

Arduino

Was ist Physical Computing?

Interaktive, physische Systeme durch die Verwendung von Hard- **und** Software.

Diese Systeme agieren im wirklichen Leben und können auf die Umwelt einwirken.

Vielfach sind es Systeme, die sich mit der Beziehung zwischen Menschen **und** der digitalen Welt befassen.

Physical Computing

Physical Computing findet man häufig im Hobby- und künstlerischen Bereich.

Zentrale Bausteine sind neben der Software

- Sensoren
- Microcontroller und
- Geräte wie Leuchtdioden und Motoren.

Der Raspberry Pi

Über GPIO-Pins kann der Raspberry Pi auch grundsätzlich für Physical Computing Projekte eingesetzt werden.

- Die Handhabung ist jedoch umständlich.
- Die Möglichkeiten sind begrenzt.

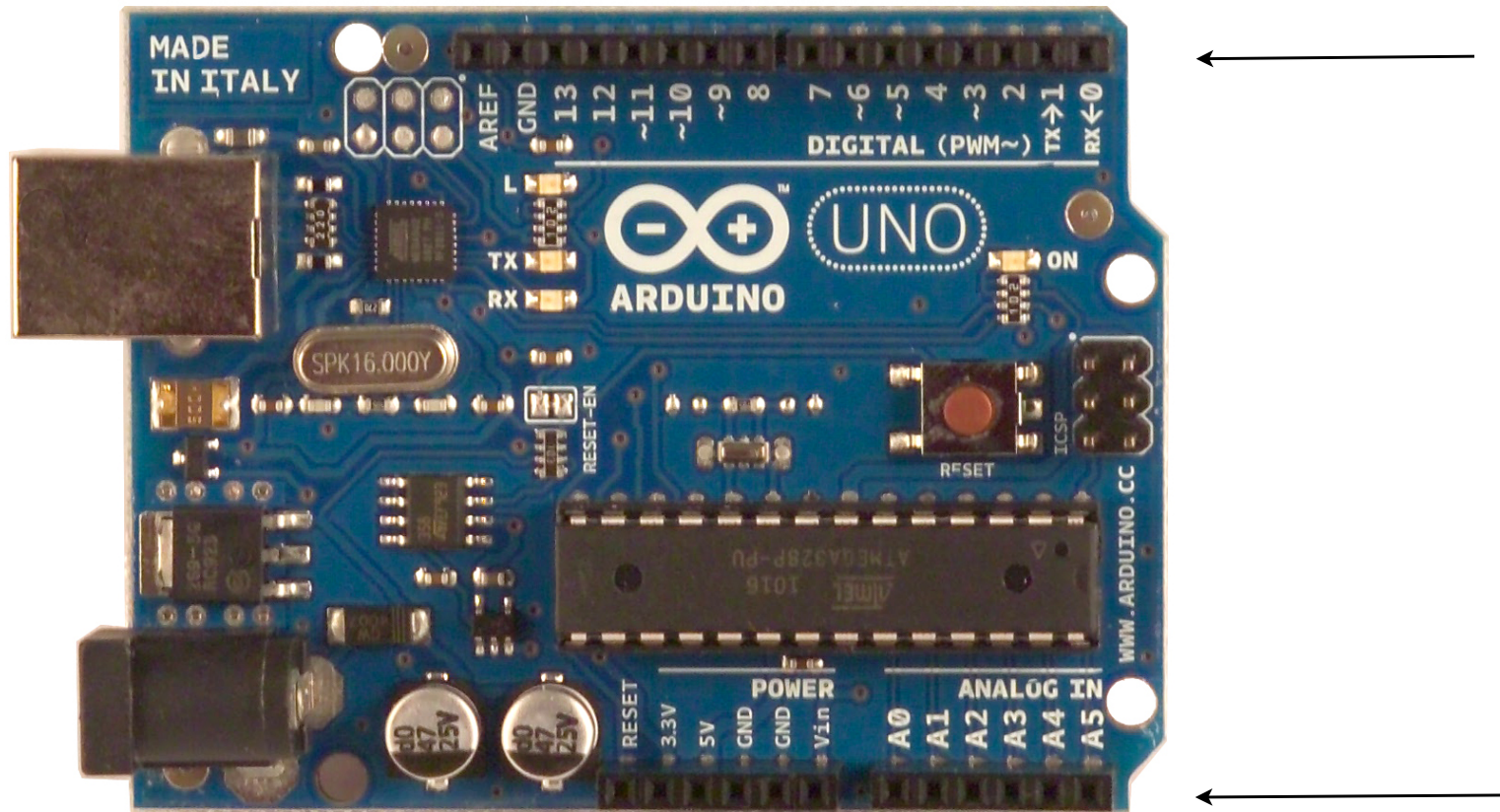
Für Physical Computing Projekte haben sich Plattformen wie **Arduino** etabliert.

Was ist Arduino?

Ähnlich wie der Raspberry Pi ist der Arduino eine Platine mit Peripherie. Sie enthält in der Regel

- einen AVR-Mikrocontroller
- Peripherie und
- Möglichkeiten die Peripherie zu erweitern

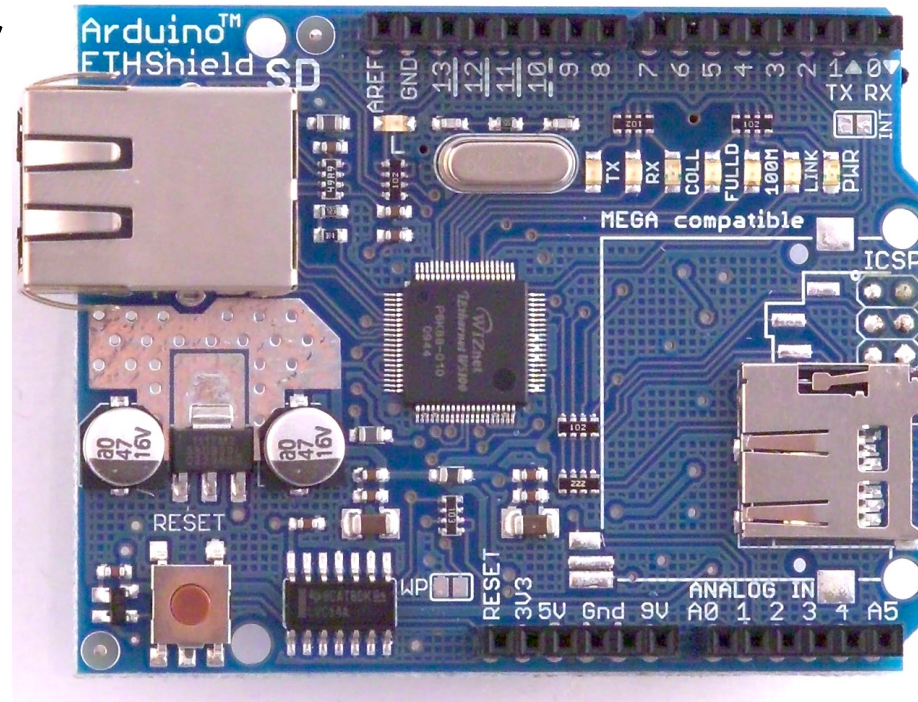
Der Arduino UNO



Über die markierten Pins können Sensordaten gelesen werden und Geräte gesteuert werden.

Shields

Ethernet Adapter



Reader für SD-Karten



Funktionalität kann etwas über aufsteckbare ‚Shields‘ nachgerüstet werden. Hier ein Ethernet Shield

Ressourcen

Der Arduino ist ein kleiner Computer mit Ein- und Ausgängen, die über den eingebauten Microcontroller programmiert werden können.

Typische Konfiguration:

- ▶ 8-16 **KB** Flash Memory
- ▶ 1 KB RAM
- ▶ 16 MHz Uhr
- ▶ 13 digitale Ein- und Ausgänge
- ▶ 5 analoge Ein- und Ausgänge
- ▶ USB Typ-B Verbindung

Je nach Modell ist eine andere Konfiguration möglich.

Ressourcen

Die Ressourcen eines Arduino sind in der Regel **seeeehr** begrenzt.

Arduinos können mit mobilen Systemen wie Raspberry Pis oder Android Geräten kombiniert werden.

Das Beste aus beiden Welten

Mobile Geräte werden oft zusammen mit der Linux-Plattform betrieben. Diese ist nur begrenzt **echtzeitfähig**:

Software hat teilweise die Anforderung, dass Aufgaben innerhalb eines festgelegten Intervalls abgeschlossen werden müssen (Beispiel: Ampelanlage).

- Microcontroller können diese Garantien bieten,
- Betriebssysteme wie Linux nur sehr bedingt.

Die Dokumentation

Die Bau- und Schaltpläne für die Arduino-Boards sind frei und offen.

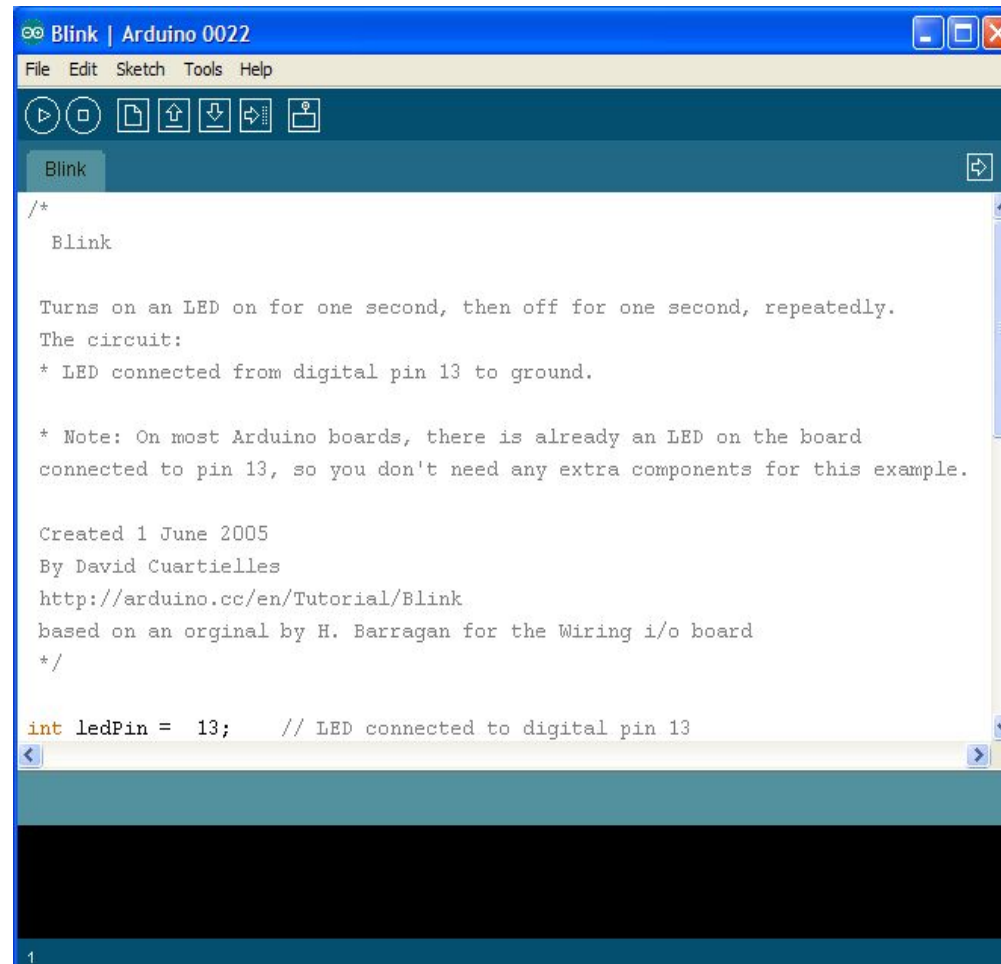
Jeder darf Arduino-Boards nachbauen und vertreiben.

Die Arduino-IDE ist

- ▶ sehr gut dokumentiert
- ▶ sehr verbreitet
- ▶ auf Windows, Linux und Mac OS X verfügbar.

Die Arduino-IDE

Zur Arduino Plattform gehört auch die leicht bedienbare Arduino IDE

A screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 0022". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for running, stopping, saving, opening, and other sketch-related actions. The main text area displays the "Blink" sketch, which is a classic example for turning an LED on and off. The code includes a multi-line comment explaining the purpose, circuit, and history of the sketch. At the bottom, the first line of C++ code is visible: `int ledPin = 13; // LED connected to digital pin 13`. The status bar at the very bottom shows the number "1".

```
/*  
  Blink  
  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  The circuit:  
  * LED connected from digital pin 13 to ground.  
  
  * Note: On most Arduino boards, there is already an LED on the board  
  connected to pin 13, so you don't need any extra components for this example.  
  
  Created 1 June 2005  
  By David Cuartielles  
  http://arduino.cc/en/Tutorial/Blink  
  based on an original by H. Barragan for the Wiring i/o board  
  */  
  
int ledPin = 13; // LED connected to digital pin 13
```

Die Programmiersprache

Zur Arduino Plattform gehören ebenfalls

- eine eigene Sprache, die sich sehr stark an C und C++ orientiert.
- einige Standardbibliotheken.

Die Sprache und die Bibliotheken sind dokumentiert unter <http://arduino.cc/en/Reference/HomePage>

Neben den Standardbibliotheken gibt es zahlreiche Bibliotheken aus der Community.

Der Lebenszyklus eines Sketches

Der Code

- wird in der IDE am Host entwickelt
- über USB auf das Board geladen

Programme, die mit Hilfe der IDE entwickelt werden, werden auch als **Sketche** bezeichnet.

Auf dem Arduino wird gleichzeitig nur **ein einziges** Programm gespeichert.

Es wird

- beim Start des Arduino ausgeführt.
- durch Drücken der Reset-Taste neu gestartet.

Einfache Software-Architektur

Arduino Sketches haben oft eine einfache Standardarchitektur:

```
void setup(){  
    ....  
}  
void loop(){  
    ....  
}
```

- Beim Programmstart wird **setup** aufgerufen. Typischerweise wird hier initialisiert.
- Anschließend wird immer wieder die **loop**-Funktion aufgerufen.

Ein Beispiel-Sketch

```
void setup() {  
    pinMode(13, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

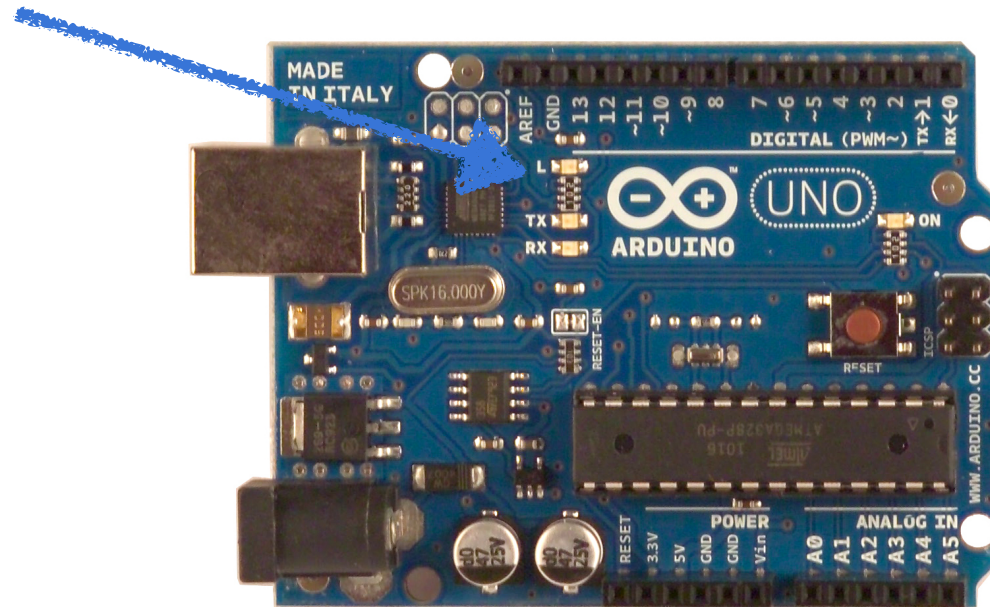
Pin 13 soll für die Ausgabe, nicht für die Eingabe genutzt werden.

Die Funktion **loop**

- aktiviert Pin 13 für 1 Sekunde und
- deaktiviert Pin 13 dann für 1 Sekunde.

Ein Beispiel-Sketch

Unter den digitalen Pins spielt Pin 13 eine Sonderrolle:
Hier ist eine eingebaute LED angeschlossen.
Der Beispiel-Sketch lässt diese LED periodisch leuchten.



Preise

Arduinos sind günstig:

- Der (originale) Arduino Uno kostet etwa 20€.
- Der Nachbau ist schon für 14€ erhältlich
- Ein Ethernet-Shield kostet 12 €

Arduinos sind aber ohne zusätzliche elektronische Bauteile kaum nutzbar.

Weihnachten steht vor der Tür

Für den Anfang empfiehlt sich die Anschaffung eines Starter Kits (ca 100€) mit zahlreichen Bauteilen und dokumentierten Beispielprojekten.



Die Philosophie

„The Arduino philosophy is based on **making** design rather than talking about it“

Make!

In Ländern wie den USA existiert eine ausgeprägte ‚Makerszene‘. Dazu ein Kommentar auf Spiegel-Online:

„...Seit Jahren beobachte ich die Makerszene in den USA und bin einfach neidisch. Es existiert einfach eine Kultur des Tüftelns und die wird auch gefördert. So waren Google, Facebook und viele andere möglich. Namhafte Universitäten bieten bei Udacity und EDX viel kostenlose Kurse an, die mit Innovation im engeren und weiteren Sinne zu tun haben. Die Jugendlichen werden begeistert und sind die künftigen Steve Jobs, Larry Page und Sergey Brin. Und bei uns? Da werden Mädchen von Heidi Klum zu Modells ausgebildet und Jungs vom Jogi zu Fußballspielern - das wars. Was ist aus dem Land der Dichter und Denker geworden?...“

Wie professionell ist Arduino?

Wichtige Instrumente wie Interrupts oder Timer stehen über die Arduino-Plattform nicht zur Verfügung.

Dadurch sind Arduino-Projekte nur sehr begrenzt für kommerzielle Produkte nutzbar.

Arduino bietet aber einen sehr guten Einstieg in moderne Technologien wie

- Internet of Things
- Industrie 4.0