

EPOS2 versus EPOS4: Connectors, Object Dictionary, Operating Modes

Topic:

- What has to be considered if an EPOS2 of an existing system should be replaced by an EPOS4 in future?
- Is it possible to use EPOS2 programming libraries also for EPOS4?

Situation:

You have EPOS2 in use and think about to migrate your current or new system designs to EPOS4 instead. Your current system might be based on your own master code, the EposCmd.dll or a 3rd party library. Even if you just have a quick look, it is obvious that there are some differentiating factors (e.g. connectors and size) in between the EPOS2 and EPOS4. There might be even more details to think about.

Technical background:

The first EPOS product generation was launched in the year 2003. The objects and operating modes specified by the CANopen DS402 standard have been partly strongly enhanced during that period of time. The EPOS4 was adapted to the newest standards to ensure a high compatibility with the current CANopen and EtherCAT standards as well as to fit the requirements by the market and maxon customers. This also meant that there are partly some remarkable differences in between the EPOS2 and EPOS4 object dictionary and operating modes present.

Solution:

1. Connectors, dimensions, mounting

The EPOS4 is more compact by its design, i.e. that the dimensions and mounting has to be adapted if an EPOS2 is replaced by an EPOS4.

The EPOS4 offers the same I/O and encoder features for each product type. This required partly different connectors in use than by the EPOS2. There will be some need to adapt the connectors of existing EPOS2 I/O and sensor cables or replace the cables by new EPOS4 types.

Practical tip:

Please find all the information about the connectors and pin assignments in use by the corresponding EPOS4 hardware reference.

2. EPOS2 objects 0x2000 ... 0x2FFF

The objects which are part of the manufacturer specific EPOS2 range 0x2000 ... 0x2FFF are different by the EPOS4 object dictionary and it has to be checked if objects with a similar functionality still exist. Typically these might have been moved to different object numbers in the range of 0x2000.... 0x2FFF or 0x3000 ... 0x3FFF due to EtherCAT compatibility reasons.

Practical tip:

The best way to find a correlation in between moved EPOS2 objects is to search for the EPOS2's object name (or part of it) in the "EPOS4 Firmware Specification". Be aware that updated objects might not have exactly the same naming and can have a slightly differing functionality although the naming is similar.

Additional information:

Details about all objects can be found in the "EPOS4 Firmware Specification.pdf".

3. Operating modes

The base profile oriented operating modes like "Homing Mode", "Profile Position Mode" and "Profile Velocity Mode" are mainly the same for EPOS2 and EPOS4.

The maxon **specific EPOS2 operating modes "Position Mode", "Velocity Mode", and "Current Mode" do not exist anymore by the EPOS4 and have been replaced by the similar (but not fully identical) standardized DS402 operating modes "CSP - Cyclic Synchronous Position", "CSV - Cyclic Synchronous Velocity", and "CST - Cyclic Synchronous Torque"**. Commanding of these new modes might slightly differ due to new standards. Especially the EPOS2's "Current mode" and EPOS4's "CST - Cyclic Synchronous Torque" mode differ concerning the set value. The EPOS4's "CST mode" is commanded by a relative motor torque value and not the motor current (like in EPOS2's "Current mode").

Restrictions:

The EPOS2 operating modes "**IPM - Interpolated Position Mode", "Master Encoder Mode", "Step-/Direction Mode**" are **not available by the EPOS4**.

It is not possible to command an EPOS4 target position by an analog set value. Analog commanding of the EPOS4 is just possible in case of current or velocity control control.

The function "**Position Compare**" is **not available by the EPOS4**.

Practical tip:

The best way to learn about the commanding of the different operating modes and related objects is to have a look in the following EPOS4 documentation:

- "EPOS4 Application Notes Collection.pdf"
-> Chapter 5 "Device Programming"
- "EPOS4 Firmware Specification.pdf"
-> Chapter 3 "Operating Modes"

4. RS232 protocols

The EPOS2 and EPOS4 are based on different RS232 protocols. If there is a RS232 communication library in use which was designed for the EPOS2, the RS232 protocol implementation has to be adapted for the EPOS4. The EPOS4 uses the same protocol for USB and RS232 which helps to reduce the effort of protocol implementations.

Additional information:

Please find more information about EPOS4's RS232 protocol and the differences compared to EPOS2 here:

- "EPOS4 Communication Guide.pdf"
-> Chapter 2 "USB & RS232 Communication"
- "EPOS4 Application Notes Collection.pdf"
-> Chapter 3 "Comparison of maxon Serial Protocols for RS232"

5. EPOS2 libraries

3rd party libraries which have been designed for the EPOS2 might not fully work for EPOS4 anymore. There have to be some modifications of the library's source code required due to the points (1., 2., 3., 4.) mentioned above.

Practical tip:

maxon's EPOS Command library (for Windows and Linux based PCs and Raspberry Pi) have been adapted for the use with EPOS2 or EPOS4. If you use maxon's "EPOS Command library" or maxon's LabView Library / VIs for PC, most of the work and adaptation has been already done and you can switch more easily your master program code in between the EPOS2 and EPOS4. Anyway there are even some limitations by the "EPOS Command library" too due to the partly different operating modes and features.

Important restriction: NI cRIO library for EPOS4 -> No!

The maxon **EPOS2 library for NI's CompactRIO is not(!) compatible with EPOS4**, i.e. this library has not(!) been adapted to the updated EPOS4 operation modes and the EPOS4 Object Dictionary. This means that only some few low level VIs of the cRIO library (like SDO and PDO access) can still be used, but especially **any functions demanding for EPOS2's "Position Mode", "Velocity Mode", and "Current Mode" operating modes** or using specific EPOS2 objects (which are no present by the EPOS4 anymore) **cannot(!) be used with the cRIO library in case of an EPOS4**.

maxon does not support NI's CompactRIO based systems for EPOS4 anymore!

Additonal information:

Please have a look in the following documentation to find out what function calls are available depending on the EPOS product line (EPOS2, EPOS2, EPOS4) in use:

- "EPOS Command Library.pdf"
-> Chapter "Appendix A - Hardware vs. Functions"

6. Control structure and parameters

The EPOS4 uses enhanced control structures and different control parameter scaling than the EPOS2. Tuning of control parameters is mandatory if an EPOS2 is replaced by an EPOS4. Due to EPOS4's enhanced control structures and more flexibility to implement new control designs in the future, the control parameters have been moved from the objects 0x60F6, 0x60F9, 0x60FB (in case of an EPOS2) to the objects 0x30A0, 0x30A1, 0x30A2, and 0x30A3 (in case of an EPOS4).

Additional information:

- "EPOS4 Application Notes Collection.pdf"
-> Chapter 2 "Controller Architecture"
- "EPOS4 Firmware Specifications.pdf"
-> "Current control parameter set"
-> "Position control parameter set"
-> "Velocity control parameter set"
-> "Velocity observer parameter set"

Conclusion:

There are a lot of points to think about in case of a broad information (like above) about the differences in between the EPOS2 and EPOS4. In practice not all of these are relevant for a concrete application. If you are aware of your master source code and operating modes in use, the effort for adaptation is often very limited and can be roughly estimated.

In case of a new design it is recommended to start with the EPOS4 from the early beginning. Comparing EPOS2 and EPOS4 there are some advantages by the EPOS4:

- More compact by design.
- Available as module, compact, and housed product types.
- USB, RS232, CAN, and EtherCAT interfaces.
- Fulfilling latest CANopen and EtherCAT standards.
- Highly sophisticated, optimized control algorithms in use.
- 2.5 times faster control than the EPOS2.
- Easy to integrate in a system design commanded by a PLC and standard NC libraries of the PLC supplier or by Windows or Linux based masters using maxon's EPOS Command library.