Report 3 Final Project
Automatic Coffee Maker "UniCoffee"

1st Camilo Duarte
Universidad de Ibagué
Ibagué, Colombia
2420162021

2nd Manuel Lopez
Universidad de Ibagué
Ibagué, Colombia
2420161047

2nd Leisson Polo
Universidad de Ibagué
Ibagué, Colombia
2420171094

***Abstract*—Our "coffee maker" project consists of several stages, the first of which is a coffee machine web page where people create an account where they link to a database of information. data, where we administer and visualize the person who is in arrears and the users account recharge requests, they can recharge their balance and look at the current status of your account, change profile picture, make suggestions, talk for the blog, and the most important thing is to order your coffee. Our users make the order of your coffee per page and you will be automatically deducted from the recharged balance and that was previously assigned by the administrator, when the user presses the button in the page to make your coffee.**

***Index Terms*—Raspberry Pi, Python, Arduino, Coffee Maker, Serial Comunication.**

I. Introduction

This work refers to the final project of digital 3. It is an investigation, study and prototype that tries to concretize an idea of innovation to solve some problem in the university of Ibagué making reference to the Internet of the things. For the project is necessary the use of the IoT (Internet of things) where according to the Group of Internet-based business solutions (IBSG, Internet Business Solutions Group) of Cisco, IoT is simply the point in time at which more "things or objects" were connected to the Internet than people[0]. This leads to the ability to need a more sophisticated device for our project. This device is the raspbeery which is a computer the size of a credit card. It consists of a motherboard on which is mounted a processor, a graphics chip and RAM. It was launched in 2006 by the Raspberry Pi Foundation with the aim of stimulating the teaching of computer science in schools all over the world. This will bring you closer to an advanced technology because the raspberry is possible to upload data to the Internet and inteconnect devices to each other. The language to use will be python as it is a language in which is characterized by its simplicity and easy data collection.

Is there a coffee machine that supplies the need to recharge the energy of the people who are in the extra days and late nights of the laboratories? There is no coffee maker in the engineering labs for the late season, which is extremely important for the nightly performance of students who stay on long overtime days either doing projects or studying for mid-term. In the campus of the University of Ibagué there are coffee machines in different points, but these are far from the zone of engineering laboratories, also, these points of cafeteria have a schedule that does not satisfy the need of the student in late hours. It should be noted that the only cafeteria that meets the need is only available in the university library's late-night days. The problem lies in the fact that these late nights are a minority compared to the extra days that the students of the Electronic Engineering program do. The solution to this problem from the innovative point of view is based on the use of IoT (Internet of Things) which is a trend that is currently lived in technologically developed countries where IoT is of utmost importance in different fields of application. IOT (Internet Of Things) platforms are key elements for smart city initiatives aimed at improving citizens' quality of life and economic growth [1]. The field of application of the project enters the part of intelligent cities, where digital interconnection will be made between different machines or objects. The IoT is used in HomeKit technology which is the name of Apple's home automation framework for developers. Apple introduced HomeKit as part of iOS 8 in September 2014. With HomeKit, iPhones and iPads will have a rational way of configuring, communicating and controlling "the Internet of things" around us, including lights, speakers, security systems, connected devices and many more accessories [2]. In the present case a connection will be made between a Raspberry and a technological device with Internet access. The device in formal definition is understood as an embedded system, we understand as embedded system to any digital electronic system based on microprocessor or microcontroller that has a specific purpose. We have an infinite number of devices at home that meet this definition of embedded system. In fact, in most homes there is only one microprocessor-based digital electronic system that is not for a specific purpose, the PC. Or maybe two, the mobile phone, which is also a general-purpose system. Although the mobile phone in its origins was not a general purpose system as we know it today. In fact the mobile phone was an embedded system to which more and more functionalities were added until it became a general purpose system. On the other hand, embedded systems have an enormous importance in all areas of technology. And this importance will continue to grow with the arrival of the Internet of things, which is nothing more than systems,

most of them embedded, connected to each other through the Internet [3]. So we proceed to design an intelligent device, which satisfies the need described above, where the coffee machine can be running 24 hours in the area of engineering laboratories. The idea is to use a Python web framework called Flask to convert the Raspberry Pi into a dynamic web server. Although there is a lot you can do with Flask "out of the box", it also supports many different extensions to do things like user authentication, form generation and database use. It also has access to the wide variety of standard Python libraries that are available [4]. With this you will be able to communicate with the Raspberry and have it run the coffee server process. The process is done in Python language so that the programming satisfies the serving of an American coffee correctly. The key to this freedom is that Flask was designed from the beginning to be expanded. It comes with a robust core that includes the basic functionality that all web applications need and expects the rest to be provided by some of the many third-party extensions in the ecosystem and, of course, by you. In this book I present my workflow for web application development with Flask. I don't pretend that this is the only true way to build applications with this framework. They should make my decisions as recommendations and not as a gospel. Most software development books provide small, focused examples of code that demonstrate the principles and principles of this framework. They should make my decisions as recommendations and not as a gospel. Most software development books provide small, focused code examples that demonstrate the different characteristics of the target technology in isolation, leaving the "sticky" code that is necessary to transform these different characteristics into a fully functional application that the reader must understand. I take a completely different approach. All the examples I present are part of a single application that starts very simply and expands in each successive chapter. This application starts life with just a few lines of code and ends up as a very well presented program [5]. And thus contributes to create a database that communicates with mobile devices for the proper operation of the coffee machine. All this with the general objective of designing a coffee machine to which you can place orders through a website.

II. Experimental method

A. To search for, analyse and interpret some marked problems in the University of Ibague and its surroundings.

After an investigation and look around the university of Ibague it can be observed that there is no coffee pot or something energy drink in the engineering laboratories for the late season and extra class schedules.

B. To design a solution to university problem.

Once we analyze and interpret the problem, we proceed to design a solution to the problem found. In order to do this, we think about innovating and making use of a global trend that is very marked in developed countries, such as the internet of things. In order to do this, an exhaustive research and search for information about the branches of the internet of things is carried out. So, it was determined to solve the problem through the branch of intelligent cities, this because the problem is carried out in a static place which will be interconnected with other machines and thus referring to the set of intelligent cities.

For the connection between the machine and the user is made use of a web page on the Internet, this page is designed through the Python framework called Flask that allows you to create web pages easily. This website will have a database for the purchase of coffee, in addition, is initially locally in the Raspberry and then make the communication with the coffee machine. Finally, the solution will be that the coffee maker will be installed in the laboratory corridor without the need for an intermediary so that the student/user does not need to go to the machine to give the order and wait for the coffee, but rather, that from the area where the student is located request through the website the coffee and that in the course of going to the coffee maker is already have the coffee ready.

III. Results

The project was divided into 2 parts, the first part is to create the website through the Python framework called Flask. In this way, make an eye-catching interface, where you will find different tabs with which you can interact as can be seen in the following figure 5.

Finally, the last tab of the page called order will be in charge of sending the signal to the Raspberry and that this begins to execute the work of preparing and serving the coffee automatically and seriously.

For this mostly mechanical part it was used: 1 servomotor, plastic jars, a Raspberry, an arduino, cables of connection, battery of 12 volts, 1 water heater, 1 relay to 12 volts and a pump dc to 12 volts, in addition to them luminous Where you program the Raspberry in Python with the program called geany that brings the Raspberry by default. For this the pins are configured as GPIO which is a system that can place the pins of the Raspberry as inputs and outputs for general use in the same way as a microcontroller. For the storage of coffee is made a mechanism with 2 jars, one with a larger diameter than the other so that the small jar can rotate within the large jar and that way will open a hole to both jars so that when the servo rotate the small inner jar, the hole opens, closes and thus dispense the coffee as can be seen in the following figure 1.

To move the servomotor was used arduino communicating with the raspberry through serial communication as can be seen in the following figure 2.

And to continue with the actions of the preparation of the coffee the code is made with remote tests. It was determined that the time of filling of water of a glass of coffee is of 3.7 seconds, also it was determined that the time of heating of the water is of 3 minutes for a good coffee and finally it was observed that with 3 cycles of movement of the servomotor the exact measurement of an American coffee was obtained. Finally the code is as follows as can be seen in the following figure 3.

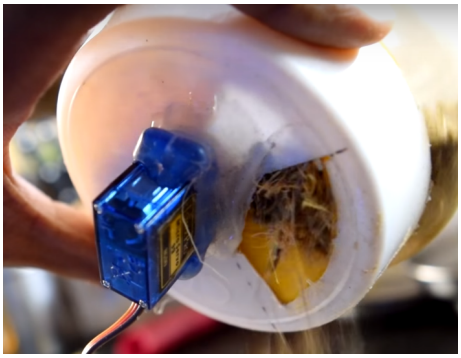

Fig. 1. Coffee Dispenser

```
Servo servo1;
int input = 0;

void setup() {
    servo1.attach(13);
    Serial.begin(9600);
}

void loop() {
  servo1.write(0);
 if ( Serial.available() > 0 ){

    input = Serial.read();
    if ( input == 48){
      servo1.write(180);
      delay(500);
      servo1.write(0);
      delay(500);
      servo1.write(180);
      delay(500);
      servo1.write(0);
      delay(500);
      Serial.println(input);
      delay(1000);
```

Fig. 2. Arduino Code

The final design of the coffee maker can be found in the annexes section

IV. Analysis and Conclusions

The project that was carried out contributed to the solution of the problem that was presented at the University of Ibague. It was necessary the use of an arduino and serial communication with python because the operation of the servomotor in the raspberry was not efficient.

The operation of the coffee machine is effective, has a good quality of coffee and energy consumption is low.

It was not possible to use the water level sensor, because when heating the water the level sensor presented problems.

This prototype can be applied throughout the university, for example. In the libraries and laboratories when there are extended hours, this directed for those lovers to a good coffee and that keep them active and awake in the works or projects that carry out at that moment.

V. References

[0] Dave Evans , "Internet de las cosas", Document distributed by Cisco Internet Business Solutions Group (IBSG)

[1] M. Fahmideh and D. Zowghi , "An exploration of IoT platform development".

[2] Racero Valcárcel, A. R , Proyecto Fin de Máster Ingeniería Industrial Integración de Raspberry en Domótica de Sistema HomeKit. Retrieved from https://idus.us.es/xmlui/handle/11441/86124

Pascual Vázquez , Proyecto Fin de Carrera Miembros del Tribunal Calificador. Retrieved from http://oa.upm.es/49301/

[4] Richardson, M. y Wallace, "Getting Started with Raspberry Pi"

[5] Grinberg, M. , "Flask Web Development"

VI. Annexes

The attached figures corresponding to the project are shown below

```
#!usr/bin/env/python

import RPi.GPIO as GPIO
import time
import serial

GPIO.setmode(GPIO.BCM)

GPIO.setup(5, GPIO.OUT)
GPIO.setup(6, GPIO.OUT)
GPIO.setup(13, GPIO.OUT)
GPIO.setup(19, GPIO.OUT)

time.sleep(5)

#El arduino hace mover el servomotor

arduino = serial.Serial('/dev/ttyACM0', 9600)

Luego se prende la resistencia que calienta el agua por 3 minutos

GPIO.output(5, GPIO.HIGH)
GPIO.output(6, GPIO.LOW)
GPIO.output(13, GPIO.LOW)
GPIO.output(19, GPIO.LOW)

time.sleep(180)

#Finalmente se prenden los leds, se apaga la resistencia calentadora
y se activa la bomba que deja caer el agua caliente por 3.7 segundos

GPIO.output(5, GPIO.LOW)
GPIO.output(6, GPIO.HIGH)
GPIO.output(13, GPIO.HIGH)
GPIO.output(19, GPIO.HIGH)
time.sleep(3.7)
```

Fig. 3. Python Code

Fig. 4. Internal coffee maker

Fig. 5. Website Design

Fig. 6. Student Sample

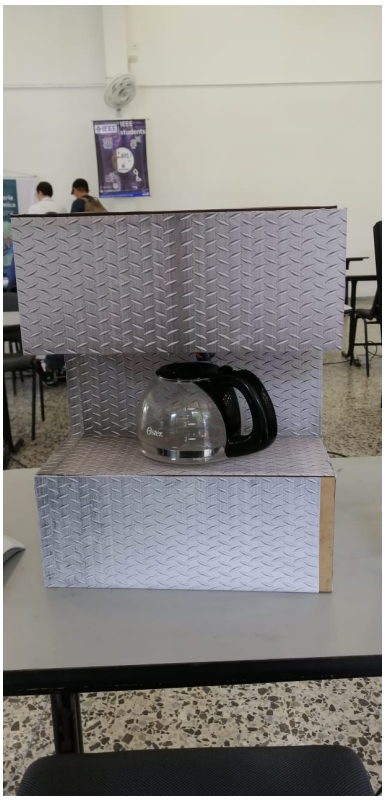

Fig. 7. Coffee Maker Externally

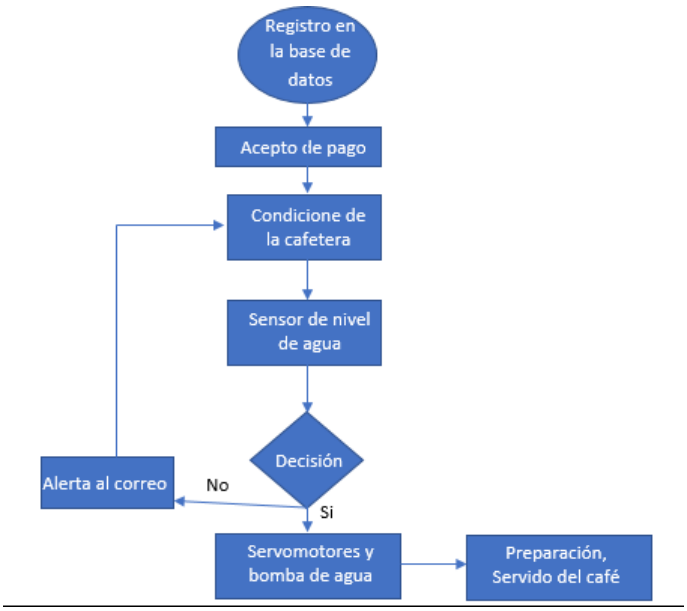

Fig. 8. Flowchart