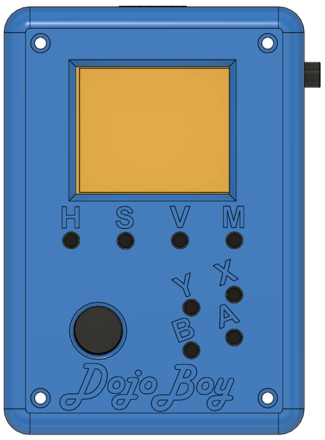


DojoBoy Class

Version 1.0 - 13/01/24

Import

from dojoboy_v1.dojoboy import DojoBoy

Class instance & init

djb = DojoBoy([show_intro=True][, width=xxx, height=yyy][, framerate=30][, refreshBeep=5])

width and height define screen resolution.

Constants

Color:

djb.display.BLACK_H or BLACK: black

djb.display.BLACK_L : dark grey

djb.display.WHITE_H or WHITE: bright white

djb.display.WHITE_L : bright grey

djb.display.BLUE_H : bright navy blue

djb.display.BLUE_L : dark navy blue

djb.display.CYAN_H : bright sky blue

djb.display.CYAN_L : dark sky blue

djb.display.MAGENTA_H : bright purple

djb.display.MAGENTA_L : dark purple

djb.display.RED_H : bright red

djb.display.RED_L : dark red

djb.display.YELLOW_H : bright yellow

djb.display.YELLOW_L : dark yellow

djb.display.GREEN_H : bright green

djb.display.GREEN_L : dark green

djb.display.color_pal[c]: c color index 0..15

Display:

djb.display.width : width of display

djb.display.height : height of display

Sound:

djb.sound.tone : tone C1, C#1,…,D8,D#8

Display

djb.display.fill(color) : fill display with color

Pixel and Lines (native methods):

djb.display.pixel(x,y[,color]) : draw pixel at x, y using the given color
djb.display.line(x1,y1,x2,y2,color) : draw line from x1,y1 to x2,y2 using the given color
djb.display.vline(x,y,h,color) : draw vertical line at x,y with h lenght using the given color
djb.display.hline(x,y,w,color) : draw horizontal line at x,y with w length using the given color

Shapes (native methods):

djb.display.rect(x,y,w,h,color[,fill]) : draw rectangle at x,y with size w,h using the given color. Pass 'True' as last parameter to fill rectangle.

djb.display.ellipse(x,y,rx,ry,color[,fill]) : draw ellipse at x,y with rx as vertical radius and ry as horizontal radius (rx = ry for circle) using the given color. Pass 'True' as last parameter to fill ellipse.

djb.display.poly(x,y,coords,color[,fill]) : draw polygon at x,y with array of relative coordonates to x,y (ex:array('h',[x0,y0,x1,y1,...xn,yn]) using the given color. Pass 'True' as last parameter to fill polygon.

djb.display.circle(x,y,rad,color[,fill]) : draw circle at x,y with rad as radius using the given color. Pass 'True' as last parameter to fill circle.

djb.display.triangle(x0,y0,x1,y1,x2,y2,color[,fill]) : draw triangle around three points at x0,y0,x1,y1,x2,y2 using the given color. Pass 'True' as last parameter to fill triangle.

Text :

djb.display.text(string, x, y, color[, scale=1]) : draw text at x,y using the given color. Set scale to 2 or 4 to scale up text.

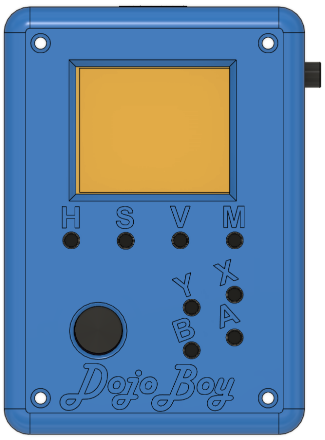
djb.display.center_text(string, color[, scale=1]) : draw text at center of screen using the given color. Set scale to 2 or 4 to scale up text.

djb.display.center_text_XY(string[,x],[y,] color[, scale=1]) : draw text at vertical or horizontal center of screen using the given color. Set scale to 2 or 4 to scale up text.

Refresh display:

djb.display.show() : show display.

djb.display.show_and_wait() : show and refresh display at defined framerate (default at 30 img/s).



DojoBoy Class

Version 1.0 - 13/01/24

Joystick and Buttons Control

Button constants:

djb.btn_Up	: Up button	djb.btn_Down	: Down button
djb.btn_Left	: Left button	djb.btn_Right	: Right button
djb.btn_A	: A button	djb.btn_B	: B button
djb.btn_X	: X button	djb.btn_Y	: Y button
djb.btn_Home	: Home button	djb.btn_Start	: Start button
djb.btn_Volume	: Volume button	djb.btn_Menu	: Menu button

Methods:

djb.scan_jst_btn() : poll joystick and buttons status

djb.pressed(button) : return 'True' if button pressed

djb.just_pressed(button) : return one time 'True' if button pressed

djb.just_released(button) : return one time 'True' if button released

Sprite

djb.display.add_sprite(buffer, w, h): add a sprite in sprite list. Buffer must be in RGB565 format. w and h are width and height of sprite in pixel. Return the index of sprite.

djb.display.add_sprite_from_file(filename, w, h[,format]): add a sprite in sprite list from file in RGB565 format. w and h are width and height of sprite in pixel. Return the index of sprite.

djb.display.add_rect_sprite(w, h, color): add a rectangular sprite with coloc in sprite list. w and h are width and height of rectangular sprite in pixel. Return the index of sprite.

djb.display.sprite(n, x, y[, transparent_color]): display sprint n at x, y. transparent_color if defined hide this color in sprite.

djb.display.sprite_width(n): return width of sprite n

djb.display.prite_height(n):return height of sprite n

Sound

djb.play_freq(freq, duration): play non blocking sound at freq Hz for duration

djb.play_tone(tone, duration]): play non blocking tone at tone note for duration

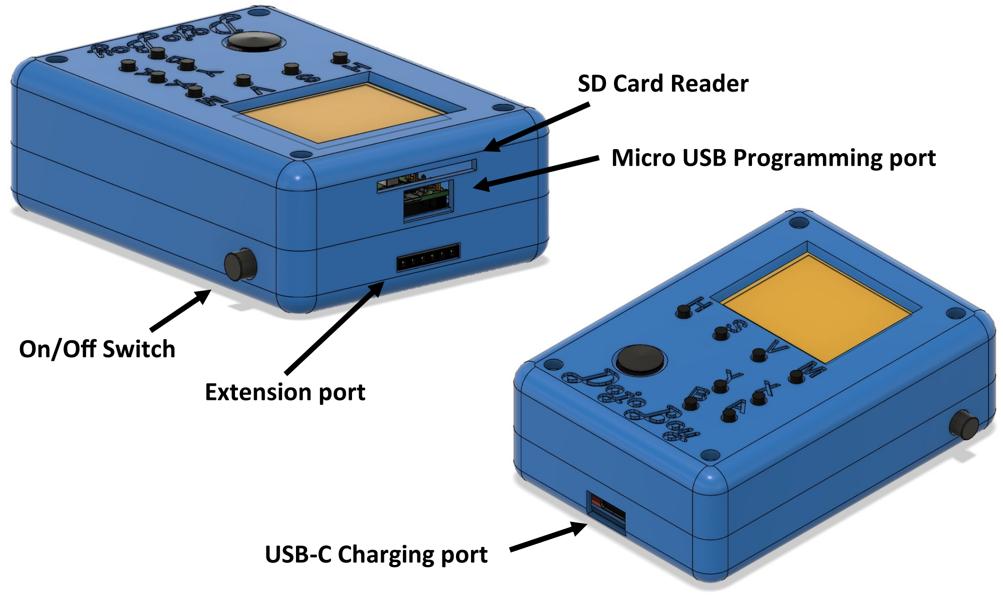
djb.bequiet(channel): silence channel 0 (sound—tone) or 1 (song)

djb.mute(status): disable all sound with True

djb.start_song(songBuf): play non blocking song

djb.stop_song(): stop playing song

Views


SD Card Reader
Micro USB Programming port
On/Off Switch
Extension port
USB-C Charging port