The Library SysLibPLCConfig.lib

这个库支持读取PLC的配置数据。这些数据在应用程序下载时也下到了控制中，runtime系统把它们写到了结构体中。这个库提供的功能块来获得指向这些结构体的指针。

由于runtime系统最初的结构体的指针已经存在，见下面：

- 结构体（指向sub-elements的指针 ）可能没有被修改。
- 要是结构体默认的参数值被修改，这将受到影响。

要是目标系统支持的话，下面这些功能块可以用（执行是实时的)：

- CfgCCGetError
- CfgCCGetHeader
- CfgCCGetRootModule
- CfgCCGetRootModuleByModuleId
- CfgCCGetRootModuleByNodeId

CfgCCGetError

注意：通常不在runtime系统下执行。错误代码永远是0。

这个功能块提供在下载配置时出现的错误信息。

Structure CCLoadError

This structure (Library SysLibPLCCOnfig.lib) provides information on the last error during the download of the PLC configuration data. It is accessed by function CfgCCGetError.

The components:

Variable	Data type	Description
ulLastError	UDINT	Error code of the last error
ulAddInfo1	UDINT	According to ulLastError, the meaning changes.
ulAddInfo2	UDINT	According to ulLastError, the meaning changes.
szLastError	STRING(32)	Last error message. A possibility to make debugging easier

CfgCCGetHeader

这个功能块返回一个指向PLC配置CCHeader的头结构体的指针。

Structure CCHeader

This structure (Library SysLibPLCCOnfig.lib) provides information on the header structure for the PLC configuration which has been loaded on the target system. It can be accessed by function CfgCCGetHeader:

Variable	Data type	Description
szTag	STRING(10)	zero-terminated STRING "CommConf"
cByteOrder	BYTE	The file data are in Intel ('I') or Motorola format ('M') vor
ulSize	UDINT	Size of the following data
lVersion	UDINT	Version number of the file

CfgCCGetRootModule

这个功能块提供PLC配置的根模块的信息。它返回一个指向CCModule结构体的指针。

Structure CCModule

This structure (Library SysLibPLCCOnfig.lib) provides information on the module, which e.g. can be accessd via function CfgCCGetRootModule.

Variable	Data type	Description
ucEntryTag	BYTE	'M' = Module
ucDummy1	BYTE	
ucDummy3		
ulModuleId		Id of the module given in the configuration file *.cfg
sModuleNumber		Number of the module in the parent module (-1 if root)

usModuleTag		Describes the kind of the module (0=3S-Module, 1=DP-Master, 2=DP-Slave, 3=CAN-Master, 4=CAN-Slave, 5=DP-SingleSlave)
byDeviceDriver	BYTE	The module needs a device driver (0=FALSE, 1=TRUE)
ucDummy4	BYTE	
ucDummy5	BYTE	
ucDummy6	BYTE	
ulNodeId	UDINT	NodeId of the module
byDefinedWithStruct:	BYTE	The module was defined with a structure (0=FALSE, 1=TRUE)
ucDummy7	BYTE	
ucDummy8	BYTE	
ucDummy9	BYTE	
ulBitOffsetInput	UDINT	Offset of the modules input area
ulBitSizeInput	UDINT	Size of the modules input area in bit
ulBitOffsetOutput	UDINT	Offset of the modules output area
ulBitSizeOutput	UDINT	Size of the modules output area in bit
ulRefIdCommonDiag	UDINT	RefId of the modules common diagnosis area
ulBitOffsetCommonDiag		Offset of the modules common diagnosis area
ulBitSizeDiag	UDINT	Size of the modules diagnosis area in bit
usParameterCount	UINT	Number of parameters
usDummy	UINT	

ppccpModuleParams	POINTER TO POINTER TO ccParam	<ccParam [0..usParameterCount]> a pointer to an array of pointers to CCModuleParam-structures. (Definition of structure CCParam see below). Dereferencing the pointer with ppccpModuleParams^ gives you the pointer to the first parameter structure. (ppccpModuleParams+4)^ gives you the pointer to the next parameter structure. See also comment (*Read pointer to parameters *) in example project.
ulSizeOfSpecificData	UDINT	Size in bytes of the module specific data
pModuleData	POINTER TO BYTE	<MODULE_SPECIFIC_DATA> Here the data, according to usModuleTag is located: pModuleData is possible to be a pointer to PBSlave, CANSlave, PBMaster, PBSlave, PBSingleSlave, see definitions below.
usChannelCount:	UINT	Number of configured channels
usModuleCount	UINT	Number of configured modules
In the following the Channels and Modules of this Module in the configured order are located! (DP-Slaves are ordered by the stationnumber!) This means, it is possible that another CCModule structure is inserted here.		
ppcccChannels	POINTER TO POINTER TO ccChannel;	<ccChannel [0..usChannelCount]> Definition of structure CCChannel see below * Dereferencing the pointer with ppccpChannels^ gives you the pointer to the first parameter structure. (ppccpChannels+4)^ gives you

		the pointer to the next parameter structure. See also comment ″(*Read pointer to parameters *)″ in example project.
ppccmSubModules	POINTER TO POINTER TO BYTE	<ccModule [0..usModuleCount]> Points to an array of variables of type POINTER TO ccModule. To view the contents, you have to assign the value to a variable of type ″POINTER TO CCModule″. Definition of structure CCModule see below. Dereferencing the pointer with ppccpSubModules^ gives you the pointer to the first parameter structure. (ppccpSubModules+4)^ gives you the pointer to the next parameter structure. See also comment ″(*Read pointer to parameters *)″ in example project.

CfgCCGetRootModuleByModuleId

这个功能块根据 module Id 来提供当前应用 PLC 配置的根 module 信息。这个 module Id 是通过 Id 来登陆配置文件来定义的。参考 PLC_Configuration_E.pdf 文档。

这个功能块返回一个指向 CCModule 结构体的指针。(参考上面,CfgCCGetRootModule 功能块。)

变量	数据类型	描述
ulModuleId	UDINT	根 module 的 module Id。

CfgCCGetRootModuleByNodeId

这个功能块根据节点 Id 来提供当前应用 PLC 配置的根 module 信息。这 module 的节点 Id 通常是 PLC 配置 module 位置的结果。细节请参考 PLC_Configuration_E.pdf 文档。这个

功能块返回一个指向 CCModule 结构体的指针。（参考上面，CfgCCGetRootModule 功能块。）

变量	数据类型	描述
ulNodeId	UDINT	根 module 的节点 Id。