Operation of the instrument using Matlab®

The USB driver (CDM20814_Setup.exe) has to be installed on your system and the Novoptel instrument needs to be connected using a USB cable. Examples of Matlab communication scripts can be downloaded from http://www.novoptel.de/Matlab_support_files.zip

Access the USB driver

Matlab needs a header file like *ftd2xx.h* from FTDI to access the driver. In the provided versions *matftd2xx.h* and *matftd2xx64.h,* we have modified the data types to be compatible to Matlab.

You will find help about communicating to a driver at http://www.mathworks.com/help/techdoc/ref/loadlibrary.html

The different functions of the driver can be seen from the header file. Information about each function is provided at http://www.ftdichip.com/Support/Knowledgebase/index.html

USB Settings

The following settings have to be applied to enable USB communication. They will be applied automatically if you use the example communication scripts.

Baud Rate	230400 baud
Word Length	8 Bits
Stop Bits	1 Bit
Parity:	0 Bit

To speed up sequential read and write operations, we recommend to set the ***USB Latency Timer*** to 2 ms.

Example communication scripts

The following four Matlab code examples allow basic communication with a Novoptel instrument:

- initdevice.m
- readdevice.m
- writedevice.m
- closedevice.m

To expand the communication to more than one Novoptel instrument, the four files have to be copied and renamed. Additionally, the name of the instruments device descriptor and the name of the handle have to be changed inside the scripts.

Example: Copy and rename initdevice.m to get the files initEPS.m and initEPC.m. In initEPS.m, put the instrument name of the EPS for the device descriptor and rename the handle to EPSHandle. In initEPC, put the instrument name of the EPC for the device descriptor and rename the handle to EPCHandle. Do the same with the other three files. Now, both instruments can be kept connected at the same time and read and written to individually.

The usage of the four scripts is described in the following:

initdevice.m (calls ***initusb.m, FindUsbDevice.m, matftd2xx64.h*** and ***matftd2xx.h***):
This script has to be called before the first write or read attempt. To be able to select a specific device, the Device Descriptor, e.g. "`EPS1000D-10M-C-AA-USB SN00000`", has to be specified at the beginning of *initdevice.m*. If the specified device is not found, all connected and unconfigured devices will be listed. The matching device descriptor can be copied from the list and pasted into *initdevice.m*. Devices opened by other programs, e.g. Novoptel's GUI, will not be listed.

readdevice.m (calls readusb.m):
Reads a value from a given register.
Example: `res=readdevice(addr)` or `[res, ok]=readdevice(addr)`

writedevice.m (calls writeusb.m):
Writes a given value into a given register.
Example: `ok=writedevice(addr, data)`

closedevice.m (calls closeusb.m):
Once a device is allocated by Matlab, it cannot be accessed by other programs, e.g. Novoptel's GUI, until it is deallocated by calling the script *closedevice.m.*