

# Einführung ARDUINO/FUNDUINO

Prof. Dr.-Ing. Peter Hecker, Dipl.-Ing. Paul Frost, Andreas Dekiert M. Sc.,  
14. Mai 2019

# Agenda

- 09. April Einführung
- 16. April Softwareprojektmanagement
- 23. April Entwicklungstools
- 30. April GitHub
- 07. Mai Software-Dokumentation und Bug-Reporting
- 14. Mai Einführung Arduino**
- 21. Mai **Frei**
- 28. Mai Dateieingabe und -ausgabe
- 4. & 11. Juni **Tag der Lehre und Exkursionswoche**
- 18. Juni Einführung von Qt
- 25. Juni GUI-Erstellung mit Qt
- 02. Juli Serielle Kommunikation
- 09. Juli API-Anleitung und Projektarbeit
- 16. Juli **Vorbereitung der Abgabe und Fragen**
- 12. August 10:00 **Abgabe**

## Einführung ARDUINO/FUNDUINO

Als Teilnehmer soll ich am Ende dieser Übung...

- ☐ die Eigenschaften eines Arduinos kennen
- ☐ die In- und Output-Pins eines Arduinos ansteuern können
- ☐ ein Arduino-Projekt erstellen können

Arduino



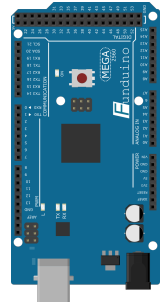
Einführung

Grundlagen

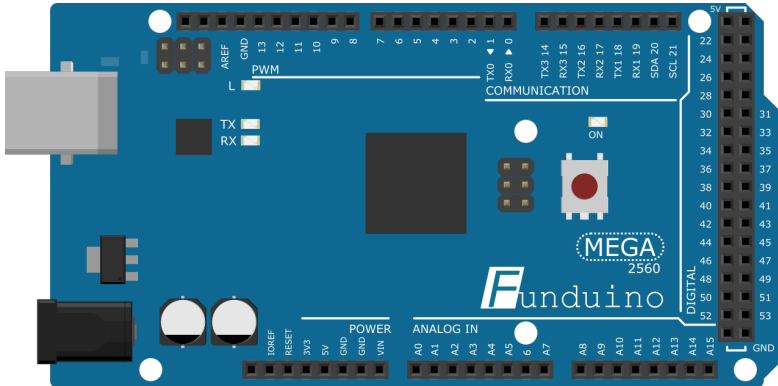
Einstiegs-  
projekt

# Arduino? Funduino?

- Open-Source Elektronik-Plattform
  - Funduinios sind Nachbauten von Arduinos
- Können analoge und digitale Signale empfangen
  - Knopf gedrückt
  - Sensorwerte
  - ...
- Können digitale Signale ausgeben
  - Steuerung von Sensoren
  - LEDs zum Leuchten bringen
  - ...
- Schnelle Prototypenentwicklung



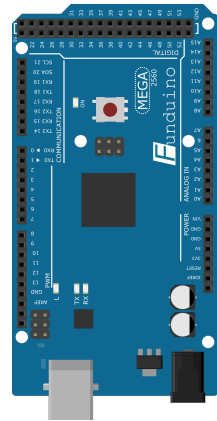
# Funduino Mega 2560



# Funduino Mega 2560

## Technische Daten

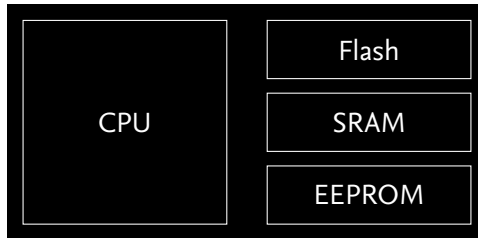
Mikrocontroller	ATmega2560
Betriebsspannung	5V
Eingangsspannung	7-12V
Limit Eingangsspannung	6-20V
Digitale I/O-Pins	54
	davon 15 PWM
Analoge Input-Pins	16
Max. Strom pro I/O-Pin	20 mA
Max. Strom 3,3V-Pin	50 mA
Max. Strom Summe	200 mA
Taktrate	16 MHz



# Funduino Mega 2560

# Speicher Mikrocontroller

Speicherart	Größe	Verwendung
Flash-Speicher	256 kB	Speicherung des Programms 8 kB vom System verwendet
SRAM	8 kB	Speicherung von Variablen <b>Ausschalten entfernt Inhalt</b>
EEPROM	4 kB	Persistenter Speicher <b>Inhalt bleibt erhalten</b>





# Quellen

- VSCode:  
<https://code.visualstudio.com>
- VSCode-Erweiterung PlatformIO:  
<https://platformio.org/get-started>
- Arduino Entwicklungsumgebung (IDE):  
<http://www.arduino.org/downloads>
- Arduino Online-IDE<sup>1</sup>  
<https://create.arduino.cc/>

---

<sup>1</sup>Nicht alle Funktionen werden von der Online-IDE unterstützt

Gibt es Fragen oder Anmerkungen zu dem Unterthema  
**Arduino Einführung?**



# Abgehakt

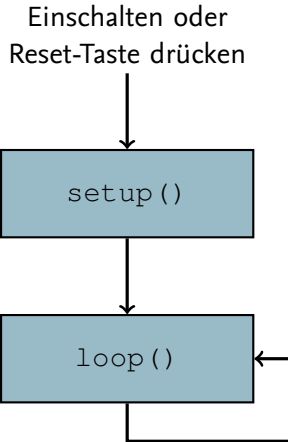
## Einführung ARDUINO/FUNDUINO

Als Teilnehmer soll ich am Ende dieser Übung...



- ☒ die Eigenschaften eines Arduinos kennen
- ☐ die In- und Output-Pins eines Arduinos ansteuern können
- ☐ ein Arduino-Projekt erstellen können

# Programmablauf



## Listing 1: Basisfunktionen

```
#include <Arduino.h>

void setup() {
    // Wird nur einmal
    // ausgeführt
}

void loop() {
    // Wird wiederholt
    // ausgeführt
}
```

# Schaltung erstellen

# Steckplatine

Mit einer Steckplatine können Bauteile temporär verbunden werden.

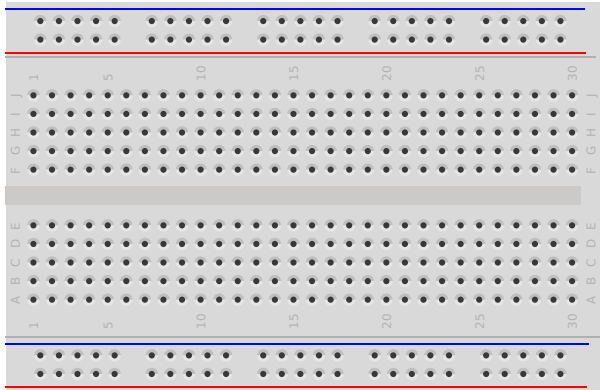


Abbildung 1: Leiterbahnen einer Steckplatine

# Schaltung erstellen

# Steckplatine

Mit einer Steckplatine können Bauteile temporär verbunden werden.

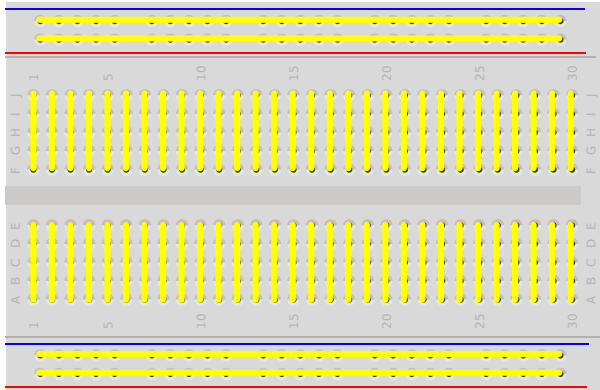


Abbildung 1: Leiterbahnen einer Steckplatine

## Wie kann ich die Pins eines Arduinos nutzen?



# Pin-Setup

Bevor die Pins eines Arduinos genutzt werden können, muss jeweils festgelegt werden, ob diese als Eingangs- oder Ausgang-Pins genutzt werden.

## Listing 2: Pin-Mode Einstellung

```
#include <Arduino.h>

void setup() //Wird als erstes aufgerufen
{
    pinMode(12, OUTPUT);
    // Ab jetzt ist Pin 12 ein Ausgang
    pinMode(4, INPUT);
    // Ab jetzt ist Pin 4 ein Eingang
}

void loop() {    /* ... */ }
```



# Pins verwenden

Nach dem Pin-Setup können die Pins verwendet werden.

## Listing 3: Nutzung der Pins 4 und 12

```
// nach void setup() {...}  
  
void loop() {  
    digitalWrite(12, HIGH); //pin 12 -> 5V  
    digitalWrite(12, LOW); //pin 12 -> 0V  
  
    // Auslesen von Pin 4  
    int wert = digitalRead(4); // HIGH oder LOW  
    int wert = analogRead(4); // 0-1023  
}
```

Wie dimme ich eine LED, wenn ich nur über 5-Volt-Ausgänge verfüge?



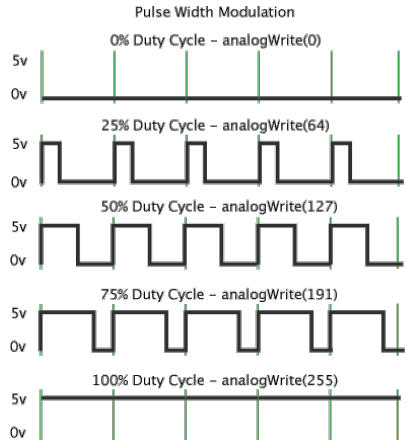
# Analoger Ausgang über PWM

- Problem:

Ausgabe nur als 5 Volt  
Highpegel möglich

→ Generierung eines  
pseudo-analogen Signals über  
eine Pulsweitenmodulation

- Bereich von 0 bis 5 Volt
- PWM-Frequenz von 500 Hz
- 255 Schritte



# Analoger Ausgang über PWM

## Listing 4: Pulsweitenmodulation über analogWrite

```
void setup()
{
    // PIN 12 wird als Ausgangspin gesetzt
    pinMode(12, OUTPUT);
}

void loop()
{
    analogWrite(12, 127); // 2,5 V ueber PWM
    delay(10); //Hauptprogramm wartet 10ms
}
```

# Wie kann ich zuverlässig einen Knopfdruck registrieren?



# Eingangssignal

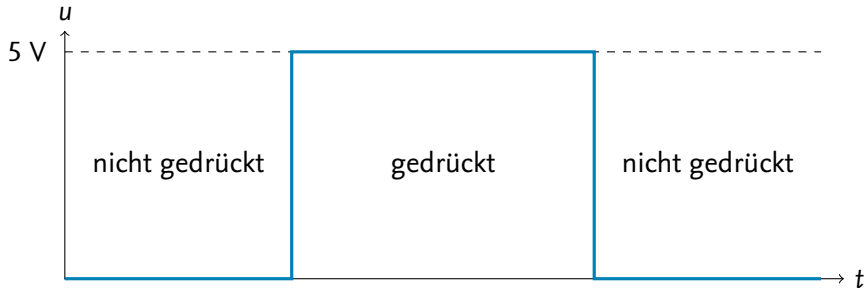
## Externer Pull-Down

### Das Problem:

Der Eingang ist anfällig für Störsignale.

### Die Lösung:

Ein Eingang muss mit einem definierten Potential versehen werden.



# Eingangssignal

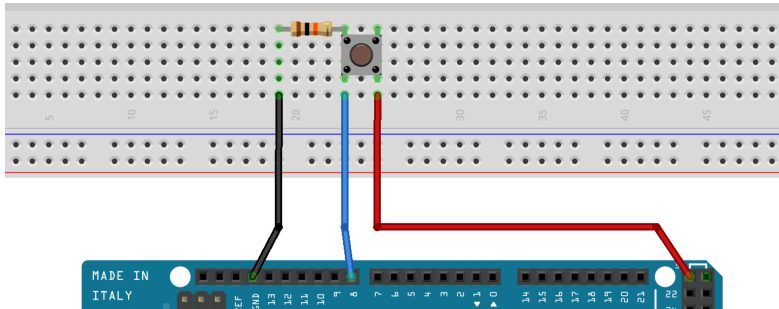
## Externer Pull-Down

### Das Problem:

Der Eingang ist anfällig für Störsignale.

### Die Lösung:

Ein Eingang muss mit einem definierten Potential versehen werden.



# Pull-Down

## Schema

```
pinMode(pin, INPUT);
```

**Zustand Schalter offen:**

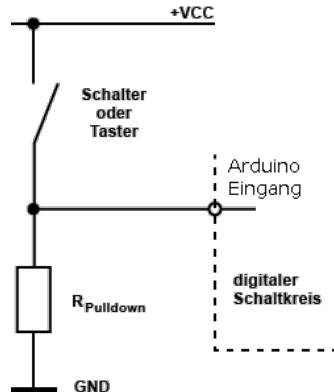
Masse liegt am Eingang an

```
digitalRead(pin) == LOW
```

**Zustand Schalter geschlossen:**

Spannung liegt am Eingang an

```
digitalRead(pin) == HIGH
```





# Pull-Up

## Schema

```
pinMode(pin, INPUT);
```

**Zustand Schalter offen:**

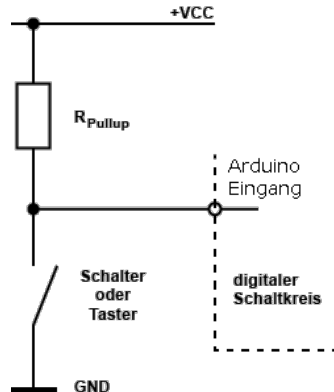
Spannung liegt am Eingang an

```
digitalRead(pin) == HIGH
```

**Zustand Schalter geschlossen:**

Masse liegt am Eingang an

```
digitalRead(pin) == LOW
```



# Interner Pull-Up

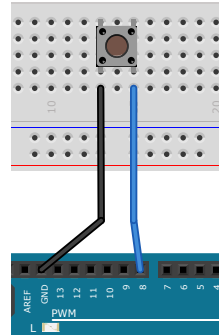
## Beispiel

Der Arduino verfügt bereits über eine interne Pull-Up-Schaltung.

Listing 5: Pull-Up über digitalRead

```
void setup() {
  pinMode(2, INPUT_PULLUP);
}

void loop() {
  if(digitalRead(2) == LOW)
  {
    // Button wurde gedrueckt
    // das wird ausgefuehrt...
  }
}
```



Ich möchte die Werte von Sensoren ausgeben und wissen, welche Programmteile gerade ausgeführt werden.



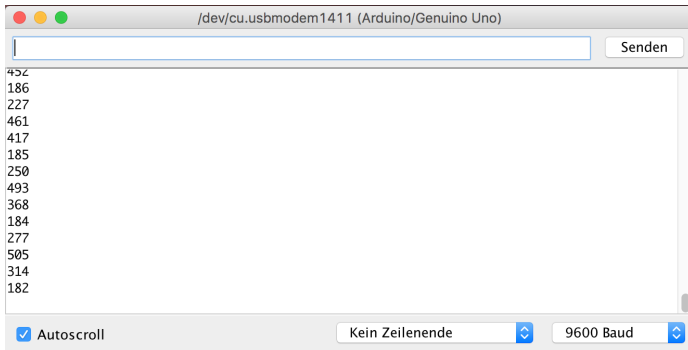
## Listing 6: Serielle Ausgabe

```
void setup() {  
    Serial.begin(9600); // Initialisierung  
}  
  
void loop() {  
    int sensorValue = analogRead(A0);  
    Serial.println(sensorValue); // Ausgabe  
    delay(1);  
}
```

# Serielle Kommunikation

# Serieller Monitor

```
int sensorValue = analogRead(A0);  
Serial.println(sensorValue); // Ausgabe
```



# Gibt es Fragen oder Anmerkungen zu dem Unterthema **Arduino Grundlagen?**



# Abgehakt

## Einführung ARDUINO/FUNDUINO

Als Teilnehmer soll ich am Ende dieser Übung...

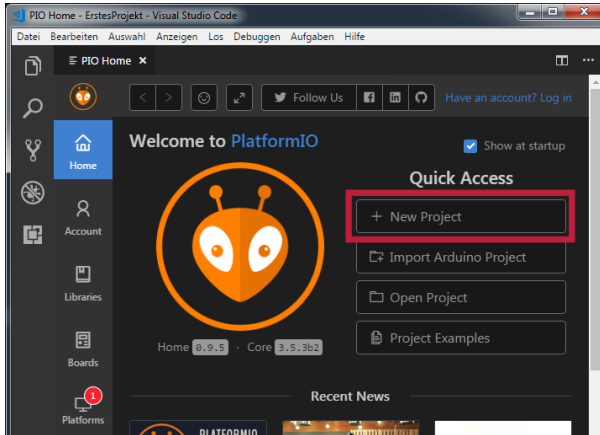
- ☒ die Eigenschaften eines Arduinos kennen
- ☒ die In- und Output-Pins eines Arduinos ansteuern können
- ☐ ein Arduino-Projekt erstellen können

# Programmieren mit einem Arduino

1. Entwicklung
  - Programmierung
  - Schaltung erstellen
2. Programm kompilieren
3. Programm auf Arduino/Funduino hochladen
4. Programm testen
5. Programm und Schaltung für die Anleitung dokumentieren



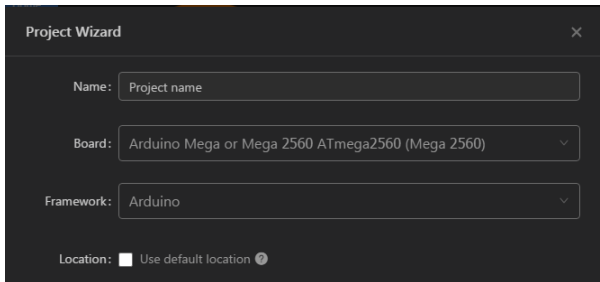
# IDE-Einstellungen



# IDE-Einstellungen

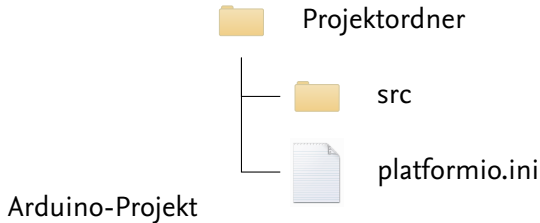
## Wichtig

Bei den Funduinos muss der Arduino Mega 2560 ausgewählt werden.



# Organisation von Arduino-Projekten

Die `platformio.ini` Datei enthält die Projekteinstellungen für das



# Programmierung

# Programm

```
void setup() //Wir starten mit dem Setup
{
    pinMode(12, OUTPUT); // Pin 12 ist ein Ausgang
    pinMode(4, OUTPUT); // Pin 4 ist ein Ausgang
}

void loop() // Das Hauptprogramm beginnt.
{
    digitalWrite(12, HIGH); // Pin12 einschalten.
    delay(1000);
    digitalWrite(12, LOW); // Pin12 ausschalten.
    digitalWrite(4, HIGH); // Pin4 einschalten.
    delay(1000);
    digitalWrite(4, LOW); // Pin4 ausschalten.
}
```

# Schaltung erstellen

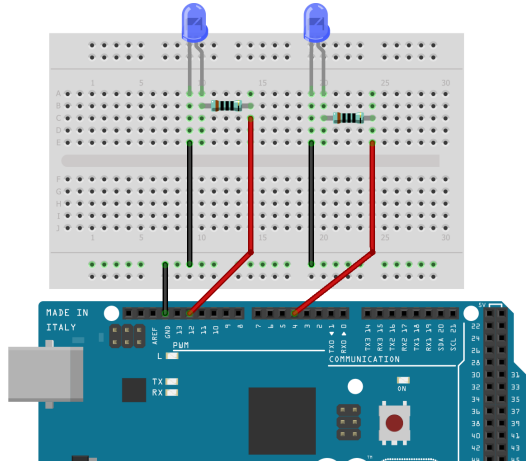
Bauteile:

2 x Widerstand 100  $\Omega$

2 x LED blau



LEDs **NIE** ohne Widerstand verwenden!



# Programm hochladen

- Über den Bootloader können Sketche direkt auf den Mikrocontroller geladen werden
- Erforderliche Schritte:
  1. Arduino per USB anschließen
  2. Programm kompilieren und hochladen



Abbildung 2: VSCode Statusleiste, von links: kompilieren, hochladen

# Weiteres Vorgehen

Wenn ich einen Sensor oder ein anderes Bauteil nutzen möchte:

- Lesen der Dokumentation  
Repository API-Materialien auf GITHUB
- Beispiele nutzen  
Beispielprojekte auf arduino.cc  
Funduino-Beispiele und -Anleitungen

Schritte zum Lösen eines Problems:

1. Fehlermeldungen und Warnungen beachten
2. <https://www.google.de>
3. <http://stackoverflow.com/>  
mit dem Tag [arduino] nach Lösungen suchen
4. Beitrag mit Fehlermeldung und Kontext in SLACK verfassen

Gibt es Fragen oder Anmerkungen zu dem Thema  
**Arduino?**





# Abgehakt

## Einführung ARDUINO/FUNDUINO

Als Teilnehmer soll ich am Ende dieser Übung...

- ☒ die Eigenschaften eines Arduinos kennen
- ☒ die In- und Output-Pins eines Arduinos ansteuern können
- ☒ ein Arduino-Projekt erstellen können

Jetzt besteht die Möglichkeit, das Sprintmeeting durchzuführen.

Protokolliert bitte

- die bearbeiteten Aufgaben der Vorwoche.
- die Zwischenstände der geplanten Aufgaben.
- die in der kommenden Woche zu bearbeitenden Aufgaben.

# Ende

Vielen Dank für eure Aufmerksamkeit!