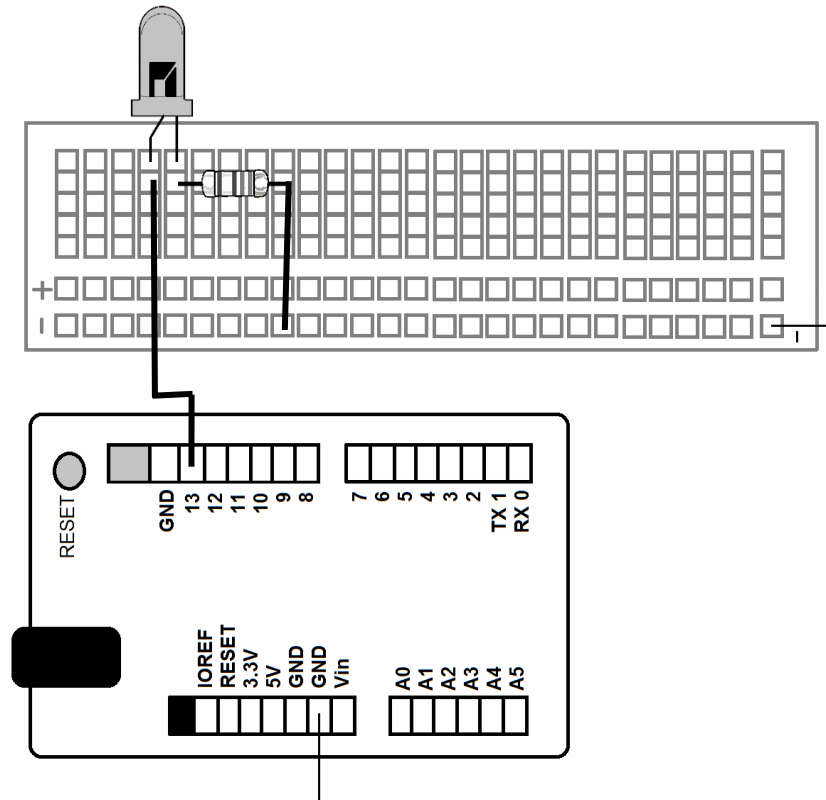


MOTIVATION

Der Bauplan entspricht dem Plan aus dem letzten Modul. Allerdings soll die Lampe beim Blinken nun immer schneller werden.



Grundidee: Die Pausen zwischen dem LED-Leuchten werden immer kürzer.

Dazu braucht man eine Variable, in der die kürzer werdenden Pausenlängen zwischen den Leuchtphasen der LED gespeichert wird. Die Variable nimmt eine ganze Zahl auf, die in jeder Loop-Ausführung verringert wird. Der zugehörige Quellcode lautet:

```
int Pausenlaenge;

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

  Pausenlaenge = 300;
}

void loop() {
  digitalWrite(13, HIGH);
  delay(Pausenlaenge);
  digitalWrite(13, LOW);
  delay(Pausenlaenge);

  Pausenlaenge = Pausenlaenge - 10;
}
```

HINWEIS: Das Blinken wird irgendwann so schnell, dass die LED zu leuchten scheint. Das Problem soll aber erst im nächsten Modul thematisiert werden.

ÜBERSICHT

Umgang mit Variablen

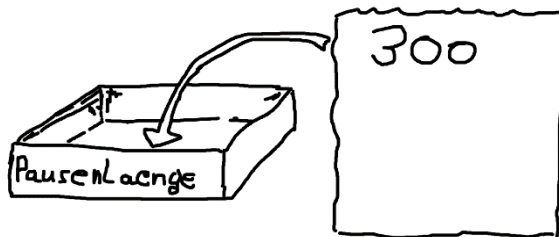
1.) Die Deklaration einer ganzen Zahl namens **Pausenlaenge** geschieht **vor** der Prozedur *setup* und reserviert Speicherplatz für die Variable. Wir stellen uns eine Schublade vor, auf der der Name der Variablen steht.

int Pausenlaenge;



2.) Der Variablen Pauselaenge wird der Wert 300 Millisekunden zugewiesen. Wir stellen uns einen Zettel vor, auf den der Wert 300 geschrieben wird. Der Zettel wird in der Schublade „Pausenlaenge“ abgelegt.

Pausenlaenge = 300;



3.) Bei jeder Ausführung der Prozedur „loop“ wird der Zettel aus der Box entnommen, die 300 werden durchgestrichen und um 10 reduziert wieder auf den Zettel geschrieben. Dann wird der Zettel wieder abgelegt.

Pausenlaenge = Pausenlaenge - 10;



Zufallszahlengenerator

Der Zufallszahlengenerator wird mit

randomSeed(analogRead(5));

initialisiert. Am Pin 5 ist dabei nichts angeschlossen. Dann werden zufällige Geräusche oder Wellen aus der Umgebung aufgefangen und als Zufallszahl interpretiert

int Zufallszahl;

Zufallszahl = random(3,10); // erzeugt eine Zufallszahl zwischen 3 und 9

Zufallszahl = random(6); // erzeugt eine Zufallszahl zwischen 0 und 5.



Der **serielle Monitor** dient zur Ausgabe von Werten auf dem Bildschirm. Mit

```
Serial.begin(9600);
```

wird er in der Prozedur *setup* initialisiert. 9600 legt dabei die Baudrate (Übertragungsgeschwindigkeit--> Bit pro Sekunde) fest. Mit

```
Serial.println(Zufallszahl);
```

können Werte ohne Zeilenumbruch ausgegeben werden, über

```
Serial.print(Zufallszahl);
```

Werte ohne Zeilenumbruch. Das „ln“ bedeutet next Line. Geöffnet wird der serielle Monitor über die **Lupe oben rechts**, nachdem das Programm übertragen worden ist.

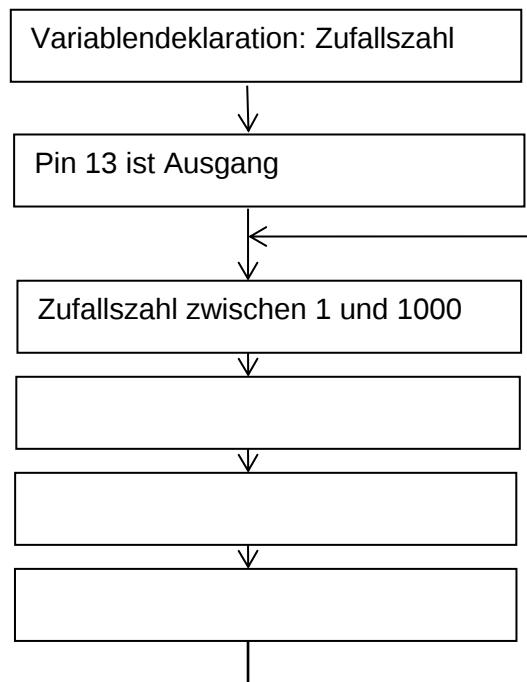


ÜBEN

Aufgabe 1

Nimm den Bauplan mit der einen LED von oben. Es soll nun eine Zufallszahl zwischen 1 und 1000 Sekunde ermittelt werden und die LED für die Zeitdauer dieser Zufallszahl in Millisekunden eingeschaltet werden (mithilfe einer Variablen). Danach bleibt die LED für 1 Sekunde ausgeschaltet.

a) Ergänze dazu zunächst das Flussdiagramm.

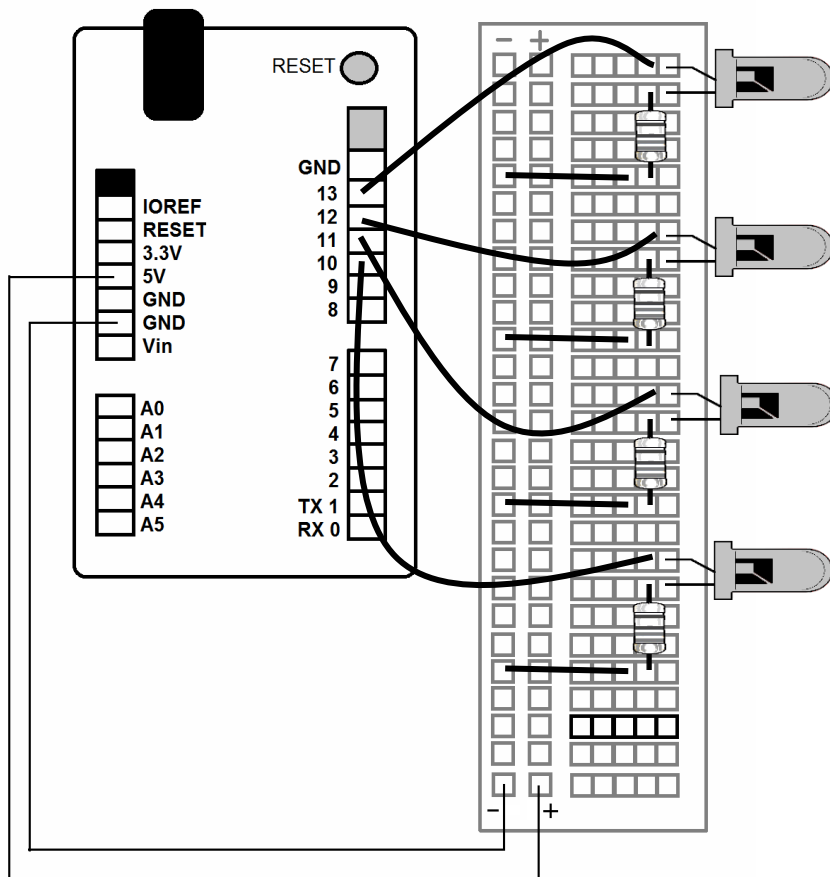


b) Setze das Programm um.

ANWENDEN

Aufgabe 2

Erweitere die LED-Schaltung um 3-4 weitere LEDs gemäß des abgebildeten Schaltplans. Alle 2 Sekunden soll eine Zufallszahl zwischen 1 und 5 erzeugt und in einer Variablen gespeichert werden. Die entsprechende Lampe soll dann eingeschaltet werden, so dass alle 2 Sekunden die LEDs zufällig wechseln.



VERTIEFEN

Aufgabe 3

Erzeuge jede Sekunde eine Zufallszahl zwischen 1 und 10 und lass die Zufallszahl im seriellen Monitor anzeigen.

Aufgabe 4

Eigentlich könnte man ja Zufallszahlen nur mit `(analogRead(5))`; ohne den Zufallszahlengenerator erzeugen.

- Teste die zufällige Ausgabe von `(analogRead(5))`; mithilfe des seriellen Monitors.
- Erläutere, warum man trotzdem nicht auf den Zufallszahlengenerator verzichten kann.