

repair

STADTBIBLIOTHEK
DITZINGEN

Entsorgen? Nein, danke!

café

4. Ditzinger Makerspace Arduino Teil 1

Ein Projekt von Repair-Café und Stadtbücherei
Ditzingen

08.07.2017 Stadtbücherei Ditzingen UG

Agenda

- Was ist ein Arduino?
- Arduino – Raspberry Vergleich
- Arduino Hardware
- Entwicklungsumgebung für den Arduino
- Softwareentwicklung für den Arduino
- Beispielprojekte

Was ist ein Arduino?

- ... ein Mikroprozessor ...
 - mit flexibel programmierbaren Ein- und Ausgängen
 - mit einem USB-Port über den er programmiert werden kann
- Es gibt >10 verschiedene Typen ...

Was ist ein Arduino?

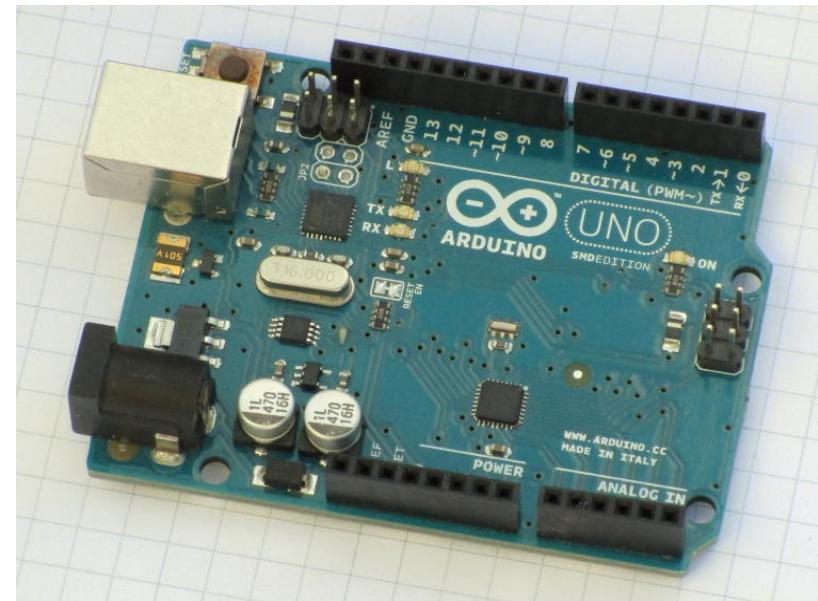
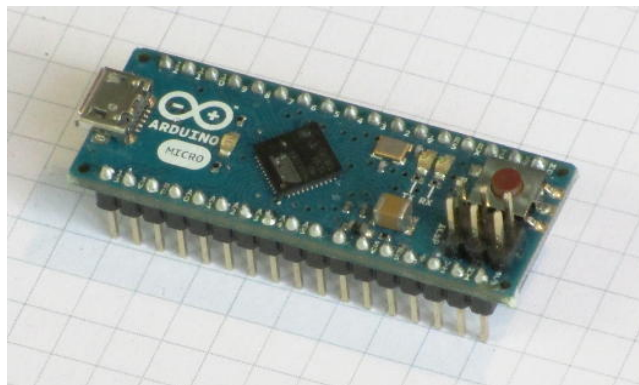
- Die Arduinos wurden in Italien zur Ausbildung von Studenten entwickelt
- Es gibt daher ein begleitendes Tutorial und Dokumentation für Programmieranfänger
- Breite Community / Foren zur Unterstützung
- Die Hardware ist Open Source ([L]GPL) → darf also legal kopiert werden

Was unterscheidet Raspberry Pi und Arduino?

Raspberry Pi	Arduino
Vollwertiger Universal Computer	Mikroprozessor / Mikro-Controller
Bereit für Bildschirm, Tastatur, Maus, Speichermedium	„Stand-alone“-Controller, komplexes User-I/O nicht vorgesehen
Vollwertiges Multi-User- / Multitasking-Betriebssystem	Kein Betriebssystem → „Main Loop“
Kann von sich selber programmiert werden	Benötigt Host-Computer zur Programmierung
Keine bzw. „weiche“ Echtzeit	„Harte“ Echtzeit → präzises Timing
Komplexe Umgebung	Abläufe sind einfach

Arduino Hardware

- Beispiele:
 - **Arduino Nano** & **Arduino Uno:**
 - 32KB Flash-Rom (→ einige tausend C-Statements)
 - 2KB RAM
 - 1KB EEPROM
 - 16 MHz
 - 3 Timer (2x8bit/1x16bit)
 - 5 10bit ADC

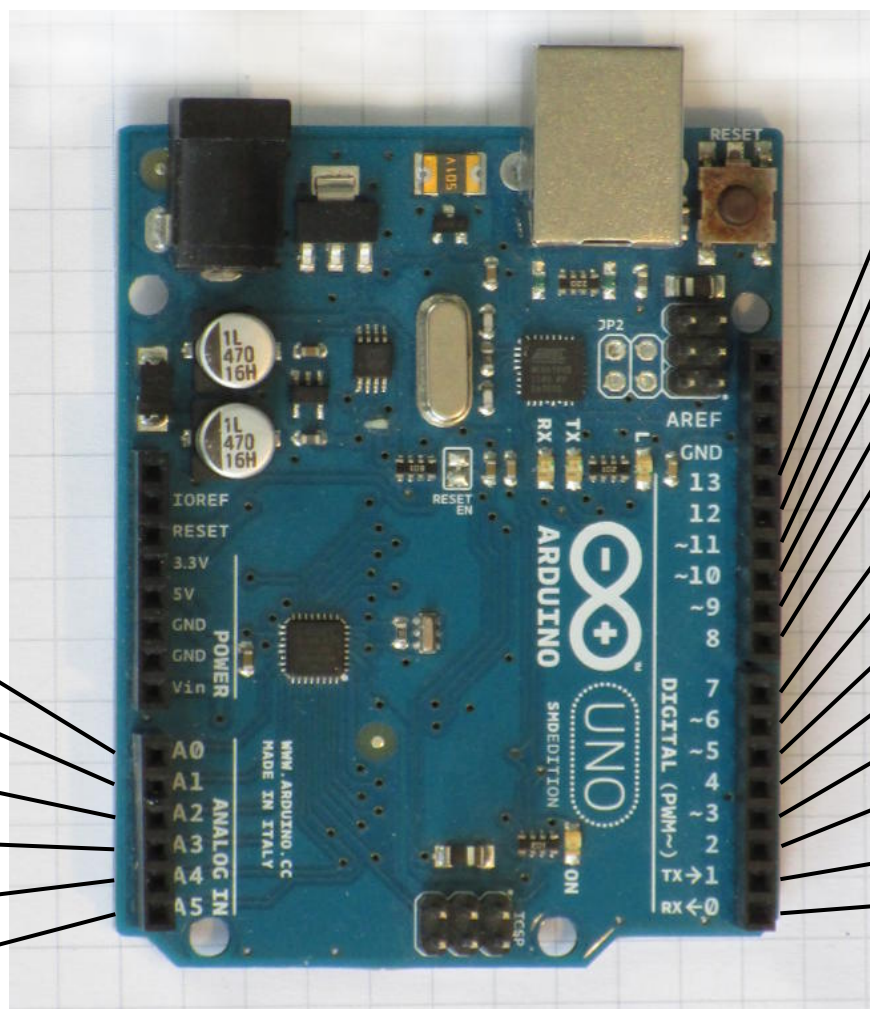


Arduino I/O Beispiel Uno

- I/O Pins können sehr flexibel genutzt werden

Alle Digital-I/O-Pins
können auch als
Change-Interrupt
genutzt werden

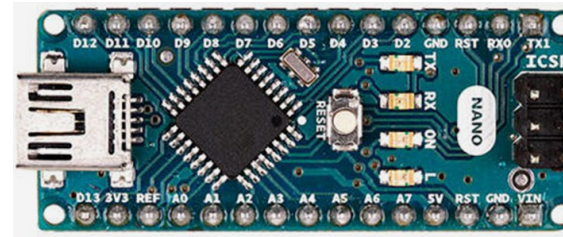
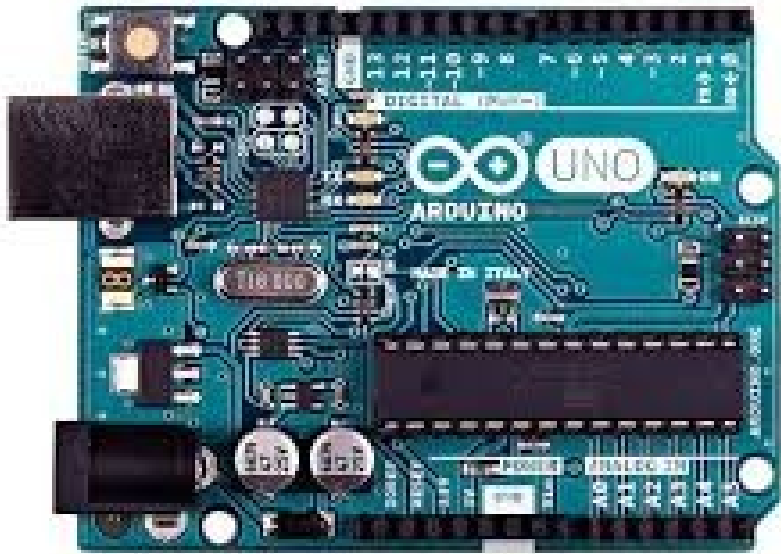
ADC5 / PC5
ADC4 / PC4
ADC3 / PC3
ADC2 / PC2
ADC1 / PC1
ADC0 / PC0



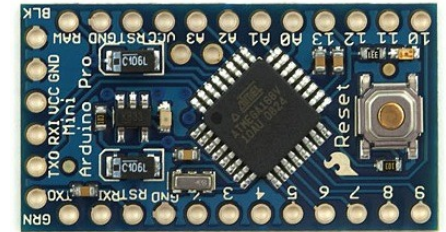
PB5 / SPI-SCK
PB4 / SPI-MISO
PB3 / SPI-MOSI
PB2 / OC1B / SPI-SS
PB1 / OC1A
PB0 / ICP1

PD7 / AIN1
PD6 / AIN0 / OC0A
PD5 / T1 / OC0B
PD4 / T0
PD3 / INT1 / OC2B
PD2 / INT0
PD1 / TXD
PD0 / RXD

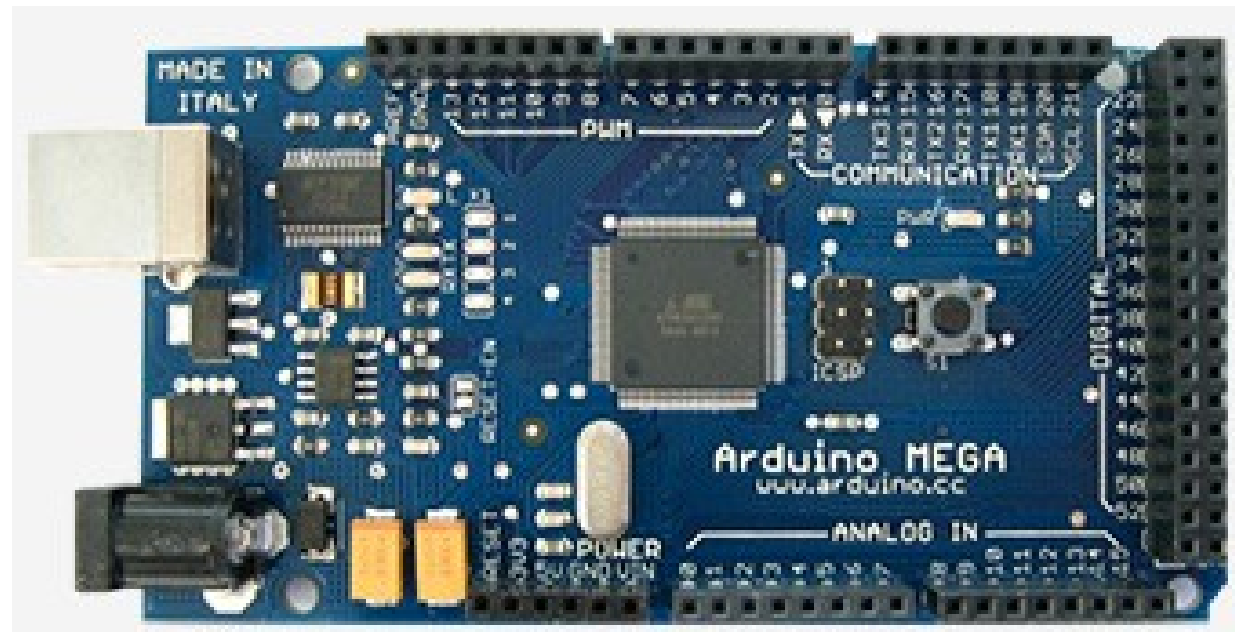
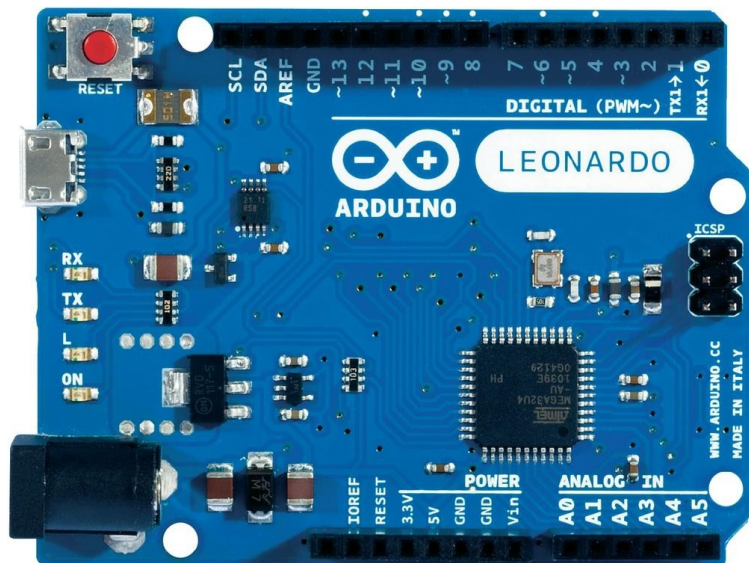
Verschiedene Arduino Boards



Arduino NANO

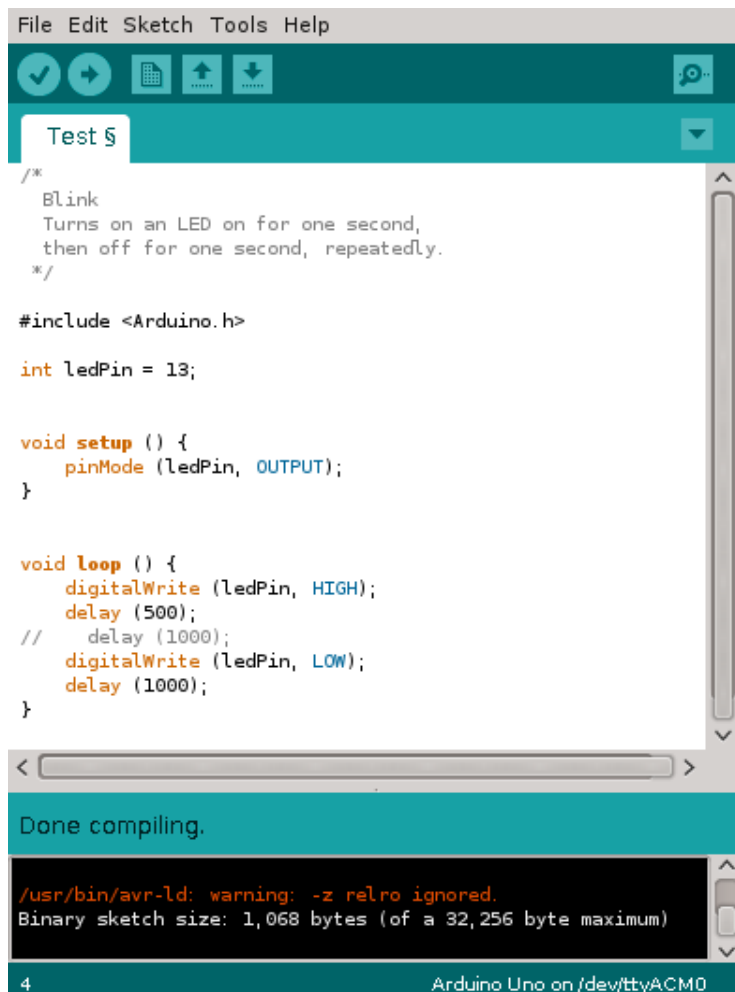


Arduino MINI



Wie wird ein Arduino programmiert?

- Arduino IDE (Integrierte Entwicklungsumgebung)



The screenshot shows the Arduino IDE interface. The menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar contains icons for opening, saving, and running. The sketch editor displays a C++ program for a blinking LED. The status bar at the bottom indicates 'Done compiling.' and 'Binary sketch size: 1,068 bytes (of a 32,256 byte maximum)'. The target board is set to 'Arduino Uno on /dev/ttyACM0'.

```
File Edit Sketch Tools Help

Test $

/*
  Blink
  Turns on an LED on for one second,
  then off for one second, repeatedly.
*/

#include <Arduino.h>

int ledPin = 13;

void setup () {
  pinMode (ledPin, OUTPUT);
}

void loop () {
  digitalWrite (ledPin, HIGH);
  delay (500);
  // delay (1000);
  digitalWrite (ledPin, LOW);
  delay (1000);
}

Done compiling.

/usr/bin/avr-ld: warning: -z relro ignored.
Binary sketch size: 1,068 bytes (of a 32,256 byte maximum)

4 Arduino Uno on /dev/ttyACM0
```

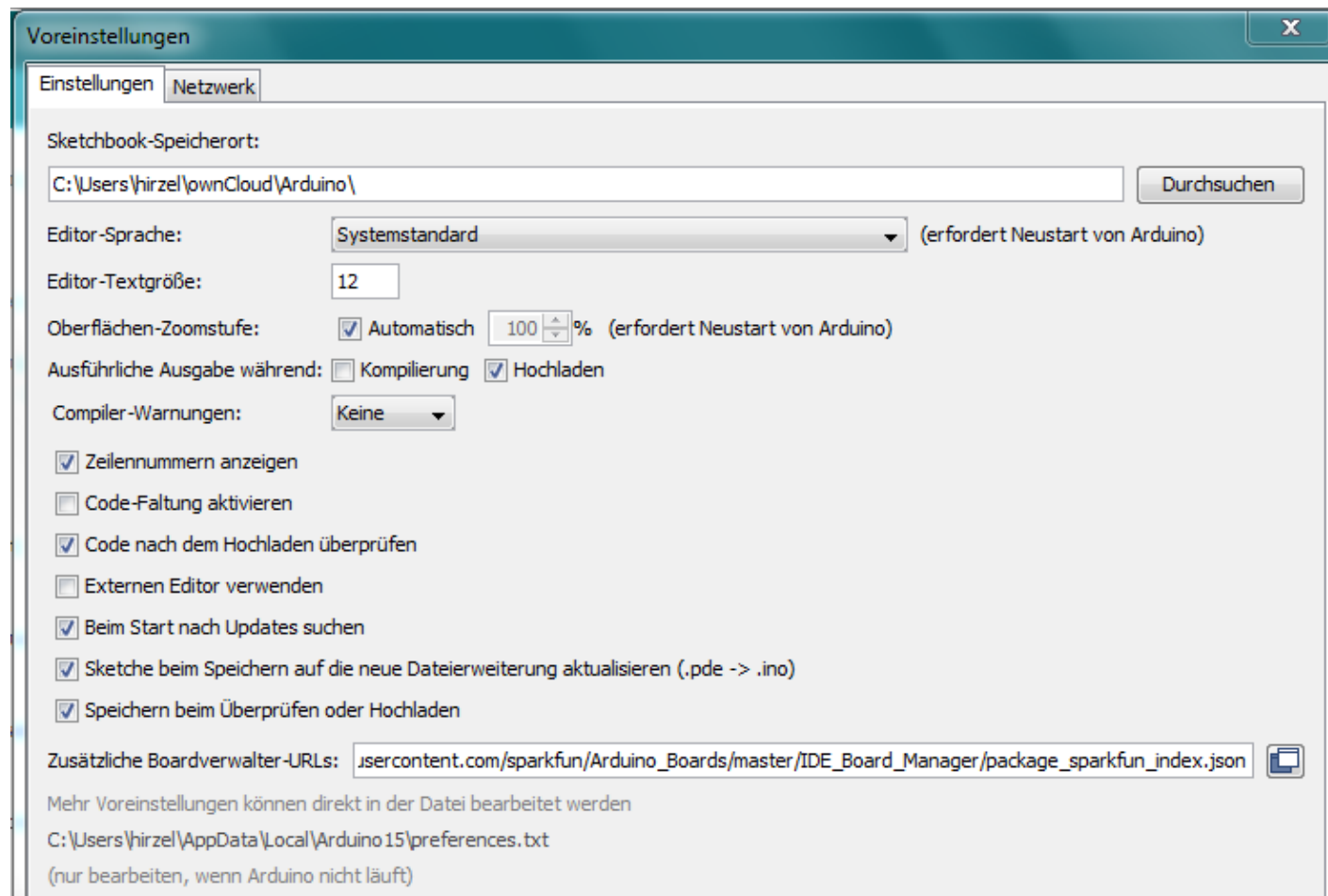
- Editieren des Quellcodes
- Kompilieren
- Auf den Arduino laden
- Anzeige der Arduino-Ausgabe
- Detaillierte Anleitung:
 - Leichter Einstieg
- GNU-Compiler:
 - Voller C++-Umfang für Experten
- Für Linux und Windows

Arduino IDE

- Für die Softwareentwicklung verwendet man die Arduino IDE (integrated development environment)
- Die IDE kann man sich kostenlos unter www.arduino.cc → Software für verschiedene Betriebssysteme herunterladen. Aktuell, Stand 7/2017, ist die Version 1.8.3
- Mit der IDE werden viele Bibliotheken (Libraries) und Beispiele (Examples) installiert.

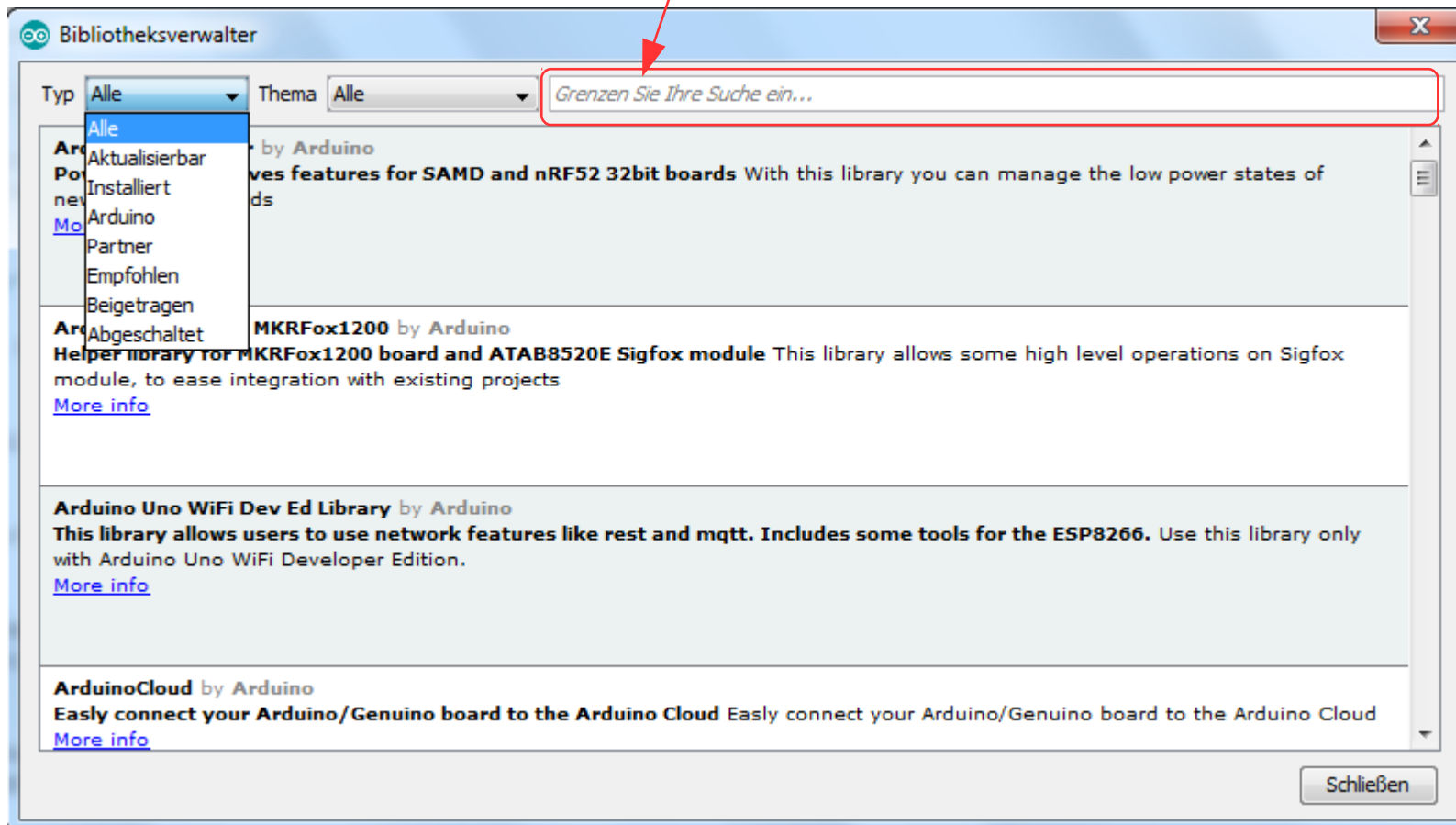
IDE Konfiguration

- Nach der Installation ist es sinnvoll einen kurzen Blick auf die Konfiguration unter Datei → Voreinstellungen zu werfen



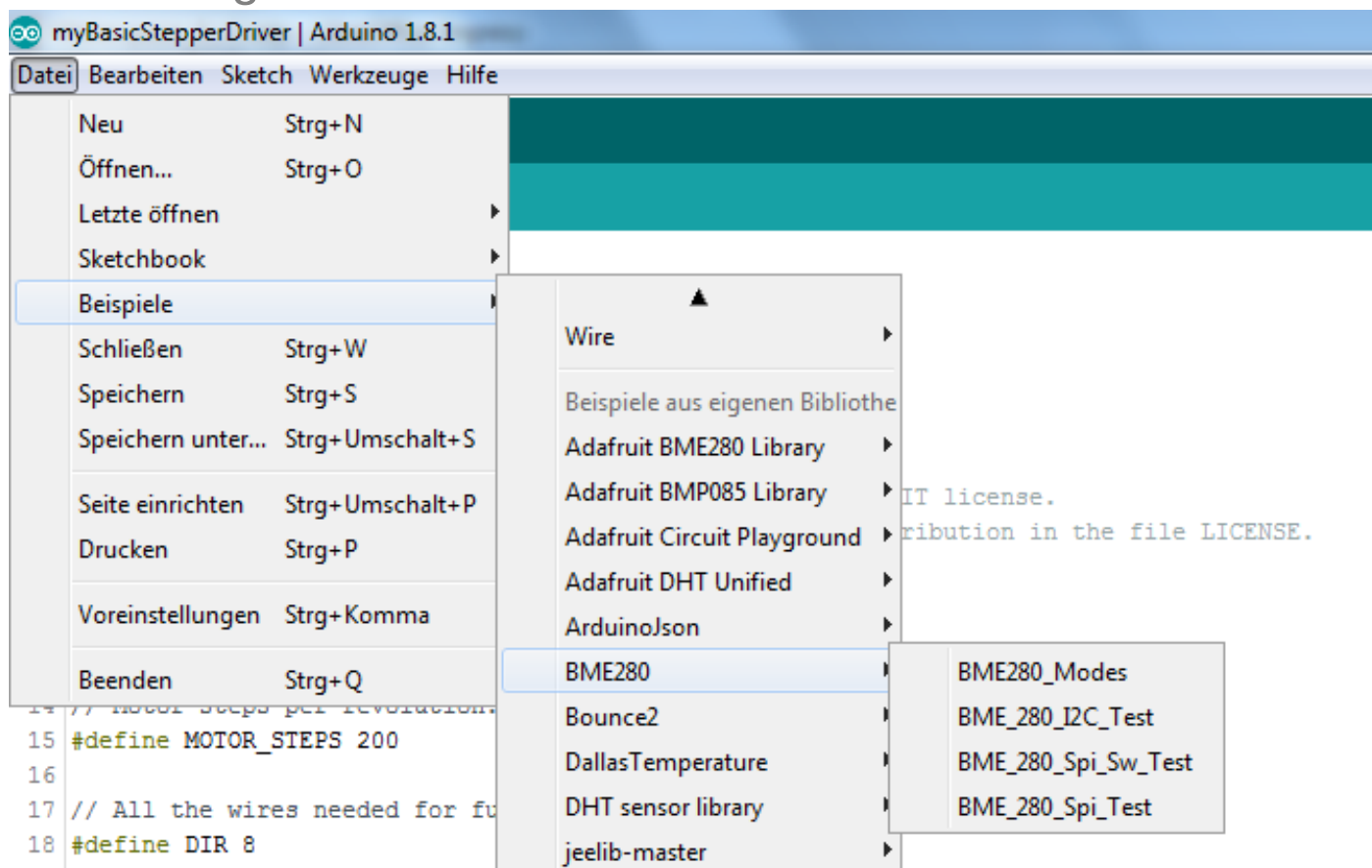
Zusätzliche Bibliotheken installieren (2)

- der Bibliotheksverwalter wird in einem eigenen Fenster angezeigt
- Bibliotheken können mit Stichworten gesucht werden



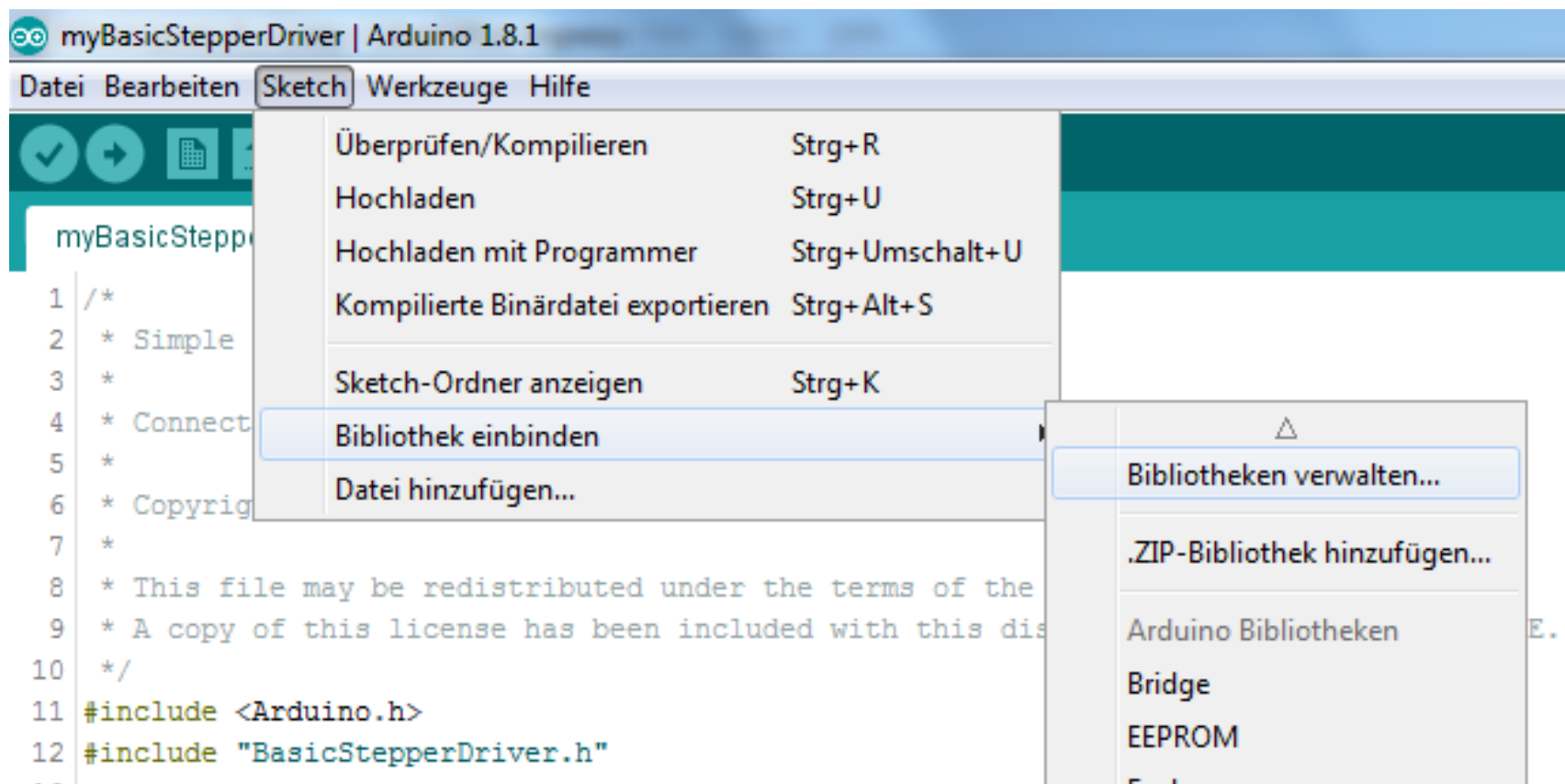
Zusätzliche Bibliotheken installieren (3)

- mit den Bibliotheken werden meistens auch Beispiele mit installiert, die die Anwendung der Bibliothek zeigen und die man für eigene Anwendungen abändern kann. Dazu speichert man das Beispiel unter neuem Namen in seinem eigenen Sketch Bereich ab.



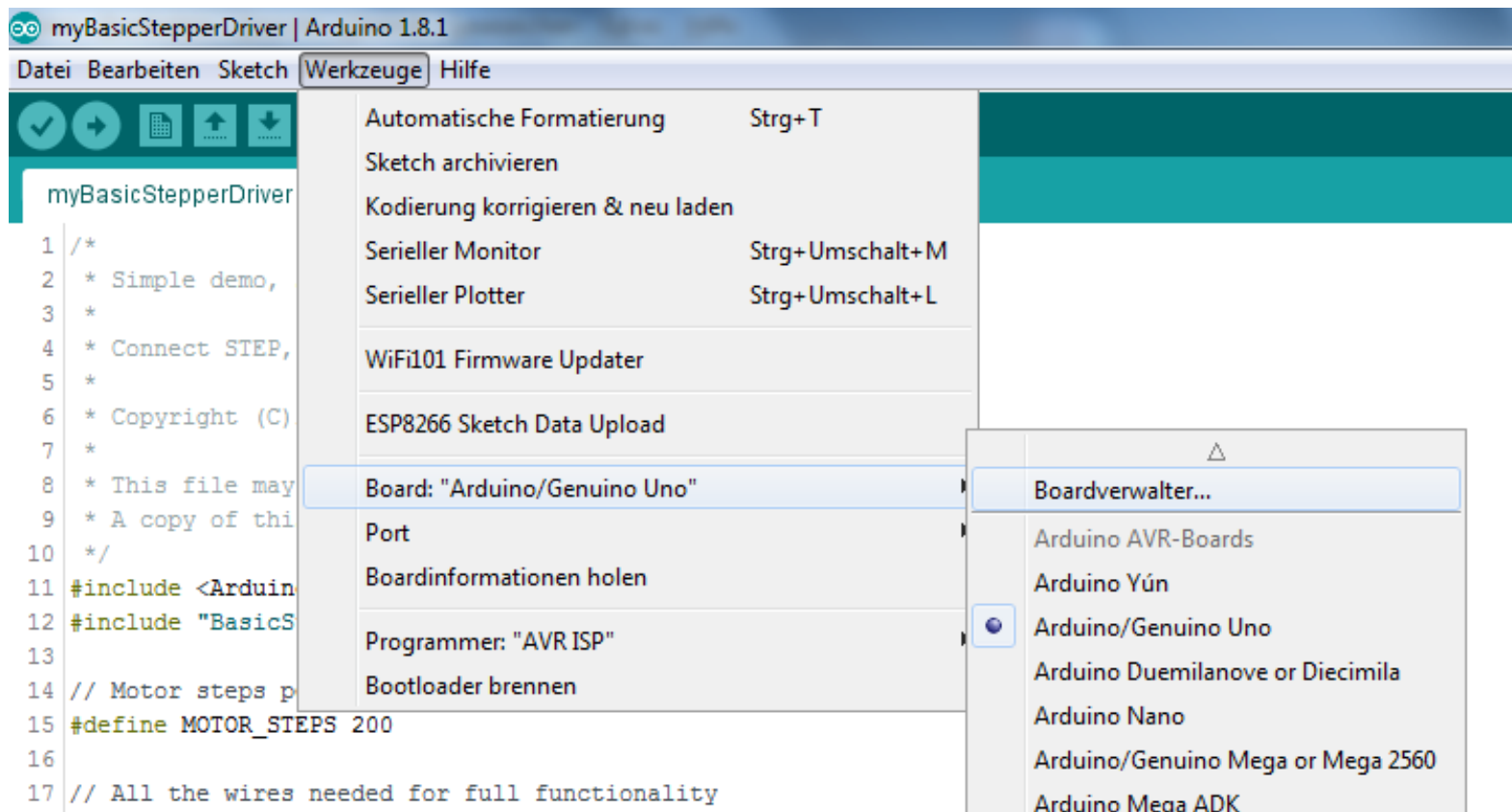
Zusätzliche Bibliotheken installieren (1)

- die mit der IDE installierten Bibliotheken können erweitert werden



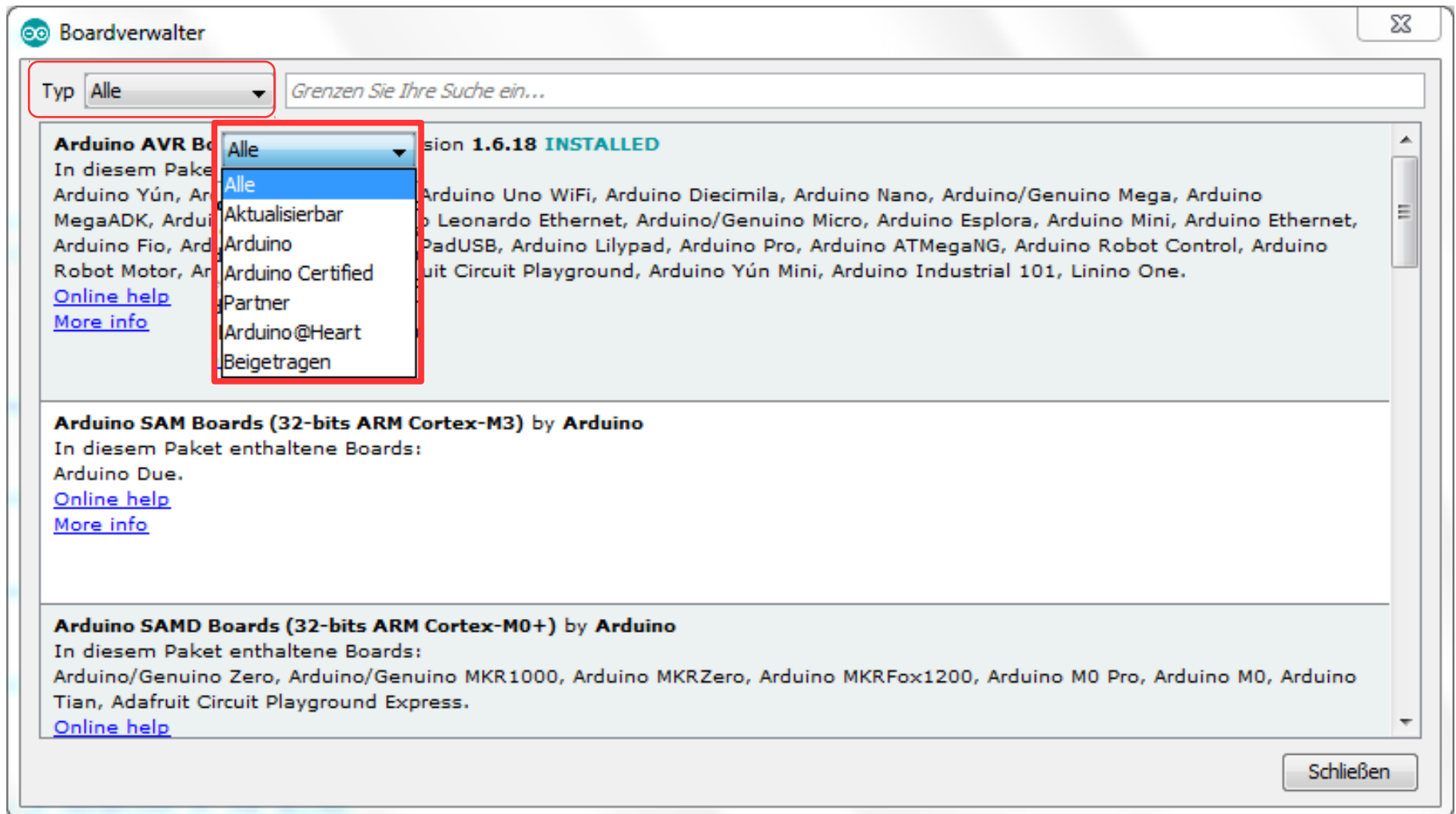
Zusätzliche 'Boards' installieren (1)

- Standardmäßig werden mit der IDE die Beschreibungen für die Standard Arduino Boards installiert.
- Die in der IDE auswählbaren Boards können über den Board Manager erweitert werden.



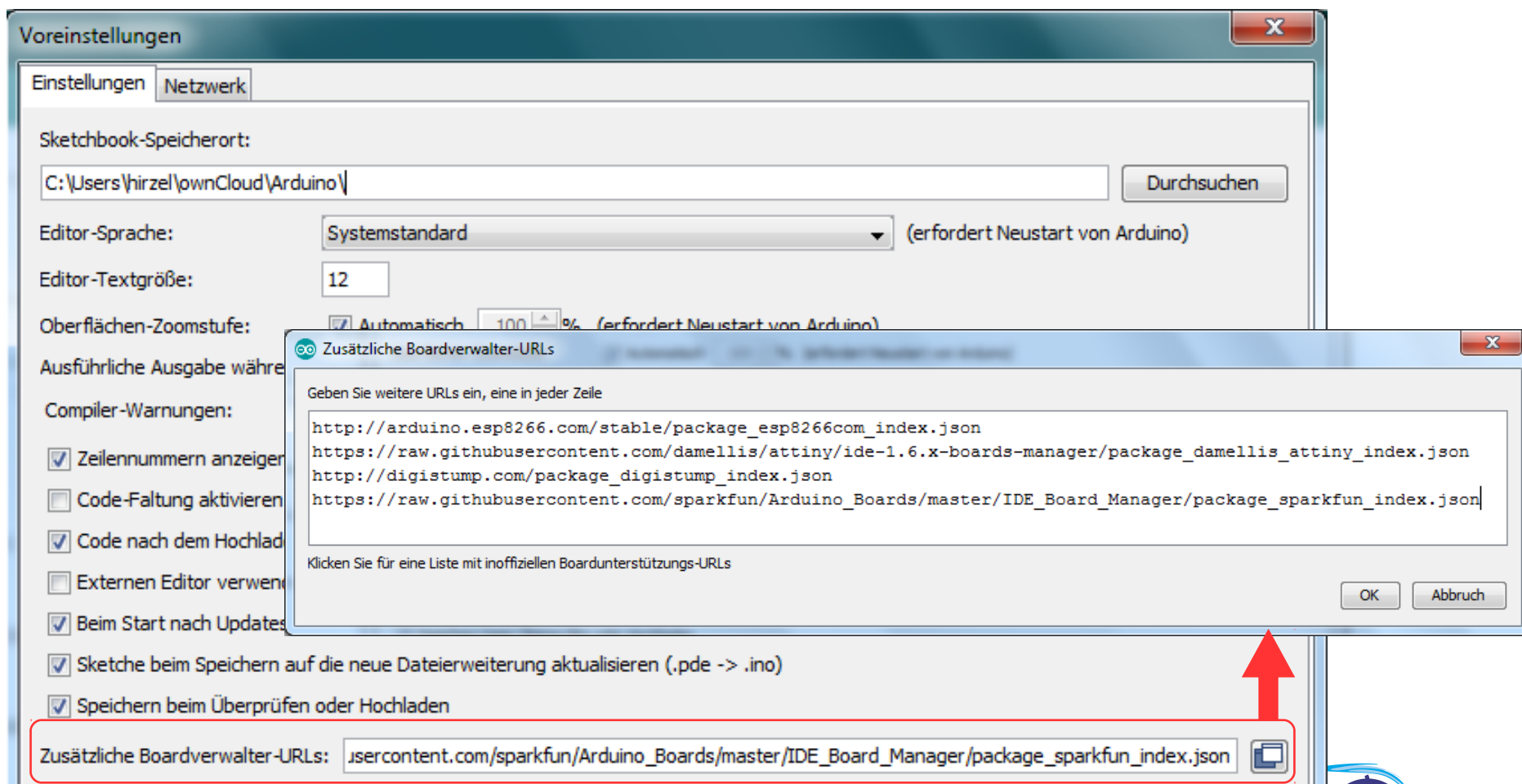
Zusätzliche 'Boards' installieren (1)

- Nach Klick auf 'Boardverwalter' öffnet sich folgendes Menü

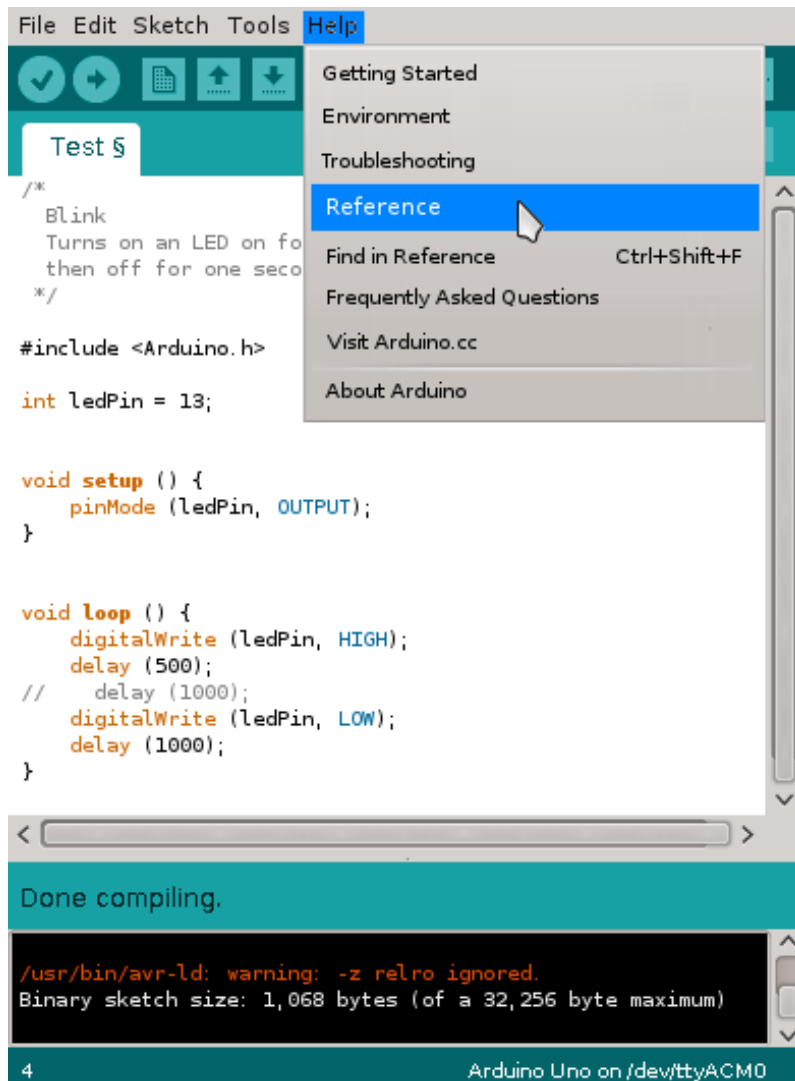


Zusätzliche 'Boards' installieren (3)

- Zusätzliche Quellen für Boards müssen unter Voreinstellungen hinzugefügt werden



Hilfe Arduino-Referenz



- Basisinformationen über die Programmiersprache
- Informationen über die Arduino-Funktionsbibliothek

Beispiel mit Arduino-Library

Digital I/O mit Serial Interface

```
#include <Arduino.h>
#include <stdint.h>

static const int ledPin = 13;

void setup () {
    pinMode (ledPin, OUTPUT);
    Serial.begin (115200);
}

void loop () {
    static uint32_t msCounter1;
    static uint8_t ledState;
    uint32_t msCounter;

    msCounter = millis ();

    if (msCounter - msCounter1 > 500) {
        ledState ^= 1;

        digitalWrite (ledPin, ledState);

        Serial.print (msCounter, HEX);
        Serial.print (' ');
        Serial.println (ledState);

        msCounter1 += 500;
    }
}
```

Beispiel ohne Library Zählerprogrammierung

- Die Implementierung der `millis()`-Routine ist ungenau (Abweichung > 1min/d)
- Quarz-genauer Ersatz ist aber nicht schwierig:

```
static void setTimer () {
    // Timer0:
    TCCR0A = (1<<WGM01);           // Normal port operation, mode:CTC
    TCCR0B = (1<<CS01) | (1<<CS00); // Clock source clkI/O, divider 64, mode:CTC
    OCR0A  = 250;                  // Timer 0 TOP value = F_CPU/64/1000
    TIMSK0 = (1<<OCIE0A);          // Enable Timer0 Compare Match A Interrupt
}

static volatile unsigned long msCounter;

ISR(TIMER0_COMPA_vect) {
    ++msCounter;
}

uint32_t millisExact () {
    unsigned long msCounterLoc;
    cli();                      // Disable interrupts
    msCounterLoc = msCounter;
    sei();                      // Enable interrupts
    return msCounterLoc;
}
```

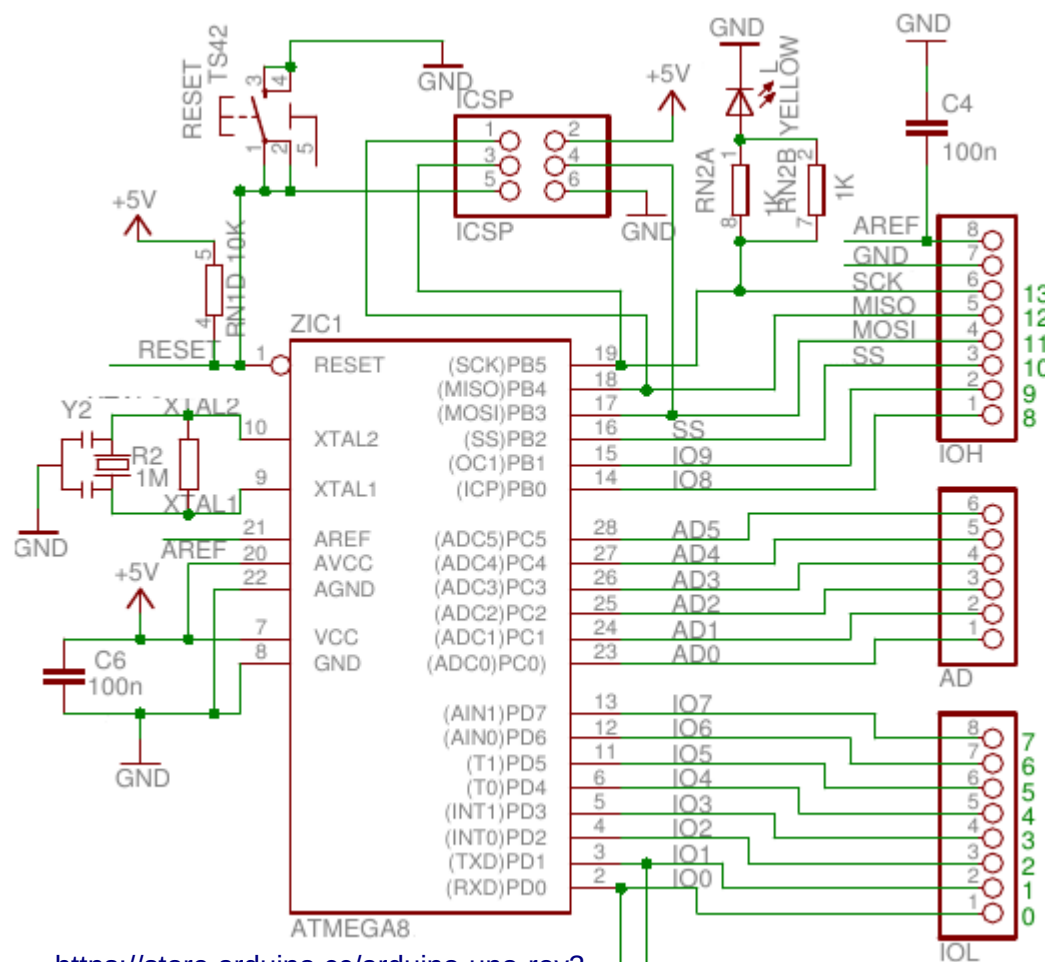
Achtung: Nicht mehr portierbar!

Timer0 wird von den Library-Funktionen `millis()`, `micros()`, `delay()` und `delayMicroseconds()` verwendet. Bei Verwendung von Timer0 funktionieren diese nicht mehr. In ähnlicher Weise kann statt Timer0 aber auch Timer2 verwendet werden.

- Atmel ATmega328
<http://www.microchip.com/wwwproducts/en/ATMEGA328>
- Features
- Pin-Beschreibung
- Funktionsbeschreibung inkl. der betreffenden Register
- Electrical Characteristics
 - mit Absolute Maximum Ratings
- Packaging Information

Arduino-Schaltplan

Schaltplan Arduino Uno* (Ausschnitt)



- Controller-Pins sind direkt mit den Board-Pins verbunden
- Achtung: alle Anschlüsse sind ungeschützt gegen Kurzschluss oder Überspannung!

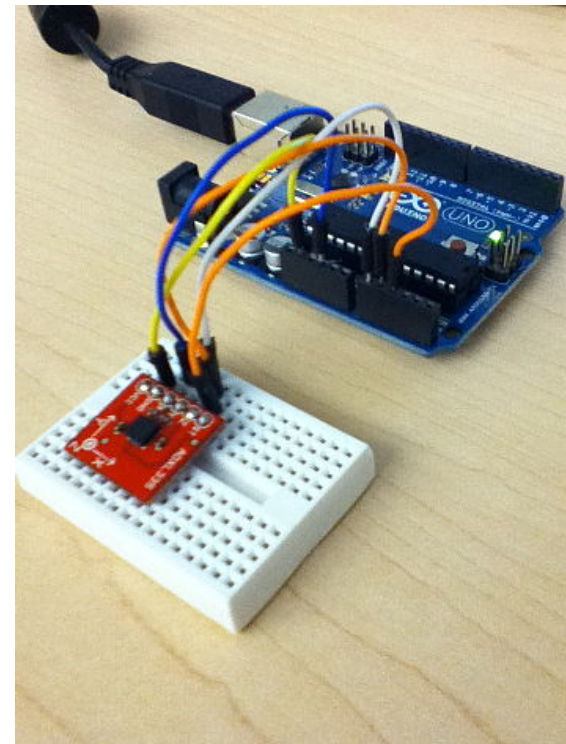
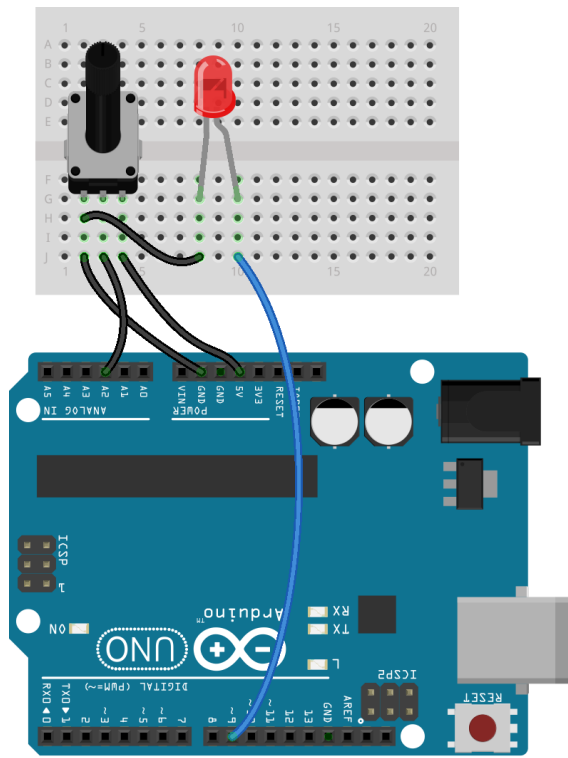
<https://store.arduino.cc/arduino-uno-rev3>
* Creative Commons Attribution ShareAlike 3.0

Welche Projekte kann ich mit einem Arduino realisieren?

- Es gibt viele tausend Beispiel-Projekte im Arduino „Playground“:
 - Intelligente Sensoren
 - Motorsteuerungen
 - Alle Arten von sonstigen Steuerungen und Regelungen
 - Roboter-Anwendungen
 - Signalwandler
 - LCD-Display-Ansteuerungen
 - Elektronische Musikinstrumente
 - USW. USW.

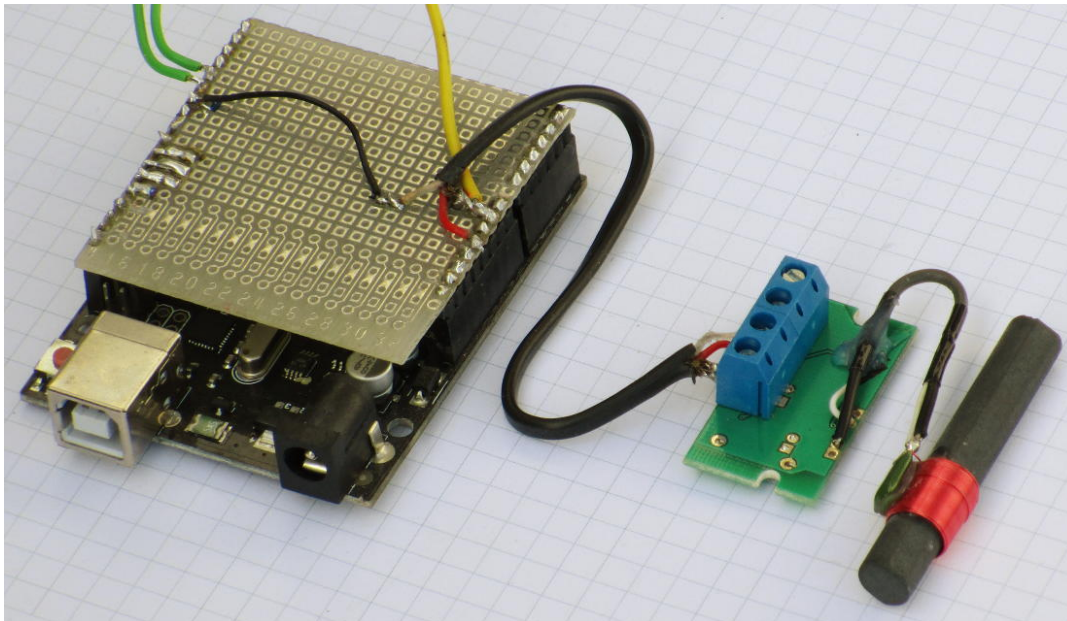
Hardware-Ankopplung (1)

- Zum Experimentieren wird HW über Steckkabel mit einem Breadboard verbunden:



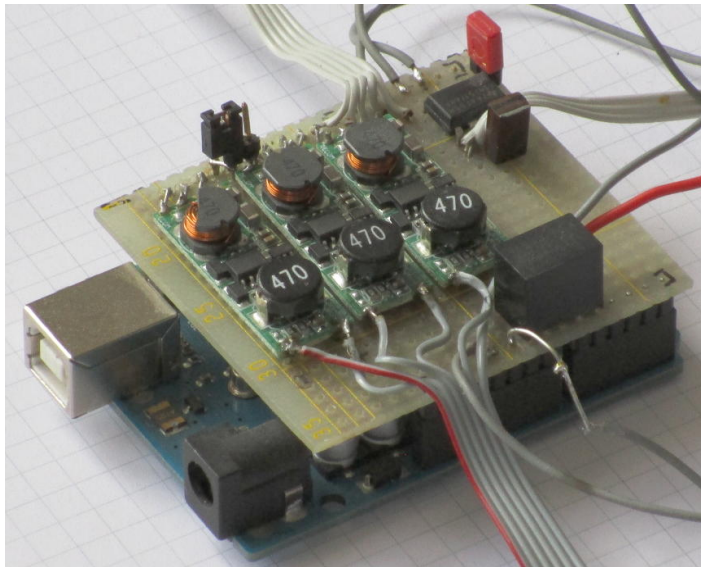
Hardware-Ankopplung (2) Shields

- Stabile Alternative: Aufgesteckte Zusatzplatine
→ „Shield“
- Einfaches Selbstbaubeispiel:



Anschluss eines
Empfängers für
DCF77-Atomuhr

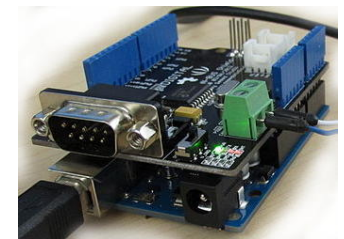
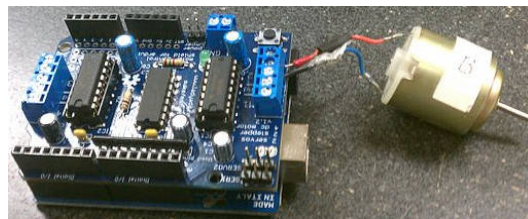
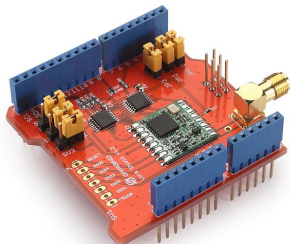
Hardware-Ankopplung (3) Shields



Noch ein Selbstbaubeispiel:

- Dreikanal Stromregler für farbige Power-LEDs

- Es gibt eine Unmenge von Shields für alle denkbare Hardware fertig zu kaufen:



Sensoren (Bsp.)



DS18B20

Temperatur-Sensor

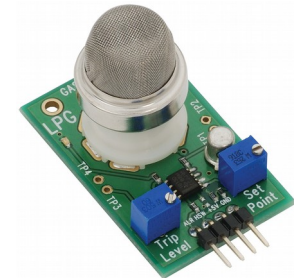


DS18B20



HC-SR04

Abstand-Sensor

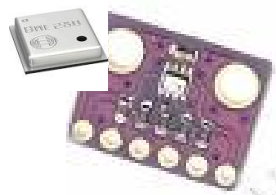


Gas-Sensor

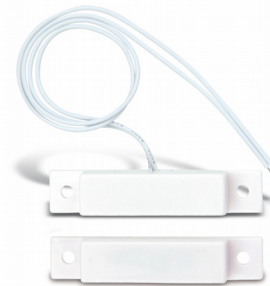


DTH22

Temperatur +
Luftfeuchtigkeit +
Druck (BME 280)



BME 280



Fenster-, Türkontakt



Servo



HC-SR501

Bewegungs-Sensor



Feinstaub-Sensor

Hilfreiche Links

- Arduino.cc Webseite:
<https://www.arduino.cc/>
- IDE Download:
<https://www.arduino.cc/en/Main/Software#>
- Arduino Reference:
<https://www.arduino.cc/en/Reference/HomePage>
-