

# Projeto de Comunicação de Dados

Ana Carolina Ribeiro Miranda  
Bruno Uhlmann Marcato  
Darlan de Oliveira  
Guilherme Henrique Soeiro Fontes  
Thomas Oliveira Rocha Sampaio Silva

Prof: Luiz Fernando Carvalho

Data de entrega: 12/12/2022

---

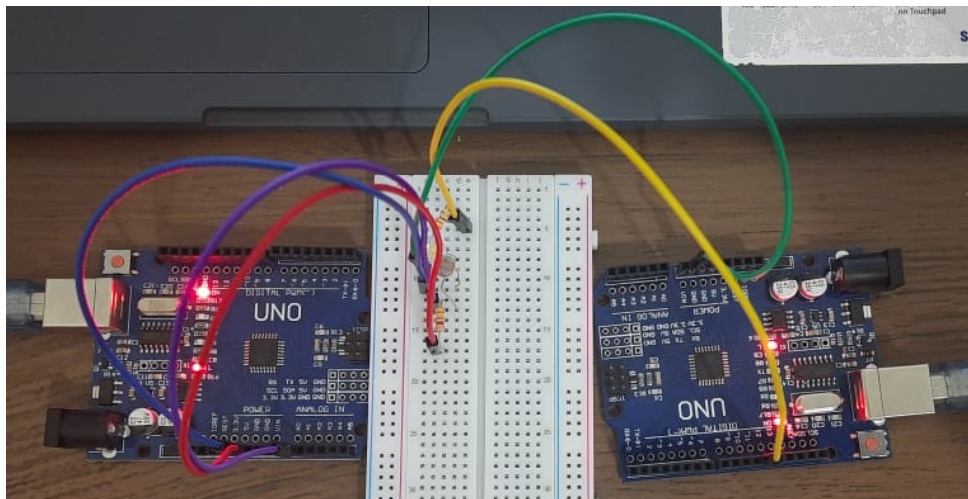
## 1. Objetivo

Utilizar os conhecimentos aprendidos na disciplina para realizar a comunicação entre dois dispositivos por meio de sinais luminosos.

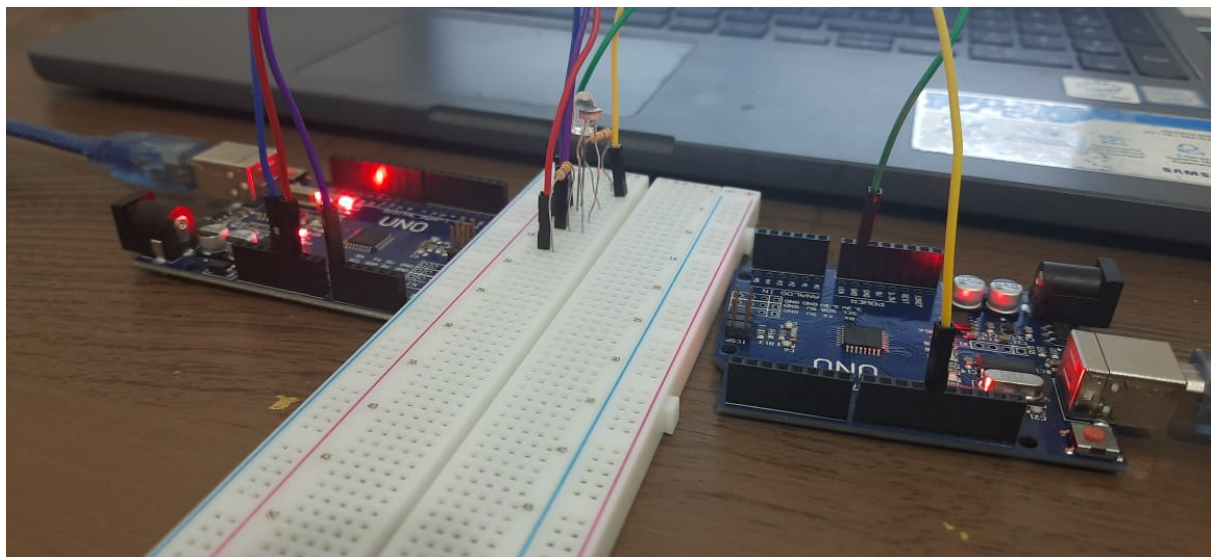
---

## 2. Materiais

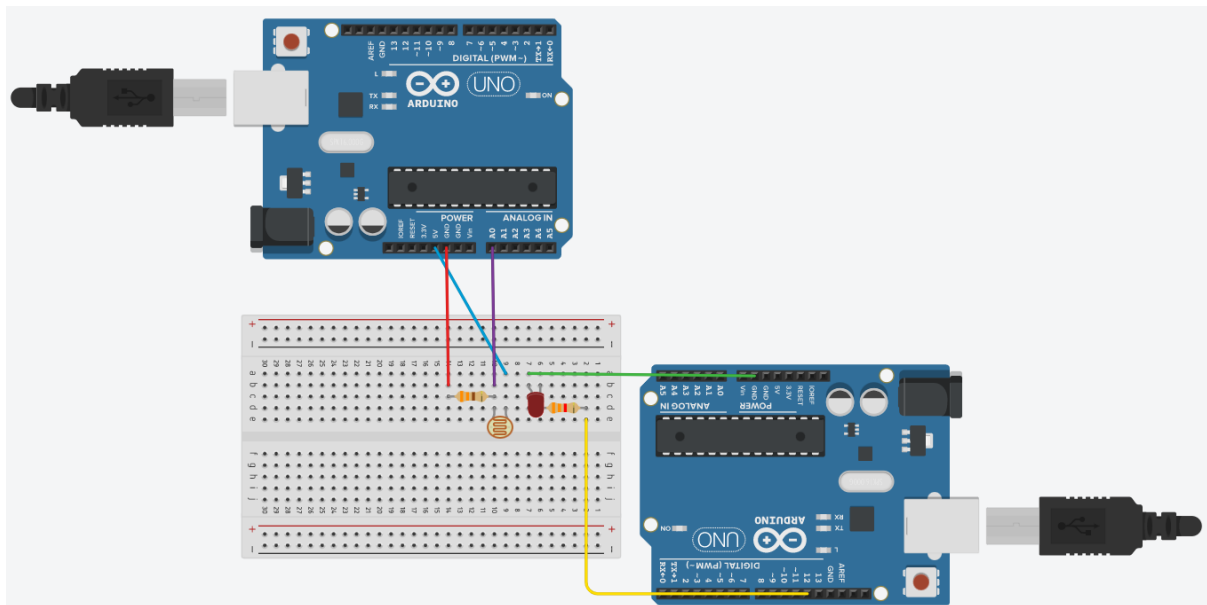
- 1 Protoboard
- 5 Jumpers
- 1 LED
- 2 Arduinos UNO
- 1 Resistor de 300  $\Omega$
- 1 Resistor de 10 k $\Omega$
- 1 Fotorresistor



**Figura 1 - Esquema montado.**  
**Fonte: Autoria Própria.**



**Figura 2 - Esquema montado.**  
**Fonte: Autoria Própria.**



**Figura 3 - Esquema representado no TinkerCad.**

**Fonte: Autoria Própria.**

### 3. Desenvolvimento

Neste trabalho foi proposto o desenvolvimento de uma interface capaz de enviar e receber dados, utilizando os conhecimentos obtidos em aula, onde será elaborado o desenvolvimento de um sistema capaz de fazer transmissão, utilizando os métodos de codificação de sinal NRZ-L e NRZ-I. Os códigos foram realizados em conjunto via reuniões de maneira presencial e remotas, onde todos os membros debatiam as ideias propostas.

Para a realização deste projeto, utilizamos dois Arduino Uno, onde sua ideia central é que seja realizada a transmissão de uma palavra, fornecida pelo usuário, de um equipamento para outro, realizada essa comunicação por sinais luminosos.

Um dos Arduinos é responsável por codificar a mensagem e enviar os sinais luminosos correspondentes à palavra, já o outro equipamento é responsável por receber os sinais, decodificá-los e mostrar a palavra recebida.

Para garantir que a transmissão foi bem sucedida, um detector de erros baseado no CRC (Cyclic Redundance Check) foi implementado nos dois códigos.

O receptor verifica o checksum a procura de incoerências, caso haja, a informação não é decodificada e é exibida na tela "fragmento corrompido", caso não haja, será decodificado, mostrando a mensagem enviada. O emissor recebe a informação e a converte para binário adicionando um checksum, para análise de possíveis erros na transmissão e então a envia para o emissor, além de mandar sinais para o LED informando que a comunicação está em andamento, logo, o LED acende de acordo com as informações transmitidas e é apagado representando o fim da transmissão.

O microcontrolador é o responsável por emitir o sinal e receber a mensagem, a palavra a ser enviada pelo Serial. Após receber essa palavra, o microcontrolador a

separa por caracteres e, para cada caractere, extrai o seu código ASCII, convertendo esse código para um binário, calcula o código CRC deste binário com um código de 4 bits (1101). Por fim, o binário e o código CRC são juntados em um binário de 11 bits e são transmitidos pelo microcontrolador utilizando a codificação NRZ-L ou NRZ-I.

A função para transformar o código ASCII em um binário é um simples código de transformação que utiliza o resto da divisão por 2, que no final, preenche os bits restantes com zero. O CRC é calculado utilizando a função XOR entre o binário do caractere e o código CRC. A operação é feita bit a bit, onde o primeiro laço de repetição itera o número binário (do primeiro ao sétimo bit) e o segundo laço itera o código CRC. Dessa forma, de 4 em 4 bits, todas as operações XOR são realizadas entre o número binário e o código CRC. Ao fim, os últimos 3 elementos do vetor resultante contém o código que será adicionado para a transmissão.

A função que codifica a transmissão em NRZ-L acende o led por 500 ms toda vez que um bit 0 for fornecido, e apaga o led por 500 ms quando um 1 for fornecido. Já a função NRZ-I, no primeiro bit transmitido faz como a função NRZ-L (acende o led em 1 e apaga em 0), mas, posteriormente, quando um bit 0 é fornecido o led mantém o seu estado anterior por mais 500 ms e se um bit 1 é fornecido o estado do led é invertido. Na hora da transmissão, o emissor avisa um novo fragmento, caractere que será transmitido, mantendo o led aceso por dois ciclos de 500 ms antes de enviar os dados do fragmento.

O receptor, verifica o valor do LDR a cada 500 ms. Quanto percebe dois valores de led aceso consecutivos, começa a capturar os dados do fragmento que será recebido. Ao receber os 11 bits de um fragmento, ele guarda o valor em um vetor de fragmentos. Se perceber que o emissor irá enviar outro fragmento, captura esses dados e adiciona ao vetor. Ao perceber que o receptor não tem mais dados para enviar nessa transmissão, para cada fragmento no vetor, decodifica a transmissão, seja ela NRZ-L ou NRZ-I, fazendo o processo inverso do emissor e verifica se os bits chegaram corretamente utilizando o código CRC, fazendo a operação XOR entre os 11 bits recebidos e o código CRC (1101). Se todos os bits resultantes dessa operação forem 0, o receptor pega os 8 bits do caractere, transforma para decimal e printa o caractere correspondente ao código ASCII. Se os bits não forem todos zeros, o receptor mostra imprimindo no serial como fragmento corrompido e parte para o próximo fragmento.

---

## 4. Casos Testes

Foram realizados diversos testes de transmissões, entre elas da palavra "Olá", "Ana", também palavras com acentuação como "paçoca" ou "pàçocâ" e até caracteres como "α β".

Mais detalhes sobre o funcionamento podem ser vistos no código, através do seguinte link Github: <https://github.com/Aninha33/Comunica-o-de-Dados.git>.

Para o exemplo do “Ana”, o emissor deveria gerar e enviar o seguinte sinal, utilizando a codificação NRZ-I:

Mensagem a ser transmitida:

Ana

Codificação binária:

01000001001

01101110100

01100001011

Transmitindo...

1	-1	-1	-1	-1	-1	-1	1	1	1	-1
---	----	----	----	----	----	----	---	---	---	----

1	-1	1	1	-1	1	-1	-1	1	1	1
---	----	---	---	----	---	----	----	---	---	---

1	-1	1	1	1	1	1	-1	-1	1	-1
---	----	---	---	---	---	---	----	----	---	----

Transmissão finalizada

Por fim, o binário e o código CRC são juntados em um binário de 11 bits e são transmitidos pelo microcontrolador utilizando a codificação NRZ-L ou NRZ-I. Para o esquema anterior, receptor mostrará o seguinte:

Transmissão sendo recebida...

1	-1	-1	-1	-1	-1	-1	1	1	1	-1
---	----	----	----	----	----	----	---	---	---	----

1	-1	1	1	-1	1	-1	-1	1	1	1
---	----	---	---	----	---	----	----	---	---	---

1	-1	1	1	1	1	1	-1	-1	1	-1
---	----	---	---	---	---	---	----	----	---	----

Mensagem recebida:

Ana

Para o exemplo do “pãçôcâ”, no NRZ-L, o emissor enviar o seguinte sinal:

Transmissão finalizada  
 Mensagem a ser transmitida:  
 pççcâ

Codificação binária:

01110000111  
 11000011011  
 10100001010  
 11000011011  
 10100111110  
 11000011011  
 10110110010  
 01100011100  
 11000011011  
 10100010000

Transmitindo...

1	-1	-1	-1	1	1	1	1	-1	-1	-1
-1	-1	1	1	1	1	-1	-1	1	-1	-1
-1	1	-1	1	1	1	1	-1	1	-1	1
-1	-1	1	1	1	1	-1	-1	1	-1	-1
-1	1	-1	1	1	-1	-1	-1	-1	-1	1
-1	-1	1	1	1	1	-1	-1	1	-1	-1
-1	1	-1	-1	1	-1	-1	1	1	-1	1
1	-1	-1	1	1	1	-1	-1	-1	1	1
-1	-1	1	1	1	1	-1	-1	1	-1	-1
-1	1	-1	1	1	1	-1	1	1	1	1

Transmissão finalizada

O receptor mostrará o seguinte esquema:

Transmissão sendo recebida...

1	-1	-1	-1	1	1	1	1	-1	-1	-1
-1	-1	1	1	1	1	-1	-1	1	-1	-1
-1	1	-1	1	1	1	1	-1	1	-1	1
-1	-1	1	1	1	1	-1	-1	1	-1	-1
-1	1	-1	1	1	-1	-1	-1	-1	-1	1
-1	-1	1	1	1	1	-1	-1	1	-1	-1
-1	1	-1	-1	1	-1	-1	1	1	-1	1
1	-1	-1	1	1	1	-1	-1	-1	1	1
-1	-1	1	1	1	1	-1	-1	1	-1	-1
-1	1	-1	1	1	1	-1	1	1	1	1

Mensagem recebida:

pâççcâ

Para o exemplo do “ç ß”, o emissor irá gerar e enviar o seguinte sinal, utilizando a codificação NRZ-I:

Mensagem a ser transmitida:



Codificação binária:

```
11100000011
10110110010
10011110110
11110000010
10010011110
10000110001
10001111010
11000011011
10011111011
```

Transmitindo...

-1	1	-1	-1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	1	-1	-1	-1	1	1
-1	-1	-1	1	-1	1	-1	-1	1	-1	-1
-1	1	-1	1	1	1	1	1	1	-1	-1
-1	-1	-1	1	1	1	-1	1	-1	1	1
-1	-1	-1	-1	-1	1	-1	-1	-1	-1	1
-1	-1	-1	-1	1	-1	1	-1	-1	1	1
-1	1	1	1	1	1	-1	1	1	-1	1
-1	-1	-1	1	-1	1	-1	1	1	-1	1

Transmissão finalizada

O receptor mostrará o seguinte esquema:

Transmissão sendo recebida...

-1	1	-1	-1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	1	-1	-1	-1	1	1
-1	-1	-1	1	-1	1	-1	-1	1	-1	-1
-1	1	-1	1	1	1	1	1	1	-1	-1
-1	-1	-1	1	1	1	-1	1	-1	1	1
-1	-1	-1	-1	-1	1	-1	-1	-1	-1	1
-1	-1	-1	-1	1	-1	1	-1	-1	1	1
-1	1	1	1	1	1	-1	1	1	-1	1
-1	-1	-1	1	-1	1	-1	1	1	-1	1

Mensagem recebida:



## 5. Pontos Positivos e Negativos

O projeto atua corretamente, exceto em raras ocasiões em que houve dessincronização entre o emissor e o receptor, consequentemente os valores não foram captados corretamente. Para solucionar este problema basta iniciar uma nova transmissão após o término da que apresentou erro.

Obteve-se com sucesso um emissor que codifica que envia mensagens utilizando os padrões de codificação Level e Invert e um receptor que decodifica o padrão de codificação Level.

---

## 6. Links e Referências

### Interrupção no Arduino

- <https://www.arduino.cc/reference/en/libraries/timerinterrupt/>

### TinkerCad:

- <https://www.tinkercad.com/>

### Programação Arduino e ESP

- [https://www.youtube.com/watch?v=oE8qL\\_3WHyo&list=PLMmiQibT0iT0OU RiNLnHkkv2\\_WkPV7NS](https://www.youtube.com/watch?v=oE8qL_3WHyo&list=PLMmiQibT0iT0OU RiNLnHkkv2_WkPV7NS)

### CRC

- slides de aula
- <https://www.youtube.com/watch?v=wW5DMawWI6E>
- Forouzan, Behrouz. Comunicação de Dados e Redes de Computadores. 4ª Edição, Capítulo 10.4. (biblioteca física e digital da UTFPR)

### GitHub

- <https://github.com/Aninha33/Comunica-o-de-Dados.git>