

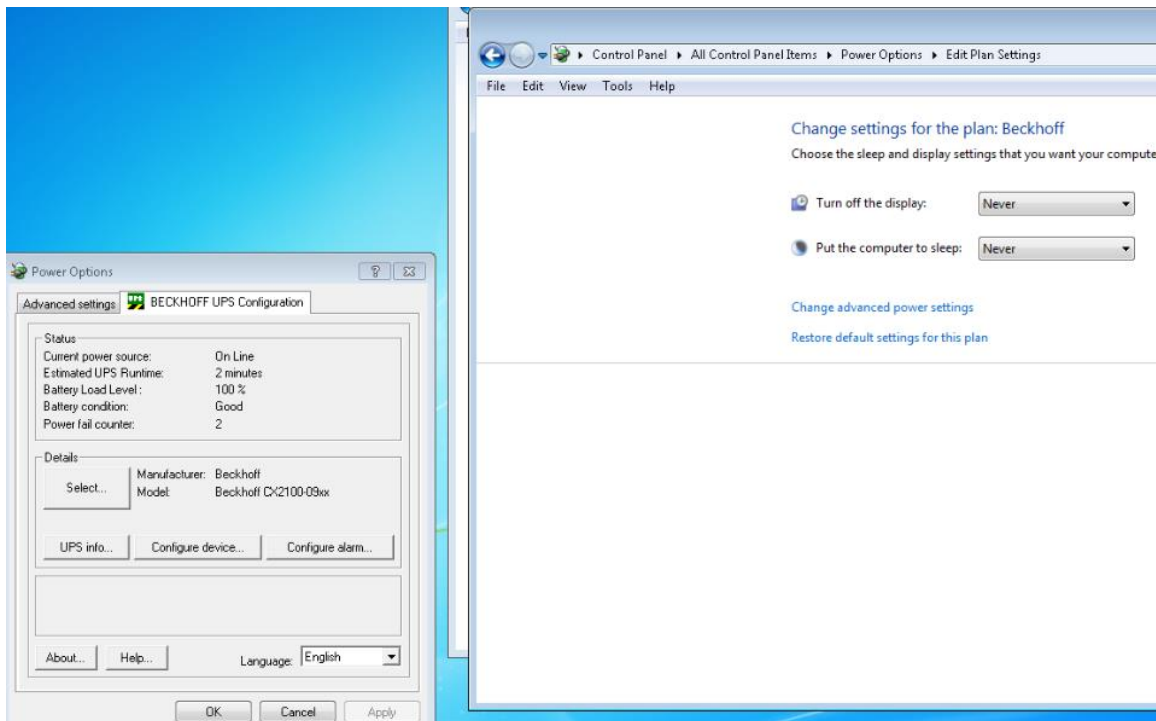
HANTERING AV UPS CX2100-0904

Hantering av UPS-modulen CX2100-0904. I WES7 och TwinCAT 2.

Inställning av UPS:en i WES7

UPS:ens konfigurationsflik

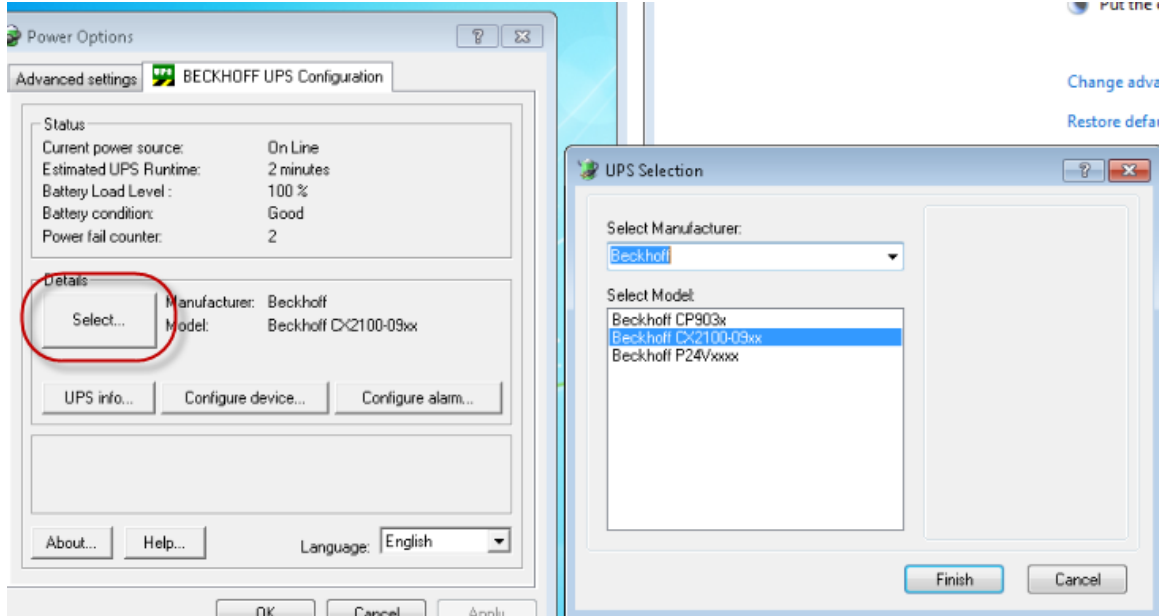
Inställningsmöjligheterna för hur UPS:en beter sig finns under ”Power options”.
Control Panel -> Power Options -> ”Change plan settings” för ”Beckhoff” -> Change advanced power settings -> Beckhoff UPS Configuration



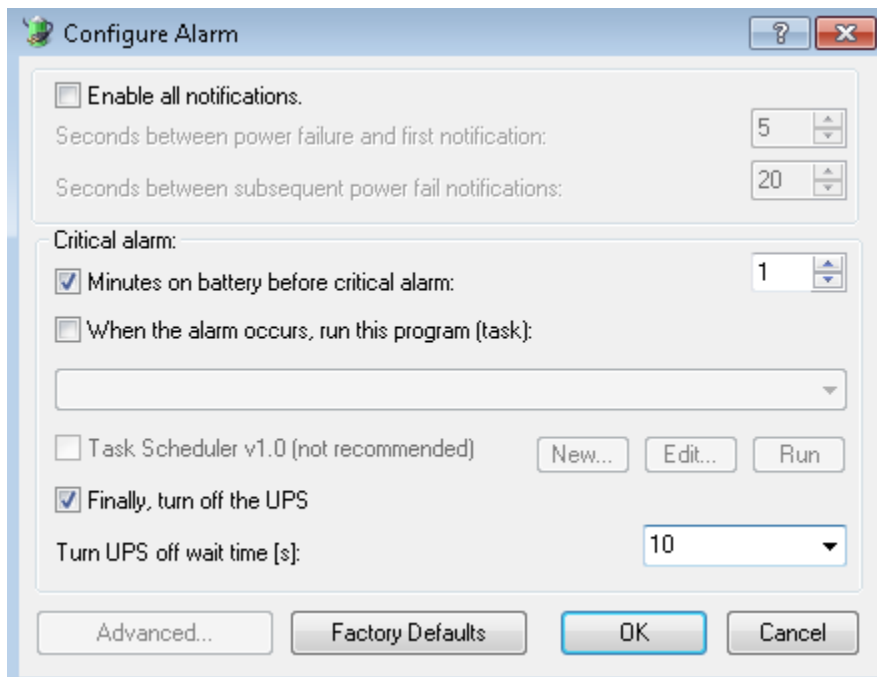
Under denna flik syns hur många minuter drifttid UPS:en klarar att hålla vid nuvarande strömförbrukning, batterinivå samt hur många gånger 24-volten fallit bort sedan senaste omstarten. Senare i dokumentet samt i exempelprojektet så demonstreras hur denna information kan lyftas in i ett PLC-projekt.

Konfiguration

Först och främst måste rätt modell av UPS vara vald:



Under ”Configure alarm”. Om ”Finally, turn off the UPS” är iöbckad, så kommer UPS:en göra en kontrollerad nerstängning innan UPS:en är urladdad.



Om 24-voltspänningen kommer tillbaka innan nerstängningen påbörjats så utförs ingen nerstängning.

Persistenta variabler

I och med att en kontrollerad nerstängning av operativsystemet utförs så kommer de variabler som är definierade som persistenta variabler i PLC-projektet sparas ner till filen TCPLC_T_X.wbp (X=numret på aktuell runtime) i boot-mappen när CX:en stängs ner och de befintliga värdena i dessa kommer åter läsas upp nästa gång CX:en startar upp sitt bootprojekt.

```
VAR_GLOBAL PERSISTENT
    nTal1 : INT;
    nTal2 : INT;
END_VAR
```

Vill man skriva ner dessa värden till filen under PLC-programmets exekvering utan att PLC:n stängs ner behöver man använda funktionsblocket FB_WritePersistentData och exekvera det varje gång en skrivning ska göras.

UPS-information och diagnostik i PLC:n

För att få ut information och diagnostik från UPS:en i PLC-projektet så behöver funktionsblocket FB_GetUPSStatus användas. Det finns att tillgå i funktionsblocksbiblioteket TcIoFunctions.lib.

```
FB_GetUPSStatus (
    sNetId:='',
    nPort:=0,
    bEnable:=TRUE,
    bValid=>bValid,
    bError=>bError,
    nErrId=>nErrId,
    stStatus=>stStatus
);
```

Ur denna kan structen ST_UPSStatus läsas ut. Denna innehåller information om bland annat: UPS:ens procentuella kvarvarande livstid, kvarvarande laddning i minuter, kommunikationsstatus mot UPS:en, strömförsörjningsstatus (om 24-voltmatningen fallit bort) samt en räknare för hur många gånger sedan senaste uppstart som det sistnämnda inträffat.

```
[-] stStatus
.....Vendor = 'Beckhoff'
.....Model = 'Beckhoff CX2100-09xx'
.....FirmwareRev = '1.0-0'
.....SerialNumber = ''
.....BatteryLifePercent = 84
.....BatteryLifeTime = 2
.....eBatteryStatus = BatteryOk
.....eCommStatus = UpsCommOk
.....ePowerStatus = PowerOnBattery
.....nPowerFailCnt = 4
.....dwChargeFlags = 0
```

En av de mest centrala är troligen E_UpsPowerStatus:

PowerUnknownStatus - Okänd status vid misslyckad inläsning

PowerOnLine - UPS:en går på nätspänning

PowerOnBattery - Nätspänningen har fallit bort och UPS:en driver CX:en.

Denna kan vara intressant att använda om man vill ha larmindikering i sitt program när strömförsörjningen försvinner.

Exempelprojekt

Exempelprojekt till TwinCAT 2.11 som läser in status och diagnostik från UPS:en två gånger i sekunden:

```
PROGRAM MAIN
```

```
VAR
```

```
    tim1 : TON;  
    FB_GetUPSStatus:FB_GetUPSStatus;  
    bEnable: BOOL;  
    stStatus : ST_UPSStatus;  
    ePowerStatus : E_UpsPowerStatus;
```

```
    FB_WritePersistentData:FB_WritePersistentData;  
    bWrite: BOOL;
```

```
END_VAR
```

```
tim1(IN:=NOT tim1.Q , PT:=T#2s , Q=>bEnable , ET=> );
```

```
FB_GetUPSStatus(  
    sNetId := '',  
    nPort := 801,  
    bEnable := bEnable,  
    stStatus => stStatus  
);
```

```
ePowerStatus := stStatus.ePowerStatus;
```

```
IF ePowerStatus = PowerOnBattery THEN  
    bWrite := TRUE;
```

```
END_IF
```

```
FB_WritePersistentData(  
    NETID:= '',  
    PORT:= 801,  
    START:= bWrite,  
    TMOUT:= t#3s,  
    MODE:= 0,  
    BUSY=> ,  
    ERR=> ,  
    ERRID=> );
```

```
IF NOT FB_WritePersistentData.BUSY THEN  
    bWrite := FALSE;
```

```
END_IF
```

Fel i och förbättringar av detta dokument meddelas till support@beckhoff.se.

This publication contains statements about the suitability of our products for certain areas of application. These statements are based on typical features of our products. The examples shown in this publication are for demonstration purposes only. The information provided herein should not be regarded as specific operation characteristics. It is incumbent on the customer to check and decide whether a product is suitable for use in a particular application. We do not give any warranty that the source code which is made available with this publication is complete or accurate. This publication may be changed at any time without prior notice. No liability is assumed for errors and/or omissions. Our products are described in detail in our data sheets and documentations. Product-specific warnings and cautions must be observed. For the latest version of our data sheets and documentations please visit our website (www.beckhoff.com).

© Beckhoff Automation GmbH, September 2009

The reproduction, distribution and utilisation of this document as well as the communication of its contents to others without express authorisation is prohibited. Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.