An "old-style" BASIC interpreter, written in standard C, with a focus on being easy to extend and port.

Introduction

This is a BASIC interpreter written from scratch in C. It runs on Windows, Linux, OSX and a number of microcontroller platforms. It aims to fully implement the old-style dialects of BASIC, meaning, line-number based code. It should be quite compatible with early 1980's interpreters, such as Microsoft BASIC (for 8080/8085/Z80, 6502, 6800 etc), EhBASIC, Commodore BASIC, and so on.

Most of the common BASIC commands are supported:

- CLEAR, CLR
- CONT
- DELETE filename
- DIR
- LIST linenr[-linenr]
- LOAD filename
- NEW
- RUN [linenr]
- SAVE filename

Most of the common BASIC statements (instructions) are supported:

- CLS
- COLOR
- DATA
- DEF
- DIM variable "(" expression ")"
- END
- FOR numeric_variable '=' numeric_expression TO numeric_expression [STEP number]
- GET variable
- GOTO expression
- GOSUB expression
- IF relation-expression THEN statement [ELSE statement]
- INPUT ["prompt";] variable-list
- LET variable = expression
- LOCATE row,col
- NEXT [variable]
- ON variable { GOTO,GOSUB }
- PRINT expression-list [; ,]
- READ variable-list

- REM [text]
- RETURN
- RESTORE
- SLEEP milliseconds
- STOP

These are the functions currently implemented:

- ABS, ATN, COS, EXP, FRE, INT, LOG, POW, RND, SGN, SIN, SQR, TAN
- ASC, CHR$, LEFT$, LEN, MID$, RIGHT$, STR$, VAL

Also, a number of system variables are implemented:

- LINE, ERRLIN, PI
- DATE$, TIME$, VERSION$

Extending the language.

Extending the language is easy - you can register a new BASIC function, as shown here with a `sleep` function for the XMEGA:

```
register_function_1(FUNC_KEYWORD, "SLEEP", do_sleep, kind_numeric);
...
static int
do_sleep(basic_var *rv, const basic_var *delay)
{
  delay_ms(delay->value.number);

  rv->kind = KIND_NUMERIC;
  rv->value.number = 0;

  return 0;
}
```

Let's use that new keyword.

```
C:\> basic
VARCem BASIC Interpreter version 1.1.1
Copyright 2015-2016 Johan Van den Brande
Copyright 2023 Fred N. van Kempen
 _                 _
| |__   __ _ ___(_) ___
| '_ \ / _` / __| |/ __|
| |_) | (_| \__ \ | (__
|_.__/ \__,_|___/_|\___|

READY.
load "hello"
READY.
list
10 INPUT "YOUR NAME?"; NAME$
20 CLS
30 FOR I=1 TO LEN(NAME$)
40 PRINT MID$(NAME$,I,1);
```

```
50 SLEEP(500)
60 NEXT
70 PRINT
READY.
```

Port

It should be easy to port the interpreter to other architectures. As an example there is a port to an XMega 128A4U included using the Batsocks breadmate board.

Use

There is a simple REPL for the BASIC interpreter. You can use it in an interactive way, just as you would do on a 80's era computer.

```
C:\> basic
VARCem BASIC Interpreter version 1.1.1
Copyright 2015-2016 Johan Van den Brande
Copyright 2023 Fred N. van Kempen
 _               _
| |__   __ _ ___(_) ___
| '_ \ / _` / __| |/ __|
| |_) | (_| \__ \ | (__
|_.__/ \__,_|___/_|\___|

READY.
10 PRINT "HELLO"
20 GOTO 10
```

You can give it a BASIC file on the command line.

```
$> basic examples/diamond.bas
          *
         ***
        *****
       *******
      *********
     ***********
    *************
   ***************
  *****************
   ***************
    *************
     ***********
      *********
       *******
        *****
         ***
          *
```

You can also use the shebang operator to make standalone BASIC scripts

```
#!/usr/bin/env basic

10 RADIUS=10
20 FOR I=1 TO RADIUS-1
30 W=INT(RADIUS*SIN(180/RADIUS*I*3.1415/180))
40 PRINT SPC(RADIUS-W);:FOR J=1 TO 2*W:PRINT "*";:NEXT J:PRINT
50 NEXT I
```

```
$> ./examples/circle.bas
       ******
     **********
  ****************
 ******************
********************
 ******************
  ****************
     **********
       ******
```

It is easy to embed the interpreter into your own application.

```
    /* Create an instance of the interpreter. */
    basic_init(16*1024, 2*1024, my_error, my_ready);

    basic_eval("10 PRINT \"HELLO\"");
    basic_eval("RUN");

    basic_destroy();
```

where `my_error` and `my_ready` are the local implementations of the ERROR and READY handlers.

On OSX/POSIX you can use the 'BASIC_PATH' environment variable to set the folder used for loading and saving BASIC programs. The 'BASIC_PATH' defaults to '.'.

BASIC programs are expected to end with '.bas'. You can use LOAD, SAVE, DELETE and DIR.

Authors and Credits

The original code was written for the AVR X/Mega platform by Johan Van den Brande, who also made that work on Apple's OSX. Support for the Linux platform was added by clarking clarkaaron@hotmail.com in August of 2022, and the project was revived and merged into the VARCem emulator project by Fred van Kempen in 2023.

Copyright