

## Key Combos

- Easiest to Write
- For Shortcuts & Hotkeys
- Any Combo of:
- Special Keys / Letters / Numbers

ENTER

CTRL s

COMMAND SHIFT 4

Type Key Name in **ALL CAPS**

## Full Docs

- [duckyPad User Manual](#)
- [duckyScript Instructions](#)

## Available Special Keys

CTRL / RCTRL		(media keys)
SHIFT / RSHIFT		MK_VOLUP
ALT / RALT		MK_VOLDOWN
WINDOWS / RWINDOWS		MK_MUTE
COMMAND / RCOMMAND		MK_PREV
OPTION / ROPTION		MK_NEXT
ESC		MK_PP (play/pause)
ENTER		MK_STOP
UP/DOWN/LEFT/RIGHT		
SPACE		(numpad keys)
BACKSPACE		NUMLOCK
TAB		KP_SLASH
CAPSLOCK		KP_ASTERISK
PRINTSCREEN		KP_MINUS
SCROLLLOCK		KP_PLUS
PAUSE		KP_ENTER
BREAK		KP_0 - KP_9
INSERT		KP_DOT
HOME		KP_EQUAL
PAGEUP / PAGEDOWN		
DELETE		(Japanese IME)
END		ZENKAKUHANKAKU
MENU		HENKAN
POWER		MUHENKAN
F1 - F24		KATAKANAHIRAGANA

# duckyScript Commands

## Timing

DELAY <b>n</b>	Wait <b>n</b> Milliseconds
DEFAULTDELAY <b>n</b>	How long to wait between <b>each line of code</b>
DEFAULTCHARDELAY <b>n</b>	How long to wait between <b>each keystroke</b>
CHARJITTER <b>n</b>	Additional random delay between 0 and <b>n</b> ms after each key stroke

## Typing

STRING <b>text</b>	Type text AS-IS
STRINGLN <b>text</b>	Same as above Press ENTER at end
STRING_BLOCK END_STRING	Type text block AS-IS
STRINGLN_BLOCK END_STRINGLN	Same as above Press ENTER after <b>each line</b>

## Mouse

LMOUSE	Left Click
MMOUSE	Middle Click
RMOUSE	Right Click
MOUSE_MOVE <b>x y</b>	Move Mouse in Pixels <b>x</b> : + Right - Left <b>y</b> : + Up - Down
MOUSE_WHEEL <b>n</b>	Scroll <b>n</b> lines + Up - Down

## OLED

OLED_CLEAR	Clear Screen
OLED_CURSOR <b>x y</b>	Set Cursor <b>x y</b> : 0 to 127 (0,0) = Top Left
OLED_PRINT <b>text</b>	Print Text at Current Cursor
OLED_CIRCLE <b>x y radius fill</b>	<b>x y</b> : Position <b>radius</b> : In Pixels <b>fill</b> : 0 or 1
OLED_LINE <b>x1 y1 x2 y2</b>	<b>x1 y1</b> : Start <b>x2 y2</b> : End
OLED_RECT <b>x1 y1 x2 y2 fill</b>	Draw Rectangle <b>x1 y1</b> : Start <b>x2 y2</b> : End <b>fill</b> : 0 or 1
OLED_UPDATE	<b>Call This to See Changes on Screen!</b>
OLED_RESTORE	Show Default Screen

## RGB Backlight

SWC_SET <b>n r g b</b>	Change RGB Colour <b>n</b> : Key ID (0 for current switch) <b>r g b</b> : 0 to 255
SWC_FILL <b>r g b</b>	Change ALL RGB <b>r g b</b> : 0 to 255
SWC_RESET <b>n</b>	Reset RGB to Default <b>n</b> : Key ID 0 = Current Key 99 = All keys

## Pressing Keys

Key Combos	See Page 1
KEYDOWN <b>key</b>	Hold key
KEYUP <b>key</b>	Release Key
<b>key</b> can be letter, number, or special key.	

## Profile Switching

PREV_PROFILE	
NEXT_PROFILE	
GOTO_PROFILE <b>name</b>	Case sensitive
Also check out <a href="#">duckyPad Autoswitcher!</a>	

## Miscellaneous

DP_SLEEP	RGB & Screen OFF Halts Execution
HALT	Halt Execution
REPEAT <b>n</b>	Repeat <b>line above</b> <b>n</b> times

## Comments

REM
//
REM_BLOCK END_REM

Constants	
DEFINE	Define a Constant
Replaced AS-IS during preprocessing	
DEFINE MY_EMAIL bob@me.com STRING My email is MY_EMAIL!	

Variables	
VAR	Declare a Variable
Starts with \$ Global scope 16-bit integer Can be printed	
VAR \$count = 10 STRING I have \$count eggs	

Operators		
Perform on Constants and Variables		
= Assign	== Equal	&& Logical AND
+ Add	!= Not equal	Logical OR
- Subtract	> Greater than	& Bitwise AND
* Multiply	< Less than	Bitwise OR
/ Divide	>= GTE	<< Left Shift
% Modulus	<= LTE	>> Right Shift
^ Exponent		

IF Statements	
Code inside is executed If <b>expression</b> is <b>non-zero</b>	
IF <b>expression</b> THEN code END_IF	
Multiple Checks	ELSE IF ... THEN ELSE

WHILE loops	
Code inside is repeated If <b>expression</b> is <b>non-zero</b>	
WHILE <b>expression</b> code END_WHILE	
Jump to start of loop	CONTINUE
Exit loop immediately	LBREAK

Functions
Run block of code efficiently
FUNCTION my_func() code END_FUNCTION
my_func() // call it
Use global variable for args & returns

Persistent Global Variables
\$_GV0 to \$_GV15
<ul style="list-style-type: none"><li>Available across all profiles</li><li>Persists over reboots</li></ul>

Reading Buttons: Blocking
VAR \$this_k = \$_BLOCKING_READKEY
Waits until a key is pressed, returns its ID

Reading Buttons: Non-Blocking
VAR \$this_k = \$_READKEY
Returns 0 if no key pressed, KeyID otherwise

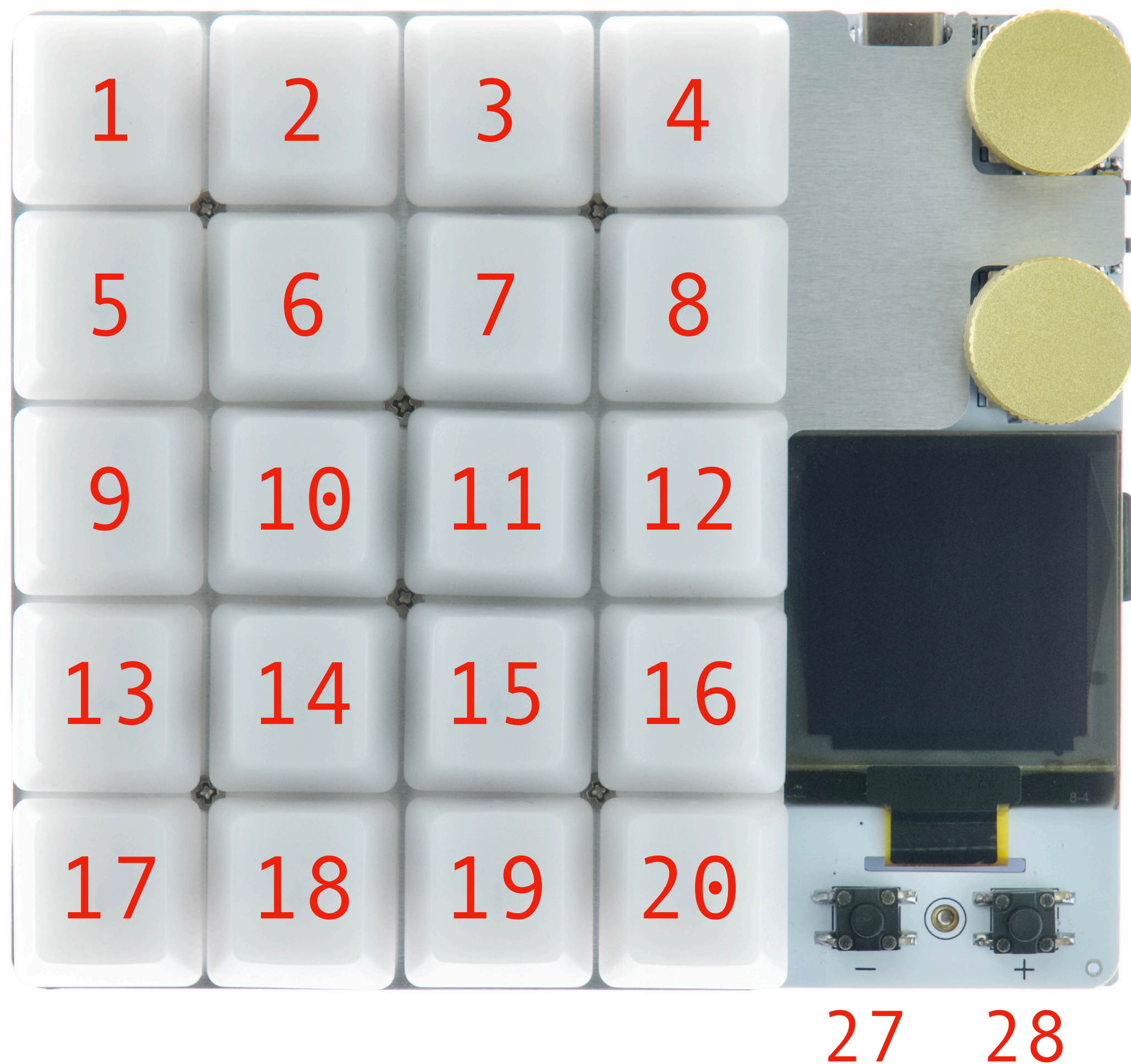
Randomisation
\$_RANDOM_MIN = 0 \$_RANDOM_MAX = 100 VAR \$eggs = \$_RANDOM_INT
Range is <b>inclusive</b>

Reserved Variables		
\$_RANDOM_MIN	\$_RANDOM_MAX	Read & Write
\$_RANDOM_INT		
\$_TIME_S		Read Only
\$_TIME_MS		
\$_READKEY		
\$_BLOCKING_READKEY		
\$_IS_NUMLOCK_ON		
\$_IS_CAPSLOCK_ON		
\$_IS_SCROLLLOCK_ON		
\$_THIS_KEYID		



# Key IDs: duckyPad Pro

- Each key on duckyPad has a **unique ID**
- Used for **reading button status** and changing RGB colour
- Key Test sample script provided



Rotary Encoder	Clockwise	Counter Clockwise	Press
Upper	21	22	23
Lower	24	25	26

Expansion Module	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8
Closest to duckyPad	37	38	39	40	41	42	43	44
2nd Closest	45	46	47	48	49	50	51	52
etc.								



# Key IDs: duckyPad (2020)



16 17