**MicroBASIC**

MicroBASIC is a simple tiny BASIC, borrowing elements from Tiny BASIC and Atom BASIC to give a useable BASIC with string options.

Arithmetic is 16 bit 2’s complement integer. There are 27 variables @A-Z, and a block of memory which is indexed from 0. The exact memory available depends on the memory requirements of the interpreter and Arduboy library.

54 bytes are required for the variables and there is a 33 byte string buffer

Terms

|  |  |  |
| --- | --- | --- |
| **Description** | **Example** | **Notes** |
| Variable access | X | Evaluates to the contents of X. |
| Array access | XX4 | Accesses the fourth 16 bit word of the memory pointed to by X. (e.g. offset 8) – this can be any term e.g. XXA is the Ath element. |
| Byte Access | ?x | Accesses the byte at the memory location specified in the following term. |
| Word Access | !x | Accesses the word at the memory location specified in the following term |
| 1’s complement | ~x | Invert all the bits in the term (1’s complement) |
| Constant | 42 | Evaluates to 42, possibly preceded by a minus sign. |
| String | “Hello world” | Copies the string into the buffer and returns a pointer to it. Must be <= 32 characters long. |
| Parenthesis | (4+2) | Evaluate the expression in the brackets. |
| Length function | LEN(X) | Length of ASCIIZ string pointed to by parameter |
| Absolute function | ABS(X) | Returns the absolute value of the parameter |
| Sign function | SGN(X) | Returns the sign value of the parameter |
| Random function | RND(X) | Returns a random number from 1 to X. If X is zero, it returns a random 16 bit value. |
| Memory function | MEM(X) | Allocates x bytes, if possible, and returns the lowest address. |
| String Conversion | STR(X) | Convert X to decimal and store in buffer, return buffer address. |

Binary Operators

All binary operators have no precedence, they are evaluated in strict left to right order.

|  |  |  |
| --- | --- | --- |
| **Description** | **Example** | **Notes** |
| + | 4+2 | Addition |
| - | 4-2 | Subtraction |
| \* | 4\*2 | Multiplication |
| / | 4/2 | Division |
| % | 4%2 | Modulus |
| & | 4&2 | Bitwise AND |
| | | 4|2 | Bitwise OR |
| ^ | 4^2 | Bitwise XOR |
| >= = < <= <> > | 4 > 2 | Comparison operators, return 0 or -1 |

Left Expressions Supported

|  |  |  |
| --- | --- | --- |
| Description | Example | Notes |
| A = | A = 42 | Simple assignment |
| AA<term> = | AA4 = 42 | Word array assignment, 42 to array item 4 (offset 8) |
| ?<term> = | ?(A+4) = 42 | Byte indirection |
| !<term> = | !(A+4) = 42 | Word indirection |
| $<term> = | $A = “xxx” | Copy string pointer of RHS to address of term (e.g. address in A) |

Non assignment instructions

IF <expr> [THEN] <instructions to end of line>

GOSUB <expr>

GOTO <expr>

RETURN

REPEAT … UNTIL <expr>

Instructions are : separated.

Mechanism for calling routines.

**Additions ?**

Possibly FOR/NEXT if not too large.

Possibly REPEAT/UNTIL

Term Keywords

? ! ~ LEN( ABS( SGN( RND( STR( MEM(

Expression Keywords

+ - \* / % & | ^ = <> > < >= <=

Other Keywords

$ IF THEN GOSUB GOTO RETURN REPEAT UNTIL

Reserved

FOR NEXT PROC TO