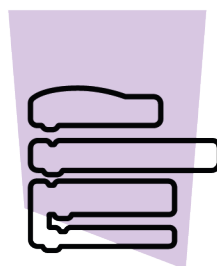
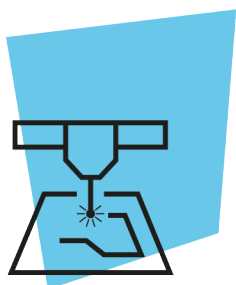
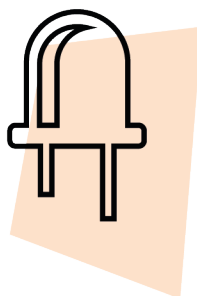
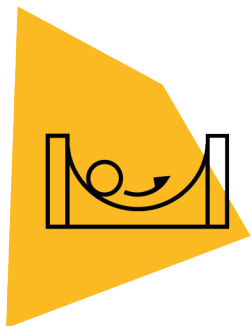
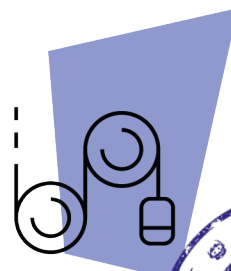


ROBÓTICA

Módulo 1



Portas Digitais

AULA 06

GOVERNADOR DO ESTADO DO PARANÁ

Carlos Massa Ratinho Júnior

SECRETÁRIO DE ESTADO DA EDUCAÇÃO

Renato Feder

DIRETOR DE TECNOLOGIA E INOVAÇÃO

Andre Gustavo Souza Garbosa

COORDENADOR DE TECNOLOGIAS EDUCACIONAIS

Marcelo Gasparin

Produção de Conteúdo

Cleiton Rosa

Simone Sinara de Souza

Revisão Textual

Adilson Carlos Batista

Leitura Crítica e Normalização Bibliográfica

Ricardo Hasper

Projeto Gráfico e Diagramação

Edna do Rocio Becker

Ilustração

Jocelin Vianna (Educa Play)

Fotografia

Stella Maris Oliveira Ludwig (Educa Play)

2021

| | |
|---------|--|
| Aula 01 | Por Que Robótica? |
| Aula 02 | Tensão, Corrente e Resistência |
| Aula 03 | Kit de Robótica |
| Aula 04 | Arduino Uno R3 |
| Aula 05 | Softwares Arduino IDE e mBlock |
| Aula 06 | Portas Digitais |
| Aula 07 | Circuito Elétrico |
| Aula 08 | LED e Resistor |
| Aula 09 | Semáforo [Carros] |
| Aula 10 | Semáforo [Cruzamento Carros] |
| Aula 11 | Semáforo [Pedestres] |
| Aula 12 | Semáforo [Cruzamento Carros + Pedestres] |
| Aula 13 | Push Button |
| Aula 14 | Feedbacks + Inventário I |
| Aula 15 | Semáforo [Carros + Pedestres com Botão] |
| Aula 16 | Display 7 Segmentos |
| Aula 17 | Fonte DC + Plug P4 |
| Aula 18 | Portas PWM |
| Aula 19 | LED Fade-In |
| Aula 20 | LED Fade-Out |
| Aula 21 | Super Máquina 80's |
| Aula 22 | Super Máquina 2008 |
| Aula 23 | Potenciômetro |
| Aula 24 | Buzzer Passivo |
| Aula 25 | LED RGB |
| Aula 26 | Arco-Iris |
| Aula 27 | Sensor LDR |
| Aula 28 | Feedbacks + Inventário II |
| Aula 29 | Sensor de Temperatura |
| Aula 30 | Sensor de Obstáculo IR |
| Aula 31 | Controle Motor DC |
| Aula 32 | Kit Chassi 2WD Robô |
| Aula 33 | Seguidor de Linha |
| Aula 34 | Sensor de Distância |
| Aula 35 | Sensor de Estacionamento |
| Aula 36 | Display LCD 16x2 |
| Aula 37 | Trena Digital |
| Aula 38 | Robô Sumô [Estrutura] |
| Aula 39 | Robô Sumô [Programação + Treinamento I] |
| Aula 40 | Robô Sumô [Programação + Treinamento II] |
| Aula 41 | Disputa de Sumôs |
| Aula 42 | Feedbacks + Inventário III |

Aula 05
Softwares Arduino IDE e mBlock

Aula 06

Portas Digitais

Aula 07
Circuito Elétrico

Sumário

| | |
|--|-----------|
| Introdução | 2 |
| Objetivos desta Aula | 2 |
| Competências Gerais Previstas na BNCC | 3 |
| Habilidades do Século XXI a Serem Desenvolvidas | 4 |
| Lista de Materiais | 4 |
| Roteiro da Aula | 4 |
| 1. Contextualização | 4 |
| 2. Montagem e Programação | 5 |
| 3. Feedback e Finalização | 13 |
| Videotutorial | 14 |
| Referências | 15 |



Introdução

A placa de prototipagem eletrônica Arduino Uno possui 14 pinos ou portas digitais que podem ser programadas para leitura ou gravação de um sinal digital, isto significa que podem ser usadas, conforme configuração, para atuarem como entrada (Input) ou saída (Output) digitais. Nesta aula, você aprenderá a configurar uma porta digital e a programar o acionamento de um LED, através de uma saída digital.



Objetivos desta Aula

- Configurar portas digitais para atuarem como entrada ou saída digitais;
- Identificar as funções `pinMode`, `digitalWrite` e `digitalRead` para uso dos pinos digitais;
- Definir a função `delay`;
- Programar um LED para piscar (acender e apagar).



Competências Gerais Previstas na BNCC

[CG02] - Exercitar a curiosidade intelectual e recorrer à abordagem própria das ciências, incluindo a investigação, a reflexão, a análise crítica, a imaginação e a criatividade, para investigar causas, elaborar e testar hipóteses, formular e resolver problemas e criar soluções (inclusive tecnológicas) com base nos conhecimentos das diferentes áreas.

[CG04] - Utilizar diferentes linguagens – verbal (oral ou visual-motora, como Libras, e escrita), corporal, visual, sonora e digital –, bem como conhecimentos das linguagens artística, matemática e científica, para se expressar e partilhar informações, experiências, ideias e sentimentos em diferentes contextos e produzir sentidos que levem ao entendimento mútuo.

[CG05] - Compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva.

[CG09] - Exercitar a empatia, o diálogo, a resolução de conflitos e a cooperação, fazendo-se respeitar e promovendo o respeito ao outro e aos direitos humanos, com acolhimento e valorização da diversidade de indivíduos e de grupos sociais, seus saberes, identidades, culturas e potencialidades, sem preconceitos de qualquer natureza.

[CG10] - Agir pessoal e coletivamente com autonomia, responsabilidade, flexibilidade, resiliência e determinação, tomando decisões com base em princípios éticos, democráticos, inclusivos, sustentáveis e solidários.



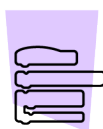
Habilidades do Século XXI a Serem Desenvolvidas

- Pensamento crítico;
- Afinidade digital;
- Resiliência;
- Resolução de problemas;
- Colaboração;
- Comunicação.



Lista de Materiais

- Computador ou notebook com porta USB;
- Placa de prototipagem Arduino Uno com cabo USB.



Roteiro da Aula

1. Contextualização (15min):

Na **Aula 4 - Arduino Uno R3**, estudamos que a placa de prototipagem eletrônica Arduino, possui pinos ou portas que atuam como entrada e saída digitais responsáveis pela interação deste Hardware com o meio externo, possibilitando a execução de programas, os quais podem ser desde o piscar de um LED até funções que envolvem programação complexa.

Partindo do conhecimento adquirido, até o momento, com as aulas de robótica, quais ações são necessárias, ao nível estrutural e de programação, para que um componente eletrônico simples como um LED possa piscar?

2. Montagem e Programação (60min):

Entre as estruturas presentes na placa de prototipagem eletrônica, destacamos, na aula de hoje, os **pinos** ou **portas** que atuam como **entradas** ou **saídas digitais**. No Arduino Uno, estas portas são representadas pelos números de 0 a 13, como mostra a figura 1. Para definir o modo de entrada e saída dos pinos, faz-se necessário configurá-los em sua programação.

Figura 1 - Portas digitais



Antes de programarmos a configuração das portas digitais do Arduino Uno, no modo entrada ou saída, precisamos entender que esses modos são responsáveis pela leitura e escrita de um valor digital, seja ele de 0V ou de 5V.

O modo **entrada digital** permite a leitura de componentes eletrônicos que enviam sinais elétricos em 0V ou 5V. Podemos citar, como exemplos, os sensores e chaves com comandos On-Off. Quando programamos a porta como entrada digital, a colocamos em um estado de alta impedância elétrica, isto significa que o circuito absorve uma corrente elétrica muito baixa e o nível de tensão presente nesta porta oscila entre 0V e 5V. A estabilidade desta oscilação é feita por resistores pull-up, os quais facilitam a ligação de teclas e sensores no valor de 5V. Estes resistores fazem parte do microcontrolador ATmega328, presente na placa de Arduino Uno.

A **saída digital** utiliza o valor 5V para acionar um componente eletrônico conectado à placa de Arduino e o valor 0V para desacionar este componente. Podemos citar, como exemplos de componentes

que utilizam a saída digital: LED, transmissor, motor CC e relé. Ao contrário da porta de entrada digital, ao configurarmos um pino como saída digital, o colocamos em um estado de baixa impedância, o que permite ao circuito fornecer ou drenar a corrente elétrica neste pino, sem a necessidade da utilização de resistores.

2.1 Configurando as portas digitais

O software Arduino IDE possui funções que permitem programar a configuração das portas de entrada ou saída digitais da placa Arduino Uno, definir o nível lógico e realizar leituras destas portas, durante a escrita do código-fonte. Essas funções são denominadas de **pinMode**, **digitalWrite** e **digitalRead**.

A função **pinMode** é responsável pela configuração das portas como entrada (Input) ou saída (Output) digital e apresenta os seguintes parâmetros: **pin** ou pino que corresponde ao número da porta que deseja configurar, no caso do Arduino Uno pode ser de 0 a 13, e **Mode** ou modo que identifica a configuração da porta, podendo ser: **Input** para **entrada** digital ou **Output** para **saída** digital. Por exemplo, para configurar a porta 10 como saída digital, utilizamos o código de programação `pinMode (10, OUTPUT)`.

A partir da configuração da porta digital, o próximo passo é determinar o nível lógico desta porta e realizar leituras durante a execução da programação. Para tanto, utilizamos a função **digitalWrite**, a qual determina o nível lógico que a porta de saída digital deve apresentar, sendo LOW para **nível baixo**, correspondendo a 0V, ou HIGH para **nível alto**, correspondendo a 5V. Na programação desta função são utilizados os seguintes parâmetros: número da porta de saída digital e valor do nível lógico presente nesta. Por exemplo, para configurar a porta 10 em nível lógico alto, utilizamos o código de programação: `digitalWrite (10, HIGH)`.

Para a leitura do valor presente na porta de saída digital, utilizamos a função **digitalRead**, a qual informa se o pino apresenta um valor baixo, representado pelo parâmetro LOW, ou um valor alto, representado pelo parâmetro HIGH. Na codificação, o parâmetro LOW pode ser representado pelo binário 0, e o parâmetro HIGH pelo binário 1.

Na função digitalRead, estes dois parâmetros são classificados como **retorno** e determinam o estado ativado ou desativado de um sensor.



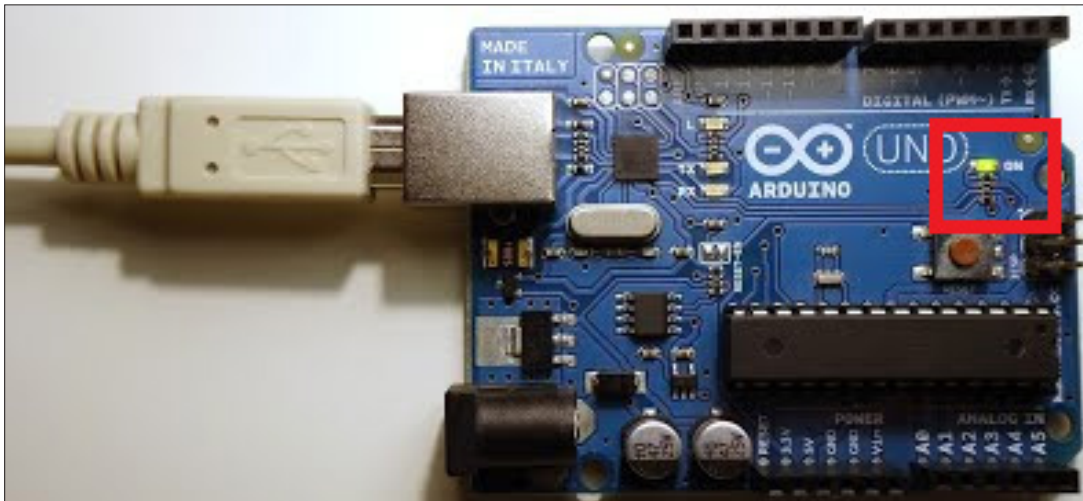
Agora, vamos programar!

Agora que você aprendeu a configurar uma porta digital e determinar seu nível lógico, vamos programar um LED, por codificação e por blocos, utilizando uma porta de saída digital para que esse pisque em intervalos de 1 segundo.

i. Linguagem de programação por código

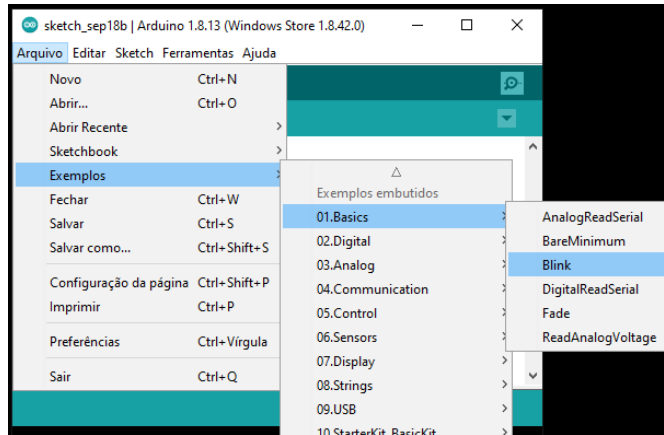
Para iniciar a programação, conecte a placa de Arduino Uno, através do cabo USB, ao computador ou dispositivo móvel. Ao realizar esta ação, observe que na placa Arduino o LED Alimentação (com indicação ON - figura 2) acenderá, apresentando a tonalidade verde, informando que a conexão entre ambos foi realizada.

Figura 2 - LED Alimentação aceso na placa de Arduino Uno



No computador ou dispositivo móvel, abra o software Arduino IDE e clique no menu **Arquivo**. Em **Exemplos**, clique em **Basics** e selecione o código **Blink**, conforme mostra a figura 3.

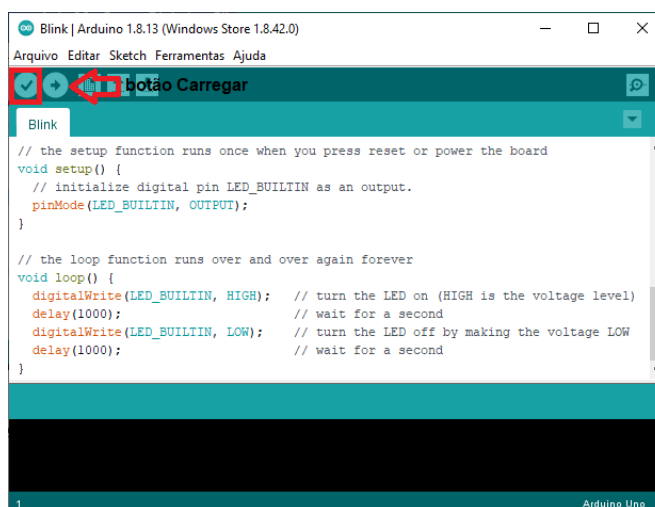
Figura 3 - Selecionando o código Blink



Fonte: site Arduino oficial

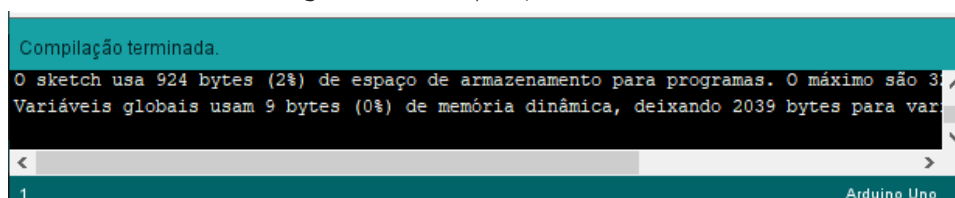
Ao selecionar a programação Blink, você será direcionado a página de Sketch contendo comentários e o código-fonte deste programa (figura 4). Observe que neste código aparece três funções, além das mencionadas na configuração das portas digitais, sendo elas: a função **void setup**, que define o pino LED_BUILTIN, já conectado à porta digital 13 da placa de Arduino Uno, como porta de saída (Output); a função **void loop**, que determina a repetição de ligar e desligar o LED; e a função **delay**, a qual determina uma pausa na execução do programa em intervalo equivalente a 1000 milissegundos, ou seja, 1 segundo.

Figura 4 - Código-fonte Blink



Após analisar o código-fonte, clique no botão **Verificar** (figura 4), representado por um Check (✓) para que ocorra a compilação do programa, ou seja, averiguar se há erros de sintaxe. Finalizada a compilação do código-fonte, a IDE apresentará uma mensagem informando a ausência de erros no programa (figura 5).

Figura 5 - Compilação finalizada



Na sequência, clique no botão **Carregar**, localizado ao lado do botão **Verificar** e representado por uma flecha apontada para a direita (figura 4), para que ocorra a transferência do programa à memória da placa de Arduino. Durante o carregamento, aparecerá, no canto inferior direito da janela, uma barra de progressão, demonstrada pela figura 6.

Figura 6
Barra de progressão da transferência do código-fonte à placa Arduino



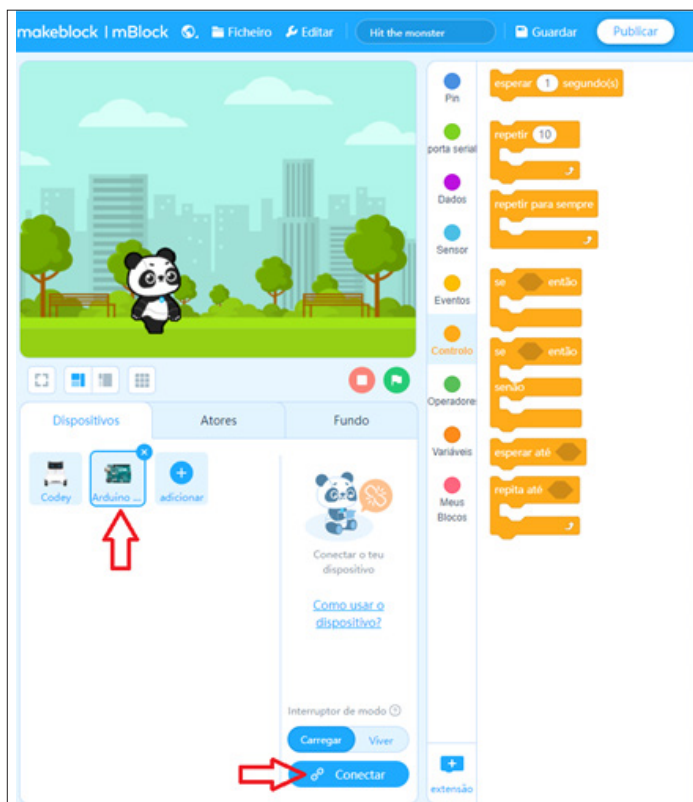
Fonte: site Arduino oficial

Finalizada a transferência do código-fonte, o Arduino Uno executará o programa Blink, fazendo com que o LED integrado à placa de prototipagem e próximo a porta digital 13, fique piscando com intervalos de 1 em 1 segundo.

ii. Linguagem de programação por blocos

Além da programação por codificação, podemos determinar o piscar de um LED, em intervalo de 1 segundo, por meio da programação em blocos. Para tal, conecte a placa Arduino Uno, através do cabo USB, ao computador ou dispositivo móvel. Na sequência, acesse o software mBlock e clique, no campo **Dispositivos**, sobre o ícone Arduino Uno para iniciar a programação em blocos, como mostra a figura 7.

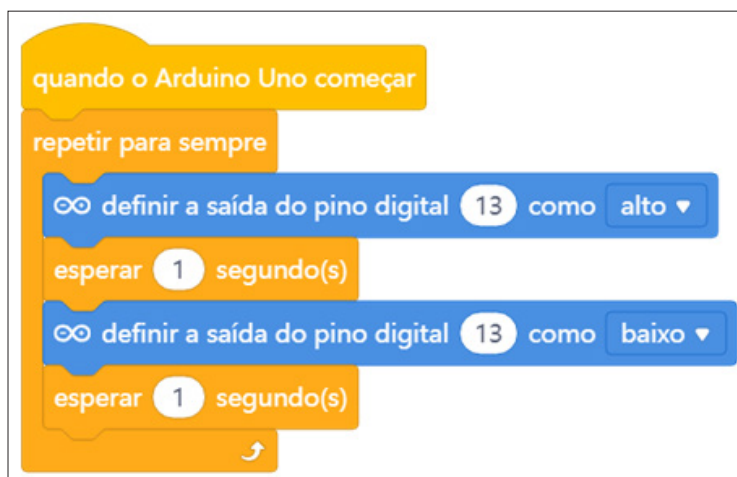
Figura 7 - Conectando Arduino Uno ao mBlock



Fonte: site mBlock oficial

A seguir, monte os blocos, arrastando e soltando, de acordo com a programação para piscar o LED, conforme mostra a figura 8.

Figura 8 - Programação em blocos



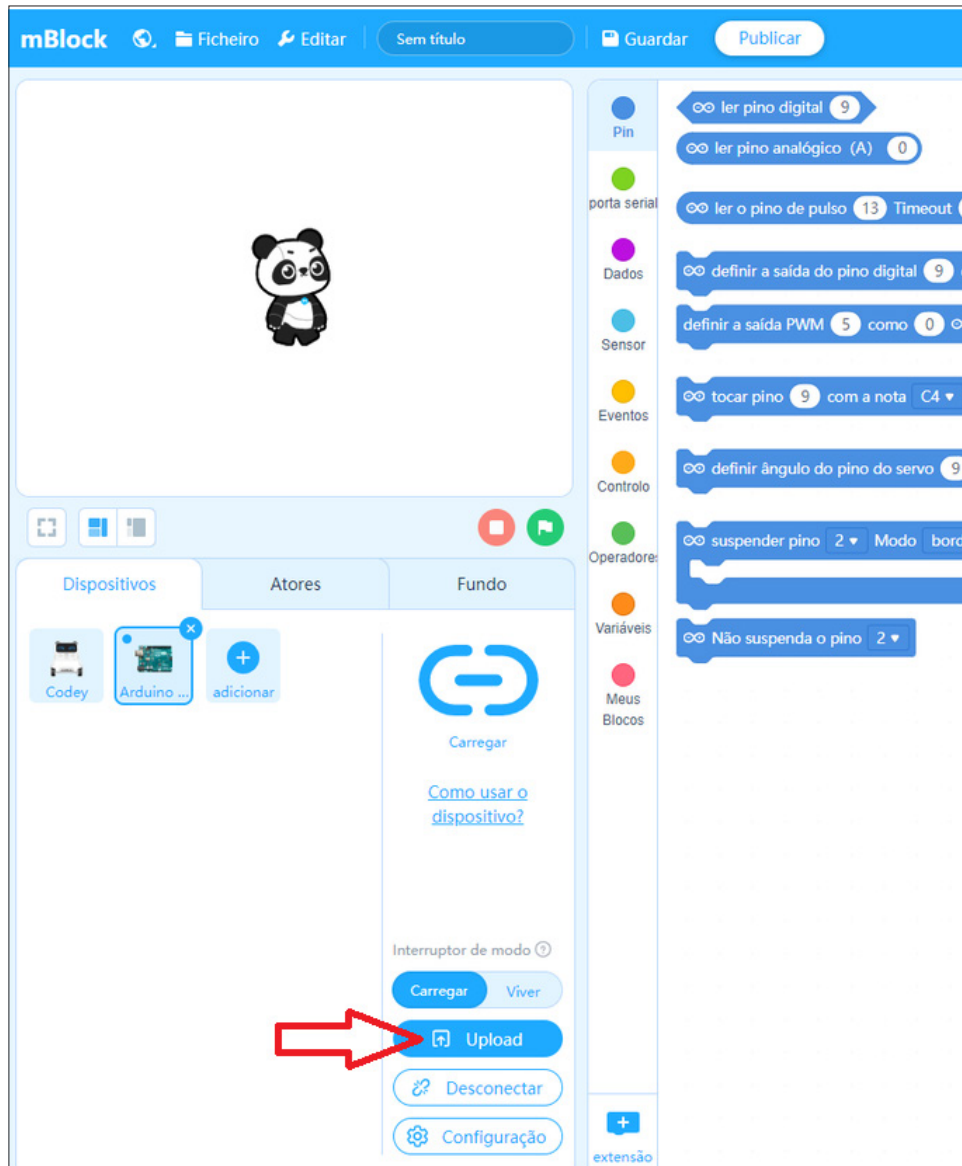
Assim que os blocos estiverem montados, clique no botão **Conectar** (figura 7) para transferir os comandos criados no mBlock para o Arduino Uno. Ao clicar neste botão, abrirá um Tooltip solicitando a confirmação da conexão entre os dois dispositivos, conforme mostra a figura 9.

Figura 9
Confirmar conexão entre mBlock e Arduino Uno



Na sequência, clique no botão **Upload** (figura 10) para compilar os blocos e carregar a programação para o Arduino. Após a transferência dos dados, as funções programadas serão executadas, ou seja, o LED irá piscar em intervalos de 1 em 1 segundo.

Figura 10 - Botão Upload para compilação da programação em blocos



Fonte: site mBlock oficial



Desafio:

Que tal alterar o ritmo que o LED pisca? Vamos lá, crie seu próprio ritmo!



E se... ?

Após as etapas da programação o LED não acender, verifique no código de programação se a porta informada é a 13 (ou LED_BUILTIN). Verifique se a programação foi transferida com sucesso através do console de feedback, caso apresente erros, confira se foi especificado corretamente o modelo do Arduino e a porta em que ele está conectado.

3. Feedback e Finalização (15min):

a. Você e seus colegas compreenderam a lógica do funcionamento das portas digitais do Arduino? Compartilhem suas impressões e as novas ideias, pontuando em seu caderno em forma de esquema.

b. Reúna todos os componentes utilizados nesta aula e os organize novamente, junto aos demais, no kit de robótica.



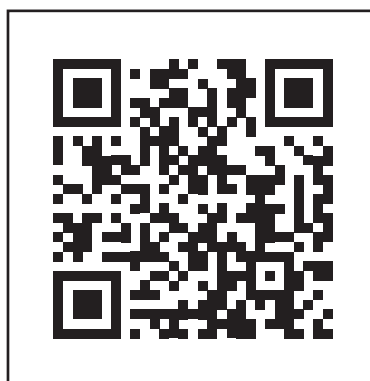
Videotutorial

Com o intuito de auxiliar na montagem e na programação desta aula, apresentamos um videotutorial, disponível em:



<https://rebrand.ly/aórobotica>

Acesse, também, pelo QRCode:





Referências

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular**. Brasília, 2018. Disponível em: http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf. Acesso em: 12 out. 2021.

MAKEBLOCK. mBlock. **Download mBlock**. Disponível em: <https://mblock.makeblock.com/en-us/download/>. Acesso em: 15 out. 2021.

MAKEBLOCK. mBlock. **Programação em blocos**. Disponível em: <https://ide.mblock.cc/>. Acesso em: 21 out. 2021.

DIRETORIA DE TECNOLOGIAS E INOVAÇÃO (DTI)
COORDENAÇÃO DE TECNOLOGIAS EDUCACIONAIS (CTE)

EQUIPE ROBÓTICA PARANÁ

Adilson Carlos Batista
Cleiton Rosa
Darice Alessandra Deckmann Zanardini
Edna do Rocio Becker
Marcelo Gasparin
Michelle dos Santos
Ricardo Hasper
Simone Sinara de Souza

Os materiais, aulas e projetos da “Robótica Paraná”, foram produzidos pela Coordenação de Tecnologias Educacionais (CTE), da Diretoria de Tecnologia e Inovação (DTI), da Secretaria de Estado da Educação e do Esporte do Paraná (Seed), com o objetivo de subsidiar as práticas docentes com os estudantes por meio da Robótica.

Este material foi produzido para uso didático-pedagógico exclusivo em sala de aula.



Este trabalho está licenciado com uma Licença
Creative Commons – CC BY-NC-SA
[Atribuição - NãoComercial - Compartilhual 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

