

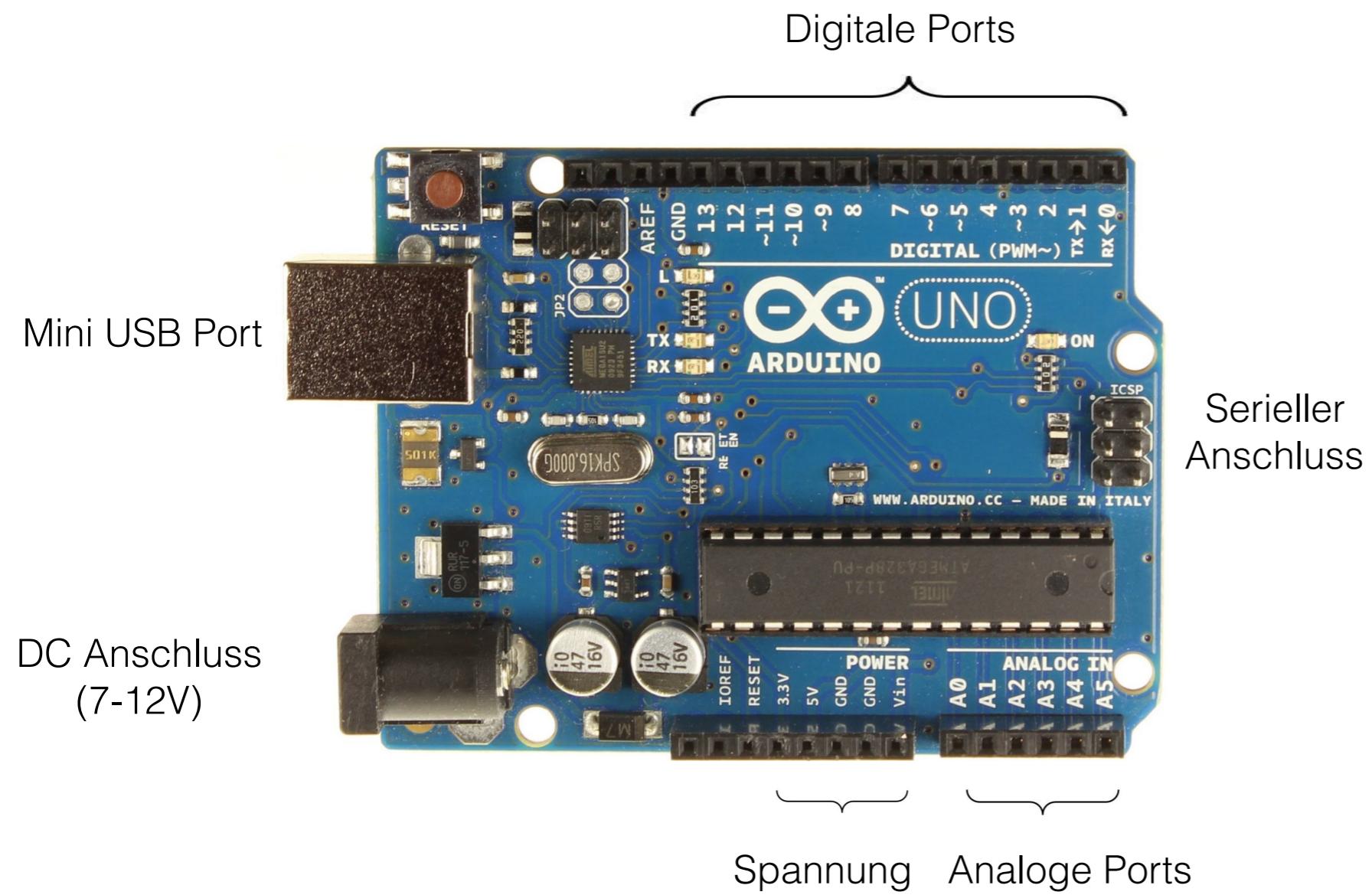
Arduino Crashkurs

30. April 2016
Aerospace Lab

Ablauf des Tages

- Kurze Erklärung zu den Themen
- Mini-Projekt um das gelernte umzusetzen.
- Codegerüste, Schaltpläne und Lösungen auf Github
<https://github.com/khinsberger/ASL-GoKart>

Aufbau der Arduino Platform



Spezifikationen

Mikrocontroller ATMega328

Systemspannung 5 V

Eingangs- spannung 7-12 V

DC Stromstärke I/O 40 mA

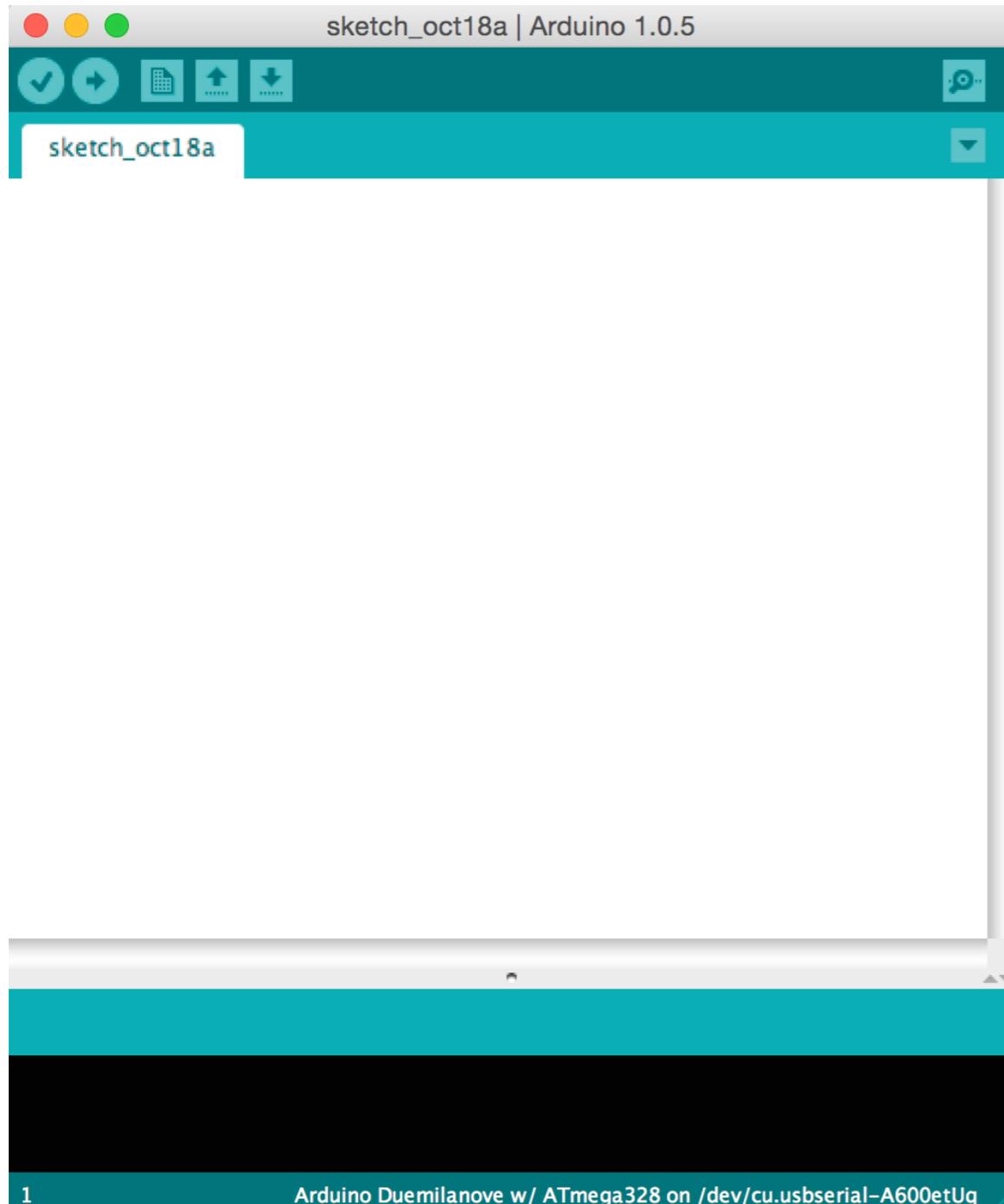
DC Stromstärke 3,3V 50 mA

Flash Speicher 32 kB

SRAM 2 kB

Taktung 16 MHz

Die Arduino Entwicklungsumgebung



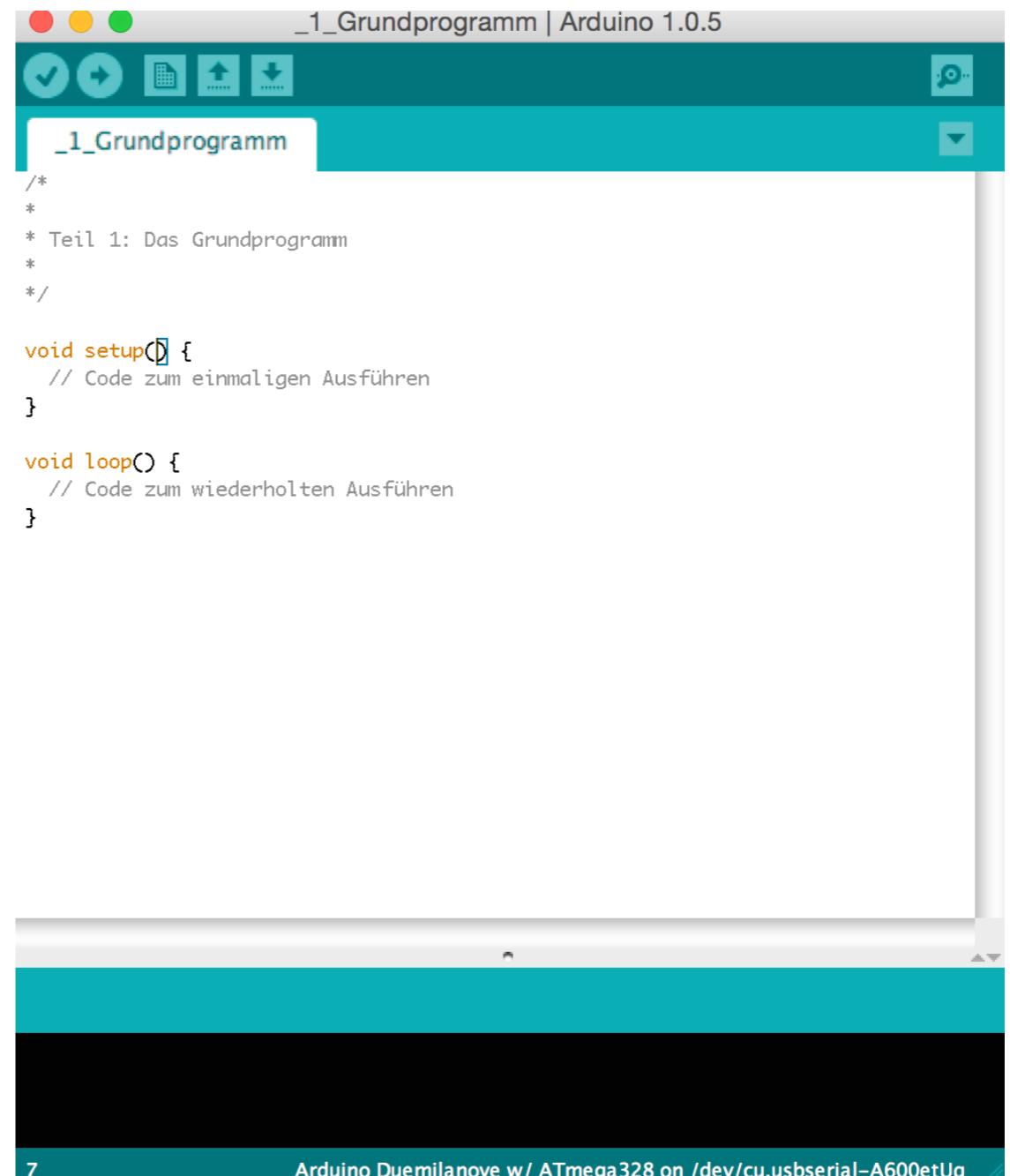
Download der Arduino IDE
<http://arduino.cc/en/Main/Software>

Unter Tools->Board das richtige
Arduino Board einstellen

Unter Tools->Serieller Port den
richtigen Port einstellen

Das Grundprogramm

- **void setup()**: Die **setup()** Funktion wird beim Start aufgerufen. Sie wird nur einmal ausgeführt.
- **void loop()**: Die **loop()** Funktion wird nach dem Ausführen der **setup()** Funktion ausgeführt. Sie wird so lange wiederholt, bis die Stromversorgung getrennt wird.
- Kommentare:
 - // Einzeiliges Kommentar
 - /*
 * Mehrzeiliges Kommentar
 */



The screenshot shows the Arduino IDE interface with the title bar '_1_Grundprogramm | Arduino 1.0.5'. The code editor contains the following code:

```
/*
 *
 * Teil 1: Das Grundprogramm
 *
 */

void setup() {
    // Code zum einmaligen Ausführen
}

void loop() {
    // Code zum wiederholten Ausführen
}
```

The code is displayed in a monospaced font, with comments starting with either a single '/' or a double '/' followed by an asterisk. The Arduino IDE interface includes toolbars and a status bar at the bottom indicating 'Arduino Due milanove w/ ATmega328 on /dev/cu.usbserial-A600etUg'.

Die Arduino Reference

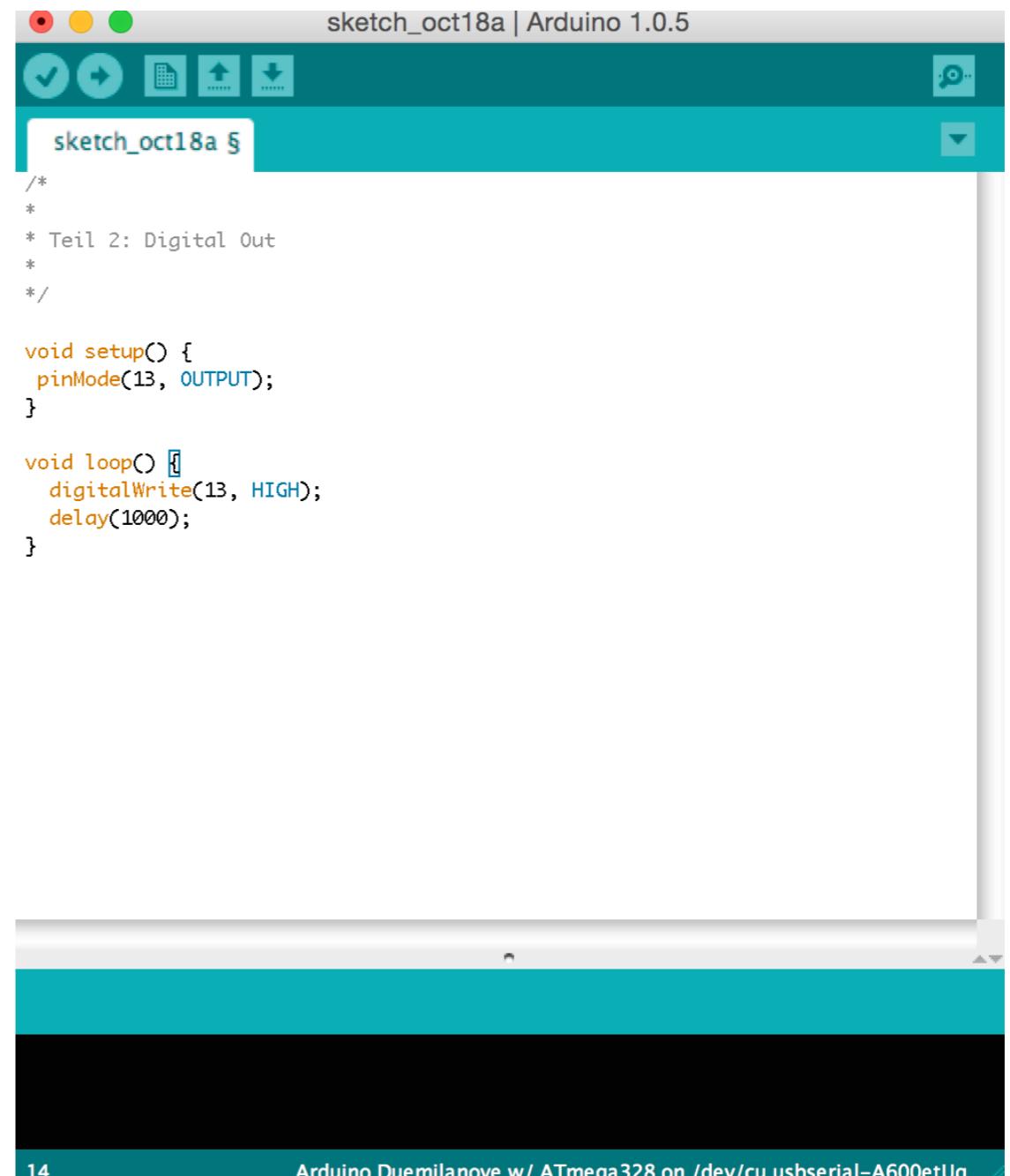
- <http://arduino.cc/en/Reference/HomePage>
- Die Arduino Reference ist eine Bibliothek, auf welcher alle grundlegenden Arduino Befehle erklärt werden.
- Falls ihr nicht mehr wisst wie ein Befehl aussieht oder funktioniert. Benutzt sie!

Mini-Projekt 1

- Lasst das Arduino Grundprogramm auf eurem Arduino Mikrocontroller laufen.

Digitale Ausgabe

- `pinMode(pin, wert)`: Deklariert ob der Pin `INPUT` oder `OUTPUT` ist.
- `digitalWrite(pin, wert)`: „Schreibt“ `HIGH` (5V) oder `LOW` (0V) an den Pin.
- `delay(dauer)`: Pausiert das Programm für die Dauer an Millisekunden



```
sketch_oct18a | Arduino 1.0.5

sketch_oct18a §

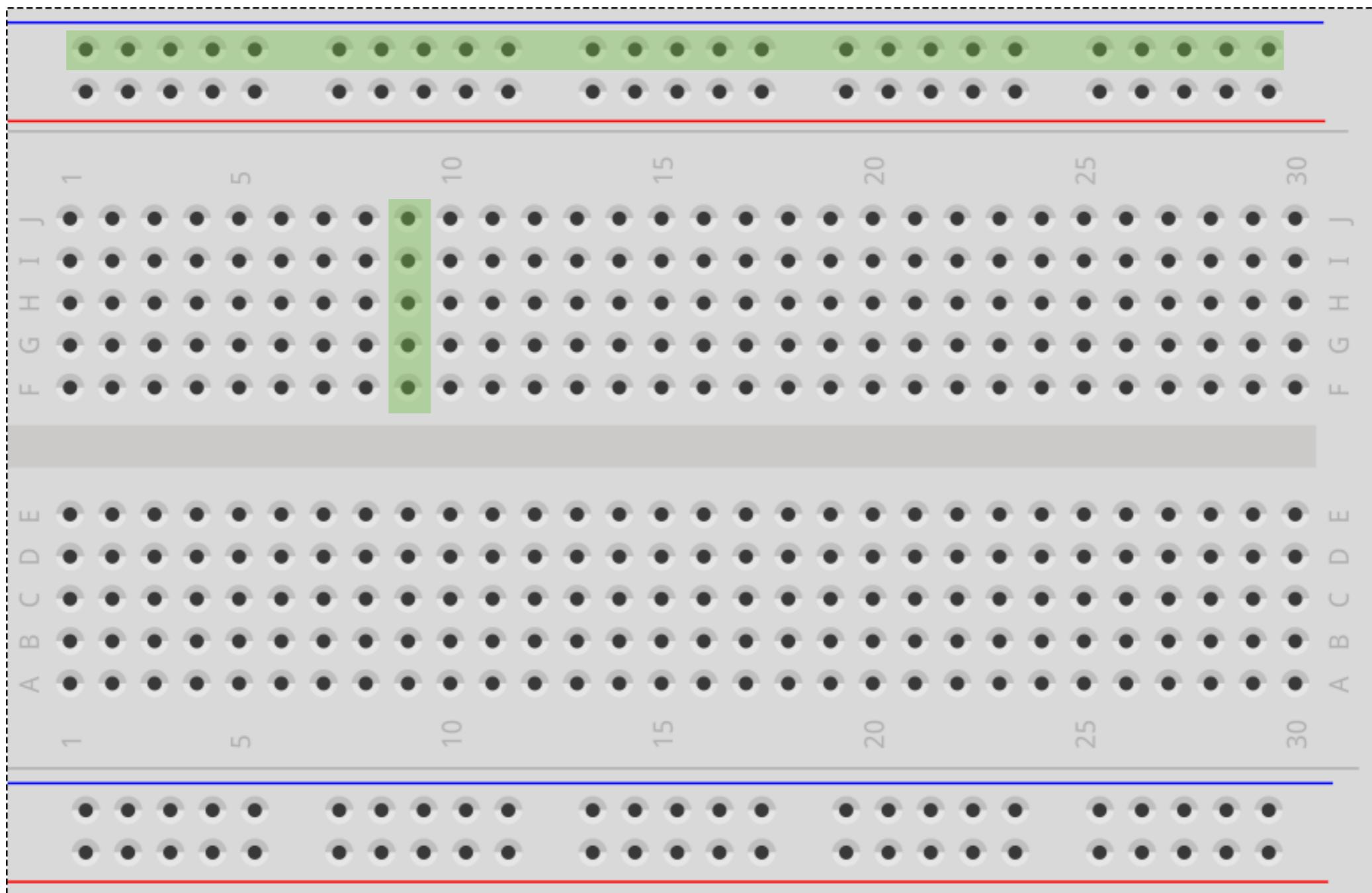
/*
 *
 * Teil 2: Digital Out
 *
 */

void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
}
```

The screenshot shows the Arduino IDE interface with a sketch titled "sketch_oct18a". The code defines a digital output pin (pin 13) and toggles it between HIGH and LOW every 1000 milliseconds. The Arduino board connected is an "Arduino Due" (ATmega328).

Labor-Steckbretter



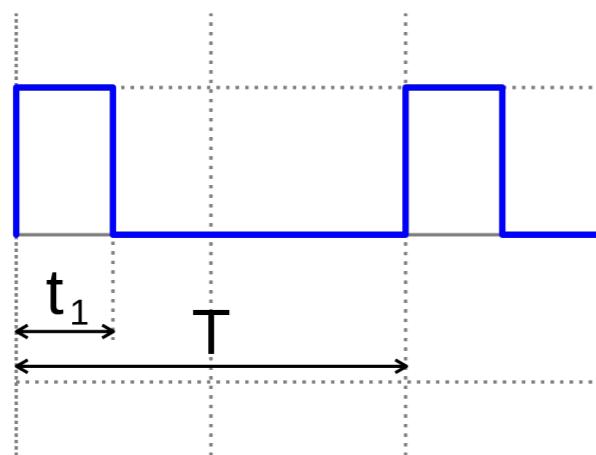
fritzing

Mini-Projekt 2

- Bringt eine LED zum blinken
- Material:
 - LED in beliebiger Farbe
 - Vorwiderstand (330Ω)

PWM

- Die Pulsweitenmodulation (PWM) macht es möglich auch an digitalen Pins, pseudo analoge Werte auszugeben.
- Achtung: Diese Funktion steht nur an digitalen Ausgängen mit „~“ vor der Zahl zur Verfügung.
- `analogWrite(pin, wert)`: Schreibt einen Wert zwischen 0 (0V) und 255 (5V) an den Pin.



sketch_oct18a | Arduino 1.0.5

```
/*
 * 
 * * Teil 3: Analog Out
 * 
 */

void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  analogWrite(13, 255);
}
```

Arduino Duemilanove w/ ATmega328 on /dev/cu.usbserial-A600etUq

The screenshot shows the Arduino IDE interface with a sketch named "sketch_oct18a". The code initializes pin 13 as an output and then uses the `analogWrite` function to set its value to 255 in the loop. The IDE shows the code in a text editor and provides various tools and status information at the top and bottom of the window.

Mini-Projekt 3

- LED eine Sekunde auf voller Helligkeit, eine Sekunde auf mittlerer Helligkeit, eine Sekunde aus.
- Material:
 - LED in beliebiger Farbe
 - Vorwiderstand (330Ω)

Variablen

- Variablen sind dazu da, um Werte zu speichern.
- Globale Variablen werden vor der **setup()** Funktion initialisiert und überdauern das gesamte Programm
- Lokale Variablen können überall initialisiert werden und überdauern nur die Ausführung dieses Blocks (zwischen 2 geschweiften Klammern).
- Variablen werden mit `int x = 0;` oder `int x;` deklariert (und initialisiert).
- Wenn man eine Variable deklariert, welche sich nicht ändern soll benutzt man `const`

boolean	Wahrheitswert	true oder false
char	Buchstabe	a–z und A–Z usw.
byte	Ganzzahl	0 – 255
int	Ganzzahl	-32.768 – 32.767
long	Ganzzahl	-2.147.483.648 – 2.147.483.647
double	Fließkommazahl	Größe 4 Byte

String	Zeichenketten
--------	---------------

Styleguide
Variablen fangen immer mit Kleinbuchstaben an und jedes weitere Wort wird ohne Leerzeichen angehängt und groß geschrieben.
<http://arduino.cc/en/Reference/StyleGuide>

Mini-Projekt 4

- Schreibt das Blinken-Programm so um, dass ihr den Pin nur einmal in eine Variable speichert und dann mit dieser arbeitet und nicht mehr mehr mit der Pinnummer
- Material:
 - LED in beliebiger Farbe
 - Vorwiderstand (330Ω)

while-Schleife

- Die while Schleife wird so lange Wiederholt bis die Bedingung in der Klammer falsch ist.

- Syntax:

```
while (Bedingung) {  
    // Schleifenrumpf  
}
```



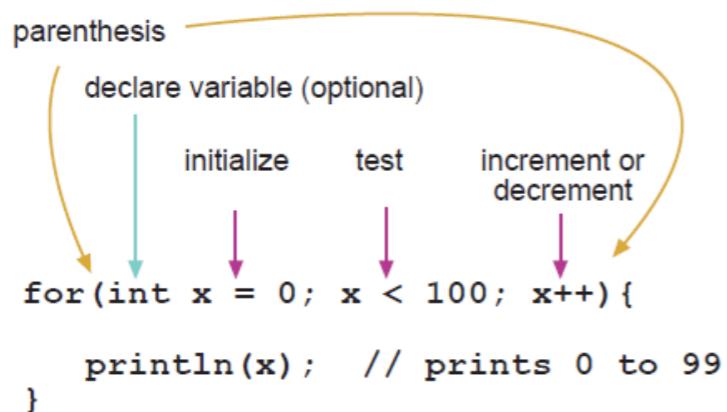
The screenshot shows the Arduino IDE interface with the title bar "sketch_oct18a | Arduino 1.0.5". The toolbar includes icons for file operations, upload, and programmer. The main code editor window contains the following code:

```
/*  
 *  
 * Teil 4: Schleifen  
 *  
 */  
  
void setup() {  
}  
  
void loop() {  
    while(x > 2) {  
        //Mach etwas  
    }  
}
```

The code defines a simple while loop that runs as long as the variable `x` is greater than 2. The loop body contains a comment `//Mach etwas`. The Arduino board dropdown at the bottom right shows "Arduino Due milanove w/ ATmega328 on /dev/cu.usbserial-A600etUg".

for-Schleife

- Die for-Schleife ist nützlich für sich wiederholende Operationen.
- Die Initialisierung wird einmal und als erstes ausgeführt. Dann wird getestet, ob die Bedingung erfüllt ist und der Schleifenrumpf wird ausgeführt. Danach wird die Variable verändert.
- $x++$ ist eine Abkürzung für $x = x + 1$
- $x--$ ist die Abkürzung für $x = x - 1$



A screenshot of the Arduino IDE interface. The title bar says "sketch_oct18a | Arduino 1.0.5". The code editor window contains the following sketch:

```
/*  
 * Teil 4: Schleifen  
 */  
  
void setup() {}  
  
void loop() {  
    for(int x = 0; x <= 100; x++) {  
        //Mach etwas mit x  
    }  
}
```

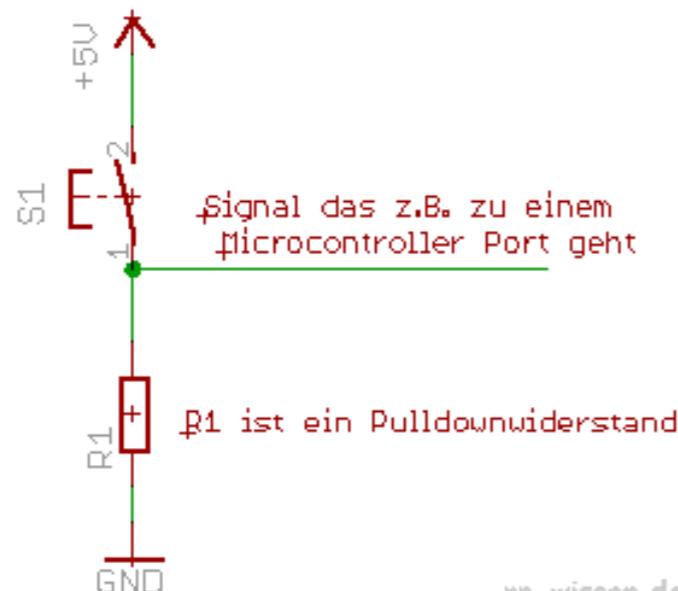
The status bar at the bottom right shows "Arduino Due (ATmega328) on /dev/cu.usbserial-A600etUa".

Mini-Projekt 5

- Dimmt eine LED von dunkel nach hell und wieder nach dunkel
- Material:
 - LED in beliebiger Farbe
 - Vorwiderstand (330Ω)

Digital In

- `digitalRead(pin)`: Liest den Wert von einem Pin, entweder 1 (`HIGH`) oder 0 (`LOW`)
- `Serial.begin(rate)`: Bestimmt die Übertragungsrate für die serielle Kommunikation (Das muss einmal im setup festgelegt werden).
- `Serial.println(wert)`: Schreibt den Wert an den seriellen Monitor (Fenster um die Kommunikation mit dem Arduino zu sehen [Lupe]).



sketch_oct18a | Arduino 1.0.5

```
/*
 * 
 * * Teil 5: Digital In
 * 
 */

void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.println(digitalRead(2));
}
```

12

Arduino Due (ATmega328) on /dev/cu.usbserial-A600etUa

Mini-Projekt 6

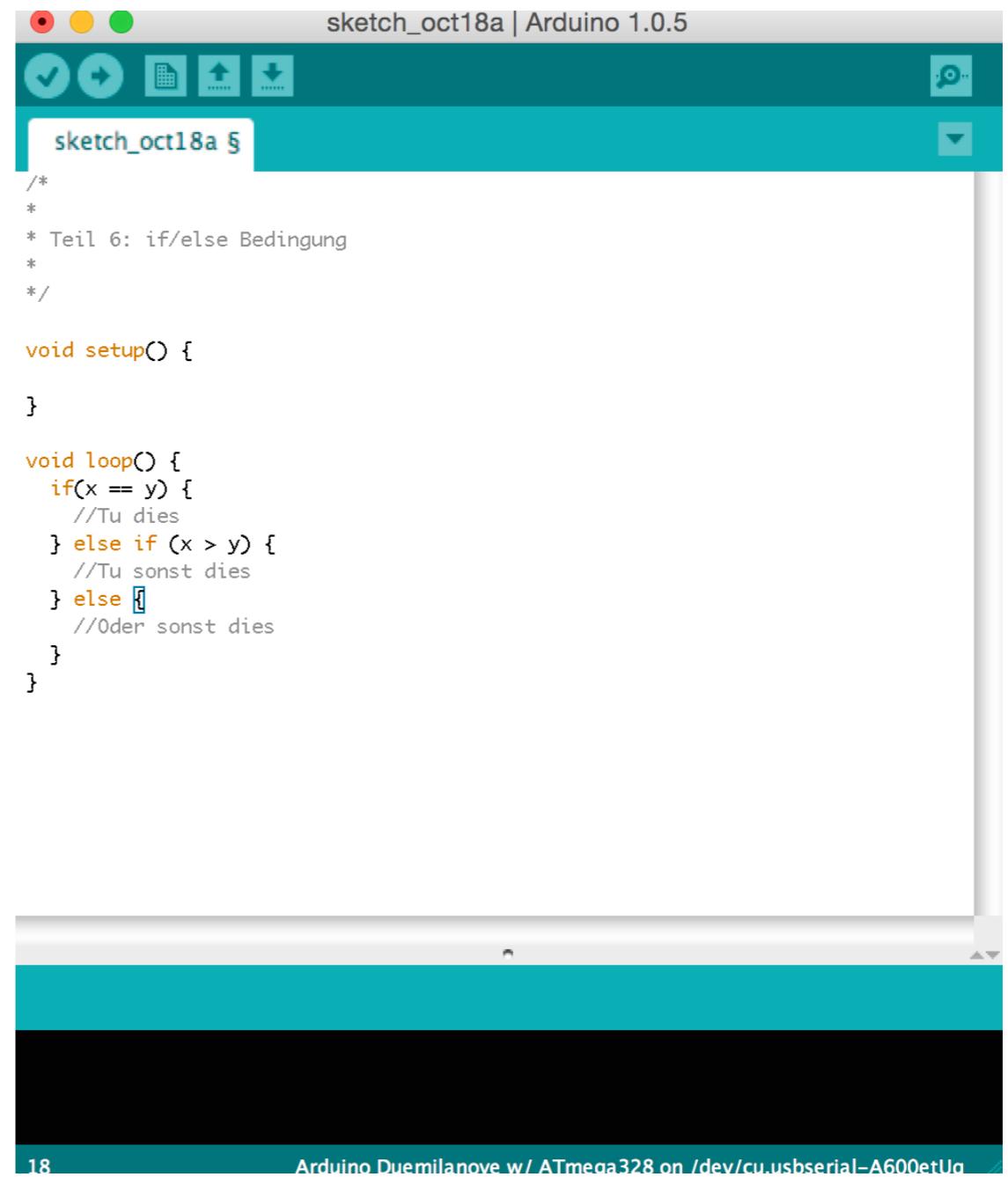
- Lest den Status eines Tasters aus und gibt ihn am Seriellen Monitor aus.
- Material:
 - Taster
 - Pullup-Widerstand ($10k\Omega$)

Verzweigungen

- Die if Bedingung ist nützlich um zu testen ob eine bestimmte Bedingung erfüllt ist.
- Wenn sie nicht erfüllt ist, kann entweder eine weitere Bedingung überprüft werden oder eine sonst Bedingung ausgeführt werden

Operatoren

$x == y$	gleich	&&	und
$x != y$	nicht gleich		
$x < y$	echt kleiner		oder
$x > y$	echt größer		
$x \leq y$	kleiner gleich	!	nicht
$x \geq y$	größer gleich		



The screenshot shows the Arduino IDE interface with a sketch titled "sketch_oct18a". The code is as follows:

```
/*
 * 
 * * Teil 6: if/else Bedingung
 * 
 */

void setup() {
}

void loop() {
    if(x == y) {
        //Tu dies
    } else if (x > y) {
        //Tu sonst dies
    } else {
        //Oder sonst dies
    }
}
```

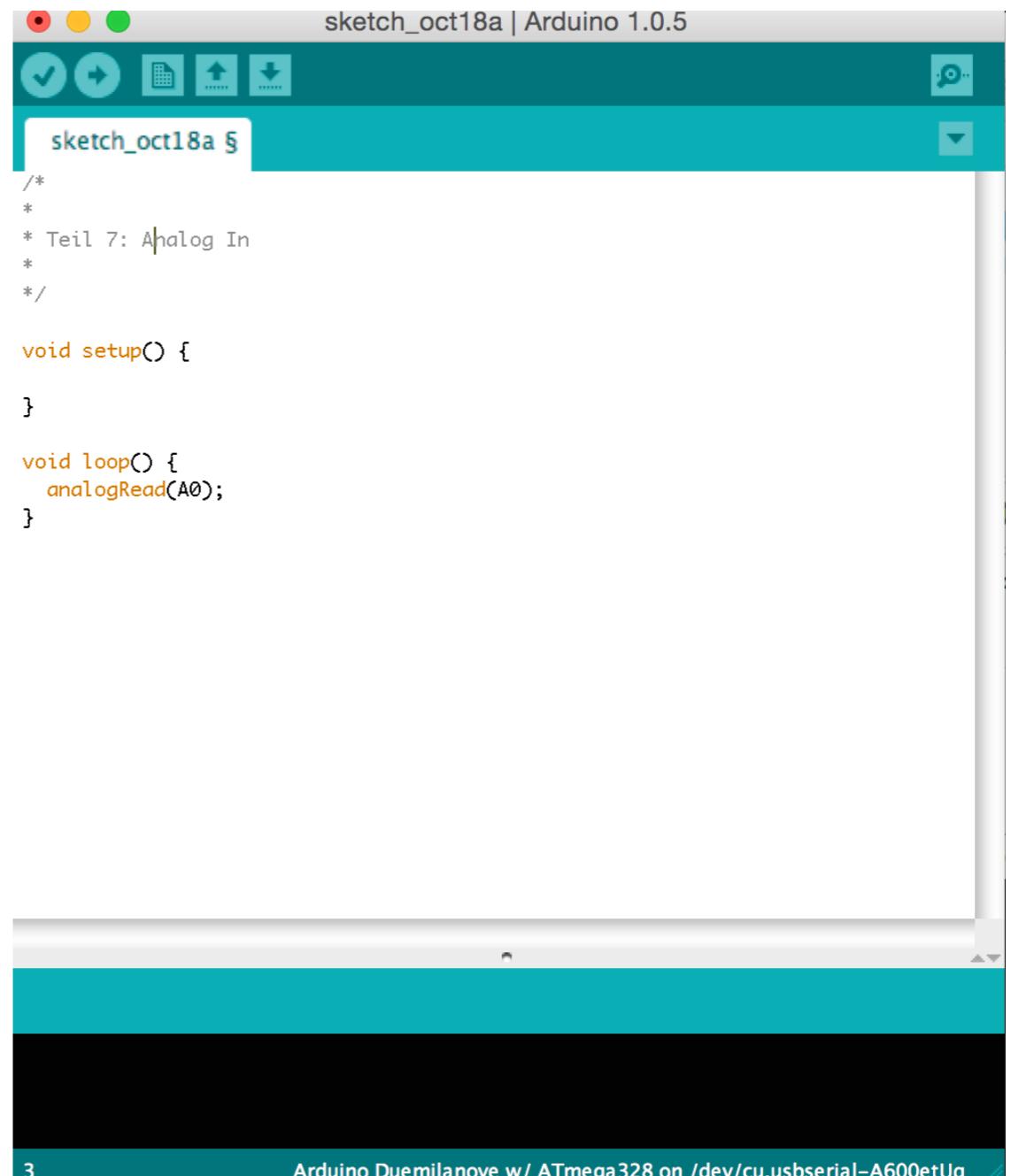
The status bar at the bottom indicates "Arduino Due milanove w/ ATmega328 on /dev/cu.usbserial-A600etUg" and the number "18".

Mini-Projekt 7

- Lest den Status eines Tasters aus und steuert damit eine LED. (Taster gedrückt = LED an)
- Material:
 - LED in beliebiger Farbe
 - Vorwiderstand (330Ω)
 - Taster
 - Pullup-Widerstand ($10k\Omega$)

Analog In

- `analogRead(pin)`: Liest die Spannung, welche an dem Pin anliegt und gibt einen Wert zwischen 0 (0V) und 1023 (5V) aus.
- Es gibt 6 analoge Eingangspins, diese sind mit A0 bis A5 beschriftet.
- Analoge Eingangspins werden mit A0 bis A5 angesprochen.



The screenshot shows the Arduino IDE interface. The title bar reads "sketch_oct18a | Arduino 1.0.5". The code editor contains the following sketch:

```
sketch_oct18a §
/*
 * 
 * * Teil 7: Analog In
 * 
 */
void setup() {
}

void loop() {
    analogRead(A0);
}
```

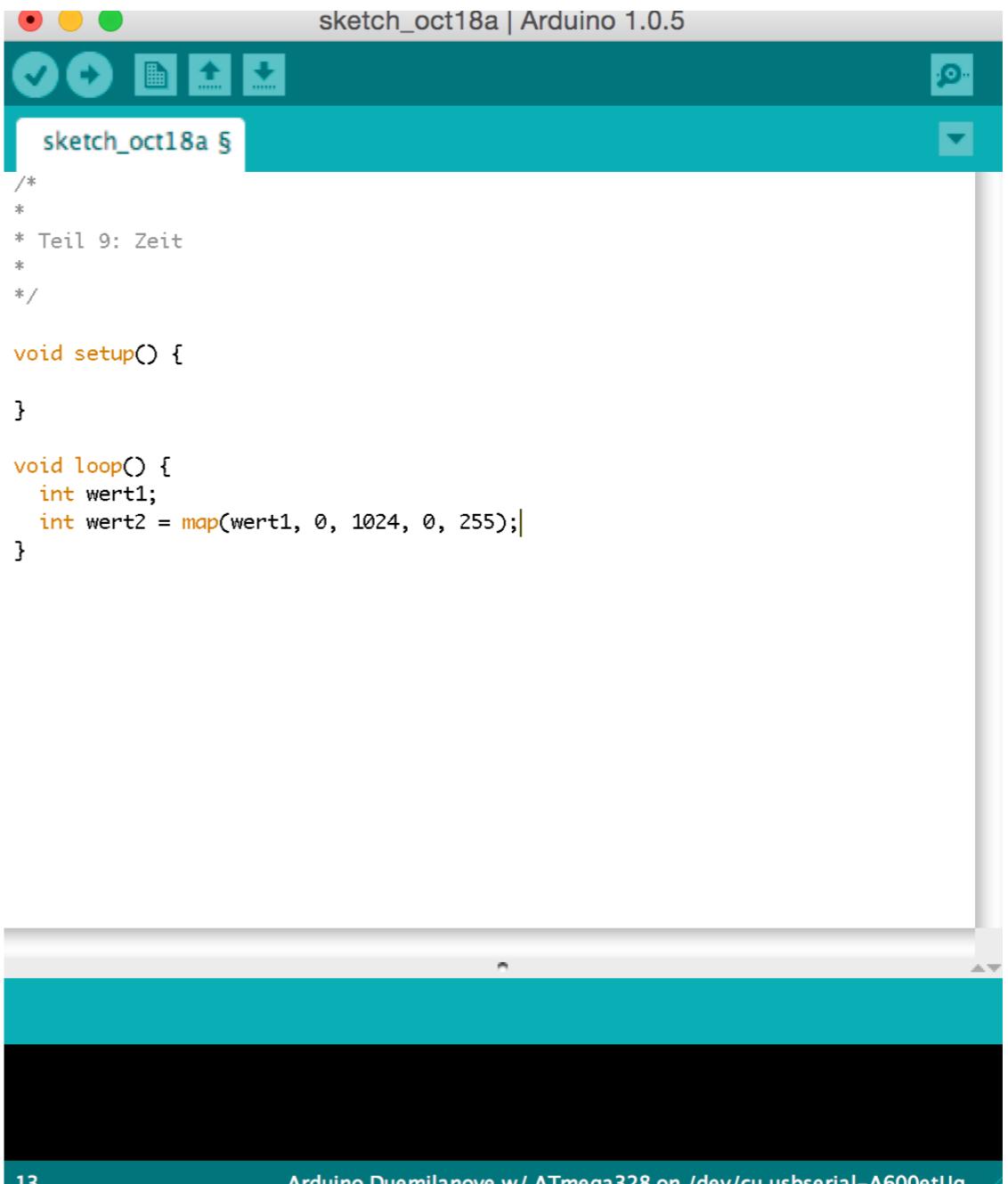
The serial monitor window at the bottom is currently empty, showing only the number "3". The status bar at the bottom right indicates "Arduino Duemillanove w/ ATmega328 on /dev/cu.usbserial-A600etUq".

Mini-Projekt 8

- Lest den Wert eines Potentiometers aus und gebt ihn am Seriellen Monitor aus.
- Material:
 - Potentiometer

Mapping

- Ist dazu da, um einen Wertebereich größer oder kleiner zu zoomen. Zum Beispiel: 0-10 (10 1er Schritte) wird auf 0-100 (100 1er Schritte) vergrößert.
- Maximal- und Minimalwerte dieser Funktion: Analoger Eingangswert zwischen 0 und 1023 aber Ausgangswert zwischen 0 und 255.
- `map(wert, inLow, inHigh, outLow, outHigh)`



The screenshot shows the Arduino IDE interface with a sketch titled "sketch_oct18a". The code uses the `map` function to map an input value from 0 to 1023 to an output value from 0 to 255. The code is as follows:

```
/*
 * 
 * * Teil 9: Zeit
 * 
 */
void setup() {
}

void loop() {
    int wert1;
    int wert2 = map(wert1, 0, 1024, 0, 255);
}
```

The Arduino board connected is an "Arduino Duemilanove w/ ATmega328" and the port is "/dev/cu.usbserial-A600etUa".

Mini-Projekt 9

- Steuert die Helligkeit einer LED mithilfe eines Potentiometers
- Material:
 - LED in beliebiger Farbe
 - Vorwiderstand (330Ω)
 - Potentiometer

Temperatursensoren

- Wir arbeiten mit dem TMP36 Temperatursensor (rote Linie)
- Die Temperatur ist linear abhängig von der Spannung welche am AnalogIn Pin anliegt
- Formel um von der Spannung auf die Temperatur zu schließen:

$$\text{Temp in } \text{C}^\circ = [(V_{\text{out}} \text{ in mV}) - 500] / 10$$

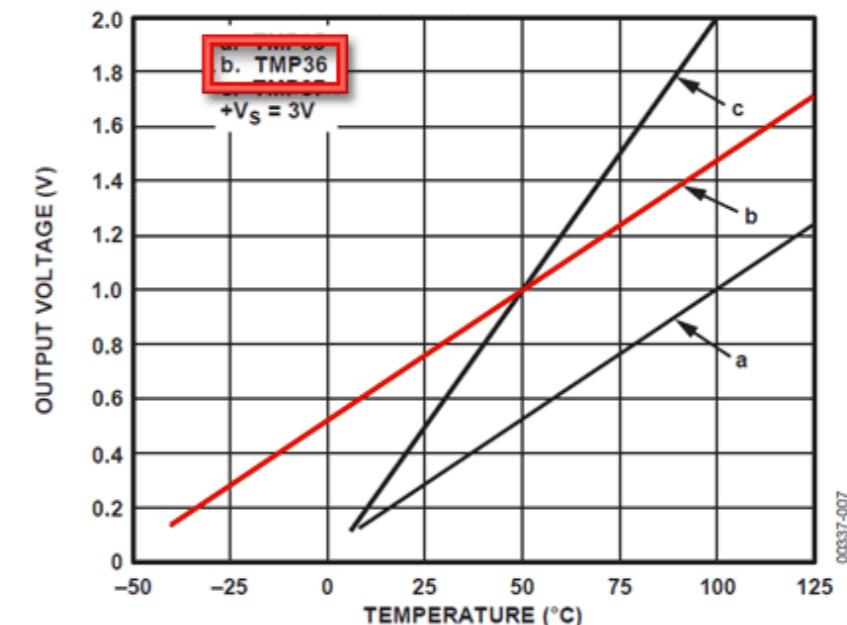
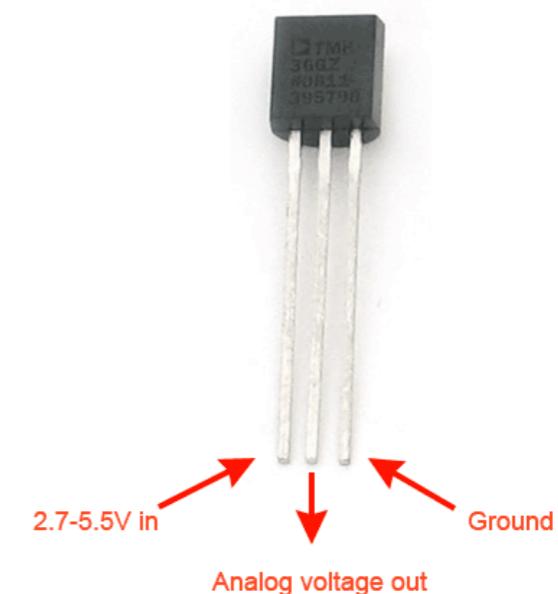


Figure 6. Output Voltage vs. Temperature



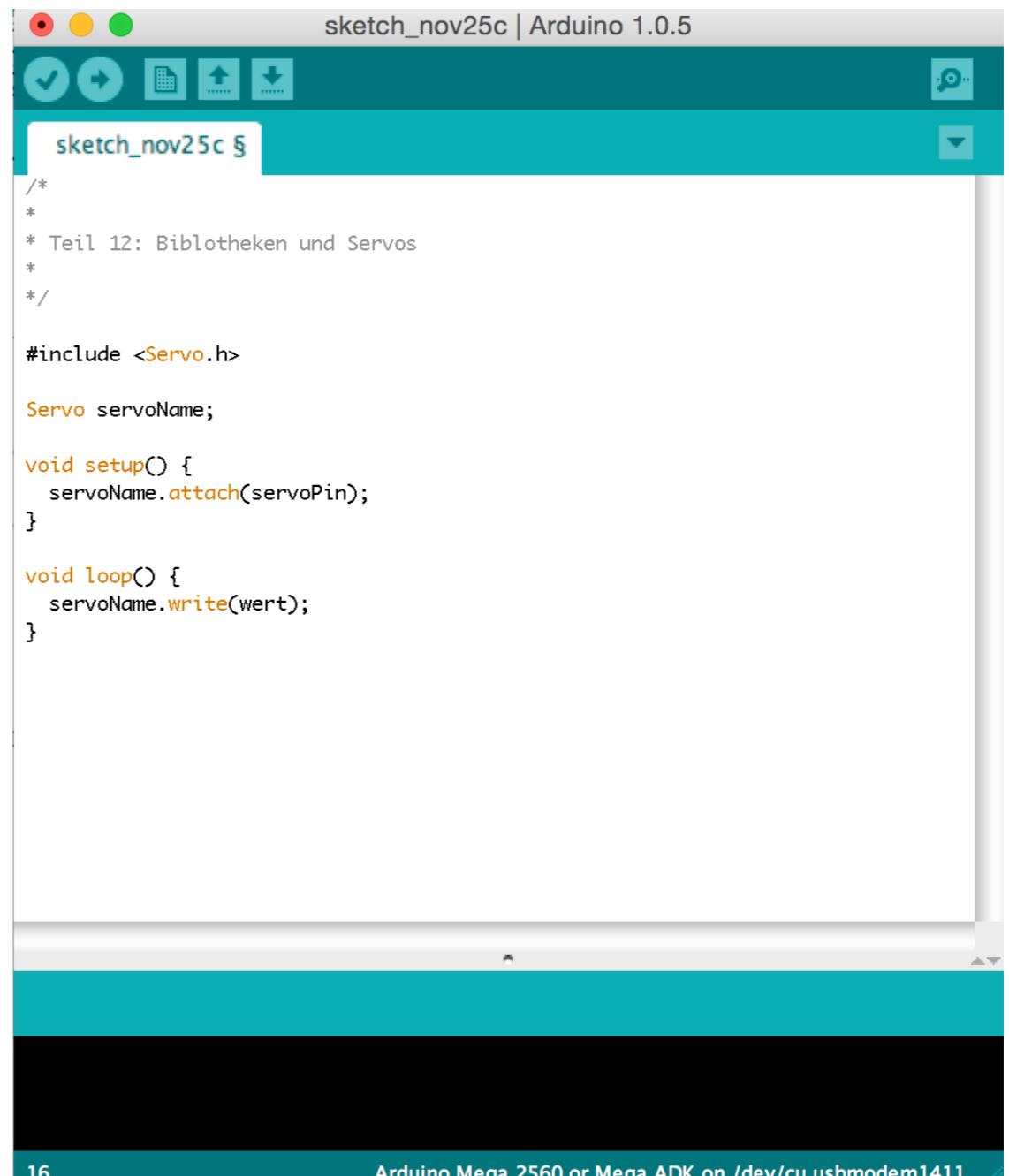
Mini-Projekt 10

- Nutzt einen dem TMP36 Temperatursensor um die Temperatur alle 5 Sekunden auf dem Seriellen Monitor auszugeben. (*Tipp: Berechnet erst die anliegende Spannung am AnalogIn Pin in Millivolt mit Mapping und daraus die Temperatur.*)
- Zusatzaufgabe: Steuert mithilfe des Temperatursensors die Helligkeit einer LED
- Material:
 - TMP36 Temperatursensor

Bibliotheken

Beispiel: Servo

- Schon implementierte Bibliothek, welche die Kommunikation mit einem Servo erlaubt.
- `#include <Servo.h>`: Einfügen der Bibliothek
- `Servo servoName`: Initialisiert den Servo
- `myservo.attach(servopin)`: Verbindet den Servo mit dem servopin (digitaler Pin).
- `myservo.write(wert)`: Verschiebt die Servoauslenkung. Der Wertebereich liegt zwischen 0 und 180, dies entspricht der Auslenkung.



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** sketch_nov25c | Arduino 1.0.5
- Toolbar:** Includes standard icons for file operations (New, Open, Save, Print, etc.).
- Sketch Name:** sketch_nov25c §
- Code Area:** Displays the following C++ code:

```
/*
 * 
 * Teil 12: Bibliotheken und Servos
 *
 */
#include <Servo.h>

Servo servoName;

void setup() {
  servoName.attach(servopin);
}

void loop() {
  servoName.write(wert);
}
```
- Bottom Status Bar:** Shows "16" and "Arduino Mega 2560 or Mega ADK on /dev/cu.usbmodem1411".

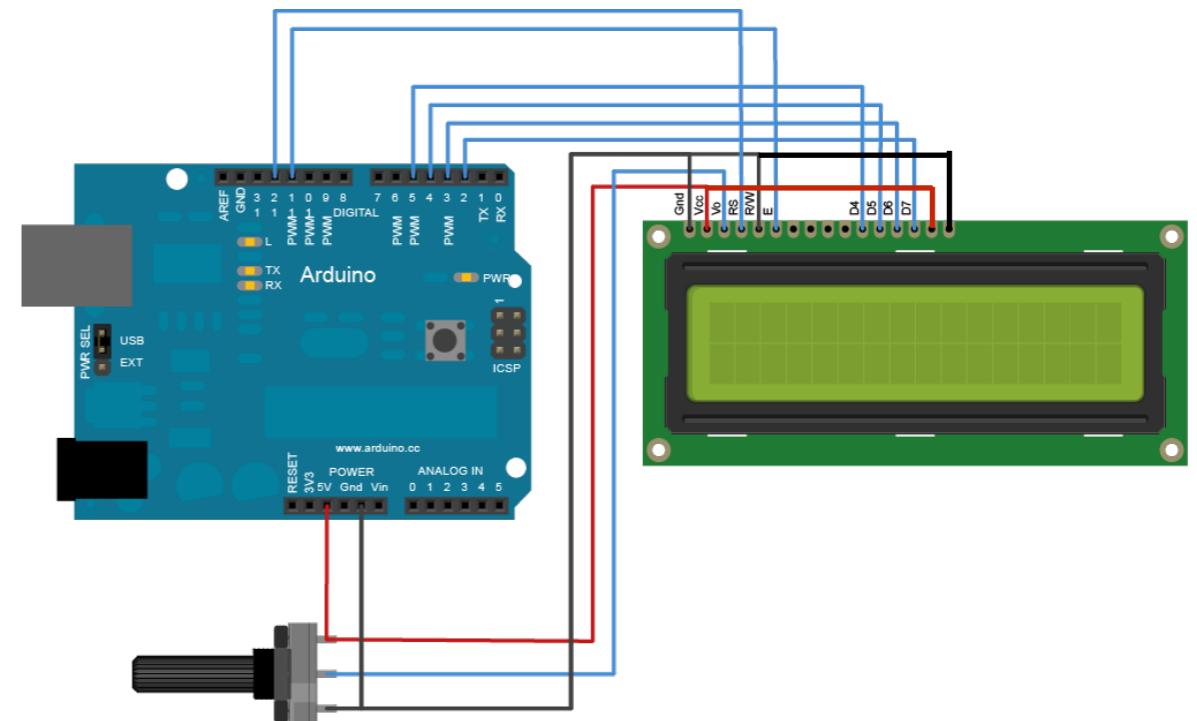
Mini-Projekt 11

- Steuert mit einem Potentiometer einen Servomotor
- Material:
 - Servo Motor
 - Potentiometer

Bibliotheken

Beispiel: LCD

- Schon implementierte Bibliothek, welche die Kommunikation mit einem LCD erlaubt.
 - `#include <LiquidCrystal.h>`: Einfügen der Bibliothek
 - `LiquidCrystal lcd(12,11,5,4,3,2)`: Initialisiert das Display
 - `lcd.begin(reihen, spalten)`: Deklariert die Größe des Displays
 - `lcd.print(wert)`: Gibt den Wert am Display aus.
 - `lcd.setCursor(reihe, spalte)`: Versetzt den Cursor an die bestimmte Position



arduino.cc/en/Tutorial/LiquidCrystal

Mini-Projekt 12

- Schreibt auf einem LCD Display euren Namen und die Zeit, die seit Programmstart vergangen ist.
- Zusatzaufgabe: Gebt die aktuelle Temperatur und die Maximaltemperatur auf dem Display mithilfe eures Temperatursensors aus.
- Material:
 - LCD Display
 - Potentiometer