## # CONSOLE SHORTCUTS

| | |
|---|---|
| F2 | RUN |
| F3 | LIST |
| F4 | EDIT |
| F10 | AUTOSAVE |
| F11 | XMODEM RECEIVE |
| F12 | XMODEM SEND |
| F1, F5–F9 | User-programmable with OPTION FNKEY |
| CTRL+C | Interrupt running program |

## # EDITOR SHORTCUTS

**FUNCTION KEYS**

| | |
|---|---|
| ESC | Exit editor (ask to save if modified) |
| F1 | Save and return to prompt |
| F2 | Save and run |
| F3 | Find text |
| SHIFT+F3 | Find next |
| F4 | Enter mark mode |
| F5 | Paste from clipboard |

**MARK MODE KEYS**

| | |
|---|---|
| ESC | Exit mark mode |
| F4 | Cut (copy + delete) |
| F5 | Copy |
| DEL | Delete marked text |

## # BASIC COMMANDS

**PROGRAMS, FILES AND DIRECTORIES**

| | |
|---|---|
| A: | Switch to flash storage |
| B: | Switch to SD-card storage |
| NEW | Clear memory |
| FILES | List files |
| LOAD filename$ | Load program |
| RUN or * | Start program |
| END | End program and return to console |
| LIST | Show program in memory |
| EDIT | Open built-in editor |
| SAVE «file.bas» | Save program to flash/SD |
| KILL «file.bas» | Delete file |
| MKDIR «name» | Create subdirectory |
| CHDIR «name» | Change into directory |
| RMDIR «name» | Remove directory |
| RENAME o$ AS n$ | Rename old file or dir o$ to new name n$ |

**OTHER**

| | |
|---|---|
| ‘ | Comment (to the end of the line) |
| /* */ | Multiline comment (must be the first non-space characters at the start of a line and have a space or end-of-line after them) |
| PRINT or ? | Write the following value to console |
| INPUT [«str»], v1 | Prompt user with optional string str, input is saved to variables v1, v2, etc. |
| PAUSE t | Delay for t number of milliseconds |
| INC v [,i] | Increments variable v by 1 or i faster than v=v+i |

## # VARIABLES

(Max var name length is 32 characters)

| | |
|---|---|
| name | Float type (default) |
| Name! | Double precision float type |
| name% | 64-bit signed integer type |
| name$ | String type |
| DIM name(s) = (c, c1) | Global array with size s and content c, c1, etc. |
| AS STRING | Set variable type or function return type |

**WITH OPTION EXPLICIT**

| | |
|---|---|
| DIM name | Global variable |
| LOCAL name | Local variable (in sub/func) |
| STATIC name | Like LOCAL but value persists between subroutine/function calls |
| CONST name | Immutable variable |

## # STRING SPECIAL CHARACTERS

(Requires OPTION ESCAPE at start of program)

| Char | Hex | Description |
|---|---|---|
| \a | 07 | Alert (Beep, Bell) |
| \b | 08 | Backspace |
| \e | 1B | Escape character |
| \f | 0C | Formfeed Page Break |
| \n | 0A | Newline (Line Feed) |
| \r | 0D | Carriage Return |
| \q | 22 | Quote symbol |
| \t | 09 | Horizontal Tab |
| \v | 0B | Vertical Tab |
| \\ | 5C | Backslash |

## # EXPRESSIONS AND OPERATORS

**ARITHMETIC OPERATORS**

| | |
|---|---|
| ^ | Exponentiation (e.g. b^2) |
| * | Multiplication |
| / \ MOD | Division, integer division, modulus |
| + - | Addition, subtraction |

**SHIFT OPERATORS**

| | |
|---|---|
| x << y | Returns x shifted by y bits to the left |
| x >> y | Returns x shifted by y bits to the right |

**LOGICAL OPERATORS**

| | |
|---|---|
| = <> | Equality, Inequality |
| < > | Less than, greater than |
| <= >= | Less than / greater than or equal to |
| AND OR | Conjunction, disjunction |
| XOR | Exclusive or |
| NOT | Invert logical value (e.g. NOT a = b) |
| INV | Bitwise inversion (e.g. a = INV b) |

## # CONTROL STRUCTURES

```
IF expression THEN statement [ELSE statement]
```

```
IF expression THEN
    <statements>
[ELSEIF expression THEN
    <statements>]
[ELSE
    <statements>]
ENDIF
```

```
DO
    <statements>
    [EXIT DO / CONTINUE DO]
LOOP
```

```
DO WHILE expression
    <statements>
    [EXIT DO / CONTINUE DO]
LOOP
```

```
DO
    <statements>
    [EXIT DO / CONTINUE DO]
LOOP UNTIL expression
```

```
FOR i = 1 TO 10
    <statements>
    [EXIT FOR / CONTINUE FOR]
NEXT i
```

## # SUBROUTINES AND FUNCTIONS

Subroutines act like commands and can take arguments. Functions act like subroutines but can also return values.

```
SUB MYSUB arg1, arg2$, arg3
    <statements>
END SUB

‘ Call subroutine, empty arguments allowed
MYSUB 23, , 55
```

```
FUNCTION FunctionName(arg1) AS FLOAT
    ‘ Return value by assigning to name of function
    FunctionName = arg1 + 0.5
END FUNCTION

‘ Call function
a = FunctionName()
```

## # GRAPHICS

**GENERAL**

| | |
|---|---|
| CLS [color] | Clear the screen |
| RGB(red, green, blue) | Generate 24 bit color number |
| COLOR fore [, back] | Set foreground and optional background color |
| FONT number, scaling | Sets the active font number and scale (1-15) |
| TEXT x, y, string$ | Display string starting at x,y (optional args alignment$ (L/C/R + T/M/B), font, scale (1-15), c (color), bc (background color) |

PM_QR
//v2//

**DRAWING**

| | |
|---|---|
| PIXEL x, y [, color] | Draw pixel of color at x,y |
| PIXEL(x, y) | Returns color of pixel at x,y |
| LINE x1, y1, x2, y2, [lw], c | Draw line from x1,y1 to x2,y2 of color c, with lw line width (lw only for vertical/ horizontal lines) |
| BOX x, y, w, h, [lw], c, fill | Draw box starting at x,y with w,h dimensions, and lw line width of color c |
| RBOX x, y, w, h, r, c, fill | Draw box with rounded corners of radius r |
| CIRCLE x, y, r, lw, a, c, fill | Draw circle centered on x,y with radius r, line width lw, and aspect ratio a (float 0-1) |
| ARC x, y, r1, [r2], a1, a2 [,c] | Draw arc centered on x,y with inner and outer radii r1 and r2, and start and end angles a1 and a2, 0deg at 12 o'clock |
| GUI BITMAP x,y,b,w,h,s,… | Draw bits in bitmap b (int or string) starting at x,y, with dimensions w,h (8x8 default), at scale s, with color c and background color bc |
| POLYGON n, x%(), y%(),… | Draw n number of polygons with x,y pairs in arrays x%() and y%() with optional border color bc and fill color fc |

**FRAMEBUFFERS**

This command can be used to avoid screen artifacts when updating SPI displays with moving elements.

| | |
|---|---|
| FRAMEBUFFER CREATE | Creates framebuffer «F» with RGB121 color space and resolution matching configured SPI display |
| FRAMEBUFFER LAYER | Creates framebuffer «L» with RGB121 color space and resolution matching configured SPI display |
| FRAMEBUFFER WRITE w | Specifies target for subsequent graphics commands, where w can be N, F or L with N being the display |
| FRAMEBUFFER CLOSE [w] | Closes a framebuffer and releases memory, which w can be F or L, and if omitted closes both |
| FRAMEBUFFER COPY f, t [, b] | Does full screen copy of one framebuffer to another, from f and to t can be N, F or L, N being the display, when copying to display parameter b enables second processor |
| FRAMEBUFFER WAIT | Pauses processing until display enters frame blanking |
| FRAMEBUFFER MERGE [c, m, u] | Copies contents of Layer buffer and Framebuffer to display, omitting all pixels of a particular color c (number 0-15), using mode m (B: second processor, R: continual update second processor, A: abort continual update), at update rate u in milliseconds, both Framebuffer and Layerbuffer must be created |
| FRAMEBUFFER SYNC | Waits for the latest update on the second processor to complete to allow drawing without tearing |

**COLOR NUMBERS**

Color numbers ranging from 0 to 15 used for certain commands.

| Number | Color | Number | Color |
|---|---|---|---|
| 0 | BLACK | 8 | RED |
| 1 | BLUE | 9 | MAGENTA |
| 2 | MYRTLE | 10 | RUST |
| 3 | COBALT | 11 | FUCHSIA |
| 4 | MIDGREEN | 12 | BROWN |
| 5 | CERULEAN | 13 | LILAC |
| 6 | GREEN | 14 | YELLOW |
| 7 | CYAN | 15 | WHITE |

**# FONTS**

In all fonts the back quote character (60 hex or 96 decimal) has been replaced with the degree symbol (°). Font #1 and #4 has extended graphics characters from CHR$(32) to CHR$(255) or 20 to FF in hex.

| Number | Size | Description |
|---|---|---|
| 1 | 8x12 | (Default) All 95 ASCII + 7F-FF hex |
| 2 | 12x20 | All 95 ASCII |
| 3 | 16x24 | All 95 ASCII |
| 4 | 10x16 | All 95 ASCII + 7F-FF hex |
| 5 | 24x32 | All 95 ASCII |
| 6 | 32x50 | 0-9 + some symbols |
| 7 | 6x8 | All 95 ASCII |
| 8 | 4x6 | All 95 ASCII |

**# AUDIO**

Sound can be generated or played from files of the supported sound file types FLAC, MOD, and WAV.

| | |
|---|---|
| PLAY t, f$ [, i] | Play an audio file with the file type t (WAV/FLAC/MODFILE), named f$, and call subroutine i once finished playing |
| PLAY TONE l, r, d, i | Generate sine waves with frequencies in Hz for left and right channels l and r, with duration d in milliseconds, and call subroutine l once finished playing |
| PLAY PAUSE | Temporarily pause current playing file or tone |
| PLAY RESUME | Resume playing file or tone that was paused |
| PLAY NEXT | Play next WAV/FLAC file in directory |
| PLAY PREVIOUS | Play previous WAV/FLAC file in directory |
| PLAY STOP | Terminate playing file or tone |
| PLAY volume l, r | Set volume (0-100) for left l and right r channels |

**# BUILT IN FUNCTIONS**

**STRINGS AND CHARACTERS**

| | |
|---|---|
| ASC(s$) | Returns ASCII code for first letter in s$ |
| EVAL(s$) | Evaluates s$ as a BASIC expression, and returns result |
| INSTR([st,] s$, p$, s) | Returns position where p$ occurs in $s, from position st (first character is position 1), returns 0 if not found, p$ is regex if size s is specified |
| LEN(s$) | Returns number of characters |
| LCASE$(s$) | Returns s$ in lower case |
| UCASE$(s$) | Returns s$ in upper case |
| LEFT$(s$, n) | Returns substring with n number of characters from beginning (left) of string s$ |
| RIGHT$(s$, n) | Returns substring with n number of characters from end (right) of string s$ |
| STR$(n) | Returns number n as string |
| STRING$(n, $s) | Returns string n characters long of the first character of $s, or $s can be replaced with int ASCII character code |

**NUMBERS AND MATH**

| | |
|---|---|
| ABS(n) | Returns absolute number n |
| CINT(n) | Returns n rounded to closest integer |
| FIX(n) | Returns n truncated to integer with no rounding |
| VAL(s$) | Returns numerical value of s$, invalid number returns 0 |

**OTHER**

| | |
|---|---|
| CHOICE(c, t, f) | If condition c is true do expression t, else if false to expression f, faster than «if then elseif» |
| INKEY$ | Returns and removes first character from console input buffer, or empty string |
| CWD$ | Returns current working directory |
| TIMER | Returns ms since last reset |
| CALL(f$, [,p, …]) | Efficiently call function named f$ with params p |
| SETPIN p, m, t [,o] | Configure external I/O pin p to mode m (OFF, AIN, DIN, FIN, PIN, CIN, DOUT), with option o. |
| PIN(p) | Returns value of pin p (DIN: 1/0, AIN: float, FIN: Hz, PIN: ms, CIN: count since reset) |
| PIN(p) = v | Set output for pin p to value v |

**# PINOUT AND PERIPHERALS**

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | USB → | | | | | | |
| PWM0A | COM1-TX | I2C-SDA | SPI-RX | GP0 | 1 | | 40 | VBUS | | | | |
| PWM0B | COM1-RX | I2C-SCL | | GP1 | 2 | | 39 | VSYS | | | | |
| | | | | **GND** | 3 | | 38 | **GND** | | | | |
| PWM1A | | I2C2-SDA | SPI-CLK | GP2 | 4 | | 37 | 3V3EN | | | | |
| PWM1B | | I2C2-SCL | SPI-TX | GP3 | 5 | | 36 | 3V3 | | | | |
| PWM2A | COM2-TX | I2C-SDA | SPI-RX | GP4 | 6 | | 35 | ADC VREF | | | | |
| PWM2B | COM2-RX | I2C-SCL | | GP5 | 7 | | 34 | GP28 SPI2-RX | I2C-SDA | COM1-TX | PWM6A | |
| | | | | **GND** | 8 | | 33 | **GND** | | | | |
| PWM3A | | I2C2-SDA | SPI-CLK | GP6 | 9 | | 32 | GP27 SPI2-TX | I2C2-SCL | | PWM5B | |
| PWM3B | | I2C2-SCL | SPI-TX | GP7 | 10 | | 31 | GP26 SPI2-CLK | I2C2-SDA | | PWM5A | |
| PWM4A | COM2-TX | I2C-SDA | SPI2-RX | GP8 | 11 | | 30 | RUN | | | | |
| PWM4B | COM2-RX | I2C-SCL | | GP9 | 12 | | 29 | GP22 | I2C2-SDA | | PWM3A | |
| | | | | **GND** | 13 | | 28 | **GND** | | | | |
| PWM5A | | I2C2-SDA | SPI2-CLK | GP10 | 14 | | 27 | GP21 | I2C-SCL | COM2-RX | PWM2B | |
| PWM5B | | I2C2-SCL | SPI2-TX | GP11 | 15 | | 26 | GP20 SPI-RX | I2C-SDA | COM2-TX | PWM2A | |
| PWM6A | COM1-TX | I2C-SDA | SPI2-RX | GP12 | 16 | | 25 | GP19 SPI-TX | I2C2-SCL | | PWM1B | |
| PWM6B | COM1-RX | I2C-SCL | | GP13 | 17 | | 24 | GP18 SPI-CLK | I2C2-SDA | | PWM1A | |
| | | | | **GND** | 18 | | 23 | **GND** | | | | |
| PWM7A | | I2C2-SDA | SPI2-CLK | GP14 | 19 | | 22 | GP17 | I2C-SCL | COM1-RX | PWM0B | |
| PWM7B | | I2C2-SCL | SPI2-TX | GP15 | 20 | | 21 | GP16 SPI-RX | I2C-SDA | COM1-TX | PWM0A | |

| | |
|---|---|
| PIN34 | ADC2 |
| PIN33 | AGND |
| PIN32 | ADC1 |
| PIN31 | ADC0 |