



Open Source Hardware als Innovation für Wandlungsfähige Fabriken

Andreas Merkel, Hendrik Hopf



Das Projekt IREKO (Nachhaltige Realisierung von Innovation im regionalen Arbeitskontext) wird mit Mitteln des Europäischen Sozialfonds und des Freistaates Sachsen gefördert.

Theorie

- Open Source Hardware
- Wandlungsfähige Fabriken
- Open Source Hardware Arduino
- Forschungsfragen

Praxis

- Programmierung & Programmierbeispiele
- Prototypen

Theorie

Open Source Hardware

Definition



- “Open source hardware is hardware whose design is **made publicly** available so that anyone **can study, modify, distribute, make, and sell the design or hardware based on that design.**”
- “The hardware's source, the design from which it is made, is available in the **preferred format** for making modifications to it.”
- “Ideally, open source hardware **uses readily-available components** and materials, standard processes, open infrastructure, unrestricted content, and open-source design tools **to maximize the ability of individuals to make and use hardware.**”
- “Open source hardware gives people the freedom to control their technology while **sharing knowledge and encouraging commerce through the open exchange of designs.**”



GNU General Public License, Free Software Foundation

The GNU General Public License is a **Free Software** license. Like any Free Software license, it grants to you the four following freedoms:

0. The freedom to run the program for any purpose.
1. The freedom to study how the program works and adapt it to your needs.
2. The freedom to redistribute copies so you can help your neighbor.
3. The freedom to improve the program and release your improvements to the public, so that the whole community benefits.

You may exercise the freedoms specified here provided that you comply with the express conditions of this license. The principal conditions are:



You must conspicuously and appropriately publish on each copy distributed an appropriate copyright notice and disclaimer of warranty and keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of the GNU General Public License along with the Program. Any translation of the GNU General Public License must be accompanied by the GNU General Public License.



If you modify your copy or copies of the program or any portion of it, or develop a program based upon it, you may distribute the resulting work provided you do so under the GNU General Public License. Any translation of the GNU General Public License must be accompanied by the GNU General Public License.



If you copy or distribute the program, you must accompany it with the complete corresponding machine-readable source code or with a written offer, valid for at least three years, to furnish the complete corresponding machine-readable source code.

Any of the above conditions can be waived if you get permission from the copyright holder.

This is a human-readable summary of the Legal Code (the full GNU General Public License). A Portuguese translation is also available. [Disclaimer](#)

<http://creativecommons.org/>

Open Source Hardware

Vorteile von Open Source Entwicklungen



- Quelle vieler guter Ideen
- Hohe Partizipation im Entwicklungsprozess
- Hohe Modularität in den Lösungen
- Maximale Transparenz im Entwicklungsprozess
- Hohe Entwicklungsgeschwindigkeit
- Zusätzliches Entwicklungspotential

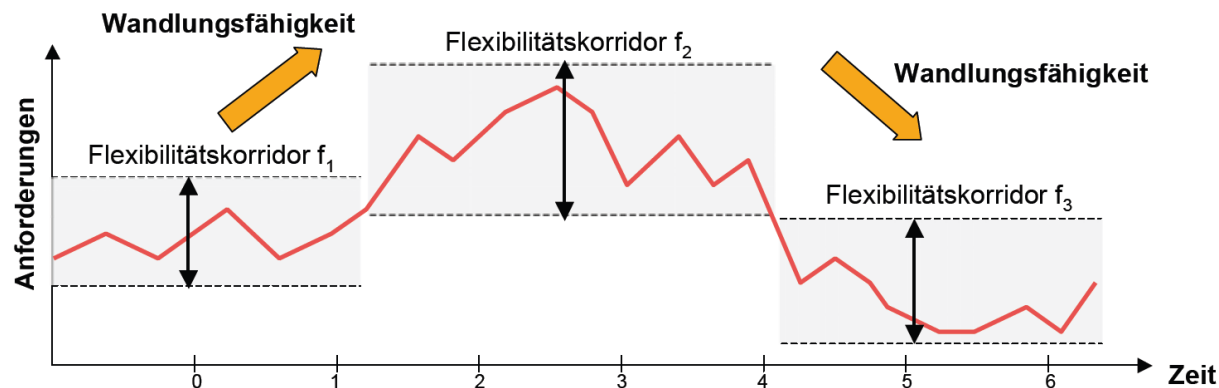
Open Source Hardware

Software vs. Hardware



		Software	Hardware
Beispiel		Office Software	CNC Fräsmaschine
Anwendung		Bürotätigkeiten	Herstellung von Gütern
Produkt		immateriell	materiell
Entwicklung	Mechanik	-	Konstruktion
	Elektrik	-	Konstruktion
	Steuerung	Programmierung	Programmierung
	Prototypenbau	Kompilieren	Beschaffung, Produktion
	Ergebnis	Source-Code Prototypisches Programm	Konstruktionszeichnungen, Stückliste, Source-Code, physischer Prototyp
Marktein- führung	Zertifizierung	aller von der Software ausgehenden Gefahren	aller von der Maschine ausgehenden Gefahren (Maschinenrichtlinien)
Produktion		Lediglich kopieren	Vervielfältigung unter Material- und Kosteneinsatz
Distribution		U.a. Transport über das Internet	Transport durch materielle Logistiksysteme

„Wandlungsfähigkeit wird als Potenzial verstanden, eine schnelle Anpassung auch jenseits vorgehaltener Korridore in Bezug auf Organisation und Technik bei einem geringen Investitionsaufwand zu ermöglichen.“ (Reinhart et al. 2008)



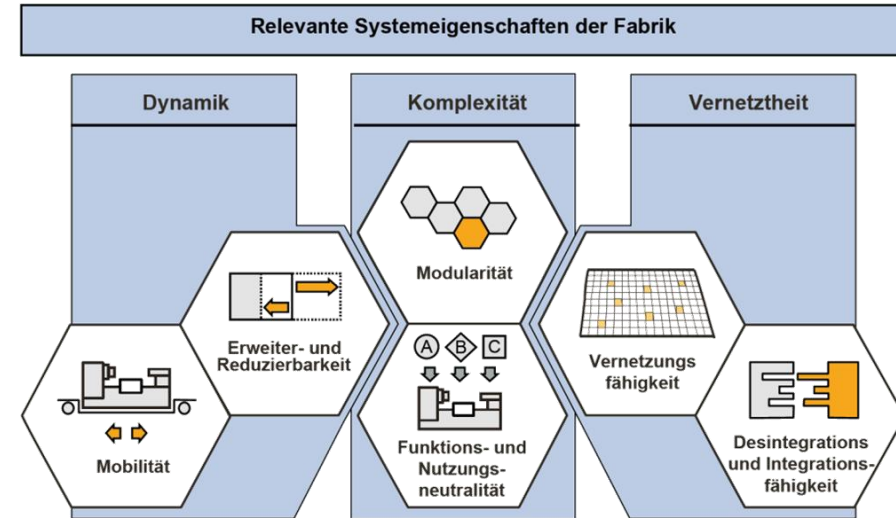
Flexibilität	Wandlungsfähigkeit
- vorgehaltener Fähigkeitsbereich - skalierbar in festgelegten Korridoren - Rückbau nicht vorgesehen	- vorgedachter Lösungsraum - Veränderung im Bedarfsfall - Rückbauoption als Grundeigenschaft

Digital Center Komponenten

- Planungs-/Visualisierungsmodule (gestalten)

Experimentier Center Komponenten

- Bearbeitungsmodule (fertigen, montieren, prüfen)
- Materialflussmodule (handlen, transportieren)
- Informationsflussmodule (steuern, überwachen)
- Energieflussmodule (versorgen)



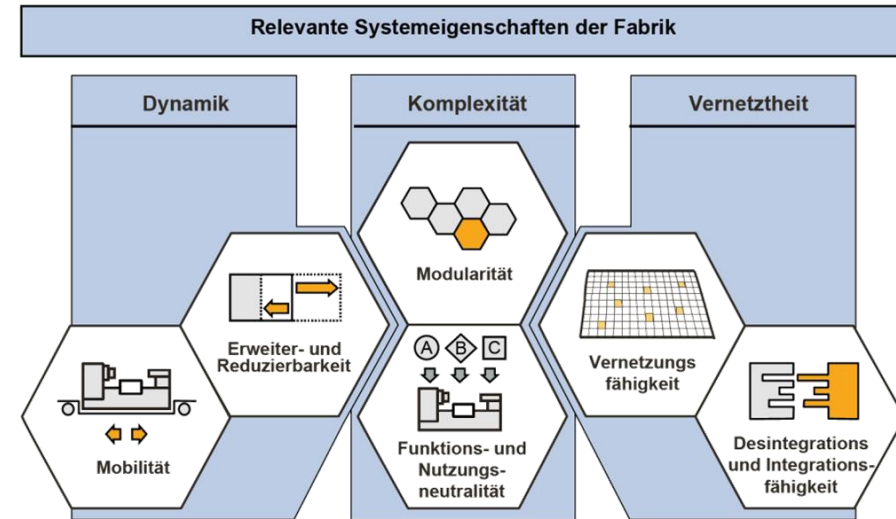
Experimentier- und Digitalfabrik

Digital Center Komponenten

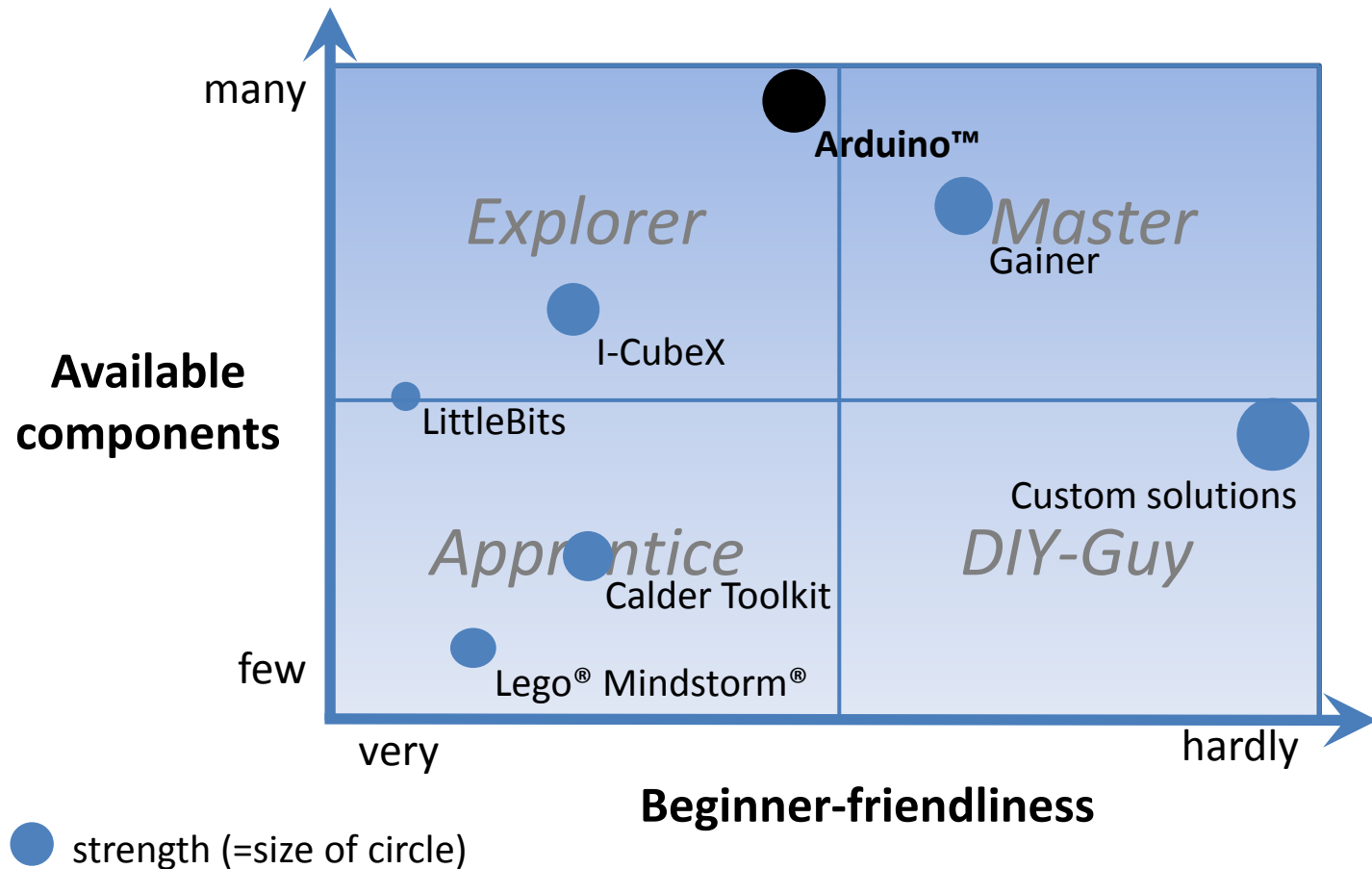
- Planungs-/Visualisierungsmodule (gestalten)

Experimentier Center Komponenten

- Bearbeitungsmodule (fertigen, montieren, prüfen)
- Materialflussmodule (handlen, transportieren)
- Informationsflussmodule (steuern, überwachen)**
- Energieflussmodule (versorgen)



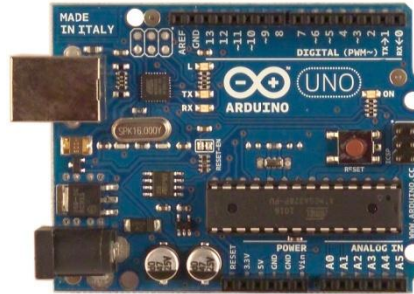
Experimentier- und Digitalfabrik



Gainer



Arduino



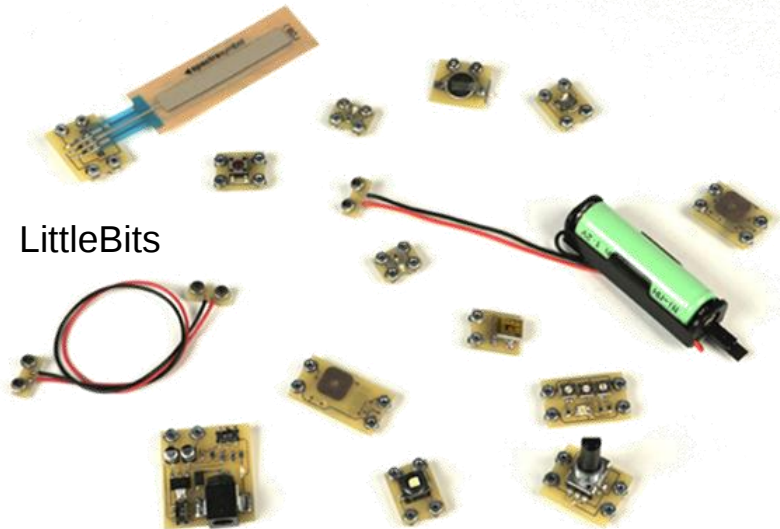
BugLabs



MICA



LittleBits



I-CubeX



Beagleboard



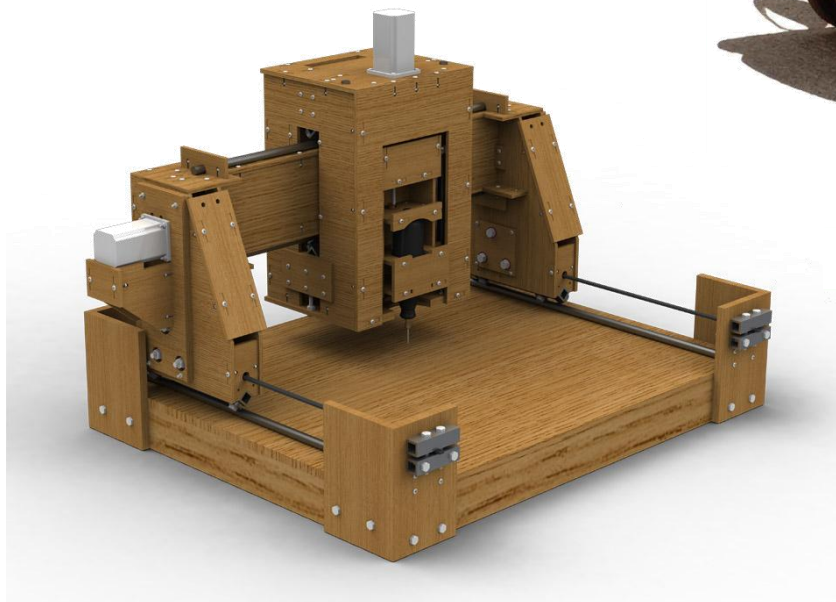
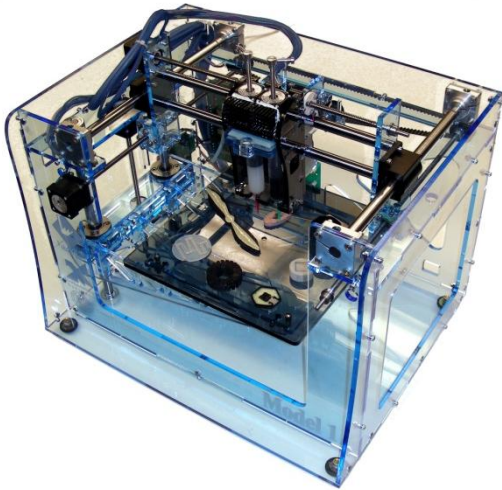
Weitere Open Source Hardware



13



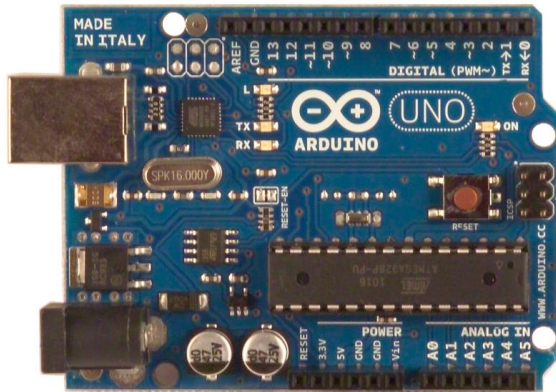
Weitere Open Source Hardware



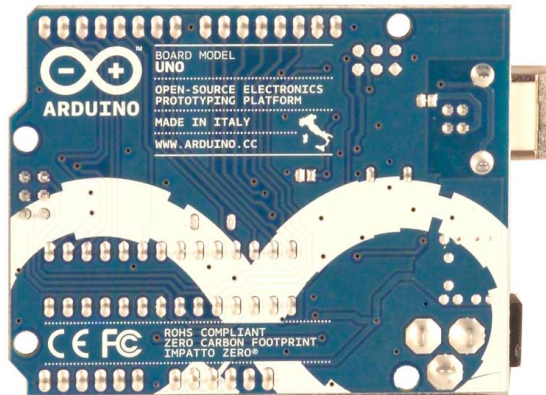
Open Source Hardware Arduino



15



Microcontroller



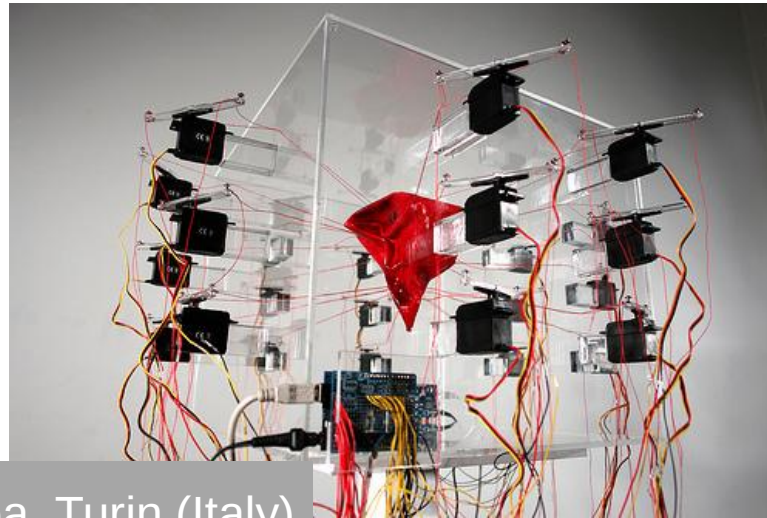
Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz (8 bit)
Price	25€
Licence	General Public License

Open Source Hardware Arduino

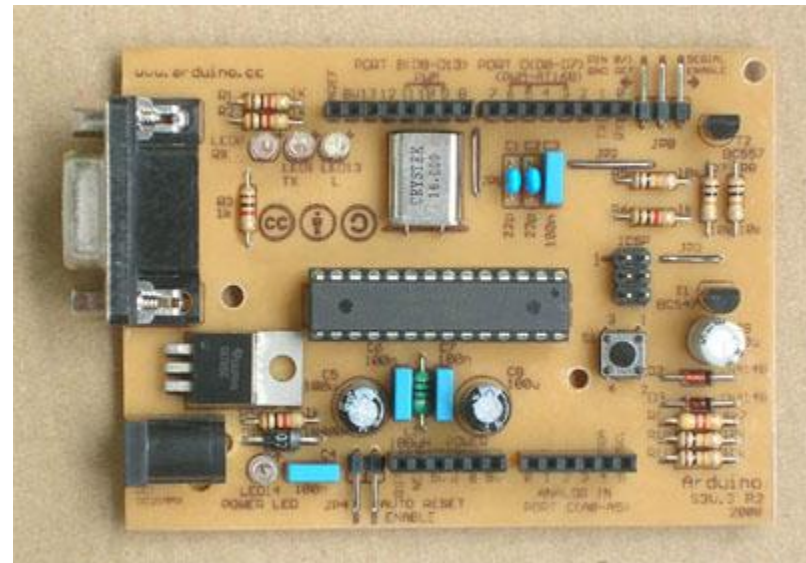
Erfinder



16



Interaction Design Institute Ivrea, Turin (Italy)



Open Source Hardware Arduino

Just-for-Fun Projekte der Arduino Community



17



TECHNISCHE UNIVERSITÄT
CHEMNITZ

Open Source Hardware Arduino

Komponenten

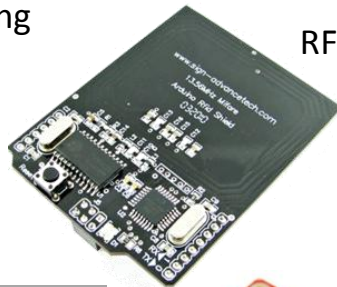


18

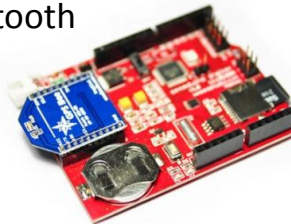
Beschleunigung



RFID



Bluetooth

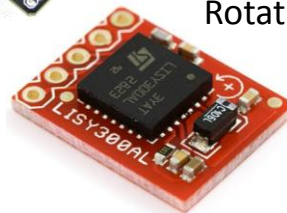


Ethernet

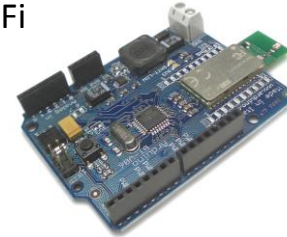


Sensordaten

Rotation

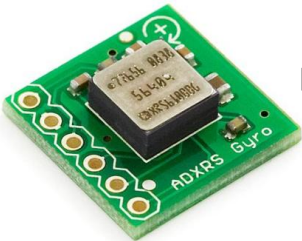


Wi-Fi



Kommunikation

Lage



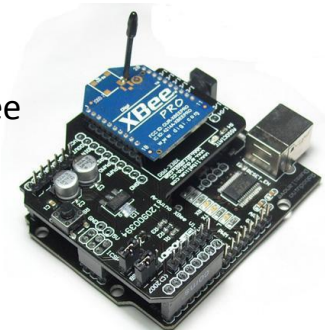
Ultraschall



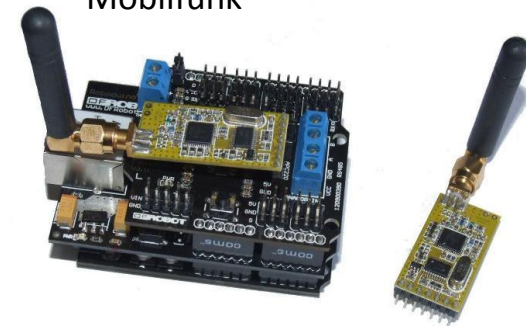
GPS



Xbee



Mobilfunk



Licht, Grauskalierung, Temperatur, Magnetismus, Vibration, Berührung

Open Source Hardware Arduino

Komponenten

Touch Screen



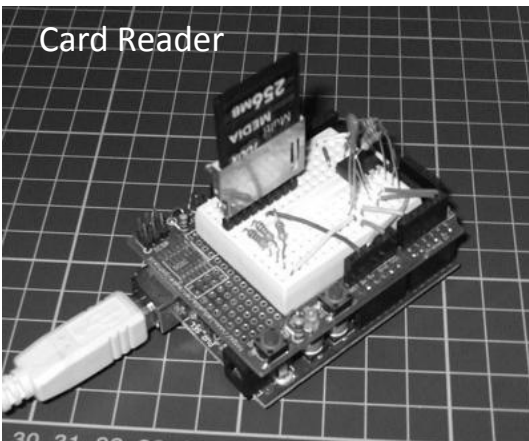
Controller



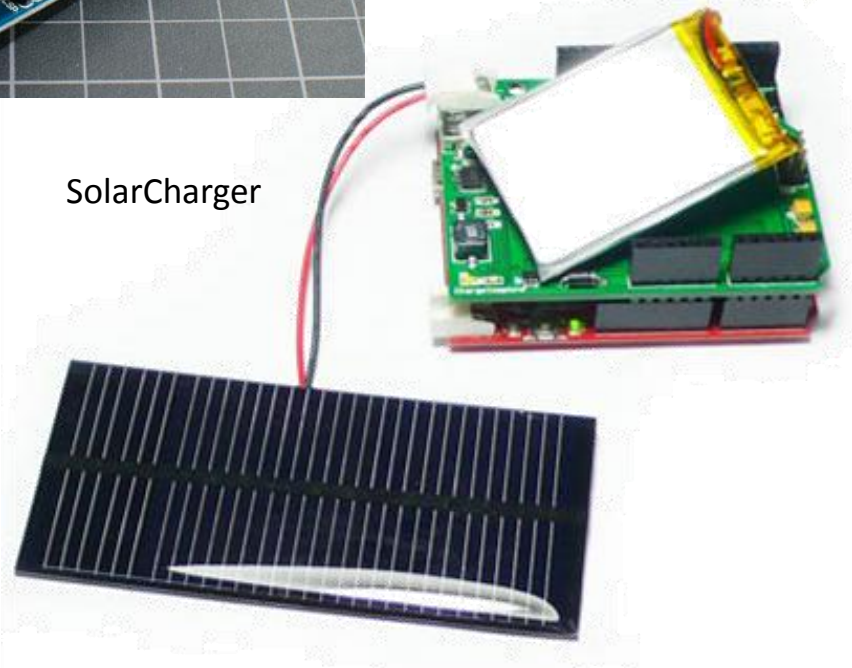
LCD



Card Reader



SolarCharger



Open Source Hardware Arduino

Software



20

```

Arduino - 0011 Alpha
File Edit Sketch Tools Help

Blink

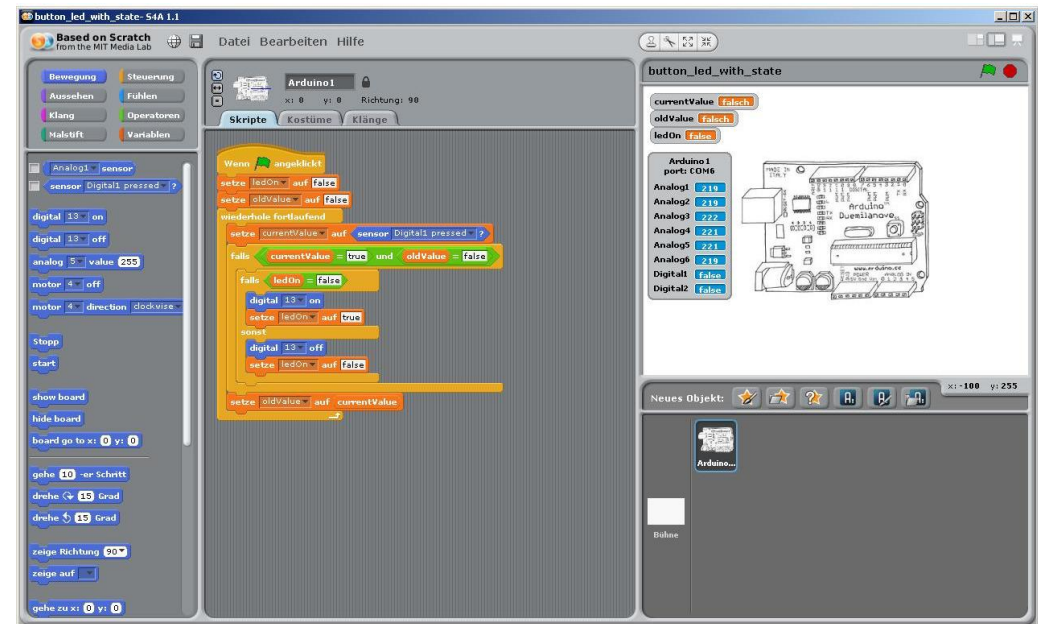
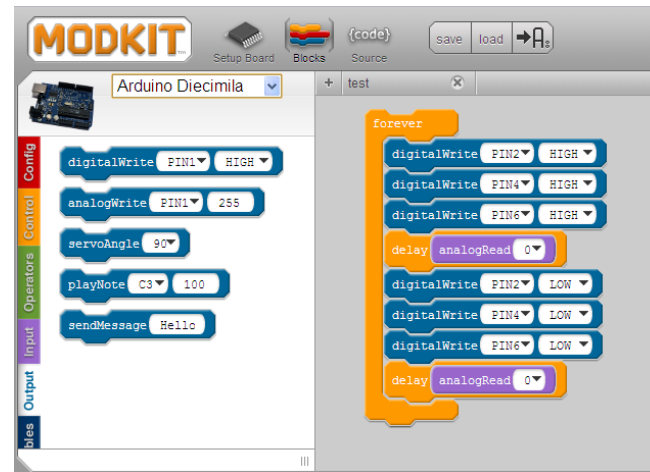
/*
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

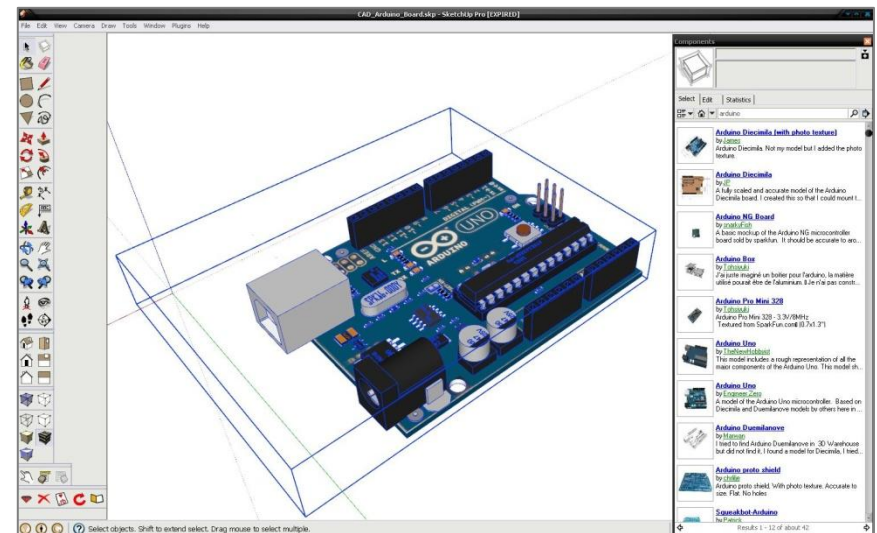
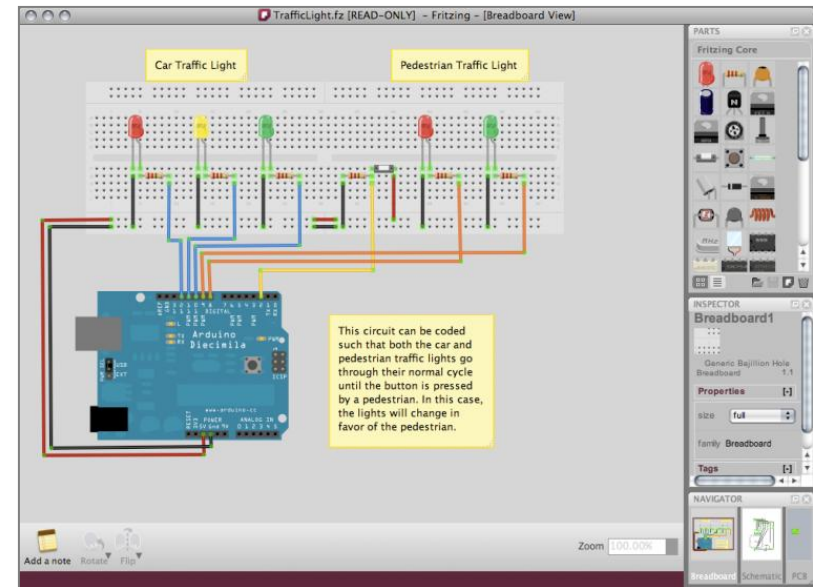
int ledPin = 13;           // LED connected to digital pin 13

void setup()               // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()                // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);                // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);                // waits for a second
}

```





Open Source Hardware Arduino

Community



22



The **playground** is a publicly-editable wiki about [Arduino](#).



Manuals and Curriculum

Board Setup and Configuration

Development Tools

Interfacing With Hardware

- [Output](#)
- [Input](#)
- [Storage](#)
- [Communication](#)
- [Power supplies](#)
- [General](#)

Interfacing with Software

Code Library and Tutorials

Suggestions & Bugs

Electronics Technique

High Resolution Photo of an Arduino Board

:: The Arduino Playground ::

Welcome to the Arduino Playground, a wiki where all the users of [Arduino](#) can contribute and benefit from their collective research.

This is the place to post and share your own code, circuit diagrams, tutorials, DIY instructions, tips and tricks, and after all the hard work, to show off your projects! Anyone can edit and add to the pages here.

Arduino Playground is a **work in progress**. We can use all the help you can give, so please read the [Participate](#) section and get your fingers typing!

NOTE (20081002): Solved the problem for adding content in non-western Latin languages, now it is possible to start typing in any language on the internet [test here](#)

Using Arduino

LEDs and Multiplexing

Controlling lots of inputs and outputs

Last post: [Re: 8x8 RGBMatrix SparkF...](#) by [johnwasser](#) on **Today** at 07:46:28 AM

889 Posts 122 Topics

Displays

LCDs, OLEDs, PAL, NTSC, etc.

Last post: [Re: DIP204B-4NLW LCD pro...](#) by [GaryP](#) on **Today** at 02:34:58 AM

1059 Posts 145 Topics

Audio

Sound processing and generation, using WAV and MP3 players, using MIDI

Last post: [Re: Saving and loading 1...](#) by [WilliamK](#) on **Today** at 06:15:19 AM

483 Posts 64 Topics

Motors, Mechanics, and Power

Controlling big heavy things that move, need high voltages, high current, or both

Last post: [moving Silk Screen with ...](#) by [David Dalmazzo](#) on **Today** at 06:46:11 AM

1457 Posts 213 Topics

Sensors

Getting input from the physical world, recommendations on sensors

Last post: [Re: Suggestions for lowe...](#) by [AWOL](#) on **Today** at 08:39:51 AM

1551 Posts 216 Topics

Networking, Protocols, and Devices

I2C, SPI, XBee, GPS, etc.

Last post: [Re: Looking for a transm...](#) by [johnwasser](#) on **Today** at 08:24:17 AM

1229 Posts 241 Topics



How to Ask Questions the Smart Way

[phalanx](#)

0

440

Tue Mar 08, 2011 11:06 pm
[phalanx](#) ➔

New User Moderation.

[phalanx](#)

14

1891

Wed Feb 16, 2011 12:44 pm
[raphman](#) ➔

Arduino for my project

[Yoda87](#)

0

11

Sat Mar 19, 2011 6:02 pm
[Yoda87](#) ➔

Need help with code!!

[vetenskapsman1](#)

5

52

Sat Mar 19, 2011 6:03 am
[fil-freak](#) ➔

LCD wraps output to 1 or 2 columns

[lcdwrap](#)

3

41

Sat Mar 19, 2011 4:58 am
[lcdwrap](#) ➔

Classroom setting: Arduino vs. BasicStamp Board
[[D](#) Go to page: 1, 2]

[SciCenter](#)

17

256

Fri Mar 18, 2011 9:53 pm
[viskr](#) ➔

mp121 measuring voltages from 13th sensor

[Stevenh](#)

5

56

Fri Mar 18, 2011 9:15 pm
[Stevenh](#) ➔

Graphic compass and inclinometer on Arduino as on iPhone

[Akella](#)

1

21

Fri Mar 18, 2011 2:47 pm
[Chagrin](#) ➔

WiFi shield and Arduino Mega 2560

[siaco](#)

0

9

Fri Mar 18, 2011 10:21 am
[siaco](#) ➔



Gruppe Arduino



slubman I'm done with my first [!arduino](#) and processing experiment, now I have to buy some sensors and also dig into Xbee

[vor ca. 2 Tagen](#) von [Adium](#)



julienguigner Ardunote project: [!Arduino](#) + NanoNote – [http://url.ca/3imv0](#) [#arduino](#) [#nanonote](#) /by @[tuxbrain](#)

[vor ca. 6 Tagen](#) von heybuddy in [Morbihan, Bretagne, Frankreich](#) im [Zusammenhang](#)



slubman Applied the firmware fix for the USB<->Serial chip on my [!arduino](#) uno. Now I can upload sketches and use serial monitor without troubles!

[vor ca. 7 Tagen](#) von gwibber



macno I'm playing with my brand new [!arduino](#) uno

[vor ca. 7 Tagen](#) von Web in [Mailand, Lombardei, Italien](#) im [Zusammenhang](#)



macno I think I'll buy my first [!arduino](#)

[vor ca. 9 Tagen](#) von Web in [Mailand, Lombardei, Italien](#) im [Zusammenhang](#)



tryphon Tryphon Red Button about to be released. Will control your [#PigeBox](#) by using [!Arduino](#) .. and a red button : [http://bit.ly/qiiAT6](#)

[vor ca. 10 Tagen](#) von Web



aztlek Photoduino: free timelapse control for you camera. [http://url.ca/3fvd9](#) [#photography](#) [!photo](#) [!photography](#) [!arduino](#)

1. Wie **industrietauglich** ist die Open Source Hardware Arduino?
2. Welche konkreten **Anwendungsfälle** sind für die Fabrik denkbar?
3. Wie können **Entwickler** weltweit gemeinsam an privaten und industriellen Produktlösungen arbeiten?
4. Inwiefern trägt Arduino zur **Beschleunigung des Produktentwicklungsprozesses** in Unternehmen bei?
5. Ist die Open Source Hardware Arduino geeignet als Informationsflussmodul für **wandlungsfähige Fabrikssysteme**?

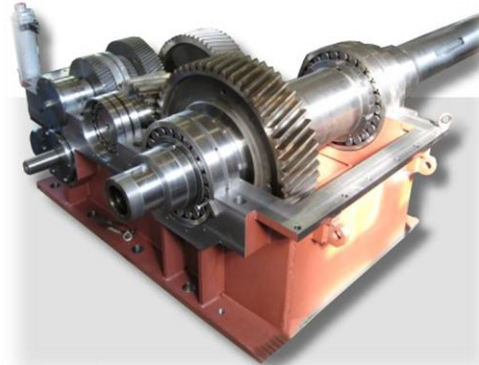
Akkumulator



Öl-Wassertrenner



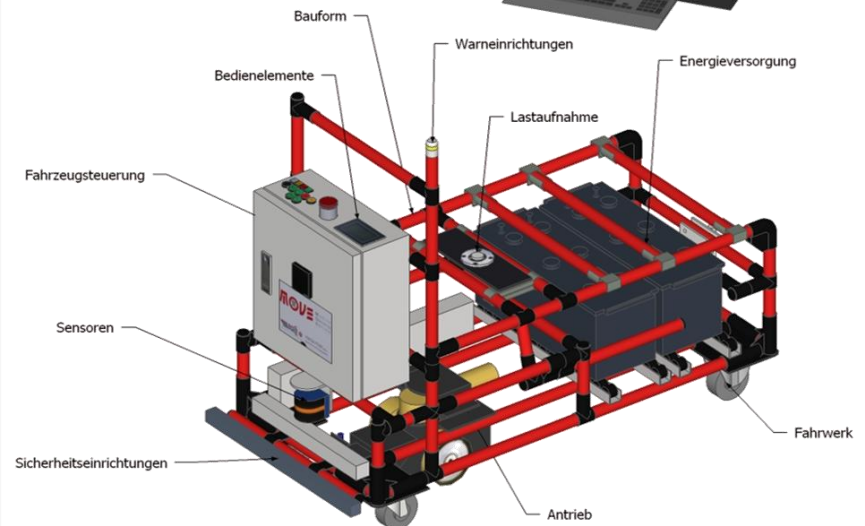
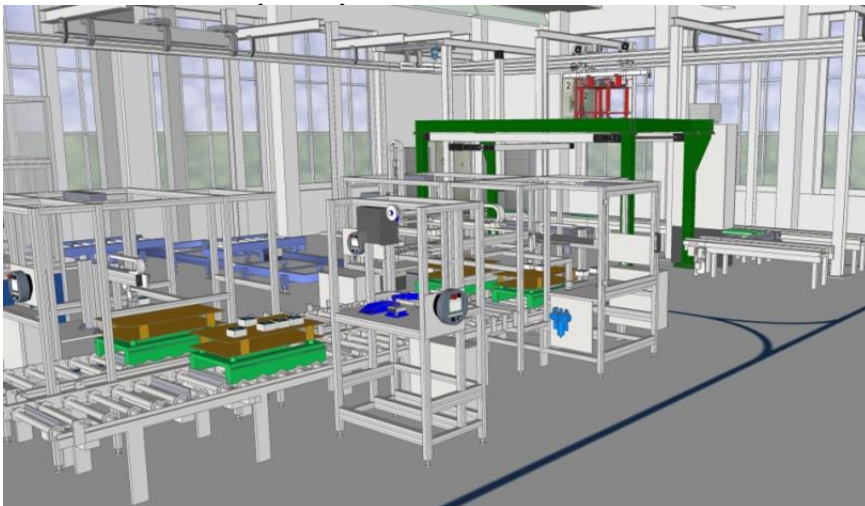
Getriebe



Überwachung



Fertigungslinie



Fahrerloses Transportsystem

Steuerung

Praxis

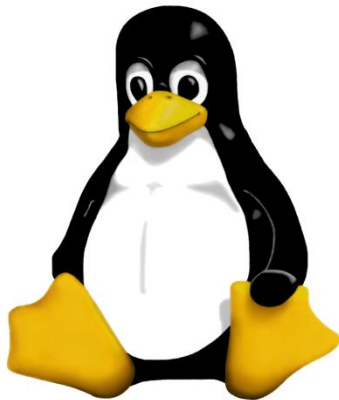
- basierend auf Processing und Wiring
- Arduino IDE:
 - Editor
 - Debugger
 - Compiler
 - Libraries
 - Examples
 - Bootloader
 - Serial Monitor

 if (strcmp(URL, "/") == 0) {
 // Use WiServer's print and println functions to write out
 WiServer.print("<html>");
 WiServer.print("Hello World!");
 WiServer.print("</html>");

 // URL was recognized
 return true;
 }
 // URL not found
 return false;
}

1

- Java Runtime
- avr-gcc-c++, avr-gcc, avr-libc
- Unterstützung FTDI für USB/Serial



- <http://www.arduino.cc/playground/Learning/Linux>

Main Site Blog **Playground** Forum


 search

The playground is a publicly-editable wiki about Arduino.

[Manuals and Curriculum](#)

[Board Setup and Configuration](#)

[Development Tools](#)

[Interfacing With Hardware](#)

- [Output](#)
- [Input](#)
- [Storage](#)
- [Communication](#)
- [Power supplies](#)
- [General](#)

[Interfacing with Software](#)

[Code Library and Tutorials](#)

[Suggestions & Bugs](#)

[Electronics Technique](#)

[Sources for Electronic Parts](#)

[Related Hardware and Initiatives](#)

Installing Arduino on Linux

For more detailed instructions, pick your distribution:

- [Debian](#)
- [Fedora](#)
- [Gentoo](#)
- [openSUSE](#)
- [Slackware](#)
- [Ubuntu](#)
- [Xandros \(Debian derivative\) on Asus Eee PC](#)

Requirements

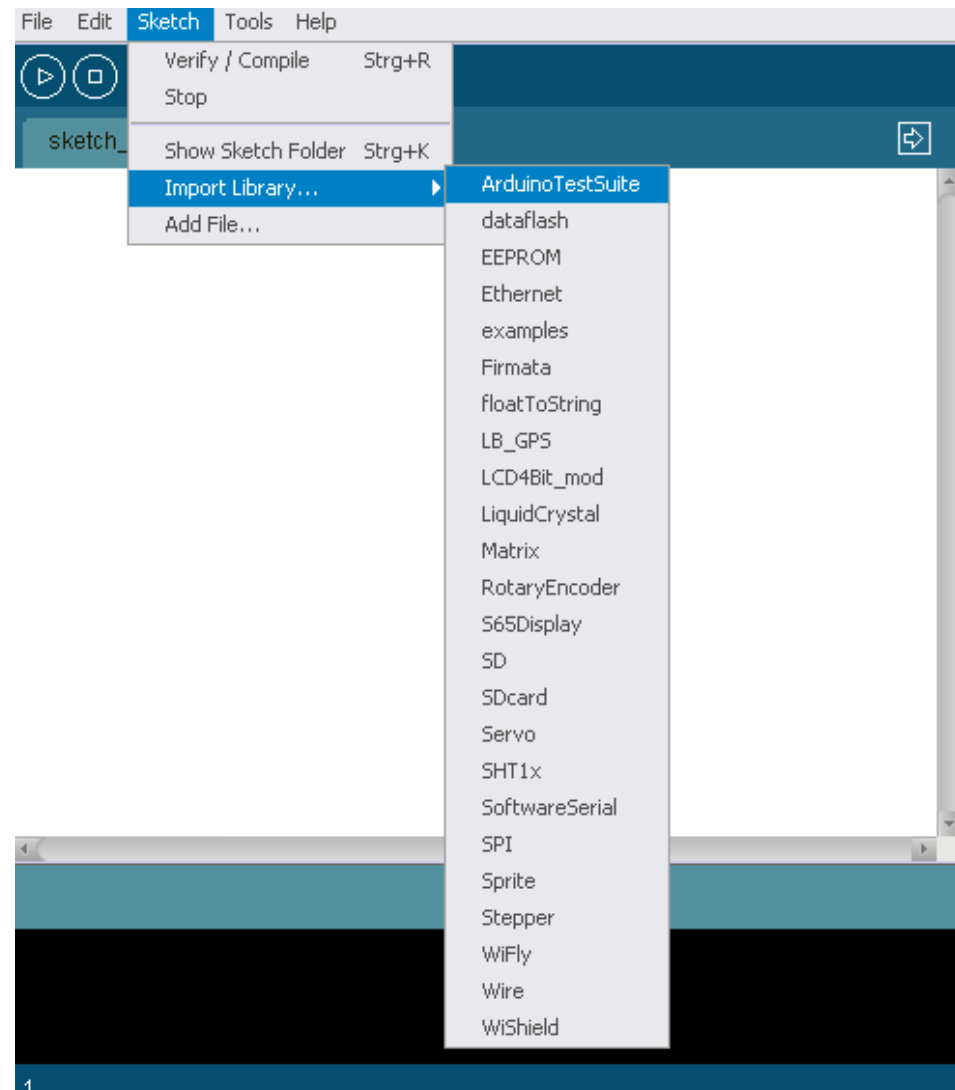
You will need to install some programs to use Arduino under Linux (the way you do this depends on your distribution):

- sun's java runtime (JRE, package sun-java6-jre)
- avr-gcc-c++ (Else you will get this error: Cannot run program "avr-g++": java.io.IOException: Error=2: No such file or directory)

- Kommentare
- Bibliotheken
- Variablen, Konstante, Funktionen
- Setup-Funktion
- Loop-Funktion

```
/* Demo ... */  
  
#include <library.h> // Including ...  
  
const int ledPin = 2;  
  
int inputvariable = 0;  
char helloString[] = "Hello World";  
  
void newfunction(parameter) {  
    command;  
}  
  
void setup() {  
    command;  
}  
  
void loop() {  
    command;  
}
```

- `#include <ArduinoTestSuite.h>`
- C/C++-Quellcode
- `../arduino-0x/libraries/`
- Beispiele:
 - EEPROM
 - Ethernet
 - Servo
 - Stepper
- <http://arduino.cc/en/Reference/Libraries>



- **Datentypen:** boolean, byte, char, float, int, long, ...
- Arithmetische (+ - * / %) und Logische (& | ^ ...) **Operationen**
- Vergleichsoperationen (== != < > <= >=), Boolesche (&& || !), zusammengesetzte (++ += ...) Operationen

- **Entscheidungen:**

```
if (A > B) {  
    Serial.println("A");  
}  
else {  
    Serial.println("B");  
}
```

```
switch (inputvariable) {  
case '1':  
    Serial.println("1");  
    break;  
case '2':  
    Serial.println("2");  
    break;  
}
```

- **Wiederholungen:**

```
while (inputvariable < 5) {  
    Serial.println("more");  
}
```

```
for (int i = 0; i < 10; i++) {  
    Serial.println(i);  
}
```

```
do {  
    Serial.println("more");  
} while (inputvariable < 5);
```

- `void setup ()`, `void loop()`
- Typ entsprechend Rückgabewert
- Beispiele für integrierte Funktionen:
 - Analog I/O
 - Digital I/O
 - Math
 - Serial
 - Time
- <http://arduino.cc/en/Reference/HomePage>

```
int analoginputfunc() {  
    int a;  
    a = analogRead(0);  
    return a;  
}  
  
analoginputfunc();
```

Digital Input - Output

- *pinMode(pin, mode)*
- *digitalRead(pin)*
- *digitalWrite(pin, value)*

```
void setup() {  
    pinMode(13, OUTPUT);  
    pinMode(2, INPUT);  
}  
  
void loop() {  
    int inputvariable = digitalRead(2);  
    digitalWrite(13, HIGH);  
}
```

Konstanten

- **HIGH** (1 / ON / 5 Volt) und **LOW** (0 / OFF / 0 Volt)
- **INPUT** und **OUTPUT**

Analog Input – Output

- keine Deklaration als Ein- oder Ausgang notwendig
- AD-Wandler mit 10 Bit Auflösung
- ***analogRead(pin)***: Rückgabewert von 0 (0 Volt) bis 1023 (5 Volt / Referenzspannung)
- ***analogWrite(pin, value)***: mittels hardwarebasierter Pulsweiten Modulation (PWM) werden “analoge” Werte geschrieben (Wert 0 = 0 Volt, Wert 255 = 5 Volt, dazwischen Wechsel zwischen 0 und 5 Volt)

```
void setup() {  
}  
  
void loop() {  
    int inputvariable = analogRead(0);  
    int outputvariable = map(inputvariable, 0, 1023, 0, 255);  
    analogWrite(9, outputvariable);  
}
```

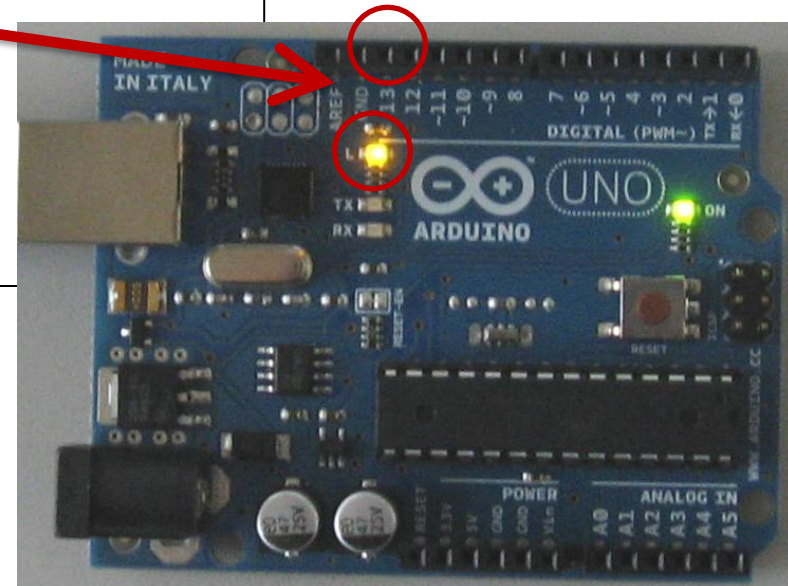
Open Source Hardware Arduino

Beispiel Basics Blink



34

```
/*  
  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  
}
```



Open Source Hardware Arduino



35

Beispiel Digital Button

```

created 2005
by DojoDave <http://www.0j0.org>
modified 28 Oct 2010
by Tom Igoe

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Button
*/

// constants won't change. They're used to
// set pin numbers:
const int buttonPin = 2;    // the number of the pushbutton pin
const int ledPin = 13;      // the number of the LED pin

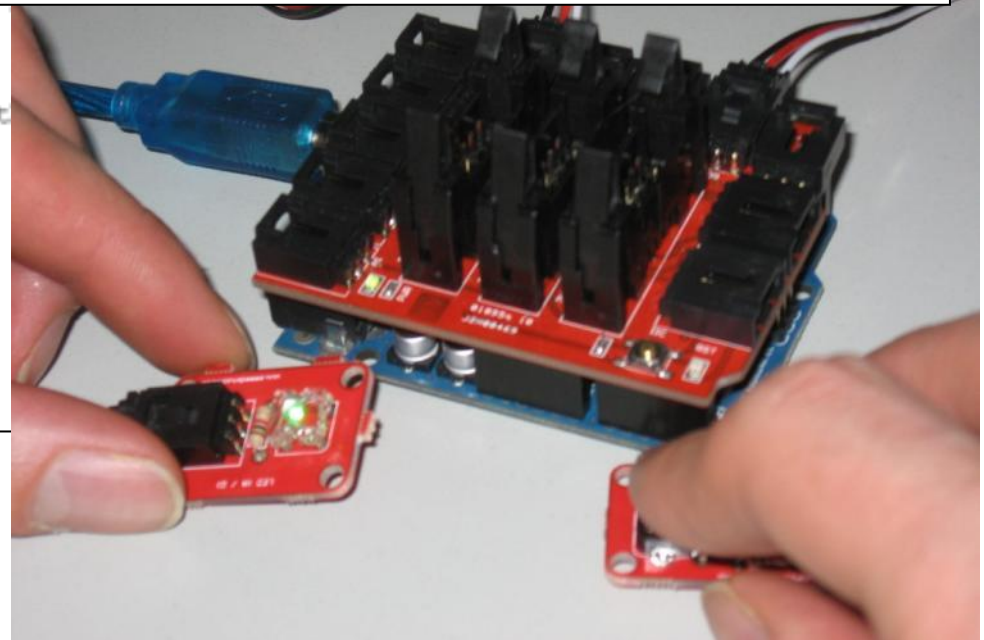
// variables will change:
int buttonState = 0;        // variable for reading the pushbutton state

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}

```



Open Source Hardware Arduino

Webserver per WLAN



```
// Wireless configuration parameters -----
#define WIRELESS_MODE_INFRA 1
#define WIRELESS_MODE_ADHOC 2
unsigned char local_ip[] = {192,168,10,20}; // IP address of WiShield
unsigned char gateway_ip[] = {192,168,10,1}; // router or gateway IP address
unsigned char subnet_mask[] = {255,255,255,0}; // subnet mask for the local network
const prog_char ssid[] PROGMEM = {"wlanrouter"}; // max 32 bytes
unsigned char security_type = 2; // 0 - open; {1 - WEP}; 2 - WPA; 3 - WPA2
// WPA/WPA2 passphrase
const prog_char security_passphrase[] PROGMEM = {"blabla"}; // max 64 characters

// WEP 128-bit keys
// sample HEX keys
prog_uchar wep_keys[] PROGMEM = { 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d, // Key 0
                                   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, // Key 1
                                   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, // Key 2
                                   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 // Key 3
                                   };

// setup the wireless mode
// infrastructure - connect to AP
// adhoc - connect to another WiFi device
unsigned char wireless_mode = WIRELESS_MODE_INFRA;

unsigned char ssid_len;
unsigned char security_passphrase_len;
```

```
// This is our page serving function that generates web pages
boolean sendMyPage(char* URL) {
    // Check if the requested URL matches "/"
    if (strcmp(URL, "/") == 0) {
        // Use WiServer's print and println functions to write out the page content
        WiServer.print("<html><body style=\"background:#000000\"><span style=\"color:#01DF01\">");
        WiServer.print("<br><h1>##### DEMO #####</h1><br>");
        WiServer.print("<h2><br>IP: ");
        WiServer.print(local_ip_str0);
        WiServer.print(".");
        WiServer.print(local_ip_str1);
        WiServer.print(".");
        WiServer.print(local_ip_str2);
        WiServer.print(".");
        WiServer.print(local_ip_str3);
        WiServer.print("<br></h2>");
        WiServer.print("<h2><br>Licht (Sensorwert): ");
        WiServer.print("<FONT COLOR=\"#FF0000\">");
        WiServer.print(statusmessage);
        WiServer.print(" ");
        WiServer.print(sensorvalue);
        WiServer.print(" ");
        WiServer.print("</FONT><br></h2>");
        WiServer.print("<br><h1>##### DEMO #####</h1><br>");
        WiServer.print("</html>");

        // URL was recognized
        return true;
    }
    // URL not found
    return false;
}
```

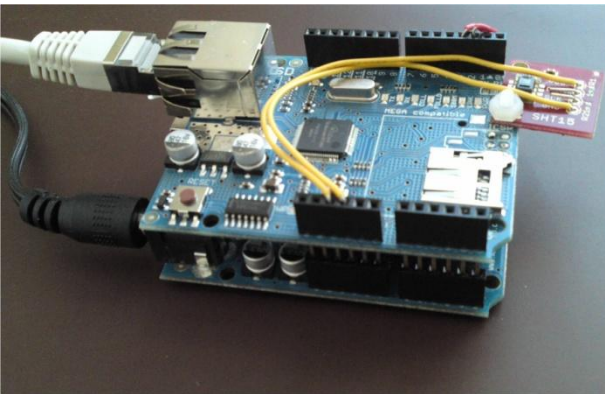
Open Source Hardware Arduino

Kühlungs-Überwachung von Servern



Middleware For Open Source Devices

company: TU Chemnitz - Ireko Projekt
[Logout](#)



Device data
▶ general device information
Device settings
▶ create a new device
▶ notification conditions
▶ notification accounts
Logging
▶ dashboard
▶ logged device data
▶ published messages
▶ executed events
▶ recognized errors
Client data
▶ publish connection types
▶ client settings
▶ login data



The first version is being developed within the scope of the project IREKO which is funded by the European Social Fund and the Free State of Saxony.



This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License.

Sensor status view				→ switch to editmode
device name	last message	status	intervall	
Arduino_IBF_Serverraum	17.03.2011 - 15:53:27	online	60	show details
testsensor	17.03.2011 - 00:58:50	online	60	show details
testsensor2		online	60	show details

Message log (last 20 messages)

[Today - 12:08:56] via ArduinoTUC, ArduinoTUC Statusnet
44.02%: bitte das [#Fenster](#) öffnen; 24.14C: bitte [#Klimaanlage](#) anschalten

[Today - 12:07:31] via ArduinoTUC, ArduinoTUC Statusnet
[ireko_sensor.1.luft]?: bitte das [#Fenster](#) öffnen; [ireko_sensor.1.temp]C: bitte [#Klimaanlage](#) anschalten

[Today - 12:06:07] via ArduinoTUC, ArduinoTUC Statusnet
[ireko_sensor.1.luft]?: bitte das [#Fenster](#) öffnen; [ireko_sensor.1.temp]C: bitte [#Klimaanlage](#) anschalten

[Today - 12:04:42] via ArduinoTUC, ArduinoTUC Statusnet
[ireko_sensor.1.luft]?: bitte das [#Fenster](#) öffnen; [ireko_sensor.1.temp]C: bitte [#Klimaanlage](#) anschalten

[Today - 12:03:17] via ArduinoTUC, ArduinoTUC Statusnet
[ireko_sensor.1.luft]?: bitte das [#Fenster](#) öffnen; [ireko_sensor.1.temp]C: bitte [#Klimaanlage](#) anschalten

[Today - 12:01:52] via ArduinoTUC, ArduinoTUC Statusnet
[ireko_sensor.1.luft]?: bitte das [#Fenster](#) öffnen; [ireko_sensor.1.temp]C: bitte [#Klimaanlage](#) anschalten

[Today - 12:00:28] via ArduinoTUC, ArduinoTUC Statusnet
[ireko_sensor.1.luft]?: bitte das [#Fenster](#) öffnen; [ireko_sensor.1.temp]C: bitte [#Klimaanlage](#) anschalten

[Today - 11:59:03] via ArduinoTUC, ArduinoTUC Statusnet
[ireko_sensor.1.luft]?: bitte das [#Fenster](#) öffnen; [ireko_sensor.1.temp]C: bitte [#Klimaanlage](#) anschalten



MONITORING WITH ARDUINOS

ibfserverraum ist Mitglied bei arduino.tuc, einem Mikroblogging-Dienst auf Basis der freien Software StatusNet. Werde ich folgen! (Mehr Informationen)

Profil Abonnements Abonnierten Gruppen



ibfserverraum
(Sensor im Serverraum des IBF)
Chemnitz, Deutschland

Abonnieren



44.02%: bitte das [#Fenster](#) öffnen; 24.14C: bitte [#Klimaanlage](#) anschalten
vor ca. 4 Stunden von Middleware For Open Source Devices in Chemnitz, Freistaat Sachsen, Deutschland

[ireko_sensor.1.luft]?: bitte das [#Fenster](#) öffnen; [ireko_sensor.1.temp]C: bitte [#Klimaanlage](#) anschalten
vor ca. 4 Stunden von Middleware For Open Source Devices in Chemnitz, Freistaat Sachsen, Deutschland

[ireko_sensor.1.luft]?: bitte das [#Fenster](#) öffnen; [ireko_sensor.1.temp]C: bitte [#Klimaanlage](#) anschalten
vor ca. 4 Stunden von Middleware For Open Source Devices in Chemnitz, Freistaat Sachsen, Deutschland

- OS Hardware Arduino:
 - einfach – modular – günstig – offen – universell einsetzbar
 - große Community

- Forschungsbereich
 - Industrietauglichkeit
 - Anwendungsfälle
 - Gemeinsame Entwicklung (privat, industriell)
 - Beschleunigung des Produktentwicklungsprozesses
 - Anpassbares Informationsflussmodul?

- Ausblick - Open Source Hardware Entwicklungsrichtung??

Name	Link
Professur Fabrikplanung und Fabrikbetrieb	http://www.tu-chemnitz.de/mb/FabrPlan/
Arduino-industry-cases	http://code.google.com/p/arduino-industry-cases/
Arduino-industry-cases-forum	http://groups.google.com/group/arduino-industry-cases-forum?pli=1
Arduino	http://arduino.cc/
Makezine	http://blog.makezine.com/archive/category/arduino
VirtualBreadboard	http://virtualbreadboard.net/
Modk.it	http://www.modk.it/
40 Arduino-Projekte	http://hacknmod.com/hack/top-40-arduino-projects-of-the-web/



Technische Universität Chemnitz
Professur Fabrikplanung und Fabrikbetrieb
Erfenschlager Straße 73
09125 Chemnitz (Deutschland)
<http://www.tu-chemnitz.de/mb/FabrPlan/>

Andreas Merkel

0371 531-37939

andreas.merkel@mb.tu-chemnitz.de

Raum - E 108

Hendrik Hopf

0371 531-38527

hendrik.hopf@mb.tu-chemnitz.de

Raum - C 112