Design of a semi-automated home brewing system

PROJECT REPORT

Autor: David Catalán Conde
Director: Manuel Moreno Eguílaz
Convocatòria: July 2017
Abstract

Beer is, by far, the most widely consumed alcoholic drink and the third most popular drink overall, only surpassed by water and tea. But no, this project will not consist of explaining its characteristics or the alleged benefits of this golden drink.

In recent years, among the growth of craft beer, the number of people who make their own beer at home has been increased, those are called homebrewers, and it seems that this tendency will continue expanding. This project aspires to create a system that allows to speed up this process making use of the technology, also, set up the basis to carry forward the idea and materialize it.

The first stage of this work will be to outline the fundamentals of the idea that we want to achieve and how. Designing the equipment, as well as which components will belong to, will let to see the final elements needed and how they will be controlled. In turn, the boundaries must be defined to know until which part of the process the system will take part in.

Once all the elements of the system and the steps that have to be controlled have been defined, set in motion the design of the software in charge will become possible.

An application will be created in order to read recipes in a standardized and widely used format by the entire brewing community. This program will also show all the relevant information of the corresponding recipe, its characteristics, ingredients, duration of each step and all those parameters that are relevant to the process. At the same time, the program itself will guide the user through the whole procedure, allowing him or her to know the temperatures in real time, control the actuators and ultimately, simplifying the way to brew beer.

From the start, the plan is to build up the project as open-source to allow its free exploitation and provide the possibility to enhance it to keep it growing.
Index

ABSTRACT .................................................................................................................. 1
INDEX ...................................................................................................................... 3
GLOSSARY ............................................................................................................... 5
LIST OF FIGURES ................................................................................................. 8
LIST OF TABLES ................................................................................................. 10
1. INTRODUCTION ............................................................................................... 11
    1.1. BACKGROUND ........................................................................................... 11
    1.2. MOTIVATION ............................................................................................ 12
2. OBJECTIVES .................................................................................................... 13
    2.1. SCOPE ....................................................................................................... 13
3. PRECURSOR PROJECTS ................................................................................... 14
    3.1. COMPARISON CHART ............................................................................. 16
4. PREVIOUS KNOWLEDGE ................................................................................. 18
    4.1. BEER ......................................................................................................... 18
        4.1.1. Water .................................................................................................. 18
        4.1.2. Cereals .............................................................................................. 19
        4.1.3. Hops (Humulus lupulus) .................................................................... 19
        4.1.4. Yeast (Saccharomyces cerevisiae) ...................................................... 20
    4.2. THE CRAFT BEER SCENE ......................................................................... 21
    4.3. LEGALITY ................................................................................................. 22
    4.4. THE HOMEBREWING PROCESS ............................................................... 23
5. PROPOSED SOLUTION ..................................................................................... 27
    5.1. DESIGN CRITERIA ..................................................................................... 27
    5.2. PHYSICAL DESIGN .................................................................................. 27
        5.2.1. Electronics ....................................................................................... 33
    5.3. OVERVIEW. FINAL EQUIPMENT .............................................................. 37
    5.4. SOFTWARE DESIGN ................................................................................ 40
        5.4.1. BeerXML Parser ............................................................................. 41
        5.4.2. Software Structure .......................................................................... 42
        5.4.3. GUI ................................................................................................. 47
6. CBBS. FINAL IMPLEMENTATION ................................................................. 52
7. VALIDATION TESTS ....................................................................................... 58
    7.1. FOLLOWING THE PROCESS ................................................................. 59
8. PROJECT SCHEDULING AND BUDGET ..................................................... 68
9. SOCIAL AND ENVIRONMENTAL IMPACT .................................................. 71
10. CONCLUSIONS ........................................................................................................ 74
10.1. FUTURE IMPLEMENTATIONS ........................................................................ 75
ACKNOWLEDGMENTS .................................................................................................... 76
BIBLIOGRAPHY AND REFERENCES ........................................................................... 77
REFERENCES: ............................................................................................................... 77
LICENSE ....................................................................................................................... 79
## Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABV:</td>
<td>Alcohol by Volume [%] - Millilitres (mL) of pure ethanol present in 100 mL of solution at 20 °C.</td>
</tr>
<tr>
<td>ALDEHYDE:</td>
<td>Organic compound containing a functional group with the structure –CHO. Provides “green” flavours, as green apples, grass or fry cider.</td>
</tr>
<tr>
<td>ALPHA/BETA ACIDS:</td>
<td>[%] - Chemical compounds found in the resin glands of the flowers of the hop plant. They are the main source of beer bitterness.</td>
</tr>
<tr>
<td>ARDUINO:</td>
<td>Open source single-board for building digital devices and interactive objects.</td>
</tr>
<tr>
<td>ATTENUATION:</td>
<td>[%] – Percentage of converted sugars into alcohol and carbon dioxide by the fermentation process.</td>
</tr>
<tr>
<td>BARLEY:</td>
<td>Cereal grain. One of the main ingredients to brew beer.</td>
</tr>
<tr>
<td>BATCH:</td>
<td>Quantity of beer made at the end of the process.</td>
</tr>
<tr>
<td>BLEB:</td>
<td>Bulge or protrusion of the plasma membrane of a cell. In this case, yeast.</td>
</tr>
<tr>
<td>BREWER’S WINDOW:</td>
<td>Adequate temperature range to perform the mash step and obtain the desired sugars.</td>
</tr>
<tr>
<td>CRAFT BEER:</td>
<td>Beer made by independently-owned commercial breweries or homebrewers that employ traditional brewing methods emphasizing its flavour and quality.</td>
</tr>
<tr>
<td>DEXTROSE:</td>
<td>$C_6H_{12}O_6$ Simple sugar obtained from the grains during the mash.</td>
</tr>
<tr>
<td>DIACETYLE:</td>
<td>$C_4H_6O_2$ Organic compound naturally formed in alcoholic beverages. Gives buttery, rancid and undesired sweet flavours.</td>
</tr>
<tr>
<td>DIODE:</td>
<td>Two-terminal electronic semiconductor device that conducts primarily in one direction.</td>
</tr>
<tr>
<td>EBC SCALE:</td>
<td>European Brewery Convention Scale. Provides the beer colour using a spectrophotometer to measure the attenuation of light of a particular wavelength, 430 nanometres, as it passes through a liquid sample.</td>
</tr>
<tr>
<td>EFFICIENCY:</td>
<td>Measurement of potential fermentables converted into sugar in the wort during the mash.</td>
</tr>
<tr>
<td>ENZYME:</td>
<td>Macromolecular biological catalysts. Related to beer, these enzymes are developed during the malting process and reduce starches and proteins during the mashing. Contributes with clarifying and providing body.</td>
</tr>
<tr>
<td>ESTER:</td>
<td>Chemical compounds derived from acids. Brings fruity flavours to beer.</td>
</tr>
</tbody>
</table>
**ETHANOL:** Principal alcohol found in alcoholic beverages.

**FERMENTATION:** Metabolic process that converts sugar to acids and gases, or alcohol.

**FUSEL ALCOHOL:** Mixture of several alcohols produced as a by-product of alcoholic fermentation. Provides an unbalanced major alcoholic character.

**GLUCOSE:** See: DEXTROSE

**GLYCOLYSIS:** Metabolic pathway that converts glucose into an acid to be converted into ethanol via fermentation.

**GRAVITY:** Relative density of the beer compared to water.

**GUI:** Graphical User Interface. Type of user interface that allows users to interact with electronic devices.

**HERMS:** Heat Exchanged Recirculating System.

**HOP:** Flowers of the hop plant Humulus lupulus. They are used mainly as a flavouring and stability agent in beer.

**IBU:** International Bitterness Unit. Iso-alpha acids milligrams for litre of wort. The major the number, the major the bitterness of beer.

**IRISH MOSS:** Algae that helps to remove proteins in suspension from the wort and clarifies the final product.

**KETONE:** Organic compound with structure RC(=O)R'. Similar properties than aldehydes.

**MALT:** Germinated cereal grains that have been dried in the malting process.

**MALTING** Process to convert raw grains into malt. It consists of various internal procedures: Drying, Cleaning, Storing, Steeping, Germination, Kilning, Deculming and Second Cleaning.

**MASHING:** Process of combining a mix of milled malt and other grains and hot water to break down the starch into sugars to produce wort.

**MICROBREWERY:** Independently owned brewery that produces small amounts of beer.

**OXIDATION:** Chemical reaction in which the oxidation states of atoms are changed. In beer, it is not desirable to happen since it can originate unpleasant flavours.

**PCB:** Printed Circuit Board. Non-conductive substrate panel which supports and electrically connects electronic components using conductive tracks.

**PRIMING:** Action based on adding sugars to beer just before bottling it, feeding the yeast to keep fermenting and produce CO₂.
<table>
<thead>
<tr>
<th><strong>Pull-up Resistor:</strong></th>
<th>Resistor connected between a signal conductor and a positive power supply voltage to ensure that the signal will be a valid logic level if external devices are disconnected or high-impedance is introduced.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Python:</strong></td>
<td>Programming language for general-purpose programming.</td>
</tr>
<tr>
<td><strong>Raspberry Pi:</strong></td>
<td>Single-board computer to develop all kinds of computing and electronics projects.</td>
</tr>
<tr>
<td><strong>Relay:</strong></td>
<td>Electrically operated switch.</td>
</tr>
<tr>
<td><strong>Rims:</strong></td>
<td>Recirculating Infusion Mash System</td>
</tr>
<tr>
<td><strong>Script:</strong></td>
<td>Computer program written in a scripting language.</td>
</tr>
<tr>
<td><strong>Sparging:</strong></td>
<td>Trickling water through the grain to extract sugars during the mash.</td>
</tr>
<tr>
<td><strong>Stage-Tier Rig:</strong></td>
<td>Structure dedicated to brew beer. It has the necessary elements compactly arranged to carry out the process adequately</td>
</tr>
<tr>
<td><strong>Starch:</strong></td>
<td>Polymeric carbohydrate consisting of a large number of bounded glucose units.</td>
</tr>
<tr>
<td><strong>Tannins:</strong></td>
<td>Astringent biomolecule that helps to precipitate proteins producing the haze during the boiling step, which should be removed.</td>
</tr>
<tr>
<td><strong>Terpenes:</strong></td>
<td>Large class of organic compounds produced by plants (hops in this case) that can provide a wide range of aromas.</td>
</tr>
<tr>
<td><strong>Tkinter:</strong></td>
<td>Standard Python interface to the Tk GUI toolkit.</td>
</tr>
<tr>
<td><strong>Transistor:</strong></td>
<td>Semiconductor device used to amplify or switch electronic signals and electrical power.</td>
</tr>
<tr>
<td><strong>Wort:</strong></td>
<td>Liquid extracted from the mashing process during the brewing of beer or whisky. It contains the sugars that will be fermented.</td>
</tr>
<tr>
<td><strong>Xml:</strong></td>
<td>Markup language that defines a set of rules for encoding documents with tags in a format readable by humans and machines.</td>
</tr>
<tr>
<td><strong>Yeast:</strong></td>
<td>Single-celled microorganisms. By fermentation, the yeast (Saccharomyces cerevisiae in brewing) converts carbohydrates to CO₂ and alcohols.</td>
</tr>
</tbody>
</table>
List of figures

Fig. 1. CBBS LOGO........................................................................................................ 12
Fig. 2. BrewPi LOGO.................................................................................................... 14
Fig. 3. CRAFTBeerPi LOGO...................................................................................... 14
Fig. 4. BrewTroller LOGO.......................................................................................... 15
Fig. 5. Beersmith LOGO............................................................................................. 15
Fig. 6. Hop Contribution by Time Chart...................................................................... 20
Fig. 7. Homebrewing Stands Configuration................................................................. 28
Fig. 8. RIMS System................................................................................................... 29
Fig. 9. HERMS System............................................................................................... 29
Fig. 10. Hydraulic Scheme.......................................................................................... 30
Fig. 11. 2-Way and 3-Way Valves............................................................................. 30
Fig. 12. Plate Heat Exchanger..................................................................................... 31
Fig. 13. Coil Heat Exchanger...................................................................................... 31
Fig. 14. Female and Male Cam Locks......................................................................... 31
Fig. 15. Electric Elements Scheme.............................................................................. 32
Fig. 16. Brushless DC Pump....................................................................................... 32
Fig. 17. Folded Tubular Heating Element ................................................................. 32
Fig. 18. Raspberry Pi V3............................................................................................ 33
Fig. 19. DS18B20 Sensor............................................................................................ 34
Fig. 20. PT100 Sensor................................................................................................ 34
Fig. 21 DS18B20 Schematic...................................................................................... 35
Fig. 22. Relay Schematic............................................................................................ 36
Fig. 23. Expansion Shield Used for This Project......................................................... 37
Fig. 24. Hot Liquor Tank............................................................................................. 38
Fig. 25. Mash Liquor Tank.......................................................................................... 38
Fig. 26. Boiler Tun........................................................................................................ 38
Fig. 27. Single Tier Structure..................................................................................... 39
Fig. 28. Designed Pump + Valves Assemblies............................................................ 39
Fig. 29. Designed Plate Heat Exchanger.................................................................... 39
Fig. 30. Proposed Solution.......................................................................................... 39
Fig. 31. Final Ensemble............................................................................................... 40
Fig. 32. CBBS Scheme................................................................................................ 43
Fig. 33. Steps Section................................................................................................ 44
Fig. 34. Plot Funcions Block...................................................................................... 45
FIG. 35. TIMER FUNCTIONS BLOCK .......................................................... 45
FIG. 36. HELPER FUNCTIONS BLOCK ..................................................... 46
FIG. 37. CONTROL FUNCTIONS BLOCK .................................................. 46
FIG. 38. SETUP CODE ............................................................................. 47
FIG. 39. TKINTER. MAIN WIDGETS ....................................................... 48
FIG. 40. TKINTER. FRAME STYLES .......................................................... 49
FIG. 41. TKINTER. BUTTON WIDGET ....................................................... 49
FIG. 42. TKINTER. BUTTON MESSAGE .................................................... 50
FIG. 43. TKINTER. LISTBOX WIDGET ..................................................... 50
FIG. 44. TKINTER. CANVAS WIDGET ...................................................... 51
FIG. 45. TKINTER. NOTEBOOK WIDGET ............................................... 51
FIG. 46. CBBS. MAIN SCREENS .............................................................. 52
FIG. 47. CBBS. GENERAL INFORMATION ............................................... 54
FIG. 48. CBBS. STYLE AND RECIPE CHARACTERISTICS ...................... 54
FIG. 49. CBBS. COLOR BY RECIPE ......................................................... 54
FIG. 50. CBBS. INGREDIENTS LIST ....................................................... 55
FIG. 51. CBBS. SELECT RECIPE WINDOW ............................................. 55
FIG. 52. CBBS. ABOUT WINDOW ........................................................... 56
FIG. 53. CBBS. RECIPE PROGRESS STEPS ........................................... 56
FIG. 54. CBBS. MANUAL CONTROL ....................................................... 57
FIG. 55. CBBS. TIMER SCREEN .............................................................. 57
FIG. 56. CBBS. PLOTS AREA ................................................................. 57
FIG. 57. BEERSMITH RECIPE ................................................................. 58
FIG. 58. GANTT CHART. TFG TIMELINE DEDICATION ........................ 68
FIG. 59. CO₂ EMISSIONS ...................................................................... 72
FIG. 60. CO₂ PRODUCED BY BEER ...................................................... 72
List of tables

TABLE 1. ALTERNATIVES. COMPARISON CHART. ................................................................. 16
TABLE 2. FERMENTATION CHEMICAL EQUATION BALANCED............................................. 26
TABLE 3. INTELLECTUAL COST. ....................................................................................... 68
TABLE 4. MATERIAL COST. .............................................................................................. 69
TABLE 5. ENERGETIC COST. ............................................................................................. 69
TABLE 6. TOTAL PRICE. ..................................................................................................... 70
1. Introduction

“We brewers don’t make beer, we just get all the ingredients together and the beer makes itself.”

- Fritz Maytag, former owner of Anchor Brewing Company

This citation summarizes the main essence not only from this project but also from the homebrewing basis.

What this introduction wants to emphasize is the importance to know that homebrewers are just partakers in the process of brewing beer. The best master brewer will never get to brew a good beer with low quality ingredients, using a deficient equipment or even if the control on the procedure is inadequate.

This thesis will focus on this last aspect mostly. By enhancing and establishing a better dominion during the brewing, the margin of error is reduced, permitting a higher performance and going a step further on the preparation, allowing, like the quote says, to obtain flawless beers.

1.1. Background

CBBC, Catalán Brothers Brewing Company, was born about 4 years ago in a kitchen thanks to my brother and me. Shortly after discovering the gorgeous world of craft beer, we raised the idea to go a step further and start brewing our own beer. We were not aware of what was needed, we did not know if we could overtake the situation and we were unsure about everything, but due to all information found on the Internet, the help from the skilled people on this field and the eagerness given over to drink our favourite drink made by us, we took the first steps to do it.

An “All Grain” basic equipment was bought firstly, composed by the sufficient and necessary elements to brew about 20 litres of beer. From there, during all this time, a few of improvements have been made by buying new and better components. The last upgrade was to control the refrigerator with an Arduino board programmed to be capable of switch the power to maintain the temperature on a specific range during the fermentation.

So, why not try to automate more parts of the process?
Whit this question, the name of CBBS, – Catalán Brothers Brewing System, was born. The idea was to create a software that could guide and simplify the brewing process. CBBS has been for few years just a preliminary sketch, always postponing it for “whenever I have time”. Until now, this TFG has offered the option to carry out with the idea and make it possible.

1.2. Motivation
The main incentive that has made feasible this project is the keen interest in the beer per se, as well as in the procedure to make it. As an engineer, I have always had a special interest in production processes and automation, and thanks to the fact that the beers made during this time have been quite decent, has led to look for a little more and not stop growing as a homebrewer.

An aspect that should be considered for the realization of this project is all the help received from the homebrewer community, thousands of people, both locally and internationally, which make their own beer, participate regularly in forums, and attend to craft beer fairs. Due to all the shared knowledge and their contributions, it had been decided to put my grain of sand in this community writing the project report and the code in English, so it can reach more people around the world. From all this, the idea of creating our own system was originated, not only looking for the sake of us, but also to engage in the homebrewer world actively.

Fig. 1. CBBS logo.
Source: own.
2. Objectives

The objectives of this project were clear since the beginning. The most relevant ones are:

- Process definition and determination of the procedures to act on.
- Equipment design and hardware selection.
- Build and develop a control application to regulate the setup and help to carry out the whole process.
- Create a graphical user interface (GUI) with a responsible design granting a greater control over the application.
- Generate and simulate a new recipe to seek to ensure the proper functioning.
- Steer the idea for a future implementation, always watching to keep alive the homebrewing essence.

2.1. Scope

In order to define the boundaries of the project, its scope must be considered:

- The project is based on the creation of a system to help to brew beer, so any beer will be made by the end of the project.
- The physical equipment will only be designed, with the exception of the control elements already bought.
- The solution will not be fully automated, since the user will need to interact and keep an eye on the process until its end.
- The used recipes must be in a specific extension and properly fulfilled. Also, they are limited by the own equipment.
3. Precursor projects

For many years, homebrewers have used a myriad of tools that have helped with the artisan brewing process. From a basic chronometer and a thermometer to control the most basic aspects, until a computer-controlled system that takes care of every detail of the manufacturing. In recent years, with the rise of technologies and the possibilities to have a custom-made equipment, some solutions have emerged and become commercialized, being widely used by thousands of people around the world.

At this point, some of the possibilities that currently exist in the market will be exposed.

BrewPi[1]

BrewPi is probably the most widely used option for brewing control around craft beer, although it is also the most limited of the following.

The system consists of a Raspberry Pi[2], connected to different temperature sensors pertinently coupled to a refrigerator, and two additional elements, one heater and another for cooling. This system regulates the temperature of the refrigerator and thus to control the fermentation. In addition, it can be connected to the network to warn the user of possible sudden temperature changes or problems during the fermentation. It has its own online store to buy several components and an installation guide to be able to have the small setup ready in few hours.

Nowadays, the authors are making some improvements in the process, manufacturing their own printed circuit boards (PCB) and designing a system that also regulates mash and boil processes.

CraftBeerPi[3]

CraftBeerPi is a relatively new concept, but more complete than any other system. It consists of a Raspberry that works as a central computer, which connects three relays that control a pump and two heating resistors, in addition to three sensors to measure the temperature of each container.

Fig. 2. BrewPi logo. Source: https://www.raspberrypi.org

Fig. 3. CraftBeerPi logo. Source: http://web.craftbeerpi.com/
This system can be configured to control from mashing to fermentation, allowing a greater control and autonomy of the process. It measures temperatures in real time and notifies the user of any anomaly in the system.

**BrewTroller[^4]**

BrewTroller is the most expensive option but at the same time the more automated. It is the only option between the previous ones that works with a PCB of its own brand and an Arduino[^5] connected to it. This system, besides monitoring the mashing and boiling steps, also controls the valves of the equipment, allowing to speed up the process being less likely to make any mistake. It also has its own store to buy the control elements and the different subsystems that form the equipment.

**Beersmith TM[^6]**

Beersmith is not exactly a control system since it works autonomously and separately from the main process. But it is thanks to this program that others have emerged. That is why it has been chosen to be in this list. Beersmith is a software that facilitates the creation, modification and diffusion of recipes as well as a section that helps to follow the process allowing to perform modifications and corrections. Thanks to this company, a significant portion of the homebrewer sector has been unified as well as the community that has been commented before expanded. Another facet to emphasize is that the creators of Beersmith bring into existence the standardized recipe format BeerXML[^7].
3.1. Comparison chart

Once these systems have been presented, a summary revealing the different options will be shown in Table 1.

<table>
<thead>
<tr>
<th>Feature</th>
<th>BREW Pi</th>
<th>CRAFT BEER Pi</th>
<th>BREW TROLLER</th>
<th>CBBS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modify recipes</td>
<td>✘</td>
<td>✘</td>
<td>✘</td>
<td>✘</td>
</tr>
<tr>
<td>Recipe reading</td>
<td>✘</td>
<td>✓</td>
<td>✘</td>
<td>✓</td>
</tr>
<tr>
<td>Mash control</td>
<td>✘</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Boil control</td>
<td>✘</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Fermentation control</td>
<td>✓</td>
<td>✓</td>
<td>✘</td>
<td>✘</td>
</tr>
<tr>
<td>Addition warnings</td>
<td>✘</td>
<td>✘</td>
<td>✘</td>
<td>✓</td>
</tr>
<tr>
<td>Real-time plots</td>
<td>✓</td>
<td>✓</td>
<td>✘</td>
<td>✓</td>
</tr>
<tr>
<td>User communication</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✘</td>
</tr>
<tr>
<td>Other uses</td>
<td>✘</td>
<td>✘</td>
<td>✘</td>
<td>✓</td>
</tr>
<tr>
<td>Open source</td>
<td>✘</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Approx. Cost [€]</td>
<td>300 – 1000</td>
<td>200 – 800</td>
<td>600 – 950</td>
<td>~150</td>
</tr>
</tbody>
</table>

Table 1. Alternatives. Comparison chart. Source: own.

1 This value is given by the range between the most basic and the full kit exposed on their webpages, and it reflects the total price it will cost to use this system at home. In case of CBBS, the cost is only the value of the used components.
It can be seen how the system on which this work is addressed, intends to be quite complex but more competitive, economically talking, than any of the others shown up. Two features were wanted to be added after observing that the other options did not count with them. CBBS will have a real-time warning system that will notify when it is time to make an addition (hops or any miscellaneous), preventing the requirement to put a separate alarm to avoid the chance of forgetting.

In addition, the fact that this software is a script running on an operating system, and not a dedicated OS by itself, allows the controller to be used for other issues and not just dedicated to brewing.
4. Previous knowledge

In order to understand how the program features behaves or are based on, some basic considerations should be taken into account and of course, it is essential to know how the process of brewing works. To do this, we will go a little further and a brief introduction about beer will be explained, the basic ingredients and processes that are carried out.

4.1. Beer

Investigators have been found signs that show that fermented beverages were begun to develop from grain and roots from 11000 BC. However, the earliest mention of the closest definition of beer as we know nowadays was over 5,000 years ago in the ancient Egypt.\[8\] Then, it was made from baked barley bread. During the building of the Great Pyramids in Giza, each worker got a daily ration of around four litres of beer, which served as both nutrition and refreshment that was crucial to the pyramids' construction.

Essentially, beer is the product of the fermentation of barley malt extracts by the action of yeast. While malt and yeast make important contributions to the profile of beer, its quality and character depends equally on the water and hops used in its production. Today, there are hundreds of documented beer styles, and all of them mainly consist of four essential ingredients: water, cereals, hops and yeast. The quality of these raw materials has a decisive influence on the quality of the obtained beer. Understanding the properties of those raw materials, their influence during the process and on the final product, provides the basis for their use and processing. For more information about Beer Styles, please see Annex 3.

The role of each ingredient will be briefly explained with regard to figure out how they affect.

4.1.1. Water

As a matter of fact, beer is composed mostly of water, sometimes being the 95% of the final product. And even then, only a slightly amount of water required is used directly in beer, while another part is required for cleaning, rinsing and other purposes. The convenient procurement and treatment of the water are of particular importance to the brewer, since the quality of the water influences substantially on the quality of beer made from it.\[9\]
Before the arrival of modern science, beer styles were limited by the type of water available in the region. Scientific knowledge and technology have enabled the artisan brewers to design and manufacture beers that are not restricted by geographical boundaries.

4.1.2. CEREALS

Grain is the main raw solid material for brewing. This grain comes from cereals, usually barley, but may be from wheat, rye, oats, corn, rice, etc., partially germinated and dried. Malting and maceration processes convert starches into different fermentable and non-fermentable sugars.

The load is composed of the base malt, special malts and other attachments (raw cereals). Those are selected according to the desired beer style and profile, contributing on the flavour and on the body of the final drink. Malt is also responsible of beer colouring. The amount of colour given by it is determined during the process of kilning\[^{10}\] throughout the malting, where the high temperatures of this process produce different levels of toast in the grains through the Maillard reaction.\[^{11}\] It is expressed with a numeric scale and it is measured in Europe with the EBC system.\[^{12}\] For more information about EBC Scale, please see Annex 4.

4.1.3. HOPS (HUMULUS LUPULUS)

It is used since the Middle Ages in the beer production to avoid the detriment of other ingredients. The dry cones of this plant of the family Cannabaceae contain bitter resins and ethereal oils that provide almost all the bitterness of beer. Hops also have antioxidant properties and a great chemical importance right through the whole process. Late hops additions and the different varieties used have a great influence on beer aroma and taste.

Dried hops are composed of:

- Bitter compounds/resins 18.5%
- Hop oils 0.5%
- Tannins 3.5%
- Proteins 20.0%
- Mineral substances 8.0%

The rest is composed by cellulose and other substances that are not of importance for brewing. The most important components for brewing are these bitter compounds and hop oils.
Bitter compounds, mainly (α)alpha acids and (β)beta acids, are aggregates that isomerize when are heated and are responsible for the bitter character of beer. Alpha-acids are measured in IBU[13] (International Bittering Units) and are defined as iso-alpha acids milligrams for litre of wort.

Hop oils are very soluble and volatile compounds, including esters, fusel alcohol, aldehydes, ketones, terpenes, and others, which contribute on the different notes of flavour and aromas. The different hop additions, which vary according to the recipe, are added during the boiling, providing different characteristics depending on the moment in which there are introduced. It is important to use fresh hops, since these compounds, as explained before, are lost causing the hops oxidation, providing unpleasant flavours or aromas.

![Graph](http://www.homebrewtalk.com)

*Fig. 6. Hop contribution by time chart. Based on a plot made by http://www.homebrewtalk.com.*

**4.1.4. Yeast (Saccharomyces cerevisiae)**

Yeast is a single-celled fungus. There are many species, and they are classified according to the characteristics of their cellular form, reproduction, their physiology and their habitat. Their natural surroundings are varied: fruits, leaves, flowers, and even the human skin. The characteristic of our interest is its ability to metabolize sugars and produce alcohol; and the class that interests us is *Saccharomyces Cerevisiae*, extraordinarily qualified to do so. The cells, if they are healthy and are in the right conditions, reproduce by budding, divide into two,
and the smallest part, the bleb, detaches from the parent cell, grows and continues reproducing.

The cycle continues while there is oxygen, and once the cells have used all the oxygen, they begin to metabolize sugars and to produce ethyl alcohol, CO₂, heat, and other by-products (esters, diacetyl, aldehydes...) that can also modify the taste and the aroma.

It should be mentioned that it is the most “dangerous” ingredient among the others. A deficient yeast preservation, an unwanted mixture, a misstep during cleaning, can allow other wild yeast (found in the environment) or bacteria to react with the beer spoiling and contaminating the whole lot. However, it is also true that there are certain styles that interacts with these elements to ferment the wort, although they are a minority.

4.2. The craft beer scene

The terminology “craft beer” has origin in the United Kingdom in the late 1970s and was used to describe the new generation of small breweries that focused on the traditional production of ale beers.

Although originally “microbrewery” was used to describe the size of breweries, used to designate breweries producing less than 15,000 litres of beer per year. Gradually it came to reflect the attitude, the alternative approach to the brewing, and the exceptional adaptability and customer service.[14]

Microbreweries have adopted a different marketing strategy, considerably far away from those conceived by mass-market breweries, offering products that compete according to their quality and diversity, rather than low prices and advertising.

It should be interesting to mention that in 2008, the number of microbreweries in Spain was only 21, but in 2011, the “revolution of artisan beers” was exposed,[15], and more recently, by 2013, the trend had spread to a large number of regions. In mid-2015, there were around 400 microbreweries spread all over the country. To date (June 1, 2017), there are 770 companies registered as such.[16]

It is noteworthy, and as a small tribute, that this revolution was initiated thanks to the initiative of Steve Huxley [1950-2015], a brewmaster who dedicated his life to the craft beer elaboration and the dissemination of them.
4.3. Legality

Surprisingly, one of the most asked questions by the general public is whether making an alcoholic beverage at home is legal or not, or even if a homebrewer, as an individual producer, could sell his beer.\[17\]

Well, starting from the roots of this point in question, it should be interesting to study the current regulations applied today. And therefore, it is necessary to read the “Real Decreto 678/2016, de 16 de diciembre, por el que se aprueba la norma de calidad de la cerveza y de las bebidas de malta”, from the Boletín Oficial del Estado (BOE), the official gazette of the Government of Spain.\[18\]

In can be observed how it is not explicitly specified which requisites exist to commercialize craft beer, but what characteristics should have beer to be considered as such, regardless of whether it is craft or not.

Likewise, it does not indicate what health or legal requirements should exist to be able to sell beer. For this reason, and getting in touch with the Barcelona City Council, it has been known that to have the right to sell beer, it is needed:

- Sanitary Registration
- Industrial Registration
- Opening licence
- CAE (Código de Actividad y Establecimiento)
- Bottling license
- Civil liability insurance

Taking into account of no regulatory authority will accept a domestic kitchen as a suitable place to brew beer, the Sanitary Registration will not be procured, so it will not be possible to commercialize (in a legal way) craft beer made at home.
4.4. The homebrewing process

Now, to finish off and begin with the explanation of the control tool, this section will go in depth into the brewing process. Already knowing the main ingredients, it can be seen how they are employed and when they join this process.

- **Cleansing**

Before starting to brew beer it must be ensured that all the equipment is cleaned, disinfected, and if necessary, sterilized. It is also crucial to keep an eye at all time on which materials are being used and what type of cleaning agents can be applied on them, since an error could mean the breakage of any of the equipment elements.

Detergents or disinfectants may be used depending on the amount of dirt to be removed and the use of this equipment. Of course, every item should be rinsed thoroughly before using it.

- **Preparation**

The whole preparation phase starts with choosing the recipe, followed by the purchase of the consequent ingredients, the arrangement of all equipment and the necessary material, and getting ready the ingredients. During the preparation, milling the grains accurately is vital to obtain a good beer.

Milling is the process of crushing malt or different grains in smaller pieces in order to enable enzymes to act on the malt components and to decompose them during maceration. It should be taken into account that although the husks do not contain fermentable sugars, they must be treated carefully and not get rid of them, since they are needed as a natural filter all along the mash step.

Additionally, an extreme caution must be taken with the dust generated when grinding the grain, in high concentrations it can be explosive and ignite due to a spark originated from the friction between the different mill parts. Another consideration about this malt powder is that it should be set apart from the production space, as it may contain traces of unwanted yeasts which could contaminate the beer.

Depending on the class of malt, its water content or the sugar amount, the grain fragments must be finest or thickest, as the quality of the grinding affects the maceration process, the ability to filter the wort and the process efficiency as well as the beer taste and character.
**Mashing**
Mashing is the most important step in the wort production. During this procedure, the fermentable grain is hydrated, the malt enzymes are activated and the grain starches are converted into fermentable sugars. There are several fundamental groups of enzymes involved in the starch conversion. These enzymes are a kind of proteins that provoke chemical reactions without changing their own structures. This part of the process is so extensive and complex that several reports could be done, although as a summary, it should be emphasized that transformations during maceration are of decisive importance in many factors, such as flavour, colour, aroma, body and acidity.

There is the so-called Brewer’s Window\footnote{\cite{19}}, which is the temperature range to perform the mash and obtain the desired sugars, knowing that if we want more alcoholic and dry beers, we must macerate between 62 and 65 °C, or between 68.5 and 70 °C to obtain beers of lower content of alcohol, sweeter and with fuller body.

This step can be done from its simplest version, “simple infusion”, where all the grain is added since the beginning, fixing a single temperature. More complex forms, as the “step infusion”, where the grain is added at different stages and modifying the temperature according the time to obtain diverse sugar types.

At the same time, if the wort is recirculated during the maceration process, the efficiency, the production of sugars and, in general, the quality of the beer, is increased significantly.

**Sparging**
Sparging is trickling water through the grain to extract the remaining sugars and transfer the wort to the boiler tun. This is a delicate step, since if it is done at wrong temperature or pH, this step will extract tannins from the chaff, resulting in an unpleasant bitter brew. Typically, is used 1.5 times more water for sparging than for mashing.\footnote{\cite{20}}

**Boiling**
Once the wort is in the boiler tun, it is brought to the boil. This step must last at least for one hour, and must be vigorous to let the desired chemical reactions occur.

During this period of time the hops and other adjuncts are added. These ingredients provide the different organoleptic characteristics.
There has already been talk about hops and its influence according its addition instant, but also the presence of other adjuncts, such as candy sugar (present in many Belgian beers), aromatic seeds, herbs, and almost as a rule, Irish Moss\textsuperscript{[21]}, an algae that helps to remove proteins in suspension from the wort and clarifies the final product.

**Chilling**

Once the boiling is finished, since the yeast can only ferment at low temperatures, the hot wort must be cooled as quickly as possible. This is fundamental because if the wort rests long between 30 and 50 °C, there will be many possibilities of irruption by uninvited organisms, such as bacteria or wild yeasts. Oxidation may also occur. It is not interesting to add oxygen until the wort is below 24 °C and the yeast has been added.

For a proper cooling, heat exchangers are often used, which reduce the temperature quickly by avoiding the exposure to air.

As soon as the wort is around 20 °C, it can be transferred to the fermenter to go on with the process.

**Fermenting**\textsuperscript{[22]}

To convert wort into beer, the contained wort sugars must be fermented by yeast enzymes to produce ethanol and carbon dioxide in a ten-step process called glycolysis. Fermentation by-products are formed in this process too, having a substantial influence on the taste, aromas and other remarkable beer characteristics.

Alcoholic or ethanol fermentation is expressed according to the following Gay-Lussac formula:

\[
C_6H_{12}O_6 \rightarrow 2 C_2H_5OH + 2 CO_2
\]
If the formed products are calculated quantitatively according to their atomic mass, the following relationships result:

\[ C_6H_{12}O_6 \rightarrow 2 C_2H_5OH + 2 CO_2 \]

<table>
<thead>
<tr>
<th></th>
<th>C:</th>
<th>H:</th>
<th>O:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>72</td>
<td>48</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>12</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>96</td>
<td>32</td>
<td>64</td>
</tr>
</tbody>
</table>

Table 2. Fermentation chemical equation balanced.

From 1 mole of glucose (180 g), 92 g of ethanol and 88 g of CO\(_2\) are formed during the alcoholic fermentation, meaning that sugar is separated in almost equal parts by mass of alcohol and CO\(_2\).

One point that should be emphasized is the major differences between industrial beers and craft beer. On these last ones, fermentation occurs throughout the beer's life cycle, since the yeast is added and until the beer is drunk. Although it is true that the activity diminishes over time. This makes to craft beer be named as a "living product". In industrial beer, a process of filtration and pasteurization will end up applying, which reduces the presence of agents present in the liquid, including yeast. For this reason, a can of an industrial beer will taste the same now that in four months, something that does not happen with the artisanal ones.

**Conditioning/Bottling**

Once the first stage of fermentation is finished, which usually lasts between 10 and 15 days and the most amount of alcohol is produced by letting the CO\(_2\) escape, it is time to transfer it to bottles (or barrels) to let it rest and be able to drink it as we know it.

What is typically done is to add sugars in a clean recipient and stir the blend to mix the parts and oxygenate the beer (this step is called priming), once finished, the beer can be bottled. Now, the yeast that is still alive has more nutrients to react with, generating a small part of alcohol and the CO\(_2\) that will make the beer has gas and bubbles. Dextrose (corn sugar) is commonly used since it is stable, neutral, and has no side effects.
5. Proposed solution

Once the brewing process has been understood, which it is necessary to recognize each step that the program will carry out, the explanation about what elements were considered necessary while the project was under development will be examined. It will be disclosed the selection of hardware and software, also the discussion of possible alternatives and an explanation of what the program by itself is, how it is composed and how it works.

5.1. Design criteria

As we have been taught during all these years of study, all projects must follow a method in order to be able to develop it and accomplish a possible solution from the given circumstances and wills. It is well known that there is no a standard or right way to build up a project or a unique procedure to solve a problem.

All along this project, a series of methodologies have been followed to start thinking about a solution. Possibly, the most important was to define what stages of the brewing process were wanted to be supervised and the elements that would make up the solution.

As mentioned before, CBBS will be put in charge from the beginning, starting by heating water for the following steps, and ending with the wort transfer to the fermenter once it has been cooled. It has been decided to limit up to this point and not to continue with the fermentation process because the large number of variables that would have to be taken into account for it, as well as preparing a refrigerator to keep the temperature controlled and the time that this entails. Even so, in future versions, it is not excluded to add this functionality.

Once all the parts of the process that will control the software have been defined, it is time to proceed with choosing which kind of equipment and installation will be built, the appliances used and how these elements will be arranged. All this will be pointed out in the next section, as well as the possible existing alternatives.

5.2. Physical design

Of course, one of the variables that most influences on which kind of system will be installed is the available space. It is not the same to brew beer in a kitchen, in a garage, or in a 300 m² room. Among the diverse craft beer rigs made, the Stage-Tiers equips are the most commonly employed. These equipments are distinguished by their containers arrangement, placed in one, two or three heights. To design the equipment that CBBS will control, it must be
considered that will be placed in a small space. Therefore, it has been decided to make a Single-Tier rig, where the three kegs are at the same height. This facilitates among other things, the addition of ingredients in each of them, the subsequent cleaning, visibility and a lower expense on construction materials, as well as the homebrewer safety, since there are no heavy elements above the head. On the contrary, it is a somewhat more expensive design in terms of control elements, since all the fluid movements are made by pumps and not by gravity, in contrast to what happens in at least one of the processes of other Stage-Tiers systems.

![Fig. 7. Homebrewing stands configuration. Single, Two and Three Tier respectively. Source: http://www.homebrewtalk.com/](image)

Once the barrels arrangement has been chosen, it is important to decide what type of recirculation method will be installed on the system. There are two options: RIMS and HERMS.

**RIMS (Recirculating Infusion Mash System)**

This system uses direct heat on the tube to heat the wort as it is recirculated. The heat source may be electric or gas, but the wort is heated as it passes through the tube and is pumped during recirculation. The pump keeps the wort moving through the tube at a steady rate to avoid scorching it. The pump must run continuously during the mash when heating, though the heater itself is often cycled on and off to control temperature. A risk with the RIMS system is scorching the wort if the pump fails for some reason.
**HERMS** (Heat Exchanged Recirculating System)

With this other system the wort is passed through a heat exchanger. The most common type of heat exchanger is an immersion setup where a coil of copper tubing is immersed in a hot liquor tun. In this type of setup the hot liquor tun is often kept at a constant temperature slightly above the target mash temperature and the pump is cycled on and off to maintain the temperature of the mash.

![HERMS System](https://www.vandelogt.nl)

The HERMS system has been demonstrated to be faster and more efficient, and therefore it is the used until the date on the current equipment.

Another made decision is to use beer barrels instead of regular pots. The average price for a 50 litre pot is around 70 €, and around 200 € if you want it with valve, thermometer and volume indicator. However, setting up your own mentioned elements into a barrel (which cost is around 30 €), the expense can be reduced in more than 100 €.

To determine the other apparatus to use, an electric and hydraulic scheme has been done, as shown in Fig 10:
Looking at this first scheme, it can be seen that along with the three barrels, five two-way valves are needed, one for each barrel and two for both pump in order to regulate the liquid flow.

Simultaneously, to assemble correctly the connections and minimize the number of tubes, four three-way valves have been added allowing fluid direction changes.

---

*Fig. 10. Hydraulic scheme*

*Scheme made with ProfiCAD software.*

*Source: own.*

---

*Fig. 11. 2-way and 3-way valves.*

*Source: [http://uk.rs-online.com](http://uk.rs-online.com)
As Figure 10 shows, a coil heat exchanger is required to carry out the HERMS system. Additionally, a plate exchanger for a faster wort cooling.

It should be noticed, and as one might expect, that each element will dispose of threaded connections or adjusting hook elements for the other parts. In this case, the ensemble will be equipped with Cam Locks. Those allow the pipes to be connected and disconnected without needing nuts, bolts or screws, since Cam Locks are safe, easy to assemble and disassemble, and ensure the correct liquid transference.
With the scheme in Fig. 15 it can be seen all those components that need electricity to work. The first thing seen on the left is the control device, a Raspberry Pi, which will be discussed later.

There are two brushless DC pumps, one to boost and move the water and another dedicated to the wort. The two pumps will only work together during the cooling step, where both liquids will be recirculating. As punctuation, remark that both pumps are not connected directly to the Raspberry, since they need 12 V to operate and the board is only able to provide 5 V. For this reason, they will be connected to relays connected to the Raspberry and to the electric network at the same time. In the same way connected, there will be a low-density electrical resistance inside the HLT, being able to raise the water temperature on this recipient in a more precise way than with the butane burner.
In the scheme, three temperature sensors are showed up, those will measure this value inside the HLT, at the MLT exit and the wort temperature at the plate exchanger exit. Furthermore, relays and sensors connections and the Raspberry Pi characteristics, will be explained in more detail in the following section.

5.2.1. ELECTRONICS

On this section it will described the hardware for monitoring the process, how it was chosen, and how it is connected. A Raspberry Pi is, with the permission of beer, the main component of the whole project, the "brain" of the operation.

Raspberry Pi is a small single-board computer developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. Despite this, due to its versatility, Raspberry has been used around the world by millions of people to carry out their projects and ideas.

In the present case, the latest released model has been used, the Raspberry Pi 3 Model B, more powerful than its predecessors and perfectly equipped for its purpose.

![Raspberry Pi V3](https://www.raspberrypi.org/)

Fig. 18. Raspberry Pi V3.

Of course, it was not chosen from the beginning. Before thinking about what kind of application it would be done, Arduino was a good option. This last one board belongs to an open-source electronic prototyping platform allowing users to create interactive electronic objects. This single-board, even being able to control some process steps such temperatures, is far more limited for other options, such as reading recipes, plotting, being able to design an interface, etc ...
Once determined the "computer" that was going to control everything, it was necessary to decide on the devices that would control the temperature and how the pumps would be connected.

In reference to the temperature sensors, two possible solutions were found. Probes based on PT100 sensors\(^2\) (RTD sensors, Resistance Temperature Detectors) or DS18B20 sensors.\(^3\)

These latter were chosen for several reasons, such as its easier usability through Raspberry, as they use 1-Wire technology for sending data (can send information from several sensors through the same cable or bus), for its resolution and reading time (despite being not so fast than the alternative, was sufficient for the requirements), and its price, around 5 times cheaper. Additionally, both offer the possibility to go encapsulated with a stainless steel tube suitable for feeding.

These sensors have 3 pins, power (VDO), Ground (GND), and Data (DQ). The power supply is 3.3 V (Raspberry Pin 1) and has a 4.7kOhm pull-up resistor\(^3\) between VDO and data bus (connected to Pin 7 or GPIO 4) as shown in Figure 21 on the next page.
Once connected, it must be understood the information sent to Raspberry from these sensors. Before begin, note that these sensors come with a 12 bit resolution by default (configurable between 9 and 12), so the maximum conversion time for this resolution is 750 ms. Although it is true that the reading could be faster if the resolution is lowered, the system will not ask for such speed to configure it as well.

To be able to read the temperatures, first of all the Raspberry must be configured to activate the One-Wire interface. Once done, it should be found which sensor is which from the recognized in "/sys/bus/w1/devices" folder, on the Raspberry root files. There are several folders named such "28-xxxxxxxxxxxxx" where the "x" refers to the "name" (each sensor has a 64 bit unique address) and within each folder among other files, a folder called w1_slave is found. These folders are read using "cat w1_slave" command, resulting in an output similar to the following one:

```
73 01 4b 46 7f ff 0d 10 41 : crc=41 YES
73 01 4b 46 7f ff 0d 10 41 t=23187
```

Two text lines are displayed\(^2\), although the last numbers are what really matters, in this example, t=23187. This value is the temperature with a resolution of three decimals but without decimal point. Therefore, the measured temperature is 23.187 °C. To obtain this temperature during the process, it will be as easy as reading this folder each time a particular temperature is requested and pull these last five characters out.

---
\(^2\) When data is communicated between two devices, it is common to use some type of error, in this case, cyclic redundancy check (CRC) is used. The general idea is that the transmitter calculates and transmits a value and the receiver performs the same calculations and compares the result with the check value.
Once having learned to use the temperature sensors connected to the Raspberry, it is necessary to do the same with the relays to control both pumps and the heating resistor. To do this it is needed a 1 kOhm resistor\[27\], a diode (1N4007 in this case)\[28\], an NPN transistor (PN2222 model)\[29\] and a solid state relay\[30\] for each circuit. So, three circuits like the figure below will be required:

![Raspberry Pi GPIO 3v3](https://raspberrypi.stackexchange.com)

Essentially, the transistor is used to energise the relay coil with the required voltage and current. To activate the relay, what the circuit does is send a few milliamps at 3.3V from the GPIO pin through a 1K resistor. This current is enough to saturate the transistor, causing current to flow on the 5V rail through the transistor, and therefore also through the relay coil. The diode in the circuit is there to conduct the current generated by the de-energising coil back across the coil, allowing the power to dissipate more gradually, avoiding a voltage spike. Once the three circuits are connected (in this project, to the physical pins 15, 16 and 18), it is time to learn how they are controlled.

The first thing to do is to configure these three pins as output pins. That can easily done on the script, not before importing the RPi.GPIO library, which allows a simpler manipulation to activate and deactivate these pins and thus control the pumps and the thermal resistance. To set up the prototype, once the connections have been checked and secured, a Raspberry Pi compatible shield has been used to provide the base to mount and weld the circuit in a compact way, being easier to move and to connect.
5.3. Overview. Final equipment

At this section, the final design of the equipment to be assembled will be explained. The most important elements and a possible layout of them will be shown, as well as those not mentioned but equally important.

As already seen, the setup will consist of 3 barrels in Single-Tier configuration, or at a single height. Those barrels are the Hot Liquor Tank (HLT), the Mash Liquor Tank (MLT) and the Boiler Tun, each of them conveniently prepared and with all necessary connections, as well as an analogue thermometer and a level control tube.

The HLT contains, apart from all those mentioned elements, the HERMS coil system and the electric heating resistor. In the MLT case, there is a round shape diffuser for a proper sprinkling of the wort during the mashing step, spreading out all around the barrel. The boiler is a standard barrel with the respective Cam Locks.

All the elements have been designed using Autodesk Inventor 2016 software or downloaded from the GrabCAD library. [31]
Of course, all these barrels must be mounted on some type of structure. To get an idea, a 1.80x0.85 m rig has been designed. It has intermediate spaces to keep the utensils and protect the Raspberry with the screen to follow the process. On the other hand, two butane burners will be installed and a space will be left for the respective gas bottle.
Both pumps and the plate heat exchanger will be suitably coupled to the structure to ensure their fastening and to be easily accessible.
Once the ensemble is finished, it will be enough separation between the barrels to be able to handle them without problems. In addition, the highest point will be at a maximum height of 1.50 m, enough to keep controlled the process and add the ingredients without stairs required, which could be dangerous for the homebrewer.

![Final ensemble](image31.png)

**Fig. 31. Final ensemble.**
*Source: own.*

### 5.4. Software design

Once the physical implementation of the system is done, let's move on to the software section, how it is done, how it has been developed and what tasks it performs.

CBBS is composed of three fundamental blocks, a BeerXML format reader, the CBBS main code and the graphical interface designed with Tkinter. The following parts will explain the particularities of each, their use and how it is integrated into the main program.

First of all, note that Python 3 has been the programming language chosen due to its versatility and because it is installed by default on Raspberry. The operating system installed on the Raspberry Pi, Raspbian, has already installed the essential modules to be able to use it and was the option that seemed to offer more possibilities.
5.4.1. BeerXML Parser

An important point was to be able to read a recipe in one of the most commonly used formats among the homebrewers, BeerSmith files (.bsmx) or BeerXML (.xml). The decision was not difficult actually, while BeerSmith files are made to read them in the eponymous program, the XML (Extensible Markup Language) format instead, is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable in different ways. The BeerSmith author by himself and other colleagues created this standard with the primary purpose to exchange recipes, but it could be used for the exchange of other brewing data too, as styles or ingredients.

Once the reading format was determined, some kind of reader should be created to read this Standard, since it would allow more versatility and advantages over modifying the code for each recipe. Before starting, it was searched if there was some kind of library created for this function, and although in the official documentation does not appear anything, a project was found uploaded to the well-known github repository. This project was defined as a recipe reader in XML format with Python. It was started by Tom Herold with MIT license two years ago, and it seems to have been abandoned, after trying to contact the author and not receive response, it was decided to use the base code and make it work.[33]

To make it operational has been spent more time than desired, but finally, it has been possible to implement a code that is able to read any recipe in .xml format and return the desired values. It should be noted that additional functions have been implemented and more time has been spent for making a universal recipe reader, and not just for the processes involved with CBBS.

Finally, the python script beerxmlparser.py has been created, being able to read and return information related to the following aspects:

- Recipe information
- Style information
- Equipment information
- Mash information
- Mash steps
- Fermentables/grain
- Hops
- Miscellaneous
- Water
- Yeast

The complete list of elements and additional information that can be procured from each of these key aspects can be found in ANNEX 2.
Beerxmlparser.py consists on one file, the main code, and individual scripts for each tag inside the recipe. In total, there are 12 independent scripts plus the main one that makes up the whole library.

The idea, once the project is finished, is to publish this script as an independently library of CBBS and allow its free distribution among the homebrewer community. In the same way, as it has taken leverage of the first version to develop it, anyone will be allowed to modify the code and make it in their own way.

The form of use it is simple, importing firstly beerxmlparser.py at the main code and make a call to it with:

```python
recipe = beerxmlparser.Recipe()
```

Where the Recipe class stores all the information in the "recipe" object, being able to get the values specified in the attachment easily. For example, writing "recipe.Name" let the brewer knows the name of the selected recipe.

A detail to keep in mind is to know that BeerXML Parser is designed to work with complete recipes, in other words, there must be no missing information. Otherwise, the program will not suffer any type of malfunction but it could lead to different troubles when following the recipe if those values are relevant.

On Validation Tests chapter, a recipe will be created to check the proper functioning of this program and it will be seen how the format is organized.

### 5.4.2. SOFTWARE STRUCTURE

Once the reading of recipes is achieved and have means for extracting all the necessary information, it is time to design the program that will control each of the necessary steps to brew, hopefully, a good beer. Before starting to conceive the code, it is necessary to specify exactly how it will be structured, what kind of objects it will use and thus define the main functionalities contained in the application. In it, the user must enter the recipe that wants to produce and the program by itself will be in charge of processing the data and use it during the guidance of the operation.
Fig. 32 shows in a very simplified way everything that the application should do. By reading the recipe data and the information received by the sensors, it must use, interpret and return somehow this data to create the beer, to act on the mechanical elements and to apprise the brewer of any consideration. It was decided to create the program only with two main scripts, the already known beerxmlparser.py and CBBS.py, which will use the first one to manage the recipes. CBBS will have all the necessary information to execute the application and carry out the process.

Libraries

In order to execute the code, some libraries need to be loaded and executed.

- **Tkinter, ttk, tkFont, tkMessageBox, FigureCanvasTkAgg.** Referring to the libraries used by the GUI. They allow to modify the fonts, images, text boxes and other features that do not come by default in Tkinter package.

- **Pylab, matplotlib.** They allow to draw plots from data easily and visually. It is quickly customizable and widely used in Python.

- **Time.** It allows to create operations with time in the necessary steps, for example, a countdown.

- **Os.** Provides a portable way of using operating system dependent functionalities. By this way, to open a file or read a document, in this case, the recipe file.

- **RPi.GPIO.** Indispensable for the Raspberry correct functioning. It allows to initialize and control the GPIO board pins, for example, to switch the relays state.
The code

Even considering that the whole code will be added in ANNEX 1, the most important parts will be explained and how they work. Looking into CBBS, it can be seen that it is composed by a homonymous class with different functions.

First, the __init__ method is started, which is a constructor that allows inheriting the variables declared in this one at any time. It is useful to create the main code variables, the program configuration, or where the GUI base will be defined. Among many other things, appear the variables used as flags, these are used to control time, relays or temperature sensors themselves, and this facilitates the control of these elements and let to know their status quickly.

In this same method, the variables to control the separate steps are found, always helping to know which step is under way and what the next ones are.

The next function in the code is Selectrecipe(self), executed at the time the button is pressed to choose the recipe. Then, the code carries out both processes of deleting the previous loaded recipe and reading the new one selected.

The next part of code is the one which refers to all the steps in the process. It is structured in such a way that each step has a function associated with it. In each function the main message and the relevant flags are updated, also, the parameters are modified if necessary, and the interface is refreshed.

There are a total of eleven steps plus the Reset function, which is executed when the process is finished.

Note that the user can move forward easily to the next step thanks to a button clicking on it, useful in case there is a problem or simply if it is desired by the homebrewer.

Fig. 33. Steps section
The next code division is built by four common blocks; Plot, Timer, Helper and Control functions.

The first block, called Plot Functions, contains the necessary functions to restart, execute and draw the graphs respectively.

As it can be seen in Figure 34, each temperature sensor has associated a group of three functions. This is due to it is much faster reading the sensors and the subsequent plotting during the process if it is done individually.

The second, Timer Functions, defines the required functions to execute and control the time, as well as those that are related to the interface.

The gettime function converts a given time in minutes (all recipe times are given in this magnitude) and convert them to hours and minutes for a correct display in the screen.

start_btn and pause_btn, are in charge of controlling the time, basically resume it or pause it when the corresponding button is pressed.

The subsequent two functions, moremin and lessmin, let the brewer to increase or decrease the time manually by pressing on the linked button. This is useful if any step is done incorrectly or by any external decision to the recipe.

Finally, the tick function is the responsible to run the time as well as advance toward the next step when the time finishes if it is the case, also printing on the screen a message if necessary.
On the Helper Functions block, the first function found is *sortingingredients*, which as its name indicates, reads the recipe and collects all data referring to the ingredients to display these parameters on screen, sorted and divided by groups. This allows its subsequent use during the process. Something similar does the *styleprops* function, which reads, interprets and shows all the information belonging to the beer style.

The next two, *read_temp* and *write_temp*, refer to the temperature sensors. *Read_temp* is responsible for reading the data (saved in the *wl_slave* folder as seen above) and *write_temp* extracts the temperature value on the text and converts it into a float number in order to use it.

*Updatesteps* function is executed each time a new step occurs during the recipe and updates the variables accordingly. It works keeping at all times under control the actual step and the remaining ones to end the process.

*Center* and *displaybuttons* play an important role in the interface, the first let align the window according to the screen resolution, although it must be configured for each computer conveniently. *Displaybuttons*, on the other hand, shows the on-screen buttons and makes them functional.

Finally, the *about* function displays the application software information on the main screen when the button is pressed.

The last block, Control Functions, contains the functions designated to control the relays, activating or deactivating these elements when the function is called changing their status or flag as convenient.
Having established the sections that form the main class, there are some lines in the end of the script to execute this code.

```
# Voltage: 5V (PIN1)
relay_list = [15, 16, 18] # [Yellow, Green, Orange]
GPIO.setmode(GPIO.BORD)
GPIO.setwarnings(False)
GPIO.setup(relay_list, GPIO.OUT)

# Voltage: 3.3V (PIN1)
# PIN7
os.system('modprobe w1-gpio')

Fig. 38. Setup code
```

As Figure 38 shows, the first part configures the Raspberry to use the GPIO pins. These braces connected to the relays are defined as output pins, in this case 15, 16 and 18, in order to be disposed to switch on and off when required, and simultaneously, the code runs `os.system('modprobe w1-gpio')` and `os.system('modprobe w1-therm')` to read and collect the data from temperature sensors.

To execute the part that builds the program interface, a basic Tkinter widget named `root` is created with `tk.Tk()`, which is included in the CBBS call.

### 5.4.3. GUI

Once the base code has been developed to carry out the process, an interface is needed to guide the homebrewer and allow the interaction between this user and the application. A GUI (Graphical User Interface) has been implemented on a screen using the aforementioned Tkinter library, a Python binding to the Tk GUI Toolkit developed for to create different widgets and objects. There are other alternatives, but Tkinter comes predefined with Python installation and it is the most commonly used, so it is also the best documented.

In this section it will be seen which kind of elements and objects the program uses to display the interface, and then, how it is applied in the final implementation.
- **Definitions**

**Window**: Rectangular area somewhere on the display screen that allows to add other widgets or objects. It is usually called “root” and it is the main widget.

**Widget**: The generic term for any of the building blocks that make up an application in a graphical user interface. For example, buttons, text fields, frames or text labels. Widgets and objects in general can be active or passive, depending on the kind of interaction with the user. Active objects are clickable or modifiable, passive instead, just show information.

**Frame**: Basic unit of organization for complex layouts. A frame is a rectangular area inside a window that can contain other widgets.

**Child/parent**: When any widget is created, a *parent-child* relationship is created. For example, if you place a text label inside a frame, the frame is the parent of the label.

**Layout**: Arrangement of different widgets or objects inside windows or others widgets. These widgets can be arranged or placed around the available areas by rows and columns. Also, to place any widget on the application screen you must “register” this widget with the `.grid()` method. If it is not done, the widget will exist internally but it will not be visible on the screen. The internal layout management is done with this option.

- **Main widgets**

As it shows in Figure 39, the principal widgets are visible, Window and Frame, additionally, the Label widget lets to add any text inside other objects. Different arrangements and child/parent bonds are done as it can be seen too.

![Figure 39. Tkinter. Main widgets. Source: own.](image-url)
Different Frame styles can be displayed as shown in Figure 40. This is useful to differentiate between parts on the same window or object.

![Frame styles example](image)

*Fig. 40. Tkinter. Frame styles. Source: own.*

- **The Button widget**

  Button is an active widget that lets the user interact with it, calling a function (callback) when it is pressed and showing a message (or not) if it is needed.

  This object will be useful to communicate with the program and to control the process.

![Button example](image)

*Fig. 41. Tkinter. Button widget. Source: own.*
As it is said, when a button is pressed a function is called. This usually shows a message (with the command `tkMessageBox.showwarning` ("text to display") and proceeds with another action written.

![Image of a button message](image)

*Fig. 42. Tkinter. Button message.*
*Source: own.*

- **The Listbox widget**

The purpose of the Listbox widget is to display a set of lines of text. Generally they are intended to allow the user to select one or more items from a list, even though, this functionality will not be used in CBBS, since the listbox will provide the ingredients list and the steps information.

![Image of a listbox widget](image)

*Fig. 43. Tkinter. Listbox widget.*
*Source: own.*
- **The Canvas widget**

A canvas is a rectangular area intended for drawing pictures or other complex layouts. Graphics, text, widgets, or frames can be placed on it. This is useful to improve the user experience and enhance the GUI.

![Fig. 44. Tkinter. Canvas widget. Source: own.](image)

- **The Notebook widget**

The last widget shown is the Notebook. It has the purpose to create different but related tabs in a window. As it is seen in Figure 45, every tab acts as an individual main frame, being able to place other widgets on it.

This widget will be used in the application to divide the recipe information and the process guidance.

![Fig. 45. Tkinter. Notebook widget. Source: own.](image)
6. CBBS. Final implementation

Now, thereafter the main code functions and how *Tkinter* works have been seen, the developed application will be finally unveiled to go into detail about how it works and what the contained elements are. Subsequently, a simulation will be done with a new recipe created for the occasion.

![Fig. 46. CBBS. Main screens. Source: own.](image-url)
REFERENCE KEYPOINTS

1. RECIPE INFO
   Notebook widget. Active object.
   App's first tab, shows the recipe information.

2. RECIPE PROCESS
   Notebook widget. Active object.
   App's second tab, shows the process information.

3. GENERAL INFORMATION
   Label. Passive object.
   Basic information about the beer.

4. STYLE AND RECIPE CHARACTERISTICS
   Label. Passive object.
   Displays the main characteristics and set these values in a range.

5. INGREDIENTS
   Listbox widget. Passive object.
   Shows the ingredients, sorted by type and time.

6. SELECT RECIPE
   Button. Active object.
   Opens a window that allows to select the recipe.

7. ABOUT
   Button. Active object.
   Display a screen that shows the app information.

8. EXIT
   Button. Active object.
   Stops and ends the app.

9. RECIPE PROGRESS THUMBNAIL
   Canvas widget. Passive object.
   Indicates the actual main step.

10. ACTUAL STEP
    Listbox widget. Passive object.
    Display the step information and other things to keep in mind.

11. NEXT STEPS
    Listbox widget. Passive object.
    Shows the next step messages. Useful to know what the next actions are.

12. MANUAL CONTROL
    Buttons. Active/passive object.
    Activate and deactivate the different elements shown in any moment.

13. TIMER
    Buttons and label widgets. Active/passive object.
    Shows the time left on the required steps and let the user start and stop the countdown and modify the time.

14. PLOTS
    Matplotlib object. Passive object.
    Draws the different temperatures and the target to reach.

15. ACTION BUTTON
    Button. Active object.
    Communicates with the app, ends steps or accepts messages…

16. QUICK NEXT STEP
    Button. Active object.
    Jumps instantly to the next step.
3 General Information

**General Information**

<table>
<thead>
<tr>
<th>Name</th>
<th>Batch size: 19 l</th>
</tr>
</thead>
<tbody>
<tr>
<td>Style</td>
<td>Robust Porter</td>
</tr>
<tr>
<td>Type</td>
<td>All Grain</td>
</tr>
<tr>
<td>Brewer</td>
<td>David Catalán</td>
</tr>
</tbody>
</table>

Fig. 47. CBBS. General information. Source: own.

- **Name**: Recipe name.
- **Style**: Beer style, according to the BJCP guide.
- **Type**: Type of recipe, it can be "All grain", "Partial grain" or "Extract".
- **Brewer**: Brewer’s name.

**Style and recipe characteristics**

- **Estimated OG**: Original Gravity [Min: 1.028 – Max: 1.130]
- **Estimated FG**: Final Gravity [Min: 0.998 – Max: 1.040]
- **Estimated ABV**: Alcohol by volume [Min: 0 – Max: >13]
- **Estimated IBU**: IBUs [Min: 0 – Max: >100]
- **Estimated Color**: EBC color [Min: 0– Max: >79]

Fig. 48. CBBS. Style and recipe characteristics. Source: own.

The estimated colour bar, modifies its colour depending on the ingredients. As it has seen, the different levels of kilning provide different colours to the grain. The recipe has associated a value and it changes the colour seen.

Estimation: 18.5 EBC

Fig. 49. CBBS. Color by recipe. Source: own.
Ingredients

In Figure 50 the list of ingredients can be seen. As it is said, sorted by type. Also, the hops and miscellaneous are sorted by addition order. All the needed quantities are also shown.

Select recipe

When the Select recipe button is clicked, a window is opened, allowing to choose the recipe. Before the process is started, the recipe can be changed freely, once it is started, the button is deactivated.
7 About

This buttons opens this window, which shows the information about the program.

9 Recipe progress thumbnail

These rectangles are highlighted depending on which of the four main steps is currently running, heating the water, mashing, boiling the wort or chilling it.

12 Manual Control

This panel allows to turn on/off the elements connected to the relays. Even any recipe is selected or the process is paused, the buttons are clickable. This provides a useful support in order to clean the kegs, pipes and any other physical element.
Timer

If the step requires it, a countdown is showed to know the time left to end the current step. This timer can be stopped/resumed if desired. Additionally, there are two buttons implemented to add or decrease minutes, this means that in any moment, the brewer can reach any point of the process.

Plots

There are three different plots that are displayed depending on the active step. The refresh rate is about one second, more than enough to have controlled the process. Also, if the step requires it, some guidance lines are shown.

Figure 56 exposes this example during the mashing step. The temperature must be around 66 ºC, and never outside the range of +/- 2 ºC. If the temperature falls under this value, the heater is activated automatically to heat the water that provides the necessary heat.
7. Validation tests

In order to show the entire process, a new recipe has been created with Beersmith software. Once the recipe was done, it was exported in .xml format to allow CBBS to read it. Honey Citra Pale Ale is an American Pale Ale with Pale Malt as base malt and others to complement the recipe. Additionally, honey will be added during the mash to provide some sweet flavouring and more fermentable sugars. It has three different hop varieties, Citra, added before during the first minutes (it will produce the beer bitterness as seen), Cascade while the wort boils and Centennial to give some floral and citrus aromas. A half tablet of Irish Moss is added as well when there are ten minutes left. Straightaway, it is going to proceed to follow all the steps viewing the parts mentioned on the last section. The brewer should get 24 litres of effective beer once the process is completed.

---

3 The estimated IBUS calculated on Beersmith and CBBS are slightly different. This is because Beersmith uses Rager formula to calculate this value, and CBBS uses the Tinseth Method.\[34\]
7.1. Following the process

Once opened the program and loaded the recipe created, it shows the main screen (See Screenshot 1) with all the information needed.

Going to the second tab (See Screenshot 2) it allows to start the process by pressing the "Go!" button.
The process starts ordering the brewer to heat 40 litres of water to 75.6 °C. Once reached, a message appears requesting to check the connections. (The user must change the valves direction properly) (See screenshots 3 and 4)

When the confirmation button is pressed, the pump nº1 starts running and the brewer must press “OK” when the Mash Liquor Tank has 17.3 litres of water. Immediately, pump nº1 stops and appears a warning reminding to close the HLT valve in order to start the mashing step. (See Screenshots 5
Now, it is time to add the fermentables and remove to ensure the proper grain bath. When all the different grains and the honey are added, the brewer must press **OK** to proceed with the next step. (See Screenshot 7)

Before start running the pump nº2, the app shows another reminder to check the connections and fill the HLT if needed, at least to cover the coil (See Screenshot 8)

Just when it is confirmed, the pump turns on and recirculates the wort, always ensuring appropriate temperature, 66 in this case, switching the heating element to keep the HLT water warm. (See Screenshot 9)
At 15 minutes left, a warning is showed to close the MLT valve and stop recirculating to let rest the grain. (See screenshots 10 and 11)

When the mashing is completed, it is the moment to start the sparging. Opening valves nº1 and nº2 and making the appropriate connections as alerted, both pumps turns on and the brewer must control the water flux as much from the HLT as from the MLT, since the water volume should remain stable inside the Mash Lauter Tun. (See Screenshot 12). When all the wort is extracted, pressing OK stops the pumps and initiates the next step. (See
To start the boiling, the user must light the burner to heat the wort, and by pressing “Go!” when it bubbles intensely, the time starts running. (See Screenshot 15)

From now on, the additions must be added when needed. In this recipe, the first addition is just in the beginning, so the user should have ready the hops. (See screenshots 1 and 16). For the next 60 minutes, the wort will be boiling.
As Screenshot 18 shows, the next addition is at 30 minutes left. After that, at 10 minutes left, the Irish Moss tablet and the last hops are added. (See Screenshots 19 and 20).

When the boiling time is ended, the wort must be cooled. First of all, some ice must be added into the HLT to cool some water. Additionally, and when the connections are checked (current water is used firstly), the pumps starts running to move the liquids. (See screenshot 21)
The wort must be cooled the fastest possible to avoid infections or contamination. A plot is shown drawing the temperature from the 3rd probe, in the plate heat exchanger wort exit. It must reach 45 ºC before using the cold water, and when it does, it is demanded to close the water circuit and connect it to the HLT. Once confirmed, the cooling is resumed to reach the final temperature. (See the above screenshots)
The wort keeps chilling, and when it reaches 20 ºC (the wort should be at this temperature to ensure the proper yeast growth), a warning comes out to remind to close the valves and also stops pump nº2. *(See Screenshot 26, 27, 28).*

To finish the process, the cold wort must be transferred to the fermenter. Following the exposed message in *Screenshot 29*, the brewer will connect the Boiler to the fermenter and pump nº2 will continue moving the wort.
Once all the wort is transferred, the brewer must press the “OK” button to turn off both pumps. (It is interesting to keep cooling the wort even when it has reached 20 °C) (See Screenshot 30)
As soon as the button is pressed and the pumps are disabled, a warning will be shown announcing the end of the process. (See Screenshot 31).
8. Project scheduling and budget

An approximate schedule and budget plan of the project is going to be carried out in this section.

The amount of credits that ETSEIB normative stipulates is 12 ECTs, and since each credit has assigned 25 hours of work, the project can be considered to need 300 hours in order to be accomplished.\(^{[35]}\) Despite that, the number of real hours dedicated has been higher, thanks to a proper counting, the amount of hours worked equals to 390 approximately.

![Fig. 58. Gantt chart. TFG timeline dedication. Source: own.](image)

As it can be seen in the figure above, the main activities are focused on coding and designing, and the principal research has been done during the first hours.

In order to estimate the costs of the project, the total will be calculated from the different items included, intellectual and material cost.

The first item includes the price of the people involved around the design and the execution of the idea.

<table>
<thead>
<tr>
<th>ITEM</th>
<th>WORKING HOURS (H)</th>
<th>COST PER HOUR (€/H)</th>
<th>COST (€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENGINEER</td>
<td>390</td>
<td>40</td>
<td>15600</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td>15600</td>
</tr>
</tbody>
</table>

*Table 3. Intellectual cost. Source: own.*
The material cost is composed of the different components used to achieve the actual project. As it has been said, CBBS and everything that goes around it is still evolving and growing, but the additional parts will not be reflected on this budget.

<table>
<thead>
<tr>
<th>ITEM</th>
<th>AMOUNT (#)</th>
<th>COST PER ARTICLE (€/#)</th>
<th>COST (€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RASPBERRY PI 3</td>
<td>1</td>
<td>39.90</td>
<td>39.90</td>
</tr>
<tr>
<td>SD CARD 32 GB</td>
<td>1</td>
<td>12.99</td>
<td>12.99</td>
</tr>
<tr>
<td>RPI SCREWS PROTOTYPE SHIELD</td>
<td>1</td>
<td>7.94</td>
<td>7.95</td>
</tr>
<tr>
<td>SSR-25 DA RELAY</td>
<td>3</td>
<td>4.95</td>
<td>14.85</td>
</tr>
<tr>
<td>DS18B20 DIGITAL SENSOR</td>
<td>3</td>
<td>2.30</td>
<td>6.90</td>
</tr>
<tr>
<td>WATER PUMP</td>
<td>2</td>
<td>16.45</td>
<td>32.90</td>
</tr>
<tr>
<td>STAINLESS STEEL HEATING ELEMENT</td>
<td>1</td>
<td>35.90</td>
<td>35.90</td>
</tr>
<tr>
<td>ELECTRONIC COMPONENTS⁴</td>
<td>1</td>
<td>2.00</td>
<td>2.00</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td><strong>153.39</strong></td>
</tr>
</tbody>
</table>

Table 4. Material cost.  
Source: own.

Also, it would be interesting calculating how much energy consumption it will be used for every batch. We will consider 4 hours of total work for the Raspberry Pi and the screen, 2.5 hours for the pumps and 1 hour the heating resistor. Finally, the cost will be calculated considering 0.128€/kWh according to the actual PVPC rate given by the Government of Spain.⁴

<table>
<thead>
<tr>
<th>ITEM</th>
<th>POWER USED [W]</th>
<th>ENERGY CONSUMED [kWh]</th>
<th>COST (€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RASPBERRY PI 3</td>
<td>2.4</td>
<td>0.0096</td>
<td>≈ 0</td>
</tr>
<tr>
<td>12&quot; SCREEN</td>
<td>20</td>
<td>0.08</td>
<td>0.01</td>
</tr>
<tr>
<td>WATER PUMPS</td>
<td>38.4</td>
<td>0.096</td>
<td>0.01</td>
</tr>
<tr>
<td>STAINLESS STEEL HEATING ELEMENT</td>
<td>3000</td>
<td>3</td>
<td>0.384</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td><strong>3.1856</strong></td>
<td><strong>0.40</strong></td>
</tr>
</tbody>
</table>

Table 5. Energetic cost.  
Source: own.

⁴ Composed by resistors, transistors, wires, tin for welding... Everything needed but not bought exclusively for the project.
As it can be seen, the electricity cost can be neglected for the final cost. Before calculating the sum, it is important to say that the software cost is another item that it should be shown. Nevertheless, it can be omitted since the cost is considered zero. SO Raspbian can be downloaded free of charge on their webpage. Autodesk Inventor have available a Student License with the essential features to develop the idea. The only software that it could be included is BeerSmith®, but since the program had been bought beforehand, it will not compute on the final price.

The final amount of this Bachelor’s Thesis is:

<table>
<thead>
<tr>
<th>ITEM</th>
<th>COST (€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTELLECTUAL COST</td>
<td>15600</td>
</tr>
<tr>
<td>MATERIAL COST</td>
<td>153.39</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>15753.39</strong></td>
</tr>
</tbody>
</table>

Table 6. Total price.
Source: own.

---

5 The actual cost for a BeerSmith license is between 27.95$ and 89.95$ depending on the version. [http://beersmith.com/order/](http://beersmith.com/order/)

6 Other costs such as internet, electricity, the computer or tools used have been neglected.
9. Social and environmental impact

Nowadays, being aware of the environmental repercussion associated with any activity is a must, in particular for the engineering field.

In our case, the main influential factors over the environment can be divided into three groups. The equipment components, the raw material and the emissions.

In the first aspect, the production cost of the different elements that constitute the project is an important point to consider. It is necessary to know that although the final construction consists of several metal containers, metal profiles, plumbing and other elements, this project only focuses on the system design and the creation of the software to control it, so the components used that must be manufactured is reduced to the electronics used. Needless to say that for the final equipment construction, the idea is to use recyclable and light materials, such as aluminium or stainless steel since they are reusable and have a longer life expectancy than plastic for example.

Thereupon the impact of the raw material used must be considered. Although as in the previous paragraph is said, is not something that directly influences this work, but it is good to look forward how the waste generated in the brewing process will affect. Once finished the batch, two leftovers must be retired to cleanse all the other elements, the hops added during the boil and the grain used in maceration. It should be noted that both ingredients are completely organic and biodegradable, so they are a good option to use as fertilizer. In the case of grain, it is quite common to use it as raw material for cooking as own consumption or feeding domestic animals, as well as cookies or porridge made from the same grain.

Finally, it would be necessary to consider the emissions generated during the process. To get an idea whether the process is actually polluting, the equivalent mass if kg of CO\(_2\) emitted by a production cycle will be estimated. It will be taken into account an approximation of energy used, butane emissions during the combustion and, as addition, the effect of fermentation for the total emissions.

Using a spreadsheet generated jointly by the European Regional Development Fund and Cámara de Zaragoza, the CO\(_2\) equivalent emissions has been calculated related to electricity, butane gas and water.\[37\]

In the previous section it has already measured the electric consumption to produce a batch, being 3.1856 kWh. Knowing that a butane bottle has the approximate gas charge for 5
batches, also that are used about 80 water litres between the production and cooling the wort, it results in:

\[
\text{Cálculo automático de emisiones totales en relación a los consumos energéticos de sus instalaciones}
\]

<table>
<thead>
<tr>
<th>Consumo</th>
<th>Unidades de medida física</th>
<th>Factor de emisión (Kg de CO2 eq/kWh)</th>
<th>Kg de CO2 eq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electricidad</td>
<td>kWh</td>
<td>0,385</td>
<td>1,22584</td>
</tr>
</tbody>
</table>

**FACTORES DE EMISIÓN COMBUSTIBLES**

<table>
<thead>
<tr>
<th>Consumo</th>
<th>Unidades de medida física</th>
<th>Factor de emisión (Kg de CO2 eq/kWh)</th>
<th>Kg de CO2 eq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gas butano</td>
<td>Nº de bombonas o Kg</td>
<td>12,4389</td>
<td></td>
</tr>
<tr>
<td>Consumo</td>
<td>kWh</td>
<td>0,2383</td>
<td>0,592837974</td>
</tr>
</tbody>
</table>

**FACTORES DE EMISIÓN DE OTROS PRODUCTOS**

<table>
<thead>
<tr>
<th>Producto</th>
<th>Consumo</th>
<th>Unidades de medida física</th>
<th>Factor de emisión (Kg de CO2 eq/ul)</th>
<th>Kg de CO2 eq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agua</td>
<td>0,01</td>
<td>m3</td>
<td>0,788</td>
<td>0,06304</td>
</tr>
</tbody>
</table>

Fig. 59. CO2 emissions. Source: own.

In addition, we will go a little further into the influence of the fermentation on CO2 emissions. Investigations published on *Life, Fermented*, showed that the total amount of CO2 produced by 18.9L is between 0.4 and 1.8 kg CO2, depending on the specific gravity (Original Gravity) and on the attenuation percentage\(^7\). Considering a 40L batch, we will define the emitted quantity as the maximum possible, around 4 kg CO2.\(^{[38]}\)

\[
\text{Attenuation } [\%] = \frac{OG - FG}{OG}
\]

---

\(^7\) The attenuation is the percentage used to describe the percent of malt sugar that is converted by the yeast strain to ethanol and CO2 during the fermentation. It is the result determined by comparing beer Original and Final gravities.
To sum up, the peak of CO₂ emitted is about 5.9 kg for every batch (being the fermentation the main source with two thirds of the whole emissions).

Considering the best-selling car of 2016 in Spain, Seat León⁴⁹, the average emissions discharged among their models, with combined driving (urban and extra-urban cycles) is 130 g/km.⁵⁰ This means that in approximately 45 km this average car would has emitted the same quantity than all the process of making the beer. Another way to look on this aspect, is with the volume of gasoline burned to expel those 5.9 kg of CO₂. Establishing that a litre of gasoline produces 2.3 kg, leads to the conclusion that the effect produced to the atmosphere by brewing a 40 litre batch of beer is the same than burning 2.56 litres of gasoline (not including the amount expelled during the exploitation process).⁴¹

So, in view of the high amount of petrol (or gasoil, with even more CO₂ expelled) consumed every day, the environmental consequences are roughly negligible.

On the other hand, the project may also raise interest within the homebrewing members, resulting in a positive impact for the actual community and to help persuading new people interested in beer.
10. Conclusions

It is not easy to mark out concrete conclusions beyond assessing and verifying the correct functioning of the program created. Even so, the present work started from some well-defined objectives that have been gradually completing point by point until finishing the project.

It should be emphasized the importance of an adequate structure as well as a clear organization when starting an assignment of these characteristics. It should be noted that it has managed to implement the base (hardware and software) to be able to semi-automatically control the brewing proceeding without forgetting at any time the homebrewer essence and leaving the doors open to continue improving the idea and implement new functionalities.

Taking into account that it has managed to have a fairly stable version of what could be perfectly a semi-professional product the latter has even more sense.

Another relevant reminder is the importance of choosing the right hardware, without gambling unnecessarily or rushing. This may slow down the early project decisions but it allows to build a more solid and concise structure.

A further outcome is the presence of secondary tasks that, without considering them from the beginning, have helped with the development of the idea and boosted the possibilities to complete it. These ones, like the recipe parser, have contributed in the project constituting a great added value.

Finally, as a particularity, the fact of have spent some considerable time to the world of beer during all these years and to have turned it into something more than a hobby, has helped with the accomplishment of this task. This permitted to be aware of feasible details that could escape in sight of an inexperienced person. This same aspect, makes to want to continue working on this project so it can serve as a model of inspiration for other future homebrewers.

A final conclusion could be that it is unequivocal that the project is still alive and a lot of work still has to be done to complete the whole idea, although it will never end.
10.1. Future implementations

As a final point, as already mentioned, there is a significant interest in continuing to develop the project, not only for personal reasons or interests but also to continue contributing in the homebrewer community.

Being not an essential part of this project and does not concerning directly with the report itself, only some future additions are considered important to incorporate in the idea that has been done until now.

- At the level of hardware and equipment, obviously, the first thing to do is build the prototype. This means obtain the adequate metal profiles, the necessary instruments to heat the liquid (an additional butane burner and the thermal resistance), all the plumbing elements and make the relevant modifications in the containers.
- Merge this control system with the fermentation one that is currently implemented independently, being able to control this last stage also from the CBBS.
- Improve the program by adding new functionalities allowing the production of more complex recipes, with several mash steps, or modifications in the water profile.
- Add the possibility to communicate with the user and be able to control the process from a Smartphone or receive alerts anywhere.
- And of course, increase the frequency of brewing beer and the batch size, as well as the speed of production and the quality of the resulting beers.
Acknowledgments

Once and for all, on a personal note, I would like to thank some people who have aided me during all this time and my studies as engineer.

Foremost, I would like to express my sincere gratitude to my supervisor, Manuel Moreno. Thanks to his help, his encouragement and all his advices. He consistently allowed this report to be my own work, but steered me in the appropriate direction whenever he thought I needed it.

I would also like to thank my brother for all his help and guidance not only in the course of this thesis, but also all these years brewing and enjoying beer.

I specially thank my fellow classmates and friends for all the support given during my university journey. Also, declare my gratitude for all the professors and personal from ETSEIB, for empowering me to grow as a student and person.

The homebrewing community also deserves an honourable mention for sharing their knowledge (as well as their beers) to build the solid foundation which I based my idea.

Last but not least, I must express my sense of gratitude for my parents, whose work provided me the strength to achieve my purposes, and their patience to encourage me throughout my years of study and those to come.

— David Catalán Conde
Bibliography and references

The following books have been read or consulted to comprehend the different aspects needed to develop this project:


References:

[10] https://beerandbrewing.com/dictionary/6811TT0AJU/kilning/
[14] https://www.brewersassociation.org/presentations/who-are-we-brewpub-micro-or-a-hybrid-of-both/
[16] https://birrapedia.com/cervecerias/de-espana?cl_tipo=ar


[19] https://missionarybrewer.wordpress.com/2012/02/01/the-brewers-window-what-temperature-should-i-mash-at/


[23] https://www.brewhardware.com/category_s/1844.htm


[27] https://www.vishay.com/docs/31014/ccf50.pdf


[29] https://www.onsemi.com/pub/Collateral/PN2222-D.PDF


[31] https://grabcad.com/

[32] https://www.python.org/

[33] https://github.com/hotzenklotz/pybeerxml

[34] http://beersmith.com/blog/2008/04/20/calculating-hop-bitterness-how-much-hops-to-use/

[35] https://etseib.upc.edu/ca/lescola/estructura-i-organitzacio/documents%20normatives/grau/Normativa%20TFG


[37] https://goo.gl/NykDQ2

[38] https://lifefermented.wordpress.com/2014/01/22/how-much-co2-is-produced-from-brewing/


License

Design of a semi-automated home brewing system by David Catalán Conde is licensed under a Creative Commons Reconocimiento-Compartir Igual 4.0 Internacional License.