

[Programa] Potenciómetro

Objectivo:

Utilizar o *ESP32-WROOM-32E DevKitC v4* [3] para ler valor de tensão de saída de um potenciómetro.

Lista de material:

- *ESP32-WROOM-32E DevKitC v4*
- Breadboards
- Potenciómetro de 10k Ω
- Fios de ligação
- Cabo USB micro

Contextualização:

Um potenciómetro é um componente eletrónico que incorpora uma resistência elétrica ajustável [5].

Potenciómetro

Um potenciómetro possui tipicamente três terminais incluindo um contacto ajustável, deslizante ou rotativo, que forma um divisor de tensão ajustável [5].

eg_out_2.png	eg_0_2.png	eg_120_2.png	eg_240_2.png
------------------------------	----------------------------	------------------------------	------------------------------

ESP32-WROOM-32E DevKitC v4

Apesar de se aconselhar a consulta da página do fabricante sobre o kit *ESP32-WROOM-32E DevKitC v4* [3] é possível considerar que em determinadas situações os terminais de entrada do microcontrolador ESP32 podem converter tensão (algo como uma quantidade analógica entre 0V e 3.3V) em um valor digital inteiro entre 0 e 4095 (num total de $2^{12}=4096$ representações possíveis, [1][2, SAR ADC]).

[esp32.png](#)

Procedimento:

Atentando ao objetivo enunciado, e escolhendo os terminais 33 (para o terminal de saída do potenciómetro), 3.3V e o GND do ESP32 como referência deve considerar-se o seguinte circuito:

[interruptor_montagem_2.png](#)

Instruções:

(para uma primeira utilização do ESP32, sugere-se o tutorial *Primeiro Programa* [4])

- 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
- Seleccionar o dispositivo ESP32 adequado
- Copiar o seguinte código

```
/*
 * Potenciómetro
 */

float floatMap(float x, float in_min, float in_max, float out_min, float out_max) {
  return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}

// 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);
}

// a função loop é executada sucessivamente, uma e outra vez, e por aí em diante
void loop() {
  // leitura do valor de entrada no terminal analógico GPIO33:
  int analogValue = analogRead(33);
  // Normalização dos valores lidos para a tensão do potenciómetro
  float voltage = floatMap(analogValue, 0, 4095, 0, 3.3);
  //
  // print out the value you read:
  Serial.print("valor lido: ");
  Serial.print(analogValue);
  Serial.print(", tensão: ");
  Serial.println(voltage);
  delay(1000);
}
```

- Compilar e carregar o código para a placa ESP32

arduino.png

- Verificar o resultado

--	--

Montagem exemplificativa:

	setup_b_1.png	

Referências:

[1] Ltd. Espressif Systems Co. Analog to Digital Converter (ADC). url:

<https://docs.espressif.com/projects/esp-idf/en/release-v4.4/esp32/api-reference/peripherals/adc.html> (acedido em 23/08/2023).

[2] Ltd. Espressif Systems Co. ESP32 Technical Reference Manual. Versão 5.0, pp. 631–637. url:

https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf (acedido em 18/08/2023).

[3] Ltd. Espressif Systems Co. ESP32-DevKitC V4 Getting Started Guide. url:

<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-devkitc.html> (acedido em 18/08/2023).

[4] Paulo Menezes. Artigos de Suporte - DEEC - Kit Eletrónica - Primeiro Programa. url:

<https://kb.deec.uc.pt/books/deec/page/primeiro-programa> (acedido em 17/08/2023).

[5]Wikipedia. Potentiometer. url: <https://en.wikipedia.org/wiki/Potentiometer> (acedido em 23/08/2023).

Revision #4

Created 2023-08-24 18:57:05 UTC by João Pedro Monteiro

Updated 2024-08-27 16:07:27 UTC by João Pedro Monteiro