

# J2534


## Table of contents


How to intall J2534-1 dll.....	1
Overview.....	2
Function Description .....	2
3.1. PassThruOpen.....	2
3.2. PassThruClose .....	2
3.3. PassThruConnect .....	2
3.4. PassThruDisconnect.....	3
3.5. PassThruReadMsgs.....	3
3.6. PassThruWriteMsgs.....	3
3.7. PassThruStartPeriodicMsg .....	4
3.8. PassThruStopPeriodicMsg .....	4
3.9. PassThruStartMsgFilter.....	4
3.10. PassThruStopMsgFilter .....	4
3.11. PassThruSetProgrammingVoltage.....	4
3.12. PassThruReadVersion.....	5
3.13. PassThruGetLastError .....	5
3.14. PassThruIoctl .....	5

## How to intall J2534-1 dll.

Run startup.bat as an administrator. This will register the WIN system registry according to the requirements of J2534-1. The DLL will be copied to the system folder. If you do not need to check the registry, you can skip this step. Just place all the DLL files in the folder to a location where they can be called (this location might vary depending on the programming language).

Download version TSMaster v2.24.2.6.1062 or a later release version. The J2534 API will be installed automatically in the system folder.

 E:\SUN\TSMaster\bin

 TSJ2534.dll

## Overview

The provided DLLs are all 32-bit DLLs, with the J2534 DLL being the one called by the user. The DLLs need to be invoked in the form of WINAPI. The supported protocols are as follows: ISO J2534-1 2004, ISO J2534-2 2019, GMW 17753. The supported J2534 version is 04.04.

## Function Description

### 3.1. PassThruOpen

The J2534 version 04.04 only supports using one CAN channel at a time. By default, the Open function will connect to the first available CAN channel on the device. When the Open function is called, the software will establish a connection with the hardware, and it might directly initiate communication. This is because the TOSUN API needs to establish a connection to obtain the hardware's handle.

If you want to specify the channel, you can make the call like this. The following call will connect to channel 2 (but still cannot specify the hardware):

```
unsigned long deviceID;  
PassThruOpen("J2534-2:TOSUN CH2", &deviceID);
```

### 3.2. PassThruClose

Disconnect from the hardware. Release the resources in the DLL.

### 3.3. PassThruConnect

Choose the protocol to use. If a non-CAN FD protocol is specified, it will directly modify the hardware's CAN baud rate settings. If a CAN FD protocol is specified, the CAN baud rate will be cached. It will only be applied when calling PassThruIoctl to set the data field baud rate.

The input variable Flags will not be checked, as the hardware will definitely support both 11-bit CAN ID standard frames and 29-bit CAN ID extended frames simultaneously.

The supported protocols are as follows:

```
CAN, CAN_PS, FD_CAN_PS, CAN_FD_PS, ISO15765, ISO15765_PS,  
FD_ISO15765_PS, ISO15765_FD_PS.
```

It should be noted that, due to lack of pin selection support, there is no practical difference

between protocols with "PS" and without "PS".

### 3.4. PassThruDisconnect

Delete all settings made after the connect function, but without disconnecting or stopping the hardware (for example, the hardware will still respond with ACK). When connect is called again, the J2534 API will clean up the received messages in between, preventing them from being read again.

### 3.5. PassThruReadMsgs

Read J2534 messages. The input Timeout parameter will not be processed.

The J2534 API maintains a receive buffer where messages are stored upon reception. The Read function retrieves data from this buffer.

Supported RxStatus include:

- CAN\_29BIT\_ID: Indicates whether the extended frame is used.
- TX\_MSG\_TYPE: Indicates if it is a message readback.
- FD\_CAN\_BRS and CAN\_FD\_BRS: Indicates the use of data field bitrate switching.
- FD\_CAN\_FORMAT and CAN\_FD\_FORMAT: Indicates if it is a CAN FD frame.
- FD\_CAN\_ESI and CAN\_FD\_ESI: Indicates whether the ESI bit of the message is set to 1.

### 3.6. PassThruWriteMsgs

Send J2534 messages. All supported types can be sent, and it is not necessary for them to match the protocol set in the Connect function. The Timeout parameter will not be processed.

If you choose to send an ISO15765 message, but the CAN ID has not been registered beforehand using PassThruStartMsgFilter, the function will automatically register a request ID that matches the current ID, but with no corresponding TP layer address mapping for the response CAN ID. This is used to trigger functional addressing requests, but if the filter is not additionally configured, you will not receive any response messages.

Supported TxFlag include:

- CAN\_29BIT\_ID: Whether to use extended frames.
- FD\_CAN\_BRS and CAN\_FD\_BRS: Whether to use data field bitrate switching.
- FD\_CAN\_FORMAT and CAN\_FD\_FORMAT: Whether it is a CAN FD frame.

### 3.7. PassThruStartPeriodicMsg:

Periodic transmission of messages. Supports up to 10 periodic messages.

CAN and CAN FD messages, as well as ISO156765 messages that are determined to be single-frame messages, can be sent periodically.

Other messages cannot be sent periodically.

### 3.8. PassThruStopPeriodicMsg

To stop the periodic transmission of messages.

### 3.9. PassThruStartMsgFilter

To set filters:

Note: According to the J2534 protocol, **if you do not set filters, you will not receive any messages. The API will discard all received messages.**

PASS\_FILTER and BLOCK\_FILTER can be set in any case; FLOW\_CONTROL\_FILTER can only be set when selecting ISO15765 related protocols during Connect.

You can set up to 10 PASS\_FILTER and BLOCK\_FILTER filters combined, and FLOW\_CONTROL\_FILTER supports 64 filters.

For FLOW\_CONTROL\_FILTER, the function does not validate the input PASSTHRU\_MSG. As long as the data length is greater than 4, the first 4 bits will be processed according to the CAN ID, and the rest will not be processed.

The GMW 17753 provides batch mapping for 29-bit IDs, but this is not supported by the API.

### 3.10. PassThruStopMsgFilter

Delete the specified filter

### 3.11. PassThruSetProgrammingVoltage

Not supported, the function will always return ERR\_NOT\_SUPPORTED.

## 3.12. PassThruReadVersion

You will get the correct PassThru version number. However, for the other two, you will not get the correct answer.

## 3.13. PassThruGetLastError

Get the detailed description of the last error. If it is an error related to J2534, it will be explained according to J2534. If it was originally an error returned by the TOSUN API, you will get the TOSUN API error description. The protocol specifies a string length of 80 characters, and any error descriptions exceeding this length will be forcibly truncated.

## 3.14. PassThruIoctl

GET\_CONFIG and SET\_CONFIG: Will be explained later.

CLEAR\_TX\_BUFFER: Not supported, but will return success directly.

CLEAR\_RX\_BUFFER: Clears the received buffer.

CLEAR\_PERIODIC\_MSGS: Stops all periodic message transmission.

CLEAR\_MSG\_FILTERS: Deletes all filters.

GET\_DEVICE\_INFO: Retrieves the protocols supported by the API.

GET\_PROTOCOL\_INFO: Retrieves specific information about the supported protocols.

WRITE\_MSG\_EXTENSION and READ\_MSG\_EXTENSION: These are handling methods specified in GMW 17753 for CAN FD ISO15765 messages exceeding PASSTHRU\_MSG bytes. Both are supported. If it is not a CAN FD ISO15765 message as specified in GMW 17753, messages exceeding the length will be discarded.

Supported CONFIG parameters:

DATA\_RATE: Modify the arbitration segment bitrate.

LOOPBACK: Indicates support for loopback. Note that loopback messages are not affected by filters.

BIT\_SAMPLE\_POINT: Not supported, but will return success.

SYNC\_JUMP\_WIDTH: Not supported, but will return success.

ISO15765\_BS: Sets the flow control frame content for responses from the TOSUN hardware.

ISO15765\_STMIN: Sets the minimum separation time (STmin) for responses from the TOSUN hardware.

ISO15765\_BS\_TX: Not supported, but will return success.

ISO15765\_STMIN\_TX: Not supported, but will return success.

ISO15765\_WFT\_MAX: Supported.

CAN\_MIXED\_FORMAT: Supported. The API supports simultaneous reception and transmission of ISO15765 and corresponding protocol normal messages.

J1962\_PINS: This parameter does not have any action and will simply return success. Even if pin pins are not set, other functions will not return an error because of this.

FD\_CAN\_DATA\_PHASE\_RATE and CAN\_FD\_DATA\_PHASE\_RATE: Both J2534-2 and GMW 17753 define this parameter, but with different values. Therefore, only the corresponding protocol can set these parameters. Used to change the data phase bitrate for CAN FD. For CAN FD protocols, the device can only enter CAN FD mode after setting this parameter.

FD\_ISO15765\_TX\_DATA\_LENGTH and CAN\_FD\_TC\_DATA\_LENGTH: Both J2534-2 and GMW 17753 define this parameter, but with different values. Therefore, only the corresponding protocol can set these parameters. Sets the maximum DLC for single-frame ISO 15765 messages in CAN FD mode.

HS\_CAN\_TERMINATION and CAN\_FD\_TERMINATION: Both J2534-2 and GMW 17753 define this parameter, but with different values. Therefore, only the corresponding protocol can set these parameters. Sets whether to use termination resistors on the TOSUN hardware. Protocols defined in J2534-1 use the parameter values defined in J2534-2.

N\_CR\_MAX: Both J2534-2 and GMW 17753 define this parameter with the same name but different values. This parameter is currently not supported, but will return success.

ISO15765\_PAD\_VALUE: Padding bits used by ISO15765, all supported.

CAN\_FD\_TYPE: Supports the use of non-ISO CAN FD.

All other parameters will return as not supported.



We take pride in providing a diverse range of services including high performing hardware, cost-effective software, and effective test solutions. Our dedication to delivering fast results, customised solutions, and driving innovation is what sets us apart from others in the industry.



Scan the code to follow