# Introduction to WhippleWay AVR Tiny BASIC Version 1.0

Dick Whipple (dickwhipple@whipplewy.com)

AVR Tiny BASIC was written by me and is based on the design originally suggested in the People's Computer Company magazine (Vol 3 No 4 Nov 1975) and published by John Arnold and me in Dr. Dobb's Journal (Vol 1 No 1 Jan 1976). I made some minor corrections and changes but it closely resembles what I published then.

I am publishing it for demonstration purposes only. I make no claims as to its worthiness for any purpose.

A program line has the following format: l*ine-number statement {:statement* . . .} CRLF

where *line-number* - An integer between 1 and 65,535
*statement* – See below
"{ }" – Optional statements
":" – Continuation character
CRLF – Carriage return (13) and line feed (10)

Valid AVR Tiny BASIC statements:

1. LET *variable* = *expression* – Assigns the computed value of expression to designated variable.
   Example: LET W = 2 * (A - 2)
   Note: LET is optional; that is, "A=0" is the same as "LET A=0".
2. INPUT *variabl*e list – Accepts numeric input from the keyboard and stores it in a designated variable(s).
   Example: INPUT A, B
3. PRINT – Outputs *string* (text enclose in quotation marks) and/or computed *expression* value. Items separated by semicolon (single space) or comma (zone spacing). A semicolon or colon at the end of the line inhibits new line.
   Example: PRINT "The result is "; 3*X
4. IF THEN *expression* – Evaluated the *relational-expression* and, if true, executes the line-number given by evaluation of *expression*[1]. If false, execution continues at the next line.
   Example: IF A<=10 THEN 200
   IF *relational-expression statement*{:*statement . . .}* – This alternate, non-standard version of the IF statement was developed to permit a more compact form. "THEN" is replaced by any executable statement.
   Example: IF A=0 LET A=1:GOTO 20
5. GOTO *expression* – Branches to the line-number given by evaluation of *expression*.
   Example: GOTO 100
   A=100:GOTO 2*A
6. GOSUB line-number – Branches to a subroutine at the line-number given by evaluation of *expression*.
7. RETURN –Returns from a subroutine to the instruction following the corresponding GOSUB.
8. OUT (*expression*1*, expression*2) – Outputs to standard Arduino pin number given by *expression*1 either (1) a "low" if *expression*2=0 or (2) a "high" if *expression*2 <>0.
9. END – When encountered, returns execution to the command mode

where variable - Letters A to Z represent signed integer variables.
expression – standard algebraic expression employing integer arithmetic over the range (-32,768 to +32,767) recognizing these operators:
"-" negation
"+" addition
"-" subtraction
"*" multiplication

[1] The "calculated" GOTO is not standard BASIC, but provides some interesting programming possibilities.

"/" division

The order of operations is as follows:

negation
multiplication, division
addition, subtraction

relational-expression – Takes the form "expression1 relational-operator expression 2" where valid relational-operators are as follows:

< less than
> greater than
= equal to
<= less than or equal to
>= greater than or equal to
<> not equal to

10.

AVR Tiny BASIC commands:

1. NEW – Clears memory so that a new Tiny BASIC program can be entered
2. RUN – Executes the Tiny BASIC program in memory
3. LIST – Lists the Tiny BASIC program in memory.

AVR Tiny BASIC functions:

1. RND(*expression*) – For *expression*<>0, returns a randomized value between 0 and 10,000. For *expression*=0, returns previous randomized value.
   Example: B=RND(1)

Custom AVR Tiny BASIC Statements

1. PWM(expression) – Outputs pulse width modulated waveform to pin 9 of SparkFun Redboard where n = percent duty cycle
   Example: PWM(75) produces a 75% duty cycle waveform on pin 9
2. OUT(expression1, expression2) - Outputs a low (expression2 = 0) or high (expresssion2 <> 0) to pin number given by expression1 of SparkFun Redboard
   Example: OUT(3,1) Outputs a high to pin 3
3. DLY(expression) - Introduces a delay of expression in milliseconds
   Example: DLY(1000) introduces 1 second time delay (1000 milliseconds)
4. PUT(expression1, expression2, expression3) - Writes expression3 (the two byte data) to EEPROM at page given by expression1 and location given by expression2
   Example: PUT(10, 0, 1250) Writes the data value 1250 to EEPROM page 10 location 0

Custom AVR Tiny BASIC Functions

1. ADC(*expression*) – Returns the raw A/D value on the A/D channel given by *expression*.
   Example: PRINT ADC(3) prints the raw A/D value on pin A3
2. PIN(expression) – Returns the status (low=0 or high=1) of standard Arduino pin number given by *expression*.
   Example: G=PIN(6) Set G to either 0 or 1 depending on whether pin 6 is low or high, respectively

3. GET(expression1, expression2) – Returns the two byte data value on EEPROM at page given by expression1 and location given by expression2

Other AVR Tiny BASIC operational information:

- To interrupt an executing program, click “STOP”, “EXAMINE” address 000:003, and click “RUN”. This performs a “Warm Boot” leaving the program undisturbed.
- Command mode prompt is ">".
- From the command mode, the user enters a line of text terminated with ENTER key. Based on the text, one of the following happens:
    - If no line number present – Attempts to directly execute a command or statement.
    - If line number present in range of 1 to 65535
        - If line number not found in program
            - Line inserted in numerical order
            - Returns to command mode
        - If line number found in program
            - If entered line consists of line number only
                - Line is deleted
                - Returns to command mode
            - If entered line consists of line number and statement(s)
                - Line is deleted
                - Entered line inserted in numerical order
                - Returns to command mode
- Run mode – After executing the RUN command, the AVR Tiny BASIC program executes beginning at the first line.
- Tiny BASIC is <u>not</u> case sensitive; i.e., let a=0 is the same as LET A=0.
- Error codes
    - 100 Syntax error
    - 101 Statement not allowed in command mode
    - 102 Command not allowed in RUN mode
    - 103 Maximum number of subroutines exceeded
    - 104 RETURN without corresponding GOSUB
    - 105 Line number not found
    - 106 Expected line number
    - 107 Numeric overflow in expression
    - 108 Arithmetic divide by zero
    - 109 Expression too complex
    - 110 System error: Please report.
    - 111 Print buffer size exceeded
    - 112 Unpaired parentheses
    - 113 Relational expression incorrectly formed
    - 114 Expected variables
    - 115 Expected end-of-line
    - 116 Expected numeric value
    - 117 Out of memory; program size exceeded
- To save a BASIC program, enter the LIST command. Copy and paste lines into a text editor like Notepad++ (no special formatting. Save it.
- To load a BASIC program, enter “NEW” then use text transfer to send the saved text file to AVR Tiny BASIC line input. Use 2 ms character delay and 50 ms end-of-line delay.

# Introduction to WhippleWay AVR Tiny BASIC Version 1.0

Dick Whipple (dickwhipple@whipplewy.com)