

Manual do Usuário

**KLIF
PIRELLI**

Controle do Documento

Histórico de revisões

Data	Autor	Versão	Resumo da atividade
04/06/2023	Eduarda	1.1	Criação da seção 1 (completa)
05/06/2023	Marcos	1.2	Criação da seção 3, 4 e 6
05/06/2023	Eduarda	1.3	Criação da seção 5
09/06/2023	Luiz	1.4	Criação da seção 2
11/06/2023	Raab	1.5	Atualização da seção 6

Índice

1. Componentes e Recursos	3
1.1. Componentes externos	3
1.2. Requisitos de conectividade	4
2. Guia de Montagem	5
3. Guia de Instalação	7
4. Guia de Configuração	8
5. Guia de Operação	9
6. Troubleshooting	11

1. Componentes e Recursos

1.1. Componentes externos

A solução proposta para resolver o problema de perda e extravio de tablets/notebooks na fábrica da PIRELLI consiste no desenvolvimento de um sistema de rastreo utilizando dispositivos com tecnologia de geolocalização. Esses dispositivos serão integrados aos tablets/notebooks utilizados pela empresa. Os dados de localização serão transmitidos em tempo real para um dashboard na plataforma Ubidots, proporcionando à empresa uma visão clara e estimada da localização de seus dispositivos. Para implementar essa solução, serão necessários um tablet/notebook e os componentes adicionais específicos:

- **Esp32 Wroom com Antena - 38 pinos (2 pcs):** É um circuito integrado, responsável pela execução e alimentação do protótipo.
- **Jumper fêmea-fêmea - 20 cm (20 pcs):** Responsável por algumas conexões eletrônicas entre o Esp e os componentes.

- **Jumper macho-fêmea - 20 cm (20 pcs):** Responsável por algumas conexões eletrônicas entre o Esp e os componentes.
- **Jumper macho-macho - 20 cm (20 pcs):** Responsável por algumas conexões eletrônicas entre o Esp e os componentes.
- **Resistor 10k ohms (3 pcs):** Responsável por limitar a diferença de potencial elétrico em partes específicas do circuito (Ex: Leds).
- **Resistor 1k ohms (3 pcs):** Responsável por limitar a diferença de potencial elétrico em partes específicas do circuito (EX: Push Button).
- **LED - Vermelho (1 pcs):** Responsável pelo alerta visual definido do escopo do projeto.
- **LED - Amarelo (1 pcs):** Responsável pelo alerta visual definido do escopo do projeto.
- **LED - Verde (1 pcs):** Responsável pelo alerta visual definido do escopo do projeto.
- **Sensor Ultrassônico HC-SR04 (2 pcs):** Responsável por medir distâncias de 2cm a 4cm entre o dispositivo e o equipamento monitorado.
- **Display LCD 16x2 (1 pcs):** Responsável pelo alerta visual escrito, definido do escopo do projeto.
- **Kit RFID Mfrc522 (1 pcs):** Responsável pela identificação e controle de acesso.

- **Sensor *BME280* (1 pcs):** Responsável pela medição de temperatura e umidade dos dispositivos *host*.
- **Cabo Micro USB (2 pcs):** Responsável pela fonte de alimentação elétrica para o Esp.

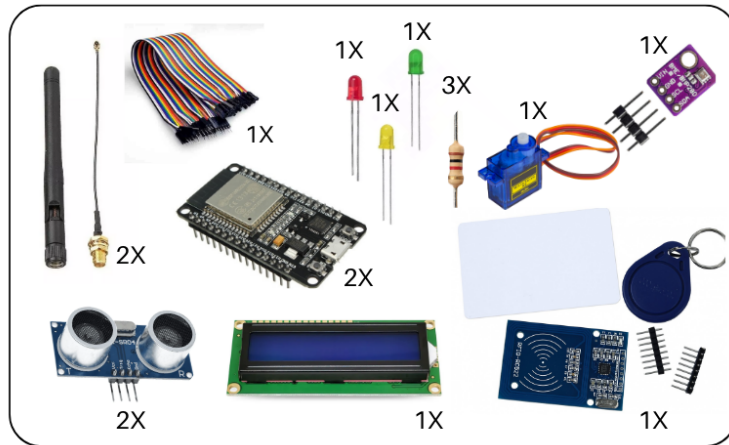


Imagem 1: Componentes utilizados e suas respectivas quantidades.

1.2. Requisitos de conectividade

Para o correto funcionamento do aplicativo, é indispensável a presença de uma conexão Wi-Fi. Essa conexão é estabelecida utilizando as bibliotecas <WiFi.h> e <WiFiClient.h>. A disponibilidade de acesso à Internet é crucial para viabilizar a comunicação entre os dispositivos e a plataforma Ubidots. É necessário que os dispositivos estejam conectados à Internet para enviar e receber dados por meio do *dashboard*. Adicionalmente, a biblioteca "UbidotsEsp32Mqtt.h" é

utilizada para implementar o protocolo MQTT, permitindo a conexão com o Ubidots.

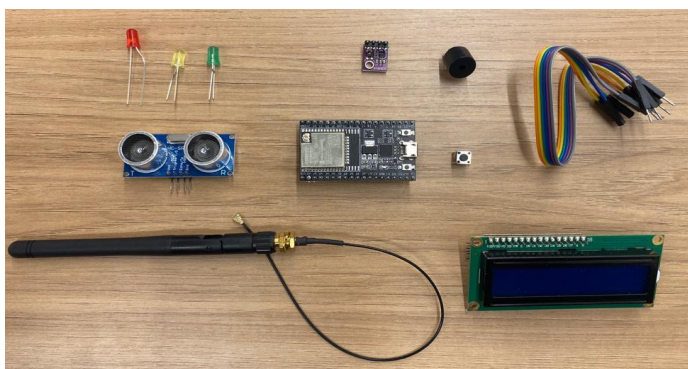
A biblioteca "UbidotsEsp32Mqtt.h" é especialmente desenvolvida para o microcontrolador ESP32 e possibilita a comunicação com a plataforma Ubidots usando o protocolo MQTT (*Message Queuing Telemetry Transport*). O Ubidots é uma plataforma de IoT (Internet das Coisas) que permite coletar, armazenar, visualizar e analisar dados de sensores e dispositivos conectados. Com a biblioteca "UbidotsEsp32Mqtt.h", é possível enviar dados de sensores ou variáveis do seu microcontrolador ESP32 para a plataforma Ubidots usando o protocolo MQTT, assim como receber comandos ou notificações da plataforma.

2. Guia de Montagem

Para facilitar a montagem do nosso equipamento, fornecemos dois guias: o Guia de Montagem do ESP *Host* e o Guia de Montagem do ESP *Client*.

HOST

No guia de montagem do ESP Host, utilizamos os seguintes itens, conforme mostrado na imagem 2, e cada um tem uma função específica:



Jumpers: São utilizados para fazer a conexão entre os itens e o ESP Host.

LEDs: Os LEDs são conectados através de jumpers às saídas quinze (LED vermelho - Desconectado), zero (LED verde - Conectado) e dois (LED amarelo - Aguardando conexão), além de uma saída GND.

Cada LED possui um resistor de 300 ohms para limitar a corrente e evitar queimaduras.

Sensor ultrassônico: A conexão com o ESP Host é feita através de jumpers. O pino VCC é conectado à entrada de energia, o pino trig é conectado à porta 26 e o pino Echo à porta 27. Além disso, o pino GND é conectado para realizar o aterramento, assim como nos LEDs.

Buzzer: É conectado à porta 25 e ao GND para aterramento.

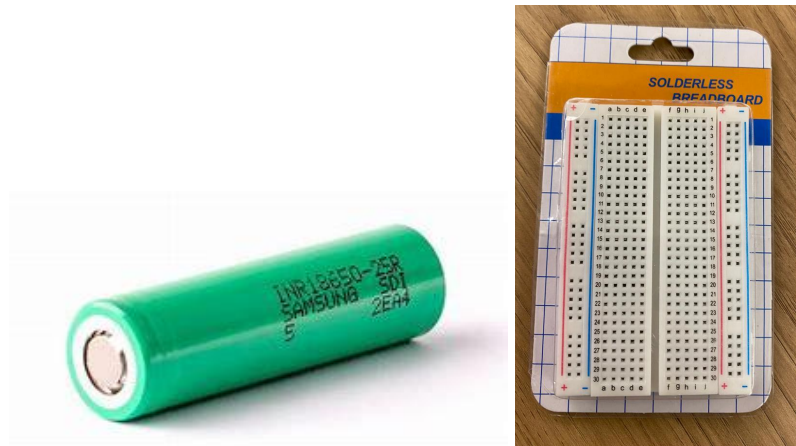
PushButton: É conectado à porta 32 e ao GND, juntamente com um resistor de 10k ohms.

Sensor BME280: A conexão com o ESP Host é feita através dos pinos SCL (porta 22), SDA (porta 21), VIN (entrada de energia) e GND (aterramento).

LCD: A conexão do LCD com o ESP Host é feita através dos pinos SCL (porta 22), SDA (porta 21), VCC (entrada de energia) e GND (aterramento).

Antena WiFi: É conectada ao ESP Host para ampliar a potência do sinal na comunicação entre os dispositivos.

Além disso, é utilizada uma protoboard para fazer a conexão entre todos os itens e uma bateria de 3.3V para alimentar cada dispositivo (*Host* e *Client*).



Observação: É importante lembrar que todas essas portas podem ser alteradas no código-fonte, caso haja necessidade, com exceção das portas para o dispositivo LCD e BME280, que se comunicam via protocolo I2C e precisam ser exclusivamente nas portas 21 e 22.

CLIENT

Jumpers: São utilizados para fazer a conexão entre os itens e o ESP Host.

LEDs: Os LEDs são conectados através de jumpers às saídas quinze (LED vermelho - Desconectado), zero (LED verde - Conectado) e dois (LED amarelo - Aguardando conexão), além de uma saída GND. Cada LED possui um resistor de 300 ohms para limitar a corrente e evitar queimaduras.

Sensor ultrassônico: A conexão com o ESP Host é feita através de jumpers. O pino VCC é conectado à entrada de energia, o pino trig é conectado à porta 26 e o pino Echo à porta 27. Além disso, o pino GND é conectado para realizar o aterramento, assim como nos LEDs.

Antena WiFi: É conectada ao ESP Host para ampliar a potência do sinal na comunicação entre os dispositivos.

Além disso, é utilizada uma protoboard para fazer a conexão entre todos os itens e uma bateria de 3.3V para alimentar cada dispositivo (*Host e Client*).



acoplado ao tablet.

Imagem 3: Ilustração do dispositivo

3. Guia de Instalação

Passo 1 - Instalação do *Host* de Comunicação:

- Instale o *Host* responsável pela comunicação com os dispositivos ESP32 na área que deseja monitorar.
- Ligue o *Host* e aguarde a conexão com a rede Wi-Fi da PIRELLI.

Passo 2 - Instalação dos Dispositivos ESP32 (*Client*):

- Após o *Host* se conectar ao Wi-Fi da PIRELLI, ligue os dispositivos ESP32 Client.
- Os dispositivos ESP32 Client procurarão pelo servidor criado pelo *Host* e estabelecerão conexão com ele.
- Percorra a área da fábrica e registre a localização de cada roteador Wi-Fi desejado.
- Crie pontos de referência com base nessas localizações.

Passo 3 - Configuração Concluída:

- Completando os passos anteriores, a instalação estará pronta.
- Através da interface web Ubidots, o responsável pelo monitoramento terá acesso à localização do dispositivo móvel.

- O sinal Wi-Fi e os pontos de referência configurados serão utilizados para monitorar e rastrear os dispositivos móveis.

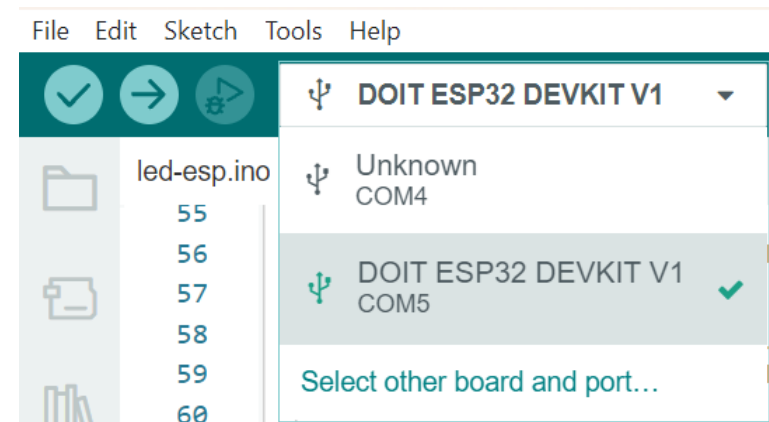


Imagem 4: Placa (DOIT ESP32 DEVKIT V1) e porta (COM5) utilizadas.

4. Guia de Configuração

Para realizar a configuração, siga os seguintes passos:

1. Conecte o protótipo ao tablet/notebook utilizando a entrada USB.
2. Abra o aplicativo "Arduino IDE" em seu tablet/notebook.
3. Verifique a porta correta para a conexão do protótipo.
4. Ligue o aparelho à conexão do tablet/notebook.
5. Acesse o seguinte link em seu navegador:
<https://profgodoiswk.iot.ubidots.com/app/dashboards/646babe6e15600ead1436fb6>.
6. Neste link, você poderá visualizar o *dashboard* que será conectado diretamente ao código.

A screenshot of a code editor interface. At the top, there are two tabs: 'Code' and 'Blame', with 'Code' being the active tab. To the right of the tabs is a toolbar with icons for 'Raw', copy, download, edit, a dropdown menu, and a code icon. The main area of the editor displays a list of Arduino library includes, each preceded by a line number from 1 to 7. The code is as follows:

```
1  #include <WiFi.h>
2  #include <WiFiClient.h>
3  #include <Wire.h>
4  #include <LiquidCrystal_I2C.h>
5  #include "UbidotsEsp32Mqtt.h"
6
7  //-----/
```

Imagem 5: Código de todas as bibliotecas utilizadas.

5. Guia de Operação

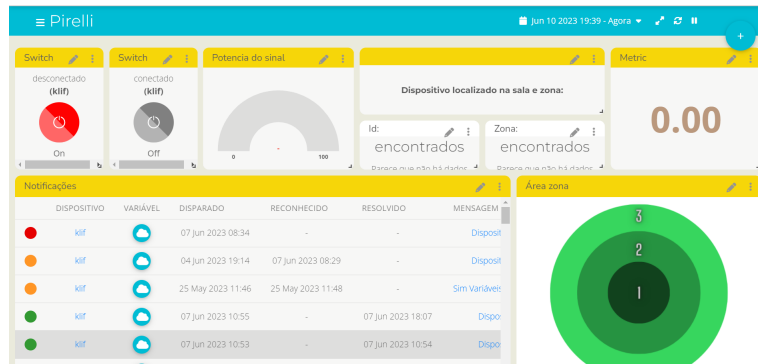


Imagem 6: Página inicial do *dashboard* no Ubidots.

O projeto utiliza a plataforma Ubidots, uma solução baseada em nuvem para Internet das Coisas (IoT), que permite coletar, armazenar, processar e visualizar dados de dispositivos conectados. Essa plataforma abrangente simplifica o desenvolvimento de aplicativos IoT, oferecendo recursos para coleta de dados de sensores, controle remoto de dispositivos e criação de painéis personalizados. O Ubidots é flexível e escalável, suportando diversos dispositivos e protocolos de comunicação. O acesso à plataforma é feito por meio do link fornecido anteriormente.

Ao acessar a plataforma Ubidots, o usuário terá acesso a um painel de controle, conforme mostrado na imagem 5. Nesse painel, o usuário poderá verificar várias informações relevantes. Por exemplo:

Conexão do dispositivo: É possível verificar se o dispositivo está conectado à rede da empresa.

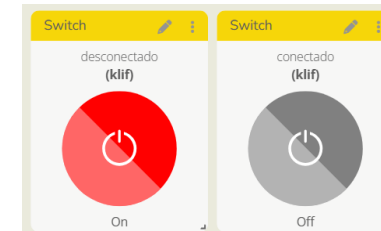


Imagem 7: Exemplo de dispositivo desconectado.

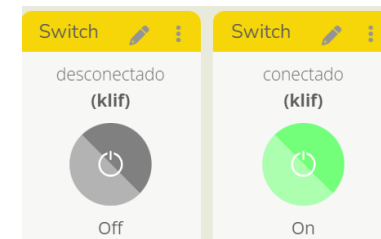


Imagem 8: Exemplo de dispositivo conectado.

Localização do dispositivo: É exibida a sala e a zona onde o dispositivo está localizado.

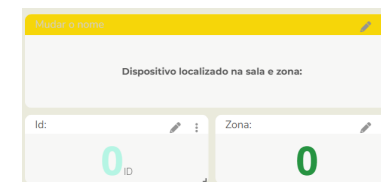


Imagem 9: Exemplo de dispositivos localizados em sala e zona.

Potência do sinal: A potência do sinal é exibida para determinar a localização mais próxima do dispositivo.



Imagem 10: Exemplo de exibição da potência do sinal.

Histórico de desconexões: É possível visualizar a quantidade de vezes que o dispositivo se desconectou da rede em um intervalo pré-definido pelo usuário (por exemplo, uma semana).

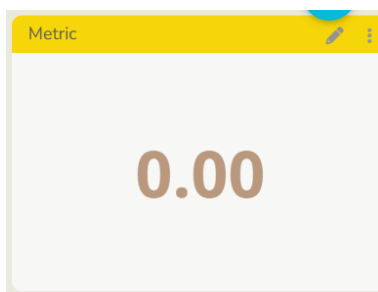


Imagem 11: Exemplo de histórico de desconexões.

Além disso, na aba "Dados" do painel de controle, o usuário tem a opção de programar eventos. Quando esses eventos ocorrem, uma notificação pré-programada é enviada por meio das ações disponíveis, como mostrado na imagem 6. Essas ações permitem ao

usuário definir notificações personalizadas para serem acionadas em situações específicas, proporcionando maior controle e gerenciamento dos dispositivos conectados.

6. Troubleshooting

Aqui estão algumas situações de falha comuns relacionadas ao dispositivo KLIF, que usa MQTT para enviar dados para a plataforma Ubidots na nuvem. Ações recomendadas para solucionar esses problemas estão incluídas. Observe que este documento não abrange todos os componentes, mas se concentra nos sintomas e métodos para identificar a possível origem dos problemas relacionados ao uso do dispositivo. Esses problemas podem ser devido a problemas de software ou configuração do dispositivo.

#	Problema	Possível solução
1	Perda de conexão do <i>Host</i> com o WiFi	Caso o próprio software não consiga restabelecer a conexão, recomenda-se que reinicie o dispositivo através do botão de reset.
2	Perda de conexão do <i>Client</i> com o WiFi	Caso o próprio software não consiga restabelecer a conexão, recomenda-se que reinicie o dispositivo através do botão de reset.
3	Falha no envio de dados	Verifique se as credenciais de

	para a plataforma Ubidots	autenticação da plataforma estão corretas e se o dispositivo está conectado à internet. Caso necessário, verifique as configurações de rede do dispositivo.
4	Falha na coleta de dados	Verifique se os sensores estão corretamente conectados e configurados.
5	Erro de autenticação MQTT	Verifique se as credenciais de autenticação MQTT estão corretas e se o dispositivo está configurado corretamente para se comunicar com o servidor MQTT. Caso necessário, atualize as configurações de autenticação.