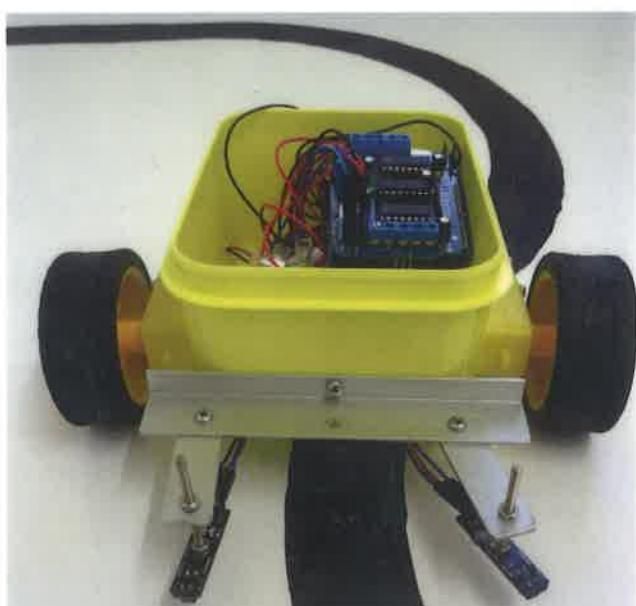


Hans Jørn Schmidt

Min Arduino

Kom godt i gang med Arduino



Annevista

Software og hardware

Arduino består af både software og hardware

Arduino IDE (Integrated Development Environment) er den software, du skal installere på din computer. Programmet bruges til at skrive, teste og uploadere programmer til Arduino UNO boardet. Arduino IDE er gratis og findes både til Windows, Mac OS X og Linux.

Arduino boardet er det stykke hardware, du arbejder på, når du bygger dine projekter. I bogen bruges det mest udbredte Arduino board, som hedder Arduino UNO.

Downloade og installere Arduino IDE

Inden du kan arbejde med hardwaren, skal du have downloadet Arduino IDE og installeret programmet og de tilhørende drivere til din computer.

Instruktioner og skærmbilleder beskriver installationen af Arduino software og Arduino UNO drivere til Windows 7 eller Windows 10.

Åbn siden <http://www.arduino.cc/> og klik på **Download**.

På denne webside skal du klikke på Windows Installer.

Windows Installer

Inden du kan komme videre, skal du tage stilling til, om du vil støtte udviklingen af Arduino. Hvis ikke klikker du på **"Just Download"**.

JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD



Klik herefter på **"Gem som"**. Brug den lille pil til højre for "Gem" til at finde "Gem som". Gem filen et sted, hvor du kan finde den igen, evt. på Skrivebordet.

Når filen **"arduino-1.6.7-windows"** er hentet, skal du klikke på den for at begynde installationen af Arduino IDE 1.6.7 (marts 2016) på din computer.

Svar **"Ja"** til at du tillader, at programmet foretager ændringer på computeren.

Home Buy Download Products Learning Forum Support Blog LOG IN SIGN UP
ENGLISH

Download the Arduino Software



ARDUINO 1.6.7

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
Windows ZIP file for non admin install
Mac OS X 10.7 Lion or newer
Linux 32 bits
Linux 64 bits
[Release Notes](#)
[Source Code](#)
[Checksums](#)

De vigtigste krav var, at Arduino boardet skulle være billigt. Det måtte ikke koste mere end, hvad en studerende skulle betale for en pizza på et spisested, og skulle være let at komme i gang med. Arduino skulle også være perfekt til både begyndere og til mere avancerede brugere.

David Cuartielles designede selve hardwaren (Arduino boardet) og en elev af Massimo Banzi, David Mellis, konstruerede softwaren (Arduino IDE) til boardet. Softwaren er skabt som et meget simpelt programmeringssprog, som gør det nemt for ikke-programmører - såsom kunstnere eller designere - og endda børn - at forstå det og udvikle projekter med det. Arduino programmeringssproget er baseret og implementeret "oven på" flere andre open source redskaber, hvoraf det mest kendte er "Processing" - en anden platform for designere og kunstnere.

Massimo kontaktede en ingeniør Gianluca Martino, som også arbejdede på instituttet, i forbindelse med at få lavet nogle få hundrede Arduino boards. Amerikaneren Tom Igoe var det femte medlem af teamet bag Arduino, og virkede som rådgiver.

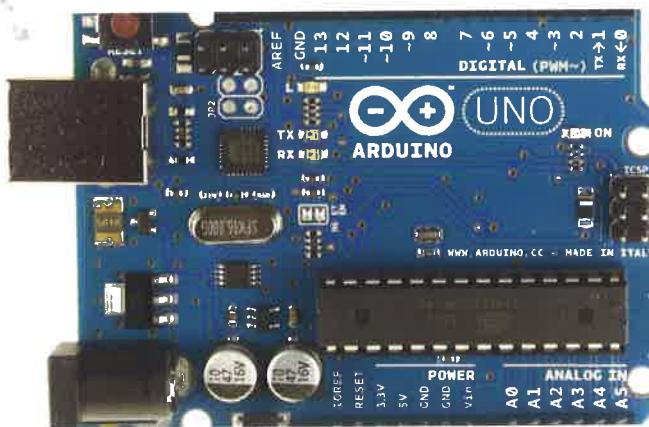
Arduino boardet blev et kæmpehit blandt de studerende. Ved hjælp af Arduino kunne de nu gøre ting, som var utænkelige før. Fx hvordan de kunne få sensorer til at kontrollere motorer, styre lys og robotter. Det

er afgørende at forstå, at Arduino er både open hardware og open software under en Creative Commons licens. Det betyder, at enhver kan gå ind på deres hjemmeside og uden omkostninger downloade alle tegninger, instruktioner, programmer og nødvendigt materiale for at bygge hardware selv, ændre det og videredistribuere det, som man har lyst til.

Grundlaget for Arduino revolutionen

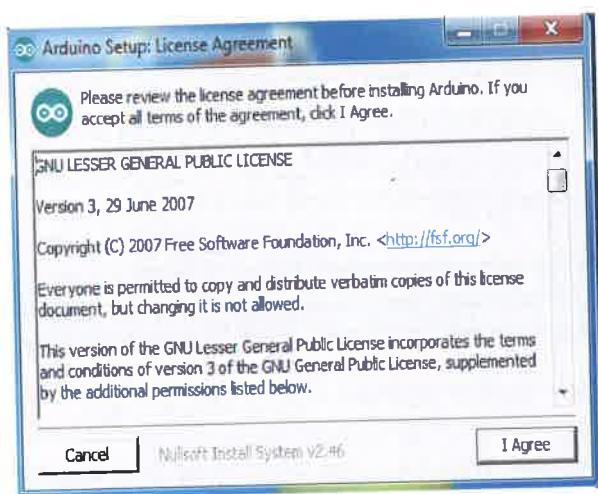
Da det er et smart og virkningsfuldt redskab, og da det er gratis, har tusindvis af mennesker downloadet det, skabt varianter eller bare implementeret det i deres projekter. Det har også haft en stor betydning for udbredelsen af Arduino, at softwaren kan bruges på alle computere med kendte styresystemer (Windows, Mac OS X eller Linux).

Vigtigst af alt så giver Arduino dig en grundlæggende viden, så du kan få dit indre geni frem og få brændstof til din kreativitet. Med Arduino kan du nemt omsætte en ide til virkelighed. Arduino handler om at få ideer til kreative projekter, der kan løse virkelige problemer. De mange projekter i bogen hjælper dig med at kontrollere den fysiske verden ved hjælp af Arduino.

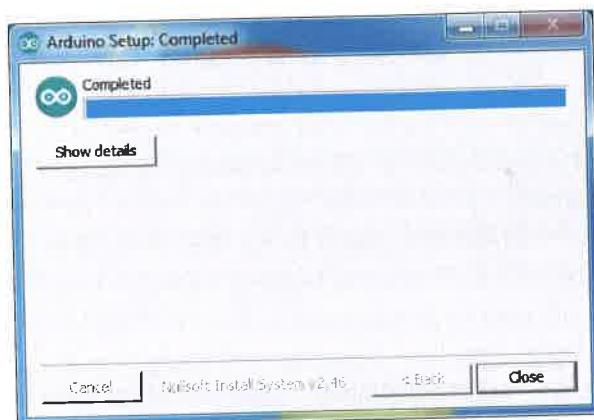


Arduino UNO board.

Software og hardware



Klik på "I Agree".

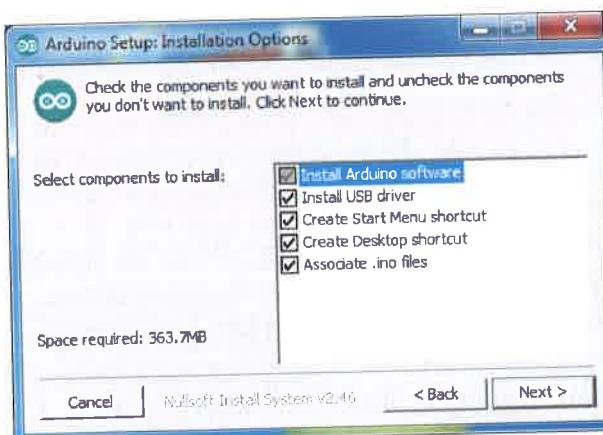


Klik på "Close" for at fuldføre installationen af programmet.

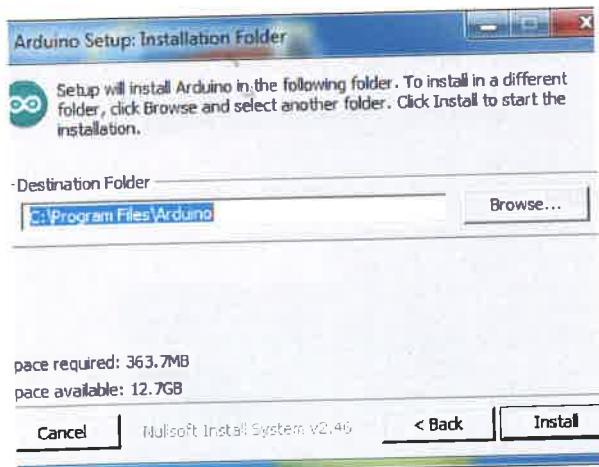
Dobbeltklik på ikonet for Arduino for at starte programmet Arduino IDE.



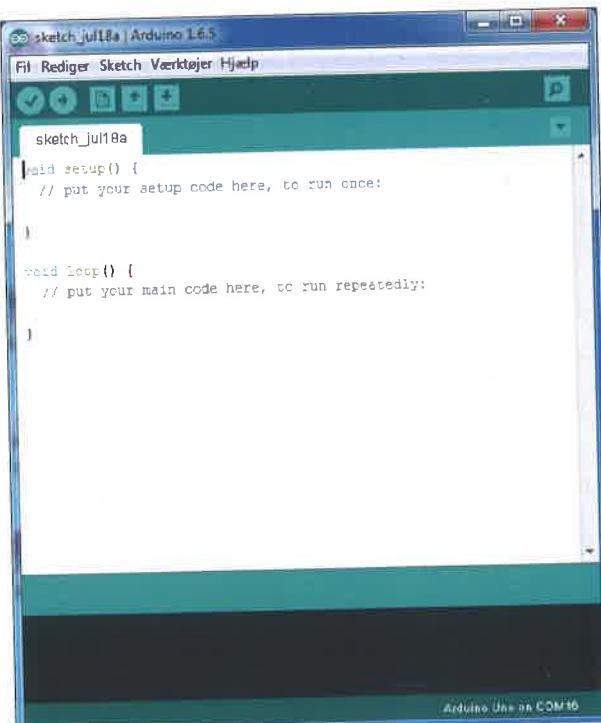
Så er Arduino IDE klar.



Klik på "Next".



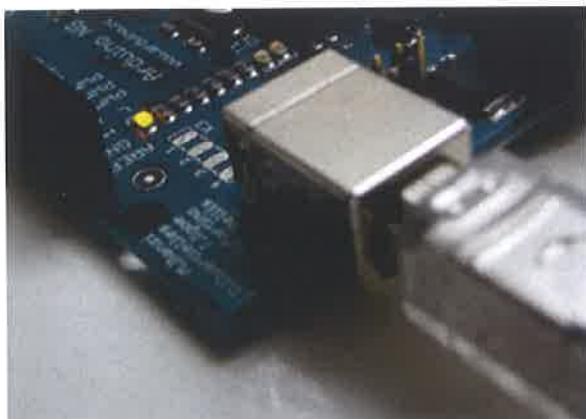
Klik på "Install".



Arduino IDE.

Upload dit første program

Du skal tilslutte dit Arduino UNO board til computeren ved hjælp af et USB kabel (type A-B). Programmer til Arduino kaldes også for sketches (skitser).



Arduino UNO tilsluttet et USB kabel.

Nu kommer det øjeblik, hvor du skal afprøve, om software og hardware virker efter hensigten.

Klik på ikonet for Arduino IDE og åbn blink programmet. Det gøres således:

Fil - Eksempler - 01.Basics - Blink



```
/*  
Blink  
Turns on an LED on for one second, then off  
for one second, repeatedly.  
Most Arduinos have an on-board LED you  
can control. On the Uno and
```

Leonardo, it is attached to digital pin 13. If you're unsure what pin the on-board LED is connected to on your Arduino model, check the documentation at <http://arduino.cc>. This example code is in the public domain.
*/

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

```
// the setup function runs once when you  
press reset or power the board  
void setup() {  
  // initialize digital pin 13 as an output.  
  pinMode(13, OUTPUT);  
}
```

```
// the loop function runs over and over again  
forever  
void loop() {  
  digitalWrite(13, HIGH); // turn the LED on  
  // (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(13, LOW); // turn the LED off  
  // by making the voltage LOW  
  delay(1000); // wait for a second  
}
```

Sådan ser blink programmet ud i tekstfeltet. Det ser svært ud, men fortvivl ikke. Du skal ikke bekymre dig alt for meget om de specielle kommandoer/funktioner i programmet. Målet er at vise dig, hvor nemt det er at få Arduino UNO boardet til at gøre noget. Alle de nævnte kommandoer/funktioner i blink programmet, vil blive gennemgået i forbindelse med Projekt 1B. Inden du uploader programmet til Arduino UNO boardet, er der nogle indstillinger, der

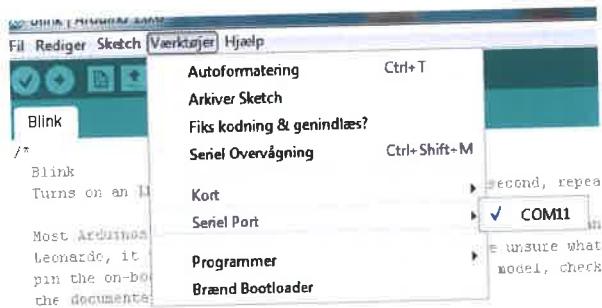
Upload dit første program

skal være i orden.

Gå ind under **Værktøjer - Board** og vælg

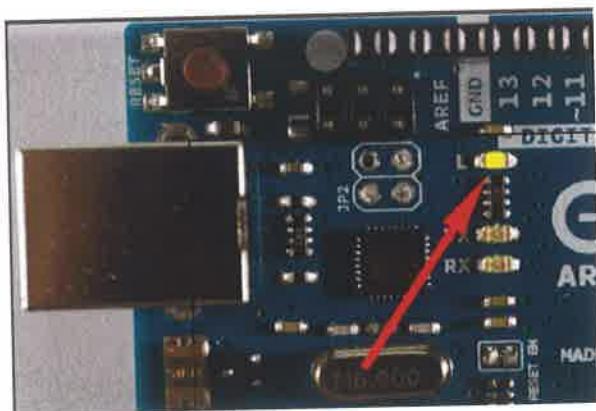
Arduino UNO.

Gå ind under **Værktøjer - Port** og vælg en af de foreslæde. I dette tilfælde kunne COM11 vælges.



Nu skal du prøve at uploadere dette program til Arduino UNO boardet.

Klik på  i værktøjslinjen for at uploadere blink programmet til Arduino UNO boardet. Efter få sekunder er programmet uploadet, og du skulle gerne kunne se, at den lille lysdiode, hvor der står L på Arduino UNO boardet, begynder at blinke én gang i sekundet. Hvis den gør det, så tillykke med det. Du har fået din første Arduino opgave til at virke.



Pilen viser placeringen af den indbyggede lysdiode.

Hvis det ikke virker

Eksempel på fejlmeddelelser fra Arduino IDE.



```
global variables use 9 bytes (0%) of dynamic memory, leaving 2,039 bytes for local variables. Maximum is 2,048 bytes.  
Serial port open failed: can't open device '\COM3'. Get help online if this error occurs.
```

Her optræder der en fejl fordi COM3 i dette tilfælde ikke er valgt. Så det er vigtigt at få valgt den rigtige port.



Denne fejl optræder, hvis du uploader et program uden at Arduino UNO boardet er tilsluttet.

Hvad har du gjort indtil nu?

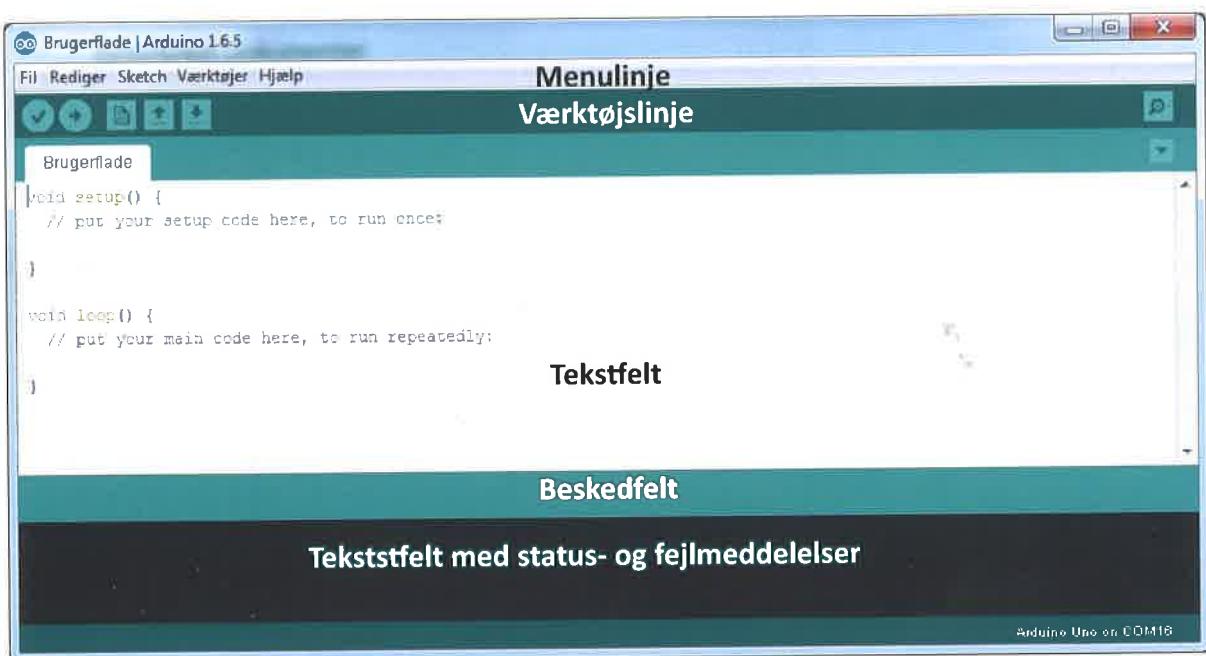
Indtil nu har du:

- Tilsluttet et Arduino UNO board til din computer
- Åbnet softwaren Arduino IDE
- Valgt et board og en COM port
- Åbnet blink programmet fra biblioteket med eksempler og uploadet det til Arduino UNO boardet

Inden du går videre med projekterne, skal du se lidt nærmere på Arduino IDE og Arduino UNO boardet m.m.

Arduino IDE

Programmets brugerflade



Menulinjen

Menulinjen med overskrifterne (**Fil – Rediger** osv.) finder du også i andre programmer, du sikkert er kendt med. Alle punkterne i menulinjen i Arduino IDE indeholder drop-down menuer, der er relevante for programmet. På en computer med Mac OSX er menulinjen i toppen af skærbilledet. I Windows og Linux er menulinjen øverst i det aktive Arduino IDE vindue.

Værktøjslinjen

Værktøjslinjen består af nogle knapper, der er nødvendige, når du skriver programmer (sketches) til Arduino UNO boardet. Med disse knapper, der også er tilgængelig på menulinjen, kan du udføre følgende funktioner:

Verifier

Kontrollerer at din kode giver mening for Arduino IDE softwaren. Når der trykkes på knappen kan man læse "Compiling sketch" i beskedfeltet. Denne proces kan sammenlignes med en stave- og grammatikkontrol i et tekstbehandlingsprogram. Vær dog opmærksom på, at selv om din sketch kontrolleres til ikke at have nogen umiddelbar fejl, er der ingen garanti for at dit program fungerer korrekt.

Upload

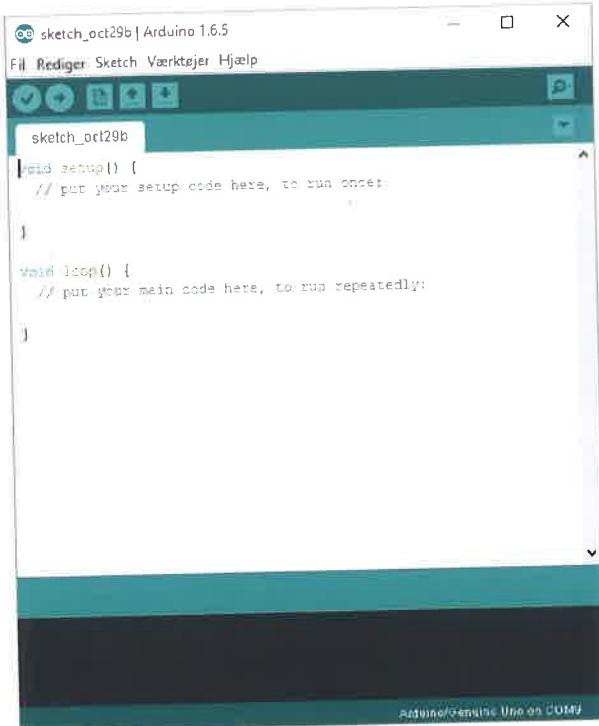
Sender dit program til et tilsluttet Arduino UNO board. Programmet kompileres automatisk før du uploader det. Det betyder at den kode, du har skrevet i tekstfelteret, bliver omsat til en ny kode, som mikroprocessoren

på dit Arduino UNO board kan forstå. Hvis der er fejl i din kode, bliver du gjort opmærksomt på det. Det kan være, at du har glemt et semikolon eller en parentes i de tilfælde, hvor du selv har skrevet programmet i tekstdfeltet.

Du kan ikke upload et program, hvis der er fejl i koden.

Ny

Opretter en ny sketch.



The screenshot shows the Arduino IDE interface with a new sketch titled "sketch_oct29b". The code in the editor is as follows:

```
sketch_oct29b | Arduino 1.6.5
FIL Rediger Sketch Værktøjer Hjælp
sketch_oct29b
void setup() {
  // put your setup code here, to run once:
}

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

At the bottom of the screen, it says "Arduino/Datalogic Uno en COM1".

Åbn

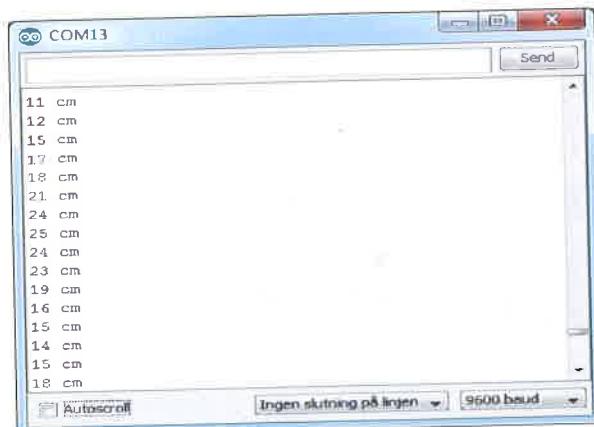
Her kan du åbne et vindue med et program, som du tidligere har gemt på computeren. Det er også samme sted, du skal åbne, hvis du vil have adgang til de mange andre eksempler på programmer, som følger med, når du installerer Arduino IDE.

Gem

Gemmer det aktuelle program.

Seriell Overvågning

Her bliver der åbnet et nyt vindue, hvor du både kan sende beskeder til Arduino UNO boardet eller læse den information, som dit Arduino UNO board sender tilbage til computeren. Det kunne fx være målinger fra en temperatursensor eller en afstandssensor.



Data fra "Seriell Overvågning" fra en afstandssensor.

Tekstfelt

Tekstfeltet er det område, du skriver dit program. Det er næsten som at skrive i et almindeligt tekstbehandlingsprogram, men har et par specielle funktioner. Nogle ord i teksten bliver farvekodet, hvis de anerkendes af Arduino IDE softwaren. Du har også mulighed for at autoformatere teksten, så den er lettere at læse.

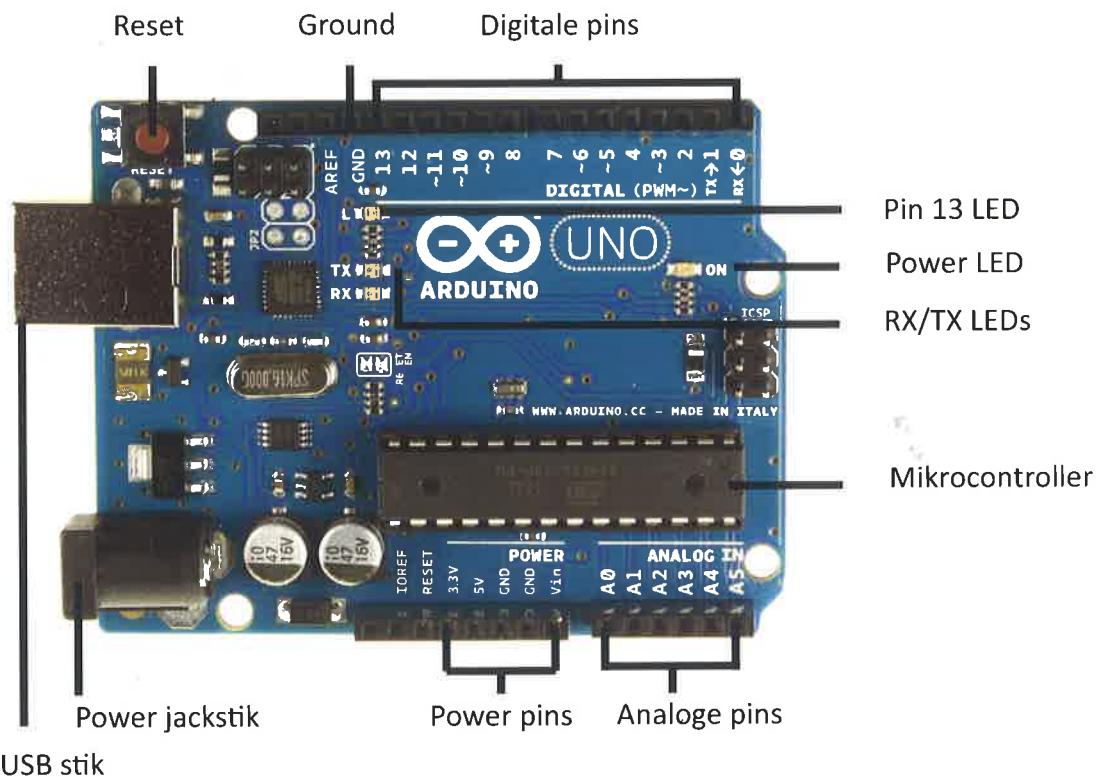
Beskedsfelt

I beskedsfeltet får du forskellige oplysninger. Bl.a. om der er fejl i dit program.

Tekstfelt med status- og fejlmeldelser

Her bliver du bl.a. gjort opmærksom på, hvad der kan være galt med dit program, hvis der er fejl i det.

Arduino UNO boardet



Arduino UNO boardet har følgende ind- og udgange:

Digitale INPUT/OUTPUT pins (porte)

Boardet har 14 digitale INPUT/OUTPUT pins (I/O pins), som kan programmeres til at være enten en indgang eller en udgang. Eller sagt på en anden måde: De digitale pins kan sende eller modtage digitale signaler. Dette indebærer at de har to tilstande: OFF eller ON – LOW eller HIGH. I elektriske termer betyder det en værdi på 0 volt eller +5 volt, men ingen værdier der imellem.

Analoge indgange

Boardet har 6 analoge indgange. Disse 6 pins kan modtage en analog værdi i intervallet

melleom 0 og +5 volt. Det kunne fx være 0,1 V eller 3,9 V.

Analoge udgange

Arduino UNO boardet har ingen analoge udgange. Men hvis du ser på de 14 digitale pins på den modsatte side af boardet, vil du se, at seks af disse digitale pins (pin 3, pin 5, pin 6, pin 9, pin 10, og pin 11) er mærket med et (~) – tilde. PWM er en forkortelse af **Pulse Width Modulation**. En PWM pin har den egenskab, at den kan simulere en analog udgang. PWM pins skal senere bruges til fx at ændre lysstyrken på en lysdiode (LED) eller variere hastigheden på en DC motor.

Arduino UNO boardet

USB stik

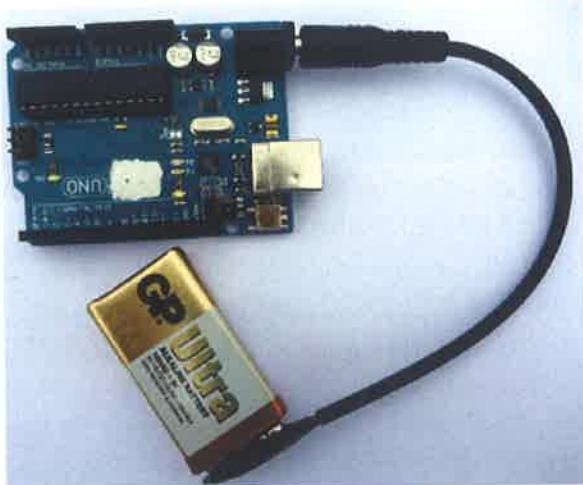
På Arduino UNO boardet er der et USB stik, som bruges, når der skal overføres data til og fra boardet. Desuden leverer USB stikket strøm til boardet fra computeren.



USB kabel tilsluttet et Arduino UNO board.

Power jackstikket

I mange tilfælde er det upraktisk, at have boardet permanent tilsluttet computeren. Så derfor kan stikket bruges til at tilslutte en ekstern spændingskilde (batteri eller batteriadapter). Det anbefales at spændingen til Arduino UNO boardet er 7-12V.



Arduino UNO boardet tilsluttet et batteri på 9V med et jackstik.

Reset

Arduino UNO boardet er udstyret med en

Reset knap, som kan genstarte programmet eller stoppe programmet, når den holdes nede et stykke tid.

Power pins

Når Arduino UNO boardet er tilsluttet en computer, kan det bruges som en strømforsyning. Som + - pol kan du enten anvende pin 5V eller pin 3,3 V. Som - - pol kan du anvende en af de tre steder, hvor der står **GND** (ground). De tre pins mærket **GND** er alle indbyrdes forbundet. Du vælger selv hvilken af dem, der er mest praktisk at bruge i forhold til et projekt.

I bogen bruger vi ikke betegnelsen - - pol, men kun GND eller 0 V.

Vin betyder Voltage in (volt ind) og er en pin, du kan bruge i forbindelse en ekstern strømforsyning.

LED'er

På Arduino UNO boardet er der placeret fire meget små lysdioder (LED'er) mærket **L**, **RX**, **TX** og **ON**. En LED er en komponent der producerer lys, når der løber en elektrisk strøm gennem den.

De fire lysdioder bruges alle til at angive en aktivitet på boardet:

Når **ON** lyser grønt, betyder det, at din Arduino er tændt.

RX og **TX** lyser, når der modtages eller sendes data.

L er en LED, der er forbundet med den digitale pin 13. Det var den lysdiode, du brugte, da du testede blink programmet.

Når du arbejder med dit Arduino UNO board er det vigtigt, at du ikke kommer til at lægge det på ting, som kan lede den elektriske strøm. Du kan risikere at kortslutte et kredsløb på Arduino UNO boardet og hermed ødelægge det.

Shields

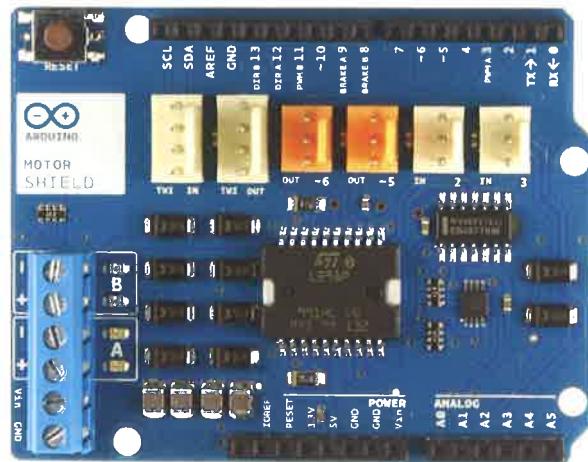
Typer af shields

Til Arduino UNO boardet findes der nogle hardware udbygninger, de såkaldte shields (skjolde). Disse shields bruges til at udvide funktionaliteten af Arduino UNO boardet, der bl.a. giver det mulighed for at kontrollere en gruppe motorer, blive koblet til Wi-Fi eller læse et SD kort.

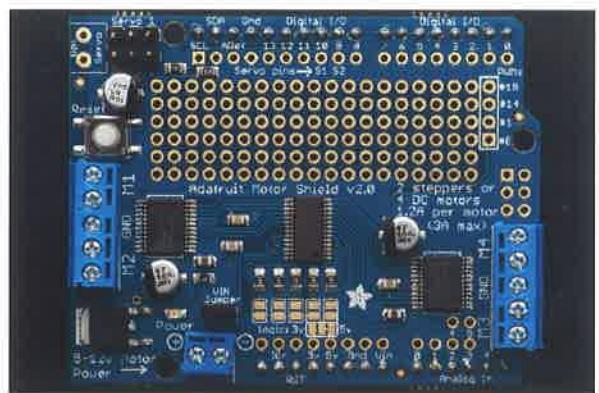
Stifterne på undersiden af et shield passer ned i plastikbøsningerne (Female Headers) på Arduino UNO boardet. Mange shields er selv udstyret med plastikbøsninger på oversiden, så de kan stables på andre shields. På denne måde kan du fx styre en 2WD eller 4WD robot via Wi-Fi.

Motor Shield

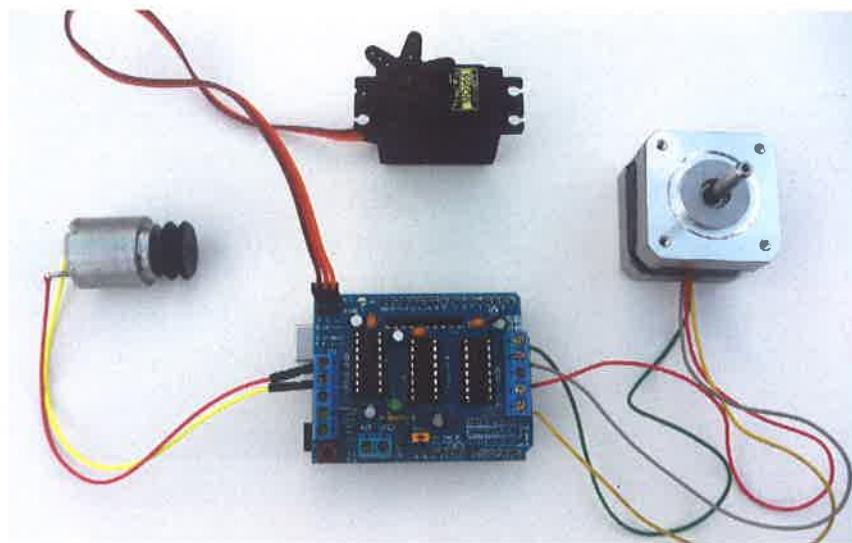
Nedenstående billede viser et meget populært motor shield (version 1.0) fra Adafruit Industries. Boardet kan tilsluttes to servomotorer, to steppermotorer eller op til fire DC motorer. Det anvendes bl.a. her i bogen.



Arduino Motor Shield.



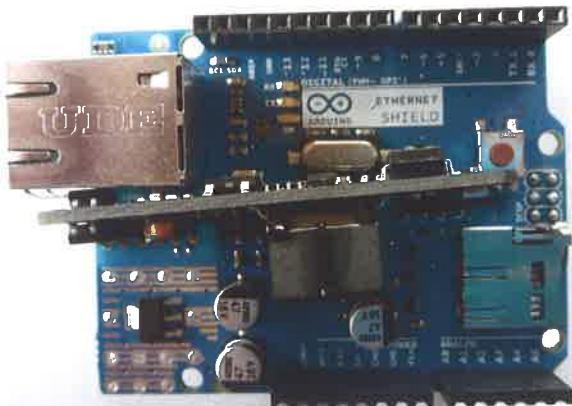
Adafruit Motor Shield (version 2.0).



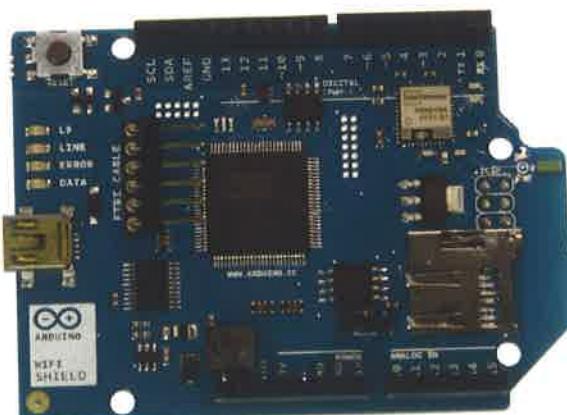
Adafruit Motor Shield (version 1.0).

Shields

Ethernet Shield



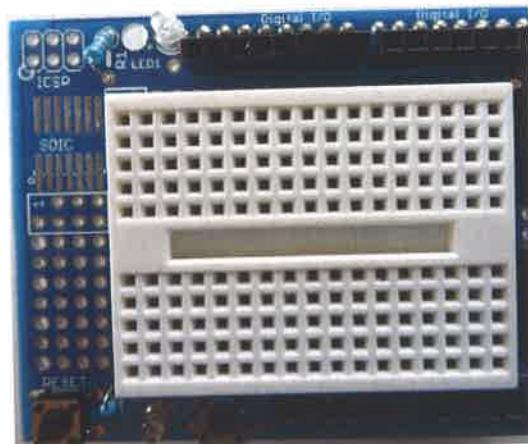
Wi-Fi Shield



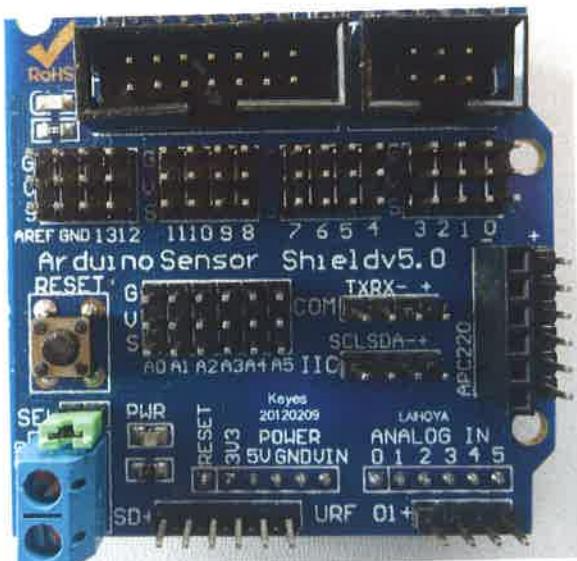
LCD & Keypad Shield



Prototyping Shield



Arduino Sensor Shield



I mange tilfælde kan det være en god idé, at anvende et Prototyping Shield og et mini breadboard for at lave en opstilling af et kredsløb mindre kompliceret.

På siden er der kun vist et lille udvalg af shields, som du kan anvende i forbindelse med et Arduino UNO board.

Sensorer og aktuatorer



Sensorer er elektroniske enheder, som tilsluttes Arduino UNO boardet, og som modtager data fra de fysiske omgivelser. Det kunne være en lyssensor, en afstandssensor, en temperatursensor eller noget så enkelt som en trykknap kontakt. Sensorer tilsluttes INPUT pins på Arduino UNO boardet. På baggrund af disse data kan et program få Arduino UNO boardet til at styre aktuatorer, som kan påvirke omgivelserne. Eksempler på aktuatorer kunne fx være lysdioder, motorer eller en piezo buzzer. Aktuatorer tilsluttes OUTPUT pins på Arduino UNO boardet. Eller sagt meget enkelt: Arduino UNO boardet lytter til sensorer og taler til aktuatorer.

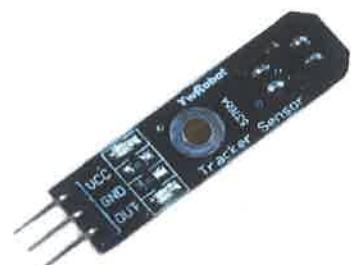
Eksempler på sensorer



Trykknap kontakt.



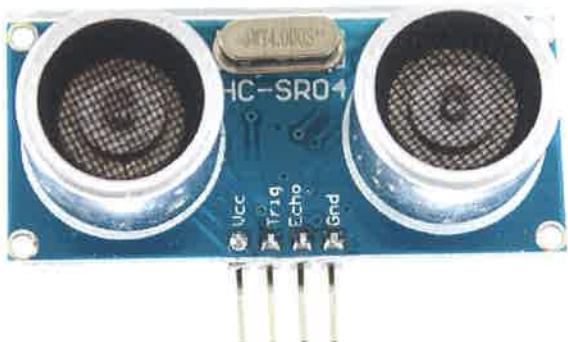
Temperatur sensor.



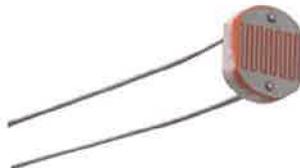
IR tracker sensor.



Anvendelse af IR tracker sensorer.

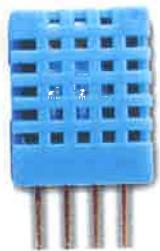


Afstandssensor.



LDR - lysfølsom resistor.

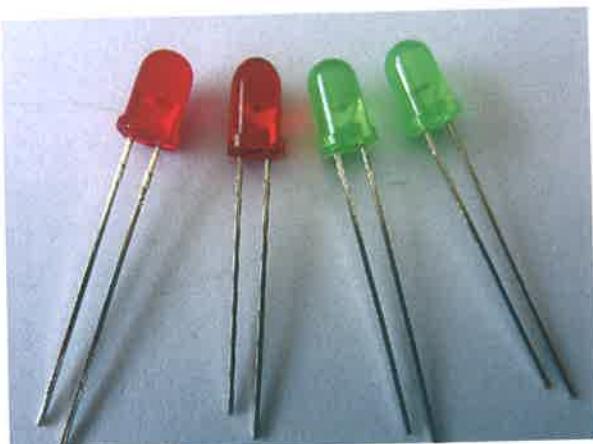
Sensorer og aktuatorer



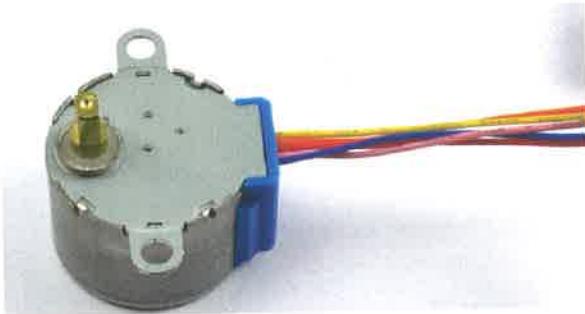
Fugtigheds- og temperatursensor.



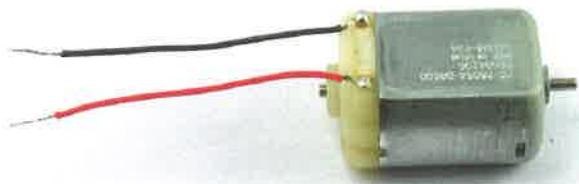
Servomotor.



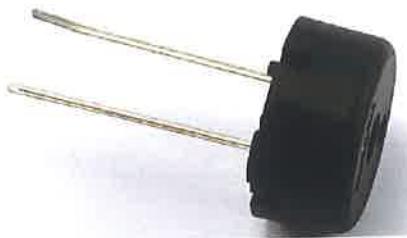
Lysdioder.



Stepper motor.



DC motor.



Piezo buzzer.



To gear DC motorer med hjul.

Dette er kun et lille udvalg af sensorer og aktuatorer, som findes til Arduino.

Breadboard og ledninger

Om breadboard og ledninger

Breadboard

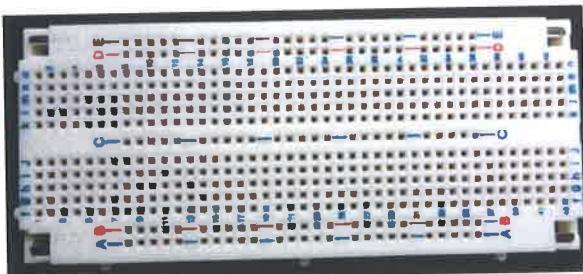
Et breadboard erstatter tidligere tiders "sømbræt", hvor man brugte loddekolber og loddetin til at fastgøre komponenter på sømhoveder. Breadboardet er dog langt smartere, idet du blot stikker komponenterne ned i hullerne på breadboardet, hvorefter der automatisk er forbindelse mellem dem. Der skal ikke loddes, så det er nemt at ændre tilslutninger og erstatte komponenter. Komponenterne vil ikke blive beskadiget, og kan bruges bagefter.

Breadboards findes i mange forskellige for-

mer og størrelser. De to følgende billeder viser et standart breadboard set fra forsiden og fra bagsiden. Dette breadboard er opbygget af 5 (A, B, C, D og E) langsgående forbindelser med hver 25 huller. I de to brede "striber" er der to gange 43 tværgående forbindelser med hver 5 huller.

På undersiden af breadboardet kan du se, hvordan kobberbanerne forbinder de forskellige grupper af huller vandret og lodret.

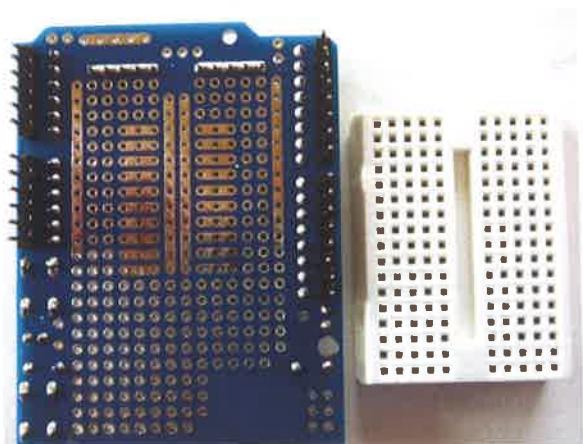
Det sker nogle gange, at du kan klare dig med et mindre breadboard. Nederst på siden kan du se et mini breadboard, som ofte bruges sammen med et Arduino Prototyping Shield.



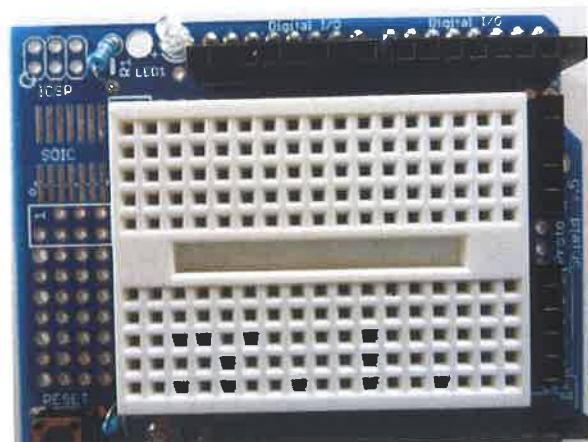
Overside på breadboard.



Underside på breadboard.



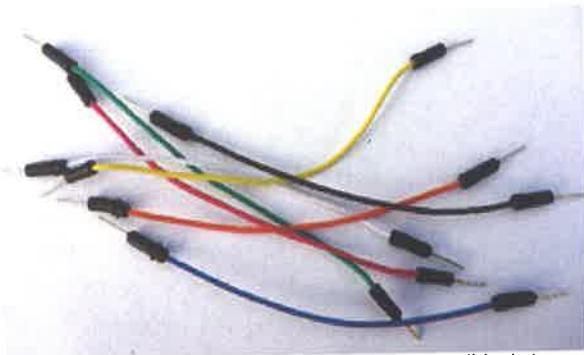
Prototyping Shield med et mini breadboard.



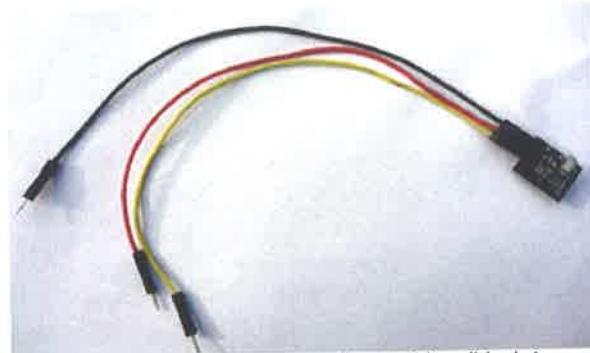
Prototyping Shield med et fastgjort mini breadboard.

Breadboard og ledninger

Ledninger



"Han til han" ledninger.



"Han til hun" ledninger.

Ovenstående billede viser ledninger af typen "han til han", som oftest bliver brugt til Arduino projekter. Det er en fordel at have ledningerne i forskellige farver for lettere at holde styr på forbindelserne til de forskellige komponenter på breadboardet og Arduino UNO boardet.

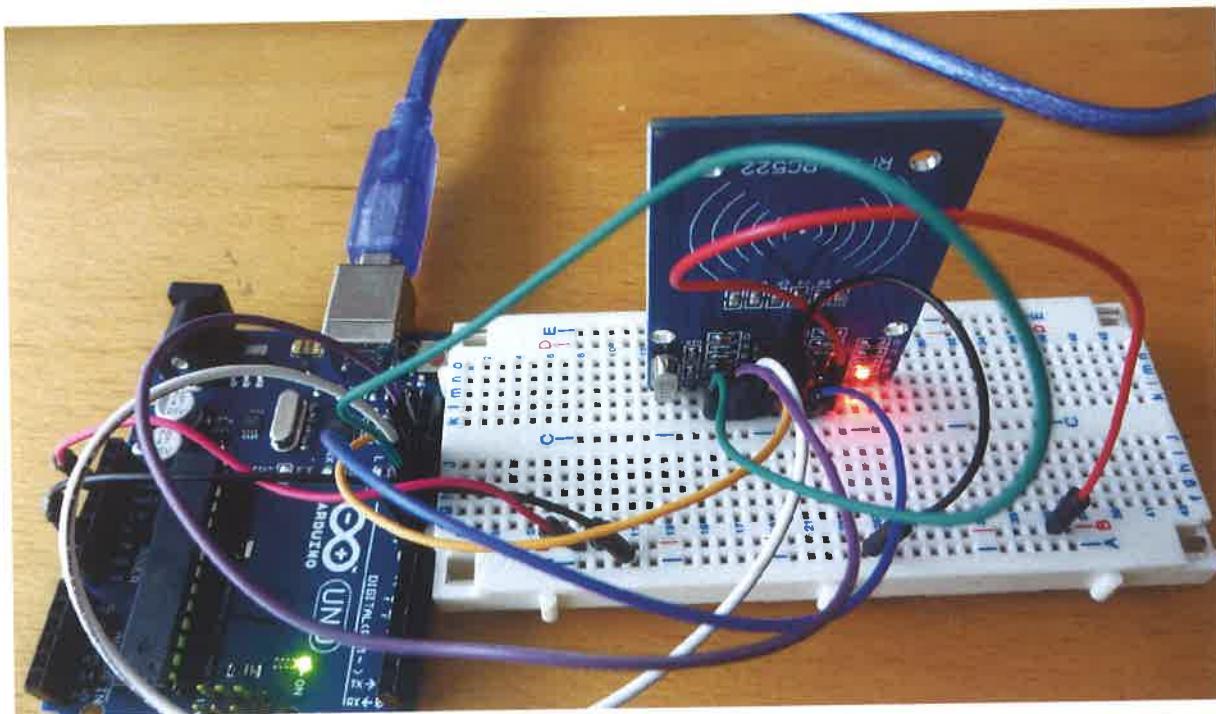
Det kan også anbefales at have nogle ledninger af typen "han til hun". De bruges til at forbinde nogle typer af komponenter og sensorer til Arduino UNO boardet.

På billedet øverst til højre er der brugt led-

ninger af typen "han til hun" i forbindelse med en IR receiver.

Om farven på ledninger

I bogen bruges de sorte ledninger til at forbinde enheder med GND og de røde ledninger til at forbinde enheder med +5V eller +3,3V. Hvilken farve ledninger du ellers vil bruge, bestemmer du selv. I opstillingen nederst på siden er der brugt syv forskellige farver.



I denne projekt er der brugt ledninger med syv forskellige farver.

Projekt 1A

Tænd en lysdiode med en tryknap kontakt

I Projekt 1A skal du blive fortrolig med at forbinde komponenter, breadboard og Arduino UNO boardet med ledninger efter en kredsløbsbeskrivelse.

I projektet skal du få en lysdiode til at lyse, når du trykker på tryknap kontakten. Arduino UNO boardet, som er forbundet til computeren med et USB kabel, bruges kun som strømforsyning.

Materialer

- Arduino UNO
- USB kabel
- Breadboard
- Ledninger
- En lysdiode (LED)
- En resistor (220 ohm)
- Tryknap kontakt (Off/On kontakt)

Om komponenterne i materialelisten:

Lysdiode

En lysdiode er en komponent, som konverterer elektrisk energi til lysenergi. Lysdioden er en polariseret komponent, hvilket betyder, at den kun kan lede strømmen i én retning. Det længste ben på en lysdiode kaldes anoden, og skal tilsluttes **+5V**. Det korte ben kaldes katoden og tilsluttes **GND** (ground). Da en lysdiode maksimalt må gennemløbes af en strømstyrke på ca. 20 mA, vil der i de fleste tilfælde være indsat en resistor i serie med denne. Lysdioder findes både i røde, grønne, gule, blå og hvide udgaver.

Resistor

En resistor bruges først og fremmest til at begrænse strømstyrken i elektriske kredsløb. Resistans måles i ohm (Ω). Værdien i ohm af en resistor kan aflæses ved hjælp af de farvede bånd på resistoren. For at få oversat rækken af farver til et tal, kan du enten bruge tabellen over farvekoder på den næste side eller installere en gratis app på fx din mobiltelefon. Du skal være opmærksom på, at resistorer både findes i udgaver med 4 og 5 farvede bånd. Det er en god idé at have et multimeter til sin rådighed, når du beskæftiger dig med elektronik, så kan du altid få tjekket om fx resistoren har den værdi, som du regner med.

Tryknap kontakt

Hvis en tryknap kontakt holdes nede i et elektrisk kredskøb, vil strømmen uhindret passere den. Hvis tryknap kontakten igen slippes, bliver den elektriske strøm afbrudt. Tryknap kontakter findes i udgaver med både 4 ben og 2 ben.

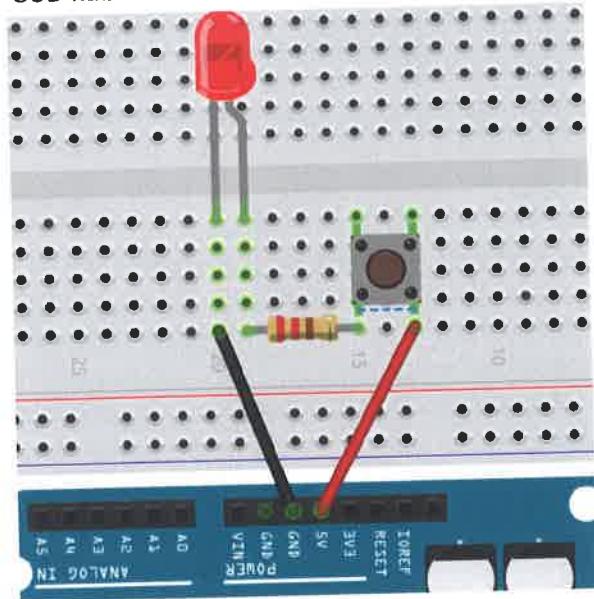


Multimeter.

Projekt 1A

Kredsløbsopgave 1A

Du skal bygge et kredsløb, der kan få en lysdiode til at lyse, når du trykker på trykknap kontakten. Dit Arduino UNO board skal være tilsluttet din tændte computer med et USB kabel.



Komponentplacering.

Komponentplacering på breadboard

Lav kredsløbet, som er vist på tegningen, og forbind Arduino UNO boardet til en tændt computer med et USB kabel. Lysdiodens længste ben i samme række som resistorens ene ben.

Det er en god idé altid at bruge sorte ledninger, som er forbundet til GND og røde ledninger, som er forbundet til +5V.

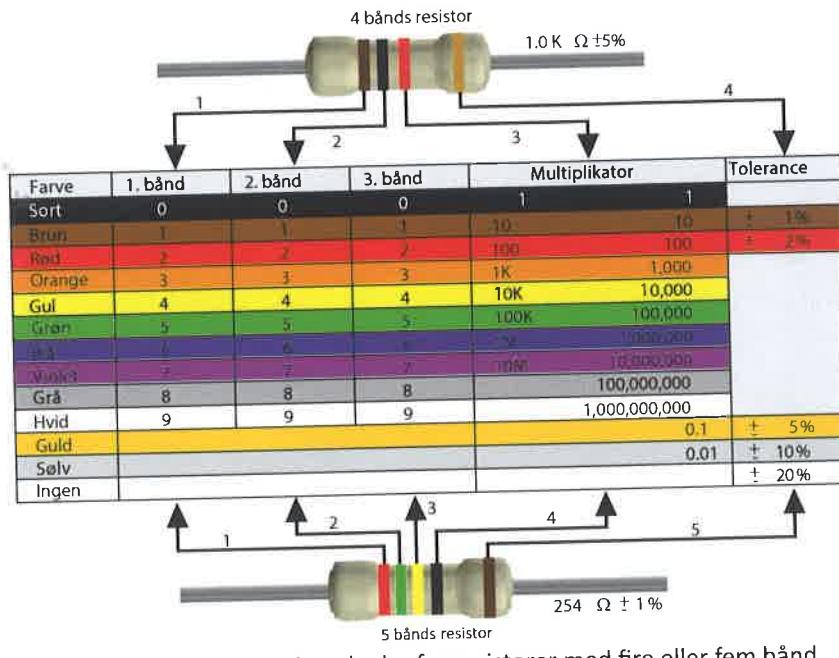
Hvad sker der hvis du trykker på trykknap kontakten og slipper den igen?

Nu skulle du gerne se, at lysdioden lyser og går ud igen, når du slipper kontakten. Hvis der ikke sker noget så gennemgå kredsløbet igen og se om komponenterne sidder i de rigtige huller på breadboardet.

Undersøg også om lysdioden vender rigtigt. Lysdiodens længste ben skal sidde i samme række på breadboardet som resistoren og lysdiodens korte ben skal være tilsluttet **GND**. Resistoren kan ikke vende forkert.

Hvis du anvender en tryknap kontakt med 4 ben, skal du også tjekke, om den vender rigtigt.

Farvekoder for resistorer



Farvekoder for resistorer med fire eller fem bånd.

Projekt 1B

Blink program med en lysdiode

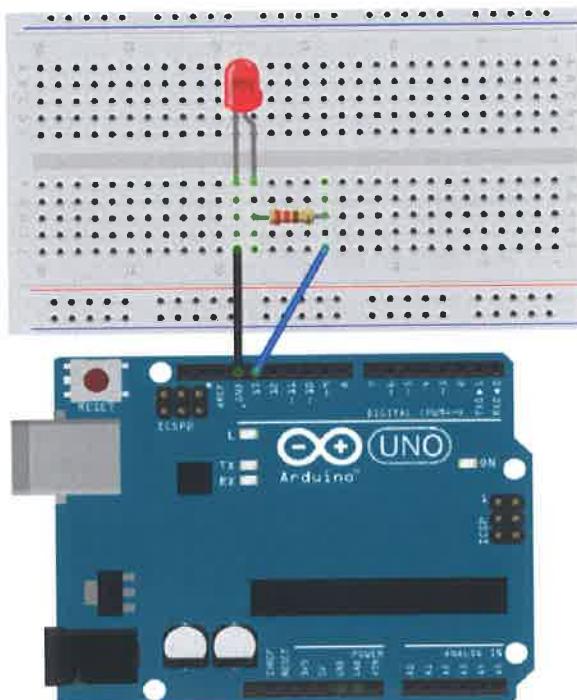
Dette projekt viser noget af det mest simple, du kan gøre med et Arduino UNO board. Du skal se, at digitale output fra en pin kan få en lysdiode til at blinke.

Materialer:

- Arduino UNO
- USB kabel
- Breadboard
- Ledninger
- En lysdiode (LED)
- En resistor (220 ohm)

Kredsløbsopgave 1B

Du skal kunne få en lysdiode til at blinke, indtil du fjerner USB kablet fra dit Arduino UNO board.

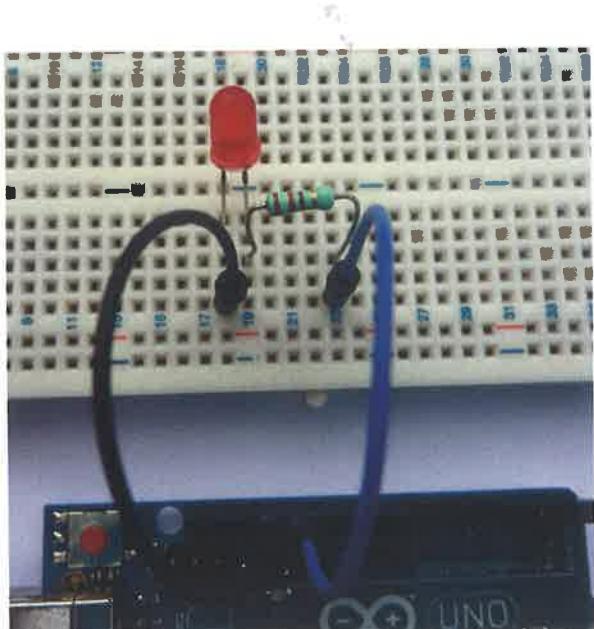


Komponentplacering.

Komponentplacering på breadboard

Lav kredsløbet som vist på tegningen nederst til venstre og forbind Arduino UNO boardet til computeren med et USB kabel. Som du kan se på tegningen, ligner dette kredsløbet fra den forrige opgave. Trykknap kontakten er fjernet, og resistoren er blevet forbundet til den digitale pin 13.

Lysdiodens korte ben er stadig tilsluttet **GND**, men på den anden side af boardet.



Projekt 1B.

Programmet

Nu skal du igen hente blink programmet i Arduino IDE.

Fil - Eksempler - 01.Basics - Blink

Husk også at gå ind under **Værktøjer - Board** og vælg **Arduino UNO** og vælg under **Værktøjer - Port** en ledig port.

Projekt 1B



```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.
Most Arduinos have an on-board LED you can control. On the Uno and
Leonardo, it is attached to digital pin 13. If you're unsure what
pin the on-board LED is connected to on your Arduino model, check
the documentation at http://arduino.cc
This example code is in the public domain.
*/
```

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

// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 13 as an output.
    pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second
    digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
    delay(1000); // wait for a second
}
```

Upload programmet til Arduino UNO boardet. Hvis lysdioden på breadboardet begynder at blinke én gang i sekundet, så er opgaven lykkedes for dig.

Selve programmet ser meget uoverskueligt

ud. Så derfor skal du se lidt nærmere på de enkelte dele i programmet. Et program skrives som en tekst på samme måde som et dokument i et tekstbehandlingsprogram. Før du uploader programmet til et tilsluttet Arduino UNO board kompileres koden i

tekstfeltet. Det betyder at den kode, der er skrevet i tekstueltet, bliver omsat til en ny kode, som mikroprocessoren på Arduino UNO boardet kan forstå.

Anatomien af et Arduino program

Den grundlæggende struktur af programmeringssproget til Arduino boardet er enkelt. Programmer vil altid indeholde mindst to sektioner, den ene er **setup()** funktionen, som er den forberedende del. Den anden sektion er **loop()** funktionen, som er den udørende del. Når du har uploadet et program

til Arduino UNO boardet, forbliver det gemt i mikrocontrollerens hukommelse. Tilslutter du igen dit Arduino UNO board til en strømkilde (USB kabel eller ekstern strømforsyning), vil mikrocontrolleren begynde at afvikle samme program.

Der er ingen tænd/sluk knap på et Arduino UNO board. For at stoppe programmet, skal du fjerne forbindelsen til strømforsyningen. Som du kan se på nedenstående sektioner, består blink programmet af fire sektioner. Disse sektioner skal du nu se nærmere på.

Sektion 1

```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.
Most Arduinos have an on-board LED you can control. On the Uno and
Leonardo, it is attached to digital pin 13. If you're unsure what
pin the on-board LED is connected to on your Arduino model, check
the documentation at http://arduino.cc
This example code is in the public domain.
*/
```

Sektion 2

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

Sektion 3

```
// the setup function runs once when you press reset or power the board
void setup() {
// initialize digital pin 13 as an output.
pinMode(13, OUTPUT);
}
```

Sektion 4

```
// the loop function runs over and over again forever
void loop() {
digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
delay(1000); // wait for a second
digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
delay(1000); // wait for a second
}
```

Forstå programmet

Sektion 1 - Kommentarer

Den første del af programmet er kommentarer, som fortæller dig, hvad meningen er med programmet. Fx at lysdioden gentagne gange vil lyse i et sekund, og herefter slukkes i et sekund.

Kommentarer i et program er ikke egentlig programkode, de ignoreres af compileren og overføres ikke til Arduino UNO boardet.

Der findes to slags kommentarer. Den ene indeholder flere linjer den anden en enkelt linje.

Kommentarer over flere linjer starter med tegnene /* og afsluttes med */.

En enkelt linje med en kommentar starter bare med //.

Læg mærke til, at Arduino IDE automatisk ændre farven på tekster i kommentarer til grå.

I blink programmet kan du se en anvendelse af begge typer af kommentarer.

Kommentarer er meget nyttige, og du opfordres til at indsætte kommentarer i dine egne programmer, så du bedre kan huske, hvilken betydning de enkelte programlinjer har. Kommentarer kan indsættes i alle fire sektioner.

Sektion 2 - Erklære variable

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

En variabel opretter en plads i Arduino UNO boardets hukommelse. Forestil dig at en variabel er en lille kasse, hvor du både kan gemme og hente data.

Der er tre ting, som kendtegner variabler.

Variabler har

- en typebetegnelse
- et navn
- en værdi

At oprette en variabel kaldes også at erklære en variabel.

`int ledPin = 13;` opretter en variabel at typen int med navnet ledPin, som har værdien 13.

Læg mærke til at erklæringen af en variabel skal afsluttes med et semikolon.

Rent praktisk betyder det, at det længste ben på lysdioden(LED) er forbundet med den digitale pin 13. Det korte ben er tilsluttet GND. `int` er en forkortelse for Integer (heltal). De heltal du kan gemme af typen `int`, er hele positive eller hele negative tal. Decimaltal kan du ikke gemme i denne variabel, da de ikke er af typen `int`. Der er en nedre og en øvre grænse for heltal af typen `int`: -32.768 og 32.767. Hvis du også vil bruge negative eller positive heltal uden for de to grænser, skal du bruge en variabel af typen `long`.

Navngivning af variable

Du kan næsten selv bestemme hvad en variabel skal hedde. Der findes dog nogle regler, som skal overholdes. Et navn på en variabel skal begynde med et bogstav. Resten af navnet kan bestå af bogstaver, tal eller understregningstegn. Softwaren skelner mellem store og små bogstaver. Endelig kan du ikke bruge nøgleord som fx main, switch, while som navne for variabler. Alle nøgleord i Arduino sproget bliver vist med en farve i Arduino IDE

Sektion 3 - setup()

Setup() er en funktion som udføres automatisk i begyndelsen af et Arduino program. Funktionen udføres en gang, og kun én gang

ved begyndelsen af programmet i modsætning til funktionen loop() i sektion 4.

Under setup() findes der nogle generelle instrukser, som er forberedelser til det program, der udføres under funktionen loop().

De krøllede parenteser efter `setup()` er med til at holde sammen på linjer af koder. Som du senere skal se, findes de også under funktionen `loop()` i sektion 4.

Nøgleordet `void` foran `setup()` fortæller, at mikrocontrolleren ikke forventer, at få returneret nogen form for information. Det samme gælder `void loop()` i sektion 4.

`pinMode(13, OUTPUT);`

Funktionen `pinMode(pinNumber, mode)` har to parametre (størrelser).

- `pinNumber`
- `mode`

Den første parameter er nummeret på den pin, det handler om. I dette eksempel er det pin 13.

Mode (tilstand) kan være enten `INPUT` eller `OUTPUT`.

I dette tilfælde, hvor du skal have en lysdiode til at lyse, vælger du `OUTPUT` som den anden parameter.

Sektion 4 - loop()

Funktionen `loop()` bliver udført efter funktionen `setup()` og gentages igen og igen indtil strømforsyningen til Arduino UNO boardet afbrydes.

`digitalWrite(13, HIGH); // turn the LED on
(HIGH is the voltage level)`

Som du kan se har funktionen `digitalWrite` to parametre. Den første parameter (13) er

Projekt 1B

nummeret på den pin, det drejer sig om. Den anden parameter fortæller om spændingen på pin 13 skal være +5V (**HIGH**) eller 0V (**LOW**). Hvis lysdioden tilsluttet pin 13 skal lyse, så skal den anden parameter være **HIGH**. Hvis lysdioden ikke skal lyse, vælges **LOW**. Funktionen **digitalWrite(13, HIGH);** får altså lysdioden tilsluttet pin 13 til at lyse.

```
delay(1000); // wait for a second
```

Funktionen **delay(timeInMs)** indlægger en pause i programmet i et bestemt tidsrum. Tiden måles i millisekunder. For eksempel vil **delay(1000)** give en pause på 1000 millisekunder = 1 sekund.

```
digitalWrite(13, LOW); // turn the LED off  
by making the voltage LOW
```

Nu bliver den anden parameter udskiftet med **LOW** i stedet for **HIGH**. Det får den betydning, at lysdioden slukkes efter at have været tændt i 1 sekund.

```
delay(1000);
```

Nu holdes der igen en lille pause i programmet, hvor lysdioden er slukket i et sekund. Herefter begynder loop () funktionen igen forfra. Det samlede resultat får lysdioden til at blinke.

Øvelser til projekt 1B

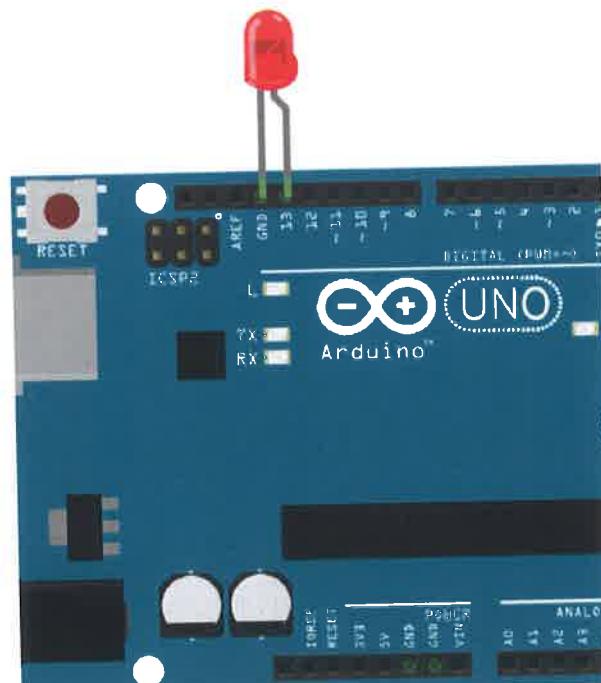
- Prøv at ændre i programmet så lysdioden er tændt i 100 millisekunder og slukket i 100 millisekunder.
- Prøv at ændre i programmet så lysdioden er tændt i 100 millisekunder og slukket i 900 millisekunder.
- Kan du ændre programmet og få lysdioden til at blinke, når den er tilsluttet pin 3 og GND?

Hvad sker hvis du ændrer i programmet så lysdioden er tændt i 50 millisekunder og slukket i 50 millisekunder? Prøv at tage ved Arduino UNO boardet og bevæg det hurtigt fra side til side. Hvad ser du?

Pin 13

Pin 13 er lidt speciel, når den skal forbindes til en lysdiode. Det er nemlig ikke nødvendigt at bruge en resistor for at begrænse strømstyrken gennem lysdioden, fordi der i forvejen er indbygget en resistor til pin 13 på Arduino UNO boardet.

Prøv at lave Projekt 1B igen, men med denne enkle kredsløbsopstilling, som vist på nedenstående tegning.



Kom godt i gang med Arduino

Bogen er den første bog på dansk med tilhørende websted, som giver en introduktion til det spændende Arduino univers og kan bruges uden erfaringer indenfor elektronik og programmering.

Arduino platformen er et fantastisk værktøj, da den indeholder både hardware og software. Da begge dele er open-source, har Arduino ved hjælp af internettet fået en enestående udbredelse overalt i verden. Tusindvis af nybegyndere og professionelle beskæftiger sig med Arduino platformen.

Den gratis software til Arduino findes både til Windows, Mac OS X og Linux.

Med Arduino er det i dag muligt at lære ny teknologi ved hjælp af sjove og spændende aktiviteter.

Pludselig bliver elektronik og programmering et kreativt værktøj, som alle kan bruge på samme måde som pensel og maling.

Som underviser bliver du præsenteret for Arduino og få inspiration til at inddrage moderne teknologi i din undervisning.

I bogen findes en del projekter, som kan bruges af forældre til at lære deres lidt større børn om programmering og elektronik, mens I bygger spændende og brugbare projekter sammen.

Bogen indeholder 50 projekter. Alle de større programmer kan hentes fra bogens websted: <http://www.myarduino.dk/>

Bogens websted indeholder også en hel del videoklip af de endelige projekter.

I slutningen af bogen findes en liste over de komponenter, der anvendes i bogen.

ISBN 978-87-998959-0-8



væk
62.3
Schmidt, Hans Jørn
Min Arduino



5118412852
5 238 733 7
12.07.2016

hb

Arduino UNO boardet

USB stik

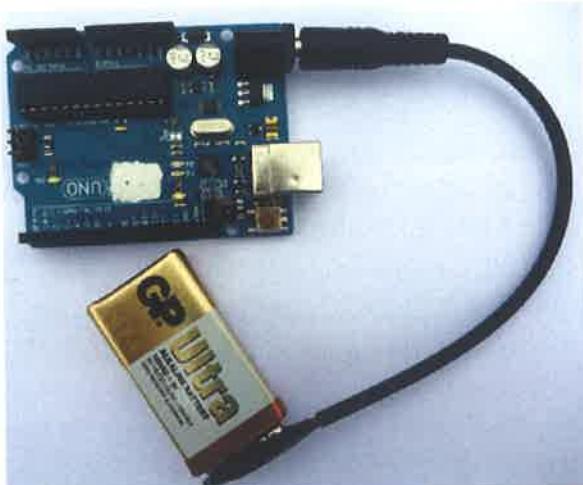
På Arduino UNO boardet er der et USB stik, som bruges, når der skal overføres data til og fra boardet. Desuden leverer USB stikket strøm til boardet fra computeren.



USB kabel tilsluttet et Arduino UNO board.

Power jackstikket

I mange tilfælde er det upraktisk, at have boardet permanent tilsluttet computeren. Så derfor kan stikket bruges til at tilslutte en ekstern spændingskilde (batteri eller batteriadapter). Det anbefales at spændingen til Arduino UNO boardet er 7-12V.



Arduino UNO boardet tilsluttet et batteri på 9V med et jackstik.

Reset

Arduino UNO boardet er udstyret med en

Reset knap, som kan genstarte programmet eller stoppe programmet, når den holdes nede et stykke tid.

Power pins

Når Arduino UNO boardet er tilsluttet en computer, kan det bruges som en strømforsyning. Som + - pol kan du enten anvende pin 5V eller pin 3,3 V. Som - - pol kan du anvende en af de tre steder, hvor der står **GND** (ground). De tre pins mærket **GND** er alle indbyrdes forbundet. Du vælger selv hvilken af dem, der er mest praktisk at bruge i forhold til et projekt.

I bogen bruger vi ikke betegnelsen - - pol, men kun GND eller 0 V.

Vin betyder Voltage in (volt ind) og er en pin, du kan bruge i forbindelse en ekstern strømforsyning.

LED'er

På Arduino UNO boardet er der placeret fire meget små lysdioder (LED'er) mærket **L**, **RX**, **TX** og **ON**. En LED er en komponent der producerer lys, når der løber en elektrisk strøm gennem den.

De fire lysdioder bruges alle til at angive en aktivitet på boardet:

Når **ON** lyser grønt, betyder det, at din Arduino er tændt.

RX og **TX** lyser, når der modtages eller sendes data.

L er en LED, der er forbundet med den digitale pin 13. Det var den lysdiode, du brugte, da du testede blink programmet.

Når du arbejder med dit Arduino UNO board er det vigtigt, at du ikke kommer til at lægge det på ting, som kan lede den elektriske strøm. Du kan risikere at kortslutte et kredsløb på Arduino UNO boardet og hermed ødelægge det.

Shields

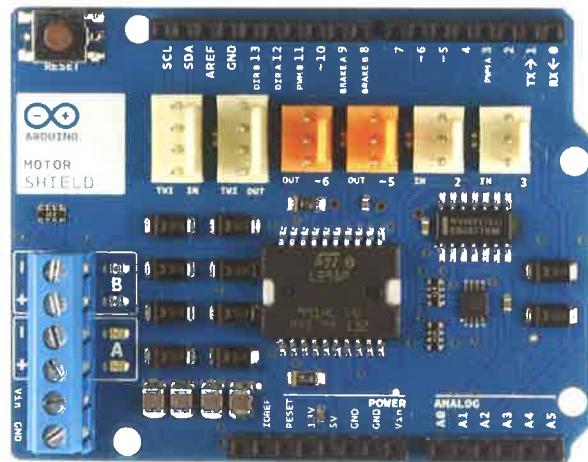
Typer af shields

Til Arduino UNO boardet findes der nogle hardware udbygninger, de såkaldte shields (skjolde). Disse shields bruges til at udvide funktionaliteten af Arduino UNO boardet, der bl.a. giver det mulighed for at kontrollere en gruppe motorer, blive koblet til Wi-Fi eller læse et SD kort.

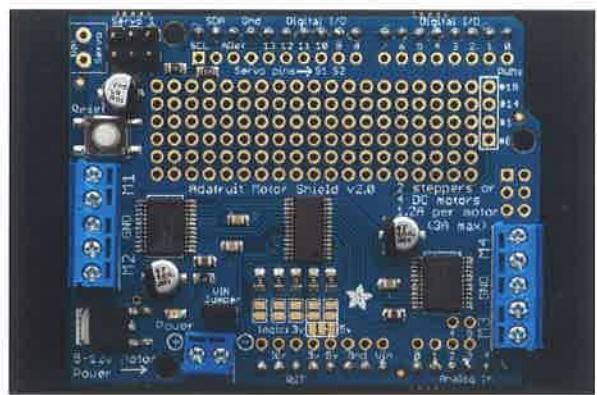
Stifterne på undersiden af et shield passer ned i plastikbøsningerne (Female Headers) på Arduino UNO boardet. Mange shields er selv udstyret med plastikbøsninger på oversiden, så de kan stables på andre shields. På denne måde kan du fx styre en 2WD eller 4WD robot via Wi-Fi.

Motor Shield

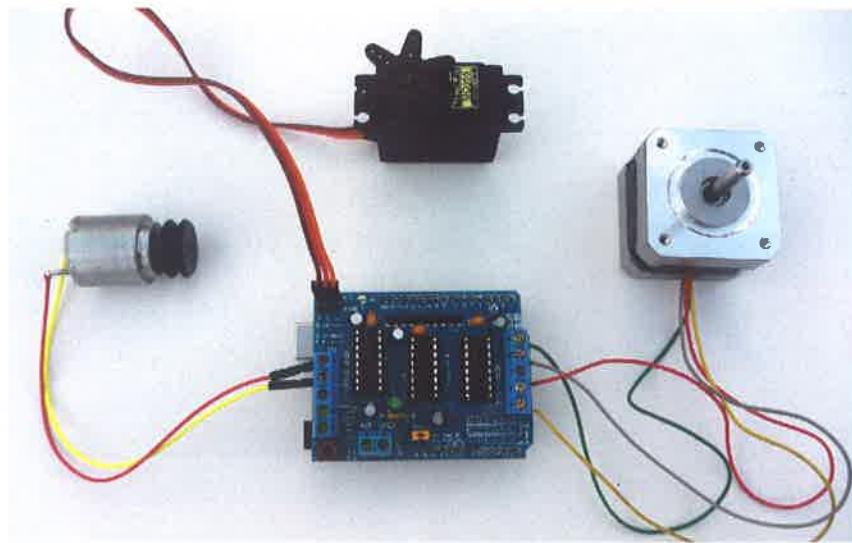
Nedenstående billede viser et meget populært motor shield (version 1.0) fra Adafruit Industries. Boardet kan tilsluttes to servomotorer, to steppermotorer eller op til fire DC motorer. Det anvendes bl.a. her i bogen.



Arduino Motor Shield.



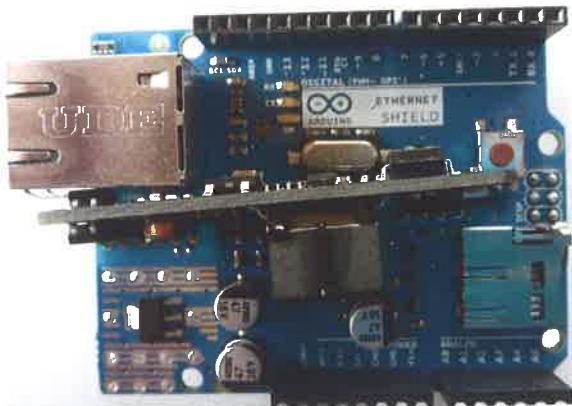
Adafruit Motor Shield (version 2.0).



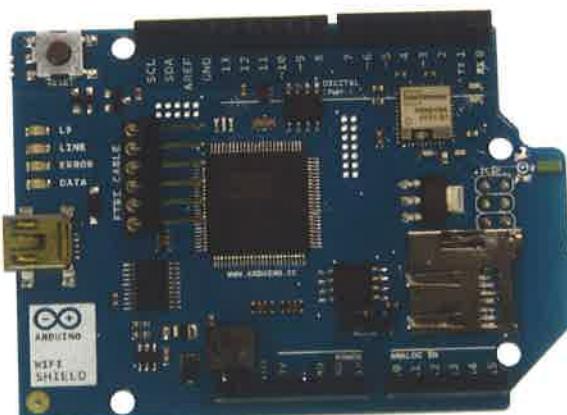
Adafruit Motor Shield (version 1.0).

Shields

Ethernet Shield



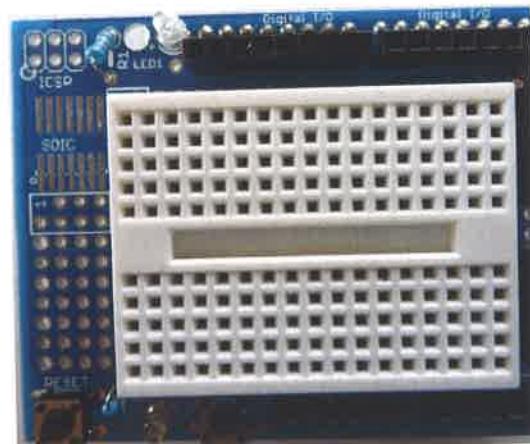
Wi-Fi Shield



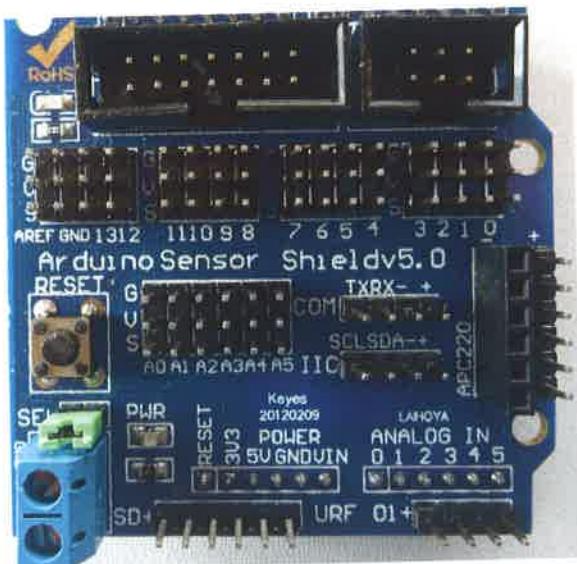
LCD & Keypad Shield



Prototyping Shield



Arduino Sensor Shield



I mange tilfælde kan det være en god idé, at anvende et Prototyping Shield og et mini breadboard for at lave en opstilling af et kredsløb mindre kompliceret.

På siden er der kun vist et lille udvalg af shields, som du kan anvende i forbindelse med et Arduino UNO board.

Sensorer og aktuatorer



Sensorer er elektroniske enheder, som tilsluttes Arduino UNO boardet, og som modtager data fra de fysiske omgivelser. Det kunne være en lyssensor, en afstandssensor, en temperatursensor eller noget så enkelt som en trykknap kontakt. Sensorer tilsluttes INPUT pins på Arduino UNO boardet. På baggrund af disse data kan et program få Arduino UNO boardet til at styre aktuatorer, som kan påvirke omgivelserne. Eksempler på aktuatorer kunne fx være lysdioder, motorer eller en piezo buzzer. Aktuatorer tilsluttes OUTPUT pins på Arduino UNO boardet. Eller sagt meget enkelt: Arduino UNO boardet lytter til sensorer og taler til aktuatorer.

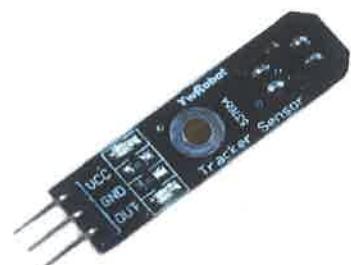
Eksempler på sensorer



Trykknap kontakt.



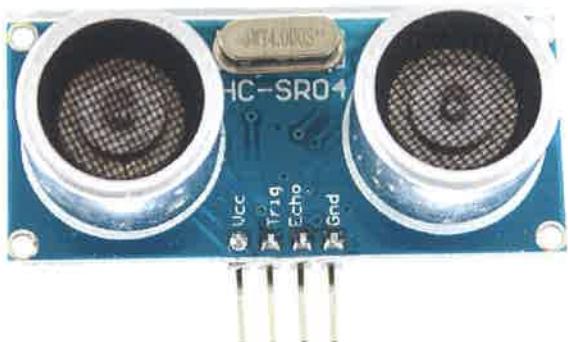
Temperatur sensor.



IR tracker sensor.



Anvendelse af IR tracker sensorer.

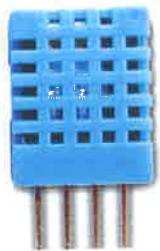


Afstandssensor.



LDR - lysfølsom resistor.

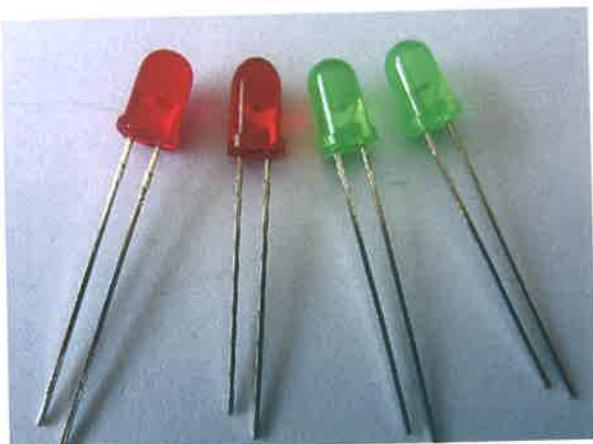
Sensorer og aktuatorer



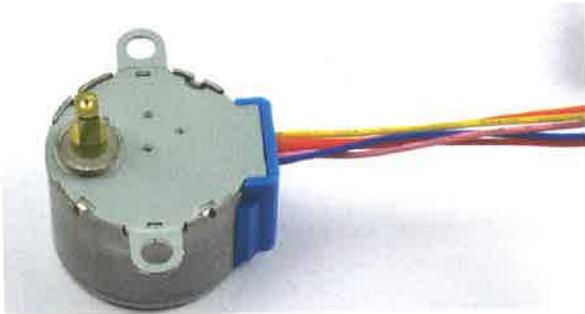
Fugtigheds- og temperatursensor.



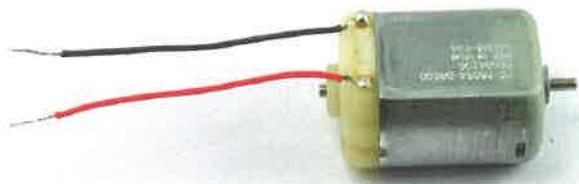
Servomotor.



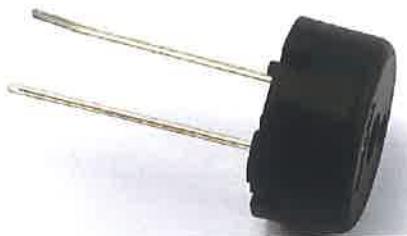
Lysdioder.



Stepper motor.



DC motor.



Piezo buzzer.



To gear DC motorer med hjul.

Dette er kun et lille udvalg af sensorer og aktuatorer, som findes til Arduino.

Breadboard og ledninger

Om breadboard og ledninger

Breadboard

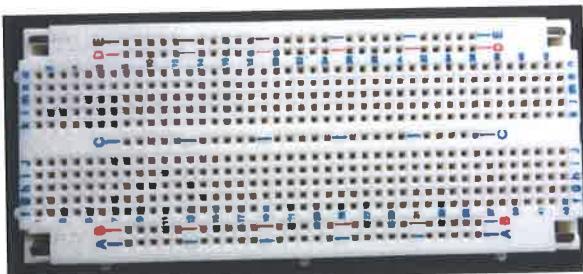
Et breadboard erstatter tidligere tiders "sømbræt", hvor man brugte loddekolber og loddetin til at fastgøre komponenter på sømhoveder. Breadboardet er dog langt smartere, idet du blot stikker komponenterne ned i hullerne på breadboardet, hvorefter der automatisk er forbindelse mellem dem. Der skal ikke loddes, så det er nemt at ændre tilslutninger og erstatte komponenter. Komponenterne vil ikke blive beskadiget, og kan bruges bagefter.

Breadboards findes i mange forskellige for-

mer og størrelser. De to følgende billeder viser et standart breadboard set fra forsiden og fra bagsiden. Dette breadboard er opbygget af 5 (A, B, C, D og E) langsgående forbindelser med hver 25 huller. I de to brede "striber" er der to gange 43 tværgående forbindelser med hver 5 huller.

På undersiden af breadboardet kan du se, hvordan kobberbanerne forbinder de forskellige grupper af huller vandret og lodret.

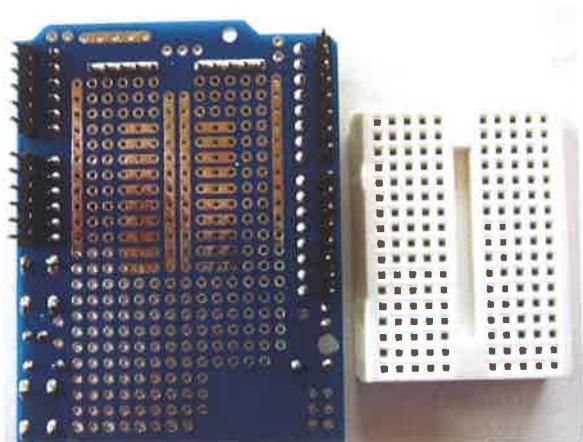
Det sker nogle gange, at du kan klare dig med et mindre breadboard. Nederst på siden kan du se et mini breadboard, som ofte bruges sammen med et Arduino Prototyping Shield.



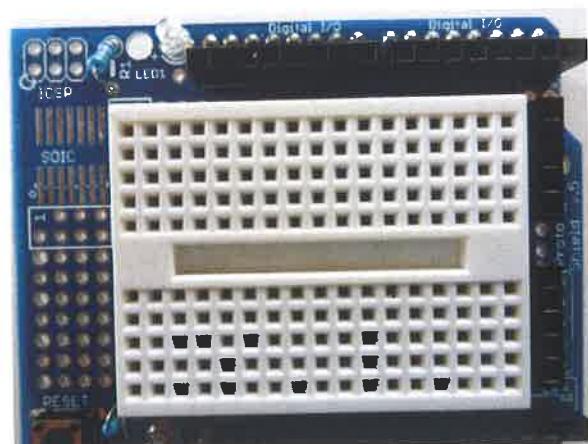
Overside på breadboard.



Underside på breadboard.



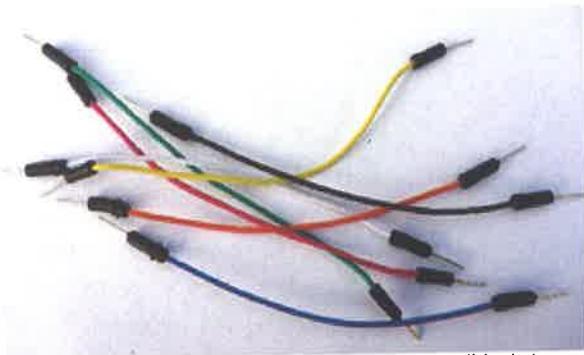
Prototyping Shield med et mini breadboard.



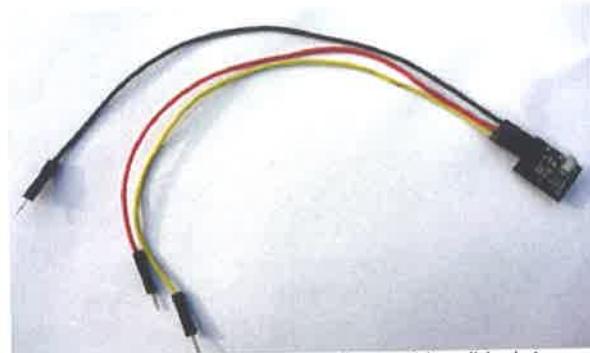
Prototyping Shield med et fastgjort mini breadboard.

Breadboard og ledninger

Ledninger



"Han til han" ledninger.



"Han til hun" ledninger.

Ovenstående billede viser ledninger af typen "han til han", som oftest bliver brugt til Arduino projekter. Det er en fordel at have ledningerne i forskellige farver for lettere at holde styr på forbindelserne til de forskellige komponenter på breadboardet og Arduino UNO boardet.

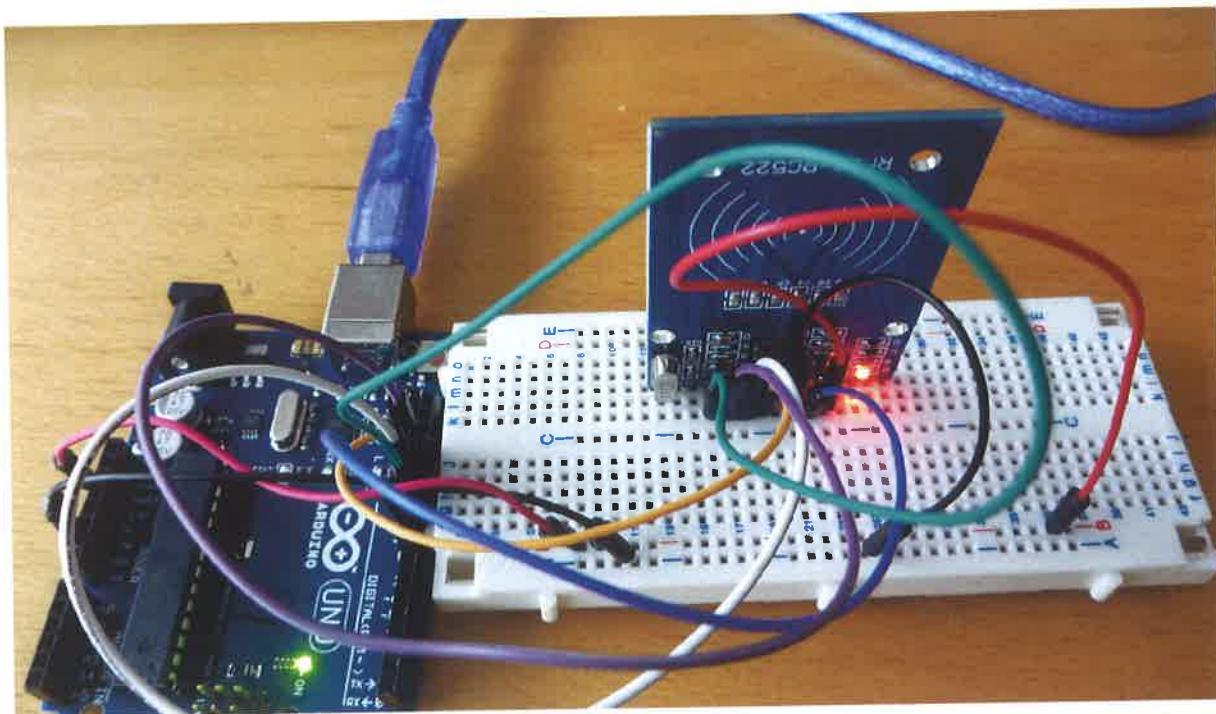
Det kan også anbefales at have nogle ledninger af typen "han til hun". De bruges til at forbinde nogle typer af komponenter og sensorer til Arduino UNO boardet.

På billedet øverst til højre er der brugt led-

ninger af typen "han til hun" i forbindelse med en IR receiver.

Om farven på ledninger

I bogen bruges de sorte ledninger til at forbinde enheder med GND og de røde ledninger til at forbinde enheder med +5V eller +3,3V. Hvilken farve ledninger du ellers vil bruge, bestemmer du selv. I opstillingen nederst på siden er der brugt syv forskellige farver.



I denne projekt er der brugt ledninger med syv forskellige farver.

Projekt 1A

Tænd en lysdiode med en tryknap kontakt

I Projekt 1A skal du blive fortrolig med at forbinde komponenter, breadboard og Arduino UNO boardet med ledninger efter en kredsløbsbeskrivelse.

I projektet skal du få en lysdiode til at lyse, når du trykker på tryknap kontakten. Arduino UNO boardet, som er forbundet til computeren med et USB kabel, bruges kun som strømforsyning.

Materialer

- Arduino UNO
- USB kabel
- Breadboard
- Ledninger
- En lysdiode (LED)
- En resistor (220 ohm)
- Tryknap kontakt (Off/On kontakt)

Om komponenterne i materialelisten:

Lysdiode

En lysdiode er en komponent, som konverterer elektrisk energi til lysenergi. Lysdioden er en polariseret komponent, hvilket betyder, at den kun kan lede strømmen i én retning. Det længste ben på en lysdiode kaldes anoden, og skal tilsluttes **+5V**. Det korte ben kaldes katoden og tilsluttes **GND** (ground). Da en lysdiode maksimalt må gennemløbes af en strømstyrke på ca. 20 mA, vil der i de fleste tilfælde være indsat en resistor i serie med denne. Lysdioder findes både i røde, grønne, gule, blå og hvide udgaver.

Resistor

En resistor bruges først og fremmest til at begrænse strømstyrken i elektriske kredsløb. Resistans måles i ohm (Ω). Værdien i ohm af en resistor kan aflæses ved hjælp af de farvede bånd på resistoren. For at få oversat rækken af farver til et tal, kan du enten bruge tabellen over farvekoder på den næste side eller installere en gratis app på fx din mobiltelefon. Du skal være opmærksom på, at resistorer både findes i udgaver med 4 og 5 farvede bånd. Det er en god idé at have et multimeter til sin rådighed, når du beskæftiger dig med elektronik, så kan du altid få tjekket om fx resistoren har den værdi, som du regner med.

Tryknap kontakt

Hvis en tryknap kontakt holdes nede i et elektrisk kredskøb, vil strømmen uhindret passere den. Hvis tryknap kontakten igen slippes, bliver den elektriske strøm afbrudt. Tryknap kontakter findes i udgaver med både 4 ben og 2 ben.

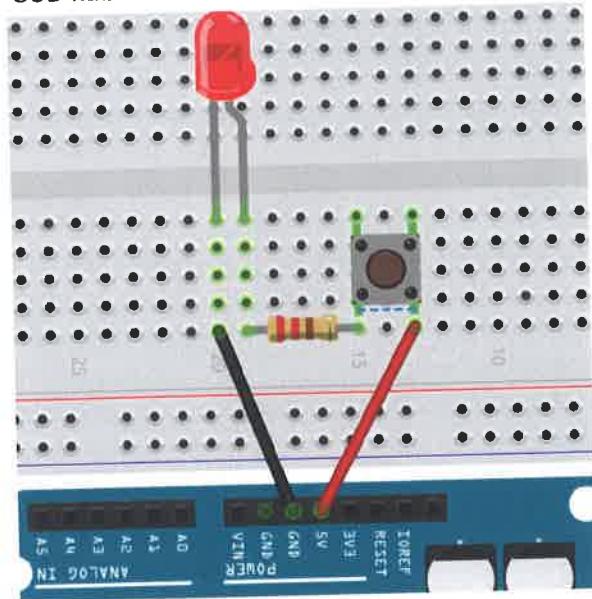


Multimeter.

Projekt 1A

Kredsløbsopgave 1A

Du skal bygge et kredsløb, der kan få en lysdiode til at lyse, når du trykker på trykknap kontakten. Dit Arduino UNO board skal være tilsluttet din tændte computer med et USB kabel.



Komponentplacering.

Komponentplacering på breadboard

Lav kredsløbet, som er vist på tegningen, og forbind Arduino UNO boardet til en tændt computer med et USB kabel. Lysdiodens længste ben i samme række som resistorens ene ben.

Det er en god idé altid at bruge sorte ledninger, som er forbundet til GND og røde ledninger, som er forbundet til +5V.

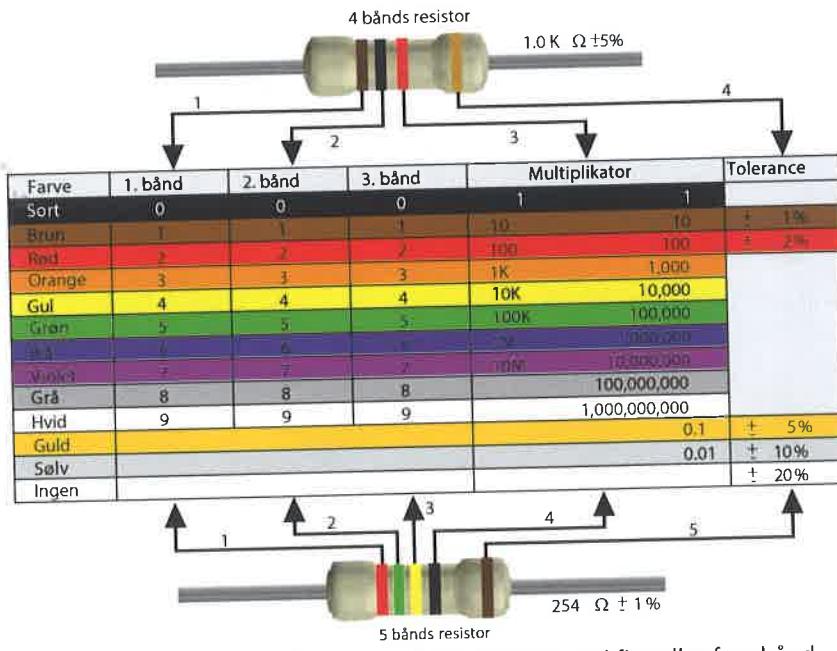
Hvad sker der hvis du trykker på trykknap kontakten og slipper den igen?

Nu skulle du gerne se, at lysdioden lyser og går ud igen, når du slipper kontakten. Hvis der ikke sker noget så gennemgå kredsløbet igen og se om komponenterne sidder i de rigtige huller på breadboardet.

Undersøg også om lysdioden vender rigtigt. Lysdiodens længste ben skal sidde i samme række på breadboardet som resistoren og lysdiodens korte ben skal være tilsluttet **GND**. Resistoren kan ikke vende forkert.

Hvis du anvender en tryknap kontakt med 4 ben, skal du også tjekke, om den vender rigtigt.

Farvekoder for resistorer



Farvekoder for resistorer med fire eller fem bånd.

Projekt 1B

Blink program med en lysdiode

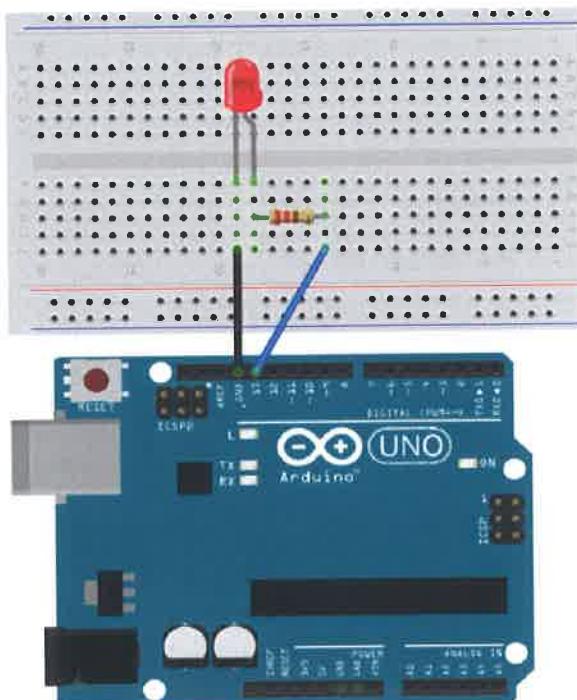
Dette projekt viser noget af det mest simple, du kan gøre med et Arduino UNO board. Du skal se, at digitale output fra en pin kan få en lysdiode til at blinke.

Materialer:

- Arduino UNO
- USB kabel
- Breadboard
- Ledninger
- En lysdiode (LED)
- En resistor (220 ohm)

Kredsløbsopgave 1B

Du skal kunne få en lysdiode til at blinke, indtil du fjerner USB kablet fra dit Arduino UNO board.

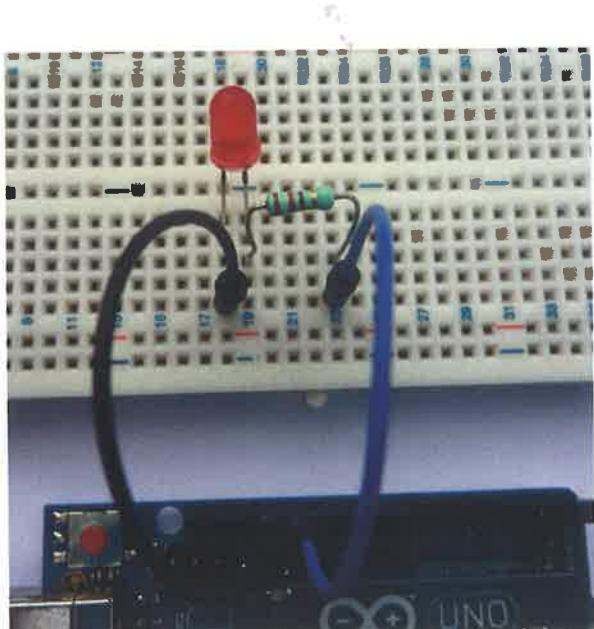


Komponentplacering.

Komponentplacering på breadboard

Lav kredsløbet som vist på tegningen nederst til venstre og forbind Arduino UNO boardet til computeren med et USB kabel. Som du kan se på tegningen, ligner dette kredsløbet fra den forrige opgave. Trykknap kontakten er fjernet, og resistoren er blevet forbundet til den digitale pin 13.

Lysdiodens korte ben er stadig tilsluttet **GND**, men på den anden side af boardet.



Projekt 1B.

Programmet

Nu skal du igen hente blink programmet i Arduino IDE.

Fil - Eksempler - 01.Basics - Blink

Husk også at gå ind under **Værktøjer - Board** og vælg **Arduino UNO** og vælg under **Værktøjer - Port** en ledig port.

Projekt 1B



```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.
Most Arduinos have an on-board LED you can control. On the Uno and
Leonardo, it is attached to digital pin 13. If you're unsure what
pin the on-board LED is connected to on your Arduino model, check
the documentation at http://arduino.cc
This example code is in the public domain.
*/
```

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

// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 13 as an output.
    pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second
    digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
    delay(1000); // wait for a second
}
```

Upload programmet til Arduino UNO boardet. Hvis lysdioden på breadboardet begynder at blinke én gang i sekundet, så er opgaven lykkedes for dig.

Selve programmet ser meget uoverskueligt

ud. Så derfor skal du se lidt nærmere på de enkelte dele i programmet. Et program skrives som en tekst på samme måde som et dokument i et tekstbehandlingsprogram. Før du uploader programmet til et tilsluttet Arduino UNO board kompileres koden i

tekstfeltet. Det betyder at den kode, der er skrevet i tekstueltet, bliver omsat til en ny kode, som mikroprocessoren på Arduino UNO boardet kan forstå.

Anatomien af et Arduino program

Den grundlæggende struktur af programmeringssproget til Arduino boardet er enkelt. Programmer vil altid indeholde mindst to sektioner, den ene er **setup()** funktionen, som er den forberedende del. Den anden sektion er **loop()** funktionen, som er den udørende del. Når du har uploadet et program

til Arduino UNO boardet, forbliver det gemt i mikrocontrollerens hukommelse. Tilslutter du igen dit Arduino UNO board til en strømkilde (USB kabel eller ekstern strømforsyning), vil mikrocontrolleren begynde at afvikle samme program.

Der er ingen tænd/sluk knap på et Arduino UNO board. For at stoppe programmet, skal du fjerne forbindelsen til strømforsyningen. Som du kan se på nedenstående sektioner, består blink programmet af fire sektioner. Disse sektioner skal du nu se nærmere på.

Sektion 1

```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.
Most Arduinos have an on-board LED you can control. On the Uno and
Leonardo, it is attached to digital pin 13. If you're unsure what
pin the on-board LED is connected to on your Arduino model, check
the documentation at http://arduino.cc
This example code is in the public domain.
*/
```

Sektion 2

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

Sektion 3

```
// the setup function runs once when you press reset or power the board
void setup() {
// initialize digital pin 13 as an output.
pinMode(13, OUTPUT);
}
```

Sektion 4

```
// the loop function runs over and over again forever
void loop() {
digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
delay(1000); // wait for a second
digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
delay(1000); // wait for a second
}
```

Forstå programmet

Sektion 1 - Kommentarer

Den første del af programmet er kommentarer, som fortæller dig, hvad meningen er med programmet. Fx at lysdioden gentagne gange vil lyse i et sekund, og herefter slukkes i et sekund.

Kommentarer i et program er ikke egentlig programkode, de ignoreres af compileren og overføres ikke til Arduino UNO boardet.

Der findes to slags kommentarer. Den ene indeholder flere linjer den anden en enkelt linje.

Kommentarer over flere linjer starter med tegnene /* og afsluttes med */.

En enkelt linje med en kommentar starter bare med //.

Læg mærke til, at Arduino IDE automatisk ændre farven på tekster i kommentarer til grå.

I blink programmet kan du se en anvendelse af begge typer af kommentarer.

Kommentarer er meget nyttige, og du opfordres til at indsætte kommentarer i dine egne programmer, så du bedre kan huske, hvilken betydning de enkelte programlinjer har. Kommentarer kan indsættes i alle fire sektioner.

Sektion 2 - Erklære variable

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

En variabel opretter en plads i Arduino UNO boardets hukommelse. Forestil dig at en variabel er en lille kasse, hvor du både kan gemme og hente data.

Der er tre ting, som kendtegner variabler.

Variabler har

- en typebetegnelse
- et navn
- en værdi

At oprette en variabel kaldes også at erklære en variabel.

`int ledPin = 13;` opretter en variabel at typen int med navnet ledPin, som har værdien 13.

Læg mærke til at erklæringen af en variabel skal afsluttes med et semikolon.

Rent praktisk betyder det, at det længste ben på lysdioden(LED) er forbundet med den digitale pin 13. Det korte ben er tilsluttet GND. `int` er en forkortelse for Integer (heltal). De heltal du kan gemme af typen `int`, er hele positive eller hele negative tal. Decimaltal kan du ikke gemme i denne variabel, da de ikke er af typen `int`. Der er en nedre og en øvre grænse for heltal af typen `int`: -32.768 og 32.767. Hvis du også vil bruge negative eller positive heltal uden for de to grænser, skal du bruge en variabel af typen `long`.

Navngivning af variable

Du kan næsten selv bestemme hvad en variabel skal hedde. Der findes dog nogle regler, som skal overholdes. Et navn på en variabel skal begynde med et bogstav. Resten af navnet kan bestå af bogstaver, tal eller understregningstegn. Softwaren skelner mellem store og små bogstaver. Endelig kan du ikke bruge nøgleord som fx main, switch, while som navne for variabler. Alle nøgleord i Arduino sproget bliver vist med en farve i Arduino IDE

Sektion 3 - setup()

Setup() er en funktion som udføres automatisk i begyndelsen af et Arduino program. Funktionen udføres en gang, og kun én gang

ved begyndelsen af programmet i modsætning til funktionen loop() i sektion 4.

Under setup() findes der nogle generelle instrukser, som er forberedelser til det program, der udføres under funktionen loop().

De krøllede parenteser efter `setup()` er med til at holde sammen på linjer af koder. Som du senere skal se, findes de også under funktionen `loop()` i sektion 4.

Nøgleordet `void` foran `setup()` fortæller, at mikrocontrolleren ikke forventer, at få returneret nogen form for information. Det samme gælder `void loop()` i sektion 4.

`pinMode(13, OUTPUT);`

Funktionen `pinMode(pinNumber, mode)` har to parametre (størrelser).

- `pinNumber`
- `mode`

Den første parameter er nummeret på den pin, det handler om. I dette eksempel er det pin 13.

Mode (tilstand) kan være enten `INPUT` eller `OUTPUT`.

I dette tilfælde, hvor du skal have en lysdiode til at lyse, vælger du `OUTPUT` som den anden parameter.

Sektion 4 - loop()

Funktionen `loop()` bliver udført efter funktionen `setup()` og gentages igen og igen indtil strømforsyningen til Arduino UNO boardet afbrydes.

`digitalWrite(13, HIGH); // turn the LED on
(HIGH is the voltage level)`

Som du kan se har funktionen `digitalWrite` to parametre. Den første parameter (13) er

Projekt 1B

nummeret på den pin, det drejer sig om. Den anden parameter fortæller om spændingen på pin 13 skal være +5V (**HIGH**) eller 0V (**LOW**). Hvis lysdioden tilsluttet pin 13 skal lyse, så skal den anden parameter være **HIGH**. Hvis lysdioden ikke skal lyse, vælges **LOW**. Funktionen **digitalWrite(13, HIGH);** får altså lysdioden tilsluttet pin 13 til at lyse.

```
delay(1000); // wait for a second
```

Funktionen **delay(timeInMs)** indlægger en pause i programmet i et bestemt tidsrum. Tiden måles i millisekunder. For eksempel vil **delay(1000)** give en pause på 1000 millisekunder = 1 sekund.

```
digitalWrite(13, LOW); // turn the LED off  
by making the voltage LOW
```

Nu bliver den anden parameter udskiftet med **LOW** i stedet for **HIGH**. Det får den betydning, at lysdioden slukkes efter at have været tændt i 1 sekund.

```
delay(1000);
```

Nu holdes der igen en lille pause i programmet, hvor lysdioden er slukket i et sekund. Herefter begynder loop () funktionen igen forfra. Det samlede resultat får lysdioden til at blinke.

Øvelser til projekt 1B

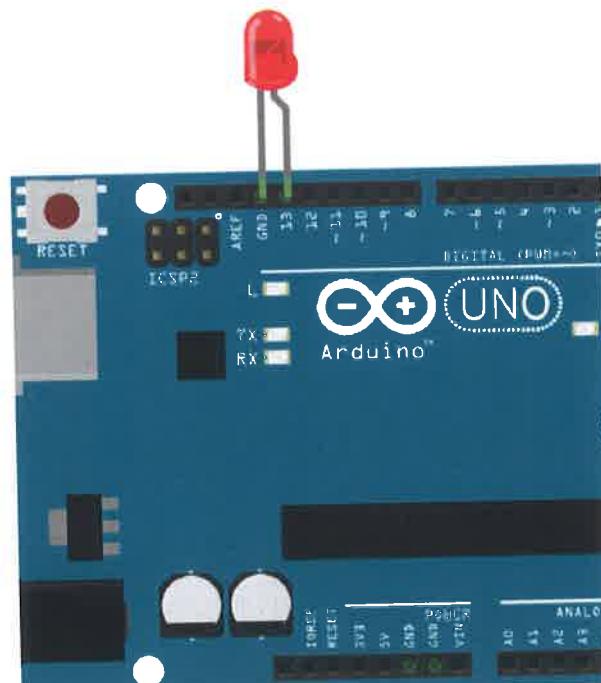
- Prøv at ændre i programmet så lysdioden er tændt i 100 millisekunder og slukket i 100 millisekunder.
- Prøv at ændre i programmet så lysdioden er tændt i 100 millisekunder og slukket i 900 millisekunder.
- Kan du ændre programmet og få lysdioden til at blinke, når den er tilsluttet pin 3 og GND?

Hvad sker hvis du ændrer i programmet så lysdioden er tændt i 50 millisekunder og slukket i 50 millisekunder? Prøv at tage ved Arduino UNO boardet og bevæg det hurtigt fra side til side. Hvad ser du?

Pin 13

Pin 13 er lidt speciel, når den skal forbindes til en lysdiode. Det er nemlig ikke nødvendigt at bruge en resistor for at begrænse strømstyrken gennem lysdioden, fordi der i forvejen er indbygget en resistor til pin 13 på Arduino UNO boardet.

Prøv at lave Projekt 1B igen, men med denne enkle kredsløbsopstilling, som vist på nedenstående tegning.



Kom godt i gang med Arduino

Bogen er den første bog på dansk med tilhørende websted, som giver en introduktion til det spændende Arduino univers og kan bruges uden erfaringer indenfor elektronik og programmering.

Arduino platformen er et fantastisk værktøj, da den indeholder både hardware og software. Da begge dele er open-source, har Arduino ved hjælp af internettet fået en enestående udbredelse overalt i verden. Tusindvis af nybegyndere og professionelle beskæftiger sig med Arduino platformen.

Den gratis software til Arduino findes både til Windows, Mac OS X og Linux.

Med Arduino er det i dag muligt at lære ny teknologi ved hjælp af sjove og spændende aktiviteter.

Pludselig bliver elektronik og programmering et kreativt værktøj, som alle kan bruge på samme måde som pensel og maling.

Som underviser bliver du præsenteret for Arduino og få inspiration til at inddrage moderne teknologi i din undervisning.

I bogen findes en del projekter, som kan bruges af forældre til at lære deres lidt større børn om programmering og elektronik, mens I bygger spændende og brugbare projekter sammen.

Bogen indeholder 50 projekter. Alle de større programmer kan hentes fra bogens websted: <http://www.myarduino.dk/>

Bogens websted indeholder også en hel del videoklip af de endelige projekter.

I slutningen af bogen findes en liste over de komponenter, der anvendes i bogen.

ISBN 978-87-998959-0-8



væk
62.3
Schmidt, Hans Jørn
Min Arduino



5118412852
5 238 733 7
12.07.2016

hb