Description of the System Menu / Setup
in programmable CAN display terminals
by MKT Systemtechnik
Version 2.6

System Menu and Setup Options for User Programmable Displays

(UPT, HBG, MKT-View II / III / IV)

This document is available in german and english language .
Dieses Dokument ist in deutscher und englischer Sprache verfügbar .

Document-Nr: 85115
Original : C:\CBProj\UptWin1\DOKU\art85115_Sysmenu_Setup_EN.odt
This document is part of the INSTALLATION PACKAGE of the Terminal Programming Tools, folder "Doku" .

Contents

1 Revision History

Versions-nummer	Datum (ISO8601)	Autor	Hinweise, Änderungen
V 1.0	2003-05-28	W.Büscher	Creation of the document (in German language)
V 1.1	2003-08-01	W.Büscher	Liste der zulässigen CAN-Baudraten geändert
V 1.2	2003-10-27	W.Büscher	div. Erweiterungen, Hinweis zum LOGGER
V 1.3	2003-12-09	W.Büscher	Freischaltung von Sonderfunktionen beschrieben
	2007-04-23	W. Büscher	Umstieg von Winword auf OpenOffice, Master-Datei nun *.odt (OpenDocumentText) statt *.doc, Export als "tagged PDF" mit Hyperlinks .
V 1.4	2008-09-11	W. Büscher	Ergänzungen für "MKT-View II"
	2008-12-02	W. Büscher	Ergänzungen zu schaltbaren CAN-Abschluss-Widerständen und "Wakeup-per-Digitaleingang"
	2009-02-02	W. Büscher	First translation into English language, filename art85115_Sysmenu_Setup_EN.odt/pdf .
V 1.5	2009-10-07	W. Büscher	Added "High-Voltage Programming Mode" / Remote Control operation
V 1.6	2010-12-22	W. Büscher	Added description of the "low power" brightness, which replaces the 'automatic light-off' function for devices with a TFT display (instead of STN).
V 1.7	2011-03-29	W. Büscher	Added the user-defineable CAN-Bus Parameters
V 1.8	2012-04-27	W. Büscher	Notes about touchscreen usage, and gesture control
V 1.9	2012-07-11	W. Büscher	Added the TRACE-display (for developers)
V 2.0	2013-07-29	W. Büscher	Described the 'Shutdown on excess temperature'
V 2.1	2014-01-14	W.Büscher	Described audio volume option 'Beeper to Speaker'. Replaced the 'bullet' character, because it appeared as '3517' in the PDF generated by OpenOffice.
V 2.2	2014-08-18	W. Büscher	Added links to clarify option 'CAN-TxEnable'
V 2.3	2014-12-10	W. Büscher	Modified links to the MKT website
V 2.4	2015-07-29	W. Büscher	VNC server port numer in the 'Network Setup'
V 2.5	2016-05-11	W. Büscher	Remote Display using 'VNC-via-UDP'

2 Preface

This document describes the structure and the operation of the "system menu", which is used in a large variety of display terminals by MKT Systemtechnik. Members of this family of programmable devices are the "UPT" series (UPT= user programmable terminal), and the display known as MKT-View and its successors. Traditionally, the UPT's have CANopen protocols implemented in the firmware, while the MKT-View family uses parametrisation from CANdb files (*.DBC) - the latter mainly used in the automotive industry for passenger cars.

MKT-View III

MKT-View IV
- for CANopen or "CANdb"
- 7" TFT, 800*480 pixel, 65535 colours
- resistive touchscreen
- 200 MHz CPU with floating point unit
- 2 * CAN, Ethernet, RS-232, USB

MKT-View IV

HBG 18

HBG 22

HBG 35

UPT 800

All these devices are programmable. The "user program"(written by the user, most likely using one of MKT's programming tools) will be called *display application* throughout this document, to emphasise the difference between "user program" and the firmware. The system menu is implemented in the display terminal's *firmware*, not in the *display application*.

Further information about the programmable display terminals can be found here:

- Help system for the UPT Programming Tool (online link to a copy of the HTML documents)
- Manual for the UPT programming tool (Doc.-Nr. 85110)
- Additional information "CANdb" (Doc.-Nr. 85111) for Terminals with CANdb (DBC) instead of CANopen
- Details about the integrated HTTP Server, available in devices with Ethernet interface.
  Can be used as an alternative to configure the devices (instead of the system menu).
  If the device is already online (in your LAN) with the default hostname (UPT), try this link: http://UPT .

An up-to-date version of the programming tool can be found in the download area at www.mkt-sys.de . After installation of the programming tool, you can find *this* document in a folder named "Doku".

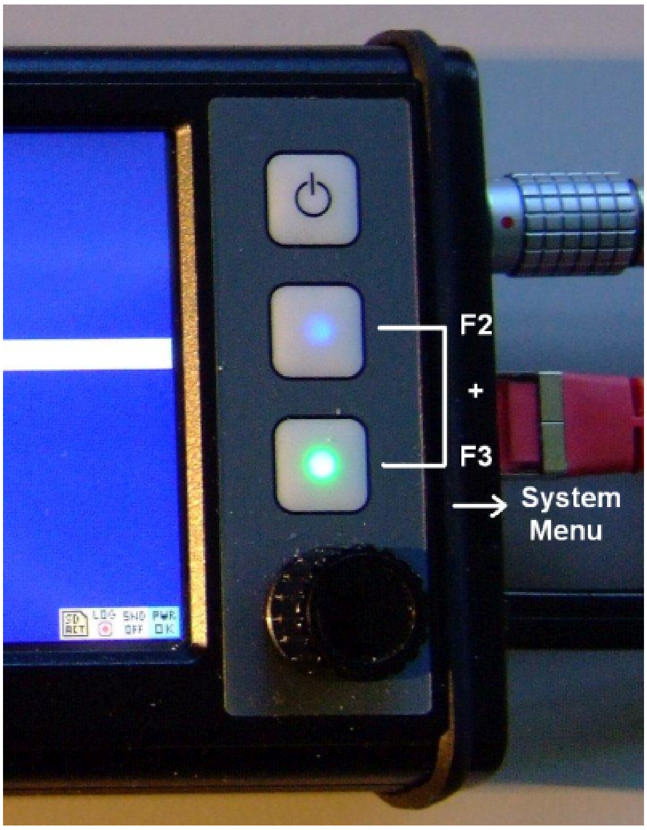
3 Invocation and operation of the system menu

3.1 Invocation of the system menu

The keyboard combination to invoke the system menu depends on the hardware. Generally, "the second and the third function key" must be held down during power-up (or, sometimes, during normal operation). For UPTs with function keys labelled F1 through F6, these keys were

F2 + F3 .

In the "MKT-View", F2 and F3 are the second and third (non-labelled) function key, counting from top to bottom. The function key F3 is close to the rotary encoder knob. We will refer to these keys as "F2" and "F3" throughout this document.

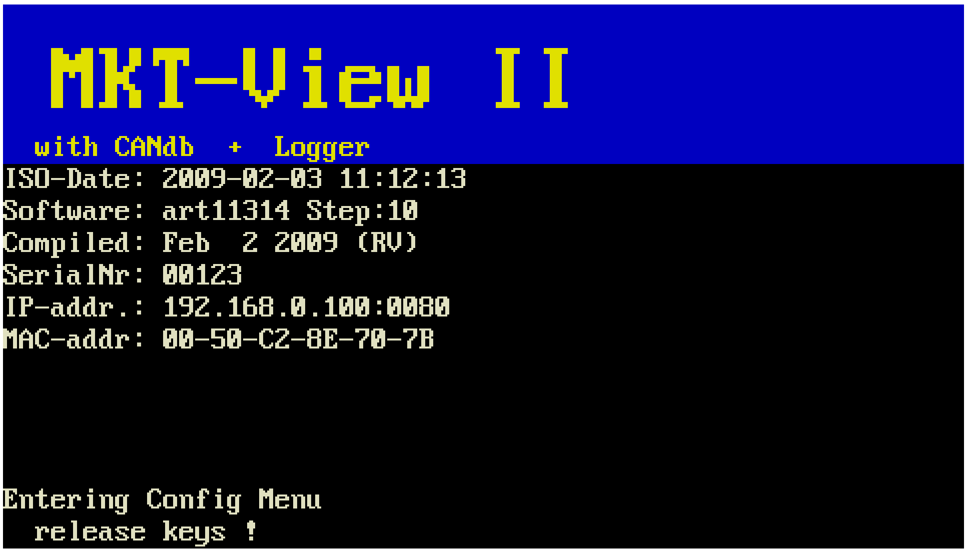


For devices without a keyboard, the invocation of the system menu will be described in a later, hardware-specific chapter (Virtual Keyboard, or via remote control, etc). If everything else fails, you can even invoke and operate the system menu via the 'service key' (on some devices).

Depending on the configuration of the application (which has been loaded into the terminal by the user), it may be possible to invoke the system menu during normal operation. If that is not possible (because someone has disabled access to the system menu in the programming tool), you can still invoke the system menu by holding the keys pressed during power-up, i.e. before the terminal loads the configuration data from an internal FLASH file.

While holding down the keys to activate the system menu, the following info is displayed on the terminal's screen :

- the name of the device (like "UPT-515", "UPT-167", "MKT-View", "MKT-View II")
- the software acticle number of the firmware, which has been loaded into the device, including the version number (major,minor), revision- and build number.
- the compilation date of the firmware.

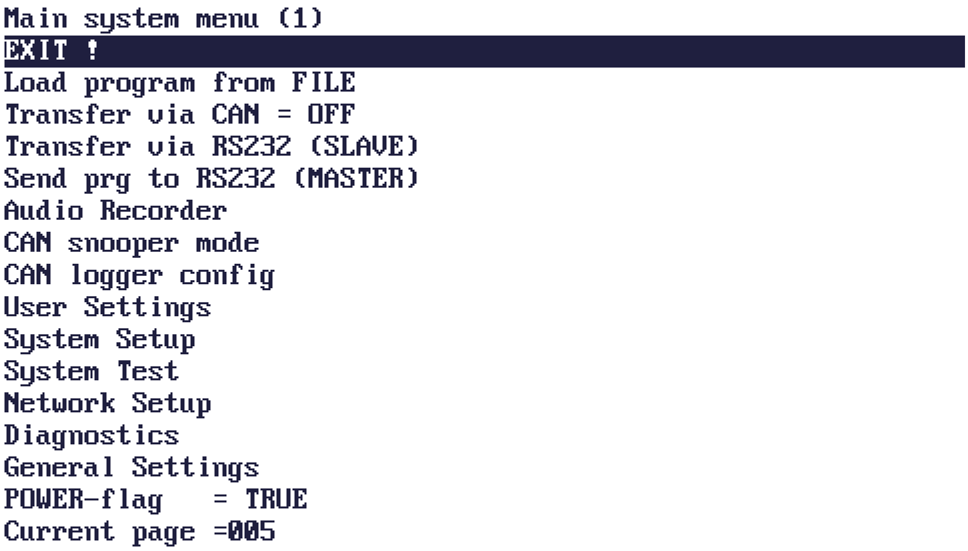


```
 MKT-View II
 with CANdb + Logger
ISO-Date: 2009-02-03 11:12:13
Software: art11314 Step:10
Compiled: Feb  2 2009 (RV)
SerialNr: 00123
IP-addr.: 192.168.0.100:0080
MAC-addr: 00-50-C2-8E-70-7B



Entering Config Menu
  release keys !
```

This info is important to identify a device, and to decide if (and how) the firmware should be updated. There is an overview of all firmware files suitable for a certain device on the MKT Server (internal), or in a certain directory within the programming tool - see ..\firmware\readme.txt .

3.2 Operation of the system menu

If the keyboard has cursor keys, you can use the up/down key to select the current menu item. The ENTER key (if present) executes the selected function, or toggles beween "Navigate" and "Edit" mode of the system menu.
In "Navigate"-mode, an entire line of the menu is marked. In "Edit"-mode, only a single digit or character is marked (the one which is currently being edited).



```
Main system menu (1)
EXIT !
Load program from FILE
Transfer via CAN = OFF
Transfer via RS232 (SLAVE)
Send prg to RS232 (MASTER)
Audio Recorder
CAN snooper mode
CAN logger config
User Settings
System Setup
System Test
Network Setup
Diagnostics
General Settings
POWER-flag   = TRUE
Current page =005
```

In devices without cursor keys, a rotary encoder button may be avaiable to navigate through the menu (turn the knob) and execute a certain function (press the knob == ENTER). So usually, turning the knob replaces the cursor UP/DOWN function. If, in certain situations, the cursor shall be moved LEFT/RIGHT, a fast double-click with the encoder toggles the cursor function between UP/DOWN and LEFT/RIGHT.
In terminals with a touchscreen, a double-click with the encoder button opens a virtual keypad with digits and virtual cursor keys. The virtual keyboard is explained in chapter 11.2 .

For devices which have neither a keyboard, nor an encoder button, nor a touchscreen, the only chance to use the build-in system menu is via remote control. Details on that are in chapter

3.3 Operation of the system menu via 'Service Key' (if everything else fails...)

In a few devices (like "UPT 800"), there is neither a keyboard, nor an encoder, and possibly not even a properly working touchscreen (after replacement or repair). In such situations, the system menu can be invoked and operated through a special 'Service Key'. In the UPT-800, this key is a miniature surface-mounted, pushbutton located on the main board, close to the memory card slot. Depending on the housing, you may need a sharp object (ball pen, screwdriver, etc) to push this key.

Since a single key cannot replace three function keys, cursor key, ENTER and ESCAPE, the 'service key' is connected to a Morse decoder. Pushing the key for less than 0.25 seconds is considered a dot (or letter "e" in Morse code), pressing it for longer than 0.25 creates a dash. The characters in Morse code consist of one to five dots or dashes. The following table contains the most important codes, and their functions in combination with the terminal's system menu.

Morse code	Character	Function in the system menu
..--..	? (questionmark)	Invocation of the system menu per 'service key'
...-.-	sk (silent key)	Shutdown, oder Shutdown-Menü, or immediate Power-Off
...-	v	Virtual Keyboard (opens the virtual keyboard)
.-	a	Replacement for function key "F1"
-...	b	Replacement for function key "F2"
-.-.	c	Replacement for function key "F3"
-..	d	Down (replaces cursor key or rotary encoder)
.	e	also replaces 'cursor down', after invoking the system menu
..-	u	Up (replaces cursor key or rotary encoder)
..-.	q	Quit (replacement for 'Quit', 'Cancel'-button, or 'Escape'-key)
.-.	r	Return (replacement for the 'Enter'-key, or 'Ok'-button)
-	t	also replaces Enter / 'Ok' after invocation of the system menu

4 Functions in the System Menu

Note: This documentation is universally valid, i.e. it refers to a wide range of products with very different hardware capabilities (especially display and keyboard). But *in principle*, the functions in the system menu are the same for all devices.

In devices with a 'huge number' of parametrizeable properties, some functions may have been moved into submenus to keep the main menu pages clean (for example, "LCD..." , "CAN ..." in the Setup Menu). This way vertical scrolling should be unnecessary while navigating through the menus. The following chapters don't explicitly mention this, so if you don't find an item "LCD Contrast" in the Setup menu, look for an item titled "LCD...", select it, and you will find the "LCD Contrast" item *in that submenu* .

5 Main level ("Main system menu")

Functions / Items in this menu :

Exit

Leaves the system menu, and returns to the *display application* .

Load program from file

Only for devices with a FLASH memory card (CF or SD). When avaiable, this function is the fastest method to copy the display application (*.upt or *.cvt, created with the programming too) into the display terminal. It opens a file selector box, from which you can select a display application. Confirm the selection with the ENTER key, and wait until the file has been copied into the terminal's onboard FLASH memory.
This is much faster than transferring the application via CAN or RS-232.

Up/Download via CAN or Transfer via CAN = ON/OFF

Use this function to transfer the display application from (or to) the PC via CAN-bus. This is only required for devices without CANopen (in CANopen terminals, the SDO protocol is used to transfer the application - so there's no need to enter "transfer mode" manually).
In devices with CANdb support (i.e. without CANopen), a certain pair of CAN Identifiers (usually 0x7F0 / 0x7F1) will be used to establish a communication between PC and display terminal. To avoid ID collisions during normal operation, the terminal will not react on reception of a message with this identifer, and will never attempt to send a message with this identifier (i.e. "Transfer via CAN = OFF").
Details about the download protocol are available in document #85116 .

Other Transfers.... (in terminals with firmware since 2009-11-05 or later)

Open this submenu to transfer the program through some less frequently used protocols, for example 3964R via RS-232, XMODEM / YMODEM via RS-232. Some of the possibilities are listed in the next paragraphs.
For IP-based protocols (like the simple file transfer mode for the programming tool), it's not necessary to enter transfer mode through the system menu. Details about file transfer via

Ethernet (LAN) can be found in the online help system of the programming tool, details in 'filetransfer_49.htm' (german) or 'filetransfer_01.htm' (english).

Up/Download via RS232 bzw. Transfer via RS232 (SLAVE)

This function may be invoked to establish a connection between the terminal and the programming tool (PC) via the serial port. This is necessary because in some terminals, the serial port may be occupied for a different purpose (for example, connection of a GPS receiver). If the serial port is ***exclusively*** used for programming, the manual activation of the serial port may be unnecessary. In that case, this menu item doesn't exist.

Receive from RS232

Could be used in some devices to "clone" two terminals through the serial port, connected via Null-Modem cable. In one terminal, this function must be started; in the other terminal, start the function "Send to RS232". One of the two partners may as well be the terminal programming tool (running on a PC with a serial port, which is not too common these days...)

Send program to RS232

See "Receive from RS232".

CAN snooper mode

Only available in display terminals with built-in CAN-LOGGER(!) . A detailed description of the CAN Logger / Snooper can be found in a separate document (part of the online help system, or doc #85118 in the logger utility's "Doku" folder, available on your harddisk after installation of the tool) .

CAN logger mode

Only available in display terminals with built-in CAN-LOGGER. Explained in a separate file (see above) .

User Settings

Numerous settings for the end user, most notably the CAN baudrates, and the non-volatile values ("nv"). See Chapter 6.

System Setup or System Setup / UNLOCK

Various system settings (hardware related), some of them also important for the end user. Details in chapter 7. In some devices, this menu also contains the feature to unlock special functions by entering a numeric key. Details on that in chapter 8.1.

System Test

Hardware test functions for various interfaces, I/O ports, etc. Details in chapter 8.3.

Network Setup

Only for devices with Ethernet port and TCP/IP protocol stack . See chapter 8.4 .

Diagnostics

Originally "only intended for use by the developer / manufacturer", for diagnostic purposes, before the advent of a real target debugger ;-) . But the diagnostic menu contains some information about the firmware compilation settings, for example the maximum number of display pages, the maximum number of display elements per page, etc.

General Settings

Can be used to check the configuration, which is usually loaded into the terminal from the programming tool (so there's no need to set this on the terminal). Most of the parameters displayed here (in the terminal) are read-only, for example the "number of strings", etc.

Current Page...

Only for debugging. The 'current page' item shows the ***NUMBER*** (array-index) of the current display page, not the current display page itself ;-). For testing purposes, you can modify the page number here (the display application will jump to that page when leaving the system menu).

6 User Settings

In contrast to the parameters in the System Setup, the *User Settings* will be lost (overwritten) when loading a new display application (via programming tool or memory card). Despite that, these parameters can be modified "on the device" (without a programming tool). To modify them, select the item *User Settings* in the programmable device's Main System Menu .

Exit

Leave this menu without saving modified parameters permanently in EEPROM .

Save & Exit

Save changes (if any) in EEPROM, and leave this menu.

CAN Baudrate 1

Baudrate for the first CAN bus interface ***during normal operation*** (in contrast to a similar entry in the System Setup, which will be used if no user configuration is loaded).
Allowed CAN baudrates for most programmable terminals are:
50,83,100,125,250,500,667,800,1000 (kBit/second)
In most terminals, this menu item is implemented as a combo box, so you can only select valid baudrates. Only in very old devices (UPT-515 for example), the baudrate was entered digit-by-digit as a decimal number.
As long as no valid user settings are available, the CAN bus will be initialized with the CAN baudrate configured in the System Setup .

CAN Baudrate 2

Baudrate for the second CAN bus interface ***during normal operation .***

More CAN Settings..

Opens a sub-menu with more advanced CAN parameters (like Sync Jump Width and Sampling Point). Available since 2011-03-28, but only for new devices with 32-bit CPU. Details in the chapter about user-defined CAN Bus Settings .

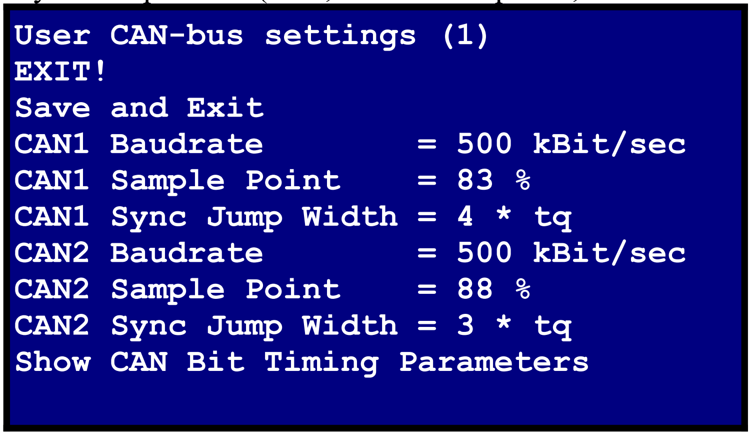
nv[0] ...

Non-volatile numeric values for the application. Each element in this array is a 32-bit integer number. The default contents are usually set (defined) in the programming tool. If you don't have access to a PC, you can change these values here. The UPT application accesses these values through the interpreter command / function "nv[index]". As usual, details about the UPT display interpreter can be found in the online help system of the programming tool.

6.1 User-defineable CAN Bus Settings (part of the User Settings)

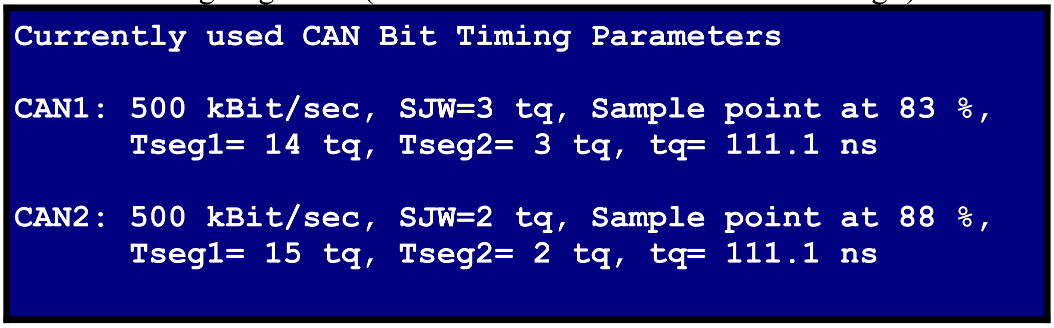
Besides the CAN baudrate, the User Settings include:

- the CAN sample point (by default, somewhere between 80 and 83 percent)
- the Sync-Jump-Width (SJW, 1 to 4 time quanta, default = 1 tq)



```
User CAN-bus settings (1)
EXIT!
Save and Exit
CAN1 Baudrate         = 500 kBit/sec
CAN1 Sample Point     = 83 %
CAN1 Sync Jump Width  = 4 * tq
CAN2 Baudrate         = 500 kBit/sec
CAN2 Sample Point     = 88 %
CAN2 Sync Jump Width  = 3 * tq
Show CAN Bit Timing Parameters
```

A summary of the currently used CAN bit timing parameters is displayed under "Show CAN Bit Timing Registers" (menu item in the 'User CAN-bus settings') :



```
Currently used CAN Bit Timing Parameters

CAN1: 500 kBit/sec, SJW=3 tq, Sample point at 83 %,
      Tseg1= 14 tq, Tseg2= 3 tq, tq= 111.1 ns

CAN2: 500 kBit/sec, SJW=2 tq, Sample point at 88 %,
      Tseg1= 15 tq, Tseg2= 2 tq, tq= 111.1 ns
```

There may be some minor differences between the 'wanted' (configured) timing parameters and the 'realizeable' ones, depending on the hardware. See example below.

Example: Baudrate = 500 kBit/sec, ideal SJW=4, 'wanted' Sample Point at 83.3 percent. CAN controller clocked with 36 MHz (MKT-View II), divided by four ("BRP"), which allows 18 time quanta. Each time quantum is 111.111 ns. To get close to the 'wanted' sample point, the firmware uses Tseg2=3 time quanta. Because the sync jump width (SJW) must never exceed Tseg2, the firmware uses SJW=3 * tq, even though the configured value is SJW=4. Note that all these calculations are highly hardware-specific, thus this is just an **example**.

Note: For safety reasons, modifying the CAN bit timing parameters will not have an immediate effect. Use 'Save and Exit', then reboot the target to activate the new settings. The 'CAN Bit Timing Paramters' shown in the screenshot above always shows the currently used settings (read back from the CAN controller's register, not the configuration data).

"tq" means "time quantum". Each CAN 'data bit' is divided into as many time quanta as possible (as far as the CAN controller permits). The duration of a single time quantum is displayed in ns or us (nanoseconds or microseconds, depending on the CAN speed).

In most cases, you don't need to care about the CAN bit timing parameters because the default values (SJW=1 time quantum, Sample point near 83 %) should be ok. For details on CAN bus timing parameters, consult the web or your company's CAN bus specification - especially for road vehicles.

6.2 CAN FD Configuration (for devices supporting ‚Flexible Datarates')

In devices supporting CAN FC („CAN with flexible data rate"), the FD-specific parameters can be configured under

Main System Menu … User Settings … more CAN settings ... CAN1 FlexDataRate

This includes the ‚flexible' bitrate (in kbit / second) for the **data field** („data phase").
In this case, zero means „no bitrate switch allowed", i.e. use the same bitrate for the data- as for the arbitration phase. For details about the complex subject ‚CAN FD', study textbooks or online literature. The MKT-View programming tool's online help system only contains a few hints, because at the time of this writing (2020-01), a CAN FD capable ‚MKT-View' only existed as a prototype.

Usual ‚data' bitrates for CAN FD are 2000, 4000, and 5000 kBit/second; in rare cases (with suitable controllers and transceivers) also 8000 kBit/sec. Please consult your equipment's hardware manual for the *maximum* bitrate (e.g. „MKT-View V") .

7 System Setup

The **System Setup** in the terminal's system menu contains most hardware-related settings. Due to the large hardware variety, some of the menu items listed below may be missing in a particular device.

Exit

Leave this menu without saving modified parameters permanently in EEPROM .

Save & Exit

Save changes (if any) in EEPROM, and leave this menu.

Display setup ... ►

Used in devices with a very 'rich' setup to open a submenu with the following, display-related items.

LCD dimming ... ►

Used in devices with a very 'rich' setup to open a submenu with backlight-related items.

LCD Brightness / CFL Brightness

LCD backlight intensity. This was once a CFL (cold flourescence tube), but meanwhile most displays use white LEDs for the same purpose.
In some devices with automatic adjustment of the display backlight (e.g., MKT-View II) you will find an item labelled "LCD Dimming...." instead. It contains different settings for day and night operation, and for the "low-power" state which may be entered after a configurable time of no activity (LCD Off-Time). For details see chapter 11.6.

Low-Power Brightness

This display brightness is used after the configurable time of 'non-activity' ("LCD Off-Time") has expired, without keystrokes, touchscreen events, or rotary encoder knob events.

LCD Off-Time

Number of seconds until the display backlight is automatically turned off (or, for TFT displays, reduced to the "low power" brightness value). A value of "0" means the LCD backlight will *not be turned off automatically*. As long as the operator "does something" (operates the keyboard, rotary knob, touchscreen,..), the backlight will not turn off. Only if 'nothing happens' for more than N seconds, the backlight *may* turn off automatically.
Additionally, the *display application* itself may contain a command to turn the backlight on/off, but that's beyond the scope of this document.

LCD Contrast Table (only required for devices with monochrome 320*240 pixel display)

Used to fine-tune the LCD contrast, depending on the temperature. This function shows a graph ("curve") of the temperature (measured with an internal sensor) and a pulse width modulator value, which finally generates a temperature-dependant LCD backplate voltage.

The F1+F2 keys are used to select one out of 20 sampling points (marked as vertical line in the diagram). The rotary knob (or cursor up/down) can be used to find the optimum backplate voltage for that point. F3 opens an extra submenu in which the "factory default" curve can be recalled (for example, after replacement of the LCD, or if someone trashed the settings). In the same submenu there is another "Save & Exit" function which saves the modified contrast curve permanently in EEPROM, and leaves that menu.
Devices with TFT display display don't need a contrast / temperature compensation table.

Bootup Display

TRUE = Show a start display, and certain popup messages at system boot.
For devices with TCP/IP and DHCP this includes a popup with the device's IP address and host name, which pops up for a short time after an external DHCP server has leased a dynamic IP address to the display.
FALSE = do not show such messages / infos at system start.
The first info, visible for the operator after turning on, is the first user-defined display page ("Page Zero", which typically contains a company logo or similar).

Bootup Debug = 0 / 1

0 = don't send any bootup-info-messages via the serial port
1 = send messages during system bootup over the serial port, with 115.2 kBit/second.
In contrast to the option 'Bootup Display', messages can be sent very early (during the bootup process) through the serial interface, including (for example) the result of the memory test.
If the serial port (RS-232) is used by the application, this option should be turned off (which is the default when devices are shipped), i.e. Bootup Debug = 0 .

Visible Gestures

TRUE = show touchscreen-gestures while they are 'painted' on the screen, as a dotted line.
FALSE = do not show touchscreen gestures.

Portrait Rotation (short: „Portr.Rotate“)

CLOCKWISE : Das Gerät wurde -gegenüber der normalen Einbaulage- um 90° im Uhrzeigersinn montiert. Beim MKT-View befindet sich die Lemo-Buchse dann *unten*.
COUNTERCLOCKWISE : Das Gerät wurde -gegenüber der normalen Einbaulage- um 90° gegen denUhrzeigersinn montiert. Beim MKT-View befindet sich die Lemo-Buchse dann *oben*.
Diese Einstellung hat nur dann einen Effekt auf die Anzeige, wenn eine für den „Portrait“-Modus, d.h. Hochformat erstellte Applikation in das Gerät geladen wird.

Touchscreen Calibration

Special function to calibrate the touchscreen (align it with the pixels on the display).
Four reference points are shown on the screen, which must be touched precisely in the center (use a stylus or pen for this). Hold the pen down until the reference marker turns green.

Other items in the setup menu; in the MKT-View II + III the audio settings were moved into 'Setup Menu' / 'Audio Settings' because the main setup menu contained too many items:

Audio Setup / Click Volume (only for devices with touchscreen and internal "buzzer" / "beeper")

Sets the volume of certain acoustic signals ("Clicks") when the user operates the touchscreen.
0 = off ... 100 = max. possible volume. Default: 20

Audio Setup / Microphone Volume (only for devices with integrated audio recorder)

Sets the microphone gain. The unit and scaling depends on the hardware. For example, in the MKT-View II, each step means an additional gain of 6 dB; 0=minimum gain (almost inaudible), 7=maximum gain.

Audio Setup / Speaker Volume (only for devices with analog audio/voice output)

This parameter controls an electronic potentiometer between D/A converter and the audio output amplifier.
Value range : 0 = off ... 100 = maximum volume.

Audio Setup / Beeper to Speaker (only for a few devices with analog audio/voice output)

If this parameter is nonzero, pulse-width modulated 'beep tones' which are normally only routed to a small transducer ('piezo' or similar) will also be fed to the D/A converter for the 'analog' audio output. Thus, with 'Beeper to Speaker' set above zero, those beeps are amplified by the 'audio power amplifier', and emitted through the loudspeaker on the front panel (instead of just the small internal 'piezo').
Value range : 0 = off ... 100 = maximum volume.
Signal tones generated this way are only audible (through the speaker) if the parameter 'Speaker Volume' is also nonzero.

Date & Time

Sets date and time, if the device has a battery-buffered real time clock. Details in chapter 9.

CAN Baudrate

Baudrate for first CAN-port, if no valid application is loaded (or if loading of the application is skipped, as explained earlier in this document). In those cases, this baudrate is used for the communication between terminal and programming tool. Ideally -and to avoid confusion- you should use the same setting here (in the System Setup) and in the User Settings (for CAN1). Otherwise you may end up with CAN bus problems. But in some cases, for example when a "very low" baudrate is used during normal operation, and a higher baudrate for programming, two different configurable baudrates can help.
Please note that by entering a "low baudrate" (like 83.33 kBit/sec) you cannot turn a HIGH-SPEED-TRANSCEIVER automatically into a LOW-SPEED-TRANSCEIVER, and vice versa ! The firmware doesn't know which kind of CAN bus transceiver is installed on the PCB, and for this reason the system menu will happily show all possible CAN baudrates supported by the *CAN-Controller* (not only those supported by the *CAN-Transceiver* !)

Note: There is a second place in the setup menu where the CAN baudrate (and other CAN bus parameters) can be set: It's under USER SETTINGS. The USER SETTINGS are only active when a valid configuration has been loaded. The CAN-baudrate in the SYSTEM SETTINGS will be used when the USER SETTINGS are invalid, or not loaded. For details, see User Settings .

Module/Node-ID

Node-ID ("number") of the CANopen slave node implemented in the terminal (if CANopen is supported at all). Valid range for CANopen is 1..127 . The same node ID must be entered in the programming tool under "Terminal Node-ID", so a connection can be established via CANopen / SDO . The default value is usually "1", but some customers use another node ID for the terminal (because node IDs 1...4 were usually occupied by other devices like I/O modules, and a node ID must be unique throughout the network).

CAN-TxEnable=x (only for certain devices, which can disable the CAN-transmitter via software)

In older devices, this parameter was a bit-combination to activate or deactivate the CAN-TX-output in the CAN-transceiver. Bit 0 controlled the first CAN interface, bit 1 the 2nd interface, and so on. In newer devices (with sufficient code memory), there is an extra menu item for each CAN bus so you don't have to hurt your brain with binary / decimal conversions :-) . More details about disabling the CAN-transmit-function is in the online help system of the programming tool, file "canrx_01.htm" .
With a passivated CAN-transmit output, the CAN controller can neither actively transmit CAN messages to the bus, nor can it send an acknowledge-bit for successfully received CAN messages (thus 'transmit enable' also affects reception).
Note: This option doesn't have anything to do with the unlockable feature 'CAN-transmit' in certain devices like MKT-View !
Alternatively, the 'CAN-Tx-Enable'-flag can be modified via the programming tool as part of the display configuration, as explained in the manual of the programming tool in the chapter CAN bus timing parameters as part of the 'Terminal Settings' / 'User Settings' . The same applies to the CAN bus termination resistor in the next paragraph.

CANx-120_Ohm=x (only for certain devices, where the CAN terminator resistors can be switched via software)

This function activates or deactivates the internal 120-Ohm CAN bus termination resistor. x=0 means "no terminator", x=1 means "terminator enabled". Caution: If the power supply is off, the termination resistors are off, too (if they are selectable via software). In contrast to that, the supply voltage did not affect the CAN terminators in (older) devices in which those terminators had to be "plugged" (or "jumpered") physically.

WakeupSource=xxx (only for certain devices, e.g. MKT-View II)

Defines how a device is turned on, or "woken up". Possible options:
Key : Turn on only through the power key (F1 in most cases)
Vin : Automatically turn on when the input voltage threshold is reached.

Note:

The hardware-specific threshold voltage for turning-on via supply voltage is specified in the device's *hardware manual*, for example Document Nr. 85512 for the MKT-View II. Below that threshold (8.5 Volt for the MKT-View II) the device can only be turned on via key at your own risk ! The device may turn itself off immediately if the voltage drop accross the cable, is too large. The 8.5 V "turn-on" threshold leaves enough headroom for long cables and weak batteries, because the device operates safely at 7 Volts or even less if the internal UPS capacitors are charged. When turning on the MKT-View II at low supply voltages, the input current may exceed 0.5 A which causes a significant drop across the supply cables / filter chokes, etc.

CAN : Automagically turn on when there is "activity" on the CAN bus
Key, Vin : Turn on via key or positive edge of the supply voltage
Key, DigIn : Turn on via key or positive edge of the first *digital input*
Key, Vin, DIN, CAN : Turn on *on any* of the above sources ("or"-combined)

Note: As a matter of course, the device won't ***turn itself off automatically***, just because any of the *wakeup*-sources is currently 'passive' ! To shutdown the device, use the command **sys.poff** in your display application (because only *you* will know when it's time to shut down), or let the operator shut down the device through the shutdown function (key).

RTC Chip Type (for devices with ***different possible RTC-chips***)

Must be set to the RTC chip equipped on the printed circuit board.
A value of '0' (zero) means: "The real time clock is not installed". In that case, the firmware (precisely: the BIOS) will not waste time trying to talk to the RTC chip. To set the clock, see chapter 9.

Display Type = XXX (type of the connected LC display)

Only used when *different, incompatible* LCDs can be connected, without a different firmware for each display. For example, in a TFT display the internal controller was replaced by the display manufacturer (from HX8312-A to HX8346-A), rendering "old" and "new" display incompatible (a nice suprise, you can imagine..). To avoid having to create and maintain a new firmware article number, the firmware now supports both "old" and "new" TFT. In the menu item "Display Type" you can see which of the displays is actually used in the terminal. Do not select this menu item with the ENTER key if you don't need to, because it starts the "semi-automatic display type detection", explained in another chapter.

GPS Rcv Type = XXX (type of the connected GPS-receiver, if any)

Only used if *different* (read: incompatible) GPS receivers can be connected to the device. When set to "NONE" in this menu, the type of the GPS receiver can still be defined in the display application, using the commands gps.prot, gps.cfg, gps.baud; details about that in the online help system of the programming tool, file gpsdec_01.htm .

Joystick Calibration

Only for devices with an integrated analogue joystick (as replacement for cursor keys). Because these devices are already calibrated when delivered to the customer, you (dear customer) will hardly ever need to invoke this function. Details may be found in the hardware-specific device manual, or may be requested from the software developer at MKT Systemtechnik.

Enter UNLOCK Code

By entering an unlock code, certain special functions (e.g. CAN logger, CAN snooper, GPS/NMEA decoder, etc) may be permanently unlocked in order to use them. Details in chapter 8.1.

Enter Password

Only used during the manufacturing process, or repairs, but not for the end user. A certain password must be entered in this field before certain other data (for example the serial number) can be modified in the setup menu. Please don't tamper with this function ;-)

Serial Number (Serial number of the device, 5 digits)

Only used during the manufacturing process, or repairs, but not for the end user. Each device receives an individual serial number during production. The end user can READ (but not modify) this number, to request an unlock code for a particular device from the manufacturer.
The serial number is also displayed while holding the system key combination.

8 Special functions

8.1 Unlock Codes for special functions / options

By entering an numeric code ("key") in the terminal's system menu, the user can unlock certain special functions and hardware featured (e.g. CAN-Logger, CAN-Snooper, etc).
Usually, such features are already known when a device is ordered at MKT Systemtechnik; in that case the unlock code has already been entered and you (dear customer) don't need to enter it yourself.
A certain *unlock code* is only valid for a certain device (because it's tied to the serial number), and usually unlocks one (MKT-View I) or more (MKT-View II) features. In the MKT-View I, two different unlock codes were required to unlock two different functions. In the MKT-View II, one unlock code can unlock one, two, or even more different features at the same time (but it doesn't have to).

Up to date, the following *unlockable features* were implemented:

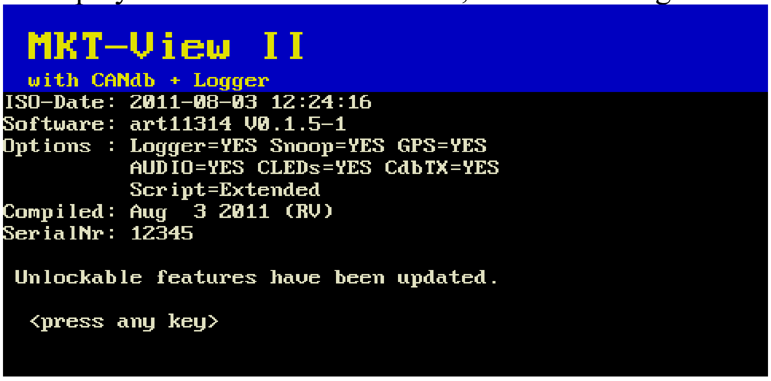
"MKT-View +" :

- Feature #1 = Activation of the CAN-LOGGER
- Feature #2 = Activation of the CAN-SNOOPER

MKT-View II / III / IV (with firmware date 2015-05-12 or later) :

- CAN-Logger (Code: "Logger"
- CAN-Snooper (Code: "Snoop")
- NMEA decoder for an external GPS receiver (Code: "GPS")
- Audio record and replay (Code: "Audio")
- Transmission of CAN signals defined in a 'CANdb' file (Code: "CdbTX")
- *Extended* functions in the script language (Code: "Script")

Here, for example, the display of an older MKT-View II, after unlocking *all* features:



```
  MKT-View II
  with CANdb + Logger
ISO-Date: 2011-08-03 12:24:16
Software: art11314 V0.1.5-1
Options : Logger=YES Snoop=YES GPS=YES
          AUDIO=YES CLEDs=YES CdbTX=YES
          Script=Extended
Compiled: Aug  3 2011 (RV)
SerialNr: 12345

 Unlockable features have been updated.

  <press any key>
```

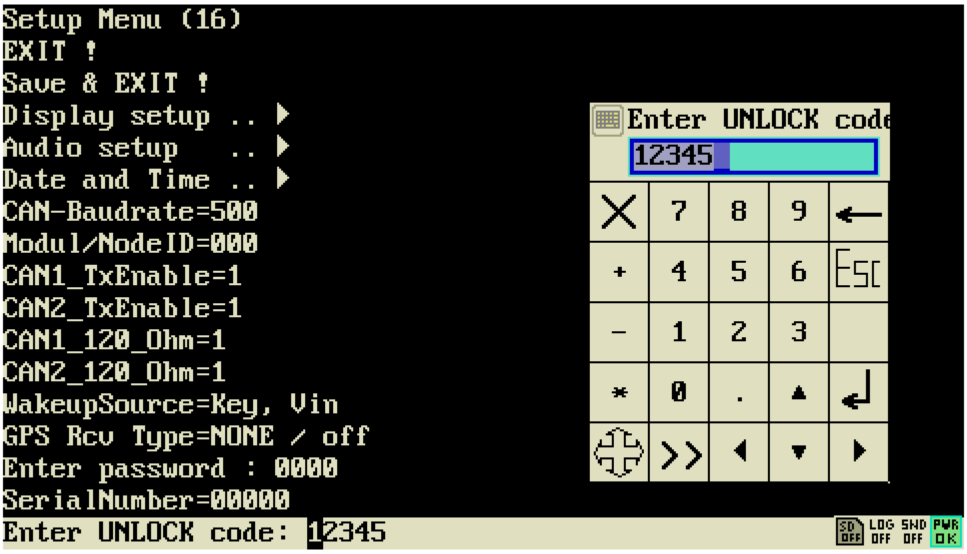
Principle ("procedure") to unlock special features (here for the older MKT-View +):

1. Invoke the system menu by pressing F2+F3 simultaneously
2. Select **System Setup / UNLOCK** via Cursor / ENTER or rotary encoder / knob
   (if you only see **System Setup** but not **System Setup / UNLOCK**, the firmware doesn't support unlockable features, or there is no need to unlock anything)
3. Write down the terminal's serial number, call MKT Systemtechnik, tell us the serial number, wait for confirmation that you really paid for the feature, and ask for the unlock code.
4. MKT: Check if the customer is authorized to use ("has paid for") the feature; if yes, retrieve the unlock code from an internal list, and send the code via fone, fax, or mail.
5. User: Select "Enter UNLOCK Code" in the system menu
   (press ENTER once, so the cursor marks a single digit, not the entire menu item)

6. Enter the unlock code in the terminal
   (MKT-View 1: Unfortunately there's only the rotary knob, use F1 / F2 as a replacement
   for cursor left/right to select the next digit .
   For devices, use the virtual keyboard as shown in the screenshot below.)

7. Finish the entry by pressing ENTER, and wait.
   The firmware now checks the code, and finally tells you which functions have been unlocked, e.g..
   CAN-Logger = 1 (or, depending on the display, CAN-Logger = YES or NO)
   CAN-Snooper = 1 (if both functions were unlocked)

8. If necessary, repeat some of these steps for the second unlock code..



```
Setup Menu (16)
EXIT !
Save & EXIT !
Display setup .. ▶
Audio setup   .. ▶
Date and Time .. ▶
CAN-Baudrate=500
Modul/NodeID=000
CAN1_TxEnable=1
CAN2_TxEnable=1
CAN1_120_Ohm=1
CAN2_120_Ohm=1
WakeupSource=Key, Vin
GPS Rcv Type=NONE / off
Enter password : 0000
SerialNumber=00000
Enter UNLOCK code: 12345
```

Enter UNLOCK code
12345

(screenshot 'Enter Unlock Code' in the MKT-View II)

Notes on unlocking :

- The information about which functions have been unlocked is stored in nonvolatile EEPROM. If that EEPROM is replaced (e.g. during repair), the information is lost (like in a brand-new device, where none of these functions is unlocked).
- When entering a wrong unlock code (by accident), none of the already unlocked features is disabled. But don't try to tramper with this function... the terminal willl notice it, and after trying out too many invalid unlock codes, the device may refuse to work !
- If a certain device has been loaded **once** with a firmware containing unlockable functions (for example art11089.hex for MKT-View I), it may never be loaded with a different firmware acticle number (for example art11087.hex) because the EEPROM region holding the unlock codes may be used for "something different" in a different firmware.
  In short form: **Once** art11089.hex was loaded -> **never** load art11087.hex into that device, because you may lose the unlocked features ! You may update art11089.hex with a newer version of that file, but not vice versa.
- Certain devices (for example MKT-View II) were equipped with a firmware without the need to unlock anything *during the product launch phase*. So all firmware functions could be used freely. By the time of this writing, it wasn't decided yet *which* functions will need to be unlocked later, and *when*. So be prepared that one day, after a firmware update, you can only use the functions which have been ordered (for example, in one of the """bundles""" by MKT).

8.2 Semi-automatic detection of the display type

In certain devices (just *a few* at the moment) different LC displays may be used, because certain displays are discontinued a few months after they appeared. For example, displays with onboard controller type HX8312-A have been replaced with incompatible displays, controller type HX8346-A.
Unfortunately, the microcontroller often cannot detect the type of the installed display. In such cases, a certain location within the configuration EEPROM contains the display type (if *one* firmware supports different, *incompatible*, LCDs). During the initial power-up (at the manufacturer), or after a repair, or after an accidental loss of EEPROM data, the CPU doesn't know the display type. To allow the microcontroller to initialize the "unknown" display, the following program sequence has been implemented in all terminals affected by "discontinued" LCDs:

1. If it doesn't find a valid display type signature in EEPROM during power-up, the microcontroller initializes all "possible" displays, one after the other, and (tries to) show a visible message on the display for a few seconds.
2. As soon as the operator can see a "readable" image on the LCD, he/she presses a key on the terminal
3. The microcontroller now exits from the "test loop", and continues normal program execution
4. The operator then enters the terminal's system menu, and selects "Save and Exit".
   The microcontroller now saves the display type signature permanently in the data EEPROM .
5. On the next power-up sequence, the microcontroller knows which display type is in use, and the above steps are skipped. The application starts immediately.

8.3 System Test and diagnostic functions

Only for software developers !

The display's system menu contains a few entries (under 'System Test' and 'Diagnostics') which may help you during software development. The following functions may, or may not, be implemented in a particular device, due to memory restrictions.

System Test - CAN-Bus-Test :

Shows the current status of the CAN-bus, and the last few CAN messages which were received by the device (only those CAN-messages which have been 'registered' for reception by the application, i.e. only message which pass through the CAN-controller's ID filter).

Diagnostics - Error History

Shows the most recent errors and warnings which have been detected by the system.

Diagnostics - TRACE History

Shows errors, warnings, and (this is most important) a history of CAN messages which have been received and sent by the application. The format is similar to Vector's 'ASCII' files for CAN logfiles. For certain devices, the contents of the TRACE memory can be read through a diagnostic interface (RS-232) or via web browser (LAN). More details are explained unter 'Trace History' in the online help files, or can be requested from the software developer (ask for Mr. Büscher at MKT Systemtechnik).

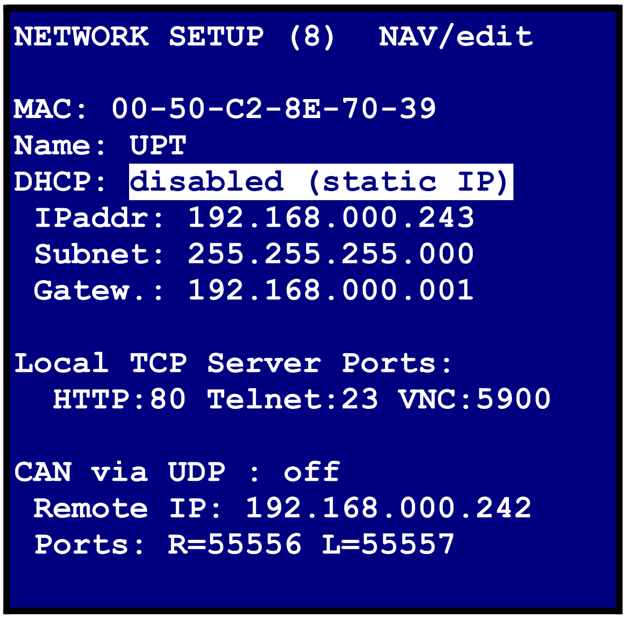
Diagnostics - Browse Display Pages

This function is only available in devices with a sufficiently large program memory.
It allows you to 'browse' all display pages in the currently loaded application - even for pages that (during normal operation) are only invoked via script or interpreter command, for example after the detection of an error, illegal value received from CAN, bus-off, etc.
On devices with a rotary button (e.g. MKT-View III / IV) rotate the button to switch pages.
On devices with cursor keys, use Cursor left / right to switch to the previous / next programmed display page.
To switch back to the 'Diagnostics' menu, press ENTER, ESCAPE, F3, or push the rotary button.

8.4 Network Setup

Only for devices with Ethernet port (hardware) and a TCP/IP protocol stack (firmware) !
This menu can be used to configure the following parameters (and possibly more) :

- the MAC address of the Ethernet port,
- the IP-adress, which can be used to access this device in the local network, for example 192.168.0.100 (must be assigned manually, if DHCP isn't available !)
- the subnet mask (255.255.255.0 in most cases)
- the IP address of the standard gateway (for example 192.168.0.1; ask your system admin)
- the port number of the embedded HTTP server
- the port number of the embedded Telnet server
- the (UDP-) port number for CAN_via_UDP, also used for a *simple* file transfer option
- the port number of the embedded FTP servers (FUTURE PLAN !)
- the port number for the optional VNC server (especially for devices with remote display)



```
NETWORK SETUP (8)  NAV/edit

MAC: 00-50-C2-8E-70-39
Name: UPT
DHCP: disabled (static IP)
 IPaddr: 192.168.000.243
 Subnet: 255.255.255.000
 Gatew.: 192.168.000.001

Local TCP Server Ports:
  HTTP:80 Telnet:23 VNC:5900

CAN via UDP : off
 Remote IP: 192.168.000.242
 Ports: R=55556 L=55557
```

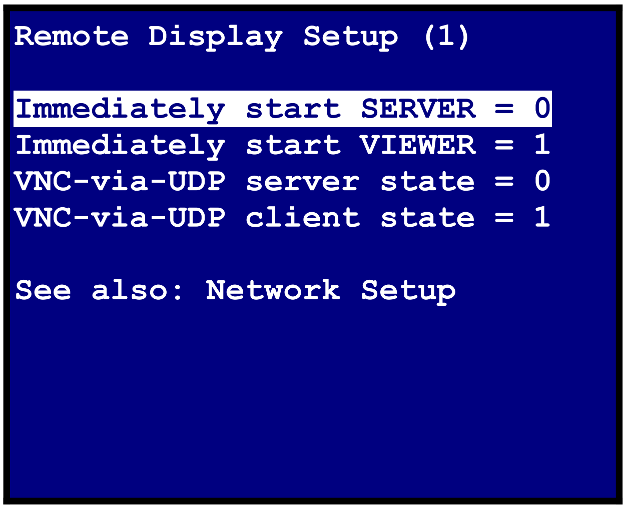
Concerning the configuration of a TCP/IP network, we highly recommend to consult a good textbook. A wrong network configuration -for example duplicate MAC addresses, or duplicate IP addresses- can cause severe network problems, up to a collapsing network !
More information about the embedded web server (in *some* of MKT's programmable displays) can be found here (online, link to the MKT website).

8.4.1 Remote Display Setup / 'VNC-via-UDP'

For display-less devices, a simplified 'Remote Framebuffer Protocol' was implemented. It doesn't use TCP/IP but UDP, and can be configured in under 'Remote Display Setup'.

Some devices, for example the MKT-View IV, have both a VNC-via-UDP server and client (aka 'Viewer' integrated in the device firmware. Thus they can either remotely control another device, or be remotely controlled by another device.

Depending on the VNC-via-UDP capabilities (server only, client only, both client and server), the Remote Display Setup page in the system menu may be slightly different than shown below.



```
Remote Display Setup (1)

Immediately start SERVER = 0
Immediately start VIEWER = 1
VNC-via-UDP server state = 0
VNC-via-UDP client state = 1

See also: Network Setup
```

The server and/or client states displayed in the setup are only for testing or debugging. Possible values are:

0 = passive (not enabled),
1 = sending announcement (server side) or waiting for announcement (client side)
2 = waiting for a client (server side) or trying to connect server (client side)
3 = sending framebuffer (server side) or connected to server (client side)

If the VNC client (aka „Viewer“) cannot connect to the server (aka „remotely controlled device“), make sure they are enabled (either by 'starting immediately' or controlled by the application/script), and check the network setup : Both devices must have a valid Ethernet MAC address, suitable IP addresses (use fixed IP addresses if no DHCP server is available), etc.

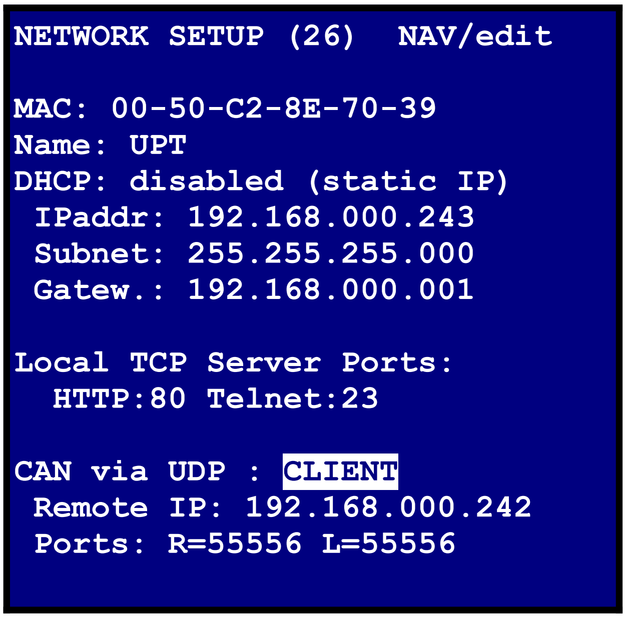
8.4.2 CAN-via-UDP

CAN-via-UDP is a simple UDP-based protocol, which was introduced to increase the number of CAN buses available for a certain device. For example, the MKT-View II has two built-in CAN bus interfaces (called "CAN1" and "CAN2"), but it can use two additional, external CAN interfaces ("CAN3" and "CAN4") connected through CAN-via-UDP.

Furthermore, the CAN-via-UDP protocol contains a very simple (compared to FTP) file transfer option, which can be used to upload display applications into the programmable device via Ethernet (LAN).

Details about CAN-via-UDP can be found in document #85140, "CAN via UDP - Configuration, Usage, and Protocol Specification". That file is located in the 'Doku' folder of the programming tool (after installation).

The device which shall use additional CAN ports (provided by an external CAN-via-UDP server) must be configured as CAN-via-UDP **Client**. In almost all cases, the default UDP port number (55556 for the remote port, and 55556 for the local port) remains unchanged (the only scenario in which the port numbers must be modified is explained in document #85140). Example:



```
NETWORK SETUP (26)  NAV/edit

MAC: 00-50-C2-8E-70-39
Name: UPT
DHCP: disabled (static IP)
 IPaddr: 192.168.000.243
 Subnet: 255.255.255.000
 Gatew.: 192.168.000.001

Local TCP Server Ports:
  HTTP:80 Telnet:23

CAN via UDP : CLIENT
 Remote IP: 192.168.000.242
 Ports: R=55556 L=55556
```

In the network setup, enter the IP address of the remote CAN-via-UDP server in the field labelled "Remote IP'. This is the IP address of a device which provides its own CAN ports for other devices. The server may be a second MKT-View II, a PC, or a special (yet to be designed) CAN/Ethernet gateway.

8.5 Firmware Update

Usually, the firmware-bootloader is activated by holding the first and third function ke (F1 + F3) pressed during power-up.

If this is impossible for some reason (for example, you cannot turn the power off, or the device doesn't have a "real" keyboard), you can still invoke the firmware-bootloader from the system menu.

To do this, invoke the system menu as described earlier, and select the submenu "Diagnostics". There, select the item "Call Bootloader" (and press return, etc).

If that is ***also*** impossible (for example, because your display "terminal" has neither a keyboard, nor a rotary encoder, nor a touchscreen), you can still activate the firmware bootloader through the *high programming voltage mode*. In that mode, the keyboard can be replaced with a remote control function via CAN-bus. Please contact the manufacturer if you really need this feature, and consult the device's data sheet for the maximum allowed supply voltage.

The terminal will then switch from normal operation (i.e. the "normal" firmware) into bootloader mode.
Details about the bootloader, and how to update the firmware via CAN-bus or memory card, can be found in the file

fwupdate.txt

in the "firmware" folder of the terminal programming tool.

An overview of all available firmware variants and -revisions can be found in the same directory, in the file

readme.txt

9 Setting date and time

If the device is "brand new" (and the RTC has not been configured yet), or after replacement of the configuration EEPROM, or after replacement of the clock/memory backup battery :

- Select the RTC chip type in the terminal's system menu (e.g. "RTC-ChipType=DS1307" for the "Dallas-Chip" which was widely used some years ago).
- Select the function "Date & Time" in the system menu / system setup. The clock doesn't "tick" yet (some chips begin to run only when "kicked alive", which they only do when the "clock-run" register is written).

To set date and time in the system menu (see alternatives further below)...

1. In the System Setup menu, select "Date & Time" (ENTER).
2. Press F1 to enter clock-set mode.
3. Set the year number (first field) via cursor keys (up,down,left,right, or rotary encoder).
4. Press ENTER (or the encoder knob) to switch to the next fields (month, day, hour, minute, second) to set those values.
5. Wenn finished, press ENTER again. The edit marker should disappear, and the second counter should be ticking. Otherwise, the chip may be defective, the wrong RTC chip selected, or the battery discharged.
6. Turn the power off, wait a few seconds, and turn on again. If the time isn't correct now, the battery is discharged and must be replaced..

ALTERNATIVE 1 :

Copy date and time from a GPS receiver..
(provided that a GPS receiver is connected, correctly initialized, and satellites in range):

7. In the System Setup menu, select "Date & Time" (ENTER).
1. Wait until a GPS-time and -date is displayed in the lower part of the screen
2. If no valid GPS-time is displayed after some minutes:
   Press F2 . The firmware will try to re-initialize the GPS receiver, possibly using a different protocol and a different serial baudrate.
3. If GPS date and -time are displayed, and look plausible:
   Press F2 . Date and time will now be copied from the GPS receiver into the registers of the real time clock (chip), of course using UTC = Universal Time Coordinated, which is almost the same as "Greenwich Mean Time", because GPS doesn't care for time zones.
   Note: The terminal neither knows in which time zone you are, nor does it know anything about daylight saving time in your country. The terminal is a "deeply embedded system", not a windoze PC ;-) !
   After copying the time from the GPS receiver, the terminal's internal clock (RTC) will continue to run in UTC. If that doesn't suit your needs, adjust the hour field.

ALTERNATIVE 2 : (for devices with firmware compiled 2008-09 or later)

When loading a terminal application ("UPT program") from the programming tool into the terminal via RS-232 or CAN bus, the terminal's clock will automatically synchronized with the PC's clock. A manual clock adjustment is not necessary in these two cases.
Of course, this doesn't work if the UPT program is copied from the PC to a memory card, and from the memory card to the programmable terminal.

10 Remote Control

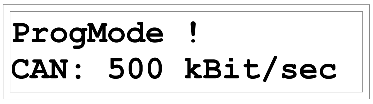
Since 2009, certain display terminals made by MKT Systemtechnik were neither equipped with a keyboard, nor with a rotary encoder button, nor with a touchscreen. To configure such 'keyboard-less' devices, a crude remote control function was added in the firmware for such devices. It allows at least to enter and operate the terminal's *system menu*, where the system can be set up.

This chapter describes how to switch the device into *remote control* mode, and how to communicate with it in that mode.

10.1 How to switch the terminal into *remote control* mode

Because there is no extra communication port dedicated to remote control, the terminal must be switched into remote control before using it. For example, only in remote control mode, certain CAN identifiers may be used for communication between the device and a remote terminal. During normal operation, there must not be any "reserved CAN identifiers" to avoid conflicts with the (unknown!) existing network.

Most of the remotely controllable terminals use a *high voltage programming mode* to enter a passive remote control state. For example, a certain terminal rated for 24 Volts continuous supply voltage may tolerate temporary operation at 31 Volts (please consult the datasheet or hardware manual of your terminal for details).
If the terminal detects the *high voltage programming voltage* for a sufficiently long time (usually a few seconds; hardware dependent), it opens a popup window showing something like



```
ProgMode !
CAN: 500 kBit/sec
```

which means "Programming Mode has been activated, and the terminal is listening on the CAN-bus, using a bitrate of 500 kBit/second at the moment" .

This message will be hardware specific; in future versions there may be other communication channels, too.
It will disappear after a few seconds. But the terminal continues to listen on the remote control interface (which may be CAN or something else).

For example, if the remote control interface is a CAN bus port, you may now connect WB's *CAN Tester for Windows* to establish the remote control connection.

Alternatively, you can remotely control the terminal (at least for the system menu) through a telnet connection. For example, if the terminal's IP address is 192.168.0.100 (which is, or at least was, the default), enter telnet 192.168.0.100 as an OS command on your system (under windows: "Start".."Run".. and type the command. German windows PC: "Start".."Ausführen"..).

10.2 How to operate the terminal's system menu via *remote control*

After preparing the terminal for remote control operation (as explained in the previous chapter), press F12 in the remote terminal program to enter the display's system menu (which replaces pressing F2+F3 simultaneously on a device with a *real* keyboard). If F12 doesn't work (HyperTerminal / VT52 via RS232), try CTRL-S (Control-S, hex code 0x13). This should also work with ASCII terminals which don't support more than four function keys.

If correctly configured (see next chapter), the terminal program or telnet client (running on a PC) will show a similar display as on the device's LCD, as far as the crude VT52 / VT100 terminal emulation permits:

```
Main system menu (1)
EXIT !
Load program from FILE
Transfer via CAN = ON
Transfer via RS232 (SLAVE)
Send prg to RS232 (MASTER)
Audio Recorder
CAN snooper mode
CAN logger config
User Settings
System Setup / UNLOCK
System Test
Network Setup
Diagnostics
General Settings
```

```
Setup Menu (1)
EXIT !
Save & EXIT !
Display setup .. >
Audio setup    .. >
Date and Time .. >
CAN-Baudrate=500
Modul/NodeID=001
CAN1_TxEnable=1
CAN2_TxEnable=1
GPS Rcv Type=NONE / off
Enter password : 0000
SerialNumber=00815
Enter UNLOCK code: 00000
```

To navigate through the system menus, use the cursor keys on the PC keyboard. To select an item, press ENTER. To cancel something, press ESCAPE.
To send function keys F1 or F2 to the terminal, use SHIFT-F1 and SHIFT-F2, because for some reason, windows (or Borland's VCL) didn't pass F1 and F2 to the terminal program (at least the one implemented in WB's CAN-Tester for Windows).

10.3 Configuration of the terminal program

The following configuration applies to the terminal program integrated in WB's CAN-Tester for Windows. For other programs (if any exist..), similar settings are required.

- Send characters immeditately (the terminal program shall send characters while you type them, not wait for the ENTER key)
- Send Acknowledge : off
- Max. Number of Text Lines : 50 or 100 (the system menus will never require more than 50 lines of text)
- Terminal Emulation: "MKT Keyboard" or "VT52" (not VT100 , not "ANSI" !)

The character stream sent from the remotely controlled device to the terminal program (on the PC) contain a few VT-52 compatible escape sequences. As of 2009-11-10, these were:

- ESCAPE H to move the cursor to the 'home' position
- ESCAPE J to clear the screen (beginning at the cursor position, for what it's worth)
- ESCAPE Y <32+row> <32+column> to set the cursor position
  (where 'row' and 'colum' start counting at zero; 0,0 = 'home')
- ?

Characters sent from the terminal program (like the CAN-tester) to the remotely controlled device use, to keep things simple, the keyboard codes used by MKT's own keyboard driver. A list of keyboard codes is in the programming tool's help system, see interpreter function "kc".

For RS-232 (serial port) or Telnet, normal 'printable' ASCII characters, or (for cursor- and function keys) VT-52 oder VT-100 are used.

11 Special features of the MKT-View II / III / IV

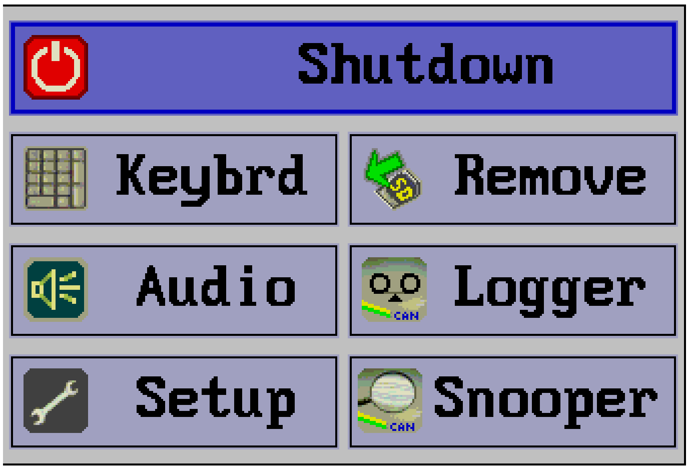
(also applies to similar devices like "UPT 320", "HBG 35", ...)

11.1 Power on / off

The MKT-View II may be turned on and off through the "power"-key (here: secondary function of the F1 key, labelled with a power key symbol for that reason). How to use:

- To turn on, press the power key shortly (about 0.5 seconds).
  In most cases, you don't need to turn on the terminal manually, because the default setting (in the terminal's Setup menu), the terminal will boot up as soon as a sufficient supply voltage is detected.

- To turn off, hold the power key (aka F1) pressed until this display appears:

  Alternative (for devices without any 'real' key): Use the touchscreen-gesture "U" .

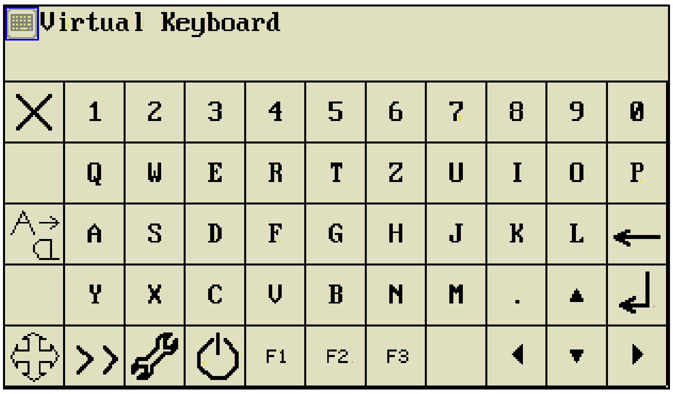
  Note: If this screen doesn't appear (for example because no suitable firmware was loaded), you can still enforce shutdown by holding the power key down for approximately 10 seconds. This works even if the main CPU has "crashed" - like a windows PC .

- In the window shown above, tap on the "Shutdown" panel (using the touchscreen), or set the focus to that field with the rotary knob, and press the knob (= ENTER).

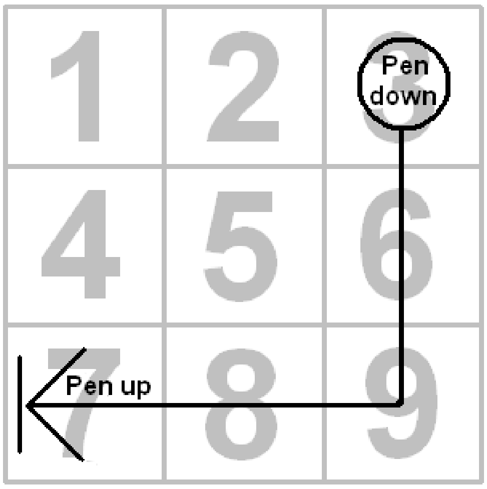
How to turn on the terminal (via key, supply voltage, CAN bus activity, etc) can be configured in the terminal's System / Setup menu. Select the "WakeupSource" function as described earlier in this document.
The "Shutdown"-window shown above also enables you to launch some special functions, which will be explained in the following chapters.

11.2 Virtual Keyboard

In the MKT-View II (and similar terminals with a touchpanel), a virtual keyboard can be used for numeric or alphanumeric input. Since there is no "real" keyboard available for this device, the virtual keyboard is the only way to enter certain characters... not as fast as a "real" keyboard, but it works sufficiently.



There are several ways to open the virtual keyboards. In some situations (for example in the system menu described earlier), it opens automatically when editing a numeric value. Otherwise, it can be opened with a (fast!) double-click of the rotary encoder button. If there is neither a (real) 'Enter' key, and no rotary encoder, use the touchscreen gesture "ENTER" to open the virtual keyboard :



„Enter“ gesture for opening the virtual keyboard

To close the virtual keyboard window (like some other windows provided by *the system*), tap on the 'close' field (large X) - see list of special fields further below.

To enter characters, tap on the required fields ("thick fingers" ? use a stylus..), or select a field with the rotary encoder, and press it to insert the marked character.

Note: In most cases, the characters are inserted in the standard keyboard buffer. So they have exactly the same effect as if a "real" keyboard was connected. For that reason, the user program (*.cvt or *.upt), but also the terminal's own system menu can use the virtual keyboard.

In certain conditions, the virtual keyboard may be connected to an edit field in the terminals sytem menu. In that case, there will be an extra edit field inside the virtual keyboard, and the title displays which field is currently being edited (instead of the title "Virtual Keyboard").

Usually, it's the operator's choice to open or close the virtual keyboard. In the user-programmable terminals (like UPT and MKT-View II), there are additional interpreter commands to show and hide the virtual keyboard, so you can use it in your programmed application.

Some fields ("keys") on the virtual keyboard have a special function:

✕ Close this window

✥ Move this window (slowly, while keeping the pen down on this field)

>> Switch to the next view (here: to the next keyboard layout)

A→a Toggle between upper and lower case letters (similar as Caps Lock)

🔧 Invoke the setup menu (system configuration), if allowed .
Not available in the first view (i.e. on the small numeric keyboard).

⏻ Turn off, or (depending on the hardware) invoke the "Shutdown"-selection window.
Not available on the first view.

← Backspace : move left, delete a character, scroll the rest of the text up

↵ Enter ("Return"): finish input, or (in rare cases) insert a new line

At the moment, there is one numeric page (with a smaller window), and two alphanumeric pages.

Details *for developers* are in the programming tool's online help system; see help/touchscreen_01.htm (english) or help/touchscreen_49.htm (german) .

11.3 "Safe" Removal of the Memory Card

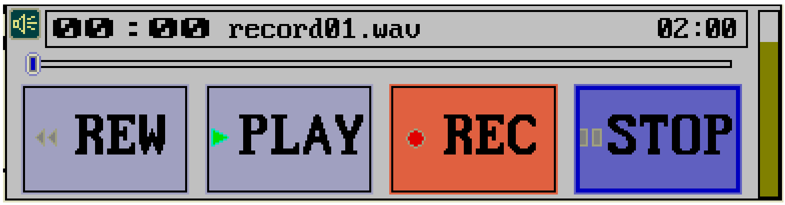
To remove the memory card savely, select "Remove" in the shutdown-window shown in a previous chapter. This causes the terminal to close all files on the memory card, update the directory and the file allocation table, and finally turn off the supply voltage for the card. After that, a small window pops open with the message "Remove !".

Only in this state the memory card may be removed !
Do never remove the memory card any other way, while the power is on !
(this applies to the programmable device as well as to the PC)

Users of a memory card reader under windows XP will be familiar with the little icon in the task bar (green arrow) to "safely remove hardware".
Ignoring to remove hardware "safely" ~~may~~ (Murphy's law: "will") damage the memory card, or cause loss of data.

11.4 Audio Recorder

The audio recorder is usually controlled by the user application (*.upt or *.cvt), but if necessary, a control panel for the audio recorder can be opened from the shutdown window shown above, or from the terminal's system menu. Depending on the LCD size, the audio recorder's control panel may look like this:



The graphic controls on that panel can be operated with the cursor keys / rotary button, or the touchscreen. They are:

- REW = Rewind to the start of the current audio file
- PLAY = Begin to play the audio file
- REC = Begin to record
- STOP = Stop recording or playing .

The coloured vertical bar at the right edge of the window is an audio level indicator.
During recording, the average level should be around 50 to 75 percent (in that case, the bar will be green to orange). The colours of the level indicator mean...

- Blue: Input volume too low, speech may be difficult to understand when playing. Get closer to the microphone, or (if supported), increase the microphone gain.
- Green: Mid range, ok .
- Yellow: Slight or sporadic overdrive, but no problem yet .
- Red : Input volume too high, overdriving the amplifier, speech may sound distorted .

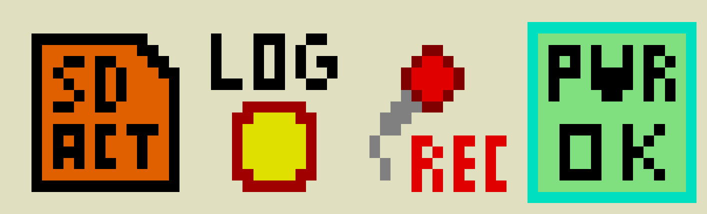
The audio recorder saves audio samples in uncompressed WAV-file format with 8 bits per sample, and 11025 samples per second.

This requires approximately 646 kByte per minute on the memory card !

During playback, recorded wave files with 8000 or 11025 samples/second are supported, but only with 8 bits per sample.

11.5 System Icons

The system icons (already known from older displays) have been extended for the MKT-View II. These icons can be shown/hidden through an interpreter command from the user application - details about that in the online help system of the programming tool, look for "Interpreter Commands"; sys.show_icons(). By default, these icons are displayed in the lower right corner of the screen but the *display application* may move them somewhere else.



The leftmost icon indicates the activity of the SD memory card:

SD OFF (gray) no memory card present, or its supply voltage turned off

SD ON (green) Active, but only files opened for ***reading***

SD ACT (orange) Active, and at least one file is opened for ***writing***

SD ERR (red) Error, card defective, not formatted, wrong file system, etc

Middle left icon: Activity of the CAN logger. Details in the manual of the programming tool !

LOG OFF CAN logger off (completely passive, not using any CPU resources)

LOG ○ CAN logger in state "Pre-Trigger" (waiting for a trigger event)

LOG ● CAN logger in state "triggered", recording to disk file

LOG ◎ CAN logger in state "Post-Trigger", still writing for a few seconds

Middle right icon: Activity of the audio recorder:

REC (mic symbol) Recording

🔈 (speaker symbol) Playing

SND OFF ("SND OFF") audio recorder passive

Rightmost icon: Power supply status (including the UPS for "safe shutdown"):

PWR OK Power "good" (input voltage ok; capacitors in the UPS are charged)

PWR bad Power "bad" (input voltage too low, or capacitors not charged yet)

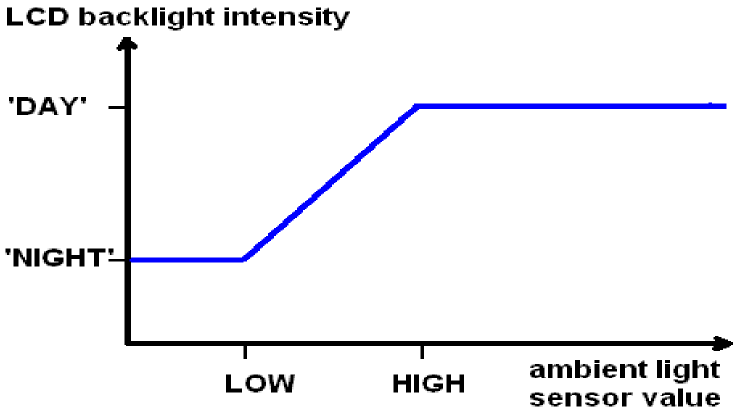
11.6 Automatic LCD Backlight Control

In the MKT-View II (and in similar future devices with TFT display), the display brightness can be automatically controlled. An integrated ambient light sensor can be configured to reduce the backlight illumination at night, to avoid blinding the driver or operator.

To configure the LCD backlight control, enter the "System Setup" (in the system menu), and select the item "LCD dimming...". It contains a submenu with the following entries :

- LCD-Brightness 'NIGHT' :
  This parameter defines the LCD brightness "at night" on a scale from 0 (=mininum brightness) to 255 (=max possible brightness)
- LCD-Brightness 'DAY' :
  This display brightness will be used, whenever the light sensor reading exceeds the value configured in the item "Light Sensor Maximum". See the diagram below for an explanation.
- Light Sensor Value, LOW :
  The ambient light sensor value for "Night". If the sensor returns this value (or less), the LCD brightness will be set to the "LCD-Brightness 'NIGHT'.
- Light Sensor Value, HIGH :
  The ambient light sensor value for "Day". If the sensor returns this value (or more), the LCD brightness will be set to the "LCD-Brightness 'DAY'.
- Low-power brightness :
  This display brightness will be used after a configurable time of "no activity", i.e. when there was neither a keystroke, rotary encoder event, or touchscreen event for as many seconds as configured in the system menu under "Display Setup".."LCD-OffTime".

If the ambient light sensor value is **between** the sensor values 'LOW' and 'HIGH', the LCD brightness follows the sensor value in a (more or less) linear function :

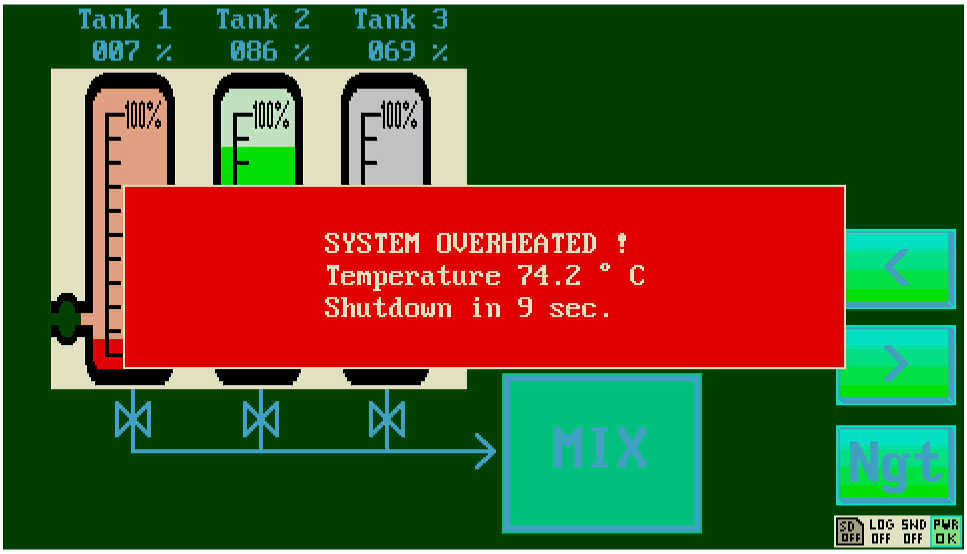


The last item in the "LCD Brightness / Dimming" menu is read-only. It shows the current value from the ambient light sensor ("Curr. Light Sensor Value"). The value ranges from 0 (="darkest night") to 255 (=direct sunlight).
The correlation between ambient light (in Lux) and sensor value (0..255) is logarithmic.

Notes on how to use the ambient light sensor in your display application, and how to implement a day/night colour scheme switching can be found in the online help system of the programming tool (file help/colours_01.htm).

1.1 Automatic Shutdown on excess temperature

In the MKT-View II / III / IV, the internal temperature is permanently monitored. When exceeding 70 °C, a red popup message appears with a warning, as in this example:



SYSTEM OVERHEATED !
Temperature 74.2 ° C
Shutdown in 9 sec.

If the temperature doesn't drop below the critical threshold within a few seconds, the device turns itself off to prevent hardware damage.

Note:
Under normal conditions (if the device is operated according to the specifications), you should never see this message. According to the hardware specification, the maximum temperature (measured on the surface of the metal enclosure) must never exceed 65 °C. In that case, the internal temperature remains below 70 °C, unless the TFT is exposed to direct sunlight.
BUT: In a passenger car, parked in the sun, the temperature behind the windscreen, above the dashboard, will easily reach and exceed 75° C - even without self-heating from electric power dissipation ! In such an extreme case, even the automatic shutdown of the device cannot prevent destruction of the TFT (which, according to the TFT manufacturer's datasheet, must not be operated above 70 °C).

In a few cases, operating the TFT above 70 °C over longer periods of time has already caused the destruction of the TFT panel ! If the device is too hot to touch, don't turn it on ! Instead, *get it out of the sun,* let it cool down, and try again.