

SysLibCallback.lib库..... 2

SysLibCallback.lib库

注意：此库文件的应用依赖于运行应用程序的目标系统。请参考文 SysLibs_Overview.pdf。请注意，在 Windows CE 下运行 SysLibDir.lib 功能可能会出现不安全线程。

这个库提供功能函数 SysCallbackRegister 和 SysCallbackUnregister，分别用来激活 callback 功能和 runtime 事件的定义。

两个功能函数都是布尔型的，当必需的 callback 功能被成功的注册或注消时，它们会分别返回 TRUE，且整个过程都是实时的。

这callback功能的原型必须如下所示：

```
FUNCTION Callback : DWORD

VAR_INPUT

    dwEvent:DWORD; // 事件

    dwFilter:DWORD; // 过滤

    dwOwner:DWORD; // 来源

END_VAR
```

注意对于 RISC 和 Motorola 68k 的目标系统：callback 功能的名字必须以“callback”为开始。

当功能函数被用来注册或注消功能调用时，都将采用下面的参数接口描述：

输入变量	数据类型	描述
iPOUIndex	INT	要用来注册/注消而调用 callback 功能的 POU 的指针，这个指针必须在以前通过帮助操作 INDEXOF(<<function name>) 命令已经获得。
Event	RTS_EVENT	被 callback 功能呼叫的 runtime 事件，其详细说明在 RTS_EVENT 的列举中。

RTS_EVENT 列举定义如下：

```
TYPE RTS_EVENT :
```

```
(
```

```
    EVENT_ALL,
```

(* 一般事件 *)

EVENT_START,
EVENT_STOP,
EVENT_BEFORE_RESET,
EVENT_AFTER_RESET,
EVENT_SHUTDOWN,

(* 实时运行产生的例外 *)

EVENT_EXCPT_CYCLETIME_OVERFLOW,	(* 循环时间溢出 *)
EVENT_EXCPT_WATCHDOG,	(* IEC—任务的软看门狗终止 *)
EVENT_EXCPT_HARDWARE_WATCHDOG,	(* 硬件看门狗终止。全局软件错误。 *)
EVENT_EXCPT_FIELDBUS,	(* 发生 Fieldbus 错误 *)
EVENT_EXCPT_IOUPDATE,	(* IO—更新错误 *)

(* 系统产生的例外 *)

EVENT_EXCPT_ILLEGAL_INSTRUCTION,	(* 不合语法的指令 *)
EVENT_EXCPT_ACCESS_VIOLATION,	(* 禁止访问 *)
EVENT_EXCPT_PRIV_INSTRUCTION,	(* 特别说明 *)
EVENT_EXCPT_IN_PAGE_ERROR,	(* 页面错误 *)
EVENT_EXCPT_STACK_OVERFLOW,	(* 栈溢出 *)
EVENT_EXCPT_MISALIGNMENT,	(* 数据类型未校准 *)
EVENT_EXCPT_ARRAYBOUNDS,	(* 数组超出范围 *)
EVENT_EXCPT_DIVIDEBYZERO,	(* 被零点分开 *)
EVENT_EXCPT_OVERFLOW,	(* 溢出 *)
EVENT_EXCPT_NONCONTINUABLE,	(* 不可连续 *)
EVENT_EXCPT_NO_FPU_AVAILABLE,	(* FPU: 没有 FPU 可用 *)
EVENT_EXCPT_FPU_ERROR,	(* FPU: 未指明的错误 *)
EVENT_EXCPT_FPU_DENORMAL_OPERAND,	(* FPU: 正规的操作数 *)
EVENT_EXCPT_FPU_DIVIDEBYZERO,	(* FPU: 被零点分开 *)
EVENT_EXCPT_FPU_INVALID_OPERATION,	(* FPU: 无效的操作 *)
EVENT_EXCPT_FPU_OVERFLOW,	(* FPU: 溢出 *)
EVENT_EXCPT_FPU_STACK_CHECK,	(* FPU: 检查栈 *)

(* IO 事件 *)

EVENT_AFTER_READING_INPUTS,
EVENT_BEFORE_WRITING_OUTPUTS,

```

(* 混杂事件 *)
EVENT_TIMER,                                     (* 标记时间表 (计时器中断) *)
EVENT_DEBUG_LOOP,                               (* 循环断点处的调试 *)

(* 在线服务 *)
EVENT_ONLINE_SERVICES_BEGIN := 500,
EVENT_LOGIN,
EVENT_CUSTOM_SERVICES,

(* 中断 *)
EVENT_INT_0:=1000,
EVENT_INT_1,
EVENT_INT_2,
EVENT_INT_3,
EVENT_INT_4,
EVENT_INT_5,
EVENT_INT_6,
EVENT_INT_7,
EVENT_INT_8,
EVENT_INT_9,
EVENT_INT_10,
EVENT_INT_11,
EVENT_INT_12,
EVENT_INT_13,
EVENT_INT_14,
EVENT_INT_15,
EVENT_INT_255:=1255,

EVENT_MAX

);

END_TYPE

```

