

Обработка событий в приложениях и библиотеках

CODESYS Users Conference 2016

1

Что такое событие в 61131-PLC?

2

Обработка событий в приложении

3

Обработка событий в библиотеках

4

Заключение



Что такое событие в 61131-PLC?

В программировании событие - это сообщение программного обеспечения (либо его части), которое указывает, что произошло.

- **Смена состояний приложений**
 - Online change
 - Download
 - Start
 - Stop

- **Смена фаз циклов ПЛК**
 - Before reading inputs
 - After writing outputs

- **Смена состояний внутри МЭК приложения**
 - события, определенные прикладным программистом

1

Что такое событие в 61131-PLC?

2

Обработка событий в приложении

3

Обработка событий в библиотеках

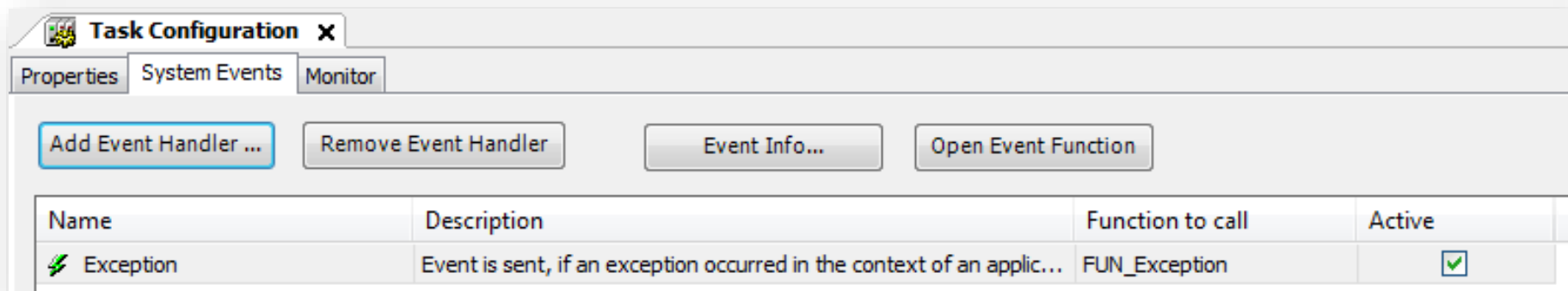
4

Заключение

Использование событий в настройках задач

Функции как обработчики predetermined events in the system

- Список доступных событий → Online help
- Система автоматически
 - создает обработчик события
 - регистрирует обработчик и событие
- Содержание обработчика событий определяется программистом.



Обработка событий в приложении

Использование событий в настройках задач

Пример: Событие „StartDone“

```

FUN_Start x
1  FUNCTION FUN_Start : DWORD
2  VAR_IN_OUT
3  |   EventPrm: CmpApp.EVTPARAM_CmpApp;
4  END_VAR

1  GVL.EventParamStart:= EventPrm;
2  GVL.diNumOfStart:= GVL.diNumOfStart + 1;
    
```

EventParamStart	CmpApp.EVTPARAM_CmpApp		
pApp	POINTER TO APPLICATION	16#025B0BA8	
pApp^	APPLICATION		
pAppParent	POINTER TO APPLICATION	16#00000000	POINTER TO parent application
iId	UDINT	0	Id of the application. Is always unique.
CodeGuid	ARRAY [0..15] OF BYTE		
DataGuid	ARRAY [0..15] OF BYTE		
udiState	UDINT	1	State of the application. See definitions Application states
udiOpState	UDINT	4097	Operating state of the application. See definitions Application operational states
bPersistentForce	DINT	0	Is persistent force enabled
szName	STRING(MAX_LEN_APPLNA...)	'Application'	Application name
udiPSVersion	UDINT	50661376	Version number of Programming system (set in download)
udiTargetSettingVersion	UDINT	50661376	Version number of Target settings used by PS (set in download)

Обработка событий в приложении

Использование событий в настройках задач

Пример: Событие „StopDone“

```

FUN_Stop x
1  FUNCTION FUN_Stop : DWORD
2  VAR_IN_OUT
3  EventPrm: CmpApp.EVTPARAM_CmpAppStop;
4  END_VAR
5  VAR
6  END_VAR

1  GVL.EventParamStop:= EventPrm;
2  GVL.diNumOfStop:= GVL.diNumOfStop + 1;
    
```

EventParamStop	CmpApp.EVTPARAM_CmpA...		
pApp	POINTER TO APPLICATION	16#025B0BA8	
pApp^	APPLICATION		
pAppParent	POINTER TO APPLICATION	16#00000000	POINTER TO parent application
iId	UDINT	0	Id of the application. Is always unique.
CodeGuid	ARRAY [0..15] OF BYTE		
DataGuid	ARRAY [0..15] OF BYTE		
udiState	UDINT	2	State of the application. See definitions Application states
udiOpState	UDINT	4097	Operating state of the application. See definitions Application operating states
uPersistentOnce	UDINT	0	Is persistent enabled
szName	STRING(MAX_LEN_APPLNA...)	'Application'	Application name
udiPSVersion	UDINT	50661376	Version number of Programming system (set in download)
udiTargetSettingVersion	UDINT	50661376	Version number of Target settings used by PS (set in download)
uStopReason	DWORD	4	

1

Что такое событие в 61131-PLC?

2

Обработка событий в приложении

3

Обработка событий в библиотеках

4

Заключение

Использование атрибутов

Простой метод использования predetermined messages

Attribute 'call_after_online_change_slot'

CODESYS Development System > Reference, Programming > Pragmas > Attribute Pragmas > Attribute 'call_after_online_change_slot'

Attribute 'call_after_online_change_slot'

The effect of this pragma is that all functions and programs containing this attribute are called after an online change. You define the order of calling by means of the attribute `<slot>`.

Syntax

```
{attribute 'call_after_online_change_slot' := '<slot>'}
```

`<slot>`: integral value that defines the ranking in the order of the calls: the lower the value, the earlier the call takes place. If several function blocks have the same ranking for the attribute, then the order of their calls remains indefinite.

Call: first line above the declaration part of functions and programs.

If a method possesses the attribute, then CODESYS determines all instances of the function block concerned. CODESYS calls all instances in the specified slot. In this case you have no influence on the order of the instances among themselves.



Регистрация существующих сообщений системы

- Пример CODESYS Store: „EventManager“
- Обработчик событий
 - Функция
 - Метод
- FB_Init
 - Открытие сообщения


```
hEventStart := EventOpen(EVT_StartDone, CMPID_CmpApp, Result);
```
 - Регистрация обработчика


```
ItfEventCallback := THIS^;
hInterfaceStart := EventRegisterCallback(hEventStart,
                                         ItfEventCallback,
                                         Result);
```
- FB_Exit Обязательно!
 - Снять с регистрации обработчик


```
EventUnregisterCallback(hEventStart, hInterfaceStart);
```

Обработка событий в библиотеках

Регистрация существующих сообщений системы

Системные компоненты имеют собственные события

VAR_GLOBAL CONSTANT EventIDs

Name	Type	Inherited from	Address	Initial	Comment
CMPID_CmpApp	DWORD			16#2	ComponentID to specify for EventOpen
EVTPARAMID_CmpApp	WORD			16#1	IDs for the event parameter structures
APP_STOP_REASON_DELETE	DWORD			6	
EVT_PrepareStart	DWORD			(SHL(UINT_TO_DWORD (EventClass.EVTCLASS_INFO), 16) OR 16#1)	** Events * Event is sent before start of the specified application * EVTPARAM_CmpApp
EVT_StartDone	DWORD			(SHL(UINT_TO_DWORD (EventClass.EVTCLASS_INFO), 16) OR 16#2)	** Events * Event is sent after start of the specified application * EVTPARAM_CmpApp
EVT_PrepareStop	DWORD			(SHL(UINT_TO_DWORD (EventClass.EVTCLASS_INFO), 16) OR 16#3)	** Events * Event is sent before stop of the specified application * EVTPARAM_CmpAppStop
EVT_StopDone	DWORD			(SHL(UINT_TO_DWORD (EventClass.EVTCLASS_INFO), 16) OR 16#4)	** Events * Event is sent after stop of the specified application * EVTPARAM_CmpAppStop
EVT_PrepareReset	DWORD			(SHL(UINT_TO_DWORD (EventClass.EVTCLASS_INFO), 16) OR 16#5)	** Events * Event is sent before reset of the specified application * EVTPARAM_CmpAppReset
EVT_ResetDone	DWORD			(SHL(UINT_TO_DWORD (EventClass.EVTCLASS_INFO), 16) OR 16#6)	** Events * Event is sent after reset of the specified application * EVTPARAM_CmpAppReset
EVT_ResetAllApplications_Done	DWORD			(SHL(UINT_TO_DWORD (EventClass.EVTCLASS_INFO), 16) OR 16#31)	** Events * Event is sent when AppResetAllApplications is finished * EVTPARAM_CmpAppResetAllApplications

Создание собственного события

1. Определить параметры события
2. Создать событие с определенными параметрами
3. Зарегистрировать событие в менеджере событий
4. Зарегистрировать обработчик (аналогично с системными событиями)
5. Посылать сообщения



Обработка событий в библиотеках

Пример в CODESYS

**1**

Что такое событие в 61131-PLC?

2

Обработка событий в приложении

3

Обработка событий в библиотеках

4

Заключение



Заключение

- Классический цикл приложения может обрабатывать сотни сигналов, все они могут приходить одновременно.
- Использование событий возможно на всех уровнях: в приложениях, библиотеках, а также в модулях системы исполнения.
- Прикладной программист может создавать собственные события и обрабатывать их.



Inspiring Automation Solutions

Thank you for your attention.