# Send data from Siemens S7-1500 PLC to MS SQL-Server via a 3rd party library [https://plcdirectsql.com/en/]

I had a requirement to send data of processed parts from PLC to a database. I choose the MS SQL-Server as this is very common one and almost all industrial IT have it on their server. This test I did as an example where there will be 6 different data for each processed part. These are:  
Datetime – String  
PartGroupID – Real  
AFM\_Param\_1 – Real  
AFM\_Param\_2 – Real  
AFM\_Param\_3 – Real  
PartID – String  
For this test I used a 3rd party Library from website [<https://plcdirectsql.com/en/>]. The unpaid demo version runs for 2 hours. Then we need to restart the PLC CPU if we want to run it again.  
A screenshot of a social media post

Description automatically generated

To test this in a more robust way instead of triggering each query with manual toggle I decided to make a small program which can generate random numbers(within a strict defined range) which simulates the part processing data, then using increment to increment to increment part number by 1. This was a good idea as I got more in-depth idea of 3 important functions and the code inside, those are:  
1. fcDTLString[FC] – Convert DTL data type (Datetime) to string  
2. fcRemovePlus[FC] – Remove the first plus sign (“+”) from a string  
3. LGF\_Random\_Real[FC] – Generates a random real number between 0.0 and 1.0

INSERT row to the table  
SQL query-  
INSERT INTO TestTable1 (Datetime,PartGroupID,AFM\_Param\_1,AFM\_Param\_2, AFM\_Param\_3, PartID) VALUES (‘2019-07-02 10:42:56′,’Deutz\_ASML\_10102A’,24,80.48,20.34,’ASML\_JDK32\_100847′)  
  
SCL code-  
‘INSERT INTO TestTable1(Datetime,PartGroupID,AFM\_Param\_1,AFM\_Param\_2, AFM\_Param\_3,PartID) VALUES($’2020-8-2 5:54:23.851$’, $’Deutz\_ASML\_10102A$’,+1.119162E+1, +5.051622E+0,+3.103162E+1,$’ASML\_JDK32\_100847$’)’

Hardware-Software setup

A screenshot of a cell phone

Description automatically generated

Steps:  
1. We need to have Siemens TIA portal 14 or above installed in our own machine or in a Virtual environment e.g. VM-Ware  
2. Install MS SQL-Server and MS SQL-Server Management Studio in our machine  
3. Connect S7-1500 with the laptop via an Ethernet cable or Switch  
4. Connect S7-1500 with Siemens TIA portal(installed in Virtual Machine in my case) via Switch or Ethernet-USB Adapter.  
5. Create the Table in SQL Database as you want with all the necessary Columns  
6. Create SCL code for SQL Query in PLC program to SELECT, INSERT, UPDATE, DELETE data in SQL Database

How to install MS SQL-Server and SQL-Server management studio  
<https://www.youtube.com/watch?v=QsXWszvjMBM>  
Learn SQL in 1 Hour – SQL Basics for Beginners  
<https://www.youtube.com/watch?v=9Pzj7Aj25lw>

Free SQL Database connection sample code for TIA Portal V16  
I did this test with a 3rd party library which needs to have paid license if we want to run it in Industrial mode continuously, demo mode runs only 2 hours. Now there is free sample for same purpose available in TIA Portal V16, unfortunately I still don’t have TIA Portal V16 so I decided to continue test with this external library. The free sample code from Siemens is here:  
https://support.industry.siemens.com/cs/document/109779336/connecting-an-s7-1200-s7-1500-to-an-sql-database-?dti=0&lc=en-YE

SCL Code of my Simulation test  
FUNCTION\_BLOCK "101\_Simulation"

TITLE = 101\_Simulation

{ S7\_Optimized\_Access := 'TRUE' }

VERSION : 0.1

VAR

State\_101\_Simulation : Int := 0;

dtlDateTime {InstructionName := 'DTL'; LibVersion := '1.0'; ExternalAccessible := 'False'; ExternalVisible := 'False'; ExternalWritable := 'False'} : DTL;

ton\_1 {InstructionName := 'TON\_TIME'; LibVersion := '1.0'} : TON\_TIME;

ton\_2 {InstructionName := 'TON\_TIME'; LibVersion := '1.0'} : TON\_TIME;

part\_processing\_data { ExternalAccessible := 'False'; ExternalVisible := 'False'; ExternalWritable := 'False'} : "part\_processing\_data";

bToggler { ExternalAccessible := 'False'; ExternalVisible := 'False'; ExternalWritable := 'False'} : Bool;

iToggler { ExternalAccessible := 'False'; ExternalVisible := 'False'; ExternalWritable := 'False'} : Int;

END\_VAR

VAR\_TEMP

"\*\*\*\*\*\*\*\*\*\*\*\*\*2" : AOM\_IDENT;

sFruitName : String;

iQuantity : Int;

rWeight : Real;

bCitrusType : Bool;

"\*\*\*\*\*\*\*\*\*\*\*\*\*1" : AOM\_IDENT;

str : String;

LGF\_Random\_Real\_error : Bool;

LGF\_Random\_Real\_status : Word;

LGF\_Random\_Real\_subfunctionStatus : Word;

LGF\_Random\_Real : Real;

END\_VAR

BEGIN

CASE #State\_101\_Simulation OF

0: //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#ton\_1(IN:=TRUE,PT:=t#5000ms);

IF #ton\_1.Q THEN

#ton\_1(IN := FALSE,PT := t#0s);

#State\_101\_Simulation := 10;

END\_IF;

10: //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#LGF\_Random\_Real:= "LGF\_Random\_Real"(error=>#LGF\_Random\_Real\_error,

status=>#LGF\_Random\_Real\_status,

subfunctionStatus=>#LGF\_Random\_Real\_subfunctionStatus);

#bToggler := NOT #bToggler;

IF #bToggler = TRUE THEN

#iToggler := 1;

ELSE

#iToggler := -1;

END\_IF;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#part\_processing\_data.sPartGroupID := "InputsDB".part\_processing\_data.sPartGroupID;

#part\_processing\_data.rAFM\_Param\_1 := "InputsDB".#part\_processing\_data.rAFM\_Param\_1+ (#LGF\_Random\_Real\*#iToggler);

#part\_processing\_data.rAFM\_Param\_2 := "InputsDB".part\_processing\_data.#rAFM\_Param\_2+ (#LGF\_Random\_Real\*#iToggler);

#part\_processing\_data.rAFM\_Param\_3 := "InputsDB".part\_processing\_data.#rAFM\_Param\_3+ (#LGF\_Random\_Real\*#iToggler);

"InputsDB".part\_processing\_data.iPartID\_2 := "InputsDB".part\_processing\_data.iPartID\_2+1;

#part\_processing\_data.sPartID\_1 := CONCAT(IN1 := "InputsDB".part\_processing\_data.sPartID\_1,

IN2 := "fcRemovePlus"(UDINT\_TO\_STRING("InputsDB".part\_processing\_data.iPartID\_2)));

// DO NOT FORGET TO UPDATE YOUR [TABLE NAME](e.g. TestTable1) IN THE BELOW FIRST STRING !

#str := 'INSERT INTO TestTable1(Datetime,PartGroupID,AFM\_Param\_1,AFM\_Param\_2,AFM\_Param\_3,PartID) VALUES($'';

#str := CONCAT(IN1 := #str, IN2 := "fcDTLString"(#dtlDateTime));

#str := CONCAT(IN1 := #str, IN2 := '$',$'');

#str := CONCAT(IN1 := #str, IN2 := #part\_processing\_data.sPartGroupID);

#str := CONCAT(IN1 := #str, IN2 := '$',');

#str := CONCAT(IN1 := #str, IN2 := REAL\_TO\_STRING(#part\_processing\_data.rAFM\_Param\_1));

#str := CONCAT(IN1 := #str, IN2 := ',');

#str := CONCAT(IN1 := #str, IN2 := REAL\_TO\_STRING(#part\_processing\_data.rAFM\_Param\_2));

#str := CONCAT(IN1 := #str, IN2 := ',');

#str := CONCAT(IN1 := #str, IN2 := REAL\_TO\_STRING(#part\_processing\_data.rAFM\_Param\_3));

#str := CONCAT(IN1 := #str, IN2 := ',');

#str := CONCAT(IN1 := #str, IN2 := '$'');

#str := CONCAT(IN1 := #str, IN2 := #part\_processing\_data.sPartID\_1);

#str := CONCAT(IN1 := #str, IN2 := '$'');

#str := CONCAT(IN1 := #str, IN2 := ')');

"DB\_PDSql\_1500".Query.Query[1] := #str;

"DB\_PDSql\_1500".Query.Query[2] := '';

IF TRUE THEN

"DB\_PDSql\_1500".Cmd.ExecuteQuery := TRUE;

#State\_101\_Simulation := 100;

END\_IF;

100: //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#ton\_1(IN := TRUE,PT := t#5000ms);

IF #ton\_1.Q AND TRUE THEN

#ton\_1(IN := FALSE,PT := t#0s);

"DB\_PDSql\_1500".Cmd.ExecuteQuery := FALSE;

#State\_101\_Simulation := 0;

END\_IF;

ELSE // Statement section ELSE

;

END\_CASE;

END\_FUNCTION\_BLOCK