Kimball Electronics

Traceability Simu V1.0.0.0

App User Manual

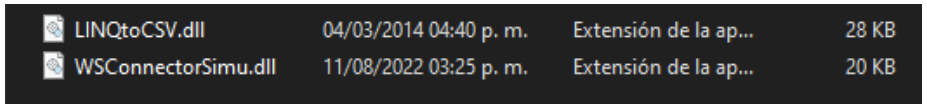
Kimball Electronics

General information

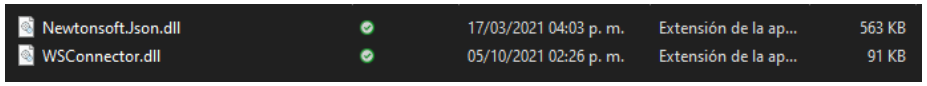
The purpose of this guide is to explain how the Traceability Simu application works and how to connect the .dll WSConnectorSimu that simulates the original .dll of the traceability system of Kimball Electronics Mexico Plant 1 & Plan 2.

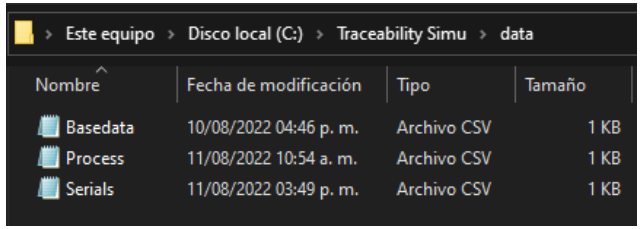
How is works?

WSConnector simu



WSConnector original





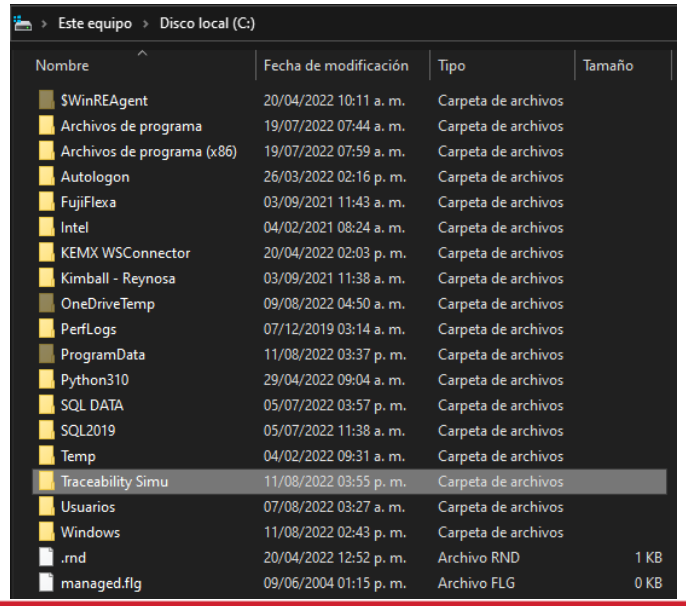
The WSConnectorSimu .dll simulates the operation using .csv files to simulate a database with the PartNumber, BackCheck and InsertProcessDataWithFails functions needed in the traceability system of the company.

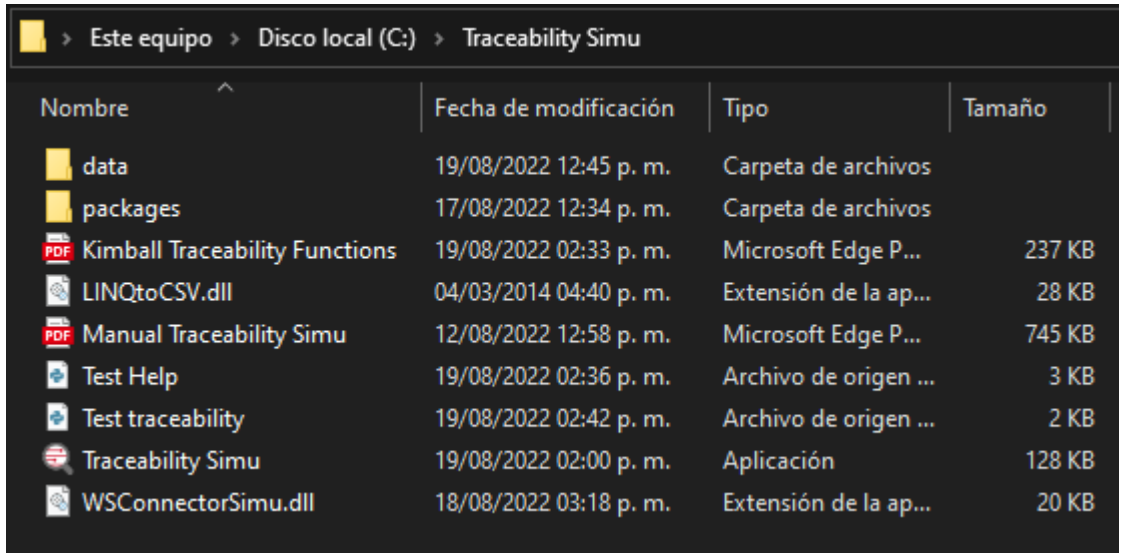
It is programmed in the same way as WSConnector, which is the original .dll. So once the application is debugged with the help of this application the changes would be minimal.

Installation

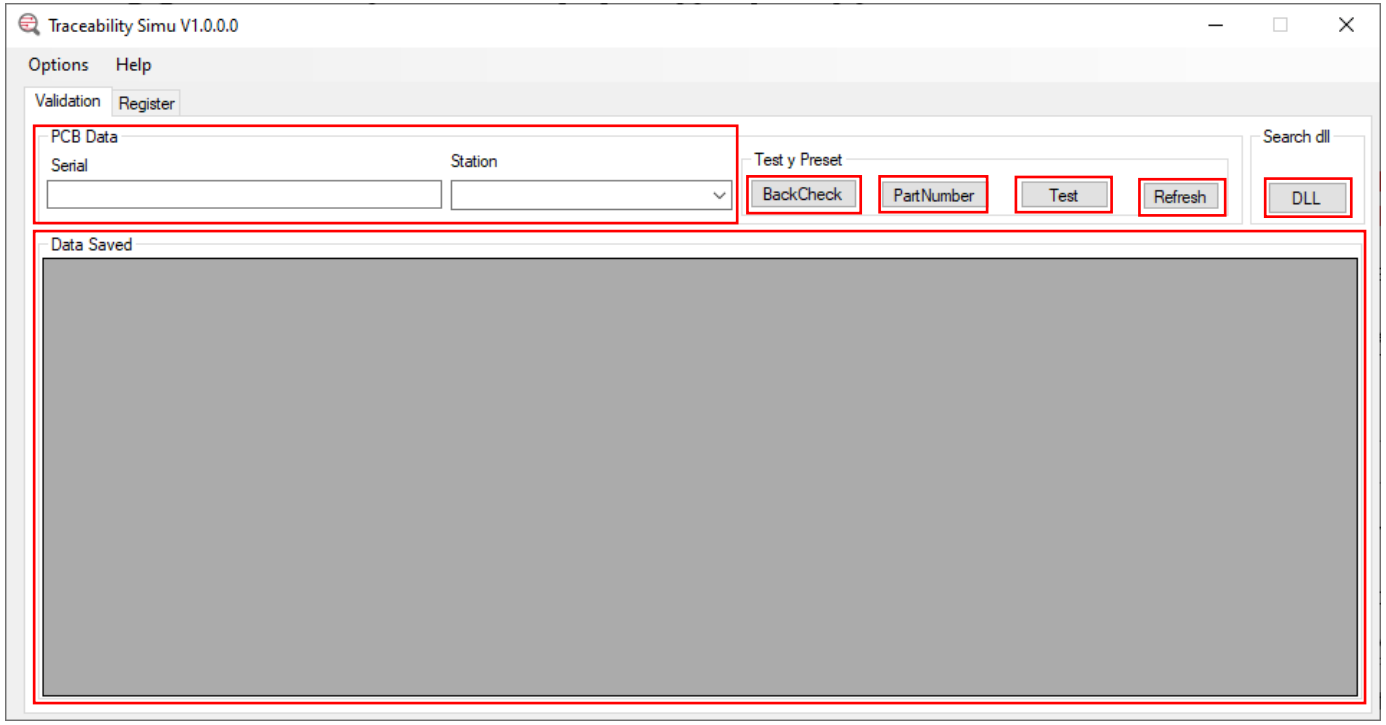
The files are in a compressed .rar

Unzip the file and move the folder to local disk C:\





Overview: Validation



In this screen you can perform BackCheck and PartNumber tests. Also add tests to a serial simulating the normal process of a PCB.

Option: Set the plant to use.

Help: Quick access to program support.

PCB DATA: Here you enter the PCB information to perform the test.

Button BackCheck: Here you check if the station where the PCB is entered is the one that corresponds to the process. Shows the string of BackCheck test: **status|Process**

Button PartNumber: Returns the part number string.

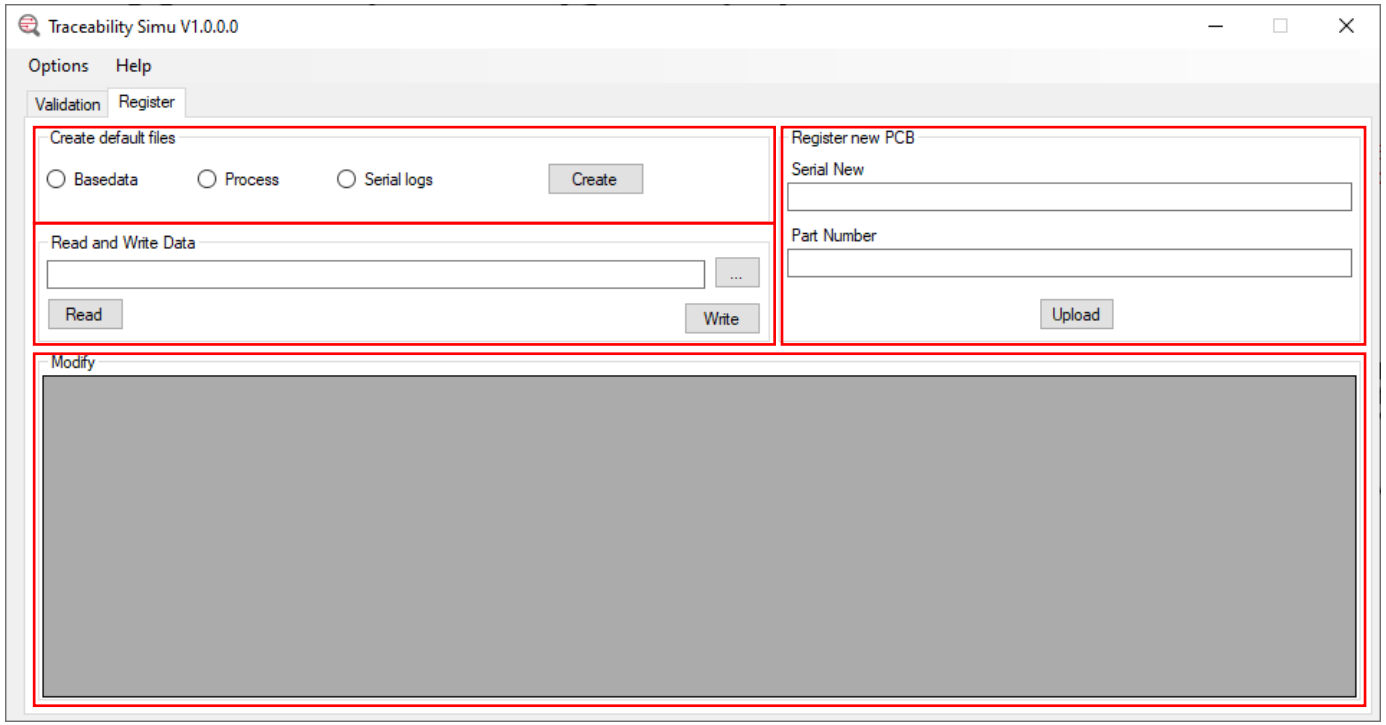
Button Test: Simulates a station test, where you can decide whether to pass or fail, and is saved in the database. Returns **"OK"** if successfully uploaded

Button Refresh: Updates the table where the saved information can be displayed.

Button DLL: Open windows explorer in the folder where the program's .dll files are located.

Data Saved: The complete PCB information is displayed.

Overview: Register



In this screen PCBs are registered, and databases can be created or modified if necessary.

Create default files: By default, the application creates the files at startup, but allows you to overwrite them if needed.

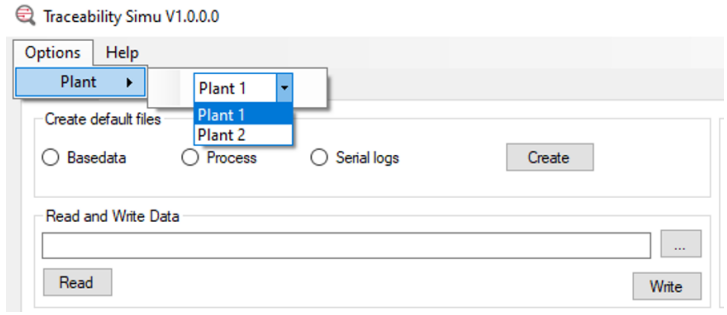
Read and Write data: Allows to select a file with "..." and "read" to view it. To modify it, type in the desired cell of the Modify table and press "Write".

Register New PCB: Register a new Serial together with the part number provided. When you upload it, you will see it in the table.

Modify: Table that allows you to view, modify or delete an entry in the selected file.

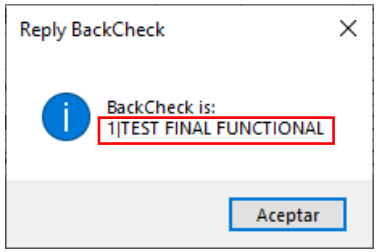
Configuration

Select the plant that to use. Depending on the plant, the responses will be different. The functions dll output can be displayed by pressing the buttons.

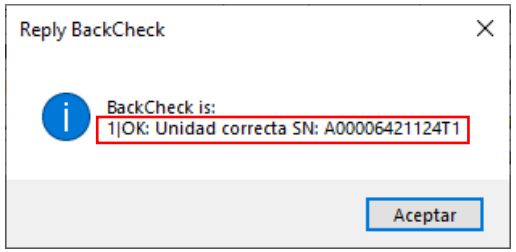


Note: Only can change the plant before to use the bottons, if want change, reset Traceability Simu.

Plant 1

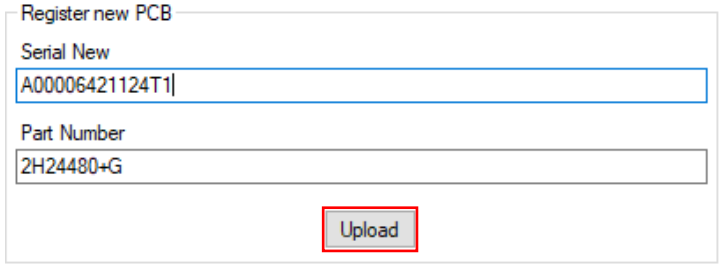


Plant 2

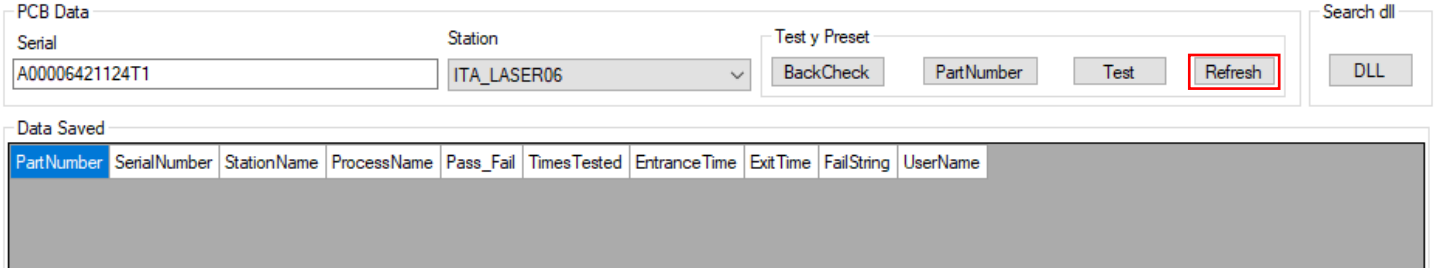


How to use

1.- Enter the PCB serial and part number and press Upload.

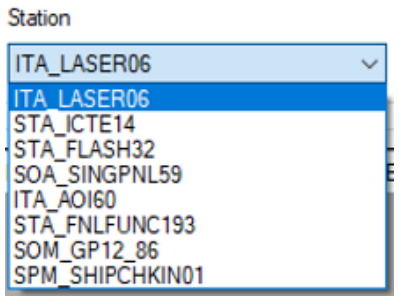


2.- Once registered, the serial number and the station where the test is being simulated are rewritten in PCB data.



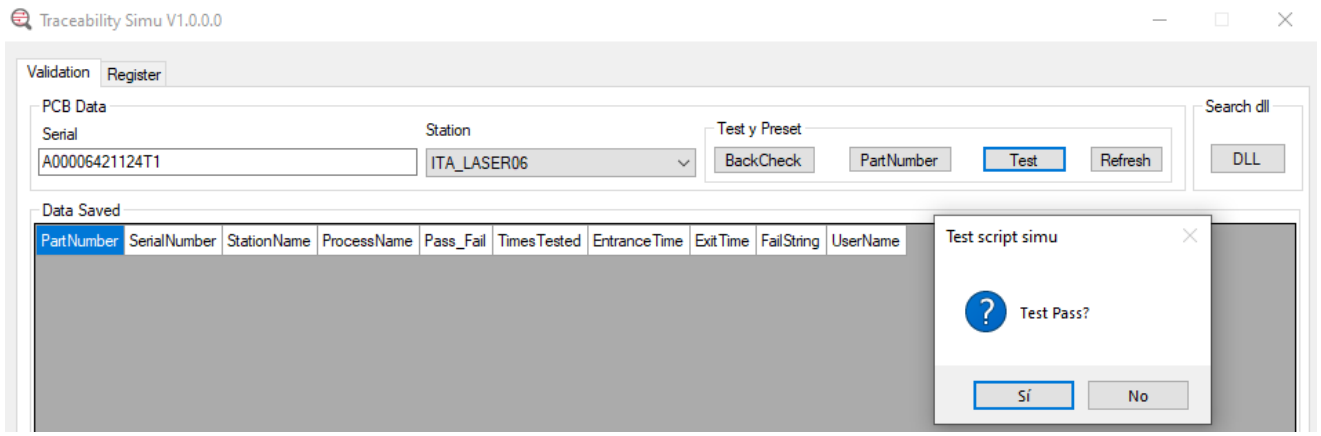
Note: The stations were preconfigured to the actual process of the company, if you want to modify the name, you can do it from the Read and Write Data with the "Process" file explained above. Once done, press the "Refresh" button to update the stations.

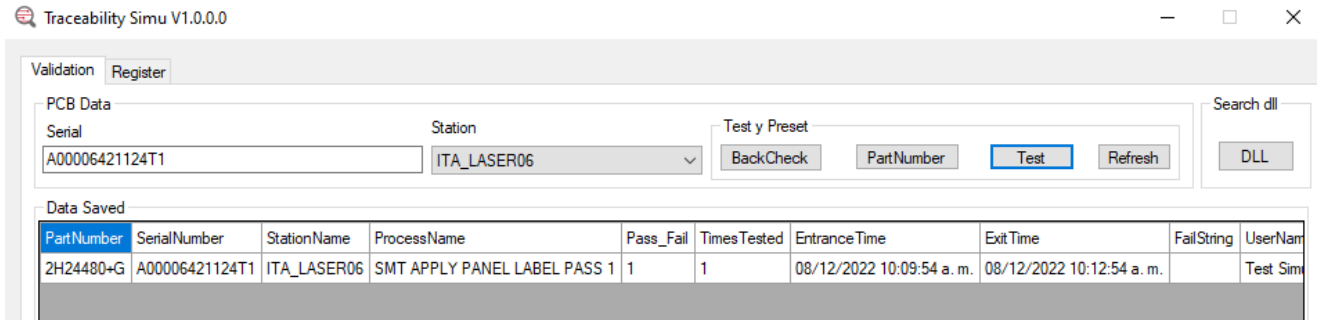
Not change the first station.



NumberProcess	Station	Process
1	ITA_LASER06	SMT APPLY PANEL LABEL PASS 1
2	STA_ICTE14	TEST ICT
3	STA_FLASH32	FLASH PROGRAM
4	SOA_SINGPNL59	GENERAL ROUTER DEPANEL
5	ITA_AOI60	CONFORMAL INSPECTION PASS 1
6	STA_FNLFUNC193	TEST FINAL FUNCTIONAL
7	SOM_GP12_86	TEST GP12 PASS 1
8	SPM_SHIPCHKIN01	SHIPPING CHECK-IN

How to use

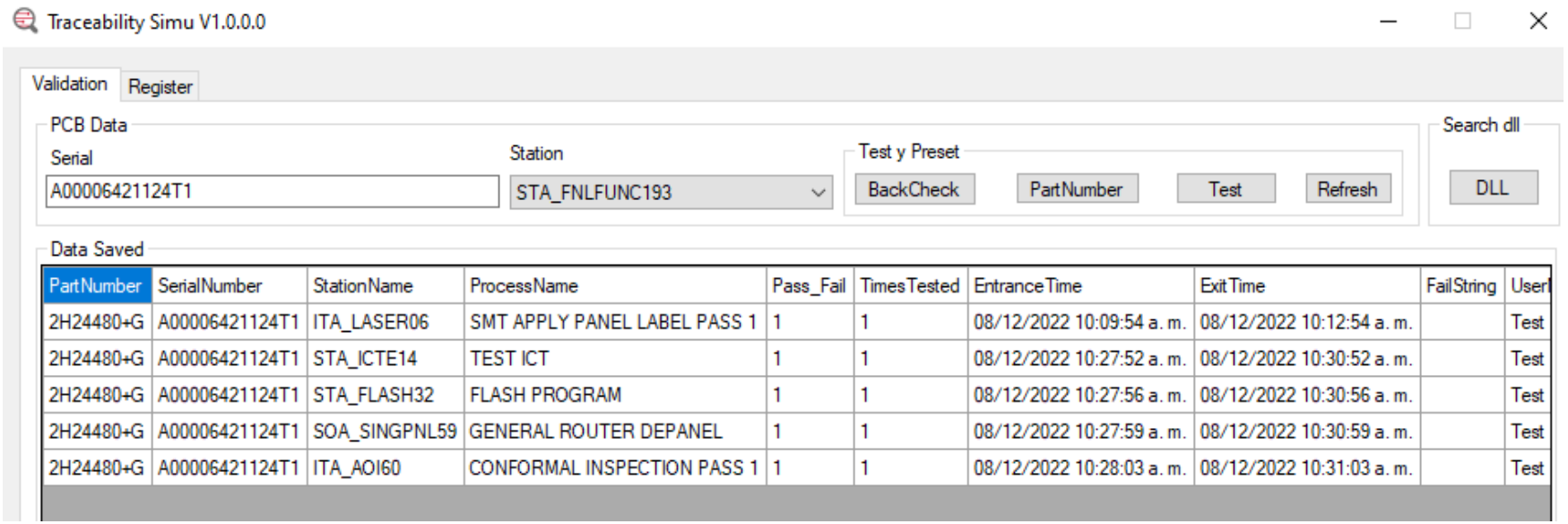




To add a test, press the Test button and select "Yes". Repeat this until the desired station is reached.

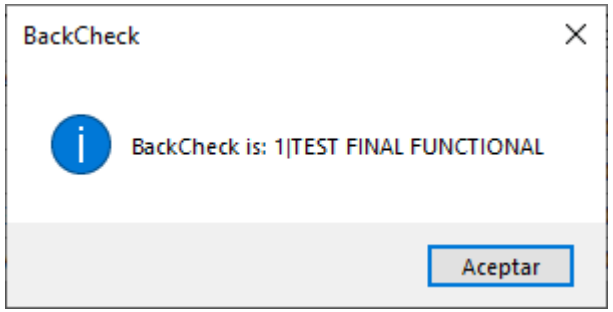
To add the next station, you must choose the next Station, because if you leave the same one you just registered, it will give error in BackCheck. If this happens, a pop-up window will appear with the correct process.

How to use



PartNumber	SerialNumber	StationName	ProcessName	Pass_Fail	TimesTested	EntranceTime	ExitTime	FailString	User
2H24480+G	A00006421124T1	ITA_LASER06	SMT APPLY PANEL LABEL PASS 1	1	1	08/12/2022 10:09:54 a. m.	08/12/2022 10:12:54 a. m.		Test
2H24480+G	A00006421124T1	STA_ICTE14	TEST ICT	1	1	08/12/2022 10:27:52 a. m.	08/12/2022 10:30:52 a. m.		Test
2H24480+G	A00006421124T1	STA_FLASH32	FLASH PROGRAM	1	1	08/12/2022 10:27:56 a. m.	08/12/2022 10:30:56 a. m.		Test
2H24480+G	A00006421124T1	SOA_SINGPNL59	GENERAL ROUTER DEPANEL	1	1	08/12/2022 10:27:59 a. m.	08/12/2022 10:30:59 a. m.		Test
2H24480+G	A00006421124T1	ITA_AOI60	CONFORMAL INSPECTION PASS 1	1	1	08/12/2022 10:28:03 a. m.	08/12/2022 10:31:03 a. m.		Test

Once you have gone through the required stations to get to the station to be tested with the test software, you can check the BackCheck to verify that everything is correct.



How to use

test.py 1 ●

C: > Traceability Simu > test.py > ...

```python
from os import stat
import clr

###this is an example of how to call functions with code###

#These names are kept because they are from the original .dll for the traceability system
newtonsoftjson_path = r"C:\Traceability Simu\LINQtoCSV.dll"
wsconnector_path = r"C:\Traceability Simu\WSConnectorSimu.dll"

clr.AddReference(newtonsoftjson_path)
clr.AddReference(wsconnector_path)

#PCB information example
partnumber = ""
serial = "A00006421124T1"
station_name = "STA_FNLFUNC193"
process = "TEST FINAL FUNCTIONAL"
test_start_time = "08/04/2022 14:45:39 p. m"
test_end_time = "08/04/2022 14:48:39 p. m"
status = 1 #choice 0 if failed test or 1 if passed test
#if status is 1, fail string is "", if is 0, string is how the example
fail_string = ""            #"|ftestres=0,'FUNCTIONAL TEST + BLOCKING'-'TEST SIGNAL CFC'-'CFC = 0-CFR=0-EFL=0-EFH=0',0.8634312,260.0,200.0,230.0,tbd,<>"
employee = "Test Simu"

#in the original is "from WSConnector import Connector"
from WSConnectorSimu import Connector

connector = Connector()
print("1.-CIMP_PartNumberRef")
print("2.-BackCheck_Serial")
print("3.-InsertProcessDataWithFails")
choice = input("What function do you want to test? ")
if choice == '1':
    reply = connector.CIMP_PartNumberRef(serial, 1, partnumber) #connector.CIMP_PartNumberRef(string SerialNumber, int BCTYPE, ref string AssemblyPartNumber) in .dll original
    print (reply)
elif  choice == '2':
    reply = connector.BackCheck_Serial(serial, station_name) #connector.BackCheck_Serial(string _serialNumber, string _stationName) in .dll original
    print (reply)
elif choice == '3':
    reply = connector.InsertProcessDataWithFails(serial,station_name,process,test_start_time,test_end_time,status,fail_string,employee) #connector.InsertProcessDataWithFails(string ser_num, string station_name, st
    print (reply)
else:
    print ("select correct option")

```

4.- To make the connection from the testing software, the functions must be called as shown in this example.

Here is shown how to do it from Python. Running the script allows you to choose which of the 3 functions you want to test. For more information see the SOW Template for Functional Test.

How to use

```
PS C:\Users\K90011729> & C:/Users/K90011729/AppData/Local/Programs/Python/Python310/python.exe "c:/Traceability Simu/test.py"
1.-CIMP_PartNumberRef
2.-BackCheck_Serial
3.-InsertProcessDataWithFails
What function do you want to test? 3
OK
```

5.- In this example you select to test the InsertProcessDataWithFails, it returns an "OK". In Traceability Simu press the "Refresh" button to visualize the data that has been sent.

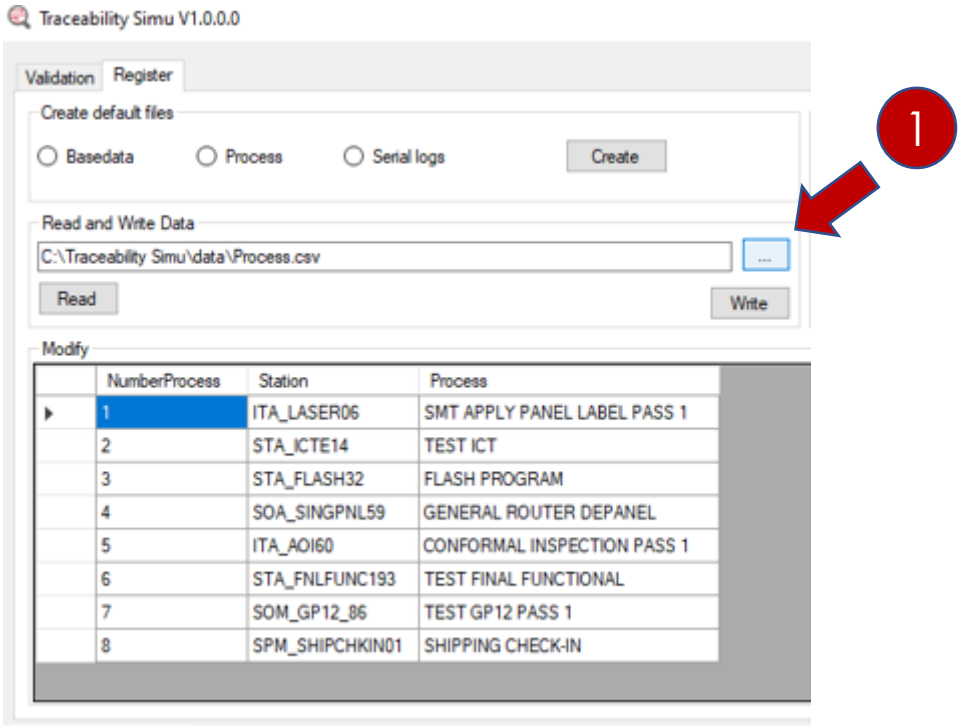
```
#PCB information example
partnumber = ""
serial = "A00006421124T1"
station_name = "STA_FNLFUNC193"
process = "TEST FINAL FUNCTIONAL"
test_start_time = "08/04/2022 14:45:39 p. m"
test_end_time = "08/04/2022 14:48:39 p. m"
status = 1 #choice 0 if failed test or 1 if passed test
#if status is 1, fail string is "", if is 0, string is how the example
fail_string = ""            #"|ftestres=0,'FUNCTIONAL TEST + BLOCKING'-'TEST SIGNAL CFC'-'CFC = 0-CFR=0-EFL=0-EFH=0',0.8634312,260.0,200.0,230.0,tbd,<>"
employee = "Test Simu"
```

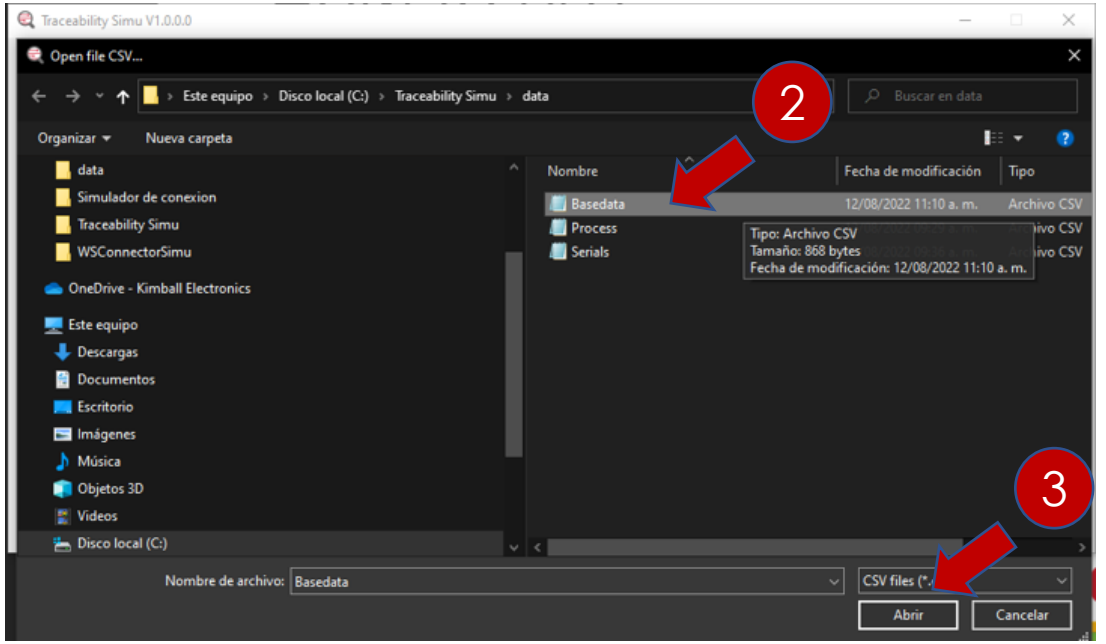
Data Saved

PartNumber	SerialNumber	StationName	ProcessName	Pass_Fail	TimesTested	EntranceTime	ExitTime	FailString	Use
2H24480+G	A00006421124T1	ITA_LASER06	SMT APPLY PANEL LABEL PASS 1	1	1	08/12/2022 10:09:54 a. m.	08/12/2022 10:12:54 a. m.		Tes
2H24480+G	A00006421124T1	STA_ICTE14	TEST ICT	1	1	08/12/2022 10:27:52 a. m.	08/12/2022 10:30:52 a. m.		Tes
2H24480+G	A00006421124T1	STA_FLASH32	FLASH PROGRAM	1	1	08/12/2022 10:27:56 a. m.	08/12/2022 10:30:56 a. m.		Tes
2H24480+G	A00006421124T1	SOA_SINGPNL59	GENERAL ROUTER DEPANEL	1	1	08/12/2022 10:27:59 a. m.	08/12/2022 10:30:59 a. m.		Tes
2H24480+G	A00006421124T1	ITA_AOI60	CONFORMAL INSPECTION PASS 1	1	1	08/12/2022 10:28:03 a. m.	08/12/2022 10:31:03 a. m.		Tes
2H24480+G	A00006421124T1	STA_FNLFUNC193	TEST FINAL FUNCTIONAL	1	1	08/04/2022 14:45:39 p. m	08/04/2022 14:48:39 p. m		Tes

How to use

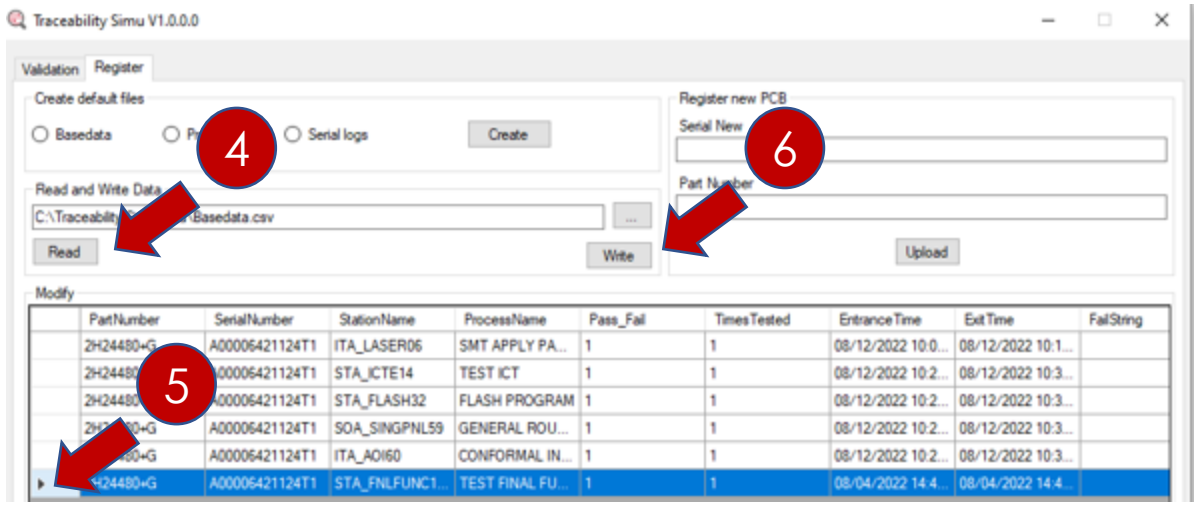
6.- To delete the station and run the test again, follow the steps below.

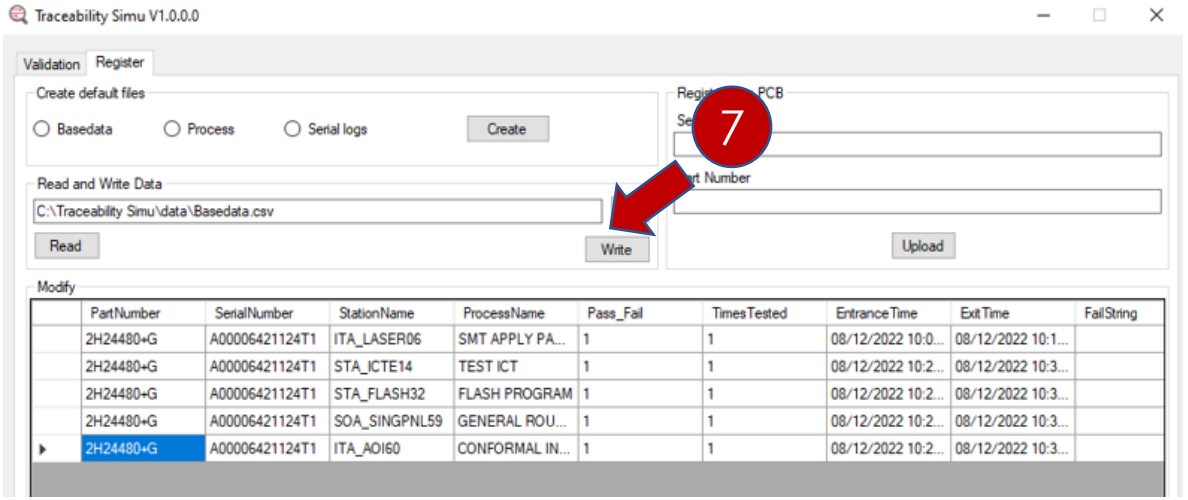




How to use

6.- Once the Basedata file address is loaded, press Read(4), delete the row of the last station with the "delete" button on the keyboard(5) and press the "Write" button(6).





Press "Refresh"(7) and the table will be updated. You can now run the test again without error in BackCheck.