



## User Manual

# ACASYA<sub>for WFC</sub>

Automatic Control and Analysis System Application for Wheel Fitted Component

September 24, 2021

---

Continental Automotive SAS  
1 Avenue Paul Ourliac  
BP 83649  
31036 TOULOUSE CEDEX 1



---

---

## TABLE OF CONTENTS

I.	ACASYA – FIRST STEPS	2
A.	Installation	2
B.	Welcome screen	5
C.	Main screen	6
II.	CREATION MODE	10
A.	Test script creation	12
1.	Create a new command	12
2.	Commands details	14
a.	SetA	14
b.	SetP	14
c.	SendLFD	14
d.	StopLFD	15
e.	SendLFCw	15
f.	StopLFCw	15
g.	LFPower	Erreur ! Signet non défini.
h.	Label	16
3.	Additional features	17
a.	Modifying the table (step positions, load existing test)	17
b.	Adding an extra end time to the test	18
4.	Saving the Test script	18
B.	Expected Results creation	19
1.	Create a new check function	19
2.	Check functions details	22
a.	CheckTimingInterFrames	22
b.	CheckTimingInterBursts	23
c.	CheckFieldValue	24
d.	CheckSTDEV	25
e.	CheckNbBursts	26
f.	CheckNbFramesInBurst	27
g.	CheckCompareP	28
h.	CheckCompareAcc	29
i.	CheckNoRF	30
j.	CheckTimingFirstRF	31

---

3.	Additional features	32
a.	Modifying the table (step positions, load existing test)	32
4.	Saving the Expected Results script	32
C.	Saving the Test Case	33
a.	Test script worksheet	33
b.	Test script worksheet	34
III.	EXECUTION MODE	37
A.	IHM	37
B.	Executing a new sequence	39
1.	Adding tests to your sequence	39
2.	Starting the execution	39
C.	Analyzing the results	40
1.	Original scripts folder	40
2.	Translated scripts folder	40
3.	Test Case folders	41
a.	Results folder	41
b.	Check_Results.txt file	41
c.	ErrorLog.txt	42
d.	Logs files	42
e.	Test Case files	42
4.	Coverage Matrix with results synthesis	42
IV.	RE-ANALYSE MODE	44
A.	IHM	44
B.	Starting an analysis	45
C.	Analysis results	45
V.	Security and monitoring	48
A.	Electric Lock and Security Loop Blue button	48
B.	Monitoring and status panel	49
VI.	APPENDIX	52
A.	Installation files	52
B.	Check Results example	53
1.	PASSED example	53
2.	FAILED example	54

---

Chapter 1

# **ACASYA – FIRST STEPS**

---

# I. ACASYA – FIRST STEPS



**ACASYA for WFC** is the brand-new version of the software formerly called Wheel Unit Testing Tool 2.0. This new version has a brand-new design and new improved features. ACASYA allows you to automate the control and analysis of the LSE test bench. In other words, you have the possibility with this application to create your own Test Case scripts, execute them and get a synthesis of the test analysis.

## A. Installation

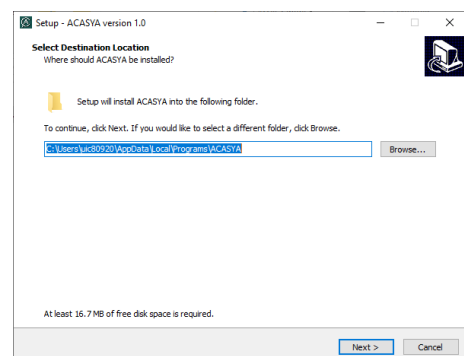
### Minimum system requirements:

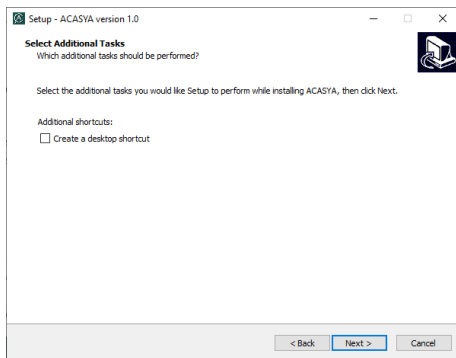
- OS: Windows 10
- Architecture: 32 or 64-Bit
- CPU: Dual-core CPU
- Memory: 4 GB RAM
- HD free space: 2 GB
- Screen resolution: 1280x1024 pixels display
- Software: Microsoft Office Excel – v. 1908 (build 11929.20934)

### Installation procedure:

Double-clicking on  **ACASYA-setup.exe** will launch the installer. Follow then the instructions.

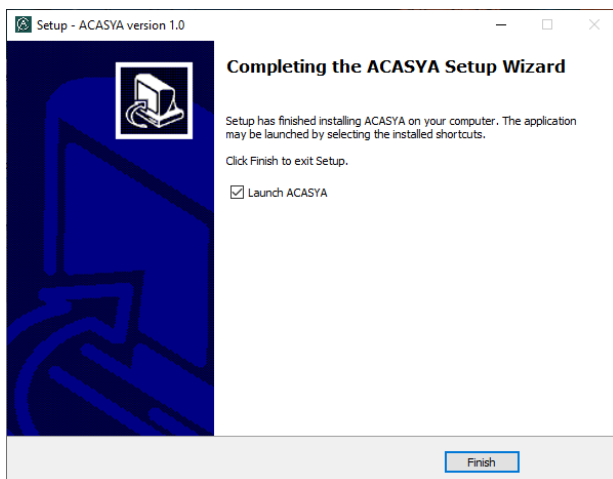
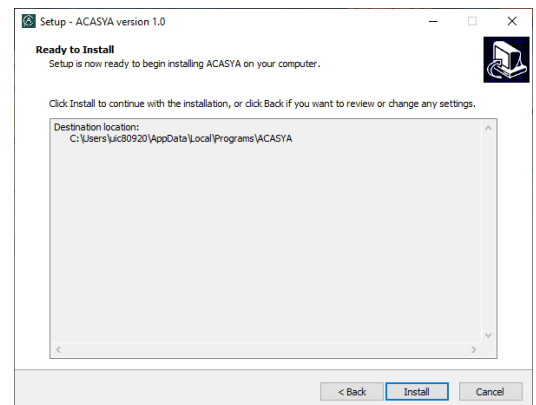
**Step 1:** Choose where to install ACASYA on your hard drive. The installation folder will contain all the necessary files for the proper using of ACASYA. See the list of installation files in the [appendix](#).





**Step 2:** The installer is able to add a shortcut on the desktop. Check the option if you need it.

**Step 3:** ACASYA is now ready to be installed. Click on **Install** to launch the installation. The operation might take few seconds.



**Step 4:** The installer informs you when the installation is complete. Check **Launch ACASYA** to immediately launch ACASYA when you click on Finish. Otherwise, click on the ACASYA icon in the Start menu or on the Desktop.

Congratulations, you have now installed ACASYA!

**Step 5:** It is recommended to do a right click on the ACASYA shortcut and click on **Properties**. Then click on **Run compatibility troubleshooter**. Follow the indication given in the windows to make sure there is no problem of compatibility and accept to save the new parameters.

**Warning:** You can have problem if your regional parameters are not good, you must select the “.” Instead of “,” for the decimal symbol in your windows parameters.





## B. Welcome screen

Once ACASYA is launched, the Welcome screen is displayed, as shown in the Figure 1. This Welcome screen indicates the status of the LSE Test Bench and allows you to create a new project.

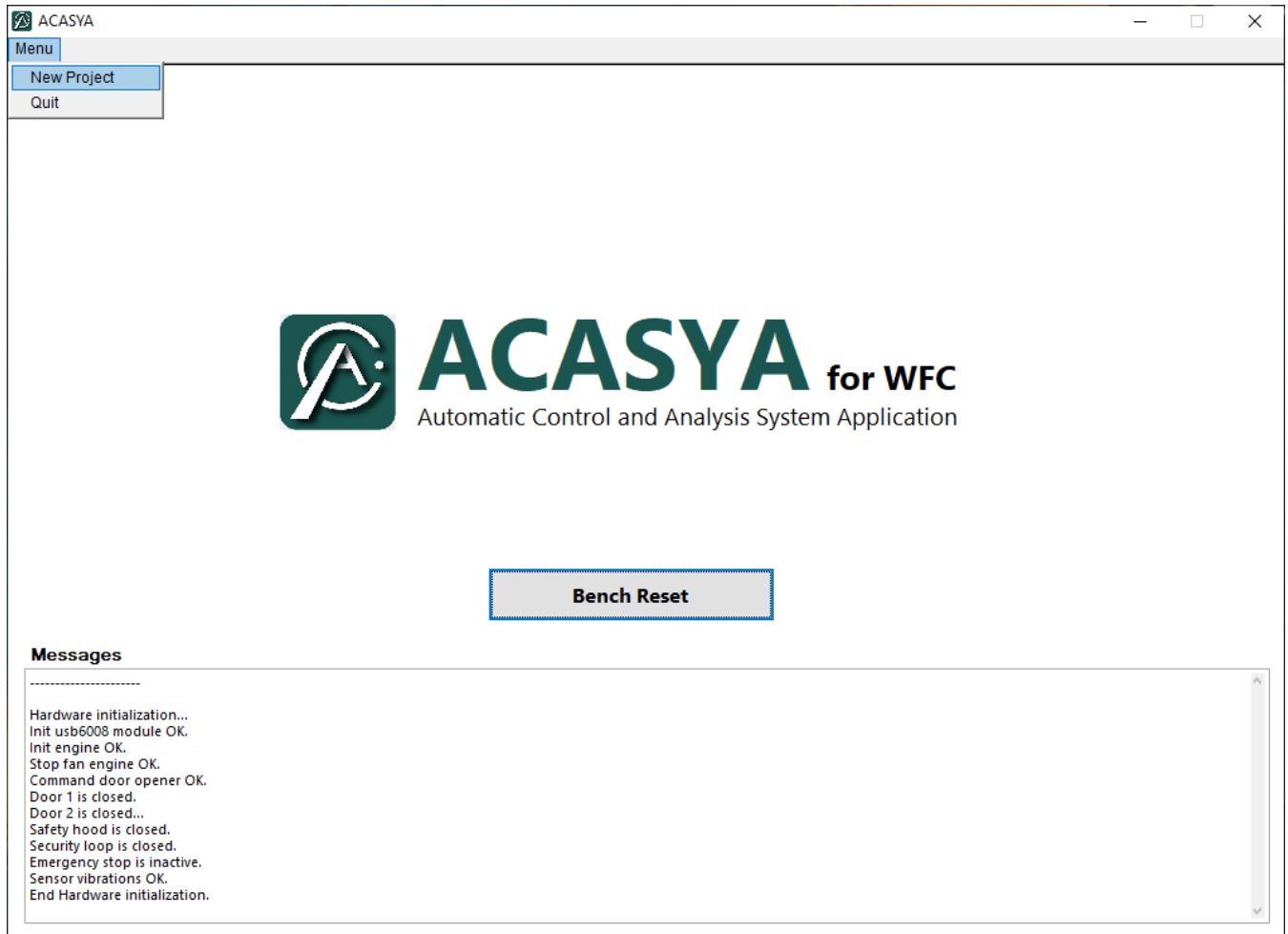


Figure 1. Welcome screen

To create a new project, click on the **Menu** tab, then click on **New Project**.

## C. Main screen

After clicking on **New Project**, the Main screen of ACASYA is displayed, as shown in the Figure 2. This window is organized as follows:

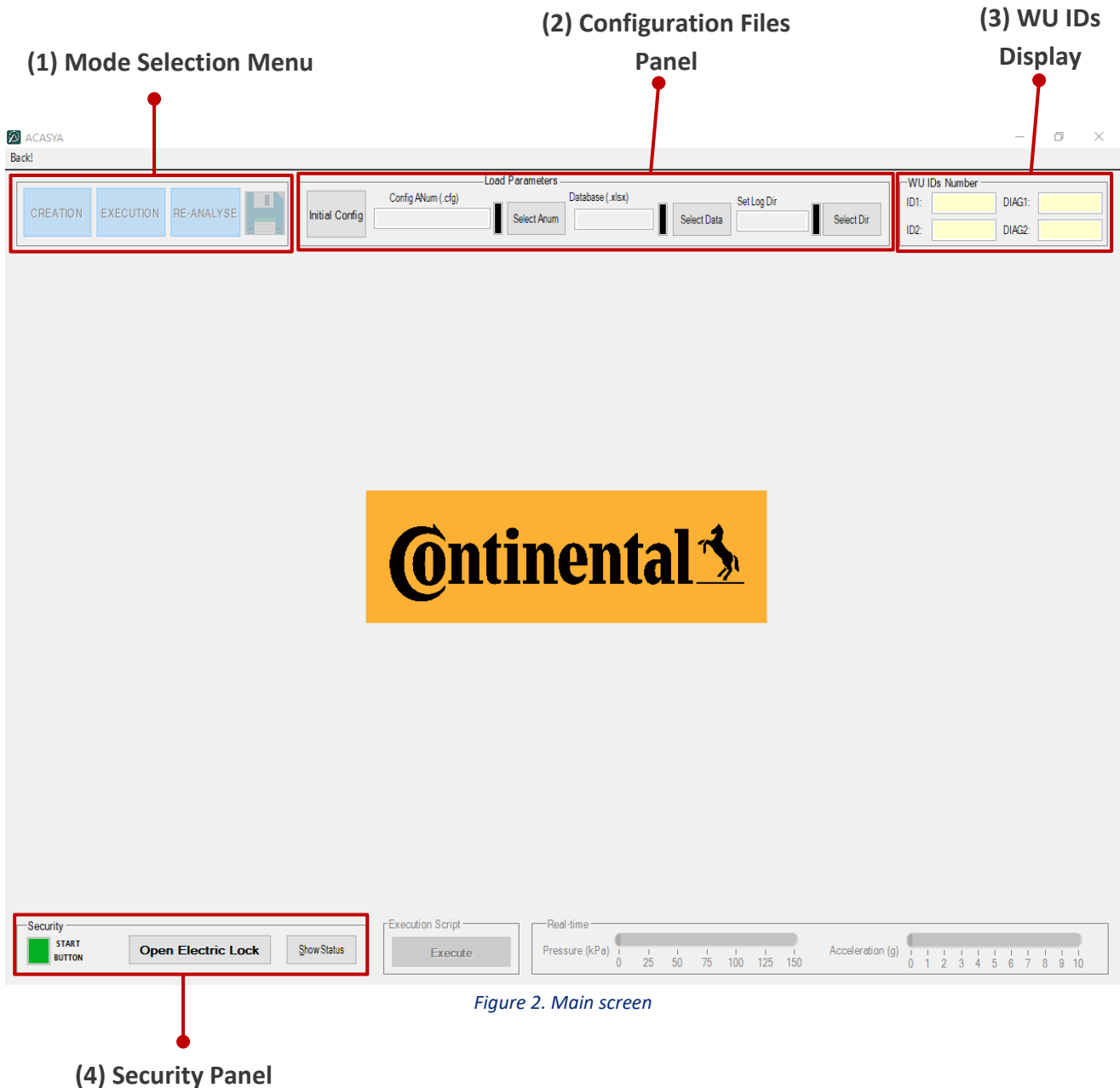


Figure 2. Main screen

**(1) Mode Selection Menu:** gives access to the 3 modes (Creation, Execution and Re-Analyse).

**(2) Configuration Files Panel:** allows to load the configurations files and working folder.

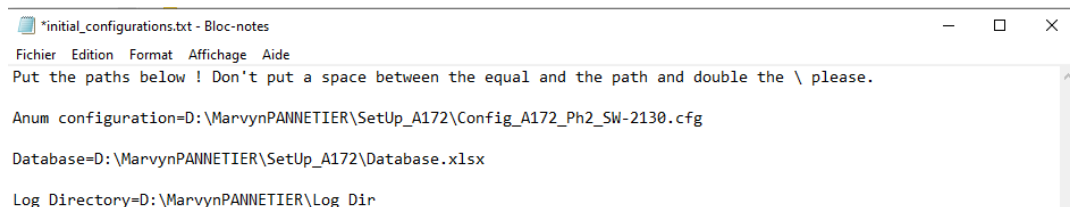
**(3) WU IDs Display:** shows the WU ID numbers of the WFC defined in the Database.

**(4) Security Panel:** indicates security status and gives access to some security tool presented in the Security and monitoring chapter.

- Before creating or executing Test Case scripts, ACASYA must be configured imperatively. The user must load the configuration files and the working folder in the [Configuration Files Panel](#) presented above. There are three different elements to load<sup>1</sup>:

- ANum configuration file (.cfg file)
- Database configuration file (.xlsx file) - Project specifics parameters file
- Storage folder for logs - a directory which will be stocked the results of the tests.

- To be more efficient and faster, a button “Initial Config” exists. This button is here to load the three elements with one click. To do that you need to fill in the file “initial\_configurations.txt” present in the ACASYA folder created by the installation, with the configuration paths like in the Figure 3.



```
*initial_configurations.txt - Bloc-notes
Fichier Edition Format Affichage Aide
Put the paths below ! Don't put a space between the equal and the path and double the \ please.

Anum configuration=D:\MarvynPANNETIER\SetUp_A172\Config_A172_Ph2_SW-2130.cfg

Database=D:\MarvynPANNETIER\SetUp_A172\Database.xlsx

Log Directory=D:\MarvynPANNETIER\Log_Dir
```

Figure 3. Configuration file

- Once this step is completed, the **Creation**, **Execution** and **Re-Analyse** buttons of the [Mode Selection Menu](#) are enabled and the WU IDs from the Database are loaded into the [WU IDs Display](#), as presented in the Figure 4. Main window after configuration.

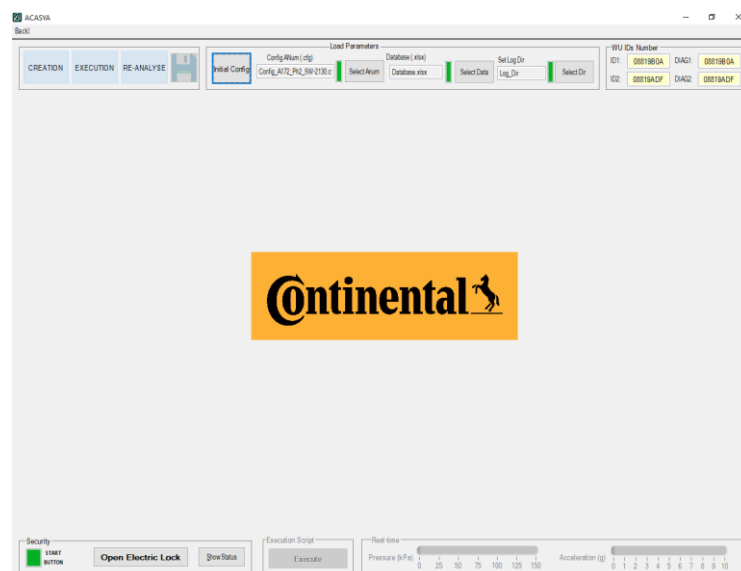


Figure 4. Main window after configuration

<sup>1</sup> The loading order of files and folder is not important.



---

Chapter 2

# CREATION MODE

---

## II. CREATION MODE

In this mode you can create the Test Case, including the Test script and the Expected Results. The main window is splitted in two parts: the one on the left side is dedicated to the creation of the Test script and the one on the right side is dedicated to the creation of the Expected Results.

(1) Test script  
creation table

ACASYA

Back!

CREATION EXECUTION RE-ANALYSE

Load Parameters

Initial Config Config ANum (.cfg) Database (.xlsx) Set Log Dir

Config\_A172\_Ph2\_SW-2130.c Database.xlsx Select Data Desktop Select Dir

WU IDs Number

ID1: 08819B0A DIAG1: 08819B0A

ID2: 08819ADF DIAG2: 08819ADF

Test Script Creation

Time	Duration	Function	Value	Nb Frame	WU ID	Interframe	Parameters	Comment
------	----------	----------	-------	----------	-------	------------	------------	---------

Add end time : 0.00

Expected Results Creation

Command Check	FC Value	Labels	Interface	Parameters	Value	Tolerance	Tolerance %
---------------	----------	--------	-----------	------------	-------	-----------	-------------

Security

START BUTTON Open Electric Lock Show Status

Execution Script

Execute

Real-time

Pressure (kPa) 0 25 50 75 100 125 150

Acceleration (g) 0 1 2 3 4 5 6 7 8 9 10

Figure 5. Creation Mode main window

(3) Expected Results script  
creation table

(1) **Test script creation table:** table to create the Test script (SetA, SetP, SendLFD...).

(2) **Expected Results script creation table:** table to create the Expected Results script (with check functions for the analysis).

---

**(3) Add end time text box:** text box to add some extra time to your Test script.

# A. Test script creation

Located at the left side of the window, this panel allows you to create the Test script. The Test script contains all the commands (set acceleration, set pressure, send LFD...) that will be executed by the bench during the test.

## 1. Create a new command

Create Steps
Insert above
Modify
Move Up
Move Down
Delete Step
Delete All
Load Test Script

A right click on the table gives you access to several features.  
Click on **Creates Steps** to add a step to your table.

A new window titled “Script definition” appears, as presented above in the 6.

Script definition

Functions : 

Select...1

Pre Condition

Script2

Post Condition

Time (s) :

Duration (s) :

Pressure (KPa) :

Acceleration (g) :

Label name :

LF Data

Name : 

Select...

Nb Frame : 

Continuous

Interframe (ms) : 

10

Power (%) : 

20

Parameter

Select parameter

Name	Value
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

Insert Parameter(s) from Database

Parameters : 

Select...

Value :

Unit :

Formula :

= 

0.00000

(

)

DEL

/

7

8

4

x

4

5

6

-

1

2

3

+

0

.

INSERT

Add

5

Exit

Figure 6. Script definition window

The window is divided as follow:

12 | Creation Mode



- (1)

**Function selection:** this contextual menu allows you to choose the command to be executed: (SetA, SetP, SendLFD, StopLFD, Label...). The various commands are detailed in the [Commands details](#) section.
- (2)

**Type of command:** this panel allows you to select the type of command. Three types are defined:
  - **Pre condition:** command to be executed before the test.
  - **Script:** command to be executed during the test.
  - **Post condition:** command to be executed after the test.
- (3)

**Parameters edition:** this panel allows you to configure all parameters linked with the command selected. The parameters are dimmed or not, according to the selected command. To enter a new parameter value, you must use the numeric keypad, presented in section 4.
- (4)

**Numeric keypad:** the numeric keypad, presented below in the Figure 7, allows you to enter the value of a parameter (Time, Duration, Pressure...).

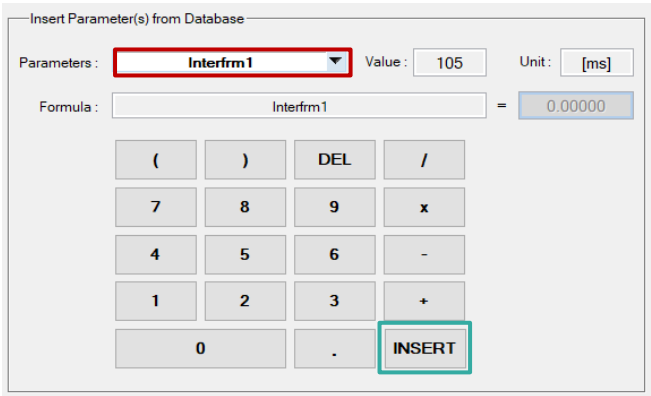


Figure 7. Numeric keypad for Test script Creation

The value can be a numeric value (e.g. “102”, “10”) or a formula using parameters from the Database. You can choose a specific parameter from the Database clicking on the contextual menu titled “**Parameters**” The unit and the value are displayed next to the parameter’s name. To validate the parameter value, click on the **INSERT** button.

- (5)

**Add and Exit buttons:** once the command has been detailed, you can add it to the table using the button **Add**. A new line like this one is added to the table:

Time	Duration	Function	Value	Nb Frame	WU ID	Interframe	Parameters	Comment
1		SendLFD	LFD_Force_RF_USER1	Continuous		10		Script

You can quit the “Script definition” window at any time with the button **Exit**.

## 2. Commands details

As we have seen above, the [Script definition window](#) allows you to configure various commands. The Figures below detail some important information about commands.

### a. SetA

Command	SetA
Interface	Bench Test
Mandatory parameters	Time [s] Duration [s] Acceleration value [g]
Description	Sets the test bench acceleration to a given value. Starts at time <b>Time</b> and produces a ramp from an initial acceleration value to the <b>Acceleration value</b> during <b>Duration</b> seconds.

### b. SetP

Command	SetP
Interface	Bench Test
Mandatory parameters	Time [s] Duration [s] Pressure value [kPa]
Description	Sets the test bench pressure to a given value. Starts at time <b>Time</b> and produces a ramp from an initial pressure value to the <b>Pressure value</b> during <b>Duration</b> seconds.

### c. SendLFD

Command	SendLFD
Interface	ANum
Mandatory parameters	Time [s] LF Data (Name, Nb Frame, Interframe) LFPower [%] WU ID and other parameters if needed
Description	Sends one or several <b>LF Data</b> frames to specific <b>WU ID(s)</b> with a signal strength at <b>LFPower</b> %. Starts at time <b>Time</b> and must be stopped with <b>StopLFD</b> command.

#### d. StopLFD

<b>Command</b>	<b>StopLFD</b>
<b>Interface</b>	ANum
<b>Mandatory parameters</b>	Time [s]
<b>Description</b>	Stops sending all LF Data frames at time <i>Time</i> .

#### e. SendLFCw

<b>Command</b>	<b>SendLFCw</b>
<b>Interface</b>	ANum
<b>Mandatory parameters</b>	Time [s] Duration set at 1000ms automatically
<b>Description</b>	Sends one or several <i>LF Carrier wave</i> frames to specific <i>WU ID(s)</i> with a signal strength at <i>LFPower</i> %. Starts at time <i>Time</i> and must be stopped with <b>StopLFCw</b> command.

#### f. StopLFCw

<b>Command</b>	<b>StopLFCw</b>
<b>Interface</b>	ANum
<b>Mandatory parameters</b>	Time [s]
<b>Description</b>	Stops sending all LF Carrier wave frames at time <i>Time</i> .

---

g. Label

Command	Label <sup>1</sup>
Interface	
Mandatory parameters	Time [s] Label name
Description	Delimits the period of analysis. E.g. From the <b><i>Label 1</i></b> at <b><i>Time 1</i></b> to the <b><i>Label 2</i></b> at <b><i>Time 2</i></b> .

---

<sup>1</sup> Label is not a function that will command the test bench or ANum. Labels are tags necessary to analyze the test.

### 3. Additional features

Once all the steps have been added, you get a table like the Figure 4, and several features are available.

Time	Duration	Function	Value	Nb Frame	WU ID	Interframe	Parameters	Comment
1		SendLFD	LFD_Force_RF_USER1	Continuous		10		Script
1		start						Script
1	1	SetA	100[]					Script
20	1	SetA	70[]					Script
40	1	SetA	40[]					Script
60	1	SetA	10[]					Script
80	1	SetA	100[]					Script
100	80	SetA	10[]					Script
180		StopLFD						Script
180		end						Script

Figure 4. Example of a Test Script table

#### a. Modifying the table (step positions, load existing test)

As we have seen in the [Create a new command](#) part, a right click on the table gives you access to additional features:

Create Steps
Insert above
Modify
Move Up
Move Down
Delete Step
Delete All
Load Test Script

- *Move Up* and *Move Down*: change the position of the selected step<sup>1</sup>.
- *Delete Step*: delete the selected step.
- *Delete All*: delete all steps of the table.
- *Load Test Script*: load the test script from an existing Test Case (.xlsx).

<sup>1</sup> Changing the position of the step does not affect the test execution order. Only the time indicated in the **Time** field is considered.

## b. Adding an extra end time to the test

Moreover, you can add some extra time to your test in **the area shown below**, because sometimes, the WFC continue to send some frames after the end of the test script.



The test will not end at the time indicated in this field, but at the total time of the script plus the extra time.

## 4. Saving the Test script

Once the Test script is created, you can create the Expected Results script too or save only the Test script. In this case, the Expected Results section will be empty.

To save the Test script, check the [Saving the Test Case](#) section.



---

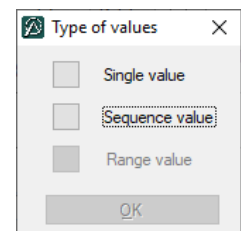
The window is divided as follow:

- (1) **Check functions selection:** this contextual menu allows you to choose the check function: (CheckTimingInterBursts, CheckTimingInterFrames, CheckFieldValue...). The various check functions are detailed in the [Check functions details](#) section.
- (2) **Function Code Value:** this text field allows you to indicate the Function Code value. The received frames contain a number named Function Code, designating the mode in which the sensor is. By indicating the Function Code value in this text field, you will filter the frames received by keeping only the frames with this value. To enter a value, you need to use the numeric keypad presented below.
- (3) **Labels:** this panel all allows you to indicate the two labels<sup>1</sup> between which the analysis will be done. The names of the labels must correspond to those defined in the Test script. You can choose to put only one label or even zero and the test will be launch between the beginning (no label 1) and the end (no label 2) of the test case.
- (4) **Interface and field to check:** this panel allows you to indicate the RF frame format (Standard\_frm, Diag\_frm ...) and the field to check.  
The frames received during a test may have a different interface<sup>2</sup>. This “Interface to check” contextual menu is used to filter the frames received according to the chosen interface.  
The “Field to check” contextual menu is used to check the existence of a parameter in the log file during the analysis. The parameter is specific to each interface.

**Note:** when you configure the CheckFieldValue function, this panel allows you to choose the parameter to check.

- (5) **Value and tolerances:** this panel allows you to indicate the expected value (timing, acceleration pressure, function code value...) and their its tolerance for the check function. The value depends of each check function.

To enter a value, you must click on the Value field and first choose the type of value (Single, Sequence or Range value) as presented here.  
Then, you need to use the numeric keypad presented below.



To enter a tolerance<sup>3</sup> (value or percentage), you need to use the [numeric keypad](#) too.

---

<sup>1</sup> The CheckTimingFirstRF functions needs only one label (Label 1).

<sup>2</sup> The interfaces are defined in the Database.

<sup>3</sup> You can search for the exact value indicating a 0 value in the tolerance field.



The unit must be the same for the value and tolerance.

- (6) **Numeric keypad:** the numeric keypad, shown below in the Figure 6, allows you to enter the value of your choice (Function Code Value, Value, Tolerances...).

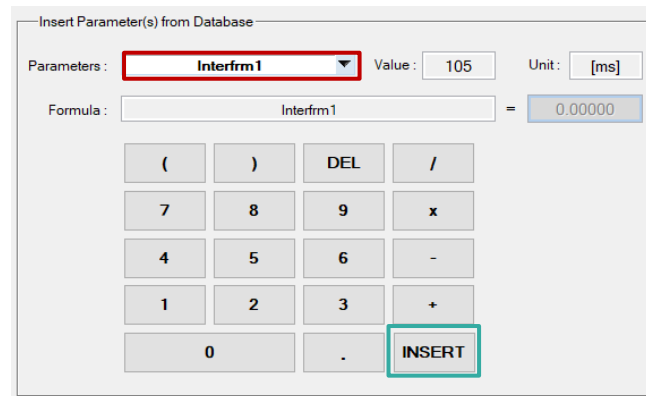


Figure 6. Numeric keypad for Expected Results creation

The value can be a numeric value (e.g. "102", "10") or a formula using parameters from the Database. You can choose a specific parameter from the Database clicking on the contextual menu titled "**Parameters**". The unit and the value are displayed next to the parameter's name.

To validate the parameter value, click on the **INSERT** button.

- (7) **Add and Exit buttons:** once the check function has been detailed, you can add it to the table using the button **Add**. A new line like shown below is added to the table:

Command Check	FC Value	Labels	Interface	Parameters	Value	Tolerance	Tolerance %
CheckCompareP		start   stop	Stand_Frm	WU_State		50;	0;

You can quit the "Expected results" window at any time with the button **Exit**.

## 2. Check functions details

As we have seen above, the Expected results window allows you to configure various Check functions. The Figures below detail some important information.

### a. CheckTimingInterFrames

**Functions**  
CheckTimingInterFrames

Function Code Value :

Label 1 :   
Label 2 :

Select interface to check :   
Field to check :   
Attribute Name:

Value  Unit [s]   
Type of value:  
Fixed (eg: 10)    Sequence (eg: 10;15;17)    Range (eg: 0:15)

☒ Tolerance value :    
☐ Tolerance (%) :

Purpose		
This function allows to check the timing between two frames.		
Mandatory fields	Type of values	PASSED conditions
Function Code value Label 1 and Label 2 Interface Value Tolerance (value or percentage)	Unit: [s]  Fixed Range	Pre-conditions: - Function Code  All the interframes between <b>[Value +/- Tolerance]</b> .
Analysis		
First, the function filters the received frames keeping only those between the two labels and with an interface <sup>1</sup> and a FC value identical to those given as parameter. Then, the function checks, for every burst <sup>2</sup> , if all timings between frames belong to the interval defined by the value(s) given as parameter, plus/minus tolerance.		

<sup>1</sup> The function checks if the “Field to check” linked to the Interface is an existing column in the log file.

<sup>2</sup> Two frames belong to the same burst if their interframe timing is below the threshold time (*Interframe/Interburst Threshold* value stored in the Database).

## b. CheckTimingInterBursts

**Functions**

Function Code Value :

Label 1 : 
  
 Label 2 :

Select interface to check : 
  
 Field to check : 
  
 Attribute Name:

Value  Unit [s] 
  
 Type of value:
   
☐ Fixed (eg: 10)
 ☐ Sequence (eg: 10;15;17)
 ☐ Range (eg: 0:15)

☒ Tolerance value :  
  
☐ Tolerance (%) :

### Purpose

This function allows to check the timing between two bursts.

### Mandatory fields

Function Code value  
 Label 1 and Label 2  
 Interface  
 Value  
 Tolerance (value or percentage)

### Type of values

Unit: [s]  
 Fixed

### PASSED conditions

Pre-conditions:  
 - Function Code  
  
 All the interbursts between [Value +/- Tolerance].

### Analysis

First, the function filters the received frames keeping only those between the two labels and with an interface<sup>1</sup> and a FC value identical to those given as parameter.

Then, the function checks if all timings between bursts<sup>2</sup> belong to the interval defined by the value(s) given as parameter, plus/minus tolerance.

<sup>1</sup> The function checks if the "Field to check" linked to the Interface is an existing column in the log file.

<sup>2</sup> Two frames do not belong to the same burst if their interframe timing is higher than the threshold time (*Interframe/Interburst Threshold* value stored in the Database).

## c. CheckFieldValue

**Functions**  
 CheckFieldValue

Function Code Value :

Label 1 :  
 Label 2 :

Select interface to check :  
 Select...

Field to check :  
 Select...

Attribute Name:

Value  
 Unit  
 [s,g,kPa...] Delete

Type of value:  
 Fixed (eg: 10)    Sequence (eg: 10;15;17)    Range (eg: 0:15)

☒ Tolerance value : 0 Delete

☐ Tolerance (%) : 0 Delete

### Purpose

This function allows to check the value of a specific field.

Mandatory fields	Type of values	PASSED conditions
Label 1 and Label 2 Interface Value Tolerance (value or percentage)	Unit: [s, g, kPa...]  Fixed Sequence Range	All the field values between [Value +/- Tolerance].

### Analysis

First, the function filters the received frames keeping only those between the two labels.

Then, the function checks for all frames with the selected interface, if the received value of the “Field to check” given as parameter belongs to the interval defined by the value(s) given as parameter, plus/minus tolerance.

## d. CheckSTDEV

**Functions**  
 CheckSTDEV

Function Code Value :

Label 1 :  
 Label 2 :

Select interface to check :  
 Select...

Field to check :  
 Select...

Attribute Name:

Value  
 Unit  
 [degree] Delete  
 Type of value:  
 Fixed (eg: 10)    Sequence (eg: 10;15;17)    Range (eg: 0:15)

☐ Tolerance value : 0 Delete  
☐ Tolerance (%) : 0 Delete

Purpose		
This function allows to check the efficiency of the LSE feature.		
Mandatory fields	Type of values	PASSED conditions
Function Code value Label 1 and Label 2 Interface Value	Unit: [degree]  Range	Pre-conditions: <ul style="list-style-type: none"> <li>- Function Code</li> <li>- Number of populations</li> </ul> Standard deviation * 3 between [Range value].
Analysis		
<p>First, the function filters the received frames keeping only those</p> <ul style="list-style-type: none"> <li>- between the two labels</li> <li>- with an interface<sup>1</sup> and a FC value identical to those given as parameter</li> <li>- corresponding to the first frame<sup>2</sup> of the burst</li> <li>- with an <i>Angle_detection</i> field equal to 1.</li> </ul> <p>Then, the function checks if the number of populations is equal to the one expected<sup>3</sup> and checks if every population has a standard deviation multiply by three between the range given as parameter.</p> <p>Finally, the function checks if the merged population<sup>4</sup> has a standard deviation multiply by three belongs to the interval defined by the range given as parameter. If the number of synchronized frames is below than 9, the standard deviation will be calculated but the status will be FAILED.</p>		

<sup>1</sup> The function checks if the "Field to check" linked to the Interface is an existing column in the log file.

<sup>2</sup> The number of the first frame corresponds to the *Frame\_number* field and must be equal to the *Frame\_number* value stored in the Database.

<sup>3</sup> The number of populations expected is stored in the Database (*Number of emission angles for LSE*).

<sup>4</sup> The n populations are plotted on a single area to check the orthogonality of the populations (using mod 360).

## e. CheckNbBursts

**Functions**
  
CheckNbBurst

Function Code Value :

Label 1 :
  
Label 2 :

Select interface to check :
  
Select...

Field to check :
  
Select...

Attribute Name:

Value
  
Unit
  
bursts
  
Delete

Type of value:
  
Fixed (eg: 10)    Sequence (eg: 10;15;17)    Range (eg: 0:15)

☐ Tolerance value : 0
  
Delete

☐ Tolerance (%) : 0
  
Delete

### Purpose

This function allows to check the number of bursts sent during the test.

Mandatory fields	Type of values	PASSED conditions
Function Code value Label 1 and Label 2 Interface Value	Unit: [number]  Fixed	Pre-conditions: - Function Code  Number of bursts = <b>Value</b> .

### Analysis

First, the function filters the received frames keeping only those

- between the two labels
- with an interface<sup>1</sup> and a FC value identical to those given as parameter.

Then, the function counts the number of bursts<sup>2</sup> and compares the number with the value given as parameter. The number of bursts counted must be exactly equal to the value given as parameter.

<sup>1</sup> The function checks if the "Field to check" linked to the Interface is an existing column in the log file.

<sup>2</sup> Two frames do not belong to the same burst if their interframe timing is higher than the threshold time (*Interframe/Interburst Threshold* value stored in the Database).

## f. CheckNbFramesInBurst

**Functions**  
 CheckNbFramesInBurst

Function Code Value :

Label 1 :  
 Label 2 :

Select interface to check :  
 Select...

Field to check :  
 Select...

Attribute Name:

Value  
 Unit  
 frames

Delete

Type of value:  
 Fixed (eg: 10)    Sequence (eg: 10;15;17)    Range (eg: 0:15)

☐ Tolerance value : 0    Delete

☐ Tolerance (%) : 0    Delete

### Purpose

This function allows to check the number of frames in the first burst sent.

Mandatory fields	Type of values	PASSED conditions
Function Code value Label 1 and Label 2 Interface Value	Unit: [number]  Fixed	Pre-conditions: - Function Code  Number of frames in first burst = <b>Value</b> .

### Analysis

First, the function filters the received frames keeping only those

- between the two labels
- with an interface<sup>1</sup> and a FC value identical to those given as parameter.

Then, the function counts the number of frames in the first burst<sup>2</sup> and compares the number with the value given as parameter. The number of frames counted must be exactly equal to the value given as parameter.

<sup>1</sup> The function checks if the "Field to check" linked to the Interface is an existing column in the log file.

<sup>2</sup> Two frames belong to the same burst if their interframe timing is below the threshold time (*Interframe/Interburst Threshold* value stored in the Database).

## g. CheckCompareP

**Functions**  
 CheckCompareP

Function Code Value :

Label 1 :  
 Label 2 :

Select interface to check :  
 Select...

Field to check :  
 Select...

Attribute Name:

Value  
 Unit  
 Delete

Type of value:  
 Fixed (eg: 10)    Sequence (eg: 10;15;17)    Range (eg: 0:15)

☒ Tolerance value : 0 Delete

☐ Tolerance (%) : 0 Delete

### Purpose

This function allows to check the accuracy of the pressure values sent by the WFC.

### Mandatory fields

Label 1 and Label 2  
 Interface  
 Tolerance (value or percentage)

### Type of values

### PASSED conditions

All the pressure values between [**Value** +/- **Tolerance**].

### Analysis

First, the function filters the received frames keeping only those

- between the two labels
- with an interface<sup>1</sup> identical to the one given as parameter.

Then, the function checks for every frame if the pressure value sent by the WFC<sup>2</sup> (from the *Pressure* field in log file) belongs to the interval defined by the reference pressure<sup>3</sup> value acquired by the bench plus/minus tolerance.

<sup>1</sup> The function checks if the “Field to check” linked to the Interface is an existing column in the log file.

<sup>2</sup> The pressure value captured by the sensor is stored in the *Pressure* field from the log file.

<sup>3</sup> The reference pressure value is calculated by interpolation between two other known pressure values (from the bench) in order to guess the pressure value at the emission time of the frame.



## h. CheckCompareAcc

**Functions:**  
 CheckCompareAcc

Function Code Value : 
 

**RIM Diameter**
 [inch]

Label 1 :   
 Label 2 :

Select interface to check : 

Select...

  
 Field to check : 

Select...

  
 Attribute Name:

Value  Unit 

Delete

  
 Type of value:  
 Fixed (eg: 10)    Sequence (eg: 10;15;17)    Range (eg: 0:15)

☒ Tolerance value : 

0

Delete

  
☐ Tolerance (%) : 

0

Delete

### Purpose

This function allows to check the accuracy of the acceleration values captured and sent by the WFC.

### Mandatory fields

Label 1 and Label 2  
 Interface  
 Tolerance (value or percentage)  
**RIM Diameter**

### Type of values

### PASSED conditions

All the acceleration values between [**Value** +/- **Tolerance**].

### Analysis

First, the function filters the received frames keeping only those

- between the two labels
- with an interface<sup>1</sup> identical to the one given as parameter.

Then, the function checks for every frame if the acceleration value sent by the WFC<sup>2</sup> belongs to the interval defined by the reference acceleration<sup>3</sup> value acquired by the bench plus/minus tolerance.

<sup>1</sup> The function checks if the "Field to check" linked to the Interface is an existing column in the log file.

<sup>2</sup> The angular speed value captured by the WFC is stored in the *ASpeed* field from the log file and converted into acceleration (g) thanks to the **RIM diameter** given as parameter.

<sup>3</sup> The reference pressure value is stored in the *Acceleration* field from the log file.

## i. CheckNoRF

**Functions**  
 CheckNoRF

Function Code Value :

Label 1 :  
 Label 2 :

Select interface to check :  
 Select...

Field to check :  
 Select...

Attribute Name:

Value  
 Unit  
 Delete

Type of value:  
 Fixed (eg: 10)    Sequence (eg: 10;15;17)    Range (eg: 0:15)

☐ Tolerance value : 0 Delete

☐ Tolerance (%) : 0 Delete

Purpose		
This function allows to check the accuracy of the pressure values captured and sent by the WFC.		
Mandatory fields	Type of values	PASSED conditions
Label 1 and Label 2 Interface		No frame received between <b>Labels</b> .
Analysis		
<p>The function checks the absence of frame with the interface<sup>1</sup> given as parameter between labels.</p> <p>Note: the function does not detect frames with an interface other than the one indicated as parameter.</p>		

<sup>1</sup> The function checks if the “Field to check” linked to the Interface is an existing column in the log file.

## j. CheckTimingFirstRF

**Functions**  
 Check TimingFirstRF

Function Code Value :

Label 1 :  
 Label 2 :

Select interface to check :  
 Select...

Field to check :  
 Select...

Attribute Name:

Value  
 Type of value:  
 Fixed (eg: 10)    Sequence (eg: 10;15;17)    Range (eg: 0:15)

Unit  
 [s]    Delete

☒ Tolerance value : 0    Delete

☐ Tolerance (%) : 0    Delete

### Purpose

This function allows to check the timing to receive the first RF.

Mandatory fields	Type of values	PASSED conditions
Function Code value Label 1 Interface Value	Unit: [s]  Range	Pre-conditions: - Function Code  Timing "Label 1 - first RF" +/- <b>Tolerance</b> between <b>[Range Value]</b>

### Analysis

First, the function filters the received frames keeping only those with an interface<sup>1</sup> and a FC value identical to those given as parameter.

Then, the function checks if the timing between the Label 1 and the first RF belongs to the interval defined by the range given as parameter.

<sup>1</sup> The function checks if the "Field to check" linked to the Interface is an existing column in the log file.

### 3. Additional features

Once all the steps have been added, you get a table like the Figure 7, and several features are available.

Expected Results Creation							
Command Check	FC Value	Labels	Interface	Parameters	Value	Tolerance	Tolerance %
CheckCompareAcc		in   out;	Diag_Fm	WU_State	18;	10;	0;

Figure 7. Example of a Test Script table

a. Modifying the table (step positions, load existing test)

As we have seen in the [Create a new check function](#) section, a right click on the table gives you access to additional features:

Create Check

Insert above

Modify

Move Up

Move Down

Delete Check

Delete All

Load Expected Results

- *Move Up and Move Down: changes the position of the check function<sup>1</sup>.*
- *Insert above: insert a new line above the mouse selector*
- *Modify: allows to modify a line*
- *Delete Step: deletes the selected check function.*
- *Delete All: deletes all check functions of the table.*
- *Load Expected Results: loads the Expected Results script from an existing Test Case (.xlsx).*

### 4. Saving the Expected Results script

To save the Expected Results script, check the [Saving the Test Case](#) section.

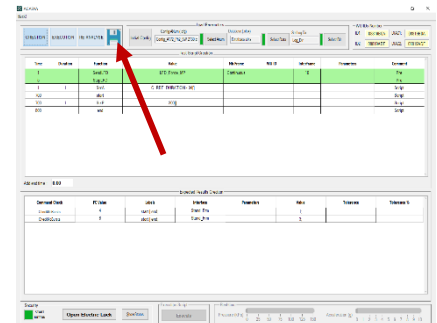
<sup>1</sup> Changing the position of the step does not affect the test execution order. Only the time indicated in the **Time** field is considered.

## C. Saving the Test Case

As a reminder, the Test Case contains the Test script and the Expected Results script.

To save your Test Case, click on the **Save Button** located in the **Mode Selection panel** at the top of your window. Then, a new window allows you to choose where to save the Test Case file.

**Note:** you can save a Test Case with an empty Expected Results table.



	A	B	C	D	E	F	G	H	I	J	K
1	Time	Duration	Function	Value	Nb Frame	WU ID	Interframe	Parameter	Comment	7 (nb of rows)	
2	1		SendLFD	FD_Force_M	Continuous		10		Pre	0 (time to add)	
3	5		StopLFD						Pre		
4	1	1	SetA	E_DURATION - 34[]					Script		
5	700		start						Script		
6	700	1	SetP	200[]					Script		
7	800		end						Script		
8	0		END								

Saving the Test Case generates a **.xlsx file**, an **Excel table**. This Excel workbook contains two worksheets, one dedicated to the Test script and the other to Expected Results script.

### a. Test script worksheet

The worksheet dedicated to the Test script is divided into 3 parts as shown below in the Figure 8:

	A	B	C	D	E	F	G	H	I	J	K
1	Time	Duration	Function	Value	Nb Frame	WU ID	Interframe	Parameter	Comment	7 (nb of rows)	
2	1		SendLFD	FD_Force_M	Continuous		10		Pre	0 (time to add)	
3	5		StopLFD						Pre		
4	1	1	SetA	E_DURATION - 34[]					Script		
5	700		start						Script		
6	700	1	SetP	200[]					Script		
7	800		end						Script		
8	0		END								

Figure 8. Test script saved in Excel

### Body of Test Script

These lines contain all the commands of the test previously created. The parameters are still present and will be translated during the execution.

### End command

This line is always located after the last command of the test. The END is a tag that indicates the end of the test script. The value in the *Time* column indicates the total execution time of the test. Initially, this

value is at zero or at the value indicated in the [Add end time](#) field. The final value is updated during the execution (at the very beginning).

**Number of steps**

This cell contains the number of rows minus the header line. It is useful during the Test Case execution.

**b. Test script worksheet**

The worksheet dedicated to the Test script is divided into 2 parts as shown below in the Figure 9:

	A	B	C	D	E	F	G	H	I
1	Command Check	FC Value	Labels	Interface	Parameters	Value	Tolerance	Tolerance%	2
2	CheckCompareAcc		in   out;			18;	Acc_tol;	0;	

Figure 9. Expected Results script saved in Excel

**Body of Test Script**

These lines contain all the commands of the test previously created. The parameters are still present and will be translated during the execution.

**Number of steps**

This cell contains the number of rows. It is useful during the Test Case execution.



---

Chapter 3

# EXECUTION MODE

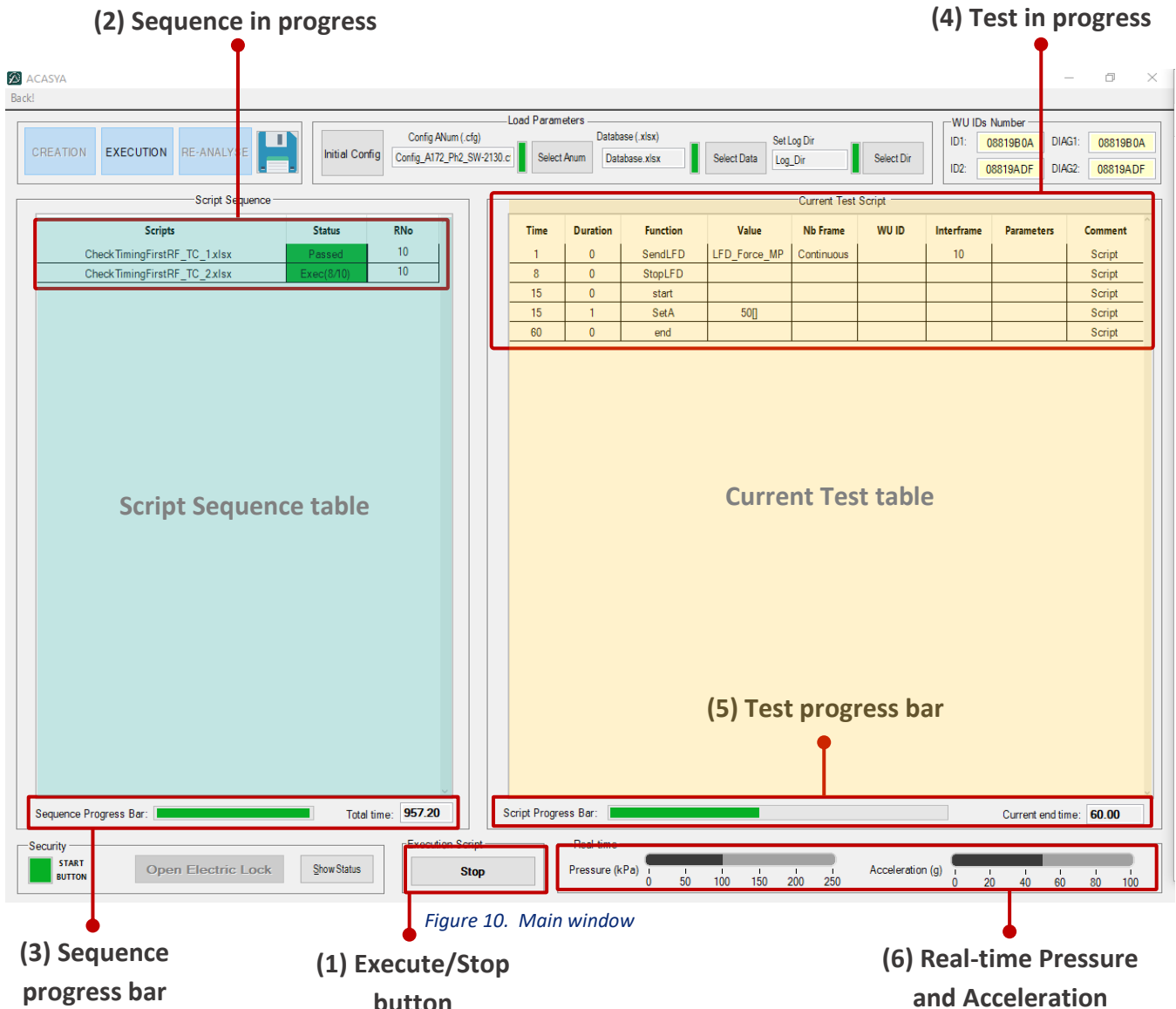
---



## III. EXECUTION MODE

### A. IHM

In this mode you can execute a sequence of tests previously created. The window is organized as follow:



(1) **Execute/Stop button:** button to start the test execution or to stop it at any time.

(2) **Sequence in progress:** table of the running sequence.

(3) **Sequence progress bar:** visual indicator to know the sequence progression.

(4) **Test in progress:** table of the running test script.

(5) **Test progress bar:** visual indicator to know the test progression.

---

**(6) Real-time Pressure and Acceleration:** visual indicator to know the Pressure and Acceleration values in real-time during the test execution.

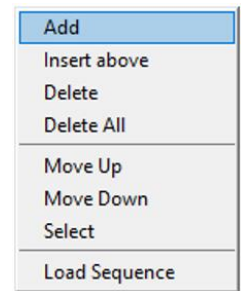
## B. Executing a new sequence

### 1. Adding tests to your sequence

A right click on the [Script Sequence Table](#) gives you access to several features. Click on **Add** to add a Test Case to your table.

A new window appears and allows you to choose one or several Test Cases.

**Note:** right clicking on the [Script Sequence Table](#) allows you to delete tests or to load an existing sequence (previously saved).

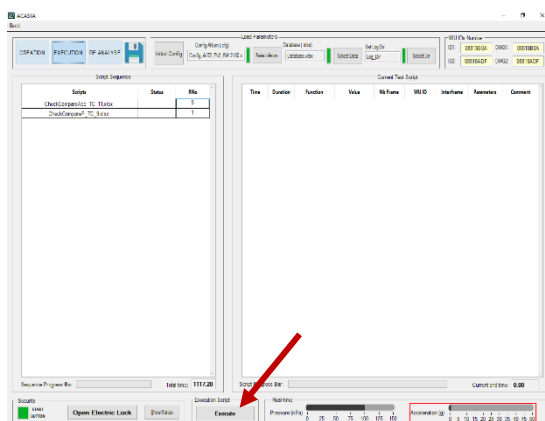
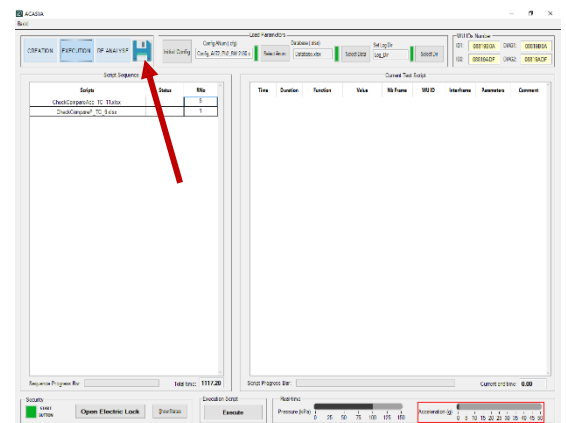


### 2. Starting the execution

**Step 1:** Once the tests are loaded, you must save the sequence in a .txt file.

To save your sequence, click on the **Save Button** located in the **Mode Selection panel** at the top of your window. Then, a new window allows you to choose where to save the sequence.

**Note:** the application always checks if you have saved the sequence at the beginning of the execution.



**Step 2:** Now, you can start your sequence.

To start your sequence, click on the [Execution Button](#) located at the bottom of your window.

A picture of a timer should appear and indicates that the execution is running.




First of all, ACASYA starts to translate every test loaded. This first step allows to translate a Test Case with parameters into a Test Case<sup>1</sup> with only values.

Then, Test Case scripts are executed and displayed in the [Current Test Table](#) one by one until the last.

<sup>1</sup> The translated Test Case scripts are stored in a folder named “\_ Translated scripts” inside the sequence folder.

## C. Analyzing the results

During the execution, ACASYA creates a sequence folder where tests, configuration files and results are stored. The sequence folder is stamped with the date and time, as presented here:  Seq\_28-08-20\_11-13-46

Once the execution is complete, the sequence folder contains several files and subfolders as shown in the Figure 11:

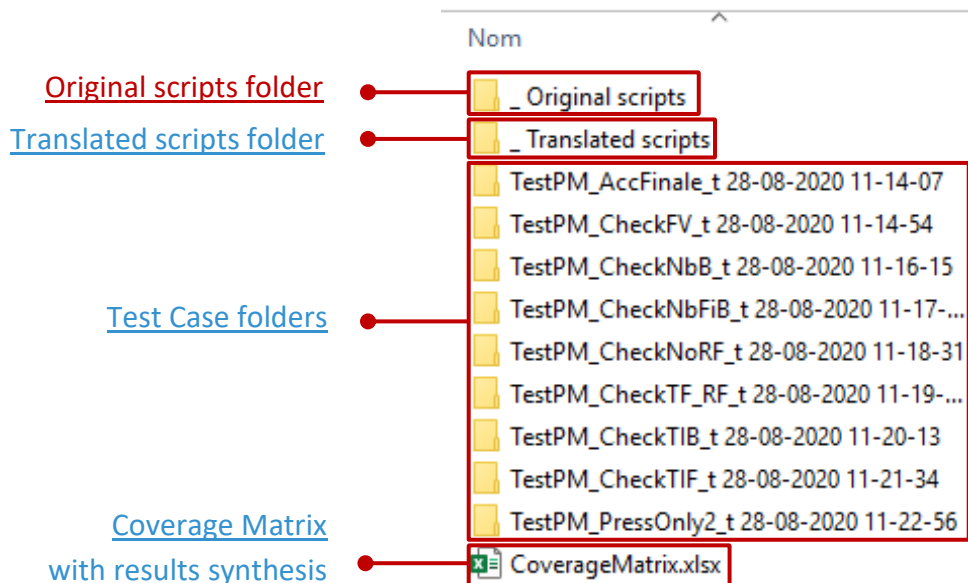
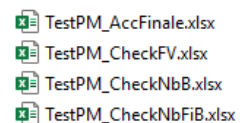


Figure 11. Sequence folder

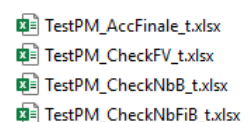
### 1. Original scripts folder

The **Original script folder** contains all the original Test Case scripts. This folder allows you to easily access the scripts with the parameters.



### 2. Translated scripts folder

The **Translated script folder** contains all the translated Test Case scripts. As have seen [above](#), ACASYA translates at the very beginning all Test Case scripts with parameters into Test Case scripts with only value. This folder allows you to easily access the scripts with values only.



### 3. Test Case folders

The **Test Case folders** are very important folders because they contain the analysis results of each Test Case. There are as many folders as Test Case scripts.

A Test Case folder contains several elements as shown in the Figure 12:

	Nom	Modifié le
Original scripts folder	Results	28/08/2020 11:14
Original scripts folder	Check_Results.txt	28/08/2020 11:14
ErroLog.txt file	ErrorLog.txt	28/08/2020 11:14
Logs file	Logs.xlsx	28/08/2020 11:14
	Logs.xml	28/08/2020 11:14
Test Case files	TestPM_AccFinale_t.xlsx	28/08/2020 11:14
	TestPM_AccFinale_t_log.xlsx	28/08/2020 11:14

Figure 12. Test Case folder

#### a. Results folder

The Results folder contains several configuration files using by ACASYA during the execution such as the Database, the Config.ini file (with some environmental parameters), the ANum configuration, the original log file with received frames, etc.

Nom
2020-08-28 11-14-14_Auto_000.csv
BenchTest.csv
Config A149_HMC sw1300_cfg
Config A149_HMC sw1300_Full#000001...
Config.ini
HMC_Database.xlsx

#### b. Check\_Results.txt file

The Check\_Results.txt file is the most useful file of the Test Case folder. This text file contains the **analyse result** of the Test Case.

For each WFC, the text file indicates:

- the WU ID value and the Test Case name.
- the test parameters (labels, tolerances, FC value, etc.).
- the test conditions (expected value).
- rejected frames with wrong value.
- and whether the test is PASSED or FAILED.

You can find an example of a PASSED and FAILED Check\_Results.txt in the Appendix.

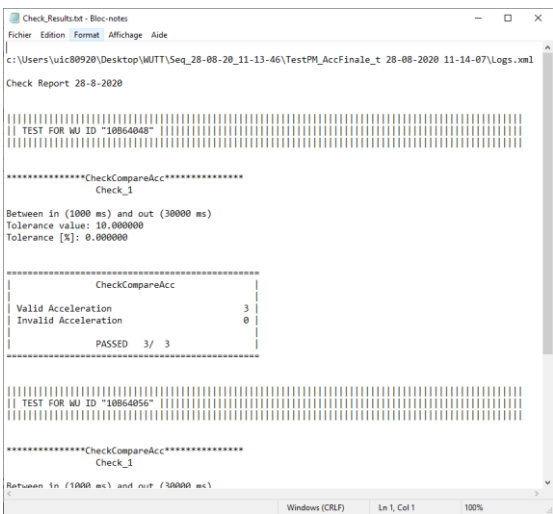


Figure 13. CheckResults.txt

#### c. ErrorLog.txt

The ErrorLog text file specifies the status of the Application during and after performing each test. The Error Log text file contains some information such as:

- the Test Case name.
- the Configuration File Parameters and Reading Status: Successful or Failed.
- the Test Case decoding errors.
- the Test Case Commands after decoding.
- and the Test Case Reading Status: Successful or Failed.

#### d. Logs files

The Logs.xml file contains all the logs with frames received.

The Logs.xlsx file is the log processed by ACASYA during the analysis including labels.

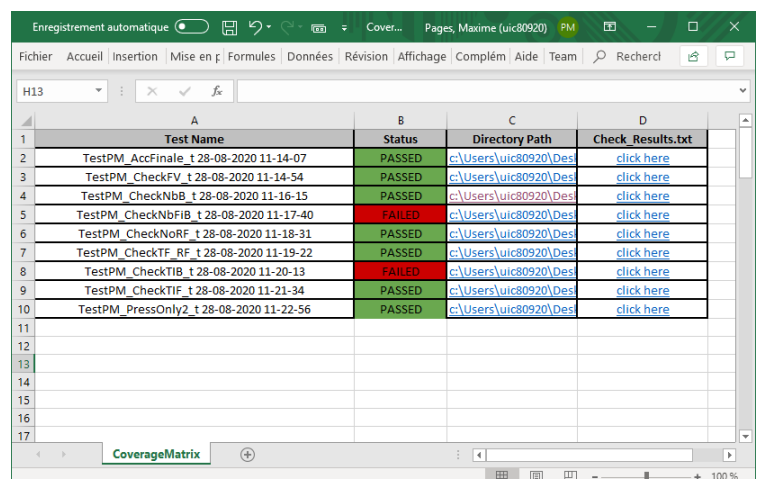
**These files are very useful for understanding where the errors are coming from.**

#### e. Test Case files

These Test Case files is the Test Case scripts used and generated by ACASYA during the analyse.

### 4. Coverage Matrix with results synthesis

The CoverageMatrix.xlsx is a very useful file to analyze your sequence. This Excel workbook allows you to quickly see which tests are passed or failed and to easily access to the Test Case folders and the CheckResults.txt files.



	A	B	C	D
	Test Name	Status	Directory Path	Check_Results.txt
1	TestPM_AccFinale_t 28-08-2020 11-14-07	PASSED	c:\Users\luic80920\Des	<a href="#">click here</a>
2	TestPM_CheckFV_t 28-08-2020 11-14-54	PASSED	c:\Users\luic80920\Des	<a href="#">click here</a>
3	TestPM_CheckNbB_t 28-08-2020 11-16-15	PASSED	c:\Users\luic80920\Des	<a href="#">click here</a>
4	TestPM_CheckNbFiB_t 28-08-2020 11-17-40	FAILED	c:\Users\luic80920\Des	<a href="#">click here</a>
5	TestPM_CheckNoRF_t 28-08-2020 11-18-31	PASSED	c:\Users\luic80920\Des	<a href="#">click here</a>
6	TestPM_CheckTF_RF_t 28-08-2020 11-19-22	PASSED	c:\Users\luic80920\Des	<a href="#">click here</a>
7	TestPM_CheckTiB_t 28-08-2020 11-20-13	FAILED	c:\Users\luic80920\Des	<a href="#">click here</a>
8	TestPM_CheckTiF_t 28-08-2020 11-21-34	PASSED	c:\Users\luic80920\Des	<a href="#">click here</a>
9	TestPM_PressOnly2_t 28-08-2020 11-22-56	PASSED	c:\Users\luic80920\Des	<a href="#">click here</a>
10				
11				
12				
13				
14				
15				
16				
17				

Figure 14. CoverageMatrix.xlsx

- Click on the hyperlinks of the “Directory Path” column to access to the Test Case folder.
- Click on the hyperlinks of the “Check\_Results.txt” column to access to the Check\_Results.txt file.

---

Chapter 4

# RE-ANALYSE MODE

---

## IV. RE-ANALYSE MODE

The Re-Analyse Mode is the last mode added to ACASYA in the last update. This very useful mode allows you to analyse Log file from a previously performed test. It is a perfect mode to save time when you don't want to execute a test once again.

### A. IHM

The main window is divided as follow:

#### (1) Files selection

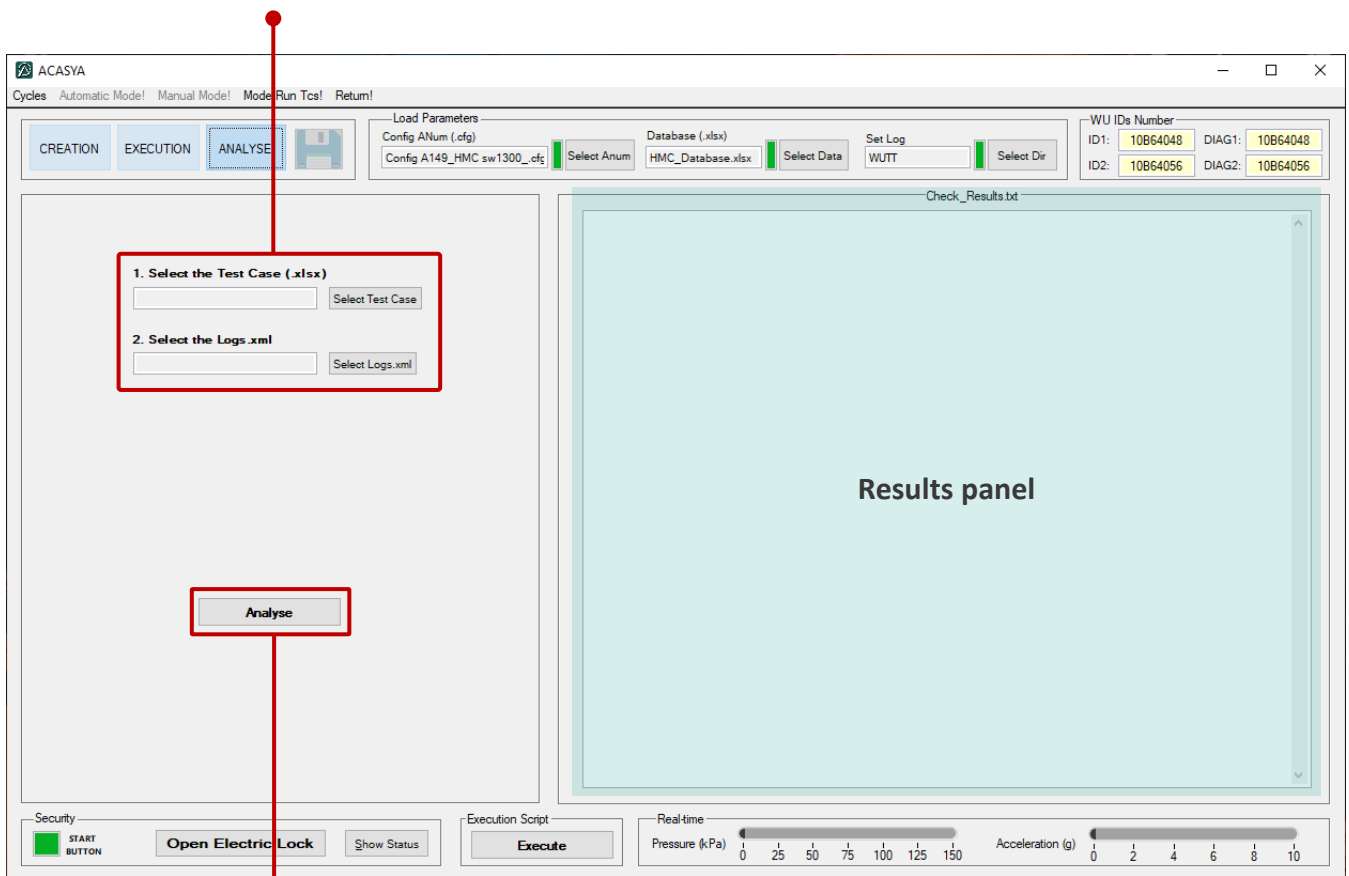


Figure 15. Re-Analyse Mode main window

#### (2) Analyse button

**(1) Files selection:** panel to select the Test Case including the Expected Results script (with check functions) and the Logs.xml file.

**(2) Analyse button:** button to start the analyse



## B. Starting an analysis

To start an analysis, you need to indicate the Test Case including the Expected Results script (with check functions) and the Logs.xml file.

**Step 1:** Click on the *Select Test Case* button in the [File Selection](#) panel to choose the Test Case.


**Step 2:** Click on the *Select Logs.xml* button in the [File Selection](#) panel to choose the Logs.xml file.

**Step 3:** Once the Test Case and the Logs.xml file are loaded, click on the [Analyse](#) button to start the analysis.

A picture of a timer should appear, and the analysis may take few seconds.



## C. Analysis results

Once the analysis completed, ACASYA creates in your working folder a new Test Case folder. The Test Case folder is stamped with the date and time, as presented here:  TestPM\_CheckNbFiB\_28-08-20\_16-18-38

The Test Case folder contains the Test Case script and the Check\_Results.txt, identical to the one presented in the [Check\\_Results.txt file](#) section.

In addition, the [Results panel](#) displays the Check\_Results.txt file created during the analyse and allows you to quickly verify whether the test is passed or failed.

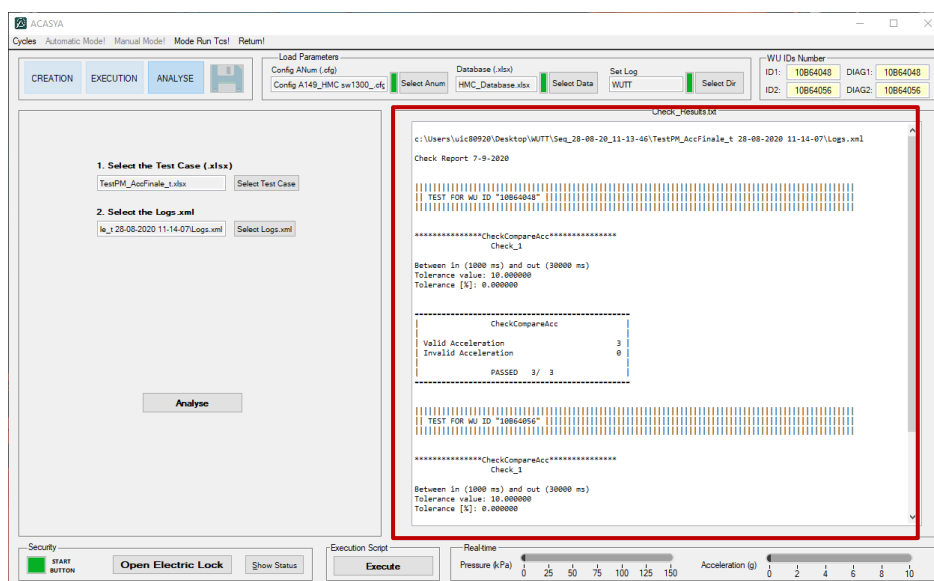


Figure 16. Analyse Results panel



---

Chapter 5

# SECURITY AND MONITORING

---

## V. SECURITY AND MONITORING

During tests, ACASYA sends pressure and acceleration commands to the LSE test bench. Some dangerous behavior for the user can occur: the pressure and acceleration values are too high, the safety doors are not closed...

To avoid any dangerous behavior, ACASYA was designed with some safety mechanisms.

### A. Electric Lock and Security Loop Blue button

If you want to have access to the WFCs, you need to open the Electric Locks clicking the **Open Electric Lock button**, presented below in the Figure 17:

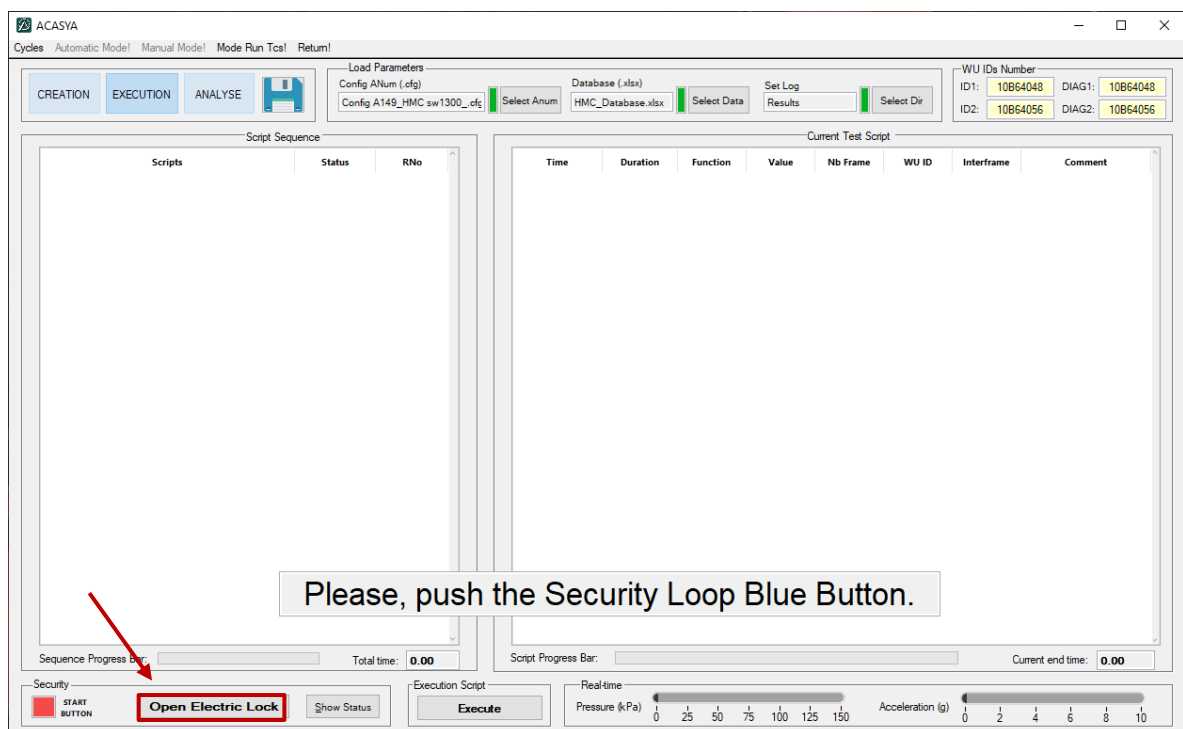


Figure 17. Electric lock and Security Loop Blue Button

Once the Electric Locks are closed, click again on the same button to validate the closing of the doors. The message *"Please, push the Security Loop Blue Button"*, as shown above, now indicates to press the Blue button located near to the LSE test bench in order to interact with the bench safely.

## B. Monitoring and status panel

Clicking on the **Show Status** button allows you to display the Status window, as presented in the Figure 19.

This window presented in the Figure 19. Status window shows the status of some elements of the bench and test equipment.

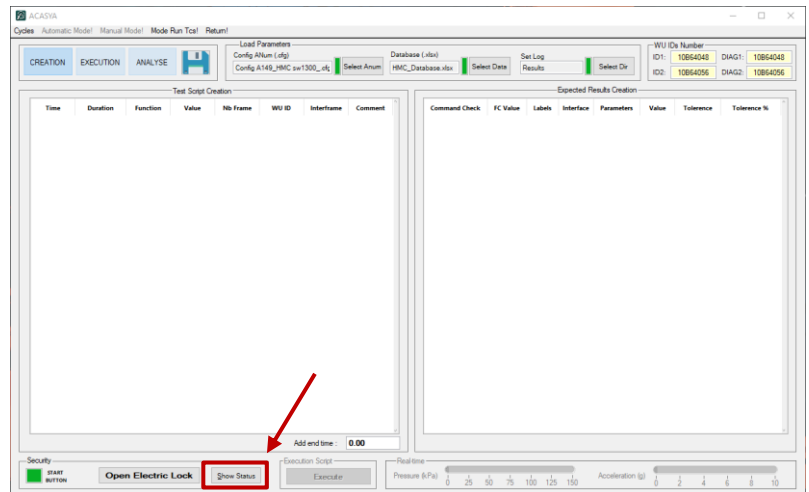


Figure 18. Status button position

This monitoring window allows you to check the status of doors, security hood, security loop, emergency stop, USB6008 communication, etc.

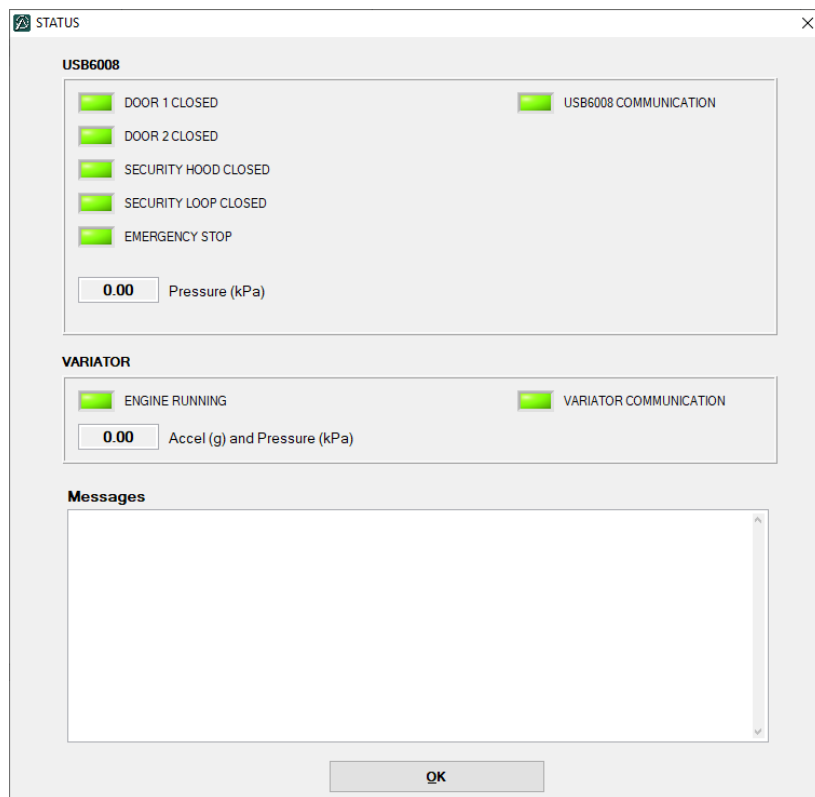


Figure 19. Status window



---

# APPENDIX

---

## VI. APPENDIX

### A. Installation files

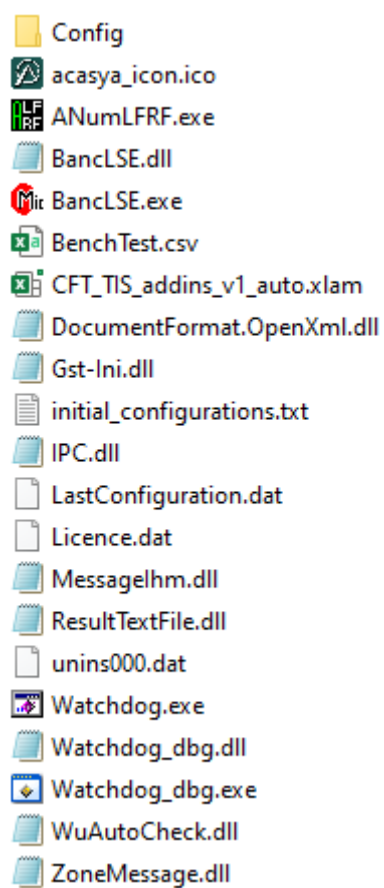


Figure 20. Installation files



## B. Check Results example

### 1. PASSED example

c:\Users\uic80920\Desktop\WUTT\Seq\_10-09-20\_11-29-04\TestPM\_AccFinale\_t 10-09-2020 11-29-37\Logs.xml

Check Report 10-9-2020

```
|||||
|| TEST FOR WU ID "10B64048"
|||||
```

```
*****CheckCompareAcc*****
          Check_1
```

Between in (1000 ms) and out (30000 ms)  
Tolerance value: 10.000000  
Tolerance [%]: 0.000000

```
=====
|          CheckCompareAcc          |
|                                     |
| Valid Acceleration                 8 |
| Invalid Acceleration               0 |
|                                     |
|          PASSED      8/   8        |
=====
```

```
|||||
|| TEST FOR WU ID "10B64056"
|||||
```

```
*****CheckCompareAcc*****
          Check_1
```

Between in (1000 ms) and out (30000 ms)  
Tolerance value: 10.000000  
Tolerance [%]: 0.000000

```
=====
|          CheckCompareAcc          |
|                                     |
| Valid Acceleration                 8 |
| Invalid Acceleration               0 |
|                                     |
|          PASSED      8/   8        |
=====
```

## 2. FAILED example

c:\Users\uic80920\Desktop\WUTT\Seq\_10-09-20\_11-29-04\TestPM\_PressOnly2\_t 10-09-2020 11-39-18\Logs.xml

Check Report 10-9-2020

```
|||||
|| TEST FOR WU ID "10B64048"
|||||
```

```
*****CheckCompareP*****
      Check_1
```

Between in (1000 ms) and out (29000 ms)  
Tolerance value: 10.000000  
Tolerance [%]: 0.000000

The pressure 100.000000 is not between [101.890974,121.890974], (frames nb 14)

```
=====
|                      CheckCompareP                      |
|                                                           |
| Valid Pressure                        7 |
| Invalid Pressure                      1 |
|                                                           |
|                      FAILED    1/   8                      |
=====
```

```
|||||
|| TEST FOR WU ID "10B64056"
|||||
```

```
*****CheckCompareP*****
      Check_1
```

Between in (1000 ms) and out (29000 ms)  
Tolerance value: 10.000000  
Tolerance [%]: 0.000000

```
=====
|                      CheckCompareP                      |
|                                                           |
| Valid Pressure                        4 |
| Invalid Pressure                      0 |
|                                                           |
|                      PASSED    4/   4                      |
=====
```



---

**ACASYA** for WFC  
- User Manual -  
Authors: Maxime PAGES, Marvyn PANNETIER  
September 2021

