

**Links til emner i dokumentet:**

Brug [Notepad++ som ekstern editor](#), Se [Alternative Editorer](#),

Kopier farvet kode til Word med [Notepad++](#). Online [Code Beautifier](#) muligheder:

Brug [Tabs i Arduino IDE](#)

[Intern Pullups på inputpins](#), [Tilføj flere I/O](#),

[Deling af kode](#)

Ændre [Opsætning af IDE](#), [Linjenumre og foldning af kodestruktur](#)

Thinkercad: [Diagramtegning m. fumleboard](#)

[Simulering](#)

Frigør [lidt RAM](#), [Printudlæg til Shields](#),

**Brug af Notepad++ som ekstern editor**

Den editor, IDE, Integrated Development Environment, der er i Arduino softwaren, kunne let have været meget bedre. Derfor kan det være interessant at bruge en anden editor, fx Notepad++ ( også kaldet NPP ) sammen med Arduino IDE'en.

Notepad udmærker sig ved, at have en meget bedre editor til kildetekst, men den har ingen compiler og upload - funktioner.

Men Arduino IDE kan indstilles til brug af ekstern editor, og kun fungere som kode-verificering og upload. !

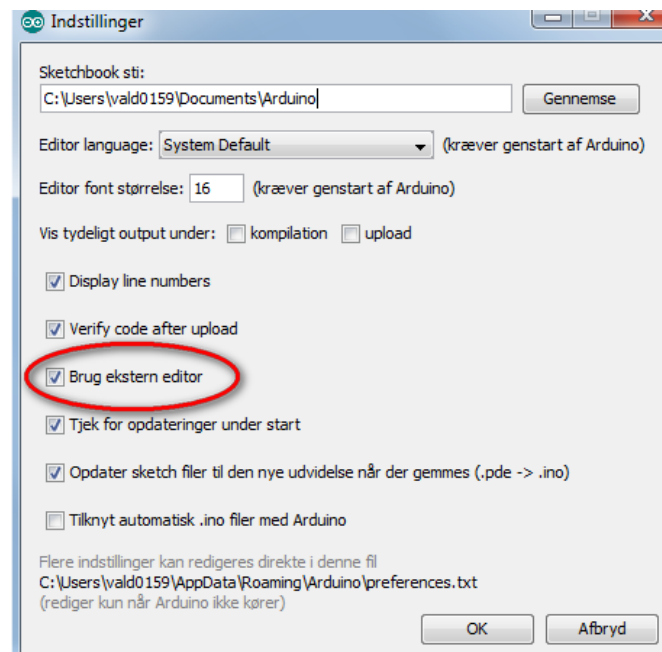
**Indstil Arduino IDE til at bruge extern editor**

Vælg Fil / Indstillinger

Sæt flueben i ” Brug ekstern editor ” og OK.

Herefter vil programmet blive gråtonet.

Når en kildetekst herefter er editert i NPP og gemt, opdateres teksten i Arduino IDE, hver gang der trykkes på Verificer eller Upload



### Hvordan bruges de to programmer sammen?

Den kildetekst, der arbejdes på, skal åbnes i Arduino IDE.

Arduino IDE skal indstilles til brug ekstern editor.

Kildeteksten skal åbnes og editeres i NPP. Eventuelt ved at browse til filen, højreklik og vælg at åbne med NPP.

Efter editering gemmes filen i NPP

Swap til Arduino IDE og tryk Verificer / eller upload.

### Opsætning af Notepad++

Notepad++ er ”blot” en teksteditor beregnet til kode. Men den har en del funktioner til forskellige typer programsprog.

Fx kan nævnes:

Farvning af forskellige keywords, Kopiering af linjer ved at Kontrol+trække en linje.

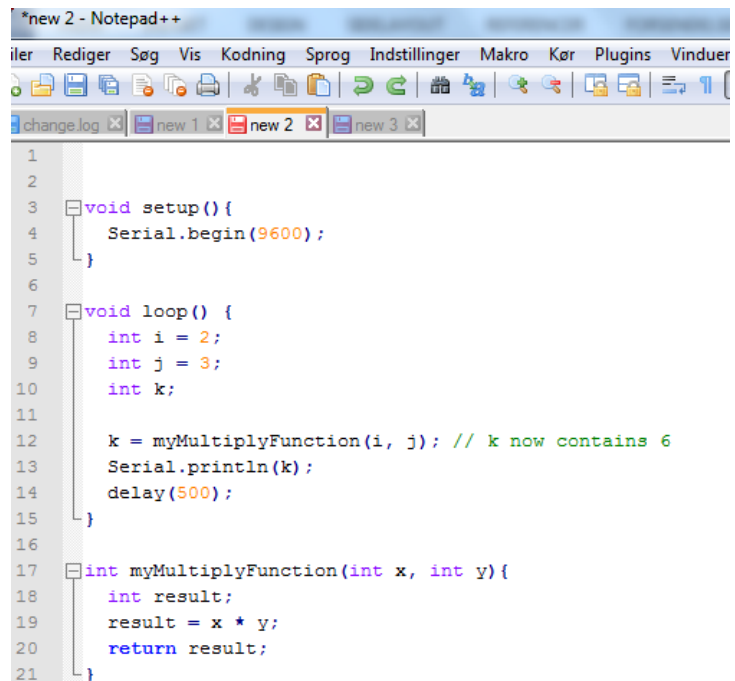
Udfolde / sammenfolde en programstruktur.

Men den eksterne editor kan ikke stå alene, idet den ikke har compiler og upload af program i sig. Derfor skal begge programmer involveres.

[Hent Notepad++](#),

Programmet er Freeware!

Et skærbillede af NPP editoren.



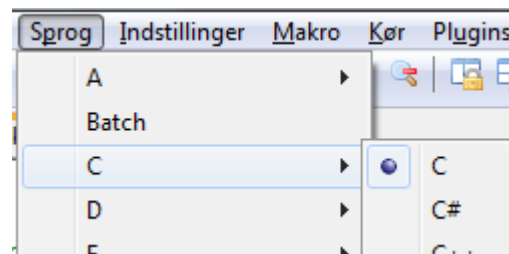
```
1
2
3 void setup() {
4     Serial.begin(9600);
5 }
6
7 void loop() {
8     int i = 2;
9     int j = 3;
10    int k;
11
12    k = myMultiplyFunction(i, j); // k now contains 6
13    Serial.println(k);
14    delay(500);
15 }
16
17 int myMultiplyFunction(int x, int y) {
18     int result;
19     result = x * y;
20     return result;
21 }
```

Når NPP er startet, skal det opsættes til brug sammen med det softwaresprog, der bruges i Arduino-verdenen.

NPP kan indstilles til mange forskellige programmeringssprog: Her skal det være C eller C++.

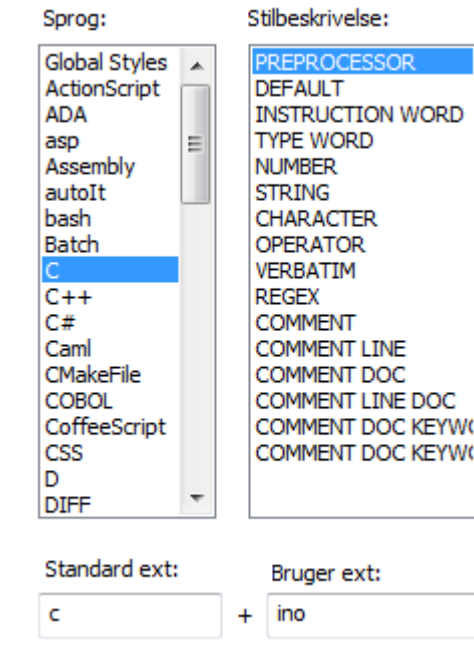
Vælg Sprog, → C

Måske C++ idet Arduino sproget er en afart af C++ med nogle ekstra kommandoer

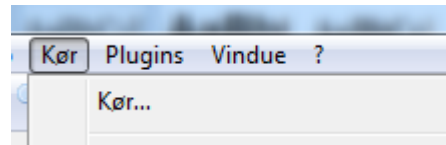


Vælg Indstillinger / stil-konfigurator

( Indstil til at .ino filer kan åbnes ved højreklik med notepad++ )



Vælg "Kør" → Kør og indsæt stien til Arduino.EXE



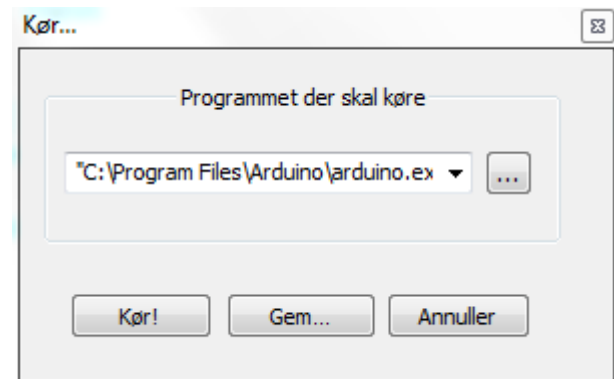
I det vindue, der fremkommer, skal der indtastes hele stien til Arduino Exe-filen.

Brug evt. Browse-funktionen:

Efter stien tastes et mellemrum, og

"\$(FULL\_CURRENT\_PATH)"

Se eksempel herunder:

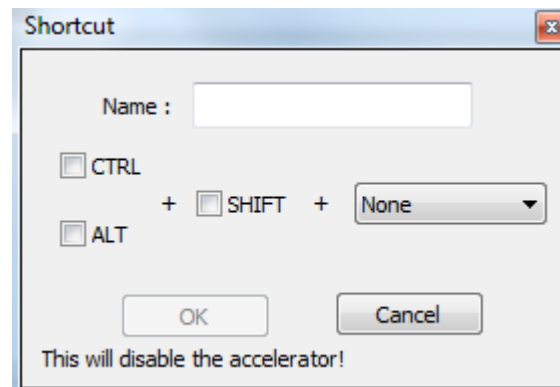


"C:\Program Files\Arduino\arduino.exe" "\$(FULL\_CURRENT\_PATH)"

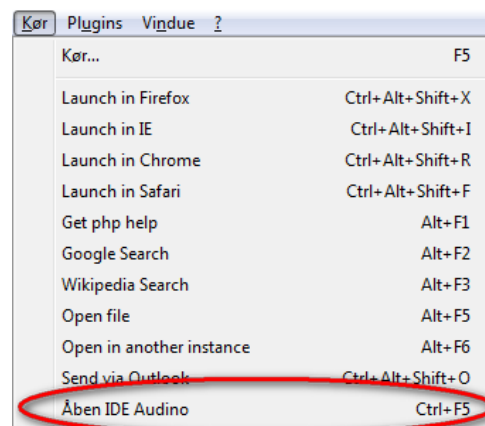
Tryk på Gem

Vælg så et navn, fx “ Åben Ardu\_IDE “

Vælg evt en shortcut, fx CTRL + F5



Det betyder, at man nu kan starte Arduino IDE fra Notepad++.



### Andre Editorer:

Sublimetext, der selv foreslår færdiggørelse af linjer og variable: <http://www.sublimetext.com/3>  
( der er også en Mac, og en linux-version ) Se: <http://playground.arduino.cc/Code/Eclipse>

For alternative editor, der har indbygget Compiler og upload: søg efter MariaMole

Editor for Apple ??? : TextWrangler ( free )

Arduino har en hjemmeside om alternative editorer: <http://playground.arduino.cc/Main/DevelopmentTools>

Alternativ editor: <http://programino.com/ide-for-arduino.html>

Citat:

“I do recommend getting away from the IDE quickly though for writing code--just use it for compilation. Set the IDE preferences to use "External Editor." Then use a professional source code editor/IDE such as [Sublime Text 3](#), [Atom](#), [Visual Studio Code](#), or [Eclipse CDT](#) to write your code. You can then click back to the Arduino IDE to compile and upload. If you go with

Eclipse, there are ways to do that in Eclipse (see the other answers here, and see the [Arduino playground Eclipse article](#) here too) so consider using those techniques too.”

Kilde: <https://arduino.stackexchange.com/questions/816/c-vs-the-arduino-language>

### **Kopier kode til Word med farvning af keywords: Code beautifier**

Der er flere muligheder for at kopiere en kildetekst til Word, og få farverne med:

Arduino IDE 2.0 beta kan bruges direkte vha. C&P.

**Notepad++** kan bruges til at farvelægge kode og eksportere den til Word.

Men der skal først installeres et plugin.

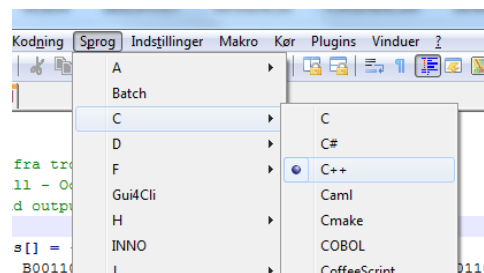
Hent fx en Zip-fil her: [https://github.com/chcg/NPP\\_ExportPlugin/releases](https://github.com/chcg/NPP_ExportPlugin/releases)

Vælg om der skal hentes 64 bit eller 32 bit ( x86 )

Placer filen nppexport.dll i c:/Programmer/Notepad++/plugins, og genstart Notepad++.

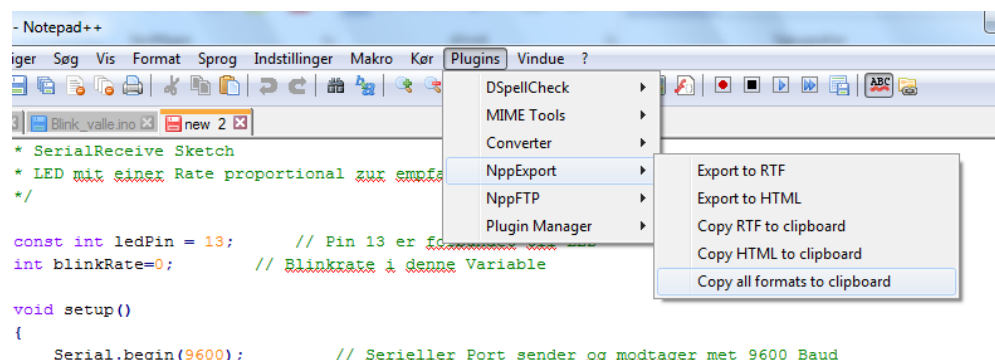
Paste den kode, der skal farvelægges i et Notepad++ vindue

Indstil Notepad++ til fx at vise C-syntax:



Vælg Plugins /  
NppExport / Copy  
all formats to  
clipboard.

Husk at indstille  
sproget i NP++ til  
C++.



### Online Code Beautifier

Her er nogle links til hjemmesider, der burde kunne hjælpe. Find selv ud af hvilken, der er OK

<http://hilite.me/>

<https://tohtml.com/>

<http://www.chami.com/colorizer/>

<http://markup.su/highlighter/>

<http://syntaxhighlight.in/>

<http://highlight.hohli.com/>

<http://www.planetb.ca/syntax-highlight-word>

<http://codepad.org/>

<http://www.chami.com/colorizer/>

<http://www.planetb.ca/2008/11/syntax-highlight-code-in-word-documents/>

### Tabs:

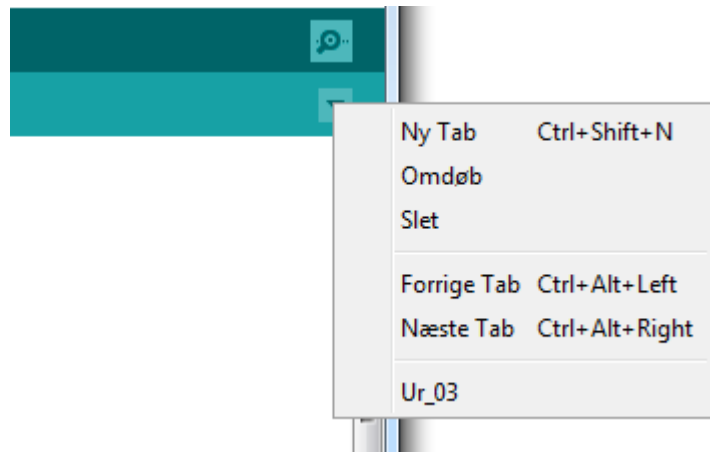
Et Arduino program ( sketch ) skal ligge i en mappe, der hedder det samme som filen.

Hvis Kildeteksten bliver for lang, kan man med fordel placere dele af programmet, fx subrutiner i selvstændige tabs.

Men al definering af pinnumre og variable, void Setup() og void loop() skal ske i hovedfilen.

Klik på den lille ” Pil ned”

Navngiv en tab, - uden extension, og flyt kodedele hertil.



Hvis man starter Tab-navnet som vist her, kan man let organisere Tab-rækkefølgen.

Kode i Tabs compiles som om den ligger i hovedtab.

( Men i nogle kilder fortælles, at tabs compiles i Alfabetisk rækkefølge!! )

### Internal Pull Up på input-pins:

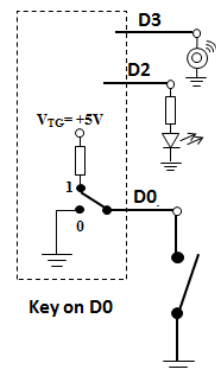
Internt kan man forbinde inputpins til en 20 K Pullup-modstand.

```
DDRD |= B00001100; // D2 og D3 er OUTPUTs
PORTD |= B00000011; // turn on pull-ups for D0 og D1
```

```
pinMode(pin, INPUT); // set pin to input
digitalWrite(pin, HIGH); // turn on pullup resistors
```

```
pinMode(2, INPUT_PULLUP); // Input and pullup!!
```

```
PORTD = 0b00001111; // Pull-ups enabled in the pins
0,1,2 // and 3 and pull-ups disabled in
// pins 4,5,6 and 7
```



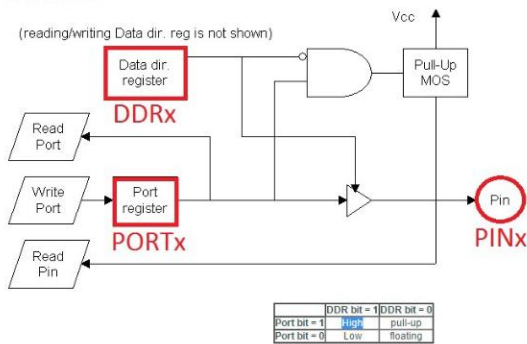
D0 = Arduino-Pin 0  
D2 = Arduino-Pin 2  
D3 = Arduino-Pin 3



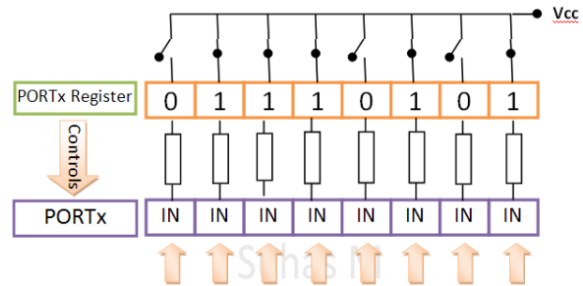
Kilde # <sup>1</sup>

Her et billede af pull-up modstande for pins defineret som input.

AVR I/O Ports:



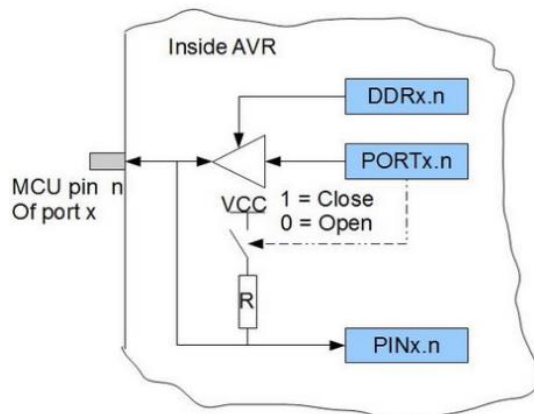
If the pin is configured as input, PORTx manages the internal pull-up



Kilde # <sup>2</sup>

Kilde: # <sup>3</sup>

Her endnu et billede på, hvordan man kontrollerer pins vha. registre.



<http://www.embedds.com/controlling-avr-io-ports-with-avr-gcc/>

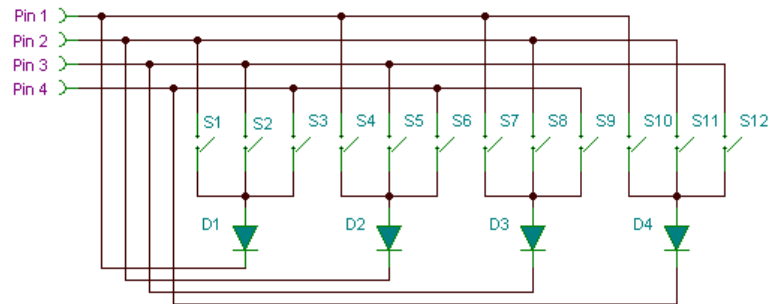
<sup>1</sup> <http://hekilledmywire.wordpress.com/2011/02/23/direct-port-manipulation-using-the-digital-ports-tutorial-part-3/>

<sup>2</sup> <https://iamsuhasm.wordpress.com/tutspoj/avr-gcc-tutorial/>

<sup>3</sup> <https://medium.com/jungletronics/meeting-assembly-hello-world-arduino-blinking-code-330386652309>

**Flere I/O-pins**

Der skal vist ses lidt på Pull  
Down-modstande !!



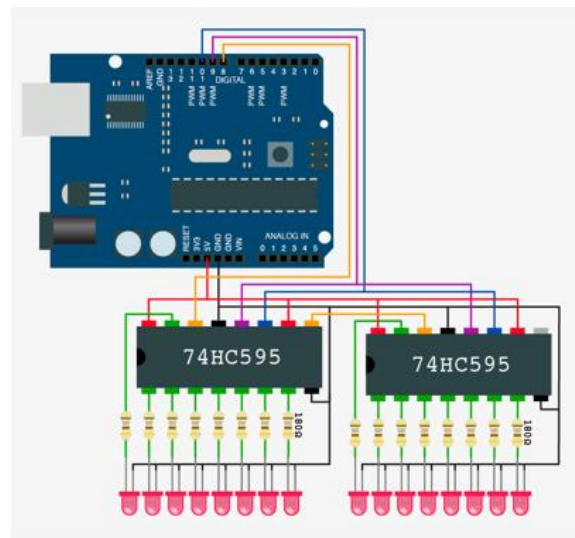
Se Kilde [her](#):

**IC-løsninger:**

Herudover kan man koble ekstra IC-er til uC-en, som kan give flere input / Outputs.

Her vist hvordan 2 stk. 74HC595 skifteregistre  
kan bruges til at give 16 flere udgange!!

Kilde: <http://bildr.org/2011/02/74hc595/>



Se her: <https://www.arduino.cc/en/Tutorial/ShiftOut>

Fx Ic-erne:

8 pins: 74HC595, 164, 165.

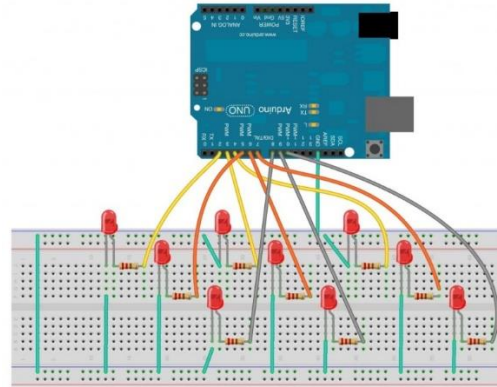
16 pins: MCP23017 ( I2C-bus, 16 bit, 10 kr @ Farnell )

**Diagramtegning: ”Sort of”, Tegne Arduino i diagrammer:**

Se: <https://circuits.io/> og <http://fritzing.org/home/>

Lavet med Fritzing !!

Det er lidt ”barnagtigt” og kan ikke erstatte et ”rigtigt” diagram tegnet med EasyEDA.



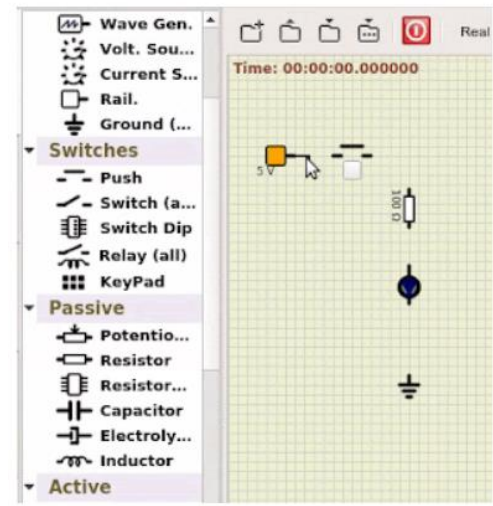
## Thinkercad

Se muligheder på <https://www.tinkercad.com/> Learn / Circuits, - udforsk !!

## Simulering

<p>Hjemmeside hvor man kan opbygge og teste et kredsløb opbygget med en Arduino Uno og et program. Ret smart, testet</p> <p>Kode kan C&amp;P over i systemet !</p> <p>( testet lidt )</p>	<pre> 1 void setup() { 2   pinMode (11,OUTPUT); 3 } 4 5 void loop() { 6   digitalWrite(11, 1); 7   delay(500); 8   digitalWrite(11, 0); 9   delay(500); 10 } 11 </pre>
---	--

<https://wokwi.com/projects/new/arduino-uno>

	<p><a href="https://www.simulide.com/p/home.html">https://www.simulide.com/p/home.html</a></p> <p>Ikke testet !</p>
---	---

### Ændring af IDE-opsætning:

Obs: Luk først programmet:

Tabulator-indrykningen er default kun på 2 spaces.

Dette kan ændres i filen C://Programmer/Arduino/lib / preferences.txt ()

Brug Wordpad i admin-mode!

Find “editor.tabs.size=”, og ændre 2-tallet til 4 og restart Arduino.

Andre smarte ændringsmuligheder:

editor.window.height.default=800

editor.window.width.default=1024

editor.font=Monospaced,plain,14

console.lines=6

( Font størrelse )

( Flere linier i vinduet i bunden af IDE-en )

### Ændring af farver i Arduino IDE:

Der er nogle muligheder for at ændre på farver mm. i IDE-en. Det sker ved at redigere i opsætningsfilen C://programmer/Arduino/lib/theme/theme.txt

Filen skal åbnes I Wordpad, nok kørt som administrator.

Highlight linjen cursoren står i:	Ændre false til true som flg:
-----------------------------------	-------------------------------

	<pre># highlight for the current line editor.linehighlight.color=#e2e2e2 # highlight for the current line editor.linehighlight=<b>true</b></pre>
Marker linje-afslutning med et "Pi"	<pre># little pooties at the end of lines that show where they finish editor.eolmarkers = <b>true</b> editor.eolmarkers.color = #999999</pre>
Ændre farve på kommentarer. De er default noget utydelige	<pre># TEXT - COMMENTS editor.comment1.style = #0147FA,plain ( farve efter // ) editor.comment2.style = #0147FA,plain ( mellem /* */</pre> <p>Find fx farveskabelon her: <a href="http://www.december.com/html/spec/colorhex.html">http://www.december.com/html/spec/colorhex.html</a></p>

Gem "theme.txt" -filen og genload IDE-en.

### Linjenumre og mulighed for sammenfoldning af kodestruktur

Linjenumre slås til i IDE-en ved at vælge Filer / opsætning ( preferences ) og sætte flueben:

Compiler warnings:

- Display line numbers
- Enable Code Folding
- Verify code after upload
- Use external editor

Find evt. en hel ny theme.txt fil her:

<https://github.com/jeffThompson/DarkArduinoTheme/tree/master/theme>

Se fx: <https://github.com/jeffThompson/DarkArduinoTheme>  
<http://mitat.tuu.fi/?p=117>  
<http://mitat.tuu.fi/?p=117>

**Deling af kode**

På nogle hjemmesider kan man droppe sin kode, hvorefter man får et link, så andre kan tilgå siden.

Fx: <https://codeshare.io/> <https://pastebin.com/>

**Flere tips & Trix**

Free up some RAM

If you're working on a project that needs a lot of memory, you can free up 100 more bytes (10% of the RAM on an ATmega168!) by lessening the serial receive buffer. By default its 128 bytes, which is quite a bit!

Open up hardware/cores/arduino (or cores/arduino) directory, and edit the file named wiring\_serial.c or HardwareSerial.cpp

Near the top is a #define RX\_BUFFER\_SIZE 128, which means 128 bytes are used for the buffer. You can change this to 32 (or even 16!). If you have almost no serial input, make it as low as you'd like as long as its > 0.

fra: <http://learn.adafruit.com/downloads/pdf/arduino-tips-tricks-and-techniques.pdf>

(<http://www.basic4ppc.com/android/forum/threads/arduino-adk-ide-installation.27512/> )

Hold Control key on Upload: To open the Serial Monitor automatically when upload complete.  
?? virker vist ikke !!!

Version 1.0 of the Arduino IDE introduced the F() syntax for storing strings in flash memory rather than RAM. e.g.

```
Serial.println(F("This string will be stored in flash memory"));
```

**Printudlæg til Shield. ( print, der kan sættes direkte ned i stikkene på Arduino-boardet:**

Download program til at tegne diagram og lave printudlæg til Arduino boards:

<http://fritzing.org/welcome/> <http://fritzing.org/>

Lille film om hvordan programmet kan bruges til at tegne diagram og lave et shield.

<http://www.youtube.com/watch?v=eHU-pF5gSnQ>

**Kilder: Se fx.**

<http://sriramiyer.net/blog/2014/02/12/using-notepad-plus-plus-instead-of-the-arduino-ide/>

<http://www.intorobotics.com/alternatives-standard-arduino-ide-one-choose/>

<http://arduino4dev.com/codeblocks/>

[http://www.atmel.com/Microsite/atmel\\_studio6/](http://www.atmel.com/Microsite/atmel_studio6/)

<http://www.arduinoos.com/2011/11/alternate-ide-part-1/>

<http://www.lydiard.plus.com/editing.htm> ( kode til at load ind i NPP for at genkende Ardu  
Keywords )

<http://gadgetsblog.wordpress.com/tag/notepad/>

<http://vpapanik.blogspot.dk/2012/08/arduino-thank-you-and-bye-bye.html>