Proyecto Final Instrumentación Virtual

Table of Contents

1. Project description

Implementar un software que dinámicamente pueda lanzar los microcontroladores necesarios, en particular para el proyecto se requiere de 3, dos físicos y uno virtual o simulado.

Cada microcontrolador ** deberá operar en la modalidad Comando-respuesta, para lo cual deberá implementar en el mismo la lógica pertinente, los microcontroladores físicos (2 mínimo) deberán tener.

· Un canal analógico conectado a un sensor*

o Temperatura

o Iluminación

o Distancia

o Inductivo

o Capacitivo

o Etc.

· Un puerto digital de entrada con 8 bits (utilice dip switch o push buttons)

· Un puerto digital de salida con 8 bits (8 leds con sus resistencias de pull up o pull down)

2. Code component descriptions

2.1. DQMH® modules

This section describes DQMH® module responsibilities and relationships.

2.1.1. Preamble

A DQMH module is the main component of an architecture based on DQMH® framework. A DQMH module is used to implement a section of the application that has one responsibility.

DQMH® framework defines two different type of DQMH module.

Singleton:

A Singleton DQMH module can have only one instance running at any given time.

Cloneable:

A Cloneable DQMH module can have one or multiple instances running in parallel.

DQMH® framework defines two different ways to carry data throughout the application and with both other DQMH modules and non-DQMH based code.

Request events:

A request is a code that fires an event requesting the DQMH module to do something. Multiple locations in the code can send events to the DQMH module.

Request events are many-to-one.

Requests are usually named using imperative tense.

Broadcast events:

A broadcast is a code that fires an event broadcasting that the DQMH module did something. Multiple Event Structures can register to handle the Broadcast Events.

Broadcast Events are one-to-many.

Broadcasts are usually named using past tense or passive voice.

Refer to the DQMH® framework official documentation (http://delacor.com/documentation/dqmh-html/) to find more details on how the framework works

The following section gives you details on the project architecture relying on this framework. It gives you an overview of the modules' interaction and detailed information on each module.

Graphs used in this section have the following legend:

Components:

```
digraph G737339 {
rankdir=LR;
edge[dir=both color=black  arrowhead=normal arrowtail=none style=filled penwidth=1]
node[color=black shape=box]
"DQMH module / Lvlib"[color=black shape=component]
"Vi"[color=skyblue shape=note]
}
```

Events:

```
digraph G707844 {
rankdir=LR;
edge[dir=both color=black  arrowhead=normal arrowtail=none style=filled penwidth=1]
node[color=black shape=box]
" "[color=white shape=box]
"  "[color=white shape=box]
"   "[color=white shape=box]
"    "[color=white shape=box]
" " -> "  " [label="Request to a DQMH module" dir=both color=forestgreen  arrowhead=normal arrowtail=none
style=filled penwidth=1];
"   " -> "    " [label="Broadcast from a DQMH module" dir=both color=goldenrod  arrowhead=normal arrowtail=none
style=dashed penwidth=1];
}
```

Start and Stop module callers:

```
digraph G888319 {
rankdir=LR;
edge[dir=both color=black  arrowhead=normal arrowtail=none style=filled penwidth=1]
node[color=black shape=box]
"Start Module
caller"[color=black shape=component]
"Start Module"[color=yellowgreen shape=note]
"Start Module" -> "Start Module
caller" [label="Called by" dir=both color=yellowgreen  arrowhead=odot arrowtail=inv style=filled penwidth=1];
}
```

```
digraph G485279 {
rankdir=LR;
edge[dir=both color=black  arrowhead=normal arrowtail=none style=filled penwidth=1]
node[color=black shape=box]
"Stop Module
caller"[color=black shape=component]
"Stop Module"[color=tomato shape=note]
"Stop Module" -> "Stop Module
caller" [label="Called by" dir=both color=tomato  arrowhead=odot arrowtail=inv style=dotted penwidth=1];
}
```

2.1.2. Modules overview

This project contains the following modules.

Table 1. Modules list

Singleton	Cloneable

This graph represents the links between all DQMH modules.

```
digraph G401690 {
rankdir=LR;
edge[dir=both color=black  arrowhead=normal arrowtail=none style=filled penwidth=1]
node[color=black shape=box]
}
```

2.2. Libraries

This section describes the libraries contained in the project.

2.2.1. UI Actor.lvlib

No description found (add content in lvlib description)

2.3. Classes

This section describes the classes contained in the project.

2.3.1. Serial Device.lvclass

Serial Device es la clase padre de todos los dispositivos seriales, en el cual se encuentran los VI padre para leer el puerto, escribir el puerto, leer el pin, escribir el pin y leer un canal de ADC.

2.3.2. Arduino.lvclass

Clase hija ARDUINO, la cuál contiene overrides de todos los comandos del padre

2.3.3. NXP.lvclass

Clase hija de Serial Device capaz de conectarse a un microcontrolador NXP

2.3.4. Simulated.lvclass

VI hijo de Serial Device

2.3.5. MyTestCase_shido.lvclass

Unit Testing funciona ampliamente para reconocer algún error en tu código en una parte en específica, te ayuda a evitar futuros errores de código.

2.3.6. Close VISA Msg.lvclass

No description found (add content in lvlib description)

2.3.7. Close VISA_Msg.lvclass

Message class for Actor: UI Actor

2.3.8. VISA init Msg.lvclass

No description found (add content in lvlib description)

2.3.9. UI Actor.lvclass

UI Actor servirá para crear una interfaz agradable para el usuario, así como para crear una estructura HAL dentro del Actor configurado con herencias

3. VI descriptions

3.1. DQMH® modules

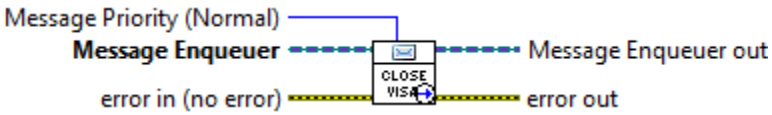
This section describes DQMH® modules events.

3.2. Libraries

This section describes libraries public VIs.

3.2.1. UI Actor.lvlib

UI Actor.lvlib:Close VISA Msg.lvclass:Send Close VISA.vi



Description:

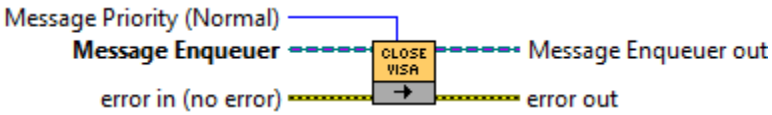
This VI sends the message to an actor.

UI Actor.lvlib:Close VISA Msg.lvclass:Do.vi

Description:

This VI delivers the message to the actor by calling the appropriate method(s) on the actor.

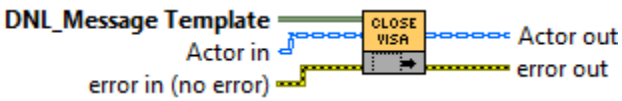
UI Actor.lvlib:Close VISA_Msg.lvclass:Send Close VISA.vi



Description:

This VI sends the message to an actor.

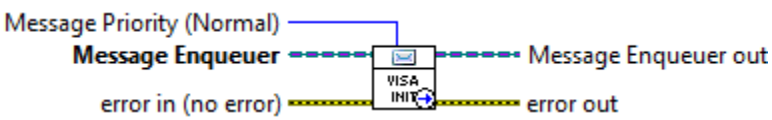
UI Actor.lvlib:Close VISA_Msg.lvclass:Do.vi



Description:

This VI delivers the message to the actor by calling the appropriate method(s) on the actor.

UI Actor.lvlib:VISA init Msg.lvclass:Send VISA init.vi



Description:

This VI sends the message to an actor.

UI Actor.lvlib:VISA init Msg.lvclass:Do.vi

Description:

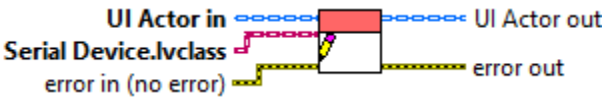
This VI delivers the message to the actor by calling the appropriate method(s) on the actor.

UI Actor.lvlib:UI Actor.lvclass:Read Serial Device.lvclass.vi

Description:

Te permite leer u obtener la marca del serial device que usarás en cada actor. Hacerlo de esta manera te permite tener acceso a este componente desde nodos de propiedad.

UI Actor.lvlib:UI Actor.lvclass:Write Serial Device.lvclass.vi



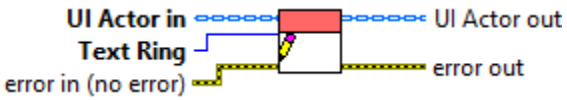
Description: No description found (add content in VI description)

UI Actor.lvlib:UI Actor.lvclass:Read Text Ring.vi

Description:

Te permite escribir o configurar la marca del serial device que usarás en cada actor. Hacerlo de esta manera te permite tener acceso a este componente desde nodos de propiedad.

UI Actor.lvlib:UI Actor.lvclass:Write Text Ring.vi



Description:

Te permite escribir o configurar la marca del serial device que usarás en cada actor. Hacerlo de esta manera te permite tener acceso a este componente desde nodos de propiedad.

UI Actor.lvlib:UI Actor.lvclass:Read VISA resource name.vi

Description:

Te permite obtener el puerto com designado a través de nodos de propiedad, así como configurarlos o cambiarlos con los mismos

UI Actor.lvlib:UI Actor.lvclass:Write VISA resource name.vi

Description:

Te permite obtener el puerto com designado a través de nodos de propiedad, así como configurarlos o cambiarlos con los mismos

UI Actor.lvlib:UI Actor.lvclass:Read VISA resource name 2.vi

Description: No description found (add content in VI description)

UI Actor.lvlib:UI Actor.lvclass:Write VISA resource name 2.vi

Description: No description found (add content in VI description)

UI Actor.lvlib:UI Actor.lvclass:Close VISA.vi

Description:

Close VISA te permite cerrar el puerto COM para un futuro uso con otras aplicaciones. Su mensaje se coloca en el stop core de actor core y te permite cerrar la comunicación antes de detener el debug

UI Actor.lvlib:UI Actor.lvclass:VISA init.vi

Description:

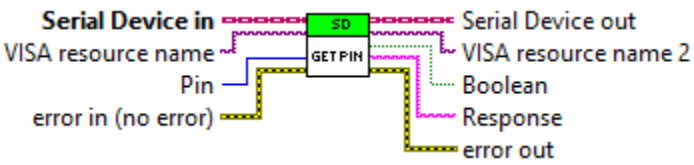
VISA init te permite inicializar tu puerto VISA antes de realizar cualquier operacion de lectura o escritura, el mensaje se utilizará en pre launch init para inicializar junto con el actor el puerto.

3.3. Classes

This section describes classes public VIs.

3.3.1. Serial Device.lvclass

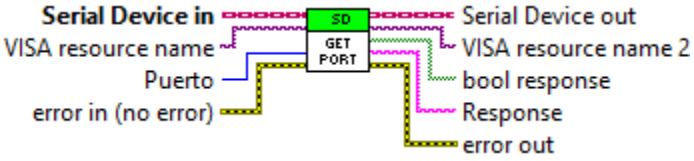
Serial Device.lvclass:Get Pin.vi



Description:

GET PIN VI. En este VI obtenemos el estado de el pin que el usuario desee dentro del microcontrolador elegido. Solo es posible obtener 4 pines de un puerto designado por el fabricante, que con facilidad puede ser modificado por el usuario

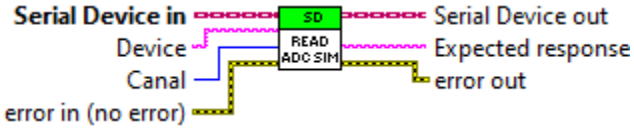
Serial Device.lvclass:Get Port.vi



Description:

Get Port VI. En este VI el usuario puede obtener el estado de un puerto completo, dicha respuesta se mostrará en el front panel con un arreglo de boleanos.

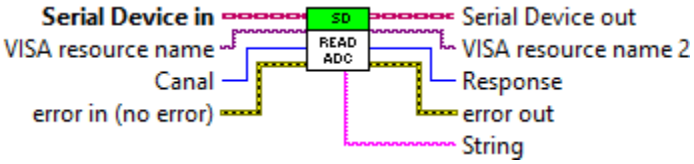
Serial Device.lvclass:Read ADC Sim.vi



Description:

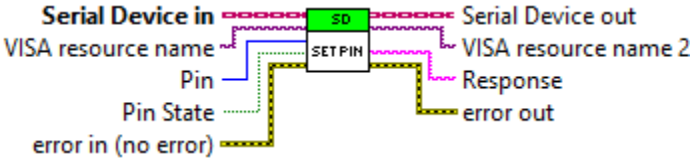
Read ADC VI. VI padre utilizado para mandar el comando de leer ADC y leerlo igualmente.

Serial Device.lvclass:Read ADC.vi



Description: No description found (add content in VI description)

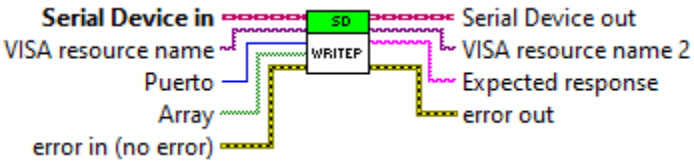
Serial Device.lvclass:Set Pin.vi



Description:

Set PIN VI. VI padre el cual dará a los hijos los parámetros para poder escribir en un pin del microcontrolador el valor deseado

Serial Device.lvclass:WritePort.vi

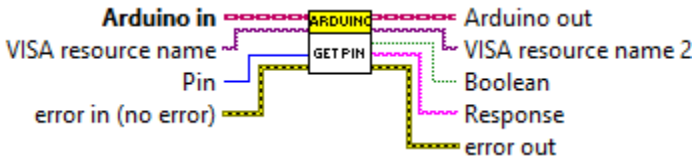


Description:

Write Port VI Vi padre con los parámetros para escribir un puerto completo del microcontrolador.

3.3.2. Arduino.lvclass

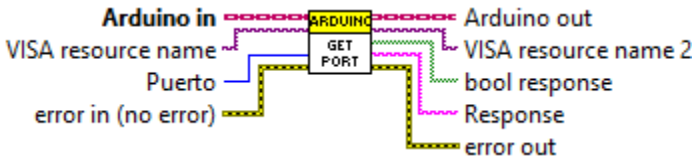
Arduino.lvclass:Get Pin.vi



Description:

Get Pin de Arduino Usando los parámetros establecidos en el Vi del papá, este VI obtiene uno de los 4 pines configurados en arduino, si esta presionado en la tarjeta me marca un true y si no me marca un false

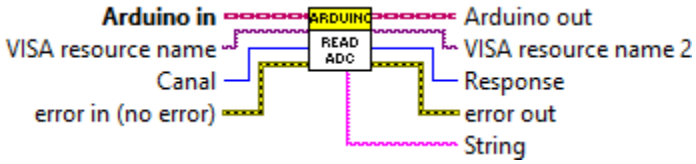
Arduino.lvclass:Get Port.vi



Description:

Arduino GET PORT. Lee un puerto de 4 bits de Arduino entregandomelo en un arreglo de boleanos en la interfaz.

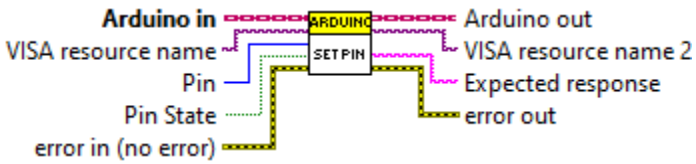
Arduino.lvclass:Read ADC.vi



Description:

Read ADC. Se encarga de mandar un comando por VISA port indicando que leerá el pin de ADC del microcontrolador, esperando 50 ms y obteniendo la respuesta al request.

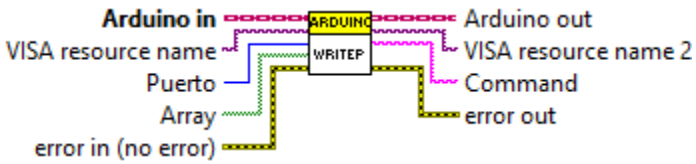
Arduino.lvclass:Set Pin.vi



Description:

Set Pin Arduino. Manda un comando indicando si quieres activar o desactivar el bit de un pin en específico en Arduino. Podrán escoger un pin del 1 al 4

Arduino.lvclass:WritePort.vi

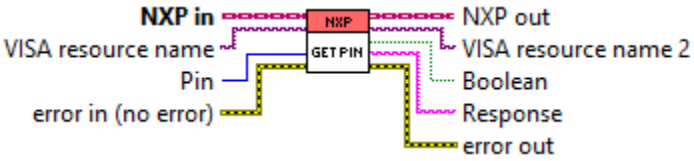


Description:

Write Port Arduino. Se encarga de escribir un puerto completo por VISA, mandando un arreglo de boleanos de modo serial y obteniendo el mismo arreglo en los leds de su arduino

3.3.3. NXP.lvclass

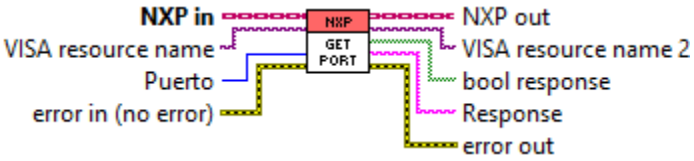
NXP.lvclass:Get Pin.vi



Description:

Get Pin de NXP Usando los parámetros establecidos en el Vi del papá, este VI obtiene uno de los 4 pines configurados en NXP, si esta presionado en la tarjeta me marca un true y si no me marca un false

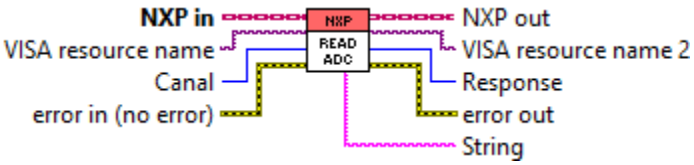
NXP.lvclass:Get Port.vi



Description:

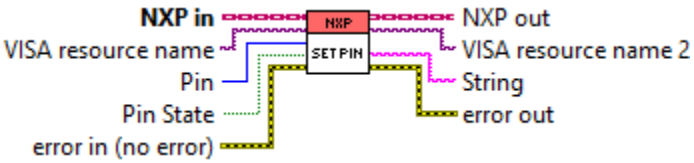
NXP GET PORT. Lee un puerto de 4 bits de NXP entregandomelo en un arreglo de boleanos en la interfaz.

NXP.lvclass:Read ADC.vi



Description: No description found (add content in VI description)

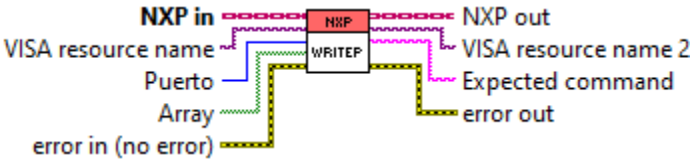
NXP.lvclass:Set Pin.vi



Description:

Set Pin NXP. Manda un comando indicando si quieres activar o desactivar el bit de un pin en específico en NXP. Podrán escoger un pin del 1 al 4.

NXP.lvclass:WritePort.vi

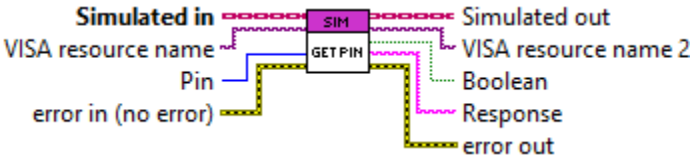


Description:

Write Port NXP. Se encarga de escribir un puerto completo por VISA, mandando un arreglo de boleanos de modo serial y obteniendo el mismo arreglo en los leds de su NXP

3.3.4. Simulated.lvclass

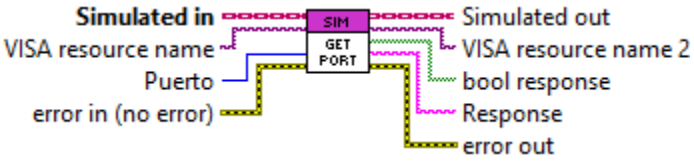
Simulated.lvclass:Get Pin.vi



Description:

Get Pin de un microcontrolador Simulado Te da una respuesta simulada en string de lo que estaría recibiendo labview de un microcontrolador fisico

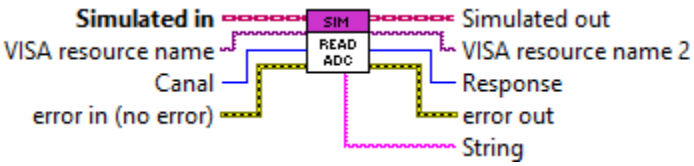
Simulated.lvclass:Get Port.vi



Description:

Get Port de un microcontrolador Simulado Te da una respuesta simulada en string de lo que estaría recibiendo labview de un microcontrolador fisico

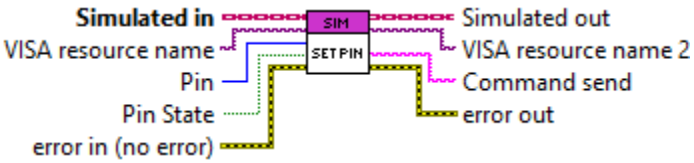
Simulated.lvclass:Read ADC.vi



Description:

Read ADC de un microcontrolador Simulado Te da una respuesta simulada en string de lo que estaría recibiendo labview de un microcontrolador fisico

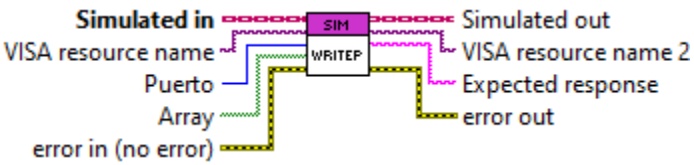
Simulated.lvclass:Set Pin.vi



Description:

Set Pin de un microcontrolador Simulado Te da una respuesta simulada en string de lo que estaría recibiendo labview de un microcontrolador fisico

Simulated.lvclass:WritePort.vi



Description:

Write Port de un microcontrolador Simulado Te da una respuesta simulada en string de lo que estaría recibiendo labview de un microcontrolador fisico

3.3.5. MyTestCase_shido.lvclass

MyTestCase_shido.lvclass:setUp.vi

Description:

setUp runs prior to the test method during test execution. Use this method to initialize any object data required by your tests.

MyTestCase_shido.lvclass:tearDown.vi

Description:

tearDown runs after the test method has completed. Use this method to clean up any operations or references that were opened by setUp or the test method. Unit tests should be independent of other unit tests so this VI should ensure that the next test can run in a 'clean' test environment.

MyTestCase_shido.lvclass:testExample.vit

Description: No description found (add content in VI description)

MyTestCase_shido.lvclass:temp_VI_UnderTest.vi

Description: No description found (add content in VI description)

MyTestCase_shido.lvclass:testExample_ADC.vi

Description:

Testea el ejemplo de leer adc de un dispositivo simulado en donde la respuesta esta previamente configurada.

MyTestCase_shido.lvclass:testExample _SX.vi

Description:

Tester de Set Port, el cual te permite mediante el microcontrolador simulado obtener una respuesta esperarda y corresta.

MyTestCase_shido.lvclass:testExample_SetPin.vi

Description: No description found (add content in VI description)

MyTestCase_shido.lvclass:testExample _GetPort.vi

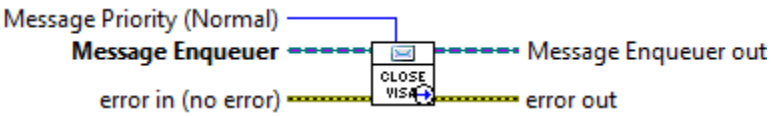
Description: No description found (add content in VI description)

MyTestCase_shido.lvclass:testExample_GetPin.vi

Description: No description found (add content in VI description)

3.3.6. Close VISA Msg.lvclass

UI Actor.lvlib:Close VISA Msg.lvclass:Send Close VISA.vi



Description:

This VI sends the message to an actor.

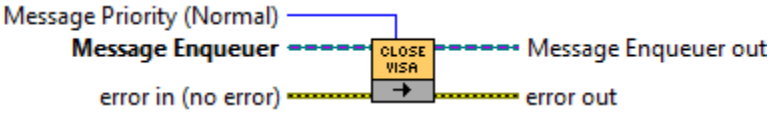
UI Actor.lvlib:Close VISA Msg.lvclass:Do.vi

Description:

This VI delivers the message to the actor by calling the appropriate method(s) on the actor.

3.3.7. Close VISA_Msg.lvclass

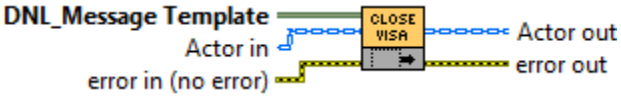
UI Actor.lvlib:Close VISA_Msg.lvclass:Send Close VISA.vi



Description:

This VI sends the message to an actor.

UI Actor.lvlib:Close VISA_Msg.lvclass:Do.vi

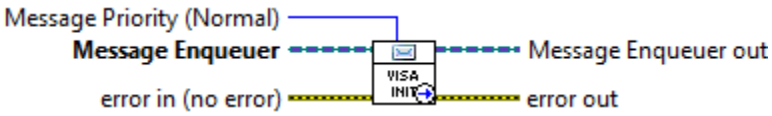


Description:

This VI delivers the message to the actor by calling the appropriate method(s) on the actor.

3.3.8. VISA init Msg.lvclass

UI Actor.lvlib:VISA init Msg.lvclass:Send VISA init.vi



Description:

This VI sends the message to an actor.

UI Actor.lvlib:VISA init Msg.lvclass:Do.vi

Description:

This VI delivers the message to the actor by calling the appropriate method(s) on the actor.

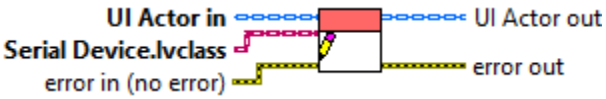
3.3.9. UI Actor.lvclass

UI Actor.lvlib:UI Actor.lvclass:Read Serial Device.lvclass.vi

Description:

Te permite leer u obtener la marca del serial device que usarás en cada actor. Hacerlo de esta manera te permite tener acceso a este componente desde nodos de propiedad.

UI Actor.lvlib:UI Actor.lvclass:Write Serial Device.lvclass.vi



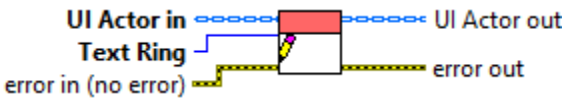
Description: No description found (add content in VI description)

UI Actor.lvlib:UI Actor.lvclass:Read Text Ring.vi

Description:

Te permite escribir o configurar la marca del serial device que usarás en cada actor. Hacerlo de esta manera te permite tener acceso a este componente desde nodos de propiedad.

UI Actor.lvlib:UI Actor.lvclass:Write Text Ring.vi



Description:

Te permite escribir o configurar la marca del serial device que usarás en cada actor. Hacerlo de esta manera te permite tener acceso a este componente desde nodos de propiedad.

UI Actor.lvlib:UI Actor.lvclass:Read VISA resource name.vi

Description:

Te permite obtener el puerto com designado a través de nodos de propiedad, así como configurarlos o cambiarlos con los mismos

UI Actor.lvlib:UI Actor.lvclass:Write VISA resource name.vi

Description:

Te permite obtener el puerto com designado a través de nodos de propiedad, así como configurarlos o cambiarlos con los mismos

UI Actor.lvlib:UI Actor.lvclass:Read VISA resource name 2.vi

Description: No description found (add content in VI description)

UI Actor.lvlib:UI Actor.lvclass:Write VISA resource name 2.vi

Description: No description found (add content in VI description)

UI Actor.lvlib:UI Actor.lvclass:Close VISA.vi

Description:

Close VISA te permite cerrar el puerto COM para un futuro uso con otras aplicaciones. Su mensaje se coloca en el stop core de actor core y te permite cerrar la comunicación antes de detener el debug

UI Actor.lvlib:UI Actor.lvclass:VISA init.vi

Description:

VISA init te permite inicializar tu puerto VISA antes de realizar cualquier operacion de lectura o escritura, el mensaje se utilizará en pre launch init para inicializar junto con el actor el puerto.

4. Legal Information

4.1. Document creation

This document has been generated using the following tools.

4.1.1. Antidoc

Project website: Antidoc (https://wovalab.gitlab.io/open-source/labview-doc-generator/)

Maintainer website: Wovalab (https://wovalab.com)

4.1.2. Asciidoc for LabVIEW™

Project website: Asciidoc toolkit (https://wovalab.gitlab.io/open-source/asciidoc-toolkit/)

Maintainer website: Wovalab (https://wovalab.com)

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

4.1.3. Graph Builder

Project website: Graph Builder (https://gitlab.com/cgambini/graph-builder)

BSD 3-Clause License

Copyright (c) 2020, Cyril GAMBINI All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

4.2. Product used in the project

Antidoc hasn’t been able to detect third party products in the project. This is the author’s responsibility to list any of the missing product used.