Arduíno - Aula 2

Entrada e saída de dados (Pin)

Grupo de Robótica do IFRS Campus Restinga - 2019

Iniciando o Arduino

Tudo se inicia no código:
void setup()
{
… [código] ...
}
Dentro das chaves fica todo o código que será executado ao se ligar o arduino.

Usando O Console

Em muitos momentos você ira precisar saber oque esta sendo gravado ou alterado nas variaveis ou mesmo passar informações para o console, para fazer isso usamos os comandos ***serial.print*** e ***serial.println.***

O codigo é usado da seguinte forma:
serial.print(“Mensagem aqui”)

Usando O Console

Tudo que for marcado por aspas duplas “”, será mostrado como somente uma mensagem. Também podemos ler informações de uma variável para isso basta escrever o nome da variável sem as aspas:

serial.print(variavel)

No caso acima, ele irá mostrar no console o valor da variável.

Usando O Console

Temos o codigo serial.print e o serial.println.

serial.print apenas retorna no console.
serial.println retorna no console e logo depois quebra a linha.

Iniciando o Arduino

Tudo se inicia no código:

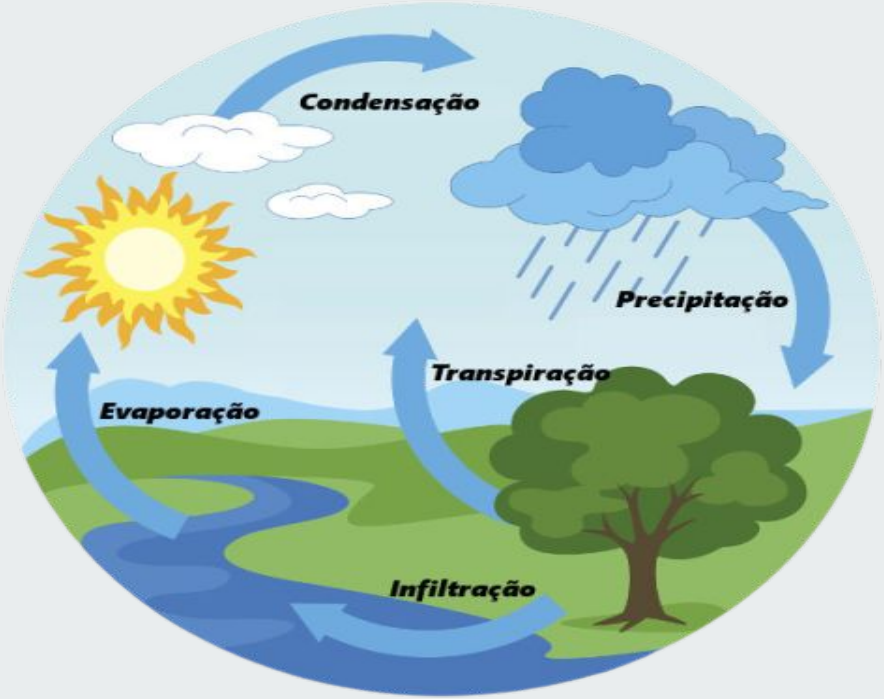
```
void setup()
{
  Serial.begin(9600); //necessário para estabelecer a comunicação com o console
  Serial.println("ligando");
}
```

O arduino executa somente uma vez o que está no setup()

Loop

Loops são conjuntos de ações cíclicas, que se repetem, que ocorrem tanto no mundo real quanto na programação.

Loop


Condensação
Precipitação
Transpiração
Evaporação
Infiltração

Loop

No Arduino nós iniciamos um loop pelo código:
void loop
(
… [código] ….
)

Dentro estará os comandos que serão executados enquanto o Arduino estiver ligado.

Loop

No Arduino nós iniciamos um loop pelo código:
void loop
(
*saída de dados simples
)

Dentro estará os comandos que serão executados enquanto o Arduino estiver ligado.

Criando variáveis

Podemos criar vários tipos de variáveis, cada uma para um respectivo tipo, abaixo terá o código de cada uma delas.

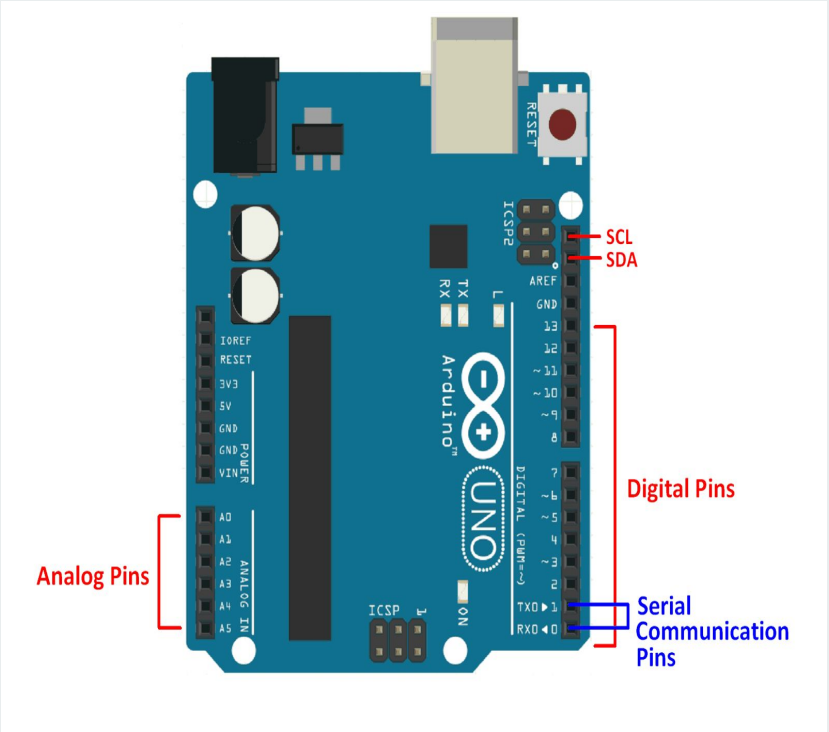
- **char**, são utilizados para armazenar caracteres e ocupam um byte.
- **byte**, podem armazenar um número entre 0 e 255.
- **int**, ocupam 2 bytes (16 bits) e tanto armazenam um número entre 2^{-15} e 2^{15}-1, ou seja, entre -32,768 e 32,767.
- **unsigned int**, também ocupam 2 bytes, mas como não possuem sinal, podem tomar valores entre 0 e 2^{16}-1, ou seja, entre 0 e 65,535.
- **long**, ocupa 32 bits (4 bytes), de -2,147,483,648 até 2,147,483,647.
- **float**, números decimais que ocupam 32 bits (4 bytes). Podem tomar valores entre -3.4028235E+38 e +3.4028235E+38.
- **double**, também armazena números decimais, mas possuem 8-bytes (64 bit).

Criando um cronômetro

Usando loop e serial.print, crie um cronômetro simples que conta os segundos, use outros códigos caso precise.

O que são Pins

Pins são as entradas e saídas digitais, sendo numeradas normalmente de 0 a 13.
Os pins devem ser configurados de modo que atuem como entradas ou saídas de dados.



Tipos de pins

Pinos digitais só podem ser atribuídos ou lidos dois valores que normalmente são usados HIGH e LOW.

Pinos analogicos podem receber diferentes valores como de 0 a 1023.

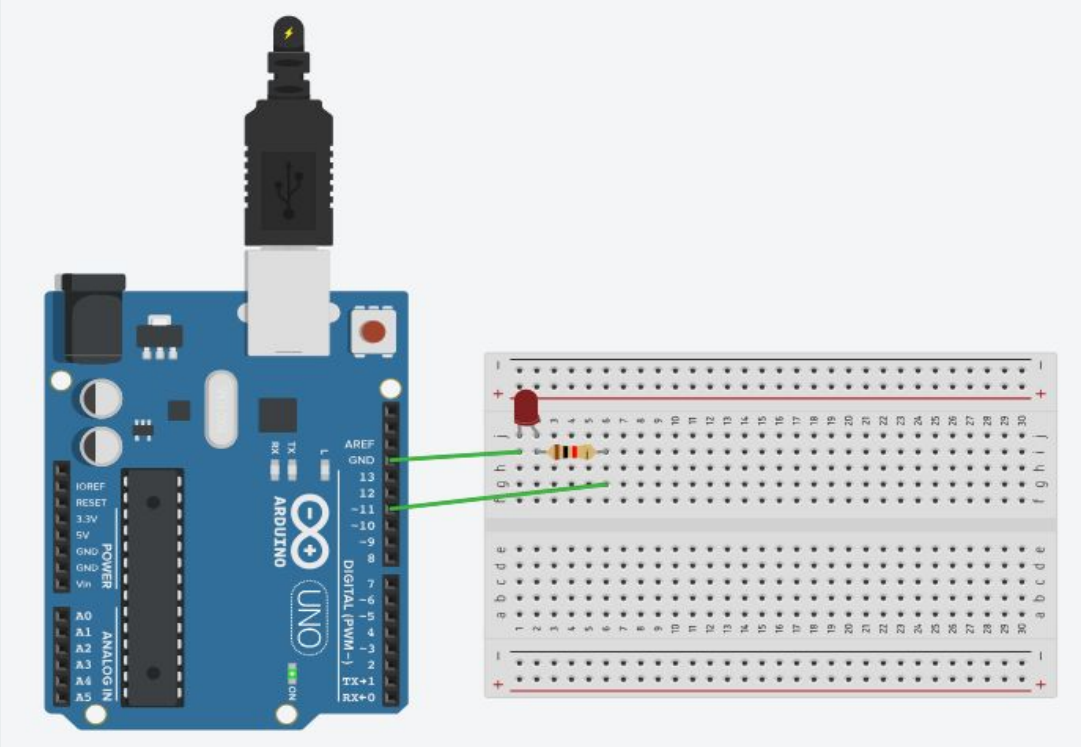
Exemplo
Como ligar um LED

Ligando um LED

Montagem

Nesse exemplo usaremos:

- 1 arduino UNO;
- 1 resistor de 100 OHMs;
- 1 led;
- 1 protoboard;
- fios jumper.



Escolhendo o pino

Para determinar o pino que será utilizado usamos o seguinte código:

pinMode(Número do pino, Entrada/Saída)

Exemplo:
pinMode(13, IN) //exemplo de entrada no pino #13
pinMode(10, OUT) //exemplo de saída no pino #10

Dando valores a o pino

Para dar algum valor para o pino é usado:

digitalWrite(Número do pino, valor);

Exemplo:
digitalWrite(10, LOW);

Ligando um LED

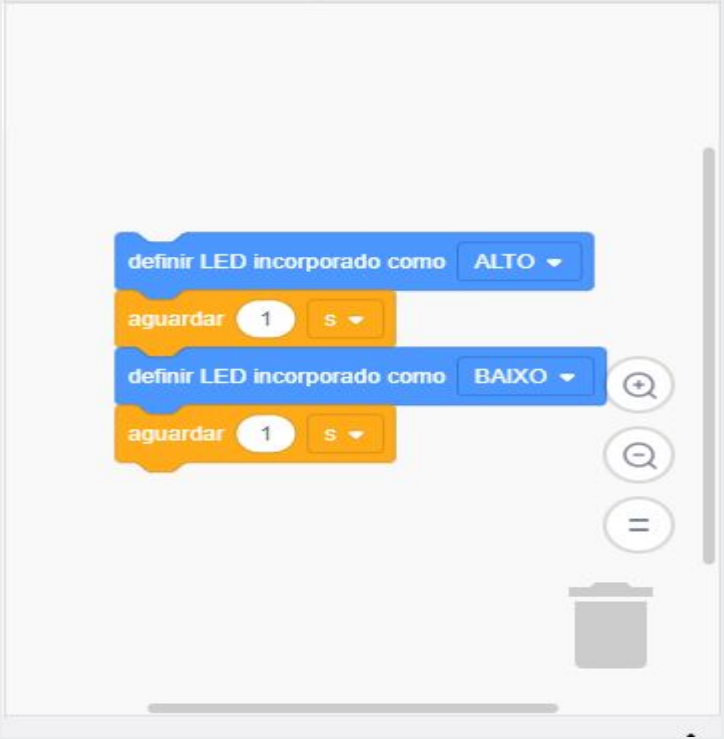
Programação em código:

```
void setup() // Inicia o Arduino
{
 pinMode(13, OUTPUT); // Escolhe o pino como saída
}

void loop() // Executa sempre
{
 digitalWrite(13, HIGH); // Da o valor HIGH para o pino
 delay(1000); // Espera 1000 milissegundos
 digitalWrite(13, LOW); // Da o valor LOW para o pino
 delay(1000); // Espera 1000 milissegundos
}
```

Ligando um LED

Programação em blocos



Usando DigitalRead

O comando digitalRead lê o que está gravado para poder usar posteriormente exemplo:

Aciona o pino 13 para o mesmo valor que o pino 7, declarado como entrada.

```
int ledPin = 13;  // LED conectado ao pino digital 13
int inPin = 7;    // botão conectado ao pino digital 7
int val = 0;      // variável para guardar o valor lido

void setup() {
  pinMode(ledPin, OUTPUT);    // configura o pino digital 13 como saída
  pinMode(inPin, INPUT);      // configura o pino digital 7 como entrada
}

void loop() {
  val = digitalRead(inPin);   // lê o pino de entrada
  digitalWrite(ledPin, val);  // aciona o LED com o valor lido do botão
}
```

Tente você

Adicione mais um led no sistema de exemplo apresentado anteriormente, faça com que um ligue e outro desligue de forma alternada a cada 1 segundo.

Tente você

Tente criar um semáforo com 3 leds de cores: Vermelha, Amarela e Verde. Faça as cores trocarem na ordem de Vermelho para amarelo e depois verde.

Tente você

- Digitalread - mostrar no console os valores (serial.println)
- Analogread - mostrar no console os valores (serial.println)
- AnalogWrite - motor mexer mais rápido, força da luz do led

Fim da Aula!!

Obrigado!!