AVR TINYBASIC PRO V2.0

BASIC REFERENCE

FOR ATMEGA1284P MICROCONTROLLERS

DOMINIK KRISTEN
JUNE 18, 2026

Table of contents

VARIABLES AND DATA TYPES 3
INTEGER NUMBERS 4
HEXADECIMAL CONSTANTS 4
STRINGS 4
EXPRESSIONS 5
RELATIONAL OPERATORS 5
PROGRAM CONTROL 7
INPUT / OUTPUT 9
GRAPHICS 10
SOUND 12
DIGITAL & ANALOG I/O 13
SD CARD COMMANDS 14
INTERNAL EEPROM 16
EXTERNAL I^2C EEPROM 17
FUNCTIONS 18
OPERATORS, VARIABLES AND NUMBERS 20

VARIABLES AND DATA TYPES

TinyBasic uses simple integer variables identified by a single letter of the alphabet.

Valid variable names are:

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

Variable names are case-insensitive. Lowercase letters entered by the user are automatically converted to uppercase.

Examples:

```
A=10
B=A+5
Z=1234
```

Only single-letter variable names are supported. Names such as COUNT, INDEX or VALUE are not valid variable identifiers.

Examples:

```
A=100    Valid
X=25     Valid
COUNT=10 Invalid
```

INTEGER NUMBERS

All numeric values are stored as signed 16-bit integers.

Valid range:

-32768 to 32767

Examples:

```
A=100
B=-25
C=32767
```

Values outside this range may overflow and produce unexpected results.

HEXADECIMAL CONSTANTS

Hexadecimal numbers are prefixed with a dollar sign ($).

Examples:

```
A=$FF
B=$100
C=$1234
```

The following digits may be used:

```
0 1 2 3 4 5 6 7 8 9 A B C D E F
```

STRINGS

Text strings are enclosed in quotation marks.

Examples:

```
PRINT "HELLO WORLD"
PRINT "TINYBASIC"
```

Strings may be printed and used in comments, but cannot be assigned to variables. TinyBasic supports numeric variables only.

EXPRESSIONS

TinyBasic evaluates expressions from left to right using standard arithmetic precedence.

Supported operators:

```
+   Addition
-   Subtraction
*   Multiplication
/   Division
%   Modulo (remainder)
```

Examples:

```
A=10+5
B=A*2
C=100/4
D=11%3
```

RELATIONAL OPERATORS

The following comparison operators are supported:

```
=    Equal to
!=   Not equal to
<>   Not equal to
>    Greater than
<    Less than
>=   Greater than or equal to
<=   Less than or equal to
```

Examples:

```
IF A=10 PRINT "OK"

IF A<>B PRINT "DIFFERENT"

IF X>=100 GOTO 1000
```

SYSTEM COMMANDS

CLS

Clears the screen and homes the cursor.

Example:

```
10 CLS
20 PRINT "HELLO"
```

MEM

Displays available BASIC program memory.

RAM

Displays RAM information.

Example:

```
RAM 6500
```

EEP

Displays internal EEPROM information.

WAIT milliseconds

Pauses program execution.

Example:

```
WAIT 1000
```

RSEED value

Initializes the random number generator.

Example:

```
RSEED 1234
```

EXIT

Leaves TinyBasic and returns to the system monitor.

PROGRAM CONTROL

==========

LIST [line, nlines]

Displays the current BASIC program or specific lines

Example:

```
  LIST
 LIST 100
 LIST 100,10
```

NEW

Deletes the current BASIC program from memory.

RUN

Starts execution of the current BASIC program.

BREAK

Stops execution of a running program.

END

Terminates program execution.

GOTO line

Branches to the specified line number.

Example:

```
  10 GOTO 100
```

GOSUB line

Calls a subroutine.

Example:

```
  100 GOSUB 1000
```

RETURN

Returns from a subroutine.

FOR variable=start **TO** end [STEP value]

Begins a FOR/NEXT loop.

Example:

```
10 FOR I=1 TO 10
20 PRINT I
30 NEXT I
```

NEXT variable

Advances a FOR loop.

Example:

```
NEXT I
```

IF condition statement

Conditional execution.

Example:

```
10 IF A=10 PRINT "READY"
```

LET variable=expression

Assigns a value to a variable.

Example:

```
10 LET A=100
```

REM (') text

Program comment.

Example:

```
10 REM DEMONSTRATION PROGRAM
10 'TEST PROGRAM
```

INPUT / OUTPUT

==========

INPUT variable

Reads a value from the keyboard.

Example:

```
10 INPUT A
20 PRINT A
```

PRINT (?) expression

Displays text or numeric values.

Example:

```
10 PRINT "HELLO WORLD"
20 PRINT A
```

INKEY

Reads a key without requiring ENTER.

Example:

```
10 A=INKEY
20 IF A<>0 PRINT A
```

GRAPHICS

==========

LOCATE X,Y

Positions the text cursor.

Example:

```
LOCATE 0,0
PRINT "HELLO"
```

PLOT X,Y[,C]

Plots a pixel at X,Y.

C is optional and specifies the pixel state.

Example:

```
PLOT 100,50
PLOT 120,50,0
```

LINE X1,Y1,X2,Y2[,C]

Draws a line.

Example:

```
LINE 0,0,100,100
```

RECT X,Y,W,H[,C]

Draws a rectangle.

Example:

```
RECT 20,20,80,40
```

CIRCLE X,Y,R[,C]

Draws a circle.

Example:

```
CIRCLE 120,100,25
```

CHAR X,Y,CHARACTER[,C]

Displays a character at a graphics position.

Example:

```
CHAR 10,10,65
```

GETPIX(X,Y)

Returns the state of a pixel.

Example:

```
A=GETPIX(100,50)
```

SOUND

==========

PLAY toneheight, tonelength

Starts tone or melody playback.

Example:

```
PLAY 100,200
```

STOP

Stops sound playback.

DIGITAL & ANALOG I/O

==========

OUT pin,status

Sets a digital output HIGH or LOW (1 or 0).

Example:

```
OUT 4,1
```

AOUT pin,value

Outputs an analog/PWM value.

Example:

```
AOUT 0,128
```

SD CARD COMMANDS

==========

DIR

Displays directory contents.

LOAD filename

Loads a BASIC program from SD card.

Example:

```
  LOAD TEST.BAS
 LOAD POKER
```

SAVE filename

Saves the current BASIC program.

Example:

```
  SAVE TEST
```

DEL filename

Deletes a file.

Example:

```
  DEL TEST
```

MD dirname

Creates a directory.

Example:

```
  MD GAMES
```

RD dirname

Removes a directory.

Example:

```
  RD GAMES
```

CD dirname

Changes the current directory.

Example:

```
  CD GAMES
```

CHKDSK

Checks the SD card file system.

INTERNAL EEPROM

==========

ELIST

Lists BASIC programs stored in EEPROM.

ELOAD

Loads a BASIC program from EEPROM.

ESAVE

Saves the current BASIC program to EEPROM.

EFORMAT

Formats the EEPROM.

EPOKE address,value

Writes a byte to EEPROM.

Example:

```
EPOKE 100,255
```

EPEEK(address)

Reads a byte from EEPROM.

Example:

```
PRINT EPEEK(100)
```

EDUMP [address]

Displays EEPROM contents.

EXTERNAL I²C EEPROM

==========

BLIST

Lists BASIC programs stored in external EEPROM.

BLOAD

Loads a BASIC program from external EEPROM.

BSAVE

Saves a BASIC program to external EEPROM.

BFORMAT

Formats the external EEPROM.

BPOKE address,value

Writes a byte to external EEPROM.

Example:

```
BPOKE 100,255
```

BPEEK(address)

Reads a byte from external EEPROM.

Example:

```
PRINT BPEEK(100)
```

BDUMP [address]

Displays external EEPROM contents.

FUNCTIONS

==========

PEEK(address)

Reads a byte from memory.

Example:

```
PRINT PEEK(1024)
```

ABS(value)

Returns the absolute value.

Example:

```
PRINT ABS(-10)
```

ADC(channel)

Reads an analog input.

Example:

```
PRINT ADC(0)
```

DIN(pin)

Reads a digital input.

Example:

```
PRINT DIN(2)
```

RND(max)

Returns a random number.

Example:

```
PRINT RND(100)
```

SQR(value)

Returns the square root.

Example:

```
PRINT SQR(144)
```

CHR$(value)

Returns the ASCII character corresponding to the numeric value n.

Example:

```
PRINT CHR$(144)
```

OPERATORS, VARIABLES AND NUMBERS

Arithmetic:

```
+  Addition
-  Subtraction
*  Multiplication
/  Division
%  Modulo
```

Relational:

```
=   Equal
!=  Not Equal
<>  Not Equal
>   Greater Than
<   Less Than
>=  Greater Or Equal
<=  Less Or Equal
```

VARIABLES

Variables are named A through Z.

Examples:

```
A=10
B=A+5
PRINT B
```

HEXADECIMAL CONSTANTS

Hexadecimal numbers begin with '$'.

Examples:

```
A=$FF
B=$1234
PRINT A
```