

Westpac Technology

Tomorrow's CAN Technology for you Today!

CAN3 Writer

User Manual v1.05



Installation
✧
Quick Start Guide

Preliminary

Table of Contents

1. PREFACE	3
2. INTRODUCTION	3
3. TERMS AND DEFINITION	3
4. SYSTEM REQUIREMENT AND INSTALLATION	3
4.1. SYSTEM REQUIREMENT	3
4.2. OPERATING SYSTEM REQUIREMENT	3
4.3. INSTALLATION	4
5. LOAD CAN MESSAGE DATABASE	4
6. IMPORT DBC FORMATTED FILE.....	5
7. PARAMETER CAN NODE MESSAGE DATA FILE (INI).....	6
7.1. NEW PARAMETER DATA FILE	6
7.2. OPEN EXISTING PARAMETER DATA FILE	6
8. EDITING A NEW INI PROJECT FILE	6
8.1. CAN MESSAGE DATA FORMAT	6
8.1.1. <i>Motorola Forward Isb</i>	7
8.1.2. <i>Motorola Backward Isb</i>	7
8.1.3. <i>Intel Standard</i>	7
9. EDITING & MAINTENANCE SCREEN	8
9.1. PROJECT TITLE	9
9.2. CAN NODE MESSAGE HEADER	9
9.2.1. <i>Add/Insert a new CAN Node Message</i>	9
9.3. CAN NODE MESSAGE DATA ENTRY AND EDITING	9
9.4. OPERATIONAL/CONTROL DATA SIGNAL TYPE	11
9.4.1. <i>Type V- Value</i>	11
9.4.2. <i>Type F- Formulae</i>	12
9.4.3. <i>Type B- Bitwise</i>	13
9.4.4. <i>Type AC- Alive Counter</i>	13
9.4.5. <i>Type RC- Rolling Counter</i>	14
9.4.6. <i>Type CS- Message Frame Checksums</i>	15
9.4.7. <i>Type G- General</i>	15
9.4.8. <i>Type S- Spare</i>	16
10. PARALLEL UNICODE DBC FILE	16
10.1. IMPORTING THE DBC WITH CHINESE TEXT INTO WORD DOCUMENT	16
10.2. IMPORT (PASTE) THE CHINESE SIMPLIFIED –GB2312 CONTENTS INTO NOTEPAD	17
10.3. GENERATE/SAVE THE UNIDBC FILE IN UNICODE ENCODING	18
REVISION HISTORY	20

This document is a property of Westpac Technology, Westpac Technology reserved the rights to change the contents of this document without prior notice, for further information, please contact us at www.westpactech.com

1. Preface

This document describes the editing and maintenance of CAN message list that can be used in **CAN3 Simulator** for Message Simulation, as well as in **CAN3 Monitor** for Message recording & Monitoring.

2. Introduction

This **CAN3 writer** for CAN Node Message editing is an integral part of the **CAN3 package** to assist in the design and implementation of ECU connecting to the CAN Network.

Reading the data bits or bytes in raw 64 binary bits of CAN message frame without meaningful translation is tedious and not practical; nor productive. Therefore, a translation media is necessary, allowing each CAN data can be read in a way that can be understood simply by a human user/operator.

The ideology of allowing users to interact the CAN message binary data with meaningful translation manner as described within this document is not limited to the naming of data, but is extended to provide simulation controls- include classification of data type- whereby, formula and data patterns are possible, henceforth allowing the translation of each data with complete visual experience- instantaneously be displayed on the screen vis-à-vis the name of the data, formula used and the resultant of the data.

This complete online data translation of CAN message binary bits would greatly simplify the development cycles, helping product developer to save time, increase in productivity and increase product quality.

3. Terms and definition

- ☐ CAN: Controller Area Network
- ☐ Numerals: Hexadecimal value will be represented with a “h” after the hexadecimal number or “0x” in front of the hexadecimal number (Motorola format); a number without “h” or “0x” will be treated as decimal number
- ☐ ECU: Electronic Control Unit

4. System Requirement and Installation

4.1. System Requirement

- ☐ CPU Speed: Pentium 1.6GHz minimum or equivalent or higher.
- ☐ Memory: 512MB Memory or above

4.2. Operating System Requirement

The **CAN3 Writer** supports the followings Operating System:

- ☐ Windows 2000
- ☐ Windows XP 32-bits
- ☐ Windows Vista 32-bits
- ☐ Windows 7 32-bits

CAN3 Writer requires Microsoft .NET Framework Version 2.0 be installed.

For more information about this, you can visit
<http://www.microsoft.com/downloads/en/default.aspx>

Search for the key words .

- **NET Framework 3.5 SP1**
- **Microsoft .Net Framework4 (x86 and x64) and update for .Net Framework 4 (KB2468871)**

4.3. Installation

To install **CAN3 Writer** simply clicks **Setup.exe** and follows the instruction given on the screen.

The **CAN3 Writer** will be installed into the **Start Menu** under the **CAN3 Utility** folder.

5. Load CAN Message Database

Upon the activation of t this **CAN 3 Writer**, an option menu will be prompted on the screen to select the following databases – Figure 1.

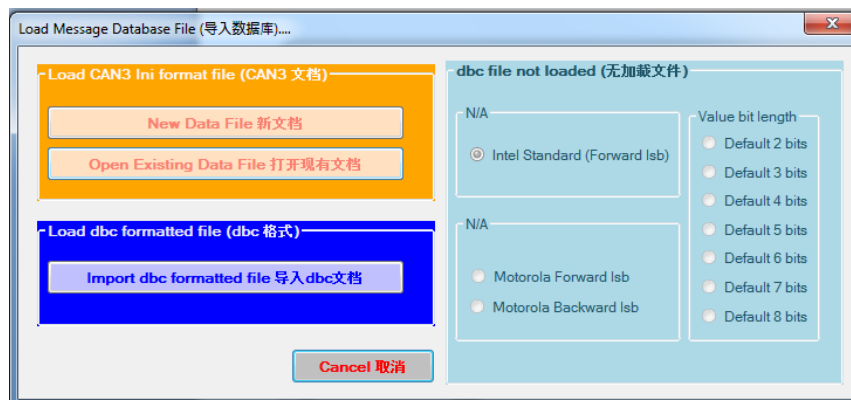


Figure 1: Load CAN message database options

- Option 1: TO load CAN3 Ini Formatted file- see section 7
- Option 2: Load dbc formatted file

Note: You can open the dialog at any time via normal file Open option as per the picture shown below:

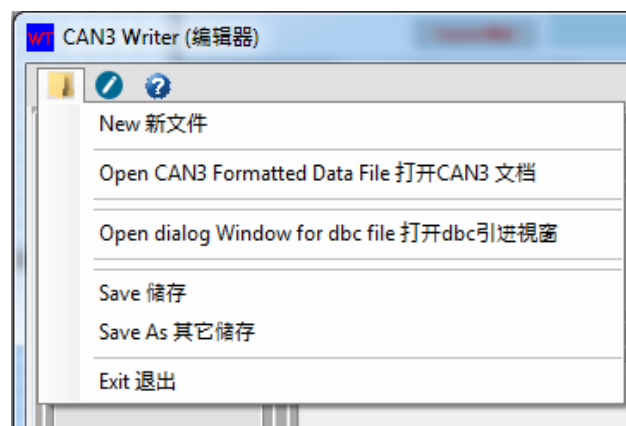


Figure 2: Select New or existing file for editing or import a dbc file

6. Import dbc formatted File

The contents of a dbc formatted file evolves regularly and CAN3 Writer will try its very best to translate it into CAN3 INI format. If you encounter difficulties to convert the dbc formatted file to CAN3 Init format, please contact us and we shall endeavour to investigate the matter and we shall try to fix it with our best ability.



Figure 3: Import dbc formatted file option

To import a dbc formatted file, click the “Import dbc formatted file” button and follows the screen dialog to retrieve the dbc file; after that the control panel as shown in Figure 4 will be enabled with the following option Radio button enabled i.e.

- In this example, the Intel format is detected. Please note that only one format in the dbc is allowed, mixture of Intel & Motorola with one dbc file is not supported
- Value bit length- in this example it detected the value data and sets the Default length to 4 bits long, but you can adjust the default length to suit your needs.

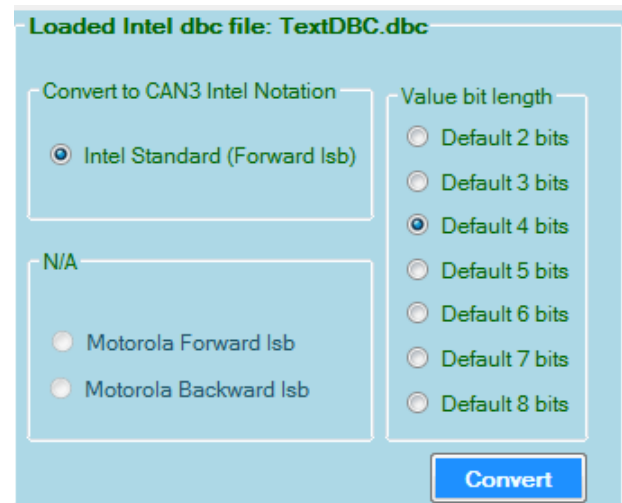


Figure 4: Control screen for converting dbc to INI format

Note: To import a dbc file that has Chinese Characters, you need to duplicate the dbc file into Unicode encoding format file with extension “.unidbc” i.e. Filename.unidbc. See section 10 Parallel Unicode dbc file for more details.

When ready, click the “Convert” button, if the conversion is without error, it will enter into the CAN3 Writer’s editing screen, and otherwise a warning message will be shown. Figure 5 shows a warning message, informing that message ID 0x291 has the Transmit time data missing and has set the interval time to 100ms, but this value can be changed in the CAN3 Writer’s editing screen

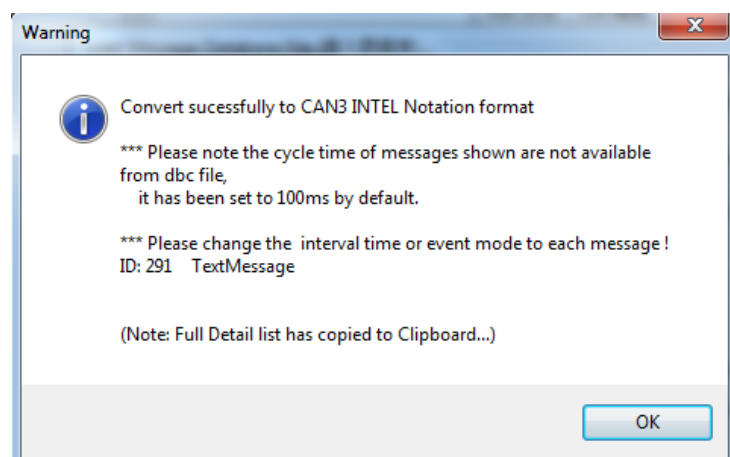


Figure 5: Warning message from the dbc conversion

7. Parameter CAN Node Message Data File (INI)

This INI file contains the translation database for all CAN Node Messages' transmit and receive (by **CAN3 Simulator** or **CAN3 Monitor**) on common layer of CAN network.



Figure 6: INI file open option

The option will be prompted on the screen for selection- see Figure 6, you can also activate this option at any time of the control as described in Figure 2 of section 5.

7.1. New Parameter Data File

To create INI file for a new project

7.2. Open exiting Parameter Data File

Re-open an existing INI data file for editing and maintenance

8. Editing a new INI project file

Upon selecting this option for editing a new INI project file, the Window will prompt you to enter the project file name. For the following discussion- file name "CAN message Demo.INI" will be used.

The maximum data size of CAN 2.0 is 64 bits; thus to address a particular data bit; arranged in the 64 bit block requires some sort of data notation to describe it. **CAN3 Writer** supports the following data notation:

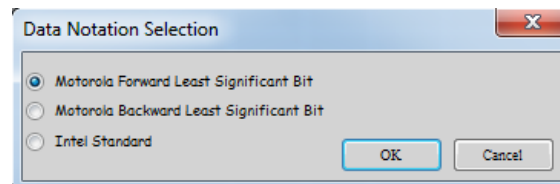


Figure 7: Data format selection

- ☐ Motorola Forward Least Significant Bit or Motorola Forward lsb
- ☐ Motorola Backward Least Significant Bit or Motorola Backward lsb
- ☐ Intel Standard

8.1. CAN Message Data Format

After the file name has entered, you will be asked to enter the message data format for the entire project- like Figure 7.

8.1.1. Motorola Forward Isb

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	7	6	5	4	3	2	1	0
	1	15	14	13	12	11	10	9	8
	2	23	22	21	msb 20	b11 19	b10 18	b9 17	b8 16
	3	←b7 31	b6 30	b5 29	b4 28	b3 27	b2 26	b1 25	lsb 24
	4	39	38	37	36	35	34	33	32
	5	47	46	45	44	43	42	41	40
	6	55	54	53	52	51	50	49	48
	7	63	62	61	60	59	58	57	56

Signal index for 13 bit data is (lsb, length)= (24, 13)

8.1.2. Motorola Backward Isb

		Bit							
		7	6	5	4	3	2	1	0
Byte	7	63	62	61	60	59	58	57	56
	6	55	54	53	52	51	50	49	48
	5	47	46	45	msb 44	b11 43	b10 42	b9 41	b8 40
	4	←b7 39	b6 38	b5 37	b4 36	b3 35	b2 34	b1 33	lsb 32
	3	31	30	29	28	27	26	25	24
	2	23	22	21	20	19	18	17	16
	1	15	14	13	12	11	10	9	8
	0	7	6	5	4	3	2	1	0

Signal index for 13 bit data is (lsb, length)= (32, 13)

8.1.3. Intel Standard

		Bit							
		7	6	5	4	3	2	1	0
Byte	0	7	6	5	4	3	2	1	0
	1	15	14	13	12	11	10	9	8
	2	←b7 23	b6 22	b5 21	b4 20	b3 19	b2 18	b1 17	lsb 16
	3	15	14	13	msb 28	b11 27	b10 26	b9 25	b8 24
	4	39	38	37	36	35	34	33	32
	5	47	46	45	44	43	42	41	40
	6	55	54	53	52	51	50	49	48
	7	63	62	61	60	59	58	57	56

Signal index for 13 bit data is (lsb, length)= (16, 13)

Note: For discussion purpose, this document will use Motorola Forward Isb format.

9. Editing & Maintenance screen

Figure 8 & Figure 9 below respectively shows the editing screen of a new and existing CAN messaging project.

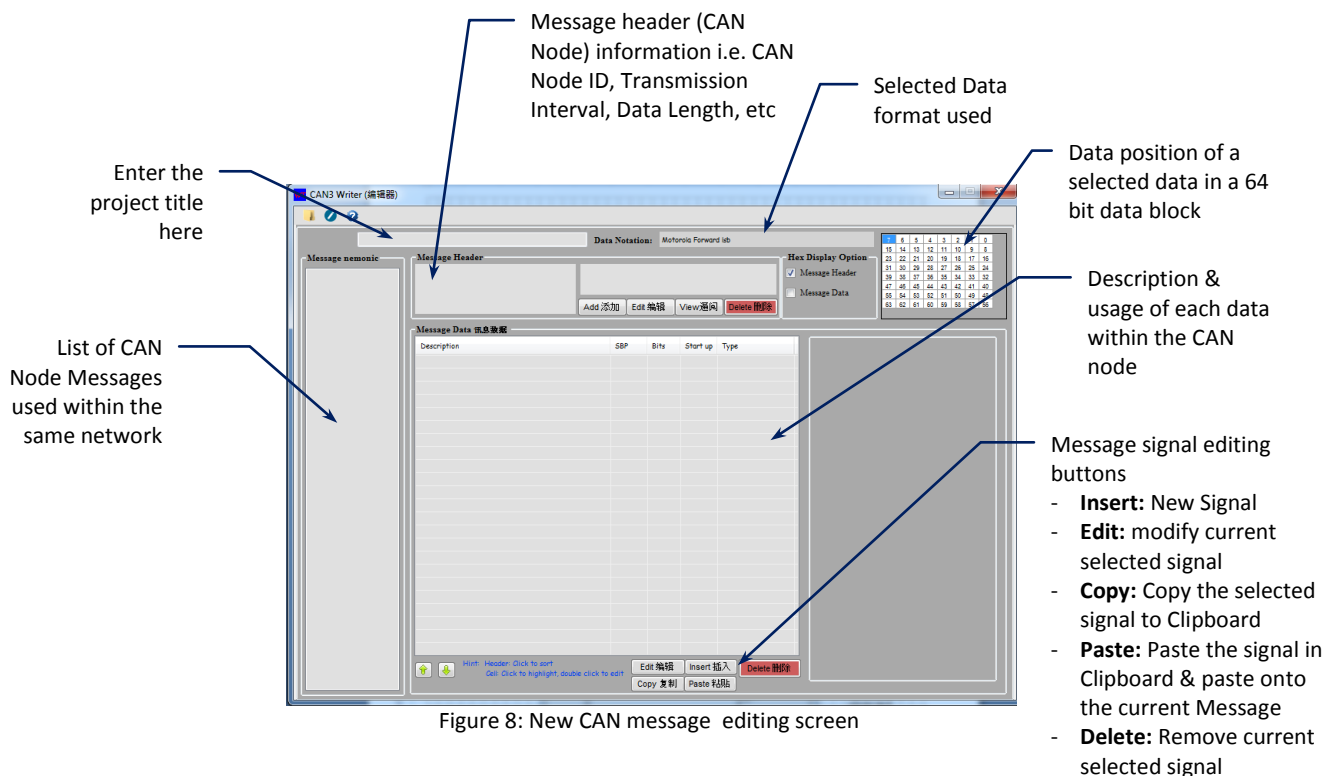


Figure 8: New CAN message editing screen

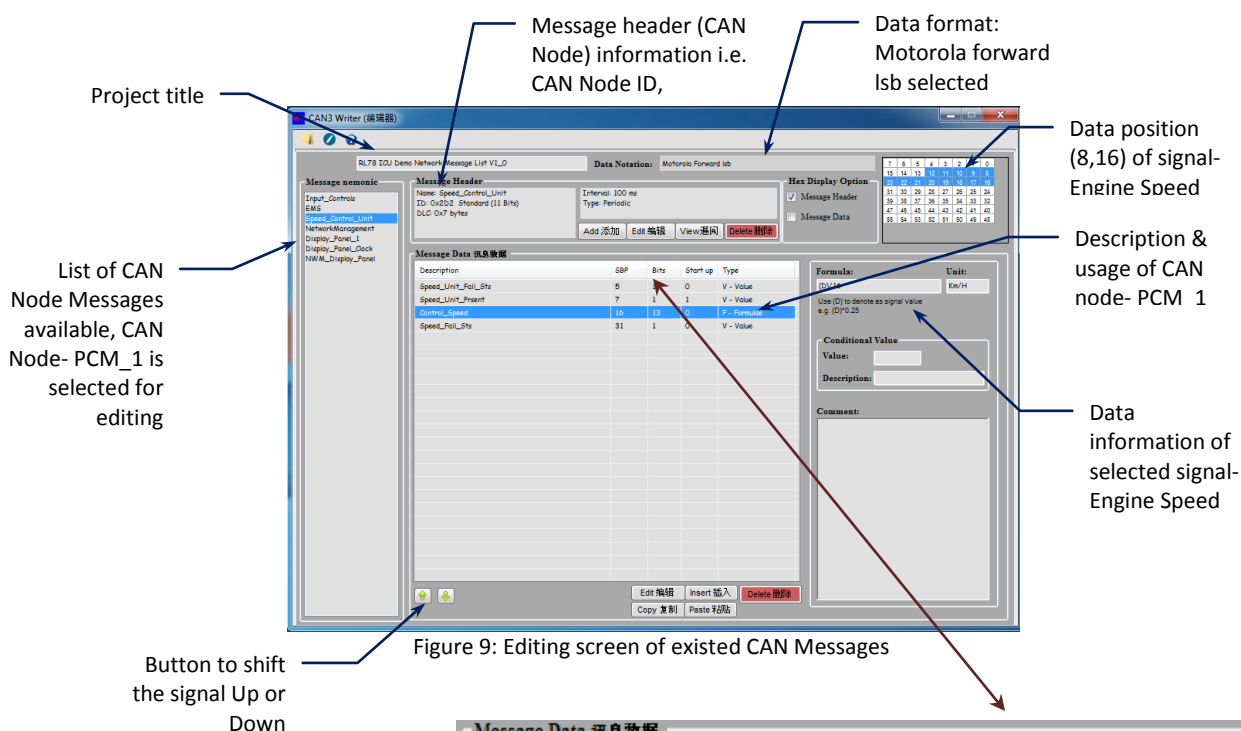


Figure 9: Editing screen of existed CAN Messages

Message Data 消息数据				
Description	SBP	Bits	Start up	Type

Hint: Click the any one of the header section to sort the signal to order sequence

9.1. Project title

Don't forget to enter the project title; it is very important for reference and maintenance purpose

9.2. CAN Node Message Header

Figure 10 shows the Message Header of a blank new project

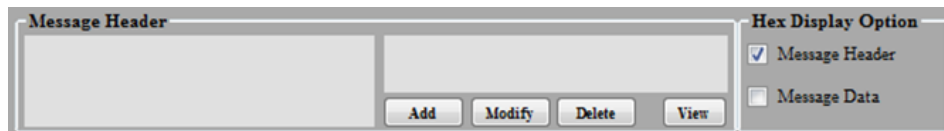


Figure 10: CAN Node Message Header

Following options are available:

- Add: To add a new CAN Node Message
- Modify: Make changes to a selected CAN Node Message
- Delete: To remove the selected CAN Node Message from the project
- View: Summary data view of the selected CAN Node Message

9.2.1. Add/Insert a new CAN Node Message

Figure 11 shows a new CAN Node Message Header for editing. It comprises of the following:

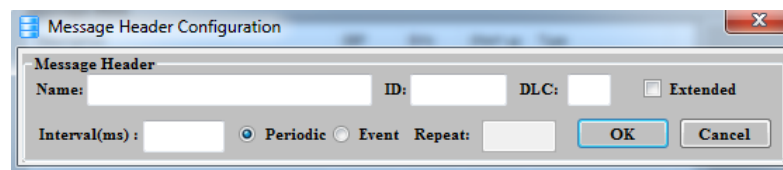


Figure 11: Editing a new CAN Node Message Header

- ☐ Name: Description or name of the CAN Node Message
- ☐ ID: 11 bit or 24 bit CAN Node ID
- ☐ DLC: Length of this CAN Node Message; Maximum 8 bytes
- ☐ Extended: Check box for 24 bit (extended) CAN Node Message ID
- ☐ Interval: Time in mili-second between transmit messages
- ☐ Periodic/Event option: For Periodic option, the CAN message will be transmitted on a set intervals; while for Event option, it works with the Repeat count, whereby when this CAN Node Message is Checked for transmission enabled in **CAN3 Simulator**, it will repeat the CAN Message transmission with number of Repeats at the set Interval Time.

9.3. CAN Node Message Data entry and editing

For the following illustration, let us assume the following CAN Node Message (Figure 12) has been added in section 9.2.1:

- ☐ CAN Node Message: PCM_1
- ☐ ID: 0x20 (Standard 11 bit addressing)
- ☐ Data Length: 7 bytes (56 bits)
- ☐ Transmission Interval: Periodic 10ms

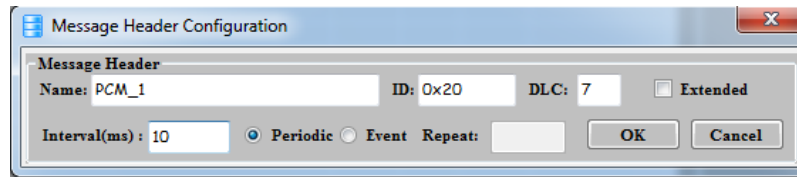
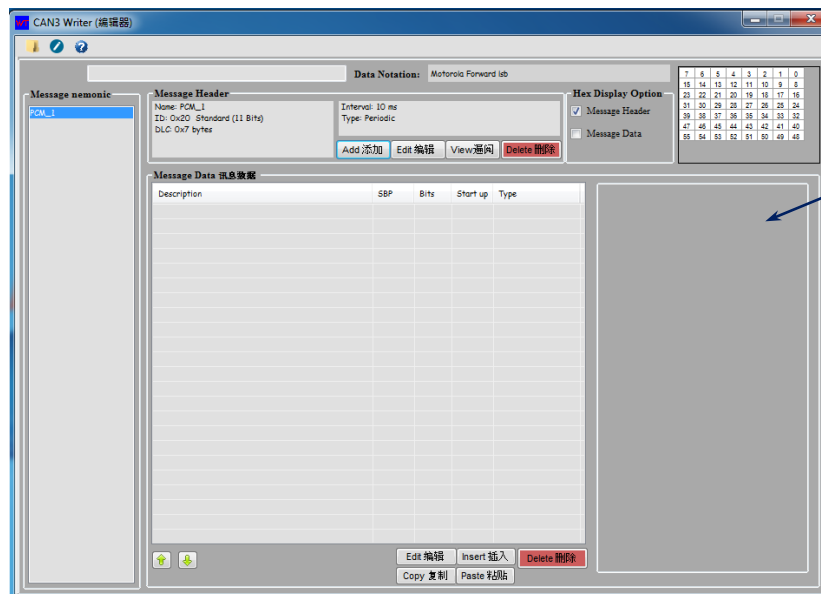


Figure 12 : Message Header entry

Figure 13 below shows the screen after PCM_1 has added.



This area will show the operational options that is in accordance to the selected data type.

Figure 13: Editing CAN Node Message- PCM_1

Now, the screen is ready to Insert data signal to the CAN Node Message PCM_1; click “Insert” button to insert a data signal.

Enter the following as example:

- ☐ Description: Engine Speed (RPM)
- ☐ SBP (start Bit Position or lsb): 8
- ☐ Bits (Length): 16
- ☐ Start up: 0
- ☐ Type : Select “F” for formulae and the available options for formula type will be shown- See Figure 14 for reference. For other available type, please refer to section 9.4 for more details.

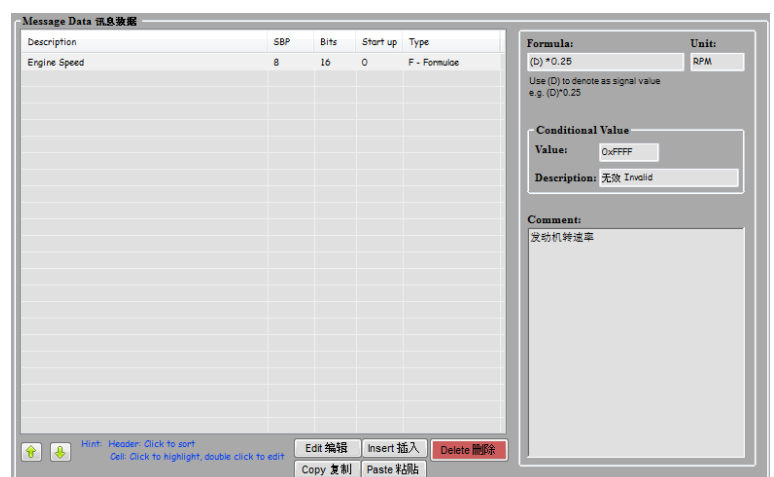


Figure 14: Operational parameter for Formula type

Repeat the above sections 9.2 & 9.3 until all CAN Node Messages and their associates’ data signals are entered.

9.4. Operational/Control Data Signal type

In CAN simulation mode, if the data cannot be changed, it is as good as a dead signal, and in order to make the data as a **Live Signal**; it needs to be changed in a manner that is meaningful and easy to control.

Generally, simulated data is done with custom build software or scripting routines to allow online changes to the data value; but this method is cumbersome and requires programming skill to do so. To overcome this restriction, **CAN3 writer** provides 7 control types for you to define the data signal, so that the **CAN3 Simulator** utility will provide you with the necessary controls for live data signal simulation controls.

The 7 control types are as follows:

- ☐ V- Value: To assign description to the data value i.e. Gear positions
- ☐ F-Formulae: Enter the formula for converting the raw value to the actual meaningful value
- ☐ B-Bitwise: Enter the description of the bit representation of the function
- ☐ AC-Alive Counter: This option type is for the **CAN3 Simulator** to auto-increment this value (with a preset value) from 0 to its maximum count value & then repeat from 0 at the set interval time
- ☐ RC-Rolling Counter: This option type is for the **CAN3 Simulator** to add the incremental value to the interval message, the receiver will then extract the difference between the two messages to obtain the count values; typical application is the Fuel Pulses, Wheel Pulses, etc
- ☐ G-General: No specific description, just raw binary data
- ☐ S-Spare: Not used data; no specific description, just raw binary data

9.4.1. Type V- Value

Figure 15 shows the editing control for type-V.

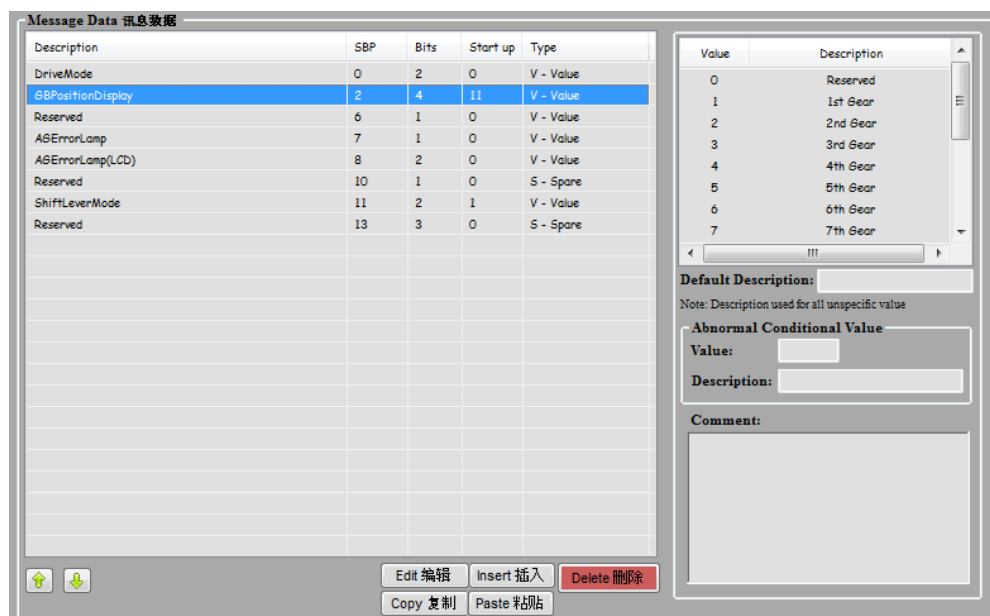


Figure 15: Editing screen for control type -V

In this example, the length of data signal “GBPositionDisplay” is 4 bits long, thus this value has 16 possible combinations, as such, the Value-Description editing box provides all 16 rows for editing as shown.

In addition, there are 3 optional sections available to describe the Data Signal during Simulation and Monitoring:

- ☐ Default Description: This is used for any value that have the description field left blank
- ☐ Abnormal Condition Value: To high light an important value
- ☐ Comments: General note of the data signal

9.4.2. Type F- Formulae

Figure 16 shows the editing control for type-F.

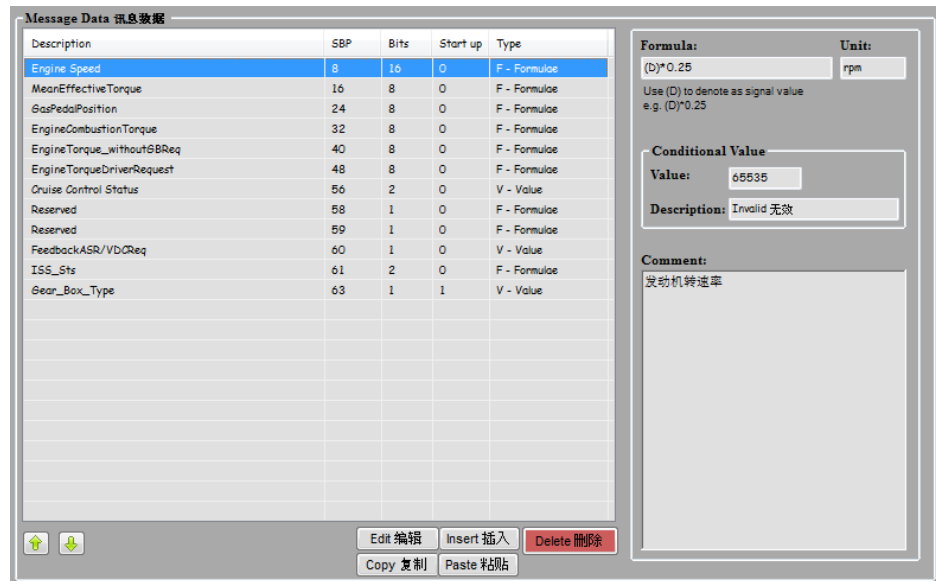


Figure 16: Editing screen for control type -F

In this example, the length of data signal “Engine Speed” is 16 bits long, the Formulae editing box for the computation formula and the resultant unit i.e. the Data Signal-D is divided by 4, so the formula is (D)*0.25 or (D)/4; and the unit is in RPM.

In addition, there are 2 optional sections available to describe the Data Signal during Simulation and Monitoring:

- ☐ Conditional Value: To high light an important value i.e. 0xFFFF (65535d)= Invalid
- ☐ Comments: General note of the data signal

9.4.3. Type B- Bitwise

Figure 17 shows the editing control for type-B.

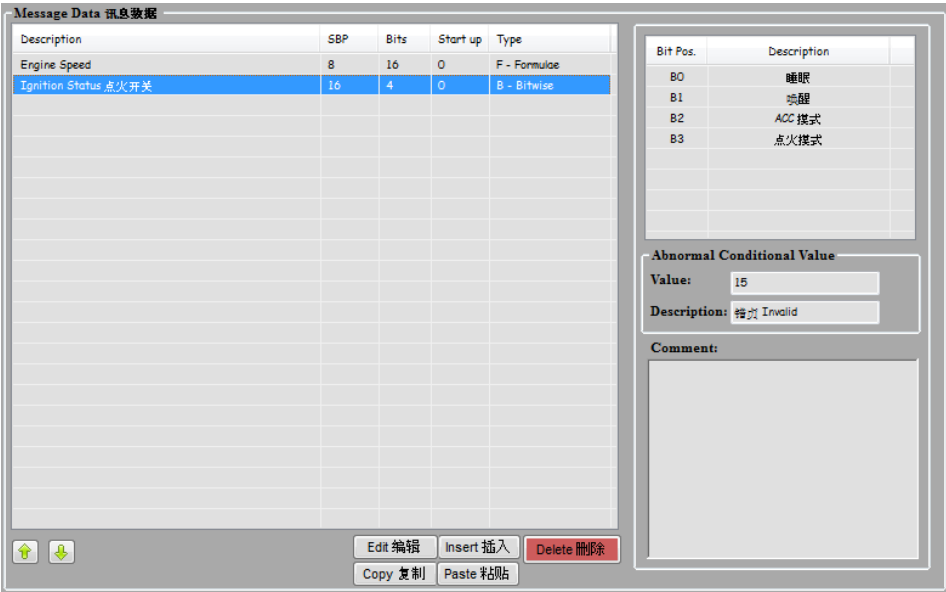


Figure 17: Editing screen for control type -B

In this example, the length of data signal “Ignition Status” is 4 bits long, thus this Signal Data has 4 statuses b0 to b3 as shown.

In addition, there are 2 optional sections available to describe the Data Signal during Simulation and Monitoring:

- ☐ Conditional Value: To high light an important value i.e. 0xF (15d)= Invalid
- ☐ Comments: General note of the data signal

9.4.4. Type AC- Alive Counter

Figure 18 shows the editing control for type-AC.

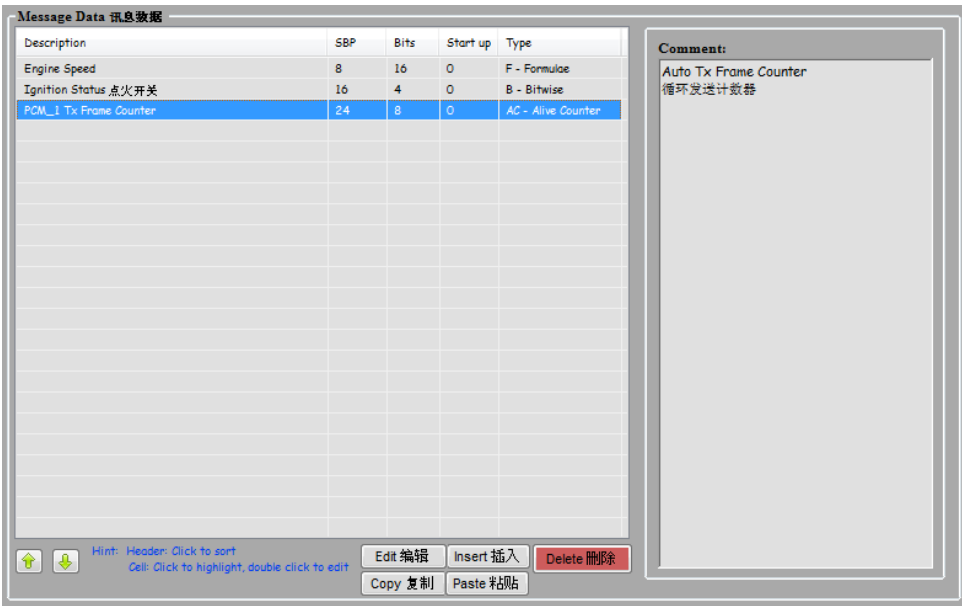


Figure 18: Editing screen for control type -AC

In this example, the length of data signal “PCM_1 Tx Frame Counter” is 8 bits long; counting from 0 to 255. Thus in the **CAN3 Simulator** it will increment this value automatically on every CAN message frame transmission i.e. from 0, 1, 2, ... , 254, 255, 0, 1

There is only COMMENT section available for this data type.

9.4.5. Type RC- Rolling Counter

Figure 19 shows the editing control for type-RC.

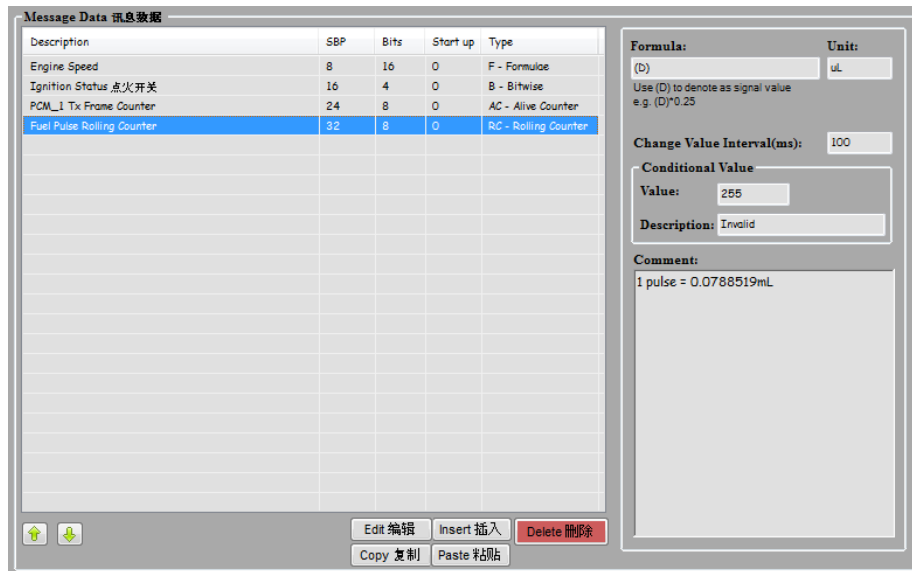


Figure 19: Editing screen for control type -RC

In this example, the length of data signal “Fuel pulse Rolling Counter” is 8 bits long with range value between 0 and 255.

The Change Value Interval (in mili-second) sets the time to update the transmit frame signal. The increment value is set in the **CAN3 Simulator**. Value of 0 means “No action or No interpretation is required”; however if the Condition Value is set to a value other than 0, it will be treated as upper boundary.

Example: If the Condition value is 255 for an 8 bit long data with description of “Invalid”, following two behaviours could happen:

- For incrementing value of 1: at every 100ms, the “Fuel pulse Rolling Counter” will increment by 1 i.e. from 0, 1, 2, ... , 253, 254, 0, 1, .. -> since 255 is an invalid signal
- To transmit a fault signal i.e. Fuel Pulse Signal is in error, set the value to 255 in the **CAN3 Simulator** control.

The Comment section is available for note/reminder usage.

9.4.6. Type CS- Message Frame Checksums

Figure 20 shows the editing control for type -CS

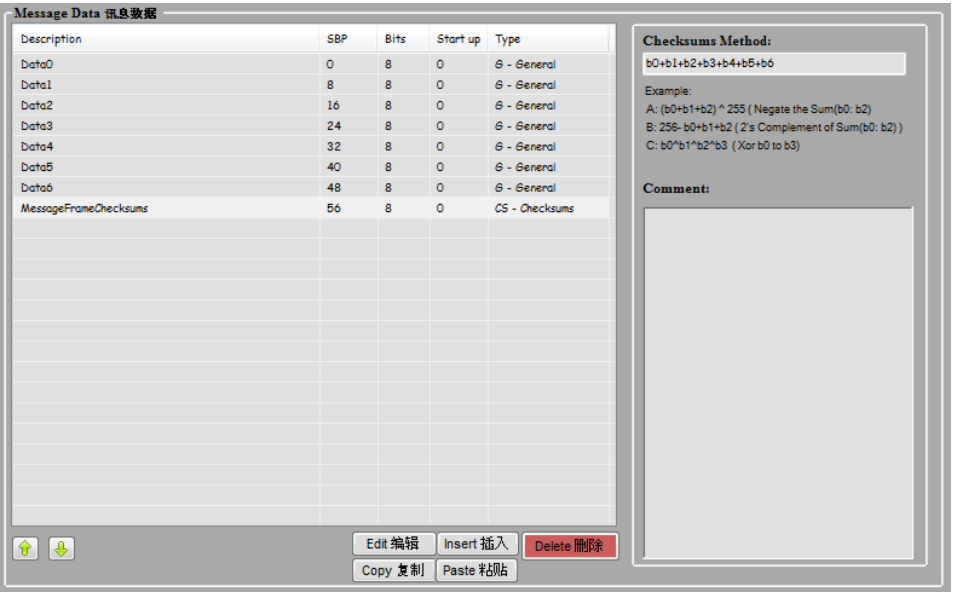


Figure 20: Editing screen for control type -CS

Above example shows the Checksums of the sum of byte 0 to byte 6 i.e. $b0+b1+b2+b3+b4+b5+b6$.

You can include logical operator such as Exclusive-Or (“^”) to derive the commonly used 1’s compliment or 2’s compliment operation.

9.4.7. Type G- General

Figure 22Figure 21 shows the editing control for type-G.

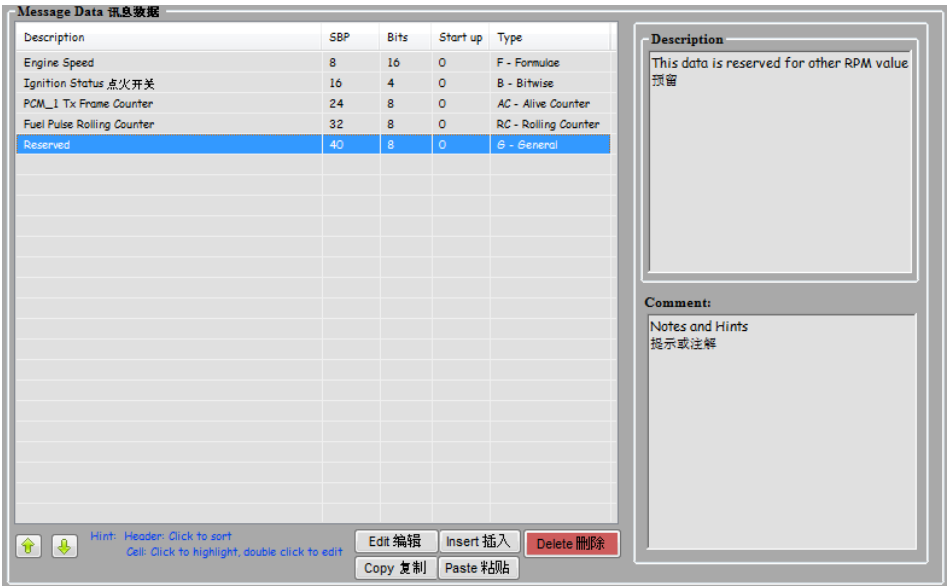


Figure 21: Editing screen for control type -G

In this example, the length of data signal is a reserved data with 8 bits long; value range between 0 and 255.

This data type is for data that is not currently in used, but as a reserved data for future usage.

“Description” and “Comment” sections are available for notes and reminder.

9.4.8. Type S- Spare

Figure 22 shows the editing control for type-S.

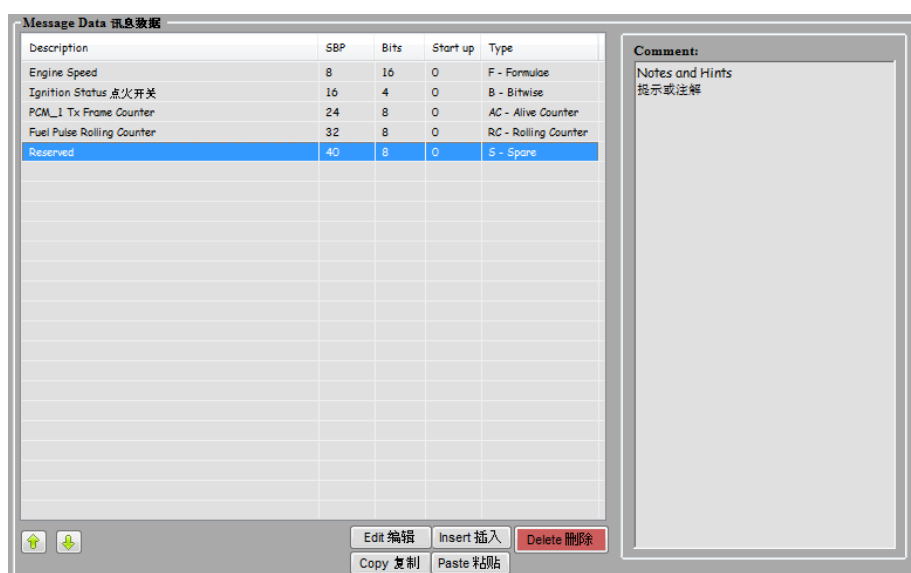


Figure 22: Editing screen for control type -S

In this example, the data signal is “Not Used” or as a Spare; it is 8 bits long with range value between 0 and 255.

Only “Comment” section is available for Notes and Hints.

10. Parallel Unicode dbc file

With recent inclusion of Chinese Simplified Character (GB2312) into dbc formatted file, we have adopted a method to import the Chinese Text into CAN3 Ini file by duplicating the dbc file into parallel Unicode dbc file with extension “.unidbc”.

Before the Chinese Characters formatted dbc file can be read by the CAN3 Writer’s dbc conversion function, you need to convert a copy of the dbc file into Unicode formatted file by the following steps:

- Use the Microsoft WORD to open the dbc file into Chinese Simplified -GB2312 encoding format
- Copy and paste all the WORD contents into a Notepad
- In the Notepad, save the file in Unicode format in the same directory (folder) with the same file name, but the extension in “unidbc” ; so to allow the CAN3 Writer to recognise it.

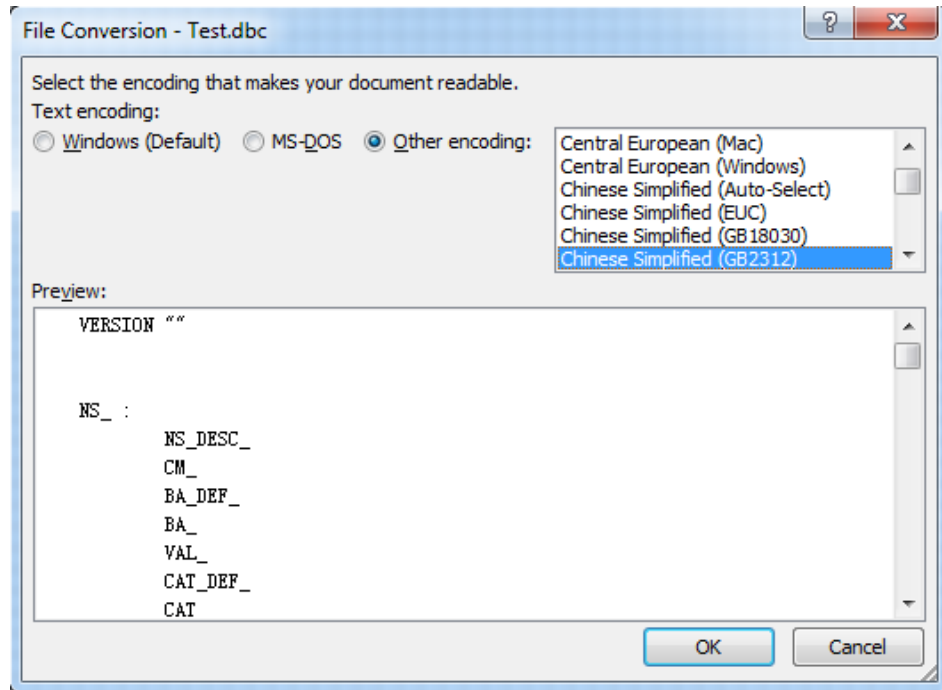
10.1. Importing the dbc with Chinese text into WORD document

Step 1: Start the WORD Window

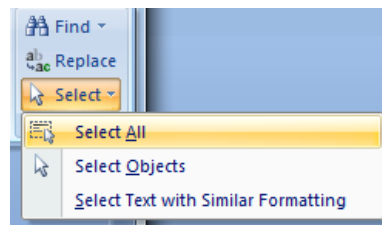
Step 2: Click “Open” to select a dbc file for import



Step 3: After selecting a dbc file with encoded Chinese Characters, it will prompt with a windows (see example picture below) asking for selection as what encoding format should be in use. Select "Other encoding" and then select "Chinese Simplified (GB2312)", then click "OK".



Step 4: In the WORD window, click "Select All" to select all the contents.



Step 5: When all the contents in the WORD are highlighted; Copy all the selection in to the Clipboard using the "Copy" control or "Ctrl-C" buttons on the keypad.

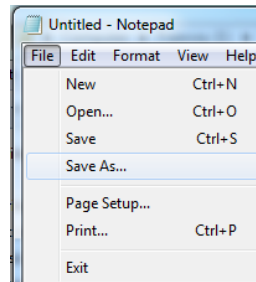
10.2. Import (Paste) the Chinese Simplified –GB2312 contents into Notepad

Open a blank Notepad Window and paste the just copied contents onto this new Notepad, the Notepad will automatically display the Chinese Characters correctly- See sample below:

```
VAL_ 578 MCU_Err_IGBT 1 "Failure" 0 "No_Failure" ;
VAL_ 578 MCU_CurrentOverload 3 "reserved" 2 "二级故障;" 1 "一级故障;" 0 "无故障;" ;
VAL_ 578 MCU_ControllerTemp_Status 3 "reserved" 2 "二级故障;" 1 "一级故障;" 0 "无故障;" ;
VAL_ 578 MCU_ControllerOverVolt_Status 3 "reserved" 2 "二级故障;" 1 "一级故障;" 0 "无故障;" ;
VAL_ 578 MCU_MotorTemp_Fault 3 "reserved" 2 "二级故障;" 1 "一级故障;" 0 "无故障;" ;
VAL_ 578 MCU_ActiveDischrgSt 3 "泄放故障" 2 "泄放完成" 1 "主动泄放中" 0 "未泄放" ;
VAL_ 578 MCU_MotorUnloadSt 3 "卸载完成" 2 "卸载中" 1 "未卸载" 0 "无效" ;
VAL_ 578 MCU_ControllerFault 3 "reserved" 2 "二级故障;" 1 "一级故障;" 0 "无故障;" ;
```

10.3. Generate/Save the Unidbc file in Unicode encoding

In Notepad Window, click File→ Save As to open the Save Window dialog.

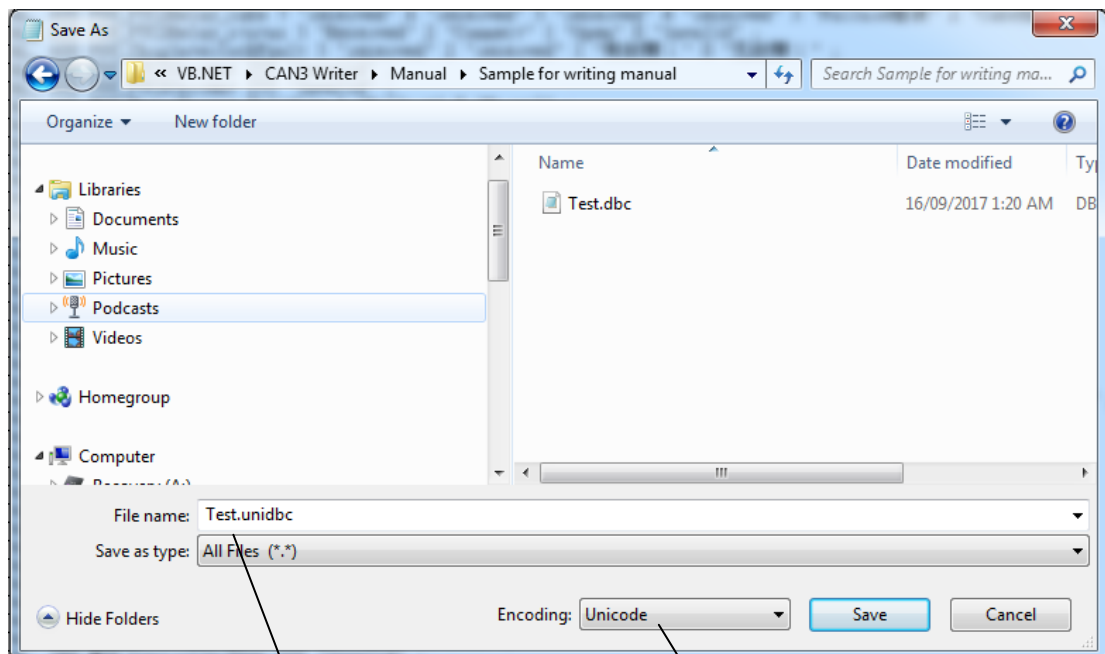


With reference to the Save File windows- See Figure 23, follows the steps below to save the Unicode formatted file:

Step 1: Select Encoding format to Unicode

Step 2: Select “Save as type” to ALL Files (*.*), so that you can see the original file

Step3: In the “File Name” section, type in the original dbc file name, but with extension **Unicode** i.e. “Original_Filename.unidbc” in the same directory i.e. side-by-side of the original dbc file.



Save the file in the original file name, but in extension “unidbc”

Select “Unicode” encoding format

Figure 23: Window dialog for Unicode formatted file

**** Now the two dbc files are side-by-side in the same folder and is ready for conversion into CAN3
Ini file

Revision History

Version	Date	Description	Edited by
1.00	10.06.17	First Draft	DS
1.01	10.10.19	Modify INI File to suit CAN Message Monitor and CAN Message Simulation	DS
1.02	11.03.11	Add Spare Type Add Rolling Counter Type Add Alive Counter Type	DS
1.03	11.05.05	Clean up	DS
1.04	15.04.29	Fixed the index pointer after deleting data When saving the INI file, rename old INI file as backup file Fixed the Rolling Counter that when the condition data is 0, no condition is applied	PC
1.05	17.10.12	Added parallel importation of the Unicode dbc file for Chinese Characters Added Chinese characters for the control buttons	PC
1.06	19.04.05	Added the option description for Checksums type –CS	PC