arduino/AVR: interactive development with `amforth`

Erich Wälde

FOSDEM 2011

The long toolchain

- controller (atmega32), board, power
- isp dongle (AVR mkII, usbasp, sp12)
- editor (emacs)
- assembler (avra, AvrAssembler2+wine)
- flash tool (avrdude)
- serial connection
- serial terminal (minicom)
- amforth source
- upload via serial tool (am4up, amforth-upload.py)

How we used to do it

- edit, save
- compile/assemble

- erase
- flash

- reset and watch

How we used to do it

- repeat {
- edit, save
- compile/assemble

- erase
- flash

- reset and watch
- } until done

How we used to do it

- repeat {
- edit, save
- compile/assemble
- move controller to burner
- erase
- flash
- move controller back
- reset and watch
- } until done

Interactivity to rescue

- log in to the controller

- compile new functions
- test/use them right away

Interactivity to rescue

- log in to the controller
- repeat {
- compile new functions
- test/use them right away
- } until done

Interactivity to rescue

- compile/assemble some system
- erase, flash controller
- log in to the controller
- repeat {
- compile new functions
- test/use them right away
- } until done

`amforth`, maybe?

- small standalone system
- interactive
- compiled
- extensible
- stack based
- postfix

`amforth`, maybe?

- small standalone system
- interactive
- compiled
- extensible
- stack based
- postfix

- hello world (message)
- hello world (blinky LEDs)
- talking to an I2C device

Demo System

- atmega32 @ 11.0592 MHz
- 32 KiB flash, 2 kiB ram, 1 kiB eeprom
- some additional words built in (assembly)
- some additional words loaded (amforth)
- serial connection at 115200 Baud, 8N1
- LEDs on PORTB 0,1,2

Contact!

```
amforth 4.2 ATmega32
>
 ok
>
```

```
\ LEDs r g b on PORTB 2 1 0
PORTB 0 portpin: blue
PORTB 1 portpin: green
PORTB 2 portpin: red

red   pin_output   red   high
green pin_output   green high
blue  pin_output   blue  high
```

```
\ we do not use builtin twi!
PORTB 3  portpin: sda
PORTB 4  portpin: scl

sda pin_output  sda high
scl pin_output  scl high
```

```
\ setting i2c levels and show it
: sda0   sda low   blue high ;
: sda1   sda high  blue low  ;
: scl0   scl low   red  high ;
: scl1   scl high  red  low  ;
```

```
\ defered wait
: wait-long  &500 0 do 1ms loop ;
: wait-short  &50 0 do 1ms loop ;
Rdefer wait
: slow  ['] wait-long  is wait ;
: fast  ['] wait-short is wait ;
slow
```

```
\ pulses testing wait
: pulses ( n -- )
  0 ?do
    red low wait red high wait
  loop
;
slow 5 pulses
fast 25 pulses
```

```
\ clock out one bit
: bit>i2c ( bit -- )
  if sda1 else sda0 then    \ set data
  scl1 wait scl0 wait       \ clock it out
;
```

```
\ get bit value (0|1)
: get.bit ( byte pos -- bit )
  1 swap lshift    \ -- byte bitmask
  and              \ -- bit
;
```

```
\ clock out one Byte, MSB first!
: byte>i2c ( byte -- )
  8 0 do
    dup   7 i -   get.bit
    bit>i2c
  loop
  drop
;
```

```
\ create start|stop condition
: i2c.start   sda0 wait scl0 wait ;
: i2c.stop    scl1 wait sda1 wait ;
```

```
\ read  ACK:t | NACK:f
: ack<i2c ( -- bit )
  sda pin_input
  scl1 wait
  sda pin_low? sda pin_output
  scl0 wait
;
```

```
\ don't forget to make it fast!
' noop is wait
```

```
\ send byte to address $40
: >8io  ( byte -- )
  i2c.start
    $40 byte>i2c ack<i2c drop
        byte>i2c ack<i2c drop
  i2c.stop
;
```

```
: i2c.scan
  $FF 0 do
    i2c.start
      i byte>i2c ack<i2c
    i2c.stop
    if   i . cr  then
  2 +loop
;
```

So what is `amforth` then?

- standalone Forth for Atmega controllers
- written by Matthias Trute
- free software (GPL v2)
- interactive
- compiled and extensible
- small (8 kiB flash, 50 B eeprom, 150 B ram)
- stack oriented and postfix
- case sensitive
- without safe guards

So what is `amforth` then? (2)

- comes with a collection of library modules
- collaborative multitasker
- math is scaled integer
- double cell arithmetic available
- floating point library module
- up to 8 wordlists
- loadable assembler for inline assembly code

Sources (small selection)

- amforth.sourceforge.net
- amforth-devel@lists.sourceforge.net
- www.atmel.com
- Leo Brodie – Starting Forth
  home.iae.nl/users/mhx/sf.html
- www.forth-ev.de

Thank You!

slide set generated with LaTeX/Beamer(Manhattan.sty)