CAN-Logger series
User Manual

Version 1.0.1, Dec. 2015

Service and usage information for

CAN- Logger100 / CAN-Logger200

Warranty

All products manufactured by ICP DAS are under warranty regarding defective materials for a period of one year, beginning from the date of delivery to the original purchaser.

Warning

ICP DAS assumes no liability for any damage resulting from 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, not for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademark

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

Contact us

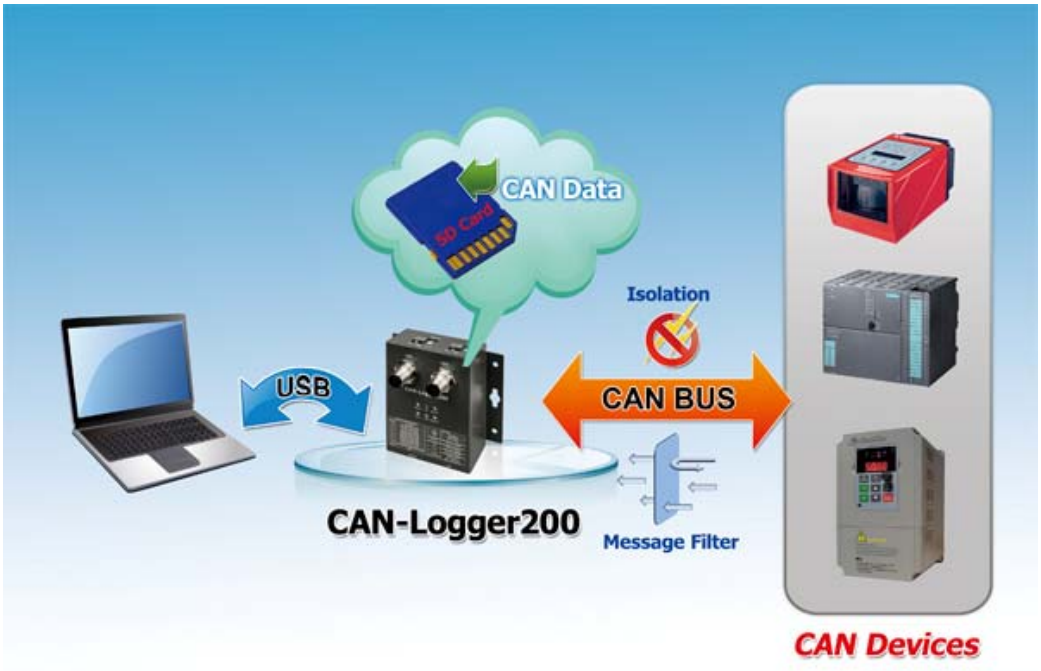
If you have any problem, please feel free to contact us.
You can count on us for quick response.

Email: service@icpdas.com

Table of Contents

1. Introduction

The CAN-Logger series devices (CAN-Logger100 / CAN-Logger200) are high-performance intelligent CAN Bus data logger device with one/two CAN port that can help to make data collection and to process on a CAN Bus network easier and quicker. The powerful CPU of the CAN-Logger devices provide the accurately time-stamp for each CAN message and supports storage media like SDHC type flash for saving these CAN messages that is useful to analysis and diagnostic the CAN Bus network. In order to enhance the portability of the CAN-Logger200, this module is powered by the USB interface or M12 connectors of CAN Bus interface.



1.1. Specifications

Model Name	CAN-Logger100	CAN-Logger200
CAN Interface		
Transceiver	NXP TJA1042	
Channel Number	1	2
Connector	5-Pin male M12 x 1 (Pin 1: F.G., Pin 2: +Vs, Pin 3: GND, Pin 4: CAN_H Pin 5: CAN_L)	5-Pin male M12 x 2 (Pin 1: F.G., Pin 2: +Vs, Pin 3: GND, Pin 4: CAN_H Pin 5: CAN_L)
Transmission Speed (bps)	10 k, 20 k, 50 k, 100 k, 125 k, 250 k, 500 k, 800 k, 1 M and user-defined baud rate	
Terminator Resistor	DIP switch for the 120 Ω terminator resistor	
Isolation	3000 VDC for DC-to-DC, 2500 Vrms for photocoupler	
Specification	ISO-11898-2, CAN 2.0A and CAN 2.0B	
CAN Filter Configuration	Utility tool	
USB Interface		
Connector	USB Type B x 1	
Compatibility	USB 2.0 High Speed	
Max. Data flow	Transmit: 4000 fps ; Receive: 1000 fps	
Software Driver	Windows 2K/XP/7/8	
Data Logger Capability		
Storage Media	SDHC type flash – support 4 to 32 GB (Class 10 would be recommended)	
Recording Format	Binary	
Time Stamp Resolution	10 us	
Configuration	Utility tool	
Trigger	Log continuously	
Data Logger	Maximum message rate, receive: 15000 msgs/s	
LED		
Round LED	Power, MS, SD, CAN1, CAN2, CAN_ST LEDs	Power, MS, SD, CAN_Rx, CAN_Tx, CAN_ST LEDs
Power		
Power supply	USB power or CAN Bus power (Unregulated +10 ~ +30 V_{DC}) delivery	
Protection	Power reverse polarity protection, Over-voltage brown-out protection	

Power Consumption	0.1 @ $24V_{DC}$
Mechanism	
Installation	DIN-Rail
Casing	Metal
Dimensions	102.0 mm x 102.0 mm x 44.0 mm (W x L x H)
Environment	
Operating Temp.	-25 ~ 75 ℃
Storage Temp.	-30 ~ 80 ℃
Humidity	10 ~ 90% RH, non-condensing

1.2. Features

- Compatible with CAN specification 2.0 parts A and B
- 2500 Vrms photocoupler isolation on the CAN side
- Full compatible with the ISO 11898-2 standard
- Supports CAN Bus acceptance filter configuration
- CAN messages are time-stamped with 10 microseconds resolution
- Programmable CAN Bus baud rate from 10 kbps ~ 1Mbps
- USB 2.0 High Speed Compatibility
- Supports 4 to 32 GB SDHC type flash for saving CAN messages
- Built-in jumper for the 120 Ω terminal resistor of the CAN side
- Power from CAN bus or from the USB side.
- Built-in real time clock with battery backup
- Provides a configuration utility that can be used to transmit/receive CAN messages

2. Technical data

2.1. Block Diagram

The following figure is the block diagram illustrating the functions of the CAN-Logger series.

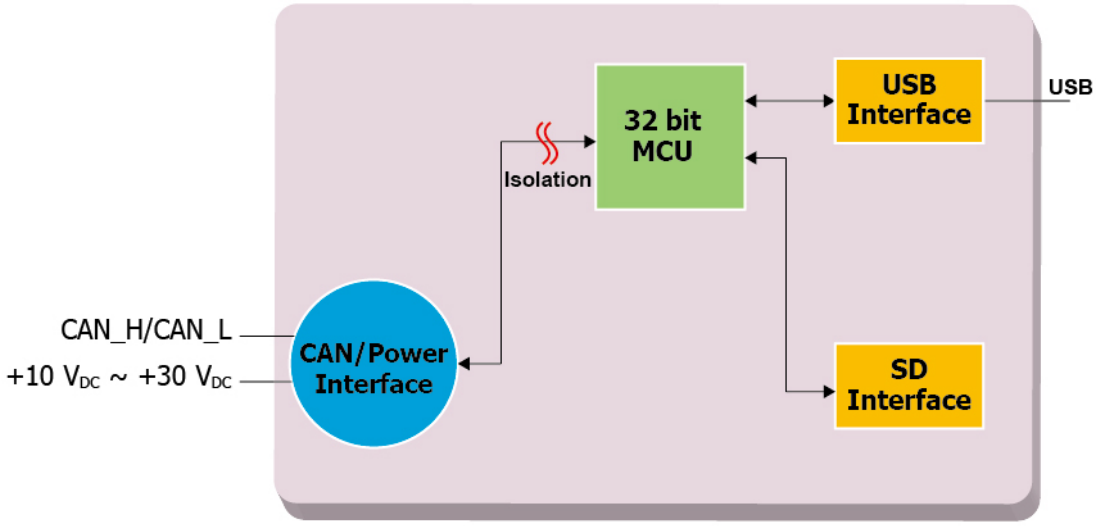


Figure 2-1 Block Diagram of CAN-Logger100

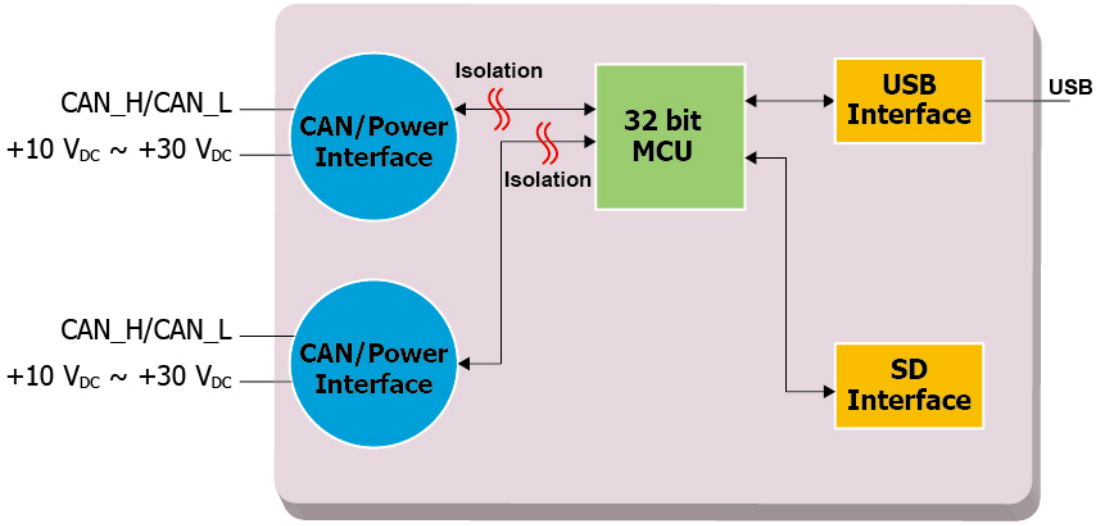


Figure 2-2 Block Diagram of CAN-Logger200

2.2. Appearance

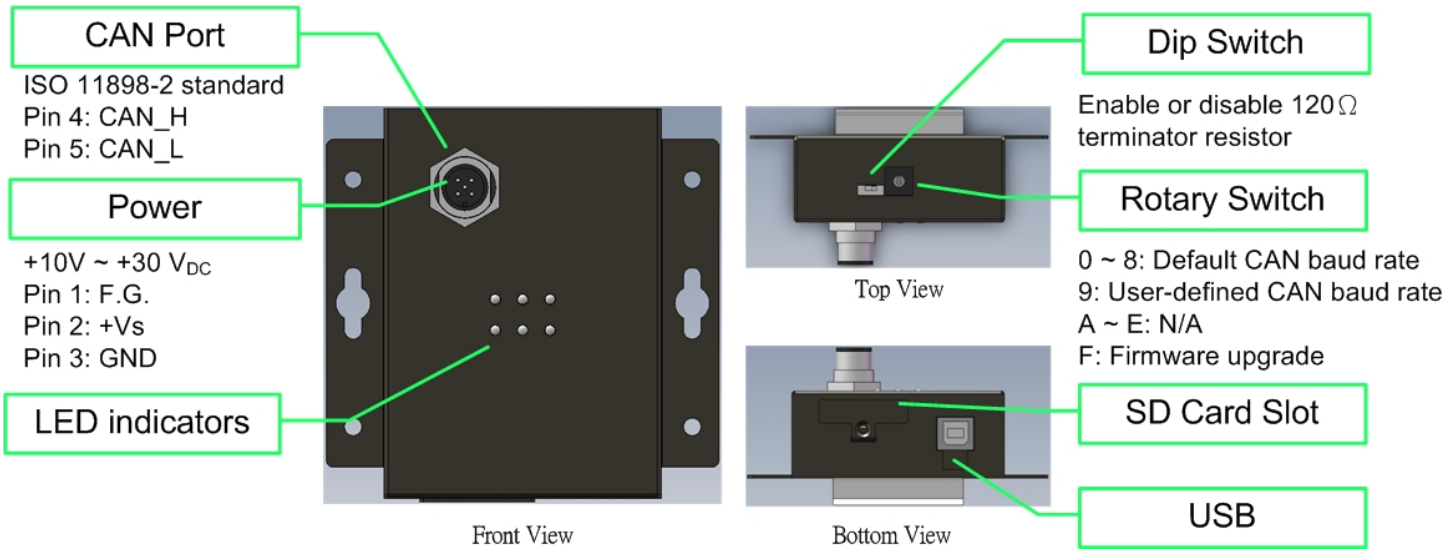


Figure 2-3 Appearance of CAN-Loger100

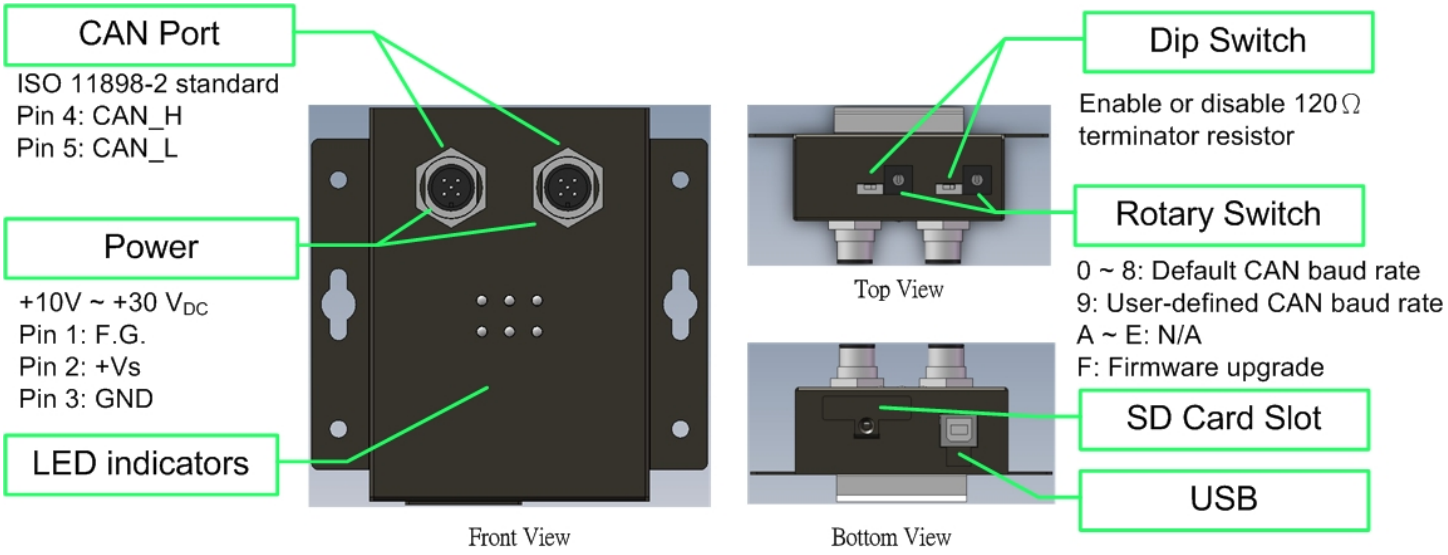


Figure 2-4 Appearance of CAN-Logger200

2.3. Pin Assignment

The pin assignments of 5-pin male M12 CAN connector of CAN-Logger series is shown in the following tables.

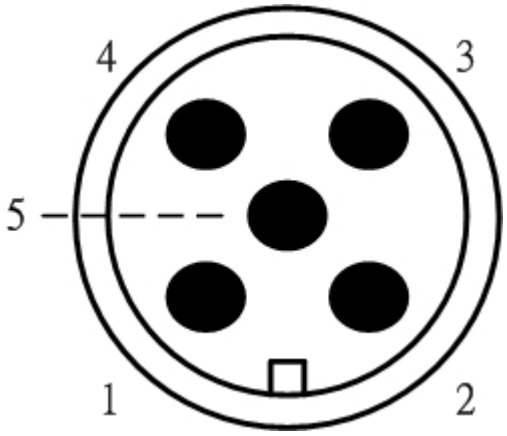


Table 2-1 Pin Assignment

Pin No	Name	Description
1	F.G.	Frame Ground.
2	+Vs	Voltage Source Input. +10V_{DC} ~ +30V_{DC}.
3	GND	Power Ground.
4	CAN_H	CAN_High, signal line of CAN port.
5	CAN_L	CAN_Low, signal line of CAN port.

Electronic circuits are always influenced by different levels of Electro-Static Discharge (ESD), which become worse in a continental climate area. F.G. provides a path for conducting the ESD to the earth ground. Therefore, connecting the F.G correctly can enhance the capability of the ESD protection and improve the module's reliability.

Wiring of F.G. is not necessary; users can modify the configuration of wiring according to real applications.

2.4. Rotary Switch

When users would like to update the module's firmware or change the CAN baud rate of the CAN-Logger series, use the rotary switch on the top side to achieve this purpose.

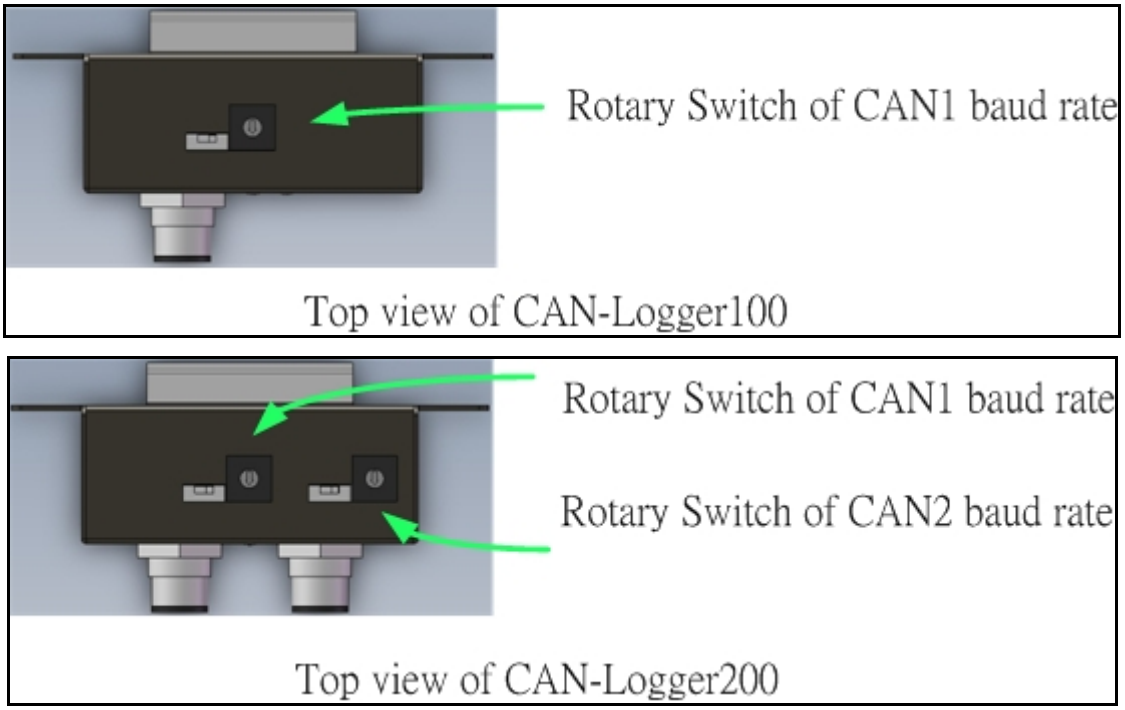


Figure 2-5 Location of CAN-Logger100/CAN-Logger200 Rotary Switch

There are 16 sections on the rotary switch. They are described on the following table.

Table 2-2 Description of Rotary Switch

Switch Value	Description
0	Set CAN Bus baud rate to 10 kbps
1	Set CAN Bus baud rate to 20 kbps
2	Set CAN Bus baud rate to 50 kbps
3	Set CAN Bus baud rate to 100 kbps
4	Set CAN Bus baud rate to 125 kbps
5	Set CAN Bus baud rate to 250 kbps
6	Set CAN Bus baud rate to 500 kbps
7	Set CAN Bus baud rate to 800 kbps
8	Set CAN Bus baud rate to 1 Mbps
9	Set CAN Bus baud rate to user-defined baud rate which is configured by CAN-Logger utility.
A~E	N/A
F	Set CAN-Logger device into firmware upgrade mode.

2.5. LED Indicator

There are 6 LEDs on the CAN-Logger series. One for power indication, one for hardware status indication, one for SD status indication and three for CAN Bus indication. The LED assignment and description are shown as follows.

Figure 2-6 LED Assignment of CAN-Logger100

Figure 2-7 LED Assignment of CAN-Logger200

Table 2-3 LED Description

LED Name	Color	Description
Power	Red	Power status of 5-pin M12 and USB port
MS	Red	Module status. OFF: no error ON: hardware malfunction
SD_ST	Red	SD card status. OFF: no error ON: access SD card error
CAN_ST	Red	CAN Bus status. OFF: no error ON: CAN Bus Off Flash: CAN Bus error
CAN_Tx	Green	OFF: no messages transmitted Flash: Transmit messages on CAN1 port
CAN_Rx	Green	OFF: no messages be received Flash: Receive messages on CAN1 port
CAN1	Green	OFF: no messages on CAN1 port Flash: Transmit/Receive messages on CAN1 port

CAN2	Green	OFF: no messages on CAN2 port Flash: Transmit/Receive messages on CAN2 port

NOTE:

In “Firmware Upgrade Mode”:

For CAN-Logger100:

These LEDs of “Power”, “MS”, “SD_ST”, “CAN_ST”, “CAN_Rx”, “CAN_Tx” would flash in the clockwise direction.

For CAN-Logger200:

These LEDs of “Power”, “MS”, “SD_ST”, “CAN_ST”, “CAN2”, “CAN1” would flash in the clockwise direction.

2.6. Terminator Resistor Setup

In order to minimize the reflection effects on the CAN Bus line, the CAN Bus line has to be terminated at both ends by two terminator resistors as in the following figure. According to the ISO 11898-2 spec, each terminator resistor is 120Ω (or between 108Ω~132Ω). The bus topology and the positions of these terminator resistors are shown as following figure.

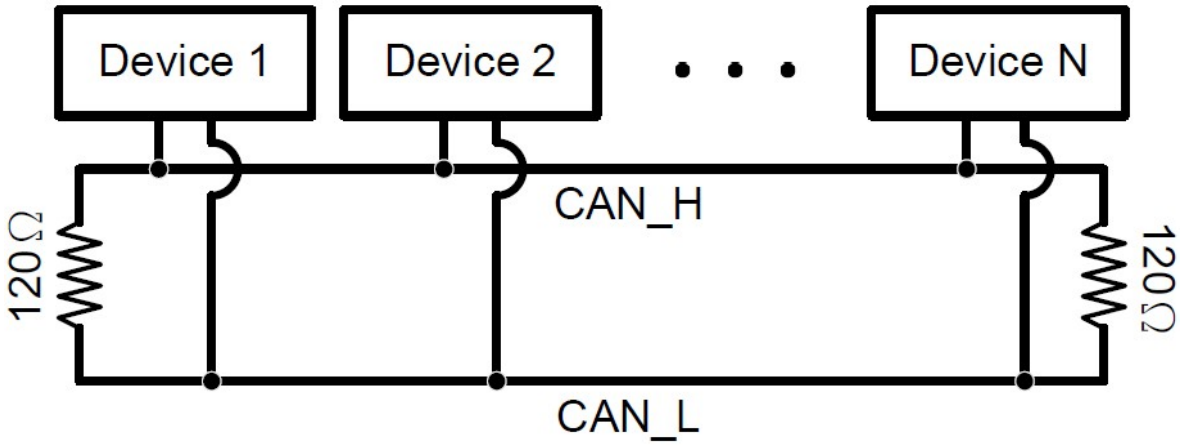


Figure 2.8 CAN Bus network topology

Each CAN-Logger series includes one build-in 120Ω terminator resistor, users can decide if it is enabled or not. The DIP switch for terminator resistor is under the top side.

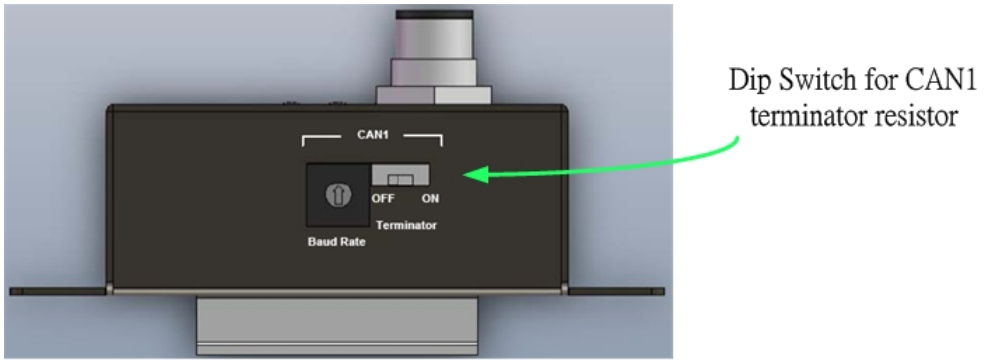


Figure 2-9 Location of Terminator Resistor DIP Switch of CAN-Logger100

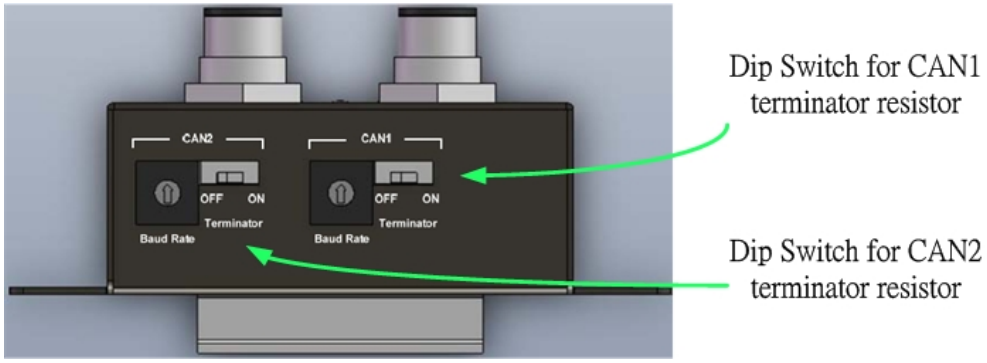


Figure 2-10 Location of Terminator Resistor DIP Switch of CAN-Logger200

The following DIP switch statuses present the condition if the terminator resistor is active (default) or inactive.

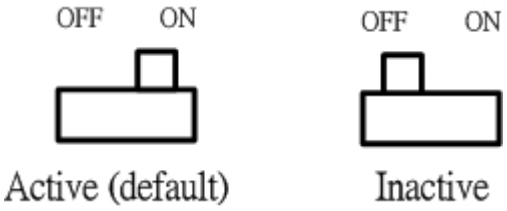


Figure 2-11 Adjustment of Terminal Resistance

Generally, if your application is as follows, we recommend you to enable the terminator resistor.

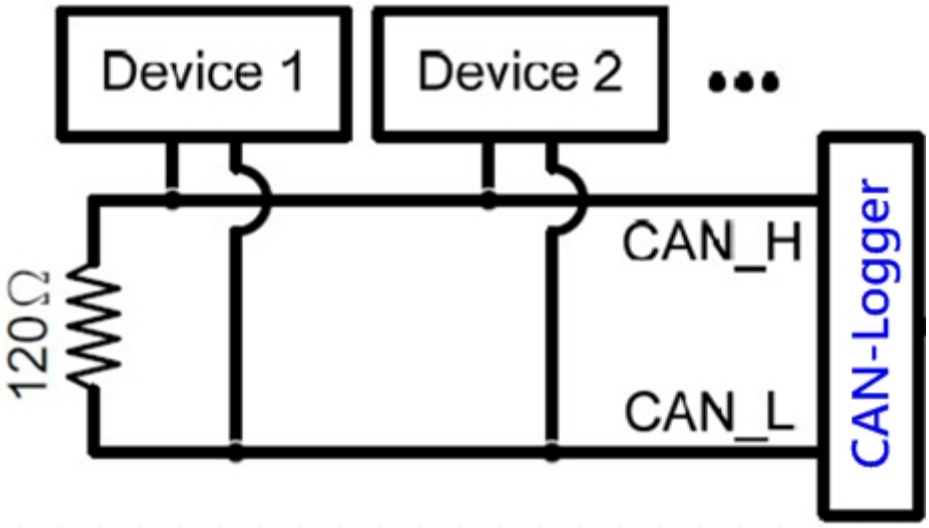


Figure 2-12 Application 1

If your application is like the structure as follows, the terminator resistor is not needed.

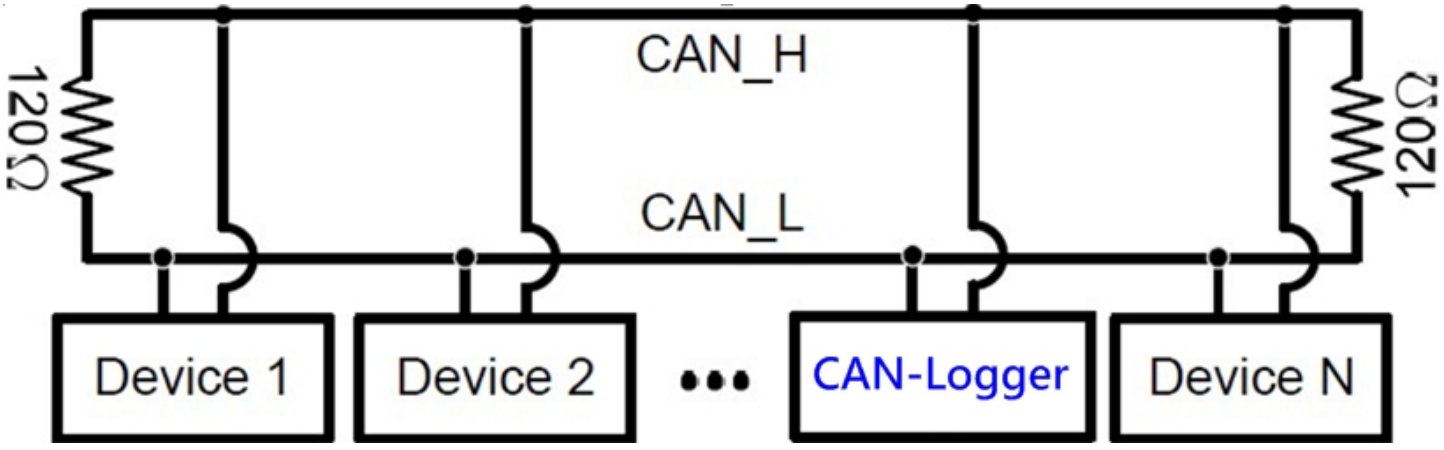
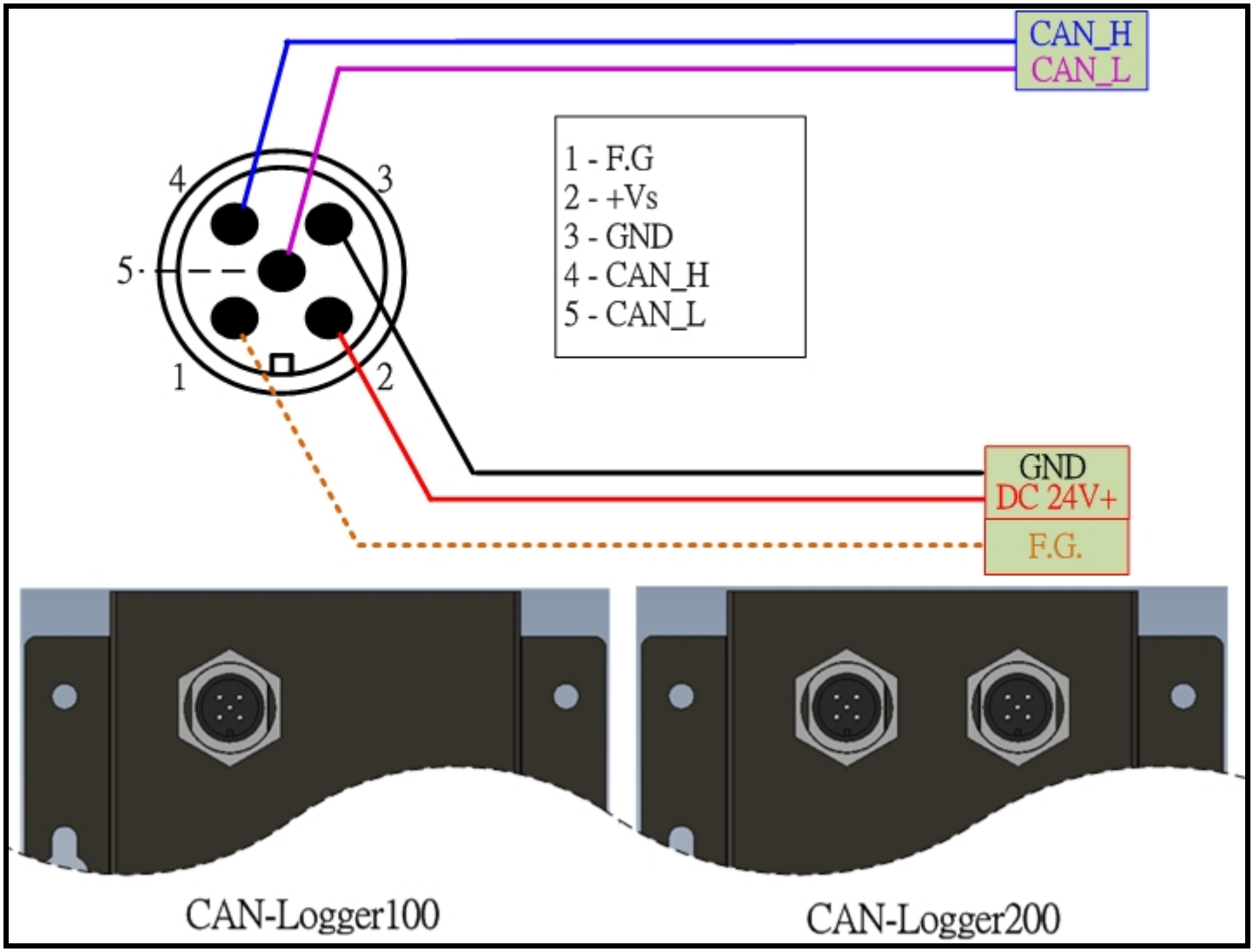


Figure 2-13 Application 1

2.7. Wire Connection

The wire connection of the CAN-Logger series is displayed below.



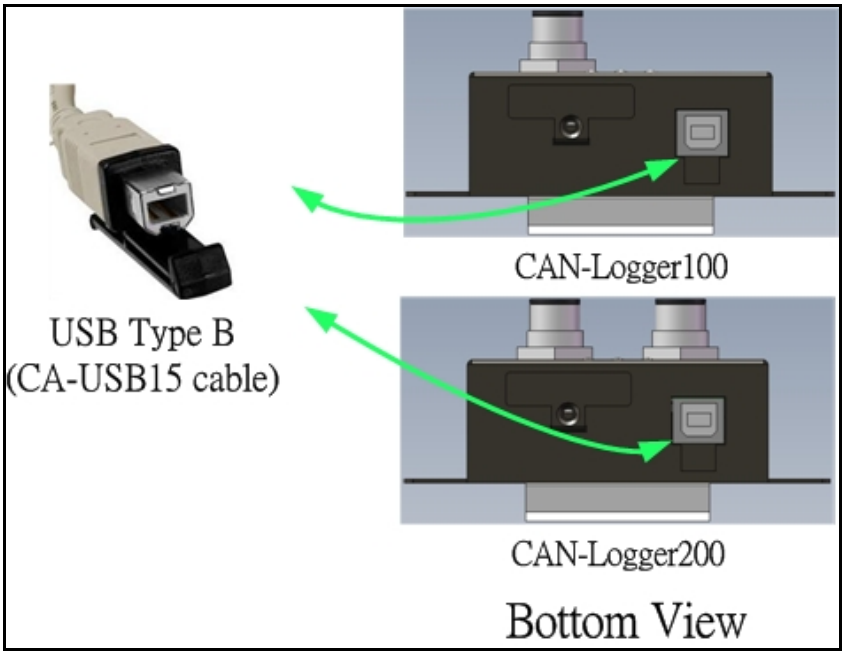


Figure 2-42 Wire Connection for CAN-Logger series

3. Network Deployment

3.1. Driving Capability

Before introducing the driving capability of the CAN-Logger series, some characteristics of copper cable must be assumed. The AC parameters are 120Ω impedance and ms/line delay, and the DC parameter follows the table show below.

Table 3-1 Recommended DC parameter for CAN Bus Line

Wire Cross-Section [mm^2]	Resistance [Ω/km]
~0.25 (AWG23)	< 90
~0.5 (AWG20)	< 50
~0.8 (AWG18)	< 33
~1.3 (AWG16)	< 20

Under the condition described above, users can refer to the following table to know the maximum node number in each segment following ISO 11898-2 and the maximum segment length when using different type of wire.

Table 3-2 Driving Capability

Wire Cross-Section [mm^2]	The maximum segment length [m] under the case of specific node number in this segment			
	16 Nodes	32 Nodes	64 Nodes	100 Nodes
~0.25 (AWG23)	< 220	< 200	< 170	< 150
~0.5 (AWG20)	< 390	< 360	< 310	< 270
~0.8 (AWG18)	< 590	< 550	< 470	< 410
~1.3 (AWG16)	< 980	< 900	< 780	< 670

4. Software Utility

When users want to use user-defined CAN baud rate, CAN message filter and diagnostic function on the CAN-Logger series, the CAN-Logger Utility tool may be needed.

4.1. Install the CAN-Logger Utility

Step 1: Get the CAN-Logger Utility

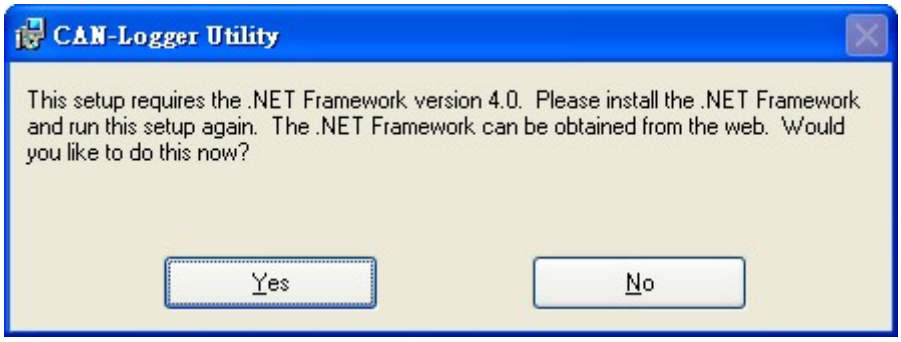
The software is located at:

Fieldbus_CD:\CAN\CAN_Logger\Software\Utility

http://www.icpdas.com/root/product/solutions/industrial_communication/fieldbus/can_bus/specific_device/can-logger.html

Step 2: Install .NET Framework 4 component

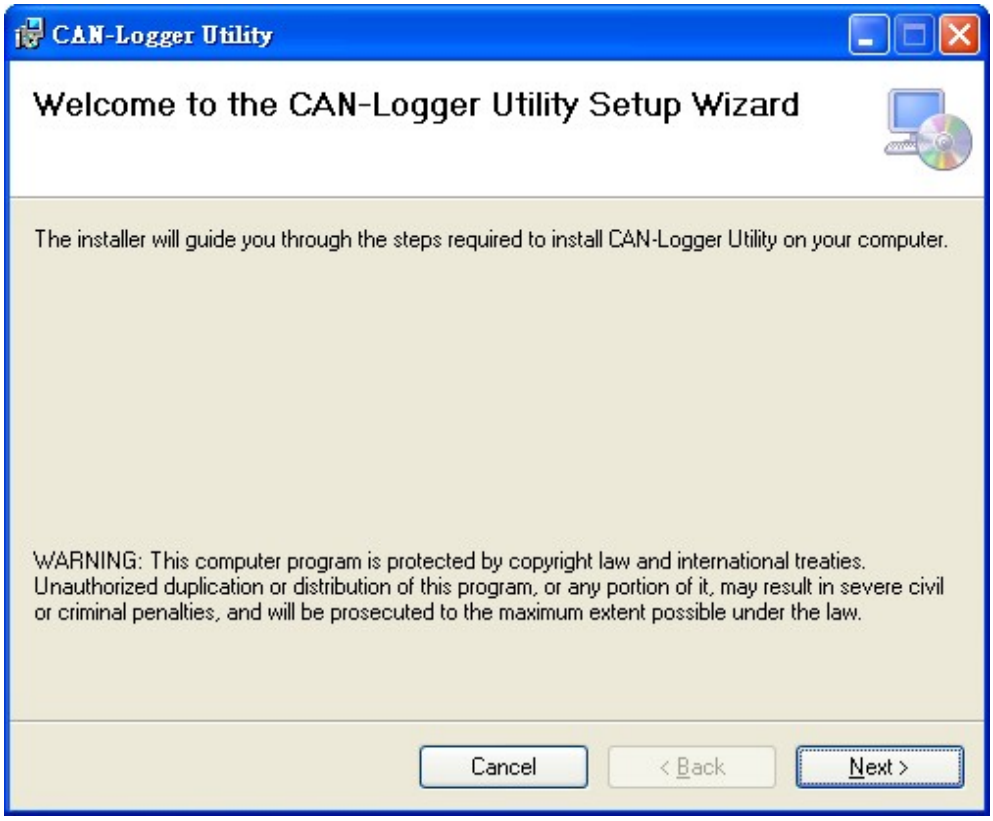
The CAN-Logger Utility tool requires the .NET Framework 4 components. After executing the “Setup.exe” file, it will start to install .NET Framework 4 components from the web site.



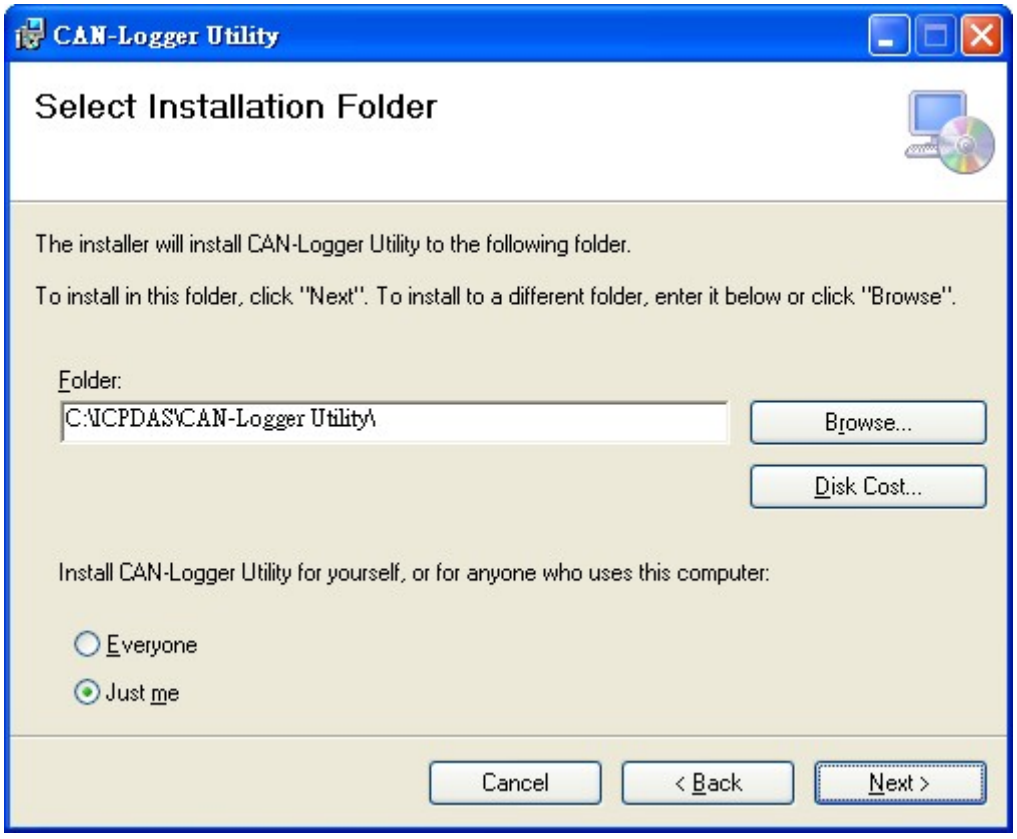
Step 3: Install Utility tool

After installing the .Net Framework components, the software will continue to install the Utility tool.

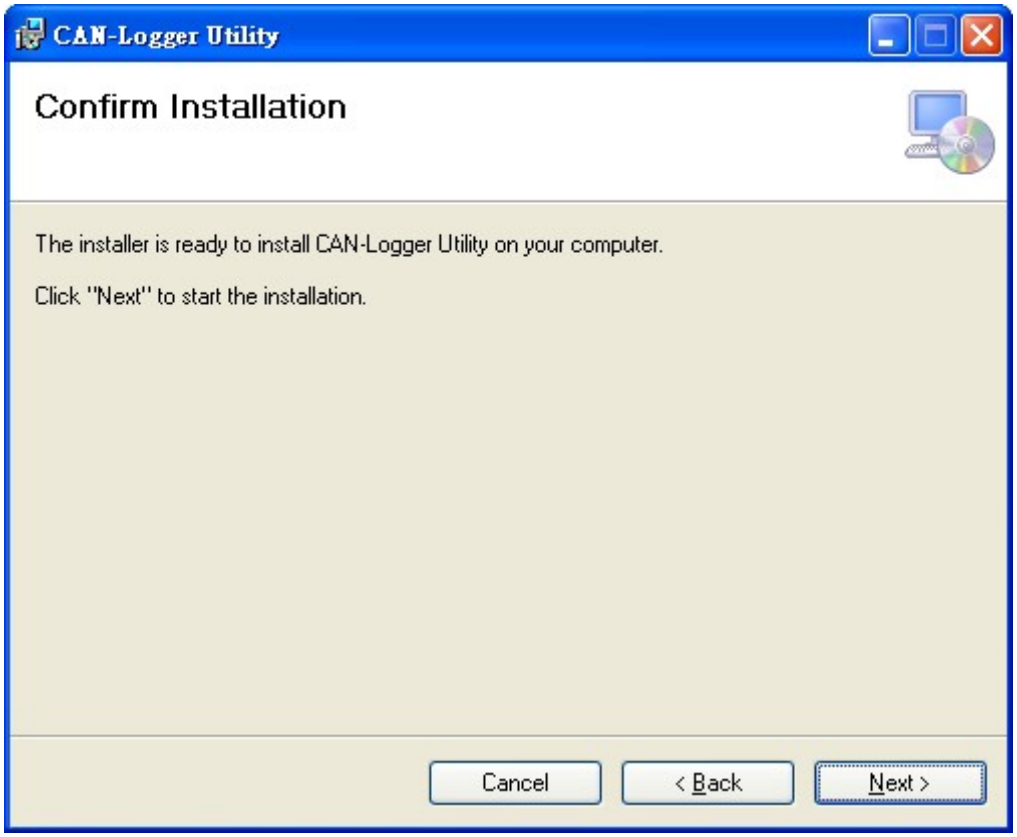
1. Click the “Next” button to continue.



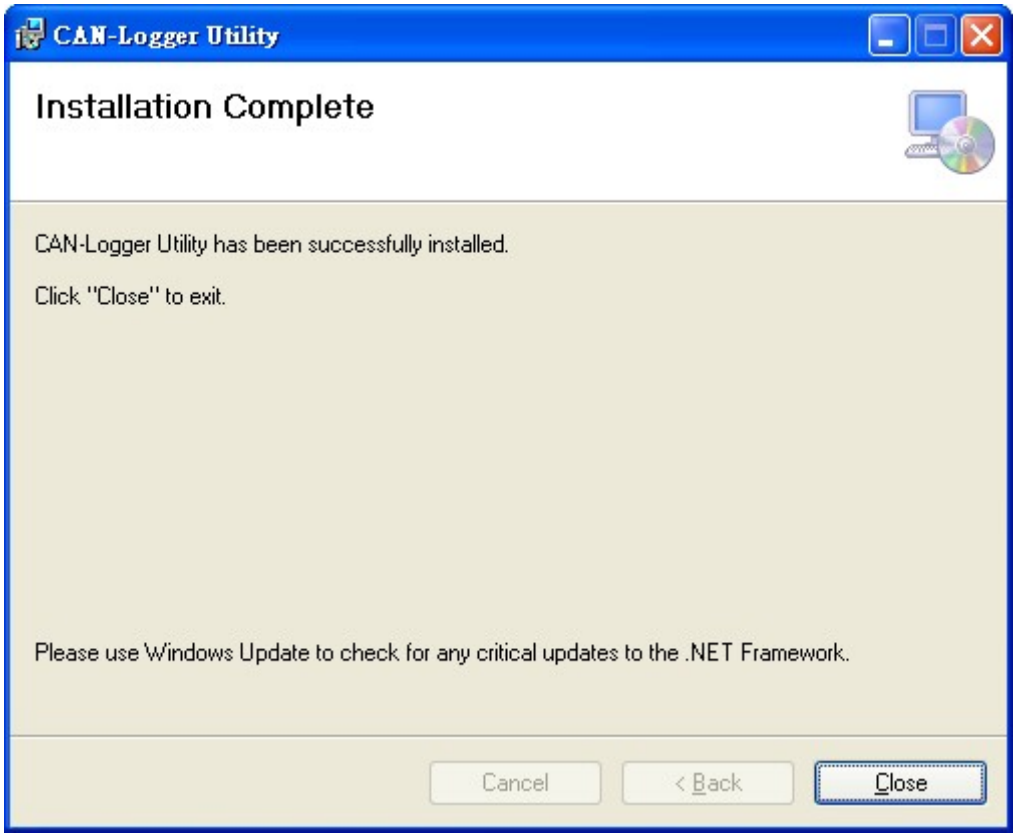
2. Select the installation path of the CAN-Logger Utility and click the “Next” button.



3. Confirm the installation. Click the “Next” button to start the installation



4. Installation complete. Click the “Close” button to exit



4.2. Setting up the CAN-Logger series

After installing the utility tool, please follow the following steps to set up the communication between the Utility and the CAN-Logger series.

Step 1: Connect the PC available USB port with the USB port of the CAN-Logger device. Users can find the communication cable (CA-USB15) in the product box.

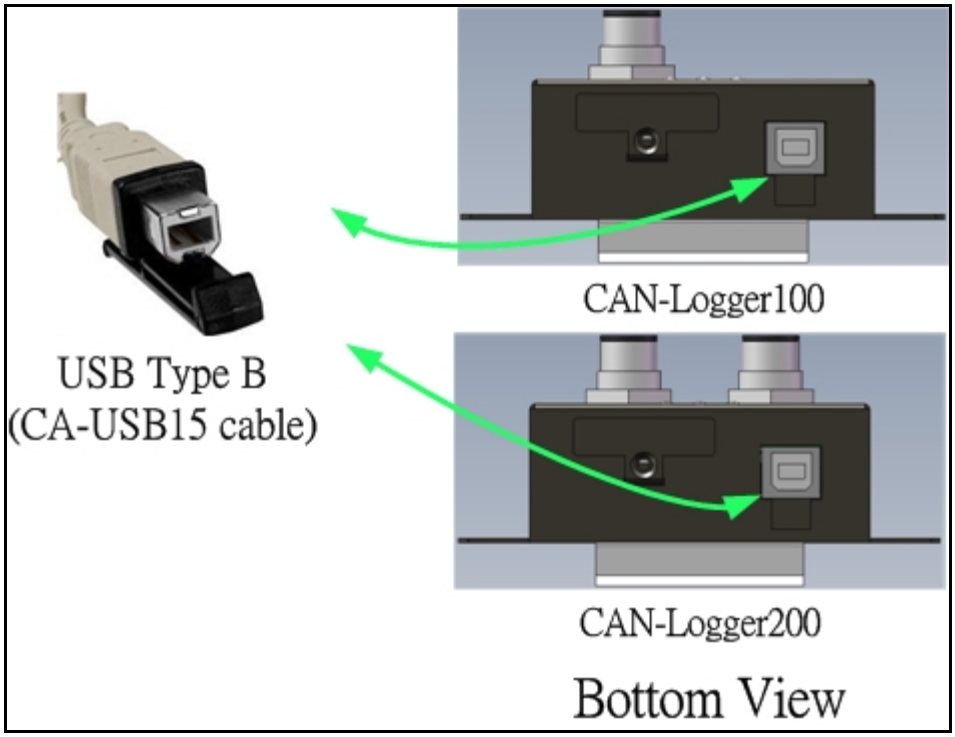


Figure 4-1 Wire connection of the USB

Step 2: Execute theCAN-Logger Utility tool.

4.3. Start to use CAN-Logger Utility tool

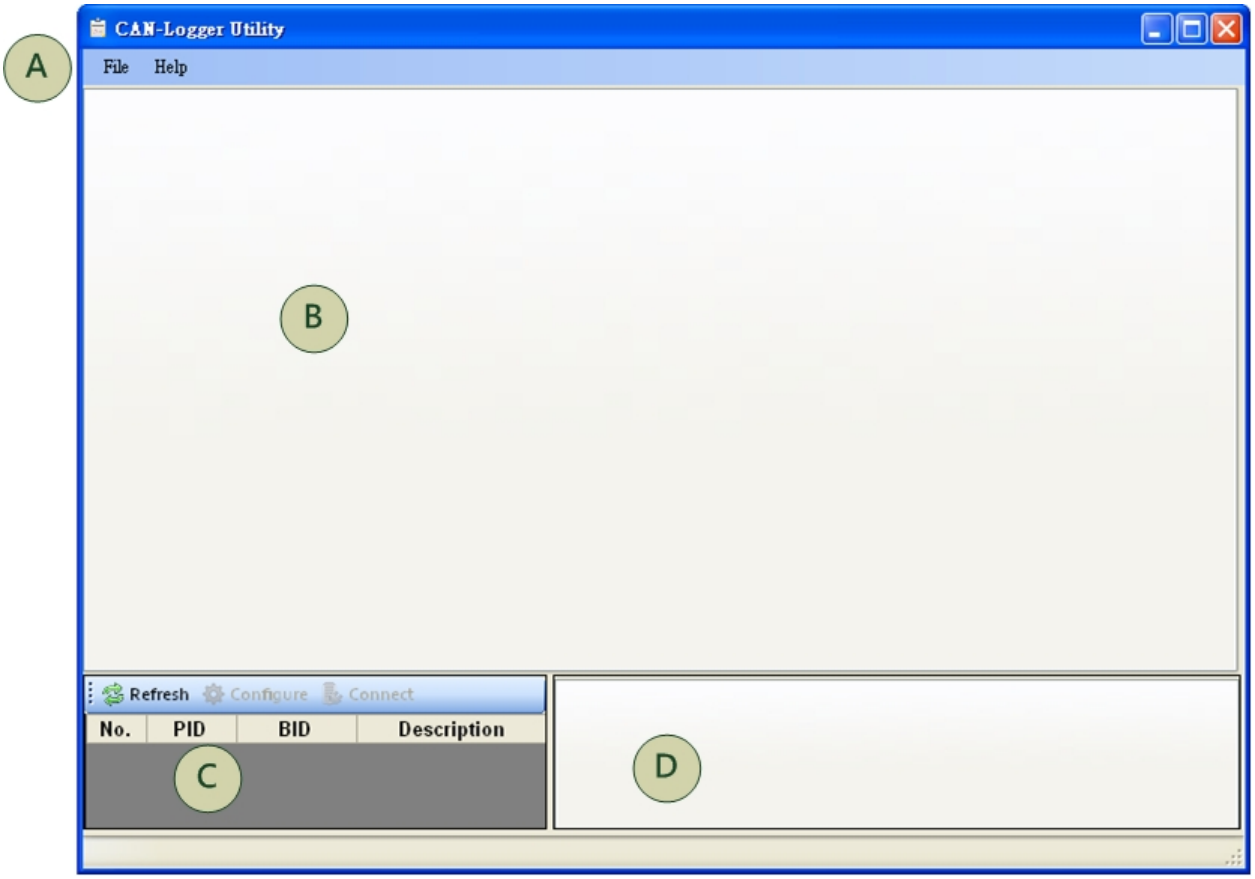


Figure 4-2 Main frame of the CAN-Logger Utility tool

A Menu tool bar. User can change the Utility communication mode between “Online” and “Offline” on the “File” item and get the Utility version information on the “Help” item. “Online” mode is used for user to transmit/receive CAN messages to the connected module and monitor the CAN Bus status. “Offline” mode is used for display the logged data on the module SD card.

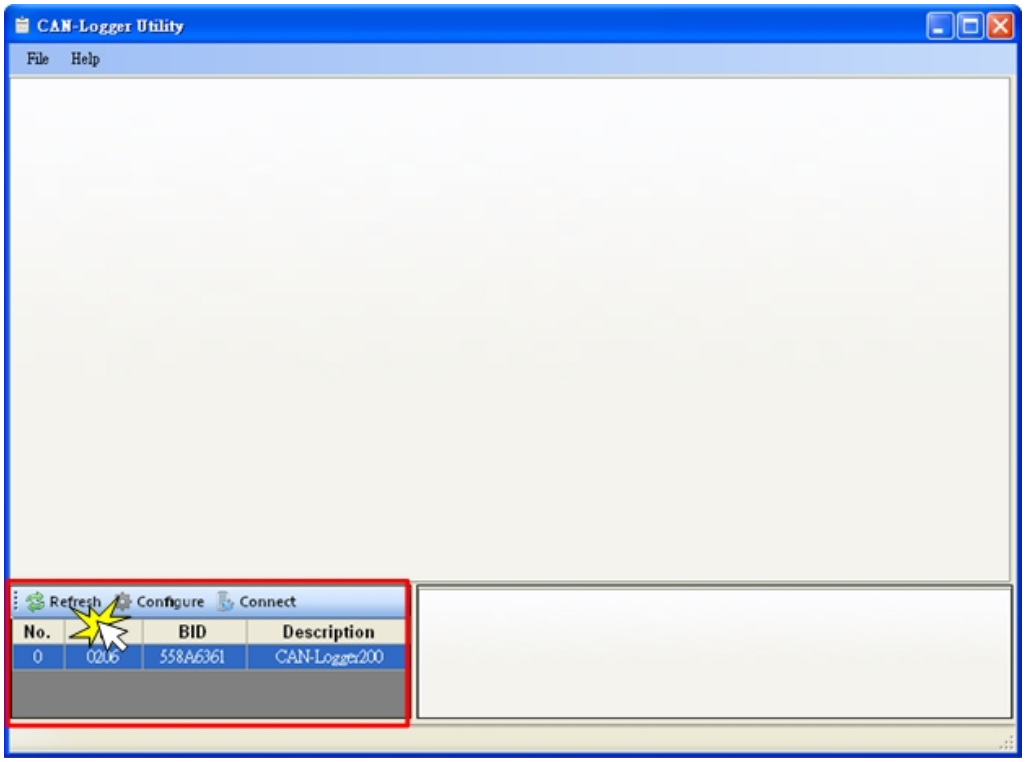
B Transmit/Receive frame. When using “Online” mode, this field will be divided into two parts after connect with module. One is used for display received CAN messages and the other is used for transmit CAN messages. When using “Offline” mode, this field is used for display the logged datas which are saved in the SD card. In this mode, user needs to use a SD card reader to get the SD data file.

C Scanned module information. On this frame, user can re-scan the new inserted module, configure the selected module and connect with the selected module.

D CAN Bus status frame. After connecting with module, user can get the CAN Bus status information on this frame.

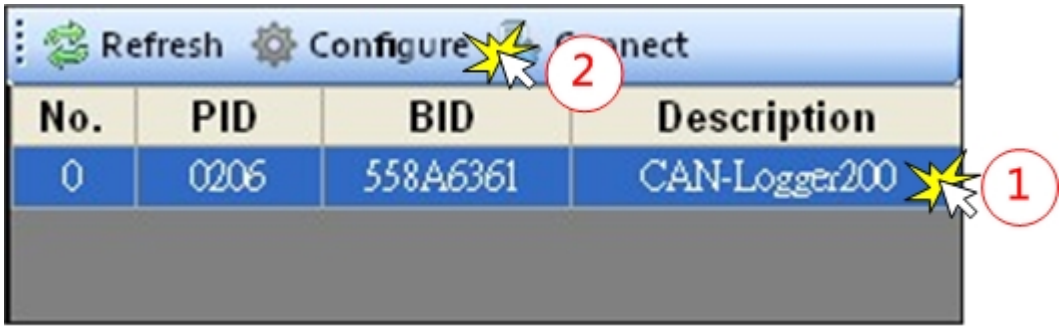
4.2.1. Configure the CAN-Logger series

When executing the Utility, the tool will try to scan all the necessary CAN-Logger modules and list all scanned module information on the left-bottom of the Utility main frame. User can press the “Refresh” buttom to re-scan the newer inserted CAN-Logger module.

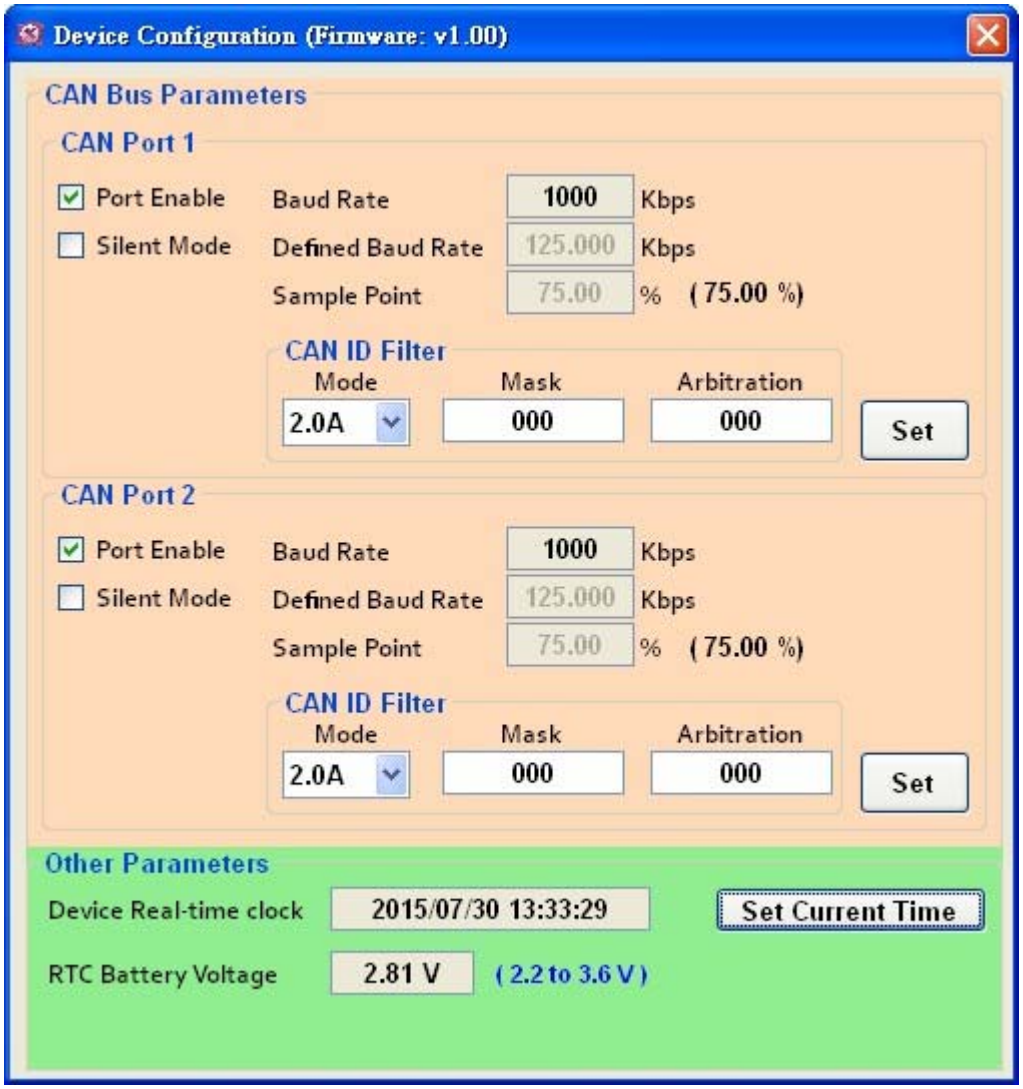


Via the CAN-Logger Utility, user can set the CAN port operation mode and filter parameter of the module. Please refer to the following steps to configure the CAN-Logger series.

Step1: Select the necessary CAN-Logger module and then press the “Configure” buttom.



Then the “Device Configuration” window will be popped up.



Step2: On the “Device Configuration” windows, user can set the CAN Bus parameter, and other prarameters. The detail functions of these parameters are list below.

[CAN Bus Parameters]

“Port Enable” : Enable/Disable the CAN1/CAN2 port.

“Silent Mode” : Set the CAN port into silent mode. When setting the CAN port into silent mode, the CAN port will just receive CAN messages, no CAN Ack command be sent to the CAN Bus.

“Baud Rate”: CAN Bus baud rate in used. Via setting the ratary switch from ‘0’ ~ ‘8’, user can change the CAN Bus baud rate from ‘10kbps’ to ‘1Mbps’. If you want to use user-defined CAN baud rate, user can set the ratary switch to ‘9’, and then set the “Defined Baud Rate” parameter to the CAN baud rate that you need.

“Defined Baud Rate”: CAN Bus baudrate used for user-defined. After set the CAN rotary switch to ‘9’, you can set this parameter to the baud rate you need.

“Sample Point”: Sample point of CAN baud rate bit timing.

“CAN ID Filter”: CAN message ID filter.

[Mode]: Mode of CAN ID, can be set to 2.0A (11-bit CAN ID) and 2.0B (29-bit CAN ID).

[Mask]: Mask CAN ID bits. Be use with [Arbitration] parameter. Bit value of 0 mean does not care the bit of corresponding “Arbitration CAN ID” and bit value of 1 means this bit need be matched with the corresponding bit of “Arbitration CAN ID”.

[Arbitration]: Arbitration CAN ID bit. The CAN ID that you want to use for the CAN ID filter.

[Example]

1. **All CAN ID passed.**
   [Mode] = “2.0A”
   [Mask] = “000”
   [Arbitration] = “000”

2. **Filter all messages except ID of 0x123.**
   [Mode] = “2.0A”
   [Mask] = “7FF”
   [Arbitration] = “123”

3. **Filter all messages except ID from 0x100 ~ 0x10F.**
   [Mode] = “2.0A”
   [Mask] = “7F0”
   [Arbitration] = “100”

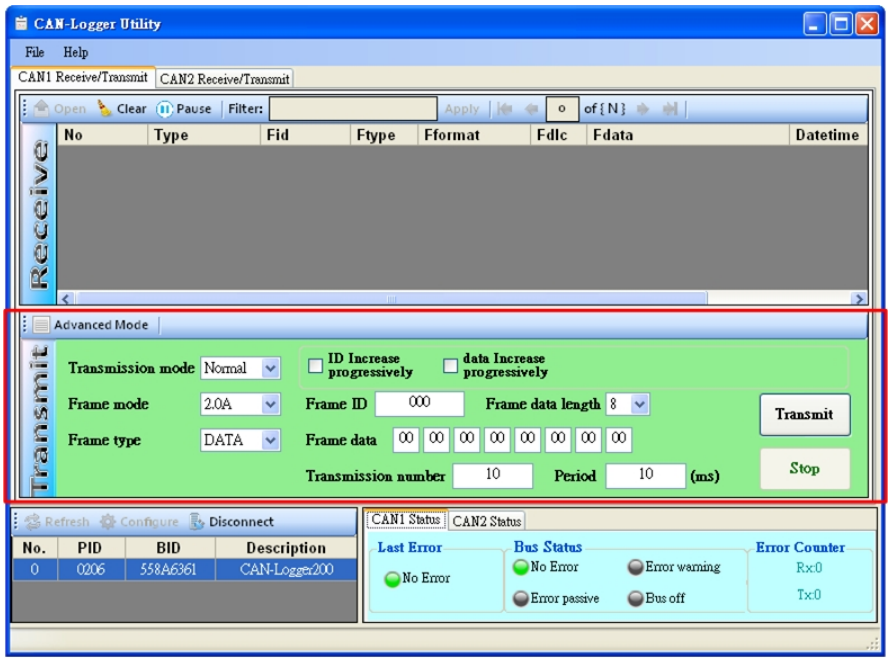
[Other Parameters]

“Device Real-time clock”: RTC time of the CAN-Logger device. This parameter is used for CAN-Logger device to log the received CAN messages into SD card.

“RTC Battery Voltage”: Crrent battery voltage that RTC used. The battery valtage must be the value between 2.2V_{DC} to 3.6V_{DC}.

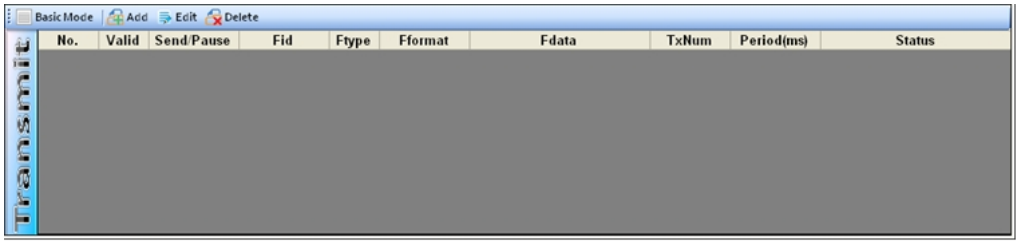
4.2.2. Transmit CAN messages

By using the CAN-Logger Utiltiy tool, user can send CAN meesages to CAN Bus via CAN-Logger devices.



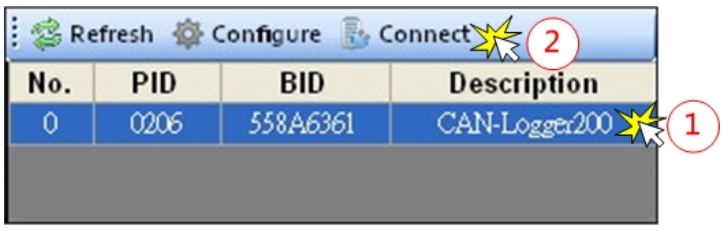
The CAN-Logger Utility supports two functions, “Basic Mode” and “Advanced Mode”, for user to send a CAN message. After connecting with the select CAN-Logger device via USB cable, user can use these two methods to send CAN messages to CAN Bus. The following steps listed how to used these two methods to send a CAN message.

“Basic Mode”



“Advanced Mode”

Step 1: Connect with the selected CAN-Logger device.



Step 2:

[Send CAN messages by using “Basic Mode”]

“Transmission mode”: Mode for transmitting a CAN message. Now the Utility tool only supports “Normal” method to send CAN messages.

“Frame mode”: Transmitted CAN meesage mode, included “2.0A” and “2.0B” items.
“2.0A” means using a CAN 11-bit ID format message.
“2.0B” means using a CAN 29-bit ID format message.

“Frame type”: Transmitted CAN meesage type, included “DATA” and “RTR” items.
“DATA” means using a CAN data frame message.
“RTR” means using a CAN remote frame message.

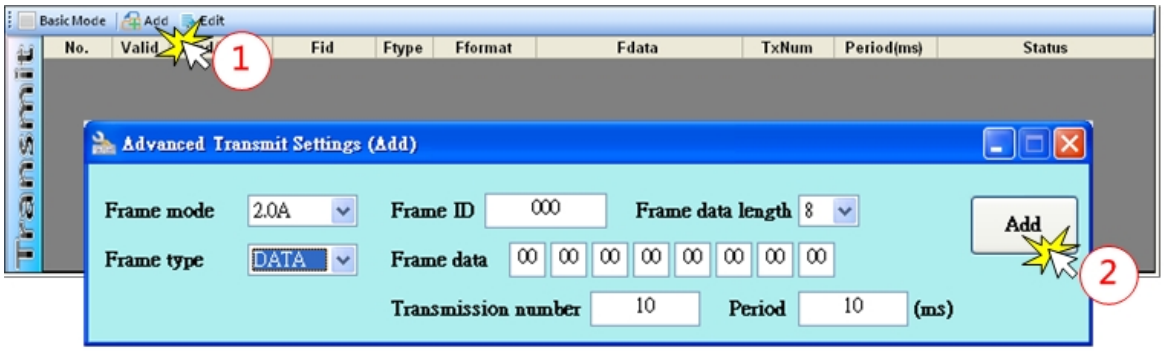
“ID increase progressively”: When enable this item, the transmitted “Frame ID” items will increase after sending a CAN message.

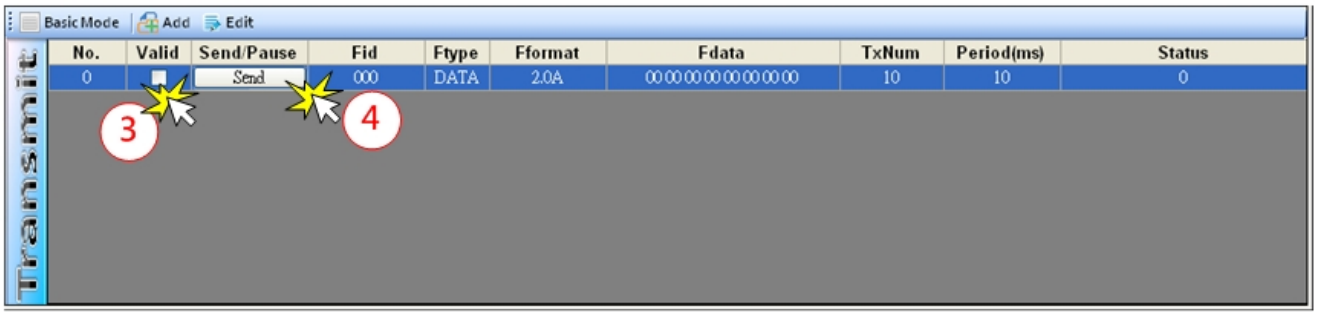
“data increase progressively”: When enable this item, the transmitted “Frame data” item will increase after sending a CAN message.

“Frame ID”: Transmitted CAN ID field.
“Frame data length”: Transmitted CAN data lengh field.
“Frame data”: Transmitted CAN data field.

“Transmission number”: Number of CAN messages need to be sent when press “Transmit” button.
“Period”: Period of sending a CAN message.

[Send CAN messages by using “Advanced Mode”]

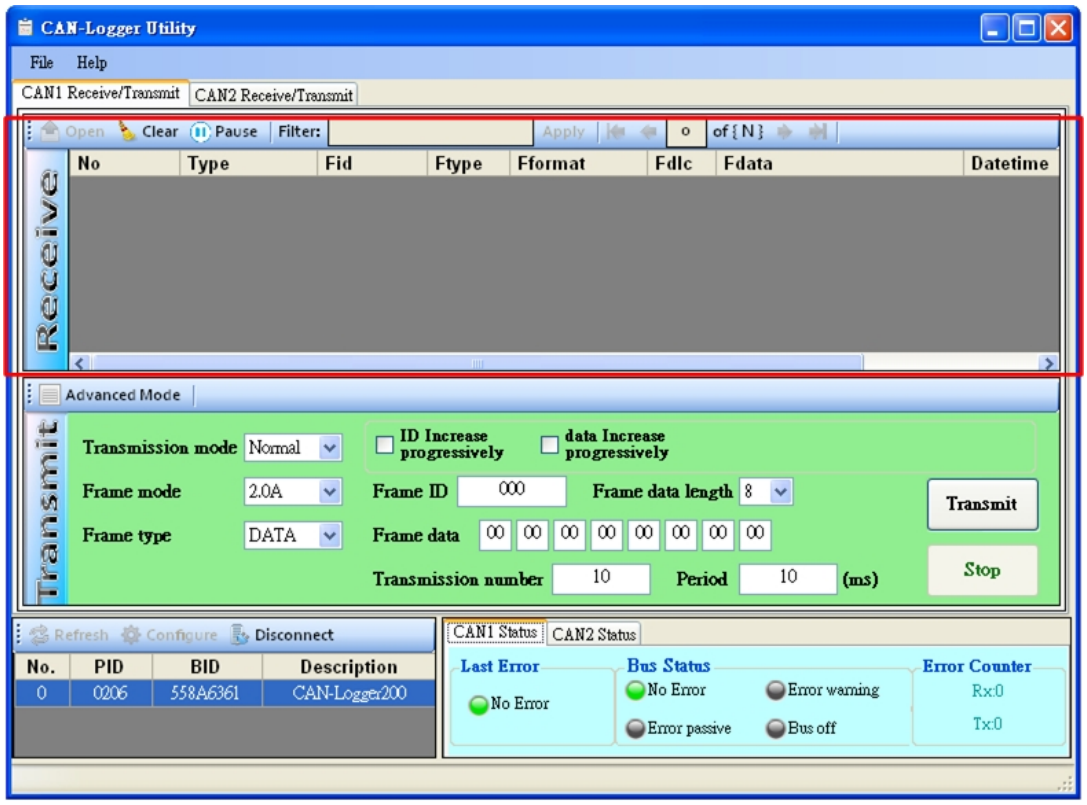




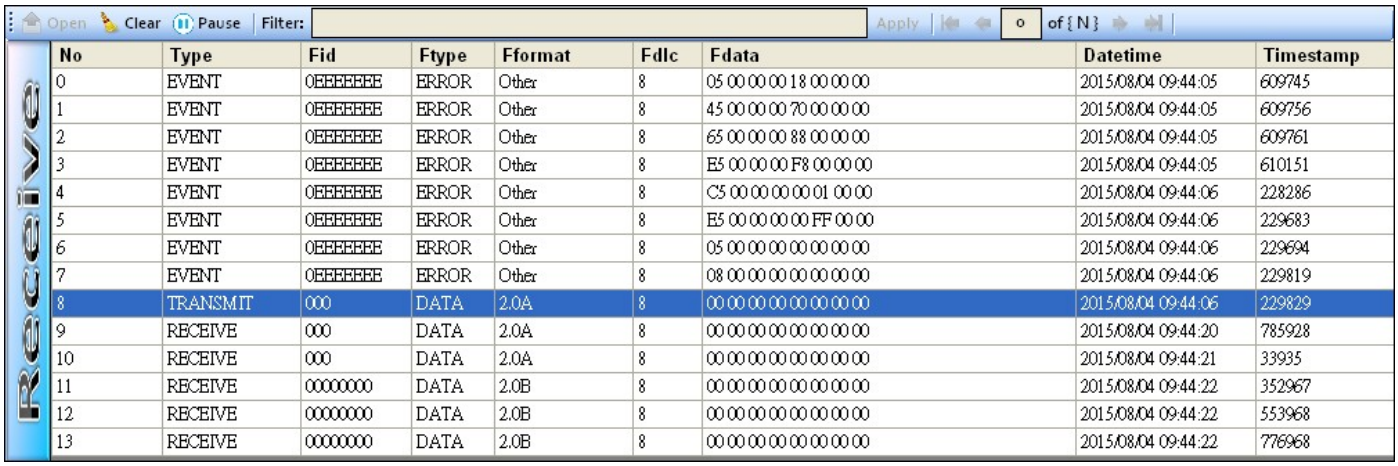
1. Click “Add” button to add a transmitted CAN message into the “Transmit list”.
2. Change the content of transmitted CAN message frame on the “Advanced Transmit Setting” frame. Then the transmitted CAN message will be saved into “Transmit list”.
3. Enable the “Valid” parameter of the message that you want to transmit.
4. Press the “Send” button to start to send data to CAN Bus via CAN-Logger device.

4.2.3. Receive CAN messages

By using the CAN-Logger Utiltiy tool, user can review the transmitted/received CAN meesages and error frame on CAN Bus via CAN-Logger devices.



After connecting with the CAN-Logger device, all transmitted, received and error messages on the CAN Bus will be shown on the “Receive frame”. The following picture is an example of data listed on the “Receive frame” and the detail information of each items on the “Receive frame” are shown below.



No	Type	Fid	Ftype	Fformat	Fdlc	Fdata	Datetime	Timestamp
0	EVENT	0EEEEEEE	ERROR	Other	8	05 00 00 00 18 00 00 00	2015/08/04 09:44:05	609745
1	EVENT	0EEEEEEE	ERROR	Other	8	45 00 00 00 70 00 00 00	2015/08/04 09:44:05	609756
2	EVENT	0EEEEEEE	ERROR	Other	8	65 00 00 00 88 00 00 00	2015/08/04 09:44:05	609761
3	EVENT	0EEEEEEE	ERROR	Other	8	E5 00 00 00 F8 00 00 00	2015/08/04 09:44:05	610151
4	EVENT	0EEEEEEE	ERROR	Other	8	C5 00 00 00 00 01 00 00	2015/08/04 09:44:06	228286
5	EVENT	0EEEEEEE	ERROR	Other	8	E5 00 00 00 00 FF 00 00	2015/08/04 09:44:06	229683
6	EVENT	0EEEEEEE	ERROR	Other	8	05 00 00 00 00 00 00 00	2015/08/04 09:44:06	229694
7	EVENT	0EEEEEEE	ERROR	Other	8	08 00 00 00 00 00 00 00	2015/08/04 09:44:06	229819
8	TRANSMIT	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/08/04 09:44:06	229829
9	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/08/04 09:44:20	785928
10	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/08/04 09:44:21	33935
11	RECEIVE	00000000	DATA	2.0B	8	00 00 00 00 00 00 00 00	2015/08/04 09:44:22	352967
12	RECEIVE	00000000	DATA	2.0B	8	00 00 00 00 00 00 00 00	2015/08/04 09:44:22	553968
13	RECEIVE	00000000	DATA	2.0B	8	00 00 00 00 00 00 00 00	2015/08/04 09:44:22	776968

“No”: number of the data listed in the frame.

“Type”: Type of data the CAN-Logger device received. This field will be “TRANSMIT”, “RECEIVE” and “EVENT”.

“TRANSMIT”: CAN Messages that send to the CAN Bus via CAN-Logger device.

“RECEIVE”: CAN messages that CAN-Logger device received.

“EVENT”: CAN error event that CAN-Logger device detected. The ”Fid”, “Ftype” and “Fformat” fields of “EVNET” message will be in the format of “0x0EEEEEEE”, “ERROR” and “Other”. And the data of “Fdata” field combines two parameters, CAN status and CAN error counter, which each parameter is 4-bytes in little-endian format. About the content of CAN status and CAN error counter, please refer to section 7.1 and 7.2 for detail information.

“Fid”: CAN ID of message frame.

“Ftype”: Type of the CAN meesage frame. This field will be “ERROR”, “DATA” and “RTR”.

“ERROR”: It means that this message is an “EVENT” message.

“DATA”: It means that this message is a CAN data message.

“RTR”: It means that this message is a CAN remote transmission request message.

“Fformat”: Format of the CAN meesage frame. This field will be “Other”, “2.0A” and “2.0B”.

“Other”: It means that this message is an “EVENT” message.

“2.0A”: It means that this message is a CAN 11-bit ID message.

“2.0B”: It means that this message is a CAN 29-bit ID message.

“Fdlc”: Data length of the “Fdata” field.

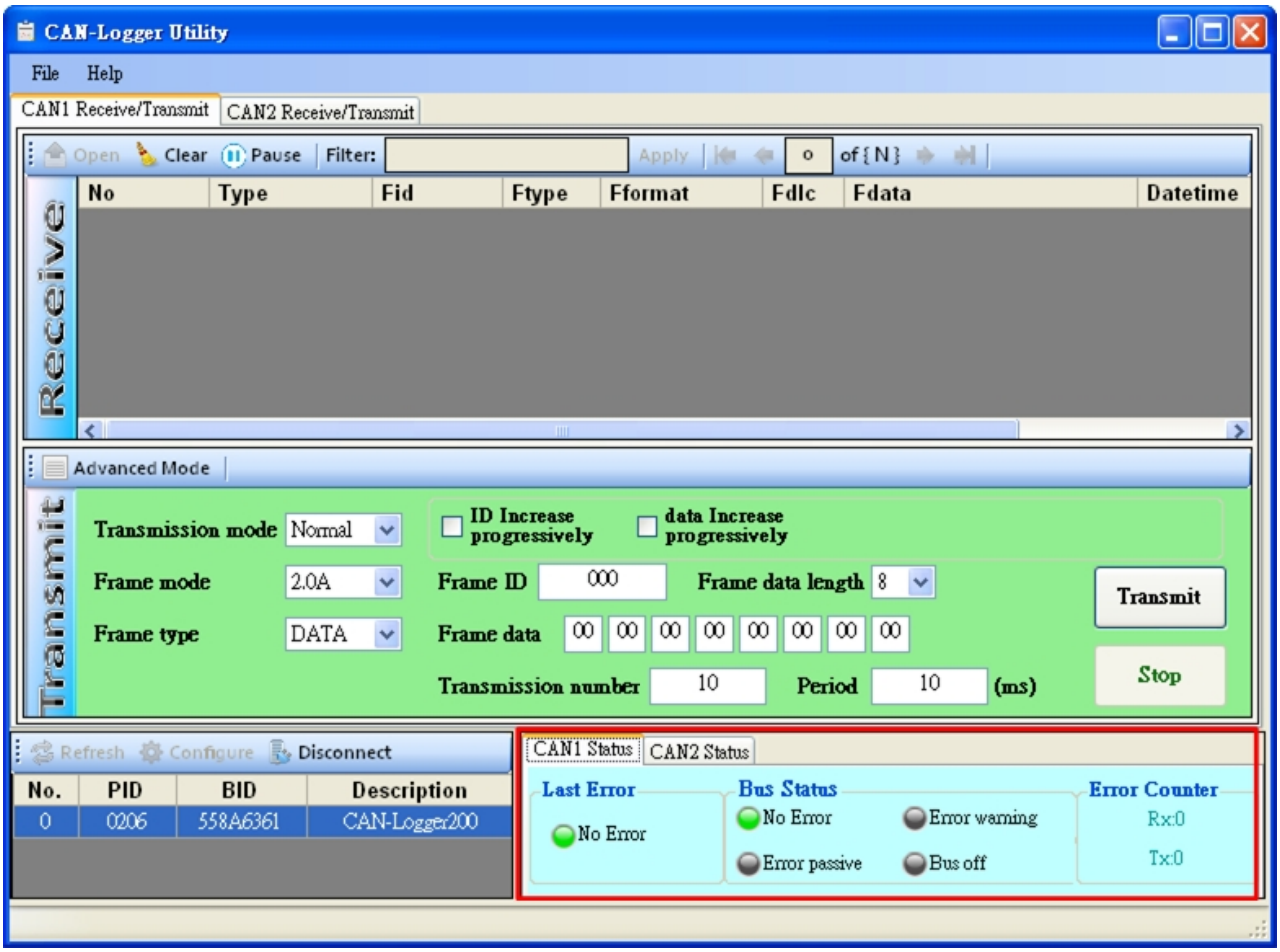
“Fdata”: Data of the CAN message. The content of this dield is dependent on the “Type” of the message. For “TRANSMIT” and “RECEIVE” message, this filed is used for the data of CAN message. For “EVENT” message, this field is used for CAN status and CAN error counter parameters.

“Datetime”: Date time of the CAN-Logger device to transmit/receive/detect the message.

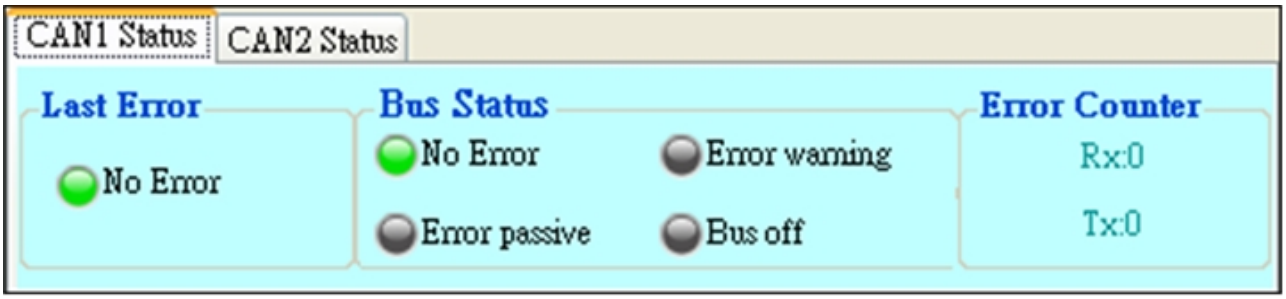
“Timestamp”: Timestamp of the CAN-Logger device to transmit/receive/detect the message. The unit of this field is micro-second.

4.2.4. Check CAN Bus state

By using the CAN-Logger Utiltiy tool, user can check the CAN Bus state of the CAN-Logger devices.



After connecting with the CAN-Logger device, the CAN Bus state of the device will be shown on the right-bottom of the main frame. The detail information of each items are shown below.



“Last Error”: Last error code.Type of the last error to occur on the CAN Bus.This field holds a status which indicates the type of the last error to occur on the CAN Bus.

Name	Description
No Error	No error occurred
Stuff Error	More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed.
Form Error	A fixed format part of a received frame has the wrong format.
Ack Error	The message this CAN core transmitted was not acknowledged.
Bit1 Error	During the transmission of a message (with the exception of the arbitration field), the device wanted to send a HIGH/recessive level (bit of logical value ‘1’), but the monitored bus value was LOW/dominant.
Bit0 Error	During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a LOW/dominant level (data or identifier bit logical value ‘0’), but the monitored Bus value was HIGH/recessive.
CRC Error	The CRC checksum was incorrect in the message received.
Unused	No CAN Bus event was detected

“Bus Status”: CAN Bus status.

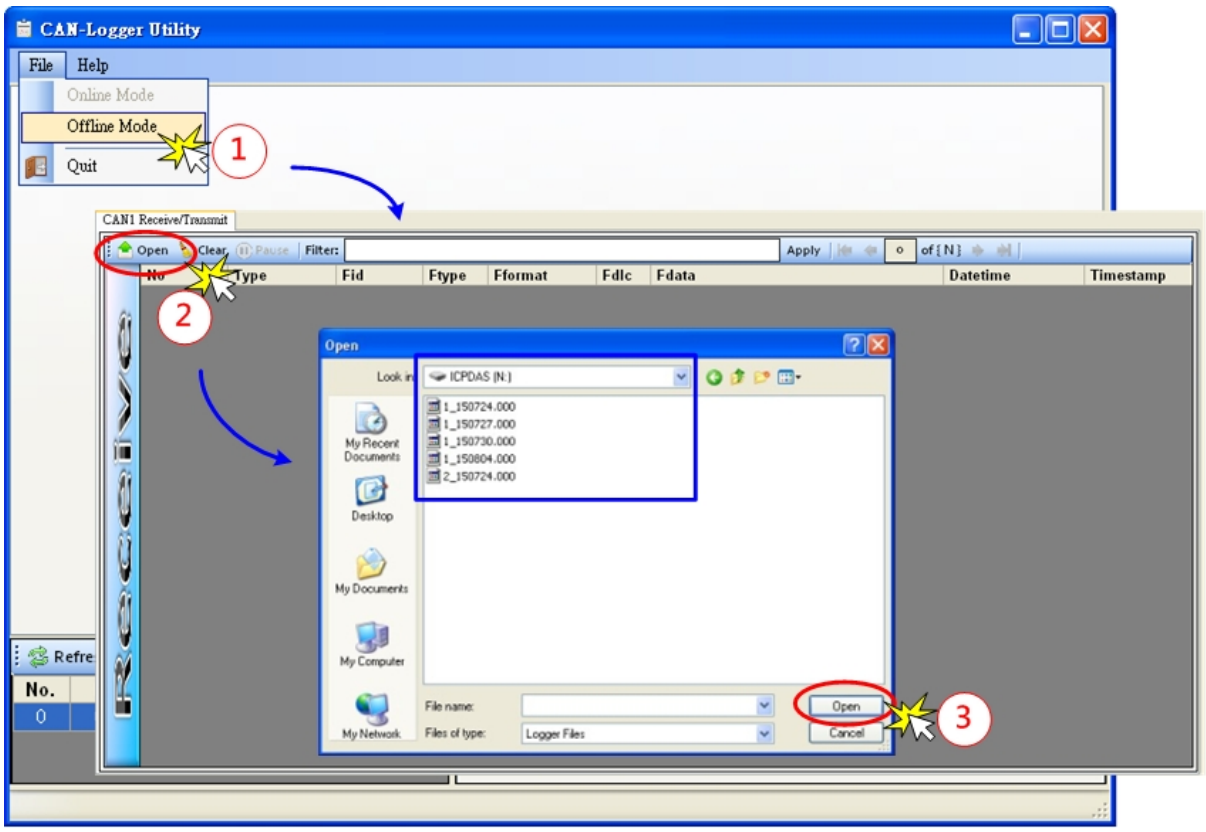
Name	Description
No Error	No error occurred
Error Warning	At least one of the transmit/receive error counters has reached the error warning limit of 96.
Error passive	The CAN controller is in the error passive state as defined in the CAN 2.0 specification.
Bus off	The CAN controller is in busoff state.

“Error Counter”: Transmit/Receve error counters.

Name	Description
Tx	Transmit error counter
Rx	Receve error counter

4.2.5. Verify data saved in the SD card

By using the “Offline Mode” function of CAN-Logger Utility tool, user can verify the data save in the SD card. Via this method, user need a SD card reader to access data save in the SD card.

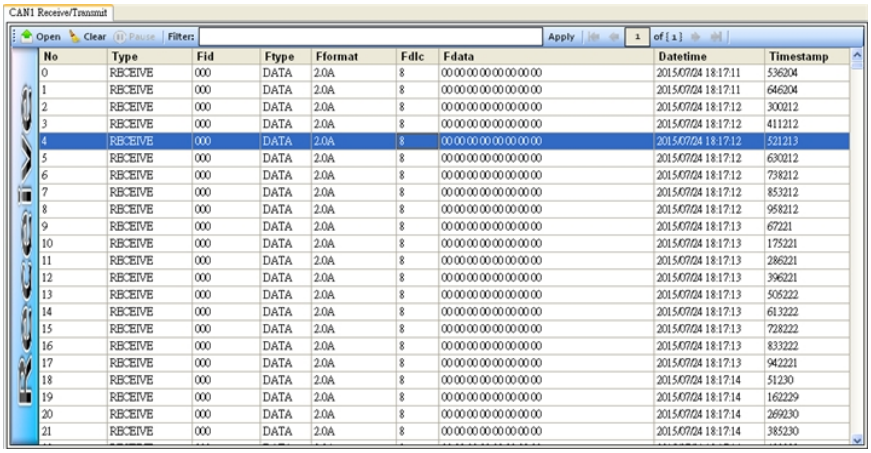


Step1: Change the Utility to “Offline Mode”.
Step2: Click the”Open” button, then the “Open” dialog will be popped up.
Step3: Select the SD card disk and the necessary file saved in the SD card.
The file name format of the data save in the SD card is listed below.

p_yymmdd.nnn

[p]: This file is used for CAN port ‘p’, value can be 1 or 2.
‘p’=1: CAN1
‘p’=2: CAN2
[yy]: Year after 2000, value from 01 to 99.
“yy”=15: 2015.
[mm]: month of the year, value from 01 to 12.
“mm”=08: August
[dd]: day of the month, value from 01 to 31.
[nnn]: file number, value from 000 to 999.

Then, press the “Open” button to open the selected file. All the data will be shown on the “Receive” frame of the Utility tool. About the message format on “Receive” frame, please refer to section 4.2.3 for detail information.



No	Type	Fid	Ftype	Fformat	Fdlc	Fdata	Datetime	Timestamp
0	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:11	536204
1	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:11	646204
2	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:12	300212
3	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:12	411212
4	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:12	521213
5	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:12	630212
6	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:12	738212
7	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:12	853212
8	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:12	958212
9	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:13	67221
10	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:13	175221
11	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:13	286221
12	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:13	396221
13	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:13	505222
14	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:13	613222
15	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:13	728222
16	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:13	833222
17	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:13	942221
18	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:14	51230
19	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:14	162229
20	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:14	269230
21	RECEIVE	000	DATA	2.0A	8	00 00 00 00 00 00 00 00	2015/07/24 18:17:14	385230

The Utility tool also support filter function for user to display the necessary messages on the “Receive” frame.

Each row header string is the key string for user to use to filter and get the necessary message.

No	Type	Fid	Ftype	Fformat	Fdlc	Fdata	Datetime	Timestamp

Except the key string, user also can combine the key string with the key word “and” or “or” for multiple filter functions. The rules about how to use the filter function are listed below.

[key string]: ‘No’, ‘Type’, ‘Fid’, ‘Ftype’, ‘Fformat’, ‘Fdlc’, ‘Fdata, ‘Datetime’, ‘Timestamp’
[operator]: ‘=’, ‘<’, ‘>’, ‘<=’, ‘>=’
[key word]: ‘and’, ‘or’

[filter data]: data that you want to combine with the key string. Need to add the char **‘** between the front side and back-end of the filter data.

Example:
Filter rule =
[key string1] **[operator]** ‘[filter data1]’ **[key word]** [key string2] **[operator]** ‘[filter data2]’

Example 1: Display range of messages that data on “No” field is between 1000 and 2000.

Filter rule: No > ‘1000’ and No < ‘2000’

Example 2: Display “RECEIVE” type message.

Filter rule: Type = ‘RECEIVE’

Example 3: Display “RECEIVE” type message and “Fid” field is equal to ‘1FF’.

Filter rule: Type = ‘RECEIVE’ and Fid = ‘1FF’

Example 4: Display range of messages that data on “Datetime” field is between 2015/01/01 and 2015/06/31

Filter rule: Datetime >= ‘2015/01/01’ and Datetime <= ‘2015/01/01’

Example 5: Display messages that data on “Fdata” field is match ’11 22 33 44 55 66 77 88’

Filter rule: Fdata = ‘11 22 33 44 55 66 77 88’

4.2.5.1. Format SD card

The CAN-Logger series maximum support 32 GB FAT format SDHC card. The default accessory of SDHC card is 4GB. If user needs large size of SDHC card, user can replace the default with the newer large size of SDHC card. If user uses their own SDHC card, inorder to make the CAN-Logger series to have the best performance to read/write data from/to SDHC card, we suggest user to format the SDHC card to FAT32 with 32KB clusters before used. User can use the following command on cmd windows of windows OS to format the SDHC card on the SDHC card reader device.

Example: Format the SDHC card (j disk) to FAT32 with 32k cluster format and label it to ”CAN_LOG”.

```
format j:/fs:fat32 /a:32K /v:CAN_LOG
```

format:	format command supported by windows OS.
j:	The SDHC card on SDHC card reader is simulated to j disk.
/fs:filesystem:	Specifies the type of the file system (FAT, FAT32). FAT32 file system is strongly recommended for CAN-Logger series use
/a:size:	Overrides the default allocation unit size. Size of 32K setting is strongly recommended for CAN-Logger series use.
/v:label:	Specifies the volume label. Default setting is “CAN_LOG’

5. Firmware Upgrade

Please refer to the following steps to upgrade the firmware of module. Here uses the CAN-Logger200 for example.

Step 1: Power off the CAN-Logger200.

Step 2: Set the CAN1 baud rate of rotary switch to ‘F’ and connect the PC available USB port with the USB port of the CAN-Logger200. Users can find the communication cable (CA-USB15) in the product box.

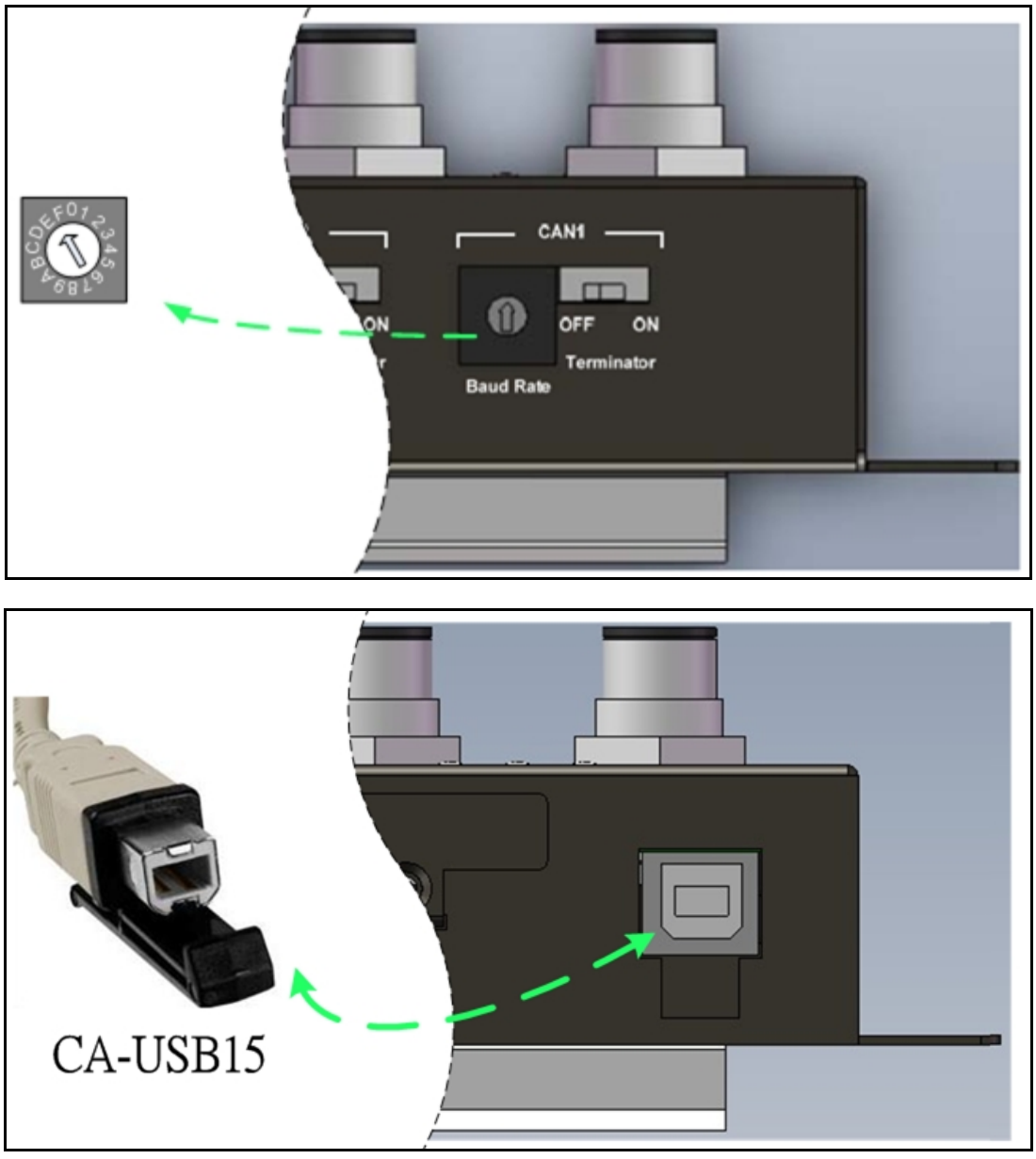
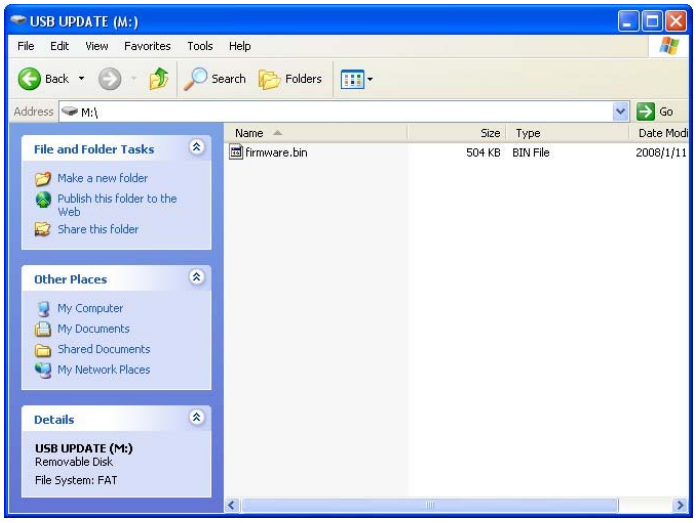


Figure 5-1 Rotary switch setting and wire connection of the USB

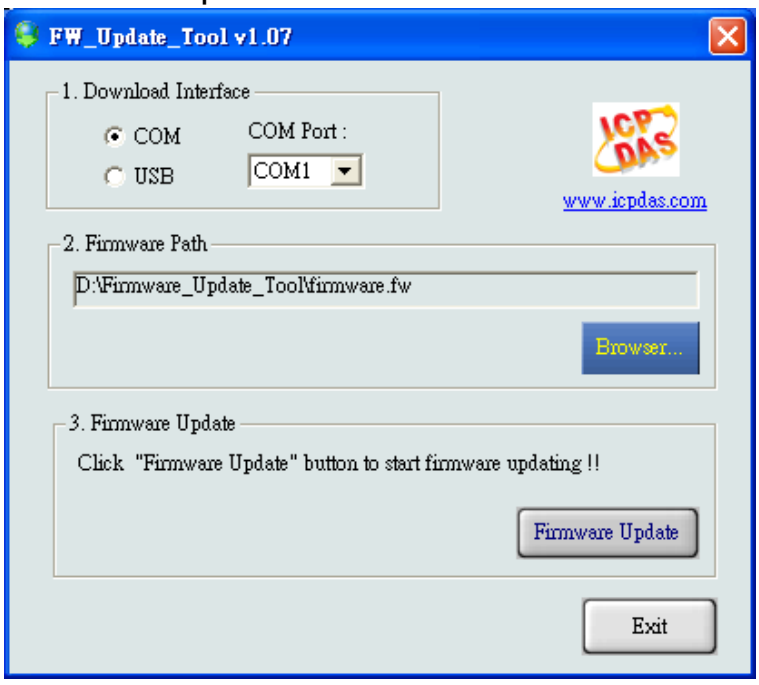
Step 3: Then, the module will be enter into “Firmware Upgrade mode”. In this mode, the Power, MS, SD_ST, CAN_ST, CAN2, CAN1 LEDs of the module will scroll to flash per 200 milliseconds and users can upgrade the firmware of the CAN-Logger series module via USB and the module will become a “USB Mass Storage Device” and also shows a folder like following picture automatically.



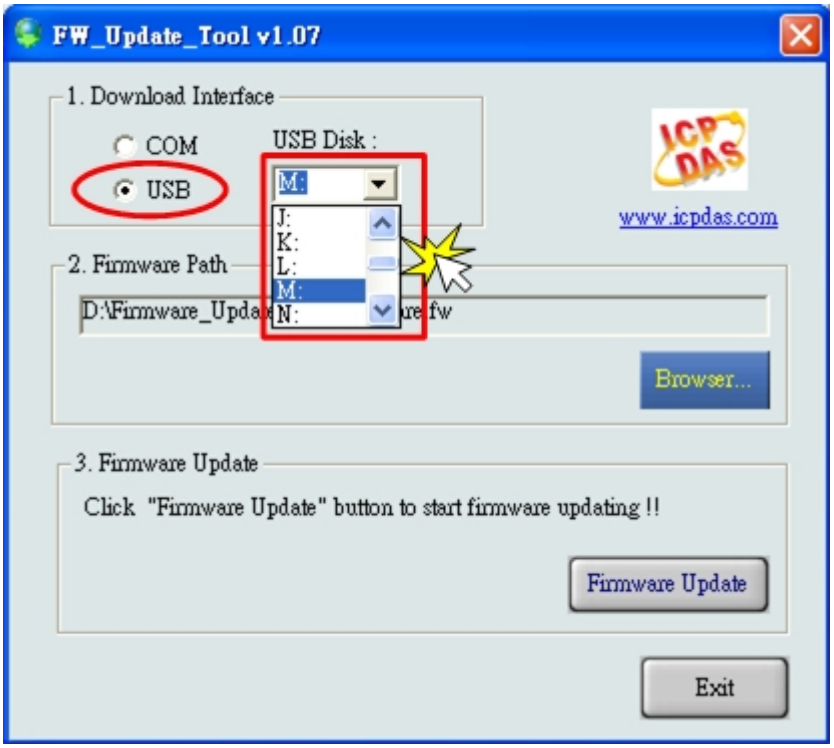
Step 4: Get the “Firmware Update Tool” and firmware file.
The software is located at:
Fieldbus_CD:\CAN\CAN_Logger\Software\Tool
The firmware is located at:
Fieldbus_CD:\CAN\CAN_Logger\Firmware\

http://www.icpdas.com/root/product/solutions/industrial_communication/fieldbus/can_bus/specific_device/can-logger.html

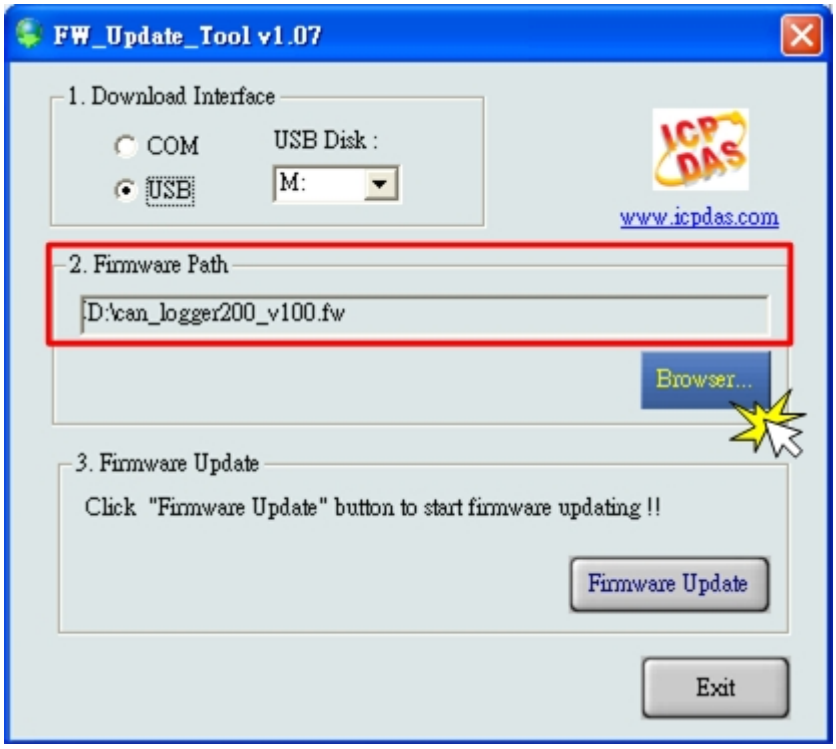
Step 5: Execute the “Firmware Update Tool”.



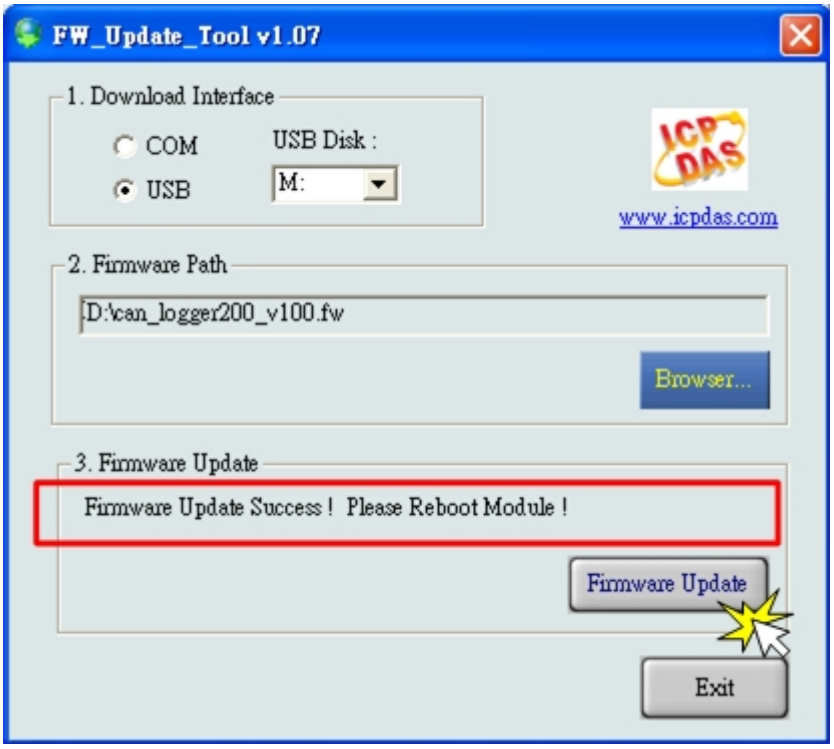
Step 6: Select USB port and the necessary USB Disk of PC.



Step 7: Press the the “Browser…” button and select the firmware file (*.fw).



Step 8: Press the “Firmware Update” button to update the firmware. After successfully to upgrade the firmware, the “Firmware Update Success! Please Reboot Module!” information will be display on the “3. Firmware Update” frame.



Step 9: Set the CAN1 baud rate of rotary switch to the necessary location.

Step 10: Reboot the module and press the “Exit” button to exit.

6. Dimension

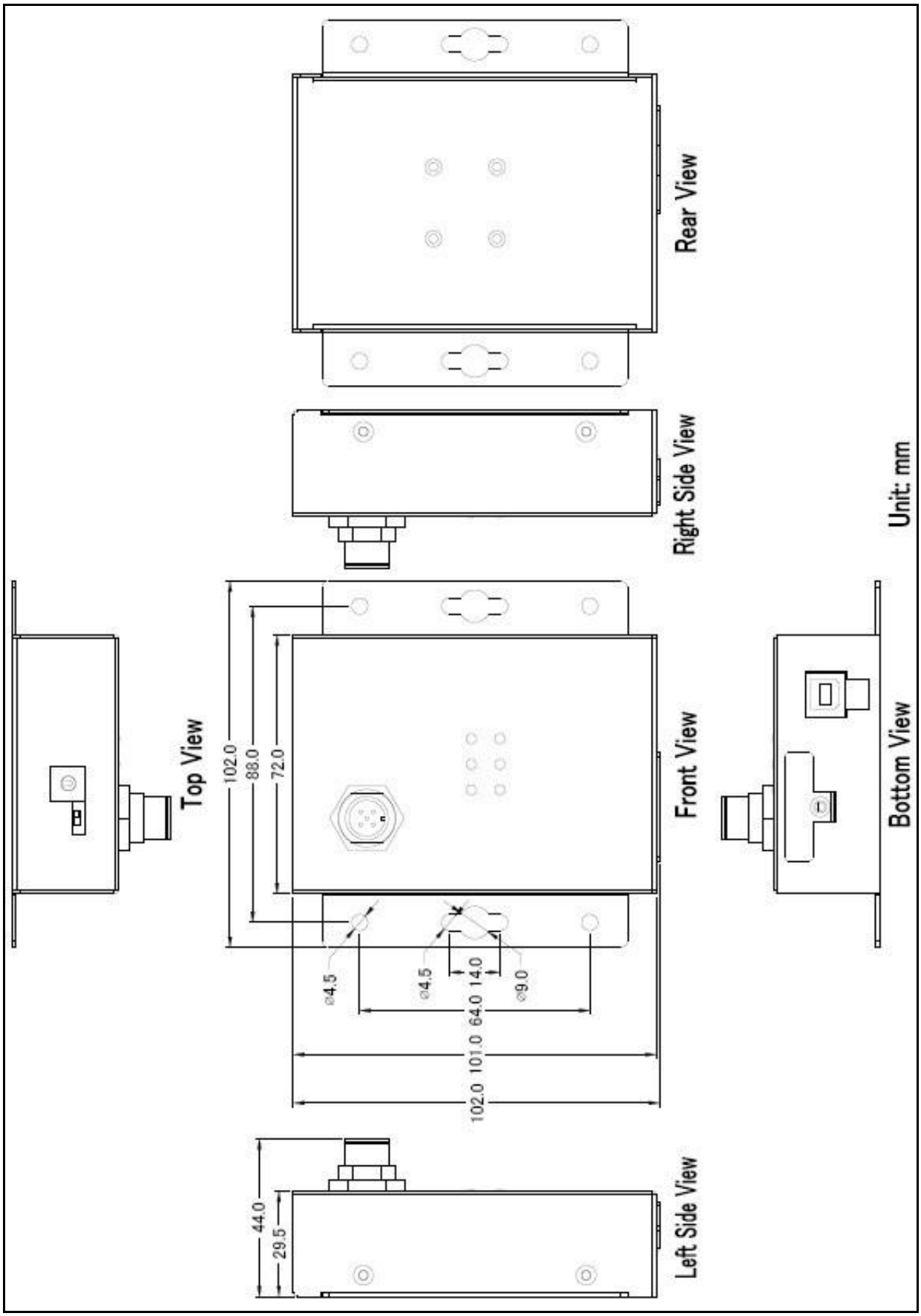


Figure 6-1 Dimension of CAN-Logger100

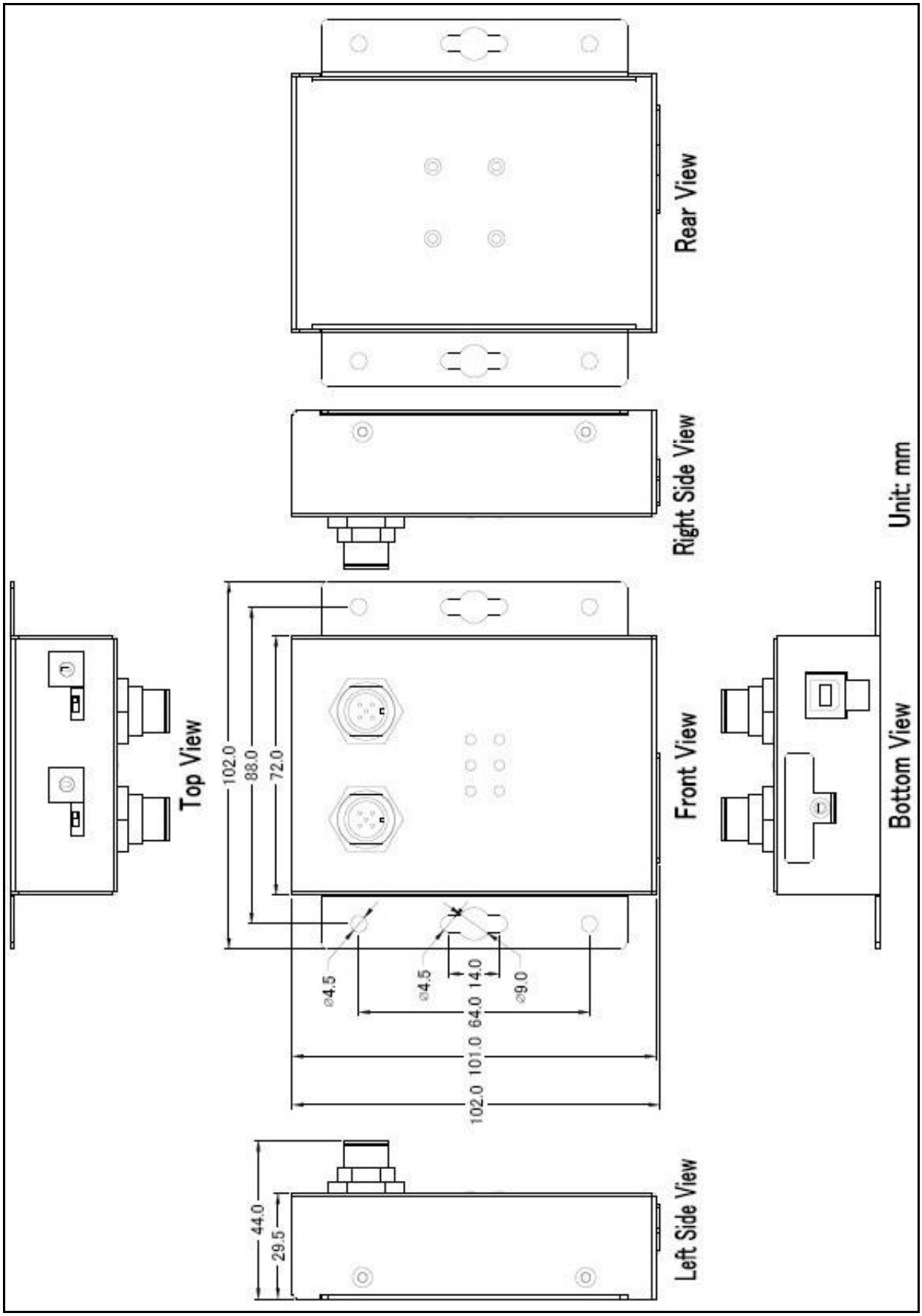


Figure 6-2 Dimension of CAN-Logger200

7. Appendix

7.1. CAN Status Register

Bit	Symbol	Value	Description
2:0	LEC		**Last error code** Type of the last error to occur on the CAN bus.The LEC field holds a code which indicates the type of the last error to occur on the CAN bus.
		0x0	**No error.**
		0x1	**Stuff error**: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed.
		0x2	**Form error**: A fixed format part of a received frame has the wrong format.
		0x3	**AckError**: The message this CAN core transmitted was not acknowledged.
		0x4	**Bit1Error**: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a HIGH/recessive level (bit of logical value '1'), but the monitored bus value was LOW/dominant.
		0x5	**Bit0Error**: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a LOW/dominant level (data or identifier bit logical value '0'), but the monitored Bus value was HIGH/recessive.
		0x6	**CRCError**: The CRC checksum was incorrect in the message received.
		0x7	**Unused**: No CAN bus event was detected
3	TXOK		Transmitted a message successfully.
		0	No message has been successfully transmitted.
		1	A message has been successfully transmitted.
4	RXOK		Received a message successfully
		0	No message has been successfully received
		1	A message has been successfully received independent of the result of acceptance filtering.
5	EPASS		Error passive
		0	The CAN controller is in the error active state.
		1	The CAN controller is in the error passive state as defined in the CAN 2.0 specification.
6	EWARN		Warning status
		0	Both error counters are below the error warning limit of 96.
		1	At least one of the error counters in the Error Counter Register has reached the error warning limit of 96.
7	BOFF		Busoff status
		0	The CAN module is not in busoff state.
		1	The CAN controller is in busoff state.
31:8	-	-	Reserved

7.2. CAN Error Counter Register

Bit	Symbol	Value	Description
7:0	TEC		Transmit error counter Current value of the transmit error counter (maximum value 255)
14:8	REC		Receive error counter Current value of the receive error counter (maximum value 127).
15	RP		Receive error passive
		0	The receive counter is below the error passive level.
		1	The receive counter has reached the error passive level as defined in the CAN2.0 specification.
31:16	-	-	Reserved

7.3. EMI Ferrite Split/Snap-On Core

Features

- Aimed to suppress low frequency noise generated by engine control units, inverters, and motors
- Split type
- Operation Temperature: -25°C ~ 75°C

Introduction

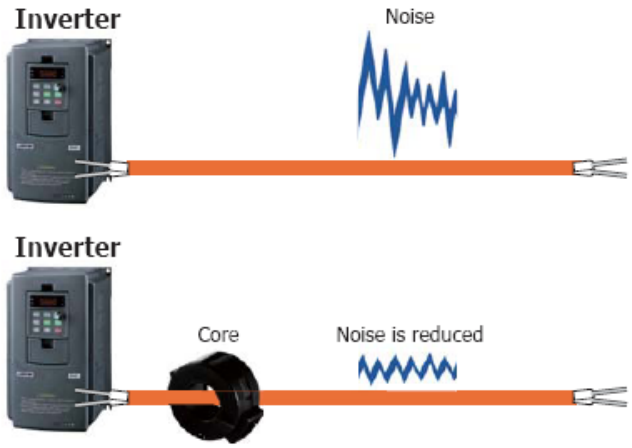
The split ferrite cable cores are designed to significantly reduce EMI/RFI for round cables.
The hinged plastic case surrounding the split core is designed to clamp onto the cable to provide a secure fixture of the ferrite onto the cable. The cores can be retrofitted onto existing installations or used in post-assembly operations on the data and power cables of electronic equipment. Ferrite cores are important for ensuring strong electronic signals through cables in environments where EMI or RFI can be an issue.

Specifications

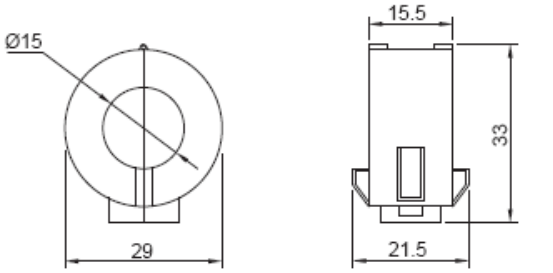
Mechanical	
Max. Cable Diameter	Ø15 mm
Material Type	Board Band Material
Additional Description	Plastic Case
Case Color	Black

Applications

RS-232, RS-422, RS-485, CAN bus, FRnet, PROFIBUS, Ethernet, USB, AC/DC Power line..etc

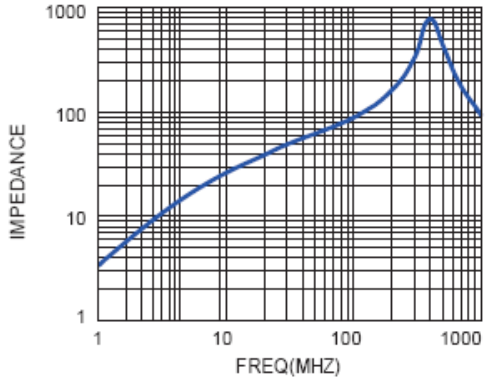


Dimensions (Units: mm)

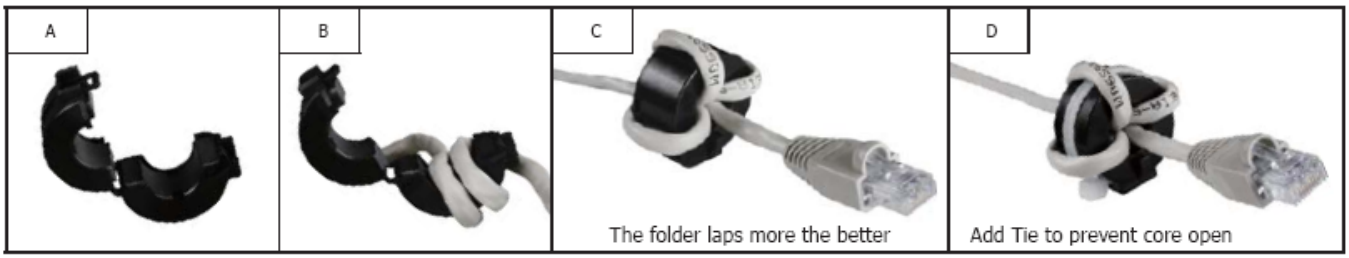


Front View | Left Side View

Characteristic



Installation



Clip-on Ferrite Core Installation