



GOVERNO DO ESTADO DO RIO DE JANEIRO
SECRETARIA DE ESTADO DE CIÊNCIA, TECNOLOGIA E INOVAÇÃO - SECTI
FUNDAÇÃO DE APOIO À ESCOLA TÉCNICA - FAETEC
ESCOLA TÉCNICA ESTADUAL FERREIRA VIANA - ETEFV
COORDENAÇÃO DE ELETRÔNICA

Trava Automática por Sequência de Toques (TAST)

Rio de Janeiro

2024

Luan Souza Amorim
Otávio Marques Corrêa
Ramon Gabriel Sossa Silva

Trava Automática por Sequência de Toques

Trabalho de Conclusão de Curso apresentado à ETE Ferreira Viana, da Fundação de apoio à Escola Técnica, como requisito parcial para a obtenção da habilitação profissional de Técnico de Nível Médio em Eletrônica sob a orientação do Professor Luis Henrique Monteiro de Castro e do Professor Marcelo de Almeida Duarte.

Rio de Janeiro
2024

Luan Souza Amorim
Otávio Marques Corrêa
Ramon Gabriel Sossa Silva

Trava Automática por Sequência de Toques

Prof. Luis Henrique M. de Castro
ETE Ferreira Viana
Orientador

Prof. Marcelo de Almeida Duarte
ETE Ferreira Viana
Orientador

Aprovada em : ____ / ____ / ____ Conceito: _____

Prof. XXXXXXXXXXXXXXXX
ETE Ferreira Viana
ID:

Prof. XXXXXXXXXXXXXXXX
ETE Ferreira Viana
ID:

Prof. XXXXXXXXXXXXXXXX
ETE Ferreira Viana
ID:

Rio de Janeiro
2024

RESUMO

AMORIM, Luan Souza; SILVA, Ramon Gabriel Sossa; CORRÊA, Otávio Marques. Trava Automática por Sequência de Toques, 2024. Trabalho de Conclusão de Curso - Curso Técnico em Eletrônica, Escola Técnica Estadual Ferreira Viana, Fundação de Apoio à Escola Técnica, Rio de Janeiro, 2024.

A trava automática por sequência de toques surge como uma inovação prática no campo da automação residencial. Utilizando tecnologia de reconhecimento acústico, essa fechadura permite a abertura de portas por meio de comandos sonoros ou padrões acústicos específicos. Nosso projeto é aumentar a segurança residencial utilizando o arduino como microcontrolador, aumentando a conveniência e eliminando a necessidade de chaves físicas. Ela pode ser integrada em sistemas de automação residencial, proporcionando uma experiência mais conectada, moderna e acessível.

Palavras-chave: Fechadura automática; Reconhecimento acústico; Automação residencial; Fechadura por toques; Conveniência residencial;

ABSTRACT

AMORIM, Luan Souza; SILVA, Ramon Gabriel Sossa; CORREA, Otávio Marques. Fechadura Inteligente por Som, 2024. Course Completion Work - Electronics Technical Course, Escola Técnica Estadual Ferreira Viana, Fundação de Apoio à Escola Técnica, Rio de Janeiro, 2024.

The automatic touch sequence lock emerges as a practical innovation in the field of home automation. Using acoustic recognition technology, this lock allows doors to be opened through sound commands or specific acoustic patterns. Our project is to increase home security using Arduino as a microcontroller, increasing convenience and eliminating the need for physical keys. It can be integrated into home automation systems, providing a more connected, modern and accessible experience.

Keywords: Automatic lock; Acoustic recognition; Home automation; Touch lock ; Residential convenience.

SUMÁRIO

RESUMO.....	5
ABSTRACT.....	6
SUMÁRIO.....	7
1. INTRODUÇÃO.....	8
2. METODOLOGIA.....	9
3. DESENVOLVIMENTO.....	16
4. CONCLUSÃO.....	22
REFERÊNCIAS.....	23

1. INTRODUÇÃO

A automação residencial tem despertado interesse e inovação como resultado das mudanças tecnológicas que impactam o dia a dia das pessoas. As Travas automáticas por sequência de toques são uma inovação significativa neste contexto, oferecendo uma solução útil e segura para o acesso a ambientes residenciais. Estas fechaduras usam sensores táteis que permitem a abertura de portas por meio de toques específicos que o usuário define.

As Travas automáticas por sequência de toques não exigem conectividade com a Internet das Coisas (IoT), o que se distingue de outras opções disponíveis. Essa diferença torna o uso mais fácil e ajuda a atender à demanda crescente por dispositivos que combinam tecnologia sofisticada com facilidade de uso e independência de redes externas. O problema de pesquisa que este trabalho se propõe a investigar é: como as fechaduras inteligentes por sequência de toques podem otimizar a segurança e a conveniência dos lares modernos, especialmente em um cenário onde a dependência tecnológica pode ser vista tanto como uma vantagem quanto como uma vulnerabilidade? A análise das características e dos efeitos das Travas automáticas por sequência de toques no cotidiano dos usuários é um dos objetivos gerais deste estudo para responder a essa questão.

Em particular, pretende-se fazer avaliações sobre a facilidade de uso, a segurança e a receptividade dos clientes a esta tecnologia, além de avaliar as perspectivas de desenvolvimento futuro desses dispositivos. A escolha desse tema justifica-se pela importância crescente das soluções de automação residencial, bem como pela necessidade de estudos que avaliem a eficiência e a acessibilidade de tecnologias emergentes, como travas automáticas por sequência de toque. O desenvolvimento de produtos mais adaptados às necessidades dos consumidores requer uma compreensão dos elementos que influenciam a adoção e o sucesso dessas tecnologias.

Como previsto, conseguimos os equipamentos necessários para montar o nosso circuito de teste e estamos desenvolvendo o código ideal para que a sequência desejada seja qualquer uma, sem interferência ou mínima interferência de ruídos. Além disso o design já foi criado, com proposta de ser versátil e rígido para que não haja interferência humana, ao mesmo tempo, que possa ser facilmente instalado e desinstalado na fechadura

2. METODOLOGIA

Para desenvolver uma trava automática por sequência de toques, precisaríamos considerar diversos componentes e etapas de desenvolvimento, desde hardware até o software. Aqui estão os elementos principais necessários para criar uma fechadura desse tipo, bem como uma explicação de cada um:

Sensores Táteis:

- **Descrição:** Sensores capazes de detectar toques e suas sequências.
- **Componente(s):** sensor piezo ou shield de microfone.
- **Função:** Capturar os toques realizados na superfície da fechadura e transmitir essas informações para o sistema de controle.

Microcontrolador:

- **Descrição:** Um pequeno computador que controla os componentes eletrônicos da fechadura.
- **Componente(s):** arduino UNO ou derivados dele (nano, mega, etc).
- **Função:** Processar os sinais dos sensores táteis e comparar a sequência de toques com a sequência pré-configurada para autorizar ou negar o acesso.

Fonte de Energia:

- **Descrição:** Bateria.
- **Componente(s):** bateria 9V.
- **Função:** Fortalecer a alimentação de todos os componentes eletrônicos da fechadura e do microcontrolador.

Mecanismo de Trava:

- **Descrição:** Mecanismo físico que abre e fecha a fechadura.
- **Componente(s):** motor de redução de engrenagem.
- **Função:** Acionar a trava física da porta com base na autorização do microcontrolador.

Mecanismo de gravar a sequência de toques:

- **Descrição:** mecanismo físico (botão) para a gravação dos toques.
- **Componente(s):** botão.
- **Função:** ao segurá-lo, o sensor irá gravar a sequência de toques.

Mecanismo para indicação do sistema:

- **Descrição:** LEDs que sinalizam, seja um toque detectado, se o mecanismo de destravamento está acionado, ou em modo de configuração.
- **Componente(s):** 2 leds.
- **Função:** 1 led acende para sinalizar os toques, o acionamento do mecanismo de destravamento/travamento estiver ligado, se a sequência foi executada corretamente. O outro led acende quando está em modo de configuração ou quando a uma nova sequência foi armazenada.

Contestando:**Em que lugares nós poderíamos utilizar essa fechadura?**

lugares que não há essa necessidade de tamanha segurança, principalmente, e em residências por entretenimento e aumento da segurança.

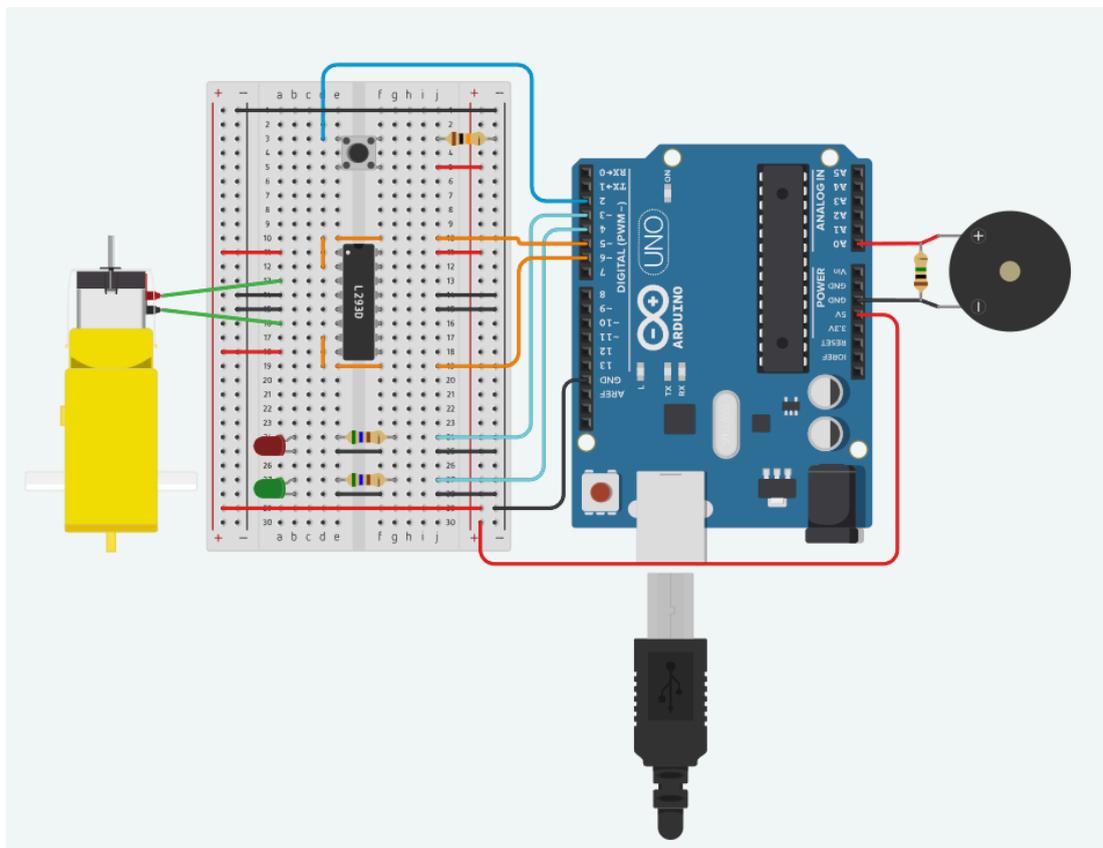
Como que essa fechadura irá abrir a porta?

A fechadura irá abrir a porta com ajuda do mecanismo do motor de engrenagem que fará a pressão para a maçaneta abrir.

Como será feita a implementação da sequência de toques?

A sequência de toques será gravada, com ajuda do Arduino e do botão, sendo que ao pressionar o botão, a pessoa poderá gravar no sensor a sequência de toques determinada, e o Arduino irá armazenar as informações.

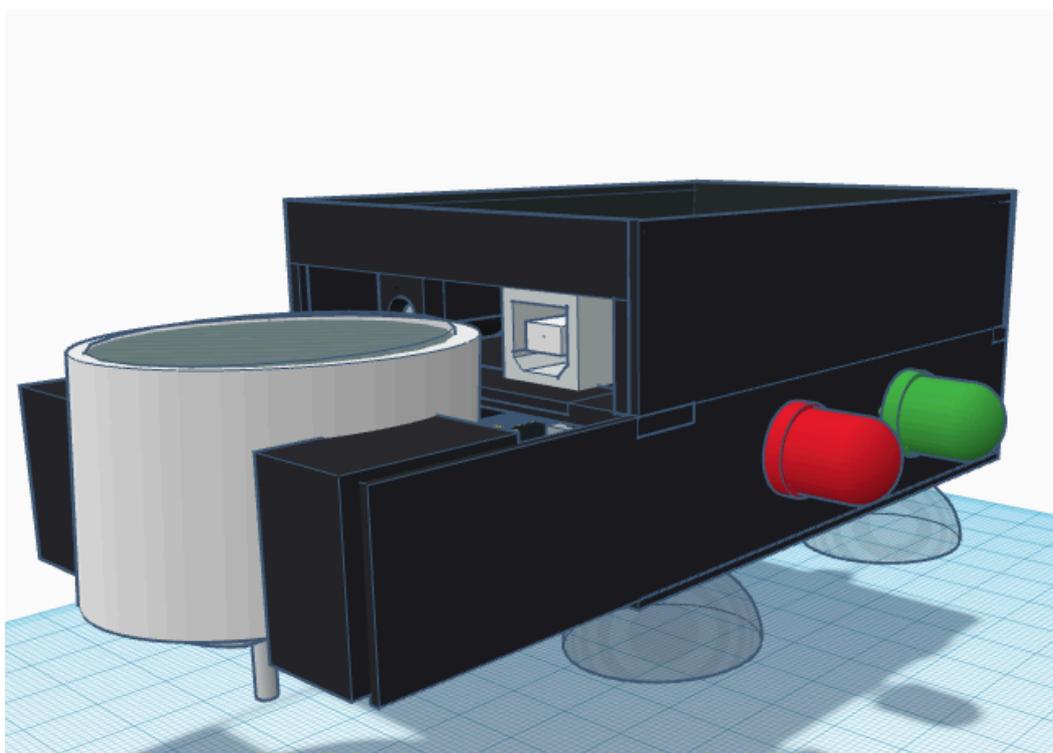
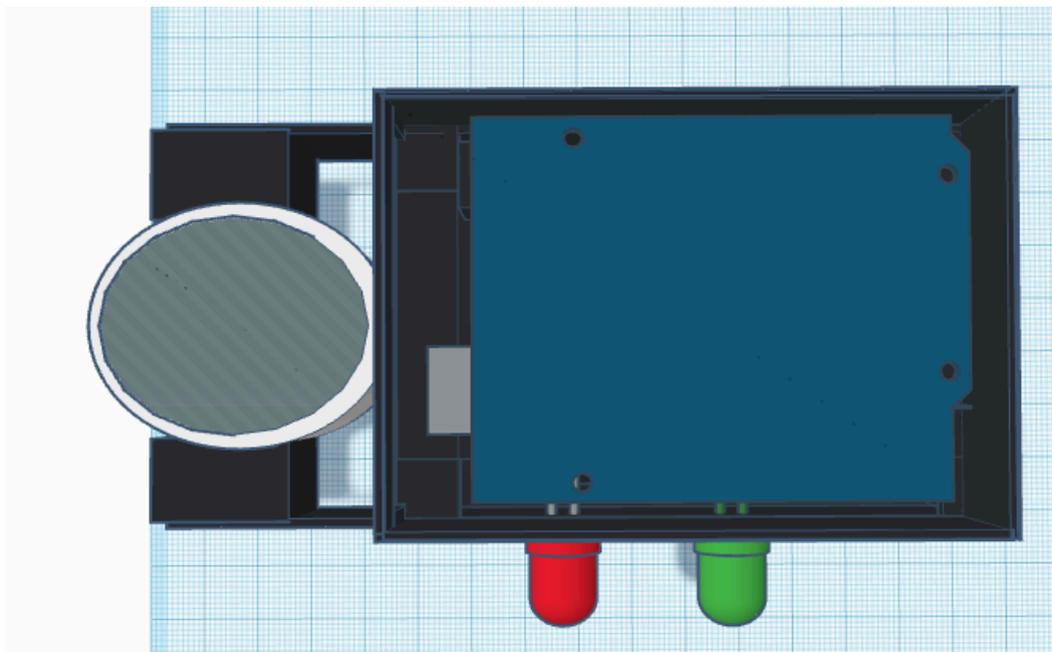
- Circuito feito no tinkercad



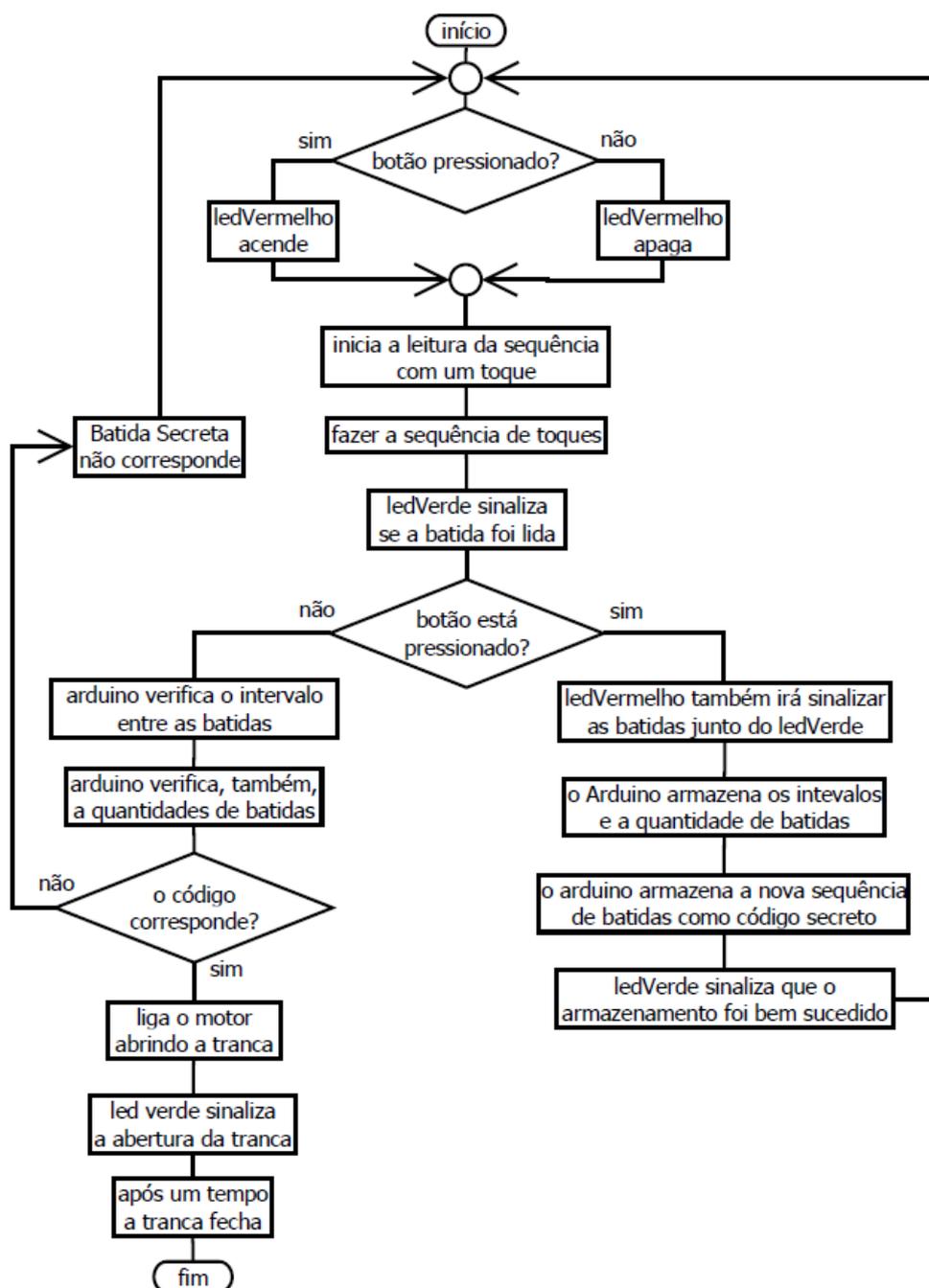
- Tabela de preços, especificações e quantidades

	A	B	C	D
1	MATERIAIS	QUANTIDADE	PREÇO	ESPECIFICAÇÕES
2	resistores	5	R\$0,5	560,560,1M,2,2k,10k
3	arduino	1	R\$46,46	uno
4	cabo do arduino	1	R\$x	x
5	protoboard	1	R\$9,40	x
6	led	2	R\$0,20	verm, verde
7	motor de engrenagem	1	R\$18,99	5v
8	transistor	1	R\$0,48	nnp
9	diodo	1	R\$0,22	retificador
10	fios	17	R\$10,09	condutor
11	microfone ou piezo	1	R\$4,66	x
12	bateria	1	R\$10	9V
13	placa de fenolite	1	R\$4,99	pequena
14	encaixe da bateria	1	R\$2	x
15	ventosa	6	R\$10	x

- Design do projeto (protótipo)



• Fluxograma



3. DESENVOLVIMENTO

Com o código feito pelo desenvolvedor original do projeto, melhoramos e adaptamos o código para Arduino UNO. Ficando assim o resultado:

```
#define botao 2
#define sensorPiezo A0
#define ledVermelho 3
#define ledVerde 4
#define motorIn1 5
#define motorIn2 6

boolean botaoPressionado = false;
byte valorSensorPiezo = 0;
#define valorRejeitar 25
#define mediaValorRejeitar 15
#define tempoMinimo 100
#define tempoDesvanecer 150
#define tempoGiroMotor 500
#define tempoCompletarBatida 1200
#define maximoBatidas 20
byte leituraBatidas[maximoBatidas];
byte codigoSecreto[maximoBatidas] = {20,50,30};

void setup(){
  pinMode(botao, INPUT);
  pinMode(ledVermelho,OUTPUT);
  pinMode(ledVerde,OUTPUT);
  pinMode(motorIn1,OUTPUT);
  pinMode(motorIn2,OUTPUT);
  pinMode(sensorPiezo,INPUT);
  Serial.begin(9600);
  digitalWrite(ledVerde,HIGH);
}

void loop(){
  digitalWrite(ledVerde,LOW);
  if(digitalRead(botao) == HIGH){
    botaoPressionado = true;
    digitalWrite(ledVermelho,HIGH);
  } else {
    botaoPressionado = false;
    digitalWrite(ledVermelho,LOW);
  }
  valorSensorPiezo = analogRead(sensorPiezo);
```

```

if(valorSensorPiezo >= tempoMinimo){
escutarBatidaSecreta();
}
}

void escutarBatidaSecreta(){
Serial.println("Iniciando Batida");
for (int i = 0; i < maximoBatidas; i++){
leituraBatidas[i] = 0;
}
byte numeroBatidaAtual = 0;
unsigned long tempolnicio = millis();
unsigned long agora = millis();
digitalWrite(ledVerde,LOW);
if(botaoPressionado == true){
digitalWrite(ledVermelho,LOW);
}
delay(tempoDesvanecer);
digitalWrite(ledVerde,HIGH);
if(botaoPressionado == true){
digitalWrite(ledVermelho,HIGH);
}
do {
valorSensorPiezo = analogRead(sensorPiezo);
if(valorSensorPiezo >= tempoMinimo){
Serial.println("Batida");
agora = millis();
leituraBatidas[numeroBatidaAtual] = agora - tempolnicio;
numeroBatidaAtual++;
tempolnicio = agora;
digitalWrite(ledVerde,LOW);
if(botaoPressionado == true){
digitalWrite(ledVermelho,LOW);
}
}
delay(tempoDesvanecer);
digitalWrite(ledVerde,HIGH);
if(botaoPressionado == true){
digitalWrite(ledVermelho,HIGH);
}
delay(tempoDesvanecer);
}
}
agora = millis();
} while ((agora - tempolnicio < tempoCompletarBatida) && (numeroBatidaAtual <
maximoBatidas));
if(botaoPressionado == false){
if(validarBatida() == true){

```

```

DestruirPorta();
} else {
Serial.println("Batida Secreta Falhou");
digitalWrite(ledVerde,LOW);
}
} else {
validarBatida();
Serial.println("Novo codigo armazenado");
digitalWrite(ledVermelho,LOW);
digitalWrite(ledVermelho,HIGH);
}
}

```

```

boolean validarBatida() {
byte BatidasAtual = 0;
byte BatidaSecreta = 0;
long intervaloMaxDasBatidas = 0;
for(int i=0; i < maximoBatidas; i++){
if(leituraBatidas[i] > 0){
BatidasAtual++;
}
if (codigoSecreto[i] > 0){
BatidaSecreta++;
}
if (leituraBatidas[i] > intervaloMaxDasBatidas){
intervaloMaxDasBatidas = leituraBatidas[i];
}
}
if (botaoPressionado == true){
for (int i = 0; i < maximoBatidas; i++){
codigoSecreto[i] = map(leituraBatidas[i], 0, intervaloMaxDasBatidas, 0, 100);
}
digitalWrite(ledVerde, LOW);
digitalWrite(ledVermelho, LOW);
delay(1000);
digitalWrite(ledVerde, HIGH);
digitalWrite(ledVermelho, HIGH);
delay(50);
for (int i = 0; i < maximoBatidas; i++){
digitalWrite(ledVerde, LOW);
digitalWrite(ledVermelho, LOW);
if (codigoSecreto[i] > 0){
delay(map(codigoSecreto[i], 0, 100, 0, intervaloMaxDasBatidas));
digitalWrite(ledVerde, HIGH);
digitalWrite(ledVermelho, HIGH);
}
}
}

```

```

delay(50);
}
return false;
}
if (BatidasAtual != BatidaSecreta){
return false;
}
int diferencaTotalTempo = 0;
int diferencaTempo = 0;
for (int i = 0; i < maximoBatidas; i++) {
leituraBatidas[i] = map(leituraBatidas[i], 0, intervaloMaxDasBatidas, 0, 100);
diferencaTempo = abs(leituraBatidas[i] - codigoSecreto[i]);
if (diferencaTempo > valorRejeitar) {
return false;
}
diferencaTotalTempo += diferencaTempo;
}
if (diferencaTotalTempo / BatidaSecreta > mediaValorRejeitar) {
return false;
}
return true;
}

void DestruarPorta() {
Serial.println("Porta destravada");
digitalWrite(motorIn1,HIGH);
digitalWrite(ledVerde,HIGH);
delay(tempoGiroMotor);
digitalWrite(motorIn1,LOW);
digitalWrite(motorIn2,HIGH);
delay(tempoGiroMotor);
digitalWrite(motorIn2,LOW);
digitalWrite(ledVerde,LOW);
}

```

● Explicação das rotinas e as variáveis.

Pinos:

botao: botão usado para configuração para implementar um novo código secreto.
(entrada digital)

sensorPiezo: sensor que irá detectar as batidas. (entrada analogica)

ledVermelho: led de sinalização de toques da configuração do código secreto.
(saída digital)

ledVerde: led de sinalização geral. (saída digital)

motorIn1 e motorIn2: pinos que são ligados à ponte H para ativar o motor. (saída digital)

Variáveis Globais:

botaoPressionado: irá dizer ao arduino se o botão está pressionado.

valorSensorPiezo: valor do sensor piezo lido pelo arduino.

valorRejeitar: Valor de rejeição para a diferença de tempo máxima entre as batidas registradas e o código secreto para considerar uma sequência válida.

mediaValorRejeitar: Média do valor de rejeição para a diferença de tempo média limite entre as batidas.

tempoMinimo: tempo mínimo de detecção para considerar uma batida válida.

tempoDesvanecer: Tempo de espera para auxiliar a sinalização dos leds.

tempoGiroMotor: Tempo de espera para girar o motor.

tempoCompletarBatida: Tempo máximo para completar uma sequência de batidas.

maximoBatidas: Número máximo de batidas na sequência. (ajustável)

leituraBatidas: Array para armazenar os tempos das batidas detectadas.

codigoSecreto: Array que armazena a sequência de batidas do código secreto.

Estruturas do código:

void setup(): Parte do código que configura os pinos do arduino e sua Tabela Serial.

void loop(): Parte do código que irá ser executado várias vezes. Verifica o valor do piezo e se é uma batida válida, se for válido é chamado a rotina “escutarBatidaSecreta” para iniciar a leitura da sequência de batidas. Também verifica se o botão está pressionado e sinaliza com o led vermelho.

void escutarBatidaSecreta(): Mostra na Tabela Serial que está sendo iniciado a leitura das batidas e zera a leitura das batidas para iniciar uma nova. “Escuta” cada batida, armazena elas e sinaliza com led verde, caso o botão esteja pressionado sinaliza com vermelho também. Se o botão não estiver pressionado a rotina “ValidarBatida” é chamada, se retornar verdadeiro destrava a porta com a rotina “DestruarPorta”, se voltar falso indica na Tabela Serial que a sequência foi inválida. Caso o botão esteja pressionado chama a rotina “ValidarBatida” para armazenar um novo código e indica na Tabela Serial e no led verde e no led vermelho.

boolean validarBatida(): Primeiro verifica as batidas detectadas, se há batidas correspondentes ao código secreto e o intervalo entre elas. Se o botão não estiver pressionado, verifica se há batidas que não correspondem ao código, se o intervalo entre elas e a média entre elas está dentro do limite estabelecido. Caso alguma

dessas verificações não corresponder retorna a função como “falso”, se todas corresponderem volta como “verdadeiro”. Se o botão estiver apertado irá registrar as batidas como um novo código secreto, sinalizando com os leds, e retorna para onde ela foi chamada.

void acionarDestravamentoPorta(): Mostra na Tabela Serial que a porta vai ser destravada. liga o motor girando ele em um sentido e sinaliza com o led verde, espera o tempo especificado e gira o motor no sentido contrário. sinaliza com o led verde que o motor está ligado.

CONCLUSÃO

REFERÊNCIAS

video:

https://youtu.be/RyBjpiHnP74?si=n7WncJZBXsR7p_mS (aos 5:05)

projeto:

<https://www.instructables.com/Secret-Knock-Detecting-Door-Lock/>