

How to call a TML function from a master

Application Note

Easy Motion Studio II

Your

Intelligent **Move**

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

Table of content

1. Application description	
2. TML function	3
3. Calling a TML function from the master	4
3.1 Calling a TML function from the RS232 master	4
3.2 Calling a TML function from the CANOpen / EtherCAT master	6

1. Application description

This application note describes how to create a TML function with EasyMotion Studio II and call it from an RS232, a CANopen or an EtherCAT master.

2. TML function

The TML functions can be created and stored to the drive, using the EasyMotion Studio II software.

First, click on the "Functions" section to access the "Functions" list and, then, type the name of the function that needs to be created.



Figure 1 – Creating a TML function

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



Figure 2 – Add the "Start_Motion" function

To write the TML code into the function, select the function's name on and insert the TML instructions to the function body.

In this example, the "Start_Motion" function contains a trapezoidal position profile that turns the motor 150 rot, in the positive direction.

		1 1 0 0 w	ork offline Set all apps offline	2 2 1	0	1	E X	15 15		
v #										-
APN - Call a TML function from the master	Trapezoidal Profiles	Search								18
APN - Call a TML function from the mat	∧ S Curve		los							n'
Setup	r∕∧ pt	// Trajecolular Profile //Position profile CACC = 0.31831; //acceleration rate = 1000.01[rad/s^2]								
 M Motion 	N PVT									
- 🚺 Homings	0 − External	CPOS = 300000L; //p	CSPD = 66.66667; //slew speed = 2000[rpm] CPOS = 300000L; //position command = 150[rot]							
- 🕑 Functions	유율 Electronic Gearing	CPR: //position com	nand is relative							
Start_Motion	Electronic Camming	TUM1; //set Target U	pdate Mode 1			٨				
🔡 Interrupts	Motor Commands	UPD; // execute imm	ediate			T T				
CAM Tables	Position Triggers	invic, where, // waters	a competion			-			-	Ļ
Memory settings	🔂 Homing	Trapezoidal Pro	files						×	
	M Contouring	Position	O Absolute							
	T/⊂ Test	○ Speed	Relative Additive							
	IC* Events		Acceleratio	n rate 100	0.01	rad/s^2	* 🗸			
	N Jumps and Function Calls		Slew :	speed 20	00	rpm				
	☐] Inputs / outputs	Position increment 150 rot +								
	16 16-bit Integer	Constate a	us trainstance starting from actuals	values of posit	ion and	on and cafe				
	32 32-bit Long or Fixed	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								
	* Arithmetic									
	Pata Transfer Between Axes	Update imn	nediately [] Wait until motion is c	completed						
	Send Data To Host	Update on event Setup motion data, but don't start execution								
	MISC Miscellaneous									
	🗧 Interrupt Settings									
	txt Free Text	OK Cancel Help				Help				

Figure 3 – The "Start_Motion" function body

The TML function can be downloaded with the *Application* | *Motion* | *Download Program* menu command. The Download Program becomes active once the motion is built with *Application* | *Motion* | *Build*. Alternatively, the TML functions will be downloaded when the TML application is launched with *Application* | *Run* menu command.

Api	New Rename Duplicate Dente Setup	Ň		New Rename Duplicate Delete Setup	ommunication			New Rename Duplicate Denne Setup	•	
	Motion	*		Motion		Build		Motion	1	Build
È.	Run	F5		Run	F5	Download CAM Tables		Run	F5	Download CAM Replay
0	Stop Program		0	Stop Program	m	Download Program	0	Stop Program		Download Program
4	Axis on	F2	4	Axis on	F2	Load from Another Application	4	Axis on	F2	Load from Another Application
5	Axis off	F3	4	Axis off	F3		1	Axis off	F3	
x	Reset drive	F4	×	Reset drive	F4		×	Reset drive	F4	
	Export	÷.		Export	*			Export	*	

Figure 4 – How to download the function to the drive

3. Calling a TML function from the master

3.1 Calling a TML function from the RS232 master

To reduce the traffic on the communication line, the repetitive tasks (e.g. homing procedures, predefined movements, emergency procedures etc.) can be implemented directly at the drive level, as TML functions. The master will trigger the execution of the respective functions instead of sending all the TML commands.

A function can be called by its address or for the first 10 functions in the list by using the "Function x" TML instruction, where "x" represents the function number, first function being 1 (the order in which they are listed in EasyMotion Studio II).

The function address can be retrieve in the "Command Interpreter" window from EasyMotion Studio II, by typing the "*?function_name*" TML instruction.

Once the address of the function is known, it can be called using the "CALL address_in_hex" TML instruction.

Command interpreter	×	Command interpreter	×
TML> ?Start_Motion Start_Motion is label with address 0x4020 TML> CALL 0x4020		TML> function 1	*
🗶 🖺 🔜 TML>	•	🗶 📋 📙 TML>	

Figure 5 – Retrieve the address of a function and call it using the address and its number

<u>Remark:</u> If the program is modified, the function address may change, so it is recommended to check the function addresses after the TML application is completed.

The "Function x" instruction is independent from program changes as EasyMotion Studio II updates the addresses of the functions when it builds the program.

The binary code of the "CALL" instruction can be generated using the "Binary Code Viewer" tool from EasyMotion Studio II (*Tools | Binary Code Viewer*).

The options for "Protocol", "Sender", and "Destination" from the bottom of the "Binary Code Viewer" window should be selected. In the case above, the information will be sent through the "RS232" protocol from "Host" with ID = 255 to the destination "Axis" with ID = 1 (Axis 1).

and Co	Construction of the second						
ition tree 🛛 🗰 Me	emory C	trl+M					
APN - Call a TML function from the r 🚎 Ou	itput	Search			0,		
APN - Call a TML function from t Log	gger C	trl+L	Profiles				
Setup	ope C	trl+E //Position profil	e e e e e e e e e e e e e e e e e e e				
M Motion	PVT	CACC = 0.31831	1; //acceleration rate = 1000.01[rad/s^2]				
Hominas	0 − External	CSPD = 66.6666 CPOS = 300000	5/; //slew speed = 2000[rpm] II: //nosition.command = 150[rot]				
Eurotions	S Electronic Gearing	CPR; //position	command is relative				
	19) Electronic Cammin	My Executine Germing MODE PP:					
C Start_Motion	D Mater Commands	UPD: // execute	get Update Mode 1 immediate				
Interrupts	S Motor Commands	IMC; WAITI: // v	valt for completion				
CAM Tables	Binary code viewer	and the second sec		×	-		
 Memory settings 							
	Source co	de	Binary code sent	Binary code received			
	CALL 0x4020	de	Binary code sent	Binary code received			
	CALL 0x4020	de	Binary code sent.	Binary code received			
	CALL 0x4020	de	Binary code sent	Binary code received			
	CALL 0x4020	de	Binary code sent	Binary code received			
	CALL 0x4020	de	Binaty code sent	Binary code received			
	CALL 0x4020	de	Binary code sent	Binary code received			
	CALL 0x4020	de	Binatry code sent	Binary code received			
	CALL 0x4020	de	Binaty code sent	Binary code received			
	CALL 0x4020	de	Binatry code sent	Binary code received			
	CALL 0x4020	de	Binairy code sent	Binary code received			
	CALL - 0x4020	code	Binaty code sent	Binary code received			
	CALL 8x4028	code	Binatry code sent	Binary code received			
	CALL 0x4020	code Destination	Binaty code sent	Binary code received			
	CALL 8x4028	code Sender Destination Relay @Ais	Binaty code sent	Binary code received			
	CALL 0x4028 CALL 0x4028 Show binary Protocol @ RS 232 @ RS 485 @ RS 485	code Sender © Relay © Host © Group	Binatry code sent Simulate answer Type a simulated value (hexa) returned by the drive in case of query Age Code to an offer the many to identify the answer in case of query Age Code to an offer the many to identify the answer in case of query	Bearly code received			
	CALL 8x4028 CALL 8x4028 Show binary Protocol © R5 232 R5 425 TMLCAN	code Sender Destination © Relay © Host © Group	Binatry code sent Simulate answer Type a simulated value (hexa) returned by the drive in case of query AB messages 7, 17 Select an usis from the group to simulate the answer in case of query 23	Binary code received			
	CALL 0x4020 Show binary Protocol 0 R5 232 R5 485 TMLCAN 0 TechnoCAN	code Sender Destination © Relay © Host © Broadcast	Binatry code sent Simulate answer Type a simulated value (heca) returned by the drive in case of guery AB messages 7, 7? Select an axis from the group to simulate the answer in case of guery [23 messages: 7?	Binary code received			

Figure 6 – Conversion of TML code to RS232 message using "Binary Code Viewer" tool

In the "Source code" box of the "Binary Code Viewer" tool enter the memory address displayed by the "Command interpreter" in the previous step, as shown below. After clicking on "Show binary code," the "Binary code sent" and "Binary code received" will be displayed.

Binary code viewer			×
Source CALL 0x4020	code	06 00 10 74 01 4	Binary code sent 3 20 EB Binary code received 4F
Show bina Protocol	Sender	Destination	Simulate answer
 RS 232 RS 485 TMLCAN 	 Relay Host 	 Axis Group Broadcast 	Type a simulated value (hexa) returned by the drive in case of query [ABBA5A5B] messages: ?, ?? Select an axis from the group to simulate the answer in case of query 23
TechnoCANTML	ID 255	ID 1	messages: ?? Help

Figure 7 - "Binary Code Viewer" tool with detailed information

A detailed description of the Technosoft serial protocol can be found in the EasyMotion Studio II help topics, *Help | Help Topics | Communication | Protocols* menu command.

3.2 Calling a TML function from the CANOpen / EtherCAT master

In CANopen / EtherCAT, a previously downloaded TML function can be triggered with 2006h object.

When a write is performed to the object 2006h, the TML function with the index specified in the provided value is called.

The first 10 TML functions defined in the EasyMotion Studio project can be called through this mechanism.

Remarks:

- the functions can be triggered only when the drive is switched to Operation Enable state.

- any attempt to call another function is signaled by the drive with an SDO abort code (0609 0030h - Value range of parameter exceeded).

- if a valid value is entered but no TML function is defined in that index, an SDO abort code will be issued (0800 0020h - Data cannot be transferred or stored to the application).

- it is not possible to call a TML function, while another one is running. If a TML function is triggered while another is running then the drive will set bits 7 (warning) in the StatusWord (6041h) and bit 14 (command error) in Motion Error Register (2000h) and the function call will be ignored.