NJoy32 Family Controllers Configuration Utility

User guide

V. 2.14, 04.03.2021.

Table of content

Chapter 1. Overview

VKB Njoy32 device configuration tool is intended to make the following actions:

- setup joystick controls,
- joystick axes calibration,
- save and load joystick parameters,
- preparing controller to firmware upgrade.

VKB Njoy32 device configuration tool (configurator) is saved in *VKBDevCfg-C.exe* file. It does not need to be installed. To launch configurator you must just execute this file. Up-to-date configurator versions you can find at VKB site http://forum.vkb-sim.pro in Download page. This page contains actual versions of firmware, profiles and firmware upgrade tool saved in *Z-Bootloader.exe* file.

1.1. Hardware and software requirements

Configurator works with the following VKB devices:

- Gunfighter SCG,
- Gunfighter MCG,
- Gladiator series,
- Mamba series,
- KingCobra series,
- pedals with TinyBox,
- ThrottleBox,
- BlackBox with any grips,
- Cobra-Z (Defender Cobra M5 USB Mk2 with Njoy32 controller). Configurator works under Windows XP, Windows 7.

1.2. What`s new

1.2.1. V.2.7

- Relative axes description (see 3.4 on p. 35).
- Radiobutton button function (see 4.3.3 on p. 55).
- RelAxes — relative axis mode control button function (see 4.3.22 on p. 87).
- Ministick mode switch button function (see 4.3.13).
- Boolean — Boolean functions for buttons (see 4.3.23 on p. 87).
- Analog POV (see Chapter 6 on p. 106).
- New Axis to buttons subfunction.
- GTR/GTR2 pulse generators.
- Trimmer return (see “Trimmer Reset, Trimmer Return” on page 78)

1.2.2. V.2.8

- RT-trigger in Boolean functions.

1.2.3. V. 2.9

- MCG Pro folding trigger setup (see 3.7.2 on p. 44).
- New event for indication — Mouse active (see 2.4.11 on p. 23).
- MCG Pro folding trigger calibration (see MCG Pro folding trigger calibration on p. 46).

1.2.4. V.2.10

- Smooth trimmer reset (see Trimmer Reset, Trimmer Return on p. 78).
- Using clipboard for physical lines setting (see Line (input control) choice on p. 51).
- Two virtual buttons switch on diagonal HAT pressing (see 6.3.2 on p. 107).
- CMP function, Comparator in Boolean functions (see CMP on p. 89).
- JMP function, Jumper in Boolean functions (see JMP on p. 90).
- Using Drag-n-Drop for profile loading (see 8.4.2 on p. 115).

1.2.5. V.2.11

- BrakeV3 axes combining mode (see BrakeV3 on p. 40).
- Simplified setup of axes response to buttons converting (see 3.7.3 on p. 45).

1.2.6. V.2.14

- Network technologies (см. Chapter 10 on p. 121).
- Controller connection using Z-Link (see sect. 10.7 on p. 128.
- Alternative colors.
- Flash LED mode.
- Forced Limit for relative axes.
- New axes combining BrakeV modes (see sect. 3.5.4 on p. 38).
- New response curve configuring mode (see sect. 3.6 on p. 40).
- Dependent Button BD (see sect. 4.3.6 on p. 58).
- uStick ministick mode switch (see sect. 4.3.13 on p. 69).
- uPOV active POV switsh (see sect. 4.3.14 on p. 70).
- Complementary button Switch CB (see sect. 4.3.15 on p. 71).
- Dynamically deadzone disable DZ Switch (see sect. 4.3.24 on p. 92).
- New features of Tempo function (see sect. 4.3.17 on p. 75).
- New auxiliary axes functions SWITCH 0, PAI INV, SPLIT Rev (see sect. 4.3.21 on p. 84).
- New ministick output types (see sect. 6.3 on p. 106).

- New ministick function, Shifter 6W (see sect. 6.3.5 on p. 108).

Edited:

- Button assignments report.

Ecluded:

- Profile switch on-the-fly,
- Group button opeations.

Chapter 2.Using configurator

2.1. Interface

After you launch *VKBDevCfg-C.exe* the main **VKBDeviceConfig** window will appear. The title bar contains configurator version (Fig. 2.1).

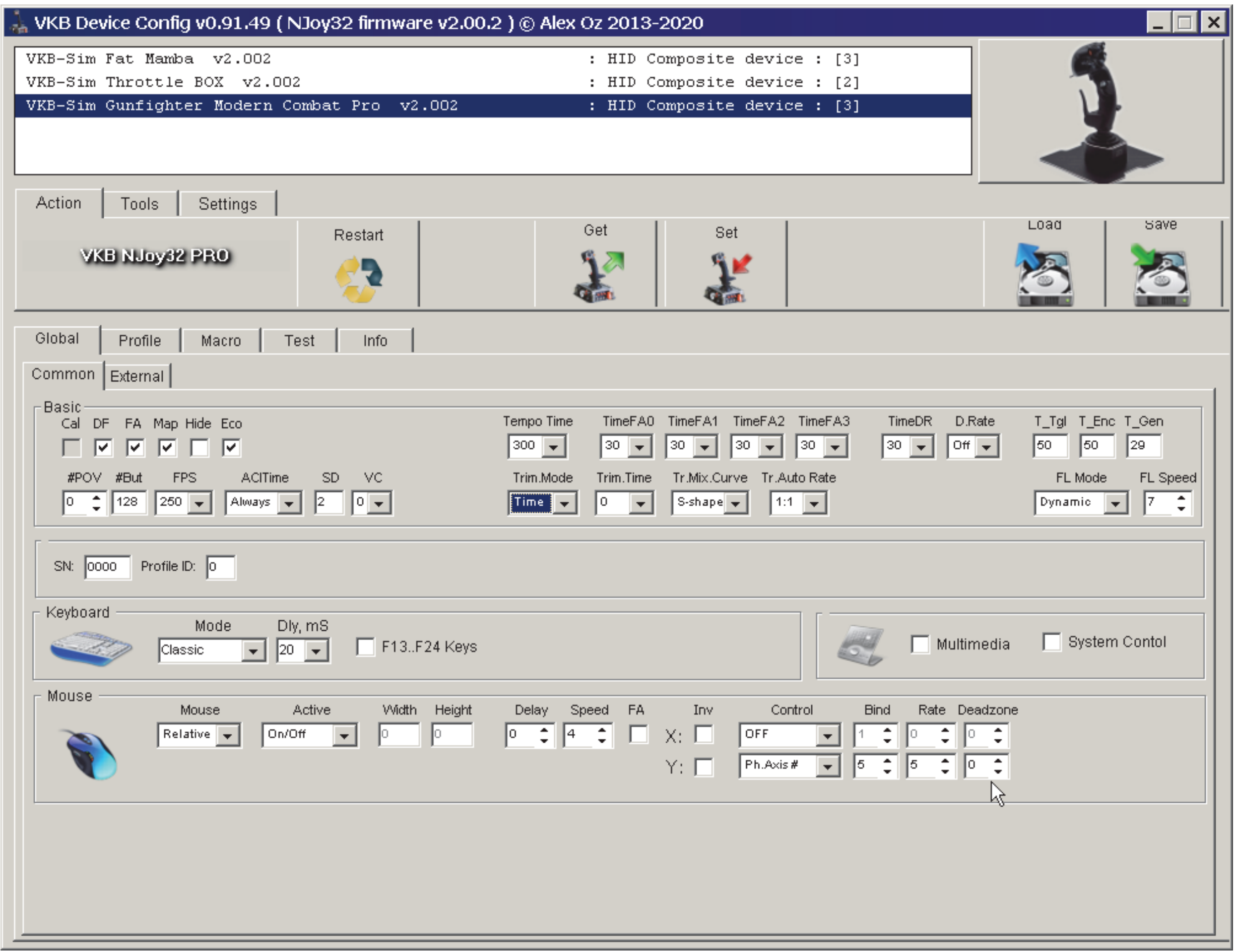


Fig. 2.1.

Connected VKB devices are listed in this window. To setup certain device you must select it in the list.

2.1.1. Device info

Info tab contains the following device data:

- product name,
- firmware version,
- device mode (pro/light),
- T-Link support ready (for pedals controller).

You can see an example of **Info** tab content on Fig. 2.2.

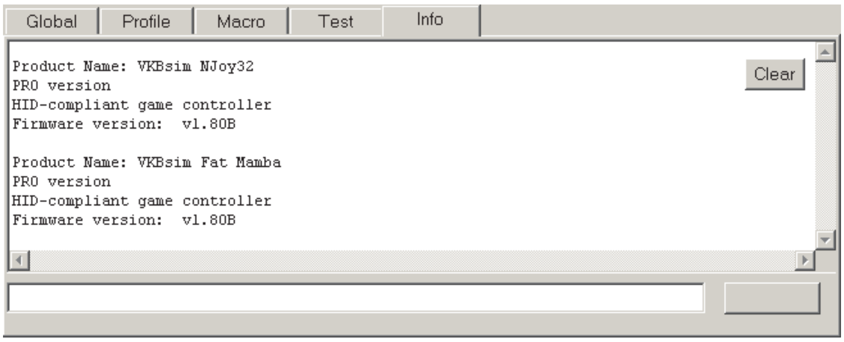


Fig. 2.2.

2.2. Firmware upgrade

2.2.1. Software activation

Use firmware updating utility saved in *ZBootloader.exe* file. Execute the following actions to upgrade firmware.

1. Connect controller to the PC.
2. Run *VKBDevCfg-C.exe*, select joystick name in the list (fig. 2.1 on p. 16).
3. Press **Bootloader** button on **Tools** tab.

You will see firmware upgrade utility window (fig. 2.3)

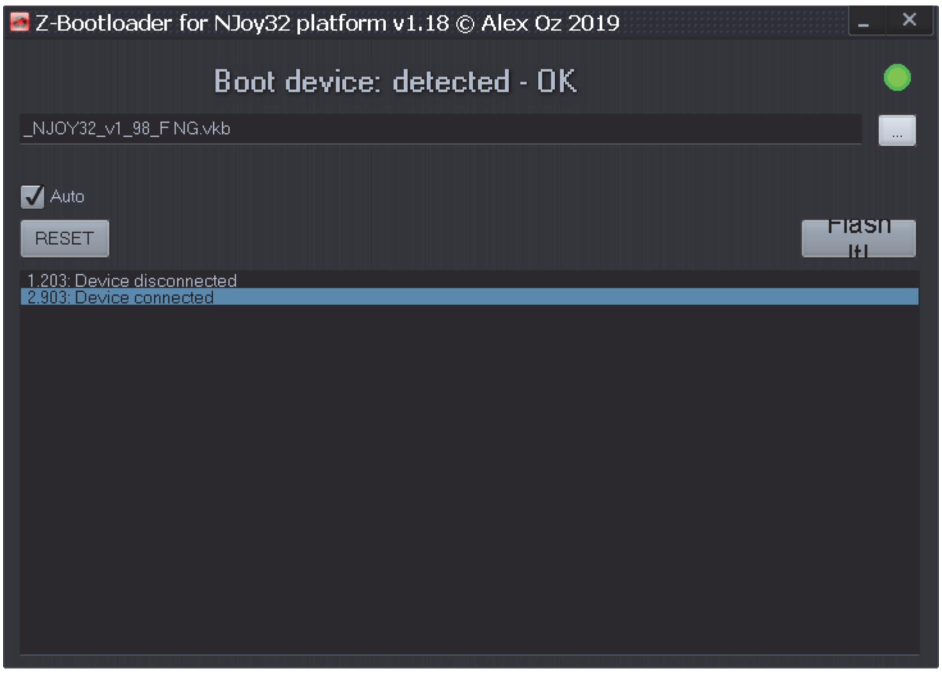
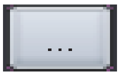


Fig. 2.3.

When *ZBootloader-C* runs, configurator window will be closed automatically. If it still for any reason runs, close it manually. Configurator and *ZBootloader-C* must NOT run simultaneously!



4. Press **Browse** button and select firmware file in standard Windows dialog.

   You will see its name in dialog.

5. Press **Flash It!** button.

   Some information messages will appear in utility window. After successful upgrade completion z-bootloader window will be closed automatically. Launch *VKBDevCfg-C.exe*. You will see device name with new firmware version number.

Some versions of firmware reset controller parameters to default values. If you have made custom settings of controller parameters save you profile to file. After firmware upgrade you can load your custom profile (see Chapter 8 on p. 114).

2.2.2. Hardware activation

In some circumstances software activation could not start as described above. For example operation system did not recognize joystick. In this case you even will not see joystick name in the list of connected devices. To activate controller execute the following actions.

1. Disconnect joystick from the PC.
2. Run ZBootloader.exe. You will see that device is turned out (Fig. 2.4).

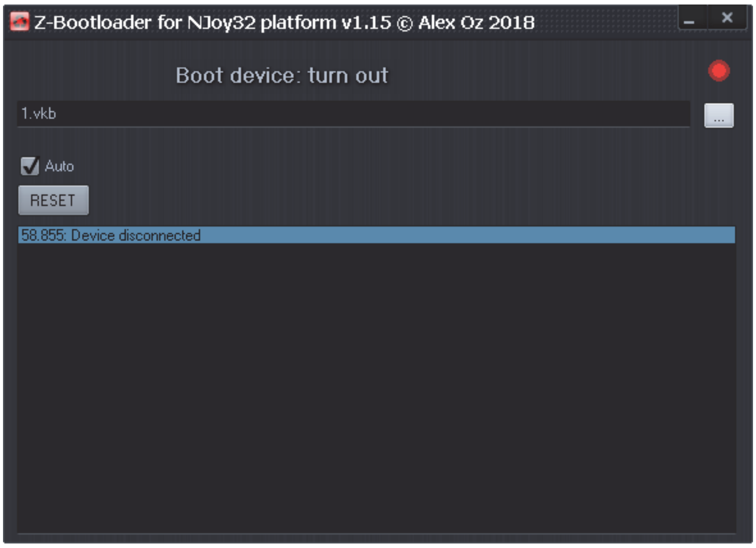
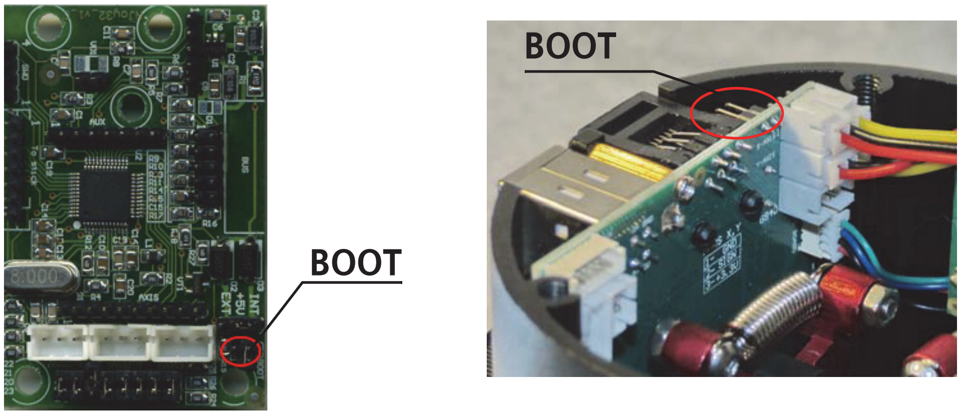


Fig. 2.4.

3. Short BOOT jumper on the controller plate (Fig. 2.5).



a) b)

Fig. 2.5.

> You can see BOOT jumper of Njoy32 1.1 (a) and VKB Mamba (b) controllers. Look for BOOT label on other controller PCBs.
> Press and hold down Eject and Rlaps buttons of Gladiator joystick, press Boot button on BlackBox Mk2 cover to activate firmware upgrade.

4. While BOOT jumper is shorten, connect joystick to the PC and then release jumper.
5. You will see that device is connected (Fig. 2.3 on p. 17). Further actions are the same as described in sect. 2.2.1 on p. 17.

2.3. Common parameters

2.3.1. Global variables

Open **Global — Common** tab to set global controller parameters. Table 2.1 on p. 19 contains control item descriptions.

Table. 2.1. Global controller parameters

Control name	Description
Checkboxes	
Cal	Calibrate status. Device axes calibration status. Checks automatically after calibration completion (see 3.8 on p. 46).
DF	Dinamic Filter. Switches dynamic axes response filter. You can assign filtration rate for each axis separately see 3.2 on p. 31.
FA	Fix Axes. Allows to fix axis state (see 4.3.20 on p. 82).
Map	Logical button mapping. Allows physical button mapping to logical ones or keyboard.
Hide	Hides current device.
Eco	Allows to control virtual keyboard connection time.

Table. 2.1. Global controller parameters

Control name	Description
Fields	
Tempo Time	Tempo function delay time (see 4.3.17 on p. 75).
T_Tgl	Time of toggle pulse. Toggle pulse width (see 4.3.9 on p. 61), milliseconds.
Time FA0, Time FA1, Time FA2, Time FA3	Axes fixation delay time (ms), see sect 4.3.20 on p. 82).
TimeDR	Double rate function (fixed rate axis response reduce) delay (ms), see sect 4.3.20 on p. 82.
D.Rate	Axis response reduction rate list.
T_Enc	Time of encoder pulse. Encoder pulse width, milliseconds.
T_Gen	Time of generator pulse. Generator pulse width, milliseconds.
Trimmer Time	Trimmer reset duration, ms (see Trimmer Reset, Trimmer Return on p. 78).
Trimmer Mode	Axes trimming modes. **Time**. Trimming rate independently of its value applies to axis in a time selected from **Trim. Time** combo box. **Rate**. Trimmer to axes engagment depends of **Trim. Time** value and trimming rate.
Trimmer Mix. Curve	Applied trimming curve type, S-shaped or linear.
Trimmer Auto Rate	TrAuto+/TrAuto- function fall and rise rate. 1:X values accelerate fall, X:1 ones slow down, see sect. Trimmer Reset, Trimmer Return on p. 78

It is recommended to set **T_Enc** value between 10 and 50 ms. If you use internal button mapper, set 10 - 20 ms. For external mappers use value not less than 20 ms.

2.4. Additional parameters

2.4.1. Control item number

#POV field allows to set Hat switch number up to 4.

> If you do not use Hat view control, for example you use Natural Point Trackir, you can set this parameter equal to 0 and use Hat switch as four buttons.

#But field allows to set maximum button number. Maximum number is equal to 128. Do not forget about Windows DirectInput restrictions. You can see 32 buttons only using Windows applet. To see buttons above 32 use VKBjoytester tool.

2.4.2. Sampling rate

FPS combo box allows to choose controller polling sampling rate by USB (Gz).

2.4.3. Automatic calibration time

AClTime allows to set time of automatic joystick calibration. It is executed on every device connection. Automatic calibration measures extreme response values and fixes center position. Its result depends of initial axis position. Automatic calibration will be executed permanently if you choose **Always**. For other values it will be executed for chosen period only.

2.4.4. Tempo Time

Tempo function delay time, ms (see 4.3.17 on p. 75).

2.4.5. Fix axis delay

Time FA0, **Time FA1**, **Time FA2**, **Time FA3** combo boxes allow to set delay time (milliseconds) before axis fixation will be applied (see 4.3.20 on p. 82).

2.4.6. Axis sensitivity reduce

Sometimes this is reasonable to reduce axis sensitivity. For example when you try to aim the enemy. **D.Rate** combo box allows to set reduce factor.

Set delay time (milliseconds) before sensitivity will be changed using **TimeDR** combo box. You must assign joystick button to control this function (see 4.3.20 on p. 82).

2.4.7. Controller restart delay

Every time you have changed controller parameters you must save new settings in its memory. On some computers sequential parameter saving causes error. You will see error message (Fig. 2.6).

Fig. 2.6.

This error is a result of specific interaction between Windows and USB devices. If you have such troubles try to set nonzero value in **SD** (Delay of Start Controller) field (Fig. 2.7).

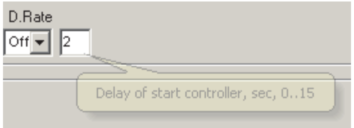


Fig. 2.7.

VC, Virtual HID controllers – virtual controllers count. Has practical meaning for Next Generation firmware.

2.4.8. View loaded profile name

You can use specific profiles for different programs. Those profile names can be indicated in configurator window. Create data set for this purpose. An example you can see in table 2.2.

Table. 2.2. Profile parameters

ID	Description	File name
1	Fat Mamba profile for BoS	Mamba_1556_BoS.cfg
2	Fat Mamba profile for CloD	Mamba_1556_BoB.cfg
3	Fat Mamba profile for “old” Il-2	Mamba_1556_Il-2.cfg

All parameters in this example are optional. Add the following lines to **[User]** section of *Zconfig.ini* configuration file (saved in the same folder with *VKBDevConfig.exe*):

[User]

Profile 1= Fat Mamba profile for BoS

Profile 2= Fat Mamba profile for CloD

Profile 3= Fat Mamba profile for “old” Il-2

To see profile name enter its ID into **Profile ID** field (Fig. 2.9).

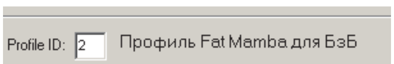


Fig. 2.9.

Appendix I. on p. 131 contains detailed description of *Zconfig.ini* file.

2.4.9. Virtual devices

You can map joystick buttons to keystrokes, use buttons to control system functions or multimedia applications. Control items of **Keyboard** group allow you to control those functions.

> Do not forget to check **MAP** to allow keyboard mapping.

Mode combo box options allow to choose virtual keyboard mode.

- **Off** — virtual keyboard disabled,
- **Classic** — standard keyboard; simultaneously pressed virtual keys allowed.
- **Modified** — simultaneously pressed virtual keys disabled; if you press some keys only the last pressed one will work. No combinations like <Ctrl+Z> are accessible.

> If **Map** checked and any virtual keyboard mode chosen new virtual device *HID Keyboard Device* will appear in device list.

> An example of **Standard** mode use ClasterFire triggers of Mamba family joysticks. If first trigger fires gun and the second cannon, than when you press both triggers you will fire all weapons. With **Modified** mode cannon only.

2.4.10. Virtual keyboard modifiers setting

The Battle of Stalingrad simulator series use nonstandard processing of keyboard modificators such as Shift, Ctrl etc. if they are mapped to joystick buttons. For example if you have mapped *Ctrl+A* combination to button, simulator randomly will process it right, i.e. *Ctrl+A*, or pure *A*. **Dly** parameter (milliseconds), resolves this issue. Value of 30 ms is recommended. Larger value can cause errors if you will press button with short intervals.

Check **F13**...**F24** to use corresponding functional keys.

Multimedia checkbox allows joystick buttons to control multimedia applications (see 5.8 on p. 103).

System checkbox allows joystick buttons to control system functions of operating system (see 5.10 on p. 104).

2.4.11. Virtual mouse

Joystick buttons and axes can be used to control virtual mouse. Use control items of **Mouse** group to set up virtual mouse.

Choose mouse type from **Mouse** combo box.

- **Off** — no virtual mouse.
- **Relative** — mouse cursor moves relatively to current cursor position.
- **Absolute** — mouse cursor moves from the center of the screen. It jumps to this point when you begin to move joystick axes or press buttons assigned to control mouse axes.

You can move cursor to the screen center forced pressing button with assigned **Set Center Point** function (see 5.5.2 on p. 101).

Virtual mouse switch mode depends on **Active** combo box items.

- **On/Off** — To switch mouse on you must press assigned button (see 5.5.2 on p. 101).
- **Always On** — Virtual mouse is turned on permanently.

Be careful using virtual mouse always turned on. Wrong controller settings or even small axis jitter that is invisible while ordinary joystick use will cause spontaneous mouse cursor move. It may be very difficult to neutralize this move with physical mouse.

Set **Width** and **Height** values (pixels) for **Absolute** mode.

If you use buttons for axes control, choose automatic cursor acceleration delay value from **Delay** combo box. Acceleration rate set using field with counter **Speed**. If you assign velocity rate for control button then cursor velocity will be constant, without acceleration.

If you use physical joystick axes to control virtual mouse ones, you can control physical axes state. If **FA** is checked than joystick axes control mouse only. Otherwise physical axes will control mouse and joystick axes simultaneously.

If virtual mouse is activated, corresponding event is generated, that can be used for LED indication (see 2.5 on p. 25).

Set mouse axes parameters usin control units on **Profile – POVs** tab (see sect. 6.3.4 on p. 108).

2.4.12. Communication ports

Controller has serial communication ports used for connection with external devices. Control items of **External** tab allow to set port modes.

SPI1 port mode combo box items allow to choose first port modes:

- **OFF** — not used,
- **S-but** — standard button registers connected.

Set register number in **RegN** field.

SPI2 port mode combo box items allow to choose second port modes:

- **OFF** — not used,
- **S-but** — standard button registers connected.
- **S-Led** — RGB LEDs connected.

2.5. Indication configuration

2.5.1. Overview

Standard LEDs

LEDs can be used to indicate joystick state. For example Mamba series joystick case has six LEDs, Gladiator has two LEDs.

Additional LEDs

You can use additional RGB LEDs WS2812 (https://www.drive2.ru/b/1646666/) for joystick buttons and axis state indication. Leds are selled as bands and differ in LED number per one meter, 30 (Fig. 2.10), 60 or 144.

Fig. 2.10.

Other variant – panels with 8 LEDs (Fig. 2.11).

Fig. 2.11.

Use fields of **Mode** group to configure using different LED types (fig. 2.12).

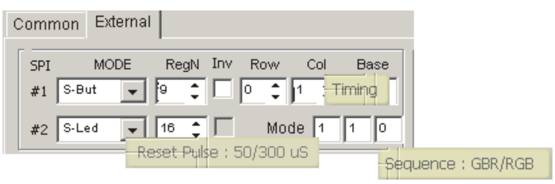


Fig. 2.12.

See configuration example in table 2.3.

Table. 2.3.

LED type	Values set
WS2812B	000
WS2812D	100
APA106	001

Use BUS connector to connect LEDs to controller. Fig. 2.13 shows NJoy32 single-plate controller connector.

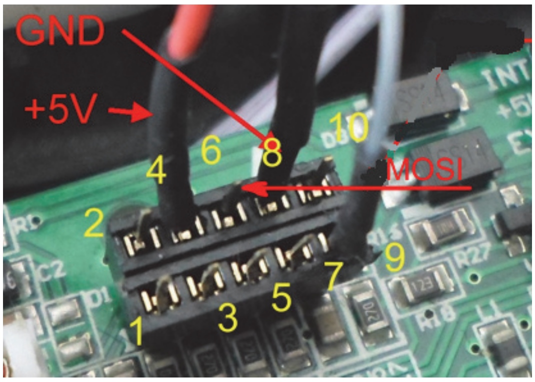


Fig. 2.13.

The following lines are used for LED connection:

- 4 - VCC +5B,
- 6 - MOSI,
- 8 - ground.

LED band contacts have corresponding designations. NJOY32 can control up to 80 LEDs with external power supply. It is not recommended to connect more than 5 LEDs without power supply. LED band has 3 signal contacts and 2 for external power connection, for example cell phone charger. LEDs connection without external power is shown on Fig. 2.14.

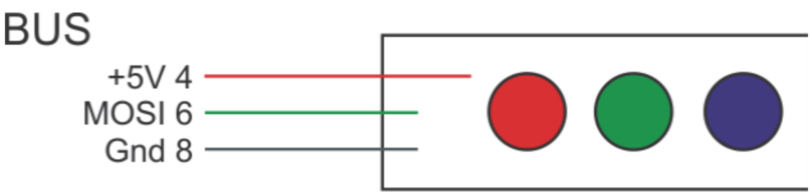


Fig. 2.14.

LEDs connection with external power is shown on Fig. 2.15.

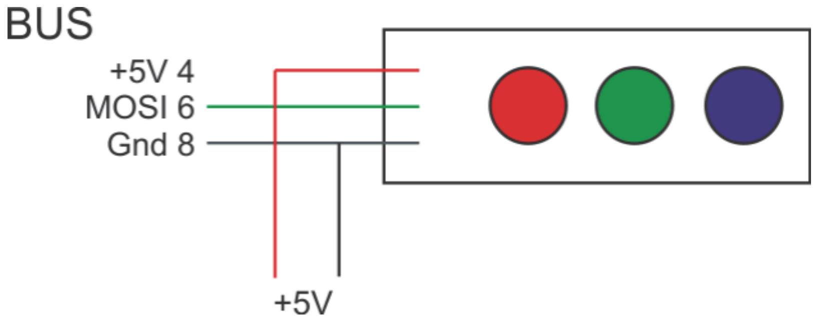


Fig. 2.15.

+5 V from power supply connect to LEDs only! Do not connect it to controller+5 V line.

Choose **WS2812** item from **#2** combo box to enable and control LEDs. Set count of additional LEDs using counter. Numbers of these LEDs begin from 8. An example of LED settings is shown on Fig. 2.16. The first additional LED with number 8 (the closest to controller on LED band) will flash ultra fast using two colors, green and light violet when SHIFT button is pressed.

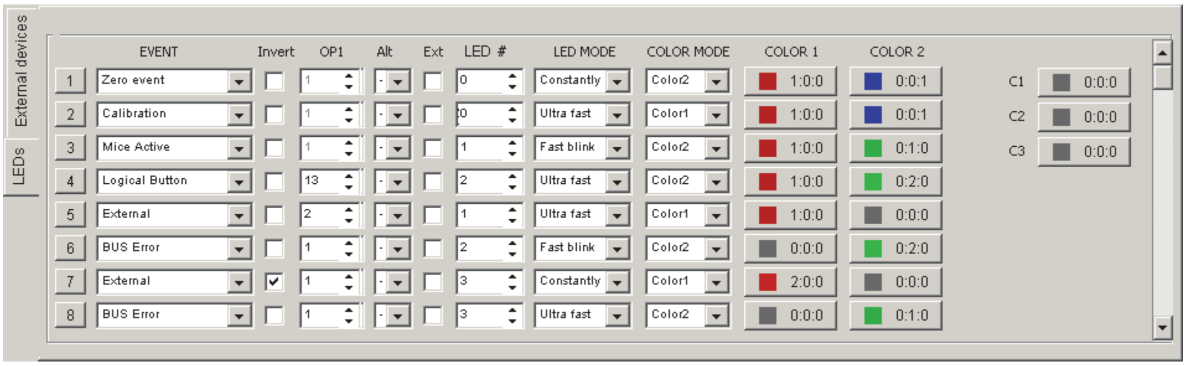


Fig. 2.16.

LED parameters

The following parameters are used to control LED light:

- color,
- frequency,
- brightness.

To set up indication open **Global — External — LEDs** tab. Names of indicated controller states and corresponding **Event flag** combo box items are shown in table. 2.4.

Table. 2.4.

Event flag item	Controller state	Note
External	No indication.	

Table. 2.4.

Event flag item	Controller state	Note
Zero event	Default controller state.	No button pressed, axes in zero or center.
SHIFT	Shift button pressed.	
SubSHIFT	SubShift button pressed.	
Fix Axis	FA (Fix Axes) button pressed.	
Calibration	Calibration is executed.	
Physical Button	Physical button pressed.	Indication can be inverted so LED will flash when button is not pressed but go out when button is pressed.
Virtual Button	Virtual button activated.	
Logical Button	Logical button activated.	
Axis in center	Axis is in center or in zero position.	Can be inverted.
MaRS fault	MaRS sensor failure.	Set testing MaRS number in **OP1** field.
Calibration fault	Calibration failure.	
Rudder connect	Pedals are connected to controller.	If controller has **Pedals** port.
BUS error	External device connection to BUS port failure.	
POV active	uStick is used as POV.	See Chapter 6 on p. 106.
Mice active	Virtual mouse is active	See 2.4.11 on p. 23.
Profile N	Active profile number	
Cyclic Switch		
P-Alternate Function	Alternative physical button activated	
V-Alternate Function	Alternative virtual button activated	

Table. 2.4.

Event flag item	Controller state	Note
Boolean Function		

2.5.2. Indication settings

Open **Global — External — LEDs** tab (Fig. 2.17).

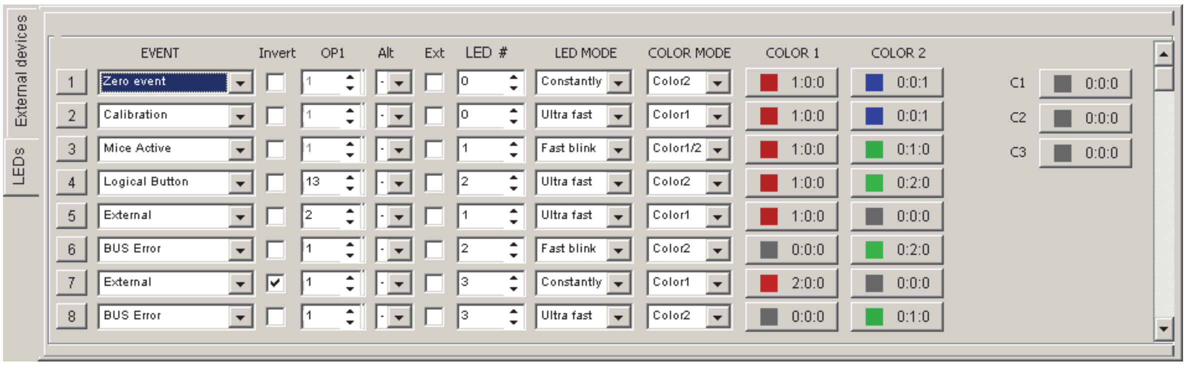


Fig. 2.17.

Choose desired event that you want to indicate from **Event** combo box. Use **Invert** checkbox to invert event. By default it is unchecked. For example if you have assigned LED to indicate axis central position than it will flash when axis is in its center. If you check **Invert**, than LED will not flash when this axis is in center. It can be useful in some cases. For example axis #8 is used to trim Axis #2 (roll). **Invert** is checked. Corresponding LED will flash only if roll axis is trimmed.

Set additional parameter in **OP1** field (needed for some events, see table 2.5).

Table. 2.5.

Event flag	Parameter
SHIFT	0 — Shift 1; 1 — Shift 2; 2 — Shift 0.
SubSHIFT	SubShift number.
MaRS fault	Testing MaRS number, 1...8.
Axis in center	Axis number, 1...8.
Physical Button, Virtual Button, Logical Button, P-Alternate Function, V-Alternate Function.	Button line number.

Alternative colour is used to indicate that its event NOT occurs. Select one of alternative colors C1, C2 or C3 from **Alternative Color** combo box. Configure selected color in **Color select** dialog. Press **Color 1** button to set corresponding color parameters. You will see setup dialog (Fig. 2.18).

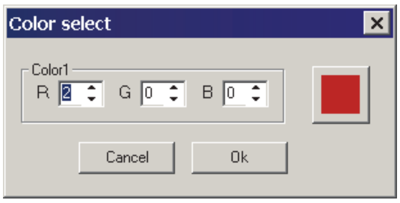


Fig. 2.18.

Using counters set up RGB components brightness. To complete setup close dialog. Set up the second color in the same manner.

Use **Alternative Color** for events with minimal index (priority) in order not to override other events for same LED. It shines in **LED mode Constantly** with chosen color.

Check **Ext.** to transmit LED state via network.

Set LED number for this event in **LED#** field. Single system LED has number 0. You can assign the same LED to indicate different events. In this case use different light parameters i.e. color, brightness, frequency. If some events assigned to single LED will occur simultaneously will be indicated event with greater number.

Choose the following LED light mode from **LED Mode** combo box:

- **Off** — LED is off,
- **Constantly** — constant light,
- **Slow Blink** — slow blink,
- **Fast Blink** — fast blink,
- **UltraFast** — ultra fast blink,
- **Flash** – short flashes with wide period.

Select the following LED color mode from **Color Mode** combo box:

- **Color1** — color 1 only,
- **Color2** — color 2 only,
- **Color1/2** — colors in sequency beginning from 1,
- **Color2/1** — colors in sequency beginning from 2,
- **Color1+2** — mixed color,
- **Color1+** — flashing brightness of color 1,
- **Color2+** — flashing brightness of color 2.

Chapter 3.Axes setup

3.1. Overview

Maximum axes number that controller can process is eight. The following rotation sensors can be used:

- digital D_MaRS sensor,
- A_MaRS sensor (Gladiator family, 4-wired),
- V_MaRS (virtual, Gladiator yaw axis),
- analog sensors, potentiometers,
- encoders,
- buttons.

Encoders and buttons are used for virtual axes.

Open **Profile — Axes** tab (Fig. 3.1) to configure axes.

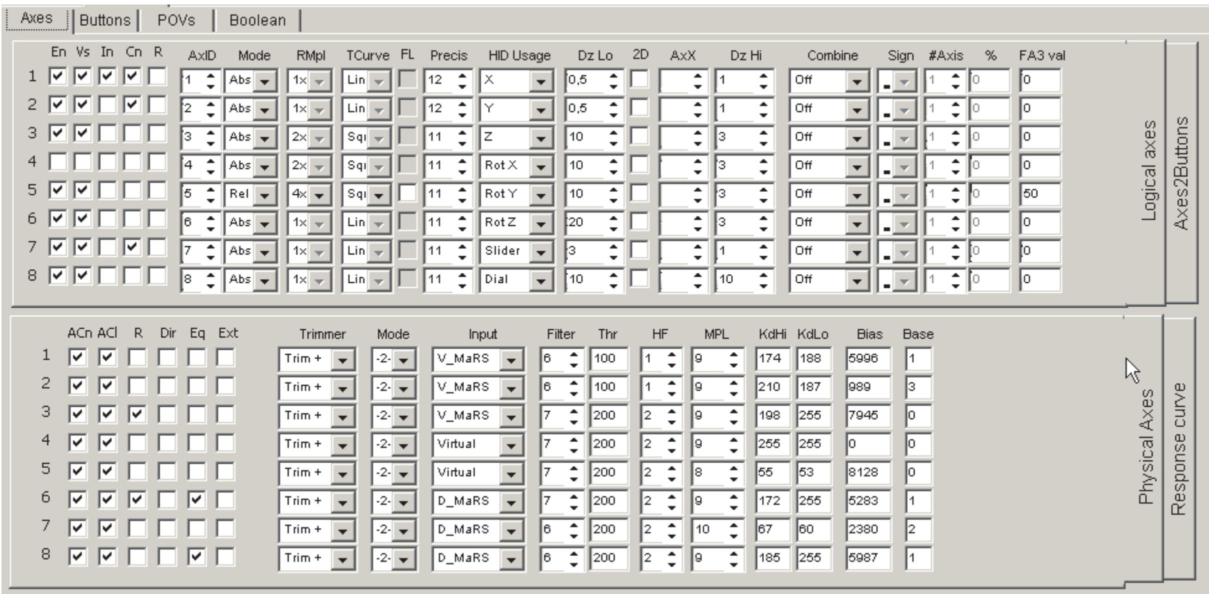


Fig. 3.1.

Every axis has a set of parameters, physical, logical etc.

3.2. Fhysical axis parameters

3.2.1. Parameter description

Physical Axes tab (Fig. 3.1) contains control items to configure physical parameters. See their description in Table 3.1.

Table. 3.1. Physical axis parameters

Control item	Parameter description
ACn	AutoCentering. Axis center is detected automatically with every controller start.
ACl	AutoCalibration. Axis is calibrated automatically with every controller start. Must be checked to allow user to calibrate it (see 3.8 on p. 46).
R	Physical data inversion. Physical axis rotation reverse.
Dir	Direction. Controls normal MaRS rotation direction. Is used to calibrate axis without center with angle range less than <175°.
Eq	Equalizer, Response curve. Enables response curve adjustment (see 3.6 on p. 40).
Ext	Axis to external device. Axis response can be transferred to external device. For example to use Joystick1 Rx axis to trim Joystick 2 X axis, check **Ext** for it (Rx).
Trimmer	Trimmer enable. Axis can be trimmed. Variants **Trim+** и **Trim–** specify trim direction.
Mode	Trimming modes Choose trimming mode from **Trimmer mode** combo box. **Standard**: trimmed axis center will be shifted, the whole range unchanged, i.e. when you move grip to one extreme position axis response will not reach its maximal value. When you move the grip to another side response value will reach maximal value while grip is not at the end of range. **Modified**: trimmed axis center will be shifted. Both parts of axis response range will be changed so extreme position of grip will correspond to extreme axis response.
Input	Type of input. Axis sensor type: ▼ D_MaRS — Digital MaRS, ▼ Analog — potentiometer, ▼ Virtual — virtual axis, ▼ A_MaRS (Gladiator MaRS type), ▼ V_MaRS (Gladiator family yaw sensor). Wrong sensor type will cause axis malfunction!

Table. 3.1. Physical axis parameters

Control item	Parameter description
Filter	Filter grade. Sensor filtering level for dynamic filtration. Default value for D_MaRS is equal to 5, for analog sensors – 6. If filter value = 0 than Manual filtering is used instead of dynamic.
Thr	Threshold of dynamic filter. Dynamic filter operation threshold. **DF** checkbox enables dynamic filter (see 2.3 on p. 19). You can set threshold level value in range from 0 to 255. Default value for D_MaRS is equal to 33, for analog sensors — 55. If **Trh**>0 than **MF** field value is used as upper dynamic filter value.
HF	H-Filter. Filter is named by Hruks (newView author), dynamically moving filter. Additional filtration for tremor compensation. It is advisably to use for analog axes especially with «noisy» potentiometer.
MPL	Multiplier. Binary multiplier for sensor response normalization. Value range is -15...+15. Default value for D_MaRS is 9, for analog sensors — 8.
KdHi	Coefficient of gain high. Factor specifying (together with **MPL**) upper level of axis response.Value range is 0...255. Default value for D_MaRS is 190, for analog sensors — 255.
KdLo	Coefficient of gain low. Factor specifying (together with **MPL**) lower level of axis response.Value range is 0...255. Default value for D_MaRS is 190, for analog sensors — 255.
Bias	Bias zero point. Specifies sensor initial position for automatic calibration at the controller start. Is used for manual calibration.
Base	Base quadrant. D_MaRS sensor base quadrant for automatic calibration at the controller start. Reference parameter.

3.2.2. Axis response filtering

High sensitivity of sensors and gimbal precision make significant hand tremor. Responce filtering allows to compensate this noise. Static or dynamic filtering can be used.

Static filtration

Uncheck **DF** to use static filtration (see 2.3 on p. 19). Use **MF** field with counter to set filtering factor. Filtering factor for static filtering is constant in all gimbal moving range.

Dynamic filtration

Check **DF** to use dynamic filtration. Filtering factor for dynamic filtration depends of gimbal deviation. Upper factor value (filter threshold) is equal **Trh** field value.

By default it is equal to 18 for 8000 counts of one way sensor response value or ~0,2% of gimbal rotation. This means that for small deviations less than 0,2% of maximum angle filtering is maximal. If deviation exceeds specified value filtration rate very rapidly falls to minimal value. If difference between sensor response counts is less than **Trh** field value, filtration factor value will gradually grow to specified value.

Set **Thr** field value equal to 0 for specific axis to disable dynamic filtering for it. Static filtering will be used instead.

> The more filtration grade you set, the more inert will be axis response. If you want maximal sharp response set **DFT**=0, **Filter**=1.

3.3. Logical axes

Use control items of **Logical Axes** tab (Fig. 3.1 on p. 31), to configure logical axes parameters. See parameters description in Table 3.2.

Table. 3.2. Logical axis parameters

Name	Description
En	Enabled, on/off axis. Enables axis. Axis parameters are calculated even if it is invisible (**Vs** unchecked). This mode you can use to convert axis rotation to button press sequence without axis response.
Vs	Visible in HID. Makes enabled axis visible. Thus to use any axis you must set it as enable and visible!
In	Logical inversion of axis. Inverts axis rotation. Instead of physical inversion (see 3.2.1 on p. 31, **R** checkbox), that is applied to axis response at the beginning of output signal processing logical inversion is applied at the end. In most cases the result will be the same but sometimes inversion mode can be significant.
Cn	Axis with center. Locates lower deadzone position – in the middle of axis range for axis with center (checked) or by extreme positions (unchecked). MUST be checked for analog sensors.
R	Physical data inversion. Inverts physical layer axis response.
AxisID	ID of binded physical axis. Binds logical axes to physical ones. Single physical axis can be binded to several logical axes.

Table. 3.2. Logical axis parameters

Name	Description
Mode	Absolute/Relative mode of axes input. Axis mode, absolute or relative (see 3.4 on p. 35).
RMpl	Relative mode multiplier. Multiplier for relative axis.
TCurve	Relative mode response curve type. Response curve type for relative axis.
FL	Forced limit. «WEP limit switch».
Precis	Precision of axis. Axis response digital capacity, bits.
HID Usage	HID axis name. It is NOT recommended to change default names. Wrong name can cause axis malfunction.
Dz Lo	Deadzone in center or bottom of axis. Size of deadzone in the center of response range (axis with center) or by lower limit (axis without center).
Dz Hi	Deadzone in top of axis. Size of two deadzones by upper and lower limits of response range (axis with center) or by upper limit (axis without center).
Combine	Type of combine Axis. Combining axes type (see 3.5 on p. 36).
Sign	Sign of combine axis. Axes combining direction (see 3.5 on p. 36).
#Axis	Number of the axis that forces on current one.
%	Maximum combining effect.
FA3 val.	Fixed value for FA3 mode. Fixed axis response value on button with FA3 function (see 4.3.20 on p. 82).

3.4. Relative axes

3.4.1. Overview

Response of common **absolute** axes corresponds to grip position. Usually absolute axes are used to control plane (roll, pitch, yaw, brakes), engine (RPM, throttle, radiators) etc. It is reasonable to use **relative** axes for ministicks. The rules for relative axes response are:

- Grip move direction determines response direction.
- Grip deviation value determines response speed. The more deviation the more speed.

When you release grip it returns to the center automatically under the impact of springs, but response remains in the last point. You must use separate control unit

(**RARst** button, see 4.3.22 on p. 87) to reset axis instantly or move grip to opposite side.

3.4.2. Relative axis parameters

Common parameters

Select **Rel** from combo box **Mode** on **Logical axes** tab. Select response multiplier from **Rmpl** combo box. **Lin** and **Sqr** variants of **Tcurve** combo box specify response curve type, linear or square-law. Use control units of **Response curve** tab (see 3.6 on p. 40) for fine tuning.

Forced limit configuration

If **FL** is checked, than when you quickly move the grip to its limit response value instantly will be set to maximum. As opposed to absolute axis response of relative one does not depend of stick position directly. Even if you moved it to extreme position very fast, response will grow with constant speed. Checking **FL** you can maximize it instantly. Configure FL using controls of **Global – Common** tab (Fig. 3.2).

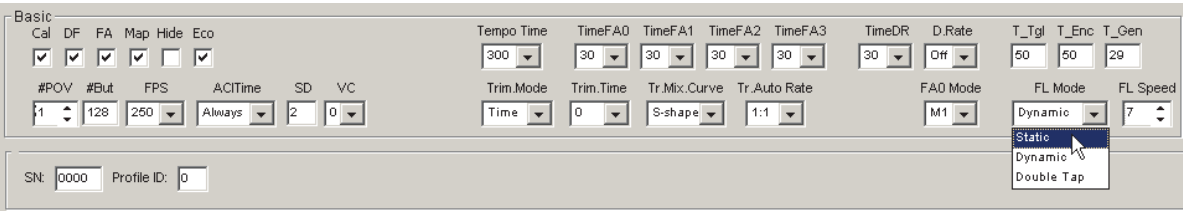


Fig. 3.2.

Static mode from **FL Mode** combo box ignores stick movement velocity, it works with any. If **Dynamic** mode selected, FL behaviour depends of **FL Speed** value. If **FL Speed** = 0 FL works as in **Static** mode. If **FL Speed** > 0 instant jump to maximum occurs on quick stick movement. The greater **FL Speed** value selected, than faster stick must be moved. It is usable to put stick to the center and throw it aside. Smooth movement won't cause FL work. Use double stick jump to engage FL in **Double tap** mode.

3.5. Axes combining

3.5.1. Overview

Axes combining allows to «rotate» one axis using another one. There is a difference between this kind of axis control and using button (see 4.3.18 on p. 78) or encoder (see 4.3.10 on p. 64) as axis trimmer. Using buttons or encoders you must not create an axis to change response of existing one. When you combine axes you must have forcing axis, physical or virtual.

You can trim aircraft axes which can not be trimmed in reality. Cheat detected)))

3.5.2. Combi modes

Problem definition: how to trim axis #1 (roll) using axis #8?

To resolve this problem do the following actions.

1. Open **Profile — Common-n-Axes — Logical Axis** tab.
2. Choose **Combi1** item from **Combine** combo box for axis #1.
3. Enter number of forcing axis, 8 in this example, using **#Axis** counter.
4. Set maximum combining effect using **%** counter. This value determines trimmed axis center shift if forcing axis will be moved to its extreme position. 50% moves center to range limit. It is recommended to use 20-25%.
5. Set center shift direction choosing item + or - from **Sign** combo box (Fig. 3.3).

Fig. 3.3.

6. Press **Set** button to save settings to controller memory.
7. Check axes combining using VKBJoytester.

   Fig. 3.4 shows axes #8 (pink) and #1 (black) response if axis #8 is rotated. Physically axis#1 is idle.

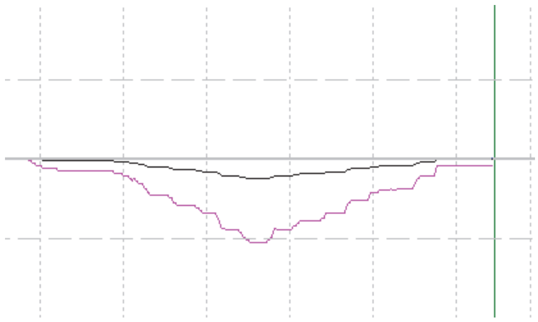


Fig. 3.4.

3.5.3. Dir modes

Dir1 and **Dir2** items of **Combine** combo box allow to align centers of two combined axes. Combined axes work is shown on Fig. 3.5.

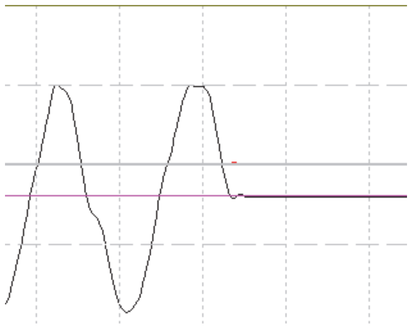


Fig. 3.5.

Dir1 and **Dir2** items differ by response sign (shift direction). Fig. 3.6 a) and b) show response directions for **Dir1** and **Dir2** items. Physically both combined axes were rotated in the same directions.

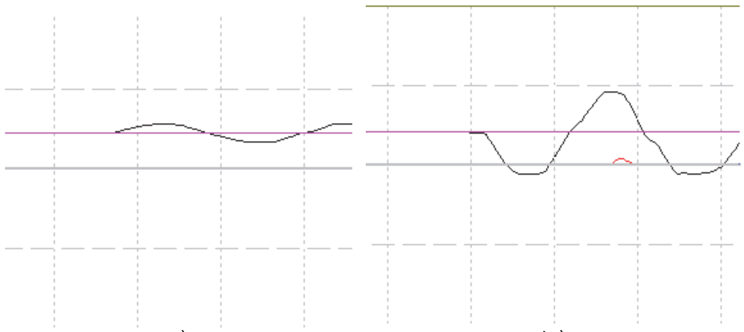


Fig. 3.7.

3.5.4. BrakeV modes

BrakeV

This combining mode is used for toe brakes. Default mode for TinyBox controller is shown on Fig. 3.8.

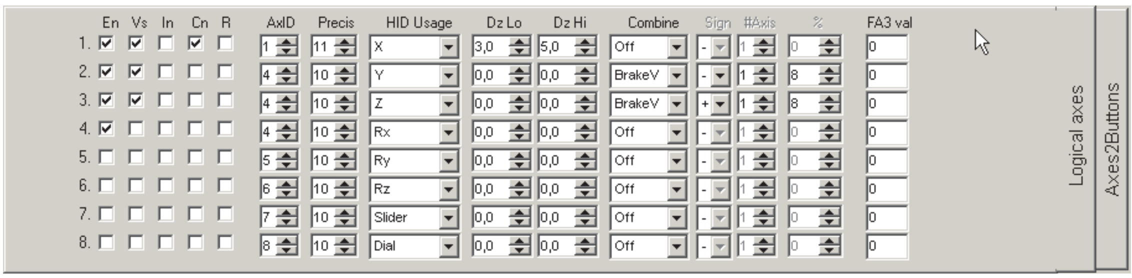


Fig. 3.8.

T-Rudder single axis is X, yaw. Virtual axis #4 is controlled by virtual button (use T-Link program to control this button). One of another joystick physical buttons is transferred to Tiny Box as virtual one. TrA+ function (see 4.3.18) is assigned to this button (Fig. 3.9).

Fig. 3.9.

Axes #2 and #3 have the same **AxID** as #4. So when you press button all three axes will response. Axis#4 is invisible (**Vs** is unchecked). Thus right and left brakes are realized (if you assign corresponding axes in simulator). Brake axes #2&#3 are combined with axis #1 (physical yaw axis) using **BrakeV** mode and 8% of range. If button is pressed both brakes work simultaneously. If you move rudder pedal, and its response will exceed 8% corresponding brake will be released immediately. Thus using single axis pedals you can control three axes.

See result in Fig. 3.13.

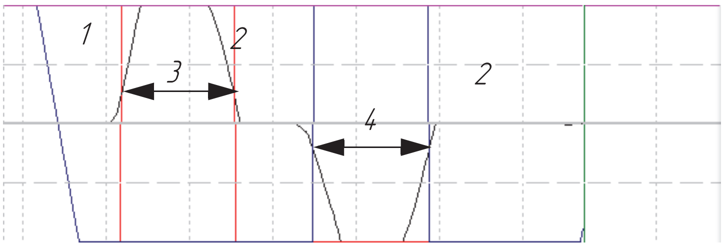


Fig. 3.10.

1 – response curves of both virtual brake axes (coincide).

2 – rudder axis response.

3 – first brake response alteration.

4 – second brake response alteration.

BrakeV1

Virtual brake axes combined with rudder with BrakeV1 mode. On rudder pedal moving differential brake of corresponding direction is engaged. It begins when rudder axis response exceeds value of **%** field (Fig. 3.11).

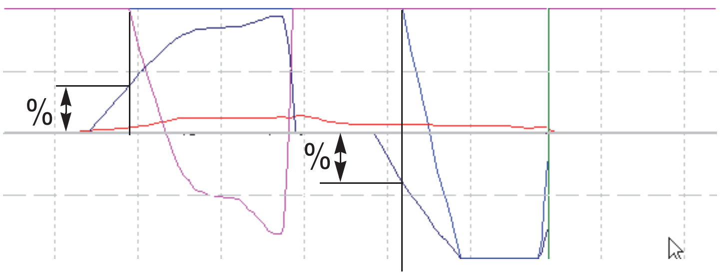


Fig. 3.11.

On brake button press both differential brakes work. On rudder moving idle axis will reach engaged then they will move together.

On brake button release both axes will be released. Idle to zero engaged to value dspsnding on rudder pedal response value.

BrakeV2

BrakeV2 combining mode is similar to BrakeV. But on brake axis moving both virtual brake axes will be engaged with dead zone. Response of two different combining modes, BrakeV2 and BrakeV, use is shown on Fig. 3.12.

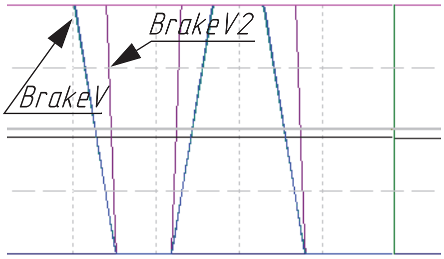


Fig. 3.12.

BrakeV3

Like in previous section when you press button all three virtual axes will response. Axis#4 is invisible (**Vs** is unchecked). If you move rudder pedal, and its response will exceed specified value (8%) corresponding brake axis response will not be released immediately. It will decrease according with rudder axis. See result in Fig. 3.13.

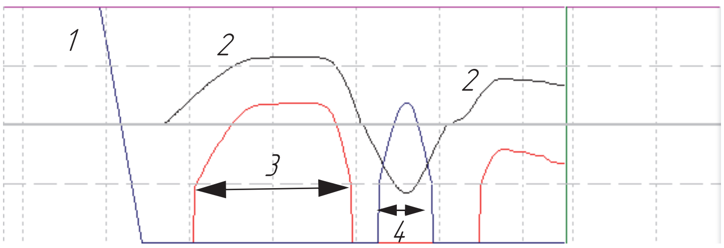


Fig. 3.13.

1 – response curves of both virtual brake axes (coincide).

2 – rudder axis response.

3 – first brake response alteration.

4 – second brake response alteration.

3.6. Responce curves

Use control items of **Response curve** tab to customize axes response curves (Fig. 3.14).

Fig. 3.14.

Make the following actions to customize axis response curve. Check **EQ** to show equalizer. Select (click) desired axis. Click curve panel. **Curve wizard** dialog appears (Fig. 3.15).

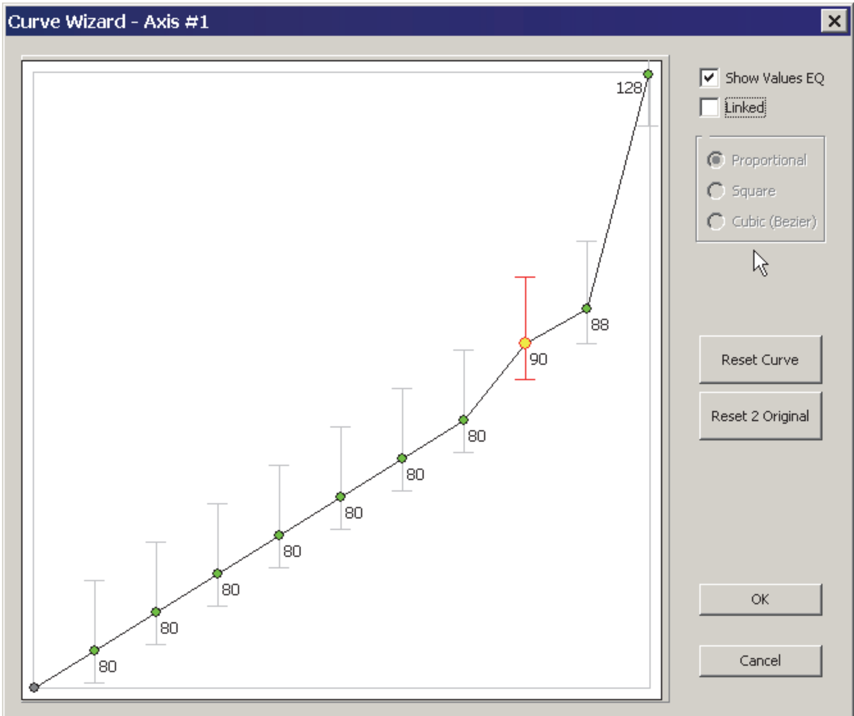


Fig. 3.15.

Check **Show Values EQ** to show equalizer values. Check **Linked** to link sliders (disabled in this example). **Proportional** item (enabled if **Linked** checked) allows to move sliders separately (Fig. 3.16).

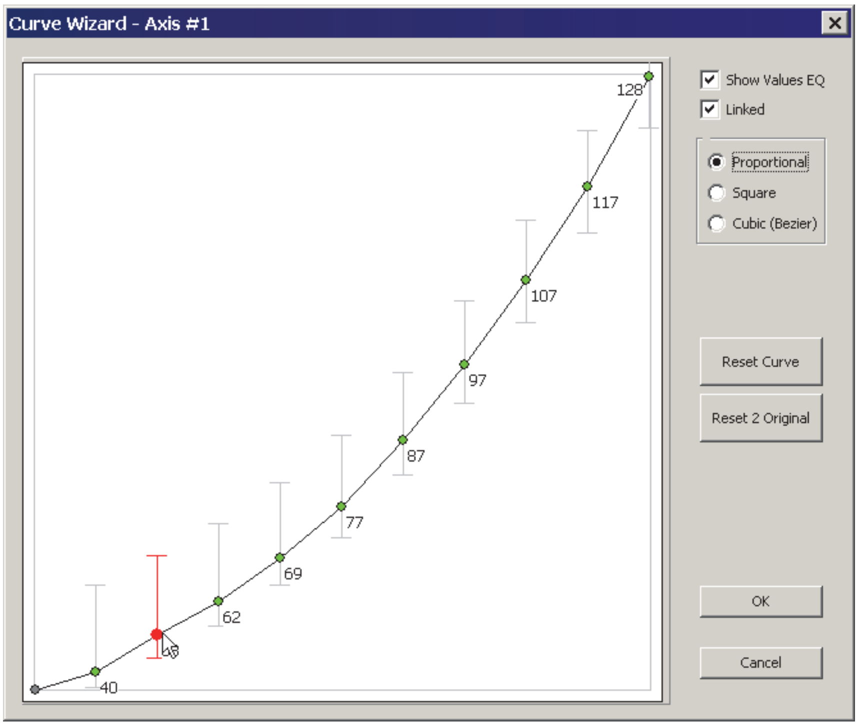


Fig. 3.16.

On **Square** item chosen middle slider enabled for moving. The rest will move after it, forming smooth curve (рис. 3.17).

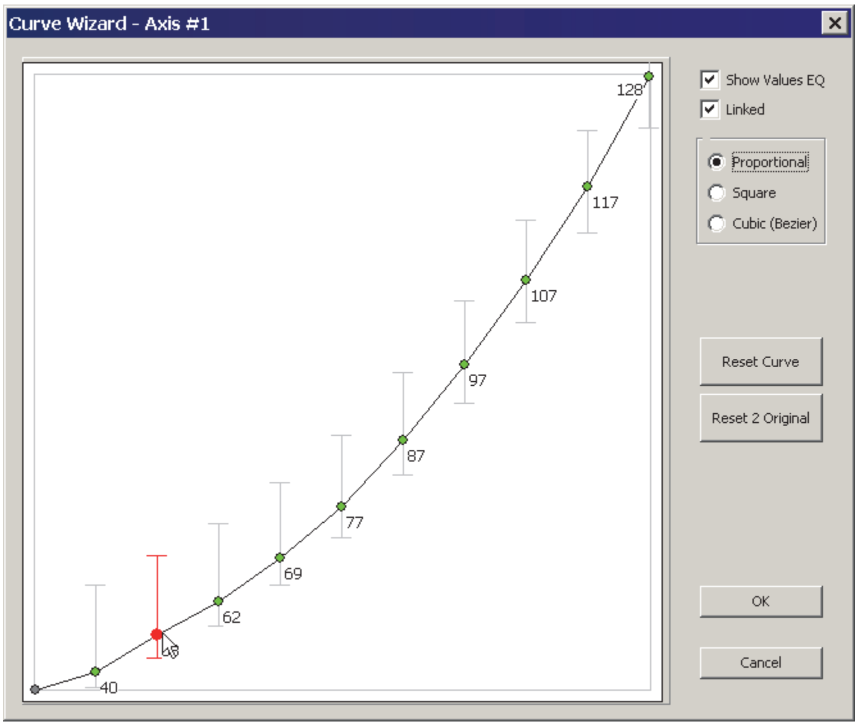


Fig. 3.17.

On **Cubic (Bezier)** item chosen sliders will form Bezier spline (Fig. 3.18).

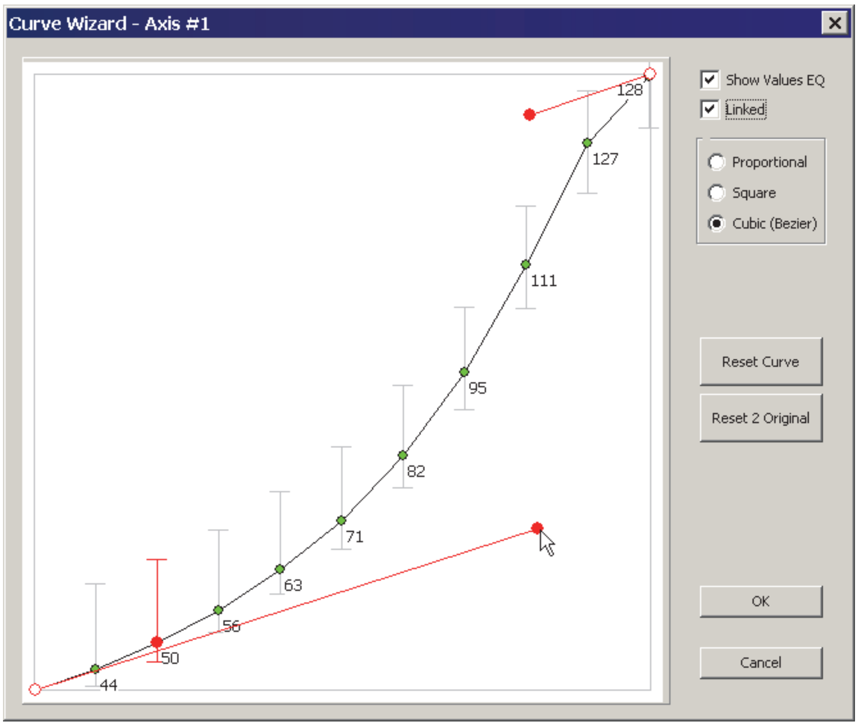


Fig. 3.18.

Press **Reset curve** to reset curve to default shape of straight line. Press **Reset 2 original** to return curve to previously configured shape.

Press **OK** button to complete configuring. Resulting response curve will be shown on the curve panel. Press **Cancel** to cancel results and close dialog.

To permanently apply customized curve to an axis check **Eq** for it (see 3.2.1 on p. 31).

To apply curve temporary use button with **CrV** (**CrVa**) function (see 4.3.11 on p. 67).

Also you can use simplified axis response control with fixed reducing rate.

- Choose reducing rate from **D.Rate** combo box (see 2.4.6 on p. 21),
- Assign **DR** function to a button (see DR on p. 83).

When you will press this button (aiming, for example) axis response will be decreased.

3.7. Conversion axis rotation to button press sequence

3.7.1. Common parameters

You can convert any axis rotation to the sequence of button pressings. The whole axis response range is divided to some zones. Every zone will have corresponding button. While you move the grip axis response varies. When it comes to one of the zones, corresponding button will be virtually pressed. Control items of **Axes2Buttons** tab (Fig. 3.19) allow to configure axes-to-buttons conversion.

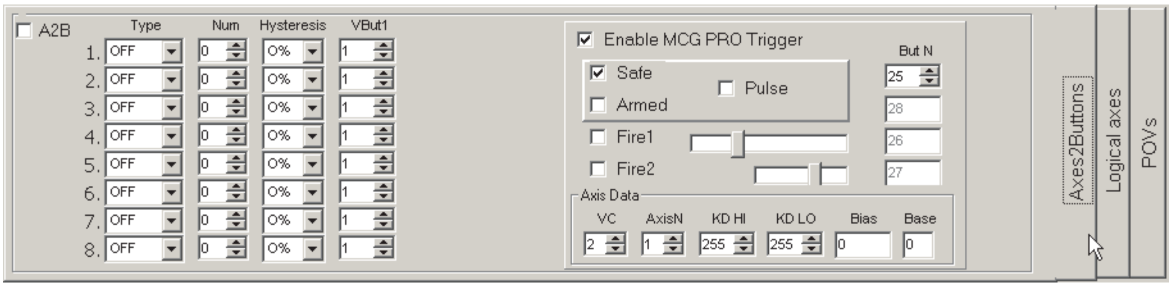


Fig. 3.19.

Choose axis response range division type from **Type** combo box.

- **OFF** — no conversion.
- **Edges1** — single button press on bound zone cross.
- **Edges2** — pair of buttons from both sides of zone bound will be pressed with bound cross.
- **Zones** — button is pressed while axis response belongs to the zone.

Set zone number using counter **Num**. 0 an 1 allow to use trailers. Button will be pressed at extreme axis positions. If **Num** = 0, lower trailer will work, if **Num** = 1, than upper.

If you set **Num** = 1 for **Edges2** then **both** trailers, upper and lower will work. If you use this function for throttle axis, when you move throttle lever to zero position, button for ignition or fuel pump stop and so on will be virtually pressed. Maximal position of this lever will press WEP button.

Histeresis combo box sets width of button actuation in percents of the whole response range. This parameter removes uncertainty of button work near boundary. If **Histeresis** = 0, the button will be pressed exactly at the boundary. Set the first but- ton number of sequence using **Vbut1** field. The following button numbers will in- crease to **Num** field value. For example four zones were created (**Num** = 4), **VBbut1** = 89 and **Zones** type was chosen. When you will rotate axis without center between extreme positions buttons 89, 90, 91, 92 will be pressed consequently. You can map keys to these buttons. If **Num** = 0 or 1 (trailers) in both cases will be pressed button with number equal to **VBut1** field value.

For throttle control axis you can stop engine automatically when throttle will be down. Set **Num** = 0 and assign button with stop engine function to **VBut1** field value. Be careful not to stop engine in flight when you throttle down diving your «Stuka».

3.7.2. MCG Pro folding trigger settings

MCG Pro folding trigger is a lever on axis with MARS sensor. It has two stable positions, folded forward (safe) and down directed (armed). If you press trigger firmly, physical button is activated. On trigger pass up to four virtual buttons can be activated too.

Check **Enable MCG PRO trigger** to enable virtual buttons. Check **Safe**, **Armed**, **Fire1**, **Fire2** to control specific virtual buttons. **Safe** button corresponds to folded position, **Armed** – down directed. **Fire1** and **Fire2** will be activated on trigger pass.

Assign **Safe** button line number using **But N** counter. Line numbers of other buttons will follow. Check Pulse to generate short pulse when button is virtually pressed and even stays in this position (**Safe** or **Armed**). Pulse duration is equal to **T_Tgl** parameter value (see 2.3.1 on p. 19).

Sliders **Fire1** and **Fire2** allow to set trigger position for corresponding buttons activation. Left limit of **Fire1** slider roughly corresponds to **Armed** position. Physical button will be activated near right limit. It is reasonable to distribute virtual buttons evenly by trigger pass. **Axis data** group control units allow to set up trigger axis similarly to other ones.

It is not recommended to change axis default settings.

3.7.3. MCG Pro Brake lever setup

MCG Pro Brake lever works as axis with contactless MARS sensor. Also virtual buttons can be assigned to its race. Controls shown in Fig. 3.20 allow to setup these buttons similarly but easier to usual Axis To Buttons function.

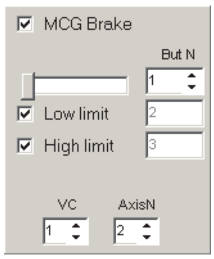


Fig. 3.20.

Check **MCG brake** to enable virtual buttons. Checkboxes **Low limit** и **High limit** allow to control virtual buttons and corresponding to extreme lever positions. Assign lever position for button intermediate button using slider. Set number of controller physical button that will correspond to intermediate button using **But N** counter. Buttons #2 and #3 will have sequent numbers.

VC counter allows to choose virtual controller of brake lever axis, by default **VC**=1. **AxisN** counter allows to choose brake lever axis number, by default **AxisN**=8.

You can set virtual buttons connected to other axes similarly.

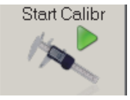
3.8. Axis calibration

3.8.1. Autocalibration

Common parameters

Usually you can calibrate joystick axes automatically. Do the following actions.

1. Check **Cl** option on **Profile — Common-n-Axes — Physical Axes** tab for axes which must be calibrated.



2. Push **Start Calibr** button on **Tools** tab.

If zconfig.ini file contains *PartialCalibration=1* string, you will see **Partial Calibration Settings** dialog (Fig. 3.21).

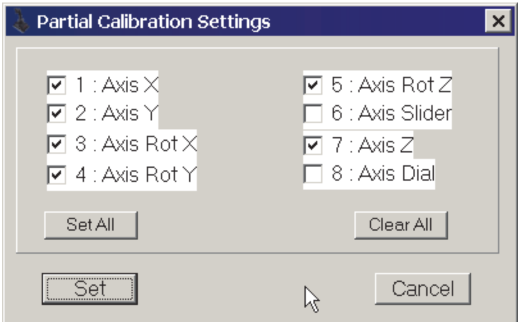
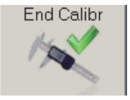


Fig. 3.21.

Check desired axes calibration and press **Start** button.

Lightning mode of LED indicator will be changed (See 2.5 on p. 25), if you have made this setting.



3. Rotate every calibrating axis between extreme positions.
4. Push **End Calibr** button.

If for some reason you want to cancel calibration without saving results, push **Cancel Calibr** button.

MCG Pro folding trigger calibration

MCG Pro folding trigger is an axis. Before you begin to calibrate it, fold it forward (safe position) then return to down directed (armed). To calibrate trigger, after you press **Start calibr** button fold it forward then rotate back and press firmly. That is all.

3.8.2. Manual calibration

Overview

In some cases the results of autocalibration may be insufficient. For example values of **KdHi** и **KdLo** may be equal to 255. Such values are too great and desensitize axis precision (best results are with values about 120 — 180). Or neutral position of the grip does not correspond with the middle of the response range. In this case it is recommended to perform manual calibration. To make this operation it is reasonable to use VKB Joytester program. Download it from VKB site http://ftp.vkb-sim.pro/Programms/VKB_JoyTester.zip. Unpack downloaded archive in the same folder with other VKB utilities.

Joytester interface

Run *VKB_JoyTester.exe* file. The window of this utility is shown in Fig. 3.22.

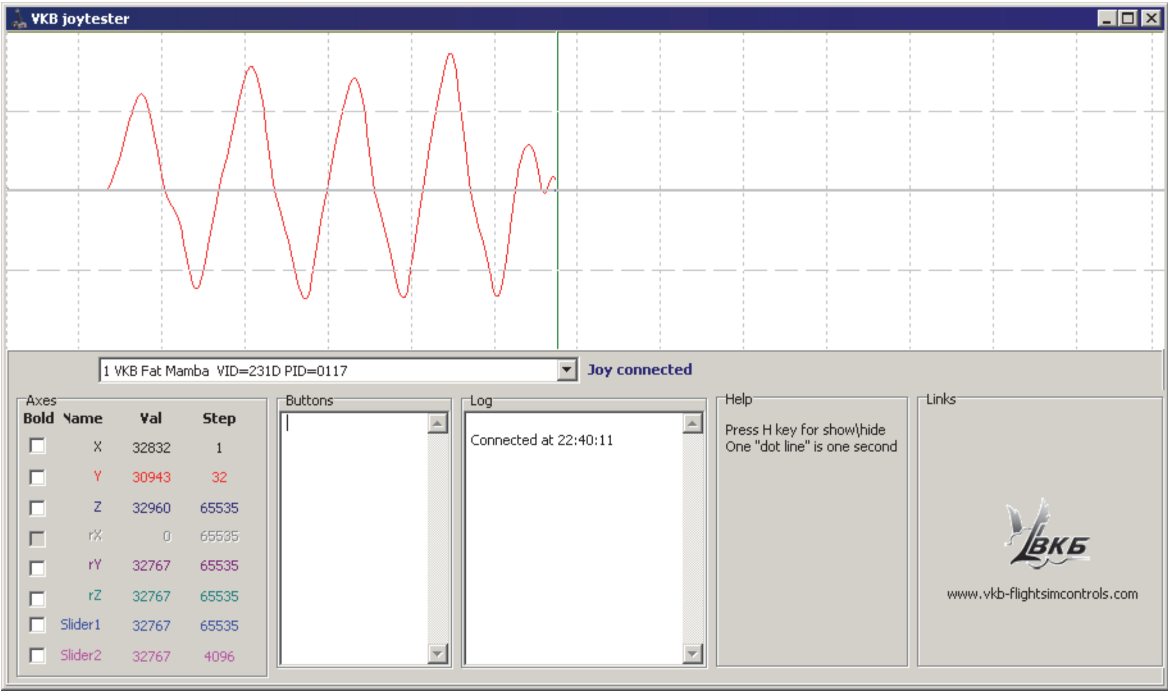


Fig. 3.22.

Select joystick name in **Joy connected** combo box. The most window area occupies axes response diagram. When you move grip, you will see graphic response and its digital value. Check **Bold** to draw axis response with bold line.

When you push buttons you see their numbers in **Buttons** field.

If special function (i.e. Shift or Fix Axes etc.) assigned to the button, you will not see its number.

Axis center correction

If an axis has centerpoint, then when you release it response value of this axis must be equal to 32767 (one half from 65535, maximal value). Inaccuracy in some digits or even tens are acceptable. But if it is too big and centerline of the axis does not match with graph center line, you must correct calibration. **Bias** parameter allows to compensate magnet and MARS positions for digital axis or potentiometer centering for analog one. Try to change **Bias** value about 100 — 150 units with + or - sign and press **Set** button on **Action** tab. Centerline position will be changed. Select such **Bias** value that when the grip stays in the center position response value is about 32767. Do not forget to push **Set** button every time you have changed **Bias** value.

Axis response range correction

Setup axis response range so in pedals extreme positions response value will be equal to 0 and 65535. **KdHi** and **KdLo** values must be in range 100 ... 180. This will provide optimal dynamic range.

Move axis between extreme positions. If response value is greater then 0 or less then 65535 or, on the contrary, the pedal is not in the extreme position but response value already is equal to 0 or 65537, you must correct the range.

Change **KdHi** value, press **Set** button, move the grip and check how axis response value in the extreme position has been changed. Select such **KdHi** value that when the grip is in extreme position response has extreme value too. If you change **KdHi** value but can not reach a goal (value is out of range 100 ... 180), try to change **MPL** value and repeat setup. Then you must setup the other range limit with the same manner, changing **KdLo** value.

While you setup response range, centerpoint can be moved. In this case setup it again.

Calibrate all axes (if it is needed) in the same manner. Calibration results can be saved to file (see 8.3 on p. 115).

Chapter 4.Physical buttons

4.1. Overview

When we describe joystick we use common term *Button*. Really it can be any device that can close two contacts. For example it can be tact switch, toggle switch, HAT switch, rotary switch, encoder etc. Every pair of button contacts is represented as single *line*. So simple button occupies one line, toggle switch On-Off-On – two lines, 4-way HAT — 4. Njoy32 controller can process up to 128 lines.

Speaking about joystick controls we must distinguish *input*, that is button, HAT etc. and *output* – the result of physical controls conversion, that is virtual axes, generators, trimmers, modifiers and even simple buttons too.

Control items of **Profile — Buttons** tab (Fig. 4.1) allow to set up joystick buttons.

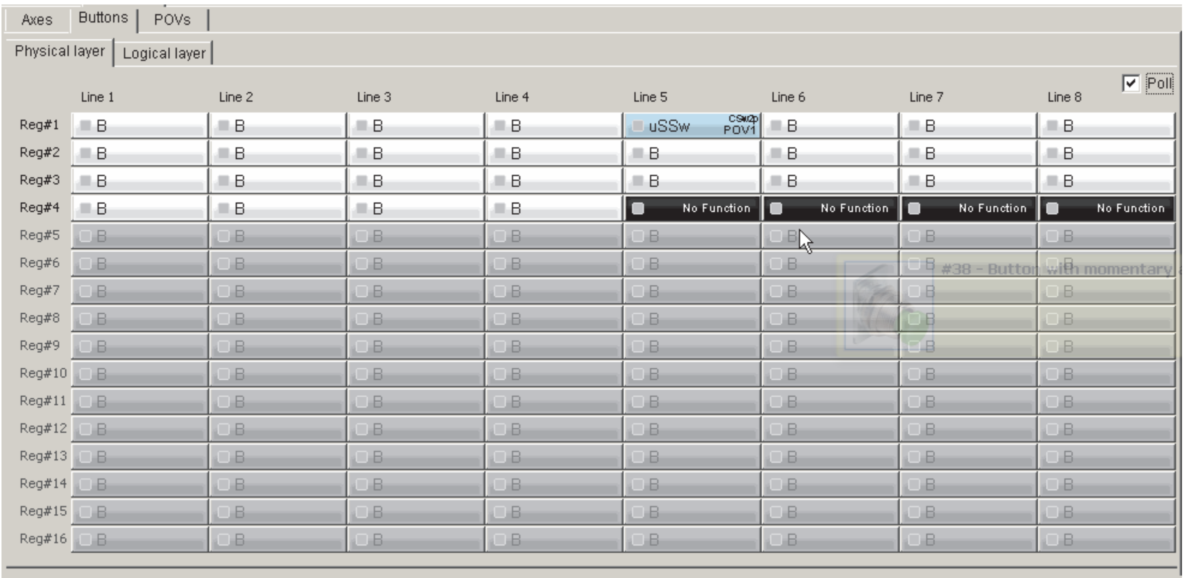


Fig. 4.1.

Every eight lines are grouped to registers, *Reg#1...Reg#16*. Lines have consequent numbers.

4.2. Physical button functions

Regardless of contact maker physical realization of its line can execute the following functions.

- Button — simple button,
- Button alternative — button with fixation,
- Radiobutton – button one of group,
- ButtonS — button under control of subshifts ##1-6,
- ButtonX — button under control of subshift #7,
- Shift — button modifier,

- SubShift — subshift, additional button modifier,
- Toggle — toggle switch,
- Encoder — encoder,
- POV switch — HAT switch,
- uStick switch — POV mode switch,
- Generator — generator,
- Tempo — two-function-button; output depends of pressing time,
- Fix_Axes — axes modifier,
- AuxAxes — axes mapping,
- RelAxes — relative axis mode control,
- Boolean — Boolean functions for buttons (OMG!!!),
- Cyclic Switch — cyclic switch,
- Trimmer — axis trimmer,
- Curves — dynamic axis response curve switch,
- Sync — toggle synchronizer,
- RPB — button replicator,
- OFF — expulsion button from the processing,
- NoF — no function.

4.2.1. Button mapping wizard dialog

Control items of **Button mapping wizard** dialog (Fig. 4.2) allow to assign any function to chosen physical button line.

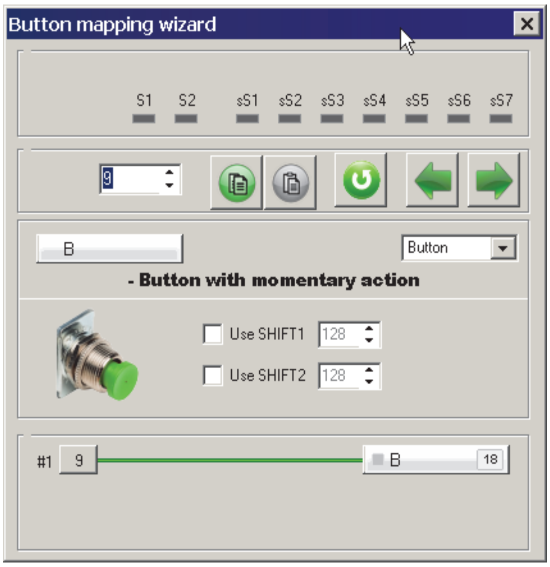


Fig. 4.2.

This dialog appears after left mouse button click on button cell.

Line (input control) choice

Parts of the string in **Physical layer** group show current line number, its register number and position in the register. Fig. 4.2 shows that current line is #9. It belongs to the first line of the second register. Press **Capture** button to check line and button correspondence (see 4.2.2 on p. 53).

Counter value shows line number too and allows to go to another button to set it up. **Next** and **Previous** buttons allow to go to corresponding lines. When you go to next button settings for current one will be fixed. Thus to go to distinct button you can close **Button mapping wizard** dialog and click desired button cell directly or use controls of this dialog.

Cancel buttons disables any current settings changes. For example you set button#9 as Shift and pressed **Next** button to setup next line. Current assignment for button #9 (*Shift*) will be fixed. Then you return to button #9 and choose *BA* function. Press **Cancel** button to restore previous (*Shift*) function.

Copy cell button allows to copy current line parameter set to clipboard.

Paste cell button allows to apply parameter set from clipboard to current line.

Output function choice

Choose output function for current line from combo box. Additional control items for this function will appear in dialog. Some functions, for example, simple button, allow alternatively using with Shift modifier. For those functions checkboxes *Use Shift 1* и *Use Shift 2* will be enabled.

You can use two independent modifiers *Shift 1* and *Shift 2.*

If for example *Use Shift 1* is checked, you can select the second line number that will be pressed if you press physical button simultaneously with Shift.

Example. Current button line number is 9. For shifted button you can choose 28.

Assigning additional line be careful. This number must NOT be the same with existing physical button line. But you CAN assign such (occupied) number. If this button line number was reassigned too. For example, gun trigger occupies physical line number 17. If you want to use it with «beautiful» number reassign it to 1.

Free line choice

To make sure that desired line number is free, double click line number counter with left mouse button. Dialog **Virtual layer** (Fig. 4.3) will appear. Red colored numbers are occupied, black – free. To choose number click it.

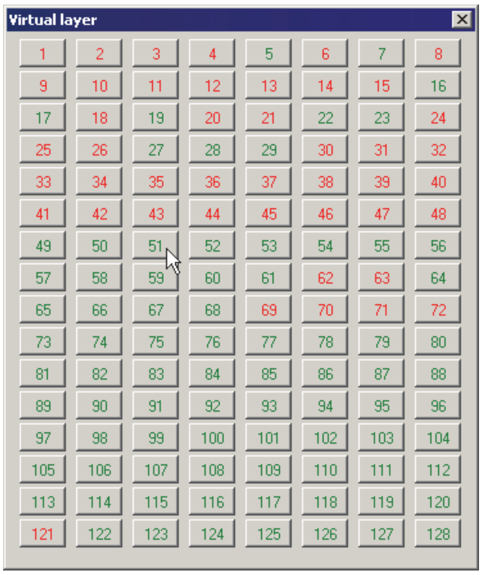


Fig. 4.3.

Logical (output) function choice

Almost all functions have subfunctons. Additional control items allow to customize parameters. Click current function field to see these controls (1 , Fig. 4.4).

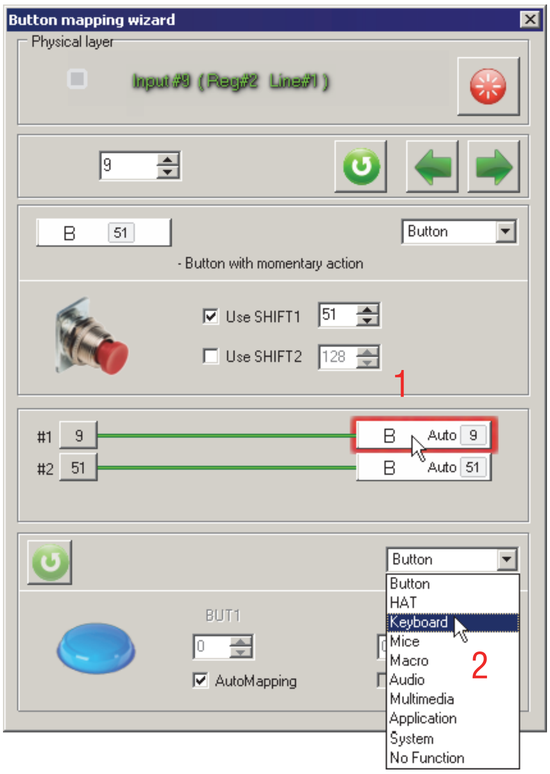


Fig. 4.4.

Subfunction names are listed in combo box (2 , Fig. 4.4).

Another way to get subfunction list is shown on Fig. 4.5).

Fig. 4.5.

Detailed description of logical functions you can find in Chapter 5 on p. 96.

4.2.2. Line number check

To see line number of any joystick control check **Poll** on **Physical layer** tab. After that if you press button, HAT, rotate encoder etc. cell of its line will flash.

Polling does not work if **Button mapping wizard** dialog is opened.

If **Button mapping wizard** dialog is opened you can check pressing too. If you press corresponding button it will be indicated.

4.3. Button customization

4.3.1. Button

Simple button

Description

When you press button by default output logical line number will correspond with physical one. To reassign logical number click current function name field (1 on Fig. 4.4 on p. 52). Additional control items will appear. **BUT1** field contains logical (output) line number. To remap button uncheck **Automapping**. **BUT1** counter will be enabled. Set desired output number (Fig. 4.6).

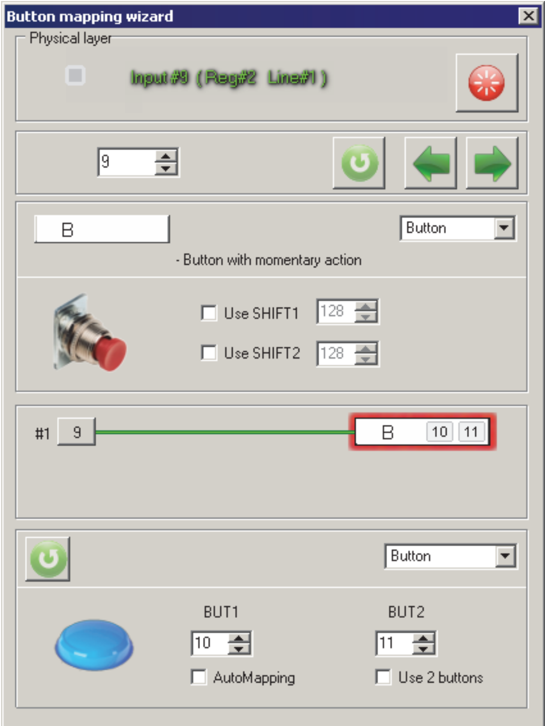


Fig. 4.6.

When remapping will be useful? Let's assume the current button number is greater than 32. Some games does not recognize button numbers upon this value. You can map your button to key or remap to line of available range.

Check **Use 2 buttons** to «press» two logical buttons simultaneously with physical one and set second line number using counter **BUT2**. See Fig. 4.6. When you press button with number 9 (Toggle switch Batt on the left side of Fat Black Mamba case) buttons 10 and 11 will work.

Modifier *Shift* can be used with simple button.

Physical button pressing can be indicated using LEDs (see 2.5.2 on p. 29).

Example.

Line 32 is shown on Fig. 4.7. It corresponds to the first trigger of HOTAS Warthog grip on Fat Black Mamba case.

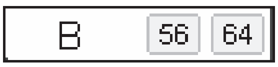


Fig. 4.7.

It is used as simple button (sign **B**). If you press trigger with Shift1 line 56 will work. For Shift 2 it will be line 64.

Restrictions:

NJoy32 controller can process up to 128physical buttons.

4.3.2. ButAlt

Button with fixation

Description

After you press and release button (input) BA will stay depressed (on hold) until you will press it again (output). You can assign another logical number for BA.

Restrictions:

Shift function not allowed.

4.3.3. RadioButton

Radio button, Selector

Description

Radio button allows to select only one line from specified group.

Button mapping wizard dialog for Radiobutton function is shown on Fig. 4.8.

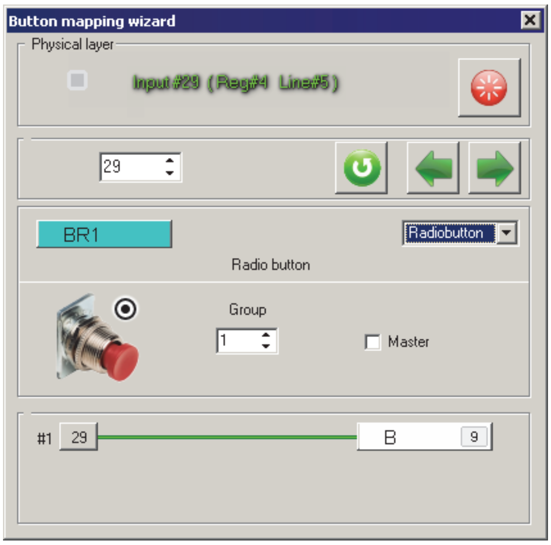


Fig. 4.8.

Set number of the group, to that belongs this button, using **Group** counter. If no group with this number exists, it will be created. Check **Master** to use this button as default. It will be «pressed» automatically on joystick start. If there are several master buttons in a group, only last assigned will be work as master.

You can assign logical functions to Radiobutton (see Logical (output) function choice on p. 52).

Example

Gladiator family joysticks have only one trigger. You can create a group of buttons, that will directly assign specific weapon to trigger. The first button allows machine-gun, the second – gun and the third, Master, i.e. default button, safety lock.

Assign weapon #1 toggle. Open **Physical Layer** tab, click desired cell, for example #48. Assign **RadioButton** function to it in group #1. Assign weapon #2 toggle for cell #49.

Assign safety lock for cell #50 in the same group and check **Master** in **Button Master wizard** dialog. Set logical function No function for this cell on **Logical layer** tab. By default your weapons will be locked.

4.3.4. ButtonS

Button controlled by subshifts 1...6

Description

Button controlled by SubSHIFT #1...6.

Set number of subshift that will control this button using **SubSHIFT #** counter and logical line number using counter **Button** (Fig. 4.9). By default it is physical one +1.

Logical line functions are the same with simple button.

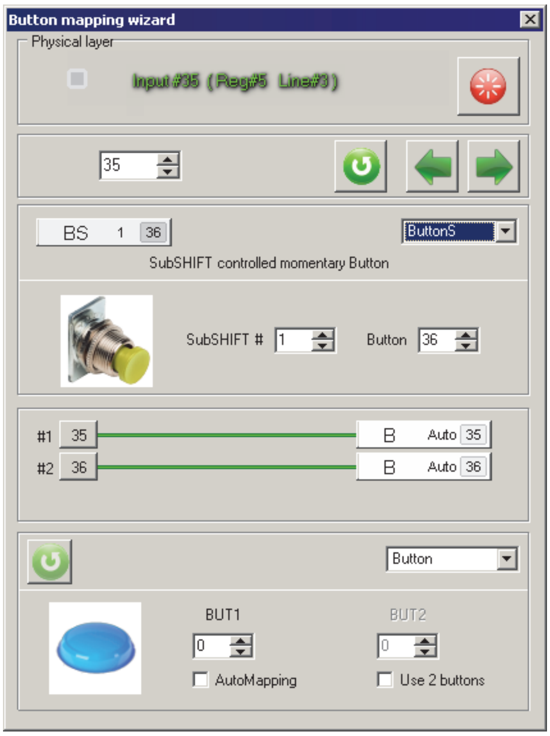


Fig. 4.9.

Subshift settings see in 4.3.8 on p. 60.

4.3.5. ButtonX

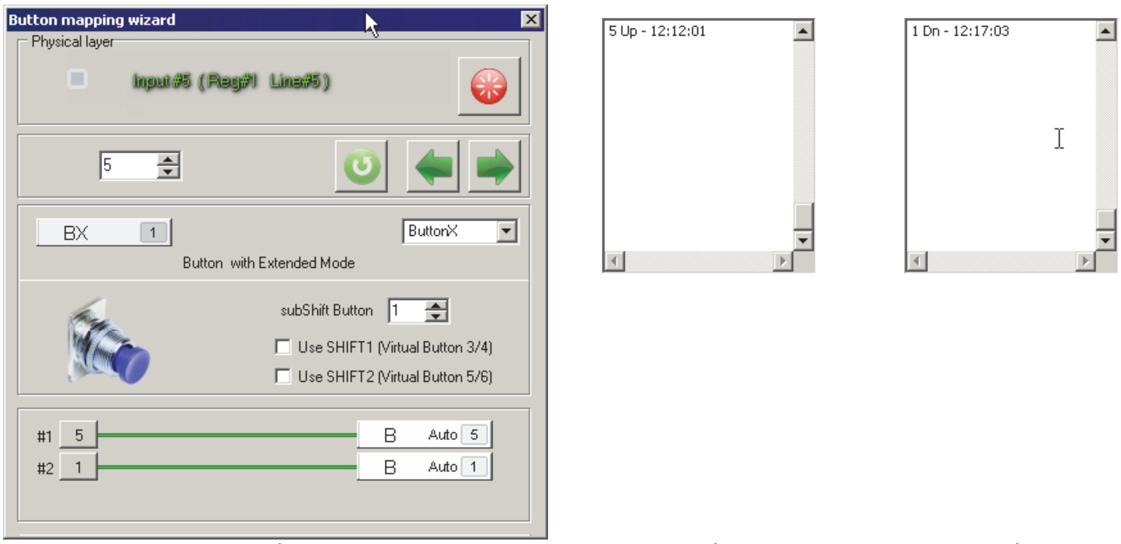
Button controlled by subshift # 7

Description

Button controlled by SubSHIFT #7.

The number of output line depends on combination of global Shift modifier and Sub-Shift local modifier.

if this button is not controlled by global Shift (Fig. 4.10, a), than output line number will correspond to physical one (Fig. 4.10, б). If you press button depressing Sub-SHIFT 7, output line number will be equal to counter **subShift Button** value (Fig. 4.10, в).



а) б) в)

Fig. 4.10.

If this button is controlled by global Shift (**Use SHIFT1 (Virtual Button 3/4)** checked, Fig. 4.11), than output line numbers will depend of Shift 1 и SubSHIFT 7 states and by default will go after line number assigned in **subShift Button** counter.

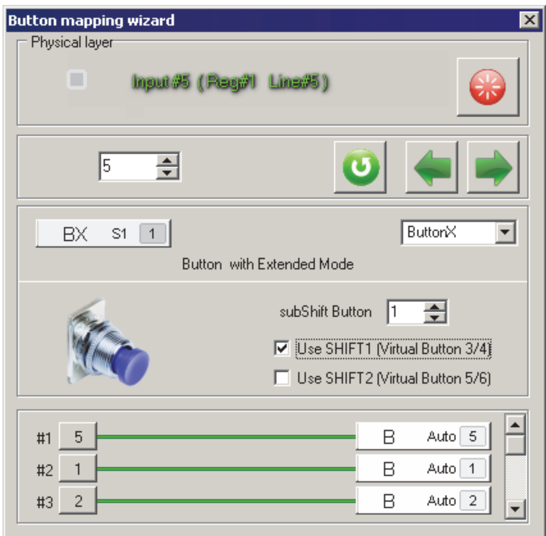


Fig. 4.11.

You can see an example of line numbers depending of modifiers state in table 4.1(for described figure).

Table. 4.1.

	SHIFT1	Off	On
SubSHIFT			
Off		5	2
On		1	3

4.3.6. ButtonD

Description

Button is pressed if linred button i.e. master button is released. May be used for example for two-stage trigger.When the second stage trigger becomes depressed, the first one will be released automatically. Button master wizard for dependent button is shown on fig.

Set Master button line number using **Master Button** field (Fig. 4.12).

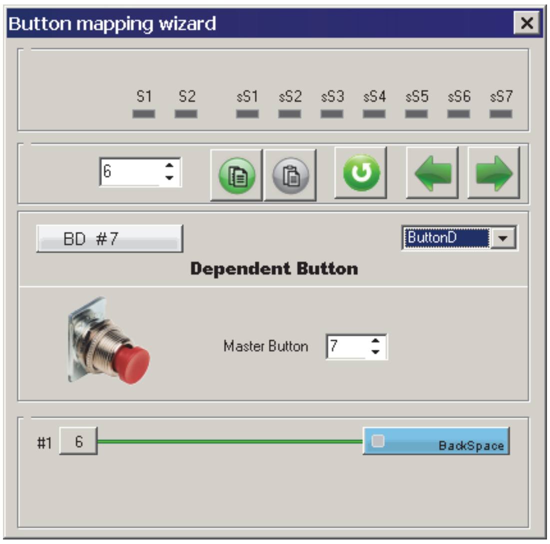


Fig. 4.12.

4.3.7. Shift

SHIFT1 / SHIFT2 / SHIFT0 modifiers

Description

Modifiers allow to multiply button number similarly to keyboard modifiers Shift, Ctrl, Alt.

Parameters

Modifier parameters are shown in **Button Mapping Wizard** dialog (Fig. 4.13).

Track as button checkbox allows to use button with **Shift** function as simple button too. In this case its parameters will be the same as for simple button (see 4.3.1

on p. 53). If checked then when you press SubShift it will modify dependent buttons AND work as joystick button too.

Select shift mode from combo box (Fig. 4.13, 1). Controller can process up to two shifts, so one input line can have three output ones. *Shift0* allows output signal if no other Shift (1 or 2) is pressed.

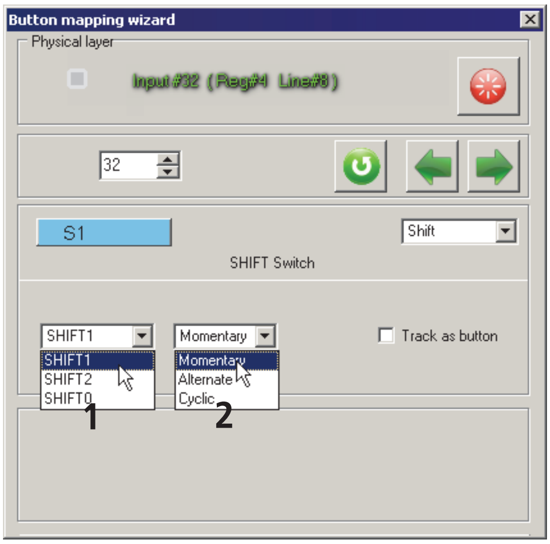


Fig. 4.13.

With *Shift0* you can, for example, use middle position of three-position slider on the Mamba family grip. Physically it is ON-OFF-ON toggle switch. It has no signal in the middle position. Assign *Shift1* and *Shift2* functions to both ON positions. Assign *Shift0* function to any button. **Track as button** will be checked automatically.

DO NOT map *Shift0* button to keystroke! It will work if no *Shift1* or *Shift2* are pressed that is practically permanent. Simple button even depressed generates continues but single output signal. Keystrokes will repeat thus flood system keyboard buffer. In this case you must disconnect joystick.

Shift1 and *Shift2* modifiers can be indicated by LEDs (see 2.5 on p. 25).

Shift1 и *Shift2* modes:

- Momentary — common, analog to Shift modifier,
- Alternate — with holding, analog to CapsLock modifier,
- Cyclic — cyclic.

Select mode from combo box (Fig. 4.13, 2).

In **Cyclic** mode every button press changes modifier type (**Shift 1** and **Shift 2**). For example button #9 was mapped to following keystrokes:

- without modifier — a,
- Shift 1 — b,
- Shift 2 — c.

Press Shift button and button #9. You will get letter <*b*>. Release buttons. Press Shift once again and button #9. You will get letter <*c*>.

4.3.8. SubSHIFT

SubSHIFT modifier

Description

When you press *Shift* button, it affects all buttons that have checked **Use SHIFT**. *SubShift* function is intended to affect limited number of buttons.

Button Mapping Wizard for **SubSHIFT** function has additional control items (Fig. 4.14).

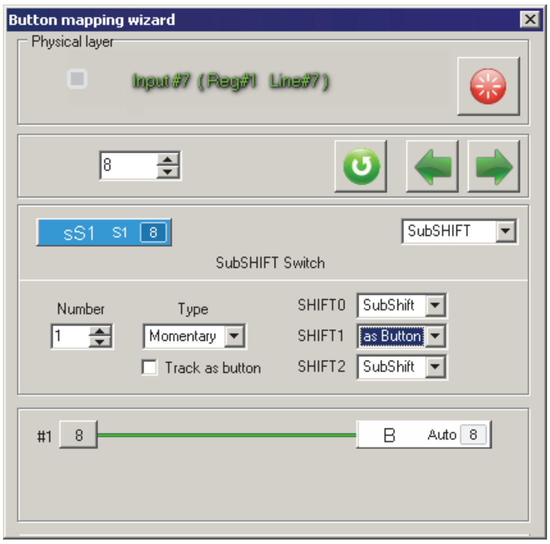


Fig. 4.14.

Set subshift number with **Number** counter. Maximum subshift number is equal to seven.

SubShift modes:

- Momentary — common, analog to Shift modifier,
- Alternate — with holding, analog to CapsLock modifier.

Button with **SubSHIFT** function may depend of **Shift** modifier. Select button mode from combo box for every Shif1 (0...2). When Shift is pressed SubShift button can work as simple button (shifted), or stay **SubShift**.

Track as button checkbox allows to use button with **SubShift** function as simple button too. In this case its parameters will be the same as for simple button (see 4.3.1 on p. 53). If checked than when you press SubShift it will modify dependent buttons AND work as joystick button too.

If **Track as button** is checked than button does not depends of global **Shift**.

Assign *ButtonS* function for buttons depending of SubShifts 1...6 (see 4.3.4 on p. 56).

Assign *ButtonX* function for buttons depending of SubShift 7 (see 4.3.5 on p. 56).

4.3.9. Toggle

Toggle switch

Description

When you close control with **Toggle** function a short pulse will be generated even if the line stays closed. Pulse length is specified by **Time of toggle pulse** global parameter (see 2.3.1 on p. 19). When you open control line with another number can work. By default this number is greater by 1 but can be changed. If you assign the same number than when you press and release control two pulses with identical numbers will be generated.

Button mapping wizard dialog for Toggle function is shown on Fig. 4.15.

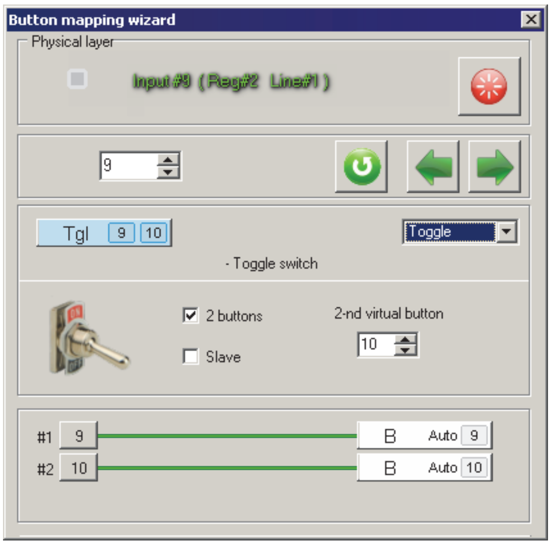


Fig. 4.15.

2buttons checkbox controls the ability of the second line work with opening control. If it is checked assign the second line number (for opening pulse) using **2nd virtual button** counter.

Three-position (On-Off-On) Toggle switch features

Synchronization

You can see three-position toggle switch on the back side of Fat Black Mamba case. It is named as **Flaps**. We use this switch to control flaps. When you push toggle stick down flaps go down, Upper position will retract flaps. Middle position does nothing. As it was mentioned above toggle signal is generated at a switch moment. When stick stays stable It does not know about its state. **Sync** function (see 4.3.26 on p. 93) allows to synchronize toggle switches state. When you press button with this function all toggle switches will be polled and controller will get there state. So if Flaps toggle was in the lower position in this time flaps will go down. There is an issue with middle toggle stick position. **Slave** function allows to get this position.

Lower position of Flaps toggle switch of Fat Black Mamba joystick corresponds to line number 13 and upper position to #14. For example assign line #15 to middle position.

Physically line #15 is used by **Pump** toggle switch situated on the left side of Fat Mamba case. We can assign another line number (logical) for this switch so it will not be lost.

Click cell #13. You will see **Button Mapping Wizard** dialog. Choose **Toggle** item from combo box. Check **2 buttons** and **Slave**. Set **Master Toggle** counter to 14 (Fig. 4.16).

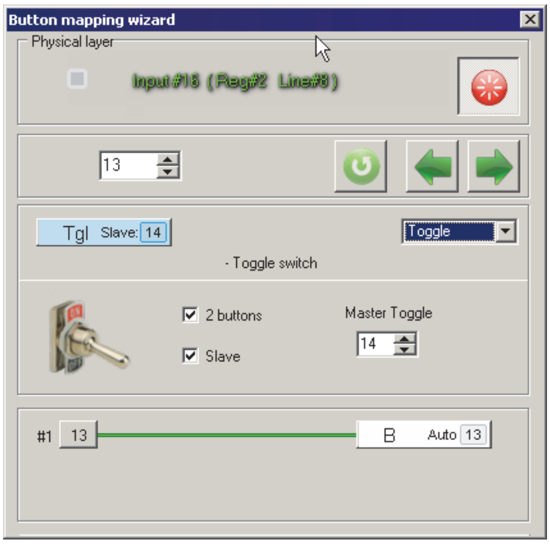


Fig. 4.16.

Go to next line settings. Check **2 buttons** and set **2-nd virtual button** counter to 15 (Fig. 4.17).

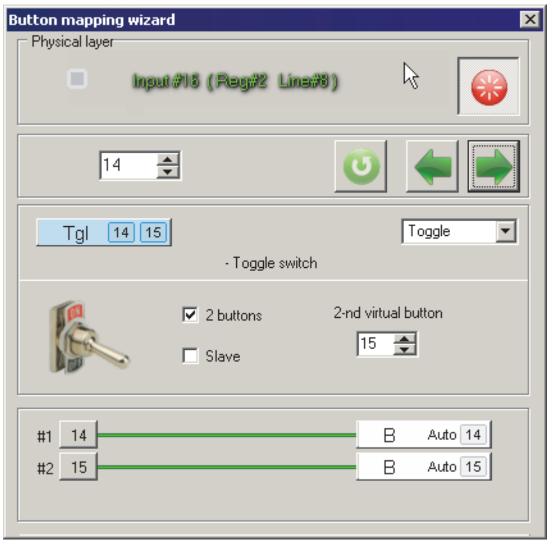


Fig. 4.17.

After you apply these settings pressing **Set** button Flaps toggle switch will generate the following signals:

- switching down — line # 13,
- switching to the middle from any position —line # 15,
- switching up — line # 14.

Physical button as toggle switch

Simple button can be configured as toggle switch. For what? Battle of Stalingrad simulator. You must press button to see briefing. Then you must press it again to cancel viewing. Using button as toggle switch you can use single press. You will see briefing while button is depressed. Release button and return to cockpit view.

How to? Choose any button and assign **Toggle** function to it. Check **2 buttons** and assign the same line number to **2-nd virtual button** counter (Fig. 4.18).

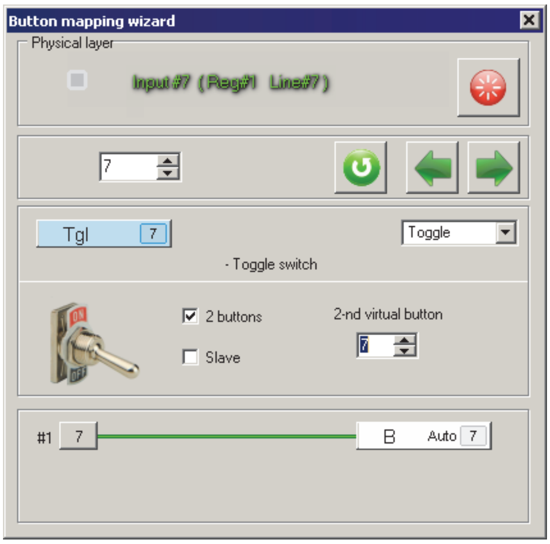


Fig. 4.18.

When you physically press button virtually it will be «pressed and released». When you release button its line will be «pressed and released» once again (Fig 4.19).

Fig. 4.19.

Restrictions

Controller can process up to 64 toggle switches. *Shift1* and *Shift 2* modifiers do not affect toggle switches.

4.3.10. Encoder

Encoder

Description

Encoder converts axis rotation to pulse sequence. Mouse wheel is an example of encoder. Encoder looks like potentiometer. Encoders of VKB joysticks are presented as three lines. Two of them generate signals when you rotate encoder wheel and the third works when you press wheel axially.

Some encoders have no axial button

For some objects or processes (zoom view, trimmers) significant is not absolute but reference position. And vice versa for example you always must know throttle or RPM lever position. Instead of potentiometer encoder has no extreme positions. Encoder occupies two lines for axis and one line for axial button if it exists. Physical encoder can be configured in the following modes:

- discrete — encoder wheel rotation is converted to pulses of two lines referenced to rotation direction.
- analogue — encoder works similarly to physical axis; this virtual axis can be considered as independent axis or be used to trim existing one.

Discrete encoder

Choose **Discrete** item from **Type** combo box. See Button mapping wizard dialog for discrete encoder on Fig. 4.20.

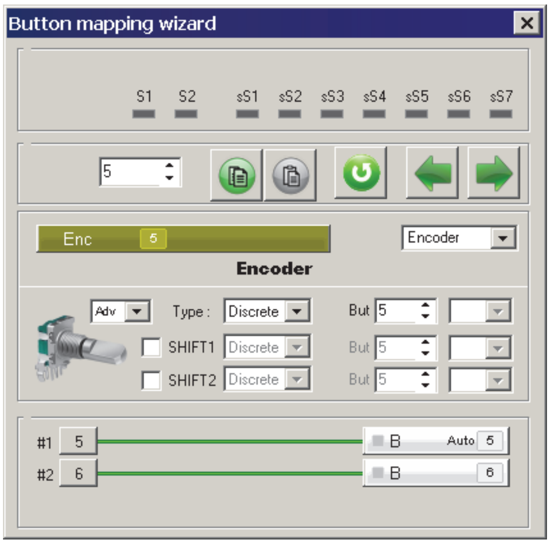


Fig. 4.20.

Encoder wheel is rotated discretely with clicks. Select number of pulses that will be generated for every click:

- 1/4 — four pulses,
- 2/4 — two pulses,
- 4/4 — single pulse.

When you assign **Encoder** function to a line adjacent one will be reserved automatically. Current line generates pulses on rotation to one direction, next for another direction. The first encoder line must be odd. If you try to assign encoder function to even line warning message appears (Fig. 4.21).

Fig. 4.21.

The number of the first encoder physical line on Fig. 4.20 is 23. By default the next line #24 is reserved. You can remap logical (seen in game) line to another number using **But** counter. You can assign logical functions to encoder lines. Encoder lines can be modified (added additional line numbers) with *Shift1* and *Shift2* modifiers. To use shifts check corresponding control items and choice line numbers using **But** counters. Modified (virtual) encoders can be set as discrete or analogue similarly to physical one.

Encoder pulse time is equal to **T_Enc** global parameter value (ms). It is recommended to set this value not less then 15 ms. You can set encoder axial button similarly to simple button (see 4.3.1 on p. 53). It is recommended to assign **Trimmer reset** function to it.

You can use single button to reset several trimmers simultaneously.

Analog trimmer

Select **Trimmer** item from Type combo box to use encoder as analog device. **Button mapping wizard** dialog for this mode is shown on Fig. 4.22.

Fig. 4.22.

Encoder wheel rotates discretely with clicks. Number of output pulses per one click you can select from combo box. This parameter together with **Multiplier**, specifies the shape of axis response curve. See examples of response curve on Fig. 4.24, 4.25, 4.26.

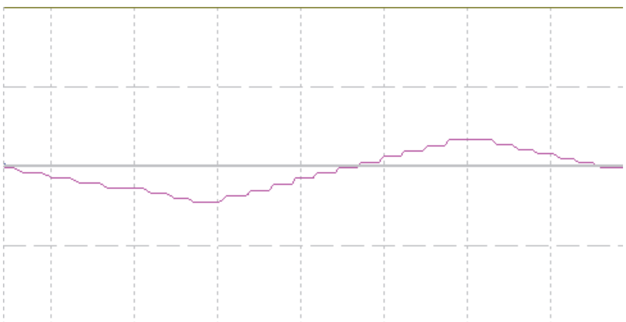


Fig. 4.24. Pulses per click 4/4, Multiplier 32

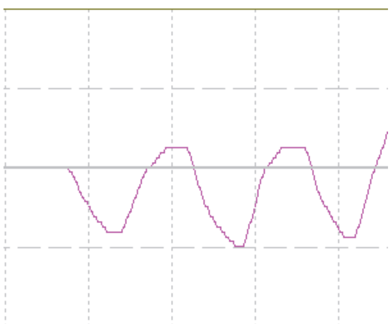


Fig. 4.25. Pulses per click 4/4, Multiplier 256

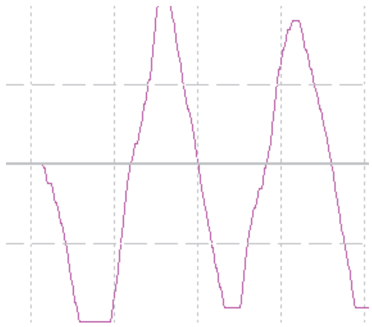


Fig. 4.26. Pulses per click 1/4, Multiplier 32

Assign number of the axis that will be controlled by encoder using **Axis** counter. If existing axis has this number it will be trimmed by encoder. If no axis has this number, new one will be created.

Do not forget to enable this axis and make it visible (check **En** and **Vs** on **Profile — Common-nAxes — Logical axes** tab). Set this axis as **Virtual** on **Physical Axes** tab.

Restrictions

Total of encoders must not be more than 64.

4.3.11. Cyclic switch

Cyclic Switch

Description

Sequential single button pressing virtually presses some adjacent lines. Button mapping wizard dialog for Cyclic switch function is shown on Fig. 4.27.

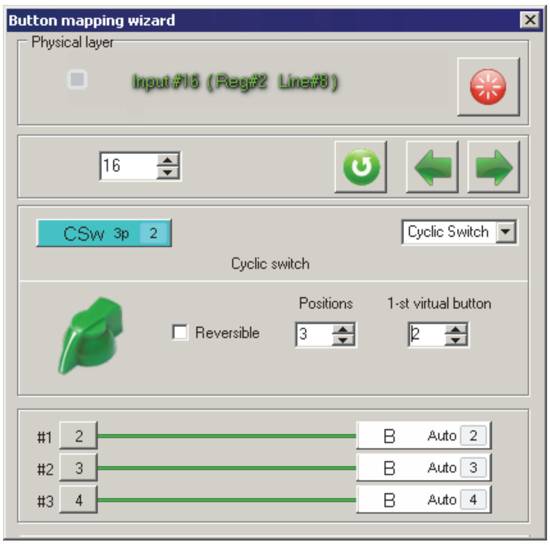


Fig. 4.27.

Positions counters sets virtual line number. **1-st virtual button** counter sets number of the first virtual line that will be pressed with the first pressing of cyclic switch button.

For example shown on Fig. 4.27 first press of button #16 (**Start** button on the right side of Fat Black Mamba case) will switch line # 2. The second – #3, the third – $4. Next press will switch line #2 again and so on. If **Reversible** is checked then lines will be switched in the following sequence: 2 3 4 3 2.

Virtual lines of Cyclic switch can use logical functions.

4.3.12. POV Switch

Discrete POV Switch

Description

Physically discrete HAT is four-position switch. It consists of four pushbutton switches with shaft. HAT uses four lines. Virtually HAT can be represented as eight-position switch. Intermediate positions are generated by software.

HAT position names are:

- HR — to right,
- HD — down,
- HL — to left,
- HU — up.

Check **ALPS** to enable HAT pushbutton.

Button mapping wizard dialog for HAT is shown on Fig. 4.28.

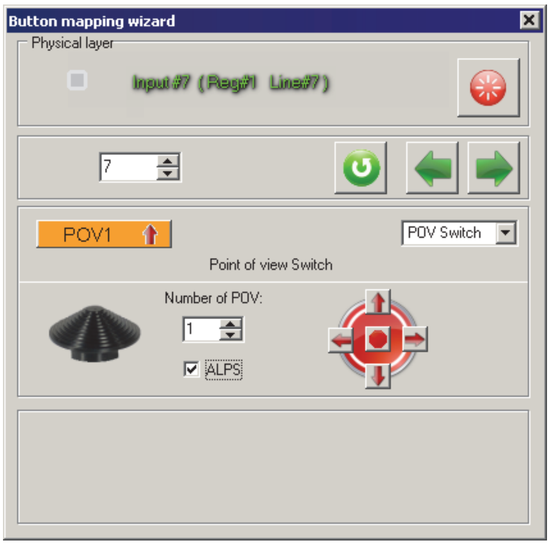


Fig. 4.28.

Set current HAT index using **Number of POV** counter. Set HAT position for current line pressing corresponding arrow pr central button on HAT image.

Usually HAT is used for viewing (if you do not use NaturalPoint TrackIR or other similar device). If you do not need HAT as view controller you can configure it as four simple independent buttons. In this case you must configure lines of HAT as buttons (see 4.3.1 on p. 53). Thus you can add up to twelve buttons (using *Shift1* and *Shift2* modifiers).

Restrictions

Total of HATs must not be more than 4. Global parameter **#Hat** (see 2.4.1 on p. 20) specifies maximum HAT number for current configuration.

4.3.13. uStick Switch

Ministick mode switch

Description

Switches analog ministick between HAT and two axes modes. Button master wizard is shown on fig. 4.29.

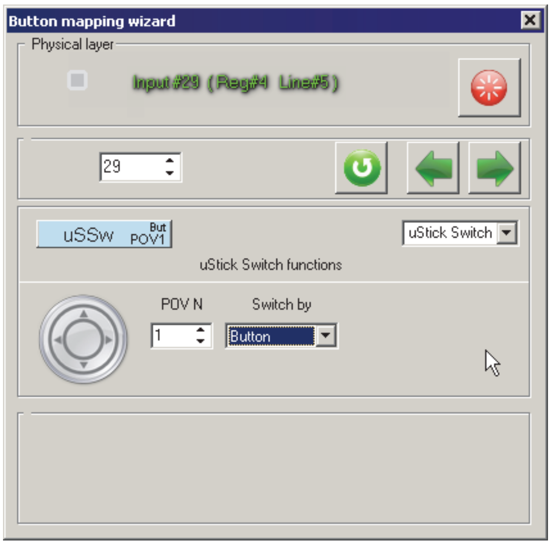


Fig. 4.29.

Set ministick number using **POV N** counter. Choose switch mode from combo box **Switch by** (table. 4.2).

Table. 4.2.

Mode	Description
Button	Simple button switch.
ButtonAlt	Alternative button switch.
Tempo	Two stage switch. Ministick state depends on press duration. Set default ministick state using **POV mode on start** checkbox.

Table. 4.2.

Mode	Description
TempoB	Two stage switch. Ministick state depends on press duration. Set default ministick state using **POV mode on start** checkbox. Instead of Tempo mode short press is registered as button press additionally. If ministick axes configured as relative, **Short Press Axis Reset** checkbox (fig. 4.30 on p. 70) controls axis reset on short press. **Example**. Ministick is in axes mode. **Short Press Axis Reset** is checked. On short press ministick keeps axes mode. Axes jump to zero. **Short Press Axis Reset** is unchecked. On short press ministick keeps axes mode. Axes do not move. Btton press is registered.
SHIFT	Shift modifier switches ministick state. Choose shift number using **SHIFT N** counter.
SubShift	SubShift modifier switches ministick state. Choose subshift number using **SubSHIFT N** counter.

Fig. 4.30.

To use uStick switch choose **Always** activity mode (**Active** combo box on **POV** tab).

4.3.14. uPOV Switch

Active POV switch

Description

Switches ativity between POV switches. Button master wizard is shown on fig. 4.31.

Fig. 4.31.

Choose first active POV using **POV N** counter. Choose switch mode from combo box **Switch by** (table. 4.3).

Table. 4.3.

Обозначение	Описание
Button	POV switches by button depressing.
Cyclic	POV swiches consequantly by button short presses.

Example.

Gunfighter SCG joystick. Two POVs enabled. They use same axes. **On/Off** POV activity mode is chosen (fig. 4.32.).

Fig. 4.32.

On joystick start POV №1 is active. uPOV button press activates POV №2. If **Mouse** output mode is selected for this POV, virtual mouse will work.

On uPOV use uStick MUST be disabled. No one button with this function.

4.3.15. SwitchCB

Complementary button

Description

Several buttons included into a group. Complementary button is pressed if all other buttons of the group are released. Button master wizard is shown on fig. 4.33.

Fig. 4.33.

Set lines of group number using **N** counter. Select complementary line number using **Compl.BUT** counter. It is the first line of the group. Line №20 will be pressed if lines 21, 22, 23 will be released.

Switch CB is usable for 3-position rotary switch. If **N** = 1, button with additional inversed output is created.

4.3.16. Generator

Pulse generator

Description

When you press button with this function pulse sequence will be generated. Global parameter **T_Gen** (see 2.3.1 on p. 19) specifies pulse frequency. The following generator types can be used:

- G1,
- G8,
- GT,
- GT+,
- GTE,
- GTE+,
- GTR,
- GTR2,

Button mapping wizard dialog for generators is shown on Fig. 4.34.

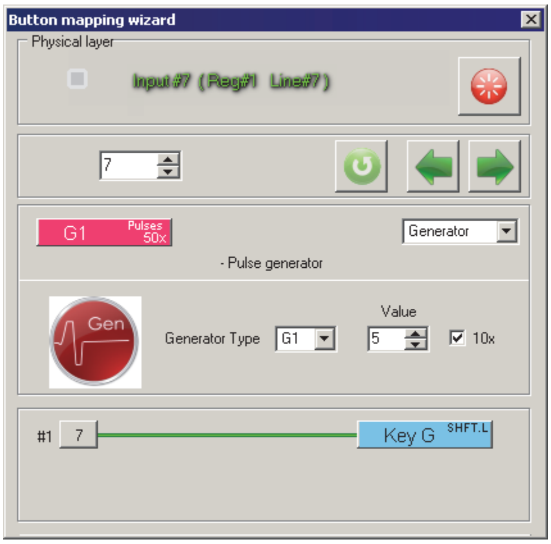


Fig. 4.34.

Choose generator type from **Generator Type** combo box. You can use logical functions for generator pulses. If generator line is used as joystick button than single pulse with specified parameters will be generated. If his line is mapped to keyboard than series of keystrokes will be generated for specified pulse length (or pulse number). Keystroke frequency corresponds wits operation system settings.

G1

Generates specified number of pulses with single button press. Set this number using **Value** counter. Checking **x10** you can multiply specified number to 10. Example. **Value** is equal to 7, **x10** checked. 70 pulses will be generated.

Flaps toggle switch (Fat Black Mamba) settings are shown on Fig. 4.34. It is used for manual gear retract. Keystroke *<Shift>+<G>* is mapped to lower toggle switch position. When you lower handle this keystroke will be generated 50 times. It guarantied retracting gear of Polikarpov I-16 in Il-2 Sturmovik.

G8

Generates pulse batches with specified frequency all the time button is pressed. Set number of pulses in a single batch using **Value** combo box.

H4U button of CMS HAT (HOTAS Warthog grip with Fat Black Mamba case) settings are shown on Fig. 4.35.

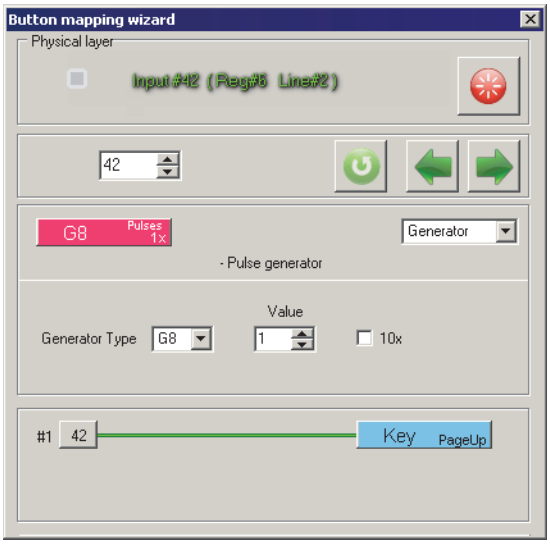


Fig. 4.35.

This button is used to zoom in. <*Page Up*> keystroke is mapped to this button. H4D button of this HAT is specified as G8 too and mapped to keystroke <*Page Down*>. Buttons control zoom in and out.

GT

Generates single pulse of specified length on button press. No matter to button state - will it stay depressed or released.

Set pulse length in **Value** field, 0,01 s, from 1 to 127. Checking **x10** you can multiply specified length to 10. Example. **Value** is equal to 20, **x10** checked. 2 s pulse will be generated. Thus GT pulse length range is 10 ms to 12,7 s.

GT+

Generates single pulse of specified length on single button press. If the button will stay depressed more than pulse length, it will be generated until you release button.

Set pulse length in **Value** field, 0,01 s, from 1 to 127. Checking **x10** you can multiply specified length to 10. Example. **Value** is equal to 20, **x10** checked. 2 s pulse will be generated. Thus GT pulse length range is 10 ms to 12,7 s.

If you press **GT+** several times pulse periods will be summarized.

GTE

Generates single pulse of specified length on button press. No matter to button state - will it stay depressed or released.

Set pulse length in **Value** field, 0,01 s, from 1 to 127. Checking **x10** you can multiply specified length to 10. Example. **Value** is equal to 20, **x10** checked. 2 s pulse will be generated. Thus GT pulse length range is 10 ms to 12,7 s.

GTE+

Generates single pulse of specified length on single button press. If the button will stay depressed more than pulse length, it will be generated until you release button.

Set pulse length in **Value** field, 0,01 s, from 1 to 127. Checking **x10** you can multiply specified length to 10. Example. **Value** is equal to 20, **x10** checked. 2 s pulse will be generated. Thus GT pulse length range is 10 ms to 12,7 s.

If you press **GT+** several times pulse periods will be summarized.

Differences between GT and GTE
GT generators are independent. One generated pulse can not be interrupted. If your generator assigned to extend gear it will work the whole time. GTE pulse can be interrupted by another one! For example keystroke F assigned to extend flaps of LaGG-3. Button has GTE function with time equal to 10 seconds. You have pressed button and flaps began to go down. Keystroke V assigned to retract flaps and its button has GTE function too.

You need to retract flaps before full extending. When you press GTE with V keystroke it will interrupt the first generator and flaps will be retracted.

GTR
Generates single pulse of specified length on button press. The second press interrupts pulse.

Set pulse length in **Value** field, 0,01 s, from 1 to 63. **Multiplier** combo box allows to select multiplication factor. **x10** will multiply specified length to 10. Example. **Value** is equal to 2, **100x** selected. 2 s pulse will be generated.

GTR2
Generates pair of short pulses divided by specified period on button press. The second press generates the second pulse of pair before period ends. Short pulse length is specified by **T_Gen** parameter (see 2.3.1 on p. 19).

Set period between pulses length in **Value** field, 0,01 s, from 1 to 63. **Multiplier** combo box allows to select multiplication factor. **x10** will multiply specified length to 10. Example. **Value** is equal to 2, **100x** selected. Period length will be equal to 2 s.

4.3.17. Tempo

Two stage button

Description
Button output depends on button depressing time. The short press will cause signal of one line but more long one – another one. Such function is used in real modern planes. **Tempo Time** global parameter (see 2.3.1 on p. 19) specifies depressing period. Select function type from **TEMPO Type** combo box.

Tempo1 and Tempo2
2-position switches.

Button mapping wizard dialog for Tempo1 and Tempo2 functions is shown on Fig. 4.36.

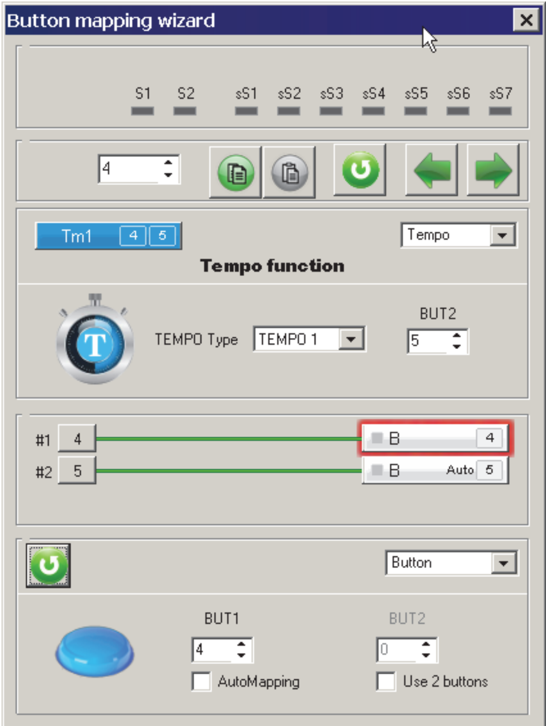


Fig. 4.36.

Use **BUT2** counter to choose the second line number (see Free line choice on p. 51). Both lines of TEMPO can use logical functions. If button depressing time is less than **Tempo Time** value, than first line pulse time will be equal to **T_Tgl** value (see 2.3.1 on p. 19). If depressing time exceeds **Tempo Time** value the result depends on **TEMPO Type** parameter.

- **Tempo 1** – second line pulse length is equal to **T_Tgl** value in no matter to real depressing time.
- **Tempo 2** – second line pulse length is equal to button depressing time.

Tempo 3

3-position switch.

Button mapping wizard is shown onfig. 4.37.

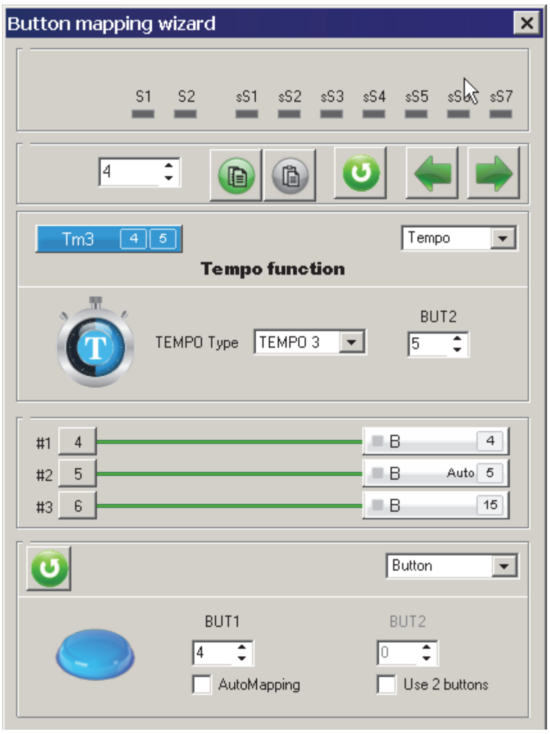


Рис. 4.37.

Double click switch mode is added for Tempo 3. For configuration shown on fig. 4.37 short press switches to line 4, long – 5. Double click enables line 15.

Tempo 3s

3-position Shift modifier.

Button master wizard is shown on fig. 4.38.

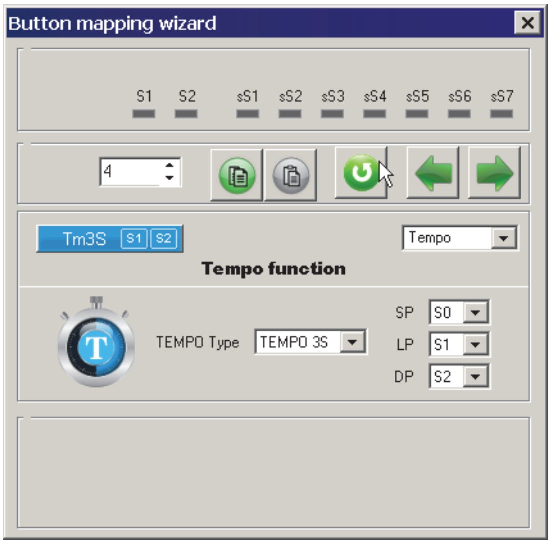


Рис. 4.38.

Associate shift modifiers with every Tempo mode (Short / Long / Double). Do not use other shiht modifiers if you use TEMPO 3s function.

Tempo 3A

Static 3-position switch.

Button mapping wizard is shown on fig. 4.39.

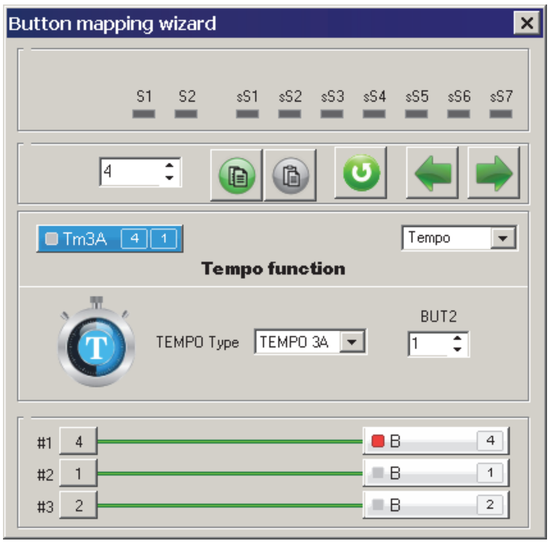


Рис. 4.39.

One of lines is «pressed» permanently. Lines are selected in the same manner as for Tempo 3.

For configuration shown on fig. 4.39 line 4 is pressed by default. Short press switches to the same line 4, long – 1. Double click enables line 2.

4.3.18. Trimmer

Trimmer

Description

Controls axes (physical or virtual) with buttons. Function modes:

- **Trimmer Reset** — Resets trimmer(s) to default condition. Returns axis with center to the central position and axis without center to zero. Also stores current axes trim value. It can be reapplied to axes using Trimmer Return function.
- **Trimmer Return** — Reapplies trim value, stored with Trimmer Reset function to selected axes.
- **Trimmer+**, **Trimmer-** — Trimming of existing axis or creation of virtual one. Axis response is changing until button is pressed. When you release button axis stops. + or - defines trimming direction.
- **Trimmer Auto+**, **Trimmer Auto-** — Alternative trimming. When you release button axis response falls to center or zero regarding to axis type. + or - defines trimming direction.
- **Trimmer SET+**, **Trimmer SET-** — Sets axis response to specified value.

Choose desired mode from **Function** combo box.

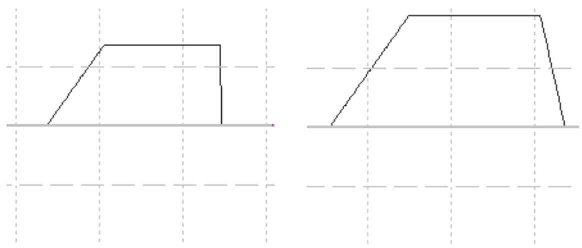
Trimmer Reset, Trimmer Return

Button mapping wizard dialog for Trimmer Reset function is shown on Fig. 4.40.

Fig. 4.40.

Check axis numbers that will be reset in **Applied axes** group. Axes 1, 2 and 8 will be reset in this example. Virtual axis # 8 (encoder as trimmer, see 3.5 on p. 36) is combined with axis #2. Single button will reset both axes simultaneously.

Global parameter Trimmer Time defines duration of trimmer reset process. If **Trimmer Time**=0, then trimmer reset will be instant. (Fig. 4.41, a). Fig. 4.41, b) shows trimmer reset if **Trimmer Time**=300.



a) b)

Fig. 4.41.

Button mapping wizard dialog for Trimmer Return function is similar to one for Trimmer Reset and is shown on Fig. 4.42.

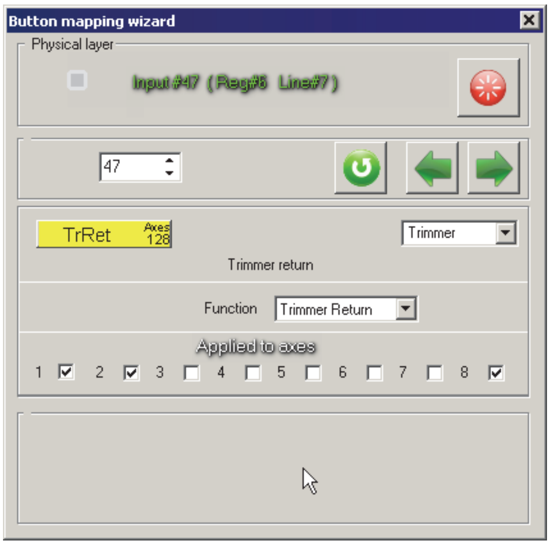


Fig. 4.42.

Check axis numbers that will be retrimmed with values, stored while resetting with Trimmer Reset function in **Applied axes** group. Trimming of axes 1, 2 and 8 will be restored in this example.

Trimmer+, Trimmer–, Trimmer Auto+, Trimmer Auto–
Button mapping wizard dialog for these trimmers is shown on Fig. 4.43.

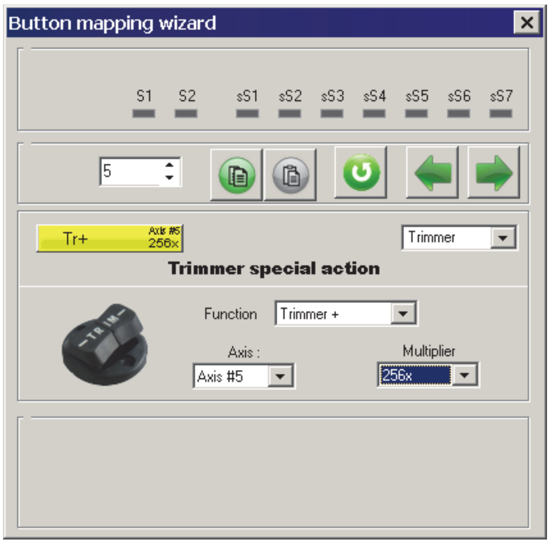


Fig. 4.43.

Signs of **Trimmer+**, **Trimmer–**, **Trimmer Auto+**, **Trimmer Auto–** specify trimming direction. Axis response for **Trimmer+** and **Trimmer–** will be fixed after button is released. Axis response for **Trimmer Auto+** and **Trimmer Auto–** will return to the center (axis with center) or zero (without center).

Choose trimmed axis number from **Axis** combo box. If there exists an axis with specified number it will be trimmed. If an axis does not exist it will be created. **Multiplier** combo box value specified response speed. This parameter is analogous to **Multiplier** for encoder-trimmer (see Analog trimmer on p. 66).

Trimmer SET+, Trimmer SET–

Button mapping wizard dialog for these trimmers is shown on Fig. 4.44. Signs of **Trimmer SET+** and **Trimmer SET–** specify trimming direction. After button press axis response will be equal to specified value.

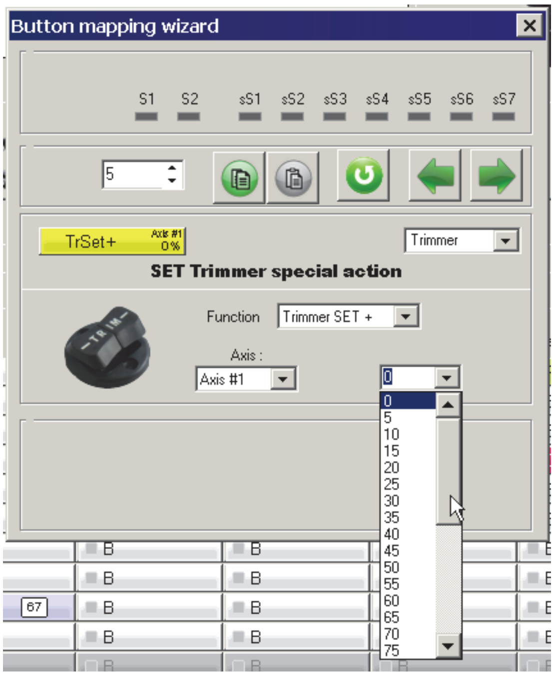


Fig. 4.44.

Choose trimmed axis number from **Axis** combo box. Choose trimmed axis number from **Axis** combo box. If there exists an axis with specified number it will be trimmed. If an axis does not exist it will be created. Items of expanded combo box allow to specify response value.

Global parameters

In order to use listed functions choose **Trimmer+** or **Trimmer–** for desired axes from **Trimmer** combo box on **Profile — Common-n-Axes — Physical Axes** tab.

If function creates new axis check **En** and **Vs** for it on **Profile — Common-nAxes — Logical axes** tab. Set this axis as **Virtual** on **Physical Axes** tab.

4.3.19. Curves

Dynamic equalizer

Description

Allows temporary apply custom response curve (see 3.6 on p. 40) to chosen axes. This function is similar to **DR** (see DR on p. 83). Button mapping wizard dialog for Curves is shown on Fig. 4.45.

Fig. 4.45.

Check customized axis numbers in **Applied axes** group. UNCHECK **Eq** for these axes on **Profile — Common-n-Axes — Physical Axes** tab.

If **Eq** is checked custom curve is applied to an axis permanently.

If **alternate action** is checked function works as a trigger. The first press enables custom mode and the second one disables it.

4.3.20. Axes fixation

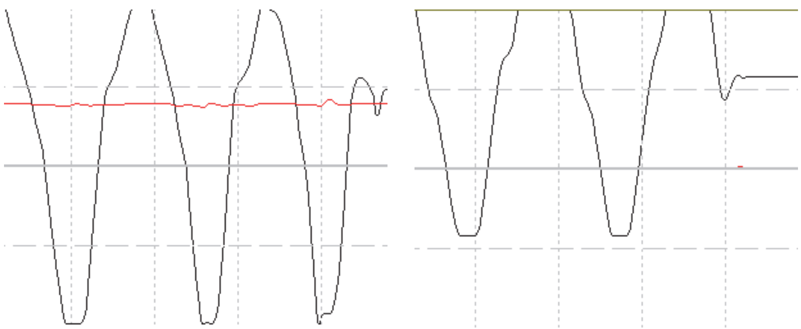
Fix Axes

Description

Axis modifiers allow to set specified axis response value.

FA0

Allows to set current axis position as its center («helicopter trimmer»). Fig. 4.46 shows response of axes X and Y with **FA0** function enabled depending on **Trimmer mode** parameter (see on p. 32).



Trimmer mode = Modified   Trimmer mode = Standard

Fig. 4.46.

FA1

Fixes axis without center (throttle, for example) in zero position and axis with center (pitch, yaw etc.) in central position. Axis is fixed until button with FA1 function will be released.

FA2

Fixes axis in the current position. Axis is fixed until button with FA2 function will be released.

FA3

Sets axis response to specified value. **FA3 val** variable specifies response value in percents of the whole response range. This parameter for each axis you can find on **Profile — Common-n-Axes** tab. Fig. 4.47 shows X and Y axes response when FA3 button is pressed four times. **FA3 val** is equal to 45.

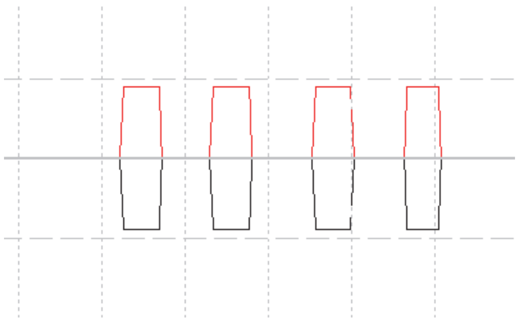


Fig. 4.47.

Indeed joystick grip stayed in the center.

DR

Reduces axis response in specified rate. **D.Rate** (**Global — Common** tab) parameter specifies rate value in percents of the whole range. This function can be useful when you aim the target. It is similar to *Curves* function (see 4.3.19 on p. 81). Fig. 4.48 shows *DR* button work. This button was depressed for some time and then released. Joystick grip was moved by the same manner within the whole range.

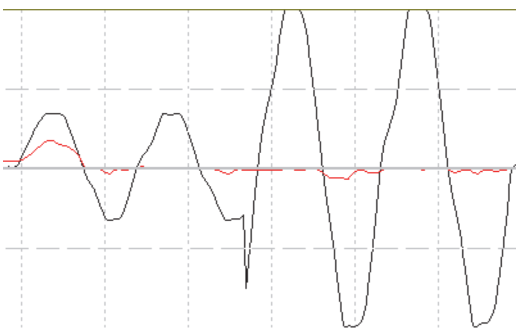


Fig. 4.48.

FA2, FA3, DR modifiers have **alternate action**. If it is checked modifier works as trigger that is the first press enables axis fixation and the second disables it.

Button mapping wizard dialog for axis fixation functions is shown on Fig. 4.49.

Fig. 4.49.

Choose fixation mode from **FA Type** combo box. Check axis numbers that will be affected in **Applied to axes** group.

Modification parameters:

- **FA** on **Global** — **Common** tab —enables modification.
- **FA0**, **FA1**, **FA2**, **FA3** variables (ms) — specify transition time for corresponding function (modifications are applied to axis response not immediately).

4.3.21. AUX Axes

Auxiliary axis

Description

Functions of this group allow «swap» axes. **Button mapping wizard** dialog for AUX axes is shown on Fig. 4.50.

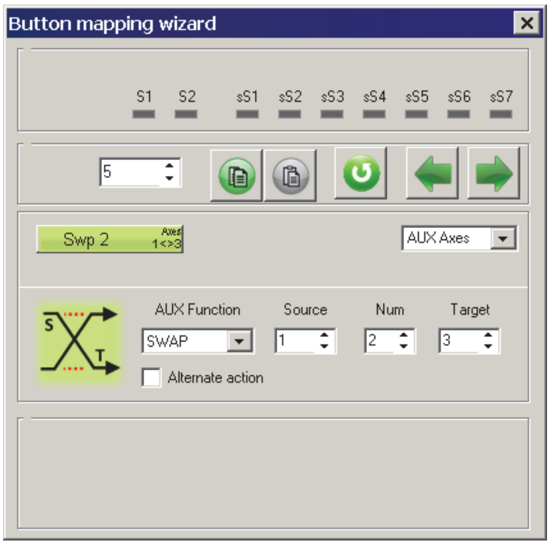
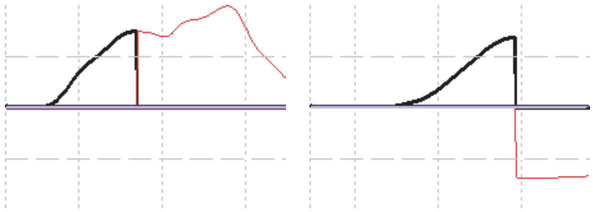


Fig. 4.50.

Choose function from **AUX Function** combo box.

SWAP

Allows to rotate physical axis (source) but get response of another one (target) if **SWAP** button is pressed. Source axis response in this case will be equal to zero for axis without center or goes to the center for axis with it. Target axes response will be equal to source one on the moment of button pressing (Fig. 4.51, a). If target axis is inverted, than response value will be inverted too (Fig. 4.51, б).



a) б)

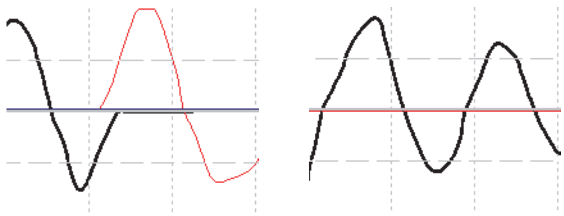
Fig. 4.51.

Select source axis number from **Source** combo box and target one from **Target** combo box. For example shown on Fig 4.50 on p. 84 when you will rotate X (#1) axis with pressed SWAP button Z (#3) axis response will vary. SWAP button can affect several axes. **Num** field value specifies modified axes number. If **Num**=2, than for this example when you press SWAP button and rotate X (#1) axis than Z (#3) axis will be «rotated». Y (#2) axis rotation will «rotate» Rx (#4) axis.

REMAP

Allows rotate several axes linked with source axis. An example is shown on Fig. 4.52.

X (#1) axis is source, Z (#3) is target. Fig. 4.52, a) – REMAP button released. When you rotate each of these axes they work. X (#1) axis response is drawn by bold line. Fig. 4.52, b) – REMAP button is pressed. When you rotate X (#1) axis it does not response. When you rotate Z (#3) axis than X axis «rotates» too.



a) b)

Fig. 4.52.

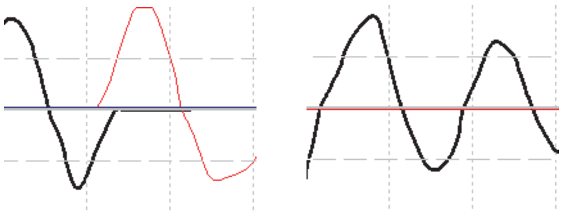
If target axis is inverted than its response will be inverted too.

Set number of linked axes in **Num** field. For example if **Num**=2, then rotation X (#3) axis will cause rotation of #3 and #1, rotation axis #4 — #4 and #2.

SWITCH

Modifier is opposite to REMAP. If button with SWITCH function is pressed target axis is disabled. Response of this axis will appear if source axis will be rotated. All other function parameters are the same to REMAP function ones.

Axis #1 is source and axis #3 is target (Fig. 4.53). Fig. 4.53, a) button with SWITCH function is not pressed. Axes #1 (bold line) and #3 response separately. SWITCH button is pressed (Fig. 4.53, б). When you rotate axis #3 it does not response. When you rotate axis #1 you can see axes #1 and #3 responses.



a) б)

Fig. 4.53.

SWITCH 0

Similarly to SWITCH. If button is pressed target axis is zeroed and disabled. Response of this axis will appear if source axis will be rotated.

PAI INV

Inverts axes. St number of first inverted axis using **Source** field. Set inverted axes count using **Num** firld. **Source** = 1, **Num** = 2. If button is pressed, axes №1 and №2 will be inverted. Fig. 4.54 shows axis response on different button state with the same axis direction.

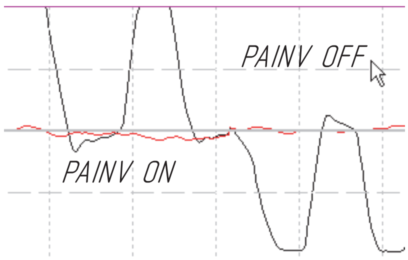


Рис. 4.54.

SPLIT Rev

Splits axis to two semiaxes symmetrically to center. Set splitted axis using **Source** field. Function result is shown on fig. 4.55.

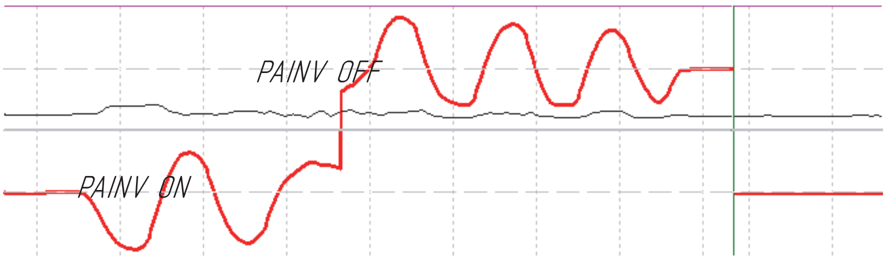


Рис. 4.55.

4.3.22. RelAxes

Relative axes control

Description

This function allows to set relative axis (see 3.4 on p. 35) response value. **Button mapping wizard** dialog for RelAxes function is shown on Fig. 4.56.

Fig. 4.56.

Select function mode from **RelAxes Function** combo box.

- **Set Value** allows to set response axis value equal to **FA3 val** (see 3.3 on p. 34) parameter.
- **RESET** allows to reset response axis value.

Check applied axis numbers in **Applied to axes** group.

4.3.23. Boolean

Boolean functions

Description

Allow to control virtual button press using other buttons state. For example Eject button will work if both specified buttons will be pressed simultaneously only. Boolean functions can be connected to chains thus generating logical sequences.

Button mapping wizard dialog for Boolean function is shown on Fig. 4.57.

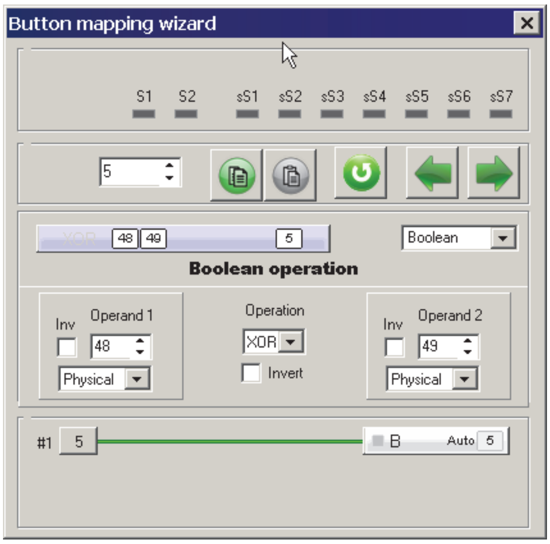


Fig. 4.57.

It is reasonable to assign target virtual button line number in range of lines not connected with physical control units, for this example #75.

Specify numbers of buttons that will control virtual one using **Operand 1** and **Operand 2** counters. **Inv** checkbox allows to invert input signal. if checked line will affect if is NOT pressed only. Items of combo box allow to select input signal type: physical button press, logical or virtual – the result of logical functions, Shift modifiers and so on.

Select Boolean function that will be applied to this line (virtual button), table 4.4, from **Operation** combo box.

Table. 4.4.

Function	Description
AND	Button «pressed» if both source buttons **Operand 1** and **Operand 2** are pressed.
OR	Button «pressed» if any of source buttons **Operand 1** or **Operand 2** is pressed.
XOR	Button «pressed» if one of source buttons **Operand 1** or **Operand 2** IS pressed AND another IS NOT.
RS	RS-trigger. Button «pressed» if **Operand 2** is pressed, **Set** input. Button «released» if **Operand 1** is pressed, **Reset** input.

Table. 4.4.

Function	Description
RT	RT-trigger. Button changes state (toggles) with every **Operand 2** pressing. But this happens ONLY if **Operand 1** is NOT pressed. If **Operand 1** is pressed, button state will be unchanged regardless of **Operand 1** pressings. If RT-trigger button was in «pressed» state when **Operand 1** was pressed, then its state will be changed to «released».
CMP	Comparator. Forces line actuation if axis response is greater or equal to specified value.
JMP	Jumper. Maps logical button line to physical one.

CMP

Line will be actuated when axis response value (**Operand 1**) becomes equal or greater to specified value (**Operand 2**), in percents of full response range. It is recommended to use **Virtual** or **Logical** axis type. For **Physical** or **Virtual** types, which are bipolar, **Inv** checkbox allows to select response direction. For **Logical** type this parameter is ignored. Fig. 4.58 shows an example of function parameters.

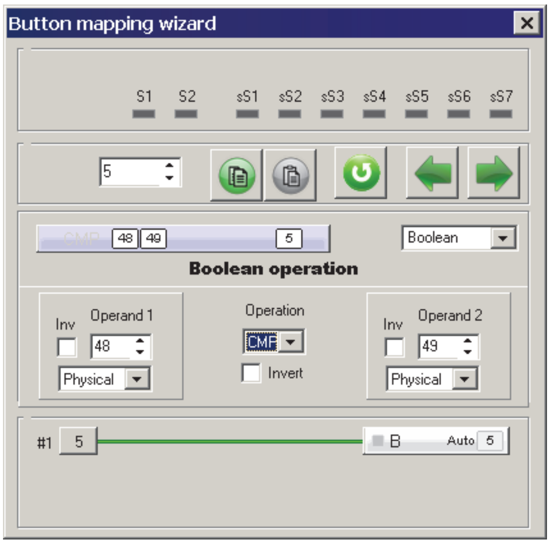


Fig. 4.58.

Fig. 4.59 shows an example of function work.

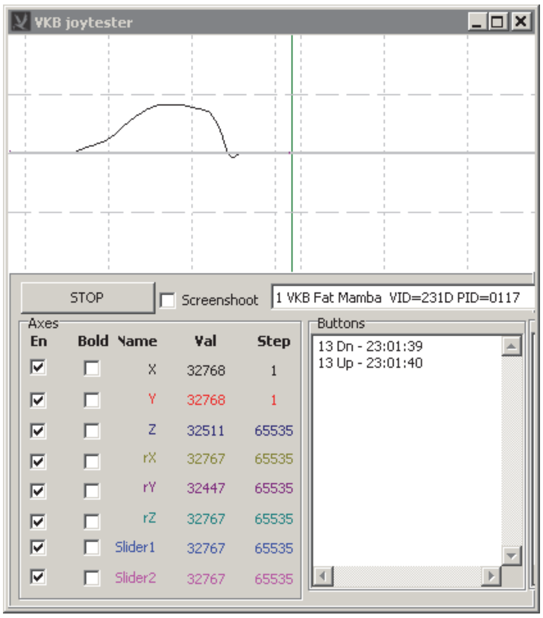


Fig. 4.59.

Fig. 4.60 shows an example of function work if **inv** for **Operand 2** is checked.

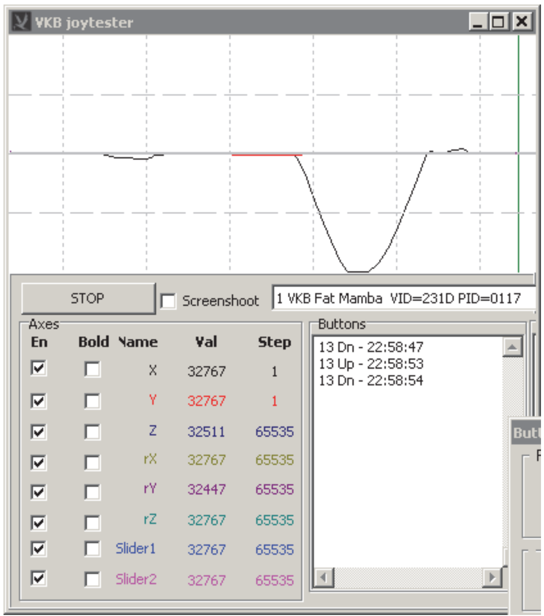


Fig. 4.60.

JMP

By default logical layer button lines can use restricted function set: button, keystroke mapping, HAT, macro launch etc. (Fig. 4.61).

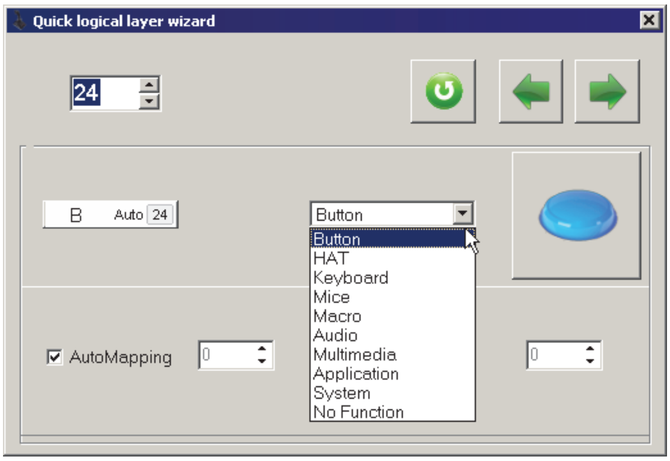


Fig. 4.61.

Generators, trimmers, encoders and a lot of other useful functions are enabled for physical layer lines only. Thus you can not assign such function to line that will act when you press button with Shift modifier.

Jumper function allows to ignore these restrictions. It maps logical line to physical layer.

Button mapping wizard dialog for Jumper function is shown on Fig. 4.62.

Fig. 4.62.

Set logical layer line number as **Operand 1**. Set mapped physical line number as **Operand 2**.

Example.

If you press Flaps toggle switch on back side of Fat Black Mamba case (line 13) together with Shift modifier, logical line 54 will be actuated (Fig. 4.63).

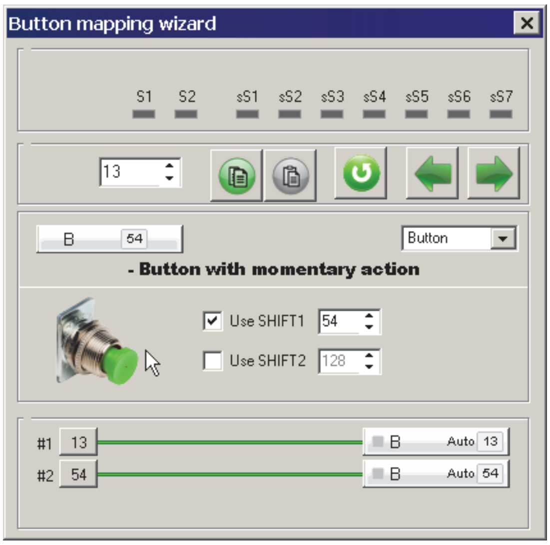


Fig. 4.63.

Jumper function with parameters shown on Fig. 4.62, maps logical line 54 to physical 52. Trimmer Auto+ function is assigned to this **physical** line (Fig. 4.64).

Fig. 4.64.

Thus Jumper allows toggle switch modified with Shift to trim an axis!

4.3.24. DZ Switch

Deadzone dynamical disable

Description

Button press disables deadzone of selected axes. Button mapping wizard is shown on fig. 4.65. Check axes numbers to disable deadzone.

Fig. 4.65.

4.3.25. RPB

Button replica

Description

Enables virtual button press simultaneously with physical one. **Button mapping wizard** dialog for RPB is shown on Fig. 4.66.

Fig. 4.66.

Choose physical button number that will be source for virtual using **Replicated button** counter. For this example when you press Start button (left side of Fat Black Mamba case, line #16) line 32 will generate a pulse too. Line-replica can use logical functions (see Logical (output) function choice on p. 52).

4.3.26. Sync

Toggles synchronizer

Description

When Sync button is pressed all lines of switched on toggles will generate pulses. For two state Toggle switch the pulse will be generated by switched on position (see 4.3.9 on p. 61).

Example. Fat King Cobra joystick has four toggles. Three of them are *On* and one 2 *Off*. When game started it «does not knows» about toggle states. Controls are not scanned automatically. After you press Sync button system will get current toggle states.

See also about three-way toggle switch features in Three-position (On-Off-On) Toggle switch features on p. 61.

Button mapping wizard dialog for Sync function is shown on Fig. 4.68.

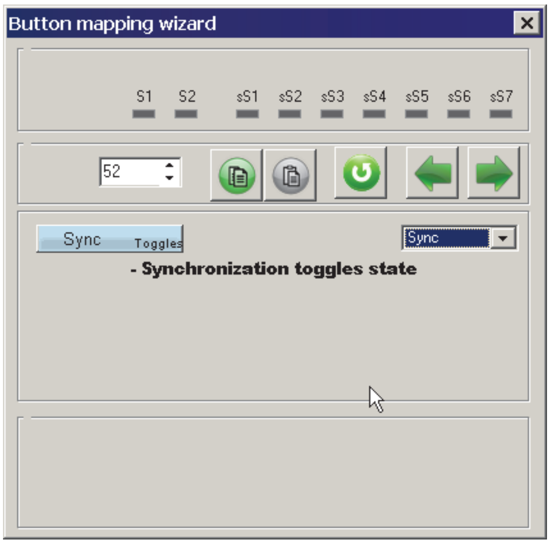


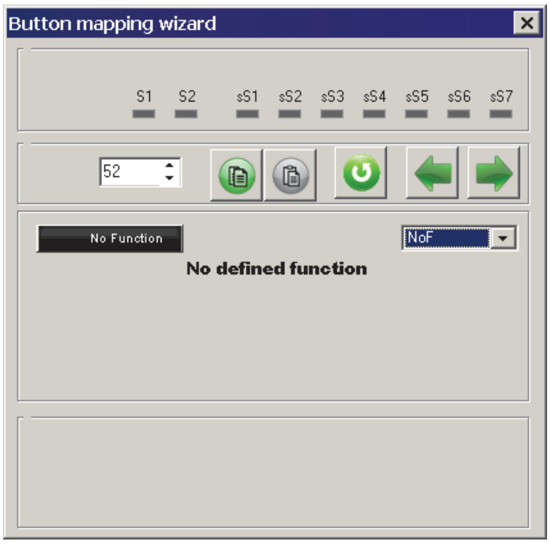
Fig. 4.68.

4.3.27. NoF

No function

Description

Disables any function for control. **Button mapping wizard** dialog for NoF function is shown on Fig. 4.69.


Button mapping wizard
S1
S2
sS1
sS2
sS3
sS4
sS5
sS6
sS7
52
No Function
NoF
No defined function
Fig. 4.69.

Chapter 5.Logical button functions

5.1. Overview

Each of joystick control is connected to controller by single (button, single-state toggle switch) or several (encoder, HAT, Two-state toggle switch) lines. Physically line corresponds to pair of conductors with a specific number. When you press button its line generates a signal. When you rotate encoder wheel in one direction you get a series of its first line pulses. Another direction generates pulses of another line with number greater (or less)) by one. Physical line signal is received by controller. But operation system and the game will receive just logical signal. By default if no logical function is applied, line numbers will be the same. Logical buttons engine executes the following tasks:

- remap control lines,
- control multimedia, use operation system functions, launch applications, work with virtual mouse and so on with use joystick controls,
- map joystick buttons to keystrokes.

5.2. Logical function setup modes

You can obtain access to logical function parameters from **Button mapping wizard** dialog on **Physical layer** tab or directly on **Logical layer** tab.

5.2.1. Access from Physical layer tab

When you set up physical function it can use logical function too. For example simple button or toggle switch can be mapped to keystroke or mouse button or even axis or wheel.

Some physical functions have no access to logical functions, Shift for example.

If physical function has access to logical one corresponding control items are accessible in **Button mapping wizard** dialog. For example **Button mapping wizard** dialog for simple button (*Button*) is shown on Fig. 5.4. It appears after you click button line number cell on **Profile — Buttons — Physical layer** tab.

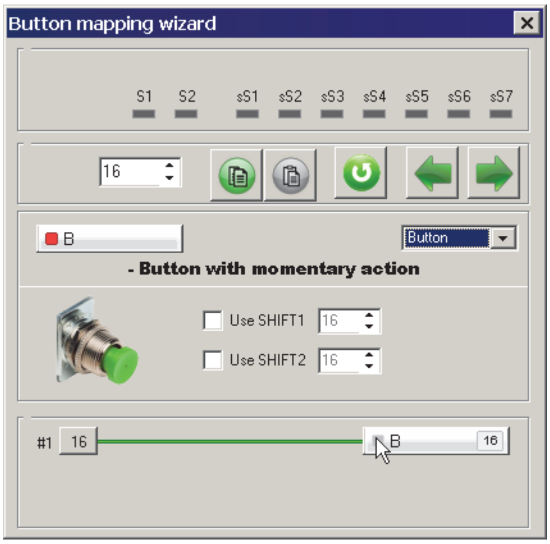


Fig. 5.1.

Line #16 is customized. For Fat Black Mamba joystick it corresponds to **Start** button on the right side of joystick case. By default logical line number will be the same with physical one i.e. 16. In order to apply logical function click logical function sign as shown on a picture. Dialog view will change. Control items for logical function parameters will appear (Fig. 5.2).

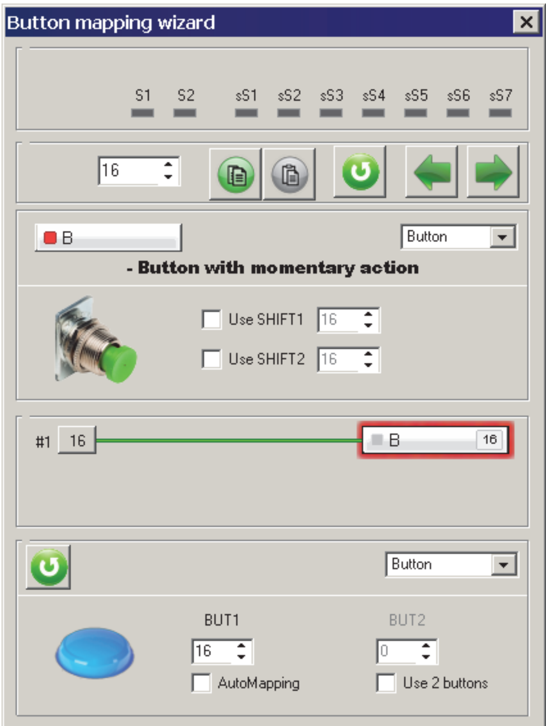


Fig. 5.2.

5.2.2. Access from Logical layer tab

In order to access logical function settings you can open **Profile** — **Buttons** — **Logical layer** tab and click the cell with number of desired line. **Quick logical layer wizard** dialog will appear (Fig. 5.3).

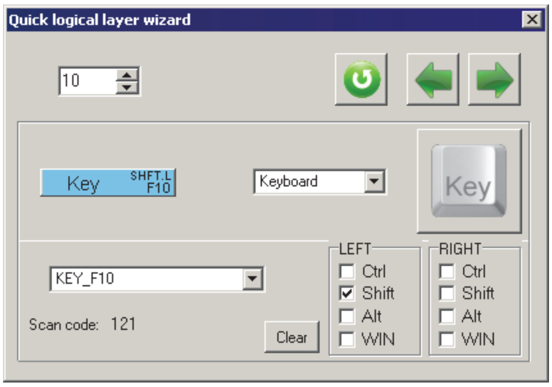


Fig. 5.3.

5.2.3. Function selection

Work with both dialogs is similar. You can select function from combo box (Fig. 5.4).

Fig. 5.4.

See brief function descriptions in table 5.1.

Table. 5.1.

Function name	Description
Button	Simple button.
HAT	HAT switch.
Keyboard	Virtual keyboard.
Mice	Virtual mouse.
Macro	Using macro.
Audio	Audio application control.
Multimedia	Multimedia application control.
Application	Launch applications.
System	System functions execution.
No Function	Logical functions disable.

Be careful to use **System** function. Improper use can cause operation system malfunction.

5.3. Virtual buttons

5.3.1. Overview

Virtual buttons engine allows to remap button numbers. By default button logical number (recognized by OS) is the same with physical one (specified by wiring). **Quick logical layer wizard** dialog is shown on Fig. 5.5.

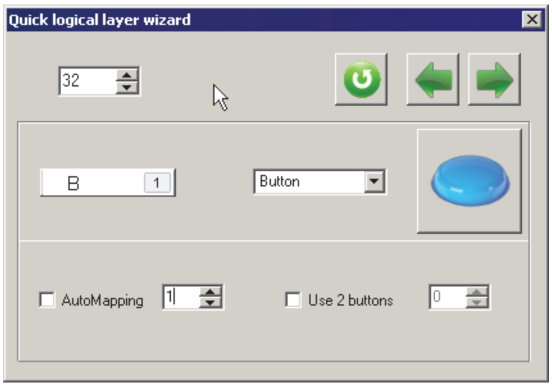


Fig. 5.5.

In order to remap default logical button number uncheck **AutoMapping** and assign number using counter.

5.3.2. Free line choice

When you remap line number you must know that new line number is not occupied. Click line number counter. **Logical layer** dialog appears (Fig. 5.7). Occupied lines are red but free – black. Click line cell to choose its number.

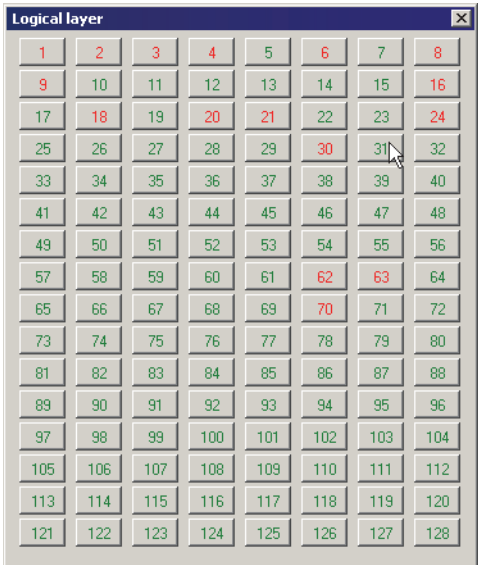


Fig. 5.7.

Using remap engine you can make button mapping more useful. For example the first trigger of Warthog grip on Fat Black Mamba case has physical (and by default logical) number 32. Usually this trigger works very often – to control weapon. It is reasonable to remap its number to #1. This example is shown on Fig. 5.5. You can remap the second trigger line number from 25 to 2 similarly.

If you use encoder as analogue trimmer its two lines are not used. For Mamba family the first encoder physically occupies lines #5 and #6. You can fearlessly use these lines for own purposes. If you want to use default mapping check **AutoMapping**. If you map several physical buttons to the same logical line it will be activated by any of them.

5.3.3. Simultaneously button activation

If you want to activate two logical lines when press single button check **Use 2 buttons** and assign second line number.

5.4. Keyboard mapping

5.4.1. Overview

Keyboard item allows to map button to keystroke. **Quick logical layer wizard** dialog for this function is shown on Fig. 5.8.

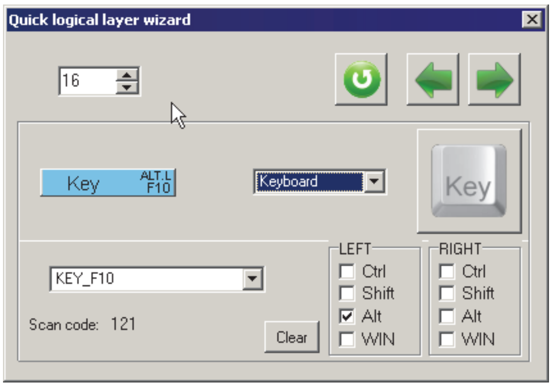


Fig. 5.8.

5.4.2. Keystroke assignment

You can assign single keystroke to button pressing physical key or choose it from combo box.

Some keystrokes for example functional keys can be assigned from combo box only.

Chosen keystroke will appear in combo box, In addition you will see its scancode.

When you work with keystroke mapping note that any key press will affect this function. Use mouse only to control process unless you really assign keystroke.

5.4.3. Keyboard modifiers

You can use keyboard modifiers *Ctrl, Alt, Shift, Win*. Right and left modifiers are distinct. You can use modifiers pressing physical keys or checking corresponding checkboxes. Simultaneously modifiers use (RCtrl+RShift+A) is allowed.

5.4.4. Mapping completion

In order to complete keystroke mapping and go to the next line processing press **Previous** or **Next** button. **Clear** button allows to cancel al current settings change.

5.5. Mouse control

5.5.1. Overview

Mice item allows you to control mouse axes, scroll wheel and buttons using joystick buttons.

5.5.2. Mouse buttons

Quick logical layer wizard dialog for this function is shown on Fig. 5.9.

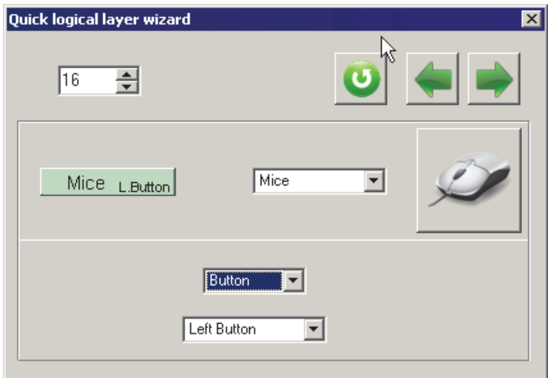


Fig. 5.9.

In order to control mouse buttons choose **Button** item.

Button type — left, right etc. — choose from combo box. **Mice On/Off** item allows to switch mouse control. You MUST assign this function to one of buttons if **On/Off** item from **Active** combo box was chosen in **Mouse** group on **Global — Common** tab (see 2.4.11 on p. 23). If **Absolute** mouse type was chosen then **Set center point** item will place mouse cursor to screen center.

5.5.3. Mouse axes control

Axis item (Fig. 5.10) allows to control mouse axis using button. You can choose mouse axis — **X** or **Y**, or mouse **Wheel**. Assign direction — **Up** or **Down** and velocity multiplier. If multiplier is equal to zero, cursor autoacceleration will be used.

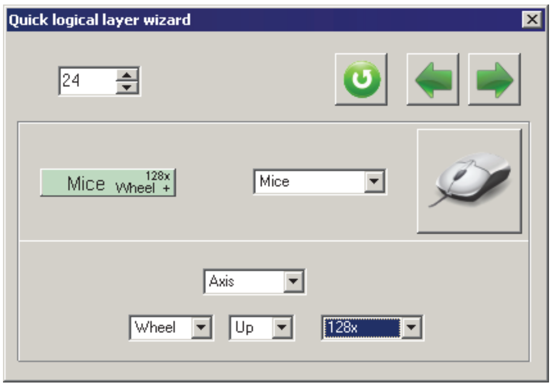


Fig. 5.10.

5.6. Macro

5.6.1. Overview

Macro is a named keystroke combination saved in a file that can be called by single button press. See Chapter 7 on p. 110 how to prepare macro files.

Macro item allows you to launch macro using joystick buttons.

5.6.2. Macro assignment

Quick logical layer wizard dialog for this function is shown on Fig. 5.11.

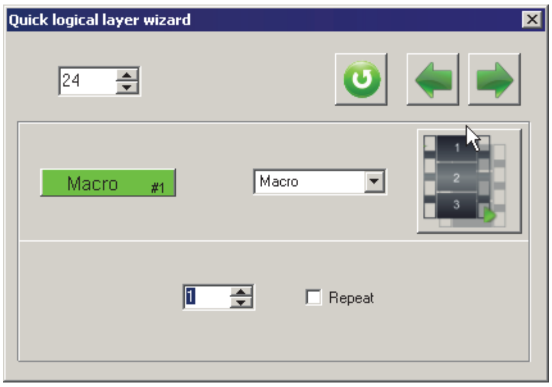


Fig. 5.11.

Set macro number using counter. If you want to automatically repeat macro if its button stays pressed at macro end check **Repeat**. If unchecked you must restart macro manually.

> You MUST set macro parameters and SAVE it on **Macro** tab BEFORE Macro function assigning to a button.

5.7. Sound control

5.7.1. Overview

Audio item allows you to control computer audio using joystick buttons.

5.7.2. Function setup

Quick logical layer wizard dialog for this function is shown on Fig. 5.12.

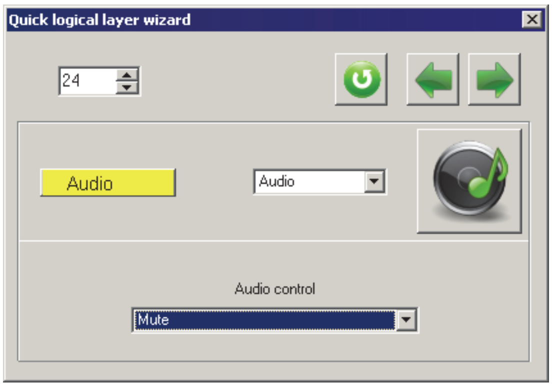


Fig. 5.12.

Select specific function (mute, volume increment and so on) for current button from **Audio control** combo box.

5.8. Multimedia control

5.8.1. Overview

Multimedia item allows you to control computer multimedia using joystick buttons.

5.8.2. Function setup

Quick logical layer wizard dialog for this function is shown on Fig. 5.13.

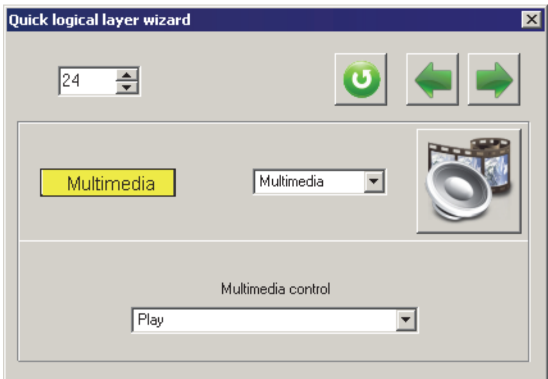


Fig. 5.13.

Select specific function (mute, volume increment and so on) for current button from **Multimedia control** combo box.

5.9. Application launch

5.9.1. Overview

Application item allows you to launch default applications for example e-mail client, word processor etc. using joystick buttons.

5.9.2. Function setup

Quick logical layer wizard dialog for this function is shown on Fig. 5.14.

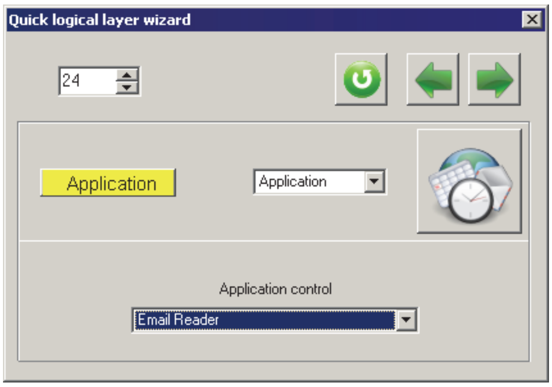


Fig. 5.14.

Select specific function (mute, volume increment and so on) for current button from **Application control** combo box.

5.10. System function control

5.10.1. Overview

System item allows you to execute some system functions such as power down, sleep etc.

Be careful using this function! Improper use can cause operation system malfunction.

5.10.2. Function setup

Quick logical layer wizard dialog for this function is shown on Fig. 5.15.

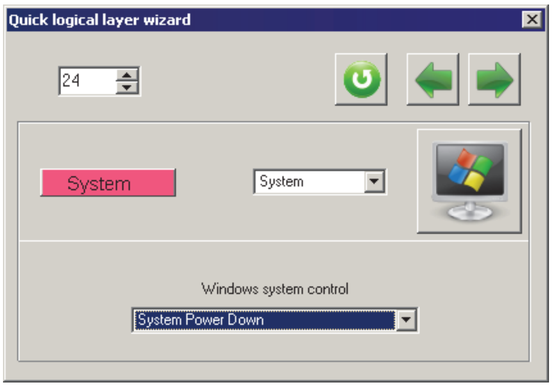


Fig. 5.15.

Select specific function (mute, volume increment and so on) for current button from **Windows system control** combo box.

5.11. Button deactivation

No Function item allows to deactivate button.

Chapter 6.HAT/POV parameters

6.1. Overview

Joystick can have up to four HATs. Physically discrete HAT is four-position switch. It consists of four pushbutton switches with shaft. There can be additional axis push button. HAT uses four lines. Analogue HAT is two-axes ministick. It can have push button too. Using NJoy32 controller you can configure POV#1 and POV#2 as analogue as well as discrete HATs. POV#3 and POV#4 can be configured as discrete devices only. Control units of **POVs** tab (Fig. 6.1) allow to configure POV#1 and POV#2.

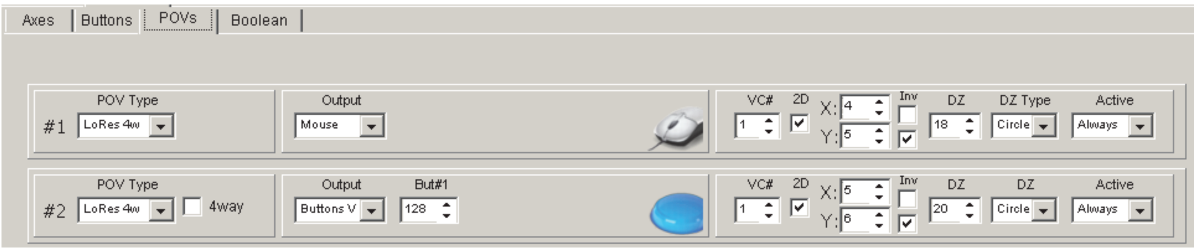


Fig. 6.1.

6.2. Ministick modes

Select ministick mode from **POV Type** combo box (Table. 6.1).

Table. 6.1.

Name	Description
Discrete	Standard discrete HAT. Detailed description of this kind of devices you can find in section 4.3.12 on p. 68.
LoRes 4W	4-way low resolution HAT.
LoRes 8W	8-way low resolution HAT.
HiResPOV	High resolution HAT. Not used in games at present time((.
Shifter 6W	6-position shifter. Virtual gear box.

6.3. Output mode

All ministick modes except **HiResPOV** can produce several types of output. Select desired output from **Output** combo box (Table. 6.2).

Table. 6.2.

Name	Description
POV	HAT. Point of View.
Buttons V	Logical buttons set.

Table. 6.2.

Name	Description
Buttons L	Virtual buttons set.
Numpad	Numpad buttons.
Mouse	Virtual mouse control.

6.3.1. POV

Ministick is represented as POV switch.

6.3.2. Virtual/Logical buttons

Ministick is represented as a set of buttons of corresponding level. Button number depends on ministick mode, **LoRes 8W** or **LoRes 4W**. Assign first button number of this set using **But#1** counter (Fig. 6.2).

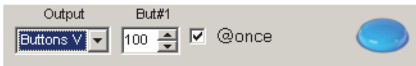


Fig. 6.2.

Other buttons will have next numbers. **@once** checkbox allows to use diagonal ministick pressing. If it is checked both buttons will be pressed simultaneously.

6.3.3. Numpad

Ministick works as numpad keys. Additional control units are enabled for this mode (Fig. 6.3).

Fig. 6.3.

When ministick is in the center *Numpad5* key can automatically be pressed. Select **Num5** item from **Center** combo box to enable this action. By default NoFunction logical function is assigned to *Numpad5* key. Check **C, S, A, W** to add modifiers *Ctrl, Shift, Alt, Win* left (row **L**) or right (row **R**).You can map a keystroke to it. To do so click rectangle with default function name. **Quick logical layer wizard** dialog appears (Fig. 6.4).

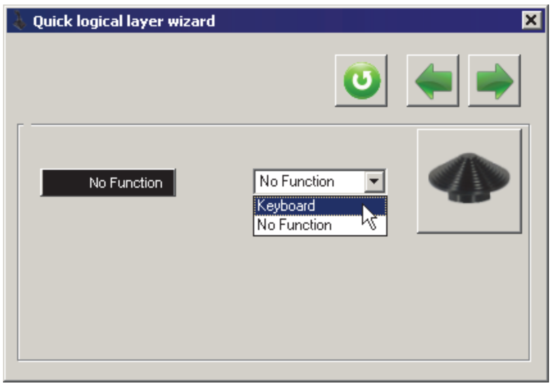


Fig. 6.4.

Keystroke mapping is described in section 5.4 on p. 100.

You must ENABLE keystroke mapping (see 2.4.9 on p. 22).

If mapping is disabled but you have decided to use ministick as numpad warning message appears (Fig. 6.5).

Fig. 6.5.

6.3.4. Mouse

Enable or disable virtual mouse using control units on **Clobal – Common** tab (see sect. 2.4.11 on p. 23).

6.3.5. Shifter 6W

Ministick is represented as set of 6 buttons or POV positions. If output is configured as POV, you can imagine Shifter as 8-posiiton POV without 2 horizontal positions.

Set first button number using counter **But#1** (рис. 6.6).

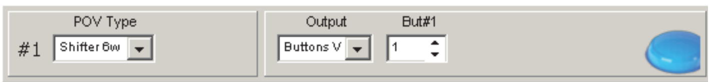


Рис. 6.6.

Button numbers for this example are shown on fig. 6.7.

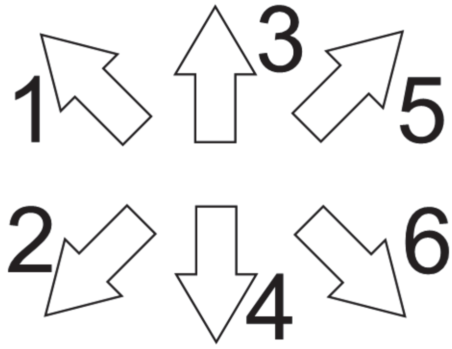


Рис. 6.7.

6.4. Ministick axes binding

If you selected **LoRes 8W** or **LoRes 4W** ministick mode, you must bind it to joystick axes using control units, shown on Fig. 6.8.

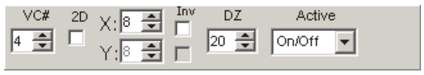


Fig. 6.8.

You can bind ministick to ANY joystick axes similarly to virtual mouse (see 2.4.11 on p. 23). If you want to know what physical axes numbers uses ministick in fact, open **Test** tab, move ministick and see its response (see 9.1.2 on p. 117).

Control units functionality is shown in Table 6.3.

Table. 6.3.

Name	Description
VC#	Virtual controller number.
2D	If checked ministick has two axes binded to pair of orthogonal axes. If unchecked ministick has single X axis.
X,Y	Counters allowing to assign joystick axes numbers binded to X and Y ministick axes.
Inv	Axes inversion.
DZ	Deadzone size in percents of full response range. Specifies button pressing treshold for **LoRes 8W** or **LoRes 4W** modes or deadzone for **HiResPOV**.
Active	Ministick activity mode. If **Always** mode selected it is active permanently. If **On/Off** you must assign button to control ministick (see 4.3.13).

Chapter 7.Macro setup

7.1. Overview

Macro can be described as a named sequence of logical button actions i.e. simple button press, keystrokes and so on. Each action or macro element is named as **point**. By default macro consists of four points. You can concatenate several macros to increase points total. Each single macro can have specific timing – whole point duration and a part of this time in which virtual button stays pressed.

7.2. Macro parameters

7.2.1. Overview

Control items of **Macro** tab (Fig. 7.1) allow to set up macro parameters.

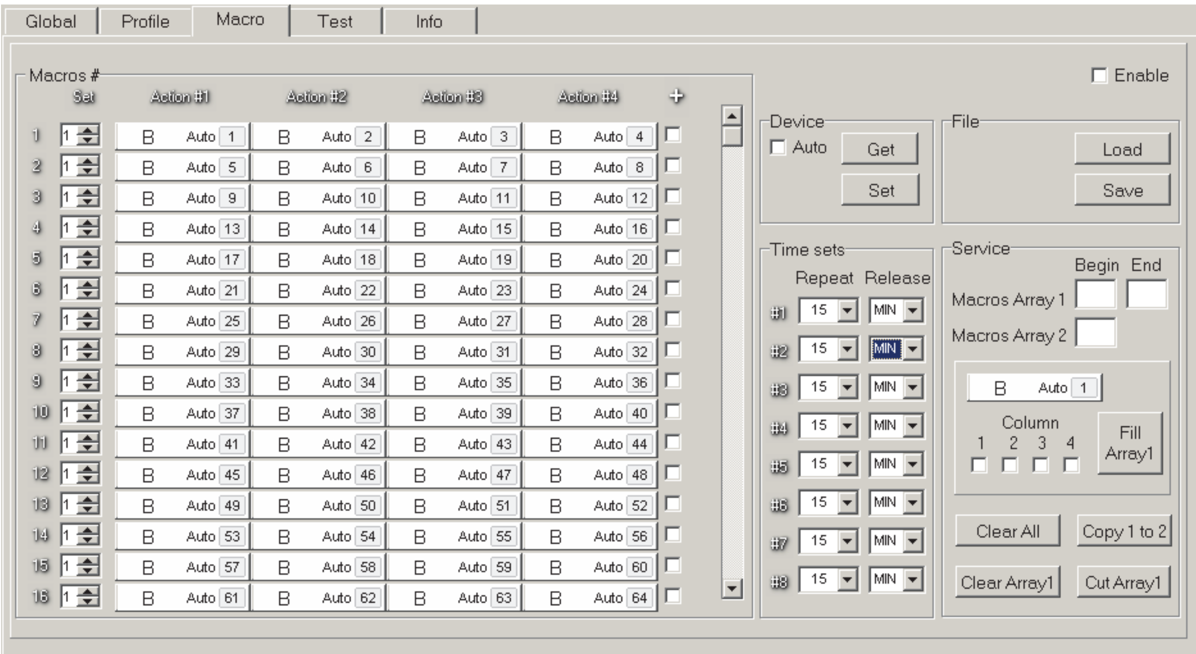


Fig. 7.1.

Each macro has its own number – up to 118. Number of the macro is its name. This parameter must be specified when you set up macro function (see 5.6 on p. 102).

7.2.2. Point parameters

In order to set up point parameters click its cell. **Quick logical layer wizard** (Fig. 7.2) will appear.

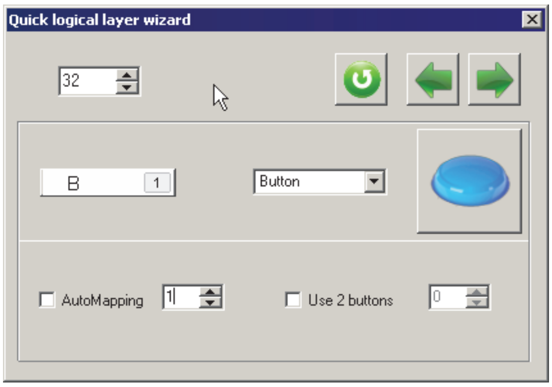


Fig. 7.2.

Each point is a logical button function. You can find detailed description of these functions in Chapter 7 on p. 110.

Check **Continued macro** to append **next** macro to current one. You can append not only next macro but one with specific number. In this case assign **Macro** function to the last point of current macro and set desired number for it (Fig. 7.3).

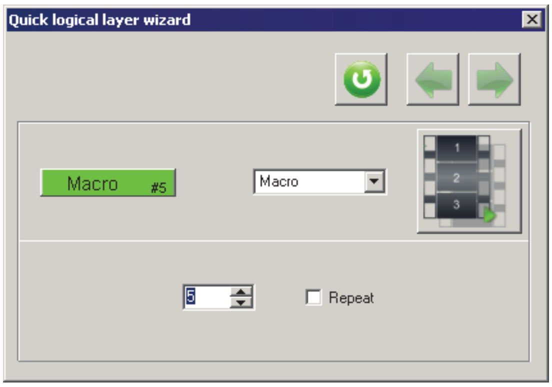


Fig. 7.3.

7.3. Macro timing

Control items of **Time sets** group allow to set up macro timings. Eight time sets can be used. Choose point duration time (15 ... 1000 ms) from **Repeat** combo box. Choose duration of virtual button pulse from **Release** combo box (table 7.1).

Table. 7.1.

Release type	Pulse duration
MAX	**Repeat** value-8 ms.
1:4	0,25 of **Repeat** value.
1:2	0,5 of **Repeat** value.
MIN	8 ms

You can specify up to 8 different time sets and assign them to a macro using **Set** counter.

7.4. Operations with macro

After you have completed macro setup press **Set** button in **Device** group to load current parameters to controller. **Get** button allows to read parameters from controller.

Save button allows to save current macro parameters to file with *mcr* extension. **Load** button allows to read parameters from file. Do not forget to press Set button to load parameters to controller!

In order to let your macro work check **Enable**.

7.5. Group operations

Some settings can be applied to several macros and points simultaneously. Use control items of **Service** group for it (Fig. 7.4).

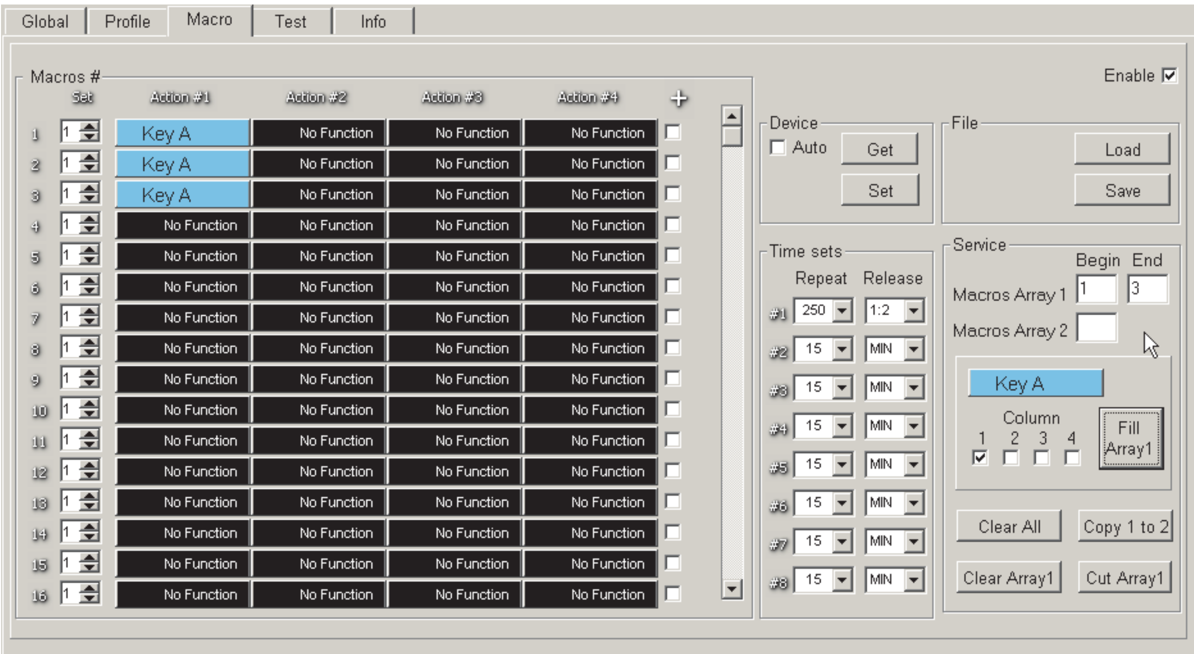


Fig. 7.4.

7.6. Point group clearing

Clear all button in **Service** group allows to reset all points of all macros.

Clear Array1 button allows to reset specified macro range. Counters **Begin** and **End** in **Macros Array1** group (Fig. 7.4) specify range limits.

7.7. Filling point array

You can fill point array with the same functions. For example Action#1 column of macros 1...2 is filled with *A* keystroke on Fig. 7.4. Click logical button cell (Fig. 7.5) and assign desired function in **Quick logical layer wizard** dialog that will appear.

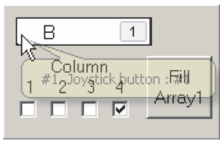


Fig. 7.5.

See detailed description of logical functions in Chapter 5 on p. 96.

Set macro range limits using **Begin** and **End** counters in **Macros Array1** group **Macros Array1** and column range limit using **Column** checkboxes, then press **Fill array1** button.

7.8. Using clipboard

Copy 1 to 2 allows to copy specified macro range to clipboard (counters **Begin** and **End** in **Macros Array 1** group specify range limits) and paste this macros from macro number specified in **Macros Array 2** field. Current macro assignments will be lost.

Chapter 8.Service functions

8.1. Loading parameters

You **must** press **Set** button to load current parameters into controller memory and apply them.

8.1.1. Forced loading

Versions of controller firmware and configuration tool must correspond. Otherwise some functions can malfunction. Profiles i.e. parameter sets also must correspond to firmware version. In some cases it is better to reassign all axes and buttons all over again. By default loading nonmatching profiles is forbidden. But if you want to load parameters anyway, for your own risk, edit configuration file *zconfig.ini* saved in the same folder that *VKBDevCfg.exe*. Add *ForcedWriteID*=1 string to *[User]* section, save this file and restart *VKBDevCfg.exe*. You will be able to load any profiles regardless of versions. If profile does not match to firmware you will see warning message (Fig. 8.1).

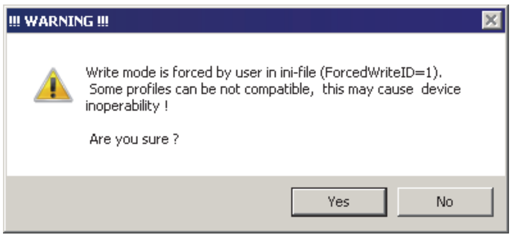


Fig. 8.1.

You must confirm or cancel loading using message buttons.

8.1.2. Partial parameter loading

Add *PartialWrite*=1 string to *[User]* section of *zconfig.ini* file and you will be able to load parameter set partially. For example you want to reassign button mapping only.

After you press **Set button Partial write settings** dialog will appear (Fig. 8.2).

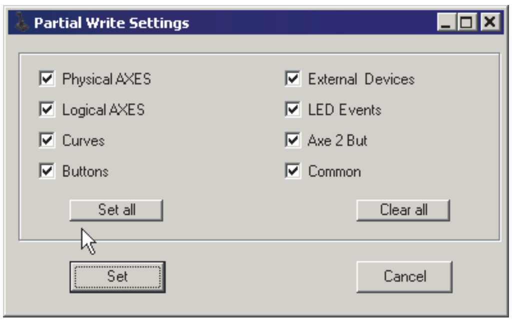


Fig. 8.2.

Check parameter names that will be loaded to controller. **Set all** and **Clear all** buttons allow corresponding choices. Press **Set** button to complete loading. **Cancel** button allows to cancel loading.

8.2. Current controller parameters

Get button allows to retrieve current controller parameters.

No matter what tabs of configuration tool are open. The whole parameter set will be retrieved.

If you have made some parameter changes and then decided not to load them to controller you can simply press **Get** button. It is possible only if you have not pressed **Set** button before.

8.3. Saving profile

You can save current profile, i.e. parameter set to file. To do so press **Save** button and choose folder and enter file name. By default file will have *cfg* extension.

8.4. Loading profile

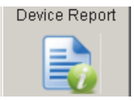
8.4.1. Load button using

Saved profile can be loaded into controller memory. Press **Load** button, open folder containing profile and choose file with *cfg* extension.

8.4.2. Drag-n-drop using

Using VKBDevCfg.exe starting with 0.85.06 version you can use drag-n-drop to load profile from file. Simply drag file name from windows explorer to configurator window and drop it. If **Set After Load** on **Settings** tab is checked then parameters will be saved to controller memory automatically. Otherwise you must press **Set** button as usual.

8.5. Button assignments report



You can create list of joystick controls assignment Fast Report format. Press **Device Report** button. **Print Preview** window will appear (Fig. 8.3).

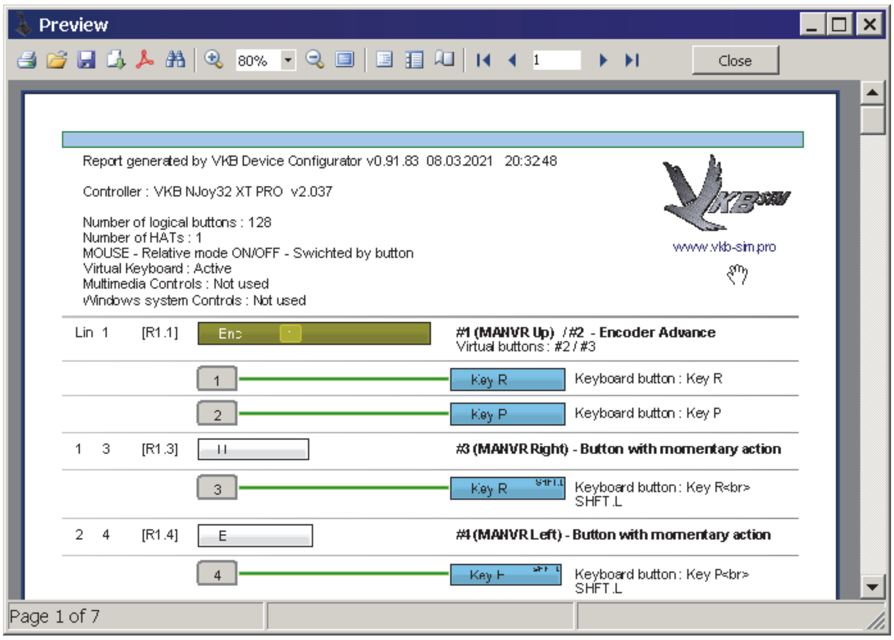


Fig. 8.3.

Physical and logical layer joystick controls assignments are shown in this window. Hover line number to see its assignments. Callout with needed data will appear (Fig. 8.4).

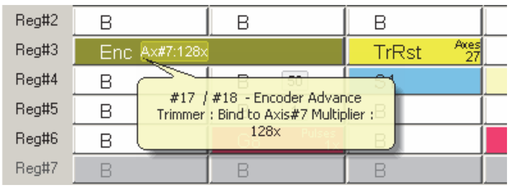


Fig. 8.4.

Chapter 9.Testing controls

9.1. Testing using configuration

Open **Test** tab.

9.1.1. Buttons testing

Open **Buttons/POVs** tab to test buttons, toggle switches, HATs, encoders etc. (Fig. 9.1). Activate controls and see results in this tab.

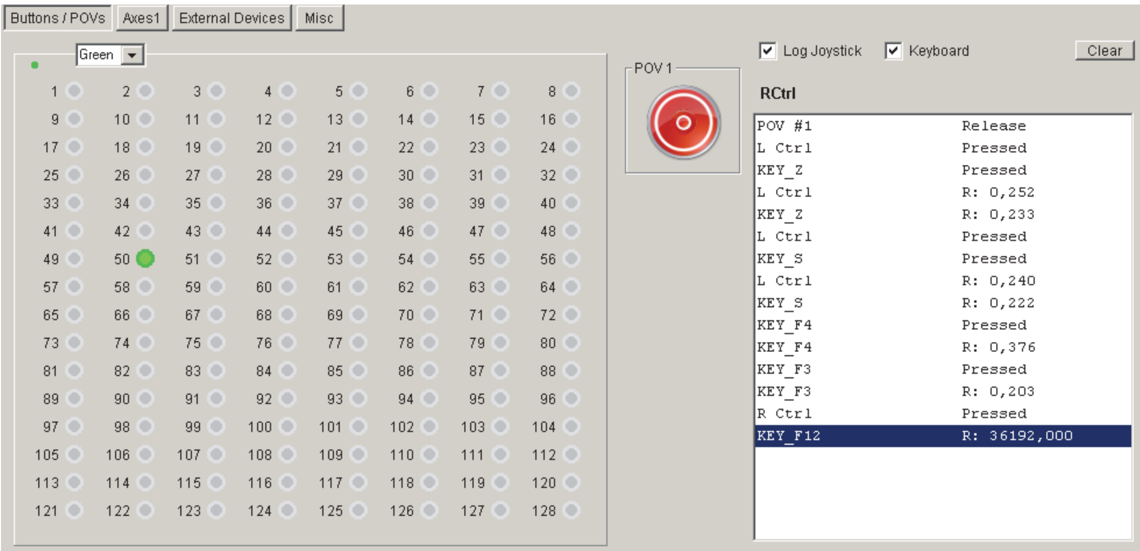


Fig. 9.1.

Check **Log enable** and **Keyboard** to test keyboard mapping.

If you have checked **Log enable** and **Keyboard** you must restart configurator after test completion.

9.1.2. Axes testing

Open **Axes1** tab to test joystick axes (Fig 9.2). Axis response is shown in graphical and digital forms.

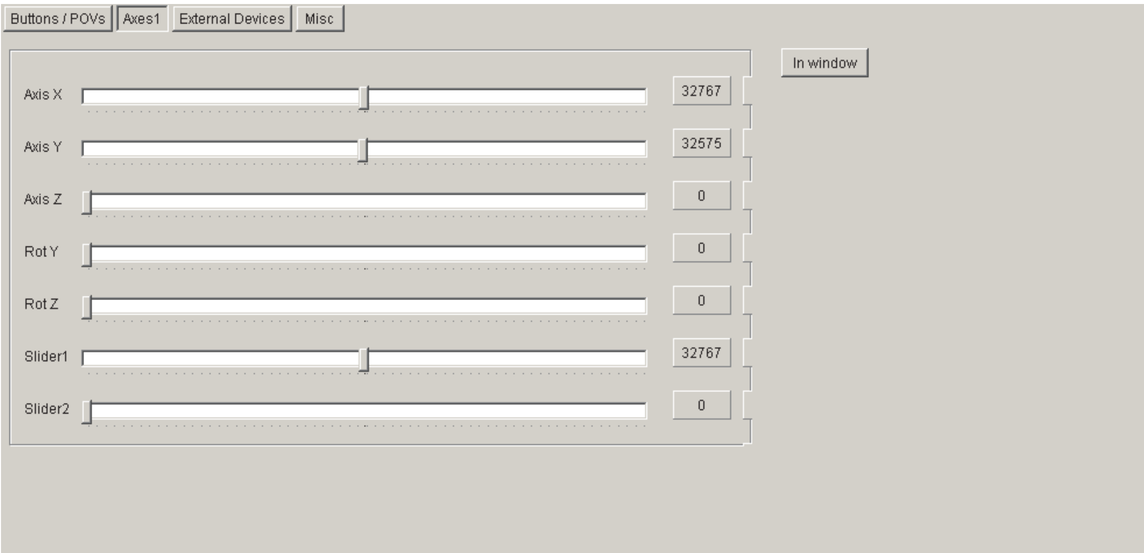


Fig. 9.2.

9.1.3. BUS testing

Open **External devices** tab to test BUS connection (Fig. 9.3). Connection error level will be shown. BUS can be tested for Master device only.

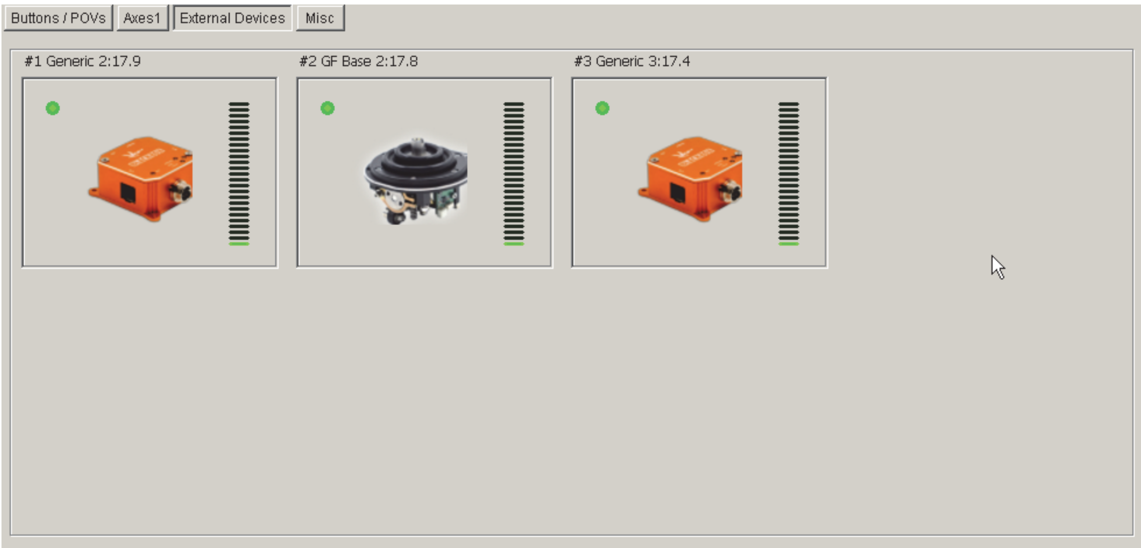


Fig. 9.3.

9.1.4. MARS and LEDs testing

Open **Misc** tab to test MARS sensors and LEDs (Fig 9.4).

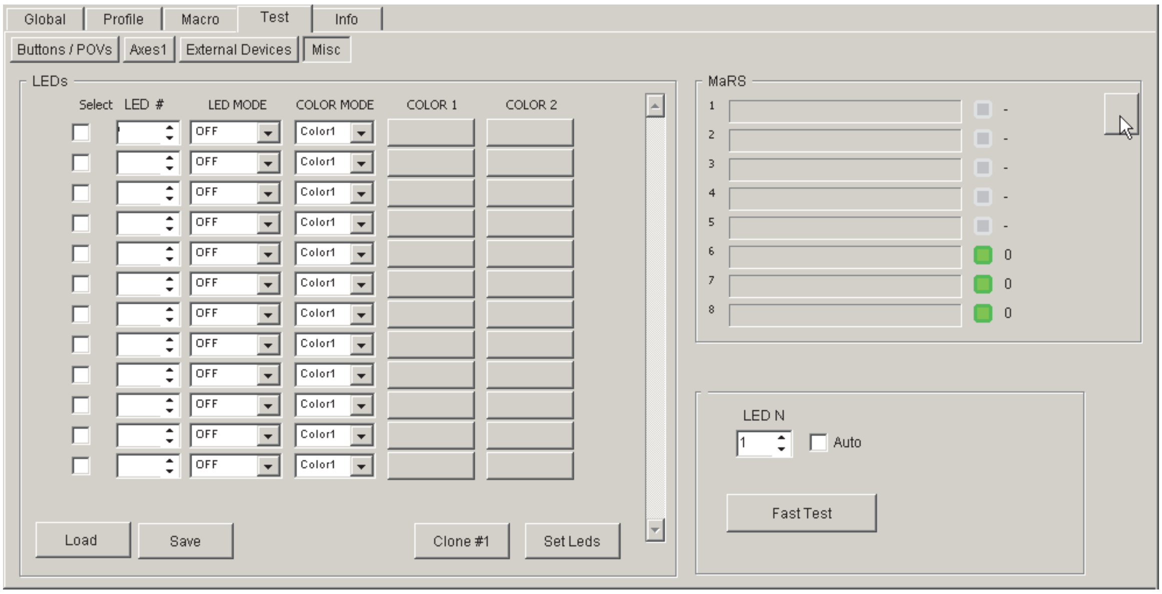


Fig. 9.4.

MARS testing
Press **MARS** button. If sensors work correctly you will see green marks. Red marks mean MARS failure.

LEDs testing
For testing purposes you can specify LED state. Choose LED number using **LED #** counter. Choose LED mode, color mode and color intensity. Press **Set Leds** button to apply settings.

Press **Restart** button on **Action** tab to restore current LED settings.

For instant test select LED number using **LED N** counter and press **Fast test** button. Chosen LED will flash.

You must append *Test Misc Enabled=1* string to *[Common]* section of *zconfig.ini* file to enable **Misc** tab.

Глава 10.Network technologies

10.1. Overview

Controller expansion ports allow to connect external devices:

- joysticks equipped with Njoy32 controllers –via BUS,
- Gladiator grip – 3-wire cable,
- combined devices such as Gunfighter base with MCG, SCG or F14 grips – 4-wire cable.

Gunfighter base with grips both have own controllers. Data packets from grip controller are transmitted to base controller via 3-wire digital interface. Data packets from the grip and base sensors are transmitted to BlackBox. BlackBox is connected to USB port. Base and grip are components of external device for BlackBox.

10.2. Hardware

BUS expansion ports realized as several connectors (table 10.1).

Table. 10.1.

Controller type	Connector	Note
Njoy32	Pins on PCB	
Mamba, ThrottleBox	RJ12 (6P6C)	
Gladiator, BlackBox	RJ12 (6P6C)	
BlackBox MkI	GX-12 4	
BlackBox MkII	GX-12 7	

BUS uses the following lines:

- VCC (+5B),
- Gnd,
- Tx1,
- Tx3.

Tx1 and Tx3 must have pull-up resistors. Gladiator and BlackBox controllers have preinstalled ones. If you want to use BUS with other controllers you must solder pull-up resistor in any circuit place.

External devices may be configured in two modes:

- all of them are visible for computer separately,
- single controller is visible while others are hidden. Controls (buttons, axes etc.) of hidden controllers belong to master controller.

10.3. Expansion port settings

Use controls of **Global – External – External devices** tab to configure BUS (Fig. 10.1).

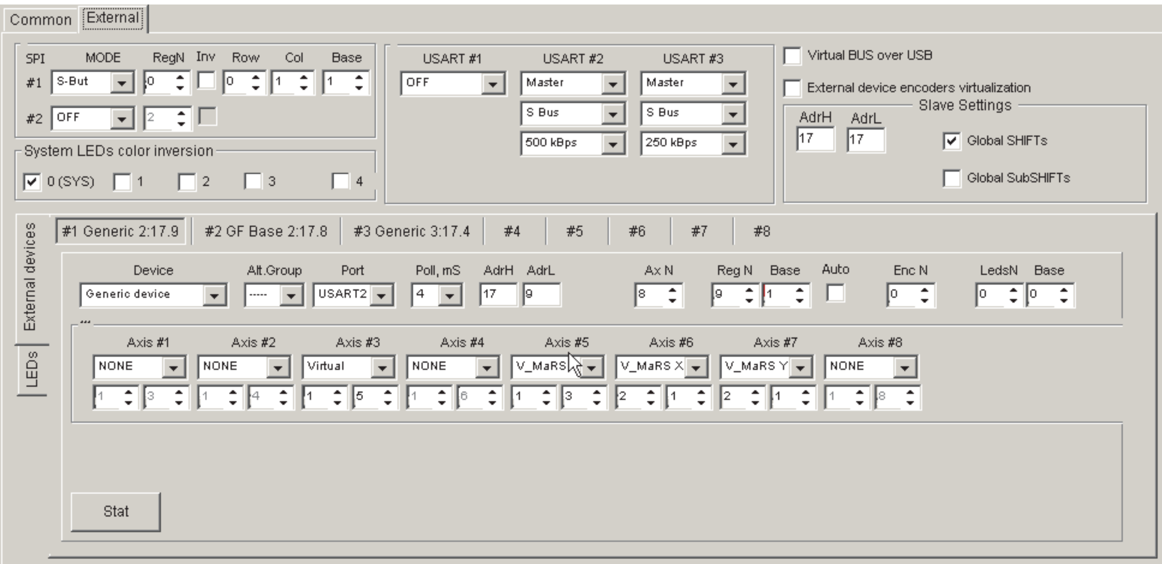


Fig. 10.1.

SPI1 port controls internal button registers. Select desired SPI1 mode from **SPI1 port mode** combo box:

- **OFF** — not used (GF+MCG for example, buttons come from external device, grip),
- **S-but** — standard button registers, 8 buttons each, connected.

Set registers number in **RegN** field.

Configure SPI2 using **SPI2 port mode** combo box:

- **OFF** — not used,
- **S-but** — standard button registers, 8 buttons each, connected.
- **S-LED** — RGB LEDs connected (see section 2.5 on p. 25).

Expansion devices are interconnected via BUS use USART #1 or USART #3 interfaces. Combined devises use USART #2. **USART #1**, **USART #2**, **USART #3 group controls** allow to configure interfaces. Select **Serial port mode – Master** or **Slave**, **Bus type** and **Speed** packets exchange rate.

10.4. Slave settings

Assign **Slave** (MCG, SCG, Gladiator grips, Njoy32-slave) high and low addresses, using **AdrH** and **AdrL** fields.

Do not use the same addresses for combined devices. Grip controllers have fixed addresses and can not be changed.

Check **External device encoders virtualization** to send encoder data via bus. Check **Global SHIFTs** or/and **Global SubSHIFTs** to use local modifiers for external devices. Check **Virtual BUS over USB** to use Zlink (see section 10.7 on p. 128) instead of cable connection.

Check **Ext** for all local controller axes you want to transmit to external devices on **Profile – Axes – Physical Axes** (Fig. 3.1 on p. 31) tab.

10.5. Master settings

Up to eight devices may be connected to master, #1...#8. Use controls on tabs with corresponding numbers to configure these devices.

10.5.1. Device types

Choose external device type from **Device** combo box (table. 10.2).

Table. 10.2.

Name	Description
Generic device	
ECS Throttle	Gametrix ECS Throttle
KG12 stick	KG12 grip
Gladiator Stick M	Gladiator grip, MARS twist.
Gladiator Stick R	Gladiator grip, potentiometer twist.
Gunfighter base	Gunfighter base
MCG	MCG grip
SCG	SCG grip

New devices will be added to list.

10.5.2. Device parameters

Device group types are listed in **Alt.Group** combo box. Several devices, for example NXT blocks, may belong to the same group. Group type assigns device presence analysis (table. 10.3).

Table. 10.3.

Group type	Device presence analysis
0 (....)	All group devices must be present. Absence of any device causes BUS Error.
1, 2	At least one device must be present.

Table. 10.3.

Group type	Device presence analysis
3	No BUS errors even no device found.

Select the same value from **Port** combo box with group **USART #1...#3**.

Set port polling period in **Poll, ms** combo box.

Set high and low device addresses in **AdrH** и **AdrL** fields.

Custom addresses are available for Generic Device only.

Set count of axes received from slave using **AxN** counter. Set count of button registers received from slave using **RegN** counter and **Base** – register number in a common register array for the first received register. Set count of received encoders using **Enc N** counter. Set count of received LEDs using **LedsN** counter and **Base** – LED number in a common LED array for the first received LED.

10.5.3. Axes parameters

Up to 8 axes can be received from slave. They must be checked as **Ext** in this device. Set received axes parameters using **Axis #1...Axis #8** control groups. Received axes type names are listed in table (table. 10.4).

Table. 10.4.

Axis type	Description
None	Axis not received.
Virtual	Virtual axis.
Normal	Axis with full parameters processing.
Trimmer	Axis - trimmer of existing one.
V_Mars X, V_Mars Y	Analog trigger axis (MCG Pro), Gunfighter base sensors.
V_Mars	Analog brake axis (MCG Pro).

Set virtual controller of master devise number using **Bind extern axis to controller # (1–8)** for received axis, usually 1. Set master axis number that will receive axis from using **Bind extern axis to axes # (1–8)** counter.

Example >.

All axes of slave-device allowed for transmission so **Ext** checked for all of them. Master device will use axes 4,5,6. Select **None** for Axes 1...3, 7 and 8

Select **Normal** for axes 4, 5 and 6 – additional axes data processing is not needed for them (Fig. 10.2).

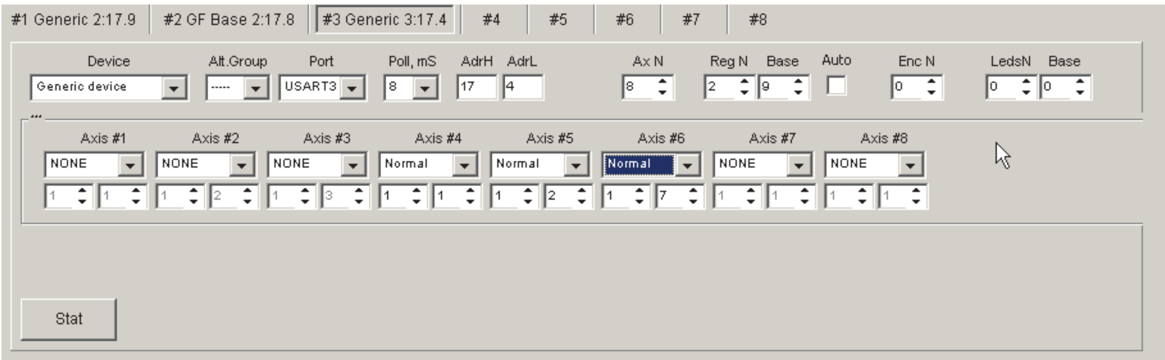


Fig. 10.2.

10.6. Combined devices parameters

10.6.1. Gunfighter base and MCG grip

Open **Global – External – External devices** tab (Fig. 10.3).

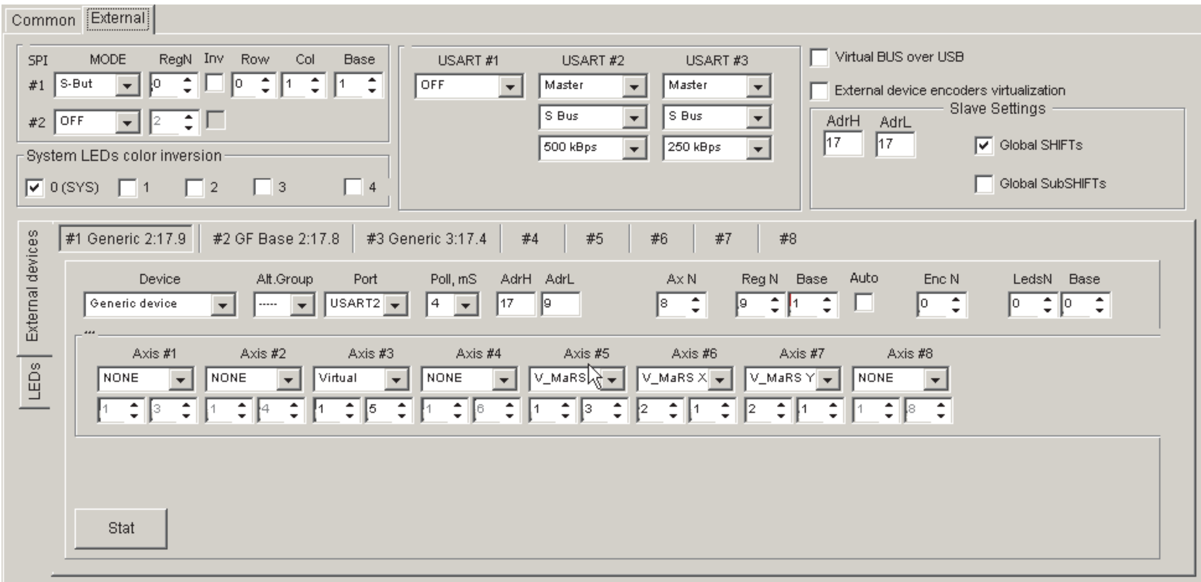


Fig. 10.3.

10.6.2. USART #2 parameters

Combined device components use **USART #2**. Set parameters as it is shown on Fig. 10.3.

- Serial port mode – Master,
- Bus type – S Bus,
- Speed – 500 kbps.

10.6.3. Base parameters

Select **Gunfighter base** for device #2 from combo box **Device**. Main parameters are locked for example, *Port=USART2*. Default parameter values are:

- Poll = 4 ms,
- Device addresses AdrH=17, AdrL=8,
- Axes count Ax N=4.
- Virtual controller =1 for all axes.
- Select V-Mars X and Y for Incoming axes 1&2. Own axis number = 1.
- Select V-Mars X and Y for Incoming axes 3&4. Own axis number = 2.

10.6.4. Grip parameters

Select **MCG** for device #1 from combo box **Device**. Main parameters are locked for example, *Port=USART2*. (Fig. 10.4).

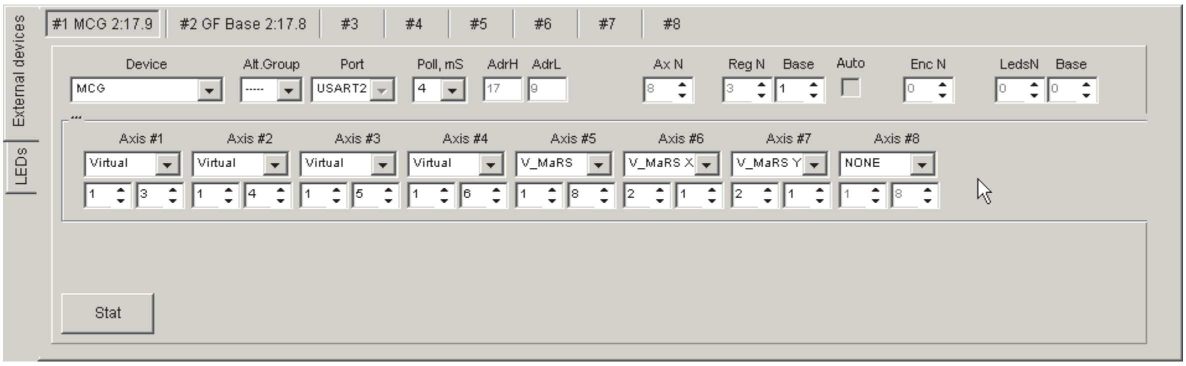


Fig. 10.4.

MCG Pro transmits 8 axes and 3 button registers. Default parameter values are:

- Poll= 4 ms,
- Device addresses   AdrH=17, AdrL=9,
- Axes count Ax N=8. Select sensors as shown in table 10.5.

Table. 10.5.

Incoming axis number, Ax #	Sensor type	Virtual controller number	Own axis number	Note
1–4	Virtual	1	3–6	Analog ministick sensors
5	V-Mars	1	8	Analog break lever sensor
6–7	V-Mars	2	1	Analog folded trigger sensor. Pro version only.

Table. 10.5.

Incoming axis number, Ax #	Sensor type	Virtual controller number	Own axis number	Note
8	V-Mars			Twist sensor (if exists).

Registers 1–3 are used for grip buttons. Register 4 is used for analog ministicks buttons and register 5 for analog brake lever buttons.

Open **Profile – Axes – Axes2Buttons** tab to configure folded trigger and break lever (Fig. 10.5).

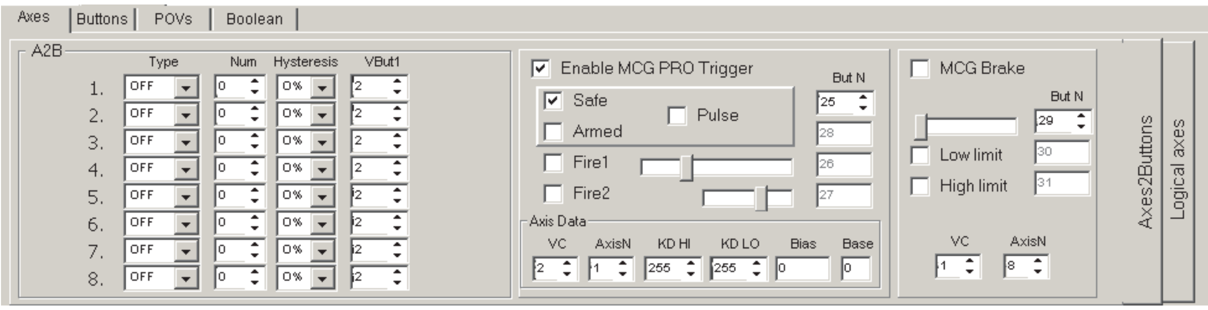


Fig. 10.5.

Folded trigger
Trigger is an axis. Assign virtual controller number using **VC** counter, and axis number using **AxisN**. **KDHi**, **KDLo**, **Bias**, **Base** parameters allow to fine tune axis.

Trigger has physical button. Check **Enable MCG Pro Trigger** to enable additional virtual buttons that are pressed on trigger path. Check **Safe** for unfolded trigger state button, **Fire2** – pressed. Sliders **Fire1** and **Fire2** set trigger positions for corresponding buttons. Use counter **But N** to set virtual button number for pressed position. Other number will follow. Check **Pulse** to use virtual buttons as toggle switches (see section 4.3.9 on p. 61). Button press duration is equal to **T_Tgl** value regardless of real trigger time in button position.

Break lever
Drake lever is an axis. Assign virtual controller number using **VC** counter, and axis number using **AxisN**. Check **MCG brake** to enable additional virtual buttons that are pressed on lever path. Check **Low limit** for default lever position, **High limit** – pressed. Use slider to tune intermediate button position. Use counter **But N** to set intermediate virtual button number. Other number will follow.

10.7. Connecting controllers via Z-Link

Z-Link program allows to convert NJOY32 controllers without wires. The following example shows connection between Mamba controller as slave and Black Box (Gunfighter MCG Pro) as master.

10.7.1. Z-Link parameters

Launch Z-Link executable. Fig. 10.6 shows Z-Link window.

Fig. 10.6.

Select slave controller that will transmit controls to master (Fig. 10.7).

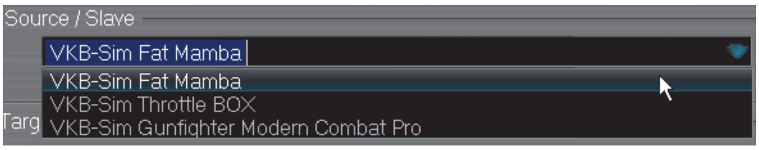


Fig. 10.7.

Remember pair of addresses in brackets (1.23 for Mamba controller). Select master controller.

10.7.2. Controllers setup in VKBDevCfg

Launch VKBDevCfg.exe.

Slave parameters

Select slave controller in the list, Fat Мамба, and open **Global – External** tab. Set **AdrH** и **AdrL** values taken from Z-Link (Fig. 10.8).

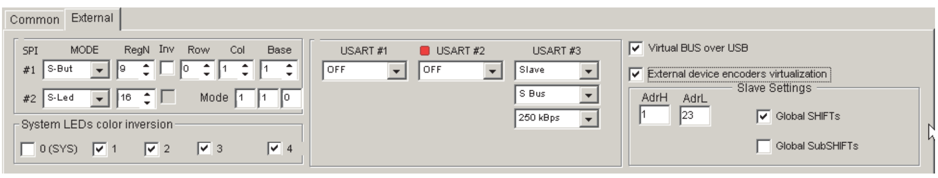


Fig. 10.8.

Check **Virtual BUS over USB**. Controller light will become green in Z-Link window, controller registered in the net. Check **Global Shifts** or/and **Global SubShifts** to use corresponding modifiers from master.

Master parameters

Select master controller in the list, GunFighter, and open **Global** – **External** tab. Configure device #3 as external virtual (Fig. 10.9).

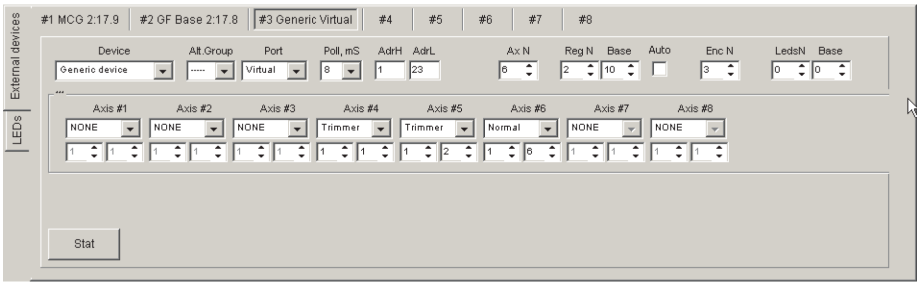


Fig. 10.9.

Select device type – **Generic device**, **Port** – **Virtual**, **Poll** = 8 ms. Enter pair of addresses from Z-Link, 1 and 23. Set received axes count, 6 for this example, Button registers count (2) and base register (10).

In this example axes 4,5 and of all six are used. On Mamba side they are virtual axes controlled by encoders. Set virtual controller #1 for all axes. Select **Trimmer** type for axis #4 and own axis number 1. Select **Trimmer** type for axis #5 and own axis number 2. Select **Normal** type for axis #6 and own axis number 6.

10.7.3. Z-Link work

When program is launched controls of slave controller are transmitted to master one. See axes transmission on Fig. 10.10.

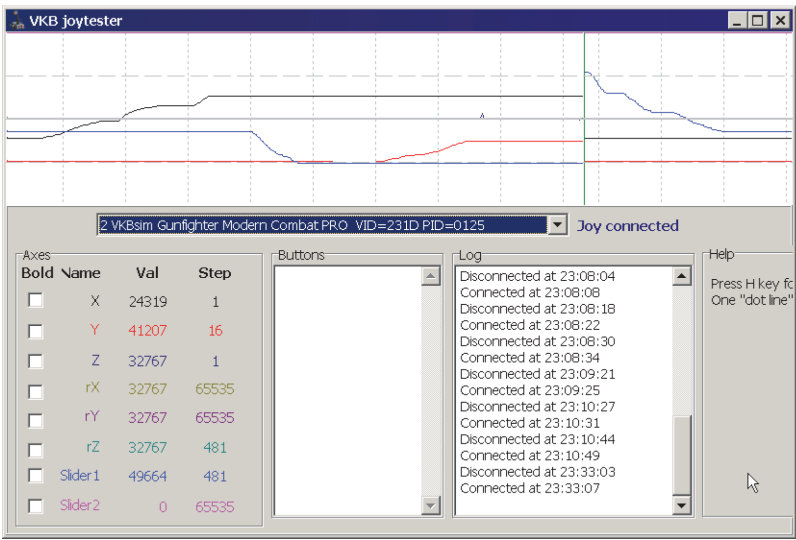


Fig. 10.10.

Appendix I.Zconfig.ini file description

Overview

When you launch configuration utility for the first time configuration file *zconfig.ini* will automatically be created in the same folder with *VKBDevCfg.exe* file. It contains miscellaneous utility parameters. Some of them MUST be appended by user because of they are not created automatically. See description of certain parameters in Table 10.6.

Table. 10.6.

Section name	Parameter	Description
[Common]	Use matrix=1	Allows **Row**, **Col**, **Base** fields on **Global — External** tab.
[Common]	PageControlTop=1/0	Tab positions on top/bottom.
[Common]	SwapPL=1	Axes panels position. For this value Physical axes panel is on top.
[Common]	Release_DI=1	Buttons and axes testing is enabled after parameters activation. In case of some problems set Release_DI=0
[User]	ForcedWriteID=1	Enables to load profiles created for firmware versions that are not current.
[User]	User=Developer	Enables Macro, LEDs, virtual mouse using.
[Common]	Test Misc Enabled=1	Enables MARS sensors and LEDs testing.