

03

Preparando Arduino IDE

Transcrição

Mais uma característica interessante do NodeMCU é que vamos utilizar a mesma IDE do Arduino com qual já estamos acostumados, da mesma forma que já programamos o Arduino.

Preparando a IDE do Arduino

Antes de utilizar a IDE do Arduino, precisamos incluir nela as bibliotecas do módulos ESP8266. Para isso, no [GitHub do módulo](#) (<https://github.com/esp8266/Arduino>), nos é disponibilizado o seguinte link:

http://arduino.esp8266.com/stable/package_esp8266com_index.json .

Nós copiamos esse link, e na IDE do Arduino, acessando o primeiro item do menu superior e clicando em **Preferências** (ou **Preferences**), nós colamos esse link no campo **URLs Adicionais para Gerenciadores de Placas** (ou **Additional Boards Manager URLs**), e clicamos em **OK**.

Após isso, acessamos o item **Ferramentas** no menu superior, depois **Placa** e clicamos em **Gerenciador de Placas...** (ou **Tools -> Board -> Boards Manager...**). Na barra de pesquisa, pesquisamos por **esp8266**, clicamos na placa que aparecer e clicamos em **Instalar** (ou **Install**).

Terminando a instalação, ao selecionar a placa na IDE, temos as opções do módulo ESP8266. O NodeMCU, em específico, corresponde à placa **NodeMCU 1.0 (ESP-12E Module)**, então basta selecioná-la, e em seguida selecionar sua porta, onde o NodeMCU está conectado.

Piscando o LED do NodeMCU

Ao selecionar a placa, a IDE nos disponibiliza alguns exemplos **relativos a ela**, ou seja, os exemplos variam de acordo com a placa. Acessando o primeiro item do menu superior, e selecionando **Exemplos -> ESP8266 -> Blink** (ou **Examples -> ESP8266 -> Blink**), temos um dos exemplos disponibilizados, cuja funcionalidade é piscar o LED do NodeMCU.

Com o exemplo aberto, podemos fazer o upload para a placa, que demora um pouco mais que o Arduino, visto que o NodeMCU tem uma capacidade menor.

Realizado o upload, o LED do NodeMCU começará a piscar, funcionalidade do exemplo que estamos executando.

Executamos esse exemplo basicamente para testar a comunicação com o NodeMCU. Ao verificar que está tudo funcionando corretamente, podemos seguir com o nosso projeto.

