

PROFESSOR DANILO

AULA 05

AULA PASSADA

Precisa consultar o conteúdo da aula passada? Não se preocupe, pois você pode baixar o arquivo pelo QR-code abaixo.

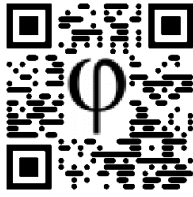


Figura 1: QR-code da aula da semana passada

Na semana passada você aprendeu um pouco sobre como funcionam as portas digitais e analógicas do Arduino.

BAIXE ESTA AULA NO SEU CELULAR

O QR-code da Figura 2 serve para você baixar este arquivo no seu celular, se você preferir.



Figura 2: Baixe aqui este arquivo

ATIVIDADE EM AULA

Objetivo da aula:

1. Utilizando a saída de 5V do Arduino e um potenciômetro, controlar o brilho do LED;
2. Entender o funcionamento de um transistor e um capacitor;
3. Entender um pouco mais sobre a diferença entre as portas digitais e analógicas;
4. Como atividade para casa, ter os primeiros contatos com a linguagem de programação C.

PRIMEIRO CIRCUITO

Conecte o potenciômetro no Arduino para que o brilho do LED possa ser controlado. Para isso você irá conectar o pino central (pino 2) do potenciômetro ao negativo do LED e o positivo do LED à um resistor e depois no 5V do Arduino. O pino 1 à terra (GND) do Arduino e o pino 3 do 5V do Arduino.

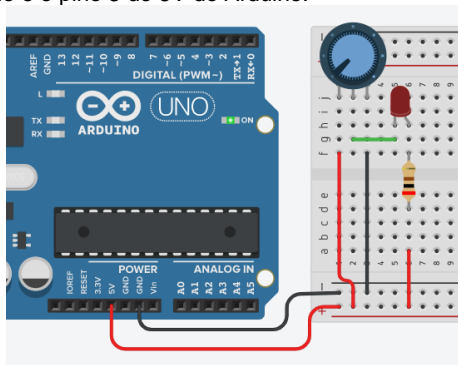


Figura 3: Detalhe do circuito a ser montado. Note que na posição do potenciômetro da figura o LED fica desligado pois ambos os lados do LED estarão conectados no 5V.

PROJETOS DE CIÊNCIAS – 21/02/2022

Lembre-se que somente poderemos acender o LED se as duas pernas do LED estiverem em potenciais diferente, isto é, conectados em GND e 5V, por exemplo.

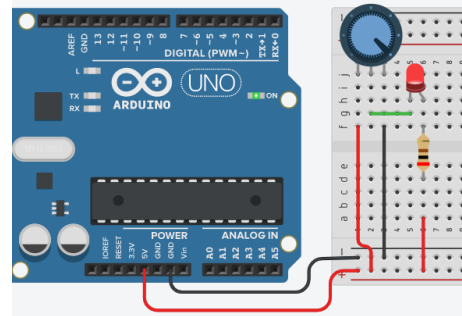


Figura 4: Mesmo circuito que o anterior, mas agora com o potenciômetro conectando o pino do meio com o pino da direita.

Vamos entender melhor este circuito começando pelo funcionamento do potenciômetro.



Figura 5: Detalhe de um potenciômetro, como o que cada grupo possui em seu kit. Chamaremos de pino 1 o pino mais a esquerda, pino 2 o central e pino 3 o da direita.

A Figura 6 mostra detalhe interno de um potenciômetro, que nos permite variar a resistência os pinos 1 e 2 e entre 2 e 3. A resistência entre os pinos 1 e 3 será sempre constante. No caso da nossa aula, será de 10 kΩ.

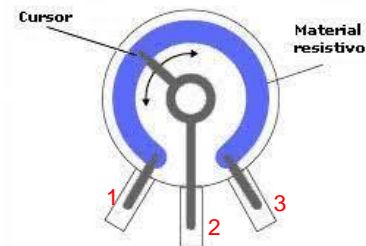


Figura 6: Detalhe do funcionamento de um potenciômetro. Conforme giramos o botão do potenciômetro no sentido anti-horário, temos a diminuição a resistência entre o pino 1 e 2 e aumentar a resistência entre os pinos 2 e 3; por outro lado, conforme giramos no sentido horário, a resistência entre os pinos 2 e 3 diminui e a resistência entre os pinos 1 e 2 aumenta.

Se tudo der certo, você vai conseguir controlar o brilho do LED sem problema nenhum e seu primeiro objetivo foi atingido.

SEGUNDO CIRCUITO

Vamos entender a base do que chamamos de transistor. Como qualquer outro componente que utilizaremos neste laboratório, o que precisamos entender é como ele se comporta no nosso circuito e o que podemos fazer com ele, portanto, não estaremos interessados em entender mais a fundo como ele funciona.

Observe a Figura 7, na qual apresentamos um esquema do transistor que você recebeu, o seu símbolo e o nome dos seus pinos.

Sempre que você tiver que trabalhar com algum transistor, a primeira coisa que você deve pensar é “datasheet do transistor”.

PROFESSOR DANILO

Por isso, na Figura 8 você tem o QR-code para baixar o datasheet do transistor que você tem em mãos.

Por hora, a única coisa que te interessa é saber se temos um transistor NPN ou PNP. Nós trabalharemos com o primeiro, por enquanto.

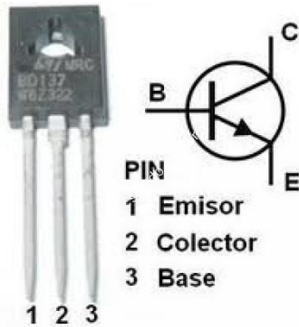


Figura 7: Transistor que utilizaremos no laboratório do lado esquerdo e o seu símbolo do lado direito.



Figura 8: Datasheet do transistor BD 137, utilizado em sala de aula. Clique na imagem para acessar o arquivo, caso esteja vendo no celular.

O transistor NPN faz referência à tensão, respectivamente, no Coletor, Base e Emissor. Neste caso, quando você ligar uma tensão **Positiva** na Base, o Coletor se conectará ao Emissor. Veja isso na Figura 9.

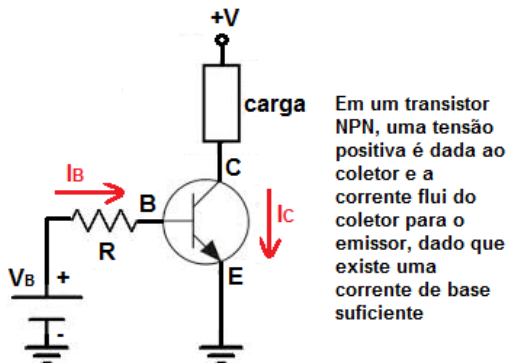


Figura 9: Circuito utilizado para o acionamento de um transistor tipo NPN. Fonte: <http://www.learningaboutelectronics.com/Artigos/Diferenca-entre-transistores-NPN-e-PNP.php>

O transistor PNP faz referência à tensão, respectivamente, no Coletor, Base e Emissor, assim como o caso anterior. Neste caso, quando você ligar uma tensão **Negativa** na Base, o Coletor se conectará ao Emissor. Veja isso na Figura 10.

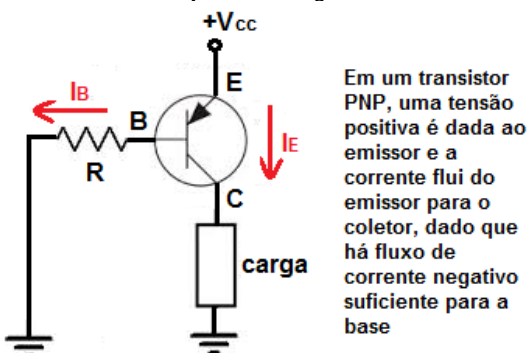


Figura 10: Circuito utilizado para o acionamento de um transistor tipo PNP. Fonte: <http://www.learningaboutelectronics.com/Artigos/Diferenca-entre-transistores-NPN-e-PNP.php>

PROJETOS DE CIÊNCIAS – 21/02/2022

Qual seria a aplicação dos transistores? Basicamente, podemos imaginá-lo como um amplificador, pois, consultando o datasheet do transistor que você recebeu, podemos ver a Figura 11, na página 2 do pdf.

Em especial, veja a quarta linha que contém o h_{FE} . Este é o chamado ganho efetivo do transistor que representar quantas vezes a corrente do coletor é maior que a corrente na base.

$$h_{FE} = \frac{i_C}{i_B} \quad \text{Equação 1}$$

em que i_C é a corrente que passa pelo coletor e i_B é a corrente que circula na base.

ELECTRICAL CHARACTERISTICS ($T_C = 25^\circ\text{C}$ unless otherwise noted)

Characteristic	Symbol	Type	Min	Max	Unit
Collector-Emitter Sustaining Voltage* ($I_C = 0.03 \text{ Adc}$, $I_B = 0$)	V_{CE0}	BD 135 BD 137 BD 139	45 60 80	—	Vdc
Collector Cutoff Current ($V_{CB} = 30 \text{ Vdc}$, $I_E = 0$) ($V_{CB} = 30 \text{ Vdc}$, $I_E = 0$, $T_C = 125^\circ\text{C}$)	I_{CBO}		—	0.1 10	μAdc
Emitter Cutoff Current ($V_{BE} = 5.0 \text{ Vdc}$, $I_C = 0$)	I_{EBO}		—	10	μAdc
DC Current Gain ($I_C = 0.005 \text{ A}$, $V_{CE} = 2 \text{ V}$) ($I_C = 0.15 \text{ A}$, $V_{CE} = 2 \text{ V}$) ($I_C = 0.5 \text{ A}$, $V_{CE} = 2 \text{ V}$)	h_{FE}		25 40 25	— 250 —	—
Collector-Emitter Saturation Voltage* ($I_C = 0.5 \text{ Adc}$, $I_B = 0.05 \text{ Adc}$)	$V_{CE(sat)}$		—	0.5	Vdc
Base-Emitter On Voltage* ($I_C = 0.5 \text{ Adc}$, $V_{CE} = 2.0 \text{ Vdc}$)	$V_{BE(on)}$		—	1	Vdc

* Pulse Test: Pulse Width $\leq 300 \mu\text{s}$, Duty Cycle $\leq 2.0\%$.

Figura 11: Analise atentamente esta tabela. Preste especial atenção na quarta linha, onde é apresentado o h_{FE} .

O transistor que estamos estudando permite um h_{FE} máximo de 250. Como exemplo, o Arduino pode fornecer no máximo 40 mA, ou 0,040 A, o suficiente para acender um LED, mas não para acionar um motor. Utilizando a Equação 1, vemos o limite da corrente que, agora, podemos acionar com o transistor:

$$h_{FE} = \frac{i_C}{i_B} \Rightarrow 250 = \frac{i_C}{0,040} \Rightarrow i_C = 10 \text{ A.}$$

Esta corrente é mais que suficiente para acionar qualquer elemento que iremos utilizar nesta disciplina. Já se você precisar de trabalhar com corrente maior e com tensão maior, é possível utilizar de outros componentes, como relé (módulo presente no seu kit).

Um segundo elemento que iremos ver agora é o capacitor: ele é capaz de acumular uma pequena quantidade de energia e é útil para criarmos temporizadores, como o limpador intermitente do para brisa dos carros.

Na Figura 12, temos o detalhe de como é um capacitor clássico. É lógico que, com o desenvolvimento tecnológico, conseguimos fazer capacitores de diversos tipos, inclusive com materiais maleáveis que possam ser enrolados e colocados em recipientes cilíndricos, como o apresentado na Figura 13.

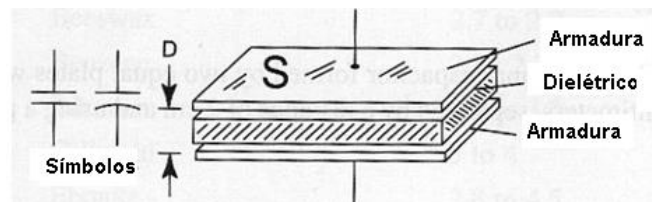


Figura 12: Detalhe de um capacitor plano à direita e correspondentes símbolos à esquerda. As armaduras são metálicas e o dielétrico é um material isolante: quando ligamos as armaduras em potenciais elétricos diferentes, cargas elétricas serão acumuladas nas placas (armaduras).

Nosso objetivo aqui é montar um circuito como o apresentado na Figura 14. Não apresento aqui o circuito no tinkercad porque a ordem dos pinos do transistor lá é diferente da ordem dos pinos do transistor que temos em mãos.

PROFESSOR DANILO

PROJETOS DE CIÊNCIAS – 21/02/2022



Figura 13: Detalhe de um capacitor eletrolítico. Note que ele é polarizado, sendo a perna maior conectada no positivo e a menor é conectada no negativo.

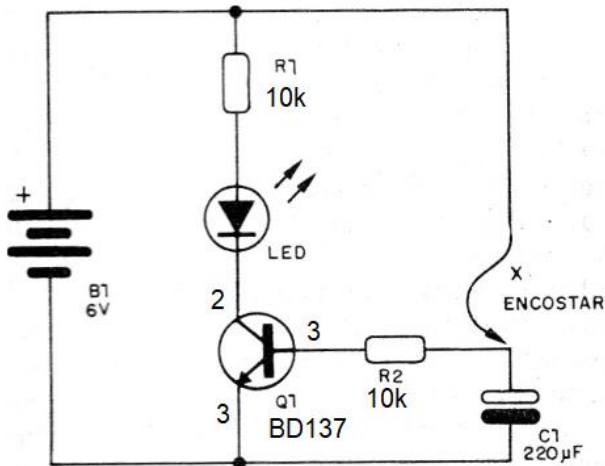


Figura 14: circuito que iremos montar. Veja circuito utilizado como inspiração em <https://www.newtonbraga.com.br/index.php/projetos/15087-quatro-circuitos-para-saber-como-funciona-um-transistor-art1698.html>

Se o circuito der certo, o LED ficará aceso quando você conectar o fio X no capacitor e permanecer aceso, por algum tempo, depois de ser removido o fio X. Isso porque o capacitor acumulará energia elétrica que será liberada lentamente na base do transistor e quando a energia acumulada no capacitor for pouca, o LED irá apagar.

COMPREENDENDO AS PORTAS DO ARDUINO

Vamos agora nos aprofundar um pouco mais sobre as portas que apresentam no Arduino.

GND

Como vimos nas aulas passadas, a energia elétrica deverá fluir do positivo ao negativo da bateria. No Arduino, o negativo será chamado de GND (ou Ground ou ainda chão/terra).

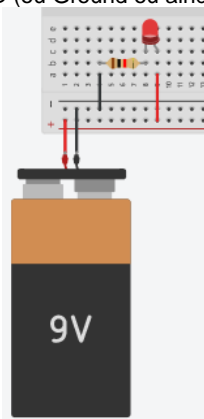


Figura 15: circuito feito no tinkercad que possibilita acender um LED. Note que a resistência é necessária para que o LED não queime e é necessário que o circuito seja fechado, isto é, que haja continuidade nos fios conectando o positivo (esquerda na bateria) ao negativo (direita na bateria).

Na figura abaixo vemos as portas com o negativo, tal como o menos (-) em uma bateria ou pilha.

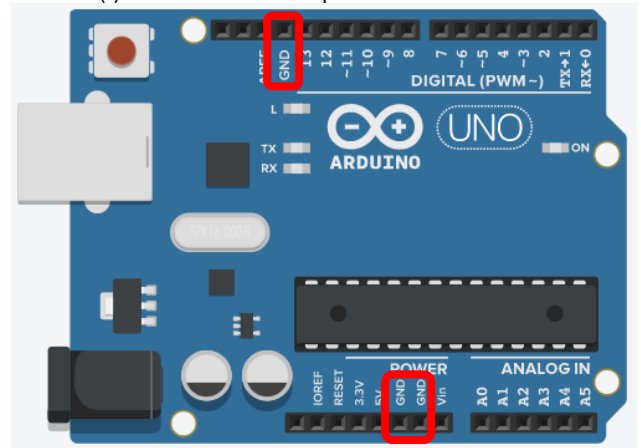


Figura 16: detalhes das portasa GND do Arduino.

PORTAS DIGITAIS

Os pinos de 0 a 13 no Arduino são chamados de portas digitais e os pinos de A0 até A5 são chamados de portas analógicas.

Os pinos de 0 a 13 podem fornecer 5 V e quando isso acontece dizemos que elas funcionam como SAÍDAS DIGITAIS. Você também pode conectar uma fonte de 5 V em uma destas portas fornecendo, portanto, energia elétrica para o Arduino (sem a intenção de alimentá-lo, é só um sinal) e neste caso dizemos que o pino que recebe 5 V está funcionando como uma ENTRADA DIGITAL.

As portas de A0 até A5 são chamadas de portas analógicas e apenas recebem sinais, portanto são apenas ENTRADAS ANALÓGICAS. O Arduino não possui portas de saídas analógicas, no entanto ele é capaz de simular tais portas.

Antes de continuarmos devemos entender a diferença entre analógico e digital: se forma simplificada, imaginemos um sinal elétrico que possui valor mínimo igual a 0 V e máximo 5 V. Um sinal digital somente aceita os valores mínimo e máximo, portanto não temos valores intermediários. Costumamos pensar que só são possíveis dois sinais: ligado e desligado (outros preferem chamar de 0 ou 1 e justamente por isso que alguns botões possuem os símbolos 0 e 1 para representar que algo está ligado ou desligado).



Figura 17: Portas digitais apresentam apenas os níveis ligado e desligado, representados, respectivamente, por 1 e 0 (um e zero). Veja as chaves acima: à da esquerda, temos dois botões, sendo que o de cima serve para ligar e o de baixo para desligar. Na figura à direita, temos a sobreposição dos botões que representam 1 e 0, portanto, o mesmo botão serve para ligar e deligar.

Graficamente, podemos representar um sinal digital como se segue:

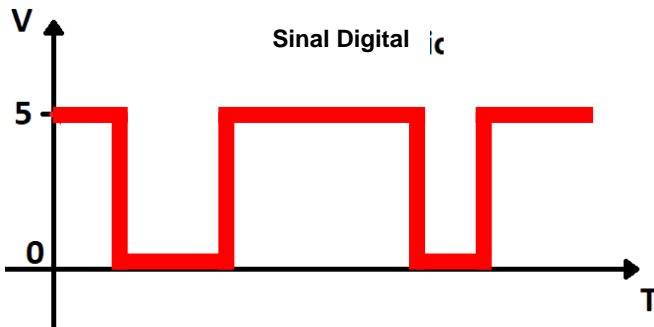


Figura 18: representação do sinal digital.

Veja a seguir quais são as portas digitais do Arduino.

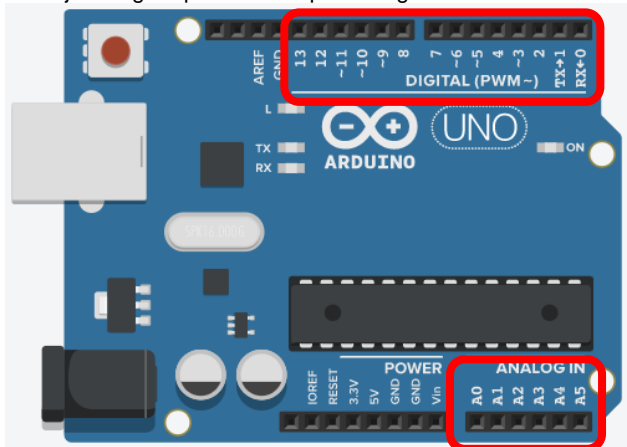


Figura 19: portas que podem ser usadas como entrada ou saída digitais. Note que até mesmo as portas analógicas (detalhada a seguir) podem ser usadas como portas digitais.

Em um programa no Arduino informaremos que uma porta deve estar ligada usando o número 1 ou a palavra *HIGH* que, em tradução livre, podemos dizer que significa alto ou ligado. Para indicarmos que uma saída digital está desligada, usamos o número 0 ou a palavra *LOW* que podemos entender como baixo ou desligado.

Se uma porta digital estiver sendo usada como entrada, usaremos comandos para ler estas portas e a leitura nos informará, de mesma maneira, 0 ou *LOW* se a porta digital estiver conectada ao GND e 1 ou *HIGH* se conectada a 5 V.

PORTAS ANALÓGICAS

E a porta analógica? Ou melhor, e o sinal analógico, como funciona? Neste caso, não é só zeros e uns que ele aceita, mas também aceita outros valores. Ainda no exemplo de 0 a 5 V, podemos usar uma resistência variável associada a um botão para mudar a tensão continuamente conforme giramos o botão. Como ótimo exemplo, temos os botões que podemos girar para aumentar o volume de um rádio ou ainda o controle analógico de um vídeo game.



Figura 20: para entendermos o que é um sinal analógico, podemos utilizar o potenciômetro, tal como o utilizado na sala de aula. Note que no nosso primeiro circuito, desta aula, vemos que podemos controlar o brilho do LED alterando o valor da resistência do potenciômetro. Esse é um ótimo exemplo de sinal analógico.

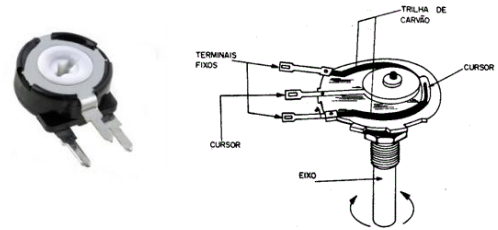


Figura 21: à esquerda, temos um outro tipo de potenciômetro e à direita temos um detalhe de funcionamento de um potenciômetro, como apresentado no primeiro circuito feito em aula.



Figura 22: dois outros bons exemplos de dispositivos que usam inais analógicos são o touch pad e os botões direcionais indicados na figura acima.

Graficamente podemos representar um sinal analógico como se segue:

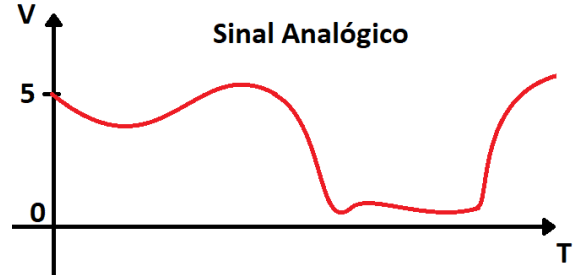


Figura 23: exemplo de um sinal analógico. Veja que sinais intermediários são possíveis.

Na figura abaixo vemos as portas analógicas do Arduino.

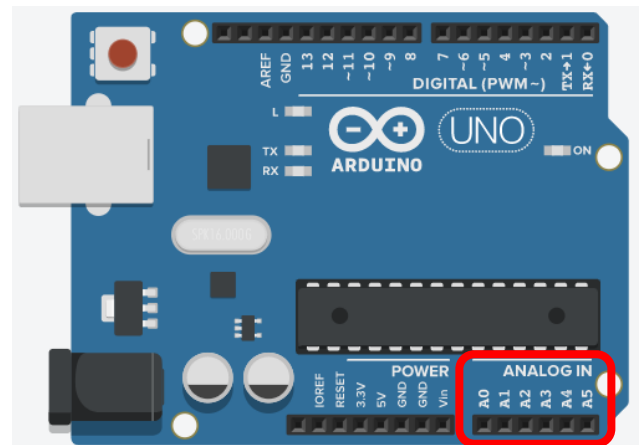


Figura 24: indicação das portas analógicas. Note que tais portas, no Arduino, só funcionam como portas de entrada. O Arduino NÃO possui saídas analógicas, mas este problema pode ser contornado de pelo menos duas maneiras: comprando um módulo conversor Digital Analógico chamado DAC ou utilizando as portas PWM. Tais soluções serão apresentadas a seguir.

Em um programa no Arduino, quando uma porta analógica for lida, esta fornecerá valores entre 0 e 255 sendo 0 quando estiver conectado ao GND e 255 quando conectado a 5 V.

Recapitulando:

- No Arduino temos portas de entrada e saída digitais;
- No Arduino temos apenas portas de entrada analógica. Não existem portas de saída analógica.

PROFESSOR DANILO

DAC

DAC, do inglês *Digital-to-Analog Converter*, que em português fica Conversor Analógico-Digital é um circuito eletrônico capaz de converter um sinal digital (por exemplo, enviado pelo Arduino) em um sinal analógico. Como vimos, o Arduino não possui saídas analógicas, mas não tem problema, pois iremos nos virar bem sem este conversor.

Existem muitos conversores no mercado, sendo um exemplo o DAC MCP4725 I2C mostrado abaixo.

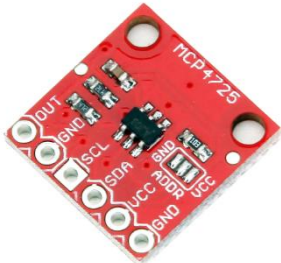


Figura 25: módulo DAC. Não utilizaremos neste curso.

Não se assuste com os nomes e estes montes de números: eles possuem significados importantes, mas por enquanto imagine apenas que são nomes dados a estes componentes. Diferentes de nomes de pessoas, não é bom dar o mesmo nome a circuitos eletrônicos diferentes e como existem muitos circuitos elétricos com funcionalidades diversas, fica mais prático darmos nomes com especificações técnicas no próprio nome. Como exemplo, I2C é um tipo de comunicação que o Arduino é capaz de fazer para, por exemplo, mostrar uma informação em um display simples e neste tipo de comunicação teremos sempre os terminais SCL e DAS como visto na figura acima.

Resumindo: o Arduino fornece um número em formato digital e o DAC fornece uma tensão (voltagem) correspondente.

PWM

Então, como simular um sinal analógico de saída no Arduino? A resposta está nas portas digitais PWM marcadas com um ~ no Arduino.

Para entender o que é PWM, recomendo que leiam o texto do seguinte link do qual extraí algumas imagens e informações:

<https://www.embarcados.com.br/pwm-do-arduino/>

Na figura a seguir destacamos as portas PWM do Arduino UNO. São elas as portas digitais 3, 5, 6, 9, 10 e 11.

Não precisa decorar, pois basta localizar onde estão as portas com o ~ ao lado. Note também que no próprio Arduino ele te lembra ao escrever "DIGITAL PWM ~".

Abaixo da figura, transcrevo parte de um texto do site sugerido acima:

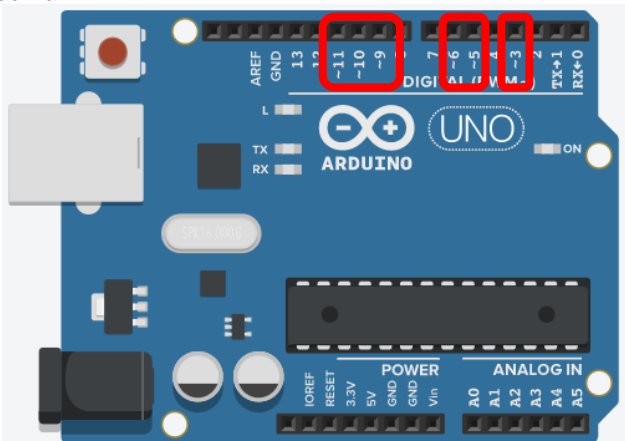


Figura 26: detalhe mostrando quais as portas do Arduino UNO funcionam como portas PWM.

PROJETOS DE CIÊNCIAS – 21/02/2022

"PWM, do inglês *Pulse Width Modulation*, é uma técnica utilizada por sistemas digitais para variação do valor médio de uma forma de onda periódica. A técnica consiste em manter a frequência de uma onda quadrada fixa e variar o tempo que o sinal fica em nível lógico alto. Esse tempo é chamado de *duty cycle*, ou seja, o ciclo ativo da forma de onda. No gráfico abaixo são exibidas algumas modulações PWM:"

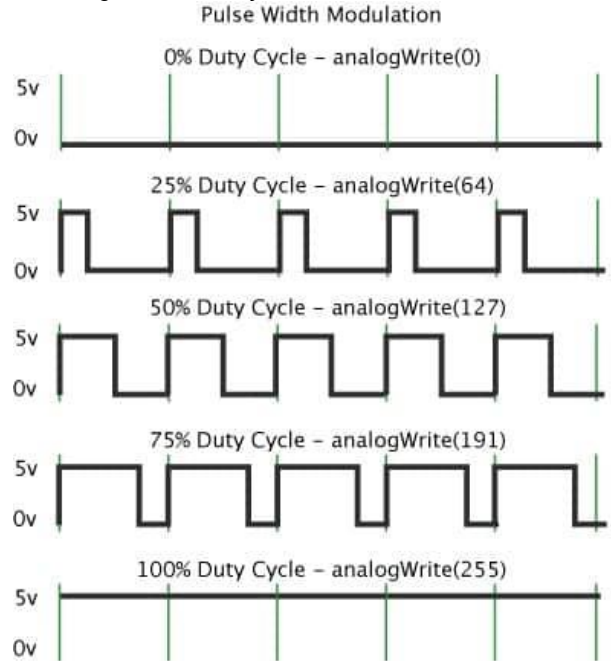


Figura 27: note que as portas PWM simulam uma saída digital ligando e desligando a porta muito rapidamente.

Olha que interessante: se você ficar ligando e desligando um LED muito rapidamente com, digamos, 5 V, o LED poderá funcionar como se estivesse sendo alimentado com uma tensão menor. Por exemplo, se usarmos uma tensão de 5 V e um *Duty Cycle* de 50% então é como se alimentássemos o LED com 2,5 V. Esta tensão equivalente chamaremos de V_{out} , a tensão máxima de alimentação (5 V no Arduino) de V_{CC} então podemos ter a seguinte relação:

$$V_{out} = \frac{Duty\ Cycle}{100} \cdot V_{CC}$$

As portas PWM podem ser usadas para controle de velocidade de motores, variação a luminosidade de LEDs, gerar sinais de áudio ou gerar sinais analógicos por razões diversas etc.

No Arduino, quando formos escrever o código, ao *Duty Cycle* não será indicado diretamente, isto é, não indicamos que o tempo de ciclo ativo é um número que varia de 0% (totalmente desligado) até 100% (totalmente ligado): usamos valores que variam de 0 a 255. Assim, a fórmula acima pode ser reescrita como:

$$V_{out} = \frac{Valor\ informado\ ao\ Arduino}{255} \cdot V_{CC}$$

Como exemplo, se informarmos 0, então a saída ficará completamente desligada; informando 255 ficará totalmente ligada; informando 127, ficará desligado o mesmo tempo que ficaria desligado e assim por diante.

FONTES DE ENERGIA

No Arduino o polo positivo poderá ser:

- 3.3V
- 5V
- Vin
- Portas digitais (de 0 a 13)

PROFESSOR DANILO

A primeira porta indicada acima está destacada na imagem abaixo e nos fornece 3,3 V. Alguns componentes eletrônicos irão funcionar com esta tensão, como módulos bluetooth.

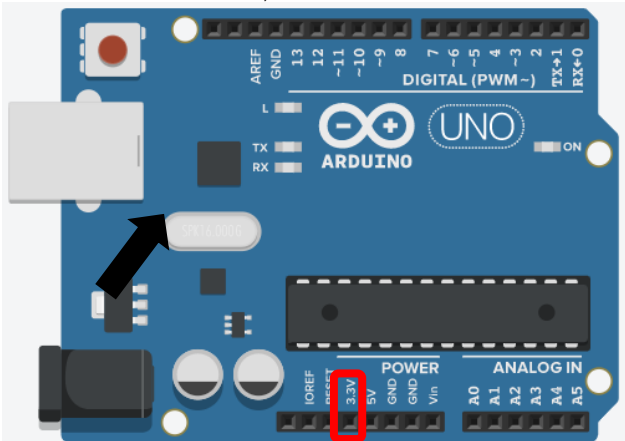


Figura 28: detalhe mostrando a saída de 3,3V. Este sinal, quando inserido numa entrada digital, também é reconhecido pelo Arduino como 1 (ligado).

Veja que a segunda porta (5V) nos fornece 5 volts (como comparação, a bateria de seu celular fornece cerca de 4,7 V e a maioria dos periféricos conectados no PC funcionam com 5 V como mouse, teclado ou mesmo seu celular – a fonte do celular fornece 5 V). Veja na figura a seguir esta porta marcada.

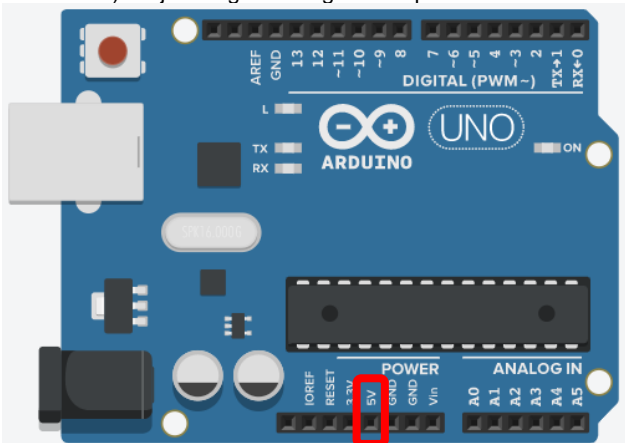


Figura 29: detalhe mostrando a saída de 5V

Vin quer dizer algo como *V input*, isto é, o valor desta tensão depende da alimentação do Arduino: quando conectado no computador, ele fornece a tensão da porta USB conectada na porta indicada pela seta na figura abaixo.

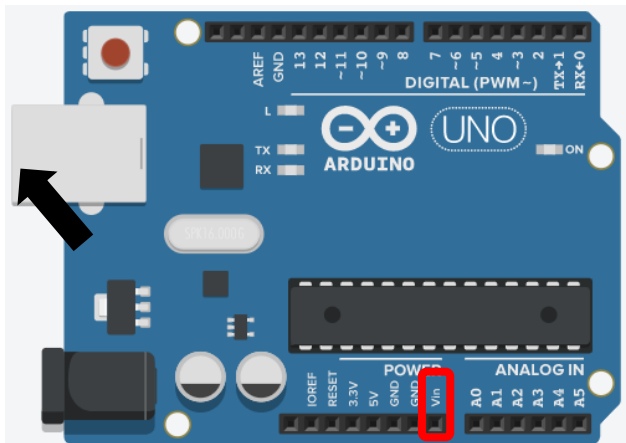


Figura 30: detalhe mostrando a saída Vin que possui o mesmo valor que o fornecido ao Arduino. Note que se ligarmos a fonte de energia no conector USB tipo B, Vin será de 5V, pois esta é a tensão padrão usada nas portas USB.

PROJETOS DE CIÊNCIAS – 21/02/2022

O Arduino pode ter outra alimentação conforme indicado na figura abaixo e você pode conectar o Arduino à esta outra fonte mesmo se a USB já estiver conectada. Se fonte externa estiver conectada e o Arduino estiver ligado na USB, Vin será 5 V se a tensão de alimentação for menor que 7,4 V, mas se a tensão de entrada for maior que 7,4 V então Vin será igual à tensão fornecida no conector de alimentação (figura abaixo). Se somente uma das alimentações estiverem conectadas, então Vin será igual à tensão usada para alimentar o Arduino (nota: a tensão de alimentação recomendada é de 7 V até 12 V).

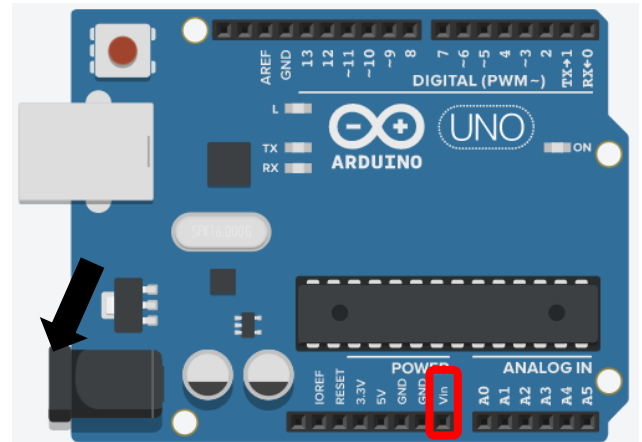


Figura 31: detalhe mostrando a saída Vin que possui o mesmo valor que o fornecido ao Arduino. Note que podemos conectar até 12V no conector fêmea P2 e que a tensão Vin será igual a esta tensão.

Ainda há mais o que estudar no Arduino, mas por enquanto isso já é o suficiente.

A atividade dissertativa será baseada neste arquivo.

ÚLTIMA ATIVIDADE

Acesse o site da disciplina, na aula 6 para ter seus primeiros contatos com a linguagem C.



Figura 32: acesse aqui, clicando na imagem ou escaneando-a, o site do professor para esta disciplina. Site: <https://fisica.professordaniilo.com/lab.html#aula5>

Você pode testar os comandos da linguagem no próprio site. É ALTAMENTE RECOMENDÁVEL que você estude isso continuamente.