

duckyScript Cheatsheet: Key Combos

Latest Resources

- [duckyPad Homepage](#)
- [duckyScript Full Doc](#)
- [Autoswitcher](#)
- [Syntax Highlighter](#)
- [Discord Chatroom](#)

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**

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

Typing

STRING <code>text</code>	Type text AS-IS
STRINGLN <code>text</code>	Type text AS-IS Press ENTER at end
STRING_BLOCK END_STRING	Type text block AS-IS No new lines
STRINGLN_BLOCK END_STRINGLN	Same as above Press ENTER after each line

Timing

DELAY <code>n</code>	Wait <code>n</code> Milliseconds
DEFAULTDELAY <code>n</code>	How long to wait between each non-letter keypresses
DEFAULTCHARDELAY <code>n</code>	How long to wait between typing each letter
CHARJITTER <code>n</code>	Additional random delay between 0 and <code>n</code> ms after typing each letter

Mouse

LMOUSE MMOUSE RMOUSE FMOUSE BMOUSE	Left / Middle / Right Forward / Backward Mouse Button Click
MOUSE_MOVE <code>x y</code>	Move Mouse in Pixels <code>x</code> : + Right - Left <code>y</code> : + Up - Down
MOUSE_SCROLL <code>h v</code>	Scroll horizontal <code>h</code> lines vertical <code>v</code> lines

OLED

OLED_CLEAR	Clear Screen
OLED_CURSOR <code>x y</code>	Set Cursor <code>x y</code> : 0 to 127 (0,0) = Top Left
OLED_PRINT <code>text</code>	Print Text at Current Cursor
OLED_CPRINT <code>text</code>	Print Text Center-Aligned
OLED_CIRCLE <code>x y radius fill</code>	<code>x y</code> : Position <code>radius</code> : In Pixels <code>fill</code> : 0 or 1
OLED_LINE <code>x1 y1 x2 y2</code>	<code>x1 y1</code> : Start <code>x2 y2</code> : End
OLED_RECT <code>x1 y1 x2 y2 fill</code>	<code>x1 y1</code> : Start <code>x2 y2</code> : End <code>fill</code> : 0 or 1
OLED_UPDATE	Commit Changes to Screen
OLED_RESTORE	Show Default Screen

RGB Backlight

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

Pressing Keys

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

Profile Switching

PREV_PROFILE	
NEXT_PROFILE	
GOTO_PROFILE <code>name</code>	Case sensitive
Also check out duckyPad Autoswitcher!	

Miscellaneous

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

Comments

//
REM_BLOCK END_REM

Constants	
DEFINE	Define a Constant
Replaced AS-IS during preprocessing	
DEFINE EMAIL bob@me.com	

Variables	
VAR	Declare a 32-bit Variable
VAR foo = 10	
Default Mode: Signed	
Unsigned Mode: Set _UNSIGNED_MATH = 1	

Persistent Global Variables	
_GV0 to _GV31	
<ul style="list-style-type: none">Available across all profilesPersists over reboots	

Operators		
= Assign		&& Logical AND
+ Add	== Equal	Logical OR
- Subtract	!= Not equal	! Logical NOT
* Multiply	> Greater than	& Bitwise AND
/ Divide	< Less than	Bitwise OR
% Modulus	>= GTE	^ Bitwise XOR
** Exponent	<= LTE	<< Left Shift
		>> Right Shift
Augmented Assignments: +=, *=, etc		

IF Statement	
Code inside is executed If expression is non-zero	
IF expression code END_IF	
Additional Checks	ELSE IF ELSE

WHILE loop	
Code inside is repeated If expression is non-zero	
WHILE expression code END_WHILE	
Jump to start	CONTINUE
Exit immediately	LBREAK

Functions	
FUN my_func(args) code= END_FUN	
my_func() // call it	
Optional args & returns	
VARs declared inside have local scope	

Randomisation	
VAR foo = RANDINT(lower, upper)	
Range is inclusive	

Reading Keys	
Blocking	VAR key = _BLOCKING_READKEY Wait until any keypress, returns KeyID
Non-Blocking	VAR key = _READKEY 0 if no key pressed, KeyID otherwise
Bitfield	VAR key = _SW_BITFIELD Each bit: 1 = Pressed, 0 = Released

Real-time Clock	
Automatically set when using Autoswitcher	
Check _RTC_IS_VALID first. Do not proceed if 0.	
_RTC_YEAR	_RTC_MINUTE
_RTC_MONTH	_RTC_SECOND
_RTC_DAY	_RTC_WDAY
_RTC_HOUR	_RTC_YDAY

VAR Print & Formatting	
VAR foo = 10 STRING Value is \$foo	
Format Specifiers: Add immediately after var name	%d: Signed Decimal %u: Unsigned Decimal %x: Hex Lower %X: Hex Upper
Zero-Pad	%02d, %04x, etc
Space-Pad	%2d, %4x, etc

duckyPad Standard Library

Additional Helper Functions
Add `USE_STDLIB` to include
[Click for details](#)

User Headers

Click “Edit Header” Button
Write your own header
Add `USE_UH` to include

Built-in Functions

Low-level Operations
See [Full Guide](#) for details

<code>PEEK8(addr)</code>	<code>RANDINT(lower, upper)</code>
<code>POKE8(addr, value)</code>	<code>PUTS(value)</code>
<code>RANDCHR(value)</code>	<code>HIDTX(addr)</code>

Random Letters

Types a random character

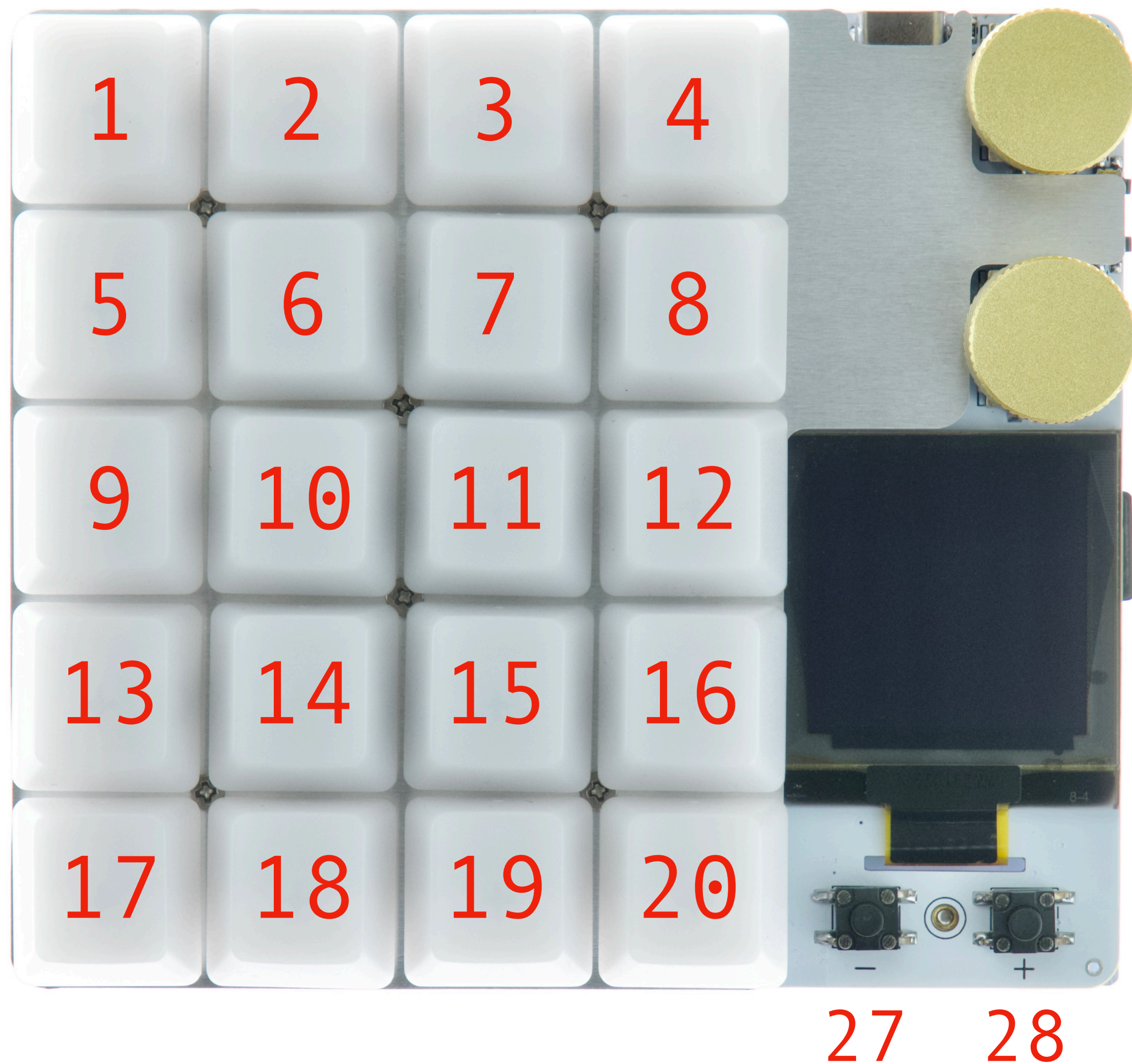
<code>RANDOM_LOWERCASE_LETTER</code>	<code>RANDOM_NUMBER</code>
<code>RANDOM_UPPERCASE_LETTER</code>	<code>RANDOM_SPECIAL</code>
<code>RANDOM_LETTER</code>	<code>RANDOM_CHAR</code>

Reserved Variables

<code>_TIME_S</code>	R0	Elapsed time since power-on
<code>_TIME_MS</code>		
<code>_BLOCKING_READKEY</code>	R0	See “Reading Inputs”
<code>_READKEY</code>		
<code>_SW_BITFIELD</code>	R0	0: LED OFF 1: LED ON
<code>_IS_NUMLOCK_ON</code>		
<code>_IS_CAPSLOCK_ON</code>		
<code>_IS_SCROLLLOCK_ON</code>	RW	0: Enable 1: Disable
<code>_ALLOW_ABORT</code>		
<code>_DONT_REPEAT</code>	R0	See “Key ID”
<code>_THIS_KEYID</code>		
<code>_DP_MODEL</code>	R0	1: duckyPad, 2: duckyPad Pro
<code>_KEYPRESS_COUNT</code>	R0	
<code>_RTC_IS_VALID</code>	R0	Check this first before reading RTC
<code>_RTC_YEAR</code>		4 Digits, e.g. 2025
<code>_RTC_MONTH</code>		1 - 12
<code>_RTC_DAY</code>		1 - 31
<code>_RTC_HOUR</code>		0 - 23
<code>_RTC_MINUTE</code>		0 - 59
<code>_RTC_SECOND</code>		0 - 60
<code>_RTC_WDAY</code>		Day of Week, 0 = Sunday
<code>_RTC_YDAY</code>		Day of Year, 0 - 365
<code>_RTC_UTC_OFFSET</code>	RW	In Minutes
<code>_UNSIGNED_MATH</code>	RW	Math mode

Key IDs: duckyPad Pro

- Each key on duckyPad has a **unique ID**
- Used for **reading button status** and changing RGB colour



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)

