

Automation Technology > 自动化技术 > 工业自动化系统 SIMATIC > 工业软件 > 标准工具 > STEP 7

#### ▲ STEP 7 -- 创建S7 程序 -- 编辑日期、时间和计数器

- \_ 什么是可修改的Julian日期？如何用STEP 7计算它？
- \_ 怎样将秒转换为天，小时，分钟和秒？
- \_ 关于“编辑日期和时间”的信息
- \_ 如何测量时间？
- \_ 如何编程实现读取定时器从启动到停止的剩余时间？
- \_ 如何通过工业以太网来设置S7-300的时间？
- \_ 如何从Step7的IEC库中导入IEC标准功能FC3, FC6, FC7和FC8？
- \_ 如何使用IEC 功能读取并且使用CPU的日期和时间？
- \_ 如何同步各 CPU 时钟？
- \_ 在 -2147483648 到 +2147483647 区域之间，如何向上计数和向下计数？
- \_ 如何用 **SFC1 (READ\_CLK)**读取时间和日期以及如何用 **SFC0 (SET\_CLK)** 设定系统时间？
- \_ 如何在不使用 “S7 Timer”的条件下，为 “S7”创建一个定时器？

如何用 SFC1 (READ\_CLK)读取时间和日期以及如何用 SFC0 (SET\_CLK) 设定系统时间？

#### 用SFC1 (READ\_CLK) 读取时间和日期的使用说明

使用系统功能 SFC1 (READ\_CLK) 可以读出CPU的系统时间。在系统功能 SFC1 中的输出参数“CDT”接收的时间和日期的格式为“DATE\_AND\_TIME”。具有“DATE\_AND\_TIME”数据类型的时间和日期是以 BCD 码的格式存储在8个字节里。这种数据类型的转换范围是：

- DT#1990-1-1-0:0:0.0 到 DT#2089-12-31-23:59:59.999

下表给出了实例表示2004年8月5日，星期四，8点12分5.250秒。并且给出了每个字节所包含的时间和日期数据的内容。

字节	内容	例子
0	年	B#16#04
1	月	B#16#08
2	日	B#16#05
3	小时	B#16#08
4	分钟	B#16#012
5	秒	B#16#05
6	毫秒的百位和十位数值	B#16#25
7 (高4位)	毫秒的个位数值	B#16#05 中的BCD 码 “0”
7 (低四位)	星期： 1: 星期日， 2: 星期一， 3: 星期二， 4: 星期三， 5: 星期四， 6: 星期五， 7: 星期六	B#16#05 中的BCD 码 “5”

更多的关于“DATE\_AND\_TIME”数据类型的信息可以通过 STEP 7 在线帮助获得，如下：

- DATE\_AND\_TIME 数据类型的格式

下面的 STEP 7 程序（图 01）是通过地址寄存器 AR1 读时间和日期。FC1 中调用系统功能 SFC1 读出 CPU 的当前时间。小时和分钟可以显示在数字显示器上。

```

FC0 : Timer
Network 1 :
CALL "DATE_TIME"
RET_VAL:=RET
CPU :=WDate_Time //Temp. variable of type DT

Network 2 :
LAP: P##Date_Time
L B [AR1,P#0]
T "Year"
L B [AR1,P#1]
T "Month"
L B [AR1,P#2]
T "Day"
L B [AR1,P#3]
T "Hour"
L B [AR1,P#4]
T "Minutes"
L B [AR1,P#5]
T "Seconds"
L W#16#FFF0
L W [AR1,P#6]
WU "Milli Seconds"
L W#16#F
L B [AR1,P#7]
WU "Day of Week"
T "Day of Week"
CLR

```

图 01

说明:

### “LAR1 P##Date\_Time”

通过上面指令将临时变量“Date\_Time”的起始地址被装载到地址寄存器 AR1 中。通过间接寻址装载命令可以分别访问年、月、日、小时等。

例如:

“L B[AR1, P#3.0] //Read hours” (读取小时数)

读出的数据赋值给输出字节用于显示。

T "Hour" // for AB3 (赋给输出字节 3)

由于字节 7 中的高四位是用于表示毫秒的个位数值的，低四位是用于表示星期的，鉴于显示的原因，如图 01 例程中，使用语句“L W#16#FFF0”和“L B#16#F”分别通过字与（“UW”即“AW”）的操作指令将 AW6 和 AB8 输出地址中相应的值屏蔽掉。否则图 02 中 AW6 的值将显示为 W#16#7853（而 3 是表示星期数的），同样 AB8 的值将显示 B#16#53（5 表示毫秒值）。

在“监视/修改变量”中您可以显示输出字节的数据。图 02 为“监视/修改变量”中对应 2004年8月24日，星期二，14点3分19.785秒的显示示例。

Address	Symbol	Display format	Status value	Modify value
AB 0	"Year"	HEX	00160004	
AB 1	"Month"	HEX	00160008	
AB 2	"Day"	HEX	00160024	
AB 3	"Hour"	HEX	00160014	
AB 4	"Minutes"	HEX	00160003	
AB 5	"Seconds"	HEX	00160019	
AW 6	"Milli Seconds"	HEX	16160785	
AB 8	"Day of Week"	HEX	00160003	

图 02

### 用 SFC0 (SET\_CLK)设定系统时钟的使用说明

使用 SFC0“SET\_CLK”（设定系统时钟）可以设定 CPU 的时间和日期。CPU 的时钟将以设定的时间和日期运行。用以下语句:

“LAR1 P##Date\_Time”

临时变量“Date\_Time”的起始地址被装载到地址寄存器 AR1 中。通过间接寻址指令将设定数据从功能块的输入变量（类型：字节）装载并传输到变量“DATE\_AND\_TIME”中的年、月、日、小时等各自的字节中。

例如:

## L #Year

//输入变量，类型：字节

//输入变量“年”如，“B#16#05”表示 2005.

## T B[AR1,P#0.0]

//输入变量“Year”的内容传送到字节 0

//“DATE\_AND\_TIME”变量的字节0。

在例程中描述的如上您所看到的数据类型“DATE\_AND\_TIME”的格式。更多的关于“DATE\_AND\_TIME”数据类型的信息可以通过 STEP 7 在线帮助获得，如下：

- DATE\_AND\_TIME 数据类型的格式

随后调用 SFC0，在输入端“PDT”参数化的临时变量“Date\_Time”的数值被传送到 SFC0。

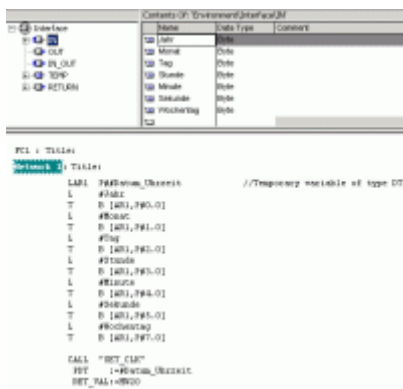


图 03

在“监视/修改变量”中您可以指定时间和日期的值。输入的年、月、日等是在 OB1 中使用 MB0, MB1, MB2 等参数化的。图 04 给出了例子：2005年1月3日，星期一，16点26分40.20秒。

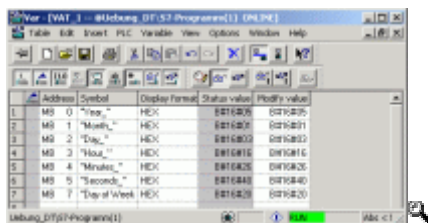


图 04

图 01

FC2 : Title:

**Network 1** : Title:

```
CALL "READ_CLK"
RET_VAL:=MW30
CDT      :=#Date_Time           //Temp. variable of type DT
```

**Network 2** : Title:

```
LAR1 P##Date_Time
L    B [AR1,P#0.0]
T    "Year"           AB0
L    B [AR1,P#1.0]
T    "Month"         AB1
L    B [AR1,P#2.0]
T    "Day"           AB2
L    B [AR1,P#3.0]
T    "Hour"          AB3
L    B [AR1,P#4.0]
T    "Minutes"       AB4
L    B [AR1,P#5.0]
T    "Seconds"       AB5

L    W#16#FFFO
L    W [AR1,P#6.0]
UW
T    "Milli Seconds" AW6

L    B#16#F
L    B [AR1,P#7.0]
UW
T    "Day of Week"   AB8
CLR
```

图 02

	Address	Symbol	Display format	Status value	Modify valu
1	AB 0	"Year"	HEX	B#16#04	
2	AB 1	"Month"	HEX	B#16#08	
3	AB 2	"Day"	HEX	B#16#24	
4	AB 3	"Hour"	HEX	B#16#14	
5	AB 4	"Minutes"	HEX	B#16#03	
6	AB 5	"Seconds"	HEX	B#16#19	
7	AW 6	"Milli Seconds"	HEX	W#16#7850	
8	AB 8	"Day of Week"	HEX	B#16#03	
9					

Uebung\_DT\S7-Programm(1) RUN

图 03

		Contents Of: 'Environment\Interface\IN'		
		Name	Data Type	Comment
[-] Interface	+	Jahr	Byte	
		Monat	Byte	
		Tag	Byte	
		Stunde	Byte	
		Minute	Byte	
		Sekunde	Byte	
		Wochentag	Byte	

FC1 : Title:

**Network 1**: Title:

```

LAR1 P##Datum_Uhrzeit           //Temporary variable of type DT
L   #Jahr
T   B [AR1,P#0.0]
L   #Monat
T   B [AR1,P#1.0]
L   #Tag
T   B [AR1,P#2.0]
L   #Stunde
T   B [AR1,P#3.0]
L   #Minute
T   B [AR1,P#4.0]
L   #Sekunde
T   B [AR1,P#5.0]
L   #Wochentag
T   B [AR1,P#7.0]

CALL "SET_CLK"
PDT   :=#Datum_Uhrzeit
RET_VAL:=MW20

```

图 04

The screenshot shows the SIMATIC Manager interface with a table of variable declarations. The table has columns for Address, Symbol, Display format, Status value, and Modify value. The data is as follows:

Address	Symbol	Display format	Status value	Modify value
MB 0	"Year_"	HEX	B#16#05	B#16#05
MB 1	"Month_"	HEX	B#16#01	B#16#01
MB 2	"Day_"	HEX	B#16#03	B#16#03
MB 3	"Hour_"	HEX	B#16#16	B#16#16
MB 4	"Minutes_"	HEX	B#16#26	B#16#26
MB 5	"Seconds_"	HEX	B#16#40	B#16#40
MB 7	"Day of Week"	HEX	B#16#20	B#16#20

The interface also shows a menu bar (Table, Edit, Insert, PLC, Variable, View, Options, Window, Help) and a toolbar with various icons. At the bottom, the status bar indicates 'RUN' mode.

条目号:21222026 日期:2007-03-08