

# Converting a wired Canon printer into wireless using a Raspberry Pi



| Design options | 1. Raspberry Pi + CUPS | 2. Raspberry Pi + Sockets | 3. Arduino    |
|----------------|------------------------|---------------------------|---------------|
| Cost           | 40\$                   | 40\$                      | 30\$          |
| Size           | Small                  | Small                     | Small         |
| Difficulty     | Easy                   | Hard                      | Hard          |
| Knowledge      | No need for knowledge  | Knowledge from course     | No knowledge  |
| Security       | No                     | No                        | No            |
| Time           | Slow                   | Fast                      | Slow          |
| Pins           | Some for LEDs          | Some for LEDs             | Some for LEDs |

## 1. Raspberry Pi + CUPS:

CUPS (Common UNIX Printing System) is a modular printing system for Unix-like computer operating systems which allows a computer to act as a print server.

This option is the easiest one, since we just have to download the CUPS software into the Raspberry Pi, get its IP address and type it with “:631” in the bar address in the browser of another computer.

## 2. Raspberry Pi + Sockets:

A socket is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to.

This option is more difficult than the first one, since we have to write our own server code and client code using sockets in order to send the file to be printed, print it, and tell both the server and the client that everything is okay.

We already have an idea about sockets thanks to one of the specialization courses, thus it is a chance for us to implement what we have learned. In contrast, we are not much sure about the security of the system, but we are willing to work more on it

## 3. Arduino:

Since we did not learn how Arduino communicates with the world, we do not know much about the feasibility of this option. (We googled this but it was not much of help). Also, Arduino doesn't have a USB port, unlike Raspberry Pi, so things will be more complicated.

⇒ **We are going for option 2 for various reasons:**

- More difficult than option 1 and less difficult than option 3
- We already have the necessary knowledge needed ○

Although security seems like a problem, but it is faster than the other two options

