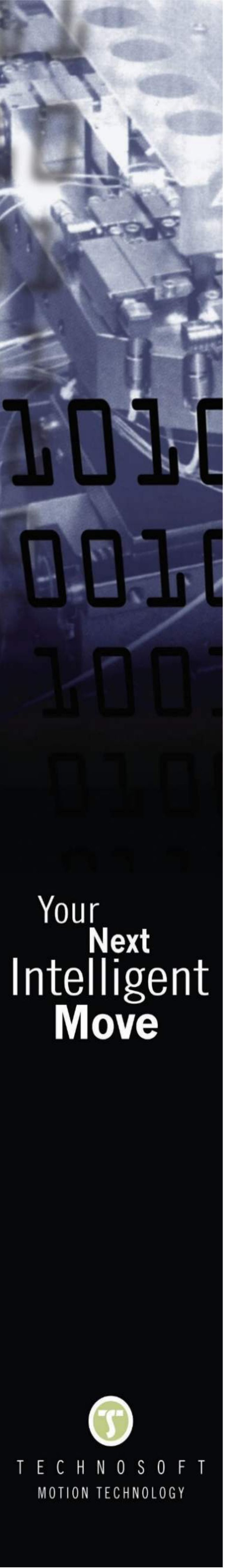


TECHNOSOFT
MOTION TECHNOLOGY

Motion start/stop using two digital inputs

Application Note

Easy Motion Studio II

Table of contents

1. Application description

This application note describes how to use two digital inputs (IN0 and IN1), to start/stop the motion profile.

2. Application flow chart

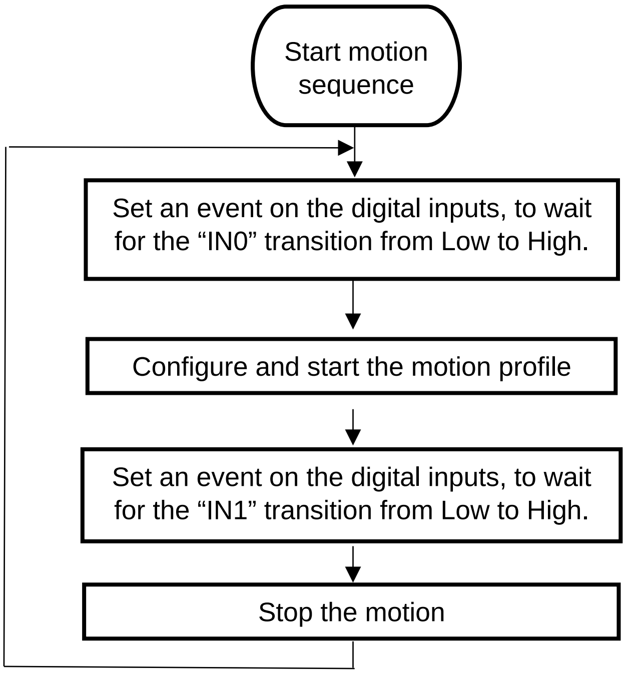
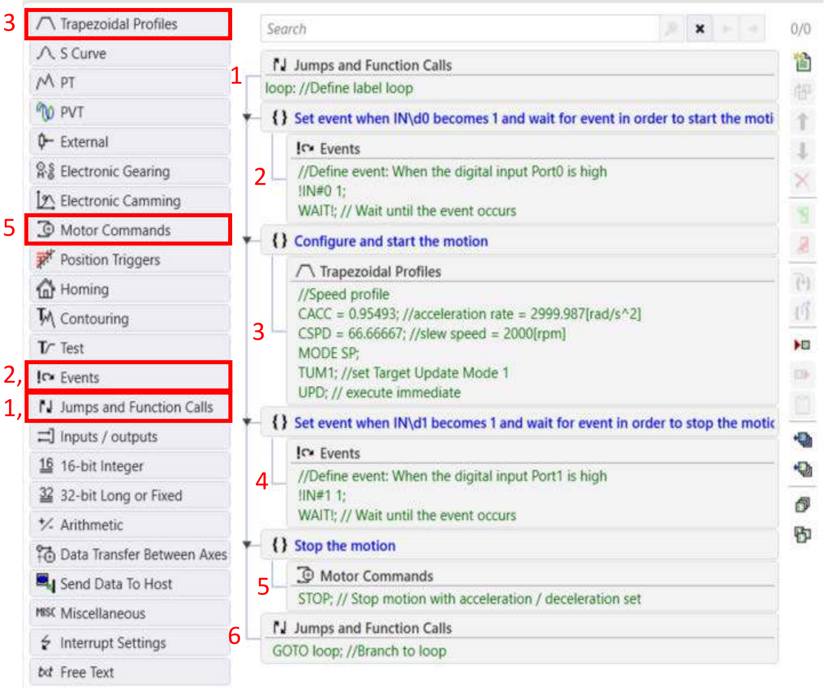

Start motion sequence
Set an event on the digital inputs, to wait for the "IN0" transition from Low to High.
Configure and start the motion profile
Set an event on the digital inputs, to wait for the "IN1" transition from Low to High.
Stop the motion

Figure 1. Application flow chart

3. EasyMotion Studio II implementation



```
Jumps and Function Calls
loop: //Define label loop
{ } Set event when IN\d0 becomes 1 and wait for event in order to start the moti
    Events
    //Define event: When the digital input Port0 is high
    !IN#0 1;
    WAIT!; // Wait until the event occurs
{ } Configure and start the motion
    Trapezoidal Profiles
    //Speed profile
    CACC = 0.95493; //acceleration rate = 2999.987[rad/s^2]
    CSPD = 66.66667; //slew speed = 2000[rpm]
    MODE SP;
    TUM1; //set Target Update Mode 1
    UPD; // execute immediate
{ } Set event when IN\d1 becomes 1 and wait for event in order to stop the motic
    Events
    //Define event: When the digital input Port1 is high
    !IN#1 1;
    WAIT!; // Wait until the event occurs
{ } Stop the motion
    Motor Commands
    STOP; // Stop motion with acceleration / deceleration set
Jumps and Function Calls
GOTO loop; //Branch to loop
```

Figure 2. EasyMotion Studio II implementation

4. Detailed description of the EasyMotion Studio II implementation

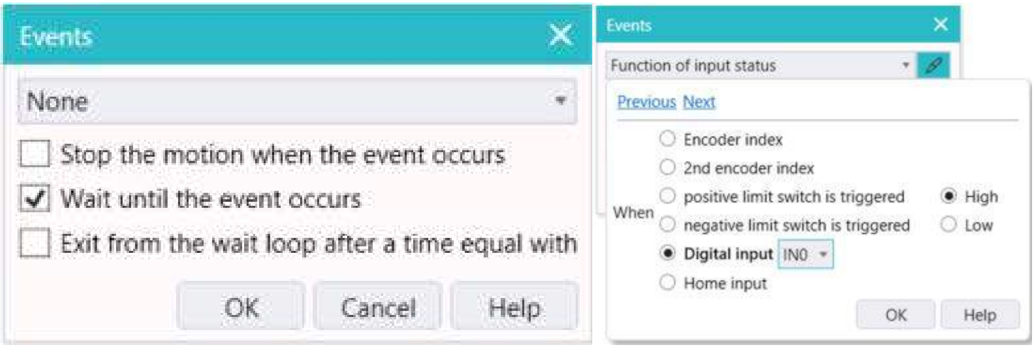
The motor is rotated using a speed profile, that starts when the “IN0” digital input goes High and stops when the “IN1” digital input becomes active (goes High).

The inputs status is monitored using an event on the digital inputs that can be generated using the “Events” dialog.

Now, we will cover each step illustrated in Figure 2, in relative order of appearance.

a) The “Events” dialogue allows defining a programmable condition and can be opened by clicking on the button (Figure 2 – step 2 and step 4).

   The “Change Event” button in the “Events” dialog will open the “Event Type” window that allows to select the needed condition and to configure it. In this application example, the “***Function of input status”*** event type was selected and set to monitor the “IN0” digital input and then the “IN1” digital input, to stop the motion.



Events

None

□ Stop the motion when the event occurs

☑ Wait until the event occurs

□ Exit from the wait loop after a time equal with

OK Cancel Help

Events

Function of input status

Previous Next

When

○ Encoder index

○ 2nd encoder index

○ positive limit switch is triggered ● High

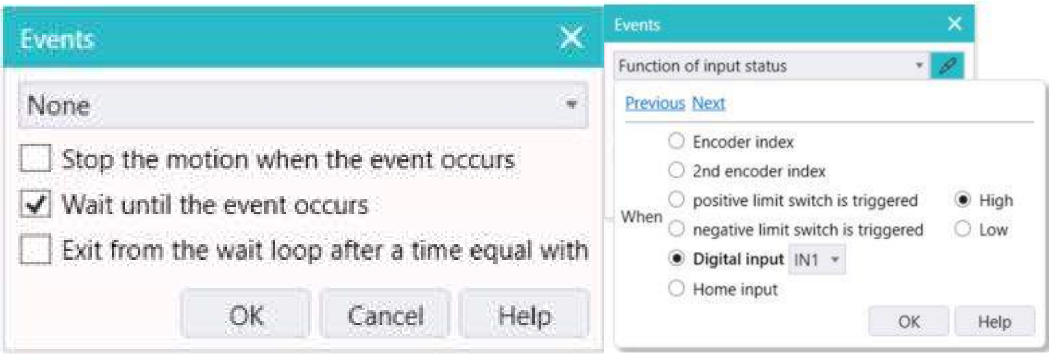
○ negative limit switch is triggered ○ Low

● Digital input IN0

○ Home input

OK Help

Figure 3. Set an event on the “IN0” digital input (Figure 2 – step 2)



Events

None

□ Stop the motion when the event occurs

☑ Wait until the event occurs

□ Exit from the wait loop after a time equal with

OK Cancel Help

Events

Function of input status

Previous Next

When

○ Encoder index

○ 2nd encoder index

○ positive limit switch is triggered ● High

○ negative limit switch is triggered ○ Low

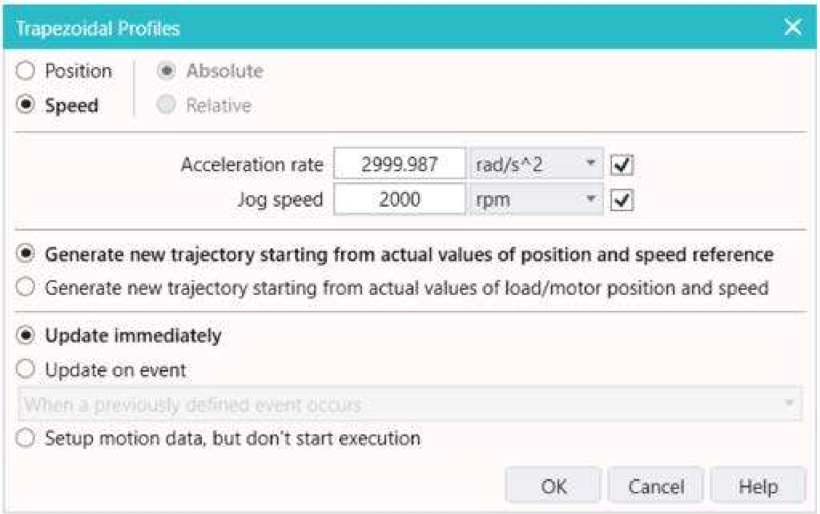
● Digital input IN1

○ Home input

OK Help

Figure 4. Set an event on the “IN1” digital input (Figure 2 – step 4)

b) The “Motion – Trapezoidal Profiles” dialogue (Figure 2 – step 3) allows to program a position or speed profile with a trapezoidal shape of the speed, due to a limited acceleration. In this case, it was used to insert the speed profile.



Trapezoidal Profiles

○ Position
● Speed

● Absolute
○ Relative

Acceleration rate 2999.987 rad/s^2 ☑
Jog speed 2000 rpm ☑

● Generate new trajectory starting from actual values of position and speed reference
○ Generate new trajectory starting from actual values of load/motor position and speed

● Update immediately
○ Update on event
When a previously defined event occurs
○ Setup motion data, but don't start execution

Figure 5. How to configure a speed profile (Figure 2 – step 3)

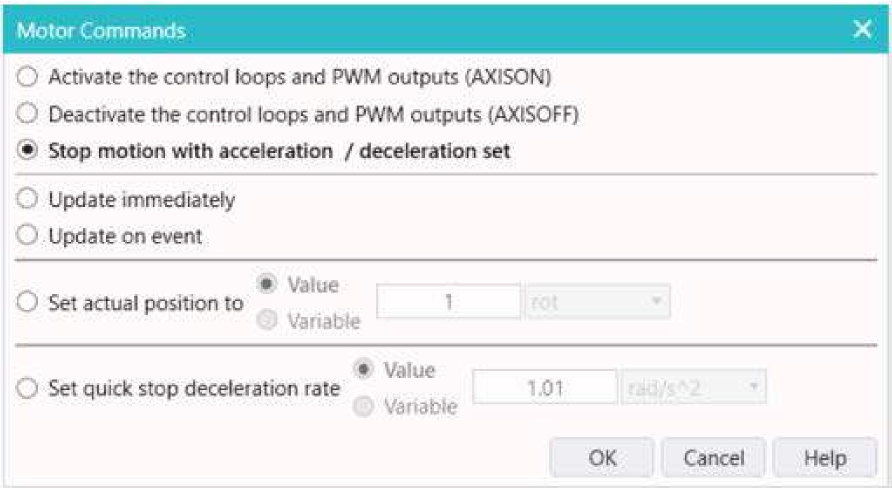
Choosing the speed profile, the load/motor is controlled in speed. The “jog speed” (speed sign specifies the rotation direction) and the “acceleration rate” are the input parameters for this motion mode.

Once the profile is executed, the load/motor accelerates until the jog speed is reached.

Remark: The speed loop must be enabled in the Control Settings and tuned in order to be able to program a speed profile.

c) The “Motion - Motor Commands” dialogue (Figure 2 – step 5), was used to stop the motor, after the “IN1” digital input transition was detected by the event described above.

The “***Stop the motor with acceleration / deceleration set”*** TML command will stop the motor using a deceleration equal with the “***CACC***” (acceleration rate) parameter value. In this case the “***CACC***” (acceleration rate) parameter value is the one that was set in the “Trapezoidal Profiles” dialogue (Figure 2 – step 3) described above (CACC = 3000 rad/s^2).



Motor Commands

○ Activate the control loops and PWM outputs (AXISON)
○ Deactivate the control loops and PWM outputs (AXISOFF)
● Stop motion with acceleration / deceleration set

○ Update immediately
○ Update on event

○ Set actual position to ● Value ○ Variable 1 rot

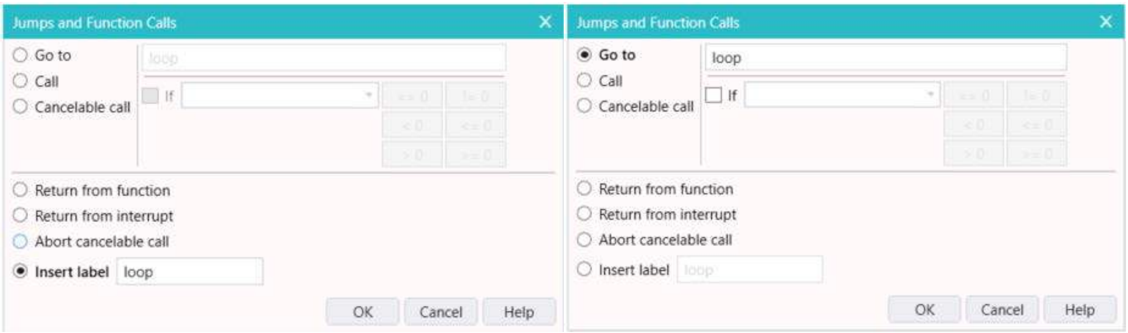
○ Set quick stop deceleration rate ● Value ○ Variable 1.01 rad/s^2

Figure 6. How to stop the motion using a controlled deceleration. (Figure 2 – step 5)

d) For the program to run continuously, the code above was introduced in an infinite loop that was created using the “Jumps and Function Calls” dialogue (Figure 2 – step 1 and step 6).

Remark: The "Jumps and Function Calls" dialogue allows to control the TML program flow through unconditional or conditional jumps and unconditional, conditional or cancelable calls of TML functions.

The infinite loop was created using a label and a "***GOTO***" TML instruction that makes the program jump to the respective label.



● Insert label loop

● Go to loop

Figure 7. **How to create a loop using a label and a "*GOTO*" TML instruction (Figure 2 – step 1 [left] and step 6 [right])**

The label name can be any string of up to 32 characters, which starts with an alphanumeric character or with underscores. In this case the label was named "loop" and placed at the beginning of the TML application program.

The "***GOTO***" TML instruction indicates the jump address as address, label or address set in variable. In this case it was specified as the label named ("loop").

5. Conclusions

This application note shows how the TML code execution can be conditioned by two digital inputs but indirectly it also presents the PLC embedded functionality of the Technosoft drives.

Technosoft drives have a series of digital inputs and outputs that can be used to interconnect the drive with a master device or with an HMI. The digital inputs can be read individually or as a group therefore the master can use several digital inputs to binary code actions stored locally in the non-volatile memory of the drive. The number of digital inputs/outputs is product dependent.

Apart from digital inputs, each drive can have up to two analog inputs. These inputs can be used, for example, to interconnect various analog sensors or to impose the position / speed reference from a potentiometer or from a device that outputs an analog signal.