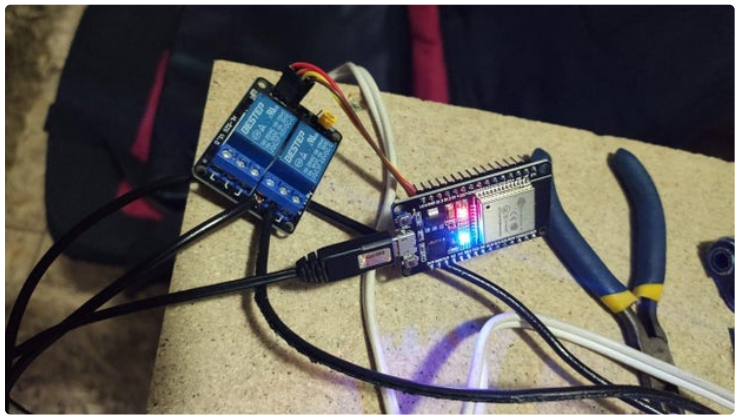
Controle Acionamento De ESP32 Por Comando De Voz

by mathbarc

O presente trabalho foi desenvolvido por **Matheus Barcelos de Oliveira** durante a aula de Internet das Coisas (IoT) lecionada pelo professor **Ilo Amy Saldanha Rivero**, no curso de Pós Graduação MBA em Indústria 4.0 da Instituição PUC Minas Unidade Coração Eucarístico.

Supplies:

- Um computador para hospedar um servidor **Mosquitto**(broker MQTT);
- Uma dispositivo **ESP32**;
- Uma placa contendo dois relés com acionamento de 5 volts;
- Um computador para realizar captura e conversão de áudio de áudio para texto.

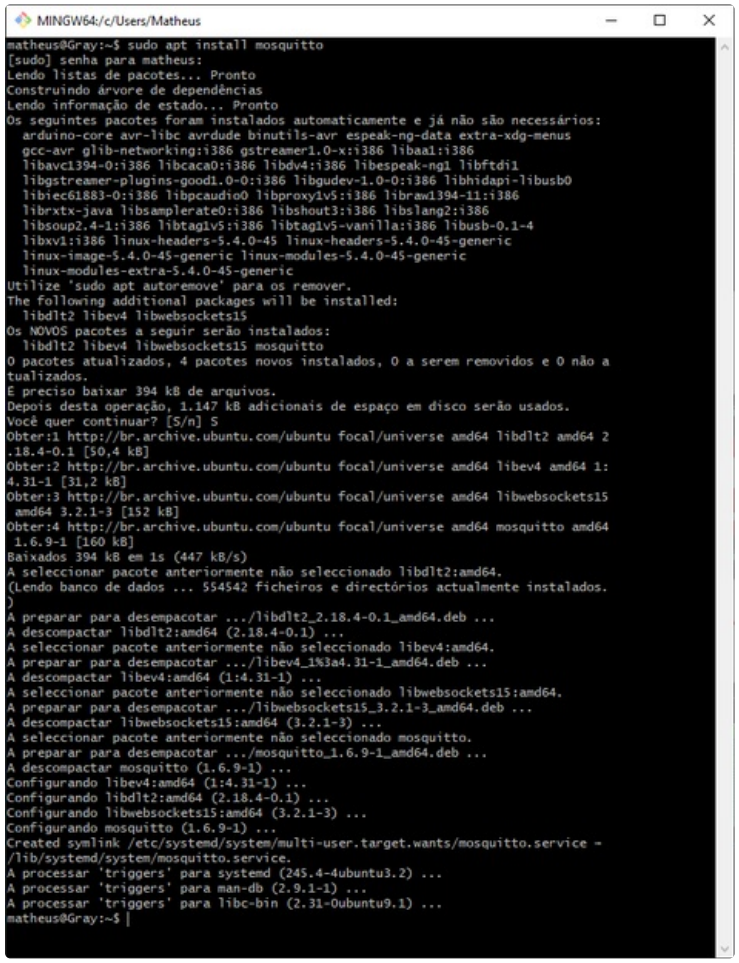


Step 1: Instalação Do Servidor MQTT

Para hospedar o broker MQTT foi utilizada uma máquina Ubuntu 20.04 e para instalá-lo o seguinte comando foi utilizado:

- *sudo apt install mosquitto*

Obtendo uma resposta conforme imagem acima.

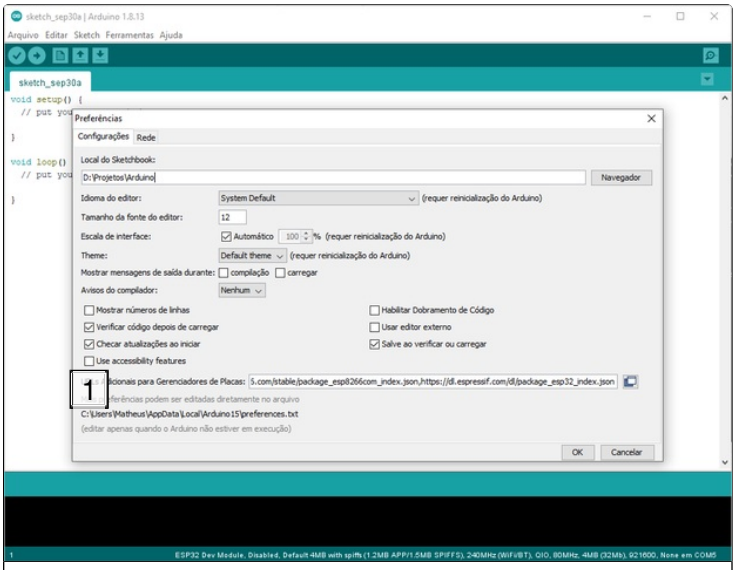


Step 2: Configurando Arduino IDE

Para programar o dispositivo ESP32 foi escolhida a **Arduino IDE** devido à sua simplicidade.

Após o download e instalação da ferramenta é necessário adicionar a seguinte linha ao item presente em *Arquivo->Preferências->Urls Adicionais para Gerenciadores de Placas*:

- https://arduino.esp8266.com/stable/package_esp8266com_index.json,https://dl.espressif.com/dl/package_esp32_index.json



1. Adicionar urls para suporte a ESP32

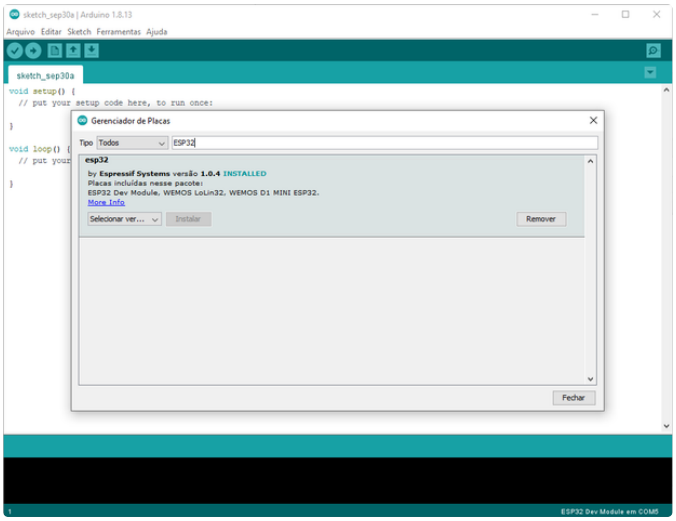
Step 3: Instalando Suporte À ESP32 No Arduino IDE

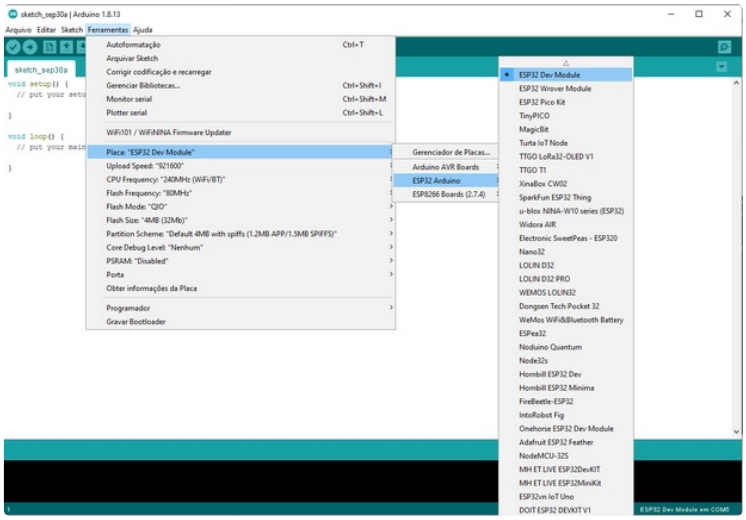
Com as urls configuradas do passo anterior, é possível acessar o gerenciador de placas no caminho *Ferramentas->Placa->Gerenciador de Placas* o que abrirá a tela conforme imagem acima.

Escrevendo ESP32 no filtro conforme a imagem, aparecerá o item esp32 by Espressif Systems, sendo necessário apenas instalá-lo.

Após instalá-lo, é necessário selecionar a placa em utilização clicando no segunte caminho: *Ferramentas->Placa->ESP32 Arduino->ESP32 Dev Module*

Após esta sequencia de passos já será possível programar o ESP32 utilizando Arduino IDE e conectando a placa ao computador por meio do cabo USB.

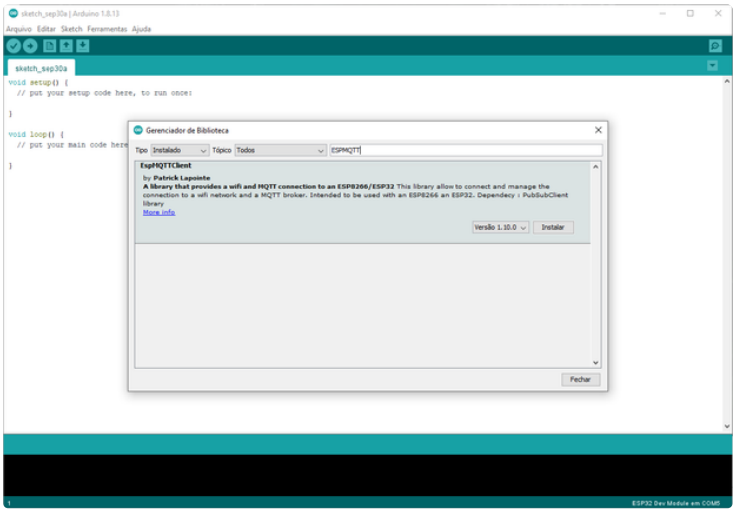




Step 4: Programando ESP32 Para Acionamento a Partir De MQTT

Após configurada a IDE, é necessário instalar a biblioteca EspMqttClient, para isso é necessário navegar para o seguinte caminho: *Ferramentas->Placa->ESP32 Arduino->ESP32 Dev Module*, o que fará a tela na imagem acima surgir.

Após a instalação já é possível desenvolvermos o firmware que será executado no dispositivo.



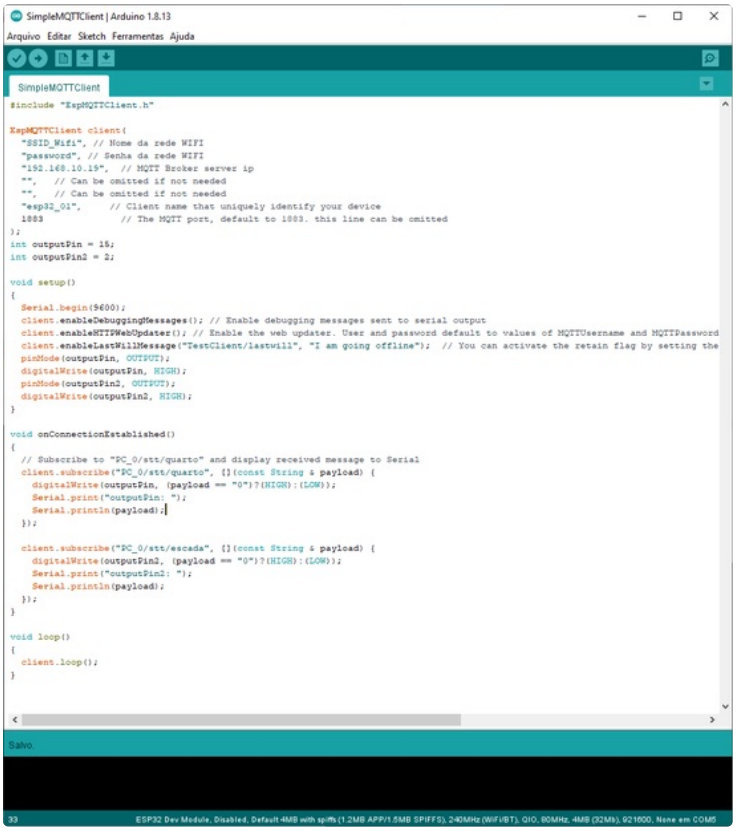
Step 5: Desenvolvimento Do Firmware

O programa de modelo SimpleMQTTClient foi modificado para permitir a subscrição em dois tópicos que serão publicados pelo dispositivo que interpretará os comandos de voz. A versão final utiliza no dispositivo ESP32 está contida na imagem.

Neste código foi criada um objeto da classe EspMQTTClient denominado client. Este cliente será responsável por buscar novas publicações disponibilizadas pelo broker. Além da criação do objeto client, é importante definir a função "*void onConnectionEstablished()*", pois esta função é responsável por realizar inscrições em assuntos disponíveis no broker ao se realizar uma conexão com sucesso. No caso foram realizadas inscrições nos assuntos

- PC_0/stt/quarto
- PC_0/stt/escada

e definidas funções lambda que tratarão a atualização dos estados de cada saída digital (pinos D15 e D2) conectadas à placa de acionamento de relés.



Step 6: Script Python Para Comandos De Voz

Após a programação do dispositivo ESP32, é necessário criar uma entidade que publique os estados a serem atualizados no dispositivo. Para isso foi criado um script Python que utiliza Speech to Text para transcrever áudio captado por um microfone instalado a um computador e quando reconhecido o comando, o estado é publicado vai MQTT para o dispositivo ESP32.

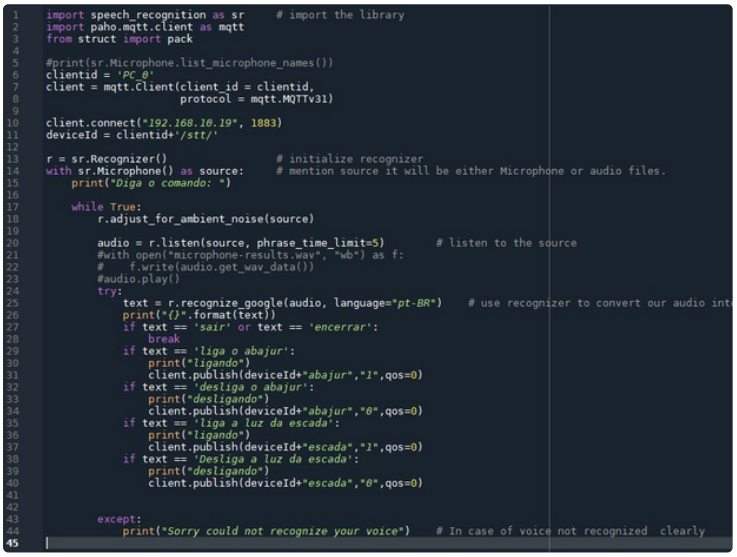
Para isso serão necessários pacotes que podem ser instalados com os seguintes comandos:

- sudo pip install speech_recognition pyaudio paho_mqtt

ou para sistemas Windows:

- *pip install pipwin*
- *pipwin install pyaudio*
- *pip install speech_recognition paho_mqtt*

Após instalados os pacotes o script contido na imagem acima cria um objeto que encapsula um cliente MQTT e utilizando a ferramenta de transcrição de áudio do GCP (Google Cloud Platform) verifica se no áudio os comandos foram ditos e assim realiza publicações nos respectivos assuntos MQTT.



```
import speech_recognition as sr     # import the library
import paho.mqtt.client as mqtt
from struct import pack

#print(sr.Microphone.list_microphone_names())
clientid = 'PC_0'
client = mqtt.Client(client_id = clientid,
                     protocol = mqtt.MQTTv31)

client.connect("192.168.10.19", 1883)
deviceId = clientid+'/stt/'

r = sr.Recognizer()                 # initialize recognizer
with sr.Microphone() as source:     # mention source it will be either Microphone or audio files.
    print("Diga o comando: ")

    while True:
        r.adjust_for_ambient_noise(source)

        audio = r.listen(source, phrase_time_limit=5)       # listen to the source
        #with open("microphone-results.wav", "wb") as f:
        #    f.write(audio.get_wav_data())
        #audio.play()
        try:
            text = r.recognize_google(audio, language="pt-BR")    # use recognizer to convert our audio int
            print("{}".format(text))
            if text == 'sair' or text == 'encerrar':
                break
            if text == 'liga o abajur':
                print("ligando")
                client.publish(deviceId+"abajur","1",qos=0)
            if text == 'desliga o abajur':
                print("desligando")
                client.publish(deviceId+"abajur","0",qos=0)
            if text == 'liga a luz da escada':
                print("ligando")
                client.publish(deviceId+"escada","1",qos=0)
            if text == 'Desliga a luz da escada':
                print("desligando")
                client.publish(deviceId+"escada","0",qos=0)


        except:
            print("Sorry could not recognize your voice")    # In case of voice not recognized  clearly
```

Step 7: Resultado

Após todos os passos anteriores foi possível atingir o resultado esperado e realizar o acionamento de um abajur utilizando comandos de voz conforme o seguinte vídeo:

https://www.youtube.com/embed/SGRybBWM_gc