

Manual de Usuario

REMOTE-LAB-LIB

Interfaz FrontEnd-Arduino para laboratorio remoto

UNIVERSIDAD DE LA MARINA MERCANTE, CABA, BUENOS AIRES ARGENTINA

leandro.cintioli@alumnos.udemm.edu.ar
pablo.tavolaro@alumnos.udemm.edu.ar

Alumnos:

Leandro Martin Cintioli

Pablo Oscar Tavolaro Ortiz

Julio 2022

Práctica profesional supervisada.

Tutor interno: **Ing. Daniel Alejandro Ramondegui.**

Tutor externo: **Ing. Marcelo Bellotti.**

Universidad de la Marina Mercante
Facultad de Ingeniería

Sumario

Objetivo	3
Antecedentes	3
Uso - Operación de la librería Remote-Lab-Lib	3
Introducción	3
Comandos de comunicación	4
Seteo de la función del Puerto serie	5
Código del usuario dentro de la librería	6
Función “Log.msg”	6
Función “Memory.get”	7
Función “Memory.set”	7
Uso - Operación del Back- End Remote-Lab-Lib	8
Introducción	8
Estado y Comprobación	8
Uso - Operación del Front-End Remote-Lab-Lib	10
Introducción	10
Puesta en Funcionamiento	10
Descripción de solapas y funciones de cada una	12
Modificación de Nombres de parámetros	18
Referencia de Autor del Cliente	19

Objetivo

Describir el modo y características de uso - operación de la librería Remote-lab-lib y funcionalidades, como herramienta de conectividad para el desarrollo de unidades (a partir de aquí serán nombradas como experimento) en el marco del proyecto de Laboratorio Remoto desplegado en la Universidad.

Describir el modo y características de uso - operación del servidor Back End proporcionado como soporte para la interfaz Arduino - Cliente

Describir el modo y características de uso - operación del Cliente proporcionado como soporte para realizar pruebas de desarrollo - operación del proyecto

Antecedentes

La intención del proyecto es suministrar una interfaz entre una página web y una plataforma Arduino, que acepta los comandos para empezar y terminar el experimento, y almacena los valores de configuración en la memoria no volátil de la placa.

Los módulos principales de son tres: Cliente , Servidor y librerías Arduino.Ver Anexo.A del Manual RLL- instalación

Recomendamos el uso de este manual en forma complementaria con toda la documentación suministrada por este proyecto:

RLL-arquitectura

RLL-instalación

RLL-protocolo-comunicación

RLLServer-arquitectura

RLLClient-arquitectura

1. Uso - Operación de la librería Remote-Lab-Lib

1.1. Introducción

En el desarrollo de su experimento podrá contar con la conectividad y manejo de datos en la memoria no volátil de una unidad Arduino Mega.

La librería cuenta con varias Clases que se enumeran a continuación y que están descriptas en el manual RLL Arquitectura suministrado en la documentación:

Clase LOG

Clase Memory

Clase Param

Clase Led

La Librería posee comandos de comunicación tanto para leer o escribir valores en forma manual, configurar modos de uso y funciones del puerto serie de la placa arduino mega, y ordenar el comienzo o reseteo y finalización del experimento

1.2. Comandos de comunicación

A Través de la línea de comandos del monitor serie del programa arduino IDE, podrá enviar los comandos de tipo Json para realizar diferentes comunicaciones que le permitirán leer y escribir estados y resultados, los mismos se listan y se describe su argumentación en la siguiente tabla 1
Los comandos siempre se escriben entre llaves

Comandos	Argumentación
Json lectura	
{read:'all-params'}	Envía todos los parámetros en formato json.
{read:'all-input'}	Envía todos los parámetros de entrada.
{read:'all-output'}	Envía todos los parámetros de salida.
{read:'all-cfg'}	Envía todos los parámetros de configuración
{read:'all-lib'}	Envía todos los parámetros de configuración de remote Lab Lib
{read:'all-result'}	Envía todos los parámetros de resultados
{read:'version'}	Envía la versión del firmware.
{read:'status'}	Devuelve el estatus del ensayo.
{read:'serial_level'}	Nivel de seriales por puerto serie.
Json escritura	
{serial_level:'0'}	serial_level:0= Desactivado.
{serial_level:'1'}	serial_level:1= Impresión de string activado.
{serial_level:'2'}	serial_level:2= Impresión de excel activada..
{serial_level:'3'}	serial_level:3= Impresión para arduino plotter activada.
Json escritura	
inputx	{inputx:'250'} input0; input1;...input4 parametro de entrada tipo float
input0	{input0:'250'} //escribe 250 en input0
outputx	{outputx:'250'} output0; output1;...output4 parametro de salida tipo float
output0	{output0:'250'} //escribe 250 en input0
cfgx	{cfgx:'250'} parámetro de configuración tipo float
cfg0	{cfg0:'250'} //escribe 250 en cfg0
resultx	{resultx:'70.123'} parámetro float

result0	{result0:'70.123'} // escribe 70123 en result0
Ejecución del experimento	
{cmd:'start'}	comienza el ensayo escrito por el usuario
Estado de la Lib de ejecución	
st_test	
{st_test:'1'}	ensayo activado
{st_test:'0'}	ensayo desactivado
st_mode	modo de operación
{st_mode:'0'}	operación normal, ensayo activado
{st_mode:'100'}	Elige Experimento de prueba 1
{st_mode:'101'}	Elige Experimento de prueba 2
{st_mode:'102'}	Elige Experimento de prueba 3
{st_mode:'103'}	Elige Experimento de prueba 4
{st_mode:'200'}	Elige experimento de demo 1

tabla 1

en caso de dudas sobre los tipos de datos asignados a cada parámetro consulte el manual 'RLL-protocolo-comunicación-v0.00'

1.3. Seteo de la función del Puerto serie

Para poder utilizar la librería ud debe en primera instancia configurar el nivel de seteo del puerto serie:

- utilice el comando {read:'version'} para verificar la comunicación de su placa arduino mega conectada al puerto usb de su ordenador. Si no obtiene respuesta revise el manual de instalación 'RLL-instalación' suministrado en la documentación y verifique las condiciones del puerto COM y la velocidad configurada.
- utilice el comando {read:'serial_level'} para verificar el valor que presenta el nivel de comunicación del puerto serie:
 - si es 0 significa que la funciones del puerto serie para el monitor serie están desactivadas y la librería está configurada para un funcionamiento normal mediante el servidor back- front end
 - Si es 1 significa que la función de impresión a través del monitor serie, tanto de mensajes como de procesos, estará disponible.
 - Si es 2 significa que la función de impresión de valores vía Excel estará disponible.
 - Sí es 3 significa que la función de ploteo de Arduino estará disponible.

- Estos niveles de trabajo del puerto serie están pensados para que los utilice durante el periodo de desarrollo de su proyecto de laboratorio remoto, le será útil para realizar ensayos de dispositivos y análisis de comportamiento o debuggear. El nivel cero es el que permite la comunicación con el servidor y el modo de ejecución de opciones desde el front-end en forma remota.
- En caso de inquietudes revise el Anexo A del manual RLL-arquitectura suministrado en la documentación.

1.4. Código del usuario dentro de la librería

Ud deberá escribir su código o sketch en lenguaje C++ con las funciones de arduino dentro de la función “experiment”, teniendo en cuenta lo siguiente:

- Las funciones Serial.Print en cualquiera de sus posibilidades serán reemplazadas por llamadas a la función LOG_MS, deberá cambiar a nivel 1 para que esté disponible este modo LOG y sean visibles los mensajes y datos invocados por estas llamadas en el serial monitor del Programa editor IDE de arduino.
- Podrá modificar y obtener valores de memoria no volátil suministradas a través de las funciones “Memory.set” y “Memory.get” explicadas en su correspondiente apartado

1.5. Función “Log.msg”

- La función Log se utilizará para mostrar en pantalla mediante el monitor serie del programa arduino ide, mensajes o datos que sean de interés para el usuario en el proceso de desarrollo, configuración y testeо de su Experimento.
- EL Formato es:
`Log.msg(F("Ejemplo de prueba texto"));` ver ANEXO A.1 en manual de RLL-Arquitectura
- No solo se podrá utilizar para enviar mensajes de texto fijo sino que también se podrá enviar el valor de una variable que esté disponible dentro del programa y sea visible globalmente.
- Ejemplo en la sentencia:
`Log.msg(F("ST_LOOP= %d"), st_loop);`
 enviamos junto al texto “ST_LOOP” el valor de la variable “st_loop” especificando luego de la coma el nombre de la variable.
- Funciona de una manera similar al formato de la función Serial.print.

1.6. Función “Memory.get”

- La Función Memory.get nos permite leer de memoria no volátil un valor asignado a una dirección mediante un parámetro que le pasamos a la función.
- Las distintas funciones definidas en la librería del tipo Memory.get o las definidas en su apartado del tipo Memory.set tienen asociado un parámetro (que puede ser un número o un DEFINE) y son las siguientes:

```
float get_input(uint8_t param);  
void set_input(uint8_t param, float value);
```

```
float get_output(uint8_t param);  
void set_output(uint8_t param, float value);
```

```
float get_cfg(uint8_t param);  
void set_cfg(uint8_t param, float value);
```

```
float get_result(uint8_t param);  
void set_result(uint8_t param, float value);
```

- La estructura a utilizar en el uso de la función Memory.get sera:
Ejemplo:
`n_blinkF= Memory.get_cfg(INPUT0_MUL)`
en la sentencia anterior se asigna el valor leído de memoria de la posición asociada al parámetro “INPUT0_MUL” y se lo guarda en la variable “n_blinkF”
Otra posibilidad es utilizar la función dentro de otra de decisión o condicional.
- La configuración de los parámetros y sus nombres asociados se puede modificar en la Clase param.h, ver Anexo A.4 del manual RLL-arquitectura

1.7. Función “Memory.set”

- Al igual que las Funciones Memory.get, las Funciones Memory.set suministradas se seleccionan con un parámetro que define la dirección de memoria a utilizar en este caso para escribir o modificar un valor en ellas.
- Las mismas fueron enumeradas en el apartado anterior y se les debe suministrar el parámetro y el valor que se desea alojar en esa posición de memoria asociada al parámetro ya sea de inputs, outputs, valores de cfg o results.
Este valor suministrado podrá ser de una variable valor constante o el valor devuelto por otra función.

- Ejemplo:

```
Memory.set_output( 10, ( Memory.get_input( 1 ) + Memory.get_cfg( 11 ) ) + ( 2 * Memory.get_cfg( 66 ) ));
```

En la sentencia anterior vemos que le asignamos a la posición de memoria Output definido por el parámetro 10 el resultado de la operación matemática realizada con valores que devuelven las funciones Memory.get allí utilizadas.

2. Uso - Operación del Back- End Remote-Lab-Lib

2.1. Introducción

Para suministrar una interfaz entre una página web y una plataforma Arduino, que acepta los comandos para empezar y terminar el experimento, y almacena los valores de configuración en la memoria no volátil de la placa, se utiliza un Servidor Flask

Es un marco web de Python pequeño y liviano, proporciona herramientas y funciones útiles para crear aplicaciones web.

Se utiliza para gestionar las solicitudes web entrantes y enviar respuestas al usuario.

Básicamente el servidor cuando recibe peticiones http validas, ejecuta el código y devuelve la respuesta HTTP.

- La comunicación entre Cliente/Servidor se efectúa por Api-REST a través de FLASK
- La comunicación Servidor/Arduino a través de la clase serial

2.2. Estado y Comprobación

- El servidor levanta si detecta la Placa Arduino si no exhibirá un cartel como el siguiente:

```
C:\Users\Leandro>cd..
C:\Users>cd..

C:\>"C:\Users\Leandro\Documents\LC\PPS\back-end-remote-lab-lib-master\back-end-remote-lab-lib.py"
Back-end RemoteLab-lib
Version 2.0.02
Error al abrir puerto.
Programa abortado.

C:\>
```

- Realizar la configuración del puerto COM correcto modificando el programa back-end-remote-lab-lib.py en la linea 19, para más detalles ver el Manual RLL-instalación sección Servidor y Anexo A2
Una vez modificado y estando conectado la Placa Arduino Mega al puerto COM concordante, correr el programa back-end-remote-lab-lib.py en la ruta de acceso donde resida.

```
C:\>"C:\Users\Leandro\Documents\LC\PPS\back-end-remote-lab-lib-master\back-end-remote-lab-lib-master\back-end-remote-lab-lib.py"
Back-end RemoteLab-lib
Version 2.0.02
Conectado en windows
  * Serving Flask app 'back-end-remote-lab-lib'
  * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
  * Running on all addresses (0.0.0.0)
  * Running on http://127.0.0.1:4000
  * Running on http://192.168.1.42:4000
Press CTRL+C to quit
```

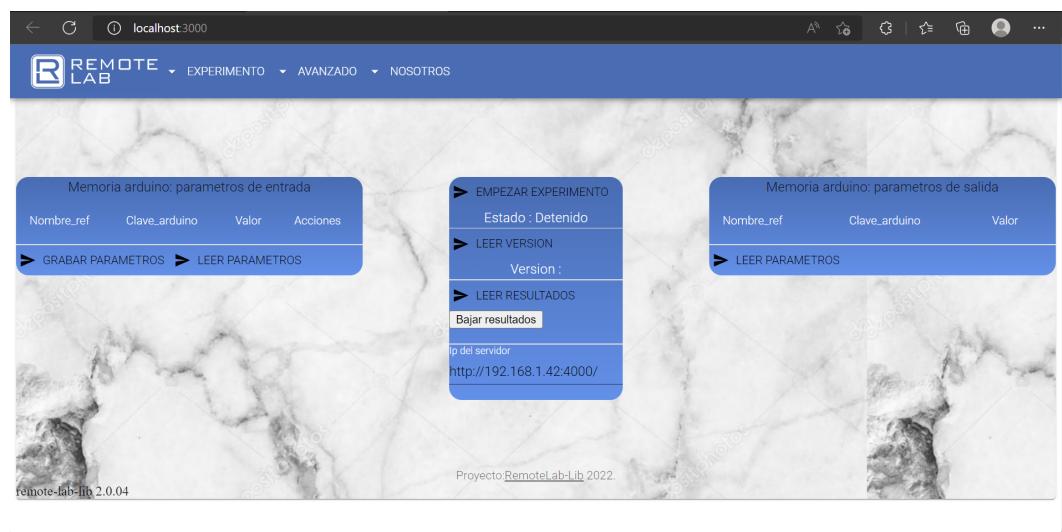
- Ahora el servidor se encuentra corriendo y está conectado al arduino Mega mediante el puerto serie.
En la última línea muestra la dirección de IP del servidor, para realizar las pruebas deberemos utilizar este dato para configurar en el Cliente Front-end-remote-lab-lib.py cuya sección es la próxima de este manual.

3. Uso - Operación del Front-End Remote-Lab-Lib

3.1. Introducción

El cliente proporciona una estructura similar a la que mostraría una interface de Cliente del sistema de Laboratorio remoto de la UdeMM.

El cliente está escrito sobre la librería React, para la parte gráfica se usó MUI. Con esto Ud se podrá realizar pruebas parciales y de funcionamiento con la cadena completa de comunicación emulando un entorno similar a través de la web



3.2. Puesta en Funcionamiento

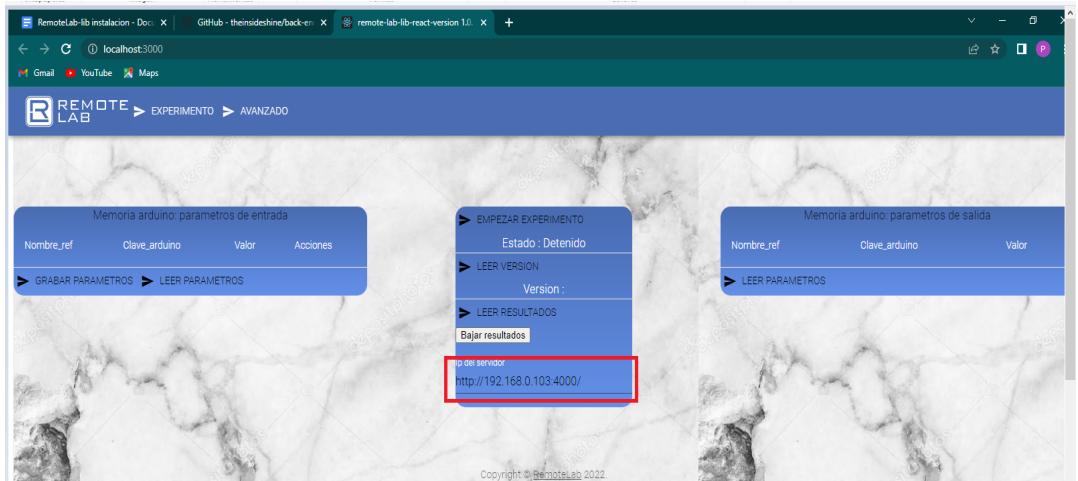
Para realizar la instalación remítase al Manual RLL-instalacion sección Cliente y Anexo A3.

- Configurar la ip en el cliente: **Este paso no debería ser necesario si configuro el archivo MemoryName.js , de no ser así puede configurarla , desde aca cada vez que levante el cliente.**

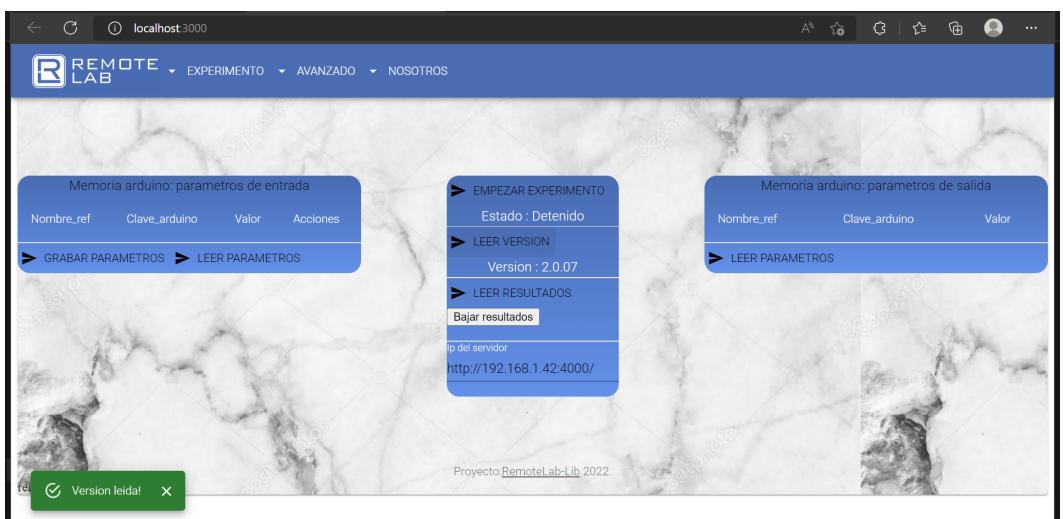
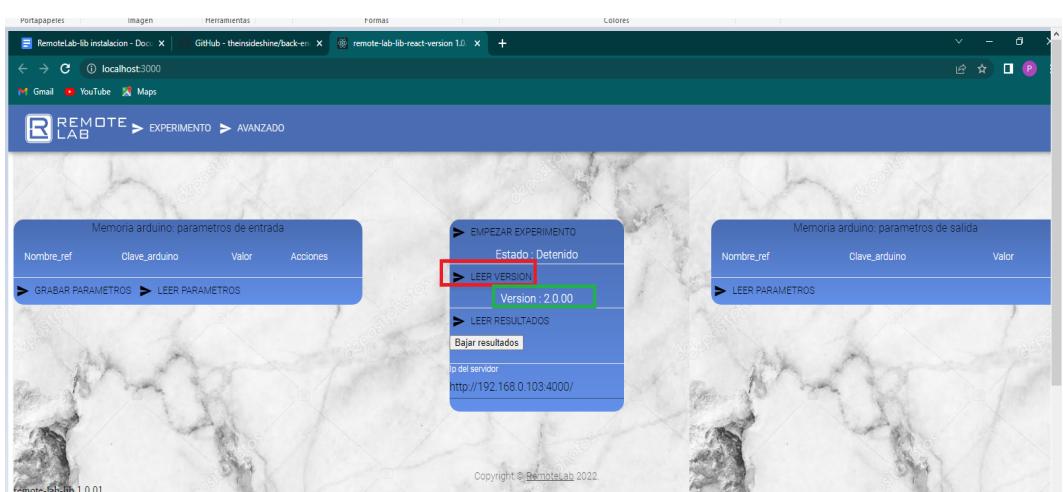
*Se marcará en cada captura con un recuadro o punto rojo el lugar donde hacer click o área de interés según sea relevante.

**Los nombres de los parámetros exhibidos son solo para ejemplo, la posibilidad de realizar edición de los mismos será mostrada en la siguiente sección.

En la siguiente imagen se muestra donde debe hacer click para configurar la Ip



- Para comprobar una comunicación inicial hacer click en el botón leer versión, se debe mostrar el número de versión de remote-lab-lib.

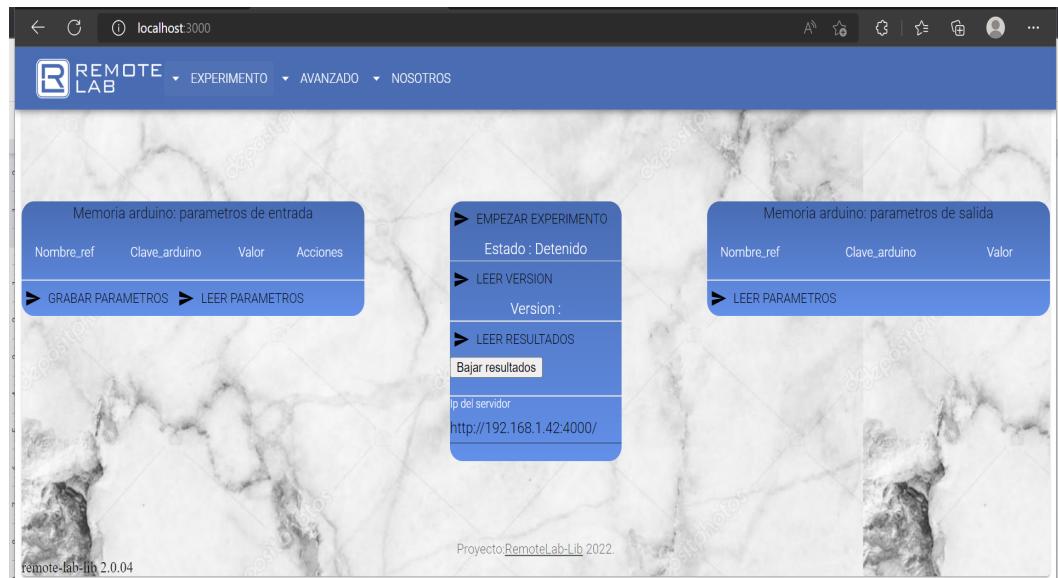


3.3. Descripción de solapas y funciones de cada una

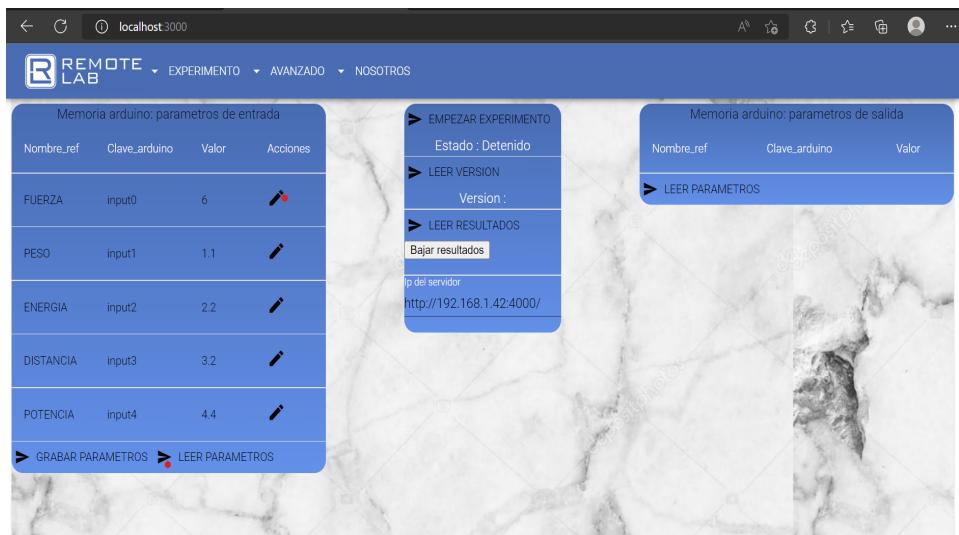
- Solapa EXPERIMENTO

Esta pantalla tiene la función principal del Cliente, desde allí podrá realizar la comunicación, obtener datos del Firmware y realizar el Experimento.

También desde aquí la interfaz le permite leer y grabar parámetros de entrada, leer parámetros de salida y resultados. También podrá bajar los resultados en una planilla de cálculo en formato EXCEL.



- Leer parámetros de entrada: le permitirá obtener los valores en la posiciones de memoria de la placa arduino llamadas “Inputs”



- Editar parámetro de entrada: le permitirá cambiar los valores de memoria de la placa arduino llamadas "Inputs"

Nombre_ref	Clave_arduino	Valor	Acciones
FUERZA	input0	6	
PESO	input1	1.1	
ENERGIA	input2	2.2	
DISTANCIA	input3	3.2	
POTENCIA	input4	4.4	

> GRABAR PARAMETROS > LEER PARAMETROS

una vez realizados los cambios deberá grabarlos en la memoria de la placa arduino:

- Grabar parámetros de entrada

Nombre_ref	Clave_arduino	Valor	Acciones
FUERZA	input0	6	
PESO	input1	1.1	
ENERGIA	input2	2.2	
DISTANCIA	input3	3.2	
POTENCIA	input4	4.4	

> GRABAR PARAMETROS > LEER PARAMETROS

Nombre_ref	Clave_arduino	Valor	Acciones
FUERZA	input0	6	
PESO	input1	1.1	
ENERGIA	input2	2.2	
DISTANCIA	input3	3.2	
POTENCIA	input4	4.4	

> GRABAR PARAMETROS > LEER PARAMETROS

Entradas guardadas!

- Leer parámetros de salida: le permitirá obtener los valores en la posiciones de memoria de la placa arduino llamadas “Outputs”

The screenshot shows the RemoteLab web interface. On the left, there is a table titled "Memoria arduino: parametros de entrada" with columns: Nombre_ref, Clave_arduino, Valor, and Acciones. The table contains five rows: FUERZA (input0, 6), PESO (input1, 1.1), ENERGIA (input2, 2.2), DISTANCIA (input3, 3.2), and POTENCIA (input4, 4.4). Below the table are two buttons: "GRABAR PARAMETROS" and "LEER PARAMETROS". On the right, a sidebar displays experiment controls: "EMPEZAR EXPERIMENTO" (Estado: Detenido), "LEER VERSION" (Version: 1.0), "LEER RESULTADOS" (Bajar resultados), and a link to "http://192.168.1.42:4000/".

This screenshot shows the same RemoteLab interface as the previous one, but with different data in the "Memoria arduino: parametros de salida" table. The table now lists six output parameters: ACCELERACION (output0, 1.825141), MILIMETROS (output1, 3.164145), AMPER (output2, 4.266854), TIEMPO (output3, 5.605857), and ANGULO (output4, 6.866097). A green button at the bottom left says "Salidas leidas!".

- Empezar Experimento: Comienza y ejecuta el Experimento a través de la placa Arduino, una vez finalizado dará aviso en pantalla.
Una vez finalizado se puede repetir las acciones de leer parámetros de salida, modificar o leer parámetros de entrada para un nueva instancia del experimento.

This screenshot highlights the "EMPEZAR EXPERIMENTO" button in the sidebar with a red rectangle. The button is labeled "Estado: Detenido". The sidebar also includes "LEER VERSION" (Version: 1.0), "LEER RESULTADOS" (Bajar resultados), and a link to "http://192.168.1.42:4000/".

Memoria arduino: parametros de entrada

Nombre_ref	Clave_arduino	Valor	Acciones
			GRABAR PARAMETROS > LEER PARAMETROS

Memoria arduino: parametros de salida

Nombre_ref	Clave_arduino	Valor	
			LEER PARAMETROS

Proyecto: RemoteLab-Lib 2022.

Experimento ejecutado!

■ Leer Resultados:

Es necesario aplicar el botón Leer resultados para consultar los valores actualizados a la Memoria de la Placa Arduino, para luego bajar los resultados.

Memoria arduino: parametros de entrada

Nombre_ref	Clave_arduino	Valor	Acciones
			GRABAR PARAMETROS > LEER PARAMETROS

Memoria arduino: parametros de salida

Nombre_ref	Clave_arduino	Valor	
			LEER PARAMETROS

Proyecto: RemoteLab-Lib 2022.

Resultados leidos!

■ Bajar resultados

localhost:3000

REMOTE LAB

Memoria arduino: parametros de entrada

Nombre_ref	Clave_arduino	Valor	Acciones
FUERZA	input0	6	editar
PESO	input1	1.1	editar
ENERGIA	input2	2.2	editar
DISTANCIA	input3	3.2	editar
POTENCIA	input4	4.4	editar

Memoria arduino: parametros de salida

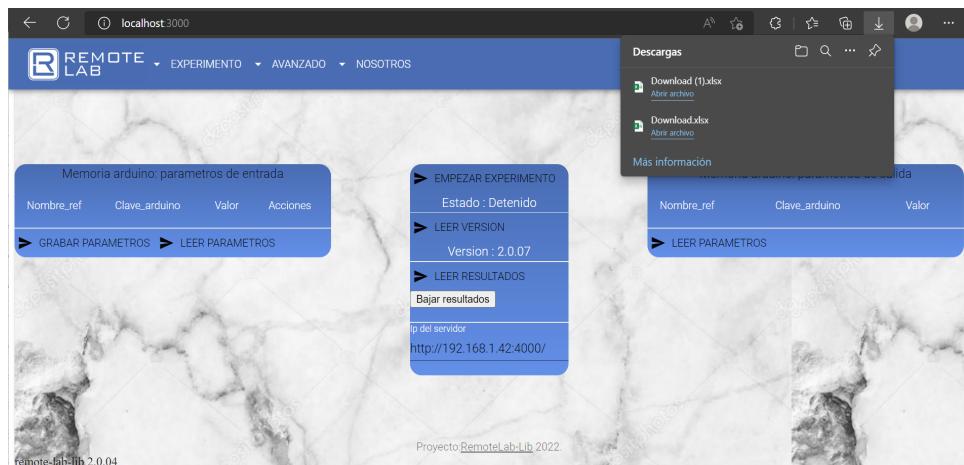
Nombre_ref	Clave_arduino	Valor
ACELERACION	output0	1.825141
MILIMETROS	output1	3.164145
AMPER	output2	4.266854
TIEMPO	output3	5.605857
ANGULO	output4	6.866097

EMPEZAR EXPERIMENTO
Estado : Detenido
LEER VERSION
Version :
LEER RESULTADOS
Bajar resultados
Ip del servidor
http://192.168.1.42:4000/

GRABAR PARAMETROS > LEER PARAMETROS

LEER PARAMETROS

Resultados bajados!

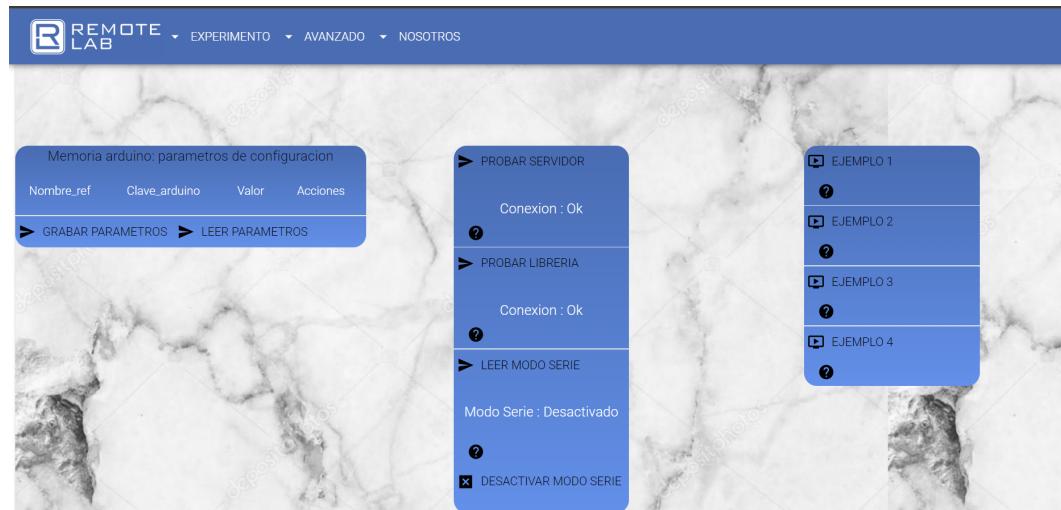


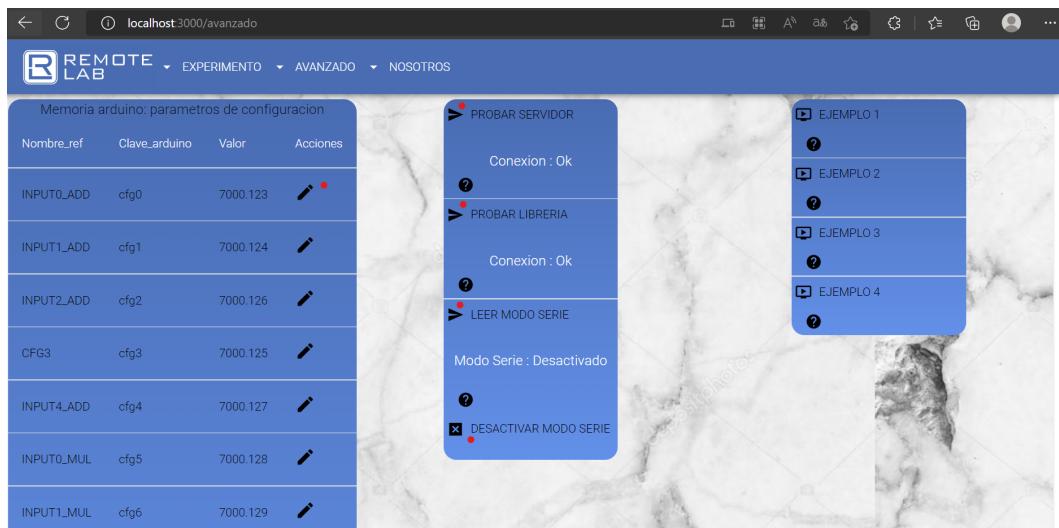
podrá encontrar el archivo en la carpeta de descargas que tenga configurada en el Browser en donde corre el cliente.

- **Solapa AVANZADO**

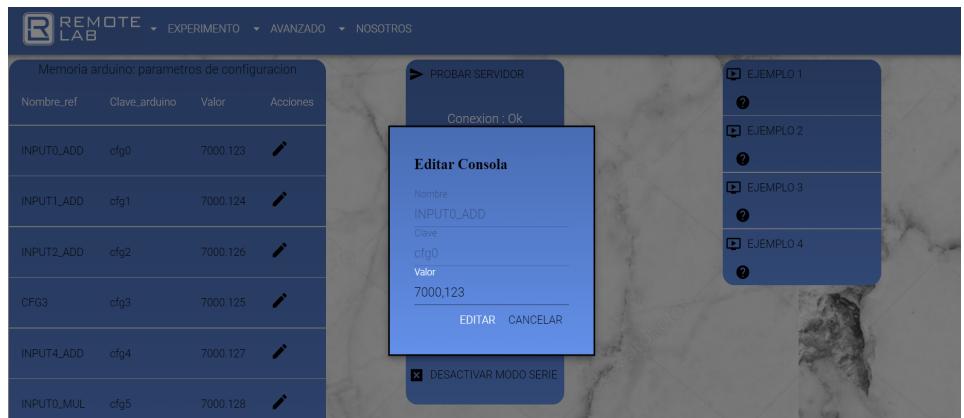
En esta solapa no solo tendrá disponible la posibilidad de leer y editar parámetros de configuración de la misma forma que se lee y edita los parámetros de Entrada en la solapa “Experimento”, sino que además podrá comprobar la conexión con el Servidor Back-end y con la librería para poder identificar cada línea de comunicación y su estado.

También podrá desactivar o activar el modo Serie para poder utilizar el monitor serie de Arduino sin tener que modificar por comando Json desde el propio monitor.



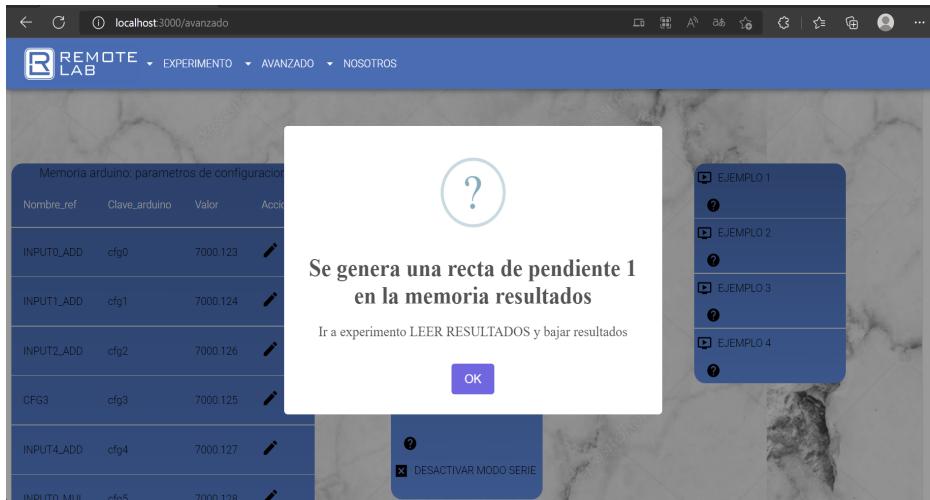


- Los botones “probar servidor”, “probar librería” y “leer modo serie” son informativos
- El botón “Desactivar modo serie” permite activar o desactivar modo serie para de forma inmediata poder utilizar el monitor serie del arduino IDE.
- El botón “leer parámetros” (configuración) tiene una función similar a su idem en la solapa Experimento tanto en entradas como salidas.
- El botón “editar” cada parámetro (configuración) permite variar el parámetro de manera similar a su idem en la solapa Experimento



- Botones de Ejemplo [1-4] son ejemplos de ejercicios que se pueden hacer incluidos en la librería RLL de arduino los cuales el cliente ejecuta en forma remota. Cada ejemplo tiene un botón “?” de ayuda que explica su funcionamiento.

El ejemplo 3:



Para más detalles de accionamiento ver el Manual RLLClient-arquitectura sección Anexo A

3.4. Modificación de Nombres de parámetros

Se puede realizar una modificación de los nombres de los parámetros en el Cliente para adecuarlos a los utilizados en el Experimento desarrollado con la librería.

Previo a la ejecución del Cliente debe modificar el **archivo MemoryName.js**

```
MemoryName: Bloc de notas
Archivo Editar Ver

export const memNameInputs =
{
    input0: 'mi I-parametro 1',
    input1: 'mi I-parametro 2',
    input2: 'ENERGIA',
    input3: 'DISTANCIA',
    input4: 'POTENCIA'
};

export const memNameOutputs =
{
    output0: 'mi O-parametro 1',
    output1: 'mi O-parametro 2',
    output2: 'AMPER',
    output3: 'TIEMPO',
    output4: 'ANGULO'
};
```

En este caso se modificaron los nombres de los 2 primeros parámetros de salida y entrada

The screenshot shows the 'REMOTE LAB' web application interface. On the left, there's a table titled 'Memoria arduino: parametros de entrada' with columns: Nombre_ref, Clave_arduino, Valor, and Acciones. It lists five parameters: 'mi I-parametro 1' (input0, 6), 'mi I-parametro 2' (input1, 1.1), 'ENERGIA' (input2, 2.2), 'DISTANCIA' (input3, 3.2), and 'POTENCIA' (input4, 4.4). The 'mi I-parametro 1' row has a red dot next to it. Below the table are buttons for 'GRABAR PARAMETROS' and 'LEER PARAMETROS'. In the center, a modal window titled 'EMPEZAR EXPERIMENTO' shows the current state: 'Estado : Detenido', 'Version :', and a button 'Bajar resultados'. At the bottom of this modal is the URL 'http://192.168.1.42:4000/'. On the right, another table titled 'Memoria arduino: parametros de salida' shows values for output pins: 'mi O-parametro 1' (output0, 0.328607), 'mi O-parametro 2' (output1, 0.328607), 'AMPER' (output2, 0.328607), 'TIEMPO' (output3, 0.249842), and 'ANGULO' (output4, 8.99275). A button 'LEER PARAMETROS' is at the bottom of this table.

Como se observa en la figura anterior la modificación se realiza en el entorno del cliente ni bien el programa vuelva a ser ejecutado

3.5. Referencia de Autor del Cliente

The screenshot shows the 'REMOTE LAB' web application interface. The browser title bar says 'remote-lab-lib-react-version 2.0'. The address bar shows 'localhost:3000/about'. The main content area has a blue header 'Interfaz Cliente-Arduino para laboratorio remoto'. The text describes the project's purpose: providing a remote lab interface between a web page and an Arduino platform, accepting commands to start and stop experiments, and storing configuration values in non-volatile memory. It mentions three main modules: Client, Server, and Arduino libraries. It provides links to GitHub repositories: 'Repository' (https://github.com/theinsideshine/frontend-remote-lab-lib, https://github.com/theinsideshine/back-end-remote-lab-lib, https://github.com/theinsideshine/remote-lab-lib). It also lists 'Contactos': LC: leandro.cintioli@alumnos.udemm.edu.ar and PT: pablo.tavolaro@alumnos.udemm.edu.ar.