DE | EN

Probing Solutions.
Made in Germany.

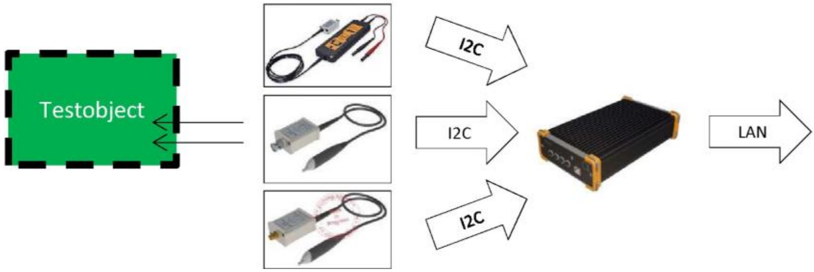
REMOTE CONTROL EXAMPLE

for Sonic®, HSDP2000 and BumbleBee® Series Active Probes

Programming Example

Overview

This document is an example for remote control the three probes BumbleBee®, Sonic® and HSDP2000 series via LAN connection, where Sonic® and HSDP2000 probe use an identic communication protocol. A schematic of connections and communication overview with the probes is shown on the following chart:



The communication with the device should be implemented as a driver. The "BumbleBeeCom-municationExample" can act as protocol specification. Note, with Sonic® and HSDP2000 series only the active offset can be configured.

EN

The user can configure the behavior of the probes by entering commands to the built in microcontroller (BumbleBee®) or to configurable circuits inside (Sonic and HSDP). There are different ways to enter user commands:

- A Keyboard (BumbleBee®)
- The XPORT TCP connection (LAN) (All three)
- The pluggable USB cable (All three)

The LANTRONIX LAN-to-UART converter

BumbleBee® and other probes can be controlled via a serial data stream protocol. This protocol is converted by the XPORT device in the power supply PS-02/03 between the internal UART protocol and the external TCP/IP protocol used within the LAN. The user may write a command word into the TCP-socket and may get the response via the same socket. Because XPORT obtains its IP-Ad- dress from the DHCP server in the LAN the user needs a way to request this address with witch a connection to XPORT can be made.

Address by LANTRONIX Device Installer

Connect the power supply with the LAN. Then open the LANTRONIX Device Installer software. Click the “Search” symbol at the command bar. Navigate to one of the XPort-Devices and open “Web configuration”. If the device is marked as not connected (RED) switch off the power supply, switch it on and try again. Click at the HOME-Symbol of the embedded web browser control and change the file at the end of the URL to “/PowerSupplyMetaData.xml”. If the XPORT belongs to a

PS-02/03 the following file should appear:


```
<?xml version="1.0" ?>
<MetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <EEPROMLayoutRev>1.1</EEPROMLayoutRev>
  <SerialNumber>0117</SerialNumber>
  <Manufacturer>http:/   /www.pmk.de</Manufacturer>
  <Model>PS-02</Model>
  <Description>2 Port Power Supply</Description>
  <ProductionDate>20141105</ProductionDate>
  <CalibrationDueDate>-</CalibrationDueDate>
  <CalibrationInstance>-</CalibrationInstance>
  <HardwareRev>Hard Coded</HardwareRev>
  <SoftwareRev>Hard Coded</SoftwareRev>
</MetaData>
```


The listing contains the serial number of the device. Use this number to identify a single device among a number of devices.

Address by UDP-Broadcast

First send a broadcast with the content bytes {0x00, 0x00, 0x00, 0xf8} to the port 30718 of the local network. Then wait for incoming packets until a timeout of some seconds is reached. If the response packet contains at begin the bytes {0x00, 0x00, 0x00, 0xf9} then this is an XPORT answer. Try to download the Document “PowerSupplyMetadata.xml” from the built-in webserver within the remote XPORT. If a download is possible a PS-02/ 03 is present. Within the *.xml file you can find the serial number of the device which is also printed on the housing.

Telnet command line example

Use telnet to test the connectivity. Plug in the LAN into the PS-02/03. Connect also one Bumble-Bee® probe on the left plug. Provide power to the device. Fetch the IP-Address with the LANTRONIX device installer. Then open a windows telnet session:

```
$>c:\windows\system32\telnet
#>unset crlf
#>set localecho
#>open <IP> 10001 // 10001 is the TCP-Port. Do not type enter in this session.
#>CTRL+B WR104W0118020002 CTRL+V CTRL+C // Increment attenuation ratio.
  Without space.
#>CTRL+B WR104W0118020102 CTRL+V CTRL+C // Decrement attenuation ratio.
  Without space.
#>CTRL+] on QUERTY or CTRL++ on QUERTZ // Comment: Escape symbol to go to the
  command mode.
#>close
#>quit
```

If it does not work then switch power off and on and try again.
Where CTRL+B is the escaped STX character and CTRL+V followed by CTRL+C is the escaped ETX character.

“WR104W0118020002” means:

Write (“WR”) to the plug “1” to the I2C-address “04” in word-mode (“W”, address is two bytes long) to the device address “0118” with the length “02” the command “0002” where “00” is the command value and “02” is the command (here “decrement” and “switch attenuation ratio”).

Usage hints of the command line example

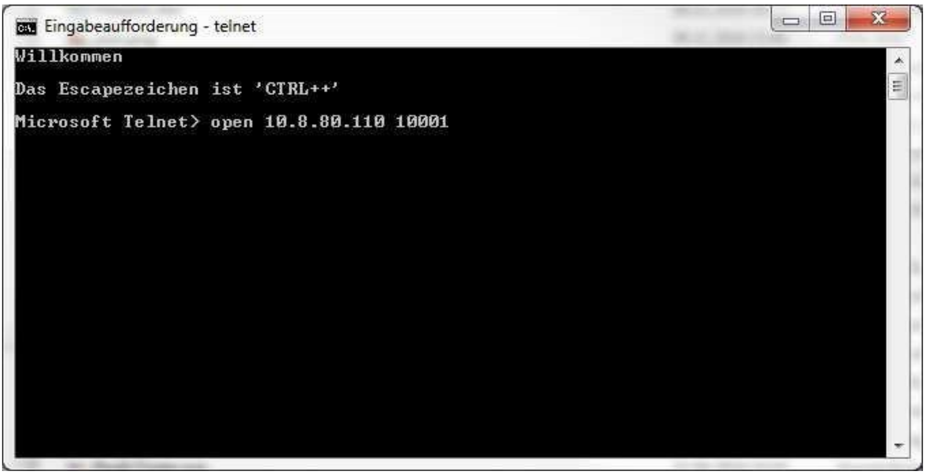
The command line example is intended only as a quick start guide or a second access. At one side the probe system will be controlled by machine on the other hand telnet supports control characters like a type writer (CR and NL) where the probe system uses the STX – ETX control character pair.

The particular steps of the example:

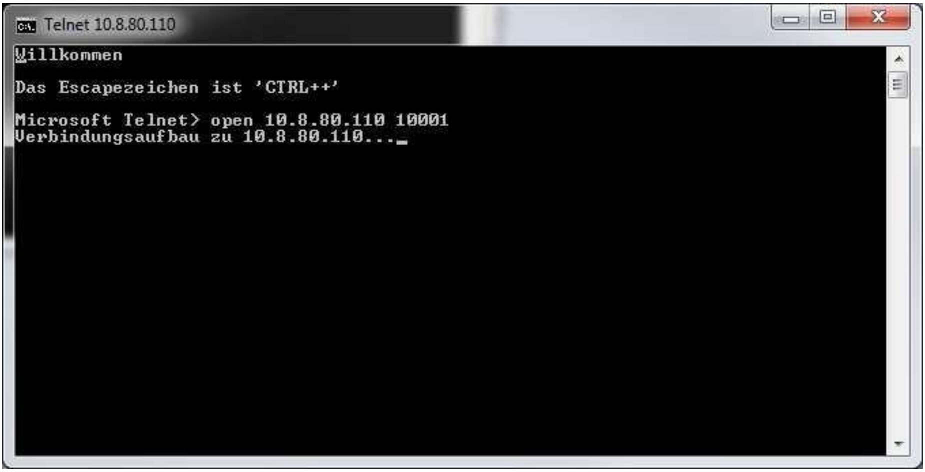
EN

1) Open a DOS-Box and make sure that the path of telnet is available. Normally telnet resides under c:\windows\system32\ so you can change the directory to this path. Enter the telnet command.

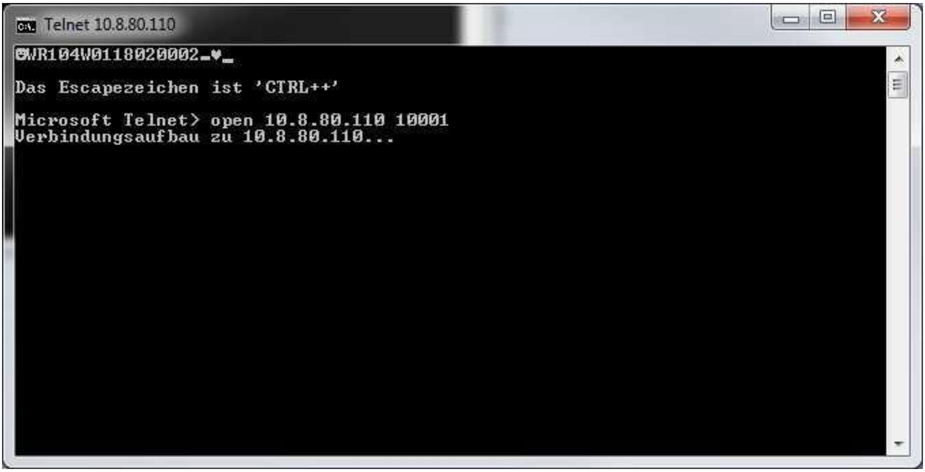
2) Telnet has two modes: Command mode and data stream mode. When started without an address it offers the command mode. On the third line telnet writes its ESCAPE character which is culture specific (different between English Japanese and German). Here this character is the CTRL key and the PLUS key pressed at the same time. A switch from the command mode to the stream mode is made by the open command. Then telnet will open a TCP socket and it will begin to transfer and display stream data in both directions. A switch from the stream mode back to the command mode will be needed to end the connection or to leave a blocked connection where the server does not respond. This is made when the user enters the ESCAPE character into the tel- net client. ESCAPE is the only character which telnet does not recognize to belong to the payload data. Do not forget to unset CRLF and to switch localecho on before open the connection.



3) Open the connection to the PS-02/03 within telnet. Normally the server will send now control characters to format the text in the DOS-box. On PS-02/03 this is not the case because the de- vice is built for machine interaction. It does not send a prompt so that it looks like the server did not respond.



4) Please ignore the missing response and enter the command to change the attenuation ratio mode from the example.



As you can see telnet prints your text in the box but instead at the current line it appears on the top of the box. STX and ETX are visualized as card game symbols. CTRL+V is used to mask a special meaning of CTRL + C. CTRL+V followed by CTRL+C together will send an ETX. When working correctly the attenuation ratio index will increase (for example from 500:1 to 250:1).

5) The response shown in telnet will consist of a number of characters.

Please keep in mind that telnet supports the ENTER key “->|” in the command mode but not in stre- am mode. Here the TCP server is responsible to interpret control characters. In the case of PS02/03 the ENTER key is not supported and will lead to a not responding device.

In case of a blocking PS-02/03 please switch the device off and on again and start a new session. Consider that the used XPORT (converter between UART and TCP) will at one side support multiple sessions when reading HTML-pages from it but that it will support only a single TCP client at one time when using the command access to the power supply on port 10001.

Command interaction

A command and its response are enclosed in the STX/ETX ASCII control character pair. The com- mand may be and the response is terminated by the CR- control character. All numbers are Hex- bytes as ASCII-strings.

In case of a write command this command consists of the following parts:

“WR” - Write prefix

“0…4” - 0 for the power supply itself and 1…4 for the device connected on the numbered plug “04” - the I2C-address of the probe

04 … BumbleBee,

50 … Sonic/HSDP2000 metadata

52 … Sonic/HSDP2000 global offset ADC (Write only, use 0x30 as byte wise address).

“W” – Address is two bytes long (“B” Address is one byte long with a dummy byte as prefix) “0000” - Start address of the bytes to write within the address space of the device. “01” - Length of the block with the bytes to write. “….” - A number of bytes.

At one side several parameters may be written to the address space. On the other hand a device command may be written to the addresses 0x0118 and 0x0119. A number of parameters will only become active if they are followed by a device command. The power supply will answer with a short response message. In case of success the first character is the ACK ASCII character. Otherwise a NACK will be sent.

In case of a read-command this command consists of the following parts:

“RD” - read prefix
“0…4” - Number of the plug, see write command.
“04” - I2C address, see write command.
“W” - Word address mode, see write command.
“0000” - Address, see write command. “01”
- Length, see write command.

The power supply will respond to such a message with an ACK character as first character in case of success and with a NACK otherwise. In case of success the device will also append a payload with the requested data. The payload will begin at byte index 9 (nine), it will contain the specified count of bytes as 2-character-ASCII representation.

The BumbleBee® communication Example

This example can act as a driver or as a source to port the commands to another programming language. Power supply, BumbleBee®, Sonic and HSDP2000 can be controlled where the protocol of Sonic and HSDP2000 are the same. If used in an end user application the application should com- pare the model of the device in the MetaData-property with the intended device.

EN

How to use the example

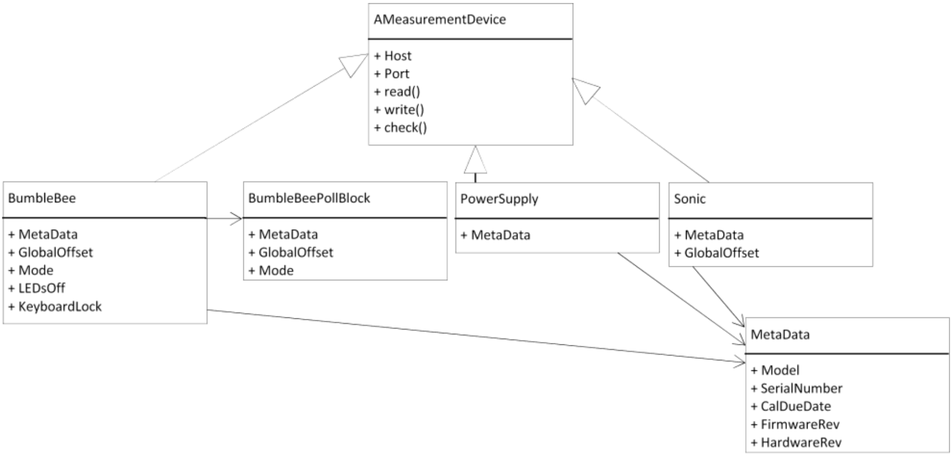
Build a Visual Studio project with the provided example files as console application. Connect the hardware at first. Then activate the #define of the connected probe (for example #define Bumble- Bee®) in the main entry file of the example (“Program.cs”). The program will show the metadata of the connected probe and it will toggle the attenuation ratio mode in case of a chosen BumbleBee-probe.

Classes of the example

Communication commands are encapsulated in the classes within the “Dev_...” – folders of the pro- ject. Properties and public functions interact with the hardware. To navigate within them please fold all regions of the file by the key combination CTRL+M followed by CTRL+A. Then open the closed regions one by one. Be aware that the properties, which send data to the hardware, will consume some time when assigned or read. All devices provide their metadata have to be read as block. An object is returned with a number of properties to be read without time consumption for communication. In the case of the BumbleBee probe the I2C bus is not fast enough to poll all keyboard data in real- time. Therefore this data are provided a second time as a “PollBlock.cs” - object with readable pro- perties similar to the metadata.

The “DeviceStream” – folder provides a TextReader - and a TextWriter - object with access to the TCP-socket of the hardware. The classes may be changed by an own implementation. The interface “IDevice” is used within the “AMeasurementDevice” - class in the send()- and receive()- function pair.

Driver classes



List of commands (BumbleBee®)

Values sent to the internal address space of the Sonic/HSDP2000 probe are executed directly. In case of BumbleBee® values sent to the address space of the BumbleBee® probe. Those written values have to be completed by an executing command. The table below shows commands with a probe on plug "1" (WR104W00118020B05). The number of the plug has to be adapted to the intended device during runtime. Let the BumbleBee® probe 100 milliseconds time to execute the command before sending the next memory variable or command.

Changed variable	Command	Remark
GlobalOffset	WR104W0118020205	Set the value of the DAC out.
OffsetStep<Small, Large, ExtraLarge>	-	Do nothing.
Mode	WR104W0118020105	Set attenuation ratio.
RGBColor	WR104W0118020305	Device marker color LED.
SyncOffset	-	Not needed.
KeyboardBuzzer	WR104W0118020A05	Behavior options.
OverloadBuzzer		
HoldOverload		
KeyLock	WR104W0118020B05	
LEDsOff		
<Clear overload counter>	WR104W0118020C05	Set overload events to zero.
<Factory reset>	WR104W0118020E05	Use default configuration. Other than common commands a delay of 3000 milliseconds has to be inserted.
<Mode inc>	WR104W0118020002	Cyclic mode iteration.
<Mode dec>	WR104W0118020102	

<OffsetStepSmall inc>	WR104W0118020103	Increment / decrement of the global offset by the stored offset steps.
<OffsetStepLarge inc>	WR104W0118020203	
<OffsetStepExtraLarge inc>	WR104W0118020303	
<OffsetStepSmall dec>	WR104W0118020603	
<OffsetStepLarge dec>	WR104W0118020503	
<OffsetStepExtraLarge dec>	WR104W0118020403	
…		

BumbleBee® configuration memory organization

Voltage values and digital voltage variables are scaled by 16.0 from floating point representation to fractional number representation. Gain variables are scaled by 16384 from floating point representation to fractional number (same meaning like fixed point number) representation.

Variable	Position	Meaning
Int16 GlobalOffset	0x0133	Output of the probe is shifted by amount (saturated).
Int16 OffsetStepSmall	0x0135	Incremental offset switch.
Int16 OffsetStepLarge	0x0137	
Int16 OffsetStepExtraLarge	0x0139	
Byte Mode	0x0131	Attenuation ratio {1… 4}.
Byte RGBColor	0x012C	{red = 0, green, blue, magenta, cyan, yellow, white, black = 7}.
Byte SyncOffset	0x012F	0x01 bit used only.
Byte OverloadBuzzerAndHoldOverload	0x012D	0x01 bit => OverloadBuzzer, 0x02 bit =>HoldOverload.
Byte KeylockAndLEDsOff	0x0130	0x01 bit => KeyLock, 0x02 bit => LEDsOff.
Byte KeyboardBuzzer	0x012E	0x01 bit used only.
Byte LEDOverloadAll	0x0132	Read only: 0x01 bit => Overload Pos, 0x02 bit => Overload Neg, 0x04 bit => Overload Main.
UInt16 OverloadPosCount	0x013B	Read only.
UInt16 OverloadNegCount	0x013D	
UInt16 OverloadMainCount	0x013F	
Int16 OffsetErrorCorrection500	0x0168	User offset error correction. If set then the device works out of specification. Can be used to shift the output to a customized range.
Int16 OffsetErrorCorrection250	0x016A	
Int16 OffsetErrorCorrection100	0x016C	
Int16 OffsetErrorCorrection50	0x016E	
Int16 FactoryOffsetErrorCorrection500	0x0178	Assign this to the offset error correction above to restore the factory setting.
Int16 FactoryOffsetErrorCorrection250	0x017A	
Int16	0x017C	

Variable	Position	Meaning
FactoryOffsetErrorCorrection100		
Int16 FactoryOffsetErrorCorrection50	0x017E	
Int16 GainErrorCorrection500	0x0170	Divide the user offset in fractional number representation by this read only value to convert it from raw to the voltage unit. Multiply by this number to convert from voltage to raw.
Int16 GainErrorCorrection250	0x0172	
Int16 GainErrorCorrection100	0x0174	
Int16 GainErrorCorrection50	0x0176	

Calculating the user offset error correction

The user offset shifts the offset error correction additional to the adjusted and calibrated factory offset error correction. It can be used to get a user demanded behavior. Set and get operations are called depending on the specific attenuation ratio mode.

u … User offset error correction (Intended offset shift). f … Factory offset error correction (Adjusted and calibrated factory value). o … Offset error correction (Offset shift which is actually set). g … Gain error correction (Convert between raw and voltage unit). u := (o – f) / g // Request to read the current value.

o := u * g + f   // Set another value which should be put out by the Hardware.

Metadata of the devices as defined in EEPROM layout revision 1.0

The metadata block is located at the beginning of the internal address space in all three devices so far (BumbleBee®, Sonic and HSDP2000). Its address is 0x0000 and its length is 0x82 (130) measured in ASCII bytes but not in the numerical 2-byte representation of such bytes (the length of the transmitted numerical representation is twice long). Metadata consist of ten strings as values of a predefined variable order. Each entry has to be terminated by a LF control character. The variable order is written here:

1. EEPROM layout rev \n
2. SerialNumber \n
3. Manufacturer \n
4. Model \n
5. Description \n
6. ProductionDate \n
7. CalibrationDueDate \n
8. CalibrationInstance \n
9. HardwareRev \n
10. FirmwareRev \n

HardwareRev of the BumbleBee® probe as Example (Mainboard, Keyboard): M2.0 K2.0

FirmwareRev of the BumbleBee® probe as Example (Mainboard, Keyboard): M3.7 K1.6

Note: Metadata should only be written on Adjustment and Calibration, a driver should not write it in user mode.

Version info of this document

Version	Author	Date
1.0	Claus Wimmer	2016-11-23
1.1	Claus Wimmer	2016-12-14
1.2	Claus Wimmer	2017-01-03

Related documents

[1] PMK: "BumbleBee 2000V Instruction Manual", 880-102-501_Manual BumbleBee_en.pdf

[2] LANTRONIX: "XPORT Device Server User Guide", www.lantronix.com/pdf/XPort_UG.pdf

EN

Notes

Manufacturer

PMK Mess- und Kommunikationstechnik GmbH
Koenigsteinerstrasse 98
65812 Bad Soden am Taunus, Germany

Phone: +49 (0) 6196 5927 - 930
Fax: +49 (0) 6196 5927 - 939

Internet: www.pmk.de
E-Mail: sales@pmk.de