

Economizador de Água e Energia para o Império

Quando o Lado Negro precisa economizar água - Tarefa realizada pelos integrantes do Planeta Terra, estudantes da FIAP para o desafio Global Solution



Inspiração para a carcaça impressa em 3D do Thumbnail do vídeo "Darth vader clock DIY - TUTORIAL" do canal Jéjé l'ingé no YouTube - <https://www.youtube.com/watch?v=9qdEhR3RgWc>

Índice

1. [Introdução](#)
2. [Funcionalidades](#)
3. [Requisitos de Hardware](#)
4. [Configuração de Pinos](#)
5. [Guia de Instalação](#)
6. [Como Funciona](#)
7. [Especificações Técnicas](#)
8. [Sistema de Alerta Imperial](#)
9. [Solução de Problemas](#)

Introdução

O Economizador de Água e Energia para o Império é uma solução inovadora projetada para ajudar o pessoal Imperial a manter um uso eficiente de água durante suas rotinas de higiene. Usando tecnologia avançada de sensores combinada com o poder motivacional da Marcha Imperial, este sistema garante que até os Stormtroopers não desperdicem recursos preciosos da Estrela da Morte. A recomendação da OMS é de que a duração do banho para a suficiente higienização do corpo e para a economia de energia e de água deve durar 5 minutos, sendo o tempo constante neste projeto.

Funcionalidades

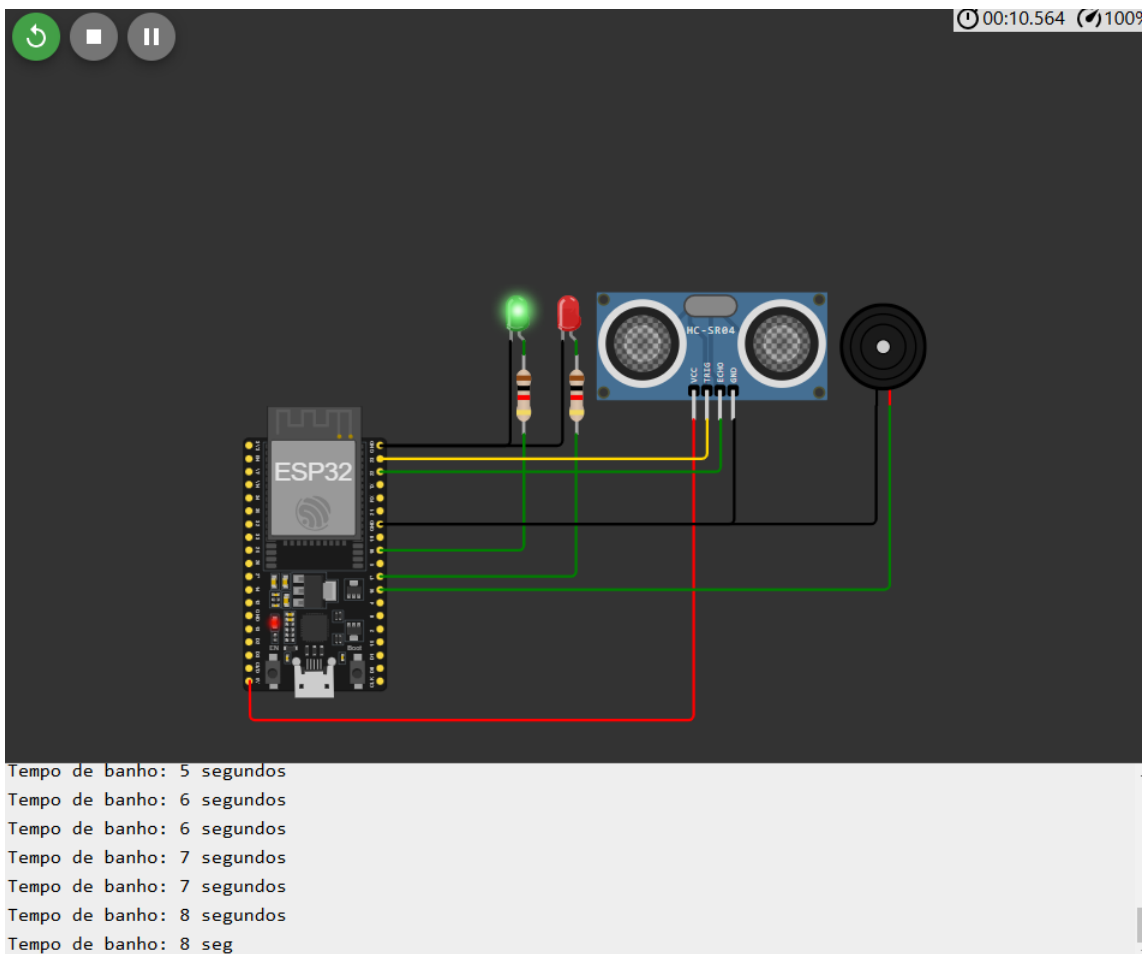
- Monitoramento em tempo real da duração do banho
- Detecção de presença por ultrassom
- Feedback visual através de LEDs indicadores
- Sistema de alerta com a Marcha Imperial
- Reset automático quando a área do banho é desocupada
- Capacidades de monitoramento serial para droides de manutenção

Requisitos de Hardware

- Microcontrolador ESP32
- Sensor Ultrassônico HC-SR04
- 2x LEDs (Verde e Vermelho)
- Buzzer
- Resistores
- Fios de conexão
- Fonte de alimentação
- Caixa à prova d'água

Configuração de Pinos

| Componente | Número do Pino |
|-------------------|----------------|
| Trigger (HC-SR04) | 23 |
| Echo (HC-SR04) | 22 |
| LED Verde | 18 |
| LED Vermelho | 17 |
| Buzzer | 16 |



*O poder da força atuando sobre o aparelho e mostrando o funcionamento

Guia de Instalação

1. Monte o sensor ultrassônico na entrada do chuveiro
2. Instale os LEDs em local visível
3. Posicione o buzzer onde o som possa se propagar efetivamente
4. Conecte os componentes de acordo com a configuração dos pinos
5. Faça o upload do código fornecido para o ESP32
6. Coloque o sistema em caixa à prova d'água
7. Ligue o sistema

Como Funciona

1. Sistema ativa quando detecta presença em um alcance de 100cm
2. LED Verde indica banho em andamento
3. Timer controla a duração do banho
4. Após o tempo limite (5 minutos):
 - LED Vermelho ativa
 - Marcha Imperial toca
 - LED Verde desativa
5. Sistema reinicia após 1 minuto sem detecção de presença

Especificações Técnicas

```
const unsigned long TEMPO_LIMITE_BANHO = 5 * 60 * 1000;    // 5 minutos
const unsigned long TEMPO_LIMITE_AUSENCIA = 1 * 60 * 1000; // 1 minuto
const int DISTANCIA_DETECCAO = 150;                       // 150 cm
```

Sistema de Alerta Imperial

O sistema apresenta a icônica Marcha Imperial como som de alerta. A sequência musical é programada usando frequências específicas:

- NOTA_A4: 440 Hz
- NOTA_F4: 349 Hz
- NOTA_C5: 523 Hz
- NOTA_G5: 784 Hz
- E outras frequências Imperiais

Solução de Problemas

| Problema | Possível Causa | Solução |
|----------------------------|--------------------------|--------------------------------|
| Sem detecção de presença | Desalinhamento do sensor | Ajustar ângulo do sensor |
| Alarme silencioso | Conexão do buzzer | Verificar conexão do pino 16 |
| Disparos falsos constantes | Interferência | Ajustar posição do sensor |
| LEDs não funcionam | Resistência incorreta | Verificar conexões dos LEDs |
| Marcha Imperial distorcida | Problemas de energia | Verificar fonte de alimentação |

Poder do Lado Negro

Lembre-se:

"A habilidade de destruir um planeta é insignificante perto do poder de economizar água."

- Darth Vader (provavelmente)

Cronograma de Manutenção

- Diário: Verificar alinhamento do sensor
- Semanal: Testar sistema de alerta
- Mensal: Verificar vedações à prova d'água
- Anual: Diagnóstico completo do sistema

Notas para o Pessoal Imperial

1. Sistema projetado para unidades de chuveiro Imperial padrão
2. Não testado em Wookiees

3. Manter longe de relâmpagos da Força
4. Protocolos de manutenção Imperial se aplicam

Que a Força (e a economia de água) esteja com você

Anexo 1: Código para inserir no Arduino IDE

```
// Economizador de Água e Energia para o Império
// Desenvolvido para a manutenção da ordem e economia na galáxia

// Definição dos pinos
const int trigPin = 23;
const int echoPin = 22;
const int ledVerde = 18;
const int ledVermelho = 17;
const int buzzer = 16;

// Definição das notas musicais para a Marcha Imperial
#define NOTE_A4 440
#define NOTE_F4 349
#define NOTE_C5 523
#define NOTE_G5 784

// Configurações de tempo
const unsigned long TEMPO_LIMITE_BANHO = 5 * 60 * 1000; // 5 minutos
const unsigned long TEMPO_LIMITE_AUSENCIA = 1 * 60 * 1000; // 1 minuto

// Variáveis de controle
bool pessoaPresente = false;
unsigned long tempoInicio = 0;
unsigned long tempoSemPresenca = 0;

// Melodia da Marcha Imperial
int melodia[] = {
    NOTE_A4, NOTE_A4, NOTE_A4, NOTE_F4, NOTE_C5,
    NOTE_A4, NOTE_F4, NOTE_C5, NOTE_A4
};

int duracoes[] = {
    4, 4, 4, 8, 16,
    4, 8, 16, 2
};

// Função para tocar a Marcha Imperial
void tocarMarchaImperial() {
    int tamanhoMelodia = sizeof(melodia) / sizeof(melodia[0]);

    for (int i = 0; i < tamanhoMelodia; i++) {
        int duracaoNota = 1000 / duracoes[i];
        tone(buzzer, melodia[i], duracaoNota);
        delay(duracaoNota * 1.3);
    }
}
```

```

    noTone(buzzer);
}
}

void setup() {
    // Inicialização da comunicação serial
    Serial.begin(9600);

    // Configuração dos pinos
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    pinMode(ledVerde, OUTPUT);
    pinMode(ledVermelho, OUTPUT);
    pinMode(buzzer, OUTPUT);

    // Estado inicial dos componentes
    digitalWrite(ledVerde, LOW);
    digitalWrite(ledVermelho, LOW);
    digitalWrite(buzzer, LOW);

    Serial.println("Sistema iniciado - aguardando presença.");
}

void loop() {
    int distancia = calcularDistancia();

    if (distancia > 0 && distancia <= 150) {
        tratarPresenca();
    } else {
        tratarAusencia();
    }

    delay(500);
}

// Função para calcular a distância usando o sensor ultrassônico
int calcularDistancia() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    long duration = pulseIn(echoPin, HIGH);
    return duration * 0.034 / 2;
}

// Função para tratar a presença detectada
void tratarPresenca() {
    if (!pessoaPresente) {
        pessoaPresente = true;
        tempoInicio = millis();
    }
}

```

```

    Serial.println("Pessoa detectada no banho. Timer iniciado.");
}

unsigned long tempoAtual = millis() - tempoInicio;
Serial.print("Tempo de banho: ");
Serial.print(tempoAtual / 1000);
Serial.println(" segundos");

if (tempoAtual >= TEMPO_LIMITE_BANHO) {
    digitalWrite(ledVerde, LOW);
    digitalWrite(ledVermelho, HIGH);
    tocarMarchaimperial(); // Toca a melodia da Marcha Imperial
    Serial.println("Tempo excedido! Aviso acionado.");
} else {
    digitalWrite(ledVerde, HIGH);
    digitalWrite(ledVermelho, LOW);
}

tempoSemPresenca = 0;
}

// Função para tratar a ausência detectada
void tratarAusencia() {
    if (pessoaPresente) {
        if (tempoSemPresenca == 0) {
            tempoSemPresenca = millis();
            Serial.println("Pessoa saiu do alcance do sensor.");
        } else if (millis() - tempoSemPresenca >= TEMPO_LIMITE_AUSENCIA) {
            pessoaPresente = false;
            tempoInicio = 0;
            digitalWrite(ledVerde, LOW);
            digitalWrite(ledVermelho, LOW);
            Serial.println("Timer resetado. Sistema aguardando nova detecção.");
        }
    }
}
}

```