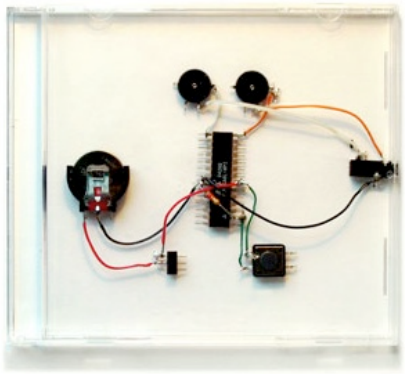
clubtransmediale08: xxxxx-workshops: one bit music /f0
-part3/4-

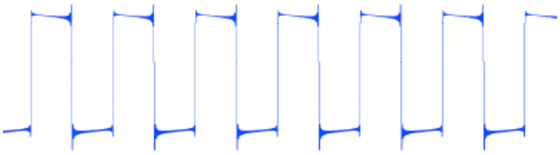
1bit?

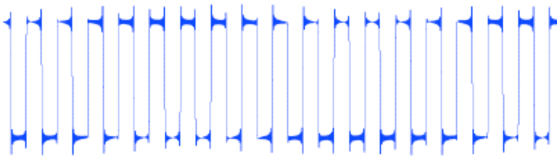
inspiration from tristan perich and others



(picture from tristan's www.onebitmusic.com)

dogma: minimal circuit (battery, chip, speaker), only squarewaves (pwm, bit-bang)
+ volume control and switches
can we at all write music with this?





bit-banging

flip pin high/low at a certain rate
very simple method - same as we just did with the blinking led
any digital output pin can be used
difficult to code anything more interesting than simple melodies
only one thread and delaying blocks other tasks (e.g. sensor input)

note that 5 volts is actually not so good for the speaker
-0.5 to +0.5 more normal for audio; centered around 0
may use voltage divider to scale and offset (2 resistors: 9.1k and 1k)

try

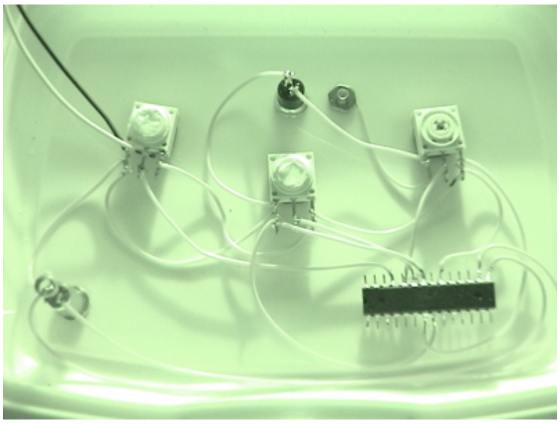
connect the speaker to pin 2 via the 10k pot
try out the onebit_bitbang examples
change the code to make it do something different

pwm

pulse width modulation
constant frequency but varying pulsewidth
used as analog output - average voltage
interrupt driven (sort of multitasking)
internal counter counts up to a max value, when reached it interrupts the main code and calls a function (isr)
more interesting sounds but still very hard to program music

demo

show monijonsyn
explain the code



try

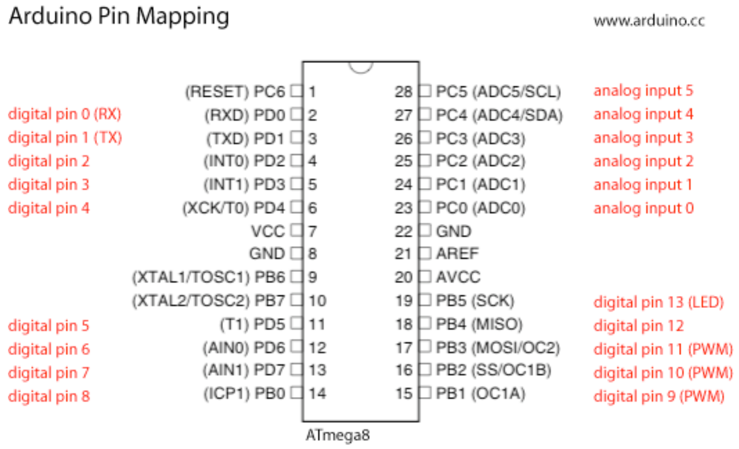
connect the speaker to pin 9 via the 10k pot
try out the onebit_pwm examples
change the code to make it do something different

today's goal

1.
code something interesting using the examples as templates
set up one speaker, one pot, 8 buttons and the arduino

2.
move your arduino code over to the ATmega8L and make a standalone chip
burn it using the stk500 (i'll help out with this)

3.
make a little player/device/instrument
solder the buttons, knob, speaker and batteries to the chip


Arduino Pin Mapping
www.arduino.cc
(RESET) PC6 1
digital pin 0 (RX) (RXD) PD0 2
digital pin 1 (TX) (TXD) PD1 3
digital pin 2 (INT0) PD2 4
digital pin 3 (INT1) PD3 5
digital pin 4 (XCK/T0) PD4 6
VCC 7
GND 8
(XTAL1/TOSC1) PB6 9
(XTAL2/TOSC2) PB7 10
digital pin 5 (T1) PD5 11
digital pin 6 (AIN0) PD6 12
digital pin 7 (AIN1) PD7 13
digital pin 8 (ICP1) PB0 14
28 PC5 (ADC5/SCL) analog input 5
27 PC4 (ADC4/SDA) analog input 4
26 PC3 (ADC3) analog input 3
25 PC2 (ADC2) analog input 2
24 PC1 (ADC1) analog input 1
23 PC0 (ADC0) analog input 0
22 GND
21 AREF
20 AVCC
19 PB5 (SCK) digital pin 13 (LED)
18 PB4 (MISO) digital pin 12
17 PB3 (MOSI/OC2) digital pin 11 (PWM)
16 PB2 (SS/OC1B) digital pin 10 (PWM)
15 PB1 (OC1A) digital pin 9 (PWM)
ATmega8

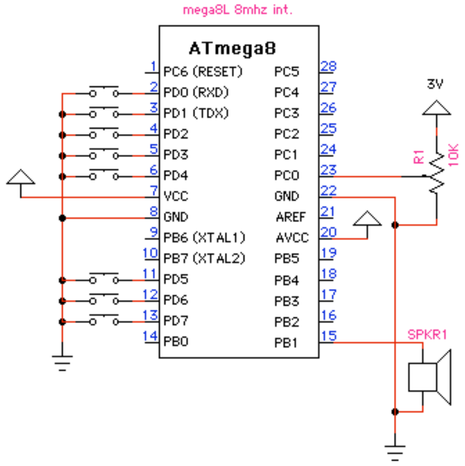
(picture from www.arduino.cc/en/Hacking/PinMapping)

notes

* your code will run at 1/2 the speed (8 instead of 16mhz) when we burn it to the low-power chip. need to compensate for that for your frequencies. i'll help to test this.
* don't start soldering until you've programmed the chip and are happy with the functionality. you can of course reprogram the chip later but it's just a hassle to connect the programmer if soldered pins
* the knob can be used either as volume control or controller (or both - but then linked)

suggested design

arduino digital inputs 0-7 to 8 buttons, pwm 9 to speaker, analog in 0 to knob



parts

atmega8L dip
drehpoti. linear, mono 10k
miniatur-lautsprecher 0,5watt
2 x batterie AAA
halter für 2 AAA, lötfahnen

8 x miniatur-drucktaster
wire