

## **COMUNICAÇÃO ENTRE DOIS DISPOSITIVOS POR MEIO DE SINAIS LUMINOSOS**

Murilo Fontana Muniz

Nicolas de Paulo Romano

Murilo Vital

Bruno Correa Borges Silva

João Pedro Neigri Heleno

Cristian Andre Sanches

Luiz Fernando Carvalho

APUCARANA - PR

JUNHO/2024



---

## SUMÁRIO

### 1. RESUMO

### 2. INTRODUÇÃO

### 3. MATERIAIS E MÉTODOS

#### 3.1. MATERIAIS UTILIZADOS

#### 3.2. METODOLOGIA

### 4. RESULTADOS E DISCUSSÃO

### 5. CONSIDERAÇÕES FINAIS

### 6. REFERÊNCIAS



## 1. RESUMO

Este projeto tem como objetivo estabelecer a comunicação entre dois dispositivos Arduino Uno utilizando sinais luminosos para a transmissão e recepção de mensagens de até 8 caracteres. O transmissor converte a mensagem em sinais luminosos, que são recebidos e decodificados pelo receptor. As codificações NRZ-L e NRZ-I são implementadas, com a possibilidade de incluir outras técnicas de codificação como Manchester e AMI. Para garantir a integridade da transmissão, utiliza-se um algoritmo de detecção de erros baseado em CRC. A atividade compreende a elaboração de um relatório detalhado que descreve a modelagem, as decisões de projeto, dificuldades encontradas e testes realizados, além da apresentação do funcionamento do protótipo para avaliação.

## 2. INTRODUÇÃO

Uma rede de computadores é construída para enviar informações de um ponto a outro. Essas informações precisam ser convertidas em sinais digitais ou analógicos para a transmissão, sendo fundamental para garantir a integridade dos dados durante a transmissão e minimizar erros. No caso deste projeto, será utilizada a conversão de dados digitais em sinais digitais a partir dos conceitos de codificação de linha.

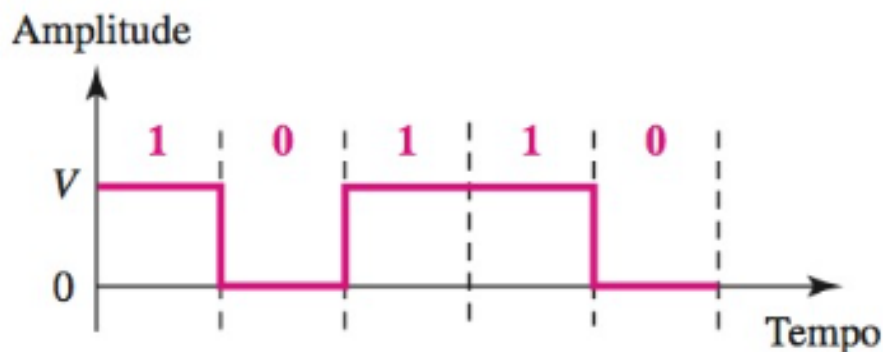
A codificação de linha converte uma sequência de bits em um sinal digital. No emissor, os dados digitais são codificados em um sinal digital; no receptor, os dados digitais são recriados, reconvertendo-se o sinal digital. Esse tópico é dividido em categorias, das quais serão abordadas somente duas, que são importantes para o entendimento deste projeto: unipolar e polar.

Primeiramente, a codificação de linha unipolar define que todos os níveis de sinal se encontram em um dos lados do eixo do tempo, acima ou abaixo dele. Dessa forma, essa categoria foi definida pelo método Non-Return-to-Zero (NRZ), ou seja, Sem



Retorno a Zero, no qual a voltagem positiva define o bit 1 e a voltagem zero define o bit 0. Ele é chamado NRZ porque o sinal não retorna a zero no meio do bit.

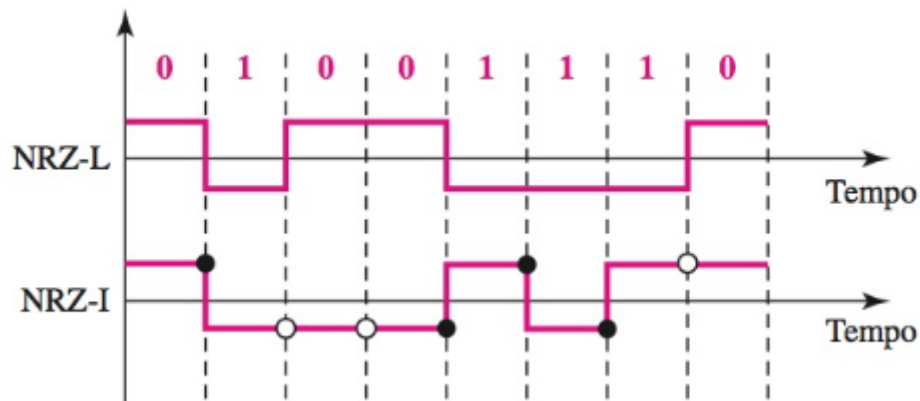
Figura 1: Representação unipolar



Fonte: Luiz F. Carvalho.

Em segundo lugar, a codificação de linha polar estabelece que as voltagens se encontram em ambos os lados do eixo de tempo, ou seja, o nível de voltagem para 0 pode ser positivo e o nível de voltagem para 1 pode ser negativo, ou vice versa. Assim como ao seu semelhante unipolar, esse método faz uso do NRZ, entretanto, adiciona ao NRZ duas versões. Em NRZ-L, o valor do bit é determinado pelo nível de voltagem. Já em NRZ-I, a alteração ou a ausência de alteração no nível de voltagem determina o valor do bit a ser transmitido. Esse método não modifica o sinal de saída ao enviar um bit "0" e inverte o sinal do estado anterior ao enviar um bit "1".

Figura 2: Representação polar



Fonte: referência Luiz F. Carvalho.

Comparando todos esses métodos, tem-se que o NRZ e NRZ-L possuem implementações simples e melhor eficiência de banda, entretanto possuem problemas de sincronização e de detecção de erros, que os fazem ter menos preferência na transmissão de dados. Já o NRZ-I, possui melhora na sincronização, mas tem mais complexidade em sua implementação.

Adicionalmente, vale entender conceitos de detecção de erro, principalmente CRC. O CRC, ou Verificação de Redundância Cíclica, é um método de detecção de erros utilizado para verificar a integridade dos dados transmitidos em redes de comunicação e sistemas de armazenamento. Nesse método, o transmissor gera uma sequência de  $(n - k)$  bits, chamada de frame check sequence (FCS), resultando em um frame de  $n$  bits, que é divisível por um número pré-determinado. O receptor, ao receber o frame, divide-o por esse número e, se não houver resto na divisão, assume que a mensagem não contém erros.

Todas essas concepções são importantes para a conclusão do projeto.

### 3. MATERIAIS E MÉTODOS

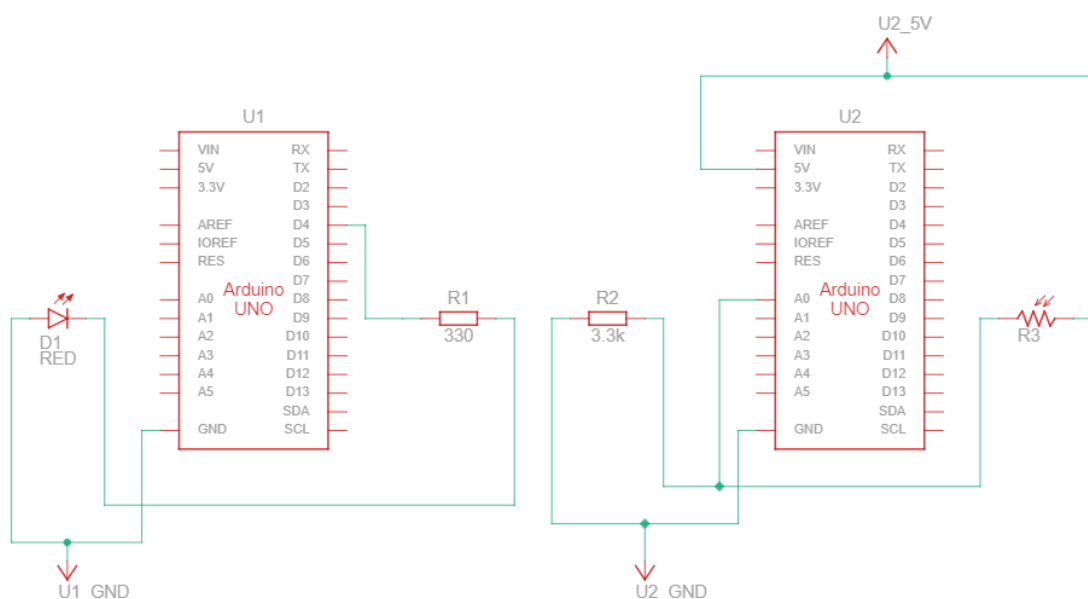
#### 3.1. MATERIAIS UTILIZADOS

- 1 Breadboard;
- 2 Arduinos UNO;
- 1 Resistor de 3,3 k $\Omega$ ;
- 1 Resistor de 330  $\Omega$ ;
- 1 LED;
- 1 Sensor LDR;
- Jumpers.

#### 3.2. METODOLOGIA

Este projeto foi realizado com dois Arduinos. Um foi dedicado ao envio das mensagens, enquanto o outro foi configurado para recebê-las, permitindo a comunicação entre ambos, conforme mostrado na Figura 3.

Figura 3: Circuito montado.



Fonte: Autoria própria.



No transmissor, o processo começa com a coleta dos dados fornecidos pelo usuário, seguido pelo processamento dessas informações e, finalmente, pela transmissão da mensagem. No receptor, as etapas envolvem a recepção da mensagem, a conversão dos dados recebidos e a exibição dos resultados.

O circuito do transmissor consiste em um Arduino, responsável por controlar todas as suas funções, um LED e um resistor de 330  $\Omega$ . O Arduino utiliza o pino de saída 4 para conectar-se ao resistor, que está ligado ao terminal positivo do LED, completando o circuito com o terminal negativo do LED conectado ao ground do Arduino.

Para a captura dos dados do usuário, é estabelecida uma comunicação serial entre o Arduino e um computador. O Arduino solicita que o usuário insira uma palavra de até 8 caracteres, que será enviada como a mensagem. Após a confirmação da entrada, o Arduino processa os dados, convertendo a palavra para binário e codificado conforme o método selecionado pelo usuário, que pode ser NRZ-L (Valores iguais ou invertidos) ou NRZ-I (Alto ou Baixo).

A seguir, é realizado o cálculo para a detecção de erros, o Cyclic Redundancy Check (CRC). Com isso, é concatenado um byte de zeros no final da mensagem, e é realizada uma divisão da mesma pelo polinômio CRC pré-determinado no emissor e receptor. Após, o resto da divisão é substituído pelos zeros adicionados no final da mensagem.

Por fim, é acrescentado dois bits iniciais a mensagem, para a tipagem correta do método de decodificação no receptor, ademais do CRC. A mensagem em seguida é transmitida, bit a bit, pelo LED do circuito, que utiliza sinais luminosos que fazem a alternância entre ligado (bit 1) e desligado (bit 0) conforme os bits que serão enviados.

Chegando ao circuito receptor, que faz a captação dos sinais luminosos com o uso do fotoresistor LDR, permite a distinção entre os sinais alto e baixo, fazendo a conversão dos sinais luminosos para elétricos, e em seguida para dados na



memória. Após isso, a verificação de erros CRC é realizada ignorando os dois primeiros bits que são correspondentes ao tipo de transmissão, com o intuito de verificar a integridade dos dados recebidos. Por se tratar do receptor, ele faz a divisão da mensagem pelo mesmo polinômio utilizado no emissor, esse resultado é a comprovação do sinal, se sofreu erro na transmissão, o resto da divisão será um valor diferente do zero, comprovando que houve erro na transmissão.

Adicionalmente, os dados são codificados usando a técnica NRZ-I que indica os bits pela mudança no sinal, e NRZ-L que utiliza o próprio nível do sinal para representar o bit.

Após a leitura e codificação, é exibida no monitor serial da IDE Arduino a palavra traduzida, ademais de seu tipo de codificação (NRZ-L ou NRZ-I), que foi avaliado pelo grupo, onde foi conferida se a saída no receptor era a mesma do emissor do sinal.

#### **4. RESULTADOS E DISCUSSÃO**

Com as etapas da obtenção de entrada do usuário, manipulação dos dados do arduino emissor, envio da mensagem, recepção dos dados pelo arduino receptor, e a codificação para exibir a saída, foi obtida uma das maneiras de como os dados são transmitidos no mundo real.





Figura 4: Monitor serial Transmissor após a execução NRZ-I Alto.

```
Senderino
81 void outputToLED(String codigo) {
82     digitalWrite(ledPin, LOW);
83     delay(50);
84     for (int i = 0; i < codigo.length(); i++) {
85         if (codigo.charAt(i) == '1') {
86             digitalWrite(ledPin, HIGH);
87         } else {
88             digitalWrite(ledPin, LOW);
89         }
90         delay(DELEY);
91     }
92 }
93
94
95
96 void setup() {
97     Serial.begin(9600);
98
99     digitalWrite(ledPin, HIGH);
100     Serial.println("Digite uma palavra, maximo 8 caracteres:");
101     while (Serial.available() == 0) {}
102     String palavra = Serial.readStringUntil('\n');
103     String binario = toBinary(palavra);
104
105     unsigned char crc;
106     String crcBin;
107
108     String nrzL_Normal = encodeNRZ_L_Normal(binario);

Output Serial Monitor x
Message (Enter to send message to 'Arduino Uno' on 'COM5') New Line 9600 baud
Palavra: adilton

Qual metodo gostaria de decodificar?
1) NRZ-L Valores iguais
2) NRZ-L Valores invertidos
3) NRZ-I Alto
4) NRZ-I Baixo

-I High:
CRC : 00010010
NRZ-I High: 1011111001000111101100011011011111010011110110101

Mensagem enviada
```

Fonte: Autoria própria.

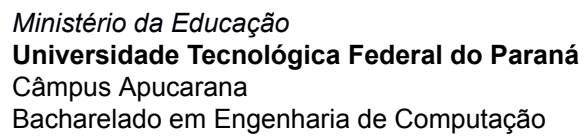


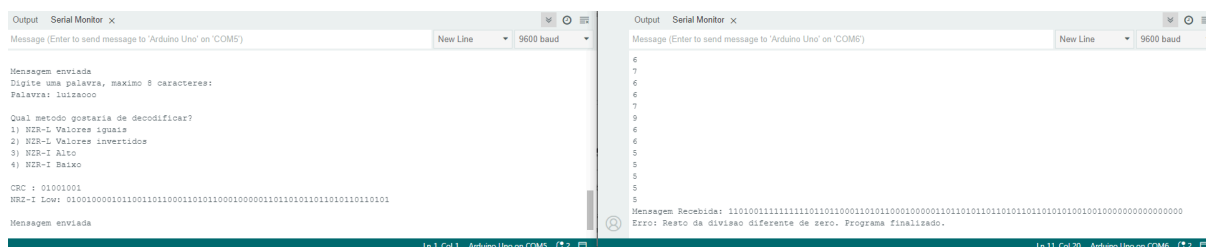
Figura 5: Monitor serial Receiver após a execução NRZ-I Alto.

```
Receiver.ino
113     if (decimal != 0) {
114         caractere += (char)decimal;
115     }
116 }
117 return caractere;
118 }
119
120
121 //receber dados
122 String recebeDados(){
123     String mensagem = "";
124     int sinal = 0;
125     delay(DELEY);
126     for(int indice = 0;indice < N;indice++){
127         sinal = analogRead(PIN_LDR);
128         Serial.println(sinal);
129         if(sinal > 60){
130             mensagem += "1";
131         } else {
132             mensagem += "0";
133         }
134         delay(DELEY);
135     }
136     return mensagem;
137 }
138
139 void loop() {
140     int sig = analogRead(PIN_LDR);
141     if(sig < THRESHOLD && enable_read == true){
```

Fonte: Autoria própria.



Figura 5: Monitor serial Transmissor e Receiver após a execução com interrupção.



Fonte: Autoria própria.

Conforme os testes desenvolvidos pelo grupo, foi obtido um resultado satisfatório na transmissão dos dados por meio de sinais luminosos. Como resultado, o grupo conseguiu perceber e interpretar as funcionalidades propostas, como a tradução de dados em alto nível (Strings e variáveis nominais) para 'vetores' binários, a validação de dados recebidos (com o método CRC), e principalmente a transmissão desses dados por meio da luminosidade (comunicação entre LED e LDR), de maneira satisfatória, levando em conta as limitações de tempo e conhecimento.

## 5. CONSIDERAÇÕES FINAIS

Este projeto mostrou a viabilidade e os desafios da comunicação entre dois dispositivos utilizando sinais luminosos. Utilizando técnicas de codificação NRZ-L e NRZ-I e um algoritmo de detecção de erros baseado em CRC, foi possível enviar e receber mensagens de até 8 caracteres com razoável precisão.

O projeto colocou em prática diversos conceitos teóricos aprendidos em campo, desde a codificação do conhecimento até a verificação da integridade e a consolidação do conhecimento adquirido em sala de aula. A montagem do protótipo utilizando Arduinos, LEDs e LDRs mostrou-se eficaz na transmissão de informações através de sinais luminosos, criando uma base sólida para futuras pesquisas e melhorias.

A análise das limitações e características da codificação NRZ-L e NRZ-I enfatizou a importância de escolher a técnica de codificação correta para diferentes



cenários de comunicação. Apesar de algumas limitações, essas técnicas ainda são amplamente utilizadas devido à sua eficiência e simplicidade. As técnicas NRZ apresentavam problemas de sincronização com longas sequências de zeros ou uns, o que poderia comprometer a integridade da transmissão em alguns casos.

A realização de testes em ambientes controlados pode minimizar interrupções e validar o desempenho do sistema em condições ideais. Além disso, uma possível melhoria seria integrar o projeto com outros sistemas de comunicação, como infravermelho ou radiofrequências, que proporcionam um método de comunicação híbrido.

Com mais tempo e experiência, o projeto poderia ser expandido para enviar mais tokens, aumentando sua usabilidade e eficiência, além de adicionar opções de reenvio do mesmo dado por diferentes codificações.

O desenvolvimento deste projeto não só fortaleceu os conceitos teóricos das telecomunicações, como também proporcionou uma valiosa experiência prática na resolução de problemas reais. Embora a comunicação por meio de sinais luminosos seja difícil, ela tem se mostrado uma opção viável e promissora, principalmente em aplicações que requerem transmissão de dados em curtas distâncias e baixa interferência eletromagnética. O aprendizado adquirido durante este projeto pode ser a base para futuras inovações e melhorias.



## 6. REFERÊNCIAS

CARVALHO, Luiz F. **Projeto de Comunicação de Dados**. Disponível em: <[https://moodle.utfpr.edu.br/pluginfile.php/2506195/mod\\_resource/content/8/UTFPR\\_CD\\_projeto\\_2.pdf](https://moodle.utfpr.edu.br/pluginfile.php/2506195/mod_resource/content/8/UTFPR_CD_projeto_2.pdf)>. Acesso em: 10 de jun. de 2024.

FOROUZAN, Behrouz A. **Comunicação de dados e redes de computadores, 2010**. E-book. ISBN 9788563308474. Disponível em: <<https://integrada.minhabiblioteca.com.br/#/books/9788563308474/>>. Acesso em: 16 de jun. de 2024.

CARVALHO, Luiz F. **Comunicação de Dados: Modos de Transmissão**. Apresentação de slides. Moodle. Universidade Tecnológica Federal do Paraná, Brasil, 2024. Disponível em: <[https://moodle.utfpr.edu.br/pluginfile.php/2308628/mod\\_resource/content/3/7%20-%20Transmissao\\_sincrona\\_e\\_assincrona.pdf](https://moodle.utfpr.edu.br/pluginfile.php/2308628/mod_resource/content/3/7%20-%20Transmissao_sincrona_e_assincrona.pdf)>. Acesso em: 10 jun. 2024.



---

CARVALHO, Luiz F. **Comunicação de Dados: Camada de Enlace (parte 1)**. Apresentação de slides. Moodle. Universidade Tecnológica Federal do Paraná, Brasil, 2024. Disponível em: [https://moodle.utfpr.edu.br/pluginfile.php/2308650/mod\\_resource/content/6/8%20-%20Controle\\_de\\_enlace\\_e\\_deteccao\\_de\\_erros\\_pt1.pdf](https://moodle.utfpr.edu.br/pluginfile.php/2308650/mod_resource/content/6/8%20-%20Controle_de_enlace_e_deteccao_de_erros_pt1.pdf). Acesso em: 10 jun. 2024.

saadsaif0333. **Data Transfer Using LED Light (Li-Fi)**. Disponível em: <https://projecthub.arduino.cc/saadsaif0333/data-transfer-using-led-light-li-fi-ee33a3>. Acesso em: 10 jun. 2024.