

Lorby “Axis And Ohs” Arduino Bridge Documentation

Version 1.03 - 19.08.2023

© 2021 Lorby-SI <http://www.lorby-si.com>

This bridge program can be used in two modes:

1. Plugin mode

Copy the exe and dll to

\Documents\LorbyAxisAndOhs Files\Plugins\ArduinoBridge

AAO will detect the Bridge and start it automatically when you connect to the sim. The bridge GUI can be accessed by clicking on the icon in the Windows task bar.

2. WebAPI mode

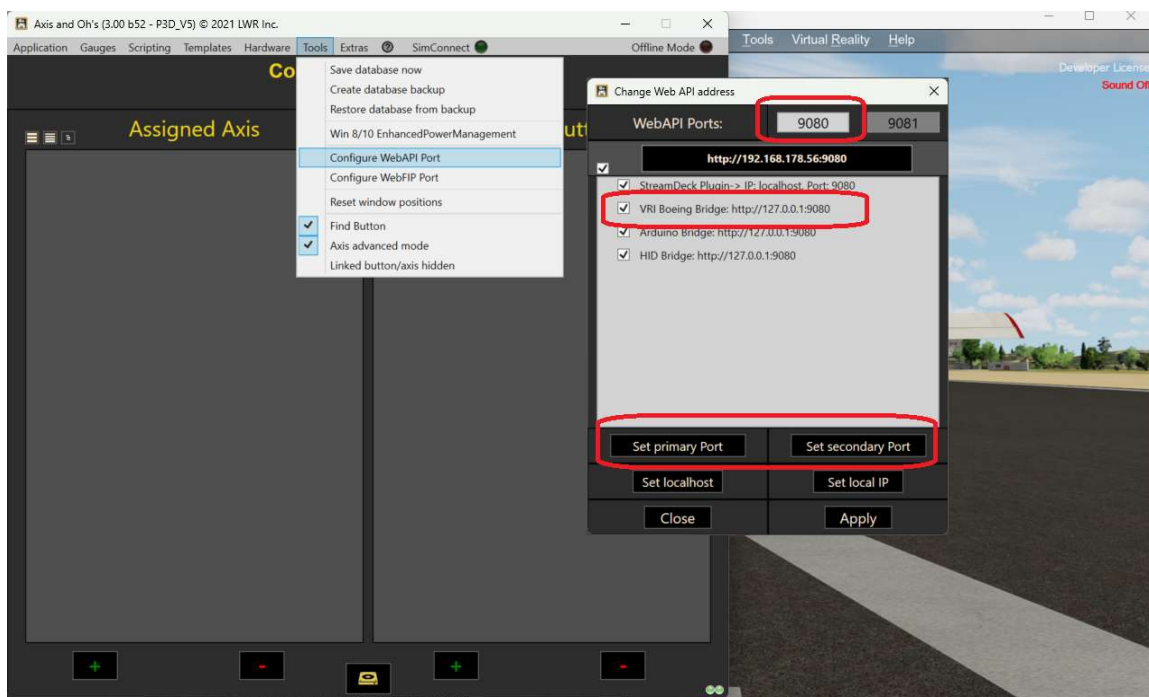
Copy the exe and dll to a location of your choosing. Doubleclick the exe to start the Bridge.

Make sure to activate the WebAPI first:

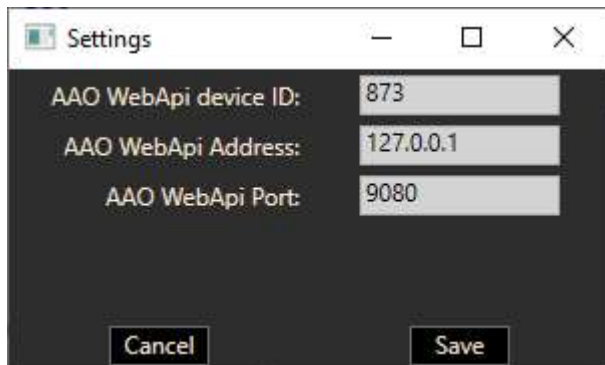
- AAO must be started “As Administrator”

You may have to open/allow access to the port in your local firewall. The port number can be changed with “Tools->Port settings for Web” in case there is a conflict with another app on your computer.

If you have another app on your computer that is already using the same port, it must be changed in the AAO Tools menu. You can choose either the primary or the secondary API port, to distribute the load depending on how many WebAPI clients you have running at the same time.



and in the settings dialog of the bridge (cogwheel button on the main dialog)



A screenshot of a 'Settings' dialog box. The dialog has a title bar with a green icon, the text 'Settings', and standard window controls (minimize, maximize, close). The main area has a dark background. It contains three labels with corresponding text input fields: 'AAO WebApi device ID:' with the value '873', 'AAO WebApi Address:' with the value '127.0.0.1', and 'AAO WebApi Port:' with the value '9080'. At the bottom, there are two buttons: 'Cancel' on the left and 'Save' on the right.

AAO WebApi device ID:	873
AAO WebApi Address:	127.0.0.1
AAO WebApi Port:	9080

Cancel Save

Compatibility with Arduino controllers

This bridge program can handle several AtMega32U4, AtMega2560 and AtMega328P boards at the same time, that are connected via USB to the computer.

The Bridge will try to flash its own controller Sketch on the Arduino board. If this fails (observed with Arduino “Every” mostly), you can flash the appropriate “.hex” file manually using “avrdude”.

The Bridge can handle several types of input and output:

Input

- Digital pin, with a value of 0/1
- Analog pin, with a value of 0-1023
- Rotary Encoder KY40, where every click is a button press in AAO

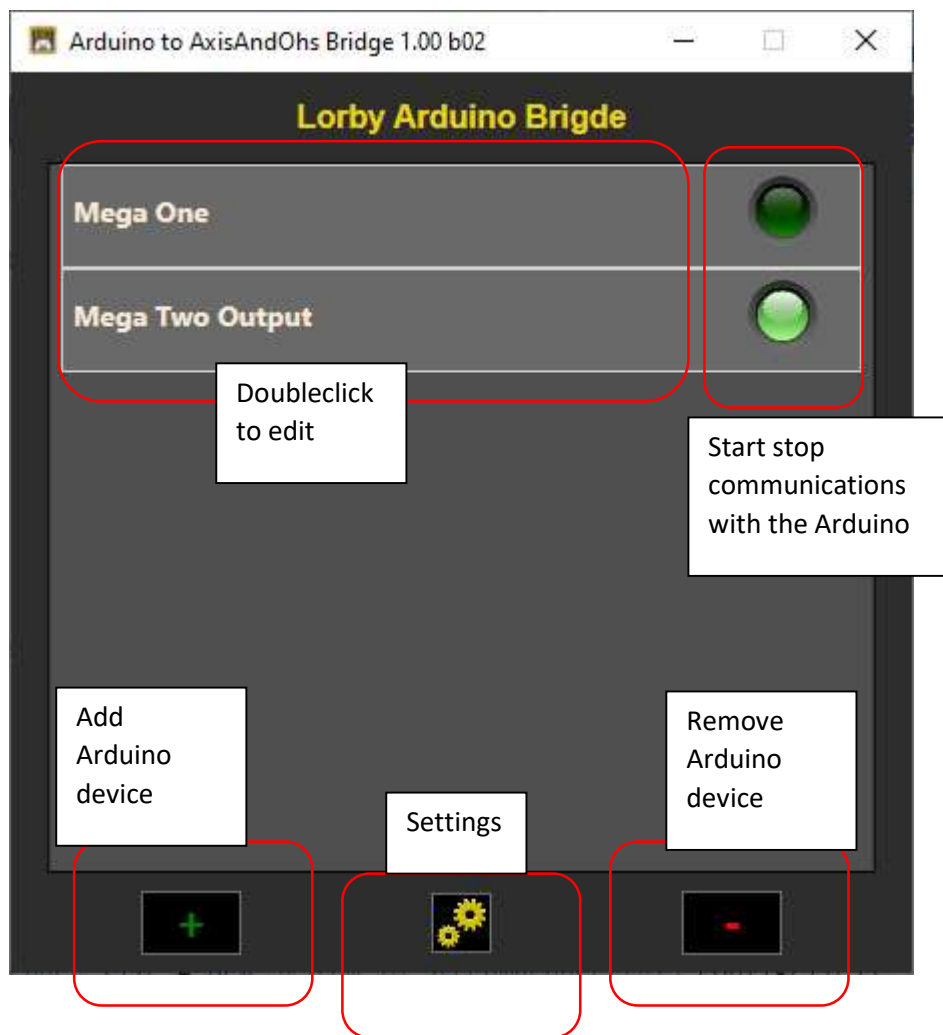
Output

- Single LED digital output
- Analog pin output
- LCD panels with IC2 serial board (1602, 1604, 2004 etc.)
- 7-segment LED numerical displays

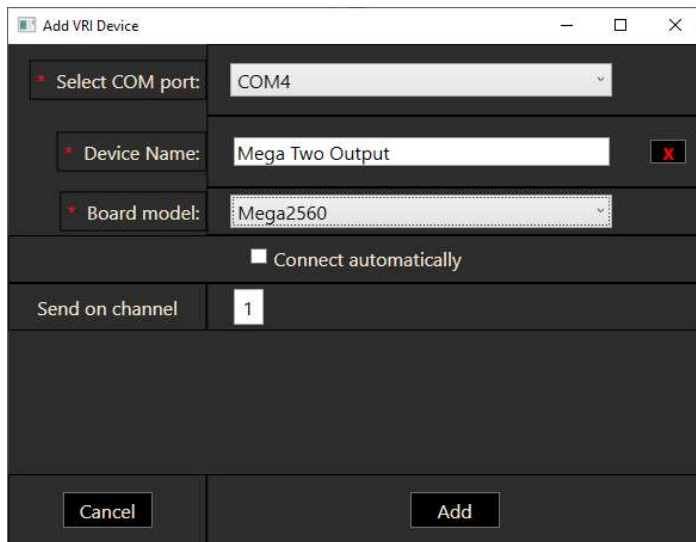
Please note that the Bridge program will load its own Sketch to the Arduino board to communicate with it!

The bridge doesn't have to be run on the same computer as AAO, the WebAPI can be accessed from any computer on your local network, provided that you open the communications port for them in your firewall(s).

Main Window



To add a device, click on the green “+”:

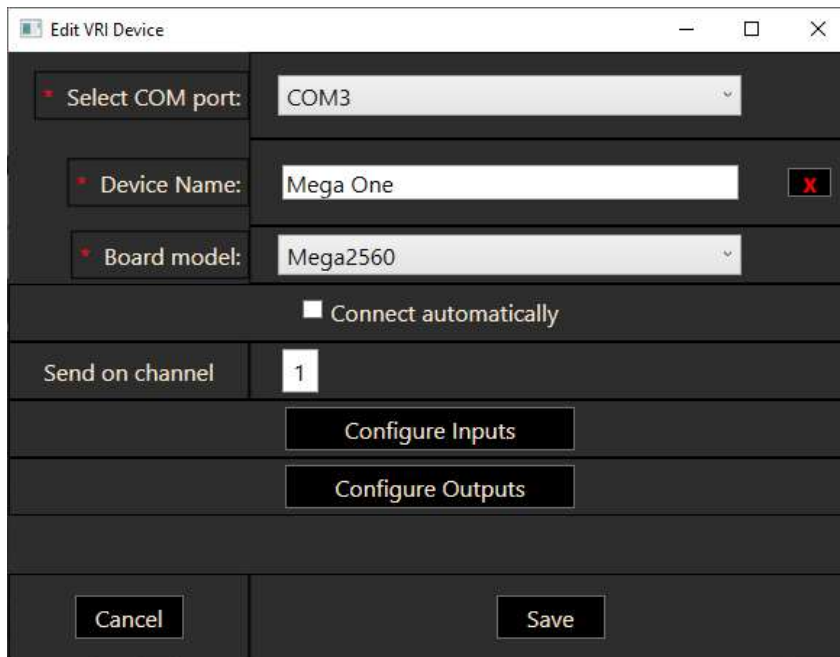


The screenshot shows a window titled "Add VRI Device". It contains the following fields and controls:

- Select COM port:** A dropdown menu with "COM4" selected.
- Device Name:** A text input field containing "Mega Two Output". To the right of the field is a small red square with a white "X".
- Board model:** A dropdown menu with "Mega2560" selected.
- Connect automatically:** A checkbox that is currently unchecked.
- Send on channel:** A text input field containing the number "1".
- Buttons:** "Cancel" and "Add" buttons at the bottom.

- Select the COM port that the Arduino is connected to
- Enter a name of your choosing so you know what it is
- Select the board model
The bridge currently supports the Mega 2560, Nano R3, Uno R3 and Leonardo/Micro boards
- Choose if the board should be connected automatically when the bridge is started
- Select a channel on which this board is sending data to AAO.
This number should be different for every board.

To edit the individual settings of a device, doubleclick it in the main list

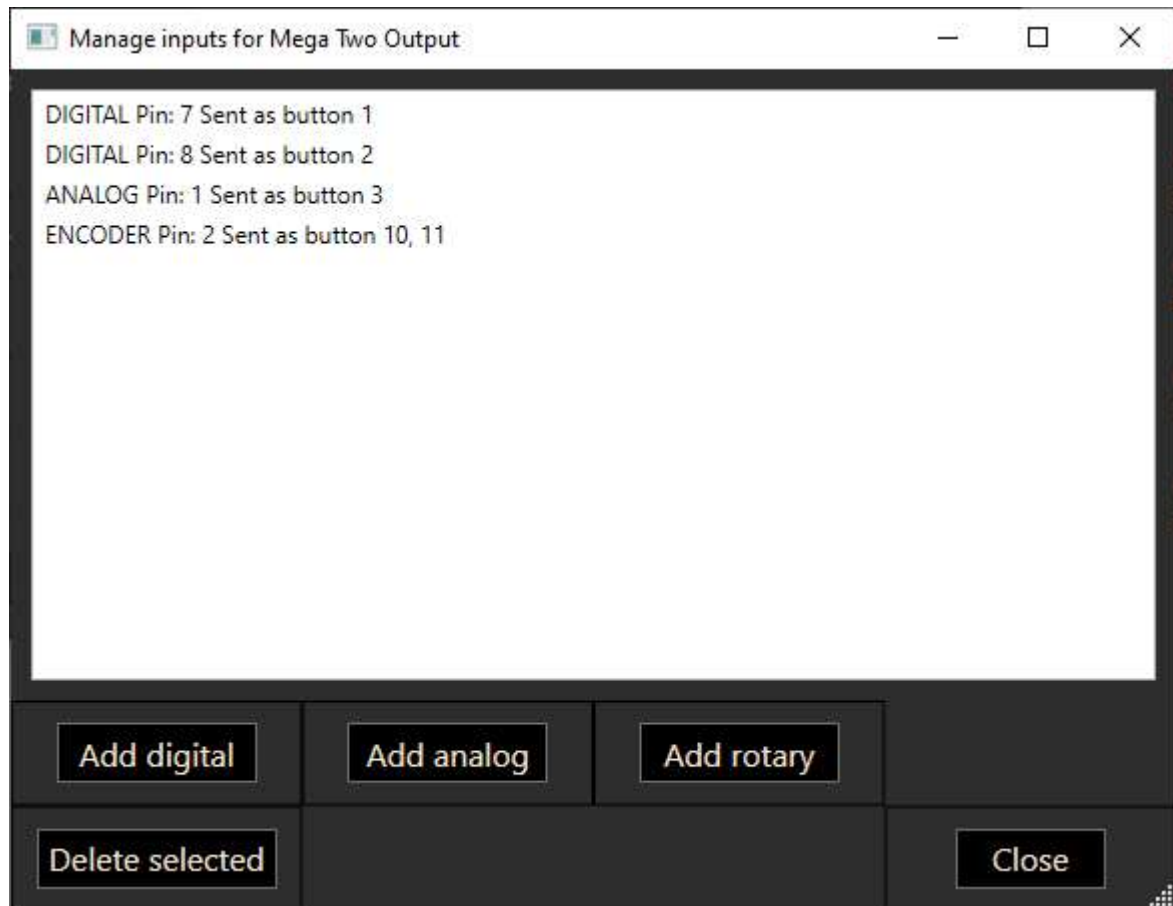


The screenshot shows a dialog box titled "Edit VRI Device" with a standard Windows window frame (minimize, maximize, close buttons). The dialog contains several configuration fields and buttons:

- Select COM port:** A dropdown menu currently showing "COM3".
- Device Name:** A text input field containing "Mega One". To the right of the field is a small red square button with a white "X".
- Board model:** A dropdown menu currently showing "Mega2560".
- Connect automatically:** A checkbox that is currently unchecked.
- Send on channel:** A text input field containing the number "1".
- Configure Inputs:** A button located below the "Send on channel" field.
- Configure Outputs:** A button located below the "Configure Inputs" button.
- Cancel:** A button at the bottom left of the dialog.
- Save:** A button at the bottom right of the dialog.

You can alter the parameters of the device here and you can configure what the board is sending to AAO (Inputs) or what should be displayed on the connected Output devices (LEDs, LCDs)

Configuring Inputs

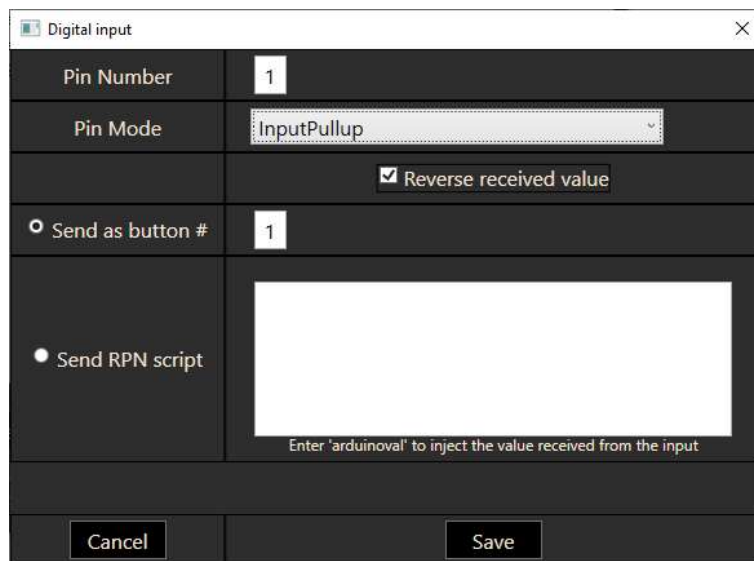


The bridge can handle three types of inputs:

- Digital PIN
- Analog PIN
- Rotary encode KY40, sending different button events upon left or right turn

To add an input, click on the appropriate button below the list. To edit an input, doubleclick it in the list.

Add digital input



Pin Number	1
Pin Mode	InputPullup
	<input checked="" type="checkbox"/> Reverse received value
<input type="radio"/> Send as button #	1
<input checked="" type="radio"/> Send RPN script	<div>Enter 'arduinova' to inject the value received from the input</div>
Cancel	Save

For a digital input choose the digital pin on your board that your button or switch is connected to. You can set the PIN mode to Input or InputPullup. If necessary, you can reverse the incoming value, if it doesn't trigger the AAO event correctly.

You have two options to send data to AAO:

1. As a button: select a unique numerical value to represent your button. In AAO these buttons will be received as MIDI input events on the Add/Change Button dialogs
2. Send an RPN script: you can enter any RPN script in this box that you want the bridge to send to AAO when you trigger the digital pin

Examples:

1 (>H:A320_Neo_PFD_BTN_LS_1, Number)

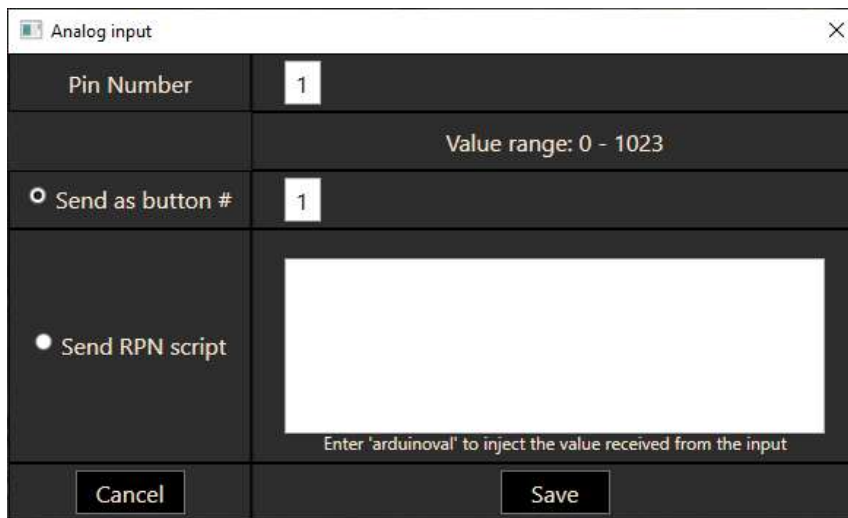
1 (>L:ASCRJ_FCP_ALT_CHANGE,·Number)

1 (>K:AUTOPILOT_MASTER)

if you want the script to use the actual value sent by the pin, use *arduinova* in the script

arduinova (>L: ASCRJ_FCP_ALT_CHANGE,·Number)

Add analog Input



The screenshot shows a dialog box titled "Analog input" with a close button (X) in the top right corner. The dialog is divided into several sections. The first section has a label "Pin Number" and a text input field containing the number "1". The second section has a label "Value range: 0 - 1023". The third section has two radio buttons: "Send as button #" (which is selected) and "Send RPN script". The "Send as button #" section has a text input field containing the number "1". The "Send RPN script" section has a large text area for entering a script. At the bottom of the dialog, there are two buttons: "Cancel" and "Save".

Pin Number	1
Value range: 0 - 1023	
<input type="radio"/> Send as button #	1
<input type="radio"/> Send RPN script	<div></div> <p>Enter 'arduinova1' to inject the value received from the input</p>
Cancel	Save

For the analog input choose the analog pin on your board that your potentiometer is connected to.

As before you have a choice to send the input as a button or use an RPN script.

The “button” would typically be used to control an Axis assignment in AAO, using the value range 0 – 1023.

When using a script instead, add “arduinova1” when you want the script to process the input value:

`arduinova1 (>L:AnLvarThatUsesThisValue,·Number)`

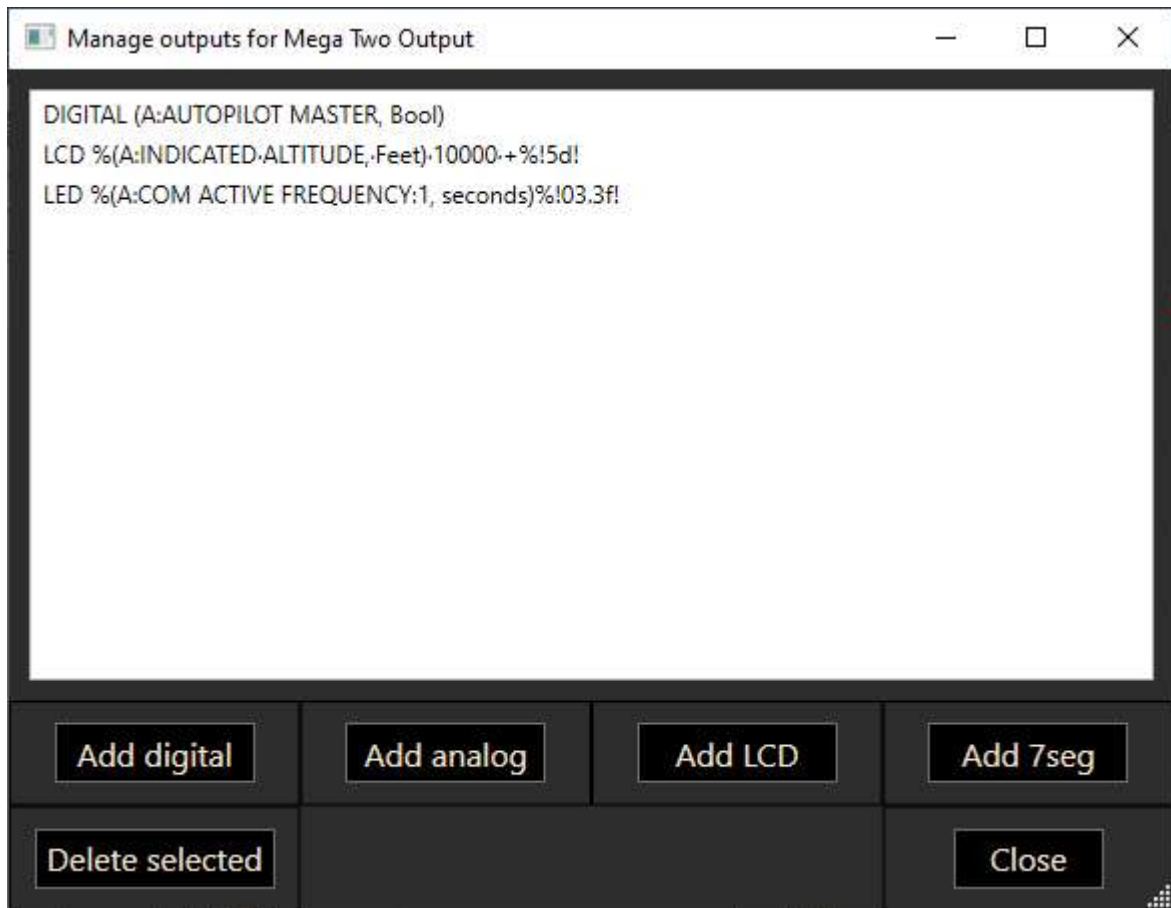
Add rotary Encoder

Rotary encoder input	
Pin CLK	4
Pin DT	3
Step type	Full steps
<input checked="" type="radio"/> Send right turn as button #	10
<input type="radio"/> Send RPN script	<div>Enter 'arduinoval' to inject the value received from the input</div>
<input checked="" type="radio"/> Send left turn as button #	11
<input type="radio"/> Send RPN script	<div>Enter 'arduinoval' to inject the value received from the input</div>
<div>Cancel Save</div>	

Rotary encoders are assumed to be of the KY40 type, with 4 or 5 pins. Two of the pins are +5V and GND, while the other two should be CLK and DT – and these PINs you enter in the dialog shown above. Depending on how your encoder works, you can choose to send 1, 2 or 4 events with every “click” of the encoder knob by selecting Full, Half or Quarter step size. If the encoder has fifth pin for a button action, that would be added as a normal digital input as shown above.

With the rotary encoder you have the same choice of events sent to AAO as above, configured separately for the left and right turn action.

Configuring outputs



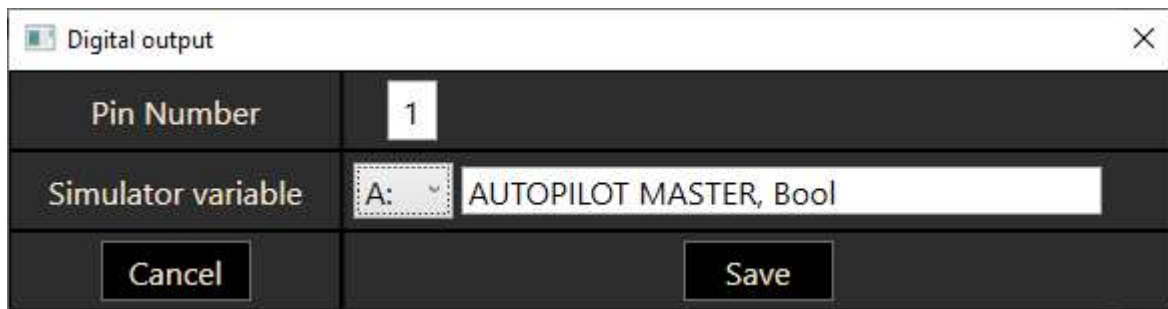
The bridge supports the following types of outputs:

- Output of 0/1 to a digital PIN
- Output of a value of 0-255 to an analog PIN
- Text output to the standard IC2 LCD screens and 7 segment LED devices

To add an output, click on the appropriate button below the list. To edit an output, doubleclick it in the list.

For performance reasons, it is best not to mix input and output configurations on the same device. Especially the LCD output is rather slow, and it might interfere with your input events that you wish to receive from the same Arduino board.

Add digital output



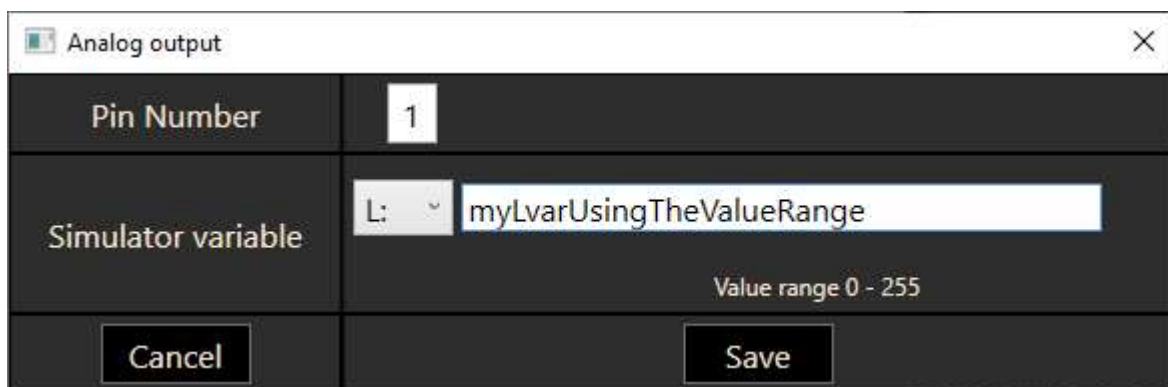
A screenshot of a 'Digital output' configuration window. It has a title bar with a close button. The window is divided into two main sections. The top section has a 'Pin Number' label and a text input field containing the number '1'. The bottom section has a 'Simulator variable' label and a dropdown menu showing 'A:' with a small arrow icon, followed by a text input field containing 'AUTOPILOT MASTER, Bool'. At the bottom of the window are two buttons: 'Cancel' on the left and 'Save' on the right.

Pin Number	1
Simulator variable	A: AUTOPILOT MASTER, Bool
<div>Cancel Save</div>	

The bridge will continuously sync the output pin with the result of the simulator variable query. You can use either A: or L: variables that return a value of 0 or “not 0” or supply an RPN script that yields a 0/1 result (selection “S:”). For example, the configuration above could also be set up as

“S:” -> “(A:AUTOPILOT MASTER, Bool)”

Add analog output



A screenshot of an 'Analog output' configuration window. It has a title bar with a close button. The window is divided into two main sections. The top section has a 'Pin Number' label and a text input field containing the number '1'. The bottom section has a 'Simulator variable' label and a dropdown menu showing 'L:' with a small arrow icon, followed by a text input field containing 'myLvarUsingTheValueRange'. Below the text input field, the text 'Value range 0 - 255' is displayed. At the bottom of the window are two buttons: 'Cancel' on the left and 'Save' on the right.

Pin Number	1
Simulator variable	L: myLvarUsingTheValueRange
Value range 0 - 255	
<div>Cancel Save</div>	

Same as above, the bridge will continuously sync the output pin with the result of the simulator variable query. This result should be in the 0-255 range, so most likely an LVar that you calculate elsewhere in RPN code.

Add LCD output

Standard LCD display with IC2 board - only one per device!

Columns: 16

Rows: 2

RPN Code

```
Alt: %(A:INDICATED-ALTITUDE,Feet)%!5d! ft
Speed: %(A:AIRSPEED-TRUE,Knots)%!3d! kn
```

Cancel Save

The LCD display with IC2 board is assumed to be connected as per spec on +5V, GND, SDA and SCL pins (A4 and A5 on the Nano).

You can select the size of the LCD output and configure the actual output values as RPN string code. Every line in the textbox represents a row on the display, every character position in the box equals a column.

%...%!..!	Formatting numbers, d = integer, f = floating point. Precede d with 0 or space for leading zeroes or spaces. %...% can contain RPN code and variables	%2%!02d! %12.34554%!4.2f!	02 12.35
-----------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------	-------------

Conditonal Strings are supported as follows:

```
%condition%{if}text{end}
```

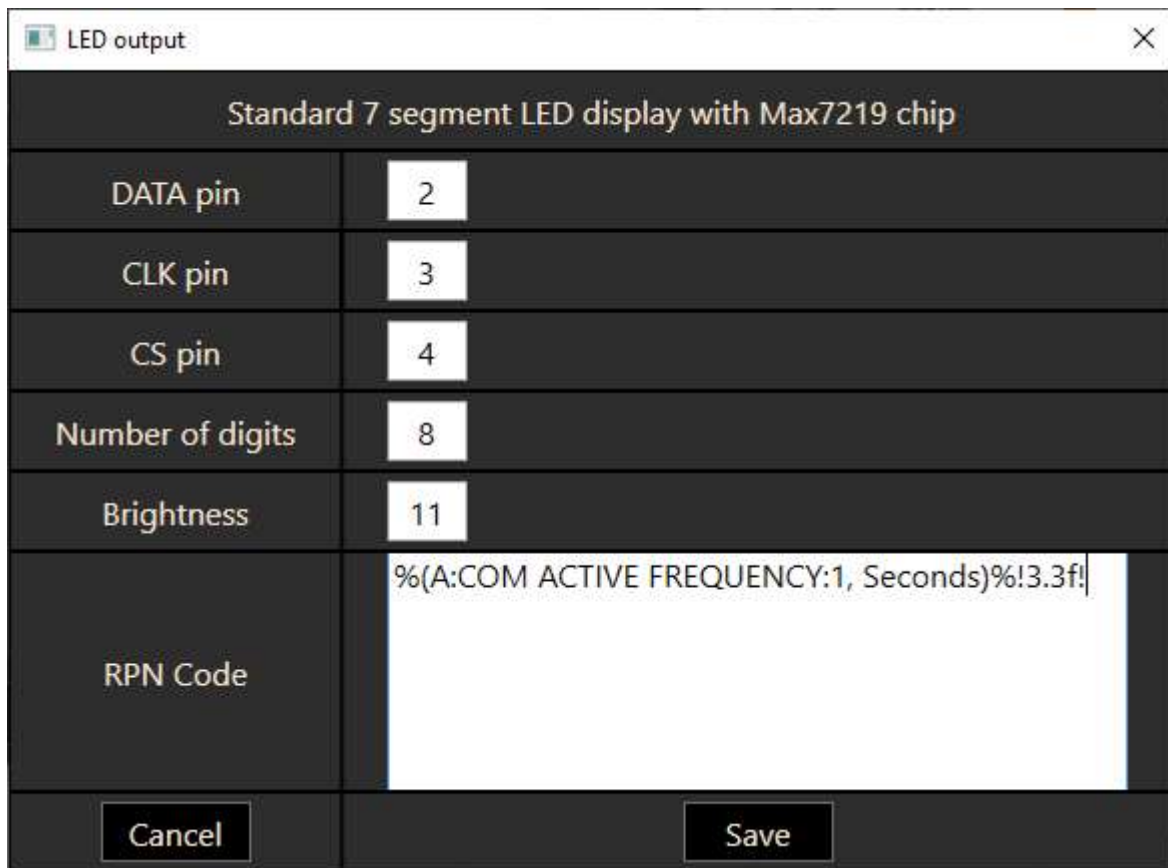
```
%condition%{if}text1{else}text2{end}
```

You can use {if} {end} inside other ifs.

```
%index to select%{case}{:0}text1{:1}text2{:2}text3{end}
```

{case} can have up to 99 {nn} index strings

Add 7 segment LED output



Standard 7 segment LED display with Max7219 chip	
DATA pin	2
CLK pin	3
CS pin	4
Number of digits	8
Brightness	11
RPN Code	<code>%(A:COM ACTIVE FREQUENCY:1, Seconds)%!3.3f!</code>
<div>Cancel Save</div>	

The LED output represents a single device in a serialized 7seg strip configuration. The devices are assumed to have Max7219 controllers. The “number of digits” parameter refers to the individual digits on a single 7seg strip (= usually 4 or 8)

If you have more than one strip, daisy-chained to each other, you will have to create an output configuration for each of them.

When the AAO variable/event lists are open, you can copy the name of the variable or event to the Windows scratchpad by left clicking then right clicking the item in the treelist. Then you can insert it into the textbox by pressing Ctrl&V

