Institut für Telenautik

Werkstatt Mixed Media / Netzkunst

https://telenautik.de
https://42loop.de/garage
https://code.hfbk.net/42loop

ulf.freyhoff@hfbk-hamburg.de
stud. Helper: Jori Kehn (jori.kehn@googlemail.com)

R240, Lerchenfeld

Intro
into Workshop Mixed Media / Net Art

appointments (also external) on request

- Options:

-- (usually) just walk in, ask questions

-- get complete project support

-- borrow equipment

-- borrow & use tools

-- find materials such as cables/screws/electronics

-- copy sd-cards

-- get programming help or complete programs

Arduino: abstract

'Arduino' describes an Open Source platform for the development of simple controls for electrically driven installations. The software runs on different very inexpensive microcontrollers, that can communicate with a Computer via USB and - on the other hand - can query and control a wide range of sensors (e.g. light, temperature, force, ...) and actors (e.g. light, motors, pumps, magnets, relais, video & audioplayers)

full documentation at https://www.arduino.cc/

but why ? / disclaimer

the name Arduino comes from a bar in Ivrea, Italy, where some of the founders of the project used to meet. The bar was named after Arduin of Ivrea, who was ... and King of Italy from 1002 to 1014.

disclaim / disclaim / disclaim

difference:
’desktop’ computer / microcontroller

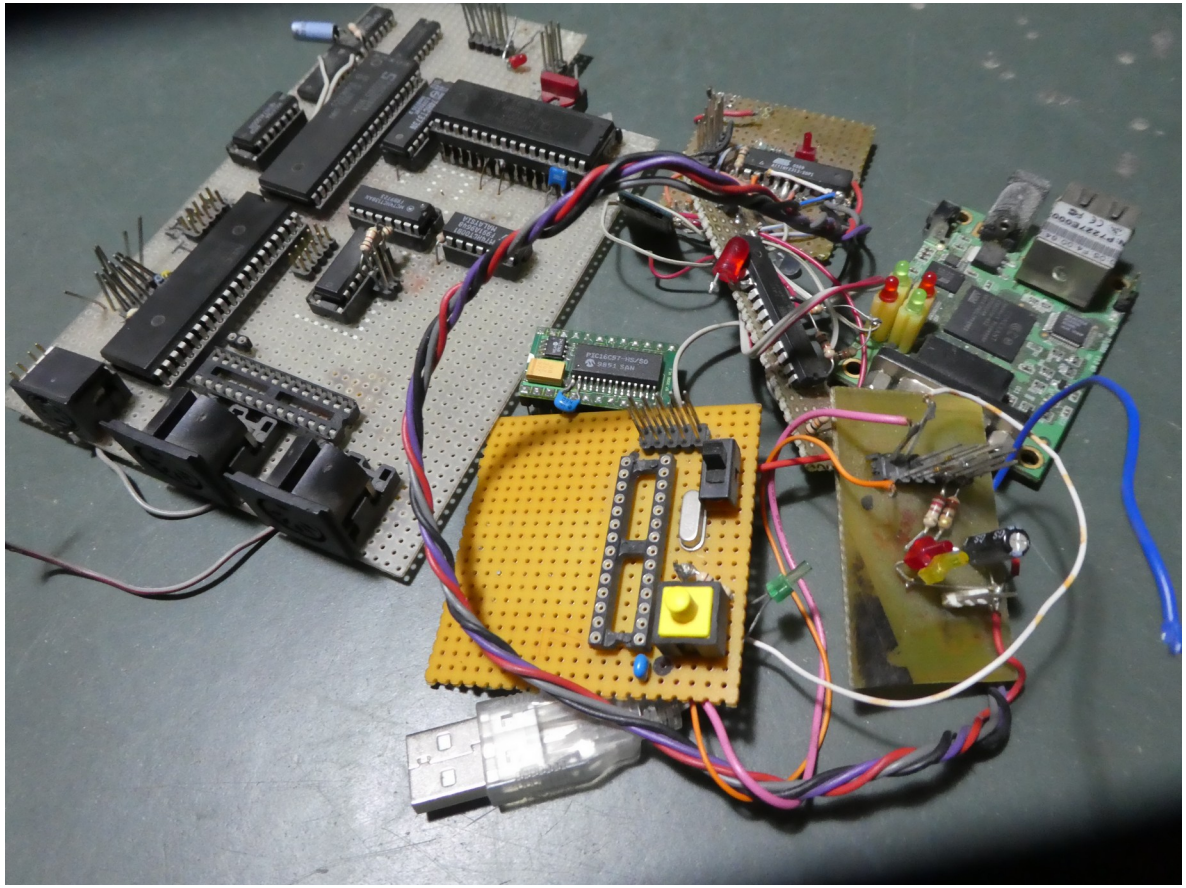
computer:

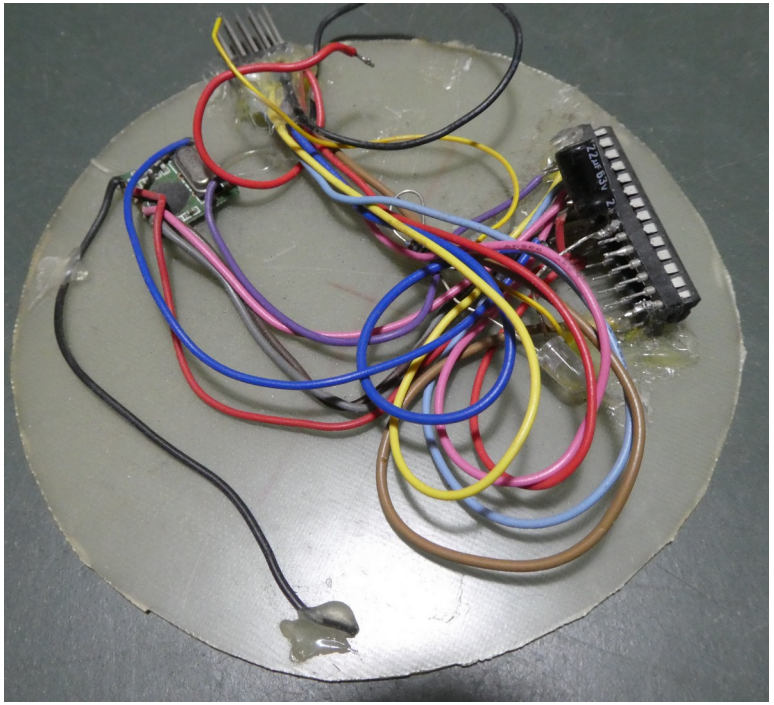
- a complete machine including screen, keyboard, speakers, USB, network, ..., Operating System

microcontroller:

- a computer without Operating System
- a computer that has many physical pins, so all kinds of sensors and actors can be connected
- needs programming interface (mostly USB) & programming
- so called 'arduino' hardware

pre-arduino times





different compatible controllers

legacy: Atmel ATMega328 (datasheet 300 pages)

used on Arduino Uno, Nano, ...

smallest: ATTiny13 (8 pins) digispark

largest: ATMega2560 used on arduino mega

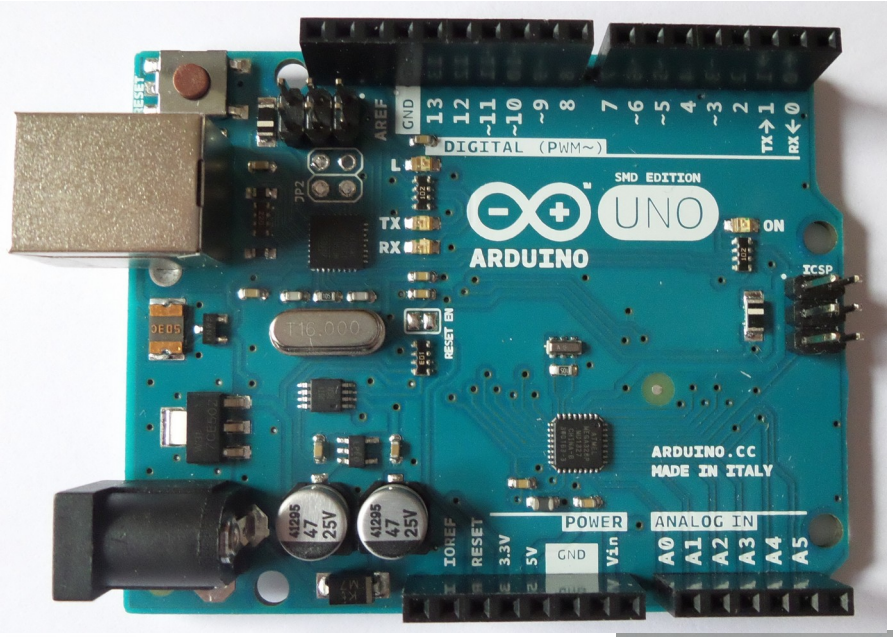
non-Atmel:

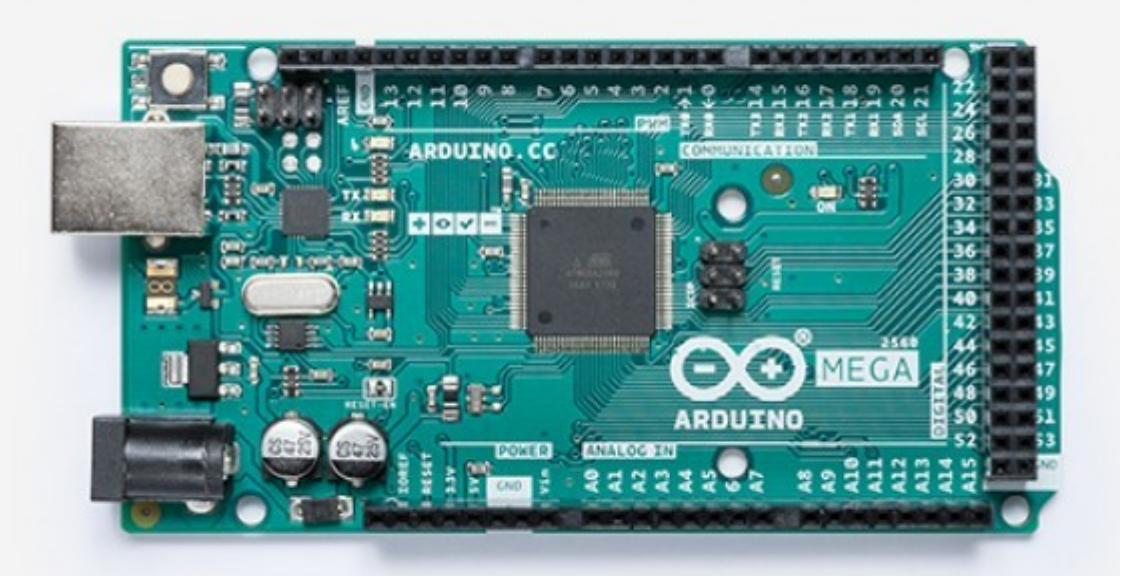
teensy

esp8266 / esp32 & others: wifi / (bluetooth)

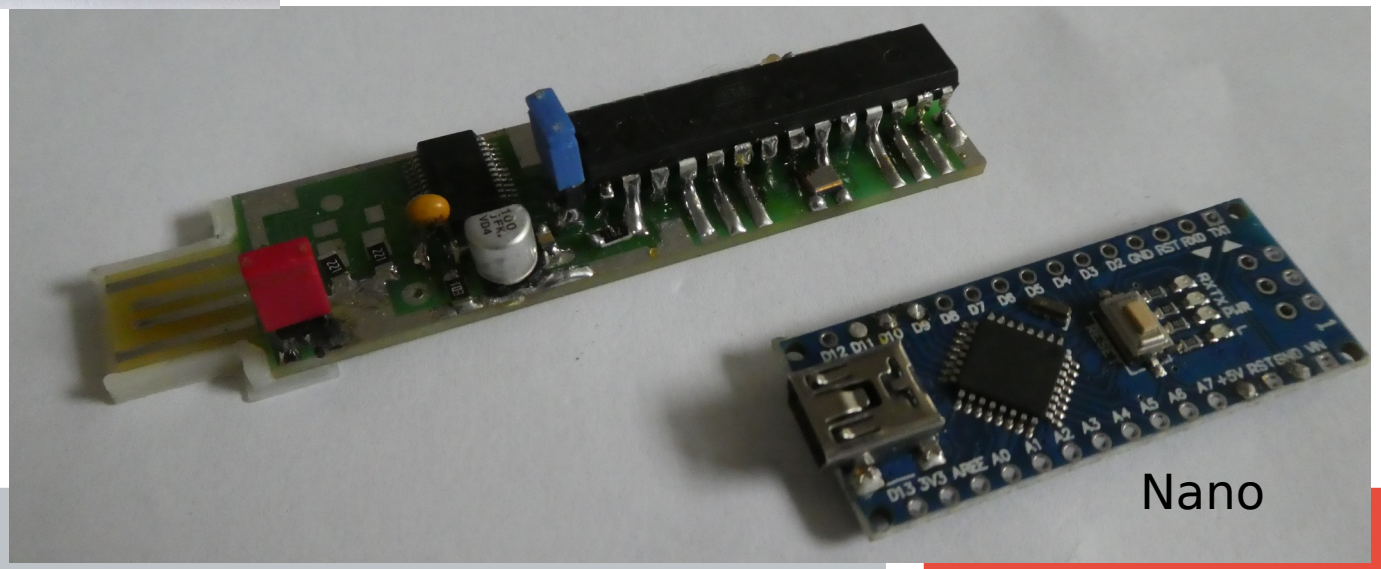
many more

a part of the zoo: (Atmel based)





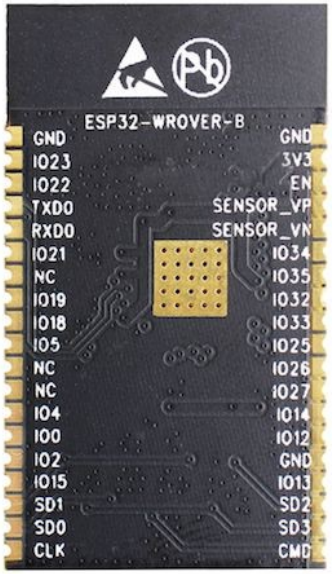
Digispark



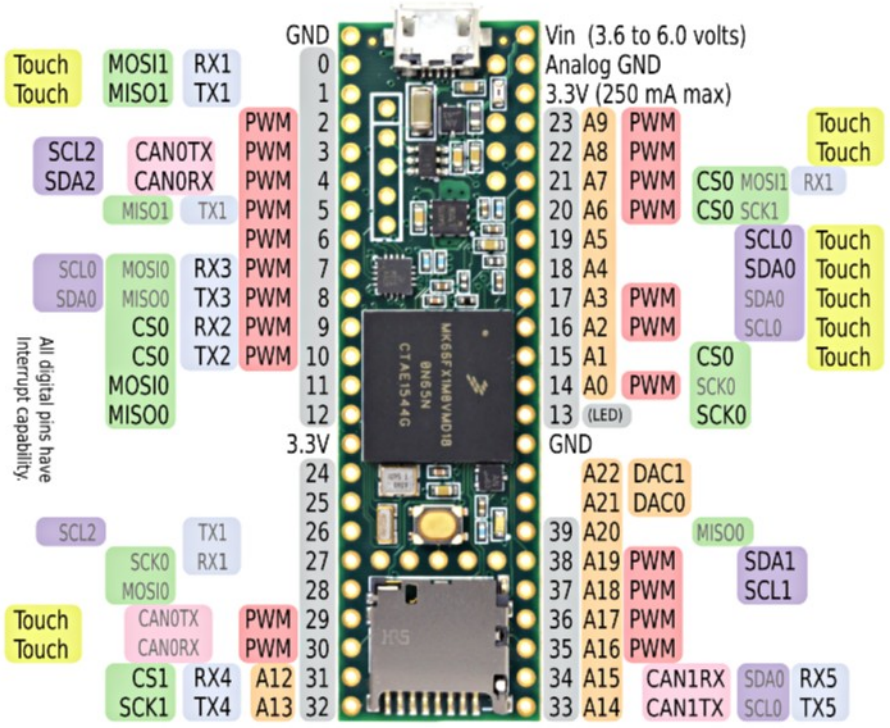
Nano

a part of the zoo: (other controllers)

ESP8266



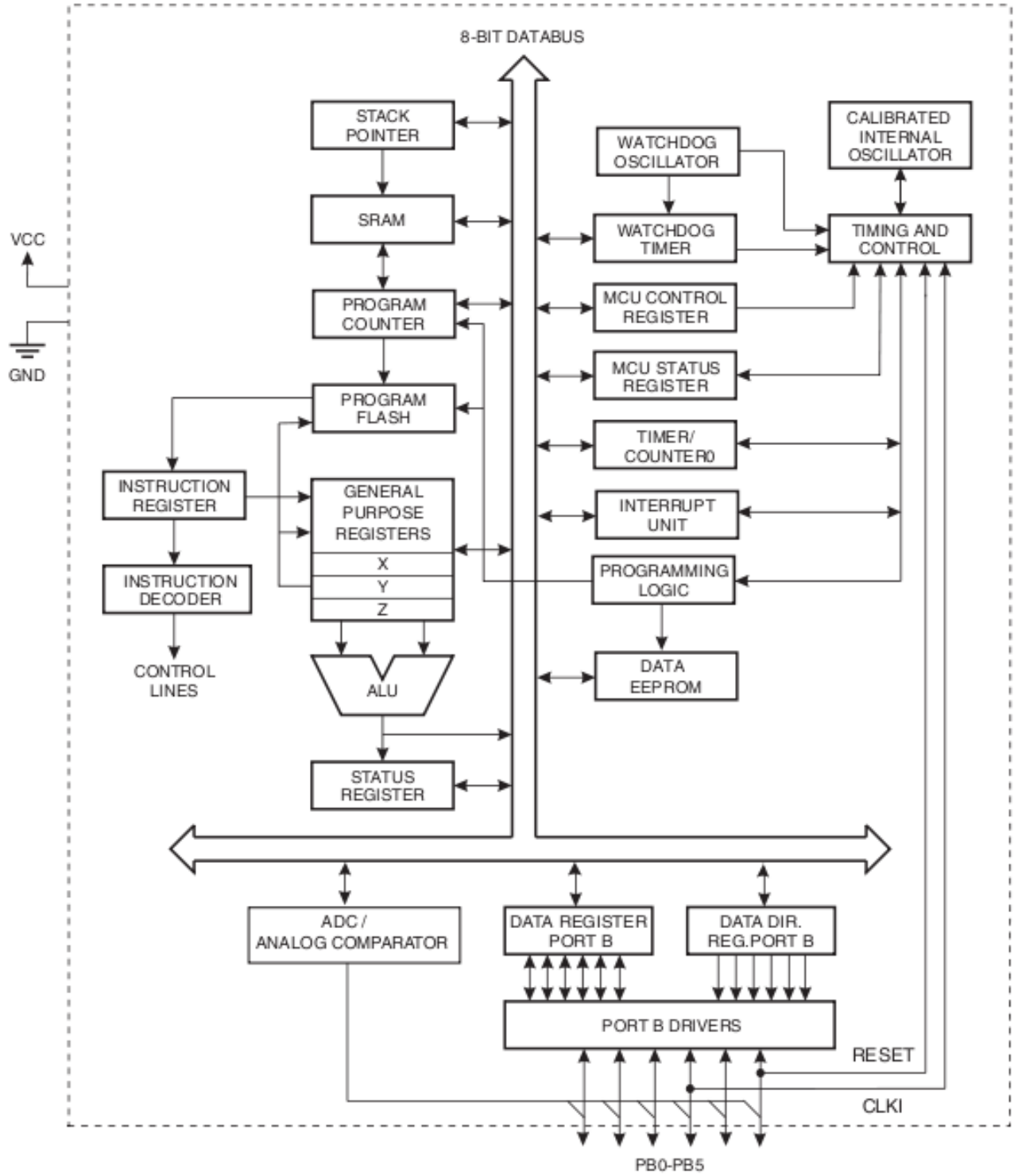
ESP32



Teensy

ATTiny13 internals

Figure 2-1. Block Diagram



power supply

basically: 5V DC

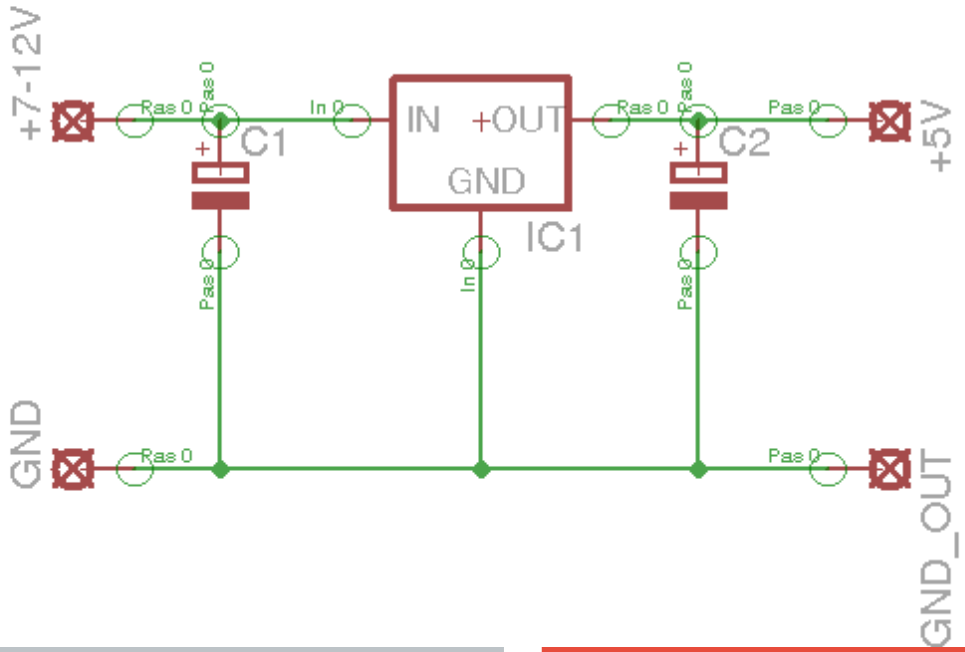
- any usb power supply

- 3x1.5V batteries

- pre-made step-down and step-up converters (wide input range)

- self made with 7805:

(voltage regulator)



The Arduino IDE

IDE: Integrated Development Environment

- Text Editor to write the software

- C++ Compiler to translate it to “binary code” understood by the controller

- Uploader to transfer binary to controller

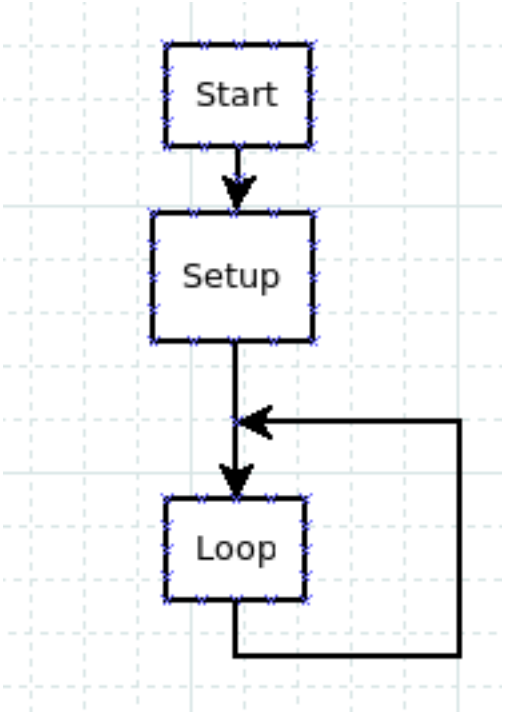
- Serial Terminal to debug via USB

- examples

- help at https://www.arduino.cc/reference/en/

Program Structure

Setup & Loop



Setup / Loop

Setup:

- define Variables: int a=0;

- define GPIO pins:

-- Input / Output

-- digital Inputs / Analog Inputs

-- setup serial connection

-- initialize external hardware

Loop:

do what’s to be done as fast as possible over and over:

- check user input

- check sensors

- step motors / adjust servos / switch digital outputs

- report to user

Blink !

```
/*
  Blink
  Turns an LED on for one second, then off for one second, repeatedly.
  This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// Pin 11 has the LED on Teensy 2.0
// Pin 6  has the LED on Teensy++ 2.0
// Pin 13 has the LED on Teensy 3.0
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

getting feedback: Serial

```
String inputString = "";         // a String to hold incoming data
bool stringComplete = false;  // whether the string is complete

void setup() {
  // initialize serial:
  Serial.begin(115200);
  // reserve 200 bytes for the inputString:
  inputString.reserve(200);
}

void loop() {
  if (stringComplete) {
    Serial.println(inputString);
    inputString = "";
    stringComplete = false;
  }
}

void serialEvent() {
  while (Serial.available()) {
    // get the new byte:
    char inChar = (char)Serial.read();
    // add it to the inputString:
    inputString += inChar;
    if (inChar == '\n') {
      stringComplete = true;
    }
  }
}
```

/*
  SerialEvent occurs whenever a new data comes in the hardware serial RX. This routine is run between each time loop() runs, so using delay inside loop can delay response. Multiple bytes of data may be available.
*/

input & output

code, digital:

```
#define myoutpin 13
#define myinpin 10
bool state=false;
pinMode(myoutpin,OUTPUT);
digitalWrite(myoutpin,HIGH);
delay(1000);
digitalWrite(myoutpin,LOW);
```

```
pinMode(myinpin,INPUT);
state=digitalRead(myinpin);
if (state) {do_something();}
```

input & output

code, analog:

```
#define myanaloginpin A0
#define myanalogoutpin 8
//no initialization necessary
int number=0;
number=analogRead(myanaloginpin);
do_something(number);
```

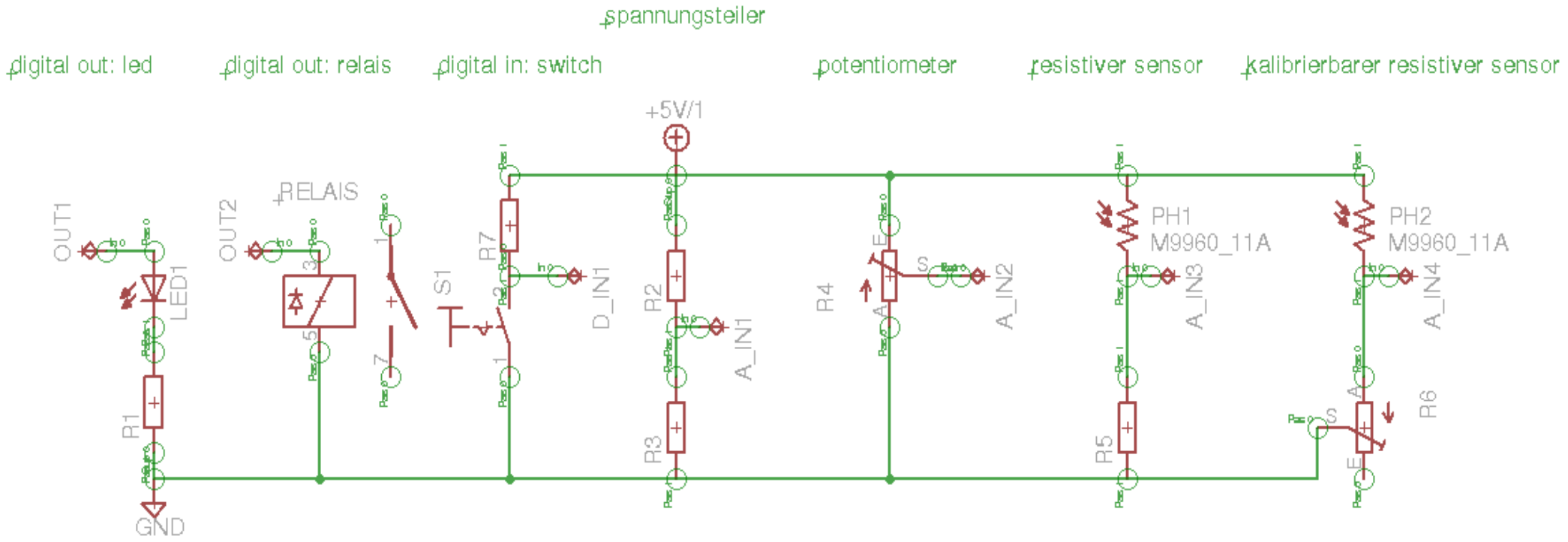
```
analogWrite(myanalogoutpin,number);
```

note: PWM - Pulse Width Modulation

Simulation

https://wokwi.com/

simple input & output: physical



advanced input / output

PIR:HC-SR501

distance sensors:

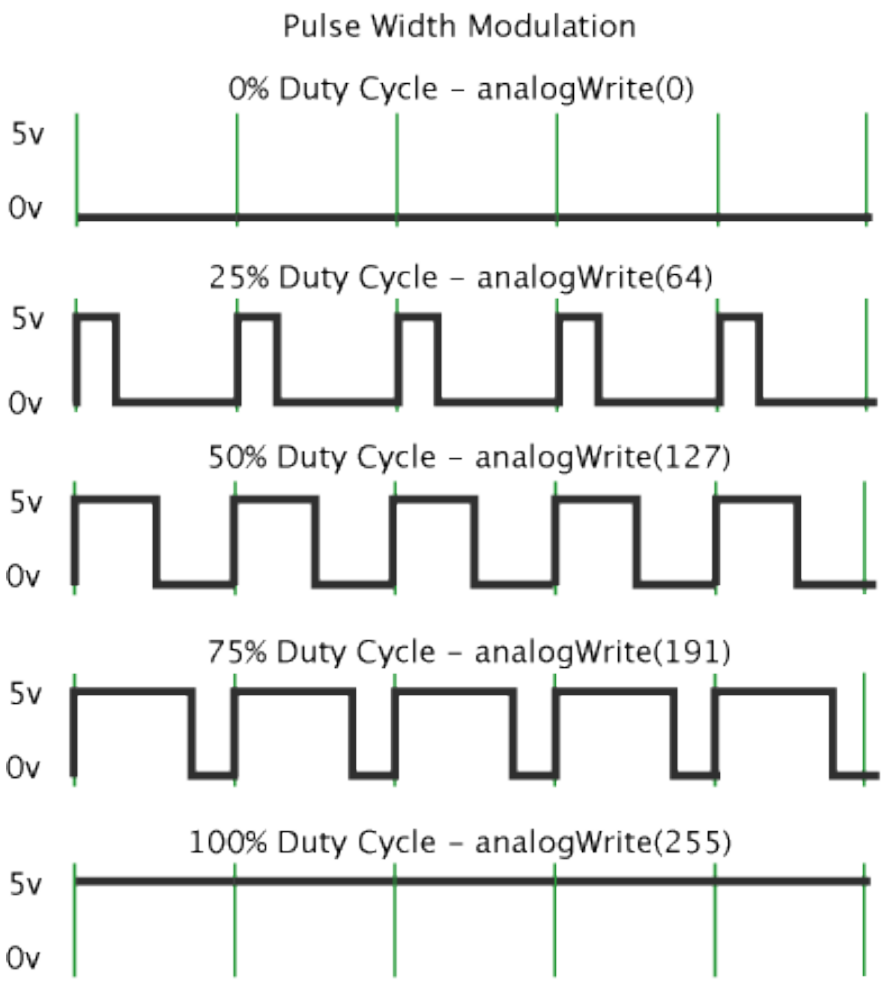
- **infrared (zb sharp gp2d12) (small angle, number)**
- **ultrasonic (zb srf04)**
- **PIR motion sensors (wide angle, digital)**

Servo

motor control:

- **dc motors / stepper motors**
- **pwm / dimming**
- **h-bridge**
- **transistor drivers zb uln2803**
- **servo**

PWM


Pulse Width Modulation
0% Duty Cycle – analogWrite(0)
5v
0v
25% Duty Cycle – analogWrite(64)
5v
0v
50% Duty Cycle – analogWrite(127)
5v
0v
75% Duty Cycle – analogWrite(191)
5v
0v
100% Duty Cycle – analogWrite(255)
5v
0v