MyForth on Arduino Hardware
Charley Shattuck

MyForth is a method for building a target compiled tethered Forth for micro-controllers.

It is an experiment in simplicity.

Being “my” Forth, it is designed to be used in Linux.

It is not an ANSI Forth. It is somewhat similar to arrayForth.

Pre-existing software... VIM, Gforth, AVRdude, Minicom

VIM, editor with custom syntax coloring

Gforth is the host Forth compiler
    Scripts 'c', 't'

AVRdude is the Arduino bootloader
    Script 'd'

Minicom is used to interactively test
    Script 'm'

Components

Simple target compiler/assembler

Decompiler/disassembler

Macros (inline compilation)

Primitives (macros and subroutines)

Standalone interpreter

Scripts (c, t, d, m)

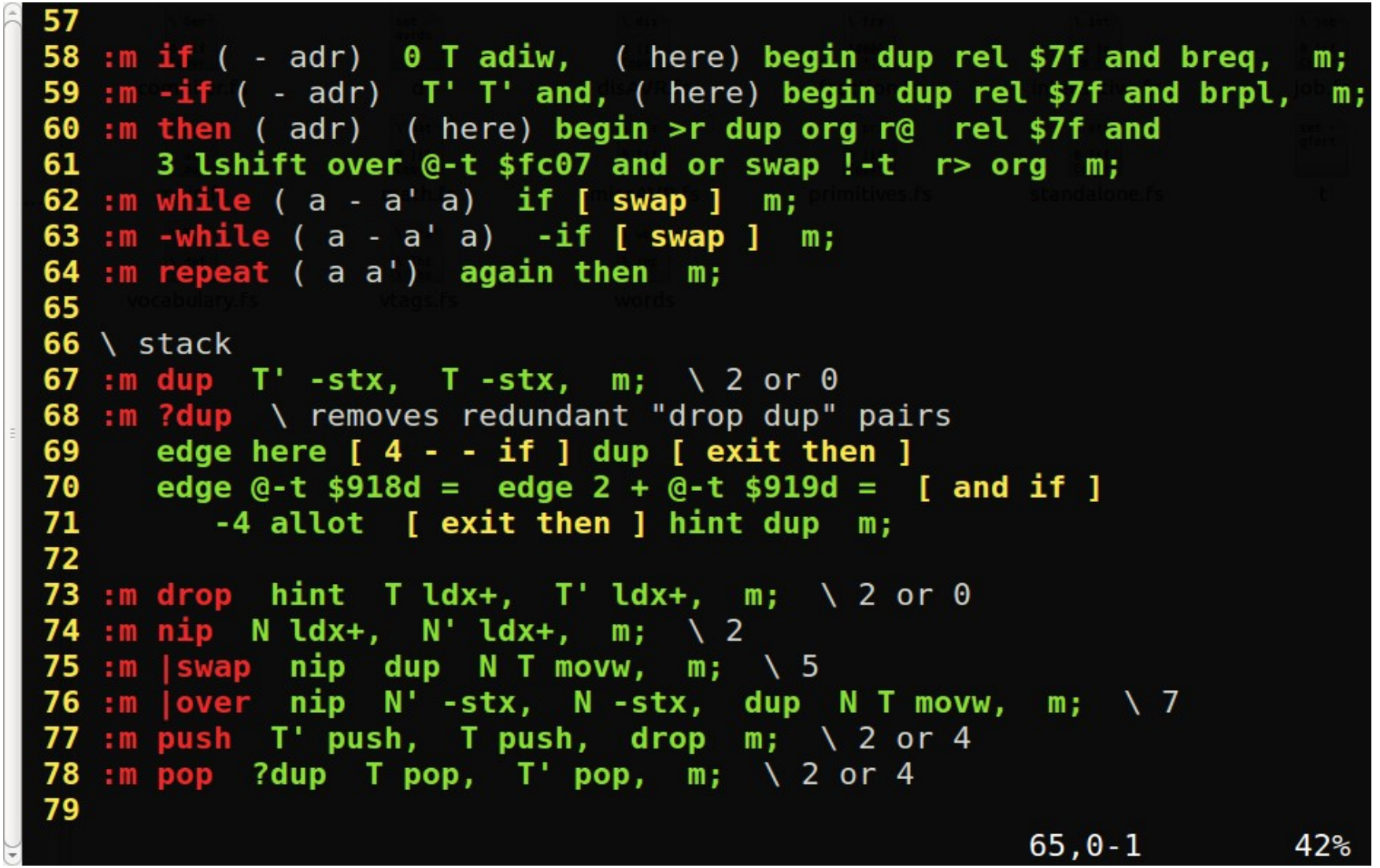
Target Compiler

```
38 \ 0 constant start  \ Reset vector.
39 \ 8192 constant target-size
40 create target-image target-size allot
41 target-image target-size $ff fill  \ ROM erased.
42 : there   ( a1 - a2)   target-image + ( start +) ;
43 : c!-t   ( c a - )   there c! ;
44 : c@-t   ( a - c)   there c@ ;
45 : !-t  ( n a - )  there over 8 rshift over 1 + c! c! ;
46 : @-t  ( a - n)  there count swap c@ 8 lshift + ;
47
48 variable tdp  \ Rom pointer.
49 :m HERE   (  - a)   tdp @ m;
50 :m ORG   ( a - )   tdp ! m;
51 :m ALLOT   ( n - )   tdp +! m;
52 :m ,   ( n - )   HERE !-t 2 ALLOT m;
53 : ,-t   ( n - )   target , m;
54
55 variable trp  \ Ram pointer.
56 : cpuHERE  (  - a)   trp @ ;
57 : cpuORG  ( a - )  trp ! ; 8 cpuORG
58 : cpuALLOT  ( n - )  trp +! ;
59 : report  cr ." HERE=" target HERE host u. cr ;
60
                                                47,0-1        41%
```

Decompiler Disassembler

```
 ok
see calibrate
     calibrate
0510 DFEE  -18 rcall, ping
0511 939E  T' -stx, (dup
0512 938E  T -stx,
0513 E182  $12 T ldi, (# 18
0514 E090  $0 T' ldi,
0515 DB53  -1197 rcall, u/mod
0516 939E  T' -stx, (dup
0517 938E  T -stx,
0518 DF5B  -165 rcall, .
0519 DFF1  -15 rcall, /inch
051A DB0F  -1265 rcall, !
051B 918D  T ldx+, (drop
051C 919D  T' ldx+,
051D 9508  ret, ;
     first
051E 939E  T' -stx, (dup
051F 938E  T -stx,
0520 E086  $6 T ldi, (# 262
0521 E091  $1 T' ldi,
0522 9508  ret, ;
```

Macros (in-line assembly)



```
57
58 :m if ( - adr)  0 T adiw,  ( here) begin dup rel $7f and breq,  m;
59 :m -if ( - adr)  T' T' and, ( here) begin dup rel $7f and brpl,  m;
60 :m then ( adr)  ( here) begin >r dup org r@  rel $7f and
61    3 lshift over @-t $fc07 and or swap !-t  r> org  m;
62 :m while ( a - a' a)  if [ swap ]  m;
63 :m -while ( a - a' a)  -if [ swap ]  m;
64 :m repeat ( a a')  again then  m;
65
66 \ stack
67 :m dup  T' -stx,  T -stx,  m;  \ 2 or 0
68 :m ?dup  \ removes redundant "drop dup" pairs
69    edge here [ 4 - - if ] dup [ exit then ]
70    edge @-t $918d =  edge 2 + @-t $919d =  [ and if ]
71       -4 allot  [ exit then ] hint dup  m;
72
73 :m drop  hint  T ldx+,  T' ldx+,  m;  \ 2 or 0
74 :m nip  N ldx+,  N' ldx+,  m;  \ 2
75 :m |swap  nip  dup  N T movw,  m;  \ 5
76 :m |over  nip  N' -stx,  N -stx,  dup  N T movw,  m;  \ 7
77 :m push  T' push,  T push,  drop  m;  \ 2 or 4
78 :m pop  ?dup  T pop,  T' pop,  m;  \ 2 or 4
79
                                                  65,0-1        42%
```

Primitives (subroutines)

Sample Application

```
 1 \ main.fs
 2 : us ( n)  for  13 toggle, next ;
 3 : pulse  12 output, 12 low, 5 # us 12 high,  5 # us 12 low, ;
 4 : hedge  750 # us ;
 5 : ready  12 input, begin 12 PIN sbis, again 100 # us ;
 6 : bail pop drop ;
 7 : measure ( 0 limit - count)
 8    for 1 #+ 5 # us 12 PIN sbis, bail ; next ;
 9 : ping ( - n) pulse hedge ready 0 # 20000 # measure ;
10
11 : /inch  variable # ;
12 : calibrate  ping 18 # u/mod dup . /inch ! drop ;
13
14 : first  variable # ;
15 : nothing  first @ if [ char . ] # emit cr 0 # first !  then drop ;
16 : feet ( n)  [ char ' ] # -if
17 : inches ( n)  [ char " ] # then swap (u.) emit space ;
18 : ft,in ( n) 12 # u/mod feet inches ;
19 : tell ( n)  -120 # + -if 120 # + ft,in cr -1 # first ! ;
20    then drop nothing ;
21
22 : in  ping /inch @ u/mod tell drop ;
23 : wait  1000 # for 1000 # for next next ;
24 : go  -1 # first ! begin in wait again
                                                    10,0-1        Top
```