TinyBasic Plus

A C implementation of Tiny Basic, with a focus on support for Arduino. It was originally written by Gordon Brandly in the form of "68000 Tiny Basic", and then ported to C by Michael Field as "Arduino Basic", though still called "Tiny Basic" in the source files.

TinyBasic Plus is an extension and modification upon the original "Tiny Basic" by adding support for a few more devices, configurable at build time. It is designed for use on the Arduino, although builds will soon be easily possible for other platforms through command line makefiles. Provided is a makefile that builds for unix-ey type OSes. It has only been tested for Darwin (OS X).

Features added include support for fileio (SD Library), autorunning a program from the SD card, smaller footprint (PROGMEM), support for pin data IO, and support for the on-chip EEProm storage for your program.

Full list of supported statements and functions

System

- BYE - *exits Basic, soft reboot on Arduino*
- END - *stops execution from the program, also "STOP"*
- MEM - *displays memory usage statistics*

- NEW - *clears the current program*
- RUN - *executes the current program*

File IO/SD Card

- FILES - *lists the files on the SD card*
- LOAD filename.bas - *loads a file from the SD card*
- CHAIN filename.bas - *equivalent of: new, load filename.bas, run*
- SAVE filename.bas - *saves the current program to the SD card, overwriting*

EEProm - nonvolatile on-chip storage

- EFORMAT - clears the EEProm memory
- ELOAD - load the program in from EEProm
- ESAVE - save the current program to the EEProm
- ELIST - print out the contents of EEProm
- ECHAIN - load the program from EEProm and run it

IO, Documentation

- INPUT variable - *let the user input an expression (number or variable name*
- PEEK(address) - *get a value in memory* (unimplemented)
- POKE address - *set a value in memory* (unimplemented)
- PRINT expression - *print out the expression, also* "?"
- REM stuff - *remark/comment, also* "'"

Expressions, Math

- A=V, LET A=V - *assign value to a variable*
- +, -, *, / - *Math*
- <,<=,=,<>,!=,>=,> - *Comparisons*
- ABS(expression) - *returns the absolute value of the expression*
- RSEED(v) - *sets the random seed to v*
- RND(m) - *returns a random number from 0 to m*

Control

- IF expression statement - *perform statement if expression is true*
- FOR variable = start TO end - *start for block*

- FOR variable = start TO end STEP value - *start for block with step*
- NEXT - *end of for block*
- GOTO linenumber - *continue execution at this line number*
- GOSUB linenumber - *call a subroutine at this line number*
- RETURN - *return from a subroutine*

Pin IO

README.md

- DELAY timems - wait (in milliseconds)
- DWRITE pin,value - *set pin with a value (HIGH,HI,LOW,LO)*
- AWRITE pin,value - *set pin with analog value (pwm) 0..255*
- DREAD(pin) - *get the value of the pin*
- AREAD(analogPin) - *get the value of the analog pin*

NOTE: "PINMODE" command removed as of version 0.11

Sound - Piezo wired with red/+ on pin 5 and black/- to ground

- TONE freq,timems - play "freq" for "timems" milleseconds (1000 = 1 second)
- TONEW freq,timems - same as above, but also waits for it to finish
- NOTONE - stop playback of all playing tones

NOTE: TONE commands are by default disabled

Example programs

Here are a few example programs to get you started...

User Input

Let a user enter a new value for a variable, enter a number like '33' or '42', or a varaible like 'b'.

```
10 A=0
15 B=999
20 PRINT "A is ", A
30 PRINT "Enter a new value ";
40 INPUT A
50 PRINT "A is now ", A
```

Blink

hook up an LED between pin 3 and ground

```
10 FOR A=0 TO 10
20 DWRITE 3, HIGH
30 DELAY 250
40 DWRITE 3, LOW
50 DELAY 250
60 NEXT A
```

Fade

hook up an LED between pin 3 and ground

```
10 FOR A=0 TO 10
20 FOR B=0 TO 255
30 AWRITE 3, B
40 DELAY 10
50 NEXT B
60 FOR B=255 TO 0 STEP -1
70 AWRITE 3, B
80 DELAY 10
90 NEXT B
100 NEXT A
```

LED KNOB

hook up a potentiometeter between analog 2 and ground, led from digital 3 and ground. If knob is at 0, it stops

```
10 A = AREAD( 2 )
20 PRINT A
30 B = A / 4
40 AWRITE 3, B
50 IF A == 0 GOTO 100
60 GOTO 10
100 PRINT "Done."
```

ECHAIN example

Write a small program, store it in EEPROM. We'll show that variables don't get erased when chaining happens

```
EFORMAT
10 A = A + 2
20 PRINT A
30 PRINT "From eeprom!"
40 IF A = 12 GOTO 100
50 PRINT "Shouldn't be here."
60 END
100 PRINT "Hi!"
```

Then store it in EEProm

```
ESAVE
```

Now, create a new program in main memory and run it

```
NEW
10 A = 10
20 PRINT A
30 PRINT "From RAM!"
40 ECHAIN
```

List both, and run

```
ELIST
LIST
RUN
```

Device Support

Current

- Arduino - ATMega 168 (~100 bytes available)
- Arduino - ATMega 368 (~1100 bytes available)
- SD cards (via SD Library, for FILES, LOAD, SAVE commands, uses 9k of ROM)
- EEProm (via EEProm Library, uses 500 bytes of ROM)

- Serial IO - command console

Future

- PS2 Keyboard for standalone use (maybe)
- Graphics support via common function names and ANSI/ReGIS escape codes

Known Quirks and Limitations

- If LOAD or SAVE are called, FILES fails subsequent listings
- SD cards are not hot-swappable. A reset is required between swaps.

Authors and Contributors

- Tiny Basic 68k - Gordon Brandly Project Page (via archive.org)
- Arduino Basic / Tiny Basic C - Michael Field Project Page
- Tiny Basic Plus - Scott Lawrence yorgle@gmail.com Github Page
- Jurg Wullschleger - Fix for unary operations and whitespace in expressions

Links

- Arduino Microcontroller

Licensing

Mike Field based his C port of Tiny Basic on the 68000 Tiny BASIC which carried the following license:

```
*****************************************************************
*                                                               *
*               Tiny BASIC for the Motorola MC68000             *
*                                                               *
* Derived from Palo Alto Tiny BASIC as published in the May 1976 *
* issue of Dr. Dobb's Journal.  Adapted to the 68000 by:        *
*       Gordon Brandly                                          *
*       12147 - 51 Street                                       *
*       Edmonton AB  T5W 3G8                                    *
*       Canada                                                  *
*       (updated mailing address for 1996)                      *
```

```
*                                                                   *
* This version is for MEX68KECB Educational Computer Board I/O.     *
*                                                                   *
*********************************************************************
*    Copyright (C) 1984 by Gordon Brandly. This program may be      *
*    freely distributed for personal use only. All commercial       *
*                      rights are reserved.                         *
*********************************************************************
```

However, Mike did not include a license of his own for his version of this. From discussions with him, I felt that the MIT license is the most applicable to his intent.

I am in the process of further determining what should be done wrt licensing further. This entire header will likely change with the next version 0.16, which will hopefully nail down the whole thing so we can get back to implementing features instead of licenses. Thank you for your time.

MIT License

Releases

2 tags

Packages

No packages published

Contributors 8

Languages

C++ 97.6% Makefile 2.1% C 0.3%