Legal information

Qualified Personnel

This function must only be handled by qualified personnel whose specialization matches the requirements. A qualified person is someone with relevant experience and the ability to identify hazards and risks associated with the operation. Improper handling by unqualified individuals can lead to operational failures and safety risks.

Disclaimer of Liability

We have thoroughly reviewed the contents of this publication to ensure that it aligns with the described hardware and software. However, as some variations may occur, we cannot guarantee complete consistency. The information in this publication is regularly reviewed, and any necessary updates or corrections will be included in future editions.

Proper use of Halow-Tech products

Products should only be utilized for the specific applications outlined in the accompanying catalog and related technical documents. If incorporating products or components from other manufacturers, they must be either recommended or authorized by the relevant company. To ensure safe and trouble-free operation, proper handling during transport, storage, installation, assembly, commissioning, operation, and maintenance is essential. It is important to adhere to the permissible environmental conditions and to follow the guidelines provided in the associated documentation.

HALOW TECH POWERING INDUSTRIAL AUTOMATION

Contents

Description

The DriveXpress PLC function block is designed specifically for precise control of a motor's rotational speed. This function generates a speed setpoint for the motor, ensuring that the acceleration and jerk (rate of change of acceleration) remain within predefined limits. This is essential for applications where smooth, controlled speed transitions are required to protect both the motor and mechanical components from excessive stress.

One of the key features of this function is the inclusion of a Stop input. This allows for an immediate stop of the motor at any time, applying a controlled deceleration based on the configured deceleration value. This ensures that the motor can be safely halted without causing sudden jolts or damaging the system.

This function is crucial for maintaining precise speed control in applications such as conveyors, pumps, and fans, where smooth operation and quick, controlled stopping are necessary. It ensures the motor operates within safe parameters, minimizing wear and tear while providing the flexibility to handle a wide range of speed requirements.

It is important to note that this function only generates the motor speed setpoint. The actual speed control and regulation must be managed by a motor frequency drive, which will adjust the motor speed according to the setpoint. Additionally, the motor control word and status word for communication with the frequency drive must be handled outside of this function, by the user. This separation ensures flexibility in integrating the function with various motor drive systems while maintaining precise control over speed setpoints.

The diagram provided illustrates how the DriveXpress function block integrates into the overall control system, highlighting its role in achieving precise and reliable speed control of the electrical motor.

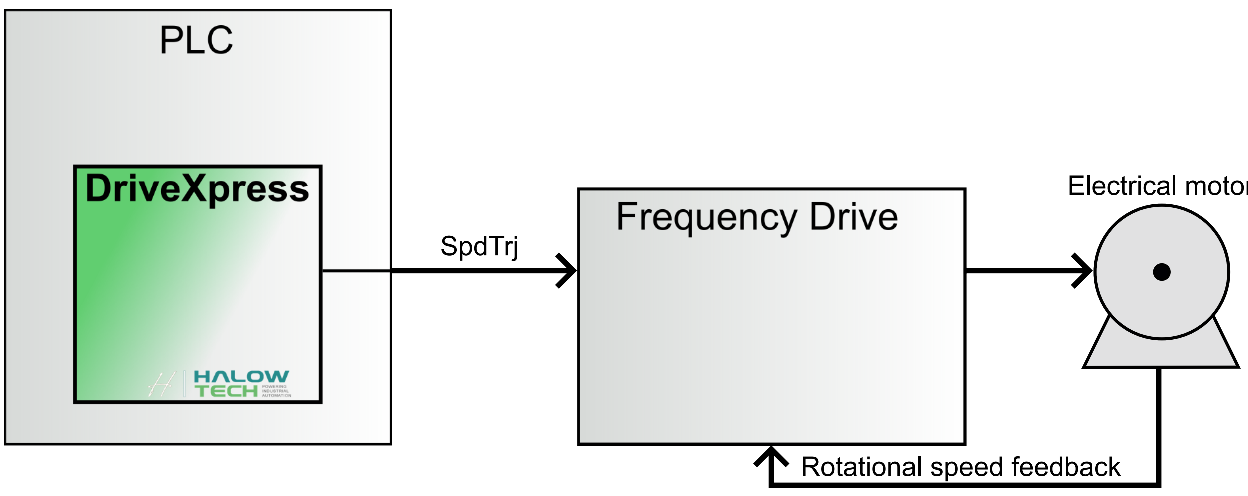
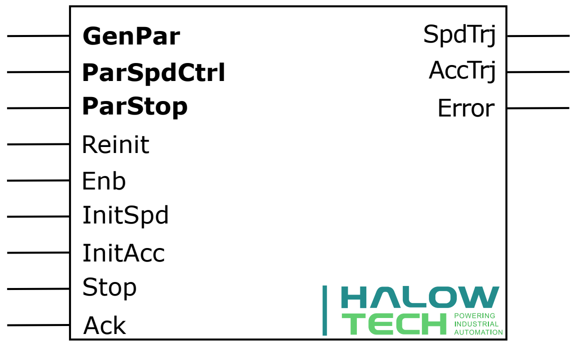


Figure 1: Integration of the function into the overall control system.

Compatible with various PLC platforms like Siemens S7, Siemens TIA Portal, Rockwell Studio AIO, Rockwell RSLogix 5000, Rexroth IndraWorks, B&R Automation Studio, and Beckhoff TwinCAT, FlexiMotion provides the same high performance on any platform.

Block diagram



[figBlock]

Inputs

GenPar	
	SampleTime <REAL>
	Digits <INT>

GenPar → SampleTime - Calling frequency of the controller, <REAL>

The sample time, in second, of the cyclic interrupt task of the plc at which the function is running. A higher sampling frequency allows for more precise control.

GenPar → Digits - Resolution of the calculated movement trajectory, <INT>

The *Digits* input is an integer value that determines the resolution of the decimal places for the calculated motor speed trajectories. This setting controls the precision of the trajectory calculations, ensuring accurate speed transitions. A value of 4 is recommended, which provides a good balance between precision and computational efficiency, allowing the function to generate smooth and accurate speed profiles.

ParSpdCtrl		
	TgtSpd	<REAL>
	MaxAccSetPnt	<REAL>
	JerkSetPnt	<REAL>

ParSpdCtrl → TgtSpd - Target for the motor speed, <REAL>

The *TgtSpd* input defines the desired motor speed setpoint. This input specifies the rotational speed the motor should achieve, and it is typically expressed in units such as RPM (revolutions per minute), rad/s, or as a percentage of the motor's nominal speed, depending on system configuration. This input works in conjunction with the predefined acceleration and jerk limits to ensure smooth and controlled transitions to the target speed. The function will gradually adjust the motor's speed towards the setpoint, adhering to the specified dynamics.

According to figure 2, when the function is enabled (as seen from the *Enb* signal at the top), it immediately responds to any changes in the *TgtSpd* input. The speed trajectory (*SpdTrj*) closely follows the target speed setpoint with smooth transitions, respecting the configured acceleration (*AccTrj*) and jerk limits. Each adjustment in the target speed is reflected in the output *SpdTrj* .

The graph also shows how the acceleration profile (*AccTrj*) adapts according to the changes in the target speed, indicating that the function continuously calculates the necessary acceleration to reach the setpoint.

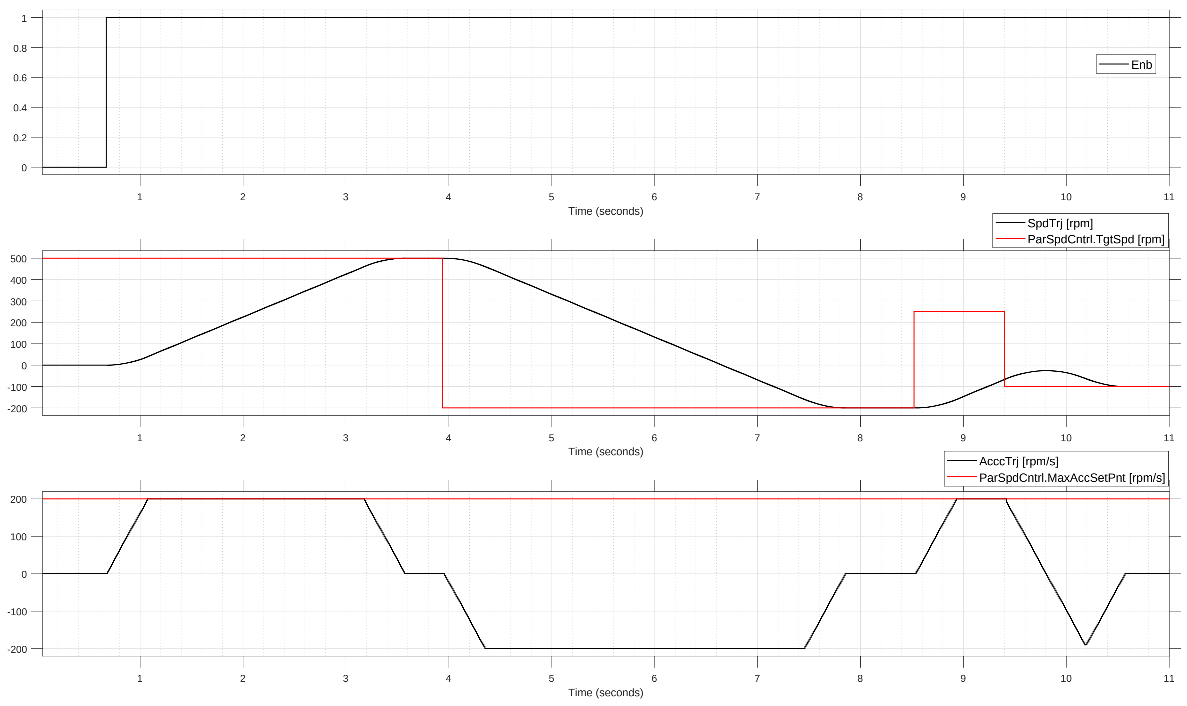


Figure 2: General behavior of the function.

ParSpdCtrl → MaxAccSetPnt - Maximum Motor Acceleration/Deceleration, <REAL>

The *MaxAccSetPnt* input defines the maximum allowable acceleration for the motor when the function is controlling the motor speed. This parameter sets a limit on how quickly the motor can accelerate or decelerate when moving toward the target speed. The value of *MaxAccSetPnt* is typically expressed in units such as rpm/s or rad/s^2 or as a percentage of the motor's nominal speed per second, depending on the system configuration. It ensures that the motor's acceleration stays within safe or desired limits, preventing excessive wear on mechanical components and ensuring smooth transitions during speed changes.

According to figure 3, changes to the *MaxAccSetPnt* input are not possible during a speed transition. As shown, once the speed trajectory is in progress, the function does not respond to adjustments in the maximum acceleration setpoint until the transition has completed.

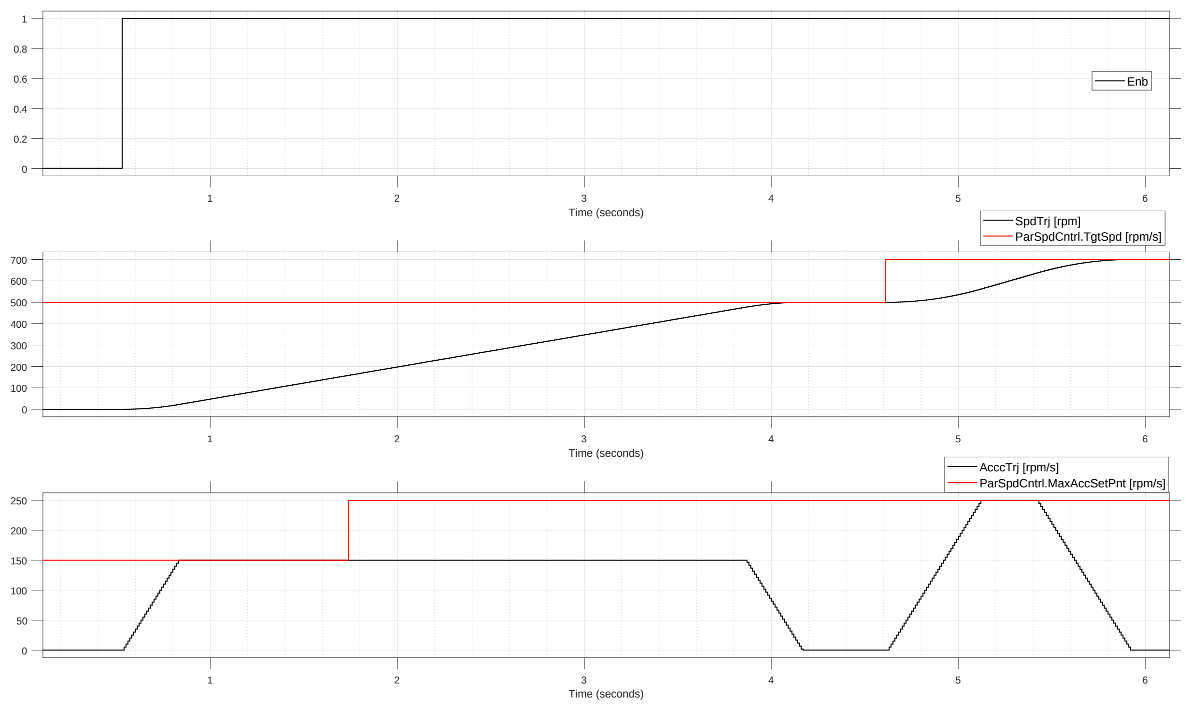


Figure 3: Behavior of the function with different acceleration limit values.

ParSpdCtrl → JerkSetPnt - Maximum Motor Jerk, <REAL>

This input defines the jerk of the motor during speed changes. Jerk is the rate of change of acceleration or deceleration and is measured in units such as rpm/s^2, rad/s^3 or as a percentage of the motor's nominal speed per second cubed.

ParStop	
	DecSetPnt <REAL>

ParStop → DecSetPnt - Maximum deceleration of the stop trajectory, <REAL>

The function block includes a stop feature that allows the motor to be halted immediately. As soon as a rising edge is detected on the *Stop* input, the function triggers a stop ramp to bring the motor to zero speed. The deceleration during this stop is defined by the *DecSetPnt* input, which can be configured in units such as [rpm/s], [rad/s^2], or as a percentage of the nominal speed per second.

After the stop input is activated, normal operation of the function can only resume once the acknowledge (*Ack*) input is activated. This ensures that the system is properly reset and ready for continued use after an emergency or controlled stop. See figure 4.

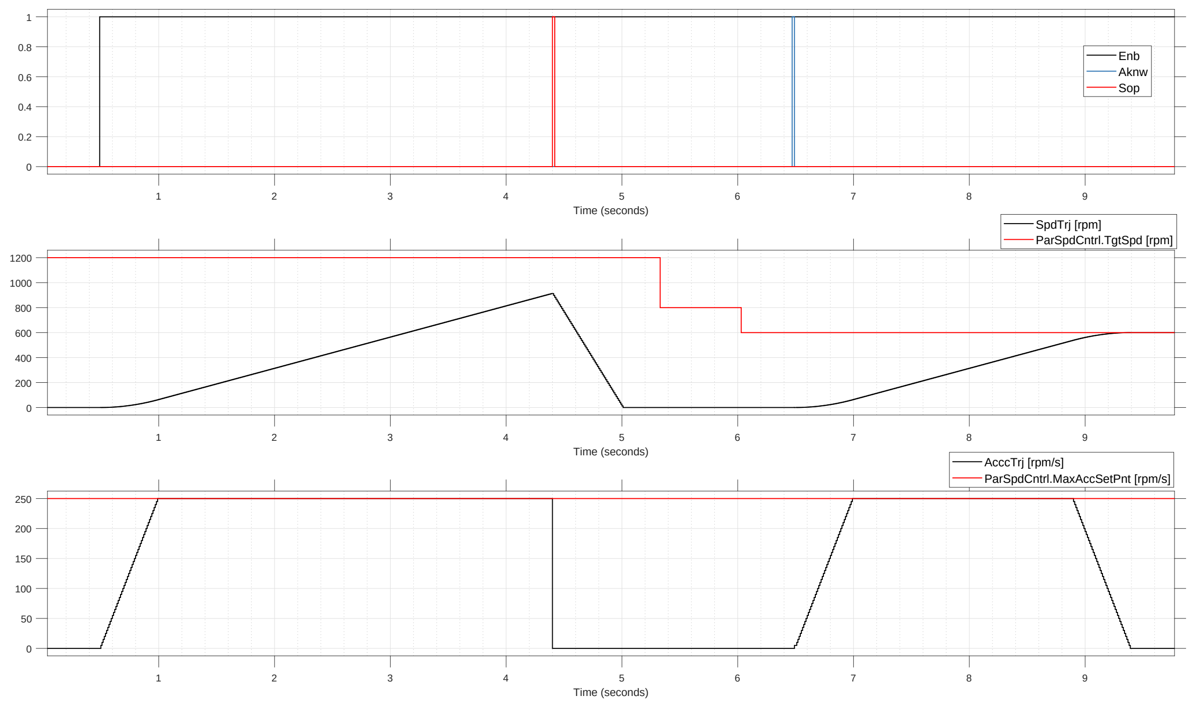


Figure 4: Maximum deceleration of the stop trajectory.

Enb - Turn controller on or off, <BOOL>

The *Enb* input is a boolean signal used to enable or disable the function. When the *Enb* input is set to True, the function block becomes active, allowing it to process inputs and generate outputs . If the *Enb* input is set to False , the function block is deactivated, and it will no longer respond to changes in the input setpoints. Additionally, when the function is disabled, the outputs are reset to zero.

When the *Enb* input transitions from False to True , the function generates a motor speed profile starting from the initial speed defined by *InitSpd* and with an initial acceleration defined by *InitAcc* . See table 1 and figure 5.

	SpdTrj	*AccTrj*
Enb =True	calculated by speed trajectory generator	calculated by acceleration trajectory generator
Enb =False	0.0	0.0

Table 1: Behavior of the output signals depending on the *Enb* input

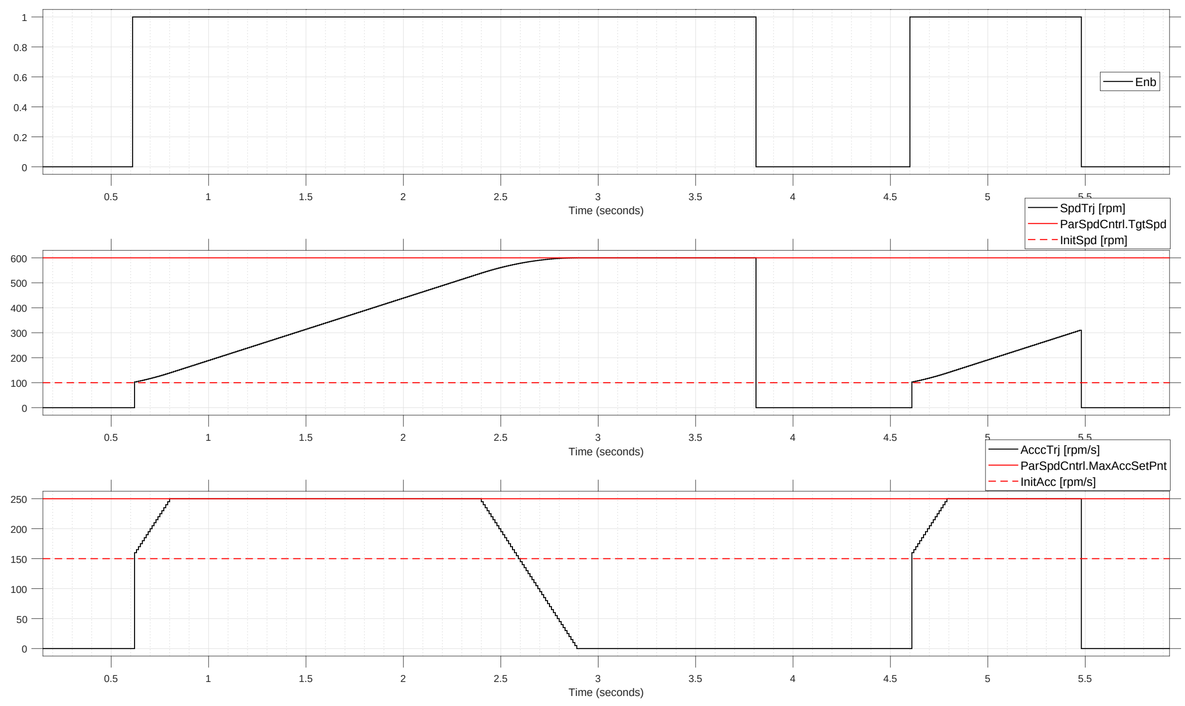


Figure 5: Behavior of the function when enabling and disabling.

InitSpd - Initial speed when enabling the function, <REAL>

See description of input *Enb* .

InitAcc - Initial acceleration when enabling the function, <REAL>

See description of input *Enb* .

Stop - Safety stop, <BOOL>

The *Stop* input is a critical safety and control feature. Regardless of the current value of the output the function block is in, when a rising edge is detected on the *Stop* input, the function immediately sends a zero-speed setpoint to the motor. This command is executed with a configurable deceleration rate, ensuring the motor comes to a controlled and safe stop. This input is essential for emergency stops or situations where an immediate halt of the motor is required, providing a reliable way to quickly bring the system to a stop. See figure 4.

Ack - Acknowledge, <BOOL>

The *Ack* input is used to reset the function block after a stop command has been executed via the *Stop* input. Once the motor has been stopped, the function block is locked, preventing any further operations until a rising edge is detected on the *Ack* input. This input serves as a safety feature, ensuring that the system cannot resume normal operation until the stop event has been acknowledged by the user. It allows the user to verify and confirm that the system is ready to continue, ensuring controlled and safe operation after a stop condition. See figure 4.

Outputs

SpdTrj - Speed trajectory, <REAL>

The *SpdTrj* output is the main output of the function block, responsible for providing the motor speed profile. This output ensures that the motor achieves the target motor speed, or safely stops the motor. The unit of this output—whether rpm, rad/s, or a percentage of the motor's nominal speed—depends on the unit of the input parameters *TgtSpd* .

The *SpdTrj* signal must be connected to a motor speed controller, such as a frequency drive, which translates the setpoint into the actual motor speed.

AccTrj - Acceleration trajectory, <REAL>

The *AccTrj* output represents the acceleration profile of the motor as it's speed transit toward the target speed. See figures 2 to 5.

Error - Error conditions of the function block, <Boolean Array of 2 Elements>

The *Error* output is a boolean array with 2 elements, each indicating a specific error condition that prevents the function block from executing certain operations. Understanding these elements is crucial for diagnosing issues and ensuring the function block operates correctly.

- Element 1, (Invalid Acceleration or Jerk setpoint): If True , no new speed trajectory *SpdTrj* can be calculated. This indicates that one or more of the inputs *MaxAccSetPnt* or *JerkSetPnt* of the struct **ParSpdCtrl** are zero or negative.
- Element 2 (Stop Condition Error): If True , no new speed setpoint can be issued. This element remains True if the stop function has been activated and will only reset when the function's main output *SpdTrj* is zero and the *Ack* input has been activated.