

Bloqueio de porta com detecção de batida secreta

Yan Correa

Universidade Federal de Mato Grosso do Sul

FACOM – Faculdade de Computação

Índice

1. Objetivos
2. Materiais Utilizados
3. Sensor piezoelétrico
4. Protótipo
5. Circuito
6. Principais Implementações
7. Dificuldades e possíveis melhorias

Objetivos

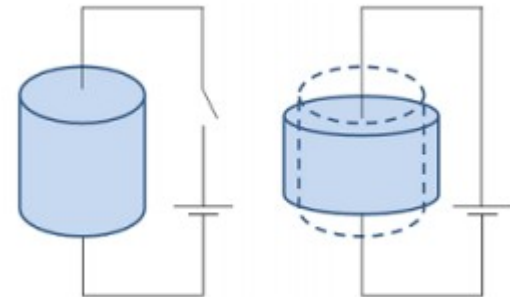
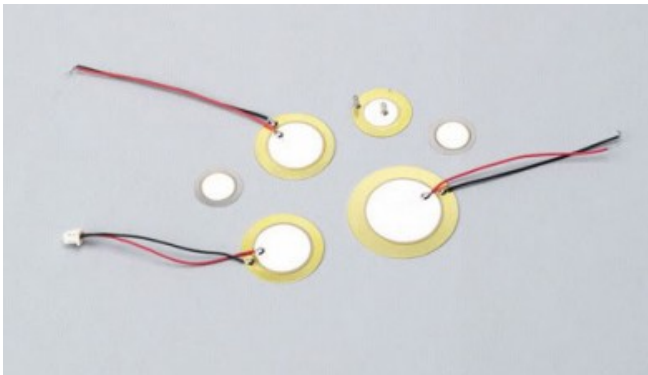
Apresentar uma aplicação prática de bloqueio de porta com detecção de batida secreta, utilizando o elemento piezoelétrico para detectar sons, o que permitiu usá-lo como um sensor de batidas, criando assim um sistema de segurança.

Materiais Utilizados

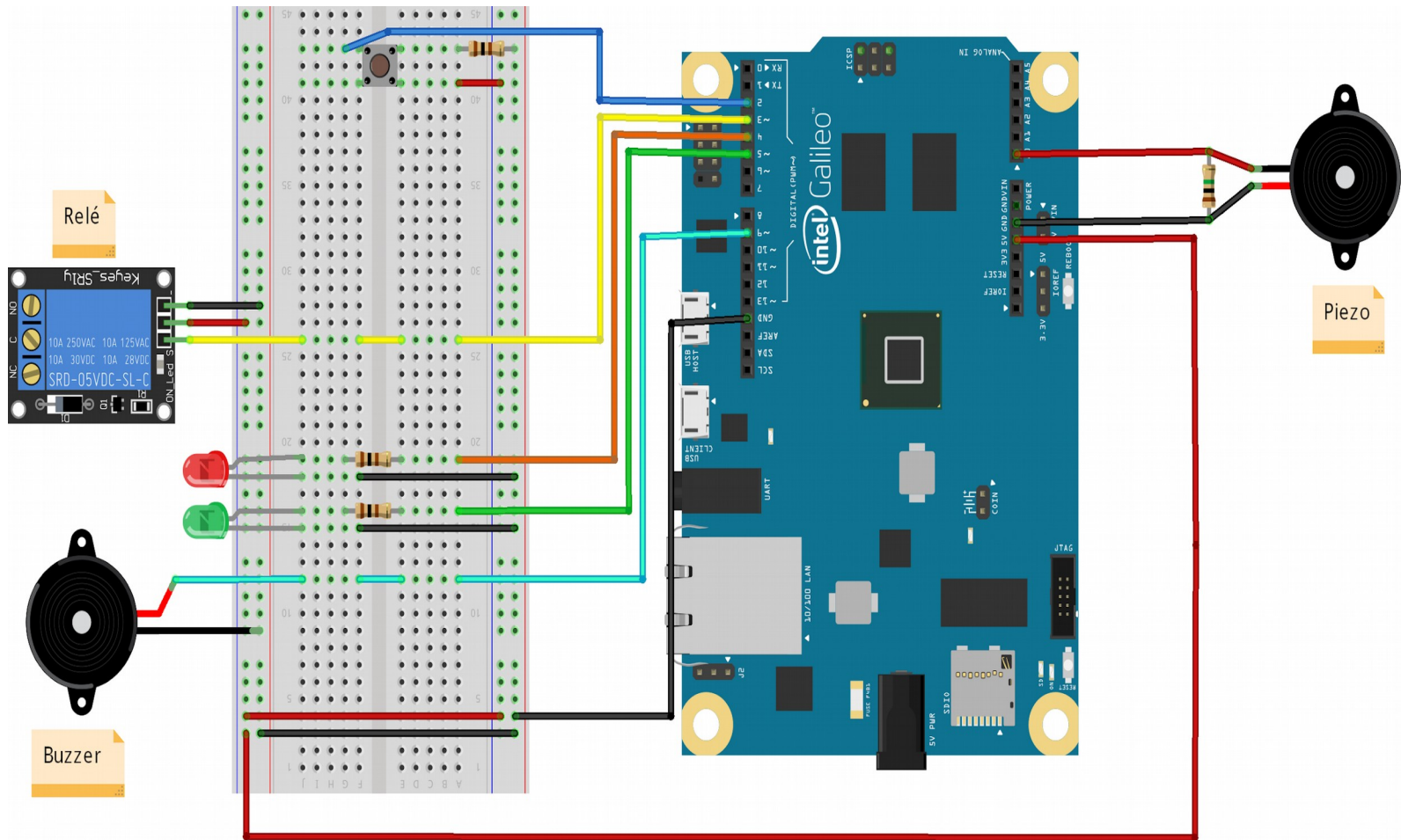
- 01 Placa Intel Galileo;
- 01 LED verde e 01 vermelho;
- 02 Resistores de 100Ω (Leds);
- 01 Resistor de $10k\Omega$ (Push-button);
- 01 Resistor de $1M\Omega$ (Piezo);
- 01 Módulo Relé;
- 01 Sensor piezoelétrico;
- 01 Buzzer;
- 01 Chave Táctil (Push-Button);
- 01 Fechadura Elétrica 12V;
- 01 Fonte 12V para a Fechadura;
- 01 Protótipo de porta de metal;
- 01 Matriz de contatos (Protoboard);
- Jumpers variados.

Sensor piezoelétrico

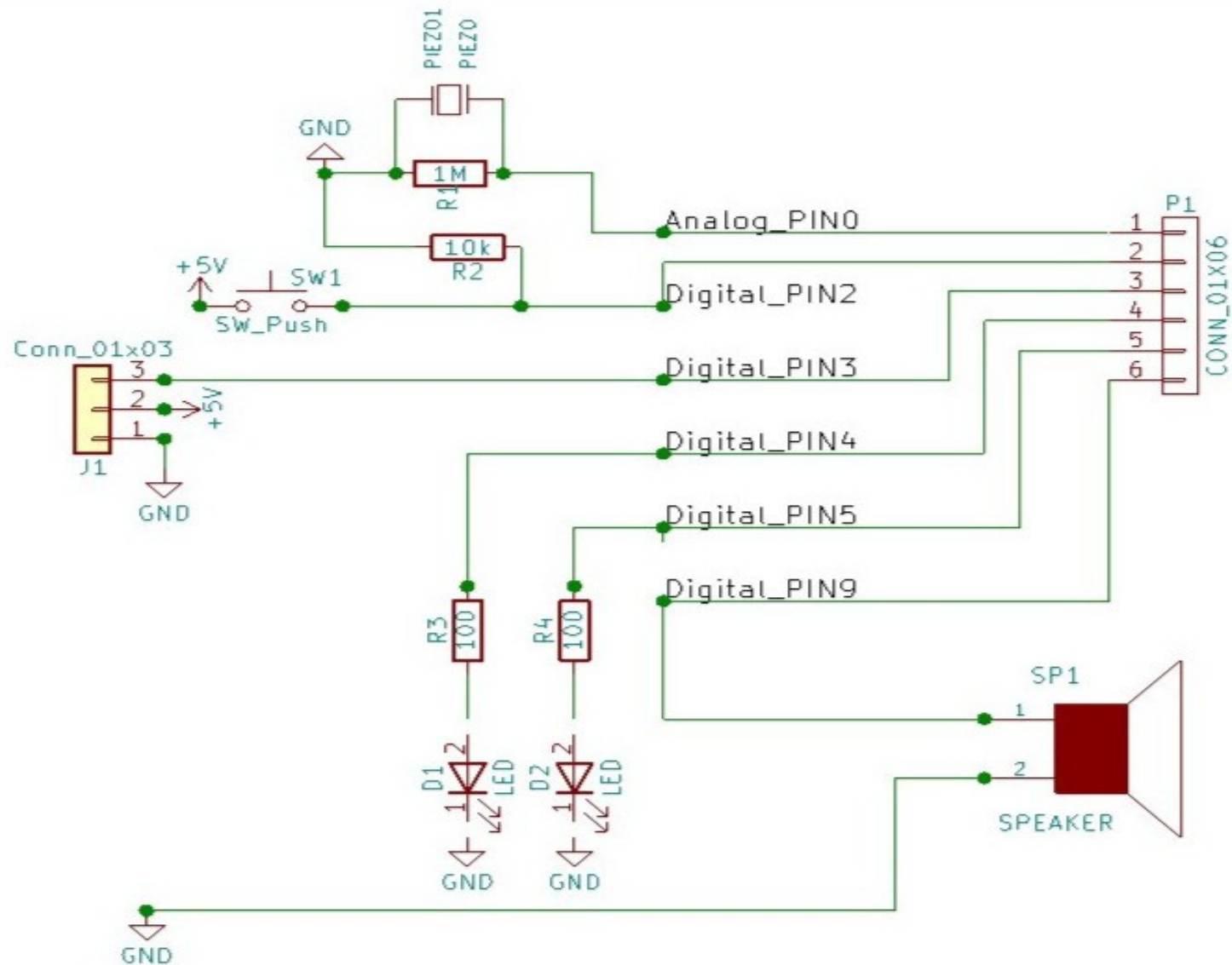
- Dispositivo que mede a pressão ou tensão utilizando a piezeletricidade, que é a capacidade de alguns [cristais](#) gerarem [tensão elétrica](#) por resposta a uma [pressão mecânica](#).
- Algumas aplicações úteis: produção e detecção de som (microfones e captadores de guitarra), geração de frequências eletrônicas, microbalanças.



Protótipo



Circuito



Principais Implementações

- `const int threshold = 3;` //configuração da sensibilidade do piezo;
- `const int rejectValue = 25;` //porcentagem de erro entre duas batidas;
- `const int maximumKnocks = 20;` //tamanho máximo da senha (batidas);
- `int secretCode [maximumKnocks];` //vetor que armazena os tempos entre as batidas;
- `boolean validateKnock();` //retorna True quando a senha for correta;

Dificuldades

- Configurar sensibilidade do sensor piezoelétrico;
- Encontrar um tempo razoável de erro entre uma batida e outra (25%);
- Acionar o relé (abrir a porta) apenas quando a senha for a correta.

Possíveis melhorias

- Armazenar vários padrões de batidas, assim, cada usuário possui seu próprio padrão;
- Implementar um mecanismo seguro para a troca de padrão de batidas;
- Acoplar uma bateria ao projeto (prevenir o problema de queda/falta de energia).
- Funcionamento no “modo silencioso”.