Visual Basic DLL API

PCI7202B Device Driver Interface

VITAL Systems Inc.

www.VITALSYSTEM.com

DLL: CN72VB.DLL

CN72VB.DLL allows applications written in Visual Basic and “C” to access the PCI7202B device driver directly.

Operation

The application must call *cpConnectDriver* to connect to the device driver, before calling any other API function.

Function Reference

---

cpConnectDriver

Parameters
None

Operation
This function opens all channels of PCI7202B boards installed in the PC.

Return value
The funcitons returns the number of PCI7202B channels available. 0 if no card is installed.

---

cpGetCANMsg

Parameters

Channel:	Zero based channel number.
Identifier:	11 bit identifier of the received message
Msg:	Message body (max 8 bytes)
MsgSize:	Size of received message

Operation
This function retrieves a message from the drivers receive FIFO queue for the specified channel. Each channel has its own receive FIFO queue. This is a non-blocking call.

Return value
Returns true is there was atleast one message in the receive queue. The contents of the message are returned in the parameters. If the queue is empty, the method returns false.

cpSendCANMsg

Parameters

Channel: Zero based channel number.
Identifier: 11 bit identifier of the message
Msg: Message body (max 8 bytes)
MsgSize: Size of message to transmit

Operation

This function transmits one CAN message. The function first checks if the CAN controller chip is ready. If the buffers are available, the data is transferred to the CAN chip for delivery. The function retries for 5 seconds if the chip buffers are not available. This is a Blocking call.

Return value

The method returns true if the CAN chip accepts the message to transmit. Returns false if no buffer is available.

---

cpResetMsgQue

Parameters

Channel: Zero based channel number.

Operation

This method empties the FIFO receive queue in the driver associated with the channel.

Return value

Returns true if successful.

---

cpBlinkCommsLED

Parameters

Channel: Zero based channel number.
Ticks: On Duration

Operation

Turns on the yellow LED on bracket for the given duration. Each tick is approx. 10mSec duration. For regular transmit and receive operations, the driver controls the LEDs.

Return value

None

---

cpBlinkErrorLED

Parameters

Channel: Zero based channel number.
Ticks: On Duration

Operation

Turns on the Red LED on bracket for the given duration. Each tick is approx. 10mSec duration. For regular transmit and receive operations, the driver controls the LEDs.

Return value

None

---

cpSetBusParam
cpSetBusParamEx

Parameters

Channel: Zero based channel number.
BTr0, BTr1: Bus Timming registers (Refer to Philips SJA1000 CAN controller specs for bit definitions.)

ACr0..ACR3: Acceptance Code Registers
AMr0..AMr3: Acceptance Mask Registers

Ident1, Ident2 Acceptance Identifier
Mask1, Mask2 Acceptance Mask

Operation

These methods set the bus speed as well as bus filters parameters. If you are familiar with the Philips CAN chip and require full control of the CAN chip, you can use the method with ACR and AMR registers. These register values are directly loaded into the CAN chip. For detailed description of these register, please refer to Philips SJA1000 CAN controller specifications.

For simple 11-bit identifier protocol, you can use the second function that calculates the ACR and AMR values automatically. With this method, you can set up two identifiers (Ranges) to accept. The mask register specifies the don't care bits. A 1 in mask bit indicate a don't care bit in the corresponding indentifier bit.

BTR0 and BTR1 set up the bus speed. Following are the values of common bus speeds (CAN chip running at 16Mhz):

BusSpeed	**125 kbps**	**250 kbps**	**500 Kbps**	**1 Mbps**
BTR0	3	1	0	0
BTR1	0x1C	0x1C	0x1C	0x14

Examples:

cpSetBusParam(3, 0x1C, 0x780, 0, 0x780, 0)
The above method sets the speed to 125 kbps. The CAN chip will only accept messages with the identifier value of 0x780.

cpSetBusParam(1, 0x1C, 0x780, 0xF, 0x080, 0x700)
The above method sets the speed to 250 kbps. The CAN chip will accept messages with the identifier value of 0x780 thru 0x78F and 0x080 thru 0x780.

cpSetBusParam(0, 0x1C, 0, 0xFFF, 0, 0xFFF)
The above method sets the speed to 500 kbps. The CAN chip will accept all messages on the network.

Return value
Returns true if successful.

---

cpGetErrorCounters

Parameters

Channel	Zero based channel number.
NetError	General Bus error Counter
TxOverRun	Transmit queue overrun
RxOverRun	Receive Queue overrun

Operation
This method retrieves the error counters associated with the channel. The TxOverRun counter is not used by the PCI7202B device driver.

Return value
Returns true if successful.

---

cpResetErrorCounters

Parameters

Channel:	Zero based channel number.

Operation
Reset error counters associated with the channel.

Return value
Returns true if successful.

---

cpDisableCAN

Parameters

Channel: Zero based channel number.

Operation

This method sets the CAN chip associated with the channel to BUS-OFF state. The CAN chip will not receive or transmit any message. The cpSetBusParam method can be used to bring the CAN chip back to operating mode.

Return value

Returns true if successful.
