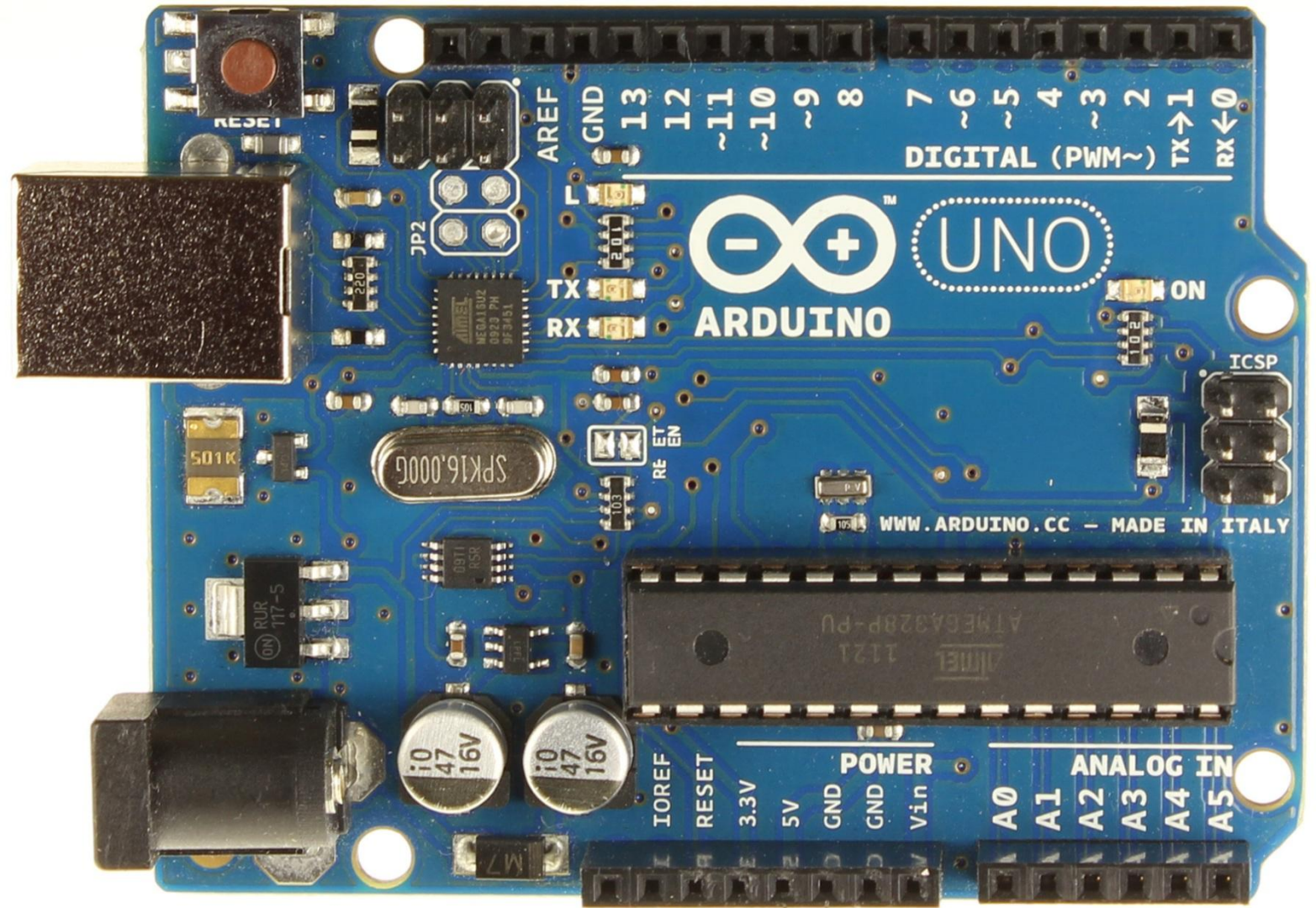
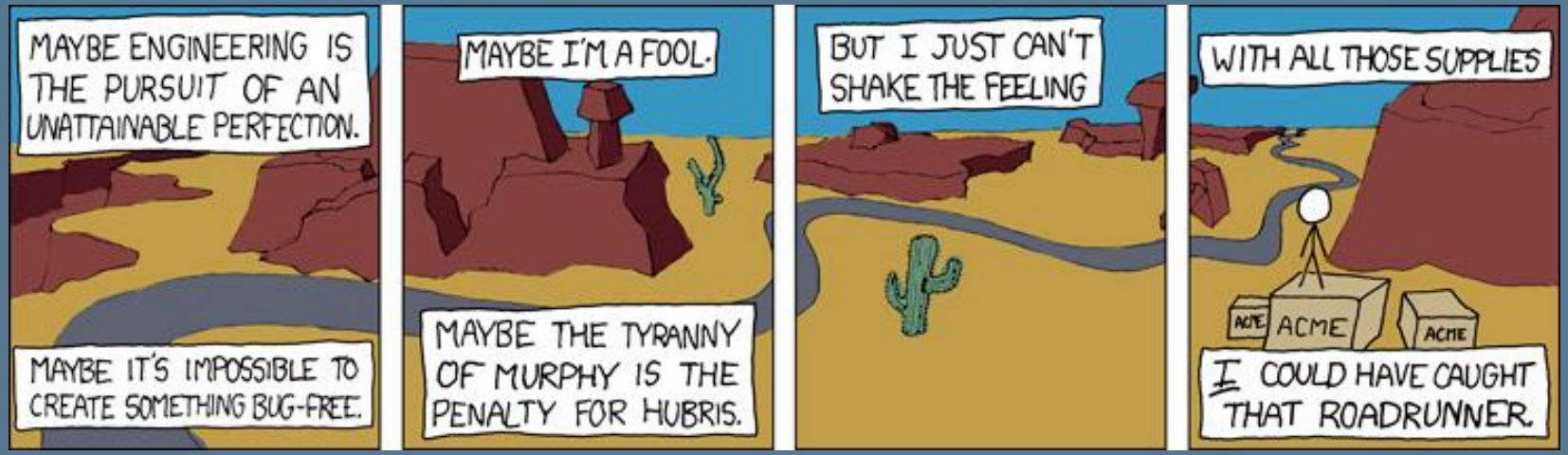


Arduino

bouvet


MAYBE ENGINEERING IS THE PURSUIT OF AN UNATTAINABLE PERFECTION.
MAYBE IT'S IMPOSSIBLE TO CREATE SOMETHING BUG-FREE.
MAYBE I'M A FOOL.
MAYBE THE TYRANNY OF MURPHY IS THE PENALTY FOR HUBRIS.
BUT I JUST CAN'T SHAKE THE FEELING
WITH ALL THOSE SUPPLIES
ACME
ACME
ACME
I COULD HAVE CAUGHT THAT ROADRUNNER.

Agenda

- Introduksjon til Arduino
- Digitalkommunikasjon for n00bs (hvis ikke alle kan forklare forskjellen på TTL, CMOS og RS232 signalering)
- Introduksjon til Arduino IDE, hvordan finne frem
- Arduino «hello world» 2.0 (Blinkende LED av og på, «soft» med PWM)
- Elektronikk 101
- Tolkning av rotary encoder (lage dimmer til LEDen)

Du trenger

Arduino 1.0 http://arduino.cc/en/Main/Software

- Arduino Uno e.l.
- USB A-B kabel

Fritzing http://fritzing.org/download/

GIT+ http://code.google.com/p/msysgit/downloads/list

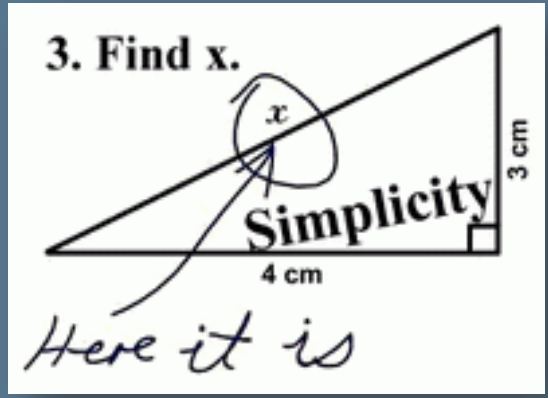
(evt høyt antall ord pr minutt)

GitHub

```
git clone git://github.com/aardalo/Arduino101.git
```

bouvet

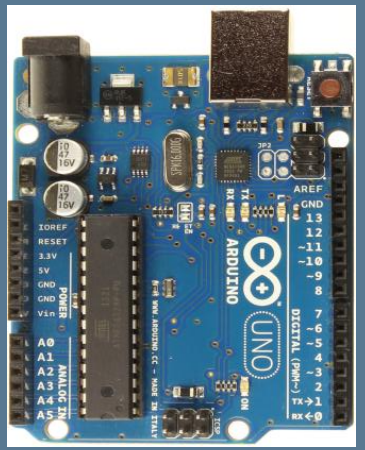
Warning



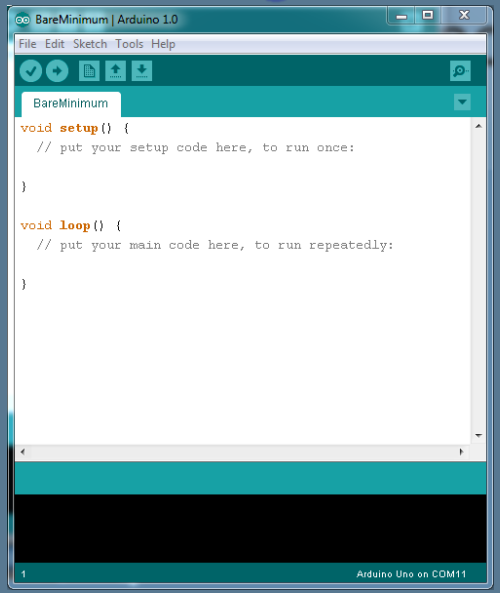
- Overforenkling!!
  - Mange gode kalkulatorer
  - Noen har sannsynligvis gjort det før

Hva er Arduino?

Hardware



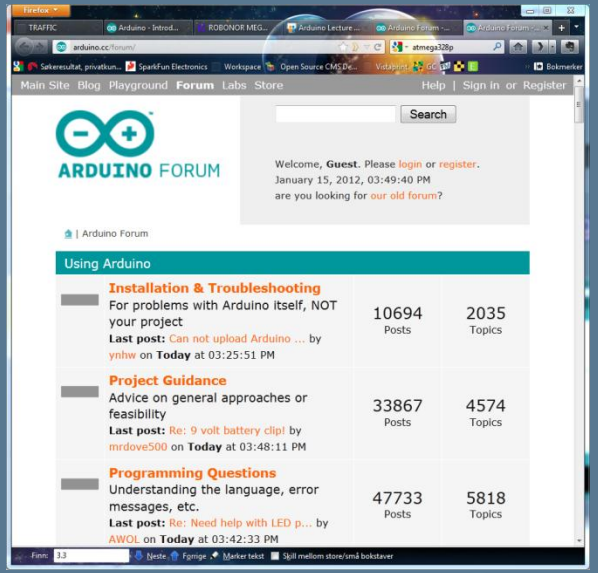
Utviklings-miljø



Community

arduino.cc/forum

arduino.cc/playground

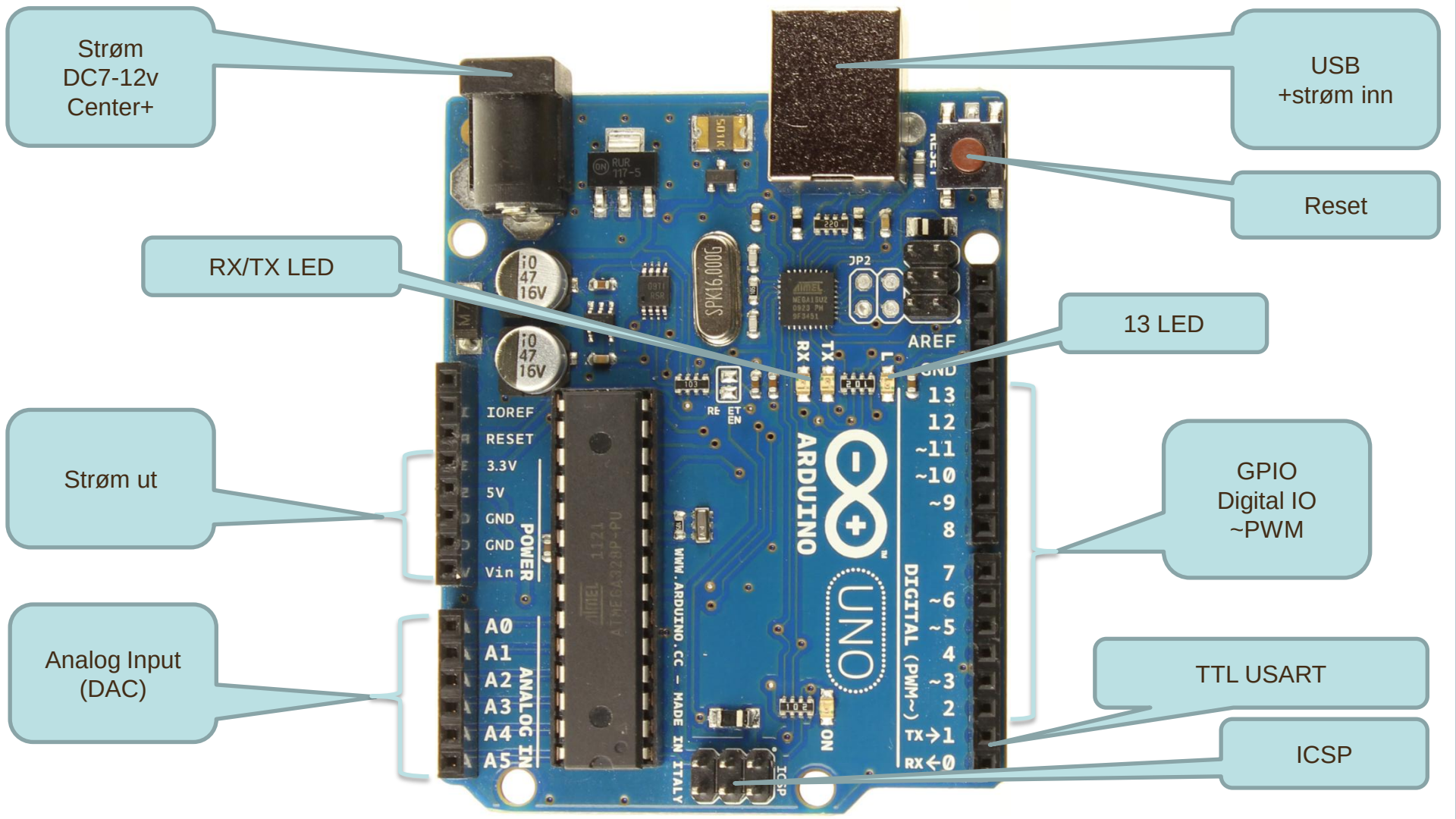


bouvet

Hardware

- Open Source Physical Computing
  - Moduler til ferdige produkter
  - Åpent design til prototyping av masseproduksjon
- AVR ATMega328p
  - 8 bit RISC 16 mHz (1-20mHz)
- Strøm, USB, kommunikasjon, «headers»
- AVR designet i Trondheim av 2 studenter og utviklet/produktifisert av Nordic VLSI så Atmel
  - (Alf og Vegars RISC prosessor … AVR)

bouvet

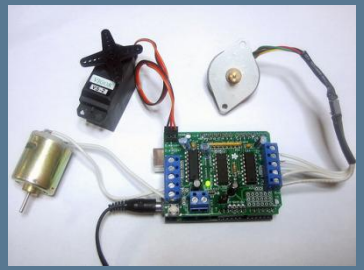

Strøm
DC7-12v
Center+
USB
+strøm inn
Reset
RX/TX LED
13 LED
Strøm ut
GPIO
Digital IO
~PWM
Analog Input
(DAC)
TTL USART
ICSP
IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
DIGITAL (PWM~)
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
AREF
ARDUINO
UNO
WWW.ARDUINO.CC – MADE IN ITALY

Shield – ferdigtenkte utvidelser

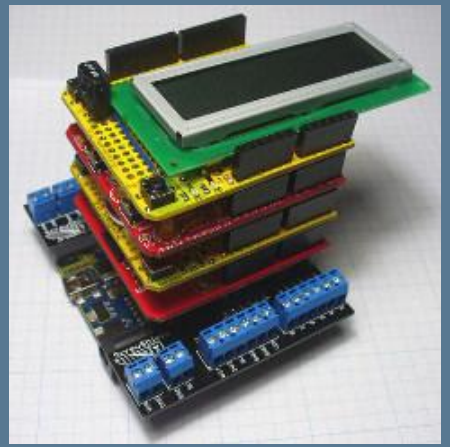
- Plugg inn ny funksjonalitet og repeterer pinnene
- Nødvendig analogteknikk vanligvis ferdig på kortet

Ethernet Shield

Wireless SD Shield



Motor Shield



og 282 andre…

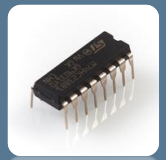
www.shieldlist.org

bouvet

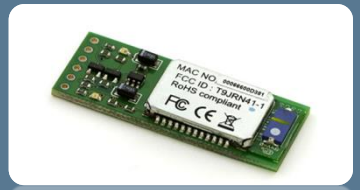
…og komponenter/sensorer

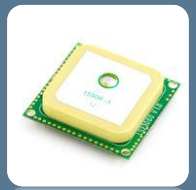
- ICer



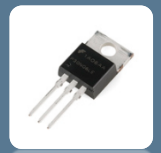


- Breakout



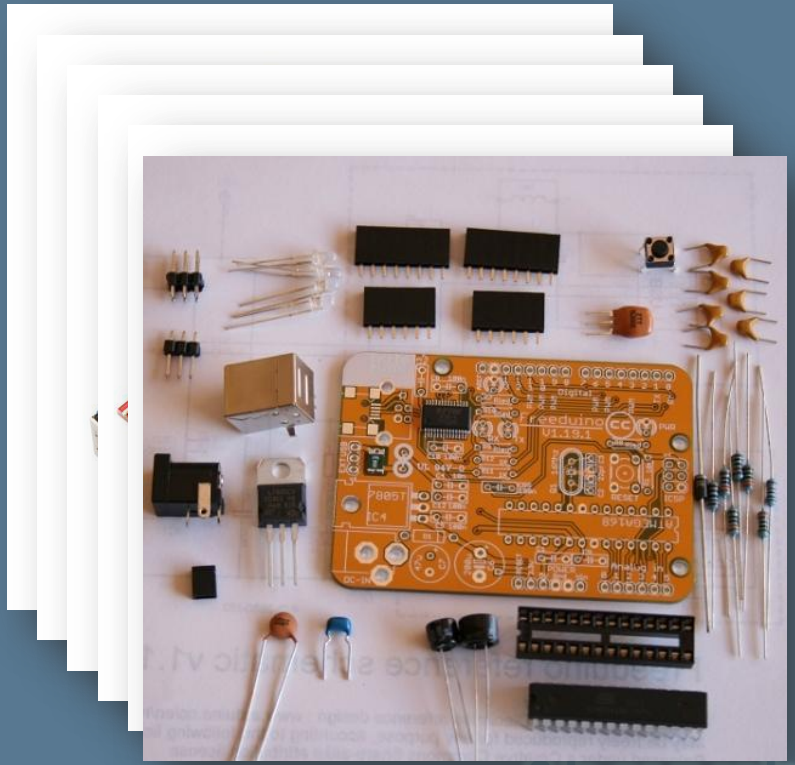
- Passive komponenter





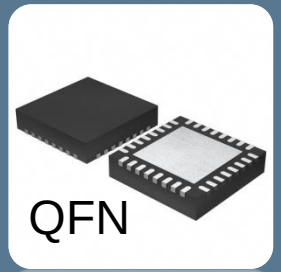
Andre versjoner

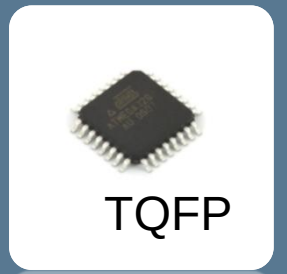
- Pro Micro
- Fio
- Mega
- Pro Ethernet
- Pro 328
- Freeduino

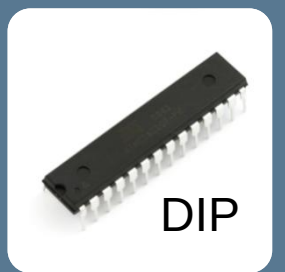


Prosessoren, ATmega328p

- 32kb FLASH (2kb bootloader), 1kb EEPROM, 2kb RAM
  - Mer minne: EEPROM eller SD kort på I2C/SPI
- 32 pin, 23 I/O herav 6 PWM
- 2 SPI, 1 TWI (I2C) og 1 UART
- 8 ADC pinner på 10 bit (0-1023), 15 ksps
- 1 MIPS pr mHz
- 5V og 3.3V
  - 5V tåler 3.3V logikk


QFN


TQFP


DIP

bouvet

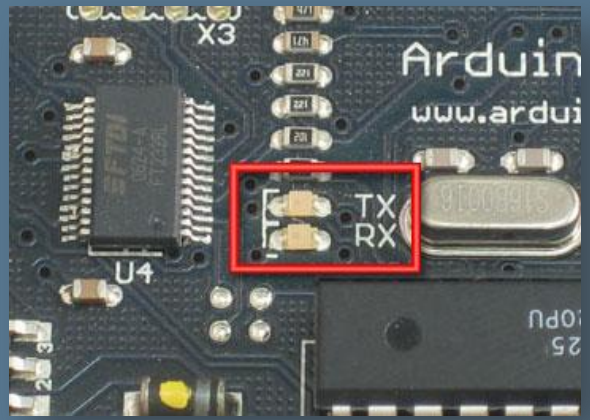
Kommunikasjon

- RS232 +/- 12v + DTR/DSR 9-25 pin
  - Krever MAX232 e.l. IC for å konvertere til TTL
- TTL 0/5v
- CMOS TTL for 0/3.3v
- TWI/I^2C 2 ledere + strøm (Phillips)
- SPI 4 ledere + strøm (Motorola)
- 1-Wire 1 leder + jord (Dallas)

Seriell kommunikasjon

- USB er TTL-liknende – COMx: på Pcen
  - Egen Atmega16U2 IC tar seg av USB
    (Som 368p men mangler ADC, I2C og RTC)
- DEBUGGING
  - Serial.print(x);
- I dybden:

http://en.wikibooks.org/wiki/Serial_Programming



bouvet

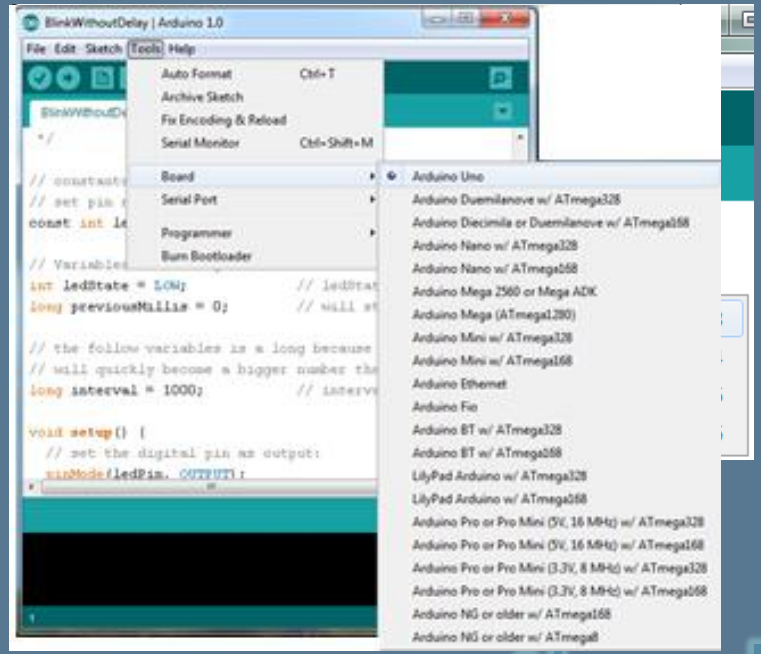
Utviklingsmiljø

- Arduino 1.0
  - Laget i Java
  - Basert på «Wiring»
  - C++, IDE og referanse HW design

Utviklingsmiljø

http://arduino.cc/en/Main/Software

- Windows
  - Device Manager
    .inf fil i Arduino katalogen
- USB A-B kabel



Legg det i:

```
c:\program files (x86)\arduino-1.0\
```

bouvet

Utviklingsmiljøet

- Banalt IDE med forenklet C++
- Et program kalles en «sketch» og har 2 hovedbestanddeler
  - void setup()
  - void loop()
- Ctrl+Shift+F slår opp hjelp
- Ctrl+R compile
- Ctrl+U upload

  (reset først hvis den feiler)

Språket

- WinAVR (*whenever*)
  - C++ uten libstdc++

http://www.nongnu.org/avr-libc/user-manual/index.html

- .cpp og .h filer til inspirasjon og referanse

  C:\Program Files (x86)\arduino-1.0\hardware\arduino\cores\arduino

```
void setup(){
  Serial.begin(115200);
  Serial.println("Hello class...");
}

void loop(){
  Serial.println("Please wait, something fantastic is about to happen!");
}
```

BTW: Hadde ikke tenkt å lære bort C eller C++ i kveld…

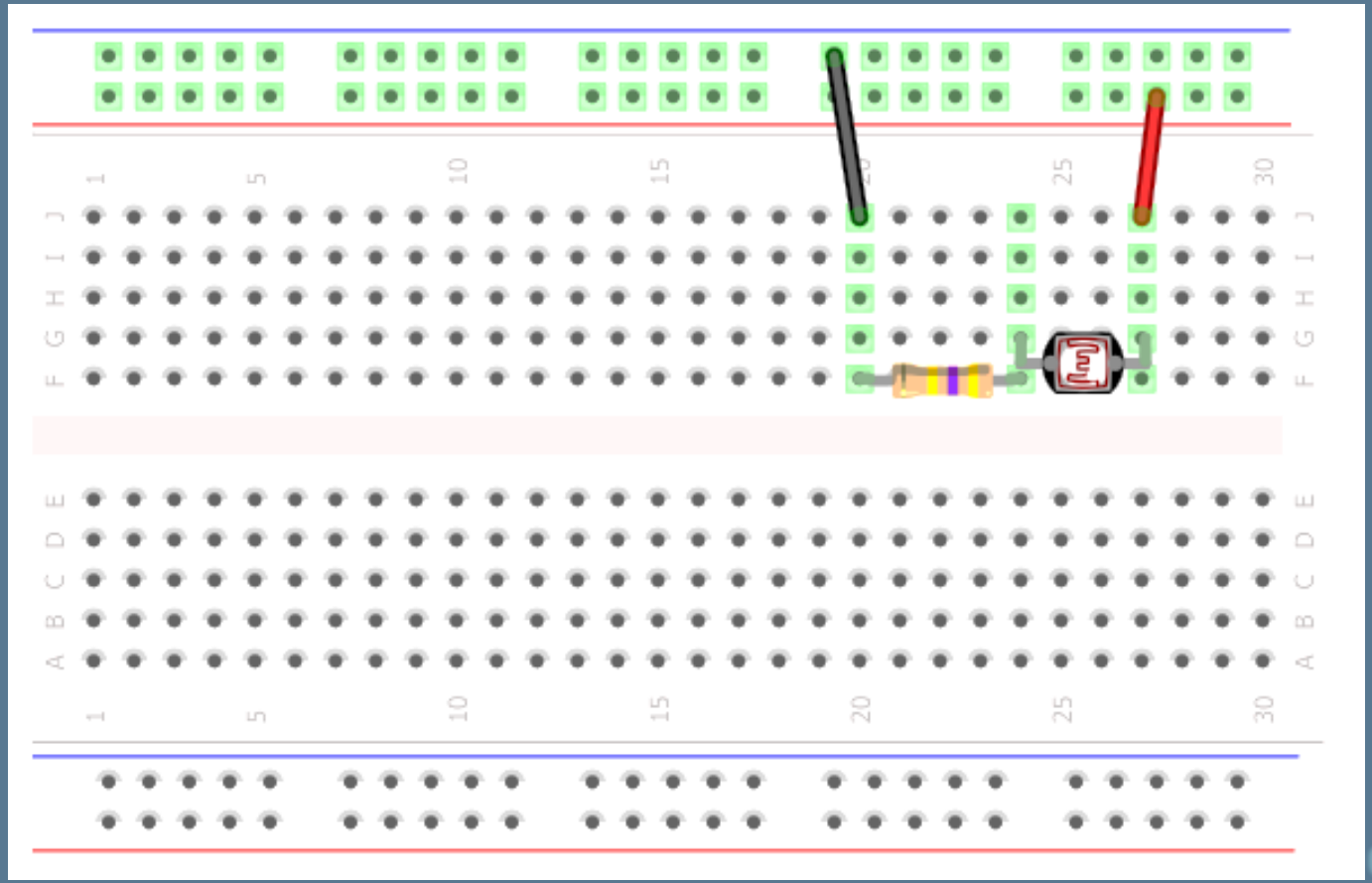
bouvet

Biblioteker – libraries

`C:\Program Files (x86)\arduino-1.0\libraries\...`

lib	funksjon
EEPROM	reading and writing to "permanent" storage
Ethernet	for connecting to the internet using the Arduino Ethernet Shield
Firmata	for communicating with applications on the computer using a standard serial protocol.
LiquidCrystal	for controlling liquid crystal displays (LCDs)
SD	for reading and writing SD cards
Servo	for controlling servo motors
SPI	for communicating with devices using the Serial Peripheral Interface (SPI) Bus
SoftwareSerial	for serial communication on any digital pins
Stepper	for controlling stepper motors
Wire	Two Wire Interface (TWI/I2C) for sending and receiving data over a net of devices or sensors.

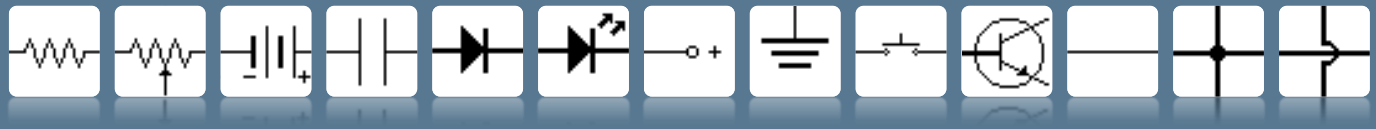
Breadboard



Schematics/Kretsskjema

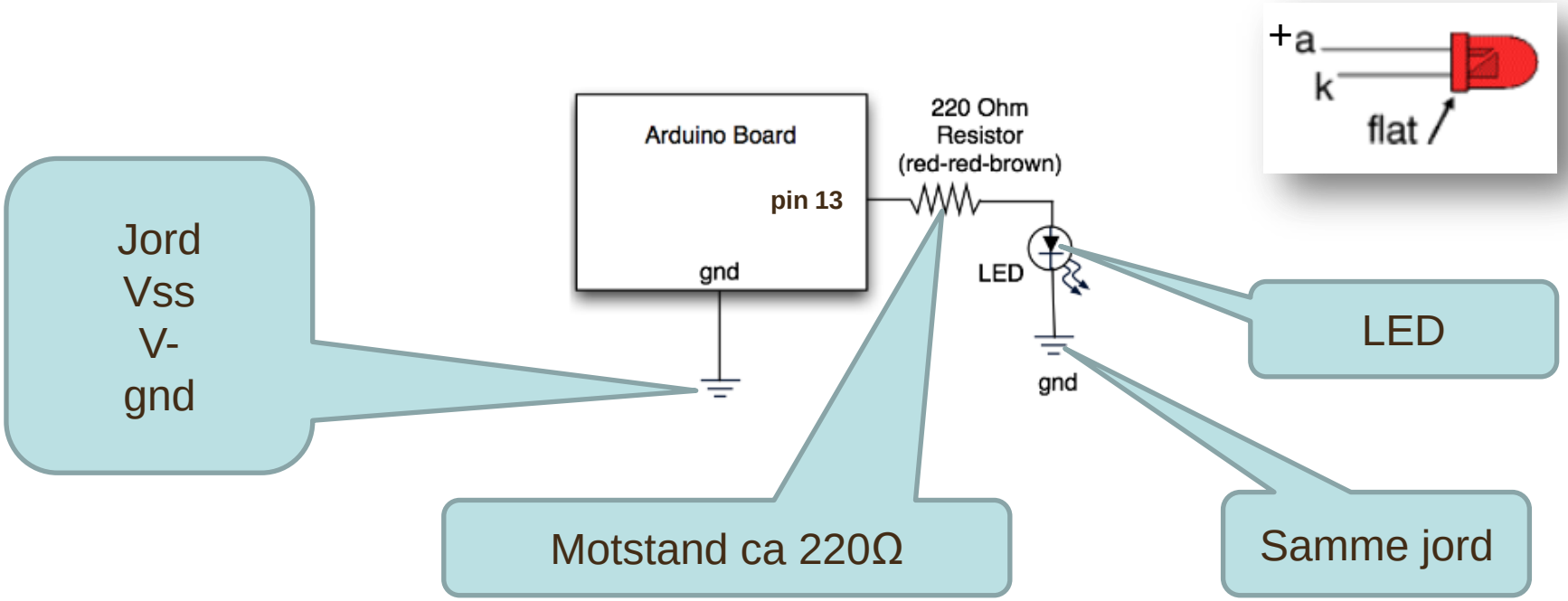
- Viser elektroniske kretser med standardiserte symboler

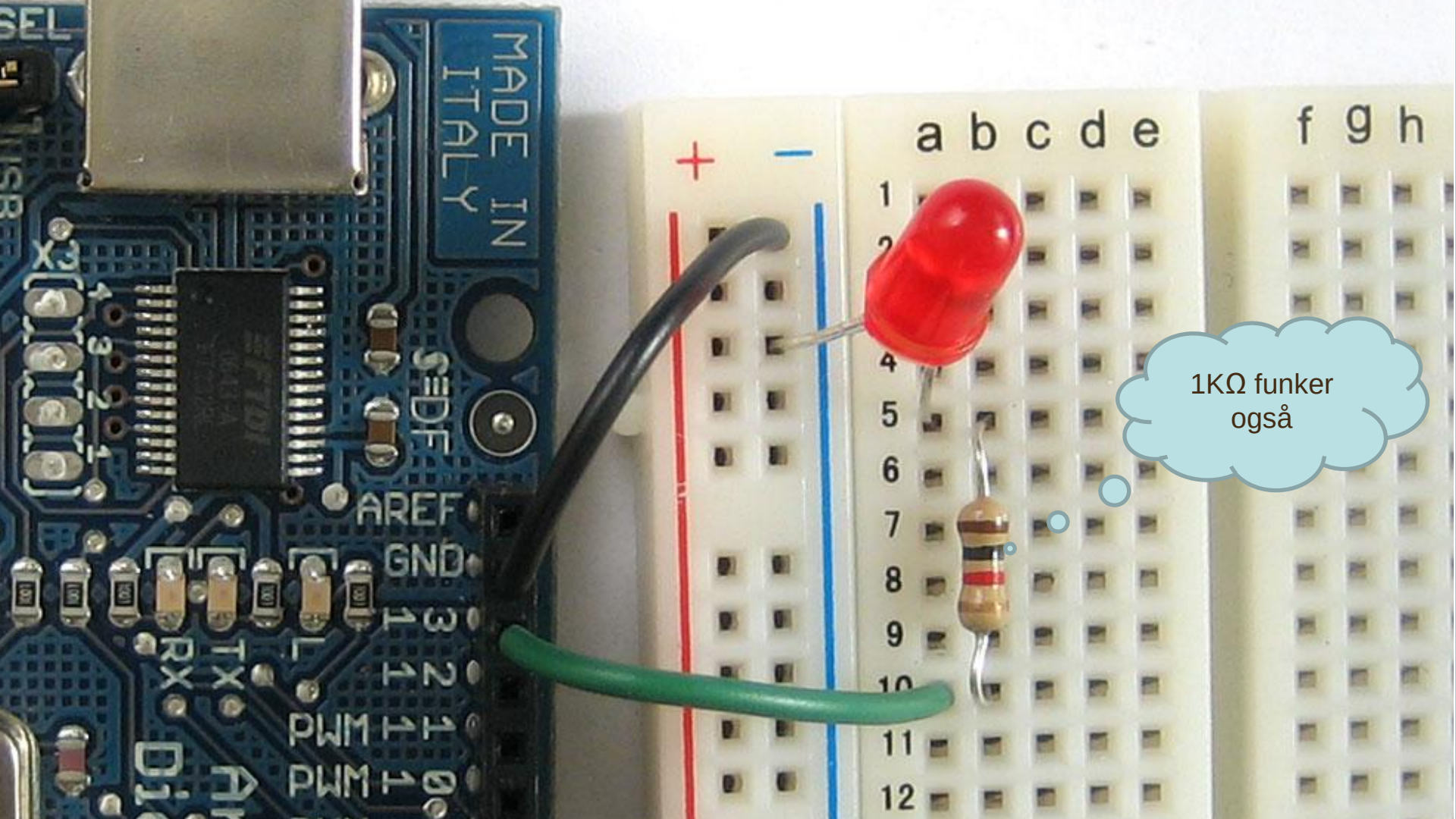
http://library.thinkquest.org/10784/circuit_symbols.html



  - Organiseres fra topp venstre, i samme sekvens som signalene går
  - Eksemplene her er lagd i FRITZING eller lånt på nett

Hello World - HW




MADE IN ITALY
AREF
GND
PWM
PWM
a b c d e
f g h
1
5
6
7
8
9
10
11
12
1KΩ funker også

Hello World - code

```
void setup() {
  pinMode(13, OUTPUT);      // Initialize pin 13 as output
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

Neste versjon

- Bryter
- «myk» av/på av LED

Litt grunnleggende elektronikk

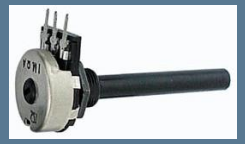
Elektronikk 101 – passive komponenter

- Motstand
- Dioder
- Kondensator
- Transistor
- Ohm
- Toleranse

Elektronikk 101 – motstand

- Fast

- Variabel



- PTC/NTC

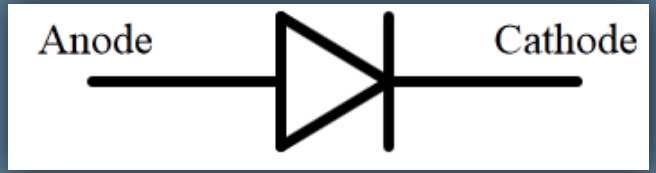
(Thermistor)

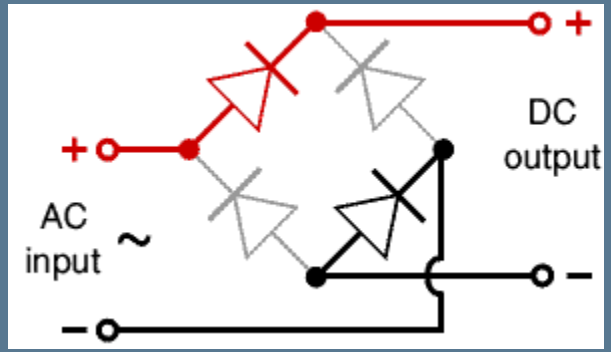
- LDR

(Photoresistor)

Elektronikk 101 – Dioder

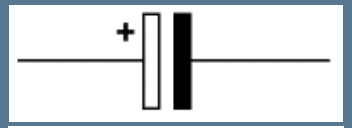
- P-N
- LED
- Zener



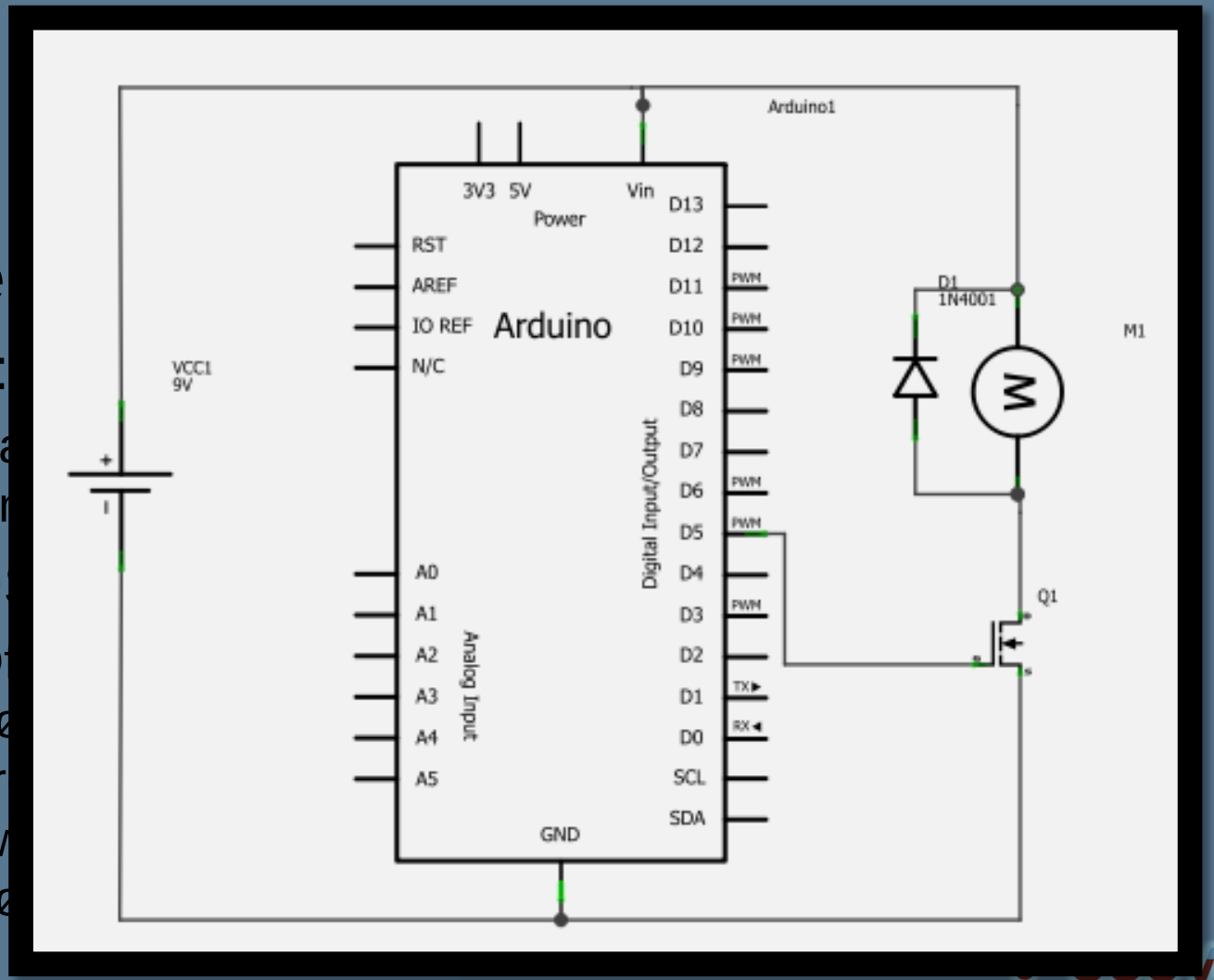


Elektronikk 101 – Kondensator

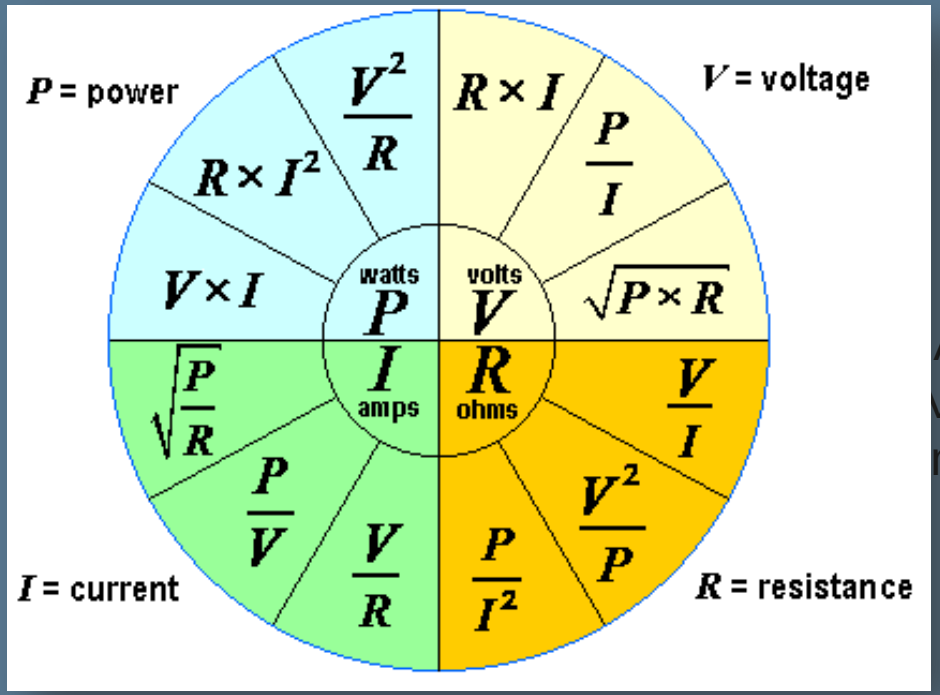
- Lagrer strøm, måles i F
  - Polariserte
  - Upolariserte



- Forste
  - BJT:
    - Ba
      Er
  - (MO
    - Ot
      hø
      kr
      sv
      hø



Elektronikk 101 – OhmΩ



I A, osv.

V V, osv.

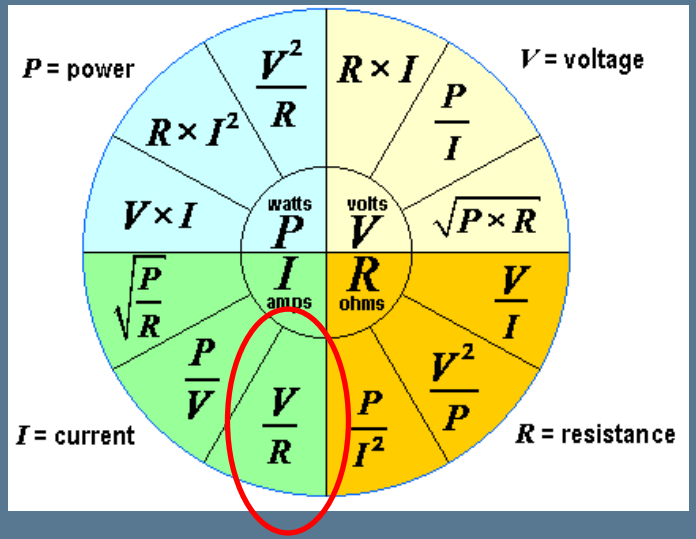
R m, kΩ, osv.

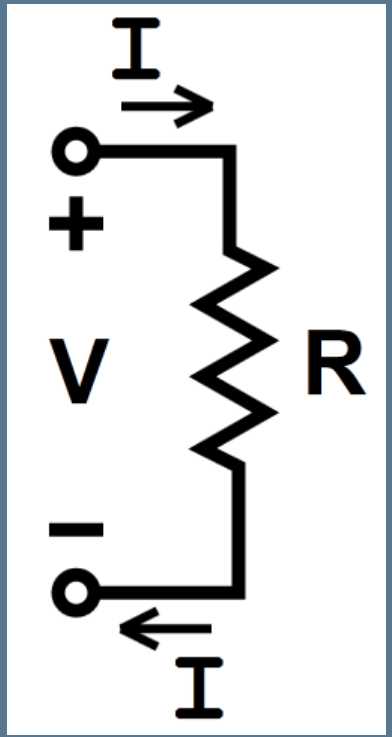
P

HUSK DESIMALPLASSEN

bouvet

Ω – eksempel LED

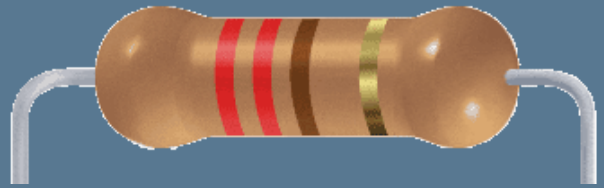




$$\text{resistance}(R) = \frac{\text{power supply voltage}(V_s) - \text{LED voltage drop}(V_f)}{\text{LED current}(I)},$$

Elektronikk 101 - toleranse

- Vanlig toleranse er fra 1% til 10% eller mer
  - 5V regulator LM7805, 4% gir 4.8-5.2V
  - 220Ω motstand (Rød-Rød-Brun-Gull, 5%)
    - 209Ω – 231Ω



- Kjøp et bra multimeter – mål!
- Les data-arkene! www.alldatasheet.com

bouvet

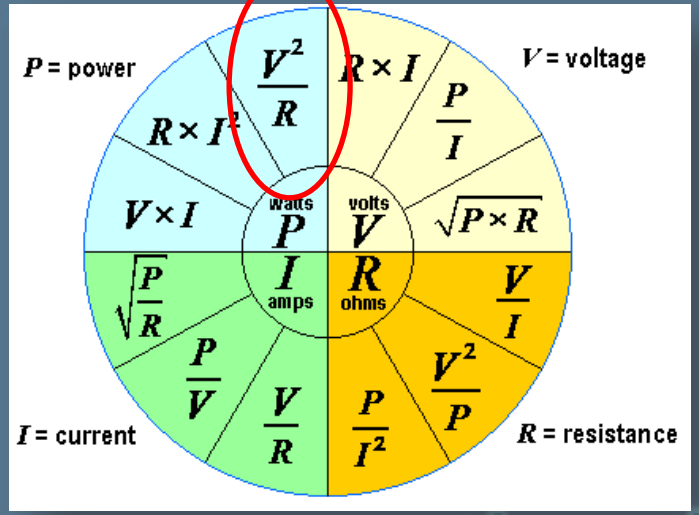
Elektronikk 101 - motstand

- Komponenter må tåle effekten de utsettes for
- P, effekt (watt) på ting

  4.8^2 / 209 = 0.1102 (watt)

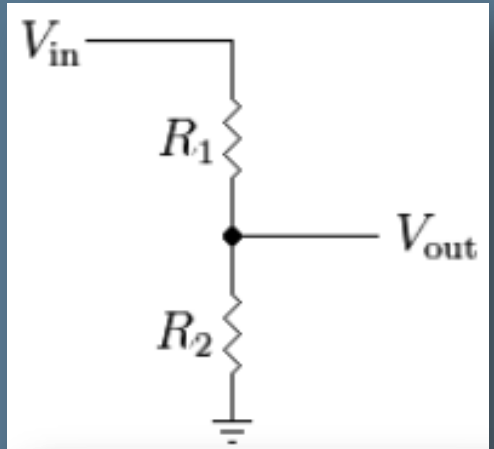
  5.2^2 / 231 = 0.1171 (watt)

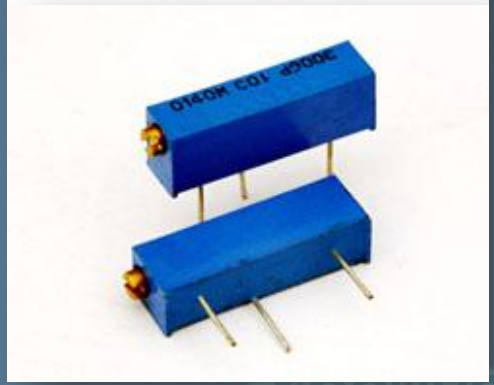
  1/8W motstand = 0,125W
- Overdriv
- Husk – det blir til



Elektronikk 101 - spenningsdeler

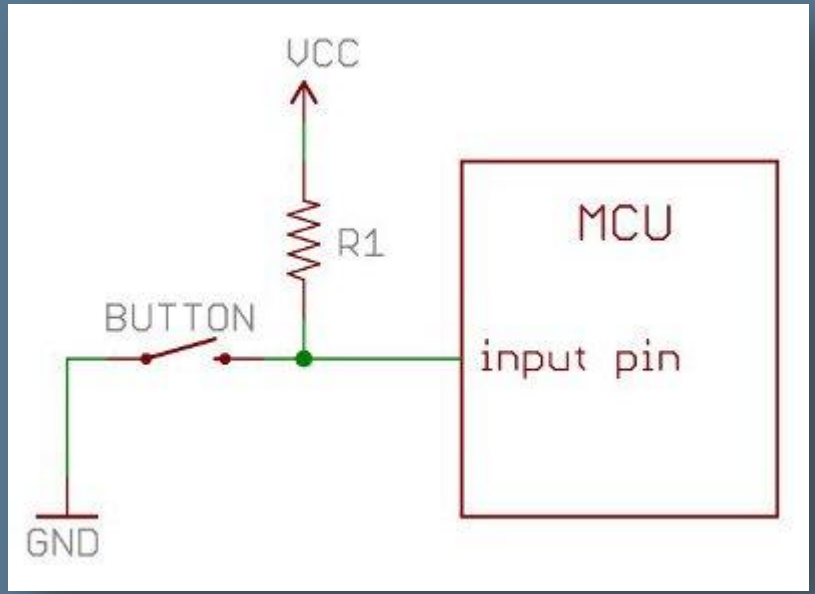
- Meget anvendelig krets
  - Reduserer spenning til å passe applikasjoner
- 2 motstander i serie
  - Mål V mellom dem
  - V_{in} deles tilsvarende motstandene
  - Eks: V_{in} = 5V + 10kΩ + 10kΩ gir V_{out} = 2.5V
- Et potensiometer er en spenningsdeler
  - Et trimpot kan brukes til å fintune en sensor
  - Den ene siden erstatter da en fast motstand



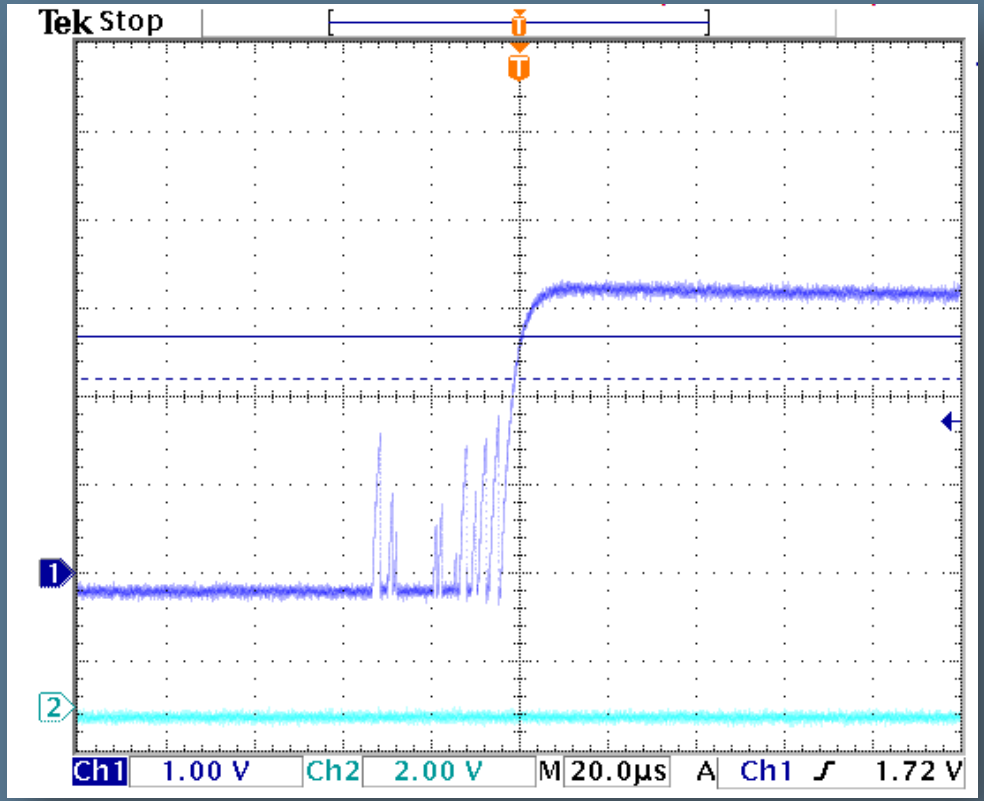


Digitalteknikk – pull up/down

- Brukes for å ha en kjent tilstand på pinner uten kontinuerlig input



Debounce



Software debounce

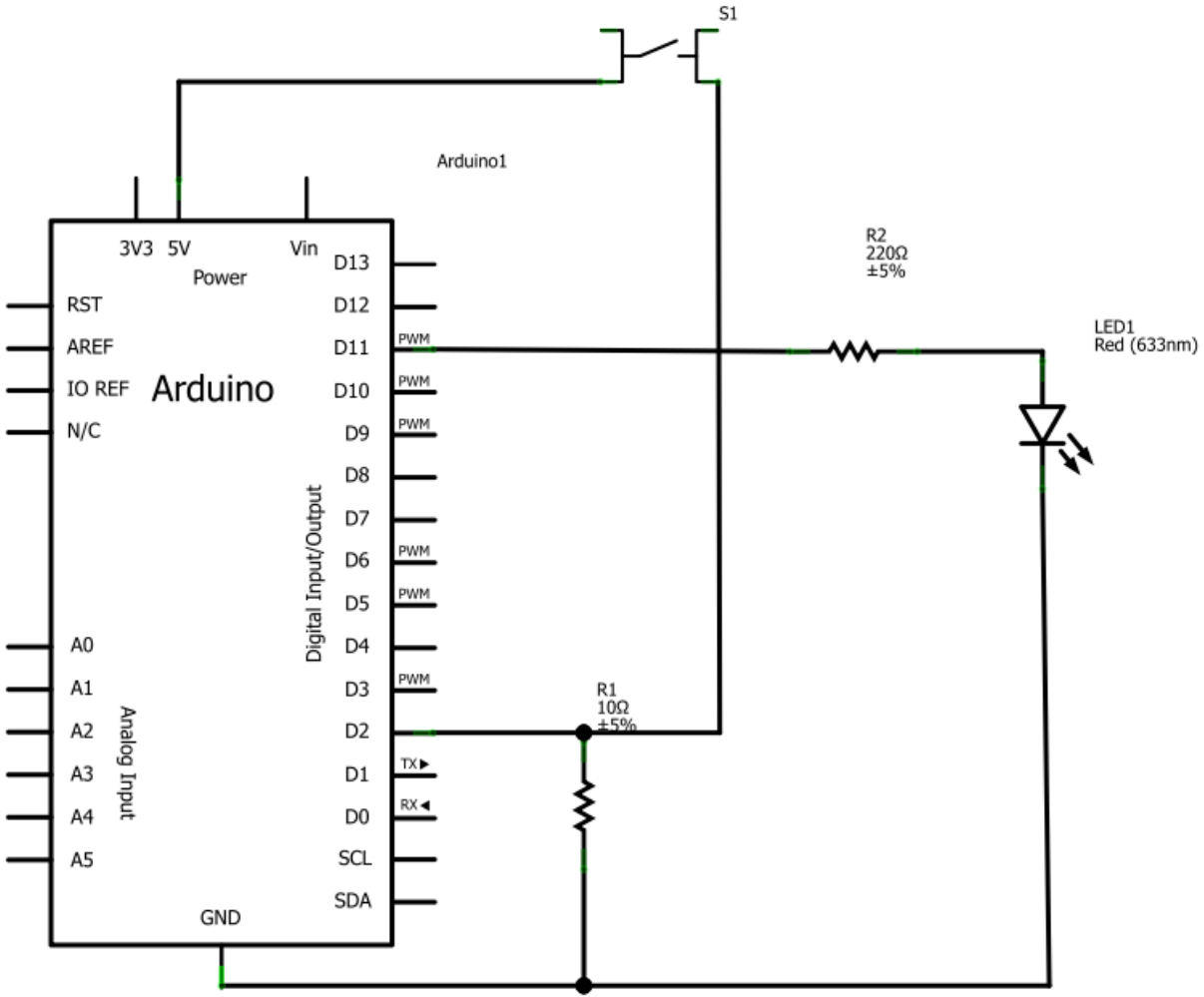
http://www.arduino.cc/en/Tutorial/Debounce

```
int lastButtonState = LOW;
long lastDebounceTime = 0;  // the last time the output pin was toggled
long debounceDelay = 50;    // the debounce time; increase if the output flickers

void loop() {
  int reading = digitalRead(buttonPin);       // read the state of the switch into a local variable:
  if (reading != lastButtonState) {           // If the switch changed, due to noise or pressing:
    lastDebounceTime = millis();              // reset the debouncing timer
  }

  if ((millis() - lastDebounceTime) > debounceDelay) {      // whatever the reading is at, it's been there for longer
    buttonState = reading;                                  // than the debounce delay, so take it as the actual current state
  }


  lastButtonState = reading;                                // save the reading.  Next time through the loop,
                                                            // it'll be the lastButtonState:
}
```

bouvet

Den magiske grå røyken

- Hvordan «drepe» en Arduino
  - Kortslutning
  - Drive elektromekanikk direkte
  - ***Maks*** 40mA på GPIO
- Har du flaks, er det bare den ene pinnen som ryker!

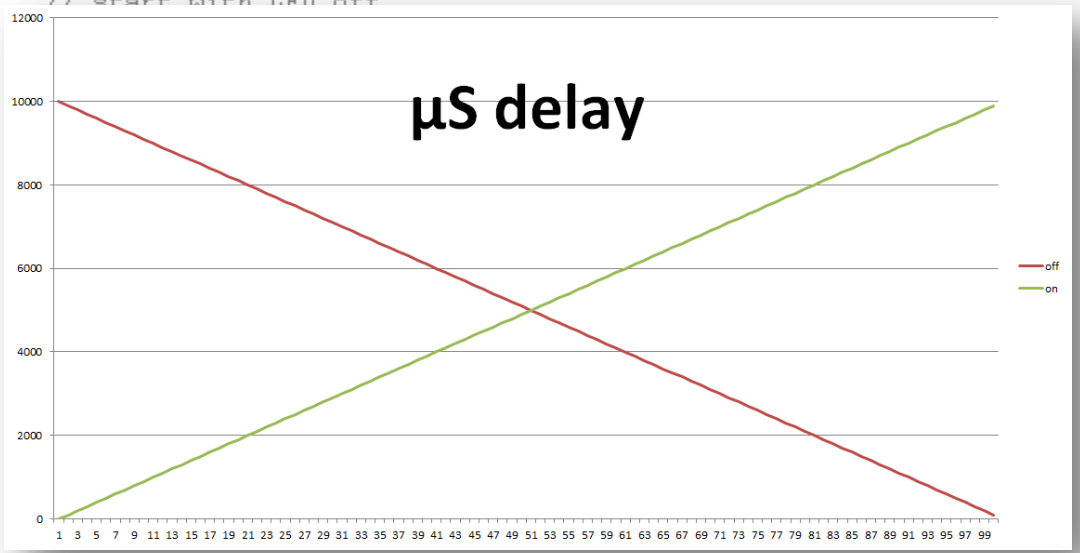

S1
Arduino1
3V3
5V
Vin
Power
RST
AREF
IO REF
N/C
Arduino
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
SCL
SDA
PWM
TX
RX
Digital Input/Output
A0
A1
A2
A3
A4
A5
Analog Input
GND
R2
220Ω
±5%
LED1
Red (633nm)
R1
10Ω
±5%

Hello World 2.1 – code init

```
#define ledPin 11                    // LED connected to pin 11
#define buttonPin 2                  // button connected to pin 2

void setup() {
  pinMode(ledPin, OUTPUT);           // write to LED pin
  pinMode(buttonPin, INPUT);         // read from button pin
}
```

Hello World 2.1 – softon/off

```
void ledSoftOn(byte pin) {                    // Soft-on function for LED
  for (int i = 1 ; i < 10000; i+=100) {       // 10000 is trial an succed number, 100 sets speed from 0 to full
    digitalWrite(pin, LOW);                   // start with LED off
    delayMicroseconds(10000 - i);
    digitalWrite(pin, HIGH);
    delayMicroseconds(i);
  }
}

void ledSoftOff(byte pin) {
  for (int i = 1 ; i < 10000; i+=100) {
    digitalWrite(pin, HIGH);
    delayMicroseconds(10000 - i);
    digitalWrite(pin, LOW);
    delayMicroseconds(i);
  }
}
```

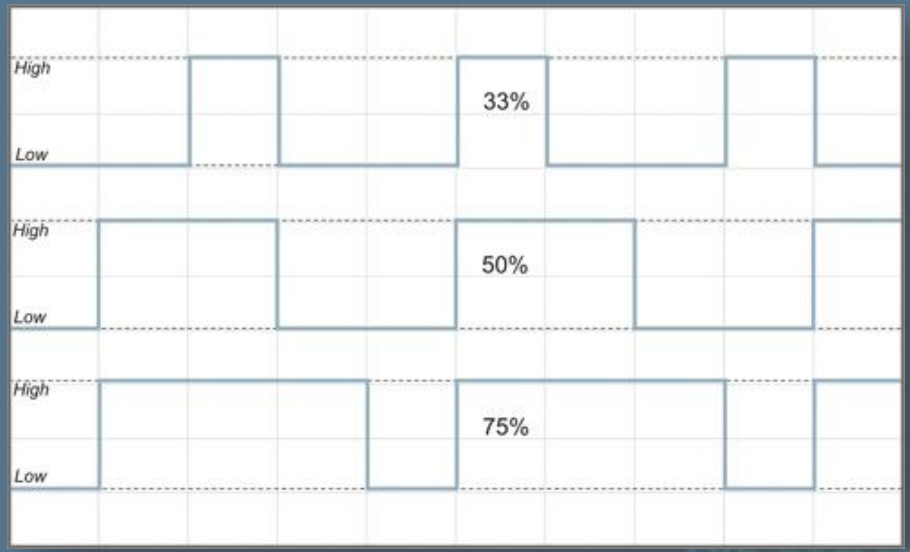


Hello World 2.1 – body

```
boolean lastState = LOW;                  // a place to remember the last state

void loop() {
  int state = digitalRead(buttonPin);     // read the current state of the button on buttonPin
  if (state == HIGH) {                    // if it's high, something's happening
    while (state == HIGH) {               // while it's still pressed
      state = digitalRead(2);             // check state
    }                                     // and there it was released...

    if (lastState == LOW) {               // if the LED was OFF
      ledSoftOn(ledPin);                  // turn it on
      lastState = HIGH;                   // and remember for next time
    }
    else {
      ledSoftOff(ledPin);                 // turn it off
      lastState = LOW;                    // and remember
    }
  }                                       // done processing the click
}                                         // goto loop()
```

PWM

- Pulse-width modulation (PWM)
  - en pinne holdes høy i en %-andel av tid
- Eksempel LED
  - Dimmer
- HW (~) eller SW implementasjon



Hello World 2.1pwm - code

```
void nop() {
  countNops++;                          // Mmm... free goo.
}

void ledSoftOn(byte pin) {              // Soft-on function for LED
  for (int i = 0 ; i <= 255; i++) {     // 0-255 is zero to full duty cycle
    analogWrite(pin,i);                 // Enable PWM with i/255 duty cycle
    unsigned long now = millis();       // Kaeu?
    while ((millis() - now) < 5) {      // Wait approx 5 ms before next iteration
      nop();                            // Totally useless but could have been ...
    }
  }
}

void ledSoftOff(byte pin) {             // same as On just the other way
  for (int i = 0 ; i <= 255; i++) {
    analogWrite(pin,255 - i);
    unsigned long now = millis();       // Kaeu?
    while ((millis() - now) < 5) {      // Wait approx 5 ms before next iteration
      nop();                            // Totally useless but could have been ...
    }
  }
}
```

PWM – Free goo!

COM9

Send

```
399483
799343
1199474
1598813
1998735
2398096
2797800
3197130
3597132
3996888
4396433
4796280
5195794
```

Autoscroll | No line ending | 115200 baud

bouvet

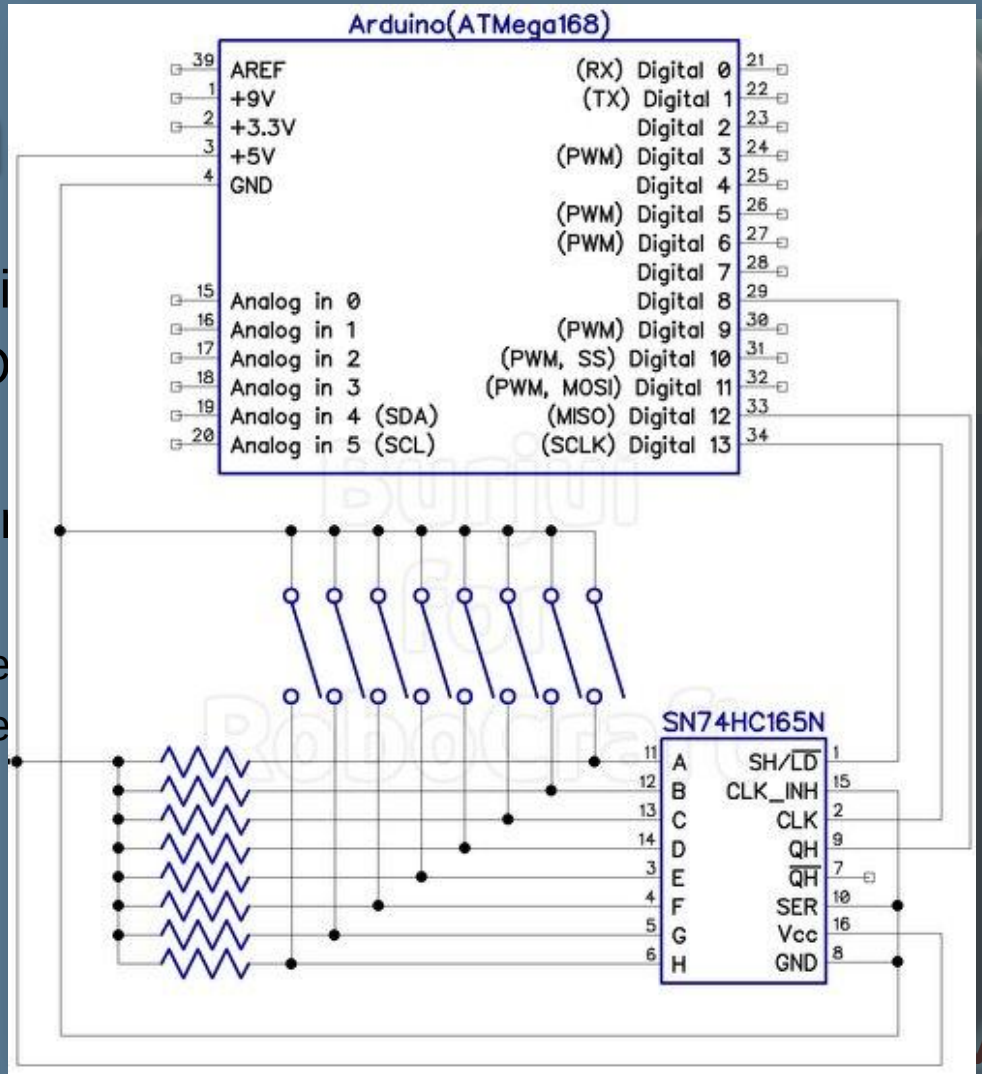
Mye brukte funksjoner

- pinMode(<pin>, INPUT|OUTPUT)
  - Default er INPUT
- digitalWrite() og digitalRead()
- analogWrite() og analogRead()
- delay() og millis()
- Serial.Begin(115200);
  - Serial.println(<val>);
  - (Ctrl+Shift+M gir Serial Monitor)

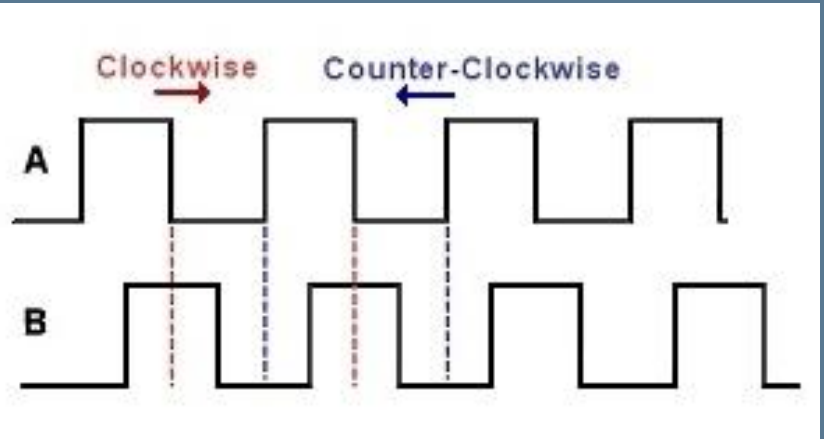
Bit-m

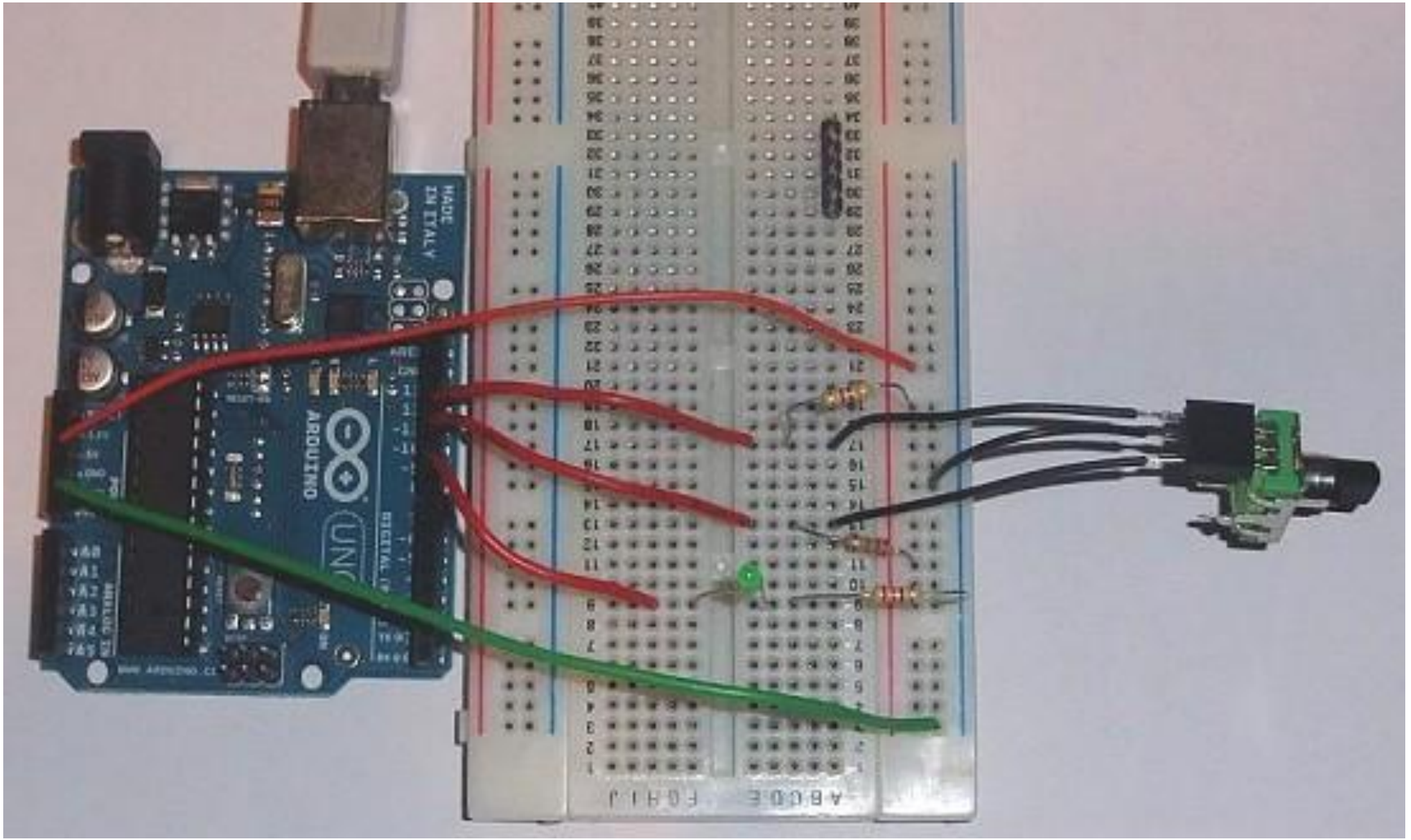
- bitRead(), bitWrite(), bi
- & (AND), | (OR), ^ (XO

- Tolke og sette flagg – r
- Multiplexing
  - 74HC595 8-bit shift re
  - 74HC165N 8-bit shift re



Neste øvelse: Rotary Encoder




MADE IN ITALY
ARDUINO
UNO

Rotary Encoder – SW part 1

```
int brightness = 120;    // how bright the LED is, start at half brightness
int fadeAmount = 10;    // how many points to fade the LED by
unsigned long currentTime;
unsigned long loopTime;
const int pin_A = 12;  // pin 12
const int pin_B = 11;  // pin 11
unsigned char encoder_A;
unsigned char encoder_B;
unsigned char encoder_A_prev=0;

void setup()  {
  pinMode(9, OUTPUT);    // declare pin 9 to be an output:
  pinMode(pin_A, INPUT);
  pinMode(pin_B, INPUT);
  currentTime = millis();
  loopTime = currentTime;
}
```

Rotary Encoder – SW part 2

```
void loop()  {
  currentTime = millis();                    // get the current elapsed time
  if(currentTime >= (loopTime + 5)){         // 5ms since last check of encoder = 200Hz

    encoder_A = digitalRead(pin_A);          // Read encoder pins
    encoder_B = digitalRead(pin_B);
    if((!encoder_A) && (encoder_A_prev)){    // A has gone from high to low
      if(encoder_B) {                          // B is high so clockwise
        // increase the brightness, dont go over 255
        if(brightness + fadeAmount <= 255) brightness += fadeAmount;
      }
      else {                                   // B is low so counter-clockwise
        // decrease the brightness, dont go below 0
        if(brightness - fadeAmount >= 0) brightness -= fadeAmount;
      }
    }
    encoder_A_prev = encoder_A;              // Store value of A for next time
    analogWrite(9, brightness);              // set the brightness of pin 9:
    loopTime = currentTime;                  // Updates loopTime
  }
}
```

Multitasking, tråder…

- Nope…
- FreeRTOS skal visstnok være OK
- «Finite State Machine» funker
  - Ressurs og særlig RAM-effektivt
  - FSM library fra *Wiring* *(kudos Alexander Brevig)*

http://wiring.uniandes.edu.co/source/trunk/wiring/firmware/libraries/FSM/

```
#include <FiniteStateMachine.h>
#define DEBUG

boolean button1press = LOW;  // HIGH when button-press is detected
unsigned long cDisplay;      // count of calls to Display state
unsigned long cButtons;      // ...
unsigned long cIncoming;
unsigned long cLong;

#ifdef DEBUG
unsigned long cDisplayLast;  // Used to calculate frequenzy if DEBUG is #defined
unsigned long cButtonsLast;  //..
unsigned long cIncomingLast;
unsigned long cLongLast;
#endif
```

```
// Initialize states
const byte NUMBER_OF_STATES = 4;

State Display = State(runUpdateDisplay);     // State Display calls function runUpdateDisplay
State Buttons = State(runPollButtons);       //..
State Incoming = State(runPollIncoming);
State LongInterval = State(runLongInterval);

FSM fsmDemo = FSM(Display);                  // Start FSM in runUpdateDisplay state

unsigned long now = 0;                       // Used to hold time of entry
unsigned long lastlong = 0;                  // last time run for the long cycle state
unsigned long lastshort = 0;                 // last time run for the short cycle state

void setup() {
  Serial.begin(115200);  // Don't waste time waiting for slow communications
  pinMode(2,INPUT);      // button connected to pin 2 with pull-down
  pinMode(11,OUTPUT);    // LED connected to pin 11 with current limiting resistor
  now = micros();        // initialize variables used to keep track of time
  lastlong = now;        // ..
  lastshort = now;
}
```

```
unsigned long freq = 0;  // counter to distribute non-timed states

void loop(){
  freq++;
  now = micros();

  if ((now-lastlong) >= 1000000){
    fsmDemo.transitionTo(LongInterval);       // do the long interval
    lastlong = micros();
  }
  else if ((now-lastshort) >= 100000) {       // do the short interval
    fsmDemo.transitionTo(Display);
    lastshort = micros();
  }
  else if (freq % 4 == 0) {                   // empty the UART max every 1 in 4 iterations - long/short executions
    fsmDemo.transitionTo(Incoming);
  }
  else {
    fsmDemo.transitionTo(Buttons);            // Spend the most time looking for button presses
  };

  fsmDemo.update();
}
```

```
boolean ledState = LOW;                    // To keep track of LED state without having to read the pin

void runUpdateDisplay(){                   // Update display
  cDisplay++;                              // count number of calls
  if (button1press == HIGH) {              // ...if button press was detected
    ledState = !ledState;                  // swap state
    digitalWrite(11,ledState);             // and write the new state to the led
    button1press = LOW;                    // done processing this button press - prepare for the next
  };
}

boolean button1in = LOW;                   // To keep trak of when a button is in the pressed state

void runPollButtons(){                     // Polls the buttons
  cButtons++;                              // counting the number of calls

  if (button1in == HIGH) {                 // If the button tracking is in the pressed state
    if (digitalRead(2) == LOW){            // and the pin is low - it has been released - PS: Just lucky debounce due to the release is
      button1in = LOW;                     // it's not pressed anymore
      button1press = HIGH;                 // but a press has been detected
    };
  }
  else {                                   // button tracking is in depressed state
    if (digitalRead(2) == HIGH){           // but the pin is HIGH, soo....
      button1in = HIGH;                    // set the button tracking to pressed state
    };
  };
};
```

```
void runPollIncoming(){                    // booooring
  cIncoming++;                             // counting the number of calls
};

void runLongInterval(){                    // used to write to the serial line if debugging -
  cLong++;                                 // counting the number of calls

#ifdef DEBUG                               // #define DEBUG to print statistics to serial
  Serial.print(now);                       // show the timer variables
  Serial.print("-");
  Serial.print(lastlong);
  Serial.print("=");
  Serial.print(now-lastlong);
  Serial.println();

  Serial.print("Calls to Display: ");      // show the number of calls to the different states
  Serial.print(cDisplay);
  Serial.print(", Buttons: ");
  Serial.print(cButtons);
  Serial.print(", Incoming: ");
  Serial.print(cIncoming);
  Serial.print(", Long: ");
  Serial.print(cLong);
```

```
  Serial.print(". Frequency/s: ");          // show the count of calls since last long interval
  Serial.print((cDisplay-cDisplayLast));
  Serial.print(", ");
  Serial.print((cButtons-cButtonsLast));
  Serial.print(", ");
  Serial.print((cIncoming-cIncomingLast));
  Serial.print(", ");
  Serial.println((cLong-cLongLast));

  cDisplayLast = cDisplay;                  // remember the last number
  cButtonsLast = cButtons;
  cIncomingLast = cIncoming;
  cLongLast = cLong;
#endif

};
```

COM9

Send

```
Calls to Display: 29, Buttons: 79612, Incoming: 26540, Long: 3. Frequency/s: 10, 26464, 8824, 1
4000132-4000140=4294967288
Calls to Display: 39, Buttons: 106078, Incoming: 35361, Long: 4. Frequency/s: 10, 26466, 8821, 1
5000140-5000148=4294967288
Calls to Display: 49, Buttons: 132541, Incoming: 44182, Long: 5. Frequency/s: 10, 26463, 8821, 1
6000164-6000172=4294967288
Calls to Display: 59, Buttons: 159003, Incoming: 53004, Long: 6. Frequency/s: 10, 26462, 8822, 1
7000184-7000192=4294967288
Calls to Display: 69, Buttons: 185465, Incoming: 61826, Long: 7. Frequency/s: 10, 26462, 8822, 1
8000200-8000208=4294967288
Calls to Display: 79, Buttons: 211929, Incoming: 70646, Long: 8. Frequency/s: 10, 26464, 8820, 1
9000220-9000228=4294967288
Calls to Display: 89, Buttons: 238391, Incoming: 79468, Long: 9. Frequency/s: 10, 26462, 8822, 1
10000240-10000248=4294967288
Calls to Display: 99, Buttons: 264854, Incoming: 88289, Long: 10. Frequency/s: 10, 26463, 8821, 1
11000260-11000268=4294967288
Calls to Display: 109, Buttons: 291309, Incoming: 97109, Long: 11. Frequency/s: 10, 26455, 8820, 1
12000284-12000292=4294967288
Calls to Display: 119, Buttons: 317763, Incoming: 105927, Long: 12. Frequency/s: 10, 26454, 8818, 1
13000304-13000312=4294967288
Calls to Display: 129, Buttons: 344214, Incoming: 114745, Long: 13. Frequency/s: 10, 26451, 8818, 1
14000316-14000324=4294967288
Calls to Display: 139, Buttons: 370663, Incoming: 123564, Long: 14. Frequency/s: 10, 26449, 8819, 1
15000336-15000348=4294967284
Calls to Display: 149, Buttons: 397113, Incoming: 132383, Long: 15. Frequency/s: 10, 26450, 8819, 1
16000348-16000356=4294967288
Calls to Display: 159, Buttons: 423563, Incoming: 141201, Long: 16. Frequency/s: 10, 26450, 8818, 1
17000360-17000372=4294967284
Calls to Display: 169, Buttons: 450015, Incoming: 150018, Long: 17. Frequency/s: 10, 26452, 8817, 1
18000388-18000396=4294967288
Calls to Display: 179, Buttons: 476468, Incoming: 158834, Long: 18. Frequency/s: 10, 26453, 8816, 1
19000412-19000420=4294967288
Calls to Display: 189, Buttons: 502921, Incoming: 167650, Long: 19. Frequency/s: 10, 26453, 8816, 1
20000436-20000444=4294967288
Calls to Display: 199, Buttons: 529371, Incoming: 176469, Long: 20. Frequency/s: 10, 26450, 8819, 1
```

Autoscroll

No line ending

115200 baud

Minne

- SRAM 1kb
  - SP peker på toppen av SRAM
    - Kall og ISR øker stacken = vokser nedover
    - Kan overskrive variabler ved rekursjon

Libraries

- Arduino IDE 1.0
  - #include <WProgram.h>

```
In file included from stateMachineDemo.cpp:1:
C:\Program Files (x86)\arduino-1.0\libraries\FSM/FiniteStateMachine.h:33:22: error: WProgram.h: No such file or directory
stateMachineDemo.cpp:2:24: error: Button.h: No such file or directory
```

  - Må endres til
    #include <Arduino.h>

bouvet

Hjelp!

- arduino.cc/playground www.freeduino.org
- Kjekke butikker
  - Arduino: Arduino.cc, Robonor, Sparkfun, Adafruit, DFRobot, (AVRfreaks)
  - Elektronikk: Elfa, Jameco, Digikey, Mouser, Futurlec
- EDA software – electronic design automation
  - Fritzing http://fritzing.org/
    Glimrende til dokumentasjon av prosjektet ditt - Arduino
    - Breadboard design
    - Schematic Capture
    - PCB layout
  - Virtual Breadboard http://www.virtualbreadboard.net/
    - Breadboard design, simulation
  - Proteus VSM http://www.labcenter.com
    - Kommersielt produkt med SPICE simulator (analog simulator)
    - Simulerer hele kretsen – laster koden og kjører realtime

bouvet

HACK A DAY