

# duckyScript Cheatsheet: Key Combos

11 Jan 2026

duckyPad Quick Ref 1/6

## 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 <b>No new lines</b>
STRINGLN_BLOCK END_STRINGLN	Same as above Press ENTER after <b>each line</b>

## Timing

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

## 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 <b>Center-Aligned</b>
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
<b>OLED_UPDATE</b>	<b>Commit Changes to Screen</b>
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 <a href="#">duckyPad Autoswitcher</a>	

## Miscellaneous

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

## Comments

//
REM_BLOCK END_REM

## Random Letters

Type a random character
RANDOM_LOWERCASE_LETTER
RANDOM_NUMBER
RANDOM_UPPERCASE_LETTER
RANDOM_SPECIAL
RANDOM_LETTER
RANDOM_CHAR

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

Variables	
VAR	Declare Signed 32-bit Variable
VAR foo = 10	

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

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

Operators (Unsigned)	
ULT(lhs, rhs)	UGTE(lhs, rhs)
ULTE(lhs, rhs)	UDIV(val, n)
UGT(lhs, rhs)	UMOD(val, n)
LSR(val, n)	

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

WHILE loop	
Code inside is repeated If <b>expression</b> is <b>non-zero</b>	
WHILE <b>expression</b> 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)	
Unsigned: RANDUINT(lower, upper)	
Range is <b>inclusive</b>	

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 <a href="#">Autoswitcher</a>	
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 <b>immediately</b> <b>after</b> 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

RANDCHR(value)

RANDINT(lower, upper)

PUTS(value)

HIDTX(addr)

Memory Access

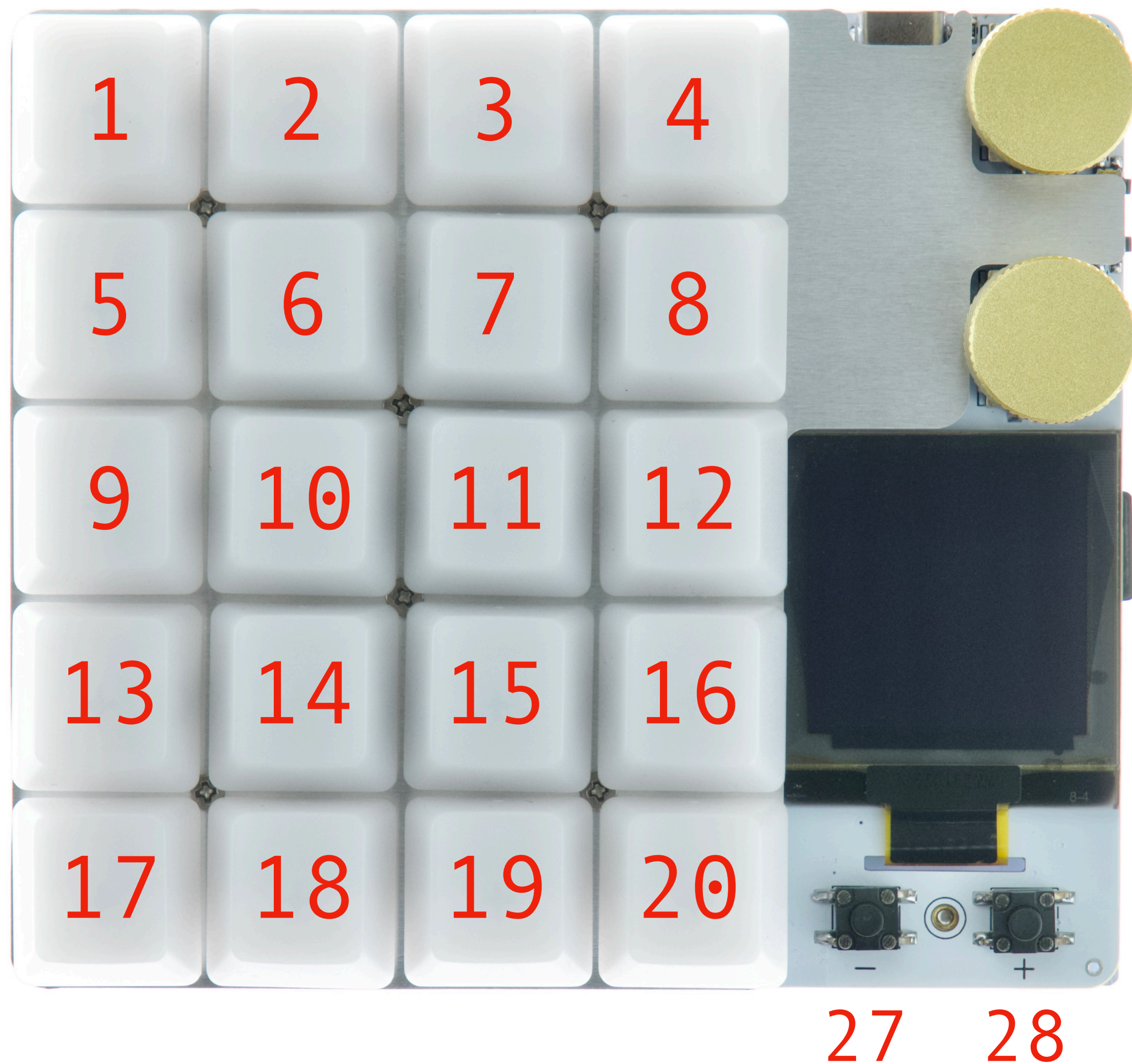
Read Signed	PEEK8(addr) PEEK16() PEEK32()
Read Unsigned	PEEKU8(addr) PEEKU16(addr)
Write	POKE8(addr, val) POKE16() POKE32()
For use in Scratch Memory Area <a href="#">Learn more about duckStack VM</a>	

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



# 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)

