



Technische
Universität
Braunschweig

Institut für
Flugführung



Arduino/Funduino

Einführung

Prof. Dr.-Ing. Peter Hecker, Dipl.-Ing. Paul Frost, 02. Mai 2017

Agenda

- 04. April Kick-Off
- 11. April Projektmanagement
- 18. April Prozessmodelle
- 25. April Versionsverwaltung
- 02. Mai Einführung Arduino/Funduino**
- 09. Mai Entwicklungsumgebungen und Debugging
- 16. Mai Dokumentation und Testing
- 23. Mai Dateieingabe und -ausgabe
- 30. Mai GUI-Erstellung mit Qt
- 06. Juni Exkursionswoche
- 13. Juni Bibliotheken
- 20. Juni Netzwerke
- 27. Juni Projektarbeit
- 04. Juli Projektarbeit
- 11. Juli Vorbereitung der Abgabe

Teil I

Wiederholung

Git als Konsolenapplikation

Sichtbarkeit

Git ist eine Konsolenapplikation und muss nicht zwingend im Programmeordner installiert sein!

Arbeitsverzeichnis

Der Benutzer muss sich in der Konsole im richtigen Arbeitsverzeichnis befinden. (siehe den Befehl `cd`)

Funktionsweise

- `git` als Befehl eingeben.
- Anschließend die Optionen

Die wichtigsten Befehle

<code>git init</code>	Erstellt ein Repository
<code>git clone</code>	Kopiert bzw. kloniert ein bestehendes Repository
<code>git status</code>	Zeigt den Status des aktuellen Zweiges
<code>git add</code>	Fügt Dateien für ein Commit hinzu
<code>git commit</code>	Speichert den aktuellen Stand
<code>git push</code>	Lädt die Commits hoch zu einem Server
<code>git pull</code>	Lädt die Commits von einem Server runter
<code>git checkout</code>	Erstellung von und Wechsel zwischen Zweigen
<code>git merge</code>	Zusammenführen von Zweigen

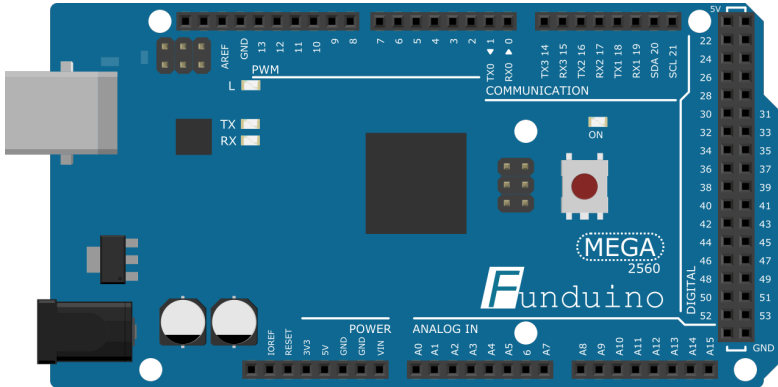
Teil II

Arduino/Funduino

Arduino? Funduino?

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

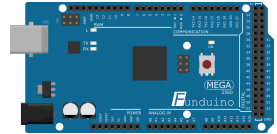
Funduino Mega 2560



Funduino Mega 2560

Technische Daten

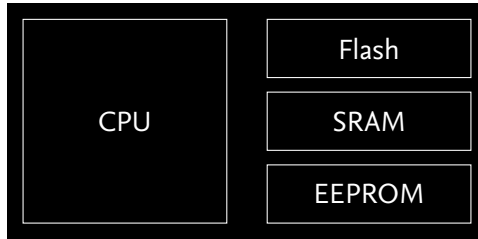
Mikrocontroller	ATmega2560
Betriebsspannung	5V
Empf. Eingangsspannung	7-12V
Grenzen Eingangsspannung	6-20V
Digitale I/O Pins	54
	davon 15 PWM
Analoge Input Pins	16
Max. Strom pro I/O Pin	20 mA
Max. Strom pro 3,3V Pin	50 mA
Taktrate	16 MHz



Funduino Mega 2560

Speicher Mikrocontroller

Speicherart	Größe	Verwendung
Flash Speicher	256 kB	Speicherung des Sketches 8 kB vom Bootloader verwendet
SRAM	8 kB	Speicherung von Variablen Ausschalten entfernt Inhalt
EEPROM	4 kB	Persistenter Speicher Inhalt bleibt erhalten



Quellen

- Arduino Entwicklungsumgebung (IDE):
<http://www.arduino.org/downloads>
- Arduino Online-IDE¹
<https://create.arduino.cc/>

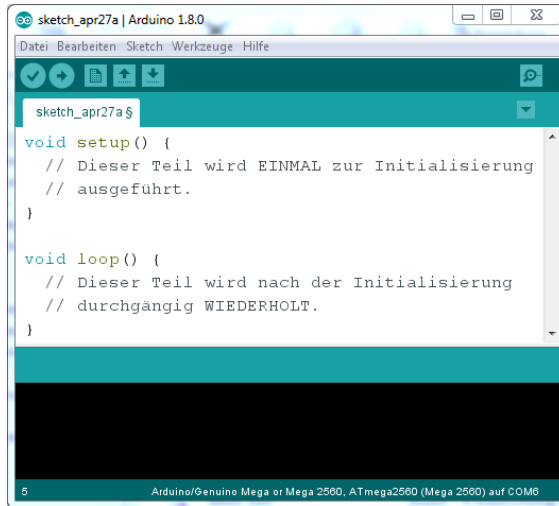
¹Nicht alle Funktionen werden von der Online-IDE unterstützt

Programmieren mit einem Arduino

1. Entwicklung
 - Programmierung
 - Schaltung erstellen
2. Sketch überprüfen
3. Sketch auf Arduino/Funduino hochladen
4. Sketch testen
5. Sketch und Schaltung dokumentieren

Programmierung

Arduino IDE



```
sketch_apr27a | Arduino 1.8.0
Datei Bearbeiten Sketch Werkzeuge Hilfe

sketch_apr27a $

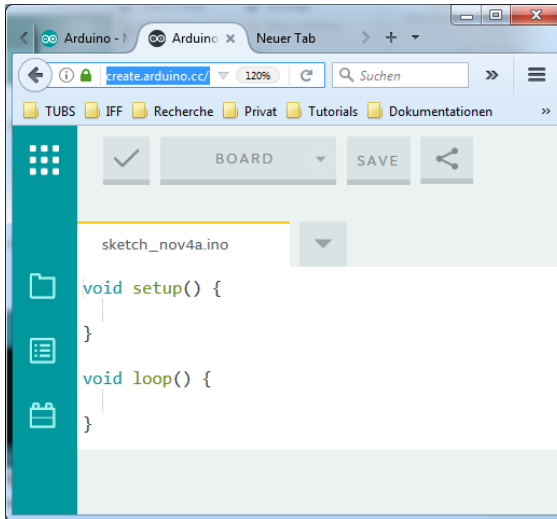
void setup() {
  // Dieser Teil wird EINMAL zur Initialisierung
  // ausgeführt.
}

void loop() {
  // Dieser Teil wird nach der Initialisierung
  // durchgängig WIEDERHOLT.
}

5
Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) auf COM8
```

Programmierung

Arduino Online-IDE



IDE-Einstellungen

- **Boardauswahl**

Werkzeuge → Board:

Arduino Mega 2560

- **Prozessorauswahl**

Werkzeuge → Prozessor:

ATmega 2560

- **Portauswahl**

Werkzeuge → Port:

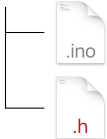
COM (*Nr. kann variieren*)

Organisation von Arduino-Projekten

- Hauptdatei besitzt den gleichen Namen wie der Projektordner
- Über Tab-Funktion können weitere Dateien angelegt werden
- *.ino Dateien werden der Hauptdatei angehängen
- *.h, *.c und *.cpp Dateien werden separat kompiliert
können z. B. über `#include "header.h"` hinzugefügt werden



ProjektName



ProjektName.ino

header.h

Programmierung

Sketch

Listing 1: code/Arduino05_01_zweiLEDs/Arduino05_01_zweiLEDs.ino

```
1 void setup() //Wir starten mit dem Setup
2 {
3   pinMode(12, OUTPUT); // Pin 12 ist ein Ausgang.
4   pinMode(4, OUTPUT); // Pin 4 ist ein Ausgang.
5 }
6
7 void loop() // Das Hauptprogramm beginnt.
8 {
9   digitalWrite(12, HIGH); // Schalte die LED an Pin12 an.
10  delay(1000); // Warte 1000 Millisekunden.
11  digitalWrite(12, LOW); // Schalte die LED an Pin12 aus.
12  digitalWrite(4, HIGH); // Schalte die LED an Pin4 ein.
13  delay(1000); // Warte 1000 Millisekunden.
14  digitalWrite(4, LOW); // Schalte die LED an Pin4 aus.
15 }
```

Schaltung erstellen

Steckplatine

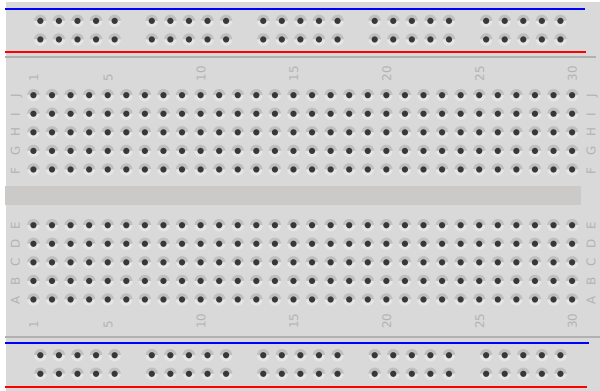


Abbildung 1: Leiterbahnen einer Steckplatine

Schaltung erstellen

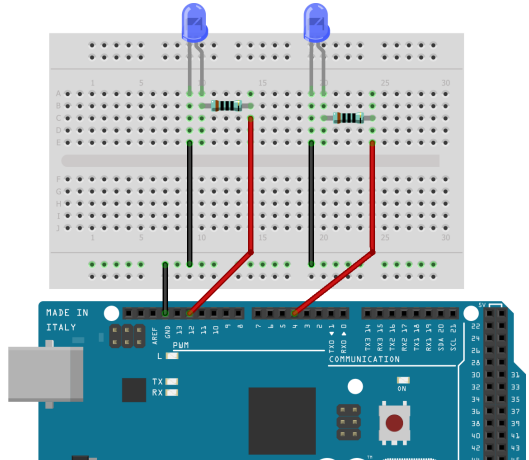
Bauteile:

2 x Widerstand 100 Ω

2 x LED blau



LEDs NIE ohne Widerstand verwenden!



Schaltung erstellen

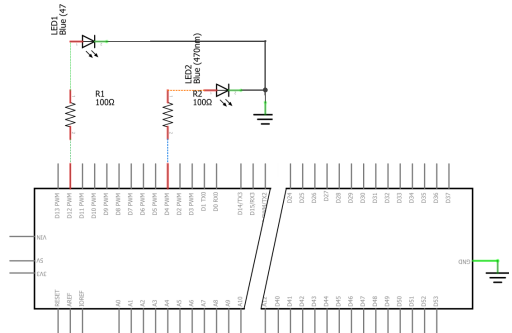
Bauteile:

2 x Widerstand 100 Ω


2 x LED blau



LEDs NIE ohne Widerstand verwenden!



Sketch hochladen

- Über den Bootloader können Sketche direkt auf den Mikrocontroller geladen werden
- Erforderliche Schritte:
 1. Funduino per USB anschließen
 2. Funduino einrichten (siehe Folie 15)
 3. Sketch compilieren
 4. Sketch über den Hochladen-Button  auf den Funduino laden

Analoger Ausgang über PWM

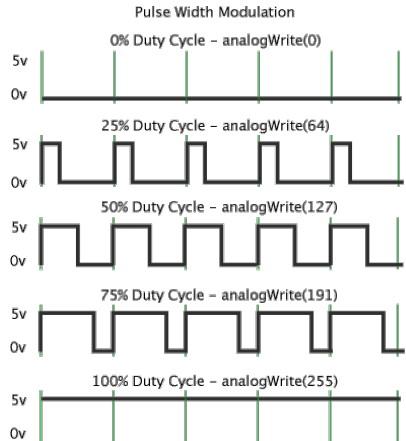
Wie dimme ich eine LED, wenn ich nur über 5-Volt-Ausgänge verfüge?

- Problem:

Ausgabe nur als 5 Volt
Highpegel möglich

→ Generierung eines analogen
Signals über eine
Pulsweitenmodulation

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



Analoger Ausgang über PWM

Listing 2: code/Arduino05_02_pwm/Arduino05_02_pwm.ino

```
1 int i = 0;
2 void setup() //Wir starten mit dem Setup
3 {
4   pinMode(12, OUTPUT); // Pin 12 ist ein Ausgang.
5 }
6
7 void loop() // Das Hauptprogramm beginnt.
8 {
9   i++;
10  analogWrite(12, i); // 0 bis 5 V ueber PWM
11  if(i == 255)
12  {
13    i=0;
14  }
15  delay(10); //Hauptprogramm wartet 10ms
16 }
```


Eingangssignal

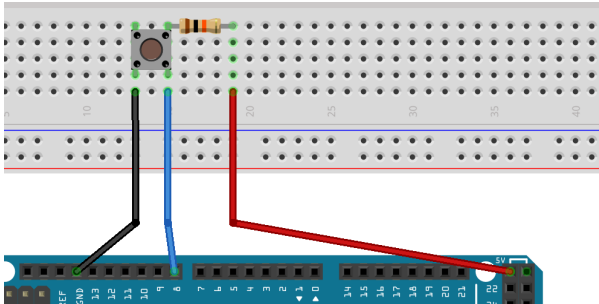
Externer Pull-Up

Das Problem:

Der Eingang ist anfällig für Störsignale.

Die Lösung:

Ein Eingang muss mit einem definierten Potential versehen werden.



Pull-Up

Schema

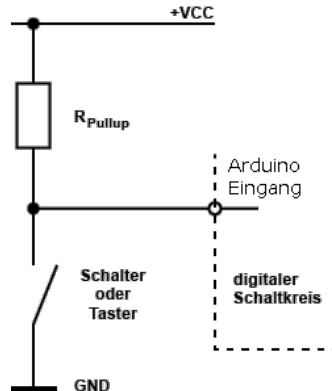
```
pinMode(pin, INPUT);
```

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:

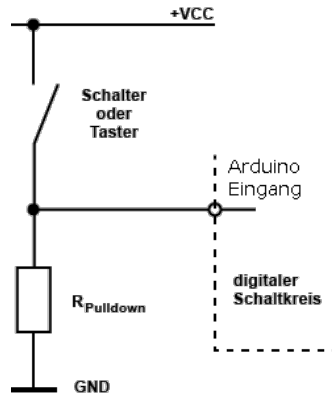
Masse liegt am Eingang an

`digitalRead(pin) == LOW`

Zustand Schalter geschlossen:

Spannung liegt am Eingang an

`digitalRead(pin) == HIGH`



Interner Pull-Up

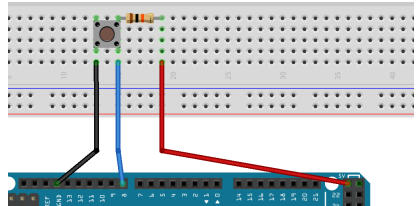
Beispiel

Listing 3: code/Arduino05_03_pullup/Arduino05_03_pullup.ino

```

1 void setup() {
2   pinMode(2, INPUT_PULLUP);
3 }
4
5 void loop() {
6   if(digitalRead(2) == LOW)
7   {
8     // Button wurde gedruickt
9     // das wird ausgefuehrt...
10  }
11 }

```



Serielle Kommunikation

Quellcode

Listing 4: code/Arduino05_04_serial/Arduino05_04_serial.ino

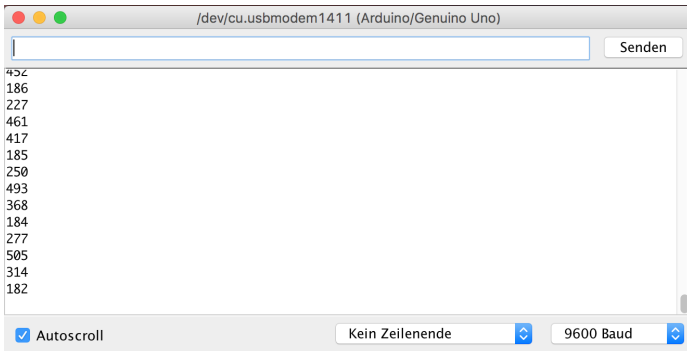
```
1 void setup() {  
2   // Initialisierung der seriellen Schnittstelle  
3   Serial.begin(9600);  
4 }  
5  
6 void loop() {  
7   int sensorValue = analogRead(A0);  
8  
9   // Schreiben des ausgelesenen Werts von Analog-Pin-0  
10  Serial.println(sensorValue);  
11  delay(1);          // 1 ms warten wegen Stabilitaet  
12 }
```

Serielle Kommunikation

Serieller Monitor

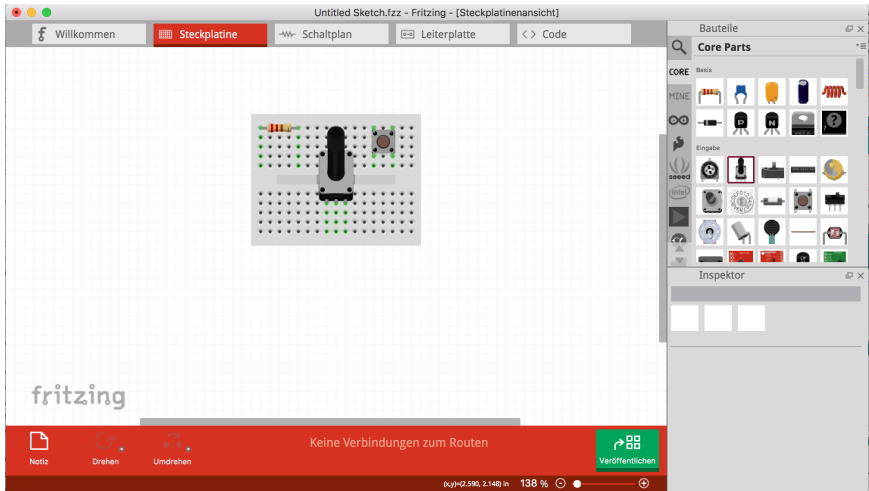
Listing 5: code/Arduino05_04_serial/Arduino05_04_serial.ino

```
1 // Schreiben des ausgelesenen Werts von Analog-Pin-0  
2 Serial.println(sensorValue);
```



Dokumentation

Fritzing



Weiteres Vorgehen

Wenn ich einen Sensor oder ein anderes Bauteil nutzen möchte:

- Recherche über das Bauteil
<http://funduino.de/anleitungen>
- Lesen der Dokumentation
- Beispiele nutzen

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. Hiwis fragen
5. Dozenten fragen

Hiwi-Github-Benutzernamen

In GitHub können Issues angelegt und Betreuern zugewiesen werden. In der Beschreibung des Issues kann über @GitBenutzer erwähnt werden. Diese erhalten direkt eine E-Mail.

Name	GitHub	Themen
Bastian Riedel	bariedel	Installationen und C++
Peter Pauly	pauly peter	Git und GitHub
Marc Ilic	Marc-K	Arduino-Bauteile
Justus Drögemüller	JustusDroege	Fragen Arduino

Wichtig:

Zunächst die Schritte 1-3 bei der Problemlösung befolgen.

Teil III

Projektarbeit

Aufgabe 1

- Festigen der git-Kenntnisse
- Hochladen von Bildern in das Repository

Aufgabe 2

- Erstes Arduino-Projekt bearbeiten. Zum Beispiel:
 - Testprojekt
 - Prototyp
 - Teilmodule
- Dokumentation nicht vergessen

Teil C API-Spiralmodell

Aufgabe 3

- Fangen Sie mit der Bearbeitung der Ziele aus Teil A, des von Ihnen definierten Spiralmodells, an.

Dokumentation Projektverlauf/Zeitplan

Aufgabe 4

- Führen Sie die Dokumentation des Projektverlaufs auf der Projektverlauf Seite im Wiki fort.

Aufgabe 5

- Erstellen Sie die Wiki-Seite Zeitplan
- Laden Sie ihren Zeitplan in das Wiki hoch

Fragen?

Gibt es noch Fragen?