Manual

UdpTwin

Rel. V1.2.x.x

INDEX

1 INTRODUCTION

UDPTWIN stands for ***U***niversal ***D***isplay ***P***rotocol ***T***est for ***Win***dow's. The software can be used for testing a Normal and Extended UDP display.

It is possible that the display firmware does not recognise the command send because the display firmware has no implementation of that command.

1.1 PURPOSE / SCOPE OF THIS MANUAL

The purpose of this manual is to give the information for use of the software i.e. the protocol is not explained like levels etc., this is outside the scope of this manual.

1.2 NORMAL / EXTENDED UDP

As written earlier the protocol is not described but the main difference between Normal and Extended is mentioned.

Normal UDP has a data length of maximum 240 bytes, excluding the SOT, network (NN) and EOT bytes. Multiple packets is not possible and a checksum is not calculated.

SOT NN DATA EOT

In Extended UDP multiple packets with a unique packet number (SS), could be used and a checksum is calculated over the whole packet excluding the SOT, EOT and CRC bytes. As an extra the inverted network number (ÒÒ) and the length of the data packet is also transmitted for an extra security. The length of the data packet may not exceed 240 bytes.

SOT TYPE NN ÒÒ SS LL DATA CRC EOT

All the restrictions will be handled by UdpTwin.

For more information about the protocol see the UDP Version 3.xx manual.

1.3 EXAMPLES / INQUIRY'S

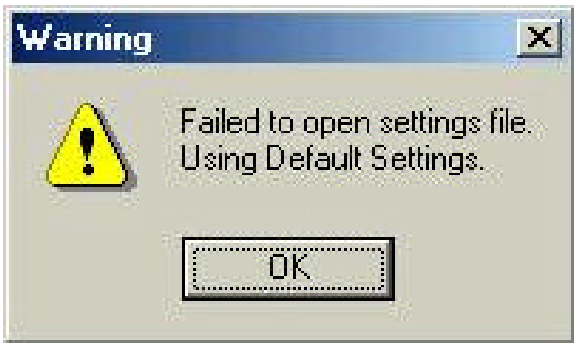
The examples 1 to 4 are available in directory \examples when menu File --> Load is selected. Inquiry's 1 to 4 are available. They represent the data which is currently available in UDP 3.xx

However it is possible that some inquiry's are differently implemented in the display firmware. The user most know what sort of data could be retrieved from the display and must create his own answer data fields. (See ***§ Answer Data box***)

2 AT START-UP

When UdpTwin is executed for the first time it attempts to load UdpTwin.set. UdpTwin.set is a file which holds the default settings.

A dialog appears with the text,



Warning

Failed to open settings file.
Using Default Settings.

Press OK.

3 LAYOUT

The software has the following layout,

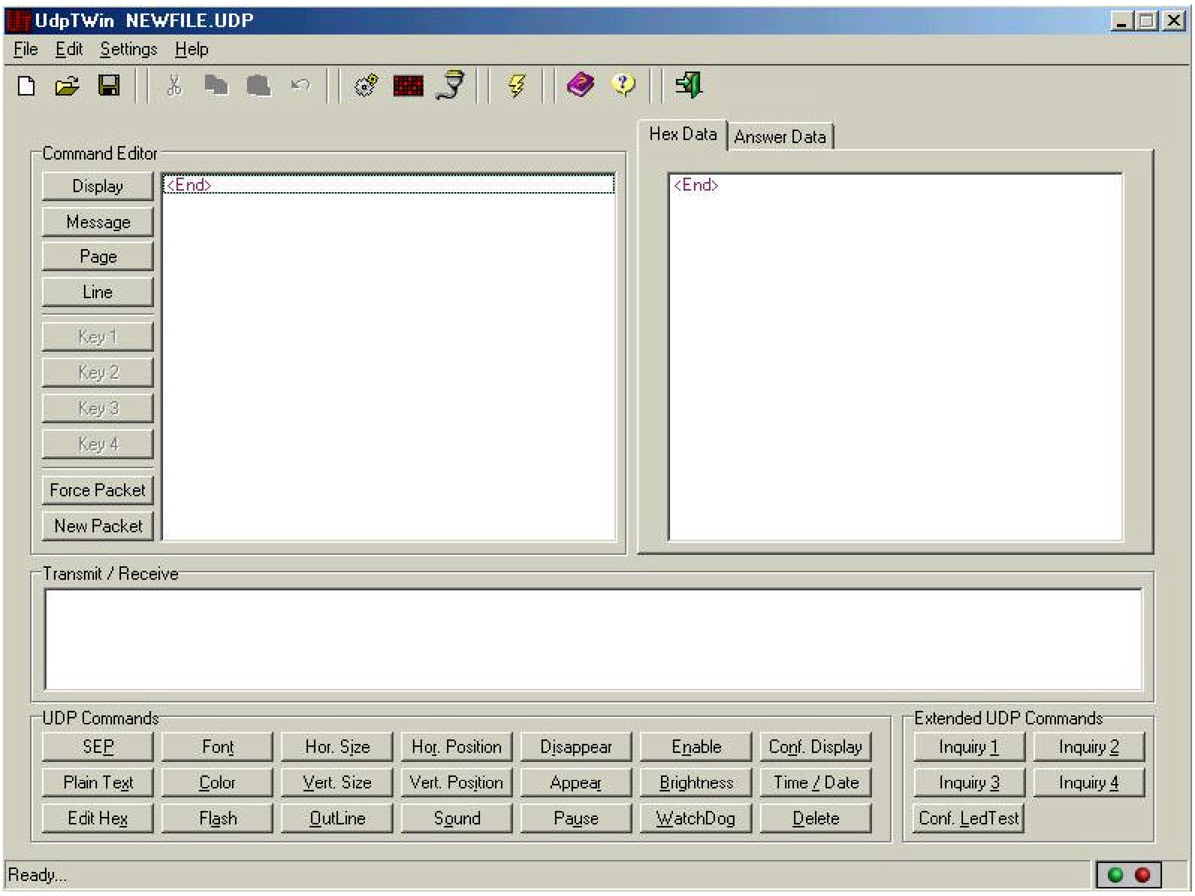


Figure 1

The layout is divided into several sections (See Figure 1),

- *Command Editor*, here are all commands etc. inserted,
- *Hex Data*, which shows the hexadecimal data of the Command Edit,
- *Answer Data*, which shows the readable data retrieved from the display,
- *Transmit / Receive*, informs the user what data is send to which display at a certain time and how long it takes to get an answer from the display,
- *The commands*, depending on the selection Normal UDP or Extended UDP the default commands are available to the user.

4 DEFAULT SETTINGS

At start-up the following settings are default and saved into a settings file, some settings could be changed by the user and are saved when exiting the program,

Serial Settings

PortName = '\\.\COM1'

Baudrate = '19200'

DataBits = 3 (Could not be changed, fixed in software)

Parity = 0 (Could not be changed, fixed in software)

StopBits = 0 (Could not be changed, fixed in software)

HwFlow = 1 (Could not be changed, fixed in software)

SwFlow = 0 (Could not be changed, fixed in software)

DTRCtrl = 0 (Could not be changed, fixed in software)

Display Settings

NetNr = 1

General Settings

ShowMainToolBar = True

CheckFileReplace = True

ShowSerialDebug = False

ExtendedUDP = True

SaveKeys = True

ClearKeys = False

DebugFile = False

CreateDebugFile = True

AppendDebugFile = False

5 TOOLBAR MENUS

Several menu items are available for the user. The menu bar has the following menu structure,

- File,
- Edit,
- Settings,
- Help.

5.1 MENU FILE

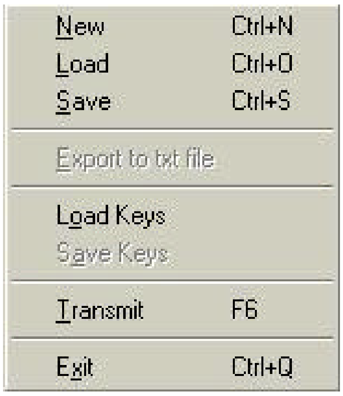


Figure 2

The following items are available (See Figure 2),

- Preparing the software for a new file (*New*), all windows are cleared even the paste.
- *Load and Save* the current Command Data in a standard UDP file format (See ***§ UDP File Format***).
- *Export to txt file*, the command box, Hex data, Answer Data and Receive / Transmit window are saved into a text file. This text file is ASCII and could be read by NotePad, a standard windows application.
- *Load and Save Keys*, the keys defined (See ***§ Defining Keys***) by the user could be loaded and saved for later use.
- *Transmit*, sends the Command Data to the display.
- *Exit*, closes the UDPTWIN.

5.2 MENU EDIT

The menu contains commands for copy, cut and paste a text item selected in the Command Box or Answer Data window. An undo is also possible but has a level of one time.

5.3 MENU SETTINGS

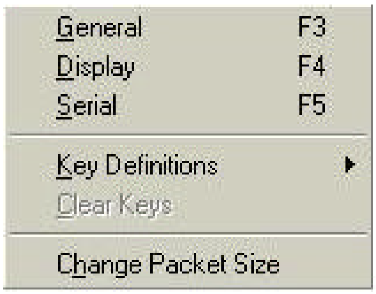
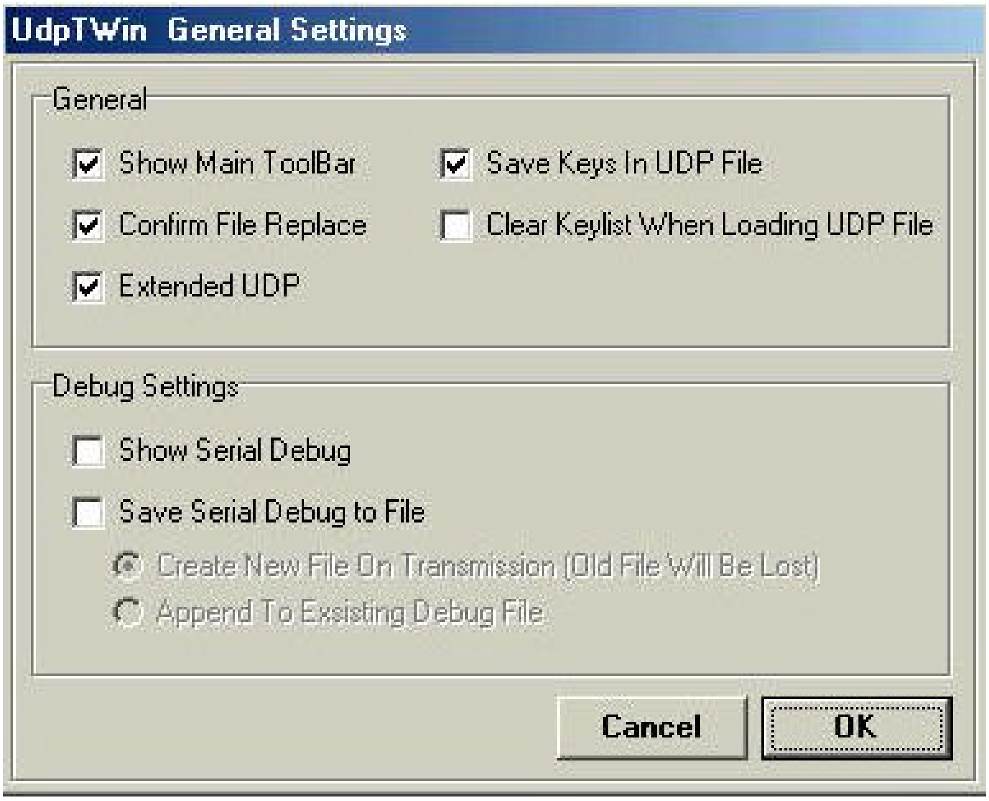


Figure 3

The following items are available (See Figure 3),

5.3.1 GENERAL

The following settings could be made,


UdpTWin General Settings

General

☑ Show Main ToolBar
☑ Save Keys In UDP File
☑ Confirm File Replace
☐ Clear Keylist When Loading UDP File
☑ Extended UDP

Debug Settings

☐ Show Serial Debug
☐ Save Serial Debug to File
● Create New File On Transmission (Old File Will Be Lost)
○ Append To Exsisting Debug File

Figure 4

- *Show Main ToolBar*, the main toolbar could be shown,
- *Confirm File Replace*, when saving a file (Udp, key or txt) a confirmation is asked when saving to an existing file,
- *Extended UDP*, select Extended UDP when the display is using the extended UDP protocol else Normal UDP is used,
- *Show Serial debug*, Select this when at a low level the transmission and receiving bytes must be inspected,

- *Save Serial Debug To File*, Select this when the transmission bytes must be saved to a debug file for further inspection. When *Create* is selected, each time a transmission is done, the debug is created and the data transmitted / received is stored in Log\debug.log. When Append is selected all transmission data is added to the debug file.
- *Save Keys*, by default the Keys which are defined by the user are saved into the UDP file ***but only when the keys are used in the Command Box, if not used save the keys using menu file --> save keys***. When loading the UDP file the key list is filled,
- *Clear Keylist*, the Key list could be cleared when loading a new UDP file so when a file is loaded the Keys could be replaced by Keys from that UDP file.

5.3.2 DISPLAY

The default network number could be changed. This network number is used when in the command editor no SOT 'NN' and EOT are defined. (See ***§ Command Editor***). The latter overrules the default network number.

5.3.3 SERIAL

The communication port and its baud rate are determined here. Default Com1 and 19200 are selected.

5.3.4 KEY DEFINITIONS

4 Keys could be defined by the user with 4 different UDP Commands which are not implemented in UdpTwin but are implemented in the UDP display firmware. How to configure and how to use the keys see ***§ Defining Keys***.

5.3.5 CLEAR KEYS

Set all keys to its default values. The keys could be cleared when loading a new UDP file. The latter is optional.

5.3.6 CHANGE PACKET SIZE

Only applicable when Extended UDP is selected. For testing it could be desirable to have a smaller packet size so the packet numbers and the reaction of the display could be tested. Another way to test the reaction of the display is the use of the commands FPACK and NPACK. See ***§ FPACK / NPACK (extended UDP only)***

5.4 MENU HELP

Shows this help and an about dialog.

6 UDP FILE FORMAT

6.1 UDP FILE FORMAT RULES

A standard UDP file could be written using Notepad or via UdpTwin. The extension of the file however must be udp. (<name.udp>)

The following rules apply for an UDP formatted file,

- Reserved words are, ESC, SEP, SOT, EOT, STX, ETX, SOP, EOP, SOL, EOL, KEY, EOC, FPACK, NPACK, HEX and ANSWER:,
- SOT, STX, SOP, SOL expect 2 data bytes, (SOL ‘01’)
- $ and ‘ are reserved characters,
- Text between ‘ and ‘ is considered as plain text or is considered data in combination with a SOT, STX, SOP, SOL or .any other command following ESC $CC [‘Data’],
- Text after a ; sign is considered as comment and is skipped by UdpTwin,
- An ESC followed with a $ sign must be followed with two bytes (CC) and is interpreted as an UDP Command,
- Depending on the command a number of data bytes is excepted. A command has the following definition, **ESC $CC [‘Data’] [EOC], see the different examples,**
    - CC in range 80 .. 9F, no data, no EOC byte,
    - CC in range A0 .. BF, 1 byte data, no EOC byte,
    - CC in range C0 .. DF, 2 byte data, no EOC byte,
    - CC in range EF.. FF, 3 or more bytes data, EOC byte

    Some Command Examples,
        - ESC $83
        - ESC $A2 'C'
        - ESC $C6 'AF'
        - ESC $FF 'BBB' EOC
- When a command is followed with data the data must be within ‘ and ‘ and must have the exact number of bytes defined by that command,
- When the reserved word ANSWER: is used, everything after that word is put into the Answer Data box, when data is received from the display (like an Inquiry) the data is converted to readable data, see UDP Example Answer,
- Several commands may be present at one line, but commands can not take more than one line,
```
–       ESC $FF ‘AAAAAAA’ EOC    ; This is a legal command definition
–          ESC $FF               ; illegal because $FF has more than 3 bytes
–            ‘AAAAAAA’           ; Interpreted as Plain text, not as data
–          EOC                   ; EOC Byte without a command
```
- Command bytes which are known are replaced with readable text.

The commands KEY, FPACK, NPACK and HEX have the following definition,

- **KEY ‘Name’ ESC $CC [‘Data’] [EOC], see the different examples**
    - Name is one of the following, Key 1, Key 2, Key 3 or Key 4, CC is a two bytes command in the range of 80 --> FF,
    - Data is optional (depending on $CC),
    - EOC is optional (depending on $CC);

    Some Key examples,
        - KEY 'Key 1' ESC $FF 'AAAAAAA' EOC
        - KEY 'Key 2' ESC $C0 '00'
        - KEY 'Key 3' ESC $A0 'F'
        - KEY 'Key 4' ESC $80

- **FPACK 'Data'**
    - Data must be 5 bytes and within range of 00000 --> 10000
- **NPACK 'Data'**
    - Data must be 5 bytes and within range of 00000 --> 10000
- **HEX ‘Data’**
    - Data must contain legal Hex values and must have the following definition,
      HEX ‘CC CC CC CC’
      Where CC lies between 00 and FF.
      Remember that 00 to 1Fh are UDP control characters and that it influences the answer of the display. Recommended is not to use them.

6.2 UDP FILE FORMAT ERRORS

When an error occurred the line number is given and the file is not loaded.

The following errors could appear when a file loaded is not complete,

Error	What to do ?
A command is defined but no ESC is found or it is text and ‘ and ‘ are missing	A command ($) is found and the whole command is not between ‘ and ‘, so the user could mean it is text but it could also be a command. In the latter case it misses a ESC code.
$ Expected	After an ESC a $ is expected.
No begin character ‘ found	Text (and data after STX, SOP etc) start with a ‘
No end character ‘ found'	Text (and data after STX, SOP etc) end with a ‘
Illegal data size	The number of parameters are invalid.
Expected 1 data byte	One byte expected.
Expected 2 data bytes	Two bytes expected.
Expected an EOC	The command which is implemented is an EOC command, but the EOC byte is not found.
An EOC command can not exist without a command	A standalone EOC is detected without a command, ESC $cc data.
Data string empty after ESC	A command is started (ESC) but no command byte etc. is found
An odd amount of Display block commands is counted	The number of SOT’s is not equal to the number of detected EOT’s
An odd amount of Message block commands is counted	The number of STX’s is not equal to the number of detected ETX’s
An odd amount of Page block commands is counted	The number of SOP’s is not equal to the number of detected EOP’s
An odd amount of Line block commands is counted	The number of SOL’s is not equal to the number of detected EOL’s
EOT Detected before SOT command	A block starts with SOT and end’s with EOT, not the other way around.
ETX Detected before STX command	A block starts with STX and end’s with ETX, not the other way around.
EOP Detected before SOP command	A block starts with SOP and end’s with EOP, not the other way around.
EOL Detected before SOL command	A block starts with SOL and end’s with EOL, not the other way around.
Insertion of a Higher Block in a Lower Block order and insertion of a Block in a Block with the same order is not allowed	Insertion of a higher block, for example a Page block, in a lower block (Line block) or at the same level (Page block) is not possible.
Data size error (this command has a fixed data size)	The command used has a fixed number of bytes.
Data size Brightness table error (this command has a fixed data size) UDP V2.xx has an entry of 32 bytes UDP V3.xx has an entry of 16 bytes	The number of bytes used is invalid. This must be 16 or 32 bytes.
Data expected after $	$ is reserved word. After an $ is the command number expected.

Illegal command	Command lies between 80h and FFh. A lower command is used (00h..7Fh).
Expected 5 data bytes	FPACK and NPACK expect 5 bytes data.
String is empty or value is illegal	Suppose a command like FPACK is used. Then the line FPACK ‘’ could generate this error. Therefore the data is empty and that is not legal.
EOC Command must have at least 3 bytes data	An EOC command is detected but has less than 3 bytes. An EOC command has 3 or more bytes data.
KEY Name exceeds maximum length (8)	The name of the key to define must have a name which does not exceeds the 8 characters.
No Key Name found	The string after KEY is probably empty.
KEY Could not be assigned	– No key name is given, – ESC code is missing, – No Hex data is available. This error should never happen. When this error occurs all checks failed which is nearly impossible. Send us the UDP file and we have a look.
Invalid HEX data	When a non Hex data is loaded. Hex is from 0 to F and fore instance AH is non hex.
No HEX Command data available	No data seen after the command HEX.

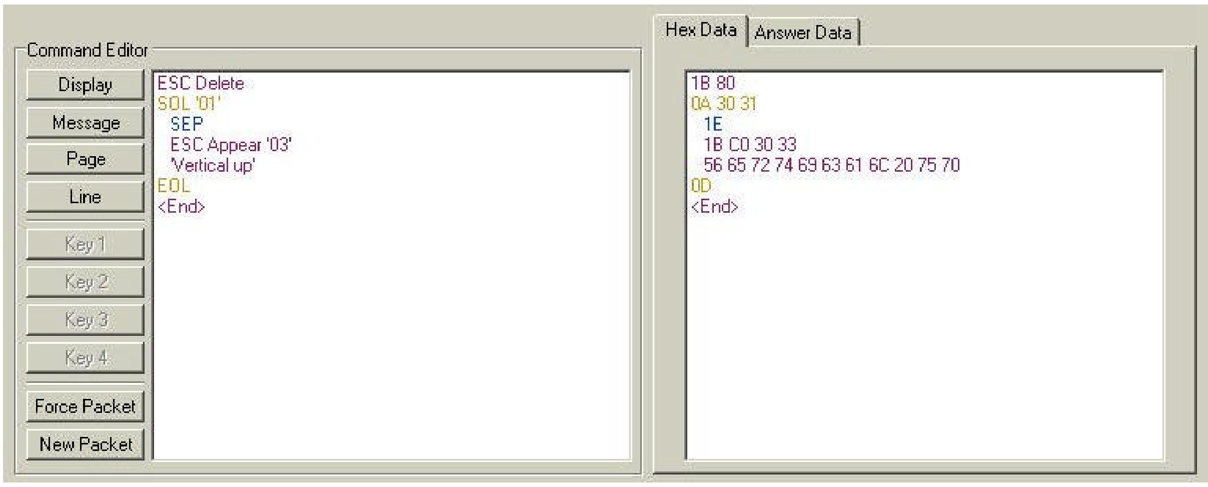
6.3 UDP FILE EXAMPLE 1

The following text file (written in NotePad) is a standard UDP file,

```
; Example 1

ESC $80                    ; Delete Command, Clear entire display
SOL '01'                   ; Line 01
  SEP
  ESC $C0 '03'             ; Appear
  'Vertical up'            ; With plain text 'Vertical up'
EOL                        ; End of line
```

When transmitted (F6) the display is cleared, and Vertical Up will appear on Line 1.

Figure 5 shows the screen after the file is converted and loaded,



```
ESC Delete
SOL '01'
  SEP
  ESC Appear '03'
  'Vertical up'
EOL
<End>
```

```
1B 80
0A 30 31
  1E
  1B C0 30 33
  56 65 72 74 69 63 61 6C 20 75 70
0D
<End>
```

Figure 5

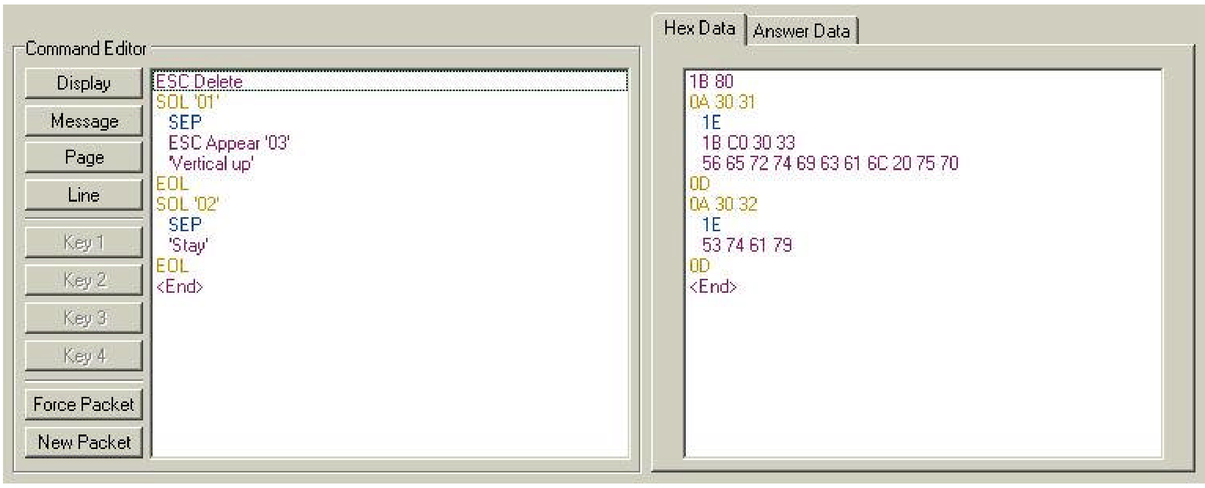
6.4 UDP FILE EXAMPLE 2

When a 2 Line (or more) display is available the next example could be executed,

```
; Example 2
; 2 Line Display


ESC $80                 ; Delete Command
SOL '01'                ; Line 01
  SEP
  ESC $C0 '03'          ; Appear
  'Vertical up'         ; With plain text 'Vertical up'
EOL                     ; End of line
SOL '02'                ; Line 02
  SEP
  'Stay'                ; With plain text 'Stay'
EOL                     ; End of line
```

When transmitted (F6) the display is cleared, and Vertical Up will appear on Line 1 and the text Stay, appears at Line 2.

Figure 6 shows the screen after the file is converted and loaded,



```
ESC Delete
SOL '01'
  SEP
  ESC Appear '03'
  'Vertical up'
EOL
SOL '02'
  SEP
  'Stay'
EOL
<End>
```

```
1B 80
0A 30 31
  1E
  1B C0 30 33
  56 65 72 74 69 63 61 6C 20 75 70
0D
0A 30 32
  1E
  53 74 61 79
0D
<End>
```

Figure 6

7 COMMAND EDITOR

In the Command Editor it is possible to build an UDP packet starting at the highest level down to the lowest level. There are 5 levels,

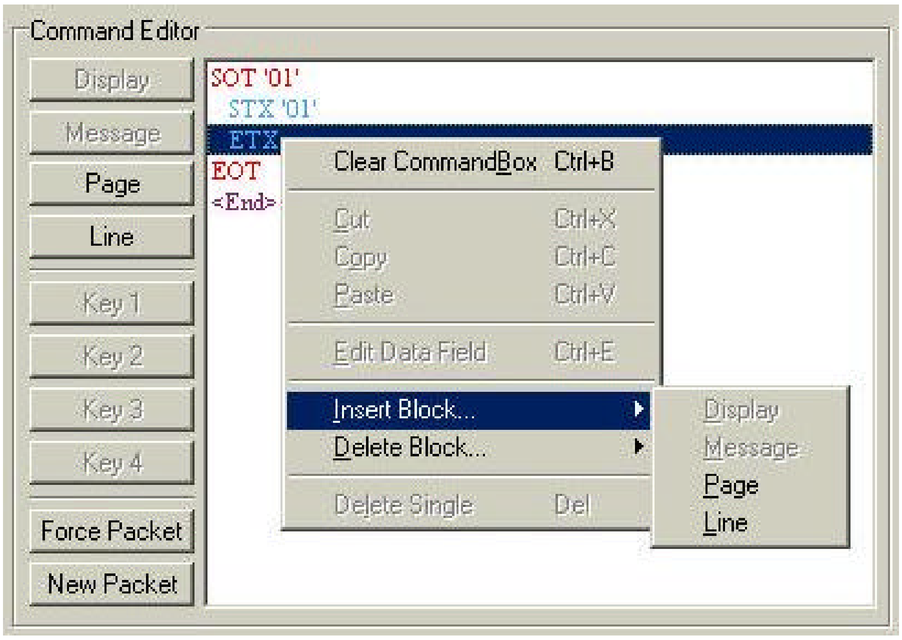
- Display (SOT, EOT),
- Message (STX, ETX),
- Page (SOP, EOP),
- Line (SOL, EOL),
- Character level (See ***§ Command Levels***).

A level is defined between two level commands i.e. a start and an end. For example, level Line is defined between SOL and EOL. (Start Of Line, End Of Line) The latter two are also called BLOCK commands.

For example, when pressing the display button a dialog appears. The user must now set the Display number and that is inserted.

A lower level can only be inserted in a higher level, for example, a Line can be inserted in a Page but a Page not in a Line. Block commands of the same level can also not be inserted. For example, a Line can not be inserted in a Line.

The buttons at the left side of the window corresponds with the level (from high to low) and when a level is valid the button will light up and therefore it is enabled.



```
SOT '01'
  STX '01'
  ETX
EOT
<End>
```

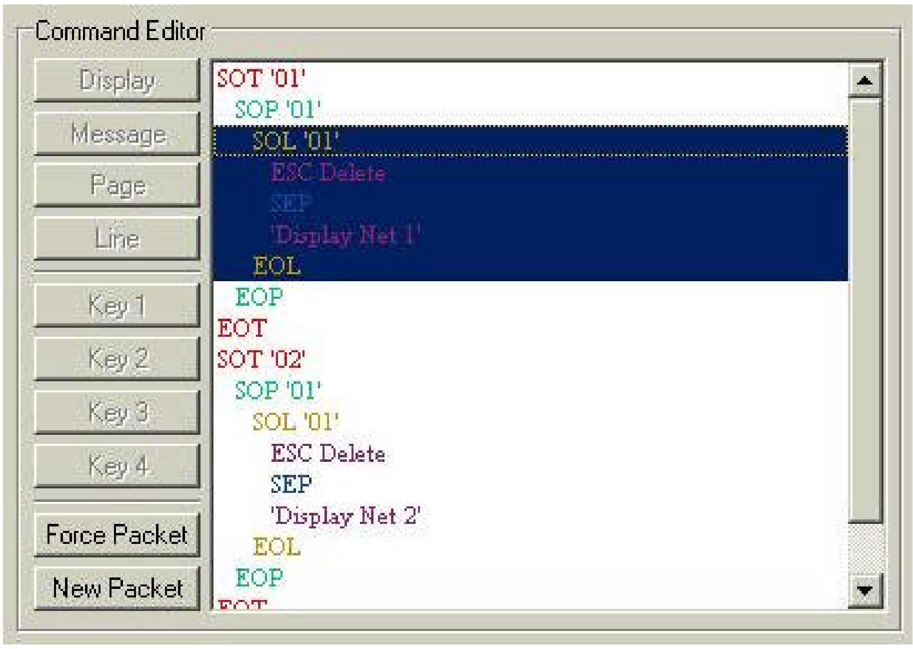
Figure 7

7.1 SINGLE MOUSE CLICK

Select a line by moving the mouse cursor into the Command Box and position it at the Line that has to be selected, and press the left mouse button. As shown in the above figure, the ETX line is selected and because this level is a lower level than Display but a Higher level than Page and Line, the Display and the Message buttons are disabled.

When clicking the right mouse button a menu appears with several commands. (See Figure 7).

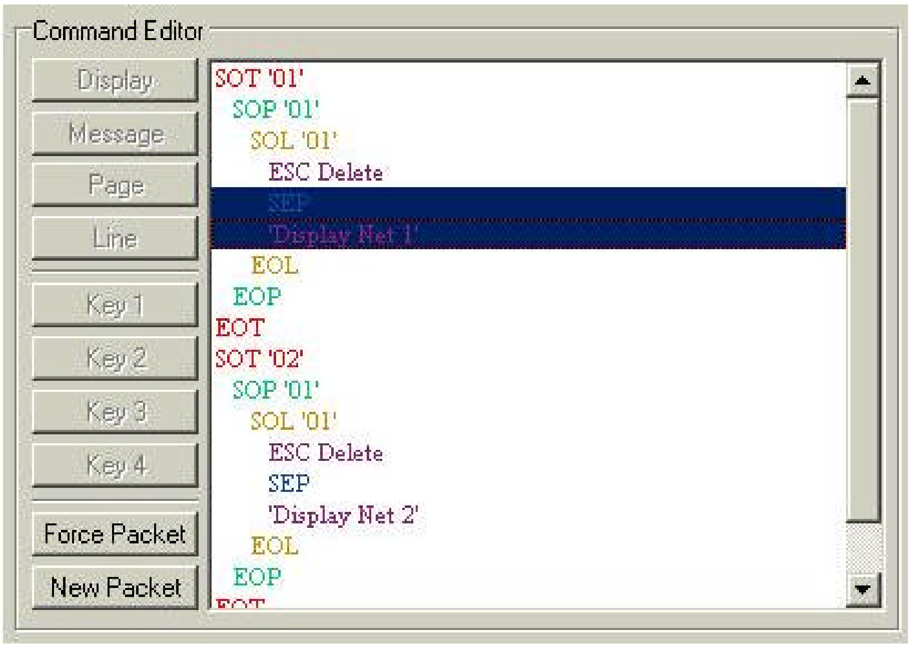
7.2 DOUBLE MOUSE CLICK



```
SOT '01'
  SOP '01'
    SOL '01'
      ESC Delete
      SEP
      'Display Net 1'
    EOL
  EOP
EOT
SOT '02'
  SOP '01'
    SOL '01'
      ESC Delete
      SEP
      'Display Net 2'
    EOL
  EOP
```

Figure 8

When double clicking on a Block command (Start or End) and at the same time holding the Shift key. The whole block will be selected.



```
SOT '01'
  SOP '01'
    SOL '01'
      ESC Delete
      SEP
      'Display Net 1'
    EOL
  EOP
EOT
SOT '02'
  SOP '01'
    SOL '01'
      ESC Delete
      SEP
      'Display Net 2'
    EOL
  EOP
```

Figure 9

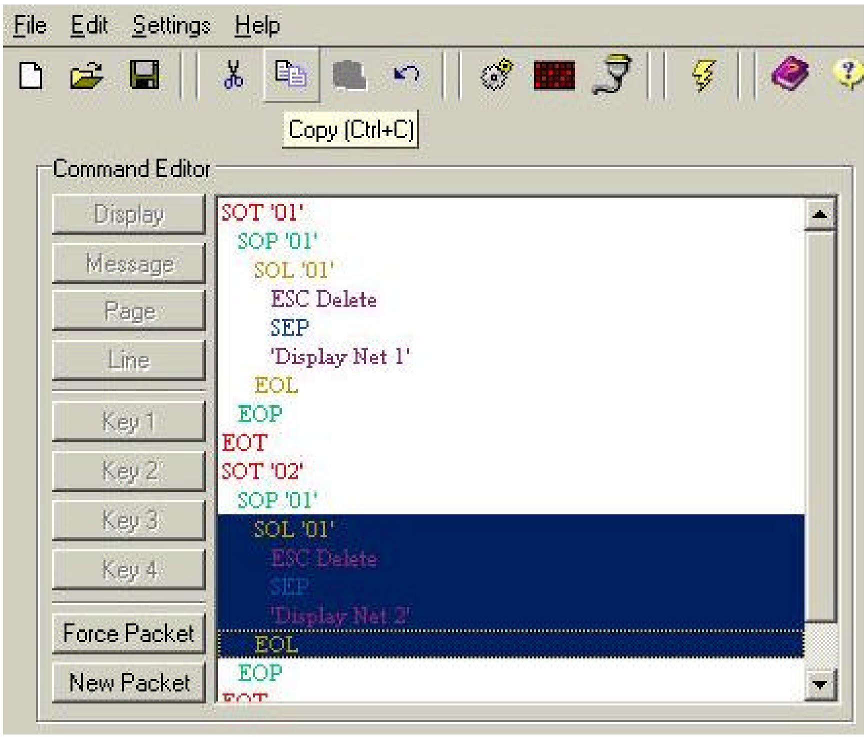
When double clicking on a command, holding the Shift key, all the commands will be selected until a block command is reached i.e. a start or end block.

When data is editable (like a SOL or Font command etc.) the data is reached by double clicking that line. A line editor will appear.

7.3 DRAG AND DROP / COPY AND PASTE

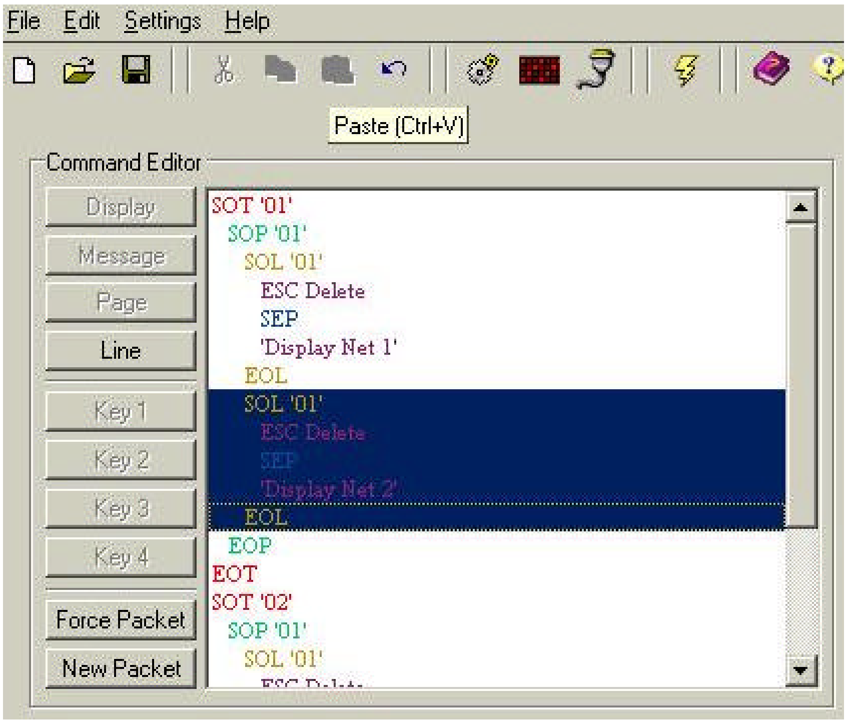
The selected block or commands can be picked up and dropped elsewhere. Dropping the selected commands / block is allowed when the condition is met, i.e. a Line block could be dropped in a Page block but a Page block could not be dropped in a Line block. In stead of moving the block elsewhere it is possible to make a copy and paste it elsewhere.

Example,



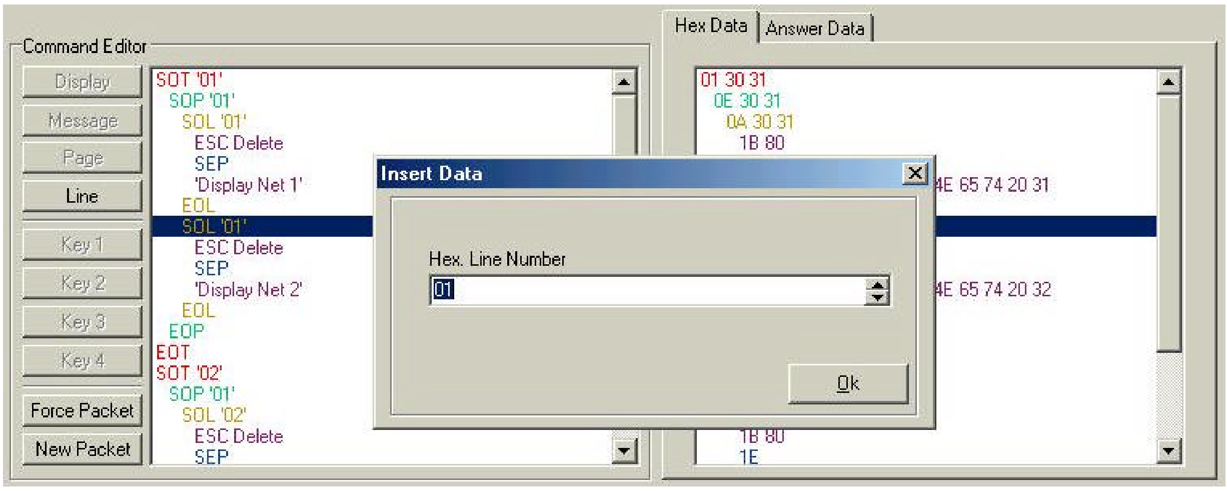
```
SOT '01'
  SOP '01'
    SOL '01'
      ESC Delete
      SEP
      'Display Net 1'
    EOL
  EOP
EOT
SOT '02'
  SOP '01'
    SOL '01'
      ESC Delete
      SEP
      'Display Net 2'
    EOL
  EOP
EOT
```

A paste could only be made when the block is inserted in a higher block. The paste button is disabled when the cursor is within the Line Block commands. Suppose we want to paste in a line 2 in page 1 then the result will be,



```
SOT '01'
  SOP '01'
    SOL '01'
      ESC Delete
      SEP
      'Display Net 1'
    EOL
    SOL '01'
      ESC Delete
      SEP
      'Display Net 2'
    EOL
  EOP
EOT
SOT '02'
  SOP '01'
    SOL '01'
```

When the above UDP commands are transmitted, the last SOL ‘01’ (See selected block) will be accepted. Change the SOL ‘01’ into SOL ‘02’ by selecting the command line and click the right mouse button or press Ctrl+E. A dialog appears and the line number can be changed.



```
SOT '01'
  SOP '01'
    SOL '01'
      ESC Delete
      SEP
      'Display Net 1'
    EOL
    SOL '01'
      ESC Delete
      SEP
      'Display Net 2'
    EOL
  EOP
EOT
SOT '02'
  SOP '01'
    SOL '02'
      ESC Delete
      SEP
```

```
01 30 31
  0E 30 31
    0A 30 31
      1B 80
4E 65 74 20 31
4E 65 74 20 32
      1B 80
      1E
```

Hex. Line Number: 01

8 FPACK / NPACK (EXTENDED UDP ONLY)

Extended UDP has the ability for transmitting packets of a maximum packet length = 240 bytes. All packets have the same length and a unique packet number. The packets are created automatically when the data exceeds the packet size. The packet numbers are automatically assigned to a packet and start from 0.

What is / are FPACK and NPACK ?

- FPACK <Time> ***(Force Packet)***
  When an FPACK is seen a packet is created and the packet number is increased.
- NPACK <Time> ***(New Packet)***
  When an NPACK is seen the packet number is reset to 0.

Time is a number between 0 and 10000 in mS. Default 5000. A count down in mS is done and when 0 is reached the next packet is transmitted.

With Change PacketSize, FPACK and NPACK the user has the ability to test the transmission with packets.

Multiple FPACK’s and NPACK’s are filtered therefore,

```
NPACK ‘5000’
NPACK ‘5000’
SOL ‘01’
  SEP
  ‘Text Line 1’
EOL
NPACK ‘5000’
NPACK ‘5000’
```

will result in,

```
NPACK ‘5000’
SOL ‘01’
  SEP
  ‘Text Line 1’
EOL
NPACK ‘5000’
```

on transmission. (This is not to be shown in the Command Editor)

The first and last NPACK will also be filtered because until the first NPACK is reached there is no data to transmit and after the data SEP and ‘Text Line 1’ is transmitted no other data is to be send, therefore a wait is also not done.

The next example makes more sense,

```
SOL ‘01’
  SEP
  ‘Text Line 1’
EOL
NPACK ‘5000’
SOL ‘02’
  SEP
  ‘Text Line 2’
EOL
```

After the first packet is transmitted (started at SOL ‘01’) a wait is done and the second packet (started at SOL ‘02’) is send. The same rules of multiple packets apply for FPACK.

9 INSERT A COMMAND

Commands which are available for are the following commands,

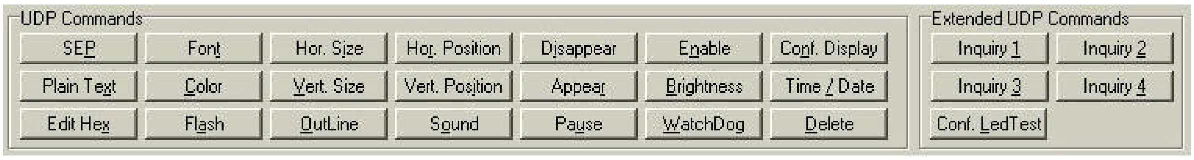


Figure 10

Some commands have no parameter, others ask for one, two or a series of data. Depending on the command a dialog window will appear.

9.1 BRIGHTNESS TABLE

From the first release of UDP until UDP V2.xx the Brightness table had 16 entries, each entry 2 bytes. For the complete table 32 bytes where reserved. From UDP V3.xx the brightness table has 16 entries each entry one byte. For the complete table 16 bytes is reserved. When inserting a brightness table the user most know witch UDP display version is valid. When inserting the command this is asked. When asked enter Yes for UDP 2.xx and No for UDP 3.xx

9.2 COMMAND LEVELS

5 Command levels are available in UDP,

- Display,
- Message,
- Page,
- Line,
- Character.

Determine the level by inserting a block, i.e. Display, Message etc. How to insert a command at character level ??

9.2.1 EXAMPLE 3, FLASH

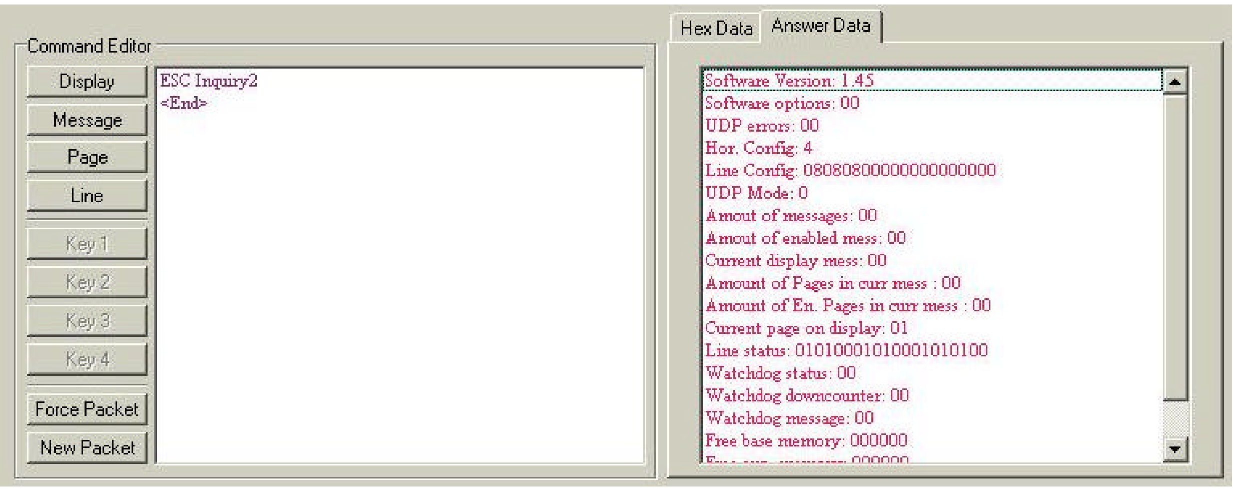
For instance, a flash command is a command which could be used at a character level. Suppose the second word ‘Text’ must flash, the next example shows the way to do it,

```
; Example 3

; Character level Flash

SOL '01'
  ESC $80              ; Delete the line
  SEP
  'This '
  ESC $A4 '1'          ; Enable Flash
  'Text '              ; Flashing text
  ESC $A4 '0'          ; Disable Flash
  'Flashes'
EOL
```

10 ANSWER DATA BOX

When loading Inquiry 2 a transmission is being done and a readable answer is created. The Inq2 command must be implemented in the display firmware else an ack answer is received. One @ stands for one byte. According to the UDP manual V3.xx is 3 bytes needed for a version number, therefore 3 @ must be insert. In other words, the number of data bytes is the number of @.

Figure 11 shows the screen after a transmission. The @ are filled and the data is readable,



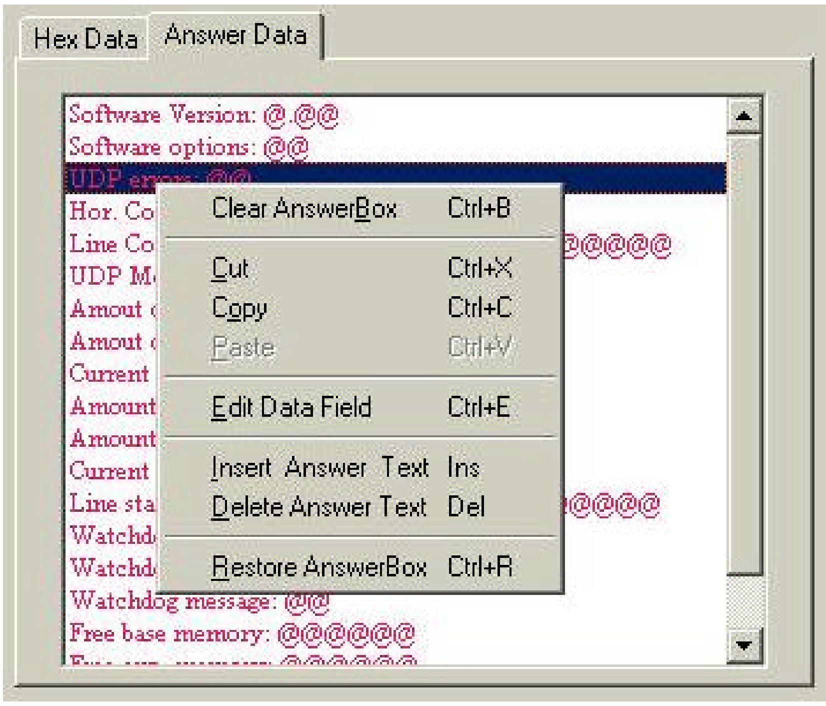
ESC Inquiry2
<End>

Software Version: 1.45
Software options: 00
UDP errors: 00
Hor. Config: 4
Line Config: 08080800000000000000
UDP Mode: 0
Amout of messages: 00
Amout of enabled mess: 00
Current display mess: 00
Amount of Pages in curr mess : 00
Amount of En. Pages in curr mess : 00
Current page on display: 01
Line status: 01010001010001010100
Watchdog status: 00
Watchdog downcounter: 00
Watchdog message: 00
Free base memory: 000000

Figure 11

The Answer Data could be saved to a file using the ***Export to File*** command in menu File.

When the mouse cursor is moved to the Answer Data box, and the right mouse button is clicked a menu appears and several menu items are valid. See Figure 12.



Software Version: @.@@
Software options: @@
Watchdog message: @@
Free base memory: @@@@@@

Figure 12

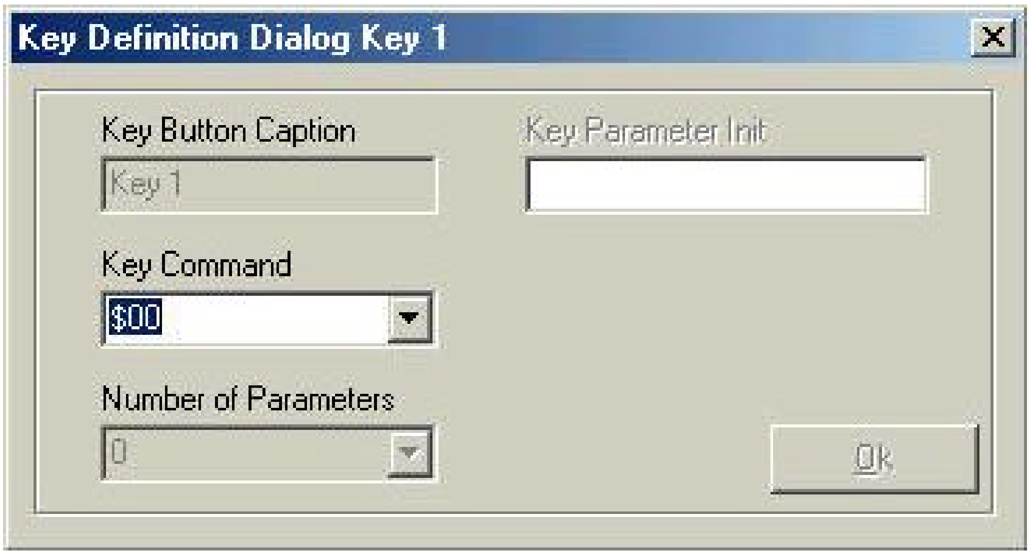
Text can be inserted, using INS, changed ,using Edit Data Field, deleted, using DEL etc. After the @ are filled, readable text appears. When another transmission is done the Answer Data is refreshed. When the data is being edited the box is restored and only the @ can be edited. The user can also restore the Answer Data by using Restore AnswerBox.

11 DEFINING KEYS

Commands which do not exist in UdpTwin and are implemented in the Udp Display firmware, for any reason, could be created via menu Settings --> Key Definitions -> Key1 .. Key4.

4 Keys can be assigned and are a useful tool for testing a new command. Keys which are defined can be saved and loaded via menu File --> Load / Save Keys.

The Keys which are defined and ***used in the command Editor*** could be saved into the UDP file using the option described in menu Settings --> General.



Key Button Caption: Key 1

Key Parameter Init:

Key Command: $00

Number of Parameters: 0

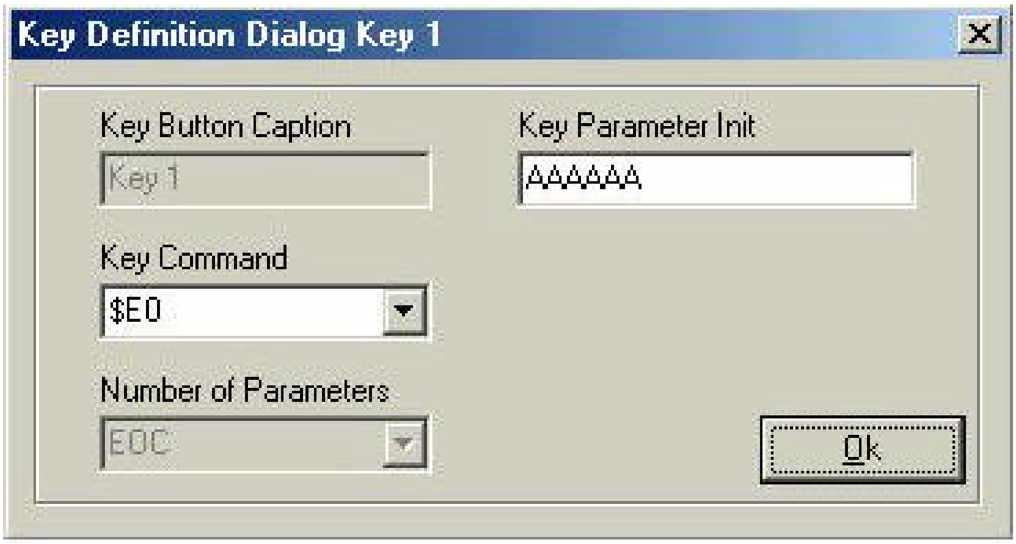
Figure 13

Two conditions must be met for enabling a key,

- A key Command is selected and is not equal to 00, which is the default,
- Depending the selected command Key Parameters could be assigned, default values are 0.

When the two conditions are met, the Ok button is enabled and can be pressed.

11.1 EXAMPLE KEY ASSIGNMENT



Key Button Caption: Key 1

Key Parameter Init: AAAAAA

Key Command: $E0

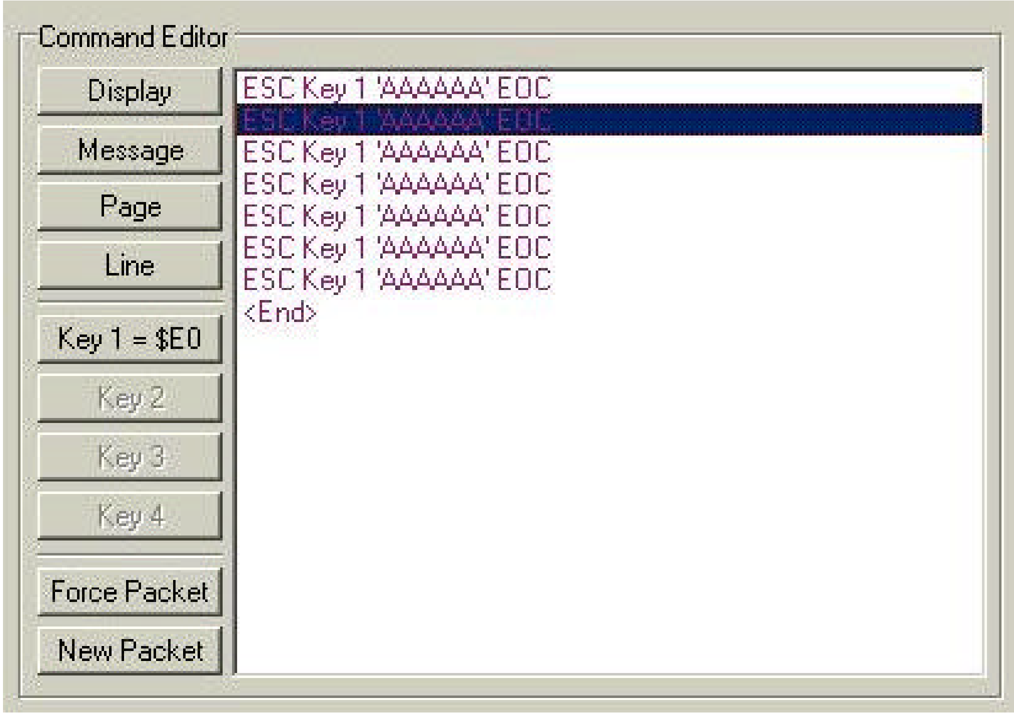
Number of Parameters: EOC

Figure 14

See Figure 14.

At this point Key1 is assigned to Command $E0, which is an EOC command, with button caption ‘Key 1’.

After the Key is confirmed Key1 is assigned and ready for use, see Figure 15.



```
ESC Key 1 'AAAAAA' EOC
ESC Key 1 'AAAAAA' EOC
ESC Key 1 'AAAAAA' EOC
ESC Key 1 'AAAAAA' EOC
ESC Key 1 'AAAAAA' EOC
ESC Key 1 'AAAAAA' EOC
ESC Key 1 'AAAAAA' EOC
<End>
```

Figure 15

When a key command is being edited there is no range check done on the data, i.e. a byte could have any value (-+/^&, 0..9, A..Z, a..z etc). Therefore one must know if the bytes transmitted to a display are valid. When a Key is assigned to an EOC command, its length of data is not fixed. Therefore when a key is initialised with 3 bytes it could have been changed into another number of bytes when using the *Edit Data Field* (Select line with mouse and press Ctrl+E).

11.2 SAVE / LOAD KEYS

There are two ways to save keys,

- The Key list (Key1 .. Key4) could be saved into a special *.kcf file (Key Configure File),
- Keys which are configured and used in the command Editor could be saved into the UDP file format when the option is checked in menu Settings --> General. The advantage is that the created UDP file has no unknown commands.
  When the UDP file is loaded it is possible to clear the configured keys and assign the keys if key definitions are available in the UDP file format.

11.3 IMPORTED NOTE

When keys are configured and not used in the command Editor, the keys are not saved into the UDP file even if in Settings --> General --> Save Keys Into UDP File is checked. When the keys are cleared and not saved into *.kcf file, the Key definitions are lost.

Keys from the UDP file (if available) are only assigned when there is a free key. When no keys are available the key commands are loaded but not assigned.

12 TRANSMITTING TO MULTIPLE NETWORKS

Insert a display level. Asked is to insert a 2 byte value what represents the network number. When a text must be placed on network 1 and there must be a text placed on network 2 do the following, see example 6,

12.1 EXAMPLE 4

```
; Transmit to NetWork 1 and NetWork 2

SOT '01'            ; NetWork 1
  ESC $80           ; Clear Display
  SOL '01'
    SEP
    'Network 1'
  EOL
EOT
SOT '02'            ; NetWork 2
  ESC $80           ; Clear Display
  SOL '01'
    SEP
    'Network 2'
  EOL
EOT
```