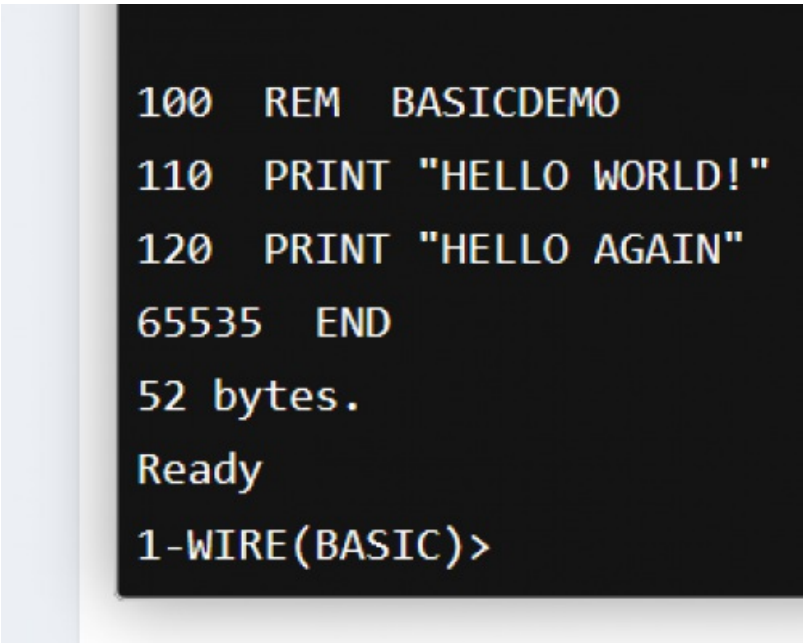

Bus Pirate BASIC script reference

Bus Pirate v3b Firmware v5.10

http://buspirate.com/tutorial/bus-pirate-basic-script-reference

Generated 2019-01-23

Table of Contents

BASIC script mode is entered by typing 's' at the Bus Pirate commandline. You need to take care of entering the correct mode and set it up (things like speed, Hiz, etc.).

Introduction

This isn't intended as a guide in learning how to program. General programming knowledge is assumed. Be aware that only basic checking is done and there are no warnings printed to the terminal (except those intended by the program with print statements). The editor is very rudimentary and does not check if the syntax is right. The language is loosely based on BASIC.

Enter script mode

```
1-WIRE>s
1-WIRE(BASIC)>
```

NEW

```
1-WIRE(BASIC)>new
Ready
1-WIRE(BASIC)>
```

The memory is cleared by entering the 'NEW' command.

LIST

```
1-WIRE(BASIC)>list

100  REM  BASICDEMO
110  PRINT "HELLO WORLD!"
120  PRINT "HELLO AGAIN"
65535  END
52 bytes.
Ready
1-WIRE(BASIC)>
```

From the basic commandline programs can be entered. The basicinterpreter uses linenumbers followed by statements. After this the program can be listed by the command 'LIST'.

RUN

```
1-WIRE(BASIC)>run
HELLO WORLD!
HELLO AGAIN

Ready
1-WIRE(BASIC)>
```

You can also run it with the command 'RUN'.

EXIT

```
1-WIRE(BASIC)>exit
Ready
1-WIRE>
```

'EXIT' leaves the script mode.

Variables

A..Z (26) variables are possible. The variable are internally 16bit signed

General basic commands

LET

assigns a variable. Another variable, constants or functions that returns a value (e.g. RECEIVE)

```
10 LET A=B+1
```

IF {ifstat} THEN {truestat} ELSE {falsestat}

Evaluate the {ifstat} if it evaluate to a value that is not zero {truestat} get executed otherwise {falsestat}.

```
10 IF A=1 THEN GOTO 100 ELSE PRINT "A IS NOT 1"
```

GOTO {line}

jumps to line {line}, without remembering where it came from (see also GOSUB)

```
10 GOTO 100
20 PRINT "line 20"
100 PRINT "line 100"
```

line 20 doesn't get executed

GOSUB {line} and RETURN

jumps to line {line}, executes from there till a RETURN and return to the line after the GOSUB.

```
10 GOSUB 1000
20 PRINT "line 20"
30 END
1000 PRINT "line 1000"
1010 RETURN
```

Stack is 10 levels deep, so 10 nested gosubs are possible.

REM {text}

Puts a remark into the code, but gets skipped.

```
10 REM A WONDERFULL PROGRAM
```

Don't use REM between DATA statements!

PRINT {text}

Prints {text} to the terminal. Variable and statement can be mixed and are seperated with a ';'. A ';' at the end suppresses a newline.

```
10 PRINT "A = ";A
20 PRINT "RECEIVED: ";RECEIVE
30 PRINT "B = ";
40 PRINT B
```

INPUT {question},{var}

Ask {question} and put the answer the user gave into {var}

```
10 INPUT "A = ",A
```

FOR {var}={minvalue} TO {maxvalue} {stats} NEXT {var}

Assigns value {minvalue} to variable {var}, executes statements {stats} until NEXT is encountered. Variable {var} wil be increased by one, {stats} is again executed, until {var} has the value {maxvalue}.

```
10 FOR A=1 TO 10
20 PRINT "A = ";A
30 NEXT A
```

for/nexts can be nested 4 deep.

READ {var}

DATA {val1}, {val2}, .. {val1}, {val2},

Read a value into variable {var}. The values are stored in DATA statements.

```
10 READ A
20 PRINT "A = ";A
30 READ A
40 PRINT "A = ";A
1000 DATA 10,20
```

Buspirate specific basic commands

Note: Not all commands are supported with every mode!

START

Same as the buspirate '[' command

```
10 START
```

STOP

Same as the buspirate ']' command

```
10 STOP
```

STARTR

Same as the buspirate '{' command

```
10 STARTR
```

STOPR

Same as the buspirate '}' command

```
10 STOPR
```

SEND {val/var}

Sends a value {val} or variable {var} over the bus.

```
10 SEND 10
20 SEND A
```

Some protocols send/receive at the same time. This is also possible:

```
10 LET A=SEND 100
20 PRINT "SEND 100 GOT ";A
```

RECEIVE

Receives data from the bus. With some protocols it returns value >255 to signal busstates (like no data, got ACK, etc), other protocols are 16 bit (like pic).

```
10 LET A=RECEIVE
20 PRINT "A = ";A
```

CLK {val}

Controls the CLK line, its behaviour depends on val; 0=low, 1=high, 2=pulse.

```
10 CLK 2
```

DAT {val}

Controls the DAT line, its behaviour depends on val; 0=low, 1=high.

```
10 DAT 0
```

DAT value can also be read:

```
10 LET A=DAT
20 PRINT "A = ";A
```

BITREAD

Same as the buspirate '!' command.

```
10 LET A=BITREAD
20 PRINT "A = ";A
```

ADC

Reads the ADC. Value returned is 10bits (no conversion!).

```
10 LET A=ADC
20 PRINT "A = ";A
```

AUX {val}

Controls the AUX line, its behaviour depends on val; 0=low, 1=high.

```
10 AUX 1
```

AUX value can also be read:

```
10 LET A=AUX
20 PRINT "A = ";A
```

AUXPIN {val}

Controls which pin is controlled by the AUX statement; 0=AUX; 1=CS

```
10 AUXPIN 1
```

PSU {val}

Controls the onboard voltage regulators; 0=off, 1=on

```
10 PSU 1
```

PULLUP {val}

Controls the onboard pullup resistors; 0=off, 1=on

```
10 PULLUP 0
```

DELAY {var}

Delays {var} ms

```
10 DELAY 100
```

FREQ {var}

DUTY {var}

Controls the onboard PWM generator. Frequency of 0 disables it. (same limits apply as regular PWM command ('g'))

```
10 freq 100
20 duty 25
```

Scan or click me to return to the original page
(http://buspirate.com/tutorial/bus-pirate-basic-script-reference)

Visit us:

BusPirate.com Breakout boards, chips, devices, debugging tutorials
DirtyPCBs.com Cheap prototype PCBs, SLA prints, custom cables, and more!
DangerousPrototypes.com Open source hardware, tools, and toys