

Библиотека SysLibCallback.lib

Данная библиотека позволяет активировать пользовательские функции обратного вызова для событий системы исполнения. Она содержит функции SysCallbackRegister и SysCallbackUnregister.

Обе функции имеют тип BOOL и возвращают TRUE, если соответствующая функция обратного вызова успешно зарегистрирована или отключена, соответственно. Функции выполняются синхронно.

Прототип функции обратного вызова должен выглядеть так:

FUNCTION Callback : DWORD

VAR_INPUT

```

    dwEvent: DWORD;    (* Событие *)
    dwFilter: DWORD;    (* Фильтр *)
    dwOwner: DWORD;    (* Источник *)
  
```

END_VAR

Внимание: в целевых платформах RISC и Motorola 68K имя функции обратного вызова должны начинаться с „callback“!

Библиотечные функции SysCallbackRegister и SysCallbackUnregister имеют одинаковый набор параметров:

Входные переменные	Тип данных	Описание
iPOUIndex	INT	POU индекс функции обратного вызова, может быть получен при помощи оператора INDEXOF(<имя функции >).
Event	RTS_EVENT	Событие в системе исполнения, по которому должна вызываться callback-функция. Определяется значением перечисления RTS_EVENT, определенного в данной библиотеке (см. ниже).

Перечисление RTS_EVENT определено следующим образом:

TYPE RTS_EVENT :

(

EVENT_ALL,

(* Общие события *)

EVENT_START,

EVENT_STOP,

EVENT_BEFORE_RESET,

EVENT_AFTER_RESET,

EVENT_SHUTDOWN,

(* Исключения системы исполнения *)

EVENT_EXCPT_CYCLETIME_OVERFLOW,
EVENT_EXCPT_WATCHDOG,
EVENT_EXCPT_HARDWARE_WATCHDOG,
EVENT_EXCPT_FIELDBUS,
EVENT_EXCPT_IOWUPDATE,

(* Переполнение времени цикла *)
(* Сработал программный сторожевой таймер МЭК задачи *)
(* Сработал аппаратный сторожевой таймер. Фатальная ошибка. *)
(* Ошибка Fieldbus *)
(* Ошибка обновления входов-выходов *)

(*Исключения выполнения *)

EVENT_EXCPT_ILLEGAL_INSTRUCTION,
EVENT_EXCPT_ACCESS_VIOLATION,
EVENT_EXCPT_PRIV_INSTRUCTION,
EVENT_EXCPT_IN_PAGE_ERROR,
EVENT_EXCPT_STACK_OVERFLOW,
EVENT_EXCPT_MISALIGNMENT,
EVENT_EXCPT_ARRAYBOUNDS,
EVENT_EXCPT_DIVIDEBYZERO,
EVENT_EXCPT_OVERFLOW,
EVENT_EXCPT_NONCONTINUABLE,
EVENT_EXCPT_NO_FPU_AVAILABLE,
EVENT_EXCPT_FPU_ERROR,
EVENT_EXCPT_FPU_DENORMAL_OPERAND,
EVENT_EXCPT_FPU_DIVIDEBYZERO,
EVENT_EXCPT_FPU_INVALID_OPERATION,
EVENT_EXCPT_FPU_OVERFLOW,
EVENT_EXCPT_FPU_STACK_CHECK,

(* Недопустимая инструкция *)
(* Ошибка доступа *)
(* Привилегированная операция *)
(* Ошибочная страница *)
(* Переполнение стека *)
(* Ошибка размещения по типу данных *)
(* Выход за границы массива *)
(* Деление на 0 *)
(* Переполнение *)
(* Авария системы, продолжение работы невозможно *)
(* Мат. сопроцессор (FPU) недоступен *)
(* FPU: Неопределенная ошибка *)
(* FPU: Ошибочный операнд *)
(* FPU: Деление на 0 *)
(* FPU: Ошибочная операция *)
(* FPU: Переполнение*)
(* FPU: Перегрузка стека *)

(* События входов-выходов *)

EVENT_AFTER_READING_INPUTS,
EVENT_BEFORE_WRITING_OUTPUTS,

(* Прочие события *)

EVENT_TIMER, (* Прерывание по таймеру*)
EVENT_DEBUG_LOOP, (* Отладка по точке останова *)

(* Онлайн сервисы *)

EVENT_ONLINE_SERVICES_BEGIN := 500,
EVENT_LOGIN,
EVENT_CUSTOM_SERVICES,

(* Прерывания *)

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