

e-puck2 Lab Induction

ACS6501, last updated on 10.10.2023

1 First Lab Session: Setting up the PC

You will be working in teams of up to two students (the allocation is available on Blackboard). When you turn on your PC, the purple bootloader screen will appear. It's important that you **select Ubuntu**, not Windows. When it boots, the password is `acselab`.

If you'd like to install the software on your own PC, see Appendices C (Windows) and D (Mac). This will allow you to work on the code in your own time using the preconfigured environment. Although you will not have a robot to run the code, you can check that your code compiles correctly. If you do not want to install the complete environment on your own PC, you can use a text editor of your choice to edit the necessary files. You should use Git to keep your files in sync across multiple computers (Section 4).

Notes:

- To open a terminal, use `ctrl + alt + t`
- To paste into a terminal, use `ctrl + shift + v` (or click the mouse wheel button after highlighting text)
- "Folders" are referred to as "directories" by computer scientists, so we use that name in this document.

1.1 Installing JAVA 8

JAVA 8 should already be installed. Verify this with the command below.

```
update-java-alternatives -l
```

If nothing appears, move to Appendix A and install as indicated there.

1.2 Creating your team directory

Every team should work in their own directory in `Documents`. Choose any name you want, but it's important there's **no spaces in the name**. We call this `RepoName` here.

At the end of the session, you should back your solutions up to GitHub (see Section 4 for details on how to do this) and **delete them from the computer** so other teams can't copy your work. At the start of the next session, clone your repository back into `Documents`.

To maintain consistency, **make sure your team directory has the same name as your Git repository**. Throughout the rest of the instructions, this is referred to as `RepoName` or your team directory.

1.3 Installing Eclipse

The IDE used in this lab is a special version of Eclipse, preconfigured to work with the e-puck2.

1. Download the `Eclipse_e-puck2` package for Linux by opening this link in a browser and pressing save: <https://tinyurl.com/UoS-epuck2>

2. Open a file explorer and navigate to the `Downloads` directory. Right click and press `Extract here`. This may take a while, but be patient: there's a circular progress bar in the top right of the file explorer.
3. When it's extracted, look inside this directory (called `Eclipse_e-puck2_Linux64_14_aug_2020` and copy the `Eclipse_e-puck2` directory to your team directory

You should end up with the following directories:

`~/Documents/RepoName/Eclipse_e-puck2`

1.4 Getting the source code

The code of the e-puck2 is open source and is available as a git repository. This repository contains the main microcontroller factory firmware together with the e-puck2 library. This library includes all the functions needed to interact with the robot's sensors and actuators; the factory firmware shows how to use these functions.

1. Open a terminal and navigate to your team directory: `cd ~/Documents/RepoName`
2. Download the source code from Github to this directory:
`git clone --recursive https://github.com/e-puck2/e-puck2_main-processor.git`
3. Remove the Git-related files from this directory to avoid conflicts later:
`rm -rf e-puck2_main-processor/.git*`

1.5 Familiarising yourself with the e-puck2

Take a look at Fig. 1 and familiarise yourself with the different parts of the e-puck2.

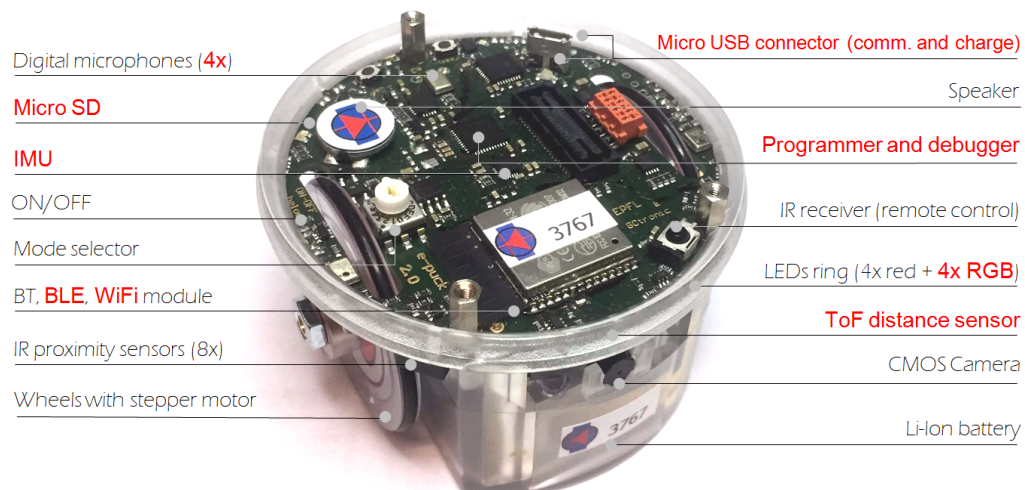


Figure 1. An e-puck2 and its components. Image reprinted from

https://www.gctronic.com/doc/index.php?title=e-puck2#Finding_the_USB_serial_ports_used

Websites that you may find useful for programming the e-puck2:

- EPFL (the developer of the robot) maintain an e-puck website: <http://www.e-puck.org/>
- In this lab, we use the e-puck2 library, an embedded system library for e-puck2. For more information: <https://github.com/e-puck2/>
- GCTronic (the manufacturer of the robot) provides a mini-doc and maintain a very resourceful Wiki: <https://www.gctronic.com/e-puck2.php>