Guide to
Parameterization and integration of microcontroller devices

Flexible data collection with method
Content

1. Introduction..................................................................................................................1

2. System Structure ..........................................................................................................1
2.1. Structure of the hardware / device and firmware ....................................................2
2.2. Usable Characters ......................................................................................................3
2.3. Language / Table .......................................................................................................4
2.4. Displaydesigner .........................................................................................................5
2.4.1. Color Setting for the Displays of the EVO 4.3 / 2.8 and 3.5 ..................................6
2.4.2. Default Setting .....................................................................................................6
2.4.3. Display function buttons on the EVO 4.3 / 2.8 display ........................................7
2.4.4. Upload images for function buttons of EVO 4.3 / 2.8 / 3.5 .................................7
2.4.5. Design examples in the designer .........................................................................8

3. Creation of a setup (program) .....................................................................................9

4. Device key and security ..............................................................................................10
4.1. Device passwords ....................................................................................................10
4.1.1. Communication password ..................................................................................10
4.1.2. Bios Menu Password ..........................................................................................10
4.2. Wireless security ....................................................................................................11

5. Description of the different communication techniques ...........................................12
5.1. Overview of the communications technologies .....................................................12
5.2. Function overview of communication techniques ....................................................13
5.3. Communication via DLL .........................................................................................14
5.3.1. Program library [Dynamic Link Library (DLL)] - General information .............14
5.3.1.1. What is a program library? .............................................................................14
5.3.1.2. Advantages of a program library in DLL or so-form. ..................................14
5.3.1.3. The communication library is available for the following systems: .............14
5.3.2. Program library (DLL) - Integration Passive mode (polling) .............................15
5.3.3. Program library (DLL) - Integration active mode (active connection) ..............16
5.3.4. Encrypting the data when using the DFCom.dll ...............................................17
5.3.4.1. Create and save a communication key for the device ................................18
5.3.4.2. Save the communication key in the StudioIV ..............................................19
5.3.4.3. Transfer the communication key for DFComDLL ........................................19
5.3.4.4. Clear the communication key .....................................................................20
5.4. http Level 0 ............................................................................................................21
5.4.1. Preconditions ......................................................................................................21
5.4.2. Sending of data records via HTTP .....................................................................22
5.4.3. HTTP response and optional parameters .........................................................22
5.4.4. Advantages and disadvantages with HTTP .......................................................22
5.4.5. Encryption of the data fields when sending via HTTP .......................................23
5.5. http Level 0 with service connection .....................................................................28
5.6. http Level 1 ............................................................................................................29
5.6.1. Requirements ......................................................................................................29
5.6.2. Changes between Level 0 and Level 1 concerning Request and Response ........29
5.6.2.1. The following changes have been made to level 0 in the request: ...............29
5.6.2.2. The following changes have been made to level 0 in the response: .............29
5.6.3. Request ................................................................................................................30
5.6.3.1. Method: GET .................................................................................................30
5.6.4. Response ...............................................................................................................31
5.6.5. Encryption ...........................................................................................................33
5.6.5.1. Encryption detection ...................................................................................33
5.7. https Communication ..............................................................................................34
5.7.1. Requirements ......................................................................................................34
5.7.2. Elements of the https infrastructure .................................................................34
5.7.3. Use of encryption / certificates .........................................................................34
5.8. Talk ..........................................................................................................................35
5.8.1. Advantages and disadvantages with Datafox Talk ............................................35
5.8.2. When do I use Talk? ..........................................................................................36
5.8.3. Overview of the Function Modules Talk ............................................................36
5.8.4. Establishment of Talk ........................................................................................37
## Changes within this document

<table>
<thead>
<tr>
<th>Date</th>
<th>Chapter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>07.08.2013</td>
<td>Alle</td>
<td>Neuauflage der Dokumentation</td>
</tr>
<tr>
<td>08.09.2013</td>
<td>Alle</td>
<td>Ausdruckkorrektur</td>
</tr>
<tr>
<td>14.05.2014</td>
<td>4.3</td>
<td>Hinweis Unterschiede Active-Mode -&gt; passive Mode</td>
</tr>
<tr>
<td></td>
<td>4.4</td>
<td>Onlinefunktion ZK über HTTP</td>
</tr>
<tr>
<td>11.05.2015</td>
<td>4.3.4 und 4.4.5</td>
<td>Verschlüsselung ergänzt</td>
</tr>
<tr>
<td>09.08.2016</td>
<td>4.2</td>
<td>Übersicht / online offline Funktionen der ZK ergänzt</td>
</tr>
<tr>
<td>07.11.2016</td>
<td>2.2</td>
<td>Language</td>
</tr>
<tr>
<td></td>
<td>2.3</td>
<td>Characters / Zeichensätze</td>
</tr>
<tr>
<td></td>
<td>4.3.4.1-4</td>
<td>Verschlüsselungsverfahren im einzelnen</td>
</tr>
<tr>
<td>22.12.2016</td>
<td>4.2</td>
<td>Display Designer</td>
</tr>
<tr>
<td>28.12.2017</td>
<td>5.6</td>
<td>http Level 1</td>
</tr>
<tr>
<td>28.12.2017</td>
<td>5.7</td>
<td>https</td>
</tr>
<tr>
<td>23.01.2018</td>
<td>4.2</td>
<td>wireless security</td>
</tr>
<tr>
<td>17.12.2019</td>
<td>5.6</td>
<td>More options for the http Level 1</td>
</tr>
</tbody>
</table>
1. Introduction

This document is a guide to training in the topics:

a) Configuration and

b.) integration of communication

The guideline applies to all microcontroller devices MasterIV series and EVO series, as well as integrated embedded modules in the industrial PC.

(Note: The built-in Industrial PC embedded modules can also be operated in HID mode.)

This document will help you to estimate the integration effort.

There you will be given links and notes, where you will find information on each topic.

The Parameterization of the devices is done via Datafox studio. This program allows you to create the desired acquisition and testing functions quickly, easily and without any programming knowledge or, as we say, to parameterize. The free of charge/licence free program can be found on our website or on the product DVD.

When integrating communication, we give you an overview of the possible communication channels with Datafox terminals. The individual channels are shown and described with their advantages and disadvantages. On the DVD, and our website you will find sample programs for the integration of the DLL in the current programming languages for your use.

2. System Structure

This overview shows the basic structure of the complete data acquisition system and its areas:

The following components are explained in the following chapters:

Structure of the hardware / device and firmware (Link)

Creating a setup (Link)

Communication techniques with Datafox devices (Link)
2.1. **Structure of the hardware / device and firmware**

This graphic shows the structure of data acquisition devices in context. Underlying hardware is fitted out as desired. Then there is the firmware, which is the operating system. The program, which we call setup file is executed by the firmware.

This connection is important because you do not have access to the individual areas of the system with every communication technology.
2.2. Usable Characters

For your information, we have put together an explanation of the characters in the standard scope of delivery. This information can also be found in the manuals. The Datafox devices support a part of the character encoding Latin-1 (ISO-8859-1) for the character output on the display and the data.

ISO 8859-1, more accurate ISO/IEC 8859-1, also known as Latin-1, is a standard for information technology on eight-bit character encoding, updated by the ISO last 1998, and the first part of the standards family ISO/IEC 8859.

<table>
<thead>
<tr>
<th>Code</th>
<th>...0</th>
<th>...1</th>
<th>...2</th>
<th>...3</th>
<th>...4</th>
<th>...5</th>
<th>...6</th>
<th>...7</th>
<th>...8</th>
<th>...9</th>
<th>...A</th>
<th>...B</th>
<th>...C</th>
<th>...D</th>
<th>...E</th>
<th>...F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2...</td>
<td>SP</td>
<td>!</td>
<td>”</td>
<td>#</td>
<td>$</td>
<td>%</td>
<td>&amp;</td>
<td>‘</td>
<td>(</td>
<td>)</td>
<td>+</td>
<td>,</td>
<td>-</td>
<td>.</td>
<td>/</td>
<td></td>
</tr>
<tr>
<td>3...</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>:</td>
<td>;</td>
<td>&lt;</td>
<td>=</td>
<td>&gt;</td>
<td>?</td>
</tr>
<tr>
<td>4...</td>
<td>@</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
<td>G</td>
<td>H</td>
<td>I</td>
<td>J</td>
<td>K</td>
<td>L</td>
<td>M</td>
<td>N</td>
<td>O</td>
</tr>
<tr>
<td>5...</td>
<td>P</td>
<td>Q</td>
<td>R</td>
<td>S</td>
<td>T</td>
<td>U</td>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>[</td>
<td>\</td>
<td>]</td>
<td>^</td>
<td>_</td>
</tr>
<tr>
<td>6...</td>
<td>a</td>
<td>b</td>
<td>c</td>
<td>d</td>
<td>e</td>
<td>f</td>
<td>g</td>
<td>h</td>
<td>i</td>
<td>j</td>
<td>k</td>
<td>l</td>
<td>m</td>
<td>n</td>
<td>o</td>
<td></td>
</tr>
<tr>
<td>7...</td>
<td>p</td>
<td>q</td>
<td>r</td>
<td>s</td>
<td>t</td>
<td>u</td>
<td>v</td>
<td>w</td>
<td>x</td>
<td>y</td>
<td>z</td>
<td>{</td>
<td></td>
<td></td>
<td>~</td>
<td></td>
</tr>
<tr>
<td>8...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A...</td>
<td>NBSP</td>
<td>i</td>
<td>ç</td>
<td>£</td>
<td>Ø</td>
<td>¥</td>
<td>!</td>
<td>&quot;</td>
<td>®</td>
<td>²</td>
<td>à</td>
<td>ā</td>
<td>À</td>
<td>Ä</td>
<td>Å</td>
<td>Æ</td>
</tr>
<tr>
<td>B...</td>
<td>°</td>
<td>±</td>
<td>²</td>
<td>³</td>
<td>µ</td>
<td>¶</td>
<td>·</td>
<td>,</td>
<td>¹</td>
<td>½</td>
<td>¾</td>
<td>à</td>
<td>á</td>
<td>à</td>
<td>à</td>
<td>à</td>
</tr>
<tr>
<td>C...</td>
<td>À</td>
<td>Ä</td>
<td>Å</td>
<td>A</td>
<td>À</td>
<td>Æ</td>
<td>Ç</td>
<td>É</td>
<td>É</td>
<td>È</td>
<td>Ë</td>
<td>i</td>
<td>i</td>
<td>î</td>
<td>i</td>
<td>î</td>
</tr>
<tr>
<td>D...</td>
<td>Ð</td>
<td>Ñ</td>
<td>Ô</td>
<td>Õ</td>
<td>Ø</td>
<td>×</td>
<td>Ø</td>
<td>Ù</td>
<td>Ú</td>
<td>Ù</td>
<td>Ú</td>
<td>Ù</td>
<td>Ù</td>
<td>Ù</td>
<td>Ú</td>
<td>Ù</td>
</tr>
<tr>
<td>E...</td>
<td>ã</td>
<td>ã</td>
<td>ã</td>
<td>ã</td>
<td>ã</td>
<td>ã</td>
<td>ç</td>
<td>é</td>
<td>é</td>
<td>é</td>
<td>é</td>
<td>i</td>
<td>i</td>
<td>i</td>
<td>i</td>
<td>î</td>
</tr>
<tr>
<td>F...</td>
<td>ô</td>
<td>õ</td>
<td>õ</td>
<td>õ</td>
<td>õ</td>
<td>õ</td>
<td>ò</td>
<td>û</td>
<td>û</td>
<td>û</td>
<td>û</td>
<td>ù</td>
<td>ù</td>
<td>ê</td>
<td>ë</td>
<td>ë</td>
</tr>
</tbody>
</table>

SP = space, NBSP = hard coded space, SHY = conditional separator

Control characters according to ISO standard can not be used
- characters can not be used
- Usable characters of Firmware-Version 04.01.xx.xx to Firmware-Version 04.03.02.xx
- Character extension Firmware-Version 04.03.03.xx (only Hardware V4)
- character extension Firmware-Version 04.03.04 when using character table Latin-1 (only Hardware V4)
2.3. Language / Table
In order to ensure language compatibility, it is possible to edit the texts and messages displayed by the firmware.
Open the editing dialog via the menu "Configuration – Language file for device (*.dfl) – Edit file for language table".

Open a device file archive (firmware)*.dfz. The default texts of the firmware with a description and the corresponding message are displayed.

Open or create a new language file for the firmware with the extension *.dfl. If you have created a new file, the right column of the list is empty.

Work within the lists with single mouse clicks only. NO double-clicks! Select a line from the list with a single click.

With another single click on the column User (Description/…) or User (Message/…) the cursor is displayed in this field.

Now you can enter or edit the text. When you finish the entry, the description form the column Default (Description/…) is taken over and you can edit it as well. You can find prepared .dfl-files on the Datafox DVD.

Note:
Cyrillic and Chinese characters can not be displayed.

You find different language file.dfl on our product DVD:
- Englisch, Niederländisch, Französisch
- Default setting is German. German is always included in the firmware file.

Datafox DVD/MasterIV_EVO_TimeboyIV/Datafox Geräte/Datafox Software MasterIV-04.03.07/Datafox Studio-IV_und_Firmware/Sprachdateien der Firmware
Beispiel: Datafox Software-MasterIV V04.03.07.zip
2.4. Displaydesigner

scope of application:

For the devices AE-MasterIV V4, PZE-MasterIV V4 and PZE-MasterIV Basic V4 the Designer is only usable with the optional color display.

With the Display-Designer, Datafox offers the possibility for partners and users to customize the display according to your requirements. But due to the necessary operating sequences, this cannot be a completely free design, but things like headlines, menu structures and footers have to be guaranteed. The aim of the display designer is to enable the feasible settings with minimal effort.

We are looking forward to many users and recommend: Create an individual Display-Design for your Company:

Example picture for EVO 4.3

Example picture for EVO 2.8 / 3.5

Example picture for PZE-/ AE- Master V4 with color display

To create an individual display design for your device, you need at least the DatafoxStudioIV 04.03.09.05.

The display designer can be opened via the Configuration menu or directly from the setup edit mask.
2.4.1. Color Setting for the Displays of the EVO 4.3 / 2.8 and 3.5

Example Picture:

2.4.2. Default Setting

The device is delivered in the default „PZE“-design.

This design is also set as default when you first create a new theme in Display Designer.

Function Key’s are not displayed in the default setting.
2.4.3. Display function buttons on the EVO 4.3 / 2.8 display

By showing the function buttons from the setup, the number of buttons displayed in the display can be adjusted.

Example:

2.4.4. Upload images for function buttons of EVO 4.3 / 2.8 / 3.5

Under this menu item "Key settings" you can import the image file for each function key.

Sample picture for the key figures:
2.4.5. Design examples in the designer

With the installation of the DatafoxStudioIV you get several design examples for the devices. Click on the "Design Examples" button to open them.

Datafox gradually extends the examples.
If you have any suggestions or wishes, please let us know.
3. Creation of a setup (program)

The creation of the setup is done with the free of charge "DatafoxStudioIV" tool. The structure of the tables for master data and bookings as well as the operating procedures, are freely definable. **There are no programming skills required.**

1. Insert the data structure of the tables individually for bookings

   **Record table: Records**
   ```plaintext```
<table>
<thead>
<tr>
<th>record code</th>
<th>ID_Person</th>
<th>Name</th>
<th>date and time</th>
<th>reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>656556</td>
<td>M. Müller</td>
<td>21.02.2013 12:31:15</td>
<td>0</td>
</tr>
<tr>
<td>K</td>
<td>656556</td>
<td>F. Mustermann</td>
<td>21.02.2013 12:32:45</td>
<td>0</td>
</tr>
</tbody>
</table>
   ```plaintext```

2. Define the structure of the tables for master data.

   **Access Table: Records access**
   ```plaintext```
<table>
<thead>
<tr>
<th>ID</th>
<th>date and time</th>
<th>Master ID</th>
<th>Reader ID</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0566236654</td>
<td>21.02.2013 12:31:15</td>
<td>1</td>
<td>010</td>
<td>20</td>
</tr>
<tr>
<td>21.02.2013 12:32:45</td>
<td>1</td>
<td>010</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>1566959651</td>
<td>21.02.2013 14:12:05</td>
<td>1</td>
<td>020</td>
<td>20</td>
</tr>
<tr>
<td>21.02.2013 16:55:14</td>
<td>1</td>
<td>020</td>
<td>42</td>
<td></td>
</tr>
</tbody>
</table>
   ```plaintext```

3. Specify the operation and signal processing.

   - menus
   - show text
   - show list
   - submenus
   - RFID-methods
   - Type of inputs

4. Create the Access Control

   - Detection order
   - Status
   - online / offline

5. Transfer the setup to the terminal.

6. Collect data
4. Device key and security

There are different techniques for the Datafox devices to protect the device from unqualified access.

4.1. Device passwords

Device passwords are used to prevent devices from being unintentionally / accidentally or intentionally read or manipulated in the settings for communication or data.

These settings are only intended to ensure the operational safety of the devices and should be part of the standard. These settings do not have the encryption passwords. Please see the chapter on encryption via http and DLL.

4.1.1. Communication password

Our software "DatafoxStudioIV" is freely available on the homepage. The devices are configured by our partners. To prevent misuse or manipulation by users, a communication password can be stored in the device. Only those who know this can change the configuration of the device. The Password is transferred to the device with the configuration (setup file).

4.1.2. Bios Menu Password

All display devices have a bios menu. Settings can be made as followed:
- IP - address
- The type of Communication (GPRS, USB, TCP/IP) etc.
- Display brightness, volume etc.

To prevent access to the bios menu, a password can be entered here. This password is then transferred to the device with the configuration (setup).
4.2. Wireless security

This overview shows you which WLAN methods are supported. **Not supported** is WPA (Predecessor of WPA2).

Not supported is multiple-input multiple-output (MIMO)

5 GHz connections are **not supported** and no mixed operation 2.4 GHz / 5 GHz.

Authentication via WPA2 Enterprise according to IEEE 802.1x is **not supported**.

---

**Attention:**

We cannot test every available Access-Point on the market. Therefore, it is not possible for us to guarantee a connection to any AP.

**Attention:**

multiple-input multiple-output (MIMO) is not supported. If you change the AP from b/g/n to b/g, only SISO is automatically used. [https://en.wikipedia.org/wiki/Single-input_single-output_system](https://en.wikipedia.org/wiki/Single-input_single-output_system)

When setting the encryption AES or WEP, only one type is used at a time. The setting AES+WEP means for some access points that AES encryption is performed first and then additionally encrypted with WEP.

In this case, only set AES.
5. Description of the different communication techniques

5.1. Overview of the communications technologies

This overview is valid for all microcontroller devices of MasterIV series and EVO series, as well as for the embedded modules integrated into the Industrial PC. (Note: The built-in Embedded modules inside the Industrial PCs can also be operated in HID mode.)

In order to enable the respective communication, the main communication must be set in the BIOS of the device. How to enter the BIOS menu, please see the respective manuals in the chapter "Display structure and Bios".

### DLL integration
- **Passive-mode** = Polling
  (free of charge)

- **Active-mode**
  Permanent Active connection to the terminal
  (free of charge)

### HTTP
- **Level 0**
  Automatic sending of data to a Web Server
  (free of charge)

- **Level 0 with Service mode**
  Sending the data to a Web server and establishment of a service connection (free of charge)

- **Level 1**
  Automatic sending of data to a Web Server and transfer list to the device
  (free of charge)

- **Datafox-Talk**
  Data exchange via file storage or database with a service
  (License Required)
## 5.2. Function overview of communication techniques

<table>
<thead>
<tr>
<th>access to:</th>
<th>description</th>
<th>DLL integration Polling</th>
<th>DLL active-mode</th>
<th>http Level 0</th>
<th>http Level 1</th>
<th>http Level 0 with Service-mode</th>
<th>Talk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data sets</td>
<td>Generated data from the data acquisition</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Transfer master data</td>
<td>- staff lists  - Article Master  - orders</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>System</td>
<td>- Firmware  - Language  - Colour/Display</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>System</td>
<td>- Setup</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>GV global variables</td>
<td>8 x free use</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>System-variables</td>
<td>GPRS COM I/O</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Message to The device</td>
<td>Direct display of text on the display</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Online Mode</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Online Mode access</td>
<td>All access authorizations are decided directly from the server</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Access control alternating online offline</td>
<td>In case the server is offline: automatic change to Offline-Mode</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Access control offline</td>
<td>The device uses its own access control lists</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Access control with assisted online</td>
<td>The device checks itself in the list and sends the result to the server, the server then checks</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Timeliness of data ca. in seconds</td>
<td>How fast is the generated data available</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Message "data sets exist" <1s
<1s HTTP via Lan, 1-2s via GPRS
<1s HTTP via Lan, 1-2s via GPRS
< 1s HTTP via Lan 1-2s via GPRS
300s or more depending on the setting; 300s rec.

Only possible after the request for a service connection „Active-Mode“
5.3. Communication via DLL

5.3.1. Program library [Dynamic Link Library (DLL)] - General information

5.3.1.1. What is a program library?
"A program library referred in programming as a set of subroutines that provides solutions to thematically related problems. Libraries, unlike programs, are not stand-alone units, but auxiliary modules that are requested or called by programs."

Wikipedia about program library

In order to address the functions (subroutines) of the program library, it must be integrated into your software solution. Depending on the development environment a certain procedure is required. In principle, all have one thing in common: the required functions have to be announced to your software solution (they must be declared).

5.3.1.2. Advantages of a program library in DLL or so-form.

Here specifically for the Datafox DLL.

- The integration of a DLL is easier and faster than the direct integration of a protocol.
- Can be used independently of the programming language.
- Single Programming Interface (API) to the different Datafox devices.
- The DLL shows defined error messages if operations can not be performed correctly.
- The DLL automatically writes log files for debugging.
- Updateable without rebuilding your software solution. Downwards compatible.

5.3.1.3. The communication library is available for the following systems:

- As DLL for Windows 32bit, DFComDLL.dll; 64bit, DFCom_x64.dll
- As Shared Library or Static Library for Linux 32/64bit libDFCom.so (Makefile)
5.3.2. Program library (DLL) - Integration Passive mode (polling)

In the passive mode the communication link is established starting from the program library to the devices. For this you need a single function to establish connections and another for disconnection.

Functional principle when transferring the bookings:
The application queries the device via the DLL regularly to collect the data.

The following connection types are supported by the passive mode:
- RS232 (via converter also RS485)
- USB (via virtual COM-Port)
- Modem (analog / GSM)
- TCP / IP (LAN / WAN / WLAN)

An exemplary query of a device serial number in an device connected via TCP / IP using the C programming language:

```c
int err, serial;
DFComOpenIV( 5, 0, 3, "192.168.0.3", 8000, 3000 );
DFCGetSeriennummer( 5, 254, &err, &serial );
DFComClose( 5 );
```

advantages:
- All connection types and device types are supported.
- All functions of the program library are fully available.

disadvantages:
- If you want to receive the generated data sets immediately after creation, a constant communication with the device is necessary (polling). In TCP / IP networks, this, depending on the number of units, may lead to an undesirable loss of bandwidth.
- Not recommended for mobile communications as it may cause high costs.
5.3.3. Program library (DLL) - Integration active mode (active connection)

In Activ mode, the communication link from the devices to the program library is made. Prior to this, the devices must be informed at the time of installation where they are to be connected. In the program library, you need a single function to start the active mode and another to end the active mode.

The "DFCStartActiveConnection" function in this case replaces the function "DFComOpenIV" in passive mode.

After activation of active mode in the program library this waits for incoming connections and is then making them available for further processing in your application.

Functional principle when transferring the bookings:

The devices sign in to the DLL. This writes a list of registered devices. If a device has a booking, this sends a trigger to the DLL. The application responds to the trigger and fetches the booking. The necessary connection data is available in the registration list. The collection of the booking is made with the same functions as with polling. As a result, polling and active connection upon collection of the data differ only slightly.

The following connection types are supported by the active mode:

- TCP / IP (LAN, WAN, WLAN, GPRS)

advantages:

- The device log on independently and also report existing records.
- The application does not require a device list, since the devices sign in automatically into the DLL and the DLL provides a list of active devices.
- All functions of the program library are fully available.

disadvantages:

- Here, since the multi-master principle is necessary, only connections by connection type TCP / IP are supported.

More information and documentation can be found here:
5.3.4. Encrypting the data when using the DFCom.dll

When using the Datafox communication DLL all data coming from the device or sent to the device may be transferred with an AES 128-bit encryption.

Thus, there are only 3 types of communication:
1. Unencrypted communication
2. Encrypt with Datafox-Key
3. Encrypt with user-Key

When integrating the Datafox DFCom.dll from the application side, only a user communication key is handed over to the .dll. The effort of integration of encryption is thus very low.

Overview of the encryption, schematic illustration.

A detailed description of how the handover of the key takes place, you will find:
- for DatafoxStudioIV in the manual in the chapter "encrypt communication with MasterIV devices"
- and for the DLL in the DLL documentation
5.3.4.1. Create and save a communication key for the device
In the menu „Configuration“ -> “System variables active mode” open the configuration file to edit. For example: „active.ini“.

Click on the line “Key” to open a new window and to create a key.

Select the type of communication.

If you want the communication to be encrypted with your own stored password, enter a password and click on the button "Create value from password".

A communication key is now created. Finish the entry with "OK". After a key has been created and the active.ini file has been transferred, communication with the device is only allowed if the password is entered.
5.3.4.2. Save the communication key in the StudioIV
If a device is using a communication key, then the DatafoxStudioIV needs the same key. Otherwise no communication with the device would be possible.
In the menu „Communication -> Settings“ you may edit the key for the Communication.

The password is used for all types of communication.

Enter your password here.

The plaintext input is only possible at the first input. If you reopen the window you won’t see the plaintext.

5.3.4.3. Transfer the communication key for DFComDLL
The key for the communication transferred to the DLL is called “DFCSetCommunicatioPassword”. The key has to be in plaintext (123456) and not the created key of the DatafoxStudioIV.

More information can be found in the documentation of the DFComDLL.
5.3.4.4. Clear the communication key

If created a communication key and transferred to the device then clear this key as follows:

Click on “KEY” to edit.

Switch to unencrypted communication.

Click on: “Value empty”. Then click on: “Create value from Password”. The created Value from the empty Password is necessary to clear the old key in the device.

Save the file and transfer to the device.

After this you can clear the created key from the .ini file.
5.4. **http Level 0**

5.4.1. **Preconditions**

Requirements for transmission of the data via HTTP:

**Hardware:**
- Device with TCP / IP or cellular function

**Software:**
- Server must accept an HTTP request and give an active response

**Functional principle when transferring the bookings:**

The devices send the data automatically and directly after their emergence to the Web server. There, the bookings are accepted by an executable script and forwarded to the main application or in a database.

**HTTP over TCP/IP**

On any device that has a TCP / IP interface, you can enable the HTTP of the device in the BIOS menu under “Communication”. For this purpose the entry "http" to "YES" is amended. Required for sending the data with HTTP over LAN, is the correct settings of the parameters in the .ini file and the communications must be set to TCP / IP.

**HTTP over GPRS mobile network**

For the use of data transfer via GPRS the device requires a SIM card, preferably with a M2M data plan. The setting of the connection parameters is carried out via the Datafox studio and is stored in the GPRS-ini. Please draw attention to the roaming settings, otherwise it may lead to high costs.
5.4.2. Sending of data records via HTTP

The Datafox device is able to send booking data promptly via GPRS or TCP / IP to a web server.

Example of a send string:

<table>
<thead>
<tr>
<th>Plaintext request</th>
</tr>
</thead>
<tbody>
<tr>
<td>getdatagv.php?table=BB&amp;TYP=Manu&amp;bLOG=Log&amp;bDAT=2011-05-24_08%3A30%3A12&amp;bPER=Per&amp;checksum=2120</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Plaintext reply</th>
</tr>
</thead>
<tbody>
<tr>
<td>status=ok&amp;checksum=2120(Prüfsumme) (always enter at the end: \r\n (carriage return line feed))</td>
</tr>
</tbody>
</table>

5.4.3. HTTP response and optional parameters

In the response String you can react to the request. Expected is always one answer "status=ok&checksum=2120". With other possible parameters, the device may perform the following actions:
- Set time
- output tones
- start Input string (Quasi press a virtual button)
- Assign / pass values to global variables
- Show messages on the display

5.4.4. Advantages and disadvantages with HTTP

advantages:
- Fast transfer of the data
- Direct online capability (eg instant display of a response on the display)
- Online access control (all permissions are executed by your software)
  Easy integration as php or other active scripts can be adapted very quickly and are very common.
- When using the online mode, you can dispense with the use of master data in the device.
- When booking, the server can sent direct messages, such as "Good day John Doe, you have successfully logged in", to the display.

disadvantages:
- Master Data can not be transmitted to the terminal via HTTP. It has to be switched to the HTTP service connection. See next chapter
- There is no access to system variables possible.

More information can be found in the manual DatafoxStudioIV in the chapter "Configuration" -> "System variables HTTP / GPRS"

Latest manuals are available on the homepage:


On our website you will find a test environment with detailed instructions. You can send directly in Live mode data to the server and see them. This is also very good for product demonstrations.
Test environment:

http://www.datafox.de/online-testumgebung.html

Instructions and test setups:

5.4.5. Encryption of the data fields when sending via HTTP

If data records are sent via HTTP, the field contents can be transmitted in encrypted form. The data fields of the data set are then encrypted using a RC4 encryption. The so encrypted characters are transferred as field contents in hexadecimal.

Overview of encryption with HTTP; schematic representation:

A detailed description of how the handover of the key takes place, can be found:
- in the manual of DatafoxStudioIV Chapter 5.4.5. "Encryption of the data fields when sending via HTTP"
- and the file "dfanalyser.php" can be found on the Datafox DVD or in the download - Software for Windows

**Note:**
If you use encryption for several customers, you have to pass a customer ID into the plain text.
This allows you to use different keys per customer.

<table>
<thead>
<tr>
<th>Port</th>
<th>80</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTPSEND</td>
<td>GET <code>/getdata.php?customer=10248</code></td>
</tr>
<tr>
<td>ALIVE</td>
<td>60</td>
</tr>
</tbody>
</table>
Activating Encryption via DatafoxStudioIV

Open the configuration file (e.g. GPRS.ini) for editing via the menu entry Configuration "GPRS / HTTP - Configuration".

By clicking on the line “KEY”, the window for creating the key opens.

Enter your password here.

By clicking the button "Create value from password", a key for transfer is generated.

Click "OK" to take the key over.
Subsequently, you can save the settings and transfer them to the Datafox device.
Disable encryption

To deactivate the key which has been transferred to the device, it is necessary to create an empty password field with the button "Clear value" and to transfer this empty key to the device.

Click on „Value empty“

Click on "Create value from Password".

and then on „OK“

Save this file with the new key.
The records are then sent unencrypted. After this you can clear the key from the file.

**Deactivating Encryption**
In order to deactivate the key transferred to the device, you must create an empty password field by clicking the button "Empty value" and transfer the empty key to the device. The data records are then sent unencrypted.

**Illustration of the GET Request**
in plaintext (unencrypted) and encrypted:

**Plaintext request**
```
```

**Plaintext response**
```
status=ok&checksum=2120
```

**Encrypted request**
```
getdatagv.php?dfcb=1000&table=e977&bTYP=14dce883&bLOG=4d7876&…&checksum=c01de865&dfce=019c1bd2
```

**Encrypted response**
```
dfcb=1000&status=2b97&checksum=1726950d&…&setup_2=a449fd9c&setup_blue=a9375c8d0672&dfce=b99239f3
```

**Detection of an Encryption**
In order to detect whether the data fields are sent encrypted, the beginning of the encryption is marked with 'dfcb' (Datafox Crypt Begin) and the end is marked with 'dfce' (Datafox Crypt End). 'dfcb' is the first field of the GET request and 'dfce' the last field.

The value of the field 'dfcb' is transferred in plaintext and is the 'public key'. It is a random number between 1000 and 9999. Combined with the user password, the value must be used for encryption and decryption.

Encryption of data thus is achieved by "private key + public key" as password key.
In the response, the field 'dfcb' must be returned exactly. This ensures that decryption has been successful and the response matches the request.

The value of the field 'dfce' is the same as 'dfcb' but it is transferred encrypted. During encryption it can be ensured that the key used is correct. The value of 'dfce' must equal the value of 'dfcb' after encryption.

If problems occur during encryption, the response 'dfc=error' must be sent. Additionally, information must be entered in the fields 'dfcb' and 'dfce'.

<table>
<thead>
<tr>
<th>The following errors must be considered by the evaluating script:</th>
</tr>
</thead>
<tbody>
<tr>
<td>'dfcb' is not a number or is outside the value range of 1000 to 9999</td>
</tr>
<tr>
<td>• Response: dfc=error&amp;dfcb=range&amp;dfce=unknown/missing</td>
</tr>
<tr>
<td>o Range means that the value is outside the value range.</td>
</tr>
<tr>
<td>o Unknown means not determined but available.</td>
</tr>
<tr>
<td>o Missing means not specified in the request.</td>
</tr>
<tr>
<td>'dfcb' without closing 'dfce'</td>
</tr>
<tr>
<td>• Response: dfc=error&amp;dfcb=1000&amp;dfce=missing</td>
</tr>
<tr>
<td>'dfce' is not a number or is outside the value range of 1000 to 9999</td>
</tr>
<tr>
<td>• Response: dfc=error&amp;dfcb=1000&amp;dfce=range</td>
</tr>
<tr>
<td>'dfce' without incipient 'dfcb'</td>
</tr>
<tr>
<td>• Response: dfc=error&amp;dfcb=missing&amp;dfce=unknown</td>
</tr>
<tr>
<td>,dfce' does not equal ,dfcb</td>
</tr>
<tr>
<td>• Response: dfc=error&amp;dfcb=1000&amp;dfce=different</td>
</tr>
<tr>
<td>o Different means that 'dfce' is different from 'dfcb' after decryption.</td>
</tr>
</tbody>
</table>

Response of the Web Server
The field content of the request is deciphered successively using the RC4 stream cipher. The field content of the response is regarded as part of the overall data stream and is ciphered again with the current status of the stream cipher after decryption. Only exception is the first field value of 'dfcb'. It is sent back exactly like in the request.
To the response, 'dfce' must be added as last encrypted field. The value of 'dfce' must equal the value of 'dfcb'.

Activation via Script
The script must use the known "plaintext" password, not the encrypted one generated at the Studio. See the example-php at the product DVD: "dfanalyser.php".
For further information see the DLL description on the product DVD under: DVD\MasterIV-Serie\Datafox Geräte\Datafox Software MasterIV-04.02.04_Release\Kommunikationsmodul DFComDLL 04.02.04 (Windows, Linux)
5.5. **http Level 0 with service connection**

This type of communication combines the advantages of fast data transfer via HTTP with the full access to the data acquisition system through the integration of the DLL "Active mode".

![Diagram of communication between HTTP and Active Mode]

**Concept of switching between HTTP and Active Mode:**

The data transmission will initially run through HTTP. Shall now master data such as personal lists, orders or article master are transmitted to the terminal, a maintenance connection is requested from the server. This is done through the response of the web server.

*example: \"status=ok&checksum=2120&service=1\"*

The terminal now interrupts the transmission of data via HTTP and builds automatically a connection to the DLL (see Active mode). Now Master Data, Setup, etc. can be transferred.

Since the requirement of a service connection is possible only in connection with the response to transmitted data, you have the option to cyclically send an "Alive" data set. This makes sense in any case to monitor from the application which devices are online..

After closing the service connection, the device sends the data captured immediately via HTTP. The possibly incurred in the meantime data are kept in the devices memory.
5.6. **http Level 1**

5.6.1. **Requirements**

Required for data transfer via http Level 1:

**Hardware V4:**
- Device with TCP/IP (LAN / WLAN) or mobile radio
- min. Firmware 04.03.10. XX

**Software:**
- Server must accept an http request and give an active response
- Server must provide master data such as personnel lists or order lists for downloading

 Basically, the same schematic structure applies as for http Level 0.


5.6.2. **Changes between Level 0 and Level 1 concerning Request and Response**

5.6.2.1. **The following changes have been made to level 0 in the request:**

- First field sent is ‘df_api=1’ always. This field is unencrypted even for active encryption requests.
- The table parameter is now ‘df_table’ after naming scheme.
- All data fields are now starting after the naming scheme with ‘df_col_’.
- In fields of the type ”Date and Time” the date and time are separated using a ‘T’ rather than ‘.’.
- The checksum parameter is omitted completely.
- With active encryption parameters ‘dfcb’ and ‘dfce’ are sent as ‘df_cb’ and ‘df_ce’ reflecting the new naming scheme.

5.6.2.2. **The following changes have been made to level 0 in the response:**

- The parameters ‘status’ and ‘checksum’ are completely eliminated.
- As first field ‘df_api=1’ has to be sent always. This field must be transmitted unencrypted even if encryption is active.
- All existing parameters are prefixed with ‘df_’ and – if necessary – adjusted to the currently common terms.
- With an active encryption parameters ‘dfc’, ‘dfcb’ and ‘dfce’ change to ‘df_c’, ‘df_cb’ and ‘df_ce’ following the new naming scheme.
- At the end of the server’s reply, no Carriage Return/Linefeed has to be sent. Should you send Carriage Return/Linefeed at the end of the message, these characters are removed silently. If you need CR/LF inside a message, they have to be encoded as following:
  1. All characters other than letters, digits, - (Minus), . (Dot), _ (Underscore) and ~ (Tilde) have to be encoded as %xx, where xx is the hexadecimal representation of the ASCII code (See RFC 3986, sections 2.3 and 2.4)
5.6.3. Request

Request from the client to the server.

5.6.3.1. Method: GET

For communication via http level 1, Datafox has developed a specific context that applies only to Datafox devices. The GET method is used for http communication. The context now offers extensive possibilities for exchanging data quickly and conveniently with Datafox devices.

Function overview via GET:

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>df_table</td>
<td>Name of data set description</td>
</tr>
<tr>
<td>df_record_state</td>
<td>online / offline state of the records</td>
</tr>
<tr>
<td>df_col_{Feldname}</td>
<td>Name of the data field and value. Entsprechend der Gerätekonfiguration „Setup“.</td>
</tr>
</tbody>
</table>
### 5.6.4. Response

The server’s response to the client.

Each dataset from a Datafox device must be acknowledged by the server.

The confirmation is carried out with:

```
df_api=1 und HTTP–Result „200 OK“
```

#### 5.6.4.1 Optional parameter specifications for the response

<table>
<thead>
<tr>
<th>Instruction Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>df_time=2016-11-17T12:13:14</td>
<td>Set the date and time on the device.</td>
</tr>
<tr>
<td>df_beep=1 (1-11)</td>
<td>OK signal / generate beep on the device</td>
</tr>
<tr>
<td>df_service=1,www.datafox.de,10047</td>
<td>Connect to the DLL. Also possible with the DatafoxStudioIV. Specification of IP/URL and port possible.</td>
</tr>
<tr>
<td>df_var=setup.1,wert</td>
<td>Change the value of a global variable in the setup.</td>
</tr>
<tr>
<td>df_ek=name</td>
<td>Trigger an action in the device. Start an input chain in signal processing.</td>
</tr>
<tr>
<td>df_msg=This\risr\rMessage,5,1,0</td>
<td>Send a text message to the display.</td>
</tr>
<tr>
<td>df_msg_icon=2</td>
<td>Defines the icon to be used when showing a message in the device. The icon is taken from the design and associate to an input sequence (F2 in this example)</td>
</tr>
<tr>
<td>df_backlight=0,5,255,255,0,192</td>
<td>Defines the colour of a device’s backlight – for a certain period of time as a RGBW value.</td>
</tr>
<tr>
<td>df_info_msg=Info\rMessage,0</td>
<td>Defines the text of an info message.</td>
</tr>
<tr>
<td>df_ac2=010,1,10,20,5</td>
<td>AC = access control. Trigger access control actions.</td>
</tr>
<tr>
<td>df_custom_msg_ac2=010,1,1,0,He\ll%20World</td>
<td>Sends a message to a device that is connect to the access control bus.</td>
</tr>
<tr>
<td>df_ao_ac2=0,1234</td>
<td>Acknowledges an action of the pre-checked access control.</td>
</tr>
<tr>
<td>df_trigger_ac2=1,011,6543210,0</td>
<td>Simulates a clocking performed at an access control RFID reader.</td>
</tr>
<tr>
<td>df_kvp=var,ID</td>
<td>Instructs the device to send the value of a system variable. The value is sent as a key-value-pair to the server.</td>
</tr>
<tr>
<td>df_set_relay=2,close,5</td>
<td>Defines the state of a relay for given period of time that is not handled by the access control module.</td>
</tr>
<tr>
<td>df_toggle_relay=2,5</td>
<td>Changes the state of a relay for a given period of time. The relay may not be handled by the access control module.</td>
</tr>
<tr>
<td>df_load_file=/path/on/server</td>
<td>Instructs the device to download a file from the server.</td>
</tr>
<tr>
<td>df_send_file=/logs/,syslog,0</td>
<td>Instructs the device to upload a file to the server.</td>
</tr>
</tbody>
</table>
### Instruction Name | Meaning
---|---
**df_remove_file**=root:datafox.cert | Instructs the device to delete a specific file.  
**df_remove_finger**=1980,all | Remove fingers from a fingerprint sensor.  
**df_setup_list**=Personal,/path/to/list.txt | Give the device a new list of personnel, for example.  
**df_ac2_list**=Identification,/path/to/list.txt | Give the device a new access control list.  
**df_table_count**=list.PID | Counts the number of entries within a list stored on the device.  
**df_table_select**=list.PID,/upload/form,Unit=Development,PID=5 | Selects on or more entries from a list and uploads them to the server.  
**df_table_append**=list.PID,9999,,Visitor, | Appends a record to a list stored on the device.  
**df_table_update**=list.PID,,Unit= | Changes values within a list stored on the device.  
**df_table_delete**=list.PID,Unit=Development | Removes rows from a list stored on the device.

*Function in work, Not yet usable.*
5.6.5. Encryption

The data fields of the data set can be encrypted using a stream cipher RC4. The field contents are then transferred to their hexadecimal representation.

<table>
<thead>
<tr>
<th>parameter name</th>
<th>importance</th>
</tr>
</thead>
<tbody>
<tr>
<td>df_cb</td>
<td>The parameter specifies that all these fields until and including df_ce have encrypted field contents. The value of df_cb contains the four-digit (1000-9999) public key of the applicable password for the stream cipher.</td>
</tr>
<tr>
<td>df_ce</td>
<td>The parameter indicates that all the following fields are not encrypted any more. If the value is decrypted correctly it must match the value of df_cb.</td>
</tr>
</tbody>
</table>

1.1.1.1. Illustrate the GET request

In plain text (unencrypted) and encrypted:

**Plaintext request**


**Plaintext Reply**

df_api=1&df_time=2017-11-22T08:24:00

**Encrypted request**

df_api=1&df_cb=6102&df_record_state=CC&df_table=66E9B37516AA8C&df_col_sn=0BDC8F79&df_col_recordtype=AB&df_col_badge=AF9B3A929994A5BD7D88&df_col_timestamp=B237B8CA4FA80FD563359C3EE70FE7FC99AF60&df_col_status=9BACFC1E5E0B&df_ce=A344D33B

**Encrypted response**

df_api=1&df_cb=6102&df_time=e1ba6575855619c4d634f7865c01c4b2bc2ec138670ac2&df_ce=a414ebd6

5.6.5.1. Encryption detection

To see whether the data fields are sent encrypted, the initial encryption is with ‘df_cb’ (Datafox Crypt Begin) in and with ‘df_ce’ (Datafox crypt end) in the end. ‘df_cb’ the first field in the request and ‘df_ce’ the last field in the request is.

The value of the field ‘df_cb’, itself is transmitted in plain text and is ‘public key’. It is a random number between 1000 and 9999. The value must be used in conjunction with the communication password for the encryption and decryption.
5.7. https Communication

5.7.1. Requirements
Requirement for using an SSL certificate (https):
Hardware V4:
- Device with TCP/IP (LAN / WLAN) or mobile radio
Minimum firmware 04.03.11. XX (currently usable as prototype firmware)
Software:
- Server must accept an https request and give an active response


5.7.2. Elements of the https infrastructure
Like http, https is a client-server protocol. The client establishes a connection to the port of the https server via TCP/IP, the data stream is encrypted to protect it against listeners.

Both asymmetric encryption (negotiation of the connection) in the form of a server certificate and symmetric encryption for (later) data exchange are used.

5.7.3. Use of encryption / certificates
Several certificates can be stored in the Datafox devices for communication. You can use a certificate signed by Datafox or your own certificate.

The firmware rejects the use of the encryption methods according to specification TLS 1.0, which are no longer considered to be up-to-date (because they are unsafe). Only procedures introduced as of TLS 1.1 are accepted.

Certificates are transferred with the DatafoxStudioIV
The menu item is available as of StudioIV version 04.03.11. XX. You will find them under: "Configuration>Transfer Certificates".
5.8. **Talk**

Datafox - Talk enables the exchange of data with Datafox AEIII+, Timeboy and the Master IV - Series on file and database level. It therefore represents an alternative to communication via DLL and has the great advantage that no programming is required. Lists and devices can be transferred manually or by timer settings data can be read. At extra cost calculation a direct access to customer databases can be provided. Here, the customer determines which database tables and fields are linked.

Datafox- Talk supports all functions for transmitting data.

5.8.1. **Advantages and disadvantages with Datafox Talk**

advantages:
- Easy data exchange via file storage or database
- Automatic fetching of the data via services
- Integrated web server to accept data via http
- No programming required
- Simple updating of master data

disadvantages:
- No access to system variables
- No online functionality
5.8.2. When do I use Talk?

Advantages:
- Expanded use possibilities of the devices
- Simplifies commissioning and maintenance
- No programming skills required
- Communication with any application software
- No need to create a special interface
- Data can be run directly into the application

5.8.3. Overview of the Function Modules Talk

- Free formatting of the output files
- ODBC connection to external databases and XML interfaces
- Special functions possible
- Wireless data transmission with 433 MHz
- Wireless network with multiple access points possible
- Management of radio addresses
- Protocol monitoring only for TimeboyIII
- Data transfer via analogue, ISDN or GSM modem.
- Data transfer directly to the Talk-data server and Maintenance Server on Windows
- Data transfer from your web server via a text file with FTP access. PHP necessary.
- Data transfer from your web server via MySQL database with access to the database.
- Data transfer from your web server via MySQL database with access via http

- Time control
- Data transmission for all Datafox devices via RS232, RS485 and TCP / IP
- Storage of read data as an ASCII file, Excel, XML, dBase, Access
- Loading lists and access lists from ASCII files.
5.8.4. Establishment of Talk

For the establishment of Talk, we recommend a training by Datafox. Current training dates can be found on our website: [http://www.datafox.de/veranstaltungen.html](http://www.datafox.de/veranstaltungen.html)


A quick tour of setting up Datafox Talk.

User interface:

Each Datafox device, data will be exchanged with, will be created here. The type of connection is established, such as TCP/IP or HTTP Comport.

For each module there is a separate service that is installed here as needed and can be started and stopped.
Settings for data export:

Determine where the data should be exported.

Set the storage format.

Settings for time control of the service:
Here are all tasks that the service performs time controlled are created.

Create Event List.

Possible events: