Dotpeenator®

BY SERMATEK

INTEGRATION GUIDE · TCP/IP & SERIAL

Command Communication Protocol.

Text-based command protocol for Dotpeenator® dot peen marking machines over TCP/IP or serial port. It lets PLCs, MES/ERP systems and custom software set mark content, manage files, control marking and motion, and work with the built-in data list.

INTERFACE	ENCODING	TERMINATOR	PROTOCOL VERSION
TCP client / server · Serial	UTF-8 / GBK / Latin1	\n	V9.0.S0513

In command names and responses (for example `Start`, `!PrinterBusy`, `>Print Stopped`), the words *print* and *printer* refer to the Dotpeenator® marking machine. Commands and responses must be used exactly as written.

1 Communication Basics

1.1 Connection Modes

MODE	DESCRIPTION
Client Mode	The controller acts as a TCP client, actively connecting to an external server
Server Mode	The controller acts as a TCP server, waiting for external clients to connect
Serial Port 1	Communication via USART1
Serial Port 2	Communication via USART2

1.2 Data Format

- **Encoding:** supports UTF-8 / GBK / Latin1 etc., selectable in settings
- Commands are delimited by `\n`; multiple commands can be sent in one transmission (separated by newlines)

- Controller responses use `>` as the start symbol by default (configurable) and are terminated with `\n`
- Command keywords are case-insensitive

1.3 General Response Rules

RESPONSE	MEANING
`>data content`	Normal response (`>` is the start symbol, configurable)
`!PrinterBusy`	Machine is busy, cannot execute the current command
`!BadFormat:...`	Command format error
`!BadProperty:...`	Unsupported property name
`!CanNotLock:...`	Cannot lock mark object
`!BadCommand:...`	Unrecognized command
`!CodingFailed:...`	Encoding failed
`!SetPropFailed:...`	Failed to set property

1.4 Boolean Value Convention

When **writing** a boolean property, the following values represent "true": `T` / `t` / `1` / `True` (case-insensitive); all other values are "false".

When **reading** a boolean property, the response is `T` or `F`.

2 Mark Content Operations (Most Commonly Used)

2.1 `@` Set Mark Property

Sets the property value of the specified mark. Commands with the `@` prefix are rejected when the machine is busy.

FORMAT

```
@MarkName Property=Value
@MarkName=Value            # Property defaults to Code when omitted
```

SUPPORTED PROPERTIES

PROPERTY	DESCRIPTION	EXAMPLE
`Code`	Mark content (default)	`@M1=ABC123` / `@M1 Code=ABC123`
`Pos`	Mark position (x,y)	`@M1 Pos=100,200`
`Select` / `Sel`	Selection state	`@M1 Select=T`
`Index` / `Idx`	Mark index	`@M1 Index=2`
`Enable`	Enable marking	`@M1 Enable=F`
Other extended properties	Depends on mark type	`@M1 FontSize=12`

EXAMPLES

```
@M1=ABC123            → Set M1 content to ABC123
@M1 Code=ABC123       → Same as above, explicitly specifying Code
@M1=Hello World       → Content may contain spaces (everything after = is the value)
@M1 Pos=100.5,200.3   → Set M1 position
@M1 Pos=100,          → Set X only, Y remains unchanged
@M1 Select=T          → Select M1
@M1 Enable=F          → Disable M1 marking
@M1 Index=3           → Set M1 index to 3
```

RESPONSE: On success, returns the mark name, e.g. `>M1` ; on failure, returns `!CodingFailed:...` etc.

2.2 `Get` Query Mark Property

FORMAT

```
Get MarkName [Property]
```

When the property name is omitted, `Code` is queried by default.

SUPPORTED PROPERTIES

PROPERTY	DESCRIPTION	RETURN VALUE
`Code`	Mark content	String
`Pos`	Mark position	`x,y`
`Select` / `Sel`	Selection state	`T` or `F`
`Index` / `Idx`	Mark index	Integer
`Enable`	Enable marking	`T` or `F`
Other extended properties	Depends on mark type	String

EXAMPLES

```
Get M1          → Query M1 content (default: Code)
Get M1 Code     → Same as above
Get M1 Pos      → Query M1 position, returns e.g. >100.5,200.3
Get M1 Select   → Query M1 selection state, returns >T or >F
Get M1 Enable   → Query whether M1 marking is enabled
Get M1 Index    → Query M1 index
```

2.3 `Set` Set Mark Property (Equivalent to @)

The `Set` command is functionally identical to the `@` prefix, differing only in command format. It is rejected when the machine is busy.

FORMAT

```
Set MarkName Property=Value
Set MarkName=Value
```

EXAMPLES

```
Set M1=ABC123        → Set M1 content
Set M1 Pos=100,200   → Set M1 position
```

2.4 Formatted Data Input — Batch Set Mark Content

By configuring "Allow receiving custom formatted strings" and "Formatted string parameters", you can batch-set all mark contents with a single formatted string.

FORMAT

```
StartSymbolData1SeparatorData2SeparatorData3...
```

The default start symbol is `^` and the default separator is `,`, i.e.: `^ABC,DEF,,GHI`

RULES

- Data is matched to marks in ascending order of Index, then by name
- Non-empty data: sets the mark content and enables marking
- Empty data: disables marking for that mark
- Marks without provided data: marking enabled (original content preserved)
- If "Print immediately after receiving formatted data" is configured, marking starts automatically

Assuming the document contains three marks: `M1(idx=1)`, `M2(idx=2)`, `M3(idx=3)`:

EXAMPLES

```
^ABC,DEF,GHI   → M1=ABC, M2=DEF, M3=GHI, all enabled
^ABC,DEF,      → M1=ABC, M2=DEF, M3 marking disabled
^ABC,,GHI      → M1=ABC, M2 disabled, M3=GHI
^ABC           → M1=ABC, M2/M3 keep original content and stay enabled
```