

WHEN YOU NEED TO BE SURE



CAN Bus Interface Setup and Use

Version 6

May 2, 2022

Developed by SGS North America, Inc.

Version History

Version	Date	Revision Description
1	5/23/2016	Initial publication
2	6/26/2017	Added Section 5, Using CAN usb Updated the following sections: <ul style="list-style-type: none"> 2. Required for installation – removed note about the CANdbc driver supporting a subset of CAN Calibration Protocol (CCP) version 2.1 commands. 6. Optional Commands for the CanDbc Driver: <ul style="list-style-type: none"> Revised to include CAN usb Removed commands for CCP 7. Troubleshooting: added column to table for issues with interface card vs. CAN usb.
3	8/23/2018	Format with SGS branding
4	4/3/2020	Retrofit to new template
5	7/20/2021	Corrected <code>byte_order</code> indicators for Motorola (big endian) and Intel (little endian) on page 4 of <i>Section 3.2 Installing the Drivers (kernel and CANdbc)</i> to 0 and 1 respectively.
6	5/2/2022	Added hypertext linked cross-references to usage help in <i>Section 5.2 Opening a Connection</i> on page 6 and in <i>Section 5.4 Optional Commands for the CanDbc Driver</i> on page 7.

Document Conventions

This document uses the following typographic and syntax conventions.

- Commands, command options, file names or any user-entered input appear in Courier type. Variables appear in Courier italic type.
Example: Select the `cmdapp-relVersion-buildVersion.zip` file...
- User interface elements, such as field names, button names, menus, menu commands, and items in clickable dropdown lists, appear in Arial bold type.
Example: **Type**: Click **Select Type** to display drop-down menu options.
- Cross-references are designated in Arial italics.
Example: Refer to *Figure 1*...
- Click intra-document cross-references and page references to display the stated destination.

Example: Refer to *Section 1 Overview* on page 1

The clickable cross-references in the preceding example are *1*, *Overview*, and on page 1.

CyFlex Documentation

CyFlex documentation is available at <https://cyflex.com/>. View **Help & Docs** topics or use the **Search** facility to find topics of interest.

Table of Contents

1	OVERVIEW	1
1.1	CAN BUS OVERVIEW	1
2	INSTALLATION REQUIREMENTS	2
3	INSTALLING THE CAN BUS DEVICE DRIVERS	3
3.1	INSTALLING BOTH DRIVERS	3
3.2	INSTALLING THE DRIVERS (KERNEL AND CANDBC).....	3
4	INSTALLING THE CAN BUS INTERFACE CARD	5
5	USING CAN USB	6
5.1	VIEWING A DEVICE AND CHECKING DRIVERS	6
5.2	OPENING A CONNECTION	6
5.3	CAN USB BAUD RATES	7
5.4	OPTIONAL COMMANDS FOR THE CANDBC DRIVER	7
6	TROUBLESHOOTING	9



LIST OF TABLES

TABLE 1: INSTALLATION FILES AND LOCATIONS 3

TABLE 2: TROUBLESHOOTING SUGGESTIONS 9

1 Overview

This document explains how to configure and use a test cell computer running CyFlex – for connection to a vehicle bus network using the Controller Area Network (CAN) bus protocol.

CyFlex can communicate with and control devices for testing purposes through a CAN bus interface adapter, or through CAN usb.

ⓘ Important:

This document only describes installation of the device drivers. For installation of a CAN bus interface card, refer to the manufacturer's instructions provided with the card.

1.1 CAN Bus Overview

CAN bus is a message-based protocol, designed to work with the vehicle bus standard. CAN bus allows microcontrollers and other devices to communicate with each other without a host computer and is commonly used in automotive applications. "Message-based" means no master controls when nodes (devices) have access to read and write data on the CAN bus.

When a node (device) on the network is ready to transmit data, it checks whether the bus is busy, and then writes a message packet onto the network. The message, called a CAN frame, does not contain addresses of the transmitting node and receiving node(s). Instead, each CAN frame contains an arbitration ID that is unique throughout the network, which assigns an importance level to the message. Messages are transmitted and received based on their importance, which manages communication traffic on the network.

CAN database files are text files used to analyze the contents of CAN bus messages, translating them into engineering units. Vector database files (*.dbc) are an example.

2 Installation Requirements

ⓘ Important:

CyFlex does not currently support using both CAN usb and a CAN interface card in the same computer.

The following are required for setting up CAN bus on a test cell. CAN usb works independently of an interface card and so does not require one.

- CAN-ACx-PCI interface card
- Kernel driver
- CANdbc driver
- CANdbc database file for CAN messages

The files needed for setting up a CAN bus interface are copied during the CyFlex software installation to the test cell directories shown below.

The test cell normally uses a Go-script to launch tasks at start-up.

3 Installing the CAN bus Device Drivers

Table 1 lists installation files and their respective locations.

Table 1: Installation Files and Locations

File	File Name	Directory Location
Kernel driver	vcanpcid	/cyflex/bin/modules/
CANdbc driver	CanDbc	/cyflex/bin/
CANdbc database file	custom_application.dbc	/specs/
Go-script	go.scp	/cell/

3.1 Installing Both Drivers

Use the following procedure:

1. Install both drivers.
2. Add the CANdbc driver to the Go-script (optional).
3. Edit the .dbc database file.
4. Skip this step if configuring the system to use CAN usb. Otherwise, install the CAN bus interface card according to the manufacturer's instructions.
5. Reboot the computer.
6. Start the drivers.

3.2 Installing the Drivers (kernel and CANdbc)

Use the following procedure:

1. Edit the `rc.local` file to make sure the kernel driver is loaded after each reboot:
 - a. Open a terminal window.
 - b. Enter:


```
Edit /etc/rc.d/rc.local
```
 - c. Locate these lines in the file:


```
## cd /cyflex/bin/modules/  
## /cyflex/bin/modules/tc_load  
## cd -
```
 - d. Add these lines below the ones found in the previous step:


```
cd /cyflex/bin/modules/  
/cyflex/bin/modules/vcan_load  
cd -
```
2. Add the CANdbc driver application to the Go-script file:
 - a. Enter:


```
Edit /cell/go.scp
```
 - b. In the file, locate the line:


```
gen_labels
```


c. Add one of the following lines, either immediately before or after the `gen_labels` line:

For use with a CAN interface adapter, add this line:

```
CanDbc /specs/eci_ecu.dbc 12 &
```

OR,

For use with CAN usb, add:

```
CanDbc eci_ecu.dbc /dev/pcanusb0 12 &
```

3. Edit the `.dbc` database file as needed using an editor.

The `CANdbc` driver requires the `.dbc` database file, which specifies which CAN messages the `CANdbc` driver can receive and send. The driver accesses the `.dbc` database file by reading it into memory at startup.

Below is an example of the `.dbc` file, using colors to indicate required information in the `.dbc` file.

```
BO_ 100 IMEP_100: 8 IndiComCANOutRTP
  SG_ IMEP1 : 0|16@1+ (0.000335698,-2) [-2|20] "" Vector__XXX
  SG_ KP_PK1 : 16|16@1+ (0.0015259,0) [0|100] "" Vector__XXX
  SG_ AI50_1 : 32|16@1+ (0.00184634,-31) [-31|90] "" Vector__XXX
  SG_ AI90_1 : 48|16@1+ (0.00184634,-31) [-31|90] "" Vector__XXX
```

- `BO_` indicates a message.
- `SG_` indicates a signal.

<CyFlex variable name>

`offset` in bits from start of message

`size` in bits

`byte_order` 0 is Motorola (big endian), 1 is Intel (little endian)

value type + is unsigned, - is signed

(`factor`, `offset`) `physical_value` = `raw_value` * `factor` + `offset`

[`minimum`, `maximum`] minimum value, maximum value

"" Used to specify units receiver of the message or `Vector__XXX`

4 Installing the CAN bus Interface Card

Refer to the manufacturer's instructions provided with the CAN bus interface card.

Use the following procedure:

1. Install both drivers.
2. Add the `CANdbc` driver to the Go-script (optional).
3. Edit the `.dbc` database file.
4. Skip this step if configuring the system to use CAN usb. Otherwise, install the CAN bus interface card according to the manufacturer's instructions.
5. Reboot the computer.
6. Start the drivers.

5 Using CAN usb

Use CyFlex version 6.2.9 or above.

The following sub-sections describe how to view a CAN usb device, open a connection, and set the baud rate.

5.1 Viewing a Device and Checking Drivers

Verify the usb driver installation by issuing the following command:

```
$ yum list pcandriver
```

If nothing is returned, use the following command to perform the installation:

```
$sudo yum install pcandriver
```

To view the CAN usb device:

1. Plug-in the device to a USB port.
2. At a terminal window, enter:

```
ls /dev/*pcan*
```
3. This displays the following (or similar):

```
/dev/pcan32 /dev/pcanusb0
```

The two device names displayed are actually the same CAN usb device.

Note:

Although using either name is acceptable, it is recommended to use the name convention that includes “usb” to make keeping track of the devices easier (for example: `pcanusb0`, `pcanusb1`, etc.). Connecting a second device displays four device names.

5.2 Opening a Connection

Refer to the following command line examples:

- Open a connection to the device at `/dev/pcanusb0` using the `cyflex.dbc` file with priority 12:

```
CanDbc cyflex.dbc /dev/pcanusb0 12 &
```
- Open a connection to the device at `/dev/pcanusb0` with a baud rate of 250000:

```
CanDbc cyflex.dbc /dev/pcanusb0 baud1=250000 12 &
```
- To open two device connections at `/dev/pcanusb0` and `/dev/pcanusb1`:

```
CanDbc cyflex.dbc /dev/pcanusb0 cyflex2.dbc /dev/pcanusb1  
12 &
```
- Open two device connections at `/dev/pcanusb0` and `/dev/pcanusb1` with the baud rates set to 250000 and 125000 respectively:

```
CanDbc cyflex.dbc /dev/pcanusb0 baud1=250000  
cyflex2.dbc /dev/pcanusb1 baud2=125000 12 &
```

Refer to [CanDBC](http://www.cyflex.com) on [cyflex.com](http://www.cyflex.com) for complete usage help .

5.3 CAN usb Baud Rates

Supported baud rates for CAN usb are:

- 1000000
- 500000
- 250000
- 125000
- 50000
- 20000
- 10000
- 5000

5.4 Optional Commands for the CanDbc Driver

The CanDbc driver can read/write up to two different CAN buses and can also communicate with a CAN USB adapter. Below are some of the arguments for CanDbc. Refer to [CanDBC](#) on cyflex.com for complete usage help .

- **Filenames:** used to specify the .dbc to use. The first .dbc file used is for the first CAN bus. The second file is for the second CAN bus.

Example:

```
CanDbc /specs/ECU_can.dbc OR CanDbc /specs/ECU_can.specs
/specs/lab.dbc
```

- **Time:** specifies how often it will try to update in milliseconds, default is 20.

Example:

```
CanDbc /specs/ECU_can.dbc time=50
```

- **Reg_name:** Used to run multiple CanDbc tasks. Only needed to use 3 or more CAN bus connections. Default is CanDbc.

Example:

```
CanDbc /specs/ECU_can.dbc name=CanDbcNOX
```

The following commands may be used to control the CanDbc driver. For more details on any of the commands, run a man command on the tasks. The (#) symbol below represents a comment.

- `candbc_set <CanMessageId|MessageName> <SendInterval> [index] [-c=n]`

Purpose: Tells the driver to send CAN message ID 100 over 50ms.

Example:

```
$ candbc_set 100 50
```

Refer to [candbc_set](#) on cyflex.com for complete usage help.

- `candbc_list`

Purpose: Lists the active CAN messages requested by any of these commands:

`candbc_set`, `candbc_ccp_get`, or `candbc_ccp_set`

Refer to [candbc_list](#) on cyflex.com for complete usage help.

- `candbc_clear`
Purpose: Clears the list of CAN messages.
Refer to [candbc_clear](#) on cyflex.com for complete usage help.
- `candbc_ccp_get <Variable Name> <Interval> [index] [-c=n]`
Purpose: Tells the driver to get a variable declared in the .dbc from the ECM and store the result in the CyFlex variable.
Examples:

```
$ candbc_ccp_get rpm 1000  
# Retrieve the 'rpm' parameter 1000ms  
$ candbc_ccp_get rpm 1  
# Retrieve the 'rpm' once.
```

Refer to [candbc_ccp_get](#) on cyflex.com for complete usage help.
- `candbc_ccp_set <Variable Name><Interval>[value] [-c=n]`
Purpose: Tells the driver to send a value for the variable once or repeatedly.
Examples:

```
$ candbc_ccp_set rpmd_gov 1 950  
# Send '950' to the ECM for 'rpmd_gov'  
$ candbc_ccp_set rpmd_gov 1000  
# Send the current value of CyFlex 'rpmd_gov' to the ECM  
once a second.
```

Refer to [candbc_ccp_set](#) on cyflex.com for complete usage help.
- `candbc_ccp_cmd <ccp_command> [data] [-c=n]`
Purpose: Sends a command that controls the Can calibration protocol in the CanDbc driver.
Refer to [candbc_ccp_cmd](#) on cyflex.com for complete usage help.

6 Troubleshooting

If a problem occurs after configuring the system and restarting the computer, refer to the suggestions in below.

Some recommendations depend on the configuration: CAN bus interface adapter vs. CAN usb.

For hardware-related information, such as pin jumper position, refer to the manufacturer's instructions provided with the CAN bus interface card.

Table 2: Troubleshooting Suggestions

Suggestion	Comment	Applies To
Check all connections and cables.	A connection may be loose or intermittent.	Interface Card or CAN usb
Verify the <code>/etc/softing</code> directory exists.	After the <code>vcan_load</code> commands (in the kernel driver) run, there should be a file called <code>candevices.conf</code> , present in the <code>/etc/softing</code> directory.	Interface Card or CAN usb
Check the boot sequence for a failed message specific to the interface card.	In most cases, boot logs are located in the directory, <code>/var/log</code> and named <code>boot.log</code> , <code>boot.msg</code> or similar.	Interface Card
View connected hardware and the associated drivers.	Open a terminal window and enter command <code>lspci</code> to list all devices connected to the PCI bus. Note: For CAN usb, view connected devices as described in <i>Section 5 Using CAN usb</i> on page 6.	Interface Card
Check the kernel buffer log.	The <code>dmesg</code> command displays all kernel messages in the buffer, including hardware initialization messages that may indicate problems or conflicts. Enter command <code>dmesg</code> to view the log, which is typically named <code>var.log</code> and kept in the directory <code>/var/log/</code> .	Interface Card or CAN usb
Check that the drivers are loaded.	Use command <code>lsmod</code> , to report which modules are loaded into the kernel.	Interface Card or CAN usb