

CODESYS Projektvorlage Grafcet-Studio / Grafcet-Engine



1 Erklärung

Diese PDF-Datei beinhaltet alle Komponenten (außer der GrafcetEngine Bibliothek), die in einem CODESYS Projekt notwendig sind, damit Grafcet-Studio auf die SPS zugreifen kann.

Normalerweise ist dieses Dokument nicht notwendig. Nur wenn das Vorlageprojekt nicht geöffnet werden kann, ist dieses Dokument hilfreich.

Folgende Komponenten sind enthalten:

- Globale Variablen-tabelle „GVLGrafcetEngine“
- FB „GrafcetEngineRT“
- Programm „PLC_PRG“
- Persistente Variablen-tabelle „PersistentVars“

2 Projekt-Komponenten

2.1 Komponente Globale Variablen-tabelle „GVLGrafcetEngine“

```
{attribute 'qualified_only'}  
VAR_GLOBAL  
    GrafcetEngineRuntime: GrafcetStudio.GrafcetEngine; //1 instance in the array  
END_VAR
```

2.2 Komponente FB „GrafcetEngineRT“

```
FUNCTION_BLOCK GrafcetEngineRT  
VAR_INPUT  
END_VAR  
VAR_OUTPUT  
END_VAR  
VAR  
    TIME1: TIME;  
    GetDT:RTCLK.GetDateAndTime;  
    GrafcetEngineInitDone: BOOL; //to recognize a PLC reset. When FALSE then the grafcet has to be  
loaded!  
    MyClock:DT;  
END_VAR  
  
VAR_TEMP  
    i:DWORD;  
    SourcePtr: POINTER TO BYTE;  
    TargetPtr: POINTER TO BYTE;  
END_VAR  
  
//Get System Date and Time  
IF GetDT.xExecute=FALSE THEN  
    GetDT.xExecute:=TRUE;  
END_IF  
IF GetDT.xDone AND GetDT.xExecute THEN  
    MyClock := GetDT.dtDateAndTime ;  
    GetDT.xExecute:=FALSE;  
END_IF  
GetDT();
```

CODESYS Projektvorlage Grafcet-Studio / Grafcet-Engine



```
//-----  
//call the grafcet engine  
//-----  
//Call the grafcet engine and pass name, clock and ms ticks.  
GVLGrafcetEngine.GrafcetEngineRuntime(InstanceName:='WAGO PFC100', Init:=FALSE,Clock:=MyClock,  
Ticks1Ms:=TIME());  
  
//-----  
//this part manages the loading and saving from/to the sd card.  
//-----  
IF GrafcetEngineInitDone=FALSE THEN //need init? -> READ  
    GrafcetEngineInitDone:=TRUE;//init only 1 time  
    //Read from backup  
    IF PersistentVars.BackupSizeByte=SIZEOF(GrafcetStudio.GrafcetEngineData) AND  
  
PersistentVars.BackupVersion=GVLGrafcetEngine.GrafcetEngineRuntime.Data.Status.GrafcetEngineVersi  
on THEN  
    TargetPtr:=ADR(GVLGrafcetEngine.GrafcetEngineRuntime.Data);  
    SourcePtr:=ADR(PersistentVars.b1);  
    FOR i := 0 TO SIZEOF(GrafcetStudio.GrafcetEngineData)-1 DO  
        TargetPtr[i]:=SourcePtr[i];  
    END_FOR;  
    GVLGrafcetEngine.GrafcetEngineRuntime.Data.Status.FirstLoop:=1; //init grafcet  
logic  
    GVLGrafcetEngine.GrafcetEngineRuntime.Data.Status.GrafcetRun:=1;  
    END_IF;  
END_IF;  
//WriteReadInit:=GrafcetEngineInit=FALSE OR GrafcetEngineRuntime[engineInstance].NewDataEvent=1;  
//first call of WriteReadToSdCard?  
//new grafcet prg?  
IF GVLGrafcetEngine.GrafcetEngineRuntime.NewDataEvent=1 THEN //->WRITE  
    GVLGrafcetEngine.GrafcetEngineRuntime.NewDataEvent:=0;  
    //Write to backup  
    SourcePtr:=ADR(GVLGrafcetEngine.GrafcetEngineRuntime.Data);  
    TargetPtr:=ADR(PersistentVars.b1);  
    FOR i := 0 TO SIZEOF(GrafcetStudio.GrafcetEngineData)-1 DO  
        TargetPtr[i]:=SourcePtr[i];  
    END_FOR;  
    PersistentVars.BackupSizeByte:=SIZEOF(GrafcetStudio.GrafcetEngineData);  
  
PersistentVars.BackupVersion:=GVLGrafcetEngine.GrafcetEngineRuntime.Data.Status.GrafcetEngineVersi  
on;  
END_IF
```

2.3 Komponente Programm „PLC_PRG“

```
PROGRAM PLC_PRG  
VAR  
MyGrafcetEnging:GrafcetEngineRT;  
  
END_VAR  
MyGrafcetEnging();
```

2.4 Komponente Persistente Variablen-tabelle „PersistentVars“

```
{attribute 'qualified_only'}  
VAR_GLOBAL PERSISTENT RETAIN  
    b1:ARRAY[1..SIZEOF(GrafcetStudio.GrafcetEngineData)] OF BYTE;  
    BackupVersion: WORD;  
    BackupSizeByte:DWORD;  
END_VAR
```