



# **Index of contents**

Introduction	2
Read and Write mode	3
The Control Byte	3
Dataset selection	3
Survey of all codes	4
Example recipe management	5
Save a recipe in the database	7
Read recipe from database	8
Delete recipe	Э
Display data block10	C
WhereClause data block1	2
Ring buffer mode14	4
QDAS mode	5
Contact	5

# Introduction

**HS**DBASE was created to log data from S7 PLCs easily for every user. The simple applicability was in the focus of attention throughout development.

For fast familiarization a popup help is integrated. It explains every button directly while using HSDBASE.

All general information for the data logging modes (*Cycle, Trigger, Change, Analyzer*) are already included in the popup help. Therefore only the *Read and Write, Ring buffer* and *QDAS* mode is explained in the following.

Additional information is available on our website under FAQ and **HS**DBASE examples.

# **Read and Write mode**

#### Advice:

This mode shouldn't be used for data logging, but for the exchange of data in both directions.

For data logging, especially fast loggings, you should use one of the other modes. Clearly higher data rates can be achieved with them.

## Control Byte

All read and write jobs are activated through the PLC via a Control Byte.

Should data be written or read, the PLC has to write the matching commando into the Control Byte. **HS**DBASE reads the Control Byte cyclic according to the adjusted time. If the value is bigger than 100, **HS**DBASE completes this commando and writes an answering code (smaller than 100) as confirmation into the Control Byte afterwards.

### Dataset selection

Inside the variable table of **HS**DBASE, there is a column named *change value*. Only the selected values in this column are overwritten in the PLC. The unchecked variables are used for the selection of the datasets.

To select the dataset, that should be read (respectively actualized), the indentifier variable has to be created under the variables in **HS**DBASE and the column *change value* has to be **unchecked**. Datasets can be selected also through the combination of several variables. Before every read/write operation the values are read out of the PLC. Then only the datasets which have the same value (in the selection variables) in the PLC and the database are handled.

If you select several datasets, then all these datasets will be actualized / deleted. On a read command the values of the first selected dataset will be used.

Please have a look on the example in this guide for better understanding.

## Survey of all codes

	Command codes
101	from database to PLC (dataset remains) - e.g. load recipe
102	from database to PLC (dataset will be deleted) - e.g. load batch
103	query if dataset present - e.g. request before overwriting
111	from PLC to database (overwrite, if dataset exists) - e.g. save recipe
112	from PLC to database (create doubly, if dataset exists) - e.g. create batch
113	delete dataset - e.g. delete recipe or batch
121	fill DisplayDB with available identifiers of recipes/batches
	Answering codes
1	data written to PLC
2	data written to PLC and dataset deleted
3	dataset created
4	dataset updated
5	dataset deleted
6	dataset present
7	DisplayDB was filled
	Fault codes
11	unknown command
12	error dataset does not exist
13	error reading database/file
14	error database/table/file not found
15	error headline does not exist
16	error wrong number of columns (headline)
17	error wrong number of columns (data line)
18	error writing database/table/file
19	error converting/dataset was not written complete
20	DisplayDB does not exist
21	DisplayDB too small
22	checkbox DisplayDB is unchecked
23	WhereClauseDB does not exist
24	WhereClauseDB too small

## Example recipe management

The following example shows a simple recipe with 3 parameters. This example can be reproduced very easily and it is relatively simple to test all commands.

Please create the following data block.

HS	DB/	AS	E > SIM_PLC [CPU 1515	5-2 PN] 🕨 Program	blocks 🕨	Recipe [DB2]								_ ∎∎×
1	1	è	🐛 🅪 는 😤 Keep ac	tual values 🔒 Sn	apshot 🛤	🛉 🖳 Copy snap	shots to star	tvalues 🖳 🛛	Loa	d start value:	as actual v	alues 其	Ð,	<b>_</b>
	Ree	cip	be											
		N	ame	Data type	Offset	Start value	Retain	Accessible f	Writa	Visible in	Setpoint	Supervis	Comment	
1	-	•	Static											
2	-		Controlbyte	Byte 🔳	0.0	16#0		<b></b>						
3	-		RecipeName	String[10]	2.0			<b></b>	<b></b>					
4	-00		Parameter1	Bool	14.0	false		<b></b>	$\checkmark$					
5	-		Parameter2	Int	16.0	0		<b></b>						
6	-	-	Parameter3	Real	18.0	0.0		<b></b>	<b></b>	<b></b>				

Create a new table in **HS**DBASE. Set *Read and Write* as mode and set the address of the Control Byte.

	×
File Edit Extras Help	
□ 22 22 13 12 12 12 12 12 12 12 12 12 12 12 12 12	
RecipeExample     Settings Variables Status	
Recipe Mode: Read and Write V ID column	
Start data Stop data min cycle(ms): 500 ID name: Id	
Control byte address: DB2.DBB0 Control byte status: 0 DisplayDB	
Save type	
Save as: MySQL Vrite new table every: Once	<b>-</b>
Table name: Recipe	=
Server: localhost User-ID: root Task Micro	
Database: SIM_PLC Password: Connection	
alternative storage location	
Write new database/folder depending on variable 🗍 Write new table depending on variable 🗍	
2022-10-18 09:11:48.191: C:\Users\HS\Desktop\manual.hsd loaded	
2022-10-18 09:12:52.994: SIM_PLC: Connection established	

Please note that the checkmarks in the column *change value* are placed correctly. Uncheck the recipe name (identifier variable), but check the parameters because they are overwritten by **HS**DBASE according to the database entry.

HS HS	DBASE	4.20 - ma	nual*											-		×
File	Edit	Extras	Help													
i 🗋 🖻	j 🛃	X h	11. I 🤊	er i Be	<b>1</b>											
	Recipe	Example			Settings	Variables 9	itatus									
	SIM_	PLC Recipe				Name	Address	Datatype	Comment	Arrayelements	Display	Change value	Database index	s	tatus	
						Timestamp	INTERN	DateTime(Intern)		1	DateTime(msec)			2022-10-1	8 09:13:2	3.567
						RecipeName	DB2.DBB2	String		10	Char					
						Parameter1	DB2.DBX14.0	Bool		1	Bin		•	0		
						Parameter2	DB2.DBW16	Int		1	Dec			0		
						Parameter3	DB2.DBD18	Real		1	Dec			0		
					<b>F</b> #											
2022 10	10.00.1	1.40 101	<u>College</u>													
2022-10	-18 09:1	2:52.994:	C:\Users\ SIM_PLC	(HS \Deskti : Connectio	op vnanu on establi	al hsd loaded shed										

Start the capture now, so that **HS**DBASE can react to the commands of the PLC.

#### Save a recipe in the database

Set random values for recipe name and parameters. Modify the Control Byte to 111 (From PLC to data base).

HSDBA	SE → SIM_PLC [CPU 15	15-2 PN] 🕨 Progran	n blocks	Recipe [DB2	2]		_ II II ×
9 9	🔩 🋃 🗮 🎇 Keep	actual values 🔒 Si	napshot 🏻	🐴 🖳 Copy sn	apshots to start values	🗽 🈹 🛛 Load start values as actual values 🛛 🜉 🔀,	
Rec	pe					Modify X	
	lame	Data type	Offset	Start value	Monitor value		
1 📲	<ul> <li>Static</li> </ul>					Operand: "Recipe".Controlbyte / %DB2.DBB0 Data type: Byte	
2 🕣	Controlbyte	Byte	0.0	16#0	16#00	Modify value: 111 Format: DEC	
3 📲	RecipeName	String[10]	2.0		'Test01'		
4 🕣	Parameter1	Bool	14.0	false	FALSE		
5 📲	Parameter2	Int	16.0	0	43	OK Cancel	
6 🕣	Parameter3	Real	18.0	0.0	3.7		

#### As answering code **HS**DBASE writes 3 (dataset was created) into the Control Byte.

HSDBASE > SIM_PLC [CPU 1515-2 PN] > Program blocks > Recipe [DB2]												
🞐 🐏 🍓 😸 🗮 🕎 Keep actual values 🔒 Snapshot 🦄 🧠 Copy snapshots to start values 🔹 Load start values as actual values 🔹 🚳												
Recipe												
Name	Data type	Offset	Start value	Monitor value	Retain	Accessible f	Writa	Visible in	Setpoint	Supervis	Comment	
1 🕣 🔻 Static												
2 🕣 🔹 Controlbyte	Byte	0.0	16#0	16#03		<b></b>						
3 🕣 = RecipeName	String[10]	2.0		'Test01'		<b></b>		<b></b>				
4 🕣 = Parameter1	Bool	14.0	false	FALSE		<b></b>						
5 📲 Parameter2	Int	16.0	0	43		<b></b>		<b></b>				
6 🕣 🔹 Parameter3	Real	18.0	0.0	3.7		<b></b>						

#### Now the recipe is in the database. Database and table have been created automatically by **HS**DBASE.

🐵 Un	nnamed\sim_plc\recipe\ - HeidiSQL 11.3.0.629	95				_		×
File E	Edit Search Query Tools Goto Help							
	- 💉   隆 🗎 🥱 🖶   🕘 - 💶 📓	<b></b>	) 🗙 🕨 – 📁	- 🖱 🛄 🔍 💭 🏹	4 100 - 49 ;	×	Donat	e
12 🛨	🚦 Host: 127.0.0.1 📄 Database: sim_plc	: 👖 Table: recipe 🔠 Data	a 🕨 Query 🐻					
` ^	sim_plc.recipe: 1 rows total (approximately)		≫ Next	🗢 Show all	▼ Sorting	Columns (6/6)	▼ Filter	
	Id 💡 Timestamp 🥊 R	RecipeName 💡 Paramet	er1 💡 Parame	eter2 💡 Parameter3	3 📍			
	1 2022-10-18 09:18:53.240	Test01	0	43	3,7			
~								
$\mathbf{O}$	× Filter: Regular expression							
33	<pre>SELECT * FROM `sim_plc`.`recipe`</pre>	LIMIT 1000;						\$
		🕓 Connected: 00 🛹 Maria	DB 10.4.22	Uptime: 5 days, 10:57 h	Server time:	09 🔿 Idle.		

This step can be repeated as often as needed with different recipe numbers and values. If you apply an already used recipe number, the dataset will be actualized.

🛞 U	nnam	ed\sim_	plc\recipe\ - HeidiSQL 11.3.0.	6295				-	- 🗆	×
File	Edit	Search	Query Tools Go to He	lp						
	<b>-</b> 💰	¥  📭	📋 匀 🚔 🛛 🗸 😫		🗘 😣 🕗 🗙 🕨	🗀 - 💾 📖	Q 💭 🥪 🔥 🗔 ⇒	⁄ ; 🛛	Donat	te
1		Host: 1	127.0.0.1 📃 Database: sim_j	olc 📑 Table: recipe	🔠 Data 🕨 Qu	iery 🖪				
^	sim	_plc.rec	ipe: 4 rows total (approximate	ly)	<b>&gt;</b> N	ext 🔶 Sho	ow all 🛛 🔷 🕶 Sorting	🔷 Columns (6/6)	▼ Filter	
	ld	9	Timestamp 💡	RecipeName 💡	Parameter1 💡	Parameter2 💡	Parameter3 💡			
		1	2022-10-18 09:18:53.240	Test01	0	) 43	3,7			
		2	2022-10-18 09:24:27.715	Test02	1	47	2,8			
		3	2022-10-18 09:25:07.727	Test03	1	53	-1,7			
		4	2022-10-18 09:25:56.723	Test04	0	) 22	. 1,2			
- v				-						
$\mathbf{O}$	×	Filter:	Regular expression	]						
47	SE	LECT *	FROM `sim_plc`.`recip	≥` LIMIT 1000;						\$
			r1:c1	Connected: 00	🛹 MariaDB 10.4.22	2 Uptime: 5 d	lays, 11:00 h 🛛 🕓 Serve	er time: 09 🔘 Idle.		

### Read recipe from database

Now the recipe 'Test01' should be transferred back to the PLC. To do this, the recipe name has to be set first. Then modify the Control Byte to 101 (from data base to PLC).

HS	DBAS	E → SIM_PLC [CPU 15	15-2 PN] 🕨 Progr	am	blocks 🕨	Recipe [DB2]							_ ⊫ ■ ×
2	<b>e</b>	🔩 🛃 🗮 🔭 Keep	actual values 🏻 🔒	Sna	pshot 🏘	🛉 📖 Copy sna	pshots to start values	<b>B</b> - 6	🕵 Load start	values as actual values 🏾 💐 🕮			
	Recip	e							Modify				X
	Na	ame	Data type		Offset	Start value	Monitor value	Retai	r in the second s		_		
1		Static							Operand:	*Recipe*.Controlbyte / %DB2.DBB	Data type:	Byte	
2		Controlbyte	Byte		0.0	16#0	16#03		Modify value:	101	Format:	DEC	<b>T</b>
з		RecipeName	String[10]		2.0		'Test01'	1				bee	-
4		Parameter1	Bool		14.0	false	FALSE						
5		Parameter2	Int		16.0	0	22	1	5			OK Cancel	-
6		Parameter3	Real		18.0	0.0	1.2						

The parameters have been overwritten and the answering code 1 was set (data was written in PLC).

н	HSDBASE → SIM_PLC [CPU 1515-2 PN] → Program blocks → Recipe [DB2]														
ji ji	2	🕴 🐛 🅪 🚞 🍄 Keep	actual values 🛛 🔒 S	napshot	🐴 🖳 Copy sr	napshots to start values	R. R. L	oad start value.	s as actu	al values 📃	, B),				
	Re	cipe													
		Name	Data type	Offset	Start value	Monitor value	Retain	Accessible f	Writa	Visible in	Setpoint	Supervis	Comment		
1	-00	▼ Static													
2	-0	Controlbyte	Byte	0.0	16#0	16#01		<b></b>							
з	-0	RecipeName	String[10]	2.0		'Test01'		<b></b>							
4	-00	Parameter1	Bool	14.0	false	FALSE		<b></b>							
5	-00	Parameter2	Int	16.0	0	43		<b></b>							
6	-0	Parameter3	Real	18.0	0.0	3.7		<b></b>							

### **Delete recipe**

Now the recipe 'Test01' should be deleted. To do this, the Control Byte has to be modified to 113 (delete dataset).

HSDBASE	DBASE → SIM_PLC [CPU 1515-2 PN] → Program blocks → Recipe [DB2]												
🥩 🔮 🖣	🐛 🋃 🗮 🏋 Keep ad	tual values 🛛 🔒 Sn	apshot 🕷	🛉 🖳 Copysnap	oshots to start values	<b>R</b> - <b>R</b> -	Load start va	luer ar actu	al valuer 🗧	. 8.		X	
Recipe	e						моатту					~	
Nar	me	Data type	Offset	Start value	Monitor value	Retain	Operand:	"Recipe".Co	ontrolbyte / %	DB2.DBB0	Data type:	Byte	
1 🕣 🔻	Static						Modifyvalue:	113			Format:	DEC	
2 📲 🖷	Controlbyte	Byte 🔳	0.0	16#0	16#01		incomy rande.						
3 🕣 🗉	RecipeName	String[10]	2.0		'Test01'								
4 📲 🔳	Parameter1	Bool	14.0	false	FALSE	<b></b>						OK Cancel	
5 📲 🖷	Parameter2	Int	16.0	0	43	<b></b>							
6 📲	Parameter3	Real	18.0	0.0	3.7	<b></b>			<b></b>				

#### The answering code is 5 (dataset was deleted).

HSDBA	SE 🔸 SIM_PLC [CPU 151	5-2 PN] 🕨 Program											_ ⊫∎×
2	🐛 🅪 🚞 😤 Keep a	ctual values 🛛 🔒 Sn	apshot 🕷	🛉 🖳 Copysnap	oshots to start values	🖳 🖳 La	ad start values	s as actu	al values 📃	, B),			-
Reci	pe												
- N	ame	Data type	Offset	Start value	Monitor value	Retain	Accessible f	Writa	Visible in	Setpoint	Supervis	Comment	
1 🐽 י	Static												
2 📲 🗉	Controlbyte	Byte 📳	0.0	16#0	16#05								
з 🐽 🕯	RecipeName	String[10]	2.0		'Test01'	Image: A state of the state							
4 🕣 🛚	Parameter1	Bool	14.0	false	FALSE	<b></b>							
5 📲 🖲	Parameter2	Int	16.0	0	43	<b></b>							
6 🕣 🗉	Parameter3	Real	18.0	0.0	3.7								

#### Now the database table looks as follows:

HS	Unn	ame	d\sim_	plc\recipe\ - I	HeidiSQL 11.3.0.0	5295									-		×
File	Ec	lit :	Search	Query To	ols Goto He	lp											
	¥ -	1		🗎 👈 🚍	🕘 👻 👥	i <b>∷   0  </b> ∢	M (	🖯 😣 🕗 🗙	. 🕨	- 🗀 - 💾 .		Q 📿 🈼 ⊿		<i>⊷</i> , , ×		Donat	e
17 🖌	7		Host: i	127.0.0.1	Database: sim_p	olc 📑 Table: re	cipe	🔠 Data 🕨	Que	ery 🗔							
'	^	sim_	olc.rec	ipe: 3 rows tot	al (approximate	y)		X	Ne)	ct 🔶	Shov	wall 🔤	<ul> <li>Sorting</li> </ul>	Columns	(6/6)	Filter	
		ld	9	Timestamp	9	RecipeName	9	Parameter1	9	Parameter2	9	Parameter3	9				
			2	2022-10-18 (	09:24:27.715	Test02			1		47		2,8				
			3	2022-10-18 (	09:25:07.727	Test03			1		53		-1,7				
11	11		4	2022-10-18 (	09:30:01.216	Test04			0		22		1,2				
$\sim$	×	×	Filter	Regular expr	ression												
48	В	SEL	ECT *	FROM `sim	_plc`.`recipe	:` LIMIT 1000	;										< >
					r1:c1	( Connecte	d: 00	利 MariaDB 10	.4.22	Uptime	: 5 da	ays, 11:09 h	🕓 Server	r time: 09 🔘 Idle.			

### Display data block

To read out which recipes are present in the database, a data block can be created. This data block will be filled by **HS**DBASE.

IS HSDBASE 4.20 - manual*							-		×
File Edit Extras Help									
i 🗅 📂 🖬 🗼 🖻 🛍 🔊 🕅 🐘	<b>≣</b>   <b>  ↓   ↓</b>   <b> </b>								
RecipeExample	Settings Variables State	us							
Recipe	Capture		Mode:	Read and Write	•		D column		
	Start data	Stop data	min cycle(ms):	500		D name: Id			
	capture	capture	Control byte address:	DB2.DBB0	Control byte status: 5		)isplayDB	10	
						Where Cl	ause DB:		
	Save type								
	Save as:	MySQL	•		Write new	table every:	Once		J
				т	able name: Recipe				
	Server:	localhost			User-ID: root				
	Database:	SIM PLC			Password:		— Cr	onnection	
		, -			,	alternativ	ve storage lo	cation 🗔	-
							-		
		Write new databas	se/folder depending on var	iable 🥅	Write	new table dep	ending on va	ariable 🥅	
2022-10-18 09:11:48.191: C:\Users\HS\Desktop 2022-10-18 09:12:52.994: SIM_PLC: Connection	p∖manual.hsd loaded n established								
_									

#### Structure of the display data block.

HS	DB	BASE > SIM_PLC [CPU 151	5-2 PN]  Program bloc	:ks ► Dis	splayDB [DB10]								_ !! ■ ×
1	1	🖗 🐛 🍢 는 😤 Keep ad	ctual values 🛛 🔒 Snapsho	ot <sup>ing</sup> <sup>ing</sup>	, Copy snapshots	to start valu	es 🖳 🖳	Load star	rt values as a	ctual values	H, H,		
	Di	splayDB											
		Name	Data type	Offset	Start value	Retain	Accessible f	Writa	Visible in	Setpoint	Supervis	Comment	
1	-	🔹 🔻 Static											
2		nr_of_datasets	Dint 🔳	0.0	0		<b></b>	<b></b>					
З	-	start_datatset	Dint	4.0	0	<b></b>	<b></b>	<b></b>					
4	-	a datasets_in_db	Dint	8.0	0	<b></b>	<b></b>	<b></b>					
5		🛛 = 🕨 recipes	Array[125] of String[10]	12.0			<b></b>	<b></b>					

The first three entries have to be type of DInt. Array length of the identifier could be any size, but the data type has to be the same type as the identifier.

When the Control Byte is modified to 121 (fill DisplayDB) **HS**DBASE writes the number of all datasets in the database into the first DINT. The third DINT contains the number of datasets, which have been written into the data block. The rest of the data block is filled up with the identifiers of the datasets. **HS**DBASE detects automatically how many identifiers match into the block. If more datasets exist in the database than space is available, the beginning of the display area can be adjusted with the second DINT. If there are less datasets than space, the remaining space will be filled with w#16#00. After modifying the Control Byte, the display data block shows all identifiers of the datasets in the database.

HS	DBA	SE	→ SIM_PLC [0	CPU 1515-2 PN] 🔸 Program bloc	ks 🕨 Displa	yDB [DB10]								_ ⊫ ≡ ×
1	9		h 🛃 🖿 🛸	Keep actual values 🛛 🔒 Snapsho	t 🖳 🖳 🤇	Copy snapshots t	o start values 🛛 🖳 🛛	Load start	values as actua	l values	<b>e</b> , 8,			=
	Dis	pla	yDB											
		Nan	ne	Data type	Offset	Start value	Monitor value	Retain	Accessible f	Writa	Visible in	Setpoint	Supe Comment	
1	-	•	Static											^
2	-	•	nr_of_datase	ts Dint	0.0	0	3		<b></b>	<b></b>				=
3	-	•	start_datatse	t Dint	4.0	0	0			<b></b>				_
4	-	•	datasets_in_	db Dint	8.0	0	3				<b></b>			
5	-	•	<ul> <li>recipes</li> </ul>	Array[125] of String[10]	12.0				<b></b>					
6	-		<ul> <li>recipes[1]</li> </ul>	String[10]	12.0		'Test02'		<b>V</b>	<ul> <li>Image: A start of the start of</li></ul>				
7	-		<ul> <li>recipes[2]</li> </ul>	String[10]	24.0		'Test03'		<b>~</b>	<ul> <li>Image: A start of the start of</li></ul>				
8	-		<ul> <li>recipes[3]</li> </ul>	String[10]	36.0		'Test04'		<b>V</b>	<b>V</b>	<b>V</b>			
9	-		<ul> <li>recipes[4]</li> </ul>	String[10]	48.0				<b>V</b>	<b>V</b>	<b>V</b>			~

### WhereClause data block

In order to create the possibility to select from a database only datasets, which fulfill certain conditions, there is in **HS**DBASE the possibility to enter conditions in a further DB, which are appended to the SQL query inside **HS**DBASE in a "Where Clause".

K HSDBASE 4.20 - manual*								-		×
File Edit Extras Help										
i 🗅 📂 🖬 i 🕹 🖻 🛍 i 🤊 🕫 i 🖍	<b>≣</b>     ↓   ↓   →									
RecipeExample     Box SIM PLC	Settings Variables Statu	us								
Recipe	Capture		М	ode: Read and Write	•	]		) column		
	Start data	Stop data	min cycle(	ns): 500		D	name: Id			
	capture	capture	Control byte addre	ess: DB2.DBB0	Control t	oyte status: 7		isplayDB	10	
							✓ Where Cla	ause DB:	11	
	Save type									
	Save as:	MySQL	•			Write new ta	able every:	Once		•
					Table name:	Recipe				
	Server:	localhost			User-ID:	root		-	Test MvSQI	
	Database:	SIM_PLC			Password:			_	Connection	
							alternativ	e storage	e location	
		Write new databas	e/folder depending or	variable 🕅		Write r	new table depe	anding on	variable 🗆	
2022-10-18 09:11:48.191: C:\Users\HS\Deskto 2022-10-18 09:12:52.994: SIM PLC: Connection	p∖manual.hsd loaded n established									

The WhereClause-DB consists of any number of strings. With these strings the internal SQL command in **HS**DBASE is extended behind the keyword WHERE. If the first string is empty, the query is executed without "WhereClause". When creating the strings, the corresponding SQL syntax of the selected database must be observed. A check does not occure.

Simple condition for the previous example:

H			► SIM_P													_ 🗉 🖬 🗙
3	¢ ₫	1	• 🛃 🗮	° Keep a	ctual values 🛛 🔒 Sna	apshot 🛤	Copysnap	shots to start values	🛃 🖳 Lo	ad start values	as actua	l values 🛛 📃	Ð,			
	Wh	ere	Clause (si	napshot crea	ated: 9/28/2022 2:5	3:56 PM)										
-		Nan	ne		Data type	Offset	Start value	Monitor value	Retain	Accessible f	Writa	Visible in	Setpoint	Supervis	Comment	
1	-	•	Static													^
2	-00	•	<ul> <li>Where</li> </ul>		Array[010] of String	0.0										=
з	-		When	re[0]	String	0.0		'Parameter1 = 1'			<ul> <li>Image: A start of the start of</li></ul>	<b>V</b>				
4	-		When	re[1]	String	256.0										~
	<															>

After modifying the control byte to 121 (fill display BD), the display DB is filled only with the data records that meet the set conditions.

H	DB	٩SI	E → S	SIM_PLC [CPU 151	5-2 PN] 🕨 Program	blocks )	DisplayDB [DB	310]								_ ⊫ ■ ×
ġ	1	6	🔍 🔜	🖌 🚞 🐨 Keep a	tual values 🔒 Sn	apshot 🛤	🛉 🖳 Copysnap	shots to start values	R- R- L	oad start values	as actu	al values 🛛 📃	, Bi,			<b></b>
	Dis	spla	ayDB													
		Na	me		Data type	Offset	Start value	Monitor value	Retain	Accessible f	Writa	Visible in	Setpoint	Supervis	Comment	
1	-	•	Static													^
2	-	•	nr	_of_datasets	Dint	0.0	0	2		<b></b>	<b></b>					=
З	-		sta	art_datatset	Dint	4.0	0	0			<b></b>	<b></b>				
4	-	•	da	atasets_in_db	Dint	8.0	0	2				<b></b>				
5	-	•	▼ re	cipes	Array[125] 🔳 💌	12.0				<b></b>	<b></b>					
6	-		•	recipes[1]	String[10]	12.0		'Test02'	<b></b>	<b>V</b>	<b>V</b>	<b>V</b>				
7	-		•	recipes[2]	String[10]	24.0		'Test03'	<b></b>		<ul> <li>Image: A start of the start of</li></ul>	<b>V</b>				
8			•	recipes[3]	String[10]	36.0		*		Image: A start and a start		<b>V</b>				~

# **Ring buffer mode**

In In this mode, data is collected in a FIFO buffer on the machine that runs *HS*DBASE. Data from this buffer is written to the database on command. The buffer is controlled by a Control Byte like in the Read and Write mode.

### Survey of all codes:

	Command codes
101	delete buffer
102	fill buffer (no acknowledgment)
103	save buffer
	Answering codes
1	buffer deleted
2	buffer saved ( and deleted)
11	unknown command

# QDAS mode

The following example shows how **HS**DBASE has to be configured to save the data as QDAS DFQ file. These files could be read by QDAS upload tool and transferred to the QDAS database automatically.

The QDAS K numbers have to be entered in the name column. A separate DFQ file with consecutive number will be created on every trigger event.

	Hei	p ••••••••••••••••••••••••••••••••••••		1.1.1.					
	Cattions	Variables	10 00 11 1 1 0 0 0						
PLC Bsp		Name	Address	Datatype	Comment	Arrayelements	Display	Check	Status
		K0100	DB1.DBW2	Int	Gesamtanzahl Merkmale in der Datei	1	Dec	~	2
		K1001	DB1.DBB4	String	Teilenummer	20	Char	~	00012345
		K1002	DB1.DBB26	String	Teilebezeichnung	20	Char	~	Musterteil
		K2001/1	DB1.DBW48	Int	Merkmalnummer	1	Dec	~	1
	•	K2002/1	DB1.DBB50	String	Merkmalbezeichnung	20	Char	~	Laenge
		K2142/1	DB1.DBB72	String	Einheit	20	Char	~	mm
		K2101/1	DB1.DBD94	Real	Nennmaß	1	Dec	<b>V</b>	10,5
		K2110/1	DB1.DBD98	Real	Untergrenze	1	Dec	<b>V</b>	9,5
		K2111/1	DB1.DBD102	Real	Obergrenze	1	Dec	<b>V</b>	11,5
		K2001/2	DB1.DBW106	Int	Merkmalnummer	1	Dec	<b>V</b>	2
		K2002/2	DB1.DBB108	String	Merkmalbezeichnung	20	Char	<b>V</b>	Breite
		K2142/2	DB1.DBB130	String	Einheit	20	Char	<b>V</b>	mm
		K2101/2	DB1.DBD152	Real	Nennmaß	1	Dec	<b>V</b>	5
		K2110/2	DB1.DBD156	Real	Untergrenze	1	Dec	<b>V</b>	4
		K2111/2	DB1.DBD160	Real	Obergrenze	1	Dec	<b>V</b>	6
		K0001/1	DB1.DBD172	Real	MesswertLänge	1	Dec	<b>V</b>	10,78
		K0004/1	DB1.DBB164	Date_And_Time	Zeit/Datum	1	DateTime QDAS		03.01.2014/12:3
		K0001/2	DB1.DBD176	Real	MesswertBreite	1	Dec	<b>V</b>	4,2
		K0004/2	DB1.DBB164	Date_And_Time	Zeit/Datum	1	DateTime QDAS	<b>V</b>	03.01.2014/12:3
	*								

# Contact

If there are any other questions, please do not hesitate to contact us at any time.

HS Automation Software Im Öderich 2 D-73663 Berglen Tel.: + 49 (0) 7195 - 588 639 E-Mail: <u>info@hs-automation-software.de</u> Web: <u>www.hs-automation-software.de</u>