OETCS/WP5/M5.3

openETCS
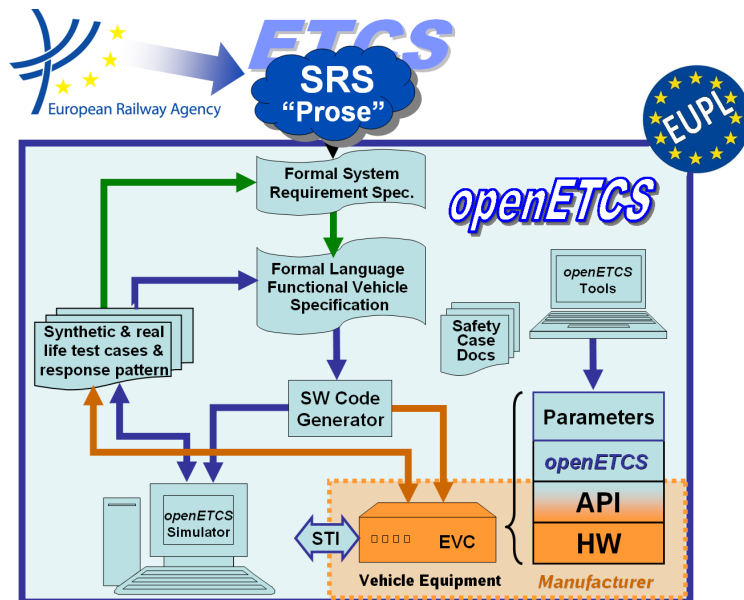
Work-Package 5: "Demonstrator"

ESP Basic Installation and User guide

A comprehensive guide for installing and using ESP Basic

Alexis Julin, Didier Weckmann, Nicolas Van Landeghem   February 2015

This page is intentionally left blank

Work-Package 5: "Demonstrator"   **OETCS/WP5/M5.3**
                                                                                        February 2015

ESP Basic Installation and User guide
A comprehensive guide for installing and using ESP Basic

Document approbation

Lead author:	Technical assessor:	Quality assessor:	Project lead:
location / date	location / date	location / date	location / date
signature	signature	signature	signature
Nicolas Van Landeghem (ERSA)	Alexis Julin (ERSA)	Ainhoa Gracia (SQS)	Klaus-Rüdiger Hase (DB Netz)

Alexis Julin, Didier Weckmann, Nicolas Van Landeghem

ERSA
5 Rue Maurice Blin
67500 Haguenau, France

Description of work

Prepared for   OpenETCS Project

Abstract: This document present how the ESP Basic can be installed and can be used to execute Baseline 3 scenarios.

Modification History

Version	Section	Modification / Description	Author
0.1	All parts	Creation	Alexis Julin
1.0	All parts	Add configuration script description	Alexis Julin

Table of Contents

Figures and Tables

Figures

Tables

1 Abstract

This document explains how to install, configure and use the ESP Basic software package and tools.

The purpose of the ESP_Basic is to provide a tool that allows to perform automatic tests of the on-board simulator through a set of scenarios. During development and tests phases, it is required to validate the changes and perform non regression tests.

The Test Environment is a graphical application integrating an on-board, as well as a simplified DMI. It can simulate all interfaces to the EVC (e.g. balises, radio, loops, odometer, TIU and driver interfaces). It is possible to test the internal state of on-board in order to check its correct behaviour according to an input scenario.

The EVC is included in an independent application EVC Server which can run on the same computer than the Test Environment or an other computer using TCP/IP connection.

This document only concerns ESP Basic installation, configuration and use. For Test Environment dedicated scenarios, report to Test_Environment_User_guide documentation.

Warning: At the moment, this documentation is only valid for CentOS 6.5, CentOS 6.6 (Atom computers or virtual machines), as well as Red Hat 6.4.

2 Installation

2.1 Prerequisites

2.1.1 Local configuration

In order to install ESP_Basic in local configuration, the following are required:

- Computer

- CentOS 6.5 or 6.6

- Internet connection

- Root access

2.1.2 Remote configuration

In order to install ESP_Basic in remote configuration, the following are required:

- 2x Computers

- 2x CentOS 6.5 or 6.6

- Internet connection for each computer

- Root access on each computer

- *Account name must be the same on both computers.*

2.2 Installation

NB: You will have to repeat this operation for each computer in remote configuration.
Warning: The installation process requires a computer reboot. Save and close all your open documents before starting this step.

- Copy ESP_Basic archive on desktop

- Open a terminal on desktop

- Unpack ESP_Basic archive
  `>tar xvfj openETCS_WP5_X_X_X.tar.bz2`

- Connect as root

- Launch installation script
  `>./Install_Basic_ESP`

- Wait the computer to complete its reboot

2.3 Getting a license (First installation only)

2.3.1 Getting computer fingerprint

- open a *non-root* terminal

- move to ESP_Basic folder
  `>cd /usr/local/ESP_Basic`

- move to Licensing folder
  `>cd Licensing`

- launch script for getting fingerprint
  `> ./echoid >> echoid_output.txt`

- send an email with file echoid_output.txt to *licenseactivation@ersa-france.com*

2.3.2 Installing license

You will receive an email with a licensing file

- open licensing file with a text editor

- open a root terminal

- move to ESP_Basic folder
  `>cd /usr/local/ESP_Basic`

- move to Licensing folder
  `>cd Licensing`

- launch license installer script
  `>./licenseinstaller -i -l 'xxxx' /opt/ERSA/license/license.rc` where xxxx must be replaced by the key in licensing file (you can find it from first character to # character *include*)

2.4 Validation

Check that the folder */usr/local/ESP_Basic/* exists.
Check that the following tree exists in */usr/local/ESP_Basic/*:

- folder *evc_server_gui*

  - executable *evc_server_gui*
  - script *RunESG.py*
  - optional folder *data*
  - optional compilation files

- folder *lib*

  - shared library *libevc.so*
  - some symbolic links to this library

- folder *test_runner2*

  - executable *test_runner2*
  - script *RunTestRunner.py*
  - folder *scenarios* which contains scenarios for Test Environment
  - optional compilation files

2.5 Upgrade ESP_Basic

- Uninstall previous ESP_Basic version
  ```
  >rpm -e ESP_Basic
  ```

- Install new ESP_Basic version
  ```
  >yum localinstall ESP_Basic_X_X_X_X.rpm
  ```

2.6 Uninstall ESP_Basic

- Uninstall ESP_Basic version
  ```
  >rpm -e ESP_Basic
  ```

3 Automatic Configuration

This chapter describes how to use configuration script in order to automatically configure ESP_Basic and avoid manual configuration.

3.1 Description

This script allows to:

- change host name on local and remote computers if needed

- change iptables to allow communication between EVC and test runner

- configure ssh communication in order to avoid to enter password for each ssh communication

3.2 Launching

Script to launch Test runner can detect if a new configuration is needed (i.e local/remote hostname or ip change). Use following command to launch configuration script

- Local configuration
  `>python configurationScript.py -l`

- Remote configuration
  `>python configurationScript.py --remote-ip 192.168.55.234`

3.3 Parameters

- -h or `--help`: show help message and exit

- -d or `--debug-trace`: activate debug trace for script

- -s or `--server-side`: Script launch on server (machine where ESG is running)

- -l or `--local`: Run local configuration for ESP (no need to specify remote-ip parameter)

- `--local-password`: password to connect to local host. Default value = dbm

- `--local-hostname`: hostname for local host (only use if current hostname is localhost.localdomain). Default value = ESP_Basic2

- `--local-root-password`: password to connect as a root on local machine. Default value = password

- `--remote-login`: login to connect to remote host. Default value = dbm

- `--remote-password`: password to connect to remote host. Default value = dbm

- `--remote-hostname`: hostname for remote host (only use if current hostname is localhost.localdomain). Default value = ESP_Basic1

- `--remote-ip`: IP address for remote host

- `--remote-root-password`: password to connect as a root on remote machine. Default value = password

- `--remote-script-path`: configuration script path on remote machine. Default value = /usr/local/ESP_Basic/test_runner2

3.4 Warnings

3.4.1 Backup current configuration

If user install ESP_Basic on a non-empty computer (or virtual machine), it is highly recommended to save on a safe place the following files and folders:

- */etc/hosts*

- */etc/sysconfig/network*

- */home/*user_name*/.ssh*

Use root access to save/restore backup file

3.4.2 ESP_Basic failure configuration

If there is a failure during execution of configuration script, try manual configuration describe in next chapter.

3.4.3 Operating System

This configuration script is only valid for CentOS 6.5, CentOS 6.6. It should work on Red Hat 6.4 too.

4 Manual Configuration

4.1 Local configuration

This step is mandatory to be performed once only, after the installation step. To achieve this step you have to retrieve the computer IP address (use command *ifconfig* to get it) and host name (use command *hostname* to get it).
For example we assume:

- ip address = 192.168.55.234

- host name = ESP_Client

4.1.1 Optionnal: Change host name

If *hostname* return *localhost.localdomain* you have to change host name before next step.

- Open a terminal

- Connect as root

- Open network file
  >gedit /etc/sysconfig/network

- Replace in the file
  HOSTNAME=localhost.localdomain
  with
  HOSTNAME=ESP_Client (for example)

- Save and close */etc/sysconfig/network* file

- Exit root access
  >exit

- Reboot computer

4.1.2   Update hosts file

- Open a terminal

- Connect as root

- open hosts file
  `>gedit /etc/hosts`

- append at the end of file
  `192.168.55.234  ESP_Client`

- Save and close *hosts* file

- Exit root access
  `>exit`

4.2   Remote configuration

This step is mandatory each time a simulation has to be started with a new server computer (where EVC server will run) or a new client (where Test Environment will run).
To achieve this step you have to retrieve the server computer IP address (use command *ifconfig* to get it) and host name (use command *hostname* to get it).
To complete this step you have to know client computer IP address (use command *ifconfig* to get it) and host name (use command *hostname* to get it).
For example we assume:

- client ip address = 192.168.55.234

- client host name = ESP_Client

- client account name = dbm

- server ip address = 192.168.55.236

- server host name = ESP_Server

- server account name = dbm

4.2.1   Optional: Change host name

Complete this step both on server computer and client computer If *hostname* return *localhost.localdomain* you have to change host name before next step.

- Open a terminal

- Connect as root

- Open network file
  `>gedit /etc/sysconfig/network`

- Replace in the file
  `HOSTNAME=localhost.localdomain`
  with
  `HOSTNAME=ESP_Client (or ESP_Server)`

- Save and close *hosts* file

- Exit root access
  `>exit`

- Reboot computer

4.2.2 Update hosts file

Complete this step on server computer and client computer both

- Open a terminal

- Connect as root

- open hosts file
  `>gedit /etc/hosts`

- append (or edit if needed) at the end of file
  `192.168.55.236  ESP_Server`
  `192.168.55.234  ESP_Client`

- Save and close *hosts* file

- Exit root access
  `>exit`

4.2.3 Copy ssh key

4.2.4 Client side

- Open a terminal

- Generate ssh key
  `>ssh-keygen`

  - For the question:
    `>Enter file in which to save the key (/home/dbm/.ssh/id_rsa):`
    press *return*
  - If command display the question:
    `>: Overwrite (y/n)?`
    press *y*
  - For all other questions press *return*

- Copy ssh key on server
  `>ssh-copy-id dbm@192.168.55.236`

- – For the question:
  >`Are you sure you want to continue connecting (yes/no)?`
  enter *yes*

- – enter password for server computer account to complete copy

- reboot computer

4.2.5 Server side

- Open a terminal

- Generate ssh key
  >`ssh-keygen`

  - – For the question:
    >`Enter file in which to save the key (/home/dbm/.ssh/id_rsa):`
    press *return*

  - – If command display the question :
    >`: Overwrite (y/n)?`
    press *y*

  - – For all other questions press *return*

- Copy ssh key on client computer
  >`ssh-copy-id dbm@192.168.55.234`

  - – For the question:
    >`Are you sure you want to continue connecting (yes/no)?`
    enter *yes*

  - – enter password for client computer account to complete copy

- reboot computer

4.2.6 Open communication port

Complete this step on server computer only

4.2.7 Trusted network

In order to allow communication between server computer and client computer you can disable firewall On Centos :

- Click on *System*

- Click on *Administration*

- Click on *Firewall*

- Enter server computer password (root password)

- Click on *Trusted Interfaces* (left panel)

- Check *eth0* and *eth1* lines

- Click on *Apply* (top panel)

- Close window

4.2.8   Untrusted network

In order to allow communication between server computer and client computer (and only this communication) it is necessary to edit iptables. For this step, values to change depend on current iptables configuration. In general, you have to use following command

- Open a terminal

- Connect as root

- Display current iptables configuration:
  ```
  >iptables -nvL --line-numbers
  ```
  For test computer we get following answer

```
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts  bytes  target  prot  opt  in  out   source       destination
1    6619  6544K  ACCEPT  all   --   *   *     0.0.0.0/0    0.0.0.0/0   ..
2       2    120  ACCEPT  icmp  --   *   *     0.0.0.0/0    0.0.0.0/0
3      10    575  ACCEPT  all   --   lo  *     0.0.0.0/0    0.0.0.0/0
4      15    900  ACCEPT  tcp   --   *   *     0.0.0.0/0    0.0.0.0/0   ..
5   14809  1514K  REJECT  all   --   *   *     0.0.0.0/0    0.0.0.0/0   ..


Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
num  pkts  bytes  target  prot  opt  in  out  source       destination
1       0      0  REJECT  all   --   *   *    0.0.0.0/0    0.0.0.0/0  ...


Chain OUTPUT (policy ACCEPT 3929 packets, 1066K bytes)
num  pkts  bytes  target  prot  opt  in  out  source       destination
```

- You have to insert authorization for client packets before REJECT all rules (5) in *Chain INPUT* and *Chain OUTPUT* if a rule exist for this section

- to allow input from client, you have to insert at position 5 in iptables
  ```
  >iptables -I INPUT 5 -s 192.168.55.234 -j ACCEPT
  ```

- to allow output to client, and as there is no rule REJECT all you can just add a rule to allow this communication
  ```
  >iptables -A OUTPUT -d 192.168.55.234 -j ACCEPT
  ```

- Display the final iptables
  ```
  >iptables -nvL --line-numbers
  ```
  For test computer we get following answer

```
Chain INPUT ( policy ACCEPT 0 packets , 0 bytes )
num pkts  bytes  target  prot opt in out source       dest
1   6619  6544K  ACCEPT  all  --  *   *  0.0.0.0/0    0.0.0.0/0
2      2   120   ACCEPT  icmp --  *   *  0.0.0.0/0    0.0.0.0/0
3     10   575   ACCEPT  all  --  lo  *  0.0.0.0/0    0.0.0.0/0
4     15   900   ACCEPT  tcp  --  *   *  0.0.0.0/0    0.0.0.0/0
5      0     0   ACCEPT  all  --  *   *  192.168.99.78 0.0.0.0/0
6  15831 1628K   REJECT  all  --  *   *  0.0.0.0/0    0.0.0.0/0

Chain FORWARD ( policy ACCEPT 0 packets , 0 bytes )
num pkts  bytes  target  prot opt in out source       dest
1      0     0   REJECT  all  --  *   *  0.0.0.0/0   0.0.0.0/0

Chain OUTPUT ( policy ACCEPT 0 packets , 0 bytes )
num pkts  bytes  target  prot opt in out source       dest
1      0     0   ACCEPT  all  --  *   *  0.0.0.0/0  192.168.99.78
```

- Save iptables
  >`service iptables save`

To remove previous configuration, in order to avoid to keep useless open connection, you have to use following commands

- Remove in *Chain INPUT*
  >`iptables -D INPUT -s 192.168.55.234 -j ACCEPT`

- Remove in *Chain OUTPUT*
  >`iptables -D OUTPUT -d 192.168.55.234 -j ACCEPT`

5 User guide

5.1 Launch ESP Basic

5.1.1 In local configuration

In order to launch ESP Basic in local configuration, you have to:

- Open a terminal

- Go in installation folder
  >`cd /usr/local/ESP_Basic/test_runner2`

- Launch script to run ESP_Basic
  >`python RunTestRunner.py scenarios/ScenarioName.sce`

As result you can see a two GUIs displayed :

- In a first time EVC_Server

- In a second time AutoTestRunner2 (Test Environment)

5.1.2 In remote configuration

In order to launch ESP Basic in remote configuration, proceed the following:

- Open a terminal on the client computer

- Go in installation folder
  `>cd /usr/local/ESP_Basic/test_runner2`

- Launch script to run ESP_Basic
  `>python RunTestRunner.py -i 192.168.55.236 scenarios/ScenarioName.sce`
  You have to specify in parameter server computer IP address.

As result you can see a two GUIs displayed:

- In a first time EVC_Server on the server computer

- In a second time AutoTestRunner2 (Test Environment) on the client computer

5.2 Close ESP Basic

In order to close properly ESP Basic, you have to:

- Close AutoTestRunner2

- Close EVC Server

5.3 Available options

Some options are available in the script *RunTestRunner.py*

- -i: specify server computer ip address for TCP/IP communication (only in remote configuration). Default value is 127.0.0.1 and corresponds to the localhost. This will use your local IP stack.

- -p: specify server computer port for TCP/IP communication (only in remote configuration). Default value is 12456.

- -c: activate auto-close option. When scenario ended, Test Environment and EVC Server close automatically.

- -d: generate debug files. When this option is not activated, ESP Basic generate debug files only for function which fail.

- -s: specify path where is installed EVC Server (if EVC Server installation folder was moved). Default value is */usr/local/ESP_Basic/evc_server_gui*

- -x: activate Server X management. This option allows to run EVC Server on a server X (use in remote configuration only).

- -m: disable automatic configuration checking. This allow to run EVC server when configuration has been done manually.

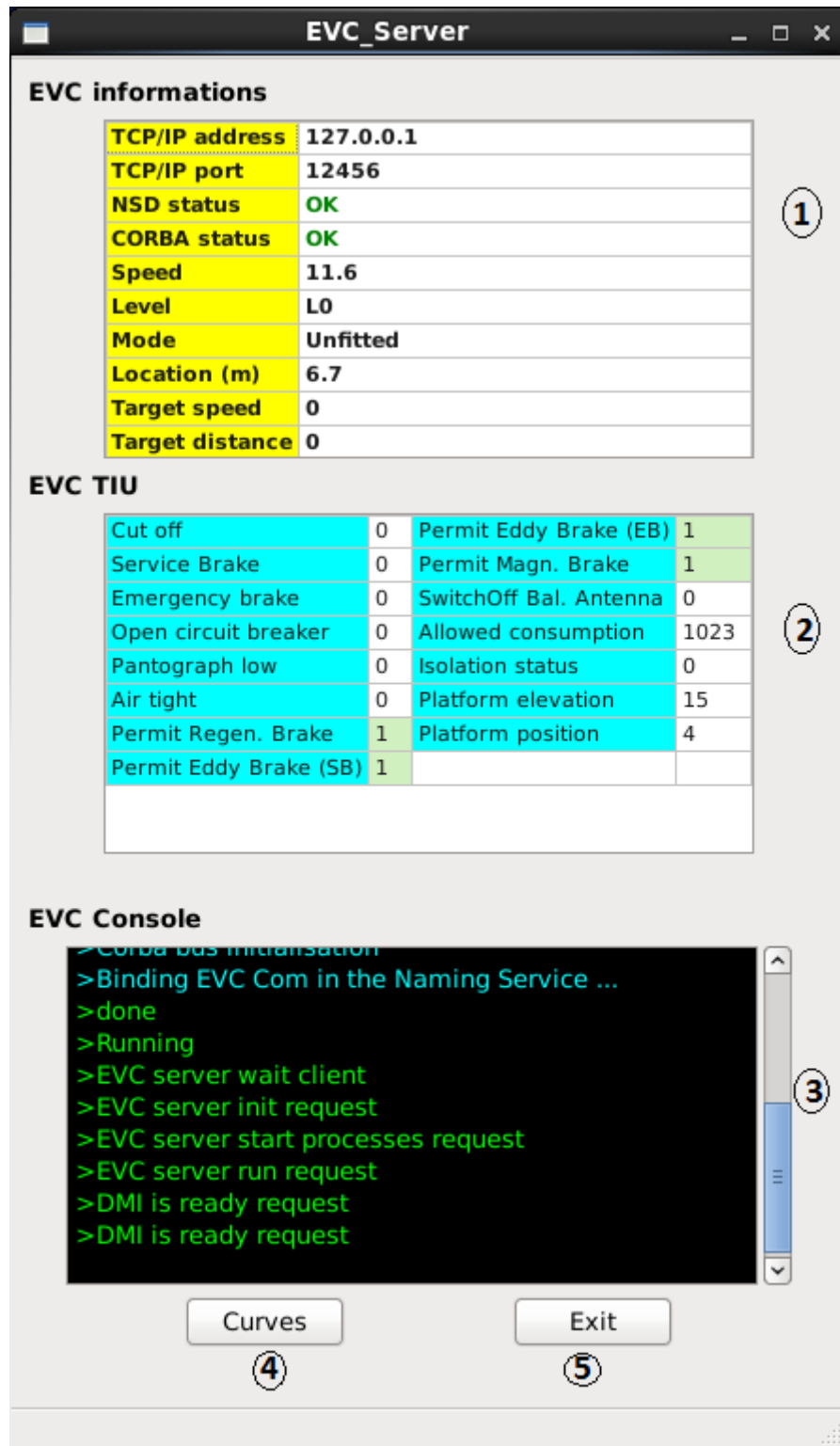### 5.4 GUI description

5.4.1 EVC Server

Figure 1. EVC Server

- 1: Simulation information

- 2: EVC TIU

- 3: EVC server console

- 4: Curves displaying

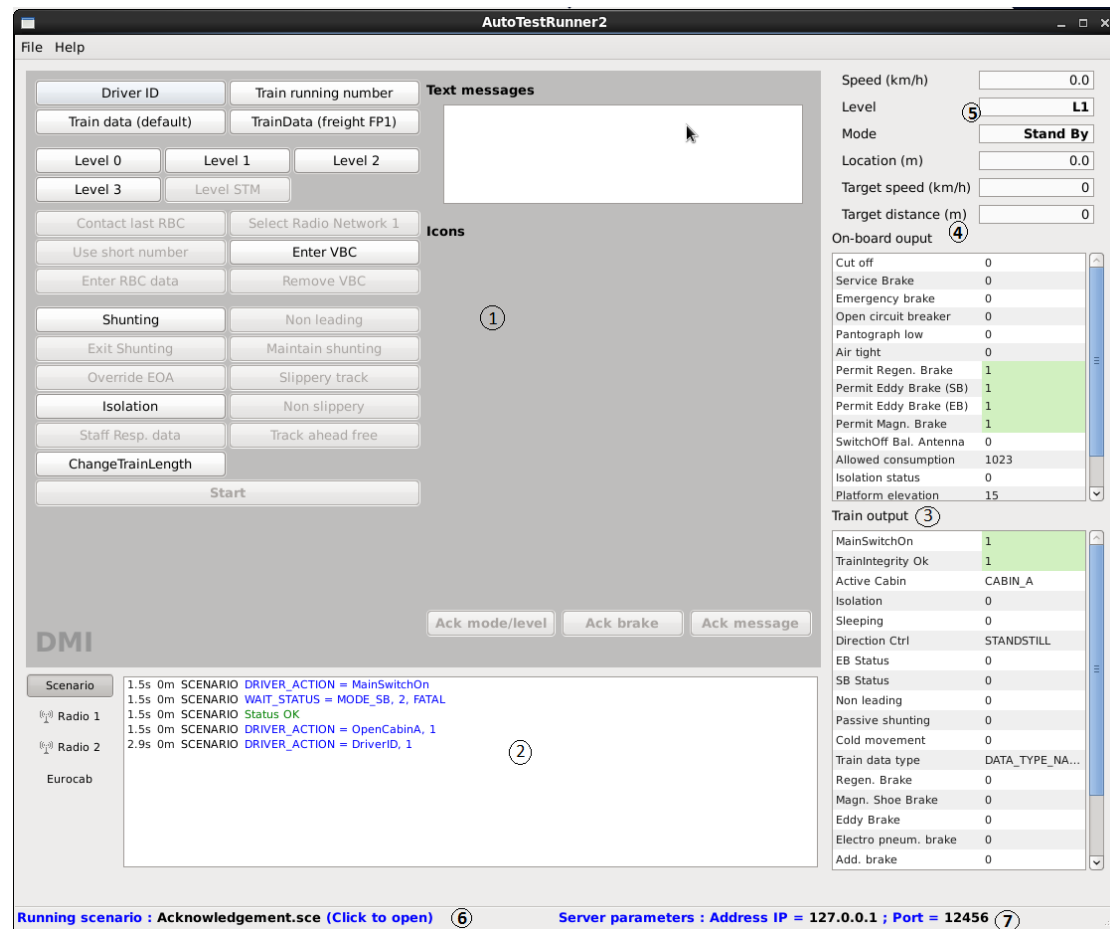- 5: Exit

5.4.2   Test Environment

Figure 2. Test Environment

- 1: Simplified DMI

- 2: Scenario messages console

- 3: Train TIU

- 4: EVC TIU

- 5: On-board status

- 6: Running scenario

- 7: Communication with server computer parameters

5.5   Output files

5.5.1 Server side

There are 3 kinds of files generated during simulation:

- EVC log files in */usr/local/ESP_Basic/evc_server_gui/data/log*

- EVC curve images in */usr/local/ESP_Basic/evc_server_gui/data/curve*

- Potential error files (syntax debug_*.stderr and debug_*.stdout) in */usr/local/ESP_Basic/evc_server*

5.5.2 Client side

There is 1 kind of files generated during simulation

- Potential error files (syntax debug_*.stderr and debug_*.stdout) in */usr/local/ESP_Basic/test_runner2*

6 Common problems and solutions

6.1 Client side displays : Fail to get x11-display

Test Environment fails to get display address. To solve this issue:

6.1.1 Solution 1

Retry the previous command. Sometimes it may happen that the first time ESP_Basic fail to get X11-display.

6.1.2 Solution 2

Activate or remove option -x in RunTestRunner command line.

6.1.3 Solution 3

On client computer and server computer, enter the command:
`>xhost +`
If it helped to work, append this command in file */home/username/.bash_profile* on client computer and server computer.

6.2 Client side asks password

Copy of the ssh key fail or ssh key is incorrect. To solve this issue:

6.2.1 Solution 1

Reboot both the client computer and server computer

6.2.2 Solution 2

Restart 4.2.2
Warning: Answer yes to question overwrite

6.3   EVC Server is launch but Test Environment not

6.3.1   File /etc/hosts is not correct

Check IP address and hostname are correct in file *etc/hosts*
Check computer host name is not present in line 127.0.0.1 and in line ::1. Remove it if you can find it in these lines.

6.3.2   Iptables are not correct

Your IP tables may be incorrect:
Try to disable firewall (3.2.4 A). If it works now your IP tables are wrong, restart 3.2.4 B and check the position where you insert your rule (*before rule with REJECT all*). Do not forget to save your iptables configuration.