

Arduino-eksperiment	130101	Stikord	Installation, valg af board og port, init og loop, indbygget lysdiode
Version	2018-05-24 / HS	Niveau	Grundkursus, modul 1
			p. 1/5

Det lærer du:

- Installér Arduino-systemet på din PC
- Afprøv, om installationen virker
- Skriv dit allerførste Arduino-program
- Lær at gemme og hente programmer

Dette modul er atypisk: Det er lidt længere end de følgende moduler og der er temmelig meget instruktion og forklaring – og kun få udfordringer. Hold ud!

Hvis Arduino-systemet allerede er installeret på den benyttede PC, kan du springe direkte frem til afsnit 2.

## 1 – Installation

### Download Arduinos IDE

Arduino-systemet omfatter både hardware og software. Hardwaren – selve Arduinoen – kan programmeres via et USB-kabel, som sluttes til en PC.

Programmerne til Arduinoen skrives man i et PC-program, som kaldes et *IDE*, som står for "Integrated Development Environment" eller på dansk et "integreret udviklingsmiljø". Det lyder dyrt, men betyder bare, at vi her har "hele pakken", som er nødvendig for at

- 1) Skrive, redigere, gemme og hente programmer
- 2) Oversætte til maskinkode
- 3) Uploade til Arduino hardwaren

Hent Arduinos IDE på hjemmesiden <https://www.arduino.cc/en/Main/Software>

**ARDUINO 1.8.5**  
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

**Windows** Installer, for Windows XP and up  
Windows ZIP file for non admin install

**Windows app** Requires Win 8.1 or 10  
**Get**

**Mac OS X** 10.7 Lion or newer

**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM

Release Notes  
Source Code  
Checksums (sha512)

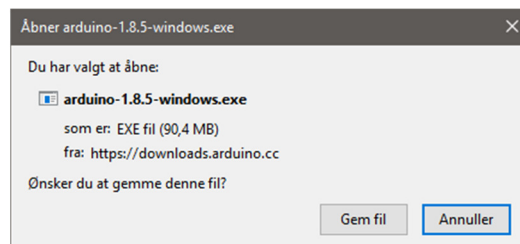
Dér ligger nyeste version – til flere forskellige styresystemer. Den øverste mulighed (her fremhævet med rødt) vælges, hvis du har en Windows-PC. (Dette antages i resten af beskrivelsen.)

På den næste side, du kommer til, kan du vælge om du vil donere penge til Arduino-projektet og derved være med til at sikre den fortsatte udvikling af systemet – eller om du bare vil hente programmet gratis (Just download).

JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

Windows vil nu spørge, hvad der skal ske med den fil, du er ved at downloade – vælg at gemme den et sted, hvor du kan finde den igen.



## Installation

Når installationsfilen er hentet, startes installationen ved at dobbeltklikke på filen.

Du bliver spurgt: "Vil du tillade, at denne App foretager ændringer på din enhed" – og det vil du jo egentlig gerne have den til, så: "Ja".

I de følgende dialogboks skal du blot acceptere standard-valgene.

Når installationen er afsluttet, kan Arduino IDE'et startes ved hjælp af ikonen på skrivebordet.



## 2 - Hardware

### Oversigt

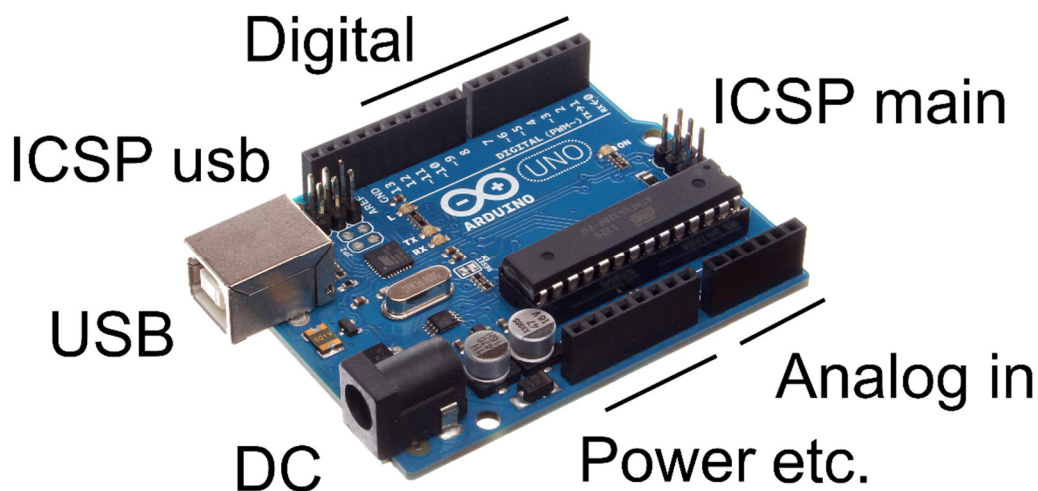
En Arduino har mange tilslutningsmuligheder, men de er nogenlunde overskueligt samlet i grupper.

I hele dette grundkursus kan du nøjes med én type spændingsforsyning: 5 V fra PC'en via USB-kablet. Men hvis du vil bruge Arduinoen separat, kan den strømforsynes gennem DC-stikket i stedet. Frederiksen har en handy adapter, så du kan tilslutte et almindeligt 9 V-batteri til DC-stikket, varenummer 663810.

En stribe digitale input/output-"ben" er samlet langs den ene side af boardet. Egentlig fremstår de som hunstik, som man kan sætte ledninger med hanstik i – men de er typisk direkte forbundet til de enkelte ben på microcontroller-chippen, så vi vil fortsat kalde dem *ben*.

Langs den modsatte side finder man en stribe analoge indgange og en række ben med forskellige spændinger – her får du typisk kun brug for 0 V og 5 V til at forsyne de opstillinger, der sluttes til Arduinoen.

Der sidder to grupper à 6 ben og stritter op to steder på boardet. Det er stik til ICSP – "In-Circuit System Programming" for henholdsvis hovedprocessoren og USB-forbindelsen. Vi vil ikke berøre brugen af disse forbindelser i dette kursus.



### Pas på din Arduino

En Arduino er blot "en printplade med lidt komponenter på". På undersiden er der masser af komponentben og lodde-ø'er, som kan kortsluttes, hvis du lægger Arduinoen ovenpå f.eks. en papirklips eller en elektronik-komponent.

- Sørg altid for, at bordet er ryddet omkring Arduinoen, så der ikke er noget, der kan trille ind under den.

Frederiksen sælger en bundplade, som Arduinoen kan monteres på – stabilt og med luft nedenunder. Den har også plads til et breadboard ved siden af Arduinoen – det kommer vi til at bruge i mange af de følgende moduler i kurset.

Bundpladen har varenummer 663820.

## 3 – Nulstilling af Arduino

### Et program, som ingenting laver

I Arduino-kredse kalder man af en eller anden grund et program for "en sketch". Det bruges f.eks. i menu-systemet i Arduino-IDE'et. I dette kursus bruger vi dog ikke denne betegnelse.

Når du starter på et ny Arduino-program, vil nedenstående skabelon altid være indsat i kode-vinduet:

```
void setup() {  
  // put your setup code here, to run once:  
  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
}
```

Her i starten vil vi ikke gå i dybden med hvert eneste ord og parentes – du skal bare være omhyggelig, når du bliver bedt om at indtaste noget nyt. Efterhånden vil du forstå alle detaljer!

De to linjer, som starter med "//", er *kommentarer* – tekst, som ikke betyder noget for programmet, men som forklarer for læseren, hvad der sker. (Vi vender tilbage til kommentarer i et senere modul.)

Det er tydeligt, at programskabelonen er delt op i to afdelinger:

- En blok betegnet "setup": Den programkode, du skriver her, bliver udført én gang før det øvrige program.
- En blok betegnet "loop": Koden her gentages igen og igen.

Hver programblok er omgivet af { krøllede parenteser }.

Planen er om lidt at lave et meget simpelt testprogram for at se, om opsætningen af Arduino IDE'et, kabelforbindelser osv. er i orden. Men det er meget tænkeligt, at din Arduino allerede har fået et lignende program installeret, så vi vil bruge ovenstående – helt tomme – program til at "nulstille" Arduinoen med.

Gem programmet under navnet "Tomt": Menuen *Fil / Gem som*. Indtast navnet "Tomt" og klik *Gem*. Det får to ting til at ske: Der oprettes en mappe med navnet "Tomt" og nede i mappen gemmes programmet under navnet "Tomt.ino". (Det er ikke sikkert, at din Windows viser endelsen ".ino", men du kan genkende Arduino-filerne på den grønne ikon.)

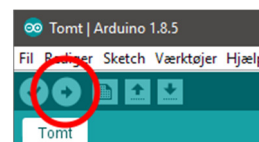
Arduino-programmer skal altid ligge i en separat mappe med samme navn som programmet.

### Upload af programmet

Nu skal programmet uploades til Arduinoen. Klik på knappen med pilen:

Det får tre ting til at ske – om alt går vel:

1. Programmet oversættes fra den læselige tekst til en form, som microcontrolleren i Arduinoen kan forstå. Man siger, at koden *kompileres* og den del af systemet, som udfører dette, kaldes en *compiler*. Programmet er nu omsat til *maskinkode*.
2. Programmet (maskinkoden) overføres via USB-forbindelsen til Arduinoen.
3. Arduinoen udfører programmet.



Undervejs kan du se, at der sker lidt på først på PC'en, dernæst på selve Arduinoen.

### Hvad der kan ses på PC'en

I den grønne bjælke neden under det lyse felt med koden vil der dukke tre statusmeldinger op efter hinanden:

- Compiling sketch...
- Uploader...
- Done uploading

I det sorte felt derunder vises mere detaljeret status. Hvis alt er OK, vises til sidst programmets pladsforbrug. Sjovt nok fylder selv det tomme program lidt i Arduinoens hukommelse.

Til slut vil bunden af programvinduet se således ud:

```
Done uploading.
Sketch uses 444 bytes (1%) of program storage space. Maximum is 30720 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.
```

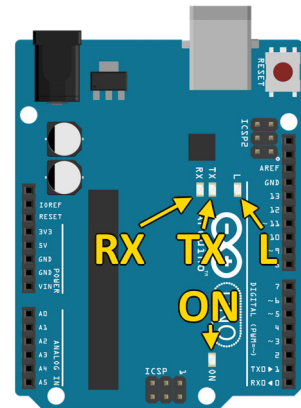
### Hvad der kan ses på Arduinoen

En Arduino Uno rev. 3 har fire små lysdioder – se tegningen:

- ON: Tændt, når der er spænding på Arduinoen
- RX: "Receive" – Lyser, når du sender information fra PC via USB-forbindelsen til Arduino.
- TX: "Transmit" – Lyser, når der sendes information fra Arduinoen til PC'en
- L: En ekstra lysdiode, som programmet kan styre – vi ser hvordan om et øjeblik. (Den blinker også løs under upload.)

Hvis alt er i orden, vil *RX*, *TX* (og *L*)-lysdioderne flimre, mens programmet uploades. Derefter vil *L* lyse konstant.

Nu er Arduinoen nulstillet, og du er klar til at skrive dit første "rigtige" program.



### Hvis det ikke virker

- Er der fejl i programmet? (Det er ret usandsynligt med dette program! – Men prøv for sjov skyld at skrive et "x" et eller andet sted, og klik Upload igen. Læg mærke til, hvordan fejlmeldingen ser ud. Slet x'et igen ☺)
- Er USB-kablet tilsluttet både Arduino og PC? (Prøv at se, hvordan systemet reagerer, hvis du prøver at uploade, når kablet ikke er forbundet.)
- Er det rigtige *Board* valgt? (Menuen *Værktøj / Board*. Normalt "Arduino/Genuino Uno".)
- Er den rigtige *Port* valgt? (Træk USB-stikket ud. Start helt forfra i menuen med *Værktøj / Port*. Kik på listen i menuen. Sæt stikket i igen. Start helt forfra i menuen med *Værktøj / Port*. Kik på listen i menuen. Den nye port på listen er den rigtige.)

## 4 – Det første Arduino-program

### Indtast og afprøv programmet

IDE'et gemmer automatisk programmet under det eksisterende navn, hver gang det oversættes (hvis du har lavet ændringer siden sidst). Det er derfor en god vane altid at gemme programmet under et *nyt* navn, inden du laver større ændringer. Gem under navnet "Blink-LED".

Indtast programmet, præcis som vist i nedenstående listning – de to kommentarlinjer er her slettet. Det er vigtigt at bruge store og små bogstaver, parenteser og semikolonier som vist.

```
void setup() {
  pinMode(13,OUTPUT);
}

void loop() {
  digitalWrite(13,HIGH);
  delay(250);
  digitalWrite(13,LOW);
  delay(750);
}
```

Programmet gennemgås om lidt, Men først vil vi prøve, om det virker! Tryk igen på Upload-knappen. Når programmet er overført til Arduinoen, vil den hvert sekund udsende et kort glimt fra lysdioden *L*.

Hvis ikke det virker, må du i gang med checklisten ovenfor igen.

## Forstå programmet

Arduinoen har en række nummererede ben, som kan bruges til at kommunikere med omverdenen. Før du kan bruge et af benene, skal Arduinoen have at vide, om det skal være en indgang eller en udgang. Arduino Uno har lysdioden L tilsluttet ben 13, som derfor skal være en udgang. Det sker ved hjælp af linjen

```
pinMode(13,OUTPUT);
```

`pinMode` er en af de indbyggede *funktioner* i Arduino-systemet. Funktionen skal kaldes med to *parametre*: ben-nummeret og en angivelse af funktionen. `OUTPUT` er en *konstant*, som er defineret i systemet. (Den har værdien 1, men det er meget nemmere at forstå, hvad der sker, når man bruger en konstant med et fornuftigt navn.)

Når først benet er defineret som en udgang, skal den blive ved med at være det – derfor er denne instruktion placeret i `setup`-blokken.

Tilsvarende er `digitalWrite` også en foruddefineret funktion med to parametre. Første parameter er ben-nummeret, anden parameter den ønskede tilstand. Konstanterne `HIGH` og `LOW` anvendes til at sætte udgangen *logisk høj* (ca. 5 V på en Arduino Uno) hhv. *logisk lav* (ca. 0 V). Lysdioden lyser, når udgangen er høj.

Og endelig er `delay` en funktion med én parameter. Funktionen laver ingenting i den tid, som angives af parameteren. Der regnes i millisekunder, så 250 svarer til et kvart sekund. Når tiden er gået, fortsætter programmet til næste instruktion.

Tilsammen bevirker linjerne i `loop`-blokken, at lysdioden er tændt et kvart sekund og slukket trekvart sekund – hvilket gentages i det uendelige.

## Udfordring 1

Gem programmet under et nyt navn: "Blink2gange".

Skriv programmet om, så der kommer to korte blink og en længere pause for hvert sekund.

Hent det oprindelige program ind igen. Læg mærke til at de to programmer – i hvert sit vindue – fint kan være tilgængeligt samtidigt. Prøv at uploade de to programmer på skift.

## 5 – Læselighed af programmer

Du vil gøre dig selv en stor tjeneste ved at stille programmet pænt op med linjeskift og indrykning – ligesom i eksemplet ovenfor. Det gælder specielt, hvis koden vokser ud over 20-30 linjer.

Compileren, som skal oversætte programteksten til maskinkode, er fløjtende ligeglad med udseendet! Den ser kun på, at hver instruktion er korrekt og afsluttes med et semikolon. Programmet kunne sagtens skrives således:

```
void setup(){pinMode(13,1);}void loop(  
) {digitalWrite(13,1);delay(250);digitalWrite(13  
'  
0);delay(750  
);}
```

– men mon ikke de fleste kan blive enige om, at den oprindelige version er en smule nemmere at læse og forstå?

## 6 – Notation

I resten af dette kursus vil der stå f.eks. "`pinMode()`", når der i teksten kort refereres til "funktionen `pinMode`". Parenteserne skal minde om, at der er tale om en funktion og ikke f.eks. en konstant.

I programlistninger vil parenteser naturligvis skrives helt ud inkl. parametre som i "`pinMode(13,OUTPUT)`".

## 7 – Hjælp!

Vælg menupunktet *Hjælp / Reference* for en engelsk manual over Arduinos programmeringssprog og indbyggede funktioner. Det er et godt sted at kigge, når du vil vide lidt mere, end vi kan dække i dette grundkursus.

Som alle opslagsværker kan referencemanualen i starten virke uoverskuelig – efterhånden lærer du at finde rundt.