



T E C H N O S O F T  
MOTION TECHNOLOGY

## Drive special inputs Limit switches

Application Note

## Easy Motion Studio II

Your  
Next  
Intelligent  
Move



T E C H N O S O F T  
MOTION TECHNOLOGY



# Table of content

<b>1. Introduction</b> .....	3
<b>2. Hardware limit switches</b> .....	3
<b>2.1 Limit switches inputs default behavior, configuration and usage</b> .....	3
<b>2.2 Limit switches related interrupts</b> .....	6
<b>2.3 Limit switches inputs usage as general-purpose inputs</b> .....	7
<b>3. Software limits configuring and usage</b> .....	9

## 1. Introduction

This application note describes the default functionality of limit switch inputs and explains how to disable it to use the respective digital inputs as general-purpose inputs. T

The document also details how to restrict load movement using the software limit switches option.

## 2. Hardware limit switches

Limit switches are control components used to determine the presence, passing, positioning, and end of travel of an object. The simplest version is a push button, but in complex applications, optical, induction, or capacitive sensors are used.

### 2.1 Limit switches inputs default behavior, configuration and usage

Present on all Technosoft servo drives, the “IN2/LSP” (positive limit switch) and “IN3/LSN” (negative limit switch) digital inputs allow connecting the system limit switch sensors.

When one of the limit switches became active, the drive performs the following default actions:

- stops the motion, using a QUICK STOP profile
- sets the correspondent bit in the MER error register (bit 6 - LSP active / bit 7 - LSN active)
- executes the code inside the correspondent limit switch interrupt routine, if it is enable and set to trigger on the inactive to active limit switch transition
- prevents any motion that moves the load against the active limit switch

The limit switches active level can be set during the drive commissioning through the “Inputs/ outputs” page in the “Setup tree”.

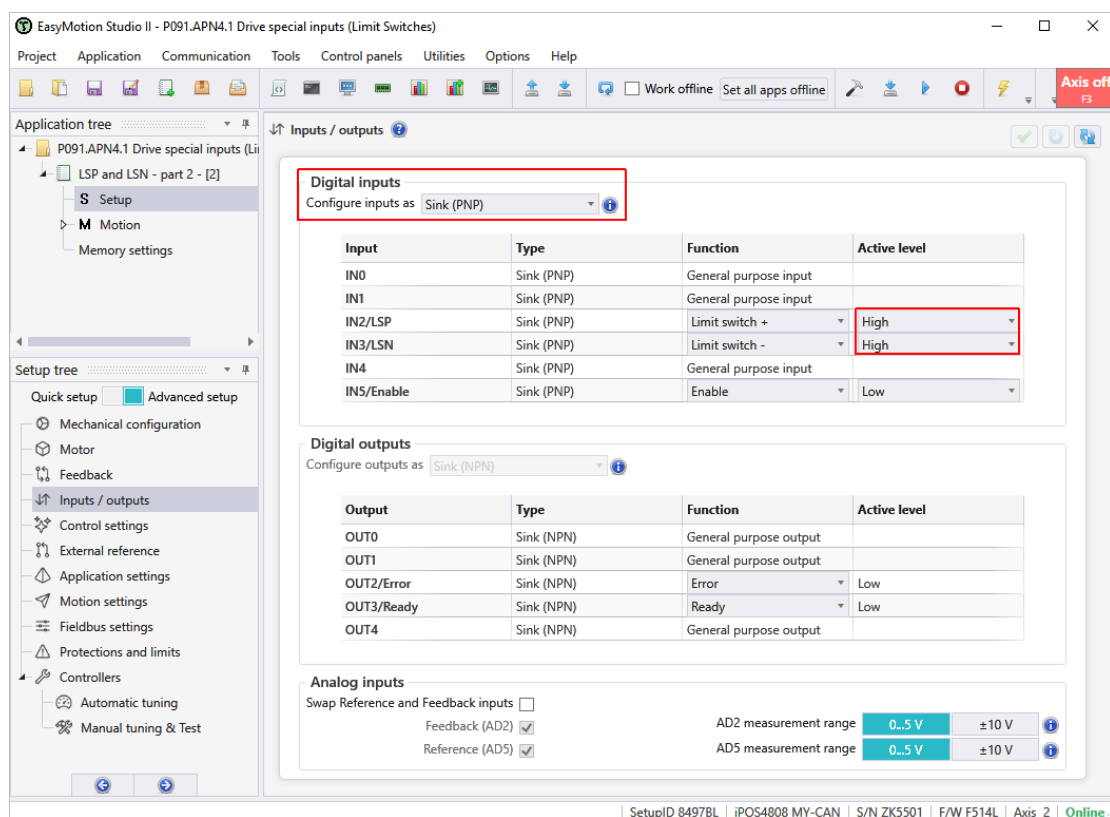


Figure 1 - Limit switches active level selection

**Remark** If the drive hardware design allows configuring the digital inputs type (NPN or PNP), it can also be done through the "Inputs/Outputs" page and applies to the limit switch inputs as well.

The default behavior of the limit switches can be easily observed by plotting the position, speed, and digital inputs status (or only the used limit switch) in Logger or Scope.

For example, the drive was programmed to perform a trapezoidal position profile as in the picture below.

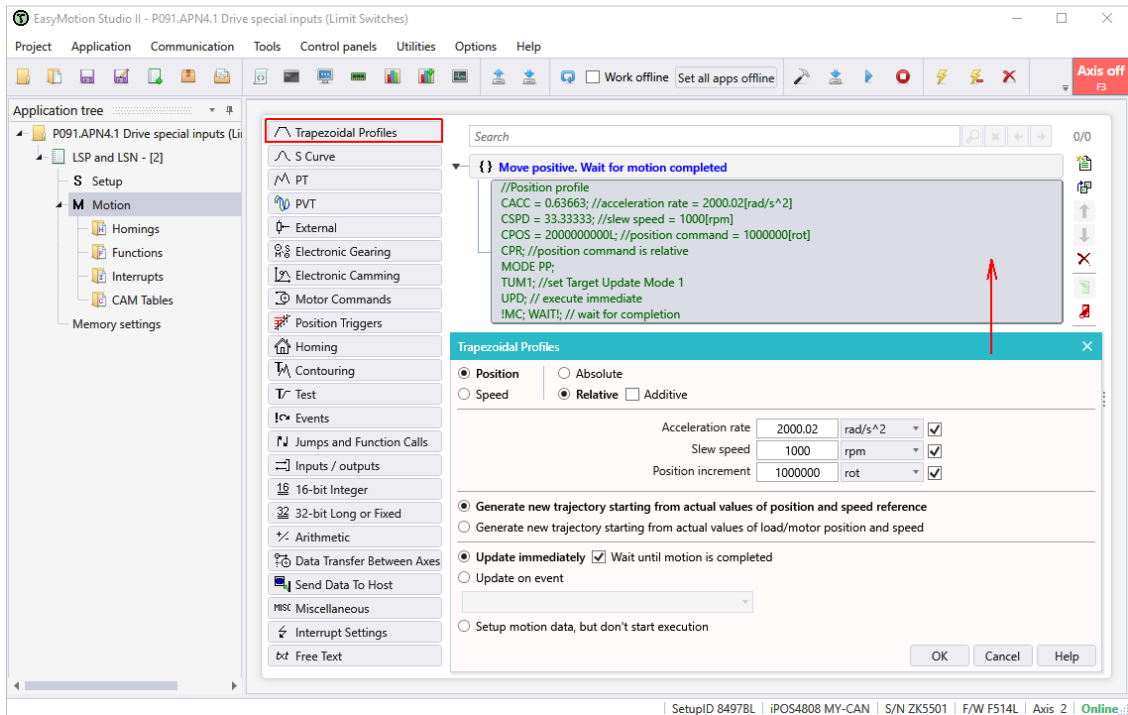


Figure 2 – Motion profile example

After running the program, the positive limit switch (IN2/LSP) was manually triggered. As a result, the motion stopped. The position, target speed and digital outputs status was plotted using the Logger tool.

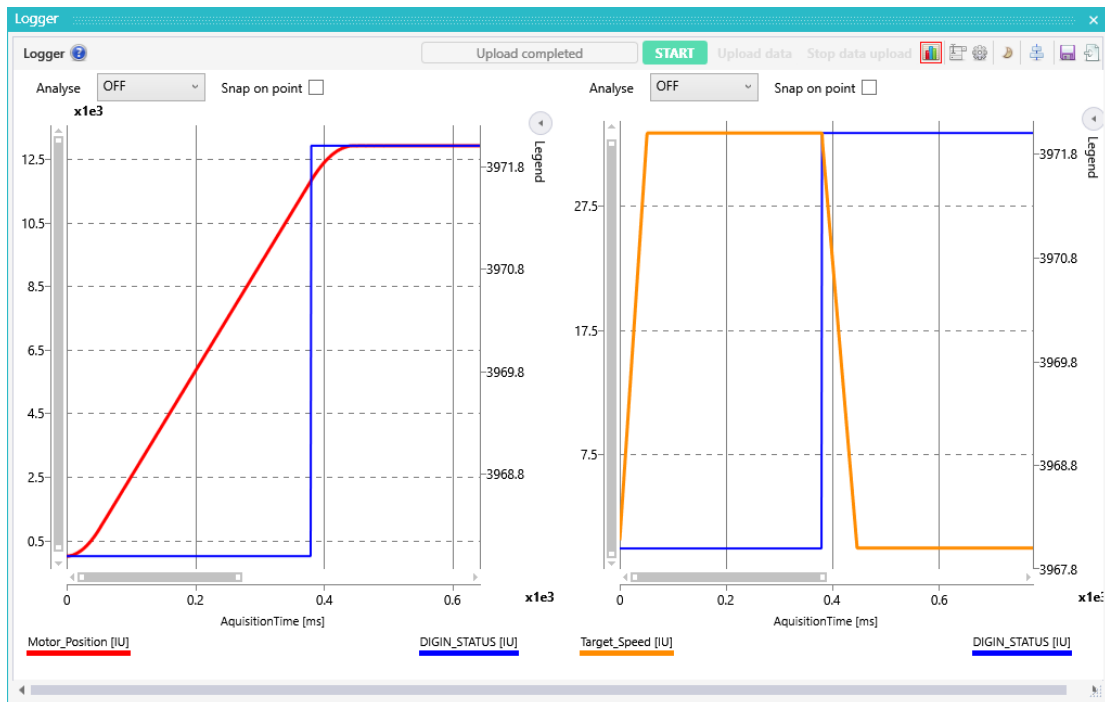


Figure 3 – Positive limit switch default behavior

**Remark** The Logger setup and usage is detailed in the EasyMotion Studio II help topics. The related topic can be open by pressing the “?” sign in the Logger window (upper left corner).

The first plot shows the motor position (red line) starting from 0 IU and following the imposed reference. When the positive limit switch (blue line) became active, the drive automatically executed a quick stop profile, stopping the motion.

The second plot shows the target speed (orange line) that goes to 0 IU, when the limit switch (blue line) triggers, with the set quick stop deceleration.

**Remark** If needed, the quick stop deceleration value can be set / changed through the “Motor Commands” wizard dialogue.

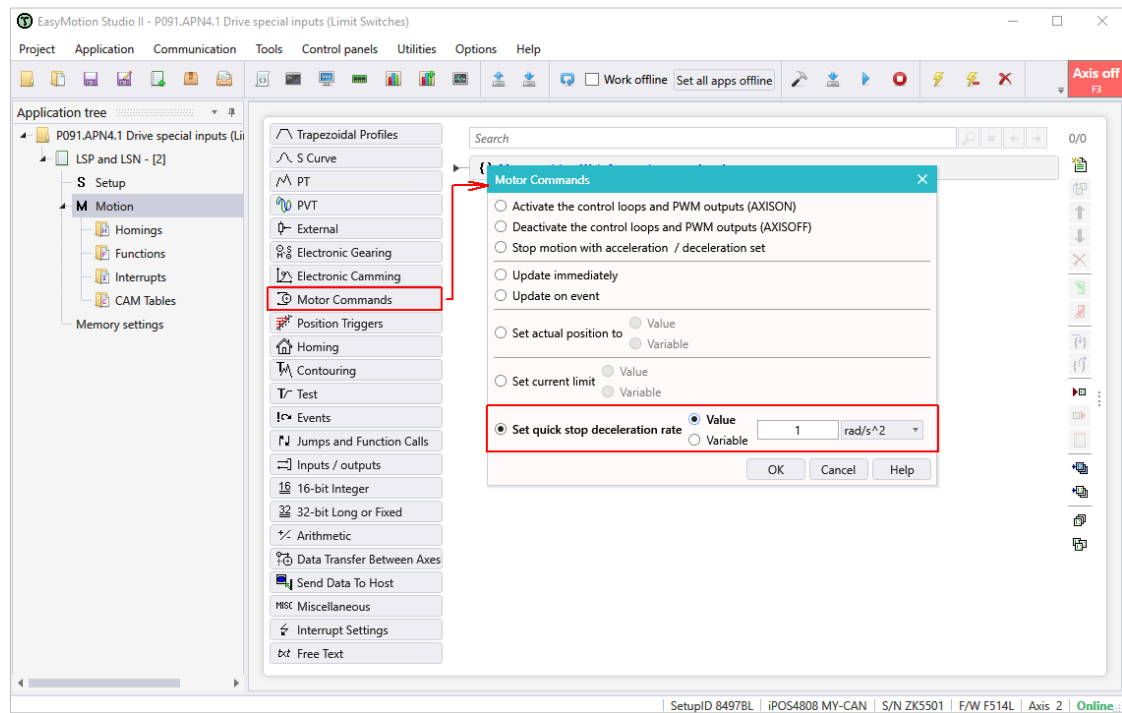


Figure 4 – Quick stop deceleration rate adjustment

The status of the limit switches is included in the MER error register and can be visualized through the "Drive Status" control panel. The screenshot below was taken while performing the above example.

SRH - Status Register High		SRL - Status Register Low		MER - Error Register		DER - Detail Error Register		DER2 - Detail Error Register 2	
15 - Fault	0	15 - Axis is ON	1	15 - Enable is inactive	0	15 - EEPROM Locked	0	15 - Output frequency	0
14 - In Cam	0	14 - Event set has occurred	0	14 - Command error	0	14 - ENA hardware error	0	14 - Reserved	0
13 - In freeze control	0	10 - Motion is completed	1	13 - Under voltage	0	13 - Self check error	0	13 - Reserved	0
12 - In Gear	0	8 - Homing/CALLS active	0	12 - Over voltage	0	12 - TML Heartbeat ignored	0	12 - Reserved	0
11 - I2t warning - Drive	0	7 - Homing/CALLS warning	0	11 - Over temp. - Drive	0	11 - Start mode failed	0	11 - Reserved	0
10 - I2t warning - Motor	0	Registers legend:		10 - Over temp. - Motor	0	10 - Encoder broken wire	0	10 - Reserved	0
9 - Target reached	1	1 - Yes / True		9 - I2t	0	9 - UPD ignored for S-curve	0	9 - Reserved	0
8 - Capture event/interrupt	0	0 - No / False		8 - Over current	0	8 - Invalid S-curve profile	0	8 - Reserved	0
7 - LSN event/interrupt	0	ENDINIT		7 - LSN (limit -) active	0	7 - Software LSN active	0	7 - Reserved	0
6 - LSP event/interrupt	0	IMPORTANT! Check SRH.0 !		6 - LSP (limit +) active	1	6 - Software LSP active	0	6 - Position wraparound	0
5 - Autorun enabled	1	Supply voltage and some status		5 - Feedback error	0	5 - Cancelable call ignored	0	5 - Hall sensor missing	0
4 - Over position trigger 4	1	or error bits are set ONLY after		4 - Serial comm. error	0	4 - UPD ignored	0	4 - AEI interface error	0
3 - Over position trigger 3	1	ENDINIT is executed.		3 - Control error	0	3 - Function not available	0	3 - BiSS sensor missing	0
2 - Over position trigger 2	1	If SRH.0 = 0, download a setup,		2 - Invalid setup data	0	2 - Homing not available	0	2 - BiSS data error	0
1 - Over position trigger 1	1	reset the drive and press the		1 - Short-circuit	0	1 - TML stack underflow	0	1 - BiSS data warning	0
0 - ENDINIT executed	1	nearby button to send an ENDINIT		0 - CANbus error	0	0 - TML stack overflow	0	0 - BiSS CRC error	0
Supply voltage [V]	24.007	Drive temperature [°C]	49.443						

Figure 5 - Drive Status control panel that shows the MER error register

**Remark** As a default drive behavior, while the positive limit switch is active, the drive accepts only negative position or speed commands.

## 2.2 Limit switches related interrupts

The TML interrupts are special routines that the drive can execute automatically when the triggering condition (continuously monitored by the drive) occurs. A detailed description of the TML interrupts can be found in the related EasyMotion Studio II help topic or in the [“TML interrupts usage”](#) application note.

The available interrupts are displayed when selecting the “Interrupts” subtree item. The “Interrupts” page also allows setting one or more interrupts as “user-defined” or reloading the default code.

For limit switch inputs, the associated interrupts are: **“int6 – LSP programmed transition detected”** and **“int7 – LSN programmed transition detected”**.

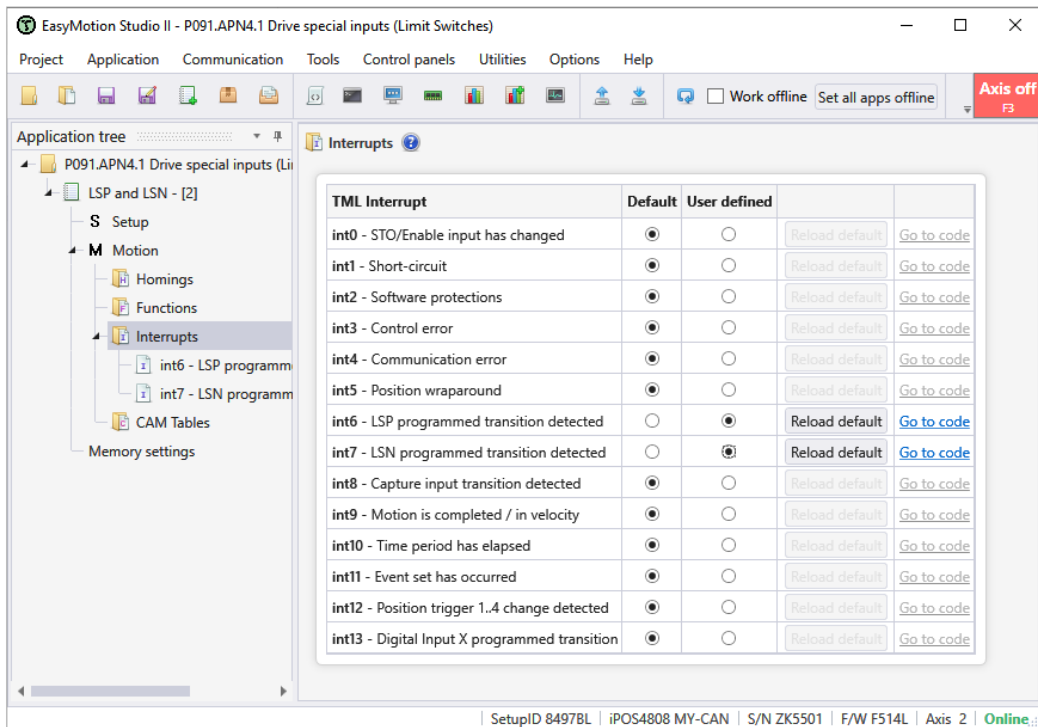


Figure 6 – Limit switches related interrupt routines

When selecting an interrupts from the list, the related code page will open, allowing to modify the existent code (if any) or to implement the needed custom functionality.

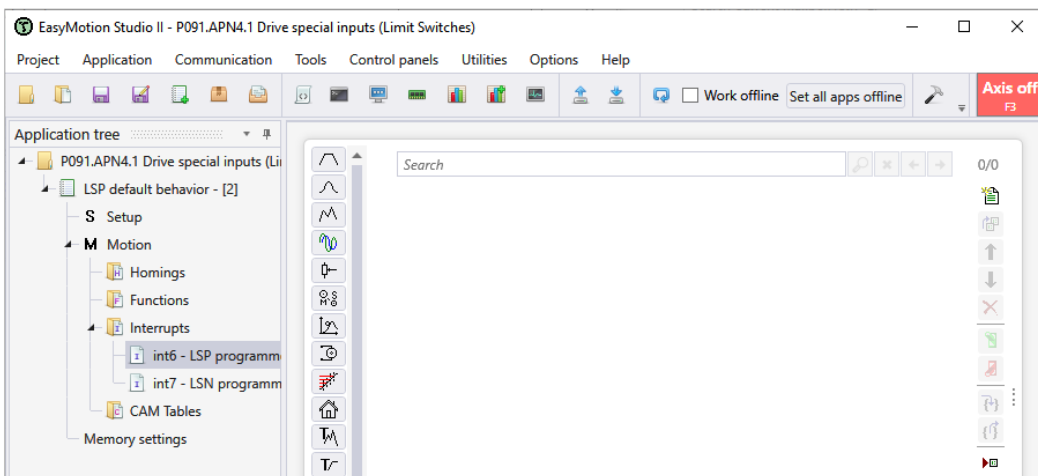


Figure 7 – LSP limit switches interrupt routine body

For limit switch interrupts, there is no code inside the interrupt body, as the related functionality is implemented at the firmware level.

The interrupts can be enabled and configured through the “Interrupts Settings” wizard.

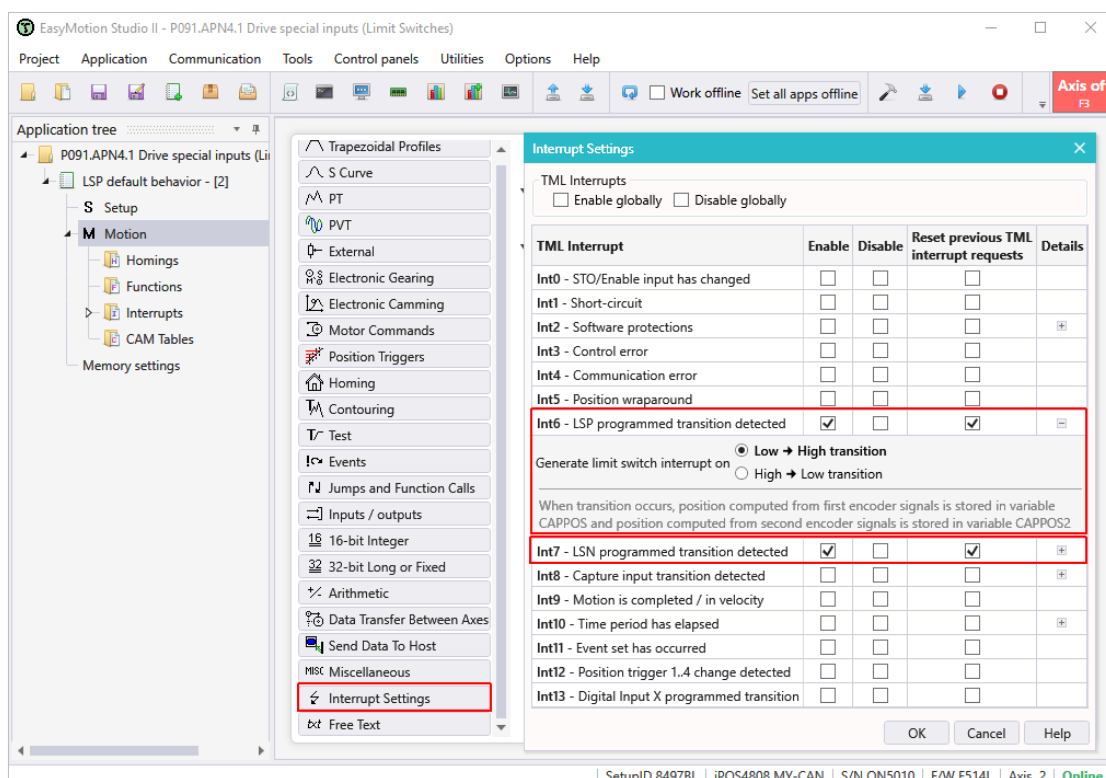


Figure 8 – Interrupts settings wizard dialogue

For limit switch interrupts, the wizard also allows setting the input transition that will generate the related interrupt.

### 2.3 Limit switches inputs usage as general-purpose inputs

The functionality of the limit switch inputs ("IN2/LSP" and "IN3/LSN") can be set during drive commissioning through the "Inputs/Outputs" page in the "Setup" tree.

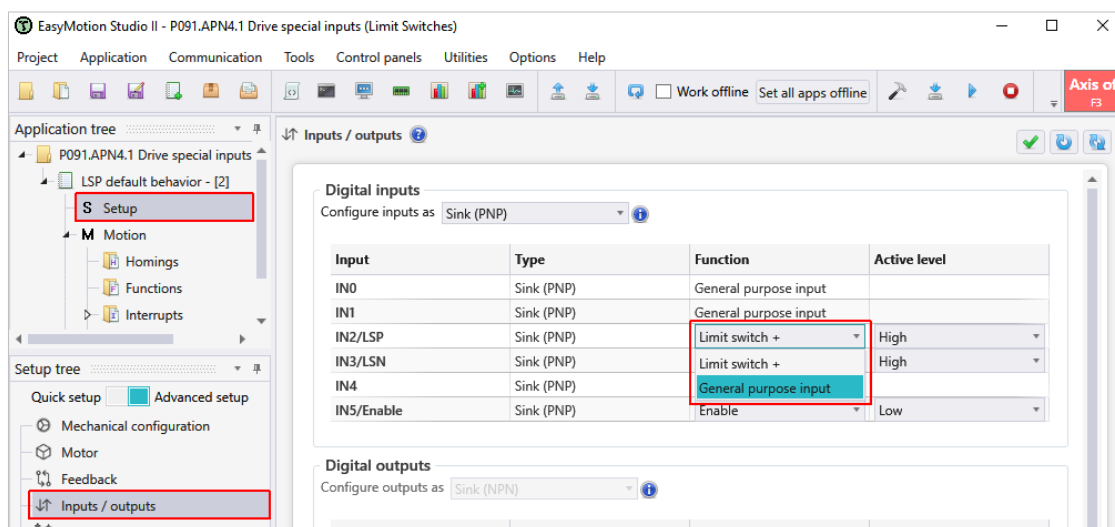


Figure 9 – Limit switches functionality selection

**Remark** When set as “general purpose inputs”, the limit switches default behavior is disabled but the related interrupt routines are still available. They can be enabled and used as when the limit switches are used with their default function.



If the application requires temporarily disabling the limit switch functionality (e.g. during the homing procedure), the “**LSACTIVE = 1;**” TML instruction can be used. This instruction disables the default functionality of the limit switches and can be inserted using the "Inputs/Outputs" or "16-bit Integer" wizard dialogues.

This instruction can also be sent via the communication line or write as a free text.

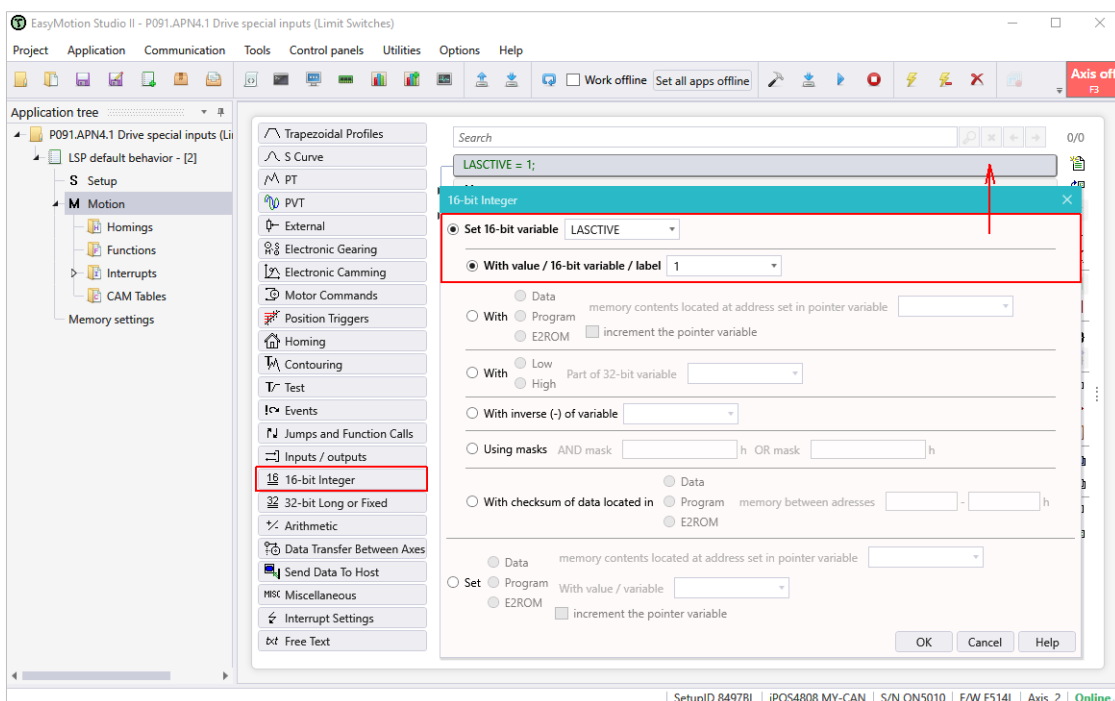
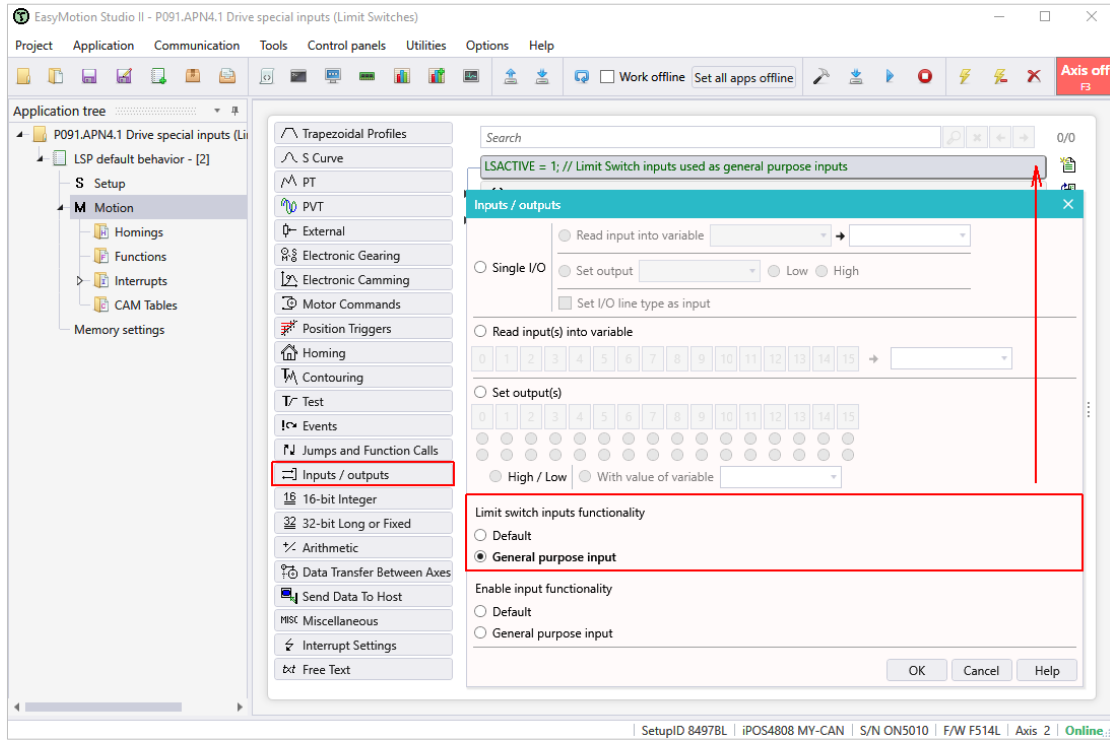


Figure 10 – Limit switches default functionality disabling

The limit switches default functionality can be restored using the “**LSACTIVE = 0;**” TML instruction, which can be generated using “Inputs / outputs” or “16-bit Integer” wizard dialogues.

If need, it can also be sent via the communication line or write in the TML application as free text.

### 3. Software limits configuring and usage

Software limits function similarly to hardware limit switches, restricting movement within set boundaries. This option can be enabled and configured through the "Protections and Limits" subtree item.

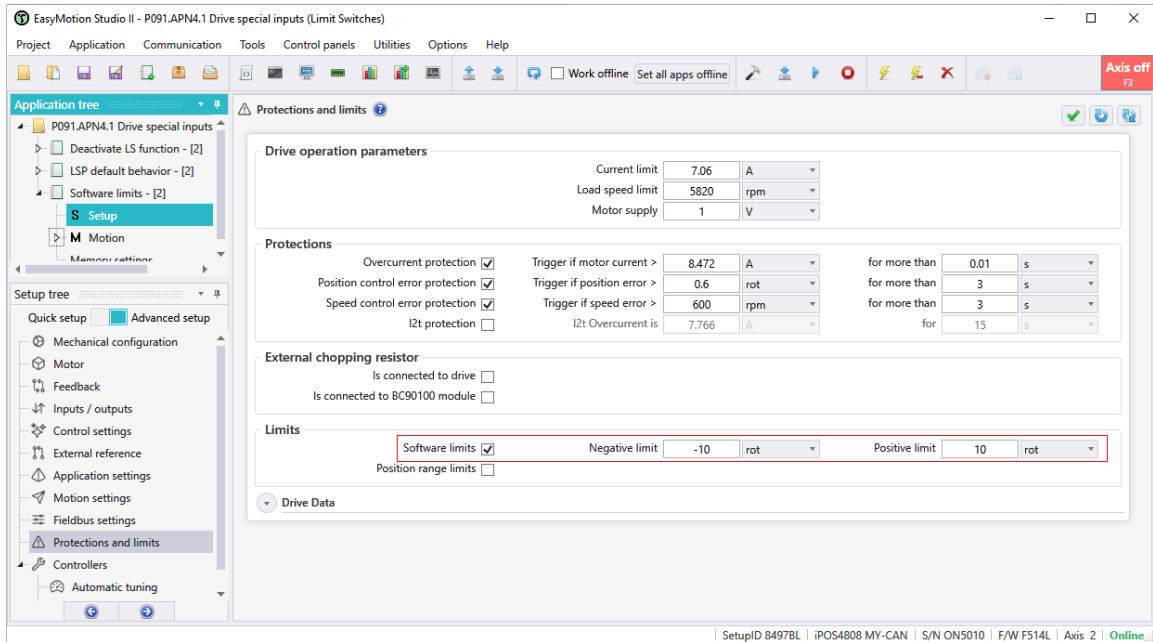


Figure 8 - Software limit switches configuration

When a software limit is reached, the drive stops the motion, using a quick stop profile, and sets the correspondent bit in the DER register.

For example purpose, the drive was set to perform 20 rot (using a relative trapezoidal position profile) while the software limits were set to -10 rot. .... +10 rot.

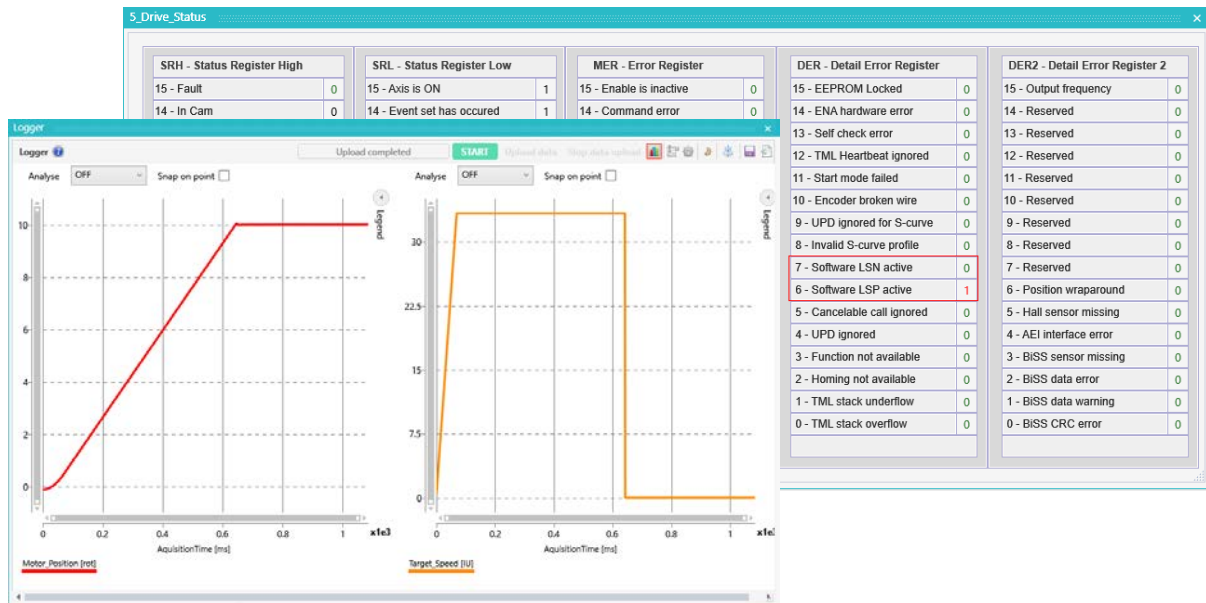


Figure 9 - Software limits default behavior

The above picture shows the motor position (red curve) that starts from 0 rot and increase according to the set profile. When reaching 10 rot, the motion is stopped as a result of reaching the positive software limit, and bit 6 in the DER register is set. The quick stop can also be observed in the second plot where the speed (orange line) drops to 0 IU, with the set quick stop deceleration value.

If the application requires enabling or disabling the software limits on the fly, it can be done through bit 12 in the ASR register:

- if **ASR.12 = 1**, the software limits are enabled
- if **ASR.12 = 0**, the software limits are disabled

The ASR register value can be changed through the “16-bit Integer” wizard, using the mask option.

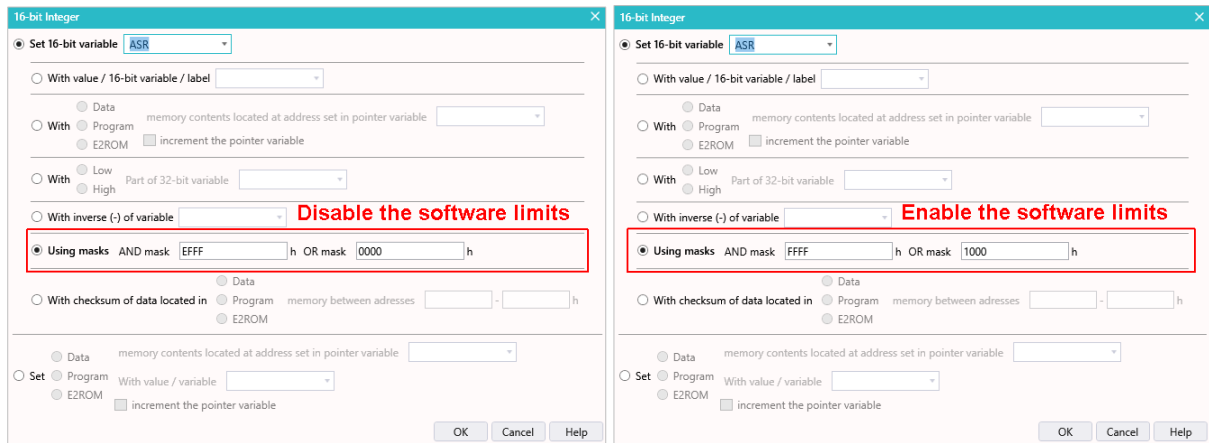


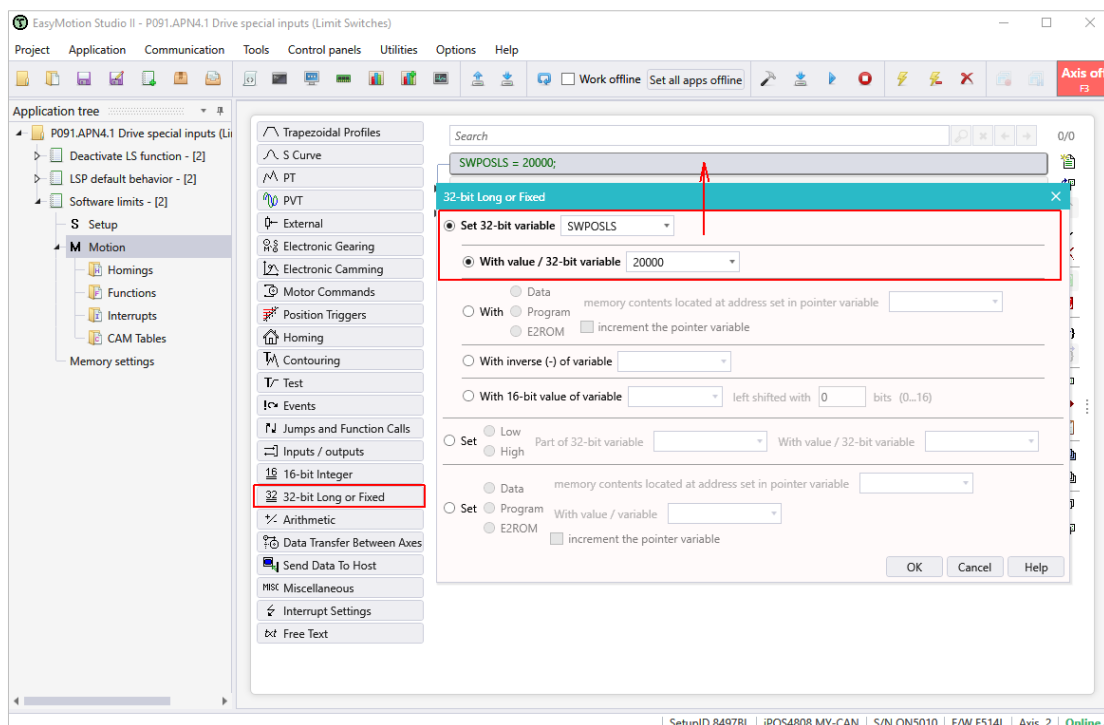
Figure 10 – On the fly software limits disabling/enabling

**Remark** The ASR register bits description is present in the EasyMotion Studio II Help topics.

If the software limits values should also be set on the fly, this can be done through the following TML parameters:

- **SWNEGLS** - the negative position limit value, in internal units
- **SWPOSLS** - the positive position limit value, in internal units

The software limits related parameter values can be set through the "32-bit Long or Fixed" wizard dialogue or via the communication line.



**Remark** The position internal units are described in the “Internal units and Scaling factors” help topic.