

Relatório do funcionamento do Roteamento de sinais digitais

Integrantes:

-Ana Carolina Baldoni Abrahão Rigatto
-Enzo Ribeiro Moraes
-João Pedro Maciel Freitas
-Samuel de Jesus Faria

Turma:31-TE

Este relatório foi desenvolvido com intuito de comprovar através de fatos o funcionamento do **equipamento de roteamento de sinais digitais entre redes**.

Tópico 1 – MT8816

O MT8816 é um comutador analógico de 16 canais. Este dispositivo permite rotear sinais analógicos de uma variedade de fontes para uma variedade de destinos. Isto é útil em sistemas de comutação de circuitos, onde as conexões físicas precisam ser estabelecidas entre diferentes pontos. Devido a sua capacidade de comutação rápida o MT8816 precisa de sinais analógicos para transmitir informações.

Funcionamento: Para fazer ligações entre os terminais do CI, é necessário inserir o endereçamento de cada porta a ser ligada, X e Y. Após isso, precisa-se executar a função “conect();” que junta o endereço de cada porta e envia um comando para o pino de Data, e em seguida para o pino Strobe, a fim de fazer a ligação. As funções principais definidas no código para controlar a matriz de comutação são: setaddress, resetaddress, conect, desconect, resetmatriz.

[Link do vídeo do CI funcionando](#)

Endereçamento de cada porta:

```

int BROADCAST[3] = {0, 0, 0}; //X0 33

int X1[3] = {1, 0, 0}; //X1 32
int X2[3] = {0, 1, 0}; //X2 31
int X3[3] = {1, 1, 0}; //X3 30
int X4[3] = {0, 0, 1}; //X4 29

int L1[2] = {0, 0}; //Y0 35
int L2[2] = {1, 0}; //Y1 37
int L3[2] = {0, 1}; //Y2 39
int L4[2] = {1, 1}; //Y3 1

```

Função que conecta os pinos do CI:

```

void conect(int wan[], int lan[])
{
    setaddress(wan, lan);
    digitalWrite(DATA, HIGH);
    digitalWrite(STROBE, HIGH);
    delay(deleio);
    digitalWrite(STROBE, LOW);
    resetaddress();
    digitalWrite(DATA, LOW);
}

```

Tópico 2 – Código

O código usa uma página HTML inserida dentro da memória do ESP, que pode ser acessada por qualquer dispositivo ligado à mesma internet que o microcontrolador, para que o usuário insira os valores de latitude e longitude de sua localização, e se deseja usar a comunicação interna ou externa.

Se o local for externo, o código define duas APIs para consultar, sendo estas um servidor de previsão do tempo (Open meteo) para obter informações sobre a chuva na localização fornecida e se conecta a um segundo servidor (192.168.100.122) para obter informações da largura de banda de interface de cada roteador, a fim de calcular qual a melhor rota a ser escolhida.

Se o tipo de local for interno, o programa ativa todos os canais de transmissão simultaneamente, fazendo assim uma comunicação Broadcast entre as saídas LANs.

[Vídeo que demonstra a escolha da localização e da rota \(interno ou externo\).](#)

Trecho do código que define os valores inseridos pelo usuário na página HTML em variáveis:

```
server.on("/", HTTP_GET, [] (AsyncWebServerRequest * request) {
    request->send(200, "text/html", "<form method='post' action='/submit'>"
        "Latitude: <input type='text' name='latitude'><br>"
        "Longitude: <input type='text' name='longitude'><br>"
        "Tipo de Local: externo ou interno: <input type='text' name='tipo-local'><br>"
        "<input type='submit' value='Enviar'>"
        "</form>");
});

server.on("/submit", HTTP_POST, [] (AsyncWebServerRequest * request) {

    latitude = request->arg("latitude");
    longitude = request->arg("longitude");
    tipoLocal = request->arg("tipo-local");

    request->send(200, "text/plain", "Dados recebidos com sucesso!");
});
```

Requisição que o ESP faz à API do servidor que informa a largura de banda de interface de cada roteador:

```
String message = "GET /";
message = message + " HTTP/1.1\r\n";

client.print(message);
client.print("Host: 192.168.100.177\r\n");
client.print("Connection: close\r\n");
client.print("\r\n\r\n");
```

Requisição que o ESP faz à API do Open Meteo, a fim de saber se está chovendo ou não:

```
String msg = "GET /v1/forecast?";
msg = msg + "latitude=" + String(latitude) + "&";
msg = msg + "longitude=" + String(longitude) + "&";
msg = msg + "current=rain&";
msg = msg + "timezone=auto";
msg = msg + " HTTP/1.1\r\n";
Serial.print(msg);

client.print(msg);
client.print("Host: api.open-meteo.com\r\n");
client.print("Connection: close\r\n");
client.print("\r\n\r\n");
```

