D-Door

Design Documentation

Renan Mougenot Breviglieri Silva

Senior Project - ECE 4020

Weber State University Computer & Electrical Engineering Department

Table of Contents: Page Number

Table of Figures:

Page Number

1. Introduction

This is the documentation of a senior project developed and designed by Renan Mougenot Breviglieri Silva, for the Computer and Electrical Engineering Department at Weber State University – Ogden, UT, USA. Pet owners must face the problem that dog doors bring – their vulnerability. Homeowners want to be tranquil, knowing that the privacy of their house will be maintained. Dog doors allow unwanted animals to come into people's homes, bringing a bad experience to the customer. D-Door is a mobile adaptation security system for dog doors, aiming to provide a solution, by controlling what goes through the door. Using a smart lock system on the door frame and a locking cap on the flap, the system locks and unlocks based on the collar's proximity to the door. The system can be operated under an *Offline mode* and an *Online mode*, where the customer is able to control the settings of the door and/or manually lock and unlock the door.

2. Scope

This document describes the electrical hardware and software behind the D-Door security system. Requirements, dependencies, and theory of operation can be found, along with a thorough description of the design details and testing. Schematics and flowcharts are appended at the end of the document for a better understanding of the hardware and software of the system.

3. Design Overview

3.1. Requirements

The following are requirements for the D-Door System:

3.1.1. Functional

I. The system utilizes an external 3.7v DC battery located on the door.
II. The system utilizes an external 3.7v DC battery located on the collar.
III. The system utilizes a *microcontroller* located on the door.
IV. The system utilizes a *microcontroller* located on the collar.
V. The system requires a *servo* to lock and unlock the system.

3.1.2. Performance

I. The system is required to unlock the door when the collar is near the door.
II. The system is required to lock the door when the collar is far from the door.
III. The system requires an *Offline* operating mode.
IV. The system requires an *Online* operating mode.

3.1.3. Usability

I. The system is required to respond to the customer's input on the user interface platform appropriately.
II. The system is required to recognize the customer's input on the user interface platform and act accordingly.
III. The *Door* and *Collar* are required to communicate with each other.

3.1.4. Aesthetic

I. The system requires a protective case for the door to protect its components.
II. The system requires a protective case for the collar to protect its components.
III. The system requires a cap to be fixated on the flap.
IV. The collar's protective case is required to be designed to be fixated on a dog collar.

3.1.5. Interface

I. The system requires a user interface platform for the customer to operate the system while on online mode.
II. The system requires a user interface platform for the customer to save the Wi-Fi information on the microcontroller.
III. The system requires a user interface platform for the customer to register the collars to the appropriate door.

3.2. Constraint

The following are dependencies for the D-Door to function:

3.2.1. Cost

I. The cost constraints from the project come from the microcontrollers, servo, battery, battery clip, connecting cables and aesthetic cases.
II. The cost constraint is estimated at $55 for a single door.

3.2.2. Physical

I. The system located on the door needs be able to fit on the door frame.
II. The system located on the collar needs to be able to fit a dog's neck.
III. The system located on the collar needs to be able to fit in a dog's collar.
IV. The cap located on the flap needs to be able to fit in a dog door's flap.
V. The system requires a 3.7v power supply.

3.2.3. Safety

I. The system located on the collar is required to be safe to be located on a dog's neck without harming them.
II. The system located on the collar is required to have zero exposed electrical contacts and/or mechanical pinch-points.
III. The system located on the door is required to be safe for a dog or human to touch without harming them.
IV. The system located on the door is required to have zero exposed electrical contacts and/or mechanical pinch-points.

3.3. Applicable Standards

The following are applicable standards for the D-Door:

I. Usage of *USB* for the programming of the microcontroller.
II. Usage of the *UART* for testing.
III. Usage of *Slave* and *Master* communication between collar and door.

IV. Usage of a connectionless communication protocol for communication between microcontrollers.
V. Usage of Wi-Fi to control the microcontroller.

3.4. Dependencies
The following dependencies for the D-Door to operate:

I. The system requires an external 3.7v DC power supply.

3.5. Theory of Operation
Upon reset or power-up, the system will display a Wi-Fi setup menu. The menu will contain a place for the customer to input their Wi-Fi information, so the microcontroller can gain internet access. If the customer chooses to continue with the *Offline mode of operation*, they must register the collar to the appropriate door, through a user interface platform – that way, the system can start to operate, and the communication door-collar can be established. In case the customer decides to have the system under the *Online mode of operation*, the customer needs to go to a user interface platform, so they can register the collar, control the settings of the door, lock and unlock the system manually and get help with the system in case they need.

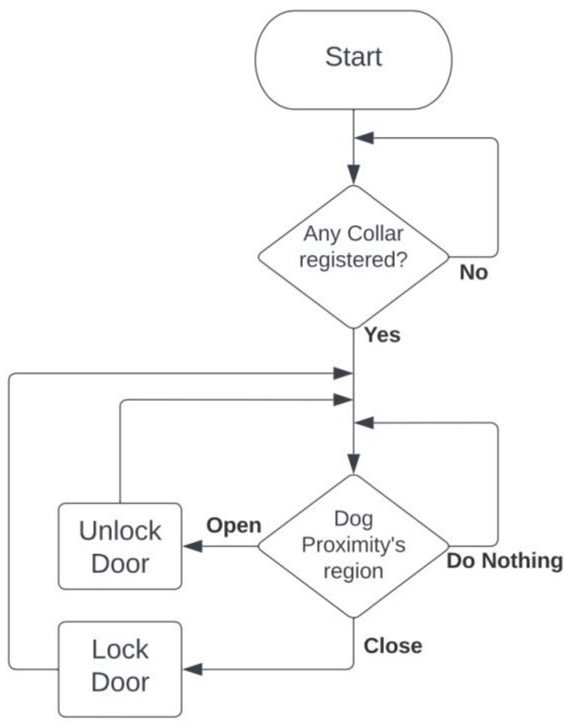

Figure 1: Offline mode of operation

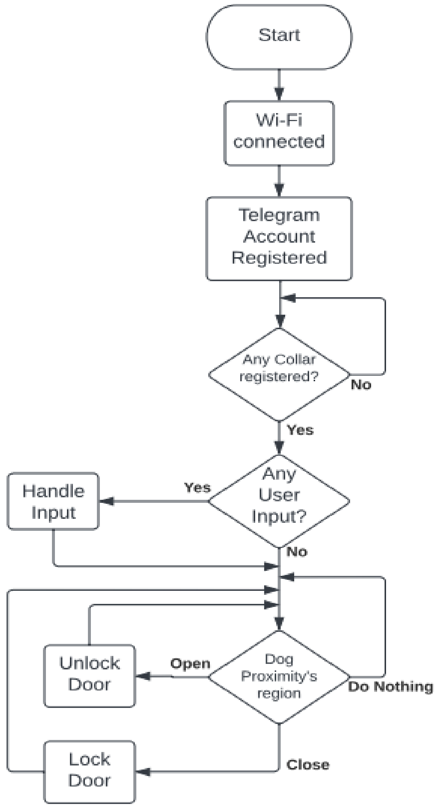

Figure 2: Online mode of operation

Using a User Interface Platform – while acting in the *Online mode of operation* – the customer should be able to change the settings of the door. There are two different functionalities for the door to have, including a *Normal Mode Functionality* and *Close Mode Functionality*. The difference between them is that the first mode has a "Do Nothing" region, where the door remains open, in case the dog decides to use the door again and to save battery. The *Close Mode Functionality* does not have that region, and once the collar is out of the *Open* range, the door locks.

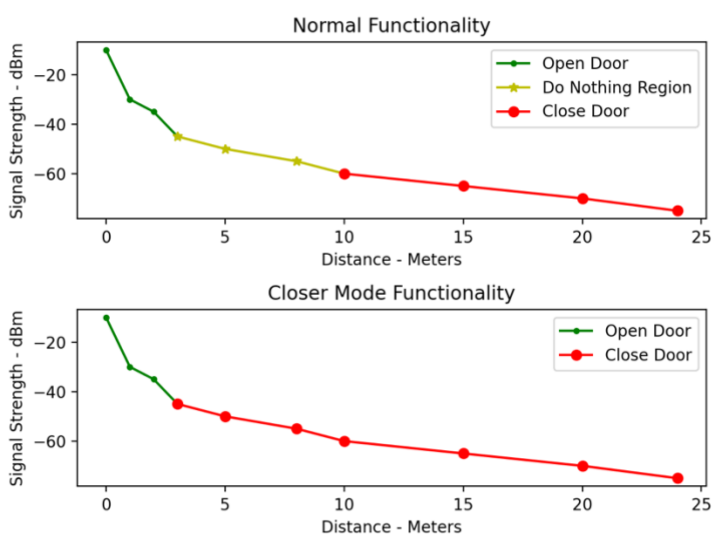

Figure 3: Range Illustration for the decisions

4. Design Details

4.1. Hardware

4.1.1. Door

For this system, the microcontroller chosen for the door was an *ESP32*, due to its Wi-Fi & Bluetooth capabilities, and due to its low cost. A 12g servo is used as a lock for the system. A power supply is needed to power the microcontroller, therefore, a 3.7v rechargeable battery is used, allowing the customer to have different ways to supply the power so the system can work. The options include (1) removing the battery and charging it, or (2) using a *Micro-USB* and connecting it to the port of the *ESP32*. The door connects to peripheral devices through female-female connector wires. The *General-Purpose Input/Output* (GPIO) chosen for the system to function were the following: *Pin 1, Pin 2, Pin 14, Pin 19, Pin 30*. The first two pins are used as connectors for the battery – *VCC and Ground*, respectively. The other mentioned pins were used for the servo – *Ground, VCC, and Signal*, respectively. The door schematic can be found in the *Appendices (figure 6)*.

4.1.2. Collar

For this system, the microcontroller chosen for the collar was an *ESP32*, due to its Wi-Fi & Bluetooth capabilities, and due to its low cost. A power supply is needed to power the microcontroller, therefore, a 3.7v rechargeable battery is used, allowing the customer to have different ways to supply the power so the system can work. The options include (1) removing the battery and charging it, or (2) using a *Micro-USB* and connecting it to the port of the *ESP32*. The collar connects to peripheral devices through female-female connector wires. The GPIOs chosen were for the system to function were the following: *Pin 1 and Pin 29*. The two pins are used as connectors for the battery – *VCC and Ground*, respectively. The collar schematic can be found in the *Appendices (figure 7)*.

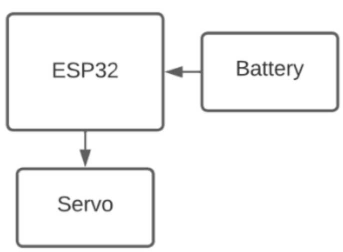

Figure 4: Door's Block Diagram

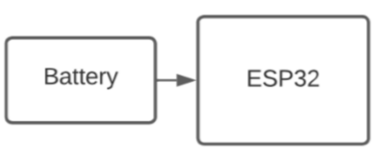

Figure 5: Collar's Block Diagram

4.2. Software

D-Door's full functionality is written using C++.

4.2.1. Door

The door was designed to be the brain of the system. In the door's microcontroller, the decision whether to lock or unlock must be made, according to the proximity region where the collar is found. Other capabilities that the door has are (1) Registering the appropriate collars to the door, (2) Listening to the *master's* message, (3) Send messages to *slaves*, (4) Get customer's input and acting accordingly, (5) Set the distance in meters based on the proximity of the collar, (6) Register multiple collars to the door. A flowchart with the decisions from the door can be found in the *Appendices (figure 8).*

4.2.2. Collar

The collar was designed to give flexibility to the system. Using the collar's microcontroller, a *message* is sent from the *collar* to the *door*, so the door can determine the collar's proximity to lock or unlock. Other capabilities that the collar has are (1) Registering the appropriate door to the collar, (2) Listening to the *master's* message, (3) Send messages to *slaves*, (4) Filter communication to have only *Door-Collar* communication (as opposed to *Collar-Collar* communication), (5) Set the distance in meters based on the proximity of the collar, (6) Determine if the collar is registered to any door. A flowchart with the decisions from the door can be found in the *Appendices (figure 9).*

5. Testing

5.1. Door

The microcontroller used on the door was tested by using a simple *Blink* example, where the LED found on the *ESP32* would turn on and off every second. The *servo* used in the door as a lock was tested by programming the *servo* to rotate (360°). The performance of the door was tested once the *collar* and the *door* were programmed to have duplex communication with each other. Using the embedded LED found on the *ESP32*, the light would turn on when the *master* sends a message and turn off when the *slave* receives a message, proving then an effective communication. The system proved to be effective when the *servo* locked the system when the collar was far away from the door and unlocked when they were near each other again. To test the *Offline Mode of Operation*, the collar was set near the door for the collar registration. Once the system realized that it had to take the *Offline* approach, it did, and it demonstrated success as the door would lock and unlock. To test the *Online Mode of Operation*, the User Interface Platform had to be done previously. It was proved to be working correctly when, after powering the door up, the Wi-Fi menu showed up, allowing the *ESP32* to connect to the internet. The system responded to the customer's input and recognized the input as a command for the system, acting accordingly. The door proved full functionality after these tests.

5.2. Collar

The microcontroller used on the collar was tested by using a simple *Blink* example, where the LED found on the *ESP32* would turn on and off every second. The performance of the collar was tested once the *collar* and the *door* were programmed to have duplex communication with each other. Using the embedded LED found on the *ESP32,* the light would turn on when the *master* sends a message and turn off when the *slave* receives a message, proving then an effective communication.

5.3. Battery

The power supply used to power the microcontrollers – found in the door and in the collar – were the 18560 rechargeable batteries, providing the 3.7v needed to operate the system. The batteries were tested with a multimeter and followed by using them in the system when they were fully charged. As expected, the system was powered up.

5.4. Aesthetics

The system required protective cases for the door and the collar, so the electrical components could be protected against external factors that could damage them. Therefore, using *Computer-Aided Design* (CAD), a unique protective case was designed for each, and 3D printed, to test using the components. After connecting all the cables, and placing the components in their respective cases, the system worked, proving the cases to be effective. Using the same methods, the cap for the flap was also designed and printed. It was tested by putting it above the *servo*, to make sure the system would lock and unlock the cap – proving successful. Lastly, the case and the system of the collar were put on a collar, to make sure it fitted and that it could be used on a dog's neck – which it did.

5.5. Interface

After the User Interface Platform was created, it was tested multiple times by adding commands that the customer could choose to control the door. Commands were chosen based on usability and need. The customer was able to input their Wi-Fi parameters, register the appropriate collars and the system responded to their commands from the platform.

6. Conclusion

D-Door was proven feasible for creation, usage, and production. Improvements along the way were needed, such as new commands, better user experience graphics for the User Interface Platform, design, and fitting of the components on their protective cases… Meeting the desired time requirements of the project, the design, construction, and implementation were all completed successfully. The appropriate testing was made to prove the functionality of all used components and to meet the requirements for a final working product. The system behaved as desired, and its capabilities were shown to be near ideal in terms of performance. For future reference, creating flowcharts, and breaking down functions is very important. That way, it saves time by being able to visualize what to do and program it in a clean method.

7. References and Bibliography

ESP NOW, https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/network/esp_now.html.

ESP-IDF Programming Guide. https://www.mouser.com/datasheet/2/891/esp_idf_en_v4_4_dev_960_gcf457d4_esp32s2-2301164.pdf.

Espressif. "Espressif Core for the ESP32." *GitHub*, https://github.com/espressif/arduino-esp32.

"Wi-Fi Manager Methods." *GitHub*, https://github.com/tzapu/WiFiManager/wiki/Methods.

8. Appendices

8.1. Schematics

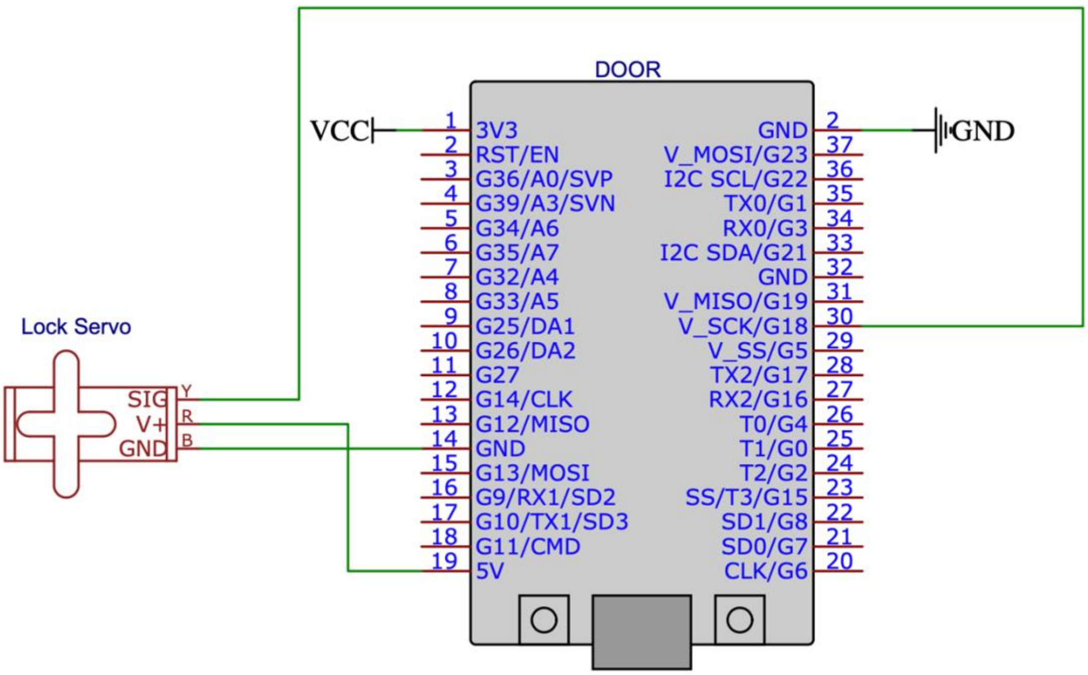

Figure 6: Door's Schematic (ESP32 NodeMCU)

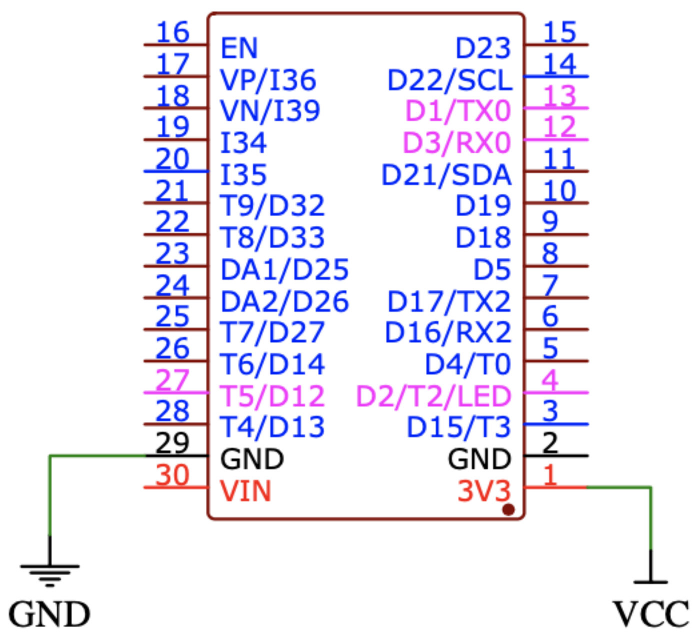

Figure 7: Collar's Schematic (Devkit V1)

8.2. Flowcharts

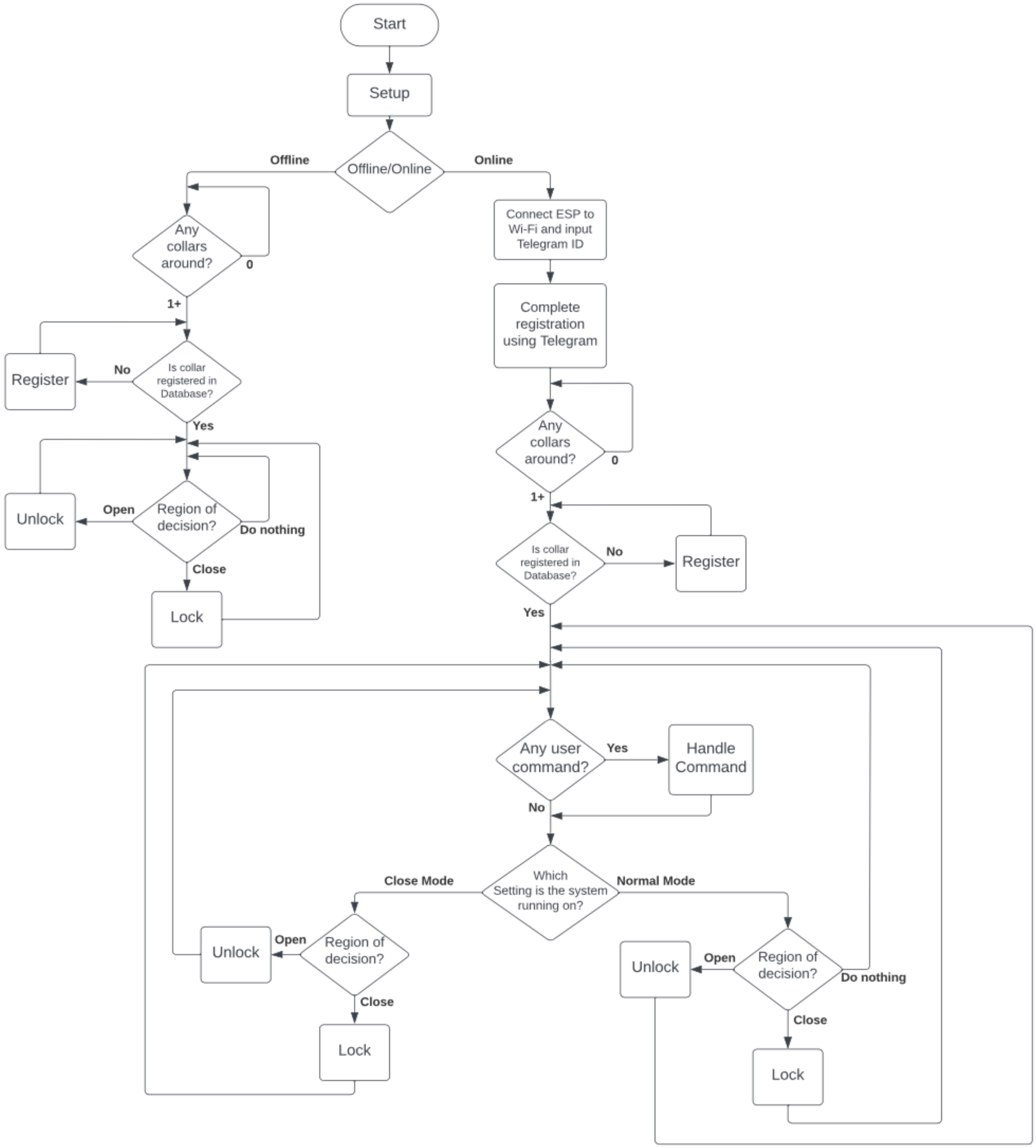

Figure 8: Door's Flowchart

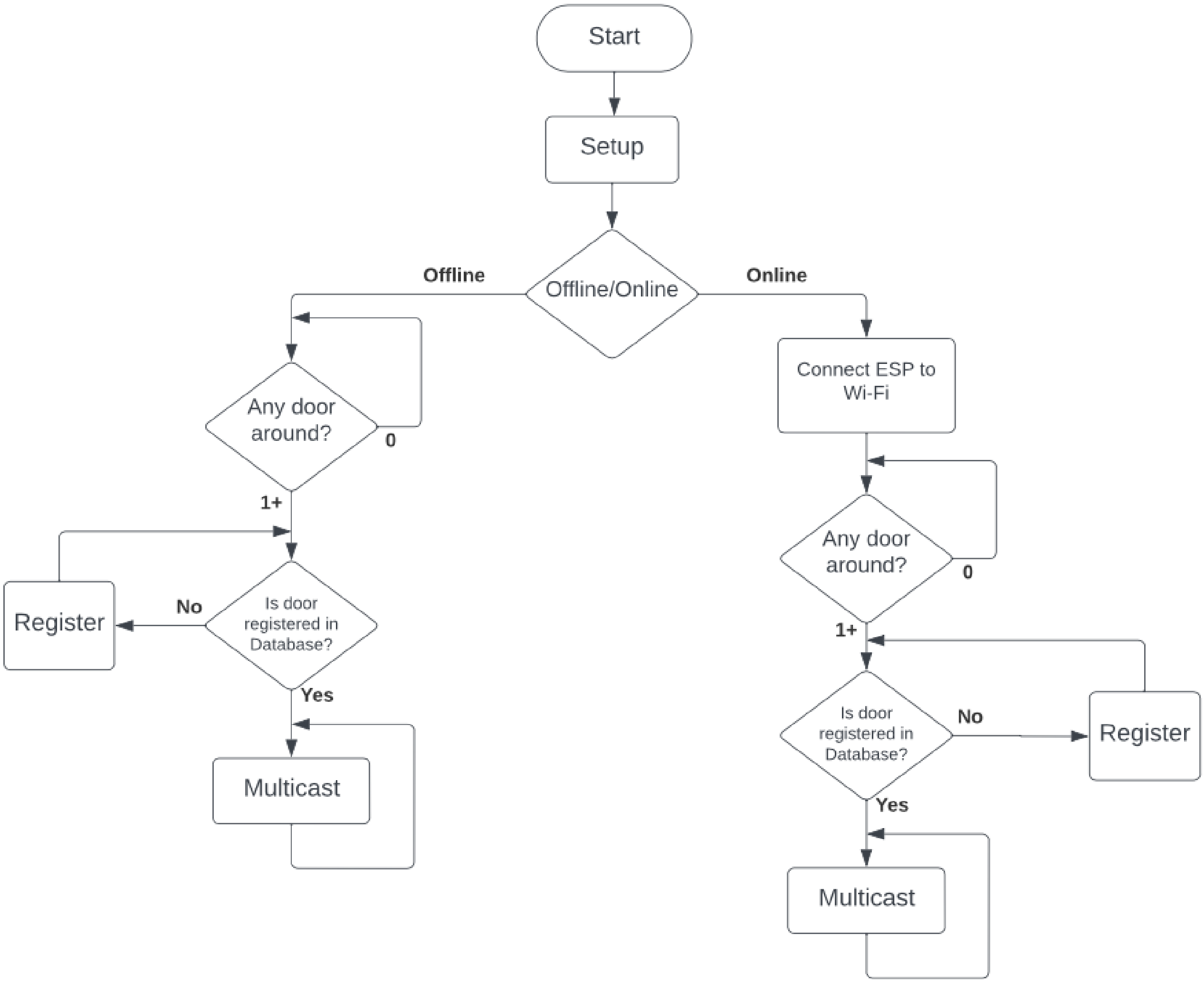

Figure 9: Collar's Flowchart

9. Acknowledgements:

I would like to take this page to acknowledge the people that supported this project from the beginning and that were important for its success

- Heliane A. Mougenot Breviglieri Silva
- José Roberto P. da Silva, PhD
- César A. Cabral Franco
- Abbigale R. Mougenot
- Liana R. Mougenot
- The Bingham Family
- Dr. Brown
- Dr. West
- Dr. Hearn
- Dr. Jackson

Figure 10: D-Door Logo