



**Ginkgo**  
**USB-CAN Interface**  
*VTG203B*  
**User Manual**  
**v1.2**

Chapter 1 Product Introduction.....	- 1 -
1.1 General Introduction.....	- 1 -
1.2 Property and Technical Specification.....	- 1 -
1.3 Typical Application.....	- 2 -
1.4 Sales Order .....	- 2 -
1.5 Operation Condition.....	- 2 -
1.6 Technical Support and Service .....	- 3 -
Chapter 2 Outline and Interface Description .....	- 4 -
2.1 Outline and Interface .....	- 4 -
2.2 Interface Definition.....	- 4 -
2.3 Wiring.....	- 4 -
Chapter 3 Software Instruction .....	- 5 -
3.1 Software Download & Install .....	- 5 -
3.1.1 Software Download.....	- 5 -
3.1.2 Software Install.....	- 5 -
3.2 Software Operation Instruction.....	- 9 -
3.2.1 Software Introduction .....	- 9 -
3.2.2 Initialization Settings .....	- 10 -
3.2.3 Filter Setting .....	- 11 -
3.2.4 Send Data .....	- 12 -
3.2.5 Color Setting.....	- 13 -
3.2.6 Alias Definition .....	- 14 -
3.2.7 CAN Bus Status Display .....	- 14 -
3.2.8 Data Receive.....	- 15 -
3.2.9 Data Sending in Identifier List Mode.....	- 16 -
3.2.10 Adapter Status Display .....	- 16 -
Chapter 4 CAN bus Frame Format.....	- 17 -
4.1 Standard Data Frame Structure .....	- 17 -
4.2 Extended Data Frame Structure .....	- 17 -
4.3 Standard Remote Frame Structure .....	- 17 -
4.4 Extended Remote Frame Structure .....	- 18 -

# Chapter 1 Product Introduction

## 1.1 General Introduction

Ginkgo USB-CAN Interface is an adapter equipped with both USB2.0 interface and 2-channel CAN interface which can be used for bidirectional data transmission.

As a standard CAN node, Ginkgo USB-CAN Interface is a powerful tool in CAN bus product development, CAN bus device test and data analysis. PC is connected to standard CAN network via USB interface to perform data collection, data process as well as data communication when forging an onsite bus testing laboratory or in industrial control and automotive electronics development etc. Ginkgo USB-CAN Interface is small in size and convenient to install thus is an ideal portable device in real application.

CAN bus configuration, data sending and receiving are realized via PC software. A set of standard universal software (for free) and quadratic programmable API are provided to the user and the user can write the PC software on the basis of real needs. Application routine of development tools provided at present includes: Labview, C#, VC, Qt, Python, Labwindows/CVI, C++ Builder, VB, VB.NET, Delphi, CodeBlocks etc. Please download at the official website of ViewTool ([www.viewtool.com](http://www.viewtool.com)). It is no need to learn complicated USB interface communication protocol when develop quadratic software.

## 1.2 Property and Technical Specification

- USB and CAN bus protocol conversion.
- CAN interface with two isolated channels.
- USB supports USB2.0 with USB1.1 compatible.
- 2500 V<sub>RMS</sub> isolation.
- Low loop delay: 150ns (Typical), 210ns (Max).
- Interface has a typical TI (transient immunity) of 50kV/μs.
- -27V to 40V bus fault protection.
- Port featured with electrostatic protection function (Air - 15kV, Contact - 8kV) .
- Supports CAN2.0A, CAN2.0B protocol, standard frame and extended frame. Each channel supports 14 filters for most. Filtering has identifier mask mode and identifier list mode for selection.
- Supports bidirectional transmission, CAN sending and receiving.
- Supports data frame and remote frame format.
- Baud rate of CAN controller is selectable in a range of 20Kbps-1Mbps which can be set via software.
- Maximum flow reaches 8000 FPS (Standard ID data frame 8 byte) or above.  
Standard remote frame rate reaches 18000 FPS.

- Each channel has an isolated buffer of 2000 frame.
- USB bus powered directly without any external power supply.
- With USB interface protection function.
- Working temperature: -40 to 85°C.
- Working current 80mA, with a consumption less than 400mW.
- Size: 103mm×53mm×28mm.

### 1.3 Typical Application

- CAN bus network data sending and receiving via USB of PC.
- CAN network data monitoring in industrial site.
- Automotive CAN bus application.
- CAN network diagnosis and test.
- CAN network data acquisition and analysis.
- CAN bus to USB gateway.
- Extend the distance of CAN bus network communication.
- USB to CAN network interface.

### 1.4 Sales Order

NO.	Name	Quantity
1	Ginkgo USB-CAN Interface	1 PCS
2	USB A→B cable	1 PCS
3	Quick guide of Ginkgo series	1 PCS

\*In case of any damage or accessories missing, please contact the nearest agency as soon as possible.

### 1.5 Operation Condition

Requirement on operating system:

Name of OS	Specification
Windows XP(32-Bit and 64-Bit version)	Support
Windows7(32-Bit and 64-Bit version)	Support
Windows VISTA(32-Bit and 64-Bit version)	Support
Windows8(32-Bit and 64-Bit version)	Support(ban digital signature of 64-Bit system )
Linux(32-Bit and 64-Bit version)	Support
MAC OS	Support
Android	Support

Requirement on hardware:

Name of Hardware	Min. Configuration Requirements	Recommend Configuration
RAM	166 MHz	900 MHz
Display	64 MB	256 MB
HDD	Resolution ratio: 800*600 or higher	Resolution ratio: 1024*768
USB	100MB or more available space	100MB or more available space
RAM	Support USB1.1	Support USB2.0

## 1.6 Technical Support and Service

- Lifetime technical support, free services and maintenance within one year.
- Down load driver, software, User's Manual and application routine source code at [www.viewtool.com](http://www.viewtool.com).
- Technical exchange forum: [www.viewtool.com/bbs](http://www.viewtool.com/bbs)
- Technical support Email: [FAE@viewtool.com](mailto:FAE@viewtool.com)
- Sales Email : [Sales@viewtool.com](mailto:Sales@viewtool.com)

# Chapter 2 Outline and Interface Description

## 2.1 Outline and Interface



## 2.2 Interface Definition

- H: CAN-High signal
- L: CAN-Low signal
- R: 120Ω terminal resistance connecting selection
- G: CAN bus GND

## 2.3 Wiring

- H: Connected to CAN-High signal of CAN bus
- L: Connected to CAN-Low signal of CAN bus
- R and L: 120Ω terminal resistance connected to CAN bus after short circuit
- G: Connected to GND of CAN bus

# Chapter 3 Software Instruction

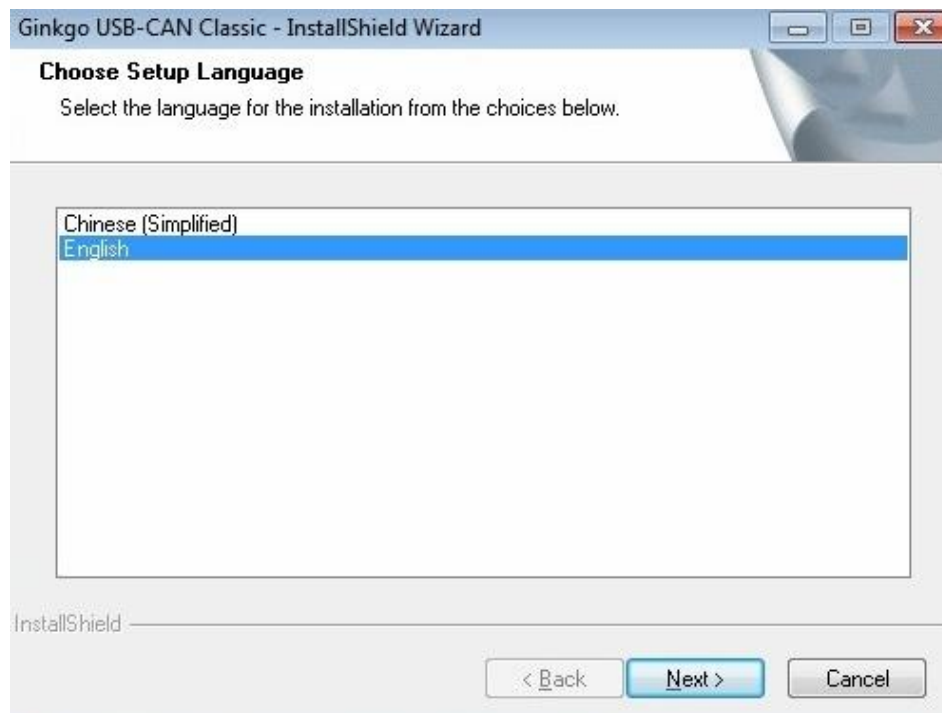
## 3.1 Software Download & Install

### 3.1.1 Software Download

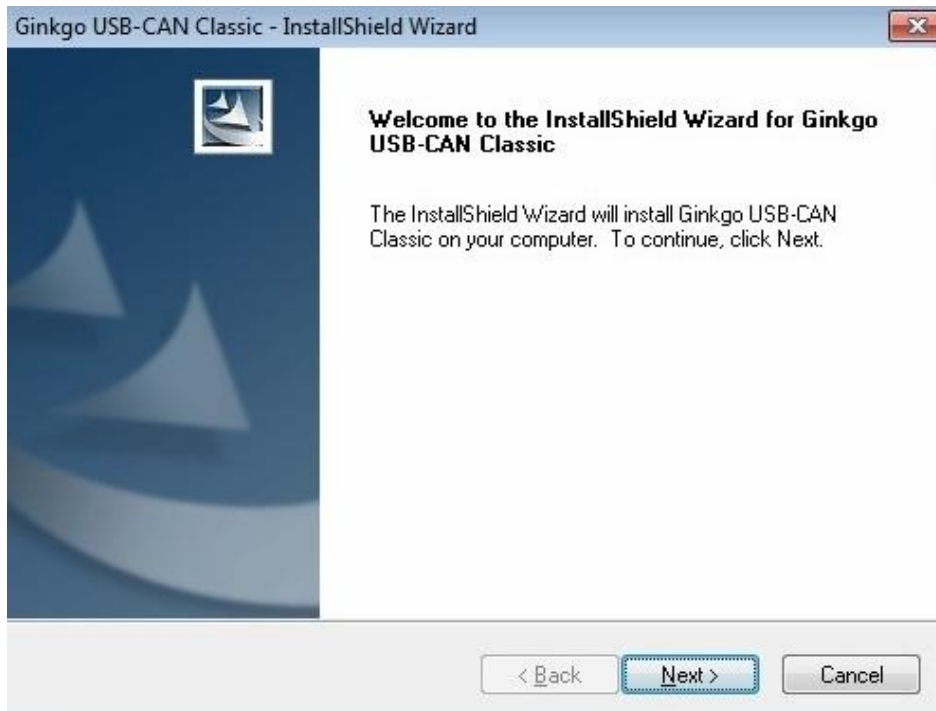
Download the software on the official website of ViewTool: [www.viewtool.com](http://www.viewtool.com), or [click here](#) to download.

### 3.1.2 Software Install

STEP 1. Unzip the RAR to Local, double click the executable file, dialog box pops up for language selection. Select the language and click “**Next**”.



STEP 2. Click “**Next**” directly.



STEP 3. Select Accept License Agreement, and click “Next”.

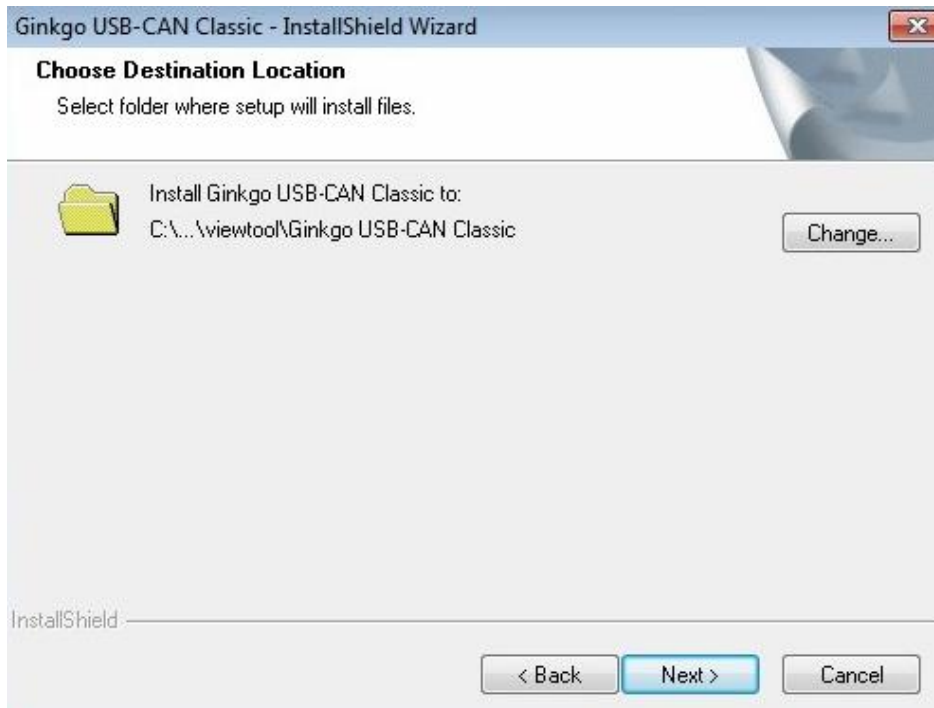


STEP 4. Enter your name and company name, click “Next”.

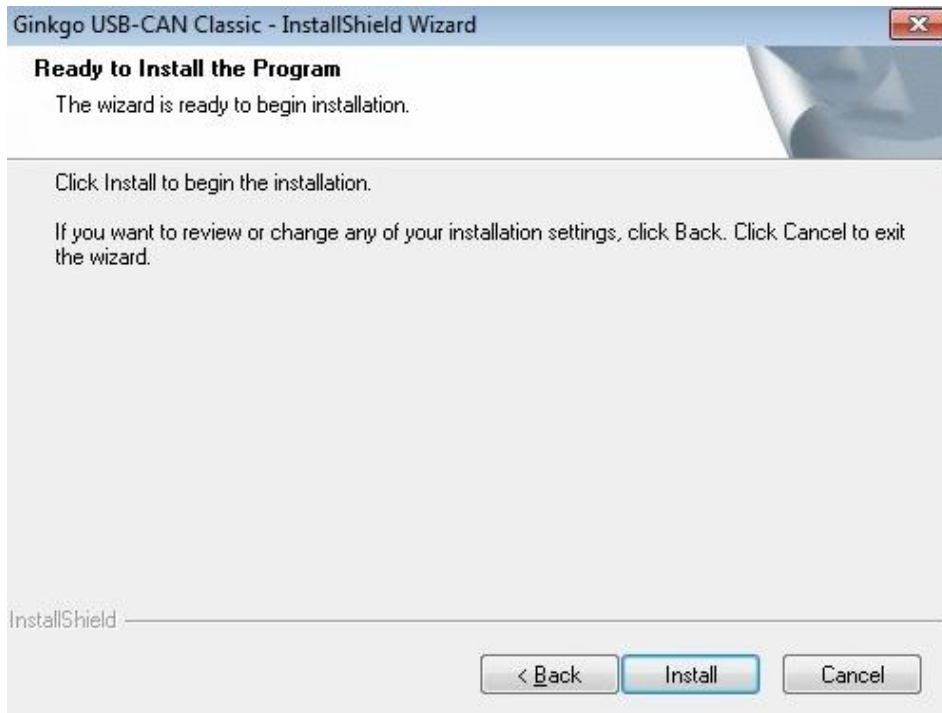




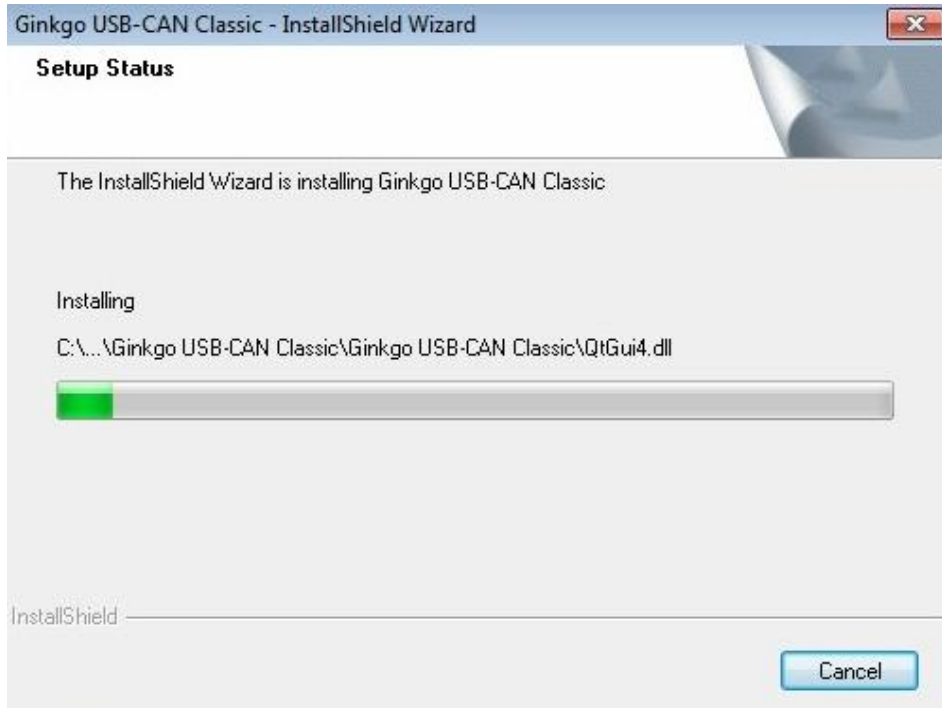
STEP 5. Select installation destination location according to your needs, click “**Next**”.



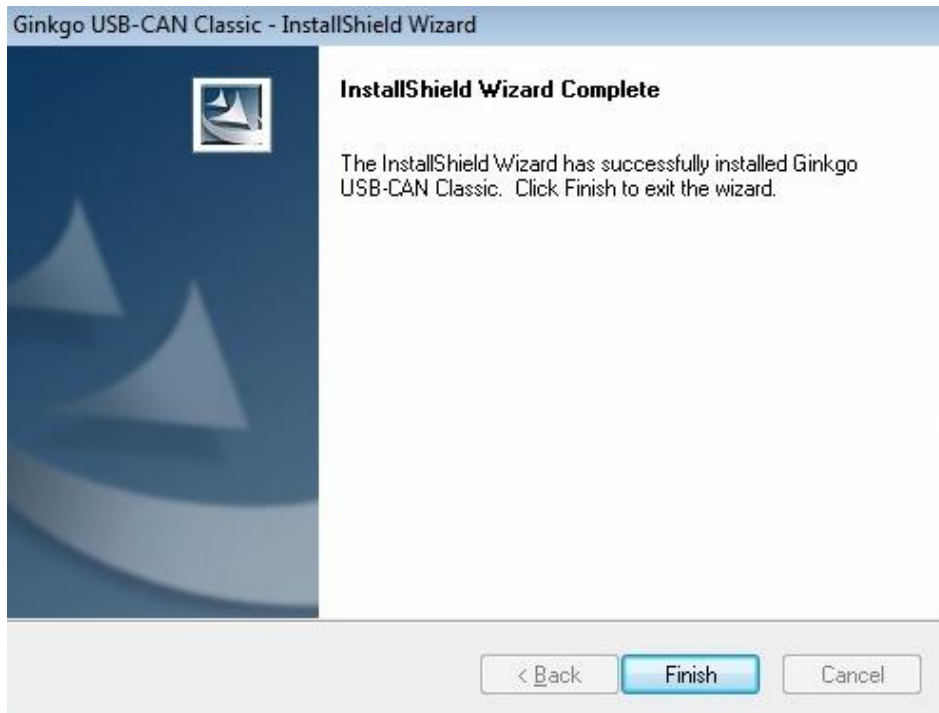
STEP 6. Click “**Install**” to begin installation.



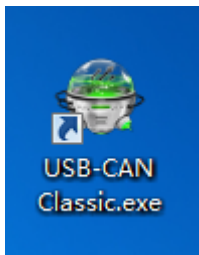
STEP 7. Installing, please wait.



STEP 8. Click "Finish" to exist the wizard.



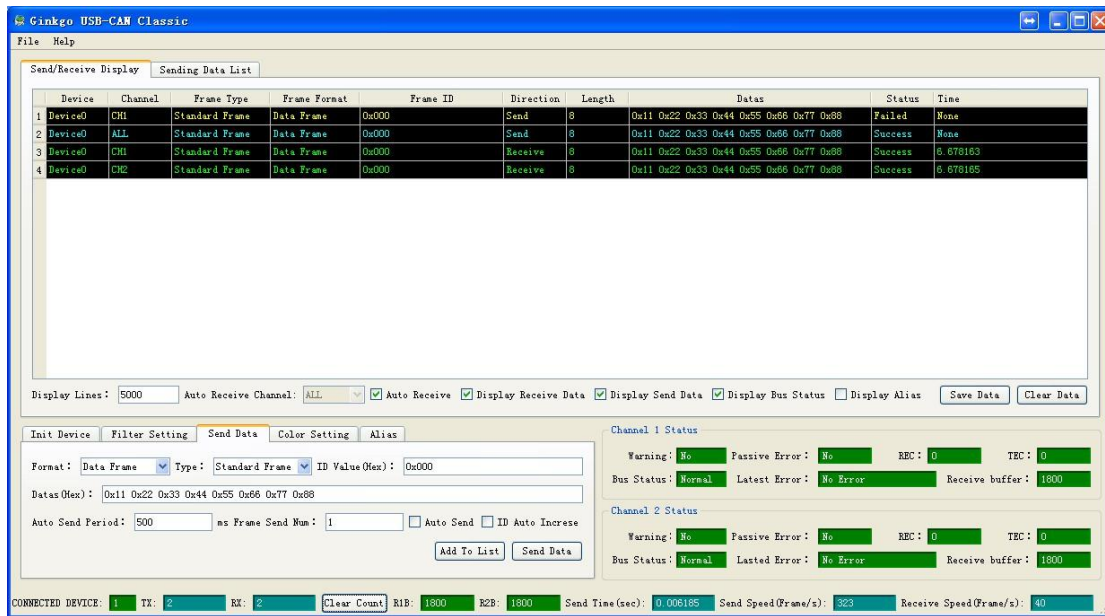
Program installation finished, and an icon as below is shown on the computer desktop.



## 3.2 Software Operation Instruction

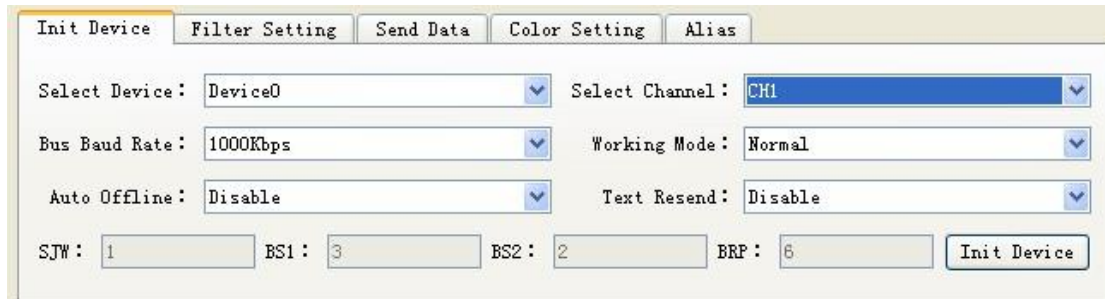
### 3.2.1 Software Introduction

Ginkgo USB-CAN Classic software is general purpose software for Ginkgo USB-CAN adapter. It can be used to control Ginkgo USB-CAN adapter data reading and writing. The main interface of the software is shown as below:



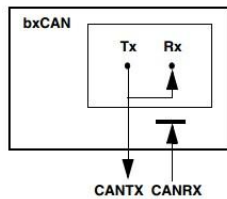
### 3.2.2 Initialization Settings

Initialize Ginkgo USB-CAN adapter before reading and writing data to CAN bus via Ginkgo USB-CAN adapter. Initialization interface of USB-CAN Classic software is shown as below.

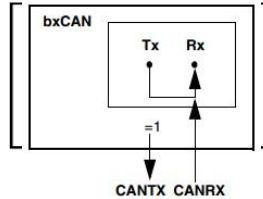


- **Select Device:** software counts the number of adapter connected to PC automatically and adapter connected are listed in the drop-down box. Select the device. The number of device connected is shown at the left bottom.
- **Select Channel:** select CAN channel that to be used. To operate two channels simultaneously, select All.
- **Bus Baud Rate:** 2kbps~1000kbps selectable, or user-defined.
- **SJW** (setting value: 1~4), upper limit of time unit to be extend or shorten in each bit.  
**BS1** (setting value: 1~16), define sampling location.  
**BS2** (setting value: 1~8), define sending point location.  
**BRP** (setting value: 1~1024), baud rate frequency divider.  
**CAN baud rate** = 36MHz / (BRP) / (SJW + BS1 + BS2)
- **Working Mode:**
  - 1) Normal: receive and send text normally;

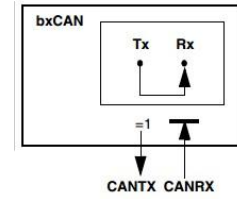
- 2) Look Back mode: adapter sends the self-test message to the bus, but cannot receive data from bus;
- 3) Silent mode: in this mode, adapter can receive data on the bus normally but cannot send message;
- 4) Look back combined with silent mode: this mode can be used for a “Hot Selftest”, which means the CAN can be tested like in Loop Back mode but without affecting bus data sending and receiving.



Look Back mode



Silent mode

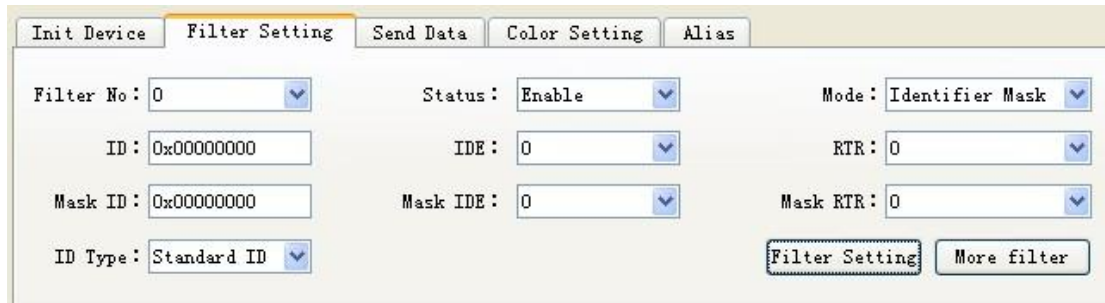


Look back combined with silent mode

- **Auto Offline:** if Enable is selected, adapter exits auto offline status once it detects 11 recessive bit for 128 times.
- **Text Resend:** if Enable is selected, system resends text continuously until success in case of text sending failure. If Disable is selected, text is sent only once no matter whether the text is sent successfully or not. If resend fails all the time, overtime error would be returned for sending timeout.

### 3.2.3 Filter Setting

Set data receiving conditions via Filter Setting based on the real application needs. Adapter can only get the data on the bus that meets the conditions. Filter setting has two modes: Identifier List mode and Identifier Mask mode. Filter setting interface is shown as below.



- **Identifier List mode:**

In Identifier List mode, adapter can receive text only when the text identifier on the bus keeps the same as the identifier set here.

For instance, provided acceptance ID is 0x35, Mask IDE (Identifier Extension Bit) is 1,

Mask RTR (Remote Transmission Request) is 0, then only the extended data frame with a identifier of 0x35 can go through filter and be received by adapter. Any other text cannot.

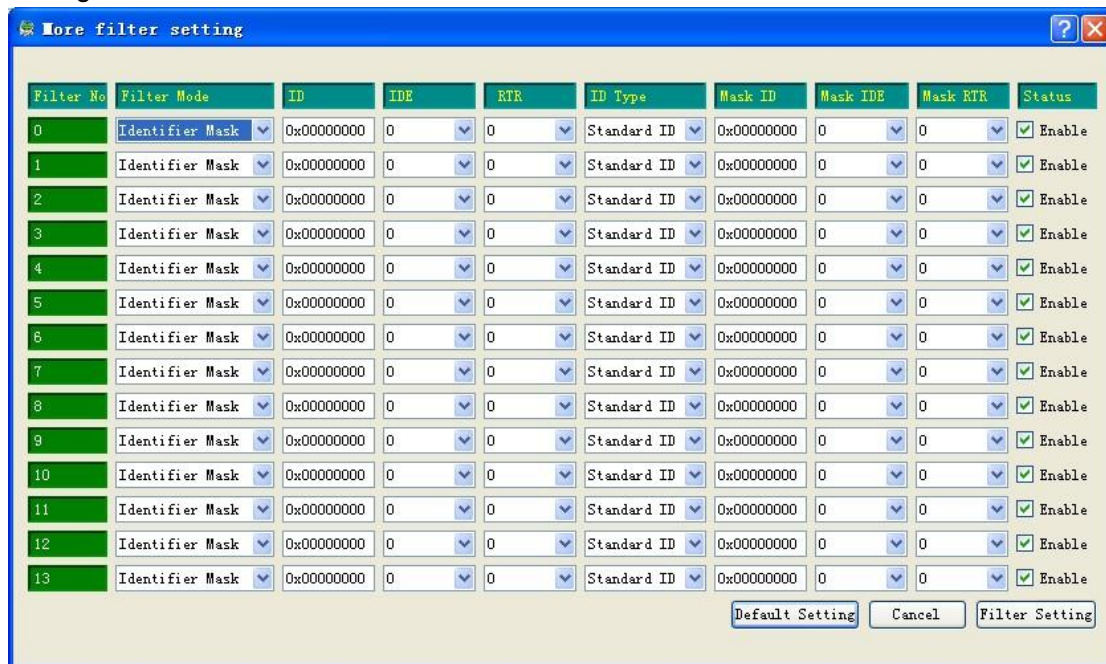
### ● Identifier Mask mode

In Identifier Mask mode, the user can appoint which bit with what value can be seen as transited. That is to say only identifiers within a certian defined range can be transitted. If Mask bit is “1”, the CAN frame of identifier to be accepted must be same as that of acceptance code. If Mask bit is “0”, then no matter same or not, the identifier can be transitted.

For instance, provided Mask ID is 1111 1111 101 (11 bits), Mask IDE is 1, Mask RTR is 0; Acceptance code ID is 0000 0000 001 (11 bits), acceptance code IDE is 1, and acceptance RTR is 0.

Except the second bit (0) from the right side of Mask ID and Mask RTR (0), other bits are all 1. So only 4 group of CAN frame with identifier of 0000 0000 001 10, 0000 0000 001 11, 0000 0000 011 10, 0000 0000 011 11 can be transitted.

Each CAN channel of Ginkgo USB-CAN adapter can match with 14 filters, and each filter works independently. If more filters are required, click “**More Filter**” to set. More filter setting interface is shown as below.

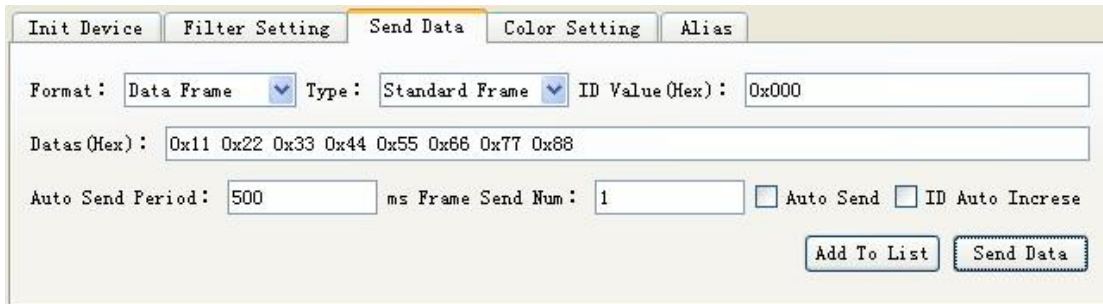


Filter No	Filter Mode	ID	IDE	RTR	ID Type	Mask ID	Mask IDE	Mask RTR	Status
0	Identifier Mask	0x00000000	0	0	Standard ID	0x00000000	0	0	Enable
1	Identifier Mask	0x00000000	0	0	Standard ID	0x00000000	0	0	Enable
2	Identifier Mask	0x00000000	0	0	Standard ID	0x00000000	0	0	Enable
3	Identifier Mask	0x00000000	0	0	Standard ID	0x00000000	0	0	Enable
4	Identifier Mask	0x00000000	0	0	Standard ID	0x00000000	0	0	Enable
5	Identifier Mask	0x00000000	0	0	Standard ID	0x00000000	0	0	Enable
6	Identifier Mask	0x00000000	0	0	Standard ID	0x00000000	0	0	Enable
7	Identifier Mask	0x00000000	0	0	Standard ID	0x00000000	0	0	Enable
8	Identifier Mask	0x00000000	0	0	Standard ID	0x00000000	0	0	Enable
9	Identifier Mask	0x00000000	0	0	Standard ID	0x00000000	0	0	Enable
10	Identifier Mask	0x00000000	0	0	Standard ID	0x00000000	0	0	Enable
11	Identifier Mask	0x00000000	0	0	Standard ID	0x00000000	0	0	Enable
12	Identifier Mask	0x00000000	0	0	Standard ID	0x00000000	0	0	Enable
13	Identifier Mask	0x00000000	0	0	Standard ID	0x00000000	0	0	Enable

After setting, click “**Filter Setting**” to ensure more filter setting works. [Click here](#) for more filters setting guide.

## 3.2.4 Send Data

Select data send format and type in Send Data interface. The selection interface is shown as below.

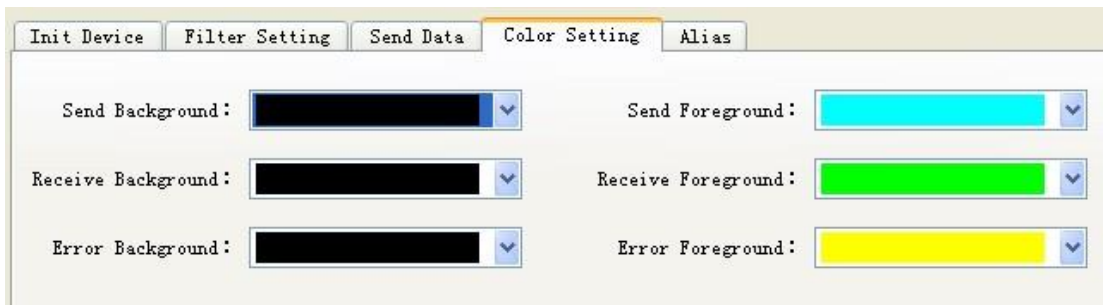


- **Format:** select the frame format of data to be sent with Data Frame and Remote Frame selectable;
- **Type:** select the frame type of data to be sent with Standard Frame and Extended Frame selectable;
- **ID Value (Hex):** when Standard Frame is selected, ID value range is 0~0x7FF; when Extended Frame is selected, ID value range is 0~0x1FFF FFFF;
- **Datas(Hex):** input hex data or int data with byte number less than or equal to 8.
- **Frame Send Num:** frame send number indicates the times this frame been sent after clicking “Send Data” button.
- **Auto Send:** when selected, data would be sent automatically in periodical time that has set in “Auto Send Period” box.
- **Add To List:** Click to add current frame to sending data list;

When auto send period time is set as 0, then fast data send mode is activated. In this mode, system do not check the data sending status, in case data sending failure, sending status is shown on bus.

### 3.2.5 Color Setting

In order to distinguish Send Data, Receive Data and Error Frame, different displaying foreground and background color are used. This software is featured with such kind of function, and the setting interface is shown as below.



- **Send Background:** background color of data sending;
- **Send Foreground:** foreground color of data sending;
- **Receive Background:** background color of data receiving;
- **Receive Foreground:** foreground color of data receiving;
- **Error Background:** background color of error frame;

- **Error Foreground:** foreground color of error frame;

Picture shown below is an example.

Device	Channel	Frame Type	Frame Format	Frame ID	Direction	Length	Datas	Status	Time
Device0	CH1	Standard Frame	Data Frame	0x000	Send	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Failed	None
Device0	ALL	Standard Frame	Data Frame	0x000	Send	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	None
Device0	CH1	Standard Frame	Data Frame	0x000	Receive	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	5.878183
Device0	CH2	Standard Frame	Data Frame	0x000	Receive	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	5.878185
Device0	CH1	Standard Frame	Data Frame	0x000	Send	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	None
Device0	CH1	Standard Frame	Data Frame	0x000	Receive	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	435.464827
Device0	CH1	Standard Frame	Data Frame	0x000	Send	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	None
Device0	CH1	Standard Frame	Data Frame	0x000	Receive	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	436.084782
Device0	CH1	Standard Frame	Data Frame	0x000	Send	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	None
Device0	CH1	Standard Frame	Data Frame	0x000	Receive	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	436.574739
Device0	CH1	Standard Frame	Data Frame	0x000	Send	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	None
Device0	CH1	Standard Frame	Data Frame	0x000	Receive	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	437.009715

Display Lines: 5000    Auto Receive Channel: ALL     Auto Receive     Display Receive Data     Display Send Data     Display Bus Status     Display Alias    Save Data    Clear Data

### 3.2.6 Alias Definition

Generally, each frame ID has a specific meaning in bus data analysis. The meaning of frame ID is displayed along with ID value in this software which is convenient for data analysis. Picture below is Alias setting interface.

ID	Alias
1 0x000	User Define Name
2 0x001	User Define Name 2
3 0x003	User Define Name 3

ID: 0x003    Alias: User Define Name 3    Add Alias    Delete Alias    Clear Alias

When received a frame with ID alias, software displays the specific alias along with frame ID value. For instance, if the alias of ID value 0x100 is set as "Motor Parameter", when the software received a frame whose ID value is 0x100, "0x100 [Motor Parameter]" would be shown on the interface. An example is given below.

Device	Channel	Frame Type	Frame Format	Frame ID	Direction	Length	Datas	Status	Time
Device0	CH1	Standard Frame	Data Frame	0x000[User Define Name]	Send	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	None
Device0	CH1	Standard Frame	Data Frame	0x000[User Define Name]	Receive	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	510.499340
Device0	CH1	Standard Frame	Data Frame	0x001[User Define Name 2]	Send	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	None
Device0	CH1	Standard Frame	Data Frame	0x001[User Define Name 2]	Receive	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	512.108222
Device0	CH1	Standard Frame	Data Frame	0x002	Send	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	None
Device0	CH1	Standard Frame	Data Frame	0x002	Receive	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	512.802171
Device0	CH1	Standard Frame	Data Frame	0x003[User Define Name 3]	Send	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	None
Device0	CH1	Standard Frame	Data Frame	0x003[User Define Name 3]	Receive	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	526.849159
Device0	CH1	Standard Frame	Data Frame	0x004	Send	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	None
Device0	CH1	Standard Frame	Data Frame	0x004	Receive	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	528.328036

Display Lines: 5000    Auto Receive Channel: ALL     Auto Receive     Display Receive Data     Display Send Data     Display Bus Status     Display Alias    Save Data    Clear Data

### 3.2.7 CAN Bus Status Display

Since unexpected things may happen for some certain reason during CAN bus data



receiving and sending such as data sending/receiving failure or adapter cannot work normally, it is required for CAN adapter to shown real time operation status.

- **Warning:** indicate whether adapter error exists or not. Error warning includes sending error warning and receiving error warning;
- **Passive Error:** indicate whether passive error exists or not;
- **REC:** indicate the counting value of adapter receiving error counter.
- **TEC:** indicate the counting value of adapter transmission error counter.
- **Bus Status:** indicate status of adapter and CAN bus connection: normal status or offline status;
- **Latest Error:** interpretation of the latest error of adapter;
- **Receive buffer:** shows the free space of adapter data receive buffer.

### 3.2.8 Data Receive

Data received is displayed in list form with Frame Type, Frame Format, Frame ID, Length, Datas and Time etc shown clearly. Displaying interface is shown as below.

Device	Channel	Frame Type	Frame Format	Frame ID	Direction	Length	Datas	Status	Time
Device0	ALL	Standard Frame	Data Frame	0x012	Send	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	None
Device0	CH1	Standard Frame	Data Frame	0x012	Receive	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	7.026138
Device0	CH2	Standard Frame	Data Frame	0x012	Receive	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	7.026138
Device0	ALL	Standard Frame	Data Frame	0x013	Send	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	None
Device0	CH1	Standard Frame	Data Frame	0x013	Receive	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	8.193049
Device0	CH2	Standard Frame	Data Frame	0x013	Receive	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	8.193051
Device0	ALL	Standard Frame	Data Frame	0x014	Send	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	None
Device0	CH2	Standard Frame	Data Frame	0x014	Receive	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	8.556026
Device0	CH1	Standard Frame	Data Frame	0x014	Receive	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	8.556024
Device0	ALL	Standard Frame	Data Frame	0x015	Send	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	None
Device0	CH1	Standard Frame	Data Frame	0x015	Receive	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	9.207977
Device0	CH2	Standard Frame	Data Frame	0x015	Receive	8	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	Success	9.207979

If adapter is required to receive data automatically, select “**Auto Receive**”. Set the display contents such as data sent or received, bus status, frame ID alias in display settings. Software works slowly when many contents are selected to display and data receive and send speed are relatively high. So please make a reasonable displaying setting when data is sent or received at a high speed.

Click “**Save Data**” button to save the data in the list to TXT or forms. Click “**Clear Data**” to clear all data in the list.

### 3.2.9 Data Sending in Identifier List Mode

Data in this software can also be sent in list format. Add multiple CAN frames into a list, and then send the data in turn. This function is very useful in realizing upper layer protocol or of that kind. Data sending interface in Identifier List mode is shown as below.

Device	Channel	Frame Type	Frame Format	Frame ID	Length	Datas
1 Device0	CH1	Data Frame	Standard Frame	0x016	0x08	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
2 Device0	CH2	Data Frame	Standard Frame	0x017	0x08	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
3 Device0	CH1	Data Frame	Standard Frame	0x018	0x08	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
4 Device0	CH2	Data Frame	Standard Frame	0x019	0x08	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
5 Device0	CH1	Data Frame	Standard Frame	0x01A	0x08	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
6 Device0	CH2	Data Frame	Standard Frame	0x01B	0x08	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
7 Device0	CH1	Data Frame	Standard Frame	0x01C	0x08	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
8 Device0	CH2	Data Frame	Standard Frame	0x01D	0x08	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
9 Device0	CH1	Data Frame	Standard Frame	0x01E	0x08	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
10 Device0	CH2	Data Frame	Standard Frame	0x01F	0x08	0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88

Auto Send Period: 1000 ms  Auto Send

Open File Save File Send Data Delete Data Clear Data

- **Auto Send:** send data periodically according to the time set in “Auto Send Period” box;
- **Open File:** open the well-edited list file. Please refer to file save format for file format;
- **Save File:** save data in the list to files. It is convenient to use for the next time by opening the file directly;
- **Send Data:** send data in the list via adapter CAN interface.
- **Delete Data:** delete data line selected.
- **Clear Data:** clear all data in the list.

### 3.2.10 Adapter Status Display

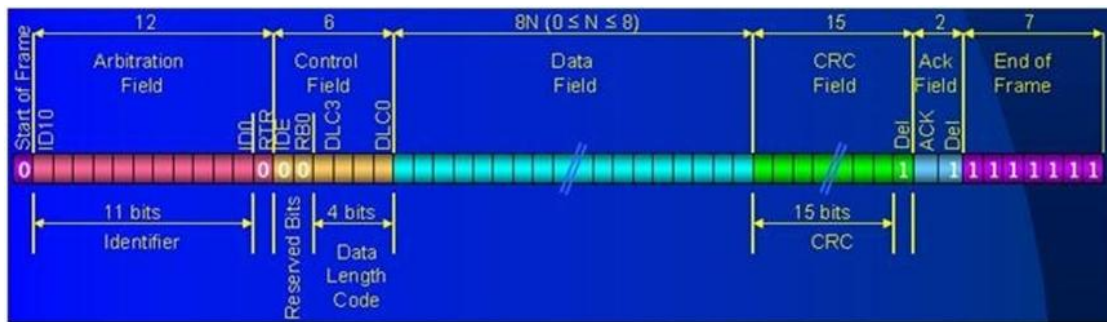
Real time operation status of software is shown in the status bar which is shown as below.

CONNECTED DEVICE: 1	TX: 28	RX: 19	Clear Count	R1B: 1800	R2B: 1800	Send Time(sec): 0.004184	Send Speed(Frame/s): 478	Receive Speed(Frame/s): 40
---------------------	--------	--------	-------------	-----------	-----------	--------------------------	--------------------------	----------------------------

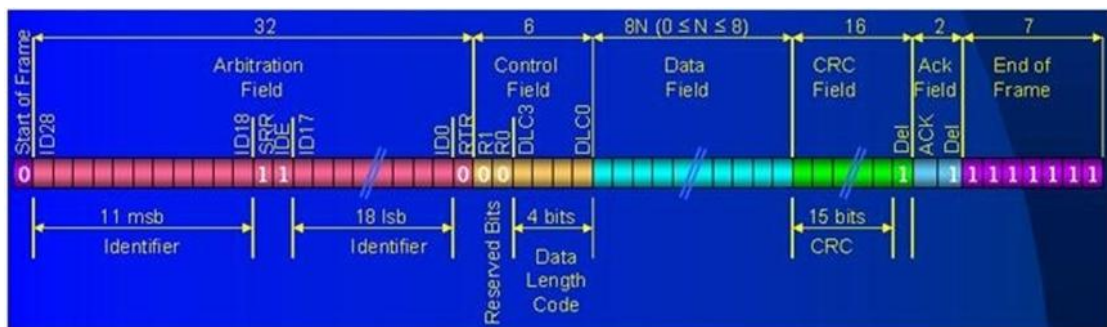
- **CONNECTED DEVICE:** the number of USB-CAN adapter devices connected to PC at present;
- **TX:** CAN frame number that software has sent successfully;
- **RX:** CAN frame number that software has received successfully;
- **Clear Count:** clear all frame sent and received, i.e. the value of “TX” and “RX”;
- **R1B:** free space of channel 1 receiving buffer;
- **R2B:** free space of channel 2 receiving buffer;
- **Send Time (sec):** time spend on data sending;
- **Send Speed (Frame/s):** data sending speed at present;
- **Receive Speed (Frame/s):** data receiving speed at present;

# Chapter 4 CAN bus Frame Format

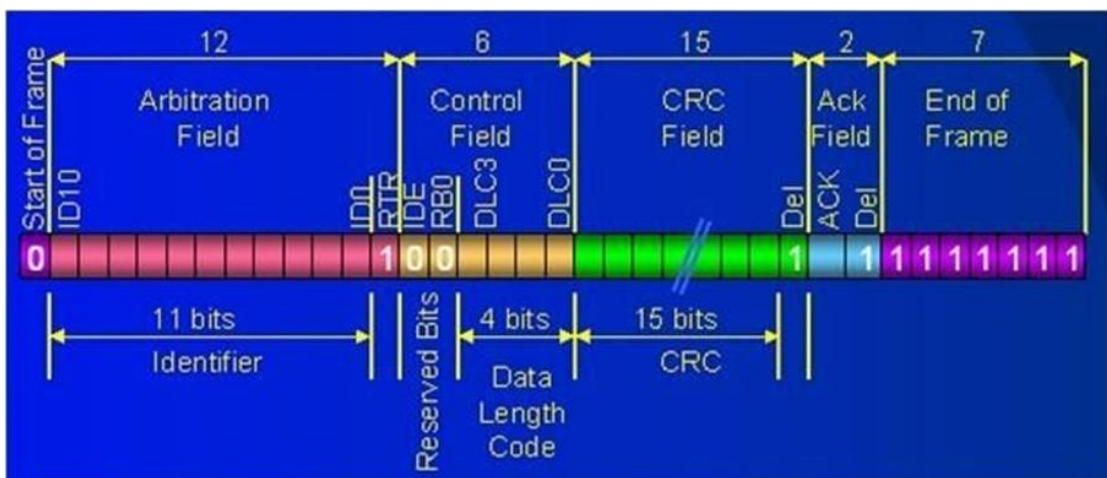
## 4.1 Standard Data Frame Structure



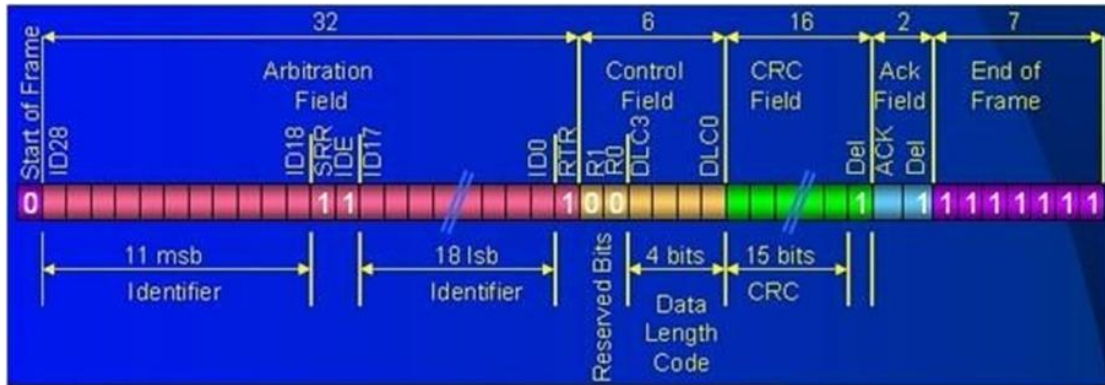
## 4.2 Extended Data Frame Structure



## 4.3 Standard Remote Frame Structure



## 4.4 Extended Remote Frame Structure



## Technical Support and Service

- Lifetime technical support, free services and maintenance within one year.
- Download driver, software, User's Manual and application routine source code at [www.viewtool.com](http://www.viewtool.com).
- Technical exchange forum: [www.viewtool.com/bbs](http://www.viewtool.com/bbs)
- Technical support Email: [FAE@viewtool.com](mailto:FAE@viewtool.com)
- Sales Email: [Sales@viewtool.com](mailto:Sales@viewtool.com)