Dotpeenator®

BY SERMATEK

GUIA DE INTEGRAÇÃO · TCP/IP E SERIAL

Protocolo de Comunicação por Comandos.

Protocolo de comandos em texto para as máquinas de marcação por micropercussão Dotpeenator®, via TCP/IP ou porta serial. Permite que CLPs, sistemas MES/ERP e softwares personalizados definam conteúdos, gerenciem arquivos, controlem a marcação e os movimentos e utilizem a lista de dados interna.

INTERFACE	CODIFICAÇÃO	TERMINADOR	VERSÃO DO PROTOCOLO
TCP cliente / servidor · Serial	UTF-8 / GBK / Latin1	\n	V9.0.S0513

Nos nomes de comandos e nas respostas (por exemplo `Start`, `!PrinterBusy`, `>Print Stopped`), as palavras *print* e *printer* referem-se à máquina de marcação Dotpeenator®. Comandos e respostas devem ser usados exatamente como escritos.

1 Fundamentos da Comunicação

1.1 Modos de Conexão

MODO	DESCRIÇÃO
Modo Cliente	O controlador atua como cliente TCP, conectando-se ativamente a um servidor externo
Modo Servidor	O controlador atua como servidor TCP, aguardando a conexão de clientes externos
Porta Serial 1	Comunicação via USART1
Porta Serial 2	Comunicação via USART2

1.2 Formato de Dados

- **Codificação:** suporta UTF-8 / GBK / Latin1 etc., selecionável nas configurações

- Os comandos são delimitados por `\n`; vários comandos podem ser enviados em uma única transmissão (separados por quebras de linha)
- As respostas do controlador usam `>` como símbolo inicial por padrão (configurável) e terminam com `\n`
- As palavras-chave dos comandos não diferenciam maiúsculas de minúsculas

1.3 Regras Gerais de Resposta

RESPOSTA	SIGNIFICADO
`>data content`	Resposta normal (`>` é o símbolo inicial, configurável)
`!PrinterBusy`	A máquina está ocupada e não pode executar o comando atual
`!BadFormat:...`	Erro de formato do comando
`!BadProperty:...`	Nome de propriedade não suportado
`!CanNotLock:...`	Não é possível bloquear o objeto de marcação
`!BadCommand:...`	Comando não reconhecido
`!CodingFailed:...`	Falha na codificação
`!SetPropFailed:...`	Falha ao definir a propriedade

1.4 Convenção de Valores Booleanos

Ao **gravar** uma propriedade booleana, os seguintes valores representam "verdadeiro": `T` / `t` / `1` / `True` (sem diferenciar maiúsculas/minúsculas); todos os outros valores são "falso".

Ao **ler** uma propriedade booleana, a resposta é `T` ou `F`.

---

2 Operações de Conteúdo de Marcação (Mais Usadas)

2.1 `@` Definir Propriedade do Objeto

Define o valor de uma propriedade do objeto de marcação especificado. Comandos com o prefixo `@` são rejeitados quando a máquina está ocupada.

FORMATO

```
@MarkName Property=Value
@MarkName=Value            # Se omitida, a propriedade padrão é Code
```

PROPRIEDADES SUPORTADAS

PROPRIEDADE	DESCRIÇÃO	EXEMPLO
`Code`	Conteúdo do objeto (padrão)	`@M1=ABC123` / `@M1 Code=ABC123`
`Pos`	Posição do objeto (x,y)	`@M1 Pos=100,200`
`Select` / `Sel`	Estado de seleção	`@M1 Select=T`
`Index` / `Idx`	Índice do objeto	`@M1 Index=2`
`Enable`	Habilitar marcação	`@M1 Enable=F`

PROPRIEDADE	DESCRIÇÃO	EXEMPLO
Outras propriedades estendidas	Depende do tipo de objeto	`@M1 FontSize=12`

EXEMPLOS

```
@M1=ABC123            → Define o conteúdo de M1 como ABC123
@M1 Code=ABC123       → Igual ao anterior, especificando Code explicitamente
@M1=Hello World       → O conteúdo pode ter espaços (tudo após = é o valor)
@M1 Pos=100.5,200.3   → Define a posição de M1
@M1 Pos=100,          → Define apenas X; Y permanece inalterado
@M1 Select=T          → Seleciona M1
@M1 Enable=F          → Desabilita a marcação de M1
@M1 Index=3           → Define o índice de M1 como 3
```

RESPOSTA: Em caso de sucesso, retorna o nome do objeto, ex.: `>M1` ; em caso de falha, retorna `!CodingFailed:...` etc.

2.2 `Get` Consultar Propriedade do Objeto

FORMATO

```
Get MarkName [Property]
```

Quando o nome da propriedade é omitido, `Code` é consultado por padrão.

PROPRIEDADES SUPORTADAS

PROPRIEDADE	DESCRIÇÃO	VALOR DE RETORNO
`Code`	Conteúdo do objeto	String
`Pos`	Posição do objeto	`x,y`
`Select` / `Sel`	Estado de seleção	`T` ou `F`
`Index` / `Idx`	Índice do objeto	Inteiro
`Enable`	Marcação habilitada	`T` ou `F`
Outras propriedades estendidas	Depende do tipo de objeto	String

EXEMPLOS

```
Get M1          → Consulta o conteúdo de M1 (padrão: Code)
Get M1 Code     → Igual ao anterior
Get M1 Pos      → Consulta a posição de M1, retorna ex.: >100.5,200.3
Get M1 Select   → Consulta o estado de seleção de M1, retorna >T ou >F
Get M1 Enable   → Consulta se a marcação de M1 está habilitada
Get M1 Index    → Consulta o índice de M1
```

2.3 `Set` Definir Propriedade do Objeto (Equivalente a @)

O comando `Set` é funcionalmente idêntico ao prefixo `@`, diferindo apenas no formato. É rejeitado quando a máquina está ocupada.

FORMATO

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

EXEMPLOS

```
Set M1=ABC123        → Define o conteúdo de M1
Set M1 Pos=100,200   → Define a posição de M1
```

2.4 Entrada de Dados Formatados — Definição em Lote

Configurando "Permitir recebimento de strings formatadas personalizadas" e "Parâmetros de string formatada", é possível definir em lote o conteúdo de todos os objetos com uma única string formatada.

FORMATO

```
StartSymbolData1SeparatorData2SeparatorData3...
```

O símbolo inicial padrão é `^` e o separador padrão é `,`, ou seja: `^ABC,DEF,,GHI`

REGRAS

- Os dados são associados aos objetos em ordem crescente de Index e, depois, por nome
- Dado não vazio: define o conteúdo do objeto e habilita a marcação
- Dado vazio: desabilita a marcação desse objeto
- Objetos sem dados fornecidos: marcação habilitada (conteúdo original preservado)
- Se "Imprimir imediatamente após receber dados formatados" estiver configurado, a marcação inicia automaticamente

Supondo que o documento contenha três objetos: `M1(idx=1)`, `M2(idx=2)`, `M3(idx=3)`:

EXEMPLOS

```
^ABC,DEF,GHI   → M1=ABC, M2=DEF, M3=GHI, todos habilitados
^ABC,DEF,      → M1=ABC, M2=DEF, marcação de M3 desabilitada
^ABC,,GHI      → M1=ABC, M2 desabilitado, M3=GHI
^ABC           → M1=ABC, M2/M3 mantêm o conteúdo original e continuam habilitados
```