MiniTouch / MiniKey

REMOTE COMMUNICATION Technical reference

Ethernet / RS232

Version:	November 2011
MT Firmware	1.65
By:	Torben Dam Jensen

Revision history

December 2011	Initial version	TDJ
14 Mar 2012	Corrected O:, where : was wrong. Should be Oobject...	TDJ

CONTENT

Introduction

The MiniTouch / MiniKey products are powerful stand-alone controllers with ability to perform almost any print task using the print jobs as templates and external communication through RS232 or Ethernet.

Design layouts once, and fill with content as required. Select jobs. Change parameters. Ask status – and many other functions are possible. Using intelligence of external software allows you to extend the possibilities of the controllers to do almost anything you want.

And with the MiniTouch you have even more flexibility because of its ability to interact with external devices. With this controller you get 7 output signals that can be controlled by events OR manually by remote control.

You do not get a pre-defined software package with Mini series. Instead, you get the building blocks which allow you to custom build a tailored solution for the individual application.

Conventions used in this guide

```
CMD:R#          Data string SENT TO controller
RES:0#          Data string RECEIVED FROM controller

<ESC>           ESCAPE CHARACTER, ascii 27.
<EOT>           End of Text character, ascii 04

{variable}      Placeholder, like for a command or error code


<a|b|c>         Alternatives, meaning a OR b OR c
```

What you should understand before continuing

In order to understand the protocol for communicating with the Mini series, you should understand:

- Basic principles of request-response communication
- Basic principles of inkjet concepts, such as offsets, buffers etc
- Fundamentals of RS232 and the ASCII character set, such as hexadecimal values and escape codes
- Ethernet communication principles such as IP address, subnet mask and sockets.

It does not matter what programming language you use to connect to the controller. Perl, Ruby, VB.NET, C++, C# or AutoIT Script are all among possible alternatives.

Please bear in mind that HSA can not support you in the above fundamentals. You should consult an external programmer if you do not fully understand this documentation.
You are recommended to read the source code of the free examples we provide and build your application from there.

Basics of communicating with the Mini Series

Before going into details about the commands, you should understand some basics about communication with the Mini Series.

Read below carefully to avoid any issues with your communication.

- For Ethernet, all commands and responses are terminated by the "#" character.
- A group of "special characters" ('#' ';' ':' '\') must be escaped by \ (backslash) to be used in content.
- ALL other character values are **returned and sent** in their raw ascii value (32 .. 255). Language sets other than English must be encoded according to ANSI / Windows 12xx. Make sure that characters above ascii 127 are not interpreted as control characters in your application. See section on encoding.
- For Ethernet: When content is returned (by a REQuest command) the content is NOT escaped, so you have to be careful if the content contains one of the special characters. Example: content ##Hello## is returned as DAT:S1=static;tex=##Hello###
  The programmer must evaluate only LAST returned separator in the string, or communication will break.
- For RS232 communication, use a NULL-modem cable on pins 2,3 and 5 with these parameters:
  You can change these parameters if your applications requires it.

Bits per Second:	115.200
Data Bits:	8
Parity:	NONE
Stop Bits	2
Flow Control:	NONE

- For RS232 you MUST enclose command strings in ESC (ascii 27) and EOT (ascii 04). These special characters are written <ESC> and <EOT> in this guide.
- For Ethernet communication, use device IP address, port 3000. You can set IP / Netmask / Gateway manually or use DHCP server on same network. DHCP is the simplest and recommended.
- The Mini Series will send a response to EVERY command you send with a result code. It is not NECESSARY to evaluate response, but strongly recommended.
- The Mini Series separates OBJECTS and CONTENT. Objects have visual properties like position, where content have options for the text you can read.
- The commands to use are abbreviated to single letters when using RS232. This is to speed up communication
- The commands AND all object names are **CASE SENSITIVE**. RO is not identical to Ro, like Object1 is not identical to object1

Examples

Below follows a few examples of the communication. If you do not understand at first, do not worry, the commands will be explained below.

The examples both demonstrate connection, changing some content and disconnecting.

RS232 communication

<ESC>CC;admin;admin<EOT>	Login
<ESC>C<ACK><EOT>	..accepted
<ESC>CF;FILE1<EOT>	Load "FILE1"
<ESC>C<ACK><EOT>	..accepted
<ESC>Obatch:T=12345<EOT>	Chg. batch to 12345
<ESC>O<ACK><EOT>	..acepted
<ESC>CD<EOT>	Logout

Ethernet communication

CMD:C;admin;admin#	Login
RES:0;Transmission OK#	..accepted
CMD:F;FILE1#	Load "File1"
RES:0;Transmission OK#	..accepted
OBJ:batch;TEX=12345#	Chg. batch to 12345
RES:0;Transmission OK#	..accepted
CMD:D#	Logout

Quick Guide

Use the quick guide as a 1-page reference for the most common tasks. This is your checklist for most communication.

RS232 quick guide

The RS232 command string has the following form:

<ESC>{command}{function}:{parameter 1};{parameter 2}<EOT>
<ESC> is ASCII char 027, hex 0x1B
<EOT> is ASCII char 004, hex 0x04

{Command}{function} and {parameters} are separated by a colon. Individual parameters separated by a semicolon. Not all commands have parameters.

For all commands, the controller will return a response in the form

<ESC>{command}<ACK><EOT>	**SUCCESS, command accepted**
<ESC><NAK>{error code}<EOT>	**FAIL, not executed**

<ACK> is ascii 006, hex 0x06
<NAK> is ascii 021, hex 0x15
{error code} is a distinct code to tell you the fail you made.

The most common commands are:

<ESC>CC<EOT>	Connect (logins disabled)
<ESC>CC;a1;xxx<EOT>	Login user a1 with pass xxx
<ESC>CF:FILENAME<EOT>	Load the layout “filename”
<ESC>RO<EOT>	List objects
<ESC>RC<EOT>	List content
<ESC>Rc;tex1<EOT>	Get information about content "tex1"
<ESC>Oobj1:T=xxx<EOT>	Change content in obj1 to xxx
<ESC>CR<EOT>	Enter print mode
<ESC>CS<EOT>	Leave print mode
<ESC>CD<EOT>	Disconnect

Ethernet quick guide

Ethernet command string has the following form:

```
{Command}:{Function};{Parameters}#
```

For each command, the Mini series will send a response:

```
RES:0;Transmission OK#                    SUCCESS
RES:{error code};{error description}#     FAIL
```

The most common commands are:

`CMD:C#`	Connect (logins disabled)
`CMD:C;a1;xxx#`	Login user a1 with pass xxx
`CMD:F;filename#`	Load the layout “filename”
`REQ:objects#`	List objects
`REQ:contents#`	List content
`OBJ:obj1;TEX=xxx#`	Change content in obj1 to xxx
`CMD:R#`	Enter print mode
`CMD:S#`	Leave print mode
`CMD:D#`	Disconnect

COMMAND REFERENCE

This chapter contains a complete reference for the commands available on the Mini Series, including examples.

RS232 command reference

Basic commands – prefix: C

Name	Function	Parameters	Reply	Description	Example	Notes
Connect	C	<user>;<password> (optional) <user> (optional) <password> (optional)	C<ACK> IU (input user) (optional) IP (input password) (optional)	Connects to MT from remote location. Login user+password is used if login is enabled on MT.	Ex1: <ESC>CC<EOT> <ESC>IU<EOT> <ESC><user><EOT> <ESC>IP<EOT> <ESC><password><EOT> <ESC>C<ACK><EOT> Ex2: <ESC>CC;<user>;<password><EOT> <ESC>C<ACK><EOT>	If a user is already connected on the terminal, remote user must have admin rights to login.
Disconnect	D		C<ACK>	Disconnect remote connection to MT	<ESC>CD<EOT> <ESC>C<ACK><EOT>	
Start print	R	(none) – just start print mode <-> (optional - wait for go) <sw> (optional - print on next 'sw' start switches)	C<ACK>	Start print mode on MT. Parameter '-' (minus sign) starts print mode but wait for go command before printing on start switch. Parameter <sw> is print go command and MT will print on next <sw> start switches.	<ESC>CR<EOT> <ESC>C<ACK><EOT> <ESC>CR;-<EOT> <ESC>C<ACK><EOT> <ESC>CR;10<EOT> <ESC>C<ACK><EOT>	
Stop print	S		C<ACK>	Stop print mode on MT	<ESC>CS<EOT> <ESC>C<ACK><EOT>	

Update next print buffer	B		C<ACK>	**User managed buffer**: Put 1 print on print queue. **Normal/no buffer**: Updates print buffer. Must be used when objects are changed during print to avoid object buffer delay.		
Load Job	F	<job_name> <path\\job_name> (optional path)	C<ACK>	Loads <Job_name> from <path> in MT memory. The <path\\> is optional if job is saved in root.	**<ESC>CF;JOBS\\EX\\MY_JOB<EOT>** **<ESC>C<ACK><EOT>**	Folder names and filenames are max 8 chars with large capital letters (DOS file names). Use '\\' to seperate folders in <path>.
User message	U	<line1>;<line2>;<line3> (line2 + line3 are optional)	C<ACK>	Displayes user text lines on MT display.	**<ESC>CU;Controlled;Important!;** **Please don't turn off<EOT>** **<ESC>C<ACK><EOT>**	The 3 user lines are displayed on MT "Remotely Controlled" menu.
Reboot	Q	R (reboot optional) (shutdown if R is omitted)	C<ACK>	Reboots or shuts down MT		
Set output	O	<output nr>;<delay(mm)>;<pulse(mm)> <output nr>;<1|0>	C<ACK>	Set an output (0-6), either with an delayed pulse (outputs 0-3 only) or with constant value (outputs 2-6 only)	**<ESC>CO;1;200;2000<EOT>** (set IO output 2 for 2000mm after 200mm) **<ESC>CO;2;1<EOT>** (set control output 1)	Output nr 0 = IO output 1 Output nr 1 = IO output 2 Output nr 2 = Control output 1 Output nr 3 = Control output 2 etc…

Object handling – prefix O

Object handling requires **Object access = On** by remotely connected user.

Multiple parameters for 1 object can be send in 1 command.

Text Objects

Name	Function	Parameters	Reply	Description	Example	Notes
Text	<text_object>	T=<new text>	O<ACK>	Sets text of text object (text object must be linked to a **static** content, and **ONLY contain ONE** content).	**<ESC>OMY_TEXT;T=this is a test<EOT>** **<ESC>O<ACK><EOT>**	Max text size is 127 chars. Only for text objects linked to static content.
Font by number	<text_object>	f=<font number>	O<ACK>	Sets text print font to font with list number <font number>.	**<ESC>OT1;f=4<EOT>**	
Font by name	<text_object>	F=<font name>	O<ACK>	Sets text print font to font with <font name>.	**<ESC>OT1;F=Arial10<EOT>**	
Position	<text_object>	P=<x>,<y>	O<ACK>	Sets text object position in print.	**<ESC>OT1;P=600,10<EOT>**	(0,0) is upper left corner. Measurement in pixel 600 pixel = 1 inch = 2.54cm

Graphic objects

Name	Function	Parameters	Reply	Description	Example	Notes
Logo	<graphic_object>	l=<logo name>	O<ACK>	Sets active logo of a Graphics logo object by logo name (7-char logo name given at upload time)	<ESC>OL1;l=mylogo<EOT>	
Logo by number	<graphic_object>	N=<logo number>	O<ACK>	Sets active logo of a Graphics logo object by logo number.		
Position	<graphic_object>	P=<x>,<y>	O<ACK>	Sets graphic object position in print.		
Line width	<graphic_object>	W=<line width in pixels>	O<ACK>	Sets graphic box or line object line width.		
Length	<graphic_object>	L=<object length in pixels>	O<ACK>	Sets graphic box width or line length.		
Height	<graphic_object>	H=<object height in pixels>	O<ACK>	Sets graphic box height.		
Fill	<graphic_object>	B=<1|0> (1=ON)	O<ACK>	Sets graphic box black fill ON/OFF.		

Barcode objects

Name	Function	Parameters	Reply	Description	Example	Notes
Text	<barcode_object>	C=<new text>	O<ACK>	Sets text of barcode object (barcode object must be linked to a **static content** and **ONLY ONE content**).	<ESC>Ob1;C=1234<EOT>	Max text size is 127 chars. Only for barcode objects linked to static content.
Human readable	<barcode_object>	H=<1|0> (1=ON)	O<ACK>	Enables/disables human readable text on barcode.		
Size	<barcode_object>	S=<line width>,<line height> (in pixels)	O<ACK>	Set barcode line size.		
Space	<barcode_object>	W=<line extra spacing> (in pixels)	O<ACK>	Set barcode extra white spacing.		

Black	<barcode_object>	B=<line extra black> (in pixels)	O<ACK>	Set barcode extra black line width.		
Checksum	<barcode_object>	O=<0|1> (1=ON)	O<ACK>	Set barcode checksum on/off.		
Type	<barcode_object>	T=<barcode type>	O<ACK>	Set barcode type.		
Font by number	<barcode_object>	f=<font number>	O<ACK>	Sets human readable font to font with list number <font number>.		
Font by name	<barcode_object>	F=<font name>	O<ACK>	Sets human readable font to font with <font name>.		
Position	<barcode_object>	P=<x>,<y>	O<ACK>	Sets barcode object position in print.		

Content Handling – Prefix O

Static content

Name	Function	Parameters	Reply	Description	Example	Notes
Text	<Static content>	T=<new text>	O<ACK>	Sets text of a static content object.	<ESC>OMY_STATIC;T=this is a test<EOT> <ESC>O<ACK><EOT>	Max text size is 127 chars.

Counter content

Name	Function	Parameters	Reply	Description	Example	Notes
Minimum value	<counter>	m=<value>	O<ACK>	Set minimum count value of a counter content.		Min value must be less than counter max value.
Maximum value	<counter>	M=<value>	O<ACK>	Set maximum count value of a counter content.		Max value must be larger than counter min value.
Current value	<counter>	C=<value>	O<ACK>	Set current count value of a counter content.		New counter value must be within min/max limits.
Digits	<counter>	D=<value>	O<ACK>	Set amount of digits in a counter content.		Max 10 digits.
Repeats	<counter>	R=<value>	O<ACK>	Set amount of prints per counter increment.		
Step	<counter>	S=<value>	O<ACK>	Set counter increment value.		
Leadin	<counter>	L=< |'0'|> (' '=spaces leadin, '0'=zeros leadin , blank/others=no leadin)	O<ACK>	Set leadin of printed counter value.	<ESC>OCOUNTER;L=x<EOT> (x = other so no leadin) <ESC>O<ACK><EOT>	

Date/Time content

Name	Function	Parameters	Reply	Description	Example	Notes
Expirery day offset	<datatime_content>	E=<day offset>	O<ACK>	Set day offset of a date content.		In the layout design you can also offset by months, years, hours, minutes and seconds.
Format	<datatime_content>	F=<format string> (see format types)	O<ACK>	Sets date/time content print format.		
Date set	<datatime_content>	D=<year/month/day>	O<ACK>	Sets date of a data/time content object.	<ESC>ODate1;D=2012/12/30<EOT> <ESC>O<ACK><EOT>	
Time set	<datatime_content>	T=<hour:minute:second>	O<ACK>	Sets time of a data/time content object.	<ESC>ODate1;T=13:37:14<EOT> <ESC>O<ACK><EOT>	

PARAMETER HANDLING – Prefix P

Parameter handling requires **Parameter access = On** by remotely connected user.

Multiple parameters can be send in 1 command.

Parameter commands are appended with <L|M|none> to set (L)ayout, (M)achine or none=Machine paremeters.
Example: Sets resolution, encoder parm, print mode and buffer mode Machine parameters in 1 command.

```
<ESC>PM:r=0;e=0,04;E=P;B=-<EOT>
<ESC>P<ACK><EOT>
```

Print parameters

Name	Function	Parameters	Reply	Description	Example	Notes
Picture length	<L|M|none>	L=<value> (in pixels)	P<ACK>	Set picture length i pixels.	<ESC>PM:L=2400<EOT>	600 pixel = 1 inch = 2.54 cm
Resolution	<L|M|none>	r=<600|300|1row300A| 1row300B>	P<ACK>	Set print resolution. (Sets both vertical and horizontal resolution to selected). 1row300A prints with row A only, 1row300B is row B only.	<ESC>PM:r=600<EOT>	
Vertical Resolution	<L|M|none>	vres=<600|300|1row300A|1r ow300B>	P<ACK>	Set vertical resolution. 600 and 300 implies 2row printing. 1row300A prints with row A only, 1row300B is row B only.	<ESC>P:vres=1row300A<EOT>	
Horizontal Resolution	<L|M|none>	hres=<75|90|110|150|200|24 0 |300|320|400|440| 480|500|600|800| 960|1200|1600|2400>	P<ACK>	Set horizontal resolution.	<ESC>P:hres=320<EOT>	
Print direction	<L|M|none>	D=<r|l|b>	P<ACK>	Set print direction (r)ight|(l)eft|(b)idir.	<ESC>PL:D=b<EOT>	
Bidir start direction	<L|M|none>	b=<r|l|1|2|3|4>	P<ACK>	Set start print direction on bidir print mode. (right)|(l)eft|input(1) select|input(2) select |input(3) select|input(4) select	<ESC>P:b=4<EOT>	On input selection, start is either set print direction or opposite set print direction

Start distance	<L|M|none>	S=<value> (in pixels)	P<ACK>	Set start distance from start signal to print i pixels.	**<ESC>PL:S=140<EOT>**	
Start dist. (bidir)	<L|M|none>	>=<value> (right direction in pixels) <=<value> (left direction in pixels)	P<ACK>	Set bidir start distances from start signal to print. (>)=right print direction (<)=left print direction	**<ESC>PM:>=1200;<=2400<EOT>**	
Start sensor edge	<L|M|none>	G=<+|->	P<ACK>	Set startsensor active edge (+=positive, -=negative).	**<ESC>P:G=+<EOT>**	
Modular parameter	<L|M|none>	M=<value>	P<ACK>	Set modular parameter.		
Velocity parameter	<L|M|none>	V=<velocity> (in meters/min)	P<ACK>	Set velocity parameter. (decimal notation is allowed)		
Encoder parameter	<L|M|none>	e=<encoder_par> (in mm/pulse)	P<ACK>	Set encoder parameter. (decimal notation is allowed)		
Quadrature	<L|M|none>	Q=<+|->	P<ACK>	Set encoder quadrature on|off.		
Repeat	<L|M|none>	R=<repeats>,<repeat_distance> (in pixels)	P<ACK>	Set print repeats and distance between repeats.		
Endless mode	<L|M|none>	U=<+|->	P<ACK>	Set endless print on/off.		
Print mode	<L|M|none>	E=<P|M|V>	P<ACK>	Set print mode (P)osition, (M)odular or (V)elocity.		
Buffer mode	<L|M|none>	B=<+|-|u>	P<ACK>	Set print buffer mode (+)normal buffer (-)no buffer (u)user managed.		
Head offset	none	O=<head nr>;<head offset>;<pen1 offset>;…	P<ACK>	Set pen offsets for a complete head. Offset are written with/without decimal point, and value may be appended with unit type (mm, in or px for mm, inches or pixels notation).	**<ESC>P:O=1;0;0;25mm;50.1mm;75.2mm<EOT>** **<ESC>P<ACK><EOT>**	Sets offsets for head 1, in this case a 4-pen stall with head offset 0 and pen offsets 0, 25mm, 50.1mm and 75.2mm.
Head parameters	None	H=<head nr.>; <parameter>=<value> parameters: offset: O=<value><unit> upside/down: U=<+|-> otherside: S=<+|-> Mirror: M=<+|-> voltage: V=<voltage>;<fire width(us)> size:s=<head_size>		Set head parameter: Offset, Upsidedown, otherside, voltage or head size.	**<ESC>P:H=1;U=+<EOT>** **<ESC>P<ACK><EOT>**	Enable upsidedown for head 1.

Pen parameters	None	P=<head nr>;<pen nr>;<parm>=<value> parameters: turn off: - turn on: + offset: O=<value><unit> overlap: L=<value> ink setup: I=<ink supply (ml)>;<ink low (ml)> ink reset: R+ pen switch: S=<+|->		Set pen parameters: turn off, turn on, offset, overlap, ink setup, ink reset and pen auto switch.	**<ESC>P:P=1;1;I=42;4<EOT>** **<ESC>P<ACK><EOT>**	Set ink supply size (42ml) and ink warning level (4ml) of pen 1 in head 1.
Set date	none	d=<year>/<month>/<day> (format yyyy/m/d)	P<ACK>	Set system date.	**<ESC>P:d=2012/12/30<EOT>**	
Set time	None	t=<hour>:<minute>:<second> (format h:m:s)	P<ACK>	Set system time.	**<ESC>P:t=17:32:59<EOT>**	
Set start sensor block distances	None	z=<start_block_dist(units)>; <end_block_dist(units)>	P<ACK>	Set start sensor start block distance and end block distance including units (mm|in|px).	**<ESC>P:z=1mm;30mm<EOT>**	

Parameter Source selection

Name	Function	Parameters	Reply	Description	Example	Notes
Select parmaeters source	SEL	<resolution_source> <sensor_source> <repeat_source> <head voltage source> (L|M)	P<ACK>	Select resolution, sensor, repeat and head voltage parameters source to (M)achine or (L)ayout	**<ESC>PSEL:MMMM<EOT>**	Set all parameters to Machine source.

Network parameter handling

Name	Function	Parameters	Reply	Description	Example	Notes
Network setup	N	D=<1|0> (DHCP on/off) N=<network name> I=<IP address> (IPV4 dot separated) G=<Gateway address> M=<network mask> P=<ip port>,<file port>	P<ACK>	Set network parameters. Include parameters which needs to be changed in command.	<ESC>PN:D=0;I=192.168.160.199;N=HSAMT<EOT> <ESC>P<ACK><EOT>	Set DHCP off + IP address + network name.

REQUEST HANDLING – Prefix R

OBJECT INFORMATION

Name	Function	Parameters	Reply	Description	Example	Notes
Grahic Object info (line)	o	<grahic_line_object>	Ro:<object_name>; <graphic_type>; <X>,<Y>; <line_length>;<line_width>	Request information about a line.	<ESC>Ro:gra1<EOT>	See enumeration for graphic types. Measurements in pixel
Graphic Object info (box)	o	<grahic_box_object>	Ro:<object_name>; <graphic_type>; <X>,<Y>; <box_width>;<box_height>; <line_width>; <box_fill+|->	Request information about a box		
Graphic Object info (logo)	o	<grahic_logo_object>	Ro:<object_name>; <graphic_type>; <X>,<Y>; <logo_width>;<logo_height>; <logo_name>	Request information about a logo		
Text Object info	o	<text_object>	Ro:<object_name>; <X>,<Y>;<rotation_type>; <font_name>; <linked_content_1>;…;<linked_content_n>; <content_seperator_char>	Request information about a text object		Linked content is the **name** of the content linked into the object
Barcode Object info	o	<barcode_object>	Ro:<object_name>; <X>,<Y>;<rotation_type>; <font_name>; <linked_content_1>;…; <linked_content_n>; <content_seperator_char>; <barcode_type>; <checksum_0|1>;	Request information about a barcode object		Linked content is the **name** of the content linked into the object See enumeration for barcodes

			<line_width>; <black_size>;<white_size>; <human_readable>; <lines_height>			

CONTENT INFORMATION

Name	Function	Parameters	Reply	Description	Example	Notes
Static Content info	c	<static_content>	Rc:<content_name>;<static_text>	Info on a static content Notice! Special characters (;\#:) are NOT escaped on request content	<ESC>Rc:Sta1<EOT> <ESC>Rc:Sta1;Hello world<EOT>	
Counter Content info	c	<counter_content>	Rc:<content_name>; <current_value>; <digits>; <min_value>;<max_value>; <repeats>; <step_value>;<leadin_type>	Info on a counter content		Leadin type = < |0|>
DateTime Content info	c	<data_time_content>	Rc:<content_name>; <format>; <offset>; <locales_name>	Info on a datetime content		Offset format: <year>-<month>-<day> <hour>:<min>:<sec>
Shift Content info	c	<shift_content>	Rc:<content_name>; <shift_datetime_1>;<shift_data_1>; …; <shift_datetime_n>;<shift_data_n>	Info on a shiftcode content		Shift data format: <year>-<month>-<day> <hour>:<min>:<sec>
System Content info	c	<system_content>	Rc:<content_name>; <system_type>	Info on a system content		
Identifier content	c	<identifier_content>	Rc:<content_name>; <ID_format>	Info on an identifier content		

OTHER INFORMATION

Name	Function	Parameters	Reply	Description	Example	Notes
Parameters list	p	<L|M|none>	RP:<picture_length>; <start_distance>; <resolution_type>; quadrature+|->; <print_dir_r|l>; <repeats>; <modular_parm>; <velocity_parm>; <encoder_parm>; <sensor_edge_+|->; <endless_+|->; <printmode_p|m|v>; <buffer_mode_+|-|u>; <head_offset_1>; <head_size_1>; <head_1_pen_offset_1>;...; <head_1_pen_offset_n>;....; <head_n....>	Get a list of parameters for either (L)ayout or (M)achine. If nothing specified, machine parameters are returned.		
File list	D	<directory_path> (optional)	RD:<file/dir_name_1>;....;<file/dir_name_n>		<ESC>RD<EOT> <ESC>RD:<SUB>;<SUB2>;file1<EOT>	directory names are indicated by '<'dir_name'>'
Active jobname	F		RF:<job_name>	Name of active job name. Does not load a file.		
Active user	U		RU:<user_name>	Name of active user.		
Object list	O		RO:<object_name_1>=<tex|bar|grp>;...; <object_name_n>=<tex|bar|grp>	List of OBJECTS		
Content list	C		RC:<static_name_1>=<sta>;..;<static_name_n>=<sta>; <counter_name_1>=<cnt>;...;<counter_name_n>=<cnt>; <date_time_name_1>=<dat>;...;<date_time_name_n>=<dat>; <shift_name_1>=<shf>;...;<shift_name_n>=<shf>; <system_name_1>=<sys>;...;<system_name_n>=<sys	List of CONTENT		

			>; <identifier_name_1>=<id>;…;<identifier_name_n>=<id>			
Version info	V		RV:<MT_type>;<firmware_version>;<build_date>; <FPGA_version>	Version information		
Ethernet info	I		RI:<network_name>;<ip_address>;<ip_netmask>; <ip_gateway>;<ip_com_port>;<ip_file_port>; <DHCP_on(+)|off(-)|failed(f)>	Ethernet settings		
Uptime	T		RT:<uptime_hours>	Uptime hours, starting from 0 at last power on.		
Print info	i		Ri:<printing_1|0>;<pictures_printed>	How many prints has been done, and print mode		
Ink info	B		RB:<ink_used_pen_1>;…;<ink_used_pen_n>	How much ink has been used (calculated), in ml.		Remaining ink = full amount minus used amount
Pen status	S		RS:<pen_1_status>;<pen_2_status>;…	See pen status enumeration		
Print Done Interrupt	d	+/-	Rd:<print_done_interrupt_on|off_1|0>		(when print done and print interrupt on:) **<ESC>SP:<prints_since_last_interrupt><EOT>**	If print is very fast, there may be more than one print before an interrupt, so <ESC>SP:2<EOT> may be returned. (or, 3,4…)

ETHERNET COMMAND REFERENCE

Command syntax: <Command>:<Function>;<Parameters>#

Reply on succes: RES:0;Transmission OK# **IN THE DOCUMENTATION EXAMPLES BELOW THIS IS SHORTED RES:0#**

Reply on fail: RES:<error_code>;<error_description>

COMMANDS – Prefix CMD

Name	Function	Parameters	Reply	Description	Example	Notes
Connect	C	<user>;<password> (optional) <user> (optional) <password> (optional)	RES:0 DAT:Please login INP:username (optional) INP:password (optional)	Connects to MT from remote location. Login user+password is used if login is enabled on MT.	Ex1: CMD:C# DAT:Please login# INP:username# <user># INP:password# <password># RES:0# Ex2: CMD:C;<user>;<password># RES:0#	If a user is already logged in on the terminal, remote user must have admin rights
Disconnect	D		RES:0	Disconnect remote connection to MT	CMD:D# RES:0#	
Start print	R	None – just start print mode. <-> (optional - wait for go) <sw> (optional - print on next 'sw' start switches)	RES:0	Start print mode on MT. Parameter '-' (minus sign) starts print mode but wait for go command before printing on start switch. Parameter <sw> is print go command and MT will print on next <sw> start switches.	CMD:R;-# RES:0;# (start print mode and wait for go command) CMD:R;1# RES:0# (print on next start switch only)	CMD:R# with no parameters starts print mode and MT will print on every start switch if valid.
Stop print	S		RES:0	Stop print mode on MT		

Update next print buffer	B		RES:0	User managed buffer: Put 1 print on print queue. Normal/no buffer: Updates print buffer. Must be used when objects are changed during print to avoid object buffer delay.		
Load Job	F	<job_name> <path\\job_name> (optional path)	RES:0	Loads <Job_name> from <path> in MT memory. The <path\\> is optional if job is saved in root.	**CMD:F;MY_JOBS\\EX\\MY_JOB#** **RES:0;…#**	Folder names and filenames are max 8 chars with large capital letters (DOS file names). Use '\\' to seperate folders in <path>.
User message	U	<line1>;<line2>;<line3> (line2 + line3 are optional)	RES:0	Displayes user text lines on MT display.	**CMD:U;User control;Do not;turn off#** **RES:0;…#**	The 3 user lines are displayed on MT "Remotely Controlled" menu.
Reboot	Q	R (reboot optional) (shutdown if R is omitted)	RES:0	Reboots or shuts down MT	**CMD:Q#** **RES:0;#** **CMD:Q;R#** **RES:0;#**	
Set output	O	<output nr>;<delay(mm)>;<pulse(mm)> <output nr>;<1|0>	RES:0	Set an output (0-6), either with an delayed pulse (outputs 0-3 only) or with constant value (outputs 2-6 only)	**CMD:O;1;200;2000#** **RES:0;…#** **(set IO output 2 for 2000mm after 200mm)** **CMD:O;2;1#** **RES:0;…#** (set control output 1)	Output nr 0 = IO output 1 Output nr 1 = IO output 2 Output nr 2 = Control output 1 Output nr 3 = Control output 2 etc…

OBJECT HANDLING – Prefix OBJ

Object handling requires **Object access = On** by remotely connected user.

Multiple parameters for 1 object can be send in 1 command.

TEXT OBJECTS

Name	Function	Parameters	Reply	Description	Example	Notes
Text	<text_object>	TEX=<new text>	RES:0	Sets text of text object (text object must be linked to a static content).	OBJ:MY_TEXT;TEX=this is a test# RES:0;#	Max text size is 127 chars. Only for text objects linked to static content.
Font by number	<text_object>	FNT=<font number>	RES:0	Sets text print font to font with list number <font number>.		
Font by name	<text_object>	FON=<font name>	RES:0	Sets text print font to font with <font name>.	OBJ:Text1;FON=Arial10# RES:0;…#	
Position	<text_object>	POS=<x>,<y>	RES:0	Sets text object position in print.		Position is in pixel, upper left corner is (0,0)

GRAPHIC OBJECTS

Name	Function	Parameters	Reply	Description	Example	Notes
Logo	<graphic_object>	PAT=<logo name>	RES:0	Sets active logo of a Graphics logo object by logo name.	OBJ:log1;PAT=complog# RES:0;…#	Request object / content list for valid names.

Logo by number	<graphic_object>	NUM=<logo number>	RES:0	Sets active logo of a Graphics logo object by logo number.		
Position	<graphic_object>	POS=<x>,<y>	RES:0	Sets graphic object position in print.		
Line width	<graphic_object>	WID=<line width in pixels>	RES:0	Sets graphic box or line object line width.		
Length	<graphic_object>	LEN=<object length in pixels>	RES:0	Sets graphic box width or line length.		
Height	<graphic_object>	HIG=<object height in pixels>	RES:0	Sets graphic box height.		
Fill	<graphic_object>	FLL=<On|Off>	RES:0	Sets graphic box black fill ON/OFF.		

BARCODE OBJECTS

Name	Function	Parameters	Reply	Description	Example	Notes
Text	<barcode_object>	CON=<new text>	RES:0	Sets text of barcode object (barcode object must be linked to a static content).	OBJ:BC1;CON=123456789012# RES:0;…#	Max text size is 127 chars. Only for barcode objects linked to static content. Content must be valid for symbology.
Human readable	<barcode_object>	HUM=<1|0> (1=ON)	RES:0	Enables/disables human readable text on barcode.		
Size	<barcode_object>	SIZ=<line width>;<line height> (in pixels)	RES:0	Set barcode line size.		
Space	<barcode_object>	SPC=<line extra spacing> (in pixels)	RES:0	Set barcode extra white spacing.		
Black	<barcode_object>	BLK=<line extra black> (in pixels)	RES:0	Set barcode extra black line width.		
Checksum	<barcode_object>	CHK=<0|1> (1=ON)	RES:0	Set barcode checksum on/off.		
Type	<barcode_object>	TYP=<barcode type>	RES:0	Set barcode type.		See barcode enumerations
Font by number	<barcode_object>	FNT=<font number>	RES:0	Sets human readable font to font with list number <font number>.		
Font by name	<barcode_object>	FON=<font name>	RES:0	Sets human readable font to font with <font name>.		

Position	<barcode_object>	POS=<x>,<y>	RES:0	Sets barcode object position in print.		

STATIC CONTENT

Name	Function	Parameters	Reply	Description	Example	Notes
Text	<static_content>	TEX=<new text>	RES:0	Sets text of a static content object.	OBJ:MY_STATIC;TEX=this is a test# RES:0;…#	Max text size is 127 chars. Special characters can be escaped using \ character. Literal \ is input as \\

COUNTER CONTENTS

Name	Function	Parameters	Reply	Description	Example	Notes
Minimum value	<counter_content>	MIN=<value>	RES:0	Set minimum count value of a counter content.	OBJ:C1;MIN=125# RES:0;…#	Min value must be less than counter max value.
Maximum value	<counter_content>	MAX=<value>	RES:0	Set maximum count value of a counter content.		Max value must be larger than counter min value.
Current value	<counter_content>	CUR=<value>	RES:0	Set current count value of a counter content.		New counter value must be within min/max limits.
Digits	<counter_content>	DIG=<value>	RES:0	Set amount of digits in a counter content.		Max 10 digits.
Repeats	<counter_content>	REP=<value>	RES:0	Set amount of prints per counter increment. (repeat value)		
Step	<counter_content>	STP=<value>	RES:0	Set counter increment value. (step value)		
Leadin	<counter_content>	LDN=< |'0'|> (' '=spaces leadin, '0'=zeros leadin , blank/others=no leadin)	RES:0	Set leadin of printed counter value.	OBJ:C1;LDN= # (SPACE) RES:0;…#	

DATE/TIME CONTENT

Name	Function	Parameters	Reply	Description	Example	Notes
Expirery day offset	<datatime_content>	EXP=<day offset>	RES:0	Set day offset of a date content.	OBJ:D1;EXP=10# RES:0;…#	Set relative date, based on no.of days. In design mode offset can also be set to years, months, hours and minutes
Format	<datatime_content>	FOR=<format string> (see format types)	RES:0	Sets date/time content print format.	OBJ:D1;FOR=yyyy-mm-dd# RES:0;…#	
Date set	<datatime_content>	DAT=<year/month/day>	RES:0	Sets date of a data/time content object.		Set absolute value, opposite to expiery day offset.
Time set	<datatime_content>	TIM=<hour:minute:second>	RES:0	Sets time of a data/time content object.		Set absolute time

PARAMETER HANDLING – Prefix PAR

Parameter handling requires Parameter acces On by remotely connected user.
Multiple parameters can be send in 1 command.
Paremeter commands are appended with <L|M|none> to set (L)ayout, (M)achine or none=Machine paremeters.
Examlpe: Sets resolution, encoder parm, print mode and buffer mode Machine parameters in 1 command.

```
PAR:M;RES=600;ENC=0,04;ENM=pos;BUF=-#
RES:0;…#"
```

NORMAL PARAMETERS

Name	Function	Parameters	Reply	Description	Example	Notes
Picture length	<L|M|none>	LEN=<value> (in pixels) size=<value> (in pixels)	RES:0	Set picture length i pixels.	PAR:L;LEN=600# RES:0;…#	600 pixel = 1 inch = 2.54 cm
Resolution	<L|M|none>	RES=<600|300|1row300A|1row300B> resolution=<600|300|1row300A|1row300B>	RES:0	Set print resolution. (Sets both vertical and horizontal resolution to selected) 1row300A prints with row A only, 1row300B is row B only.		
Vertical Resolution	<L|M|none>	vres=<600|300|1row300A|1row300B>	RES:0	Set vertical resolution. 600 and 300 implies 2row printing. 1row300A prints with row A only, 1row300B is row B only.		
Horizontal Resolution	<L|M|none>	hres=<75|90|110|150|200|240|300| 320|400|440|480|500| 600|800|960|1200|1600|2400>	RES:0	Set horizontal resolution in DPI	PAR;hres=440# RES:0;…#	
Print direction	<L|M|none>	DIR=<right|left|bidir> direction=<right|left|bidir>	RES:0	Set print direction right, left or bidirectionel.		

	<L|M|none>	BDR=<right|left|inp1|inp2|inp3|inp4> bidirection=<right|left|inp1|inp2|inp3|inp4>		Set print start direction on bidirectionel mode. Right, left, input1 select, input2 select,…		Start direction is FIRST print direction after print mode has been activated.
Start distance	<L|M|none>	DIS=<value> (in pixels) start=<value> (in pixels)	RES:0	Set start distance from start signal to print i pixels.		
Start dist. (bidir)	<L|M|none>	DRT=<value> (right direction in pixels) DLT=<value> (left direction in pixels) start right=<value> (right direction in pixels) start left=<value> (left direction in pixels)	RES:0	Set bidir start distances from start signal to print.		
Start sensor edge	<L|M|none>	EDG=<pos|neg> edge=<positive|negative>	RES:0	Set startsensor active edge (+=positive, -=negative).		
Modular parameter	<L|M|none>	MOD=<value> modular=<value>	RES:0	Set modular parameter.		
Velocity parameter	<L|M|none>	VEL=<velocity> (in meters/min) velocity=<velocity> (in meters/min)	RES:0	Set velocity parameter. (decimal notation is allowed)	**PAR;VEL=25.33#** **RES:0;…#**	
Encoder parameter	<L|M|none>	ENC=<encoder_par> (in mm/pulse) encoder=<encoder_par> (in mm/pulse)	RES:0	Set encoder parameter. (decimal notation is allowed)	**PAR;ENC=0.00716#** **RES:0;…#**	
Quadrature	<L|M|none>	QDT=<+|-> quadrature=<+|->	RES:0	Set encoder quadrature on|off.		
Repeat	<L|M|none>	REP=<repeats>,<repeat_distance> (in pixels) repeat=<repeats>,<repeat_distance> (in pixels)	RES:0	Set print repeats and distance between repeats.		
Endless mode	<L|M|none>	ENL=<+|-> endless=<+|->	RES:0	Set endless print on/off.		
Print mode	<L|M|none>	ENM=<pos|mod|vel> mode=<position|modular|velocity>	RES:0	Set print mode to Position, Modular or Velocity.	**PAR;mode=velocity#** **RES:0;…#**	
Buffer mode	<L|M|none>	BUF=<+|-|u> buffermode=<+|-|u>	RES:0	Set print buffer mode (+)normal buffer (-)no buffer (u)user managed.		
Head offset	<L|M|none>	OFF|offset=<head nr>;<head offset>;<pen1 offset>;…	RES:0	Set offsets for a complete head. Offset are written with/without decimal point, and value may be appended with unit type (mm, in or px for mm, inches or	**PAR:OFF=1;0;0;25mm;50.1mm;75.2mm#** **RES:0;…#** **PAR:OFF=2;3in;0;1in#** **RES:0;…#**	Sets offsets for head 1, in this case a 4-pen stall with head offset 0 and pen offsets 0, 25mm, 50.1mm and 75.2mm.

				pixels notation).		Second example for head 1, head offset = 3 in, pen offset 2 = 1 inch
Head parameters	<L|M|none>	HEA|head=<head nr.>;<parameter>=<value> parameters: offset: OFF|offset=<value><unit> upside/down: UPS|updown=<+|-> otherside: OTH|othside=<+|-> voltage: VOL|voltage=<volt>;<fire width(us)> size: SIZ|size=<head_size> mirror: MIR|mirror=<+|->	RES:0	Set head parameter: Offset, Upsidedown, otherside, voltage or head size.	**PAR:head=1;UPS=+#** **RES:0;…#** **PAR:HEA=2;MIR=+#** **RES:0;…#**	Enable upsidedown for head 1. Set MIRROR = on for head 2
Pen parameters	<L|M|none>	PEN|pen=<head nr>;<pen nr>;<parm>=<value> parameters: turn off: OFF|off|- turn on: ON|on|+ offset: OFS|offset=<value><unit> overlap: OVL|overlap=<value> ink setup: INK|ink=<ink supply (ml)>;<ink low (ml)> ink reset: RST|reset pen switch: SWH|switch=<+|->	RES:0	Set pen parameters: turn off, turn on, offset, overlap, ink setup, ink reset and pen auto switch.	**PAR:PEN=1;1;INK=42;4#** **RES:0;…#**	Set ink supply size (42ml) and ink warning level (4ml) of pen 1 in head 1.
Set date	none	date=<year>/<month>/<day> (format yyyy/m/d)	RES:0	Set system date.	**PAR:date=2009/9/17#** **RES:0;…#**	
Set time	none	time=<hour>:<minute>:<second> (format h:m:s)	RES:0	Set system time.	**PAR:time=13:42:00#** **RES:0;…#**	
Set start sensor block distances	none	BLK|block=<start_block_dist(units)>;<end_block_dist(units)>	RES:0	Set start sensor start block distance and end block distance including units (mm|in|px).		
					PAR:M;RES=600;ENC=0,04:ENM=pos;BUF=-# **RES:0;…#**	Sets resolution, encoder parm, print mode and buffer mode Machine parameters in 1 command.

Parameter set selection

Name	Function	Parameters	Reply	Description	Example	Notes
Select parmaeters source	SEL	<resolution_source><sensor_source><repeat_source><head setup source> (L|M)	RES:0	Select resolution, sensor and repeat parameters source to (M)achine or (L)ayout	PAR:SEL;MMMM#	Set all parameters to Machine source.

COMMUNICATION PARAMETER selection

Name	Function	Parameters	Reply	Description	Example	Notes
Network setup	N	DHCP=<on|off> NAME=<network name> IP=<IP address> (dot '.' seperated) GATE=<Gateway address> MASK=<network mask> PORT=<ip port>,<file port>	RES:0	Set network parameters. Include parameters which needs to be changed in command. Note when changing network settings via network, the connection may be disconnected.	PAR:N;DHCP=off;IP=192.168.160.199;NAME=MT1# RES:0;…# PAR:N;PORT=9000,9300# RES:0;…#	Set DHCP off + IP address + network name. Example 2 changes default port settings
UART setup	U	BAUD=<value> (bits per sec) BITS=<5|6|7|8> PARITY=<none|even|odd> STOP=<1|2>	RES:0	Set RS232 connection parameters.	PAR:U;BAUD=115200;BITS=8;PARITY=none;STOP=2#	

REQUEST – Prefix REQ

TEXT OBJECT

Name	Function	Parameters	Reply	Description	Example	Notes
Text Object info	OBJ object	<text_object>	DAT:<object_name>=text; pos=<X>,<Y>pix; ori=<rotation_type>; font=<font_name>; <content_1_type>=<linked_content_1>;…; <content_n_type=<linked_content_n>; sep=<content_seperator_char>	Return information about a text object		See enumerations on rotation types and content types

BARCODE OBJECT

Name	Function	Parameters	Reply	Description	Example	Notes
Barcode Object info	OBJ object	<barcode_object>	DAT:<object_name>=barcode; pos=<X>,<Y>pix; ori=<rotation_type>;font=<font_name>; <content_1_type>=<linked_content_1>;…; <content_n_type=<linked_content_n>; sep=<content_seperator_char>; bartype=<barcode_type>; checktype=<checksum_0|1>; modules=<line_width>; blacksize=<black_size>; whitesize=<white_size>; human read=<human_readable>;	Return information about a barcode object		See enumeration on barcode types

			height=<lines_height>			

GRAPHIC OBJECT

Name	Function	Parameters	Reply	Description	Example	Notes

Grahic Object info (**line**)	OBJ object	<grahic_line_object>	DAT:<object_name>=graphic;
			type=<graphic_type>;
			pos=<X>,<Y>pix;
			length=<line_length>pix;
			line=<line_width>pix
				Request info on a line		

Graphic Object info (**box**)	OBJ object	<grahic_box_object>	DAT:<object_name>=graphic;
			type=<graphic_type>;
			pos=<X>,<Y>pix;
			width=<box_width>pix;
			height=<box_height>pix;
			line=<line_width>pix;
			fill=<box_fill+|->
				Request info on a box		

Graphic Object info (**logo**)	OBJ object	<grahic_logo_object>	DAT:<object_name>=graphic;
			type=<graphic_type>;
			pos=<X>,<Y>pix;
			width=<logo_width>pix;
			height=<logo_height>pix;
			logo=<logo_name>
				Request info on a logo		

CONTENT information

Name	Function	Parameters	Reply	Description	Example	Notes
Static Content info	CON content	<static_content>	DAT:<content_name>=static;tex=<static_text>	Request value of static content	REQ:CON;MyStatic# DAT:MyStatic=static; tex=Hello, World#	
Counter Content info	CON content	<counter_content>	DAT:<content_name>=counter;value=<current_value>; digits=<digits>;min=<min_value>;max=<max_value>; rep=<repeats>;step=<step_value>;leadin=<leadin_type>	Properties of counter content		
DateTime Content info	CON content	<data_time_content>	DAT:<content_name>datetime;format=<format>; offset=<offset>;locale=<locales_name>	Properties of date/time content		Offset format: <year>-<month>-<day> <hour>:<min>:<sec>
Shift Content info	CON content	<shift_content>	DAT:<content_name>=shift; timeshift=<shift_datetime_1>;data=<shift_data_1>;…; timeshift=<shift_datetime_n>;data=<shift_data_n>	Properties of shift content		Shift data format: <year>-<month>-<day> <hour>:<min>:<sec> There are maximum 5 rules.
System Content info	CON content	<system_content>	DAT:<content_name>=system;type=<system_type>	Properties of system content.		See types enumeration. Returns the type, not the actual value
Identifier content	CON content	<identifier_content>	DAT:<content_name>=identifier;ID=<ID_format>	Properties of GS1 identifier		

Other information that can be returned

Name	Function	Parameters	Reply	Description	Example	Notes

Parameters list	PAR
	parameters
		<L|M|none>	DAT:size=<picture_length>pix;
			start=<start_distance>pix;
			res=<resolution_type>;
			quad=<quadrature+|->;
			dir=<print_dir_right|left>;
			repeat=<repeats>,<repeat_dist>pix;
			mod=<modular_parm>;
			vel=<velocity_parm>;
			enc=<encoder_parm>;
			edge=<sensor_edge_pos|meg>;
			endless<endless_+|->;
			mode=<printmode_pos|mod|vel>;
			buffer=<buffer_mode_+|-|u>;
			headoffset=<head_offset_1>;headsize=<head_size_1>;
			penoffset<head_1_pen_offset_1>;...;
			penoffset<head_1_pen_offset_n>;....;
			head...=<head_n....>
				Return a lot of print parameters.	REQ:PAR;M#
					DAT:size=2560pix;
					start=400pix;
					res=600;
					….
						If the argument is left out, default is machine parameters

File list	DIR dir	<directory_path> (optional)	DAT:dir;<file/dir_name_1>;….;<file/dir_name_n>			directory names are indicated by '<'dir_name'>'
Active jobname	FIL filename		DAT:file=<job_name>	Request active file name. Does not change file.		
Active user	user		DAT:user=<user_name>	Request active username		

Object list	OLS objects		DAT:objects;<object_name_1>=<tex|bar|grp>;...; <object_name_n>=<tex|bar|grp>	Get an object list	**REQ:OLS#** **DAT:objects;S1=tex;** **S2=tex;obj145=tex#**	
Content list	CLS contents		DAT:contents; <static_name_1>=<sta>;..;<static_name_n>=<sta>; <counter_name_1>=<cnt>;...;<counter_name_n>=<cnt>; <date_time_name_1>=<dat>;...;<date_time_name_n>=<dat>; <shift_name_1>=<shf>;...;<shift_name_n>=<shf>; <system_name_1>=<sys>;...;<system_name_n>=<sys>; <identifier_name_1>=<id>;...;<identifier_name_n>=<id>	Get a content list	**REQ:Contents#** **DAT:contents;** **static1=sta;** **MyBarcode1=sta;** **TimeDate1=dat#**	
Version info	VER version		DAT:version; System=<MT_type>;ver=<firmware_version>; build=<build_date>;FPGA=<FPGA_version>	Get version information of controller firmware.	**REQ:version#** **DAT:version;System=** **MiniKey;ver=1.65C;** **build=15. jan 2011;** **FPGA=49.3#**	
Ethernet info	IP tcpip		DAT:tcpip;name=<network_name>;ip=<ip_address>; mask=<ip_netmask>;gateway=<ip_gateway>; ipport=<ip_com_port>;fileport=<ip_file_port>; dhcp=<on|off|failed>	Get information on TCP settings		Since you need to be connected to request it, must be known already ☺ Useful to uniquely identify each controller in a network
Uptime	UP uptime		DAT:uptime=<uptime_hours>hr	Get uptime since last power on		Integer hours
Print info	PI print info		DAT:print info;print=<on|off>;prints=<pictures_printed>	Get status of print mode and no. of prints done		

Ink info	II ink info		DAT:ink info;<ink_used_pen_1>;...;<ink_used_pen_n>	Get ink status	**REQ:II#** **DAT:ink info;** **3;4;40;13#**	Ink remaining = ink size minus ink status
Pen status	PS pen status		DAT:pen1=<pen_status_nr>;<pen_status_text>;pen2=…	Get status of each pen	**REQ:pen status#** **DAT:pen1=12;pen2=12;** **pen3=0;pen4=0#**	See penstatus enumeration. 12 is ”OK” status.
Print done interrupt	PD print done	<on|off>	DAT:print done=<on|off> (when print done and print interrupt on: SYS:PRD;<prints_since_last_interrupt>)	Request print done interrupt	**REQ:PD#** [print] **SYS:PRD;1#**	When set on and print done. MT returns SYS:PRD;<prints>

ENUMERATIONS

BARCODES

Barcodes types	Limitations
EAN8	8 digits (7 + checksum)
EAN13	13 digits (12+checksum)
UPCA	13 digits (12+checksum)
ITF	Even number of digits (uneven number + checksum)
Code39	Digits only
Code128	None
EAN128	Must have identifier structure
DM	None
GS1DM	Must have identifier structure

DATE / TIME

Date & Time format types | separates alternatives		
Day of week	#d | #D	
Day of year	#j | #J (zeros leadin)	
Week of year	#w | #W (zeros leadin)	
Year (4 chars)	yyyy | YYYY	
Year (2 chars)	yy | YY	
Year (1 char)	y | Y	
Month (2 chars)	mm | MM	zero leadin
Month (1-2 chars)	m | M	
Month (short names)	mmm | MMM	
Month (long names)	mmmm | MMMM	
Day (2 chars)	dd | DD	zero leadin
Day (1-2 chars)	d | DD	
Day (short names)	ddd | DDD	
Day (long names)	dddd | DDDD	
Hour (1-2 chars)	h | H	
Hour (2 chars)	hh | HH	zero leadin
AM/PM hours	pp | PP | Pp | pP	(Casing as indicated, eg. hhpp -> 02am, hPp -> 2Am)
Minute (1-2 chars)	n | N	
Minute (2 chars)	nn | NN	zero leadin
Seconds (1-2 chars)	s | S	
Seconds (2 chars)	ss | SS	zero leadin

CONTENT / OBJECTS

Content types	
sta	Static
cnt	Counter
dat	Date/Time
shf	Shift Code
sys	System var.
id	Identifier

Object types	
tex	Text objects
bar	Barcode objects
grp	Graphic objects

System types	
user	print user name
line	print line name
file	print job filename
print	print print number

Grahic types	
0	Line
1	Box
2	Logo

PARAMETERS

Pen status decode:
0: Offline
1: Power off
2: Hot
3: Warm
4: Short
5: Power low
6: Power high
7: Fault
8: New pen installed
9: Stall open
10: Ink low
11: No ink
12: Pen online

Print resolutions types (Vertical)	
resolution text	resolution
600	600 DPI
300	300 DPI (2 rows highspeed)
1row300A	300 DPI (one row, row A)
1row300B	300 DPI (one row, row B)

ERROR CODES

Code		Description	Explanation / when returned
RS-232	Ethernet		
0	0	Transmission OK	Command understood
1	2	Unknown command	Initial command incorrect. Take care when sending data from a script, it's easy to send **<CR><LF>Command…** which will fail since CR and LF are also interpreted
2	300	Object not found	Request object names first
3	301	OBJ: not a number	Numeral expected
4	310	POS: x not a number	Numeral expected
5	311	POS: y not a number	Numeral expected
6	318	FONT: not a number	Numeral expected (existing font)
7	321	FONT: invalid font	
8	340	LOGO: unknown logo	
9	343	LOGO: invalid logo	
10	352	BARCODE: function failed	
11	353	BARCODE: unknown type	
12	354	BARCODE: invalid checksum	
13	600	SYSTEM: unknown variable	
14	602	TEXT: function failed	
15	1010	PAR: not a number	
16	1020	PAR: unknown edge	
17	1050	PAR: unknown printmode	
28	220	Printing, can't start now	Tried to start print when already printing
29	221	Stopped, can't stop now	Tried to stop print when already stopped
30	104	User remote login not allowed	A user is already logged into the controller. Remote user MUST have admin rights to overrule
31	105	Not connected	No commands can be executed if not logged in
32	101	Username not found	See userlist on controller.
33	102	Password not accepted	
34	210	File not found	
35	504	Not found	
36	106	Parameters changes not allowed	Parameter access off for user name
37	404	Object changes not allowed	Object access off for user name
44	4001	BUF: Print buffer full	Tried to add more than 4 images to user-managed buffer
45	4002	BUF: Print buffer empty	Tried to print on user managed (empty) buffer

ENCODING

Is is important to understand that MiniTouch and MiniKey are NOT using UniCode standards (multiple bytes per character) to store character values. Instead, all characters are stored as a SINGLE byte value.

The actual character that will appear on the screen and in print will depend on the character encoding selected on the font. In other words, a font encoded in "Western" may display a totally different character from a font encoded in "Cyrillic". Whereas with UniCode, the character would always be the same.

MiniTouch / MiniKey supports the following encodings:

Codepage	Language group
Win-1250	Central / Eastern Europe
Win-1251	Cyrillic
Win-1252	English / Western Europe
Win-1253	Greek (Modern)
Win-1254	Turkish
Win-1257	Baltic
Win-1258	Vietnamese
Win-932	Japanese (Shift-JIS), half-width KATAKANA only

Notice: Although encodings exist for Hebrew and Arab, these languages are not supported as the controller can not combine characters in the correct way, and does not respect the R2L direction.

All other languages (character sets) not covered in the table above are generally NOT supported. This especially includes all "complex scripts" (chinese, japanese, korean, arab, farsi, pashtun, hebrew, thai etc).

Notice also that UniCode contains many more diacritics not found in the encodings above. (example: **Ẑ)**

The encodings above are fully supported given the right byte values. This includes screen menus, locales, keyboard and variable print.

In order to communicate strings the raw byte values must be sent and interpreted when received. It is up to the controlling software / device to know the encoding.

Encoding tables

Windows 1250

Central and Eastern Europe

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL 0000	STX 0001	SOT 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
10	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
20	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
30	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
40	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
50	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
60	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
70	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	DEL 007F
80	€ 20AC		‚ 201A	ƒ 0192	„ 201E	… 2026	† 2020	‡ 2021	ˆ 02C6	‰ 2030	Š 0160	‹ 2039	Œ 0152		Ž 017D	
90		‘ 2018	’ 2019	“ 201C	” 201D	• 2022	– 2013	— 2014	˜ 02DC	™ 2122	š 0161	› 203A	œ 0153		ž 017E	Ÿ 0178
A0	NBSP 00A0	¡ 00A1	¢ 00A2	£ 00A3	¤ 00A4	¥ 00A5	¦ 00A6	§ 00A7	¨ 00A8	© 00A9	ª 00AA	« 00AB	¬ 00AC	- 00AD	® 00AE	¯ 00AF
B0	° 00B0	± 00B1	² 00B2	³ 00B3	´ 00B4	µ 00B5	¶ 00B6	· 00B7	¸ 00B8	¹ 00B9	º 00BA	» 00BB	¼ 00BC	½ 00BD	¾ 00BE	¿ 00BF
C0	À 00C0	Á 00C1	Â 00C2	Ã 00C3	Ä 00C4	Å 00C5	Æ 00C6	Ç 00C7	È 00C8	É 00C9	Ê 00CA	Ë 00CB	Ì 00CC	Í 00CD	Î 00CE	Ï 00CF
D0	Ð 00D0	Ñ 00D1	Ò 00D2	Ó 00D3	Ô 00D4	Õ 00D5	Ö 00D6	× 00D7	Ø 00D8	Ù 00D9	Ú 00DA	Û 00DB	Ü 00DC	Ý 00DD	Þ 00DE	ß 00DF
E0	à 00E0	á 00E1	â 00E2	ã 00E3	ä 00E4	å 00E5	æ 00E6	ç 00E7	è 00E8	é 00E9	ê 00EA	ë 00EB	ì 00EC	í 00ED	î 00EE	ï 00EF
F0	ð 00F0	ñ 00F1	ò 00F2	ó 00F3	ô 00F4	õ 00F5	ö 00F6	÷ 00F7	ø 00F8	ù 00F9	ú 00FA	û 00FB	ü 00FC	ý 00FD	þ 00FE	ÿ 00FF

Windows 1251

Cyrillic

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL 0000	STX 0001	SOT 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
10	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
20	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
30	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
40	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
50	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
60	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
70	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	DEL 007F
80	Ђ 0402	Ѓ 0403	‚ 201A	ѓ 0453	„ 201E	… 2026	† 2020	‡ 2021	€ 20AC	‰ 2030	Љ 0409	‹ 2039	Њ 040A	Ќ 040C	Ћ 040B	Џ 040F
90	ђ 0452	‘ 2018	’ 2019	“ 201C	” 201D	• 2022	– 2013	— 2014		™ 2122	љ 0459	› 203A	њ 045A	ќ 045C	ћ 045B	џ 045F
A0	NBSP 00A0	Ў 040E	ў 045E	Ј 0408	¤ 00A4	Ґ 0490	¦ 00A6	§ 00A7	Ё 0401	© 00A9	Є 0404	« 00AB	¬ 00AC	- 00AD	® 00AE	Ї 0407
B0	° 00B0	± 00B1	І 0406	і 0456	ґ 0491	µ 00B5	¶ 00B6	· 00B7	ё 0451	№ 2116	є 0454	» 00BB	ј 0458	Ѕ 0405	ѕ 0455	ї 0457
C0	А 0410	Б 0411	В 0412	Г 0413	Д 0414	Е 0415	Ж 0416	З 0417	И 0418	Й 0419	К 041A	Л 041B	М 041C	Н 041D	О 041E	П 041F
D0	Р 0420	С 0421	Т 0422	У 0423	Ф 0424	Х 0425	Ц 0426	Ч 0427	Ш 0428	Щ 0429	Ъ 042A	Ы 042B	Ь 042C	Э 042D	Ю 042E	Я 042F
E0	а 0430	б 0431	в 0432	г 0433	д 0434	е 0435	ж 0436	з 0437	и 0438	й 0439	к 043A	л 043B	м 043C	н 043D	о 043E	п 043F
F0	р 0440	с 0441	т 0442	у 0443	ф 0444	х 0445	ц 0446	ч 0447	ш 0448	щ 0449	ъ 044A	ы 044B	ь 044C	э 044D	ю 044E	я 044F

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL 0000	STX 0001	SOT 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
10	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
20	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
30	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
40	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
50	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
60	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
70	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	DEL 007F
80	€ 20AC		‚ 201A	ƒ 0192	„ 201E	… 2026	† 2020	‡ 2021	ˆ 02C6	‰ 2030	Š 0160	‹ 2039	Œ 0152		Ž 017D	
90		‘ 2018	’ 2019	“ 201C	” 201D	• 2022	– 2013	— 2014	˜ 02DC	™ 2122	š 0161	› 203A	œ 0153		ž 017E	Ÿ 0178
A0	NBSP 00A0	¡ 00A1	¢ 00A2	£ 00A3	¤ 00A4	¥ 00A5	¦ 00A6	§ 00A7	¨ 00A8	© 00A9	ª 00AA	« 00AB	¬ 00AC	- 00AD	® 00AE	¯ 00AF
B0	° 00B0	± 00B1	² 00B2	³ 00B3	´ 00B4	µ 00B5	¶ 00B6	· 00B7	¸ 00B8	¹ 00B9	º 00BA	» 00BB	¼ 00BC	½ 00BD	¾ 00BE	¿ 00BF
C0	À 00C0	Á 00C1	Â 00C2	Ã 00C3	Ä 00C4	Å 00C5	Æ 00C6	Ç 00C7	È 00C8	É 00C9	Ê 00CA	Ë 00CB	Ì 00CC	Í 00CD	Î 00CE	Ï 00CF
D0	Ð 00D0	Ñ 00D1	Ò 00D2	Ó 00D3	Ô 00D4	Õ 00D5	Ö 00D6	× 00D7	Ø 00D8	Ù 00D9	Ú 00DA	Û 00DB	Ü 00DC	Ý 00DD	Þ 00DE	ß 00DF
E0	à 00E0	á 00E1	â 00E2	ã 00E3	ä 00E4	å 00E5	æ 00E6	ç 00E7	è 00E8	é 00E9	ê 00EA	ë 00EB	ì 00EC	í 00ED	î 00EE	ï 00EF
F0	ð 00F0	ñ 00F1	ò 00F2	ó 00F3	ô 00F4	õ 00F5	ö 00F6	÷ 00F7	ø 00F8	ù 00F9	ú 00FA	û 00FB	ü 00FC	ý 00FD	þ 00FE	ÿ 00FF

Win 1251

English and Western Europe

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL 0000	STX 0001	SOT 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
10	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
20	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
30	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
40	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
50	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
60	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
70	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	DEL 007F
80	€ 20AC		‚ 201A	ƒ 0192	„ 201E	… 2026	† 2020	‡ 2021		‰ 2030		‹ 2039				
90		‘ 2018	’ 2019	“ 201C	” 201D	• 2022	– 2013	— 2014		™ 2122		› 203A				
A0	NBSP 00A0	΅ 0385	Ά 0386	£ 00A3	¤ 00A4	¥ 00A5	¦ 00A6	§ 00A7	¨ 00A8	© 00A9		« 00AB	¬ 00AC	- 00AD	® 00AE	― 2015
B0	° 00B0	± 00B1	² 00B2	³ 00B3	΄ 0384	µ 00B5	¶ 00B6	· 00B7	Έ 0388	Ή 0389	Ί 038A	» 00BB	Ό 038C	½ 00BD	Ύ 038E	Ώ 038F
C0	ΐ 0390	Α 0391	Β 0392	Γ 0393	Δ 0394	Ε 0395	Ζ 0396	Η 0397	Θ 0398	Ι 0399	Κ 039A	Λ 039B	Μ 039C	Ν 039D	Ξ 039E	Ο 039F
D0	Π 03A0	Ρ 03A1		Σ 03A3	Τ 03A4	Υ 03A5	Φ 03A6	Χ 03A7	Ψ 03A8	Ω 03A9	Ϊ 03AA	Ϋ 03AB	ά 03AC	έ 03AD	ή 03AE	ί 03AF
E0	ΰ 03B0	α 03B1	β 03B2	γ 03B3	δ 03B4	ε 03B5	ζ 03B6	η 03B7	θ 03B8	ι 03B9	κ 03BA	λ 03BB	μ 03BC	ν 03BD	ξ 03BE	ο 03BF
F0	π 03C0	ρ 03C1	ς 03C2	σ 03C3	τ 03C4	υ 03C5	φ 03C6	χ 03C7	ψ 03C8	ω 03C9	ϊ 03CA	ϋ 03CB	ό 03CC	ύ 03CD	ώ 03CE	

Windows 1253

Modern Greek

Win 1254

Turkish

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL 0000	STX 0001	SOT 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
10	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
20	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
30	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
40	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
50	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
60	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
70	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	DEL 007F
80	€ 20AC		‚ 201A	ƒ 0192	„ 201E	… 2026	† 2020	‡ 2021	ˆ 02C6	‰ 2030	Š 0160	‹ 2039	Œ 0152			
90		‘ 2018	’ 2019	“ 201C	” 201D	• 2022	– 2013	— 2014	˜ 02DC	™ 2122	š 0161	› 203A	œ 0153			Ÿ 0178
A0	NBSP 00A0	¡ 00A1	¢ 00A2	£ 00A3	¤ 00A4	¥ 00A5	¦ 00A6	§ 00A7	¨ 00A8	© 00A9	ª 00AA	« 00AB	¬ 00AC	- 00AD	® 00AE	¯ 00AF
B0	° 00B0	± 00B1	² 00B2	³ 00B3	´ 00B4	µ 00B5	¶ 00B6	· 00B7	¸ 00B8	¹ 00B9	º 00BA	» 00BB	¼ 00BC	½ 00BD	¾ 00BE	¿ 00BF
C0	À 00C0	Á 00C1	Â 00C2	Ã 00C3	Ä 00C4	Å 00C5	Æ 00C6	Ç 00C7	È 00C8	É 00C9	Ê 00CA	Ë 00CB	Ì 00CC	Í 00CD	Î 00CE	Ï 00CF
D0	Ğ 011E	Ñ 00D1	Ò 00D2	Ó 00D3	Ô 00D4	Õ 00D5	Ö 00D6	× 00D7	Ø 00D8	Ù 00D9	Ú 00DA	Û 00DB	Ü 00DC	İ 0130	Ş 015E	ß 00DF
E0	à 00E0	á 00E1	â 00E2	ã 00E3	ä 00E4	å 00E5	æ 00E6	ç 00E7	è 00E8	é 00E9	ê 00EA	ë 00EB	ì 00EC	í 00ED	î 00EE	ï 00EF
F0	ğ 011F	ñ 00F1	ò 00F2	ó 00F3	ô 00F4	õ 00F5	ö 00F6	÷ 00F7	ø 00F8	ù 00F9	ú 00FA	û 00FB	ü 00FC	ı 0131	ş 015F	ÿ 00FF

Windows 1257

Baltic Region

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL 0000	STX 0001	SOT 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
10	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
20	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
30	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
40	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
50	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
60	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
70	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	DEL 007F
80	€ 20AC		‚ 201A		„ 201E	… 2026	† 2020	‡ 2021		‰ 2030		‹ 2039		¨ 00A8	ˇ 02C7	¸ 00B8
90		‘ 2018	’ 2019	“ 201C	” 201D	• 2022	– 2013	— 2014		™ 2122		› 203A		¯ 00AF	˛ 02DB	
A0	NBSP 00A0		¢ 00A2	£ 00A3	¤ 00A4		¦ 00A6	§ 00A7	Ø 00D8	© 00A9	Ŗ 0156	« 00AB	¬ 00AC	- 00AD	® 00AE	Æ 00C6
B0	° 00B0	± 00B1	² 00B2	³ 00B3	´ 00B4	µ 00B5	¶ 00B6	· 00B7	ø 00F8	¹ 00B9	ŗ 0157	» 00BB	¼ 00BC	½ 00BD	¾ 00BE	æ 00E6
C0	Ą 0104	Į 012E	Ā 0100	Ć 0106	Ä 00C4	Å 00C5	Ę 0118	Ē 0112	Č 010C	É 00C9	Ź 0179	Ė 0116	Ģ 0122	Ķ 0136	Ī 012A	Ļ 013B
D0	Š 0160	Ń 0143	Ņ 0145	Ó 00D3	Ō 014C	Õ 00D5	Ö 00D6	× 00D7	Ų 0172	Ł 0141	Ś 015A	Ū 016A	Ü 00DC	Ż 017B	Ž 017D	ß 00DF
E0	ą 0105	į 012F	ā 0101	ć 0107	ä 00E4	å 00E5	ę 0119	ē 0113	č 010D	é 00E9	ź 017A	ė 0117	ģ 0123	ķ 0137	ī 012B	ļ 013C
F0	š 0161	ń 0144	ņ 0146	ó 00F3	ō 014D	õ 00F5	ö 00F6	÷ 00F7	ų 0173	ł 0142	ś 015B	ū 016B	ü 00FC	ż 017C	ž 017E	˙ 02D9

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL 0000	STX 0001	SOT 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
10	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
20	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
30	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
40	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
50	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
60	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
70	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	DEL 007F
80	€ 20AC		‚ 201A	ƒ 0192	„ 201E	… 2026	† 2020	‡ 2021	ˆ 02C6	‰ 2030		‹ 2039	Œ 0152			
90		' 2018	' 2019	" 201C	" 201D	• 2022	– 2013	— 2014	˜ 02DC	™ 2122		› 203A	œ 0153			Ÿ 0178
A0	NBSP 00A0	¡ 00A1	¢ 00A2	£ 00A3	¤ 00A4	¥ 00A5	¦ 00A6	§ 00A7	¨ 00A8	© 00A9	ª 00AA	« 00AB	¬ 00AC	- 00AD	® 00AE	¯ 00AF
B0	° 00B0	± 00B1	² 00B2	³ 00B3	´ 00B4	µ 00B5	¶ 00B6	· 00B7	¸ 00B8	¹ 00B9	º 00BA	» 00BB	¼ 00BC	½ 00BD	¾ 00BE	¿ 00BF
C0	À 00C0	Á 00C1	Ã 00C2	Ă 0102	Ä 00C4	Å 00C5	Æ 00C6	Ç 00C7	È 00C8	É 00C9	Ê 00CA	Ë 00CB	̀ 0300	Í 00CD	Î 00CE	Ï 00CF
D0	Đ 0110	Ñ 00D1	̉ 0309	Ó 00D3	Ô 00D4	Ơ 01A0	Ö 00D6	× 00D7	Ø 00D8	Ù 00D9	Ú 00DA	Û 00DB	Ü 00DC	Ư 01AF	̃ 0303	ß 00DF
E0	à 00E0	á 00E1	ã 00E2	ă 0103	ä 00E4	å 00E5	æ 00E6	ç 00E7	è 00E8	é 00E9	ê 00EA	ë 00EB	́ 0301	í 00ED	î 00EE	ï 00EF
F0	đ 0111	ñ 00F1	̣ 0323	ó 00F3	ô 00F4	ơ 01A1	ö 00F6	÷ 00F7	ø 00F8	ù 00F9	ú 00FA	û 00FB	ü 00FC	ư 01B0	₫ 20AB	ÿ 00FF

Win 1258

Vietnamese

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL 0000	STX 0001	SOT 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
10	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
20	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
30	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
40	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
50	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	¥ 005C	] 005D	^ 005E	_ 005F
60	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
70	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	DEL 007F
80		81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
90	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
A0		。 FF61	「 FF62	」 FF63	、 FF64	・ FF65	ヲ FF66	ァ FF67	ィ FF68	ゥ FF69	ェ FF6A	ォ FF6B	ャ FF6C	ュ FF6D	ョ FF6E	ッ FF6F
B0	ー FF70	ア FF71	イ FF72	ウ FF73	エ FF74	オ FF75	カ FF76	キ FF77	ク FF78	ケ FF79	コ FF7A	サ FF7B	シ FF7C	ス FF7D	セ FF7E	ソ FF7F
C0	タ FF80	チ FF81	ツ FF82	テ FF83	ト FF84	ナ FF85	ニ FF86	ヌ FF87	ネ FF88	ノ FF89	ハ FF8A	ヒ FF8B	フ FF8C	ヘ FF8D	ホ FF8E	マ FF8F
D0	ミ FF90	ム FF91	メ FF92	モ FF93	ヤ FF94	ユ FF95	ヨ FF96	ラ FF97	リ FF98	ル FF99	レ FF9A	ロ FF9B	ワ FF9C	ン FF9D	゛ FF9E	゜ FF9F
E0	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC			

Windows 932

Japanese (Shift-JIS)

NOTICE: Shaded byte values are meant for double byte characters. These are NOT available.

This table shows half-width KATAKANA only.

Tools and Example Code (coming soon)

HSA Systems has developed a number of tools and example code that you can use as inspiration for your projects.

This section will contain an overview of the tools and examples.

Support

For technical support in remote communication to MiniTouch and MiniKey, please contact

applications@hsasystems.com