Entrega Trabalho Prático 1

Sistemas Embebidos e de Tempo Real

Aluno/os:
21140 - Pedro Vieira Simões
21145 – Gonçalo Moreira da Cunha
21152 – João Carlos da Costa Apresentação

Professor/es: Pedro Cunha

Licenciatura em Engenharia de Sistemas Informáticos

Barcelos, janeiro de 2023

IPCA ESCOLA SUPERIOR DE TECNOLOGIA

Detetor de incêndios

Resumo

Este trabalho prático, relativo à unidade curricular de **Sistemas Embebidos e de Tempo Real,** é focado no desenvolvimento de um sistema de deteção e alarme de incêndios e demonstrar técnicas e conceitos abordados inter e extracurricular.

Índice

Conteúdo

Índice de Imagens

1. Introdução

1.1. Contextualização

O presente relatório tem como objetivo apresentar um projeto de sistema de alerta de incêndio e deteção de inclinação utilizando a plataforma Arduino. O projeto foi desenvolvido com o intuito de proporcionar segurança e proteção em ambientes residenciais ou comerciais.

1.2. Motivação e Objetivos

A motivação para o desenvolvimento deste projeto surge da necessidade de se ter um sistema de alerta eficiente e de fácil implementação em ambientes residenciais ou comerciais. Incêndios e inclinações são situações que podem colocar em risco a segurança das pessoas e dos bens, e, por isso, é importante ter um sistema que possa alertar de forma rápida e precisa sobre esses eventos.

O objetivo principal deste projeto é proporcionar segurança e proteção aos ambientes onde o sistema for instalado, através da deteção de incêndios e inclinações. Além disso, outros objetivos incluem:

- Facilitar a implementação do sistema, de forma a torná-lo acessível a um público amplo;
- Utilizar componentes eletrônicos de baixo custo, para que o sistema seja econômico;
- Garantir a precisão e eficiência do sistema, através da utilização de interrupções.

1.3. Estrutura do Documento

O projeto foi dividido em seções lógicas e coerentes, visando a facilidade de leitura e compreensão do conteúdo.

2. Montagem

Para iniciar o projeto, foi realizada a montagem virtual através da plataforma Tinkercad, utilizando o seguinte link: https://www.tinkercad.com/things/aD1JZw4TUNs-alarmeincendio/editel?sharecode=pMnPEpZj4mMRv8XEz4Oszm61ojMQMfR0DBsrap04-mY.
Depois, o projeto foi montado fisicamente com o uso de componentes eletrônicos e o Arduino IDE para o desenvolvimento do código fonte.

2.1. Materiais

Para a construção foram utilizados os seguintes materiais:

Arduino Uno R3	1
Piezo / Buzzer	1
Sensor de inclinação	1
LED Vermelho	1
Foto resistor	1
Resistência 10 kΩ	1
Resistência 220 Ω	4
LED RGB	1
Botão	1
Resistência 200 Ω	1

2.2. Arquitetura

Foi criada uma arquitetura de forma a suportar todos os materiais e forma a implementar o idealizado.

Começando pelos leds, o led vermelho foi inserido de forma a ter o terminal positivo conectado a uma resistência de 220 Ohms que liga ao ponto 11, e o seu terminal negativo liga-se ao caminho terra (negativo). Já o led RGB é conectado 3 das suas pernas as resistências de 220 Ohms que ligam aos pontos 11, 10 e 9, e a restante perna será conectada ao caminho terra.

O botão terá o seu terminal negativo conectado ao caminho positivo, terminal negativo liga á resistência de 200 Ohms que liga ao caminho terra. Depois ainda tem um fio que liga ao ponto 12.

O Buzzer liga o seu terminal positivo ao ponto 13 e o negativo liga ao caminho terra.

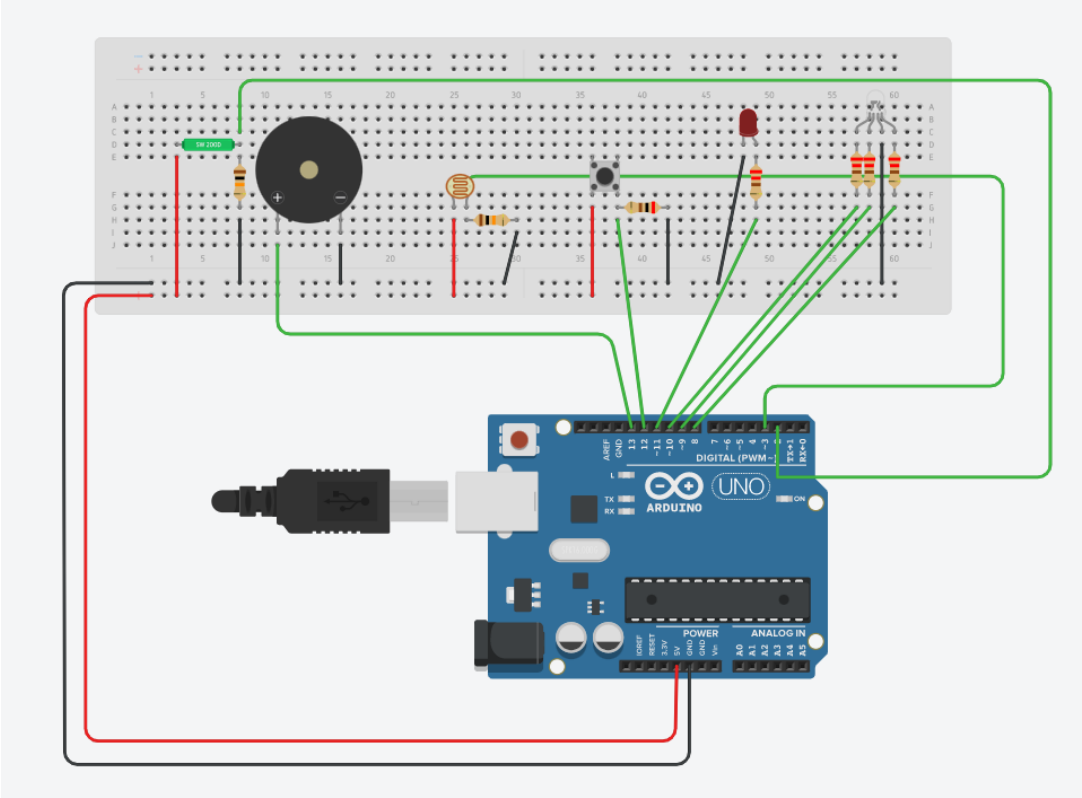
O sensor de inclinação tem o seu terminal positivo conectado ao caminho positivo e o negativo conecta-se a uma resistência 10k Ohms que liga ao caminho terra. Ainda tem um fio que se conecta ao ponto 2.

O sensor de luminosidade tem o seu terminal positivo conectado ao caminho positivo e o negativo conecta-se a uma resistência 10k Ohms que liga ao caminho terra. Ainda tem um fio que se conecta ao ponto 1.

Todos os fios que se conectam ao caminho terra são conectados ao GND e os que se conectam ao caminho positivo ao 5V.

As resistências usadas devem-se ao facto de ser a melhor medida encontrada para evitar danificações e garantir o bom funcionamento do programa.

Segue-se a representação desta arquitetura:



1- Montagem Tinkercad

2.3. Funcionamento

O alarme de incêndio é ativado quando o sensor de luminosidade (fotoresistore) deteta uma mudança brusca na luminosidade. Quando isso acontece, a campainha é ativada e o LED vermelho acende. O alarme de inclinação é ativado quando o sensor de inclinação deteta uma inclinação. Quando isso acontece, a campainha é ativada e os LEDs RGB são acionados, alternando entre as cores do arco-íris. O usuário pode desativar o alarme de inclinação pressionando o botão.

3. Código Desenvolvido

```
const int fotoresistorePin = 3;
const int buzzerPin = 13;
const int ledRed1 = 11;
const int buttonInclinacao = 12;
int estadoButton = 0;
const int sensorPin = 2;
bool alarmInclinacao = false;
const int redPin = 10; //Pino do RED
const int bluePin = 9; //Pino do BLUE
const int greenPin = 8; //Pino do GREEN
const bool invert = true;
int color = 0; // valor HUE
int R, G, B; // variáveis RGB
```

2 - Variáveis

Constantes para armazenar pinos dos dispositivos conectados:

- fotoresistorePin;
- buzzerPin;
- ledRed1;
- buttonInclinacao;
- sensorPin;
- RGB
    - redPin;
    - bluePin;
    - greenPin.

Armazenar o estado atual do botão de desativação do alarme de inclinação:

- estadoButton.

Armazenar o estado atual do alarme de inclinação (ativo ou inativo):

- alarmInclinacao.

Constante usada para inverter ou não o sinal de saída dos LEDs RGB:

- invert.

Armazenar o valor atual da cor que será mostrada pelos LEDs RGB:

- color.

Armazenar os valores dos componentes vermelho, verde e azul, respetivamente:

- R;
- G;
- B.

```
void setup() {
  Serial.begin(9600);

  pinMode(fotoresistorePin, INPUT); // configura o fotoresistore como entrada
  pinMode(ledRed1, OUTPUT); // configura o LED multicolorido como saída
  pinMode(buzzerPin, OUTPUT); // configura a campainha como saída
  pinMode(sensorPin, INPUT_PULLUP); // configura o detetor de inclinação como entrada com pull-up interno
  pinMode(buttonInclinacao , INPUT);

  attachInterrupt(digitalPinToInterrupt(fotoresistorePin), ativarAlarmeFogo, CHANGE); // ativa a interrupção para o fotoresistore
  attachInterrupt(digitalPinToInterrupt(sensorPin), ativarAlarmeInclinacao, FALLING); // ativa a interrupção para o detetor de inclinação
}
```

3 - Setup()

Na função setup(), a serial é inicializada e os pinos são configurados como entradas ou saídas. As interrupções são ativadas para o fotoresistore e o detetor de inclinação. Isso permite que o programa reaja imediatamente quando esses dispositivos detectarem uma mudança na luminosidade ou na inclinação, respectivamente.

```
void loop() {

  // ativa o alarme de inclinação
  if(alarmInclinacao == true)
  {
    digitalWrite(buzzerPin, HIGH);

    // Valor do brilho (0 - 255)
    int brightness = 255;

    hueToRGB(color, brightness);

    // Fornecer valores às variáveis
    analogWrite(redPin, R);
    analogWrite(greenPin, G);
    analogWrite(bluePin, B);

    color++;

    if (color > 255)
      color = 0;

    delay(5);
  }
  else
  {
    analogWrite(redPin, 0);
  analogWrite(greenPin, 0);
  analogWrite(bluePin, 0);
    digitalWrite(buzzerPin, LOW);
  }

  // desativa o alarme de inclinação
  estadoButton = digitalRead(buttonInclinacao);
  if (estadoButton == HIGH)
    {
        Serial.print("Alarme de inclinação desativado!");
        alarmInclinacao = false;
    delay(100);
    }
}
```

4- Loop()

Na função loop(), o alarme de inclinação é ativado ou desativado de acordo com o estado da variável alarmInclinacao. Se o alarme estiver ativo, a campainha e os LEDs RGB são acionados e a cor dos LEDs é alterada a cada 5 milissegundos. Se o alarme estiver inativo, a campainha e os LEDs RGB são desligados. O estado do botão de desativação do alarme
Também é verificado o estado do botão de desativação do alarme de inclinação. Se o botão estiver pressionado, o alarme é desativado e é exibida uma mensagem no monitor serial.

```
// esta função é chamada quando o fotoresistore detecta uma mudança brusca na luminosidade
void ativarAlarmeFogo() {
  // ativa a campainha e o LED multicolorido
  digitalWrite(buzzerPin, HIGH);
  digitalWrite(ledRed1, HIGH);

  // espera 1 segundo
  delay(1000);

  // desativa a campainha e o LED multicolorido
  digitalWrite(buzzerPin, LOW);
  digitalWrite(ledRed1, LOW);
}

// esta função é chamada quando o detetor de inclinação detecta uma inclinação
void ativarAlarmeInclinacao() {
  alarmInclinacao = true;
}
```

5 - Funções de ativar alarmes

As funções ativarAlarmeFogo() e ativarAlarmeInclinacao() são chamadas pelas interrupções correspondentes e realizam as ações específicas de cada alarme, ligando o buzzer e os LEDs correspondentes.

```
//Função de conversão da cor para os componentes RGB
void hueToRGB(int hue, int brightness) {
  //Hue à escala
  unsigned int scaledHue = (hue * 6);

  //Segmento de 0 a 5 (ciclo de cores)
  unsigned int segment = scaledHue / 256;

  //Posição no segmento
  unsigned int segmentOffset = scaledHue - (segment * 256);

  unsigned int complement = 0;
  unsigned int prev = (brightness * (255 - segmentOffset)) / 256;
  unsigned int next = (brightness * segmentOffset) / 256;

  if (invert) {
    brightness = 255 - brightness;
    complement = 255;
    prev = 255 - prev;
    next = 255 - next;
  }

  switch (segment) {
    case 0:  // red
      R = brightness;
      G = next;
      B = complement;
      break;
    case 1:  // yellow
      R = prev;
      G = brightness;
      B = complement;
      break;
    case 2:  // green
      R = complement;
      G = brightness;
      B = next;
      break;
    case 3:  // cyan
      R = complement;
      G = prev;
      B = brightness;
      break;
    case 4:  // blue
      R = next;
      G = complement;
      B = brightness;
      break;
    case 5:  // magenta
    default:
      R = brightness;
      G = complement;
      B = prev;
      break;
  }
}
```

6 - função hueToRgb

A função hueToRGB() é utilizada para converter o valor da cor em valores RGB para os LEDs.

4. Motivos de escolha

Um dos principais motivos pela escolha deste projeto deve-se ao facto de a ideia inicial do grupo ser a criação de um leitor de cartões RFID, termos adquirido todo o hardware e no final ter sido detetado danos nos mesmo. Para tal ocorrência foi decidido realizar um projeto que pudesse envolver material que tivéssemos disponível e então surgiu a ideia de criar alarmes de deteção de fogo e inclinação com recurso a Interrupts e alguns outputs para criar toda uma dinâmica.

4.1. Hardware

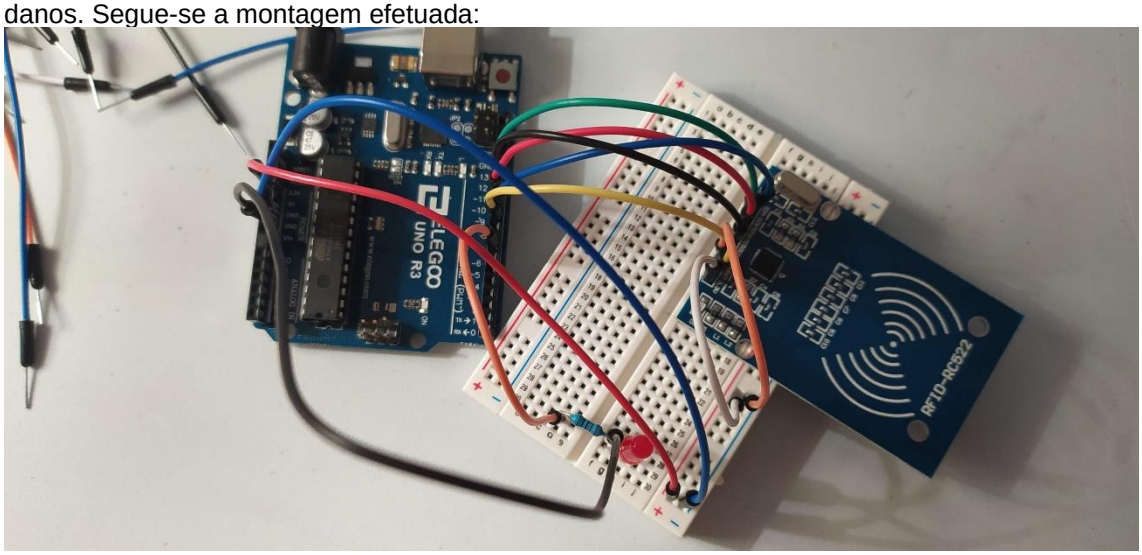
Para o hardware foi selecionada as peças que achamos mais interessantes em implementar para a nossa ideia, mas também algumas por necessidade de dependência para o bom funcionamento.

4.2. Software

Quanto ao software foi aplicado de forma que satisfizesse a arquitetura fornecida e de forma a realizar as pedidas Interrupts.

4.3. Antigo Projeto

De forma a não desperdiçar todo o projeto anterior desenvolvido foi realizado uma montagem em Arduino para um leitor de cartões RFID não tendo tido sucesso devido a danos. Segue-se a montagem efetuada:



7- Antigo projeto

Detetamos que o leitor se encontrava a receber energia mas devido a falta de tempo e recursos danificados (cartões), não progredimos mais neste e avançamos para outro projeto com os materiais que nos restaram.

5. Conclusão

Com o desenvolvimento deste projeto foi possível criar um sistema de alarme para deteção de fogo e inclinação utilizando o Arduino e pode ser útil em situações de emergência para alertar sobre possíveis incêndios ou situações de perigo de inclinação.

6. Bibliografia

- Aulas;
- Projeto anterior: https://www.youtube.com/watch?v=ARdQB9O372U&t=405s