

FAQ: Using a SMDXXE2 Absolute Encoder

AMCI SMDXXE2 integrated motors and drivers can be ordered with an optional absolute encoder. This encoder has a fixed 2048 counts per turn and can measure 2²¹ (2,097,152) turns before rolling over to zero.

This FAQ shows where and how to use the absolute encoder position data.

Power UP

AMCI SMDXXE2 motion products will always power up with the Position Invalid status bit set and with the Current Motor Position equal to zero.

The following three rungs monitor the network communication status of the AMCI SMDXXE2 device and use the Preset Motor to Encoder command to set the motor position to the absolute encoder position when either the position invalid status bit is set or if the current motor position is not equal to the current encoder position.

A GSV (Get System Value) instruction can be used to determine if a device on the Ethernet IP network is communicating with the PLC. The device is communicating if the value in the DEST address is equal to something other than zero.	ess is equal to zero. The device is not
Not whe com	communicating equal to zero en not SSV- Get System Value Class Name Module Instance Name AMCL_SM034E2 Attribute Name FautCode Dest smd_communication_status 0 ←
0=communicating Not equal to zero when not communicating 	set when motor position has been set to encoder position at power up power_up_preset (V)
Source A smd_communication_status Source B 0 0 set when motor 0=communicating	
position has been Not equal to zero set to encoder when not position at power up communicating position invalid bit power_up_preset EQU buffered_SMDXKE2_data[0] 10	Motor_to_Encoder
Equal Source A smd_communication_status 0 + NEQ AMCL_SMD34E2_output_data AMCL_SMD34E2_output_data	LPreset_motor_to_encoder AMCL_SMD34E2:0
0 ← Source B AMCLSMD34E2:LENCODER_POSITION 0 ←	set when motor position has been set to encoder position at power up power_up_preset



Manually setting the Motor Position to the Absolute Encoder Position

It is also possible to manually send the command to preset the motor position to the absolute encoder position.

In this example, the **preset_motor_to_encoder** bit is an internal bit that must be set by your ladder logic program to make this operation occur.

set when stopped	
preset_motor_to_encoder buffered_SMDXXE2_data[0].3	-AMCI_SMD34E2_Preset_Motor_to_Encoder
	AMCL_SMD34E2_Preset_Moto manual_preset_motor_to_encoder AMCL_SMD34E2_output_data AMCL_SMD34E2:0

Presetting the Absolute Encoder Position

The absolute encoder position will probably not match the machine position. Instead of turning the motor's shaft until it does, the SMDXXE2 allows you to set the absolute encoder's position to a desired value within the range of +/- 8,388,607.

The Preset Encoder command has a Save In Flash option. The results of this command will be lost the next time the SMDXXE2's power is cycled if there is a "0" in the Save_In_Flash field when issuing this command.

Because the SMDXXE2's flash memory has a life of 10,000 write cycles, the Preset Encoder and Save in Flash command must only be used during setup and calibration operations.

In this example, the **preset_absolute_encoder** bit is an internal bit that must be set by your ladder logic program to make this operation occur.

set when stopped preset_absolute_encoder buffered_SMDXXE2_data[0].3	AMCL_SMD34E2_Preset_Encoder_Position	
	AMCI_SMD34E2_Preset_Enco Preset_encoder save_in_flash_0_no_1_yes 1	
	Target_Position 0	
	AMCI_SMD34E2_output_data AMCI_SMD34E2:0	



Recovering from a Stall Detected Condition

Any SMDXXE2 unit with either an Absolute or Incremental encoder can be configured to detect when the motor's shaft has stalled. Specifically, the SMDXXE2 will stop the move if the motor position and the encoder position are out of alignment by more than 45 degrees.

In this example, the **clear_stall_detect_error** bit is an internal bit that must be set by your ladder logic program to make this operation occur.

set when stall set when stopped condition detected set when stopped buffered_SMDXXE2_data[1].14 buffered_SMDXXE	52_data[0].3	stal_condition_detected
Issuing the Reset Errors command will clear the stall de	tect error. set when stall	
clear_stall_detect_error stall_condition_detected	condition detected buffered_SMDXXE2_data[1].14	AMCI_SMD34E2_Reset_Errors-
<u>∃</u> €∃€∃€		AMCI_SMD34E2_Reset_Error Reset_Stall_Detect_Error AMCI_SMD34E2_output_data AMCI_SMD34E2:0
Issuing the Preset Motor to Encoder command will set th clear_stall_detect_error stall_condition_detected	set when stall condition detected pos	encoder position and make the position valid. tion invalid bit ffered_SMDXXE2_data[0].10
		CI_SMD34E2_Preset_Moto Preset_motor_to_encoder_after_stall Game CI_SMD34E2_output_data AMCI_SMD34E2:0
Unlatch the internal stall_condition_detected bit after bo	set when stall	on invalid status bits have been reset. ition invalid bit
clear_stall_detect_error stall_condition_detected	buffered_SMDXXE2_data[1].14 but	

Resetting the Acknowledge Bit

The SMDXXE2 units set an Acknowledge Bit as a response to a Reset Error or any of the Preset commands. The following rung resets the command word to zero to allow the next Reset Error or Preset command to be immediately accepted.

Resetting the command word to zero will cause the SMDXXE2 to reset the Acknowledge bit.

Acknowledge bit buffered_SMDXXE2_data[1].13	Move				
10	Source 0				
	Dest AMCI_SMD34E2:0.COMMAND_WORD_0 2#0000_0000_0000_0000 <				
		_			



Out and Back Absolute Moves

The following two rungs perform out and back Absolute Moves.

The Position Invalid Status bit must be reset before any Absolute Moves can be performed.

In this example, the **absolute_move_out** and **absolute_move_back** bits are internal bits that must be set by your ladder logic program to make either of these absolute moves occur.

absolute_move_out absolute_move_back	position invalid bit buffered_SMDXXE2_data[0].10	AMCI_SMD34E2_At	osolute_Move	1
)/	AMCI_SMD34E2_Absolute_Mo Axis_Status_Bits buf	absolute_move_out_AOI fered_SMDXXE2_data[0] 0 ←	-(EN)
		Target_Position	10000	
		Programmed_Speed	2000	
		Acceleration	20	
		Deceleration	10	
		AMCI_SMD34E2_output_data	AMCI_SMD34E2:0	
absolute_move_out absolute_move_back	position invalid bit buffered_SMDXXE2_data[0].10	AMCI_SMD34E2_Ab	solute_Move]
	J' L	AMCI_SMD34E2_Absolute_Mo al Axis_Status_Bits buf	fered_SMDXXE2_data[0]	-(EN)
		Target_Position	0 ← 0	-(ER)
		Programmed_Speed	1500	
		Acceleration	20	
		Deceleration	10	
		AMCI_SMD34E2_output_data	AMCI_SMD34E2:0	

SMD34E2 Resetting Driver Faults

The AMCI SMD34E2 is different from AMCI's other SMDXXE2 products is that it will set a Driver Fault bit when the Main Power is removed while the Auxiliary Power is maintained.

The following logic shows how to recover from a SMD34E2 Driver Fault.

NOTE: For this logic to work correctly, the PLC must be aware of whether or not the main power is applied to the SMD34E2. (The SMD34E2 <u>does not</u> report this information to the network.) In the following example, the state of the main power is represented by the **main_power** bit, "0" if the main power is removed or "1" if the main power is applied.

			ero to Output)	Nord 0, which is also calle	d Command Word 0. This will prepare
the SMD34E2 to accept the n 0=power removed AMC	lext command after the main ISMD34E set when	n power has been restored.			
	r fault				
	fered_SMD34E2_data[1].7		MC	W	main_power_removal_detected
/		Move			())()()()()())()()())()()())()())())_()
	50	Source		0	
		Dest Al		O.COMMAND_WORD_0 0000 0000 0000 0000 +	
			21	0000_0000_0000_0000 (
After the Main Power has be	en restored, and if there is	a Driver Fault, send the Reset Erro	or command to	the SMD34E2.	
This command will also enab	le the driver causing it to se	end power to the motor.			
	0=power removed	AMCI SMD34E set when			
	1=power applied	driver fault			
main_power_removal_det	ected main_power	buffered_SMD34E2_data[1].7		AMCI_SMD34E	2_Reset_Errors
] L	AMO	1 SMD34E2 Depat Error	. main power reset errors
				I SMD34E2_Reset_Error	AMCI SMD34E2:0
-		id Status bit to become set. This a rent absolute encoder position.	status bit must	be reset before you can p	perform Absolute Moves.
	0=power removed	AMCI SMD34E set when			
	1=power applied	driver fault	position inva	lid bit	
main_power_removal_dete	ected main_power	buffered_SMD34E2_data[1].7	buffered_S	MD34E2_data[0].10	
][][/			
			($\leftarrow \leftarrow \leftarrow$	
			- ↓ r	AMCI_SMD34E2_P	reset_Motor_to_Encoder
			'	AMCI SMD34E2 Preset I	Noto main_power_restore
				AMCI_SMD34E2_output_c	
			L		_
This area a second stars a second		D			
This rung completes recoveri					
	0=power removed 1=power applied	AMCI SMD34E set when driver fault	position inva	lid hit	
main power removal dete		buffered SMD34E2 data[1].7		MD34E2 data[0].10	main_power_removal_detected
		//		/	(U)
		5.0			

File: FAQ_SMD_using_absolute_encoder.doc Date: 2/11/2021