

Objetivo

- Realizar comunicação serial entre o Arduino e um computador

Exercícios

- Realize a montagem do circuito e utilize o código abaixo na IDE do Arduino para implementar o controle de iluminação. Conecte o Arduino ao computador, execute o projeto e observe o monitor serial na IDE.

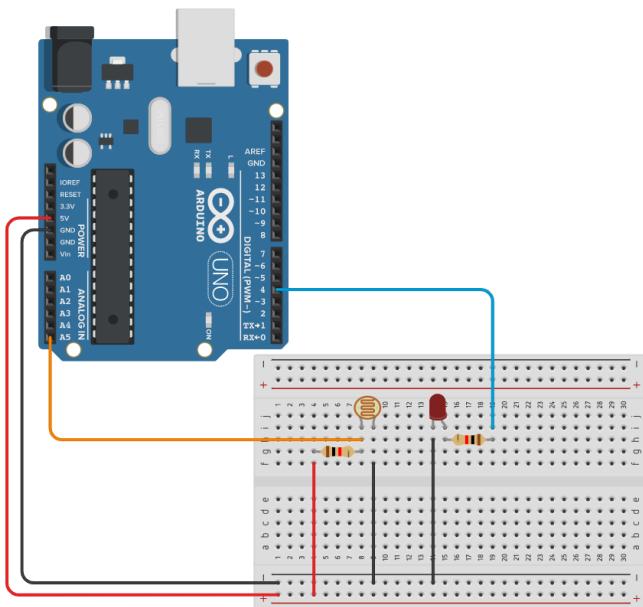


Figura 01 – Circuito e código da montagem do controle de iluminação

- Crie uma pasta no computador e execute o comando abaixo:

- `npm install serialport`

A biblioteca serialport fornece uma interface para comunicação serial assíncrona. Essa biblioteca é normalmente usada em projetos que envolvem a comunicação entre dispositivos eletrônicos via portas seriais.

- Crie um arquivo `comunicacao.js` e copie o código abaixo:

```
const { SerialPort } = require('serialport');
const { ReadlineParser } = require('@serialport/parser-readline');
const port = new SerialPort({ path: 'COM3', baudRate: 9600 });
const parser = port.pipe(new ReadlineParser({ delimiter: '\n' }))

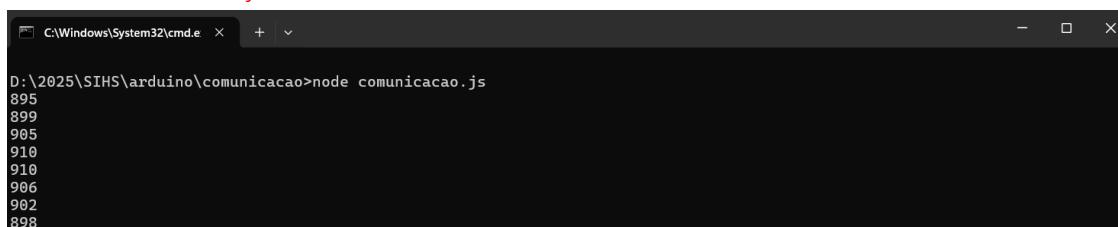
parser.on("data", function (leitura) {
  console.log(leitura);
})
```

Observações:

- O código acima foi adaptado da documentação oficial da biblioteca SerialPort <https://serialport.io/docs/api-parser-readline>
- A porta serial (`COM3`) deve ser configurada conforme a conexão do computador utilizado.

- Execute o programa criado no arquivo `comunicacao.js` com o comando abaixo:

- `node comunicacao.js`



- Os valores impressos correspondem à leitura do sensor de luminosidade LDR (*Light Dependent Resistor*).
- ATENÇÃO:** o Monitor Serial na IDE do Arduino deve estar fechado.

5. Execute o comando abaixo na pasta do arquivo **comunicacao.js**:

- `npm install axios`

A biblioteca Axios é utilizada para realizar requisições HTTP. Ela oferece uma interface para fazer solicitações HTTP para servidores web.

6. Faça as **modificações** no código do **comunicacao.js**:

```
const { SerialPort } = require('serialport');
const { ReadlineParser } = require('@serialport/parser-readline');
const port = new SerialPort({ path: 'COM3', baudRate: 9600 });
const parser = port.pipe(new ReadlineParser({ delimiter: '\n' }));

const axios = require('axios'); ←

parser.on("data", function (leitura) {
  let valor = leitura.trim();
  axios.post('http://163.176.234.128/arduino/sensor.php', {
    valor: valor,
    identificacao: "Odilon" ←
  })
  .then((retorno) => {
    console.log('Resposta do servidor: ', retorno.data);
  })
  .catch((erro) => {
    console.error('Erro na requisição: ', erro);
  });
  console.log(leitura);
});
```

Coloque os nomes dos alunos que fizeram o Guia

7. Execute novamente o programa **comunicacao.js**:

- `node comunicacao.js`

```
C:\Windows\System32\cmd.exe x + v

D:\2025\SIHS\arduino\comunicacao>node comunicacao.js
901
Resposta do servidor: { 'Dados armazenados com sucesso. ID gerado: ': '32' }
901
Resposta do servidor: { 'Dados armazenados com sucesso. ID gerado: ': '33' }
902
Resposta do servidor: { 'Dados armazenados com sucesso. ID gerado: ': '34' }
902
Resposta do servidor: { 'Dados armazenados com sucesso. ID gerado: ': '35' }
902
Resposta do servidor: { 'Dados armazenados com sucesso. ID gerado: ': '36' }
902
```

8. Acesse o endereço abaixo para verificar os dados enviados.

- <http://163.176.234.128/arduino>

Identificação	Valor	IP	Data do registro
Odilon	895	201.182.50.123	25-11-2025 02:33:16
Odilon	896	201.182.50.123	25-11-2025 02:33:17
Odilon	896	201.182.50.123	25-11-2025 02:33:18
Odilon	897	201.182.50.123	25-11-2025 02:33:19
Odilon	897	201.182.50.123	25-11-2025 02:33:20
Odilon	896	201.182.50.123	25-11-2025 02:33:21
Odilon	896	201.182.50.123	25-11-2025 02:33:22
Odilon	895	201.182.50.123	25-11-2025 02:33:23
Odilon	895	201.182.50.123	25-11-2025 02:33:24
Odilon	895	201.182.50.123	25-11-2025 02:33:25

Mostrando de 1 até 10 de 11 registros