
User Guide: Representation and relative positioning from visual information

Submitted by:
ASMA BRAZI

Supervised by:
CÉDRIC HERPSON

Laboratory of Computer Sciences, Paris 6
Sorbonne University - Faculty of Sciences and Engineering

June - July 2019



Contents

1	Introduction	2
2	Getting started	3
2.1	Hardware prerequisites	3
2.2	OS/Software prerequisites	4
2.2.1	Java 8	5
2.2.2	Java IDE	5
2.2.3	JMonkey Engine	5
2.2.4	Python 3	6
2.2.5	Pip 3	6
2.2.6	OpenCV	6
2.2.7	Asebamedulla	6
2.3	Environment configuration	7
2.3.1	Wireless configuration	7
3	Launching the application	8

Chapter 1

Introduction

This document represents a guide for users. It is necessary to be consulted to ensure a proper configuration of the environment. It is designed to:

- Enumerate the different hardware and software prerequisites.
- Provide a step-by-step instructions to establish the environment configuration.
- Present a brief scenario of how the application runs.

Chapter 2

Getting started

We present in this section the necessary equipment for the use of our application. Also, we specify the essential programs to install.

2.1 Hardware prerequisites

The global system can be seen as two independent physical subsystems in interaction. The first one is composed of a computer, and the second one is what we call the autonomous robot. and whose list of components is showed in the figure below.

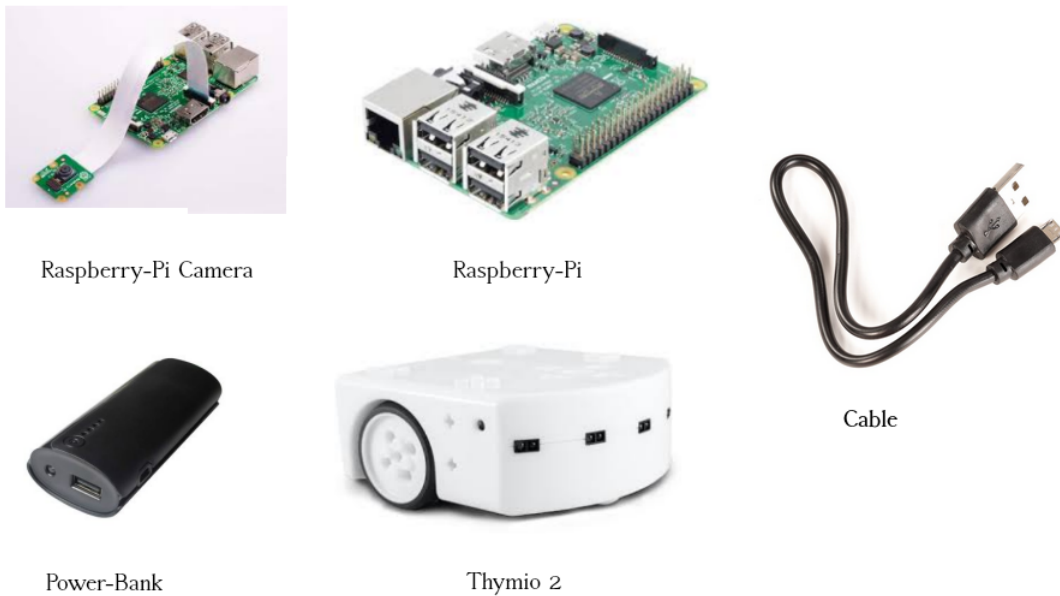


Figure 2.1: The components of the autonomous robot

The computer and the autonomous robot are exchanging via Wi-Fi. Nevertheless, the autonomous robot's components are connecting with cables. So, to connect the autonomous robot's components, we insert the Raspberry-Pi Camera's cable by into the Raspberry-Pi, so as that the cable is placed between Ethernet and HDMI ports, see Fig 2.2.

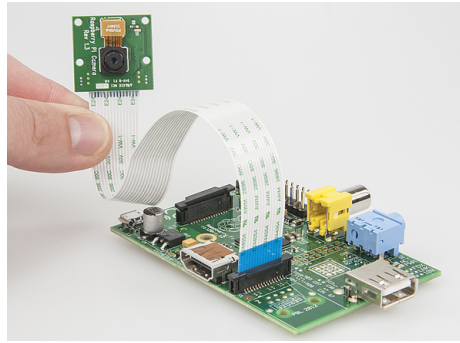


Figure 2.2: The connection of the Raspberry-Pi camera

Then we connect the Thymio to the Raspberry-Pi with a cable. Finally, in order to power the Raspberry-Pi, we use the power-bank which is connected to it with USB. see Fig 2.3.



Figure 2.3: The connected autonomous robot's components

2.2 OS/Software prerequisites

We need several programs for both subsystems to be able to launch the application. The programs that we have to install on the computer are:

- Java 8.
- Java IDE.
- JMonkey Engine.

Then those to install on the Raspberry-Pi are:

- Python 3 and Pip 3.
- OpenCV.

- Asebamedulla.

Then, those to install on the computer are:

We begin with the set of programs to install on the computer.

2.2.1 Java 8

Java can be installed with the following commands:

- Go to the website https://www.java.com/fr/download/linux_manual.jsp
- Download the version corresponding to your OS (64x or 32x)
- The complete instructions to install it are available on the same web page. They are available by clicking on the button *instructions* beside your the file to download.
- Go to the website <https://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Download the latest version of JDK available.
- The installation instructions can be consulted on the same web page.

2.2.2 Java IDE

There are many Java IDE that may be used (IntelliJ IDEA, Eclipse, Netbeans, etc). In our project scope, we use Eclipse 2019-03. The following steps indicate how to install this IDE:

- Go to the website <https://www.eclipse.org/downloads/>
- Download the version of Eclipse IDE corresponding to your OS (64x or 32x)
- Extract the downloaded archive, open the extracted directory and launch *eclipse-inst*
- Now, click on *Eclipse IDE for Java Developers*
- The followed instructions ask for the installation folder and the environment configuration (Eclipse's theme, etc), an installation by default can be done too.

2.2.3 JMonkey Engine

JMonkey Engine can be installed with the following commands:

- Go to the website <https://github.com/jMonkeyEngine/sdk/releases>
- Download the latest stable version corresponding to your OS (64x or 32x).
- Open a terminal and execute the command `sudo ./jmonkeyplatform-linux-*`
- The followed instructions ask for the installation folder and the environment configuration (Eclipse's theme, etc), an installation by default can be done too.

2.2.4 Python 3

Python can be installed with the following commands:

- *sudo apt-get update*
- *sudo apt-get install python3.6*

The following command is used to verify if Python has been correctly installed: *python3 - -version*

2.2.5 Pip 3

Pip can be installed with the following commands:

- *sudo apt-get update*
- *sudo apt install python3-pip*

The following command is used to verify if Pip has been correctly installed: *pip3 - -version*

2.2.6 OpenCV

OpenCV can be installed with the following commands:

- *sudo apt-get update*
- *sudo pip install opencv-python*
- *sudo pip install opencv-contrib-python*

To verify if OpenCV has been correctly installed, open a terminal and execute the following commands:

- *python*
- *import cv2*

If the error *ImportError: No module named cv2* appears, this means that OpenCV has not been installed correctly, otherwise, the installation is successful.

2.2.7 Asebamedulla

Asebamedulla can be installed with the following commands:

- Go to the website <http://wiki.thymio.org/en:linuxinstall>
- Download the version corresponding to your OS (64x or 32x)
- Open a terminal and execute the command *sudo dpkg -i aseba_**

2.3 Environment configuration

We chose to establish a connection through Wi-Fi rather than Ethernet, because the autonomous robot have to move in its environment. If the environment is large, even a very long Ethernet cable could not be enough.

2.3.1 Wireless configuration

As mentioned above, the computer and the Raspberry-Pi must be connected on the same network. In the case of the computer, it is easy to establish the connection, by selecting the name of the desired Wi-Fi network. Nonetheless, connecting the Raspberry-Pi to the network does not seem to be that straightforward.

In the case of the Raspberry-Pi, we have to assign an IP address statically to it, so that this IP address is available. In this way, every time the Raspberry-Pi starts, it connects to the router automatically. This is why we opted to fix the IP address.

So, we can assign the IP address of the Raspberry-Pi via SSH, or by connecting it to a screen and a keyboard at least. Once we can manipulate the Raspberry-Pi, we should perform the following steps:

- Open a terminal on the Raspberry-Pi.
- Execute the command: `sudo nano /etc/wpa-supPLICant/wpa-supPLICant.conf`, in order to modify the file **wpa-supPLICant.conf**
- At the end of the current file, specify the router's id and password as follows:

```
network=
{
ssid=network_id
psk=network_password
}
```
- Save and close the file by typing CTRL-X, then Y.
- Load the file to establish the connection with the command:
`wpa-supPLICant -iwlan0 -c /etc/wpa-supPLICant.conf & dhcpcd wlan0`

At this stage, the Raspberry-Pi is connected to the network. To retrieve its address IP, the command to execute on a terminal is: `hostname -I`

After that, we need to set the IP address attributed to the Raspberry-Pi on the file **config.properties**. We can find this file in the root of the application's directory. Some other attributes may be set like: **RPi_id** and **RPi_password** (To set the id and the password of the Raspberry-Pi).

Chapter 3

Launching the application

After having succeeded the configuration of the environment. You can now go on the website <https://github.com/AsmaBRZ/robot-explo/> and download the project. After that, launch Eclipse and import the project. Then, run the main class which bears the name of *Principal.java*, and whose complete path is: *robot-explo/Code/src/explorator/Principal.java*

The launching of the application opens a JME window. In this window, we can see a representation of the environment explored by the autonomous robot in real-time. We show in the figure below an instance of the execution of the application.

At the beginning, the application will ask the user to enter the name of the target object from the database. We remind that the target object is the object which the robot will try to find in its environment.

After that, the autonomous robot explores its environment, builds it and try to recognize the target.

Finally, the robot indicates whether it finds the target or not.

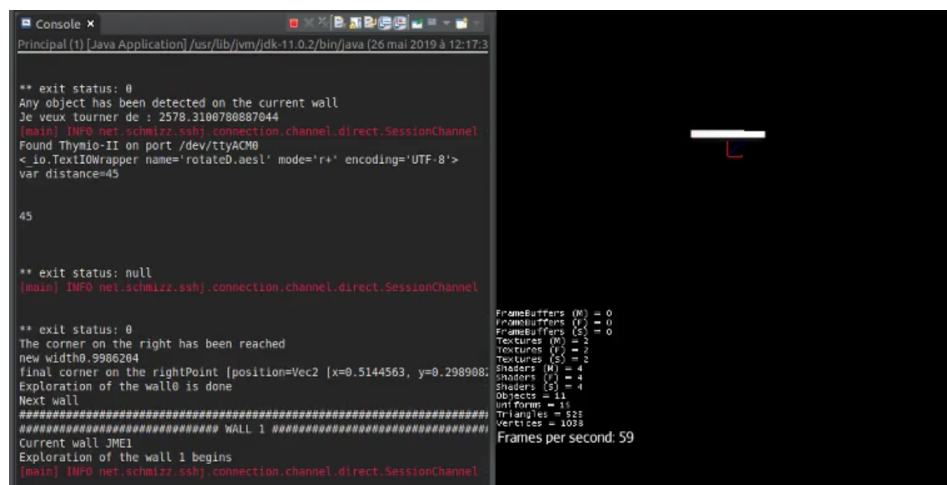


Figure 3.1: An instance of the robot's exploration