The Library SysLibIECTasks.lib

If the target system supports the functionality, this library can be used to call information on the configuration of IEC tasks. The execution is synchronous. (To create, delete, prioritize, stop and restart a task you can use the library SysLibTasks.lib.)

The functions for getting information:

- SysIECTaskGetConfig

- SysIECTaskGetInfo

Auxiliary function:

- SysIECGetFctPointer

- SysIECTaskResetEvent

SysIECTaskGetConfig

This function of type BOOL serves to retrieve the configuration parameters of an IEC task.

The task is addressed by its name resp. its index, which it has got assigned in the CoDeSys task configuration.The structure SysIECTaskConfEntry contains all parameters which are used in the task configuration.

As soon as the task has been found, TRUE will be returned, otherwise FALSE.

Input Variable	Data Type	Description
udiTaskId	UDINT	Task Id (Index in the CoDeSys task configuration); (optional, you also can use *stTaskName*)
pTaskInfo	POINTER TO SYSIECTASKCONFENTRY	Information on the task configuration (structure, see below)

Components of structure **SysIECTaskConfEntry**:

byTaskNr	USINT	Task Number
byPriority	USINT	Priority, see dialog 'Taskattributes' in CoDeSys
lInterval	DINT	Interval of cyclic tasks, see dialog 'Taskattributes in CoDeSys; (in this case *ldrEvent* has an invalid entry)
ldrEvent	LDATAREF_TYPE	Event for an event controlled task, see dialog 'Taskattributes' in CoDeSys; Structure *LdataRef_Type* see below; in this case Interval =0;
wIndex	UINT	Index of the POU which is called by the task (matches with the index retrieved by INDEXOF())
uiNameLen	UDINT	Length of the task name
szName	STRING(80)	Name of the task, see dialog 'Taskattributes' in CoDeSys

Components of structure **LdataRef_Type**:

POURef	UINT	POU-ID of the event variable
Offset	UDINT	POU-ID of the event variable
Size	UDINT	POU-ID of the event variable

SysIECTaskGetInfo

This function returns the current time values of an IEC task. The task is identified by the task name or by the index, it has got in the task configuration.

As soon as the task has been found, TRUE will be returned, otherwise FALSE.

Input Variable	Data type	Description
stTaskName	STRING	Name of the task (optionally you can use the input variable UdiTaskId, see below)
pTaskInfo	POINTER TO SYSIECTASKINFO	current data of the IEC task (structure, see below)

Components of structure **SysIECTaskInfo**:

dwCount	DWORD	Number of cycles since start of task
dwCycleTime	DWORD	Current cycle time
dwCycleTimeMin	DWORD	Minimum cycle time
dwCycleTimeMax	DWORD	Maximum cycle time
dwCycleTimeAvg	DWORD	Average cycle time
wStatus	WORD	Current status of the PLC: 0 = RUN, 1 = STOP
wMode	WORD	Current task mode: 1 = running; 2 = stopped (maybe by a runtime error)

SysIECGetFctPointer

This auxiliary function of type DWORD returns a function pointer, which is required as input parameter for the function SysTaskCreate which is used to create a new task (see library *SysLibTasks.lib*).

The function requires as an input parameter the internal index of the POU, which should be called by the task. This index can be acquired with the aid of the INDEXOF operator.

Input Variable	Data type	Description
wIndexOf	WORD	internal index of the POU, which is to be called by the task

SysIECTaskResetEvent

This auxiliary function of type BOOL resets the event variable of an event triggered IEC task.

The function has no input parameter. It is working on the current task. It returns TRUE in case of success, otherwise 0 (e.g. if the task is not an event triggered task).

The function sets the BOOLean IEC-variable, which is used as an event, to FALSE, and the internal flag of the CoDeSys runtime system task management to 0.

So it is achieved that a rising edge of the event variable will be regarded at the next cycle of the scheduler.