Using FreeJoy software with G-FORCE CP-28 control panel.

1/ **NOT CHANGE CONFIGURATION IN PIN CONFIG SECTION! NOT CHANGE FIRMWARE!**

2/ Load configuration from your panel to FreeJoy sofware and safe config to file in your computer as init backup file. Archive this file in case of restoring the correct functionality of the control panel.

3/ After loading configuration check PID number in section Advanced Setup. The PID number is last 4 digit of serial number. The serial number is located on the back side of your controler.

4/ For CP-28 series control panel is possible to configure only in the sections Button Config, Encoders and Advance Setting. Settings in other sections are not supported in this device.

5/ If you want to experiment with the controller settings, we recommend changing only the function and timer item in the button config section. Changing the mapping of the physical buttons will result in incorrect functionality of the controls elements (buttons, switches, encoders).

NOTE: ***The correct function of the control panel is guaranteed only with the factory configuration. How to save the configuration file is described in point 2.***

CAUTION: ***Changing firmware or configuration settings is at the customer's own risk, and the manufacturer does not provide any warranty for incorrect device functionality.***

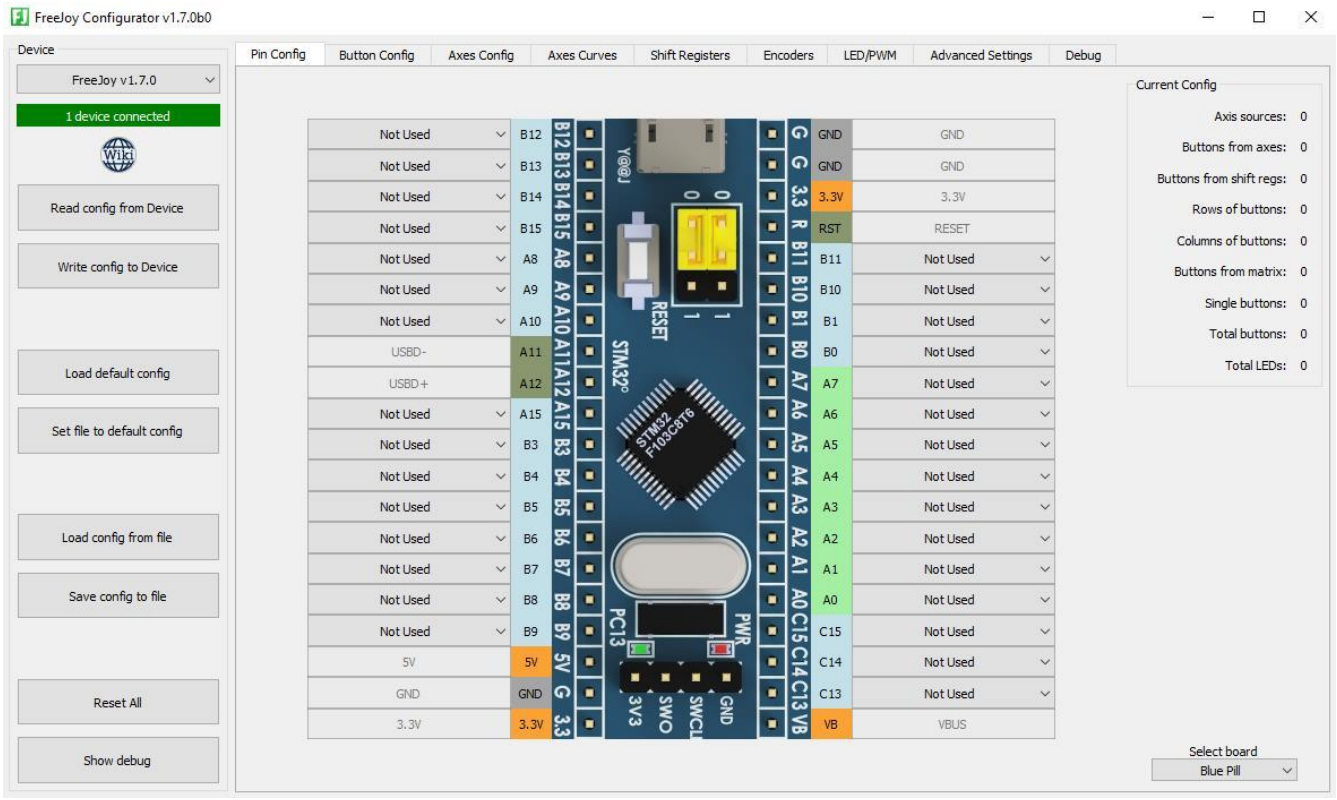
More about using FreeJoy with G-FORCE products you find on website: https://g-force.sk/

G-FORCE official YouTube channel: https://www.youtube.com/channel/UCLRczoJMzHSiavERZhjBXlg/videos

More information about FreeJoy can be found on the website: https://github.com/FreeJoy-Team

FreeJoy

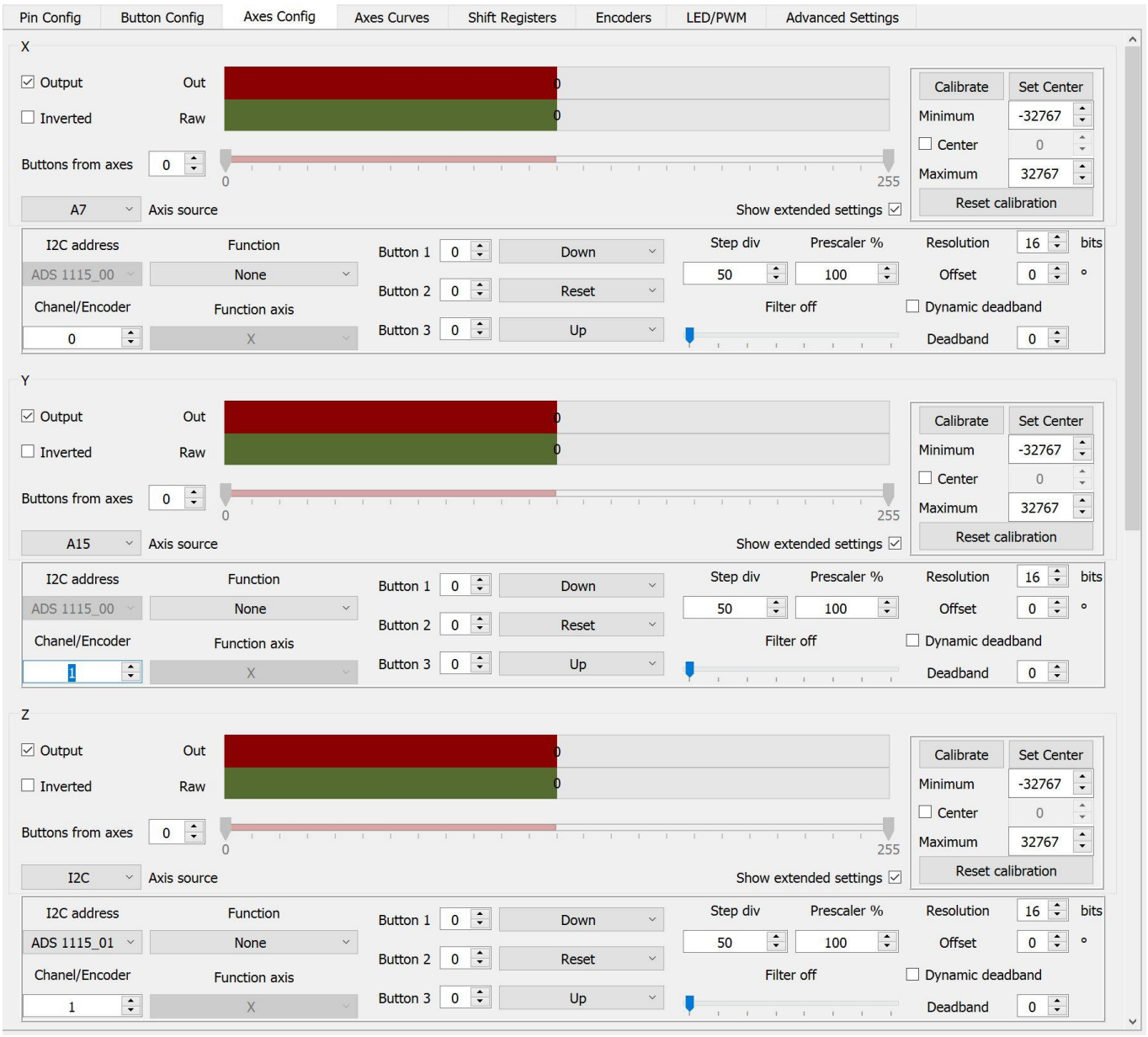
FreeJoy is a customizable gaming device controller based on the low-cost STM32F103C8 microcontroller. It allows you to create your own HOTAS-systems (sticks, throttles, various expansion panels), pedals, car control systems (steering wheels, pedals, gearbox shifters, etc.) and configure the designed device.



Features:

- Up to 8 - analog axes (output resolution 12 bits);
- Up to 128 buttons or toggle switches;
- Up to 4 HAT switches;
- Up to 16 incremental encoders (1 high-resolution encoder);
- Ability to assign button presses to certain positions of the analog axis (up to 12 buttons per axis);
- Support for shift registers 74HC165 and CD4021 to increase the number of connected buttons;
- Support for digital Hall sensors TLE5010/TLE5011, TLE5012B, AS5048A, AS5600, MLX90393 (only SPI interface);
- Support for external ADCs ADS1115 and MCP3201/02/04/08;
- 4 channels PWM for backlight control;
- 24 LEDs (single or in the matrix), mapped to the states of the buttons;
- Setting the device name and other USB parameters;
- Convenient utility for configuration;
- Upgrade firmware via USB;
- Save and load device configuration from file.

Axes:

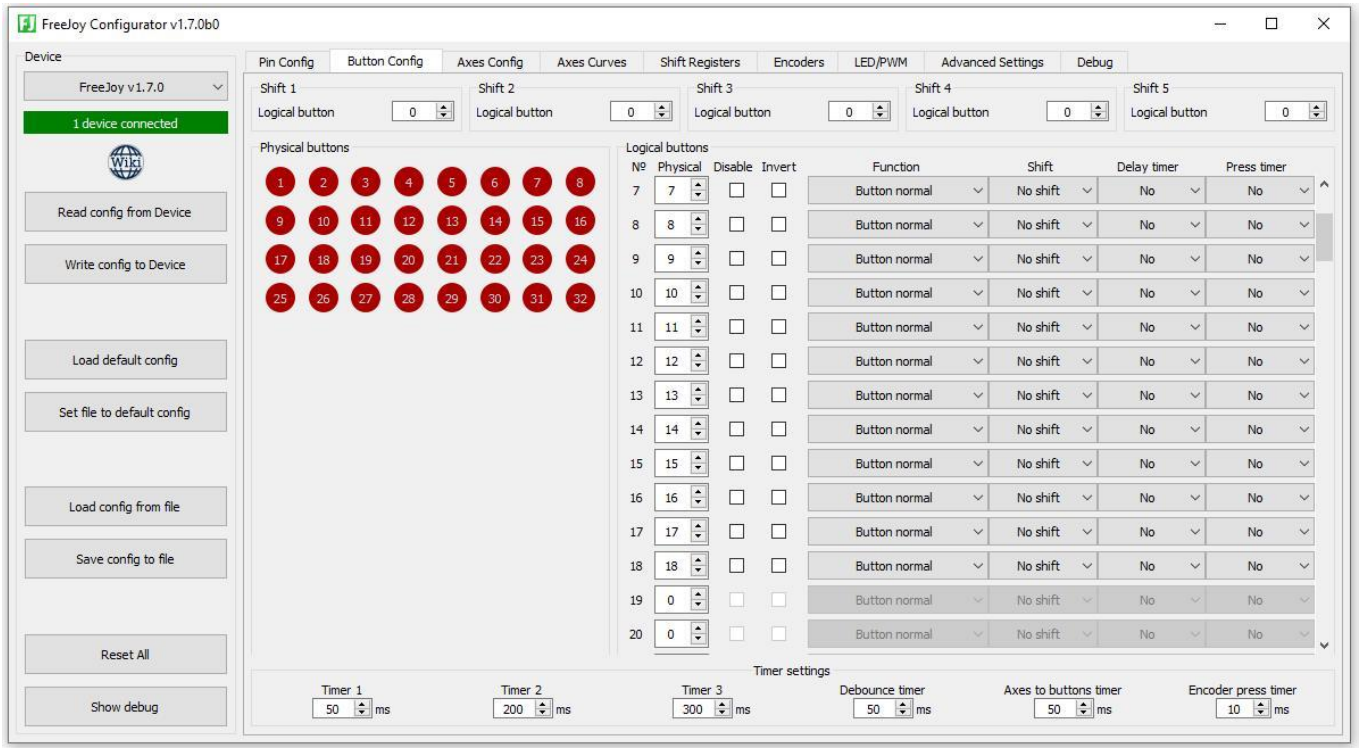


FreeJoy supports up to 8 axes. Analog inputs (potentiometers, hall sensors) on the A0-A7 terminals, digital sensors (TLE5010/5011, AS5600, MLX90393), or external ADCs (ADS1115 and MCP3201/02/04/08) can be used as sources for the axes. All axes have the following settings:

- Source/destination of the axis (X, Y, Z, Rx, Ry, Rz, Slider1, Slider2);
- Enable/disable axis output;
- Resolution;
- Calibration (manual / automatic);
- Smoothing (off or 7 levels of filter settings);
- Inversion;
- Dynamic or center deadband;
- Axis offset (magnet offset);
- Response curves;
- Axis from buttons/encoders;

- Trimming axis by buttons
- Axis prescaler
- Ability to generate button presses in certain axis positions (up to 12 sections).
- Combined axes functions

Buttons:



FreeJoy supports up to 128 buttons connected as single buttons (shorting the signal contact to GND or VCC), as a matrix of buttons, via shift registers or through the axis-to-button function. Buttons can be configured as:

- Normal button;
- Inverted button;
- Toggle switch on/off;
- Toggle switch on;
- Toggle switch off;
- HAT switch;
- Input incremental encoder;
- Radiobutton;
- Sequential button;
- Sequential toggle button
- 5 shift modificators.

The device configuration can be saved and loaded from a file. The configuration can be read and written to the device, you can also clear the configuration and reset it to the default settings. To do this, use the buttons located on the left side of the configurator.

Saving the configuration to file

To save the device configuration to a file, click the "Save Config to File" button. Select the folder to save, specify the file name and click "Save". The saved file has an XML extension and can be opened with any XML editor.

Loading configuration from file

To load a configuration from a file, click the "Load Config from File" button. Select the file to upload and click "Open." The configuration will be loaded into the program.

**Attention! Downloading a configuration file modified by third-party programs can lead to unpredictable device behavior! **

**Attention! Configuration files are not backward compatible! Do not use configuration files created in an earlier version! **

Read and write configuration to device

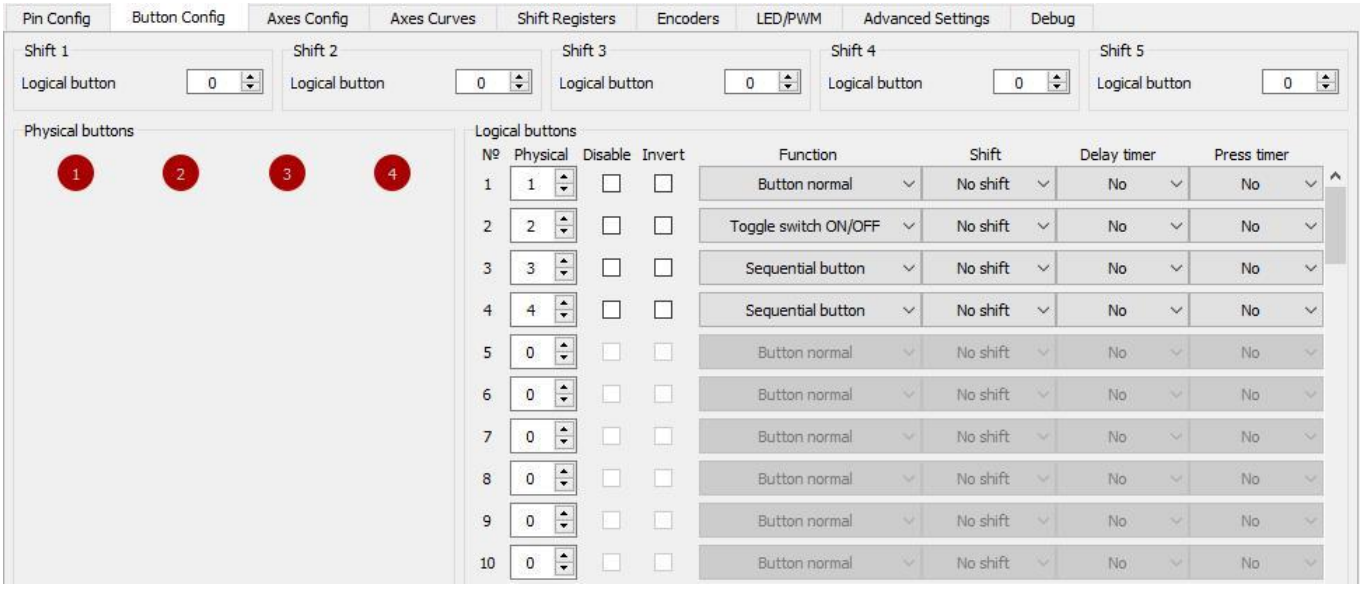
To read the configuration from the device, use the "Read Config from Device" button, and to write - "Write Config to Device".

Clearing the configuration and resetting the default settings is done with the buttons "Reset All Pins" and "Set Default Config", respectively.

Basic definitions:

- Physical button - a physically existing pair of contacts connected in some way to the controller (a single button, a button from the button matrix, a button connected to the shift register, or a toggle switch connected using one of these methods). The list of Physical buttons and their status (red - released, green - pushed) can be seen in the Physical Buttons field of the Button Config tab.
- Logical button - information transmitted to the computer about pressing a button. A list of logical buttons, physical buttons assigned to them, button functions, and the shift modificators assigned to them is displayed in the Logical Buttons field of the Button Config tab.

Assignment of logical buttons (Button Config tab).



1. On the Button Config tab, in the Logical buttons field in the line with the number of the logical button, you must specify the corresponding number of the physical button. (You can find out the number of a specific physical button by pushing it. In the Physical Buttons field, the circle with the number of pressed physical button is highlighted in green)

After assigning a logical button and writing the configuration to the controller, you can check its operation. When you press the physical button the line with the description of the logical button will be highlighted in green in accordance with the logic of the button function. one physical button may be binded to any number of logical buttons.

Debounce time can be set for all buttons (simultaneously) in "Button debounce time" field. In this case, pressing the button shorter than the time specified in this field is not taken into account. Increasing this parameter allows you to avoid repeated operation of the buttons with a single press, if the buttons are of poor quality.

2. Button function:

- Button_normal - a normal button;
- Button_Toggle - toggle button, pressing of physical button inverts the state of the logical button (If the logical button was not pressed, then pressing the physical button will make it pressed until the physical button is pressed again).

Functions for toggle switches:

- ToggleSwich_OnOff - closing and opening contacts of the toggle switch connected to the physical button generates pressing of the logical button (the time while the logical button remains pressed is set by Press Timer field);
- ToggleSwich_On - closing contacts of the toggle switch connected to the physical button generates pressing of the logical button (the time while the logical button remains pressed is set by Press Timer field);

- ToggleSwich_Off - opening contacts of the toggle switch connected to the physical button generates pressing of the logical button (the time while the logical button remains pressed is set by Press Timer field);

Functions for HAT switches (multi-position buttons/hats):

- PovX_Down, PovX_Up, PovX_Left, PovX_Right, PovX_Center (where X is the number of the hat) - pressing the direction down/up/left/right of the corresponding hat. PovX_Center is only used for Alps RKJX hats and provides neccesary logic for its operation.

Radio Button Function:

- RadioButtonX (Where X is the number of the group of radio buttons) - all buttons belonging to one group of radio buttons will work as follows: When you press any of the physical buttons of the group all logical buttons of the group are released and only the logical button associated with the pressed physical button remains pressed. The logical button remains pressed until another physical button from this group is pressed. When the controller is turned on, the first radio button in the list will be turned on. Radio buttons can be used as shift modificators.

- Sequental_Button - sequential switching mode - assigning one physical button to several logical ones with the enumeration function, when the physical button is pressed, the current pressed logical button will be released and the next logical button from the list will be pressed.

- Sequental_Toggle - sequential switching toggle mode - assigning one physical button to several logical ones with the enumeration function, when the physical button is pressed, the current pressed logical button will be released and the next logical button from the list will be latched.

Eco Encoder Features:

- Encoder_A, Encoder_B - see Connecting encoders

Additional settings

There are some additional settings for buttons such as:

- Invert - logical state invesion;
- Disable - disabling system output for this button;
- Shift - setting activation modificator (shift);
- Delay timer - delay before logical button will be pressed after physical button press;
- Press timer - time while logical button will be latched

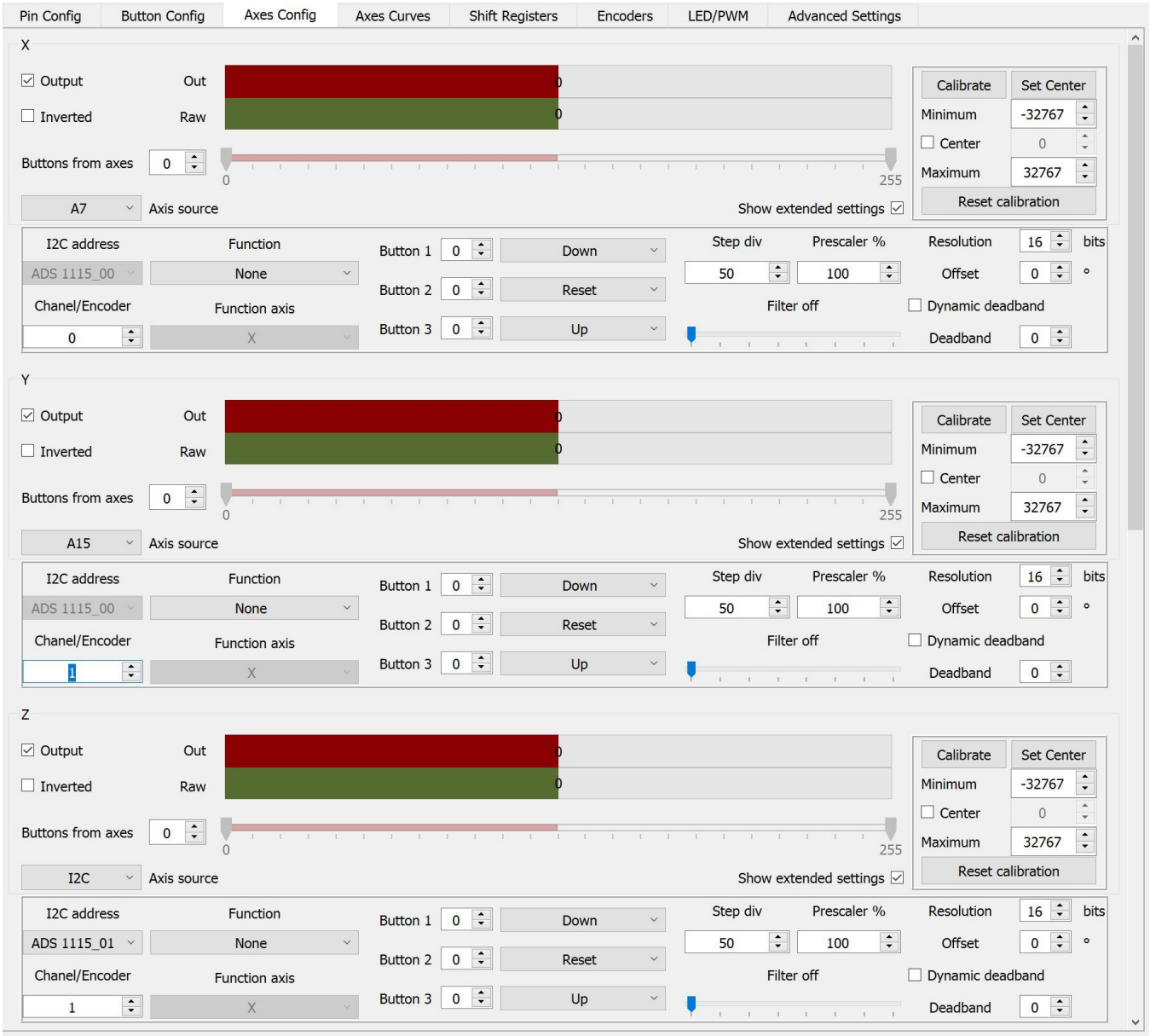
Using shift modificators:

Any logical button can be used as shift modificator. In this case, the logical button to which the shift modificator is assigned will work if two conditions are met: the logical button that is assigned by the shift modificator in the pressed state, and the logical button to which the shift modificator is assigned in the pressed state. If physical button has both logical buttons with shift modificator and without them, then the logical button which has no modificator will work only if shift modificators are not pressed.

Checking the operation of buttons

Finally, the operation of the buttons can be checked: (for Windows 10) “Start”, “Settings”, “Devices”, “Devices and Printers” with the right button on the device, “Settings for game control devices”, select the device, “Properties”.

Assignment of logical axes:



Like physical and logical buttons we must assign physical axes to the necessary logical axes.

The name of the analog axis output is indicated in the upper left corner of the field of this axis. There can be 8 axes in total: X, Y, Z, Rx, Ry, Rz, Slider1, Slider2.

You can assign any physical axis to any logical axis in the "Axis source 1" field. Wherein:

- for analog physical axes (potentiometer or Hall sensors) the designation will be the signal contact (middle contact of the potentiometer or the signal output of the Hall sensor, in the example an analog sensor is assigned to the X axis, the signal contact of which is connected to A7)
- for digital SPI sensors (TLE5010/5011, MLX90393, external ADCs MCP32XX) - chip select contact (in the example, the MLX90393 sensor whose SC pin is connected to A15 is assigned to the Y axis)
- for digital I2C sensors (AS5600, external ADCs ADS1115) - the name of the I2C interface is indicated, and field "I2C address" indicates the type of sensor and address (for ADS1115) specified by connecting the ADDR pin (in the example, the external ADC [ADS1115] is assigned to the Z axis (Connecting-analog-axes-to-ADS1115.md) with address 01)

In addition for multi-channel sensors (MLX90393) and ADC (ADC ADS1115 and MCP32XX) in the "Channel/Encoder" field, specify the channel number (channels are numbered from 0) from which we want to receive data. (in the example on the Y-axis, we take readings from the second channel of the sensor MLX90393, and on the Z-axis from the first channel of the external ADC ADS1115 with address 01). In case of encoder source you should specify encoder number from "Encoders" tab.

For unused axes in the project, uncheck Output enabled.

The red scale "Out" shows what values of the logical axis are issued to the operating system, the green scale "Raw" shows what values of the physical axis the controller receives from the sensor.

Axis calibration.

After assigning the axes, the axes must be calibrated (to make the full stroke of the physical axis correspond to the full stroke of the logical axis).

Calibrate an axis without a center.

1. Press the Start calibration button.
2. Move the axis from one extreme position to another.
3. Press the Stop calibration button.
4. Uncheck the “Center” checkbox.
5. Record calibration results to the controller.

Calibrate an axis with a center.

1. Press the Start calibration button.
2. Move the axis from one extreme position to another and leave it in the central position.
3. Press the Stop calibration button.
4. Record calibration results to the controller.

Calibration of the TLE5011 Digital Sensor with Range Offset.

If you installed a digital sensor and when moving it from one extreme position to another, the physical value of the axes pass through zero (i.e., they grow to the maximum value, reset to zero and continue to grow from zero), it is necessary to shift the range of sensor readings:

1. In the field of the logical axis put a daw "Offset".
2. Set “Offset” to 180.
3. Write settings from the controller.
4. Check by moving the axis that now the axis does not pass through zero.
5. If necessary, if the axis still passes through zero: correct the values of the “Offset” field, repeat steps 2-5.
6. Calibrate an axis as described above for axes with or without a center.

Additional axis settings:

The following settings are available in the field of each logical axis:

- Inverted - inverts the logical axis (the axis will be adjusted in the opposite direction);
- Resolution - resolution of the axis in bits. Moreover, the number of samples per axis movement range will be 2 to the extent indicated in the Resolution field. For analog axes, as a rule, values of no more than 12 make sense. For digital sensors, you can increase the resolution to 16 (65536 samples);
- Dinamic Deadband - setting for the axis of the dynamic deadband. The dynamic deadband value is indicated in the Deadband field below. In this case, changes in the value of the axis n will not be taken into account by a value not exceeding the value of the Deadband field. To eliminate axis jitter, it makes sense to gradually increase the values of the dynamic meter zone (not forgetting to write the configuration to the controller each time the value is changed) until the values of the logical axis cease to change spontaneously in the absence of movement of the physical axis. When the “Dynamic deadband” box is unchecked, the specified deadband size is applied to the central deadband;
- Filter - a slider that allows you to specify an axis filter. It can be: Off, Level1 ... Level 7. Moreover the higher the filter value the greater the lag of the change in the values of the logical axis behind the physical axis, so for critical axes it is not recommended to set the filter to a value higher than Level 3;
- Prescaler - you can set this value from 1% to 255% to scale your axis output;
- On the Axis Curves tab, you can specify a pattern of change in the values of each logical axis from a physical one other than linear. You can use the predefined curves (linear dependence (default), exponential, inverse exponential, arbitrary curves) or edit any of these curves by moving the slider corresponding to each point up / down.

Combined axes functions:

In the Function field, you can specify the method of adding / subtracting two axes.

In this case, the first source is the physical axis (indicated in the Axis source 1 field (hereinafter in the AS1 formulas), the second is the logical axis (indicated in the Axis source 2 field (hereinafter in the AS2 formulas). The following functions are available:

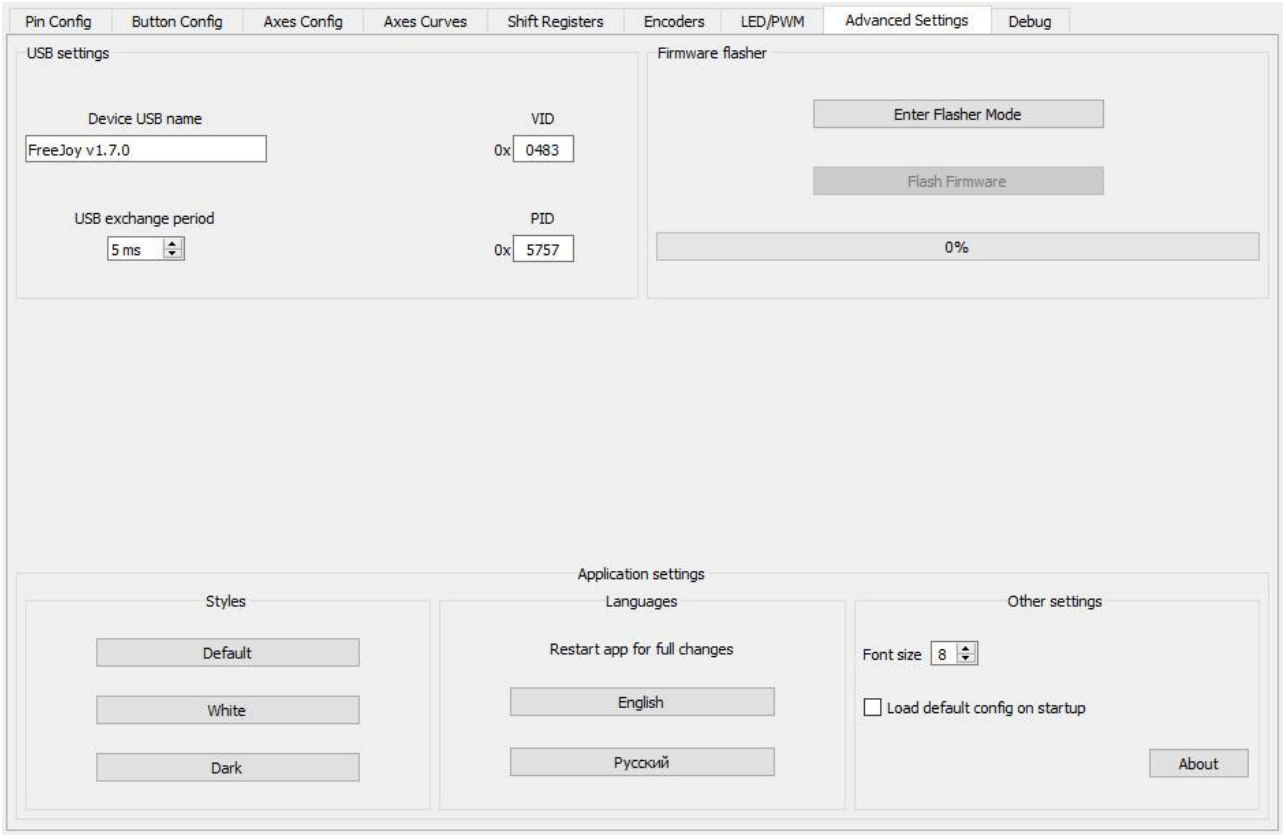
- Plus_Absolute = (AS1 + AS2) / 2
- Plus_Relative = (AS1 + AS2) +32767
- Minus_Absolute = (AS1-AS2) / 2
- Minus_ Relative = (AS1-AS2) -32767
- Equal AS1 = AS2

Buttons for axis functions

It is possible to set up to 3 buttons for each axis for these functions:

- Down - decrement axis value by value 32768/Step_Divider
- Up - increment axis value by value 32768/Step_Divider
- Reset - reset axis value modificatioin
- Center - set current raw axis value as output 0 value
- Function_Enable - enable combined axes function for this axis
- Prescaler_Enable - enable prescaler for this axis

Additional settings are located on the Advanced settings tab:



HID Settings:

- VID - The VID ID of the USB device;
- PID - The PID ID of the USB device. If your system uses more than one FreeJoy device, it is recommended to assign different PIDs for them;
- USB exchange period - the time between the serial sending of two data packets from the controller to the PC via USB.

Firmware Flasher

For more information about downloading firmware via the configurator see link below:

- [Firmware Loader] (Firmware-flasher.md)

Styles

It is possible to set one of several color styles for the configurator program:

- Default (Gray)
- White
- Dark

Languages

The configurator program supports switching interface languages. The following languages are currently supported:

- Russian
- English

Other settings

In this section you can change the font size of the application, as well as set the check mark for loading the default config at startup