WiFi 16 Relay Board
TCP ModBus Controlled

User Manual
Date: 21 Aug 2017

Content

Document Revision History

Version	Date	Comments
1.00	17/05/02	The initial release
2.00	17/08/21	Second release

Trademark Notices

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Apple, Mac, Mac OS, Mac OS logo are either trademarks or registered trademarks of Apple Computer Inc. in the United States and/or other countries.

Other product names and company names described in this document are trademarks or registered trademarks.

1. Specification

Relays:

- 16 x SPDT relays, 10A / 250VAC, 15A / 120VAC, 10A / 28VDC

Power supply requirement, selectable during purchase:

- 12V DC / 600mA
- 24V DC / 400mA

Wireless Wi-Fi IEEE 802.11 standart:

- Wireless Wi-Fi 802.11 b/g/n;
- External Wi-Fi antenna for better range;
- Supported modes: AP (Access Point), STA (Station) and AP+STA (Access Point + Station);
- Supported encryption modes:
  - WEP;
  - WPAPSK-TKIP;
  - WPAPSK-AES;
  - WPA2PSK-TKIP;
  - WPA2PSK-AES;

Web Interface:

- Configuration of system parameters;
- Secure login authorization;

Network Protocols:

- Standart protocols: DHCP, DNS, ICMP (ping);
- Virtual Serial Port;
- ModBus TCP/IP;

PCB Parameters:

- Led indicators: Relays, Power ON, Wi-Fi link, Ready Status;
- FR4 / 1.5mm / two layers / metalized holes / HAL / white stamp / solder mask / Extra PCB openings for better voltage isolation / Doubled PCB tracks for better voltage isolation;

Physical and Environment:

- Operating temperature range: from 0□ºC to +70□ºC;
- Offered in two versions: PCB only and with DIN RAIL BOX;
- Dimensions:
  - PCB: W=82mm x L=203mm x H=24mm
  - BOX: W=210mm x L=85mm x L=58mm

2. Applications examples

2.1. Control electrical devices wirelessly

The main feature of the module is to control electrical devices wirelessly. It carries 16 relay channels. With lots of free and low cost Modbus applications this module is turned in really cost effective solution for wireless control of electrical devices.

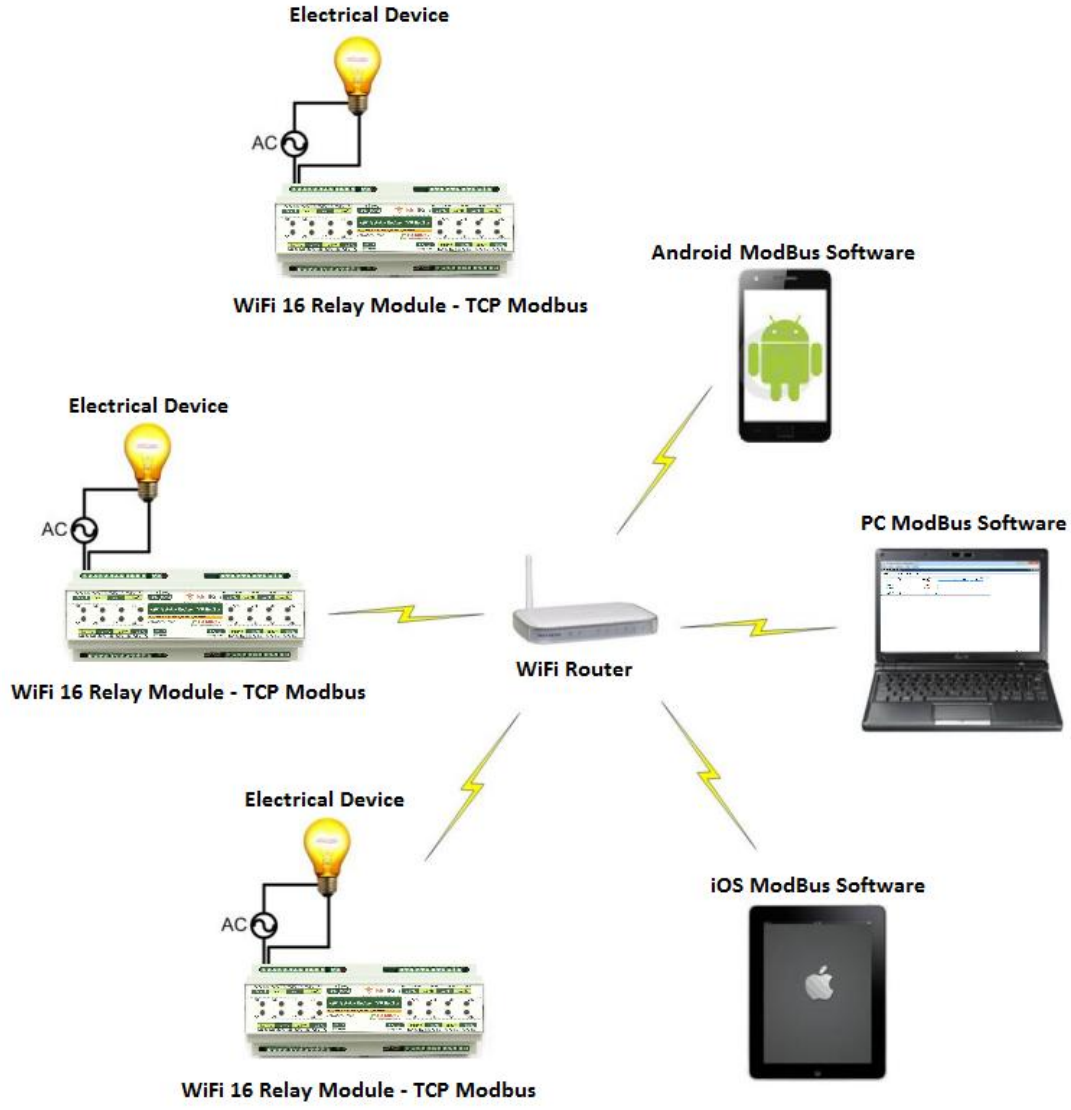


Figure 1. Wi-Fi 16 Relay Module – TCP ModBus' main feature is to provide wireless control for appliances

2.2. Control electrical devices via Internet

Usually the next step is to control the system via the Internet. The module supports TCP ModBus Protocol. There are lot of software on the market for TCP ModBus so it's really simple to be organized this scenario.

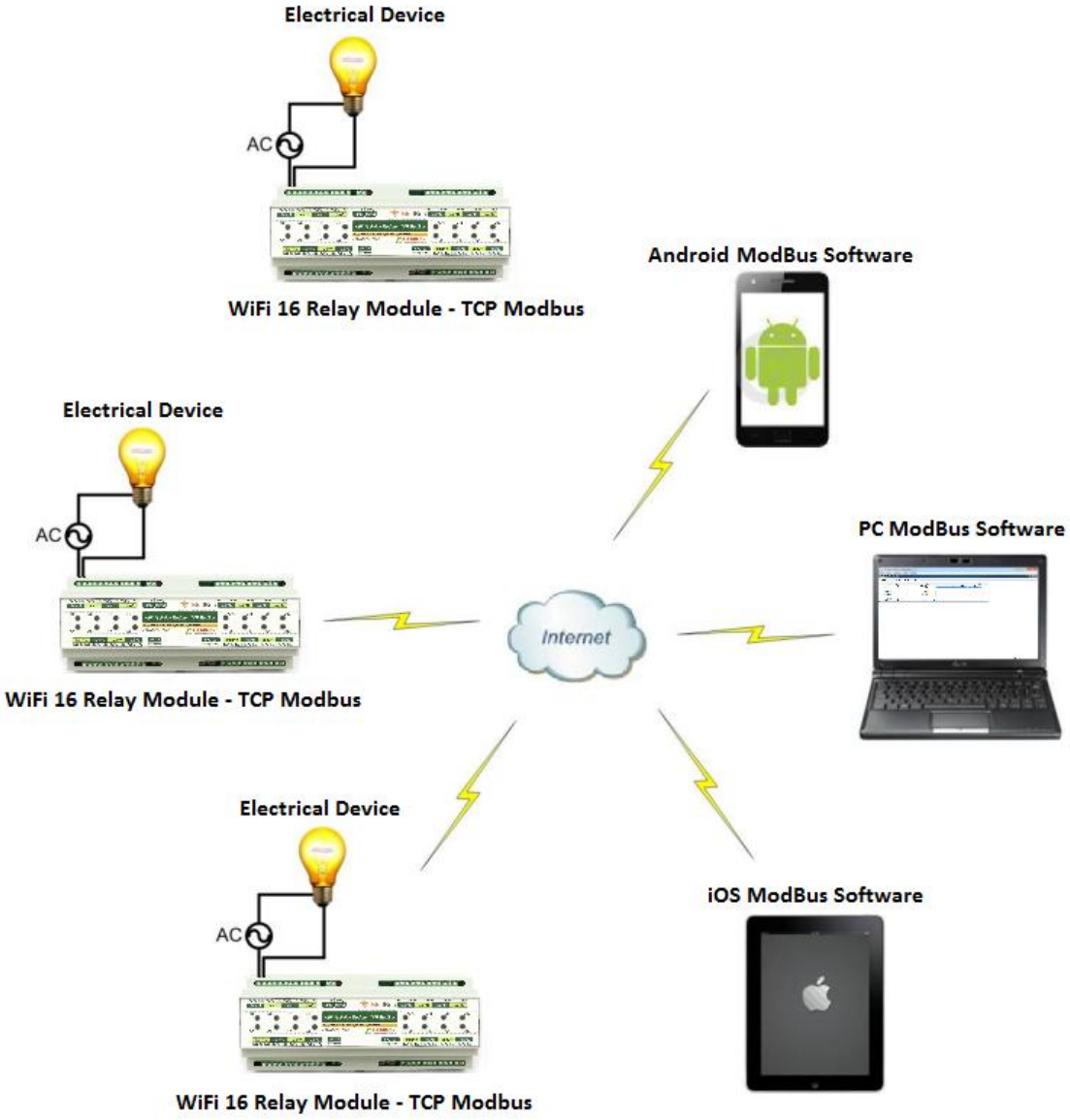


Figure 2. Access via Internet

3. Technical parameters

Table 1. Physical parameters

Parameter	Value
Box size, mm	210 x 85 x 58
PCB size, mm	203 x 82
Box weight, gr	420
PCB weight, gr	285
Operating temperature, °C	0 to 70

Table 2. System parameters

Parameter	Value
Power supply voltage,VDC	12 or 24 (depends on the model) ± 5%
Maximum current consumption at 12VDC (when all relays are ON), mA	600
Maximum current consumption at 24VDC (when all relays are ON), mA	400
Protection against reverse polarity	Yes
Default settings button	Yes
Reset button	Yes

Table 3. Relays

Parameter	Value
Relays JQC-3FC/T73 maximum switchable current / voltage	(7A / 250VAC, 10A / 125VAC, 12A / 120VAC, 10A / 28VDC)
Relays RAS-12-15 maximum switchable current / voltage	(10A / 250VAC, 15A / 120VAC, 15A / 24VDC)
Relays RAS-24-15 maximum switchable current / voltage	(10A / 250VAC, 15A / 120VAC, 15A / 24VDC)

Table 4. Network

Parameter	Value
DHCP	Yes
DNS	Yes
ICMP	Yes
Network parameters	IP/Mask/Default gateway
Web server for configuration/access	Yes
Control Relays via WEB server	**No**
TCP/IP socket support	Yes
UDP socket support	Yes

4. Connectors, ports and led indicators

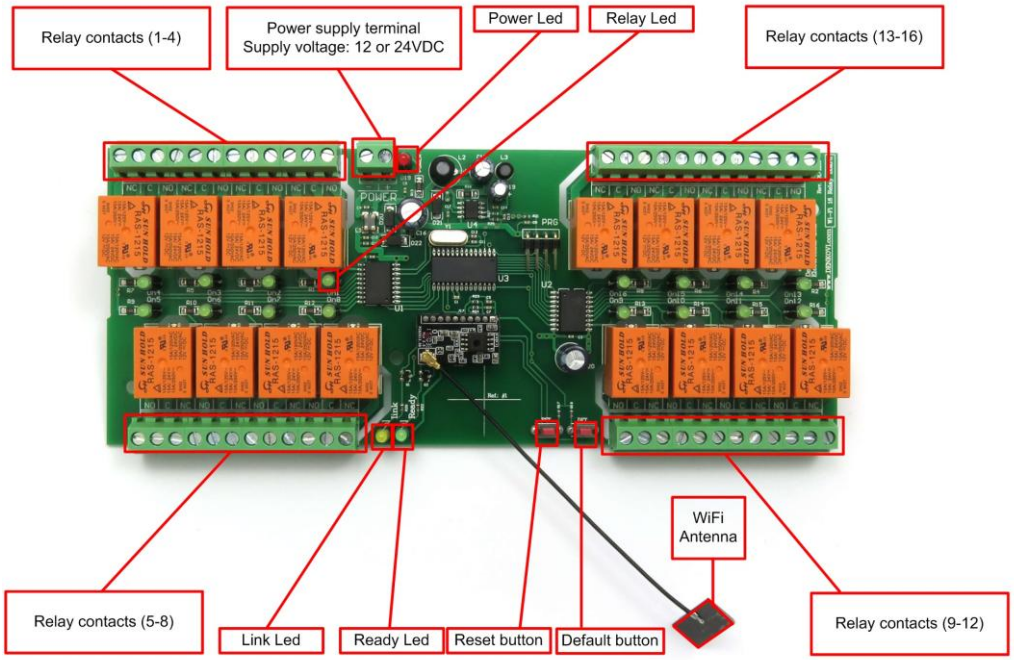


Figure 3. PCB Device overview

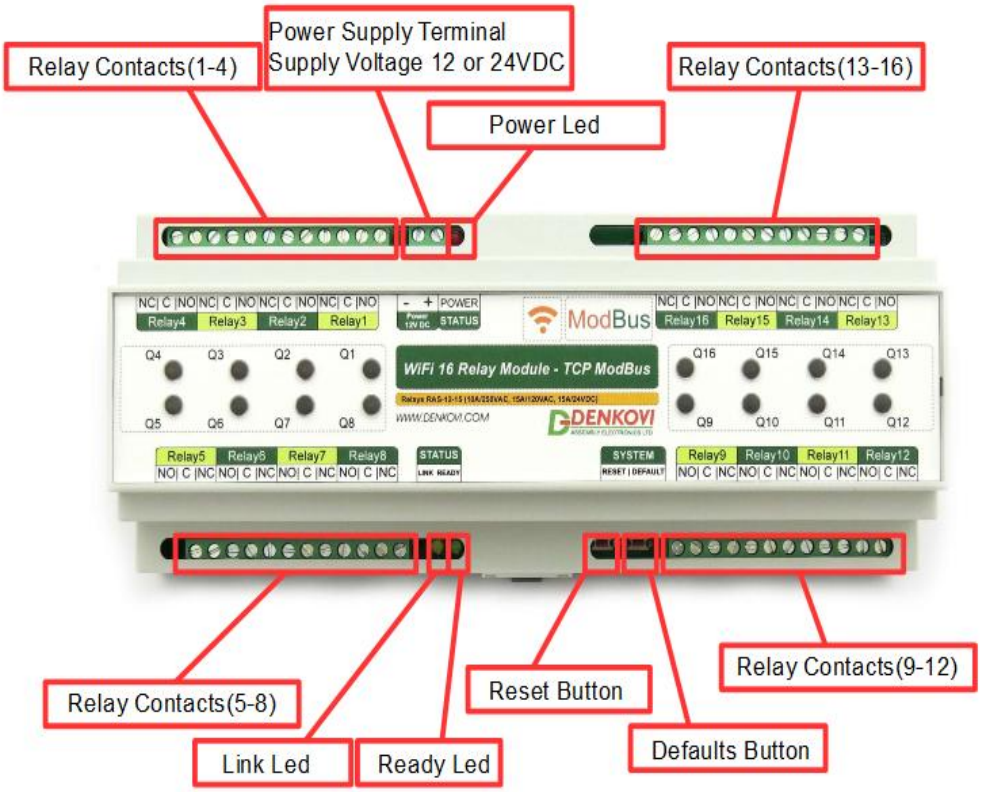


Figure 4. DIN Rail Device overview

5. Installation

- This device must be installed by qualified personnel;
- This device must not be installed directly outdoors;
- Installation consists of mounting the device, connecting to an IP network, connecting the relays, providing power and configuring via a web browser.

5.1. Box mounting

Figure 5. Mounting the device to DIN rail

WiFi 16 Relay Module – TCP ModBus (DIN BOX version) can be mounted to a standard (35mm by 7.55mm) DIN rail. Attach the module to the DIN rail by hooking the hook on the back of the enclosure to the DIN rail and then snap the bottom hook into place.

5.2. Power supply

Figure 6. Power supply

Depending on the selected model during purchase the power supply source for **WiFi 16 Relay Module – TCP ModBus** must be with voltage either **12VDC** or **24VDC** stabilized and filtered. After power on, the power led must be on and **ready led indicator** must on as well.

- Please keep the polarity and supply voltage range!
- **WiFi 16 Relay Module – TCP ModBus** does not accept AC supply voltage. It is highly recommended to check the power supply source parameters before supply the module.
- The power supply equipment shall be resistant to short circuit and overload in secondary circuit.
- When in use, do not place the equipment so that it is difficult to disconnect the device from the power supply.

5.3. Relay connection

The module carries 16 SPDT relays with parameters specified in the technical parameters section. Every relay channel has normally open (NO) and normally closed (NC) contacts connected directly to the terminals.

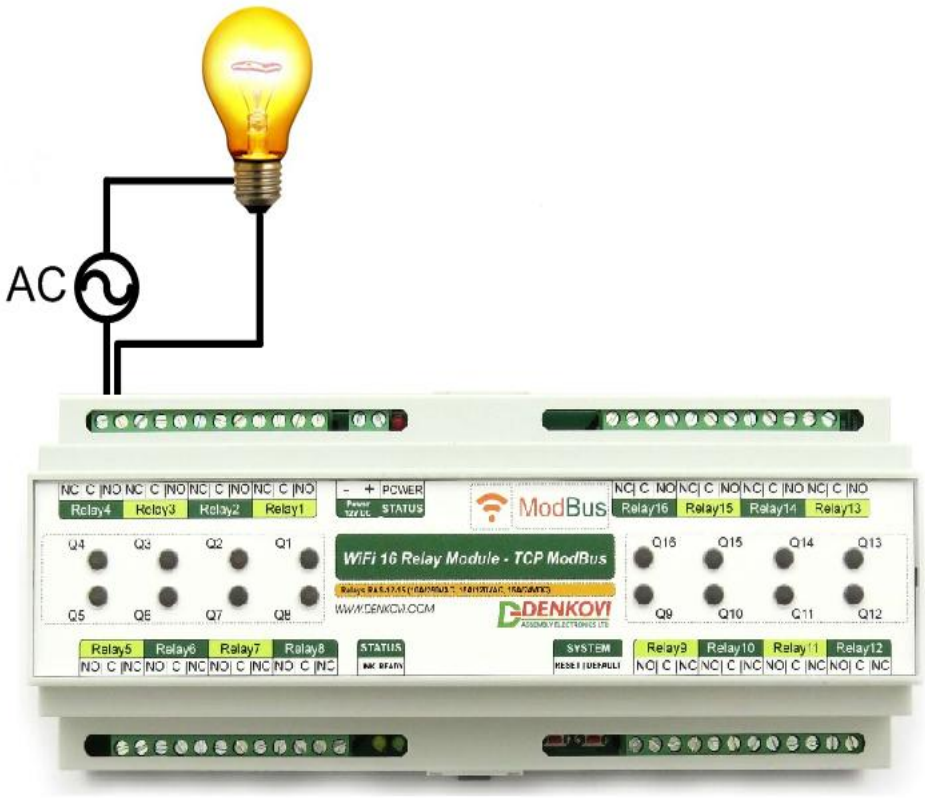


Figure 7. Connecting a lamp to relay

5.4. WiFi connection for very first time

The module is shipped with set Wi-Fi mode **AP (Access Point).** The default SSID is ***WiFi16_ModBus*** and there is not password. When the module is powered on and the **Ready Led** is ON as well, it should appear in your Wi-Fi scanned networks list.

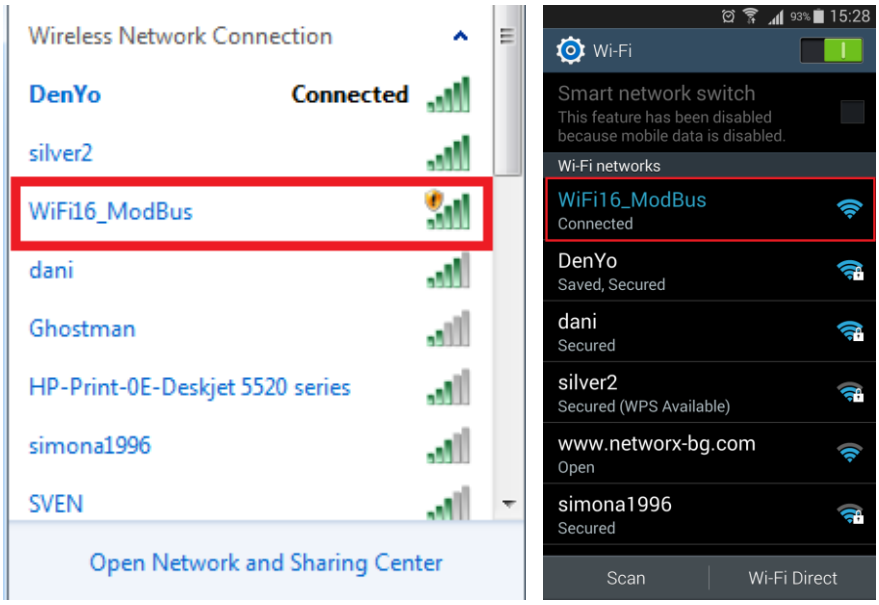


Figure 8. Default AP network name

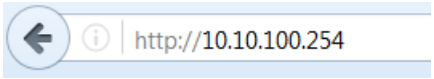
Next is to connect to this network.

Please note DHCP of your Wi-Fi network adapter must be turned ON in order to receive IP address from the board.

After successful connection, the **Link Led** is turned ON.

5.5. Set-up the module via web browser

Open web browser and type in address bar: **http://10.10.100.254**



Login to the web server, the default username/password is **admin/admin.** Now it should appear the welcome page (**Figure 9**).

Figure 9. Web server welcome page

Go to "Sockets" tab and be sure the following settings are entered (they are for the software communication):

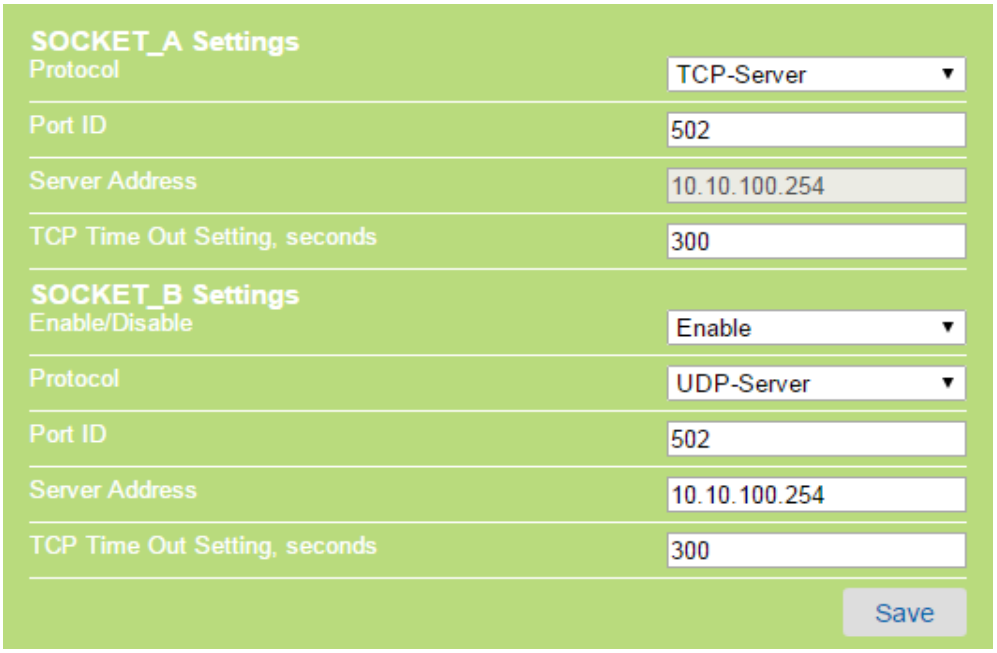


Figure 10. Sockets settings

A reboot is required in order the new settings take effect.

5.5. Control with Modbus Poll

There is a wide variety of free and low cost software in the Internet. For example Modbus Poll gives you an idea of how to communicate with WiFi 16 Relay Module – TCP ModBus.

You can download and install Modbus Poll Software from this link:
http://www.modbustools.com/download.html

Run the software, a screen on the **Figure 11** should appear.

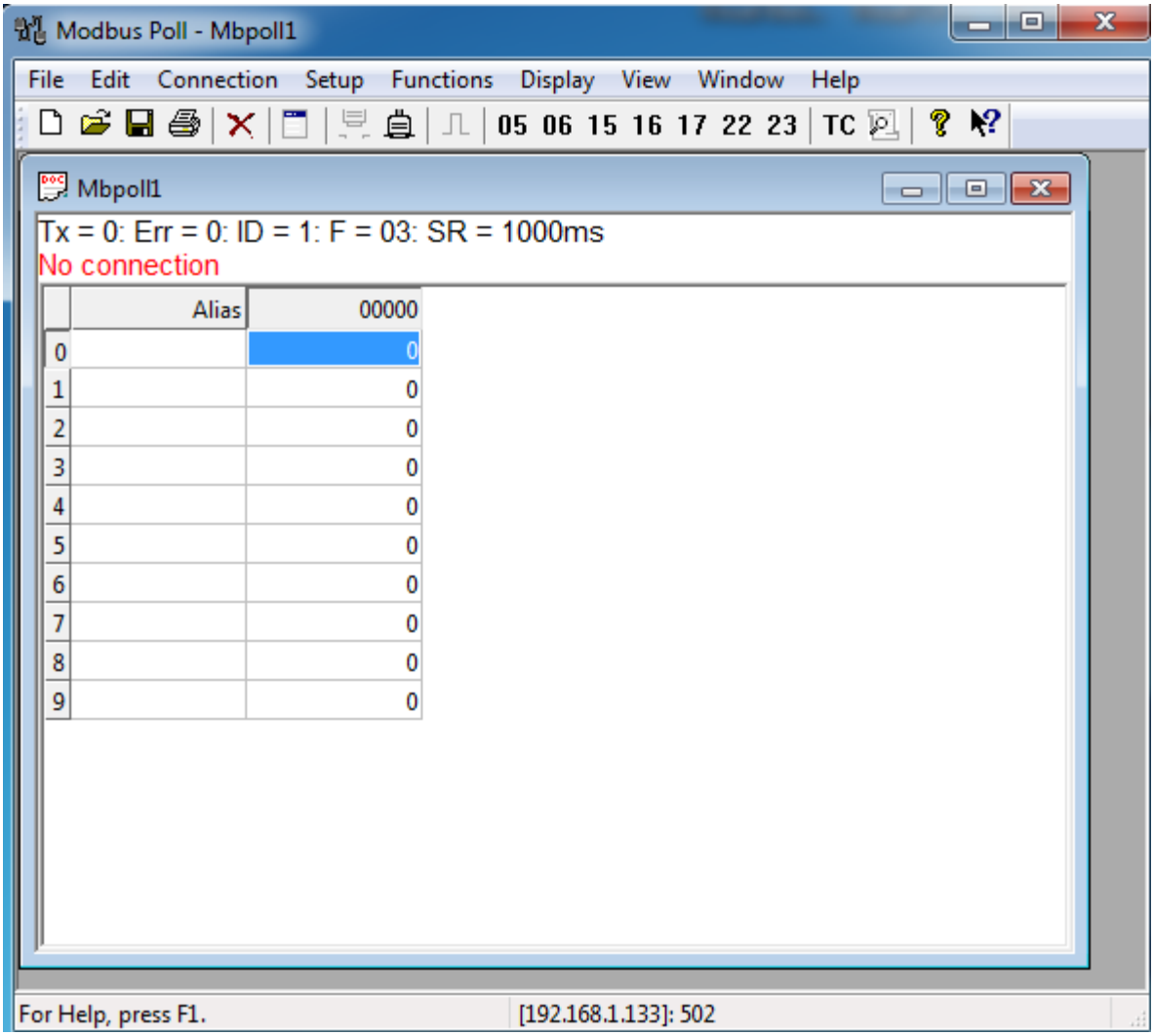


Figure 11. Modbus Poll Startup Screen

Go to Setup → Read/Write Definition and check the settings from **Figure 12**. Then press OK.

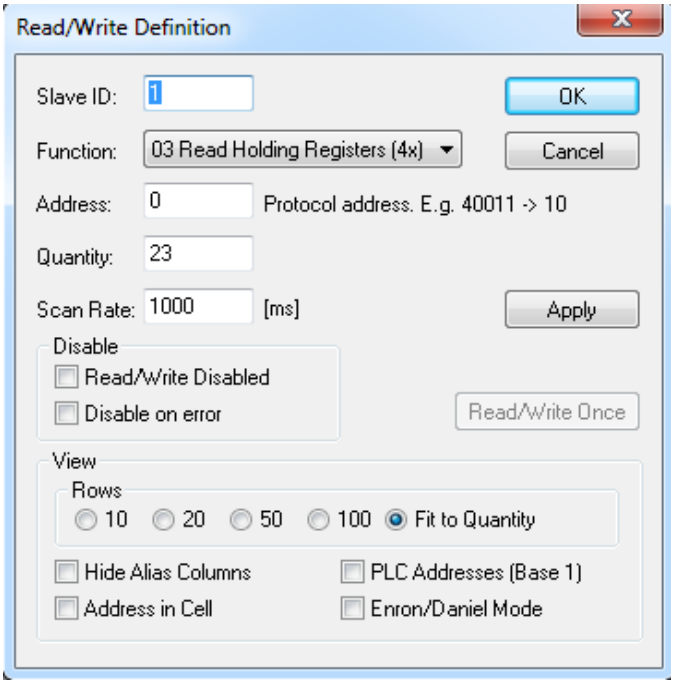


Figure 12. Modbus Poll Read/Write Definition

Next step is to create the connection. Go to Connection → Connect... **Figure 13** shows example of connection settings.

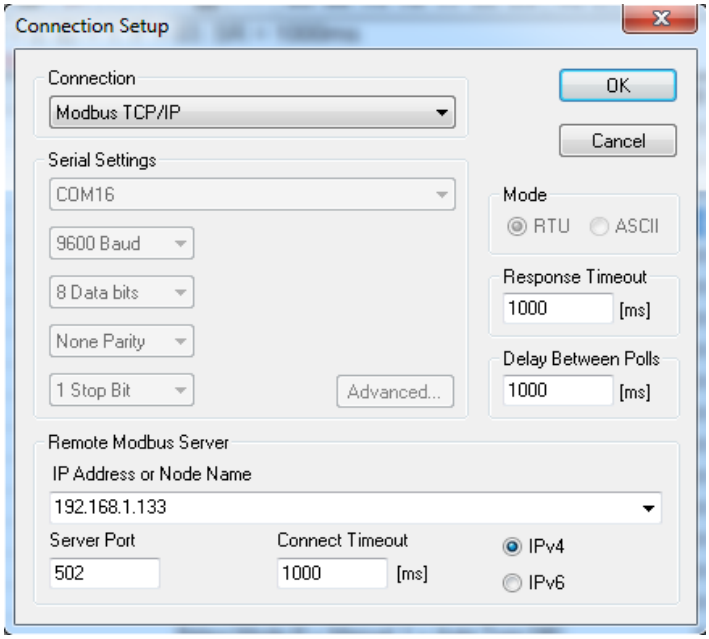


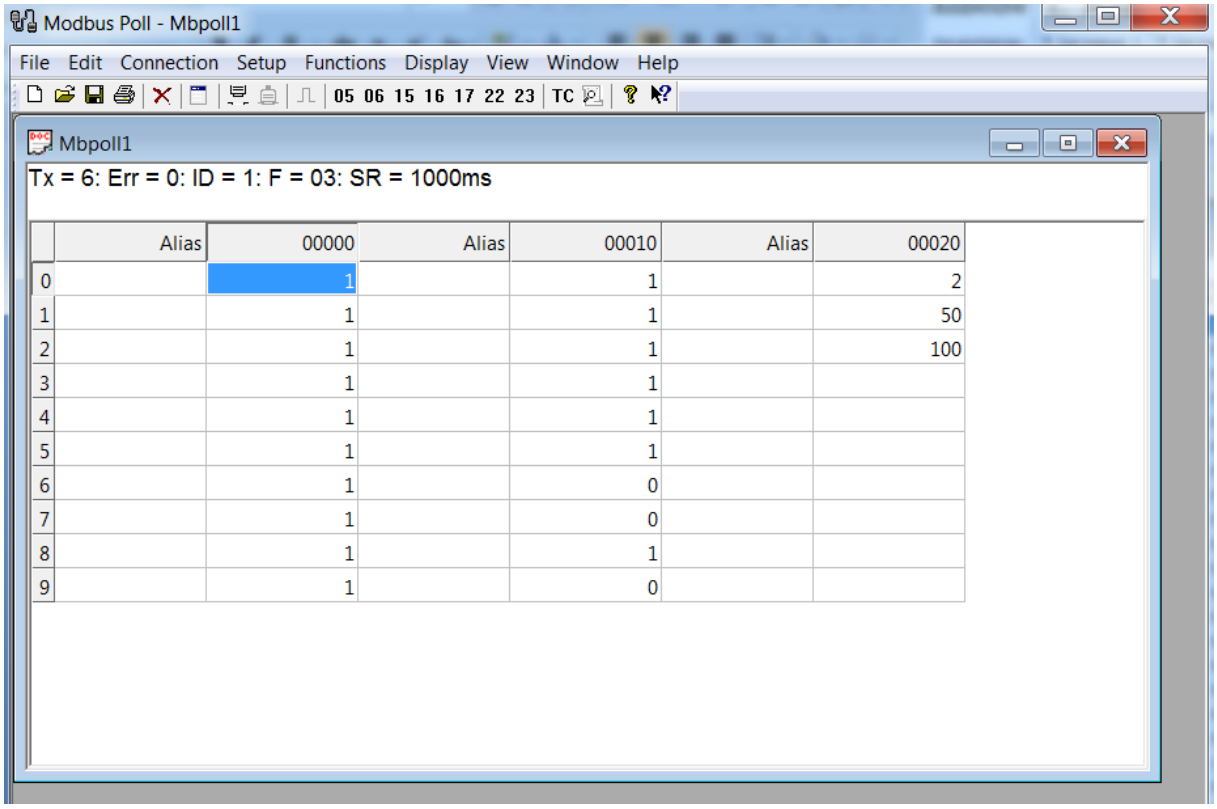
Figure 13. Modbus Poll Connection Setup

Please check the following settings:

- **IP Address** must be the **IP address** of the module;
- **Server Port** must be the **Port ID** from the module webserver -> Network -> SOCKET_A SETTINGS or SOCKET_B SETTINGS. By default **Port ID** is 502. We recommend using default port because almost all third-party software use it.

Please note Modbus Poll software will work with the relay module only if the **Protocol** parameter is "**TCP-Server**" from webserver -> Network -> SOCKET_A SETTINGS.

When all is done TX counter will start incrementing itself if successful connection is made and you should view Holding Registers values (Figure 14).



	Alias	00000	Alias	00010	Alias	00020
0		1		1		2
1		1		1		50
2		1		1		100
3		1		1		
4		1		1		
5		1		1		
6		1		0		
7		1		0		
8		1		1		
9		1		0		

Figure 14. Successfully connected screen

6. Default settings

6.1. List with factory settings

Table 5. Default (Factory) settings

Settings group	Parameter (according Web pages)	Value
WiFi Settings	WiFi Work Mode	AP (Access Point)
	SSID	WiFi16_ModBus
	Wireless AP Security Setting	Disable
Network Settings	DHCP	Enabled
	IP Address	10.10.100.254
	Gateway	10.10.100.254
	Subnet Mask	255.255.255.0
Communication Sockets	Socket_A Protocol	TCP-Server
	Port ID	502
	Server Address	10.10.100.254
	TCP Time Out Setting	300
	Socket_B	enable
	Protocol	UDP-Server
	Port ID	502
	Server Address	10.10.100.254
	TCP Time Out Setting	300
Web Server Settings	Server Port	80
	User name	admin
	Password	admin

6.2. Procedure for loading the factory settings

The default (factory) settings can be loaded if ***button Default*** (or ***DFT***) is pressed and hold for more than 3 seconds at any time.

7. Web access

To access the setup pages, run a web browser (Internet Explorer, Mozilla Firefox or similar), and enter the WiFi 16 Relay Module – TCP ModBus' IP address, for example: http://10.10.100.254

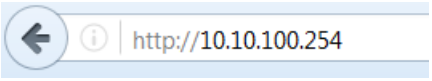


Figure 15. Open via browser

Note: You will need to have JavaScript enabled in your browser.

7.1. Home

The welcome page is shown below.

Figure 16. Home page

From this page it is possible to make fast scan of the Wi-Fi networks within the range and to connect the module to this network.

7.2. System

This page displays general information for the module.

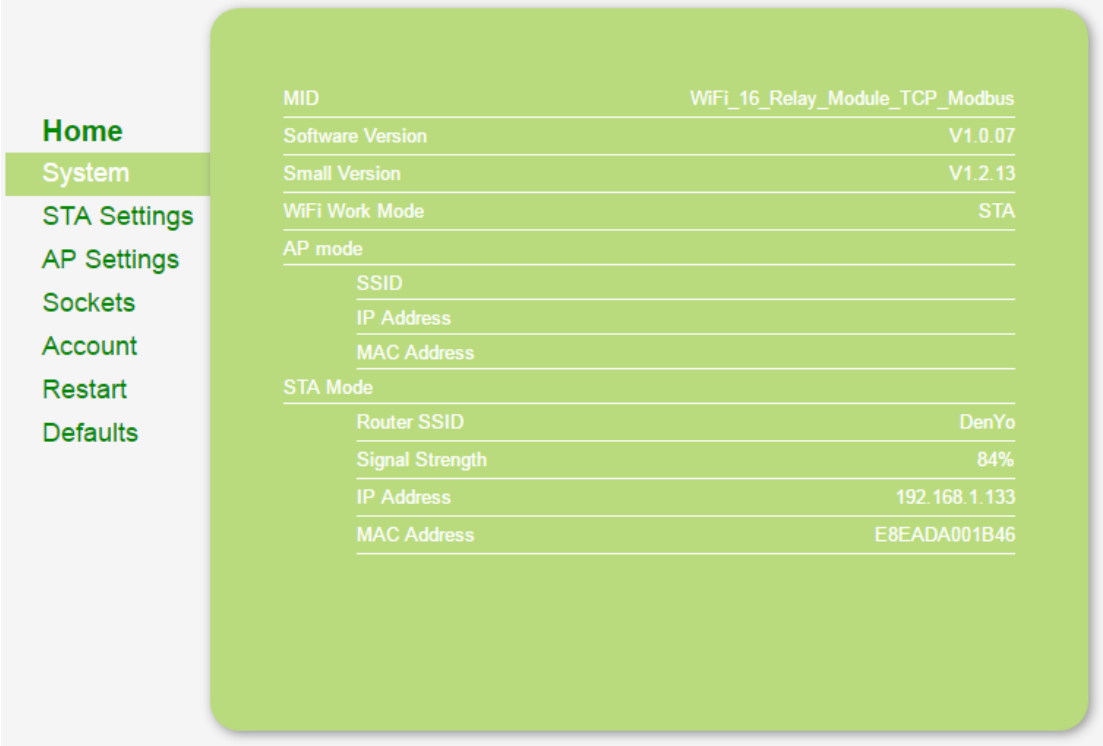


Figure 17. System page

- **MID:** Module ID, for this device it is WiFi_16_Relay_Module_TCP_Modbus;
- **Software Version:** The firmware version of the Wi-Fi interface;
- **Small Version:** The minor firmware version of the Wi-Fi interface;
- **Wi-Fi Work Mode:** The working mode of the Wi-Fi interface - it may be AP (Access Point), STA (Station) or APSTA (Access Point and Station at the same time);
- **AP Mode:** In this mode the current module creates network and acts like **access point**. if this mode is active it will display the bellow settings:
  - **SSID:** The SSID of the AP network;
  - **IP Address:** The IP of the AP network;
  - **MAC Address:** The MAC address of the AP network;
- **STA Mode:** In this mode the current module connects to another network. If this mode is active it will display the bellow settings:
  - **Router SSID:** The SSID of the network, to which is connected the current module (usually router);
  - **Signal Strength:** This signal strength of the network;
  - **IP Address:** The IP address of the network (usually this is the AP router IP address) to which is connected the current module;
  - **MAC Address:** Usually the MAC address of the router or AP;

7.3. STA Settings

These are the settings for STA mode. In this mode the relay module is connected to another Wi-Fi network - usually created by AP router.

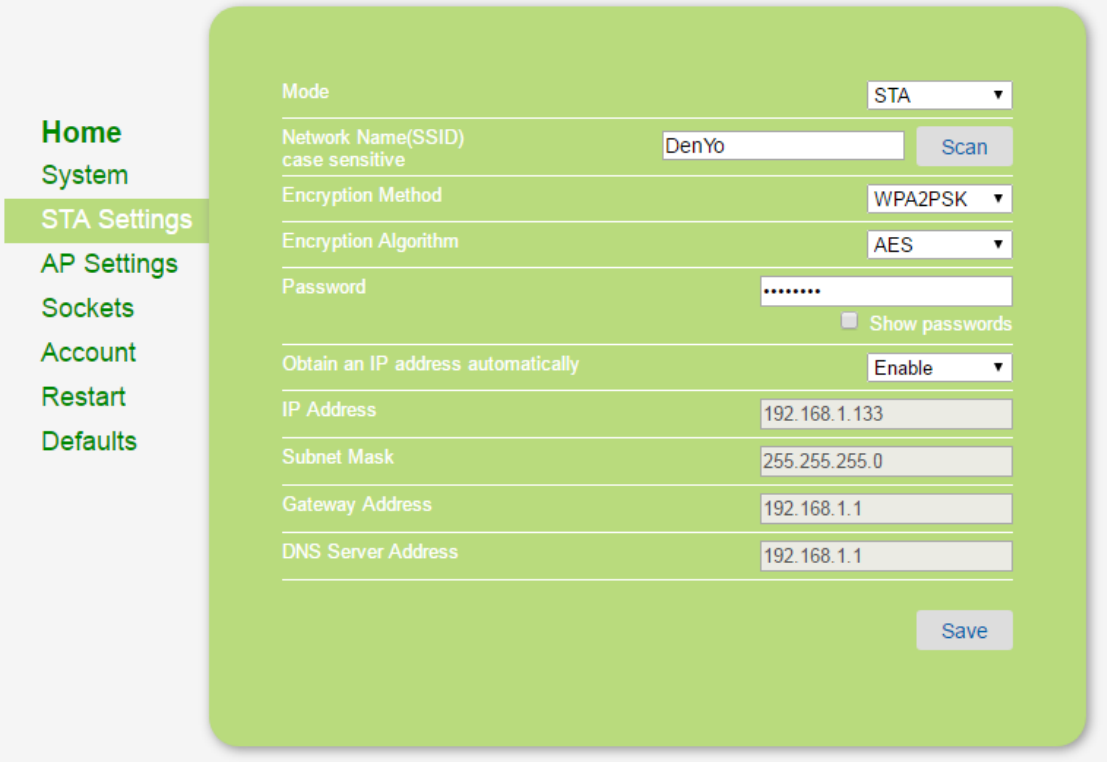


Figure 18. STA Settings page

- **Mode:** The working mode of the Wi-Fi interface. It can be:
  - **STA:** In this mode the board connects to another Wi-Fi network (usually created by AP router);
  - **AP+STA:** The above mode plus additional Wi-Fi network created by the module. Here it is AP and client at the same time.
  - **Network Name (SSID):** The name (SSID) of the network to which must be connected the module;
- **Scan button:** Starts scanning the available Wi-Fi networks within the range;
- **Encryption Method:** The encryption method for the Wi-Fi network (OPENWEP, SHAREDWEP, WPAPSK, WPA2PSK);
- **Encryption Algorithm:** AES or TKIP;
- **Encryption Type:** HEX or ASCII;
- **Password:** password for the network. When method is OPENWEP pr SHAREDWEP length must be 10 or 26 symbols (for HEX encryption type) and 5 or 13 symbols (for ASCII encryption type). For rest of cases when password is required it may be maximum 64 symbols and minimum 8 symbols;
- **Obtain an IP address automatically:** Enable or disable DHCP. When disabled, the bellow settings will be active to be set;
- **IP Address:** The IP address of the module;

- **Subnet Mask**: The mask;
- **Gateway Address:** The gateway IP address (usually the address of the AP or router);
- **DNS Server Address**: The DNS IP address (usually the address of the AP or router);
- **Save button:** Once you have changed the settings as required, click this button.

When the ***Scan button*** is pressed, the scanning will begin and the bellow screen will appear asking for select the Wi-Fi network to connect to.

Figure 19. List with available Wi-Fi networks within the range

You have to reboot the device for these settings to apply.

7.4. AP Settings

These are the settings for AP (Access Point) mode.

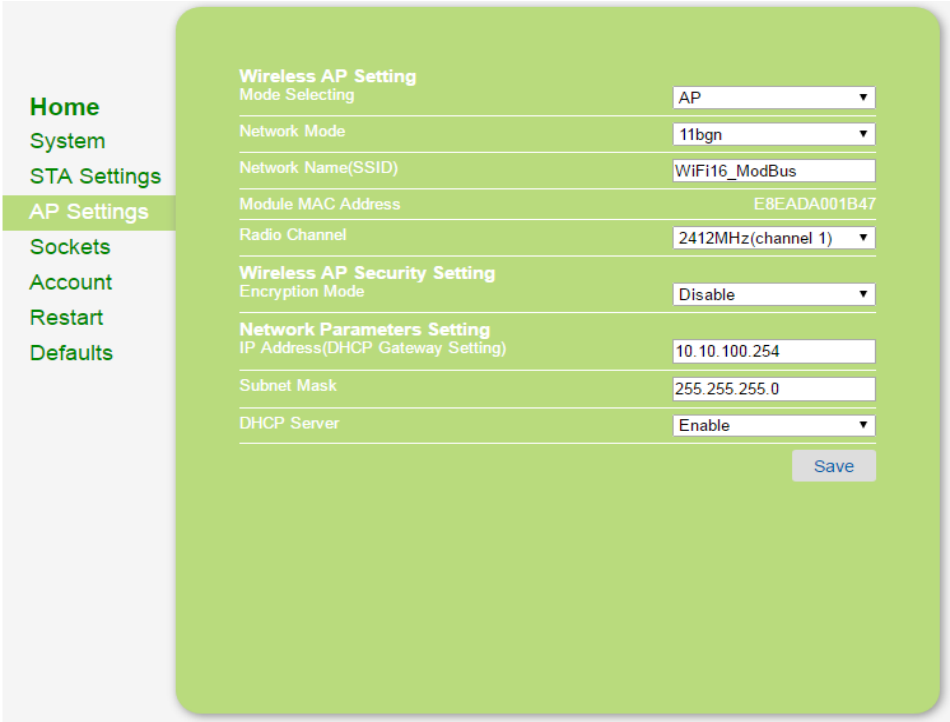


Figure 20. AP Settings

- **Mode:** The working mode of the Wi-Fi interface. It can be:
  - **AP:** In this mode the board works as Access Point only;
  - **AP+STA:** Access Point and station at the same time;
  - **Network Name (SSID):** The name of the Wi-Fi network which will be created by the module;
- **Module MAC Address:** The MAC address of the AP interface;
- **Radio Channel:** Select the radio channel for the Wi-Fi network;
- **Wireless AP Security Setting:** It may be either disabled either WPA2-PSK;
- **WPA Encryption:** If security setting is WPA2-PSK, it may be TKIP, AES or TKIPAES;
- **Password:** If security setting is WPA2-PSK, this is the AP Wi-Fi network. May be max 20 characters;
- **IP Address (DHCP Gateway Setting):** The IP address of the module in this mode;
- **Subnet Mask:** The network mask;
- **DHCP Server:** Enable or disable the DHCP setting of the module in AP mode;
- **Save button:** Once you have changed the settings as required, click this button.

You have to reboot the device for these settings to apply.

7.5. Sockets

From the bellow page it is possible to be adjusted the communication sockets parameters. The sockets are used for communication between software (computer, mobile apps) and the relay module.

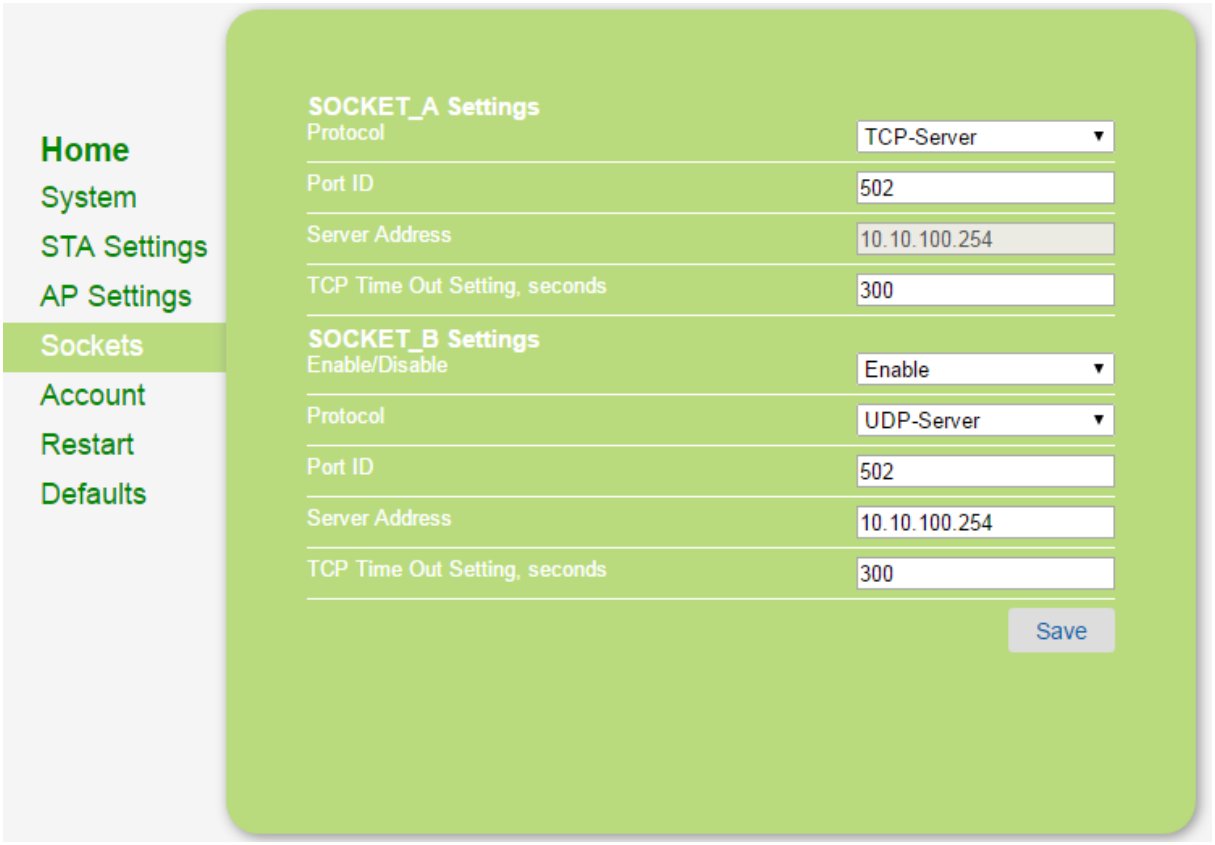


Figure 21. Sockets page

7.5.1. SOCKET_A Settings

- **Protocol:** This parameter may be:
  - **TCP-Server:** In this mode the socket works like TCP/IP server;
  - **TCP-Client:** In this mode the socket works like TCP/IP client;
  - **UDP-Server:** In this mode the socket works like UDP server;
  - **UDP-Client:** In this mode the socket works like UDP client;
- **Port ID:** The port number of the socket (1...65535);
- **Server Address:** Valid only if socket is server;
- **TCP Time Out Setting:** timeout (in seconds), only if TCP mode is selected (0...600 sec);

7.5.2. SOCKET_B Settings

- **Enable/Disable:** Enable or disable the second socket;
- **Protocol:** This parameter may be:
  - **TCP-Client:** In this mode the socket works like TCP/IP client;
  - **UDP-Server:** In this mode the socket works like UDP server;
  - **UDP-Client:** In this mode the socket works like UDP client;
- **Port ID:** The port number of the socket (1...65535);
- **Server Address:** Valid only if socket is server;
- **TCP Time Out Setting:** timeout (in seconds), only if TCP mode is selected (0...600 sec);

You have to reboot the device for these settings to apply.

7.5.3. Sockets Mechanism

Socket A supports four modes: TCP Server, TCP Client, UDP Client and UDP Server. Because of acting as a Modbus slave device sockets should be set up as TCP Server or UDP Server.
When Socket A is configured as TCP Server, it supports "Multi-TCP link" connection and maximum 5 TCP clients are permitted to connect.
"Multi-TCP link" connection works in this way:

- All data from different TCP connections or clients will be transmitted to the module as a sequence.
- The answer from the module will be duplicated and broadcasted to every TCP connection or client (it doesn't matter if it is Socket A or Socket B).

Socket B supports three modes: TCP Client, UDP Client and UDP Server.

7.6. Account

Set the username and password for web server access.

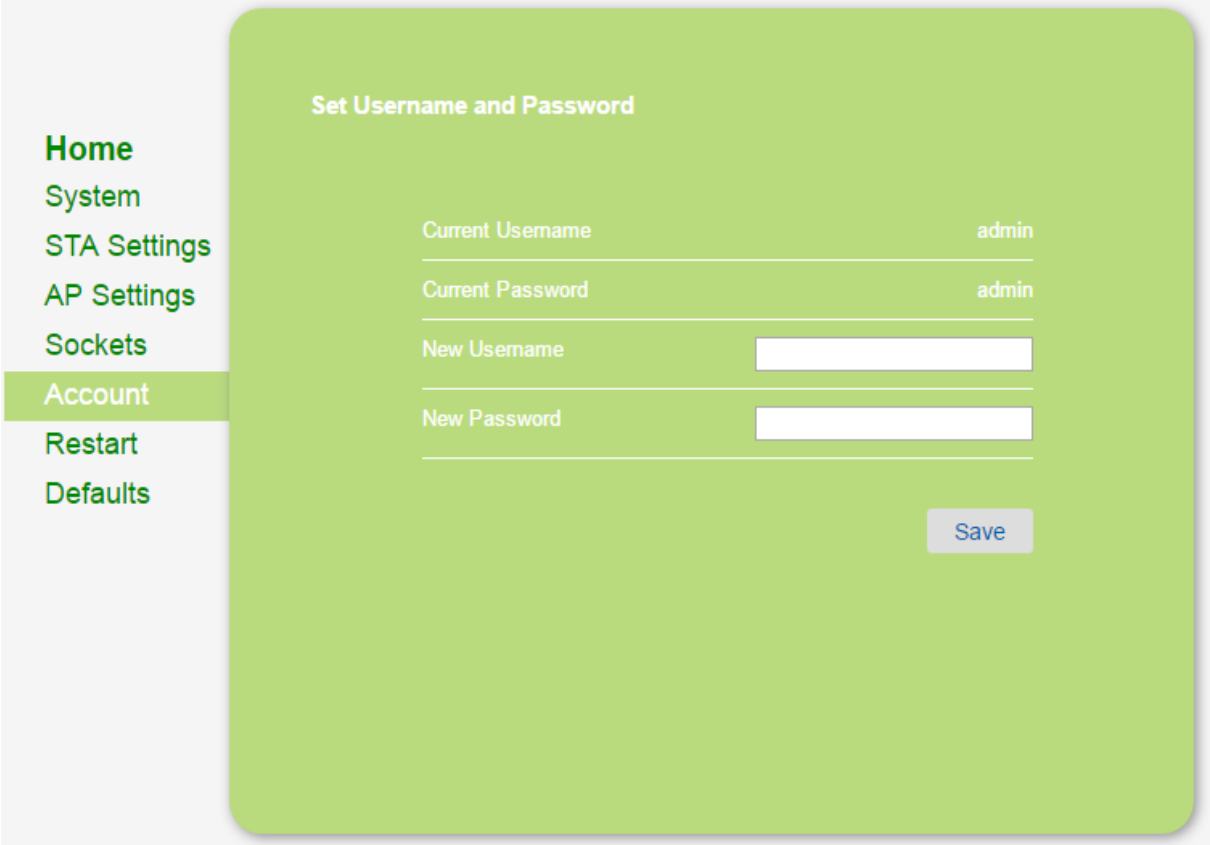


Figure 22. Account page

The maximum length for the username and password is 20 characters and the minimum is 1 character.

You have to reboot the device for these settings to apply.

7.7. Restart

From this page it is possible to restart the module (only the Wi-Fi interface).

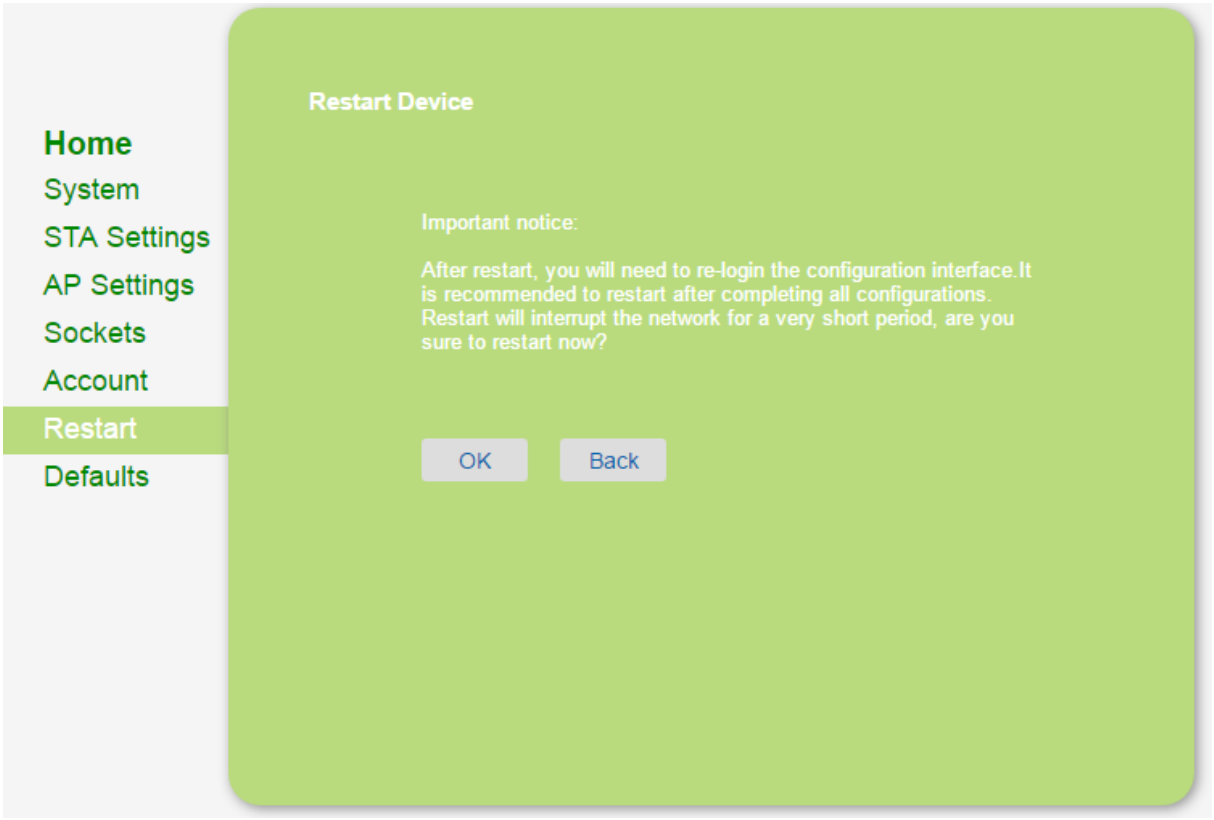


Figure 23. Restart page

7.8. Defaults

Loads the default (factory) settings.

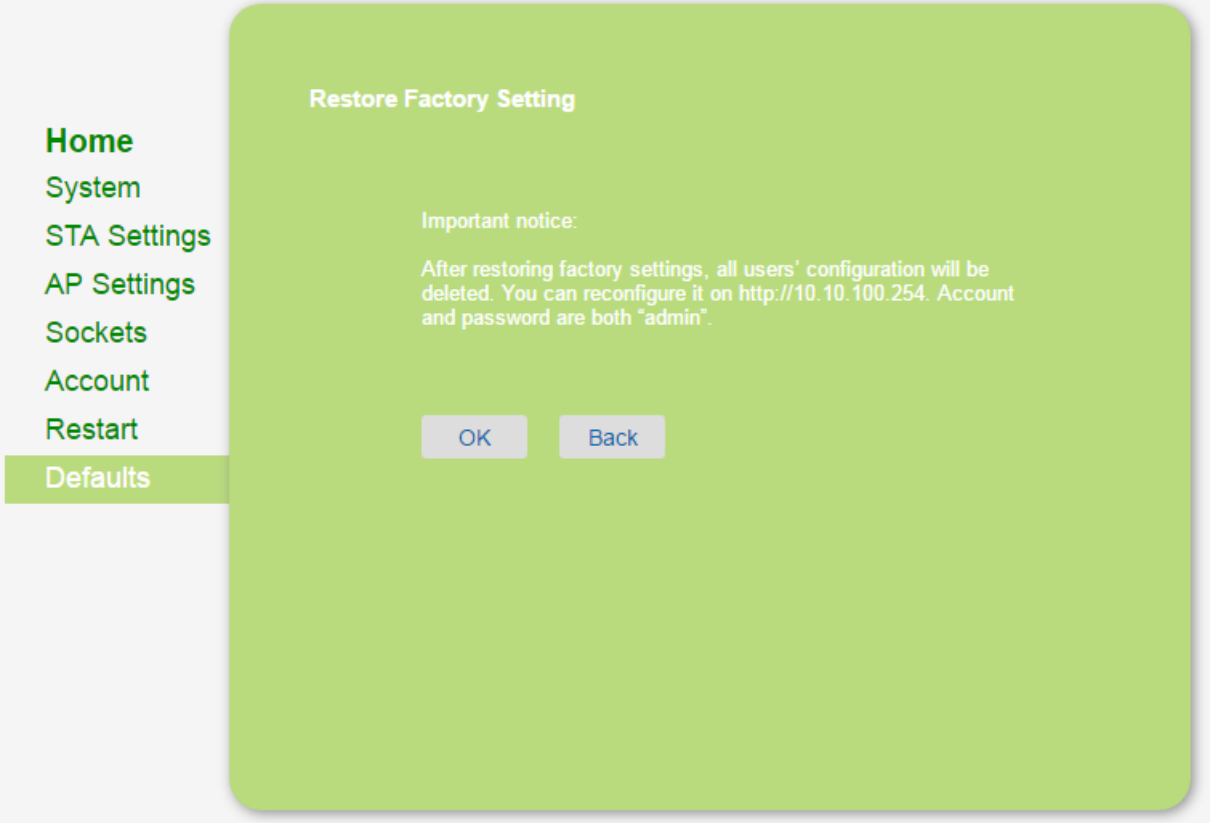


Figure 24. Defaults page

8. ModBus TCP

The **Wi-Fi 16 Relay Module – TCP ModBus** supports TCP Modbus protocol for setting/getting the relays status. The communication may be used over TCP/IP or UDP sockets. All the relays are represented as coils so there are 16 coils supported by the device (Coil 0 to Coil 15). There are 23 holding registers for additional settings (Holding Register 0 to Holding Register 22). TCP Modbus Protocol frame is illustrated on **Figure 24**.

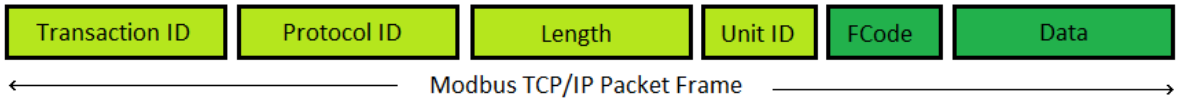


Figure 25. Modbus TCP/IP Packet Frame

- **Transaction ID** is 2 byte ID starting from 0x0000 which is incremented on every request;
- **Protocol ID** is always 0x0000;
- **Length** is message length i.e. it tells how many bytes are following this byte;
- **Unit ID** is Slave ID;
- **FCode** is the function code (several function codes will be described);
- **Data** is the appropriate data for the used function code;

Modbus frame always starts with Transaction ID, Protocol ID, Length and Unit ID. Successful respond should consist of same Transaction ID. Protocol ID should be always 0x0000, which stays for TCP/IP Modbus. Message length will vary depends on Function Code and Data.

Unit ID by default is 1, but can be changed.

More about Modbus protocol could be found at https://en.wikipedia.org/wiki/Modbus and https://www.prosoft-technology.com/kb/assets/intro_modbustcp.pdf.

8.1. Coils commands

Coils are mapped to Relays. Reading and Writing Coils reads/writes relays status. There are 16 coils for reading and writing – Coil0 to Coil15 which refer to Relay1 to Relay16.

8.1.1. Read Coils (0x01)

Read Coils (0x01) command will return current relays status. Example for reading all relays status:

Table 6. Read coils command format

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12
Transaction ID		Protocol ID		Length		Unit ID	FC	Data			
0x00	0x00	0x00	0x00	0x00	0x06	0x01	0x01	0x00	0x00	0x00	0x10

- Byte9 and Byte10 represents the data address of first coil to read (data address is 0x00 in our example);
- Byte11 and Byte12 represents the total number of coils requested (total number of coils is 0x10 in our example);

Please note that all bytes are in hexadecimal format!

Table 7. Read coils expected answer

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11
Transaction ID		Protocol ID		Length		Unit ID	FC	Data		
0x00	0x00	0x00	0x00	0x00	0x05	0x01	0x01	0x02	0x05	0xA0

- Byte9 indicates how many bytes follow this byte (in our example is 2);
- Byte10 represents coils 7 - 0 (0x05 = 0000 0101) ;
- Byte11 represents coils 15 – 8 (0xA0 = 1010 0000);

This example illustrate the following relays status – Relay1, Relay3, Relay14 and Relay 16 are ON. All other relays are OFF.

8.1.2. Write Single Coil (0x05)

Write Single Coil (0x05) command will set relay ON or OFF. Example of how to set Relay11 ON is given in table below:

Table 8. Write single coil command format

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12
Transaction ID		Protocol ID		Length		Unit ID	FC	Data			
0x00	0x01	0x00	0x00	0x00	0x06	0x01	0x05	0x00	0x0A	0xFF	0x00

- Byte9 and Byte10 represents the data address of the coil to set ON/OFF (data address is 10 in our example which is Relay11);
- Byte11 and Byte12 represents the state to set (in our example we set relay ON);

Note: If you want to set a relay OFF, Byte11 and Byte12 should be 0x00.

Table 9. Write single coil expected answer

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12
Transaction ID		Protocol ID		Length		Unit ID	FC	Data			
0x00	0x01	0x00	0x00	0x00	0x06	0x01	0x05	0x00	0x0A	0xFF	0x00

- Byte9 and Byte10 represents the data address of the coil that was set;
- Byte11 and Byte12 represents the state that was set;

8.1.3. Write Multiple Coils (0x0F)

Write Multiple Coils (0x0F) command will set multiple sequential relays ON or OFF. Example of how to set Relay1 to Relay10 with single command is given in table below.

Table 10. Write multiple coils command format

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte 10	Byte 11	Byte 12	Byt e13	Byt e14	Byte 15
Transaction ID		Protocol ID		Length		Unit ID	FC	Data						
0x00	0x02	0x00	0x00	0x00	0x09	0x01	0x0F	0x00	0x00	0x00	0x0 A	0x0 2	0x5 5	0x01

- Byte9 and Byte10 represents the data address of the first coil to set ON/OFF (data address is 0 in our example which is Relay1);
- Byte11 and Byte12 represents the number of coils to set (in our example we 0x000A);
- Byte13 represents how many bytes follow this byte (in our example is 0x02);
- Byte14 represents coils 7 – 0 (0x55 = 0101 0101);
- Byte15 represents 6 leading zeroes and coils 9 – 8 (0x01 = 0000 0001);

Table 11. Write multiple coils expected answer

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12
Transaction ID		Protocol ID		Length		Unit ID	FC	Data			
0x00	0x02	0x00	0x00	0x00	0x06	0x01	0x0F	0x00	0x00	0x00	0x0A

- Byte9 and Byte10 represents the data address of the first coil that was set;
- Byte11 and Byte12 represents the number of coils that was set;

This example illustrate the following relays status – Relay1, Relay3, Relay5, Relay7 and Relay9 will be turned ON. Relay2, Relay4, Relay6, Relay8 and Relay10 will be turned OFF.

8.2. Holding Registers commands

There are 23 readable and 22 writable Holding Registers. The table below describes each Holding Register's function. Each Holding Register is 2 byte long. Each given numbers in the table below are in decimal unsigned type.

Table 12. Supported holding registers

Holding Register's Address	Description
0	Relay 1 Auto Off Time Units (1 ÷ 65535)
1	Relay 2 Auto Off Time Units (1 ÷ 65535)
2	Relay 3 Auto Off Time Units (1 ÷ 65535)
3	Relay 4 Auto Off Time Units (1 ÷ 65535)
4	Relay 5 Auto Off Time Units (1 ÷ 65535)
5	Relay 6 Auto Off Time Units (1 ÷ 65535)
6	Relay 7 Auto Off Time Units (1 ÷ 65535)
7	Relay 8 Auto Off Time Units (1 ÷ 65535)
8	Relay 9 Auto Off Time Units (1 ÷ 65535)
9	Relay 10 Auto Off Time Units (1 ÷ 65535)
10	Relay 11 Auto Off Time Units (1 ÷ 65535)
11	Relay 12 Auto Off Time Units (1 ÷ 65535)
12	Relay 13 Auto Off Time Units (1 ÷ 65535)
13	Relay 14 Auto Off Time Units (1 ÷ 65535)
14	Relay 15 Auto Off Time Units (1 ÷ 65535)
15	Relay 16 Auto Off Time Units (1 ÷ 65535)
16	Relays Mode Register (0 is Manual, 1 is Auto Off)*
17	Timer Units Register (0 = 100ms/unit, 1 = 1s/unit)*
18	Slave ID Register (1 ÷ 247)**
19	Relays State Register*
20	Relays Startup Mode Register (0 will turn off all relays, 1 will set relays in states written to Relays State Register, 2 will set relays in state from last change)
21	Reception Timeout Register – timeout for waiting the whole packet frame (100ms/unit)
22	Version Register*** – Value of this register divided by 100 gives the version of the module

**Bitwise settable register – MSB is Relay16 and LSB is Relay1.*
***Setting Slave ID out of this range (1 ÷ 247) will be accepted but the device will take only the least significant byte as ID.*
****Read only register.*

8.2.1. Write Single Holding Register (0x06)

Write Single Holding Register (0x06) command will set single Holding Register. Example of how to set Relay1 Auto Off Time Units with single command is given in table below.

Table 13. Write single holding register command format

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12
Transaction ID		Protocol ID		Length		Unit ID	FC	Data			
0x00	0x03	0x00	0x00	0x00	0x06	0x01	0x06	0x00	0x00	0x00	0x64

- Byte9 and Byte10 represents the data address of the Holding Register (data address is 0x0000 in our example which is Relay 1 Auto Off Time Units' Register);
- Byte11 and Byte12 represents the value of the Holding Register (value is 0x0064 units in our example);

Table 14. Write single holding register expected answer

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12
Transaction ID		Protocol ID		Length		Unit ID	FC	Data			
0x00	0x03	0x00	0x00	0x00	0x06	0x01	0x06	0x00	0x00	0x00	0x64

- Byte9 and Byte10 represents the data address of the Holding Register that was set;
- Byte11 and Byte12 represents the value of the Holding Register that was set;

8.2.2. Read Multiple Holding Registers (0x03)

Read Multiple Holding Registers (0x03) command will read multiple sequential Holding Registers. Example of how to read 20 Holding Registers starting from data address 0x00 is given in table below:

Table 15. Read holding register command format

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12
Transaction ID		Protocol ID		Length		Unit ID	FC	Data			
0x00	0x04	0x00	0x00	0x00	0x06	0x01	0x03	0x00	0x00	0x00	0x17

- Byte9 and Byte10 represents the data address of the First Holding Register to read(data address is 0x0000 in our example which is Relay 1 Auto Off Time Units' Register);
- Byte11 and Byte12 represents the total number of Holding Registers to read (value is 0x0017 in our example which means all registers should be read);

Table 16. Read holding register expected answer

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12
Transaction ID		Protocol ID		Length		Unit ID	FC	Data			
0x00	0x04	0x00	0x00	0x00	0x31	0x01	0x03	0x2E	0x00	0x64	0x00
Byte13	Byte14	Byte15	Byte16	Byte17	Byte18	Byte19	Byte20	Byte21	Byte22	Byte23	Byte24
Data											
0x01	0x00	0x01	0x00	0x01	0x00	0x01	0x00	0x01	0x00	0x01	0x00
Byte25	Byte26	Byte27	Byte28	Byte29	Byte30	Byte31	Byte32	Byte33	Byte34	Byte35	Byte36
Data											
0x01	0x00	0x01	0x00	0x01	0x00	0x00	0x00	0x00	0x00	0x00	0x00
Byte37	Byte38	Byte39	Byte40	Byte41	Byte42	Byte43	Byte44	Byte45	Byte46	Byte47	Byte48
Data											
0x00	0x00	0x00	0x00	0x00	0x00	0x53	0x80	0x21	0x00	0x01	0x03
Byte49	Byte50	Byte51	Byte52	Byte53	Byte54	Byte55					
Data											
0x58	0x00	0x00	0x00	0x01	0x00	0x64					

- Byte9 represents the bytes following this byte (bytes following this bytes are 46 in our example – 23 Holding Registers x 2 bytes each);
- Byte10 and Byte11 represents Relay 1 Auto Off Time Units(0x0064 units in our example);
- Byte12 and Byte13 represents Relay 2 Auto Off Time Units (0x0001 units in our example);
- Byte14 and Byte15 represents Relay 3 Auto Off Time Units (0x0001 units in our example);
- Byte16 and Byte17 represents Relay 4 Auto Off Time Units (0x0001 units in our example);
- Byte18 and Byte19 represents Relay 5 Auto Off Time Units (0x0001 units in our example);
- Byte20 and Byte21 represents Relay 6 Auto Off Time Units (0x0001 units in our example);
- Byte22 and Byte23 represents Relay 7 Auto Off Time Units (0x0001 units in our example);
- Byte24 and Byte25 represents Relay 8 Auto Off Time Units (0x0001 units in our example);
- Byte26 and Byte27 represents Relay 9 Auto Off Time Units (0x0001 units in our example);
- Byte28 and Byte29 represents Relay 10 Auto Off Time Units (0x0001 units in our example);
- Byte30 and Byte31 represents Relay 11 Auto Off Time Units (0x0001 units in our example);

- Byte32 and Byte33 represents Relay 12 Auto Off Time Units (0x0001 units in our example);
- Byte34 and Byte35 represents Relay 13 Auto Off Time Units (0x0001 units in our example);
- Byte36 and Byte37 represents Relay 14 Auto Off Time Units (0x0001 units in our example);
- Byte38 and Byte39 represents Relay 15 Auto Off Time Units (0x0001 units in our example);
- Byte40 and Byte41 represents Relay 16 Auto Off Time Units (0x0001 units in our example);
- Byte42 and Byte43 represents Relays Mode Register (0x0053 in our example which is 0000 0000 0101 0011 – Relay1, Relay2, Relay5 and Relay7 are in Auto Off Mode and all others are in Manual Mode);
- Byte44 and Byte45 represents Timer Units Register (0x8021 in our example which is 1000 0000 0010 0001 – Relay1, Relay6 and Relay 16 units are 1s/unit and all others are 100ms/unit);
- Byte46 and Byte47 represents Slave ID Register (Slave ID is 0x0001 in our example);
- Byte48 and Byte49 represents Relays State Register (0x0358 in our example which is 0000 0011 0101 1000 – Relay4, Relay5, Relay7, Relay9 and Relay10 will be turned on all other will be turned off (if Relays Startup Mode Register is 0x01 or 0x02));
- Byte50 and Byte51 represents Relays Startup Mode Register (All relays will be turned off on startup in our example);
- Byte52 and Byte53 represents Reception Timeout Register (0x0001 in our example which means 100ms);
- Byte54 and Byte55 represents Version Register (0x0064 (100 decimal) in our example which means v1.00);

8.2.3. Write Multiple Holding Registers (0x10)

Write Multiple Holding Registers (0x10) command will write multiple sequential Holding Registers. Example of how to write 5 Holding Registers starting from data address 1 is given in table below:

Table 17. Write multiple holding registers command format

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12
Transaction ID		Protocol ID		Length		Unit ID	FC	Data			
0x00	0x05	0x00	0x00	0x00	0x11	0x01	0x10	0x00	0x01	0x00	0x05
Byte13	Byte14	Byte15	Byte16	Byte17	Byte18	Byte19	Byte20	Byte21	Byte22	Byte23	
Data											
0x0A	0x00	0xC8	0x01	0x2C	0x01	0x90	0x01	0xF4	0x02	0x58	

- Byte9 and Byte10 represents the data address of the first Holding Register (0x0001 in our example);

- Byte11 and Byte12 represents the total number of Holding Registers to be written (5 in our example);
- Byte13 represents the bytes following this byte (0x0A in our example – 5 Holding Registers x 2 bytes each);
- Byte14 and Byte15 represents the value to be written on the first data address (value is 0x00C8 in our example and will be written to data address 0x01 which is Relay 2 Auto Off Time Units Register);
- Byte16 and Byte17 represents the value to be written on the second data address (value is 0x012C in our example and will be written to data address 0x02 which is Relay 3 Auto Off Time Units Register);
- Byte18 and Byte19 represents the value to be written on the third data address (value is 0x0190 in our example and will be written to data address 0x03 which is Relay 4 Auto Off Time Units Register);
- Byte20 and Byte21 represents the value to be written on the fourth data address (value is 0x01F4 in our example and will be written to data address 0x04 which is Relay 5 Auto Off Time Units Register);
- Byte22 and Byte23 represents the value to be written on the fifth data address (value is 0x01F4 in our example and will be written to data address 0x05 which is Relay 6 Auto Off Time Units Register);

Table 18. Write multiple holding registers expected answer

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12
Transaction ID		Protocol ID		Length		Unit ID	FC	Data			
0x00	0x05	0x00	0x00	0x00	0x06	0x01	0x10	0x00	0x01	0x00	0x05

- Byte9 and Byte10 represents the data address of the first Holding Register that was set;
- Byte11 and Byte12 represents the total number of Holding Registers that were written;

8.3. Report Slave ID (0x11)

Report Slave ID (0x11) command will return current Slave ID of WiFi 16 Relay Module – TCP ModBus. Example of how to read Slave ID is given below.

This command (with unit ID 0xF8) can be always used in order to recover the slave ID in case it was forgotten.

Table 19. Report slave ID command format

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
Transaction ID		Protocol ID		Length		Unit ID	FC
0x00	0x06	0x00	0x00	0x00	0x02	0xF8	0x11

Table 20. Report slave ID expected answer

Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12
Transaction ID		Protocol ID		Length		Unit ID	FC	Data			
0x00	0x06	0x00	0x00	0x00	0x06	0xF8	0x11	0x03	0x01	0x00	0x64

- Byte9 represents the number of bytes following this byte (it will be always 3);
- Byte10 represents current Slave ID* (0x01 in our example);
- Byte11 and Byte12 represents the version of the board – version is formed when this value is divided by 100 (value is 100 dec (0x64) in our example which means it is v1.00);

**Note that if you change Slave ID Holding Register, the new Slave ID will take effect on next reboot!*

9. Communication methods

9.1. TCP/IP Sockets

There are really lot of software for sending messages via TCP/IP sockets. We also provide several examples demonstrating the communication with the module.

For example one such free tool for Windows and Linux (there is and command line support) is Packet Sender:

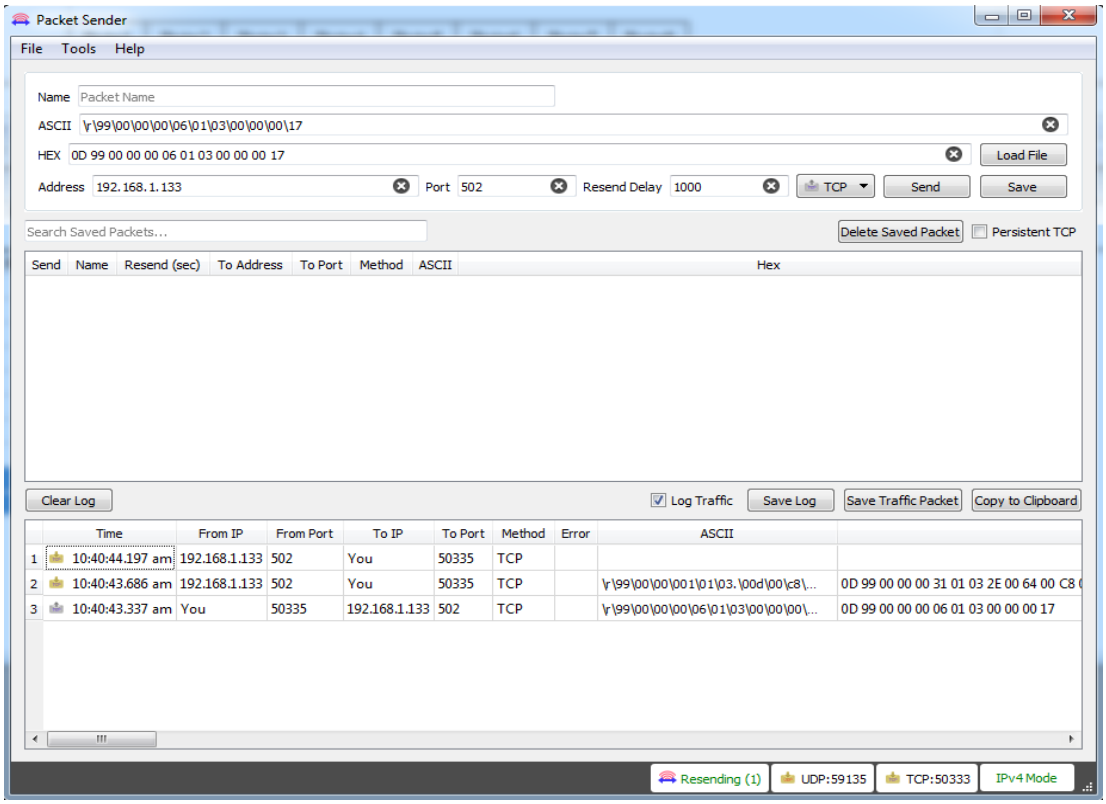


Figure 26. Packet Sender

Of course there is lot of mobile apps as well. For Android a good tool is TCP socket test:

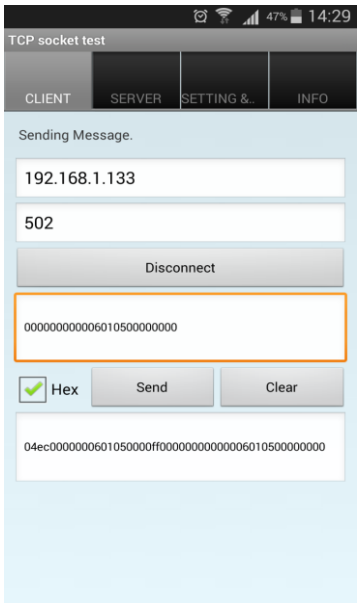


Figure 27. Android software

9.2. Virtual Serial (COM) Port

This method is also communication via TCP/IP or UDP socket, but this time the end software works with serial (com) port mapped to the certain socket. We assume the settings of the **Wi-Fi Relay Module** in that example are:

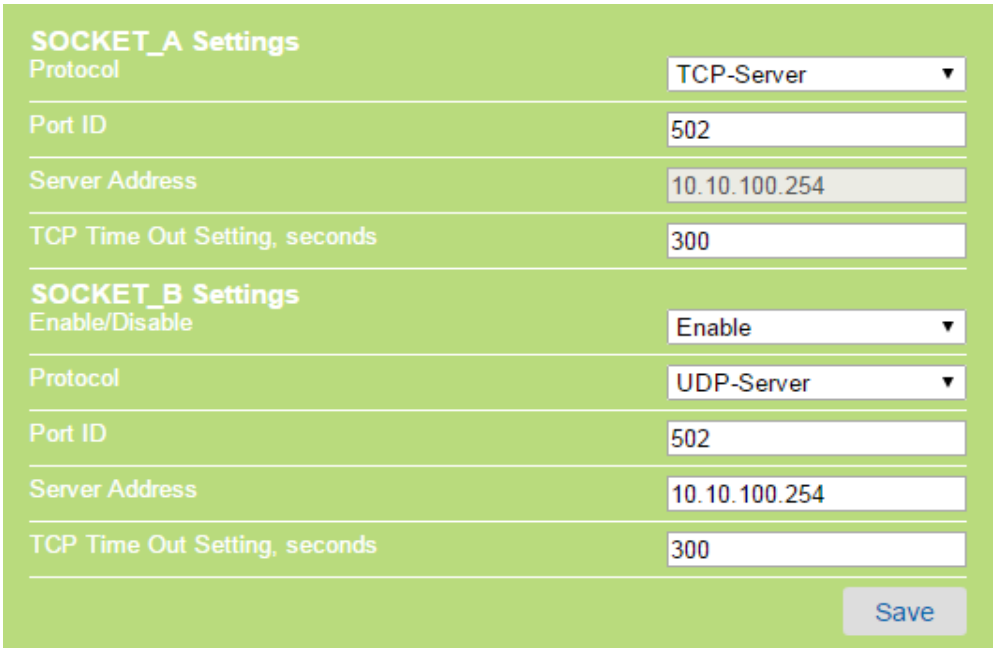


Figure 28. Sockets settings for Virtual Com Port

9.2.1. Windows

There are many applications for Windows OS for creating Virtual Com Port connected to TCP/UDP socket. For example one such tool is VSP Manager which is part of ***Tibbo Device Server Toolkit***. It is available for download from here, or here:

- 32 bit Windows installation: download
- 64 bit Windows installation: download

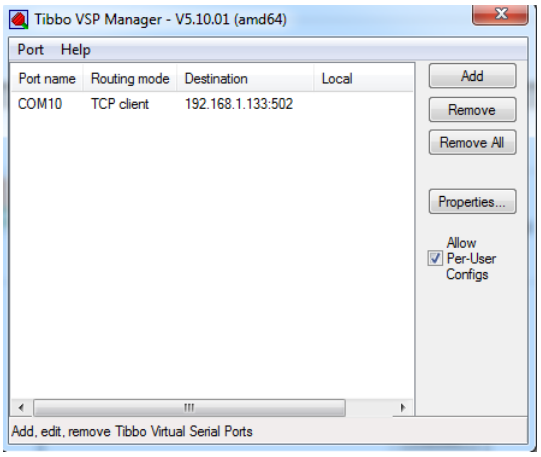


Figure 29. VSP Manager

To create virtual com port, you have to do the following settings. Make sure the IP, Port, and TCP/IP or UDP settings are the same with the Wi-Fi 16 Relay Module – TCP ModBus. Here the **Routing mode** must be "client" because the relay module is the server.

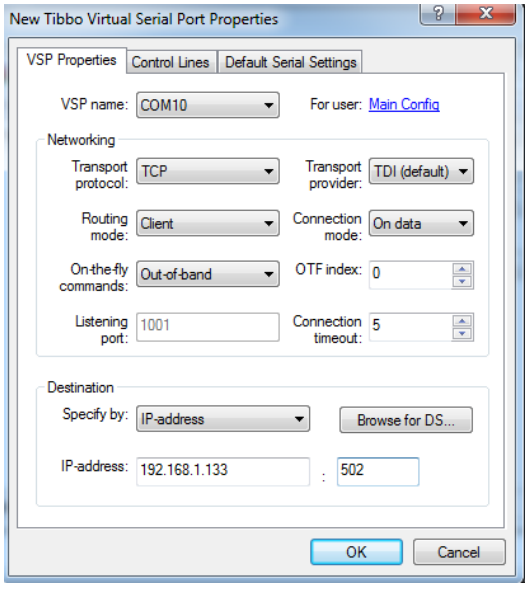


Figure 30. VSP Manager settings

After that, there should be created new com port in Device Manager (COM10):

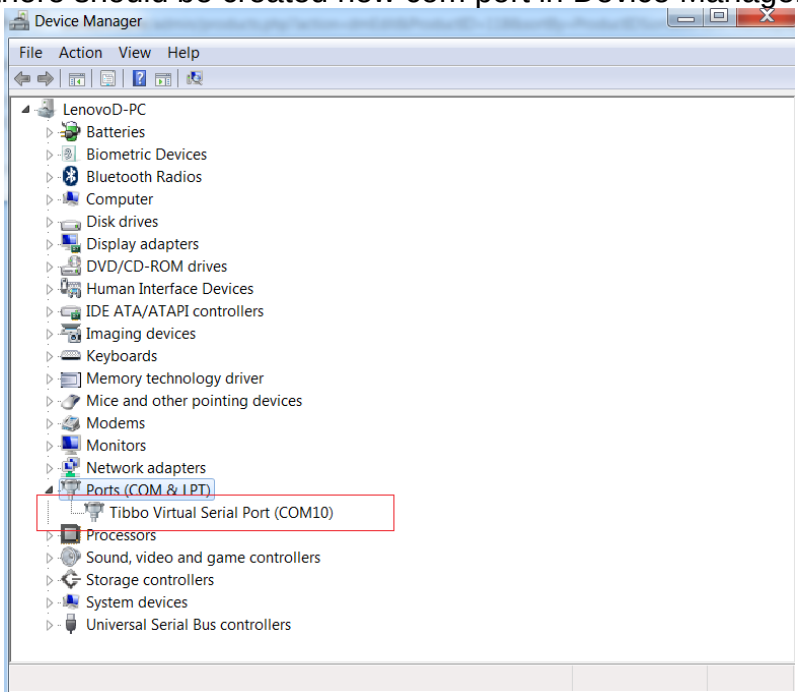


Figure 31. COM port shown in device manager

Sending and receiving data from Serial Terminal Program (in our case IO Ninja) to and from COM10 in such case is simple (example shows command Read Multiple Holding Registers).

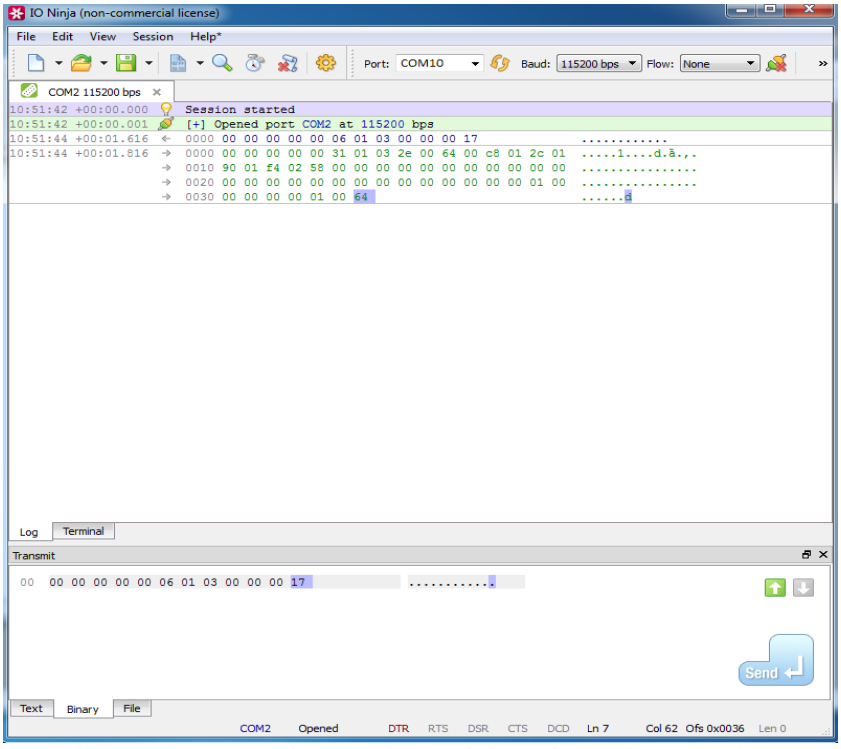


Figure 32. Sending and receiving data via IO Ninja

9.2.2. Linux

In Linux, a very simple way to interact between serial port and TCP/UDP is using socat tool.

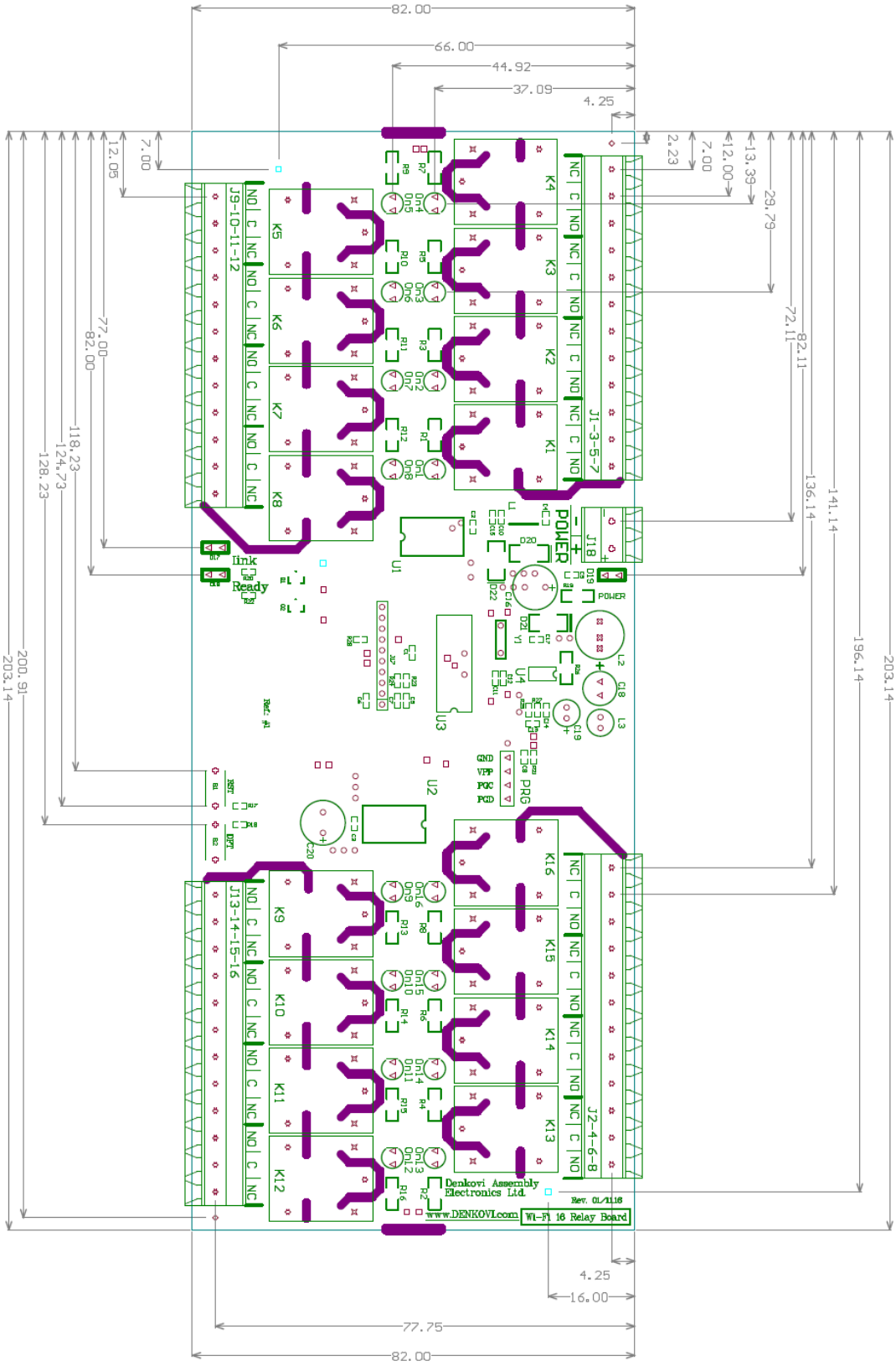
Firstly, you will need to crate link (map) between the virtual serial port and **Wi-Fi Relay Module**. In the example the module is with IP address 10.10.100.254, port 502.

socat pty,link=/dev/virtualcom0,raw tcp:10.10.100.254:502&

In order to send hex data when needed to Read Holding Registers for example, you can use this command:

echo -e "\x00\x00\x00\x00\x00\x06\x01\x03\x00\x00\x00\x17" > /dev/virtualcom0

10. Appendix 1. PCB dimensions



11. Appendix 2. BOX dimensions