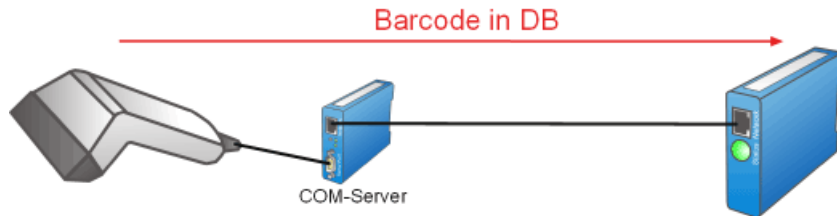


Application for Motherbox:

Store bar codes box-internally in SQLite database

Product overview

Application overview



The example shown here illuminates the following function features of the Motherbox:

- Execution of a PHP script when data are received on a TCP port
- Reading of the received data through the standard input
- Saving the read information in a box-internal SQLite database
- The example shown here illuminates the following function features of the Motherbox:

Objective

A company has a number of loading ramps onto which packages are loaded for delivery in trucks. Before the individual packages are loaded, their bar codes need to be read by an employee with a hand scanner. The scanned codes need to then be saved with time stamp at a central location in a database.

The hand scanner used has an RS232 port and is connected to a W&T Com-Server. When a bar code is read it is passed to the Motherbox via network through the COM-Server, and the Motherbox reads the received data and saves it internally in an SQLite database. The database can then be read out by other programs using the traditional SQL syntax.

The PHP file for implementing this example, **save_data.php**, can be downloaded [here](#). In the compressed file you will also find the file **show_data.php**, which opens, reads and displays the database when opened in the browser.

Configuring the COM-Server

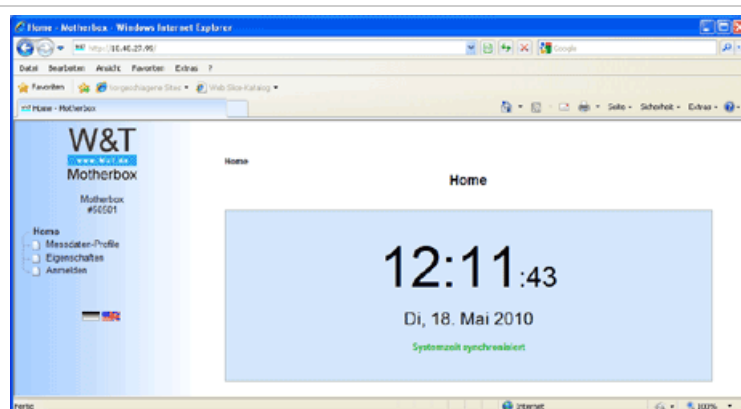
In order for the COM-Server used to be able to pass the data read by the hand scanner via network to the Motherbox, the following configuration steps are necessary:

- The serial configuration (baud rate, handshake, parity etc.) must be matched to the hand scanner when setting up the COM-Server.
- The COM-Server must be set to TCP Client mode, which enables passing of serially received data to a preset IP address and port number (here: Port 8000) on the Motherbox.

Configuring the Motherbox

Save the file **save_data.php** using FTP or SMB sharing in the folder named **programs** on the Motherbox. Then use the following invoking URL from any browser: **show_data.php** in the folder named **websites**. Then the following configuration steps are necessary:

Open the **Homepage** of the Motherbox by entering the **IP address** in the address line of your browser.



Select **Login** in the menu tree and log in from the dialog box with administrator rights (user name: **admin** and your password).

Anmelden

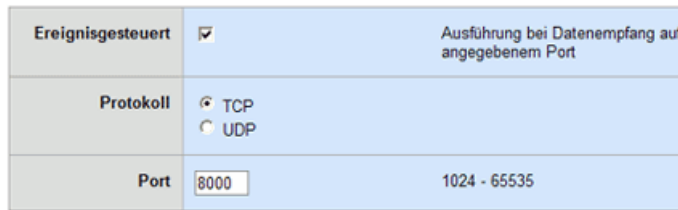
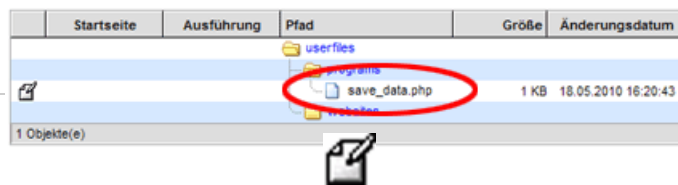
Benutzername	<input type="text" value="admin"/>
Passwort	<input type="password"/>
<input type="button" value="Senden"/>	

Navigate to the page **Home >> Configuration >> User files**. In the file overview click on the folder named

programs. The file `save_data.php` is displayed.

Clicking on the icon **Edit file properties** (left next to `save_data.php`) opens the Edit dialog.

Check the option **Event controlled**. Select **TCP** and set **Port** to 8000. Save your changes by clicking on **Save...** you're done!



Now when data are sent from the hand scanner through the COM-Server to TCP port 8000 on the Motherbox, the script `save_data.php` is run. The received data are read in by the script, a time stamp added and the data are stored in an SQLite database on the Motherbox. In general you should note that running of the PHP interpreter is automatically stopped after two minutes. The programmed action should therefore be capable of being processed in a shorter time.

Source text

In the first line the standard input for reading data is opened.

```
$in = fopen('php://stdin', 'r');
```

The incoming data are read character by character in a loop and the string variable `$input` is appended. The loop is exited when a line break (Line Feed, ASCII table: 10) is detected.

```
$input = '';  
do  
{  
$input .= fread($in, 1);  
}  
while(substr($input, -1, 1) != chr(10));
```

Then the standard input is closed again.

```
fclose($in);
```

To be able to save the data in the database, a database object for write access must be created, here `$db`. If the database does not yet exist, e.g. the first time the script is invoked, the required table `received` with the two columns, `time` for the time stamp and `data` for the received data, are created.

```
$db = new PDO('sqlite:../data.sqlite');  
$db -> exec('CREATE TABLE IF NOT EXISTS received (time INTEGER, data TEXT)');
```

By using an `SQL-INSERT` command a new entry is added in the table `received`. The current time is written to the `time` column in UTC format, and `data` contains the received text. Finally the database is closed again.

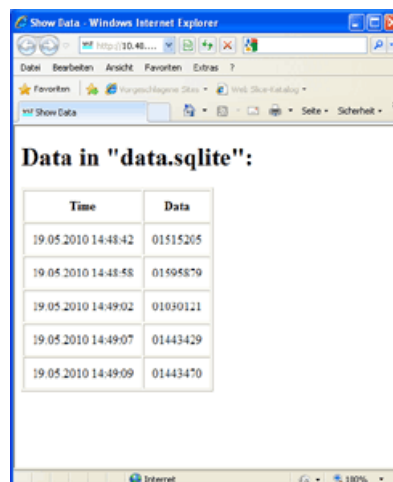
```
$db -> exec('INSERT INTO received (time, data) VALUES (' . time() . ', "' . $input . '")');  
$db = null;
```

The PHP script has now fully run and is exited. When data are received again the script is started again and runs as described above.

If you have previously copied the file `show_data.php` to the sub-folder `websites`, you can open it in your browser using the following URL.

```
http://<IP-Adresse>/websites/show_data.php
```

The page reads the data received so far from the database and displayed in table format.



Summary

Simple server applications can be implemented with the Motherbox with no prior knowledge beyond PHP. The server functionality is already part of the Motherbox. All that is required is that the action you want to run when data are received is programmed.



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](#)