



Installation heißer Draht 2.0

auf Grundlage des GDC-Sets

Stand 2020



■ Problem

- Es gibt Streit darum, ob die Schlaufe den Draht berührt hat. Eine hat den Signalton wahrgenommen, die andere nicht. Es wäre daher praktisch, wenn sich der Heiße Draht einen Kontakt merken könnte.
- Wenn man die Schwierigkeit ändern will, so muss Schlaufe oder Draht gebogen werden. Es wäre doch schön, wenn sich das einfach eingeben lassen könnte.

Was muss programmiert werden?

- Der Arduino soll sich merken, ob es Kontakt zwischen Schlaufe und Draht gab. Wir brauchen also einen Speicher, der sich das merkt. Dieser Speicher braucht zwei Schaltzustände: 1. Es ist wahr, dass es Kontakt gab und 2. Es ist unwahr, gerade gab es keinen Kontakt.
- Der Arduino soll in einer schnellen Abfolge prüfen, ob es zu einem Kontakt kam. Die Zeitabstände sollen jedoch einstellbar sein – so kann man einstellen, wie empfindlich die Schaltung reagiert.

Wie sieht das Programm aus?

- Dem Arduino muss noch erklärt werden, woher er Informationen bekommt und wohin er welche abgibt. Aufgrund unserer Verkabelung ist der Dateninput bei Pin 2 und der -output bei Pin 5.
- Jetzt muss dem Arduino noch der Ablauf erklären werden: Er soll solange arbeiten, bis wir ihn wieder abstellen, das nennt man (Endlos-) schleife. Dort soll er ständig folgendes wiederholen:
 - Er soll am Input immer wieder neu feststellen, ob eine Kontakt stattgefunden hat.
 - Wenn er das feststellt, soll er sich merken, dass Kontakt stattfand.
 - Wenn Kontakt stattfand, soll er den Output ansteuern
 - Wenn er aber wieder feststellt, dass Kontakt stattfand, soll er dem Output wieder abschalten und seinen Speicher zurücksetzen
 - Bevor er diesen Ablauf wiederholt, soll er einige Millisekunden warten

■ Aufbau eines Sketches

- Ein Sketch besteht aus drei Teilen:
- 1. Festlegen von Variablen
- 2. Festlegen von Anschlüssen im Setup
- 3. Programmablauf in einer Endlosschleife (Loop)

Unser Sketch:

```
bool kontakt = false; // eine Variable zum Speichern des Schaltzustandes vom Typ wahr/falsch
int drahtPin = 2;      // eine Variable namens drahtPin ist mit dem Anschluss Pin2 verbunden
int ledPin = 5;        // eine Variable namens ledPin ist mit dem Anschluss Pin5 verbunden

void setup()           // festlegen von Anschlüssen
{                       // hier beginnt das setup
  pinMode( drahtPin, INPUT_PULLUP); // drahtPin ist ein Eingang und zusätzlich mit Pull-up-Widerstand versehen
  pinMode( ledPin, OUTPUT);        // ledPin ist ein Ausgang
}                           // hier endet das setup

void loop()            // hier beginnt die Endlosschleife
{                       // hier beginnt das Programm
  if (digitalRead(drahtPin) == LOW) // wenn Drahtschleife und Draht sich berühren
  {                       // hier beginnt ein Programmabschnitt logischer Speicher
    if (kontakt == false) // wenn die Variable geschaltet auf falsch steht
    {                       // hier beginnt ein Unterprogramm
      kontakt = true;      // lege die Variable geschaltet auf wahr
    }                       // hier endet ein Unterprogramm
    else                   // ansonsten also wenn die Variable geschaltet nicht auf falsch steht
    {                       // hier beginnt ein Programmabschnitt
      kontakt = false;    // lege die Variable geschaltet auf falsch
    }                       // hier endet ein Programmabschnitt
  }                       // hier endet ein Programmabschnitt logischer Speicher

  if (kontakt == true)    // wenn die Variable geschaltet wahr ist
  {                       // hier beginnt ein Programmabschnitt Einschalten Ausgang
    digitalWrite(ledPin, HIGH); // schalte den Anschluss ledPin also Pin5 auf ein
  }                       // hier endet ein Programmabschnitt Einschalten Ausgang

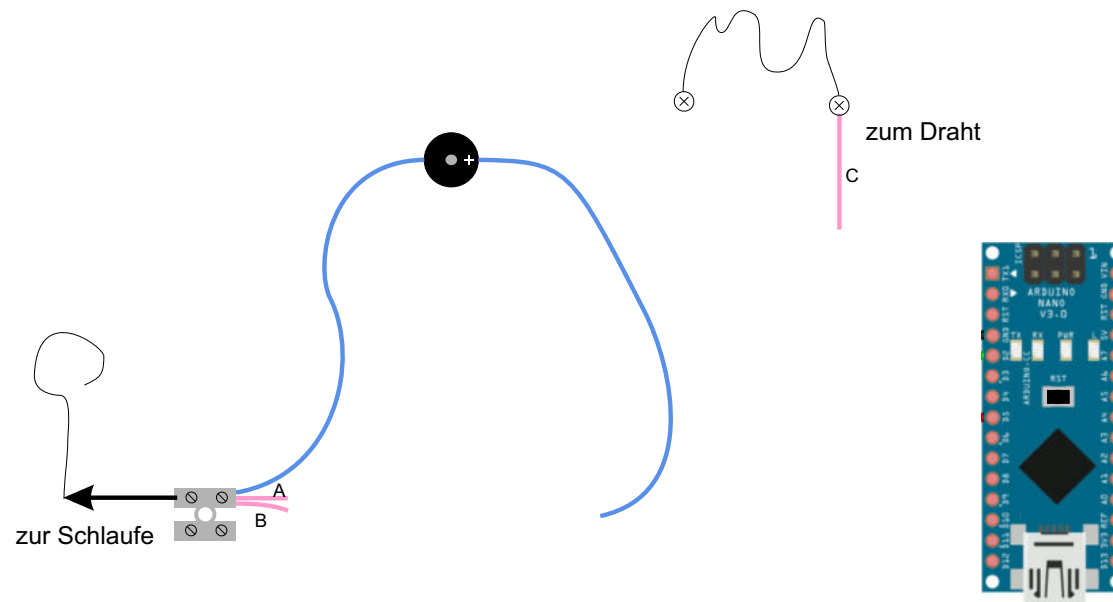
  if (kontakt == false)  // wenn die Variable geschaltet unwahr ist
  {                       // hier beginnt ein Programmabschnitt Ausschalten Ausgang
    digitalWrite(ledPin, LOW);  // schalte den Anschluss ledPin also Pin5 auf aus
  }                       // hier endet ein Programmabschnitt Ausschalten Ausgang

  delay(200);            // warte 200/1000 Sekunden Abtastempfindlichkeit
}                       // hier endet die Schleife und beginnt von vorne
```


Verdrahtung

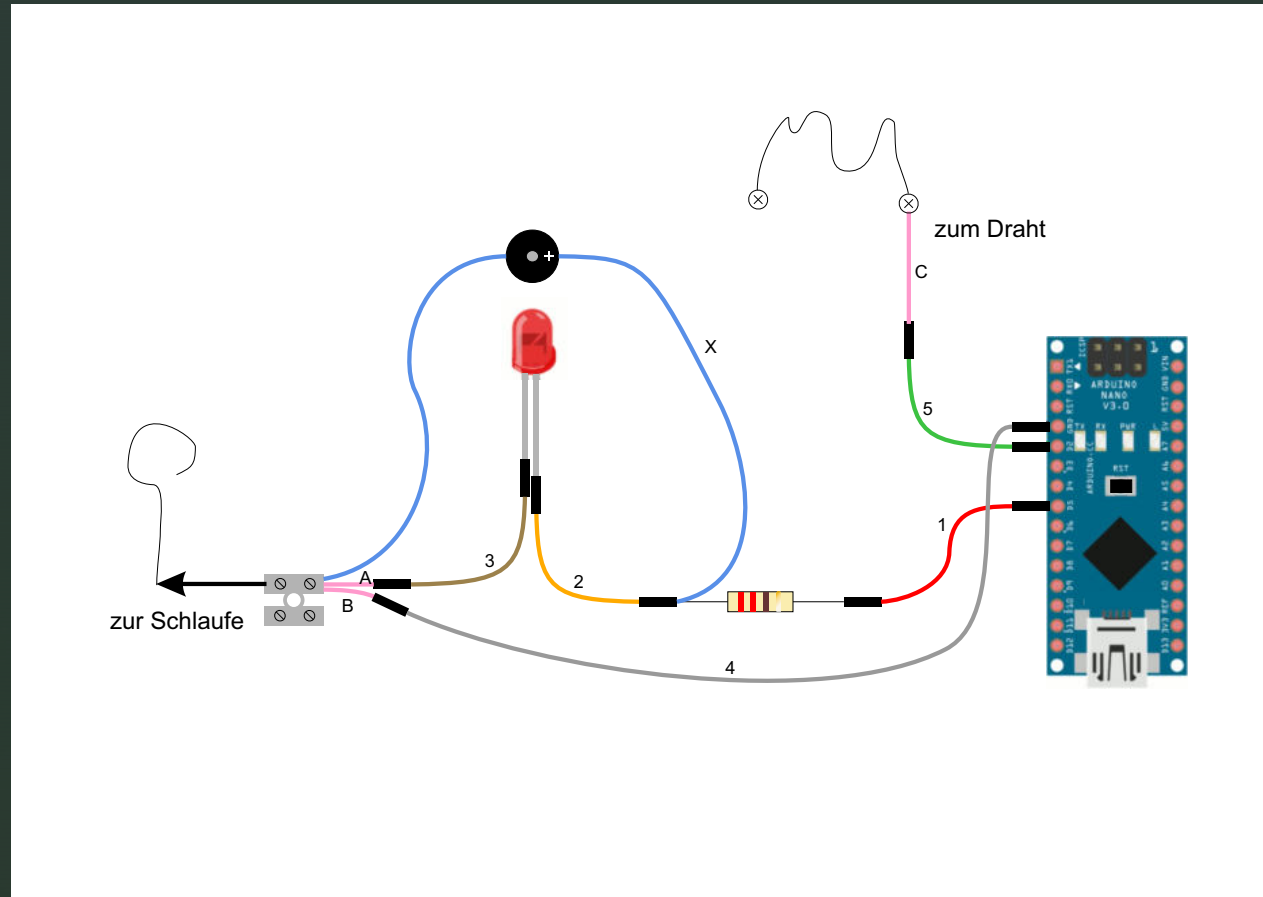
1. Entferne alle Kabel vom Heißen Draht 1.0 außer deinen Schlaufenanschluss (siehe Umbau.pdf)
2. Schließe an die Lüsterklemme rechts drei Kabel an:
 1. zwei Drahtstücke A und B (hier rosa) und
 2. den Minusanschluss des Tonmoduls (Piezzo)
3. Klemme unter eine Schraube vom Draht ein Drahtstück C an (hier rosa)
4. Nun stecke das Jumperkabel 1 (hier rot) von hinten an den Anschluss D5 des Arduinos (siehe Fertig.pdf)
5. Auf die andere Seite des Jumperkabel 1 kommt der Widerstand.
6. Der zweite Anschluss des Widerstands wird in das Jumperkabel 2 (hier gelb) gesteckt.
7. An das andere Ende des Jumperkabels 2 kommt das lange Anschlussbein der Leuchtdiode (Plusseite der Diode)
8. Das kurze Anschlussbein der Leuchtdiode wird mit dem Jumperkabel 3 (hier braun) verbunden
9. Das Jumperkabel 3 wird dann noch auf das Drahtstück A aufgesteckt
10. Das Drahtstück B wird mit dem Jumperkabel 4 (hier grau) verbunden
11. Das Jumperkabel 4 kommt noch von hinten auf den Anschluss GND des Arduino
12. Verbinde das Jumperkabel 5 (hier grün) von hinten mit dem Anschluss D2 des Arduino
13. Das Jumperkabel 5 wird dann noch auf das Drahtstück C aufgesteckt
14. Es fehlt noch der Anschluss X (hier blau), der zusätzlich zum Widerstand in das Jumperkabel 2 (hier gelb) eingesteckt wird

So sieht der Umbau aus:



©H.-J. Wahner mit Grafikprogramm CorelDraw erstellt

So sieht es fertig verdrahtet aus:



gefördert von:



Baden-Württemberg

MINISTERIUM FÜR WIRTSCHAFT, ARBEIT UND WOHNUNGSBAU