



TFT – Internet das Coisas

Abstract

This article discusses the integration of the MQTT protocol with a TFT display using ESP32 as a development platform. The objective is to establish the MQTT communication between the ESP32 and a public MQTT broker, allowing the display of messages on the TFT display. The steps to configure the development environment will be presented, including the necessary libraries, and the implementation of the code to connect the ESP32 to the Wi-Fi network, establish the MQTT connection with the broker and display messages on the TFT display. The article highlights the importance of using MQTT in IoT projects, as well as the versatility and ease of integration of ESP32 with external devices.

Through this article, readers will be able to understand and implement MQTT communication and message visualization on a TFT display using ESP32, opening possibilities for the development of monitoring and control projects in real time.

Introdução

A comunicação MQTT (Message Queuing Telemetry Transport) é um protocolo leve de mensagens utilizado para a troca de informações entre dispositivos conectados à internet. Neste artigo, vamos explorar a integração do protocolo MQTT com um display TFT utilizando o ESP32, uma plataforma amplamente utilizada para o desenvolvimento de projetos IoT (Internet of Things).

Objetivo

O objetivo deste artigo é demonstrar como estabelecer a comunicação MQTT entre o ESP32 e um broker MQTT público, e utilizar essa comunicação para exibir mensagens em um display TFT.

Materiais necessários

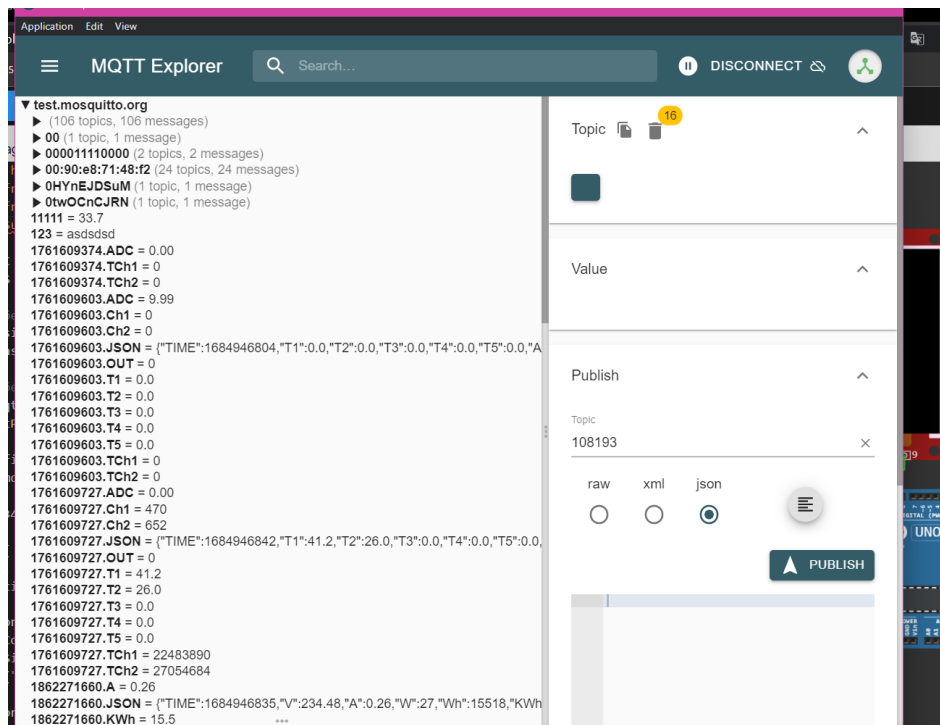
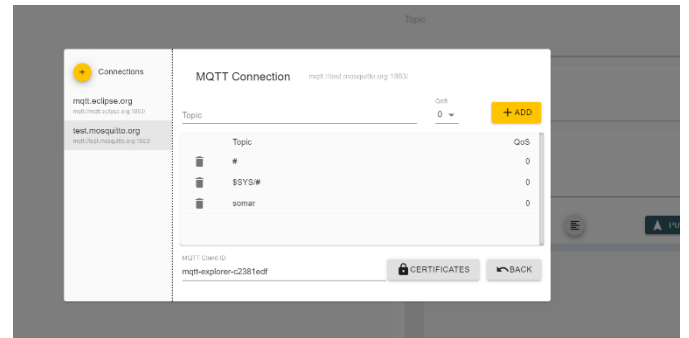
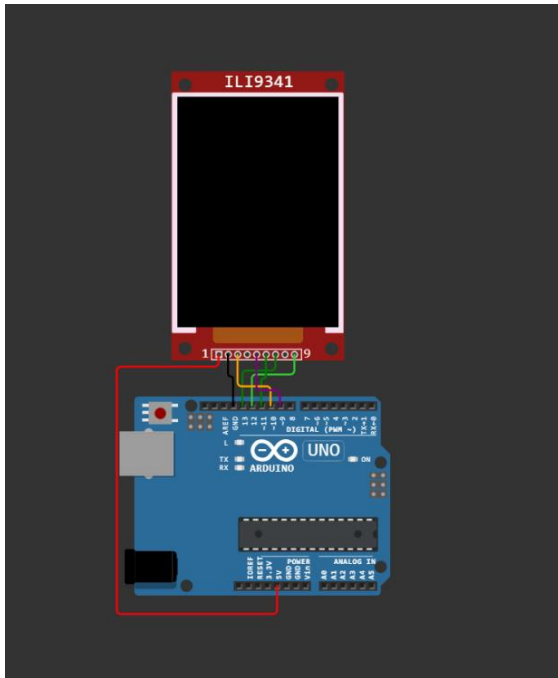
1. ESP32 (ou placa compatível)
 2. Display TFT ILI9341
 3. Cabos de conexão
 4. Acesso à rede Wi-Fi
-
1. Materiais:
 - ESP32 ou placa de desenvolvimento compatível
 - Ecrã TFT ILI9341



- Fios de jumper
 - Acesso à rede wi-fi
2. Configuração do software:
- Instale o Arduino IDE: Baixe e instale o Arduino IDE do site oficial do Arduino (<https://www.arduino.cc>).
 - Instale o pacote de suporte da placa ESP32: Siga as instruções fornecidas pela documentação oficial do ESP32 (<https://docs.espressif.com/projects/arduino-esp32/en/latest/>).
 - Instalar as bibliotecas necessárias: Instale as seguintes bibliotecas usando o Arduino Library Manager:
 - SPI
 - Adafruit_GFX
 - Adafruit_ILI9341
 - PubSubClient
3. Configuração de hardware:
- Conecte o display TFT ILI9341 ao ESP32 de acordo com a configuração de pinos definida no código. Normalmente, o pino TFT_DC é conectado ao pino 9 e o pino TFT_CS é conectado ao pino 10. Consulte a folha de dados ou o diagrama de pinagem de seu monitor específico para obter as conexões corretas.
 - Certifique-se de que o ESP32 esteja conectado corretamente ao seu computador via USB para programação e alimentação.
4. Configuração do corretor MQTT:
- Escolha um corretor MQTT adequado. Neste exemplo, usamos "test.mosquitto.org" como corretor.
 - Certifique-se de que o broker MQTT esteja acessível e em execução. Se você estiver usando um agente local, verifique se ele está sendo executado na sua rede ou localmente no mesmo dispositivo.
5. Implementação de Software:
- Inclua as bibliotecas necessárias no início do código: SPI.h, Adafruit_GFX.h, Adafruit_ILI9341.h e PubSubClient.h.
 - Defina a configuração de pinos para a tela TFT, credenciais de Wi-Fi e detalhes do servidor MQTT.
 - Configure a conexão Wi-Fi usando o SSID e a senha fornecidos.
 - Configure o cliente MQTT e configure-o com o endereço e a porta do servidor MQTT.
 - Conecte-se ao broker MQTT e assine o(s) tópico(s) desejado(s) para receber mensagens.



- Na função `loop()`, chame o método `mqttClient.loop()` para manter a conexão MQTT.
 - Implemente a lógica para exibir as mensagens MQTT recebidas na tela TFT. Você pode personalizar a aparência e o comportamento com base em seus requisitos específicos.
6. Teste e Execução:
- Faça o upload do código para o ESP32 usando o Arduino IDE.
 - Abra o Serial Monitor para monitorar a saída do ESP32 e verifique a conexão Wi-Fi bem-sucedida e a conexão MQTT com o corretor.
 - Publique mensagens MQTT no(s) tópico(s) inscrito(s) de outro dispositivo ou cliente MQTT.
 - Verifique se as mensagens são recebidas e exibidas corretamente no visor TFT.
7. Resultados e Avaliação:
- Observe a saída no visor TFT e certifique-se de que as mensagens MQTT recebidas sejam exibidas com precisão.
 - Avalie a funcionalidade geral e o desempenho do sistema, incluindo a conexão Wi-Fi, a comunicação MQTT e a funcionalidade de exibição.
 - Faça quaisquer ajustes ou modificações necessárias no código ou na configuração do hardware com base nos resultados obtidos.
8. Conclusão:
- Resuma a integração bem-sucedida da comunicação MQTT com o display TFT usando o ESP32.
 - Discuta quaisquer limitações ou desafios encontrados durante a implementação.
 - Destaque as possíveis aplicações e benefícios dessa configuração em projetos de IoT e sistemas de monitoramento e controle em tempo real.



Código da execução:

```
#include <SPI.h>
#include
<Adafruit_GFX.h>
#include
```

```
<Adafruit_ILI9341.h>
#include <PubSubClient.h>

#define TFT_DC 9
#define TFT_CS 10

// Configurações da rede Wi-Fi
const char* ssid = "wokwi-GUEST";
const char* password = "";
```



```
// Configurações do broker MQTT
const char* mqttServer = "test.mosquitto.org";
const int mqttPort = 1883;

WiFiClient wifiClient;
PubSubClient mqttClient(wifiClient);

Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);

void setup() {
  tft.begin();
  tft.setRotation(3); // Ajusta a rotação da tela, se necessário

  tft.setCursor(26, 120);
  tft.setTextColor(ILI9341_RED);
  tft.setTextSize(3);
  tft.println("Hello, TFT!");

  tft.setCursor(20, 160);
  tft.setTextColor(ILI9341_GREEN);
  tft.setTextSize(2);
  tft.println("I can has colors?");

  // Conecta-se à rede Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Conectando ao Wi-Fi...");
  }

  // Configura o servidor e as credenciais do MQTT
  mqttClient.setServer(mqttServer, mqttPort);
  mqttClient.setCredentials(mqttUser, mqttPassword);

  // Conecta-se ao broker MQTT
  while (!mqttClient.connected()) {
    if (mqttClient.connect("ESP32Client")) {
      Serial.println("Conectado ao broker MQTT");
    } else {
      Serial.print("Falha na conexão com o broker MQTT - Tentando novamente em 5 segundos...");
      delay(5000);
    }
    mqttClient.subscribe("somar");
  }
}

void loop() {
```



```
mqttClient.loop();  
  
}
```

Conclusão

- Os objetivos propostos não foram totalmente alcançados com o código fornecido. Embora o código estabeleça a conexão Wi-Fi, conexão MQTT e exiba mensagens na tela TFT, ele não implementa a lógica para receber e processar as mensagens MQTT no tópico "somar" ou executar ações com base nessas mensagens. Portanto, o código precisa ser expandido e personalizado para atingir os objetivos específicos do projeto.
- Os principais problemas enfrentados estão relacionados à ausência de implementação para receber e processar as mensagens MQTT e executar ações com base nelas. Para resolver esse problema, é necessário adicionar a lógica adequada dentro do loop de comunicação MQTT para lidar com as mensagens recebidas.
- Vantagens:
 - O código fornece uma base para se conectar a uma rede Wi-Fi, estabelecer uma conexão MQTT e exibir mensagens em uma tela TFT.
 - O uso do protocolo MQTT permite uma comunicação leve e assíncrona entre o dispositivo e o broker MQTT.
 - O projeto permite a comunicação e controle via internet usando o protocolo TCP/IP.
- Desvantagens:
 - O código fornecido é um exemplo básico e requer expansão e personalização para atender aos requisitos específicos do projeto.
 - A ausência de implementação para receber e processar as mensagens MQTT limita a funcionalidade do projeto.
 - A falta de detalhes sobre o hardware utilizado e a interface com outros dispositivos dificulta a avaliação completa das vantagens e desvantagens do projeto.
- Para melhorar o projeto, as seguintes ações podem ser consideradas:
 - Implementar a lógica para receber e processar as mensagens MQTT no tópico "somar" e executar ações com base nessas mensagens.
 - Adicionar detalhes sobre o hardware utilizado, como sensores, atuadores e interfaces com outros dispositivos.



- Aperfeiçoar a documentação do código, fornecendo mais comentários e explicações para facilitar a compreensão e modificação do projeto.
- Explorar outras funcionalidades e recursos do ESP32, como o uso de interrupções, timers e bibliotecas adicionais para expandir as capacidades do projeto.
- Considerar a implementação de mecanismos de segurança, como autenticação e criptografia, para proteger a comunicação MQTT.
- Realizar testes e depuração adequados para garantir a estabilidade e confiabilidade do projeto.

REPOSITÓRIO GITHUB: <https://github.com/imendesm/TFT>

YOUTUBE: https://youtu.be/_ARIRR3fXJk