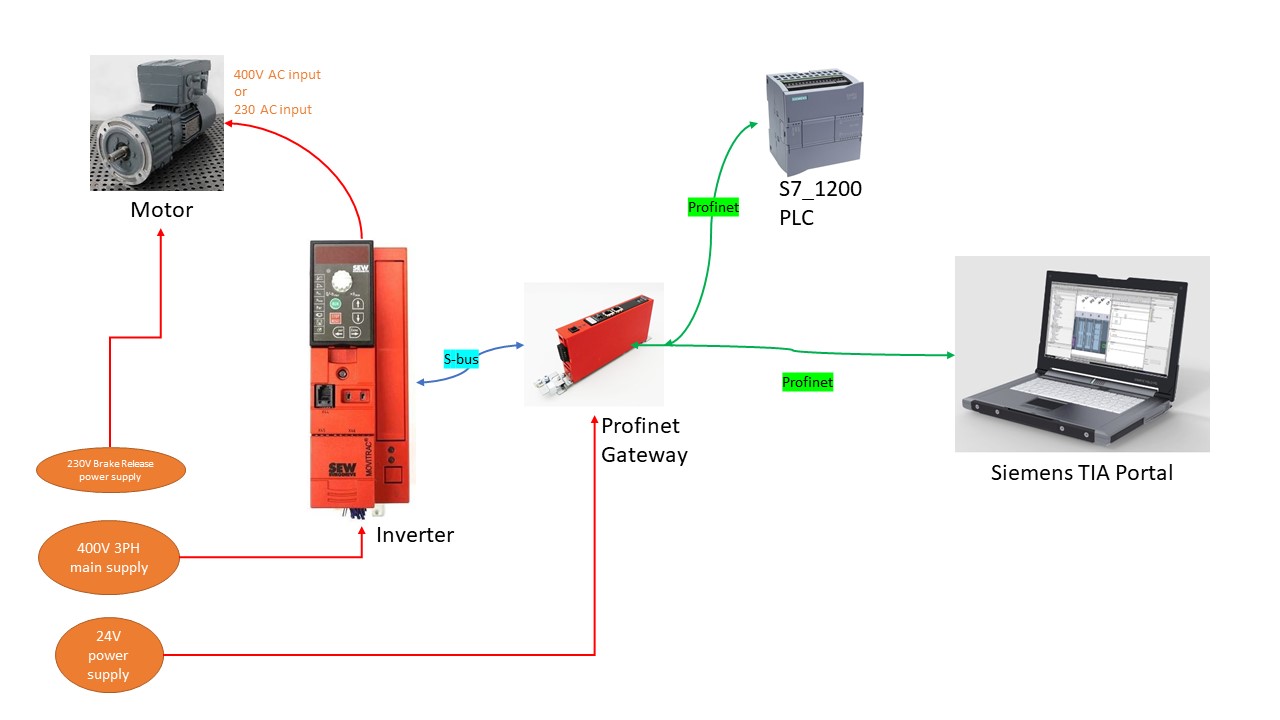
# How to command (speed/ramp/ON\_OFF) a “SEW Movitrac B” Inverter for a “SEW 0.55kW motor” with Siemens S7-1200 PLC



##Motor

SEW Eurodrive

DRS71M4BE1/FI80M/TF

50 Hz

1380 rpm

Vin 230 AC

0.55 KW S1

##Inverter

Movitrac B Inverter

MC07B0030-5A3-4-00/FSC11B

##Gateway

Profinet IO Gateway

DFE32B

Handbook Fieldbus interface DFE32B, Profinet IO

***We used Siemens PLC of model [S7-1212 AC/DC] to send/receive command to/from the inverter. In******case to control ramp/motor-speed from PLC we need to do following things:***

>Connect the UVW wires of the 400V power line to the input of Inverter properly

>Connect the output of Inverter to the input of the motor

>Connect the Inverter and Profinet gateway properly with S-bus cable. This is basically a twisted pair cable which is working with CAN protocol.

>Connect the Gateway to the PLC with an Ethernet cable.

>Connect the Gateway to your Laptop to work with Siemens TIA portal IDE with an Ethernet cable.

>Check all these communications are working.

>Download the sample program from SEW official website to control SEW inverter from Siemens TIA portal. Clean-up the example project by removing the unnecessary hardware from Hardware network diagram, also select the PLC which you have instead of the available default PLC in project.

>Param 8-13 are inputs from INVERTER to PLC, the rest are command from PLC to the Inverter.

>Set proper hardware address in the Function Block and check for “Communication OK” signal to be True.

>The communication in between Gateway and Inverter is in CANbus (S-bus) protocol and the communication in between Gateway and PLC is in Profinet protocol.

***There are 2 key documents to help connect, install and start with SEW inverter.***

# SEW Movitrac B Operating Instructions

Download: <http://www.seweurodrive.com/support/documentation_result.php?gruppen_id=A34&img=241854476>

# SEW DFE32B Profinet IO Gateway Manual

Download: <http://www.seweurodrive.com/support/documentation_result.php?gruppen_id=A34&img=241854476>

***We also need 3 key software files for this setup.***

>GSDML file for Profinet Gateway.

>Movitools Motion Studio

>PLC sample program (TIA portal) for Speed control.

Download: <http://www.seweurodrive.com/support/software_result.php?woher=index&software_produkt=A34&software_searchword=&software_gruppe=&software2=Suchen>

A screenshot of a cell phone

Description automatically generated

A screenshot of a social media post

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Some Key settings:  
>After the S-bus communication is set successfully then the DIP switch on the Gateway should be: DEF IP:=0, AS:=1, Otherwise after power shutdown the IP address of the Gateway will reset & PLC won’t find it anymore !  
>Connect system bus terminating register (S1=ON) at the end of the system bus connection. In this case the S1 DIP switch on the inverter should be switched to 1 as we have only one Inverter.  
>Remember to Bridge the 24V output from X12 terminal of Inverter into DI01 (digital input 01) of the same terminal. This is an input signal to Inverter to be able to run.  
> Remember to check this following setting from Movitools, if it is “CANbus” then we will always get S-bus error. It should be “MoviLink”.

A screenshot of a cell phone

Description automatically generated

Key SCL Code

FUNCTION "speed\_control\_S71xxx\_scl" : Void

{ S7\_Optimized\_Access := 'FALSE' }

VERSION : 0.1

VAR\_INPUT

HW\_IDENTIFIER : HW\_IO; // S7 1x00: assign HW\_Identifier

ENABLE\_RAPID\_STOP : Bool;

ENABLE : Bool;

RESET : Bool;

SETPOINT\_SPEED : Int;

RAMP : Int;

END\_VAR

VAR\_OUTPUT

COMMUNICATION\_OK : Bool;

INVERTER\_READY : Bool;

FAULT\_OF\_AXIS : Bool;

FAULT\_NO : Int;

ACTUAL\_SPEED : Int;

OUTPUT\_CURRENT : Int;

END\_VAR

VAR\_TEMP

PI : Struct

STATUS : Byte;

ENABLE\_OUTPUT\_STAGE : Bool;

INVERTER\_READY : Bool;

PO\_DATA\_ENABLE : Bool;

STW\_Bit\_3 : Bool;

STW\_Bit\_4 : Bool;

ERROR : Bool;

STW\_Bit\_6 : Bool;

STW\_Bit\_7 : Bool;

ACTUAL\_SPEED : Int;

OUTPUT\_CURRENT : Int;

END\_STRUCT;

PO : Struct

DIRECTION\_MOTOR\_POTI : Bool;

MOTOR\_POTI\_UP : Bool;

MOTOR\_POTI\_DOWN : Bool;

FIXED\_INTERNAL\_SETPOINTS : Bool;

FIXED\_INTERNAL\_SETP\_1 : Bool;

FIXED\_SETP\_SWITCHOVER : Bool;

CTW\_Bit\_14 : Bool;

CTW\_Bit\_15 : Bool;

CONTR\_INHIBIT : Bool;

RAPID\_STOP : Bool;

STOP : Bool;

CTW\_Bit\_3 : Bool;

CTW\_Bit\_4 : Bool;

CTW\_Bit\_5 : Bool;

RESET\_FAULT : Bool;

CTW\_Bit\_7 : Bool;

SETPOINT\_SPEED : Int;

RAMP : Int;

END\_STRUCT;

DPRD\_RET\_VAL : Int; // Return Value DPRD\_DAT; <> 0 => error

DPWR\_RET\_VAL : Int; // Return Valoe: DPWR\_DAT; <> 0 => error

Com\_SFC14\_OK : Bool;

Com\_SFC15\_OK : Bool;

END\_VAR

BEGIN

//This sample program shows the basic approach only.

//We do not accept legal or other liability for

//incorrect program functions and their consequences!

//

(\*This function FC134 "speed\_control\_S71xxx\_scl" can be used to control the MOVITRAC frequency inverters on

PROFIBUS DP in conjunction with the Gateway DFP21B or PROFINET IO in

confunction with the gateway DFE32B.

The following parameters deviate from the factory setting:

P100/101 Setpoint source/control source => SBUS;

P881 SBUS address <> 0; see manual "Fieldbus Interface DFP21B"

P872 Setpoint description PO3 => Ramp

=> P876 Enable PO data = ON

FC inputs:

"HW\_IDENTIFIER" S7 1x00: Assign Hardware Identifier of the correspondening slot;

Enable of drive:

Enable via dig. inputs (CW/Stop & Enable/Stop) &

Enable via control word (Enable\_rapid\_stop & Enable)

"Reset" = true: Confirm error

"SETPOINT\_SPEED": Preset of set speed in rpm (integer),

"RAMP": Preset of ramp in ms (integer)

FC outputs:

If data transmission/PROFIBUS is faulty, all outputs will be deleted.

=> "Communication\_OK" = false (self-confirming)

"INVERTER\_READY" = true: No error present & mains voltage ON

"FAULT\_OF\_AXIS" = true: Error is present.

"FAULT\_NO" <> 0: The error codes are listed in the system manual and are

displayed

in MOVITOOLS (status, bus monitor).

"ACTUAL\_SPEED": Actual speed (INT) in rpm

"OUTPUT\_CURRENT": Apparent current (INT) in percentage of rated unit current\*)

#DPRD\_RET\_VAL := DPRD\_DAT(LADDR := #HW\_IDENTIFIER, RECORD => #PI);//DPRD\_DAT: Read consistent data

IF #DPRD\_RET\_VAL = 0 THEN

#COMMUNICATION\_OK := true;//IF no error occurs WHILE the instruction DPRD\_DAT is being executed THEN...

#ACTUAL\_SPEED := #PI.ACTUAL\_SPEED/5; //actual speed 1 digit = 0.2 rpm

#OUTPUT\_CURRENT := #PI.OUTPUT\_CURRENT/10;//actual current 1 digit = 0.1% In => 1% In

#INVERTER\_READY := #PI.INVERTER\_READY;

#FAULT\_OF\_AXIS := #PI.ERROR;

IF #PI.ERROR = true THEN

#FAULT\_NO := #PI.STATUS;

ELSE

#FAULT\_NO := 0;

END\_IF; //depending on the error-bit =1 the fault\_no is displayed; else fault = 0;

//Control bits:

#PO.RAPID\_STOP := #ENABLE\_RAPID\_STOP;

#PO.STOP := #ENABLE;

#PO.RESET\_FAULT := #RESET;

//convert and assign "analog"-values:

#PO.SETPOINT\_SPEED := #SETPOINT\_SPEED \* 5;//actual speed 1 digit = 0.2 rpm

#PO.RAMP := #RAMP;

//Reset of all control word bits not in use:

#PO.CONTR\_INHIBIT := false;

#PO.CTW\_Bit\_3 := false;

#PO.CTW\_Bit\_4 := false;

#PO.CTW\_Bit\_5 := false;

#PO.CTW\_Bit\_7 := false;

#PO.DIRECTION\_MOTOR\_POTI := false;

#PO.MOTOR\_POTI\_UP := false;

#PO.MOTOR\_POTI\_DOWN:=false;

#PO.FIXED\_INTERNAL\_SETPOINTS := false;

#PO.FIXED\_INTERNAL\_SETP\_1:=false;

#PO.FIXED\_SETP\_SWITCHOVER := false;

#PO.CTW\_Bit\_14 := false;

#PO.CTW\_Bit\_15 := false;

ELSE

GOTO END; //If an error occurs during processing of the DP\_RD\_DAT-function, the function will be exited

END\_IF;

#DPWR\_RET\_VAL := DPWR\_DAT(LADDR := #HW\_IDENTIFIER, RECORD := #PO);//DPWR\_DAT: Write consistent data of a DP standard slave

END:

IF #DPRD\_RET\_VAL <> 0 OR #DPWR\_RET\_VAL <> 0 THEN

#COMMUNICATION\_OK := false; //all outputs will be deleted, if processing of the DPRD\_DAT or DPWR\_DA-function is faulty

#INVERTER\_READY := false;

#FAULT\_OF\_AXIS := false;

#FAULT\_NO := 0;

#ACTUAL\_SPEED := 0;

#OUTPUT\_CURRENT := 0;

END\_IF;

END\_FUNCTION

\*\*\* \*\*\* \*\*\* \*\*\* \*\*\* \*\*\* \*\*\* \*\*\* \*\*\* \*\*\* \*\*\* \*\*\* \*\*\* \*\*\* \*\*\* \*\*\* \*\*\* \*\*\*

Explanation:

There are 2 key function in this program. One is Reading data from Inverter to PLC, another writing data from PLC to Inverter.

Following line Reads data from inverter into PLC cyclically:

#DPRD\_RET\_VAL := DPRD\_DAT(LADDR := #HW\_IDENTIFIER, RECORD => #PI);//DPRD\_DAT: Read consistent data

How does this work function [DPRD\_DAT] ? Here is explanation from Siemens help file.

A screenshot of a social media post

Description automatically generated

Following are the parameters we can read from SEW inverter into PLC. Notice that the total length of all the input data is 6 continuous bytes. Within these 6 continuous bytes following parameters are available.

A screenshot of a cell phone

Description automatically generated

Following line Writes data from PLC into inverter cyclically:

#DPWR\_RET\_VAL := DPWR\_DAT(LADDR := #HW\_IDENTIFIER, RECORD := #PO);//DPWR\_DAT: Write consistent data of a DP standard slave

How does this function [DPWR\_DAT] works? Here is explanation from Siemens help file.

A screenshot of a social media post

Description automatically generated

Following are the parameters we can write from PLC into SEW inverter. Notice that the total length of all the output data is 6 continuous bytes. Within these 6 continuous bytes following parameters are available.

A screenshot of a cell phone

Description automatically generated