

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO
RELATÓRIO TÉCNICO - COMUNICAÇÃO DE DADOS

GABRIEL MOLINA DE LIMA
GUILHERME DE ALMEIDA DO CARMO
GUSTAVO GEOVANE TAMIÃO DE SOUZA
PEDRO MIAN PARRA

COMUNICAÇÃO ENTRE DOIS ARDUINOS ATRAVÉS DE SINAIS
LUMINOSOS

APUCARANA, 2022

**GABRIEL MOLINA DE LIMA
GUILHERME DE ALMEIDA DO CARMO
GUSTAVO GEOVANE TAMIÃO DE SOUZA
PEDRO MIAN PARRA**

COMUNICAÇÃO ENTRE DOIS ARDUINOS ATRAVÉS DE SINAIS LUMINOSOS

Relatório Técnico do Projeto desenvolvido na disciplina de Comunicação de Dados, do curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná, campus Apucarana.

Orientador: Prof. Dr. Luiz Fernando Carvalho

APUCARANA, 2022

RESUMO

Este presente relatório apresenta a comunicação entre dois Arduinos Uno, onde é transmitida uma mensagem através de sinais luminosos. Um equipamento codifica a frase e envia os sinais luminosos, enquanto o outro os recebe, decodifica e exibe a mensagem recebida.

Para tal, implementamos os métodos de transmissão NRZ-L e NRZ-L, onde se é possível escolher qual se deseja utilizar. Para a garantia de que a transmissão foi bem sucedida, acrescentamos também um detector de erros baseado no CRC (Cyclic Redundance Check).

Palavras-chave: Comunicação. Transmissão. Dados. Arduino.

SUMÁRIO

1 INTRODUÇÃO.....	5
2 DESENVOLVIMENTO.....	6
2.1 Transmissor	6
2.2 Receptor	9
4 RESULTADOS.....	13
5 CONSIDERAÇÕES FINAIS.....	13
REFERÊNCIAS.....	14

1 INTRODUÇÃO

Com o objetivo de colocar em prática os conteúdos abordados na disciplina de Comunicação de Dados, o seguinte projeto busca realizar a comunicação entre dois dispositivos diferentes, utilizando dois microcontroladores do tipo Arduino Uno. Para tal, acoplamos um dispositivo com um LED, onde o mesmo será programado para pegar uma mensagem de texto e codificá-la. Para o receptor, acoplou-se um sensor fotovoltáico LDR, programado para decodificar as mensagens recebidas.

Para realizar o envio, a mensagem é convertida de modo texto para binário, onde se terá a possibilidade de transmiti-la através de dois métodos unipolares, o NZR-L e ou NZR-I.

Para o NZR-L, o nível do sinal determinará o valor do bit. Caso o valor do bit seja 0 o LED enviará um sinal, caso seja 1 o LED não acenderá.

Para o NZR-I, a mudança ou a falta de uma mudança no nível do sinal determinará o valor do bit transmitido. Caso o valor do bit seja zero, o LED se mantém na sua atual condição, seja ela acesa ou apagada, caso seja 1, o LED inverterá seu estado.

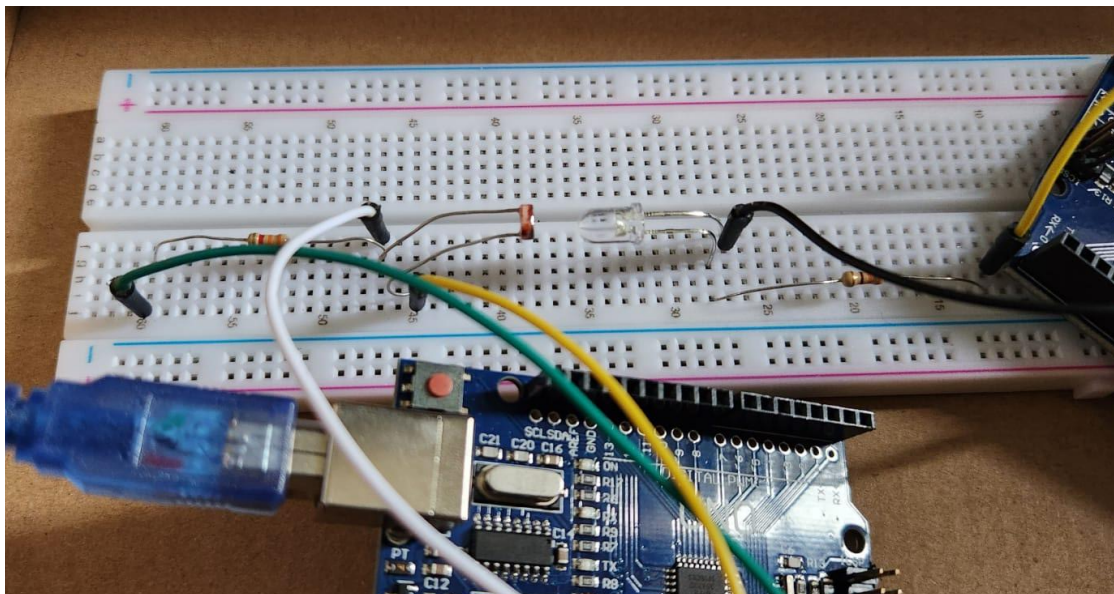
Como certificação de que a transmissão foi concluída com sucesso, o projeto conta com a implementação de um detector de erros baseado no CRC (Cyclic Redundance Check), onde o transmissor gera uma sequência de bits que resultará em um frame. Este frame será dividido por um número pré-determinado. Caso não haja resto nesta divisão, poderemos concluir que a transmissão foi bem sucedida.

2 DESENVOLVIMENTO

Para garantir o bom funcionamento do projeto, utilizamos uma caixa de sapato para inibir qualquer claridade que possa atrapalhar a comunicação. Além disso, é preciso ligar primeiramente o receptor, que irá aguardar uma emissão de luz do transmissor, iniciando assim em ambos uma contagem regressiva de cinco segundos.

Conforme apresentado na Figura 1, temos a montagem realizada do circuito para a execução deste projeto:

Figura 1 - Montagem do circuito.



Fonte: Autoria Própria.

3.1 Transmissor

Como primeiro passo, temos o dispositivo transmissor, onde se é preciso informar logo no início do código a mensagem que será enviada, e o método a ser utilizado:

Figura 2 - Informe da mensagem e do método.

```

22 // ==--== Mensagem para ser enviada ==--==
23
24 | | | | String Mensagem = "Mensgaem Teste";
25
26 // -----
27
28 // ==--== Escolha do método de envio ==--==
29 //          0 Para NRZ-Level
30 //          1 Para NRZ-Invert
31
32 | | | | int metodo = 1;
33

```

Fonte: Autoria Própria.

Após isso, o primeiro passo é converter essa mensagem para binário. Para isso, é preciso percorrer caractere por caractere, e passá-los para a função *itoa*, que irá retornar o número binário correspondente. Além disso, tem-se também um tratamento para os espaços, onde será acrescentado o bit '0':

Figura 3 - Conversão da mensagem para binário.

```

114 for (int i = 0; i < Mensagem.length(); i++) {
115     // Transformar caractere por caractere para binario
116     Caracter = Mensagem.charAt(i);
117     itoa(Caracter, binCaracter, 2);
118     binMensagem[strlen(binMensagem)] = '0';
119     // Tratamento para espaço
120     if (Caracter == ' '){
121         binMensagem[strlen(binMensagem)] = '0';
122     }
123     strcat(binMensagem, binCaracter);
124 }
125
126 Serial.print("Mensagem: ");
127 Serial.print(Mensagem);

```

Fonte: Autoria Própria.

Para o controle de erros, temos o cálculo do CRC (Cyclic Redundance Check) para enviar:

Figura 4 - Cálculo do CRC.

```

67     for (int j = keylen; j <= codelen; j++){
68         for (int i = 0; i < keylen; i++){
69             temp[i] = rem[i];
70         }//for
71
72         if (rem[0] == '0'){
73             for (int i = 0; i < keylen - 1; i++){
74                 rem[i] = temp[i + 1];
75             }//for
76         }else{
77             for (int i = 0; i < keylen - 1; i++){
78                 rem[i] = exor(temp[i + 1], key[i + 1]);
79             }// for
80         }// if else
81         if (j != codelen){
82             rem[keylen - 1] = data[j];
83         }else{
84             rem[keylen - 1] = '\0';
85         }// if else
86     }// for
87
88     for (int i = 0; i < keylen - 1; i++){
89         data[datalen + i] = rem[i];
90     }// for

```

Fonte: Autoria Própria.

Com base no método escolhido, NRZ-I ou NRZ-L, será montada a string de bits que será utilizada para informar se é preciso acender ou apagar o led:

Figura 5 - Definição da string de envio com base no método escolhido.

```

140
141     switch (metodo){
142         // NRZ-Level
143         case 0:
144             for (int i = 0; i < (strlen(binFinal))+1; i++){
145                 if(binFinal[i]== '1') {
146                     binFinal[i] = '0';
147                 }else if (binFinal[i]== '0') {
148                     binFinal[i] = '1';
149                 }// eske if
150             }// for
151             break;
152
153         // NRZ-Invert
154         case 1:
155             for (int i = 0; i < (strlen(binFinal))+1; i++){
156                 if(binFinal[i]== '1') {
157                     if(aux=='1'){
158                         aux='0';
159                     }else{
160                         aux='1';
161                     }//if else
162                     binFinal[i] = aux;
163                 }else if (binFinal[i]== '0') {
164                     binFinal[i] = aux;
165                 }// if else
166             }//for
167             break;
168     }// switch

```

Fonte: Autoria Própria.

É emitida então uma luz para ambos os aparelhos iniciarem a contagem regressiva:

Figura 6 - Processo de sincronização.

```

183
184     digitalWrite(9, HIGH);
185     Serial.println("Envio comecara em 5 segundos");
186     delay(4000);
187     digitalWrite(9, LOW);
188     delay(1000);
189

```

Fonte: Autoria Própria.

3.1 Receptor

Para o receptor, logo no início do código também é preciso informar o método a ser utilizado:

Figura 7 - Informe do método.

```

29 // ==--== Escolha do método de envio ==--==
30 //          0 Para NRZ-Level
31 //          1 Para NRZ-Invert
32
33 | | | | int metodo = 1;
34

```

Fonte: Autoria Própria.

Ele então aguarda a emissão da luz para iniciar a contagem regressiva. Após, temos então duas condições, uma para o led aceso e outra caso esteja apagado. Dentro de ambas as condições, utilizamos então outra condicional com base em qual método está sendo utilizado para definir o valor do bit:

Figura 8 - Recebimento da mensagem com base no método escolhido, com LED aceso.

```

217 // LED aceso
218 if (valor > 400){
219     switch (metodo){
220         // NRZ-Level
221         case 0:
222             mensagem[i] = '1';
223             binFinal[i] = '0';
224             break;
225
226         // NRZ-Invert
227         case 1:
228             mensagem[i] = '1';
229
230             if(mensagem[i]==anterior){
231                 binFinal[i] = '0';
232             }else{
233                 binFinal[i] = '1';
234             }
235             anterior = mensagem[i];
236
237             break;
238
239         default:
240             Serial.println("Metodo de envio invalido!");
241             break;
242     }// switch
243

```

Fonte: Autoria Própria.

Figura 9 - Recebimento da mensagem com base no método escolhido, com LED apagado.

```

245 }else{
246     switch (metodo){
247         // NRZ-Level
248         case 0:
249             mensagem[i] = '0';
250             binFinal[i] = '1';
251             break;
252
253         // NRZ-Invert
254         case 1:
255             mensagem[i] = '0';
256
257             if(mensagem[i]==anterior){
258                 binFinal[i] = '0';
259             }else{
260                 binFinal[i] = '1';
261             } // if else
262             anterior = mensagem[i];
263             break;
264
265         default:
266             Serial.println("Metodo de envio invalido!");
267             break;
268     } // switch
269

```

Fonte: Autoria Própria.

A maneira que encontramos para informar o fim do recebimento da mensagem é emitir uma meia-luz, por isso que é importante que o equipamento esteja dentro de um ambiente onde se tenha o controle da iluminação (no nosso caso, a caixa de sapato). Com isso então, caso a intensidade da iluminação esteja dentro dos parâmetros especificados, o receptor interpretará que toda a mensagem já foi recebida, e começará a receber o CRC:

Figura 10 - Leitura da meia-luz.

```

135
136 // Caso receba uma 'meia luz', significa que começou o envio do CRC
137 if (valor > 400 && valor < 650){
138
139     Serial.println("Mensagem Recebida!");
140     Serial.println("Começando recebimento do CRC");
141

```

Fonte: Autoria Própria.

Ao final do recebimento de todos os dados, o receptor irá então decodificar a mensagem recebida, convertendo de oito em oito bits para o caractere correspondente através da função *strtol*:

Figura 11 - Mostra da mensagem recebida

```

285     Serial.print("Mensagem Final recebida: ");
286     for(int i = 0; i < (strlen(binFinal))-16;i+=8){
287         for(int p = 0; p < 8;p++){
288             auxChar[p] = binFinal[i+p];
289         }
290         c = strtol(auxChar, 0, 2);
291
292         Serial.print(c);
293     }
294 }
295

```

Fonte: Autoria Própria.

Após então, é chamada a função que calculará o CRC, realizando então a divisão por um número pré-definido. Caso o resto desta divisão seja zero, concluiremos então que a mensagem foi recebida corretamente:

Figura 12 - Cálculo do CRC no receptor.

```

66
67     for (int j = keylen; j <= codelen; j++){
68         for (int i = 0; i < keylen; i++){
69             temp[i] = rem[i];
70         }// for
71
72         if (rem[0] == '0'){
73             for (int i = 0; i < keylen - 1; i++){
74                 rem[i] = temp[i + 1];
75             }// for
76         }else{
77             for (int i = 0; i < keylen - 1; i++){
78                 rem[i] = exor(temp[i + 1], key[i + 1]);
79             }// for
80         }// if else
81
82         if (j != codelen){
83             rem[keylen - 1] = data[j];
84         }else{
85             rem[keylen - 1] = '\0';
86         }// if else
87     }//for
88
89     for (int i = 0; i < keylen - 1; i++){
90         data[datalen + i] = rem[i];
91     }// for
92
93     data[codelen] = '\0';
94
95     Serial.print("O resto da divisao: ");
96     Serial.println(rem);
97

```

Fonte: Autoria Própria.

3 RESULTADOS

Aqui então apresentamos um teste de transmissão realizado, onde através do método NZR-I enviamos o escrito “Mensagem Teste”. Pelo Serial Monitor é possível acompanhar todo o processo, desde a conversão para binário, até o envio de bit a bit:

Figura 13 - Serial Monitor do transmissor.

```
Envio comecara em 5 segundos
Mensagem: Mensgaem Teste
Codigo binario: 010011010110010101101110011100110110011101100001011001010110110100100000010101000110010101110011011
CRC para envio:
011010001001010
Data para envio:
010011010110010101100111001101100111011000010110010101101101001000000101010001100101011100110111010001100101011
Transmissor iniciando
Envio comecara em 5 segundos
Envio iniciado!
1 0 0 0 1 0 0 1 1 0 1 1 1 0 0 1 1 0 1 1 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 1 0 1 0 0 1 0 0 0 0 0 1 1 0 1 1 1 0 0 1 1 0
Iniciando envio do CRC
1 0 1 1 0 0 0 0 1 1 1 0 0 1 1 □ □ Dados transmitidos
Encerrando...
```

Fonte: Autoria Própria.

Pelo Serial monitor podemos então acompanhar o recebimento bit a bit. Ao final, temos então a exibição da mensagem já decodificada, bem como o resto da divisão efetuada no cálculo do CRC. Como é possível observar, o resto da divisão igual a zero nos informa que a mensagem foi recebida corretamente, sem a ocorrência de nenhuma interferência:

Figura 14 - Serial Monitor do receptor.

```
Luminosidade: 939
Em 5 segundos comecara o recebimento da mensagem
Iniciando o recebimento da mensagem
1 0 0 0 1 0 0 1 1 0 1 1 1 0 0 1 1 0 1 1 0 1 0 0 0 1 0 1 1 1 0 1 1 1 0 1 0 0 1 0 0 0 0 0 1 1 0 1 1 1 0 0 1 1
Comecando recebimento do CRC
1 0 1 1 0 0 0 0 1 1 1 0 0 1 1 Mensagem:10001001101110011011010001011101101110100100000110111001101101000111111100
CRC: 101100001110011
Mensagem Final recebida: Mensgaem Teste
O resto da divisao: 0000000000000000
```

Fonte: Autoria Própria.

4 CONSIDERAÇÕES FINAIS

Concluimos então que foi possível desenvolver este projeto com um certo êxito. Para se alcançar a excelência, seria necessária a implementação do “timer interrupt”, onde seria possível garantir que não houvesse nenhuma perda de sincronização durante o processo de transmissão da mensagem.

REFERÊNCIAS

- [1] VIANA, C. C.; SOUZA, J.; SANTANA, C. D. S. **Utilizando o sensor de Luminosidade Ldr no Arduino.** Disponível em: <<https://www.blogdarobotica.com/2020/09/29/utilizando-o-sensor-de-luminosidade-ldr-no-arduino/>>. Acesso em: 02 de Dezembro de 2022.
- [2] FOROUZAN, B. A.; FEGAN, S. C.; GRIESI, A. Comunicação de Dados e redes de computadores. Tradução . São Paulo: McGraw Hill, 2008.
- [3] DISSE:, R.; DISSE:, F. G. Controlando O Brilho de um led (PWM) - aula 6 - ai. Disponível em: <<https://mundoprojetado.com.br/controlando-o-brilho-de-um-led-pwm/>>. Acesso em: 02 de Dezembro de 2022.