[Programa] Potenciómetro e LED

Objetivo:

Utilizar o ESP32-WROOM-32E DevKitC v4 [1] para ler valor de tensão de saída de um potenciómetro, para alimentar um LED e, fazendo uso do valor de lido no terminal de saída do potenciómetro, controlar o brilho do LED.

Lista de material:

- *ESP32-WROOM-32E DevKitC v4*
- Breadboards
- Resistência de 330 ohm (Ω)
- Díodo emissor de luz (LED) vermelho
- Potenciómetro de 10kΩ
- Fios de ligação
- Cabo USB micro

6_material_list.png

Contextualização:

Um potenciómetro é um componente eletrónico que incorpora uma resistência elétrica ajustável [4]. Por seu turno, um circuito LED é um circuito usado para alimentar um díodo emissor de luz (LED). O circuito deve fornecer corrente suficiente para acender o LED e evitar danos ao mesmo. O circuito mais simples para acionar um LED é através de uma resistência em série [3].

Procedimento:

Escolhendo os terminais 16, 33, GNDs e 3.3V do ESP32 como referência deve considerar-se o seguinte circuito:

Instruções:

(para uma primeira utilização do ESP32, sugere-se o tutorial *Primeiro Programa* [2])

- Montar o circuito esquematizado anteriormente (ver Secção **Montagem exemplificativa** , para mais detalhe)
- Ligar a placa ESP32 ao computador por via de cabo USB micro
- Abrir o IDE Arduino
- Selecionar o dispositivo ESP32 adequado
- Copiar o seguinte código

```
/*
* Potenciómetro e LED
*/

#define POTENTIOMETER_PIN  33  // terminal GPIO33 do ESP32 ligado ao terminal de saída
// do potenciómetro
#define LED_PIN            16  // terminal GPIO16 do ESP32 ligado  ao LED

// a função setup é executada pontualmente quando o botão reset é premido ou
// a placa é alimentada:
void setup() {
  // inicializa a comunicação série a 115200 bits por segundo:
  Serial.begin(115200);

  // declara o terminal do LED como saída:
  pinMode(LED_PIN, OUTPUT);
}

// a função loop é executada sucessivamente, uma e outra vez, e por aí em diante
void loop() {
  // leitura do valor no terminal analógico (valor entre 0 e 4095)
  int analogValue = analogRead(POTENTIOMETER_PIN);

  // normaliza valor lido para valor de interesse (valor entre 0 e 255)
  int brightness = map(analogValue, 0, 4095, 0, 255);

  // define intensidade do LED
  analogWrite(LED_PIN, brightness);

  // imprime valores para a comunicação série
  Serial.print("valor lido = ");
  Serial.print(analogValue);
  Serial.print(" => intensidade = ");
  Serial.println(brightness);
```

```
  delay(100);
}
```

- Compilar e carregar o código para a placa ESP32

- Verificar o resultado

Montagem exemplificativa:
