

Physical Computing und Arduino

Jürgen Plate

3. März 2012

„Ich denke, es gibt einen Weltmarkt für vielleicht fünf Computer.“
Thomas Watson IBM-Vorstandsvorsitzender, 1943

„Es gibt keinen Grund, warum sich irgend jemand zu Hause einen Computer wünschen sollte.“
Ken Olsen, Gründer, Präsident und Vorstandsvorsitzender von Digital Equipment Corp., 1977

Physical Computing

„**Physical Computing** is about prototyping with electronics, 'turning sensors, actuators and microcontrollers into materials for designers and Artists.'“

„It involves the design of interactive objects that can communicate with humans using sensors and actuators controlled by a behaviour implemented as software running inside a microcontroller.“

/it Massimo Banzi

Physical Computing

Physical Computing bedeutet im weitesten Sinne, interaktive, physische Systeme durch die Verwendung von Hardware und Software zu erstellen. Diese Systeme reagieren auf Ereignisse in der realen, analogen Welt und wirken auf sie ein.

Der „Computer“

Viele kennen den Computer nur in der Form, wie er sich als Standardsystem präsentiert:

- ▶ Desktop-System mit Bildschirm, Tastatur, Maus und gegebenenfalls weiterer Standard-Peripherie,

Der „Computer“

Viele kennen den Computer nur in der Form, wie er sich als Standardsystem präsentiert:

- ▶ Desktop-System mit Bildschirm, Tastatur, Maus und gegebenenfalls weiterer Standard-Peripherie,
- ▶ Laptop bzw. Notebook, in dem fast alle Standard-Peripherie vereint ist oder

Der „Computer“

Viele kennen den Computer nur in der Form, wie er sich als Standardsystem präsentiert:

- ▶ Desktop-System mit Bildschirm, Tastatur, Maus und gegebenenfalls weiterer Standard-Peripherie,
- ▶ Laptop bzw. Notebook, in dem fast alle Standard-Peripherie vereint ist oder
- ▶ Tablet-PC, der ganz ohne Peripherie auskommt bzw. auskommen muss.

Gemeinsamkeit

Die Hardware ist ziemlich fest vorgegeben und man verwendet diese nur als Arbeitsgerät für softwareorientierte Arbeiten wie Programmieren, Konstruieren, Schreiben, Planen usw.

Physical Computing

Darunter werden hingegen Systeme verstanden, die sich mit der Beziehung zwischen dem Menschen und der digitalen Welt befassen:

- ▶ physikalische Größen der Umwelt (z. B. Temperatur, Helligkeit, Sound, Berührung etc.) erfassen

Physical Computing

Darunter werden hingegen Systeme verstanden, die sich mit der Beziehung zwischen dem Menschen und der digitalen Welt befassen:

- ▶ physikalische Größen der Umwelt (z. B. Temperatur, Helligkeit, Sound, Berührung etc.) erfassen
- ▶ darauf reagieren mittels elektromechanische Geräte wie Motoren, Servos, Leuchtdioden usw.

Physical Computing

Darunter werden hingegen Systeme verstanden, die sich mit der Beziehung zwischen dem Menschen und der digitalen Welt befassen:

- ▶ physikalische Größen der Umwelt (z. B. Temperatur, Helligkeit, Sound, Berührung etc.) erfassen
- ▶ darauf reagieren mittels elektromechanische Geräte wie Motoren, Servos, Leuchtdioden usw.
- ▶ die Software erzeugt die Reaktionen auf die Sensoren und steuert die Hardware an.

Physical Computing

Darunter werden hingegen Systeme verstanden, die sich mit der Beziehung zwischen dem Menschen und der digitalen Welt befassen:

- ▶ physikalische Größen der Umwelt (z. B. Temperatur, Helligkeit, Sound, Berührung etc.) erfassen
- ▶ darauf reagieren mittels elektromechanische Geräte wie Motoren, Servos, Leuchtdioden usw.
- ▶ die Software erzeugt die Reaktionen auf die Sensoren und steuert die Hardware an.
- ▶ oft für Projekte mit einem Hintergrund aus Kunst und Design oder für Do-it-yourself-Hobbyprojekte verwendet.

Hintergrund

Ein wichtiges Motiv ist, die Kontrolle über die Funktion der Hardware wieder zu erlangen („to take control back of what is inside“). Genau dieses Insiderwissen ist bei Geräten wie dem iPhone von Apple sorgsam abgeschirmt.

- ▶ Was Du nicht öffnen kannst, gehört Dir nicht vollständig.

Hintergrund

Ein wichtiges Motiv ist, die Kontrolle über die Funktion der Hardware wieder zu erlangen („to take control back of what is inside“). Genau dieses Insiderwissen ist bei Geräten wie dem iPhone von Apple sorgsam abgeschirmt.

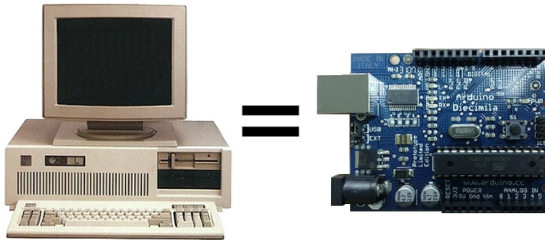
- ▶ Was Du nicht öffnen kannst, gehört Dir nicht vollständig.
- ▶ beim Physical Computing liegt alles offen zutage, wird jedes Gerät vom Benutzer selbst zusammengelötet, geschraubt oder gesteckt.

„Der Hauptunterschied zwischen etwas, was möglicherweise kaputtgehen könnte, und etwas, was unmöglich kaputtgehen kann, besteht darin, dass sich bei allem, was unmöglich kaputtgehen kann, falls es doch kaputtgeht, normalerweise herausstellt, dass es unmöglich zerlegt oder repariert werden kann.“

Douglas Adams

Mikrocontroller

Als Mikrocontroller werden Halbleiterchips bezeichnet, die mit dem Prozessor auch Peripheriefunktionen auf einem Chip vereinen. In vielen Fällen befindet sich der Arbeits- und Programmspeicher ebenfalls teilweise oder komplett auf dem gleichen Chip → Ein-Chip-Computersystem



Arduino

Physical Computing für Jedermann

- ▶ Entwickler: Massimo Banzi + Projektteam
2005, Universität Ivrea, Italien

Arduino

Physical Computing fur Jedermann

- ▶ Entwickler: Massimo Banzi + Projektteam
2005, Universitat Ivrea, Italien
- ▶ Aufbau: für Einsteiger geeignet, schnelle Erfolge
primäre Zielgruppe nicht Elektroniker/Informatiker, sondern
Künstler, Designer, usw.

Physical Computing fur Jedermann

- ▶ Entwickler: Massimo Banzi + Projektteam
2005, Universitat Ivrea, Italien
- ▶ Aufbau: für Einsteiger geeignet, schnelle Erfolge
primäre Zielgruppe nicht Elektroniker/Informatiker, sondern
Künstler, Designer, usw.
- ▶ Verfügbarkeit: Billige und verfügbare Standard-Komponenten

Physical Computing fur Jedermann

- ▶ Entwickler: Massimo Banzi + Projektteam
2005, Universitat Ivrea, Italien
- ▶ Aufbau: für Einsteiger geeignet, schnelle Erfolge
primäre Zielgruppe nicht Elektroniker/Informatiker, sondern
Künstler, Designer, usw.
- ▶ Verfügbarkeit: Billige und verfügbare Standard-Komponenten
- ▶ Offen: Open-Source-Software und -Hardware

Physical Computing fur Jedermann

- ▶ Entwickler: Massimo Banzi + Projektteam
2005, Universitat Ivrea, Italien
- ▶ Aufbau: für Einsteiger geeignet, schnelle Erfolge
primäre Zielgruppe nicht Elektroniker/Informatiker, sondern
Künstler, Designer, usw.
- ▶ Verfügbarkeit: Billige und verfügbare Standard-Komponenten
- ▶ Offen: Open-Source-Software und -Hardware
- ▶ Infos unter *<http://www.arduino.cc/>*

Arduino-Plattform

Arduino ist eine Physical-Computing-Plattform:

- ▶ besteht aus Hard- und Software

Arduino-Plattform

Arduino ist eine Physical-Computing-Plattform:

- ▶ besteht aus Hard- und Software
- ▶ Hardware: Mikrocontroller + USB-Schnittstelle (virt. seriell), parallele, serielle und analoge Schnittstellen

Arduino-Plattform

Arduino ist eine Physical-Computing-Plattform:

- ▶ besteht aus Hard- und Software
- ▶ Hardware: Mikrocontroller + USB-Schnittstelle (virt. seriell), parallele, serielle und analoge Schnittstellen
- ▶ Stromversorgung über USB oder autonom (Akku, Netzteil)

Arduino-Plattform

Arduino ist eine Physical-Computing-Plattform:

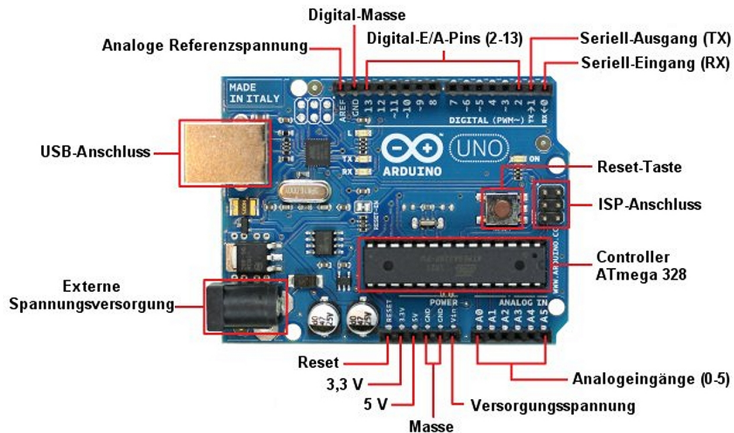
- ▶ besteht aus Hard- und Software
- ▶ Hardware: Mikrocontroller + USB-Schnittstelle (virt. seriell), parallele, serielle und analoge Schnittstellen
- ▶ Stromversorgung über USB oder autonom (Akku, Netzteil)
- ▶ Software: leicht bedienbare Entwicklungsumgebung beruhend auf Processing (Java-Dialekt) und Wiring (C-Dialekt)

Arduino-Plattform

Arduino ist eine Physical-Computing-Plattform:

- ▶ besteht aus Hard- und Software
- ▶ Hardware: Mikrocontroller + USB-Schnittstelle (virt. seriell), parallele, serielle und analoge Schnittstellen
- ▶ Stromversorgung über USB oder autonom (Akku, Netzteil)
- ▶ Software: leicht bedienbare Entwicklungsumgebung beruhend auf Processing (Java-Dialekt) und Wiring (C-Dialekt)
- ▶ **kein** spezielles Programmiergerät nötig

Arduino-Hardware



Arduino-Hardware

- ▶ Mikrocontroller ATmega 328

Arduino-Hardware

- ▶ Mikrocontroller ATmega 328
- ▶ 32 KByte Flash (davon 0,5 KByte für Bootloader)

Arduino-Hardware

- ▶ Mikrocontroller ATmega 328
- ▶ 32 KByte Flash (davon 0,5 KByte für Bootloader)
- ▶ 2 KByte RAM, 1 KByte EEPROM

Arduino-Hardware

- ▶ Mikrocontroller ATmega 328
- ▶ 32 KByte Flash (davon 0,5 KByte für Bootloader)
- ▶ 2 KByte RAM, 1 KByte EEPROM
- ▶ 16 MHz Takt

Arduino-Hardware

- ▶ Mikrocontroller ATmega 328
- ▶ 32 KByte Flash (davon 0,5 KByte für Bootloader)
- ▶ 2 KByte RAM, 1 KByte EEPROM
- ▶ 16 MHz Takt
- ▶ 14 digitale I/O-Pins

Arduino-Hardware

- ▶ Mikrocontroller ATmega 328
- ▶ 32 KByte Flash (davon 0,5 KByte für Bootloader)
- ▶ 2 KByte RAM, 1 KByte EEPROM
- ▶ 16 MHz Takt
- ▶ 14 digitale I/O-Pins
- ▶ davon 6 für Pulsweitenmodulation (PWM) nutzbar

Arduino-Hardware

- ▶ Mikrocontroller ATmega 328
- ▶ 32 KByte Flash (davon 0,5 KByte für Bootloader)
- ▶ 2 KByte RAM, 1 KByte EEPROM
- ▶ 16 MHz Takt
- ▶ 14 digitale I/O-Pins
- ▶ davon 6 für Pulsweitenmodulation (PWM) nutzbar
- ▶ 6 analoge Eingänge (10 Bit)

Arduino-Hardware

- ▶ Mikrocontroller ATmega 328
- ▶ 32 KByte Flash (davon 0,5 KByte für Bootloader)
- ▶ 2 KByte RAM, 1 KByte EEPROM
- ▶ 16 MHz Takt
- ▶ 14 digitale I/O-Pins
- ▶ davon 6 für Pulsweitenmodulation (PWM) nutzbar
- ▶ 6 analoge Eingänge (10 Bit)
- ▶ On-Board USB-Schnittstelle

Arduino-Hardware

- ▶ Mikrocontroller ATmega 328
- ▶ 32 KByte Flash (davon 0,5 KByte für Bootloader)
- ▶ 2 KByte RAM, 1 KByte EEPROM
- ▶ 16 MHz Takt
- ▶ 14 digitale I/O-Pins
- ▶ davon 6 für Pulsweitenmodulation (PWM) nutzbar
- ▶ 6 analoge Eingänge (10 Bit)
- ▶ On-Board USB-Schnittstelle
- ▶ 5 V und 3,3 V Betriebsspannung, Speisung über USB oder extern (7 – 12 V)

Anschlussmöglichkeiten

- ▶ Taster, Schalter, Kontakte

Anschlussmöglichkeiten

- ▶ Taster, Schalter, Kontakte
- ▶ Sensoren (Temperatur, Entfernung, Beschleunigung, ...)

Anschlussmöglichkeiten

- ▶ Taster, Schalter, Kontakte
- ▶ Sensoren (Temperatur, Entfernung, Beschleunigung, ...)
- ▶ Lichtschranken, Infrarot-Fernbedienungen, Mikrofone, ...

Anschlussmöglichkeiten

- ▶ Taster, Schalter, Kontakte
- ▶ Sensoren (Temperatur, Entfernung, Beschleunigung, ...)
- ▶ Lichtschranken, Infrarot-Fernbedienungen, Mikrofone, ...
- ▶ LEDs, Motoren, Servos, Magnetschalter, ...

Anschlussmöglichkeiten

- ▶ Taster, Schalter, Kontakte
- ▶ Sensoren (Temperatur, Entfernung, Beschleunigung, ...)
- ▶ Lichtschranken, Infrarot-Fernbedienungen, Mikrofone, ...
- ▶ LEDs, Motoren, Servos, Magnetschalter, ...
- ▶ serielle Schnittstelle, USB, Netzwerk, Bluetooth, ZigBee, ...

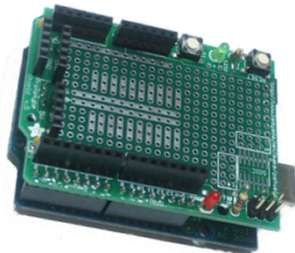
Anschlussmöglichkeiten

- ▶ Taster, Schalter, Kontakte
- ▶ Sensoren (Temperatur, Entfernung, Beschleunigung, ...)
- ▶ Lichtschranken, Infrarot-Fernbedienungen, Mikrofone, ...
- ▶ LEDs, Motoren, Servos, Magnetschalter, ...
- ▶ serielle Schnittstelle, USB, Netzwerk, Bluetooth, ZigBee, ...
- ▶ Flash-Speicher, SD-Card

Anschlussmöglichkeiten

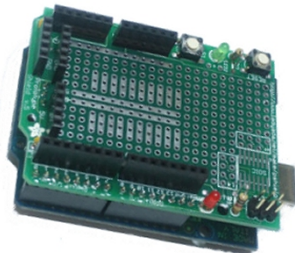
- ▶ Taster, Schalter, Kontakte
- ▶ Sensoren (Temperatur, Entfernung, Beschleunigung, ...)
- ▶ Lichtschranken, Infrarot-Fernbedienungen, Mikrofone, ...
- ▶ LEDs, Motoren, Servos, Magnetschalter, ...
- ▶ serielle Schnittstelle, USB, Netzwerk, Bluetooth, ZigBee, ...
- ▶ Flash-Speicher, SD-Card
- ▶ und andere, unvorstellbare Dinge

Shields



- Shields: Platinen, die direkt auf den Arduino gesteckt werden können

Shields



- ▶ Shields: Platinen, die direkt auf den Arduino gesteckt werden können
- ▶ erweitern Arduino um viele interessante Möglichkeiten

Shields



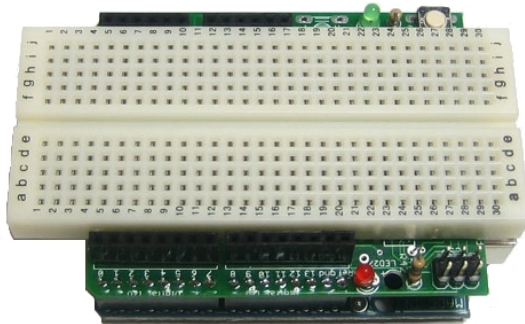
- ▶ Shields: Platinen, die direkt auf den Arduino gesteckt werden können
- ▶ erweitern Arduino um viele interessante Möglichkeiten
- ▶ Liste der Shield sehr lang: <http://www.shieldlist.org/>

Shields



- ▶ Shields: Platinen, die direkt auf den Arduino gesteckt werden können
- ▶ erweitern Arduino um viele interessante Möglichkeiten
- ▶ Liste der Shield sehr lang: <http://www.shieldlist.org/>
- ▶ Für Bastler: Proto Shield

Proto-Shield



Software

- ▶ IDE erhältlich für Windows, Linux, Mac

Software

- ▶ IDE erhältlich für Windows, Linux, Mac
- ▶ benötigt installierte JRE

Software

- ▶ IDE erhältlich für Windows, Linux, Mac
- ▶ benötigt installierte JRE
- ▶ Mac und Windows Version bringen Compiler mit

Software

- ▶ IDE erhältlich für Windows, Linux, Mac
- ▶ benötigt installierte JRE
- ▶ Mac und Windows Version bringen Compiler mit
- ▶ Linux benötigt GCC mit einigen AVR Bibliotheken

Software

- ▶ IDE erhältlich für Windows, Linux, Mac
- ▶ benötigt installierte JRE
- ▶ Mac und Windows Version bringen Compiler mit
- ▶ Linux benötigt GCC mit einigen AVR Bibliotheken
- ▶ Programmiersprache ist ein Subset von C++

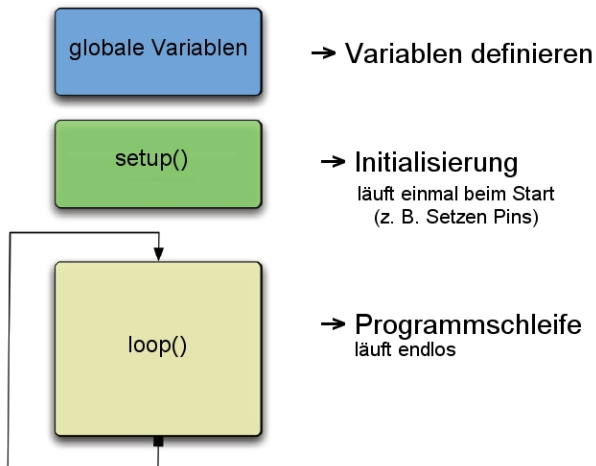
Software

- ▶ IDE erhältlich für Windows, Linux, Mac
- ▶ benötigt installierte JRE
- ▶ Mac und Windows Version bringen Compiler mit
- ▶ Linux benötigt GCC mit einigen AVR Bibliotheken
- ▶ Programmiersprache ist ein Subset von C++
- ▶ ein Programm wird **sketch** (Skizze) genannt

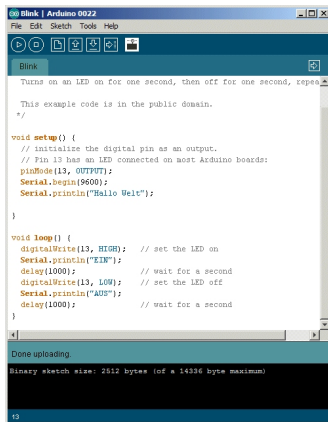
Software

- ▶ IDE erhältlich für Windows, Linux, Mac
- ▶ benötigt installierte JRE
- ▶ Mac und Windows Version bringen Compiler mit
- ▶ Linux benötigt GCC mit einigen AVR Bibliotheken
- ▶ Programmiersprache ist ein Subset von C++
- ▶ ein Programm wird **sketch** (Skizze) genannt
- ▶ es gibt unzählige Bibliotheken

Standard-Programmablauf



Entwicklungsumgebung



The screenshot shows the Arduino IDE interface with the 'Blink' sketch loaded. The title bar reads 'Blink | Arduino 0022'. The menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. The toolbar contains icons for opening, saving, compiling, uploading, and erasing. The sketch text area contains the following code:

```
Turns on an LED on for one second, then off for one second, repeats.

This example code is in the public domain.

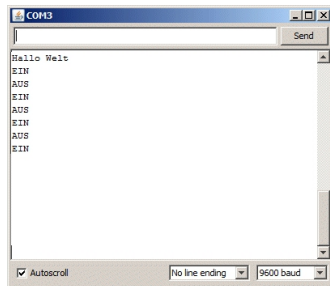
*/

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
  Serial.begin(9600);
  Serial.println("Hallo Welt");
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  Serial.println("EIN");
  delay(1000);             // wait for a second
  digitalWrite(13, LOW);  // set the LED off
  Serial.println("AUS");
  delay(1000);             // wait for a second
}
```

At the bottom, the status bar shows 'Done uploading.' and 'Binary sketch size: 2512 bytes (of a 14336 byte maximum)'. The pin number '13' is visible in the bottom left corner of the IDE window.

Entwicklungsumgebung



Funktionen der IDE

Über die Schaltflächen an der Oberkante des Fensters oder die Menüleiste lassen sich alle Funktionen der IDE steuern:

- ▶ Verify: das Programm kompilieren (Syntaxcheck)

Funktionen der IDE

Über die Schaltflächen an der Oberkante des Fensters oder die Menüleiste lassen sich alle Funktionen der IDE steuern:

- ▶ Verify: das Programm kompilieren (Syntaxcheck)
- ▶ New: neues Programm erstellen

Funktionen der IDE

Über die Schaltflächen an der Oberkante des Fensters oder die Menüleiste lassen sich alle Funktionen der IDE steuern:

- ▶ Verify: das Programm kompilieren (Syntaxcheck)
- ▶ New: neues Programm erstellen
- ▶ Open: Programmdatei laden

Funktionen der IDE

Über die Schaltflächen an der Oberkante des Fensters oder die Menüleiste lassen sich alle Funktionen der IDE steuern:

- ▶ Verify: das Programm kompilieren (Syntaxcheck)
- ▶ New: neues Programm erstellen
- ▶ Open: Programmdatei laden
- ▶ Upload: Programm kompilieren und über USB in den Arduino laden

Funktionen der IDE

Über die Schaltflächen an der Oberkante des Fensters oder die Menüleiste lassen sich alle Funktionen der IDE steuern:

- ▶ Verify: das Programm kompilieren (Syntaxcheck)
- ▶ New: neues Programm erstellen
- ▶ Open: Programmdatei laden
- ▶ Upload: Programm kompilieren und über USB in den Arduino laden
- ▶ Serial Monitor: Serielle Ausgaben des Arduino anzeigen

Funktionen der IDE

Über die Schaltflächen an der Oberkante des Fensters oder die Menüleiste lassen sich alle Funktionen der IDE steuern:

- ▶ Verify: das Programm kompilieren (Syntaxcheck)
- ▶ New: neues Programm erstellen
- ▶ Open: Programmdatei laden
- ▶ Upload: Programm kompilieren und über USB in den Arduino laden
- ▶ Serial Monitor: Serielle Ausgaben des Arduino anzeigen
- ▶ Stop: seriellen Monitor anhalten

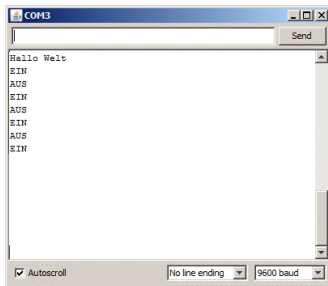
Funktionen der IDE

Über die Schaltflächen an der Oberkante des Fensters oder die Menüleiste lassen sich alle Funktionen der IDE steuern:

- ▶ Verify: das Programm kompilieren (Syntaxcheck)
- ▶ New: neues Programm erstellen
- ▶ Open: Programmdatei laden
- ▶ Upload: Programm kompilieren und über USB in den Arduino laden
- ▶ Serial Monitor: Serielle Ausgaben des Arduino anzeigen
- ▶ Stop: seriellen Monitor anhalten
- ▶ Help: Hilfe

Programmablauf

Anzeige über den seriellen Monitor der IDE



Evolution

Inzwischen gibt es eine ganze Arduino-Familie:

- ▶ Arduino Uno
- ▶ Arduino Mega
- ▶ Arduino Nano
- ▶ Lilipad
- ▶ LEDuino
- ▶ und auch Clone

Unterlagen und Links

<http://www.netzmafia.de/skripten/hardware/Arduino/>

