Professional Education Center Hélio Augusto de Souza

Autonomous Room

Home Automation

Professional Education Center Hélio Augusto de Souza
Technical Course in Electronics
Night Shift - Module III
Curricular Component: Project

Students: Camilly Gurgel dos Santos Nº02

Débora Cavalcante da Silva N°03

Julio Oscar Retamales Nº08

Lucineide Maria de Brito Nº 09

Wanderson Augusto da Cunhas Santos Nº14

Advisor: Antônio Carlos Cintra Pereira

São José dos Campos - SP

2023

Camilly Gurgel dos Santos
Débora Cavalcante da Silva
Julio Oscar Retamales
Lucineide Maria de Brito
Wanderson Augusto da Cunha Santos

Autonomous Room

Automation Arduino + Bluetooth

Project or Thesis presented as a partial requirement for obtaining the title of Electronics Technician at the Hélio Augusto de Souza Professional Training Center in São José dos Campos.

São José dos Campos - SP

2023

Acknowledgments

We dedicate this work to everyone who supported us on our challenging journey, balancing work and study, facing sleepless nights and constant challenges. Every obstacle overcome was an essential part of this trajectory.

We also especially dedicate it to our advisor Antônio Carlos Cintra, who helped us in the development of our project.

I, Julio, dedicate this work especially to my mother, a hardworking woman who taught me that difficult moments will always exist, but it is precisely those moments that make us stronger to move forward. For without obstacles along the way, there is no glory in reaching the destination.

"Be sincere and deeply desire, and you will be able to shake the world." - Raul Seixas

Abstract

This project involves the development of an automation system for a room, using **sensors, motors, and a Bluetooth module** to remotely control doors, windows, and electronic devices. The automation allows the door to open automatically when detecting a person's presence, while the window can be controlled remotely. Additionally, LEDs were used to represent household appliances, simulating a connected environment.

Communication between the user and the system occurs via **mobile phone or computer**, ensuring practicality and accessibility. Unlike internet-based systems, the use of **Bluetooth** enables implementation in areas with limited network connectivity.

The project covers everything from component selection to device programming, including practical tests to validate its functionality. The results show that automation can bring more convenience to daily life and serve as a foundation for future innovations in the field.

Keywords: Automation, Bluetooth, Sensors, Remote Control, Accessibility

Table of Contents

1. Introduction

Home automation has become an increasingly present reality in people's daily lives. With technological advancements and the popularization of smart systems, it is now possible to control various household devices remotely, efficiently, and in a personalized way. The concept of a smart home involves the use of sensors, actuators, and communication systems that automate daily tasks, providing convenience, security, and energy savings.

In recent years, automation has shifted from being a luxury limited to large corporations and high-end residences to becoming accessible to a broader audience. Solutions such as lighting control, automated opening and closing of doors and windows, environment monitoring, and remote activation of household appliances are becoming increasingly common in the market. This growth is driven by the demand for greater convenience and the advancement of technologies such as microcontrollers, sensors, and wireless communication.

This project proposes the development of an automated room using an Arduino Mega, stepper motors, sensors, and a Bluetooth module. The system enables the remote control of doors, windows, and simulated devices through LEDs, creating an interactive and functional environment. Unlike solutions that rely on an internet connection, the use of Bluetooth allows the automation to function in areas with limited infrastructure, making it a viable alternative for various scenarios.

1.1. Objective

The main objective of this project is to **demonstrate, in a didactic and accessible way, the operation of an automated system** and its possible applications in daily life. We aim to present **home automation as a practical and functional solution**, highlighting its viability in different contexts and encouraging its adoption in new projects.

1. **Specific Objectives**
    i. **Develop an automation system** using **Arduino Mega, sensors, and motors**, exploring fundamental concepts of **electronics and programming**.
    ii. **Create a functional and interactive system**, allowing **remote control of doors, windows, and devices** through a **Bluetooth module**.
    iii. **Provide an introductory resource** for students and technology enthusiasts, covering **automation implementation with Arduino and C++**.

1.2. Justification

The increasing demand for **technological solutions that simplify daily tasks** has driven the development of **home automation**. However, many of these technologies remain **underexplored by students and small developers**, either due to a lack of knowledge or access to **educational materials**.

This project aims to **expand understanding of automation**, demonstrating how **simple systems** can be **implemented practically and efficiently**. By using **accessible components** such as **Arduino and Bluetooth modules**, we seek to **democratize this knowledge**, encouraging the creation of **new applications and solutions** aimed at **improving quality of life and technological accessibility**.

Furthermore, the project seeks to **enable more people to work in this field**, fostering the development of **autonomous devices** that facilitate daily life.

2. Materials

A **scaled-down demonstration model** of a room was created, featuring **automated windows and doors** that move independently without the need for human intervention. These devices are equipped with **perception, decision-making, and control systems**, allowing autonomous interaction with individuals. The theoretical foundation of this automated room involves **various fundamental concepts and technologies**, which are discussed in the following sections.

We chose to **demonstrate only a single room** instead of a complete environment due to the **ease of prototyping and presentation** of our model.

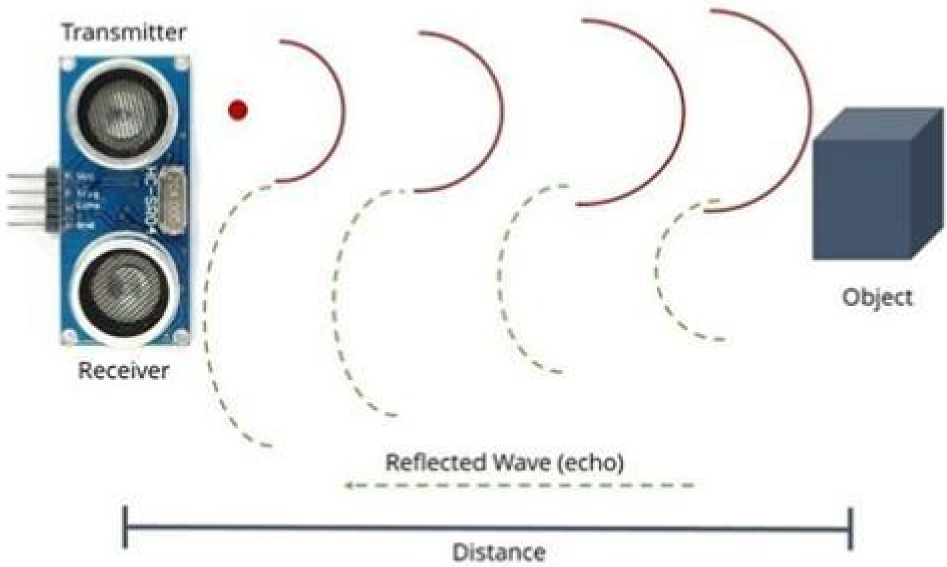
2.1. Sensors

Sensors are essential components of the autonomous model, as they provide information about the external environment and the presence of individuals.

2.1.1. Sensors Ultrasonic

We use ultrasonic sensors, which operate by measuring the time it takes for an emitted sound pulse to travel and return as an echo. The sensors emit ultrasound pulses cyclically. When an object reflects these pulses, the resulting echo is received and converted into an electrical signal.

This sensor is the primary component for analyzing the area near the door, determining when it should open, remain open, or close.

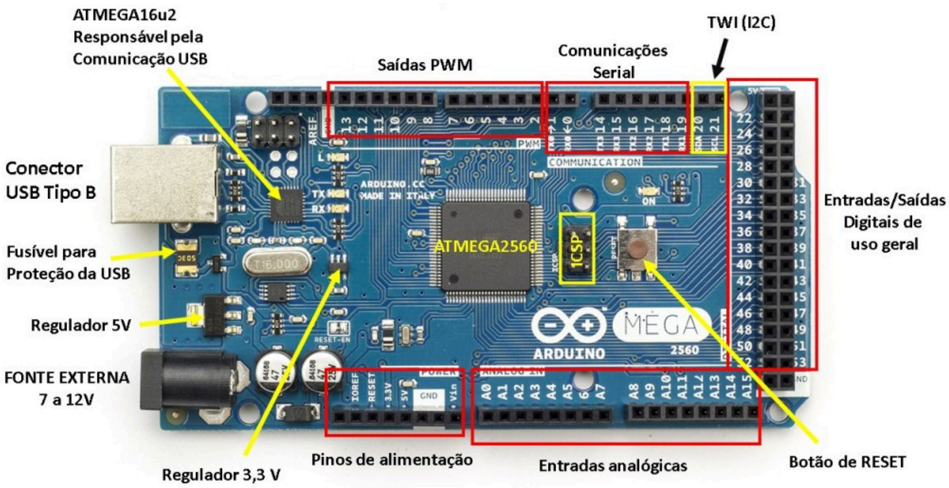


2.1.2. Data Processing

After the **sensor data is collected**, it must be processed to allow the **automatic door** to make decisions. **Data processing** involves steps such as **filtering, sensor fusion, and extracting relevant characteristics**. Machine learning algorithms and computer vision techniques are widely used to **process sensor data** and extract useful information.

2.2. Arduino

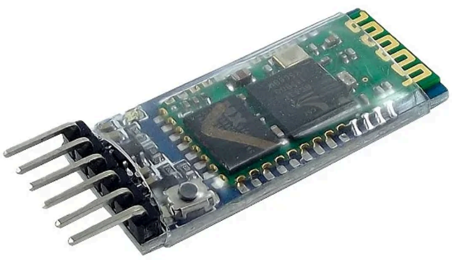
We use **Arduino** for data processing because it is an **open-source electronic prototyping platform** with **free hardware**. It is **easy to use and program**, allowing us to **store all necessary data** for analyzing the **automated door and window** operations.



Fonte:portal.vidadesilicio.com.br

Modulo Bluetooth hc-05

O The Bluetooth HC-05 module is one of the key components of the system, responsible for enabling wireless communication between the Arduino Mega and an external device, such as a smartphone or computer. It operates as a serial communication bridge, receiving commands sent by the user and transmitting them to the microcontroller for execution.



In the automated room project, the Bluetooth module allows for the remote control of the door, window, and LEDs through an application or serial terminal. This enables the user to send specific commands to open or close the window, activate the door, or simulate the operation of electronic devices represented by LEDs.

The choice of Bluetooth over an internet-based connection was made to ensure that the system functions even in areas without network access, making it an accessible and functional solution for various environments

.

2.3. Decision-Making

Based on the collected and processed data, the automatic door must make decisions regarding its movement. Path planning algorithms are used to determine the most suitable route, taking into account speed limitations, directional rules, and other relevant factors.

Additionally, the decision-making systems must also consider safety aspects, prioritizing actions that minimize risks while ensuring smooth and efficient automation.

2.3.1. C++

We use **C++** because it is highly **flexible** for **algorithm development** and integrates directly with the **Arduino IDE**. This allows the **robot** to operate based on the **pre-programmed algorithm**, ensuring that it makes **decisions** according to the system's logic..

2.4. Controls

2.4.1. Motor Linear

A **linear motor** was implemented in the **door and window automation system**. This motor efficiently **converts direct current (DC) electrical energy into mechanical movement**, making it the preferred choice due to its **precise control and reversibility**.

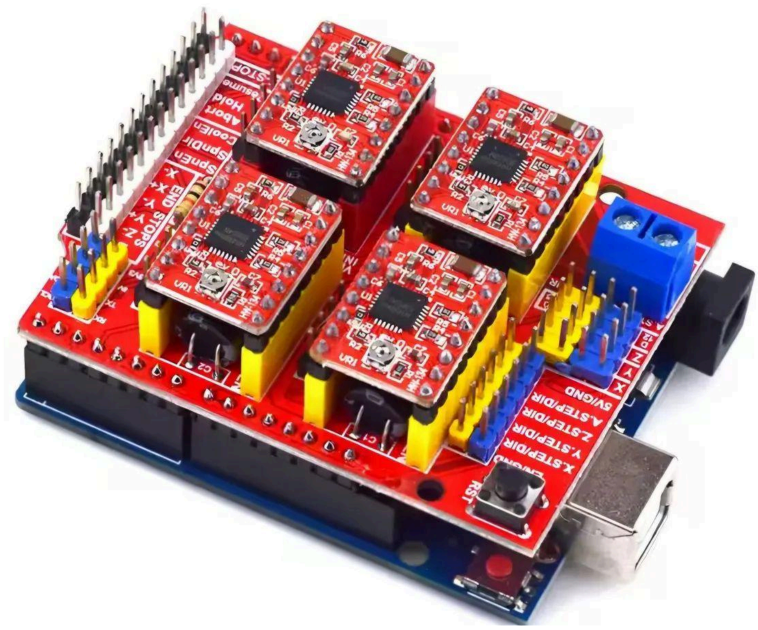
Widely used in **home automation**, the **linear motor** provides **energy efficiency and fast response**, making it ideal for **systems that require smooth and bidirectional movements**.

Fonte:https://www.amazon.com.br/lineares-acabamento-deslizante-equipamentos-desl izantes/dp/B097X521CQ

2.4.2. CNC Shield

An **extension of the Arduino** for **motor control**, the **CNC Shield** improves **circuit management** by receiving information from the **Arduino** and distributing it **more precisely** to each motor.

It is also highly efficient because it allows **multiple components** to work together, following the programmed instructions sent from the **Arduino**. Examples include **ultrasonic sensors and servo motors**, which can be integrated into the system for enhanced automation.



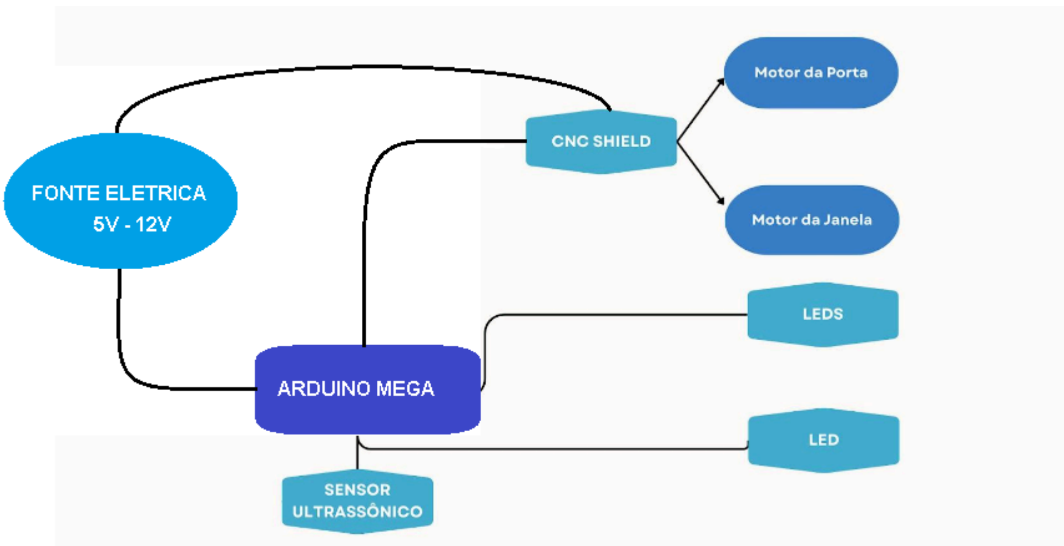
Fonte:arduinoecia.com.br

3. Methods and Procedures

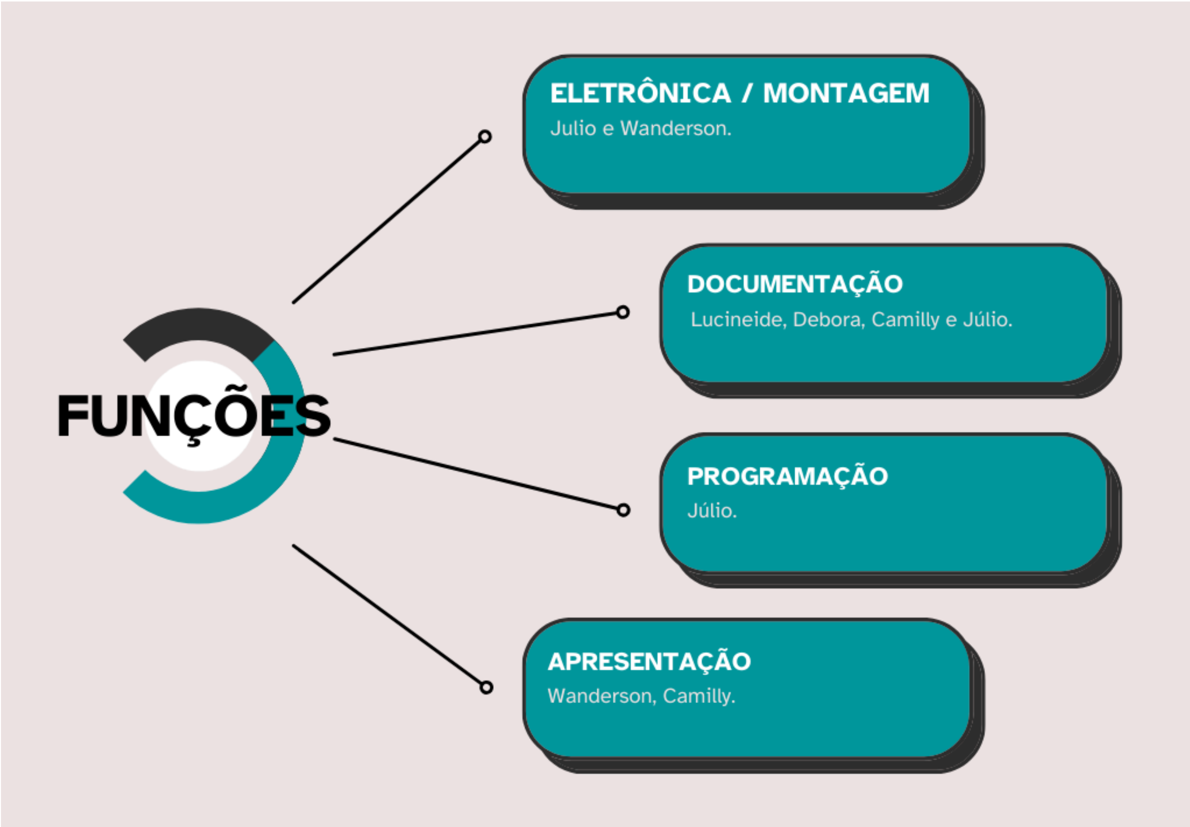
3.1. Schedule

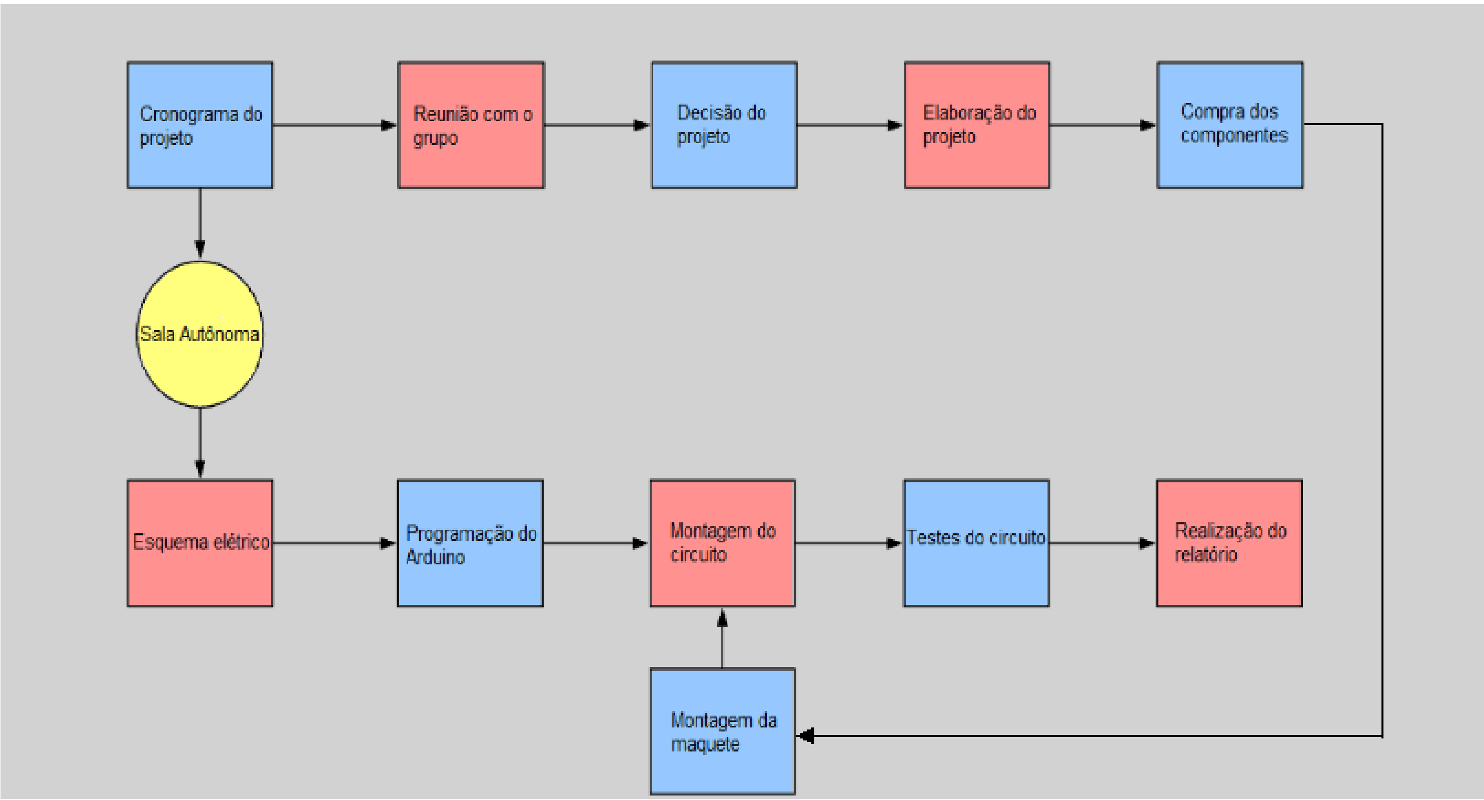
Materias / Custos	Produção da programação	Conceito de Montagem	Processo de montagem	Testes/ Finalizações
4 Semanas	4 Semanas	2 Semanas	2 Semana	2 Semanas

3.2. Block Diagrams



3.3. Task Division





4. Development

4.1. Assembly Description

To begin the development phase, we first worked on the **programming code** that would be used in the system. We inserted **serial monitor commands** for each component in the model, with special attention to the **CNC Shield commands**, testing them in simulators such as **Tinkercad**.

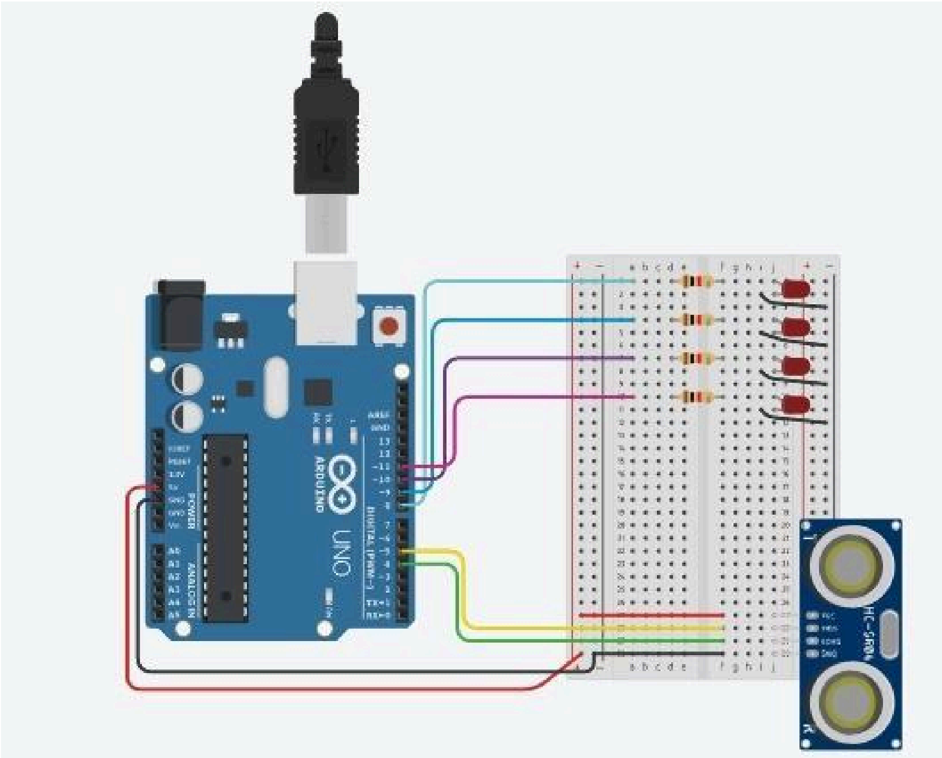
For connecting the **Linear Motors** to the **Shield**, it was necessary to **solder the wires** (this process was repeated for both the **door and the window** motors).

Next, each **linear motor** was placed in its respective location, along with the **structural components** representing the **door and window**. The door was additionally equipped with an **ultrasonic sensor**, which detects the presence of an individual within a defined range, **automatically triggering the door's opening**.

After completing these steps, we carefully mounted and positioned **each component** in its designated location. **Screws** were used to secure the **linear motors**, ensuring precise movement for the **door and window**. The **model was drilled** to allow proper placement of the **Ultrasonic Sensor** next to the linear motor for the **door**, and all necessary connections to the **Shield** were made.

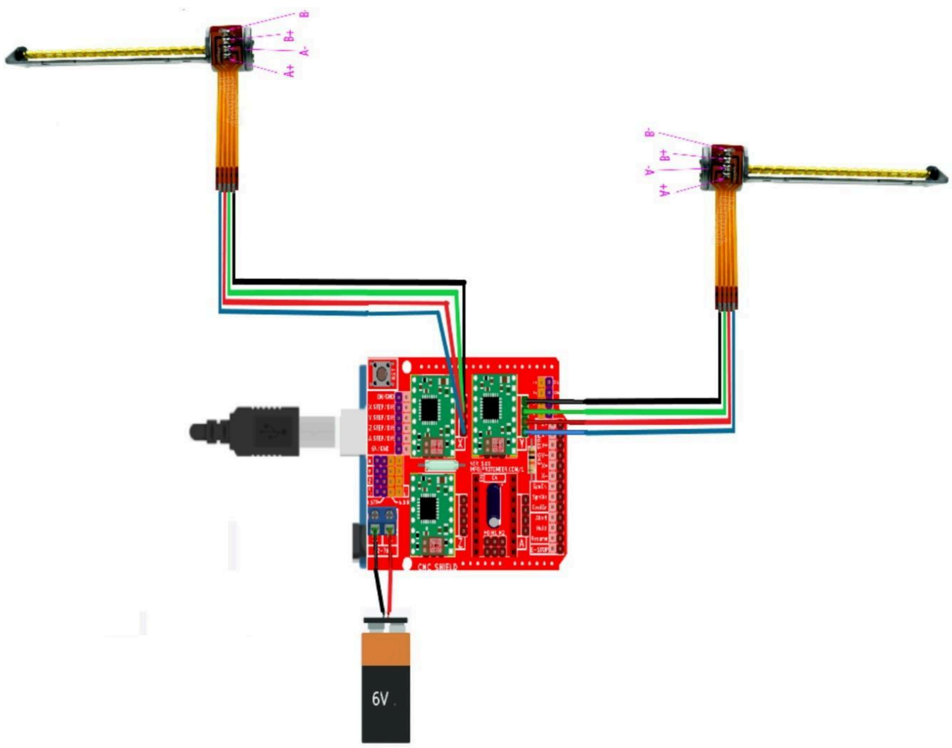
Once this phase was completed, the **Arduino Mega** was placed at the **center of the model**, with the **Shield positioned on top** for optimal integration.

As a final step, we integrated the **model and programming code** with additional **LEDs**, which were included to represent the **Ultrasonic Sensor operation, lighting, coffee maker, and other devices** that could be activated in this **automation process**.



Next, we connected the **USB cable** to the **Arduino** and linked it to the computer to **transfer the programmed code**. The code was developed using **C++ programming**.

Finally, to provide the necessary **power supply for the motors**, we used a **12V power source** along with a **voltage regulator**, allowing us to **control it at 6V**, which is the optimal voltage for the **proper operation of the linear motors**.



In the representation above, **6V batteries** were used to **simulate the 12V power source with the voltage regulator**.

4.2. Code

To access the code, visit:: https://github.com/Xulio1337/TCC_Tec-Eletronico

```
const int enPin = 8;
const int stepXPin = 2;  // X.STEP
const int dirXPin = 5;   // X.DIR
const int stepYPin = 3;  // Y.STEP
const int dirYPin = 6;   // Y.DIR

int pulseWidthMicros = 100;  // micro segundos
int millisBtwnSteps = 1000;

const int openDuration = 300;
const int pinLedA = 31; //lampada do quarto
const int pinLedB = 33; //cafeteira
const int pinLedC = 35; //radio
const int pinLedD = 39; //Luz do sensor
const int pinTrig = 24;
const int pinEcho = 26; //branco

const int Media_Interactions = 10;
const int Distance_Treshold = 10;
bool has_AutoGate = false;
String Serial_Data = "";
String Serial1_Data = ""; //Esse segundo Serial_Data serve para exibir
// a imagem no celular atraves do modulo Bluetooth

void setup()
{
Serial.begin(9600);
Serial.setTimeout(10);
Serial.flush();
Serial.println("Serial USB Inicializada.");

Serial1.begin(9600);
Serial1.setTimeout(10);
Serial1.flush();
Serial.println("Serial 1 Inicializada.");
Serial1.println("Serial 1 Inicializada.");
pinMode(enPin, OUTPUT);
digitalWrite(enPin, LOW);
pinMode(stepXPin, OUTPUT);
pinMode(dirXPin, OUTPUT);
pinMode(stepYPin, OUTPUT);
pinMode(dirYPin, OUTPUT);
pinMode(pinLedA, OUTPUT);
pinMode(pinLedB, OUTPUT);
pinMode(pinLedC, OUTPUT);
pinMode(pinLedD, OUTPUT);
pinMode(pinTrig, OUTPUT);
pinMode(pinEcho, INPUT);
Serial.println(F("CNC Shield iniciada"));
}
```

Code Overview

The implemented code automates a room using an Arduino Mega, a CNC shield, a Bluetooth module, two stepper motors, an ultrasonic sensor, and several LEDs. It allows remote control of doors, windows, and electronic devices through Bluetooth commands or the computer's serial terminal.

1. Code Structure

The code is divided into key sections:

Variable Declaration and Pin Definitions: Assigns connections for motors, LEDs, and sensors.

Setup: Initializes serial communication, configures input and output pins, and displays initial messages.

Main Loop: Continuously reads serial commands and checks the ultrasonic sensor.

Auxiliary Functions: Manage motor control, LED activation, and command processing.

2. System Initialization

In the setup(), the code configures pins and initializes serial ports:

Serial.begin(9600) and Serial1.begin(9600): Start serial communication for the computer and Bluetooth, respectively.

pinMode(): Configures motor, LED, and sensor pins as input or output.

digitalWrite(enPin, LOW): Enables motors by deactivating the enable pin.

3. Communication and Command Processing

The code has two functions to read commands:

Read_Serial(): Reads commands sent via USB serial.

Read_Serial1(): Reads commands sent via Bluetooth.

Both functions store incoming data in strings (Serial_Data and Serial1_Data) until a newline character \n is detected, triggering command processing via Split_String() and Split_String_1().

Once received, the command is interpreted and executed by the Switch_Action(String command) function.

4. Motor Control

Stepper motors are controlled by two functions:

moveMotor(int stepPin, int dirPin, int duration): Moves the motor in the desired direction for a specified time.

reverseMoveMotor(int stepPin, int dirPin, int duration): Moves the motor in the opposite direction.

Each motor operates for a duration defined by the openDuration constant.

Numeric commands (1 to 8) control motor movement, for example:

- 1: Opens the door.
- 2: Closes the door.
- 4: Opens the window.
- 5: Closes the window.
- 7: Opens both the door and window simultaneously.
- 8: Closes both the door and window simultaneously.

5. LED Control

LEDs simulate electronic devices such as a lamp, coffee maker, and radio. They are controlled by the following commands:

a: Toggles the bedroom lamp.

b: Toggles the coffee maker.

c: Toggles the radio.

6. Automation via Ultrasonic Sensor

The distancia() function reads data from the ultrasonic sensor to detect movement. If an object is detected within 5 cm, the door automatically opens, waits a few seconds, and then closes.

If the distance is less than 5 cm, the door opens and closes after 3 seconds.

7. Combined Command Execution

The system also allows executing multiple actions simultaneously using special commands:

N: Closes the door and window, and turns off all LEDs**.**

D: Opens the door and window, activates LEDs, and executes a programmed sequence.

Conclusion

The code successfully implements a smart automation system, enabling remote control of doors, windows, and electronic devices via Bluetooth or computer commands. It detects presence and interacts with the environment automatically, making it a viable automation solution for locations without internet access.

```
pinMode(pinEcho, INPUT);
Serial.println(F("CNC Shield iniciada"));
}
void loop()
{

Read_Serial();
Read_Serial1();
distancia();
}
 void Read_Serial()
 {
 if (Serial.available() > 0)
 {
 int inChar = Serial.read();
 if (inChar != '\n')
 {
 Serial_Data += (char)inChar;
 }
 else
 {
 Split_String();
 }
 }
 }
 void Read_Serial1()
 {
 if (Serial1.available() > 0)
 {
 int inChar = Serial1.read();
 if (inChar != '\n')
 {
 Serial1_Data += (char)inChar;
 }
 else
 {
 Split_String_1();
```

```
}
void Split_String()
{
int CommaIndex = 0;
for (int i = 0; i < Serial_Data.length(); i++)
{
if (Serial_Data[i] == ',')
{
CommaIndex = i;
}
}
int  State  =  (Serial_Data.substring(Serial_Data.length()  -  2,
Serial_Data.length())).toInt();
int Port = (Serial_Data.substring(1, CommaIndex)).toInt();
Switch_Action(Serial_Data);
Serial_Data = "";
}
void Split_String_1()
{
Switch_Action(Serial1_Data);
//Serial.print("===========> Serial BT DATA ");
//Serial.println(Serial1_Data);
Serial1_Data = "";
}
void moveMotor(int stepPin, int dirPin, int duration)
{
unsigned long startTime = millis();

while (millis() - startTime < duration)
{

digitalWrite(dirPin, HIGH);
digitalWrite(stepPin, HIGH);
delayMicroseconds(pulseWidthMicros);
digitalWrite(stepPin, LOW);
delayMicroseconds(millisBtwnSteps);
```

```
 }
void reverseMoveMotor(int stepPin, int dirPin, int duration)
{

unsigned long startTime = millis();
while (millis() - startTime < duration)
{

digitalWrite(dirPin, LOW); // Reverso
digitalWrite(stepPin, HIGH);
delayMicroseconds(pulseWidthMicros);
digitalWrite(stepPin, LOW);
delayMicroseconds(millisBtwnSteps);
}
 }
void Switch_Action(String command)
{
char Value[20];
command.toCharArray(Value, 20);
Serial.println(Value[0]);
switch (Value[0])

{
case '1':
Serial.println(F("Opening Motor X"));
Serial1.println(F("Opening Motor X"));
moveMotor(stepXPin, dirXPin, openDuration);
Serial.println(F("Motor X Stopped"));
Serial1.println(F("Motor X Stopped"));
break;
case '2':
Serial.println(F("Closing Motor X"));
Serial1.println(F("Closing Motor X"));
reverseMoveMotor(stepXPin, dirXPin, openDuration);
Serial.println(F("Motor X Stopped"));
Serial1.println(F("Motor X Stopped"));
break;
case '4':
Serial.println(F("Opening Motor Y"));
Serial1.println(F("Opening Motor Y"));
moveMotor(stepYPin, dirYPin, openDuration);
Serial.println(F("Motor Y Stopped"));
Serial1.println(F("Motor Y Stopped"));
break;
case '5':
Serial.println(F("Closing Motor Y"));
Serial1.println(F("Closing Motor Y"));
reverseMoveMotor(stepYPin, dirYPin, openDuration);
Serial.println(F("Motor Y Stopped"));
Serial1.println(F("Motor Y Stopped"));

break;
case '7':
Serial.println(F("Simultaneously Opening Motors X and Y"));
Serial1.println(F("Simultaneously Opening Motors X and Y"));
moveMotor(stepXPin, dirXPin, openDuration);
moveMotor(stepYPin, dirYPin, openDuration);
Serial.println(F("Motors X and Y Stopped"));
Serial1.println(F("Motors X and Y Stopped"));
break;
case '8':
Serial.println(F("Simultaneously Closing Motors X and Y"));
Serial1.println(F("Simultaneously Closing Motors X and Y"));
reverseMoveMotor(stepXPin, dirXPin, openDuration);
reverseMoveMotor(stepYPin, dirYPin, openDuration);
Serial.println(F("Motors X and Y Stopped"));
Serial1.println(F("Motors X and Y Stopped"));
break;
case '3':
Serial.println(F("Custom Command: Opening Motor X and Closing Motor Y"));
Serial1.println(F("Custom Command: Opening Motor X and Closing Motor Y"));
moveMotor(stepXPin, dirXPin, openDuration);
reverseMoveMotor(stepYPin, dirYPin, openDuration);
```

```
break;
case '6':
Serial.println(F("Custom Command: Opening Motor Y and Closing Motor X"));
Serial1.println(F("Custom Command: Opening Motor Y and Closing Motor X"));

moveMotor(stepYPin, dirYPin, openDuration);
reverseMoveMotor(stepXPin, dirXPin, openDuration);
Serial.println(F("Motors X and Y Stopped"));
Serial1.println(F("Motors X and Y Stopped"));
break;
case 'a':
digitalWrite(pinLedA, !digitalRead(pinLedA)); // Inverte o estado do LED
break;
case 'b':
digitalWrite(pinLedB, !digitalRead(pinLedB));
break;
case 'c':
digitalWrite(pinLedC, !digitalRead(pinLedC));
break;
case 'd':
Serial.println(F("Opening Motor X"));
Serial1.println(F("Opening Motor X"));
moveMotor(stepXPin, dirXPin, openDuration);
Serial.println(F("Motor X Stopped"));
Serial1.println(F("Motor X Stopped"));
Serial.println(F("Opening Motor Y"));
Serial1.println(F("Opening Motor Y"));
moveMotor(stepYPin, dirYPin, openDuration);
Serial.println(F("Motor Y Stopped"));
Serial1.println(F("Motor Y Stopped"));
delay(100);
digitalWrite(pinLedA, LOW);
digitalWrite(pinLedC, HIGH);

delay(100);
digitalWrite(pinLedB, HIGH);
delay(5000);
```


```
delay(100);
digitalWrite(pinLedB, HIGH);
delay(5000);
```


```
digitalWrite(pinLedC, LOW);
break;
case 'n':
Serial.println(F("Closing Motor X"));
Serial1.println(F("Closing Motor X"));
reverseMoveMotor(stepXPin, dirXPin, openDuration);
Serial.println(F("Motor X Stopped"));
Serial1.println(F("Motor X Stopped"));
Serial.println(F("Closing Motor Y"));
Serial1.println(F("Closing Motor Y"));
reverseMoveMotor(stepYPin, dirYPin, openDuration);
Serial.println(F("Motor Y Stopped"));
Serial1.println(F("Motor Y Stopped"));
digitalWrite(pinLedA, LOW);
digitalWrite(pinLedB, LOW);
digitalWrite(pinLedC, LOW);
break;
case 't':
Serial.println(F("Opening Motor X"));
Serial1.println(F("Opening Motor X"));
moveMotor(stepXPin, dirXPin, openDuration);
Serial.println(F("Motor X Stopped"));
Serial1.println(F("Motor X Stopped"));
Serial.println(F("Closing Motor Y"));
Serial1.println(F("Closing Motor Y"));

reverseMoveMotor(stepYPin, dirYPin, openDuration);
Serial.println(F("Motor Y Stopped"));
Serial1.println(F("Motor Y Stopped"));
digitalWrite(pinLedA, HIGH);
delay(100);
digitalWrite(pinLedB, LOW);
delay(100);
digitalWrite(pinLedC, HIGH);
```

```

break;
}
}
void distancia() {
digitalWrite(pinTrig, LOW);
delayMicroseconds(2);
digitalWrite(pinTrig, HIGH);
delayMicroseconds(10);
digitalWrite(pinTrig, LOW);

long duration = pulseIn(pinEcho, HIGH);
int distance = duration * 0.034 / 2;
//Aqui existia um Serial.Print para a distancia, retirado por motivos convencionais

if (distance <= 5) {

digitalWrite(pinLedD, HIGH);   // Liga o LED
reverseMoveMotor(stepXPin, dirXPin, openDuration);
Serial.println(F("Opening Motor X"));
Serial1.println(F("Opening Motor X"));
// Aguarda 3 segundos
delay(1000);
Serial.println(F("aguardando X"));
Serial1.println(F("aguardando X"));
delay(3000);
//Inverte o motor após 3 segundos
moveMotor(stepXPin, dirXPin, openDuration);
digitalWrite(pinLedD, LOW);
Serial.println(F("Closing Motor X"));
Serial1.println(F("Closing Motor X"));
// Mantém-se disponível para detecção contínua
}
// Delay opcional para evitar leituras muito frequentes
delay(1000);
}
```

5. Result and discussion

5.1. General Result

The model successfully performs the assigned functions, utilizing either a sensor or mobile command to activate the motors, enabling the opening and closing of the door and window.

6. Conclusion

The development of this automated room with Bluetooth control project demonstrated, in practice, how automation can make environments more accessible, efficient, and intelligent. Using an Arduino Mega, stepper motors, ultrasonic sensors, and a Bluetooth module, we created a functional system that allows remote control of doors, windows, and simulated devices via LEDs.

The implementation of Bluetooth as a communication method was a strategic choice, ensuring that the system functions independently of an internet connection, making it a viable solution for remote locations or areas with limited infrastructure. Additionally, the use of ultrasonic sensors to detect user presence and trigger door opening reinforces the idea of a system that minimizes the need for manual intervention, making the environment more automated and intuitive.

During development, we faced challenges related to component integration, sensor calibration, and code structuring. However, these difficulties were overcome through testing, adjustments, and system optimization, resulting in a functional project aligned with the initial objectives.

Finally, this work not only contributes to understanding home automation but also serves as a foundation for future improvements and expansions. The application of new technologies, such as Wi-Fi connectivity, artificial intelligence, and greater integration with IoT devices, can further expand the possibilities of this system, making it even more efficient and applicable to different everyday scenarios.

Referências

https://web.tecnico.ulisboa.pt/~ist186955/wordpress/2016/12/14/162/#:~:text=Um%2
0motor%20DC%20%C3%A9%20um,for%C3%A7as%20produzidas%20por%20cam
pos%20magn%C3%A9ticos.

http://www.um.pro.br/arduino/index.php?c=ArduinoUno

https://srituhobby.com/how-to-use-a-cnc-shield-step-by-step-guide-with-grbl

/ https://smallhammer.cc/product/sd1

https://www.usinainfo.com.br/sensor-ultrassonico-502