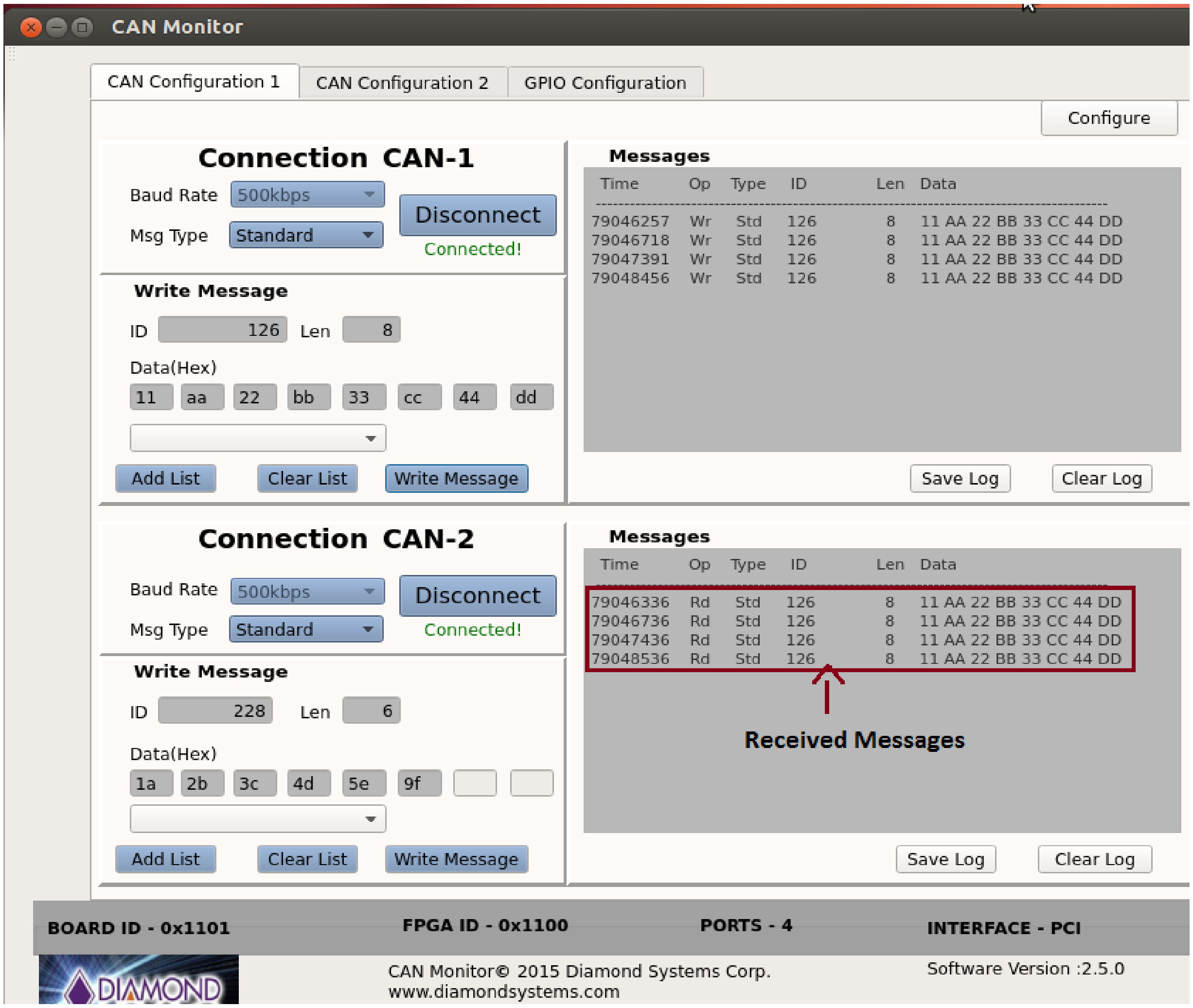
BOARD ID - 0x1101   FPGA ID - 0x1100   PORTS - 4   INTERFACE - PCI

CAN Monitor© 2015 Diamond Systems Corp.
www.diamondsystems.com

Software Version :2.5.0

12. Receive CAN Message

Received CAN message will be displayed in the CAN message box for that particular CAN port.



Connection CAN-1

Baud Rate: 500kbps

Msg Type: Standard

Connected!

Write Message

ID: 126 Len: 8

Data(Hex): 11 aa 22 bb 33 cc 44 dd

Messages

```
Time        Op   Type   ID        Len   Data
------------------------------------------------------------------------
79046257    Wr   Std    126        8    11 AA 22 BB 33 CC 44 DD
79046718    Wr   Std    126        8    11 AA 22 BB 33 CC 44 DD
79047391    Wr   Std    126        8    11 AA 22 BB 33 CC 44 DD
79048456    Wr   Std    126        8    11 AA 22 BB 33 CC 44 DD
```

Connection CAN-2

Baud Rate: 500kbps

Msg Type: Standard

Connected!

Write Message

ID: 228 Len: 6

Data(Hex): 1a 2b 3c 4d 5e 9f

Messages

```
Time        Op   Type   ID        Len   Data
------------------------------------------------------------------------
79046336    Rd   Std    126        8    11 AA 22 BB 33 CC 44 DD
79046736    Rd   Std    126        8    11 AA 22 BB 33 CC 44 DD
79047436    Rd   Std    126        8    11 AA 22 BB 33 CC 44 DD
79048536    Rd   Std    126        8    11 AA 22 BB 33 CC 44 DD
```

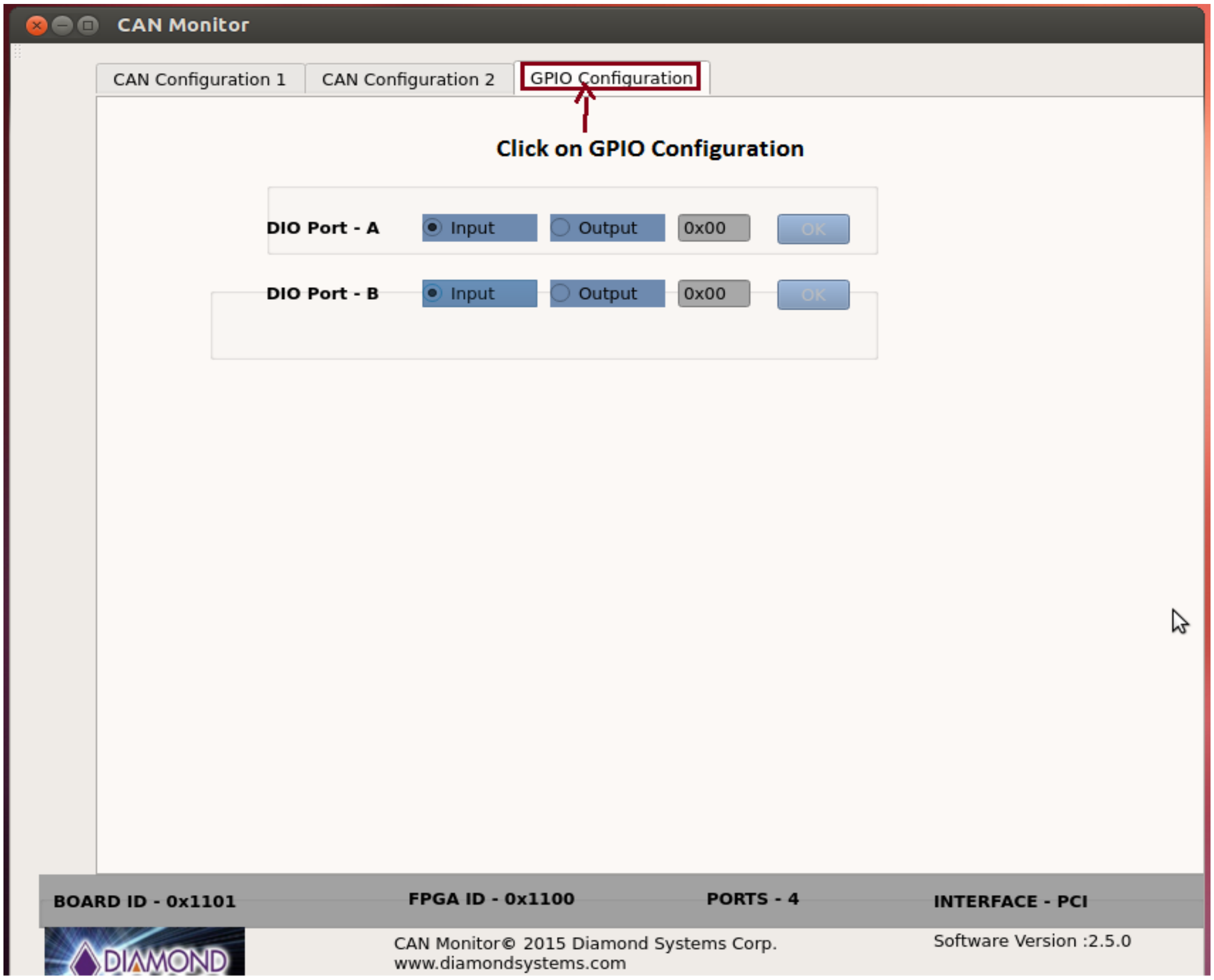
BOARD ID - 0x1101 FPGA ID - 0x1100 PORTS - 4 INTERFACE - PCI

CAN Monitor© 2015 Diamond Systems Corp.
www.diamondsystems.com

Software Version :2.5.0

13. GPIO Test

Click on the GPIO Configuration tab to configure DIO Port-A and Port-B as either input ports or output ports.



DIO Port - A ● Input ○ Output 0x00 OK

DIO Port - B ● Input ○ Output 0x00 OK

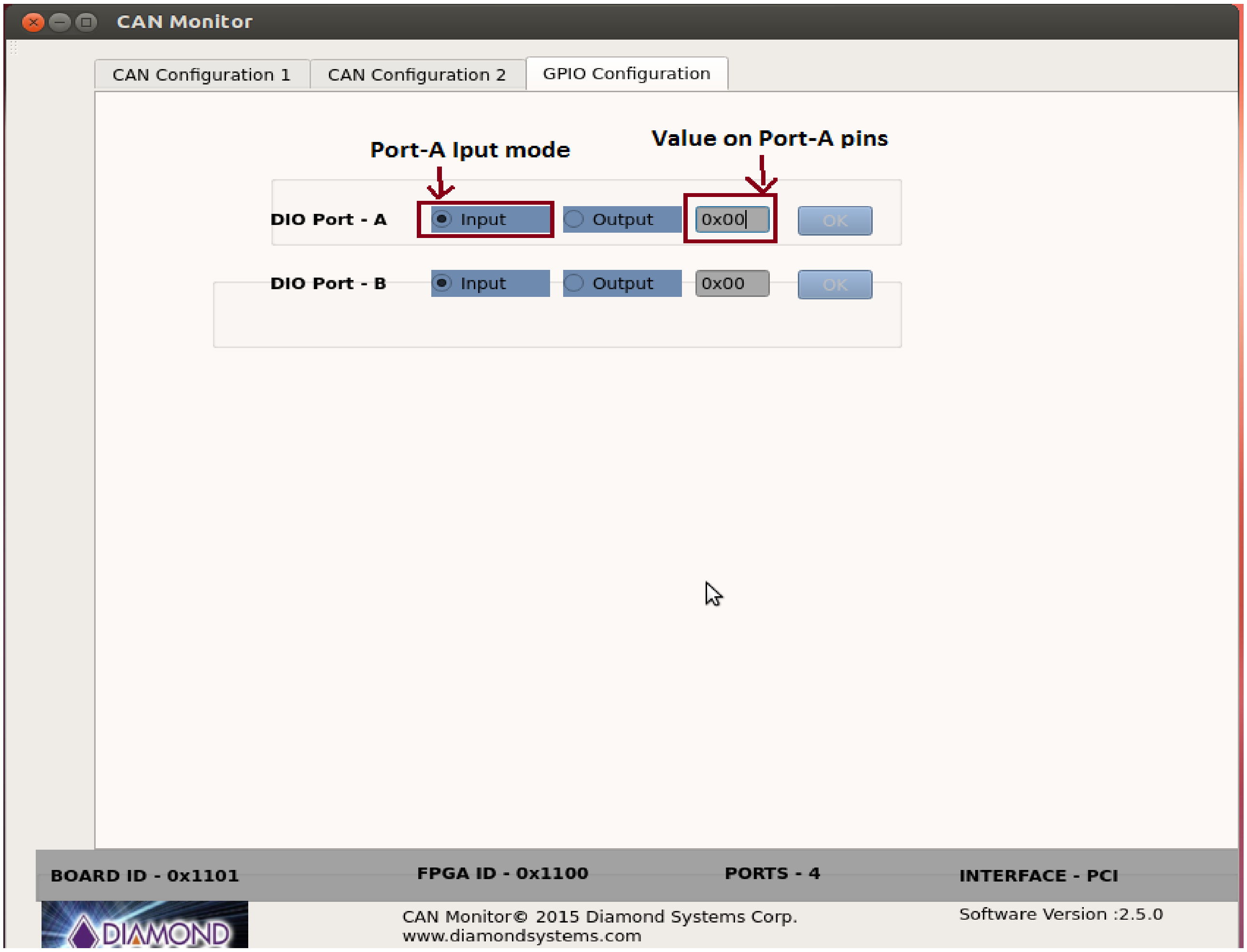
BOARD ID - 0x1101 FPGA ID - 0x1100 PORTS - 4 INTERFACE - PCI

CAN Monitor© 2015 Diamond Systems Corp.
www.diamondsystems.com

Software Version :2.5.0

14. Setting DIO as Input Port

Click on DIO Port-A **Input** button to configure the DIO Port-A as an input port. In this case, Port-A value will be displayed.



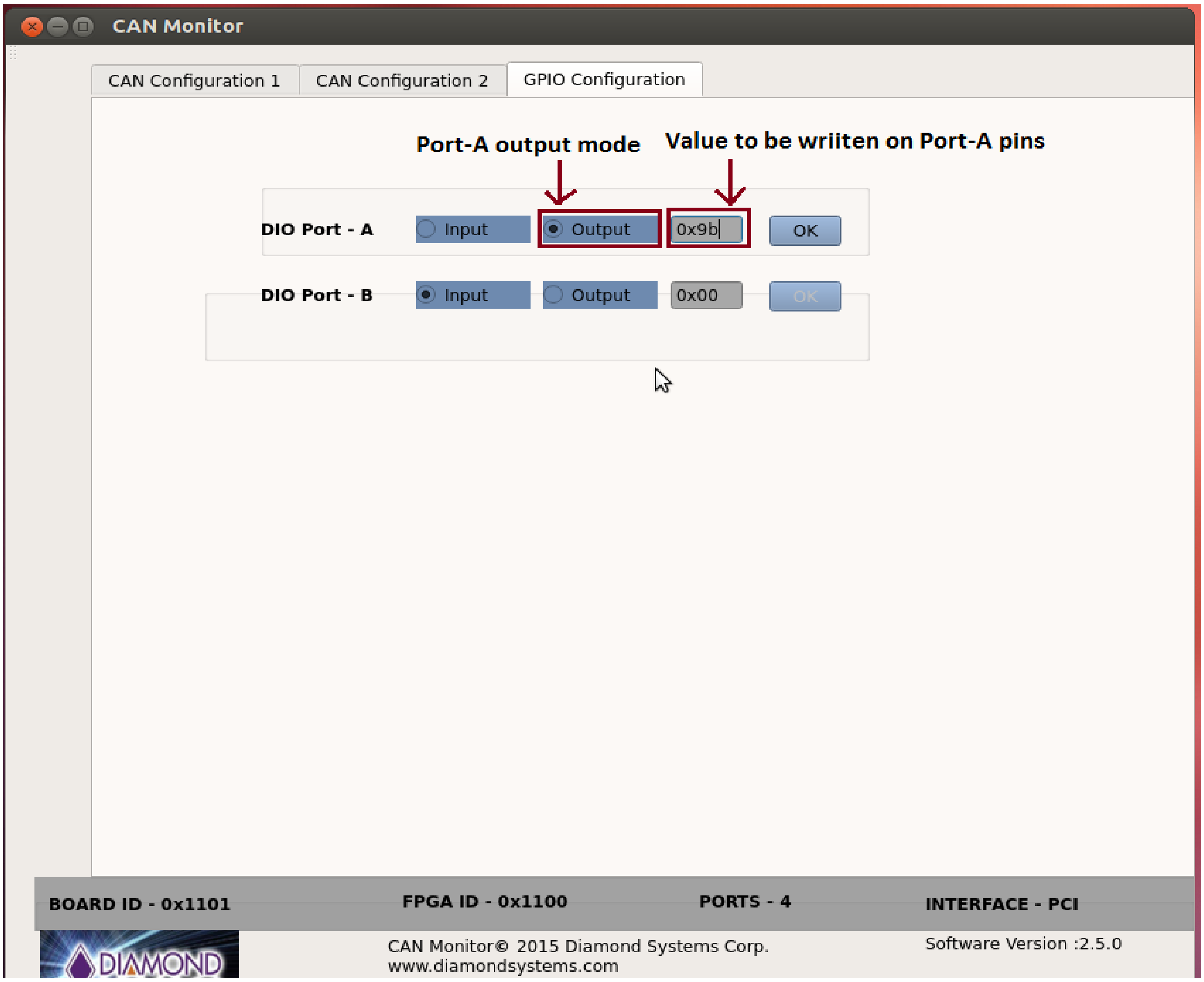
BOARD ID - 0x1101 FPGA ID - 0x1100 PORTS - 4 INTERFACE - PCI

CAN Monitor© 2015 Diamond Systems Corp.
www.diamondsystems.com

Software Version :2.5.0

15. Setting DIO as Output Port

Click on DIO Port-A **Output** button to configure the DIO Port-A as an output port. In this case, enter 8-bit port value and click on the Ok button to write to Port-A.



Port-A output mode

Value to be wriiten on Port-A pins

DIO Port - A ○ Input ● Output 0x9b

DIO Port - B ● Input ○ Output 0x00

BOARD ID - 0x1101 FPGA ID - 0x1100 PORTS - 4 INTERFACE - PCI

CAN Monitor© 2015 Diamond Systems Corp.
www.diamondsystems.com

Software Version :2.5.0

Similarly, Port-B can be configured as either an input or output port.

16. FPGA ID and Board-ID

FPGA ID: Displays FPGA version

E.g. 0X1100:- it's the Artix-7   series FPGA.

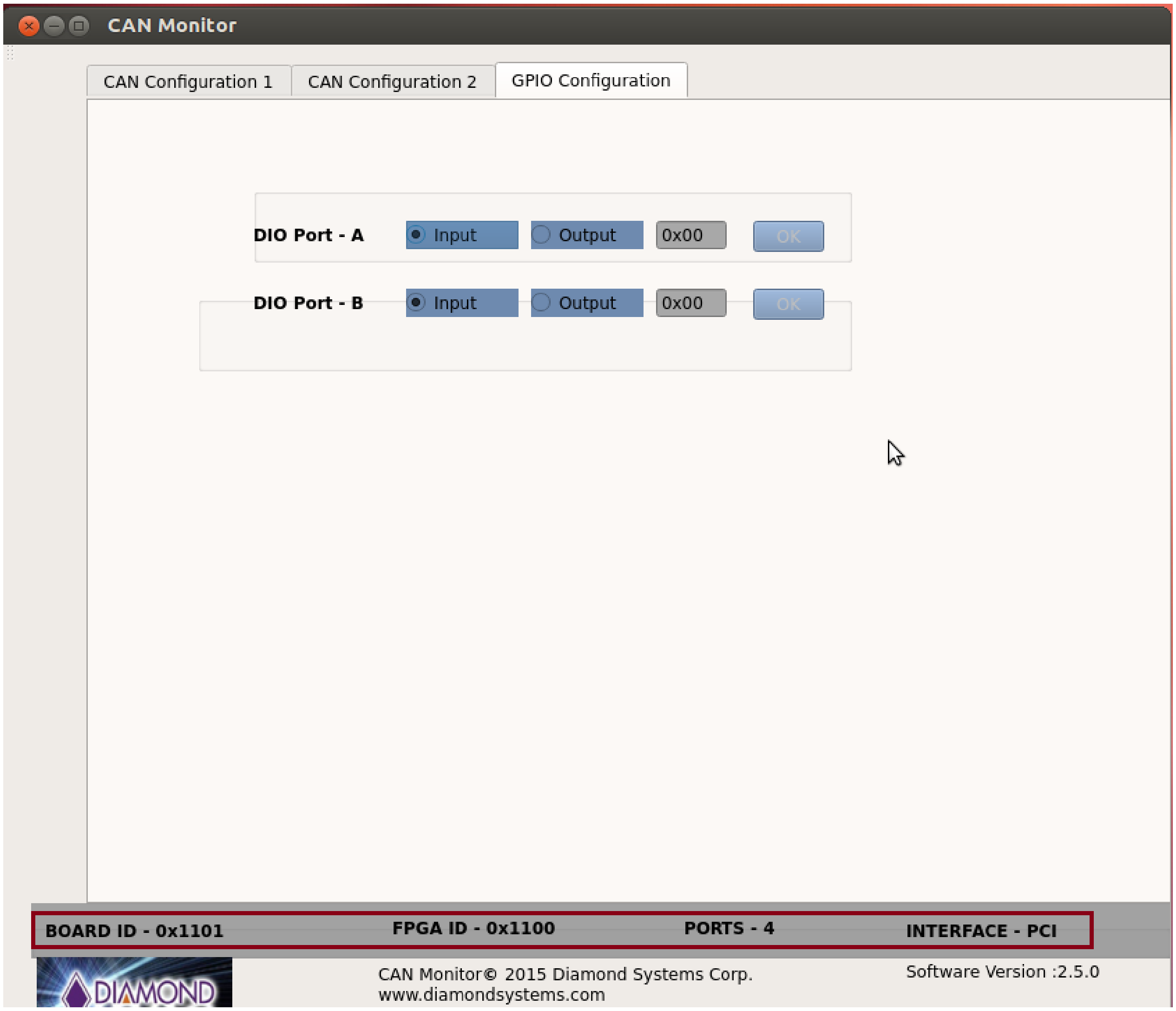
Board ID: Displays board ID

0x1101:- PCI Board.

0X1100:- ISA Board.

Ports: Displays number of CAN ports it can be either 2 or 4 ports.

Interface: Displays interface it can be either ISA or PCI.



CAN Monitor

DIO Port - A ● Input ○ Output 0x00

DIO Port - B ● Input ○ Output 0x00

BOARD ID - 0x1101 FPGA ID - 0x1100 PORTS - 4 INTERFACE - PCI

CAN Monitor© 2015 Diamond Systems Corp.
www.diamondsystems.com

Software Version :2.5.0

17. APIs to Configure and Manage CAN Ports

init_can(int can_ch): This API accepts the CAN interface number as an argument for the initialization. This will initialize CAN#0, CAN#1, CAN#2, and CAN#3 ports.

This function will return CAN file descriptor (fd). The return value of this function should be retained for all subsequent operations. Its prototype is defined in the can.h file. Declaring four CAN file descriptors which will retain its return values.

```
#include “can.h”
// TO initialize CAN#0 channel, pass the argument value as 0 to init_can function as shown below.
//Other declaration
…
can_fd_0 = init_can(0) ; // Will initialize CAN#0 channel
```

Similarly, CAN#n can be initialized by passing ‘n’ as argument to init_can function as shown below

```
can_fd_n = init_can(n) ;

…
```

Where, n varies from 1 to 3.

Baud rate configuration.

set_baudrate() : This function will configure the baud rate for the specified CAN port. By default it will not configure any baud rate.

```
// Set 500k Baud rate for CAN#n, where n varies from 0 to 3. Corresponding can fd values should be passed.
ret_val = set_baudrate(can_fd_0, CAN_SPEED_500K ) ;
if ( ret_val < 0 )
{
    printf("Error while setting the baud rate \n") ;
    exit(0) ;
}
```

Use below macros for setting the different baud rates. These macros can also be found in can.h file.

```
CAN_SPEED_1M
CAN_SPEED_800K
CAN_SPEED_500K
CAN_SPEED_250K
CAN_SPEED_125K
CAN_SPEED_100K
```

CAN_SPEED_50K

CAN_SPEED_20K

CAN_SPEED_10K

CAN Transmit & Receive :

can_tx() & can_rx() : These function will be used to Transmit and Receive the CAN messages respectively.

CAN Transmit Prototype.

```
int can_tx( int can_fd, unsigned char  msgType, unsigned int can_id, int len,unsigned char *data) ;
```

Assign the appropriate values, before calling the can_tx function.

```
can_fd_0: CAN descriptor, return value from init_can() function
msgType = MSG_STANDARD ; //  or MSG_EXTENDED .
can_id = 0x12 ; // CAN ID, if the msgType is MSG_STANDARD then it can be 11-Bit CAN Message ID
        // if the msgType is MSG_EXTENDED then it can be 29-Bit CAN Message ID
len = 4 ; // CAN Transmit Data Length
data : CAN message data.
data[0] = 0x1A ;
data[1] = 0xAB ;
data[2] = 0x22 ;
data[3] = 0x4D ;

ret_val = can_tx(can_fd_0, msgType, can_id, dlc, data) ;
if ( ret_val < 0 )
{
        printf("Error while transmitting the CAN message.\n") ;
        close(can_fd_0) ;
        exit(0) ;
}
```

The above sample code will transmit the CAN Standard message with CAN ID=0x12 of data length=4 and message data = {0x1A, 0xAB, 0x22, 0x4D } ;

CAN Receive Prototype.

```
int can_rx(int can_fd, unsigned char  *msgType, unsigned char *rx_data, unsigned int *can_id, unsigned char *can_msg_len) ;
```

Pass the appropriate pointers for calling the can_rx function.

```
if ( can_rx(can_fd_0, &msgType, data, &can_id, &dlc) )
{
    If (msgType == MSG_STANDARD )
    {
        // Received message is CAN Standard Message.
    }
    else if  (msgType == MSG_EXTENDED)
    {
        // Received message is CAN Extended Message.
    }
    // dlc : Received CAN Data Length
    // can_id : Will contain the CAN Message ID
    // Data of dlc length
    printf("ID=%x DLC=%d Data : ", can_id, dlc) ;
    for (i=0; i< dlc; i++ )
        printf("%x ", data[i] ) ;
    printf("\n") ;
}
```

Example programs for both transmit and receive can be found in the CANLib directory for the reference.

Compiling CAN Application using CANLib Library

Export the Library path using below command.

```
export LD_LIBRARY_PATH=$ LD_LIBRARY_PATH:/path-to-CANLib
```

To compile the application, use the below command.

```
g++ can_app.c -lCAN –L/path-to-CANLib  -o can_app
```