

ROBOT CLASS



Guia Didático De Montagem - TRANCA

Este guia é para todos que têm interesse em robótica e que querem implementar uma solução de tranca eletrônica para moradias. A tranca eletrônica oferece uma maneira moderna e segura de garantir o acesso às residências, usando tecnologia avançada para aumentar a segurança dos moradores.

Neste guia, vamos aprender desde o básico, como os princípios de eletrônica e programação. Utilizaremos tecnologias incríveis como microcontroladores (que são tipo cérebros eletrônicos) e sensores (que são como olhos e ouvidos eletrônicos) para construir algo realmente incrível. Vamos explorar como detectar se a porta está fechada ou aberta, usando sensores de proximidade, e como criar métodos de autenticação super seguros, como senhas!

No Robot Class, vamos começar com os conceitos básicos e, gradualmente, avançar para projetos mais complexos. Este guia é projetado para ser interativo e prático, permitindo que você aplique os conhecimentos de robótica e programação que você adquiriu em um contexto real e significativo.

Sumário

1. Introdução

- 1.1 Fechadura eletrônica com Arduino 4
- 1.2 Quais são as exigências para o funcionamento desse projeto ... 4

2. Primeiros passos

- 2.1 Componentes Necessários 5
- 2.2 Como funciona? 8
- 2.3 Montagem Passo a Passo 9
- 2.3 Estrutura do código 10

1.0 Introdução:

1.1 Fechadura Eletrônica com Arduino

Objetivo desse projeto é Desenvolver uma fechadura eletrônica utilizando uma placa arduino conhecida como UNO, um teclado numérico e um servomotor. O projeto visa criar uma solução simples para controle de acesso, permitindo que usuários autorizados desbloqueiem a fechadura ao digitar uma senha válida, acionando um servomotor que controla o mecanismo de trava física.



1.2 Quais são as exigências para o funcionamento desse projeto?

Autenticação Rápida

O sistema deve fornecer uma resposta de autenticação (ou negação) em tempo real após a digitar o código correto;

Consumo de Energia

O sistema deve ser projetado para ter um consumo eficiente de energia, aumentando a vida útil da fonte de energia;

Durabilidade

Os componentes devem ser montados de maneira robusta e segura, garantindo a durabilidade da fechadura eletrônica;

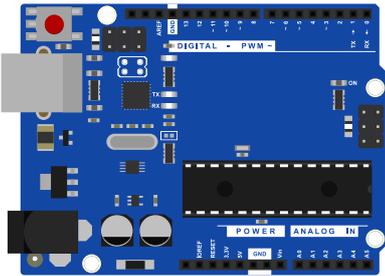
Recuperação em Falhas

Em caso de falha no sistema, como travamento do *software* e problemas no *hardware*, o sistema deve ser de fácil manutenção.

2.0 Primeiros Passos:

2.1 Componentes necessários

- Placa Arduino UNO;



- Teclado numérico para acionar o servomotor;



- Servomotor para controlar a trava física;



2.1 Componentes necessários

- Jumpers (Fios conectores).



- Luz *LED* verde;

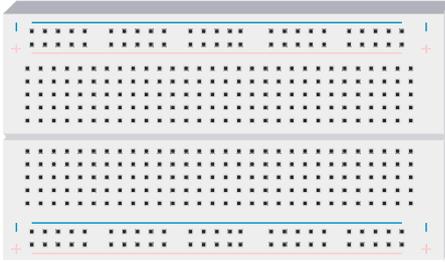


- HC-SR04 (Sensor de aproximação);

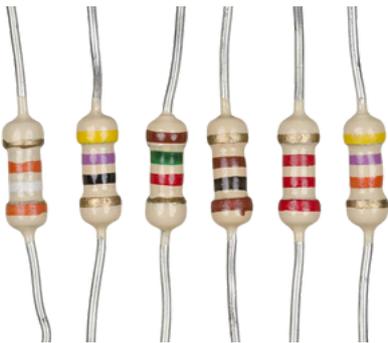


2.1 Componentes necessários

- *Protoboard* (Placa de Ensaio);



- Resistores;



- Fonte de energia (usaremos nossos computadores) para alimentar os componentes;
- Estrutura física para montagem dos componentes.

2.2 Como funciona?

1.Preparação: No início, o código diz ao computador quais ferramentas vai usar, como um teclado, um motor e um sensor de distância.

2.Configuração Inicial: Antes de começar, o programa prepara as partes importantes, como ligar o motor e o *LED* verde.

3.Loop Principal: O programa faz as mesmas coisas repetidamente:

Leitura das Teclas: Olha se alguma tecla do teclado foi pressionada.

Verificação da Senha: Confere se a senha está correta. Se estiver, abre a "porta" e acende o *LED* verde.

Contagem de Tentativas: Se errar a senha, dá algumas chances antes de parar por um tempo.

Controle da Porta: Verifica se algo está muito perto da "porta" usando o sensor de distância e, se estiver, fecha a "porta".

Basicamente, é um sistema de segurança que protege uma "porta" e mantém as coisas seguras.

2.3 Montagem Passo a Passo:

Passo 1: Conexão do Arduino

- Conecte o Arduino Uno à protoboard.
- Certifique-se de que o Arduino esteja alimentado corretamente.

Passo 2: Teclado Matricial

- Conecte o teclado numérico ao Arduino usando os jumpers
- O teclado deve ter uma matriz de pinos para conexão.

Passo 3: Sensores ou Botões

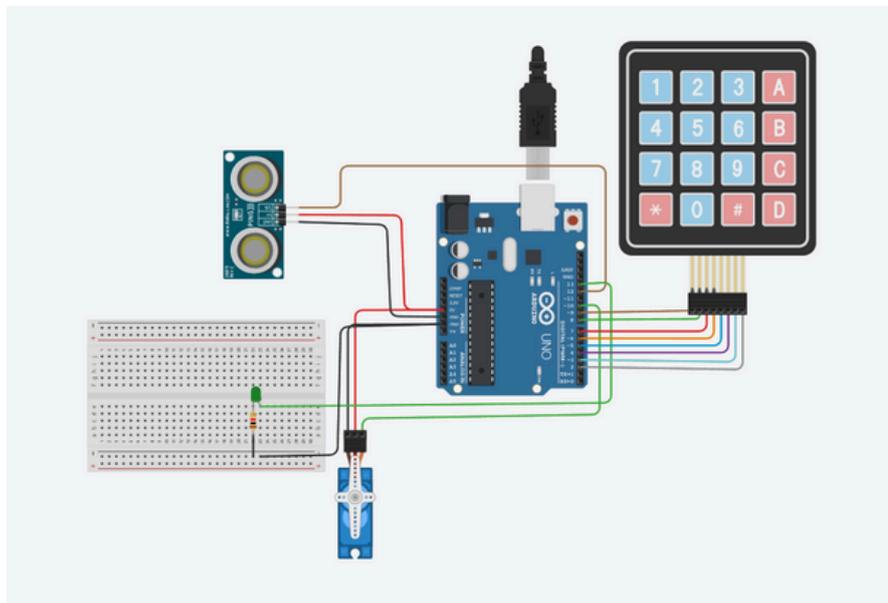
- Conecte os sensores diretamente ao Arduino.

Passo 4: LEDs

- Insira os LEDs na breadboard.
- Conecte o LED verde a um pino digital do Arduino.

Passo 5: Servo Motor

- Conecte o servo motor ao Arduino.
- O servo motor deve ser conectado a um pino PWM (Pulse Width Modulation) para controlar sua posição.



2.4 Estrutura do código

1. Inclusão de bibliotecas:

```

1  #include <Keypad.h>
2  #include <Servo.h>
3  #include <Ultrasonic.h>
  
```

Essas linhas incluem as bibliotecas necessárias para usar um teclado matricial, um servo motor e um sensor ultrassônico.

2. Definição de constantes e variáveis globais:

```

5  const byte ROW_NUM = 4; // Quatro linhas
6  const byte COL_NUM = 4; // Quatro colunas
7
8  const char TECLAS_MATRIZ[ROW_NUM][COL_NUM] = { // Matriz de caracteres
   (mapeamento do teclado)
9    {'1', '2', '3', '*'},
10   {'4', '5', '6', '0'},
11   {'7', '8', '9', '#'},
12   {'A', 'B', 'C', 'D'}
13 };
14
15 const byte pin_rows[ROW_NUM] = {9, 8, 7, 6}; // Pinos de conexão com as
   linhas do teclado
16 const byte pin_column[COL_NUM] = {5, 4, 3, 2}; // Pinos de conexão com as
   colunas do teclado
17
18 Keypad keypad = Keypad(makeKeymap(TECLAS_MATRIZ), pin_rows, pin_column,
   ROW_NUM, COL_NUM);
19
20 Servo servo;
21 String password = "";
22 String correctPassword = "123";
23
  
```

Essas linhas definem o layout do teclado matricial, os pinos de conexão, e inicializam objetos para o teclado e o servo motor. Também são definidas variáveis para armazenar a senha e a senha correta.

2.4 Estrutura do código

3. Configuração inicial:

```

39 void setup() {
40   servo.attach(10); // Conecte o servo ao pino digital 10
41   pinMode(greenLedPin, OUTPUT); // Define o pino do LED verde como saída
42   Serial.begin(9600);
43 }
  
```

No `setup()`, o servo motor é anexado ao pino digital 10, e o pino do LED verde é configurado como saída. Além disso, a comunicação serial é iniciada.

4. Loop principal:

```

45 void loop() {
46   char key = keypad.getKey();
47
48   if (key) {
49     password += key;
50     Serial.println(password);
51
52     // Verifique se a senha digitada está correta
53     if (password == correctPassword) {
54       Serial.println("Senha correta!");
55       digitalWrite(greenLedPin, HIGH); // Acenda o LED verde
56       servo.write(180); // Gire o servo em 180 graus
57       password = ""; // Limpe a senha
58       doorClosed = false; // Redefinir a variável doorClosed
59       attempts = 0; // Redefinir o contador de tentativas
60       delay(5000);
61       digitalWrite(greenLedPin, LOW); // Desligue o LED verde
62     } else if (password.length() >= correctPassword.length() && password !=
        correctPassword) {
63       attempts++;
64       if (attempts >= maxAttempts) {
65         Serial.println("Número máximo de tentativas atingido. Aguarde antes de tentar
            novamente.");
66         delay(60000); // Bloqueie novas tentativas por 60 segundos
67         attempts = 0; // Redefinir o contador de tentativas
68       } else {
69         Serial.println("Senha incorreta!");
70         for(int i=0; i<5; i++){ // Faça o LED verde piscar 10 vezes
71           digitalWrite(greenLedPin, HIGH); // Acenda o LED verde
  
```

No loop principal, o código lê as teclas digitadas no teclado e processa-as. Em seguida, lê a distância medida pelo sensor ultrassônico. Dependendo da distância e do estado da porta, o servo motor pode ser controlado para abrir ou fechar a porta.