

VISA R/W

NI-VISA Read: basic exercises

1. INTRODUCTION

One of the many applications that LabView has is an I/O software known as VISA (Virtual Instrument System Architecture). A standard tool to create interfaces that allow us to control (configure, program, debug) instruments/equipment or acquire data from external sensors. All the above tasks are done through any communication port like Ethernet, GPIB, USB, and more.

2. GOAL

Follow the directions to complete all exercises using the advantages of VISA to control and communicate with an external device (ARDUINO). After completing this activity, you will be able to connect, acquire, and send data through the USB serial port with the LabView program.

3. MATERIAL

- LabView software
- Arduino UNO + USB Cable
- Protoboard
- Jumper Kit
- Resistors (120W)
- Push-Button

4. METHODOLOGY

Use the knowledge acquired in the previous sessions:
Creating an interface in the front panel, variables, data processing and error handling.

Previous and Autonomous tasks:

Exercise NI-VISA Write

5. CHALLENGES

Digital Input: Visa Read Function

In this challenge you will learn how to use the Digital Inputs from your Arduino, and the Visa read function to communicate your hardware (Arduino board) and software (LabView) through the USB Serial Port.

✓ Exercise 6.3

The aim of this task is to receive data, either a "string" of letters "a" or "b", from the Arduino board to the LabView application.

Pressing an external push-button (protoboard) is the action that will allow the "string" to switch values. Add a "read buffer" indicator to show the String value.

1.- Open a new project in LabView and create the user interface. (Figure 1)

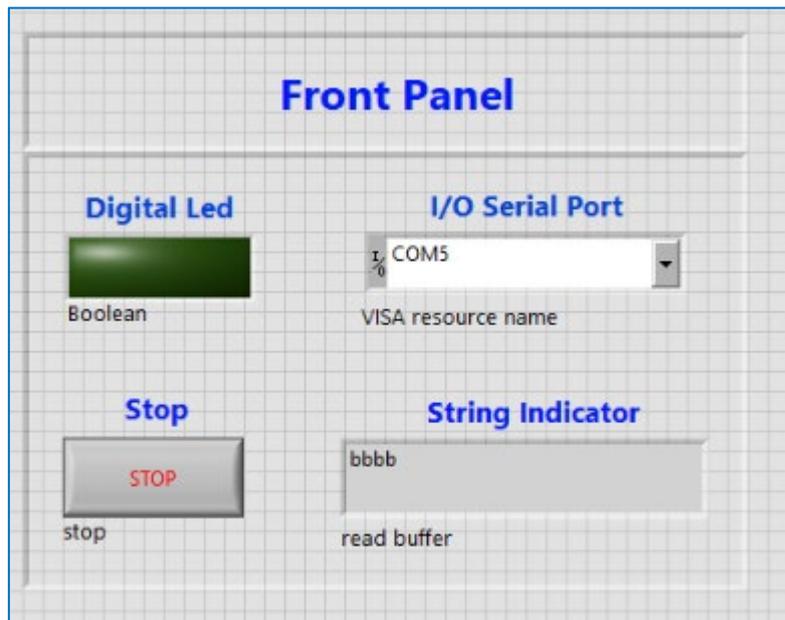


Figure 1. Front panel view

2.- On the Block Diagram: (Figure 2)

- 2.1 Configure de Serial Port by indicating the resource name and the baud rate.
- 2.2 Use the Property Node "Number of Bytes at Serial Port" to know the number of bytes currently available to read.
- 2.3 Read the data "VISA Read", display it in the "Read Buffer" and compare whether this data is equal to "a". If so, the LED will turn on.
- 2.4 Close the VISA.

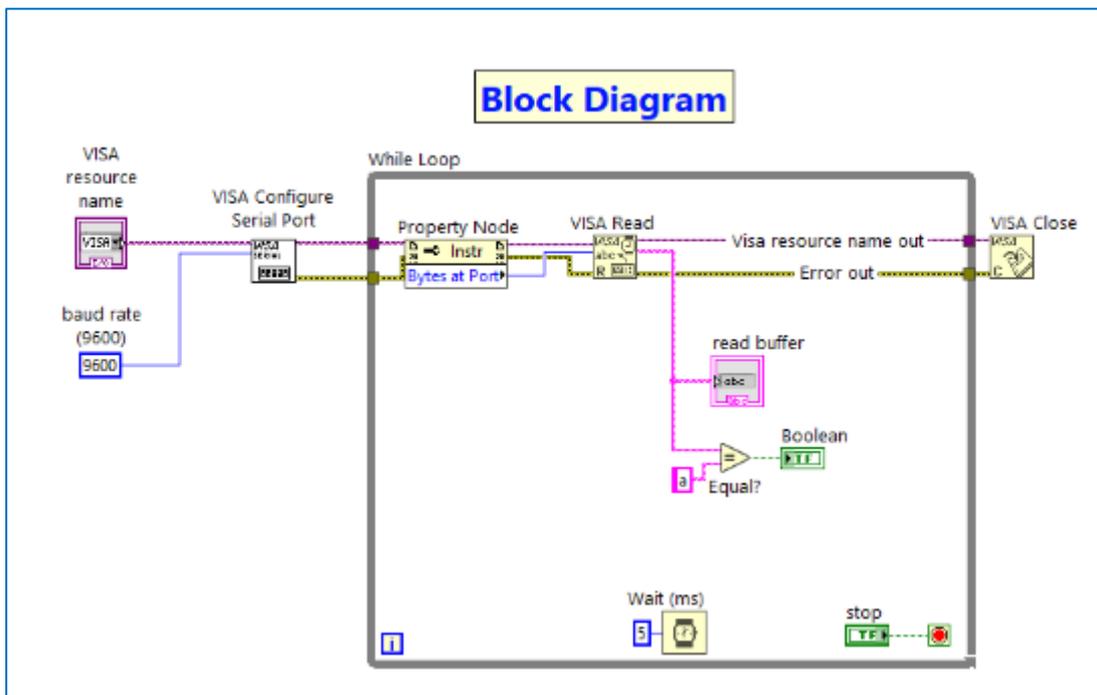


Figure 2. Block Diagram view

3.- Create a new Arduino Sketch to define the variables, the communication rate (be aware to use the same as in LabView Code), and the Main Code to send a variable according to the push-button input.

```

Archivo Editor Programa Herramientas Ayuda
20220323_VISA_Read_PushButton $

// Define variables
int pin8=8; //Output pin, variable

int variable; // create an integer variable

// Code to run only once
void setup()
{
  Serial.begin(9600); // communication rate 9600
  pinMode(pin8, INPUT); //set pin8 as an output
}

// Main code to run repeatedly
void loop()
{
  variable=digitalRead(pin8);
  if(variable==HIGH)
  { Serial.write('a');}
  else
  { Serial.write('b');}
}

```

Figure 3. Arduino Sketch

4.- Run the code to verify any errors that might have.



5.- Upload your code to the Arduino microcontroller.



6.- Connect the Arduino and protoboard as follows

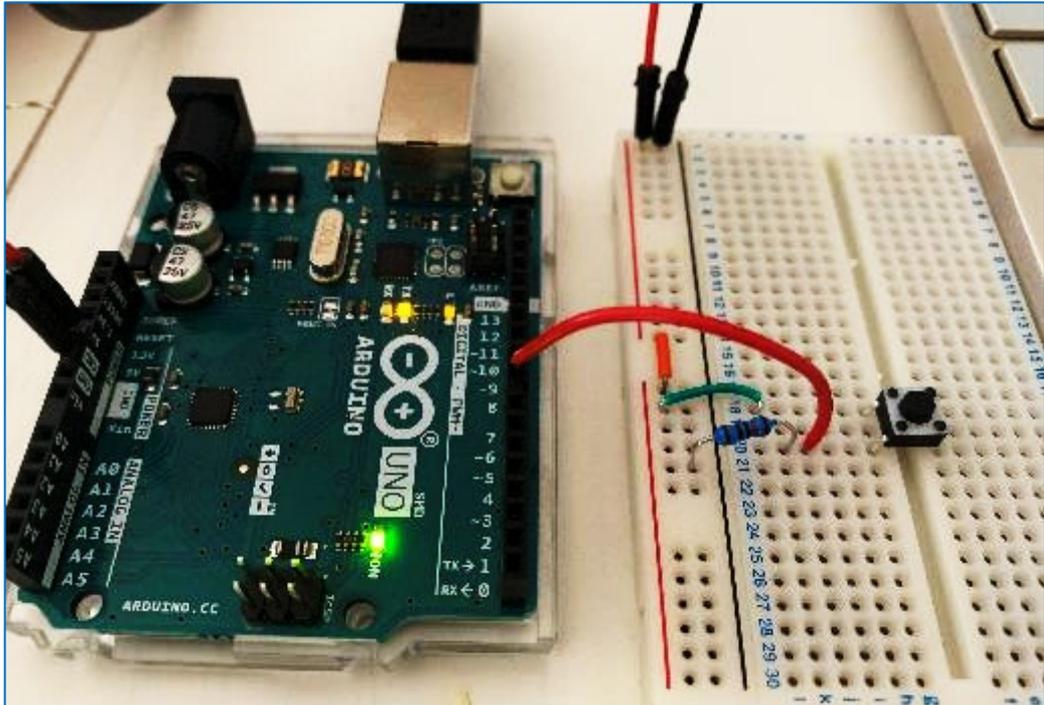


Figure 4. Hardware set-up

7.- Run the LabView program and see what happens to the Front Panel LED. ***Is it turning on when you press the button? Why is this happening?***

✓ Exercise 6.4

1.- Modify the previous software to detect whether the pushbutton is pressed “a” or not “b”. See the images below to carry out the relevant changes.

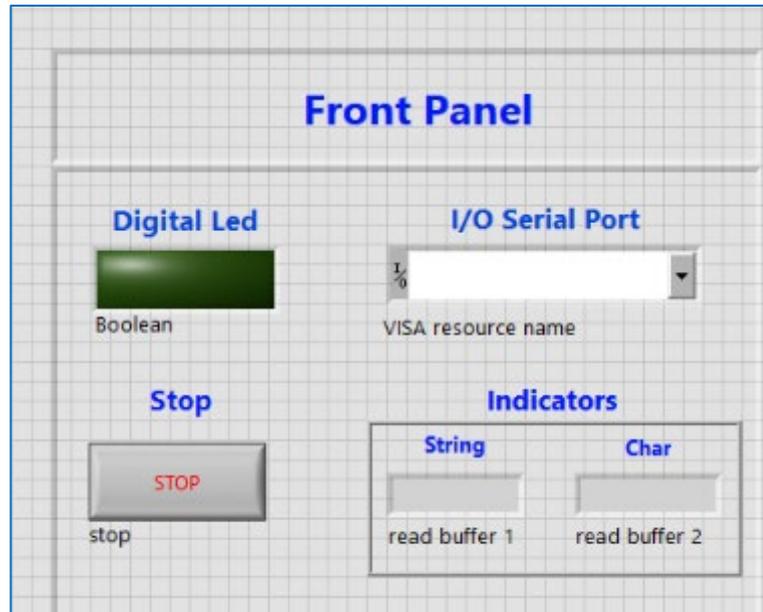


Figure 4. LabView Front Panel view

2.- Up to now, you are getting a string of characters on your String Indicator. Use the “String Subset” function (Figure 5) on your block diagram to get one string at the time (Char). Then, compare this result with the chart “a” to turn the LED on.

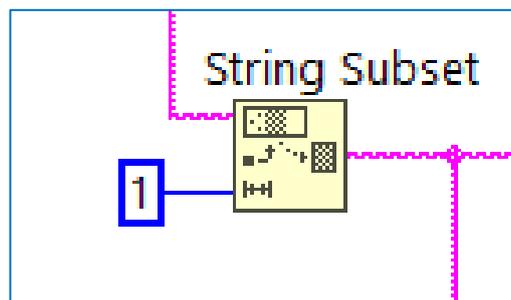


Figure 5. LabView String Subset icon

3.- Upload your VI to the “Campus Virtual”.

References:

- NI-VISA user manual: <https://www.ni.com/pdf/manuals/370423a.pdf>

- Arduino and LabView communication: Digital Inputs:

<https://www.youtube.com/watch?v=0ebDCdg-hlo&t=581s>