

# ***Remote Access Gateway IGW/92X-W*** ***with DIL/NetPC ADNP/9200***

## **First Steps**



### **SSV Embedded Systems**

Dünenweg 5  
D-30419 Hannover  
Phone: +49 (0)511/40 000-0  
Fax: +49 (0)511/40 000-40  
E-mail: [sales@ssv-embedded.de](mailto:sales@ssv-embedded.de)

Document Revision: 1.0  
Date: 2014-10-20

# CONTENT

<b>1</b>	<b>INTRODUCTION .....</b>	<b>3</b>
1.1	Checklist .....	3
1.2	Conventions .....	3
<b>2</b>	<b>SAFETY GUIDELINES .....</b>	<b>4</b>
<b>3</b>	<b>HARD- AND SOFTWARE REQUIREMENTS .....</b>	<b>5</b>
<b>4</b>	<b>CONNECTIONS .....</b>	<b>6</b>
4.1	RS232 Serial Link .....	6
4.2	Ethernet Link .....	7
4.3	Power Supply .....	8
<b>5</b>	<b>OPERATION .....</b>	<b>9</b>
5.1	Serial Connection .....	9
5.2	Booting the Embedded Linux .....	11
5.3	Testing the TCP/IP Communication .....	13
5.4	Access via Telnet .....	15
5.5	Accessing the SSV Web ConfigTool .....	16
5.6	Changing the IP Address .....	17
5.7	Configuring the WiFi Module .....	18
<b>6</b>	<b>DATA LOGGER DEMO .....</b>	<b>20</b>
<b>7</b>	<b>PLOTLY DEMO .....</b>	<b>21</b>
<b>8</b>	<b>WEBAPP DEMO .....</b>	<b>22</b>
8.1	WebApp Description .....	23
8.2.1.	Commands from Frontend to Backend .....	23
8.2.2.	Data from Backend to Frontend .....	24
8.2	Creating an own WebApp .....	25
<b>9</b>	<b>MAINTENANCE .....</b>	<b>26</b>
<b>10</b>	<b>DISPOSAL .....</b>	<b>26</b>
<b>11</b>	<b>TROUBLE SHOOTING .....</b>	<b>26</b>
<b>12</b>	<b>TECHNICAL DATA .....</b>	<b>27</b>
<b>13</b>	<b>EMC INFORMATION .....</b>	<b>27</b>
<b>14</b>	<b>PINOUT SCREW TERMINALS .....</b>	<b>28</b>
<b>15</b>	<b>HELPFUL LITERATURE .....</b>	<b>29</b>
	<b>CONTACT .....</b>	<b>29</b>
	<b>DOCUMENT HISTORY .....</b>	<b>29</b>

# 1 INTRODUCTION

This documentation is valid for both, the Remote Access Gateway IGW/922-W as well as the IGW/925-W. Therefore the name IGW/92X-W is used for both devices in this document.

Although the examples in this document are made with the IGW/925-W, they all work the same with the IGW/922-W. If there is any difference, it is clearly pointed out.

This documentation gives you an overview about the initial operation and the first steps of use with your new IGW/92X-W.

It also gives a short description of the contained demo programs which offer a simple introduction into the development of applications for the Internet of Things (IoT).

## 1.1 Checklist

Compare the content of your IGW/92X-W start-up package with the checklist below. If any item is missing or appears to be damaged, please contact SSV.

- ✓ IGW/922-W or IGW/925-W
- ✓ Documentation
- ✓ CD-ROM
- ✓ Adapter cable with power and RS232 connector
- ✓ Null modem cable
- ✓ Screwdriver



**Please note:** For operation you will need a 12 – 24 VDC DIN-rail power supply and at least one Ethernet patch cable!

## 1.2 Conventions

Convention	Usage
<b>bold</b>	Important terms
<code>monospace</code>	Filenames, pathnames, program code, user inputs, command lines
<b>[Button]</b>	Name of a button like <b>[OK]</b> or <b>[Apply]</b>

Table 1: Conventions used in this document

## 2 SAFETY GUIDELINES

---



**Please read the following safety guidelines carefully! In case of property or personal damage by not paying attention to this manual and/or by incorrect handling, we do not assume liability. In such cases any warranty claim expires.**

- To provide the IGW/92X-W with power an external DIN-rail power supply is needed. The power supply should be in immediate proximity to the device.
- The DIN-rail power supply must provide a stable output voltage between 12 - 24 VDC. The output power should be at least 2.5 W.
- Please pay attention that the power cord and the other cables are not squeezed or damaged in any way when you set up the IGW/92X-W.
- The installation of the IGW/92X-W should be done only by qualified personnel.
- Discharge yourself electrostatic before you work with the IGW/92X-W, e.g. by touching a heater of metal, to avoid damages.
- Stay grounded while working with the IGW/92X-W to avoid damage through electrostatic discharge.
- The case of the IGW/92X-W should be opened only by qualified personnel.

### 3 HARD- AND SOFTWARE REQUIREMENTS

To work with the IGW/92X-W a software development environment is needed.

This development environment requires a PC to access the IGW/92X-W via serial console or Telnet. Therefore usually a PC with Windows or Linux is used.

The PC should comply with the following hard- and software requirements:

- Windows 7 (or higher) or Linux
- Serial interface (COM port)
- Terminal program (e.g. HyperTerminal or Minicom)
- 10/100 Mbps Ethernet network controller and TCP/IP configuration
- Web browser (e.g. Firefox or Konqueror)
- Telnet client
- CD-ROM or DVD drive
- Internet connection (necessary for the IoT demos)

Please check which IP address your PC currently has and keep this address in mind.

To find out the IP address on a Linux-PC just open a shell console and enter:

```
ifconfig
```

On a Windows-PC open a command prompt (you can find it in the Windows Start menu) and enter:

```
ipconfig
```



**Please note:** With a Linux-PC as development system it is possible to develop programs in all languages for the IGW/92X-W. Linux is especially for programming in C/C++ as PC operating system required because of the GNU cross tools. With a Windows-PC the shell-script programming as well as the HTML- and Python programming of the Web server is possible.

## 4 CONNECTIONS

For a quick and easy start with the IGW/92X-W there are a few connections necessary.

The following chapters describe how these connections between the development system (host PC) and the IGW/92X-W have to be made.

### 4.1 RS232 Serial Link

For a basic communication with the IGW/92X-W use the null modem cable and connect the **COM1** port of the IGW/92X-W with a free COM port of your host PC.

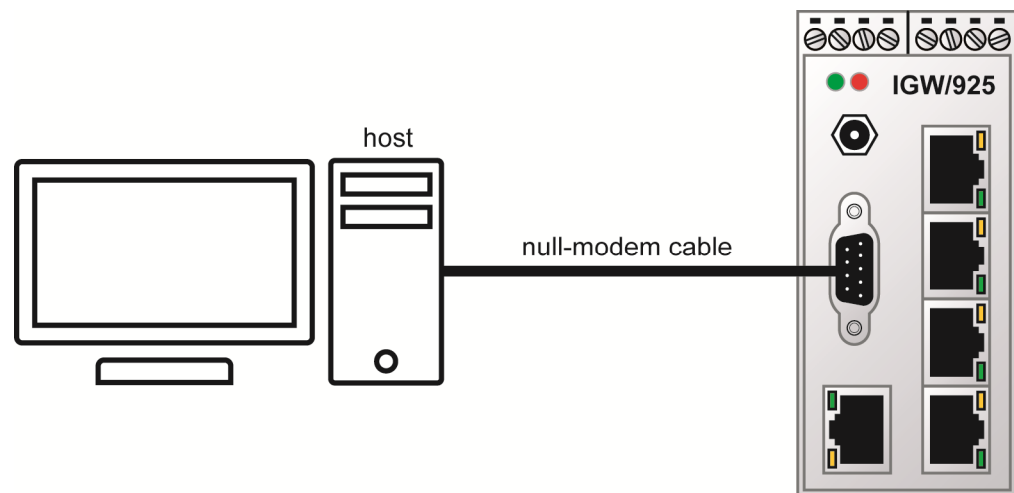


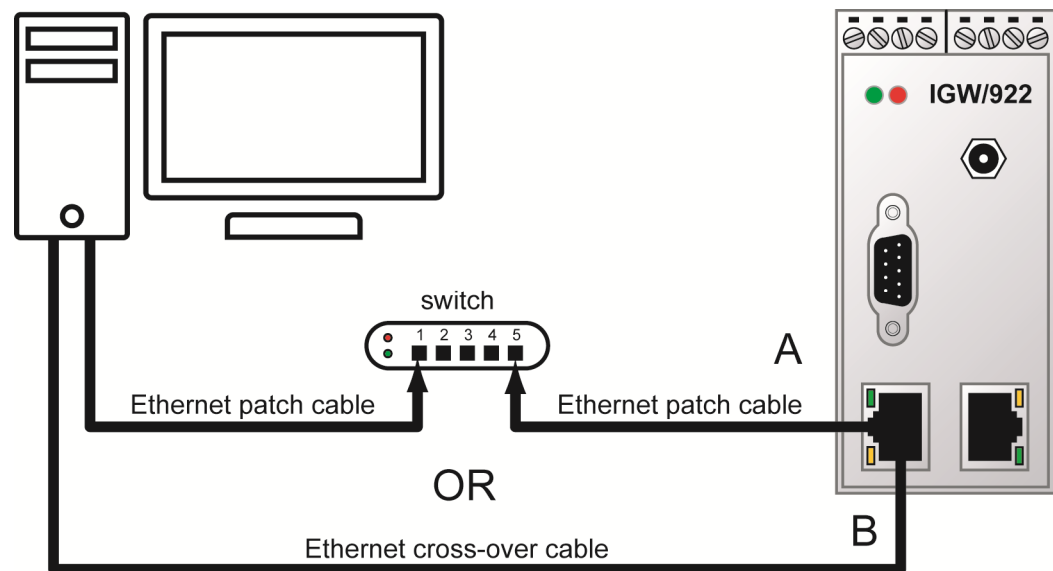
Figure 1: RS232 link on serial port COM1

## 4.2 Ethernet Link

For an Ethernet link between an **IGW/922-W** and the host PC there are two possible connection setups like shown in **fig. 2**.

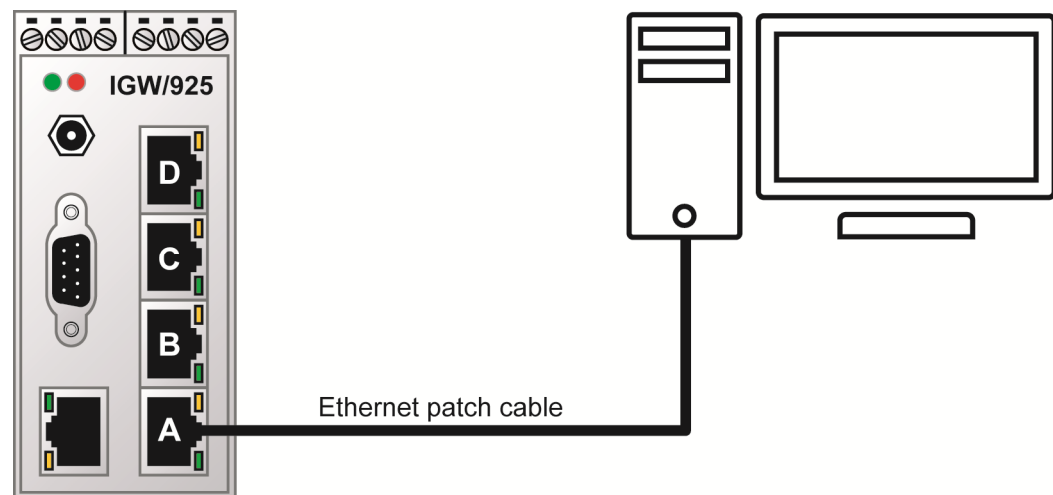
For connection **A** use two standard Ethernet patch cables with a LAN switch in between.

For connection **B** use one Ethernet cross-over cable.



**Figure 2: Ethernet link between IGW/922-W and host PC**

For an Ethernet link between an **IGW/925-W** and the host PC use an Ethernet cross-over cable like shown in **fig. 3**.



**Figure 3: Ethernet link between IGW/925-W and host PC**

### 4.3 Power Supply

The IGW/92X-W needs a supply voltage of **11 .. 28 VDC** to work.

Use the power adapter cable to provide the system with the necessary power like shown in the figure below.

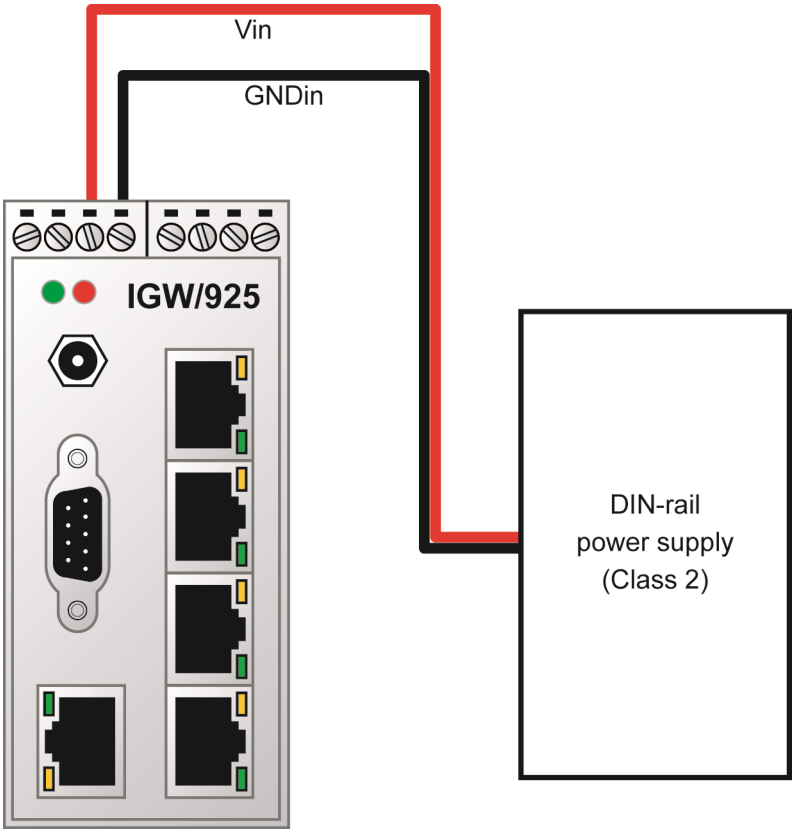


Figure 4: Power supply for the IGW/92X-W

Terminal	Signal
A3	Vin (11 .. 28 VDC)
A4	GNDin

Table 2: Screw terminal power



**CAUTION!**  
Providing the IGW/92X-W with a higher voltage than the regular 11 .. 28 VDC could cause damaged device components!



## 5 OPERATION

### 5.1 Serial Connection

To make communication possible between the IGW/92X-W and the development system a terminal program is needed.

Under Windows it is normally **HyperTerminal**, for Linux exists **Minicom**, an equivalent program. In the following we use HyperTerminal, but all settings can be made similarly with other terminal programs.

If there is no terminal program installed on your development system (this could be the case for a Linux-based PC), you have to install it manually from your operating systems installation CD-ROM.

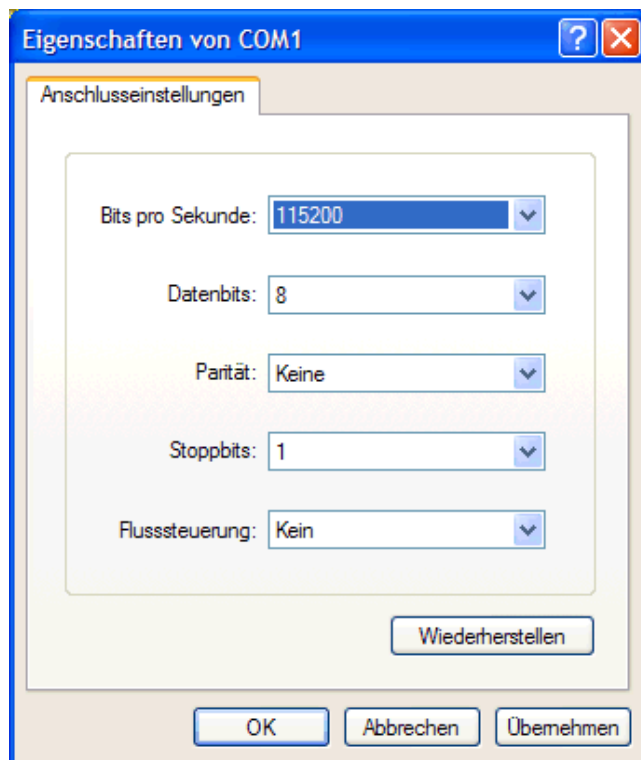
First open a new **HyperTerminal** session and enter a name for it (e.g. "IGW/925-W").

Choose now under „**File->Properties**“ under „**Connect via**“ the entry „**direct link cable connection via COM1**“ (or the COM-port you used). In the next step click on the button „**configure**“ to get to the next dialog box.



Figure 5: Interface dialog box

Now change the settings to the parameters shown in **fig. 6** and close the dialog box with a click on „OK“.



**Figure 6: Communication parameter settings**

These settings can also be made in other terminal programs. Important are the following parameters and values:

Parameter	Wert
Connection speed	115.200 bps
Data bits	8
Parity	None
Stop bits	1
Handshake	None

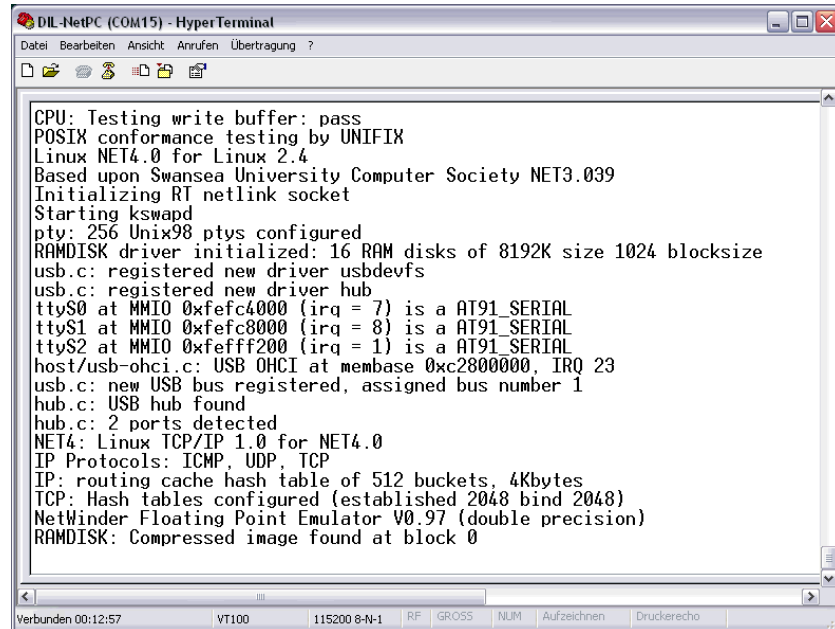
**Table 3: Terminal program settings**



**Please note:** Do not forget to save the session settings with the connection parameters under the name you have chosen (e.g. “IGW/925-W”). With this name you can start the terminal program for future IGW/92X-W-access.

## 5.2 Booting the Embedded Linux

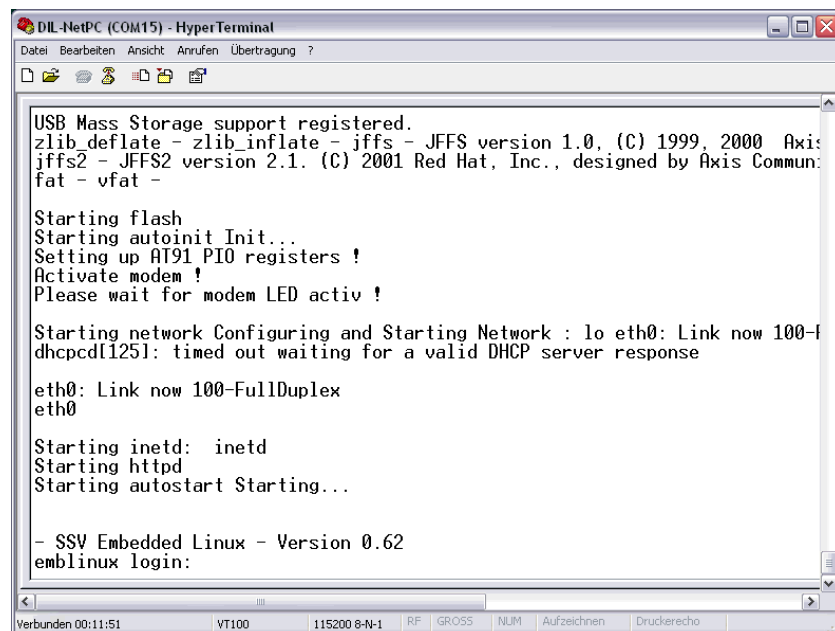
Now you may power up the IGW/92X-W. The boot process starts immediately. The IGW/92X-W boots thereby an embedded Linux out of its Flash memory. The **fig. 7** shows the corresponding messages during this procedure with the Windows terminal program **HyperTerminal**.



```
CPU: Testing write buffer: pass
POSIX conformance testing by UNIFIX
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Initializing RT netlink socket
Starting kswapd
tty: 256 Unix98 ptys configured
RAMDISK driver initialized: 16 RAM disks of 8192K size 1024 blocksize
usb.c: registered new driver usbdevfs
usb.c: registered new driver hub
ttyS0 at MMIO 0xfefc4000 (irq = 7) is a AT91_SERIAL
ttyS1 at MMIO 0xfefc8000 (irq = 8) is a AT91_SERIAL
ttyS2 at MMIO 0xffeff200 (irq = 1) is a AT91_SERIAL
host/usb-ohci.c: USB OHCI at membase 0xc2800000, IRQ 23
usb.c: new USB bus registered, assigned bus number 1
hub.c: USB hub found
hub.c: 2 ports detected
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 2048 bind 2048)
NetWinder Floating Point Emulator V0.97 (double precision)
RAMDISK: Compressed image found at block 0
```

**Figure 7: Embedded Linux boot process**

The Linux boot process of the IGW/92X-W ends with the following command prompt:



```
USB Mass Storage support registered.
zlib_deflate - zlib_inflate - jffs - JFFS version 1.0, (C) 1999, 2000 Axis
jffs2 - JFFS2 version 2.1. (C) 2001 Red Hat, Inc., designed by Axis Commun
fat - vfat -

Starting flash
Starting autoinit Init...
Setting up AT91 PIO registers !
Activate modem !
Please wait for modem LED activ !

Starting network Configuring and Starting Network : lo eth0: Link now 100-f
dhcpcd[125]: timed out waiting for a valid DHCP server response

eth0: Link now 100-FullDuplex
eth0

Starting inetd: inetd
Starting httpd
Starting autostart Starting...

- SSV Embedded Linux - Version 0.62
emblinux login:
```

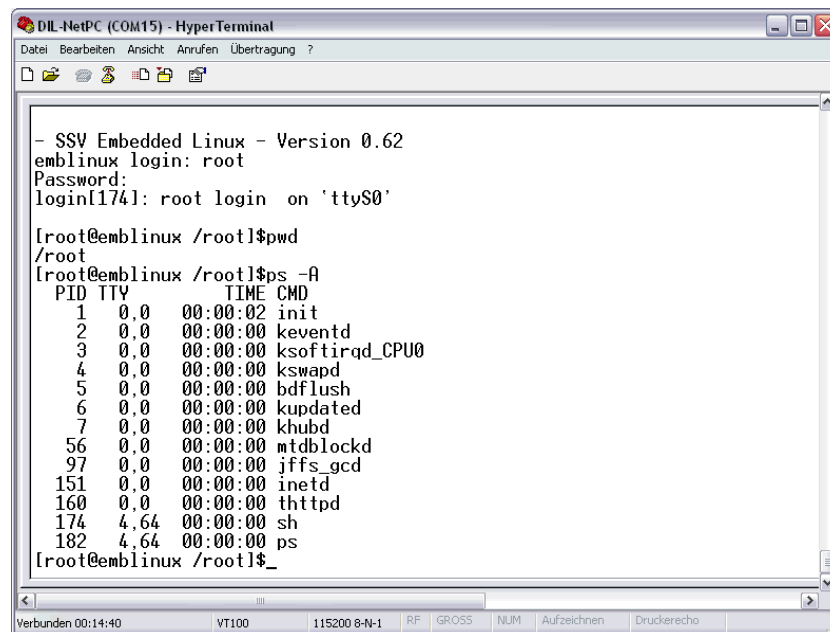
**Figure 8: Linux command prompt**

The command prompt is a part of the serial console of the DIL/NetPC ADNP/9200 within the IGW/92X-W. To login enter **root** as username and password. You can now enter any Linux commands, which will be executed by the DIL/NetPC operating system.

Command	Function
cat	Show file content or write input in file
cd	Change directory
chmod	Change access right for a file
cp	Copy file
ls	List directory- and filenames, access rights, etc.
mkdir	Make new directory
mv	Move file
ps	Overview of active processes
pwd	Show name of actual directory
rm	Remove file
rmdir	Remove empty directory

**Table 4: Some important DIL/NetPC Linux commands**

Every Linux command must be confirmed with a carriage return. Some commands need additional parameters.



```

- SSV Embedded Linux - Version 0.62
emlinux login: root
Password:
login[174]: root login on 'ttyS0'

[root@emlinux /root]$pwd
/root
[root@emlinux /root]$ps -A
  PID TTY          TIME CMD
    1 0.0    00:00:02 init
    2 0.0    00:00:00 keventd
    3 0.0    00:00:00 ksoftirqd_CPU0
    4 0.0    00:00:00 kswapd
    5 0.0    00:00:00 bdfush
    6 0.0    00:00:00 kupdated
    7 0.0    00:00:00 khud
   56 0.0    00:00:00 mtdblockd
   97 0.0    00:00:00 jffs_gcd
  151 0.0    00:00:00 inetd
  160 0.0    00:00:00 thttpd
  174 4.64   00:00:00 sh
  182 4.64   00:00:00 ps
[root@emlinux /root]$

```

**Figure 9: Executing Linux commands with a serial console**



**Please note:** The same DIL/NetPC Linux commands are also executable a Telnet session (see [chapter 5.4](#)).

### 5.3 Testing the TCP/IP Communication

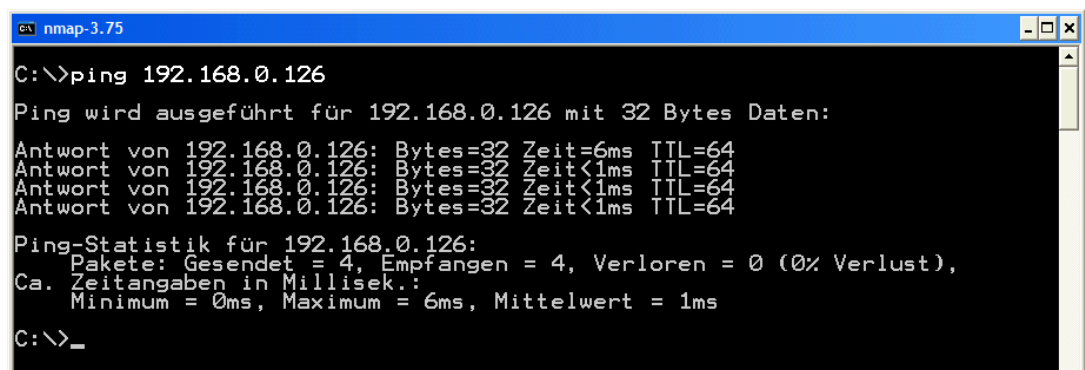
To test the Ethernet connection on the basis of TCP/IP-protocols between the IGW/92X-W and the host PC we use the very popular TCP/IP-utility program **ping**.

Open a command prompt on the host PC (you can find it in the Windows Start menu) and enter:

```
ping 192.168.0.126
```

Directly after this command test data is send from the host PC to the IGW/92X-W via Ethernet LAN. The IGW/92X-W must answer on the same way within a certain period of time.

The **fig. 10** shows the entry of the command as well as the successful execution. In case of an error **ping** reports a timeout.



```
C:\>ping 192.168.0.126
Ping wird ausgeführt für 192.168.0.126 mit 32 Bytes Daten:
Antwort von 192.168.0.126: Bytes=32 Zeit=6ms TTL=64
Antwort von 192.168.0.126: Bytes=32 Zeit<1ms TTL=64
Antwort von 192.168.0.126: Bytes=32 Zeit<1ms TTL=64
Antwort von 192.168.0.126: Bytes=32 Zeit<1ms TTL=64
Ping-Statistik für 192.168.0.126:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),
    Ca. Zeitangaben in Millisek.:
        Minimum = 0ms, Maximum = 6ms, Mittelwert = 1ms
C:\>_
```

**Figure 10: Communication check via ping**

The IGW/92X-W must answer every **ping**. Otherwise an error will occur. In this case you have to check all parts of your LAN connection, including the IP address of the host PC. The value of the IP address should be **192.168.0.1**.


For an easy check of the IP address within the command prompt, use the following command:

```
ipconfig
```

The **fig. 11** shows the **ipconfig**-messages of a Windows-XP-PC.



**Please note:** We assume at this point that the IGW/92X-W has the ex factory IP address **192.168.0.126**. If you have changed this address, you must regard this for the ping parameter.



```
C:\>ipconfig

Windows-IP-Konfiguration

Ethernetadapter LAN-Verbindung:

    Verbindungsspezifisches DNS-Suffix:
    IP-Adresse. . . . . : 192.168.0.1
    Subnetzmaske. . . . . : 255.255.255.0
    Standardgateway . . . . . :

Ethernetadapter Drahtlose Netzwerkverbindung:

    Medienstatus. . . . . : Es besteht keine Verbindung
Ethernetadapter LAN-Verbindung 3:

    Medienstatus. . . . . : Es besteht keine Verbindung
Ethernetadapter LAN-Verbindung 2:

    Medienstatus. . . . . : Es besteht keine Verbindung

C:\>_
```

**Figure 11: Communication check via ipconfig command**

Once the ping was successful, you are ready to start a web browser (e.g. Firefox or Konqueror).

## 5.4 Access via Telnet

To access the IGW/92X-W with a Telnet client open a command prompt on your host PC (you can find it in the Windows Start menu) and enter the following command to activate a Telnet session:

```
telnet 192.168.0.126
```

The Linux of the IGW/92X-W is configured in a way, which you have to login with the username **root** and the password **root**.

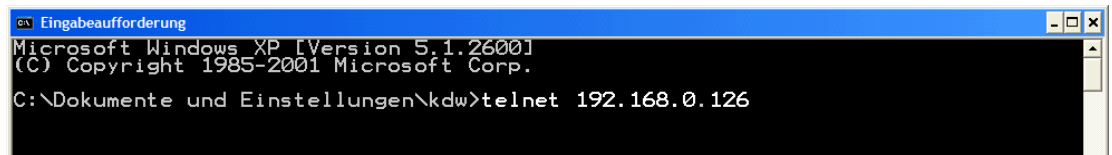


Figure 12: Starting the Telnet client

In the upcoming Telnet window you can enter any Linux commands, which will be executed by the IGW/92X-W operating system (similar to the serial console described in **chapter 5.2**).

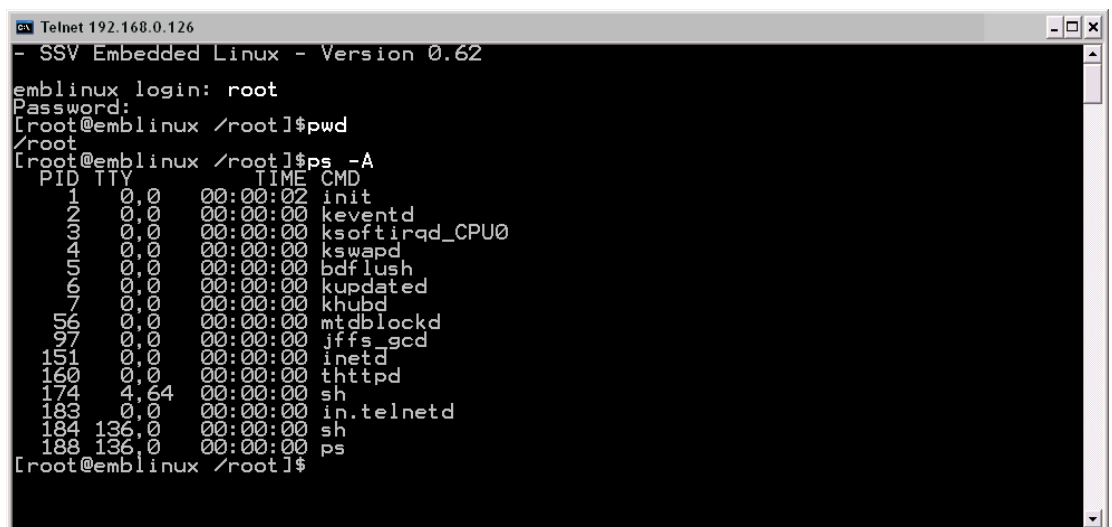


Figure 13: Access via Telnet client



**Please note:** The Telnet command in **fig. 13** requires the current IP address of the IGW/92X-W as parameter.

## 5.5 Accessing the SSV Web ConfigTool

To open the login page of the SSV Web ConfigTool enter the following URL in a Web browser (do not forget to enter the port number 7777!):

`http://192.168.0.126:7777.`

The following passwords can be used to login:

- The default **standard user** password is: **adnp**.

The standard user has an **idle timeout**. The password and timeout can be changed in the menu over “**System > Administration**”.

- The default **master user** password is: **ssvadmin**.

The master user has **no idle timeout**. The password can be changed in the menu over “**System > Administration**”.

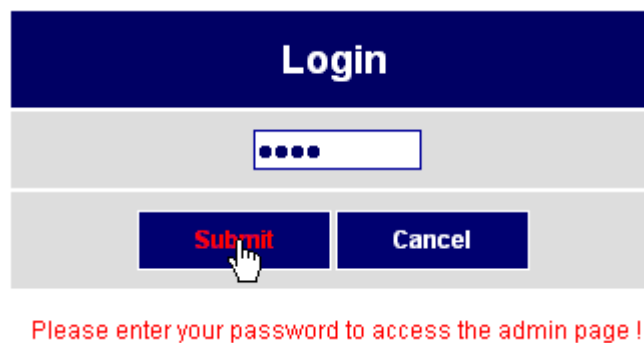


Figure 14: Login page of the SSV Web ConfigTool



**Please note:** If a standard user is already logged in, he will be automatically logged out when the master user logs in. In contrast to the standard user it is possible to log in more than once at the same time with the master password. **Although it is possible it is not recommended!**



## 5.6 Changing the IP Address

To change the IP address of the IGW/92X-W enter the Web ConfigTool and choose from the menu “**Network > LAN**”. In this section you can configure the LAN settings.

To save the settings click on the button [**Apply**] and the changes are immediately active.

Network configuration for LAN1 (10/100 MBit)			
Enable/Disable interface LAN1 :	<input checked="" type="checkbox"/>		Enable or disable interface LAN1
Obtain an IP address automatically :	<input type="radio"/>		Device configuration through DHCP server
Use the following IP address :	<input checked="" type="radio"/>		Manual device configuration
IP address :	192	168	0 75
Subnet mask :	255	255	255 0
Enable/Disable alias IP address :	<input type="checkbox"/>		Enable or disable alias IP address

Figure 15: LAN settings

- **Enable/Disable interface LAN:** Enable or disable interface LAN.
- **Obtain an IP address automatically:** Device configuration through DHCP server.
- **Use the following IP address:** Manual device configuration.
- **IP address:** IP address of the device.
- **Subnet mask:** Subnet mask of the local network.
- **Enable/Disable alias IP address:** Enable or disable the alias IP address.
- **Alias IP address:** Secondary static IP address for the same interface.
- **Alias subnet mask:** Subnet mask of the alias network.

For more detailed information about the Web ConfigTool please read the “SSV Web ConfigTool user manual”.

## 5.7 Configuring the WiFi Module

To configure the integrated WiFi module of the IGW/92X-W enter the Web ConfigTool and choose from the menu “**Network > WiFi**”.

To save the settings click on the button [**Apply**] and the changes are immediately active.

Wireless network configuration		
Enable/Disable WiFi AP :	<input checked="" type="checkbox"/>	Select to enable or disable the AP.
SSID :	IGW922-W	Specify the name of the wireless network.
Channel :	5	Specify the channel for the AP.
DHCP IP range :	192.168.1.1 /24	Specify DHCP IP range.
Security suite:	WPA2	Specify the security suite to be used.
Passphrase :	IGW922-W	Enter the passphrase.

HTTP Server configuration		
Enable/Disable HTTP server :	<input type="checkbox"/>	Select to enable or disable the HTTP server.

TCP tunnel configuration		
Enable/Disable TCP tunnel :	<input checked="" type="checkbox"/>	Select to enable or disable the TCP tunnel.
TCP accept port :	80	Enter the port to listen on.
TCP timeout :	3 sek	Close port if idle after timeout.

**Figure 16: WiFi module configuration**

In the first section the WiFi modules **general settings** can be configured.

- **Enable/Disable WiFi AP:** Enable or disable the WiFi access point (AP).
- **SSID:** Specify the name of the wireless network.
- **Channel:** Select the desired WiFi channel.
- **DHCP IP range:** Specify the IP range for the WiFi modules DHCP server.
- **Security suite:** Select the desired security suite.
- **Passphrase:** Enter a password for the WiFi network.

In the second section the **internal Web server** of the WiFi module can be configured.

- **Enable/Disable HTTP server:** Enable or disable the WiFi modules Web server.
- **Server port:** Enter the port number under which the Web server is available.

The WiFi modules Web server offers an own Web configuration. In the standard configuration the WiFi Web server is available under the following URL (do not forget to enter the server port number!):

`http://192.168.1.1:8080`



**Figure 17: Status page of WiFi modules own Web configuration**

To access the WiFi modules own Web configuration enter the following login data:

User: **admin**  
Password: **ssvadmin**

In the third section the **TCP tunnel** of the WiFi module can be configured.

- **Enable/Disable TCP tunnel:** Enable or disable the WiFi modules TCP tunnel.
- **TCP accept port:** Enter the port to listen on.
- **TCP timeout:** Select the idle time after which the port is closed.

## 6 DATA LOGGER DEMO

---

The data logger demo is a python script which collects weather data (the preconfigured weather region is Hanover) and saves it in an SQLite database on the internal microSD card.

The data is obtained over the Internet from Yahoo in an interval of 15 minutes.

The SQLite database is stored on the microSD card at `/media/mmc/data.db`.

The data logger demo is called `yweather.py` and is stored in the flash memory of the IGW/92X-W at `/flash/demo`.

To run and exit the script use the following commands in a terminal program (see **chapter 5.2**) or during a Telnet session (see **chapter 5.4**):

To start the demo enter: `/flash/demo/yweather.py`

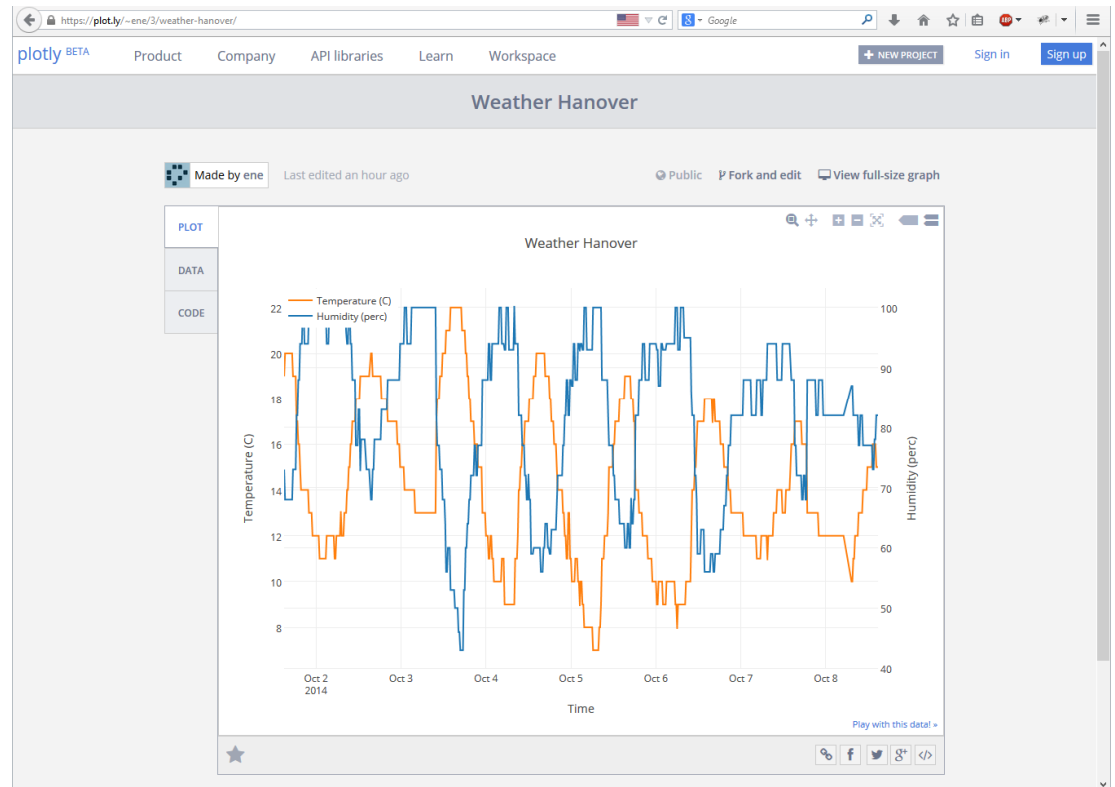
To stop the demo enter: `/flash/demo/yweather.py -s`

To start the demo in the foreground enter: `/flash/demo/yweather.py -f`

This demo needs a working Internet connection.

## 7 PLOTLY DEMO

The plotly demo is a python script which creates a graph out of the collected weather data with the help of the online framework **plotly** (<https://plot.ly>).



**Figure 18: Screenshot of the plotly demo**

The plotly demo is called **plotly.py** and is stored in the flash memory of the IGW/92X-W at **/flash/demo**.

The plotly demo requires the running **data logger demo** (see **chapter 6**). The plotly graph is updated in an interval of one hour.

To use the demo it is recommended to create a plotly account. The account data like user-name and API key have to be entered in the demo. The complying variables are called **PLOTLY\_USER** and **PLOTLY\_KEY**.

To run and exit the script use the following commands in a terminal program (see **chapter 5.2**) or during a Telnet session (see **chapter 5.4**):

To start the demo enter: **/flash/demo/plotly.py**

To stop the demo enter: **/flash/demo/plotly.py -s**

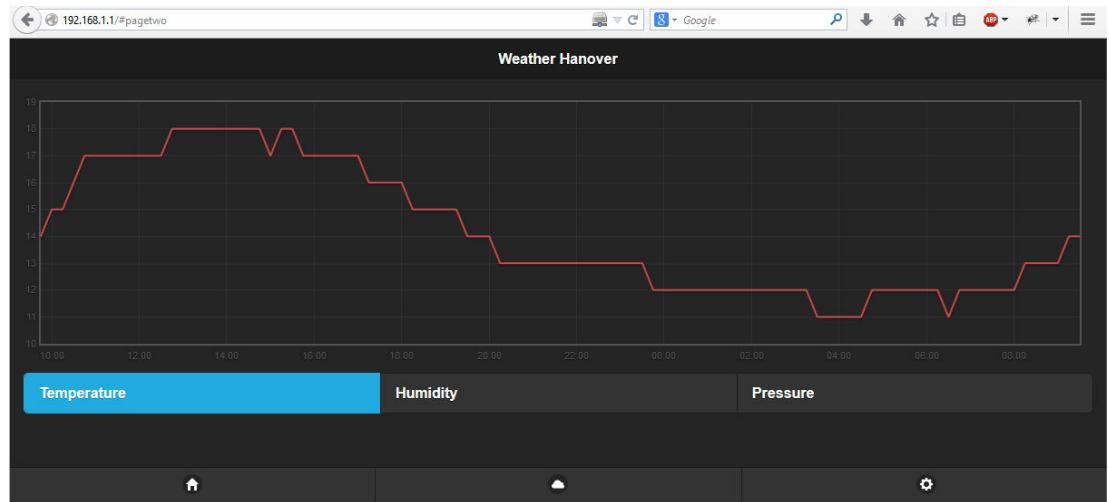
To start the demo in the foreground enter: **/flash/demo/plotly.py -f**

This demo needs a working Internet connection.

## 8 WEBAPP DEMO

The WebApp demo is enabled by default and can be opened with the following URL:

**`http://192.168.1.1.`**



**Figure 19: Screenshot of the WebApp demo**

The WebApp demo is a python script which shows how the serial tunnel of the WiFi module can be used to create a mobile web application. The example consists of a web application based on the **jQuery mobile framework 1.4.3** and offers three pages:

- “SolarLog” is an example to display rapid changing data.
- “Weather Hanover” is an example to display graphs (here the weather data from the SQLite database).
- “Control” is an example to control actors (here the IGW/92X-W user LED and the alarm relay).

The web pages include all required resources (CSS, JavaScript and images), because the WiFi module can only serve a single connection at the configured port, while a browser loads all resources parallel. In addition the single pages are compressed with **gzip** for a faster download.

The WebApp demo is called **webapp.py** and is stored in the flash memory of the IGW/92X-W at **/flash/demo**.

To run and exit the script use the following commands in a terminal program (see **chapter 5.2**) or during a Telnet session (see **chapter 5.4**):

To start the demo enter: **`/flash/demo/webapp.py`**

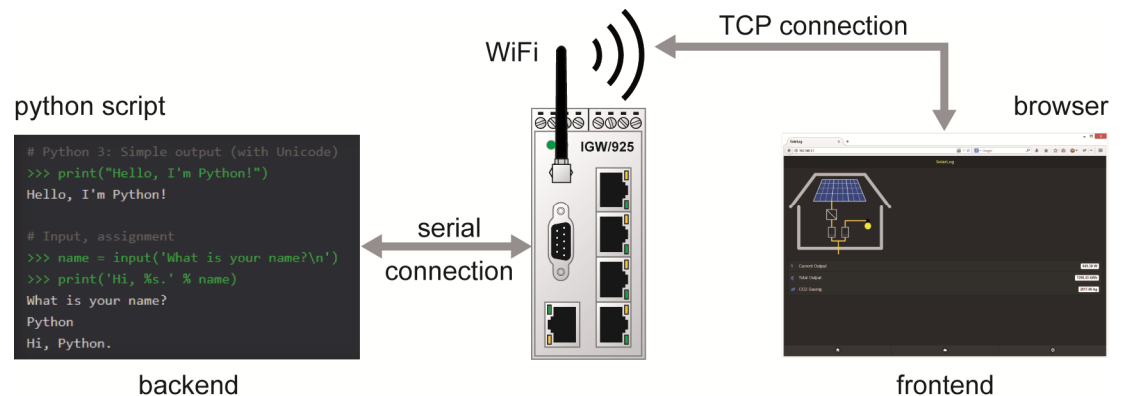
To stop the demo enter: **`/flash/demo/webapp.py -s`**

To start the demo in the foreground enter: **`/flash/demo/webapp.py -f`**

## 8.1 WebApp Description

The WebApp backend is the python script `webapp.py` which runs on the IGW/92X-W and responses to the browser requests.

The internal communication between the python script and the WiFi module is serial with a baud rate of 230400 and hardware handshake.



**Figure 20: Communication between python script, WiFi module and browser**

The WebApp frontend is an HTML-file called `webapp.html.gz`.

The frontend and backend communicate via TCP over **WebSockets**. The data is transferred in **JSON** format. The WebSocket address ist:

```
http://192.168.1.1/ws
```

The API offers five different commands which can be transferred alone or together.

The API syntax is as follows:

```
{ <command1>:<data>, <command2>:<data>, ... }
```

### 8.1.1. Commands from Frontend to Backend

**"s" = Subscribe:** Registers/subscribes one or more variables at the backend. If the variable changes, the server sends it automatically to the frontend. The data is an array with the variable name(s).

```
Example:
{ "s" : ["v_temp", "v_volt", "v_hum"] }
```

**"u" = Unsubscribe:** Deletes one or more registrations/subscriptions.

```
Example:
{ "u" : ["v_temp", "v_volt"] }
```

**"r" = Read:** Reads one or more variables.

```
Example:
{ "r" : ["v_temp", "v_volt"] }
```

"w" = **Write**: Sends one or more variables to the backend which writes it into the SQLite database. A written/changed variable is then sent automatically back to the frontend.

Example:

```
{ "w" : { "v_temp":20, "v_volt":40.1 } }
```

"c" = **Command**: Sends one or more commands to the backend. Currently there is only the command "**exit**" available, which stops a connection.

Example:

```
{ "c" : ["exit"] }
```

### 8.1.2. Data from Backend to Frontend

The backends response to a frontends request is always a JSON object with **key:value** pairs (which can be also empty).

Example:

```
{ "v_temp":20, "v_volt":40.1 }
```



## 8.2 Creating an own WebApp

To create an own WebApp use the template `/flash/demo/demo.src.tgz`. This archive contains two files:

- `demo.py` (for the backend server)
- `demo.html` (for the frontend client)

Unzip the files and follow these steps:

1. Rename the python script `demo.py` to `demo1.py` and customize it like wanted.
2. Rename the HTML file `demo.html` to `demo1.html` and customize it like wanted. If there are external CSS or JavaScript files linked embed them as inline code into `demo1.html`.

### Important:

The script `/flash/demo/build.sh` substitutes external jQuery and CSS files within the HTML file automatically, if they are linked as follows:

```
<link rel="stylesheet" href="jquery.mobile.inline-png-1.4.3.min.css">
```

```
<script src="jquery.min.js"></script>
```

```
<script src="jquery.mobile-1.4.3.min.js"></script>
```

If the `build.sh` script finds these lines, they are substituted by the content of the linked files.

3. If there are images linked in the `demo1.html` they must be converted to **inline Base64 images**.
4. Now start the script `build.sh` with the following command:

```
sh build.sh demo1
```

This command creates two files:

<code>webapp.py</code>	backend server (copy of <code>demo1.py</code> )
<code>webapp.html.gz</code>	frontend client (gzipped <code>demo1.html</code> with included jQuery framework)

5. Transfer the files `webapp.py` and `webapp.html.gz` into the folder `/flash/demo` on the IGW/92X-W and start the WebApp with the command:

```
/flash/demo/webapp.py
```

---

## 9 MAINTENANCE

---

A special maintenance for the IGW/92X-W is not necessary. For technical questions please send an e-mail to:

`support@ssv-embedded.de`

---

## 10 DISPOSAL

---

A special disposal of the IGW/92X-W beyond the regulation by law is not necessary. For questions please send an e-mail to

`support@ssv-embedded.de`

---

## 11 TROUBLE SHOOTING

---

If you experience troubles with the IGW/92X-W please proceed as follows:

Check all cable connections carefully. Create a serial RS232-connection – like described in **chapter 4.1** – with your development system and try to access the IGW/92X-W with a terminal program.

Reset if necessary the IGW/92X-W by interrupting the power supply for a moment. Afterwards the IGW/92X-W should execute a new boot process, which you can watch with a terminal program.

If you still have troubles, please send an e-mail to:

`support@ssv-embedded.de`

## 12 TECHNICAL DATA

Supply voltage..... 11 – 28 VDC

Weight ..... < 0,5 kg

Mechanical Dimensions (LxWxH) ..... 112 mm x 45 mm x 100 mm

Temperature range ..... 0° C – 70° C

Rel. air humidity ..... max. 85 %

## 13 EMC INFORMATION

The IGW/92X-W meets following EU directives about electromagnetic compatibility (EMC):

Emitted interference	
Radio disturbance	EN 55022 (industrial environment)
Interference immunity	
<b>Case</b>	
Radiated RF immunity	EN 61000-4-3
Electrostatic Discharge (ESD)	EN 61000-4-2
<b>Signal connections</b>	
Conducted RF immunity	EN 61000-4-6
Electrical fast transients/bursts	EN 61000-4-4
<b>Direct current input and output</b>	
Conducted RF immunity	EN 61000-4-6
Electrical fast transients/bursts	EN 61000-4-4

**Table 5: EMC information**



**Please note:** Do not connect cables longer than 3 meters with the screw terminals.

# 14 PINOUT SCREW TERMINALS

















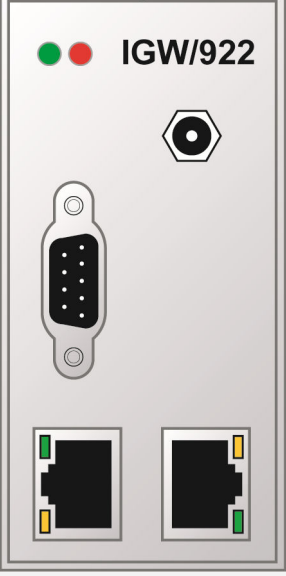
								Pin	Power	COM2	Alarm
A				B							
1	2	3	4	1	2	3	4				
								A1			Alarm –
								A2			Alarm +
								A3	V+ IN		
								A4	GND IN		
								B1			
								B2		TXD (RS232) RX/TX – (RS485)*	
								B3		RXD (RS232) RX/TX + (RS485)*	
								B4		GND	

Table 6: Pinout of the screw terminals



\* **Please note:** The RS485 (officially called TIA/EIA-485-A) connection between your IGW/92X-W and the field devices needs termination resistors on both ends for proper operation. The IGW/92X-W does **not** offer internal termination resistors. Please make sure, that the RS485 cable connection is equipped with external termination resistors.

## 15 HELPFUL LITERATURE

- IGW/922-W hardware reference manual
- IGW/925-W hardware reference manual
- E2W/ESL2 hardware reference manual
- DIL/NetPC ADNP/9200 hardware reference manual
- SSV Web ConfigTool user manual

## CONTACT

### SSV Embedded Systems

Dünenweg 5  
D-30419 Hannover

Phone: +49 (0)511/40 000-0  
Fax: +49 (0)511/40 000-40  
E-mail: [sales@ssv-embedded.de](mailto:sales@ssv-embedded.de)

Internet: [www.ssv-comm.de](http://www.ssv-comm.de)  
Forum: [www.ssv-comm.de/forum](http://www.ssv-comm.de/forum)

## DOCUMENT HISTORY

Revision	Date	Remarks	Name	Review
1.0	2014-10-20	First version	WBU	ENE

The content of this document can change any time without announcement. There is taken over no guarantee for the accuracy of the statements. The user assumes the entire risk as to the accuracy and the use of this document. Information in this document is provided 'as is' without warranty of any kind. Some names within this document can be trademarks of their respective holders.

© 2014 SSV EMBEDDED SYSTEMS. All rights reserved.