

Redes de Computadores

2º. Ano da Licenciatura em Informática de Gestão

Arduíno

Ano letivo 2018/2019

Autores:

Hugo Parreiral nº. 16532

Joana Figueiredo nº. 16530

Thayná Pereira nº. 16537

Elaborado em

Janeiro de 2018

Índice

Introdução	3
Aplicação Find	4
Aplicação Roboremo	5
Dificuldades encontradas	5
Explicação do código – Configuração para o Wifi e função <i>setup</i>	6
Explicação do código – Função <i>loop</i>	7
Conclusão	9
Referências	10

Introdução

No âmbito da disciplina de Redes de Computadores, foi proposta a realização de um trabalho prático dividido em 2 partes bem distintas. Primeiramente foi proposto testar a viabilidade da Aplicação Find. Esta aplicação permite recolher um conjunto de assinaturas digitais em diversos espaços do ISCAC (salas de aula, corredores, bar, etc) para mais tarde, através da mesma podermos validar a precisão do algoritmo de localização para um conjunto de espaços registados anteriormente.

A segunda parte do trabalho a desenvolver consiste na manipulação de um LED azul controlado através de uma aplicação designada Roboremo, utilizando para isso o microcontrolador ESP8266. Para a realização desta parte do trabalho foi fundamental escolher um bom ambiente de desenvolvimento, por isso instalamos o Arduino que nos permite programar todo o código, para que o LED acenda, apague, pisque uma vez por segundo, e ainda, varie a sua intensidade. Todas estas instruções são comandadas através da aplicação que se encontra conectada a uma rede Wifi.

Aplicação Find

O trabalho foi realizado por etapas, a primeira parte do trabalho consistia em instalar a aplicação Find no smartphone. De seguida o docente atribuiu ao nosso grupo duas salas a 2.7 e 3.7. O nosso grupo após a atribuição das salas tinha como objetivo recolher um conjunto de assinaturas digitais nas respetivas salas e a calendarização dos mesmos.

Para poder efetuar a recolha das assinaturas digitais tivemos que nos dirigir às salas atribuídas e efetuar os seguintes passos:

Exemplo da execução do processo com uma sala proposta:

Direcionámo-nos para o meio da sala 2.7 e preenchemos os requisitos abaixo para poer fazer a recolha da assinatura digital.

Group: mcv_dev

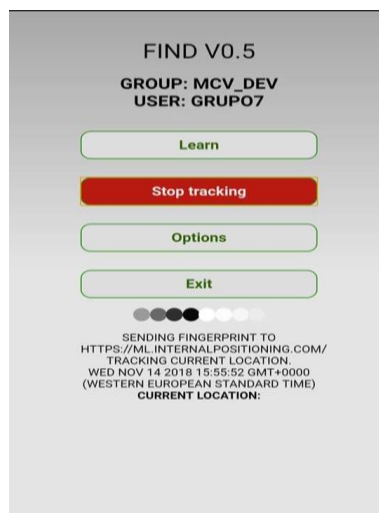
User: Grupo7

server: <http://socialiteorion2.dei.uc.pt:9012/>

exemplo de localização: S207

De seguida colocámos o smartphone sobre a mesa e carregámos no botão “learn”.

Para concluir este ponto do trabalho, o professor realizou uma troca de salas pelos diferentes grupos. As salas que nos foram atribuídas foram: 2.3 e 3.3, posteriormente tivemos que validar a precisão do algoritmo de localização. Para isso dirigimos-nos às respetivas salas, colocámos o smartphone sobre uma mesa no meio da sala e carregámos no botão “track”.



Ao realizar esta etapa do trabalho o nosso grupo não conseguiu localizar nenhuma das salas atribuídas, ou seja, não foi possível precisar a localização do algoritmo.

Aplicação Roboremo

Optamos por escolher a aplicação Roboremo que permita controlar o led.

Para o desenvolvimento desta aplicação que permite controlar o led azul, é feita através de um telemóvel que utilizamos a aplicação. Essa aplicação permite controlar o led através de botões com as seguintes operações:

- Acender led;
- Apagar led;
- Piscar led uma vez por segundo;
- Variar intensidade;

A conexão entre a aplicação Roboremo e a rede Wifi é feita através de uma instrução que permite preencher o endereço IP, o qual será estabelecido através do Monitor serial do Software Arduino, que neste caso está definido como 192.168.0.1:1234.

Cada um dos botões da aplicação Roboremo está configurado com um comando definido através da “*set press action*” que está no menu da aplicação para editar um botão.

A imagem a seguir mostra uma melhor exemplificação do que essa aplicação permite para a elaboração do trabalho:



Dificuldades encontradas

Para a realização deste trabalho encontramos algumas dificuldades com alguns erros que nos iam aparecendo ao longo da realização do trabalho, como a ligação entre o Arduino e a rede wifi.

As dificuldades neste trabalho desde inicio foi encontrar informações na internet, por parecer que pudéssemos encontrar muita coisa, mas afinal a informação disponível era pouca.

Explicação do código – Configuração para o Wifi e função setup

O primeiro passo é conectar o ESP8266 à rede local. Para isso criam-se variáveis com o SSID e com a password. De seguida, cria-se um servidor no próprio ESP8266, onde é necessário criar uma variável do tipo WiFiServer, indicando também qual a porta que irá ser utilizada pelo ESP8266. Sempre que o ESP8266 está conectado à internet é-lhe destinado um IP local que esteja disponível. Todas estas variáveis são criadas antes da declaração da função obrigatória setup() no início do script.

Na função Setup serão chamados todos os comandos de inicialização do programa. Portanto, esse programa irá:

1. Iniciar a comunicação serial com o PC.
2. Declarar os pinos utilizados como saídas.
3. Conectar o ESP8226 ao WiFi.
4. Iniciar o servidor.

Primeiramente deve-se indicar a velocidade da comunicação serial (baud rate). Isso é determinado pelo comando "Serial.begin()". O baud rate deve ser colocado a 115200 bits/s. Isto permitirá uma melhor comunicação com o computador, sem problemas na hora de se verificar o monitor serial. É válido ressaltar que o número de pinos designados como saídas depende do número de lâmpadas que desejam ser acionadas, no nosso caso, temos apenas um led a controlar.

```
#include <ESP8266WiFi.h> //INCLUSÃO DA BIBLIOTECA NECESSÁRIA PARA FUNCIONAMENTO DO CÓDIGO
#include <WiFiClient.h>
```

```
// CONFIG:
```

```
const char *ssid = "RoboRemo"; //VARIÁVEL QUE ARMAZENA O NOME DA REDE SEM FIO EM QUE VAI CONECTAR
```

```
const char *pw = ""; //VARIÁVEL QUE ARMAZENA A SENHA
```

```
IPAddress ip(192, 168, 0, 1); //PARTE 1 DA CONEXÃO DO ROBOREMO
```

```
IPAddress netmask(255, 255, 255, 0);
```

```
const int port = 1234; //PARTE 2 DA CONEXÃO DO ROBOREMO → PARTE 1 + 2 → 192.168.0.1:1234 (chamada da URL → IP:PORTA)
```

```
WiFiServer server(port);
```

```
WiFiClient client;
```

```
void setup() {
```

```
// DECLARAR OS PINOS UTILIZADOS COMO SAÍDAS.
```

```
pinMode(LED_BUILTIN,OUTPUT);
```

```
Serial.begin(115200); //INICIALIZA A SERIAL
```

```
WiFi.softAPConfig(ip, ip, netmask); // CONFIGURA O ENDEREÇO IP PARA SOFTAP
```

```
WiFi.softAP(ssid, pw); // CONFIGURA O SSID E A PASSWORD PARA SOFTAP
```

```
server.begin(); //INICIA O SERVIDOR PARA RECEBER DADOS NA PORTA DEFINIDA EM "WiFiServer  
server(porta)"
```

```
Serial.println("ESP8266 RC receiver 1.1 powered by RoboRemo");
```

```
Serial.println((String)"SSID: " + ssid + " PASS: " + pw); //ESCREVE O NOME DA REDE NA SERIAL E  
DA RESPETIVA PASS
```

```
Serial.println((String)"RoboRemo app must connect to " + ip.toString() + ":" + port); //ESCREVE  
NA SERIAL O IP RECEBIDO DENTRO DA REDE SEM FIO
```

Explicação do código – Função *loop*

```
void loop() {                                     // Vai retornar o número de bytes disponíveis,  
    if(!client.connected()) {                     ou seja, a quantidade de dados que foram  
        client = server.available();              gravados no cliente pelo servidor ao qual está  
        return;                                   conectados.
```

```
    }

    // If client connected                         // se o cliente for conectado ao servidor, terá dados disponíveis  
                                                para a leitura.
    if(client.available()) {                       // O arduino mandará esta mensagem para o ecrã do  
        Serial.println("connect client with code: ");      computador

        char c = (char)client.read(); // Leitura do carácter introduzido pelo cliente

        Serial.println(c);

        if(c == 'A'){ // Se o carácter introduzido pelo cliente for o 'A'

            digitalWrite(LED_BUILTIN, HIGH); // // O LED está conectado a um pino digital, e  
                                                o seu número poderá variar de placa para  
                                                placa. Então o LED_BUILTIN permite  
                                                controlar facilmente o LED.

                                                // O comando LED_BUILTIN, HIGH permite  
                                                que o LED ligue.

            else if(c == 'B'){

                digitalWrite(LED_BUILTIN, LOW); // // Se o cliente introduzir o carácter 'B': o  
                                                comando LED_BUILTIN, LOW irá permitir  
                                                desligar o LED.
```

```

else if(c == 'C'){
    int intensidade= 0;
    do{
        digitalWrite(LED_BUILTIN, HIGH);
        analogWrite(LED_BUILTIN, intensidade);
        delay(30);
        intensidade +=5;
    }while(intensidade <=1000);
    intensidade= 1000;
    do{
        analogWrite(LED_BUILTIN,intensidade);
        delay(30);
        intensidade -=5;
    }while(intensidade >=0);
}

```

// Se o cliente introduzir o caracter 'C': o LED irá iniciar o seu "processo" com intensidade 0.

De seguida é criado um ciclo para que seja possível haver uma variação de intensidade.

"analogWrite ()" → o LED terá uma luz variável ou ativará um motor com várias velocidades.

Inicia com o LED ligado, permanece 0,03 segundos com a intensidade inicial.

Ao longo do decurso a intensidade vai sempre aumentando de cinco em cinco volts ate chegar aos mil volts. Quando chegar ao máximo da intensidade possível aguarda 0.03 segundos e começa a diminuir com o mesmo intervalo de volts com que aumentou até chegar à intensidade 0.

```

else if(c == 'D'){
    for( int i=0; i<5; i++){
        digitalWrite(LED_BUILTIN, HIGH);
        delay(1000);
        digitalWrite(LED_BUILTIN, LOW);
        delay(1000);
    }
}

```

// Quando for introduzido o caracter 'D': o LED irá piscar cinco vezes segundo a segundo. Da seguinte maneira:

digitalWrite(LED_BUILTIN, HIGH); → Desliga o LED.

delay(1000); → Aguarda um segundo.

digitalWrite(LED_BUILTIN, LOW); → Liga o LED.

delay(1000); → Aguardará novamente um segundo.

Repetindo este ciclo cinco vezes de acordo com as restrições do professor.

Conclusão

Apesar de cumpridas todas as etapas enunciadas pelo professor, o resultado final obtido com a utilização da aplicação Find não foi o que esperávamos, pois não conseguimos precisar a localização do algoritmo das salas atribuídas ao nosso grupo.

A realização da segunda parte do trabalho foi muito mais satisfatória, pois apesar das dificuldades enunciadas, conseguimos realizar todas as operações pedidas com êxito.

Referências

<https://www.circuitar.com.br/tutoriais/programacao-para-arduino-primeiros-passos/>

<https://www.instructables.com/id/RoboRemo-ESP8266-Easy-Control/>

<http://blogmasterwalkershop.com.br/embarcados/nodemcu/nodemcu-como-criar-um-web-server-e-conectar-a-uma-rede-wifi/>