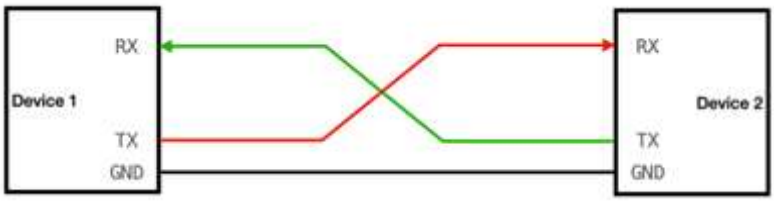
Revisar envio do teste: avaliação final

Usuário	ENRICCO GEMHA
Curso	CAMADA FÍSICA DA COMPUTAÇÃO - 4ENGCOMA 2022/2
Teste	avaliação final
Iniciado	08/12/22 13:30
Enviado	08/12/22 14:29
Data de vencimento	08/12/22 15:45
Status	Completada
Resultado da tentativa	6,5 em 10 pontos
Tempo decorrido	58 minutos

Pergunta 1

0 em 2 pontos

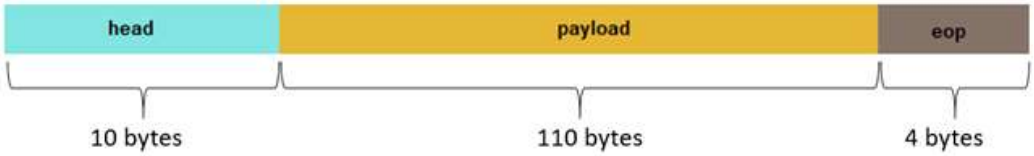
Durante os projetos de transmissão serial com o protocolo UART utilizamos dois Arduinos conectados como mostra a figura abaixo. Lembre-se de que você podia configurar parâmetros (nem todos, algumas coisas são obrigatórias) de sua comunicação UART através da classe *fisica*, como *baudrate*, número de *stop bits* e número de *bits* de *paridade*.



Considere uma transmissão feita com a configuração mostrada na figura abaixo:

```
15  ################################
16  # Interface com a camada física #
17  ################################
18  class fisica(object):
19      def __init__(self, name):
20          self.name        = name
21          self.port        = None
22          self.baudrate    = 115200
23          self.bytesize    = serial.EIGHTBITS
24          self.parity      = serial.PARITY_ONE
25          self.stop        = serial.STOPBITS_ONE
26          self.timeout     = 0.1
27          self.rxRemain    = b""
```

Usando esse modo de transmissão (*frame*) exemplificado acima, você ainda obedeceu à regra de uma camada superior que exigia o datagrama mostrado na figura a seguir para envio de arquivos. Nesse datagrama, apenas o *payload* pode assumir tamanho variável, entre 0 e 110 bytes.

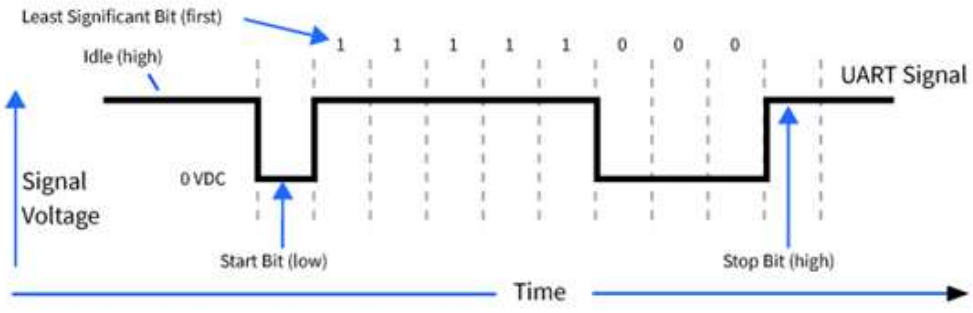


a) Com essa configuração UART e esse datagrama, qual o menor tempo possível, em milisegundos, para o envio de um arquivo de 1k bytes?

Pergunta 2

2 em 2 pontos

Ainda sobre a situação descrita na questão 1, suponha que você tenha alterado a configuração UART e o novo *frame* passou a ser o mostrado na figura:



Qual o novo tempo, em milisegundos, de transmissão com o novo frame?

Pergunta 3

1,5 em 1,5 pontos

No projeto 6 do curso de Camada Física da computação você implementou um código que serializava um byte e o enviava através de um pino do Arduino. Nesse código você utilizou funções de espera, que inativavam o processamento por um período, como mostradas na figura as fuções para espera de 1 período e de meio período.

```
81  void _sw_uart_wait_half_T(due_sw_uart *uart) {
82    for(int i = 0; i < NUMERO_CK; i++)
83      asm("NOP");
84  }
85
86  void _sw_uart_wait_T(due_sw_uart *uart) {
87    _sw_uart_wait_half_T(uart);
88    _sw_uart_wait_half_T(uart);
89  }
```

Sendo o *clock* do processador de 5,3MHz , por quanto você teria de fixar o valor da variável **NUMERO_CK**, presente no *loop for* da função *_sw_uart_wait_half_T* , para que o baudrate seja o de 115200 bits/s ?

Pergunta 4

1,5 em 1,5 pontos

Ainda sobre a questão anterior, qual a melhor justificativa para a necessidade de se ter uma função de espera de meio período além da outra de um período inteiro?

Pergunta 5

1,5 em 1,5 pontos

Nos projetos DTMF e Modulação AM você utilizou a digitalização de um sinal de áudio seguida de geração de arquivos contento os dados digitalizados. Suponha que você utilizou uma placa de áudio que digitalizou um sinal por 5 segundos em apenas 1 canal, com taxa de amostragem de 44,1kHz . Suponha ainda que cada amostra foi digitalizada com 16 bits de resolução.

Qual o tamanho do arquivo, em **quilobytes** contendo os dados digitalizados?

Pergunta 6

0 em 1,5 pontos

O código abaixo mostra uma parte de um código de transmissão de 2 sinais com **modulação AM** através de 2 portadoras. O filtro tem uma frequência de corte de 2 kHz.

```
####################
# Gera portadora
####################
freqPortadora1 = 13000
freqPortadora2 = 19000
xPortadora1, yPortadora1 = generateSin(freqPortadora1, 1, samplesAudio1/samplerate, samplerate)
xPortadora2, yPortadora2 = generateSin(freqPortadora2, 1, samplesAudio2/samplerate, samplerate)


####################
# Gera sinal AM
# AM-SC
####################
yAM1 = Y1 * yPortadora1
yAM2 = Y2 * yPortadora2

####################
# Aplica filtro no sinal
####################
yAM1 = signal.lfilter(taps, 1.0, audio1Norm)
yAM2 = signal.lfilter(taps, 1.0, audio2Norm)
```

A respeito desse código é correto afirmar que:

Quinta-feira, 25 de Maio de 2023 14h31min44s BRT

← **OK**