

Van den Berg, Industriële Automatisering

biaCYL_lib product brochure

Further information on www.berg-automatisering.nl
info@berg-automatisering.nl

Library 'biaCYL_lib' is intended for B&R platforms.

The 'biaCYL_lib' library contains functions and function blocks that can be used for the control of pneumatic and hydraulic valves and cylinders.

Function block 'CYL_DiscreteCylinder' (described on the next pages)
Contains functionality for operating a valve and/or cylinder (via a discrete valve).

This functionality includes:

- *support of 5/2 monostable and bistable valves, and 5/3 valves
- *support of position detection on 'work' and/or 'return' position
- *support of a simulated travel-time when no position detection used
- *monitoring of status and performance (traveltime, raising of alarms and warnings on several criteria)
- *inhibit-functionality (inhibiting of movements)
- *support of manual operation (via HMI or SCADA system)
- *interface built with 'enumerate' type variables (easy application debugging)

Configurable are:

- *valve-type and end-effector detection type
- *device ID (numerical and in text)
- *endstate definitions (in text, e.g. 'up', 'reject' etc. for HMI and SCADA)
- *default- and maximum allowable traveltimes
- *error-, warning- display-offsets (for HMI purposes)

Monitoring of operation:

- *monitoring and guarding of traveltimes
- *monitoring and guarding of end effector performance and failures
- *guarding of inhibit-inputs (inhibiting of movements)
- *guarding of invalid commands or configurations

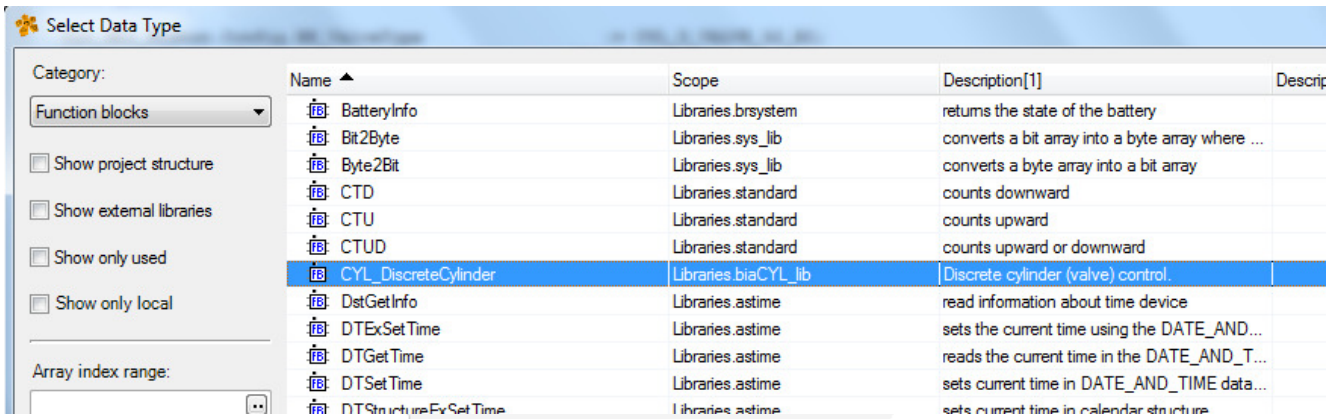
Disclaimer

Although every attempt has been made to keep the information in this brochure up to date, Van den Berg Industriële Automatisering cannot guarantee that all information is current at all points in time. Information in this brochure is periodically reviewed and subject to change.

Van den Berg Industriële Automatisering is not liable to anyone for inaccuracies, errors, or omissions in the provided information.

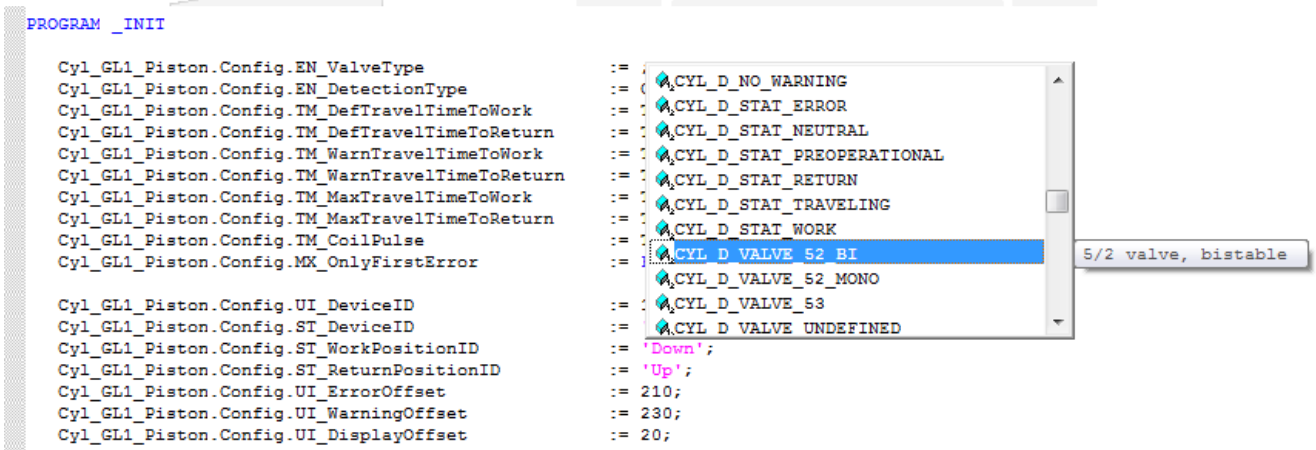
Van den Berg Industriële Automatisering is not liable for any damages incurred as a result of inaccuracies, errors, or omissions in the provided information.

Declaration



Easy function block declaration in Automation Studio.

Configuration



Easy configuration of the function block due to the use of 'enumerate' configuration variables, and the 'IntelliSense' feature in Automation Studio.

PROGRAM _INIT

```

Cyl_GL1_Piston.Config.EN_ValveType           := CYL_D_VALVE_S2_BI;
Cyl_GL1_Piston.Config.EN_DetectionType      := CYL_D_DETECTION_WR;
Cyl_GL1_Piston.Config.TM_DefTravelTimeToWork := T#99s;
Cyl_GL1_Piston.Config.TM_DefTravelTimeToReturn := T#99s;
Cyl_GL1_Piston.Config.TM_WarnTravelTimeToWork := T#1500ms;
Cyl_GL1_Piston.Config.TM_WarnTravelTimeToReturn := T#800ms;
Cyl_GL1_Piston.Config.TM_MaxTravelTimeToWork := T#2000ms;
Cyl_GL1_Piston.Config.TM_MaxTravelTimeToReturn := T#1000ms;
Cyl_GL1_Piston.Config.TM_CoilPulse          := T#500ms;
Cyl_GL1_Piston.Config.MX_OnlyFirstError     := FALSE;

Cyl_GL1_Piston.Config.UI_DeviceID           := 123;
Cyl_GL1_Piston.Config.ST_DeviceID          := 'GL1_Piston';
Cyl_GL1_Piston.Config.ST_WorkPositionID     := 'Down';
Cyl_GL1_Piston.Config.ST_ReturnPositionID   := 'Up';
Cyl_GL1_Piston.Config.UI_ErrorOffset        := 210;
Cyl_GL1_Piston.Config.UI_WarningOffset      := 230;
Cyl_GL1_Piston.Config.UI_DisplayOffset      := 20;

Cyl_GL1_Piston.Control.MX_Powerdown         := FALSE;
Cyl_GL1_Piston.Control.MX_Reset             := ;
Cyl_GL1_Piston.Control.MX_AcknowledgeError  := ;
Cyl_GL1_Piston.Control.MX_AcknowledgeWarning := FALSE;

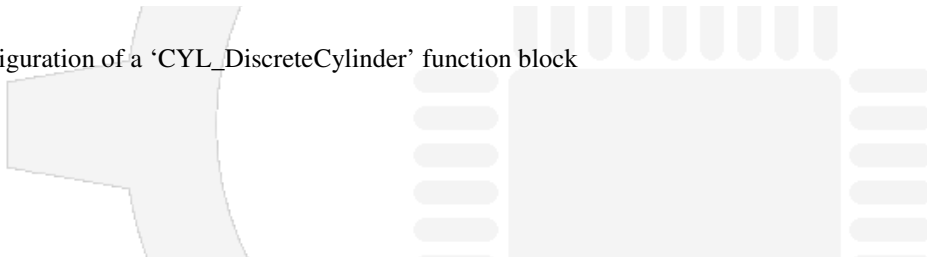
Cyl_GL1_Piston.Control.MX_Initialize        := TRUE;
Cyl_GL1_Piston(HmiControl := CYL_Cylinder_HmiControl, HmiStatus := CYL_Cylinder_HmiStatus);

(* HMI related *)
Cyl_GL1_Piston.Control.MX_HmiActive         := TRUE;

CYL_Cylinder_HmiControl.MX_Authorized       := ;
CYL_Cylinder_HmiControl.MX_Exclusive       := FALSE;

```

Complete configuration of a 'CYL_DiscreteCylinder' function block



PROGRAM _INIT

```

Cyl_GL1_Piston.Config.EN_ValveType           := CYL_D_VALVE_S2_BI;
Cyl_GL1_Piston.Config.EN_DetectionType      := CYL_D_DETECTION_WR;
Cyl_GL1_Piston.Config.TM_DefTravelTimeToWork := T#99s;
Cyl_GL1_Piston.Config.TM_DefTravelTimeToReturn := T#99s;
Cyl_GL1_Piston.Config.TM_WarnTravelTimeToWork := T#1500ms;
Cyl_GL1_Piston.Config.TM_WarnTravelTimeToReturn := T#800ms;
Cyl_GL1_Piston.Config.TM_MaxTravelTimeToWork := T#2000ms;
Cyl_GL1_Piston.Config.TM_MaxTravelTimeToReturn := T#1000ms;
Cyl_GL1_Piston.Config.TM_CoilPulse          := T#500ms;
Cyl_GL1_Piston.Config.MX_OnlyFirstError     := FALSE;

```

TIME CYL_DiscreteConfig_type.TM_MaxTravelTimeToWork
 local
 Maximum allowed traveltime to work position (used when work position sensor configured)

```

Cyl_GL1_Piston.Config.UI_DeviceID           := 123;
Cyl_GL1_Piston.Config.ST_DeviceID          := 'GL1_Piston';

```

All members of the configuration, control and status data structures are fully documented. Automation Studio pops up description boxes when hovered over the structure-members.

Application use

```

END_IF

DispenseGlue:
(* Send glue piston upwards if dispense-time reached *)
IF TON_GlueDispenseTime.Q THEN
  Cyl_GL1_Piston.Control.EN_Command := CYL_D_CMD_RETURN;

  SEQ_GlueCycle.Control.MX_StepUp := TRUE;
END_IF

WaitForPistonUpwards:
(* If glue piston in upper position, start dry-pulse *)
IF Cyl_GL1_Piston.Status.EN_State = CYL_D_STAT_RETURN THEN
  MX_StartDryPulse := TRUE;

  SEQ_GlueCycle.Control.MX_StepUp := TRUE;
END_IF

```

The use of libraries (or function blocks) will keep your main application concise and well-arranged.

Commissioning and debugging

[-] Cyl_GL1_Piston	CYL_DiscreteCy	local	
[-] Config	CYL_DiscreteCo		
[-] Control	CYL_DiscreteCo		
[-] MX_Initialize	BOOL		FALSE
[-] EN_Command	CYL_DiscreteCo		CYL_D_CMD_IDLE
[-] MX_AllowTravelToWork	BOOL		TRUE
[-] MX_AllowTravelToReturn	BOOL		FALSE
[-] MX_Powerdown	BOOL		FALSE
[-] MX_Reset	BOOL		FALSE
[-] MX_AcknowledgeError	BOOL		FALSE
[-] MX_AcknowledgeWarning	BOOL		FALSE
[-] MX_HmiActive	BOOL		TRUE
[-] IX_InWorkPosition	BOOL		TRUE
[-] IX_InReturnPosition	BOOL		FALSE
[-] Status	CYL_DiscreteSt:		
[-] EN_State	CYL_DiscreteSt:		CYL_D_STAT_WORK
[-] TM_TravelTimeToWork	TIME		T#1s_214ms
[-] TM_TravelTimeToReturn	TIME		T#733ms
[-] MX_Error	BOOL		FALSE
[-] EN_Error	CYL_DiscreteErr		CYL_D_NO_ERROR
[-] UI_ErrorNumber	UINT		0
[-] DI_ErrorExtraInfo	DINT		0
[-] MX_Warning	BOOL		FALSE
[-] EN_Warning	CYL_DiscreteW:		CYL_D_NO_WARNING
[-] UI_WarningNumber	UINT		0
[-] IX_InReturnPosition	BOOL		FALSE
[-] Status	CYL_DiscreteSt:		
[-] EN_State	CYL_DiscreteSt:		CYL_D_STAT_ERROR
[-] TM_TravelTimeToWork	TIME		T#1s_196ms
[-] TM_TravelTimeToReturn	TIME		T#738ms
[-] MX_Error	BOOL		TRUE
[-] EN_Error	CYL_DiscreteErr		CYL_D_ERR_START_TO_RETURN_BLOCK
[-] UI_ErrorNumber	UINT		223
[-] DI_ErrorExtraInfo	DINT		0

Easy testing, debugging and commissioning due to the use of 'enumerate' variables in a 'watch-screen'