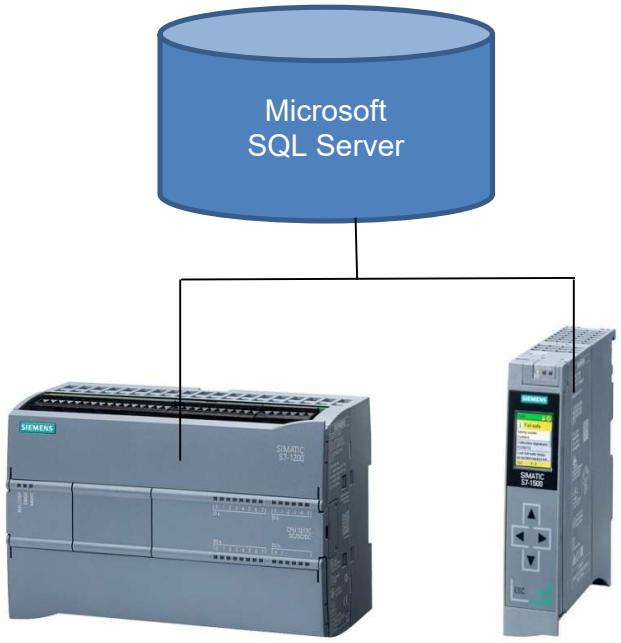
Installation Manual

PLCSQL link With Siemens TIA S7 1200 and S7 1500

SQL Client in a
Siemens S7 PLC



Revision date: 2018-09-13
Revised by: Anders Jorsal, Alsmatik A/S
Version: 1.30
Revision date: 2019-033-12
Revised by: FBH, Alsmatik A/S
Version: 1.40

Contents

Introduction

You can operate the PLCSQL link with Microsoft SQL Server
But to get it to work, it is important to setup the Database, to fit with the design for PLCSQL link.

You also need to follow tutorial “MS SQL Installation Manual_2014”.

If you still have questions after reading this manual, please send them to info@plcsql.com

Prerequisites

The user of the PLCSQL software must have good knowledge of using Tia Portal and the possibilities of “drag and drop” between different projects.

 Attention!

We have been exposed to problems regarding to integrate the PLCSQL Project into a User project. The problem that comes up, is that the “SQL_Client” (protected block) has to be compiled again before there can be downloaded to the PLC.

If the user project is integrated into the PLCSQL project, there is no problems!

Software requirements

This example project is based on following software tools:

PLC Program: Siemens Tia Portal V 14 SP1 upd 3

HMI: Siemens Tia Portal V 14 SP1 upd 3

In both cases the latest updates are required

If you need another version, please let us know, and we will convert it for you.

Hardware requirements

S7 1200 PLC with firmware **4.2**.

S7 1500 PLC with firmware **1.8** or **2.X**.

Open Controller with firmware **2.X**

How does the system work?

We have tried to make a system where it is possible to communicate with a SQL data base without being an “SQL expert”, nor being an “PLC expert” regarding communication etc. the system contains a “standard” PLC program, and a “standard” SQL data base that “fits” together.

In the PLC, we are using the basic tag types:

Bool.	Is stored in an “Bool” table in the data base.
Int.	Is stored in an “Int” table in the data base.
Dint.	Is stored in an “Dint” table in the data base.
Real	Is stored in an “Real” table in the data base.
String	Is stored in an “String” table in the data base.

To distinguish between the different tags, every tag has a specific number.
In the PLC, there is an “Array” that contains all the tags, and in the database the different tables contain the corresponding data types and numbers as in the PLC, so you have complete control with the tags.

On the following pages, there is a schematic view of the layout and the possibilities you have with the PLCSQL system.

Please note the following.

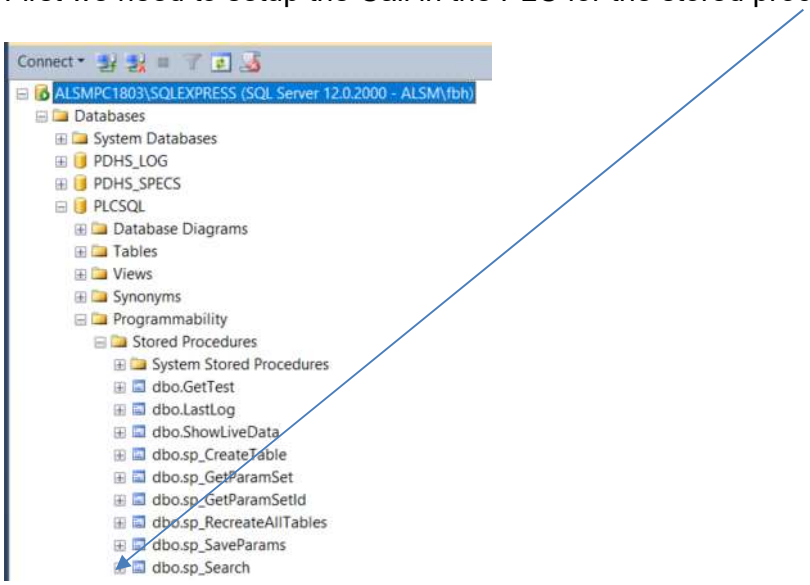
Parameter 10001, 15001, and 30001 is used internally in the “Log” parameters and in the “Recipe” parameters. DON’T write to these parameters. (can be changed if necessary)

10001:	SetCount	(Default setup)
15001:	SetID	(Default setup)
30001:	DateTimeStamp.	(Default setup)

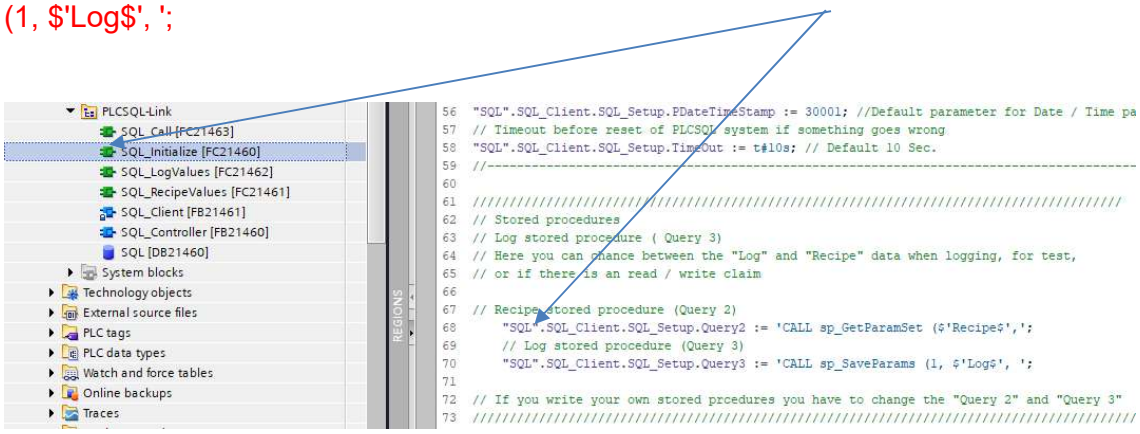
How does PLCSQL work, Log, write to SQL server

In Mssql we want to store the Value 2.3009 in the ParamID[1].

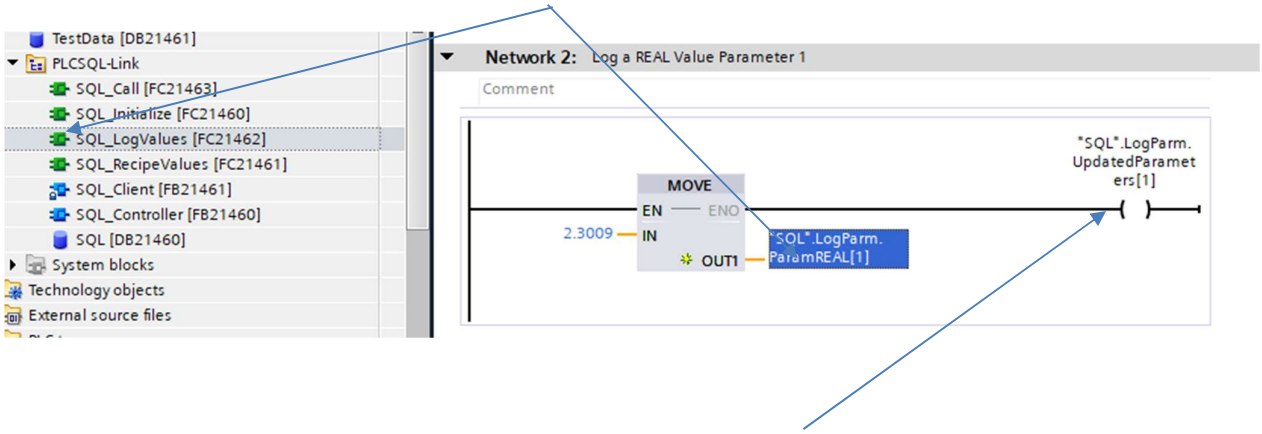
First we need to setup the Call in the PLC for the stored procedure in Mssql.



In SQL_Initialize DB you setup the SQL.SQL_Client.SQL_Setup.Query3 to match the procedure in Mssql: "SQL".SQL_Client.SQL_Setup.Query3 := 'EXEC sp_SaveParams (1, $'Log$', ';

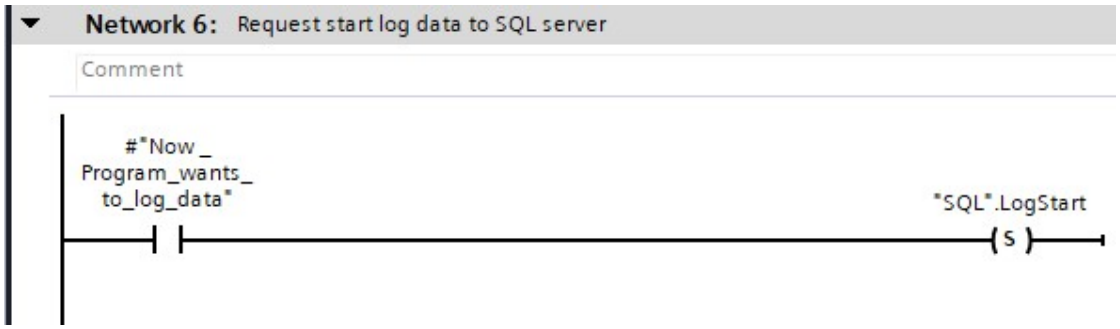


In the PLC we move the value 2.3009 to the SQL DB

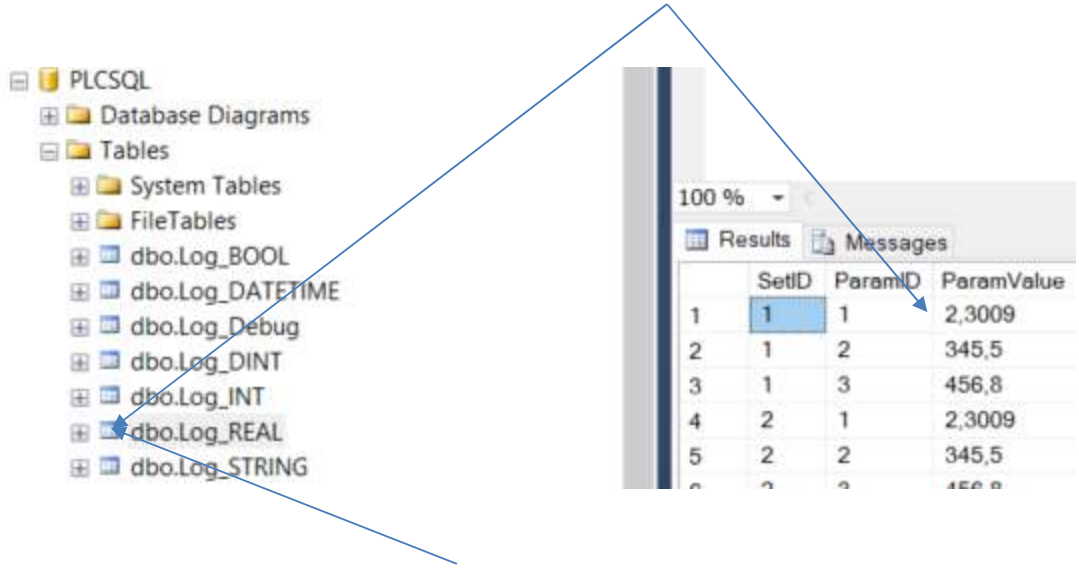


And we tell the system that there is an updated value on ParamID[1].

Last we set the bit SQL.LogStart and the value is stored in mssql.

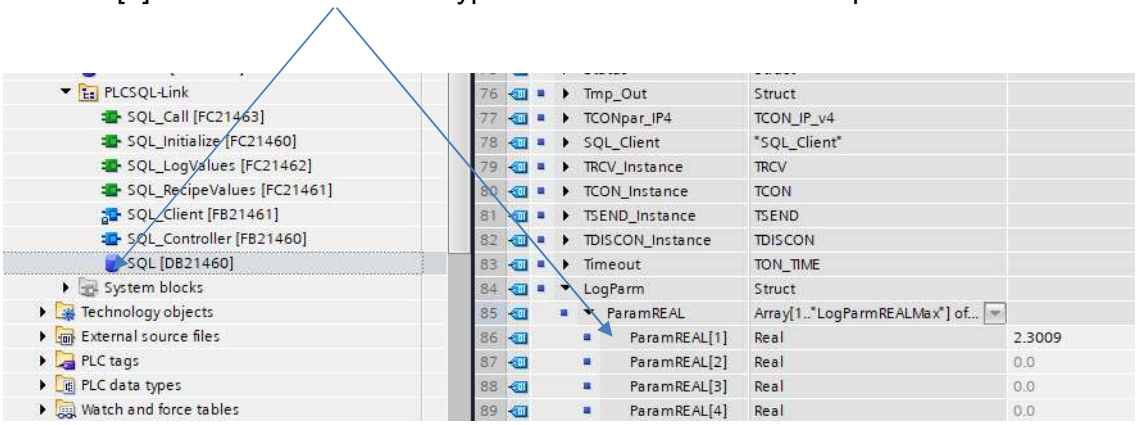


In Mssql we can see that the Value 2.3009 is stored in ParamID[1].



Right click on dbo.Log_REAL and select top 1000 rows to see the view.

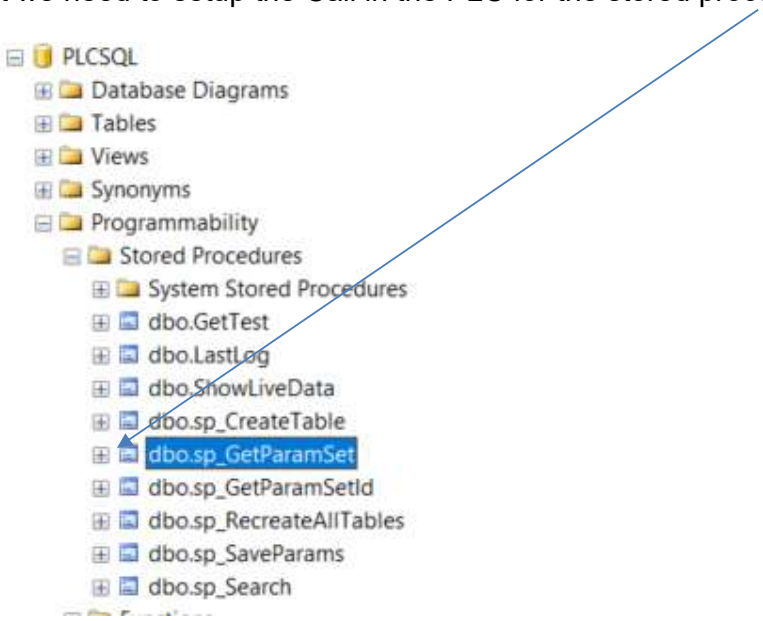
ParamId[1] is defined as a REAL type in both the PLC and Mssql.



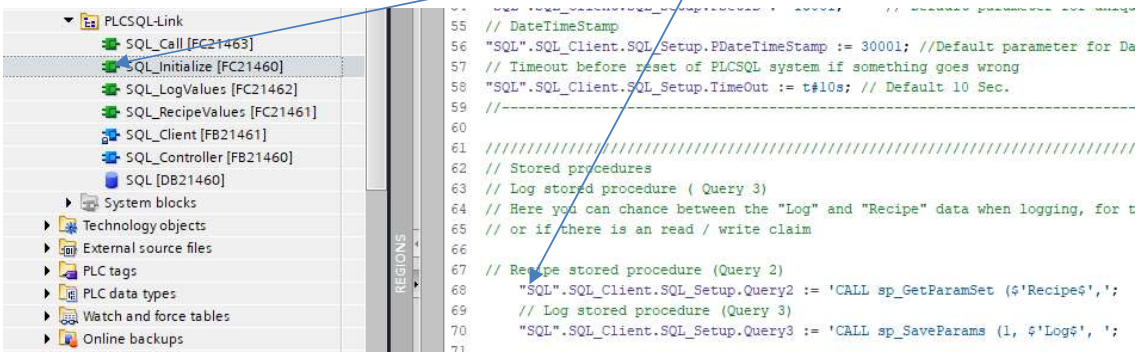
How does PLCSQL work, Recipe, read from SQL server

In Mssql we want to read the stored value in ParamID[3].

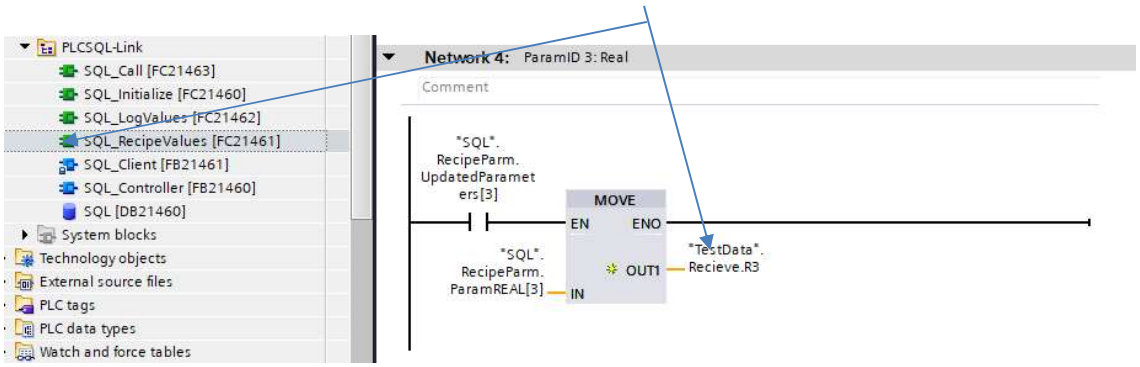
First we need to setup the Call in the PLC for the stored procedure in Mssql.



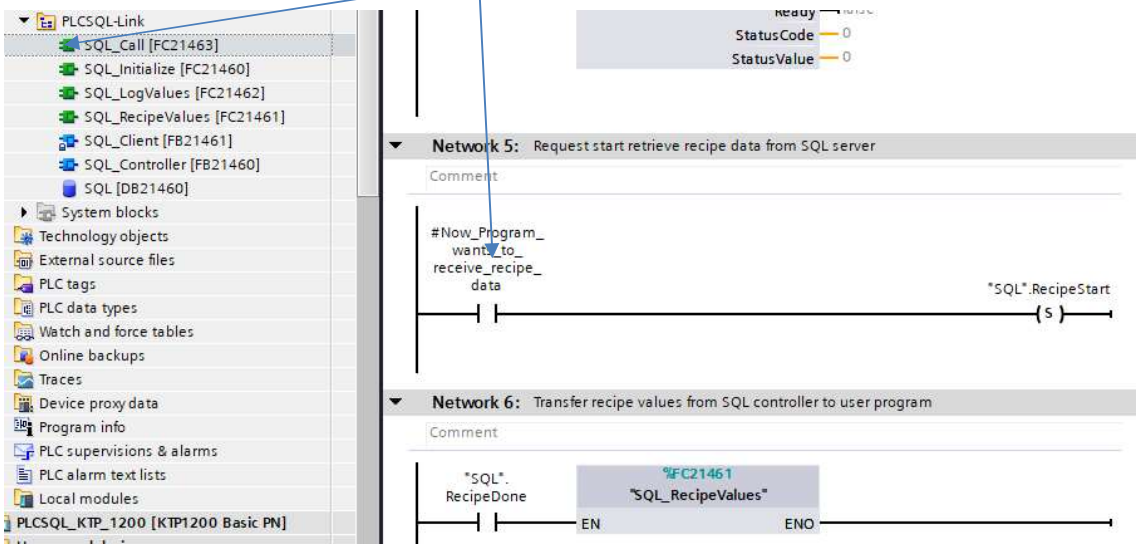
SQL_Initialize DB you setup the SQL.SQL_Client.SQL_Setup.Query2 to match the procedure in Mssql: "SQL".SQL_Client.SQL_Setup.Query2 := EXEC sp_GetParamSet;



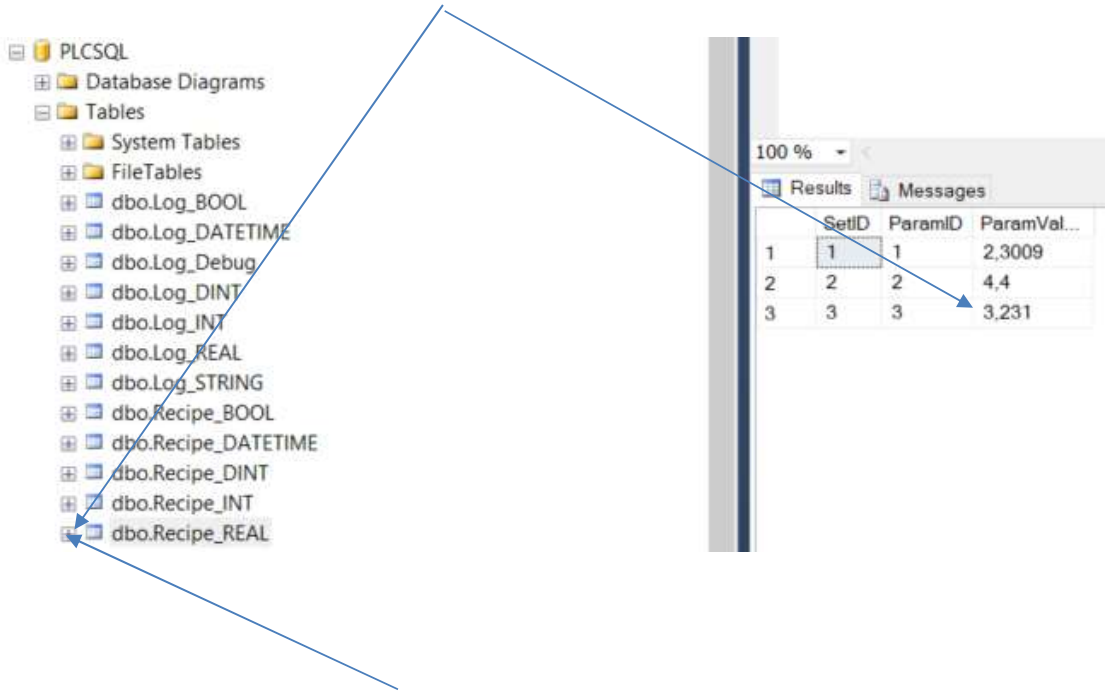
Move the received data from Mssql to your variable in the PLC.



In the Plc we need to trigger the start log bit.

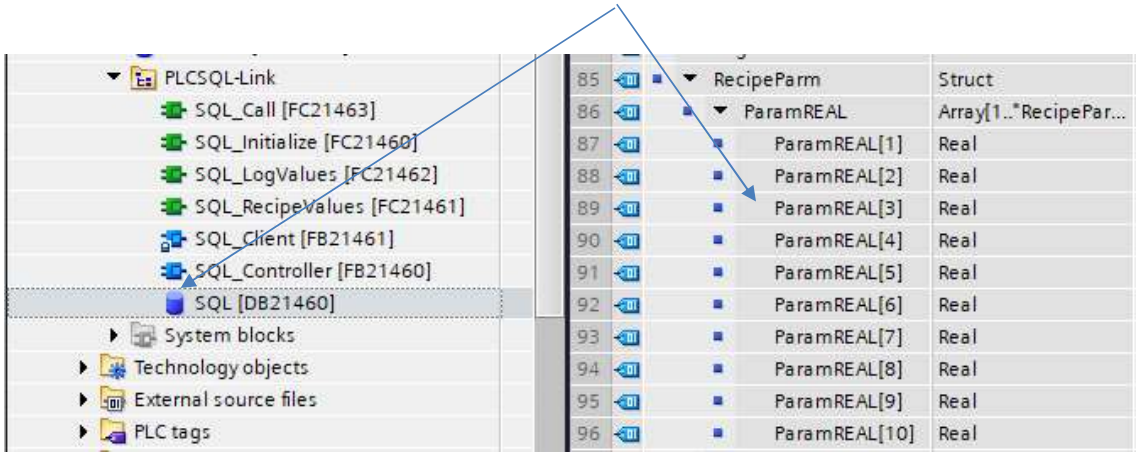


In Mssql we want to read a Real value with ParamID 3 from Recipe



Right click on dbo.Log_REAL and select Top 1000 rows to see the view.

ParamId[3] is defined as a REAL type in both the PLC and Mssql.



Used Blocks in this example project

The block numbers are the same for S7 1200 and S7 1500 PLC's but there are major differences in the code due to differences in the hardware of the 2 types of PLC's, so be careful to use the correct blocks.

In the example project, we are using the following blocks:
FC 21463 SQL_Call
FC 21460 SQL_Initialize
FC 21462 SQL_LogValues
FC 21461 SQL_RecipeValues
FB 21461 SQL_Client (Protected, cannot be renumbered)
FB 21460 SQL_Controller
DB 21460 SQL
DB 21461 Test_Data

Option Blocks, used to test the system, not mandatory, can be removed.
FB 22000 ControlTest
FB 22001 TestEnviroment
DB 22000 ControlTestDB
DB 22001 TestEnviromentDB

Be sure these blocks are free if you copy them into an existing project, or renumber the blocks.

Option HMI, KTP 1200 Basic
The Basic panel is chosen because you always have the possibility to run this type of panel, no matter what version of Tia Portal (Basic / Professional) you have, and you don't need the option software "WinCC".

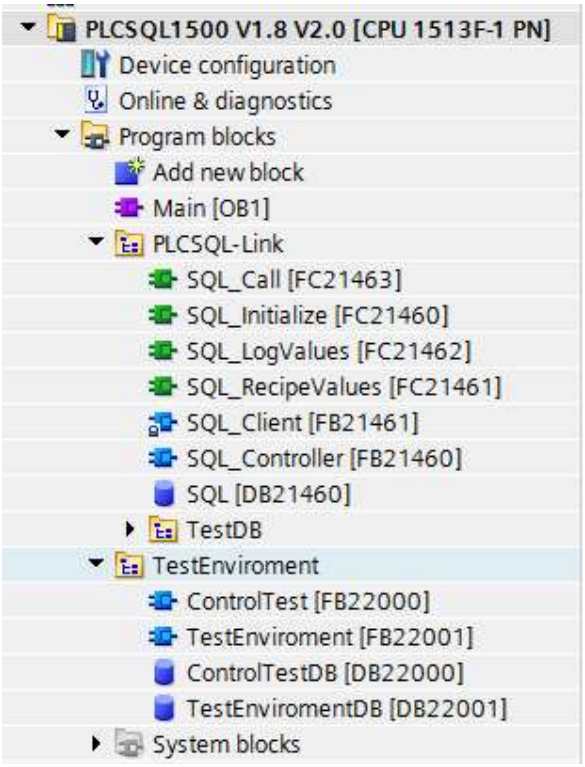
Setting up PLCSQL Link in Siemens TIA Portal

The PLCSQL software is supplied for the specific PLC type you are using (1200 / 1500 / Open Controller).
The software project is a “complete” project with PLC and HMI, if you only want the code blocks in a library, please let us know, then you will get a library.

When you opened the project, you must

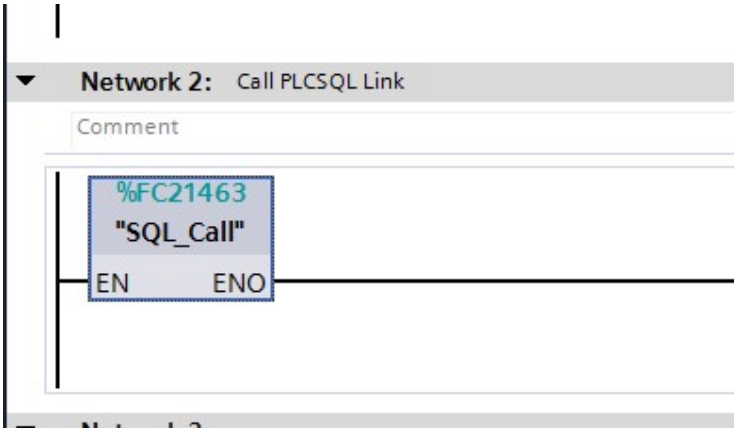
- Set up the hardware configuration to your needs
- Set IP address for the Ethernet port
- Compile it (rebuild all blocks)
- Download the configuration to the PLC.

Software layout



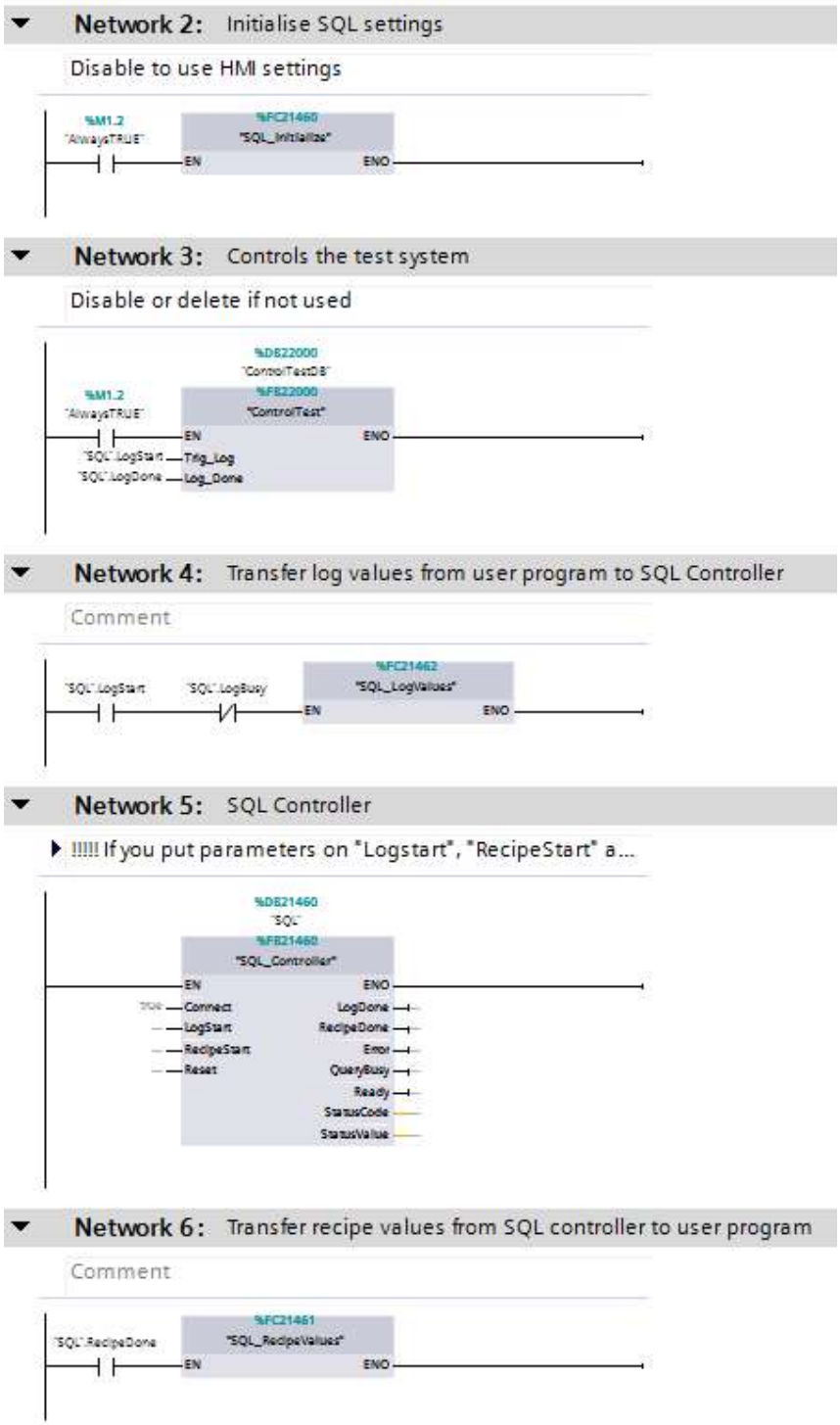
Overview of the software, the “TestEnviroment” folder can be deleted if you are familiar with the system.

Main OB1



This network controls the PLCSQL system.

SQL_Call FC 21463



Call structure of the PLCSQL Link system, if the order is changed, there is no warranty for correct function.

SQL_Initialize FC 21460

```
// Licens Key
//
"SQL".SQL_Client.SQL_Setup.Key1 := 16#027E; // 1 key, letters must be UPPERCASE
"SQL".SQL_Client.SQL_Setup.Key2 := 16#01D9; // 2 key, letters must be UPPERCASE
"SQL".SQL_Client.SQL_Setup.Key3 := 16#455C; // 3 key, letters must be UPPERCASE
"SQL".SQL_Client.SQL_Setup.Key4 := 16#1708; // 4 key, letters must be UPPERCASE
"SQL".SQL_Client.SQL_Setup.Key5 := 16#182C; // 5 key, letters must be UPPERCASE

// Connection
// IP Address Server
"SQL".SQL_Client.SQL_Setup.ServerIP[1] := 172; // Must be the same as in the PLC or router
"SQL".SQL_Client.SQL_Setup.ServerIP[2] := 20; // Must be the same as in the PLC or router
"SQL".SQL_Client.SQL_Setup.ServerIP[3] := 92; // Must be the same as in the PLC or router
"SQL".SQL_Client.SQL_Setup.ServerIP[4] := 100;// Range 1-255

```

“License Key”, here you type the license key that match the serial number of the CPU or the serial number of the used memory card.

“IP Address Server”, here you type the address of the SQL server.

Hardware setup S7 1200 /1500 / Open Controller PLC’s

```
// Port SQL Server
//
"SQL".TCONpar_IP4.RemotePort := 1433; //MS-SQL
// Port Local > 2000 or just = 0
//
"SQL".TCONpar_IP4.LocalPort := 2000;
// Device ID
//
"SQL".SQL_Client.SQL_Setup.DeviceID := 1;       // Logical connection number, must be unique
// Connection ID
//
"SQL".SQL_Client.SQL_Setup.InterfaceID := 64;  // Hardware Identifier of selected ethernet port  (64 = onboard interface 1)

```

“Port SQL Server”, here you type the port number of the SQL server.
“Port Local”, here you type the port number to use in the local PLC.

REMEMBER to restart the PLC if you change “Device ID” in RUN.
“Device ID”, here you typical type a “1”, if you want to call the “SQL” system multiple times, then this number must be unique for every instance.

“Interface ID”, here you type the “Hardware ID” of the selected Ethernet card that connects to the SQL server.
The “first” (build in) network card has always the ID “64” in all PLC types.
The “second” (build in) network card has the ID “72”, that applies only to 1500 PLC.

```
// PLC name in SQL Server
//
"SQL".SQL_Client.SQL_Setup.Hostname := 'Test';  // "Free" name
//User Name
"SQL".SQL_Client.SQL_Setup.Username := 'plcsql';    // User name in the PLCSQL-Link database (Default) (1)
// Password
"SQL".SQL_Client.SQL_Setup.Password := 'link';      // Password for user "plcsql" (Default)
// Database Name
"SQL".SQL_Client.SQL_Setup.Schema := 'plcsql';      // Name of database (Default)

```

“PLC name in SQL Server”, here you can type just what you want.
“User Name”, here you type the name of the “user” that connects to the SQL Server.
!! It is the “user” that decide which database there is connected to.
“Password”, here you type the password of the “user” that connects to the server.
“Database Name”, option, no use.

```

// Log stored procedure ( Query 3)
// Here you can chance between the "Log" and "Recipe" data when logging, for test,
// or if there is an read / write claim
IF "SQL".HMI.Ch_Log_Recipe THEN
    "SQL".SQL_Client.SQL_Setup.Query3 := 'sp_SaveParams 1, $'Recipe$', ';
ELSE
    "SQL".SQL_Client.SQL_Setup.Query3 := 'sp_SaveParams 1, $'Log$', ';
END_IF;

// Recipe stored procedure (Query 2)
"SQL".SQL_Client.SQL_Setup.Query2 := 'sp_GetParamSet ';

// If you write your own stored prcedures you have to change the "Query 2" and "Query 3"

```

The “standard” stored procedures that are called in the SQL database.

“Query2” is always used to get data from the SQL server to the PLC.
“Query3”, is always used to write data from the PLC to the SQL server, and here are there 2 options, 1 write to the “Log” tables (default), or 2 write to the “Recipe” tables.

```

// Setcount
"SQL".SQL_Client.SQL_Setup.PSetCount := 10001;  // Default parameter for numbers of send / recieve parameters
// SetID
"SQL".SQL_Client.SQL_Setup.PSetID := 15001;     // Default parameter for unique setid of send / recieve parameters
// DateTimeStamp
"SQL".SQL_Client.SQL_Setup.PDateTimeStamp := 30001; //Default parameter for Date / Time parameters saved.


```

At the end, the possibility to change the default addresses for the “SetCount”, “SetID” and “DateTimeStamp”.

Constants

The size of the different data types etc. is defined as constants in the “SQL_Controller” block.

Constant				
SizeRecieve	UInt	5120		Size Recieve block. MAX 65535
SizeQuery	UInt	5120		Size Query block. MAX 65535
LogParmREALMax	UInt	50		End ARRAY Parameter REAL Range 2 to 9999
LogParmINTMax	UInt	10051		End ARRAY Parameter INT Range 10003 to 14999, ParamNr. 14001 = ParamCont
LogParmDINTMax	UInt	15051		End ARRAY Parameter DINT Range 15003 to 19999, ParamNr. 15001 = ParamSetID
LogParmBOOLMax	UInt	20160		End ARRAY Parameter BOOL Range 20002 to 29999
LogParmSTRINGMax	UInt	30021		End ARRAY Parameter STRING Range 30003 to 30999, ParamNr. 30001 = DateTime saved.
LogParmSTRINGLength	UInt	40		Length String MAX 254
RecipeParmREALMax	UInt	50		End ARRAY Parameter REAL Range 2 to 9999
RecipeParmINTMax	UInt	10051		End ARRAY Parameter INT Range 10003 to 14999, ParamNr. 14001 = ParamCont
RecipeParmDINTMax	UInt	15051		End ARRAY Parameter DINT Range 15003 to 19999, ParamNr. 15001 = ParamSetID
RecipeParmBOOLMax	UInt	20160		End ARRAY Parameter BOOL Range 20002 to 29999
RecipeParmSTRINGMa	UInt	30021		End ARRAY Parameter STRING Range 30003 to 30999, ParamNr. 30001 = DateTime saved.
RecipeParmSTRINGLen	UInt	40		Length String MAX 254

Here are the definitions of the adjustable parameters in the system.
You may adjust the “max” values if you want to use parameters outside the selected areas, or to reduce the space used in the PLC memory.

The “SizeRecieve” and “SizeQuery” has to be adjusted if there is a major change in the number of parameters that are send (Query) or received.
With the shown parameters, you can send and receive all listed parameters (330 parameters).

S7 1500 PLC: Overview memory

Resources of PLCSQL1500_v1.8

	Objects	Load memory	Code work-memory	Data work-memory	Retain memory
1		23 %	18 %	25 %	0 %
2					
3	Total:	4 MB	153600 bytes	1048576 bytes	90784 bytes
4	Used:	967316 bytes	27519 bytes	257146 bytes	0 bytes
5	Details				
6	OB	6981 bytes	339 bytes		
7	Main [OB1]	6981 bytes	339 bytes		
8	FC	29978 bytes	1191 bytes		
9	SQL_Initialize [FC1460]	8390 bytes	455 bytes		
10	SQL_RecipeValues [FC14.	9460 bytes	310 bytes		
11	SQL_LogValues [FC1462]	12128 bytes	426 bytes		
12	FB	488574 bytes	25989 bytes		
13	SQL_Controller [FB1460]	171375 bytes	16886 bytes		
14	SQL_Client [FB1461]	276471 bytes	5552 bytes		
15	ControlTest [FB3000]	9088 bytes	348 bytes		
16	TestEnviroment [FB3001]	31640 bytes	3203 bytes		
17	DB	439757 bytes		257146 bytes	0 bytes
18	SQL [DB1460]	431775 bytes		256348 bytes	0 bytes
19	Test_Data [DB2000]	1383 bytes		276 bytes	0 bytes
20	ControlTestDB [DB3000]	2063 bytes		102 bytes	0 bytes
21	TestEnviromentDB [DB3...	4536 bytes		420 bytes	0 bytes
22	Objects for Motion Technology	-		-	0 bytes
23	Data types	-			
24	PLC tags	2026 bytes			0 bytes

Here is an example of the used memory in a 1511 PLC.
The 1500 system operates with 3 types of memory, the “Load memory”, the “Code work memory”, and “Data work memory”, the “Load memory” is the memory where you download the software to, the Load memory is always an external memory card. The “Code work memory” is a fixed area, where all executable code are placed, the “Data work memory” is a fixed area where all data are stored. All blocks (except SQL_Client) are compiled with “Optimized” option, so the maximum size of the “SQL” DB can be 16 Megabytes (depending of available space), that is the maximum limit in the S7 1500 system.
In the above configuration there is defined 5000 “Real”,1000 ”Int”,1000 ”Dint”,100 “String” with 40 char, and 1000 bool, in both “Logparm” and “RecipeParm”.

In this version of PLCSQL, there is a limit on the amount of data that can be send and received in 1 “job”. The maximum “Send” request is 65535 byte (adjustable), the maximum “Receive” request is 65535 bytes (adjustable in “constant” area SQL_Controller).
The “full” address area are selectable, but take care about the mentioned limits.

S7 1200 PLC: Overview memory

Resources of PLC_1200

	Objects	Load memory	Work memory	Retain memory
1		9 %	30 %	0 %
2				
3	Total:	4 MB	153600 bytes	10240 bytes
4	Used:	397910 bytes	46785 bytes	0 bytes
5	Details			
6	OB	3718 bytes	40 bytes	
7	FC	40704 bytes	1319 bytes	
8	SQL_Initialize [FC21460]	10444 bytes	405 bytes	
9	SQL_RecipeValues [FC21.	10600 bytes	277 bytes	
10	SQL_LogValues [FC2146:	12298 bytes	342 bytes	
11	SQL_Call [FC21463]	7362 bytes	295 bytes	
12	FB	313968 bytes	21114 bytes	
13	SQL_Controller [FB21460	183342 bytes	12810 bytes	
14	SQL_Client [FB21461]	84158 bytes	5329 bytes	
15	ControlTest [FB22000]	10334 bytes	256 bytes	
16	TestEnviroment [FB2200	36134 bytes	2719 bytes	
17	DB	38558 bytes	24312 bytes	0 bytes
18	SQL [DB21460]	31791 bytes	23666 bytes	0 bytes
19	TestData [DB21461]	2474 bytes	260 bytes	0 bytes
20	ControlTestDB [DB22000	1830 bytes	38 bytes	0 bytes
21	TestEnviromentDB [DB2...	2463 bytes	348 bytes	0 bytes
22	Objects for Motion Technology	-	-	0 bytes
23	Data types	-		
24	PLC tags	962 bytes		0 bytes

Here is an example of the used memory in a 1217 PLC.
The 1200 system operates with 2 types of memory, the “Load memory”, and the “Code work memory”, the “Load memory” is the memory where you download the software to, the Load memory can be an external memory card, or the build in memory. The “Code work memory” is a fixed area, where all executable code and the data are placed. All blocks (except SQL_Client) are compiled with “Optimized” option, so the maximum size of the “SQL” DB can be 16 Megabytes (depending of available space), that is the maximum limit in the S7 1200 system.
In the above configuration there is defined 50 “Real”,51 ”Int”,51 ”Dint”,21 “String” with 40 char, and 160 bool, in both “Logparm” and “RecipeParm”.

In this version of PLCSQL, there is a limit on the amount of data that can be send and received in 1 “job”. The maximum “Send” request is 65535 byte (adjustable), the maximum “Receive” request is 65535 bytes (adjustable in “constant” area SQL_Controller).
The “full” address area are selectable, but take care about the mentioned limits.

Open Controller PLC: Overview memory

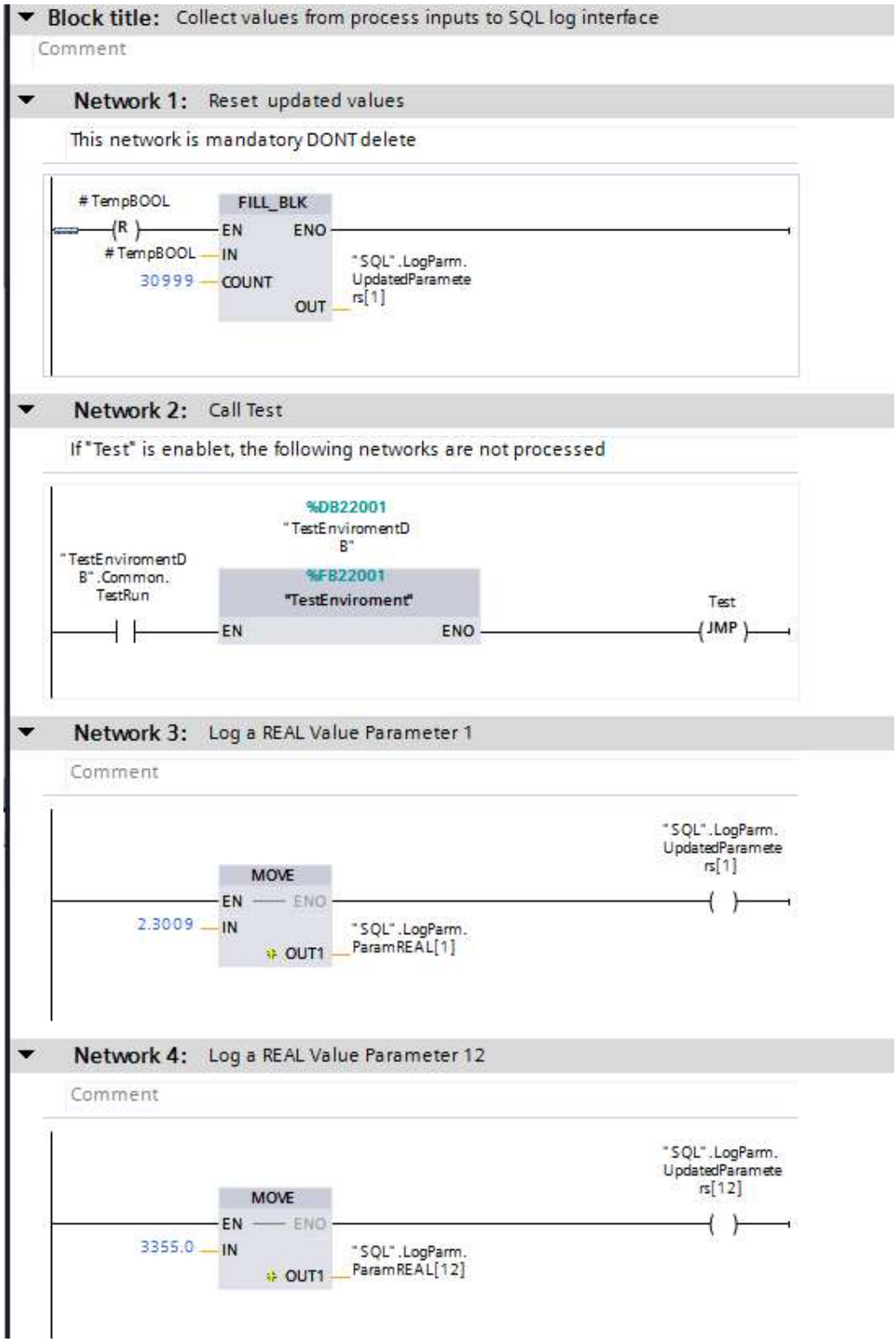
Resources of PLCSQL

Objects	Load memory	Code work-memory	Data work-memory	Retain memory
	0 %	2 %	2 %	0 %
Total:	320 MB	1048576 bytes	5242880 bytes	419840 bytes
Used:	453141 bytes	23284 bytes	79134 bytes	0 bytes
Details				
OB	3793 bytes	104 bytes		
Main [OB1]	3793 bytes	104 bytes		
FC	40229 bytes	1567 bytes		
SQL_Initialize [FC21460]	10469 bytes	470 bytes		
SQL_RecipeValues [FC21.	10288 bytes	304 bytes		
SQL_LogValues [FC2146:	12074 bytes	426 bytes		
SQL_Call [FC21463]	7398 bytes	367 bytes		
FB	315131 bytes	21613 bytes		
SQL_Controller [FB21460	183955 bytes	13047 bytes		
SQL_Client [FB21461]	84708 bytes	5396 bytes		
ControlTest [FB22000]	10323 bytes	336 bytes		
TestEnviroment [FB2200	36145 bytes	2834 bytes		
DB	93026 bytes		79134 bytes	0 bytes
SQL [DB21460]	86266 bytes		78288 bytes	0 bytes
TestData [DB21461]	2468 bytes		324 bytes	0 bytes
ControlTestDB [DB22000	1830 bytes		102 bytes	0 bytes
TestEnviromentDB [DB2...	2462 bytes		420 bytes	0 bytes

Here is an example of the used memory in a 1515 SP PLC. (Open Controller)
The 1500 system operates with 3 types of memory, the “Load memory”, the “Code work memory”, and “Data work memory”, the “Load memory” is the memory where you download the software to, the Load memory is always an external memory card. The “Code work memory” is a fixed area, where all executable code are placed, the “Data work memory” is a fixed area where all data are stored. All blocks (except SQL_Client) are compiled with “Optimized” option, so the maximum size of the “SQL” DB can be 16 Megabytes (depending of available space), that is the maximum limit in the S7 1500 system.
In the above configuration there is defined 50 “Real”,51 ”Int”,51 ”Dint”,21 “String” with 40 char, and 160 bool, in both “Logparm” and “RecipeParm”.

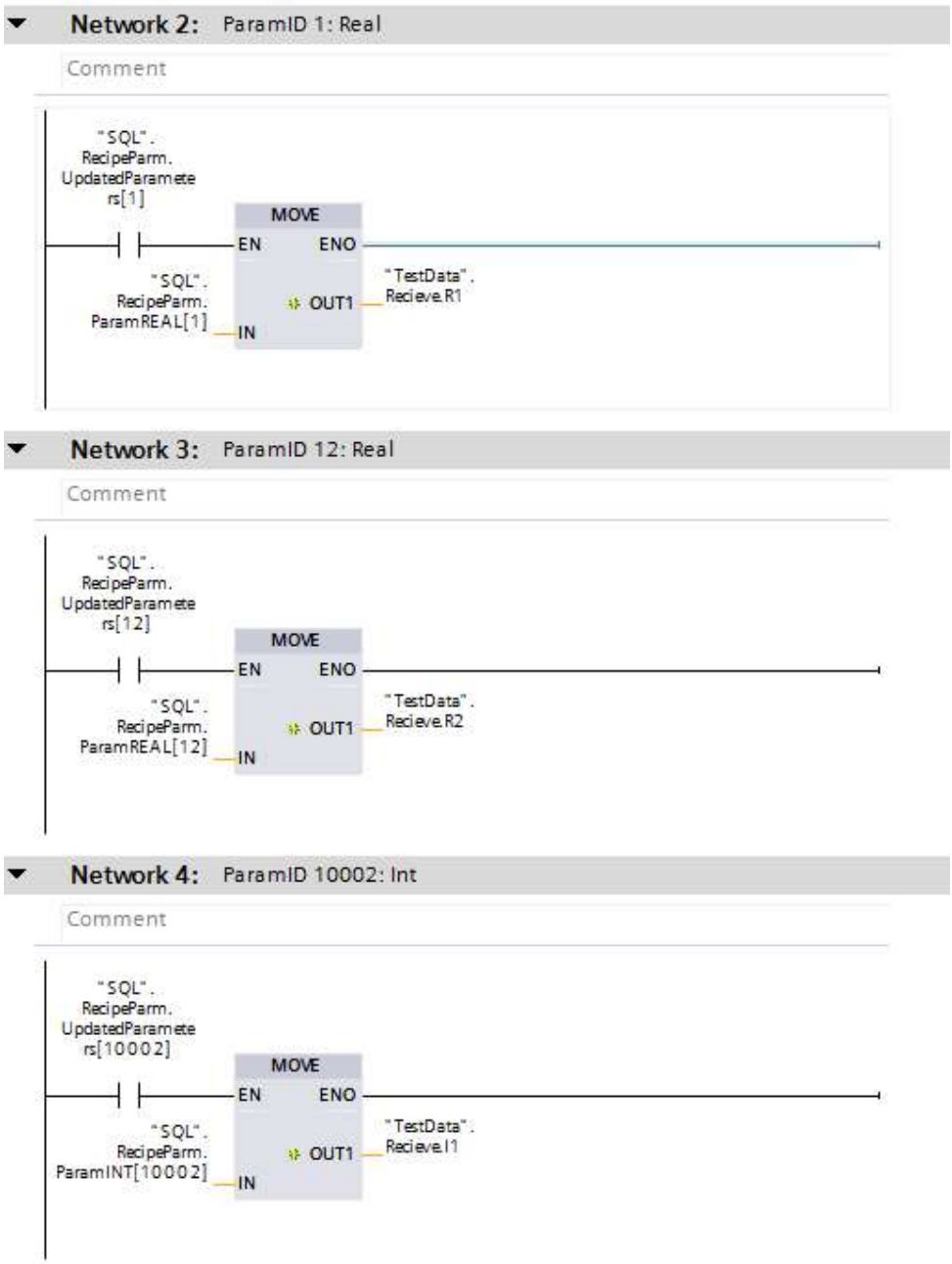
In this version of PLCSQL, there is a limit on the amount of data that can be send and received in 1 “job”. The maximum “Send” request is 65535 byte (adjustable), the maximum “Receive” request is 65535 bytes (adjustable in “constant” area SQL_Controller).
The “full” address area are selectable, but take care about the mentioned limits.

SQL_LogValues FC 21462



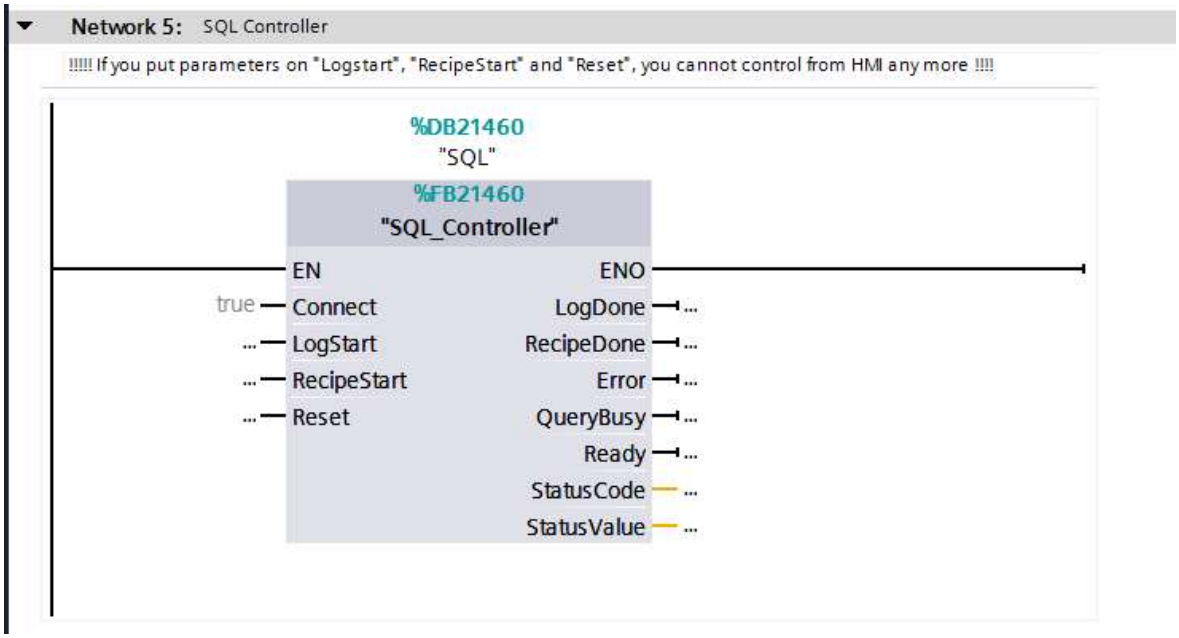
Here are some examples how to “Log” different types of values.
Network 2 can be deleted, together with network 13.

SQL_RecipeValues FC 21461



Here are some examples how to use “Recipe” data with different types of values.

SQL_Controller FB 21460



Overview of the “SQL_Controller”, all parameters can be controlled and seen from the HMI, if you want to control the block from both the HMI and the PLC, the you have to use the “Set” output in the PLC on the parameters “LogStart”, “RecipeStart”, and “Reset”.

SQL_Client FB 21461

“SQL_Client” is called from “SQL_Controller”, the block is protected and cannot be read or renumbered.

SQL DB 21460

This DB is the Instance DB for the hole SQL system, the DB contains all data areas needed for the SQL system.

Due to the use of only 1 Instance DB, it is very easy to use the SQL system as an “multiple” system, where you can call “SQL_Controller” multiple times by just using a new “SQL” data block, the only limitation is the amount of memory in the PLC.

Test_Data DB 21461

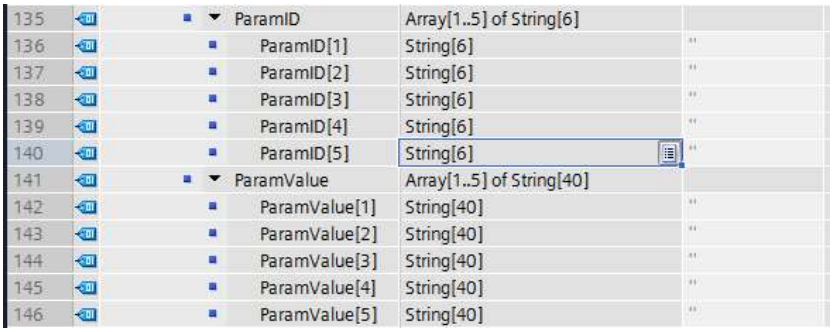
Test_Data

		Name	Data type	Start value	Sna
1		Static			
2		R1	Real	0.0	---
3		R2	Real	0.0	---
4		I1	Int	0	---
5		I2	Int	0	---
6		DI1	DInt	0	---
7		DI2	DInt	0	---
8		B1	Bool	false	---
9		B2	Bool	false	---
10		S1	String[40]	''	---
11		S2	String[40]	''	---

This DB is used to present the data received from the SQL server, if you use your own DB, this block can be deleted.

GetParmSet function (SQL -> PLC)

If you trigger the “RecipeStart”, then the stored procedure “sp_GetParamSet” is run on the SQL server. If you just run the command, you will get the data that have the highest “Recipe SetID”.
If you want to get “specific” data from the SQL server, you have the possibility to add some parameters to the “sp_GetParamSet” command.



The parameters are placed in “SQL.SQL_Client.SQL_Setup”
You have 5 parameters sets available.
“ParamID[X]”, contains the parameter number.
“ParamValue[X], contains the “value” you are looking for.
The parameters are treated with logical “and” function.
An example.

Parameter 15010 contains the part number of different “boxes”

Box 1.

15010 = 2134	SetID = x	DateTime = q	Part number.
10005 = 100	SetID = x	DateTime = q	Length in mm
10006 = 25	SetID = x	DateTime = q	Hight in mm
10023 = 100	SetID = x	DateTime = q	Wide in mm.

Box 2.

15010 = 9134	SetID = y	DateTime = q+1	Part number.
10005 = 100	SetID = y	DateTime = q+1	Length in mm
10007 = 50	SetID = y	DateTime = q+1	Hight in mm
10123 = 150	SetID = y	DateTime = q+1	Wide in mm.

Box 3.

15010 = 9134	SetID = z	DateTime = q+2	Part number.
10005 = 100	SetID = z	DateTime = q+2	Length in mm
10007 = 75	SetID = z	DateTime = q+2	Hight in mm
10123 = 150	SetID = z	DateTime = q+2	Wide in mm.

Syntax of the request schematic

“sp_GetparamSet,ParamID[1],ParamValue[1], ParamID[2],ParamValue[2], ParamID[3],ParamValue[3], ParamID[4],ParamValue[4], ParamID[5],ParamValue[5]”

Request 1
“sp_GetParamSet,15010,2134,,,,,,,,”

Result = Box 1

15010 = 2134	SetID = x	DateTime = q	Part number.
10005 = 100	SetID = x	DateTime = q	Length in mm
10006 = 25	SetID = x	DateTime = q	Hight in mm
10023 = 100	SetID = x	DateTime = q	Wide in mm.

Request 2
“sp_GetParamSet,15010,9134,,,,,,,,”

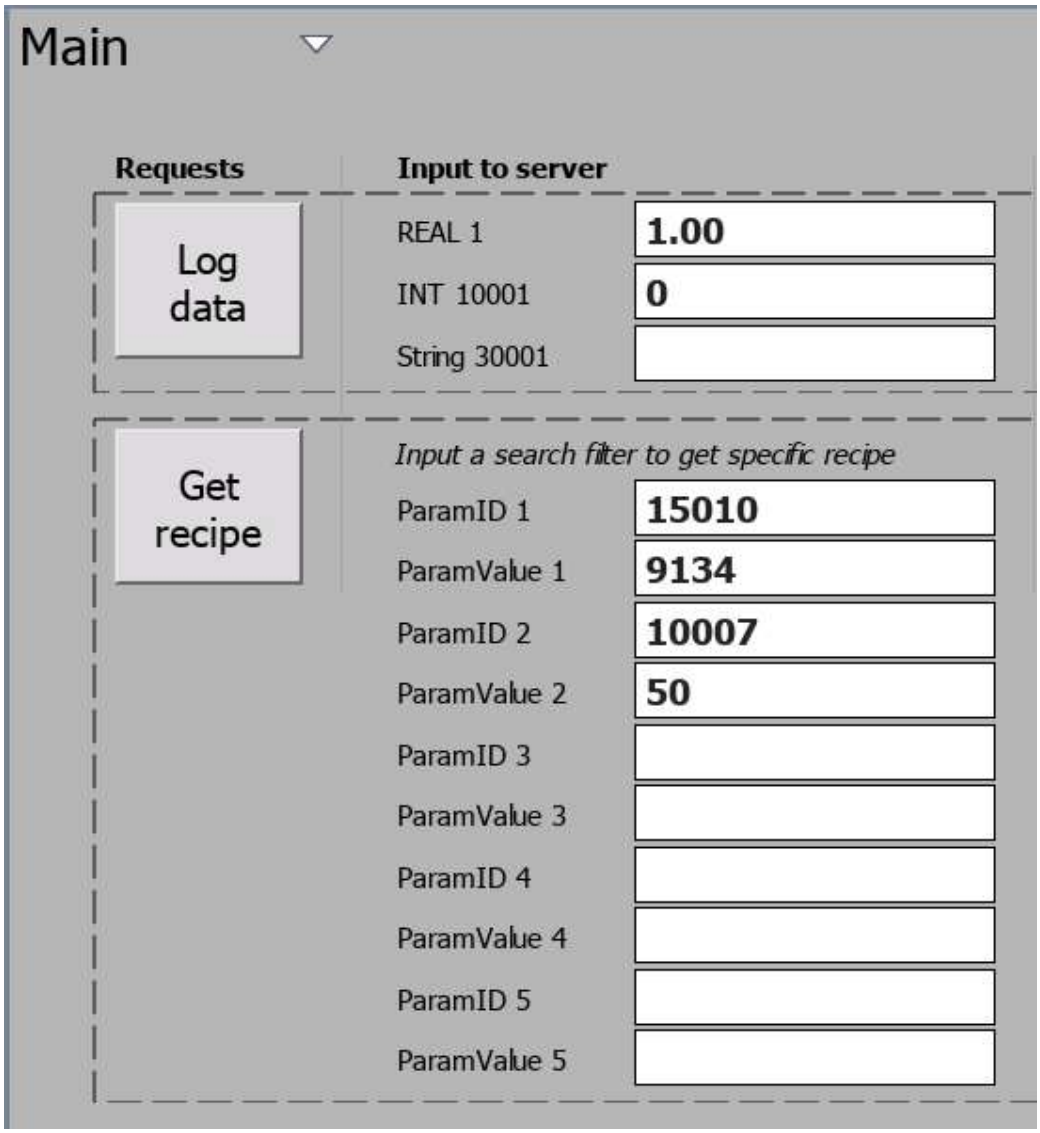
Result = Box 3, because box 3 is saved last, (newest timestamp)

15010 = 9134	SetID = z	DateTime = q+2	Part number.
10005 = 100	SetID = z	DateTime = q+2	Length in mm
10007 = 75	SetID = z	DateTime = q+2	Hight in mm
10123 = 150	SetID = z	DateTime = q+2	Wide in mm.

Request 2
“sp_GetParamSet,15010,9134,10007,50,,,,,,”

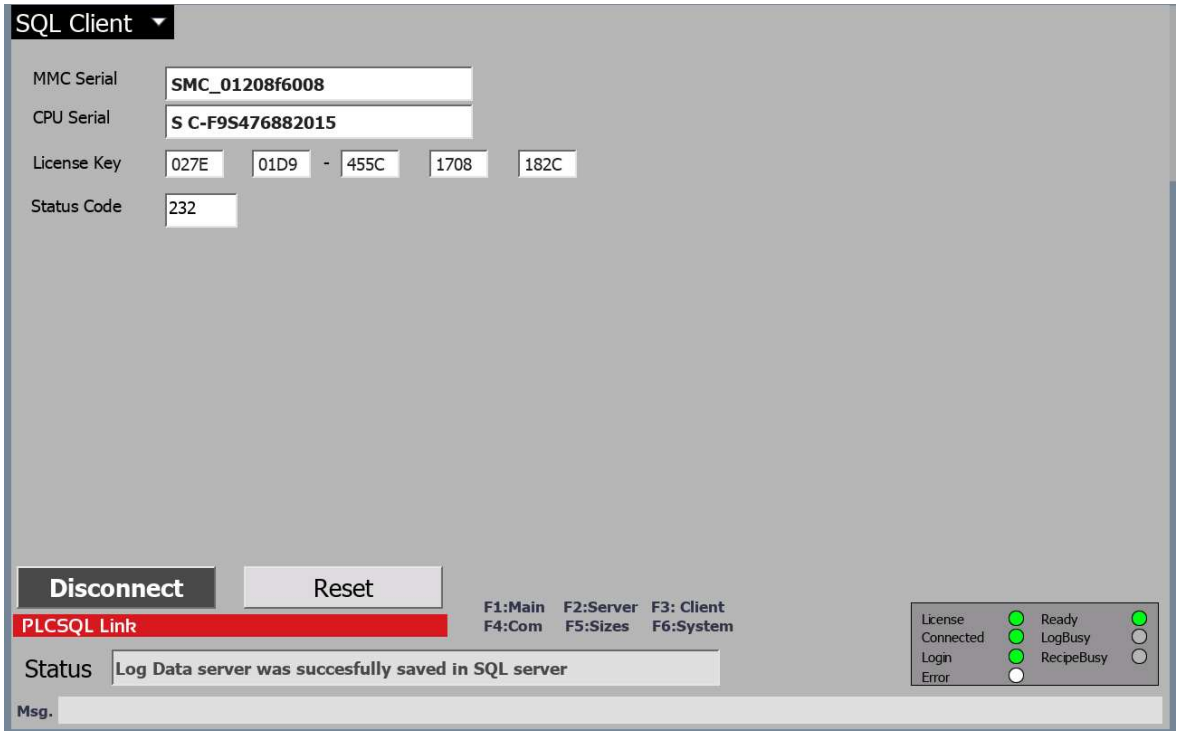
Result = Box 2, because box 2 only has parameter 10007 = 50

15010 = 9134	SetID = y	DateTime = q+1	Part number.
10005 = 100	SetID = y	DateTime = q+1	Length in mm
10007 = 50	SetID = y	DateTime = q+1	Hight in mm
10123 = 150	SetID = y	DateTime = q+1	Wide in mm.



Part of “Main” picture, where you can test, or see the parameters that are written from the program.

HMI-SQL Client (License key)



The “SQL Client” picture, read out the serial numbers of the CPU and of the memory card.
Here you also can type the license key that you got from Automatic Syd A/S.

If “SQL_Initialize” is running, you must type the license key in the block.
As default, there is no remanence data in the “SQL” DB, so the hole block is set to default when restarting the PLC.

HMI-SQL Server

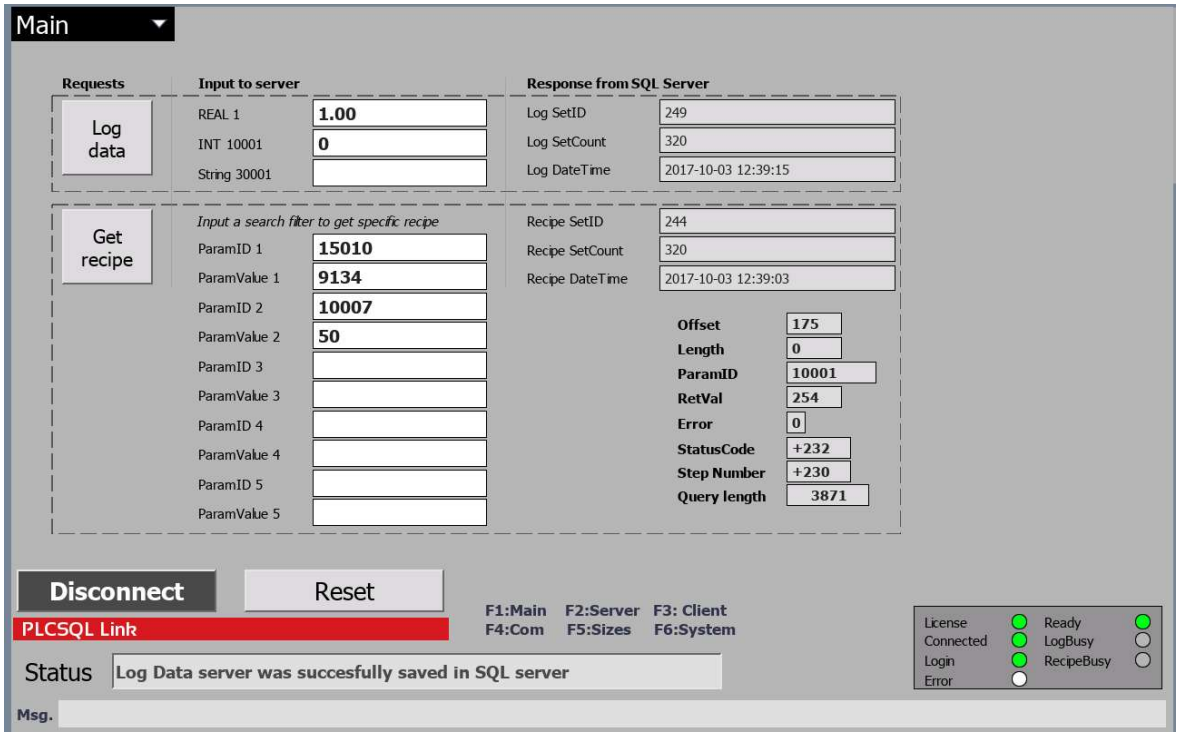
Here you select all the server relevant data.
The shown setup is the DEFAULT setup to match the DEFAULT setup of the Microsoft SQL Server.

If “SQL_Initialize” is running, you must type the changes in this block.
As default, there is no remanence data in the “SQL” DB, so the hole block is set to default when restarting the PLC.

On the “Recipe” button, you can change to “Log” data in the “Recipe” database, (as shown), then you have the possibility to write and read to the database from the PLC.

Default setup, where you log in the “Log” database.

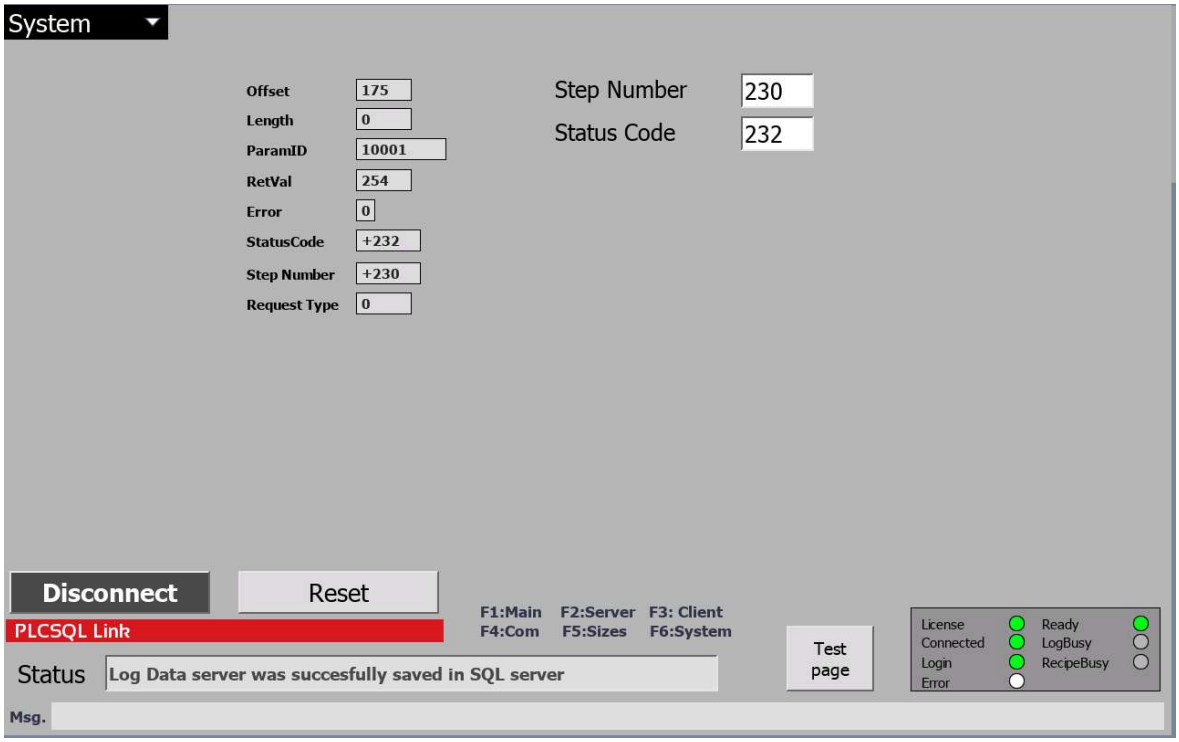
HMI-Main



Here you can trigger a “Log” request and a “Recipe” request.

You can also see the response from the SQL server.
“Log/Recipe SetID” is the unique number that every transaction with the server gets.
“Log/Recipe SetCount” is the number of “variables” that vas in the actual request.
“Log/Recipe DateTime” is the date and time where the actual data was stored in the SQL server.

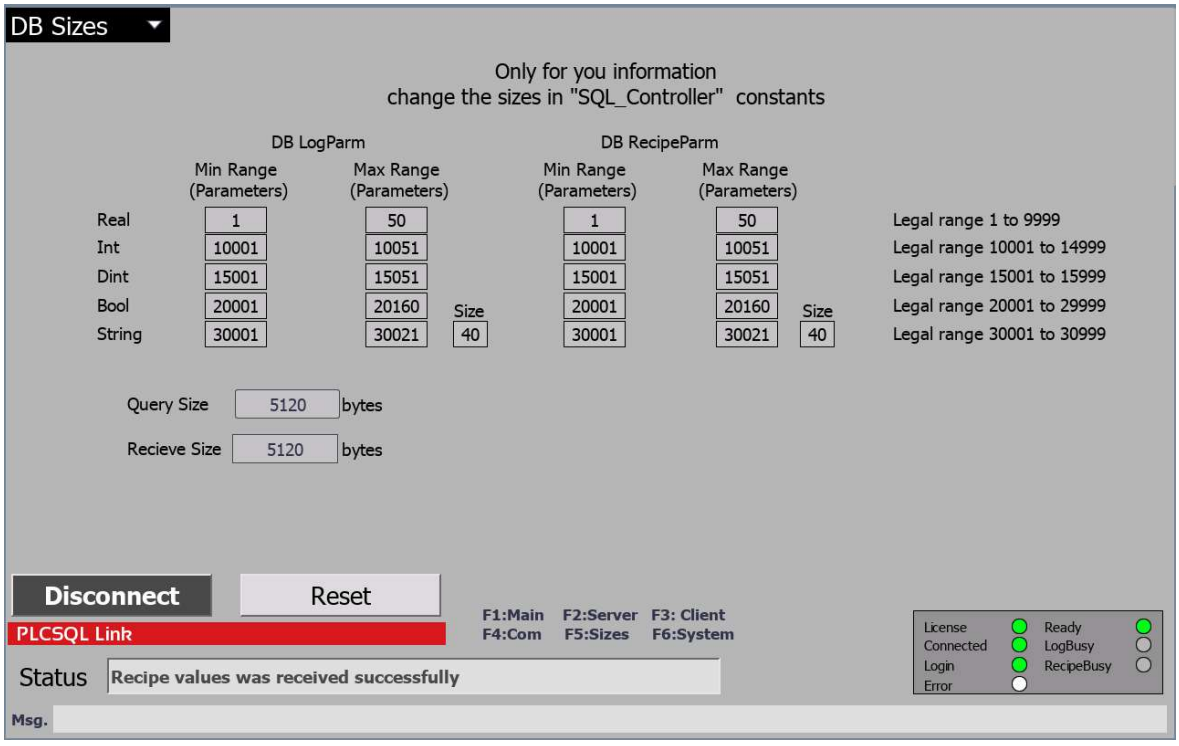
HMI-System



In the case of errors from the PLCSQL system, then it is important to get the status from these parameters.

The “Test page” button is used to start the test system, if you delete the “Test” picture you also must delete this button.

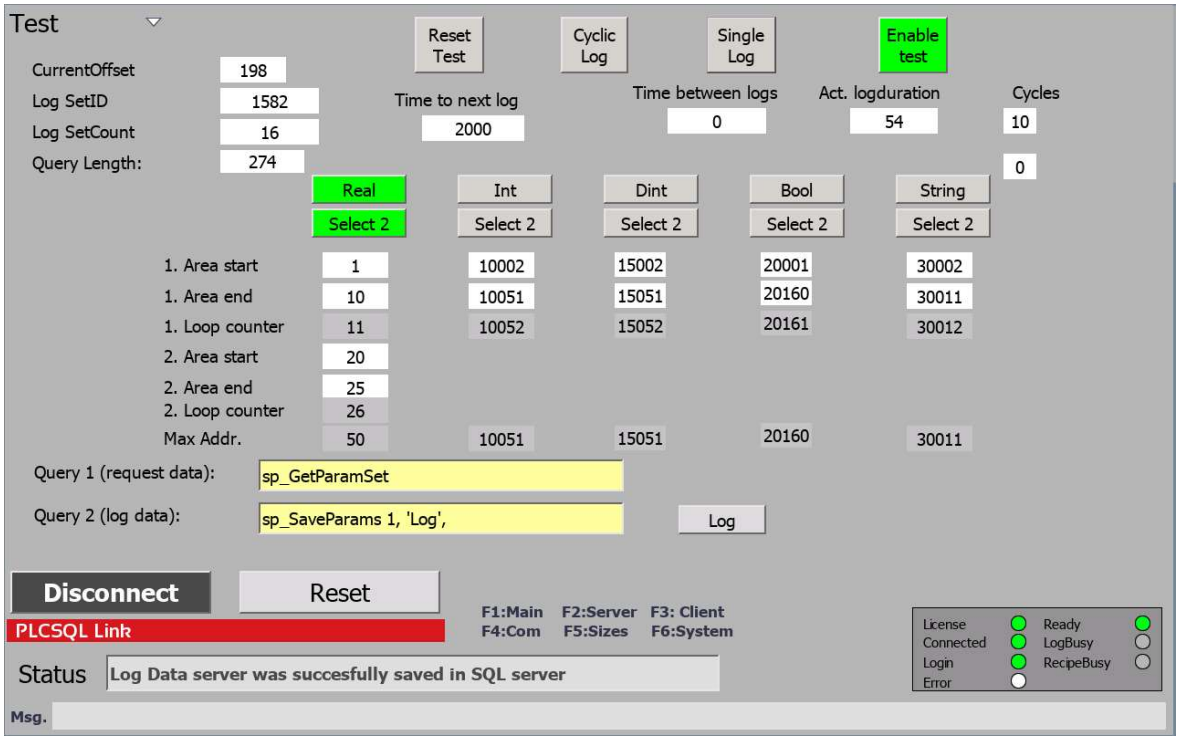
HMI-DB Sizes



Only as information to the user.

Adjust the size(s) in the “Constant” area of “PLCSQL_Controller”

HMI-Test



The system that is used to make “in house” test.
The datasets are generated in “loops”, every dataset has unique values, so it is possible to check in the database if there are errors in the transmission.
If you save the data in the “Recipe” tables, you also have the possibility to check the data stream from SQL server to PLC.

Should not be accessible in a “Production” environment.

The test system works only with the “default” SQL Server setup.

There is no check of the addressing, that means the possibility that the PLC goes in the STOP state is very high.

Revisions

2017-03-22, added comment regarding restart of PLC if “Device ID” is changed.

2017-10-05.	1.20, KJA, Automatic Syd A/S New numbering of the blocks. Change in HMI pictures. Change in program structure. Minor changes in the text. Added memory pictures.
2018-09-13	1.30, AJO, Alsmatik A/S