

# FEELT31109 - Enriquecimento Instrumental

## Roteiro Lab.02

Prof. Igor Peretta

16-Abril-2019

### 1 Introdução

O *Pulse Width Modulation* (PWM) – em português, modulação por largura de pulso – pode ser usado para controlar o valor da alimentação entregue à carga (veja mais em: <https://pt.wikipedia.org/wiki/PWM>). Com isso, através de controle dos períodos ligado (*duty cycle*) e desligado da onda quadrada, podemos “simular” valores intermediários entre o nível baixo e o alto de uma saída digital (ver Figura 1).

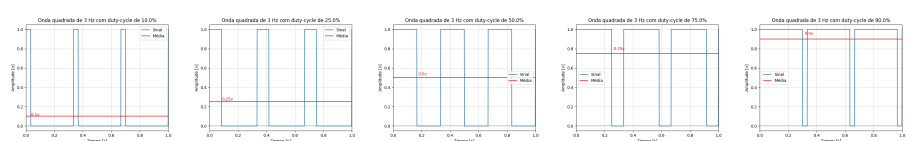


Figura 1: Exemplo de diferentes transferências de carga para diferentes *duty cycles*

No NodeMCU, as seguintes pinos, ou GPIOs (*General-purpose input/output*), possuem PWM por hardware: GPIO4 (**D2**), GPIO14 (**D5**), GPIO12 (**D6**) e GPIO15 (**D8**), veja na Figura 2. Usando `analogWrite()` podemos escrever até 1024 valores distintos (10 bits de conversor digital analógico), ou seja, de 0 (0v) a 1023 (3.3v). Assim, não precisamos nos preocupar com o *duty cycle* mas sim com o valor desejado de tensão no pino (GPIO).

### 2 Experimento 2A: Variação do brilho do LED

Neste projeto (adaptado da fonte: <https://www.filipeflop.com/universidade/kits-maker-arduino/projeto-3-brilho-oscilante/>), iremos fazer variar o brilho de um LED usando PWM em uma porta que suporta essa estratégia. A montagem se encontra na Figura 3.

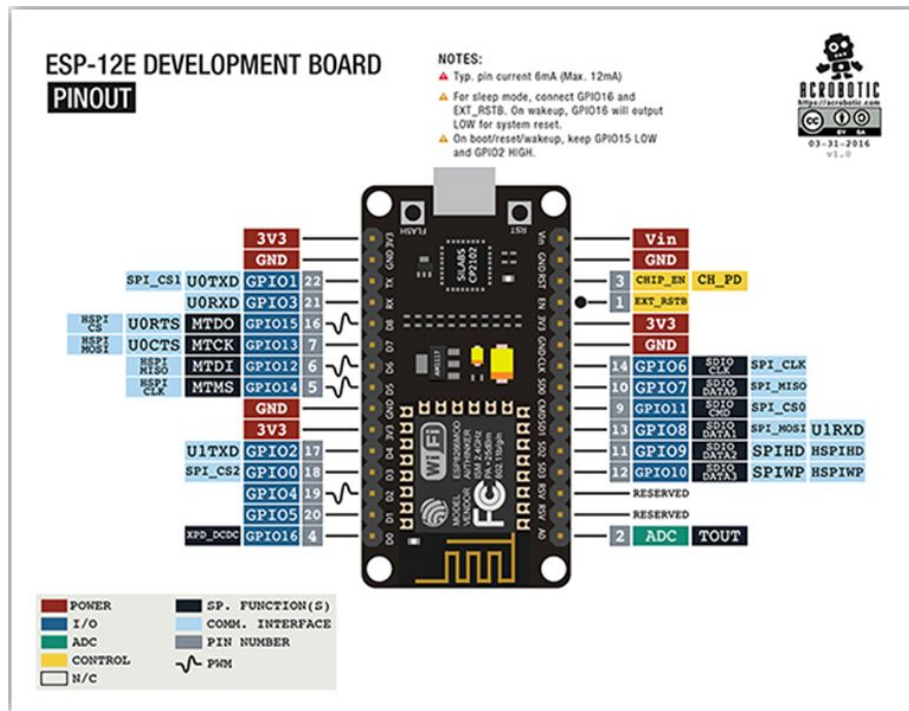


Figura 2: Esquema de pinagem da placa NodeMCU com ESP8266

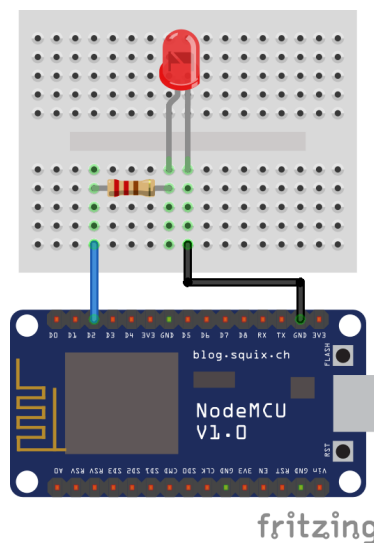


Figura 3: Esquemático para variação do brilho do LED

## 2.1 Material

- 1x LED Vermelho 5mm
- 1x Resistor 220  $\Omega$

Mais: *Protoboard*, *jumpers*, placa **NodeMCU ESP8266**

## 2.2 Código

Código-Fonte 1: Código-fonte para variação do brilho do LED

```
1 int pinoLed = D2;
2 int delta = 5;
3
4 void setup() {
5     pinMode(pinoLed, OUTPUT);
6 }
7
8 void loop(){
9     static byte value = 0;
10    // aumenta o brilho com la o "for"
11    for (value = 0; value < 255; value += delta){
12        // controla o brilho no pino do LED
13        analogWrite(pinoLed, value);
14        delay(30);
15    }
16    // diminui o brilho com la o "for"
17    for (value = 255; value > 0; value -= delta){
18        // controla o brilho no pino do LED
19        analogWrite(pinoLed, value);
20        delay(30);
21    }
22 }
```

## 3 Experimento 2B: Liga-desliga

Uma chave táctil (*push-button*) funciona como um contato que abre e fecha, sendo assim, uma chave possui dois valores, 0 ou 1, aberto ou fechado. Quando a chave é do tipo “normalmente aberto” (NA), significa que quando pressionada fecha-se o contato (ver Figura 4). Conectando uma chave a uma porta do NodeMCU podemos ler o valor 0 ou 1 da chave e assim tomar uma ação. Por só ter dois valores possíveis, utilizaremos uma das portas digitais da placa.

Geralmente, é útil direccionar um pino de entrada para um estado conhecido, se nenhuma outra entrada estiver presente. Isso pode ser feito adicionando um resistor de *pullup* (a 3,3 a 5V) ou um resistor de *pulldown* (resistor ao terra) na

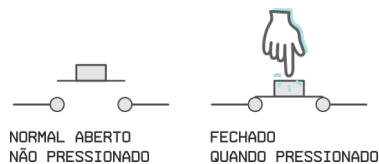


Figura 4: Chave tátil normalmente aberta (fonte: <https://uploads.filipeflop.com/2018/12/push-button.png>)

entrada. Um resistor de  $10k\Omega$  é um bom valor para um resistor de *pullup* ou *pulldown*.

Neste projeto (adaptado da fonte: <https://www.filipeflop.com/universidade/kit-maker-arduino/projeto-5-interruptor-de-luz/>), iremos fazer um controle liga-desliga de um LED. A montagem se encontra na Figura 5.

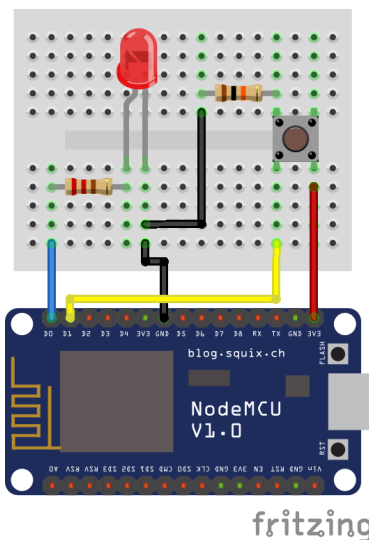


Figura 5: Esquemático para liga-desliga do LED

### 3.1 Material

- 1x LED Vermelho 5mm
- 1x Chave tátil (*push-button*) N/A
- 1x Resistor  $220\Omega$
- 1x Resistor  $10k\Omega$

Mais: *Protoboard*, *jumpers*, placa **NodeMCU ESP8266**

## 3.2 Código

Código-Fonte 2: Código-fonte para liga-desliga do LED

```
1 int botao = D1;
2 int led = D0;
3 bool estadoLed = 0;
4
5 void setup()
6 {
7     pinMode(botao , INPUT);
8     pinMode(led , OUTPUT);
9 }
10
11 void loop()
12 {
13     // se o botao for pressionado:
14     if(digitalRead(botao) == HIGH)
15     {
16         estadoLed = !estadoLed; // troca o estado do LED
17         digitalWrite(led , estadoLed);
18         // aguarda botao liberado:
19         while(digitalRead(botao) == HIGH);
20         delay(100);
21     }
22 }
```

## 4 (Opcional) Experimento 2C: Liga e desliga com variação de brilho

Com base nos códigos vistos nesse laboratório, monte um circuito com chaves tácteis liga/desliga para dois LEDs. Enquanto estiver ligado, um LED deve entrar em um ciclo de ir acendendo aos poucos, chegar no brilho máximo e voltar a apagar aos poucos, assim indefinidamente. Quando desligado, o LED não acende. O esquemático está na Figura 6.

### 4.1 Material

- 1x LED Vermelho 5mm
- 1x LED Verde 5mm
- 2x Chave táctil *push-button* NA
- 2x Resistores 220  $\Omega$
- 2x Resistores 10k  $\Omega$

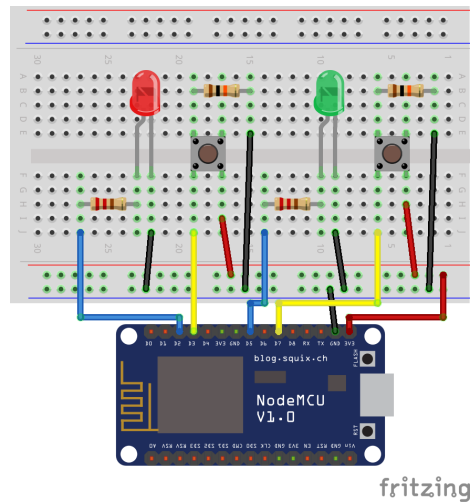


Figura 6: Esquemático do projeto opcional

## 5 Relatório

Redigir um relatório em  $\text{\LaTeX}$  (de 1 a 3 páginas) contendo, para cada projeto:

- Identificação da disciplina e do laboratório
- Identificação do(s) discente(s)
- Breve descrição do que era esperado
- Relato das montagens e das dificuldades encontradas
- Relato dos resultados obtidos
- (Opcional) Descrição das modificações feitas ao(s) projeto(s) e resultados obtidos a partir das mesmas

Publicar em PDF e enviar anexo para o e-mail: [iperetta@ufu.br](mailto:iperetta@ufu.br) com o assunto: **FEELT31109 2019-1 Relatório do Lab XX**, onde **XX** é o número identificador deste laboratório.

Dica: o site <http://www.overleaf.com/> é um dos sites em que se pode editar documentos  $\text{\LaTeX}$  online, além de permitir a colaboração de vários autores para o documento.