Matthews MPERIA Command Protocol Manual

for MPERIA 9.2.x

Table of Contents

1 - Overview....5
2 - Protocol....6
2.1 - Availability....6
2.2 - Commands and responses....6
2.3 - Installation context....7
2.4 - ASCII control characters....7
3 - Command overview....8
3.1 - Installations....8
3.2 - Messages....10
3.3 - Counters....10
3.4 - Database tables....11
3.5 - Variables....11
3.6 - L-series support....11
3.7 - T-series support....12
3.8 - V-series support....12
3.9 - I/O control....12
3.10 - Miscellaneous....12
4 - Command reference....13
4.1 - BACKUP....14
4.2 - CODEPAGE....15
4.3 - CNT_DEL....15
4.4 - CNT_GET....16
4.5 - CNT_GET_I....17
4.6 - CNT_GET_R....18
4.7 - CNT_GET_RC....19
4.8 - CNT_LIST....20
4.9 - CNT_RESET....21
4.10 - CNT_SET....22
4.11 - CNT_SET_I....23
4.12 - CNT_SET_R....24
4.13 - CNT_SET_RC....25
4.14 - CNT_UPDATE....26
4.15 - DB....27
4.16 - DB_ADD_ROW....28
4.17 - DB_ADD_TABLE....29
4.18 - DB_DEL_ROW....30
4.19 - DB_DEL_TABLE....31
4.20 - DB_GET....32
4.21 - DB_INC....33
4.22 - DB_INIT_TABLE....34
4.23 - DB_SET....35
4.24 - ENC_BLOCK....36
4.25 - ENC_PPM....37
4.26 - ENC_RESET_REVERSE....38
4.27 - GOOD....39
4.28 - INPUT_GET....40

4.29 - INST....41
4.30 - L_DECAP....42
4.31 - L_DECAP_MARKER....43
4.32 - L_DECAP_TABLE....44
4.33 - LOG....45
4.34 - MSG_DEL....46
4.35 - MSG_LIST....47
4.36 - MARKER_ENABLE....48
4.37 - MARKER_DISABLE....49
4.38 - MARKER_DIR....50
4.39 - MARKER_MARGIN....51
4.40 - MARKER_UPSIDEDOWN....52
4.41 - MARK_GAP....53
4.42 - MARK_INTERVAL....54
4.43 - MARK_REPEAT....55
4.44 - NONCRITICAL....56
4.45 - OFFLINE....57
4.46 - ONLINE....58
4.47 - OUTPUT_SET....59
4.48 - PEEKMSG....60
4.49 - PLUGIN_ACTION....61
4.50 - PRINT_CNT....62
4.51 - PRINT_CNT_RESET....63
4.52 - QUEUE....64
4.53 - QUEUE_LEN....65
4.54 - QUEUE_RESET....66
4.55 - SEL....67
4.56 - SELDB....68
4.57 - SELMSG....69
4.58 - SET_CLOCK....70
4.59 - SET_SPEED....71
4.60 - START....72
4.61 - STATE....73
4.62 - STATUS....74
4.63 - STOP....77
4.64 - T_CLEAN....78
4.65 - T_CLEAN_MARKER....79
4.66 - TRIG....80
4.67 - TRIG_ACTIVE....81
4.68 - USR_LVL_GET....82
4.69 - USR_LVL_SET....82
4.70 - V_FLUSH....83
4.71 - V_FLUSH_MARKER....84
4.72 - V_SET_CLEANER....85
4.73 - V_SET_CLEANER_MARKER....86
4.74 - V_SET_DROP_TOF....87
4.75 - VAR_DEL....88
4.76 - VAR_GET....89
4.77 - VAR_LIST....90

4.78 - VAR_SET....91
4.79 - VAR_SET2....92
4.80 - VERSION....93
5 - Revision history....94

1 Overview

The MPERIA platform is designed from the ground-up to be an automation controller for marking/coding, packaging and warehouse/distribution control systems. Real-time inkjet and laser printing, and labeling all require data input from a host. Most marking and printing technologies have specific formats that need to be sent by a host.

MPERIA supports three different protocols that can be used by remote systems. Each protocol has it's own strengths and weaknesses, which makes it important to pick the right protocol to use.

Line Action Protocol	Advantages: • Simple line based protocol. Just send the line of text. • The GUI is used to configure what the received line is used for, e.g. selecting a message or setting a variable. Disadvantages: • Can only do one thing per port, e.g. select a message, but a single port cannot set a variable and a counter. • Can only do simple things. No support for triggering, performing maintenance, reading status, etc.
XML Protocol	Advantages: • Based on the XML standard, which means that most high level systems have tools to generate and parse XML. • Supports advanced functions such as creating messages. Disadvantages: • Requires XML support. • Cannot be used on serial ports.
Command Protocol	Advantages: • Easy to use. • Extremely easy to parse. Responses are single bytes. • Low overhead. Disadvantages: • While more powerful than the Line Action Protocol, it cannot for example create messages.

2 Protocol

2.1 Availability

The Command Protocol is always available on TCP/IP ports 10005 and 10006. The difference between the two ports is that port 10006 does not send any responses back.

The Command Protocol can also be enabled for serial ports using the Protocol Configuration screen.

The Command Protocol was introduced in MPERIA 2.5.

2.2 Commands and responses

The Command Protocol is a command/response protocol. Commands are sent from the remote host to the Controller. Every command causes a response to be sent from the Controller back to the remote host.

All commands sent to the Controller has the following structure:

<**ESC**>*command*<**FS**>*arg0*<**FS**>*arg1*<**FS**> **...** <**FS**>*argN*<**ETB**>

If a command does not have any arguments, the structure simply becomes

<**ESC**>*command*<**ETB**>

A command with a single argument becomes

<**ESC**>*command*<**FS**>*arg0*<**ETB**>

A command with two argument becomes

<**ESC**>*command*<**FS**>*arg0*<**FS**>*arg1*<**ETB**>

Every command sent to the Controller causes the Controller to send a single byte back to indicate whether the command was successfully executed or not. If the command was successful, an <**ACK**> is sent back, and if the command failed, a <**NAK**> is sent back.

Some commands also provide data after the <**ACK**>. Such data always end with a single <**ETB**> byte. If the data contains several values, they are separated by <**FS**> bytes.

The protocol uses UTF-8 to encode Unicode strings.

It is recommended to set the NoDelay property on the socket when using the Command Protocol over TCP/IP and not use the Nagle algorithm to reduce communication delay.

2.3 Installation context

Several commands, e.g. STOP, START and SELMSG, operate on Installations. These commands do not take names of Installations as arguments. Instead, the INST command is used to specify which Installations later commands operate on. This has several advantages:

- Single Installation Controllers never need to execute an INST command.
- Multiple operations on a single Installation (or a set of Installations) do not need to repeat the Installation name for each command, which makes the communication more compact.

2.4 ASCII control characters

The Command Line Protocol uses the following ASCII control characters:

ASCII	Hex	Dec	Description
<ACK>	06	6	Acknowledge. Sent from the Controller to indicate the success of a command sent to the Controller.
<NAK>	15	21	Negative acknowledge. Sent from the Controller to indicate the failure of a command sent to the Controller.
<ETB>	17	23	End of Transmission Block. Used as the final character in all commands sent to the Controller, and as the end of data sent from the Controller as a response to a command.
<ESC>	1B	27	Escape. Used as the starting character in all commands sent to the Controller.
<FS>	1C	28	Unit Separator. Separates the command and arguments sent to the Controller, and multiple responses sent from the Controller as a response to a command.
<GS>	1D	29	Group Separator. Separates components of command arguments into groups.
<RS>	1E	30	Record separator used to separate groups into records.
<US>	1F	31	Unit separator used to separate records into units.

3 Command overview

3.1 Installations

Commands to manage Installations:

Command	Description
OFFLINE	Set Installations in offline mode.
ONLINE	Set Installations in online mode.
STOP	Stop Installations.
START	Start Installations.
STATE	Get the state of a single Installation.
STATUS	Get the status of a Installations.
GOOD	Test if Installations are good, meaning that they do not require service and have no critical problems.
NONCRITICAL	Test if Installations are having any critical problems preventing them from printing.

Commands to manage message selection:

Command	Description
SELMSG	Select message.
SELDB	Select database row for printing.
SEL	Select message and database row for printing.
PEEKMSG	Get the name of the currently selected message.

Commands to manage printing:

Command	Description
TRIG	Trigger printing.
TRIG_ACTIVE	Get the current trigger status (active/inactive).
QUEUE	Fixate and queue the selected message.
QUEUE_RESET	Reset the print queue.
QUEUE_LEN	Get the print queue length.
MARKER_ENABLE	Enable marker
MARKER_DISABLE	Disable marker
MARKER_DIR	Set the marker print direction.
MARKER_MARGIN	Set the marker margin.
SET_SPEED	Set the fixed speed.

Command	Description
PRINT_CNT	Get the print count for messages and Installations.
PRINT_CNT_RESET	Reset the print count for messages and Installations.
ENC_BLOCK	Blocks encoder.
ENC_RESET_REVERSE	Resets the encoder’s backward state.

3.2 Messages

Commands used to manage messages:

Command	Description
MSG_DEL	Delete a message.
MSG_LIST	List all available messages.

3.3 Counters

Commands used to manage counters:

Command	Description
CNT_LIST	List all available counters.
CNT_GET	Get the current value of a counter.
CNT_SET	Set the current value of a counter.
CNT_RESET	Reset the current value of a counter to the start value.
CNT_DEL	Delete a counter.
CNT_GET_RC	Get the current repeat iteration of a counter.
CNT_SET_RC	Set the current repeat iteration of a counter.
CNT_GET_I	Get the increment of a counter, which is by how much it is incremented when it is incremented.
CNT_SET_I	Set the increment of a counter, which is by how much it is incremented when it is incremented.
CNT_GET_R	Get the repeat count of a counter, which is by how many times a counter value is used before it is incremented.
CNT_SET_R	Set the repeat count of a counter, which is by how many times a counter value is used before it is incremented.
CNT_UPDATE	Update the counter value

3.4 Database tables

Commands used to manage database tables:

Command	Description
DB	Select the database table to operate on.
DB_ADD_TABLE	Create a new database table.
DB_DEL_TABLE	Delete a database table.

Commands used to manage the selected database table:

Command	Description
DB_INIT_TABLE	Initialize the database table.
DB_ADD_ROW	Add a row to the database table.
DB_DEL_ROW	Delete one or more rows from the database table.
DB_GET	Get values from a row in the database table.
DB_SET	Set values in a row in the database table.
DB_INC	Increment a value in a row in the database table.

3.5 Variables

Commands used to manage variables:

Command	Description
VAR_LIST	List all available variables.
VAR_GET	Get the value of a variable.
VAR_SET	Set the value of a variable.
VAR_SET2	Set the value of one or more variables.
VAR_DEL	Delete a variable.

3.6 L-series support

Commands specific to L-series print heads:

Command	Description
L_DECAP	Decap all L-series cartridges in the selected Installations.
L_DECAP_MARKER	Decap all L-series cartridges in the selected Marker.
L_DECAP_TABLE	Set the decap configuration table for a L-series ink.

3.7 T-series support

Commands specific to T-series print heads:

Command	Description
T_CLEAN	Clean all T-series print heads in the selected Installations.
T_CLEAN_MARKER	Clean all T-series print heads in the selected Marker.

3.8 V-series support

Commands specific to T-series print heads:

Command	Description
V_FLUSH	Flush all V-series print heads in the selected Installations.
V_FLUSH_MARKER	Flush all V-series print heads in the selected Marker.
V_SET_CLEANER	Enable cleaner for all V-series print heads in the selected Installations.
V_SET_CLEANER_MARKER	Enable cleaner for all V-series print heads in the selected Marker.

3.9 I/O control

Command for controlling external I/Os:

Command	Description
INPUT_GET	Get input state.
OUTPUT_SET	Set output state.

3.10 Miscellaneous

Miscellaneous commands:

Command	Description
BACKUP	Create a backup on an attached USB memory stick.
CODEPAGE	Set Code page encoding.
LOG	Create a log entry in the MPERIA log.
PLUGIN_ACTION	Run a plugin function.
USR_LVL_GET	Get the current UI access level.
USR_LVL_SET	Limit user’s access to the UI.
VERSION	Get the version of the MPERIA Controller.

4 Command reference

4.1 BACKUP

Description:

Creates a backup on an attached USB memory stick. The backup will fail if a USB memory stick is not attached.

The backup procedure may take a very long time which requires a long timeout on the call.

Command:

<**ESC**>BACKUP<**FS**>fileName<**FS**>includeDebugInfo<**ETB**>

Field	Description
fileName	Name of the backup file. The name will automatically be extended with ".lbb" by the system.
includeDebugInfo	Optional boolean value indication whether or not to include debug information in the backup. Default value is "false".

Successful response:

<**ACK**>

Example:

Create a backup with the name "BackupFile.lbb" including debug information:

<**ESC**>BACKUP<**FS**>BackupFile<**FS**>true<**ETB**>

Example response:

<**ACK**>

4.2 CODEPAGE

Description:

Set code page encoding. Default value is UTF8.

Command:

<ESC>CODEPAGE**<FS>**name**<ETB>**

Field	Description
name	Available code pages are latin1 and utf8.

Successful response:

<ACK>

Example:

Change code page to Latin1.

<ESC>CODEPAGE**<FS>**latin1**<ETB>**

Example response:

<ACK>

4.3 CNT_DEL

Description:

Delete a counter. Counters can only be deleted if they are not in use. Attempting to delete a counter that is in used causes a **<NAK>** to be returned.

Command:

<ESC>CNT_DEL**<FS>***name***<ETB>**

Field	Description
name	Name of the counter.

Successful response:

<ACK>

Example:

To delete the counter MYCOUNTER, send:

<ESC>CNT_DEL**<FS>**MYCOUNTER**<ETB>**

Example response:

<ACK>

4.4 CNT_GET

Description:

Get the current value of a counter.

If this command is used to set a non-global counter, the following rules apply:

- If the *INST* command has been executed with an empty list of Installations, the global value of the counter will be returned.
- If the *INST* command was executed with a list of Installations, only the local value of the first Installation will be returned.

Note: Use the command INST to select which installations to operate on.

Command:

<ESC>CNT_GET**<FS>***name***<ETB>**

Field	Description
name	Name of the counter.

Successful response:

<ACK>*value***<ETB>**

Field	Description
value	Value of the counter.

Example:

To get the value of the counter MYCOUNTER, send:

<ESC>CNT_GET**<FS>**MYCOUNTER**<ETB>**

Example response:

<ACK>42**<ETB>**

4.5 CNT_GET_I

Description:

Get the increment of a counter, which is by how much it is incremented when it is incremented.

Command:

<ESC>CNT_GET_I**<FS>***name***<ETB>**

Field	Description
name	Name of the counter.

Successful response:

<ACK>*value***<ETB>**

Field	Description
value	The counter's increment.

Example:

To get the increment of the counter MYCOUNTER, send:

<ESC>CNT_GET_I**<FS>**MYCOUNTER**<ETB>**

Example response:

<ACK>1**<ETB>**

4.6 CNT_GET_R

Description:

Get the repeat count of a counter, which is by how many times a counter value is used before it is incremented.

Command:

<ESC>CNT_GET_R**<FS>***name***<ETB>**

Field	Description
name	Name of the counter.

Successful response:

<ACK>*value***<ETB>**

Field	Description
value	The counter's repeat count.

Example:

To get the repeat count of the counter MYCOUNTER, send:

<ESC>CNT_GET_R**<FS>**MYCOUNTER**<ETB>**

Example response:

<ACK>1**<ETB>**

4.7 CNT_GET_RC

Description:

Get the current repeat iteration of a counter. The returned value is zero based, so if the repeat is set to *N*, the repeat iteration counts from zero to *N-1*.

If this command is used to set a non-global counter, the following rules apply:

- If the *INST* command has been executed with an empty list of Installations, the global repeat iteration of the counter will be returned.
- If the *INST* command was executed with a list of Installations, only the local repeat iteration of the first Installation will be returned.

Note: Use the command INST to select which installations to operate on.

Command:

<**ESC**>CNT_GET_RC<**FS**>*name*<**ETB**>

Field	Description
name	Name of the counter.

Successful response:

<**ACK**>*value*<**ETB**>

Field	Description
value	The counter's repeat iteration.

Example:

To get the current repeat iteration of the counter MYCOUNTER, send:

<**ESC**>CNT_GET_RC<**FS**>MYCOUNTER<**ETB**>

Example response:

<**ACK**>0<**ETB**>

4.8 CNT_LIST

Description:

List all available counters.

Command:

<**ESC**>CNT_LIST<**ETB**>

Successful response:

<**ACK**>name<**FS**>...<**ETB**>

Example:

To list all available counters:

<**ESC**>CNT_LIST<**ETB**>

Expected response:

<**ACK**>BATCH<**FS**>MYCOUNTER<**FS**>My Pretty Counter<**ETB**>

4.9 CNT_RESET

Description:

Reset the current value of a counter to the start value.

If this command is used to set a non-global counter, the following rules apply:

- If the *INST* command has been executed with an empty list of Installations, all instances of the counter will be set to the start value.
- If the *INST* command was executed with a list of Installations, only the local values for the listed Installations will be set to the start value.

Note: Use the command INST to select which installations to operate on.

Command:

<ESC>CNT_SET**<FS>***name***<ETB>**

Field	Description
name	Name of the counter.

Successful response:

<ACK>

Example:

To reset the value of the counter MYCOUNTER, send:

<ESC>CNT_RESET**<FS>**MYCOUNTER**<ETB>**

Expected response:

<ACK>

4.10 CNT_SET

Description:

Set the current value of a counter.

If this command is used to set a non-global counter, the following rules apply:

- If the *INST* command has been executed with an empty list of Installations, all instances of the counter will be set to the specified value.
- If the *INST* command was executed with a list of Installations, only the local values for the listed Installations will be set to the specified value.

Note: Use the command INST to select which installations to operate on.

Command:

<ESC>CNT_SET**<FS>***name***<FS>***value***<ETB>**

Field	Description
name	Name of the counter.
value	New value of the counter.

Successful response:

<ACK>

Example:

To set the value of the counter MYCOUNTER to 42, send:

<ESC>CNT_SET**<FS>**MYCOUNTER**<FS>**42**<ETB>**

Expected response:

<ACK>

4.11 CNT_SET_I

Description:

Set the increment of a counter, which is by how much it is incremented when it is incremented.

Command:

<ESC>CNT_SET_I**<FS>***name***<FS>***value***<ETB>**

Field	Description
name	Name of the counter.
value	New increment of the counter.

Successful response:

<ACK>

Example:

To set the increment of the counter MYCOUNTER to 42, send:

<ESC>CNT_SET_I**<FS>**MYCOUNTER**<FS>**42**<ETB>**

Expected response:

<ACK>

4.12 CNT_SET_R

Description:

Set the repeat count of a counter, which is by how many times a counter value is used before it is incremented.

Command:

<**ESC**>CNT_SET_R<**FS**>*name*<**FS**>*value*<**ETB**>

Field	Description
name	Name of the counter.
value	The counter's repeat count.

Successful response:

<**ACK**>

Example:

To set the repeat count of the counter MYCOUNTER to 10, send:

<**ESC**>CNT_SET_R<**FS**>MYCOUNTER<**FS**>10<**ETB**>

Expected response:

<**ACK**>

4.13 CNT_SET_RC

Description:

Set the current repeat iteration of a counter. The value is zero based, so if the repeat is set to *N*, the repeat iteration counts from zero to *N-1*.

If this command is used to set a non-global counter, the following rules apply:

- If the *INST* command has been executed with an empty list of Installations, the repeat iteration of all instances of the counter will be set to the specified value.
- If the *INST* command was executed with a list of Installations, only the repeat iteration for the listed Installations will be set to the specified value.

Note: Use the command INST to select which installations to operate on.

Command:

<**ESC**>CNT_SET_RC<**FS**>*name*<**FS**>*value*<**ETB**>

Field	Description
name	Name of the counter.
value	The counter's repeat iteration.

Successful response:

<**ACK**>

Example:

To set the current repeat iteration of the counter MYCOUNTER to 0, send:

<**ESC**>CNT_SET_RC<**FS**>MYCOUNTER<**FS**>0<**ETB**>

Expected response:

<**ACK**>

4.14 CNT_UPDATE

Description:

Causes the counter to update to next value

If this command is used to set a non-global counter, the following rules apply:

- If the *INST* command has been executed with an empty list of Installations, the value of all instances of the counter will be updated.
- If the *INST* command was executed with a list of Installations, only the listed Installations will be updated to a new value.

Note: Use the command INST to select which installations to operate on.

Command:

<**ESC**>CNT_UPDATE<**FS**>*name*<**ETB**>

Field	Description
name	Name of the counter.

Successful response:

<**ACK**>

Example:

To make the counter MYCOUNTER count up, send:

<**ESC**>CNT_UPDATE<**FS**>MYCOUNTER<**ETB**>

Expected response:

<**ACK**>

4.15 DB

Description:

Select the database table that following database commands will operate on.

Command:

<ESC>DB**<FS>***name***<ETB>**

Field	Description
name	Name of the database table. An empty name selects the default database table. If the name is not empty, and the database table does not exist, the command will fail.

Successful response:

<ACK>

Example:

To select the database table “My table”, send:

<ESC>DB**<FS>**My table**<ETB>**

Expected response:

<ACK>

To select the default database table, send:

<ESC>DB**<FS><ETB>**

Expected response:

<ACK>

4.16 DB_ADD_ROW

Description:

Add a row to the database table selected by the DB command.

Command:

<**ESC**>DB_ADD_ROW<**FS**>*columns*<**ETB**>

Field	Description
columns	Columns separated by <**GS**>. The command will fail if the number of columns to be added is greater than the number of columns in the table.

Successful response:

<**ACK**>

Example:

Assume table with the three columns “SKU”, “Name” and “Weight”. To add a new row to the table with SKU 42, name “Apples” and weight “10 kg”, send:

<**ESC**>DB_ADD_ROW<**FS**>42<**GS**>Apples<**GS**>10 kg<**ETB**>

Expected response:

<**ACK**>

4.17 DB_ADD_TABLE

Description:

Create a new database table.

Command:

<ESC>DB_ADD_TABLE**<FS>***name***<ETB>**

Field	Description
name	Name of the new database table. The command will fail if a table by this name already exist.

Successful response:

<ACK>

Example:

To create a new database table called “My table”, send:

<ESC>DB_ADD_TABLE**<FS>**My table**<ETB>**

Expected response:

<ACK>

4.18 DB_DEL_ROW

Description:

Delete rows from the database table selected by the DB command..

Command:

<ESC>DB_DEL_ROW**<FS>***selector***<ETB>**

Field	Description
selector	A list of column/value pairs, separated by **<GS>**. Each column/value pair is separated by **<RS>**.

All rows with a matching value will be deleted.

Successful response:

<ACK>

Example 1:

To delete all rows where the column “Name” is “Apples”, send:

<ESC>DB_DEL_ROW**<FS>**Name**<RS>**Apples**<ETB>**

Expected response:

<ACK>

Example 2:

To delete all rows where the column “Name” is “Apples” and the column “Origin” is “Italy”, send:

<ESC>DB_DEL_ROW**<FS>**Name**<RS>**Apples**<GS>**Origin**<RS>**Italy**<ETB>**

Expected response:

<ACK>

4.19 DB_DEL_TABLE

Description:

Delete a database table.

Command:

<ESC>DB_DEL_TABLE**<FS>***name***<ETB>**

Field	Description
name	Name of the database table to be deleted.

Successful response:

<ACK>

Example:

To delete a database table called “My table”, send:

<ESC>DB_DEL_TABLE**<FS>**My table**<ETB>**

Expected response:

<ACK>

4.20 DB_GET

Description:

Get values from the database table selected by the DB command..

Command:

<**ESC**>DB_GET<**FS**>*selector*<**FS**>*columns*<**ETB**>

Field	Description
selector	A list of column/value pairs, separated by <**GS**>. Each column/value pair is separated by <**RS**>.
columns	Columns for which to get the values, separated by <**GS**>.

Successful response:

<**ACK**>*columns*<**ETB**>

Field	Description
columns	List of values, separated by <**GS**>.

Example 1:

Given the following database table:

SKU	Name	Origin
13	Kiwi	Israel
42	Apples	Italy
88	Oranges	Morocco

To get the value of the column “Origin” from the first row where the column “Name” is “Apples”, send:

<**ESC**>DB_GET<**FS**>Name<**RS**>Apples<**FS**>Origin<**ETB**>

Expected response:

<**ACK**>Italy<**ETB**>

Example 2:

Given the database table from example 1, to get the values of the columns “SKU” and “Origin” from the first row where the column “Name” is “Apples”, send:

<**ESC**>DB_GET<**FS**>Name<**RS**>Apples<**FS**>SKU<**GS**>Origin<**ETB**>

Expected response:

<**ACK**>42<**GS**>Italy<**ETB**>

4.21 DB_INC

Description:

Increment a numeric value in a cell in the database table selected by the DB command..

Command:

1. <**ESC**>DB_INC<**FS**>*selector*<**FS**>*column*<**ETB**>
2. <**ESC**>DB_INC<**FS**>*selector*<**FS**>*column*<**FS**>*increment*<**ETB**>

Field	Description
selector	A list of column/value pairs, separated by <**GS**>. Each column/value pair is separated by <**RS**>.
column	Column to increment.
increment	Optional value to increment with. Equals to one if not supplied.

All rows with a matching value will be deleted.

Successful response:

<**ACK**>

Example 1:

Given the following database table:

SKU	Name	Count
13	Kiwi	10
42	Apples	7
88	Oranges	0

To increment the Count column by one in the rows where the column “Name” is “Apples”, send:

<**ESC**>DB_INC<**FS**>Name<**RS**>Apples<**FS**>Count<**ETB**>

This will change the Count from 7 to 8.

Example 2:

Given the same table as in Example 1, to subtract four from Count column in the rows where the column “SKU” is 13, send:

<**ESC**>DB_INC<**FS**>SKU<**RS**>13<**FS**>Count<**FS**>-4<**ETB**>

This will change the Count from 10 to 6.

4.22 DB_INIT_TABLE

Description:

Initialize the database table selected by the DB command.

Command:

1. **<ESC>**DB_INIT_TABLE**<FS>***columns***<ETB>**
2. **<ESC>**DB_INIT_TABLE**<FS>***columns***<*FS*>***defcol*<ETB>**

Field	Description
columns	Columns separated by **<GS>**.
defcol	Default column index, where zero refers to the first column.

Successful response:

<ACK>

Example 1:

To initialize the current database table to have three columns “SKU”, “Name” and “Weight”, send:

<ESC>DB_INIT_TABLE**<FS>**SKU**<GS>**Name**<GS>**Count**<ETB>**

Example 2:

To initialize the current database table to have three columns “SKU”, “Name” and “Weight”, and make “Name” the default column in the Controller's user interface, send:

<ESC>DB_INIT_TABLE**<FS>**SKU**<GS>**Name**<GS>**Count**<FS>**1**<ETB>**

4.23 DB_SET

Description:

Set the value in a cell in the database table selected by the DB command.

Command:

<**ESC**>DB_SET<**FS**>*selector*<**FS**>*values*<**ETB**>

Field	Description
selector	A list of column/value pairs, separated by <**GS**>. Each column/value pair is separated by <**RS**>.
values	A list of column/value pairs, separated by <**GS**>. Each column/value pair is separated by <**RS**>.

Successful response:

<**ACK**>

Example 1:

Given the following database table:

SKU	Name	Origin
13	Kiwi	Israel
42	Apples	Italy
88	Oranges	Marocco

To change the Origin of SKU 42 (Apples) to Sweden, send:

<**ESC**>DB_SET<**FS**>SKU<**RS**>42<**FS**>Origin<**RS**>Sweden<**ETB**>

Example 2:

Given the following database table:

SKU	Name	Origin	Color	Price
13	Kiwi	Israel	Green	100
42	Apples	Italy	Green	200
43	Apples	Norway	Red	220
88	Oranges	Morocco	Yellow	150

To change the Color of Norwegian apples to Green and its price to 210, send:

<ESC>DB_SET
<FS>Origin**<RS>**Norway**<GS>**Name**<RS>**Apples
<FS>Color**<RS>**Green**<FS>**Price**<RS>**210
<ETB>

Note: The above should be sent as a single line without extra spaces. It has been split into lines to make it more readable.

4.24 ENC_BLOCK

Description:

Blocks the encoder for all Installations selected using the INST command from detecting motion. Locking their position to the same value even if the physical encoder signals motion.
If a failure to lock any Installation's encoder was detected **NAK** will be returned.

This command can not be used for Installations using fixed speed.

Command:

<ESC>ENC_BLOCK<FS>*state***<ETB>**

Field	Description
state	True if the encoder should be blocked, false to unblock it and start detecting motion.

Successful response:

<ACK>

Example:

To block the encoder for Installation "a" from detecting motion:

<ESC>INST**<FS>**Production Line 1**<ETB>**

<ESC>ENC_BLOCK**<FS>**true**<ETB>**

Example response:

<ACK>

<ACK>

4.25 ENC_PPM

Description:

Sets the encoder resolution for all Installations selected using the INST command.
If a failure to set encoder resolution on any Installation's encoder was detected **NAK** will be returned.

This command can not be used for Installations using fixed speed.

Command:

<ESC>ENC_PPM<FS>*ppm***<ETB>**

Field	Description
ppm	Encoder resolution value

Successful response:

<ACK>

Example:

To set encoder resolution to 25010 for Installation "Production Line 1":

<ESC>INST**<FS>**Production Line 1**<ETB>**

<ESC>ENC_PPM**<FS>**25010**<ETB>**

Example response:

<ACK>

<ACK>

4.26 ENC_RESET_REVERSE

Description:

Resets the encoder's back filter for all Installations selected by the INST command, making the next forward motion detectable. If the call to reset failed for any Installation **NAK** will be returned.

This does not work when the Installation is configured for fixed speed.

Command:

<ESC>ENC_RESET_REVERSE**<ETB>**

Successful response:

<ACK>

Example:

To reset the back state for all encoders on Installations:

<ESC>INST**<ETB>**
<ESC>ENC_RESET_REVERSE**<ETB>**

Example response:

<ACK>

<ACK>

4.27 GOOD

Description:

Test if all is good about all Installations selected using the INST command, meaning that they do not require service and have no critical problems.

Command:

<ESC>GOOD**<ETB>**

Successful response:

<ACK>

Example:

To get the status of the selected Installation, send:

<ESC>GOOD**<ETB>**

Example response if all is well:

<ACK>

4.28 INPUT_GET

Description:

Get the current state of an input pin.

Command:

<**ESC**>INPUT_GET<**FS**>io<**ETB**>

Field	Description
io	Name of the I/O as defined in the External I/O configuration.

Successful response:

<**ACK**>*state*<**ETB**>

Examples:

To get the state of the input pin defined as I0:

<**ESC**>INPUT_GET<**FS**>I0<**ETB**>

Example response if all is well:

<**ACK**>0<**ETB**>

4.29 INST

Description:

Define the Installation or Installations on which other command operate.

Command:

1. **<ESC>**INST**<ETB>**
2. **<ESC>**INST**<FS>***installation*$_1$ ... **<FS>***installation*$_N$**<ETB>**

Field	Description
installation$_k$	Name or numeric ID of the Installation.

Successful response:

<ACK>

Examples:

To make subsequent command operate on all Installations:

<ESC>INST**<ETB>**

To make subsequent command operate only on Installation “Production Line 1”:

<ESC>INST**<FS>**Production Line 1**<ETB>**

To make subsequent command operate only on the Installations “Production Line 2” and “Production Line 3”:

<ESC>INST**<FS>**Production Line 2**<FS>**Production Line 3**<ETB>**

4.30 L_DECAP

Description:

Decap all L-series cartridges in all Installations selected using the INST command.

Command:

<ESC>L_DECAP**<FS>***count***<ETB>**

Field	Description
count	Number of times to decap. Must be less than 1024.

Successful response:

<ACK>

Example:

To decap 100 times of ink with all L-series cartridges in the selected Installations, send:

<ESC>L_DECAP**<FS>**100**<ETB>**

Expected response:

<ACK>

4.31 L_DECAP_MARKER

Description:

Decap all L-series cartridges in the specified Marker in all Installations selected using the INST command.

Command:

<**ESC**>L_DECAP_MARKER<**FS**>*marker*<**FS**>*count*<**ETB**>

Field	Description
marker	Name of the Marker.
count	Number of times to decap. Must be less than 1024.

Successful response:

<**ACK**>

Example:

To decap 100 times of ink with all L-series cartridges in the Marker “Operator Side” in the selected Installations, send:

<**ESC**>L_DECAP_MARKER<**FS**>Operator Side<**FS**>100<**ETB**>

Expected response:

<**ACK**>

4.32 L_DECAP_TABLE

Description:

Set the decap configuration table for a L-series ink and reconfigure all Installations selected using the INST command.

Note: Use the command INST to select which installations the command shall have immediate effect on.

Command:

<**ESC**>L_DECAP_TABLE<**FS**>*ink*<**FS**>*configuration*<**ETB**>

Field	Description
ink	Name of the ink.
configuration	A list of time/count pairs, separated by <**GS**>. Each time/count pair is separated by <**RS**>.

Time is in seconds since the last printed mark and count is the number of times to decap. Count must be less than 1024.

Successful response:

<**ACK**>

Example:

To set the decap configuration table for ink LS-6020 so that it will decap 30 times after 10 seconds and 110 times after 200 seconds, send:

<**ESC**>L_DECAP_TABLE<**FS**>LS-6020<**FS**>10<**RS**>30<**GS**>200<**RS**>110<**ETB**>

Expected response:

<**ACK**>

4.33 LOG

Description:

Create a log entry in the MPERIA log.

Command:

<ESC>LOG**<FS>***severity***<FS>***text***<ETB>**

<ESC>LOG**<FS>***severity***<FS>***text***<FS>***info***<ETB>**

Field	Description
severity	Severity. Must be one of NONE, DEBUG, INFO, SERVICE, CRITICAL.
text	Text to add to the log. Must be a single line.
info	Optional extra information. Can be multiple lines.

Successful response:

<ACK>

Example:

To add a service log message, send:

<ESC>LOG**<FS>**SERVICE**<FS>**Replace whozz-o-matic now!**<ETB>**

Expected response:

<ACK>

4.34 MSG_DEL

Description:

Delete message from the message database. If the message can not be found **NAK** is returned.

Command:

1. **<ESC>**MSG_DEL**<FS>***message***<ETB>**

Field	Description
message	Message name.

Successful response:

<ACK>

Example 1:

To delete the message “My Message”, send:

<ESC>MSG_DEL**<FS>**My Message**<ETB>**

Expected response:

<ACK>

4.35 MSG_LIST

Description:

List all messages from the message database.

Command:

1. **<ESC>**MSG_LIST**<ETB>**

Successful response:

<ACK>

Example 1:

To delete the message “My Message”, send:

<ESC>MSG_LIST**<ETB>**

Expected response:

<ACK>

4.36 MARKER_ENABLE

Description:

Enable of specified Marker in all Installations selected using the INST command.

Command:

<ESC>MARKER_ENABLE**<FS>***marker***<ETB>**

Field	Description
marker	Name of the Marker.

Successful response:

<ACK>

Example:

To enable the Marker “Operator Side” in the selected Installations, send:

<ESC>MARKER_ENABLE**<FS>**Operator Side**<ETB>**

Expected response:

<ACK>

4.37 MARKER_DISABLE

Description:

Disable the specified Marker in all Installations selected using the INST command.

Command:

<ESC>MARKER_ENABLE**<FS>***marker***<ETB>**

Field	Description
marker	Name of the Marker.

Successful response:

<ACK>

Example:

To disable the Marker “Operator Side” in the selected Installations, send:

<ESC>MARKER_DISABLE**<FS>**Operator Side**<ETB>**

Expected response:

<ACK>

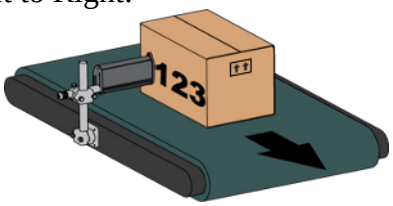
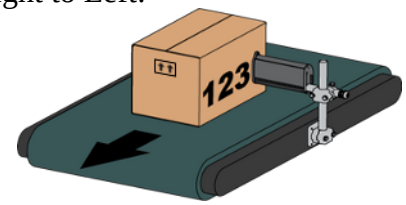
4.38 MARKER_DIR

Description:

Set the print direction of the specified Marker in all Installations selected using the INST command.

Command:

<**ESC**>MARKER_DIR<**FS**>*marker*<**FS**>*direction*<**ETB**>

Field	Description
marker	Name of the Marker.
direction	Print direction: • 0 = Left to Right:  • 1 = Right to Left: 

Successful response:

<**ACK**>

Example:

To set the print direction of the Marker “Operator Side” in the selected Installations to Left-to-Right, send:

<**ESC**>MARKER_DIR<**FS**>Operator Side<**FS**>1<**ETB**>

Expected response:

<**ACK**>

4.39 MARKER_MARGIN

Description:

Set the margin of the specified Marker in all Installations selected using the INST command.

Command:

<ESC>MARKER_MARGIN**<FS>***marker***<FS>***margin***<ETB>**

Field	Description
marker	Name of the Marker.
direction	Margin in millimeters.

Successful response:

<ACK>

Example:

To set the margin of the Marker “Operator Side” in the selected Installations to 25.4 mm, send:

<ESC>MARKER_MARGIN**<FS>**Operator Side**<FS>**25.4**<ETB>**

Expected response:

<ACK>

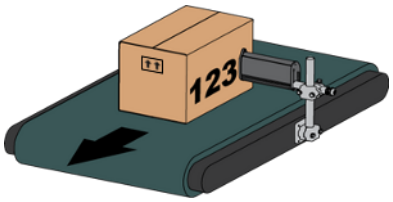
4.40 MARKER_UPSIDEDOWN

Description:

Set whether the specified Marker in all Installations selected using the INST command shall print upside down or not.

Command:

<**ESC**>MARKER_UPSIDEDOWN<**FS**>*marker*<**FS**>*upsidedown*<**ETB**>

Field	Description
marker	Name of the Marker.
upsidedown	Print upside down: • 0 = Normal:  • 1 = Upside down:

Successful response:

<**ACK**>

Example:

To make the Marker “Operator Side” in the selected Installations print upside down, send:

<**ESC**>MARKER_MARGIN<**FS**>Operator Side<**FS**>1<**ETB**>

Expected response:

<**ACK**>

4.41 MARK_GAP

Description:

Set the gap between repeated marks for the specified Marker in all Installations selected using the INST command.

Command:

<ESC>MARK_GAP**<FS>***marker***<FS>***gap***<ETB>**

Field	Description
marker	Name of the Marker.
gap	Mark gap in millimeters.

Successful response:

<ACK>

Example:

To set the mark gap for the Marker “Operator Side” in the selected Installations to 25.4 mm, send:

<ESC>MARK_GAP**<FS>**Operator Side**<FS>**25.4**<ETB>**

Expected response:

<ACK>

4.42 MARK_INTERVAL

Description:

Set the interval between repeated marks for the specified Marker in all Installations selected using the INST command.

Command:

<ESC>MARK_INTERVAL**<FS>***marker***<FS>***interval***<ETB>**

Field	Description
marker	Name of the Marker.
interval	Mark interval in millimeters.

Successful response:

<ACK>

Example:

To set the mark interval for the Marker “Operator Side” in the selected Installations to 25.4 mm, send:

<ESC>MARK_INTERVAL**<FS>**Operator Side**<FS>**25.4**<ETB>**

Expected response:

<ACK>

4.43 MARK_REPEAT

Description:

Set the number of times to repeat a mark for the specified Marker in all Installations selected using the INST command.

Command:

<ESC>MARK_REPEAT**<FS>***marker***<FS>***repeat***<ETB>**

Field	Description
marker	Name of the Marker.
repeat	Number of times that marks are repeated. Zero means the mark is printed once. A negative number means the mark is printed continuously.

Successful response:

<ACK>

Example:

To set the mark repeat for the Marker “Operator Side” in the selected Installations to 5 times, send:

<ESC>MARK_REPEAT**<FS>**Operator Side**<FS>**5**<ETB>**

Expected response:

<ACK>

4.44 NONCRITICAL

Description:

Test if all Installations selected using the INST command do not have critical problems preventing them from printing.

Command:

<ESC>NONCRITICAL**<ETB>**

Successful response:

<ACK>

Example:

To test if there are no critical errors in the selected Installations, send:

<ESC>NONCRITICAL**<ETB>**

Example response if all is well:

<ACK>

4.45 OFFLINE

Description:

Set all Installations selected by the INST command in offline mode.

Command:

<ESC>OFFLINE**<ETB>**

Successful response:

<ACK>

Example:

To set all Installations in offline mode, send:

<ESC>INST**<ETB>**
<ESC>OFFLINE**<ETB>**

Example response if all is well:

<ACK>
<ACK>

To set Installations “First Installation” and “Second Installation” in offline mode, send:

<ESC>INST**<FS>**First Installation**<FS>**Second Installation**<ETB>**
<ESC>OFFLINE**<ETB>**

Example response if all is well:

<ACK>
<ACK>

4.46 ONLINE

Description:

Set all Installations selected by the INST command in online mode.

Command:

<ESC>ONLINE**<ETB>**

Successful response:

<ACK>

Example:

To set all Installations in online mode, send:

<ESC>INST**<ETB>**
<ESC>ONLINE**<ETB>**

Example response if all is well:

<ACK>
<ACK>

To set Installations “First Installation” and “Second Installation” in online mode, send:

<ESC>INST**<FS>**First Installation**<FS>**Second Installation**<ETB>**
<ESC>ONLINE**<ETB>**

Example response if all is well:

<ACK>
<ACK>

4.47 OUTPUT_SET

Description:

Set the value of an external I/O.

Command:

<**ESC**>OUTPUT_SET<**FS**>io<**FS**>state<**FS**>delay<**FS**>onTime<**FS**>offTime<**FS**>numTransitions<**ETB**>

Field	Description
io	I/O pin configuration name specified under the External I/O interface.
state	Initial state to set.
delay	Time before the state specified by the state attribute gets set the first time.
onTime	Time spent in positive state specified in ms.
offTime	Time spent in negative state specified in ms.
numTransitions	Number of state transitions after the initial value has been set.

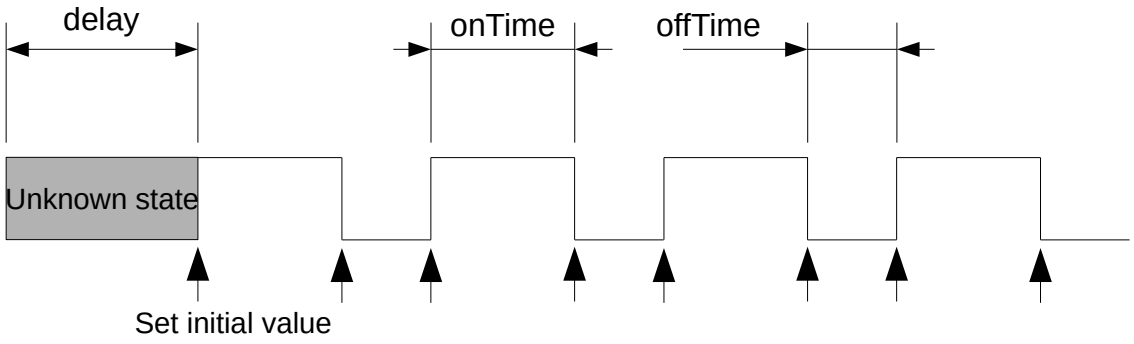
Successful response:

<**ACK**>

Example:

The request below for a pin with an negative initial state will give a result described by the following picture. Transitions are marked as arrows at the bottom of the picture.

<**ESC**>OUTPUT_SET<**FS**>O0<**FS**>1<**FS**>1200<**FS**>1000<**FS**>600<**FS**>7<**ETB**>



4.48 PEEKMSG

Description:

Get the name of the currently selected message on the Installation selected using the INST command. This command fails if more than one Installation is selected.

Command:

<ESC>PEEKMSG**<ETB>**

Successful response:

<ACK>*name***<ETB>**

Field	Description
name	Name of the currently selected message, or an empty value if no message is selected.

Example:

To get the name of the currently selected message, send:

<ESC>PEEKMSG**<ETB>**

Example response:

<ACK>My Message**<ETB>**

4.49 PLUGIN_ACTION

Description:

Run the *plugin_action* entry point in the specified Lua plugin. See the Lua plugin manual for more information.

Command:

1. **<ESC>**PLUGIN_ACTION**<FS>***plugin name***<ETB>**
2. **<ESC>**PLUGIN_ACTION**<FS>***plugin name***<FS>***input***<ETB>**
3. **<ESC>**PLUGIN_ACTION**<FS>***plugin name***<FS>***input***<FS>***config***<ETB>**

Field	Description
plugin name	The name of the plugin.
input	Input to the plugin_action function.
config	Configuration to the plugin_action function.

Successful response:

<ACK>*response***<ETB>**

Field	Description
response	The returned string value from the plugin function.

Example:

To call example_plugin.lua, send:

<ESC>PLUGIN_ACTION<**FS**>example_plugin.lua<**FS**>Hello World!**<ETB>**

Example response:

<ACK>

4.50 PRINT_CNT

Description:

Get the print count for all messages, or the for a specific message, on all Installations selected using the INST command. Note that if the INST command is used without arguments to select all Installation, the print count will also include Installations that are no longer available.

Command:

1. **<ESC>**PRINT_CNT**<ETB>**
2. **<ESC>**PRINT_CNT**<FS>***message***<ETB>**

Successful response:

<ACK>*count***<ETB>**

Example:

To get the print count of all messages, send:

<ESC>PRINT_CNT**<ETB>**

Example response:

<ACK>56723**<ETB>**

To get the total print count for the message “My Message”, send:

<ESC>PRINT_CNT**<FS>**MyMessage**<ETB>**

Example response:

<ACK>42**<ETB>**

4.51 PRINT_CNT_RESET

Description:

Reset the print count for all messages, or the for a specific message, on all Installations selected using the INST command. Note that if the INST command is used without arguments to select all Installation, the print count will also be reset for Installations that are no longer available.

Command:

1. **<ESC>**PRINT_CNT_RESET**<ETB>**
2. **<ESC>**PRINT_CNT_RESET**<FS>***message***<ETB>**

Successful response:

<ACK>

Example:

To reset the total print count of all messages, send:

<ESC>PRINT_CNT_RESET**<ETB>**

Example response:

<ACK>

To reset the total print count for the message “My Message”, send:

<ESC>PRINT_CNT_RESET**<FS>**MyMessage**<ETB>**

Example response:

<ACK>

4.52 QUEUE

Description:

Fixate and queue the selected message on all Installations selected using the INST command. The installations must have Automatic Message Fixation disabled for the QUEUE command to succeed.

Command:

<**ESC**>QUEUE<**ETB**>

Successful response:

<**ACK**>

Example:

To fixate and queue the selected message on all Installations, send:

<**ESC**>INST<**ETB**>
<**ESC**>QUEUE<**ETB**>

Example response:

<**ACK**>
<**ACK**>

4.53 QUEUE_LEN

Description:

Get the queue length on the Installation selected using the INST command. The command fails if more than one Installation is selected.

Command:

<ESC>QUEUE_LEN**<ETB>**

Successful response:

<ACK>*length***<ETB>**

Field	Description
length	The current queue length.

Example:

To get the version of the MPERIA Controller, send:

<ESC>QUEUE_LEN**<ETB>**

Example response:

<ACK>5**<ETB>**

4.54 QUEUE_RESET

Description:

Reset the queue on all Installations selected using the INST command. The installations must have Automatic Message Fixation disabled for the QUEUE command to succeed.

Command:

<ESC>QUEUE_RESET**<ETB>**

Successful response:

<ACK>

Example:

To reset the queue on all Installations, send:

<ESC>INST**<ETB>**
<ESC>QUEUE_RESET**<ETB>**

Example response:

<ACK>
<ACK>

4.55 SEL

Description:

Select message and database row for printing on all Installations selected using the INST command. The database column and value must exist in each Installation's configured database table. If the message cannot be selected for an Installation, or the database cell cannot be found, that currently selected message is deselected, the database row becomes invalid, and **NAK** in returned.

Command:

1. **<ESC>**SEL**<FS>***message*<***FS***>*DB column*<***FS***>*DB value***<ETB>**
2. **<ESC>**SEL**<FS>***message*<***FS***>*DB column*<***FS***>*DB value*<***FS***>*max print***<ETB>**

Field	Description
message	Message name. An empty value deselects the message.
DB column	The database column to search in.
DB value	The database value to search for.
max print	Optional maximum number of times the message can be printed without being deselected. A negative value (which is the default) means that the message will never be automatically deselected.

Successful response:

<ACK>

Example:

To select the message “My Message” and the database row where the column “Origin” is “India”, send:

<ESC>SEL**<FS>**My Message**<FS>**Origin**<FS>**India**<ETB>**

Expected response:

<ACK>

4.56 SELDB

Description:

Select database row for printing on all Installations selected using the INST command. The database column and value must exist in each Installation's configured database table. If the database cell cannot be found, the database row becomes invalid, and **NAK** in returned.

Command:

1. **<ESC>**SELDB**<FS>***DB column***<FS>***DB value***<ETB>**
2. **<ESC>**SELDB**<FS>***DB column***<FS>***DB value***<FS>***max print***<ETB>**

Field	Description
DB column	The database column to search in.
DB value	The database value to search for.
max print	Optional maximum number of times the message can be printed without being deselected. A negative value (which is the default) means that the message will never be automatically deselected.

Successful response:

<ACK>

Example:

To select the database row where the column “Origin” is “India”, send:

<ESC>SELDB**<FS>**Origin**<FS>**India**<ETB>**

Expected response:

<ACK>

4.57 SELMSG

Description:

Select message for printing on all Installations selected using the INST command. If the message cannot be selected for an Installation, that currently selected message is deselected, and **NAK** in returned.

Command:

1. **<ESC>**SELMSG**<FS>***message***<ETB>**
2. **<ESC>**SELMSG**<FS>***message***<*FS*>***max print*<ETB>**

Field	Description
message	Message name. An empty value deselects the message.
max print	Optional maximum number of times the message can be printed without being deselected. A negative value (which is the default) means that the message will never be automatically deselected.

Successful response:

<ACK>

Example 1:

To select the message “My Message”, send:

<ESC>SELMSG**<FS>**My Message**<ETB>**

Expected response:

<ACK>

Example 2:

To select the message “My Message” to be printed only once, send:

<ESC>SELMSG**<FS>**My Message**<FS>**1**<ETB>**

Expected response:

<ACK>

4.58 SET_CLOCK

Description:

Set the system clock

Command:

<**ESC**>SET_CLOCK<**FS**>datetime<**ETB**>

Successful response:

<**ACK**>

Field	Description
datetime	A string with 12 or 14 digits of date and time in the format: yyyymmddHHMMss ss is optional, and will be set to 0 if omitted

Example:

To set sytem time, send:

<**ESC**>SET_CLOCK<**FS**>201411260838<**ETB**>

Example response:

<**ACK**>

4.59 SET_SPEED

Description:

Set the fixed speed of all Installations selected by the INST command. Note that the installations must already be configured to use fixed speed for this command to succeed. This command will not change the configuration from using an encoder to using fixed speed.

Command:

<**ESC**>SET_SPEED<**FS**>*speed*<**ETB**>

Field	Description
speed	Fixed speed in meters per second as a floating point number using period (.) as decimal separator.

Successful response:

<**ACK**>

Example:

To set the fixed speed to 1.42 m/s, send:

<**ESC**>SET_SPEED<**FS**>1.42<**ETB**>

Expected response:

<**ACK**>

4.60 START

Description:

Start all Installations selected by the INST command.

Command:

<ESC>START**<ETB>**

Successful response:

<ACK>

Example:

To start all Installations, send:

<ESC>INST**<ETB>**
<ESC>START**<ETB>**

Example response if all is well:

<ACK>
<ACK>

To start Installations “First Installation” and “Second Installation”, send:

<ESC>INST**<FS>**First Installation**<FS>**Second Installation**<ETB>**
<ESC>START**<ETB>**

Example response if all is well:

<ACK>
<ACK>

4.61 STATE

Description:

Get the current state of a single Installation selected using the INST command.

Command:

<ESC>STATE**<ETB>**

Successful response:

<ACK>*state***<ETB>**

The following states are possible:

Status	Description
OFFLINE	The Installation is stopped.
INCOMPLETE	The Installation is missing PTMs.
STARTING	The Installation is starting up.
STARTED	The Installation is started.
PAUSED	The Installation is stopped.
STOPPING	The Installation is stopping.
RESTART_DELAY	The Installation failed, and will be restarted after a delay.
UNLICENSED	The Installation cannnot be started because of a license problem.

Example:

To get the state of the selected Installation, send:

<ESC>STATE**<ETB>**

Example response:

<ACK>STARTED**<ETB>**

4.62 STATUS

Description:

Get the status of all Installations selected using the INST command.

Command:

<ESC>STATUS**<ETB>**

Successful response:

1. **<ACK><ETB>**
2. **<ACK>***status*$_1$ ... **<FS>***status*$_N$**<ETB>**

Field	Description
status$_k$	Status descriptor.

The first form is returned if all is good. The second form returns a list of status strings. The following status items are possible:

Status	Description
missing	PTM is missing
updating	PTM is being updated
broken	PTM communication error
brokenMsg	Message Error
stolen	PTM used by other Controller
inkLow	Low ink level
inkEmpty	Out of ink
tempLow	Low temperature
crtUnknown	Unknown cartridge
crtMixed	Mixed cartridges
crtUnauthorized	Cartridge not permitted
crtUnsupported	Unsupported cartridge
bulkUnauthorized	Bulk ink is not permitted
crtMissing	No cartridge
crtPowerProhibited	Power disabled due to issues with another cartridge in the same device.
crtBroken	Broken cartridge
crtOvertemp	Overheating cartridge
crtOvercurrent	Cartridge is consuming too much current.
crtAlmostWornOut	Bulk cartridge almost worn out.

crtOther	Other cartridge error
inkLicenseWarning	Ink license warning
inkLicenseExpireWarning	Ink license expire warning
inkLicenseExpired	Ink license expired
inkLicenseError	Ink license error
ptmLicenseError	License for Number of PTMs has been exceeded
crtLicenseError	License for number of cartridges has been exceeded
techLicenseError	The license for number of technologies has been exceeded
dodCleaner	The marker is using cleaner instead of ink
wrongState	Wrong state

The following status items are specific to Autolabel markers.

Status	**Description**
paper_low	Paper low detected
low_air_pressure	Air pressure below set limit
ribbon_low	Ribbon low detected
rewinder_almost_full	The rewinder for backing paper is almost full
trig_outside_window	The trigger was activated outside the print window
trig_during_print	The trigger was activated during print
trig_without_data	The trigger was activated but no dynamic data was available
trig_open_head	The trigger was activated with the print head open
trig_righter_not_ready	The trigger was activated but the righter was not ready
trig_applicator_not_ready	The trigger was activated but the applicator was not ready
missing_trigger	Missing apply trig within the specified time after print trig
paper_out	Paper out detected
no_air_pressure	The air pressure is below the set limit
printhead_open	The print head is open
rewinder_full	The rewinder for backing paper is full
layout_not_found	The selected layout is not found in the internal database
rendering_error	There was an error when rendering the layout
ribbon_out	Out of thermal transfer ribbon
applicator_error	An error was detected when applying the label
applicator_stopped	Pallet applicator stop button is active
label_error	Label stuck/label lost detected when applying the label
adjusting_media	Adjusting label media after closing the print head

Example:

To get the status of the selected Installation, send:

<ESC>STATUS**<ETB>**

Example response if all is OK:

<ACK><ETB>

Example response if the ink level is low and a cartridge is missing:

<ACK>inkLow**<FS>**crtMissing**<ETB>**

4.63 STOP

Description:

Stop all Installations selected by the INST command.

Command:

<ESC>STOP**<ETB>**

Successful response:

<ACK>

Example:

To stop all Installations, send:

<ESC>INST**<ETB>**
<ESC>STOP**<ETB>**

Example response if all is well:

<ACK>
<ACK>

To stop Installations “First Installation” and “Second Installation”, send:

<ESC>INST**<FS>**First Installation**<FS>**Second Installation**<ETB>**
<ESC>STOP**<ETB>**

Example response if all is well:

<ACK>
<ACK>

4.64 T_CLEAN

Description:

Clean all T-series print heads in all Installations selected using the INST command.

Command:

<ESC>T_CLEAN**<FS>***duration***<ETB>**

Field	Description
duration	Optional duration in milliseconds. The default value is 300 ms.

Successful response:

<ACK>

Example:

To clean all T-series print heads in the installation “Production Line 1”, send:

<ESC>INST**<FS>**Production Line 1**<ETB>**
<ESC>T_CLEAN**<ETB>**

Expected response:

<ACK>
<ACK>

4.65 T_CLEAN_MARKER

Description:

Clean all T-series print heads in the specified Marker in all Installations selected using the INST command.

Command:

<ESC>T_CLEAN**<FS>***name***<FS>***duration***<ETB>**

Field	Description
name	Marker name.
duration	Optional duration in milliseconds. The default value is 300 ms.

Successful response:

<ACK>

Example:

To clean all T-series print heads in the marker “Operator Side” in Installation “Production Line 1”, send:

<ESC>INST**<FS>**Production Line 1**<ETB>**
<ESC>T_CLEAN_MARKER**<FS>**Operator Side**<ETB>**

Expected response:

<ACK>
<ACK>

4.66 TRIG

Description:

Soft trigger all Installations selected using the INST command.

Command:

<ESC>TRIG**<FS>***activate***<ETB>**

Field	Description
activate	Optional bool value indicating if the trigger shall be activated or deactivated. If true, the trigger is activated, if false, the trigger is deactivated. If not present the trigger will first be activated and then immediately deactivated.

Successful response:

<ACK>

Example:

To soft trigger all installations, send:

<ESC>INST**<ETB>**
<ESC>TRIG**<ETB>**

Example response:

<ACK>
<ACK>

4.67 TRIG_ACTIVE

Description:

Get the current trigger state of the first Installation selected using the INST command.

Command:

<ESC>TRIG_ACTIVE**<ETB>**

Successful response:

<ACK>*state***<ETB>**

Examples:

To get the state of the trigger:

<ESC>TRIG_ACTIVE**<ETB>**

Example response if the trigger is active:

<ACK>1**<ETB>**

4.68 USR_LVL_GET

Description:

Get the current user level.

Command:

<ESC>USR_LVL_GET**<ETB>**

Successful response:

<ACK>*userlevel***<ETB>**

Field	Description
userlevel	Name of the user level.

4.69 USR_LVL_SET

Description:

Limits the user's access to the UI to the specified level

Command:

<ESC>USR_LVL_SET**<FS>**userlevel**<ETB>**

Field	Description
userlevel	Name of the user level or 0 as a fallback to grant full access.

Successful response:

<ACK>

Example:

To change user level, send:

<ESC>USR_LVL_SET**<FS>**Operator**<ETB>**

Example response:

<ACK>

4.70 V_FLUSH

Description:

Flush all V-series print heads in all Installations selected using the INST command.

Command:

1. **<ESC>**V_FLUSH**<ETB>**
2. **<ESC>**V_FLUSH**<FS>***columns***<ETB>**
3. **<ESC>**V_FLUSH**<FS>***columns***<FS>***dotsize***<ETB>**

Field	Description
columns	Optional number of columns. Note that a too large number of columns can overheat and permanently damage the print head.
dotsize	Optional dotsize. Note that a too large dotsize can overheat and permanently

Successful response:

<ACK>

Example:

To flush all V-series print heads in the installation “Production Line 1”, send:

<ESC>INST**<FS>**Production Line 1**<ETB>**
<ESC>V_FLUSH**<ETB>**

Expected response:

<ACK>
<ACK>
<ACK>

4.71 V_FLUSH_MARKER

Description:

Flush all V-series print heads in the specified Marker in all Installations selected using the INST command.

Command:

1. **<ESC>**V_FLUSH_MARKER**<FS>***name***<ETB>**
2. **<ESC>**V_FLUSH_MARKER**<FS>***name***<FS>***columns***<ETB>**
3. **<ESC>**V_FLUSH_MARKER**<FS>***name***<FS>***columns***<FS>***dotsize***<ETB>**

Field	Description
name	Marker name.
columns	Optional number of columns. Note that a too large number of columns can overheat and permanently damage the print head.
dotsize	Optional dotsize. Note that a too large dotsize can overheat and permanently

Successful response:

<ACK>

Example:

To flush all V-series print heads in the marker “Operator Side” in Installation “Production Line 1”, send:

<ESC>INST**<FS>**Production Line 1**<ETB>**
<ESC>V_FLUSH_MARKER**<FS>**Operator Side**<ETB>**

Expected response:

<ACK>
<ACK>

4.72 V_SET_CLEANER

Description:

Enable cleaner for all V-series print heads in all Installations selected using the INST command.

Command:

<**ESC**>V_SET_CLEANER<**FS**>*enable*<**ETB**>

Field	Description
enable	1 to enable cleaner, 0 to enable ink.

Successful response:

<**ACK**>

Example:

To enable cleaner for all V-series print heads in the installation “Production Line 1”, send:

<**ESC**>INST<**FS**>Production Line 1<**ETB**>
<**ESC**>V_SET_CLEANER<**FS**>1<**ETB**>

Expected response:

<**ACK**>
<**ACK**>

4.73 V_SET_CLEANER_MARKER

Description:

Enable cleaner for all V-series print heads in the specified Marker in all Installations selected using the INST command.

Command:

<ESC>V_SET_CLEANER_MARKER**<FS>***name***<FS>***enable***<ETB>**

Field	Description
name	Marker name.
enable	1 to enable cleaner, 0 to enable ink.

Successful response:

<ACK>

Example:

To enable cleaner for all V-series print heads in the marker “Operator Side” in Installation “Production Line 1”, send:

<ESC>INST**<FS>**Production Line 1**<ETB>**
<ESC>V_SET_CLEANER_MARKER**<FS>**Operator Side**<FS>**1**<ETB>**

Expected response:

<ACK>
<ACK>

4.74 V_SET_DROP_TOF

Description:

Set drop time of flight for a V-series specified Marker in all Installations selected using the INST command.

Command:

<**ESC**>V_SET_DROP_TOF<**FS**>*name*<**FS**>*tof*<**ETB**>

Field	Description
name	Marker name.
tof	Drop time of flight in milliseconds as a floating point number using period (.) as decimal separator.

Successful response:

<**ACK**>

Example:

To set drop time of flight to 2.25 milliseconds for the marker “Operator Side” in Installation “Production Line 1”, send:

<**ESC**>INST<**FS**>Production Line 1<**ETB**>
<**ESC**>V_SET_DROP_TOF<**FS**>Operator Side<**FS**>2.25<**ETB**>

Expected response:

<**ACK**>
<**ACK**>

4.75 VAR_DEL

Description:

Delete a variable. Variables can only be deleted if they are not in use. Attempting to delete a variable that is in used causes a **<NAK>** to be returned.

Command:

<ESC>VAR_DEL**<FS>***name***<ETB>**

Field	Description
name	Name of the variable.

Successful response:

<ACK>

Example:

To delete the variable PRODUCT, send:

<ESC>VAR_DEL**<FS>**PRODUCT**<ETB>**

Example response:

<ACK>

4.76 VAR_GET

Description:

Get the value of a variable.

If this command is used to set a non-global variable, the following rules apply:

- If the *INST* command has been executed with an empty list of Installations, the global value of the variable will be returned.
- If the *INST* command was executed with a list of Installations, only the local value of the first Installation will be returned.

Note: Use the command INST to select which installations to operate on.

Command:

<ESC>VAR_GET**<FS>***name***<ETB>**

Field	Description
name	Name of the variable.

Successful response:

<ACK>*value***<ETB>**

Field	Description
value	Value of the variable.

Example:

To get the value of the variable PRODUCT, send:

<ESC>VAR_GET**<FS>**PRODUCT**<ETB>**

Example response:

<ACK>Bananas**<ETB>**

4.77 VAR_LIST

Description:

List all available variables.

Command:

<**ESC**>VAR_LIST<**ETB**>

Successful response:

<**ACK**>name<**FS**>...<**ETB**>

Example:

To list all available variables:

<**ESC**>VAR_LIST<**ETB**>

Expected response:

<**ACK**>BATCH<**FS**>OPERATOR<**FS**>My Pretty Variable<**ETB**>

4.78 VAR_SET

Description:

Set the value of a variable. If the variable does not exist, a new variable is created.

If this command is used to set a non-global variable, the following rules apply:

- If the *INST* command has been executed with an empty list of Installations, all instances of the variable will be set to the specified value.
- If the *INST* command was executed with a list of Installations, only the local values for the listed Installations will be set to the specified value.

Note: Use the command INST to select which installations to operate on.

Command:

<ESC>VAR_SET**<FS>***name***<FS>***value***<ETB>**

Field	Description
name	Name of the variable.
value	New value of the variable.

Successful response:

<ACK>

Example:

To set the value of the variable PRODUCT to Bananas, send:

<ESC>VAR_SET**<FS>**PRODUCT**<FS>**Bananas**<ETB>**

Expected response:

<ACK>

4.79 VAR_SET2

Description:

Set the values of one or more variables. If the variables does not exist, new variables are created.

If this command is used to set non-global variables, the following rules apply:

- If the *INST* command has been executed with an empty list of Installations, all instances of the variable will be set to the specified value.
- If the *INST* command was executed with a list of Installations, only the local values for the listed Installations will be set to the specified value.

Note: Use the command INST to select which installations to operate on.

Command:

<ESC>VAR_SET2**<FS>***name*$_1$**<GS>***value*$_1$ ... **<*FS*>***name*$_N$**<*GS*>***value*$_N$**<ETB>**

Field	Description
$name_k$	Name of the kth variable.
$value_k$	New value of the kth variable.

Successful response:

<ACK>

Example:

To set the value of the variable PRODUCT to Bananas, send:

<ESC>VAR_SET2**<FS>**PRODUCT**<GS>**Bananas**<ETB>**

Expected response:

<ACK>

To set the value of the variable PRODUCT to Oak, WEIGHT to 10, and SIZE to Large, send:

<ESC>VAR_SET2**<FS>**PRODUCT**<GS>**Oak**<FS>**WEIGHT**<GS>**10**<FS>**SIZE**<GS>**Large**<ETB>**

Expected response:

<ACK>

4.80 VERSION

Description:

Get the version of the MPERIA Controller.

Command:

<ESC>VERSION**<ETB>**

Successful response:

<ACK>*version***<ETB>**

Field	Description
version	MPERIA Controller version.

Example:

To get the version of the MPERIA Controller, send:

<ESC>VERSION**<ETB>**

Example response:

<ACK>2.5.0**<ETB>**

5 Revision history

Revision	Date	Changes
00	2013-12-10	First revision. For 2.5.x.
01	2014-05-16	Updates for 3.0.x Added commands: L_DECAP_TABLE, SET_SPEED, CNT_RESET, MARKER_MARGIN, V_FLUSH, V_FLUSH_MARKER Updated description of fault handling for SEL, SELDB and SELMSG.
02	2014-06-02	Add optional activate argument to trigger command.
03	2014-08-28	Updates for 3.2.x Added commands: PRINT_CNT, PRINT_CNT_RESET, V_SET_CLEANER, V_SET_CLEANER_MARKER.
04	2014-10-23	Updates for 3.3.x Added commands: MARKER_UPSIDEDOWN Corrected MARKER_DIR arguments.
05	2015-01-26	Updates for 4.0.x Added commands: MARKER_UPSIDEDOWN, SET_CLOCK.
06	2015-xx-xx	Updates for 5.0.x Added commands: MARK_GAP, MARK_INTERVAL, MSG_DEL, MSG_LIST
07	2015-06-05	Updates for 5.1.x Added Autolabel specific status codes.
08	2015-10-07	Updated for 5.2.x Added DEBUG level to the LOG command.
09	2015-10-14	Updated for 5.3.x Added commands: VAR_SET2.
10	2016-03-03	Added missing status codes: *crtOvertemp*, *crtAlmostWornOut*, *techLicenseError*, *dodCleaner*.
11	2016-05-20	Updated for 5.3.x Added commands: CNT_UPDATE. Fixed documentation of commands: NONCRITICAL.
12	2016-11-09	Updated for 6.0.x Added command: INPUT_GET, OUTPUT_SET, TRIG_ACTIVE.
13	2016-12-23	Updated for 6.1.x Added commands: ENC_BLOCK, ENC_RESET_REVERSE, OFFLINE, ONLINE

Revision	Date	Changes
14	2017-06-26	Updated for 6.3.x Added command: BACKUP, USR_LVL_SET Added optional arguments to commands: V_FLUSH, V_FLUSH_MARKER. Added more statuses to the list. Fixed errors in examples.
15	2018-09-29	Updated for 7.1.x Added commands: USR_LVL_GET, USR_LVL_SET
16	2019-09-18	Updated for 9.0.x Added commands: V_SET_DROP_TOF Added commands: ENC_PPM

Table 1: Revision history.