

E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

Q³ Teaching programming to kids using instrumented tangible cubes



Computer Engineering Degree

End of Degree Project

Carla Sotés Garciandía

Asier Marzo Pérez

Pamplona, 19/01/2023

upna

Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

*“People usually know what it has to be done, and you must show them
the path: that’s wisdom”.*
(Christopher Paolini, 2003, [6])

Abstract

Our world is, progressively, entering in a full-technological era. That's why children are learning earlier how to use high-tech.

However, teaching them it's not easy and many schools haven't enough resources for it.

Moreover, this technology revolution is making that children want to develop their computing skills; and the main one is programming. Nevertheless, once more, schools don't use the right ways or haven't the resources. Therefore, those children who would like to study programming at university, won't have the minimum skills to have an easy university start.

That's why Q³ will be that useful for them: they will learn playing, which is the best way to do it.

Table of Contents

| | |
|-----------------------------------|-----------|
| Table of figures | viii |
| Nomenclature | ix |
| 1 Introduction | 1 |
| 1.1 Other projects | 2 |
| 2 Hardware and Software | 3 |
| 2.1 Hardware | 3 |
| 2.1.1 Sensors | 3 |
| 2.1.2 Actuators | 10 |
| 2.1.3 Micro Controllers | 13 |
| 2.1.4 Batteries | 14 |
| 2.1.5 Communications | 16 |
| 2.2 Software | 18 |
| 3 Analysis and Design | 19 |
| 3.1 Analysis | 19 |
| 3.2 Design | 25 |
| 3.2.1 First version | 25 |
| 3.2.2 Final version | 29 |
| 4 Tests and Results | 35 |
| 4.1 Adults tests | 36 |
| 4.2 Children tests | 36 |
| 5 Conclusion | 39 |

| | |
|---|----|
| References | 40 |
| Appendix A: ESP32 Communication Code | 41 |

Table of figures

| | |
|--|----|
| 2.1 Heart rate sensor | 4 |
| 2.2 Accelerometer sensor | 4 |
| 2.3 Piezo sensor | 5 |
| 2.4 Microphone sensor | 5 |
| 2.5 LDR sensor | 6 |
| 2.6 Ultrasonic distance sensor | 6 |
| 2.7 Temperature sensor | 7 |
| 2.8 Button sensor | 7 |
| 2.9 Proximity IR sensor | 8 |
| 2.10 Tilt sensor | 8 |
| 2.11 Magnetic Hall sensor | 9 |
| 2.12 Magnetic Hall sensor | 9 |
| 2.13 CO ₂ sensor | 10 |
| 2.14 LED actuator | 10 |
| 2.15 Laser actuator | 11 |
| 2.16 Motor CC actuator | 11 |
| 2.17 Servo motor actuator | 11 |
| 2.18 Stepper motor actuator | 12 |
| 2.19 OLED screen actuator | 12 |
| 2.20 Speaker actuator | 13 |
| 2.21 Arduino UNO micro controller | 13 |
| 2.22 Arduino NANO micro controller | 14 |
| 2.23 ESP32 micro controller | 14 |
| 2.24 LiPo cell battery | 15 |
| 2.25 Li-ion cell battery | 15 |
| 2.26 Charger manager | 16 |
| 2.27 WiFi communication device | 16 |

| | |
|---|----|
| 2.28 Bluetooth communication device | 17 |
| 2.29 Radio communication device | 17 |
| 2.30 Arduino program skeleton | 18 |
| 3.1 Inputs data table | 20 |
| 3.2 Outputs data table | 20 |
| 3.3 Communications data table | 20 |
| 3.4 Micro controllers data table | 21 |
| 3.5 Batteries data table | 21 |
| 3.6 WiFi-Bluetooth comparison table | 21 |
| 3.7 Tension divider | 23 |
| 3.8 Tension divider | 23 |
| 3.9 Motion cube faces | 26 |
| 3.10 Emotions cube faces | 27 |
| 3.11 Lights cube faces | 28 |
| 3.12 Sound cube faces | 29 |
| 3.13 Sensors and actuators distribution's table | 29 |
| 3.14 Tokens distribution's table | 30 |
| 3.15 Red's tokens associated resistors | 30 |
| 3.16 Blue's tokens associated resistors | 30 |
| 3.17 Green's tokens associated resistors | 31 |
| 3.18 Yellow's tokens associated resistors | 31 |
| 3.19 Lights cube faces | 32 |
| 3.20 Emotions cube faces | 32 |
| 3.21 Sounds cube faces | 33 |
| 3.22 Motion cube faces | 33 |
| 3.23 Cubes tokens | 34 |
| 4.1 Cubes changes | 37 |

Nomenclature

Units of Measurement

- A* Amperage. Measure for electrical current.
- Hz* Hertz. Unit of frequency; number of cycles per second.
- Ω Resistance. Opposition to current flow in an electrical circuit.
- V* Voltage. Physical capacity of an electric circuit.
- W* Watts. Quantifies the rate of energy transfer.

Devices

- Accelerometer* Inclination and speed sensor.
- BMS* Battery Management System
- CO₂* Carbon Dioxide.
- ESP* Espressif Systems Processor.
- Hall* Hall Effect to detect magnetic field.
- IR* Infrared.
- Laser* Light Amplification by Stimulated Emitted Radiation.
- LDR* Light-Dependent Resistor.
- LED* Light-Emitting Diode.
- Li – ion* Lithium ion.

LiPo Lithium Polymer.

MicroController Programmable circuit.

OLED Organic Light Emission Diode.

Piezo Device that generates a voltage.

Reed Electromagnetic switch.

Ultrasonic Sound waves over human hearing skills.

WiFi Wireless fidelity.

Current

CC Continuous Current.

DC Direct Current.

Computing

IT Information Technology

Chapter 1

Introduction

While we are entering in a full technological era, our ways and subjects of study must evolve as we do. It's not novelty that IT is daily getting more into our lives and we must know how to manage it.

As in every generation, children are the future and they are the ones who are getting more comfortable in this new era. That's why they are learning earlier all the skills IT requires.

I took that idea and I decided to create something to help them study and improve their computing skills better than staying in front of a computer for hours not understanding a single word. From my experience I know that it is not a good way to learn. When I entered University I didn't even know what a command was or a variable or an if. I only knew a bit of Word and PowerPoint! Not even Excel!

This project was christened as Q³ and it consists in four cubes that communicate with each other in order to follow the instructions that children will give them. How do they learn to program with them? Easy: the cubes have a pack of tokens -instructions- that will be stocked by the children in the faces of the cubes. These faces will have different sensors, that will call other cubes, and actuators that will receive the orders of the sensors. This way, they will learn and understand what an IF is in computing: **IF** *something happens* **THEN** *another thing will happen*.

1.1 Other projects

This project is based on the idea created by 4 people from the University of Zaragoza called The ifs [\[5\]](#).

My aim is to improve that project and, how am I supposed to achieve it? I investigated why that product wasn't being sold; there even is a project called SaveTheIfs because they didn't make it. The main problem was that they were too expensive (100€ per cube) and even worse: it was very difficult to buy them! Indeed, these were the prizes:

- Early bird: 279€
- Regular: 319€
- The ifs school: 1499€

Besides, all the project has been made thinking about saving money, recycling and saving power too. The goal? Less than 40€ per cube. What has been reached? 41.64€ for the four cubes! It's true that it's just the total price of electronics, but it's far from the initial price per cube.

Chapter 2

Hardware and Software

First of all, I should enumerate all the sensors, actuators, micro-controllers, batteries and communication systems that have been tested. This way, in the next chapter I will discuss and argument why I have chosen or not some of them.

In this section I will describe briefly what they are and how they work.

2.1 Hardware

A hardware is the physical part of an IT. It's made up of different electrical, electronic, electromechanical and mechanical components such as wires, lights, disks, etc.

2.1.1 Sensors

A sensor is a device that senses changes in the environment and transforms them into digital signals so humans can read that data and apply it.

- Heart rate

Heart rate can be measured with electrical or optical methods. The latter consists in a light that shines through the finger and it measures how much light is absorbed. This goes up and down as blood is pumped through the finger. Usually it's a combination of an infrared light sender and a light receiver [3].



Figure 2.1: Heart rate sensor

- Accelerometer

This device measures the vibration or acceleration of a motion object. It uses an electromechanical sensor that measures static or dynamic acceleration. Static acceleration is the constant force acting on a body, i.e. gravity or friction; and dynamic acceleration forces are non-uniform, i.e. vibration or shock [2].

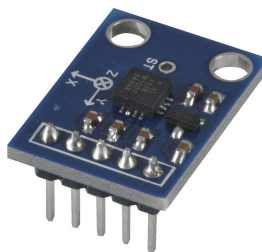


Figure 2.2: Accelerometer sensor

- Piezo

A block of piezo it's a ceramic; thus it's brittle, shatters when dropped, and not very flexible. At this stage it doesn't exhibit piezoelectric properties. However, what we use it's a disk of piezo with silver electrodes. In order to get the usable state, it must be cut to shape, have the electrodes applied to its surface, and then it has to be polarized by applying a very high voltage to its electrodes [1].



Figure 2.3: Piezo sensor

- Microphone

Most of them employ light modulation, piezoelectric generation, capacitance change and electromagnetic induction to produce an electrical voltage signal from mechanical vibration. It has a diaphragm: a thin piece of material that vibrates when it is struck by sound waves. That causes surrounding components of the microphone to vibrate. Conversion of these vibrations is delivered as an audible signal [4].

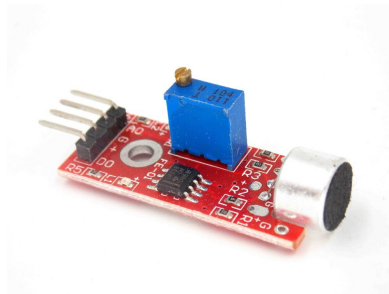


Figure 2.4: Microphone sensor

- LDR

It's a resistor that varies depending on the quantity of light that impacts over the surface. That's what makes them sensitive to light.

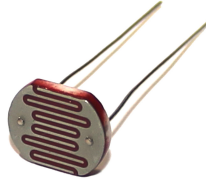


Figure 2.5: LDR sensor

- Ultrasonic distance

This device is made up of two ultrasonic speakers who produce that kind of waves and what it measures is the time that takes the wave to go against an obstacle and coming back to the device.

```
initializeTrigger();  
time = pulseIn(ECHO, HIGH);  
distance = time * 0.000001 * 34300/2;
```



Figure 2.6: Ultrasonic distance sensor

- Temperature

The temperature change is directly proportional to the diode's resistance. The cooler the temperature is, the lower will be the resistance, and vice-versa. This resistance is measured and converted into readable units (Celsius, Kelvin...).

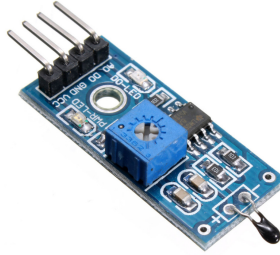


Figure 2.7: Temperature sensor

- Button

Its operation is really simple: when the button is pressed, it closes the circuit which connects the pin to 5V. If not, the circuit is open and no signal is sent.



Figure 2.8: Button sensor

- Proximity IR

It has an infrared emission source that emits light pulses and a photodiode with peak sensitivity at a wavelength that matches the peak intensity of the transmitter. That's how it can detect a target when it enters in the sensor's range [7].



Figure 2.9: Proximity IR sensor

- Tilt

This device detects changes in angle. If exceeds the threshold values, a relay will be engaged.



Figure 2.10: Tilt sensor

- Magnetic (Hall)

It switches on when it detects a magnetic field, applying what is called The Hall Effect.



Figure 2.11: Magnetic Hall sensor

- Magnetic (Reed)

It has a pair of magnetizable metal reeds whose end portions are separated. Moreover, they are hermetically sealed within a tubular glass envelope. A magnetic field will cause the reeds to attract each other, thus closing the circuit. The spring force of the reeds causes them to separate.



Figure 2.12: Magnetic Hall sensor

- CO₂

This sensor measures not just CO₂ but other gases such as Ammonia, Nitrogen Dioxide, etc. It's made up of an electro-chemical that changes the resistor when it makes contact with some substances.



Figure 2.13: CO₂ sensor

2.1.2 Actuators

An actuator is a device that uses the income energy to activate an automatic process such as an engine or a bulb, among others.

- LEDs

These LEDs have three diodes (red, blue and green) inside the same bulb. It has two properties: the color and the temperature. The color is within the rainbow spectre and the temperature is the intensity of the light.



Figure 2.14: LED actuator

- Laser

Laser is the acronym of: Light Amplification by Stimulated Emitted Radiation. Inside, there are two mirrors in each side of the tube that reflects the light as a concentrate current called laser.

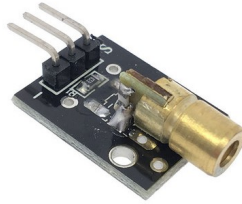


Figure 2.15: Laser actuator

- Motor CC

It's a machine that takes current energy and converts it into mechanical energy causing a rotatory movement thanks to a magnetic field.



Figure 2.16: Motor CC actuator

- Servo Motor

Similar to the CC motor, this one contains a system that allows to control both speed and position.



Figure 2.17: Servo motor actuator

- Stepper Motor

It divides a full rotation into a number of steps. It needs a driver to control it.

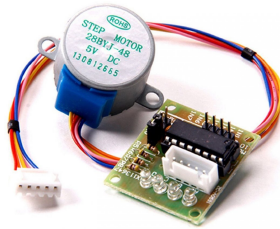


Figure 2.18: Stepper motor actuator

- OLED Screen

OLED stands to Organic Light Emission Diode. They are some organic diodes that emit light without an additional light source, that's the biggest difference. Each diode is a pixel and they are capable of generate their own light, even being completely black.

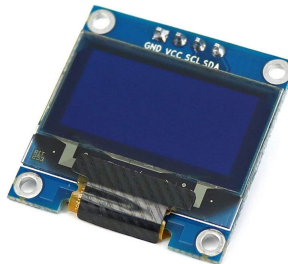


Figure 2.19: OLED screen actuator

- Speaker

Sound moves in pressure waves. When air particles compress fast, we hear sound. Faster the changes, higher the frequency of sound. When a speaker moves forward and backwards, it pushes the air particles changing air pressure and creating sound waves.

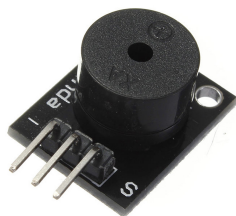


Figure 2.20: Speaker actuator

2.1.3 Micro Controllers

Basically it's a programmable circuit capable of executing orders that have been sent. It has the same functional units than a computer: CPU, memory and I/O pins.

In general, all the micro controllers work in the same way, the only differences are their sizes, if they have other abilities such as Bluetooth or WiFi, etc. That's why here I will explain what is the principal distinctive factor of each one.

- Arduino UNO

It's the base controller, it has some pins and it's quite big. No extra facts. No need of plugins or drivers to be installed in the computer.



Figure 2.21: Arduino UNO micro controller

- Arduino NANO

It's very similar to the UNO controller but smaller. However, is the most expensive and consumes more; nevertheless I will explain this in other sections.

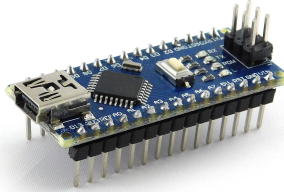


Figure 2.22: Arduino NANO micro controller

- ESP32

It's the most used one. Most of the pins read analogical and digital signals, not as Arduino's that differentiated those pins. Moreover, it has communication skills; therefore, it has Bluetooth and WiFi and it's the cheapest one.

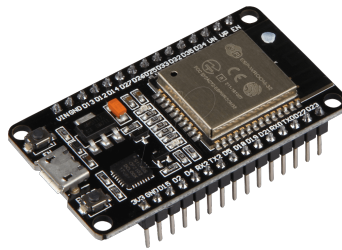


Figure 2.23: ESP32 micro controller

2.1.4 Batteries

Made up of electro-chemical cells, a battery can transform chemical energy into electric energy. It can power different electrical circuits depending on its capacity and power. I searched for rechargeable batteries just because it's more comfortable and affordable than non-rechargeable ones.

- LiPo cell

LiPo stands to Lithium and Polymer. It has more voltage than other types of batteries. It works with exchange of electrons between the negative electrode and the positive one. To forbid the contact of the electrodes, there's a material with microscopic pores that only allows ions to migrate from one electrode to another.

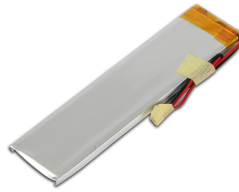


Figure 2.24: LiPo cell battery

- Li-ion cells

The idea is to use recycled ones due to we can find them in devices such as mobile phones, speakers, computers, etc. It uses as electrolyte the lithium salt that gets the necessary ions for the reversible electro-chemical reaction between the anode and the cathode.



Figure 2.25: Li-ion cell battery

- Charger manager or BMS

The idea of this device is to manage the charge of the batteries and to control them so they don't use more than what is needed.

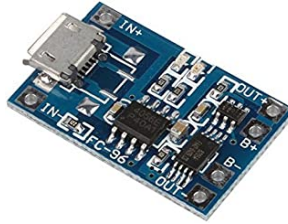


Figure 2.26: Charger manager

2.1.5 Communications

As the project consists in 4 cubes that communicate with each others, it's needed any kind of communication system if the ESP32 is not used. The options are: WiFi, Bluetooth or radio frequency.

- WiFi

It's based on radio waves; therefore, it transmits the information by air using them. However, WiFi is in another frequency, exactly 2,4 GHz to the 802.11n standard and 5 GHz in 802.11ac. The router receives data from Internet and afterwards, it converts them in radio waves that our wireless device has asked for to download the information.

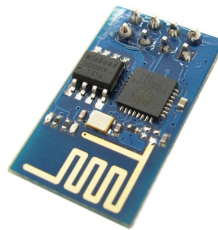


Figure 2.27: WiFi communication device

- Bluetooth

It's based on master/slave mode. This way, there's a net formed by one device and the ones in its range. It's low range data exchange. It transmits in ISMB range: 2402-2483,5 MHz.

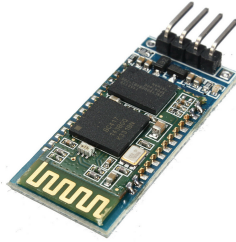


Figure 2.28: Bluetooth communication device

- Radio

Electromagnetic radiation is formed by a magnetic field and an electrical field. They travel in waves that are perpendicular from each other and that can be classified by the wave length. When particles get faster or change direction, they emit radiation among the spectre including wide-range waves.

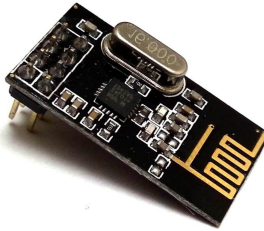


Figure 2.29: Radio communication device

2.2 Software

In this section I will explain which software I have finally decide to use and why.

The best way to use all these input and output devices is to use Arduino. It has many libraries and documentation all around the Internet. Moreover, this project allowed me to amplify my knowledge about this software that I already acquired in the subject "Sistemas Multimedia".

The software also has examples of the different devices that can be used and if it is needed to install libraries for some of the devices (i.e. OLED screen), there are also examples of how to program it.

It has many tools to debug the programs the user has written such as the Serial Monitor or the Serial Plotter that allows the user to see the values of the devices along a live time-stamp and/or the interactions with the system.

A screenshot of the Arduino IDE interface. The window title is "sketch_oct25a Arduino 1.8.19". The main editor area shows a C++ program skeleton with the following code:

```
sketch_oct25a $
1 /**
2  * Libraries and variables
3  */
4
5 void setup() {
6   // put your setup code here, to run once:
7
8 }
9
10 void loop() {
11   // put your main code here, to run repeatedly:
12
13 }
```

The status bar at the bottom indicates the board is "DOIT ESP32 DEVKIT V1, 80MHz, 921600, None en /dev/cu.usbserial-0001".

Figure 2.30: Arduino program skeleton

Chapter 3

Analysis and Design

In this chapter I will argument all the decisions I have made and that have led me to the final result. It's going to be divided into two sections: analysis (where I will describe all the process of the project) and design (where it will be possible to see the different versions of the cubes).

3.1 Analysis

The first step that was taken was to analyzed all the devices that could be used to buy them. However, as one of the goals of the project was the cubes to be cheap and compact, it was needed to pay attention in these facts. The following tables show all the data that was analyzed.

| Inputs | | | | | | | |
|------------------------|---|-------------|----------------|-------|----------------------|----------------|--------------|
| Description | Link to purchase | Total price | Price per unit | Units | Dimensions (mm) | Voltage | Others |
| Accelerometer | https://www.amaz | 12.99 | 3.25 | 4 | 20x15, 20x10 | 3 - 5V | (no info) |
| Piezo | https://www.amaz | 11.59 | 0.39 | 30 | 27 | 3 - 5V | (no info) |
| Mic | https://www.amaz | 8.99 | 1.80 | 5 | 3 (no info) | 3 - 5V (DC) | (no info) |
| LDR | https://www.amaz | 12.19 | 0.08 | 160 | 5 x 4 | 150V max. | 100mW |
| Ultrasonic Distance | https://www.amaz | 9.99 | 2.00 | 5 | (lab ones - no info) | 5V (DC) | (no info) |
| Temperature | https://www.amaz | 20.45 | 4.09 | 5 | (lab ones - no info) | 3 - 5.5V (DC) | (no info) |
| Button | https://www.amaz | 7.99 | 0.10 | 80 | 12 x 12 x 7,3 | 3 - 5V | (no info) |
| proximity IR Sensor | https://www.amaz | 14.99 | 0.75 | 20 | 31 x 15 | 3 - 5V (CC) | (no info) |
| Tilt sensor | https://www.amaz | 10.49 | 0.10 | 100 | 52 x 115 | 12V max. | 20mA |
| Magnetic Sensor (Hall) | https://www.amaz | 8.64 | 0.86 | 10 | 15 | 6.5V max. | (no info) |
| Magnetic Sensor (Reed) | https://www.amaz | 8.99 | 0.45 | 20 | 2 x 14 | 100V max. (DC) | 1A, 10W |
| CO2 Sensor | https://www.amaz | 11.99 | 4.00 | 3 | 30 x 20 | 5V | 180mA, 350mW |
| Heart rate | https://www.amaz | 23.05 | 4.61 | 5 | 20.6 x 15.5 | 3 - 5V | (no info) |

Figure 3.1: Inputs data table

| Outputs | | | | | | | |
|-----------------|---|-------------|----------------|-------|-------------------|-----------|-------------|
| Description | Link to purchase | Total price | Price per unit | Units | Dimensions (mm) | Voltage | Others |
| LED | https://www.amazc | 19.77 | 0.33 | 60.00 | 1000 | 5V (CC) | (no info) |
| Laser | https://www.amazc | 9.49 | 0.47 | 20.00 | 6 | 5V (CC) | 20mA, 10mW |
| Vibration motor | https://www.amazc | 4.85 | 0.49 | 10.00 | 1 x 0.3 | 3V (CC) | 0.1A, 0.01W |
| Wheels | https://www.amazc | 8.05 | 2.68 | 3.00 | 11 x 7 | 3-5V (CC) | (no info) |
| | https://www.amazc | 15.99 | 3.20 | 5.00 | 1.37 x 1.18 x 0.6 | 5V (CC) | (no info) |
| LCD screen | https://www.amazc | 18.89 | 3.78 | 5.00 | 128 x 64 | 3-5V (CC) | 0.08W |
| Speakers | https://www.amazc | 13.99 | 2.33 | 6.00 | 12 x 9.5 | 5V (CC) | (no info) |

Figure 3.2: Outputs data table

| Communications | | | | | | | |
|----------------|---|-------------|----------------|-------|-----------------|---------------|-----------|
| Description | Link to purchase | Total price | Price per unit | Units | Dimensions (mm) | Voltage | Others |
| Wifi | https://www.amazo | 21.99 | 4.40 | 5 | 34 x 26 x 8 | 3.2V | (no info) |
| BT | https://www.amazo | 13.9 | 4.63 | 3 | (no info) | 5V | 30mA |
| IR | https://www.amazo | 7.99 | 4.00 | 2 | (no info) | 5V | (no info) |
| RF 2.4Ghz | https://www.amazo | 13.99 | 1.40 | 10 | 15 x 29 | (no info) | 9mA |
| RF 433Mhz | https://www.amazo | 13.99 | 1.40 | 10 | 19 x 19 | 12V max. (DC) | 10mW |

Figure 3.3: Communications data table

| Microcontrollers | | | | | | | |
|------------------------|---|----------------|-----------------------|------------|-------|------------|-------------|
| Description | Link to purchase | Price per unit | Dimensions (mm) | Voltage | GPIOs | Analog IOs | Others |
| ATTiny85 | https://es.rs-online.com | 1.71 | 10.16 x 7.112 x 4.826 | 1.8 to 5.5 | 4 | 2 | |
| ArduinoRaw (Atmega328) | https://es.rs-online.com | 2.84 | 7.49 x 4.06 x 34.79 | 1.8 to 5.5 | 16 | 8 | |
| ARduinoNAo | https://www.amazon.com | 8.33 | 45 x 18 x 7,5 | 1.8 to 5.5 | 16 | 8 | |
| ESP32 | https://www.amazon.com | 4.4 | 48 x 13 x 26 | 2.2 to 3.6 | 20 | 16 | Wifi and BT |

Figure 3.4: Micro controllers data table

| Batteries | | | | | | | |
|-------------------------|---|-------------|----------------|-------|-----------------|---------|------------------|
| Description | Link to purchase | Total price | Price per unit | Units | Dimensions (mm) | Voltage | Battery Capacity |
| Li-po 1 cell | https://www.amazon.com | 27.99 | 5.60 | 5 | 46 x 20 x 8 | 3.7V | 500mAh |
| Li-ion cells (recycled) | https://www.amazon.com | 12.49 | 4.16 | 3 | 142 x 502 | 3.2V | 600mAh |
| Charger manager | https://www.amazon.com | 6.99 | 1.40 | 5 | 26 x 17 | 5V | (no info) |

Figure 3.5: Batteries data table

After showing all this data, the cheaper devices were selected and the first version of the cubes was made (see 3.2.1 for more details).

The next step was choosing the micro controllers that would rule the cubes; I called them: the brains. It was obvious from the very first moment that the best option was the ESP32 because it was cheaper than the NANO, easier to control (taking into account the project I did in the subject 'Multimedia' in where I used that device and gave to my partner and I many problems) and different communication methods that would be later tried.

After having our "brain", the next step was to try the communication protocols. After reading many documentation, it seemed that Bluetooth was the best way to communicate them and it had even low consumption. In the following image the advange of consumption of Bluetooth against WiFi can be seen:

| | BT | WIFI |
|---|-------|-------|
| A | 0.078 | 0.103 |
| V | 4.96 | 4.96 |

Figure 3.6: WiFi-Bluetooth comparison table

It was quite easy to communicate two ESP32 and send messages one to another with Bluetooth. However, it had more problems; for this project, if Bluetooth was used, it is needed a client-server (or master-slave) configuration. It's not a big deal for two controllers but, what if the client was connected firstly? It couldn't be connected to the other one. That's why it was tried to make that if one cube couldn't find its server, it started as a server. Nonetheless, it didn't work always and that's not a good job. Therefore, ignoring this case, more than one cube couldn't connect to another; in other words, the server cube didn't allow many connections.

The next option was BLE that seemed better because it allowed more than one connection. However, I had the same problem: it was needed to start the first one the server. That's why I gave up trying with BT and I started with ESP Now.

What is ESP Now? It's, indeed, a communication protocol that not many micro controllers have, and ESP32 is one of them; it's low band WiFi communication, that means less consumption. Moreover, it was very easy to program: to include the library in Arduino, discovering the MAC addresses of each brain and, in a send function, write the receptor and the message. Really really easy; piece of cake ;)

Therefore, all Bluetooth protocols increased their consumption when exchanging messages, it's not the case for ESP Now. While BT increased to almost 0.2A, ESP Now stayed at 0.126. What at first seemed the worst option, it turned to be the best one at the end.

The next step was to enable the tokens system: activate or deactivate the sensors and actuators depending on our desires. The system is based on the signal sent by a Tension Divider. It was not need to read the difference of tension caused by this system but to read the signal as if it was another sensor.

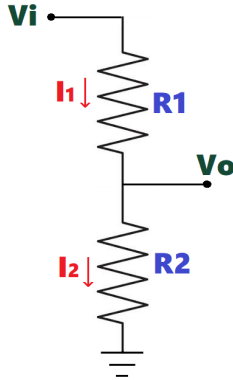


Figure 3.7: Tension divider

The aim was to try different resistor's values and write them down so they could be picked up. The maximum tokens needed were 8 (as it can be seen in 3.2.2). The problem of this system was that some resistor's combinations gave similar values; however, it's easy to solve; as it has been said, it was only needed 3-4 different values in the same instruction (CALL or ON CALL). In the following table, it is possible to see all the values that the different resistor's combinations give; the red-marked ones are the finally chosen, not just because of the values they produced but also it was taken into account that they are the most used ones.

| R1/R2 | 10 | 100 | 220 | 330 | 1k | 2k | 5k | 10k | 100k | 1m |
|-------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 10 | 1762-1913 | 3762-3833 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 | 4095 |
| 100 | 190-588 | 1865-1917 | 2640-2774 | 2831-2896 | 3811-4095 | 4095 | 4095 | 4095 | 4095 | 4095 |
| 220 | 10-52 | 1107-1138 | 1856-1940 | 2290-2404 | 3251-3349 | 3729-3761 | 4095 | 4095 | 4095 | 4095 |
| 330 | 0 | 739-812 | 1386-1511 | 1777-1942 | 2926-3007 | 3465-3501 | 4015-4073 | 4095 | 4095 | 4095 |
| 1k | 0 | 123-243 | 562-602 | 821-875 | 1872-1935 | 2566-2586 | 3314-3453 | 3790-3830 | 4095 | 4095 |
| 2k | 0 | 26-173 | 231-304 | 314-441 | 987-1395 | 1883-1904 | 2786-2819 | 3327-3345 | 4095 | 4095 |
| 5k | 0 | 0 | 0 | 0 | 483-656 | 699-1008 | 1888-1922 | 2546-2571 | 4095 | 4095 |
| 10k | 0 | 0 | 0 | 0 | 114-369 | 509-530 | 1218-1232 | 1868-1905 | 4095 | 4095 |
| 100k | 0 | 0 | 0 | 0 | 0 | 0 | 41-53 | 186-250 | 1897-1921 | 3779-3850 |
| 1m | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 258-320 | 2090-2203 |

Figure 3.8: Tension divider

I will explain briefly how this system works: when there is no token, the circuit is open, so no signal is sent, that means no device is calling or being called; when there is a token, the neodymium magnets close the circuit and there is an analog signal read by the cube's brain that will do whatever action.

For example: in the red cube I put in the microphone the token CALL GREEN; and in the green one, in the speaker, I put the token ON CALL HAPPY BIRTHDAY. When the red cube reads the value of its token (predefined) it knows that it's CALL GREEN in the pin of the microphone so when this one receives a signal, it will send the values to the green cube. At the same time, the green cube will wait for the data and when it receives it, the speakers will play the song. What if there is no CALL token? Then the ON CALL one won't receive the data and will do anything. The other way round, if there isn't an ON CALL token, any cube can react.

Another big problem were motors and the brilliant idea of moving them frontward and backwards. Not every motor can do it. First we tried to modify Servos: taking out the pieces that don't allow them to make a 360 degrees turn. However, still not working. We didn't want Steppers because they were big and needed an extra piece (the driver) but it turned to be the best option.

The following step was to decide the batteries, the best option were Li-ion because they are really easy to find and re-use. However, here there were two types: the purple ones and the blue. The main difference was that the purple had the double of power the the blues (around 5.5V). If the blue ones are not used, they discharge, while the purple ones no. That's why two cubes have purple and the others, blue; so it's possible to see the difference. Moreover, the cubes have inside a BMS (Battery Management System) that controls the charge of the cells so they don't overcharge or discharge. It also has a LED that varies its intensity depending on the amount of charge that the cell has; when is fully charged, instead of blinking, it always brights in red.

Finally, what was left was to join everything and create the four cubes; first without the boxes, just the circuits and then integrate everything. After having tested each part, it's much easier to write the code, software always works (or not).

3.2 Design

I have used a program to design the cubes in 3d. It's called Tinkercad and I used it because it's really easy to manage and not only that but it also has many electronic components such as the sensors and actuators. Even being in the 3d design world, other programs are really difficult to manage and it is compulsory to even design those components.

3.2.1 First version

This version was supposed to show the devices (sensors and actuators) that each cube would have so I had a sketch of a first idea of what I wanted to do. The tool doesn't have some devices, that's why in some of the faces of the cubes there are words with the device is supposed to be.

I must remember that when this first version was thought I didn't understand the tokens and the inputs worked with on/off buttons, that's why I haven't took them into account.

- Motion cube

This cube was supposed to have the sensors: ultrasonic distance, IR (Microwave) and button; and the actuators: LEDs, wheels (two CC engines) and servo engine. With these devices (plus the ESP32) and following the prices I described in the previous section, the price of this cube would be around 14,25€.

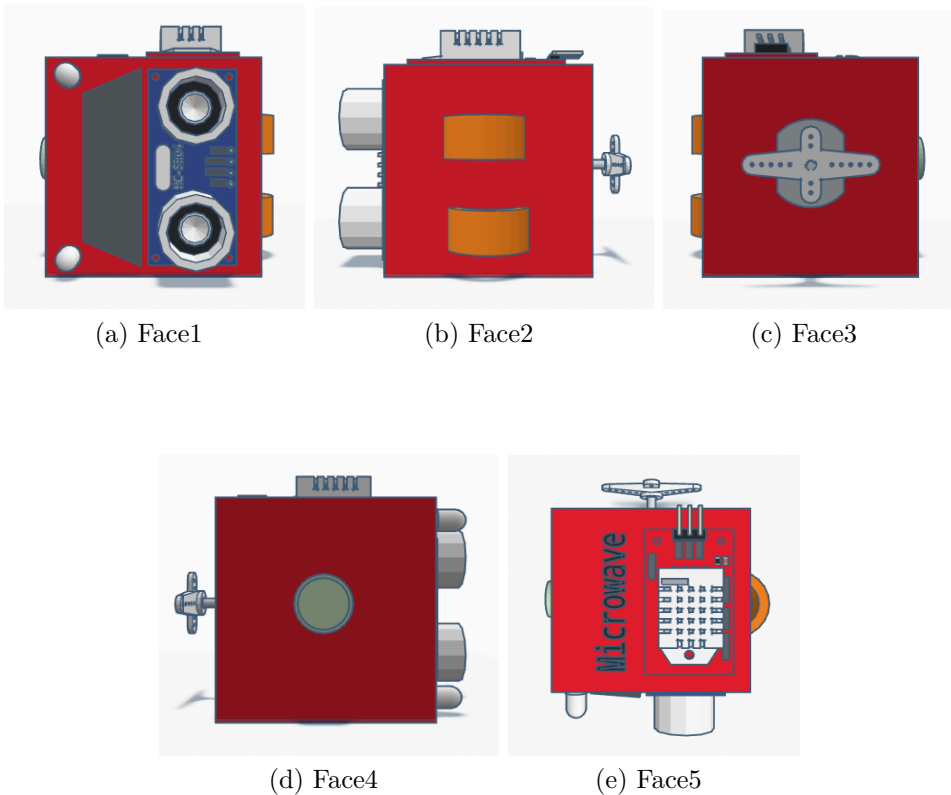


Figure 3.9: Motion cube faces

- Emotions cube

This cube was supposed to have the sensors: microphone and heart rate; and the actuators: OLED screen and vibration engine. With these devices (plus the ESP32) and following the prices I described in the previous section, the price of this cube would be around 17,47€.

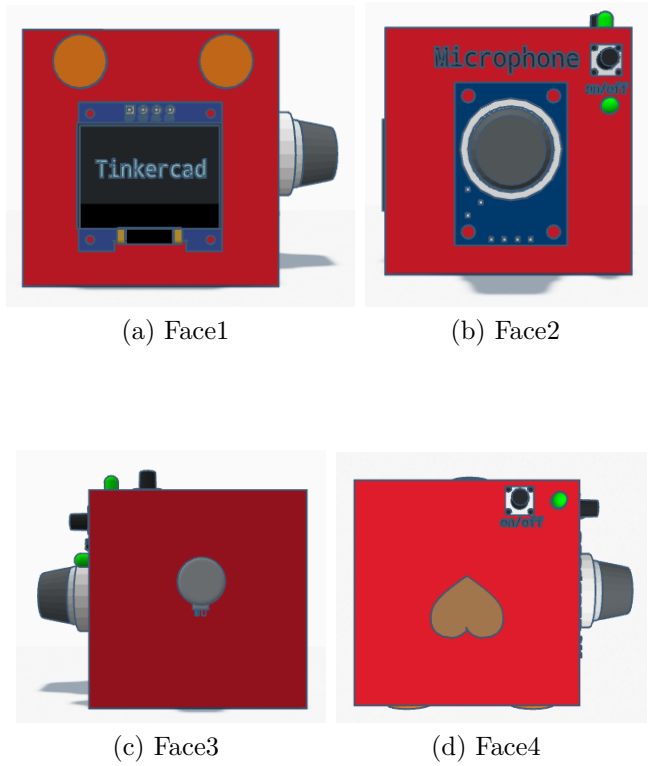


Figure 3.10: Emotions cube faces

- Lights cube

This cube was supposed to have the sensors: accelerometer and temperature; and the actuators: LEDs and lasers. With these devices (plus the ESP32) and following the prices I described in the previous section, the price of this cube would be around 17,2€.

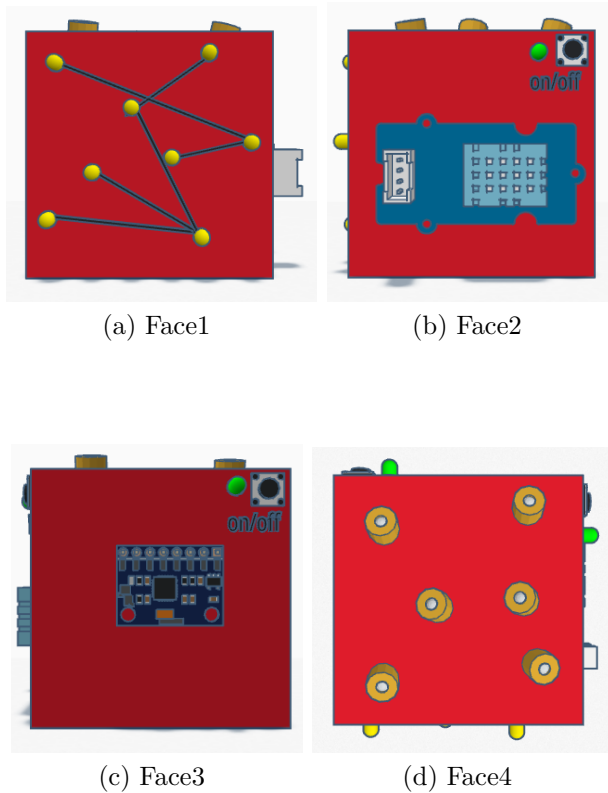


Figure 3.11: Lights cube faces

- Sound cube

Its first version had the sensors: LDR and the tilt; and the actuators: 4 speakers. With these devices (plus the ESP32) and following the prices I described in the previous section, the price of this cube would be around 13,9€.

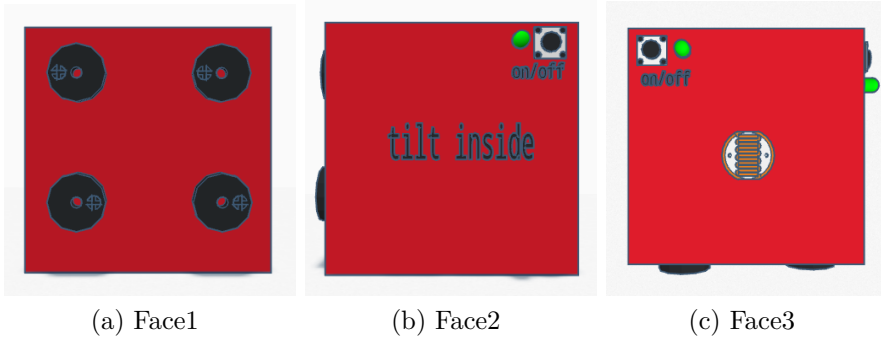


Figure 3.12: Sound cube faces

3.2.2 Final version

What it was compulsory to decide were the sensors and actuators that the cubes would finally have. These ones were in the final result because we find it funnier for the kids and easy to manage for them too.

| | Sensors | | | | | | | | | | |
|------------------|---------|-----|---------------|-----|-------|------|------------|------------|--------|------------|--------|
| | Speaker | LDR | Accelerometer | LED | Laser | OLED | Microphone | Heart beat | Wheels | Ultrasonic | Button |
| Sound - red | | | | | | | | | | | |
| Lights - blue | | | | | | | | | | | |
| Motion - yellow | | | | | | | | | | | |
| Emotions - green | | | | | | | | | | | |

Figure 3.13: Sensors and actuators distribution's table

Also, the colors of the cubes have a relation. Sound is red because sound is vibration, it's strength; emotion is green because this color symbolizes peace and relax, balance; motion is yellow because it is a happy color, it indeed means movement; and finally, light is blue because when fire is very hot, it is not red, it's blue, moreover, warm stars are blue while cold ones are red.

Finally, the other compulsory part were the tokens, which ones were needed. In the following table is possible to see their distribution.

| RED CUBE | BLUE CUBE | YELLOW CUBE | GREEN CUBE |
|----------------|-------------|-------------|-------------|
| Call blue | Call red | Call blue | Call blue |
| Call yellow | Call yellow | Call red | Call yellow |
| Call green | Call green | Call green | Call red |
| On call | On call x2 | On call | On call |
| Happy Birthday | orange LED | Forward | Happy face |
| Harry Potter | violet LED | Backwards | Sad face |
| | cyan LED | | |

Figure 3.14: Tokens distribution's table

And in the following tables is possible to see which tension divider is associated to each token.

| RED CUBE (ON CALL speaker & CALL ldr) | | |
|---------------------------------------|-------------|------------|
| R1/R2 | 100 | 220 |
| 10 | CALL BLUE | ON CALL HP |
| 100 | CALL GREEN | ON CALL HB |
| 220 | CALL YELLOW | |

Figure 3.15: Red's tokens associated resistors

| BLUE CUBE (ON CALL led,laser & CALL accelerometer) | | |
|--|-------------|--------------------|
| R1/R2 | 100 | 330 |
| 10 | CALL RED | ON CALL orange led |
| 100 | CALL GREEN | ON CALL cyan led |
| 220 | CALL YELLOW | ON CALL violet led |
| 330 | | ON CALL laser |

Figure 3.16: Blue's tokens associated resistors

| GREEN CUBE (ON CALL oled & CALL microphone) | | |
|---|-------------|---------------|
| R1/R2 | 100 | 220 |
| 10 | CALL BLUE | ON CALL happy |
| 100 | CALL RED | ON CALL sad |
| 220 | CALL YELLOW | |

Figure 3.17: Green's tokens associated resistors

| YELLOW CUBE (ON CALL wheels & CALL button,ultrasonic) | | |
|---|-------------|-------------------|
| R1/R2 | 100 x2 | 330 |
| 10 | CALL RED | ON CALL forward |
| 100 | CALL GREEN | ON CALL backwards |
| 220 | CALL YELLOW | |

Figure 3.18: Yellow's tokens associated resistors

Design

As I have said before, the idea was to design them in 3D and print them. However, it was easier to use the laser cut and doing them in wood because it was going to be easier to test prototypes and make faster models. While printing this cube would take around 3 hours in a 3D printer, in the laser cut it takes 5 minutes; so if there is any mistake it is much easier to solve it.

Final result

- Lights cube

With this final result, the price of this cube would be around 10.38€ just in electronics.

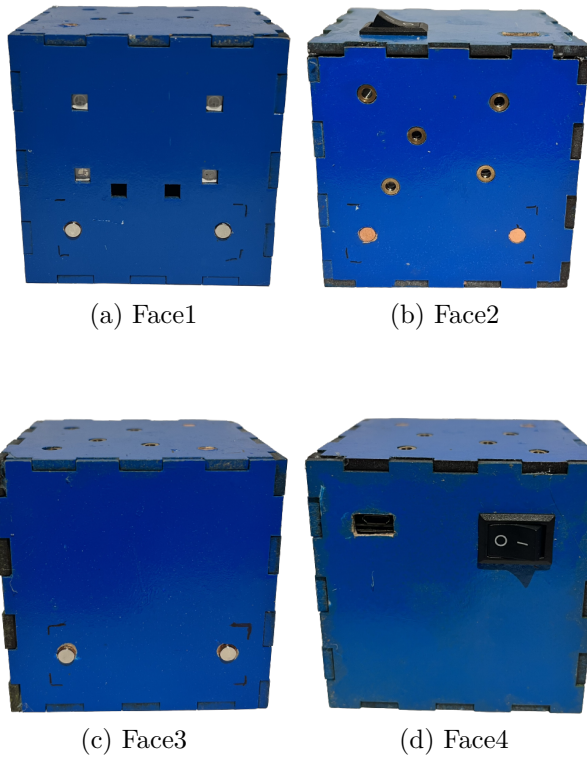


Figure 3.19: Lights cube faces

- Emotions cube

The price of this cube would be around 9.98€ just in electronics.

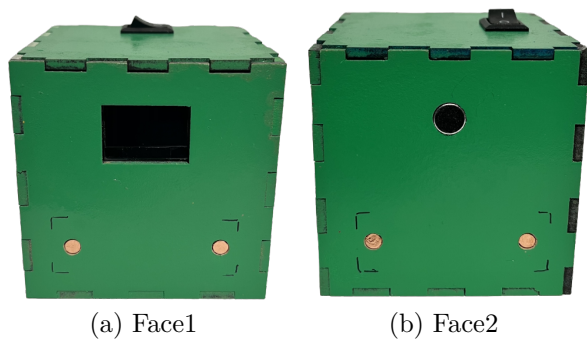
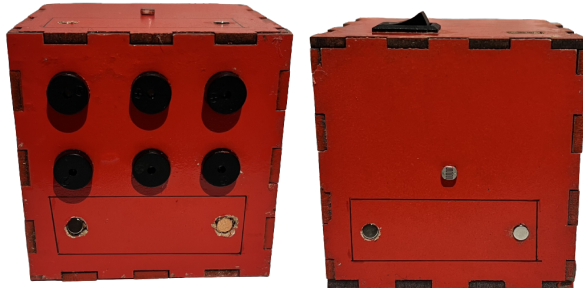


Figure 3.20: Emotions cube faces

- Sounds cube

The price of this cube would be around 13.8€ just in electronics.



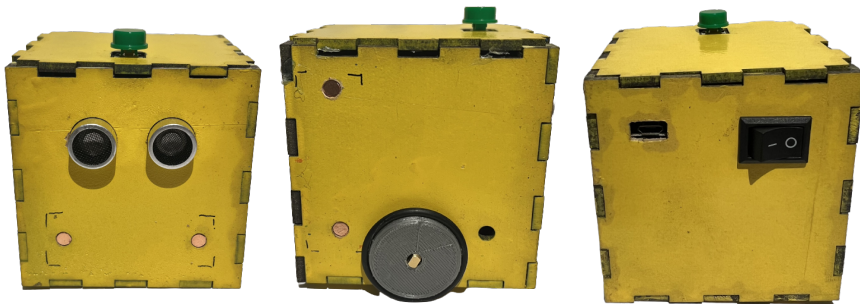
(a) Face1

(b) Face2

Figure 3.21: Sounds cube faces

- Motion cube

The price of this cube would be around 7.48€ just in electronics.



(a) Face1

(b) Face2

(c) Face3

Figure 3.22: Motion cube faces

- Tokens



(a) Tokens

Figure 3.23: Cubes tokens

Chapter 4

Tests and Results

To conclude the investigations within this project, the last step was to test the cubes with real-life users called children. However, before test with them, all the cubes were tested by me and ten more other adults. In this section I will explain what I have observed and the feedback given by the real-users followed by the improvements that have been made and if with them, other users find it easier to use the cubes.

There were no quizzes or exercises to try this because, the aim of the cubes is to learn playing and this requires the development of imagination and the freedom to decide. It may seem as lack of testing quality but, in my opinion, children can't be tested with written-paper exercises; the best is to let them be children: play and experiment by themselves.

4.1 Adults tests

After assembling all the cubes and create the tokens, it was seen that the contact surface of the magnets was too small to make an easy contact. The ideas to improve this can be read in the conclusions where I discuss the future lines.

Despite this, the adults who tested them, found it easy to understand the concept and knew how to use the cubes from the very first time (obviously after explaining them how they work). The only drawback they found was the small magnets: code works perfectly but, if magnets were bigger, it would be easier to use and code would be better appreciate.

With adults I did time them with a couple of little exercises just because, and these are facts, adults have less imagination percentage than children and they prefer being told what to do. Just because of that I did so with the ten of them. Results were very similar, they only vary in few seconds. The main seconds difference was because the tokens sometimes didn't work as supposed as I have said before, due to the non-straight surface.

4.2 Children tests

The first time I tested with a child I discovered a few things. The child was 8 years-old and it was a bit difficult for him to understand the game, that lead me to think that maybe is for older children.

Therefore, the most difficult part for him, was to understand where the CALL and ON CALL tokens should go, that's why I modify a bit the cubes writing in the faces the orders that can go. In the following picture is possible to see an example of how I write so the following children could understand it better:

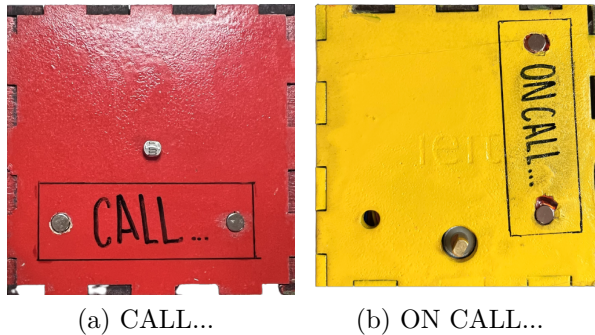


Figure 4.1: Cubes changes

After showing him many examples, he started to know how the cubes worked but, even though, it was difficult. At the end, as to the other children, he was asked if he would like a game like this; he wasn't convinced.

After this little boy, I tried with seven other children and just writing those messages in the faces of the cubes made a great difference. It's true that with 8-ish years-old children was quite similar than with the first "lab-rat": for them it's difficult to understand how they worked and what to do; even more taking into account that tokens are in English and many didn't understand it. But that was not a big deal.

Four out of those seven children were around 10 years-old and the difference was clear. They understood everything almost from the beginning and even liked to play with them. As with the first one, at the end they were asked if they would like to have toys like these and the answer was that maybe yes but... with easier-to-stick magnets.

What I found difficult was to explain the children how everything worked in an easy way. I tried not to use the real sensor's names. Instead of calling them: accelerometer, ultrasonic distance or LDR; I called them: position, distance and light sensor; but even though... was difficult for them to understand everything. Maybe having the aid of an education student would have been useful.

Chapter 5

Conclusion

This project has been a clear example of the different steps a project has: from earlier investigations and comparisons with other projects, to integrate hardware and software and finishing testing with humans and getting feedback.

Moreover, about the tokens, they were printed in 3D which, and sometimes, doesn't give the expected results, as was the case of the blue tokens. The printing wasn't straight enough so the surface was bent and the neodymium magnets weren't in a straight contact either as it can be seen in the final result. If the magnets were bigger, it would have been easier for everyone to manage them. And not just the way of printing them or the size of the magnet, but also the way to weld them to the resistors, it's very difficult because neodymium demagnetise really fast so maybe it should be good to find another way to do so.

This game should be for elder children who like computer games or so because, if not, they don't find it funny or difficult to understand. My guesses are for children around 11 years-old.

References

- [1] R. CARTER, *What is a piezo?*
<https://blog.piezo.com/what-is-a-piezo>, 2020.
- [2] O. ENGINEERING, *Accelerometer sensor*.
<https://www.omega.com/en-us/resources/accelerometers>,
2022.
- [3] T. HAREENDRAN, *Heart rate sensor*.
<http://precog.iiitd.edu.in/people/anupama>, 2022.
- [4] K. KAUR, *Microphone - sound sensor*.
<https://www.azosensors.com/Article.aspx?ArticleID=229>,
2013.
- [5] MAKERONI, *The ifs*. <https://theifs.cc/kickstarter-es/>, 2018.
- [6] C. PAOLINI, *Eragon*, Alfred A. Knopf, United States, 2003.
- [7] ROBOCRACE, *How proximity sensor works?*
<https://robocraze.com/blogs/post/how-proximity-sensor-works>,
2022.

Appendix A

ESP32 Communication Code

The following code shows the basic code for the communication protocol: ESP Now; the one used for the project.

```
#include <esp_now.h>
#include <Wire.h>
#include <WiFi.h>

// MACs of the other 3 cubes to which data will be sent
uint8_t broadcastAddress1[] = {0x7C, 0x9E, 0xBD, 0x37, 0x1D, 0x74};
uint8_t broadcastAddress2[] = {0x84, 0xCC, 0xA8, 0x54, 0xB5, 0x38};
uint8_t broadcastAddress3[] = {0x84, 0xCC, 0xA8, 0x5B, 0x80, 0x00};
uint8_t broadcastAddress4[] = {0x84, 0xCC, 0xA8, 0x53, 0x72, 0xCC};

//Structure to send data
typedef struct struct_message {
    String cube;
    int value;
} struct_message;

// Create a struct_message called BME280Readings to hold sensor readings
struct_message BME280Readings;
// Create a struct_message to hold incoming sensor readings
struct_message incomingReadings;
```

```

esp_now_peer_info_t peerInfo;
esp_now_peer_info_t peerInfo2;
esp_now_peer_info_t peerInfo3;

void setup() {
    // Get ESP32 MAC
    WiFi.mode(WIFI_MODE_STA);
    Serial.println(WiFi.macAddress());

    // Set device as a Wi-Fi Station
    WiFi.mode(WIFI_STA);

    // Init ESP-NOW
    if (esp_now_init() != ESP_OK) {
        Serial.println("Error initializing ESP-NOW");
        return;
    }

    // Register the ESPNow
    esp_now_register_send_cb(OnDataSent);

    // Register peers
    memcpy(peerInfo.peer_addr, broadcastAddress2, 6);
    peerInfo.channel = 0;
    peerInfo.encrypt = false;

    memcpy(peerInfo2.peer_addr, broadcastAddress3, 6);
    peerInfo2.channel = 0;
    peerInfo2.encrypt = false;

    memcpy(peerInfo3.peer_addr, broadcastAddress4, 6);
    peerInfo2.channel = 0;
    peerInfo2.encrypt = false;

    // Add peers
    if (esp_now_add_peer(&peerInfo) != ESP_OK){
        Serial.println("Failed to add peer 1");
        return;
    }
}

```

```

    if (esp_now_add_peer(&peerInfo2) != ESP_OK){
    Serial.println("Failed to add peer 2");
    return;
    }

    if (esp_now_add_peer(&peerInfo3) != ESP_OK){
    Serial.println("Failed to add peer 3");
    return;
    }

    // Register for a callback called when data is received
    esp_now_register_recv_cb(OnDataRecv);
}

void loop() {
    value = 10;
    sendValues(value, BME280Readings, broadcastAddressN);
}

void sendValues(int sensor_value, struct_message BME280Readings,
                uint8_t broadcastAddress[]){
    // 1. Print read values
    Serial.print("Value of the sensor: ");
    Serial.println(sensor_value);

    // 2. Send the values
    BME280Readings.cube = "RED";
    BME280Readings.value = sensor_value;
    esp_err_t result = esp_now_send(broadcastAddress, (uint8_t *) &BME280Rea

    // 3. Test if everything is fine
    if (result == ESP_OK) {
        Serial.println("Sent with success");
    }
    else {
        Serial.println("Error sending the data");
    }
}
}

```
