

Arduino cursus

Eelco Dijkstra

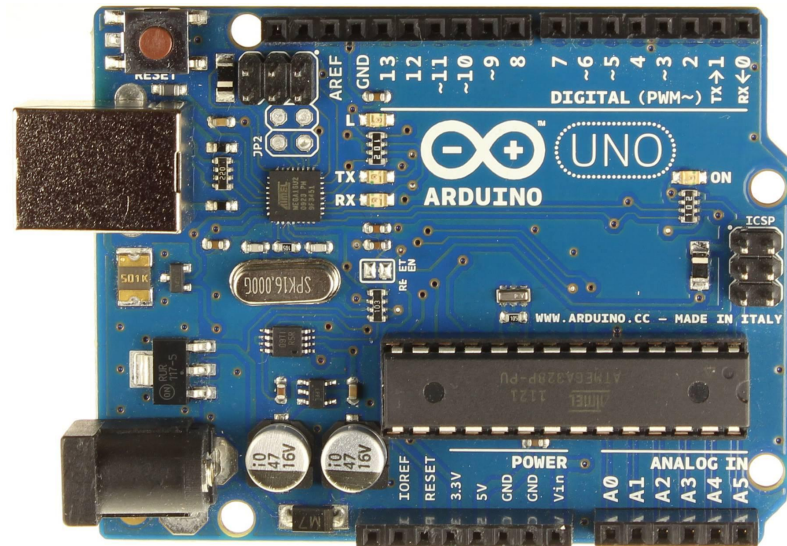
Steunpunt Informatica - Its Academy

24 september 2015

Wat is “Arduino”?

Een Open Source hardware/software prototyping eco-systeem.

- Open Source
- Hardware/software
 - “sketch”
- Prototyping
- Eco-systeem



Arduino – de basis

- microcontroller (kleine/complete computer)
- sensoren (van druktoets tot luchtdruk)
- actuatoren (van LED tot motor)
- communicatie met host; met internet

- host computer
 - software-ontwikkelomgeving (Arduino IDE)
 - schema-tekenomgeving (Fritzing)

Wat kun je er mee?

- Control 3D printer
- Maze Solver Robot (& many other robots)
- Christmas Tree Shield with Addressable LEDs and "Jingle Bells"
- Poo and Pee detector, with a Diaper Shield, for use on newborn newbies.
- Thermostat
- Arduino + vineyard = Vinduino. Soil moisture measurement
- Whole house climate control: Control HVAC and ceiling fans
- Arduino [theremin using ultrasonic sensor or capacitive plates](#)
- [Brewmigo - An All Grain Home Brewery using an Arduino Uno](#)
- [Automate your Chicken Coop](http://arduino-coop.blogspot.com/) <http://arduino-coop.blogspot.com/>

Zie bijv.: <http://playground.arduino.cc/Projects/Ideas>

Arduino

Combineren en verbinden van

- software
- computer-hardware
- internet, web
- elektronica
- mechanica, fysica, biologie
- dingen (Internet of Things – draadloos)
- mensen
 - interfaces: sensoren en actuatoren
 - *vormgeving*

Basiskennis

Digitaal

input

- button – toestand
- button - overgang

output

- niveau
- codering (in tijd)

Analoog

input

- ongeveer (LDR)
- precies (temperatuur)

output ("PWM")

- frequentie
- duty-cycle

Arduino: software ↔ hardware

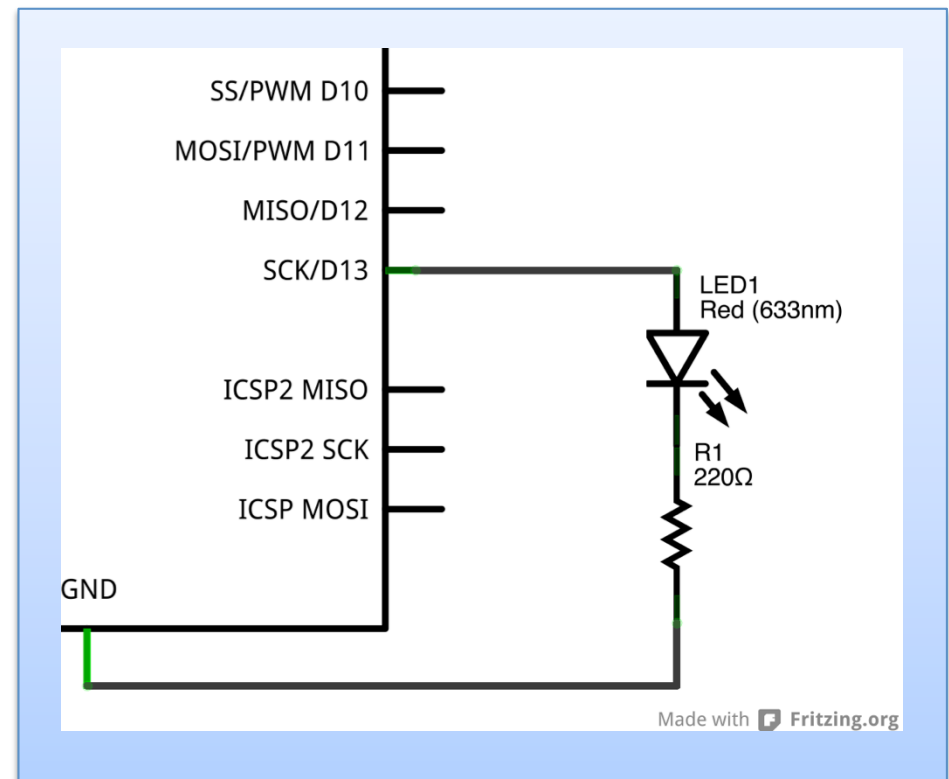
Software sketch

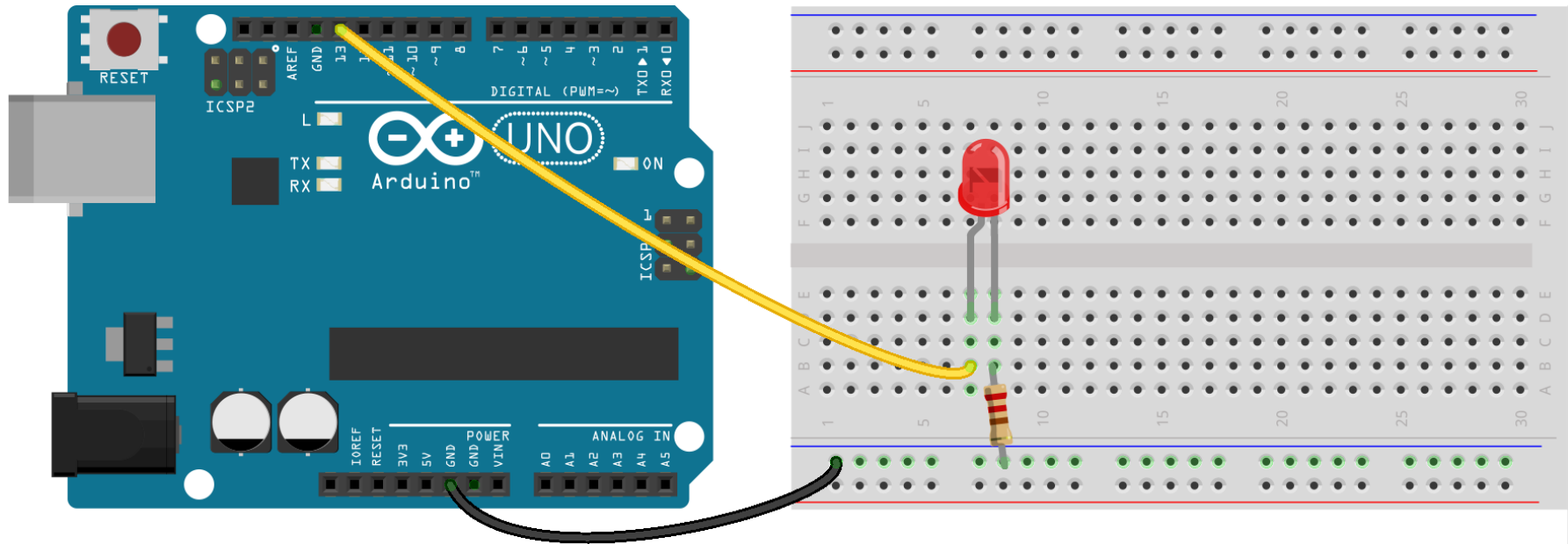
```
const int ledPin = 13; // 1

void setup() { // 2
  pinMode(ledPin, OUTPUT);
}

void loop() { // 3
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

Hardware schakeling





Ontwikkelen: kleine stappen

Testen van software

input:

vaste 0V/5V

button (overgang)

potmeter (analoog)

output:

gebruik van test-LED

Testen van hardware

input:

gebruik van test-LED

output:

0V/5V

button

potmeter

Software-deel van “sketch”

1. declaraties
(data, functies)

2. setup
initialisatie

3. loop
“oneindig”
herhaald

```
const int ledPin = 13; // 1

void setup() { // 2
  pinMode(ledPin, OUTPUT);
}

void loop() { // 3
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

Impliciet: “main”

impliciet:

“main” - met aanroepen
van setup en loop

```
void setup() {  
  ...  
}  
  
void loop() {  
  ...  
}  
  
void main() {  
  setup();  
  while (true) {  
    loop();  
  }  
}
```


Spanning & stroom

Hydraulisch

- * Druk
- * Stroom
- * Klep
- * Kraan
- * Smalle buis
- * Wijde buis
- * Terugslagklep

Elektrisch

- * Spanning
- * Stroom
- * Schakelaar
- * Regelbare
weerstand
- * Grote weerstand
- * Kleine weerstand
- * Diode

Pull-down

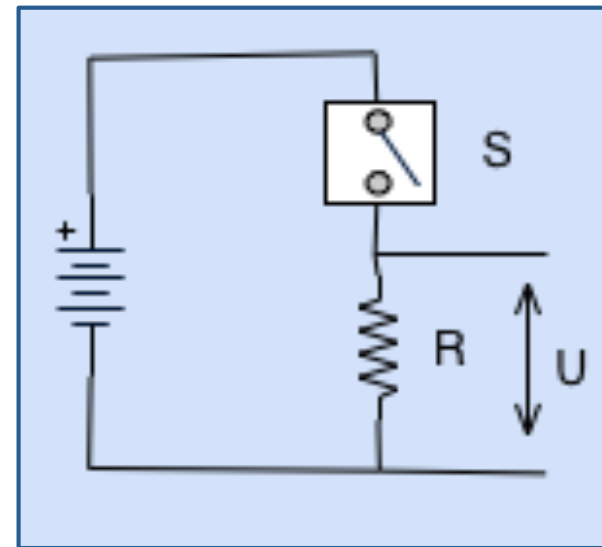
schakelaar => spanning (0V, 5V)

schakelaar open:

- $U = 0V$

schakelaar gesloten:

- $U = 5V$



Pull-up

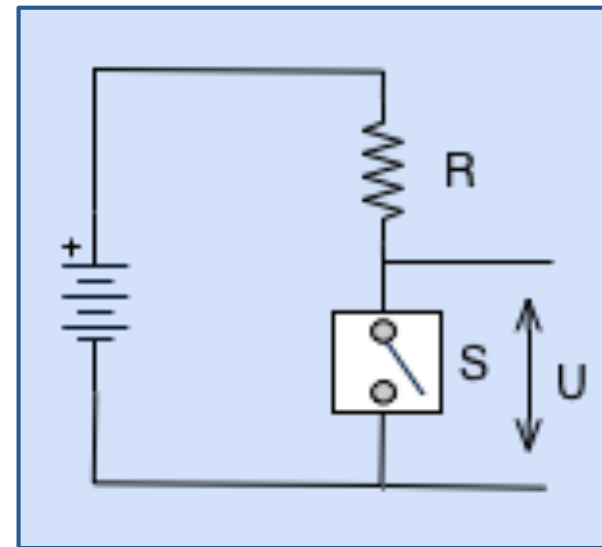
schakelaar => spanning (5V, 0V)

schakelaar open:

- $U = 5V$

schakelaar gesloten:

- $U = 0V$



Wet van Ohm

$$U = I \times R$$

- stroom evenredig met spanning
 - bij zelfde weerstand (buis, kraan):
 - grotere druk geeft grotere stroom
- stroom omgekeerd evenredig met weerstand
 - bij gelijke druk:
 - kraan verder open (=kleinere weerstand): grotere stroom

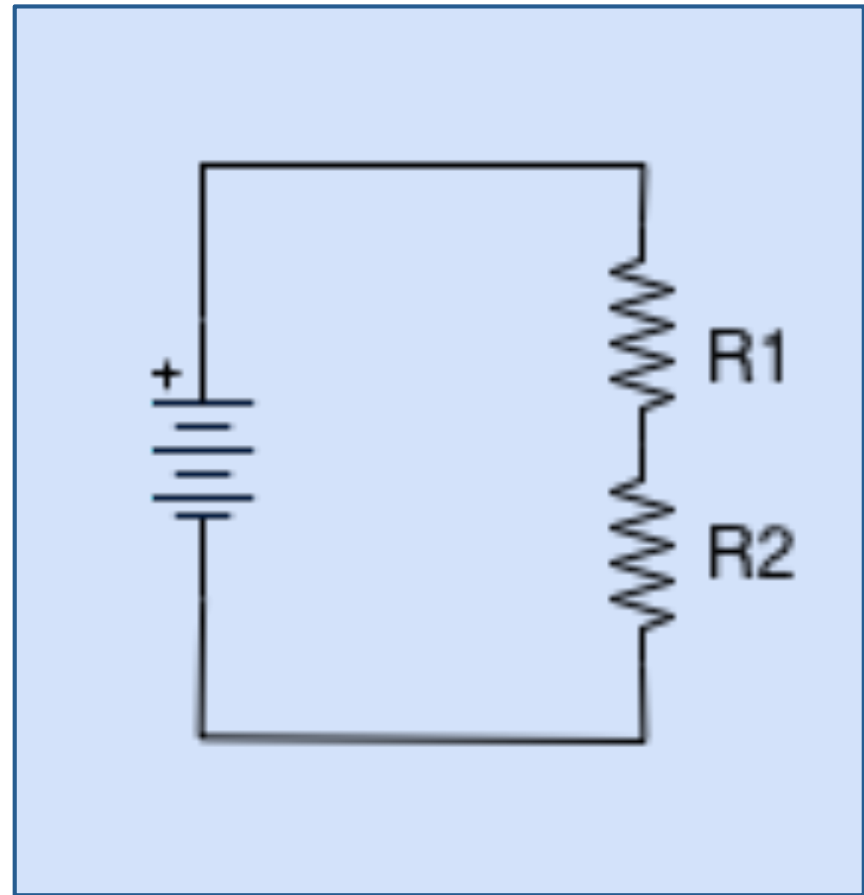
Spanningsdeler

$$I(R1) = I(R2) = I(\text{batt})$$

$$I = U/R \text{ (wet van Ohm)}$$

$$U(R1)/R1 = U(R2)/R2$$

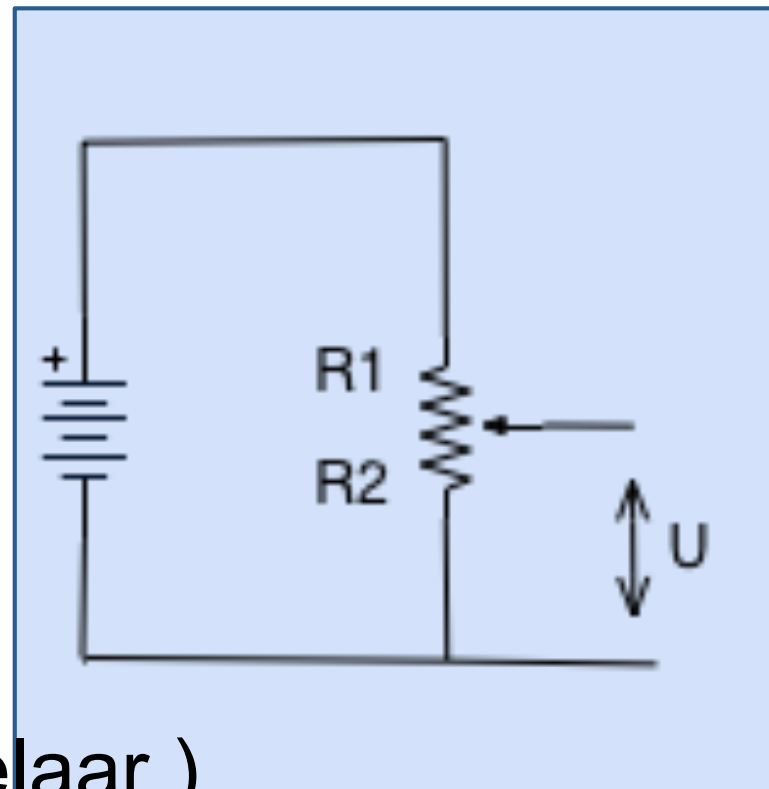
$$U(R1)/U(R2) = R1/R2$$



Potmeter -> variabele spanning

Potmeter: weerstand met verplaatsbare aftakking.

=> variabele spanning

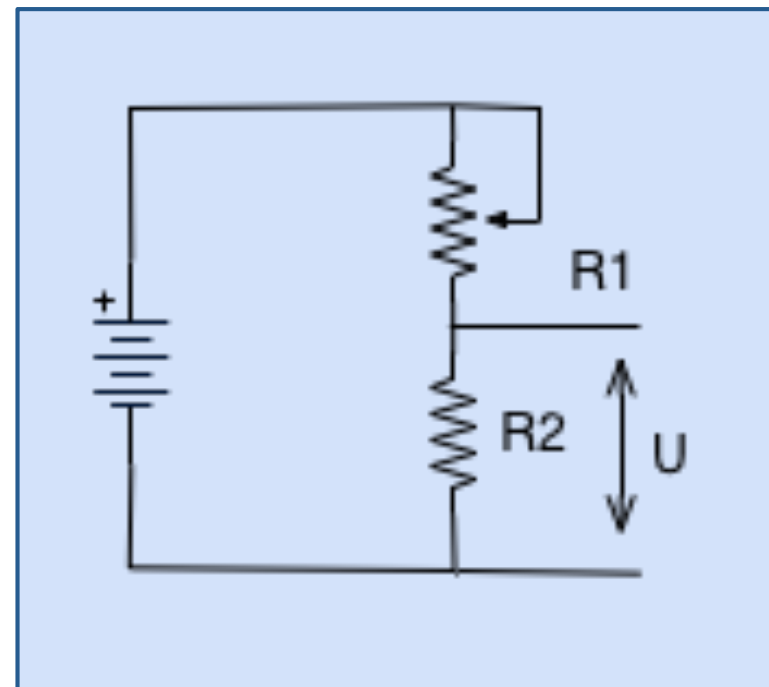


(Gebruik o.a.: volumeregelaar.)

Variabele weerstand -> var. spanning

omzetten van
variabele weerstand
in
variabele spanning

Veel sensoren vormen
variabele weerstand
LDR, NTC/PTC,
Flexsensor, ...



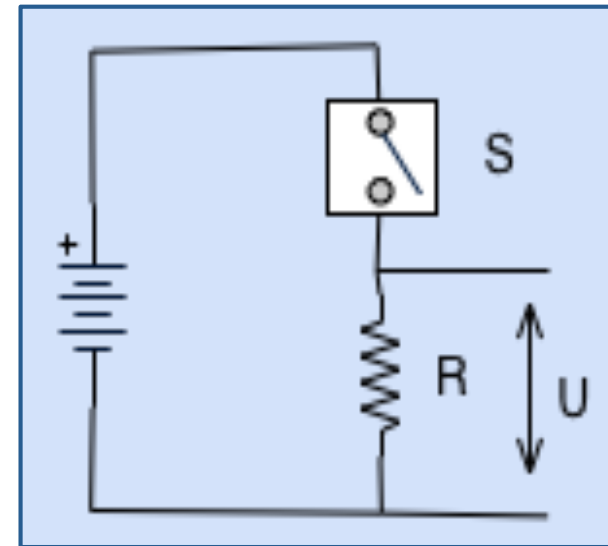
Pull-down

schakelaar => spanning (0V, 5V) => (LOW, HIGH)

```
const int buttonPin = 2;
const int ledPin = 13;

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  if (digitalRead(buttonPin) == HIGH)
  {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```



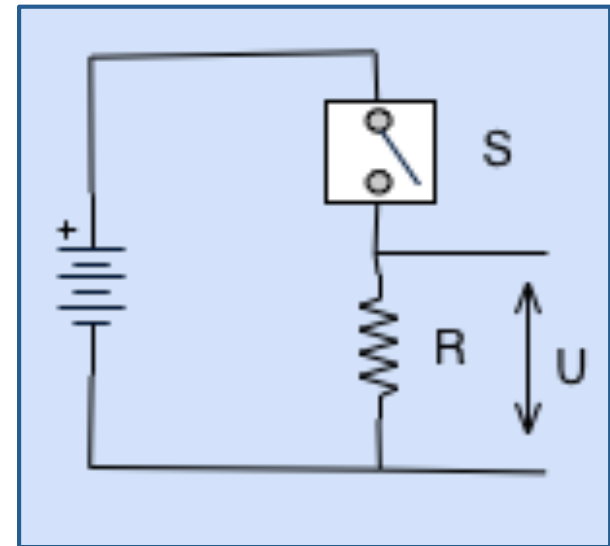
Pull-down

schakelaar => spanning (0V, 5V) => (LOW, HIGH)

```
const int buttonPin = 2;
const int ledPin = 13;

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  digitalWrite(ledPin,
    digitalRead(buttonPin));
}
```



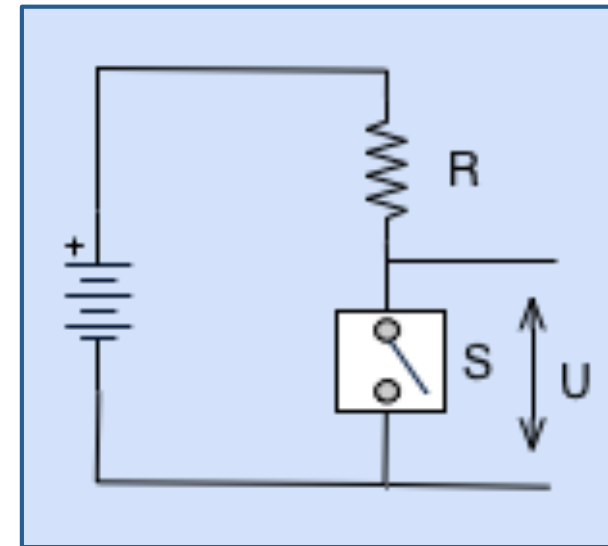
Pull-up (internal)

schakelaar => spanning (5V, 0V) => (HIGH, LOW)

```
const int buttonPin = 2;
const int ledPin = 13;

void setup() {
  pinMode(buttonPin, INPUT_PULLUP);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  digitalWrite(ledPin,
    HIGH -
  digitalRead(buttonPin));
}
```



Tijd (toestand, transitie)

Standaard “Blink”:

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

- Wat als je twee LEDs hebt, die onafhankelijk van elkaar knipperen?
- Wat als je een tweede LED met een druktoets wilt aan- en uitschakelen?

Simpele Button

Kopiëren van toestand van button -> LED

- triviaal systeem (eenvoudig in hardware)

```
void loop(){
  int button = digitalRead(buttonPin);
  digitalWrite(ledPin, buttonState);
}
```

Button: omschakelen

- Bijhouden van *toestand*:
 - prevButton - vorige waarde van button
 - ledState - huidige waarde van LED

```
int prevButton = LOW;
int ledState = LOW;
...
void loop() {
    int button = digitalRead(buttonState);
    if (prevButton == LOW && button == HIGH) { // L->H
transition
        ledState = HIGH - ledState; // reverse LED
        digitalWrite(ledState);
    }
    prevButton = button;
}
```

Timer: LED brandt enkele sec.

```
int ledTimer = infinite;
int prevButton = LOW;
...
void loop() {
  int button = digitalRead(buttonPin);
  if (prevButton == LOW && button == HIGH){
    digitalWrite(ledPin, HIGH);
    ledTimer = millis() + 5000;
  }
  if (millis() > ledTimer) {
    digitalWrite(ledPin, LOW);
    ledTimer = infinite;
  }
  prevButton = button;
}
```

Timers en andere events: “asynchroon”

- *vermijd wachten* (“delay”, of wachten op input)
 - Arduino is dan “doof” voor andere invoer
 - wachten kost energie (beter: uitzetten)
- gebruik timers (en andere input-events)
 - verschillende inputs/onderdelen onafhankelijk(er)
 - afhandelen van event: “zo kort mogelijk”
- vgl. event-handling in OS (GUI), Javascript

Enkele bronnen

Arduino.cc

- "de bron"
- documentatie, tutorials
- downloaden van software
- nieuws: blog
- voorbeelden van projecten
 - <http://playground.arduino.cc>
- verkoop van boards, en andere onderdelen
 - voor kleine aantallen: handiger om lokaal te kopen

Adafruit

<https://www.adafruit.com> (en)
<http://www.ladyada.net>

- shop (...heel veel...)
 - ARDX experimenter's kit
 - FLORA
- blog
- **tutorials**



Sparkfun

<https://www.sparkfun.com>

- products (hardware)
 - Sparkfun Inventor's kit
 - Makey Makey
 - ...
- blog
- ***tutorials***



Make:

<http://makezine.com>

- tijdschrift
- veel voorbeelden van projecten (o.a. met Arduino)
- shop



Hardware (NL-shops):

- iPrototype - <https://iprototype.nl>
- floris.cc - <http://floris.cc/>
- EOO – <http://eoo-bv.nl>
- Tinitronics – <http://tinytronics.nl>
- ... e.v.a.

Ontwikkelingen

Ontwikkelingen

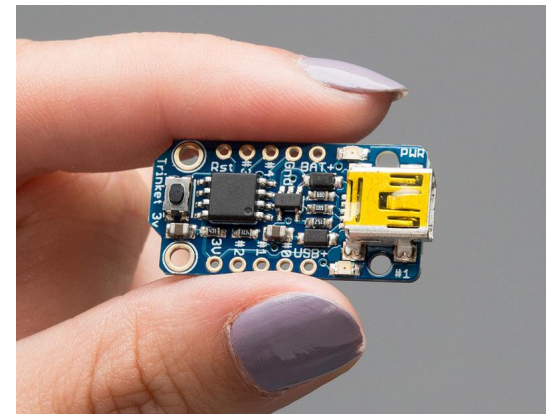
Steeds *groter*

- krachtig; Linux
- Hub in IoT?
- Arduino Due
- Arduino Tre



Steeds *kleiner*

- goedkoop; low power
- draadloos; IoT node
- M0 (ARM M0 core)
- Trinket



Ontwikkelingen

Steeds *groter*

- krachtig; Linux
- Hub in IoT
- Raspberry Pi



Steeds *kleiner*

- goedkoop; low power
- draadloos; IoT node
- ESP8266 (WiFi)
- BBC micro:bit (BT)



Arduino in het onderwijs

Arduino in het onderwijs

Arduino is heel geschikt voor het onderwijs – als hulpmiddel voor experimenten of als "onderwerp".

- Natuurkunde, NLT, Biologie, Informatica

Arduino is niet (speciaal) gemaakt voor het onderwijs

- je hebt soms aanpassingen nodig
- het is "echter" dan veel onderwijsmateriaal

Arduino in het onderwijs

- Natuurkunde/scheikunde/NLT:
 - Besturing/Logica
 - Aansturen van experimenten
 - Concepten: meten/nauwkeurigheid; “sampling”, analoog vs. digitaal;
 - ...
- Techniek/creatieve projecten
 - *integratie* van theorie & praktijk; *vakoverstijgend*

Arduino in het onderwijs?

- Informatica:
 - Hardware concepten (digitale signalen, logica, elektronica, vermogen, tijd, ...)
 - Hardware/software interfacing (digitaal, analoog/digitaal, tijd, voorstelling van getallen)
 - Embedded systems; real-time systems
 - “Internet of Things”
 - ...

Arduino in het onderwijs: hardware?

Verschillende manieren:

- Breadboard, losse componenten
 - iets grotere barrière in begin
 - minder robuust ("zorgvuldig werken")
 - goedkoop
 - universeel
- "voorgekookte" componenten (e.g. Grove)
 - eenvoudig begin
 - robuuste verbindingen
 - iets duurder
 - minder keuze in componenten

Arduino in het onderwijs: software?

- Arduino IDE: <http://arduino.cc>
- Arduino Studio: <http://arduino.org>
 - "de Italiaanse tak"
- in de "cloud": <https://codebender.cc>
- grafisch:
 - Scratch 4 Arduino: <http://s4a.cat>
 - Snap 4 Arduino: <http://s4a.cat/snap/>
-

Lesmateriaal

- in ontwikkeling
 - infvo.com/lab (CC-BY-SA licentie)
 - Delft/Amsterdam (regionale INF steunpunten)

Wensen, suggesties?