YaDeLi
„Yet another Device Link for IL-2“

What *YaDeLi* does:

- ► Connects game controller axes with controls inside IL-2 (using IL-2s DeviceLink).
- ► Supports user defined profiles (collections) of such connections (e.g. one for single engine fighters, another for a twin engined Mosquito, ...).
- ► Supports the following controls in IL-2:
  Ailerons
  Brakes
  Elevator
  Flaps
  Power
  Power Left Engines
  Power Right Engines
  Propeller Pitch
  Propeller Pitch Left Engines
  Propeller Pitch Right Engines
  Rudder
  Trim Ailerons
  Trim Elevator
  Trim Rudder
- ► Supports up to 8 axes for each game controller and an unlimited number of game controllers.
- ► Offers a fix for the „G940 Reversal Bug“.

What *YaDeLi* doesn‘t do:

- ► It doesn‘t support game controller buttons (do use AutoHotkey, it‘s open source and almost perfect).
- ► It doesn‘t *read* from DeviceLink (would be useless online, anyway).

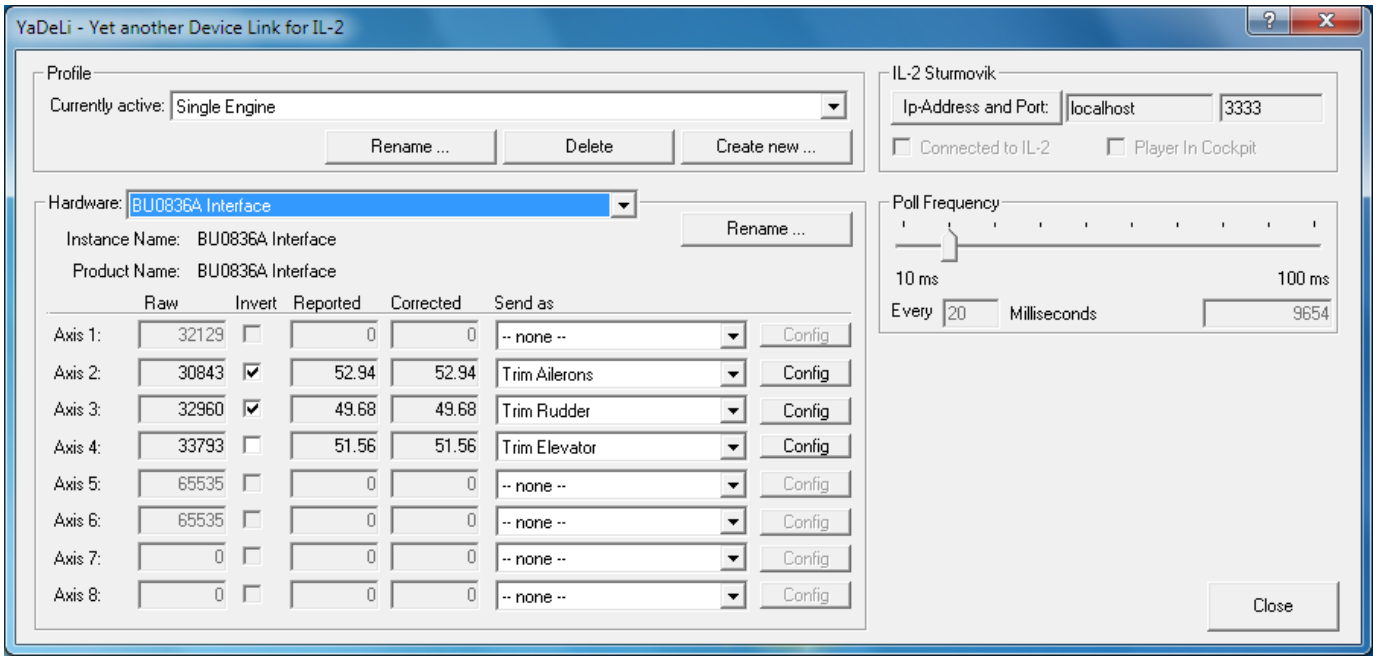
***YaDeLi* has been made for:**

- ► Those of us, who use more than the 4 game controllers that IL-2 supports.
- ► Those owners of a Logitech G940 Flight System who have *not yet* modded the „Reversal Bug“ away.
- ► Those owners of a Logitech G940 Flight System who *have* modded the „Reversal Bug“ and have found that the modded pedals will not show under „Game Controllers“, but will still block ID3 within IL2 ....
- ► ... suggestions, anyone?

Getting Started

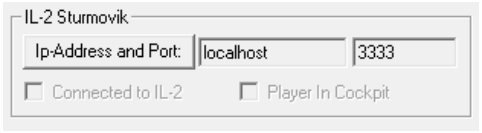
When you run *YaDeLi*, it will show a tray icon:

Right click on the tray icon and choose „Configure...“. This is what you will see next:



For *YaDeLi* to connect to IL-2, you must do two things:

1. Edit the file „conf.ini“ in your IL-2 directory. Search for „DeviceLink“. If you find the section...
   ```
   [DeviceLink]
   port=3333
   ```
   ... fine. If not, create it.
   Attention: Changes in the „conf.ini“ are best done when IL-2 is not running - apart from the fact that they will take effect only after a restart of IL-2, anyway.
2. Make sure that the number behind `port=` is the same as in *YaDeLi*:



- You can use any other ports, just make sure that it‘s the *same* number on *both* sides.
- The setting „localhost“ is just fine if both *YaDeLi* and IL-2 run on the same machine (which normally should be the case).
- To change the settings, click on „IP-Address and Port:“
- YaDeLi and DeviceLink use the UDP network protocol. Check your firewall if it doesn‘t work.

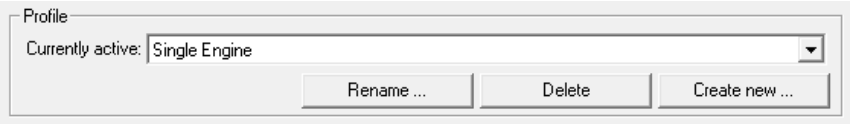
YaDeLi checks once every second for IL-2 and, when connected, if the player (you!) happens to be in the cockpit of a plane. The status will be indicated by a checkmark at „Connected to IL-2“ (and by the tray icon color: green, if connected) and a checkmark at „Player In Cockpit“ respectively.

When the player enters a cockpit, *YaDeLi* transmits the status of *all* configured axes once and after that only the changes in the axes.

When the player leaves the cockpit, *YaDeLi* stops transmitting. Makes sense, doesn‘t it?

Profiles

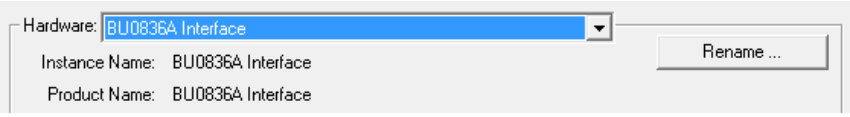
You can create, rename, and delete profiles. A profile contains a collection of connections between game controller axes and controls in IL-2. The idea is, that the same axis could be „Power“ for the single engine of a Bf 109 in one profile and „Power“ for the right engine in a „Mosquito-profile“. The active profile is the one selected,



and only the active profile‘s connections are transmitted to IL-2. What those connections are, you can see under „Hardware:“

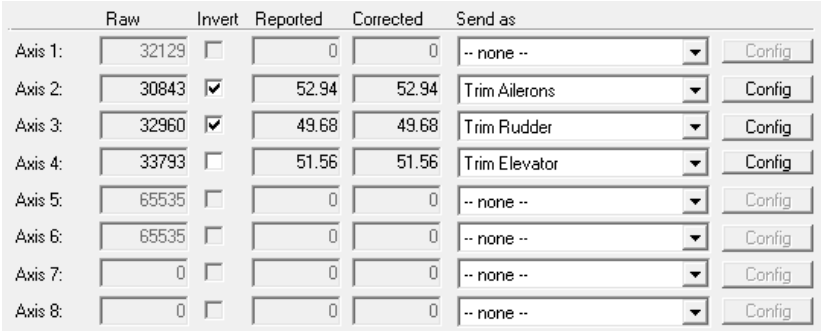
Hardware

YaDeLi list all *currently connected* game controllors under „Hardware“ (*YaDeLi* remembers game controllers which had been connected once, but will not list them):



You can „Rename :..“ a game controllor - but this will have effect only within *YaDeLi*. The reason for this is that some people (including me) run two of Bodnar‘s BU0836A side by side -and keeping them apart is a pain. The „Instance Name“ and „Product Name“ is what Windows chooses to reveal about the hardware (using DirectInput) - and displayed only for the unlikely case that you renamed a controller to something you‘ve no idea what you originally meant by it.

For every game controller *YaDeLi* lists the maximum of 8 possible axes - regardless of how many the game controller has in reality:



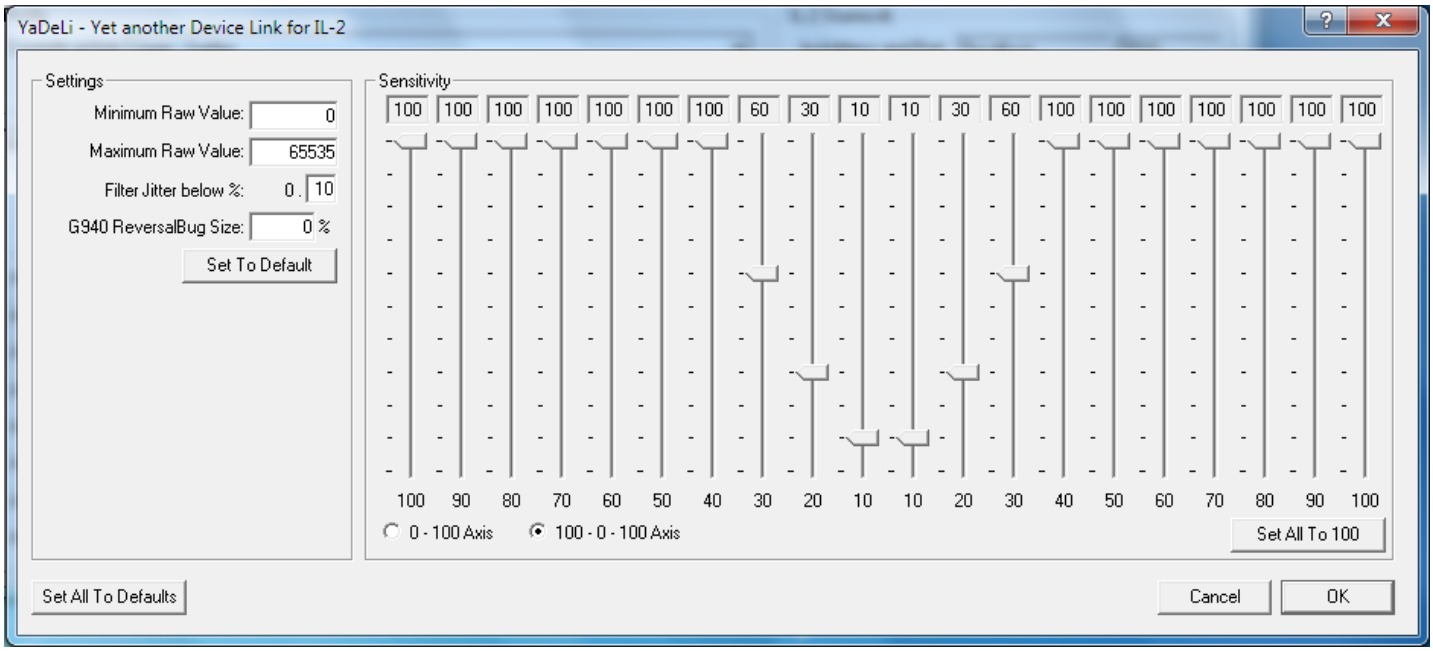
- Just move an axis you have on the (currently selected) game controller and hopefully you‘ll see the appropriate value in the column „Raw“ change.
- You can invert the axis by checkmarking in column „Invert“
- You can assign the axis to a control in IL-2 by choosing one in the column „Send as“. Currently *YaDeLi* (and DeviceLink) support: „Ailerons“, „Brakes“, „Elevator“, „Flaps“, „Power“, „Power Left Engines“, „Power Right Engines“, „Propeller Pitch“, „Propeller Pitch Left Engines“, „Propeller Pitch Right Engines“, „Rudder“, „Trim Ailerons“, „Trim Elevator“, and „Trim Rudder“. Suggestions (with a detailed howto!) are welcome.

Note: *Within one profile*, you can assign a IL-2 control only once - would not make much sense having the elevator connected to three axes at the same time, or would it? If you assign „Trim Rudder“ to an axis, it is automatically removed from any axis it was assigned before (just as in IL-2).

In case you‘re wondering what the columns „Reported“ and „Corrected“ are for, well, here it is: The „Reported“ column simply shows what the „Raw“-values translate to in percent. For the mystery of the „Corrected“-column, you must click on „Config“ of the axis...

The Axis Configuration

Click on „Config“ of an axis. This will open the following dialog:



What you can do here:

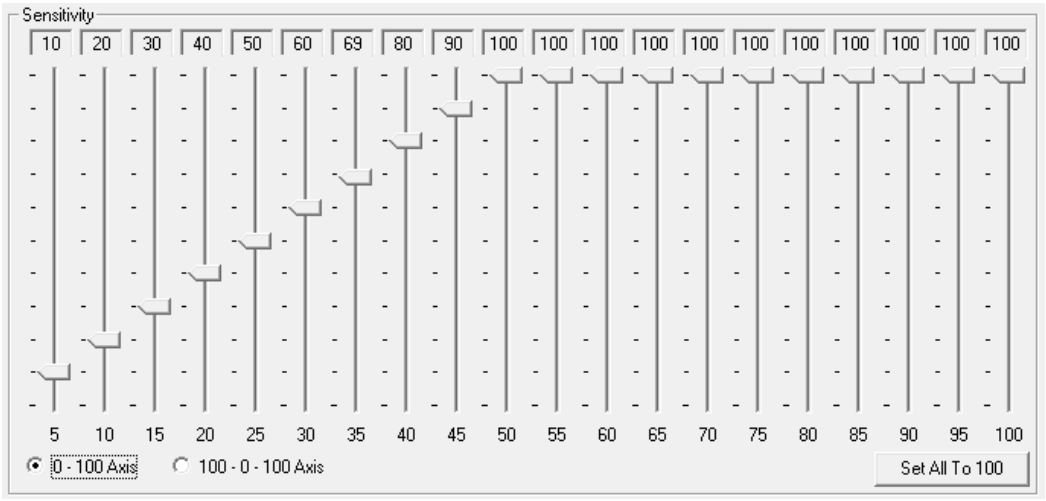
- You can adjust the translation of „raw“-values into percent. Normally, a game controller‘s axis has a range from 0 to 65,535 (after calibration). Now, with some controller‘s axis I‘ve made the experience that they don‘t: They range from 500 to 64,578, from whatever to whatever. By setting the „Minimum Raw Value“ to „500“ and the „Maximum Raw Value“ to „64,578“ you instruct *YaDeLi* to treat those numbers as 0% and 100% respectively. This will affect the value shown in the „Reported“-column in the „Hardware“ section.
- You can set a „Jitter Filter“ by specifying a fraction of a percent in „Filter Jitter below %:“, which must be exceeded to make *YaDeLi* process the change further. Some controllers have a little noise (i.e. ±0.09%) which would make the percent-value oszillate between 71% and 72%, which in turn would constantly display Power changes in IL-2. This will affect the value shown in the „Reported“-column in the „Hardware“ section, too.
- You can set a „G940 Reversal Bug Size“, which will cause *YaDeLi* to make an effort to correct this bug before sending information into IL-2. If you wonder, what this „Reversal Bug“ is: You move an axis slowly to one side: ..., 21%, 22%, 23%, 24%, 25%. Then you very, very carefully reverse your movement. What you would expect is the axis to report 24%. Well, it doesn't. It jumps to 21%. You reverse again and it jumps to 25%. The G940 does this on all axis except (since recently) the stick‘s x and y axes (aileron and elevator): both throttles, both R1 and R2 on the throttle, rudder, brakes, and all trims.

  What *YaDeLi* does to mitigate this, is the following: You know that your „Reversal Bug“ on the axis is 4%. You set this as „G940 Reversal Bug Size“ for the axis. When the „Reversal Bug“ now says „21%“ (see example above), *YaDeLi* reports 24%. While you go on moving to 23%, 22%, 21%, etc. *YaDeLi* slowly makes up the difference it has created in the first place. You can see this taking effect in the „Corrected“ column in the „Hardware“ section.

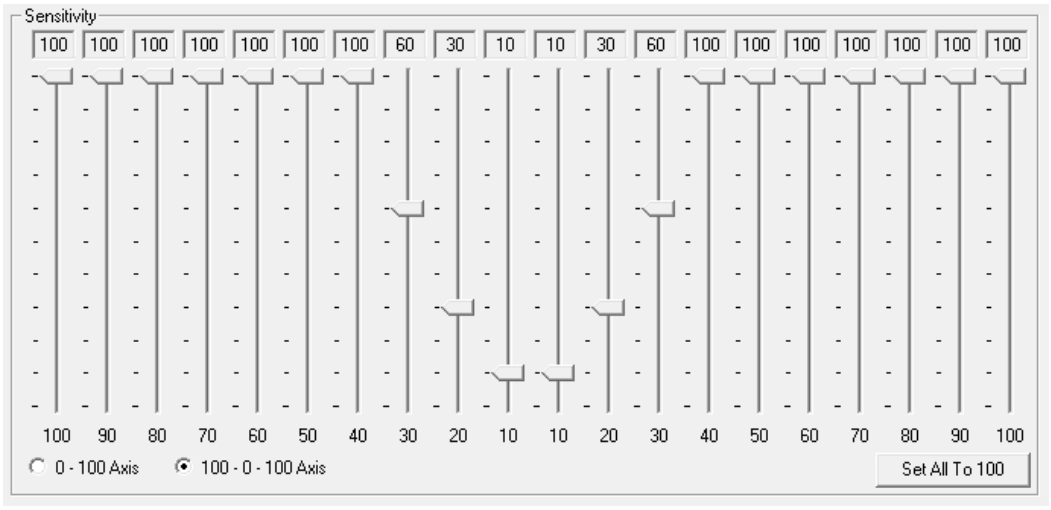
  Note: This works just fine on the „0 - 100“ axes (throttles, R1, R2), but not so well on the „100 - 0 - 100“ axes (such as trims and rudder) as the middle position of the physical stick and the 50% reported may not always coincide due to multiple „Reversal Bug“-corrections. With the trims that might be bearable - on the rudder pedals it‘s hell. Better mod your G940 with a BU0836X...

- Finally you can adjust the values actually send to IL-2 according to the range they are in („Sensitivity“). I assume that you‘ve seen this inside IL-2 already („Hardware Setup“, „Input“) and are familiar with the concept. A speciality here is that you‘ve to tell *YaDeLi* if the axis concerned is a „0 - 100“ (such as power or proppitch) or „100 - 0 - 100“ axis (such as rudder, elevator, ailerons, trims).

  In the first case, you‘ve got the full range from 0 to 100 in 5% steps to play with (I wonder what use that could be, suggestions welcome!):



  In the second, more likely case, you‘ve got 0 to 100 in 10% steps mirrored around the pivot „0“ (the example shows possible settings for the rudder).



The Poll Frequency

YaDeLi polls the status of game controller‘s axes. You can adjust the poll rate between 10ms (=100 times per second) and 100ms (10 times a second). I am not sure if this really must be configurable, but maybe somebody has to... (The increasing number at the bottom right is just a check if *YaDeLi is* polling at all).

Finally: Have fun.