maxon

Motion Helper

A C++ example for the maxon Motion Library: connect to an ESCON2 and run a motor in Profile Velocity Mode.

The Motion Helper is two small C++17 classes, `MotionDeviceBase` and `MotionDevicePVM`, that wrap the C functions of the maxon *Motion Library* (`VCS2_*`) into a device object with a clear sequence: connect, select the operation mode, set the parameters, enable the power stage, command a velocity. The example program in `examples/profile_velocity_mode/` shows this sequence end to end and is meant to be copied as the starting point for your own program.

Scope of this version: ESCON2 over USB, Profile Velocity Mode (PVM), Windows, Visual Studio 2022 or 2026. Other operation modes and interfaces are not included; `MotionObjects.h` explains how to add objects for them.

- **Getting Started** - installing the Motion Library and Visual Studio, and using the Motion Helper in your own CMake project.
- **Examples** - building the examples with Visual Studio, Visual Studio Code or the command line.
- **Example: Profile Velocity Mode** - what the example does, the device state machine in one minute, and what to do when something goes wrong.

Quickstart (run the PVM example)

> **Warning**
>
> Running the example turns the motor shaft. Read the safety notes in the example's README first.

1. Unpack the ZIP archive from the maxon Support Portal, e.g. to `C:\dev\motion_helper\`.
2. Open a **Developer PowerShell for VS 2022** (or 2026) from the Start menu. It puts `cmake`, `ninja` and the compiler `cl.exe` on the PATH; a plain PowerShell does not work with the presets.
3. Configure, build and run:

```
cd C:\dev\motion_helper
cmake --preset x64-release
cmake --build build\x64-release
build\x64-release\examples\profile_velocity_mode\example_pvm.exe
```

Requires an ESCON2 connected over USB, the Motion Library installed (see Getting Started), and motor and sensor parameters already configured with Motion Studio. Motion Studio must be closed while the example runs, because it holds the USB port. The build copies `MotionCmd64.dll` next to the `.exe`; the

executable does not run without it.

If the configure step ever fails with `The CMAKE_C_COMPILER: cl.exe is not a full path`, you are not in a Developer PowerShell. Delete the `build\` folder before retrying: the failed run leaves a cache behind that later runs would reuse.

Contents of the archive

Path	What it is
`source/`	The Motion Helper: `MotionDeviceBase`, `MotionDevicePVM`, the object dictionary, the error type
`examples/profile_velocity_mode/`	The PVM example: `main.cpp`, its `CMakeLists.txt`, its README
`motion_helper.cmake`	CMake module that finds the Motion Library and defines the targets `maxon_MotionLibrary` and `maxon_MotionHelper`
`CMakeLists.txt`, `CMakePresets.json`	Top-level project that builds all examples; the four presets
`documentation/`	Getting Started and the build instructions

The same documentation is included as PDF in the archive.

© 2026 maxon. All rights reserved.