

Einführung ARDUINO/FUNDUINO

Prof. Dr.-Ing. Peter Hecker, Dipl.-Ing. Paul Frost, Andreas Dekiert M. Sc.,
08. Mai 2018

Agenda

- 03. April Einführung
- 10. April Softwareprojektmanagement
- 17. April Entwicklungstools
- 24. April GitHub
- 08. Mai Einführung Arduino/Fundduino**
- 15. Mai Dateieingabe und -ausgabe
- 22. Mai Exkursionswoche
- 29. Mai Dokumentation und Bug-Reporting
- 05. Juni Einführung von Qt
- 12. Juni GUI-Erstellung mit Qt
- 19. Juni Anleitung erstellen
- 26. Juni Projektarbeit
- 03. Juli Vorbereitung der Abgabe
- 10. Juli Abgabe

Themen der letzten Woche:

- Zielkriterien und Aufgaben
 - Was sind Aufgaben?
- Sprint
 - Unterschied Projektstatus und Aufgabenstatus?



Aufgaben der Vorwoche

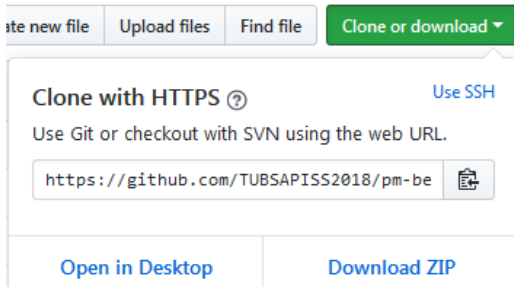
Projekt-Code-Repo

1. Klont das Code-Repository.
2. Optional: Mit der Programmierung beginnen.

Projekt-Wiki-Repo

1. Wie können Bilder in das Projekt-Wiki hochgeladen werden?
2. Fügt das Projektschema in das Projekt-Wiki hinzu.
3. Überträgt sämtliche Punkte für die Projektmappe in das Projekt-Wiki.

- Das Code-Repository befindet sich auf der Projekt-Hauptseite
<https://github.com/TUBSAPISS2018/pm-...>



- Über GITHUB DESKTOP klonen.


Clone a repository

GitHub.com

Enterprise

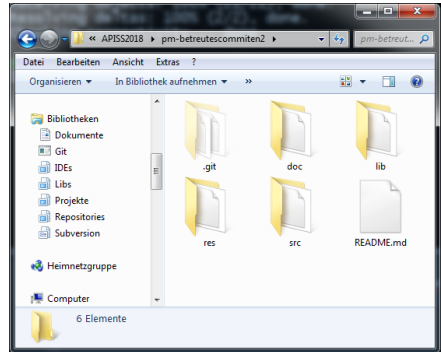
URL

TUBSAPISS2018

 TUBSAPISS2018/pm-betreutescommitten2

Local path

- `src`-Ordner
Quelltext und Projektdateien
- `doc`-Ordner
ggf. Projektschema, Anleitung, ...
- `lib`-Ordner
Externe Bibliotheken
- `res`-Ordner
Projektressourcen
z.B.: Bilder für die Software





- Bild in das Wiki-Repository sichern

In unserem Fall wird das Bild `sidebarheader.png` in den Ordner `img` gesichert

- Bild in gewünschter Wiki-Seite hinzufügen (hier: `_Sidebar.md`):

`[[Bildpfad | Alternativtext]]`

Beispiel: `[[/img/sidebarheader.png | Sidebar_header]]`

- Den neuen Stand über **git add** und **git commit** versionieren
- Die Änderungen über **git push** hochladen

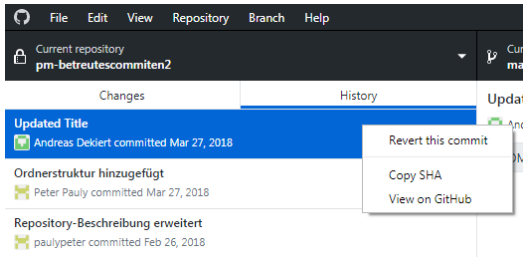
Aufgaben

Angabe von Commits im Wiki-Repo

Damit nur die besten 10 Commits bewertet werden, soll jedes Gruppenmitglied seine 10 Commits im Wiki referenzieren.

Commits referenzieren

1. Hash-Code kopieren



2. Hash-Code auf der Commit-Wiki-Seite eintragen:

Andreas Dekiert:

* 7f8424bf09d646b852ee4e2eec0e1d21b65a843b

Sprintplanung

| To Do | Doing | Done |
|--------------------------------------------------|-------------------------------|---------------------------------------------------------------|
| <div>USE KARDAN</div> <div>GET KARDAN TUGA</div> | <div>LEARN ABOUT KARDAN</div> | <div>GET SOME STICKS NOTES</div> <div>GET A WHITE BOARD</div> |

Beispiel-
projekt

Aufgaben-
generierung

Abhängig-
keiten

Beispielprojekt

Rahmenbedingungen

IFF: 55+ Mitarbeiter

250+ Kaffeebezüge pro Woche

Zwei Preise: Kaffee, Kaffeespezialität

Abrechnung erfolgt per Strichliste

Grobe Idee

Elektronisches Abrechnungssystem als Ersatz für die Strichliste.



User-Story und Zielkriterien

User-Story

Als Mitarbeiter möchte ich meinen Kaffee automatisch über das Abrechnungssystem bezahlen können, damit ich schneller den Kaffee beziehen kann.

4 Zielkriterien

1. Der Mitarbeiter wird anhand seiner Kaffeetasche identifiziert
2. Falls der Mitarbeiter genug Guthaben hat, werden ihm die Kaffeeoptionen angezeigt
3. Nachdem der Kaffee bezogen wurde, werden die Kosten für den Kaffee von dem Guthaben abgezogen
4. Der Restbetrag wird dem Mitarbeiter angezeigt

Aufgabengenerierung

ZIELKRITERIUM 1

Jedes Zielkriterium kann weiter heruntergebrochen werden zu Aufgaben, die Arbeitsschritte beschreiben, bis das Zielkriterium erfüllt wird.

Ein Teammitglied zeichnet sich verantwortlich, die Bearbeitung der Aufgaben zu überwachen.

Der Mitarbeiter wird anhand seiner Kaffeetasse identifiziert *Marc*

- 1.1 RFID-Sensor an Funduino einbinden und darüber auslesen
- 1.2 RFID-Tag erkennen
- 1.3 Eine RFID-Tag-Mitarbeiterzuordnung über eine Liste erstellen
- 1.4 RFID-Tag an Tasse anbringen und testen, ob es erkannt wird

Aufgabengenerierung

ZIELKRITERIUM 2

Die Anzahl der generierten Aufgaben ist frei wählbar.

Falls der Mitarbeiter genug Guthaben hat, werden ihm die Kaffeeoptionen angezeigt

Peter

- 2.1 Mitarbeiterliste mit Guthaben erstellen
- 2.2 Mitarbeiter in Liste finden
- 2.3 Guthaben des ausgelesenen Mitarbeiters bestimmen
- 2.4 Guthaben mit Preisen der Kaffeeoptionen vergleichen
- 2.5 Erlaubte Kaffeeoptionen per Debugausgabe anzeigen

Aufgabengenerierung

ZIELKRITERIUM 3

Der Arbeitsaufwand je Aufgabe ist frei wählbar.

Nachdem der Kaffee bezogen wurde, werden die Kosten für den Kaffee von dem Guthaben abgezogen *Andreas*

- 3.1 Preis der gewählten Kaffeeoption ermitteln
- 3.2 Neues Guthaben ermitteln
- 3.3 Guthaben belasten
- 3.4 Guthaben in Liste bei dem ausgelesenen Mitarbeiter aktualisieren

Aufgabengenerierung

ZIELKRITERIUM 4

Eine ungenaue Planung kann den tatsächlichen Aufwand verbergen.

Der Restbetrag wird dem Mitarbeiter angezeigt

Paul

4.1 Restbetrag auf Display anzeigen

4.2 Nach 10 Sekunden Display leeren

Verborgener Mehraufwand:

Die erste Aufgabe erfordert zunächst die Hardwareinstallation und -Anbindung sowie softwareseitige Initialisierung des Displays.

Abhängigkeiten zwischen den Aufgaben

Nicht *alle* generierten Aufgaben können unabhängig voneinander bearbeitet werden.

→ Abhängigkeiten müssen festgehalten werden.

Beispiel: Mehrere Aufgaben setzen eine Mitarbeiterliste voraus.

- 2.1 „Mitarbeiterliste mit Guthaben erstellen“ ist Voraussetzung für:
 - 1.3 Eine RFID-Tag-Mitarbeiterzuordnung über eine Liste erstellen
 - 2.2 Mitarbeiter in Liste finden
 - 3.4 Guthaben in Liste bei dem ausgelesenen Mitarbeiter aktualisieren

Abhängigkeiten zwischen den Aufgaben

Mögliche Art der Darstellung:

| Aufgabe | Setzt voraus |
|---------|--------------|
| 1.1 | - |
| 1.2 | 1.1 |
| 1.3 | 2.1 |
| 1.4 | 1.2, 1.3 |
| 2.1 | - |
| 2.2 | 2.1 |
| 2.3 | 2.2 |
| 2.4 | 2.3 |
| 2.5 | 2.4 |

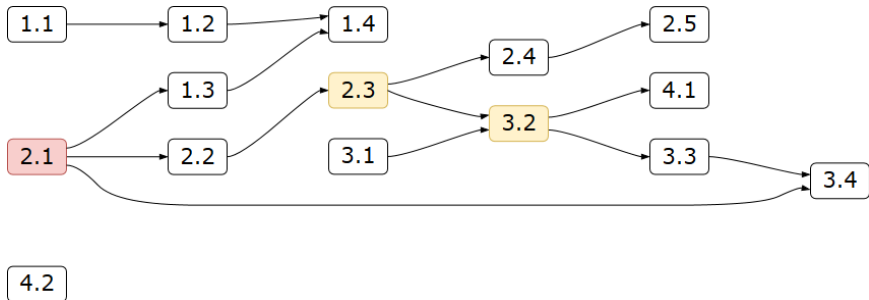
| Aufgabe | Setzt voraus |
|---------|--------------|
| 3.1 | - |
| 3.2 | 2.3 |
| 3.3 | 3.2 |
| 3.4 | 3.3, 2.1 |
| 4.1 | 3.2 |
| 4.2 | - |

Abhängigkeiten im Diagramm

- Von 2.1 hängen 3 weitere Prozesse ab.

→ Höhere Priorisierung von 2.1

- 1.1, 2.1 und 4.2 können parallel bearbeitet werden



Lehrziele

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



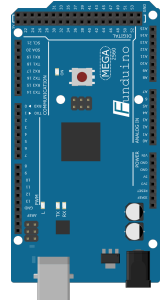
Einführung

Grundlagen

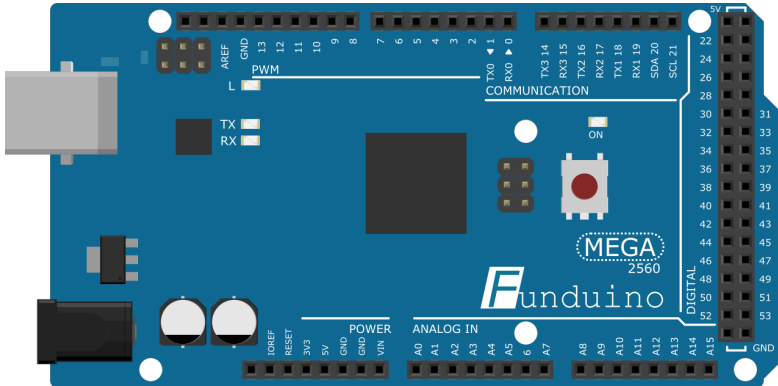
Einstiegs-
projekt

Arduino? Funduino?

- Open-Source Elektronik-Plattform
 - Funduinos 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



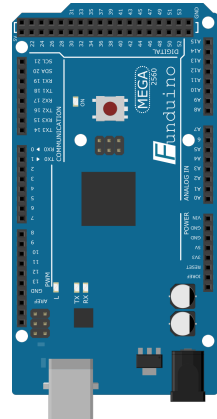
Funduino Mega 2560



Funduino Mega 2560

Technische Daten

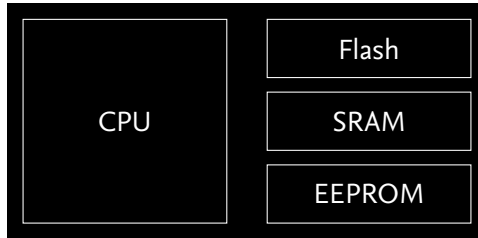
| | |
|--------------------------|--------------|
| Mikrocontroller | ATmega2560 |
| Betriebsspannung | 5V |
| Empf. Eingangsspannung | 7-12V |
| Grenzen Eingangsspannung | 6-20V |
| Digitale I/O-Pins | 54 |
| | davon 15 PWM |
| Analoge Input-Pins | 16 |
| Max. Strom pro I/O-Pin | 20 mA |
| Max. Strom pro 3,3V-Pin | 50 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 Ausschalten entfernt Inhalt |
| EEPROM | 4 kB | Persistenter Speicher Inhalt bleibt erhalten |



Quellen

- VSCode:
<https://code.visualstudio.com>
- VSCode-Erweiterung PlatformIO:
<https://platformio.org/get-started>
- Arduino Entwicklungsumgebung (IDE):
<http://www.arduino.org/downloads>
- Arduino Online-IDE¹
<https://create.arduino.cc/>

¹Nicht alle Funktionen werden von der Online-IDE unterstützt

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

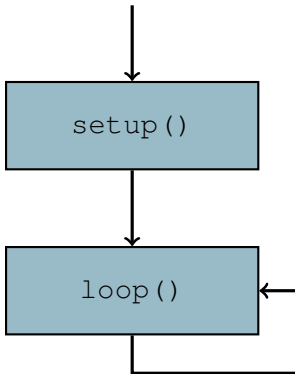
☐ ein Arduino-Projekt erstellen können

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



Programmablauf

Einschalten oder
Reset-Taste drücken



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.

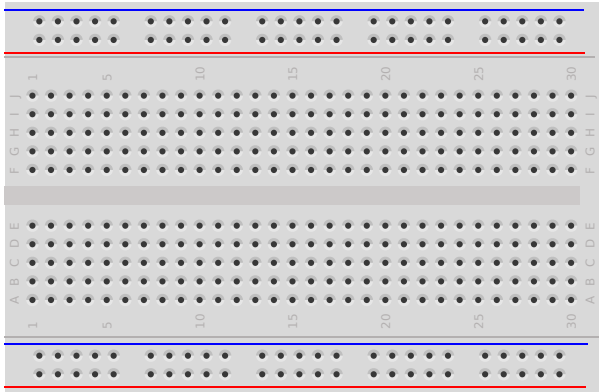


Abbildung 1: Leiterbahnen einer Steckplatine

Schaltung erstellen

Steckplatine

Mit einer Steckplatine können Bauteile temporär verbunden werden.

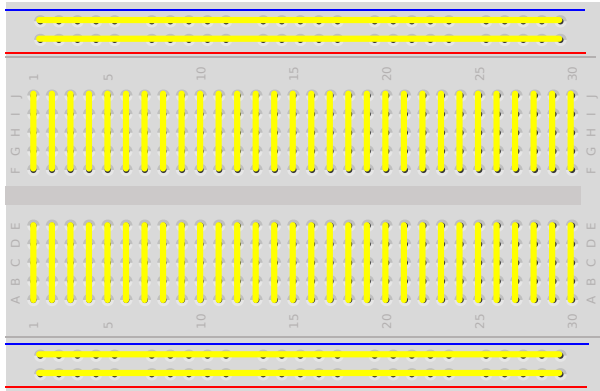


Abbildung 1: Leiterbahnen einer Steckplatine

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?



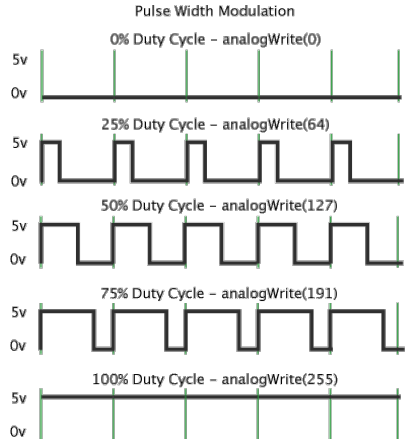
Analoger Ausgang über PWM

- 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

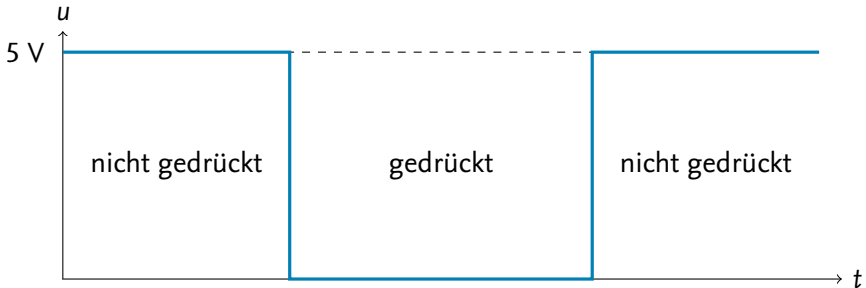
Externer Pull-Up

Das Problem:

Der Eingang ist anfällig für Störsignale.

Die Lösung:

Ein Eingang muss mit einem definierten Potential versehen werden.



Eingangssignal

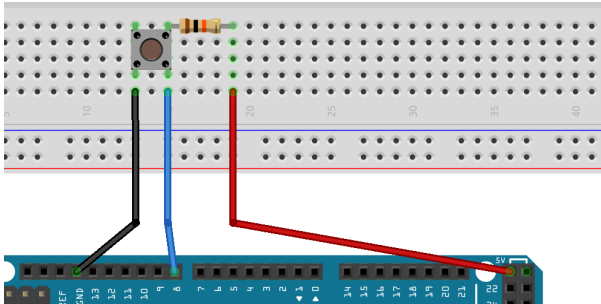
Externer Pull-Up

Das Problem:

Der Eingang ist anfällig für Störsignale.

Die Lösung:

Ein Eingang muss mit einem definierten Potential versehen werden.



Pull-Up

Schema

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

Zustand Schalter offen:

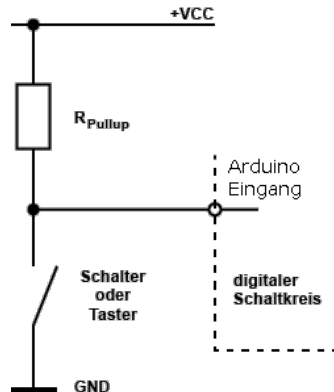
Spannung liegt am Eingang an

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

Zustand Schalter geschlossen:

Masse liegt am Eingang an

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



Pull-Down

Schema

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

Zustand Schalter offen:

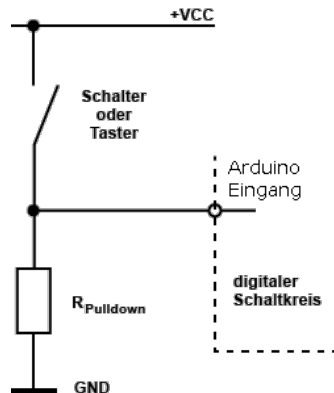
Masse liegt am Eingang an

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

Zustand Schalter geschlossen:

Spannung liegt am Eingang an

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



Interner Pull-Up

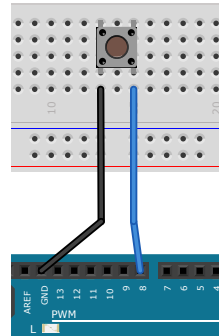
Beispiel

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.



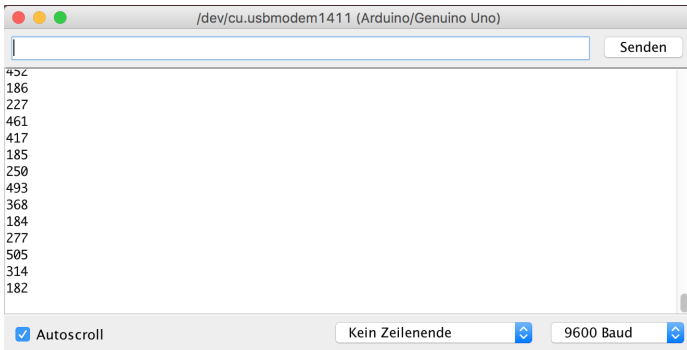
Listing 6: Serielle Ausgabe

```
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

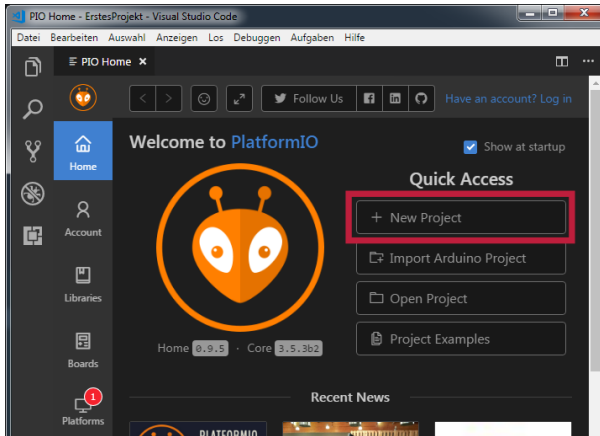
Als Teilnehmer soll ich am Ende dieser Übung...

- ☒ 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

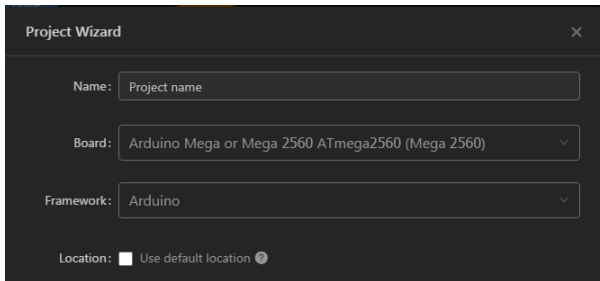
IDE-Einstellungen



IDE-Einstellungen

Wichtig

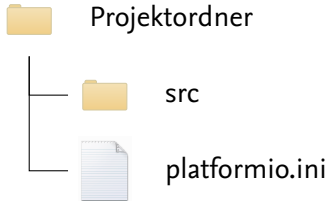
Bei den Funduinios muss der Arduino Mega 2560 ausgewählt werden.



The screenshot shows the 'Project Wizard' dialog box in the Arduino IDE. It has a dark theme and a close button (X) in the top right corner. The dialog contains three main sections: 'Name' with a text input field containing 'Project name'; 'Board' with a dropdown menu showing 'Arduino Mega or Mega 2560 ATmega2560 (Mega 2560)'; and 'Framework' with a dropdown menu showing 'Arduino'. At the bottom, there is a 'Location' section with a checkbox (which is unchecked) and the text 'Use default location' followed by a help icon.

Organisation von Arduino-Projekten

Die `platformio.ini` Datei enthält die Projekteinstellungen für das Arduino-Projekt



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

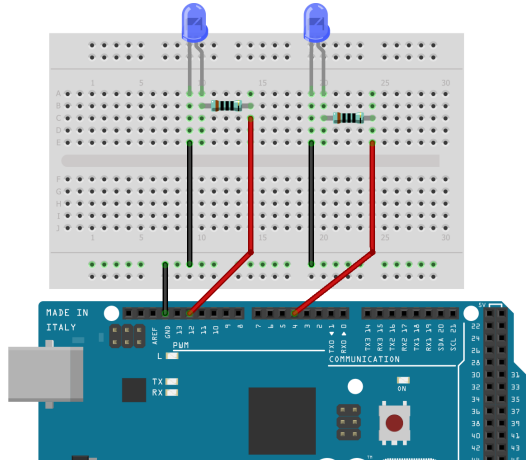
Bauteile:

2 x Widerstand 100 Ω

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 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!