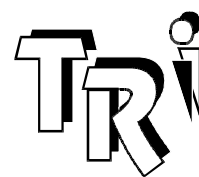


# T100MD1616+

超功能可编程序控制器

用户手册（下册）

电科研国际有限公司



## 版 权 声 明

TRiLOGI1.X, 2.X, 3.X和4.X为新加坡电科研国际私人有限公司(“TRi”)所拥有的商标及版权©1989-1999。

版权所有, 不经该公司的书面允许, 严禁将手册中的任一部分以任何形式或方式复制。

Tri对手册内容并不作承诺和保证。另外, 内容的改变也不作事先通知。此手册经过慎密而周全的编写, 但是TRi对因为使用此手册时而造成的错误,疏忽, 及其破坏不承担责任。

IBM PC, XT和AT皆为国际商用机器公司(IBM)的注册商标。

MSDOS是微软公司(Microsoft)的注册商标。  
其他商标皆属商标各自相关所有者。

## 销售条件和产品保证

新加坡电科研国际私人有限公司(“TRi”)和购买者遵守如下有关销售和购买条款:

1. T100M+可编程控制器保证期为一年,自登记购买之日起计算。在此期间内,任何部件发现问题,由TRi公司认定后,给予修理或调换。
2. TRi公司对由使用者因更换,疏忽,误操作或不当安装而引起的部件损坏不承担责任。
3. TRi公司对在使用中产生的特殊损害或间接损害而造成的损失或追索不承担责任。该产品不得用于会因产品故障而导致人体受伤或损失之处。购买者必须对其所购买的产品进行测试,以遵守不同行业的安全准则。
4. 由TRi公司分销而非制造的产品同样拥有原生产商的所有保证。这些产品包括但不限于:电源,感应器,输入输出模块和蓄电池存储器。
5. TRi公司保留在任何时间修改产品特征和说明的权利。

**请购买者注意:**若您对上述条款和条件持有异议,并于购买之日起30日内将本产品退还给生产厂家或分销商,您可获得全额退款。

## 目 录

第一章 特殊I/O及模拟量接口	页码
1.1 简介	1-1
1.2 特殊数字输入/输出	1-1
1.3 步进电机控制器输出	1-2
1.4 脉宽调制输出	1-4
1.5 用高速计数器输入连接回转式编码器	1-6
1.6 中断输入的使用	1-8
1.7 脉冲测量输入的应用	1-9
1.8 模拟输入/输出	1-9
1.9 串行通讯端口	1-11
1.9.1    COMM1: 带DB9型阴连接器的RS232C端口	1-11
1.9.2    COMM3: 双线RS485端口	1-12
1.9.3    改变波特比率和通信方式: 使用SETBAUD语句	1-14
1.9.4    支持多重通信协议	1-15
1.9.5    从TBASIC内读取COMM端口	1-16
1.9.6    用调制解调器遥控编程/监视T100MD1616+PLC	1-18
1.9.7    构造一2 <sup>nd</sup> 多层网络	1-19
1.10 DIP开关	1-19
1.11 CPU状态显示器	1-20
1.12 内部继电器、定时器和计数器	1-21
第二章 操作程序	
2.1 编程	2-1
2.2 模拟	2-1
2.3 传送梯形逻辑	2-1
2.4 出错与问题	2-2
2.5 在线监控	2-2
2.5.1    监控PLC输入/输出逻辑状态	2-2
2.5.2    观察与更改PLC的内部变量	2-3
2.5.3    输入/输出位的强制设置和复位	2-3
2.5.4    中止PLC的梯形程序	2-3
2.6 梯形监视	2-3
2.7 从PLC上载梯形图程序	2-3
2.8 改变定时器和计数器的设置值	2-4
2.9 设定PLC实时时钟	2-4
2.10 排除通讯故障	2-5
第三章 上位通讯	
3.1 点对点通讯	3-2
3.2 多点式通讯系统	3-3

3.2.1 RS485网络接口硬件	3-3
3.2.2 RS485接口的保护	3-4
3.2.3 单个主控制器RS485网络基础	3-5
3.2.4 多个主控制器RS485网络基础	3-5
3.2.5 指令/应答块格式 (多点通讯)	3-7
3.2.6 通讯步骤	3-8
3.3 网络TRiLOGI的使用	3-9
3.4 RS485网络的故障排除	3-10

## 第四章 指令与应答格式

4.1 读出识别符	4-1
4.2 写入识别符	4-1
4.3 读出输入通道	4-2
4.4 读出输出通道	4-3
4.5 读出内部继电器通道	4-3
4.6 读出定时接触器	4-4
4.7 读出计数接触器	4-5
4.8 读出定时器当前值(P.V.)	4-5
4.9 读出定时器设置值(S.V.)	4-6
4.10 读出计数器当前值	4-6
4.11 读出计数器设置值	4-7
4.12 读出整数变量(A至Z)	4-7
4.13 读出变量字符串(A\$至Z\$)	4-7
4.14 读出数据存储器变量(DM[1]值DM[4000])	4-8
4.15 读出系统变量	4-9
4.16 读出变量 - 高速计数器HSCPV[ ]	4-10
4.17 写入输入	4-10
4.18 写入输出	4-10
4.19 写入内部继电器	4-11
4.20 写入定时接触器	4-11
4.21 写入计数接触器	4-11
4.22 写入定时器当前值(P.V.)	4-11
4.23 写入定时器设置值(S.V.)	4-12
4.24 写入计数器当前值(P.V.)	4-12
4.25 写入计数器设置值(S.V.)	4-13
4.26 写入整数变量(A至Z)	4-13
4.27 写入字符串变量(A\$至Z\$)	4-13
4.28 写入数据存储器变量(DM[1]至DM[4000])	4-14
4.29 写入变量 - 系统变量	4-14
4.30 写入变量 - 高速计数器HSCPV[ ]	4-15
4.31 更新实时时钟模块	4-15

4.32 PLC的暂停	4-16
4.33 恢复PLC的作业	4-16
4.34 上位通讯程序的例子	4-16
4.35 使用NETCMD\$命令连成内部网络	4-17
4.36 内部网络之间使用MODBUS协议	4-18
<b>第五章 支持MODBUS/OMRON协议</b>	
5.1 MODBUS ASCII协议	5-1
5.2 MODBUS ASCII协议	5-4
5.3 支持OMRON主连接命令	5-5
5.4 应用实例：与SCADA软件接口	5-5
5.5 使用T100M+PLC作为主MODBUS	5-6
<b>第六章 用EMIT 3.0 与Internet 连接</b>	
6.1 EMIT 3.0介绍	6-1
6.2 T100M+ PLCs 和 EMIT 3.0	6-1
6.3 获得并安装EMIT 3.0	6-2
6.4 安装演示文件	6-2
6.5 运行 Internet 演示程序	6-3
6.6 编写 Java 用户自己的界面程序	6-5

第一章 至 第三章

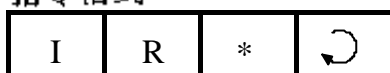
请参阅用户手册（上册）

## 第四章 指令与应答格式

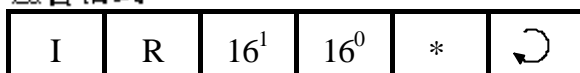
本章描述所有T100MX系列PLC主连接指令的指令块与应答块的详细格式。这里仅说明用于点对点通讯系统的格式，但所有这些指令同样也可用于多点系统，若要将指令用于多点系统，只需要在指令字头前面增加识别符 (@m) 以及在数据结尾处增加帧检查序列（参见第3章中有关多点通讯指令格式的详细描述）。

### 4.1 读出识别符

指令格式



应答格式



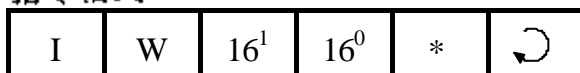
识别符 (00 - FF)

识别符用于下面一类多点式通讯方式，即主计算机可以同连接到公用RS485总线的任何PLC进行选择式通讯（详情请参见第3章）。识别符不会影响点对点通讯。

识别符储存在PLC的EEPROM中，因此会留在PLC中，直至下一次变更。

### 4.2 写入识别符

指令格式



识别符 (00 - FF)

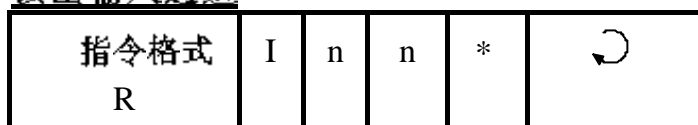
应答格式



若要把PLC识别符定义为0A，可向PLC发送指令串“TW0A\*”。

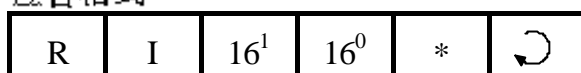


### 4.3 读出输入通道



8位通道# (16进制)

应答格式



8位数据 (16进制)

### 输入通道的定义

下面的表列出了TRiLOGI输入表中定义的输入编号与输入通道编号之间的对应关系。

	输入/输出编号							
	第7位							第0位
CH00:	8	7	6	5	4	3	2	1
CH01:	16	15	14	13	12	11	10	9
CH02:	24	23	22	21	20	19	18	17
CH03:	32	31	30	29	28	27	26	25
CH04:	40	39	38	37	36	35	34	33
CH05:	48	57	56	45	44	43	42	41
CH06:	56	55	54	53	52	51	50	49
CH07:	64	63	62	61	60	59	58	57
CH08:	72	71	70	69	68	67	66	65
CH09:	80	79	78	77	76	75	74	73
CH0A <sub>16</sub> :	88	87	86	85	84	83	82	81
CH0B <sub>16</sub> :	96	95	94	93	92	91	90	89
CH0C <sub>16</sub> :	104	103	102	101	100	99	98	97
CH0D <sub>16</sub> :	112	111	110	109	108	107	106	105
CH0E <sub>16</sub> :	120	119	118	117	116	115	114	113
CH0F <sub>16</sub> :	128	127	126	125	124	123	122	121

每个通道的8位输入用2个字节的ASCII字符串来表示，用的是16进制值。例如，若输入1-3为逻辑“0”，输入4-10为逻辑“1”，而所有其它输入都为逻辑“0”，则发送指令“RI00\*”时会得到应答“RIF8\*”（F8<sub>16</sub>=111110

00<sub>2</sub>)。

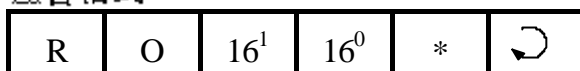
#### 4.4 读出输出通道

指令格式



8位通道# (16进制)

应答格式



8位数据 (16进制)

细节请参阅第4.3节“读出输入通道”。

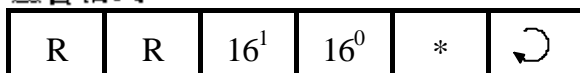
#### 4.5 读出内部继电器通道

指令格式



8位通道# (16进制)

应答格式



8位数据 (16进制)

继电器通道编号的定义

所有T100MXPLC都支持256个内部继电器，前128个内部继电器的通道编号与输入输出相同，其余的继电器及其分配到的通道示于下面表格中：

	内部继电器编号							
	第7位							第0位
CH10 <sub>16</sub> :	136	135	134	133	132	131	130	129
CH11 <sub>16</sub> :	144	143	142	141	140	139	138	137
CH12 <sub>16</sub> :	152	151	150	149	148	147	146	145
CH13 <sub>16</sub> :	160	159	158	157	156	155	154	153
CH14 <sub>16</sub> :	168	167	166	165	164	163	162	161
CH15 <sub>16</sub> :	176	175	174	173	172	171	170	169
CH16 <sub>16</sub> :	184	183	182	181	180	179	178	177
CH17 <sub>16</sub> :	192	191	190	189	188	187	186	185
CH18 <sub>16</sub> :	200	199	198	197	196	195	194	193
CH19 <sub>16</sub> :	208	207	206	205	204	203	202	201
CH1A <sub>16</sub> :	216	215	214	213	212	211	210	209
CH1B <sub>16</sub> :	224	223	222	221	220	219	218	217
CH1C <sub>16</sub> :	232	231	230	229	228	227	226	225
CH1D <sub>16</sub> :	240	239	238	237	236	235	234	233
CH1E <sub>16</sub> :	248	247	246	245	244	243	242	241
CH1F <sub>16</sub> :	256	255	254	253	252	251	250	249

#### 4.6 读出定时接触器

指令格式

R	T	n	n	*	↻
---	---	---	---	---	---

8位通道# (16进制)

应答格式

R	T	16 <sup>1</sup>	16 <sup>0</sup>	*	↻
---	---	-----------------	-----------------	---	---

8位数据 (16进制)

##### 定时接触器通道编号定义

定时接触器是一个单位储存器，8个定时接触器即可组成类似于输入/输出通道的8位通道。

下面表示出了TRiLOGI定时器输入表中所定义的定时器编号与通道号的对应关系。

CH0:	8	7	6	5	4	3	2	1
CH1:	16	15	14	13	12	11	10	9
CH2:	24	23	22	21	20	19	18	17
CH3:	32	31	30	29	28	27	26	25
CH4:	40	39	38	37	36	35	34	33
CH5:	48	57	56	45	44	43	42	41
CH6:	56	55	54	53	52	51	50	49
CH7:	64	63	62	61	60	59	58	57

#### 4.7 读出计数接触器

指令格式

R	C	n	n	*	↻
---	---	---	---	---	---

8位通道# (16进制)

应答格式

R	C	16 <sup>1</sup>	16 <sup>0</sup>	*	↻
---	---	-----------------	-----------------	---	---

8位数据 (16进制)

##### 计数接触器通道编号定义

64个计数接触器的通道编号分配方法与64个定时器完全相同。请参阅第4.6节“读出定时接触器”中的细节。

#### 4.8 读出定时器当前值 (P.V.)

指令格式

R	M	n	n	*	↻
---	---	---	---	---	---

mn: 定时器1=00, …定时器16=0F, …定时器64=3F

应答格式

R	M	10 <sup>3</sup>	10 <sup>2</sup>	10 <sup>1</sup>	10 <sup>0</sup>	*	↻
---	---	-----------------	-----------------	-----------------	-----------------	---	---

### 定时器当前值（10进制）

被指定的定时器的当前值（P.V.）以10进制形式返回，它是4个字节的ASCII字符，从0000到9999。

## 4.9 读出定时器设置值（S.V.）

### 指令格式

R	m	n	n	*	↻
---	---	---	---	---	---

mn: 定时器1 = 00, ...定时器16 = 0F, ...定时器64 = 3F

### 应答格式

R	m	10 <sup>3</sup>	10 <sup>2</sup>	10 <sup>1</sup>	10 <sup>0</sup>	*	↻
---	---	-----------------	-----------------	-----------------	-----------------	---	---

定时器设置值（10进制）

被指定的定时器的设置值（S.V.）以10进制形式送回，它是4个字节的ASCII字符，从0000到9999。请注意，该个指令的字头部分含有小写字母“m”，而不是“RM”指令串中的大写“M”。

## 4.10 读出计数器当前值

### 指令格式

R	U	n	n	*	↻
---	---	---	---	---	---

mn: 计数器1 = 00, ...计数器16 = 0F, ...计数器64 = 3F

### 应答格式

R	U	10 <sup>3</sup>	10 <sup>2</sup>	10 <sup>1</sup>	10 <sup>0</sup>	*	↻
---	---	-----------------	-----------------	-----------------	-----------------	---	---

计数器当前值（10进制）

被指定的计数器的当前值以10进制形式返回，它是4个字节的ASCII字符，从0000到9999。

### 4.11 读出计数器设置值

指令格式

R	u	n	n	*	↻
---	---	---	---	---	---

mn: 计数器1=00, …计数器16=0F, …计数器64=3F

应答格式

R	u	10 <sup>3</sup>	10 <sup>2</sup>	10 <sup>1</sup>	10 <sup>0</sup>	*	↻
---	---	-----------------	-----------------	-----------------	-----------------	---	---

计数器设置值 (10进制)

所指定的计数器的设置值以10进制形式送回，它是4个字节的ASCII字符，从0000到9999。请注意，这里的字头含小写字母“u”，而不是“RU”指令中的大写“U”。

### 4.12 读出整数变量 (A至Z)

指令格式

R	V	I	<i>alphabet</i>	*	↻
---	---	---	-----------------	---	---

A,B,C...Z

应答格式

R	V	I	16 <sup>7</sup>	16 <sup>6</sup>	16 <sup>5</sup>	16 <sup>4</sup>	16 <sup>3</sup>	16 <sup>2</sup>	16 <sup>1</sup>	16 <sup>0</sup>	*	↻
---	---	---	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	---	---

8位16进制整数，代表了32位整数

例如：为读出变量“K”的值，应发送主连接指令“RVIK\*”。如果变量K所含值为123456<sub>10</sub> (= 1E240<sub>16</sub>)，PLC将发送应答字符串“RVI0001E240\*”。

### 4.13 读出变量字符串 (A\$至Z\$)

指令格式

R	V	\$	<i>alphabet</i>	*	↻
---	---	----	-----------------	---	---

A,B,C...Z

应答格式

R	V	\$	<i>a</i>	<i>a</i>	<i>a</i>	...	...	<i>a</i>	<i>a</i>	<i>a</i>	*	↻
---	---	----	----------	----------	----------	-----	-----	----------	----------	----------	---	---

字符串的ASCII字符（长度可变）

例如：为读出字符串变量“M\$”的值，应发送主连接指令“RVSM\*”。若变量M\$含字符串“HelloWorld”，则PLC将发送“RV\$HelloWorld\*”应答字符串。

#### 4.14 读出数据存储器变量（DM[1]值DM[4000]）

指令格式

R	V	D	$16^3$	$16^2$	$16^1$	$16^0$	*	↻
---	---	---	--------	--------	--------	--------	---	---

0001 至0FA0 (4000<sub>10</sub>)

应答格式

R	V	D	$16^3$	$16^2$	$16^1$	$16^0$	*	↻
---	---	---	--------	--------	--------	--------	---	---

4个16进制数字代表16位整数

例如：为读出DM[3600]的值，应发出主连接指令“RVD0E10\*”。若变量DM[3600]所含值为12345<sub>10</sub> (= 3039<sub>16</sub>)，则PLC将发送应答字符串“RVD3039\*”。

#### 4.15 读出系统变量

通过该指令可读出T100MXPLC所有的16位系统变量，如Input[]、Output[]、Relay[]、counters[]、timer[]、定时器的当前值、计数器的当前值、CLK[]以及DATA[]。虽然输入、输出等也可通过“RI”、“RO”、“RR”等指令取得，但是用RVS指令取得的是16位字，而不是用上述指令取得的8位字节。取出32位系统变量HSCPV[]时，应当利用下一节中所叙述的“RVH”指令。某些SCADA软件驱动器可能更多地使用单个字头的指令“RVS

”来读取所有的输入和输出，读取不同类型的输入/输出时仅需改变“类型”编号。

RVS指令还能用来读取用于存储ADC、DAC和PWM值的内部变量，这些值是在最后一次执行ADC[]，setDAC或者setPWM命令过程中所获取的。但从TBASIC意义上来说，它们并不是系统变量。例如，在TBASIC中利用ADC[2]来读取ADC通道#2是非法的，（必须使用ADC[2]命令）。一个8位的16进制数用来表示系统变量的“类型”，如下表所示：

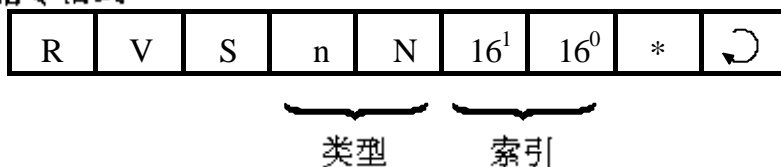
系统变量	类型
input[ ]	01
output[ ]	02
relay[ ]	03
timer[ ]	04
ctr[ ]	05
timerPV[ ]	06
ctrPV[ ]	07

系统变量	类型
clk[ ]	08
date[ ]	09
-	0A
ADC*	0B
DAC*	0C
PWM*	0D

\*

表示在TBASIC  
中不是系统变量

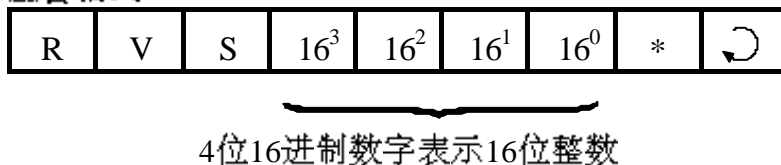
#### 指令格式



类型（01至0D）- 表示要读取的系统变量的类型。

索引（01至1F）- 阵列索引，从01开始

#### 应答格式



例如：为读取DATA[2]的值（表示实时时钟的月份），应发送指令“RVS09 02\*”。若PLC用“RVS0005\*”应答，则表示月份为5月。



#### 4.16 读出变量 – 高速计数器HSCPVI

指令格式

R	V	H	n	*	↻
---	---	---	---	---	---

通道：1或2

应答格式

R	V	H	$16^7$	$16^6$	$16^5$	$16^4$	$16^3$	$16^2$	$16^1$	$16^0$	*	↻
---	---	---	--------	--------	--------	--------	--------	--------	--------	--------	---	---

8个16进制数字代表32位整数

例如：为读取HSCPVI[2]的值，应发送主连接指令“RVH2\*”。若变量HSCPVI[2]所含值为123456<sub>10</sub> (= 1E240<sub>16</sub>)，PLC将发送应答字符串“RVH0001E240\*”。

#### 4.17 写入输入

指令格式

W	I	n	n	$16^1$	$16^0$	*	↻
---	---	---	---	--------	--------	---	---

通道编号#(00至0F)

数据

应答格式

W	I	*	↻
---	---	---	---

#### 4.18 写入输出

指令格式

W	O	n	n	$16^1$	$16^0$	*	↻
---	---	---	---	--------	--------	---	---

通道编号#

数据

(00 to 0F)

应答格式

W	O	*	↻
---	---	---	---

---

#### 4.19 写入内部继电器

指令格式

W	R	n	n	16 <sup>1</sup>	16 <sup>0</sup>	*	↻
---	---	---	---	-----------------	-----------------	---	---

└──────────┘
└──────────┘  
 通道编号#      数据

应答格式

W	R	*	↻
---	---	---	---

---

#### 4.20 写入定时接触器

指令格式

W	T	n	n	16 <sup>1</sup>	16 <sup>0</sup>	*	↻
---	---	---	---	-----------------	-----------------	---	---

└──────────┘
└──────────┘  
 通道编号#      数据  
 (00 至 07)

应答格式

W	T	*	↻
---	---	---	---

---

#### 4.21 写入计数接触器

指令格式

W	C	n	n	16 <sup>1</sup>	16 <sup>0</sup>	*	↻
---	---	---	---	-----------------	-----------------	---	---

└──────────┘
└──────────┘  
 通道编号#      数据  
 (00 至 07)

应答格式

W	C	*	↻
---	---	---	---

---

#### 4.22 写入定时器当前值 (P.V.)

## 指令格式

W	M	n	n	$10^3$	$10^2$	$10^1$	$10^0$	*	↻
---	---	---	---	--------	--------	--------	--------	---	---

定时器1=00,                  新定时器 PV

.....

定时器64=3F (16进制)

## 应答格式

W	M	*	↻
---	---	---	---

请注意：定时器编号从00开始，00表示定时器#1，01表示定时器#2……依此类推。

## 4.23 写入定时器设置值 (S.V.)

## 指令格式

W	m	n	n	$10^3$	$10^2$	$10^1$	$10^0$	*	↻
---	---	---	---	--------	--------	--------	--------	---	---

定时器1=00,                  新定时器 SV

.....

定时器64=3F (16进制)

## 应答格式

W	m	*	↻
---	---	---	---

说明：第2个字符为小写“m”，而不是“WM”指令中的大写“M”。

## 4.24 写入计数器当前值 (P.V.)

## 指令格式

W	U	n	n	$10^3$	$10^2$	$10^1$	$10^0$	*	↻
---	---	---	---	--------	--------	--------	--------	---	---

计数器1=00,                  新的 PV

....

计数器64=3F (Hex)

应答格式

W	U	*	↻
---	---	---	---

#### 4.25 写入计数器设置值 (S.V.)

指令格式

W	u	n	n	$10^3$	$10^2$	$10^1$	$10^0$	*	↻
---	---	---	---	--------	--------	--------	--------	---	---

计数器1=00,      新的计数器 PV

....

计数器64=3F (Hex)

应答格式

W	u	*	↻
---	---	---	---

注意：第2个字符为小写“u”，而不是“WU”指令中的大写“U”。

#### 4.26 写入整数变量 (A至Z)

指令格式

W	V	I	<i>alphabet</i>	$16^7$	$16^6$	$16^5$	$16^4$	$16^3$	$16^2$	$16^1$	$16^0$	*	↻
---	---	---	-----------------	--------	--------	--------	--------	--------	--------	--------	--------	---	---

A,B,C...Z

8位16进制数字代表32位整数

应答格式

W	V	I	*	↻
---	---	---	---	---

例如：若要将变量“K”指定为数字 $56789_{10}$  ( $=0DD5_{16}$ )，则应发送主连接指令“WVIK00000DD5\*”。

#### 4.27 写入字符串变量 (A\$至Z\$)

指令格式

W	V	\$	<i>alphabet</i>	<i>a</i>	<i>a</i>	...	...	<i>a</i>	<i>a</i>	*	↻
---	---	----	-----------------	----------	----------	-----	-----	----------	----------	---	---

A,B,C...Z      字符串的ASCII字符（长度可变）

应答格式

W	V	\$	*	↻
---	---	----	---	---

例如，为了将字符串“T100MXSuperPLC”分配至字符串变量P\$, 则应发送主连接指令“WV\$PT 100MX Super PLC\*”。

#### 4.28 写入数据储存器变量 (DM[1]至DM[4000])

指令格式

W	V	D	$16^3$	$16^2$	$16^1$	$16^0$	$16^3$	$16^2$	$16^1$	$16^0$	*	↻
---	---	---	--------	--------	--------	--------	--------	--------	--------	--------	---	---

16位阵列索引

0001 至 0FA0 ( $4000_{10}$ )

16位整数数据

应答格式

W	V	D	*	↻
---	---	---	---	---

例如：为写入值 $1234_{10}$  ( $=4D2_{16}$ ) 至DM[1000], 则应发送主连接指令“WV D03E804D2\*” ( $1000_{10}=3E8_{16}$ )。

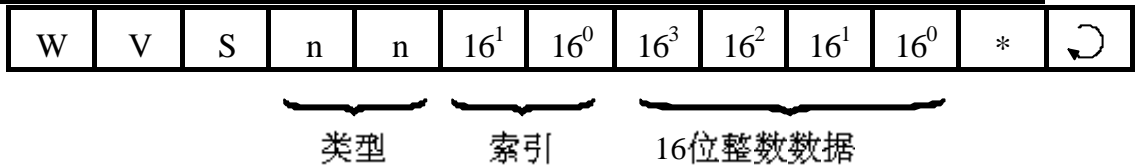
#### 4.29 写入变量 - 系统变量

系统变量	类型
Input[ ]	01
Output[ ]	02
Relay[ ]	03
Timer[ ]	04
ctr[ ]	05
TimerPV[ ]	06
CtrPV[ ]	07

系统变量	类型
clk[ ]	08
date[ ]	09
-	0A
ADC*	0B
DAC*	0C
PWM*	0D

\*表示在TBASIC中不是系统变量

指令格式



类型 (01 to 0D) - 表示要存储的系统变量的类型

索引 (01 to 1F) - 表示阵列索引，从01开始

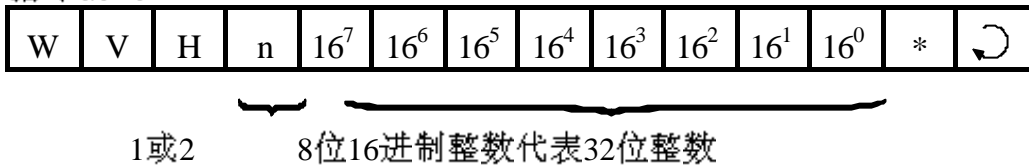
应答格式



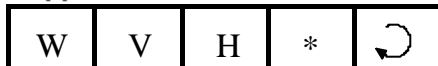
例如：若要将CLK[1]（它表示实时时钟）设定为14，则应发送指令“WVS0801000E\*”给PLC。

#### 4.30 写入变量－高速计数器HSCPVI ]

指令格式



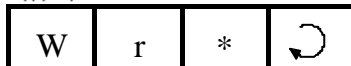
应答格式



例如，要清除HSCPVI[2]的值，则应发送主连接指令“WVH200000000\*”。

#### 4.31 更新实时时钟模块

指令格式



应答格式



若已装有电池带动的MX-RTC模块，则该指令使PLC强迫将TIME[ ]和DATE[ ]变量的值写入RTC模块。若PLC无RTC模块，则对此指令予以忽视。

### 4.32 PLC的暂停

指令格式

C	2	*	↻
---	---	---	---

应答格式

C	2	*	↻
---	---	---	---

当PLC接到该指令时，在当前扫描完成后，暂停执行PLC的梯形图程序。但PLC仍继续对输入/输出进行扫描。并处理发送给它的主连接指令，再将当前的输入/输出数据与内部变量报告给主计算机。

### 4.33 恢复PLC的作业

指令格式

C	1	*	↻
---	---	---	---

应答格式

C	1	*	↻
---	---	---	---

如果先前已用“C2”指令使其暂停，当PLC接到该指令时，便会恢复执行梯形图程序。否则，该指令无效。

### 4.34 上位通讯程序的例子

您可在“Controller”菜单中利用TRiLOGI内装的“Host - Link - Command”功能来试用本章中介绍的所有主连接指令。您可输入一个点对点或多点式指令串，再观察应答字符串。如果您利用写入指令已经将其中的一些数据更改过，则可激励在线监控方式后，再利用“View Variables”窗口来检查所作的更改。

TRiLOGI磁盘中提供了两个样本程序，可帮助编程人员进行初步熟悉，一个用微软公司QBASIC语言写成（HOST485.BAS），另一个用Borland-International公司的Turbo

C写成（HOST485.C）。这两个程序基本上执行相同的功能，即为：

- a) 提示用户通过PC机键盘输入所需的指令块。

- b) 启动通讯对话（仅适用于点对点方式），并向PLC发送指令串。
- c) 等候从PLC接收应答块，再将应答块显示在PC机屏幕上。

上述两个程序包含了以点对点或多点方式同M系列PLC进行成功通讯（用BASIC语言或C语言）所需的全部编码。此程序可在COMM1(RS232)端口和COMM3(RS485)端口上运行。因此编程人员可以将其中任意一个用作积木块来构造自己的应用程序。

这两个程序以下面的假定条件为基础，即使用了RS232 - RS485适配器，因而RS485总线的通讯方向是由RS232C的/RTS线路的状态来控制的。程序接收从键盘来的点对点指令和多点指令，并自动启动同PLC的正确通讯方式。如果RS485适配器工作有偏差，则必须更改指令“transmit485[]”和“receive485[]”，以便控制半双工RS485总线方向。可参阅RS485适配器技术手册中的细节。

在发送上述指令时，可以在“Controller”菜单中选TRiLOGI的“Host\_Link\_Command”项，用“IW”主连接指令将识别符分配给PLC。。

---

### 4.35 使用NETCMD\$命令连成内部网络

所有M - 系列PLC都可以使用TBASIC内置的NETCMD\$()函数发主联接命令给其他M - 系列或H - 系列PLC。此函数以多点形式接收主联接命令，计算帧频检测字符，并添加到命令串中，然后将整个命令随终止符一起发送出去。之后此函数等待一回响字符串并检查错误及整体性。如收到回响字符串正确此函数将返回一字符串。关于此命令的细节问题请参考TBASIC手册。

因此对处理PLC之间的网络问题，NETCMD\$()函数大大简化了编程任务。根据此章中规定的形式编程者只需构造正确的命令字符串，将构造的字符串传递给NETCMD\$()函数并检查其回响指令。M - 系列PLC可以使用NETCMD\$命令将其他PLC的I/O映射至内部继电器，并使用其他PLC作为远程I/O。

在“TRiLOGI\TL4”目录下有二个编程实例可很好得说明NETCMD()函数是如何将子PLC的I/O映射至主PLC。“REMOTE - H.PC4”和“REMOTE - M.PC4”详细地说明了PLC之间映射I/O的原理。其中“REMOTE - H.PC4”适用于H - 系列和M - 系列PLC作为子PLC，而“REMOTE - M.PC4”仅适用于M - 系列。这是因为M - 系列主联接命令级别高于H - 系列主联接命令设置



。此列中使用更有效的M-系列主联接命令来读/写M-系列PLC之间的16位数据。

#### **4.36 内部网络之间使用MODBUS协议**

T100M+PLC可以使用特殊MODBUS命令彼此之间传递数据，比起NETCMD\$就显得更加容易。但获得的变量仅限于以MODBUS地址结构映射的变量。对READMODBUS和WRITEMODBUS详细用法请参考下一章和TBASIC手册。

## 第五章 支持MODBUS/OMRON协议

T100M+PLC支持OMRON™和MODBUS™子集的(这两个都支持ASCII和RTU方式)兼容通信协议,因此那它能很容易得连接到如SCADA软件,触摸板等第三方控制的软/硬件产品。PLC自动地识别命令格式的类型并且返回正确响应。这些都不需要任何中间介入和PLC的任何配置。

MODBUS和Omron协议均使用同样的设备ID地址(从00到FF)与第三章所述的NATIVE协议相同。由于I/O的地址和内部变量在T100M+PLC中从OMRON或Modicon PLCs组织是不同的,因此需要将这些地址映射到其它PLC对应的内存区域,以便其根据相应的协议来读取。所有I/O,计时器,计数器,内部中继,数据内存DM[1]到DM[4000]的映射如表5.1所示。然而32位变量和字符串变量并不映射,因为应用在不同的PLC之中他们基本不同。不被映射的变量仍然可通过复制变量内容至数据内存DM[n](这可由CusFn来完成),并由第三方协议来读取。

通常情况下表5.1可作为第三方控制产品如触摸屏LCD面板的接口。

### 5.1 MODBUS ASCII 协议

T100M+支持下列命令和响应格式的MODBUS ASCII 协议:

START	Address	Function	Data	LRC Check	CRLF
:	2 chars	2 chars	# chars	2 chars	2 chars

支持下列函数码:

02	读输入状态
03	读保持寄存器
05	激活单个线圈。Coil#1=40001.1, Coil#1=40001.2…… Coil#17=40002.1,等等
06	单个寄存器复位
16	多个寄存器复位

对于MODBUS命令和响应方式的准确的址和数据格式,请在<http://www.modicon.com>参考由Groupe Schneider出版的MODBUS协议。

表格5.1: T100M+到其它PLC的内存映射

T100M+I/O#		OMRON	MODBUS
输入	1 to 16	IR00.0 to IR00.15	40001.1 to 40001.16
	17 to 32	IR01.0 to IR01.15	40002.1 to 40002.16
	33 to 48	IR02.0 to IR02.15	40003.1 to 40003.16
	49 to 64	IR03.0 to IR03.15	40004.1 to 40004.16
输出	1 to 16	IR16.0 to IR16.15	40017.1 to 40017.16
	17 to 32	IR17.0 to IR17.15	40018.1 to 40018.16
	33 to 48	IR18.0 to IR18.15	40019.1 to 40019.16
	49 to 64	IR19.0 to IR19.15	40020.1 to 40020.16
定时器	1 to 16	IR32.0 to IR32.15	40033.1 to 40033.16
	17 to 32	IR33.0 to IR33.15	40034.1 to 40034.16
	33 to 48	IR34.0 to IR34.15	40035.1 to 40035.16
	49 to 64	IR35.0 to IR35.15	40036.1 to 40036.16
计数器	1 to 16	IR48.0 to IR48.15	40049.1 to 40049.16
	17 to 32	IR49.0 to IR49.15	40050.1 to 40050.16
	33 to 48	IR50.0 to IR50.15	40051.1 to 40051.16
	49 to 64	IR51.0 to IR51.15	40052.1 to 40052.16
延时器	1 to 16	IR64.0 to IR64.15	40052.1 to 40052.16
	17 to 32	IR65.0 to IR65.15	40065.1 to 40065.16

	33 to 48	IR66.0 to IR66.15	40066.1 to 40066.16
	49 to 64	IR67.0 to IR67.15	40067.1 to 40067.16
	68 to 80	IR68.0 to IR68.15	40068.1 to 40068.16
	81 to 96	IR69.0 to IR69.15	40069.1 to 40069.16
	97 to 112	IR70.0 to IR70.15	40070.1 to 40070.16
	113 to 128	IR71.0 to IR71.15	40071.1 to 40071.16
	129 to 144	IR72.0 to IR72.15	40072.1 to 40072.16
	145 to 160	IR73.0 to IR73.15	40073.1 to 40073.16
	161 to 176	IR74.0 to IR74.15	40074.1 to 40074.16
	177 to 192	IR75.0 to IR75.15	40075.1 to 40075.16
	193 to 208	IR76.0 to IR76.15	40076.1 to 40076.16
	209 to 224	IR77.0 to IR77.15	40077.1 to 40077.16
	225 to 240	IR78.0 to IR78.15	40078.1 to 40078.16
	241 to 256	IR79.0 to IR79.15	40079.1 to 40079.16

\* MODBUS 是Groupe Schneider 的一个注册商标.

OMRON 是OMRON 公司的一个注册商标.

T100M+ 变量		OMRON	MODBUS
定时器数值	1 to 64	IR128 to IR191	40129 to 40192
计数器数值	1 to 64	IR256 to IR319	40257 to 40320
时钟	TIME[1] TIME[2] TIME[3]	IR512 IR513 IR514	40513 40514 40515
日期	DATE[1] DATE[2] DATE[3] DATE[4]	IR516 IR517 IR518 IR519	40517 40518 40519 40520
数据内存	DM[1] DM[2] ..... DM[4000]	DM[1] DM[2] ..... DM[4000]	41001 41002 ..... 45000

## 5.2 支持MODBUS RTU 协议

改进版的D版T100MD1616+也支持MODBUS和RTU协议。ASCII和RTU协议之间的不同是后者直接传送二进制数据而不是将二进制数据的单字节转换成二个ASCII字符。信息结构在COMM端口收到的字符之间由3.5倍字符空闲时间间隔决定。其它函数代码和内存映射都与MODBUSASCII协议相同。表格5.1因此也可应用于MODBUS RTU协议：

MODBUS RTU 有如下的命令和响应格式：

起始	地址	功能	数据	CRC 16	结束
3.5倍 字符空 闲时间	1字节	1字节	#字节	2字节	3.5倍字 符空闲 时间

支持下列函数代码：

02	读输入状态
03	读保持寄存器
05	激活单个线圈。Coil#1=40001.1, Coil#1=40001.2…… Coil#17=40002.1,等等
06	单个寄存器复位
16	多个寄存器复位

### 5.3 支持OMRON主连接命令

命令形式	开头	支持级别
a) TEST	TS	完全支持
b) STATUS READ	MS	完全支持
c) ERROR Read	MF	Dummy (always good)
d) IR Area READ	RR	完全支持 (0000 to 1000)
e) HR, AR, LR Area & TC Status READ	RH	Dummy (always returns '0000')
f) DM AREA READ	RD	完全支持
g) PV READ	RC	Dummy (always returns '0000')
h) Status Write	SC	Dummy (always OK)
i) IR Area WRITE	WR	完全支持
j) HR, AR, LR Area & TC Status WRITE	WH, WJ, WL, WG	Dummy (always OK)
k) DM Area WRITE	WD	完全支持 (from DM0001-DM4000)
l) FORCED SET	KSCIO KRCIO	仅完全支持IR区域 Dummy for other areas.
m) Registered I/O Read for Channel or Bit	QQMR/ QQIR	仅完全支持IR和DM Dummy for other areas (always 0000)

至于每个OMRON命令的细节描述和响应格式，请查阅由OMRON公司出版C20H/C28H/C40HPLC操作指南。

### 5.4 应用实例：与SCADA软件接口

SCADA软件或MMI系统(如LCD触摸面板)通常使用面向对象的程序设计方法。例如开关，指示灯或表，等等；都可以从库中选出并赋予具体I/O或PLC内部的数据地址。当设计SCADA系统时首先需要定义PLC类型。可从MODBUSASCII或MODBUSRTU或OMRONC20H中任选一个。一旦图型对象建立，则需要编辑其连接，此时根据PLC类型的内存映射表将有一选择表格被显示。

例子1: 将PLC的输入#9连接一盏指示器灯.

根据OMRON协议需将开关编程与IR0.8连接。然而，如PLC定义为MODBUS类型者指示器灯应该是连接到地址40001.9.(参阅表格5.1)。其它情况下就不需要了解协议本身的实际命令格式。因为SCADA软件的本身自动产生要求的命令来获得所选取对象的输入地址。

例子2: 在SCADA以柱状图形式显示ADC#3读数。

由于在表格5.1中ADC#3的数据并不直接映射到MODBUS或OMRON。需在用户函数中添加一语句将ADC#3拷贝至数据内存，如：

DM[100] = ADC(3)

如使用OMRON协议此时可在SCADA屏幕上编程将柱状图连接到DM[100]。对于MODBUS协议对象则应该连接到地址：41100如表格5.1所示。

## 5.5 使用T100M+PLC作为主MODBUS

T100M+PLC不仅仅作为MODBUS的从属设备支持MODBUS协议。在TRiLOGI 4.1版本中可使用READMODBUS和WRITEMODBUS命令来发送MODBUSASCII命令读取任何其它T100M+PLC 或任何第三方MODBUS的从属设备。

注意当使用READMODBUS或WRITEMODBUS命令时，表格5.1中的40001地址应解释为0000，40002应为地址0001的……41001应为地址1000的，等等。这是根据MODBUS协议中规定的。MODICON为MODBUS命令定义了零偏移量地址，在I/O定义中I/O通道地址都是从40001开始，因此它们一一对应。为了保持与MODBUS规定的兼容性，则必须遵守此项定义。

### M+PLC作为主MODBUS RTU

新的D改进版T100MD1616+可作为主MODBUS，RTU。同样READMODBUS和WRITEMODBUS命令也可以被用来发送并且接收MODBUS，RTU命令。不同的是COMM端口数字需增加10（十进制），处理器根据此来区别希望使用MODBUS，RTU而不是ASCII与从设备通讯。如：在COMM1上使用RTU命令应该指定为端口#11，在COMM3上使用RTU命令指定应为端口#13。如：语句DM[10]=READMODBUS(13,8,16)，将由COMM3经ID=08的从设备读寄存器#16的内容。这寄存器对应MODICON地址为40017，是从设备PLC的输出[1]。

支持MODBUS，RTU的能力极大地扩展了T100MD1616+使用外设的类型。这样就可以使用更多的现有设备和第三方RTU设备，来扩展PLC能力，使T100MD1616+成为真正意义上的超级PLC!

---

---

## 第六章 用EMIT 3.0与Internet 连接

### 6.1 EMIT 3.0介绍

emWare的EMIT™软件(嵌入式互连网络技术)是一种系统的软件解法,它可以把廉价但功能强大的PC作为一种介质将许多设备连接到是因特网。EMIT定义了一套使用RS232/RS485通信协议,允许任何支持这协议的电子设备经因特网远读取。EMIT减轻了设备由于连接到是因特网而造成的CPU紧张,内存短缺, TCP/IP堆栈的混乱。

在将设备连接到因特网上, "emGateway"软件起着重要作用,同时它必须在运行Windows95/98/NT上。而PC才是真正地通过局域网将它连接到因特网。emGateway执行地使用由PC建立的internet/intranet基座在设备和一远程因特网中传送数据,它可以使用Netscape4或InternetExplore4.作为网络浏览器。因为emGateway在后台运行,公司的网络以内的任何PC都能使用,因此不要求指定的PC运行。

除了传送杂乱无章的数据外, EMIT利用Java技术使OEM在一个Java小程序的表单创造一富有的图形用户界面。当一个远程的浏览器存取设备时,Java小程序将经因特网传送到浏览器。无论在哪里,一个图形界面将展示于用户之前。由于Java小程序在浏览器上运行(客户)而不是在服务器上运行,因此客户机器速度快则变化的响应也快。

一旦Java小程序传送到浏览器,在浏览器和设备之间仅是二进制数据的交换,因此对变化可以有快速响应。与许多以SCADA为基础internet存取方法相比,EMIT方法将更简洁并便宜。设备不仅能从公司内部LAN中任何节点并可以通过internet从任何地方存取,同时在每一节点终端也不需要安装或用户许可就可访问到设备(不同于SCADA)。最终用户仅仅需要为emGateway软件付emWare的许可费用就可以了。

---

---

### 6.2 T100M+ PLCs 和 EMIT 3.0

EMIT3.0是一完整的软件包,包括汇编和以为C为源代码"emMicro",并允许设计员把emMicro代码合并到他们的产品中的内置系统。与emMicro相结合的设备将使用"emGateway"来通讯。然而将"emMicro"代码与单个设备的结合要求程序员有完整的硬件知识及对微控制器结构的深入理解。由于开发时间长,因此比较适用于产量大的OEM,如:冷冻机,空调机和其他家用电器等。

对于中小批量者, emMicroenabled的产品的开发成本就显得过高。但Tri国际研究中心正致力于解决开发工作的难题,以使OEM用户更容易得接收EMIT技术



TRi作法就是从emWare获得emMicro编码的使用权并且把他们合并到我们的T100MD+和T100MX+的PLC中。emMicro集成化中的所有难题已经由我们的程序开发小组解决了,因此PLC用户不必再担心emMicro的集成化问题。使用T100M+PLC的OEM设计者可集中于开发Java界面,这对于熟悉I.T技能的程序员将更有利,因为他们不再需要去熟悉内嵌的硬件知识。

Tri将EMIT使用的emVariables集成进我们的易用的TRiLOGI软件4.1版本中,并且整个系统设计得很友好。即使机器设计完工后,仍可以添加internet连接功能,而不需要对已有的工作软件作很大的改动。OEM和最端用户将非常满意于以T100MD+和T100MX+PLC设计的机器,因为它可以持续发展的。internet集成功能可在任何时候加入,而用户并不需要对此功能化额外的花费。

---

### 6.3 获得并安装EMIT 3.0

emWare公司出版的EMIT3.0SDK(软件开发套件)使OEM用户可开发EMIT - enable d产品。SDK包括emGateway软件,Symantec VisualCafeJava开发软件和emMicro源代码,等等。

如果希望试验EMIT - enabledT100MD+PLC的功能,只要能从以下列网站下载60天有效的EMIT SDK软件和30天试用版本的Symantec VisualCafe软件:

<http://www.emstore.com/freestuff.master.html>

如emWare删除了该页,则可以尝试<http://www.emstore.com>

如需获得关于EMIT的其他信息,可访问<http://www.emware.com>

请按照步序安装emGateway,同时PC上必须运行MSWindows95/98/NT,这样就可以使用一个Netscape或IE4浏览器来演示一个应用T100MD1616+PLC控制和监控程序。如成功安装则一个嵌有"em"字样的红色球形将出现在底端屏幕的右端。

如一个大'X'出现在红色球形中,则表明emGateway并未成功安装,因为TCP/IP堆栈并未初始化,(如PC并未联网可通过控制板中指定一IP地址到PC)。

---

### 6.4 安装演示文件

检查TRiLOGI的安装磁盘中是否有"INTERNET.ZIP"文件 或下载该文件从:

[www.tri-plc.com/internet.zip](http://www.tri-plc.com/internet.zip).

之后可使用Winzip软件来解压,Winzip中可选定具体目录,以下为具体解压的文件:

文件	Unzip 路径
INTERNET.PC4	C:\trilogi\tl4
DEFAULTPAGE.JAR	C:\emware\emgateway\html
DEFAULTPAGE.Java DEFAULTPAGE.VEP	C:\visualcafewde\projects\tri\

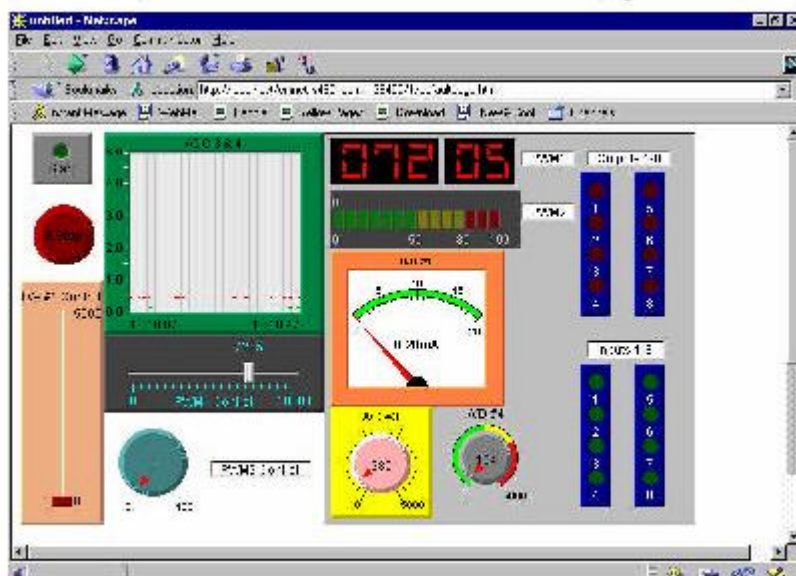
注意: 如果已装入的TRiLOGI, emGateway或VisualCafe与以上目录不同, 则需将实际的路径名代替上表。

文件"defaultpage.jar"是Java的档案文件, 它必须拷贝到C:[emware路径名]\emGateway\html\目录下以便浏览器能随时检索到它。在TRiLOGI目录中装了"INTERNET.PC4"文件后,运行TRiLOGI4.1时就可以将程序传送至T100MD1616+PLC。注意需使用"Controller>TargetAccess>WriteIDAddress"命令将你的PLC的ID地址设置到ID = 01。

## 6.5 运行 Internet 演示程序

我们将首先在本地主机上用Netscape浏览器演示程序, 同时emGateway也运行在同一PC上。如本地主机运行正常, 则可推广至intranet或internet上的其它终端试验, 只要与运行emGateway的PC实际的IP地址与本地主机相同。

- a) 将PC的串行口的COM1与T100MD1616+的COMM1连接。(注意:如果使用的是T100MX, 则需要一特殊电缆, 其DB9接口的第四针可为T100MXPLCs的光电隔离RS232接口提供外部+9V电压, 不同与TRiLOGI, emGateway并不自动提供电源至针4, 而此电源对无光电隔离T100MD+的S232接口是很必要的。
- b) 运行Netscape版本4.0浏览器, IE4或更高版本,将下列URL输入至"Location"项方框中:<http://localhost/emnet.rs485+com1+38400/1/defaultpage.htm>



- c) 如进展顺利看到"StartingJava.."的信息在浏览器左下方显示，一分钟以后如图6.1的Java小程序应出现在浏览器窗口上。
- d) 首先按"启动"按钮，PLC的输出1,3,5应点亮。按“停止”按钮则全部关闭。输出的变化将反应在屏幕指示灯上标记为"Output1-8"。
- e) 将"PWM1control"的标签滑标往右拉。在T100MD1616+PLC上观察PWM#1(输出7)红色LED。灯的亮度将随滑标移动而变化，PWM1的负载循环百分比将显示在滑标上，它与7段编码LED所显示的值相同，都用于显示PWM1的控制变量。但屏幕上output#7的指示灯并未改变，因为output#7仅表明PWM1在被使用，与output#7置ON不同。
- f) 下一步，在蓝色的旋钮上的红色指示器上单击"PWM2Control"，按住鼠标，旋转拨盘。PWM2应该根据旋钮的位置而改变灯的强度。旋钮和LED指针显示上都表明了当前设定值。
- g) 模拟输出#1可由标记"D/A#1"的滑标控制。模拟量部分需外接电源，当滑动滑标时，用万用表在模拟输出1处可测量到0-20mA电流回路。
- h) 在ADC#3，ADC#4分别连接1K电位计。当旋转电位计旋钮时，可以观察到屏幕上的图形"旋钮"也相应旋转。同时ADC#3和#4的值将连续地被记录在时间表图上。

---

---

## 6.6 编写 Java 用户自己的界面程序

emWare已拥有许多现成Java对象，如滑标,开关,指示器灯,等等。之所以称为"emObjects",因为它们是有EMIT3.0SDK的授权许可，并可以独立销售。我们大量使用了emObjects来编写演示程序。使用VisualCafe，Java开发系统的加载emObjects，就可简单编写用户自己的Java程序。用户必须按照VisualCafe和EMIT的规程，将emVariables与对象连接。对于在TBASIC中emVariables变量名及emGateway中对应的名称，可参考TBASIC参考手册第三章第4节“用EMIT3.0连接变量(Internet连接)”。如用户想建立自己的Java程序通过因特网来控制T100MD1616+，演示程序"defaultpage.java"和"INTERNET.PC4"为此提供很好样例。