

Application:

Access to serial port using PHP

In the following we explain how you can access the serial port on the pure.box using a simple PHP script, for example to control a weather station:

In many cases permanent communication with a PC having the controlling software is necessary when operating a serial terminal device. For example with a weather station having a serial connection in which a PC needs to be used as a data logger around the clock to record a continuous series of measurement. This often used configuration can be implemented at great cost, energy and space savings using the pure.box.

For example to poll and receive the current temperature once a minute requires just this small PHP script, which must be started cyclically by the pure.box.

```
<?php
$fd = dio_open( '/dev/ttyS1', O_RDWR );
dio_tcsetattr( $fd, array(
    'baud'      => 9600,
    'bits'      => 8,
    'stop'      => 1,
    'parity'     => 0,
    'flow_control' => 0,
    'is_canonical' => 0 ));
dio_write( $fd, 'temperatur?' );
$input = '';
while( strlen( $input ) <= 10 )
{
    $input .= dio_read( $fd, 1 );
}
echo $input;
dio_close( $fd );
?>
```

After the script is started the following actions are performed:

- The serial port on the pure.box 2 is opened for read and write access.
- The serial format is set to 9600 baud, 8 data bits, 1 stop bit, no parity, no hardware handshake.
- By setting the parameter `is_canonical` to 0 each character received on the serial port is sent directly to the PHP. If this parameter is set to 1, transmission only takes place after receipt of a line break character.
- To request the current temperature the string "temperature?" is sent to the connected device over the serial interface.
- 10 characters are received and then output.
- Finally the serial port is closed again.

The received temperature value can be then stored for example in the MySQL or SQLite database located in the box or passed to a processing server over the network.

Load this script via FTP or SMB to the folder **websites** in the pure.box, or you can run it for test purposes from the browser:

<http://<IP der pure.box 2>/websites/<Dateiname des Skriptes>>

You configure the cyclical execution by the pure.box after uploading in Web Based Management. To do this, navigate in the menu tree to **My files** and in **Properties** for the file enter a time-based execution.

The above script assumes that the reply has a fixed length. If that is not the case, the read routine can be exited after receiving a specified terminal character.

To also account for the case that no or an incomplete reply is sent, a timeout must be integrated which automatically ends the read process after this time has expired.

```

<?php
$TIMEOUT = 10;
$fd = dio_open( '/dev/ttyS1', O_RDWR | O_NONBLOCK );
dio_tcsetattr( $fd, array(
    'baud'    => 9600,
    'bits'    => 8,
    'stop'    => 1,
    'parity'   => 0,
    'flow_control' => 0,
    'is_canonical' => 0 ));
dio_write( $fd, 'tempertur?' );
$input = '';
$time_start = time();
while($time_start + $TIMEOUT > time())
{
    $input .= dio_read( $fd );
    usleep( 100000 );
}
echo $input;
dio_close( $fd );
?>

```

In contrast to the first script example the port here is opened non-blocking. Whereas in the first example execution of the script in the line `$input .= dio_read($fd, 1);` pauses until a character has been read, in the second example the line `$input .= dio_read($fd);` is skipped if there is no serially received character in the input buffer. Exit from the read loop is here time-based. The duration of the read process is specified in the variable `$TIMEOUT`.

Guaranteed exit from the while loop allows error handling and ensures consistent performance.



We are available to you in person:

Wiesemann & Theis GmbH
 Porschestra. 12
 42279 Wuppertal
 Phone: +49 202/2680-110 (Mon.-Fri. 8 a.m. to 5 p.m.)
 Fax: +49 202/2680-265
info@wut.de

© Wiesemann & Theis GmbH, subject to mistakes and changes: Since we can make mistakes, none of our statements should be applied without verification. Please let us know of any errors or misunderstandings you find so that we can become aware of and eliminate them.

[Data Privacy](#)