



ETEC JORGE STREET

TRABALHO DE CONCLUSÃO DO CURSO TÉCNICO EM ELETROTÉCNICA

IMOBILIZADOR BIOMÉTRICO

**Jorge Luiz dos Santos
José Lucian da Silva
Rodrigo Antônio de Oliveira
Wander Ferreira dos Santos**

**Professor Orientador:
Renato Francisco Agostinho**

**São Caetano do Sul / SP
2018**

IMOBILIZADOR BIOMÉTRICO

Trabalho de Conclusão de Curso
apresentado como pré-requisito para
obtenção do Diploma de Técnico em
Eletrotécnica.

**São Caetano do Sul / SP
2018**

AGRADECIMENTOS

Aos colegas do curso com os quais tive oportunidade de conviver durante a aquisição da aprendizagem e repartir incertezas na caminhada em busca desta ascensão cultural.

Às nossas famílias que de maneira direta ou indireta apoiaram nossos esforços.

RESUMO

O projeto Imobilizador Biométrico foi desenvolvido para mais uma segurança do usuário de motocicleta com partida elétrica, permitindo o cadastramento da digital do usuário para a ligação do veículo, dificultando desta forma o furto do mesmo.

Palavras-chave: imobilizador, biométrico, digital, ligação.

LISTA DE FIGURAS

| | |
|--|---------|
| Figura 1 – MicroControlador Arduíno | 10 |
| Figura 2 – Esquema elétrico do Arduino | 142 |
| Figura 3 – Esquema elétrico do Arduino | 103 |
| Figura 4 – Tipos de Placas de MicroControlador Arduíno..... | 14 |
| Figura 5 – Estrutura de programa de MicroControlador Arduíno..... | 14 |
| Figura 6 – Programação de MicroControlador Arduíno..... | 14 |
| Figura 7 – Leitor Biométrico | 14 |
| Figura 8 – Módulo Relé | 20 |
| Figura 9 – Diagrama em Blocos | 21 |
| Figura 10 – Desenho Leitor Biométrico no Case..... | 23 |
| Figura 11 – Desenho do Sistema Imobilizador..... | 23 |
| Figura 12 – Leitor Biométrico Instalado no Case..... | 26 |
| Figura 13 – Arduíno Fixo no Case..... | 27 |
| Figura 14 – Relé Fixo na Tampa do Case..... | 27 |
| Figura 15 – Esquema Elétrico do Imobilizador Biométrico | 28 |
| Figura 16 a Figura 16.5 – Programação do Imobilizador Biométrico..... | 29 a 34 |
| Figura 17 a Figura 17.1 – Montagem / Teste do Imobilizador Biométrico | 35 |
| Figura 18 a Figura 18.1 – Ligação dos Conectores..... | 36 a 37 |
| Figura 19 – Esquema de Ligação do Imobilizador Biométrico..... | 38 |
| Figura 20 – Teste do Cadastro da Digital do Imobilizador Biométrico | 39 |
| Figura 21 – Digital não Apresentada não Libera o Pulso Elétrico do Imobilizador Biométrico | 39 |
| Figura 22 – Liberação do Pulso Elétrico Representado pelo Led do Imobilizador Biométrico | 40 |
| Figura 23 – Localização do Leitor Biométrico..... | 41 |
| Figura 24 – Moto Ligada e Leitor Biométrico Energizado..... | 42 |
| Figura 25 – Verificação da Digital Cadastrada | 43 |
| Figura 26 – Relé de Partida Acionado após Verificação da Digital | 44 |
| Figura 27 – Liberação da Partida da Moto | 45 |
| Figura 28 – Moto Ligada | 46 |

Sumário

| | |
|---|---------|
| Introdução | 8 |
| 1 – Fundamentação Teórica..... | 10 |
| 1.1 – Micro Controlador Arduíno | 16 |
| 1.1.1 – Estrutura de um programa em Arduíno | 20 |
| 1.2 – Leitor Biométrico | 16 |
| 1.2.1 – Especificações | 20 |
| 1.2.2 – Características | 20 |
| 1.3 – Módulo Relé | 20 |
| 1.3.1 – Informações Técnicas | 20 |
| 2 – Planejamento do Projeto | 21 |
| 2.1 – Parte Eletroeletrônica | 21 |
| 2.1.1 – Diagrama em Blocos | 21 |
| 2.1.2 – Pesquisa de Componentes Tecnologias | 21 |
| 2.1.3 – Previsão de Custos | 22 |
| 2.2 – Parte Lógica - Fluxograma do Processo | 23 |
| 2.3 – Parte Mecânica | 24 |
| 2.4 – Cronograma Inicial | 25 |
| 2.5 – Cronograma Final..... | 26 |
| 3 – Desenvolvimento do Projeto..... | 26 |
| 3.1 – Material para Utilização do Imobilizador Biométrico..... | 26 |
| 3.2 – Esquema Elétrico do Imobilizador Biométrico | 29 |
| 3.3 – Programação do Imobilizador Biométrico..... | 30 a 35 |
| 3.4 – Montagem / Teste do Imobilizador Biométrico | 36 |
| 3.5 – Ligações dos Conectores do Imobilizador Biométrico..... | 37 |
| 4 – Resultados Obtidos (Funcionamento Detalhado)..... | 39 |
| 4.1 – Resultado Obtido: Instalação do Imobilizador Biométrico na Moto | 41 |
| Conclusão | 48 |
| Referências | 49 |

Introdução

Apresentado a ideia do projeto de desenvolver o imobilizador biométrico capaz de proporcionar mais segurança ao usuário de motocicleta, foi proposto um projeto simples, porém de grande utilidade ao dia-a-dia do usuário, trazendo mais segurança ao estacionar sua moto.

Com o avanço das tecnologias nos dias atuais a tendência será cada vez mais a utilização de componentes eletrônicos para a identificação do usuário através de digitais, voz ou até mesmo da retina.

Sendo assim o presente trabalho é uma proposta para o desenvolvimento de uma nova tecnologia para motos com partida elétrica.

Tema e delimitação.

O projeto imobilizador biométrico enquadra-se na área de estudo Eletroeletrônica.

Objetivos

O projeto Imobilizador Biométrico tem como objetivo geral facilitar o dia-a-dia do usuário de moto com partida elétrica a fim de promover mais segurança ao estacionar a moto em locais perigosos ao furto do mesmo, dificultando as ações de pessoas com má índole.

A proposta do projeto é de fato a segurança e a redução dos furtos de motos.

Justificativa

A motivação para a realização deste projeto surgiu a partir de um levantamento de ideias realizado pelo grupo para o projeto de TCC. A ideia do imobilizador biométrico deu-se devido a sua viabilidade econômica, bem como a praticidade do sistema de biometria ser mais eficiente do que o sistema convencional de alarmes encontrado hoje no mercado.

Metodologia

A metodologia adotada foi método de engenharia que pretende construir um protótipo composto por itens como leitor biométrico, microcontrolador arduino, módulo relé, também por pesquisas com referencial teórico através de reuniões dos integrantes do grupo, bem como pesquisas de campo realizadas em diversas mecânicas de manutenção de motos com os seus usuários.

1 – Fundamentação Teórica

Neste capítulo apresentaremos as tecnologias utilizadas para o desenvolvimento do projeto “Imobilizador Biométrico”.

1.1 – Microcontrolador Arduíno:

O Arduino foi criado em 2005 por um grupo de 5 pesquisadores: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis. O objetivo era elaborar um dispositivo que fosse ao mesmo tempo barato, funcional e fácil de programar, sendo dessa forma acessível a estudantes e projetistas amadores. Além disso, foi adotado o conceito de hardware livre, o que significa que qualquer um pode montar, modificar, melhorar e personalizar o Arduino, partindo do mesmo hardware básico.

Assim, foi criada uma placa composta por um microcontrolador Atmel, circuitos de entrada/saída e que pode ser facilmente conectada à um computador e programada via IDE (Integrated Development Environment, ou Ambiente de Desenvolvimento Integrado) utilizando uma linguagem baseada em C/C++, sem a necessidade de equipamentos extras além de um cabo USB.



Figura 1 – MicroControlador Arduíno

Depois de programado, o microcontrolador Arduino pode ser usado de forma independente, ou seja, você pode colocá-lo para controlar um robô, uma lixeira, um ventilador, as luzes da sua casa, a temperatura do ar condicionado, pode utilizá-lo como um aparelho de medição ou qualquer outro projeto que vier à cabeça.

A lista de possibilidades é praticamente infinita. Você pode automatizar sua casa, seu carro, seu escritório, criar um novo brinquedo, um novo equipamento ou melhorar um já existente. Tudo vai depender da sua criatividade.

Para isso, o Arduino possui uma quantidade enorme de sensores e componentes que você pode utilizar nos seus projetos. Grande parte do material utilizado no Arduino está disponível em módulos, que são pequenas placas que contém os sensores e outros componentes auxiliares como resistores, capacitores e leds.

Informações técnicas:

| | |
|---|--|
| Tamanho: | 5,3cm x 6,8cm x 1,0cm |
| Microcontrolador: | ATmega328 |
| Tensão de operação: | 5 V |
| Tensão de entrada (recomendada): | 7-12 V |
| Tensão de entrada (limites): | 6-20 V |
| Pinos de entrada/saída (I/O) digitais: | 14 (dos quais 6 podem ser saídas PWM) |
| Pinos de entrada analógicas: | 6 |
| Corrente DC por pino I/O: | 40 mA |
| Corrente DC para pino de 3,3 V: | 200 mA* |
| Memória Flash: | 32 KB (dos quais 0,5KB são usados pelo <i>bootloader</i>) |
| SRAM: | 2 KB |
| EEPROM: | 1 KB |
| Velocidade de Clock: | 16 MHz |
| Temperatura de operação: | de 10 °C a 60 °C |
| Garantia de 6 meses para defeitos de fabricação | |

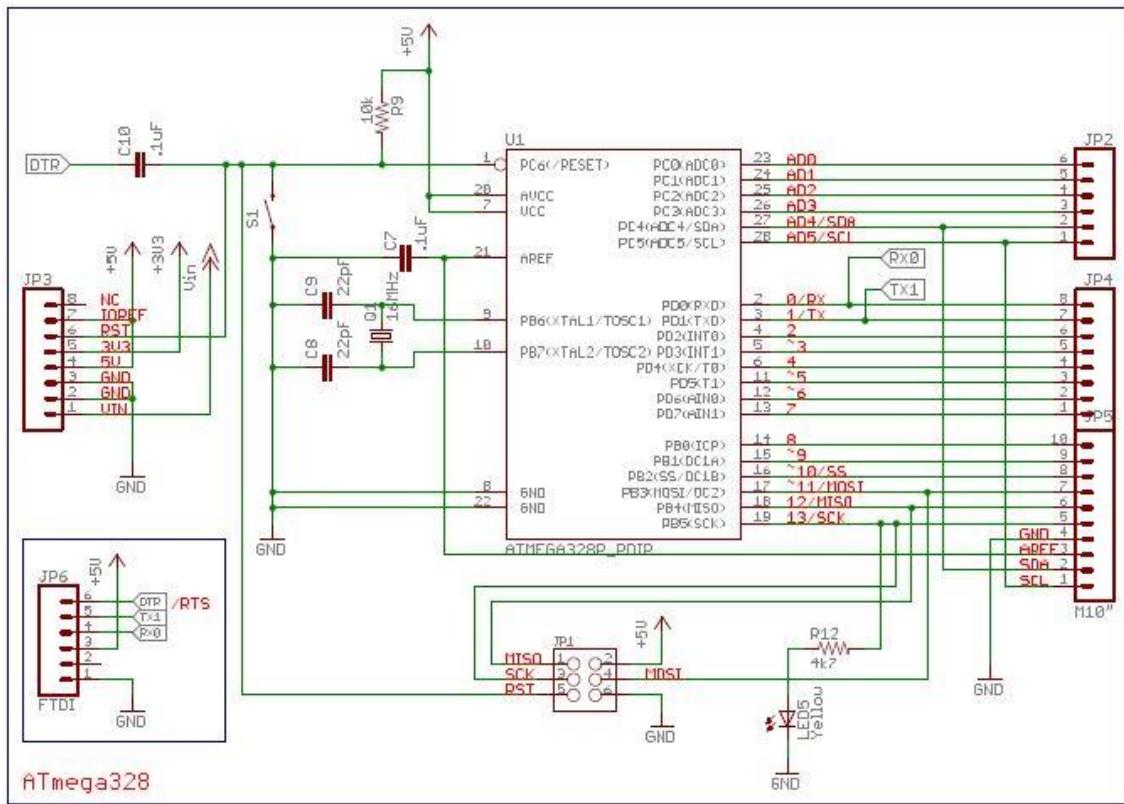


Figura 2 – Esquema elétrico do Arduino

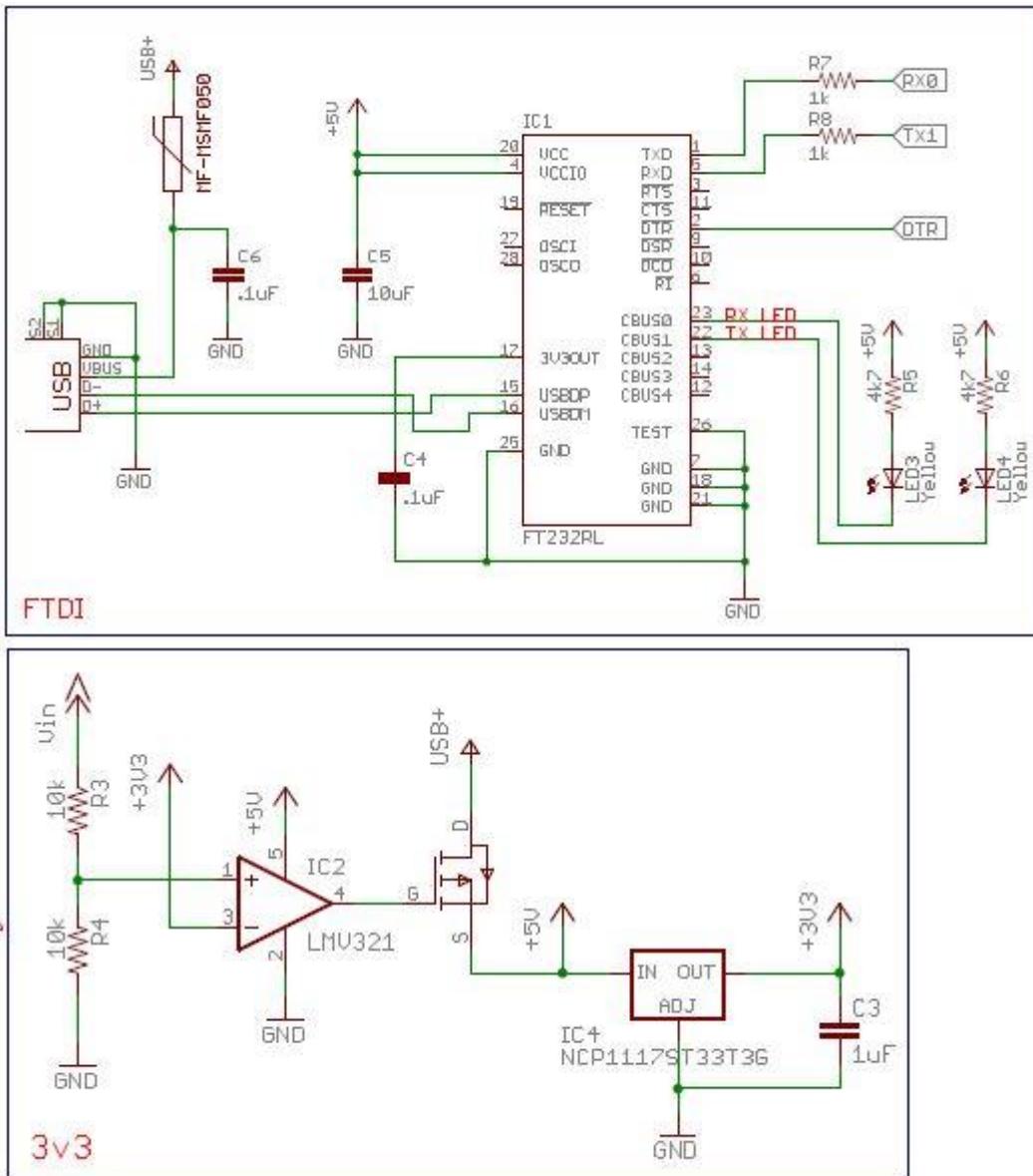


Figura 3 – Esquema elétrico do Arduino

O tipo de placa que você vai utilizar depende muito do projeto a ser desenvolvido e o número de portas necessárias. As opções vão das mais comuns, como o Arduino Uno e suas 14 portas digitais e 6 analógicas, passando por placas com maior poder de processamento, como o Arduino Mega, com microcontrolador ATmega2560 e 54 portas digitais, e o Arduino Due, baseado em processador ARM de 32 bits e 512 Kbytes de memória.

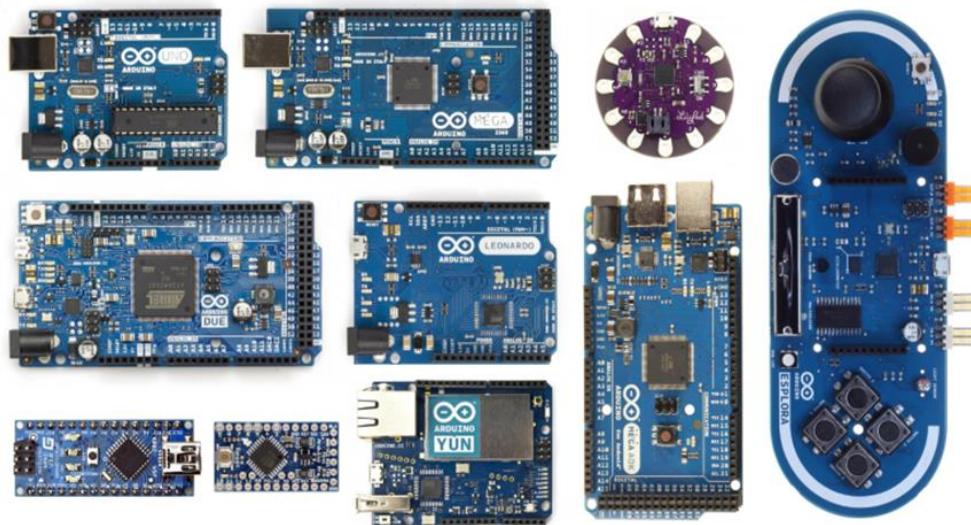
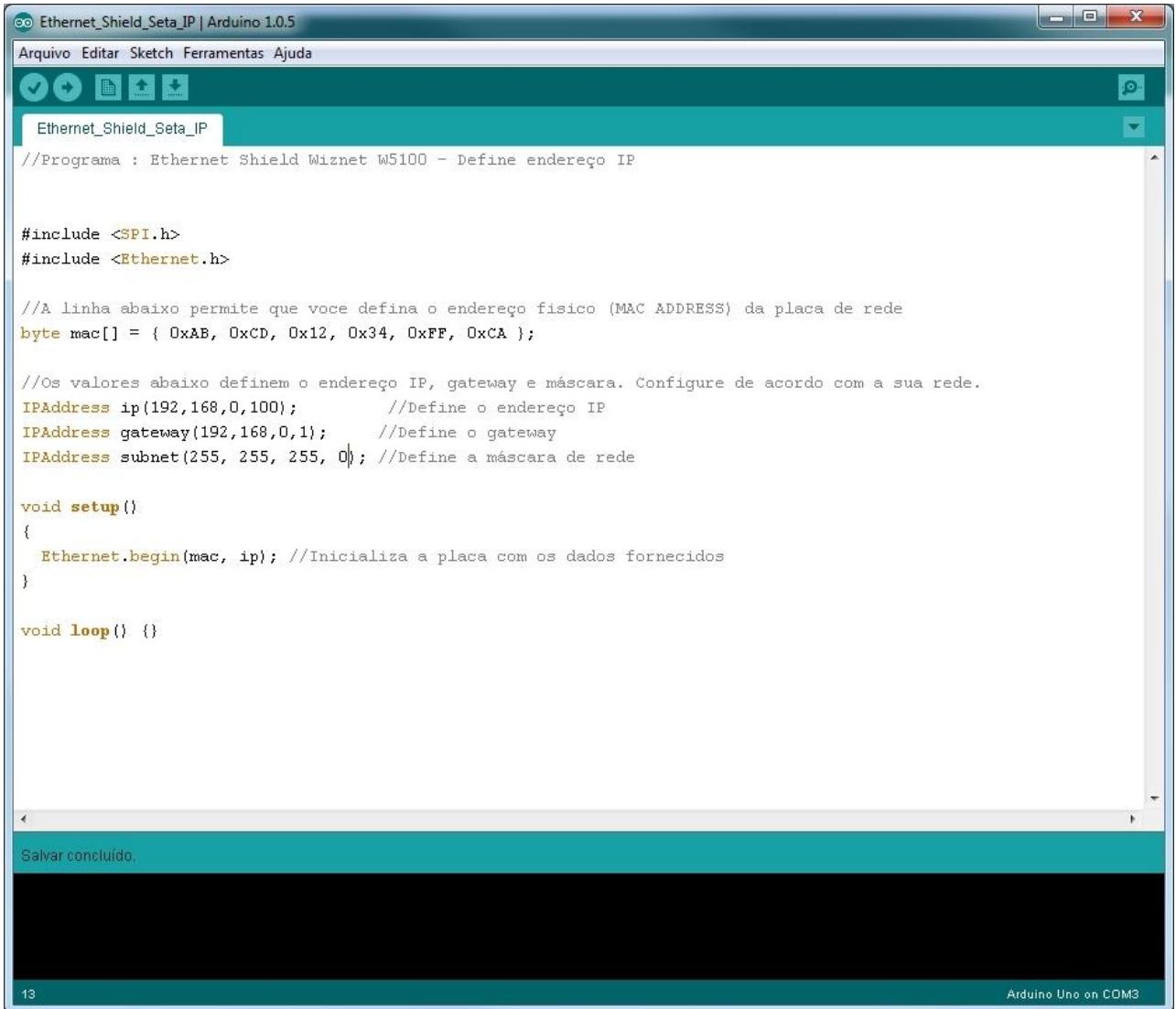


Figura 4 – Tipos de Placas de MicroControlador Arduíno

1.1.1 - Estrutura de um programa em Arduino:

Escrever um programa em Arduino é muito simples. Tudo o que você precisa é conectar o Arduino ao computador por meio de um cabo USB e utilizar um ambiente de programação chamado IDE, onde você digita o programa, faz os testes para encontrar eventuais erros e transfere o programa para o Arduino.

Na imagem abaixo temos a IDE já com um programa carregado.



The screenshot shows the Arduino IDE interface with the title bar "Ethernet_Shield_SetIP | Arduino 1.0.5". The menu bar includes "Arquivo", "Editar", "Sketch", "Ferramentas", and "Ajuda". Below the menu is a toolbar with icons for file operations. The main window displays the C++ code for the "Ethernet_Shield_SetIP" sketch. The code includes comments defining the MAC address, IP settings, and setup/loop functions. A status bar at the bottom indicates "Salvar concluido.", the board is "Arduino Uno", and the port is "COM3".

```

// Programa : Ethernet Shield Wiznet W5100 - Define endereço IP

#include <SPI.h>
#include <Ethernet.h>

//A linha abaixo permite que voce defina o endereço fisico (MAC ADDRESS) da placa de rede
byte mac[] = { 0xAB, 0xCD, 0x12, 0x34, 0xFF, 0xCA };

//Os valores abaixo definem o endereço IP, gateway e máscara. Configure de acordo com a sua rede.
IPAddress ip(192,168,0,100);           //Define o endereço IP
IPAddress gateway(192,168,0,1);         //Define o gateway
IPAddress subnet(255, 255, 255, 0);    //Define a máscara de rede

void setup()
{
  Ethernet.begin(mac, ip); //Inicializa a placa com os dados fornecidos
}

void loop() {}

```

Figura 5 – Estrutura de programa de MicroControlador Arduíno

Uma vez feito o programa, basta transferí-lo para o Arduino e o mesmo começa a funcionar.

Você não precisa ser expert em linguagem C para programar o Arduino. Você pode começar um programa utilizando a estrutura básica do Arduino, que é composta por duas partes, ou dois blocos:

setup() – É nessa parte do programa que você configura as opções iniciais do seu programa: os valores iniciais de uma variável, se uma porta será utilizada como entrada ou saída, mensagens para o usuário, etc.

loop() – Essa parte do programa repete uma estrutura de comandos de forma contínua ou até que alguma comando de “parar” seja enviado ao Arduino.

Vamos ver exatamente como isso funciona, levando em consideração o programa abaixo, que acende e apaga o led embutido na placa Arduino em intervalos de 1 segundo:

```

1 //Programa : Pisca Led Arduino
2 //Autor : FILIPEFLOP
3
4 void setup()
5 {
6     //Define a porta do led como saida
7     pinMode(13, OUTPUT);
8 }
9
10 void loop()
11 {
12     //Acende o led
13     digitalWrite(13, HIGH);
14
15     //Aguarda o intervalo especificado
16     delay(1000);
17
18     //Apaga o led
19     digitalWrite(13, LOW);
20
21     //Aguarda o intervalo especificado
22     delay(1000);
23 }
```

Figura 6 - Programação de MicroControlador Arduíno

A primeira coisa que fazemos no início do programa é colocar uma pequena observação sobre o nome do programa, sua função e quem o criou:

```

1 //Programa : Pisca Led Arduino
2 //Autor : FILIPEFLOP
```

Comece uma linha com barras duplas (//) e tudo o que vier depois dessa linha será tratado como um comentário. Uma das boas práticas de programação é documentar o seu código por meio das linhas de comentário. Com elas, você pode inserir observações sobre como determinada parte do programa funciona ou o que

significa aquela variável **AbsXPT** que você criou. Isso será útil não só para você, se precisar alterar o código depois de algum tempo, como também para outras pessoas que utilizarão o seu programa.

Após os comentários, vem a estrutura do **SETUP**. É nela que definimos que o pino 13 do Arduino será utilizado como saída.

```
4 void setup()
5 {
6     //Define a porta do led como saída
7     pinMode(13, OUTPUT);
8 }
```

Por último, temos o **LOOP**, que contém as instruções para acender e apagar o led, e também o intervalo entre essas ações:

```
10 void loop()
11 {
12     //Acende o led
13     digitalWrite(13, HIGH);
14
15     //Aguarda o intervalo especificado
16     delay(1000);
17
18     //Apaga o led
19     digitalWrite(13, LOW);
20
21     //Aguarda o intervalo especificado
22     delay(1000);
23 }
```

A linha do código contendo `digitalWrite(13, HIGH)` coloca a porta 13 em nível alto (HIGH, ou 1), acendendo o led embutido na placa. O comando `delay(1000)`, especifica o intervalo, em milisegundos, no qual o programa fica parado antes de avançar para a próxima linha.

O comando `digitalWrite(13, LOW)`, apaga o led, colocando a porta em nível baixo (LOW, ou 0), e depois ocorre uma nova parada no programa, e o processo é então reiniciado.

1.2. – Leitor Biométrico

O Leitor Biométrico para Arduíno é um sensor de impressão digital óptico desenvolvido especialmente para a criação de projetos de automação residencial, sendo capaz de realizar a leitura das impressões digitais dos dedos com o auxílio de

um chip DSP de alta potência que possui grande precisão, além de proporcionar muita simplicidade para utilização.

- Para que entre em operação, é necessário integrar o Leitor Biométrico em uma plataforma de prototipagem, entre elas, o Arduino. Quando em funcionamento o Leitor Biométrico possui a capacidade de inscrever (salvar) uma diversidade muito grande de impressões dentro de sua memória flash onboard, chegando a gravar até 500 impressões digitais.

- Além de ser fácil de usar, o Leitor Biométrico para Arduino também possui um software bastante simples, que junto de sua compatibilidade com qualquer microcontrolador ou sistema TTL, o que o torna ideal para projetos profissionais ou estudantis.

- Na prática o Leitor Biométrico para Arduino é utilizado comumente para dar maior segurança aos seus projetos, evitando a utilização de terceiros não autorizados. No caso da automação residencial é empregado para proibir a entrada não autorizada de pessoas em condomínios e residências, liberando a passagem apenas de pessoas cadastradas, ou ainda, é utilizado para dar maior segurança contra roubos em cofres.

- O princípio de funcionamento do Leitor Biométrico é simples, primeiramente ele faz a rapidamente a leitura das impressões digitais comparando as informações com o seu banco de dados, em seguida, caso a pessoa seja reconhecida, enviará sinais ao Arduino, se for o caso, para que libere a passagem.

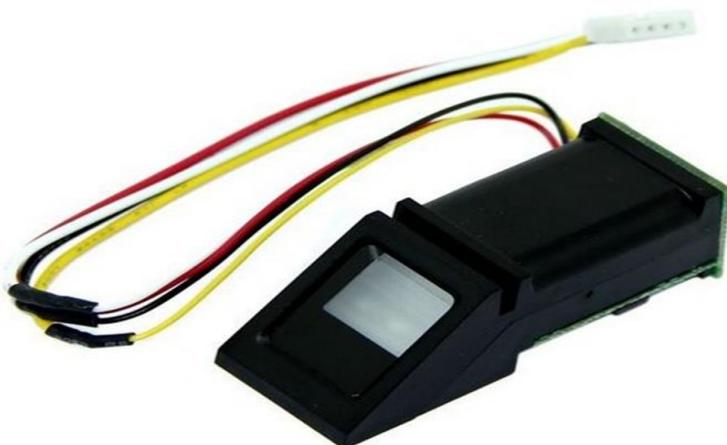


Figura 7 – Leitor Biométrico

1.2.1. – Especificações:

- Modelo: LBA35;
- Tensão de alimentação DC: 3,6 a 6V;
- Corrente de operação: <120mA;
- Corrente de pico máx: 150mA;
- Tempo de impressão digital de imagem: <1,0 segundo;
- Capacidade de armazenamento on-board: 500 digitais;
- Nível de segurança configurável: 1-baixo, 3-médio, 5-alto;
- Taxa de aceitação falsa: <0,001% (nível de segurança 3);
- Taxa de rejeição falsa: <1,0% (nível de segurança 3);
- Interface: UART (Serial TTL);
- Taxa de transmissão: 9600, 19200, 28800, 38400, 57600 (o padrão é 57600);
- Temperatura de operação: -20°C a +50°C;
- Umidade de trabalho: <85%;
- Comprimento do cabo: 150mm;
- Área da janela de leitura: 14x18mm;
- Dimensões totais: 55x 21x21mm;
- Peso: 20g.

1.2.2 – Características:

- Leitor Biométrico para Arduino;
- Sensor de impressão digital óptico;
- Chip DSP;
- Produto de alta potência e precisão;
- Simplicidade e facilidade incomparável;
- Chega a gravar 120 impressões digitais;
- Memória Flash Onboard;
- Compatível com qualquer microcontrolador;
- Compatível com sistema TTL;
- Garante segurança e conforto;
- Leitura diferenciada e própria;
- Método rápido, confiável;
- Inclui cabo para conexão;

1.3. – Módulo Relé:

Este Módulo Relé tem uma ampla gama de microcontroladores como Arduino, AVR, PIC, ARM. A partir das saídas digitais pode-se, através do relé, controlar cargas maiores e dispositivos como motores AC/DC, eletroímãs, solenóides, lâmpadas incandescentes e eletrodomésticos por exemplo.

Este módulo elimina todo aquele trabalhoso circuito de acionamento com transistores, relés, conectores, leds e diodos. Se precisar de mais canais de acionamento.

1.3.1. - Informações técnicas:

- Tensão de alimentação: 12V
- Tensão de Clock/Data: 5V
- Sinal de controle: Nível TTL
- Bobina: 12VDC 30mA
- Carga nominal do relê: 7A 30VDC, 10A 125VAC , 7A 220VAC
- Carga nominal do módulo: 10^a
- Tempo de acionamento de contato: 10ms

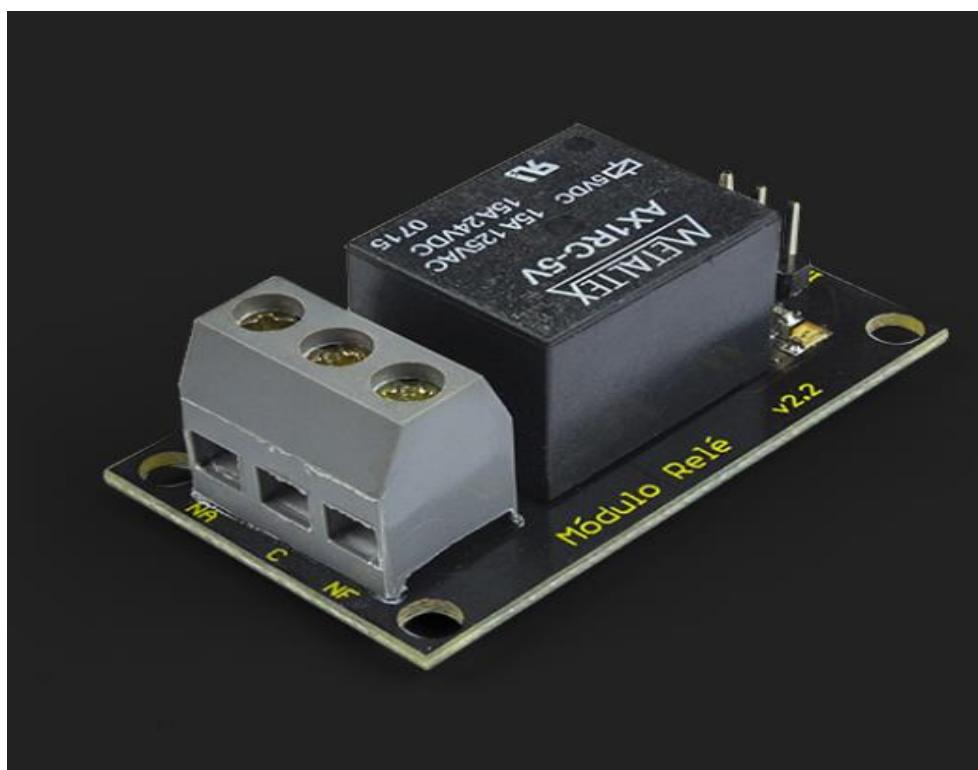


Figura 8 – Módulo Relé

2 – Planejamento do Projeto

Este capítulo apresenta como foi todo o decorrer do projeto, desde o início, contendo a parte elétrica/eletromecânica, a parte lógica e a parte mecânica do projeto, além é claro do cronograma feito pelo grupo para organizar melhor o trabalho.

2.1. - Parte eletroeletrônica:

Neste capítulo é explicado todo o funcionamento da parte elétrica do projeto, contendo o circuito e suas entradas e saídas.

2.1.1.- Diagrama em Blocos

O diagrama em blocos mostra todas as entradas e saídas que contém em nosso projeto e por onde passam essas entradas e saídas, como podemos ver no esquema abaixo:

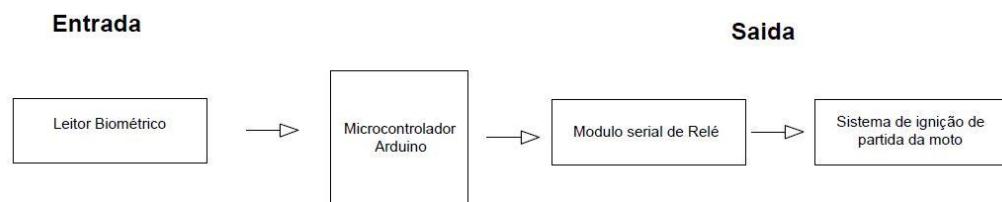


Figura 9 – Diagrama em Blocos

2.1.2. - Pesquisa de Componentes/Tecnologias

O grupo realizou as pesquisas dos componentes em lojas de produtos eletroeletrônicos, bem como em lojas de robóticas e sites.

2.1.3. - Previsão de Custos

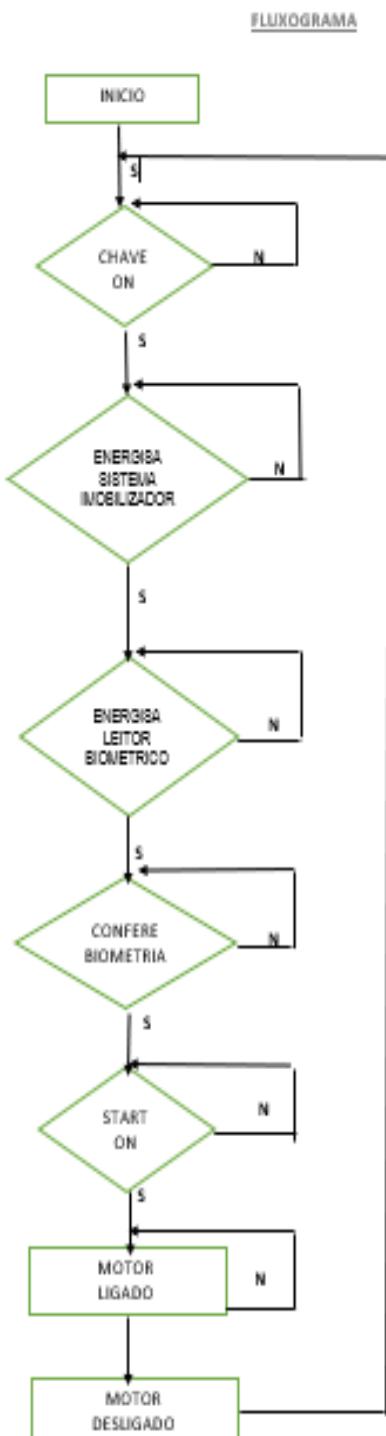
A previsão de custos realizada foi conforme tabela.

| Componentes | Valor R\$ |
|---|---------------|
| BlackBoard V1.0 | 85,00 |
| Módulo Relé | 12,90 |
| Pacote com 10 Jumper Premium | 10,00 |
| Sensor Leitor Biométrico De Impressão Digital | 229,00 |
| Case de proteção para o Leitor Biométrico | 30,00 |
| Case de proteção do sistema imobilizador | 70,00 |
| Diodos normal | 2,00 |
| Mão-de-Obra (elétrica de moto) | 50,00 |
| TOTAL | 488,90 |

2.2. - Parte Lógica:

- Fluxograma do Processo

Apresentamos o fluxograma do processo do Imobilizador Biométrico:



2.3. - Parte Mecânica:

- Desenho:

Leitor Biométrico.

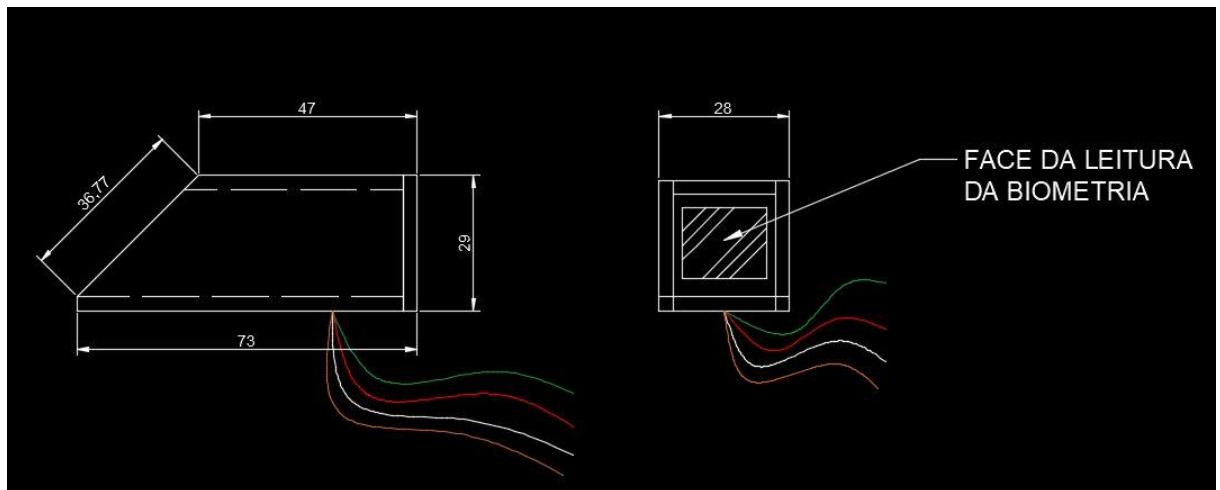


Figura 10 – Desenho Leitor Biométrico no Case

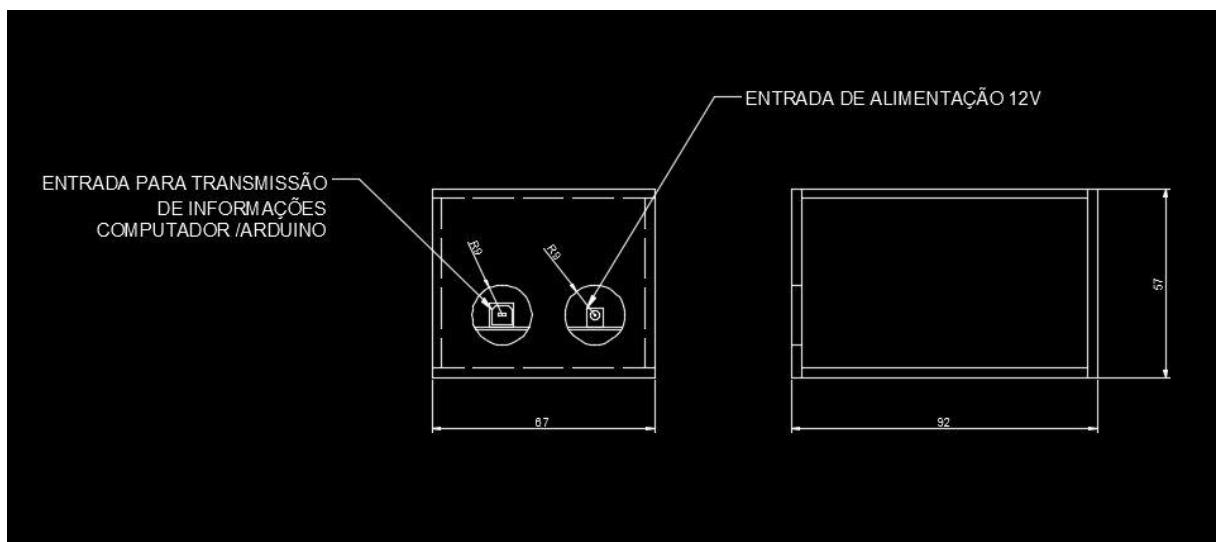


Figura 11 – Desenho do Sistema Imobilizador

- Pesquisa de Material:

A pesquisa de material utilizado no processo do Imobilizador Biométrico foi realizada em lojas especializadas e em sites. Efetuamos a compra de todos os materiais necessários para a montagem do imobilizador biométrico.

- Previsão de Custos:

A previsão de custos realizada na parte mecânica do projeto para a instalação do imobilizador biométrico são:

| Componentes | Valor R\$ |
|---|---------------|
| Case de proteção para o Leitor Biométrico | 30,00 |
| Case de proteção do sistema imobilizador | 70,00 |
| Mão-de-obra (elétrica da moto) | 50,00 |
| TOTAL | 150,00 |

2.4. - Cronograma Inicial

O projeto teve o seguinte cronograma inicial realizado.

| ATIVIDADE | 24/jul | 31/jul | 07/ago | 21/ago | 04/set | 25/set | 16/out | 30/out | 06/nov | 13/out | 27/nov |
|-----------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| PESQUISA DO TEMA | ■ | ■ | | | | | | | | | |
| COLETA DE DADOS | | | | | | | | | | | |
| LISTA DE PEÇAS PREVISÃO DE CUSTOS | | | | | ■ | | | | | | |
| ESQUEMA ELETRICO | | | | | | ■ | ■ | ■ | ■ | | |
| MONOGRAFIA | | | | | | ■ | | | | | |
| LAY-OUT | | | | | | ■ | ■ | ■ | ■ | | |
| APRESENTAÇÃO | | | | | | ■ | ■ | ■ | ■ | ■ | ■ |

LEGENDA

| | |
|------|--------------|
| ■ | ENTREGUE |
| ■■ | ATRASADO |
| ■■■ | PRAZO |
| ■■■■ | DATA ENTREGA |

2.5. - Cronograma Final:

O projeto do imobilizador biométrico concluiu o cronograma final.

| ATIVIDADE | 05/mar | 19/mar | 09/abr | 23/abr | 07/mai | 14/mai | 21/mai | 28/mai | 04/jun | 08/jun |
|--|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| MONTAGEM / TESTE | | | | | | | | | | |
| AQUISIÇÃO DE CASES | | | | | | | | | | |
| PROGRAMAÇÃO FINAL | | | | | | | | | | |
| PESQUISA DE MAO-DE-OBRA | | | | | | | | | | |
| LIGAÇÕES DOS CONECTORES DO IMOBILIZADOR BIOMETRICO | | | | | | | | | | |
| MONTAGEM FINAL | | | | | | | | | | |
| MONOGRAFIA FINAL | | | | | | | | | | |

| LEGENDA | |
|--------------|--|
| ENTREGUE | |
| ATRASADO | |
| PRAZO | |
| DATA ENTREGA | |

3 – Desenvolvimento do Projeto.

Concluímos a fase da aquisição dos itens leitor biométrico, microcontrolador arduino, módulo relé, os cases e os conectores.

Realizamos a montagem / testes, a programação final e as ligações dos conectores do imobilizador biométrico conforme as imagens que apresentaremos em ordem sequencial.

3.1 – Material para utilização do imobilizador biométrico:

- ✓ Leitor Biométrico instalado no Case
- ✓ Arduíno fixo no case.
- ✓ Relé fixo na tampa do case.



Figura 12 - Leitor Biométrico instalado no Case.



Figura 13 - Arduíno fixo no case.



Figura 14 – Relé fixo na tampa do case.

3.2. – Esquema Elétrico do Imobilizador Biométrico.

Apresentamos abaixo o esquema elétrico do Imobilizador Biométrico.

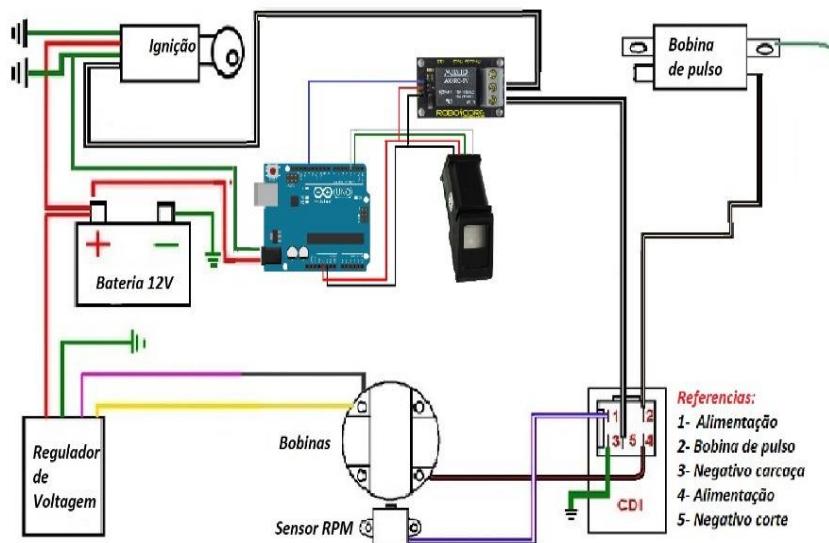


Figura 15 – Esquema Elétrico do Imobilizador Biométrico

3.3. – Programação do Imobilizador Biométrico:

Mostraremos nas imagens abaixo a programação do imobilizador biométrico em ordem sequencial nas figuras de 16 a 16.5.

```
finalleitor

// Programa para teste de funcionamento do Leitor Biometrico
// Traduzido e adaptado por Usinainfo

#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>

int getFingerprintIDez();
int sinalparaorele = 13;

SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup() {
    Serial.begin(9600);
    Serial.println("Iniciando Leitor Biometrico");
    pinMode(11, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(sinalparaorele, OUTPUT);

    finger.begin(57600);

    if (finger.verifyPassword()) {
        Serial.println("Leitor Biometrico Encontrado");
    } else {
        Serial.println("Leitor Biometrico nao encontrada");
        while (1);
    }
}
```

Figura 16 – Programação do Imobilizador Biométrico

```
finalleitor

if (finger.verifyPassword()) {
    Serial.println("Leitor Biometrico Encontrado");
} else {
    Serial.println("Leitor Biometrico nao encontrada");
    while (1);
}
Serial.println("Esperando Dedo para Verificar");
}

void loop()
{
    getFingerprintIDez();
    digitalWrite(12, HIGH);
    delay(50);
}

uint8_t getFingerprintID() {
    uint8_t p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Imagen Capturada");
            break;
        case FINGERPRINT_NOFINGER:
            Serial.println("Dedo nao Localizado");
            return p;
        case FINGERPRINT_PACKETRECIEVEERR:
            Serial.println("Erro ao se comunicar");
            return p;
    }
}
```

Figura 16.1 – Programação do Imobilizador Biométrico

```
finalleitor
switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Imagen Capturada");
        break;
    case FINGERPRINT_NOFINGER:
        Serial.println("Dedo nao Localizado");
        return p;
    case FINGERPRINT_PACKETRECEIVEERR:
        Serial.println("Erro ao se comunicar");
        return p;
    case FINGERPRINT_IMAGEFAIL:
        Serial.println("Erro ao Capturar");
        return p;
    default:
        Serial.println("Erro desconhecido");
        return p;
}

p = finger.image2Tz();
switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Imagen Convertida");
        break;
    case FINGERPRINT_IMAGEMESS:
        Serial.println("Imagen muito confusa");
        return p;
    case FINGERPRINT_PACKETRECEIVEERR:
        Serial.println("Erro ao se comunicar");
    <
```

Figura 16.2 – Programação do Imobilizador Biométrico

```
finalleitor

p = finger.image2Tz();
switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Imagen Convertida");
        break;
    case FINGERPRINT_IMAGEMESS:
        Serial.println("Imagen muito confusa");
        return p;
    case FINGERPRINT_PACKETRECEIVEERR:
        Serial.println("Erro ao se comunicar");
        return p;
    case FINGERPRINT_FEATUREFAIL:
        Serial.println("Impossivel localizar Digital");
        return p;
    case FINGERPRINT_INVALIDIMAGE:
        Serial.println("Impossivel Localizar Digital");
        return p;
    default:
        Serial.println("Erro Desconhecido");
        return p;
}

p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
    Serial.println("Digital Encontrada");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Erro ao se comunicar");
    return p;
}
```

Figura 16.3 – Programação do Imobilizador Biométrico

```
finalleitor  
}  
  
p = finger.fingerFastSearch();  
if (p == FINGERPRINT_OK) {  
    Serial.println("Digital Encontrada");  
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {  
    Serial.println("Erro ao se comunicar");  
    return p;  
} else if (p == FINGERPRINT_NOTFOUND) {  
    Serial.println("Digital Desconhecida");  
    return p;  
} else {  
    Serial.println("Erro Desconhecido");  
    return p;  
}  
  
Serial.print("ID # Encontrado");  
Serial.print(finger.fingerID);  
Serial.print(" com precisao de ");  
Serial.println(finger.confidence);  
}  
  
int getFingerprintIDez() {  
    uint8_t p = finger.getImage();  
    if (p != FINGERPRINT_OK)  return -1;  
  
    p = finger.image2Tz();  
    if (p != FINGERPRINT_OK)  return -1;  
    <
```

Figura 16.4 – Programação do Imobilizador Biométrico

```
finalleitor

int getFingerprintIDez() {
    uint8_t p = finger.getImage();
    if (p != FINGERPRINT_OK)  return -1;

    p = finger.image2Tz();
    if (p != FINGERPRINT_OK)  return -1;

    p = finger.fingerFastSearch();
    if (p != FINGERPRINT_OK)  return -1;

    digitalWrite(sinalparaorele, HIGH);

    digitalWrite(12, LOW);
    digitalWrite(11, HIGH);
    delay(1000);
    digitalWrite(11, LOW);
    delay(1000);
    digitalWrite(12, HIGH);
    Serial.print("ID # Encontrado");
    Serial.print(finger.fingerID);
    Serial.print(" com precisao de ");
    Serial.println(finger.confidence);
    return finger.fingerID;
}

<
```

Figura 16.5 – Programação do Imobilizador Biométrico

3.4. Montagem / Teste do Imobilizador Biométrico:

Após a programação do imobilizador biométrico partimos para a sua montagem / teste de funcionamento conforme imagens abaixo.

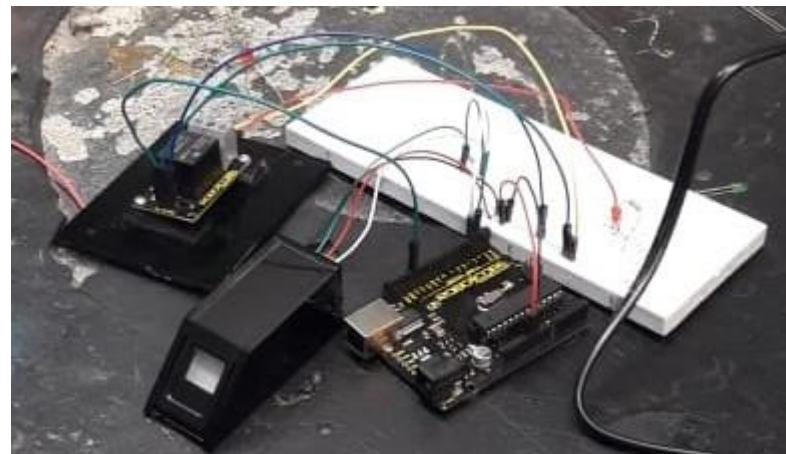


Figura 17 – Montagem / Teste do Imobilizador Biométrico

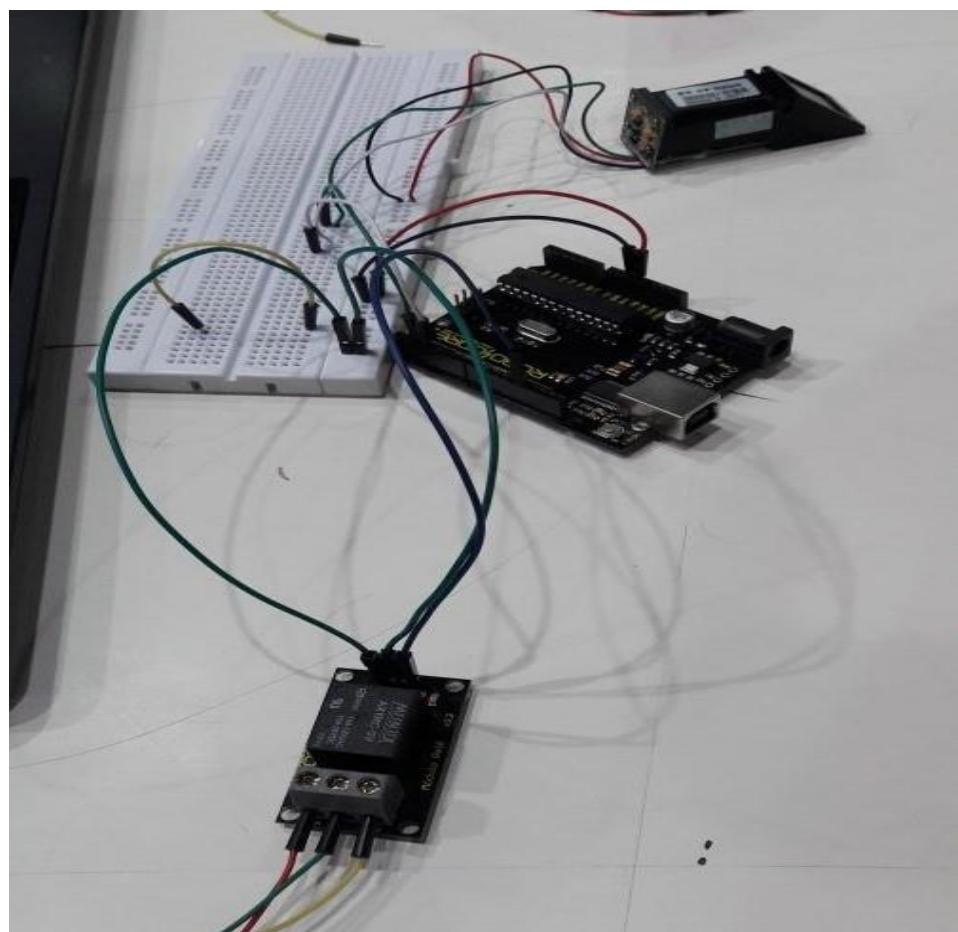


Figura 17.1 – Montagem / Teste do Imobilizador Biométrico

3.5. – Ligações dos Conectores do Imobilizador Biométrico.

Concluída a montagem do imobilizador biométrico passamos para as ligações dos conectores conforme as imagens abaixo.



Figura 18 – Ligação dos Conectores



Figura 18.1 – Ligação dos Conectores

4 – Resultados Obtidos: Funcionamento Detalhado.

O resultado obtido nesta fase do projeto foram de que os componentes leitor biométrico, microcontrolador arduino e módulo relé estão funcionando de acordo com o esperado e com isso obtivemos como resultado, de que o acionamento por impressão digital correspondeu bem aos testes efetivados.

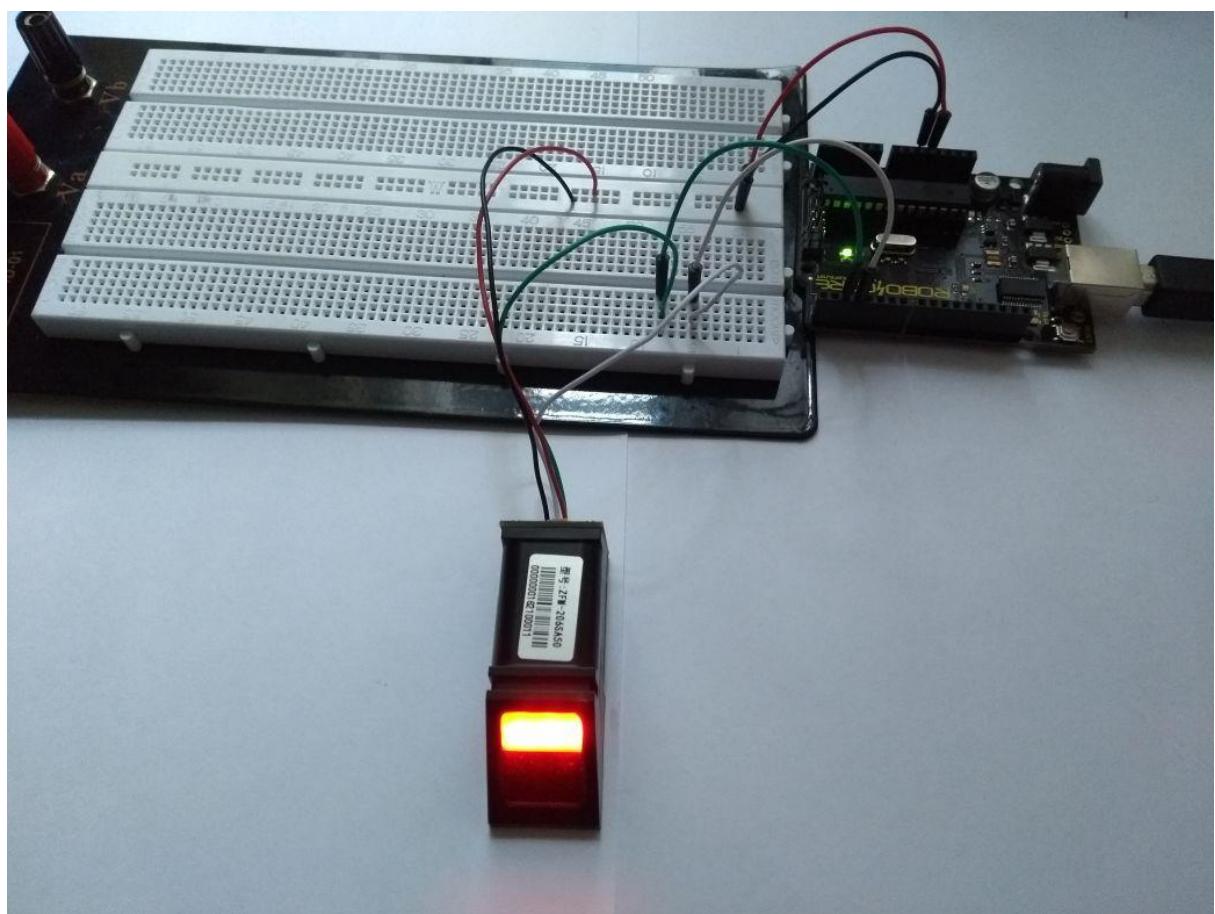


Figura 19 – Esquema de Ligação Imobilizador Biométrico

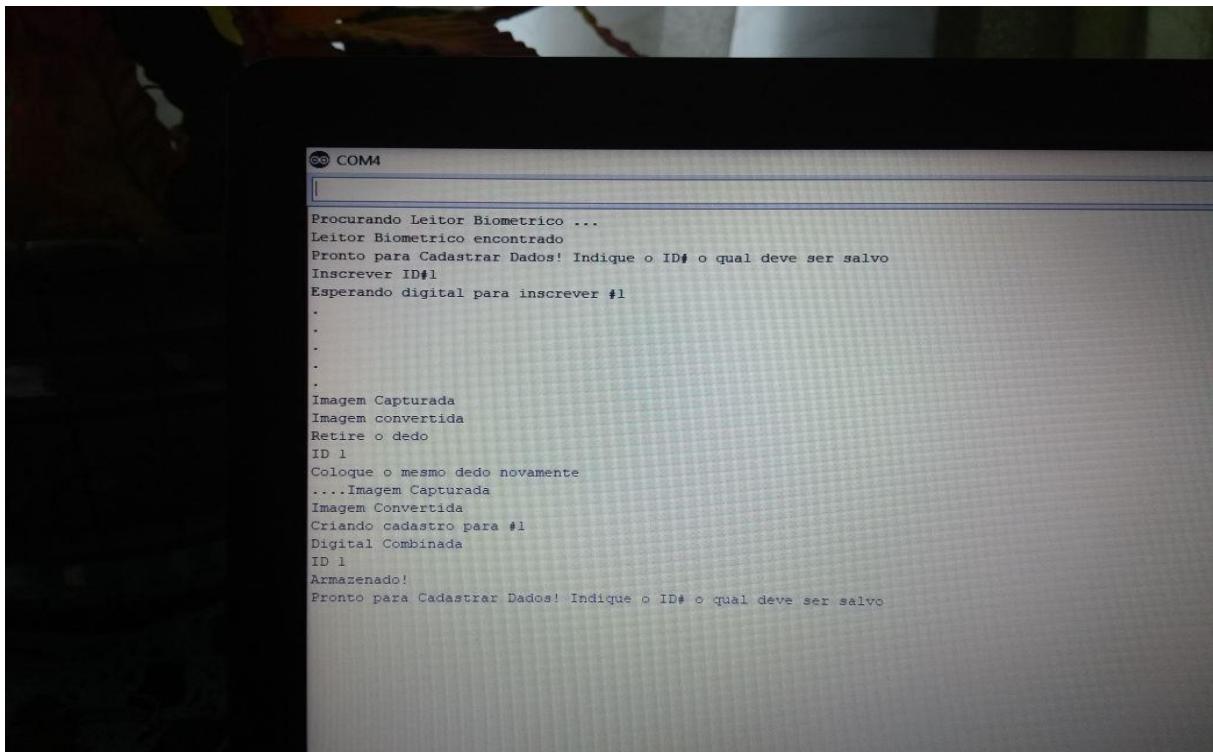


Figura 20 – Teste do Cadastro da Digital do Imobilizador Biométrico

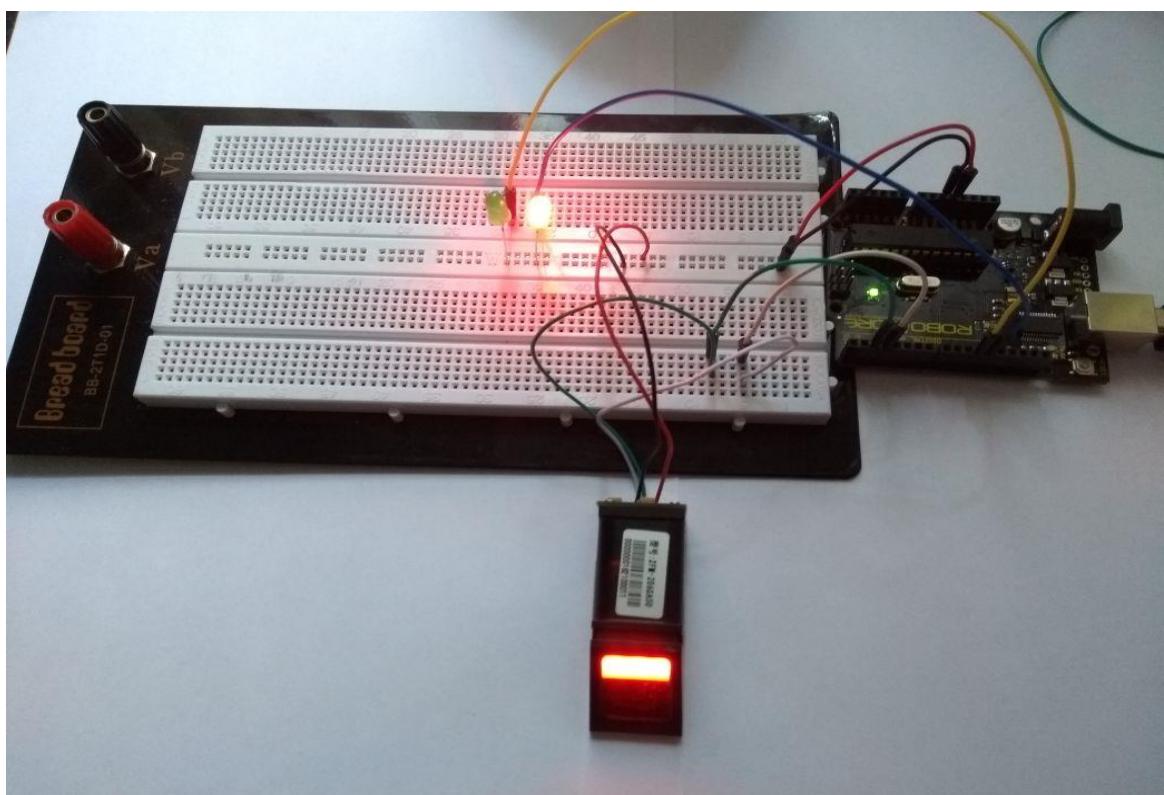


Figura 21 – Digital não apresentada não libera o pulso elétrico do Imobilizador Biométrico

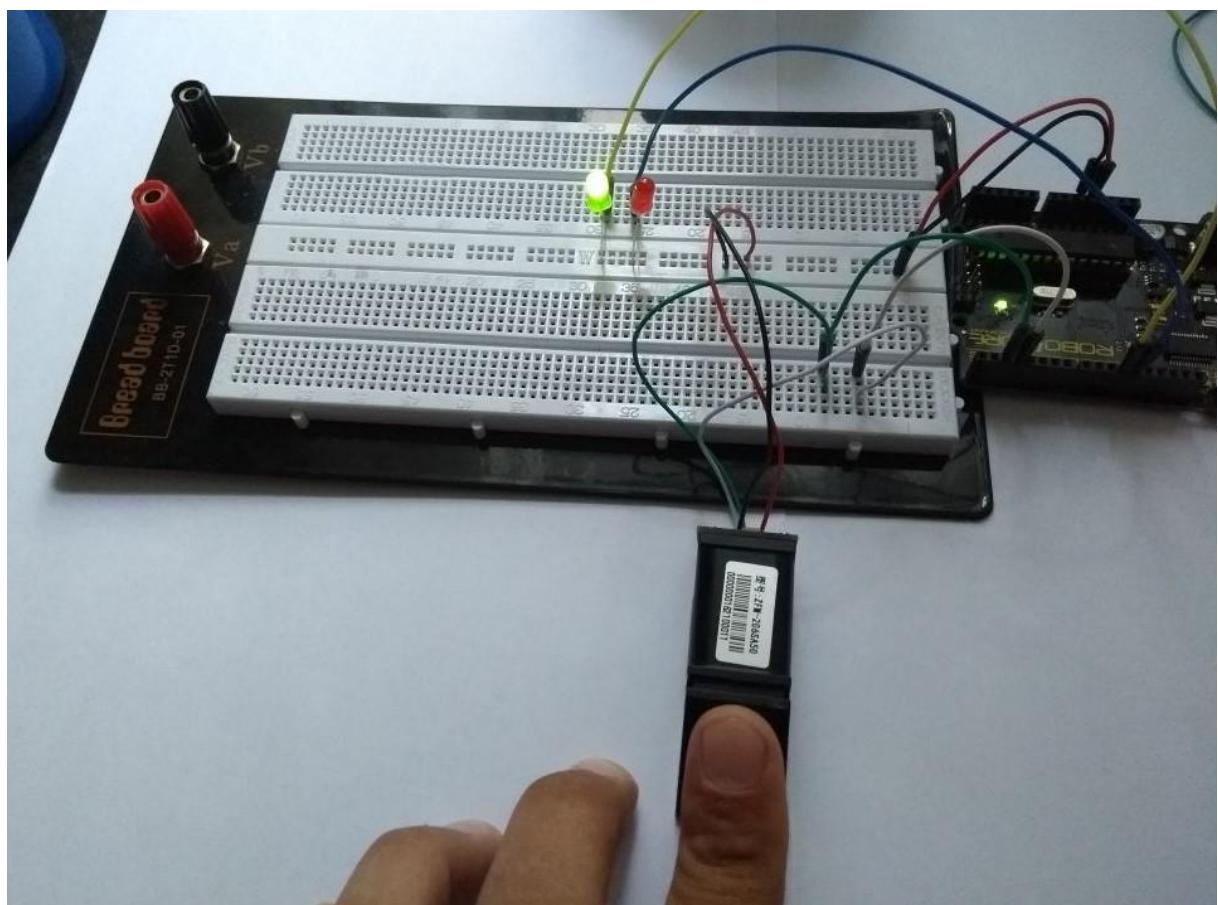


Figura 22 – Liberação do pulso elétrico representado pelo led do Imobilizador Biométrico após a digital cadastrada apresentada.

4.1 – Resultados Obtidos: Funcionamento Detalhado / Instalação do Imobilizador Biométrico na Moto.

Após os testes realizados nos componentes e a obtenção do resultado esperado, partimos para a última fase do projeto; a instalação do imobilizador biométrico na parte elétrica da moto. Com isso vamos concretizar todas as fases do projeto na prática podendo comprovar que a ignição da moto será ativada após o reconhecimento da digital, conforme as imagens anexas abaixo.



Figura 23 – Localização do Leitor Biométrico



Figura 24 – Moto ligada e leitor biométrico energizado.



Figura 25 – Verificação da digital cadastrada.



Figura 26 – Relé de partida acionado após verificação da digital.



Figura 27 – Liberação da partida da moto.



Figura 28 – Moto ligada.

Conclusão

Com o presente trabalho conclui-se que o projeto do Imobilizador Biométrico desde a fase de pesquisa até a fase final desenvolveu-se conforme o esperado.

Em nosso projeto falamos do Imobilizador biométrico, sendo o funcionamento através da ativação após o reconhecimento da digital cadastrada, da seguinte forma, em primeiro lugar, a moto não está ligada mesmo após virar a chave e acionar o botão de ignição.

Próximo passo é ligar a chave novamente, com isso o sistema é energizado entrando em funcionamento. Na frente da moto no painel localiza-se o leitor biométrico no qual ficará piscando um led, onde ocorre o reconhecimento da digital cadastrada. Após esse reconhecimento acenderá um led localizado no relé do sistema. Em seguida pode ser feito o acionamento de ignição da moto que ocorre a liberação da partida.

Toda vez que for dada a partida terá que ser feito o reconhecimento da digital cadastrada, caso contrário a moto não entrará em funcionamento.

Esse tipo de tecnologia tem a vantagem de ser integrado com outros sistemas de segurança e ter baixa vulnerabilidade a falhas, tornando a moto mais segura contra furtos e roubos.

Referências

Sites Consultados:

Microcontrolador Arduíno: <https://www.filipeflop.com/blog/o-que-e-arduino/>

Leitor Biométrico: <https://www.usinainfo.com.br/outros-modulos-arduino/leitor-biometrico-para-arduino-2816.html>

Módulo Relé Serial: <https://www.filipeflop.com/produto/modulo-rele-5v-4-canais/>

Módulo Relé Serial: https://www.robocore.net/loja/produtos/modulo-rele-serial.html?gclid=Cj0KCQiAgNrQBRC0ARIsAE-m-1yb1vyOO6i_N3-udWrEkwaOPCopiyH8NTIF9J_4W3urK3mpnl8a9YEcAq4tEALw_wcB#inf_tecnicas

Módulo Relé Serial: https://www.robocore.net/loja/produtos/modulo-rele-serial.html?gclid=Cj0KCQiAgNrQBRC0ARIsAE-m-1yb1vyOO6i_N3-udWrEkwaOPCopiyH8NTIF9J_4W3urK3mpnl8a9YEcAq4tEALw_wcB#inf_tecnicas