

# CAMADA FÍSICA DA COMPUTAÇÃO

## ENGENHARIA DA COMPUTAÇÃO - Rodrigo Carareto

### PROJETO LOOP BACK

Neste projeto você deverá construir um código em Python para transmissão e recepção serial **simultâneas**!

#### Entrega:

**Esta primeira atividade não é computada para nota. Porém, você e sua dupla (os projetos serão sempre realizados em dupla) deverão mostrar ao seu professor o software em funcionamento durante a aula da próxima quinta.**

**Faça testes! Explore as classes! Você irá utilizá-las como base para outros projetos e deverá ter segurança para alterá-las.**

#### Objetivo:

Ao rodar seu arquivo aplicação, o seu software deve:

- 1) Enviar uma imagem (a menor possível) através da porta de comunicação serial.
- 2) Receber a imagem simultaneamente ao envio e salva-la como uma cópia. Para isso a recepção do Arduino (*pino rx*) deve estar curto-circuitada com o pino de transmissão (*pino tx*).

#### Material:

Você irá utilizar um Arduino, um computador e 5 arquivos código fornecidos.

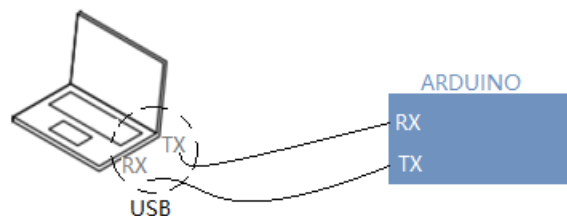
No console python, 'pip install pyserial'

Verificar em gerenciador qual porta com está o Aduino

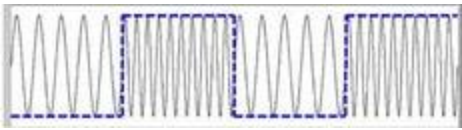
Talvez mudar a porta USB ou tirar e recolocar quando não tem autorização :-\

#### Montagem:

Seu sistema operacional irá abrir uma porta de comunicação serial com o Arduino. Através dessa comunicação você terá acesso a tudo que o Arduino enviar ao seu computador (saindo do pino TX do Arduino e RX da sua porta USB). Tudo que seu computador enviar ao Arduino (saindo do pino TX da sua USB em seu computador). Desta maneira, ao conectar o Arduino ao seu computador, o pino de envio do seu computador (TX) estará conectado ao pino de recepção do Arduino (RX). Da mesma forma, o pino de envio do Arduino (TX) estará conectado ao pino de recepção do seu computador (RX).

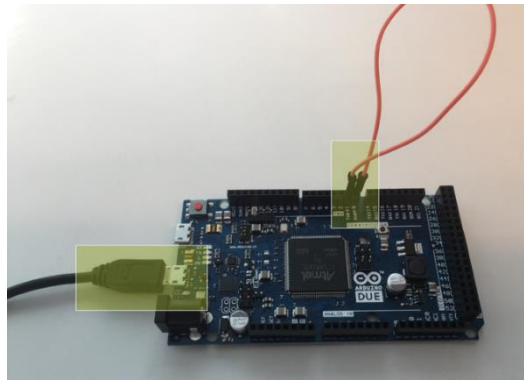


O queremos é que ao enviarmos uma mensagem (lista de bytes) ao Arduino, este responda com os mesmos bytes. Queremos que o Arduino seja um espelho para os bytes enviados. Para isso basta que conectemos o pino TX do Arduino ao seu pino RX! Assim, ao enviarmos uma sequência de bytes para o RX do Arduino, a mesma sequência de zeros e uns são produzidas no pino de envio do Arduino e recebidos de volta em seu computador. Observe a figura:



# CAMADA FÍSICA DA COMPUTAÇÃO

## ENGENHARIA DA COMPUTAÇÃO - Rodrigo Carareto



Você precisará verificar qual os pinos TX e RX de seu Arduino.

**ATENÇÃO! ALGUNS ARDUINOS (UNO) PRECISAM FICAR COM O BOTÃO DE RESET PRESSIONADO. OU O PINO RESET ATERRADO!**

### Imagens em python.

Você terá que transformar uma imagem em uma lista de bytes. No mesmo modo, salvar a lista de bytes recebida, como imagem. Para isso pode se basear nos seguintes trechos de código:

Caminhos das imagens

```
# Endereco da imagem a ser transmitida
imageR = "./imgs/image.png"

# Endereco da imagem a ser salva
imageW = "./imgs/recebidaCopia.png"
```

Lista de bytes com a imagem a ser transmitida

```
# Carrega imagem
print ("Carregando imagem para transmissão :")
print (" - {}".format(imageR))
print("-----")
txBuffer = open(imageR, 'rb').read()
```

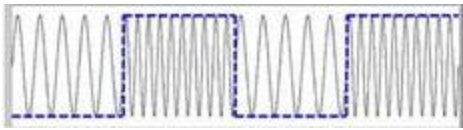
Escreve arquivo cópia

```
print ("Salvando dados no arquivo :")
print (" - {}".format(imageW))
f = open(imageW, 'wb')
f.write(txBuffer)

# Fecha arquivo de imagem
f.close()
```

### Pronto! Consegui!

Se você conseguiu fazer o arquivo cópia através da comunicação serial e este arquivo abre normalmente, parabéns! Vamos aproveitar então para explorar um pouco mais nossas classes.



# CAMADA FÍSICA DA COMPUTAÇÃO

ENGENHARIA DA COMPUTAÇÃO - Rodrigo Carareto

- 1) Sua primeira tarefa é explicar o motivo pelo qual a função *getStatus* está retornando zero. Tente solucionar o problema.
- 2) Na classe *física* no arquivo *interfaceFisica* há uma propriedade chamada *self.baudrate* que está definida como 115200. Se a alterar para 9600, qual deverá ser o efeito?
- 3) Você saberia explicar a diferença entre as funções *getBuffer* e *getAllBuffer*?