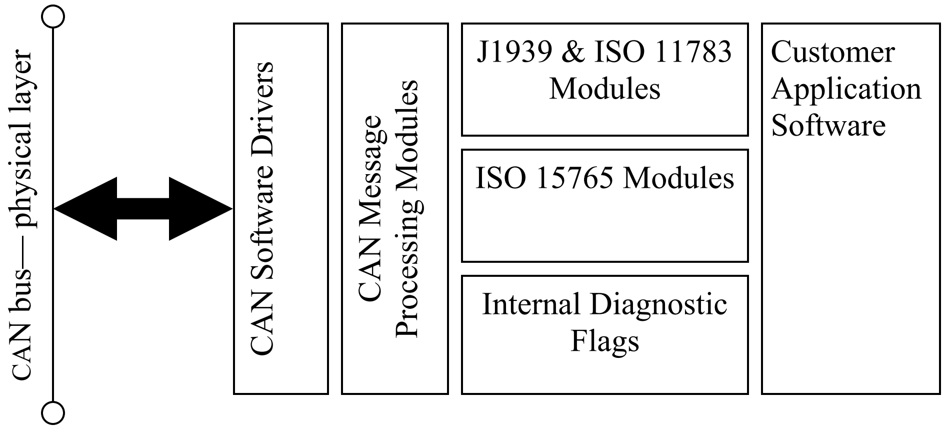
Software Protocol Stack, J1939, ISO 11783 & ISO 15765

Choose the stack that will enable your hardware to speak with other devices over a CAN network utilizing the SAE J1939, ISO 11783 and/or ISO 15765 protocols. This software stack incorporates aspects of:

- The J1939 protocol as defined in the documents: J1939-21, -71, -73, and -81.
- The ISO 11783 protocol, including the extended transport protocol.
- The ISO 15765 protocol, all transport protocol methods.

Modular Design

The stack's modular design separates the major functions of the software into seven major categories. The ISO 11783 functionality is incorporated into the J1939 modules.


CAN bus— physical layer
CAN Software Drivers
CAN Message Processing Modules
J1939 & ISO 11783 Modules
ISO 15765 Modules
Internal Diagnostic Flags
Customer Application Software

1. The CAN Software Drivers allow the transmission/reception of J1939/ISO messages to/from the physical layer of the CAN bus. These drivers need to be adjusted to work with the hardware platform that the stack is being run on.
2. The CAN Message Processing Modules contain the code for buffering messages that have been received and routing those messages to the different channels in the J1939/ISO Modules and Customer Application Software. It also handles the cyclic broadcasting of messages for the different channels within the J1939/ISO Modules and Customer Application Software.
3. The J1939/ISO Message Processing Modules handle the encoding and decoding of messages that use the different aspects of the J1939/ISO protocol. The different aspects these modules handle are as follows:
    - Automatic handling of arbitration for a source address as defined by J1939-81/ISO 11783. This includes transmission of the address claim message, dynamic changing of the source address claimed in the event of a conflict, rebroadcast of the address claim message if needed, and claiming the null address if a source address cannot be

claimed.

- Automatic response to the receipt of the Request message. The response is the message that was requested or a transmission of the Acknowledgment message indicating the reason for not sending the message requested. The rules outlined in J1939-21/ISO 11783 for the reception of the Request message and broadcast of the Acknowledgment message are followed.
- Automatic encoding and decoding of messages that utilize the Transport Protocol, Connection Management message. The Customer Application Software doesn't need to worry about handling the transmission or reception of messages that are transmitted using multiple packets. The software within these modules will build and transmit any messages that are longer the 8 bytes using the TP_CM format. Likewise messages that are received using the TP_CM format are assembled and routed to the channels that require the messages.
- Automatic encoding and decoding of messages utilizing the Extended Transport Protocol (ISO 11783-6).
- Utilization of the ISO 15765 transport protocol. All for methods of transmitting and receiving transport protocol messages are supported (Normal, Normal Fixed, Mixed and Extended).
- Reception and Transmission of active fault codes and previously active fault codes. This allows the user to pass a pointer to a list of fault codes active and/or previously active. The user can activate/deactivate the broadcast of the active fault codes with the call to a function. Previously active fault codes are automatically broadcast upon receipt of the Request message.

4. The Internal Diagnostic Flags are a group of flags that when set indicate that there is a potential problem with the stack. Each flag can be cross referenced with a list of flags in the user manual to determine in which function the flag was set, the problem that was encountered, and a potential solution to the problem. Most flags indicate buffer overflow conditions.
5. The Customer Application Software is the code that is written on top of the stack. This is software that is written by the users of the stack to call the stack's API.

Configurability

The stack was designed for ease of use and configuration. At the heart of the application is a CAN engine that, through the includes.h configuration file, can be scaled up or down to meet the memory confinements of just about any hardware platform.

The includes.h file is used to easily adjust the size of the different buffers within the stack. This allows the user to set the amount of RAM the stack will utilize during operation. The includes.h file also allows the user to choose what aspects of the J1939/IOS protocol they wish to incorporate into the stack at compile time. For example if the hardware device is a listen only device then the Address Claim message would not be required. The compiler switch for including the Address Claim message can be removed from the includes.h file, thus the software concerning that aspect of the J1939/ISO protocol will be removed at compile time. This frees up flash and RAM that would have been consumed by that messages inclusion in the program.

The following is a list of the different aspects of the J1939/ISO protocol that can be remove from the software at compile time just by removing the associated compiler switch:

- TX_DM1—Transmission of the Diagnostic Message 1 (J1939-73/ISO 11783).
- TX_DM2—Transmission of the Diagnostic Message 2 (J1939-73/ISO 11783).
- TX_DM3—Transmission of the Diagnostic Message 3 (J1939-73/ISO 11783).
- RX_DM1—Reception of the Diagnostic Message 1 (J1939-73/ISO 11783).
- RX_DM2—Reception of the Diagnostic Message 2 (J1939-73/ISO 11783).
- TP_CM_BAM_RX—Reception of the Transport Protocol, Connection Management, Broadcast Announce message (J1939-21/ISO 11783).
- TP_CM_BAM_TX—Transmission of the Transport Protocol, Connection Management, Broadcast Announce message (J1939-21/ISO 11783).
- TP_CM_CCONNECT_RX—Reception of the Transport Protocol, Connection Management, Direct Connection message (J1939-21/ISO 11783).
- TP_CM_CONNECT_TX—Transmission of the Transport Protocol, Connection Management, Direct Connection message (J1939-21/ISO 11783).
- ETP_CM_CONNECT_RX—Reception of the Extended Transport Protocol message (ISO 11783-6).
- ETP_CM_CONNECT_TX—Transmission of the Extended Transport Protocol message (ISO 11783-6).
- ADDRESS_CLAIM—Transmission and reception of the Address Claim message (J1939-81/ISO 11783).
- REQUEST—Transmission and reception of the Request message (J1939-21/ISO 11783).
- ACKNOWLEDGMENT—Transmission and reception of the Acknowledgment message (J1939-21/ISO 11783).
- COMMANDED_ADDRESS—Reception of the commanded address message (J1939-81)
- MAX_CAN_PORTS—Defines the number of external CAN ports that are connected to the stack through the physical layer.

Channels

The use of channels is the means by which the J1939 stack routes received messages. The application software will setup a channel or a series of channels with the stack. A list messages that each channel wishes to receive from the CAN bus is passed to the stack. Each list is assigned a channel tag to indicate with channel wants what messages. Each message that is received is assigned a channel tag that corresponds to each channel that needs the message. The channels needing this data call the stack and receive a pointer to the messages that are assigned that channels tag. Thus channels only receive messages that that channel has requested from the stack. Messages that are received that are not needed by any of the channels are discarded.

Drivers

Since we are more familiar with the lower level working of the J1939 stack we offer our services in writing the software drivers for the hardware platform that the stack will be operating on or assisting the engineer who is writing these drivers.

CAN Ports

The CAN stack is capably of simultaneously handling connections to multiple CAN buses. The application code when setting up the messages transmitted and received by the stack will tell the stack what CAN port to accept the received messages from and what port to use when transmitting each message.

Guarantee

We at DakotaSoft Inc. stand by our stack 100%. If there is a problem with the stack that we can’t fix, we will refund the purchase price of the stack.