The definition of the serial (USB) conversion CAN protocol

The serial port baud rate default is 2000000, 8 bit data bit, 1 bit stop bit, no check. Two communication protocols are supported within the converter, one is a fixed 20 byte communication protocol, and the other is a variable length communication protocol.

1. Variable length communication format

USBCAN V7.20		
COM Configure	Reply to reply	
COM: COM3 - Refresh	No Receive ID Reply Frame Type Reply Frame Format Reply ID Reply Data	Clear Delete
COM bps: 2000000 - Change bps		Add
Normal Close Open		🔲 Auto Answer
CAN Configure	More frames to send	
Mode: Normal mode 👻	Send No Frame Type Frame Format Frame ID Data Interval(ms)	Clear Delete
Type: Extended Frame 👻		Add
CAN bps: 250kbps 👻		Cont and other from
Manual set bps 📃 Only send once		Send Serected Itame
Fixed 20 bytes to send and receive		Sequence Cycle
Filter ID:0x 00000000 Set and Stant		Stop
Mask ID: 0x 00000000	Format Data frame V TD Ox 00000000 Data Ox 11 22 33 44 55 66 77 88	Send a single frame
Bus State	5000 Digital an anning and Digital Clear Pause Continue Save	Auto seve Exit
Receive error: 0	Jispiay receive only Auto heiresh	
Transmit error: 0 No Dire	ction Time-scale Frame Type Frame Format Frame ID Data Length Data(LDouble-cl	ick Hex->Dec)
Error Normal		
Bus State: Bus-on		
Monitor		
Receive ID Configure		
Delete Add		
No Receive ID		
Disable Enable		

When the software is not selected

Fixed 20 bytes to send and receive

(for example, the red line is checked in the box above), click setup and start button. The internal communication of the converter is a variable length communication protocol. The specific protocol is as follows:

	Definition			
Packet header	Охаа			
Туер	0хс0 Туер	0хс0 Туер		
	bit5(frame	type 0- standard frame	(frame ID 2	bytes), 1-extended
	frame (fram	ne ID 4 bytes))		
	bit4(frame	bit4(frame format 0- data frame, 1 remote frame)		
	Bit0~3 Frame data length (0~8)			
		1~8 bit (BYTE1)		1~8 bit (BYTE1)
Frame ID	Extended	9~16 bit (BYTE2)	Standard	
	frame	17~24 bit (BYTE3)	Frame	9~11 bit (BYTE2)
		25~29 bit (BYTE4)		
	CAN sends or accepts data 1			
Frame data	CAN sends or accepts data 2			
(frame data	CAN sends or accepts data 3			
may be 0~8	CAN sends or accepts data 4			
data)	CAN sends or accepts data 5			
	CAN sends	or accepts data 6		

	CAN sends or accepts data 7
	CAN sends or accepts data 8
End code	0x55

Sample description (standard frame):

1. For example, send the CAN standard frame ID to 0x123, and the corresponding

data as 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88

Then the serial port sends data format to AA C8 23 01 11 22 33 44 55 66 77 88 55

The following is an explanation

AA Packet header

C8 Bit5 is 0 standard frame, bit4 is 0, data frame bit0~3 is 8, data length is 8,

sending 8 bytes

23 01 For the frame ID 0x123

11 22 33 44 55 66 77 88 Send frame data for CAN

55 End code

For example, the serial port is received:

AA C8 23 01 11 22 33 44 55 66 77 88 55

The received CAN is the standard frame, ID is 0x123, the corresponding data is 11

22 33 44 55 66 77 88, 8 bytes, and the resolution is the same as above

2. For example, send CAN standard frame ID to 0x103, and the corresponding data

is 0x 11 0x22

Then the serial port sends data format to:

AA C2 03 01 11 22 55

The following is an explanation:

AA Packet header

C2 bit5 is 0 standard frame, bit4 is 0, data frame bit0~3 is 2, data length is 2,

sending 2 bytes

03 01 For the frame ID 0x103

11 22 Send frame data for CAN

55 End code

For example, the virtual serial port is received:

AA C2 03 01 11 22 55

The received CAN is the standard frame, ID is 0x0103, the corresponding data is 11

22, 2 bytes, and the resolution is the same as above

Sample description (extended frame):

1. For example, the CAN extended frame ID is 0x1234567, and the corresponding data is 0X11,

0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88

Then the serial port sends data format to:

AA E8 67 45 23 01 11 22 33 44 55 66 77 88 55

The following is an explanation:

AA Packet header

E8 Bit5 is 1, extended frame bit4 is 0, data frame bit0~3 is 8, data length is 8,

sending 8

bytes

67 45 23 01 For the frame ID 0x1234567

11 22 33 44 55 66 77 88 Send CAN frame data

55 End code

For example, the serial port is received:

AA E8 67 45 23 01 11 22 33 44 55 66 77 88 55

The accepted CAN is an extended frame, ID is 0x1234567, and the corresponding

data is 11 22 3344 55 66 77 88, 8 bytes, parsed as above

2. For example, the CAN extended frame ID is 0x1033021, and the corresponding

data is 0x 11 0x22:

Then the serial port sends data format to:

AA E2 21 30 03 01 11 22 55

The following is an explanation:

AA Packet header

E2 Bit5 is 0, extended frame bit4 is 0, data frame bit0~3 is 2, data length is 2,

sending 2 bytes

21 30 03 01 For the frame ID 0x1033021

11 22 Send CAN frame data

55 End code

For example, the serial port is received

AA E2 21 30 03 01 11 22 55

The accepted CAN is an extended frame, ID is 0x1033021, and the corresponding

data is 11 22, 2 bytes, parsed as above

2. Fixed 20 byte transceiver protocol

USBCAN V7.20		
COM Configure	Reply to reply	
COM: COM3 - Refresh	No Receive ID Reply Frame Type Reply Frame Format Reply ID Reply Data	Clear Delete
COM bps: 2000000 - Change bps		Add
Normal Close Open		
ain a ci		Auto Answer
Mode: Normal mode	More frames to send	
Trees: Extended Frame	Send No Frame Type Frame Format Frame ID Data Interval(ms)	Clear Delete
250kbps		Add
		Send selected frame
Fixed 20 bytes to send and receive		🗖 Sequence 📄 Cycle
TILL TR 0 0000000		Stop
Mask ID: 0x 00000000 Set and Start	00000000 11 22 22 44 55 66 77 22	Sent a single from
	Format: Jata frame V ID: Ox 0000000 Data:Ox 11 22 33 44 33 00 11 00	Seur a studte trame
Bus State Total: Receive error: 0	5000 Display receive only Auto Refresh Clear Pause Continue Save	Auto save Exit
Transmit error: 0 No Dire	ction Time scale Frame Type Frame Format Frame ID Data Length Data(LDouble=cli	ck Hex->Dec)
Error Normal		
Bus State: Bus-on		
Monitor		
Receive ID Configure		
Delete Add		
No Receive ID		
Disable Enable		

When the software is selected Fixed 20 bytes to send and receive, (for example, the red line is checked in the box above), click setup and start button. The definition of the converter's internal communication is fixed 20 byte transceiver definition. The specific protocol is as follows.

		Definition
0	Packet header	Охаа
1	Packet header	0x55
2	ТҮРЕ	0x01-Data
3	Frame type	0x01-Standard frame, 0x02-Extended frame
4	Frame format	0x01- Data frame, 0x02-Remote frame
5	Frame ID data 1	1~8 bit, high bytes at the front, low bytes at the back
6	Frame ID data 2	9~16 bit, high bytes at the front, low bytes at the back
7	Frame ID data 3	17~24 bit, high bytes at the front, low bytes at the back
8	Frame ID data 4	25~32 bit, high bytes at the front, low bytes at the back
9	Frame data length	The data length of the CAN bus that is sent or accepted

10	Frame data 1	CAN sends or accepts data 1
11	Frame data 2	CAN sends or accepts data 2
12	Frame data 3	CAN sends or accepts data 3
13	Frame data 4	CAN sends or accepts data 4
14	Frame data 5	CAN sends or accepts data 5
15	Frame data 6	CAN sends or accepts data
16	Frame data 7	CAN sends or accepts data 7
17	Frame data 8	CAN sends or accepts data 8
18	Reserve	0x00
19	Check code	From frame type to error code, accumulating and low 8 bits
		(red part)

For example, we accept or send the data standard frame ID to 0x123, and the data

is 11 22 33 44 55 66 77 88

It is configured as follows

		Data
0	Packet header	Охаа
1	Packet header	0x55
2	ТҮРЕ	0x01
3	Frame type	0x01
4	Frame format	0x01
5	Frame ID data 1	0x23
6	Frame ID data 2	0x01
7	Frame ID data 3	0x00
8	Frame ID data 4	0x00
9	Frame data	0x08
	length	
10	Frame data 1	11
11	Frame data 2	22
12	Frame data 3	33
13	Frame data 4	44
14	Frame data 5	55
15	Frame data 6	66
16	Frame data 7	77
17	Frame data 8	88

18	Reserve	0x00
19	Check code	<mark>0x93</mark> (0x01+0x01+0x01+0x23+0x01+0x00+0x00+0X08+
		0x11+0x22+0x33+0x44+0x55+0x66+0x77+0x88+0x00=0x293
		Low 8 bits are 0x93)

For example, we accept or send the data extension frame ID to 0x12345678, and

the data is	01 02 03 04 05 06 07 0	08, It is configured as follows

		Data
0	Packet header	Охаа
1	Packet header	0x55
2	ТҮРЕ	0x01
3	Frame type	0x02
4	Frame format	0x01
5	Frame ID data 1	0x78
6	Frame ID data 2	0x56
7	Frame ID data 3	0x34
8	Frame ID data 4	0x12
9	Frame data	0x08
	length	
10	Frame data 1	01
11	Frame data 2	02
12	Frame data 3	03
13	Frame data 4	04
14	Frame data 5	05
15	Frame data 6	06
16	Frame data 7	07
17	Frame data 8	08
18	Reserve	0x00
19	Check code	<mark>0x44</mark> (0x01+0x02+0x01+0x78+0x56+0x34+0x12+0x08
		+0x01+0x02+0x03+0x04+0x05+0x06+0x07+0x08+0x00=0x144
		Low 8 bits are 0x44)