Bravo++ multi-mode

DISCLAIMER

This is a beta release of a script I developed for personal use in the hopes that others would find it useful and fun. I am distributing it for free personal use and I appreciate feedback, but please don't expect me to provide full-time support on this.

If the script doesn't work for you, you can submit the `log.txt` file and a description of the problem on the X-Plane forum and I will try to see if I can solve the problem, but it may take time. For platform users that are not on Windows, I can only provide limited help, since I only run X-Plane on Windows 11. That doesn't mean it won't work on other platforms and I encourage you to try, but if you get a platform specific problem I probably won't be able help you.

Description

Bravo++ allows you to configure multi-mode functionality, so that you get more out of your Honeycomb Bravo than just the basic autopilot. The default mode (AUTO) will retain the standard autopilot functionality, but you can configure additional modes so that you can use the selector switch, buttons and rotating button to control other functionality in the aircraft. I have minimized the use of the Honeycomb Configurator and the only two controls that still need to be configured there are the right knob and the trim wheel (these can be configured outside the Honeycomb configurator, but the behavior won't be as good). There are some configuration files provided for the default aircraft such as the Cessna 172, the King Air C90B, and the Cirrus SF50 along with configurations for the Aerobask DA42 and DA62. Hopefully these will be enough so that you can configure you're own aircraft and perhaps submit it to the collection.

Prerequisites:

- X-Plane 12
- Honeycomb configurator (optional, but I found it necessary for my installation runnning Windows 11)
- FlyWithLua NG

The functionality is provided as a FlyWithLua script and consists of 3 parts:

- BravoMultiMode.lua - This is the main script that provides all the functionality and is placed in the FlyWithLua/Scripts directory.
- log.lua - This is a log utility that is used by BravoMultiMode.lua and is located in the FlyWithLua/Modules directory.
- config file - The file can be called either bravo_multi-mode.cfg or bravo_multi-mode..cfg and is where you onfigure all the different modes you want to have on your specific aircraft. Some example files are included in the FlyWithLua/conf directory and should be placed directly under the corresponding aircraft folder.

There is also an extra utility called ButtonLogUtility.lua that is used to determine which buttons the selector knob is mapped to in X-Plane. By default it is set to 0 and will use the HID to determine the state of the left selector knob, but this will introduce lag (at least on Windows 11). So to have a more responsive update to the GUI it is better to determine the button number X-Plane has assigned to the selector knob when it is set to "alt".

Installation

You should begin by installing the FlyWithLua plugin for X-Plane 12. If you already have it installed make sure that it is the NG version.

Next you can either download the zip archive from the X-Plane forum or get the latest release from the GitHub repository. All relevant files are found under the FlyWithLua directory and the entire FlyWithLua directory should be copied under the `plugins` folder.

Finally, you can optionally install the Honeycomb configurator developed by Aerosoft. For Windows 11, I notice that through HID, the knob is polled roughly every second leading to "clicks" being missed which in turn make the knob and trim unusable. This is the reason I need to have it installed on my machine, but it is a minimal profile that just assigns the right knob and the trim wheel to the configurator. This may not be a problem on Mac and Linux, but I have not tested on those platforms. Anyhow, the links to download are:

- For Windows
- For Mac

Configuration

Determining the button number for the left selector knob

In order to determine which number X-Plane has assigned to the selector knob you need to enable the `ButtonLogUtil.lua` that should be located under the `Resources\plugins\FlyWithLua\Scripts` directory. You do this by opening the file and setting the `local write_log = false` to `local write_log = true`. This will allow the script to write output to the `log.txt` file located directly under the X-Plane 12 directory.

Now you simply load up an aircraft and once it is loaded you can twist the left selector knob through the full range of selection and finally select the "Alt" setting.

You can now open the `log.txt` file and look at the output. If all went well you should have some lines with the text "Last button pressed:" with a number. Note the last number you see.

Now you open the `BravoMultiMode.lua` file under the `Resources\plugins\FlyWithLua\Scripts` directory and you look for the `local alt_selector_button = 0` and replace the "0" with the number you noted from the log.

Go back to the `ButtonLogUtil.lua` and disable the logging by setting the `local write_log` back to false.

Configuring the buttons in X-Plane

Next you need to confgure the Honeycomb Bravo buttons to use Bravo++. You may want to create a base profile (called Bravo++) that X-Plane uses, since this can be reused between aircraft configurations. Otherwise chose an existing profile and start configuring the buttons.

Here are the corresponding datarefs for each button:

-

HDG = FlyWithLua/Bravo++/hdg_button
- NAV = FlyWithLua/Bravo++/nav_button
- APR = FlyWithLua/Bravo++/apr_button
- REV = FlyWithLua/Bravo++/rev_button
- ALT = FlyWithLua/Bravo++/alt_button
- VS = FlyWithLua/Bravo++/vs_button
- IAS = FlyWithLua/Bravo++/ias_button
- AUTOPILOT = FlyWithLua/Bravo++/autopilot_button

For finding the corresponding datref in X-Plane just search for "Bravo++" and you should see all the available datrefs (you need to hover to get the actual dataref) you can map to. Otherwise the descriptions are pretty self-explanatory.

There is one more dataref that needs to be mapped and that is the one that toggles the modes. You need to determine where you want that button or key to be. I would suggest having it on a button on the joystick or yoke, and then you map the `FlyWithLua/Bravo++/mode_button` dataref to the desired button.

Configuring the right twist knob and the trim wheel

You can try to configure the twist knob and the trim wheel in X-Plane, but I personally get issues with latency that results in not all the clicks getting registered as I turn the knob or wheel. I may delve deeper into this issue to see if I can resolve it, but for now I use the Honeycomb Configurator from Aerosoft to solve the problem. I am aware that Mac users have issues with the software, but perhaps it will work with the minimal setup config file I have provided.

You will find the Honeycomb Configurator file under `Resources\plugins\FlyWithLua\conf\Bravo++_honecomb_configurator.json` and you import it using the following steps:

- Select "Actions > Open settings"
- Click on "Import profiles"
- Select the file `Resources\plugins\FlyWithLua\Scripts\Bravo++_honecomb_configurator.json`
- Select the profile `Bravo++ Multi-mode`
- Click "Ok"

Once imported you need to activate the profile either before starting up X-Plane or if you do it while X-Plane is running, you need to ensure the current aircraft is using the profile by selecting in the X-Plane menu `Plugins > HoneyComb > BFC_Throttle > Reload bindings`.

Make sure that the right twist knob and trim wheel are not configured in X-Plane; i.e. they should be set to "Do nothing". On the other hand, if you want to try the functionality without using the Honeycomb Configurator, then you need to set the appropriate datarefs in X-Plane.

For reference, the relevant datarefs that are configured are as follows:

- Increase value (turn knob to the right) = FlyWithLua/Bravo++/knob_increase_handler
- Decrease value (turn knob to the left) = FlyWithLua/Bravo++/knob_decrease_handler
- Nose up (turn wheel down) = FlyWithLua/Bravo++/trim_nose_up_handler
- Nose down (turn wheel down) = FlyWithLua/Bravo++/trim_nose_down_handler

Configuring the aircraft

The easiest way to start is to use one of the predefined G1000 configurations (all aircraft, but the King Air C90B) like the for the Cessna 172. I won't go into the details on the content of the config file in this section and assume you want to get going as quickly as possible.

So let's configure the Cessna 172 that uses the G1000.

Start by copying the file called `Resources\plugins\FlyWithLua\conf\bravo_multi-mode.Cessna_172SP_G1000.cfg` to the `Aircraft\Laminar Research\Cessna 172 SP` directory. Note that the Cessna has 3 .acf files and the configuration contains the name that is in `Cessna_172SP_G1000.acf`. This is how the script knows which configuration to use when it starts up. If you start any of the other two variants that aren't G1000 equipped, the script will just stop, since it can't find a corresponding config file.

Once the file is copied, you can load the aircraft and hopefully you will now see the Bravo++ window that contains the current mode and status of the buttons. If you don't then either the script couldn't find the config file, the Honeycomb Bravo device is not plugged in or something went wrong with the script. In the latter case you will probably hear FlyWithLua complaining and telling you that it has moved the bad script to `Script (Quarantine)` folder. This shouldn't happen, but if it does check the `log.txt` file for any errors.

The Bravo++ can be popped out as a separate window, which is useful if you have multiple monitors. I personally have 4 monitors, where I have the G1000 PFD, MFD and the Bravo++ window on the smallest monitor.

So initially you will see the default mode on the left (AUTO in green) and the currently selected value for the left selector knob. On the bottom you will see all the corresponding buttons in grey and if they are active they will be in white. If they are white the corresponding led on the Bravo device will also be lit. Finally on the right you have the "outer" and "inner" selection which are used when using the other modes. This controls whether the inner or outer knob is to be turned when using the right twist knob. So once you start up the aircraft you can test out the functionality by pressing the "HDG" button and if all is well you should see that the "HDG" button on the device will light up and the Bravo++ window will now show the text in white. By pressing the "HDG" again it should make the button inactive again.

To change the mode, you need to click the button you assigned to it. Clicking the button allows you to cycle through the different modes. For the Cessna 172, there are 3 modes (AUTO, PFD and MFD) and it is possible to add additional modes if desired, but that is for another day. If you are curious about additional modes you can look at the DA42 or DA62 configuration which contains an additonal mode called "SYS" that allows settings the lights, operating the anti-ice system, ignition and auxiliary pumps. You can basically configure whatever you want, but there are some known limitations which I won't take up here now.

If you select the "MFD" mode you will notice that the text changes for most of the content. If you turn the left selector knob to "ALT" you will notice that in the Bravo++ window it now indicates "COM". On the bottom, you will also notice that the labels for the buttons are now different. Some of the buttons do nothing, while others will performa action. So from this selection your are able to tune the com radio frequencies using the right twist knob and the buttons. The "IAS" button on Bravo device now toggles whether the twist knob controls the inner or outer ring of the knob. So when it is set to outer it will control the MHz values of the frequency (118 - 136 MHz), while inner will control the KHz frequencies. The "VS" button will control which frequency is active by swapping the frequencies. The "ALT" button allows you to swicth between COM1 and COM2. Notice that the text for these buttons on the Bravo++ window are blue-green. This indicates that they toggle something without the causing the led light to go on. The "REV" button, on the other hand, is dark grey and this indicates that the led light will be activated if pressed. In this case it will unmute the COM2 speaker and cause the led light to go on. So try dialing in an ATIS/AWOS frequency at the airport you are at on COM2 and then unmute it by pressing the "REV" button. You should hear the ATIS/AWOS track.

I suggest you explore the rest of the functionality, especially the "FMS" selection, which allows you to access the flighplan menu and procedure menu without using the mouse.

More documentation to come...

Known bugs

I am aware of some minor annoying bugs:

- The button for switching the com frequency doesn't work all the time. You just have to be persistent and press it multiple times. This doesn't happen with the nav frequency, so I am not sure why it doesn't work.
- The leds may freeze. This hasn't happen for a while and it may no longer be an issue, but if it happens you need to restart the Lua script from the Plugins menu.