UDP

Universal Display Protocol Version 3.33

Including eUDP Protocol Version 3.33

206-064-01	1	07-01-04	First Issue	GB	SB	SB
Part	**Issue**	**Date**	**Description**	**Produced**	**Checked**	**Approved**

UDP Version History

Doc version	Date	Date Remarks / Changes
1.00	Jan 1996	Preliminary version
2.00	Feb 1996	First release SEP introduced All parms in hex
2.01	Aug 1999	Extended UDP separated Commands added: 82 : clear A7 : vertical size E3 : overwrite II E4 : configure / security
3.00	Oct 2001	New software, Preliminary doc Commands removed: 81 : synchronise 82 : clear C2 : short pause C3 : horizontal position I C5 : overwrite I C7 : set output E3 : overwrite II E5 : memory dump Commands added : C7 : sound E5 : configure led test E6 : set time date Commands changed : A6 : inquiry 1, 2, 3, 4 A5 : enable E4 : configure display
3.10	Nov 2001	This version overview Config display command : Lx . L16 (fixed) Config led test . extended udp only Set time / date E5 . E6 (typing error) Outline and Hor. Position cmd. combination inhibit Extended udp : Nack 19 added Inquiry 2: max amount of lines 10 . 16 UDP data error bit 4: ledtest configuration Line status bit 1 removed, bit 2 . 1 Watchdog status bit 3 : 1=watchdog running Inquiry 3 : added parameter amount of leds tested Configure ledtest : disable ledtest MM>0.
3.20	Sep 2002	Commands added : A0 : Font 0 A3 : Outline

		A8 : Mirror A9 : Substitution AA : Or Pixel / Colour AB : Graphic Configuration E3 : Horizontal Pixel Configuration E7 : Output E8 : Month / Day Configuration Added extra information on Enable and Watchdog command.
3.31	March 2003	Commands changed : Inquiry 2 : Added two extra field with project name and project version.
3.32	March 2003	Commands changed : Inquiry 2 : UDP Data error E1 : Watchdog C6 : Pause command
3.33	April 2003	Commands changed : C6 : Pause command (Default time)

1 UDP SERIAL PROTOCOL INTRODUCTION

This document describes the Universal Display Protocol for serial communications with LED displays. The protocol is designed to support multiple display types with a single serial control structure.
In applications where simplicity or communication speed is more important than error detection and status control, the standard UDP is a suitable protocol. Simple microcontroller based devices re-send data continuously via relatively reliable communication lines.

Complex CRC algorithms are often undesirable or not required and the controlling systems rarely have an intelligent mechanism to anticipate possible error messages. For some display types, an extended version of the UDP protocol is available, offering the following extra's:

Communication error checking with packet CRC's.
Supporting large data structures with numbered packets
Return packets offering display status information

The extended UDP protocol is fully described in the appendix (section 6)

1.1 IMPLEMENTED FEATURES OF UDP:

In general, the characteristics supported by the UDP software can be summarised as follows:
Multiple displays: RS232 or Rs422/Rs485 network.
Multiple messages per display.
Multiple pages per message.
Multiple lines per page.
Flexible line size.
Multiple character fonts.
Multiple character features: flash, stretch, colour.
Features per page or line: appear, delay, disappear
Default definitions per display/message/page.
Overwrite text partially.
Various remote checks (inquiry, watchdog).
Flexible and effective protocol
Optional:
Clock/calendar/temperature display.
Conditional display (dep. on time, date, input levels).
Bit mapped graphics

The properties and physical limits of the individual displays determine whether a feature is actually included or not. Where necessary, defaults are adapted to the display type and non supported commands or parameters are ignored.

1.2 LEVEL STRUCTURES

Data is sent to the displays using addressed packets. The data in one or more of these packets may be subdivided in one or more messages. A message consists of one page or multiple pages, and a page consists of one or more lines. This results in the following structure:

Displays
Messages
Pages
Lines
Characters

To be able to address each of these levels, the same subdivision is used in the serial protocol. Each level is addressed by means of a number. To access a certain data structure, an address construction is used that (where necessary) successively specifies network (display) number, message number, page number and line number. The serial protocol scheme has 4 levels which can be visualised as follows:

```
SOT NN

DATA                                    ← display level
    STX MM
    DATA                                ← message level
        SOP PP
        DATA                            ← page level
            SOL LL
            DATA                        ← line level
        EOL
    EOP
    ETX
EOT
```

1.2.1 CONTROL CHARACTERS:

SOT = start of transmission
EOT = end of transmission
STX = start of message (text)
ETX = end of message (text)
SOP = start of page
EOP = end of page
SOL = start of line
EOL = end of line

Hex Numbers '00' .. 'FF':
NN = Network number
MM = Message number
PP = Page number
LL = Line number

In the above protocol scheme the levels for messages, pages and lines are optional. Any of these levels may be skipped if not required. In the most simple application, where no multiple pages or messages are required, e.g. on a single line display, only the highest level can be specified, as shown below :

SOT NN DATA EOT

1.3 DEFAULTS AND ZERO STRUCTURES

For each level, commands can be applied that act on the current structure and the substructures on its lower levels, if present. Commands that affect certain settings, overrule the settings made by higher levels. A parameter is effective on the current level (where it was set) and all lower levels, as long as it is not overruled again by other (local) commands.

Structures with number zero have a special function. If no lower level is specified, the data goes automatically to item number 0 of that lower level. At the same time, the values set by commands in that data act as a default for all other items on that level.

For example, the commands that are specified on display level, are actually stored in message 0, and act as a default for all other messages. In the above example, where no sub level is specified at all, the display finally stores the data as line 0 of page 0 of message 0. The reason for this is that all data ultimately has to be stored on the lowest level, the line level.

While actually displaying the programmed data, the software interprets the data in a sequential order byte by byte, and acts as consistent and logical as possible, on each level of the structure. The mechanism that is responsible for this is referred to here as the display parser.

During the displaying process, the items on each level are parsed in sequence (the lines are parsed parallel), but before each item, default item 0 is parsed in between. A sequence of pages would be parsed in this order: page 0, page 1, page 0, page 2, page 0, page 3…

This accentuates the default effect of items with number 0. Not only their *values* are used as default, their display time commands are actually executed each time.

Upon finishing a certain level and continuing further execution one level higher, the parser restores the settings that belong to that level as if taking these settings from a virtual 'stack', that acts as temporary storage. This way, parameters can be set for an entire structure, and will act as a default value for the lower levels. When this principle is used properly, the required memory use and communication time can be reduced to a minimum.

1.4 THE SEPARATOR

The commands that follow immediately after the network number, message number, page number and line number are executed immediately at communication time.

Commands that follow the separator (SEP), are stored in memory, together with possible text characters (or graphics), on the level where the command was placed.

These commands are interpreted by the parser at display time. Even more, these commands are not checked for validates during communication. Displays that do not support certain commands will simply ignore them.

Although the protocol allows almost any combination of commands and levels, commands are usually dedicated to be used either only at communication time, or only at display time, and some (like the sound command) support both.

SOT NN
commands ← comm. time commands display level
SEP commands ← display time commands display level
    SOL 01
    commands ← comm. time commands line level
    SEP commands & text ← display time commands & text line level
    EOL
EOT

1.5 MEMORY MANAGEMENT

Upon sending data to a non-existing line, page or message, this page and/or message will be created (inserted) automatically. When already existing, the new data will replace (overwrite) the current data. A special command is used to delete a line, a page, a message or all messages from the display's memory. Inside the display, a double memory buffer is used to keep any visible interference with currently displayed data during serial update to a minimum.

During communication, the data is stored in battery backuped memory together with message based checksums. At power up, the checksums are verified. Messages will be deleted automatically if a checksum failure is detected. Also, if the checksum of the main message table fails, the entire memory will be cleared. Under normal circumstances, this does not happen. The software is designed to avoid unpredictable situations even in case of e.g. random power downs during communications.

1.6 LINE MODE VS. PAGE MODE

Depending on the way the display is programmed, it operates in either the “line mode” or in “page mode”.

1.6.1 LINE MODE

In line mode, each line runs its cycle, *independent* from the other lines. A display will operate in line mode if it contains one single message nr 0 with only one page nr 0. Instead of specifying these message and page levels, they can be omitted completely. The following packet format remains:

SOT NN

SEP DATA ← defaults
SOL 01 SEP DATA EOL ← line 1
SOL 02 SEP DATA EOL ← line 2
EOT

1.6.2 PAGE MODE

In page mode, the display parser waits until all lines have been finished, before moving on to the next page (which can also be the same page again). This means that all lines start at the same time, they are more or less synchronised. A display will operate in page mode if it is not programmed with only one page 0 in one message 0. This means that at least a non-zero message number or a non-zero page number must be specified :

SOT NN

SEP DATA ← defaults
SOP 01 ← page
SOL 01 SEP DATA EOL ← line 1
SOL 01 SEP DATA EOL ← line 2
EOP
EOT

2 COMMANDS

A variety of commands is implemented in UDP, to manipulate the way how text is presented on the display, and to control the special functions. The effect of a command does not only depend on the type of command, but also on the exact place (level) where it is applied. Usually, the addressed item plus its lower level structures are affected. Furthermore, a distinction applies between two classes of commands:

Communication time commands: Immediately executed (once). In the protocol, the position of these commands is at the outset of a level, before the SEP character (if any).

Display time commands: Stored with text, interpreted by display parser at every cycle. In the protocol, the position of these commands is after the SEP character.

Some commands can be used in either way.

Commands consist of escape sequences with the following form:
Esc Cmd [Par] [EOC]

Where

Esc = escape control character
Cmd = command identifier. See below
Par = optional parameter character(s)
EOC = optional end of command

Dependent on the amount of parameter characters used, the commands are subdivided in 4 groups:

Group	***Range***	***Amount of parameter chars***
1	80..9F	0
2	A0..BF	1
3	C0..DF	2
4	E0..FF	3 or more, terminated with EOC

The bit map of the Cmd identifier indicates to which group the command pertains. The following applies the Cmd byte, if represented as 8 bits, from D7 to D0:

D7 = most significant bit is always set to 1
D6, D5 = group indication: 0, 0: group 1
0, 1: group 2
1, 0: group 3
1, 1: group 4
D4..D0 = command type number.

2.1 UDP COMMAND SUMMARY

The following is an overview of standard commands of the UDP protocol:

Description	*Grp*	*Cmd*	*Parameter range*	**D**	**M**	**P**	
Delete	1		-	B	-	B	B
Disable Preparser [2]	1	$9F	-	B	-	-	-
Font	2	$A0	0..	A	A	A	A
Colour	2	$A1	0..3	A	A	A	A
Horizontal size	2	$A2	1..15	A	A	A	A
Outline	2	$A3	0=left 1=cent 2=right	A	A	A	A
Flash	2	$A4	0=Off, 1=On	-	-	-	A
Enable	2	$A5	0..7, bit mapped	B	B	B	B
Inquiry	2	$A6	(extended udp only)	E	E	E	E
Vertical size	2	$A7	1..15	A	A	A	A
Mirror	2	$A8	0..3, Bit mapped	A	A	A	A
Substitution	2	$A9	0=Off, 1=On	A	A	A	A
Or Pixel / Colour	2	$AA	0..3, Bit mapped	A	A	A	A
Graphic Configuration	2	$AB	0..7, Bit mapped	A	A	A	A
Appear feature	3	$C0	0..2 see list	A	A	A	A
Disappear feature	3	$C1	0..2 see list	A	A	A	A
Vertical position	3	$C4	-128..+127	A	A	A	A
Pause	3	$C6	0..255	A	A	A	A
Sound	3	$C7	0..255	X	X	X	X
Brightness table [1]	4	$E0	16 steps of 16 levels	B	-	-	-
Watchdog	4	$E1	timeout, message nr	B	-	-	-
Horizontal position	4	$E2	-2048..+2047	-	-	-	A
Horizontal Dots configuration [1]	4	$E3	Dots hor., line sizes	B	-	-	-
Configure display [1]	4		line sizes, modules	B	-	-	-
[1]		$E5	(extended udp only)		-	-	
Set time / date	4	$E6		B	-	-	-
Output		$E7	Sets Output high / low		X	X	
Month / Day_1	4	$E8	Month's and day's strings	B		-	-
[1]							
		$E9	Limit the dissipation of the display	B			

1 This command is stored in memory - up (or when the checksum failed, defaults are valid) and could not be removed by using the Delete ($80) command. The command can only be restored by sending the command again with the default settings.

2 Command is reset to default state when a command mentioned in note 1 is send or when the display is switched off and on.

D = Display level
M = Message level
P = Page level
L = Line level

A = Display time command, to be placed AFTER separator
B = Communication time command, to be placed BEFORE separator
X = Can be used as communication and display time command
E = See extended UDP documentation for specific information
- = Command not defined for this level

2.2 DELETE

syntax : Esc, $80
levels : display, message, page, line
class : communication-time command
default : none

Can be used to delete a line, a page, a message or the entire display's memory (all messages). The display will immediately exclude the deleted structure, and if just shown, it is removed from screen. If the delete command involves data that is not currently shown on screen, no visible effect should occur.

Upon receiving a delete command, the display may have to re-organise its internal memory and when ready, an acknowledgement is sent.

The following example shows how to delete page 3 of message 2 of display 1.

SOT '01' STX '02' SOP '03' Esc $80 EOP ETX EOT

The following example shows how to delete the entire contents of display 10.

SOT '0A' Esc $80 EOT

2.3 DISABLE PREPARSER

syntax : Esc, $9F

levels : display

class : communication-time command

default : Enabled

Sending this command will disable the preparser and therefore the auto scroll is disabled.

This command is useful when graphics are used with a lot of data so not every byte is counted. The preparser is reset to default state under three conditions,

- Switching the display off and on,
- Sending the delete command $80,
- Sending a display configuration command, (see note 2, section 2.1, UDP Command Summary)

2.4 FONT SELECT

syntax : Esc, $A0, F
levels : display, message, page
class : display-time command
default : 1

Parameter F is a hex representation of the font number (0..15).
Font number 0 enables bit mapped graphics.
For more information on bit mapped graphics see the graphic configuration command.
For more information about the supported fonts check the version specific properties.

2.5 COLOUR SELECT

syntax : Esc, $A1, C
levels : display, message, page, line/char.
Class : display-time command
default : 1

The colour select command defines the (default) colour for the current level and sub levels. On line level, the command defines the colour of the subsequent characters.

Parameter C is a hex representation of the colour number (0..15). The display default value of the colour parameter is for all display types number 1. Colour number 1 does not always represent the same particular colour. With monochrome displays, where only one colour is supported, colour number 1 is the colour of the led's fitted in the displays. The following colours have been implemented:

0 : no colour (“black”)
1 : if tri-colour: red, else: colour of leds. Default.
2 : if tri-colour: green
3 : if tri-colour: yellow

The following example shows two lines on a tri-colour display, the first line is green, the second is red (colour 1, default).

The colour command is applied at line level:
SOT NN

SOL 01 SEP Esc $A1 $32 STRING EOL
SOL 02 SEP STRING EOL
EOT
In the next example, the same colour command is used at display level. Both lines are green.

SOT NN SEP Esc $A1 $32
SOL 01 SEP string EOL
SOL 02 SEP string EOL
EOT

2.6 HORIZONTAL SIZE

syntax : Esc, $A2, S
levels : display, message, page, line.
class : display-time command
default : 1

The horizontal size command sets the horizontal character enlarge factor. Parameter S is a hex representation of the horizontal enlarge factor. Default factor is '1'. With S='2' the characters are shown in double width. The maximum magnifying factor is 15 (S='F').

2.7 OUTLINE

syntax : Esc, $A3, T
levels : all: display, message, page, line/char
class : display-time command
default : 0

This command specifies the outline type for the subsequent appear features on this level, or sub-levels. Parameter T is the hex representation of one of the following values:

0 : aligned to the left (default)
1 : centred to middle of display
2 : aligned to the right

The outline command may not be combined with the horizontal position command. Attempts to do so will give unpredictable results.

2.8 FLASH

syntax : Esc, $A4, F
levels : line
class : display-time command
default : 0 (off), reset at begin of each line

The flash command switches the 1 Hz flash mode either on or off. These switches may be placed anywhere in the character strings. The flash function can only be made visible in combination with the pause command. During every second pause, the flashing characters are put on during the first 1/2 second and put off during the second half. The combination of flash and horizontal scroll features is not supported.

The parameter F is either '0' or '1' to set the flash off or on respectively.

2.9 ENABLE

syntax : Esc, $A5, E
levels : display, message, page, line
class : communication-time command
default : 0

This command is used to enable or disable structures to be displayed, either immediately or not. When a structure is disabled, it is not removed from memory.
Parameter E is a hex coded half byte, bit mapped as follows:

Bit	***function***
0	disable (hide)
1	execute one time
2	execute immediately
3	reserved

2.9.1 Disable

When the disable flag of an item is set, it is excluded from the display cyclus, although it is not deleted from the memory. This function is normally used in combination with other functions. A message that is programmed to be 'hidden' can be activated by certain events, like the expiration of the watchdog time.

2.9.2 Execute One Time

After having displayed the item once, the message is disabled but not deleted from memory. Some applications require external control of the display timing process. Instead of letting the display repeating the cycles independently, the controlling system can determine the start timing of each new text. This often requires a decent synchronisation between the display and the controlling system, whereby the latter can follow the status of the display each moment. See also inquiry 2 (extended UDP in appendix).

2.9.3 Execute Immediately

This function will (without the disable bit set) immediately interrupt the displaying cyclus, stop any display feature, remove the current item from the screen, and start displaying the selected item. Although this may not give a visually attractive effect, it is sometimes the only way to make sure that very urgent messages ("Stand back, train approaching") are shown immediately.

The enable command can be used for the following purposes:

To (re)start a message immediately
To "hide" a message, to let it appear with the watchdog command
To "hide" a message, to let it appear with the enable command
To show a message only once

Notes :

The enable command is a communication time command (before SEP)
The enable command must be send AFTER the message on which it acts.
If the display is used in "line mode", the enable command can be applied in message 0.

Possible combinations of these functions :

E	***bit 2***	***bit 1***	***Bit 0***	***Function***
0	0	0	0	enable , continue cycles at current point
1	0	0	1	disable, but finish if currently on display
2	0	1	0	include in cycles only one time
3	0	1	1	(no valid combination)
4	1	0	0	show immediately, interrupt current item
5	1	0	1	disable and stop displaying immediately
6	1	1	0	show immediately, but one time only.
7	1	1	1	(no valid combination)

Please check the specifications of your product carefully for further details on this command. Some of the functions are not available in certain products, or have a different code.

2.10 INQUIRY

syntax : Esc, $A6, I
levels : see extended UDP in appendix
class : communication-time command
default : none

With the inquiry commands, information can be retrieved from the display. This can serve various purposes, like debugging, monitoring and service. The following sub commands are defined:

Inquiry 1 : BIOS level information
Inquiry 2 : UDP status information
Inquiry 3 : Self test results
Inquiry 4 : Page monitor
Inquiry 5 : Temperature limitation statistics

For more information about inquiry commands, see the extended UDP in ection 6 appendix.

2.11 VERTICAL SIZE

syntax : Esc, $A7, S
levels : display, message, page, line.
class : display-time command
default : 1

The vertical size command sets the vertical character enlarge factor. Parameter S is a hex representation of the vertical enlarge factor. Default factor is '1'. With S='2' the characters are shown in double height. The maximum magnifying factor is 15. Pixels that exceed the top and bottom limits of the lines will not be displayed.

2.12 MIRROR

syntax : Esc, $A8, M
levels : display, message, page, line.
class : display-time command
default : 0

This command is used for mirroring the display (scan) and characters. Parameter M is the hex representation of a bit mapped character, with the following functions:

Bit nr	***function***
0	Mirror display (Scan)
1	Mirror characters
2..7	not defined

2.13 SUBSTITUTION

syntax : Esc, $A9, S
levels : display, message, page, line.
class : display-time command
default : Off

In earlier versions UDP firmware version (<1.81), a certain range in the character set was reserved for so called “substitution” characters of “real time variable characters”. These characters show a changeable value, such as the time or date or temperature, rather than one fixed character from the font. In later UDP firmware versions (>= 1.81), the number of substitution characters had increased to such an amount that conflicts with the standard code page definitions became problematic. To solve this, a new command was introduced to separate the variables from the fixed fonts by means of a switch.

Value of S	***Effect***
0	Substitution disabled, only fixed font characters shown (default)
1	Substitution enabled, certain values show real time variables.

Which substitution characters are supported may vary with the version of the firmware.
Check the UDP version specific properties documents for more information, especially UDP version before 1.81. From version 1.81, the following substitution characters are implemented,

Character	Meaning
176	Hours tenths
177	Hours units
178	Min tenths
179	Min units
180	Semicolon
181	Sec tenths
182	Sec units
183	Datum tenths
184	Datum units
185	1st day character
186	2nd day character
187	3rd day character
188	Month tenths
189	Month units
190	1 st month character
191	2 nd month character
192	3 rd month character
193	Year tenth
194	Year units
195	1 st temp sign (±)
196	2 nd temp character
197	3 rd temp character
198	Degree sign symbol

2.14 OR PIXEL / COLOUR

syntax : Esc, $AA, O
levels : display, message, page, line.
class : display-time command
default : 0

This command is usually applied in combination with the Horizontal position command. The Horizontal position command can be used to return to a previous position on a line, to add text or graphic information in a different colour or on a different vertical position. Normally, rewriting text or graphics would clear all information previously written onto this positions. The OR command can be used to keep the former images intact, and ADD the (text or graphic) pixels to the existing image. Some practical applications are :

Showing two small characters above each other on one line.
Overlapping two graphics, one red and one green portion (1-colour mode).

Parameter O is the hex representation of a bit mapped character, with the following functions:

Bit nr	***function***
0	OR pixels
2	OR colours
2..7	not defined

The following combinations are possible :

Value 0	***function***
0	do not OR images, clear while overwrite (default)
1	OR pixels, the colour is determined by the new information
2	OR colour, but clear existing pixels
3	OR pixels and colour, red and green is combined to yellow

Example,

```
ESC $80                                  ; Delete
SOL '01'                                 ; Line 1
        SEP
        ESC $A0 '2'                      ; Big font
        'AEX'                            ; Text
        ESC $A0 '1'                      ; Small font
        'AHL'                            ; Text
        ESC $AA '3'                      ; OR Pixels / Colour
        ESC $E2 $30 $31 $46 0C           ; Set horizontal position back
        ESC $C4 $30 $38                  ; Set vertical position
        ESC $A1 '2'                      ; Set Green colour
        €1.70'
EOL                                      ; End Of Line
```

2.15 GRAPHIC CONFIGURATION

syntax : Esc, $AB, G
levels : display, message, page, line.
class : display-time command
default : 1 (1 colour, 8 pixel data)

By selecting font number 0, the display shows bit mapped graphics instead of text characters. The bit mapped graphics are defined per vertical column of 8 or 16 pixels. The most significant bit corresponds with the top pixel, the least significant bit with the bottom pixel. The bytes are send in hexadecimal format, with the most significant nibble first (top) or in case of 16 bits, compression could be used.

2.15.1 8 PIXEL MODE

If the 8 pixel mode is selected, each column is defined by one byte (8 bits), thus by two hexadecimal characters.

2.15.2 16 PIXEL MODE

If the 16 pixel mode is selected, each column is defined by two bytes (16 bits), thus by four hexadecimal characters.

2.15.3 1 COLOUR MODE

In 1-colour mode, the graphics data is defined for only one colour. This colour is defined by the colour command ($A1). To obtain multiple colours, the same graphiccan be re-written in different colours, using the Horizontal position command and theOR command.

Note : this method can not be used on scrolling lines, because the horizontal position command does not work on scrolling lines.

2.15.4 2 COLOUR MODE

In 2-colour mode, the graphics data is defined for red and green alternately. The first column defines the red portion, the next column defines the green portion for the same position. The next column defines the red portion for the next position, etc.. In 16 pixel mode, each 2-colour column require 8 hexadecimal characters to be send. The colour command ($A1) has no effect in 2-colour mode.

The bit mapped hex representation of G has the following value,

Bits (G)	***Meaning***
1,0	00b, Not used. Future expansion, 01b, 8 pixel data (1 byte), (Default, bit 3 automatically cleared) 10b, 16 pixel data (2 bytes), 11b, Not used. Future expansion
2	0b, 1 Colour (Default) 1b, 2 Colour
3	0b, Graphic compression off 16 pixel data (Default) 1b, Graphic compression on 16 pixel data
7..4	x, Don't care

The following combinations are possible,

Hex value of G	***Meaning***
1	1 Colour / 8 Bits (Default)
2	1 Colour / 16 Bits (No Graphic compression)
5	2 Colour / 8 Bits
6	2 Colour / 16 Bits (No Graphic Compression)
A	1 Colour / 16 Bits with graphic compression
E	2 Colour / 16 Bits with Graphic Compression
0,3,4,7,8,9,C,D,F	Not used.

2.15.5 Graphic Compression

There are two ways of graphic compression which can be used together,

1) Insert the same column multiple times using the option h, r and R,
2) 16 bit compression, 4 bytes converted to 3 bytes.

Add 1

A column is inserted more then one time when one of the following characters is detected,

h : print the column 2 times,
r<n> : print the column n times with n the hex representation of 0 .. 15,
R<nn> : print the column nn times with nn the hex representation of 00 .. 255.

When n or nn equals 0 the column is printed one time. The advantage of h versus r2 is one byte (no overhead). The advantage of r4 above R04 is also one byte, therefore no overhead.

This form of compression is in depended of the graphic configuration.

Add 2

When 16 bit graphics are used normally the 16 bits are represented by 4 hexadecimal bytes. Therefore a column 0001001000110100 is send as 31h, 32h, 33h, 34h. When large bmp pictures are used this causes overhead and compression of the 4 bytes gives of reduction 25%. A compression byte starts with bit7 = 1 added with 7 column bits. Therefore 3 compression bytes are used. Applying the compression rule gives the next result,

Compression byte 1 : 10001001 = 89h,
Compression byte 2 : 10001101 = 8Dh
Compression byte 3 : 100xxxxx with x = don't care, = 80h

Example 1, Multiple columns,

```
ESC $80                         :Delete
SOL '01'                        :Line 01
        SEP
        ESC $A0 '0'             ; Enable Bit mapped
        ESC $AB '2'             ; Graphic configuration, 1 colour / 16 Bits
        '1234'                  ; graphic data 4 bytes
        'RFF'                   ; Column is printed 255 times
EOL                             ; End Of Line
```

Example 2, Compression, Multiple columns,

```
ESC $80 ; Delete
SOL '01'                        ; Line 01
        SEP
        ESC $A0 '0'             ; Enable Bit mapped
        ESC $AB 'A'             ; 1 colour / 16 Bits / Compressed
        HEX '89 8D 80'          ; graphic data 3 bytes
        'RFF'                   ; Column is printed 255 times
EOL ; End Of Line
```

Example 3,

Example colour display with a line height of 16 rows.
Example shows a RED arrow with a Yellow square.

```
ESC $80                                 ; Delete
SOL '01'                                ; Line 01
        SEP
        ESC $A0 '0'                     ; Enable Bit mapped
        ESC $AB '6'                     ; 2 colour / 16 Bits
        'FFFFFFFF8001800180818001'              ; graphic data 4 bytes RED, 4 Green etc
        '80C1800180E1800180F18001'
        '9FF980019FFD80019FFD8001'
        '9FF9800180F1800180E18001'
        '80C1800180818001800180018001'
        'FFFFFFFF'
EOL                                     ; End Of Line
```

2.16 APPEAR FEATURE

syntax : Esc, $C0, AA
levels : display, message, page, line
class : display-time command
default : 0

To specify the visual effect used to let a line of text appear on the display screen. The two characters AA are the hex representation of the feature number:

AA	*Feature Name*	*Description of the visual effect*
00	Jump	text appears at once
01	scroll (horizontal)	text appears scrolling from right to left
02	auto scroll	jump if the text fits one the screen, else scroll
03	vertical (up)	text appears scrolling up from the bottom to the top

The appear feature command itself does not invoke the execution of the feature, it merely sets the appear parameter. The features starts being executed as soon as a printable character (or graphic) is encountered by the display parser. The piece of text string (including commands) that is subject to the appear feature is delimited by this first character and one of the following possible end-conditions:

the end of the line
another appear feature command
a disappear feature command

Immediately after the feature has been executed, a pause is carried out, according to the current value of the pause parameter. See pause command.

After the pause, the disappear feature is carried out. After the disappear feature, the parser continues tracing through the line, possibly encountering another piece of string, in which case the above described sequence starts again.

Jump, vertical and other (optional) features behave more or less according to the above described standard procedure. The horizontal scroll feature behaves a bit differently in some situations :

Event	*Non-scroll features*	*Scroll feature*
text is larger than display	text is truncated	no problem, the scroll feature can handle long strings
text contains (a) flash commands	flash is executed during a pause only	flash is not executed during a scroll
text contains (a) pause commands	pause command is not executed at that point	pause command is executed its point in the text
end of feature (see end conditions above)	execute pause, then the disappear command	Scroll the text off the screen, unless explicitly specified otherwise by commands

2.17 DISAPPEAR FEATURE

syntax : Esc, $C1, DD
levels : display, message, page, line
class : display-time command
default : 0
To specify the visual effect used to let a line of characters disappear from the display screen. The two characters DD are the hex representation of the feature number:

DD	***Feature Name***	***Description of the visual effect***
00	Jump	text disappears at once
01	scroll (horizontal)	text disappears scrolling from right to left
02	Stay	text does not disappear
03	vertical (up)	text disappears scrolling up from the bottom to the top

Disappear features are used to emphasise the changeover of successive lines of information. In general, disappear features have the same effect as the corresponding appear features would have when applied to an empty screen, except disappear feature number 2. “Stay” means that the current information disappears during the next appear function, as if it was pushed away by the new text.

Disappear features are executed after a pause following an appear. For non-scroll features, the order is always : appear, delay, disappear. For more details on features, see also appear feature command.

2.18 VERTICAL POSITION

syntax : Esc, $C4, PP
levels : display, message, page, line/char
class : display-time command
default : 0

This command is used to set the vertical signed (positive or negative) start position (in pixels) where text characters or graphics are placed on the line. Parameter PP is a two byte hex representation of the offset value, ranging from -128 to + 127.

The display will not allow any character to go over the top or bottom limits of the line. A positive value moves the characters up from the base line. When the character size plus this offset is more than the line size, the exceeding top pixels of the character will be invisible.

A negative value lets the characters sink under the base line, whereby the bottom pixels of the characters will disappear accordingly.

2.19 PAUSE

syntax : Esc, $C6, PP
levels : display, message, page, line
class : display-time command
default : 1

The pause command specifies the amount of seconds to wait after an appear feature has been completed. During this time, the screen image remains static on the display. If the appear feature was NOT a horizontal scroll, then the flash feature and the substituted characters (time, date, etc) are refreshed every second.

The two parameters PP are the hexadecimal representation of the number of seconds (0..255) When the parameters PP = 255 is executed the pause is forever. This means that the current page will never end and when all features are finished, the page stays. Every other page, line will not be shown. Use the enable command ($A5) to force another message.

2.20 SOUND

syntax : Esc, $C7, SS
levels : display, message, page, line
class : communication-time or display-time command
default : none

The sound command starts the playback of one of the programmed sounds on the sound module. Parameter SS is the number of the WAV file in the module. The maximum value of SS depends on the way how the module was programmed. The shorter the sounds, the more sounds can be programmed.

The sound command can be programmed as communication time command or as display time command. The result is very different. When programmed as communication time command, the sound is invoked immediately and only once. When programmed as display time command (in a certain message, page or line) the sound is invoked each time when the sound command passes the display parser while the accompanying text appears on the screen.

2.21 BRIGHTNESS TABLE

syntax : Esc, $E0, TABLE(16 characters), EOC
levels : display
class : communication-time command
default : depends on display and led type

The brightness intensity of the display is adjusted automatically to the surrounding light level.The relationship between the value measured by the photo sensor and the display brightness is defined by a table of 16 entries. Each entry has a value from 0 to 15, where 15 is the maximum brightness. The first entry sets the brightness of the display in a dark environment, the last entry corresponds with direct sunlight.

Each value is sent as a 1 char hexadecimal representation (‘0’..’F’).

2.22 WATCHDOG

syntax : Esc, $E1, FF, TT, [MM], EOC
levels : display
class : communication-time command
default : off

Is used to assure that the display does not show old information if, for any reason, the controlling system or communication fails seriously. The command resets an internal timer at every transmission. If the time TT expires before the next watchdog command is received, the display starts displaying the message MM, or clears the screen. Parameter TT and optional parameter MM are hexadecimal representation of respectively the time (0..255 seconds, minutes, hours) and the message number (0..255).

The two FF characters are the hex representation of a bit mapped flag:

bit nr	***value***	***Meaning***
1,0	0,0 0,1 1,0 1,1	disable watchdog count down TT in seconds count down TT in minutes count down TT in hours
2	0 1	clear display (MM has no meaning) show message MM
3	0	Reserved

Message MM in this case will normally contain a text to report a “system error”. Use the Enable command to program message MM as a “hidden” message, so that it will not appear as part of the normal display cycles.

Notes:
The accuracy of the time period is “minus 0 plus 1”. To ensure the best relative accuracy, use large numbers of small units. For example, if the time out period was programmed as 1 hour, the watchdog may take between 1 or 2 hours to expire. If 60 minutes was programmed, the period would be between 60 and 61 minutes.
Use the watchdog command as follows:

1) Send the watchdog message.
2) Disable this message with bit 0 of the Enable command.
3) Send the watchdog command.
4) Keep repeating the watchdog command regularly.

2.23 HORIZONTAL POSITION

syntax : Esc, $E2, PPP, EOC
levels : line
class : display-time command
default : 0 = first pixel on the left.

This command is used to set the horizontal start position (in pixels) where subsequent characters or graphics are placed on the line. Specially when using proportional fonts, this is a helpful command, for example to make tables.

The 3 byte hex representation of the horizontal offset has a range from -2048 to +2047. Extreme positive or negative offset values can be used to chain multiple displays together horizontally, which goes beyond the purpose this documentation.

The horizontal position command may not be combined with the outline command.
Attempts to do so will give unpredictable results.

2.24 CONFIGURE DISPLAY (PIXELS)

syntax : Esc, $E3, DDDD, L1,L2, L3…L15,L16, EOC
levels : display
class : communication-time command
default : max amount of horizontal dots, lines equal to hardware lines

This command has the same function as the Configure Display command ($E4). The only difference is that this command ($E3) defines the horizontal size in pixels, where the other command ($E4) defines the horizontal size in boards. A pixel wise definition is more accurate and allows more variation in hardware configuration, like combinations of boards with different length’s, or partially equipped boards.
DDDD is a 4 byte hex representation of the total horizontal size of the display in pixels. The other parameters and functions are identical to the Configure display command ($E4). For more information, see the section of this manual that describes this command.

2.25 CONFIGURE DISPLAY

syntax : Esc, $E4, BB, L1,L2, L3…L15,L16, EOC
levels : display
class : communication-time command
default : max amount of boards, lines equal to hardware lines

The display configure command determines how the led matrix is to be subdivided in lines (from which the total vertical size is calculated) and what the horizontal size of the display is, counted in display modules. The display configuration not only fixes the matrix composition, it also settles the memory organisation.

Therefore does sending this command not only clear the screen, but also clear the entire memory including all messages, pages and lines.

Many types of display boards do not allow much variation in line size, because the pixels are organised line wise by the hardware, with physical gaps in between. However, the full-matrix types of display boards allow any combination of line heights to be configured, although each product will have certain limits.

In any case a display needs to be configured at least once with this command, as part of its initialisation, to ensure proper operation.

The display configuration can be changed as often as desired, for example to change the screen layout in between text transmissions. In that case, a proper synchronisation is required between the controlling system and the display, using the inquiry 2 command.

Parm	***chars***	***Dec / Hex***	***Description***	***Minimum Value***	***Maximum Value***
BB	2	H	Hor. Amount of boards	1	8
Lx	2	H		0=line not used	16

2.26 CONFIGURE LED TEST

syntax : Esc, $E5 HH MM EOC
levels : display
class : communication-time command
default : 24:00 = led test off

The led test configure command sets the time at which the led test procedure is started. For more information about the led test command, see the description of Inquiry command 3 (led test results, extended UDP in appendix).

2.27 SET TIME / DATE

syntax : Esc, $E6, YY, MM, DD, HH, mm, SS, [ZZ], EOC
levels : display
class : communication-time command
default : none

The set time/date command synchronises the display's on board system time and date with the time and date of the controlling system. The deviation is normally not more than one second per day. During power down, the display's on board time/date chip keeps running on a lithium cell. Synchronisation of the time and date can be done as often as desired, but actually needs to be done only occasionally, or during daylight saving time changeovers.

Parm	*chars*	*Dec / Hex*	*Description*	*Minimum Value*	*Maximum Value*
YY	2	D	Year mod 100[1]	00	99
MM	2	D	Month	01	12
DD	2	D	Date	01	31
HH	2	D	Hours	00	23
Mm	2	D	Minutes	00	59
SS	2	D	Seconds	00	59
ZZ[2]	2	D	1/100 seconds	00	99

[1] note 1: for the calculation of day of week, the display assumes the year to be between 2000 and 2099

note 2: a leap year fault will occur in the year 2100.

[2] note 1 : From ramcode version 2.01 ZZ is optional. Therefore the command is downwards compatible.

2.28 OUTPUT

syntax : Esc, $E7, <string>, EOC
levels : display, message, page, line
class : communication time or display-time command
default : output = low at reset

With this command, the output pin IO_1 on the controller board (FDS101) can be set to a high or low logic level. This signal can then be used to drive a relay or other equipment, to activate a device that produces an audible or visible signal. The command can be used to make the output simply high or low, or to produce a sequence of high and low levels with definable delays in between.
The <string> has a maximum length of 32 of the following characters:

'+'	= command to make output high
'-'	= command to make output low
'0'..'99'	= delay of max 9.9 seconds (units of 0.1 sec)

For an accurate calculation of the timing, an extra delay of 0.1 seconds must be included for each step.

Esc $E7 + EOC	Makes output high
Esc $E7 - EOC	Makes output low
Esc $E7 + - + - EOC	Two pulses of 0.1 second with a pause of 0.1sec in between
Esc $E7 + 0 – 0 + 0 - EOC	Two pulses of 0.2 second with a pause of 0.2 sec in between
Esc $E7 + 3 – 8 + 13 – EOC	Produces one pulse of 0,5 second and one of 1,5 second, with 1 second in between

Notes:

If the command is repeated before the previous command was finished, the latter command will interrupt the first, and take over control of the output.

The command can be used either communication time (before the SEP) or display time (after the SEP). A combination of the two will give unpredictable results and must be avoided.

The Sound command ($C7) uses the same output pin IO_1 as the Output command. Any combination of these two commands must be avoided or else unpredictable things may occur.

2.29 MONTH / DAY CONFIGURATION

Syntax : Esc, $E8, MONTH1, MONTH2, ..., MONTH12, DAY1, DAY2,.., DAY7 EOC
levels : display
class : communication time command
default : English

This command is used to configure another Month / Day string other than the default English 3 character abbreviation.

Each MONTH(1..12) and each DAY(1..7) is represented by 3 characters.
Access to the month / day table can be done using the substitution characters.

Default MONTH abbreviation	***Default DAY abbreviation***
JAN	MON
FEB	TUE
MAR	WED
APR	THU
MAY	FRI
JUN	SAT
JUL	SUN
AUG	
SEP	
OCT	
NOV	
DEC	

2.30 TEMPERATURE LIMITATION

syntax : Esc, $E9, F, TT, B, EOC
levels : display
class : communication time command
default : Disabled (F='0')

High-resolution full matrix displays in small housings in combination with direct sunlight can cause undesirable high temperatures. This command can be used to limit the dissipation of the display. The temperature sensor input on the controller board of the display is used to measure the temperature.

Above temperature value TT, the brightness of the display is limited to value B.

The flag F is used to enable (F='1') or disable (F='0') the function.

Parameter TT is a hex representation of the temperature, only positive values are valid:
'00'..'7F'
.
Parameter B is a hex representation ('0'..'F') of the maximum value to which the display brightness is limited when the measured temperature is higher than TT.

If F='0' then the parameters TT and B are meaningless, but still have to be send.

Sending command $E9 resets the temperature limitation statistics (see inquiry 5).

3 SERIAL COMMUNICATIONS

The amount of bytes that can be received by the display at a time is limited to 255. Attempts to send larger packets to the display will inevitably cause (overrun) problems. The standard UDP protocol has no possibility to subdivide data over multiple packets (the extended UDP does have this option).

To get away with this limitation, the data must be sent by adding the smallest possible structures (lines or pages), one by one. Still, this means that the maximum size of these structures is limited. For larger displays, especially when using bit mapped graphics, it is advised to use the extended version of UDP.

3.1 ACKNOWLEDGEMENTS

Upon receiving a complete packet (with the appropriate address), the display first interprets the data and executes the commands. When ready, the display sends a single-character answer, indicating that it is ready to receive the next packet. The answer byte gives brief information about whether the preceding transmission was successful or not:

Answer [hex]	*Meaning*
30	Ok
31	Time out (max 100 mSec between bytes in packet)
32	Invalid/unexpected control code
33	Invalid hex or decimal character
34	Invalid message/page/line number
35	Invalid command
36	Invalid text character
37	Parameter value out of range
38	Insufficient memory error
39	General format error
41	Other error, multiple errors

3.2 RS 485

To support 2-wire RS-485 communications, the output driver is enabled immediate after having received the EOT character in the forward packet (if addressed correctly), and disables immediately after having transmitted the single character ack.

3.3 NETWORK NUMBERS

If the network number in the transmitted packet = 0 then all displays will accept the data, regardless of the display's network number. Also, if a display's network number is set to 0, it will always accept data, regardless of the network number in transmitted packet. If using Rs485 or Rs422 however, a display will not send back an answer, but keeps its output in high impedance state instead, if its address is 0 or if it is addressed with general address 0, although it does accept the data.

4 PRINTABLE CHARACTERS

Characters in the range from $20 to $AF are valid printable characters and comply with the standard ASCII definitions.

Some custom-made fonts however, only support a sub-set of these characters. Also, special symbols may be incorporated on certain positions. Not supported characters will be displayed as character number $20 (space).

Characters above $AF may be used in text strings as well, but these are usually reserved for special functions. When displayed, these characters are substituted by variable values (when the substitution = ON), like time and date, temperature. Check your version specific document for more information.

5 CONTROL CHARACTERS

The control characters mentioned throughout this document, are unique and may only be used for the purposes as described in this document. All characters with a value beneath $20 are considered as (non-printable) control characters and if not mentioned in the table below, considered as invalid control characters. Control characters may under no circumstances be included in a text string (except if representing a part of a command: ESC, EOC).

For this reason the UDP protocol uses hexadecimal notation in cases where parameters would otherwise possibly fall in the range of control codes, and accidentally be interpreted as such.

Name	***Value [Hex]***	***Description***
SOT	01	Start of transmission
STX	02	Start of message (text)
ETX	03	End of message (text)
EOT	04	End of transmission
ENQ	05	Forward packet
ACK	06	Answer packet
SOL	0A	Start of line
EOC	0C	End of command
EOL	0D	End of line
SOP	0E	Start of page
EOP	0F	End of page
ECS	1B	Start of command
SEP	1E	Separator

6 Appendix

Extended Universal Display Protocol Version 3.33

7 INTRODUCTION

This document is an appendix to the standard UDP document V3.33.

In contrast to the standard UDP, the extended UDP includes error checking by means of a CRC, packet numbering, and extensive answer packets. This makes the extended UPD suitable for large area applications, where communication errors have a potential to occur, where error checking is required, or where large data structures are used.

8 EXTENDED UDP PACKET FORMAT

One of the limitations of the standard UDP is caused by the maximum receive buffer size of the display. Since the standard UDP merely allows whole structures to be used within one packet, this implies a limitation to the maximum size of these structures. The extended UDP provides the possibility to share structures out among multiple packets, and thus allows structures of almost any size to be transmitted. To assure that the separate data parts are re-assembled correctly at the receive end, the extended UDP requires some extra packet data, such as packet numbers and checksums, which also requires some extra intelligence at the host system (in comparison with the standard UDP).

Although the command principles and level structures of the extended UDP is in general the same as the standard UDP, the packet format is quite different. To illustrate the difference, both packet formats are shown here successively. The packet format of the standard UDP:

SOT, NN, DATA, EOT

The packet format of the extended UDP:

SOT, TYPE, NN, ÑÑ, SS, LL, DATA, CRC, EOT

Field name	***Chars***	***Hex/ Dec***	***Description***
SOT	1	-	Start of transmission SOT
TYPE	1	-	Packet type forward (ENQ) or answer (ACK)
NN	2	H	Network number
ÑÑ	2	H	Network number, inverted
SS	2	H	Packet number / flags, see below
LL	2	H	Length of DATA
DATA	LL	-	DATA, max 240 chars, see below
CRC	4	H	16 bit cyclic redundancy check from TYPE to DATA
EOT	1	-	End of transmission EOT

Note : For all hex notations the most significant byte comes first.

Forward packets (sent to display) have the same format as returned answer packets, apart from the TYPE character, which is an ENQ in forward packets, and an ACK in answer packets.

A display is considered as addressed, if the NN field as well as the inverse of the ÑÑ field correspond with the network number of the display, as set by the dip switches (network number 0 has a special function, see below). If the network number does not comply with this, the display ignores the entire packet, and no answer is returned.
The maximum amount of bytes for the entire packet is 255. That leaves only 240 bytes for the DATA field. Multiple SOT bytes may be used to improve synchronisation, but the maximum amount of DATA bytes will decrease correspondingly.

If the total amount of DATA to be transmitted exceeds the maximum per packet, the DATA must be subdivided over multiple packets. The SS field keeps up the consecutive numbers of these packets, to assure that transmission is accomplished without missing or duplicated pieces.

For independent packets, the number in the SS field is arbitrary. The transmitted number in SS is simply duplicated in the answer packet. Still, it is advisable to use successive packet numbers to secure a relationship between forward packets and answer packets.
The first packet of a sequence may have any arbitrary start value, but each following packet within that sequence is obliged to use successive numbers.

The SS field is a two character hexadecimal representation of a byte that holds a packet number as well as a bit mapped packet status:

bit 5..0 : packet number, cyclic counter ...62,63,0,1...
bit 6, 7 : 0, 0 = independent packet (both first and last)
0, 1 = first packet of sequence
1, 0 = last packet of sequence
1, 1 = Intermediate sequence packet

8.1 RETURN PACKET FORMAT

The DATA field in the returned answer packet has the following format:

(N)ACK, PARM, [OPT]

where:

(N)ACK = 2 byte hex repr. of the (N)ACK type number ('00'..'FF')
PARM = 2 byte hex repr. of the (N)ACK parameter ('00'..'FF')
[OPT] = Optional data from the display, applied for enquiry commands.

(N)ACK value	***PARM***	***Description***
00	00 : no data in [opt] 01 : data in [opt]	ACK : positive acknowledgement, no errors detected ACK+answer : no errors, answer data in [opt] field
NACK		***Packet level errors :***
01	00	Timeout between bytes of packet. (max 100 msec)
02	PP	Invalid/unexpected control or hex character XX
03	00	CRC error
04		Length error
05	PP	Packet number error. PP=expected packet number
NACK		***Data level errors :***
10	PP	Invalid / unexpected control code PP
11	PP	Invalid hex or decimal char PP
12	PP	Invalid message, page or line number PP or level
13	PP	Invalid command PP
14	PP	Invalid text character PP
15	PP	Parameter PP out of range
16	PP	Memory error, memory full. PP= bit mapped error.*
17	00	General format error
18	00	Other errors, multiple errors
19	PP	Data size error, sending char PP

note : memory bit mapped errors are not explained here.

9 EXTENDED UDP COMMANDS

The extended UDP supports the same commands as the standard UDP, as well as some extra commands. These commands require extensive answer packets rather than a singlebyte ack. The following is an overview of commands which are exclusively supported by the extended UDP.

Description	***Grp***	***Cmd***	***Parameter range***	**D**	**M**	**P**	**L**
Inquiry 1 *	2	$A6	1: BIOS level info	B	-	-	-
Inquiry 2 *	2	$A6	2: UDP status info	B	-	-	-
Inquiry 3 *	2	$A6	3: Self test results	B	-	-	-
Inquiry 4 *	2	$A6	4: Page monitor	B	-	B	-
Inquiry 5 *	2	$A6	5: Temperature limitation statistics	B			
Configure led test	4	$E5	Time 00:00..23:59	B	-	-	-

Note : The Inquiry commands can't be sent in combination with other commands or data in the same packet, because this would cause a conflict in the data format of the returned packet. Attempts will invoke a returned error message.

9.1 INQUIRY 1 : BIOS LEVEL INFORMATION

syntax : Esc, $A6, $31
levels : display level only
class : communication-time command
default : none

Inquiry command 1 returns low level hardware related parameters, and general system (BIOS) parameters, which are not directly related to the UDP protocol. This information can be helpful for service and hardware maintenance.

Chars	***Dec / Hex***	***Description***
3	D	ROM BIOS version
4	H	ROM BIOS checksum
2	H	ROM BIOS options (bit mapped, product specific)
2	H	controller type
3	D	controller pcb version
2	H	pld code version
3	D	display driver type
2	H	display scan rate [Hz]
2	H	display brightness level [0..1F]
2	H	display mode (see below)
2	H	reset counter [0..FF]
8	D	time in HHmmSSss [hrs, mins, sec, 0.01secs]
7	D	date YYMMDDW [year mod 100, mon, dat, weekday] *
2	H	dipswitch bank 1 [0..FF]
2	H	dipswitch bank 2 [0..FF]
2	H	dipswitch bank 3 [0..FF]
2	H	base memory size (blocks of 4 kbyte)
2	H	expanded memory size (blocks of 32 kbyte)
2	H	IO status (bit mapped, see below)
2	H	photo sensor value [0..1F]
2	H	temperature sensor value [0..FF]
2	H	input port 1 (if available)
2	H	input port 2 “
2	H	output port 1 “
2	H	output port 2 “

Note : Weekday 0 = Sunday

Display mode value	***Meaning***
0	normal operation: UDP application running
1	stripes test
2	blocks test
3	freeze / brightness test
4	info screen
5..7	reserved

IO Status Bit Number	***Meaning***
0	IO-0 level (input or output)
1	IO-1 level "
2	IO-2 level "
3	1 = brightness sensor circuit detected
4	1 = temperature sensor circuit detected
5..7	reserved

9.2 INQUIRY 2 : UDP STATUS INFORMATION

syntax : Esc, $A6, $32
levels : display level only
class : communication-time command
default : none

Inquiry command 2 returns UDP software and data related parameters. This information can be used for debugging, synchronisation and software maintenance.

Chars	***Dec / Hex***	***Description***
3	D	software version.
2	H	software options (bit mapped, product specific)
2	H	UDP data errors (bit mapped, see below)
1	H	horizontal configuration, amount of boards
32	H	32 H line 1..16 configuration: amount of rows
1	H	UDP mode (0=no active text, 1=line mode, 2=page mode)
2	H	amount of messages in memory
2	H	amount of enabled messages
2	H	current message on display
2	H	amount of pages in current message
2	H	amount of enabled pages in current message
2	H	current page on display
32	H	line 1..16 status, bit mapped, see below
2	H	watchdog status, bit mapped, see below
2	H	watchdog downcounter
2	H	watchdog message
6	H	free base memory
6	H	free exp. Memory
8	D*	* project name
3	D*	* project version

Note *: Valid from software version V2.10

UDP data error,bit nr	***Meaning***
0	display configuration invalid, set to default
1	Main message table invalid, all messages cleared
2	one or more messages or pages found invalid and cleared
3	brightness table invalid, set to default
4	led test configuration invalid
5	month day configuration invalid, set to default
6	temperature limitation invalid, set to default
7	reserved

Note : these error bits remain set until the corresponding data has been updated correctly.Sending any text will clear both error bits 1 and 2.

Line statusbit nr	***Meaning***
0	0 = line 1 = line available
1	0 = line busy 1 = line ready (used in combination with one-time enable)
2..7	Reserved

Watchdog statusbit nr	***Meaning***
0,1	0,0 = watchdog disabled 0, 1 = counting seconds 1, 0 = counting minutes 1, 1 = counting hours
2	0 = display is cleared if watchdog times out 1 = message activated if watchdog times out
3	0 = watchdog timer zero 1 = watchdog timer running
4..7	reserved

9.3 INQUIRY 3 : SELF TEST RESULTS

syntax : Esc, $A6, $33
levels : display level only
class : communication-time command
default : none

Inquiry command 3 returns the results of the display self tests. Two kinds of self test can be performed, a loop test and a led test. In both cases, the data from the last display module is linked back to the controller. Both tests require some extra hardware which is not available on all products.

1) The loop test is performed only on the logic level. It continuously tests the data transport through opto fibre or cables, coders and decoders and shift registers. It does also cover the power supplies and connectors. But it does not test the led drivers or the leds. The result of the test is either Ok or False and it does not specify any particular part. A positive loop test result is a reliable indication that the most critical parts of the display are working properly, but it does not guarantee an absolutely error free led matrix.

2) The led test is a more profound test, including the leds and the led driver circuits, besides parts that are covered by the loop test. During a led test procedure, each individual led is switched on shortly while the resulting current is accurately determined. This procedure is visible as a pattern moving over the screen during some seconds. Since it is normally not desired to expose such a test pattern to public, the led test can be programmed to be performed on a certain time of day (or night) when it is less disturbing. This is done with the ledtest configure command. The result of a led test contains a list of positions of faulty leds.

Inquiry 3 returns the following led test result data :

Chars	***Dec / Hex***	***Description***
2	H	self test status (bit mapped, see below)
4	D	led test time HH:MM (see led test configure command)
4	H	amount of leds tested
4	H	amount of faulty leds detected *
160	H	first 20 faulty led positions (of 8 chars each, see below) *

Note * : the led test result values are only valid if bit 1, 2, 3 and 4 of the status bit are 0:

Self test status bit nr	***Meaning***
0	0 = loop test ok 1 = loop test fail
1	0 = led test ok, results are valid 1 = led test not performed yet, result data invalid
2	1 = led test failed, level remains low, result data invalid
3	1 = led test failed, level remains high, results data invalid
4	1 = led test busy, result data invalid

Faulty led chars	***Dec / Hex***	***Description***
2	H	horizontal module number [1..] *
2	H	line number (hardware) [1..] *
2	H	horizontal led position [1..] *
2	H	led row number [1..] *

note : counted from top and from left.

9.4 INQUIRY 4 : MONITOR PAGE

syntax : Esc, $A6, $34
levels : display level or page level
class : communication-time command
default : none

If applied on display level, this command returns the currently shown page.

If applied on page level, this command returns the contents of that page.

If applied on another level, this command will invoke a nack error.

This command is meant to be used for monitoring purposes only, and is not intended to be used as a data upload function. Since the amount of data in one page can easily exceed the maximum size of one UDP packet, it will only return the first part of each line. Each line starts with SOL and ends with the EOL code, in conformity with the UDP protocol, but the line contents is truncated at a certain point, depending on the actual configuration.

9.5 INQUIRY 5 : TEMPERATURE LIMITATION STATISTICS

syntax : Esc, $A6, $35
levels : display level
class : communication-time command
default : none

Inquiry command 5 returns information about the temperature limitation function. See also command $E9.

Chars	***Dec / Hex***	***Description***
1	H	Temperature limit status flag (set by command $E9). '0' = disabled, '1' = enabled.
2	H	Temperature limit value (set by command $E9).
1	H	Brightness limit value (set by command $E9).
2	H	Current temperature value [00..FF] (signed hex).
2	H	Highest temperature value [00..FF] (signed hex). (is reset to current temperature by sending command $E9)
4	H	Amount of minutes temperature limit exceeded and stops counting at FFFF. (Counter is reset to 0000 by sending command $E9)

9.6 CONFIGURE LED TEST

syntax : Esc, $E5 HH MM EOC
levels : display level only
class : communication-time command
default : 24:01 = led test off

The led test configure command sets the time at which the led test procedure is started. For more information about the led test command, see the description of Inquiry 3 command

Parm	***Chars***	***Dec / Hex***	***Description***	***Minimum value***	***Maximum value***
HH	2	D	Hours	0	Hours 0 23 : set led test time 24 : disable ledtest if MM>0 24 : immediate ledtest if MM=0
MM	2	D	Minutes	0	59

10 RS 485

To support 2-wire RS-485 communications, the output driver is enabled immediate after having received the EOT character in the forward packet (if addressed correctly), and disabled immediately after having transmitted the last byte of the return packet.

11 CONTROL CHARACTERS

Name	*Value[Hex]*	*Description*
SOT	01	Start of transmission
STX	02	Start of message (text)
ETX	03	End of message (text)
EOT	04	End of transmission
ENQ	05	Forward packet
ACK	06	Answer packet
SOL	0A	Start of line
EOC	0C	End of command
EOL	0D	End of line
SOP	0E	Start of page
EOP	0F	End of page
ECS	1B	Start of command
SEP	1E	Separator

12 CRC CALCULATION

A program example in C language is given here for the CCITT CRC polynomial.

```
unsigned int upcrc(unsigned int crc, char c)

{
char i ;                          // loop iterator
unsigned int a;                   // temporary var
a = (unsigned int) c ;            // convert char to unsigned integer (word)
a = a << 8 ;                      // shift char to high byte
crc = crc ^ a) ;                  // xor this with current crc value
for (i=0; i<8; ++i)               // loop 8 times :
{                                 //
if (crc & 0x8000)                 // test MSB of crc
{                                 // if set then..
crc = crc << 1 ;                  // .. shift left one bit..
crc = crc ^ 0x1021;               // .. and xor with polynomial.
}                                 // if not set then..
else crc = crc << 1;              // .. shift left one bit only.
}                                 //
crc = crc & 0xFFFF ;              // result only 16 bits
return crc ;                      //
}
```