>>> C-Programmering for begyndere
>>> Del 4 - Microcontrollerprogrammering, Arduino, Hardware og physical computing

Name: Jacob B. Pedersen[†] og Jakob S. Nielsen[‡]
Date: 23. april 2018

[†]jacob.bp@mvb.net
[‡]jakob990@gmail.com

>>> Indhold

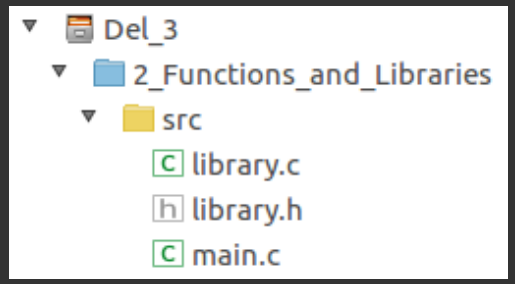
1. Repetition
   - Hvad lavede vi sidste gang?

2. I dag
   - Dagens program
   - Hvad er Arduino
   - Installation af Arduino IDE
   - Introduktion til Arduino-funktionerne
   - Blinky
   - Input/Output
   - Analog I/O

3. Kreative Opgaver

>>> Hvad lavede vi sidste gang? - Libraries

* Vi skrev vores egne funktioner
* Satte dem i libraries
  * Lærte at bruge headers og implementationsfiler:
    * .h og .c




```c
#include "library.h"
```

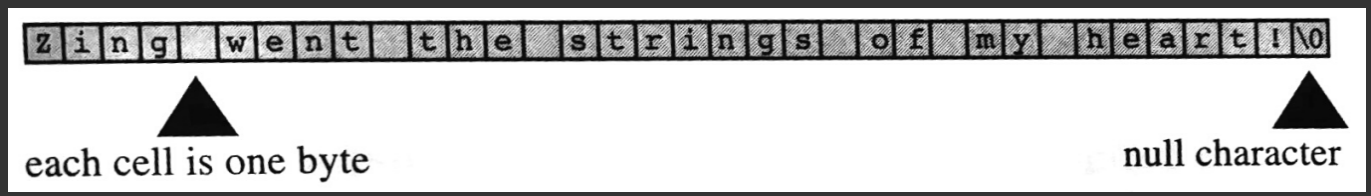
```c
#ifndef LIBRARY_H
#define LIBRARY_H

// Prototype af multiply funktionen:
int multiply(int x, int y);

#endif /* LIBRARY_H */
```

>>> Hvad lavede vi sidste gang? - Arrays

* Vi blev klogere på arrays
  * Og hvordan vi holdt tekststrenge i char arrays
  * Arrays kunne også holde lister af int eller float




```
1       char string[numberOfChars]; // Char array
2       int intArray[number of ints]; // Integer array
3       float floArray[number of floats]; // Float array
```

>>> Hvad lavede vi sidste gang?- Typedef, enum

* Vi fik også lært at repræsentere data vha. egne datatyper:
  * typedef gav en datatype et alias
  * enum gav integer værdier navne i stedet

```
1       // Et smart eksempel på enums kunne være at gøre en
            fysisk udregning overskuelig:
2       typedef float length;
3       typedef float width;
4       typedef float height;
5       typedef float volume;
6
7       // Enums kunne være gode til at skelne mellem ting som
            f.eks. farve:
8       enum color { red, orange, yellow, green, blue, purple };
```

>>> Hvad lavede vi sidste gang? - Structs, unions

* Til sidst organiserede vi data vha. structs og unions
  * structs var en samling af data

```
1        struct book{
2           char title[40]; // Bogens titel
3           char author[40]; // Bogens forfatter
4           float price; // Bogens pris
5           int stars; // Bogens rating fra 1-5
6           };
```

  * union kunne antage mere end én type

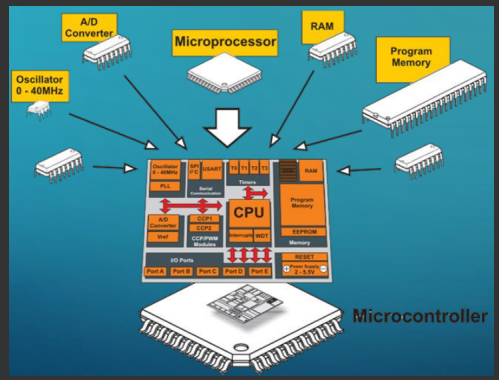
```
1        union varchar{
2           int digit;
3           double bigFloat;
4           char letter;
5           };
```

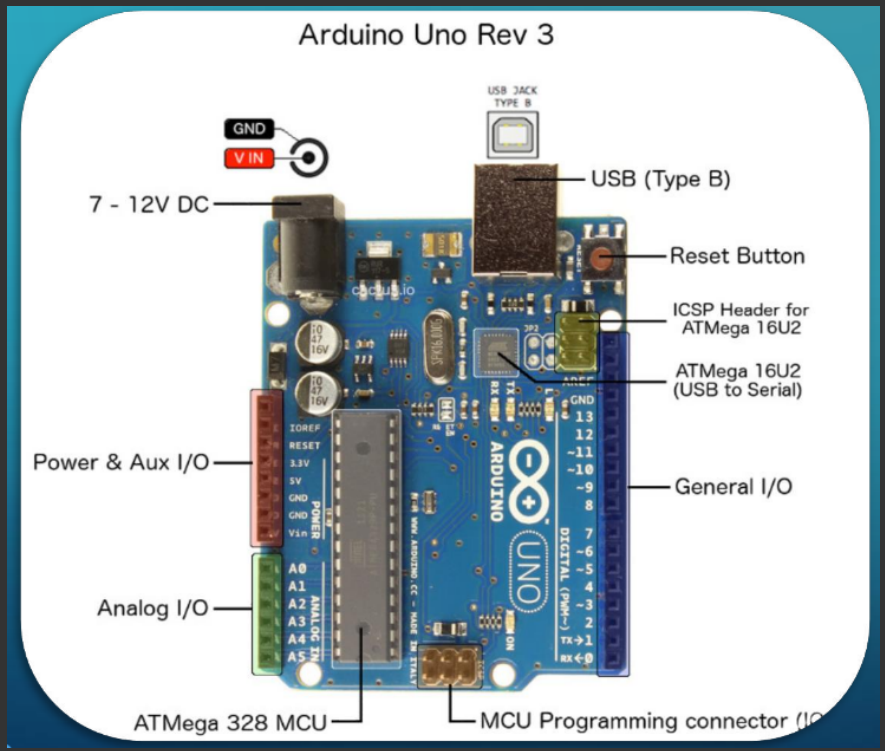
>>> Dagens program

* I dag skal vi se på programmering af microcontrollers!
* Arduino er den hurtigste af slagsen at gå til!
* Vi skal have installeret Arduino IDE
* Lære om Arduino funktioner:
  * pinMode(), delay(), digitalRead() mfl.
* Hvordan vi bruger det sammen med vores nuværende C-viden og kobler det på omverdenen!

>>> Hvad er Arduino?

* Arduino er et forsøg på at gøre microcontrollers lette at programmere i C.
* Microcontrollers er enheder, der indholder alle delene i en lille computer
* Bygget specielt til at styre anden elektronik vha. software
* Arduinoen giver dette inteface gennem sine "pins" som vist:





>>> Installation af Arduino IDE

* Gå ind på https://arduino.cc/download/
* Vælg versionen til dit OS, og tryk "Just download", når de beder om donationer
* Kør installeren og sig ja til alle drivers!

>>> Introduktion til Arduino-funktionerne

* Arduino IDE indeholder faktisk blot en skjult C compiler
* Den skriver C "lidt om" for at gøre Microcontrollere lettere for nybegyndere
* Det hele gøres vha. "arduino.h" som er automatisk include'd
* en af de vigtigste forskelle er at den fjerner main(), og erstatter den med dette:

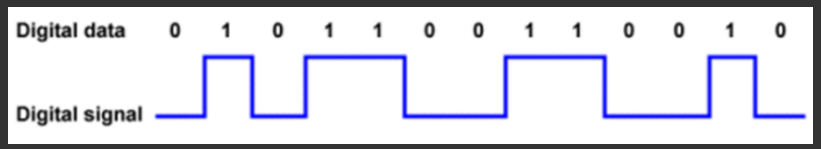
```
1        void setup(){
2           // Kode der kører en gang i starten af Arduino
                programmet
3        }
4        void loop(){
5           // Kode der kører i ring resten af tiden.
6        }
```

* Det betyder i virkeligheden bare:

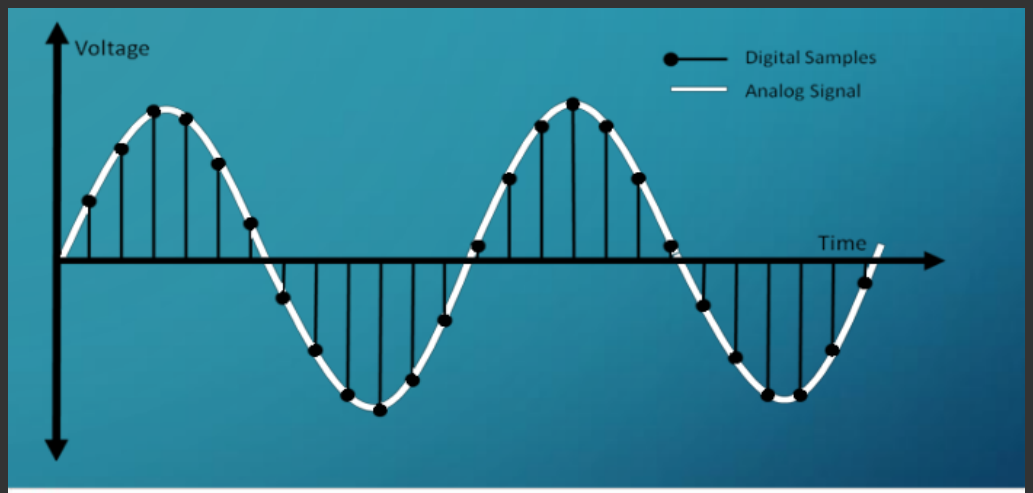
```
1         int main(void){
2            // Koder der kører i starten af programmet
3
4            while(1){
5               // Kode der kører i ring resten af tiden.
6            }
7            return 0;
8         }
```

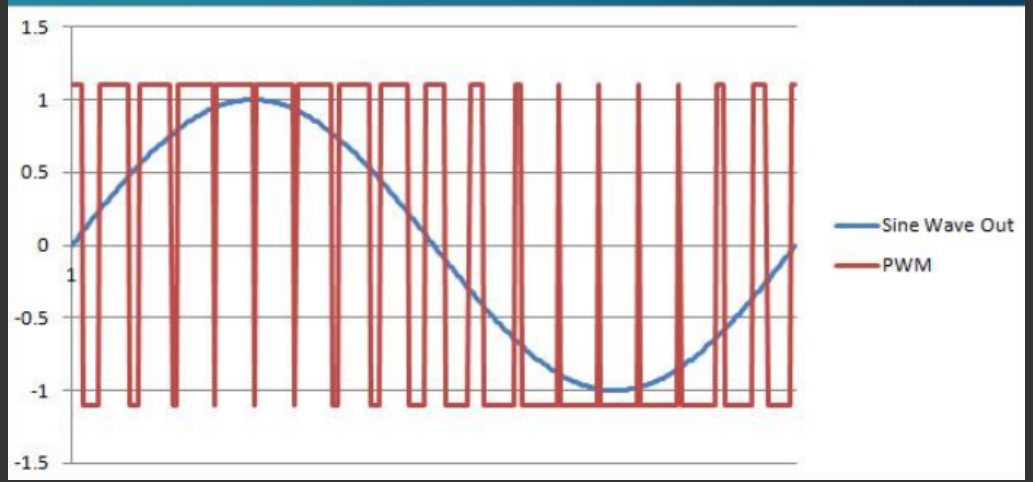
>>> Introduktion til Arduino-funktionerne

* Arduino behandler typisk to former af fysisk data:
* Digital:



* Analog:





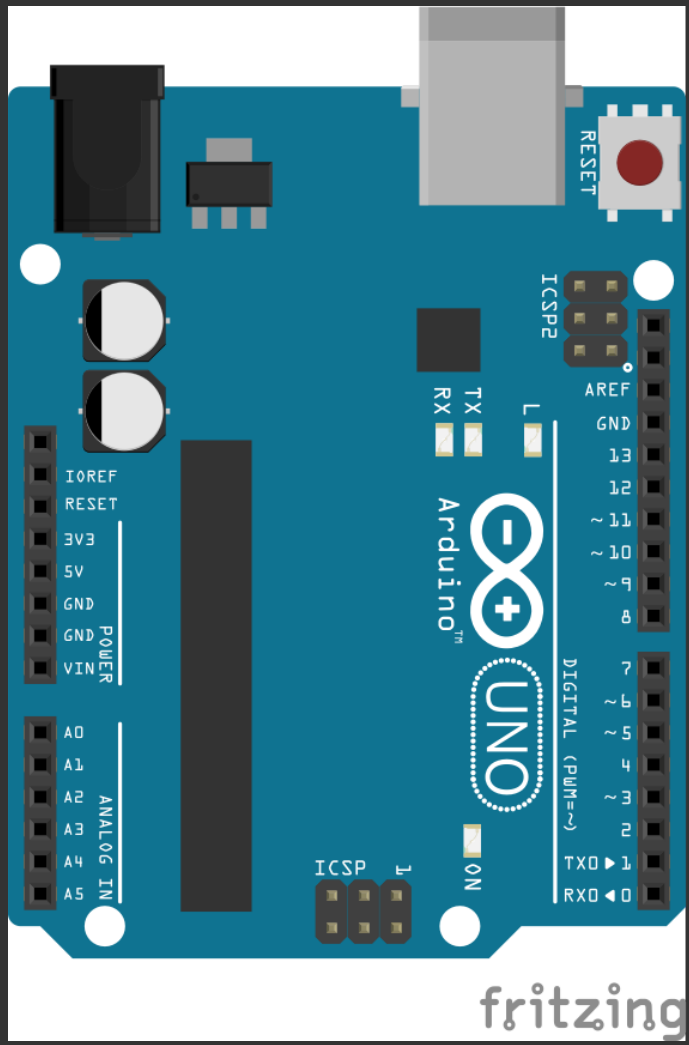
>>> Introduktion til Arduino-funktionerne

* En kort oversigt over nogle af Arduinos vigtigste funktioner:

```
1       pinMode(pin, mode); // Mode kan være enten INPUT, eller
            OUTPUT i det fleste tilfælde,  men også
            INPUT_PULLUP, som er god til trykknapper.
2       digitalRead(pin); // Læser digital værdi  true eller
            false, 1, 0, HIGH, LOW, fra pin.
3       digitalWrite(pin, value); // Value kan gives som true,
            false, 1, 0, eller HIGH og LOW.
4       analogRead(analogPin); // Læser en værdi fra 0-1023 fra
            en analog pin A0-6
5       analogWrite(pwmPin, value); // Skriver et PWM signal til
            en PWM pin (semi analogt signal)
6       delay(milliseconds); // Bremser processoren i en
            tidsperiode, for at vente med næste handling.
```

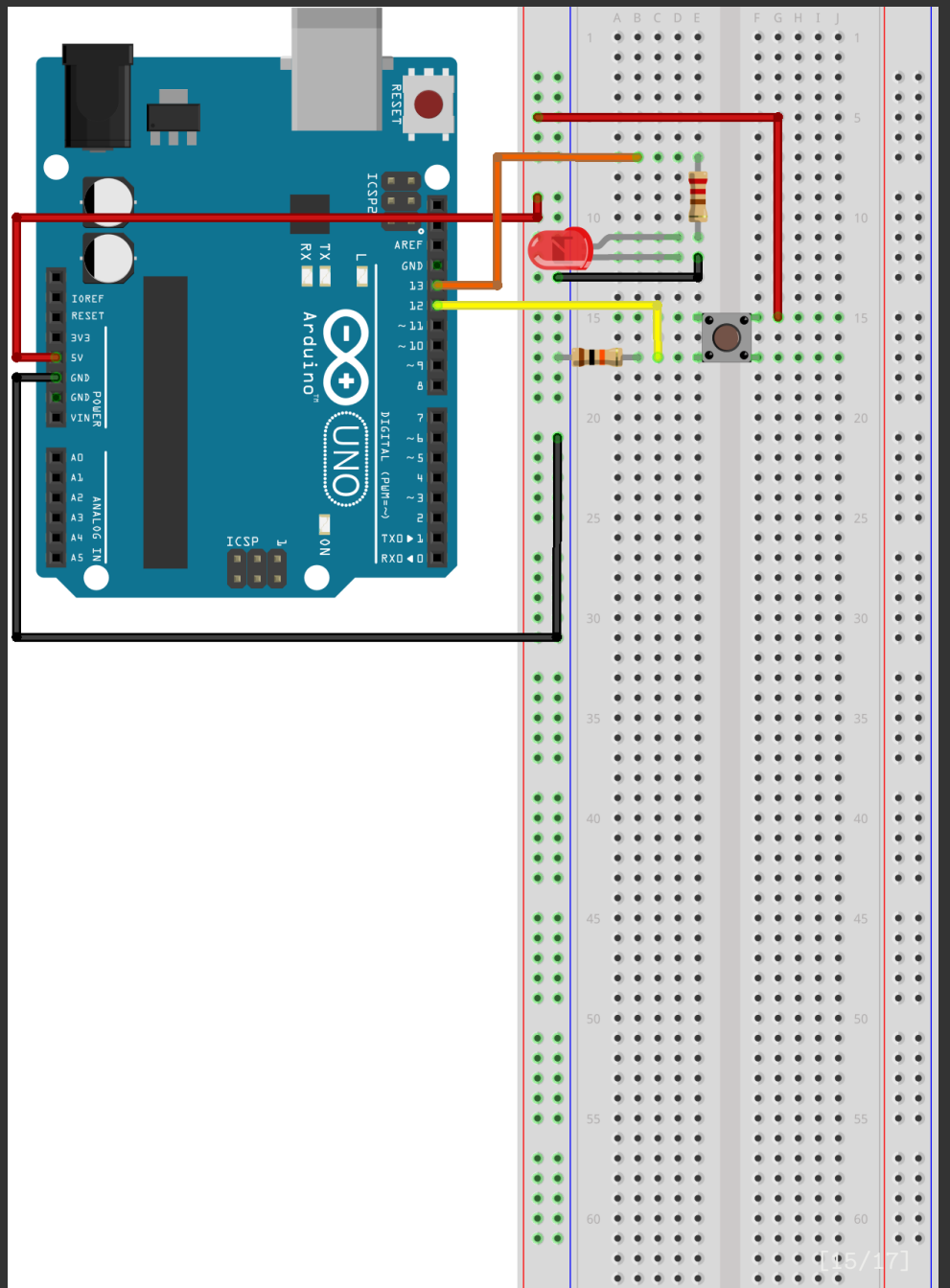
>>> Blinky

* Blinky er Arduinos "Hello World"
* Vi skal kun bruge Arduinoen selv
* Og et kabel til at uploade sketchen!

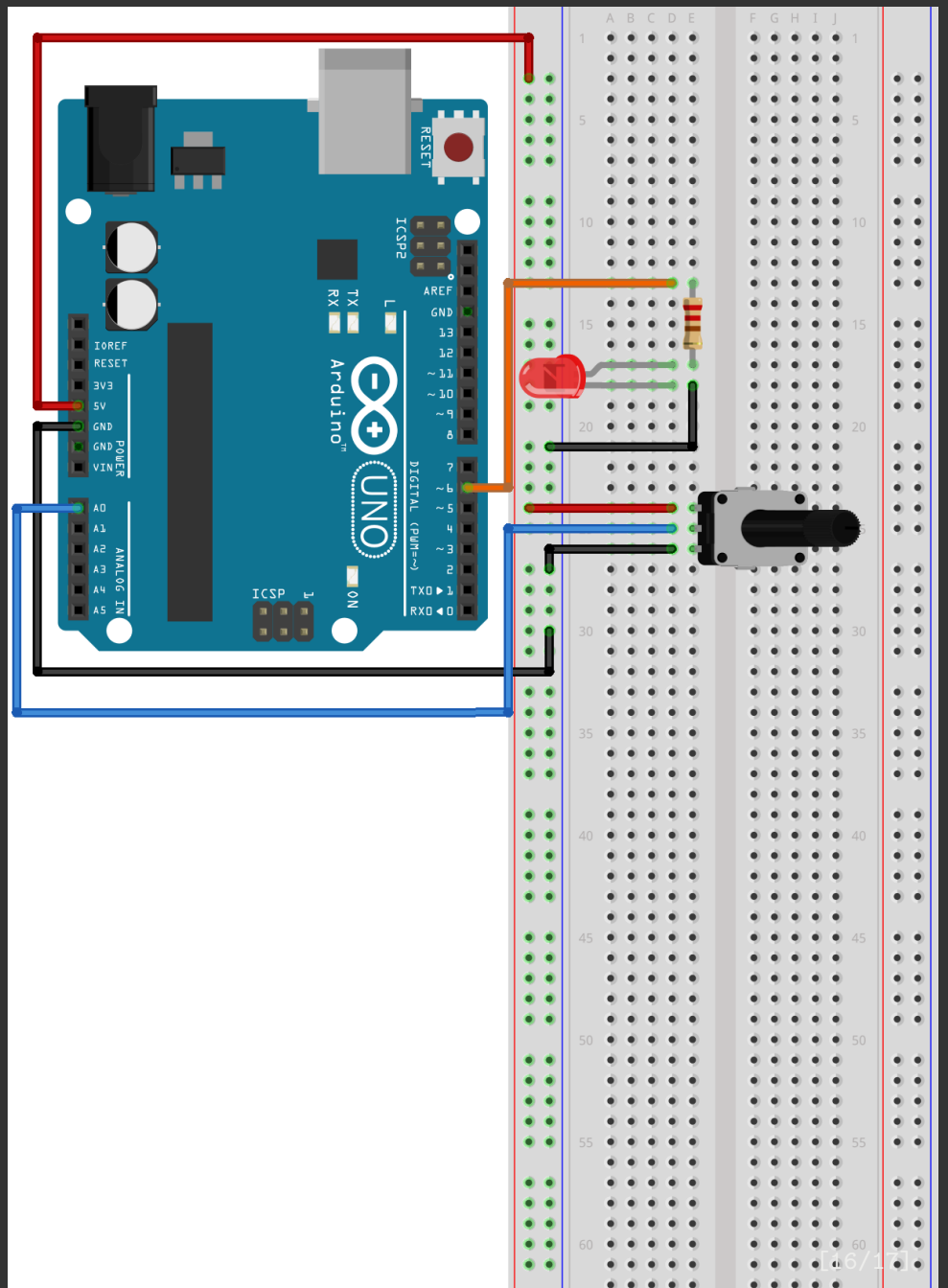


>>> Input/Output

* Nu kobler vi selv en LED på
* Vi tilføjer også en trykknap med pulldown
  * Betyder at den som udgangspunkt læses som false/LOW



* Vi kigger også lidt på den analoge måde at gøre det
* I stedet for en trykknap bruger vi nu et potentiometer
    * Den kan ændre den analoge spænding henover sig fra 0-5V
* Vi styre LED'ens spænding med den og en PWM udgang!



>>> Kreative Opgaver

* Nu kan I starten af Arduino
* Opgaverne i dag kommer fra min Arduino Workshop
* Der ligger kreative opgaver tilgængelige, prøv at koble jeres udvidede C på:
  * ../Del_4/Exercises/C_exercises_4_dansk.pdf
* Der er hjælp at hente her på workshoppen
* God arbejdslyst! - Happy Hacking!