

Custom homing modes implementation

Application Note

Easy Motion Studio II

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

Your

Intelligent Move

Table of content

1. Application description	2
2. Application flow chart	2
3. EasyMotion Studio II implementation	3
3.1 Homing procedure implementation using functions	3
3.2 Main motion program	

1. Application description

This application note outlines the steps to implement a custom homing routine that positions the load at the center of the working area.



Figure 1 - Homing steps

The homing code will be stored in the drive memory and assumes the system is equipped with both positive and negative limit switch sensors.

2. Application flow chart



3. EasyMotion Studio II implementation

The application implementation in EasyMotion Studio II, includes two main parts:

- 1) The function ("Axis_Homing" in this case) that will include the homing procedure code;
- 2) The main program, in the "Motion" branch, which in this particular case will include only a function call instruction.

3.1 Homing procedure implementation using functions

The "Axis_Homing" function can be created using the wizard under the "Function" tree item.



Figure 2 – Functions tree item options

Once added, the function will appear under the "Function" tree item.



Figure 3 – Add the "Axis_Homing" function

Clicking on the function name will open its body, where the needed code can be added.

The homing functionality implementation will follow the flow chart in chapter 2. The first step is to set the limit switches as general purpose inputs. This can be done through the "Inputs/outputs" wizard.

EasyMotion Stud	lio II - P091.APN3	Homing to	the mi	ddle of the w	orking a	rea														-]
roject Applicatio	on Communica	ation Too	ls C	ontrol panels	Utilit	ties	Opti	ons	Help													_
	1 💷 🖻	0	1	<u> </u>				\$	*	Q	Worl	offline	Set all ap	ps offline	2	*	k		ž ž	X		Axis ∓ F
pplication tree	······	4																				
- 📙 P091.APN3 H	loming to the mid	ddle /	¬∖ Trap	ezoidal Profi	les		Se	arch														0/0
🖌 📃 APN3 - [2	2]		∧ s ci	Jrve			- 1	3 Dies	ble th	a dafa	ut functi		of the limit	owitch in	mute I S	N and L	D					*
Setup	•	1	∧ pt						TIVE -	1.7/1	imit Swit	h input	s used as c	eneral nu	imose in	muts						i fil
- M Motio	on	6	🔰 рут				_			1,77 5		minput	s uscu us g	ferrer or pe	npose n	puo						
— 順 на	omings)← Exte	ernal			Inpu	ts / ou	tputs												× au	Ţ.
4- 📭 Fu	unctions	Ş	h§ Elec	tronic Gearin	g					Read	input int	o variabl	e								-	×
- 2	Axis_Homing	1	🖄 Elec	tronic Camm	ing		0.0		0						- ···							-
— 🔃 In	terrupts		🗿 Mot	tor Command	ds		03	ingle i/		Set or	utput			Lov	V 🔘 Ha	gh						
— 🌆 C/	AM Tables		Posi	ition Triggers						Set I/	O line ty	be as inp	out									_
Memory	settings	1	🔂 Hon	ning			O R	ead in	put(s) i	nto var	riable											()
			A Con	touring												5 →			*			{()}
			[∕− Test																		- 1	Þ
			🗘 Ever	nts			O S	et outp	out(s)													
			Jum	ps and Funct	tion Calls	5																
			Ξ) Inpι	uts / outputs																		•0)
			<u>16</u> 16-1	bit Integer				High	/ Low	○ w	/ith value	of varia	ble		Ŧ							•0)
			32-1	bit Long or Fi	xed		1.1.1.1														-1	a
			🗠 Ariti	hmetic			Limit	t switci	1 input	s funct	ionality											- Fa
		9	🗟 Dati	a Transfer Be	tween Ax	xes	0 G	erault	purpo	se inn	ut											<u> </u>
			Sen	d Data To Ho	st				Po												_	
		P	ISC Mise	cellaneous			Enab	le inpu	ut funct	tionalit	y											
			🗧 Inte	rrupt Setting	s			etault		re inc:												
			×t Free	e Text				eneral	haiho	se inpl											_	
																	OK		Cancel	Help	•	
		,																				
												1	SetupID 84	97BL iP	OS4808	MY-CA	N S/N	ON50	10 F/V	V F514L	Axis	2 Onl

Figure 4 – Set the limit switches as general purpose inputs

Even the limit switches are set as general digital inputs, the associated interrupts can still be enabled. This feature will be used to capture the precise limit switches position.

The limit switches interrupts will be enabled and set through the "Interrupt Settings" wizard.

ject Application Communication Tools Control panels Utilities	s C	ptions	Help									
		1	*	Q	Work offline	Set all apps offlin	ne Z	2	k (D 🐔	<u>£</u> X	
lication tree												
P091.APN3 Homing to the middle		Search										
APN3 - [2]		0.0	cable th	a dafar	It functionality	of the limit quitch	innute I	SN and I	cD			
- S Setup /^ PT	1	0.0	sable ui		in functionality		inputs t					
M Motion					P //Reset Bits of	Interrupt Status R	non occ	urs, posit	ion com	iputed from i	irst encod	er sig
- 🖟 Homings 🕴 External		EI	NLSP1; /	//Positiv	E Limit Switch tri	iggers rising edge	- gister					
▲ 🕑 Functions 🛛 🖓 🖁 Electronic Gearing		E	NLSN1; / RB ICR 0	//Negat 0xBFFF	ive Limit Switch 0x00C0: //Set/Re	triggers rising edg	e at Contre	l Register				
E Axis_Homing		E T	a motor	e etorte	with a pogative	transmidal enoug	profile					,
- 🕼 Interrupts 🔅 Motor Commands	1	0	Inter	rrupt Se	ttings							
CAM Tables 🗱 Position Triggers			TM	1L Interr	upts							1
Memory settings		{} Se	*(C	Enab	le globally 🗌	Disable globally						
TA Contouring		{} n	e						Racet	previous TM		
T/~ Test		{} se	TM	L Interr	upt		Enable	Disable	interru	previous TW	Details	
IC4 Events		{} Pc	si Int0	0 - STO/	Enable input has	changed						
N Jumps and Function Calls	l H	{} <mark>(</mark>	n Int1	I - Short	-circuit							
□ Inputs / outputs		{} M	ov Int2	2 - Softv	are protections						۰	
16 16-bit Integer		{} Re	Int3	3 - Cont	rol error							
32 32-bit Long or Fixed	i 🖵		Int4	4 - Com	munication erro	r						
* Arithmetic			Int5	5 - Posit	on wraparound							
Conta Transfer Between Axes			Int6	5 - LSP p	rogrammed tra	nsition detected				✓		
Send Data To Host			Int7	7 - LSN	programmed tra	nsition detected	✓					
MISE Miscellaneous			Int8	8 - Capti	ire input transiti	on detected					*	
✓ Interrupt Settings	1		Int9	9 - Mote	on is completed	/ in velocity						
bit Free Text			Inti	iu - Timi 11 - Even	t set has occurre	ad ad					-	
a rice text			Inti	12 - Pori	tion trigger 1 /	change detected						
			Intl	13 - Digi	tal Input X prog	rammed transition						
				is bigi	a mpar x prog	annea aunsion			<u> </u>		_	
									OK	Cancel	Help	

Figure 5 – Enable and set the limit switches interrupts

The wizard is also allowing to set the LSN and LSP inputs transition (Low-High or High-Low) that will generate the associated interrupt.

The main idea behind this homing method is to move the load from one of the limits to the other, while measuring the distance. This allows computing the middle point coordinate.

The load can be moved using a position profile (with a position command higher than the max. possible distance between the 2 limit switches), without waiting for motion to be completed or using a speed profile. In this case, the axis is fist driven negatively, using a "Trapezoidal Profile – Speed".

ct Application Communication Tools	Control panels Utilities Options Help
	1 📟 🖷 👔 📓 🚖 😫 🗔 🗌 Work offline Set all apps offline 🥕 🖄 🕨 🔍 🖉 🕺 🗙
cation tree 🔹 🔹 🖛	
P091.APN3 Homing to the middle	rapezoidal Profiles Search
APN3 - [2]	Curve Solution Curve Disable the default functionality of the limit switch inputs LSN and LSP
- S Setup	T
- M Motion	VT
Homings 🖓 🗠 E	xternal //Speed profile
→ IF Functions	ectronic Gearing CACC = 0.31831; //acceleration rate = 1000.01[rad/s^2]
F Axis_Homing	Actionic Camming MODE SP;
🖉 Interrupts 🖉 M	Iotor Commands TUM1; //set Target Update Mode 1
CAM Tables	osition Inggers
Memory settings	oming Trapezoidal Profiles X
M C	Ontouring O Position
	est Speed
1×2 EV	Acceleration rate 1000.01 rad/s^2 v
	Jog speed -100 rpm *
16 1/	Schit Integer
32 32	2-bit I ong or O Generate new trajectory starting from actual values of load/motor position and speed
*/- A	dithmetic 🖉 Undets immediatels. 🗌 Weit until mation is completed
ះត p	Jata Transfer B O Update on event
E. Se	end Data To F
MISC M	Aiscellaneous
ź In	terrupt Settir
txt Fr	ree Text

Figure 6 – Trapezoidal Profile - Speed wizard

The "Events" wizard was used to set the drive to detect when the negative limit switch digital input goes high and stop the movement.

Project Application Communication	Tools Control panels Utilities	0	ptions Help			Events		×
			🔹 🛓 📮 🗌 Work offline Set all ap	ops offline 🎤 🚊 🕨	0	Function	n of input status	-
Application tree • # • P091 APN3 Homing to the middle • ARN3 - [2] S Setup • M Modion Phomings • Porticons • Ang, Homing • More the end	A trapezoidal Profiles A S Curve A PT By PVT Constant B Electronic Gearing L'S Electronic Comming Destronic Comming		Search C Disable the default functionality of the limit S setup TML Interrupts for LSN and LSR Whe The motor starts with a negative trapezoid S set event when the digital input LSN is 110 (Note: Stop: When the digital input LSN is 110 (Stop: V) Searche motion when even occur	t switch inputs LSN and LSP in transition occurs, position corr al spend profile H and wait for event in order to SN is high	nputed f stop the	When	Encoder index Znd encoder index positive limit switch is triggered negative limit switch is triggered Digital input IN3/LSN Home input	High Low OK Help
Memory setting:	Position Triggers Action Triggers Action Triggers Action Triggers Action Action		WAIT; // Wait until the event occurs works None None When actual motion is completed Function of motor or load speed After a wait time Function of reference Function of reference	Cancel Help	op the	motion		
	To Data Transfer Retween Asses Send Data To Host MK Miscellancous Minerupt Settings Mr Free Text		Function of a 32-bit variable value			Events Function Stop th Wait un Exit fro	of input status he motion when the event occurs ntil the event occurs m the wait loop after a time equal wit	* //

Figure 7 – Events wizard

<u>**Remark**</u> More details on the wizard dialogues options are available in the EasyMotion Studio II help topics that can be open by clicking on the "Help" button available in all the wizard dialogues.

After reaching the negative limit, the actual position is set to 0 IU, using the "Motor Commands" wizard. This will make the middle point coordinate computing easier.

roject	Applic	ation	Com	munic	ation	Tools	Cont	trol p	anels	Util	ities	Opti	ions	Help														
	C 1	2				0		-		1	1		\$	2	Q	Wo	k offline	Set all	apps of	ffline	\geq	*	Þ	0	Ź	<u>7</u>	×	
plicatio	on tree			•	4																							
📙 P0	91.APN	3 Homi	ng to t	he mi	ddle	\frown	Trapez	oidal	Profile	s		Se	arch															
- 🗌	APN3	- [2]				\wedge	S Curve	e				► ₹) Dis	ble th	e defai	ult funct	onality	of the lir	nit swit	tch inn	uts I S	N and	ISP					
	S Se	tup				M	PT) Set	in TMI	Interr	unts for	I SN and		hen trai	nsition			ition o	omput	ted from	n first	encod	ler sir
-	MM	otion				No.	PVT) The	-p - tot	ctorte	with a r	ogativa	******	idal eno	and are	filo	., pos		omput			ciicou	er się
	- 🃭	Homi	ngs			¢-	Externa	al				1	1 6-1	motor	starts	withat	eyauve	u apezoi	uai spe	seu pro	///ie							
	- 1	Functi	ons			M.S.	Electro	nic G	earing			12) Set	event v	when t	ne digit	n input	LOIN IS H		u wan	ior ev	ent in i	order	to stop	p uie m	ouon		
		🕑 Ax	is_Hor	ning		12	Electro	nic C	ammin	g		T 4	Set	positio	on of th	e moto		er to sta	art mea +1	isuring	the ir	terval						_
	_ 12	Intern	upts			3	Motor	Com	mands				SAI	01,77	Secaci	uar posi	uon valu	e to Ulro										
	- 17	CAM	Tables			F	Positio	n Trig	gers				Motor	Comma	ands	-l		1114	de la c		1			×	-		-1	
	Memo	ory setti	ngs				Homin	g) Acti	vate th	e conti	rol loops	and PW	M outpu	its (AXI	SON)					the m	otion		
						M.	Contou	uring) Dea	ctivate	the co	ntrol loc	ps and I	WM out	tputs (A	XISOF	F)				_			
						1/	Test					\mathbf{H}	🔾 Stop	o motic	on with	accelera	tion / c	ecelerati	on set									
						lo.	Events					H a		late im	mediat	ely												
						ru.	Jumps	and I	unctio	on Call	ls	H) Upc	late on	event													
						=	Inputs	/ out	puts			HD				۲	Value											
						10	16-bit	Integ	er			0	Set	actual	positic	on to 🔾	Variable		0	rot		Ŧ						
						<u>≝</u> .	32-bit	Long	or Fixe	ed						🔍 Value												
						- · ·	Arithm	etic					∪ Set	current	t limit	🔍 Varia	ble											
						֩	Data Ir	ranste	er betw	/een A	vxes	-					. (Value										
						MISC	sena U Miscoll	ata I	o nost			(∪ Set	quick s	top de	celeratio	n rate	Variab	le									
						4	Intornu	aneo	us ttinar										0	ОК	Ca	ncel	He	lp				
						- K - K	Eron To	pr se	rungs				_	_	_										1			

Figure 8 - Motor Commands wizard

The next step is to move positively until the positive limit switch is reached. This will be done using the "Trapezoidal Profile – Speed" and the "Events" wizard dialogues.

EasyMotion Studio II - P091.APN3 Homin Project Application Communication	ig to the middle of the working area — [Tools Control panels Utilities Options Help	×
	💿 🔤 🖳 🖬 📓 🔹 🛓 📮 🗋 Work offline Set all apps offline 🥕 🖄 🕨 🔮 🗲 🧲 🗡	Axis off F3
Project Application Communication Application tree Application Application	Tools Control panels Utilities Options Help Image: Control panels Image: Control panels	Axis off 7 0/0 2 1 1 1 1 2 7 7 7 7 7 7 7 7 7 7 7 7 7
	<pre> finterrupt Settings bit Free Text </pre>	

Figure 9 – Positive limit switch detection

<u>Remark</u>: By default, when a limit switch became active, if the associated interrupt is enabled, the drives automatically saves the encoder position indication in the "CAPPOS" variable, if the encoder is connected to Feedback #1 connector or in the "CAPPOS2" variable, if the encoder is connected to Feedback #2 connector.

In this application case, the "CAPPOS" variable will contain the exact distance (in encoder counts) between the 2 limit switches, because the position was set to 0 IU, when the negative limit was detected.

The middle point can be computed by dividing the value in "CAPPOS" to 2 or using a right shift operation with 1-bit, that is equivalent with a division to 2^1.

	0 🔳 🖳	- 1		1 🗶	Q Work offline Set all apps offline	2	*	0	Z	<u>۶</u>	x	Axi
plication tree	F											
P091.APN3 Homing to the midd	le Trapezoidal F	rofiles	Se	arch								0/0
APN3 - [2]	∧ S Curve		► {	Disable the	e default functionality of the limit switch inc	outs LSN	and LS	p				1
- S Setup	∧^ pt			Setup TMI	Interrupts for LSN and LSP. When transition	n occurs	nositir	on comput	ed from	first en	coder si	. @
 M Motion 	NO PVT			The motor	starts with a negative transcoidal speed pr	ofile	, positi	on compa			icouci siç	
- 🕞 Homings	0 External			Set mont	when the digital input I SN is HIGH and wait	for our	nt in on	der to stor	the me	tion		1
 Functions 	兴용 Electronic Ge	aring	1			tor ever		der to stop	, the life			×
Axis_Homing	Electronic Ca	mming	17		n or the motor of in order to start measuring	g the into	ervai					1
- Interrupts	Motor Comm	ands) The load is	moved until the other limit switch is reach	ed						
CAM Tables	Position Trigg	gers	1÷	Set event	when the digital input LSP is HIGH and wait	for even	nt in ord	ler to stop	the mo	tion		- Ai
 Memory settings 	The second		*	Position is	stored in variable CPOS							=
	W Contouring			CPOS = C	APPOS;							
	17 Test		+ {	Compute I	niddle point of interval							
	Events	unction Calls		CPOS >>=	: 1;							
	- I locute / oute	uts	+ {	Move to the second seco	ne middle of the distance between the 2 lim	its						<u> </u>
	16 16-bit Intege	r	+ {	Reset posi	tion value							1
	32 32-bit long	y Fived		Enable the	default functionality of the limit switch inp	uts LSN a	and LSI	Р				~2
	* Arithmetic											đ
	유 Data Transfer	Retween Aves										lip lip
	Send Data To	Host										
	MISC Misselles and	- HUSL										
	(Lise in Col											
	2 Interrupt set	ings										
	ox Free lext											
	•											
	>				SetupID 84978L iPO	S4808 M	IY-CAN	S/N ON	5010	F/W F51	4L Axis	s 2 On
	•				SetupID 8497BL iPO	S4808 M	IY-CAN	S/N ON	5010	F/W F51	4L Axis	s 2 On
bit Integer					SetupiD 84978L iPO	S4808 M	IY-CAN	S/N ON	5010	F/W F51	4L Axis	s 2 Or
bit Integer Set 16-bit variable CPOS					SetupiD 84978L iPO	S4808 M	IY-CAN	S/N ON	5010	F/W F51	4L Axis	s 2 Or
bit Integer Set 16-bit variable CPOS	•				∣ SetupiD 84978L iPO	S4808 M	IY-CAN	S/N ON	5010	F/W F51	4L Axis	5 2 Or
bit Integer Set 16-bit variable CPOS	*	21	Ŧ		∣ SetupiD 84978L iPO	S4808 M	IY-CAN	S/N ON	5010	F/W F51	4L Axis	s 2 Or
oit Integer Set 16-bit variable CPOS With value / 16-bit varia	* Ible / label CAPPC	22	¥		SetupID &4978L iPO	S4808 M	IY-CAN	S/N ON	5010	F/W F51	4L Axis	s 2 Or
oit Integer Set 16-bit variable CPOS With value / 16-bit varia	* ible / label CAPPC	20	v		SetupID 84978L iPO	S4808 M	IY-CAN	S/N ON	5010	F/W F51	4L Axis	s 2 Or
bit Integer Set 16-bit variable CPOS ● With value / 16-bit varia O Data O With O Progra Arithm	* able / label CAPPC	20	•		SetupiD 84978L iPO	S4808 M	IY-CAN	S/N ON	5010	F/W F51	4L Axis	s 2 Or
bit Integer Set 16-bit variable CPOS Image: With value / 16-bit varia Image: Data Image: With Progra Image: Progra Arithr Image: E2RON Arithr	* able / label CAPPC	25	¥.		SetupiD 84978L iPO	S4808 M	IY-CAN	S/N ON	5010	F/W F51	4L Axis	s 2 Or
bit Integer Set 16-bit variable CPOS Image: With value / 16-bit varia Image: Data Image: Data <tr< td=""><td>* able / label CAPPC netic d to variable</td><td>20</td><td>•</td><td>^r] The va</td><td> SetupiD 84978L iPO</td><td>S4808 M</td><td>IY-CAN</td><td>S/N ON</td><td>5010 </td><td>F/W F51</td><td>4L Axis</td><td>s 2 On</td></tr<>	* able / label CAPPC netic d to variable	20	•	^r] The va	SetupiD 84978L iPO	S4808 M	IY-CAN	S/N ON	5010	F/W F51	4L Axis	s 2 On
bit Integer Set 16-bit variable CPOS With value / 16-bit variation Data With Progra E2RON With Low With Low	* able / label CAPPC	25	▼	7] The va	SetupiD &4978L iPO	S4808 M	Y-CAN	S/N ON	5010	F/W F51	4L Axis	s 2 Or
bit Integer Set 16-bit variable CPOS With value / 16-bit variant Data With Progra Arithr E2RON With Low High Su	+ + + + + + + + + + + + + + + + + + +	ns e	v	The v	SetupID 84978L iPO	S4808 M	IY-CAN	S/N ON	5010	F/W F51	4L Axis	s 2 Or
bit Integer Set 16-bit variable CPOS With value / 16-bit varia Data Data Progra Arithr E2RON With E2RON With High Su		S e	•	·] The v	SetupID 84978L iPO	S4808 M		S/N ON	5010	F/W F51	4L Axis	5 2 On
bit Integer Set 16-bit variable CPOS With value / 16-bit varia With Progra Arithr E2RON With Low High Su With inverse (-) See	whee / label CAPPC	PS e th the prod	*	The v	SetupID 84978L iPO	▼		S/N ON	5010	F/W F51	4L Axis	<u>s 2 On</u>
bit Integer Set 16-bit variable CPOS Image: Set 16-bit variable Data Image: Set 16-bit variable CPOS Image: Set 16-bit variable Data Image: Set 16-bit variable CPOS Image: Set 16-bit variable		is e	* luct of) The variable	SetupiD 84978L iPC	▼		S/N ON	5010	F/W F51	4L Axis	s 2 Or
bit Integer Set 16-bit variable CPOS With value / 16-bit varia Data Data Progra Arithr E2RON Au With Low With High With inverse (-) Using masks A		e	* luct of	The variable	SetupID &4978L iPO	S4808 M	IY-CAN	S/N ON	5010 1	F/W F51	4L Axis)
bit Integer Set 16-bit variable CPOS With value / 16-bit varia Data Data Progra Arithr E2RON Ad With Low With High Su With inverse (-) Using masks A And V		e	* luct of	The v	SetupID 84978L iPO	54808 M	IY-CAN	S/N ON	5010 1	F/W F51	4L Axis)
bit Integer Set 16-bit variable CPOS Image: Set 16-bit variable		e e ble	T I I I I I I I I I I I I I I I I I I I	The v	SetupID 84978L iPO	54808 M		S/N ON	5010	F/W F51	4L Axis	s 2 Or
bit Integer Set 16-bit variable CPOS With value / 16-bit varia With Progra Arithr E2RON Ac With Low With inverse (-) Using masks A With checksum Dir		e	T Uuct of	The variable	SetupID 84978L iPO alue / variable The value / variable U Comparison Comparison D Comparison D Comparison	×		S/N ON	5010	F/W F51	4L Axis	s 2 On
bit Integer Set 16-bit variable CPOS With value / 16-bit varia Data Data Progra Arithr E2RON With Cheve Using masks A With checksum Data Progra Arithr Su Su Data Add Su Data Data Add Data Data Data Add Data	ble / label CAPPC captor captor d to variable btract from variable t PROD register wi ralue / 16-bit variable	e	v luct of	The variable	SetupID &4978L iPO alue / variable The value / variable C Left shifted (<<) Wit By the value of 16-bit variable	s4808 M			5010	F/W F51	4L Ακίσ	s 2 On
bit Integer Set 16-bit variable CPOS With value / 16-bit varia Data Data Progra Arithm E2RON With E2RON With Low With inverse (-) Using masks A Mith checksum Bit With checksum		e th the prod ble	v luct of	The variable	SetupID &4978L iPO	54808 M		S/N ON	5010 I	F/W F51	4L Axis)
bit Integer Set 16-bit variable CPOS With value / 16-bit varia With Progra Arithm E2RON Ad With High With inverse (-) Using masks A Using masks A Mith checksum Data n		e e ble ft << V yht >> V	* luct of	The variable	SetupID 84978L iPO	• • • • • • • • • • • • • • • • • • •		s/N ON	5010 1	F/W F51	4L Axis)
bit Integer Set 16-bit variable CPOS With value / 16-bit variable Vith Progra Arithr E2RON Ad With High With inverse (-) Using masks A With checksum Data Data Set Program		e	* luct of	The variable	SetupID 84978L iPO	54808 M		S/N ON	5010	F/W F51	4L Axis)
bit Integer Set 16-bit variable CPOS With value / 16-bit varia With Progra Arithr E2RON Ac With Low With inverse (-) Using masks A With checksum Data Data Program E2ROM Set Program E2ROM		e the prod ble t << v jht >> v Left <	* luct of /ariable	The variable	SetupID &4978L iPO	54808 M	IY-CAN	S/N ON	5010 	F/W F51	4L Axis)
bit Integer Set 16-bit variable CPOS With value / 16-bit varia Data Data Progra Arithr E2RON With E2RON With inverse (-) Using masks A Mith checksum Data E2RON Set Data Progra V Set Data CPOS Add Su Set Data CPOS Add Su Su Su Su Su Su Su Su Su Su	ble / label CAPPC ble / label CAPPC blic d to variable btract from variable btract from variable ift variable C Let Rig ift variable Rig ift variable	e th the prod ble t << V ht >> Left < Right	* luct of /ariable	The variable	SetupID &4978L iPO	54808 M		S/N ON	5010 1		4L Axis)

Figure 10 – Middle point coordinate computing

The implementation was done using the wizards for the "16-bit Integer" and "Arithmetic" operations.

The right shift operation was preferred here because it consumes fewer DSP resources than the division and there's no need to declare an extra 16-bit variable to be used only for this division.

After the shift operation is done, the "CPOS" variable will contain the absolute position of the middle point, so the load can be moved there, using an absolute position profile.

The absolute position profile can be inserted using the "Trapezoidal Profile" wizard dialogue.

EasyMotion Studio II - P091.APN3 Homi Project Application Communication	ng to the middle of the working area Tools Control panels Utilities	- C	X
	o 🔤 🖳 🚥 👔 👔	Trapezoidal Profiles	× ^{off}
Application tree • 4 • 0991APN3 Homing to the middle • APN3 - [2] • S Setup • M Motion • Homings • D Functions • Asis, Homing • Interrupts • CAM Tables Memory settings	A Scurve Trapezoidal Profiles Scurve A Scurve A PT Py Py Py External SElectronic Gearing SElectronic Gearing Motor Commands Position Triggers A Contouring Trest Ice Sents Injury and Function Calls Injury and Function Calls Sents Injury and Function Calls Sents Sents	Tapezoidal Profiles Position Absolute Speed Relative Acceleration rate 1000.01 Position to reach Image: Comparison of the second sec	Help
٠	α Free Text		

Figure 11 – Absolute trapezoidal position profile

<u>Remark</u> The "position to reach" field was disabled because the commanded position (the "CPOS" value) was computed in the previous step.

After the reaching the middle point (the home position), the actual position will be set to 0, using the "Motor commands" wizard.



Figure 12 – Set the middle point as home position (0 IU)

At this point, the load is placed in the middle of the working area, and the system can be operated using absolute position commands.

Before returning from the function, the limit switches default functionality will be restored using the option in the "Inputs/outputs" wizard dialogue.



Figure 13 – Limit switches default functionality restoring

The main purpose of the "IN2/LSP" and "IN3/LSN" digital inputs is to allow connecting some limit switch sensors that will prevent the load from moving outside the safe are.

By default, when one of the limit switches became active, the drive stops the motor (using a quick stop profile), sets the correspondent bits in the MER error register and executes the code inside the correspondent limit switch interrupt routine (if it's active on the respective transition).

A detailed description of the limit switches functionality can be found in the "<u>Drive special inputs - Limit</u> <u>switches</u>" application note.

3.2 Main motion program

In this application note case, the "Motion" section will contain only an instruction that will trigger the "Axis_Homing" function execution. This instruction can be generated using the "Call" option in the "Jumps and Function Calls" wizard dialogue.



Figure 14 - How to call a TML function

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

For more details about the functions call and usage, see the "Functions calling from a master" application note.

The "Axis_Homing" function can be called any time the homing procedure execution is required.