Dotpeenator®

BY SERMATEK

ENTEGRASYON KILAVUZU · MODBUS TCP

Modbus Register Kılavuzu.

Dotpeenator® nokta vuruşlu markalama makinelerini Modbus TCP üzerinden PLC ve ana bilgisayarlara bağlamak için register haritası ve çerçeve düzeyinde örnekler: dosya seçimi, marka içeriği, konum ve boyut, markalama kontrolü ve G/Ç.

PROTOKOL	PORT	UNİT ID	KODLAMA
Modbus TCP	502	1	UTF-8

Durum adlarında ve olay tanımlayıcılarında (örneğin `Printing`, `PrintFinished`, `OnPrintStart`) geçen *print* kelimesi Dotpeenator® makinesindeki markalamayı ifade eder. Bu tanımlayıcılar kontrolcünün kullandığı şekliyle aynen verilmiştir.

1 Genel Bakış

Dotpeenator® markalama makinesi, harici master istasyonlarla (PLC, ana bilgisayar vb.) Modbus protokolü üzerinden haberleşir. Makine (slave) Modbus sunucusu olarak çalışır ve master'ın okuyup yazabileceği dört veri alanı sunar:

VERİ ALANI	TİP	ADRES ARALIĞI	ADET	ERİŞİM
Coils	Bit	0 ~ 8	9	Okuma/Yazma
Discrete Inputs	Bit	0 ~ 11	12	Salt okunur
Input Registers	16-bit word	0 ~ 9	10	Salt okunur
Holding Registers	16-bit word	0 ~ 65534	65535	Okuma/Yazma

VARSAYILAN HABERLEŞME PARAMETRELERİ

PARAMETRE	DEĞER
Protokol	Modbus TCP
Port	502
Slave adresi (Unit ID)	1

PARAMETRE	DEĞER
Karakter kodlaması	UTF-8

FONKSİYON KODLARI

FONKSİYON KODU	İŞLEM	VERİ ALANI
`0x01`	Coil oku	Coils
`0x02`	Discrete Input oku	Discrete Inputs
`0x03`	Holding Register oku	Holding Registers
`0x04`	Input Register oku	Input Registers
`0x05`	Tek coil yaz	Coils
`0x06`	Tek Holding Register yaz	Holding Registers
`0x0F`	Çoklu coil yaz	Coils
`0x10`	Çoklu Holding Register yaz	Holding Registers

2 Register Modları

Kontrolcü, Holding Register'larda metin saklamak için iki mod destekler. Mod, kontrolcü yazılımındaki Modbus modülü ayar penceresinden seçilir.

MOD	AÇIKLAMA
Direct (tek bayt)	Her bayt, yalnızca alt 8 biti kullanılarak bir register'a yazılır; sıfır değerli register metni sonlandırır
Packed (çift bayt)	Her iki bayt bir register'a paketlenir (ilk bayt alt 8 bitte, ikinci bayt üst 8 bitte); sıfır değerli register metni sonlandırır

İki mod da aynı register adres düzenini kullanır; yalnızca metnin register içinde kodlanma şekli farklıdır. Varsayılan mod **Direct**'tir ve bu doküman açıklamalarda ağırlıklı olarak bu modu kullanır.

3 Holding Register Adres Haritası

Holding Register'lar ana veri alanıdır; dosya işlemleri, markalama kontrolü ve marka verisi okuma/yazma için kullanılır.

ADRES ARALIĞI	KULLANIM	REGİSTER SAYISI
0 ~ 99	Dosya adı (uzantısız)	100
100	Rezerve	1
101	Markalama kontrol komutu	1
102 ~ 999	Rezerve	898
1000 ~ 1099	Marka 0 veri alanı	100
1100 ~ 1199	Marka 1 veri alanı	100

ADRES ARALIĞI	KULLANIM	REGİSTER SAYISI
...	...	...
1000+i×100 ~ 1099+i×100	Marka i veri alanı	100
...	...	...
10900 ~ 10999	Marka 99 veri alanı	100
11000 ~ 65534	Rezerve	54535

HER MARKA VERİ ALANININ YAPISI (100 REGİSTER)

OFSET	ADRES (MARKA İ)	KULLANIM	REGİSTER
+0 ~ +91	1000+i×100 ~ 1091+i×100	Marka içeriği (metin)	92
+92 ~ +93	1092+i×100 ~ 1093+i×100	X konumu (CDAB float)	2
+94 ~ +95	1094+i×100 ~ 1095+i×100	Y konumu (CDAB float)	2
+96 ~ +97	1096+i×100 ~ 1097+i×100	Genişlik W (CDAB float)	2
+98 ~ +99	1098+i×100 ~ 1099+i×100	Yükseklik H (CDAB float)	2

4 Direct Mod (Tek Bayt) Ayrıntıları

4.1 Marka İçerik Register'ları

> **Adres:** `1000+i×100 ~ 1091+i×100` (marka başına 92 register)

Direct modda metnin her baytı bir register'ın alt 8 bitine yazılır; üst 8 bit her zaman sıfırdır. Son bayttan sonra sonlandırıcı olarak sıfır değerli bir register gelmelidir.

KODLAMA KURALLARI

1. Marka içeriğini UTF-8 (veya ayarlı kodlama) ile bayt dizisine çevirin
2. Her baytı bir register'a yazın (değer aralığı `0x0001 ~ 0x00FF`)
3. Son bayttan sonra sonlandırıcı olarak sıfır değerli bir register ekleyin
4. Kalan register'ları 0 ile doldurun

ÖRNEK: MARKA 0 İÇERİĞİ "ABC"

UTF-8 baytları: `0x41 0x42 0x43`

REGİSTER ADRESİ	DEĞER (HEX)	ANLAMI
1000	0x0041	'A'
1001	0x0042	'B'
1002	0x0043	'C'
1003	0x0000	Sonlandırıcı
1004 ~ 1091	0x0000	Kullanılmıyor

Modbus çerçeve örneği — marka 0 içeriğine "ABC" yaz (FC 0x10)

İSTEK

00 01	00 00	00 0F	01	10	03 E8	00 04	08	00 41	00 42	00 43	00 00
TID	PID	Len	Unit	FC	Addr	Qty	BC	Reg0	Reg1	Reg2	Reg3
	0	15	1	0x10	1000	4	8	'A'	'B'	'C'	Son

YANIT

00 01	00 00	00 06	01	10	03 E8	00 04
TID	PID	Len	Unit	FC	Addr	Qty
	0	6	1	0x10	1000	4

ÖRNEK: MARKA 1 İÇERİĞİ "HELLO"

UTF-8 baytları: `0x48 0x65 0x6C 0x6C 0x6F`. Marka 1 başlangıç adresi = 1000 + 1×100 = **1100**.

REGİSTER ADRESİ	DEĞER (HEX)	ANLAMI
1100	0x0048	'H'
1101	0x0065	'e'
1102	0x006C	'l'
1103	0x006C	'l'
1104	0x006F	'o'
1105	0x0000	Sonlandırıcı
1106 ~ 1191	0x0000	Kullanılmıyor

Modbus çerçeve örneği — marka 1 içeriğine "Hello" yaz (FC 0x10)

İSTEK

00 02	00 00	00 13	01	10	04 4C	00 06	0C	00 48	00 65	00 6C	00 6C	00 6F	00 00
TID	PID	Len	Unit	FC	Addr	Qty	BC	Reg0	Reg1	Reg2	Reg3	Reg4	Reg5
	0	19	1	0x10	1100	6	12	'H'	'e'	'l'	'l'	'o'	Son

YANIT

00 02	00 00	00 06	01	10	04 4C	00 06
TID	PID	Len	Unit	FC	Addr	Qty
	0	6	1	0x10	1100	6

ÇOK BAYTLI KARAKTER ÖRNEĞİ: MARKA 0 İÇERİĞİ "ÇELİK"

UTF-8 baytları: `0xC3 0x87 0x65 0x6C 0x69 0x6B` (6 bayt — "Ç" 2 bayt yer kaplar)

REGİSTER ADRESİ	DEĞER (HEX)	ANLAMI
1000	0x00C3	'Ç' karakterinin 1. UTF-8 baytı
1001	0x0087	'Ç' karakterinin 2. UTF-8 baytı
1002	0x0065	'e'
1003	0x006C	'l'
1004	0x0069	'i'
1005	0x006B	'k'
1006	0x0000	Sonlandırıcı