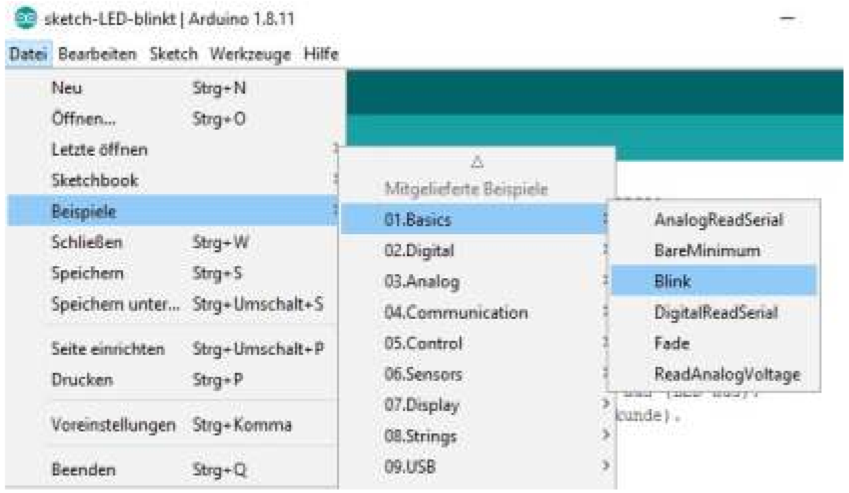
content

Mic-7.1 program examples and commands

Some of the programs we've seen
can also be found in the menu “File” - “Examples”.

Videos of some experiments are also available on the Internet.

If you’re looking for an order or you’re not sure,
how it works, go to the Help menu and
a table opens with “Reference”
with all orders.
We only need a few of them:



Structure

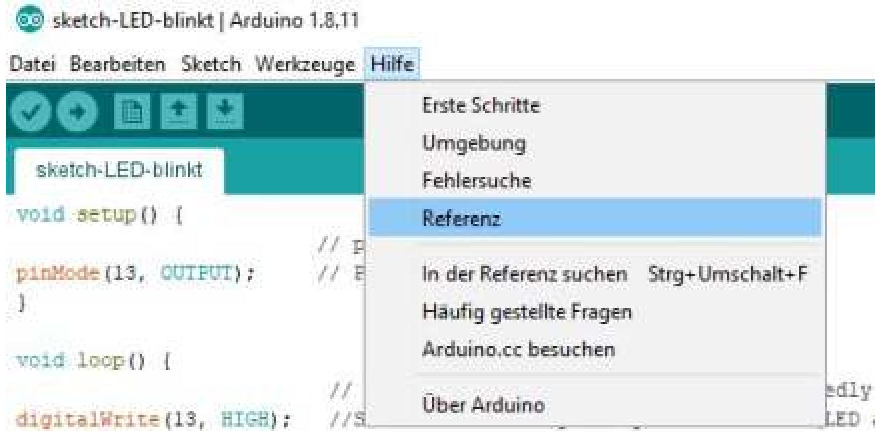
see 4.1

- setup()
- loop()

Control Structures

see 6.3

- if
- if...else
- < (less than)
- > (greater than)



```
void setup() {
pinMode(13, OUTPUT);
}

void loop() {
digitalWrite(13, HIGH);
```

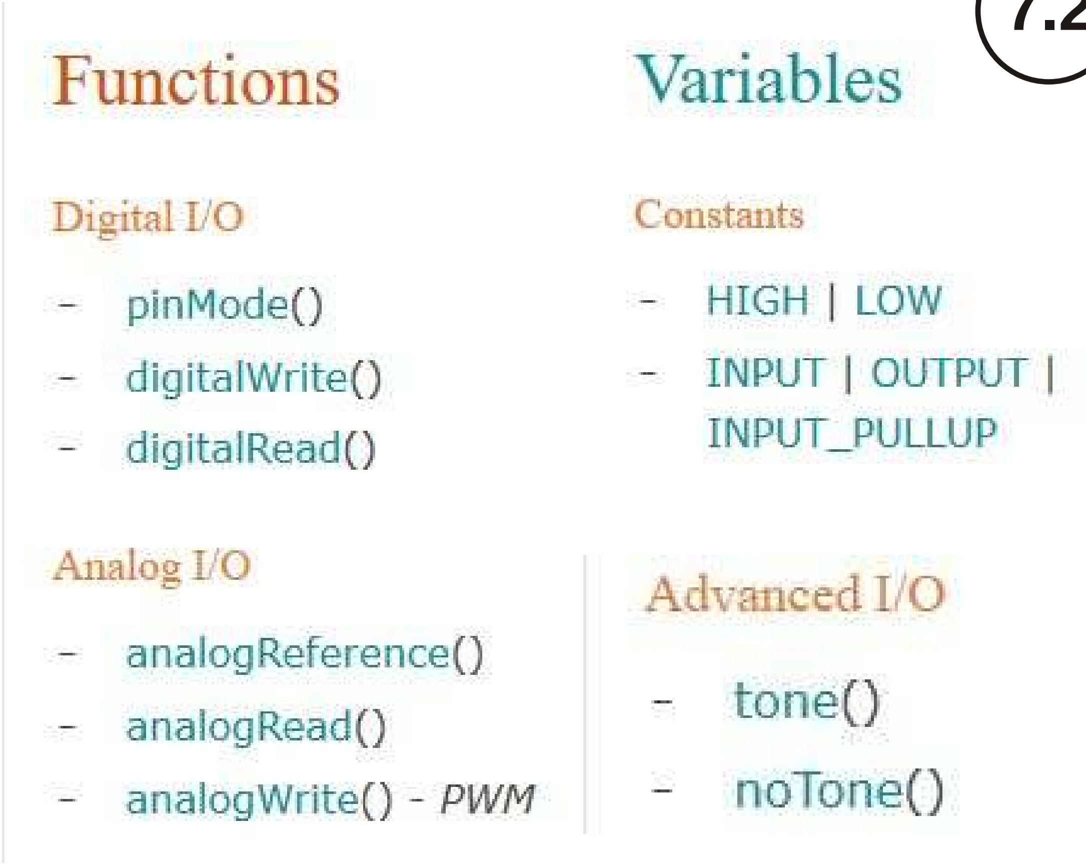
7.2

Here are the commands with which we think of the Port pins signals can output and read. (see 4.2)

Define with pinMode, whether he exit or entrance should be.

With the port pins A0 - A5 for analog voltages also. (see 3.1 and 5.3)

With the tone () command we will in a later Example sounds on one give small speaker and program an entire song.



Functions

Digital I/O

- pinMode()
- digitalWrite()
- digitalRead()

Analog I/O

- analogReference()
- analogRead()
- analogWrite() - *PWM*

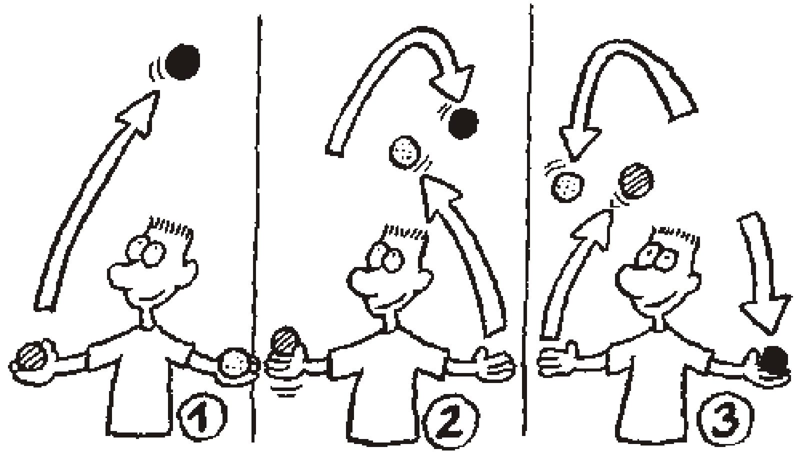
Variables

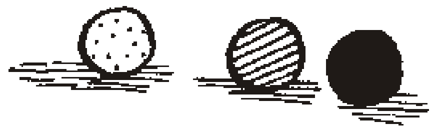
Constants

- HIGH | LOW
- INPUT | OUTPUT | INPUT_PULLUP

Advanced I/O

- tone()
- noTone()





7.3

Experiments with engines or components,
who cannot use more power directly
connected to the Arduino board.
The current on a pin must not be higher
than 15 mA and all pins together
in total not more than 100 mA
be charged.

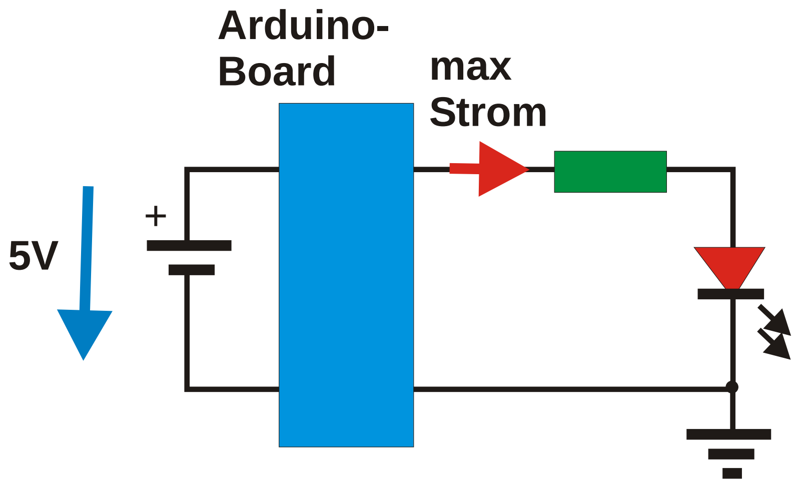
For components with higher performance
need a power amplifier and one
own battery with 5V or higher voltage.
The amplifier can be a transistor.
In the set is e.g. a circuit board for control
a stepper motor.
It can also be a relay.

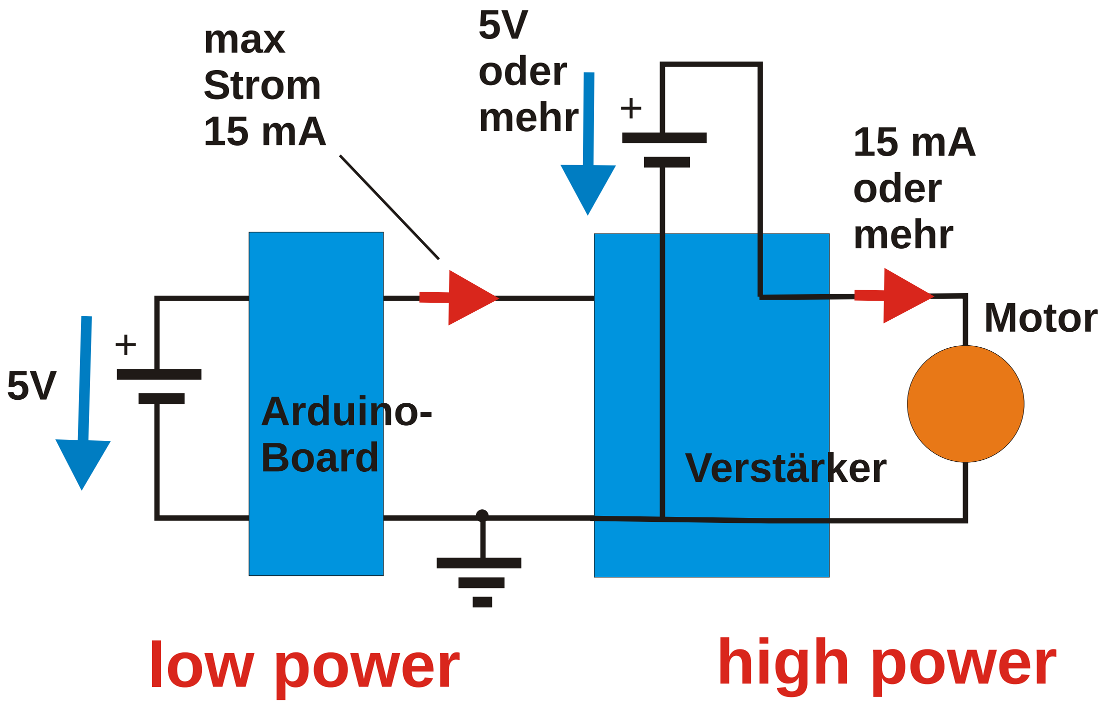
The power P is determined:

P = U * I P in watts (W)
U in volts (V)
I in amperes (A)

So the Arduino can
5 V * 100 mA = 500 mW = 0.5 W.

If you have enough practice with the Arduino
have some knowledge about electronics,
you can also contact such
Dare to experiment.





7.4

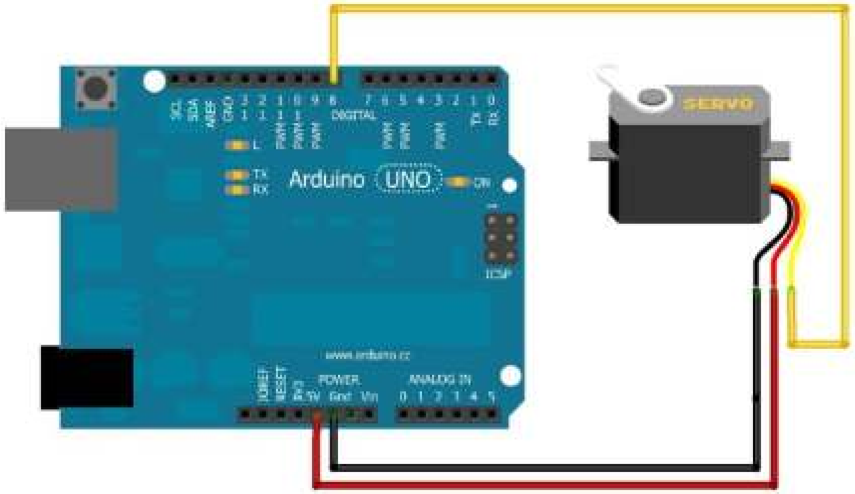
The servo is a small motor with a gear.
Since it needs little power,
it can be connected directly to the Arduino.

Find the description of the program
you in the instructions on page (49 German)
and (62 engl).

In this program a finished
Program part called and by the
Command #include <Srvo.h> in your program
involved.

#include <Servo.h> // The servo library is called.

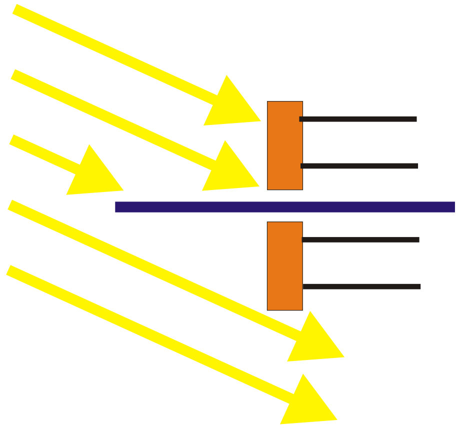
Now you can use the **servoblau.write(90);**



determine in which direction (e.g. 90 °)
he should show.

And what do we do with it?

With 2 photo resistors separated by a wall (cardboard)
we can read in two measured values. From the difference of
We know the two measured values ??from which side the light comes
and can tell the servo to turn in that direction.
A small paper flag can be attached to the servo,
that always moves towards the light like a sunflower.
You want to try something like that, just write a program.
If necessary, request help at mnargast@web.de



8.1 Generate tones

Now it's time for the music lovers.
Output tones with the small speaker (buzzer)
and then program a little melody.

On the back of the buzzer we can see which one is active
and which is passive. There is a “+” on the front. This
Wire must go through the series resistor with pin 8 and the other
be connected to GND.

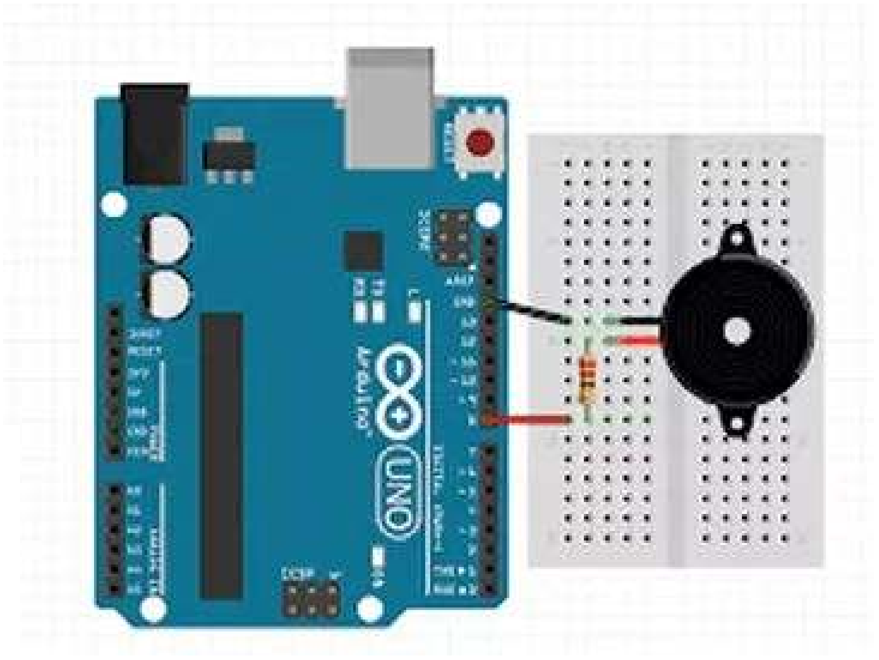
With this command we can output a signal at pin 8;
that e.g. 262 times per second (frequency) from 0 to 5V
and going back again. In buzzer this gives the tone “C4”.

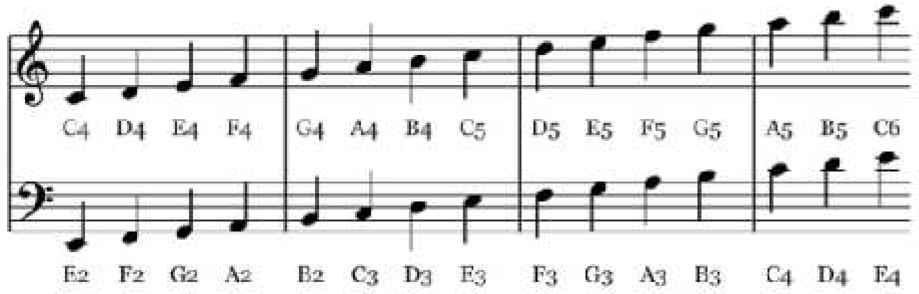
tone (pin, frequency, duration)

and so we alternate two tones
listen for half a second
still frequency = 523 (tone “C5”).

So we write in “loop”:
tone (8,262,500);
tone (8,523,500);

passiv buzzer **activ buzzer**




C4 D4 E4 F4 G4 A4 B4 C5 D5 E5 F5 G5 A5 B5 C6
E2 F2 G2 A2 B2 C3 D3 E3 F3 G3 A3 B3 C4 D4 E4

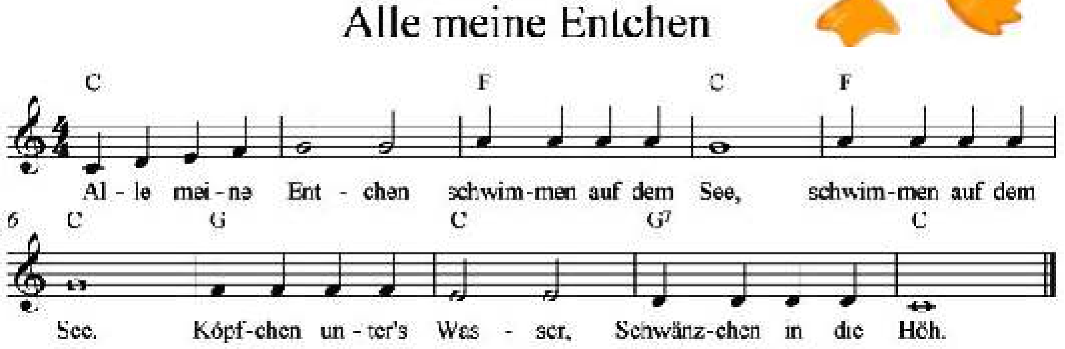
So you don't always get the frequency
, you can use the #define command
(Assignment in setup)

#define constantName value

assign a number to a letter,
e.g. A4 should always be in your program (constant)
Be 440. (! no comma, no semicolon)

#define A4 440

Just copy the table on the right
in your setup.


Alle meine Entchen
C
F
C
F
Al - le mei - ne Ent - chen schwim-men auf dem See, schwim-men auf dem
6
C
G
C
G7
C
See. Köpf-chen un - ter's Was - ser, Schwänz-chen in die Höh.

In the loop, the beginning of this song would look like this:

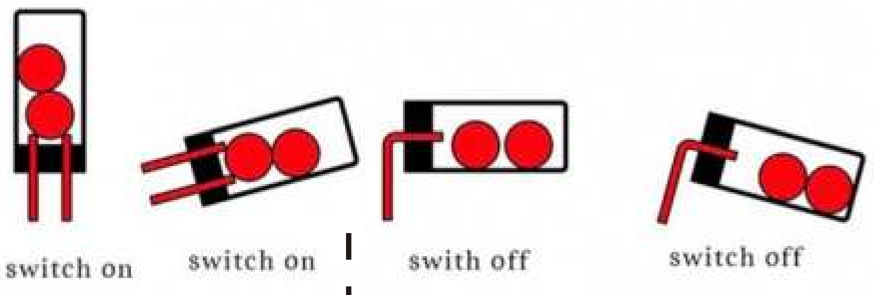
```
tone (8, C4,500);
tone (8, D4,500);
tone (8, E4,500);
tone (8, F4,500);
tone (8, G4, 1000);
tone (8, G4, 1000);
notone (8);                   // The sound turns off
delay (2000);                // for 2 seconds
```

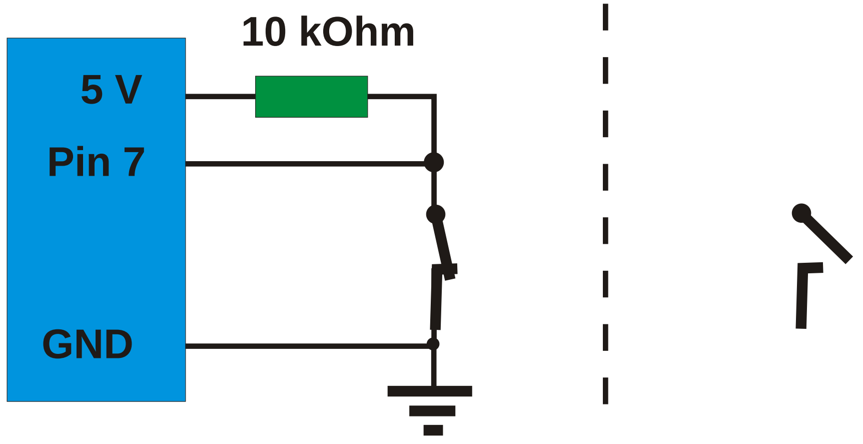
```
#define C3  131
#define CS3 139
#define D3  147
#define DS3 156
#define E3  165
#define F3  175
#define FS3 185
#define G3  196
#define GS3 208
#define A3  220
#define AS3 233
#define B3  247
#define C4  262
#define CS4 277
#define D4  294
#define DS4 311
#define E4  330
#define F4  349
#define FS4 370
#define G4  392
#define GS4 415
#define A4  440
#define AS4 466
#define B4  494
#define C5  523
#define CS5 554
#define D5  587
#define DS5 622
#define E5  659
#define F5  698
#define FS5 740
#define G5  784
#define GS5 831
#define A5  880
#define AS5 932
#define B5  988
#define C6  1047
```

The melody should not play continuously.
With a switch we can do the song
start.
For this we take the ball switch,
who is in the set and switch it
via a 10 kOhm resistor at 5V.
If the ball switch is vertical,
the contact is closed and on
Pin 7 is read in 0V.

If the ball switch tilts,
the contact is opened and on
Pin 7 are now 5V and thus
the song can be started.

In “loop” we read the digital value
at pin 7.
Using the if query
then we decide
that the song starts as soon as
the switch status of pin 7
"High" is





In the setup:

```
int buttonStatus;                    // Status query of the ball switch
pinMode (7, INPUT);                  // pin 7 as input
```

In the loop:

```
buttonStatus = digitalRead (7);      // Query the status of the ball switch
if (buttonStatus == HIGH)            // If switch is open
{
   play the song here
}
```