red lion
Tel +1 (717) 767-6511
Fax +1 (717) 764-0839
www.redlion.net

Raw 29-bit CAN Port Driver

Information Sheet for Crimson v3.0+

Compatible Devices

CAN devices supporting 29-bit identifiers.
A Red Lion device with a CANOpen option card is mandatory to use this driver.

Overview

The Raw 29-bit CAN Port allows users to communicate with compatible CAN devices with the use of user functions described in the Accessing Data section of this information sheet.

Please review the following information to achieve successful communications.

Driver Settings

The Raw 29-bit CAN Port driver is listed under the manufacturer of <System> in the Driver Picker for the CAN Option card.

An On Update field is provided. Actions entered here will be executed each time an initialized (as described in the Accessing Data section) CAN message is received. Typically, a User Program is called to retrieve data.

Message Limits can also be set as needed. This will allow the driver to only allocate the memory required for each application.

Accessing Data

CAN messages must be initialized successfully before being used. This can be accomplished by using one of the following user functions for each message:

```
Int RxCANInit (int port, int ID, int DLC)
Int TxCANInit (int port, int ID, int DLC)
```

Each initialization function listed above will return a value of 1 upon success or a value of 0 indicating failure.

NOTE: Since memory is allocated during the first call to RxCANInit() and TxCANInit(), please be sure to make these calls **after** the system has started. Each 29-bit identifier should only be initialized one time via RxCANInit() and/or TxCANInit().

Received CAN messages initialized via the RxCANInit user function can be retrieved by the following user function:

Int RxCAN (int Port, int [] Data, int ID)

[] Data is an element of a numeric array. The first four bytes of the received messaged will be packed (big endian) in the indicated element of this array, while the remaining 4 bytes will be stored (big endian) in the next consecutive element of the array.

ID is the 29-bit identifier that must be specified in the call to RxCAN().

Upon success this function will return a value of 1.

NOTE: During initialization all data bytes are set to 0. In the event that RxCAN() is called before a message is received, RxCAN will return successful with all data bytes holding a value of 0.

CAN messages initialized via the TxCANInit user function may be sent by using the following user function:

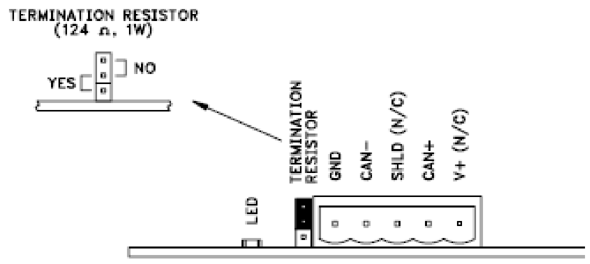
Int TxCAN (int Port, int [] Data, int ID)

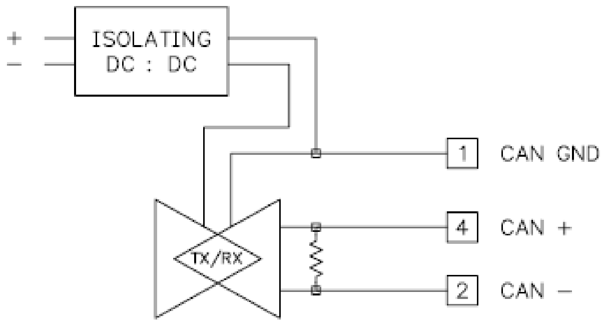
[] Data is an element of a numeric array. The first four bytes of the transmitted message should be set (big endian) in the indicated element of this array, while the remaining 4 bytes should be set (big endian) in the next consecutive element of the array.

ID is the 29-bit identifier that must be specified in the call to TxCAN().

Upon success this function will return a value of 1.

Cable Information





Revision History

11/26/12 – Created
11/28/12 – Added notes and ID specification in Accessing Data section.