atelier B

# PLUGINS CSSP

## User Manual

### version 1.0 — May 2024

CLEARSY
Safety Solutions Designer

# ATELIER B
# PLUGINS CSSP – User Manual
version 1.0

Document établi par CLEARSY.

CLEARSY
Maintenance ATELIER B
Parc de la Duranne
320 avenue Archimède
Les Pléiades III - Bât.A
13857 Aix-en-Provence Cedex 3
France

Tél 33 (0)4 42 37 12 99
Fax 33 (0)4 42 37 12 71
email : maintenance.atelierb@clearsy.com

# Table des matières

# Chapitre 1

# Installing dependencies

## 1.1 List of dependencies

The Atelier B CSSP edition has more dependencies than the other versions. Some of them cannot be provided in the various installation packages and will have to be installed by the user.
These various dependencies are listed in the following table.

| Dependencies | Uses | Platforms |
|---|---|---|
| `python 3.6` or greater with `tkinter` | ● Creating a CSSP project<br>● CSSP Monitor<br>● CSSP Emulator<br>● Uploading a program on CSSP | ● Windows<br>● Linux |
| `pip` | ● CSSP Emulator<br>● Uploading a program on CSSP | ● Windows<br>● Linux |
| `make` | ● CSSP Emulator<br>● Uploading a program on CSSP | ● Linux |
| `mingw32-make.exe` | ● CSSP Emulator<br>● Uploading a program on CSSP | ● Windows |
| `cmake` | ● CSSP Emulator<br>● Uploading a program on CSSP | ● Windows<br>● Linux |

| Dependencies | Uses | Platforms |
|---|---|---|
| `gcc` | ● CSSP Emulator<br>● Uploading a program on CSSP | ● Windows<br>● Linux |
| `g++` | ● CSSP Emulator<br>● Uploading a program on CSSP | ● Windows<br>● Linux |
| `qml-module-qtmultimedia` | ● Uploading a program on CSSP | ● Linux |
| `qml-module-qtquick-controls2` | ● Uploading a program on CSSP | ● Linux |
| `qtquickcontrols2-5-dev` | ● Uploading a program on CSSP | ● Linux |
| `qml-module-qtquick-dialogs` | ● Uploading a program on CSSP | ● Linux |
| `qml-module-qtquick-extras` | ● Uploading a program on CSSP | ● Linux |
| `qtdeclarative5-dev` | ● Uploading a program on CSSP | ● Linux |

## 1.2 Installing for a Linux distribution

For the Debian and Ubuntu versions of Atelier B CSSP edition, an installation script is provided. It should be run with administrator rights by placing it in the same directory as the .deb file also supplied. All the dependencies and Atelier B will be installed using the following command :

```
sudo cssp_install.sh
```

## 1.3 Installing for Windows

For the Windows version of Atelier B CSSP edition, you first need to run the Atelier B installer. You then need to install the dependencies.

### 1.3.1 Installing `python`

To install `python`, it is recommended to use the microsoft store. By doing so, `pip` and `tkinter` should also be installed.

### 1.3.2 Installing `cmake`

An installer for `cmake` is available online at : `https://cmake.org/download/`. You will need to select the option to add its path to the environment variable `PATH`.

### 1.3.3 Installing MinGW

We recommend installing `MinGW` xith : `https://winlibs.com/` (take the latest UCRT version with the POSIX). This also provides an executable `mingw32-make.exe` and compilers needed to emulate the CSSP card. Note that you will need to add the environment variable `PATH` the path to the folder `bin` contained in `MinGW`.

## 1.4 Note on reinstallation

When Atelier B is uninstalled, the directory containing the Clock and Combinatorial projects is not deleted. This means that when the same version of Atelier B is reinstalled, the user will not be asked where he wishes to store these projects. It is therefore advisable to delete this directory before reinstalling.

# Chapitre 2

# Use of CSSP-related functions

## 2.1 Creation of a CSSP project

To create a CSSP project, follow these instructions :

1. Open the menu *Atelier B* → *New* → *Project*.
2. Tick the boxes *Software development* and *Define as CSSP project*.
3. Enter the project name in the field *Project Name*.
4. Click on the button *Finish*.
5. Select *SK0* as the type of card and press *Ok*.
6. Select *Create new board*
7. Click on the button *Next* then on the button*Finish*.
8. A warning message should then be displayed. Click on *Yes*.

A project is then created and the necessary components are also imported. It should be noted that if no components are imported, this means that the installation script has failed, which may be due to the due to the absence of *tkinter* or a version of *python* earlier than 3.6.

## 2.2 Using the CSSP Monitor

When a CSSP is connected, it can be useful to check that it is working properly. To do this, you can use the monitor to observe exchanges taking place via the card's USB port. Use the menu *Project* → *CSSP Monitor*.

## 2.3 Uploading a binary on a CSSP

To upload a binary on a CSSP, open the menu *Project* → *CSSP Runner*, then follow the instructions on the screen. Note that under Linux, the user must have write, read and execute rights on the serial port to which the card is connected. This port is generally the file `/dev/ttyUSB0`.

## 2.4 Emulating a CSSP

If there is no CSSP card, it is possible to emulate its behaviour. To do so, open the menu *Project → SK0 Emulation.* A window will then open showing the progress of the compilation. When the compilation is complete, press the button *Ok*. It will then be possible to watch the execution progress by following the values of the variables. You can also click on them to change their values. Finally, click on the *reset* of the card will reset the execution of the program .