DATA ACQUISITION TECHNOLOGIES AND SENSOR NETWORKS

Project Report

Smart Outlet: Access and Control Your Power Outlets Remotely

By Eno Ciraku, Diogo Cosin, & Ralph Florent

November 13, 2019

Abstract

[Here goes the absctract of the documentation...]

1 Introduction

[Here goes the introductory part of the documentation...]

2 Theoretical Background

[Here goes the theoretical part of the documentation...]

3 Intrumentation & Resources

Recalling that the SO prototype is the complete integration of a set of hardware and software, we detail in this section more in-depth specifications on them. Below are enlisted the tools used to set up and carry out successfully the current version of the project.

3.1 Hardware and other materials

The SO hardware refers to the physical components that add up to the *Webduino* module. Although there are many possible options in choosing different kinds of hardware to build up the circuitry, we opt for the most reasonable[1] choice, which is to reuse already-prebuilt modules and integrate them into one. Hence, in Table 1 are listed the items with their corresponding details.

Hardware & Other Materials			
	Series	**Quantity**	**Cost**
Microcontroller Board	MEGA 2560 R3	1	€ 14,99
4-Channel Relay	GE-EL-SM-006	1	€ 6,99
Wi-Fi Module	ESP8266-01	1	€ 6,99

Table 1: Detailed information on the hardware and materials used for the SO prototype.

Additionally, other useful materials such as:

- 12x male-to-male jumper cables (20 cm)
- 12x male-to-female jumper cables (20 cm)

[1] Building a separate module from scratch requires time and work, plus other tasks to refine a working component.

- 12x female-to-female jumper cables (20 cm)
- 1x access point or router (tp-link TL-WR940N)
- 2x computing devices or laptop (Lenovo T460p and Macbook Air)

remain useful to connect the different components and have them work as one stable module. More technical specifications are given on each one of the materials and devices, including their working conditions, in the section *State of Art*.

3.2 Tools and software

Next, we present the set of tools and software that are used at the time of implementing the prototype:

- Operating systems (GNU/Linux, Mac OS, and Windows)
- Visual Studio Code (lightweight text editor)
- Git[2] (version control)
- GitHub (web-based hosting service for Git versioning system)
- Jupyter Notebook (workspace for scripting and simulation)
- Arduino IDE (development environment for Arduino boards)

Regarding the software versions, it is highly recommended to use the exact versions mentioned in Table 2 to avoid conflicts and compiling errors. On the other hand, the developer can always dig into the breaking changes (if that is the case) that might requires to refactor part of the implementation to have a fully working prototype. However, it is recommended to check the changelog of the updates/releases, if any.

Note that some tools mentioned above are just a matter of personal preferences. Other preferred options are more than welcome as long as the developer keeps in mind development speed and productivity. For example, many developers would choose Sublime over *Visual Studio Code*. But they both end up facilitating the same routine: text edition.

3.3 Programming languages

Finally, we use the following programming languages:

- C/C++ (for the webduino)

[2]Git is also available as a bash emulation for other platforms for free (e.g., Git Bash for Windows).

- Python (for the web API service)
- Angular Framework - JavaScript/TypeScript (for the web application)

Important: *Though we highly recommend that the exact versions of the software and the exact series/models of the materials/devices are used to test out or replicate this project, keep in mind that these hardware might no longer be available in the market as well as the software components might be outdated at some point in time. If that is the case, stay alerted to the updates as we intend to support this project until 2022.*

Appendix A Code Repository

All the code implemented during the execution of the prototype described in this report is available on the GitHub repository https://github.com/ralflorent/smart-outlet.