

Function block H4U.app Position Force

H4U.app xF - TwinCat 3

Copyright

© Bosch Rexroth AG ©2024

All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

Disclaimer

The data specified above only serve to describe the product. As our products are constantly being further developed, no statements concerning a certain condition or suitability for a certain application can be derived from our information. The information given does not release the user from the obligation of own judgment and verification. It must be remembered that our products are subject to a natural process of wear and aging.

English

Table of contents

1	About this documentation	4
1.1	About this documentation.	4
2	Description	4
2.1	Brief description.	4
2.2	Interface description.	4
2.3	Functional description.	6
2.4	Integrating the library.	6
2.5	Implementation example.	7
3	Service and support	9
	Index	11

Table of contents

1 About this documentation

1.1 About this documentation

This documentation is valid for the following products:

- **H4U.app** Position Force



This Quick Start Guide is valid only in conjunction with the application manual "Function block **H4U.app** Position Force" (R.01939-FK).

Read the commissioning instructions and, in particular, the above document completely before working with the function block.



This document describes the function block for a TwinCat 3 system. This function block is therefore referred to as TC3_H4Uapp_pQ in the following description.

2 Description

2.1 Brief description

The software module **H4U.app** xF controls the position (x) or force (F) of a hydraulic axis. It is used to activate valves and pump systems (fixed or variable displacement pump), which control the hydraulic axis.

The software module can be integrated directly in the PLC application of the existing machine control.

The **H4U.app** xF supports various system topologies for valve-controlled and displacement-controlled axes. In the case of valve-controlled axes, the software monitors and compensates system pressures for an optimized velocity feedforward control. For pump-based control concepts the app determines limiting values for the control on the basis of the performance data of the pump and the motor. Operating state monitoring ensures that the pump is operated within the admissible operating limits.

To allow accelerated parameterization pre-defined data sets are available with technical parameters and characteristic curves for Bosch Rexroth pumps and valves. These data sets are directly used as data structure in the PLC application

2.2 Interface description

The input and output data of the software module **H4U.app** xF are shown in the following figure and the following table.

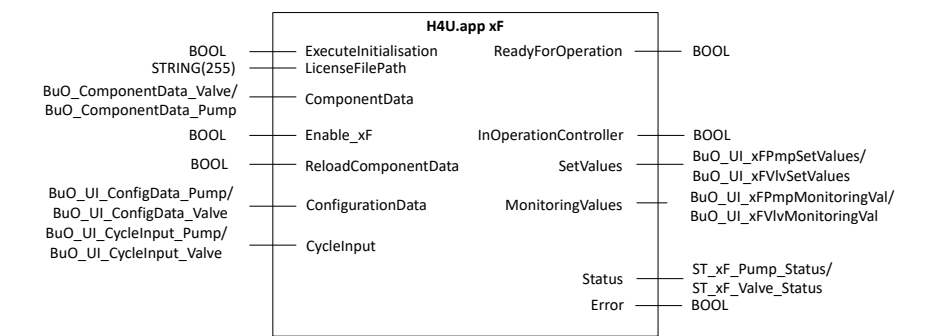


Fig. 1: I/O assignment of the function block.



The structures and functionalities used are described in detail in the Application Manual “Function block **H4U.app** Position Force” (R.01939-FK).

VAR_INPUT						
Name	Type	Min	Max	Def	Unit	Comment
Executelnitialisation	BOOL	False	True	False	[-]	Enable of the non-real-time license and component check
LicenseFilePath	STRING(255)	-	-	-	[-]	Path to license file
ComponentData	BuO_Component-Data_Valve/BuO_Component-Data_Pump	-	-	-	[-]	Configuration of components used
Enable_xF	BOOL	False	True	False	[-]	Enable of real-time-capable controller functionality
ReloadComponent-Data	BOOL	False	True	False	[-]	Reload component data at runtime
ConfigurationData	BuO_UI_ConfigData_Pump / BuO_UI_ConfigData_Valve	-	-	-	[-]	Configuration parameters of the controller. (For details, see application manual)
CyclicInput	BuO_UI_CyclicInput_Pump / BuO_UI_CyclicInput_Valve	-	-	-	[-]	Cyclically required input data (for details, see application manual)
VAR_OUTPUT						
ReadyForOperation	BOOL	False	True	False	[-]	The initialization of the H4U.app has been completed and it is possible to call the functionality of the H4U.app .
InOperation-Controller	BOOL	False	True	False	[-]	The function block executes the controller.
SetValues	BuO_UI_xFPmpSetValues/ BuO_UI_xFVlvSetValues	-	-	-	[-]	Actuating values (for details, see application manual)
MonitoringValues	BuO_UI_xFPmpMonitoringVal/ BuO_UI_xFVlvMonitoringVal	-	-	-	[-]	Diagnostic values (for details, see application manual)

Status	ST_xF_Pump_Status/ ST_xF_Valve_Status	-	-	-	[-]	Status of the app (for details, see application manual)
Error	BOOL	False	True	False	[-]	Shows that an error occurred during the execution.

2.3 Functional description

The function block **H4U.app xF** makes available functions for the closed-loop control of position and force control for hydraulic axes. Before the controller functionality can be used, initializing with license and component database check has to be executed. To this end, first the path to the license file has to be written at input *LicenseFilePath* and the component data at input *ComponentData*. The license and component data check is started with a rising edge at input *ExecutelInitialisation*. After the check is completed, the output switches to *ReadyForOperation* = TRUE.

After successful initialization the controller functionality can be utilized. The function block is parameterized with input *ConfigurationData* and the required actual and command values are applied to input *CyclicInput*. The controller functionality is activated with input *Enable_xF* = TRUE.

Reloading of the component data at runtime can be triggered with the input *ReloadComponentData*. During reloading of component data, the output remains *ReadyForOperation* = TRUE.

As long as the input is set *Enable_xF* = TRUE, license enabling is stored. For a change in licensing, a new license file has to be handed over and initializing has to be restarted with *ExecutelInitialization*. Afterwards, the change is only accepted upon an edge reversal at *Enable_xF*. The license file may only be read in anew while the system is in a safe state.

In the event of an error, output *Error* changes from FALSE to TRUE and the cause of error can be determined from the structure at output *Status*. Invalid component data will not result in an error (for details, see application manual).

2.4 Integrating the library

To install the **H4U.app xF** in the library manager and then to integrate it in an application project proceed as follows:

1. ➤ Download the installation file for library managers of the **H4U.app xF** from the Bosch Rexroth homepage
2. ➤ Activate the license in the Bosch Rexroth licensing portal (for this you require the serial number of the control unit) and download the license file: ➡ <https://licensing.boschrexroth.com/>
3. ➤ Install the **H4U.app** in the library manager of TwinCat 3: ➡ https://infosys.beckhoff.com/content/1031/tc3_plc_intro/41891384434359666059.html?id=4234406035749999087
4. ➤ Store the license file in the file system of the control. The path to the license file then has to be handed over to the **H4U.app**.
 - 📁
 - For some systems, a suitable card reader is required for the memory card used for storing the license file.
 - The file name and the license name can be adapted if you wish to use, for example, a common file path for several projects.

5. ➤ Depending on the actuator use the appropriate module *TC3_H4Uapp_xF_Pump* or *TC3_H4Uapp_xF_Valve* for the next steps.
6. ➤ Integrate the function block in the application project as illustrated in the implementation example. It is recommended that the function block be globally instantiated. Initializing should be carried out in a task with low priority, which can be blocked until initializing is completed. The controller functionality has to be executed in a fast task that meets hard real-time requirements.
7. ➤ Select the component data for Rexroth pumps or valves from the global variables list in the library, which is included in the scope of delivery, and hand it over to the input *ComponentData*. If required, updated component data can be made available by the Bosch Rexroth Service in the form of an XML file.

Alternative: Hand over own component data, for details, see application manual "Function block **H4U.app** Position Force" (R.01939-FK).
8. ➤ Configuring the **H4U.app**: To this end, import the pre-configuration from the available XML, adapt it according to the application at hand and transmit it to the input *ConfigurationData*. Details can be found in the Application Manual "Function block **H4U.app** Position Force" (R.01939-FK).
9. ➤ Link values of the input structure *CyclicInput* and control values in the output structure *SetValues* according to the system topology. For details, see Application Manual "Function block **H4U.app** Position Force" (R.01939-FK).
10. ➤ For controller optimization, diagnostics and troubleshooting: Details can be found in the Application Manual "Function block **H4U.app** Position Force" (R.01939-FK).

2.5 Implementation example

The function block can be integrated as follows:

```
(*##-##-##-##-##-##-##-##-##-##-##-##-##-##-##-##-##-##-##*)

VAR GLOBAL
    // Global functionblock instance
    fbH4Uapp_xF_Pump : TC_H4Uapp_xF_Pump;
END_VAR

(*##-##-##-##-##-##-##-##-##-##-##-##-##-##-##-##-##-##-##*)

FUNCTION_BLOCK Initialisation_Pump
VAR
    bError      : BOOL; // Indicates that an error occurred
    bDoInit     : BOOL; // Initialisation is done
    stStatus    : ST_xF_Pump_Status; // Status of the building block components

    strRexrothPumpData : BuO_ComponentData_Pump; // Configuration of the controller

    fbRTrig       : R_TRIG; // Trigger FB to detect rising edge of Execute
    fbPumpDataBase : PumpDataBase; // fb provides the pump data of the choosen pump
    sFilePath     : STRING(255) := 'Hard Disk\Rexroth\license.lic'; //license path
END_VAR

(*##-##-##-##-##-##-##-##-##-##-##-##-##-##-##-##-##-##-##*)
```

```
// Detect rising edge of trigger signal
fbRTrig(CLK:= bDoInit);

// Check if initialisation shall be started
IF fbRTrig.Q THEN
    bDoInit := FALSE;
    GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.ExecuteInitialisation := TRUE;
    GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.LicenseFilePath := sFilePath;
    // Get Pump data from provided database
    fbPumpDataBase(CfgPumpType:= BRIHC_PumpIdentifier.A10FZO_003, PumpDb=>
stRexrothPumpData.PumpDataBase[0]);
    // Set Pump data from provided database
    GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.ComponentData          := stRexrothPumpData;
END_IF

// Call functionblock only when initialisation is triggered or still active
IF (fbRTrig.Q OR
GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.Status.StatusLicensing.ActiveInitialisation
OR (GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.Status.Status_xF_Pump.ActiveInit
AND NOT GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.ReadyForOperation)) THEN
    GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump();
END_IF

// Reset execute bit after initialisation is done
IF NOT
GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.Status.StatusLicensing.ActiveInitialisation
AND NOT GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.Status.Status_xF_Pump.ActiveInit THEN
    GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.ExecuteInitialisation := FALSE;
END_IF

IF GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.Error THEN
    bError      := TRUE;
    stStatus     := GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.Status;
ELSE
    bError      := FALSE;
END_IF

(***-#-#-#-#-#-#-#---Declaration- Real Time Task---#-#-#-#-#-#-#-#-#***)

FUNCTION_BLOCK Controller_Pump
VAR
    bReloadComponentData : BOOL; // Reinitialisation of component data at runtime
    bError                : BOOL; // Indicates that an error occurred
    stStatus              : ST_xF_Pump_Status; // Status of the building block components

    // controller input values
    rActForce             : REAL; // Actual Force value
    rDesForce            : REAL; // Desired Force value
    rActPosition          : REAL; // Actual Position value
    rDesPosition         : REAL; // Desired Position value

    // controller output Values
    rValveCmd            : REAL;
    rSpeedCmd           : REAL;

    stConfiguration      : BuO_UI_ConfigData_Pump; // Configuration of the controller

    fbRTrig              : R_TRIG; // Trigger FB to detect rising edge of Execute
    fbPumpDataBase       : PumpDataBase; // fb provides the pump data of the choosen pump
END_VAR

(***-#-#-#-#-#-#-#---Implementation - Real Time Task---#-#-#-#-#-#-#-#-#***)

// code for real time controller task
fbRTrig(CLK:= bReloadComponentData);
```



```

// Set configuration Values
GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.ConfigurationData:= stConfiguration;

// Map actual and desired values
GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.CyclicInput.ForceAct := rActForce;
GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.CyclicInput.ForceCmd := rDesForce;
GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.CyclicInput.PositionAct := rActPosition;
GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.CyclicInput.PositionCmd := rDesPosition;

// // Get Pump data from provided database when positive flank occurs
IF fbTrig.Q AND bReloadComponentData THEN
    fbPumpDataBase(CfgPumpType:= BRIHC_PumpIdentifier.A10FZO_003,PumpDb =>
GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.ComponentData.PumpDataBase[0]);
END_IF

// Call functionblock when it is ready for operation
IF GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.ReadyForOperation AND bReloadComponentData
THEN
    GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump(ReloadComponentData := bReloadComponentData);
ELSE
    GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump();
END_IF

// Reset flag after reloading component data
IF GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.ReadyForOperation
AND GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.Status.Status_xF_Pump.DoneInit THEN
    bReloadComponentData := FALSE;
END_IF

//Set outgoing command values
rValveCmd := GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.SetValues.ValveCmd[0];
rSpeedCmd := GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.SetValues.SpeedCmd[0];

// Check for errors
IF GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.Error THEN
    GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.Enable_xF := FALSE;
    stStatus := GVL_H4Uapp_xF_Pump_Sample.fbH4Uapp_xF_Pump.Status;
    bError := TRUE;
ELSE
    bError := FALSE;
END_IF

```

3 Service and support

Contact

Bosch Rexroth AG

Service Industriehydraulik

Bürgermeister-Dr.-Nebel-Straße 8

97816 Lohr am Main

Telefon +49 (0) 9352 405060

Germany

Email: service@boschrexroth.com

Internet: ➔ www.boschrexroth.com/service

For questions about the product

Bosch Rexroth AG

Phone +49 (0) 9352 403020

Email: my.support@boschrexroth.com

Preparation of information

We can help you in a fast and efficient way if you keep the following information available:

- Details about the product concerned, particularly the type code and serial number.
- Your contact details (phone number, e-mail address)

Index

A
About this documentation. 4

B
Brief description. 4

D
Description. 4

F
Functional description. 6

I
Implementation example. 7
Interface description. 4

L
Library. 6

S
Service and support. 9

Bosch Rexroth AG
Zum Eisengießer 1
97816 Lohr a.Main
Germany
Tel. +49 9352 18-0
www.boschrexroth.com



RE018801-Z