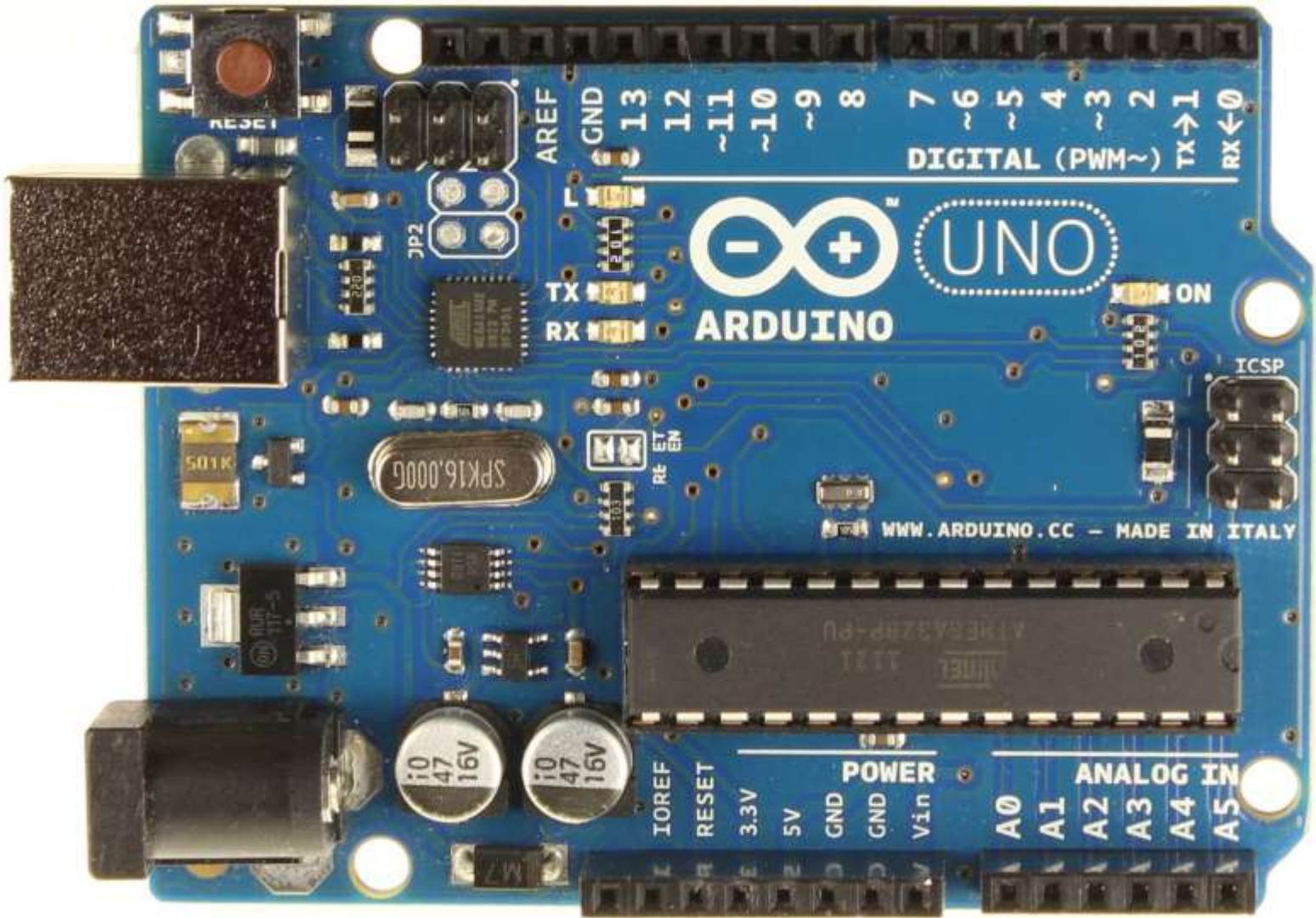


Arduino



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)

Hva er Arduino?

Hardware



Utviklings-
miljø



Community

arduino.cc/forum
arduino.cc/playground



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)

Strøm
DC7-12v
Center+

USB
+strøm inn

Reset

RX/TX LED

13 LED

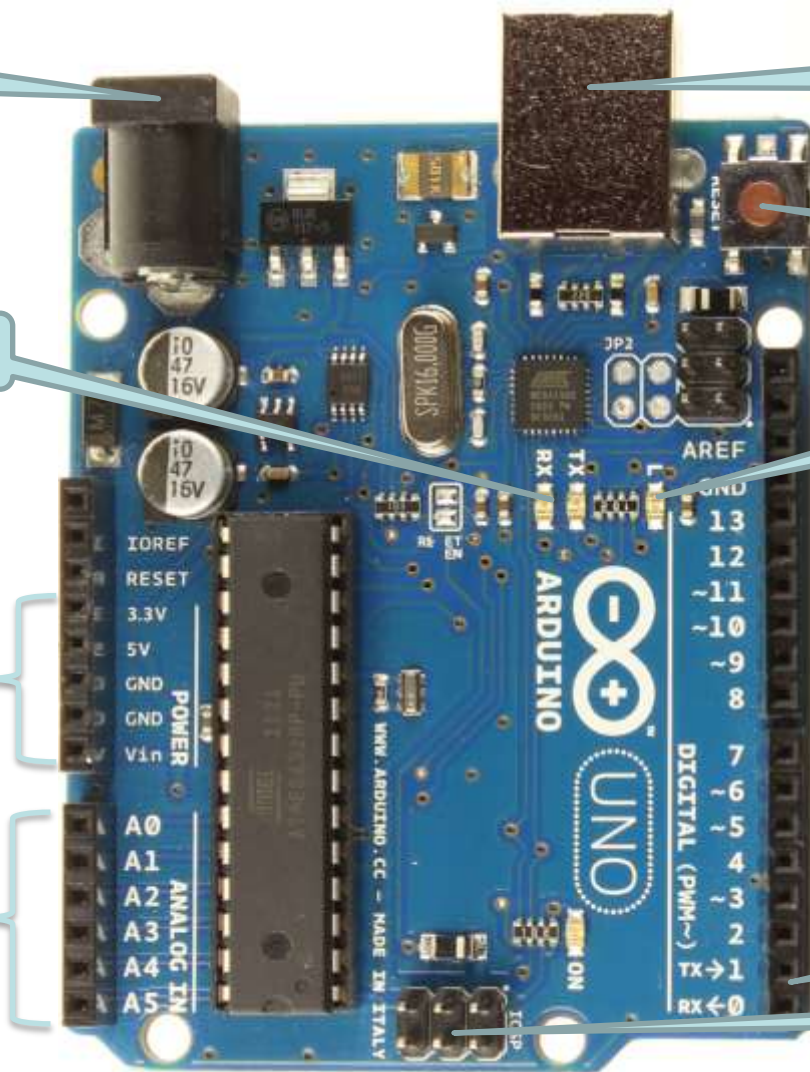
Strøm ut

GPIO
Digital IO
~PWM

Analog Input
(DAC)

TTL USART

ICSP



Shield – ferdigtenkte utvidelser

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



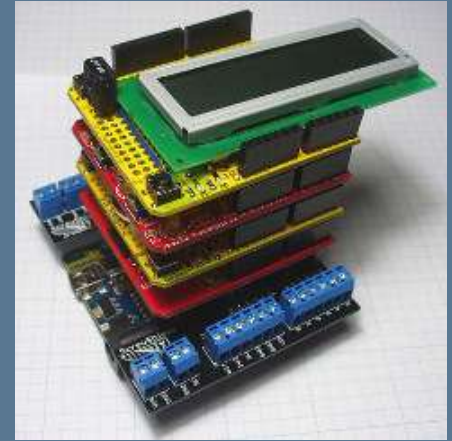
Ethernet
Shield



Wireless
SD Shield



Motor
Shield



og 282 andre...
www.shieldlist.org

...og komponenter/sensorer

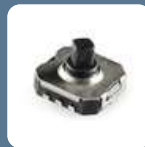
- ICer



- Breakout

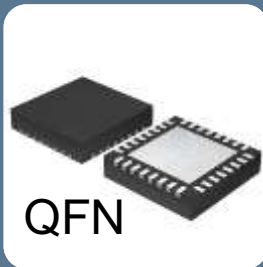


- Passive komponenter

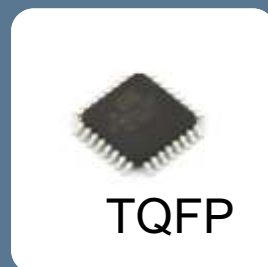


Prosessoren, ATmega328p

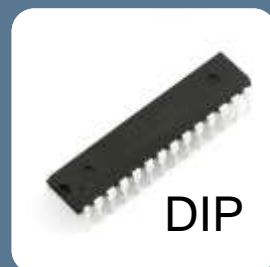
- 32kb FLASH (2kb bootloader), 1kb EEPROM, 2kb RAM
 - Mer minne: EEPROM eller SD kort på I2C/SPI
- 32 pin, 23 I/O herav 6 PWM
- 2 SPI, 1 TWI (I2C) og 1 UART
- 8 ADC pinner på 10 bit (0-1023), 15 ksps
- 1 MIPS pr mHz
- 5V og 3.3V
 - 5V tåler 3.3V logikk



QFN



TQFP



DIP

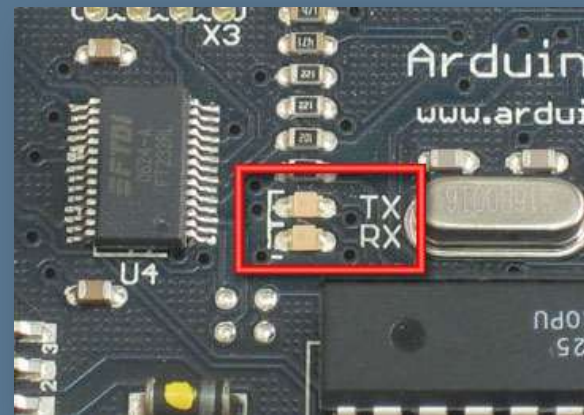
Kommunikasjon

- 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/I²C 2 ledere + strøm (Phillips)
- SPI 4 ledere + strøm (Motorola)
- 1-Wire 1 leder + jord (Dallas)

Seriell kommunikasjon

- USB er TTL – 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



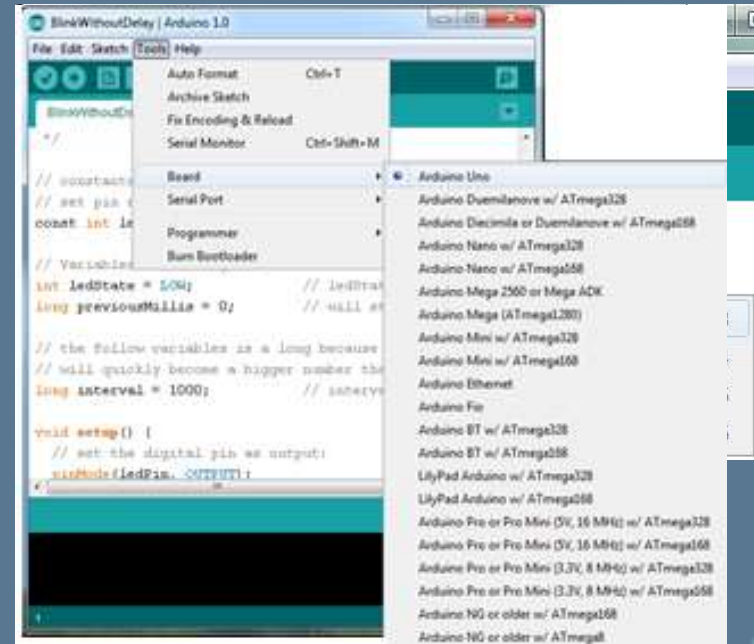
Utviklingsmiljø

- Arduino 1.0
 - 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
- Extract
 - c:\program files\arduino



Utviklingsmiljøet

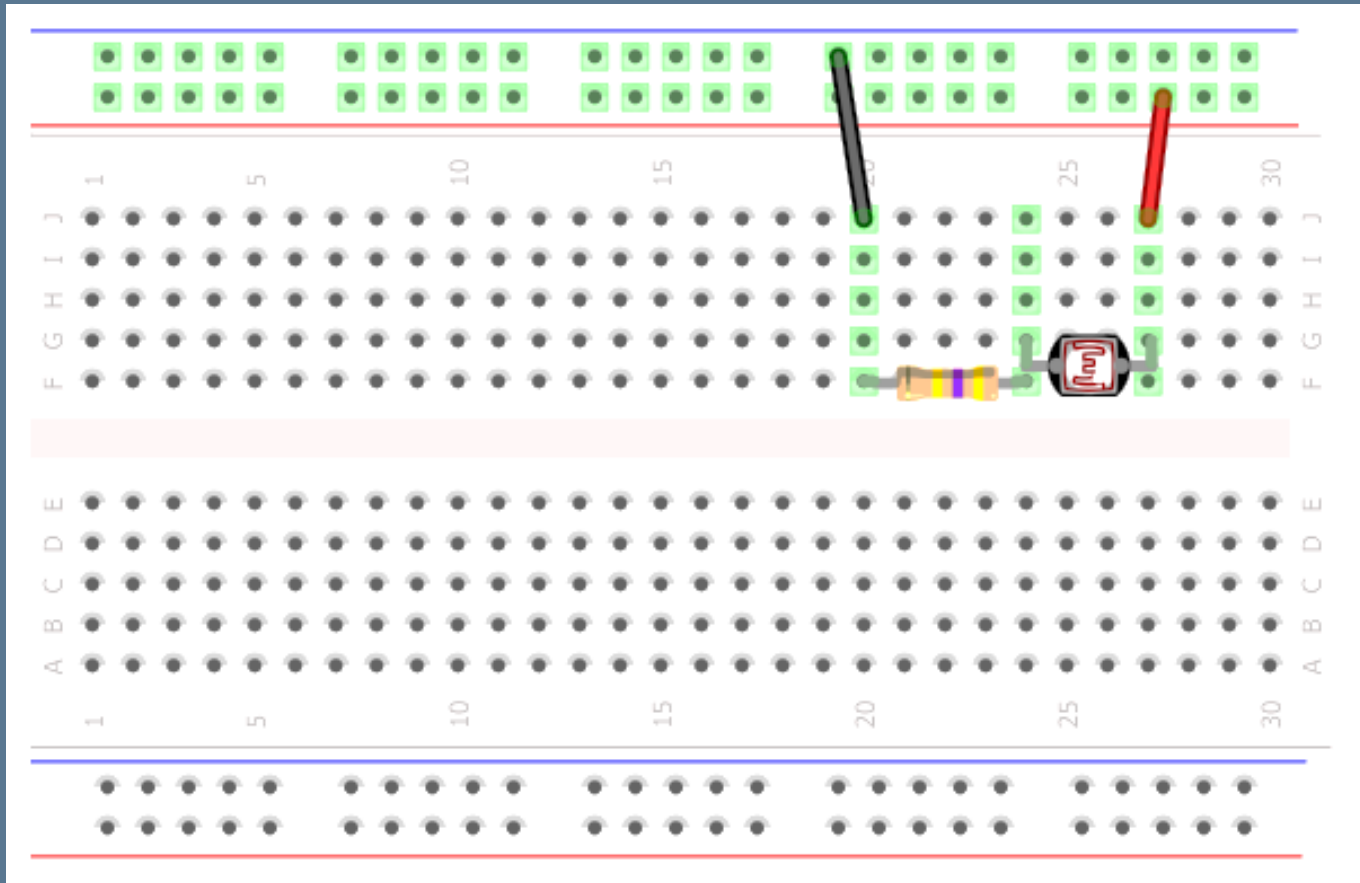
- IDE med forenklet C++ basert på «Wiring»
- Et program kalles en «sketch» og har 2 hovedbestanddeler
 - void setup()
 - void loop()
- Ctrl+Shift+F slår opp hjelp
- Ctrl+R compile
- Ctrl+U upload
(reset først hvis den feiler)

Biblioteker – libraries

– c:\program files\arduino-1.0\libraries\...

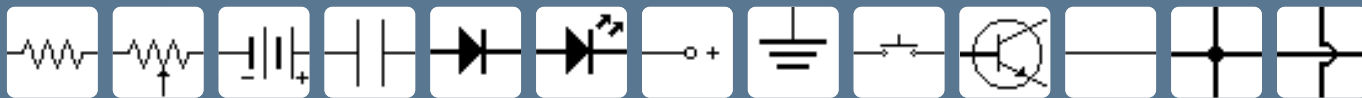
lib	funksjon
EEPROM	reading and writing to "permanent" storage
Ethernet	for connecting to the internet using the Arduino Ethernet Shield
Firmata	for communicating with applications on the computer using a standard serial protocol.
LiquidCrystal	for controlling liquid crystal displays (LCDs)
SD	for reading 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



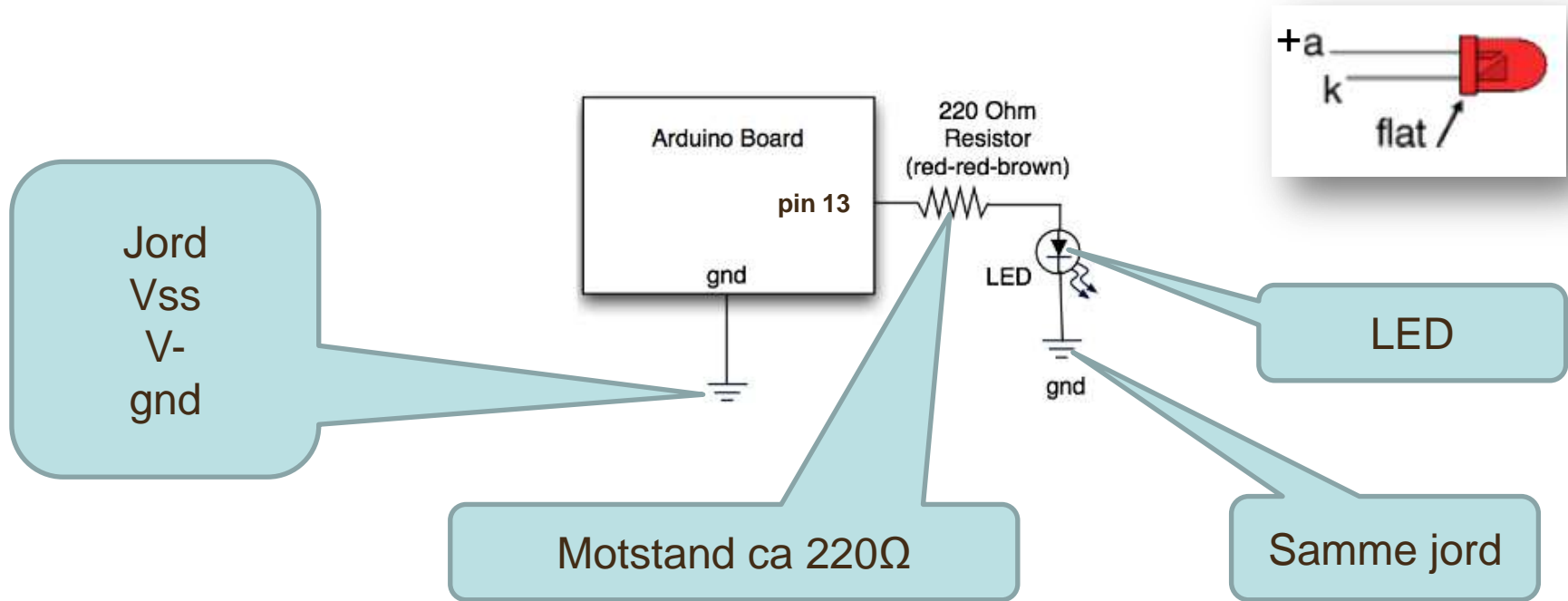
Schematics/Kretsskjema

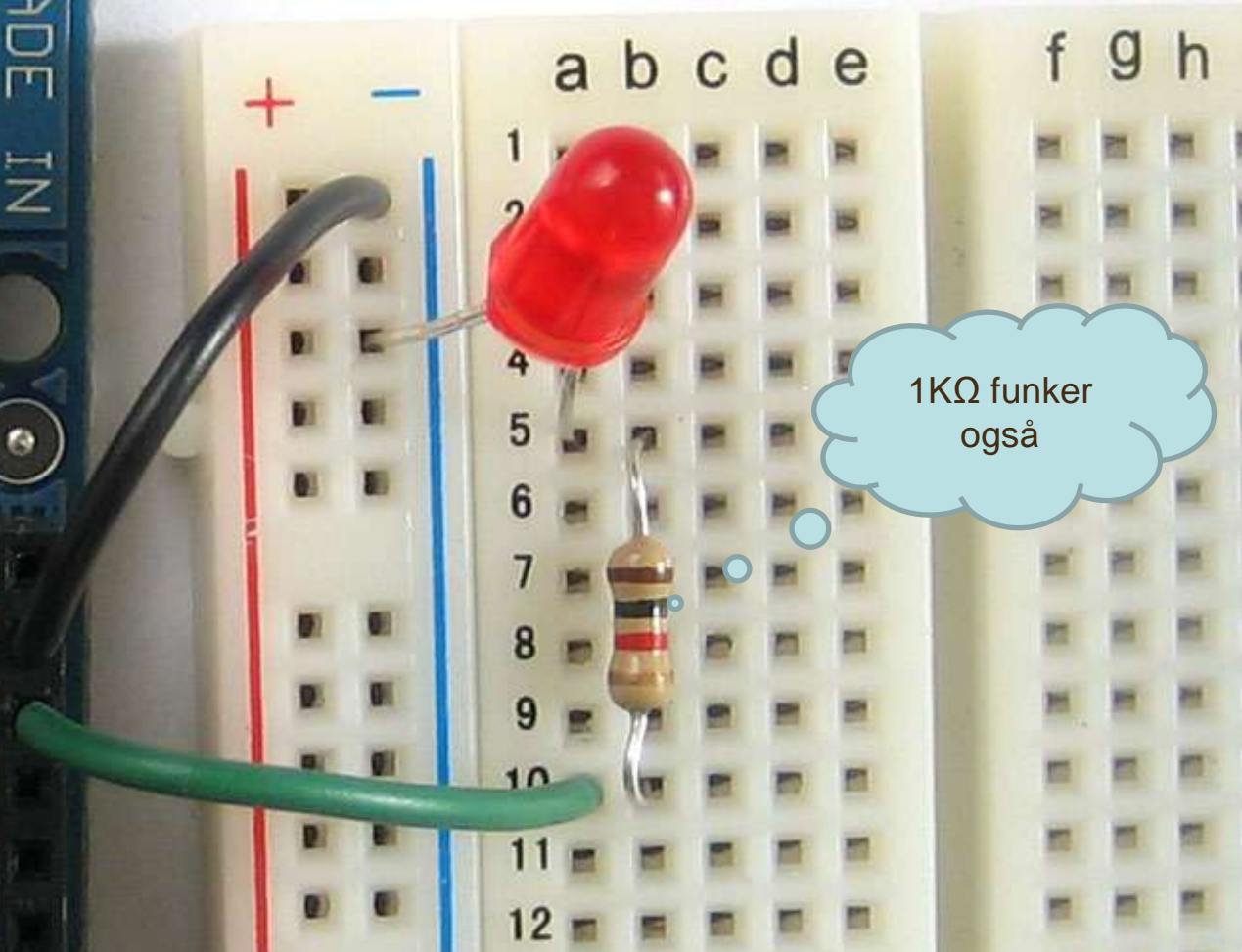
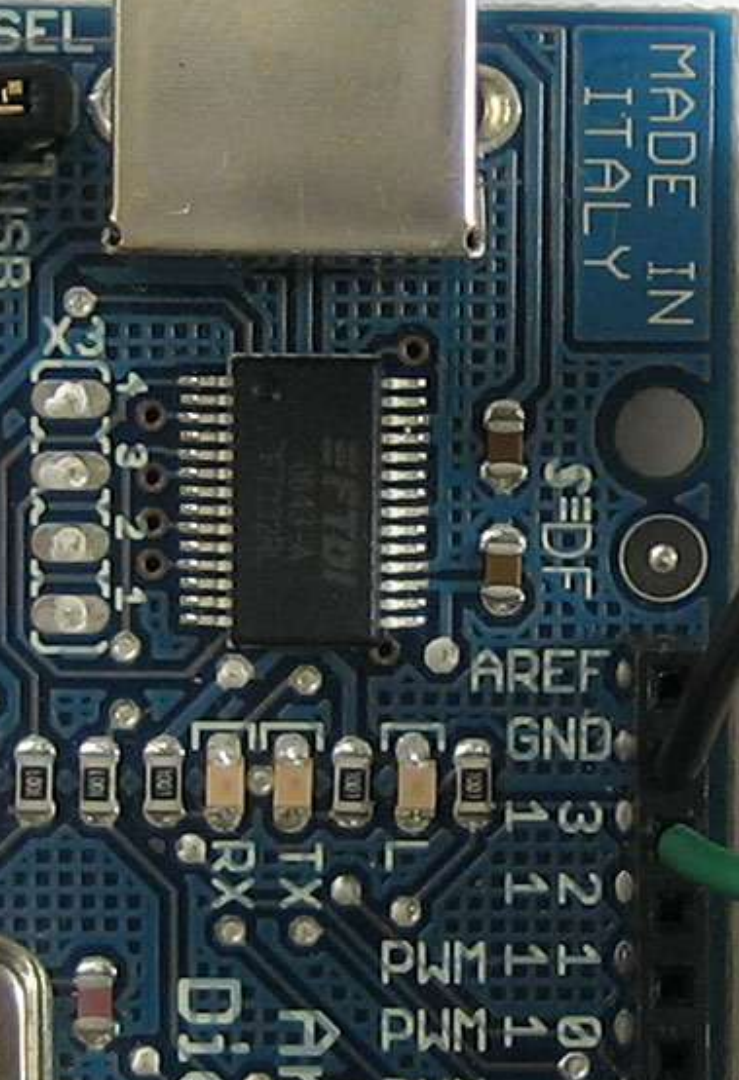
- Viser elektroniske kretser med standardiserte symboler
 - http://library.thinkquest.org/10784/circuit_symbols.html



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

Hello World - HW





1KΩ funker
også

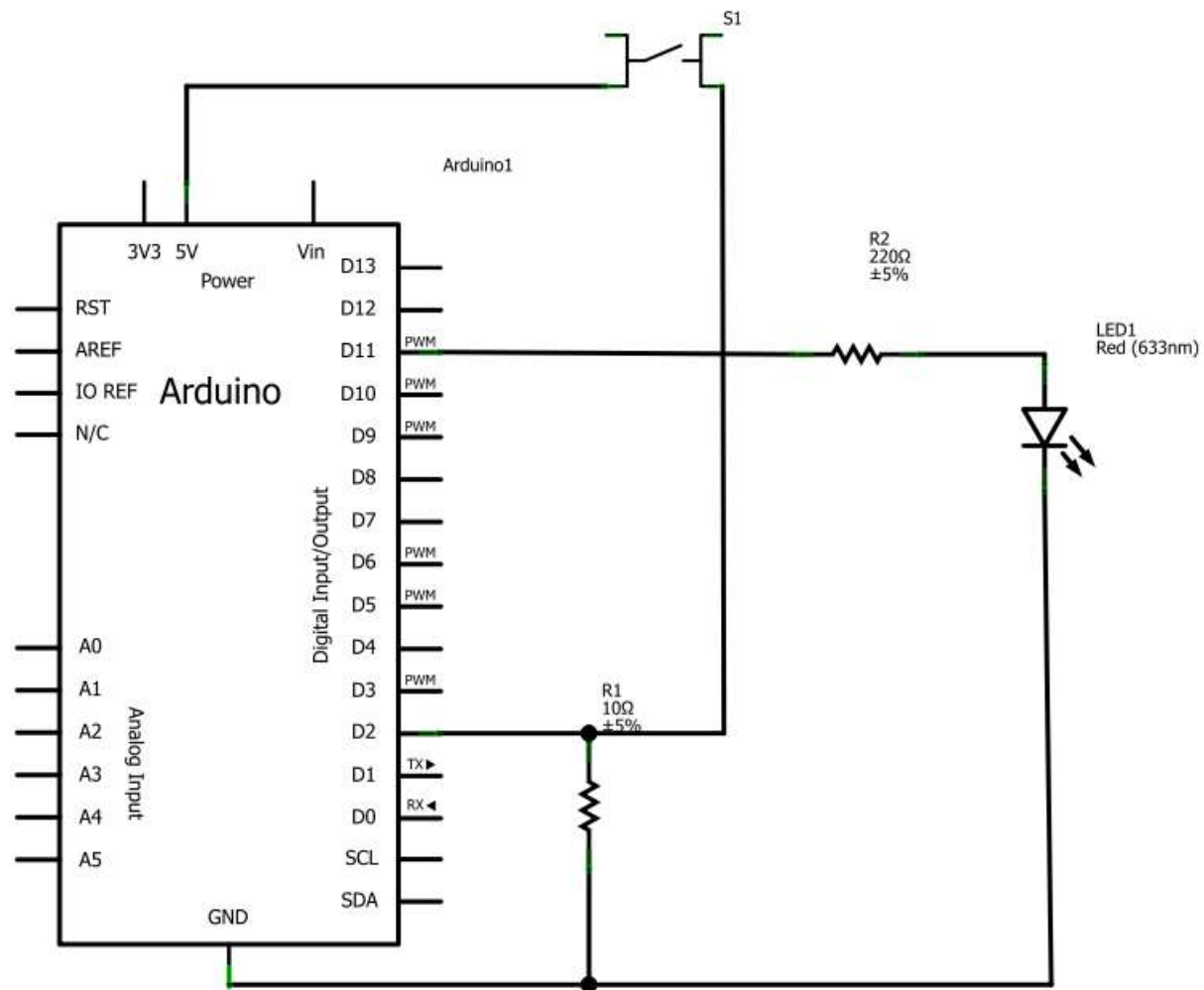
Hello World - code

```
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



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

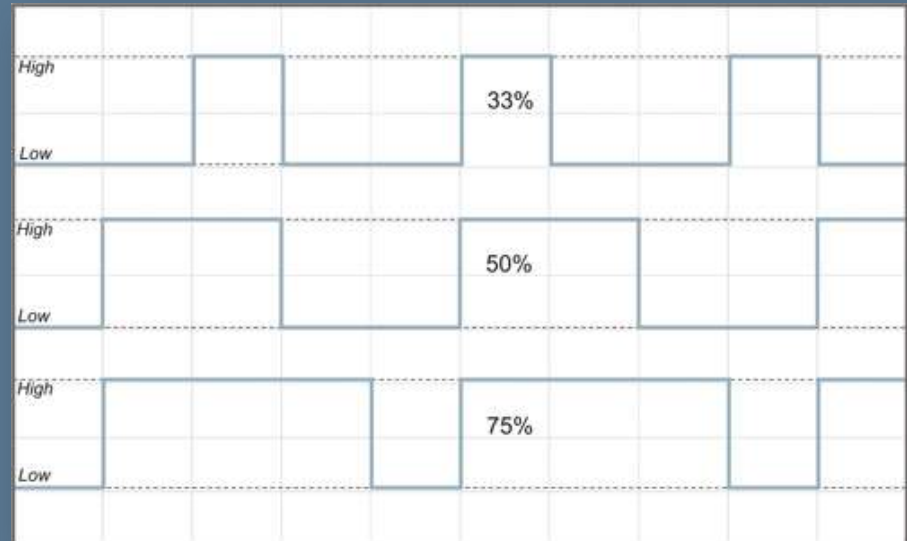
```
boolean lastState = LOW;           // a place to remember the last state

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

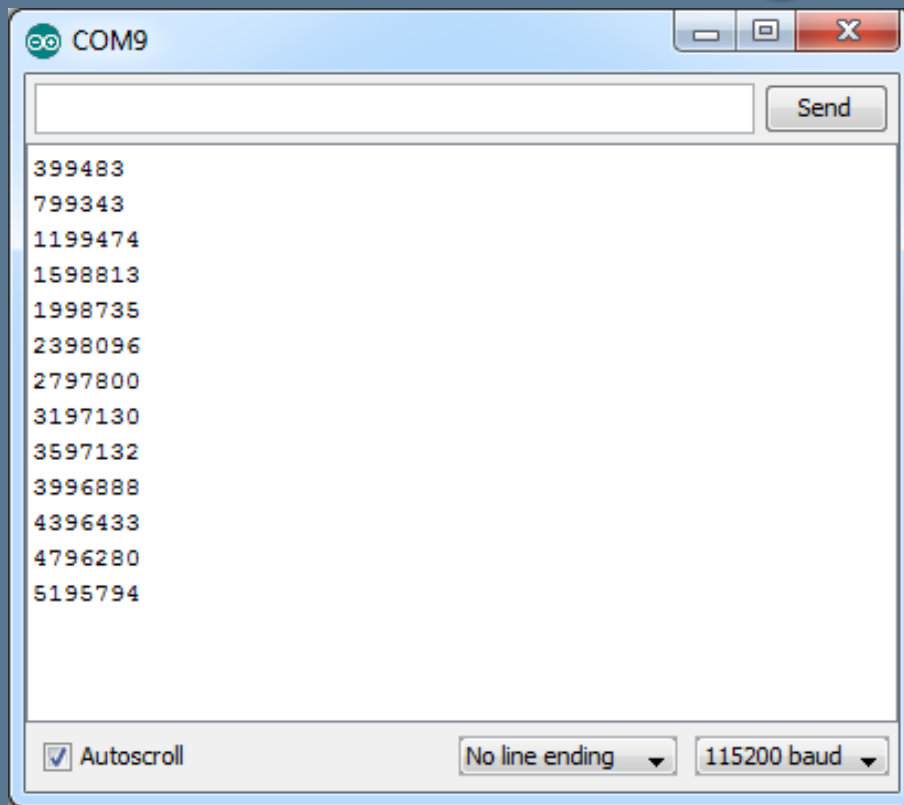
- Pulse-width modulation (PWM)
 - en pinne holdes høy i en %-andel av tid
- Eksempel LED
 - Dimmer
- HW (~) eller SW implementasjon



Hello World 2.1pwm - code

```
void nop() {  
    countNops++;           // Mhm... 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!



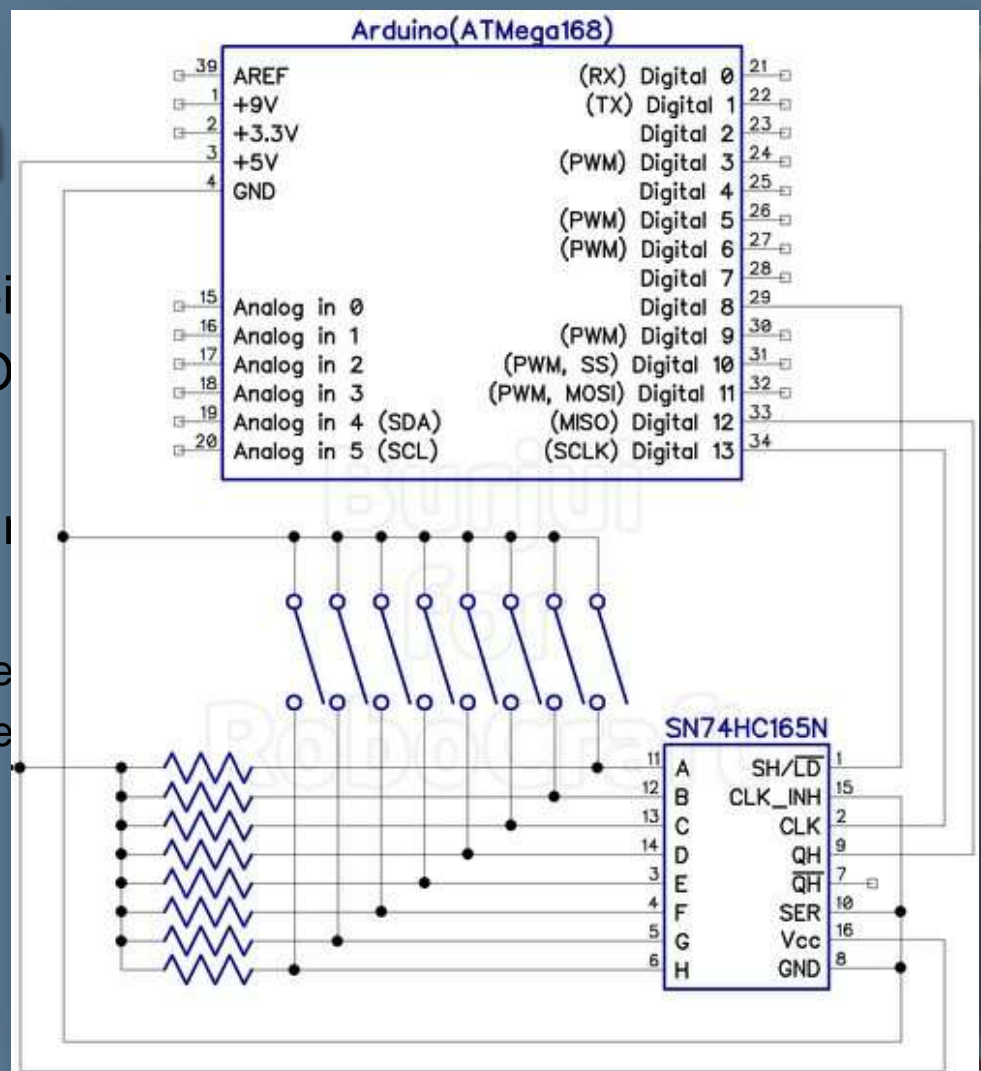


Mye brukte funksjoner

- `pinMode(<pin>, INPUT|OUTPUT)`
 - Default er INPUT
- `digitalWrite()` og `digitalRead()`
- `analogWrite()` og `analogRead()`
- `delay()` og `millis()`
- `serial.Begin(9600);`
 - `serial.println(<val>);`
 - (Ctrl+Shift+M gir Serial Monitor)

Bit-m

- `bitRead()`, `bitWrite()`, `bitSet()`, `bitClear()`
- `&` (AND), `|` (OR), `^` (XOR), `~` (NOT)
- Tolk og sette flagg – I/O
- Multiplexing
 - 74HC595 8-bit shift register
 - 74HC165N 8-bit shift register



A decorative vertical pattern on the left side of the slide, consisting of stylized circuit traces in blue, green, and yellow, with various circular and rectangular shapes representing components or nodes.

Litt grunnleggende elektronikk



Elektronikk 101 – passive komponenter

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

Elektronikk 101 – motstand

- Fast
- Variabel
- PTC/NTC
- LDR

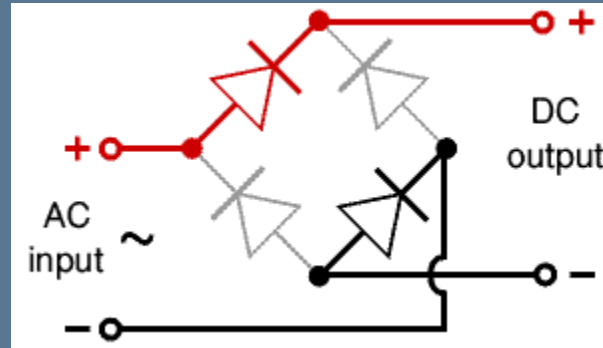
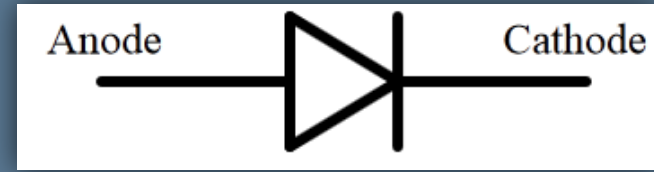


(Thermistor)

(Photoresistor)

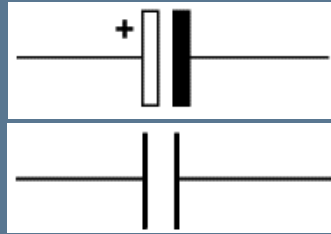
Elektronikk 101 – Dioder

- P-N
- LED
- Zener



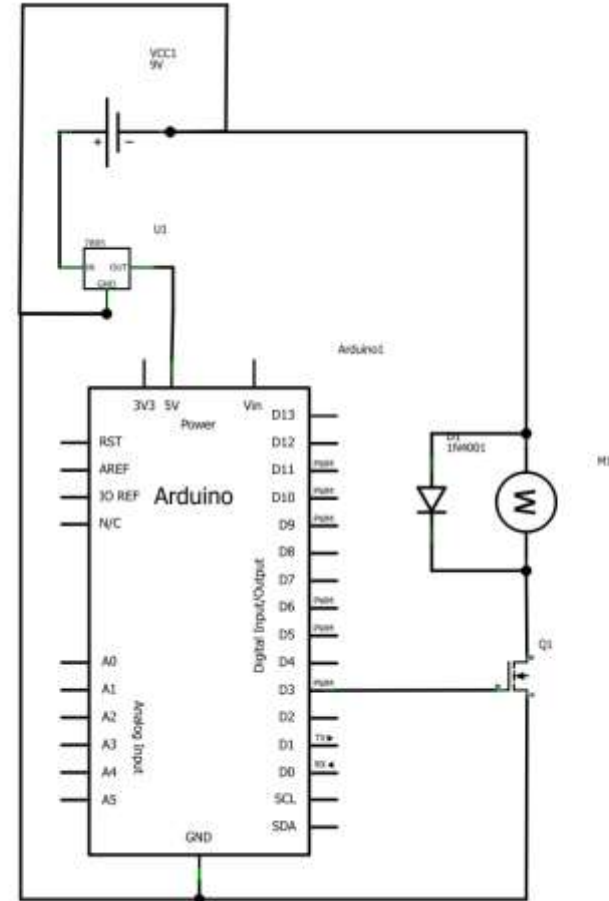
Elektronikk 101 – Kondensator

- Lagrer strøm, måles i F
 - Polariserte
 - Upolariserte



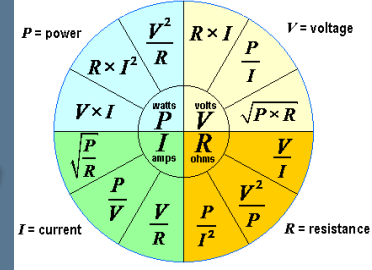
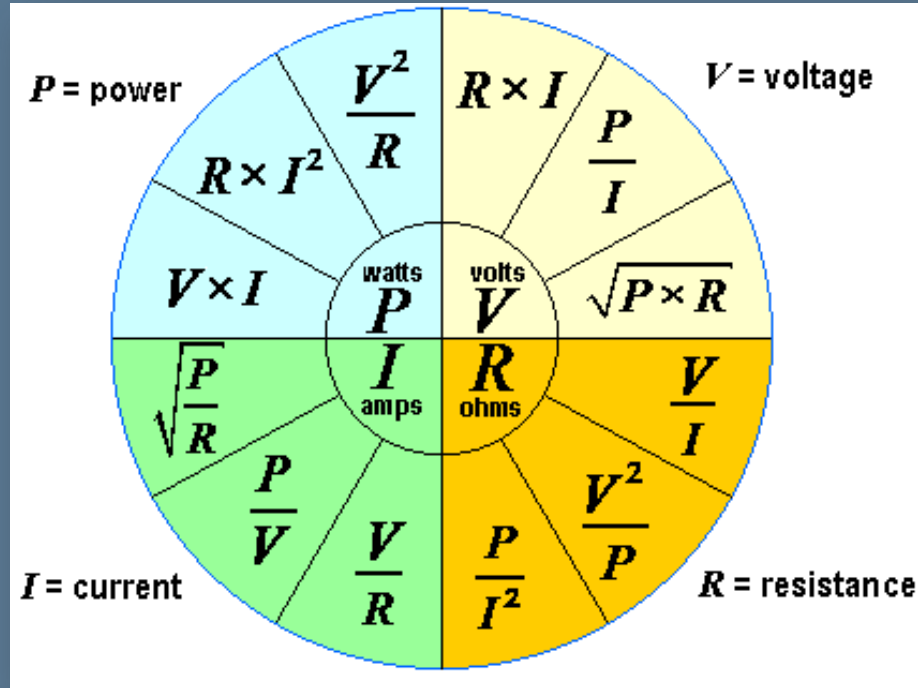
Elektronikk 10

- Forsterker strøm
 - BJT: NPN, PNP
 - Base, Collector, Emitter
 - (MOS)FET
 - Ofte brukt i høyeffekts logiske kretser raskt switchehastighet og høy effekt (A)



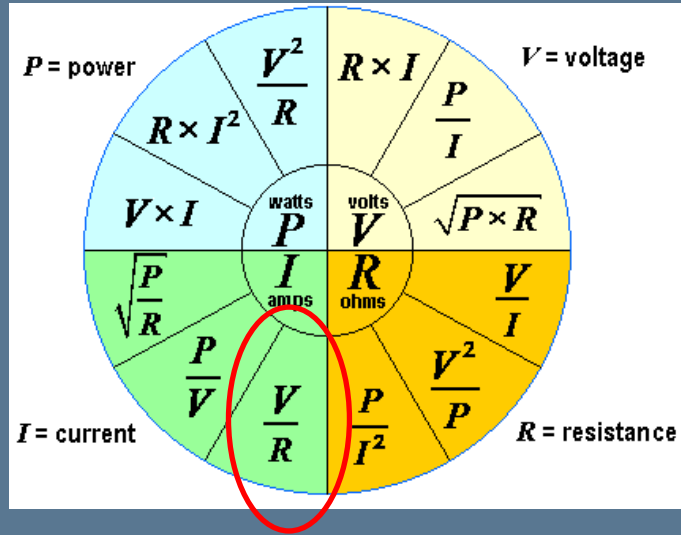
Elektronikk 101 – Ohm Ω

I
 V
 R
 P

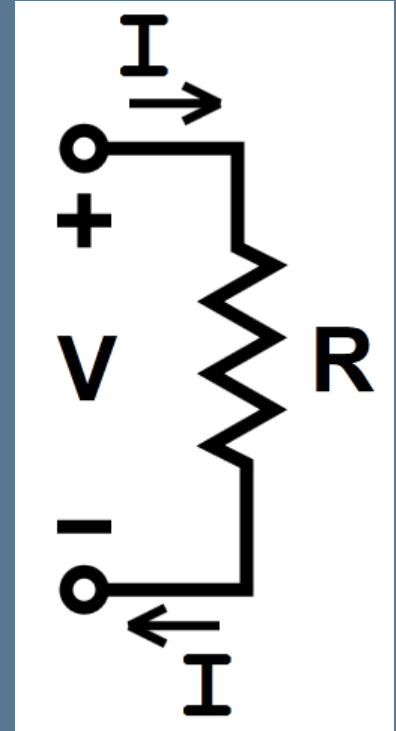


A, osv.
V, osv.
m, k Ω , osv.

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
 - 220 Ω motstand (Rød-Rød-Brun-Gull, 5%)
 - 209 Ω – 231 Ω



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

Elektronikk 101 - motstand

- Komponenter må tåle effekten de utsettes for

- P, effekt (watt) på ting

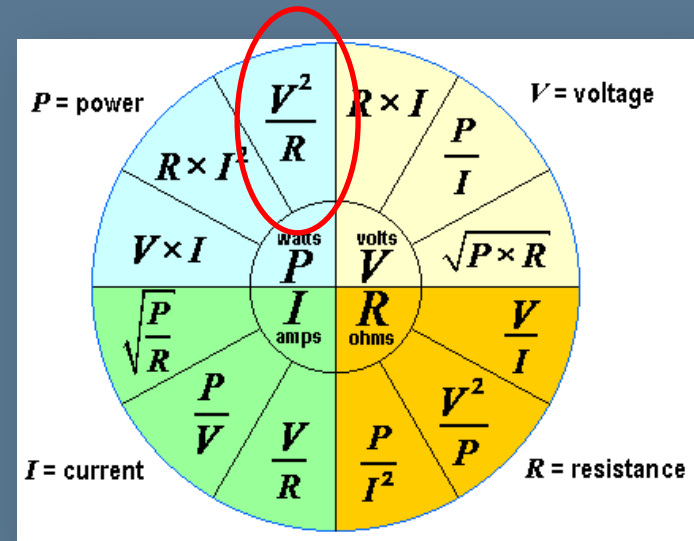
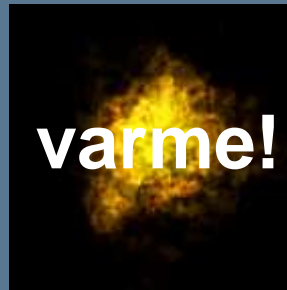
$$4.8^2 / 209 = 0.1102 \text{ (watt)}$$

$$5.2^2 / 231 = 0.1171 \text{ (watt)}$$

$$1/8\text{W motstand} = 0,125\text{W}$$

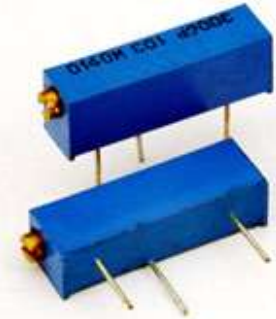
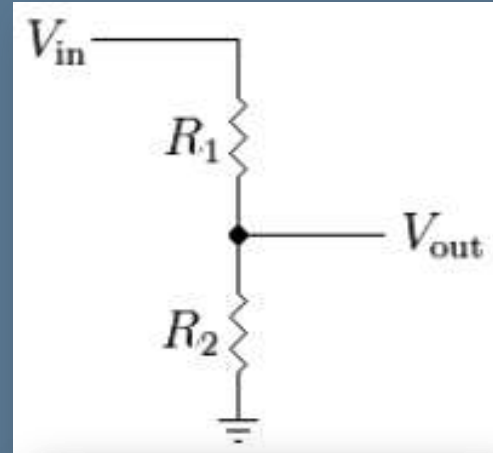
- Overdriv

- Husk – det blir til **varme!**



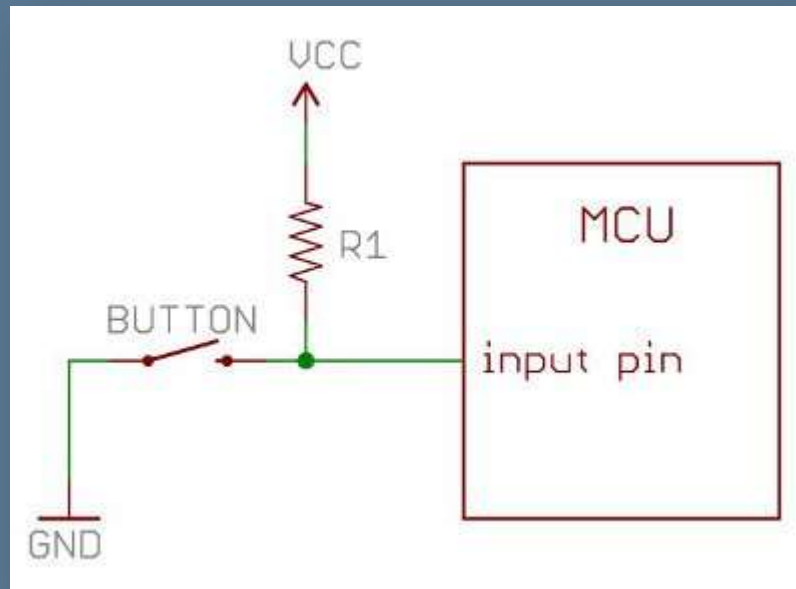
Elektronikk 101 - spenningsdeler

- Meget anvendelig krets
 - Reduserer spenning til å passe applikasjoner
- 2 motstander i serie
 - Mål V mellom dem
 - V_{in} deles tilsvarende motstandene
 - Eks: $V_{in} = 5V + 10k\Omega + 10k\Omega$ 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

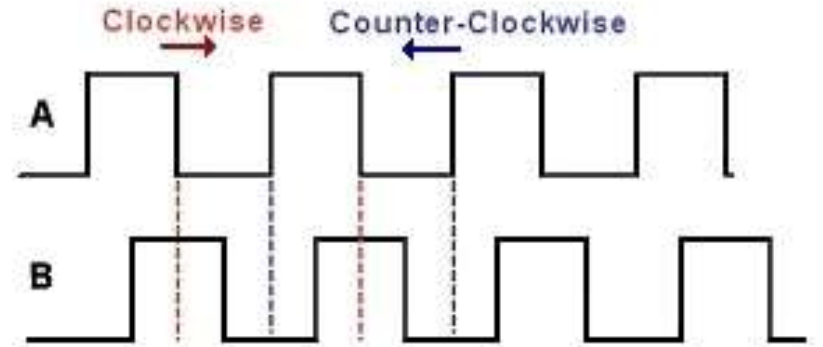


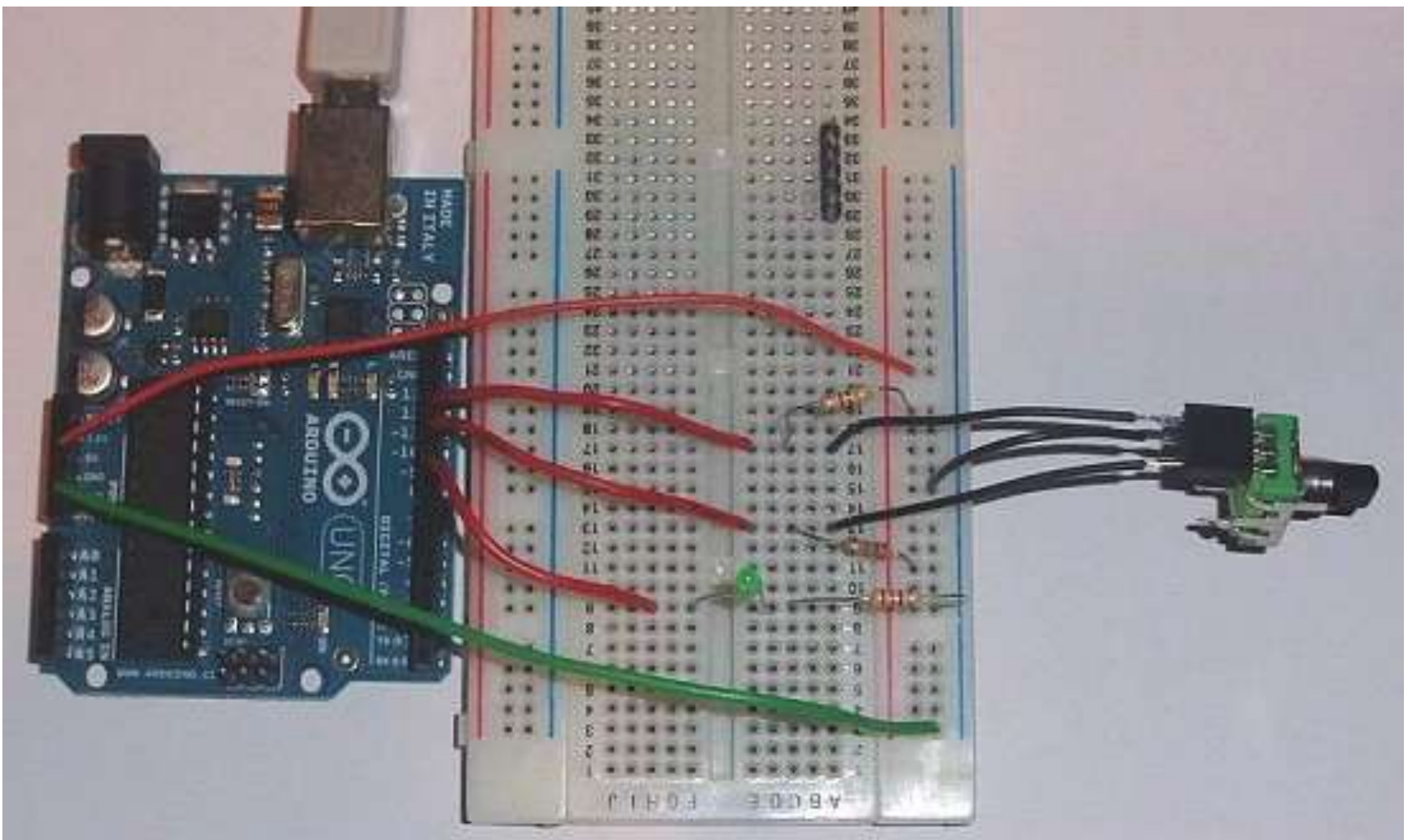


Den magiske grå røyken

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

Neste øvelse: Rotary Encoder





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, osv...

- D'oh!
- FreeRTOS skal visstnok være OK
- Konseptet «State Machine» funker
 - Ressurs og særlig RAM-effektivt
 - FSM library

<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 frequency 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 buttonlin = 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 (buttonlin == 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
      buttonlin = 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....
      buttonlin = 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

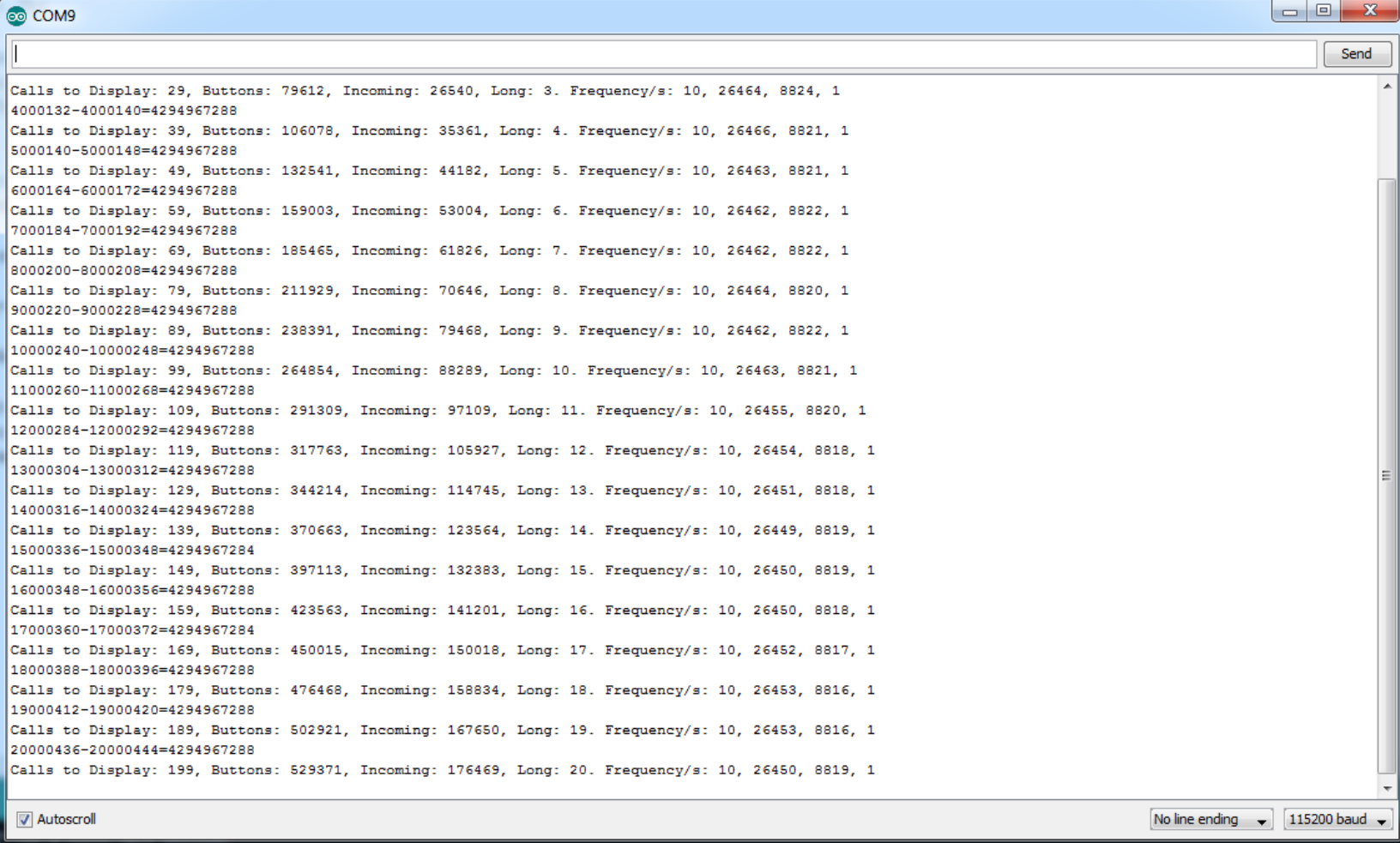
#ifdef DEBUG                                           // #define DEBUG to print statistics to serial
    Serial.print(now);                                // show the timer variables
    Serial.print("-");
    Serial.print(lastlong);
    Serial.print("=");
    Serial.print(now-lastlong);
    Serial.println();

    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

};
```



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>

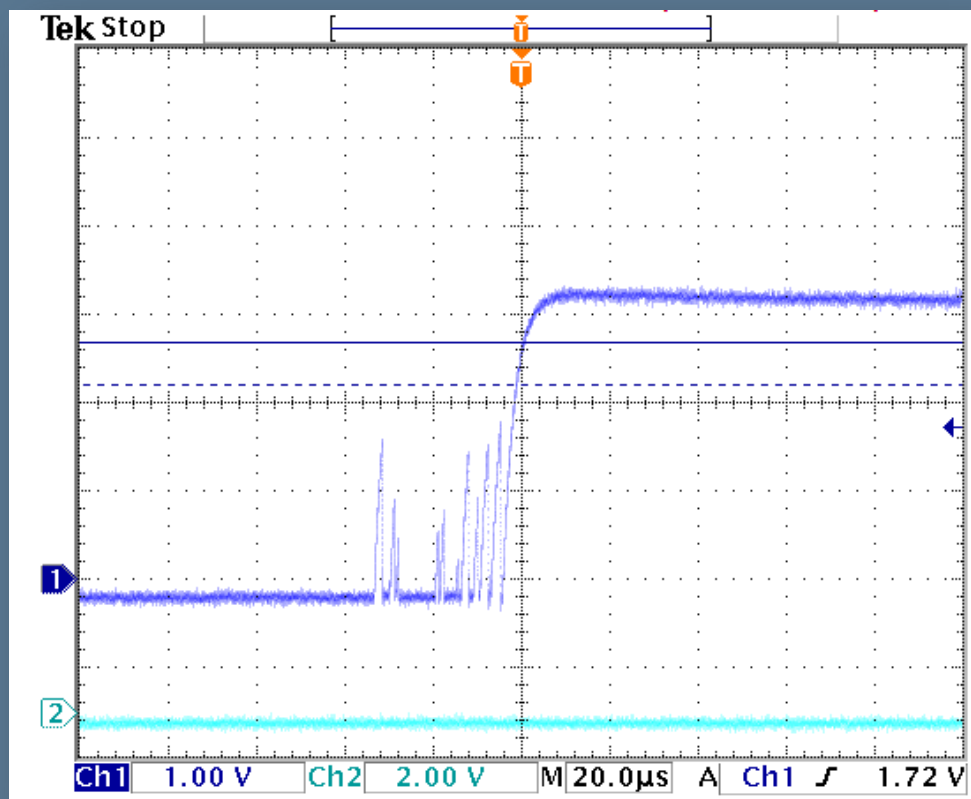
```
In file included from stateMachineDemo.cpp:1:
```

```
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>

Debounce



Hjelp!

- arduino.cc/playground www.freeduino.org
- Kjekke butikker
 - Arduino: Arduino.cc, Robonor, Sparkfun, Adafruit, (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

Andre versjoner

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





HACK A DAY