

UNIVERSIDADE FEDERAL DA PARAÍBA  
CENTRO DE ENERGIAS ALTERNATIVAS E RENOVÁVEIS (CEAR)  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

# Tutorial: Comunicação com Arduino e LabVIEW

Josman Silva Rodrigues  
Prof. Dr. Juan Moises Mauricio Villanueva





# Utilizando a biblioteca LINX

- Após baixar o LabVIEW, abrir o aplicativo VI Package Manager, que vem junto com a instalação.



Símbolo do VI Package Manager



# Utilizando a biblioteca LINX

Caixa de pesquisa. Pesquisar por LINX

Name ^	Version	Repository	Company
✓ Diligent LINX (Control Arduino, R	3.0.1.192	NI LabVIEW Tools Network	Diligent
Diligent WaveForms VIs	1.0.3.26	NI LabVIEW Tools Network	Diligent inc
Distributed State Machines (DSM)	0.0.0.28	NI LabVIEW Tools Network	Ovak Technologies
DMC GUI Suite	7.0.1.1	NI LabVIEW Tools Network	DMC
DQMH	6.0.0.83	NI LabVIEW Tools Network	DQMH Consortium
DQMH Event Scripter	6.0.0.122	NI LabVIEW Tools Network	DQMH Consortium
DQMH Palette	6.0.0.14	NI LabVIEW Tools Network	DQMH Consortium
DQMH Project Template	6.0.0.91	NI LabVIEW Tools Network	DQMH Consortium
DQMH Thermal Chamber Examp	6.0.0.49	NI LabVIEW Tools Network	DQMH Consortium
DSM Framework Example	1.0.0.5	NI LabVIEW Tools Network	Ovak Technologies
Dual Port SPI Example for LabVIEW	1.0.0.1	NI LabVIEW Tools Network	National Instruments
DVR Helper	1.0.0.14	VIPM Community	EUV Tech
Dynamic Signal Analyzer	1.0.0.35	NI LabVIEW Tools Network	MIT
Dynamic VI Handler	1.5.0.23	NI LabVIEW Tools Network	PhysicsConsult
EAL for LabVIEW	2.0.0.10	NI LabVIEW Tools Network	Bosch Rexroth AG
EasyXML Toolkit for LabVIEW	3.0.0.170	VIPM Community	JKI
Enclose With Error Case Structure	1.0.0.1	VIPM Community	Bienkowski Michal
Encryption Compendium for Lab	2.2.0.13	NI LabVIEW Tools Network	Lvs-Tools.Co.Uk
Enhanced State Diagram Toolkit	1.0.0.7	NI LabVIEW Tools Network	National Instruments
Epoch Date & Time	1.0.0.2	VIPM Community	LabVIEW Open Source Project
Epson Robotics Library	1.1.0.47	NI LabVIEW Tools Network	ImagingLab
Epson Robotics Library	3.2.0.84	NI LabVIEW Tools Network	DigiMetrix
Error Logger	1.0.0.2	NI LabVIEW Tools Network	CPE Systems NZ Ltd
Error Structure Drawer	2.2.0.2	VIPM Community	Stuart J. Box
Eulogy	1.0.1.52	NI LabVIEW Tools Network	Field R&D Services
Event Source Actor	1.1.0.5	NI LabVIEW Tools Network	NI
Expression Scanner	1.0.0.10	NI LabVIEW Tools Network	GROVF
Extended Tiny Encryption Algorit	1.0.0.2	NI LabVIEW Tools Network	National Instruments
Fanuc Robotics Library	1.9.3.2	NI LabVIEW Tools Network	DigiMetrix
Flat UI Controls Suite	1.0.0.1	NI LabVIEW Tools Network	Ovak Technologies
Flat User Interface Controls Suite	1.0.0.1	NI LabVIEW Tools Network	Ovak Technologies
Flatline Controls	1.9.1.33	NI LabVIEW Tools Network	JDP Science
Flexible eHS Real Time Power Elec	2016.0.0.1	NI LabVIEW Tools Network	OPAL-RT Technologies
FPGA IP (IPNet): Digital Buses anc	1.0.0.4	NI LabVIEW Tools Network	National Instruments
FPGA IP (IPNet): LabVIEW FPGA E	1.0.0.4	NI LabVIEW Tools Network	National Instruments
Frame APIs for PDC	1.0.0.2	NI LabVIEW Tools Network	National Instruments
FTDI Driver API for Serial Commu	1.0.3.20	NI LabVIEW Tools Network	Ovak Technologies
FTDI Driver for I2C SPI GPIO	1.0.0.31	NI LabVIEW Tools Network	AI

Botão de *refresh* para atualizar a lista de bibliotecas. Pode ser necessário na primeira utilização do *VIPM Package Manager*



# Utilizando a biblioteca LINX

The screenshot shows the LabVIEW Package Manager interface. The search bar contains 'linx'. The results table is as follows:

Name ^	Version	Repository	Company
Digilent LINX (Control Arduino, R	3.0.1.192	NI LabVIEW Tools Network	Digilent
LINX Raspberry Pi Addons	1.0.0.1	VIPM Community	MediaMongrels
NI LabVIEW LINX Toolkit	1.0.0.9	NI LabVIEW Tools Network	NI
Xilinx ChipScopePro Debugging E	2.1.0.11	NI LabVIEW Tools Network	B&A Engineering Inc. (BAI)

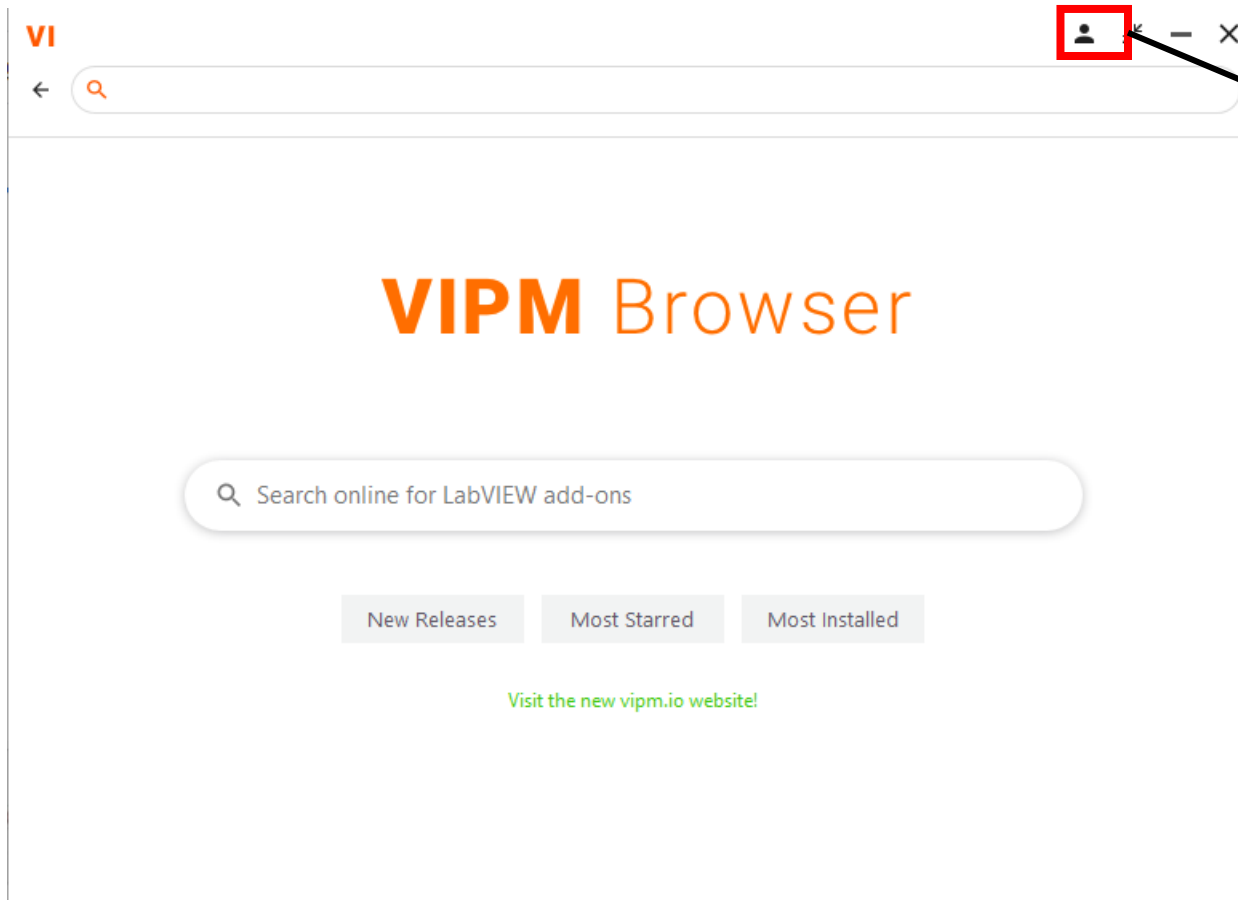
The 'Install' button for the selected package is highlighted with a red box.

- Selecionar o *Digilent LINX (Control Arduino, Raspberry Pi, BeagleBone and more)* e clicar em *Install*.
- Aceitar os termos e esperar a conclusão da instalação.



# Utilizando a biblioteca LINX

- Caso não encontre a biblioteca diretamente, pode-se utilizar outro aplicativo que vem junto da instalação do LabVIEW, o *VIPM Browser*

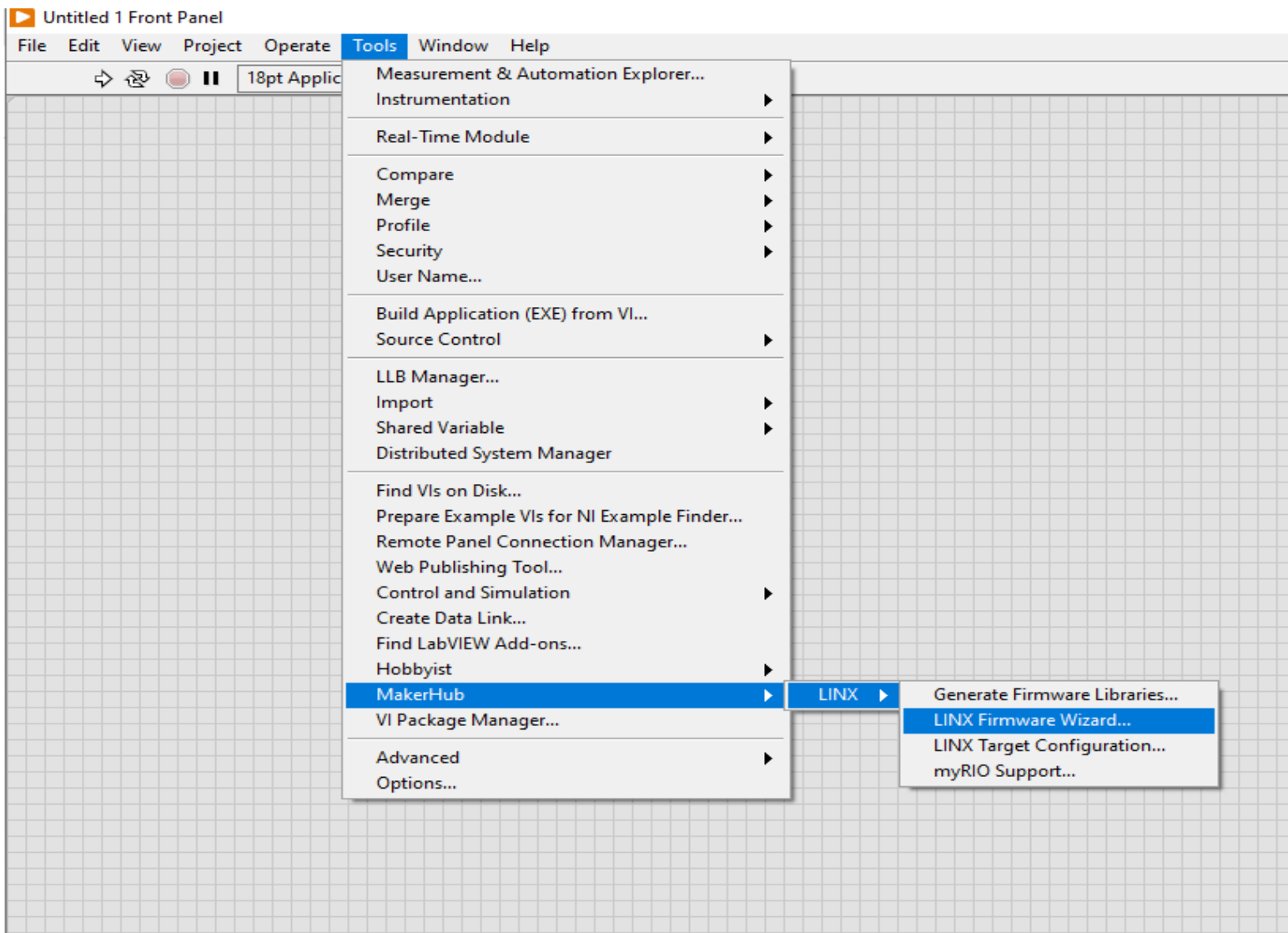


- Pode ser necessário criar uma conta antes de iniciar a instalação



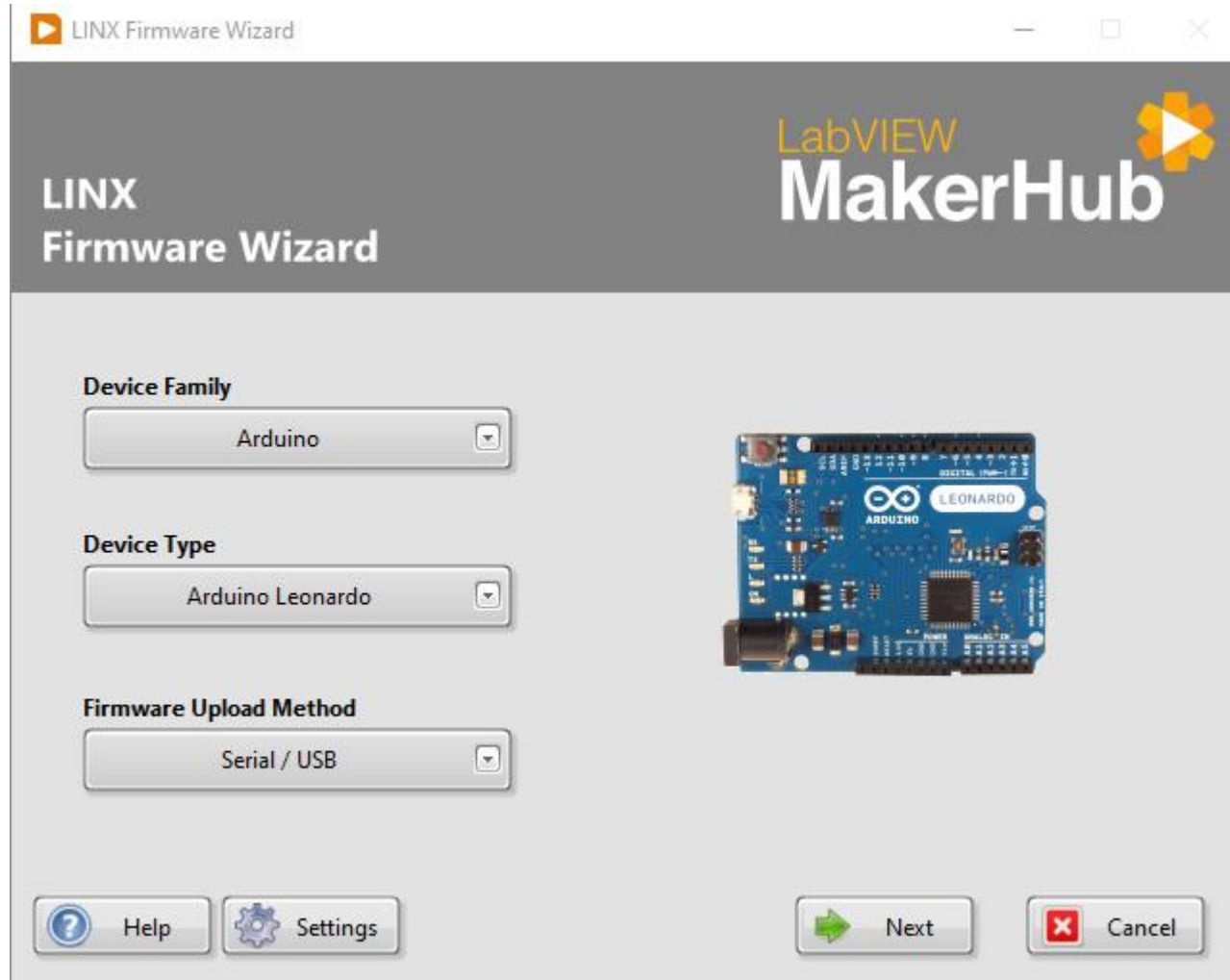
# Utilizando a biblioteca LINX

- Após instalar a biblioteca LINX, é necessário configurá-la para utilização. Portanto, conecte o Arduino ao computador e abra o LabVIEW e clique em *Tools* → *MakerHub* → *LINX* → *LINX Firmware Wizard*





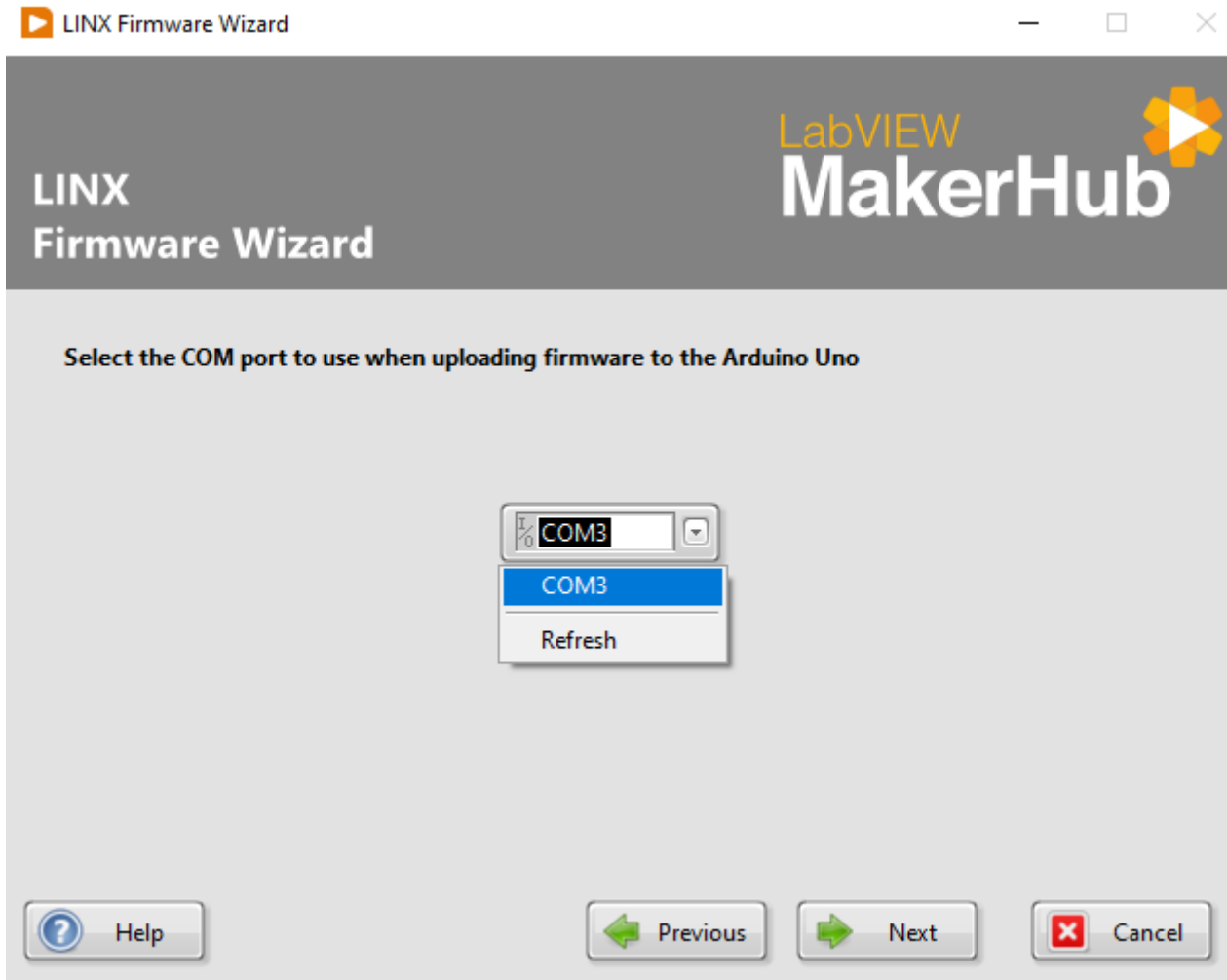
# Utilizando a biblioteca LINX



- Selecione a placa Arduino em *Device Family* e o tipo em *Device Type* (Uno, Mega, Leonardo...)
- O método de comunicação pode ser mantido em Serial/USB.
- Clique em *Next*.



# Utilizando a biblioteca LINX

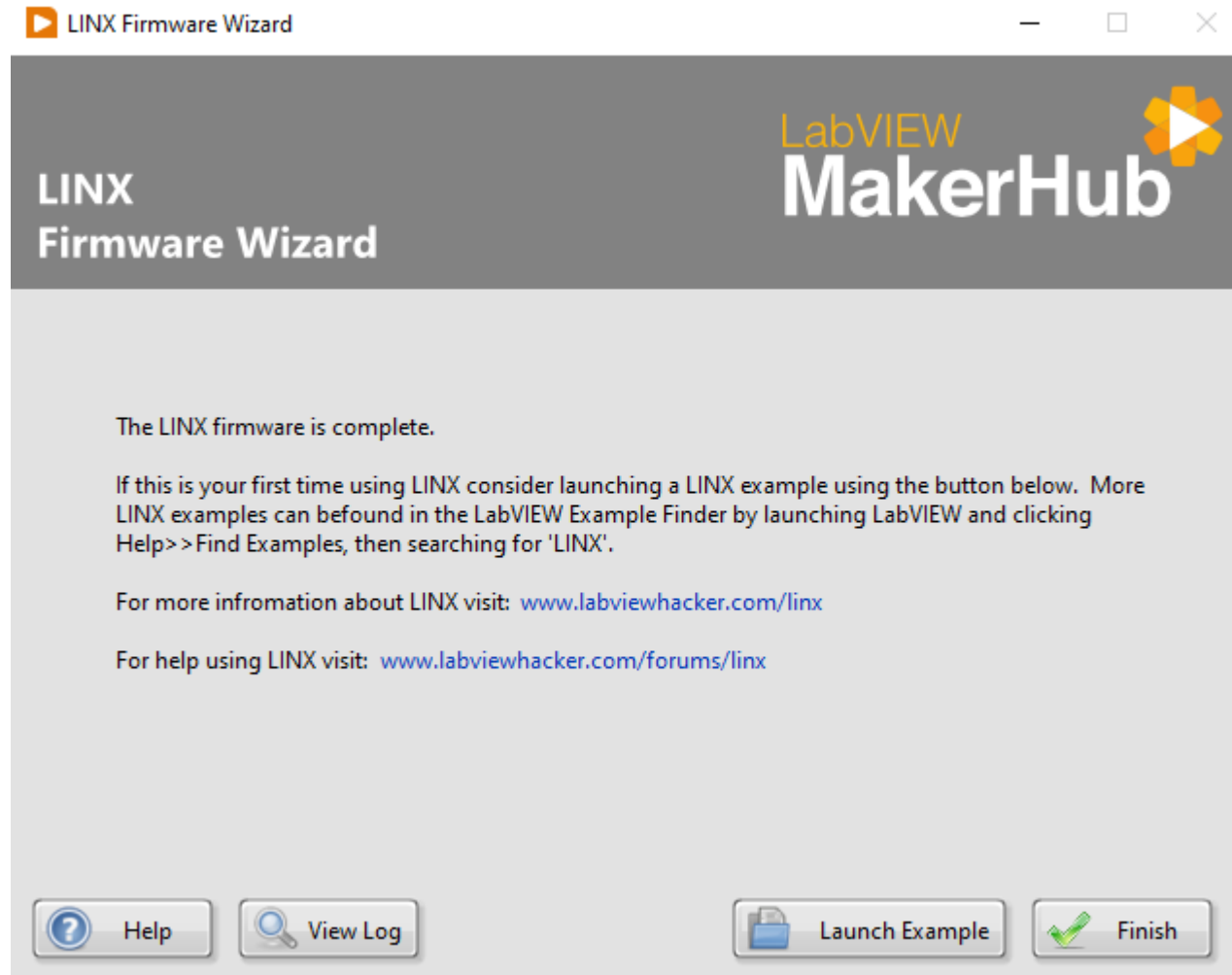


- Selecione a porta de comunicação COM (que pode ser visualizada na IDE do Arduino) e clique em *Next*.
- Caso não encontre de início, clique em *Refresh*.
- Clique em *Next* e espere finalizar a configuração





# Utilizando a biblioteca LINX



- Ao final da instalação, pode-se utilizar um exemplo da própria biblioteca clicando em *Launch Example*



# Utilizando a biblioteca LINX

LINX - Blink (Simple).vi Front Panel

File Edit View Project Operate Tools Window Help

18pt Application Font

### Manual Blink Example

This example demonstrates how to blink an LED on a LINX device by clicking on an LED control in LabVIEW.

**Instructions**

1. Select the **Serial Port** associated with the LINX Device.
2. Select the **Digital Output Channel** connected to the LED.
3. Click the **Run Arrow**.

Loop Rate (Hz)

#### Circuit Schematic

LED

Digital Output

R1  
220Ω

GND

#### LINX Device Settings

Serial Port

Digital Output Channel

#### LED Control

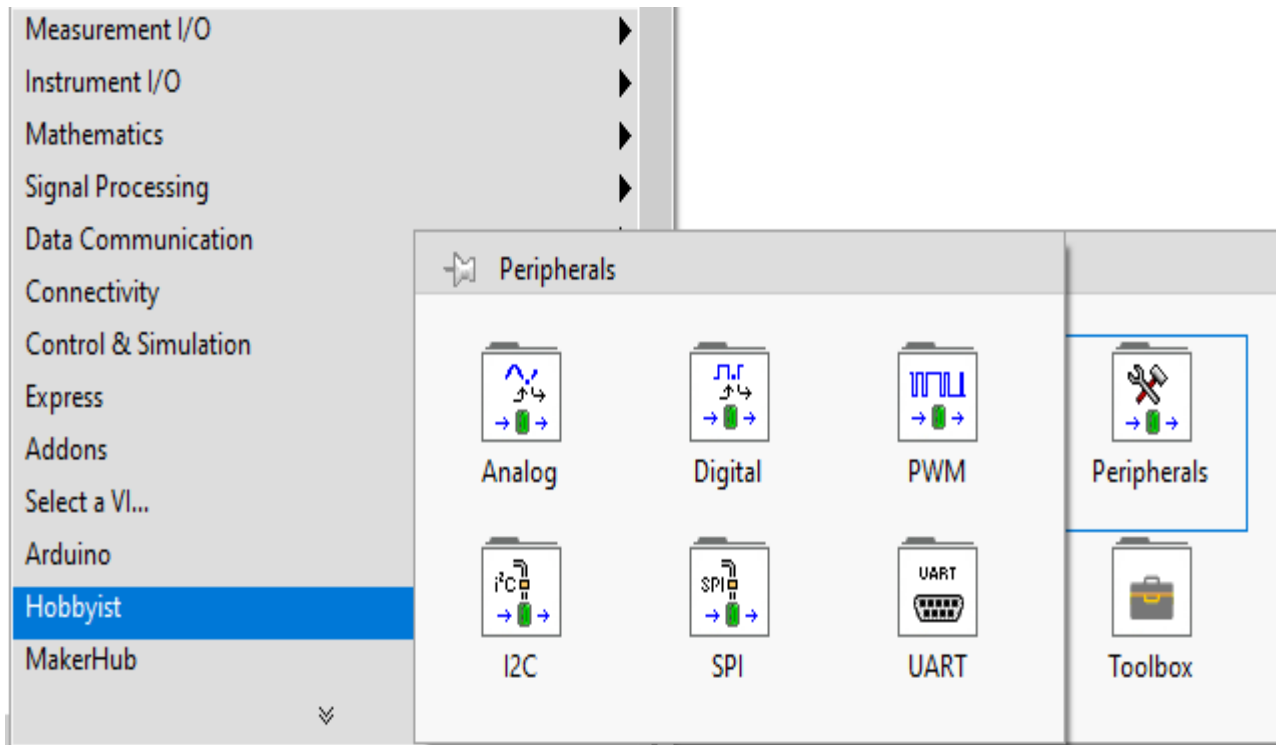
^^ Click Here ^^

- O exemplo da biblioteca é acender e apagar o LED do pino 13 do Arduino clicando no botão verde do LabVIEW (*LED Control*).
- Selecione a porta serial (COM) em *Serial Port* e o pino digital em *Digital Output Channel* (o pino do LED embutido no Arduino é o 13).





# Utilizando a biblioteca LINX



- Para acessar os elementos individualmente, clique no botão direito do mouse e vá em *Hobbyist* ou *MakerHub*.
- Em *Peripherals* estarão os blocos de leitura e escrita analógicos e digitais, além de outros blocos.

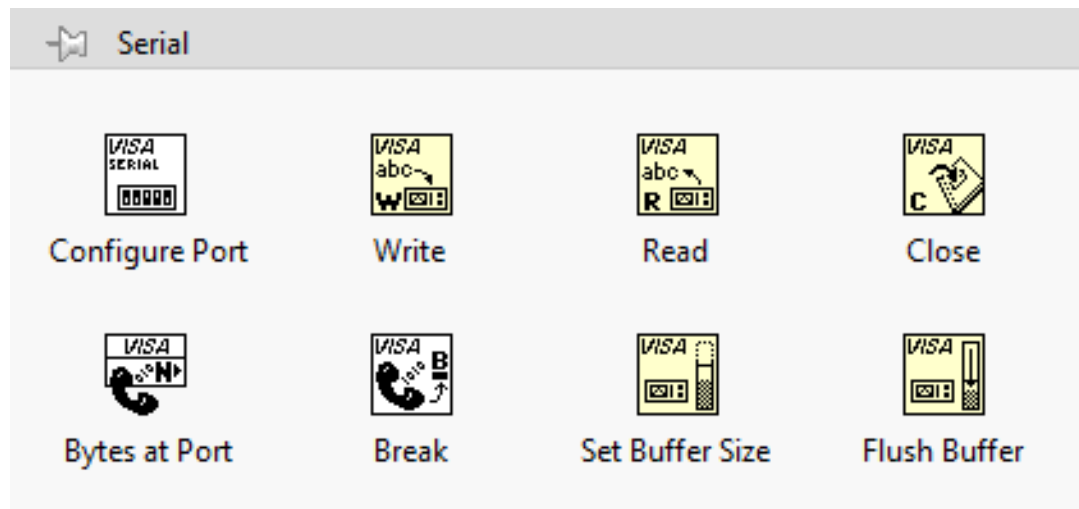
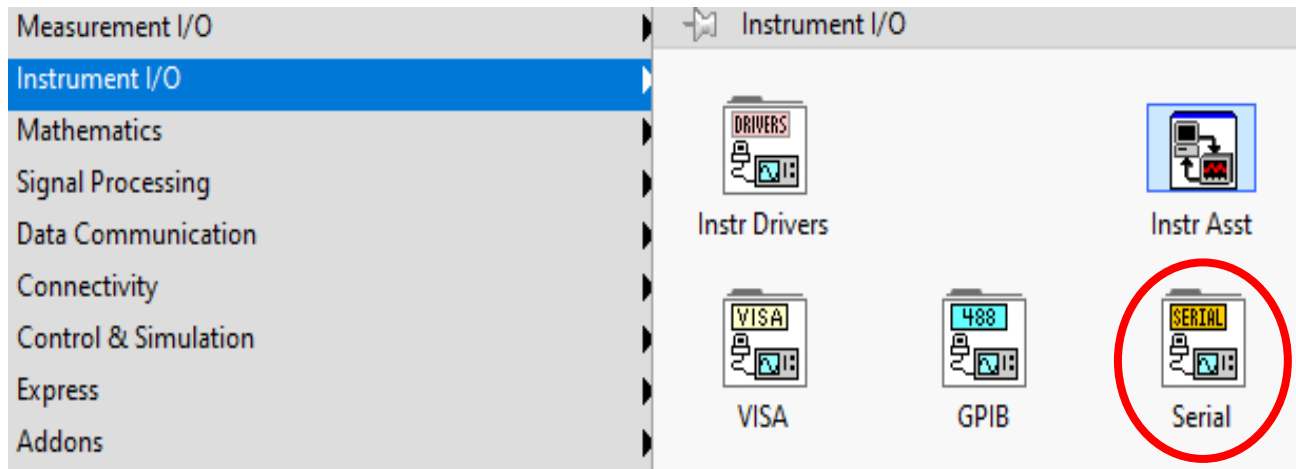


# Utilizando a biblioteca Serial do LabVIEW

- A comunicação com o Arduino pode ser feita diretamente via comunicação serial;
- É possível fazer operações de escrita e leitura da serial separadamente, com utilização de código implementado na própria IDE do Arduino.



# Utilizando a biblioteca Serial do LabVIEW

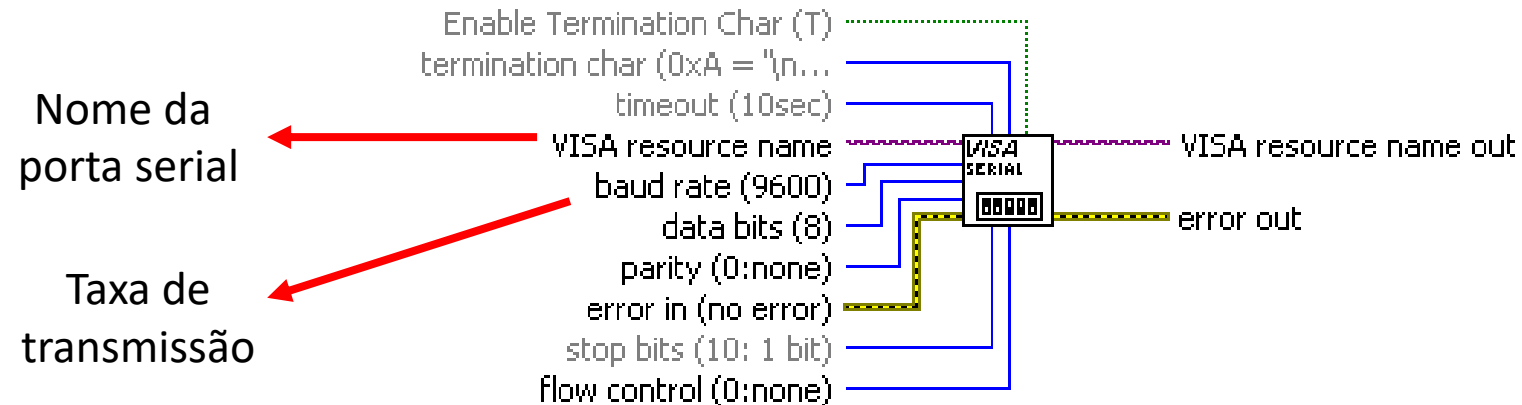


- A biblioteca serial pode ser encontrada clicando-se com o botão direito do mouse no espaço do diagrama de blocos e selecionando *Instrument I/O*;
- Os principais blocos são o *Configure Port*, *Write*, *Read* e *Close*



# Utilizando a biblioteca Serial do LabVIEW

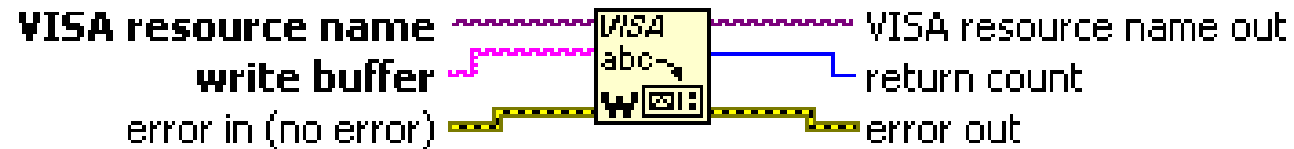
- O bloco *Configure Port* é utilizado no início do programa para inicializar e configurar a porta serial





# Utilizando a biblioteca Serial do LabVIEW

- O bloco *write* é utilizado para enviar informações (escrever) via serial. Em nosso caso, do LabVIEW para o Arduino. A informação a ser enviada é colocada na entrada *write buffer* do bloco.

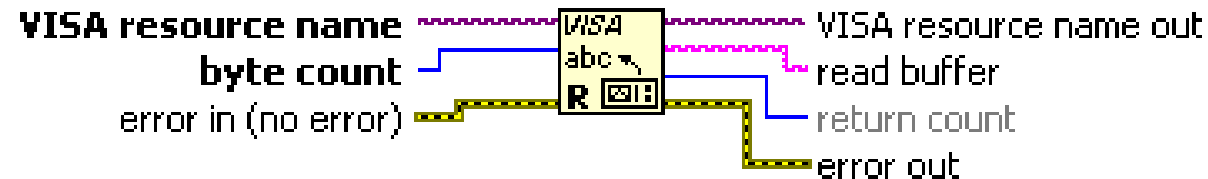






# Utilizando a biblioteca Serial do LabVIEW

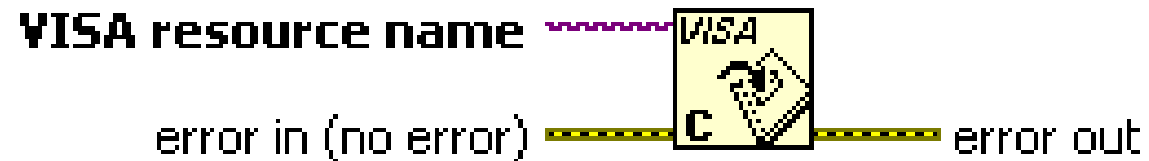
- O bloco *read* é utilizado para receber informações (ler) via serial. Em nosso caso, do Arduino para o LabVIEW. A informação a ser lida é colocada na saída *read buffer* do bloco.





# Utilizando a biblioteca Serial do LabVIEW

- O bloco *close* deve ser colocado no final do diagrama de blocos para finalizar o processo de comunicação serial;
- OBS: Os blocos *Configure Port* e *Close* geralmente são colocados fora do *loop while*.

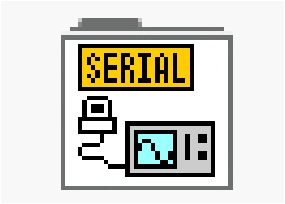




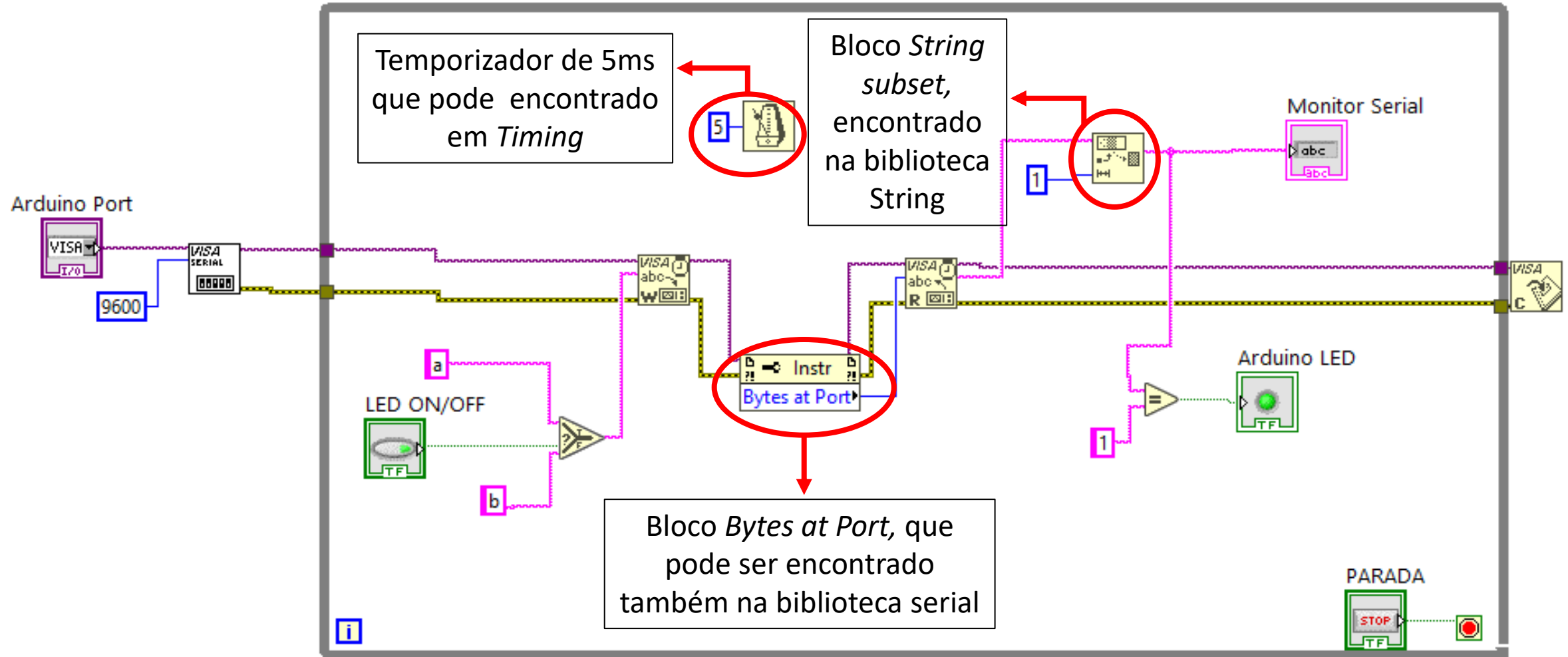
# Utilizando a biblioteca Serial do LabVIEW

- Exemplo: Ligar e desligar o LED do pino 13 do Arduino via comunicação serial e observar na interface do LabVIEW.





# Utilizando a biblioteca Serial do LabVIEW





# Utilizando a biblioteca Serial do LabVIEW

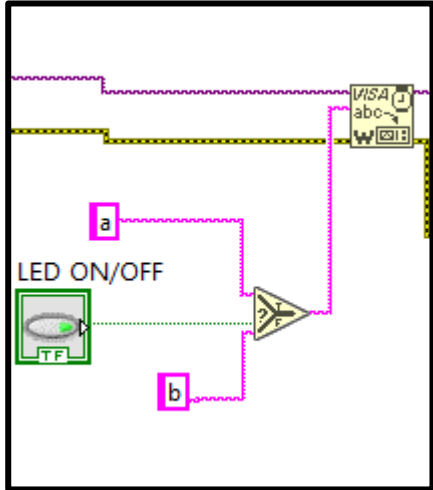
Código .ino a ser carregado antes de executar o programa em LabVIEW:

```
void setup() {  
    Serial.begin(9600);  
    pinMode(13, OUTPUT);  
}  
byte x;  
void loop() {  
  
    if(Serial.available()>0) x = Serial.read();  
  
    if (x == 'a') digitalWrite(13,HIGH);  
    else if (x=='b') digitalWrite(13,LOW);  
    if (digitalRead(13)==HIGH) Serial.write('1');  
    else Serial.write('2');  
}
```



# Utilizando a biblioteca Serial do LabVIEW

1



2

```
if (Serial.available() > 0) x = Serial.read();  
  
if (x == 'a') digitalWrite(13, HIGH);  
else if (x == 'b') digitalWrite(13, LOW);
```

```
if (digitalRead(13) == HIGH) Serial.write('1');  
else Serial.write('2');
```

3

4

Durante a execução do programa, o valor lido da serial pelo LabVIEW, isto é, 1 ou 2, é mostrado no Monitor Serial adicionado na interface.

## • Funcionamento:

1 - O LabVIEW envia via serial (escreve) o caractere 'a' se o botão (no LabVIEW) for pressionado e 'b' caso contrário;

2 - O Arduino faz a leitura da serial e liga ou desliga o LED (pino 13) de acordo com o valor recebido (a ou b);

3 - O Arduino envia via serial (escreve) os números 1 ou 2, indicando que o pino 13 está ativo ou não ativo, respectivamente;

4 - O valor enviado pelo Arduino via serial é lido pelo LabVIEW e há uma operação de comparação, isto é, se tal valor recebido for 1, indicando que o LED do Arduino está ligado, o LED adicionado no LabVIEW também liga, caso contrário, permanece desligado.

