Secondary Development Serial Conversion Definition of CAN Protocol

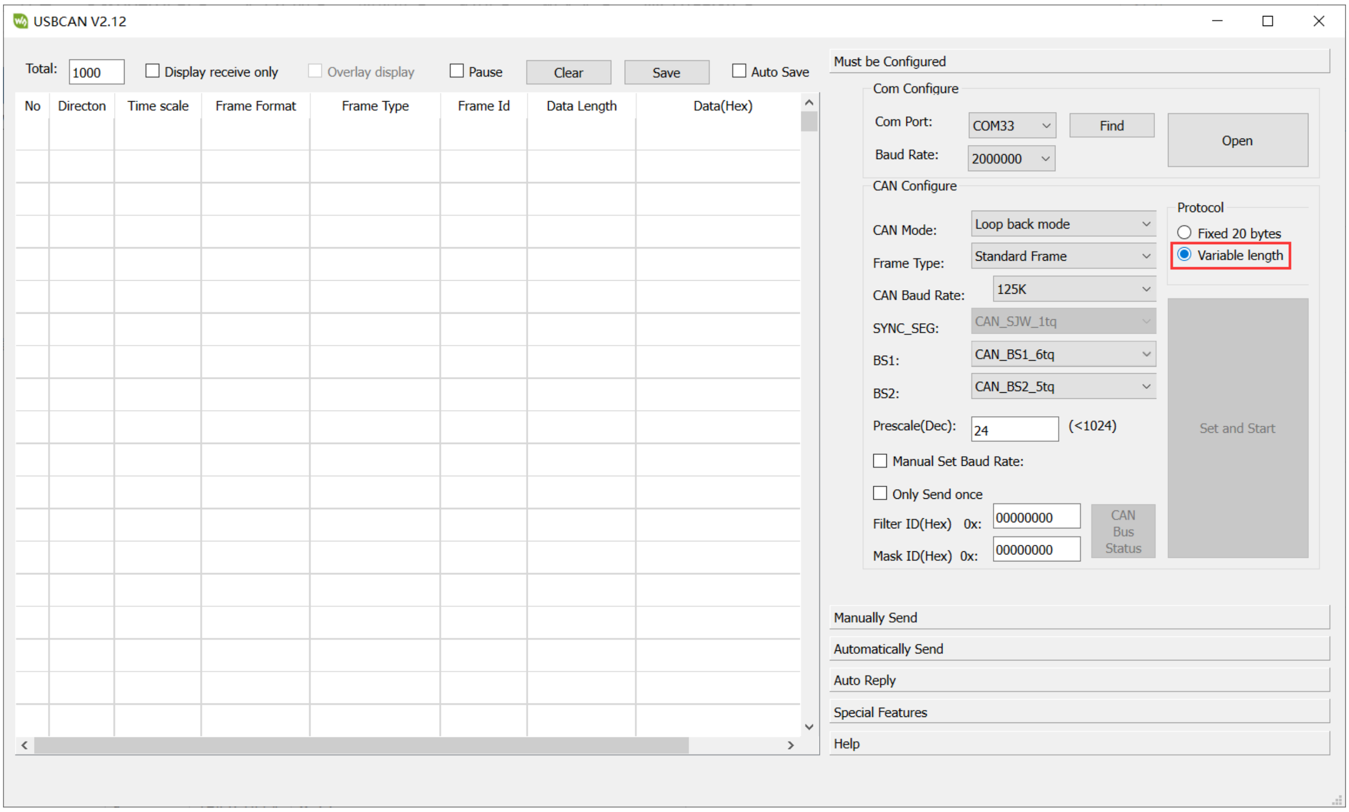
From Waveshare Wiki
Jump to: navigation, search

Serial (USB) to CAN Protocol Conversion

The serial port baud rate is default to 2 Mbps, 8 data bits, 1 stop bit, no parity check. The converter supports two communication protocols: one is a fixed 20-byte communication protocol, and the other is a variable-length communication protocol.

Variable Length Communication Format

Please select Variable length in the Protocol box, click the "Set and Start" button



USBCAN V2.12

Total: 1000 □ Display receive only □ Overlay display □ Pause □ Auto Save

Must be Configured

Com Configure
Com Port: COM33
Baud Rate: 2000000

CAN Configure
CAN Mode: Loop back mode
Frame Type: Standard Frame
CAN Baud Rate: 125K
SYNC_SEG: CAN_SJW_1tq
BS1: CAN_BS1_6tq
BS2: CAN_BS2_5tq
Prescale(Dec): 24 (<1024)
□ Manual Set Baud Rate:
□ Only Send once
Filter ID(Hex) 0x: 00000000
Mask ID(Hex) 0x: 00000000

Protocol
○ Fixed 20 bytes
● Variable length

(/wiki/File:CAN-Variable-length.png)

The internal communication of the converter uses a variable-length communication protocol, as specified below:

Variable-length communication protocol format

Field	Definition	Description
Packet header	0xAA	Packet start flag
Type	0xC0 type	▪ bit5: Frame type 0 - standard frame (frame ID 2 bytes), 1 - extended frame (frame ID 4 bytes) ▪ bit4: Frame format 0 - data frame, 1 - remote frame ▪ Bit0~3: Frame data length 0~8
Frame ID	Extended frame	1~8 bits (byte 1) 9~16 bits (byte 2) 17~24 bits (byte 3) 25~29 bits (byte 4)
	Standard framework	1~8 bits (byte 1) 9~11 bits (byte 2)
Frame data	Variable length	Can send or receive data 1~8 (0~8 pieces of data)
End code	0x55	Data packet end flag

Example Explanation (Standard Framework)

For example, if you send the CAN standard frame ID to 0x123, the data is 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88, and the data format sent by the serial port is:

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

```
  * AA: Message header
  * C8: bit5 is 0 (standard frame), bit4 is 0 (data frame), bit0~3 is 8 (data len
gth is 8)
  * 23 01: Frame ID 0x123
  * 11 22 33 44 55 66 77 88: CAN sends frame data
  * 55: End code
```

For example, if the CAN standard frame ID is sent to the 0x103, the data is 0x11 0x22, and the data format sent by the serial port is:

```
AA C2 03 01 11 22 55
```

```
  * AA: Message header
  * C2: bit5 is 0 (standard frame), bit4 is 0 (data frame), bit0~3 is 2 (data length is 2)
  * 03 01: Frame ID 0x103
  * 11 22: CAN sends frame data
  * 55: End code
```

Example Description (Extended Framework)

For example, if you send the CAN extension frame ID to 0x1234567, the data is 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88, and the data format sent by the serial port is:

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

```
  * AA: Message header
  * E8: bit5 is 1 (extended frame), bit4 is 0 (data frame), bit0~3 is 8 (data length is 8)
  * 67 45 23 01: Frame ID 0x1234567
  * 11 22 33 44 55 66 77 88: CAN sends frame data
  * 55: End code
```

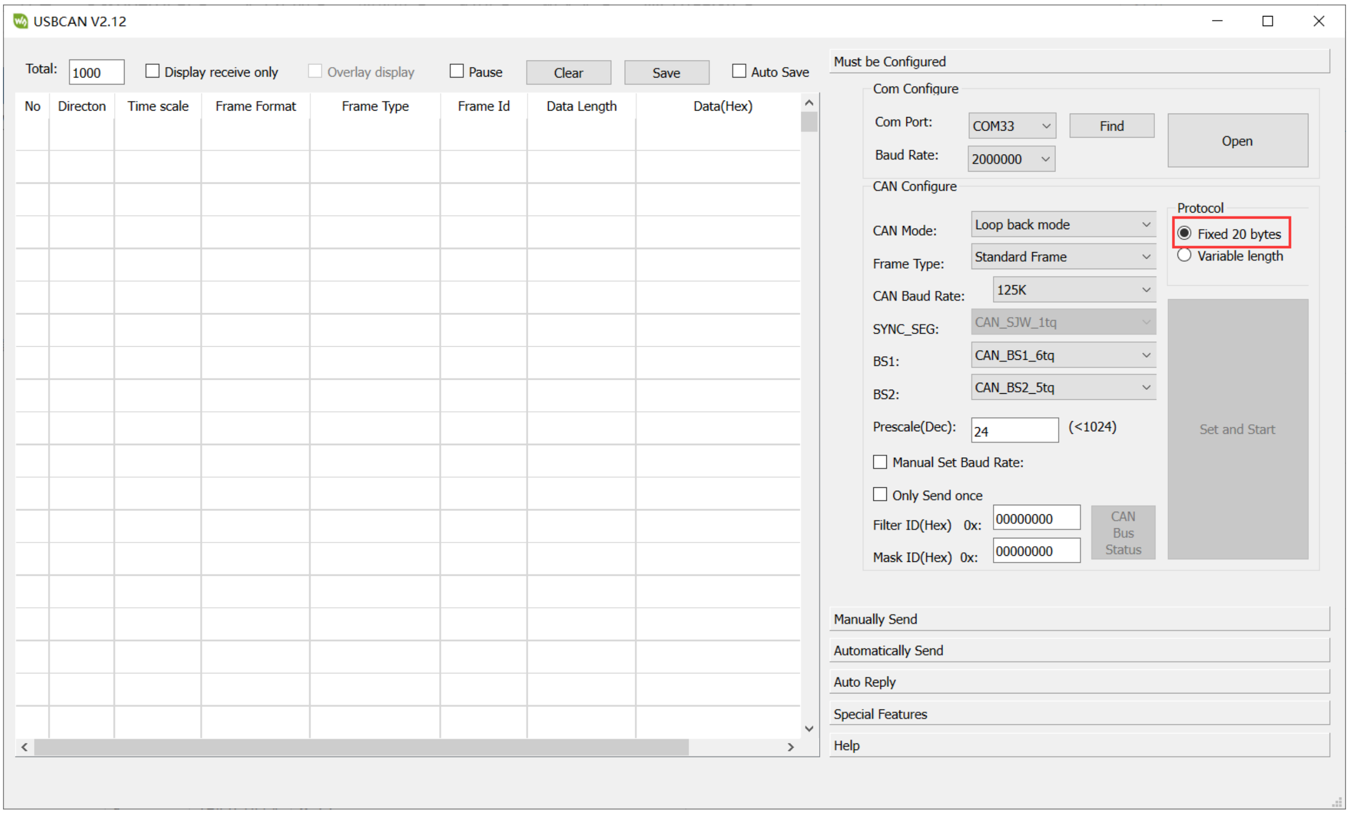
For example, if the CAN extension frame ID is sent to the 0x1033021, the data is 0x11 0x22, and the data format sent by the serial port is:

```
AA E2 21 30 03 01 11 22 55
```

```
  * AA: Message header
  * E2: bit5 is 1 (extended frame), bit4 is 0 (data frame), bit0~3 is 2 (data length is 2)
  * 21 30 03 01: Frame ID 0x1033021
  * 11 22: CAN sends frame data
  * 55: End code
```

Fixed 20-Byte Transceiver Protocol

Please select Fixed 20 bytes in the Protocol box, click the "Set and Start" button



Total: 1000 □ Display receive only □ Overlay display □ Pause □ Auto Save

Com Port: COM33

Baud Rate: 2000000

CAN Mode: Loop back mode

Frame Type: Standard Frame

CAN Baud Rate: 125K

SYNC_SEG: CAN_SJW_1tq

BS1: CAN_BS1_6tq

BS2: CAN_BS2_5tq

Prescale(Dec): 24 (<1024)

□ Manual Set Baud Rate:

□ Only Send once

Filter ID(Hex) 0x: 00000000

Mask ID(Hex) 0x: 00000000

Protocol: ● Fixed 20 bytes ○ Variable length

(/wiki/File:CAN-Fixed-20-bytes.png)

The internal communication of the converter uses a fixed 20-byte communication protocol, as specified below:

Fixed 20-byte protocol format

Byte Position	Field	Description
0	Packet header	0xAA
1	Packet header	0x55
2	Type	0x01
3	Framework type	0x01
4	Framework 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 length	0x08
10	Frame data 1	0x11
11	Frame data 2	0x22
12	Frame data 3	0x33
13	Frame data 4	0x44
14	Frame data 5	0x55
15	Frame data 6	0x66
16	Frame data 7	0x77
17	Frame data 8	0x88
18	Reserve	0x00
19	Check code	The low 8 bits of the cumulative sum from frame type to error code (e.g., 0x93)

Example

For example, if you send a standard frame ID 0x123 with data of 11 22 33 44 55 66 77 88, the check code is calculated as follows:

`0x01 + 0x01 + 0x01 + 0x23 + 0x01 + 0x00 + 0x00 + 0x08 + 0x11 + 0x22 + 0x33 + 0x44 + 0x55 + 0x66 + 0x77 + 0x88 + 0x00 = 0x293`

The lower 8 bits are `0x93`.

CAN Configuration Commands

CAN types of commands have two options: ox02 and ox12. One type sends and receives data in a fixed 20 bytes, while the other type sends and receives data in variable lengths. These two methods cannot communicate with each other. Please distinguish between them and choose one according to your needs

No.	Definition	Description
0	Message header	0xaa
1	Message header	0x55
2	Type	0x02-Setting (for sending and receiving data with a fixed 20-byte protocol); 0x12-Setting (for sending and receiving data with a variable protocol) These two methods cannot communicate with each other. Please distinguish and choose one based on your needs
3	CAN baud rate	0x01(1Mbps), 0x02 (800kbps), 0x03 (500kbps), 0x04 (400kbps), 0x05 (250kbps), 0x06 (200kbps), 0x07 (125kbps), 0x08 (100kbps), 0x09 (50kbps), 0x0a (20kbps), 0x0b (10kbps), 0x0c (5kbps)
4	Frame type	0x01-Standard frame, 0x02 Extended frame
5	Filter ID1	1~8 bits, high byte first, low byte last
6	Filter ID2	9~16 bits, high byte first, low byte last
7	Filter ID3	17~24 bits, high byte first, low byte last
8	Filter ID4	25~32 bits, high byte first, low byte last
9	Block ID1	1~8 bits, high byte first, low byte last
10	Block ID2	9~16 bits, high byte first, low byte last
11	Block ID3	17~24 bits, high byte first, low byte last
12	Block ID4	25~32 bits, high byte first, low byte last
13	CAN mode	0x00-Normal mode, 0x01-Silent mode, 0x02-Loopback mode, 0x03-Loopback silent mode
14	Auto-retransmit	0x00—Auto-retransmit 0x01—Disable auto-retransmit
15	Backup	0x00
16	Backup	0x00
17	Backup	0x00
18	Backup	0x00
19	Checksum	The low 8 bits (red part) of the cumulative sum from the frame type to the error code

Retrieved from "https://www.waveshare.com/w/index.php?title=Secondary_Development_Serial_Conversion_Definition_of_CAN_Protocol&oldid=104582 (https://www.waveshare.com/w/index.php?title=Secondary_Development_Serial_Conversion_Definition_of_CAN_Protocol&oldid=104582)"