
PISO-CAN200-D/T PISO-CAN400-D/T LabVIEW CAN Driver User's Manual

Warranty

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICP DAS assume no liability for damages consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Copyright 2005 by ICP DAS. All rights are reserved.

Trademark

The names used for identification only maybe registered trademarks of their respective companies.

Tables of Content

1 General Information.....	3
1.1 CAN Bus Introduction.....	3
1.2 LabVIEW CAN Driver Characteristics.....	4
1.3 Configurator Specification.....	5
1.4 LabVIEW CAN Driver Specification.....	5
2 Software Installation.....	6
2.1 Installation Driver Step by Step.....	7
3 Configurator Description.....	13
3.1 Operation Interface.....	13
3.2 CAN Hardware Setting.....	16
4 Driver Function Description.....	27
4.1 Function Definition.....	27
4.2 LabVIEW CAN Driver Application Flowchart.....	29
4.3 Function Description.....	30
4.3.1 ExpCANStart.....	30
4.3.2 ExpCANStop.....	31
4.3.3 ExpCANin.....	32
4.3.4 ExpCANout.....	33
5 Demo Program for Window.....	34
5.1 1xCAN_1xRTR Configurator Settings of Demo1.....	34
5.2 1xCAN_1xRTR LabVIEW Program of Demo1.....	41

1 General Information

1.1 CAN Bus Introduction

The CAN (Controller Area Network) is a serial communication protocol, which efficiently supports distributed real-time control with a very high level of security. It provides the error process mechanisms and message priority concepts. These features can improve the network reliability and transmission efficiency. Furthermore, CAN supplies the multi-master capabilities, and is especially suited for networking “intelligent” devices as well as sensors and actuators within a system or sub-system. In CAN networks, there is no addressing of subscribers or stations in the conventional sense, but instead prioritized messages are transmitted.



Figure 1.1 Example of the CAN network

The CAN system is generally applied in the embedded systems as well as automation control applications. More and more application stories are proposed recently. The main CAN application fields include passenger car, off-highway and off-road vehicle, maritime electronics, factory automation, lifts and escalators, building automation...etc.

1.2 LabVIEW CAN Driver Characteristics

ICP DAS LabVIEW CAN DLL driver provides users to establish CAN communication network rapidly. It is special for ICPDAS PISO-CAN200/400 PCI interface card. By the way of using the library, most of the CAN communication protocols will be handled by the VI modules of this DLL library automatically. Therefore, it can help users reducing the complexity of developing a CAN network without the detail CAN bus technology background. The general application architecture is demonstrated as the Figure 1.2.

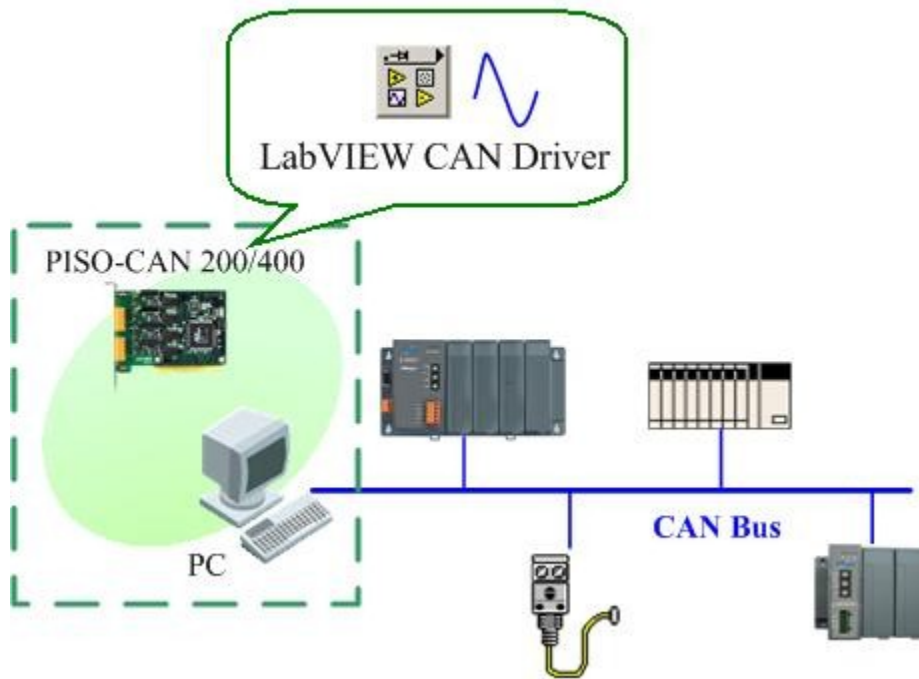


Figure 1.2 Application architecture

1.3 Configurator Specification

- Supported for Windows 2000/XP.
- Supported PISO-CAN 200/400-D/T.
- Two languages interface, German and English.
- The range of sample rate is from 50Hz to 5000Hz.
- Block size range is from 1 to 4096.
- Provide Intel mode and Motorola mode for remote CAN device.

1.4 LabVIEW CAN Driver Specification

- Support LabVIEW 7.1
- Driver is only for Windows 2000/XP platform
- Provide 8 kinds of CAN baud rate, such as 10K, 20K, 50K, 125K, 250K, 500K, 800K, and 1M.
- Allow user-defined CAN baud rate.
- Two languages interface, German and English.
- Support both CAN 2.0A and CAN 2.0B specification.
- Provide debug mode for receive module.

2 Software Installation

The LabVIEW CAN DLL driver is the CAN function collections for the PISO-CAN200/400 CAN card used on LabVIEW 7.1 environment on Windows 2000/XP. Before users use this driver to develop the CAN interface, the PISO-CAN200/400 driver must be installed because the LabVIEW CAN driver need to call the function of PISO-CAN200/400 driver. The driver architecture is shown in the following Figure2.1.

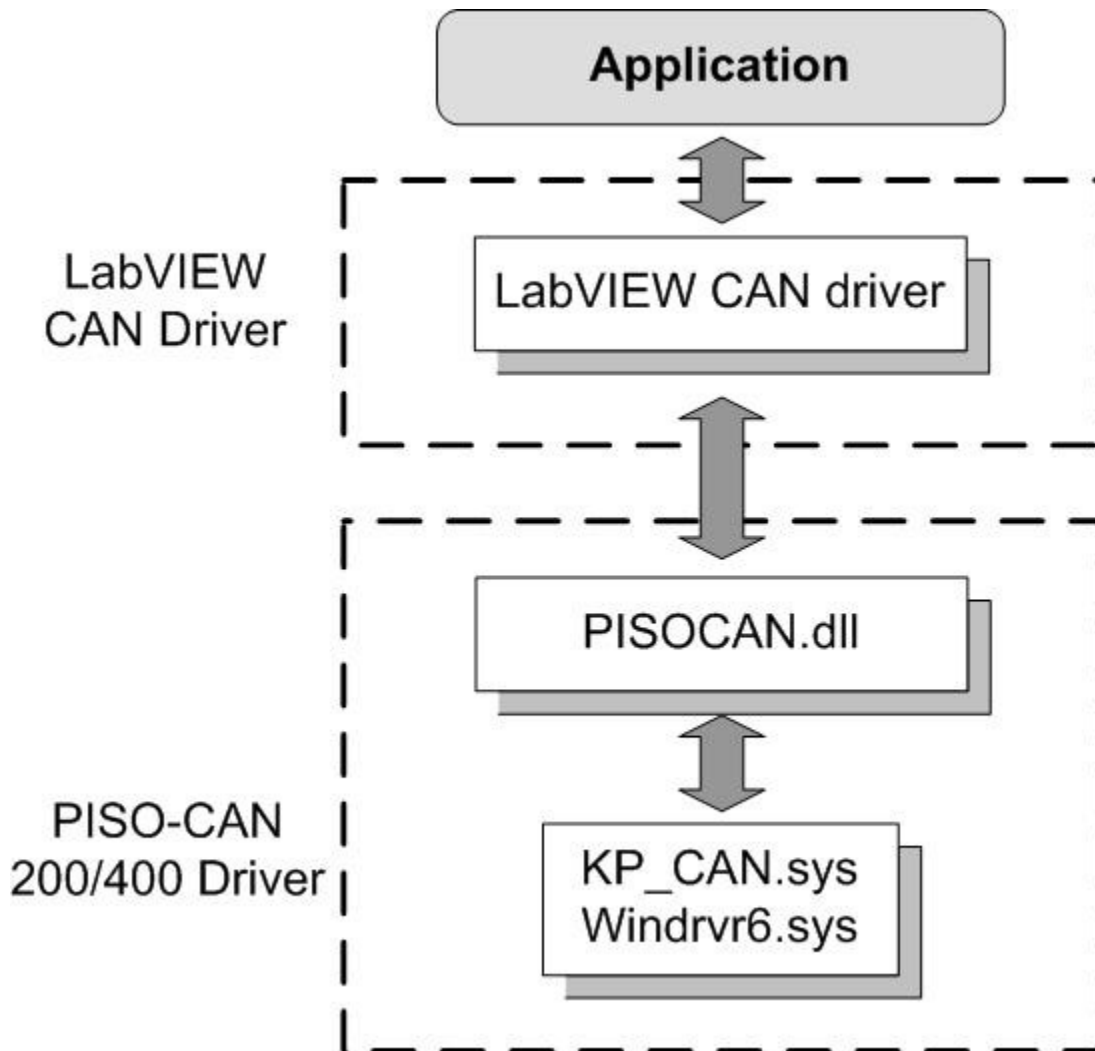


Figure 2.1 Driver concept of LabVIEW CAN driver

2.1 Installation Driver Step by Step

When users want to use the LabVIEW CAN driver, the PISO-CAN200/400 CAN card driver and LabVIEW 7.1 must be installed firstly. Afterwards, users should install LabVIEW CAN driver. After finishing the installation process, the demo programs may be a good reference for users to build a simple CAN LabVIEW application. In the demo programs, basic functions and operation methods of CAN VI blocks are introduced. It is very helpful for users to understand how to use these functions and how to develop their CAN applications. For more information about the demo programs, please refer to the section 5. The following description shows the step-by-step procedures about how to install the PISO-CAN200/400 driver and LabVIEW CAN driver.

Install the PISO-CAN200/400 CAN card driver

Please refer to the PISO-CAN200/400 user manual. It can be found in the product package box. Users also can download it in the following website.

<http://www.icpdas.com/oldweb/download/pci/piso-can/manual>

Install the LabVIEW CAN Driver

Step 1: Insert the product CD into the CD-ROM and find the setup file with the path `\CAN\PCI\PISO-CAN200_400\LabVIEW_CAN_driver\` or download it from the web site:

<http://www.icpdas.com/download/pci/piso-can/index.htm>.

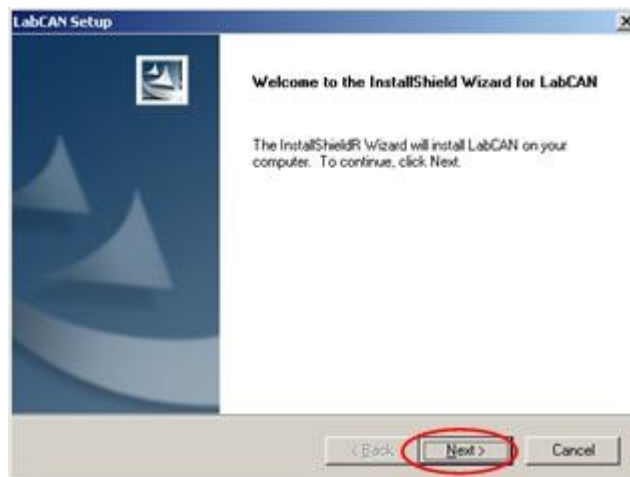
Then, execute the LabCAN.exe to install the LabVIEW CAN driver.



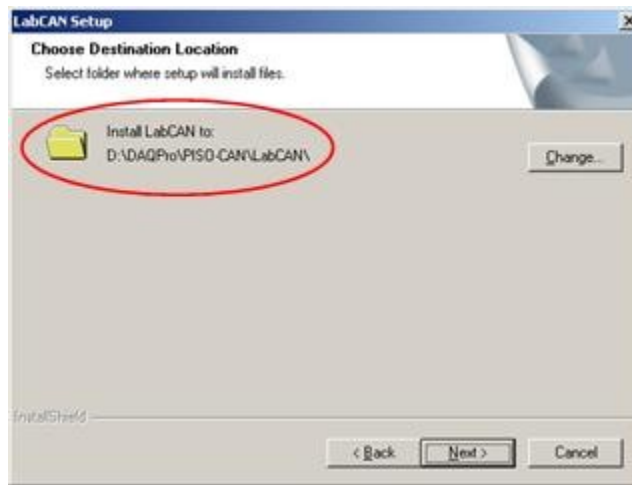
Step 2: Wait until the installation wizard has prepared.



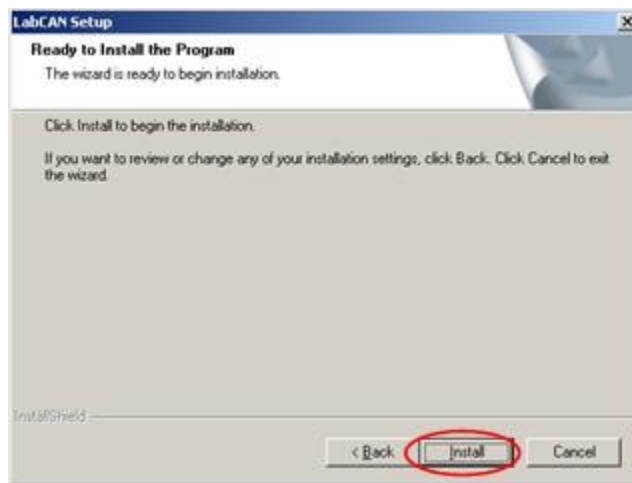
Step 3: Click “Next” to start the LabVIEW CAN driver installation.



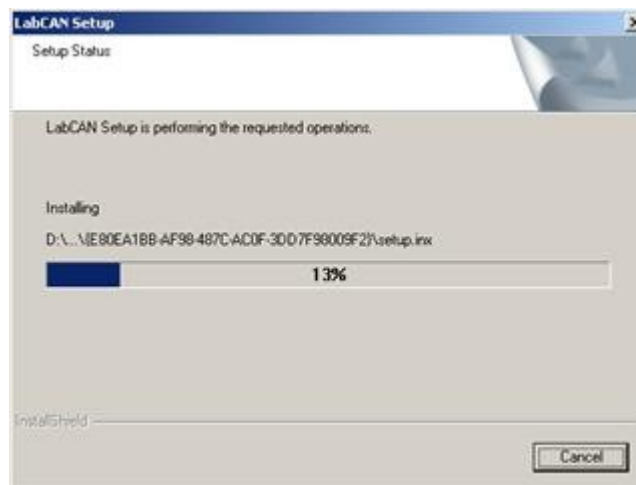
Step 4: Select the folder where the driver will be installed and click “Next” button to continue. Here, default path is used.



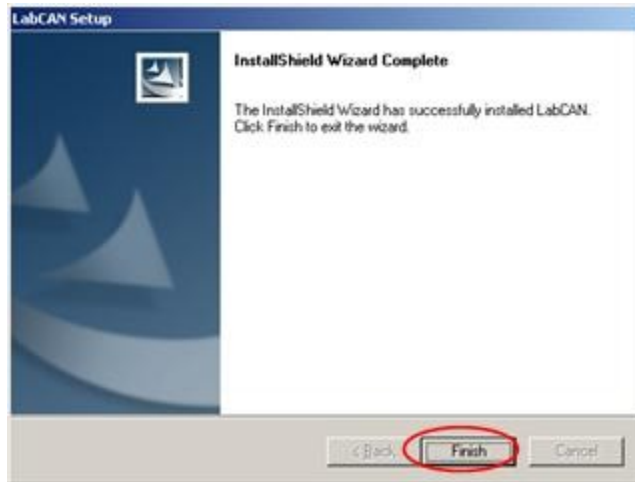
Step 5: Click the button "Install" to continue.



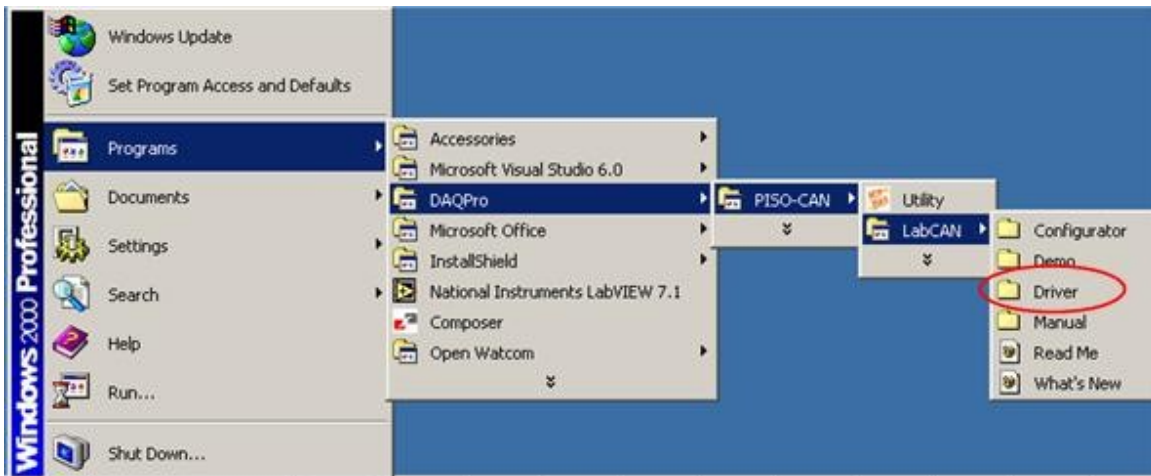
Step 6: Wait for the LabVIEW CAN driver installation.



Step 7: After finishing the process, click “Finish” to complete the installation.

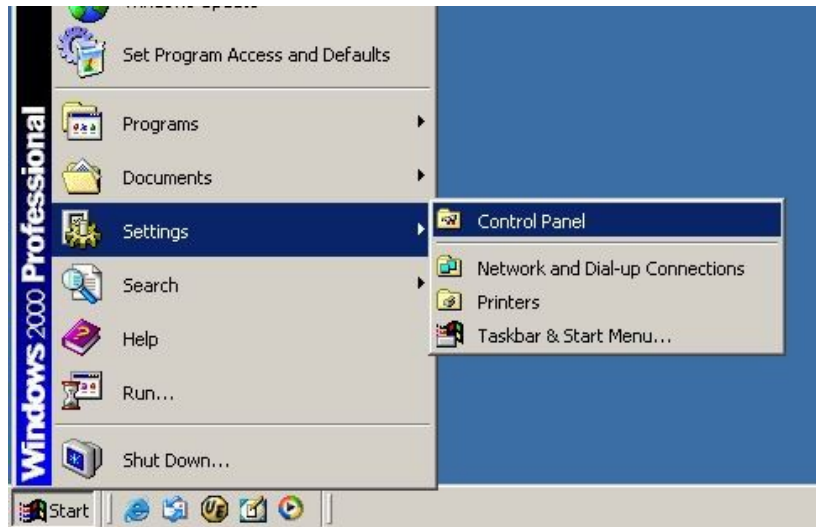


Step 8: When finishing the LabVIEW CAN driver installation, the LabCAN folder will be found at the Start menu shown as below.



Remove the LabVIEW CAN Driver

Step 1: Click “Start” in the task bar. Select the Settings/Control Panel as shown in the following figure.



Step 2: Click the “Add/Remove Programs” icon to open the dialog.



Step 3: Find out the LabCAN, and click the button “Change/Remove”.



Step 4: Choose “Remove” option and click the button “Next” to remove the software.



Step 5: Click the button “Yes” to remove the software.



3 Configurator Description

Configurator is the PISO-CAN200/400 hardware setting toolkit for LabVIEW CAN driver. It can be found in the following path:

[\DAQPro\PISO-CAN\LabCAN\Configurator](#)

It must be executed before using the LabVIEW CAN driver. After user executes the configurator.exe, the .ini file will be produced in the “setup” folder whose path is the same as configurator.exe. These .ini files will provide configuration information about CAN baud rate, arbitration ID, data length, and etc. Afterwards, LabVIEW CAN driver uses the configuration information to implement the VI block in LabVIEW environment.

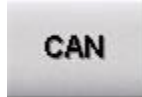
3.1 Operation Interface

Configurator is a simple and easy-to-use configuration tool. Figure 3.1 is the Configurator user interface. It has only seven buttons, and the functions of these buttons are explained as follow.



Figure 3.1 Configurator Interface

CAN Button



This button includes the main configuration function. It help users to set the CAN communication parameters, such as CAN baud rate, VI block sample rate, arbitration ID, data length, etc. For more detail information, please refer to the section 3.2.

DAQ Button



It is a reserved button for further use.

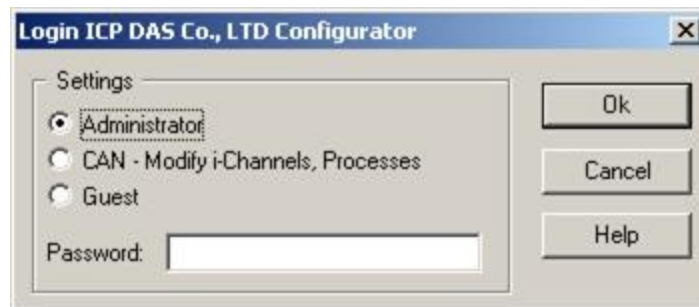
User Button



If users do not hope that the settings of Configurator can be changed by other people. Checking “User Management with Login dialog“ item can lock the Configurator.



Afterwards, if users want to change the Configurator settings, users need to select the Administrator item and key the correct password “Admin” in the Password filed. Or, only the Guest item can be used for view the Configurator settings.



Language Button



There are two languages, German and English, for user to select. Use the language button to change which language the user-interface window will be.



Save Button



It is a reserved button for further use.

Help Button



Click this button to show the help information of LabVIEW CAN driver and Configurator. These help information is made by using both Germany and English interface.

Close Button




Click this button to exit Configurator.

3.2 CAN Hardware Setting

In this section, operation method of Configurator will be described. There are three tabs, Hardware, i-Channel, and Process in the setting window. They are explained below.

Hardware




In hardware tab, users can double click CAN channel icon to set CAN baud rate for each PISO-CAN200/400 CAN port on PCI bus. For example, double click CAN0 icon  to set the Baud of CAN port 0. After clicking the CAN port icon, the "Port Settings" window pops up.

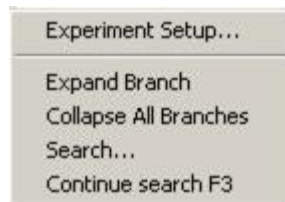
If checks the "Baud Rate from list" combo box, there are 8 kinds baud rate to select.



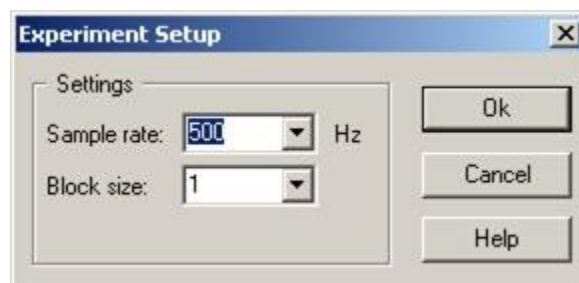
And if users check the second item, “Baud Rate over Timing Register”, users must have the background of the CAN chip, SJA1000. The CAN baud rate is calculated by using SJA1000 specification with 16MHz oscillator. For example, setting the BTR0 = 01H and BTR1 = 1CH of SJA1000 can get 1M bps baud rate.



Besides, users can right click the hardware icon  Hardware to do some configurations and adjustments. The functions of these items are shown below.



1. Select “Experiment Setup” to set the sample rate and block size for all VI blocks of all CAN channels.



Sample rate parameter defines the sampling rate for all channels. The range of sample rate is from 50Hz to 5000Hz, and the minimum increase step is 50Hz. For example, if set 4430Hz to the sample rate value, it will be round to 4450Hz automatically. If type 4420Hz to the sample rate value, it will be truncated to 4400Hz automatically. Block size parameter will influence the size of latency period. The range of block size is from 1 to 4096. This parameter strongly influences the real-time performance of the system and the speed of the measuring process.

2. Use “Expand Branch” function to expand the branch. Then, the tree view shows the detail information.



3. The “Collapse All Branches” function can concentrate the branch on one icon. Then, the tree view is more compact

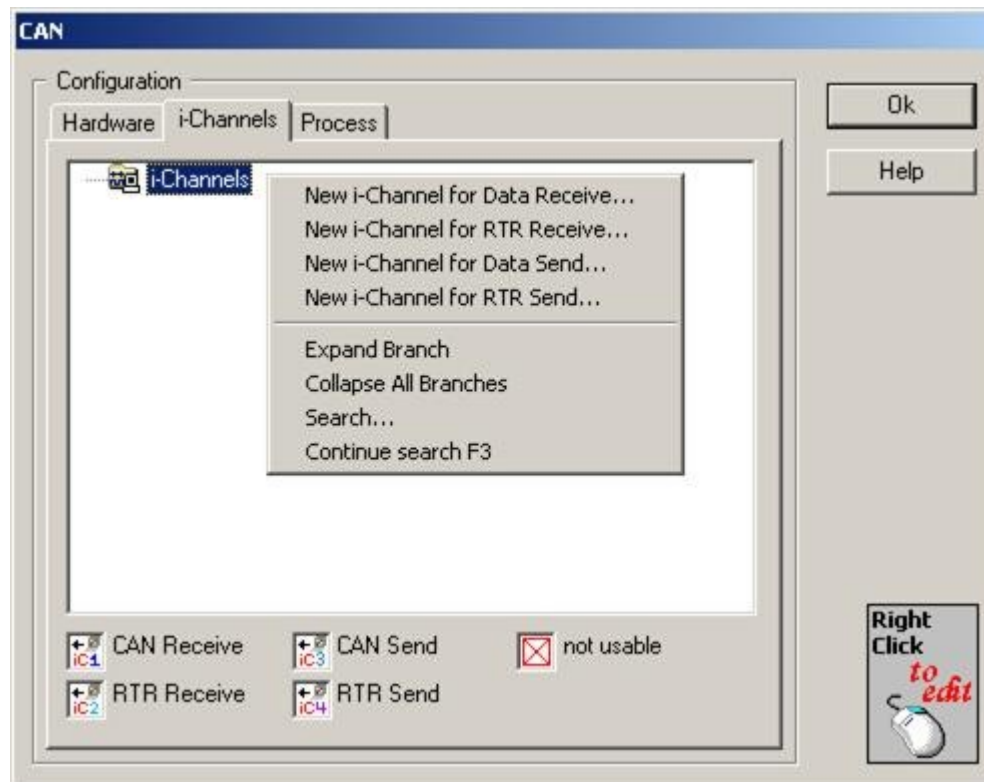



-
4. Use “Search...” function to search string that user is looking for. For example, when user inputs the string “CAN3” and press ok, the “CAN3” in tree view will be found. If the “Case sensitive” check box has been checked, the string that you key in the window must match case with the target string that you want to search.




5. Press key “F3” or use the “Continue search F3” function to search the next one.

i-Channel

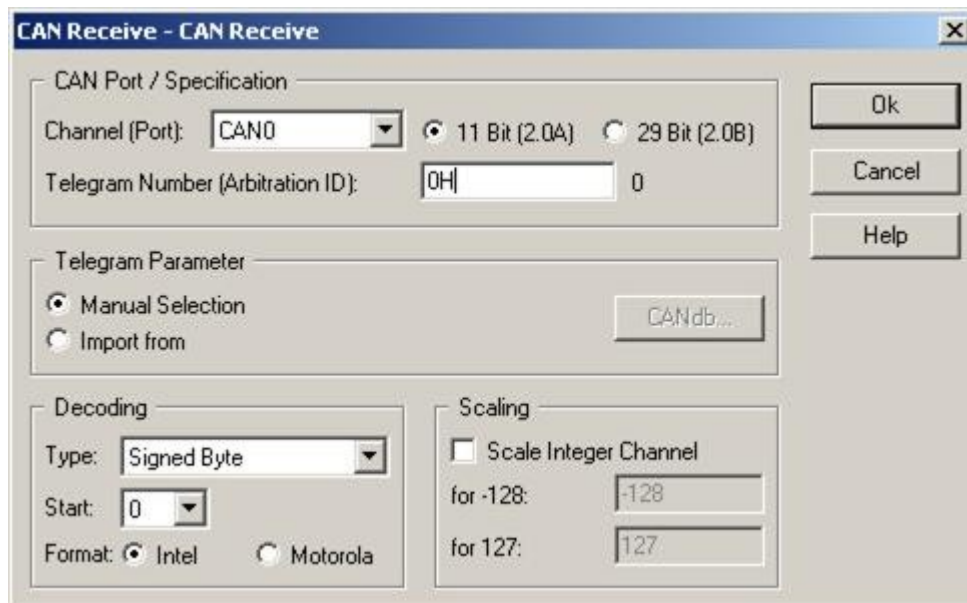


This tab is used to set the arbitration ID that wants to send or receive by specific CAN channels. There are four kinds of channels, CAN send, CAN receive, RTR send, and RTR receive. When right click the i-Channels icon  , users need to choose one kind of channels for addition.


The “CAN send”/“CAN receive” channels can send/receive CAN data. The “CAN send” channel provides the arbitration ID, data type, and data length for configuration. And, “RTR send”/“RTR receive” channels can send/receive remote-transmit-request message to/from the CAN network. Take “CAN receive” channel for example, right click  and select “New i-Channel for Data Receive...” to add a new data receive channel. When add a channel, user must input a channel name below.

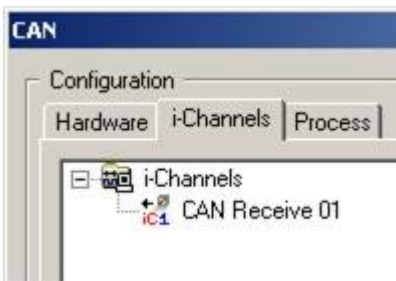


After setting the name, some parameters must be setting for the channel. Choose the Channel (Port) to define the CAN port which the CAN receive channel will receive the CAN message from. Select the CAN specification 2.0A/2.0B to decide which CAN specification is used. Set the arbitration ID. Then the CAN receive channel will receive CAN data according to this arbitration ID. The telegram parameters have two part, Decoding, and Scaling. If you select the Manual Select item, the parameters in Decoding and Scaling frame must be configured by yourself. Or, you can select the Import from to import these parameters configuration from the .dbc file produced by CANdb editor. The CANdb editor is made by Vector Company. Please refer to Vector web site for more information.

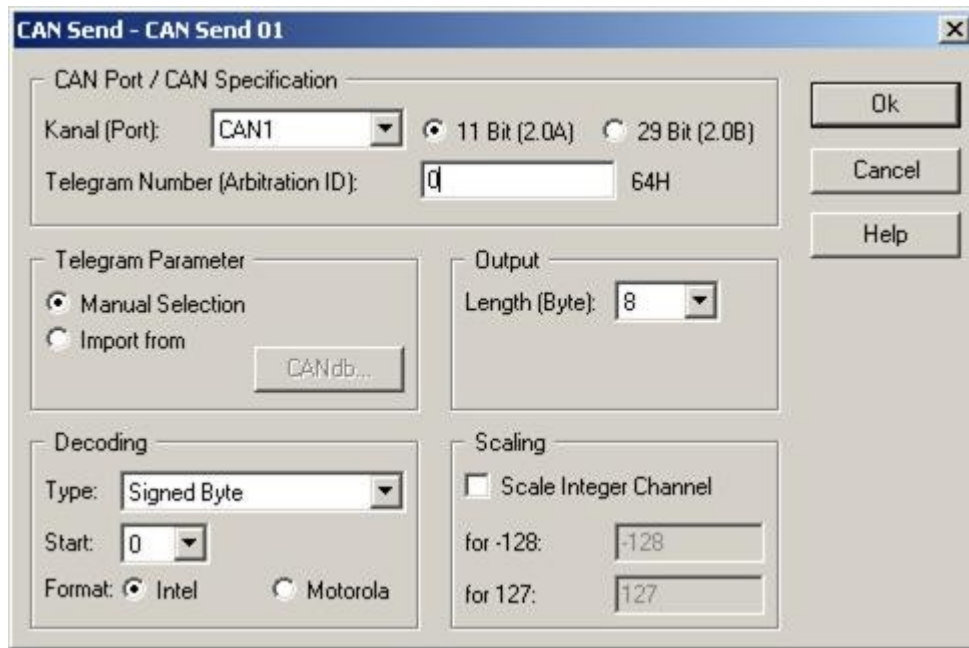


In Decoding frame, the “Type” combo box decides the decoding method of received message. It has seven kinds of integer data type, such as single bit, signed byte, unsigned byte, signed short integer, unsigned short integer, signed long integer, unsigned long integer. It also provides two kinds of float types float 32 bit IEEE and double 64-bit IEEE. The “Start” combo box can select the start byte of the data. The message decoding will start at this byte. For example, if type is unsigned short integer and start is 2, the receive channel will output the unsigned short integer value which is decoded from the 3rd and 4th bytes of received CAN message. The Scaling parameter is used to mapping the received data to the real physical value. The original scaling range is given by the Type parameter setting. For example, if the unsigned short integer is used, the original range is from 0 to 65535. The mapping range is according to user’s application. If users application need to obtain the voltage from 0V~10V, the mapping range is 0~10. After setting the scaling 0 and 10, the 0~65535 of decoded value is mapping to the 0~10 application range.

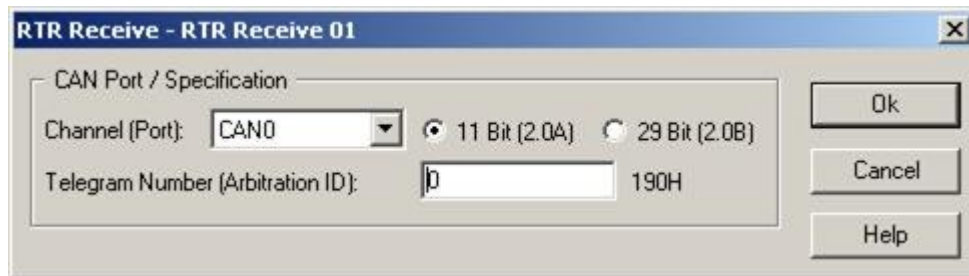
If users finish the configuration, the data receive channel icon  will be added to the i-Channel tree view as follows.



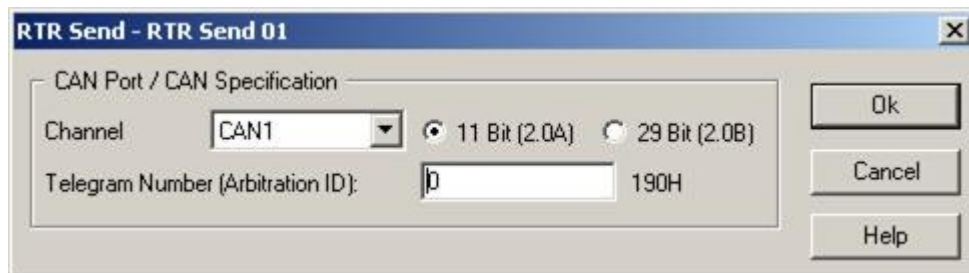
In order to add CAN send, RTR receive, or RTR send channel, user must to input a channel name first, too. For CAN send channel, its setting is almost the same as CAN receive channel, except the data length parameter. The data length parameter is used to set the output data length from 1 byte to 8 bytes. For RTR receive and RTR send channel, this two kinds of channels only have two parameters, CAN port and arbitration id. These parameters configurations are the same as those in the CAN receive channel. The setting window of CAN send, RTR receive, and RTR send channel is shown below respectively.



CAN send channel setting window

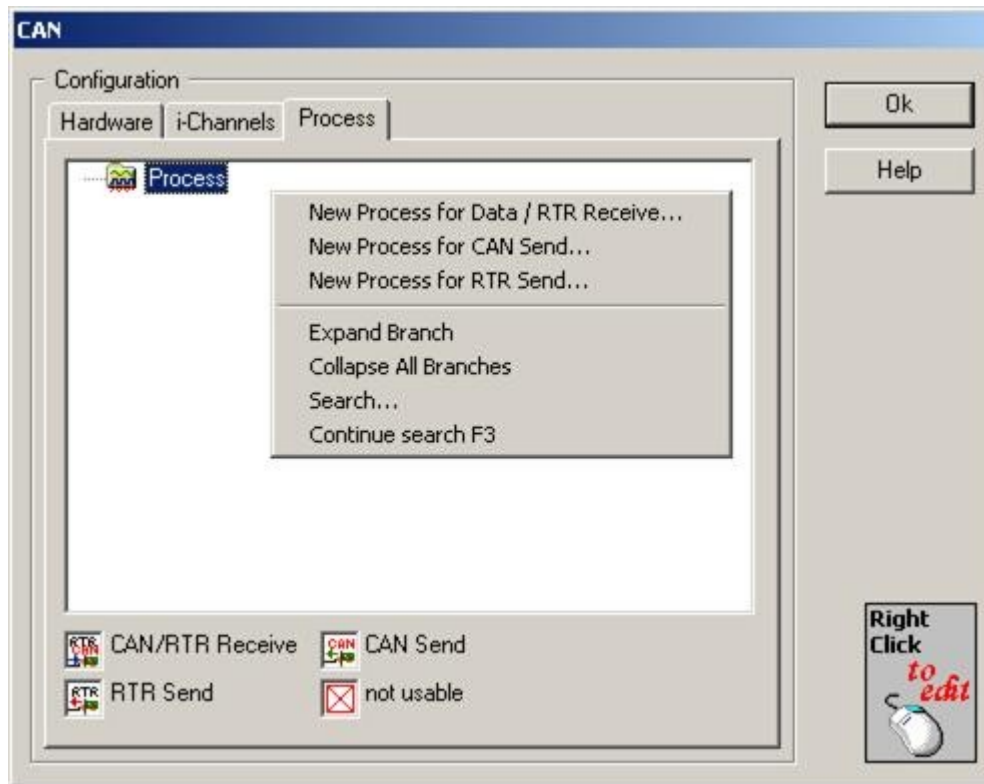



RTR receive channel setting window



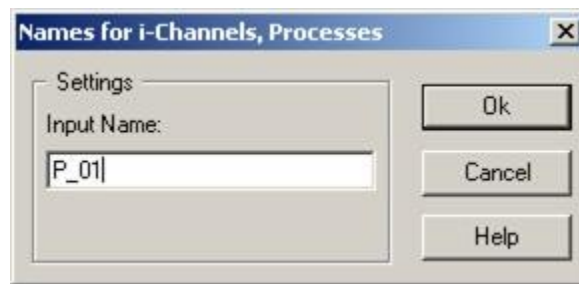
RTR send channel setting window

Process

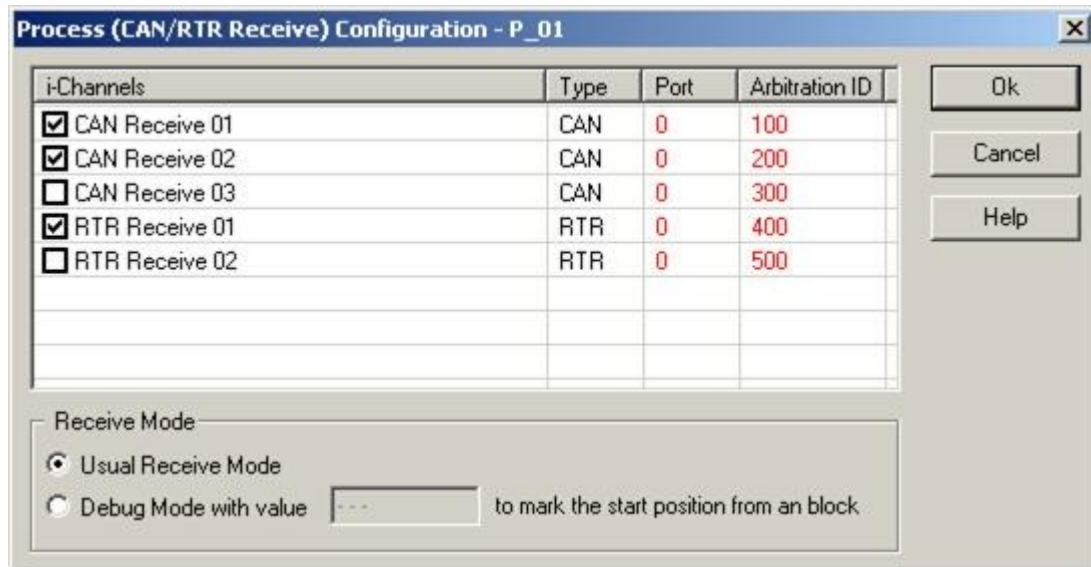



After finishing i-Channel configuration, Process interface can enable the send or receive channels defined in i-Channel window, and use these channels in the LabVIEW CAN VI or subVI. Right click the process icon  Process, there are three kinds of process to be able to add. A Data/RTR Receive process can include more than one CAN receive channel or RTR receive channel. But a "CAN send process" / "RTR send process" can use only one "CAN send channel" / "RTR send channel". Take the Data/RTR receive process for an example, when users add this process, input a channel name below.

Note: Process name cannot exceed 7 characters.



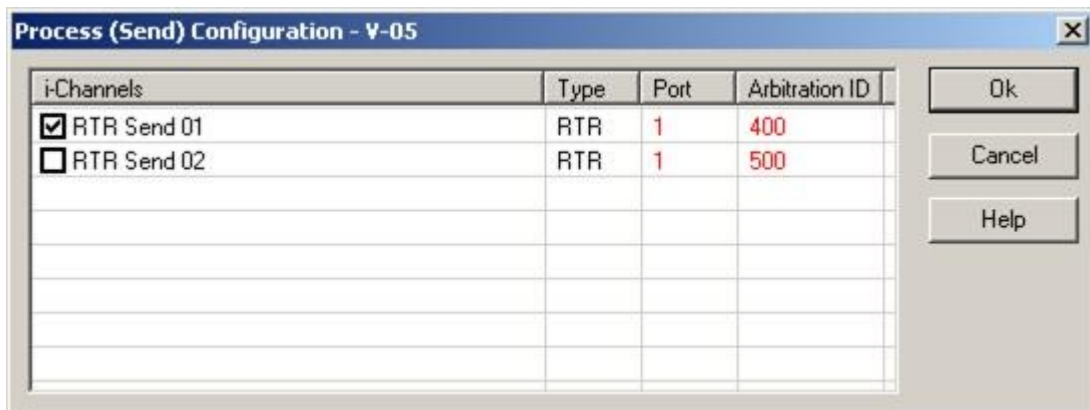
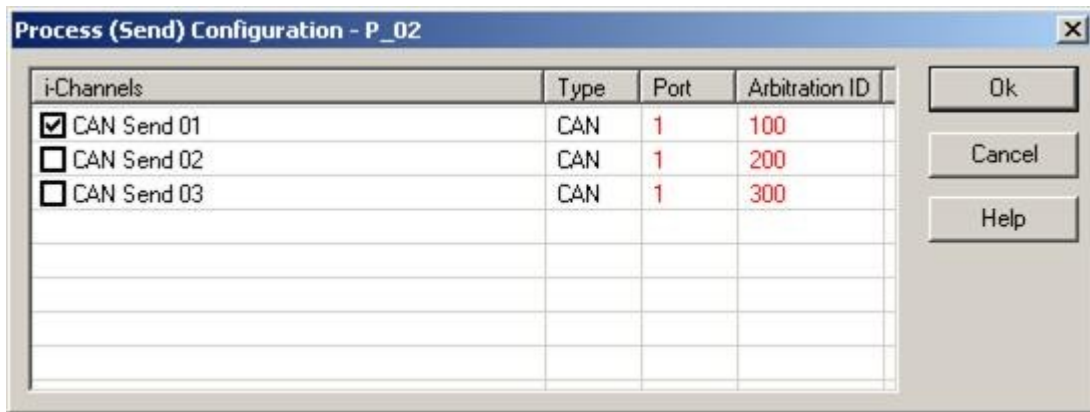
Then, click Ok button, and the configuration window will pop up. All of the receive channels defined in the i-Channel window before will display in the configuration window. Users can check the check box to select which needs to be applied in this process.



If the configuration is finished, click Ok button. Then, the CAN/RTR receive icon  will be added in the process view.











If users add a CAN send process, the entire CAN send channels defined in the i-Channel window will be shown after giving the CAN send channels a name and clicking Ok button. But users can check only one of them, and apply the checked channels in the CAN send process. The situation in RTR send process is the same as that in CAN send process. The following two figure point out these condition.



4 Driver Function Description

4.1 Function Definition

Every function provides two kinds of VI for user to use in the LabVIEW environment. They are standard subVI and Express VI. The main differences of these two VI are the design flexibility. In Express VI, some of parameters are decided by users when the VI is put into the LabVIEW block diagram. Therefore, when running the users LabVIEW program, only few parameters of Express VI could be set dynamically. The standard subVI provides all the parameters to modify even if the users LabVIEW program is running. It is more flexibility to change the VI parameters. However, the Express VI could be easier to use because of its compact functionality. Users can decide which kind of VIs is proper for users application. All the functions provided by the LabIcpdas.dll are listed in the following table. The detail information and operation method for each function will be presented in the section 4.3.

Function Icon		Function Name	Description
Express	Standard		
 <p>ExpCANStart</p> <p>▶ error in (no error)</p> <p>error out ▶</p>	 <p>▶ chProcessName</p> <p>▶ error in (no error)</p> <p>▶ Language</p> <p>error out ▶</p>	ExpCANStart	ExpCANStart VI activates the CAN card and applies all desired processes under LabVIEW.
 <p>ExpCANStop</p> <p>▶ SyncInput</p> <p>▶ Language</p> <p>SyncOutput ▶</p>	 <p>▶ Language</p> <p>▶ SyncInput</p> <p>SyncOutput ▶</p>	ExpCANStop	This VI closes the CAN card and releases all the processes applied to LabVIEW.
 <p>ExpCANin</p> <p>▶ Language</p> <p>▶ error in (no error)</p> <p>error out ▶</p> <p>FifoPercent ▶</p> <p>mprVal Out ▶</p> <p>Ready ▶</p>	 <p>▶ chProcessName</p> <p>▶ error in (no error)</p> <p>▶ Language</p> <p>error out ▶</p> <p>FifoPercent ▶</p> <p>mprVal Out ▶</p> <p>Ready ▶</p>	ExpCANin	The VI receives all input data from receive process applied by ExpCANStart.
 <p>ExpCANout</p> <p>▶ Language</p> <p>dValue</p> <p>▶ error in (no error)</p> <p>error out ▶</p>	 <p>▶ chProcessName</p> <p>▶ dValue</p> <p>▶ error in (no error)</p> <p>▶ Language</p> <p>error out ▶</p>	ExpCANout	ExpCANout VI can send the CAN data via send process applied by ExpCANStart.

4.2 LabVIEW CAN Driver Application Flowchart

In this section, it is described that the operation procedure about how to use the LabVIEW CAN driver to build users' applications. This information is helpful for users to apply the LabVIEW CAN driver easily. Furthermore, we provide a demo program to show how to use these VI step by step in section 5. It would be a good reference for users to learning the operation method of LabVIEW CAN driver.

When users' programs apply the LabVIEW CAN driver functions, the function ExpCANStart must be call first. This function is used to initialize the CAN card and start the LabVIEW CAN driver. After executing the ExpCANStart function, the ExpCANin and ExpCANout function would be used to receive and send data. If users want to exit this program, the function ExpCANStop must be used to release the system resources of CAN card and LabVIEW CAN driver.

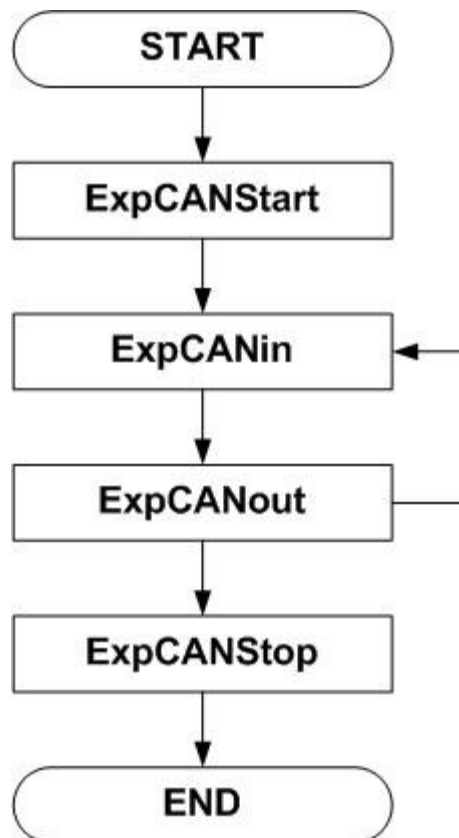


Figure 4.1 programming flowchart

4.3 Function Description

In order to make the descriptions more simply and clear, the attributes for the both input and output parameter functions are given as [input] and [output] respectively. If a parameter is only used in standard subVI but not in Express VI, the mark “[standard]” will be given.

4.3.1 ExpCANStart

- **Description:**

ExpCANStart VI activates the CAN card and applies all desired processes to LabVIEW. It will initialize all parameters of LabCANIcpdas.dll, and must be called once before using other functions.

- **Function VI:**

Standard VI: SubExpCANStart.vi

Express VI: ExpCANStart

- **Parameter:**

chProcessName: [input][standard] Input all process name that will be used latter. Each different process name need a comma mark to separate themselves.

Language: [input][standard] There are two kinds of languages, German and English, to select for error message. The input value 0 is German, 1 is English. The default value is 0.

error in: [input] The input parameter is an “Error In 3D” cluster of form panel. If this parameter is not used, the default parameter status is “ok”. If the status is “ok”, the function will start to initialize the LabVIEW CAN driver. And if the status is “error”, the function will not initialize the CAN driver.

error out: [output] The output parameter is an “Error Out 3D” cluster of form panel. If some error happens, the error state will be outputted.

- **Note:**

Even if this function is used in a while loop, the function is still executed only one time.

4.3.2 ExpCANStop

- **Description:**

This VI closes the CAN card and releases all the LabVIEW CAN driver resources. When users want to close their applications, the ExpCANStop function must be used firstly.

- **Function VI:**

Standard VI: SubExpCANStop.vi

Express VI: ExpCANStop

- **Parameter:**

SyncInput: [input] The type of this parameter is Boolean format. If the input is "True", the driver will be stopped and released.

Language: [input] There are two kinds of languages, German and English, to select for error message. The input value 0 is German, 1 is English. The default value is 0.

SyncOutput: [output] The type of this parameter is Boolean format. When the driver is stopped successfully, the state "True" will be output.

- **Note:**

If the ExpCANStop function is no used when the users program is closed, this may lead serious faults and uncertain system behavior of the network.

4.3.3 ExpCANin

- **Description:**

This VI receives all the CAN messages whose arbitration IDs are previously defined in receive process.

- **Function VI:**

Standard VI: SubExpCANin.vi

Express VI: ExpCANin

- **Parameter:**

chProcessName: [input][standard] Input all process name that will be used latter. Each two different processes name need a comma mark to separate themselves.

Language: [input][standard] There are two kinds of languages, German and English, to select for error message. The input value 0 is German, 1 is English. The default value is 0.

error in: [input] The input parameter is an “Error In 3D” cluster of form panel. If this parameter is not used, the default parameter status is “ok”. If the status is “ok”, the function will start to initialize the LabVIEW CAN driver. And if the status is “error”, the function will not initialize the CAN driver.

error out: [output] The output parameter is an “Error Out 3D” cluster of form panel. If some error happens, the error state will be outputted.

FifoPercent: [output] This parameter will output a integer to tell users how many record of CAN message store in the receive data buffer of LabVIEW CAN driver. When the parameter value is over 100, the buffer is overflow and some data will be lost.

MprValOut: [output] When CAN message is received, the data part of this CAN message will be passed by this parameter. The output data will match the data format setting of the process made by Configurator.

Ready: [output] When the ExpCANin module is no error and ready to receive data, the “Ready” parameter is “1”, and if not, the “Ready” is “0”.

- **Note:**

One ExpCANin function can include more than one receive process made by Configurator.

4.3.4 ExpCANout

- **Description:**

ExpCANout VI can send the CAN data via send process applied by ExpCANStart.

- **Function VI:**

Standard VI: SubExpCANout.vi

Express VI: ExpCANout

- **Parameter:**

chProcessName: [input][standard] Input all process name that will be used latter. Each two different processes name need a comma mark to separate themselves.

Language: [input][standard] There are two kinds of languages, German and English, to select for error message. The input value 0 is German, 1 is English. The default value is 0.

dValue: [input] This value is the data part of CAN message which will be send out. When give the value to this parameter, this value needs to match the data format setting of the process made by Configurator.

error in: [input] The input parameter is an “Error In 3D” cluster of form panel. If this parameter is not used, the default parameter status is “ok”. If the status is “ok”, the function will start to initialize the LabVIEW CAN driver. And if the status is “error”, the function will not initialize the CAN driver.

error out: [output] The output parameter is an “Error Out 3D” cluster of form panel. If some error happens, the error state will be outputted.

- **Note:**

Every ExpCANout function can only use one send process. For example, if users have three send processes, they need three ExpCANout functions to achieve. If the applied send process is an RTR send process, the dValue must be set to 5.

5 Demo Program for Window

This section displays the demo program contours and the operation procedure for the demo programs. After installing the LabVIEW CAN driver correctly, users can see the demo programs in the folder [DAQPro\PISO-CAN\LabCAN\Demo\LabVIEW7.1](#). Here, a give step-by-step demo procedure is given to demonstrate how to use the LabVIEW CAN driver in LabVIEW environment. In the following description, only one demo is displayed. If users want to check other demos, please refer to the demo folders in the demo path. Before using this demo, the CAN port 0 and CAN port 1 of PCI CAN card need to connect with each other.

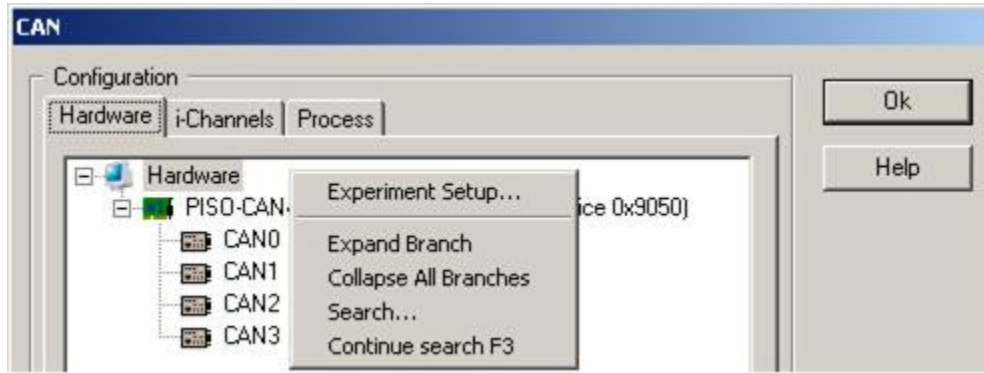
5.1 1xCAN_1xRTR Configurator Settings of Demo1


Open Configurator.exe in [DAQPro\PISO-CAN\LabCAN\Demo\Configurator\](#).

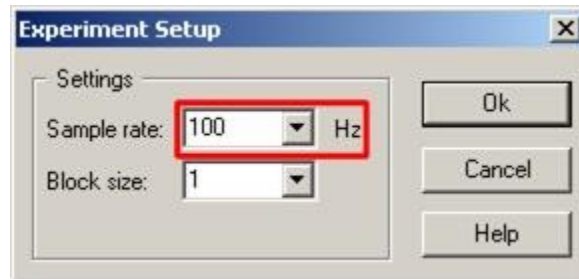




Hardware Setting

Click  icon to pup up the CAN configuration window.



Right click  Hardware icon. Then select experiment setting below. Here, the sample rate 100 Hz and block size 1 are used.

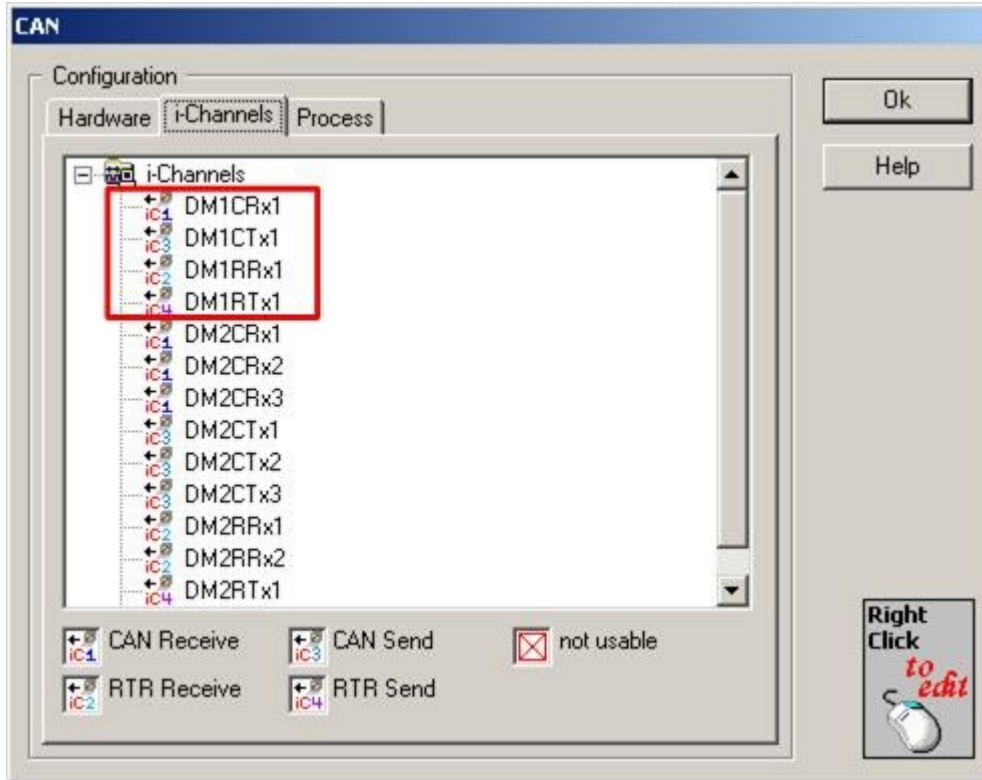


Double click  CAN0 and  CAN1 to set the baud rate of CAN0 and CAN1. Use the 500K bit predefined baud for these two CAN ports.



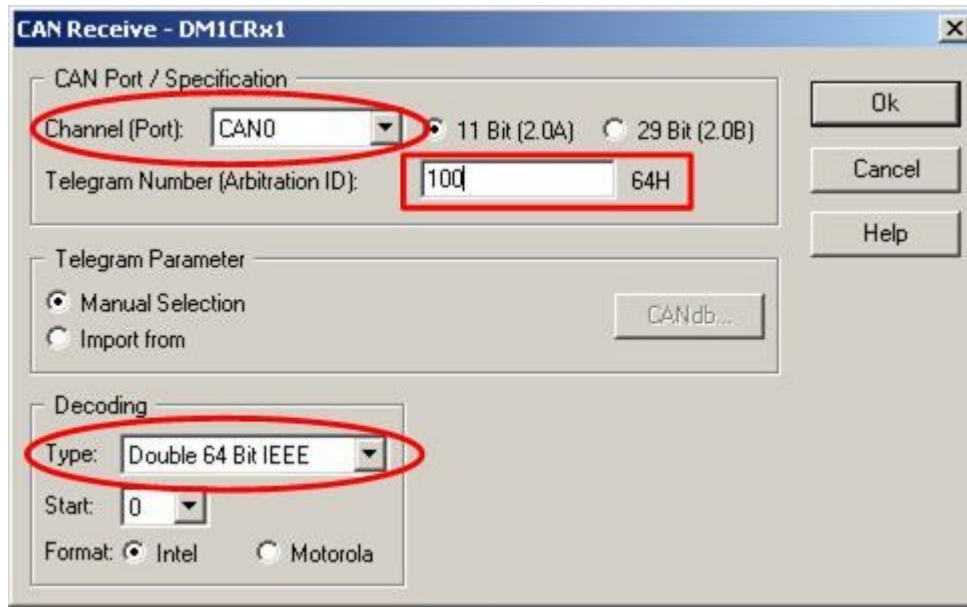
i-Channels Setting

When users execute the Configurator, there are many channels defined in the i-Channel window. Only the four channels, DM1CRx1, DM1CTx1, DM1RRx1, and DM1RTx1 are for demo1. All the other channels are for demo2.

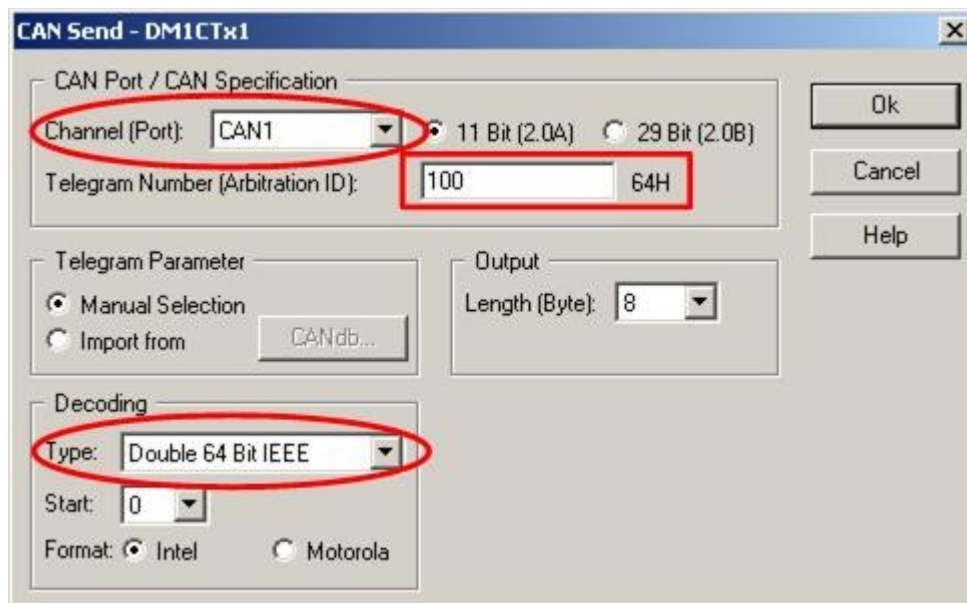


In this demo, we just use 1 CAN receive channel, 1 CAN send channel, 1 RTR receive channel, and 1 RTR send channel. Therefore, the four channels, “DM1CRx1”, “DM1CTx1”, “DM1RRx1”, and “DM1RTx1” will be introduced.

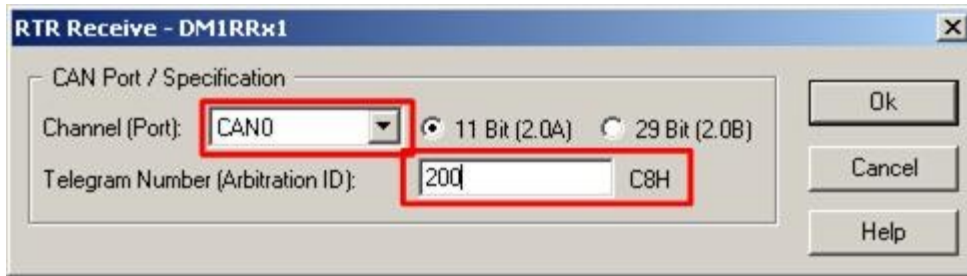
The DM1CRx1 channel setting window is shown below. The “Channel” parameter is CAN 0, and the CAN specification is 2.0A. The attribution ID is 100. The telegram parameter is set as Manual Selection. The decoding type is double 64 bit, and the start byte is 0. The decoding format is Intel format.



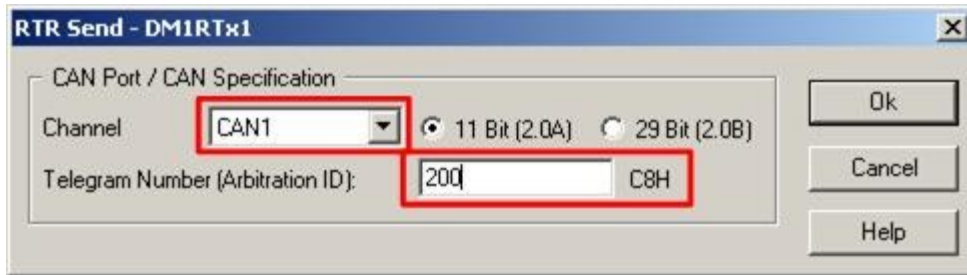
The DM1CTx1 channel parameter setting is similar with the DM1CRx1, expect the Channel and output length parameters. The DM1CTx1 use CAN port 1 to transmit data. Therefore, the Channel filed is set to CAN1. The length of output data is 8 bytes.



The DM1RRx1 channel settings are as follows. The channel is CAN 0, and the CAN specification is 2.0A. The arbitration ID is 200.

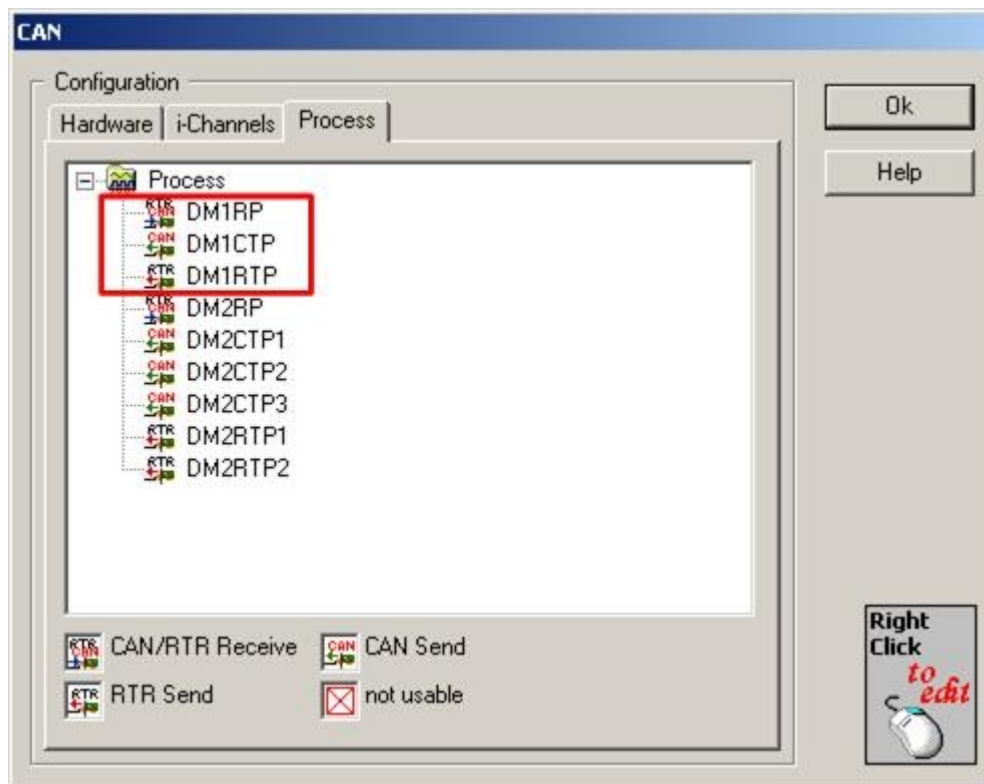


The DM1RTx1 channel setting is as follows. The CAN channel is CAN 1, and the CAN specification is 2.0A. The arbitration ID is 200.

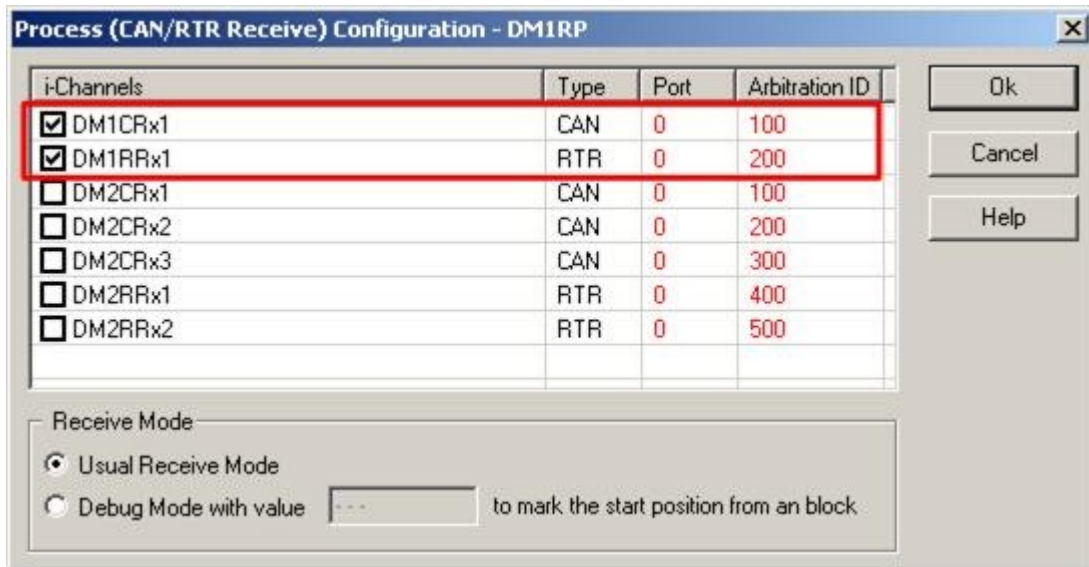


Process Setting

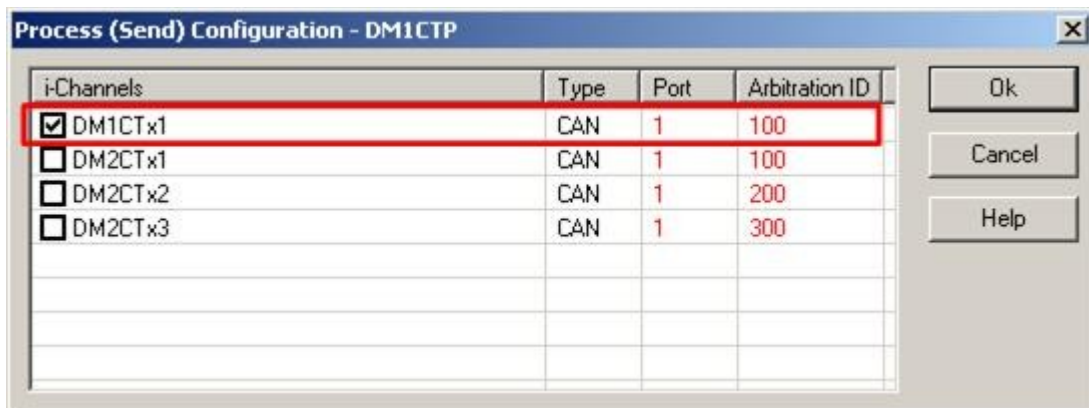
There are 9 processes in the Process window. But only the three processes, DM1RP (CAN / RTR receive process), DM1CTP (CAN send process), and DM1RTP (RTR send process) have been use for this demo. So, we only check DM1RP, DM1CTP, and DM1RTP three processes at this part.



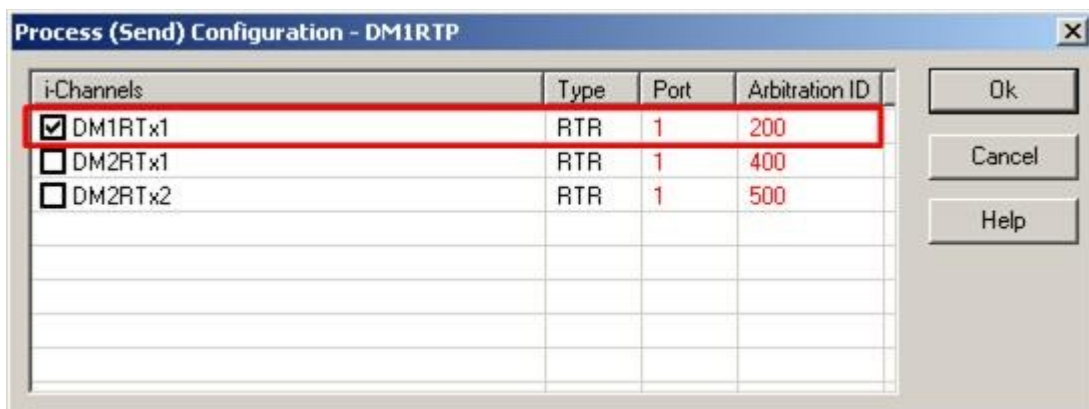
Double click DM1RP to check the setting of DM1RP process as follows. Check the “DM1CRx1” and “DM1RRx1” check box. It means that there are only two channels, “DM1CRx1” and DM1RRx1”, applied in the DM1RP process. The receive mode is set to “Usual Receive Mode” for normal application.



The setting of the CAN send process, DM1CTP, is displayed below. The DM1CTP process uses the DM1CTx1 channel.



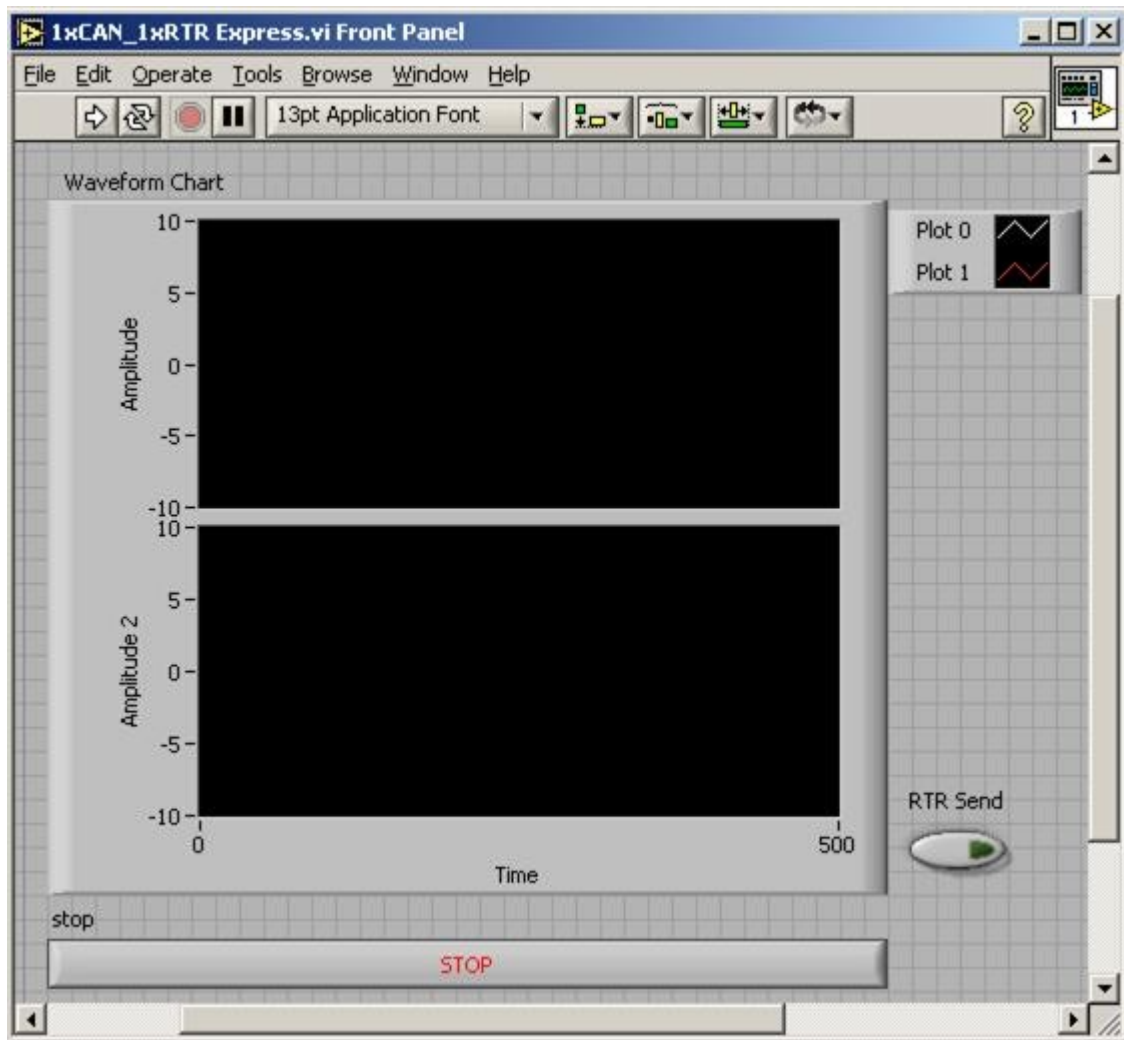
The RTR Send process setting of DM1RTP is shown below. The DM1RTP process uses the DM1RTx1 channel.



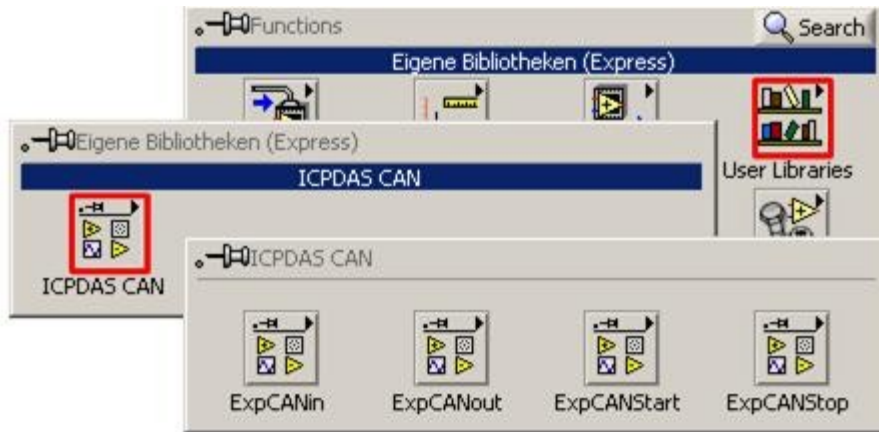
5.2 1xCAN_1xRTR LabVIEW Program of Demo1

After finishing the Configurator settings, Run the Demo1.vi demo in the demo folder. The front panel and the block diagram of the demo1 are shown below.

Adds a Waveform Chart VI, a Stop button VI, and a Push button VI in the front panel window. The Waveform Chart is set to two plots. The button behavior of the Push button is set to “Latch when pressed” type, and its label changes to “RTR Send”.



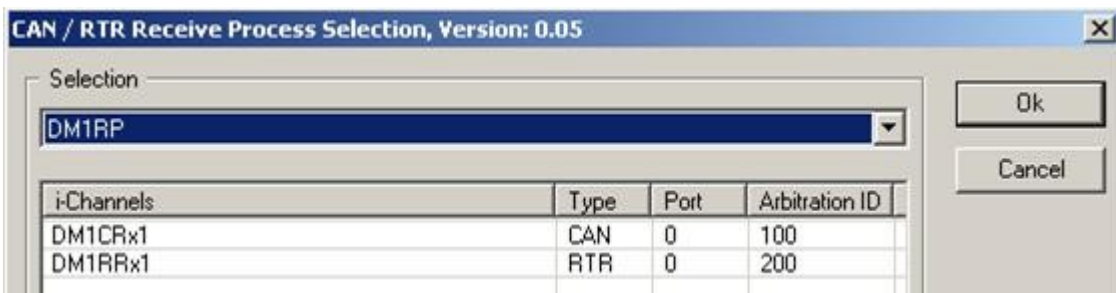
In the block diagram window, right click to import a VI. Select User Library → ICPDAS CAN, and then there are four VIs to be able to use.



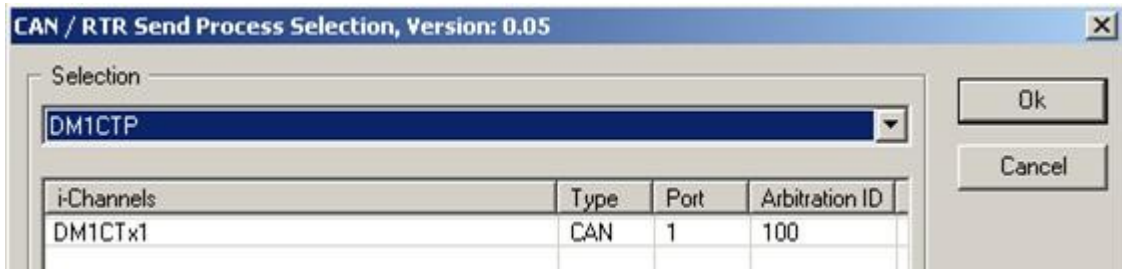
Put one ExpCANStart, one ExpCANStop, one ExpCANIn, and two ExpCANOut VI on the block diagram window. And the settings of these four VI are shown below (ExpCANStop VI does not need to configure).



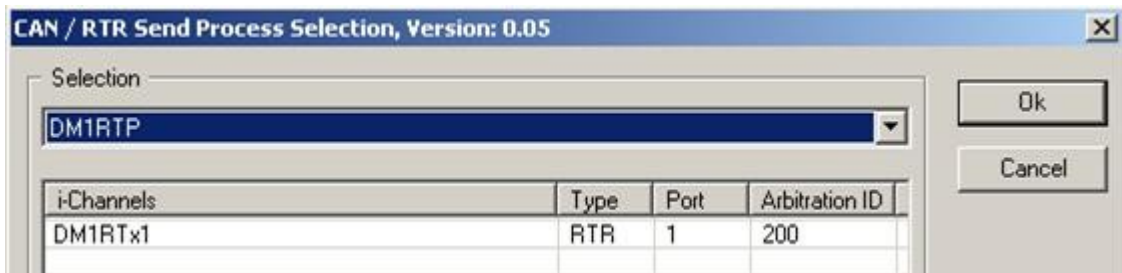
ExpCANStart parameter setting



ExpCANIn parameter setting

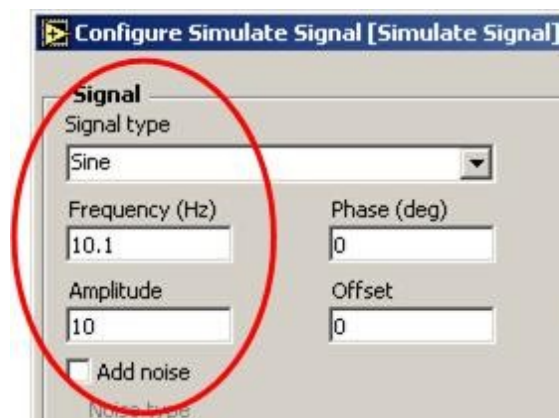


ExpCANout1 parameter setting



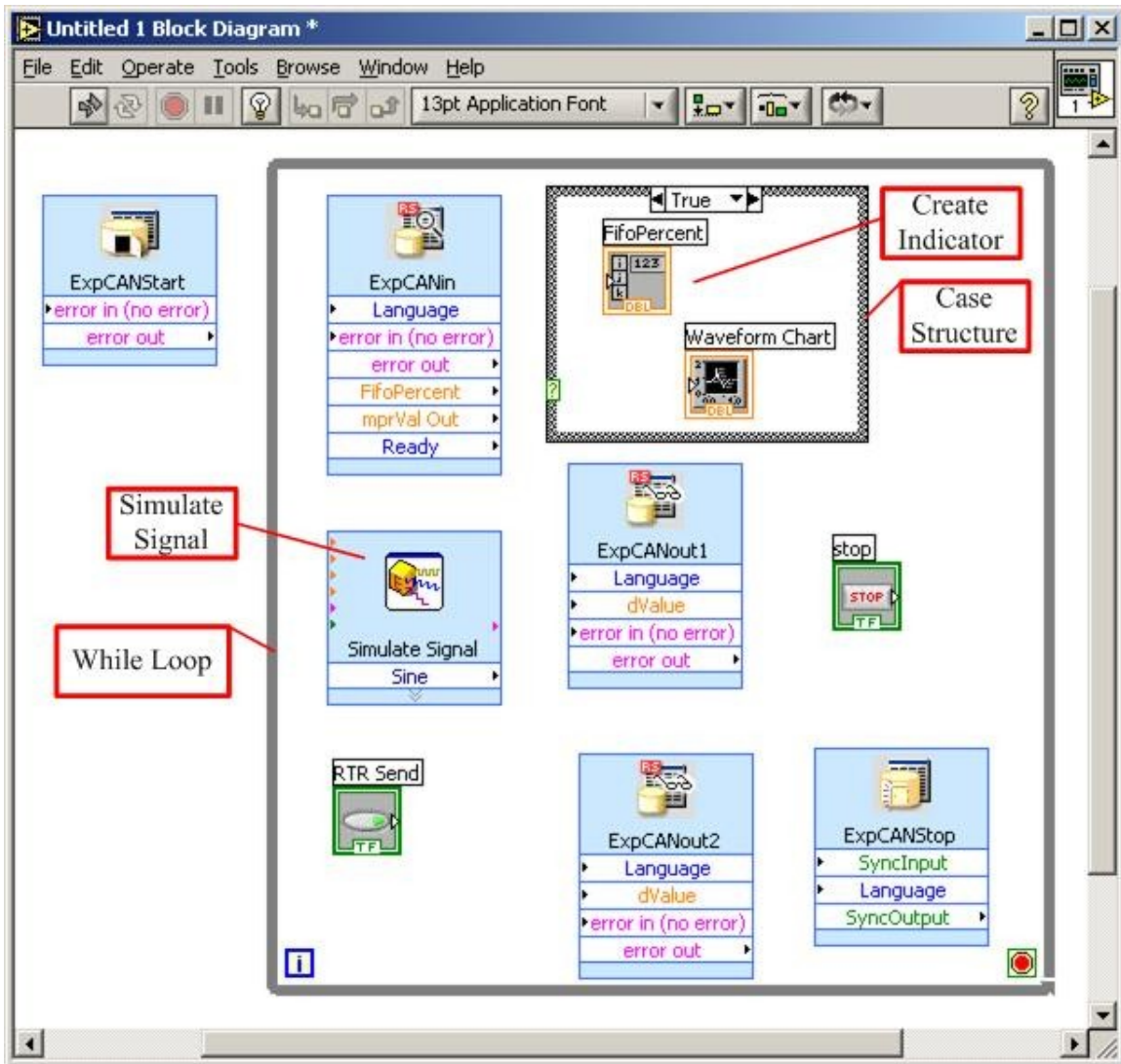
ExpCANout2 parameter setting

Add a While Loop VI, a Case Structure VI, a Simulate Signal VI, and an indicator VI with FifoPercent type. The function of Case Structure is that when the ready parameter of ExpCANin VI returns 1, the Waveform Chart will start to show the data wave. The Simulate Signal provides a sequential sine wave to ExpCANout1 VI to output CAN message into the CAN network. The Simulate Signal setting is shown below.

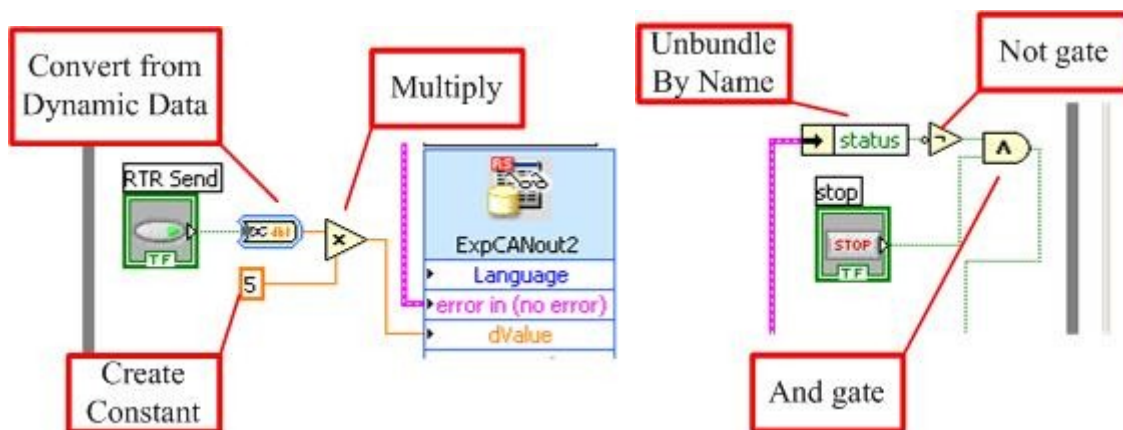


Simulate Signal setting

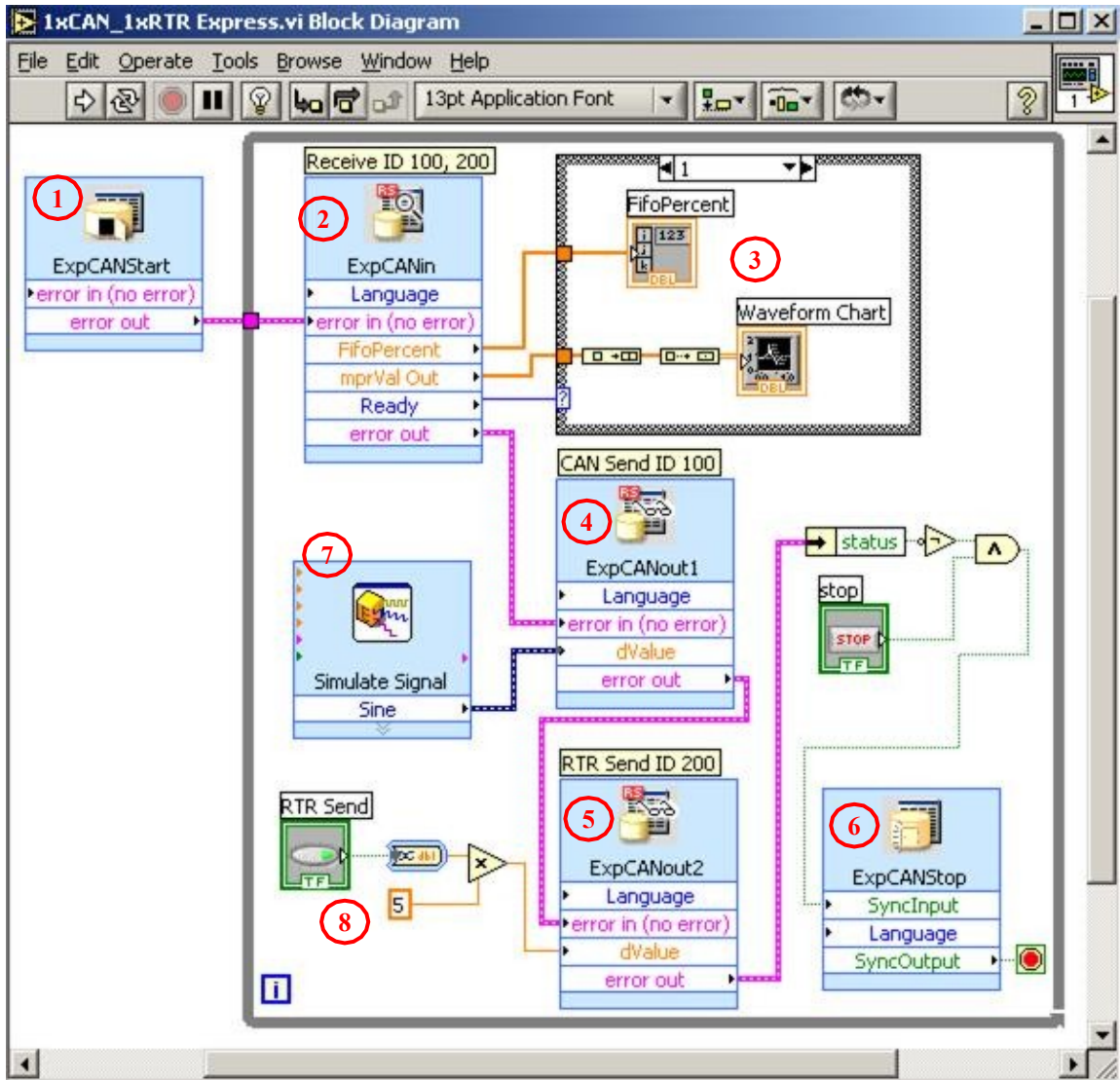
After finishing the steps above, the block diagram looks like this figure.



Add some logic VI below and connect them with proper VI respectively.

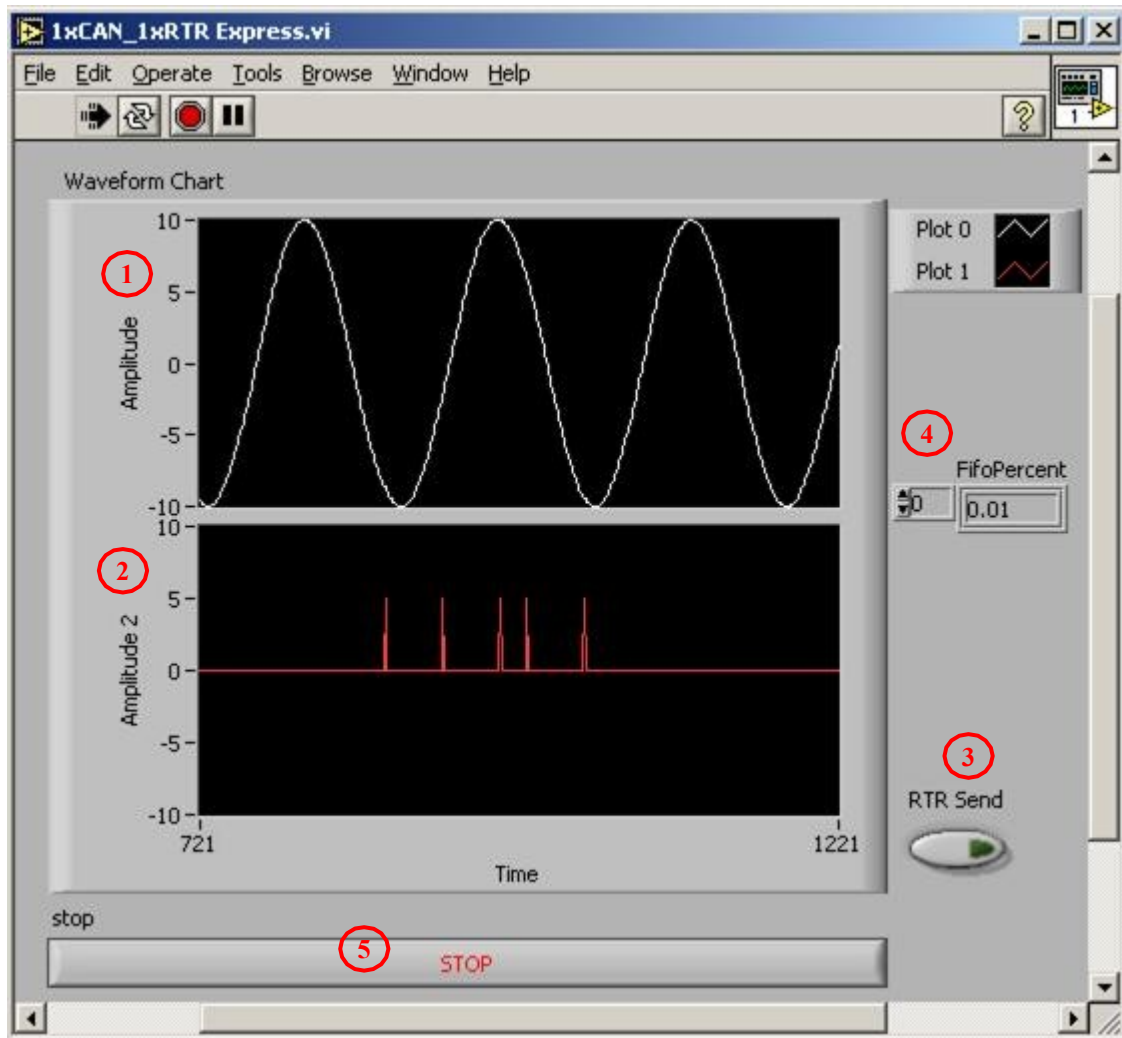


When the program is done, the block diagram is display as follows.



- ① ExpCANStart : Include DM1CRP, DM1CTP, and DM1RTP processes.
- ② ExpCANin : Use the process DM1CRP (CAN/RTR Receive Process).
- ③ Waveform Chart can show the data wave and the FifoPercent can show the numbers of data stored in the driver FIFO.
- ④ ExpCANout : Use the process DM1CTP (CAN Send Process).
- ⑤ ExpCANout : Use the process DM1RTP (RTR Send Process).
- ⑥ ExpCANStop : Click stop button to execute stop function of ExpCANStop.
- ⑦ Input Signal : Input sine wave to DM1CTP process
- ⑧ Input value 5 to DM1RTP RTR process

The demo front panel is display as follows.



- ① The waveform shows a sine wave form the received data with CAN ID 100.
- ② The waveform shows a pulse with value 5 form the data with CAN ID 200.
- ③ Click this button to send an RTR data with CAN ID 200.
- ④ Receive buffer, can show the numbers of data stored in the driver FIFO.
- ⑤ Stop button to stop the VI.

When CAN port 1 send the sine wave data with double type by using message ID100, the CAN port 0 can receive these data. Users can check it in the upper part of the waveform chart. If users click the RTR send button, CAN port 1 will send a RTR message with message ID 200. Then, the CAN port 0 will receive this message. Users can also check it in the lower part of the waveform chart.