MCR08N Ethernet Terminal

ISO14443 MIFARE®

HMI User Manual

Feb. 2022 Rev 5.1

Minova Technology GmbH

Company Headquarters
Auf dem Wall 29
78628 Rottweil
Germany

www.minovatech.de

Revision History

Changes of this document are listed below:

Date	Revision	Note
01.07.2017	2.0	Second release
18.08.2017	2.1	Added HMI configuration
10.10.2017	2.2	Added more parameters to config.json
27.10.2017	2.3	Added serial cable pinout
28.10.2017	2.4	Added slave reader support
18.11.2017	2.5	Added keyboard option and new mifare commands
22.11.2017	2.6	Added offline operation
05.12.2017	2.7	Added offline functions
13.12.2017	2.8	Added keyboard parameter
14.12.2017	2.9	Added offline message screen option
30.01.2018	3.0	Added GSM interface
20.05.2018	3.1	Added encrypted communication mode
12.06.2018	3.2	Changed AES encryption mode to CBC
26.09.2018	3.3	Added wait command
12.11.2018	3.4	Added serial interfaces
20.11.2018	3.5	Added new parameters
15.02.2019	3.6	Added MCRN2 slave support
01.07.2019	3.7	Added dateTime parameters
30.10.2019	3.8	Added offline mode parameter 2
20.01.2020	3.9	Added configuration commands
30.01.2020	4.0	Added FTP upload command
03.08.2020	4.1	Added offline activity sending
20.09.2020	4.2	Added MCR08N variant
20.09.2020	4.3	Added hash function
01.12.2020	4.4	Added sound synthesizer parameters
01.01.2021	4.5	Changed Buzzer command parameters
01.03.2021	4.6	Added JSON protocol
01.05.2021	4.7	Added whitelist file synch function
22.07.2021	4.8	Added fixed button IDs for extended functions
18.10.2021	4.9	Added extRFID parameter into config.json
01.01.2022	5.0	Added new photos
10.02.2022	5.1	Added file synch via FTP

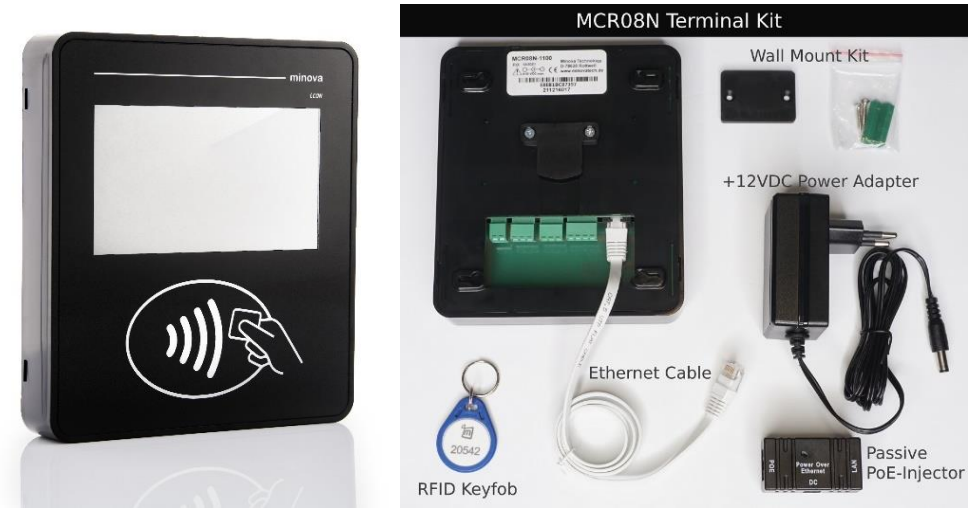
Table of Contents

1 Precautions Before Setup & First Run

- **If your network contains managed/smart switches such as (Cisco, Allied Telesis etc.)**
  - RSTP, STP (Spanning Tree Protocols) and related protocols must be turned off or disabled from the management console of the switch. These protocols may cause the terminals to start to gain ip late at first power-on or unable to take an IP address on the network properly.

- **If your network contains a Firewall**
  - Make sure that your network does not have a MAC-Filter.
  - Make sure that UDP 65535 port (terminal discover port i.e miFinder Config) should not be blocked.
  - If your device is unable to gain an IP address from the DHCP server, please define or give freedom to the MAC ID of the terminal in your network (via management console of firewall, router etc.)
  - Make also sure that TCP ports used by the terminal should not be blocked.

2 Introduction

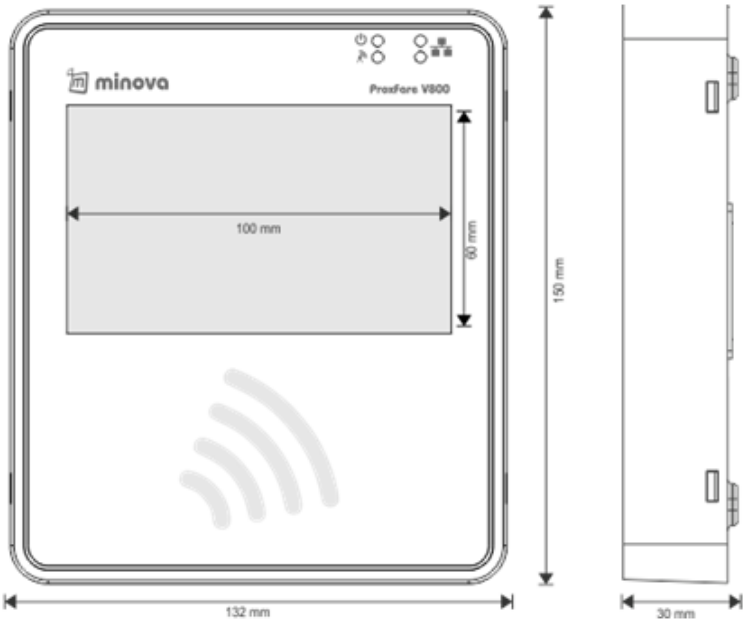


MCR08N is a high performance and flexible ISO14443-A contactless smart card reader terminal supporting read-write capabilities. The terminal is based on the 13.56 MHz contactless technology and is fully compatible with the entire MIFARE® family, as well as supporting ISO14443A contactless standard. The terminal comes with Ethernet connectivity and has extensive software support package that is optimized for easy integration.

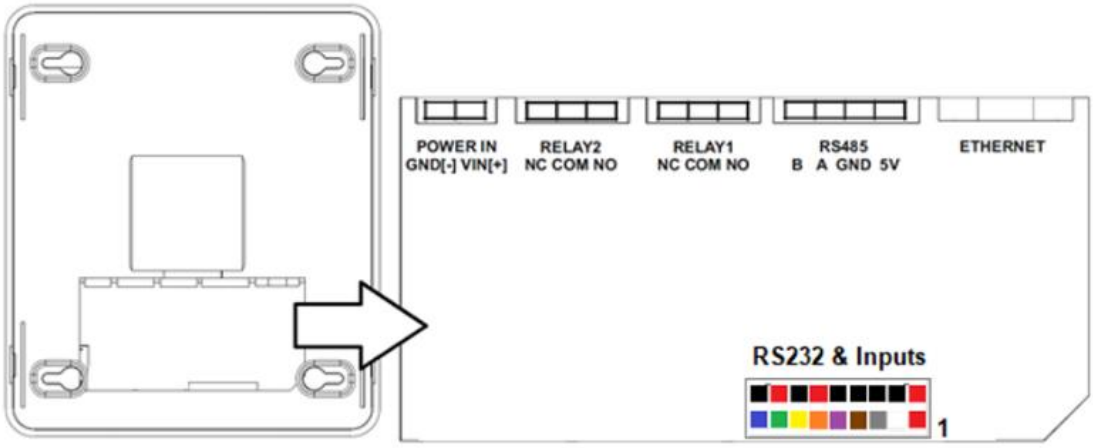
3 Features

- MIFARE® ISO14443A, Classic, Plus, DESFire and Ultralight cards read/write
- TFT Color LCD with 480 x 272 pixel resolution
- LED backlight
- +12V nominal, 9-36V DC operation
- Current consumption is 250mA avg. @12V
    - +12V and >2A power supply is recommended
- 32-bit ARM Cortex High Performance CPU
- Up to 8MB internal flash
- Up to 4GB SD/MMC memory support
- Audio (speaker or buzzer) and LED visualization
- Communications:
    - Ethernet 10/100 Base communications
    - RS232 port (300-115200 Baud)
    - RS485 port (300-115200 Baud)
    - MDB slave
    - GSM 2G/4G interface (optional)
- Inputs & Outputs
    - Relay outputs with NO and NC (1.5A dry contact)
    - 4xDigital opto-isolated inputs
- Offline operation with internal flash
- Built-in Real Time Clock (RTC) with battery back-up
- Etherhet:
    - Ethernet 10/100 Base
    - Globally unique MAC address
    - Integrated TCP/IP stack
    - TCP/IP Client-Server communication
    - UDP support
    - DHCP or static IP operation
    - ICMP ping support
    - Configurable over local network with miFinder software
- -20 °C … +85 °C industrial operating temperature

4 Mechanical Dimensions



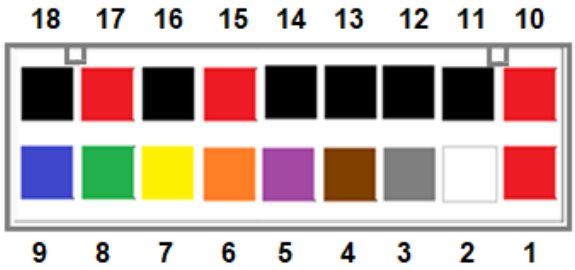
5 Ports



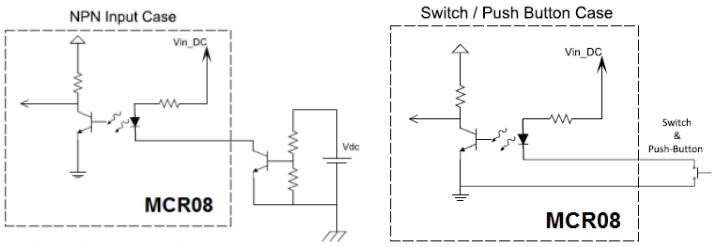
POWER IN: +12V power supply input (alternatively through Ethernet injector)
RS485: General Purpose External Communication for RS485 Peripherals

6 Pinning

Main Connector	
1	+5V Output (optional voltage ref. for inputs)
2	EXT_IN0
3	EXT_IN1
4	EXT_IN2
5	EXT_IN3
6	COM1-RXD (RS232)
7	COM1-TXD (RS232)
8	COM2-RXD (empty in GSM version)
9	COM2-TXD (empty in GSM version)
10	+5V Output
11	GND
12	GND
13	GND
14	GND
15	+5V Output
16	GND
17	+5V Output
18	GND



Input Circuits



6.1 Powering the reader

The reader may be powered directly through the power input connector or alternatively through the Ethernet cable via a PoE-Injector.

7 Basic Functionality

MCR08 can be shipped with a pre-configured software for standard operation purposes upon customer request. Device initializes by self-testing the hardware and checks the presence of micro SD card. SD card is required for performing standard functions.

Configuration file is loaded for screen messages and device parameters. After initialization, device is ready for accepting contactless cards.

Device connects to the configured server to send the login data packet. After connecting to internet (via Ethernet, GSM/GPRS or Wi-Fi), if configured properly, built in FTP server is started and device will be ready to accept incoming FTP connections.

A card read operation in online mode, triggers an Http (GET) transfer to the configured web service. After sending the card information, device waits for the server to respond to accept or reject the card and relay output (if used) is triggered upon a positive response.

In offline mode, the result of the card read/write operation is saved to an offline log file. Current date is used for the file naming to ease the file processing process. When device goes back online, a data packet is sent to the server to inform the presence of offline records. Server is responsible for connecting the device as an FTP client, pulling the offline log file from the log folder and deleting the file after a successful transfer.

Screen messages should be configured by transferring the appropriate configuration file via FTP. Device reads messages and screen background colors during initialization.

8 Operating Modes & Setup

MCR0x Ethernet Terminals can be used in either Client or Server. In client mode the terminal connects to a remote server that it is listening a TCP/UDP port. Server may accept multiple connections. MCR0x Terminals can be used as Server. The terminal listens own port and can accept a connection request from outside. In this case terminal's IP number is to be static. It depends on the application whether the terminal is in Client or Server mode.

8.1.1 TCP/IP Client Mode Operation

When the terminal is set to operate in client mode it tries to connect a remote server IP & Port set in configuration.

8.1.2 TCP/IP Server Mode Operation

When the terminal is set to operate in server mode it listens own TCP port to accept outcoming request from other clients.

8.2 Terminal Setup & Settings

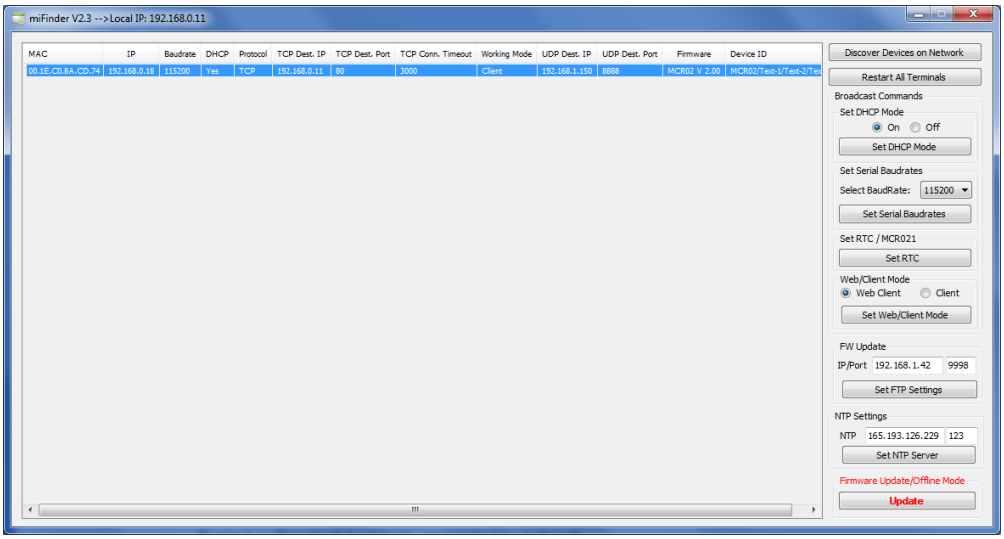
The terminal can be configured on a network (LAN). To start setup terminal must be in a network that supports DHCP. The terminal needs to acquire an IP from a DHCP server on your network. Configuration is made through and UDP protocol so it advisable to use a firewall free network. Most of the firewalls filter UDP.

For the first time setup you can use miFinder software. miFinder can discover all terminals on your network. After MCR0x is up i.e. (after gained an IP from your network) you can use miFinder. It is also advisable to turn off any firewall & antivirus software before running miFinder. As stated before, firewalls on PC may prevent to discover the network.

8.3 miFinder Configuration Software

Using miFinder you can set various parameters related to terminals. Some parameters are specific to each terminal and some parameters are global to all terminals. After all setup, your device is listed or discovered as given below.

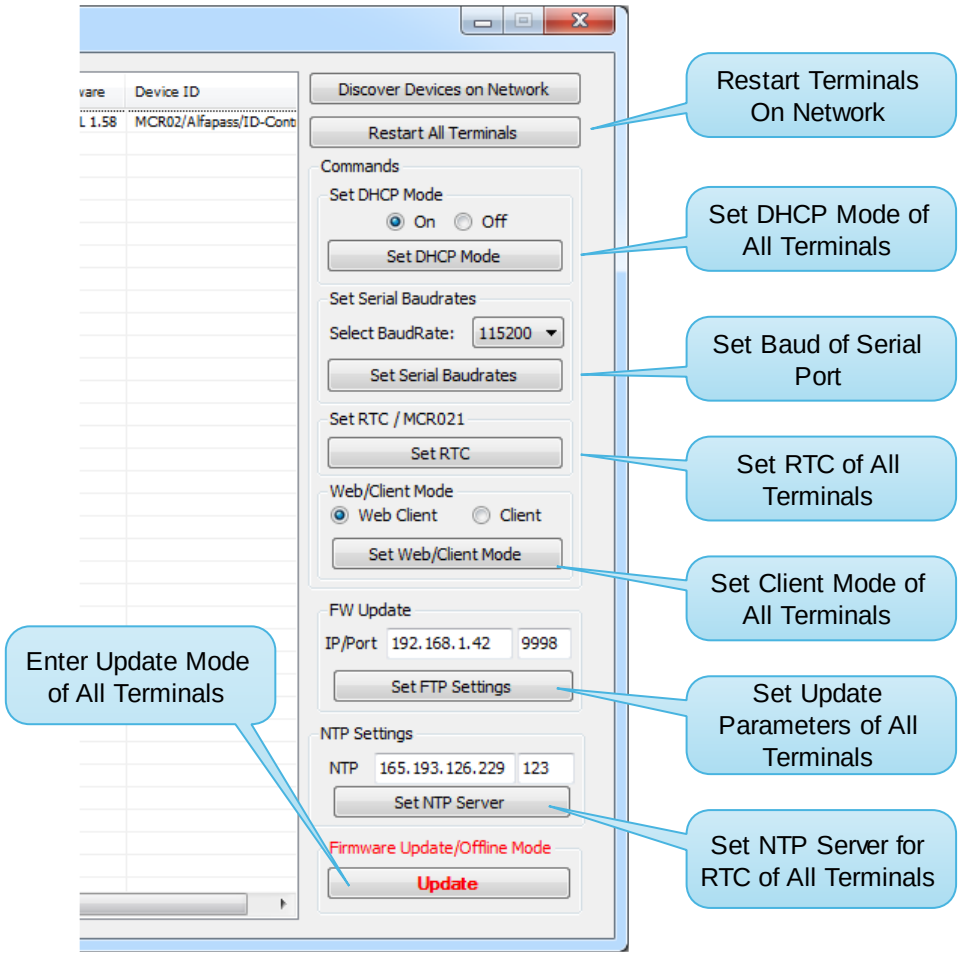
If your terminal is not discovered, press Discover button again.



miFinder Main Screen

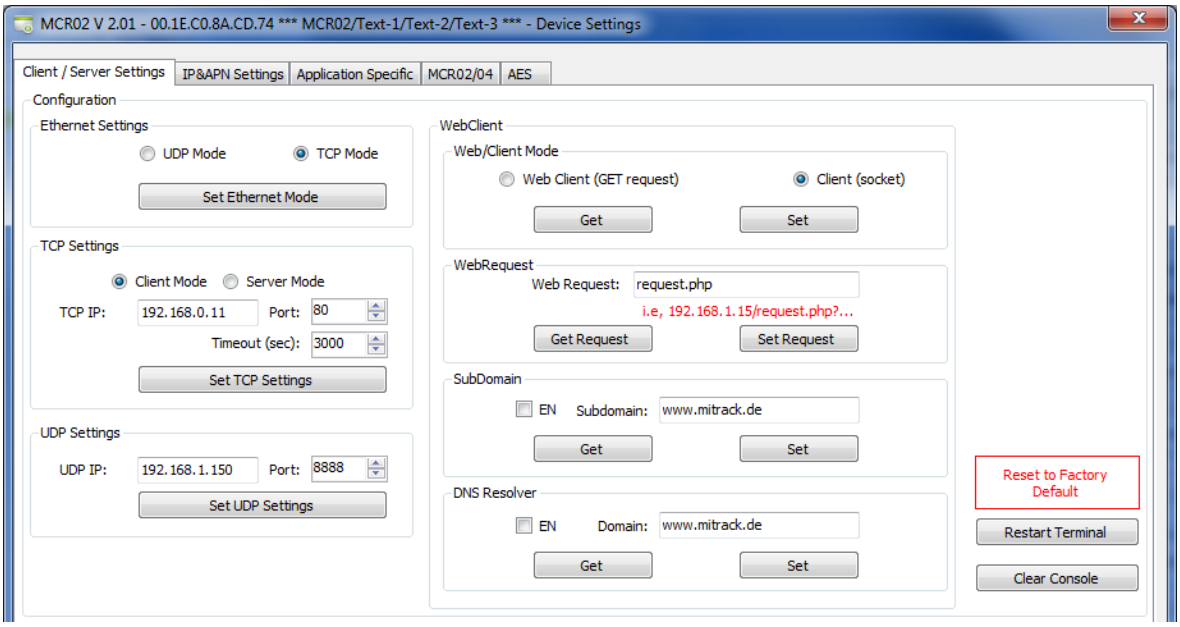
In main window of miFinder you can the following parameters

- Restart or Reset all terminals connected to network.
- Setting DHCP parameter of all terminals connected to network.
- Setting the baud-rate of RS232 / RS485 port.
- Setting the Real Time Clock of all terminals connected to network.
- Setting the Client mode type: Web or Normal Socket Client of all terminals connected to network.



miFinder main window view

To enter a detailed setup of a particular terminal select a device from the list and double click to see a particular terminal setting window in miFinder. This window gives you a detailed setup of each terminal. Please note that these settings are specific to each terminal. Below given a snapshot of detailed settings window of miFinder.



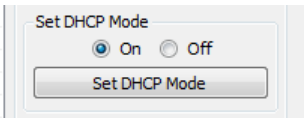
miFinder Terminal Setting Window

This setting window gives you to:

- Set the terminal IP static or dynamic
- Set protocol type of operation of the terminal: UDP or TCP
- Set UDP remote ip & port
- Set TCP operation mode: Client or Server
- Set TCP remote ip and server port
- Set remote request file with GET in Web-Client mode
- Set / Clear Relays to test
- Set RTC synchronized with PC clock.
- Reset to factory default configuration of selected terminal.
- View Firmware version of the terminal.

8.3.1 Automatic IP (DHCP) Mode

In miFinder's main screen, in Set DHCP Mode section, select ON and press Set DHCP Mode button. Then all terminals restart and try to access a DHCP server to get an IP address from your network. Please note that your network must have a DHCP enabled management device.



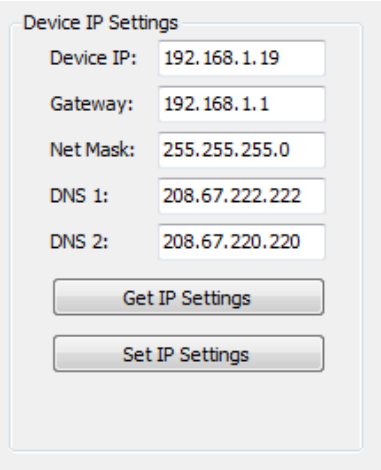
DHCP Mode Setting

8.3.2 Constant / Static IP Mode

The terminal is set up with the following IP parameters for static operation at factory.

Parameter	Value
IP Address	192.168.1.100
Gateway Address	192.168.1.1
Net Mask	255.255.255.0
Primary DNS	192.168.1.1
Secondary DNS	192.168.1.1

However, most of the terminals manufactured are set to operate in DHCP mode. Static IP mode is not preferred for mass productions. The default setting for all terminals is DHCP.
To set a terminal to be run in static IP mode, in miFinder's main window enter the desired terminal's settings screen. Then enter your desired IP, GW, Mask and DNS values in Device IP Settings section.



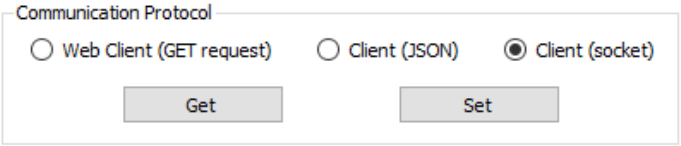
IP Parameters Settings Section

Note that, after opening settings screen, this section gives your terminal's current IP parameters. After entering the values as above figure, then press Set IP Settings button. Then the terminal restarts again.

The last step is to set DHCP mode to OFF in main window of miFinder as given in above figure. The terminal restarts again in Static IP mode. Please note that you can skip this step if your terminal is already operating in static IP mode.

8.3.3 Client and Web Client Modes of Operation

Any MCR0x terminal can connect to web server or server as client. The terminal's client mode of operation can be altered in main screen of miFinder as in below figure.

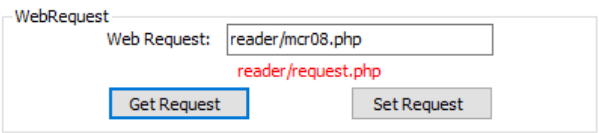


Web Client or Normal Client Mode Setting

When a contactless card is detected by the terminal, it tries to send card's UID to server as follows:

Operation Type	Example Terminal Request
Web Client	***GET /reader/mcr08.php?devID=MCR08-2310&UID=F0C189A5***
Client (JSON)	***{"devID":"MCR08-2310","MSG1":"UID;F0C189A5"}***
Client (socket)	***MCR08-2310,UID=F0C189A5***

It is seen that the terminal tries to send data to a web-server by requesting a PHP file. You can also change this request in miFinder's settings window.



Http request file name for Web-Client Mode

8.3.4 Message Format (client)

The message format from server to terminal is given by the following syntax.
<Device ID>,<CMD1;parameter1;…;parameterN>,<CMD2;parameter1;…;parameterN>,…

This packet can be sent by a specific TCP server via socket_send API's or simple echo statements defined in a web server protocol.

Max. 20 commands can be sent, and each command can have max. 50 chars.

Example:
MCR08-2130,ACK;Thank you;Card valid,RELAY1=1500,QUERY;Leaving now?,NOTIFY;Please call;the secretary,MSG;Thank you

In this example following 5 different commands are send in one message
ACK;Thank you;Card valid
RELAY1=1500
QUERY;Leaving now?
NOTIFY;Please call;the secretary
MSG;Thank you

The message format from terminal to server is given by the following syntax.
<Device ID>,<ANSWER;VALUE1;VALUE2>
or
<Device ID>,ACK
<Device ID>,NAK

Examples:
MCR08-2130,UID=4FA20135
MCR08-2310,IN=0F,OUT=01
MCR08-2130,ACK
MCR08-2130,NAK

All command strings should be terminated with CR LF (0x0D 0x0A) characters.

8.3.5 Message Format (JSON)

Server-to-Terminal:
{"devID":"*DEVICEID*","MSG1":"*CMD;PAR1;PAR2..*","MSG2":"*CMD;PAR1;PAR2..*"}
Maximum 6 command messages are supported, maximum message size is 128 bytes.

Terminal-to-Server:
{"devID":"*DEVICEID*","MSG1":"*RESP;PAR1;PAR2..*","MSG2":"*RESP;PAR1;PAR2..*"}

Example 1:
TX: {"devID":"MCR08-7134","MSG1":"VERSION?"}
RX: {"devID":"MCR08-7134","MSG1":"VERSION;MCR08GN.2.81"}
Example 2:
TX: {"devID":"MCR08-7134","MSG1":"ACK;Welcome;Access granted;Thank you;Have a nice day;32CD32"}
RX: {"devID":"MCR08-7134","MSG1":"ACK"}
Example 3:
TX: {"devID":"MCR08-7134","MSG1":"BLOCKREADX;0"}
RX: {"devID":"MCR08-7134","MSG1":"BLOCKDATAX;1cf46edc5a880400468eb41041704607"}
Example 4 (3 commands):
TX: {"devID":"MCR08-7134","MSG1":"RELAY1;ON","MSG2":"DELAY;500","MSG3":"RELAY1;OFF"}
RX: {"devID":"MCR08-7134","MSG1":"ACK"}
Example 5:
UID Event: {"devID":"MCR08-7134","MSG1":"UID;1CF46EDC"}
ALIVE Event: {"devID":"MCR08-7134","MSG1":"ALIVE"}
BUTTON Event: {"devID":"MCR08-7134","MSG1":"BUTTON;150"}

9 Messages and Command Set

Terminal to server (events)

Message	Description	Example
ALIVE	Send periodically every 30s	*MCR08-2310,ALIVE*
UID	Card ID	*MCR08-2310,UID=F543A9B8*
INPUTS	Input change	*MCR08-2310,IN=0F*
BUTTON	Button press	*MCR08-2310,BUTTON=100*

Server to terminal (command)

Command	Description	Example
VERSION?	Gets the firmware version	MCR08-2310,VERSION? *Answer: MCR08-2310,VERSION=MCR08.5.5*
RELAY1=ON/OFF **RELAY2=ON/OFF**	Set/release a relay	*MCR08-2310,RELAY1=ON*
RELAY1=ms **RELAY2=ms**	Activate relay by a delay in ms	*MCR08-2310,RELAY1=1000 (one second long)*
ACK;TEXT1;TEXT2;TEXT3;TEXT4;COLOR **NAK;TEXT1;TEXT2;TEXT3;TEXT4;COLOR**	Approve/deny entrance by displaying an ACK/NAK message HEX color background is optional	*MCR08-2310,ACK;Card valid;Access granted,RELAY1=1500* *MCR08-2310,NAK;Invalid card;No access*
MSG;TEXT1;TEXT2	Show a message	*MCR08-2310,MSG;Card is valid;Thank you*
NOTIFY;TEXT1;TEXT2	Send a message to the user	*MCR08-2310,NOTIFY;Please call the secretary;Thank you*
QUERY;TEXT	Ask user a query with yes/no buttons	*MCR08-2310,QUERY;Start loading?* *Answer: QUERY=0->No, QUERY=1->Yes*
REQUEST;TEXT	Ask user to enter a number	*MCR08-2310,REQUEST;Please enter the amount* *Answer: ENTRY=210*
PINPAD;TEXT	Ask user to enter a number	*MCR08-2310,*PINPAD*;Please enter the amount* *Answer:* PINPAD *=210*
KEYBOARD;TEXT;DEF	Ask user to enter a text DEF 0: Alpha, 1: numeric	*MCR08-2310,KEYBOARD;Please enter the order* *Answer: KEYBOARD=NR2017*
LOADSCREEN;NR	Call a predefined screen	*MCR08-2310,LOADSCREEN;2*
ACTIVESCREEN	Get actual screen number	*MCR08-2310,ACTIVESCREEN* *Answer: ACTIVESCREEN=1*
SETITEM;ID;TYPE;NEWVAL	Dynamically change a property of an item	*MCR08-2310,SETITEM;10;text;Minova Technology (Change text on label with ID 10)*
SETSCREEN;TYPE;NEWVAL	Dynamically change a screen property	*MCR08-2310,*SETSCREEN;bgcolor;000000 (Change backcolor to black)
TSYNC=UNIXTIME	Set RTC	*MCR08-2310,TSYNC=1412625197*
SOUND;n	Play a sound (speaker)	*MCR08-2310,SOUND;0*
BUZZER;DURATION;COUNT	Play a sound (buzzer)	*MCR08-2310,BUZZER;100;2 (2x 100ms beep)*
IOSTAT?	Get IO status	*MCR08-2310,IOSTAT?* *Answer: MCR08-2310,IN=0F,OUT=01*
TRST	System reset	*MCR08-2310,TRST*
COM1TX;DATA	Transmit data via comport	*MCR08-2310,COM1TX;Test print*
COM1RX	Get data from comport	*MCR08-2310,COM1RX*
RFID Commands		
GETUID	Activates an RFID tag	*MCR08-2310,GETUID* *Answer; MCR08-2310,UID=FA523C84*
LOADKEYS;TYPE;KEYA;KEYB	Load mifare keys	*MCR08-2310,LOADKEYS;0;A0A1A2A3A4A5;* *B0B1B2B3B4B5*
BLOCKREAD;BLOCKNR **BLOCKREADX;BLOCKNR**	Read 16 bytes mifare block Read 16 bytes in HEX mode	*MCR08-2310,BLOCKREAD;2* *Answer: BLOCKDATA=Test string 1* *Answer: BLOCKDATAX=000102030405060708090A0B0C0D0E0F* *Answer: NAK block authentication error*
BLOCKWRITE;BLOCKNR;DATA **BLOCKWRITEX;BLOCKNR;DATA**	Write max 16 bytes mifare block Write max 16 bytes in HEX mode	*MCR08-2310,BLOCKWRITE;2;Test* *MCR08-2310,BLOCKWRITEX;2;000102030405..*
FORMATSECTOR;SECTORNR;DATA	Format a sector	*MCR08-2130,FORMATSECTOR;1;* *FFFFFFFFFFFFFF078069FFFFFFFFFFFF*
SECTORREAD;SECTORNR **SECTORREADX;SECTORNR**	Read 48 bytes of sector data Read 48 bytes in HEX mode	*MCR08-2130,SECTORREAD;1* *MCR08-2130,SECTORREADX;1*

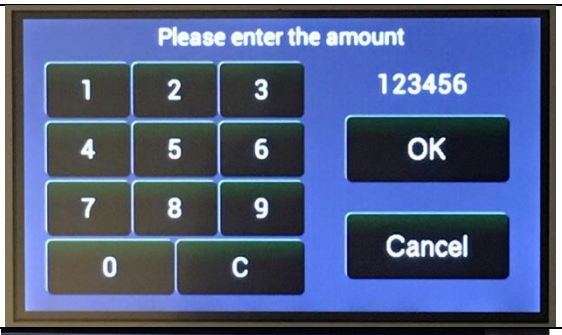
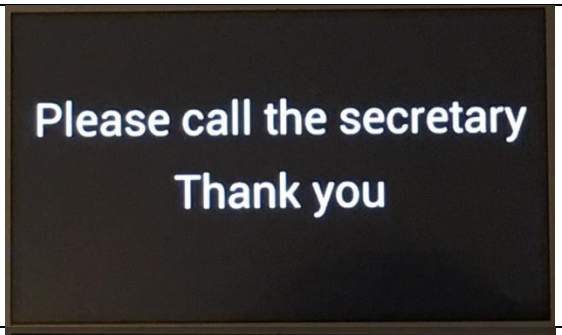
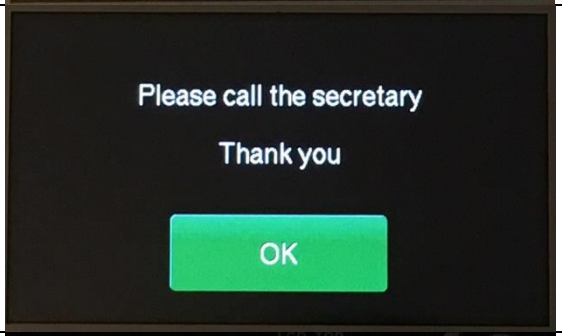
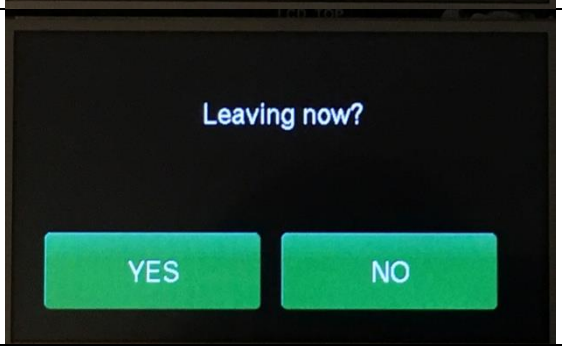
SECTORWRITE;SECTORNR;DATA **SECTORWRITEX;SECTORNR;DATA**	Write max 48 bytes of sector data Write max 48 bytes in HEX mode	*MCR08-2130,SECTORWRITE;1;MAX MUSTERMAN MUSTERSTRASSE 2 MUSTERSTADT*
CAPDU;APDU[0]..APDU[n]	Send APDU command DESFire or T=CL card	*SELPPSE: MCR08-2130,CAPDU;* *00A404000E325041592E5359532E444446303100* *Answer: MCR08-2130;RAPDU=06675041259000*
WAIT;TIME	Time in milliseconds	*WAIT;1000 (Waits one seconds as a delay)*

* Write commands: Remaining blocks will be filled with spaces in ASCII mode and with 0x00s in HEX mode

Server to terminal (configuration)

Command	Description
SETWEB	Sets the web-client settings ***MCR08-2130,SETWEB;<par1>;<par2>;<par3>;>par4>*** *par1: Enable/disable (0/1) web-client mode* *par2: Get-request path* *par3: Enable/disable (0/1) HTTP 1.1 header* *par4: HTTP1.1 host header (virtual domain name)* ***Examples:** (up to 4 parameters)* ***Send:** MCR08-2130,SETWEB;0;* ***Send:** MCR08-2130,SETWEB;1;api/rfid.php;* ***Send:** MCR08-2130,SETWEB;0;api/rfid.php;0;* ***Send:** MCR08-2130,SETWEB;0;api/rfid.php;1;login.mitrack.de*
GETWEB	Gets the web-client settings ***Send:** MCR08-2130,GETWEB;* ***Answer:** MCR08-2130,GETWEB;0;api/rfid.php;0;login.mitrack.de*
SETALIVE	Sets the alive message period ***Send:** MCR08-2130,SETALIVE;60* ***Answer:** MCR08-2130,ACK*
GETALIVE	Gets the alive message period in seconds ***Send:** MCR08-2130,GETALIVE* ***Answer:** MCR08-2130,GETALIVE;60*
SETTCP	Sets the server TCP settings for the GSM interface ***MCR08-2130,SETTCP;<par1>;<par2>;<par3>;>par4>*** *par1: Server IP* *par2: Server Port* *par3: Enable/disable (0/1) DNS lookup (connect using domain name)* *par4: Server domain name* *Examples: (up to 4 parameters)* ***Send:** MCR08-2130,SETTCP;85.214.201.95;* ***Send:** MCR08-2130,SETTCP;85.214.201.95;80;* ***Send:** MCR08-2130,SETTCP;85.214.201.95;80;0;* ***Send:** MCR08-2130,SETTCP;85.214.201.95;80;1;login.mitrack.de;* *Terminal restarts after this command!* ***Answer:** MCR08-2130,ACK,RESTART*
GETTCP	Gets the TCP/IP settings for GSM mode ***Send:** MCR08-2130,GETTCP* ***Answer:** MCR08-2130,GETTCP;85.214.201.95;80;0;login.mitrack.de;*
LOGCOUNT?	Gets the number of activity files in the LOG directory ***Send:** MCR08-2130,LOGCOUNT?* ***Answer:** MCR08-2130,LOGCOUNT=2*
FTPSYNCH	Uploads all activity files to the FTP server ***MCR08-2130,FTPSYNCH;<par1>;<par2>;<par3>;>par4>;*** *par1: IP address* *par2: Port number (21)* *par3: Username* *par4: Password* ***Example:*** ***Send:** MCR08-2130,FTPSYNCH;81.169.145.88;21;fw@minovatech.de;123456;* ***Answer:** MCR08-2130,ACK* Sent files will be removed to the “SENT” folder. Already existing files will be removed to the “ERR” folder.

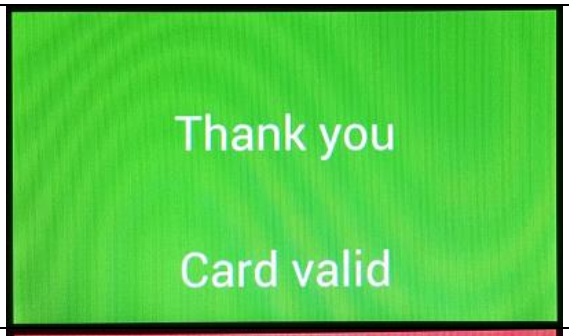
9.1 Command Examples

Host-to-Terminal MCR08-2130,PINPAD;Please enter the amount ***Terminal-to-Host*** MCR08-2130,PINPAD=123456	
Host-to-Terminal MCR08-2130,KEYBOARD;PLEASE ENTER FLIGHT NUMBER ***Terminal-to-Host*** MCR08-2130,KEYBOARD=LH1703	
Host-to-Terminal MCR08-2130,MSG;Please call the secretary;Thank you ***Terminal-to-Host*** MCR08-2130,ACK	
Host-to-Terminal MCR08-2130,NOTIFY;Please call the secretary;Thank you ***Terminal-to-Host*** MCR08-2130,ACK	
Host-to-Terminal MCR08-2130,QUERY;Leaving now? ***Terminal-to-Host*** MCR08-2130,QUERY=0 (no click) MCR08-2130,QUERY=1 (yes click)	

<table>
<tr>
<td>Host-to-Terminal

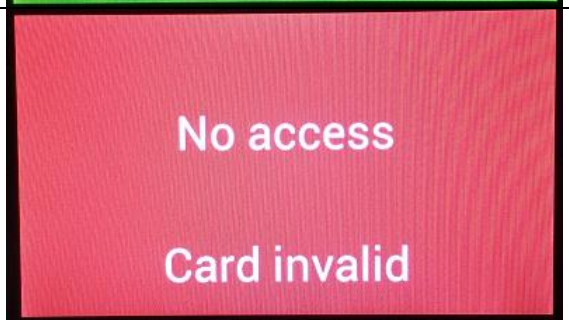
MCR08-2130,ACK; ;Thank you; ;Card valid;32CD32
(Line1 and 3 are spaced, color limegreen)

Terminal-to-Host

MCR08-2130,ACK (+ playing ACK sound)</td>
<td> 
</td>
</tr>
<tr>
<td>Host-to-Terminal

MCR08-2130,NAK; ;No access; ;Card invalid;FF0000
(Line1 and 3 are spaced, color red)

Terminal-to-Host

MCR08-2130,ACK (+ playing NAK sound)</td>
<td> 
</td>
</tr>
<tr>
<td>Host-to-Terminal

MCR08-2130,LOADSCREEN;1

Terminal-to-Host

MCR08-2130,ACK</td>
<td> 
</td>
</tr>
<tr>
<td>Host-to-Terminal

(Change button text and backcolor)
MCR08-2130,SETITEM;100;text;ButtonText,
SETITEM;100;bgcolor;0x0000FF

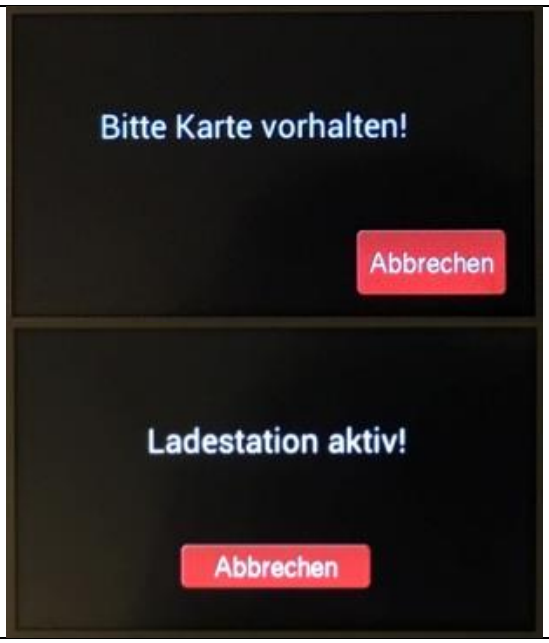
Terminal-to-Host

MCR08-2130,ACK</td>
<td> 
</td>
</tr>
<tr>
<td>Host-to-Terminal

(Hide a button)
MCR08-2130,SETITEM;100;display;0

Terminal-to-Host

MCR08-2130,ACK</td>
<td> 
</td>
</tr>
</table>

Host-to-Terminal MCR08-2130,LOADSCREEN;2 or MCR08-2130,LOADSCREEN;3 ***Terminal-to-Host*** MCR08-2130,ACK	

9.2 Loading mifare® Keys

The terminal needs the sector keys in order to read/write the related blocks. There are two keys (KeyA and KeyB) for each sector.

MCR08-2130,LOADKEYS;TYPE;KEYA;KEYB

Example; MCR08-2130,LOADKEYS;0;FFFFFFFFFFFF;FFFFFFFFFFFF

The key usage is defined in the following table.

TYPE	READ	WRITE
0	Key A	Key A
1	Key A	Key B
2	Key B	Key A
3	Key B	Key B

9.3 Formatting mifare® Sectors

Blocks 3,7,11,15,..63 are sector trailer blocks and store the KEYA, KEYB and the access conditions.

The sector trailer data must be defined correctly.

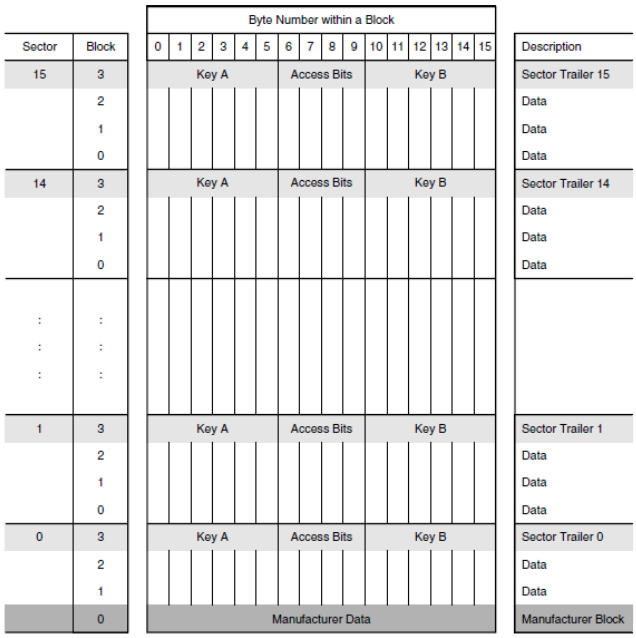
MCR08-2130,FORMATSECTOR;SECTORNR;DATA
SECTORNR = 0 to 15
DATA = KEYA-ACCESSBITS-KEYB

Examples:

```
MCR08-2130,FORMATSECTOR;1;FFFFFFFFFFFFFF078069FFFFFFFFFFFF   // Transport config R&W with KEYA
MCR08-2130,FORMATSECTOR;1;FFFFFFFFFFFF78778800FFFFFFFFFFFF   // R/W-Blocks read: KEYA, write: KEYB
MCR08-2130,FORMATSECTOR;1;FFFFFFFFFFFF08778F00FFFFFFFFFFFF   // INC/DEC-Blocks
MCR08-2130,FORMATSECTOR;1;FFFFFFFFFFFF7F00F800FFFFFFFFFFFF   // DEC-Only-Blocks
MCR08-2130,FORMATSECTOR;1;FFFFFFFFFFFF68778900FFFFFFFFFFFF   // B0;INC/DEC, B1-2 R/W blocks
```

9.4 Mifare Card Memory Layout

1024 × 8 bit EEPROM memory



10 Device Configuration

10.1 Main Configuration

MCR08-HMI is configured by placing configuration files in SD memory. Main configuration file should be named config.json and placed in the root of SD memory.
config.json should be in the format depicted below,

```
{ "config":{
 "term_id":"ALL",            → “ALL”: Unique ID,  “0000” to “FFFF” fixed ID (must be 4 digit hexadecimal)
 "relay1":"1500",            → Relay 1 duration for offline mode, set to 0 to disable it
 "relay2":"0",               → Relay 2 duration for offline mode, set to 0 to disable it
 "volume":"50",              → Speaker volume 0-to-100
 "offlinemode":"0",          → Disable offline mode. Set to 1 to enable. Set to 2 to enable without ACK/NAK messages.
 "offlinescr":"0",           → Screen number for offline management (set to 0 to disable)
 "offlinemsgscr":"0",        → Automatically switch to this screen if the server is not reachable (after disp_timeout)
 "disp_timeout":"5",         → Display timeout (time to return to main screen after display change)
 "screen_timeout":"10",      → Screen timeout for PINPAD and KEYBOARD screens
 "valid_card":"Card Valid",              → ACK message in offline mode
 "invalid_card":"Invalid Card!",         → NAK message in offline mode
 "anti_pass":"3000",         → Anti-Pass-Back time in ms
 "touchscreen":"true",       → Enable touch screen
 "forecolor":"0xFFFFFF",     → Color of main texts (DateTime, Terminal_ID and connection status)
 "ackfonth":"31",            → Font size of ACK/NAK messages
 "httpheadername":"TEST",    → Custom HTTP header name (Web-mode only)
 "httpheadervalue":"12346"   → Custom HTTP header value (Web-mode only)
 "hideDateTime":"0",         → Hide/show date time on display
 "dateDotSeparator":"1",     → Use “.” as date separator e.g. 01.05.2019 or 01/05/2019
 "rfidRemoveEvent":"1",      → Send “RFID tag is removed” event after tag exits the RF field
 "extRFID":"0"               → Integrated RFID Module (0: Disable, 1: Enable, 2 Enable with polling)
 "reconTime":"10"            → GSM variant reconnection time (minutes) in case of failed connectivity
}}
```

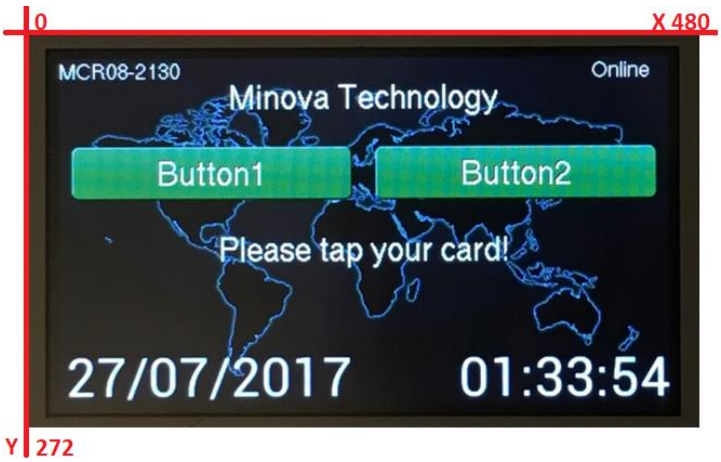
10.2 Screen Configurations

MCR08-HMI uses screen configuration files for decorating screens. Each screen should be defined in a separate file. Screen configuration files are placed in SD card. File names must be in the format of screen<screen number>.json e.g. screen1.json screen2.json etc.

Device, by default, begins with screen 1. Screen configuration file only contains decoration information and some variable values. File is in json data format, when editing, formatting rules must be strictly followed.

Sample screen1.json configuration

```
{
"itemcount":"4",          ← 4 items will be used in this screen
"screenbgcolor":"0x000000",       ← Backcolor is black
"bgimage":"background.JPG",       ← Background image is used
"hideDateTime":"0",       ← Datetime is visible
"hideDevID":"0",          ← Device ID is visible
"hideNetStatus":"0",      ← Network status is visible
"rfidActive":"1",         ← Card reading is active in this screen
"relay1Active":"0",       ← 0: disable, 1: enable, >1 trigger in ms
"relay2Active":"0",       ← Relay2 is disabled in this screen
"dateDotSeparator":"1",← Use “.” as date separator
"dateFonth":"31",         ← Date font size
"timeFonth":"31",         ← Time font size
"dateXPos":"5",           ← Date X position
"dateYPos":"251",         ← Date Y position
"timeXPos":"315",         ← Time X position
"timeYPos":"251",         ← Time Y position
"item1": {
            "itemnr":"10",          ← Unique item number
            "display":"1",          ← Is visible
            "type": "label",        ← Type is label
            "xpos": "240",          ← Top left is 0:0 position
            "ypos": "30",           ← Display size is 480x272
            "fgcolor":"0xFFFFFF",   ← Forecolor is white
            "fonth":"24",           ← Font size
            "text": "Minova Technology",  ← Label text
            "center": "1”           ← Center text
            },
```



```
"item2": {
                "itemnr":"11",
                "display":"1",
                "type": "label",
                "xpos": "240",
                "ypos": "150",
                "fgcolor":"0xFFFFFF",
                "fonth":"24",
                "text": "Please tap your card!"
                },
"item3": {
                "itemnr":"100",
                "display":"1",
                "type": "button",
                "width": "220",
                "height": "40",
                "xpos":"10",
                "ypos":"70",
                "fgcolor":"0xFFFFFF",
                "bgcolor":"0x009933",
                "fonth":"24",
                "text": "Button1"
                },
"item4": {
                "itemnr":"101",
                "display":"1",
                "type": "button",
                "width": "220",
                "height": "40",
                "xpos": "250",
                "ypos": "70",
                "fgcolor":"0xFFFFFF",
                "bgcolor":"0x009933",
                "fonth":"24",
                "text": "Button2"
                },
{
```

Dynamically changing an item property

Command: SETITEM
iItemnr: e.g. 11 (label with ID 11)
property: text, display or bgcolor

Changing the text property:
MCR08-2310,SETITEM;11;text;Please tap your card!

Hiding and item:
MCR08-2310,SETITEM;11;display;0

Changing color:
MCR08-2310,SETITEM;11;bgcolor;0x009933

* The item property will be set to default after a restart or a loadscreen command

Sample screen2.json configuration

```
{
"itemcount":"2",
"screenbgcolor":"0x000000",
"hideDateTime":"1",
"hideDevID":"1",
"hideNetStatus":"1",
"rfidActive":"1",
"item1": {
                "itemnr":"111",
                "display":"1",
                "type": "button",
                "width": "140",
                "height": "60",
                "xpos":"320",
                "ypos":"200",
                "fgcolor":"0xFFFFFF",
                "bgcolor":"0xFF0000",
                "fonth":"24",
                "text": "Abbrechen"
},
"item2": {
                "itemnr":"15",
                "display":"1";
                "type": "label",
                "width": "120",
                "height": "40",
                "xpos":"220",
                "ypos":"100",
                "fgcolor":"0xFFFFFF",
                "bgcolor":"0xFF0000",
                "fonth":"30",
                "text": "Bitte Karte vorhalten!",
                "center": "1"
                },
}
```

← **No background image is defined, instead back color is active**

Sample screen3.json configuration

```
{
"itemcount":"2",
"screenbgcolor":"0x000000",
"hideDateTime":"1",
"hideDevID":"1",
"hideNetStatus":"1",
"rfidActive":"0",
"item1": {
                "itemnr":"11",
                "display":"1",
```

← **Card reading is disabled**

```
                "type": "label",
                "xpos": "240",
                "ypos": "100",
                "fgcolor":"0xFFFFFF",
                "fonth":"30",
                "text": "Ladestation aktiv!",
                "center": "1"
                },
"item2": {
                "itemnr":"121",
                "display":"1",
                "type": "button",
                "width": "180",
                "height": "40",
                "xpos":"150",
                "ypos":"200",
                "fgcolor":"0xFFFFFF",
                "bgcolor":"0xFF0000",
                "fonth":"24",
                "text": "Abbrechen"
                },
}
```


10.2.1 Defining screen names

Please use this format screen1.json, screen2.json…

10.2.2 Defining number of items

"itemcount":"6" (e.g. 2 Buttons 4 labels)

Maximum 10 items can be defined for each screen.

10.2.3 Numbering of items “itemnr”

Buttons: “button” 100-150 (this number will be sent on button press to the server)
Labels: “label” 10-50 (nr. 50 is used for cardholder name in offline mode)

Please define for each item a unique ID number.

10.2.4 Adding a background image to the screen

"bgimage":"Background1.JPG",

Define the image name in screenx.json and copy the JPG image under *media* folder

Size of image: 480 x 272 pixel

10.2.5 Coloring of items

HTML Color Codes are used to define back and fore colors for items.

Following free color tools can be used to find easily colors for your screen configuration.

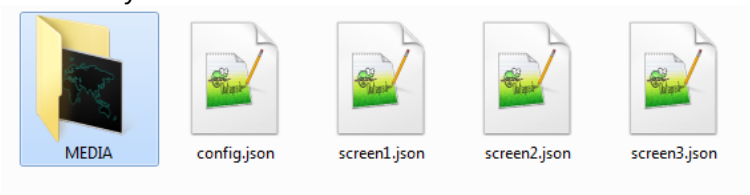
http://html-color-codes.info/
http://htmlcolorcodes.com/

10.2.6 Loading and Testing a Screen Configuration

- Upload the screen configuration file ***screenX.json*** via FTP in the root of the SD memory
- Load the screen configuration by calling the following command

MCR08-2130,LOADSCREEN;X

SD Memory Content



MCRN08N Version with UTF-8 Support

MCR08N uses a 32MB NAND-Flash Memory instead of an SD card.

- No sub folders are supported
- All files are in the root directory

10.2.7 Dynamically Changing a Screen Configuration

Command: SETSCREEN
property: bgcolor, bgimage or rfidActive

Changing the bgcolor (background color) property:
*MCR08-2310,*SETSCREEN;bgcolor;000000

Changing the bgimage (background image) property:
*MCR08-2310,*SETSCREEN;bgimage;BG1.JPG

Changing the rfidActive (polling) property:
*MCR08-2310,*SETSCREEN;rfidActive;0 (stop polling)

10.2.8 Fixed Button IDs with Extended Functions

Following fixed button IDs request to tap the card, type a value or date.

Button ID	UID	Button ID	Num. Value	Date 1	Date 2
151-to-158	x	x			
159 and 160	x	x	x		
161 and 162	x	x		x	
163 and 164	x	x		x	x

Examples:

```
MCR08-2310,UID=1CF46EDC,BUTTON=151
MCR08-2310,UID=1CF46EDC,BUTTON=160,VALUE=12
MCR08-2310,UID=1CF46EDC,BUTTON=161,DATE=1.2.2021
MCR08-2310,UID=1CF46EDC,BUTTON=163,DATE1=1.2.2021,DATE2=2.2.2021
```

Following default texts may be defined in ***config.json***

..
"scan_msg":"Tap your card please!",
"value_msg":"Enter a value please!",
"date_msg":"Enter a date please!"
}

11 Offline Mode of Operation

If the server connection fails or offline operation is desired, the MCR08 searches the UID in the white list and logs all activities.

11.1 White List File "*cards.json*"

```
{
        "whitelist ":{
        "cards":[
                { "id":"0E8AD4A4", "profile":"1","name":"Barbara Scott", "nr":"12346"},
                { "id":"26E400BE", "profile":"1","name":"John Winter", "nr":"12225"}
                ]
        }
}
```

"id": Card or Tag UID (4 or 7 Bytes hexadecimal)
"profile": Profile number from profiles.json (if limited access is supported)
"name": Card holder name
"nr": Personal ID number (in case of pin-pad enabled entry)

11.2 Offline Settings in Main Configuration "*config.json*"

```
{
 "config":{
 "term_id":"ALL",
 "relay1":"0",              // Relay 1 time in milliseconds (set to 0 if offlinescr>0)
 "relay2":"0",              // Relay 2 time in milliseconds (set to 0 if offlinescr>0)
 "volume":"80",
 "offlinemode":"0",         // Disable offline mode (1: to enable, 2: to accept all cards)
 "offlinescr":"4",          // Screen number for offline management (set to 0 in access control mode)
 "offlinemsgscr":"0",       // Set this to zero in offline management mode
 "disp_timeout":"5",
 "valid_card":"Card Valid",                 // ACK message
 "invalid_card":"Invalid Card!",            // NAK message
..
```

11.2.1 Access Control Mode

"offlinescr":"0" (do not load another screen)

All presented card UIDs will be logged with access result (true "found in the white list" or false "not found") information.

The desired relay can be triggered automatically according to the "relay1/2" setting in the "***config.json"*** file.

If the relays need to be set until the user ends the session, an offline screen configuration could be used.

11.2.2 Offline Screen Configuration Mode

"offlinescr":"4" (use a screen number >1)

In this mode, the defined offline screen is loaded if the UID is found in the white list.

Each button can trigger a new screen load action according to its offline screen setting which is defined in the item parameters.

"item1": {
…
"offlinescr":"1" -> 1 returns to main screen

…

Sample screen4.json offline configuration

```
{
        "itemcount":"3",
        "screenbgcolor":"0x000000",
        "hideDateTime":"1",
        "hideDevID":"1",
        "hideNetStatus":"1",
        "rfidActive":"0",
        "relay1Active":"1",       -> relay1 is set as soon as this screen is loaded
        "relay2Active":"0",
        "item1": {
                "itemnr":"111",
                "display":"1",
                "type": "button",
                "width": "140",
                "height": "60",
                "xpos":"175",
                "ypos":"150",
                "fgcolor":"0xFFFFFF",
                "bgcolor":"0xFF0000",
                "fonth":"24",
                "text": "Beenden"
                "offlinescr":"1"  -> return to main screen in offline mode
                },
        "item2": {
                "itemnr":"15",
                "display":"1";
                "type": "label",
                "width": "120",
                "height": "40",
                "xpos":"245",
                "ypos":"40",
                "fgcolor":"0xFFFFFF",
                "bgcolor":"0xFF0000",
                "fonth":"30",
                "text": "Betriebsmodus"
                },
        "item3": {
                "itemnr":"50",            -> nr. 50 is used for carholder name, in this case this label shows the matched cardholder
                "display":"1";
                "type": "label",
                "width": "120",
                "height": "40",
                "xpos":"245",
                "ypos":"100",
                "fgcolor":"0xFFFFFF",
                "bgcolor":"0xFF0000",
                "fonth":"28",
                "text": "cardholder"      -> This will be overwritten by the cardholder name from the card.json file
                },
}
```

11.3 Offline LOG File *YYYY.MM.DD.json*

Sample records:

{"accessLog":"offline","deviceID":"3103","cardUID":"26E400BE","personID":"","sequenceNr":"0","date":"22.11.2017","time":"02:04:47","accessType":"0","accessDetail":"0","readerSource":"0","accessResult":"true"}

{"accessLog":"offline","deviceID":"3103","cardUID":"26E400BE","personID":"","sequenceNr":"0","date":"22.11.2017","time":"02:04:54","accessType":"120","accessDetail":"1","readerSource":"3","accessResult":"button"}

- accessType
  - 0 : Card read event
  - 100-150 : ButtonID (itemnr)
- accessDetail
  - 0 : Card read event
  - 1-n : screennr (button press event)
- accessResult
  - true : UID is found in the whitelist
  - false : UID is not found in the whitelist
  - button : Button press event
- readerSource
  - 0 : Master reader (terminal itself)
  - 1 : Slave reader (RS485-Network)
  - 2 : Pin-pad entry (Personel ID number)
  - 3 : Button entry

11.4 Offline Activity Manager

The MCR08 saves all offline activities into the memory in order to send them later when the connection is established (max 250 bookings).

To use this mode, set the parameter **"offlinemode"** in config.json to **2**!

```
{ "config":{
 "term_id":"ALL",        → "ALL": Unique ID, "0000" to "FFFF" fixed ID (must be 4 digits hexadecimal)
 "relay1":"1500",        → Relay 1 duration for offline mode, set to 0 to disable it
 "relay2":"0",           → Relay 2 duration for offline mode, set to 0 to disable it
 "volume":"50",          → Speaker volume 0-to-100
 "offlinemode":"2",      → Disable offline mode. Set to 1 to enable. Set to 2 to enable without ACK/NAK messages.
```

Example:

Standard Message: `MCR08-C16C,UID=E28C69AB`
Offline Message: `MCR08-C16C,OID=E28C69AB,UTIME=1596480291`
`OID is here the offline UID and UTIME the time stamp in Unix time format`

If the message includes "OID", extract the timestamp and send an ACK as answer to the reader. The reader waits until ACK is received to mark the activity as sent.

Example:

```
Receive: MCR08-C16C,OID=E28C69AB,UTIME=1596480257
Send: MCR08-C16C,ACK
Receive: MCR08-C16C,OID=4C3C3CD5,UTIME=1596480273
Send: MCR08-C16C,ACK
Receive: MCR08-C16C,OID=E28C69AB,UTIME=1596480291
Send: MCR08-C16C,ACK
```

`The ACK command` **`should not`** `include any parameters separated by ";" char.`

Offline files older than 30 days will be removed permanently. However, the json logs are always present.

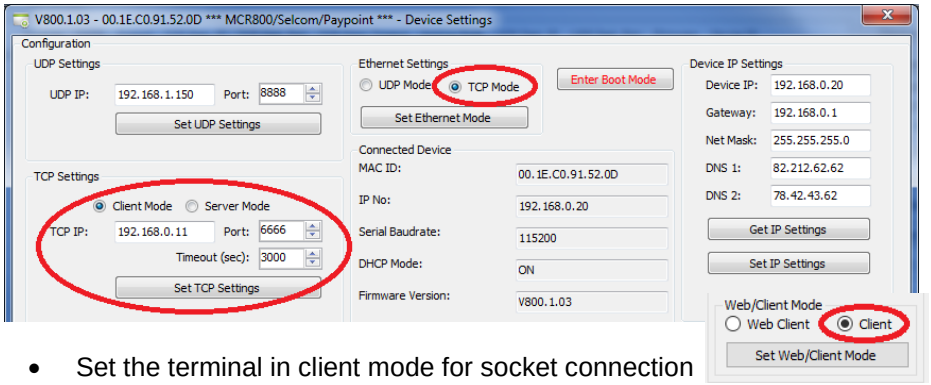
The following command erases all offline records.

```
MCR08-C16C,DELOFFLINE
```

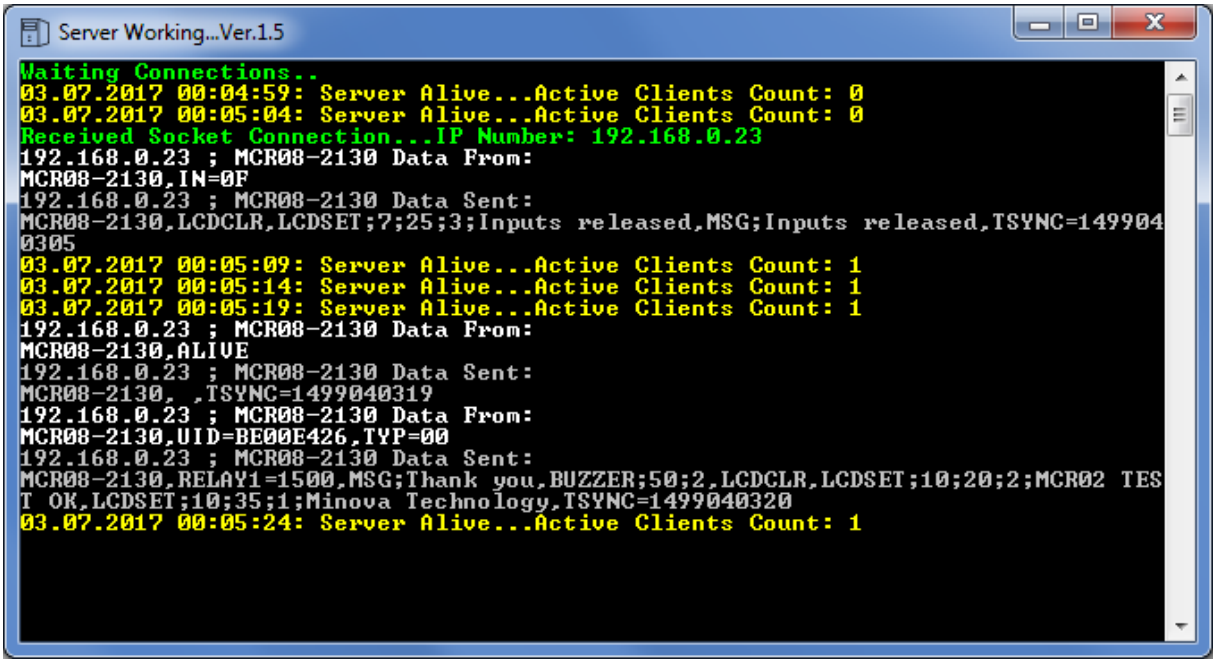
12 Example Operation with GSA Software

The ***GSA_Testserver.exe*** can be used to test the server connection. More than one terminal can connect to this multi-thread server application.

- Set your servers (PC) TCP IP and port (port is defined in *.ini file)



- Set the terminal in client mode for socket connection
- Run the ***GSA_Testserver.exe*** application and wait until the terminal is connected
- Present a contactless card to the terminal



The Terminal sends the following to Server:

MCR08-2310,UID=1E2C8E94

The Server may send the following to Terminal:

To approve:
MCR08-2310,ACK;Card is valid;Thank you,RELAY1=1500,TSYNC=1475792451
To deny:
MCR08-2310,NAK;Invalid card;Access denied,TSYNC=1475792451

The server application adds to each response the TSYNC command with the actual Unixtime. This way the RTC is always synchronized with the server.

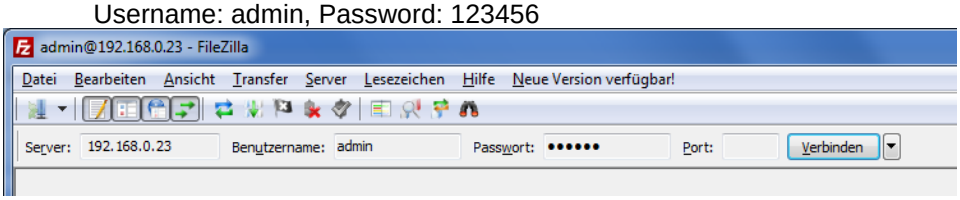
The source code of this server project is included in the SDK.

13 Firmware Update

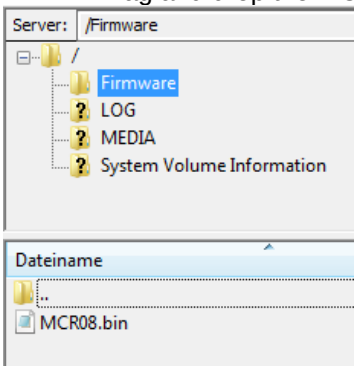
We will describe the basic steps to update/load/modify or program a new firmware for the MCR08 terminal family.

13.1 Via FTP-Client

1- Explore the SD-Card using an FTP-Client (e.g. Filezilla)
Username: admin, Password: 123456



2- Drag and drop the MCR08.bin file into the Firmware folder (into root in case of MCR08N)



3- Perform a system reset

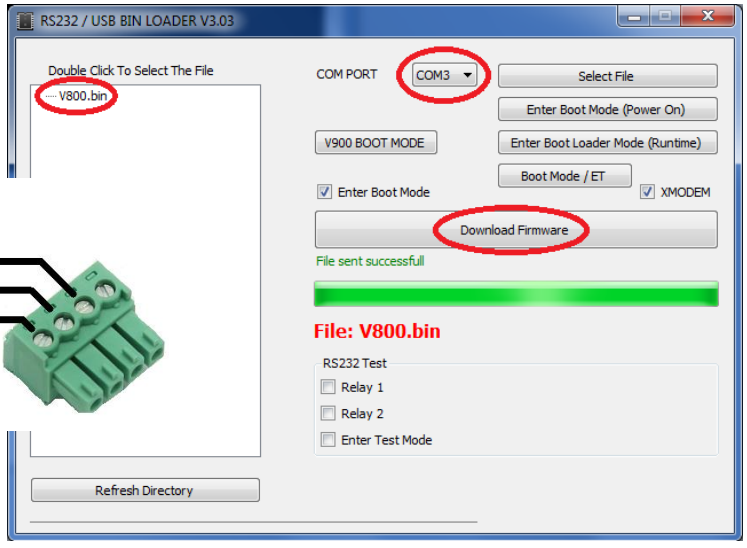
13.2 Via Serial Cable

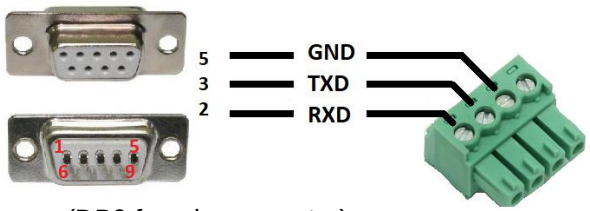
Requirements:

- RS232 download cable (USB – Serial converter if PC does not have a serial port)
- PC with Windows 7 or newer Windows OS
- RTU-FW-Downloader.exe program (.NET framework is required to run this program)

Sequence:

- Connect the RS232 download cable supplied with the terminal between **COM1** port of the terminal and com port of the PC.
- You can use an USB-Serial converter if PC does not have any serial port.
- Run the ***RTU-FW-Downloader.exe***
- Select the bin file
- Select the ***COM PORT***
- Click on ***Download Firmware***
- The new firmware will be programmed automatically



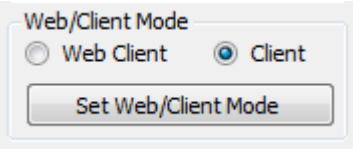


(DB9 female connector)

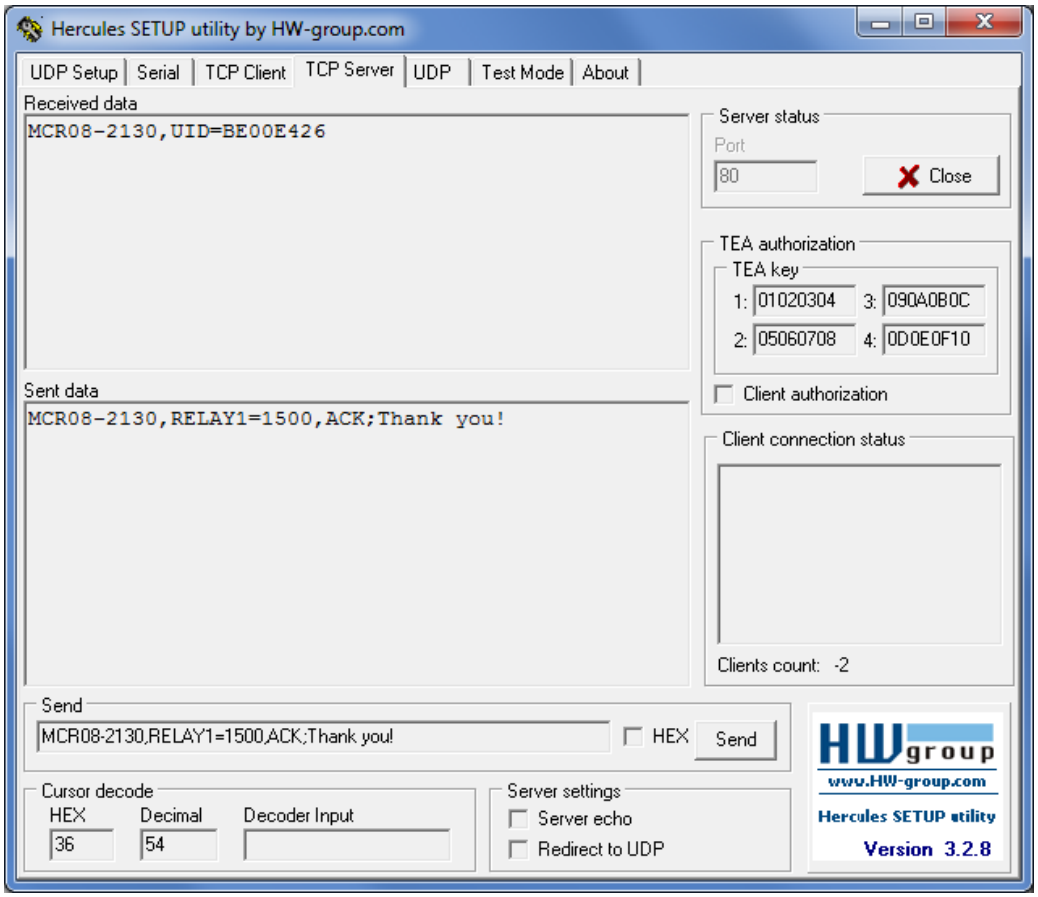
14 Test Connection with Hercules

Hercules Setup Utility can be used to test the terminals behavior.

- Set the terminal in Client mode (skip this step if the terminal is already in client mode)



- Select TCP Server and enter the Terminals port number
- Click on ***Listen***
- The terminal will connect automatically as seen in the connection status
- After presenting a card, the message will be displayed in the ***Received data*** window
- Enter the response message and send to the terminal. The device ID must be the same in the received and sent data
- As the TCP connection is open, we can send commands directly to the terminal

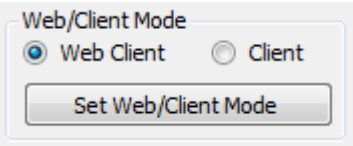


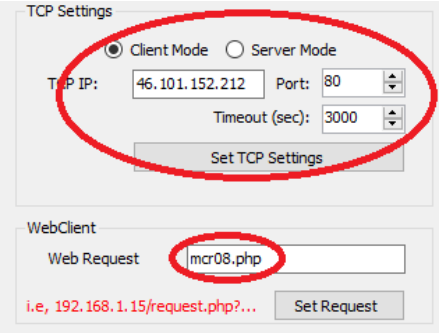
Use the ***TCP Client*** tab if the terminal is configured as a server.

15 Web-Client Mode Server Example

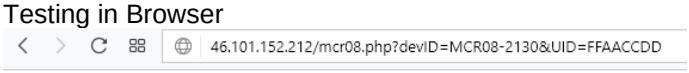
The following web-client setting is available to test the terminal quickly.

Test configuration





Testing in Browser



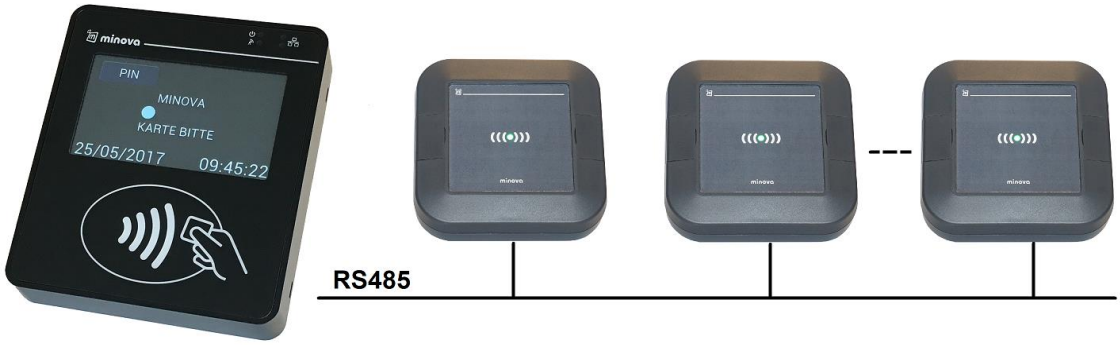
MCR08-2130,ACK;Valid card;Thank you,TSYNC=1503480703

PHP example code

```
<?php
	date_default_timezone_set('Europe/Berlin');

	$timeDiff = 2;		//Time diff, if server time differs from local time on site, set to 0 if server and device are at the same time zone
	$unixTime = time() + $timeDiff * 3600;			//Prepare time variable for responses

	$datetime = date("Y-m-d H:i:s");

	if(isset($_GET['devID']))
	{
		$devID = $_GET['devID'];
		if(isset($_GET['UID']))
		{
			$uid = $_GET['UID'];
			echo $devID.',ACK;Valid card;Thank you,TSYNC='.$unixTime.'';
		}

		if(isset($_GET['BUTTON']))
		{
			$button = $_GET['BUTTON'];
			if($button=='100')
				echo $devID.',MSG;Access Granted;Thank you,TSYNC='.$unixTime.'';
			else if($button=='101')
				echo $devID.',QUERY;Leaving now?,TSYNC='.$unixTime.'';
			else if($button=='102')
				echo $devID.',NOTIFY;Please call the secretary!;Thank you,TSYNC='.$unixTime.'';
			else if($button=='103')
				echo $devID.',REQUEST;Please enter the amount!,TSYNC='.$unixTime.'';
			else
				echo $devID.',MSG;Thank you,TSYNC='.$unixTime.'';
		}
		if(isset($_GET['QUERY']))
		{
			$query = $_GET['QUERY'];
			if($query=='1')
				echo $devID.',MSG;YES was clicked,TSYNC='.$unixTime.'';
			else
				echo $devID.',MSG;NO was clicked,TSYNC='.$unixTime.'';
		}
		if(isset($_GET['ENTRY']))
		{
			$entry = $_GET['ENTRY'];
			echo $devID.',MSG;Entry '.$entry.' is received!,TSYNC='.$unixTime.'';
		}

		if(isset($_GET['IN']))
		{
			$input = $_GET['IN'];
			echo $devID.',MSG;Input '.$input.' is received!,TSYNC='.$unixTime.'';
		}
	}
?>
```

16 MCRN2 Slave Reader Support

MCRN2 slave readers can be attached to the RS485 port. The default baud rate is 9600 Baud.



Each MCRN2 reader has a rotary switch (0-to-9) for RS485 addressing. The default address is ‘0’.

Communication Examples

Slave to Host UID Packet:

<Device ID>,<UID=XX>,<YY=XX>,<RS485=X>
UID: UID packet
YY: Optional other data
RS485: Reader network address (0-to9)

MCR08-1000,UID=5D2155D5,RS485=0 (RS485 data packet with ADR:0)

Host Command: <Device ID>,<RS485SND;PAR1;PAR2>
PAR1: Reader network address (0-to-9) PAR2: Command in HEX mode

Reply to Host Command (non-web mode): <Device ID>,<RS485RCV;PAR1;PAR2>
PAR1: Reader network address (0-to-9) PAR2: DATA in HEX mode (refer to MCRN2 cmd manual)

Command Examples:

ACK indication with double buzzer beep and blue led for 2 seconds
MCR08-1000,RS485SND;0;6000

NAK indication with single long buzzer beep and red led for 2 seconds
MCR08-1000,RS485SND;0;6001

Single BUZZER command
MCR08-1000,RS485SND;0;4201
4200 – short beep
4201 – double beep
4202 – long beep
4203 – low frequency beep
4210020200 – custom frequency and time <PERIOD, TIME (2 bytes MSB)>

LED Control
4101 – RED LED
4102 – GREEN LED
4104 – BLUE LED
4107 – All 3 LEDs on (white color)

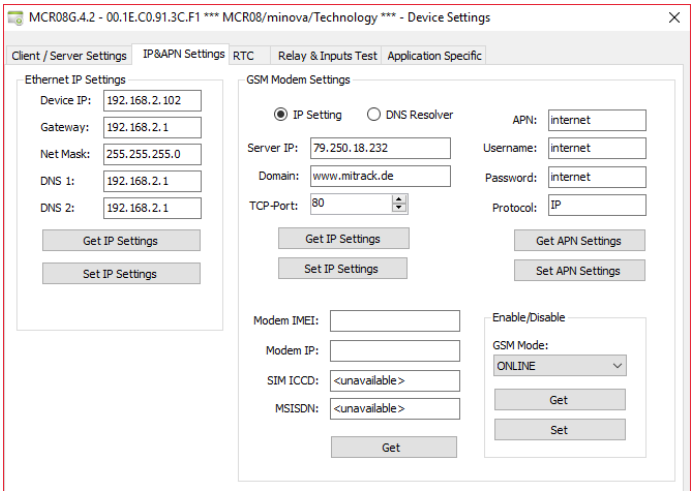
Changing slave configuration (send only once)
31001200018B – Enable automatic buzzer and LED indication
310012000109 – Disable automatic buzzer and LED indication

17 GSM Interface (MCR08G only)

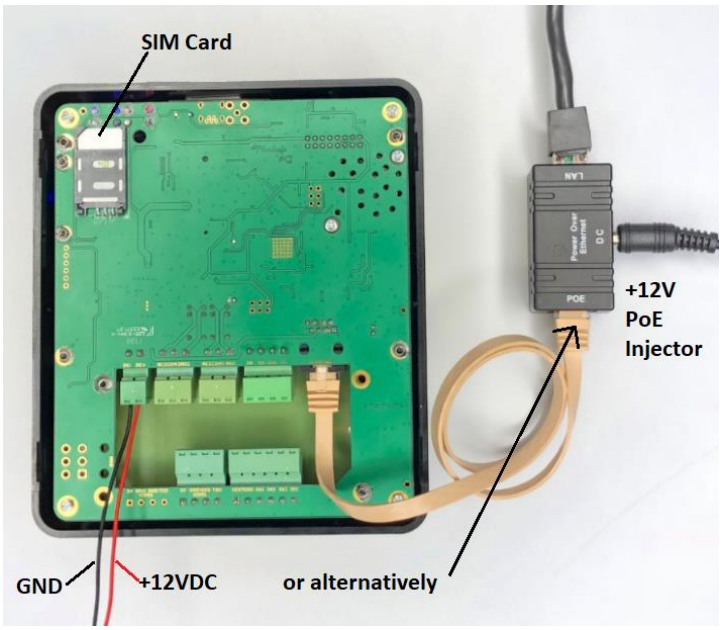
MCR08G is equipped with a Quad-Band 2G GSM Modem.

17.1 Configuring the GSM Modem

1- Attach the reader into the ***Ethernet*** network
2- Run miFinder.exe (V > 2.1)
3- Double click on the listed reader to open the device settings
4- Select ***IP&APN Settings*** tab



5- Enter the APN settings of your SIM card
6- Set the server ***IP/Port***
7- Set optionally the domain name of your server and select the ***DNS Resolve***r option. The MCR08 will automatically resolve the IP address of your domain during power-up.
8- Enable the GSM interface by setting the ***GSM Mode*** to ***ONLINE***. The GPRS connectivity is periodically checked in the ALIVE intervals. If the ALIVE message is disabled (by entering 0) the reader checks the GPRS connectivity each 60s and reconnects if necessary.



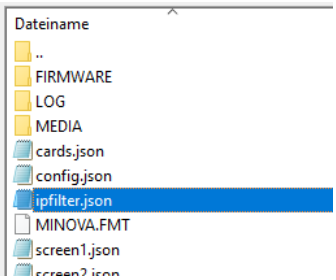
SIM Card location and power input alternatives

** At least a 2A power supply is needed for proper GSM connectivity*

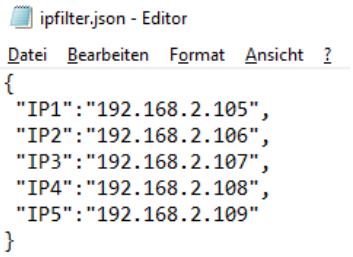
18 MCR08 IP-Filter Function

The MCR08 can be configured to enable IP filtering of TCP, UDP and FTP protocols. Maximum 5 IP numbers can be defined. These IP numbers must be defined in the ipfilter.json file. If the ipfilter.json file doesn't exists, all IPs will be accepted.

FTP-Client view



Sample json data



After power-up, IP filtering feature is deactivated.

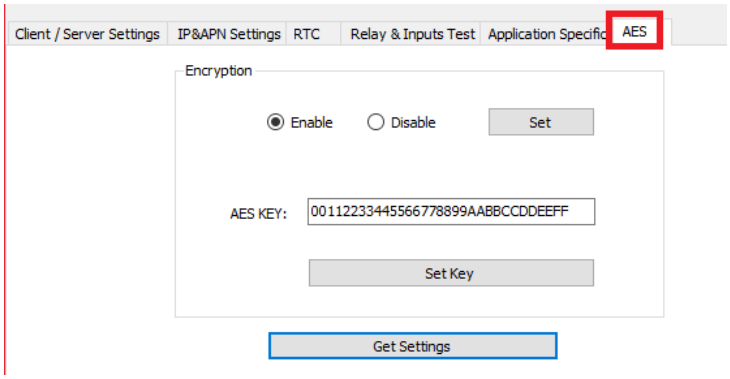
TCP command to activate IP filtering:
ENIPFILTER (*Example: MCR08-5000,*ENIPFILTER*)*

TCP command to deactivate IP filtering:
DISIPFILTER (*Example: MCR08-5000,* DISIPFILTER*)*

TCP command to delete this file:
REMFILE (*Example: MCR08-5000,REMFILE;ipfilter.json)*

19 Encrypted Client-Server Communication

Encrypted communication can be activated using the miFinder tool.



Cipher Type: AES **Mode:** CBC **Key Size:** 128 bits **Block Size:** 128 bits

Input data should be a multiple of the block size (16 bytes), so messages may have to be padded with 0x00 to bring them to this length.

Server-to-Client example: *CIPHERDATA+CRLF(0D0A)*

ASCII	MCR08-4CC0,ACK;THANKS
HEX	4d435230382d344343302c41434b3b5448414e4b530000000000000000000000 (padded)
KEY	00112233445566778899AABBCCDDEEFF
IV	00000000000000000000000000000000
CIPHER	D13DFD1B9BB3117B83D76357A3D195713DD7947D2C03D4B9B04132B3C5444F5C

Client-to-Server example: *CIPHERDATA+CRLF(0D0A)*

CIPHER	885C685AFD3DB684D4D0F1E7AA03B9369F24048D442F60E82FB1E75E5E43BCE7*0D0A*
KEY	00112233445566778899AABBCCDDEEFF
IV	00000000000000000000000000000000
HEX	4D435230382D344343302C5549443D4532384336394142000000000000000000
ASCII	MCR08-4CC0,UID=E28C69AB

19.1 IV Initialization Vector

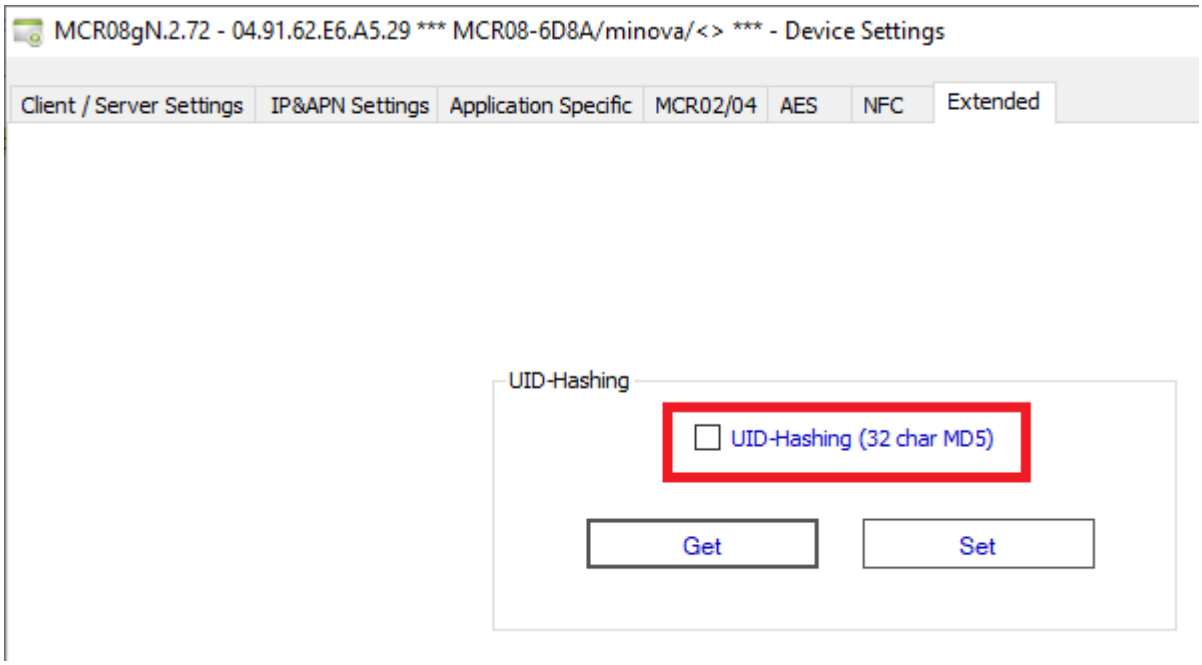
The initialization vectors are randomized and send to the server (in plain text) at the beginning of each new TCP session.

Example:
MCR08-4CC0,IV=903FA4E02A8931A55D4D0FF888BBCBFF

During the TCP session, all cipher blocks are chained with their own IVs (RX and TX). The initial IVs are the same.

20 UID Hashing

UID hashing function enables hashing of all UIDs sent to the host. The hash output will be nonreversible and adds a security to the system.



Example:

Standard UID sending

MCR08-1000,UID=04284D82FB4380

Hashed UID sending

MCR08-1000,UID=F8F34D899EE3494D512FFB9B6CC6A913

Following hash algorithm is used

128 Bit Message-Digest Algorithm 5 (MD5)
32 Char HEX value

21 RS232/RS485 Interface

The serial interfaces may be used as the main communication interface. In applications with cable length >3m, RS485 interface is recommended.

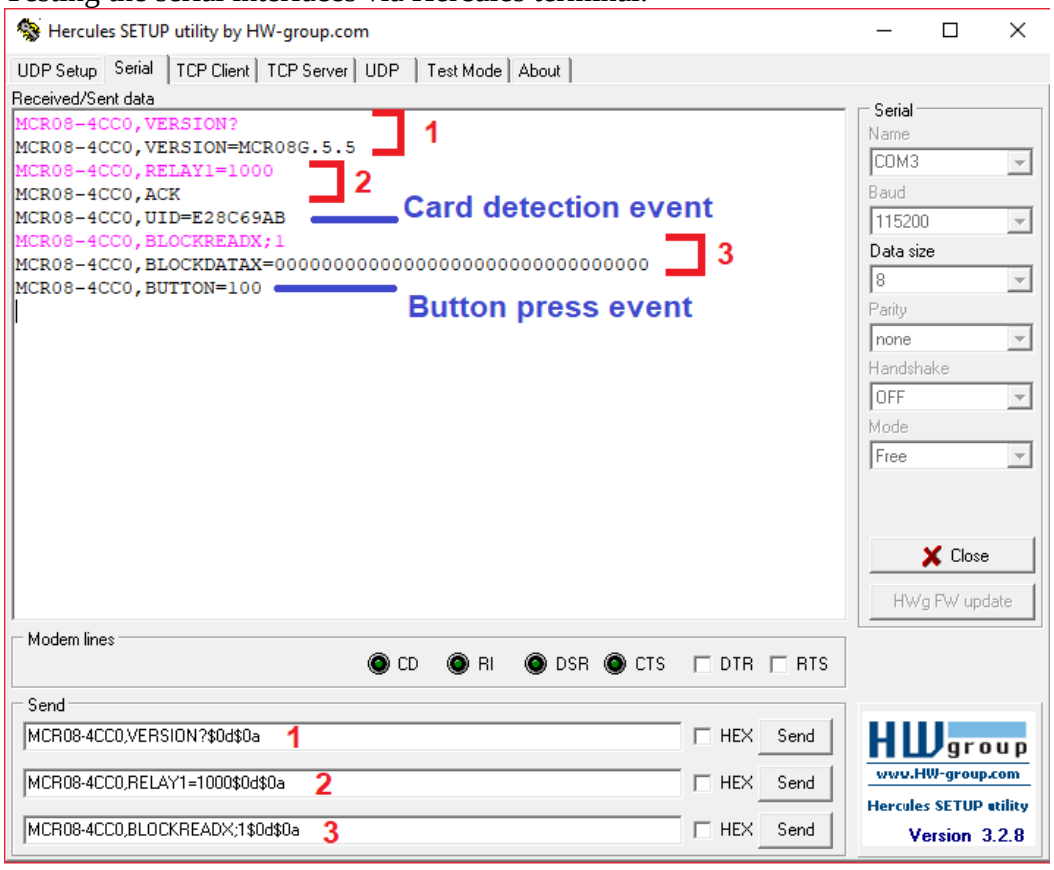
Default Settings:
RS232/COM1: 115200 bps, 8N1 (8 bit - no parity - 1 stop bit)
RS485: 9600 bps, 8N1 (8 bit - no parity - 1 stop bit)

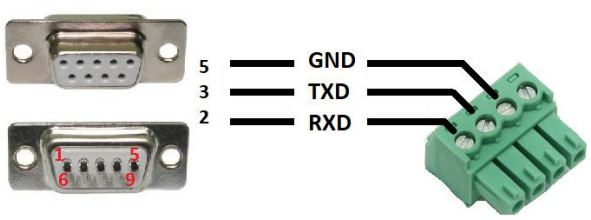
Frame format:
MCR08-XXXX,<DATA><CRLF>

Examples:
MCR08-4CC0,VERSION?*0D0A*
MCR08-4CC0,VERSION=MCR08G.5.5*0D0A*

MCR08-4CC0,PINPAD;Please enter the amount*0D0A*
MCR08-4CC0,PINPAD=123*0D0A*

Testing the serial interfaces via Hercules terminal.



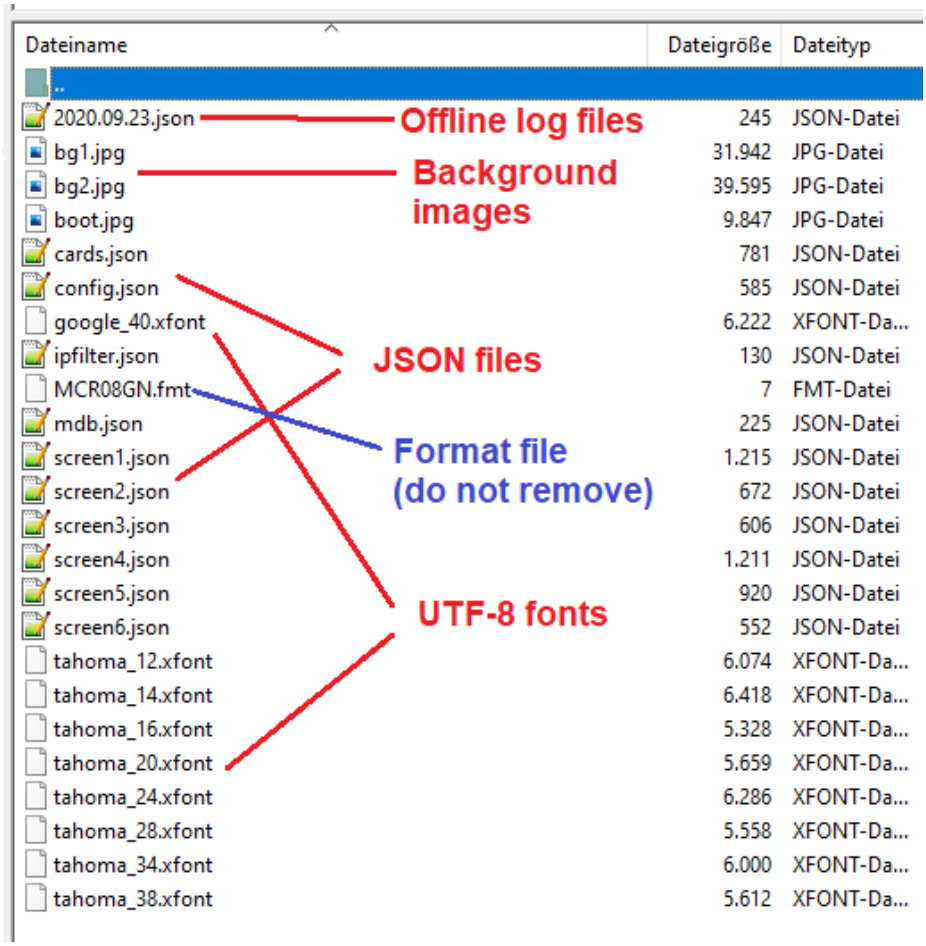


RS232 connector pinning

22 MCR08N Features

The new version of the MCR08 supports UTF-8 coding and the all files are stored in a NAND-Flash memory. No sub folders are allowed.

Flash content via FTP-Client



23 Sound Synthesizer

A sound processor, AUDIO ENGINE, generates the sound effects from a small ROM library of wave stable. To play a sound effect listed in the below table, send the SOUND command with the parameter.

Value	Effect	Conti nuous	Pitch adjust
00h	Silence	Y	N
01h	square wave	Y	Y
02h	sine wave	Y	Y
03h	sawtooth wave	Y	Y
04h	triangle wave	Y	Y
05h	Beeping	Y	Y
06h	Alarm	Y	Y
07h	Warble	Y	Y
08h	Carousel	Y	Y
10h	1 short pip	N	Y
11h	2 short pips	N	Y
12h	3 short pips	N	Y
13h	4 short pips	N	Y
14h	5 short pips	N	Y
15h	6 short pips	N	Y
16h	7 short pips	N	Y
17h	8 short pips	N	Y
18h	9 short pips	N	Y
19h	10 short pips	N	Y
1Ah	11 short pips	N	Y
1Bh	12 short pips	N	Y
1Ch	13 short pips	N	Y
1Dh	14 short pips	N	Y
1Eh	15 short pips	N	Y
1Fh	16 short pips	N	Y
23h	DTMF #	Y	N
2Ch	DTMF *	Y	N
30h	DTMF 0	Y	N
31h	DTMF 1	Y	N

Value	Effect	Conti nuous	Pitch adjust
32h	DTMF 2	Y	N
33h	DTMF 3	Y	N
34h	DTMF 4	Y	N
35h	DTMF 5	Y	N
36h	DTMF 6	Y	N
37h	DTMF 7	Y	N
38h	DTMF 8	Y	N
39h	DTMF 9	Y	N
40h	harp	N	Y
41h	xylophone	N	Y
42h	tuba	N	Y
43h	glockenspiel	N	Y
44h	organ	N	Y
45h	trumpet	N	Y
46h	piano	N	Y
47h	chimes	N	Y
48h	music box	N	Y
49h	bell	N	Y
50h	click	N	N
51h	switch	N	N
52h	cowbell	N	N
53h	notch	N	N
54h	hihat	N	N
55h	kickdrum	N	N
56h	pop	N	N
57h	clack	N	N
58h	chack	N	N
60h	mute	N	N
61h	unmute	N	N

The sound parameter should be entered as decimal value.

Example:

MCR08-1234,SOUND;80*0D0A* // Click sound

24 Whitelist Synch Function

The whitelist file “cards.json” can be updated using the following sequence.

Sample file to send

cards.json

```
{
    "whitelist ":{
    "cards":[
        { "id":"FAE6A24C", "profile":"0","name":"Max Mustermann", "nr":"45665"},
        { "id":"7A145632", "profile":"1","name":"Barbara Scott", "nr":"12346"},
        { "id":"AC4A44D5", "profile":"1","name":"John Winter", "nr":"12225"},
        { "id":"5CD54ED5", "profile":"1","name":"Nina Cloud", "nr":"445588"},
        { "id":"55898999", "profile":"1","name":"Parcel 1", "nr":"445589"},
        { "id":"12121554", "profile":"1","name":"Parcel 2", "nr":"445590"}
        ]
    }
}
```

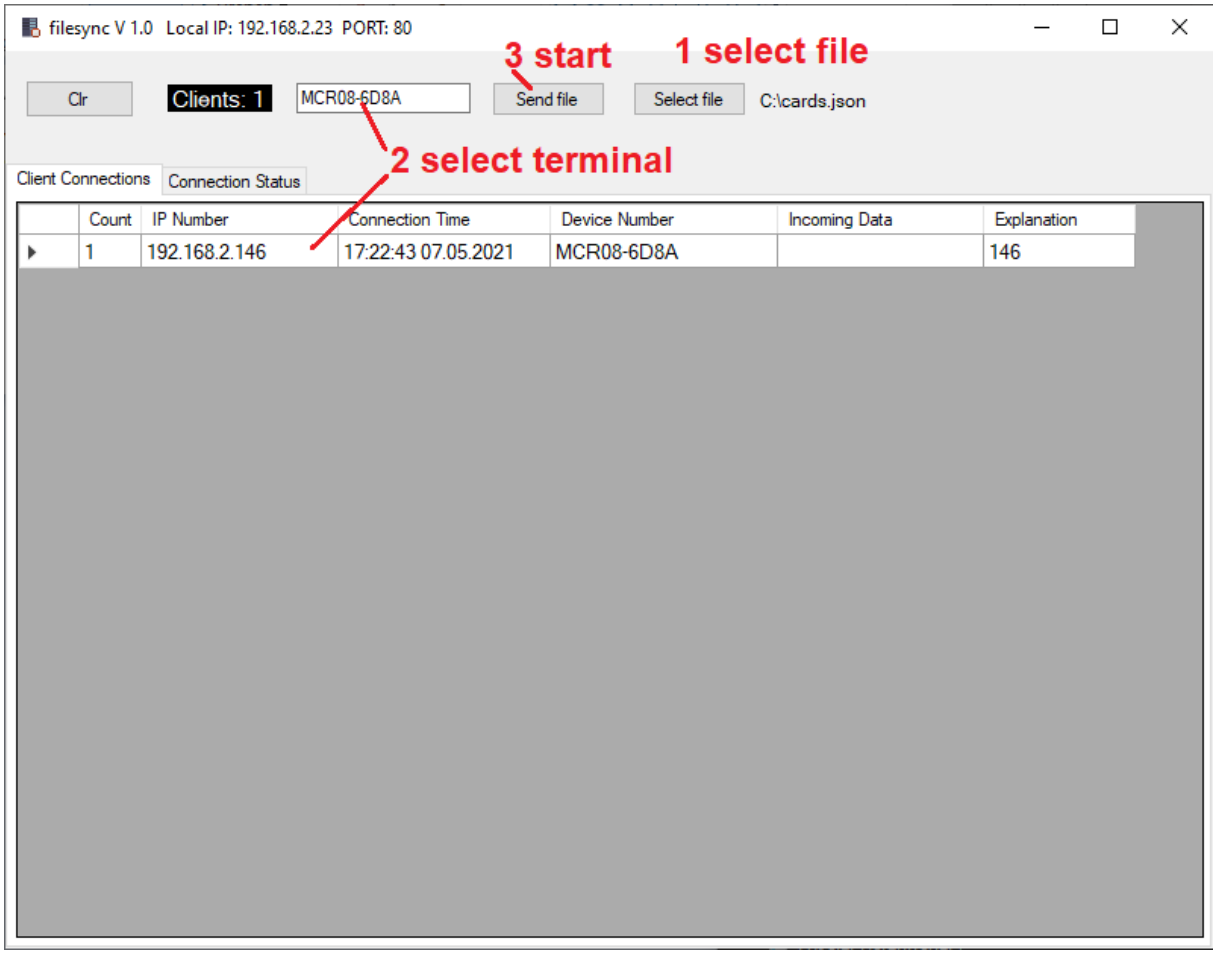
Send the file synch command to start
MCR08-4CC0, FILESYNC

The terminal sends following json frame to request the new file line by line
{"sync":"req","seq":"0","ip":"192.168.2.146","md5":"39623dc4cb5c557f18aea67a07248573"}
The Server may check the md5 of the current file and start sending the new file line by line

Server: *{*
Terminal: *{"sync":"ack","seq":"1"}*
Server: *"whitelist ":{*
Terminal: *{"sync":"ack","seq":"2"}*
Server: *"cards":[*
Terminal: *{"sync":"ack","seq":"3"}*
Server: *{ "id":"FAE6A24C", "profile":"0","name":"Max Mustermann", "nr":"45665"},*
Terminal: *{"sync":"ack","seq":"4"}*
Server: *{ "id":"7A145632", "profile":"1","name":"Barbara Scott", "nr":"12346"},*
Terminal: *{"sync":"ack","seq":"5"}*
Server: *{ "id":"AC4A44D5", "profile":"1","name":"John Winter", "nr":"12225"},*
Terminal: *{"sync":"ack","seq":"6"}*
Server: *{ "id":"5CD54ED5", "profile":"1","name":"Nina Cloud", "nr":"445588"},*
Terminal: *{"sync":"ack","seq":"7"}*
Server: *{ "id":"55898999", "profile":"1","name":"Parcel 1", "nr":"445589"},*
Terminal: *{"sync":"ack","seq":"8"}*
Server: *{ "id":"12121554", "profile":"1","name":"Parcel 2", "nr":"445590"}*
Terminal: *{"sync":"ack","seq":"9"}*
Server: *]*
Terminal: *{"sync":"ack","seq":"10"}*
Server: *}*
Terminal: *{"sync":"ack","seq":"11"}*
Server: }
Terminal: *{"sync":"ack","seq":"12"}*

File synch is finished, nothing to send else.
After 5-10 seconds, the file will be closed automatically.

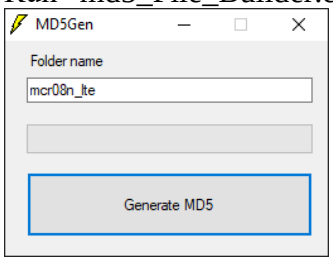
File synch software tool.



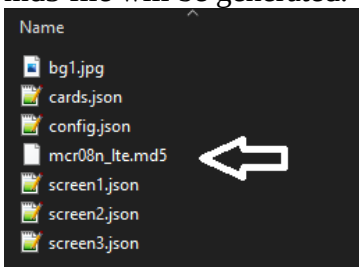
25 File Synch via FTP (LTE Variant only)

MCR08N can synchronize its files via FTP. The MD5 list of all files should be calculated.

Run “md5_File_Builder.exe” and process the related folder.



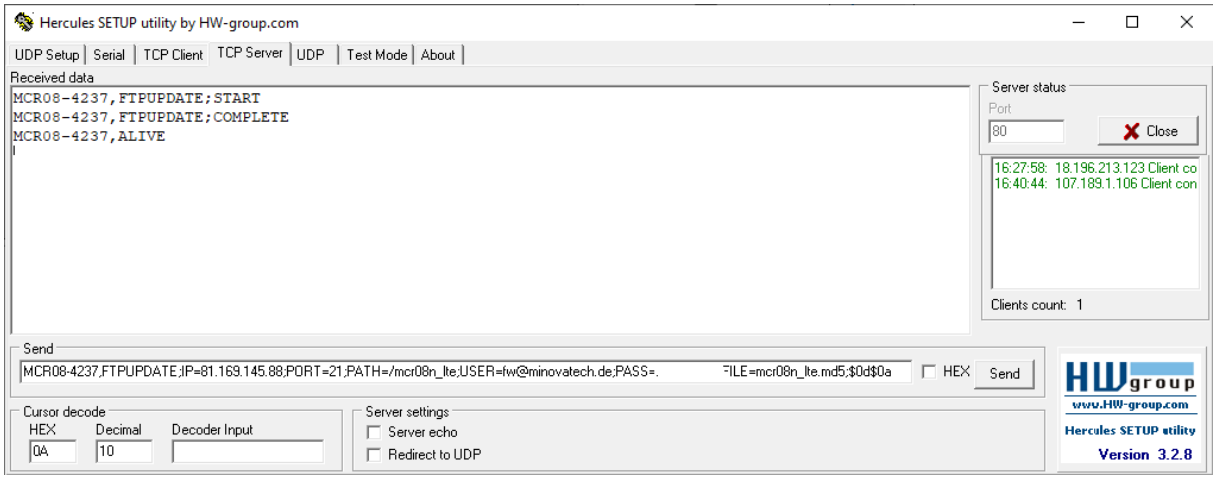
md5 file will be generated.



This file contains all filenames with md5 hashes. The terminal will check and download only changed files.

{ "name":"bg1.jpg", "size":"11184", "path":"", "md5":"29c4a0eff59a17ed35b8628269285777" }
{ "name":"cards.json", "size":"335", "path":"", "md5":"b1fb7640774282b2d61372da31dbc317" }
{ "name":"config.json", "size":"585", "path":"", "md5":"c16827d9820fdbbe63206cf1aa1d60e7" }
{ "name":"screen1.json", "size":"2122", "path":"", "md5":"e14ddaeca8098e3ed89d53a2ccb9ebb5" }
{ "name":"screen2.json", "size":"1135", "path":"", "md5":"5cb155e229c4593dc81272eb5976945a" }
{ "name":"screen3.json", "size":"1137", "path":"", "md5":"167815f508da2da30ddcd71e19ad56d4" }

Command to start synchronization (enter your credentials)
MCR08-1000,FTPUPDATE;IP=81.169.145.88;PORT=21;PATH=/mcr08n_lte;
USER=fw@minovatech.de;PASS=xxxxx;FILE=mcr08n_lte.md5;



Example with hercules