BiDiB-Wizard Spy

Ab Build 3013 vom BiDiB-Wizard gibt es das Tool *BiDiB-Wizard Spy* um die BiDiB-Messages über eine serielle Schnittstelle tracen zu lassen. Dabei wird mit com0com eine virtuelle serielle Schnittstelle zur Verfügung gestellt, auf welche sich das Steuerungsprogramm verbindet. Der Spy leitet dann die Meldungen an das BiDiB-Interface (GBMBoost, IF2) weiter und zurück.

Installation und Konfiguration com0com

Es gibt von com0com einen aktuellen Installer mit signierten Treibern auf der SourceForge-Seite (https://sourceforge.net/projects/com0com/files/com0com/3.0.0.0/): https://kent.dl.sourceforge.net/project/com0com/com0com/3.0.0.0/com0com-3.0.0.0-i386-and-x64-signed.zip

Anschliessend muss man diesen Virtuellen COM-Port im System konfigurieren. Man kann es über ein UI-Tool konfigurieren oder über Console (setupc.exe).

Mit den nachfolgenden Einstellungen kann man sich mit dem Wizard über die Spy-Applikation (plain messages, COM rein, COM raus) verbinden.

Konsole von com0com mit setupc.exe (im Installationsverzeichnis von com0com, bei mir C:\Program Files (x86)\com0com) starten.

Erzeugen der virtuellen com0com Port CNCA0 und CNCB0. CNCA0 wird als COM13 umbenannt und ist nicht sichtbar solange CNCB0 nicht geöffnet ist. Sobald CNCB0 geöffnet wird, ist COM13 für Applikationen sichtbar. Wird CNCB0 geschlossen, wird auch COM13 nicht mehr sichtbar.

Eingabe in Konsole

```
command> install - -
       CNCA0 PortName=-
       CNCB0 PortName=-
command> change CNCB0 ExclusiveMode=yes
       CNCA0 PortName=-
       CNCB0 PortName=-
change CNCB0 PortName=-,ExclusiveMode=yes
Restarted CNCB0 com0com\port \Device\com0com20
command> change CNCA0 PlugInMode=yes
       CNCA0 PortName=-
change CNCA0 PortName=-,PlugInMode=yes
Restarted CNCA0 com0com\port \Device\com0com10
       CNCB0 PortName=-,ExclusiveMode=yes
command> change CNCA0 PortName=COM13
       CNCA0 PortName=-,PlugInMode=yes
change CNCA0 PortName=COM13,PlugInMode=yes
Restarted CNCA0 com0com\port \Device\com0com10
       CNCB0 PortName=-,ExclusiveMode=yes
```

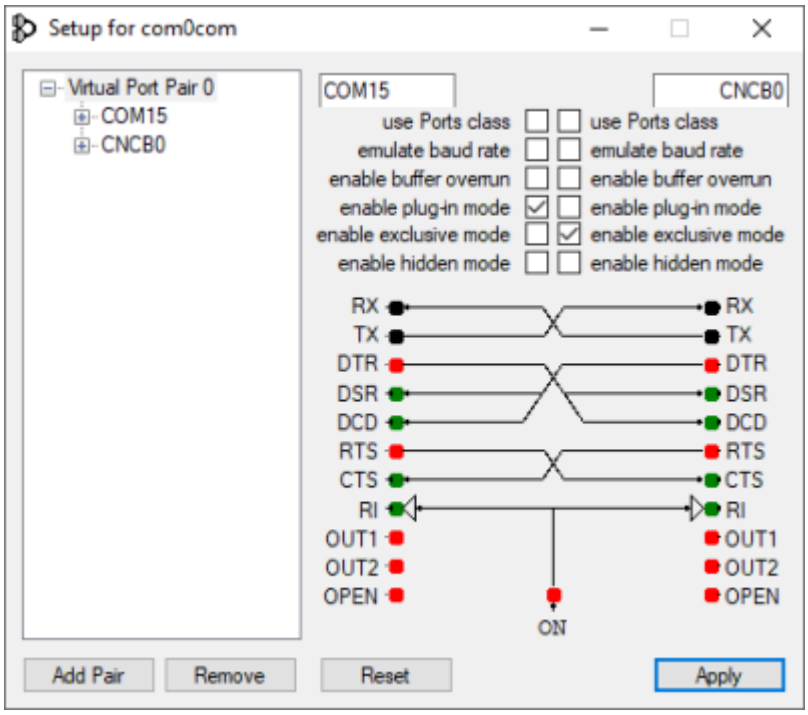
```
ComDB: COM13 - logged as "in use"
command>
```

```
Fertig in Konsole
```

Jetzt sollten im Device Manager von Windows die com0com Ports angezeigt werden.

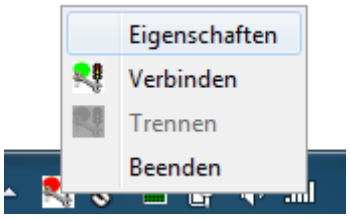
Konfiguration über das Setup for com0com

Erzeugen der virtuellen com0com Port CNCA0 und CNCB0. CNCA0 wird als COM15 umbenannt und ist nicht sichtbar solange CNCB0 nicht geöffnet ist. Sobald CNCB0 geöffnet wird, ist COM15 für Applikationen sichtbar. Wird CNCB0 geschlossen, wird auch COM15 nicht mehr sichtbar.

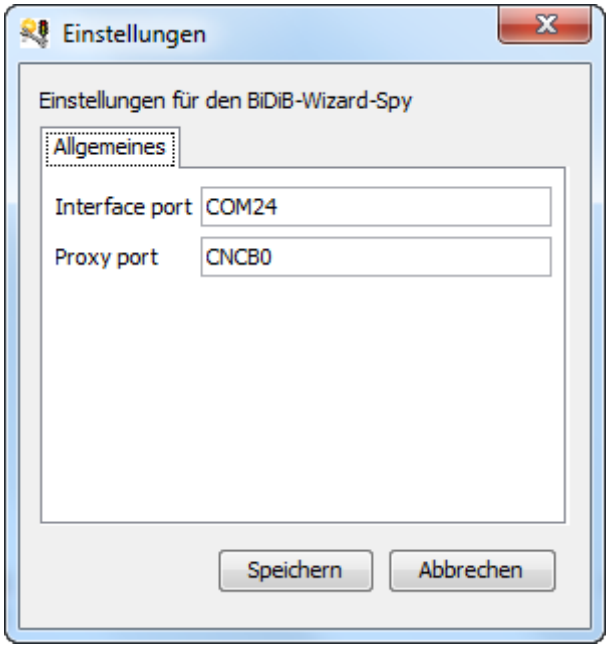


Konfiguration Spy

Nach dem Starten des Spy muss ide Konfiguration über das Kontext-Menü im Tray geöffnet werden.



Zuerst muss in den Einstellungen den Port vom BiDiB-Interface (GBMboost, IF2, o.ä.) als „Interface Port“ eintragen (war bei mir COM24). Dann den „Proxy Port“ welcher von com0com verwendet wird, z.B. CNCB0 wenn die obige Anleitung verwendet wird.



Anschliessend muss über das Kontextmenü vom Tray „Verbinden“ ausgewählt werden. Dann sollte der virtuelle COM-Port COM13 für das Steuerungsprogramm (BiDiB-Wizard oder iTrain o.ä.) sichtbar werden. Der rote Punkt im Tray-Icon ändert sich in einen grünen Punkt wenn es geklappt hat.

Dann in iTrain den COM-Port auf COM13 umstellen und verbinden. Die Verbindung läuft jetzt über den Spy und die Meldungen sind in *BiDiBWizard-Proxy-RXTX.log* sichtbar. Diese Logfile liegt am gleichen Ort wie die Logfiles vom BiDiB-Wizard, haben aber *Proxy* im Namen.

Die Meldungen sehen dann so aus:

```
25.11.2017 14:57:31.582: [receiveQueueWorker] - >>
MSG_SYS_DISABLE[[0],num=0,type=4,data=[]] : 03 00 00 04
25.11.2017 14:57:31.583: [receiveQueueWorker] - >>
MSG_SYS_GET_MAGIC[[0],num=1,type=1,data=[]] : 03 00 01 01
25.11.2017 14:57:31.602: [receiveQueueWorker] - <<
MSG_SYS_MAGIC[[0],num=0,type=129,data=[254, 175]] : 05 00 00 81 FE AF
25.11.2017 14:57:33.129: [receiveQueueWorker] - >>
MSG_NODETAB_GETALL[[0],num=2,type=11,data=[]] : 03 00 02 0B
25.11.2017 14:57:33.136: [receiveQueueWorker] - <<
MSG_NODETAB_COUNT[[0],num=1,type=136,data=[1]] : 04 00 01 88 01
25.11.2017 14:57:33.144: [receiveQueueWorker] - >>
MSG_NODETAB_GETNEXT[[0],num=3,type=12,data=[]] : 03 00 03 0C
25.11.2017 14:57:33.151: [receiveQueueWorker] - <<
MSG_NODETAB[[0],num=2,type=137,data=[1, 0, 144, 0, 13, 132, 0, 33, 0]] : 0C
00 02 89 01 00 90 00 0D 84 00 21 00
25.11.2017 14:57:33.158: [receiveQueueWorker] - >>
MSG_SYS_GET_MAGIC[[0],num=4,type=1,data=[]] : 03 00 04 01
25.11.2017 14:57:33.164: [receiveQueueWorker] - <<
MSG_SYS_MAGIC[[0],num=0,type=129,data=[254, 175]] : 05 00 00 81 FE AF
25.11.2017 14:57:33.177: [receiveQueueWorker] - >>
MSG_SYS_CLOCK[[0],num=5,type=24,data=[0, 134, 69, 193]] : 07 00 05 18 00 86
45 C1
25.11.2017 14:57:33.283: [receiveQueueWorker] - >>
MSG_SYS_GET_MAGIC[[0],num=6,type=1,data=[]] : 03 00 06 01
```

```
25.11.2017 14:57:33.289: [receiveQueueWorker] - <<
MSG_SYS_MAGIC[[0],num=0,type=129,data=[254, 175]] : 05 00 00 81 FE AF
25.11.2017 14:57:33.299: [receiveQueueWorker] - >>
MSG_SYS_GET_SW_VERSION[[0],num=7,type=6,data=[]] : 03 00 07 06
25.11.2017 14:57:33.305: [receiveQueueWorker] - <<
MSG_SYS_SW_VERSION[[0],num=1,type=133,data=[3, 4, 2]] : 06 00 01 85 03 04 02
25.11.2017 14:57:33.310: [receiveQueueWorker] - >>
MSG_SYS_GET_P_VERSION[[0],num=8,type=2,data=[]] : 03 00 08 02
25.11.2017 14:57:33.316: [receiveQueueWorker] - <<
MSG_SYS_P_VERSION[[0],num=2,type=131,data=[6, 0]] : 05 00 02 83 06 00
25.11.2017 14:57:33.321: [receiveQueueWorker] - >>
MSG_FEATURE_GETALL[[0],num=9,type=16,data=[]] : 03 00 09 10
```

Am Ende der Zeile ist immer die Meldung in HEX-Darstellung. Die Doppelpfeile zeigen die Richtung an:

```
>> vom Programm zum Interface
<< vom Interface zum Programm
```

Linux: interceptty

Unter Linux besteht die Möglichkeit den Datenverkehr auf der seriellen Schnittstelle mit *interceptty* protokollieren zu lassen. Dazu kann das Repo `https://github.com/akuhtz/interceptty` gecloned werden. Anschliessend müssen die Schritte im README ausgeführt werden:

- ./configure
- make
- sudo make install

Anschliessend kann der serielle Port mit folgendem Kommando getraced werden:

```
sudo ./interceptty -l -s 'ispeed 115200 ospeed 115200 raw' -u pi
/dev/ttyAMA0 /dev/ttyBidib | interceptty-nicedump
```

Die Angabe von `-u pi` setzt die Berechtigung für den User `pi` mit welchem der Wizard gestartet werden muss. Der Link `dev/ttyAMA0` ist der (echte) serielle Port welcher getraced werden soll. Der Wizard muss sich auf den Link `dev/ttyBidib` verbinden.

 Als Serieller Port-Treiber SPSW oder SCM verwenden.

From:
https://forum.opendcc.de/wiki/ - **BiDiB Wiki**

Permanent link:
https://forum.opendcc.de/wiki/doku.php?id=wizard:spy

Last update: **2020/08/16 09:48**