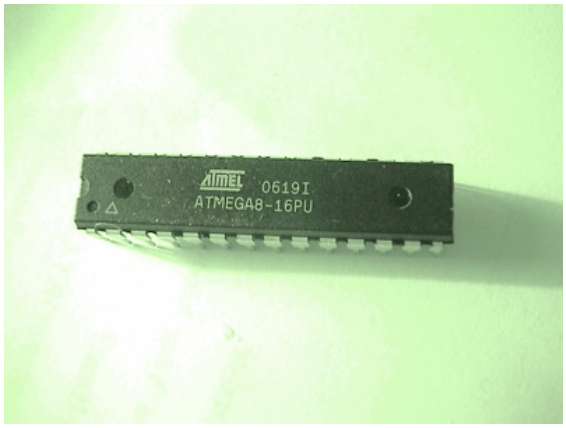
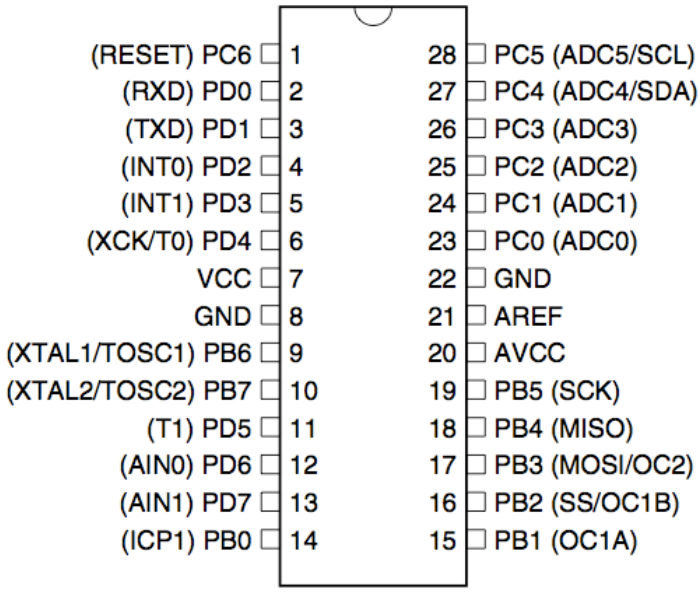
clubtransmediale08: xxxxx-workshops: one bit music /f0
-part1/4-

ATmega8

8bit microcontroller with 8kb program memory (8192 bytes)
one of the most used in the avr family (www.atmel.com)
allround and fairly cheap (~€2)
lots of example code online (www.avrbeginners.net www.avrfreaks.net)
clock speed up to 16mhz
other models from ATtiny13 with 6pins, to ATmega6490 with 100pins





(picture from mega8 datasheet)

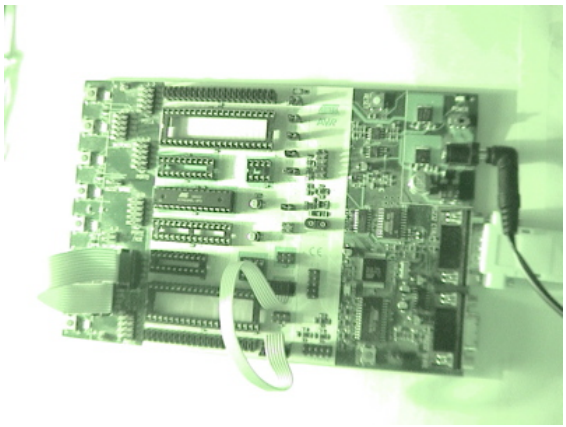
microcontroller ≈ tiny computer
usually you run them on 5 volts
many digital/analog inputs/outputs, usart/spi communication, internal timers/counters/oscillator, eeprom/sram memory, registers, interrupts, stack etc.

write programs for them in c, c++ or assembly
upload with a programmer (serial/parallel/usb)

we'll use the ATmega8L. this is the low-power version that can run directly on 2 aaa batteries (3 volts). but has a maximum of 8mhz (note: not 16 as the standard non-L version)
you'll want to reduce cpu speed to save battery power anyway
to keep external components at a minimum, we'll also use the internal oscillator. which is 8mhz max

programming

to program microcontrollers we need a programmer
'burns' a binary file (.hex or .rom) onto the chip's flash memory (aka firmware)
erases what was there before. write/erase cycle thousands of times
the stk500 is atmel's own programmer. very flexible but expensive and there are many other brands
buy or build one yourself
building is simple if you have a parallel port, a little bit harder with serial or usb



avrstudio is a freeware programming environment from atmel (windows only)
we'll use the open source avr-gcc and avrdude
winavr (winavr.sourceforge.net)
osx-avr (www.osx-avr.org) - very nice installer with everything
linux (debian: sudo apt-get install gcc-avr avr-libc avrdude)

gcc is used to compile and link your source code into a binary (.hex)
avrdude is used to upload the binary to the chip (burn it to flash memory)
avrdude also lets you set fuses on the chip (e.g. set the internal oscillator's speed)

(
arduino could also be used as a programmer: avrisp emulation (code.google.com/p/mega-isp)
interesting but under development
)

demo

avr-project test
code a blink led
edit makefile to match programmer, make, make flash
try with speaker, change to audible frequency

```
#include <avr/io.h>
```

```
int main(void) {
    DDRB= 0xff;
    int i;
    for(;;) {
        for(i= 0; i<0xffff; i++)
            PORTB= 4;
        for(i= 0; i<0xffff; i++)
            PORTB= 0;
    }
    return 0;
}
```