



# Installation heißer Draht 3.0

auf Grundlage des Kurses 2.0

Stand 2020



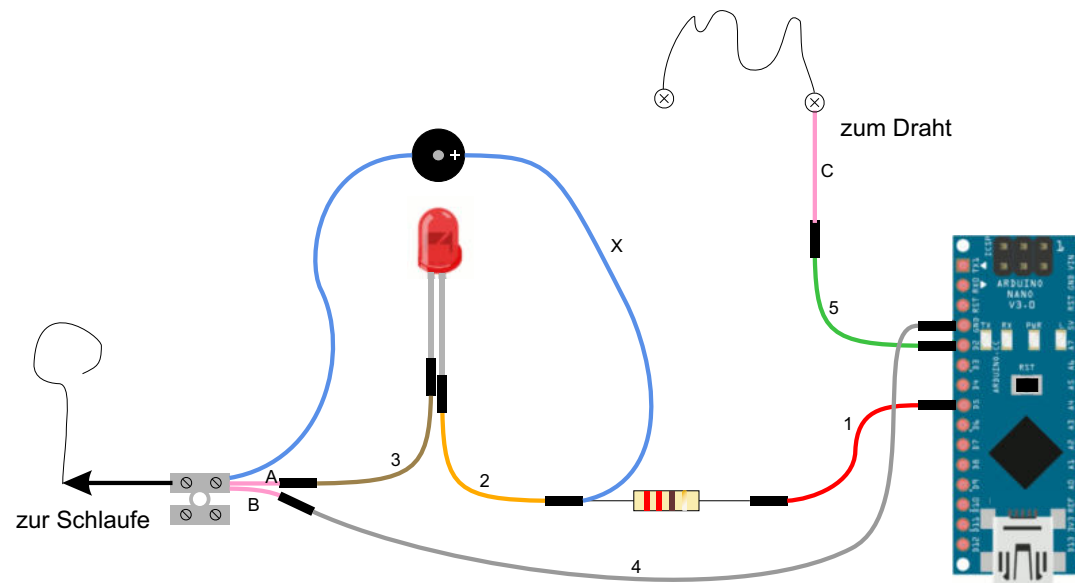
## Problem

- Es wäre wünschenswert, wenn jede die gleiche Zeit zur Verfügung bekommt, also der Arduino nach einstellbarer Zeit ein Endsignal setzen könnte.
- Das Spiel sollte mit uns kommunizieren können, z.B. schreiben, wann es los geht, ob ein Fehler gemacht wurde und wann das Programm zu Ende ist.

## Verdrahtung

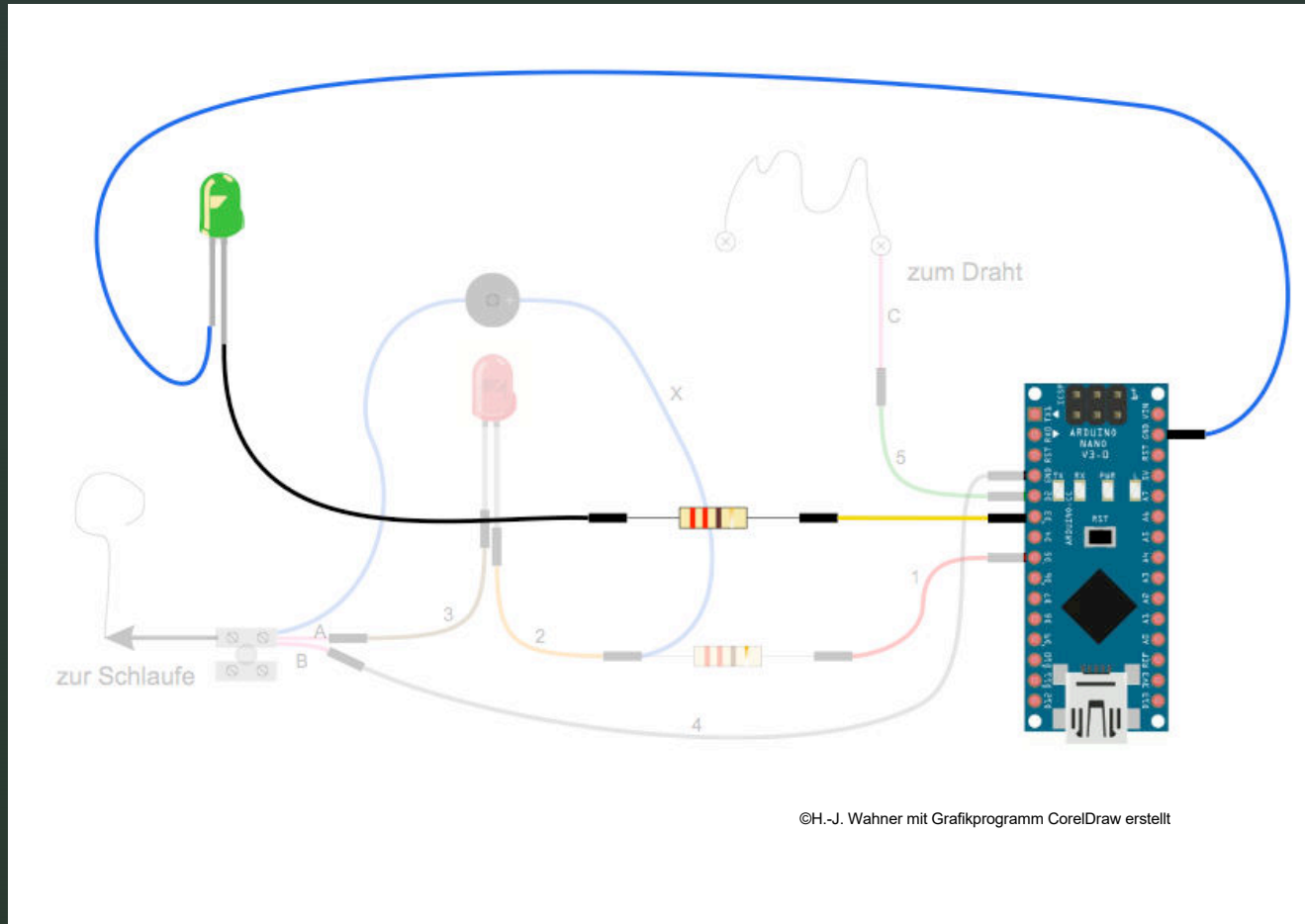
1. Du brauchst
  1. Zweite LED
  2. Widerstand (220 Ohm)
  3. Drei Jumperkabel
2. Verbinde das erste Jumperkabel 1 (hier gelb) mit PIN3
3. Auf die andere Seite des Jumperkabel 1 kommt der Widerstand.
4. Der zweite Anschluss des Widerstands wird in das Jumperkabel 2 (hier schwarz) gesteckt.
5. An das andere Ende des Jumperkabels 2 kommt das lange Anschlussbein der Leuchtdiode (Plusseite der Diode)
6. Das kurze Anschlussbein der Leuchtdiode wird mit dem Jumperkabel 3 (hier blau) verbunden
7. Das andere Ende des Jumperkabel 3 wird auf den GRD-Anschluss am Arduino gesteckt

## So sieht es vor dem Umbau aus:



©H.-J. Wahner mit Grafikprogramm CorelDraw erstellt

## So sieht es nun ergänzt aus:



## ■ Was muss programmiert werden?

- Der Arduino soll einen Timer starten, wenn das Programm losgeht. Wenn dieser Timer abgelaufen ist, leuchtet eine weitere LED
- Der digitale Monitor der Arduino IDE am PC soll zu bestimmten Zeiten Texte aufgeben, die gelesen werden können.

## Wie wird das Programm verändert?

- Dem Arduino muss noch erklärt werden, dass die weitere LED an Pin 3 angeschlossen wird und er Zeitinformationen aus dem Befehl `Millis` entnehmen soll.
- 1. Wenn die Zeit abgelaufen ist, soll der Pin 3 auf High gesetzt werden
- 2. An drei Programmstellen soll ein Text im Monitor erscheinen:
  - Beim Start des Spiels
  - Wenn ein Fehler unterlaufen ist
  - Wenn das Spiel zuende ist



## ■ Aufbau des Sketches

- Ein Sketch besteht aus drei Teilen:
- 1. Festlegen einer weiteren Variablen für LED
- 2. Festlegen von dem Anschluss sowie Starten Monitor
- 3. Programmergeänzungen im Loop



## Unser Sketch:

```
bool kontakt = false;
int drahtPin = 2;
int ledPin = 5;
int ledStop = 3;

void setup()
{
  pinMode(drahtPin, INPUT_PULLUP);
  pinMode(ledPin, OUTPUT);
  pinMode(ledStop, OUTPUT);
  Serial.begin(300);
  Serial.println(F("Starte mit dem Spiel."));
}

void loop()
{
  if (digitalRead(drahtPin) == LOW)
  {
    if (kontakt == false)
    {
      kontakt = true;
    }
    else
    {
      kontakt = false;
    }
  }

  if (kontakt == true)
  {
    digitalWrite(ledPin, HIGH);
    Serial.println(F("Fehler!"));
  }

  if (kontakt == false)
  {
    digitalWrite(ledPin, LOW);
  }

  delay(200);

  if (millis() > 5000)
  {
    digitalWrite(ledStop, HIGH);
    Serial.print(F("Ende!"));
  }
}

// eine Variable zum Speichern des Schaltzustandes vom Typ wahr/falsch
// eine Variable namens drahtPin ist mit dem Anschluss Pin2 verbunden
// eine Variable namens ledPin ist mit dem Anschluss Pin5 verbunden
// NEU: eine Variable namens ledStop ist mit dem Anschluss Pin3 verbunden

// festlegen von Anschlüssen
// hier beginnt das setup
// drahtPin ist ein Eingang und zusätzlich mit Pull-up-Widerstand versehen
// ledPin ist ein Ausgang
// NEU: ledStop ist ein Ausgang
// NEU: Starte Monitor mit Baudrate 300
// NEU: Schreibe auf Monitor Starte... verwende Flash nicht RAM
// hier endet das setup

// hier beginnt die Endlosschleife
// hier beginnt das Programm

// wenn Drahtschleife und Draht sich berühren
// hier beginnt ein Programmabschnitt logischer Speicher
// wenn die Variable geschaltet auf falsch steht
// hier beginnt ein Unterprogramm
// lege die Variable geschaltet auf wahr
// hier endet ein Unterprogramm
// ansonsten also wenn die Variable geschaltet nicht auf falsch steht
// hier beginnt ein Programmabschnitt
// lege die Variable geschaltet auf falsch
// hier endet ein Programmabschnitt
// hier endet ein Programmabschnitt logischer Speicher

// wenn die Variable geschaltet wahr ist
// hier beginnt ein Programmabschnitt Einschalten Ausgang
// schalte den Anschluss ledPin also Pin5 auf ein
// NEU: Schreibe auf Monitor Fehler!
// hier endet ein Programmabschnitt Einschalten Ausgang

// wenn die Variable geschaltet unwahr ist
// hier beginnt ein Programmabschnitt Ausschalten Ausgang
// schalte den Anschluss ledPin also Pin5 auf aus
// hier endet ein Programmabschnitt Ausschalten Ausgang

// warte 200/1000 Sekunden Abtastempfindlichkeit

// NEU: Es wird die Zeit seit Programmstart gemessen mit Millisekunden
// NEU: Hier beginnt der Programmabschnitt Ende des Spiels
// NEU: Die ledStop wird eingeschaltet
// NEU: Schreibe auf Monitor: Ende!
// NEU: hier endet die Bedingung

// hier endet die Schleife und beginnt von vorne
```

gefördert von:



Baden-Württemberg

MINISTERIUM FÜR WIRTSCHAFT, ARBEIT UND WOHNUNGSBAU