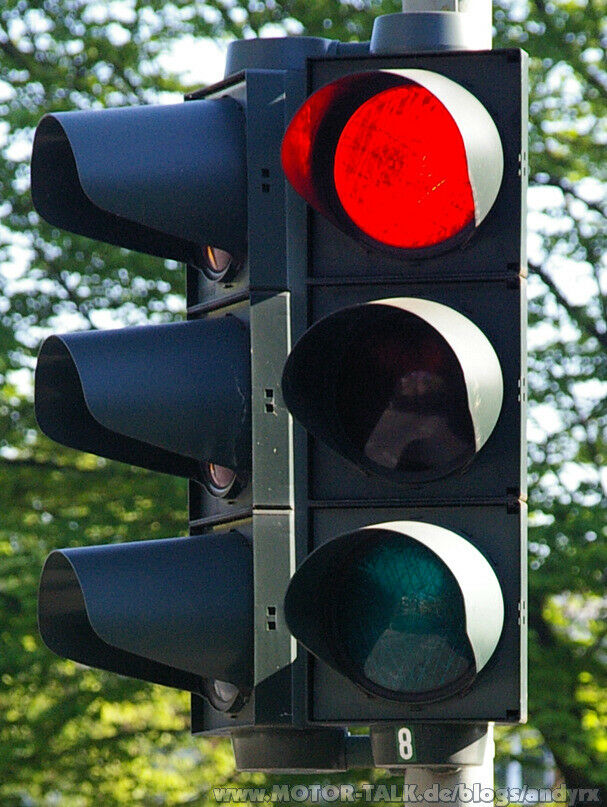
Ampel-

Steuerung

mit Arduino



Manuel Haag

April 2020

# Einleitung

In diesem Projekt bauen wir eine Ampelanlage bestehend aus:

* einer Ampel für Autos
* einer Ampel für Fussgänger mit Knopf
* eine Steuerung mit Arduino

Die Anlage läuft selbständig mit einer 9V Batterie.

## Funktionsweise der Ampel

Beim Start (oder nach RESET) blinken beide Ampeln orange.

Sobald der Knopf bei der Fussgängerampel gedrückt wird, schalten zuerst beide Ampeln auf Rot, danach folgt die Grünphase (10 Sekunden) für Fussgänger.

Danach stellt die Fussgängerampel auf Rot und die Autos bekommen grün (zeitlich unbeschränkt).

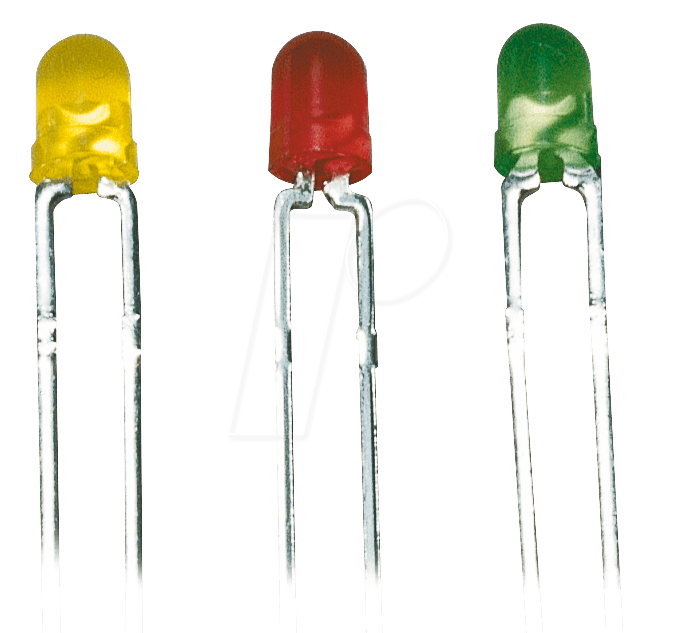
Bei erneutem Drücken des Knopfes erfolgt wieder eine Grünphase für die Fussgänger.

Die Grünphase für Autos dauert jedoch mindestens 12 Sekunden (falls der Knopf zu früh gedrückt wird).

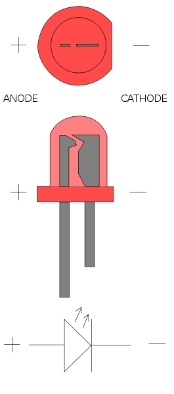
# Bauteile

## LED

Leuchtdioden (Englisch: Light Emitting Diode – *LED*) sind kleine Lämpchen. Es gibt sie in verschiedenen Grössen und Farben. Wir verwenden Gelb, Rot und Grün.



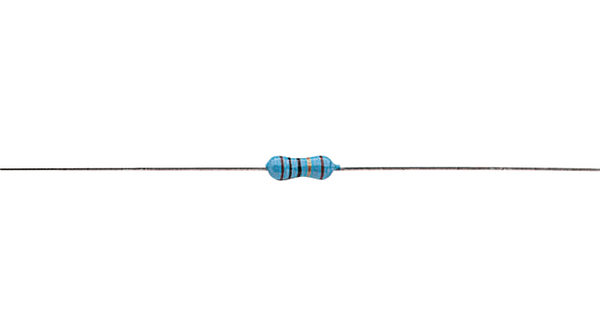
Die LEDs funktionieren nur, wenn Plus und Minus richtig angeschlossen werden. Das kurze Bein ist der Minuspol:



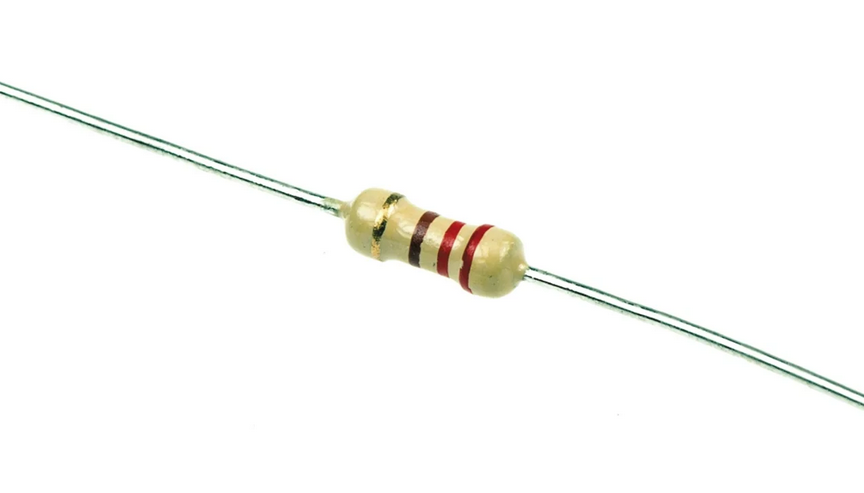
Damit die LEDs nicht kaputt gehen, braucht es vor jeder LED einen Widerstand.

## Widerstand

Widerstände «bremsen» den Stromfluss. Es gibt schwache und starke Widerstände. Der Wert des Widerstands ist mit Farbringen gekennzeichnet und heisst «Ohm».  
Beim Widerstand spielt es keine Rolle, wie man ihn anschliesst (Plus/Minus).

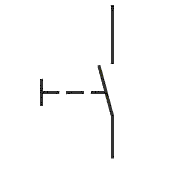


Widerstände können zum Beispiel dazu dienen, die LEDs vor zu hohen Strömen zu schützen.  
Für unsere Ampel braucht es vor jeder LED einen kleinen Widerstand von 220 Ohm. Dieser hat die Farbcodierung ROT-ROT-BRAUN:



## Taster Knopf

Ein Taster ist ein kleiner Schalter, welcher den Stromkreis schliesst, solange er gedrückt wird.  
Damit kann man der Ampelsteuerung mitteilen, dass jemand über den Fussgängerstreifen gehen will.

Wir verwenden diesen Taster mit eingebautem LED-Lämpchen, welches aufleuchtet, wenn der Knopf gedrückt wurde.



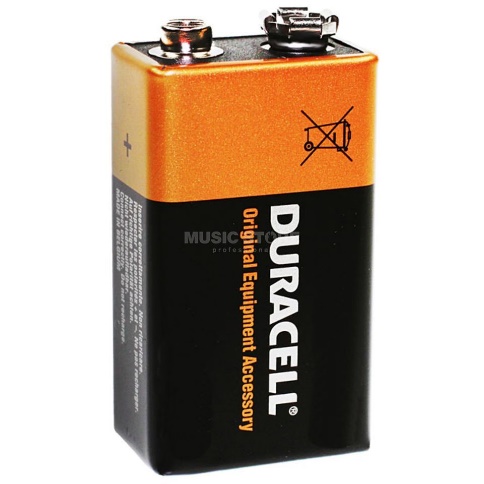
## Stromversorgung

Die Ampel wird mit einer 9V Batterie betrieben und über einen Stecker mit Kabel am Arduino angeschlossen.

ACHTUNG: die Batterie darf NIE verkehrt angeschlossen werden, sonst geht die Steuerung kaputt!

Beim Anschliessen des Steckers an die Batterie immer darauf achten, dass die Polung stimmt!

Rot = Pluspol  
Schwarz = Minuspol

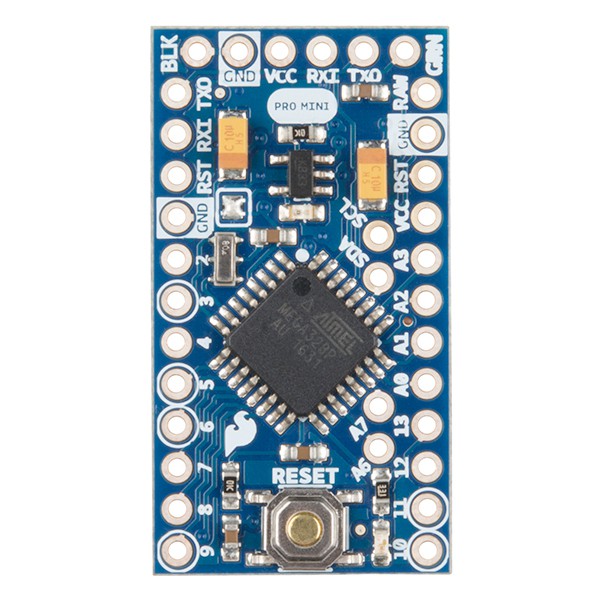


## Arduino

Arduino ist ein kleiner Computer, der die Ampel steuert. Er besteht aus einem Mikrocontroller, den man programmieren kann und er hat mehrere Anschlüsse für Ein- und Ausgänge. An den Ausgängen werden die LEDs angeschlossen, an einem Eingang der Taster.

Der Arduino benötigt eine Stromversorgung (9V Batterie) und kann mit dem RESET-Knopf jederzeit neu gestartet werden.

Zusätzlich hat es auf dem Arduino eine kleine grüne LED, welche nützlich ist, wenn man schnell etwas Testen will.



9V Batterie

**Arduino**

RESET-Knopf

Mikrocontroller  
(programmierbar)

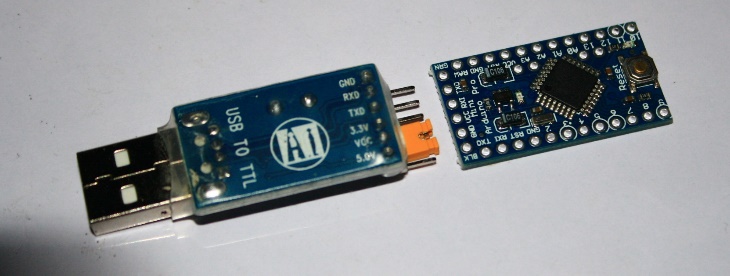
Ein- oder Ausgänge

kleine LED

# Programmierung des Arduino

## Programmierung über FTDI-Kabel

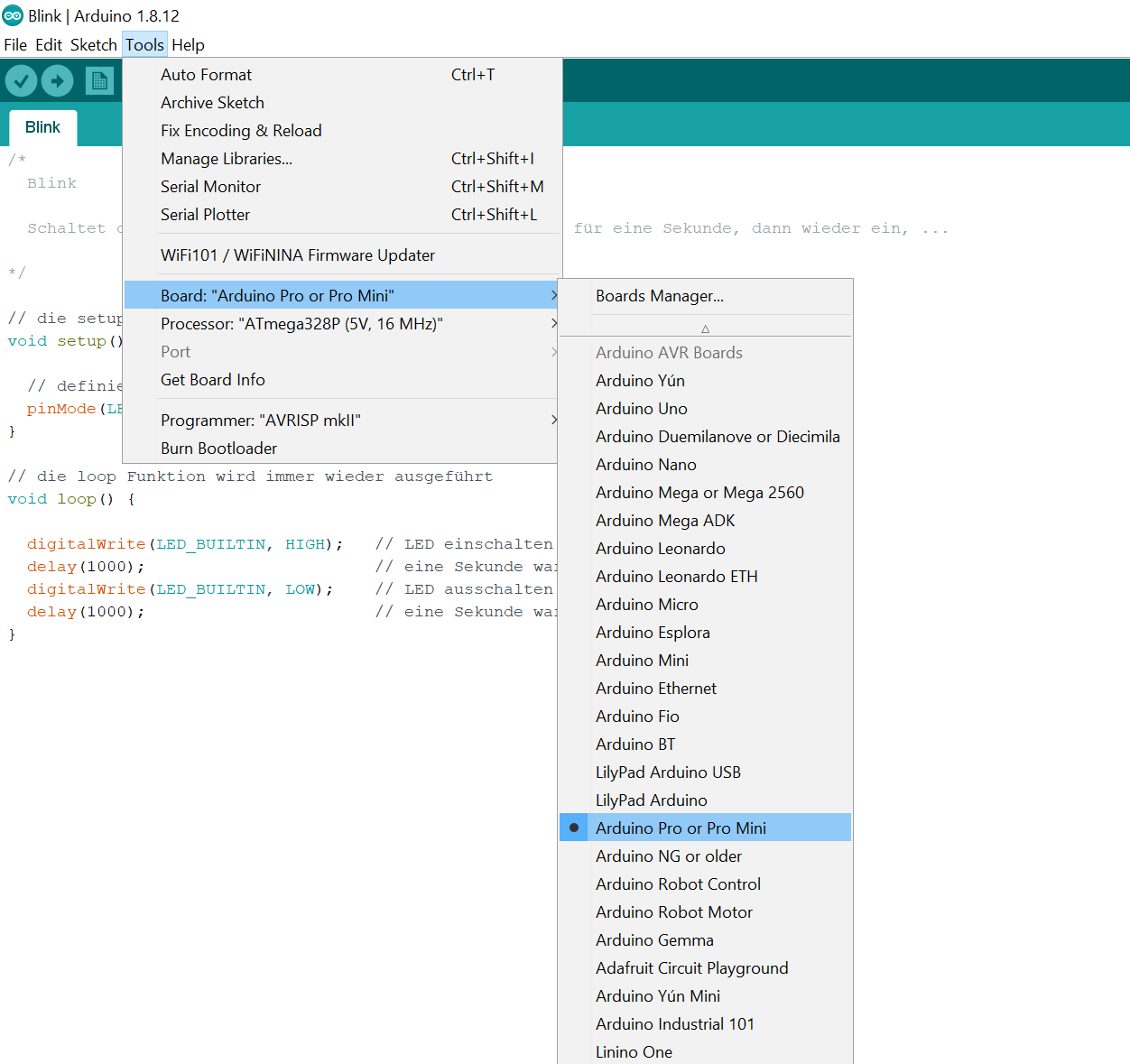
Der Arduino wird über einen normalen PC oder Laptop mit einem speziellen FTDI Kabel über USB programmiert. Wenn er einmal programmiert ist, läuft der Arduino selbständig (ohne PC). Jedesmal, wenn man am Programm etwas ändern will, muss man den Arduino neu programmieren.



Für die Programmierung benötigt man eine Entwicklungsumgebung auf dem PC namens Arduino IDE. Diese kann hier downgeloadet werden:

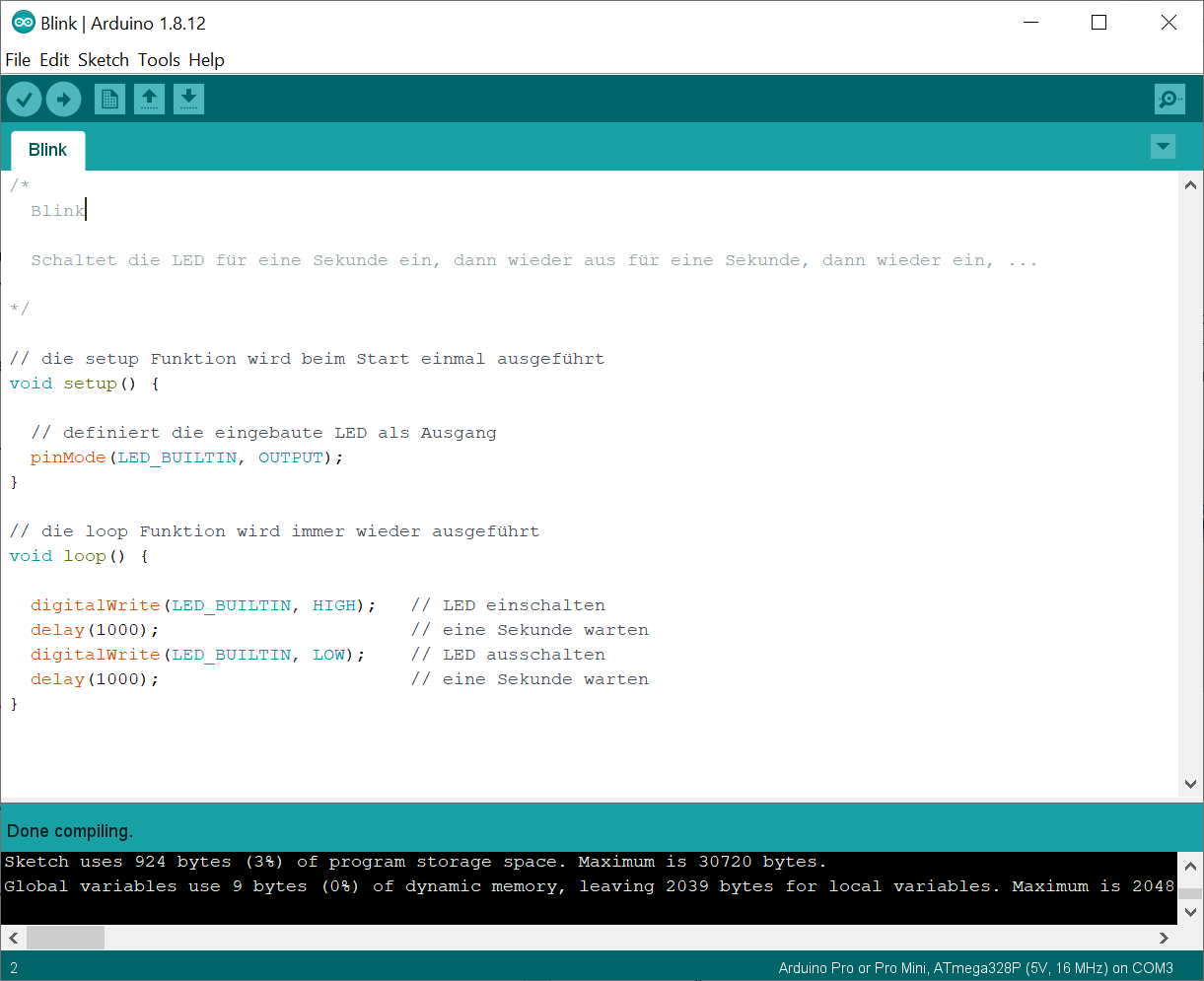
<https://www.arduino.cc/en/Main/Software>

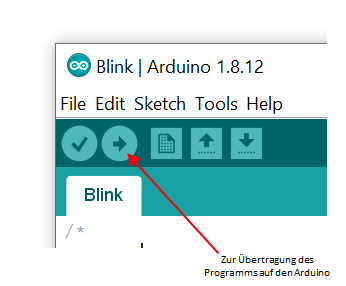
Unter Tools – Board muss «Arduino Pro or Pro Mini” ausgewählt werden



## Erstes Testprogramm «Blink»

Wir schreiben ein erstes kleines Programm, welches die kleine grüne LED auf dem Arduino blinken lässt.





## Programm für die Ampelsteuerung

Der Quellcode ist hier zu finden: <https://github.com/helijunky/Ampel>

/\*

Ampelsteuerung mit Arduino

Manuel Haag

März 2020

Funktionsweise

--------------

Beim Start (oder nach RESET) blinken beide Ampeln orange.

Sobald der Knopf bei der Fussgängerampel gedrückt wird, schalten zuerst

beide Ampeln auf Rot, danach folgt die Grünphase für Fussgänger

(10 Sekunden).

Danach stellt die Fussgängerampel auf Rot und die Autos bekommen grün

(zeitlich unbeschränkt).

Bei erneutem Drücken des Knopfes erfolgt wieder eine Grünphase für die

Fussgänger. Die Grünphase für Autos dauert jedoch mindestens 12 Sekunden

(falls der Knopf zu früh gedrückt wird).

\*/

// Definition der Namen für Ein- und Ausgänge am Arduino

#define LED\_AUTO\_ROT 9              // Ausgang 9 für Auto Rot

#define LED\_AUTO\_GELB 8             // Ausgang 8 für Auto Gelb

#define LED\_AUTO\_GRUEN 7            // Ausgang 7 für Auto Grün

#define LED\_FUSSG\_ROT 6             // Ausgang 6 für Fussgänger Rot

#define LED\_FUSSG\_GELB 5            // Ausgang 5 für Fussgänger Gelb

#define LED\_FUSSG\_GRUEN 4           // Ausgang 4 für Fussgänger Grün

#define LED\_KNOPF 3                 // Ausgang 3 für LED in Knopf

#define KNOPF 2                     // Eingang 2 für Knopf

// Variablen

bool blinken = true;                // true:  Blinken eingeschaltet

                                   // false: Blinken ausgeschaltet

volatile bool gedrueckt;            // true:  Knopf wurde gedrückt

                                   // false: Knopf nicht gedrückt

// die setup Funktion wird beim Start einmal ausgeführt

void setup() {

 // alle LEDs werden als Ausgang definiert

 pinMode(LED\_BUILTIN, OUTPUT);

 pinMode(LED\_AUTO\_ROT, OUTPUT);

 pinMode(LED\_AUTO\_GELB, OUTPUT);

 pinMode(LED\_AUTO\_GRUEN, OUTPUT);

 pinMode(LED\_FUSSG\_ROT, OUTPUT);

 pinMode(LED\_FUSSG\_GELB, OUTPUT);

 pinMode(LED\_FUSSG\_GRUEN, OUTPUT);

 pinMode(LED\_KNOPF, OUTPUT);

 // der Knopf wird als Eingang definiert

 pinMode(KNOPF, INPUT\_PULLUP);

 // sobald der Knopf gedrückt wird, wird die Funktion "druecken" aufgerufen

 // (Interrupt-Funktion)

 attachInterrupt(digitalPinToInterrupt(KNOPF), druecken, FALLING);

 // zum Testen werden alle LEDs kurz eingeschaltet

 digitalWrite(LED\_AUTO\_ROT, HIGH);

 digitalWrite(LED\_AUTO\_GELB, HIGH);

 digitalWrite(LED\_AUTO\_GRUEN, HIGH);

 digitalWrite(LED\_FUSSG\_ROT, HIGH);

 digitalWrite(LED\_FUSSG\_GELB, HIGH);

 digitalWrite(LED\_FUSSG\_GRUEN, HIGH);

 digitalWrite(LED\_KNOPF, HIGH);

 digitalWrite(LED\_BUILTIN, HIGH);

 delay(2000);

 // und wieder ausgeschaltet

 digitalWrite(LED\_AUTO\_ROT, LOW);

 digitalWrite(LED\_AUTO\_GELB, LOW);

 digitalWrite(LED\_AUTO\_GRUEN, LOW);

 digitalWrite(LED\_FUSSG\_ROT, LOW);

 digitalWrite(LED\_FUSSG\_GELB, LOW);

 digitalWrite(LED\_FUSSG\_GRUEN, LOW);

 digitalWrite(LED\_KNOPF, LOW);

 digitalWrite(LED\_BUILTIN, LOW);

 delay(1000);

 gedrueckt = false;  // erster Knopfdruck löschen (Interrupt wird bei jedem Start

                     // einmal ausgeführt)

}

// Interrupt-Funktion wird aufgerufen, sobald der Knopf gedrückt wird

void druecken() {

 if (!gedrueckt) {                      // falls noch nicht gedrückt wurde...

   gedrueckt = true;                   // sich merken, dass der Knopf gedrückt wurde

   digitalWrite(LED\_KNOPF, HIGH);      // LED im Knopf einschalten

   digitalWrite(LED\_BUILTIN, HIGH);    // kleine LED auf dem Arduino einschalten

 }

}

// die loop Funktion wird immer wieder ausgeführt

void loop() {

 // Blinken der orangen LEDs

 if (blinken) {                        // wenn Blinken aktiv ist

   digitalWrite(LED\_AUTO\_GELB, HIGH);  // gelbe LED für Auto einschalten

   digitalWrite(LED\_FUSSG\_GELB, HIGH); // gelbe LED für Fussgänger einschalten

   digitalWrite(LED\_BUILTIN, HIGH);    // kleine LED auf dem Arduino einschalten

   delay(750);                         // 750 Millisekunden warten

   digitalWrite(LED\_AUTO\_GELB, LOW);   // gelbe LED für Auto ausschalten

   digitalWrite(LED\_FUSSG\_GELB, LOW);  // gelbe LED für Fussgänger ausschalten

   digitalWrite(LED\_BUILTIN, LOW);     // kleine LED auf dem Arduino ausschalten

   delay(750);                         // 750 Millisekunden warten

 }

 if (gedrueckt) {                      // wenn gedrückt wurde...

   if (blinken) {                      // wenn noch Blinken aktiv ist...

     blinken = false;                  // Blinken ausschalten

     // Fussgänger Ampel auf Orange stellen

     digitalWrite(LED\_FUSSG\_ROT, LOW);

     digitalWrite(LED\_FUSSG\_GELB, HIGH);

     digitalWrite(LED\_FUSSG\_GRUEN, LOW);

   }

   // Auto Ampel auf Orange stellen

   digitalWrite(LED\_AUTO\_ROT, LOW);

   digitalWrite(LED\_AUTO\_GELB, HIGH);

   digitalWrite(LED\_AUTO\_GRUEN, LOW);

   delay(4000);                        // 4 Sekunden warten

   // Auto Ampel auf Rot stellen

   digitalWrite(LED\_AUTO\_ROT, HIGH);

   digitalWrite(LED\_AUTO\_GELB, LOW);

   digitalWrite(LED\_AUTO\_GRUEN, LOW);

   // Fussgänger Ampel auf Rot stellen

   digitalWrite(LED\_FUSSG\_ROT, HIGH);

   digitalWrite(LED\_FUSSG\_GELB, LOW);

   digitalWrite(LED\_FUSSG\_GRUEN, LOW);

   delay(4000);                        // 4 Sekunden warten

   // Fussgänger Ampel auf Grün stellen

   digitalWrite(LED\_FUSSG\_ROT, LOW);

   digitalWrite(LED\_FUSSG\_GELB, LOW);

   digitalWrite(LED\_FUSSG\_GRUEN, HIGH);

   digitalWrite(LED\_KNOPF, LOW);       // LED im Knopf ausschalten

   digitalWrite(LED\_BUILTIN, LOW);     // kleine LED ausschalten

   delay(10000);                       // 10 Sekunden warten

   // Fussgänger Ampel auf Orange stellen

   digitalWrite(LED\_FUSSG\_ROT, LOW);

   digitalWrite(LED\_FUSSG\_GELB, HIGH);

   digitalWrite(LED\_FUSSG\_GRUEN, LOW);

   delay(4000);                        // 4 Sekunden warten

   // Fussgänger Ampel auf Rot stellen

   digitalWrite(LED\_FUSSG\_ROT, HIGH);

   digitalWrite(LED\_FUSSG\_GELB, LOW);

   digitalWrite(LED\_FUSSG\_GRUEN, LOW);

   gedrueckt = false;                  // ab jetzt kann man wieder drücken

   delay(2000);                        // 2 Sekunden warten

   // Auto Ampel auf Rot+Orange stellen

   digitalWrite(LED\_AUTO\_ROT, HIGH);

   digitalWrite(LED\_AUTO\_GELB, HIGH);

   digitalWrite(LED\_AUTO\_GRUEN, LOW);

   delay(2000);                        // 2 Sekunden warten

   // Auto Ampel auf Grün stellen

   digitalWrite(LED\_AUTO\_ROT, LOW);

   digitalWrite(LED\_AUTO\_GELB, LOW);

   digitalWrite(LED\_AUTO\_GRUEN, HIGH);

   delay(12000);                       // 12 Sekunden warten

 }

}

# Bauanleitung

## Schema

Auf dem folgenden Schema siehst du, wie die verschiedenen Teile miteinander über Kabel verbunden werden müssen.

Achtung: Plus (+) und Minus (-) immer beachten!

