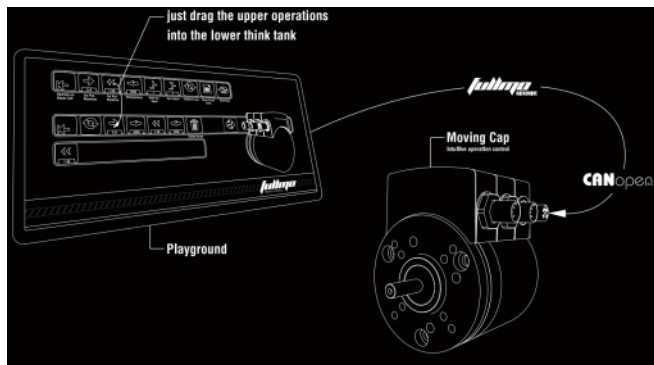


# MovingCap 349 Playground Demo

Author: Oliver Heggelbacher, rev: 05.09.2019  
[www.fullmo.de](http://www.fullmo.de) / [www.kickdrive.de](http://www.kickdrive.de)



The 24 VDC power connector for the **MovingCap 349** compact drive is on the side of the drive shaft (the angle plug in the picture below).



## Safety Notices



### For Qualified Personnel only

While the „Playground Demo“ has some toy-like appeal, it is not a toy.

**Fullmo MovingCap** is a compact drive for **industrial applications**. Severe bodily injury or considerable damage to property will result if the appropriate safety measures are not taken.



Fullmo MovingCap drives may be preloaded with Python script programs and may **start moving immediately** after connecting the power supply.

Start the fullmo **DriveScriptPlayground** demo on your Windows Tablet or PC. You can download the current version from here:

[www.kickdrive.de/sw/DriveScriptPlayground.exe](http://www.kickdrive.de/sw/DriveScriptPlayground.exe)



Additional documentation (German) on MovingCap 349 can be found at:

<https://fullmo.de/rotative-antriebstechnik/>

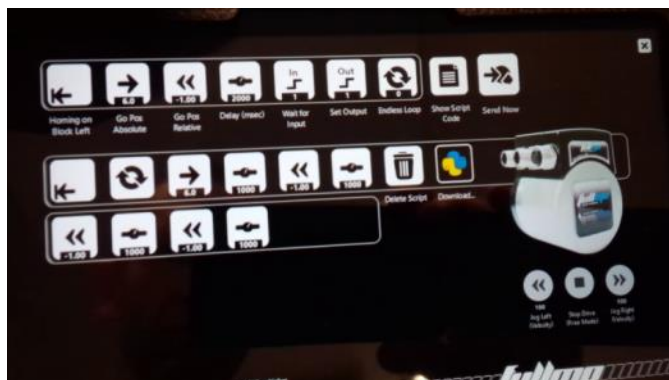
Connect the black Fullmo **USB2Drive** device via USB to your PC. Connect the USB2Drive via the CAN Bus cable (usually purple color) to the drive (the middle M12 connector).



The **Send Now** tile on the top right (re-)loads the default example sequence into the drive.



If a problem is reported here, check according to [fullmoBasicSetupMovingCap349\\_en.pdf](#) the basic settings of the drive (**baud rate = 125K, NodeId = 1**).

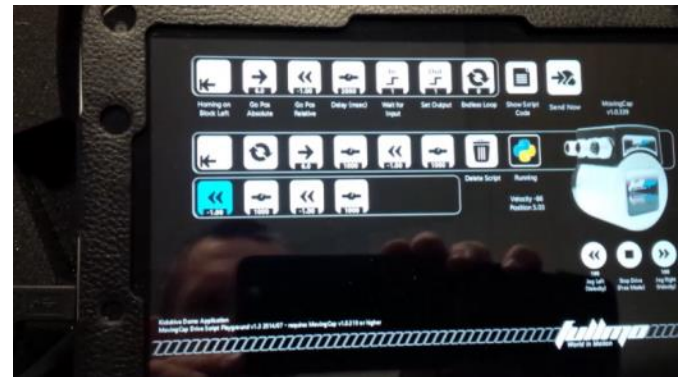


The now performs referencing, i.e. it tries to detect the mechanical limit stop.



For referencing, the drive operates with reduced torque (for example, parameter **Max\_current 6073h.00h = 150** during homing, corresponding to 15% of nominal torque). If no actual mechanical limit is present, referencing is aborted / skipped after approx. 30 seconds.

After successfully completing the homing/referencing procedure, the drive functions after the **Endless Loop** tile (two round arrows) are executed.



Individual tiles from the program flow can be moved to other positions or moved to the trash.



Additional drive functions, such as **Go Pos Absolute** can be configured at the top row: point to the numeric value to edit the position.



Then drag the tile from the top into the “program flow” below.



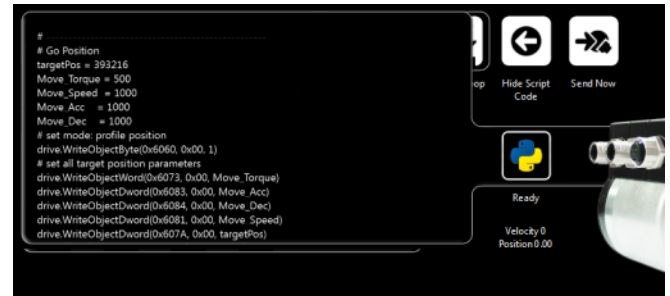
You can insert new elements at any position or move their position in a later step.



After the tile arrangements / drive sequence has been modified, the Python script program is automatically re-written and transmitted into the drive (**Download...**). The drive then re-starts the Python program at the first tile / drive function.



Btw – the actual python program code can be shown using the **Show Script Code** button.



Congratulations – you have just written your **first Python program** for drive control!

Real-world applications are usually more complex than what can be made with the predefined tile functions. They usually include i/o signaling and coordination with a master control. The Playground Demo is intended to provide you a starting point and example code, but not the full end application.

A real-world application for teaching 4 single positions and using I/O control to position can be found here: <http://www.kickdrive.de/sw/KickdriveApp-4SlotFeeder-deutsch.exe> (German only for now.)



Complete drive configuration including a scope functions and full CANopen object editor is offered by our **fullmo Kickdrive** software. For more information see [www.kickdrive.de](http://www.kickdrive.de).

Additional documentation (German) on MovingCap 349 can be found at:

<https://fullmo.de/rotative-antriebstechnik/>