

24V production line

Data blocks



Table of contents

6 C	Data	blocks	.1
6.1	Int	troduction	.1
6.2	(Global data blocks	3
6.3		Schematic structure of a data block	4
6.4		Instance data block	6
6.5		Testing data blocks	8
6	5.5.1	Observing in the data module	8
6	5.5.2	Controlling operands in the data block	9
6	5.5.3	Observing in an observation table	0
6.6		Exercise: Declaring workpiece management	11

6 Data blocks

6.1 Introduction

Data blocks can be used in the TIA project to store data. Unlike flags, data can be stored in the data block in a structured manner and made available for processing. Variables of different data types can be stored in any order. The following image shows examples of data to be stored:



Picture 1 Structured data management

Unlike functions and function blocks, data blocks do not contain any program code information, i.e. no executable program.

They offer the following advantages:

- Order and structure in the project
- Better overview of system data
- Faster access in the PLC thanks to optimized access (no absolute addressing)
- Flag variables can be dispensed with
- Reusability in different projects without conflicts with addresses
- DB variables can be pre-assigned with start values
- Monitoring of variables directly in the DB possible
- Simple backup of actual values





fischertechnik 📼

Data block types

There are basically two types of DB.

Global data blocks

- The content and data structure are defined by the user.
- The data can be addressed globally by the user.

Instance data blocks

- The content and the data structure are defined by the calling function module through its interface.
- The instance data block is generated automatically when the FB is called.
- The FB instance (call) accesses the data of its instance DB directly via local interfaces.



Picture 2 Data block types



6.2 Global data blocks

Global data blocks are used to store user data (variables) that can be used by all code blocks. The data structure within a global data block is defined by the programmer.

Global data blocks in the user program

Each FB, FC or OB can read data from a global DB or write data to a global DB.

Examples of data to be stored are

- Setpoint or parameter values
- Data area for error messages
- Interface data
- Material tracking data
- Time specifications for timer modules

The use of data blocks is recommended for data traffic between different systems:

- Frequency inverter control
- CPU-CPU communication
- HMI connection



Picture 3 Access to data of a global DB



6.3 Schematic structure of a data block

You can think of a data block as a shelf. The shelf has a name, the data block name. You can create compartments (variables) of different sizes on this shelf (data type). The size is defined by the data type. Each compartment is also given a name.

The data is then accessed via:

Shelf name and subject name \rightarrow "Datenbausteinname". Variable name



Picture 4 Schematic structure of a data block

The length of a data block can be up to 16 MB.

In the standard definition, the data is automatically stored by the system in the data module. This is an optimized data block.

The maximum number of data blocks is limited by the PLC's memory. **Structure of the data block in the editor** The following figure shows the structure of a data block in the declaration of the structure of a data block in the declaration of the structure of a data block in the declaration of the structure of a data block in the declaration of the structure of a data block in the declaration of the structure of a data block in the declaration of the structure of a data block in the declaration of the structure of a data block in the structure of a data b

The following figure shows the structure of a data block in the declaration view.





m	Fir	stProject → PLC_1 [C	PU 1214C A	C/DC/Rly]	Program	olocks ▶ Gl	obalDat	aBlock [DB	3]		
1	1	й 🔩 🛃 🧱 🧐 Ка	eep actual val	ues 🔒 S	napshot 🛤	tal, Copysr	apshots	to start value	s B. B.	Load start values as actual values 🛛 🗐	
	Glo	obalDataBlock									
		Name	Data type	Start value	Retain	Accessible f	Writa	Visible in	Setpoint	Comment	
1	-	▼ Static									
2	-0	resultVolume	Real	0.0						Current volume from Calculate block	
3	-0	value1	Dint	0						Value 1	
4	-0	value2	Dint	0						Value 2	
5	-	resultCalculation	Dint	0						Result	
6		Add new>									

Picture 5 Structure of a data block

The meaning of the individual columns is shown in the following table:

Column	Meaning
Name	Name of the variable
Data type	Data type of the variables
Start value	Value that the variable should assume during PLC start-up
Remanence	Marks the variables as retentive.
Accessible from	Indicates whether these variables can be accessed at runtime by
HMI/OPC UA	HMI/OPC UA.
Writable from	Indicates whether the variable
HMI/OPC UA	can be written to by HMI/OPC UA at runtime
Visible in	Indicates whether the variable is visible in the operand selection of
HMI Engineering	the HMI by default.
Setting value	Values that will probably need to be fine-tuned during
	commissioning. After commissioning, the actual values can be
	adopted as start values in .
Comment	Commentary on variables

Picture 6 Columns in the declaration view



6.4 Instance data block

Instance data modules are data modules that are automatically generated when a system function module or a function module is called.



Picture 7 Instance data block

The data of the module is saved in these instance data modules. They are assigned directly to the function block. A new instance data block must be created each time a function block is called.

The instance blocks created are stored under System blocks → Program resources or Program blocks in general.

The data structure depends on the programming of the function block used. The following figure shows an instance data module of the system function module of a counter.

		Ins	t_I	EC_Counter			
%DB8			Na	me	Data type	Start value	Monitor value
"Inst_IEC_	1	-0	•	Static			
Counter"	2	-		CU	Bool	false	FALSE
CTUD	З	-0		CD	Bool	false	FALSE
	4	-01		R	Bool	false	FALSE
CD	5	-01		LD	Bool	false	FALSE
R QD	6	-0		QU	Bool	false	TRUE
LD CV	7	-0		QD	Bool	false	FALSE
- PV QU-	8	-0		PV	Int	0	6
	9	-00		CV	Int	0	8

Picture 8 Instance data block of an IEC meter

Instance data blocks can also be created manually using the "Add new block
→ data block" function and selecting the corresponding function block.





Variables of an instance DB can be accessed both read and write by FC, FB and OB. The following figure shows access to QU and QD of a meter.



Picture 9 Access to instance data

Read-only access is recommended, **not** write access. The program can become confusing, especially with write access, as no cross-references are displayed.



6.5 Testing data blocks

To test the values of a data block, you can monitor and control the variables via an existing online connection directly in the open data block or via an observation table.

6.5.1 Monitoring in the data module

To do this, activate the "Monitor all" button in the function bar of the declaration table. A new column "Observation value" is added to the declaration table, in which the current actual value can be observed.

Button: "Watch all"

All variables are displayed with the current value of the PLC as an "observed value" and are constantly updated.



Picture 10 Data block in the observation function





6.5.2 Control operands in the data block

In the data module, you have the option of changing the observation value by right-clicking on a variable with the "Control operand..." function.

In the following "Control" window, enter the control value. Click on "OK" to adopt this as the new observation value; the start value remains unchanged.

* 🔹 📲	h 🛃 🗮 😤	Keep actual	values 🔒 Sna	ipshot 🛰 🔍	Copy snapshots to	start values 🛛 🖳	Load start valu	ues as actual values 🛛 🔒
Test_D	DB							
Nan	ne	Data type	Start value	Monitor value	Retain	Accessible f Writa	Visible in S	Setpoint Comment
	Static							
	variable1	Bool	🔳 false	FALSE	Modify operand	Ctrl+Shift+7		variable1
	variable2	Bool	false	FALSE	mouny operation	curtorinet2		variable2
	variable3	Bool	false	FALSE	insert row	Ctrl+Enter		variable3
	variable4	Bool	false	FALSE	Add row	Alt+Ins		variable4
	<add new=""></add>				X Cut	Ctrl+>		
					Сору	Ctrl+C		
					📋 Paste	Ctrl+V		
Example					Y Delete	De		
Mod	ify					× F2		
Oper	and:	DR" variable1	D	ata type:	ool			
open	lesci	DD .vanabier		ete type.				
Modif	fy value: true			3 t: B	ool	•		
						-Shift+G		
And the second se						Chife, D		

Picture 11 Control value data block

It is also possible to control binary variables in this way. Boolean variables can also be switched directly by double-clicking on their observation value.

Pr	ogr	am l	blocks 🕨 Test	_DB [DB9]								
1	Te	st [L ₽ 🖿 🎬	Keep actual va	lues 🔒 Sn	apshot 🙀 🖏	Copy snapshots to	o start values	B- B-	Load start v	alues as act	tual values 🛛 🛛 🕮
_		Nan	ne	Data type	Start value	Monitor value	Retain	Accessible f	Writa	Visible in	Setpoint	Comment
1	-	•	Static									
2	-0		variable1	Bool 🔳	false	FALSE						variable1
3	-0		variable2	Bool	false	FALSE						variable2
4	-00	•	variable3	Bool	false	FALSE						variable3
5	-01		variable4	Bool	false	FALSE						variable4
6			<add new=""></add>									
						ggle value (060 Toggle Do you v	D1:000045) Value vant to toggle the v ot show this messa	value of the tag: ge again. 'es 2	×			

Picture 12 Switch data block boolean variable



6.5.3 Observing in an observation table

You can enter and monitor the variables of the data block in an observation table. From here, you can also change the value of the variable via the control value.

Watch a	ind force tables 🕨 Watch	table_1					
		000 000					
* *	II 19 10 71 70 3	Address	Dicolay format	Monitorvalue	Modificualue	4	Comment
1	"Test_DB".variable1	Address	Bool	TRUE	TRUE		comment
2	"Test_DB",variable2		Bool	FALSE			
3	"Test_DB".variable3		Bool	FALSE		2	
4	"Test_DB".variable4		Bool	FALSE			
5		<add new=""></add>					

Picture 13 Observation table

You can open a data block, copy the desired variables

from the Windows clipboard and paste them into the observation table.



6.6 Exercise: Declaring workpiece management

Target:

I can declare variables in a global data block.

Task:

Create a global data block that serves as workpiece management so that the statuses from the system description can be mapped in the controller. The variables are described and evaluated at a later stage from the function blocks of the individual system components.

Fe	Fertigungslinie 24V > -KF1 [CPU 1214C DC/DC/Rly] > Program blocks > Workpieces [DB13]										
10	1	2 1	🔩 🛃 🛅 😚 Keep act	ual values 🔒	Snapshot 🚔 🖳 Copysnapshots to start values 🕵 🅵						
	Workpieces										
		Na	me	Data type	Comment						
1	-	-	Static								
2	-		magazinePart	Bool 🔳	Nest in position Magazine is occupied with a workpiece						
3	-		drillingRawPart	Bool	Nest in drilling position is occupied by a non-drilled workpiece						
4	💷 🔹 drillingFinishedPart			Bool	Nest in drilling position is occupied by a drilled workpiece						
5	-	weldingRawPart		Bool	Nest in welding position is occupied by a non-welded workpiece						
6	-	weldingFinishedPart Bool			Nest in welding position is occupied by a non-welded workpiece						
7	🛛 💶 transferPart Bool				Nest in position transfer is occupied by a finished part						

Picture 14 Example of workpiece management



Procedure:

1. Create a new global data block and assign a meaningful name:



2. Declare the variables:

Project Edit View Insert Online Options Tools	Window Help	a cooffine 🎒 🖪 🖪 🗙 🚽 💷 Gearch in projects 🔒
Project tree	Fertigungslinie 24V + -KF1 [CPU	1214C DC/DC/Rly] > Program blocks > Workpieces [DB13]
Devices		
1 I I I I I I I I I I I I I I I I I I I	💣 🔮 🐛 🛃 🗮 😋 Keep actu	al values 🍓 Snapshot 🍬 🧠 Copysnapshots to start values 😹 🛃 Load :
2	Workpieces	
🔽 🔽 Fertigungslinie 24V	Name	Data type Comment
Add new device	1 📲 💌 Static	
Devices & networks	2 📶 = magazinePart	Bool III Nest in position Magazine is occupied with a workpiece
KF1 [CPU 1214C DC/DC/Rly]	3 📶 = drillingRawPart	Bool Nest in drilling position is occupied by a non-drilled workpiece
Device configuration	4 📲 🔹 drillingFinishedPart	Bool Nest in drilling position is occupied by a drilled workpiece
🖞 Online & diagnostics	5 📶 = weldingRawPart	Bool Nest in welding position is occupied by a non-welded workpiece
Program blocks	6 📶 🔹 weldingFinishedPart	Bool Nest in welding position is occupied by a non-welded workpiece
Add new block	7 📲 🔹 transferPart	Bool Nest in position transfer is occupied by a finished part
- Main [OB1]		
Uvorkpieces [DB13]		
Technology objects		



Data blocks - Exercise: Declaring workpiece management



Solution



Solution:

The solution can be found in the TIA Portal project "Fertigungslinie_01_Werstueckverw.zap17".

