Technical Note

A000T0021, valid for LINUX PCs

MRa, BRo, ASt, 2016-08-23

Working with PI Controllers under Linux Operating Systems

This document gives a summary of notes regarding the usage of Physik Instrumente software on Linux operating systems. Some software features still have to be adapted to Linux operating systems. The list below shall be understood as temporarily since we permanently improve the Linux software. Please write to info@pi.de to find out the current availability of the features.

Legal Information

Software products provided by PI may incorporate and/or make use of third-party software components. For more information, please refer to the Third Party Software Note linked below.
General Software License Agreement
Third Party Software Note

General Limitations

- **GPIB Interface:**
  The GPIB (General Purpose Interface Bus) interface can only be used with our GCS LabVIEW driver set. The GCS library does not yet support the GPIB interface.
- **Write permissions:**
  If you want to modify files which are not located in your home directory, you first have to change the write permission for these files (e.g. for LabVIEW vi files).

Limitations for LabVIEW

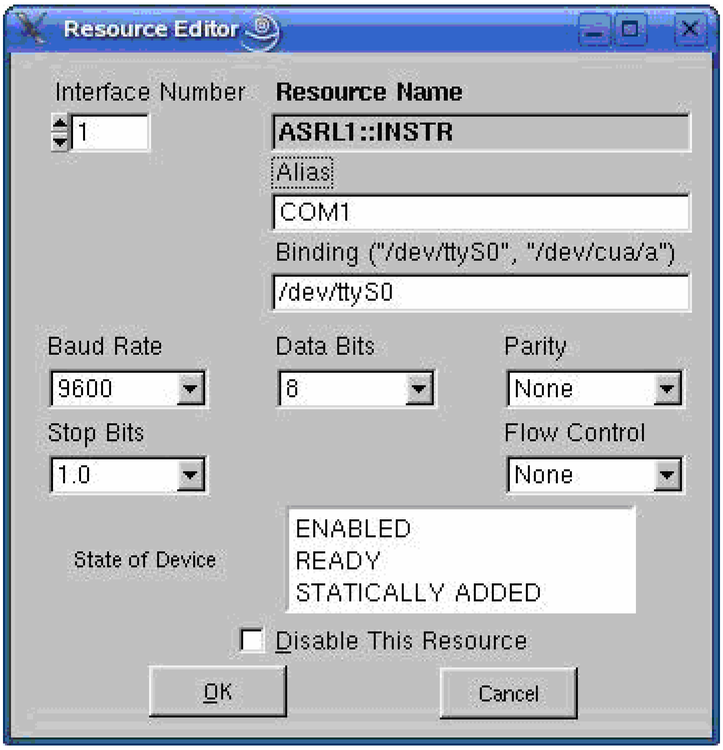
- **Linux Distributions** supported by National Instruments:
  National Instruments supports the following Linux distributions (for 32-bit x86 platforms only):
  - Red Hat Enterprise Linux WS
  - SUSE Linux
- **Analog GCS LabVIEW driver set**:
  These drivers have not yet been tested and are therefore not supported.

- **libPIGraph3D**:
  The libwxPIGraph3D.so shared object is not yet able to run on all of the most common Linux distributions. If you want to use vi files which make use of libwxPIGraph3D.so (eg. 2D Scan.vi and Show_Save_Load_ XYZ_Data.vi), please look for software updates on our website or contact our customer service.
- **Using Serial Ports in LabVIEW under Linux**:
  When you are working with LabVIEW on an Linux PC, you have to enable the serial ports before you can use them. To do this, use the visaconf tool which is installed with LabVIEW. By default, it is located in

  /usr/local/vxipnp/linux/NIvisa/visaconf

  Proceed as follows:

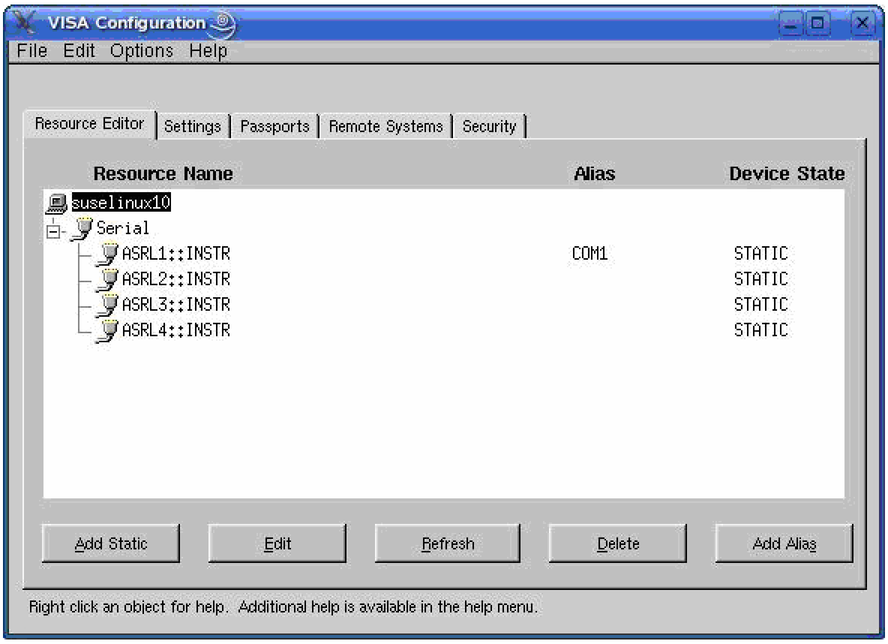
  1. Open the visaconf tool.
  2. In the "VISA Configuration" window, open the "Resource Editor" tab. Select the serial port you want to configure, and click the ***Edit*** button.
     The "Resource Editor" window opens.
  3. In the "Resource Editor" window, enter COMn in the "Alias" field, where n is the number of the COM port (n is arbitrary, but there has to be a COMn entry in the list of interfaces in the LabVIEW connection dialog). We recommend you use COM1.



Interface Number: 1
Resource Name: ASRL1::INSTR
Alias: COM1
Binding ("/dev/ttyS0", "/dev/cua/a"): /dev/ttyS0
Baud Rate: 9600
Data Bits: 8
Parity: None
Stop Bits: 1.0
Flow Control: None
State of Device: ENABLED
READY
STATICALLY ADDED
□ Disable This Resource

Leave all other settings unchanged and confirm the selection with ***OK***.

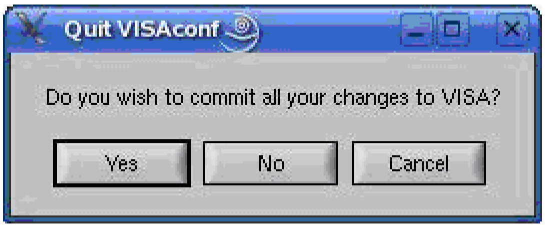
4. In the "VISA Configuration" window, check the assignment of ASRLm::INSTR to COMn on the "Resource Editor" tab.



Resource Name	Alias	Device State
suselinux10		
Serial		
ASRL1::INSTR	COM1	STATIC
ASRL2::INSTR		STATIC
ASRL3::INSTR		STATIC
ASRL4::INSTR		STATIC

Right click an object for help. Additional help is available in the help menu.

5. Close the visaconf tool and, when prompted to confirm your changes, click ***Yes***.



Do you wish to commit all your changes to VISA?

Technical Note

A000T0021, valid for LINUX PCs

MRa, BRo, ASt, 2016-08-23

Running PIMikroMove on Linux Systems

With **Wine** (**W**ine **I**s **N**ot an **E**mulator), a Windows-compatible runtime environment for POSIX-compatible systems, PIMikroMove can be run on Linux systems (tested with version 1.6.2 on ubuntu 14.04).

1. Install Wine.
   With popular Linux distributions, Wine can be directly installed from the standard packet sources.
2. Run the PI setup file for Windows from the product CD.

   In PIMikroMove, the communication between controller and PC via **TCP/IP** should immediately work.

   The communication via native **USB** is not possible with this set-up. For controllers with an FTDI USB implementation, a virtual COM port can be used. In this case proceed with step 3.
3. If you want to use **RS-232** communication:
   i. Open "/home/<user>/.wine/system.reg".
   This is the configuration file for the HKEY_LOCAL_MACHINE key of the Windows registry file. The serial interfaces have to be added to the file, since PIMikroMove gets the available interfaces from this registry key.

   ii. Add the serial interfaces.
   Example:
   ```
   Hardware\\Devicemap\\Serialcomm 1231984861
   "DeviceSerial0"="COM1"
   "DeviceSerial1"="COM2"
   "DeviceSerial2"="COM3 FTDI USB 1"
   "DeviceSerial3"="COM4 FTDI USB 2"
   ```

   The "DeviceSerialX" keys refer to the actual interfaces. Their values (the text following the = sign) determine by which name they are shown in the selection lists in PIMikroMove.

   iii. Create the soft links to the actual devices, in /home/<user>/.wine/dosdevices.
   For example:
   ```
   ln -s /dev/ttyS0 /home/<user>/.wine/dosdevices/com1
   ln -s /dev/ttyS1 /home/<user>/.wine/dosdevices/com2
   ln -s /dev/ttyUSB0 /home/<user>/.wine/dosdevices/com3
   ln -s /dev/ttyUSB1 /home/<user>/.wine/dosdevices/com4
   ```

   The "comX" entries correspond to the "DeviceSerialX" keys.

   Note: Under Wine, the "baud rate scan" of the DLL does not work. This means that the baud rate set in the controller has to be used.