Federal University of Technology - Paraná - Brasil – UTFPR
Academic Departament of Eletronic – DAELN
Academic Departament of Informática – DAINF
Computer Engineering
Integration Workshop 3 (EEX23) – S71 – 2019/2

---

Technical Report

# Firebot - A FireFighter robot design and construction

Erik Ryuichi Yamamoto – erikyamamoto@alunos.utfpr.edu.br
José Vitor Moreno – josemoreno@alunos.utfpr.edu.br
Juan Simão Mansano – juanmansano@alunos.utfpr.edu.br
Luís Henrique Beltrão Santana – luis.2012@alunos.utfpr.edu.br

November, 2019

**Abstract**

This project was designed as a way to help firefighters in their daily work, to prevent unnecessary risks. So, the Firebot is a remote controlled robot that can move, has a camera, throws a jet of water (with controllable angle and its water flux is controlled by a valve) and a temperature sensor. In order to give control to the user, there's a base with a front-end implementation where the camera image (sent via UDP protocol) and the sensor readings are shown. A joystick that is connected to the base is used to control the robot. To make sure that the commands are sent between the base and the robot, there's a back-end implementation that works as a server to the remaining of the project.

## 1 Introduction

Firefighter activity is an extremely dangerous profession. There are numerous cases where firefighters suffered accidents while working to minimize life losses in fires and accidents.

Environments with fires are extremely hazardous. For instance, in an industry, there is a risks of explosion of flammable products employed in the production process, collapse of the building because a burning structure is subject to structural failure and collapse at any time due to steel expansion within reinforced concrete.

Many times the Firefighters aren't aware of the complete situation before reaching the location, which can make the situation even worse.

In this context, it is clear that a system that allows firefighters to respond to a dangerous call without endangering additional human lives is needed and a remotely controlled robot, operated by an experienced firefighter, is the best solution.

This work describes Firebot, a solution composed of one or more mobile robots, a Base Station comprised of a server and a laptop computer to act as Front-end, to help firefighters to avoid this dangerous situations without endangering additional human lives [1]. The figure bellow is a schematic that shows how the Firerobot is organized:

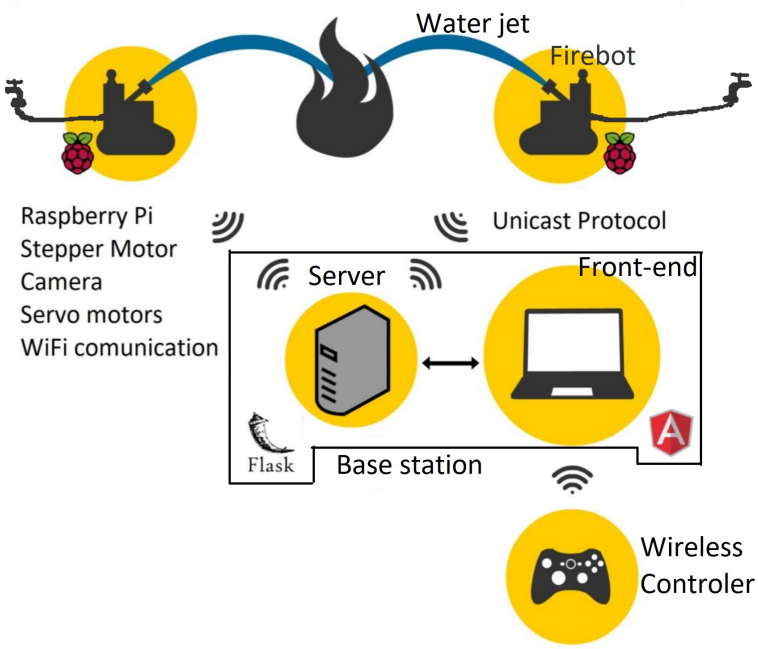


Figure 1: Firebot schematic

Figure 1 shows that the Firebot solution is comprised of one or more robot that uses a Raspberry PI, two DC motors to move the robot, a stepper motor to open the water valve, two servo motors to control the camera direction and another one to control the hose vertical position and has a temperature sensor. The robot sends the data to a server (developed in Flask [2], as a back-end). In a front-end, on the notebook, there is a system developed in Angular [3] that shows to the user the interpreted data received by the robot and interprets the data of the user's wireless controller. The front-end and back-end composes the Base station.

All possible robots are controlled by the same base station, which can switch to each one desired to be controlled. In this implementation, one complete robot was developed to be controlled by the user and there is a computer that responds to the server as a robot, but is not capable of doing anything related to the requirements of this project (presented further), is only a way to present the multi-robot of the Firebot project.

## 1.1 System requirements

Table 1 shows the main requirements of the Firebot project.

Table 1: Firebot requirements

| Robot requirements | |
|---|---|
| **Code** | **Requirement description** |
| FR01 | The robot will be able to perform movements remotely, by commands from the Base Station, like walk forward, walk backwards, turn left and turn right; |
| FR02 | The robot will have a Temperature Sensor that will show its temperature; |
| FR03 | The robot will move by track system to ensure movement in the most adverse terrain possible; |
| FR04 | The camera will be able to move 180 degrees, both horizontal and vertical; |
| FR05 | The robot will be able to compensate its track system so that the water jet's reaction force does not move it backwards; |
| FR06 | The robot will send the image from the camera to the base station; |
| Base requirements | |
| **Code** | **Requirement description** |
| FR07 | Live robot video streaming will be displayed on screen; |
| FR08 | The measurements collected from the temperature sensor will be displayed on the screen; |
| FR09 | The User can control the robot through a joystick; |
| FR10 | The User can control the intensity of the water jet through a joystick by opening the valve; |
| FR10.1 | The User can choose three position of the valve: opening of 0%, opening of 50% and opening of 100%; |
| FR11 | The base station will be programmed for use on a computer; |
| FR13 | Many firebots can be controled by the same Base Station. |

The following sections in this report will present the technologies used in the project, the development, the tests and results, along with the problems on the implementation and the conclusions.

# 2 Tecnologies

In this section the technologies used to implement the Firebot are presented and discussed.

## 2.1 Robot

This subsection presents technologies used in the robot.

### 2.1.1 MG996R Servo Motor

This motor [4] is more powerful than the common use servo motors. It has a stall torque of 9.4 kgf.cm, while a simpler motor has a 2.5 kgf.cm. For this kind of operation, this motor needs 5 V and a current range from 0.5 A and 0.9 A, depending of the weight that is being lifted.

### 2.1.2 28BYJ-48 Stepper Motor

28BYJ-48 [5] is a simple stepper motor, has a reduction box, can be operated from 5V to 12V, depending on the model, can require up to 0.5 A.

### 2.1.3 DC motors for robot movement

To move the robot, a specific DC motor normally used on car windows was used, that works with 12V and has 9.1 Nm of maximum torque [6].

Another point of interest in this motor is that it's waterproof. But, even with this specifications, it's not much expensive.

### 2.1.4 DHT11 sensor

The DHT11 [7] is a temperature and humidity sensor. It has a range from 20 to 50 ºC and, depending on the temperature, a humidity range from 20%RH until 90%RH. It has a resolution of 1ºC with 8 bits. The response time is not very quick, going from 6 seconds until 30 seconds, but the typical is 10 seconds.

### 2.1.5 Raspberry Pi

Raspberry Pi [8] is a small computer that can be used by connecting to a monitor, keyboard and mouse. For the Firebot, the team choose the Raspberry Pi 3 B, because it provides all the features necessary to all requirements of the robot. It was released on February 2016, and is equipped with a 4 core Cortex-A53 CPU, 1GB of memory RAM, wireless communication, 40 communication pins, a slot for a camera and a slot for a micro SD card.

### 2.1.6 Camera: Raspberry PI 5mp

This is a camera made to be used on the Raspberry Pi. It is a 5 megapixels camera so it can provide the necessary conditions and has a good cost benefit. [9]

### 2.1.7 H Bridge

The H Bridge is a potency electronic circuit of E class chopper type. So, it can determine the current direction, the tension polarity and the tension of a given system. The used one was the L298N [10].

### 2.1.8 ULN2003

The ULN2003 is a driver that allows a easily use of the stepper motor 28BYJ-48. [11].

## 2.2 Base station

This subsection presents the base station technologies that were used.

### 2.2.1 Angular

Angular [3] is a web application and front-end development platform, based on TypeScript. It provides an environment to the developer create a complete application, with necessary compatibility to work with many different types of systems. This development platform is efficient and easy to use, providing all the necessary features to create the front-end application of the Firebot project.

### 2.2.2 Flask

Flask [2] is a small web framework wrote in Python. Is a platform to develop the back-end connection of micro servers. It was created to be simple and extendable. This development platform is simple to use and provide the necessary features to create the front-end application of the Firebot project.

### 2.2.3 RabbitMQ

RabbitMQ [12] is a open source message broker and was chosen because it's a free message broker that provides a complete interface to make the communication between the robot, the two servers and the front-end.

### 2.2.4 Heroku

Heroku [13] is a platform that supports applications on the cloud. This kind of platform has support to many languages, including Python. Since can be used for free, the platform will work as a cloud server of the project, maintaining a more continuous connection.

### 2.2.5 Raspbian

Raspbian [14] is an operational system, based on Debian, created to work on the Raspberry PI. It's simple to install and have support to many programming languages, include Python, that is used in the project.

## 3 Project overview

The Firebot robot is 57 cm long, 25 cm wide and 35 cm high, its chassis is made of wood and is equipped with a track system made of bicycle crown and motorcycle pinion with the gears acting as a reduction box for an increase in the traction force of the movimentation system.

Preliminary calculations show that the system has a torque force of over 800N, being able to face several challenges that the terrain may impose on the system during its action in service.

As can be deduced by the requirements, Firebot is capable to move forward, backward, turn left and turn right and it hose angle can be controlled to adjust vertically, the horizontal adjust can be achieved by turning the entire robot, it have a built in camera and sensors to measure the temperature of the robot for self-preservation.

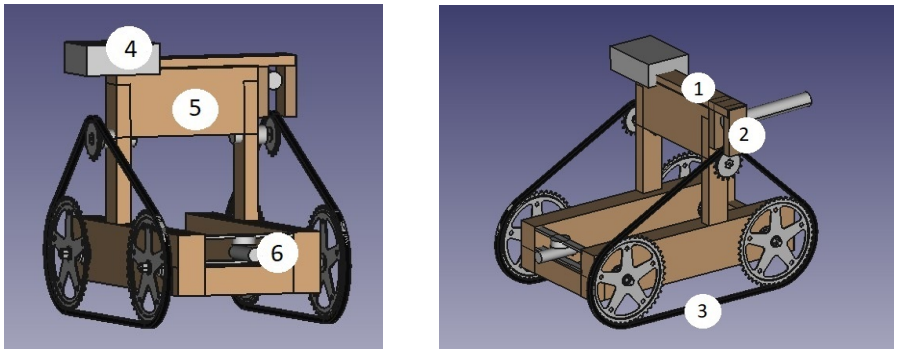


Figure 2: Robot design

Figure 2 shows the robot structure. The "Hose nozzle control system" (2) adjust the angle which the water jet is thrown and has only vertical movement. The camera (1), above the structure, shows the robot "point of view" for the user. The "valve control system" (6) controls the flux of water on the hose. The robot uses a track system to move (3), has a battery close to the center of gravity and the hardware is above the hose (5), with the temperature sensor close, bellow the camera. The Raspberry PI (4) is at the side of the camera, protected by a wood shield.

In Figure 3 there is a chart of the Firebot connections:

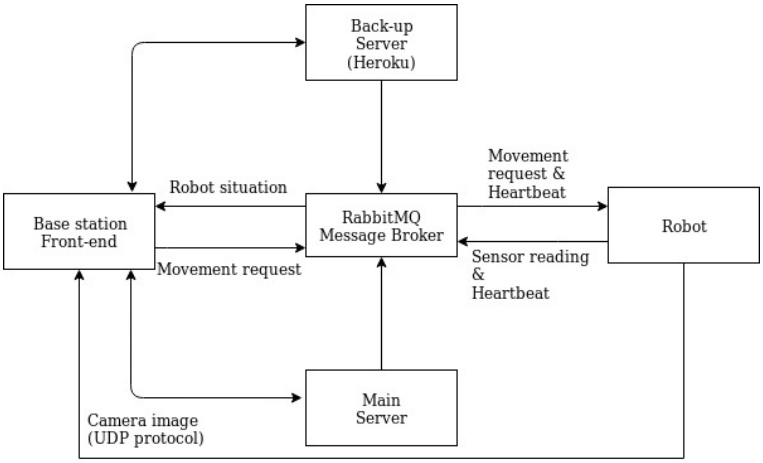


Figure 3: Connections

The front-end receives and treats commands and camera images so it can be shown to the user on the display. The camera images are sent directly from the robot to the front-end, while the robot readings passes through the back-end.

As Figure 3 shows, the front-end sends the robot commands, received from the wireless controller, to the back-end. The back-end sends the commands, received from the front-end, to the robot.

The message broker is a central part of the robot connection, because it's almost a controller for the back-end. It receives the commands and put them on queues, to be available to be read from the back-end and the robot.

# 4 Development

This section describes the development of the Firebot.

## 4.1 Robot structure

This subsection presents the development of the robot structure, that contains the mechanical and hardware parts.

### 4.1.1 The construction

For the construction of the robot base, that supports all electronic and the modules, wood was chosen as the raw material. Pieces of beam of cambara, a very hard and difficult to break wood, were used.

The holes for the bicycle hub were opened with a drill, with the material being gradually removed. In order to allow the bicycle hub to snap into place and allow a good fixation with a flat bolt, the hole area has been cut triangularly, and was attached with screws later. The edges of the hole have been worn with

a rasp to acquire the conical shape of the bicycle hub and to allow a proper fit. As shown in Figure 4.

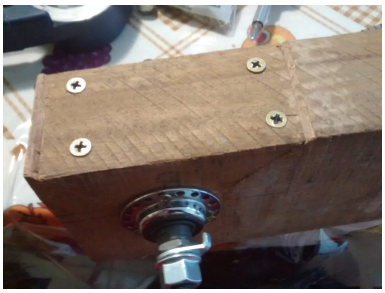

Figure 4: Bycicle hub spot

To attach the bicycle crown to the hub, a washer was used and for the finishing of the weld a grinder was used.

The final part required the use of the motorcycle pinion, but to be usable, was necessary to cut off on half, on the radius direction. After that, it was adapted to be attached to the glass window motor with Durepoxi glue.

To attach the motors to the brackets, a wing nut was used and the bracket shaft was cut to pass through the fixing bolts, so it is possible to adjust the position of the motors to stretch the chain being used as a belt.

To finish the construction, the motor was placed, with the pinion and the motor chain, in order to end the track system. The final result is shown in the Figure 5.

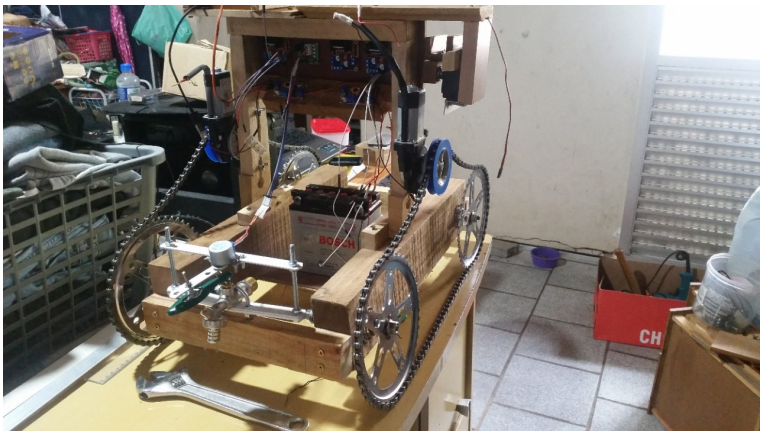

Figure 5: Robot structure

## 4.2 The modules

The modules are complete robot sub-systems, that use the actuators and have specific functionalities. Because they are parts of the robot, they have a mechanical structure, that was adapted on the robot structure later, with a software part, that was used on the embedded system as functions.

### 4.2.1 Valve control system

This module uses the 28byj-48 stepper motor, and the whole system is shown in the Figure 6. The movement of motor works with the half-step method. The half-step was stronger to move the lever than the full-step and required less amount of current. To increase the time for the stabilization of the motor on each pulse, the delay used between them was of 20 ms. This type of adjustment increases the torque of the motor, since it gets the maximum force due to the maximum current passing through.

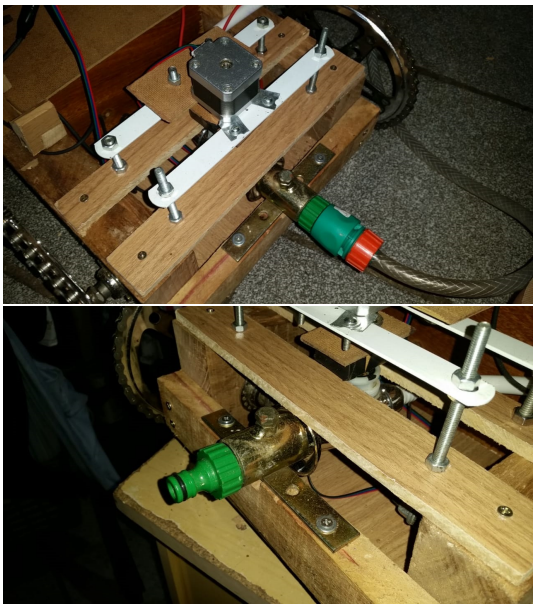

Figure 6: Valve control system

### 4.2.2 Hose vertical control system

The robot has a system to control the vertical angle of the hose. It uses a servo-motor to make the movements as shown in the Figure 7.

The servo motor used in this part is the MG996R from TowerPro, because it has enough torque to lift the wooden structure holding the hose nozzle and can support the force caused by the water that will pass through.

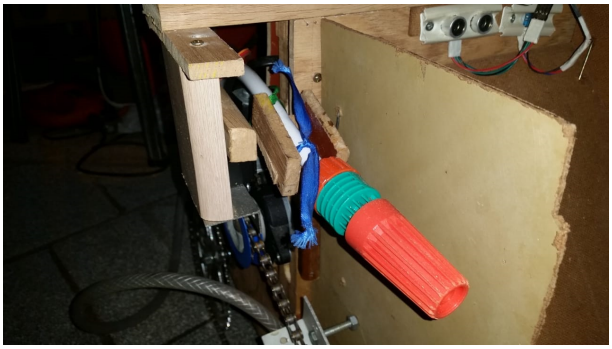

Figure 7: Hose nozzle control system

### 4.2.3 Camera movement control system

This module has two degrees of freedom that allows the movement on the vertical and the horizontal axis. These movements are controlled by two servo motors of the model SG90.

The base was bought on the internet, is stable and made of metal ,made for this purpose. In the final form is shown in Figure 8.

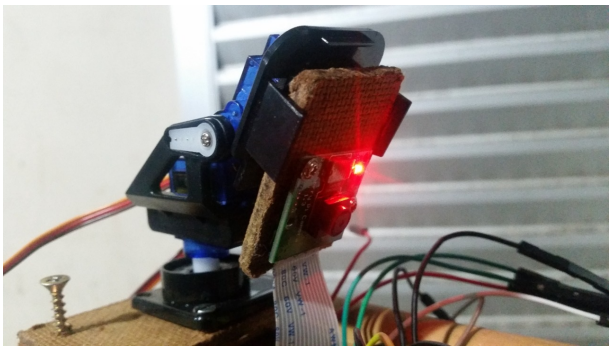

Figure 8: Camera Position Control System

## 4.3 Embedded system

The use of Raspbian as the operational system makes a robust system coordination. This is necessary because, along with robot movements, there's the communication with the base that needs to be interpreted and treated. The robot needs to keep working even when receiving data. It's not a real time system, but the response time needs to be efficient. Along with that, is easy to configure the camera and the WiFi connections.

## 4.4 The integration of the modules

The integration of the modules is composed of two parts: the mechanical integration and the software integration.

The Hose nozzle movement control system was placed on the right part of the top of the robot structure, so it can project the water jet forward. The camera was placed on the top of the robot, so it can have a good freedom of movement and a good vision. The valve control system was fitted behind the robot, because of the necessary space for this module and to be close to the water source. The hose that connects the valve with the hose vertical control will pass bellow the hardware part, to avoid problems with water leakage.

The software part consists in adapting the preliminary functions that were made and make their control as functions of the operational system, being called when necessary.

The hardware was fitted on the center of the robot bellow the camera, so it can be above the hose and the ground, providing some defense against water. The temperature sensor will be fitted together with the hardware, so it can read the temperature of the robot. The battery was fitted on the center of the robot, below the hardware.

## 4.5 Base station

### 4.5.1 The front-end

As a front-end made in Angular, it presents the camera images and the sensors readings on a screen. Along with it, it reads the commands sent by a controller to control the robot.

This front-end has a REST connection part, that takes care to organize and send data to the back-end (with TCP), treating the data received from it, so it can be presented to the user. It receives a video image stream via UDP protocol. Finally, it makes the readings of the user's controller, to determine the robot movement.

### 4.5.2 The back-end

The back-end was implemented in flask, with python. It has the objective to make the connection between the front-end and the robot system. As a separate server that uses the REST protocol, it can be developed separately from the front-end part. Along with that, the use of this back-end gives the possibility of a backup server. The main back-end server is put on the Heroku servers, since it gives a broad support to be connected and already has a necessary infrastructure for this purpose. When the main server connection fails, there is a possibility of connection with a local back-end back-up. Since the local server can have problems to be connected because of the type of internet network, it was decided to be a back-up server.

The Flask-Python allows the use of RabbitMQ, which is a server-side message broker that allows the use of a server and a cloud backup server. Working as a kind of feed for the system, through a REST API client that will command it, it will manage a REST requests and trigger the MQTT broker when necessary, so the robot always has a connection with the base station and the requests won't be lost, since the RabbitMQ saves the messages on queues.

### 4.6 Network connection

Firebot uses wifi/4G/3G to communicate, so, it uses a server that is in sync with the local server at the base station to ensure a redundant control system.

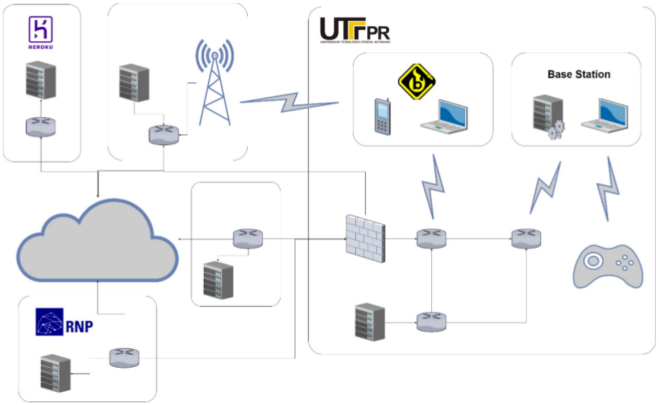


Figure 9: Network connection

Figure 9 Shows how the WiFi connections of the project. The physical server, the front-end and the robot will be sharing the same UTFPR network. In case of disconnection of the robot with the physical server, the robot will make use of the connection with the cloud server on Heroku, using a 4G. The cloud server will make the connection with the front-end, via cable.

## 5 Tests

Many tests were made during the development of the project and, after then, the parts were considered concluded for the final integration. This section will present the final tests that were made after the final construction of the Firebot, in order to conclude if the project meets all the requirements. This is the order that the tests were made, using the base to control the robot:

- Camera movement: Move 180 degrees horizontally and 180 degrees vertically.
- Movement of the servo-motor of the hose nozzle controller: Move from the lower angle to the upper angle.

- Movement of the stepper motor of the valve controller: Opening the valve and closing the valve.
- The two DC motors moving the robot around.
- Interspersed actuation of the modules and the DC motors.
- Connection establishment between the robot and the base.
- The camera images and the sensor readings appearing on the screen, live.
- The movements using the joystick.

At the beginning, some connectives problems appeared, which required modifications on the programming codes. After the connection was stabilized, the rest of the tests were made.

All the tests was concluded successfully, except for test of the water valve, which didn't worked as required. Modifications were made on the structure in order to make it work properly. After that, the same test was remade, and was successfully concluded.

# 6 Problems

Since the beginning of the project, the robot structure was very complicated. To make all the wood parts was demanded a lot of time, because the hardness of the wood and the difficulty of the work.

The metallic parts, that make up the walk system, was the most difficult problem. To cut the motorcycle pinion a lot of time (was expend around 100 hours), which could be used in others parts of the project. If the necessary material and machinery weren't available, this project wouldn't be possible to develop.

The current control and the right leveling were problems that came later in the project. It required more time than predicted and use of different types of power sources until the better one was found.

The problem that afflicted the team more was the lack of synergy. As the time passed, the members started to present different levels of work done. It became critical close to the second checkpoint, where almost all the members made the tasks but the parts that were late was harming the project. One of the members left the team, due to the problems cited, all the task that should be done were rescheduled and a more effective communications was established. Personal meetings to work on the project became more usual.

# 7 Conclusion

The project had different levels of difficulty: the mechanical was the most challenging one, the software was difficult, but the members were more used to it, which minimized the work. In the hardware, the team had problems only with the power supply.

The problem that troubled the team the most was the team coordination. Since most of the team members had a good knowledge of the technologies, the real problem became turning this knowledge into concrete work. It is difficult to do a work you had never done before and more difficult to make prediction schedules. Things can become worse when you don't know how much your teammates are committed to the work.

What the team learned more was how important is to bring teammate together so the work can get done. Along with that, know that in a project, you are in need of your teammates work and the team is in need of you work.

As for more technical conclusions, this kind of project requires the integration of different types of technologies, which, at most part of the time, is developed alone. In this course, it was necessary to put things working together and, to do that many things learned in the university are used to do such work.

At the end, it was clear that a successful project requires: A good project design; A committed team, working in the same pace, sharing knowledge and helping each-other when necessary; A coordinate use of technologies and their implementation.

Table 2 shows the a comparison of hours spent and the hours estimated on the schedule.

Table 2: Table of hours

| Robot requirements | | | | | |
|---|---|---|---|---|---|
| **Activities** | **Estimated hours** | **Hours + 30%** | **Hours spent** | **Spent / Estimated** | **Spent / (Hours + 30%)** |
| Mechanical | 187 | 243 | 224 | 129% | 92% |
| Hardware | 106 | 138 | 168 | 158% | 122% |
| Base station | 75 | 98 | 123 | 164% | 126% |
| Documantation | 85 | 110 | 89 | 104% | 81% |

Table 3 shows the a comparison of the estimated budget and total spent.

Table 3: Budget

| Budget | | | | |
|---|---|---|---|---|
| **Component** | **Quantity** | **Price uni.** | **Total predicted** | **Total spent** |
| Raspberry Pi | 1 | 260 | 219 | 260 |
| Stepper motor | 1 | 27.80 | 55.15 | 27.80 |
| Camera | 1 | 67.30 | 82.63 | 67.30 |
| Tap | 1 | 14.99 | 8.99 | 14.99 |
| Wood | 1 | 60.00 | 12.03 | 60.00 |
| Servo 9g | 2 | 22.45 | 36.32 | 44.90 |
| Bicycle Hub | 4 | 12.00 | 85.98 | 48.00 |
| Bic. Crown | 4 | 14.00 | 26.84 | 56.00 |
| Chain | 1 | 17.00 | 39.25 | 17.00 |
| Misc. | - | - | 228.68 | 213.9 - |
| MG996R | 1 | 45.00 | 0 | 45.00 |
| Camera supp. | 1 | 19.00 | 0 | 19.00 |
| Joystick | 1 | 45.99 | 36.90 | 45.99 |
| Battery | 1 | 80.00 | 61.90 | 80.00 |
| H bridge | 3 | 20.00 | 90 | 60.00 |
| DHT11 | 1 | 5.00 | 15.90 | 5.00 |
| Sprocket | 1 | 22.00 | 26.28 | 22.00 |
| Regulator | 3 | 18.90 | 0 | 56.70 |
| **Total value** | - | - | 1025.85 | 1143,58 |

The following links contains the project's promotional video:
https://youtu.be/luq-b-SVXp4
https://1drv.ms/v/s!AlTZT-x2BicJgoJPb6t4Kw6q0rvURw?e=IuZmJ8
The following links cotains the project's blogs
https://firebotiw3.wordpress.com/

# References

[1] Firebot Team. The firebot blog, 2019. https://firebotiw3.wordpress.com/.

[2] Pallets. http://flask.palletsprojects.com/en/1.1.x/.

[3] Angular Contributors. https://angular.io/.

[4] Tower Pro. MG996R: High torque metal gear dual ball bearing servo.

[5] KiaTronics. 28byj-48 - 5v stepper motor. Download: http://robocraft.ru/files/datasheet/28BYJ-48.pdf.

[6] Denso. Motor specification. Download: https://firstfrc.blob.core.windows.net/frc2016manuals/Denso-Window-Motor.pdf.

[7] Mouser. Dht11 humidity temperature sensor. https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf.

[8] Raspberry Pi. Raspberry datasheet, 2016. https://www.raspberrypi.org/documentation/hardware/computemodule/datasheets/rpi_DATA_CM_1p0.pdf.

[9] Raspberry Pi. https://www.raspberrypi.org/documentation/hardware/camera/.

[10] Handson Technology. L298n dual h-bridge motor driver. http://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf.

[11] Diodes incorporated. Uln200xa. https://www.diodes.com/assets/Datasheets/ULN200xA.pdf.

[12] Pivotal. https://www.rabbitmq.com/.

[13] Heroku Developers. https://www.heroku.com/.

[14] Raspberry Pi. https://www.raspberrypi.org/downloads/raspbian/.