rexroth
A Bosch Company

Quick Start Guide

Software module H4U.app Pressure Flow

H4U.app pQ - TIA Portal

RE01936-02-Z Edition 2025-10

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.

Table of contents

1 About this documentation

1.1 About this documentation

This documentation is valid for the following products:

- **H4U**.app Pressure Flow

– This Quick Start Guide is valid only in conjunction with the Application Manual “software module **H4U**.app Pressure Flow" (R.01936-FK).
– Read the commissioning instructions and, in particular, the above document completely before working with the software module.
– Please observe the Open Source License linked below:
➦ Royalty-free Software provided by Siemens on sharing platforms for developers/users of Siemens products

This document describes the software module for a TIA Portal. This software module is therefore referred to as TIA_H4Uapp_pQ in the following description.

2 Description

2.1 Brief description

The software module **H4U**.app pQ is used for open and closed-loop pressure and flow control. It supports the following pump types as actuator:

- Fixed displacement pumps
- Variable displacement pumps

A component database is available for the following Bosch Rexroth pumps:

- Internal gear pumps: PGH, PGF, PGM
- Axial piston pumps: A10VSO, A4VSO, A10FZO, A10VZO, A4VBO

For axial piston pumps, various Bosch Rexroth actuating and control systems are supported:

- HS and DFE control
- EZ control
- DR closed-loop control

The software module **H4U**.app pQ controls the pressure and/or flow of a system. In addition, it features operating strategies for optimizing the system. The valid operating limits for this are generated in the operating state control of the **H4U**.app pQ. The software module **H4U**.app pQ includes an initialization call, in which a license check is run and the component database is evaluated.

2.2 Interface description

The input and output data of software module “TIA_H4Uapp_pQ“ are shown in the following figure and the following table.

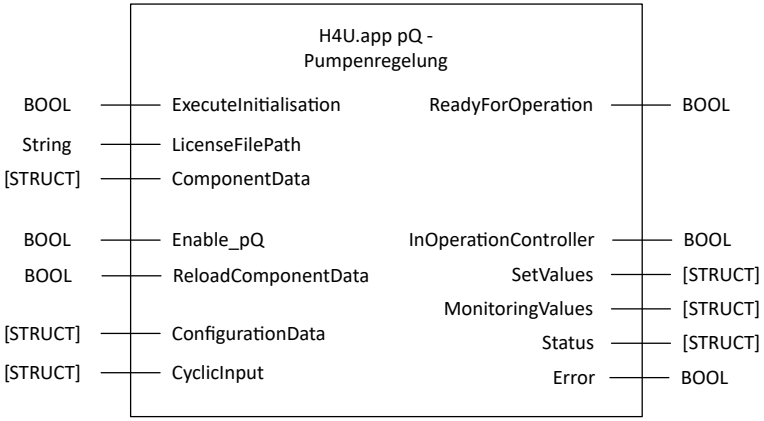

Fig. 1: I/O assignment of the software module

The structures and functionalities used are described in detail in the Application Manual “software module **H4U**.app pQ" (R.01936-FK).

VAR_INPUT

Name	Type	Min	Max	Def	Unit	Comment
ExecuteInitialisation	BOOL	False	True	False	[-]	Enable of the non-real-time license and component check
LicenseFilePath	STRING(255)	-	-	-	[-]	Path to license file
ComponentData	BuO_ComponentDatabase	-	-	-	[-]	Configuration of components used
Enable_pQ	BOOL	False	True	False	[-]	Enable of real-time-capable controller functionality
ReloadComponentData	BOOL	False	True	False	[-]	The component database is reloaded
ConfigurationData	BuO_Configuration	-	-	-	[-]	Configuration parameters of the controller. (For details, see application manual)
CyclicInput	BuO_CyclicInput	-	-	-	[-]	Cyclically required input data (for details, see application manual)

VAR_OUTPUT

Description

ReadyForOperation	BOOL	False	True	False	[-]	The license and component check has been completed and the software module is ready for the controller enable
InOperationController	BOOL	False	True	False	[-]	The software module executes the controller
SetValues	BuO_SetValues	-	-	-	[-]	Controller outputs (for details, see Application Manual)
MonitoringValues	BuO_MonitoringValues	-	-	-	[-]	Diagnostic values (for details, see Application Manual)
Status	ST_pQ_Status	-	-	-	[-]	See definition of ST_pQ_Status
Error	BOOL	False	True	False	[-]	Shows that an error occurred during the execution.

2.3 Functional description

Software module TIA_H4Uapp_pQ provides functions for the closed-loop control of pressure and flow of hydraulic pumps. Before the controller functionality can be used, initializing with license and component database check has to be executed. To this end, the path to the license file has to be written at input *LicenseFilePath* and the component data at input *ComponentData*. The check is started with a rising edge at input *ExecuteInitialisation.* Once the check is successfully completed, the output *ReadyForOperation* changes from FALSE to TRUE. 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*.

After successful initialization the controller functionality can be utilized. The software module 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_pQ* = TRUE.As soon as the controller function is active, output *InOperationController* switches from FALSE to TRUE.

It is recommended that the software module 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 cyclic interrupt OB. Real-time requirements fulfilled.

2.4 Integrating the library

To be able to use the **H4U**.app pQ in TIA Portal, the provided library has to be imported:

1. 1. Download the library file of the H4U.app pQ from the Bosch Rexroth server and open it in the TIA Portal.
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/

   For licensing enter the serial number of the control without blank before the year.
3. Open the H4U.app as global library in the TIA Portal.

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.

   ℹ For some systems, a suitable card reader is required for the memory card used for storing the license file.

   ℹ 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. Integrate the software module in the application project as illustrated in the implementation example.

6. Select the component data for Rexroth pumps from the available database and provide it to the input *ComponentData*. (Alternatively, provide own pump data)

7. Select the available pre-configuration, adapt it according to the application at hand and transmit it to the input *ConfigurationData*. Details can be found in the Application Manual “software module **H4U**.app pQ" (R.01936-FK).

8. Link values of the input structure *CyclicInput* according to the system topology. Details can be found in the Application Manual “software module **H4U**.app pQ" (R.01936-FK).

9. For controller optimization, diagnostics and troubleshooting: Details can be found in the Application Manual “software module **H4U**.app pQ" (R.01936-FK).

2.5 Implementation example

The software module can be integrated as follows:

```
DATA_BLOCK "GVL"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
NON_RETAIN
   VAR
      bDoInit : Bool;
      bError : Bool;
      sFilePath : String;
      bCallController : Bool;
      stStatus { S7_SetPoint := 'False'} : "ST_pQ_Status";
      stConfigData : "BuO_Configuration";
      stComponentData : "BuO_ComponentDatabase";
      stCyclicInput : "BuO_CyclicInput";
   END_VAR


BEGIN
   sFilePath := 'UserFiles/License.lic';

END_DATA_BLOCK

DATA_BLOCK "fbTIA_H4U_app_pQControlPump"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
NON_RETAIN
"TIA_H4U_app_pQControlPump"

BEGIN

END_DATA_BLOCK
```

```
ORGANIZATION_BLOCK "Initialization"

TITLE = "Main Program Sweep (Cycle)"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
  Initial_Call : Bool;(*Initial call of this OB*)
  Remanence : Bool;(*=True, if remanent data are available*)
END_VAR

VAR_TEMP

END_VAR

VAR CONSTANT

END_VAR

BEGIN
          //Detect Rising Edge for Initialisation
          "fbR_TRIG"(CLK := "GVL".bDoInit);
          IF "fbR_TRIG".Q THEN
            "GVL".bDoInit := FALSE;
            "fbTIA_H4U_app_pQControlPump".ComponentData := "GVL".stComponentData;
            "fbTIA_H4U_app_pQControlPump".LicenseFilePath := "GVL".sFilePath;
            "fbTIA_H4U_app_pQControlPump".ExecuteInitialisation := TRUE;
          END_IF;

          //Call function block only when initialisation is triggered or still active
          IF "fbR_TRIG".Q
         OR "fbTIA_H4U_app_pQControlPump".Status.StatusLicensing.ActiveInitialisation
         OR "fbTIA_H4U_app_pQControlPump".Status.ActiveInit
         AND NOT "fbTIA_H4U_app_pQControlPump".ReadyForOperation THEN
                 "fbTIA_H4U_app_pQControlPump"();
          END_IF;

          //Reset Execute after initialisation is done and enable call of controller
          IF "fbTIA_H4U_app_pQControlPump".Status.DoneInit THEN
            "fbTIA_H4U_app_pQControlPump".ExecuteInitialisation := FALSE;
             "GVL".bCallController := TRUE;
          END_IF;

          //Check for errors
          IF "fbTIA_H4U_app_pQControlPump".Error THEN
            "GVL".bError := TRUE;
            "GVL".stStatus := "fbTIA_H4U_app_pQControlPump".Status;
          ELSE
            "GVL".bError := FALSE;
          END_IF;
END_ORGANIZATION_BLOCK



ORGANIZATION_BLOCK "Cyclic interrupt"

{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
  Initial_Call : Bool;(*Initial call of this OB*)
  Event_Count : Int;(*Events discarded*)
END_VAR

VAR_TEMP
```

```
END_VAR

VAR CONSTANT

END_VAR

BEGIN
  IF "GVL".bCallController THEN
          //Set Configuration Data and cyclic input
          "fbTIA_H4U_app_pQControlPump".ConfigurationData := "GVL".stConfigData;
          "fbTIA_H4U_app_pQControlPump".CyclicInput := "GVL".stCyclicInput;

          //Set enabling of controller when ready for operation
          //Add additional custom check for enabling control --> Moving machine
          IF "fbTIA_H4U_app_pQControlPump".ReadyForOperation THEN
            "fbTIA_H4U_app_pQControlPump".Enable_pQ := TRUE;
          END_IF;

          //Check for errors
          IF "fbTIA_H4U_app_pQControlPump".Error THEN
            "fbTIA_H4U_app_pQControlPump".Enable_pQ := FALSE;
            "GVL".bError := TRUE;
            "GVL".stStatus := "fbTIA_H4U_app_pQControlPump".Status;
          ELSE
            "GVL".bError := FALSE;
          END_IF;

          //Call fb H4u app pQ
          "fbTIA_H4U_app_pQControlPump"();
  END_IF;
END_ORGANIZATION_BLOCK


DATA_BLOCK "fbR_TRIG"
{InstructionName := 'R_TRIG';
 LibVersion := '1.0';
 S7_Optimized_Access := 'TRUE' }
AUTHOR : SIMATIC
FAMILY : BIT
NAME : R_TRIG
VERSION : 1.0
NON_RETAIN
R_TRIG

BEGIN

END_DATA_BLOCK
```

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

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

RE01938-02-Z