



Fechadura digital

Gabriel Marotti Pestana, Geraldo Tenório

Universidade Presbiteriana Mackenzie (UPM)
Rua da Consolação, 930 Consolação, São Paulo - SP, 01302-907 – Brasil

`gabriel.marotti.pestana@gmail.com, g.tenorio19@gmail.com.`

Abstract. *This project aims to increase personal safety and likewise create a technological interaction for accessing environments. From the Arduino, we are going to develop a remote actuation lock. This lock has the functions, user presence sensor and user validation from the environment's WI-FI network.*

Resumo. *Este projeto tem a finalidade de aumentar a segurança pessoal e da mesma maneira criar uma interação tecnológica para acesso a ambientes. A partir do Arduino Uno R3, vamos desenvolver uma tranca de acionamento remoto. Esta tranca tem como funções, sensor de presença do usuário e validação de usuário a partir de uma UID.*

1. Introdução

O projeto Fechadura digital tem como tarefa trancar e destrancar o ambiente de forma automática, presencialmente ou remotamente. Após o usuário aproximar o cartão ou a tag de identidade e ser validado na rede, o acesso ao ambiente é liberado. Caso, ele não seja autenticado, vamos notificar o usuário administrador através do log de tentativas de acesso. Ao deixar o ambiente é acionado um sensor de movimento que irá te enviar notificações caso alguém esteja no local apresentado um código inválido.

2. MATERIAIS E MÉTODOS

Para realizar a implementação desta automação será utilizado os seguintes componentes de hardware abaixo:

2.1. MATERIAIS

2.1.1 Arduino Uno R3

O Arduino Uno R3 consiste em um micro controlador, ele irá gerenciar a automação, por meio de um *Sketch* desenvolvido no Arduino IDE a partir da linguagem de programação C++, realizando a comunicação via hardware, através de um cabo USB 2.0 A/b (FIGURA 1).

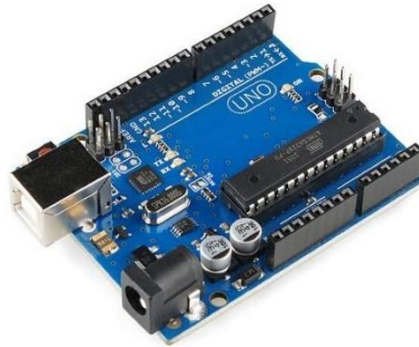


Figura 1: Placa Arduino UNO R3.

Fonte: (GIOVANNI BAUEMEISTER, 2018).

Especificações Técnicas do Arduino Uno R3	
Micro controlador	ATmega328 (<i>datasheet</i>)
Tensão de Operação:	5v
Tensão de Entrada	7-12V
Portas Digitais	14
Portas Analógicas	6
Corrente Pinos I/O	40mA
Corrente Pinos 3,3V	50mA
Memória Flash	32KB
SRAM	2KB
EEPROM	1KB
Velocidade do <i>Clock</i>	16MHz

Tabela 1: Especificações Técnicas do Arduino Uno R3.

Fonte: (GIOVANNI BAUEMEISTER, 2018).

2.1.2. Sensor de presença PIR-HC-SR501

A proposta de utilizar este sensor é que ao identificar um movimento, a fechadura receba o UID cadastrado no sistema e então efetue a liberação, caso não seja apresentado o UID ou um válido, a fechadura não efetua a liberação e enviamos uma notificação ao administrador, desta forma ele pode consultar o log de tentativas de acesso e qual o UID apresentado (FIGURA 2).

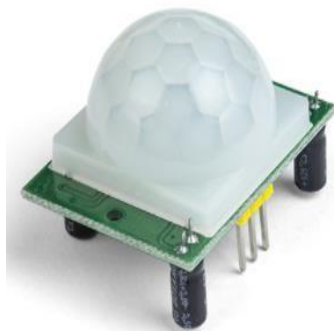


Figura 2: Sensor de Presença PIR - HC-SR501.

Fonte: (CACPNRJ, 2020).

Especificações Técnicas do Sensor PIR-HC-SR501	
Alimentação	de 5V a 20V
Tensão de Saída	3,3V
Potenciômetro para ajuste de tempo e sensibilidade Ângulo de abertura	aproximadamente 120°
Dimensões	33 x 25 x 25 mm

Tabela 2: Especificações Técnicas do Sensor PIR-HC-SR501.

Fonte: (CACPNRJ,2020).

2.1.3. Jumpers

Os Jumpers serão utilizados para realizar a conexão entre o Arduino Uno R3 com a Protoboard, sensor PIR-HC-SR501 e RFID-RC522, através deles será possível estabelecer a energia entre as peças (FIGURA 3).

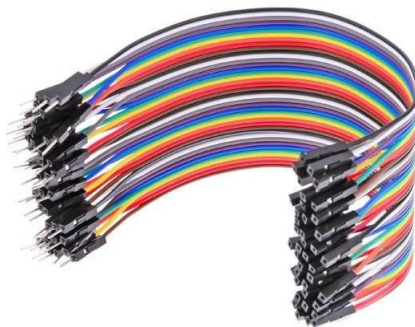


Figura 3: Jumpers Macho e Fêmea.

Fonte: (AUGUSO CAMPOS,2015).

2.1.4. Protoboard

A Protoboard de 400 pontos, será utilizada para realizar a prototipagem e conexões condutoras, desta forma a partir dos Jumpers, iremos transmitir a energia suficiente e conectar: Arduino Uno R3, sensor PIR-HC-SR501, LED Difuso 5mm e Resistor 1K Ω 1/4W (FIGURA 4).

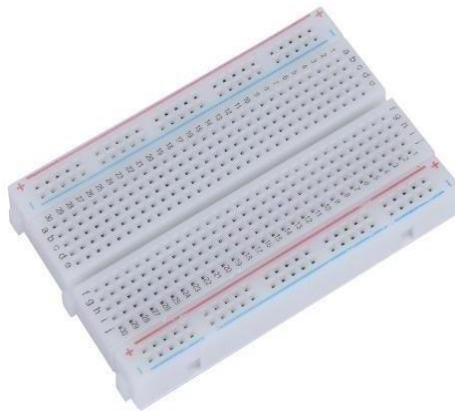


Figura 4: Protoboard.

Fonte: (ALLAN MOTA, 2018).

Especificações Técnicas da Protoboard	
Dimensões	83.5 x 54.5 x 8.5mm
Quantidade de barramento para alimentação	2
Quantidade colunas	30
Quantidade linhas	10
Quantidade de pontos	400
Compatível com fios	de 29AWG até 20AWG

Tabela 3: Especificações Técnicas da Protoboard.

Fonte: (ALLAN MOTA, 2018).

2.1.5. Kit Módulo Leitor RFID MFRC522 Mifare

Este kit será utilizado como um comunicador para efetuar leitura em uma frequência específica seguindo o padrão Mifare. A leitura consiste na aproximação da tag ou do cartão para validação de acesso. Para efetuar a liberação, foi cadastrado o UID da tag e do cartão (FIGURA 5).



Figura 5: Kit Módulo Leitor RFID MFRC522 Mifare.

Fonte: (EULER OLIVEIRA, 2019).

Especificações Técnicas do Módulo Leitor RFID MFRC522 Mifare	
Modelo	MFRC522
Tensão de operação	3,3VDC
Corrente de operação	13mA a 26mA
Tensão em modo inativo	3,3VDC
Corrente em modo inativo	10mA a 13mA
Frequência de operação	13,56MHz
Interface	SPI
Taxa de transferência	10 Mbit/s
Alcance	0 a 3cm
Cartões suportados	Mifare1 S50, Mifare1 S70, Mifare Ultralight, Mifare Pro, Mifare Desfire
Temperatura de operação	-20° a 80° celsius

Tabela 4: Especificações Técnicas do Módulo Leitor RFID MFRC522 Mifare.

Fonte: (EULER OLIVEIRA, 2019).

Especificações e características (Cartão 13,56MHz)	
Frequência de operação	13,56MHz
Taxa de transferência	106 Kbaud
Capacidade	8 Kbit / 16 partições
Alcance	2 a 10cm
Tempo de leitura e escrita	1 a 2ms
Temperatura de operação	-20° a 55° celsius
Material	PVC

Tabela 5: Especificações e características (Tag Cartão 13,56MHz).

Fonte: (EULER OLIVEIRA, 2019).

Especificações e características (Tag Chaveiro 13,56MHz)	
Frequência de operação	13,56MHz
Taxa de transferência	106 Kbaud
Capacidade	8 Kbit / 16 partições
Alcance	2 a 10cm
Tempo de leitura e escrita	1 a 2ms
Temperatura de operação	-20° a 80° celsius
Material	ABS

Tabela 6: Especificações e características (Tag Chaveiro 13,56MHz).

Fonte: (EULER OLIVEIRA, 2019).

2.1.6. LED Difuso 5mm Verde e Vermelho

Os LEDs nas cores verde e vermelho serão utilizados para identificar o status de verificação do acesso. Quando o LED vermelho acender haverá acesso negado para entrada, já quando o verde acender haverá acesso permitido (FIGURA 6).



Figura 6: LED Verde e Vermelho.

Fonte: (ALLAN MOTA, 2018).

Especificações Técnicas do LED Difuso 5mm Verde e Vermelho	
Modelo	LVD5M
Tensão de alimentação	2 a 2.2V DC
Corrente máxima	20 mA
Ângulo de abertura	120°
Diâmetro do LED	5mm
Comprimento com terminais	~37mm
Peso unidade	0,3g

Tabela 7: Especificações Técnicas do LED Difuso 5mm Verde e Vermelho

Fonte: (ALLAN MOTA, 2018).

2.1.7. Resistor 1K Ω 1/4W

Vamos utilizar dois resistores com a finalidade de limitar a corrente elétrica na Protoboard, a fonte de alimentação neste caso será conduzida pelo cabo USB 2.0 A/b, que será conectado em uma entrada UBS do *Notebook* ou Computador, e na porta serial do Arduino Uno R3 (FIGURA 7).

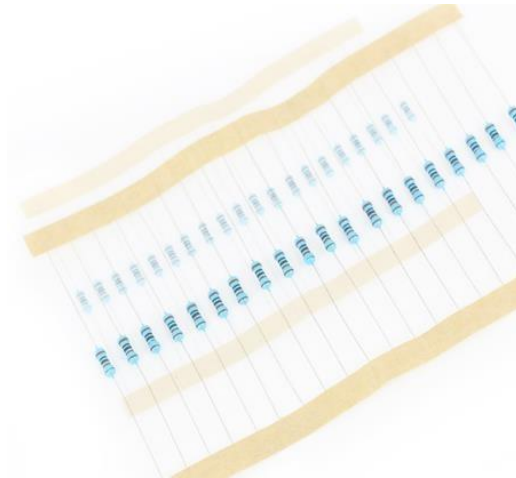


Figura 7: Resistor 1K Ω 1/4W

Fonte: (ALLAN MOTA, 2018).

Especificações Técnicas do Resistor 1K Ω 1/4W	
Padrão	CR25
Resistência	1K Ohm
Tolerância	$\pm 5\%$
Potência	1/4W
Comprimento total	58mm

Tabela 8: Especificações Técnicas do Resistor 1K Ω 1/4W.

Fonte: (ALLAN MOTA, 2018).

2.1.8. Cabo USB 2.0 A/b

Com este cabo temos a fonte de energia para funcionamento da placa Arduino Uno R3, como demonstra a imagem abaixo, vamos realizar a conexão do cabo em uma porta USB do computador ou Notebook e conectar a outra ponta na porta serial da placa Arduino Uno R3. (FIGURA 8).



Figura 8: Cabo USB 2.0 A/b.

Fonte: (FÁBIO SOUZA, 2013).

Especificações Técnicas do Cabo USB 2.0 A/b	
Conector 1	USB A
Conector 2	USB B
Versão USB	USB 2.0
Gênero do conector	Macho/Macho
Cor	Azul

Tabela 9: Especificações Técnicas do Cabo USB 2.0 A/b
(FÁBIO SOUZA, 2013).

2.2. MÉTODOS

2.2.1. MQTT

Para realizar a comunicação com a internet, seguimos a sugestão de realizar a conexão com o Mosquitto e através do *NodeRed* desenvolver e estabelecer uma comunicação com o protocolo MQTT.(FIGURA 9).

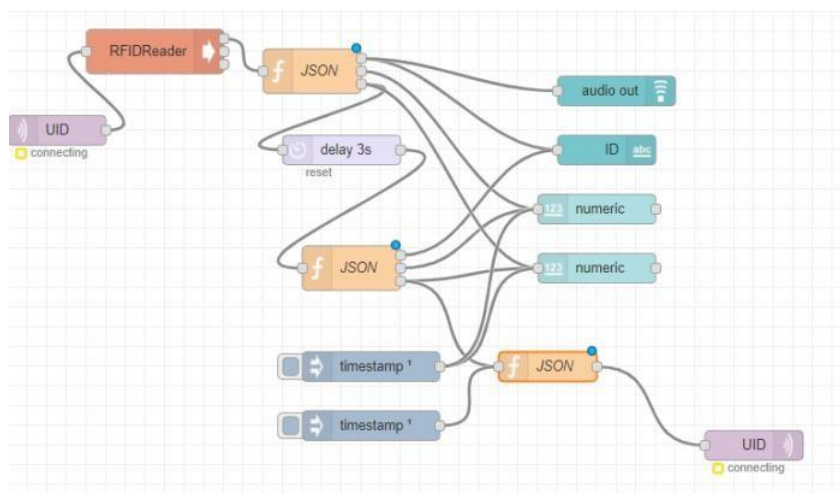


Figura 9: Node-Red.

Fonte: (Autores, 2021).

Como mostra o fluxo acima, configuramos um nó de entrada para o MQTT, informando o IP e porta de conexão no caso `http://127.0.0.1:1883`.

Em seguida criamos uma função em Json, neste código informamos o UID da tag e do cartão de acesso, desta forma será validado o UID apresentado e a qual usuário ele se refere, a saída de audio informa qual usuário foi liberado de acordo com o o UID.

Após realizar um acesso, inserimos um *delay* de 3 segundos, que é o tempo que entendemos ser necessário para um proximo acesso.

A segunda função Json, é apenas uma validação de Led's, acesso liberado deve ascender o led verde, acesso negado o led vermelho, está função também esta ligada a um ID, para identificarmos o usuário.

Os demais nós “*numeric*” e “*timestamp*”, se referem a medidas de segurança, ou seja caso o usuário perca o cartão de acesso ou tag, ele pode se conectar na plataforma web com seu acesso administrador e através de uma senha numérica, a fechadura pode ser liberada, após 3 segundos uma nova tentativa pode ser efetuada.

Por fim, armazenamos as trocas dessas mensagens em uma função Json, e conectados ao nó de saída do MQTT, desta forma todo o processo esta registrado no *Broker*

Sendo assim, abaixo temos a tela de execução do Node JS, através da conexão do *Node- Red* com o mosquitto, neste log apontamos que um servidor foi iniciado e que foi estabelecida uma conexão com o *Broker*. Em seguida o log informa que está enviando e recebendo mensagens via protocolo MQTT/ TCP via QoS 2. (FIGURA 10).

```
Welcome to Node-RED
=====
1 Dec 22:22:46 - [info] Node-RED version: v0.18.4
1 Dec 22:22:46 - [info] Node.js version: v5.7.0
1 Dec 22:22:46 - [info] Windows_NT 10.0.19042 ia32 LE
1 Dec 22:22:47 - [info] Palette editor disabled : npm command not found
1 Dec 22:22:47 - [info] Loading palette nodes
1 Dec 22:22:48 - [info] Dashboard version 2.9.4 started at /ui
1 Dec 22:22:48 - [warn] -----
1 Dec 22:22:48 - [warn] [node-red/rpi-gpio] Info : Ignoring Raspberry Pi specific node
1 Dec 22:22:48 - [warn] [node-red/tail] Not currently supported on Windows.
1 Dec 22:22:48 - [warn] -----
1 Dec 22:22:48 - [info] Settings file : C:\Users\AS-Informatica\Downloads\NodeJSPortable_nodered_mosquitto\NodeJSPortable_nodered_mosquitto\Data\.node-red\settings.js
1 Dec 22:22:48 - [info] User directory : C:\Users\AS-Informatica\Downloads\NodeJSPortable_nodered_mosquitto\NodeJSPortable_nodered_mosquitto\Data\.node-red
1 Dec 22:22:48 - [warn] Projects disabled : set editorTheme.projects.enabled=true to enable
1 Dec 22:22:48 - [info] Flows file : C:\Users\AS-Informatica\Downloads\NodeJSPortable_nodered_mosquitto\NodeJSPortable_nodered_mosquitto\Data\.node-red\flows_DESKTOP-
P-E407TLQ.json
1 Dec 22:22:48 - [info] Server now running at http://127.0.0.1:1880/
1 Dec 22:22:48 - [info] Starting flows
1 Dec 22:22:48 - [info] Started flows
1638408168: New connection from 127.0.0.1 on port 1883.
1638408168: New client connected from 127.0.0.1 as mqtt_6526eb5c.9ad914 (c1, k60).
1638408168: Sending CONNACK to mqtt_6526eb5c.9ad914 (0, 0)
1 Dec 22:22:48 - [info] [mqtt-broker:cd5780d5.b1056] Connected to broker: http://127.0.0.1:1883/
1638408168: Received SUBSCRIBE from mqtt_6526eb5c.9ad914
1638408168: UID (QoS 2)
1638408168: mqtt_6526eb5c.9ad914 2 UID
1638408168: Sending SUBACK to mqtt_6526eb5c.9ad914
1638408168: Received PUBLISH from mqtt_6526eb5c.9ad914 (d0, q2, r0, m62219, 'UID', ... (3 bytes))
1638408168: Sending PUBREC to mqtt_6526eb5c.9ad914 (Mid: 62219)
1638408168: Received PUBREL from mqtt_6526eb5c.9ad914 (Mid: 62219)
1638408168: Sending PUBCOMP to mqtt_6526eb5c.9ad914 (Mid: 62219)
1638408168: Sending PUBLISH to mqtt_6526eb5c.9ad914 (d0, q2, r0, m1, 'UID', ... (3 bytes))
1638408168: Received PUBREC from mqtt_6526eb5c.9ad914 (Mid: 1)
1638408168: Sending PUBREL to mqtt_6526eb5c.9ad914 (Mid: 1)
1638408168: Received PUBCOMP from mqtt_6526eb5c.9ad914 (Mid: 1)
```

Figura 10: Node JS: Conexão com MQTT.

Fonte: (Autores, 2021).

2.2.2. Software

Para este projeto utilizamos o *Software* Arduino IDE, que permite realizar a programação do micro controlador (Arduino Uno R3), a partir de *Sketcht's* com a linguagem de programação C++. De acordo com orientação em aula, optamos por inserir no código a comunicação com o protocolo Firmata, para que fosse possível a comunicação com o protocolo MQTT, de acordo com *Node-Red* acima (FIGURAS 10 e 11).

Através do protocolo MQTT, vamos enviar ao *Broker* o log de acesso da fechadura. Desta forma o administrador do sistema poderá consultar as tentativas de acesso e códigos apresentados.

2.2.3. Fluxograma de Funcionamento, Protoboard e Esquemático.

O fluxograma de funcionamento, Circuito desenvolvido e o Esquemático, abaixo ilustra o modo em que foi feita a montagem do protótipo (FIGURA 11, 12 e 13).

Como podem visualizar abaixo, o sensor PIR-HC-SR501 fica ativo esperando detectar um movimento, caso seja identificado o movimento, o usuário deve aproximar o cartão ou tag de acesso do leitor RFID-RC522, se o UID apresentado for o cadastrado no sistema, a fechadura ira liberar o acesso. (FIGURA 11).

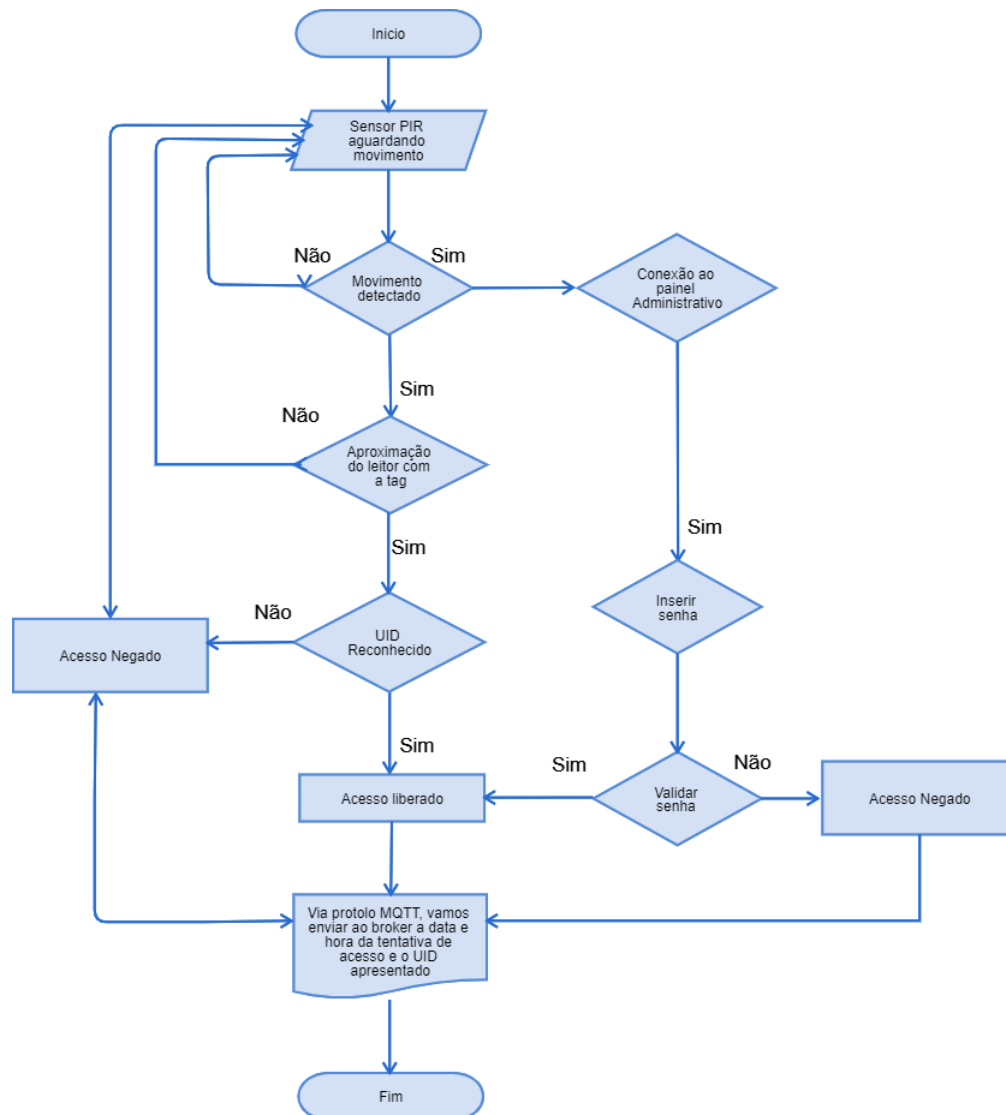


Figura 11: Fluxograma de Funcionamento

Fonte: (Autores, 2021).

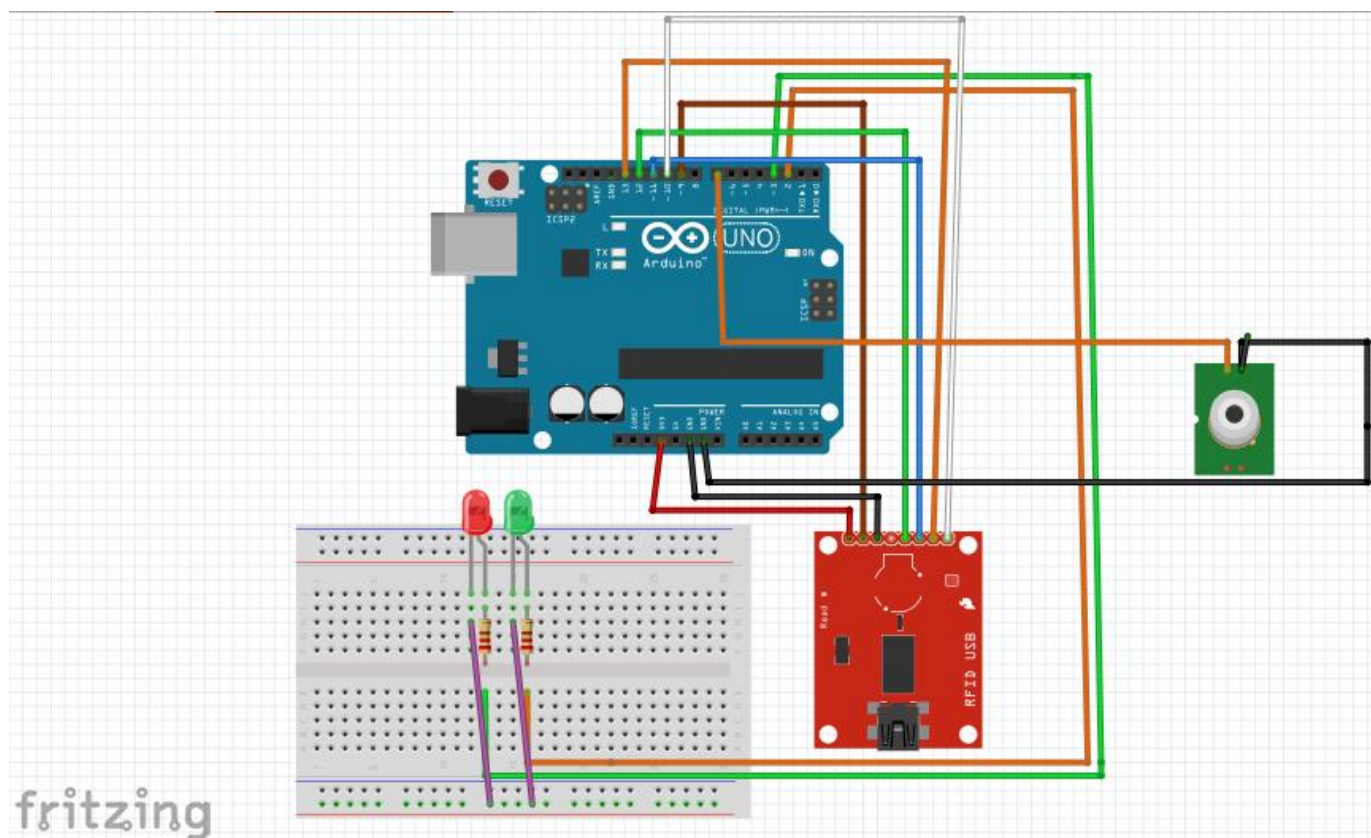


Figura 12: Circuito desenvolvido

Fonte: (Autores, 2021).

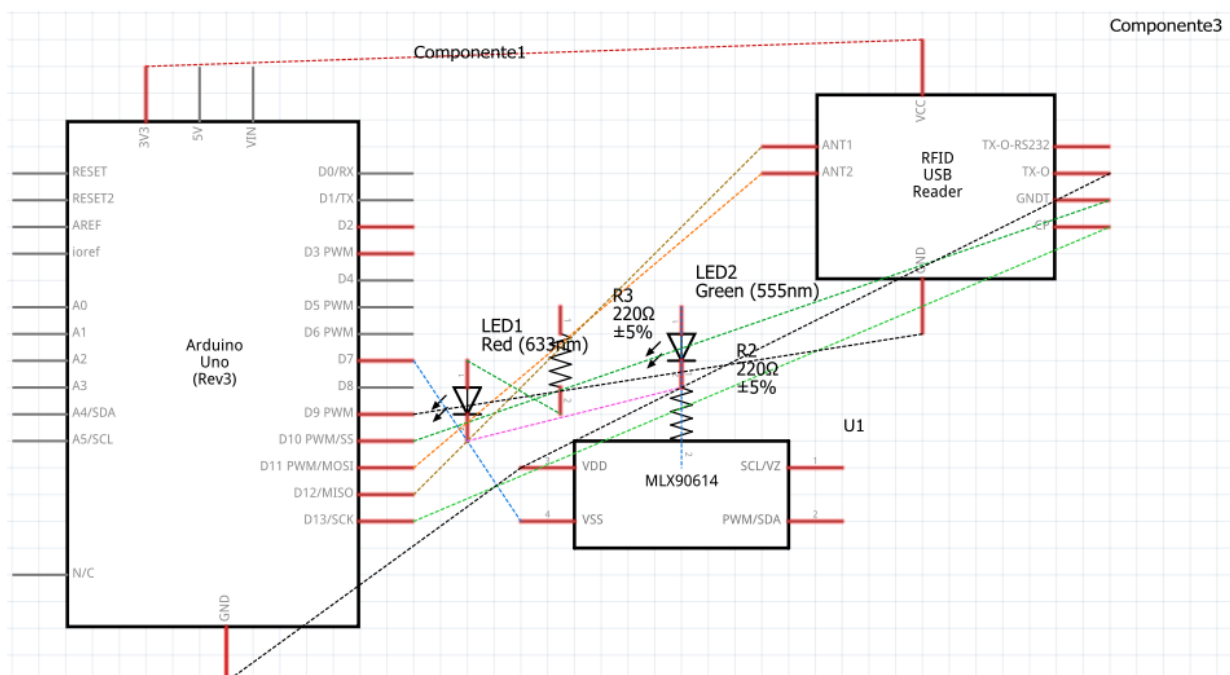


Figura 13: Esquemático

Fonte: (Autores, 2021).

2.2.4. Descrição Arquitetural

Sobre a arquitetura deste projeto, utilizamos o sensor PIR-HC-SR501 e fizemos uma validação do modulo RFID-RC522, com o intuito de leitor de aproximação. A função deste modulo é reconhecer as UID's que foram cadastradas no sistema.

Sendo assim, quando o sensor PIR-HC-SR501 identificar movimento deverá ser apresentado uma UID. Caso esta UID seja uma das que foram cadastradas, a tranca será liberada para acesso, do contrário o acesso será negado.

A forma para realizar a conexão com o dispositivo é pelo modulo RFID-RC522, o usuário deverá apresentar uma UID para conseguir destravar e realizar o acesso.

Para que possamos ter uma ideia de tentativas de acesso, através do código fonte desenvolvido, vamos enviar via MQTT o log de acesso ao *Broker*.

Arduino UNO R3	Protoboard	PIR-HC-SR501	RFID-RC522
5V	Pino 2		
RES		Conectado ao sensor	
3.3V			3.3V
5v	Pino 8		
GND		Conectado ao sensor	GND
GND			
13			SCK
12			MOSI
11			SOI
10			S D A
9			RST
7		Conectado ao sensor	
3			
2			
TX -> 1			
RX -> 0			

Tabela 10: Especificação de montagem do protótipo.

Fonte: (Autores, 2021).

3. RESULTADOS

3.1. Descrição do Produto Final e Apresentação do Funcionamento.

No início do projeto, pensamos em uma fechadura digital que poderia efetuar liberação com o porte de uma UID. A comunicação com a internet e por questão de segurança de acesso, pensamos em restringir apenas em registro contendo o log de tentativas de acesso.

Quando o cartão magnético ou a tag são aproximados da placa RFID-RC522, o mesmo gera uma frequência que valida os acessórios e consegue liberar o acesso ao ambiente que a fechadura está empregada. (FIGURA 14).

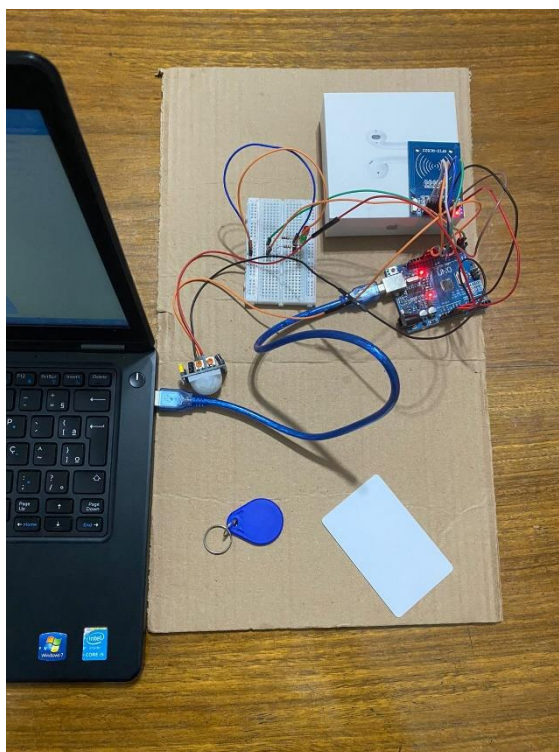


Figura 14: Protótipo desenvolvido

Fonte: (Autores, 2021).

3.2. Principais Problemas

Os principais problemas levantados foram:

- Falta da fonte 12V para efetuar conexão com o Módulo Relé e a Mini trava elétrica Solenoide 12V.

3.3 Apresentação do Projeto

< <https://www.youtube.com/watch?v=HsHWyOG0670> >

3.4 Repositório do Projeto

< <https://github.com/gmarotti/Objetos-Inteligents> >

4. CONCLUSÃO

Conseguimos atingir a ideia que tivemos no início do projeto, e até incluir adicionais, como o sensor PIR-HC-SR501, não tínhamos percebido como exatamente iríamos utilizar neste projeto.

A vantagem de utilizar o nosso projeto, é proporcionar ao usuário uma maior segurança ao ambiente que a fechadura está instalada, e que ele consiga receber *reports* em tempo real caso ela seja acionada.

A maior desvantagem, é que hoje infelizmente ainda não está disponível um suporte para conexão com dispositivos de *Home Assistant* ou *Bluetooth*, dispositivos móveis em geral.

Para melhorar o projeto, a ideia é um sistema de segurança completo, digo estabelecer uma comunicação entre as ferramentas, então com o mesmo projeto de fechadura digital, envolver câmeras, e controlar por um *Smartphone*, conectar com dispositivos de *Home Assistant*.

Referências

- Bauermeister, G. (2018). “Fechadura eletrônica com Arduino e RFID” In: < <https://blog.fazedores.com/fechadura-eletronica-com-arduino-e-rfid/>>.
- Cacpnrj, (2020). “PIR HC-SR501”. In: < <https://capsistema.com.br/index.php/2020/12/10/como-usar-o-sensor-de-movimento-pir-hc-sr501-com-arduino/>>
- Mota, A. (2018). “Led Difuso 5mm Verde”. In: < <https://portal.vidadesilicio.com.br/protoboard//>>
- Mota, A. (2018). “Led Difuso 5mm Vermelho”. In: < <https://portal.vidadesilicio.com.br/protoboard//>>
- Oliveira, E. (2019). “Kit Módulo Leitor RFID Mfrc522 Mifare”. In: < <https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-kit-rfid-mfrc522/>>
- Mota, A. (2018). “Resistor 1KΩ 1/4W”. In: < <https://portal.vidadesilicio.com.br/protoboard//>>
- Mota, A. (2018). “Protoboard”. In: < <https://portal.vidadesilicio.com.br/protoboard//>>
- Campos, A. (2015). “Jumpers Macho e Fêmea” In: < <https://uploads.filipeflop.com/2017/07/2CB08-2.jpg> >
- Ribeiro, C. (2019). “Fechadura digital com Arduino, RFID e Teclado.” In: < <https://medium.com/@caio.ribeiro/fechadura-digital-com-arduino-rfid-e-teclado-e0f836d0e11b>>.