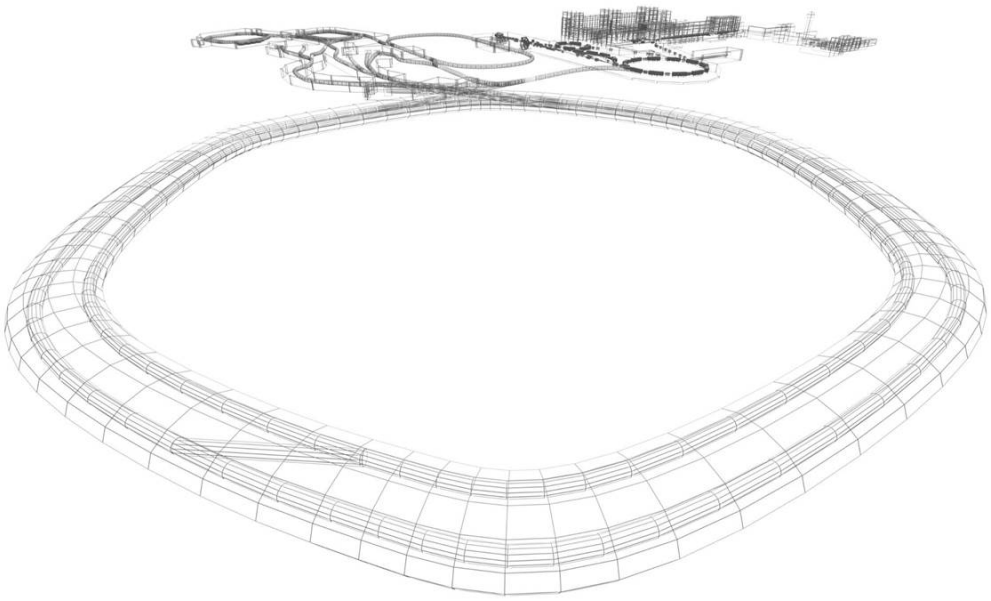
How to exchange a pneumatic drive and save it's process values in the Step 7 V13 Programmable Logic Controller (PLC) program for CRYRING

André Petit

August 21, 2015
v1.0



Abstract

This document is a tutorial explaining the necessary steps that have to be done in Step 7 in order to exchange a drive. It shows how to save it to a local or network storage folder and how to assign an already stored drive to a specific location in the main program. This feature is important because of the welded edge bellow around the lifting rod that prevents from a vacuum leak during motion. This bellow has a certain life expectancy. It's because of this why we count and have to preserve the number of times the welded bellow is compressed and stretched. Please not that this tutorial is based on the PLC source code V0.3 that is available for download at ftp://lxpool.gsi.de:22

Contents

1 How to save the process values of a pneumatic actuator

In order to save a pneumatic drives values on a local or network folder you have to do the following steps:

1. Open the function "callDrives".
   In this example, you are going to change the drive that is installed at the location "YR07DF2" and displayed in Network 1. Here you have to look for the Function block (FB) with the nomenclature of the location. Once found, check the name of it's instance Data block (DB) that is representing the drive (see figure 1).

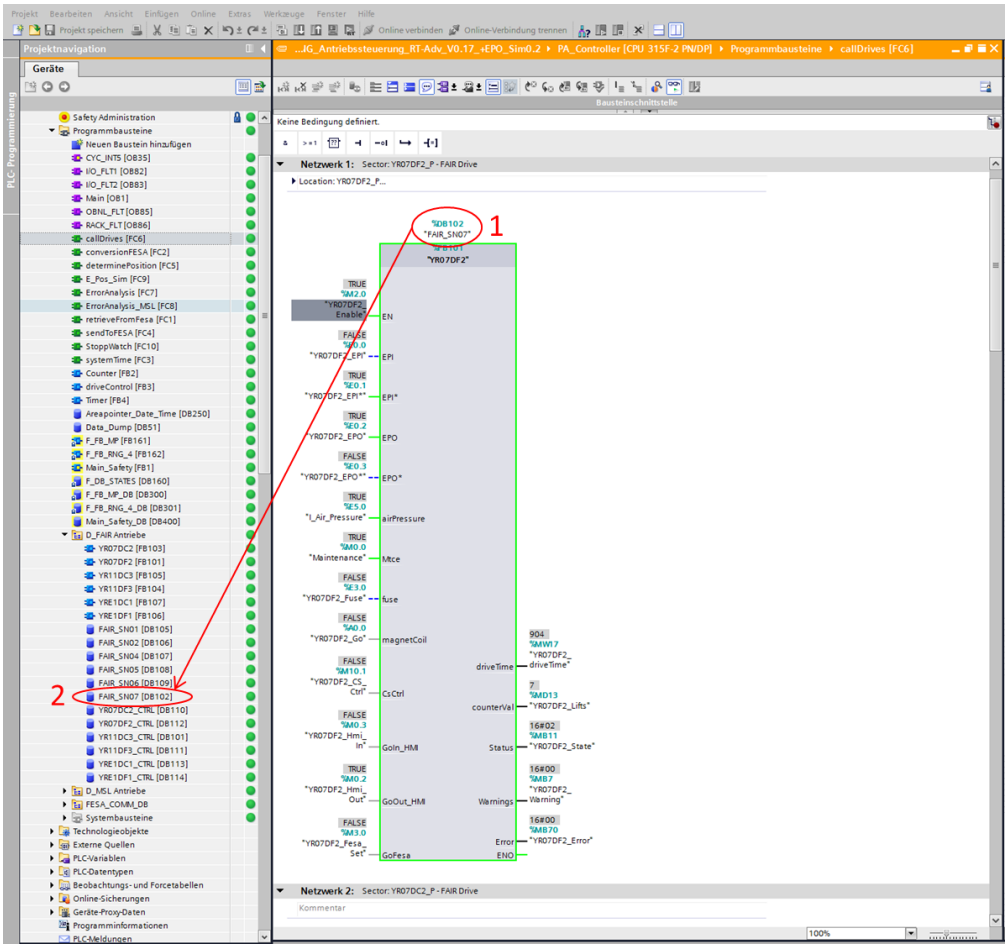


Figure 1: FB and DB that represent the drive at a specific location

2. Select the corresponding DB from the project tree
3. Right click on it → snapshot of the monitor values.
4. Right click again → Apply snapshot values as start values → All values (Figure 2).
5. Confirm the pop up message by pressing "OK".

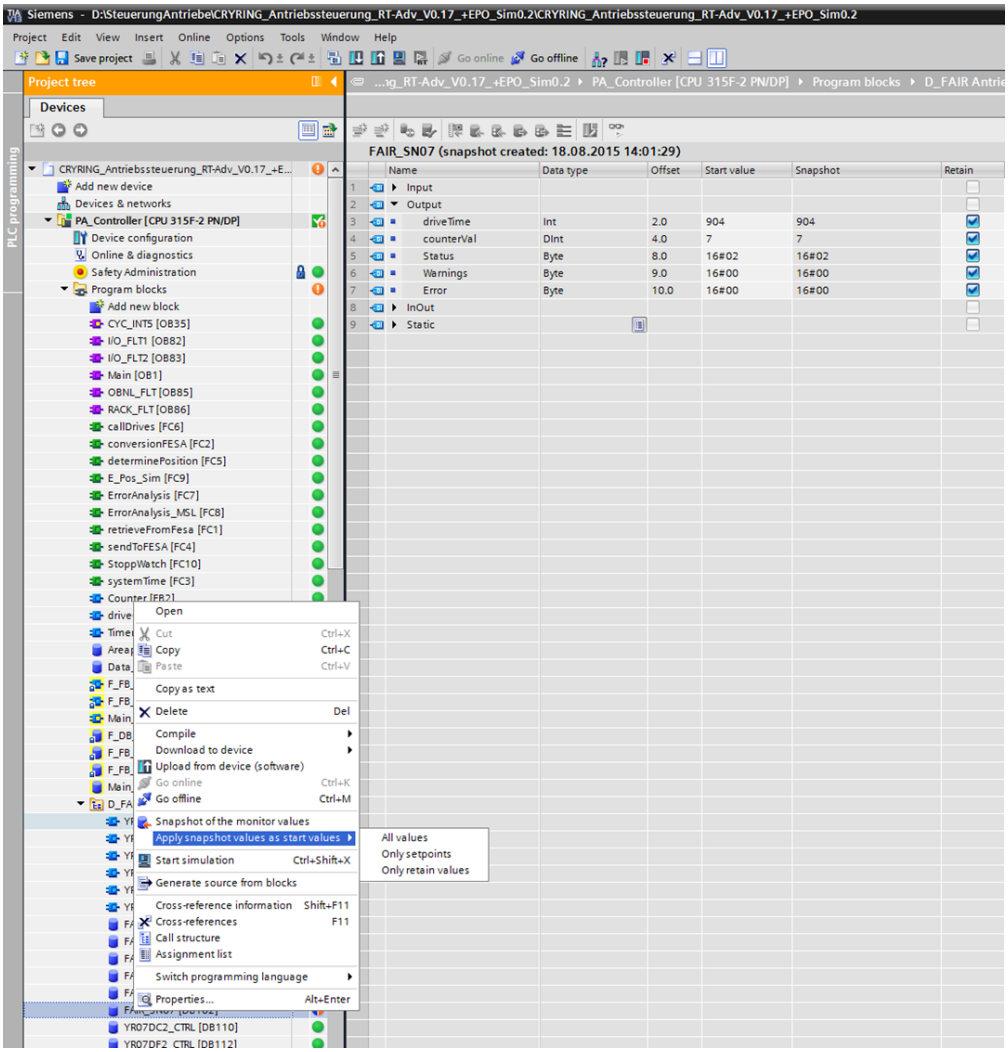


Figure 2: Apply the snapshot values of the DB as start values

6. Last time, right click on the DB again → Generate source from blocks (Figure 3).

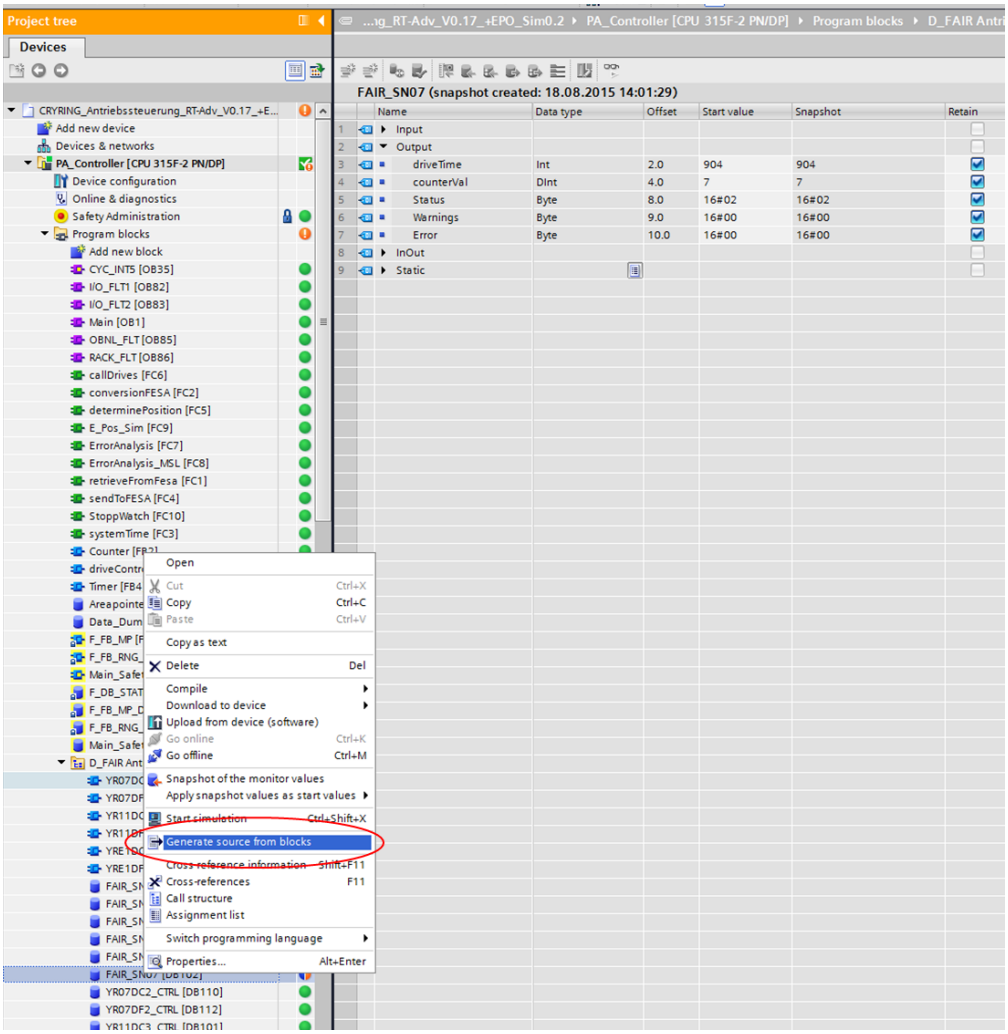


Figure 3: Generate a source file with the DB values

7. Choose the location where you want to save the file. This can be either on the hard drive or a network folder. A .db-file was generated. It's content should be similar to the following:

```
DATA_BLOCK "FAIR_SN07"
{ S7_Optimized_Access := 'FALSE' }
VERSION : 0.1
"YR07DF2"

BEGIN
EPI := FALSE;
"EPI*" := TRUE;
EPO := TRUE;
"EPO*" := FALSE;
airPressure := TRUE;
Mtce := TRUE;
fuse := FALSE;
driveTime := 904;
counterVal := 7;
Status := 16#02;
Warnings := 16#00;
```

```
Error := 16#00;
magnetCoil := FALSE;
CsCtrl := FALSE;
GoIn_HMI := FALSE;
GoOut_HMI := TRUE;
GoFesa := FALSE;
"Error_#"[0] := FALSE;
"Error_#"[1] := FALSE;
Warning[0] := FALSE;
Warning[1] := FALSE;
Warning[2] := FALSE;
Warning[3] := FALSE;
Warning[4] := FALSE;
Warning[5] := FALSE;
Warning[6] := FALSE;
State[0] := FALSE;
State[1] := TRUE;
State[2] := FALSE;
State[3] := FALSE;
State[4] := FALSE;
timeDifference_Out := T#544MS;
timeDifference_IN := T#1S_264MS;
OutsideToInside.Trigger := FALSE;
OutsideToInside.STOP := FALSE;
OutsideToInside.DifferenceTime := T#544MS;
OutsideToInside.StartTime := DT#2015-06-08-15:36:20.499;
OutsideToInside.StopTime := DT#2015-06-08-15:36:21.043;
OutsideToInside.HM_Trigger_Start := FALSE;
OutsideToInside.HM_Trigger_Stop := FALSE;
OutsideToInside.Flank_Start := FALSE;
OutsideToInside.Flank_Stop := FALSE;
InsideToOutside.Trigger := TRUE;
InsideToOutside.STOP := TRUE;
InsideToOutside.DifferenceTime := T#1S_264MS;
InsideToOutside.StartTime := DT#2015-06-08-15:36:26.227;
InsideToOutside.StopTime := DT#2015-06-08-15:36:27.491;
InsideToOutside.HM_Trigger_Start := TRUE;
InsideToOutside.HM_Trigger_Stop := TRUE;
InsideToOutside.Flank_Start := FALSE;
InsideToOutside.Flank_Stop := FALSE;
Hubz hler_Instance.Trigger := FALSE;
Hubz hler_Instance.counterVal := 7;
Hubz hler_Instance.counterWarning_Val2 := FALSE;
Hubz hler_Instance.counterWarning_Val1 := FALSE;
Hubz hler_Instance.MerkerH be := 7;
Hubz hler_Instance.MERKER_N := FALSE;

END_DATA_BLOCK
```

The values in this file can be edited at any time.

The next section describes how to assign a pneumatic actuator, that has already been stored on the hard drive, to the location "YR07DF2".

2 How to implement an actuator that is saved on a local or network folder

Before you can assign a drive from the local storage to a new location you have to edit the .db-file with a text editor of your choice (e.g. Notepad++).

1. Open the .db-file that you want to use.
2. Change the line between "VERSION" and "BEGIN" from the last location (e.g. "YR11DF3") to "YR07DF2". See the following example:

   Before:

   ```
   DATA_BLOCK "FAIR_SN11"
   { S7_Optimized_Access := 'FALSE' }
   VERSION : 0.1
   "YR11DF3"

   BEGIN
   EPI := FALSE;
   ...
   ...
   ```

   After:

   ```
   DATA_BLOCK "FAIR_SN11"
   { S7_Optimized_Access := 'FALSE' }
   VERSION : 0.1
   "YR07DF2"

   BEGIN
   EPI := FALSE;
   ...
   ...
   ```

   In case you also want to change the drives counter value, you only need to modify the number in the following line:

   ```
   ...
   Hubzaehler_Instance.MerkerHuebe := 164;        //Change only the number
   ...
   ```

3. Save the file and close the editor.
4. In Step 7 go back to the function "callDrives".
5. Select the instance DB of the FB where you want to change the drive.
6. Write down the drives DB number, in our case 102.
7. Right click → delete (Figure 4).

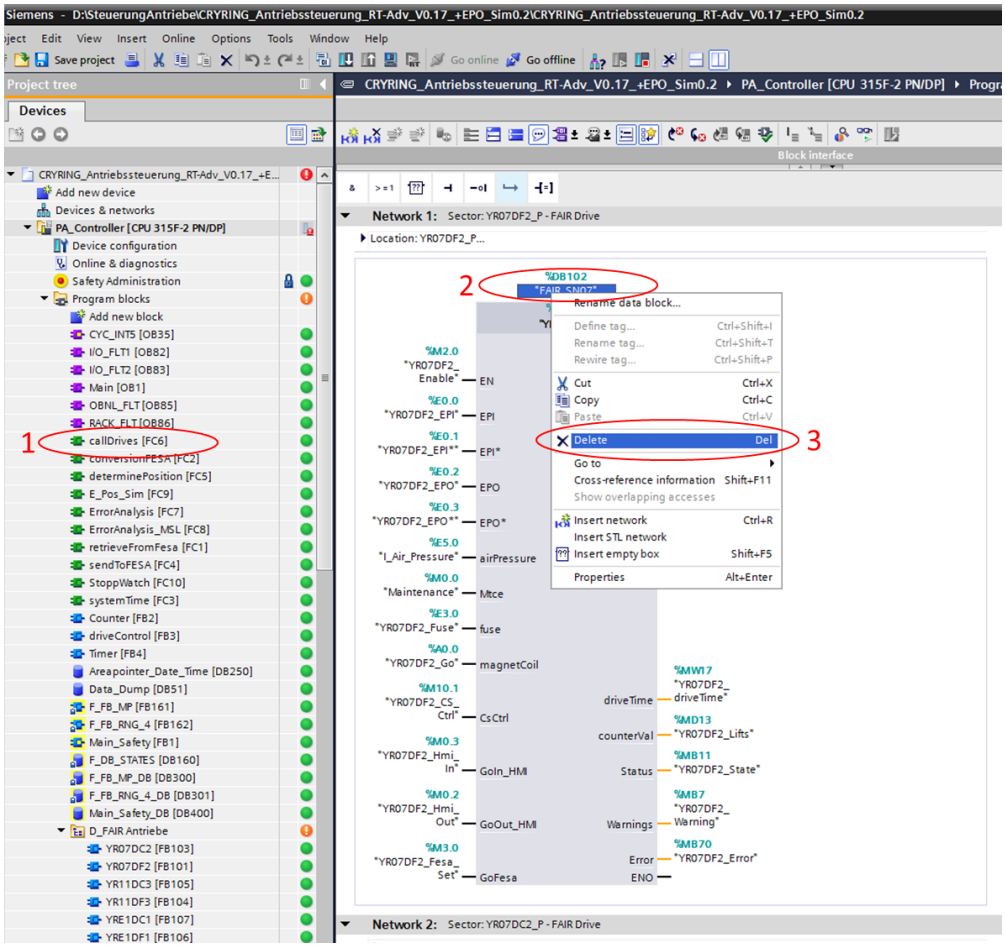


Figure 4: Delete instance DB

8. Delete it also from the project (There should appear three red question marks on top of the FB instead of the drives serial number).

9. Go to the project tree → External source files → add new external file → select the new drive (.db-file) → Open.

10. Right click on the added file → "Generate blocks from source" (Figure 5).
The DB is now available in the project tree.

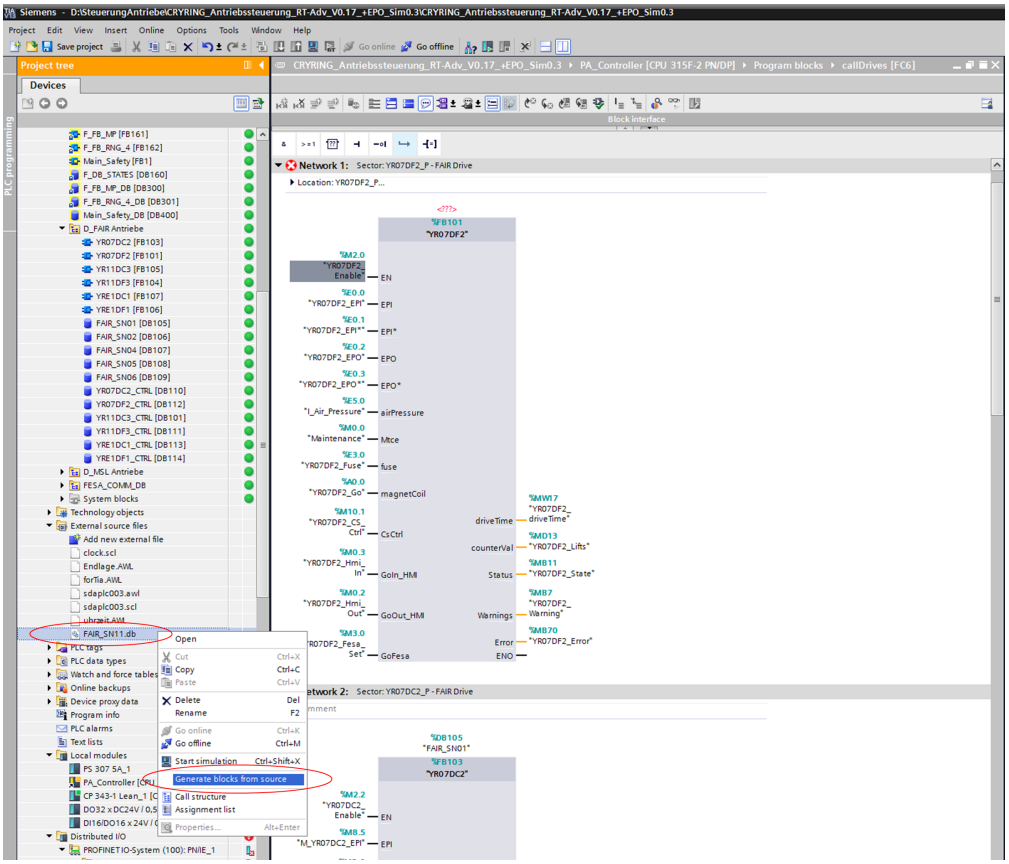


Figure 5: Generate DB from source

11. Select the DB → right click → properties.
    Under "General" choose "manual" and change the drives number to 102 (Figure 6).

12. Press "OK"

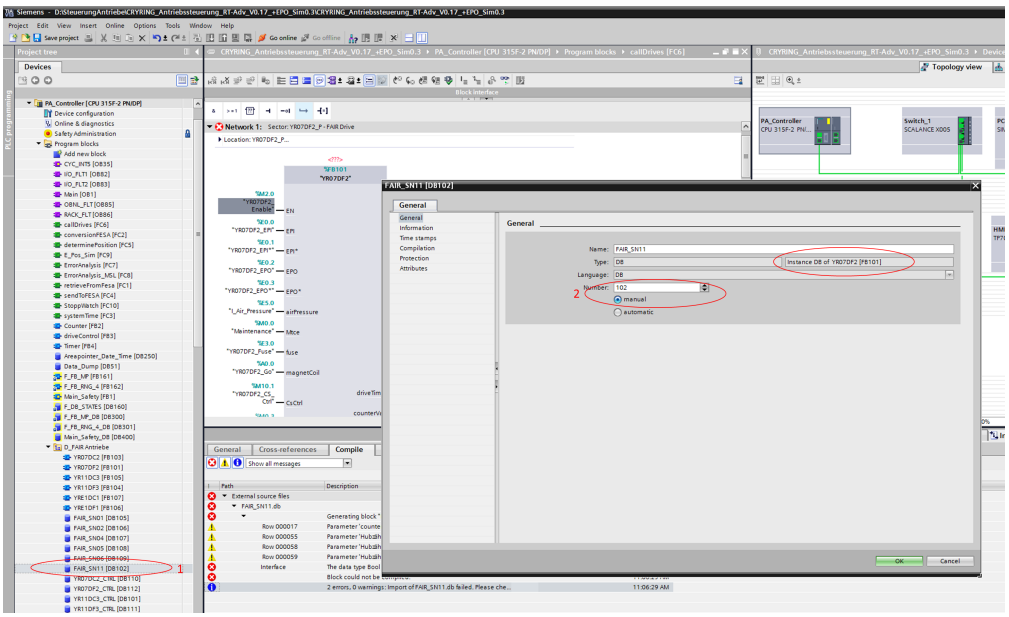


Figure 6: Change DB number

13. Drag the DB from the project tree and drop it on the question marks over the FB (Figure 7)

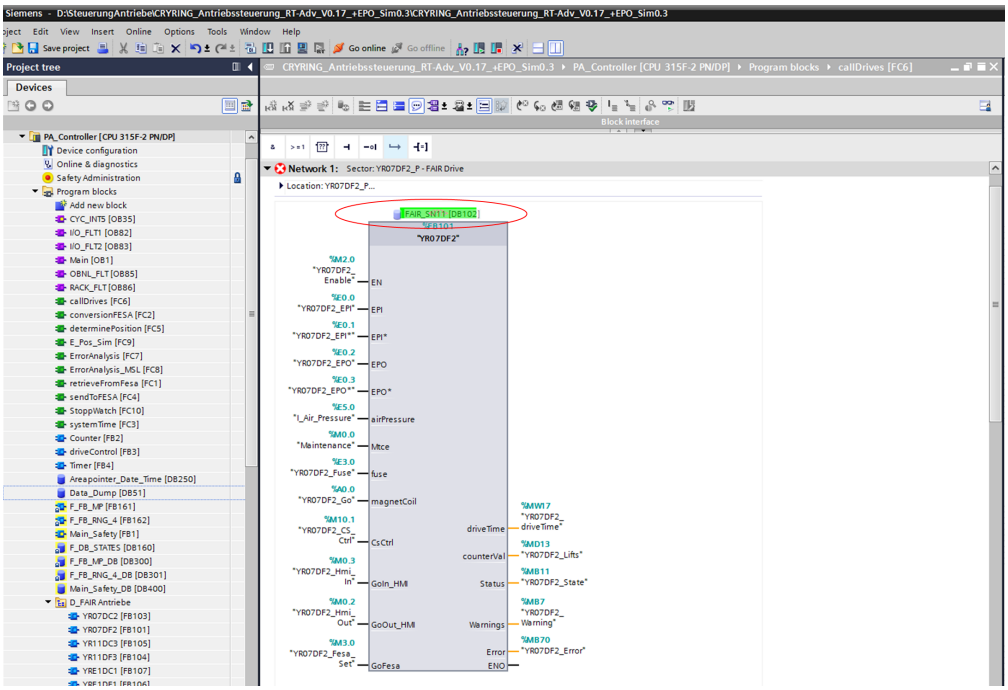


Figure 7: Drag DB over FB

14. Compile only the software changes (Figure 8)

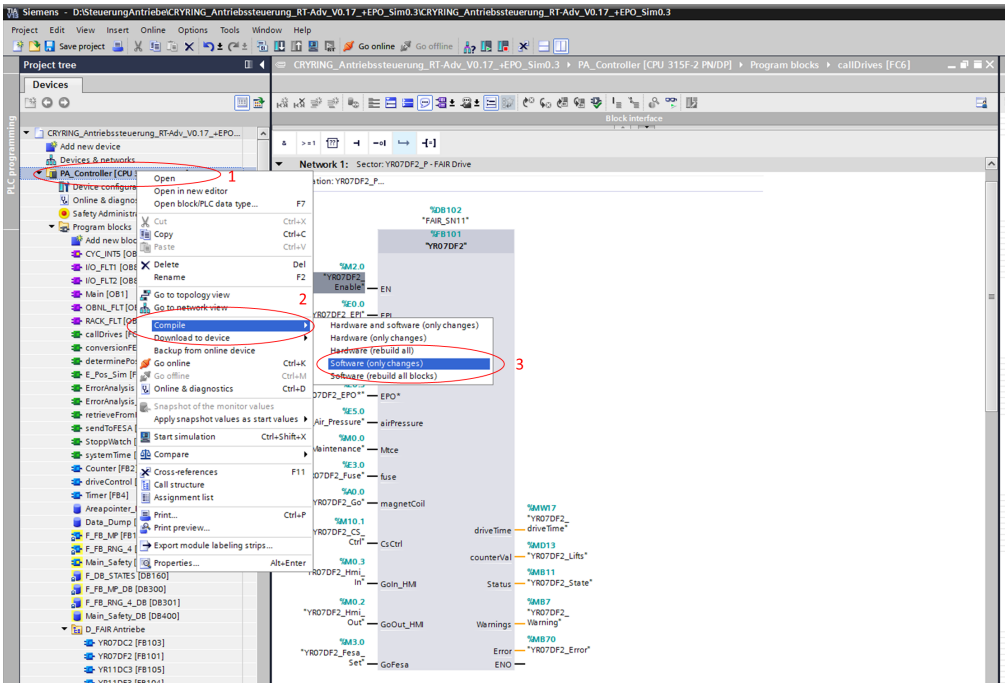


Figure 8: Compile the software changes

15. Load them to the PLC-Controller and reinitialize the DB.

List of Figures

References

Acronyms

DB Data block
FB Function block
PLC Programmable Logic Controller