

Redes Eléctricas Inteligentes

Aplicações práticas de IoT em redes inteligentes usando **NODE-RED** e **ESP32**

6 de Junho de 2024*

Resumo

O desempenho das redes eléctricas inteligentes está fortemente dependente da conectividade e da capacidade de monitorização de um elevado número de parâmetros muitos deles distribuídos ao longo de vastas áreas geográficas. Ter a capacidade de obter informação, em tempo-real, acerca dos parâmetros técnicos e económicos da rede permite avaliar rapidamente o desempenho da rede, otimizar a distribuição de energia e detetar pontos de falhas. Neste quadro de referência, a Internet-das-Coisas (IoT) refere-se a um conjunto de tecnologias que tem vindo a ser incluída nas redes eléctricas inteligentes de modo a promover a aquisição da informação proveniente de diferentes fontes como é o caso de sensores instalados ao longo da rede. Assim, nesta parte da matéria, uma rede de nós de medida IoT serão desenvolvidos a fim de monitorizar diferentes grandezas físicas numa rede inteligente fictícia. Com base no protocolo MQTT, utilizando o ESP32 e a plataforma Node-Red, será possível ilustrar, de forma prática, de que forma uma rede IoT opera.

Instalação de software

Para a realização deste trabalho, o aluno deverá trazer para a aula o seu computador pessoal com o software NODE-RED instalado. As instruções para a instalação deste software podem ser encontradas em:

<https://nodered.org/docs/getting-started/local>

Para além disso, e caso ainda não tenham, devem instalar o Arduino IDE:

<https://www.arduino.cc/en/software>

assim como o *add-on* necessário para programação da plataforma ESP32 com Arduino IDE. Relativamente a este último, copie e cole o endereço https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json no campo “URL Adicionais do Gestor de Placas” na janela

*Criado por João Paulo Coelho (jpcoelho@ipb.pt) e Ângela Ferreira (aferreira@ipb.pt)

de “Preferências” disponível a partir do menu “Ficheiro”. A seguir, no “gestor de placas”, existente no menu “Ferramentas”, faça a pesquisa por “ESP32” e instale o módulo. Mais detalhes sobre este processo em <https://randomnerdtutorials.com/installing-esp32-arduino-ide-2-0/>.

Os blocos de código que serão utilizados no decorrer deste módulo encontram-se disponíveis no repositório GIT da disciplina em:

<https://tinyurl.com/2syasrnj>

Introdução

Esta última parte da matéria assenta na **digitalização** de redes elétricas. Neste momento, é claro para todos o que é uma rede elétrica, as suas características, técnicas assim como os desafios que enfrenta no panorama atual onde as condições de operação são dinâmicas e envolvem a necessidade de se considerar vertentes societárias, políticas e ambientais. No entanto, é provável que o termo **digitalização** não seja tão claro e, por isso, deve ser devidamente definido e contextualizado.

Quando se fala em digitalização de um processo, seja ele de que natureza for, entende-se a conversão da informação que, historicamente era realizada utilizando um suporte diferente, num formato **digital**. Inicialmente, a intenção era de tornar a informação de mais fácil acesso para a tomada de decisão. No entanto, atualmente a digitalização envolve também o desenvolvimento de algoritmos para processamento dos dados digitalizados. Algoritmos de aprendizagem máquina permitem antecipar cenários com base na informação digitalizada, tanto presente como passada, e aprender ou adaptar-se a novas condições. Os recentes avanços da inteligência artificial abre ainda mais o leque das possibilidades que podem ser alcançadas através da digitalização.

No caso das redes elétricas inteligentes, a adaptação a condições dinâmicas de demanda e operação está fortemente dependente da aquisição e processamento de dados em tempo real. Alguns desses dados, como a quantidade de energia elétrica a ser gerada, a sua proveniência, padrões de consumo de energia e cargas assim como o estado da rede são grandezas que, enquanto técnicos, facilmente percebemos a sua necessidade. No entanto, outras variáveis devem ser tidas em linha de conta como por exemplo o conhecimento das condições ambientais, económicas e regulamentares. Por exemplo, a temperatura, humidade, velocidade do vento, radiação solar e muitos outros fatores tem influência direta no desempenho da rede e nos prontos de produção de origem renovável. Para além disso, e visto que atualmente se atua num mundo globalizado, os preços da energia em diferentes mercados devem ser também considerados no comportamento da rede. Finalmente, é necessário entender que há mecanismos de regulação externa que existem para garantir a qualidade de serviço. Assim, é necessário que a rede tenha a capacidade de **adquirir os dados** necessários para reportar às entidades reguladoras. Neste quadro de referência, uma outra

tecnologia designada por “Internet-das-Coisas” (normalmente definida pela sigla IoT do Inglês “*Internet-of-Things*”) desempenha um papel fundamental na aquisição desses dados. Para além disso, um nó IoT permite realizar o processamento local da informação e atuação sobre diferentes variáveis e grandezas físicas.

A “Internet-das-Coisas” (IoT)

O IoT é uma visão tecnológica que assenta na interligação de diferentes sistemas digitais utilizando a maior rede de dados do mundo: a Internet. Por exemplo, ao nível doméstico, é possível adquirir medidores de energia elétrica e tomadas que se ligam à rede WiFi de casa e que permitem acesso remoto a partir de qualquer ponto do mundo. A arquitetura simplificada de um sistema IoT deste tipo apresenta-se na Figura 1.



Figura 1: Arquitetura de um sistema IoT doméstico capaz de realizar a medição da energia elétrica e controlar o estado de uma tomada.

Com frequência, um sistema de controlo e aquisição de dados envolve uma rede de dispositivos interligados entre si utilizando uma rede de dados que assenta num protocolo de comunicação diferente. Nesse caso, existem várias topologias distintas. Uma das mais comuns é a ligação em estrela onde cada nó comunica com um dispositivo central que, posteriormente, encaminha os pacotes via TCP/IP para um servidor onde os dados são agregados e processados. Existem também outras alternativas como é redes “*mesh*” onde a informação é roteada de um ponto para outro passando por diferentes nós. No contexto das redes inteligentes, um dos elementos fundamentais para avaliar os perfis de consumo são os contadores inteligentes que, há já cerca de uma década, têm vindo a substituir os contadores de energia eletromecânicos. Nesse caso, a rede

possui uma tipologia em estrela onde um conjunto de unidades comunica com uma unidade central local utilizando um protocolo de comunicação específico. Em zonas urbanas, o PLC (*Power Line Carrier*) é um dos protocolos mais utilizados e utiliza a própria rede elétrica como infraestrutura física de comunicação. Em zonas rurais, muitas vezes, por questões económicas, opta-se pelo GPRS¹. A Figura 2 mostra o exemplo de uma arquitetura associada ao sistema de medição da energia elétrica por telecontagem.

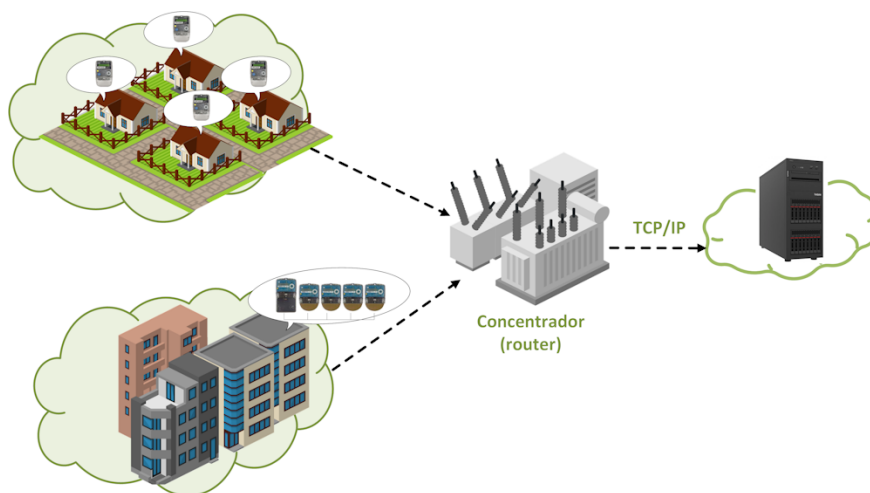


Figura 2: Arquitetura de um sistema de comunicação de telecontagem (PLC).

Rede de sensores IoT

Nesta parte do trabalho, deve imaginar o cenário retratado na Figura 3 onde se apresenta uma rede elétrica simplificada. O objetivo é criar um sistema de monitorização e controlo desta rede. Para isso, admite-se um conjunto de dispositivos IoT distribuídos ao longo da rede que partilham informação com uma estação central.

Considere que os nós IoT têm por base a plataforma de desenvolvimento ESP32-WEMOS cujas opções de conectividade permitem a sua ligação a uma rede WiFi. Associados a esses nós existirá um conjunto de sensores que irão permitir a medição da temperatura, radiação luminosa e corrente elétrica. Existirá ainda a possibilidade de alterar o estado de relés que irão emular o controlo remoto de seccionadores. A informação dos sensores será encaminhada para uma plataforma remota onde, através de um painel de instrumentos (*dashboard*), será possível aceder aos dados provenientes dos sensores e ativar os atuadores.

¹Neste documento https://www.e-redes.pt/sites/eredes/files/2019-04/Novos_Equipamentos.pdf a *e-redes* descreve os aparelhos utilizados para telecontagem.

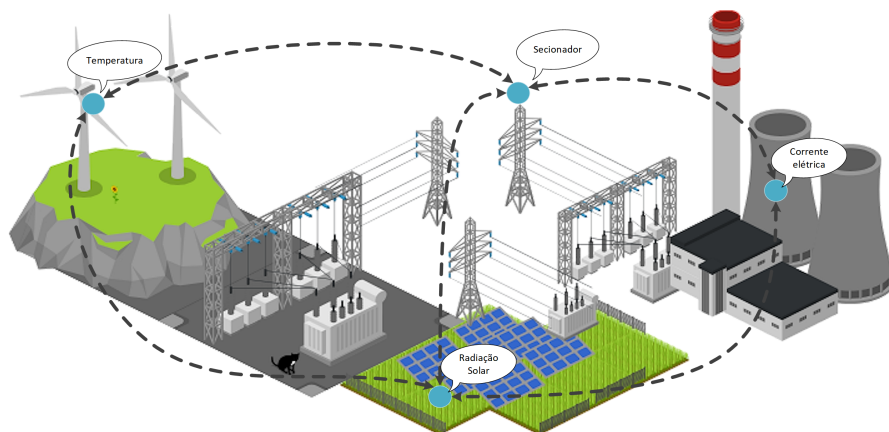


Figura 3: Uma rede elétrica hipotética à qual se pretende adicionar uma rede de dispositivos IoT.

Parte 1: telemetria com MQTT

Nesta parte do trabalho, será desenvolvida a componente de comunicação de dados entre os nós IoT e a plataforma de tratamento de dados. A partilha de dados entre eles será realizada através da utilização de um protocolo de comunicação muito utilizado em redes de comunicação IoT designado por **MQTT**.

A sigla MQTT deriva do Inglês Message Queuing Telemetry Transport e trata-se de um protocolo para troca de mensagens que foi desenvolvido para ser utilizado em ambientes e dispositivos² onde existem restrições em termos de largura de banda, consumo de energia, etc.

Este protocolo funciona segundo o modelo **cliente-servidor** onde os dispositivos (clientes) comunicam através de um intermediário central (servidor). No caso do protocolo MQTT, o servidor é designado por **broker**.

Nesta arquitetura, os clientes subscrevem e podem publicar mensagens sobre **tópicos** específicos. Quando um cliente publica uma mensagem num tópico, o *broker* encaminha a mensagem para todos os clientes que subscreveram esse tópico.

Existem muitos detalhes acerca deste protocolo de entre os quais se salientam a capacidade em permitir três níveis de garantia de entrega de mensagens e a possibilidade de encriptação recorrendo a TLS/SSL. No entanto, neste trabalho, por simplicidade, ignoram-se esses detalhes.

Procedimento Experimental

1º Passo: Aceda ao GitHub e descarregue o *sketch* que se encontra no diretório /ESP32/Parte1 com o nome PARTE_1_OLA.ino. Abra esse ficheiro no

²Designados por (*constrained devices*).

Arduino IDE e transfira o programa para o ESP32.

Nota A compilação deste *sketch* requer que a biblioteca `PubSubClient` esteja instalada. Se necessário, para instalar siga os passos ilustrados na Figura 9.

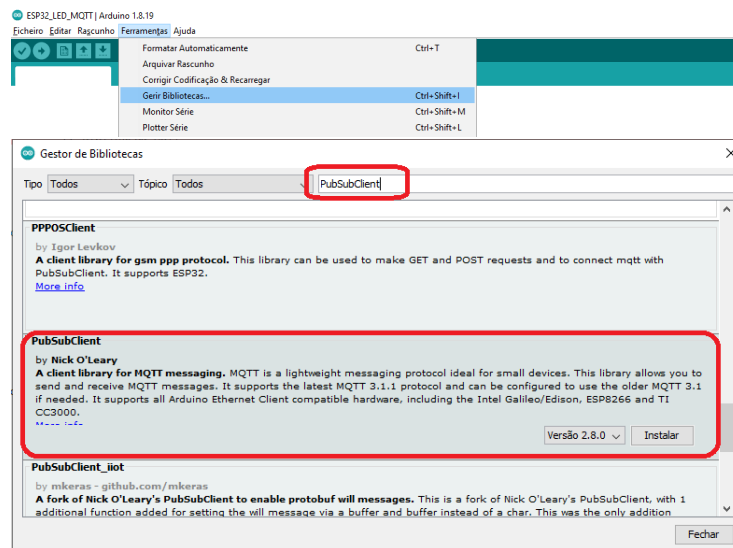


Figura 4: Procedimento para instalação da biblioteca `PubSubClient`.

Execute o código, abra o *Serial Monitor* configurado com *baudrate* igual a 115200 bps e, como se mostra na Figura 5, tome nota do **ID** do nó. Essa informação será necessária adiante.

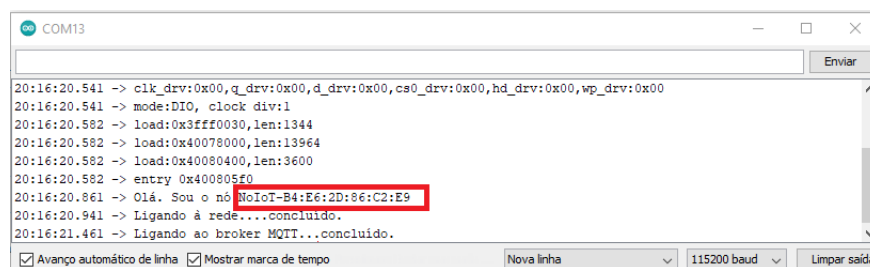
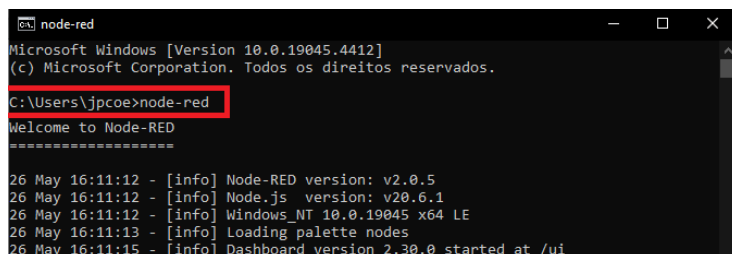


Figura 5: Identificação do nó IoT.

2º Passo Abra a linha de comandos do Windows® e escreva:

```
> node-red
```

O servidor deverá arrancar apresentando diversas mensagens de execução como se mostra na Figura 6.

A terminal window titled 'node-red' showing the startup process. The command 'node-red' has been executed at the prompt 'C:\Users\jpcor>'. The output includes a welcome message and several informational logs: 'Node-RED version: v2.0.5', 'Node.js version: v20.6.1', 'Windows_NT 10.0.19045 x64 LE', 'Loading palette nodes', and 'Dashboard version 2.30.0 started at /ui'.

```
node-red
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. Todos os direitos reservados.
C:\Users\jpcor>node-red
Welcome to Node-RED
=====
26 May 16:11:12 - [info] Node-RED version: v2.0.5
26 May 16:11:12 - [info] Node.js version: v20.6.1
26 May 16:11:12 - [info] Windows_NT 10.0.19045 x64 LE
26 May 16:11:13 - [info] Loading palette nodes
26 May 16:11:15 - [info] Dashboard version 2.30.0 started at /ui
```

Figura 6: Execução do Node-Red através da linha de comandos.

3º Passo: Abra o *browser* (Chrome, Edge ou equivalente) e escreva na barra de enderços o URL:

> localhost:1880

O *browser* deverá abrir uma pagina semelhante à que se apresenta na Figura 7.

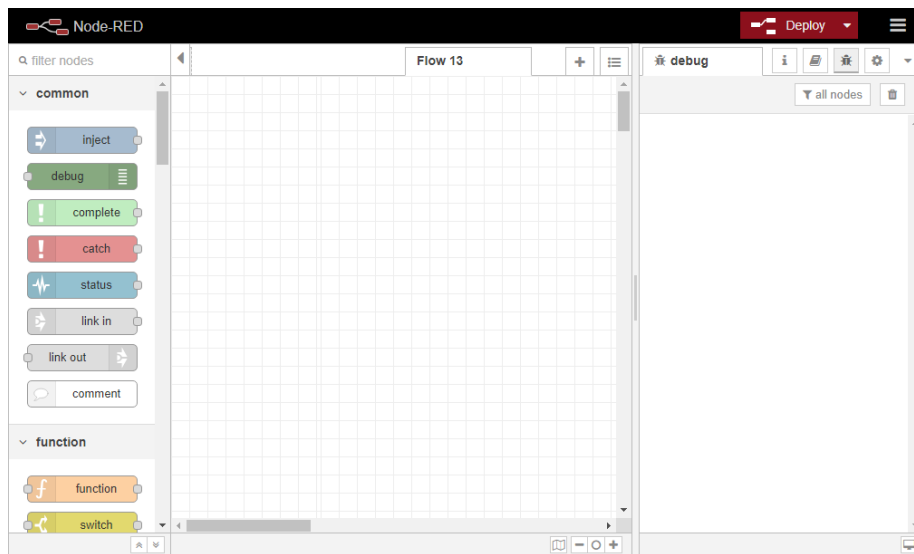


Figura 7: Aspeto gráfico da interface Node-Red.

4º Passo No ambiente de trabalho do Node-Red (*flow*) comece por adicionar, através de procedimento *drag-and-drop*, os nós MQTT IN e MQTT OUT.



Clique duas vezes no nó MQTT IN para configurar os detalhes do *broker* cujos campos se mostram na Figura 8.

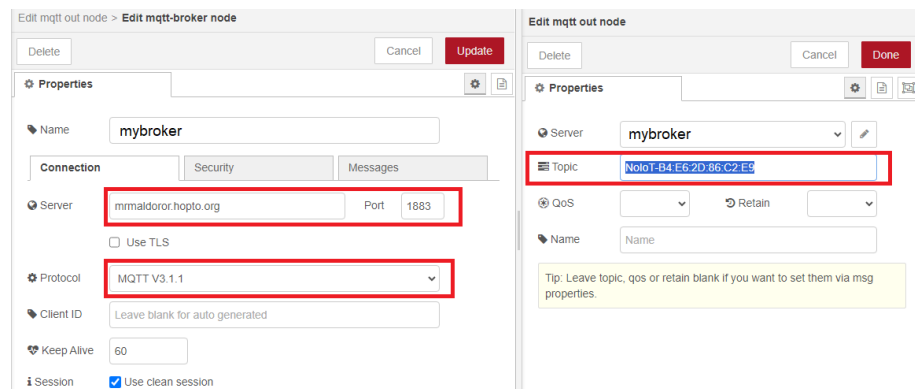


Figura 8: Configuração dos blocos de MQTT IN/OUT.

Em seguida, desenhe e configure o diagrama de blocos que se mostra na Figura 9. Depois disso, execute o **deploy** e confirme o seu funcionamento.

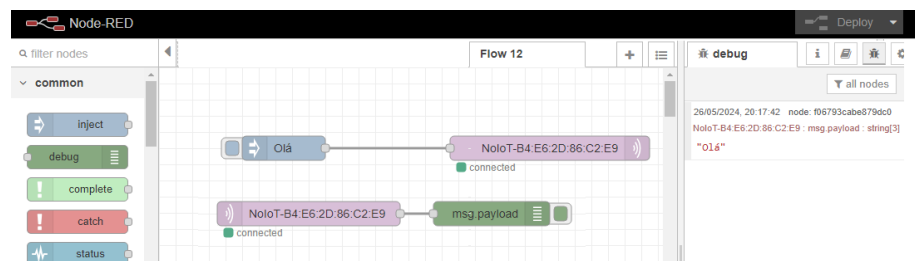


Figura 9: *Flow* para Node-Red (Parte 1).

5º Passo: Depois de comprovar o envio e receção de mensagens entre o ESP32 e a plataforma Node-Red, aceda novamente ao **GitHub** e descarregue o *sketch* que se encontra no diretório `/ESP32/Parte1` com o nome `PARTE_1_LED.ino`. Abra esse ficheiro no Arduino IDE e transfira o programa para o ESP32.

6º Passo: Altere o *flow* no Node-Red como se mostra na Figura 10. Confirme

que consegue controlar o estado do LED embutido na placa ESP32 em função da mensagem publicada no tópico.

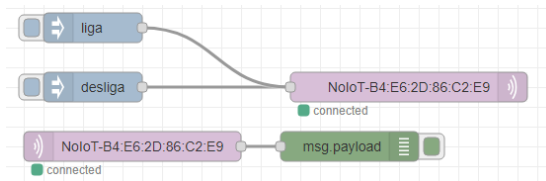


Figura 10: *Flow* para controlo do LED do ESP32 (Parte 1).

Parte 2: *dashboard* com NODE-RED

Nesta segunda parte, o objetivo é criar uma interface de utilizador usando o Node-Red. Comece por aceder ao menu (ao lado do botão “deploy”) e seleccionar a opção “Manage palette”. Na janela que será aberta, selecione a aba “Install” e procure o módulo “node-red-dashboard”.

Depois de instalar o *add-on*, execute os seguintes passos:

- 1º Passo** Criar o diagrama de blocos da Figura 11. No interior do bloco “*Function*” deverá colocar o código (JavaScript) que se apresenta na Figura 12.

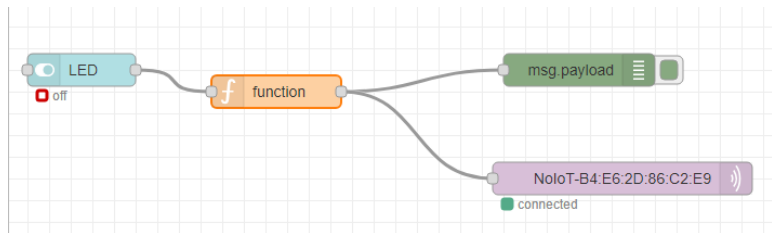


Figura 11: *Flow* do *dashboard* para controlo do LED (Parte 2).

- 2º Passo** Parametrize a janela do painel de comando como se mostra na Figura 13.

- 3º Passo** Agora, adicionam-se mais dois elementos gráficos ao painel. Neste caso, um objeto do tipo “*Chart*” e outro objeto do tipo “*Gauge*”. O novo diagrama de blocos que deve implementar tem o aspeto apresentado na Figura 14.

Neste caso, um gerador de números aleatórios vai debitando números inteiros entre 1 e 10 com uma cadência definida pelo bloco “*Inject*” que deve estar parametrizado para enviar uma mensagem automaticamente a cada segundo.

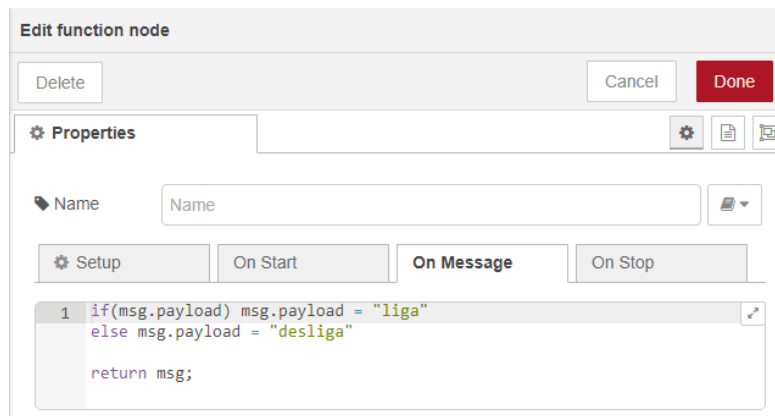


Figura 12: Conteúdo do bloco “Function” (Parte 2).

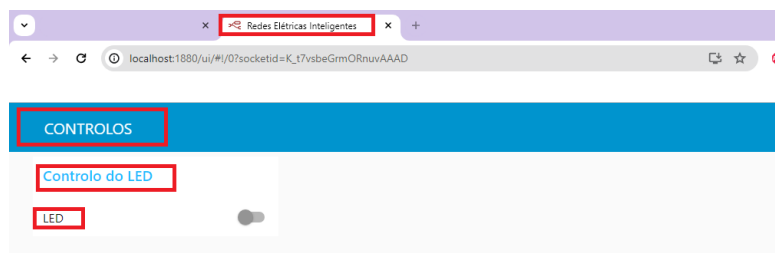


Figura 13: *Front-end* do painel de comando (Parte 2).

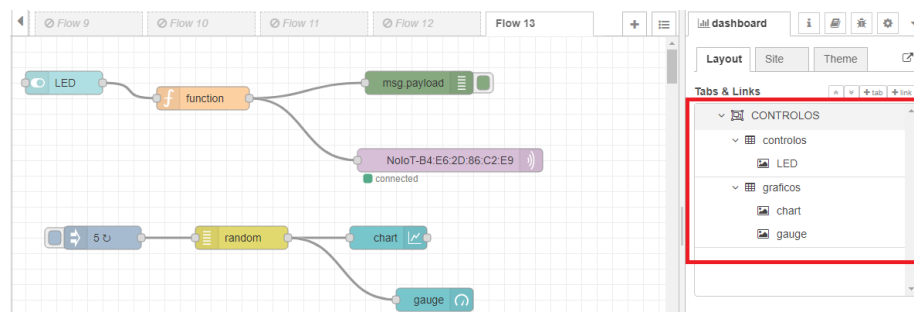


Figura 14: Diagrama de blocos para criação do *dashboard* onde se inclui um “Switch”, um “Chart” e um “Gauge” (Parte 2).

4º Passo Acesse ao GitHub e descarregue o programa `Parte2_RND.ino` e transfira-o para o ESP32. Este programa permite, não só controlar o LED, mas também envia periodicamente, a cada segundo, um número aleatório. Cons-



Figura 15: *Front-end* do painel de comando com os objetos “chart” e “gauge” (Parte 2).

trua um *dashboard* que permita controlar o estado do LED e apresente os valores enviados pelo microcontrolador.

Parte 3: aquisição de dados com ESP32

Nesta terceira parte, um sensor/atuador será conetado ao ESP32. Cada grupo/elemento irá selecionar um dispositivo de entre as seguintes possibilidades:

1. LDR para medição da radiação solar junto aos painéis solares da Figura 3;
2. Um NTC para medição da temperatura junto aos aerogeradores;
3. Um sensor de efeito *Hall* para medir a corrente elétrica junto à central termoelétrica;
4. Um sensor resistivo para medir a corrente elétrica junto ao ponto de consumo;

Para além disso, e em função da disponibilidade de material, será distribuído um **relé** a cada grupo.

Com base no ficheiro `Parte2_RND.ino` disponível no GitHub, devem efetuar as alterações necessárias de modo a poderem publicar a informação do vosso sensor e permitirem a atuação sobre o relé.

Devem também implementar um painel que permita a apresentação dos dados publicados assim como atuar sobre o relé.

Parte 4: Juntando tudo...

Nesta última parte, o *dashboard* deverá ser capaz de apresentar, não só a informação do vosso nó, mas a de todos os nós IoT da rede inteligente (os nós dos vossos colegas). Devem organizar devidamente o *front-end* de modo a que a informação seja facilmente identificável.

Anexo

Medida da corrente elétrica

Para a medição da corrente elétrica, será utilizado um sensor HST023R cujo funcionamento assenta no efeito *Hall*. Trata-se de um sensor do tipo indutivo, “*split core*”, cuja tensão de saída é proporcional à corrente elétrica.

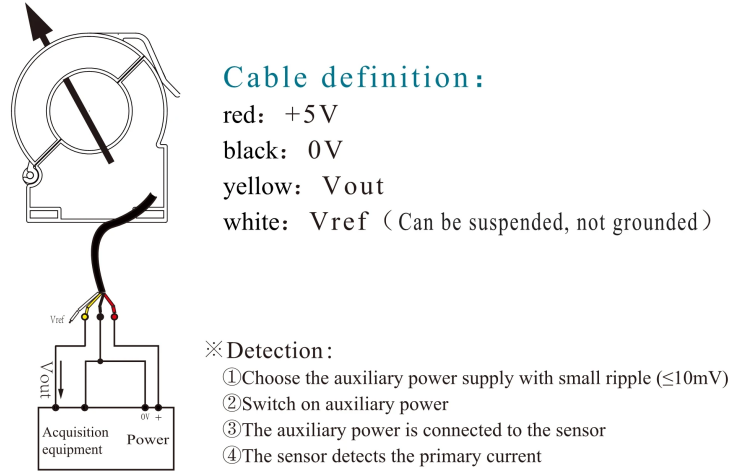


Figura 16: Ligação do sensor HST023R (retirado das folhas de dados do fabricante disponíveis em www.poweruc.pl).

Medida da temperatura

Para a medição da temperatura será utilizado um termistor NTC. A relação entre a resistência do termistor e a temperatura é dada por:

$$R(T) = R_0 \exp \left(\beta \left(\frac{1}{T} - \frac{1}{T_0} \right) \right)$$

onde T se refere à temperatura absoluta em kelvin, T_0 uma temperatura de referência (normalmente 298.15 K correspondendo a 25°C) e R_0 a resistência do termistor à temperatura de referência T_0 . O coeficiente β representa a sensibilidade do sensor à temperatura. Os parâmetros desta equação dependem do tipo de NTC mas, para aquele que integra a *breakout board* KY-013, $\beta = 3950$ e $R_0 = 10000 \Omega$. Na Figura 17 mostra-se o exemplo de uma dessas *breakout boards* assim como o esquema associado.

Com base na medição da entrada analógica, ADC , um inteiro entre 0 e $2^n - 1$ onde n se refere ao número de bits de resolução do conversor A/D, o valor da

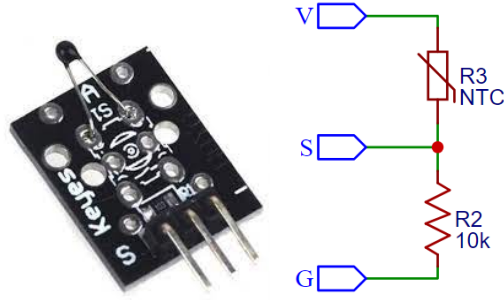


Figura 17: Ligação do LDR (sensor de iluminância).

iluminância em *lux* é obtida a partir das seguintes equações:

$$U_{SG} = \frac{A0}{2^n - 1} \times 5$$

$$R(T) = \frac{R_{pol}}{U_{SG}} 5 - R_{pol}$$

$$T = \frac{1}{\frac{1}{\beta} \ln \left(\frac{R(T)}{R_0} \right) + \frac{1}{T_0}}$$

Medida da intensidade de radiação solar (iluminância)

A medição da luminância, em lux, será realizada recorrendo a resistências sensíveis à luminosidade (LDR). Com base na medição da entrada analógica, *ADC*, um inteiro entre 0 e $2^n - 1$ onde n se refere ao número de bits de resolução do conversor A/D, o valor da iluminância em *lux* é obtida a partir das seguintes equações:

$$U_{SG} = \frac{A0}{2^n - 1} \times 5$$

$$R_{LDR}(\phi) = \frac{R_{pol}}{U_{SG}} 5 - R_{pol}$$

$$\phi = \left(\frac{R_{LDR}(\phi)}{R_0} \right)^{-\left(\frac{1}{\gamma}\right)}$$

Dependendo da *breakout board* a utilizar, considere os seguintes valores:

Na Figura 17 mostra-se o exemplo de uma dessas *breakout boards* assim como o esquema associado.

	γ	R_0	R_{pol}
Keystudio	0.6511	69700	47 000
HW486	0.5895	66862	10 000
KY018	0.7785	130000	10 000

Tabela 1: Parâmetros a utilizar para a medição da iluminância em função do *hardware*.



Figura 18: Ligação do LDR (sensor de iluminância).