

This pad builds on [[hacklab-xmega/rev.4129]], created by Antti & heikkki & tkln & suovula & rambo & Heikki & jautero & mokis & [unnamed author] & Kremmen & jari & netl & ExTechOp & Hukka

Atmelin XMEGA mikrokontrollereihin liittyvää tauhkaa

(nämä testattu nyt Mac OS X + CrossPack-AVR-20121207.dmg)

Vaihda 32a4 -> 128a1, jos käytät isoa punaista lautta!

Testisoftien kääntö ilman makefilehivvityksiä

- \*compile and link

- \*avr-gcc -mmcu=atxmega32a4u -I. -O3 -funsigned-char -funsigned-bitfields -fpack-struct -fshort-enums -Wall -Wa,-adhlns=./xtest.lst -Wl,-Map=xtest.map,--cref xtest.cpp -o xtest.elf

- \*store disassembly

- \*avr-objdump -Dx xtest.elf > xtest.asm

- \*convert .elf to .hex

- \*avr-objcopy -O ihex xtest.elf xtest.hex

- \*upload with AVRISPMkII

- \*avrdude -c avrisp2 -P usb -p x32a4 -U flash:w:xtest.hex

Tällä saadaan DFU-bootloaderi sisään

- \*download Atmel AVR1916: USB DFU Boot Loader for Atmel XMEGA

- \*<http://www.atmel.com/Images/AVR1916.zip>

- \*XMEGA\_bootloaders\_v104/binaries/atxmega32a4u\_104.hex

- \*XMEGA\_bootloaders\_v104/binaries/atxmega128a1u\_104.hex

- \*<http://www.atmel.com/Images/doc8429.pdf>

- \*some documentation

- \*Suovulan binääripatchätyt bootloaderit

- \*<https://github.com/HelsinkiHacklab/partyhatwork/tree/master/software/xmegabootloaders>

- \*set fuses

- \*avrdude -c avrisp2 -P usb -p x32a4 -U fuse1:w:0x00:m -U fuse2:w:0x9F:m -U fuse4:w:0xFF:m -U fuse5:w:0xF0:m

- \*WDWP = 8CLK

- \*WDP = 8CLK

- \*BOOTRST = BOOTLDR <- resetissä bootloaderiin

- \*TOSCSEL = ALTERNATE <- 32768 Hz kellokide PEx ja PEy (32a4)

- \*BODPD = DISABLED

- \*RSTDISBL = UNSET

- \*SUT = 0MS

- \*WDLOCK = UNSET

- \*JTAGEN = DISABLED <- B-portin 4 ylintä bittiä käyttöön (128a1)

- \*BODACT = DISABLED

- \*EESAVE SET

- \*BODLVL 3V0

- \*convert hex to bin for avrdude

- \*avr-objcopy -I ihex -O binary atxmega32a4u\_104\_Pxy.hex atxmega32a4u\_104.bin

- \*program with AVRISPMkII

- \*avrdude -e -c avrisp2 -P usb -p x32a4 -U boot:w:atxmega32a4u\_104.bin

Tällä saadaan piiri menemään bootissa DFU-moodiin

- \*short pin Pxy to GND

- \*connect USB-cable

```

*in Linux
*lsusb
*...
*in Mac OS X
*system_profiler SPUSBDataType
*Etsi: Vendor ID: 0x03eb (Atmel Corporation)
*Laiteid:t vaihtelee bootloaderin version mukaan.
*32A4U sample: "Composite Device:, Product ID: 0x2fe4, Vendor ID: 0x03eb (Atmel Corporation),
Version: 0.04, Speed: Up to 12 Mb/sec, Location ID: 0x1d100000 / 3, Current Available (mA): 500, Current
Required (mA): 100"
*128A1U sample: "DFU ATXMEGA128A1U:, Product ID: 0x2fed, Vendor ID: 0x03eb (Atmel
Corporation), Version: 0.04, Speed: Up to 12 Mb/sec, Manufacturer: ATMEL, Location ID: 0x1d100000 / 3,
Current Available (mA): 500, Current Required (mA): 100"
*upload with DFU
*dfu-programmer atxmega32a4u flash xtest.hex

```

```

----hombrew updated formula (brew edit dfu-programmer)-----
require 'formula'

```

```

class DfuProgrammer < Formula
  head 'https://github.com/nedos/dfu-programmer.git'
  homepage 'https://github.com/nedos/dfu-programmer'

  depends_on 'libusb-compat'

  def install
    system "./configure", "--disable-debug", "--disable-dependency-tracking",
      "--prefix=#{prefix}"
    system "make install"
  end
end
----/hombrew updated formula-----

```

```

-suovulan dfu-programmer korjaus 128A1U:lle, julkaistaan kun on paremmin testailtu-
*https://github.com/nedos/dfu-programmer
*arguments.h add to enum targets_enum line ~105
*tar_atxmega128a1u,
*arguments.c add to static struct target_mapping_structure target_map[] line ~127
*{ "atxmega128a1u", tar_atxmega128a1u, adc_AVR32, 0x2FED, 0x03eb, 0x20000, 0x2000, true, 128,
true, false, 32, 0x0800 },
*make
*sudo make install

```

Binary batching different sense pin to DFU-bootloaders

```

/usr/local/CrossPack-AVR/avr/include/avr/iox32a4u.h
/usr/local/CrossPack-AVR/avr/include/avr/iox128a1u.h

```

```

atxmega32a4u_104.hex = pin PC3
*800A:  92 F0 06 40  sts 0x640, R15  // PORTC_DIR(0x0640)

```

```

*8010:  93 00 06 53  sts 0x653, R16  // PORTC_PIN3CTRL(0x0653)
*801C:  91 00 06 48  lds R16, 0x648  // PORTC_IN(0x0648)
*8020:  FF 03          sbrs R16, 3      // PORT_PIN(3)
*80DE:  92 F0 06 53  sts 0x653, R15  // PORTC_PIN3CTRL(0x0653)

```

PC3 -> PD3

```

F0924006 -> F0926006
00935306 -> 00937306
00914806 -> 00916806
03FF      -> 03FF
F0925306 -> F0927306

```

Pin 3 -> 0

```

00937306 -> 00937006
03FF      -> 00FF
F0927306 -> F0927006

```

PF0/5 -> PJ7

```

F092A006 -> F0920007
0093B506 -> 00931707
0091A806 -> 00910807
00FF      -> 07FF
F092B506 -> F0921707

```

PD0 -> PE0

```

F0926006 -> F0928006
00937006 -> 00939006
00916806 -> 00918806
00FF      -> 00FF
F0927006 -> F0929006

```

```

PORTA_DIR = 0x0600, PORTA_IN = 0x0648, PORTA_PINxCTRL = 0x061x
PORTB_DIR = 0x0620, PORTB_IN = 0x0628, PORTB_PINxCTRL = 0x063x
PORTC_DIR = 0x0640, PORTC_IN = 0x0648, PORTC_PINxCTRL = 0x065x
PORTD_DIR = 0x0660, PORTD_IN = 0x0668, PORTD_PINxCTRL = 0x067x
PORTE_DIR = 0x0680, PORTE_IN = 0x0688, PORTE_PINxCTRL = 0x069x
PORTF_DIR = 0x06A0, PORTF_IN = 0x06A8, PORTF_PINxCTRL = 0x06Bx

```

```

PORTH_DIR = 0x06E0, PORTH_IN = 0x06E8, PORTH_PINxCTRL = 0x06Fx
PORTJ_DIR = 0x0700, PORTJ_IN = 0x0708, PORTJ_PINxCTRL = 0x071x
PORTK_DIR = 0x0720, PORTK_IN = 0x0728, PORTK_PINxCTRL = 0x073x

```

```

PORTQ_DIR = 0x07C0, PORTQ_IN = 0x07C8, PORTQ_PINxCTRL = 0x07Dx
PORTR_DIR = 0x07E0, PORTR_IN = 0x07E8, PORTR_PINxCTRL = 0x07Fx

```

Atomiset käyskyt jotka löytyvät uudemmissa xmegaista:

```

inline uint8_t atomicExchange(volatile uint8_t * ptr, uint8_t val) {
    // uint8_t tmp = *ptr; *ptr = val; val = tmp;
    asm volatile ("xch    Z, %0" : "=r" (val) : "z" (ptr), "0" (val));
    return val;
}

```

```
}
```

```
inline uint8_t atomicLoadAndSet(volatile uint8_t * ptr, uint8_t val) {  
    // uint8_t tmp = *ptr; *ptr = tmp | val; val = tmp;  
    asm volatile ("las    Z, %0" : "=r" (val) : "z" (ptr), "0" (val));  
    return val;  
}
```

```
inline uint8_t atomicLoadAndClear(volatile uint8_t * ptr, uint8_t val) {  
    // uint8_t tmp = *ptr; *ptr = tmp & ~val; val = tmp;  
    asm volatile ("lac    Z, %0" : "=r" (val) : "z" (ptr), "0" (val));  
    return val;  
}
```

```
inline uint8_t atomicLoadAndToggle(volatile uint8_t * ptr, uint8_t val) {  
    // uint8_t tmp = *ptr; *ptr = tmp ^ val; val = tmp;  
    asm volatile ("lat    Z, %0" : "=r" (val) : "z" (ptr), "0" (val));  
    return val;  
}
```