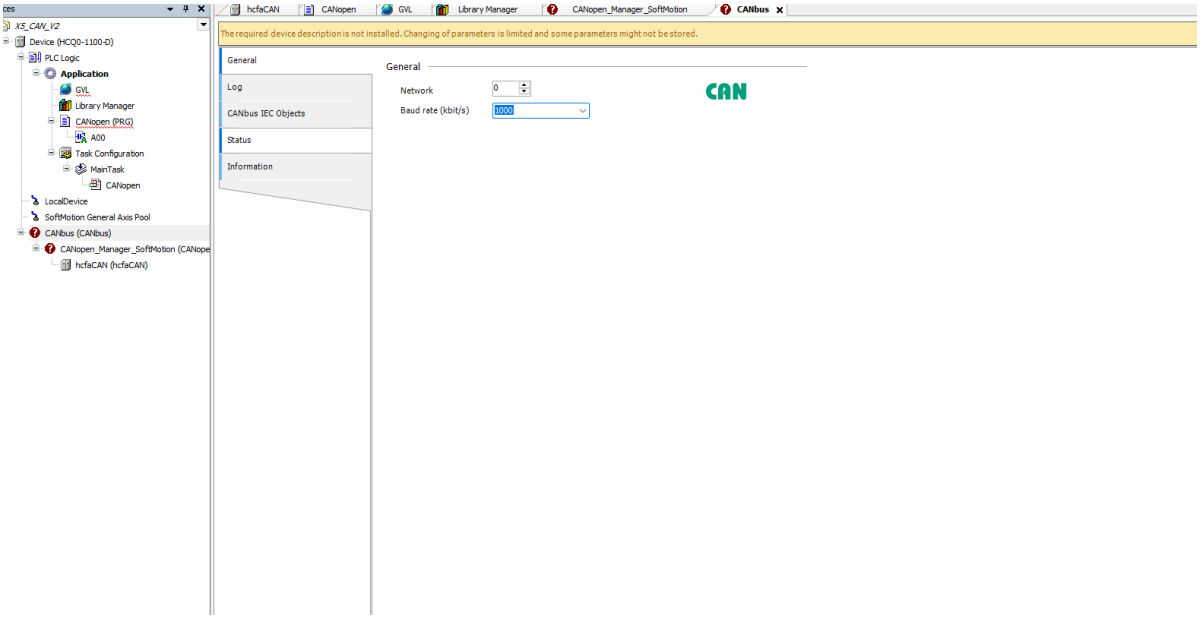
Quick Start Configuration of HCFA CANopen

Since CANOPEN bus servo is relatively more complex to configure compared with EtherCAT bus servo, in order to facilitate users' use and quick start, a set of configuration that is the most applicable and has the highest bus performance utilization rate is made for users to refer to or directly copy and use.

The default baud rate of the Demo program is 1000K. If the bus length is long or the interference is large, the communication rate can be appropriately reduced. The following data are all based on 1000K. If the rate is lower than 1000K, please reduce the data standard accordingly.

At a 1000K CAN bus rate, a maximum 4ms synchronization cycle can be achieved to control 8-axis CANOPEN servos.

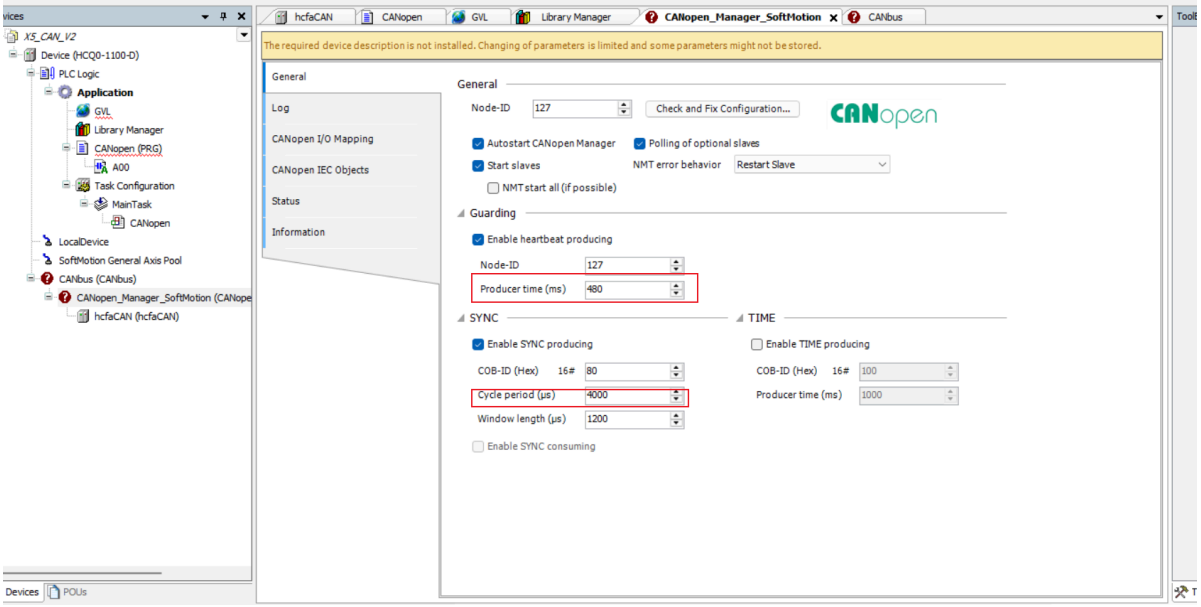
The following is a brief description of the configuration and precautions in the Demo program.



Set the baud rate according to the CANopen Servo.

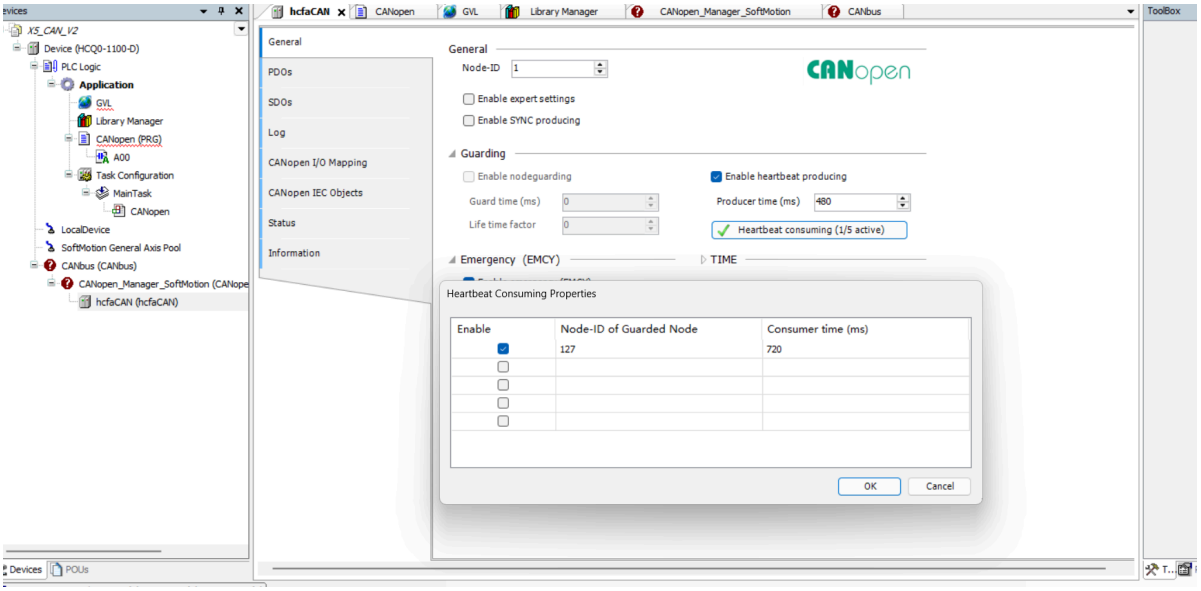
2.CANopen basic setting

If there are many axes and the bus load is too high, the cycle period can be increased. Producer Time is the heartbeat production time and must be a multiple of the cycle period. If the cycle period is modified, it needs to be modified synchronously.

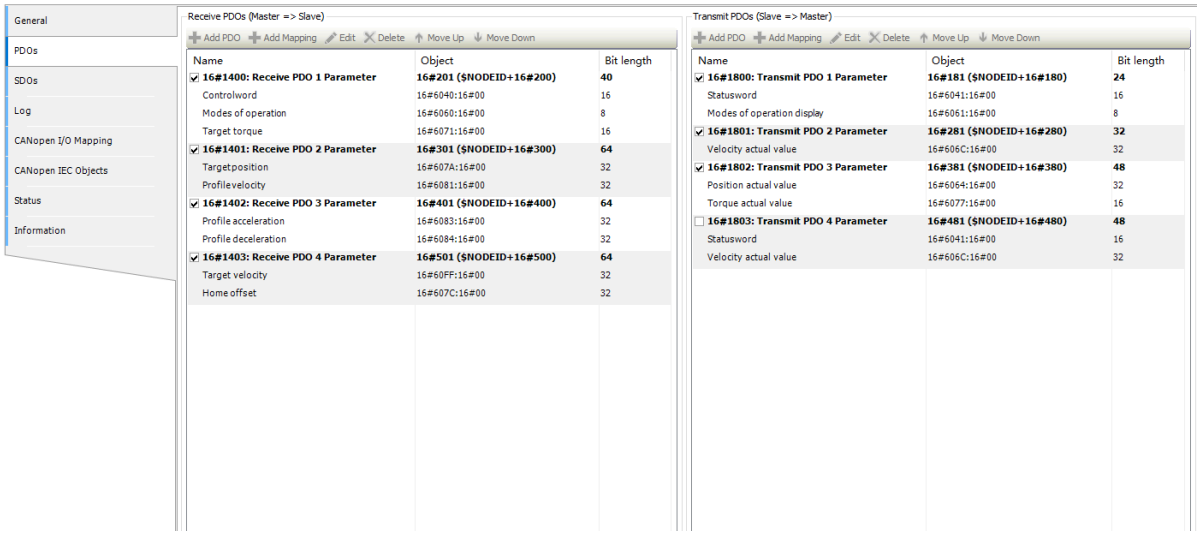


3.hcfaCAN setting

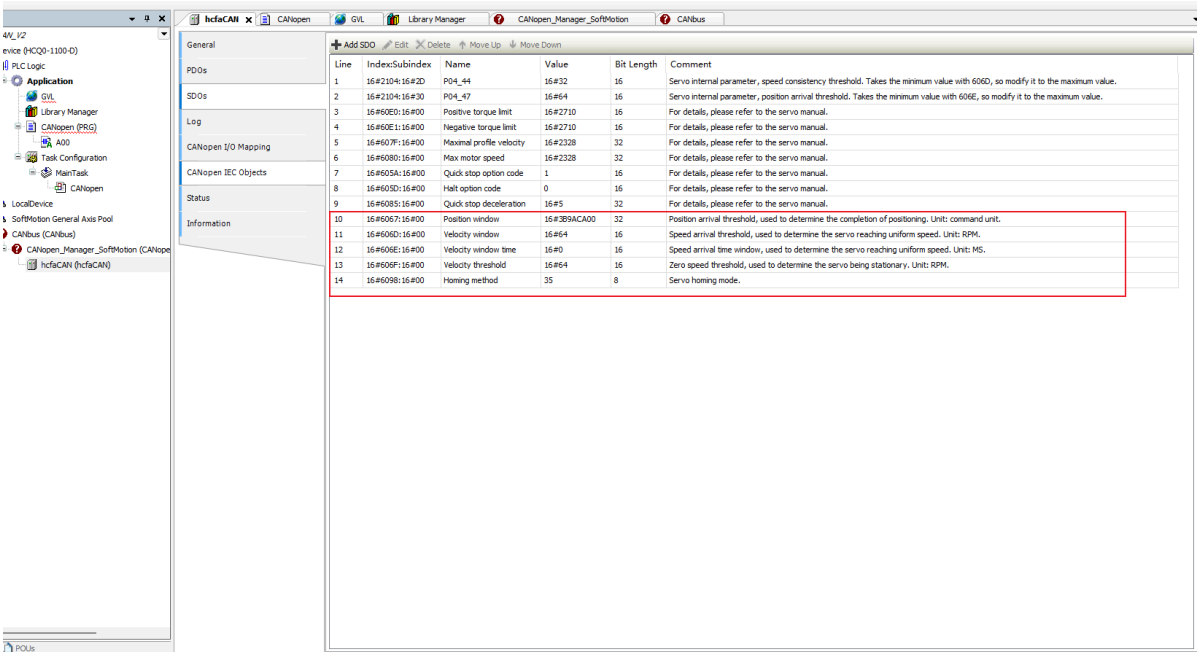
Slave station overview tab: In addition to modifying the station number according to the configuration, you need to check the "Enable Heartbeat Production" option, set the production time as required, and it is generally recommended to be longer than the heartbeat production time of the master station. Set the heartbeat consumption option. Click the heartbeat consumption option, check the first checkbox in the pop-up window (if it is already checked, there is no need to check it again), confirm whether the protected node ID is the master station. If not, modify it to the master station ID, and set the consumption time to 1.5 times the heartbeat of the master station.



4.CANopen PDO setting



5.CAN SDO setting

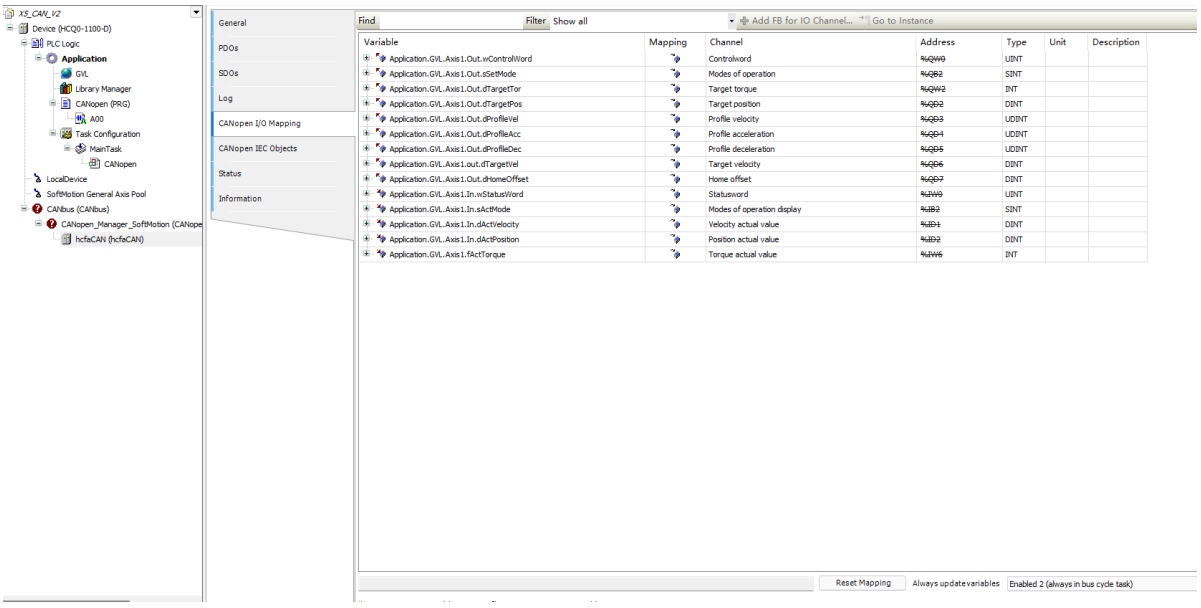


When the positioning accuracy is insufficient, or the positioning completion signal output is not timely enough (for example, the Done signal of MC_MoveAbsolute_CO), the parameter value of 6067 can be modified accordingly.

When the speed consistency judgment accuracy is insufficient, or the speed consistency signal output is not timely enough (for example, the InVelocity signal of MC_MoveVelocity_CO), the parameter values of 606D and 606E can be modified accordingly.

Add more slave stations: Right-click the configured slave station in Demo -> click Copy -> right-click the CANPEN master station -> click Paste to copy a configured slave station. Only need to modify the slave station number and the corresponding axis variable in the mapping. Each slave station must correspond to a separate axis variable.

6.hcfaCAN Axis mapping



7.project Example

```
PROGRAM CANopen
VAR
    MC_Power_C1 :MC_Power_CO;
    MC_Home_CO1 :MC_Home_CO;
    MC_Reset_CO1 :MC_Reset_CO;
    MC_Stop_CO1 :MC_Stop_CO;
    MC_MoveVelocity_CO1 :MC_MoveVelocity_CO;
    MC_MoveAbsolute_CO1 :MC_MoveAbsolute_CO;
    MC_MoveRelative_CO1 :MC_MoveRelative_CO;
    MC_Jog_CO1 :MC_Jog_CO;


END_VAR
```

```
GVL.Axis1.pDevice :=ADR (hcfaCAN); //---Specify the corresponding CANOPEN device. Must be assigned a value
GVL.Axis1.fEncoderFactor :=10000; //---Assign the gear ratio. Gear ratio = (number of motor revolution commands : application units corresponding to one motor revolution)
A00();


b_伺服通信状态 :=GVL.Axis1.bCommunication; //--- Read the communication status of the slave station
e_轴运行状态 :=GVL.Axis1.nAxisState; //--- Read the axis running status
e_轴故障代码 :=GVL.Axis1.eErrorID [0]; //--- Read the current fault code
f_轴当前位置 :=GVL.Axis1.fActPosition; //--- Read the servo feedback position
f_轴当前速度 :=GVL.Axis1.fActVelocity; //--- Read the servo feedback speed
f_轴当前转矩 :=GVL.Axis1.fActTorque; //--- Read the servo feedback torque
```

8.CAN axis

1.Maintenance of CANOPEN Slave Devices: To automatically maintain the communication status of CANOPEN slave stations, the address of the CANOPEN slave station instance needs to be assigned to the axis variable pDevice pointer.

```
Axis1.pDevice :=ADR(hcfaCAN); //---Specify the corresponding CANOPEN device,
assignment is mandatory
```

2.Gear Ratio Conversion: The fEncoderFactor variable in the axis variable is the user gear ratio. The user gear ratio conversion can be completed by modifying this value. The calculation formula is:

$$Gearratio = (\frac{\text{Number of commands per motor revolution}}{\text{Application units corresponding to one motor revolution}})$$

corresponding to one motor revolution. The fActPosition and fActVelocity in the axis variable are values calculated from the corresponding PDO data through the gear ratio.

```
Axis1.fEncoderFactor :=10000; //---Set the gear ratio
```