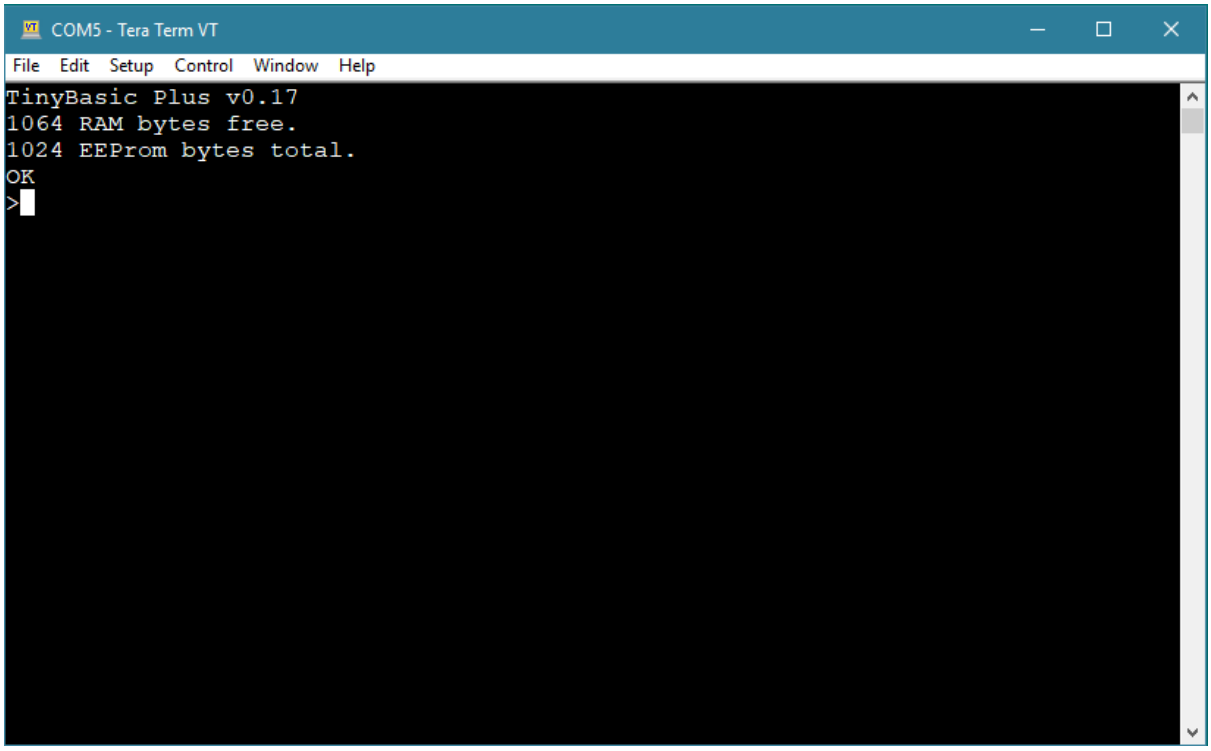
TinyBasicPlus 0.17

Description

TinyBasicPlus is a TinyBasic interpreter running on and Arduino UNO/NANO and can be used for small automation tasks or simple robotics (SUMOBOT).

The compiled version is around 11k in size (depending the compiler) and leaves roughly 1kbyte free for programs.



TinyBasic is a very simple basic, it has only one variable type: signed integer (-32768 to +32768). There is no floating point variable or arithmetic, no string variable and string manipulation. TinyBasic uses line numbers (like all classic computers from the 80's).

The commandset is extended with IO control commands (reading and writing Arduino pins, both analog and digital) and controlling RC servo's (2x).

The user interface is through serial port (USB) using a terminal program like Tera Term or Putty at 9600 baud, through the same connection Arduino uses to program the software.

There is no build in editor. Retyping the whole line is typical way of editing.

An alternative is edit the program in Geany, saving it, and sending it to the board using

CAT "%F" > /dev/ttyACM0 (for UNO)

CAT "%F" > /dev/ttyUSB0 (for NANO)

Specific features of this version

TinyBasic stores programs in ASCII format in RAM. That implies that every character you type uses up one byte in RAM. There is only 1k of RAM free in the Arduino UNO/NANO. To allow more functionality, Basic commands can be abbreviated in this version of TinyBasic. Abbreviation is achieved using the period "." Character.

Since the command parser uses a pre-defined order, some commands can be abbreviated shorter than others.

Example:

"NEXT" can be abbreviated as "N."
"NEW" cannot abbreviated at all, since it shares the N and E with NEXT, and the W is the first unique character.

The Arduino has 1 kbyte EEPROM that can be used to store program non volatile. Through use of ELOAD and ESAVE programs can be downloaded from, and stored to this memory. Its is possible to compile TinyBasic sucht that automatic execution of EEPROM stored programs is initiated at power up. This is currently not configurable through commandline.

Command

This chapter describes commands integrated in interpreter version 0.17 and some examples
The commands are listed in alphabetical order

Command	Explanation	Example
'	identifies a remark (see REM)	10 ' THIS IS A REMARK
>= <> > = <= < !=	qualifiers in an IF statement	10 IF A<=20 GOTO 60
?	short for PRINT	10 ? "Hello World"
ABS	returns absolute value	10 X = ABS(-23)
AREAD	analog read Arduino pin and return value (0-1023). Forces pin to ADC input.	10 Y = AREAD(A0) *read pin A0, return value in Y*
AWRITE	analog write (PWM) to the 6 Arduino pins that can support this. Servo pins excluded. Forces pin to PWM.	10 AWRITE X,Y *'pin,value*
DELAY	execution delay in miliseconds	10 DELAY 200 *= 200 miliseconds delay*
DREAD	read digital pin. Value is 0 or 1. forces pin to INPUT.	10 X=DREAD(12) ' reads pin 12, boolean value in X
DWRITE	write to digital pin. value 0 or 1. forces pin to OUTPUT	10 DWRITE 8,1 'write logic "1"to pin 8
ECHAIN	loads program from EEPROM, and executes it. EEPROM programs are non volatile.	-
EFORMAT	format EEPROM	-

ELIST	lists program in EEPROM without corruption of program in RAM.	-
ELOAD	loads program from EEPROM into RAM.	-
END	stops basic program and returns to command line.	-
ESAVE	save the current program in RAM to EEPROM.	-
FOR	controlled loop, in combination with TO and NEXT	10 FOR I=1 TO 10 20 PRINT I 30 NEXT I
GOSUB	call to subroutine. Used in combination with RETURN	10 GOSUB 80 20 'rest of program 80 PRINT "HI" 90 RETURN
GOTO	direct jump to line number	10 GOTO 80 20 'rest of program 80 PRINT "HI" 90 GOTO 20
HELP	lists available commands in TinyBasicPlus	
HIGH -or- HI	implementation of "true" or logic "1"	10 DWRITE 12,HIGH
IF	conditional branch. Used in combination with GOTO or GOSUB	10 IF X=23 GOTO 20 20 30 IF A=0 GOSUB 90
INPUT	user input via the terminal only accepts numeric values	10 PRINT "ENTER A NUMBER"; 20 INPUT A 30 IF A=5 GOTO 45
LET	obsolete command to assign a value to a variable. Can be omitted.	10 LET A=B+45 *identical to* 10 A=B+45
LIST	shows the basic program on the terminal. Accepts a start line number for the listing.	LIST LIST 23 *lists from line 23 and further*
LOW LO	implementation of "false" or logic "0"	10 IF X=LO GOTO 40
MAX	compares 2 values and stores the highest in variable1	10 A=1:B=17 20 MAX A,B *A will change to 17 since B is higher.* 30 MAX A,123 *A will be 123 since higher.*
MEM	shows available free memory in EEPROM and RAM	-

MILS	shows elapsed time since last MILS in miliseconds	10 MILS A 20 PRINT "HELLO" 30 MILS A 40 PRINT A *shows how long it took to print "HELLO"*
MIN	compares 2 values and stores the lowest in variable1	10 A=100:B=17 20 MIN A,B *A will change to 17 since B is higher.*
NEW	clear program memory (RAM)	-
NEXT	end of controlled loop (see FOR).	
NOSERVO	Turns off the SERVO pulses at indicated pin, and frees up pin for normal use.	10 SERVO 9,128 20 'some program 30 NOSERVO 9 40 DWRITE 9,HI
PEEK	reads value from memory (register file of program memory)	10 PRINT PEEK(24)
POKE	writes value to memory (untested)	10 POKE 23,116
PRINT	prints a value or string to the terminal. Multiple values can be printed, separated by a comma ","	10 PRINT "WEIGHT" , X , Y , "KILO"
REM	comment in program. This line is skipped during execution	10 REM THIS IS A TEST 20 ECHAIN 30 END
RETURN	return from a subroutine. See GOSUB.	
RND	returns a random number in the range of 0 up to (excluding) the value given	10 PRINT RND(2) prints zero's and one's (2 is excluded).
RSEED	seed value for the random generator	
RUN	execute the program in RAM	-
SERVO	starts outputting a 20mSec RC servo frame at indicated pin, and set the pulsewidth such an angle between 0 and 180 degrees. (angles may vary between servo's)	10 SERVO 10,90 controls a servo connected to pin 10 to the mid position (90 degrees).
STEP	Part of the FOR-NEXT loop, determines the step size.	10 FOR I=1 TO 20 STEP 3 will step I through 1,4,7,10,13,16,19

TO	determines the end of the FOR-NEXT loop. See FOR	
VARS	list all 26 variables and their current value on the terminal.	-