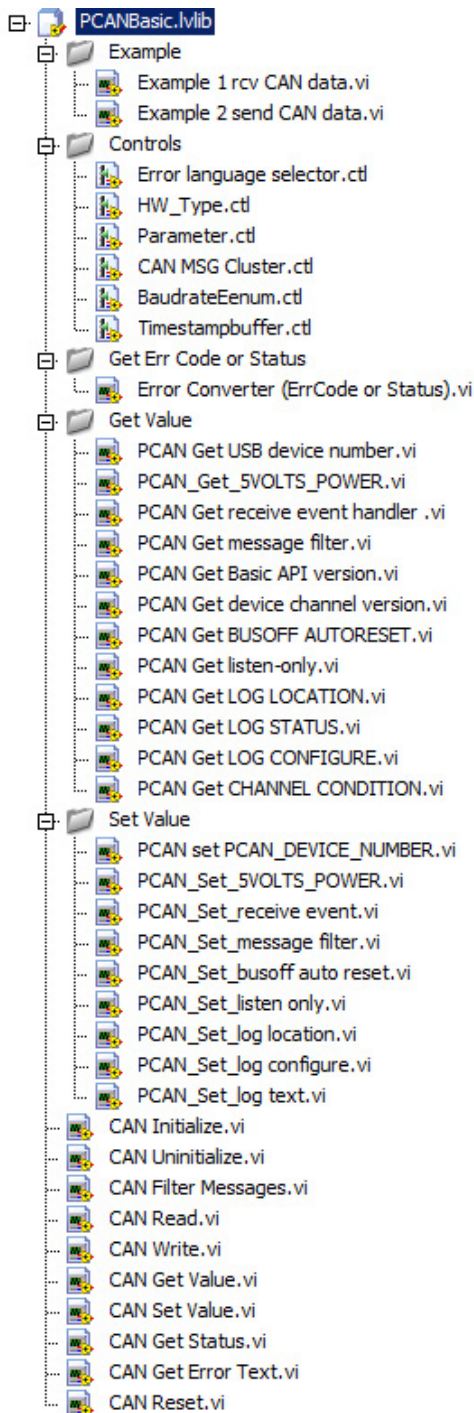


PCAN Basic API for LabVIEW 2009

Purpose and scope: Design of an easy to use LabVIEW API for the PCAN-Basic Interface DLL.



Examples folder:

Two examples are included to demonstrate how to send and receive CAN messages.

Controls folder:

Some strict typed def. ENUM- and cluster-controls are defined to simplify usage. They are all created according to the PCAN API description.

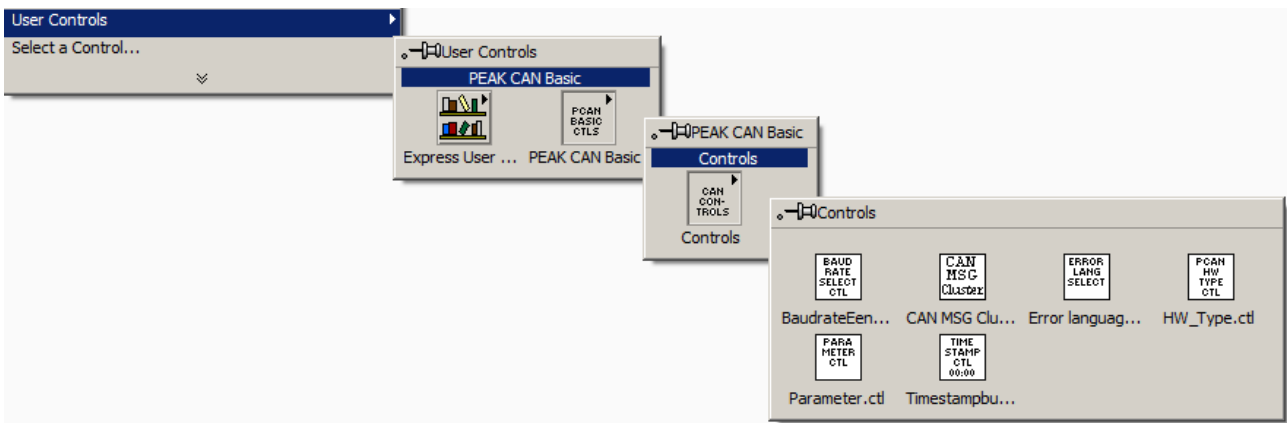
Get and Set Value VIs are stored in separated directories while there functionalities of input / output values are different according to the requested / set value. For easy usage they are combined into two polymorphic VIs called 'CAN Get Value.vi' and 'CAN Set Value.vi'

The top level VIs are shown on the left hand side:

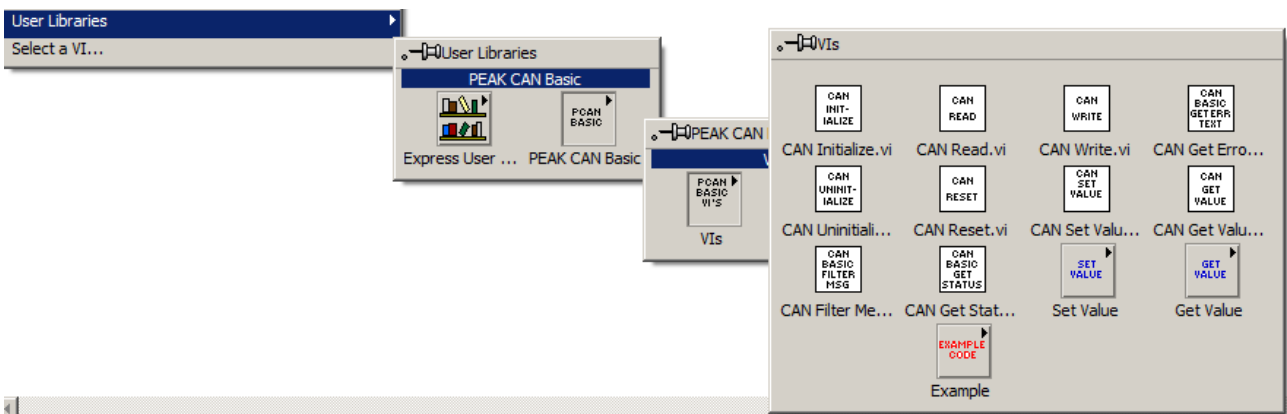
Each and every Vi has a context help for the purpose, usage, and return values.

All VIs and controls reside within the LabVIEW User.lib

Access the PCAN-Basic controls by opening the user lib from the front panel.



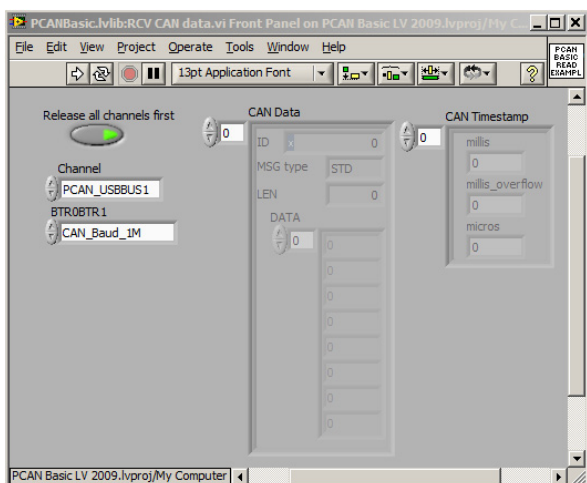
Access the PCAN Basic Vis by opening the user lib from the block diagram.



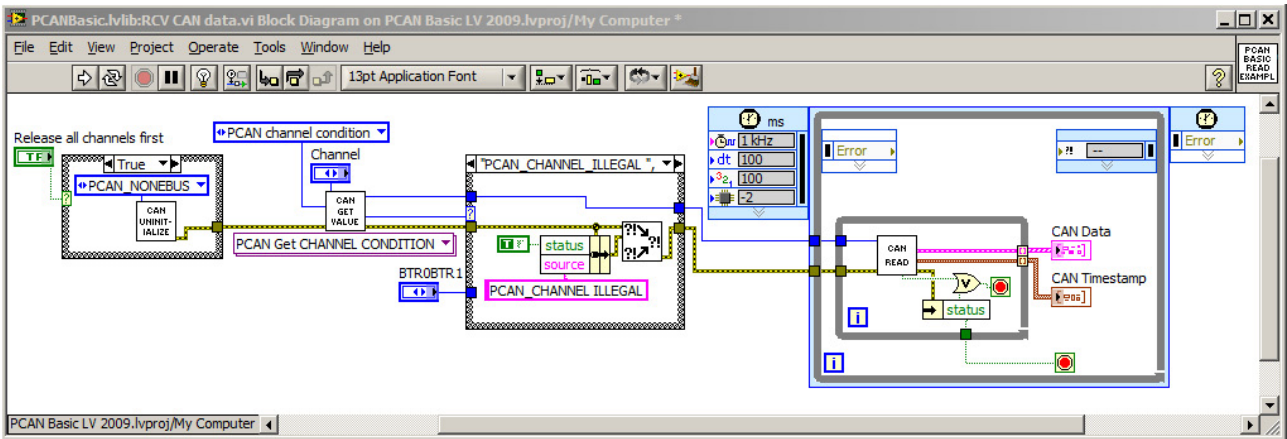
Top level VIs are accessible from here. Vis for set and get value are separated into sub pallets due to the fact that they are not needed very often while two polymorphic top level Vis are available to call these VIs.

Additionally there is a sub pallet available for the example Vis. The examples are created to demonstrate how to receive and send data using the API.

Receive data example:

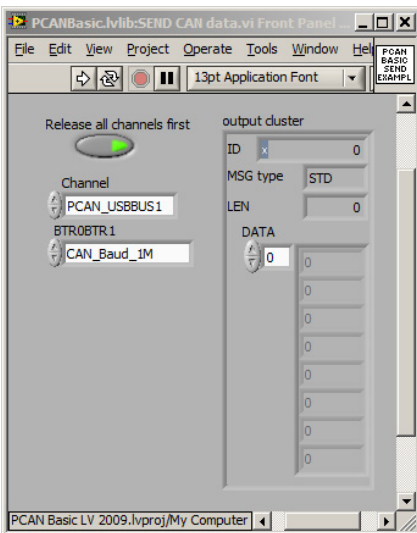


On the left hand side are the input variables like 'release all channels first' up to BTR0BTR1 listed. Outputs are CAN Data array and CAN Timestamp array.



The block diagram illustrates the API usage. The inner loop, on the right hand side, is ended if an error occurs or the receive queue is empty. Receive queue empty is not an error and will be suppressed by CAN READ. While receive queue empty doesn't lead to an abort of the loop the occurrence of an real existing error will stop the program. In this case the user has to decide what to do if an error occurs. The example just demonstrates what could be and not what should be done.

Send Data example:



On the left hand side the input controls are listed while on the right hand side the send data cluster is visible.

