

Was ist Arduino ?



- Historie
- Der Kern
- Entwicklungsumgebung
- Hardware
- Software
- Und wozu das Ganze ?

Das Arduino-Projekt

- Arduino ist ein Open-Source-Prototypen-System, daß die Herren *Massimo Banzi* und *David Cuartielles* am *Interaction Design Institute* in Ivrea / Italien entwickelt und nach einem lokalen Potentaten (Arduin von Ivrea) genannt haben.
→ ***Namensgeber könnte auch die gleichnamige Kneipe neben dem Institut gewesen sein !***
- Ursprünglich sollte es ein einfaches System sein, das Designer und Künstler anregen sollte ihre Objekte intelligent zu animieren (Licht, Farbe, Ton, Bewegung...)
→ ***Möglichst einfach !***
- Es sollten übergreifende Systeme entstehen, die mit Windows, LINUX, Mac zusammenarbeiten können.
→ ***Entwicklungsumgebung arbeitet unter all diesen Betriebssystemen !***
- Programmiersprache auf Controller-Ebene ist C (C++) mit angepassten Bibliotheken aus dem WIRING-Projekt. Es sind alle C-Befehle des GNU-Compilers sowie die Register des Prozessors nutzbar.
→ ***Einfache Anwendung der Sprache C !***

Arduino-Hardware

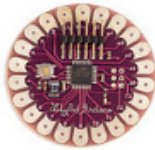
- Es wurde eine Serie von Hardwareboards entwickelt, die Basisboards und Shields (Peripherieboards) darstellen:

Official Arduino boards

Official Arduino Shields



Arduino Uno



Arduino LilyPad



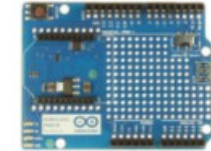
Arduino Ethernet Shield



Arduino Mega ADK



Arduino Pro



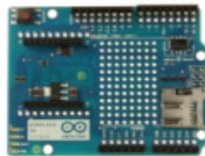
Arduino Wireless Proto Shield



Arduino Mega 2560



Arduino Fio



Arduino Wireless SD Shield



Arduino Ethernet



Arduino Nano



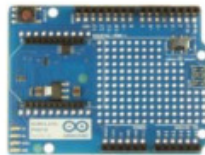
Arduino Motor Shield



Arduino Mega ADK



Arduino Pro



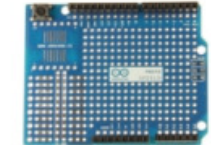
Arduino Wireless Proto Shield



Arduino BT



Arduino Mini



Arduino Proto Shield

- Bis Mai 2011 wurden weltweit mehr als 300.000 Boards verkauft oder über Bausätze selbstgebaut

Aber : Was braucht man mindestens ?

Entwicklungsumgebung



Entwicklungsumgebung
OPEN-SOURCE

USB/
RS232-



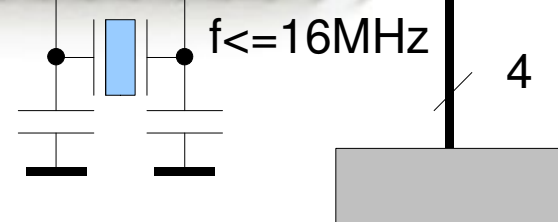
USB

RX/
TX

Atmega328P
Atmega168
Atmega16U2
Atmega8U2 up to version R2)



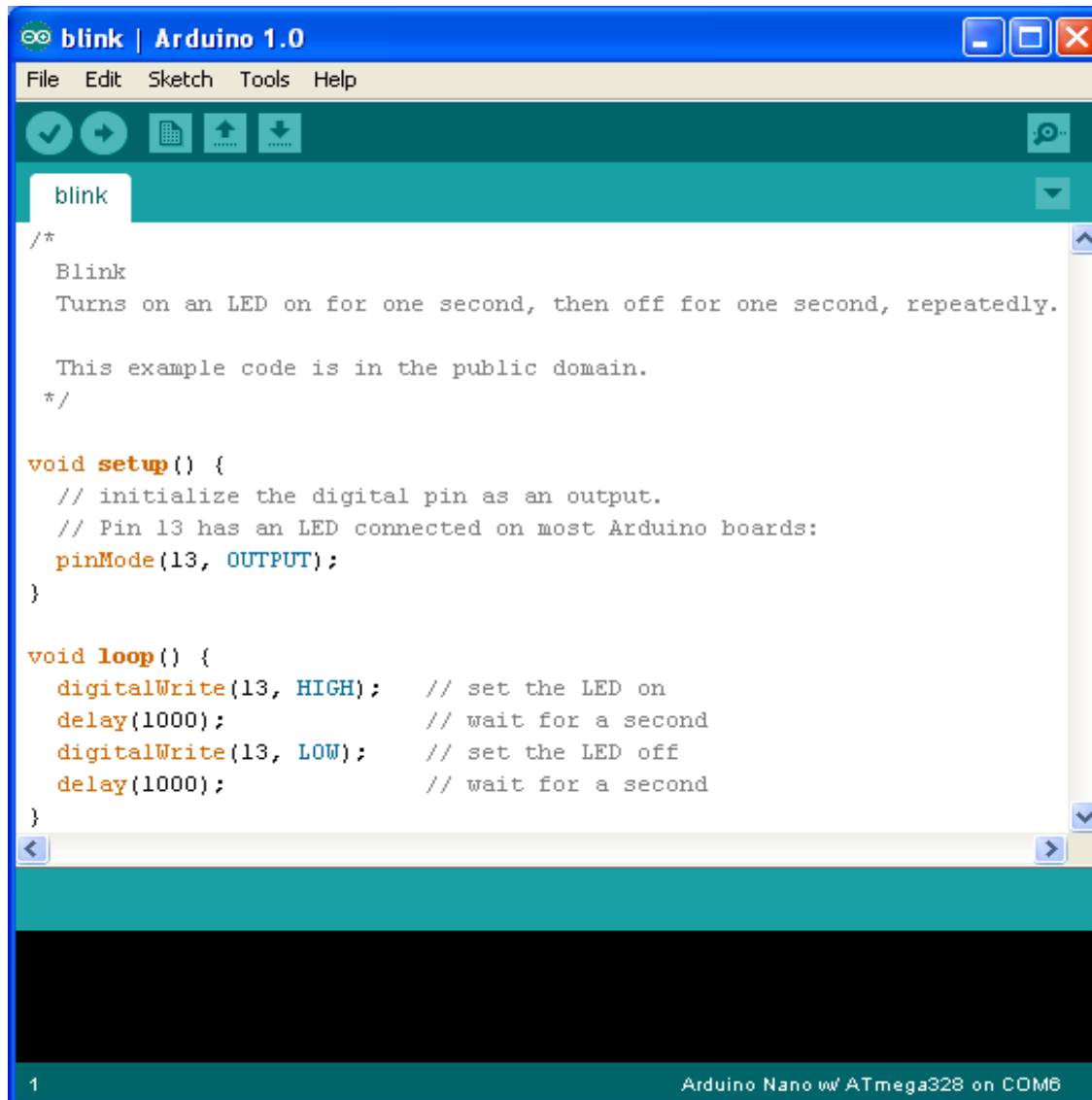
Bootloader
OPEN-SOURCE
(selbst flashbar)



4p-ISP-
Stecker

- Man kann leicht eigene kleine Systeme aufbauen
- Man kann eigene Ein-/Ausgaben anschliessen
- Man kann AVR-Controller ohne Bootlader kaufen und selbst den Bootloader brennen
- Man kann direkt USB-Schnittstellen des PCs verwenden (evtl . Inkl. Spannungsvers.)
- Preise: Atmega328p:4€, FT232:4€, Quarz 0,5€
- Der Controller ist in der Schaltung umprogrammierbar

Entwicklungsumgebung



The screenshot shows the Arduino IDE interface. The title bar reads 'blink | Arduino 1.0'. The menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. Below the menu bar is a toolbar with icons for running, saving, and other functions. The main text area contains the following code:

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

  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);
}

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

At the bottom of the window, the status bar shows '1' on the left and 'Arduino Nano w/ ATmega328 on COM6' on the right.

- Programm mit Bedienfenster auf PC
- Laden und Speichern von eigenen Programmen
- Editor zum Schreiben und Ändern des eigenen Programmes
- Upload des entwickelten Programms auf μC per USB
- Auswahl des verwendeten μC -Typs
- Serielles Terminal zum Empfang von Meldungen des laufenden Programms per USB
- Bedienung eines Programmers zum Flashen eines Bootloaders auf neuen μCs

Programmierung

Programmstruktur

Inkl. Bibliotheken
Definition Symbole
Def. Globale Variablen

Programmteil **setup**
(alle Initialisierungen)

Programmteil **loop**

(Hauptprogramm in
einer Endlosschleife)

Interrupt-Routinen

Unterprogramme

void setup()

```
{  
  // initialize the digital pin as an output.  
  // Pin 13 has an LED connected on most Arduino boards:  
  pinMode(13, OUTPUT);  
}
```

void loop()

```
{  
  digitalWrite(13, HIGH); // set the LED on  
  delay(1000);           // wait for a second  
  digitalWrite(13, LOW); // set the LED off  
  delay(1000);           // wait for a second  
}
```



Sprache C und Bibliotheken

Structure

- setup()
- loop()

Control Structures

- if
- if...else
- for
- switch case
- while
- do... while
- break
- continue
- return
- goto

Für jeden Befehl oder Funktionsaufruf ist eine umfangreiche Beschreibung mit Beispiel anzeigbar.

Variables

Constants

- HIGH | LOW
- INPUT | OUTPUT
- true | false
- integer constants
- floating point constants

Data Types

- void
- boolean
- char
- unsigned char
- byte
- int

Eingebautes Handbuch für C und Arduino-Bibliotheken

Further Syntax

- ; (semicolon)
- {} (curly braces)
- // (single line comment)
- /**/ (multi-line comment)
- #define
- #include

Arithmetic Operators

- = (assignment operator)
- + (addition)
- - (subtraction)
- * (multiplication)
- / (division)
- % (modulo)

- word
- long
- unsigned long
- float
- double
- string - char array
- String - object
- array

Conversion

- char()
- byte()
- int()
- word()
- long()
- float()

Sprache C und Bibliotheken

Libraries

Libraries provide extra functions in a sketch, select it from **Sketch**

Standard Libraries

- EEPROM - reading and writing
- Ethernet - for connecting to
- Firmata - for communicating
- LiquidCrystal - for controlling
- SD - for reading and writing
- Servo - for controlling servos
- SPI - for communicating with
- SoftwareSerial - for serial communication
- Stepper - for controlling stepper
- Wire - Two Wire Interface

...und viele weitere auf Arduino Playground oder im Internet

Communication (networking and protocols):

- Messenger - for processing text-based messages from the computer
- NewSoftSerial - an improved version of the SoftwareSerial library
- OneWire - control devices (from Dallas Semiconductor) that use the One Wire protocol.
- PS2Keyboard - read characters from a PS2 keyboard.
- Simple Message System - send messages between Arduino and the computer
- SSerial2Mobile - send text messages or emails using a cell phone (via AT commands over software serial)
- Webduino - extensible web server library (for use with the Arduino Ethernet Shield)
- X10 - Sending X10 signals over AC power lines
- XBee - for communicating with XBees in API mode
- SerialControl - Remote control other Arduinos over a serial connection

Sensing:

- Capacitive Sensing - turn two or more pins into capacitive sensors
- Debounce - for reading noisy digital inputs (e.g. from buttons)

Displays and LEDs:

- Improved LCD library fixes LCD initialization bugs in official Arduino LCD library
- GLCD - graphics routines for LCD based on the KS0108 or equivalent chipset.
- LedControl - for controlling LED matrices or seven-segment displays with a MAX7221 or MAX7219.
- LedControl - an alternative to the Matrix library for driving multiple LEDs with Maxim chips.
- LedDisplay - control of a HCMS-29xx scrolling LED display.

Arduinio - Unterstützung

www.netzmafia.de:

Arduino Programmierhandbuch (deutsch)

<http://www.ladyada.net/learn/arduino/lesson4.html>:

Sehr ausführliches Tutorial mit Installation (Englisch)

<http://www.freeduino.de/books/arduino-tutorial-lady-ada>

Teilweise Übersetzung von Ladyada-Tutorial (deutsch)

<http://www.youtube.com/playlist?list=PLAB63281B90FB376E&feature=plcp>

Youtube Tutorial Serie

<http://www.youtube.com/watch?v=avOZaRoKVgg>

Youtube Tutorial Serie mit Installation vom Internet auf PC (deutsch)

Ende

