

ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

INSTITUTO POLITÉCNICO DA GUARDA

Medição da intensidade do som através de sinais luminosos

Projeto final da unidade curricular

|  |  |
| --- | --- |
| **Curso** | Mestrado Computação Móvel |
| **Unidade Curricular** | Sistemas Embebidos |
| **Ano Letivo** | 2016/2017 |
| **Docente** | Luís Figueiredo |
| **Coordenador da área disciplinar** | António Martins |
| **Data** | 28-06-2017 |
| **Alunos** | Anabela Tavares Nº1011109  Rui Trigo Nº1011106 |

**Índice**

[1. Introdução 3](#_Toc486444563)

[2. Objetivos 3](#_Toc486444564)

[3. Análise do Problema 3](#_Toc486444565)

[3.1. Máquina de Estados para confirmar dados enviados 4](#_Toc486444566)

[4. Algoritmos 5](#_Toc486444567)

[4.1. Algoritmos Arduino 5](#_Toc486444568)

[4.1.1. Algoritmo Setup 5](#_Toc486444569)

[4.1.2. Algoritmo Loop 5](#_Toc486444570)

[4.1.3. Algoritmo acendeLed 5](#_Toc486444571)

[4.1.4. Algoritmo apagarLed 6](#_Toc486444572)

[4.1.5. Algoritmo abreLatas 6](#_Toc486444573)

[4.1.6. Algoritmo blackout 6](#_Toc486444574)

[4.1.7. Algoritmo equalizer 6](#_Toc486444575)

[5. Código 7](#_Toc486444576)

[6. Resultados 7](#_Toc486444577)

[7. Conclusão 7](#_Toc486444578)

# Introdução

Este relatório tem como objetivo documentar o projeto desenvolvido durante o semestre letivo nas aulas da unidade curricular Sistemas Embebidos.

A estrutura deste relatório encontra-se da seguinte forma, primeiramente os objetivos onde damos uma perspetiva do foi estipulado para a realização do projeto. Em seguida, uma pequena analise do problema onde particularizamos quais os desafios e metas que definimos logo após estudarmos o objetivo do projeto. Além disso, também nesta fase dizemos quais os problemas que fomos tendo ao longo do desenvolvimento do projeto.

Imediatamente depois, especificamos os passos que demos (algoritmos) e, a partir destes passos elaboramos o código necessário para que o projeto funcionasse. Por fim, mas não menos importante, demonstramos os resultados que obtivemos e, também uma conclusão sobre o projeto em si.

# Objetivos

Neste projeto foi construído um sistema de medição de som captado por um microfone de um dispositivo Android, que envia os dados para um microcontrolador Arduino através de Bluetooth. O microcontrolador, consoante os dados recebidos, desencadeia o acender de LEDs ligados a este.

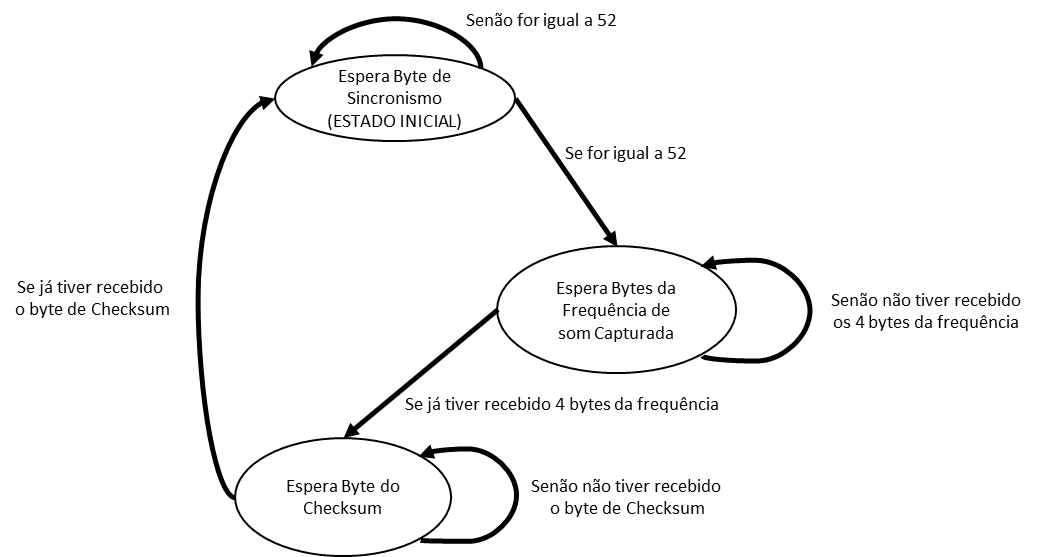
O sistema possui mecanismos de comunicação bidirecional, ou seja, nos dois sentidos. A comunicação é manipulada por botões ligados ao microcontrolador para que este, por sua vez, envie através de Bluetooth o modo desejado para o dispositivo Android. Este modo é alterado consoante o botão que é pressionado e faz com que seja aplicado um filtro nos valores enviados para o microcontrolador.

# Análise do Problema

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Byte* de Sincronismo | *Bytes* da Frequência do som capturado | | | | Bytes de checksum |
| 1 byte | 1 byte | 1 byte | 1 bytes | 1 byte | 1 byte |

O dispositivo Android vai enviar por Bluetooth um pacote composto por um valor inteiro precedido de um byte de sincronização (52) e um *checksum* para garantir fiabilidade dos dados.

* 1. Máquina de Estados para confirmar dados enviados



# Algoritmos

* 1. Algoritmos Arduino
     1. Algoritmo *Setup*

1. Configurar a porta 2 a 9 como saída;
2. Configurar a porta 11, 12, 13 como entrada;
3. Configurar a porta série 0 e 1 com um baud rate de 9600 Hz.
   * 1. Algoritmo *Loop*
4. Se a porta série 1 tiver dados disponíveis
   1. Ler os dados que estão na porta série;
   2. Chamar a função abre-latas para analisar os dados que estão na porta série.
5. Guardar o modo atual numa variável (modo);
6. Se o botão dos graves estiver pressionado
   1. Igualar o modo atual a 10;
7. Se o botão dos médios estiver pressionado
   1. Igualar o modo atual a 20;
8. Se o botão dos agudos estiver pressionado
   1. Igualar o modo atual a 30;
9. Senão estiver nenhum botão pressionado
   1. Igualar o modo atual a 0;
10. Se o modo for diferente do modo atual
    1. Escrever na porta serie 1 o byte de sincronização (52);
    2. Escrever na porta serie 1 o modo atual;
       1. Algoritmo acendeLed

1. Ligar o led correspondente;

* + 1. Algoritmo apagarLed

1. Desligar o led correspondente;

* + 1. Algoritmo abreLatas
       1. Se o estado for igual a 0;
  1. Se o primeiro byte recebido for igual a 52
     1. Igualar o estado a 1;
     2. Igualar o contador a 0;

1. Se o estado for igual a 1
   1. Guardar os bytes recebidos;
   2. Incrementar o contador;
   3. Se o contador for igual a 4
      1. Guardar os bytes recebidos numa variável (valorRecebido);
   4. Se o contador for igual a 5
      1. Guardar o ultimo byte recebido;
      2. Enquanto o contador for menor que 4
         1. Somar os bytes recebidos;
      3. Se o ultimo bytes recebido for igual à soma dos bytes recebidos;
         1. Chamar a função blackout;
         2. Chamar a função equalizer;
      4. Igualar o estado a 0;
      5. Algoritmo blackout

1. Chamar a função apagarLed;

* + 1. Algoritmo equalizer

1. Se o valor recebido for maior ou igual que 0
   1. Chamar a função acenderLed;
   2. Se o valor recebido for maior ou igual que o limite dos graves a dividir por 3
      1. Chamar a função acenderLed;
      2. Se o valor recebido for maior ou igual que o limite dos graves a dividir por 3 \* 2
         1. Chamar a função acenderLed;
         2. Se o valor recebido for maior ou igual que o limite dos graves
            1. Chamar a função acendeLed;
            2. Se o valor recebido for maior ou igual que o limite dos médios a dividir por 3

Chamar a função acendeLed;

Se o valor recebido for maior ou igual que o limite dos médios a dividir por 3 \* 2

Chamar a função acendeLed;

Se o valor recebido for maior ou igual que o limite dos médios

Chamar a função acendeLed;

Se o valor recebido for maior ou igual que o limite dos médios \* 2

Chamar a função acendeLed;

# Código

# Resultados

# Conclusão