

Special Aspects of HCI: Prototyping with Arduino

Using the Arduino Open Hardware Platform to sketch and develop physical
interactions and tangible user interfaces

Today: Introduction

About this course

- Lecture
 - Theoretical background and hand on sessions
- Project Work
 - Create a interactive thing including a Arduino (or some other kind of microcontroller)
 - Presenting your project idea in the first week of June
 - In groups with up to 3 persons
 - Document your process of creating
 - Fix deadline: 30.9.2018 (early submission is possible)

Timetable

Session	Date	Topic
1		Introduction
2		Crash course electrical engineering
3		Analog vs digital signals
4		Communication
5		
6		Presentation of project ideas
7		
8		
9		

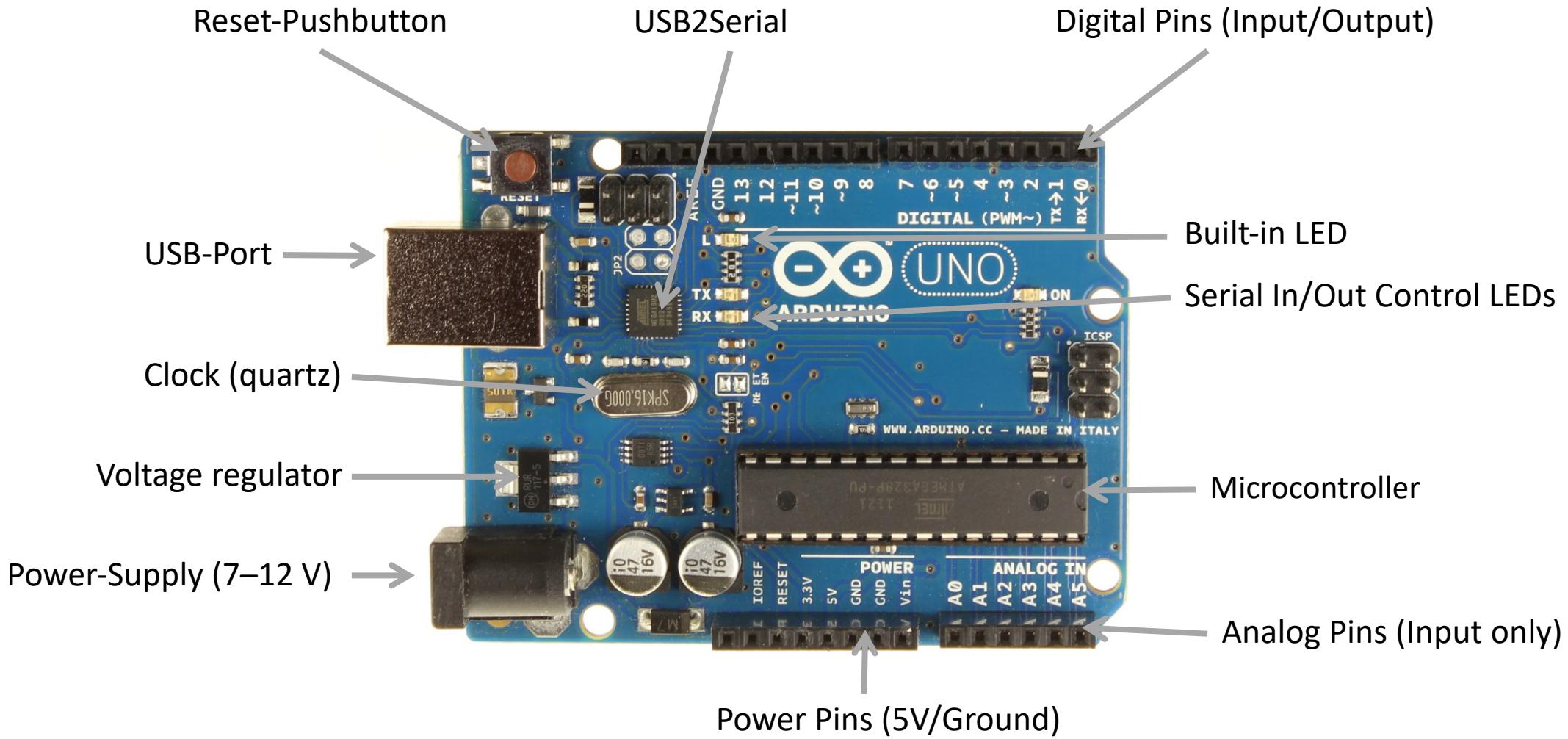
After that: project work.

Old projects TBD

Where to get information about Arduinos and inspiration for your project?

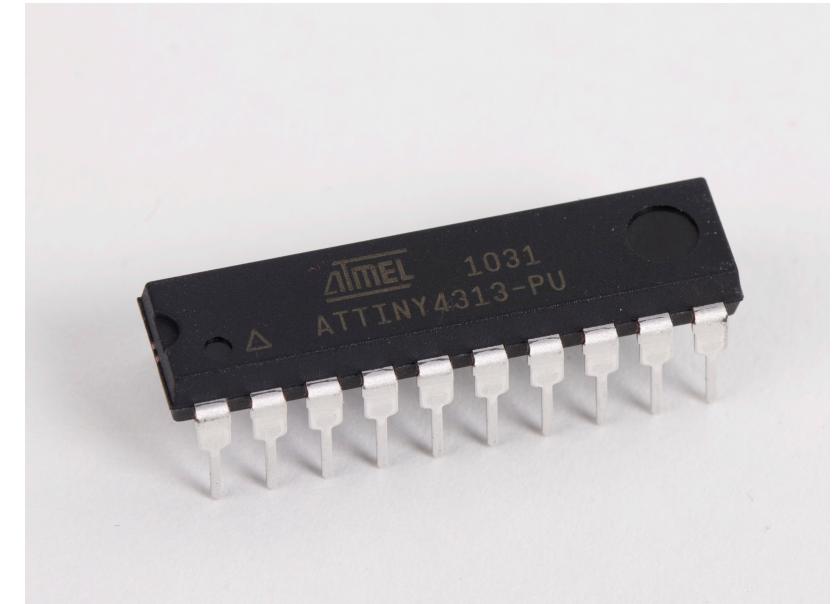
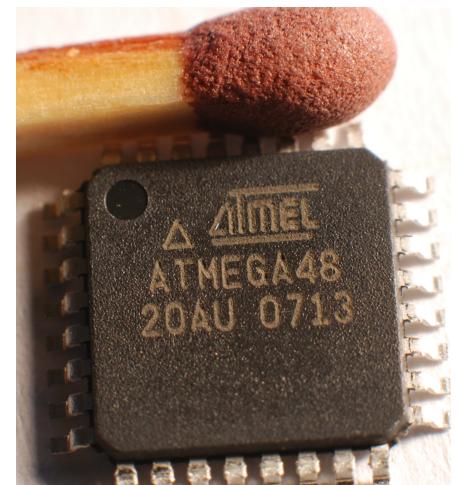
- Books and magazines
 - Arduino Cookbook (Michael Margolis, O'REILLY)
 - Programming Interactivity (Joshua Noble, O'REILLY)
 - MAKE: MAGAZINE
- Internet
 - arduino.cc
 - instructables.com

Let's have look at an Arduino Uno



What is a microcontroller?

- Small computer on a single integrated circuit (IC)
- Contains a processor core, memory, and programmable input/output peripherals
- Program memory is often included on chip
- Typically small amount of RAM (4-8kb in Arduino ATmega case)
- Microcontrollers are designed for embedded applications, usually programmed for one specific task
- Usually just one process at a time



1. „[Chip](#)“ by Henner Zelleris licensed under [CC BY-SA 2.0](#).
2. „[ATTiny4313-PU](#)“ by Windell Oskay licensed under [CC BY 2.0](#).

Arduino Platform



- Open source hardware and software platform
- Designed to make the process of using electronics in multidisciplinary projects more accessible
- Based on different Atmel AVR microcontrollers
- Make the functions of the microcontroller easily accessible through:
 - Pin bar for input and output
 - USB interface for programming
 - Power supply
 - Reset-Button

1. „[Genuino UNO](#)“ by Arduino licensed under [CC BY-SA 3.0](#).
2. „[Arduino IDE](#)“ by Wlanowski licensed under [CC BY-SA 4.0](#).

The screenshot shows the Arduino IDE interface with the following details:

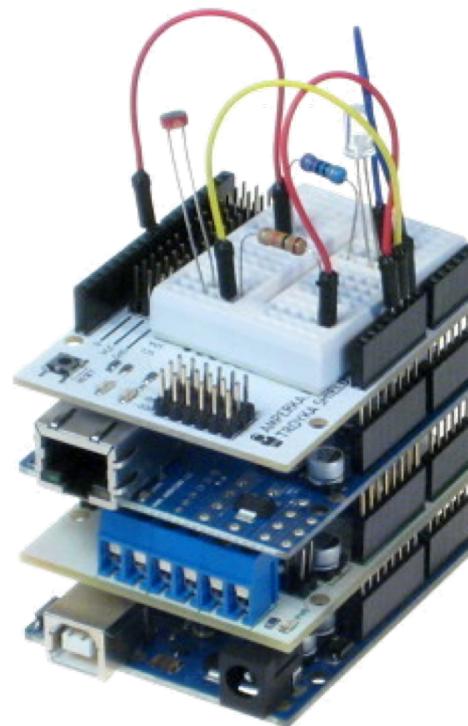
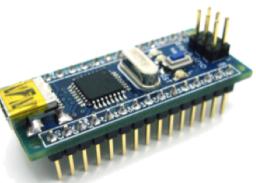
- Title Bar:** Fade | Arduino 1.6.0
- Menu Bar:** Datei Bearbeiten Sketch Werkzeuge Hilfe
- Toolbar:** Includes icons for Open, Save, Upload, and Download.
- Sketch Editor:** Displays the code for the 'Fade' sketch:

```
pinMode(led, OUTPUT);
void loop() {
    analogWrite(led, brightness);
    brightness = brightness + fadeAmount;
    if (brightness == 0 || brightness == 255) {
        fadeAmount = -fadeAmount ;
    }
    delay(30);
}
```
- Compile Status:** Kompilieren abgeschlossen.
- Memory Usage:** Der Sketch verwendet 2.020 Bytes (0%) des Programmspeicherplatzes. Das Maximum sind 253.952 Bytes.
- Global Variables:** Globale Variablen verwenden 15 Bytes (0%) des dynamischen Speichers, 8.177 Bytes für lokale Variablen verbleiben. Das Maximum sind 8.192 Bytes.
- Bottom Status:** Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM3

Arduino Boards & Shields



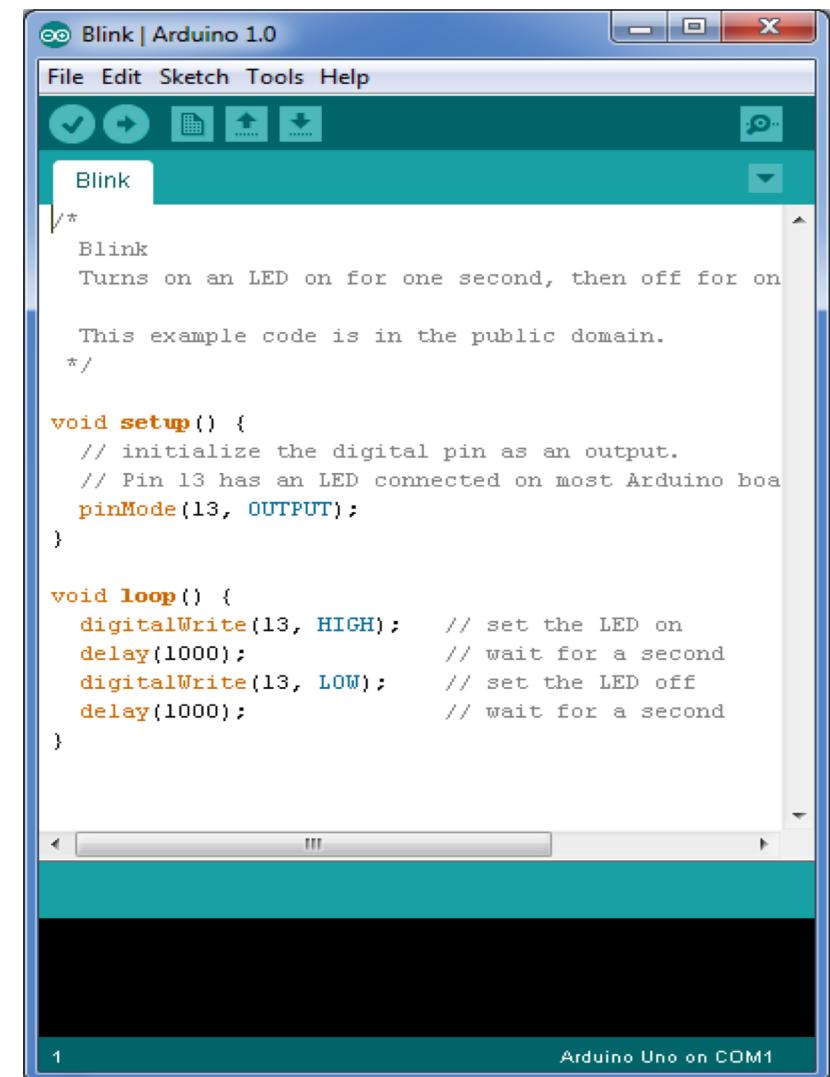
- Arduino Duemilanove
 - ATmega168/328P
 - 14/6 Pins (digital/analog)
- Arduino Mega(2560)
 - ATmega1280/2560
 - 54/16 Pins (digital/analog)
- Arduino Nano
 - ATmega168 or ATmega328
 - 14/8 Pins (digital/analog)
- Arduino Mini Pro
 - ATmega168
 - 14/6 Pins (digital/analog)



- Shields are stackable
- Shields adding functionality to Arduino boards like:
 - Networking
 - Controlling electrical motor
 - Sound
 - ...

Arduino programming

- Arduino programming language is a combination of C and C++
- Arduino IDE
- Plugin for Eclipse and Visual Studio
- Each Arduino program have to consists at least out of a setup and a loop function
 - void setup() – initializing the microcontroller
 - void loop() – measuring and processing input generate output



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.0". The menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar has icons for Save, Load, Upload, and Refresh. The code editor window displays the "Blink" sketch. The code is as follows:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second.
 *
 * 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);
}

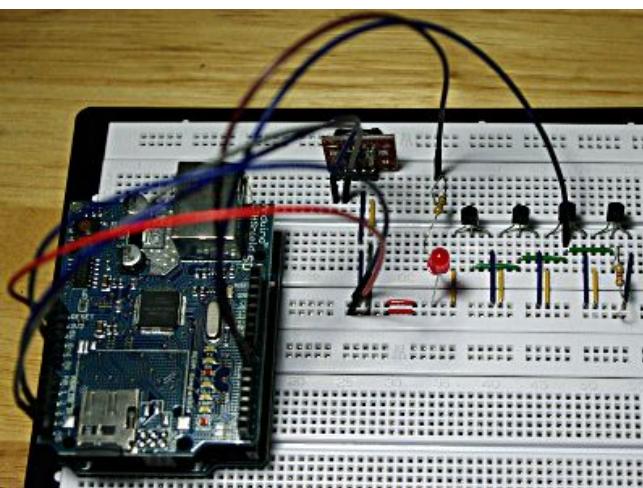
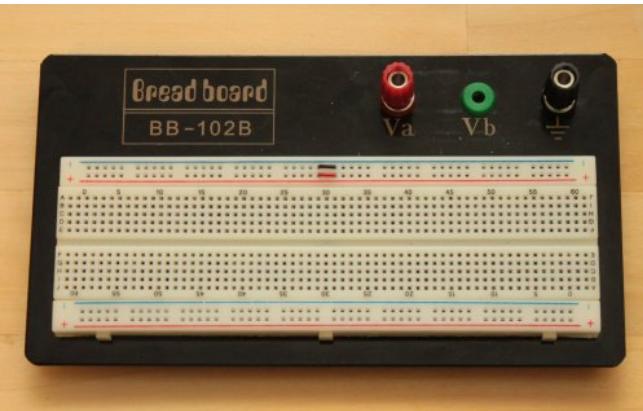
void loop() {
    digitalWrite(13, HIGH);      // set the LED on
    delay(1000);                // wait for a second
    digitalWrite(13, LOW);       // set the LED off
    delay(1000);                // wait for a second
}
```

The status bar at the bottom indicates "1" and "Arduino Uno on COM1".

IPO Model

- Measure **Input**
 - Analog and digital: Buttons, temperature, light, sound, serial devices, ...
- **Process**
 - Process input through the program code
- Generate **Output**
 - Digital: High/Low, PWM, serial signals

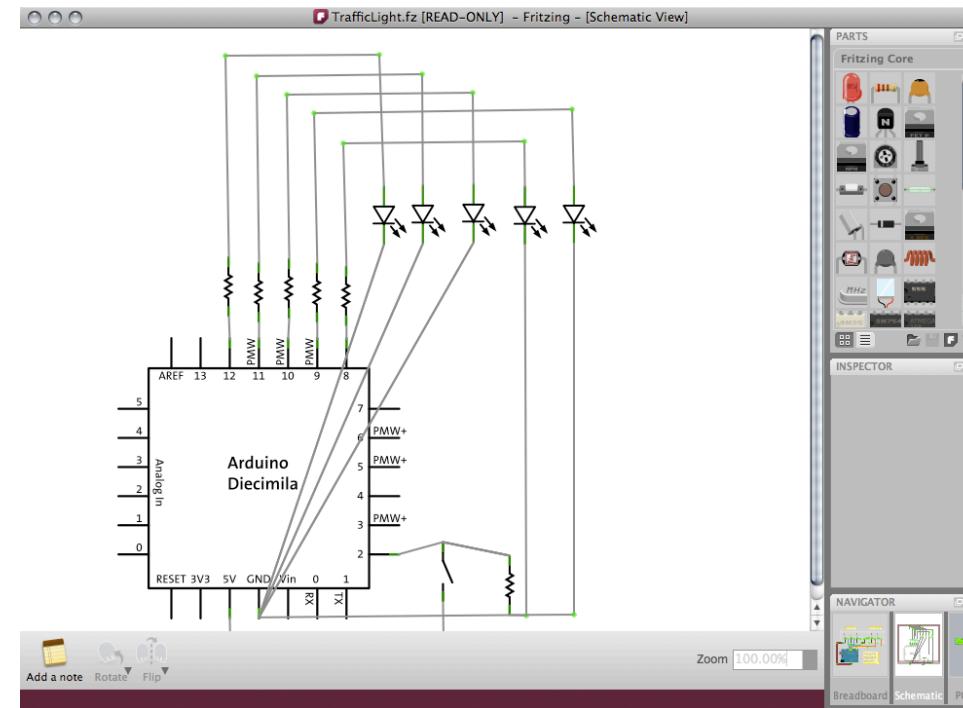
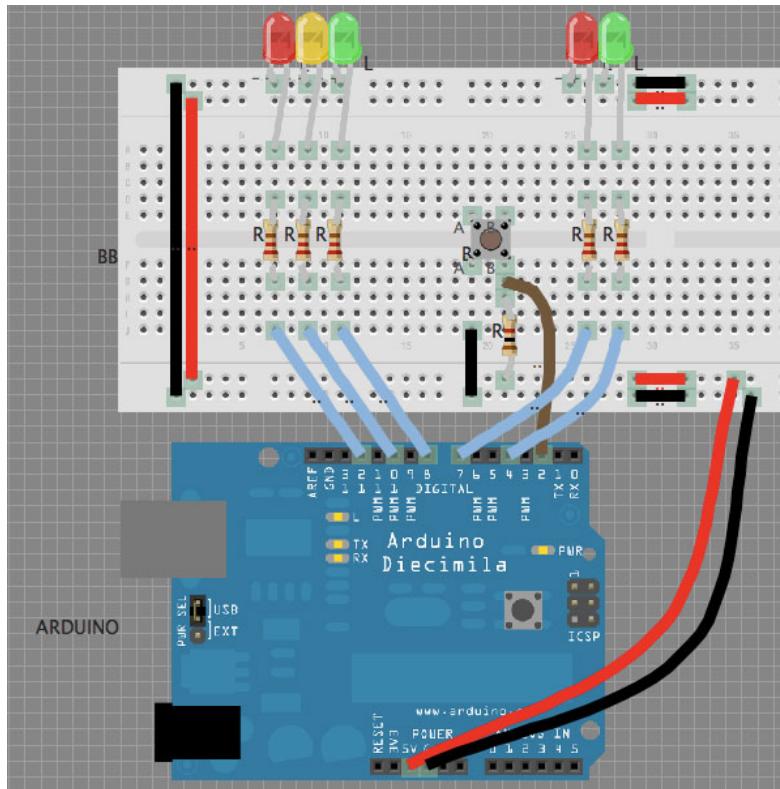
Prototyping Tools



- Protopboard / Breadboard
 - Vertical and Horizontal connectors
 - Plug wires and connect components
 - Avoid soldering
 - Speed up sketching
 - Avoid complex planning of electrical circuits

Planning and documentation

Fritzing (www.fritzing.org)



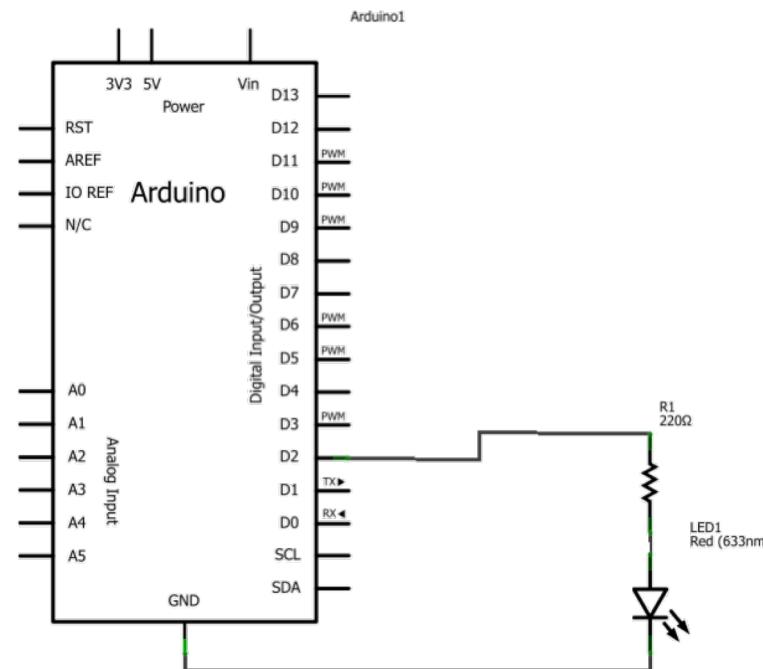
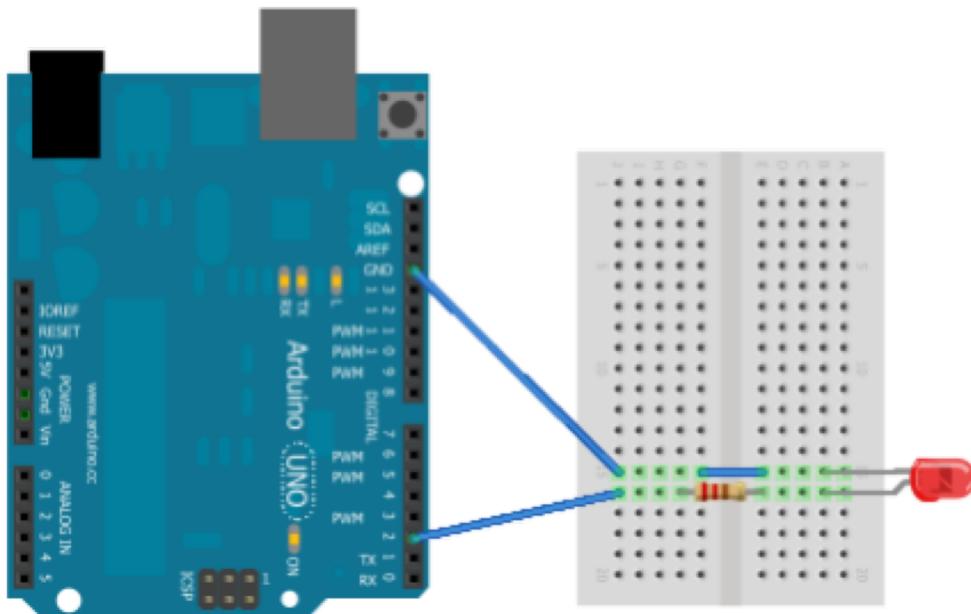
Where to get parts for your project?



Hands on!

- Goal: Let LED blink
- Steps to go:
 - See through the kits
 - Create an electronic circuit
 - Connect electronic circuit with Arduino board
 - Write code to let LED blink 5 times/second
 - Upload code to the Arduino board
- Play around:
 - Change parameters, add more LEDs
 - Be inspired for more complex projects
 - Have fun!

Wiring the circuit



Long leg of the Led is the positive pole.

Use a 220 Ohm resistor to limit the current (Why? We'll learn it in a later session)

- Use this basic structure

```
// the setup function runs once when you press reset or power the board
```

```
void setup() {  
    // insert initialization here  
}
```

```
// the loop function runs over and over again forever
```

```
void loop() {  
    // insert program logic here  
}
```

- Methods to get the job done

- **pinMode(pin, mode);**
 - pin: the pin number
 - mode: INPUT, OUTPUT, or INPUT_PULLUP
- **digitalWrite(pin, value);**
 - pin: the pin number
 - value: HIGH or LOW
- **delay(time);**
 - Time: time in milliseconds

- One possible solution

```
const int pinNumber = 2;  
const int waitingTime = 100; // in ms  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
    // initialize digital pin 2 as an output.  
    pinMode(pinNumber, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
    digitalWrite(pinNumber, HIGH);    // turn the LED on by making the voltage HIGH  
    delay(waitingTime);            // wait for 100 ms  
    digitalWrite(pinNumber, LOW);    // turn the LED off by making the voltage LOW  
    delay(waitingTime);            // wait for 100 ms  
}
```