

Effektvoller Blinker mit 12 roten LED

Josef Straßhofer

```
// Blinker - 12 LED mit Arduino Nano - Atmega328P

int port[12] = {2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13};
int zeilenposition = 0;
const int zeilenmax = 13; // Zeilenanzahl der Bitmuster
int bitmuster[zeilenmax][13] = {
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 100}, // Zeit in ms
  {1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 50},
  {1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 50},
  {1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 50},
  {1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 50},
  {1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 50},
  {1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 50},
  {1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 50},
  {1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 50},
  {1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 50},
  {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 50},
  {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 50},
  {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 300}
};

void setup() {
  for (int i = 0; i < 12; i++) {
    pinMode(port[i], OUTPUT);
  }
}

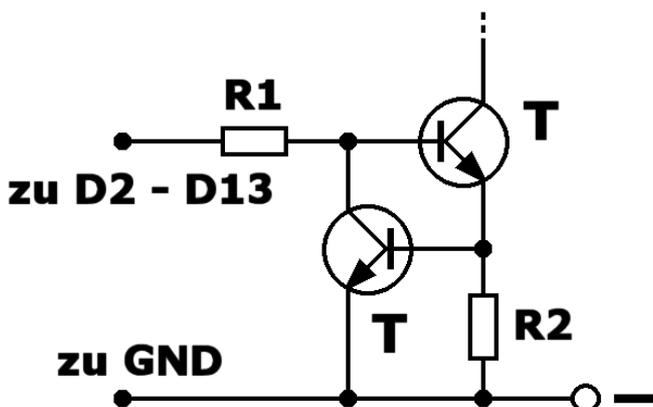
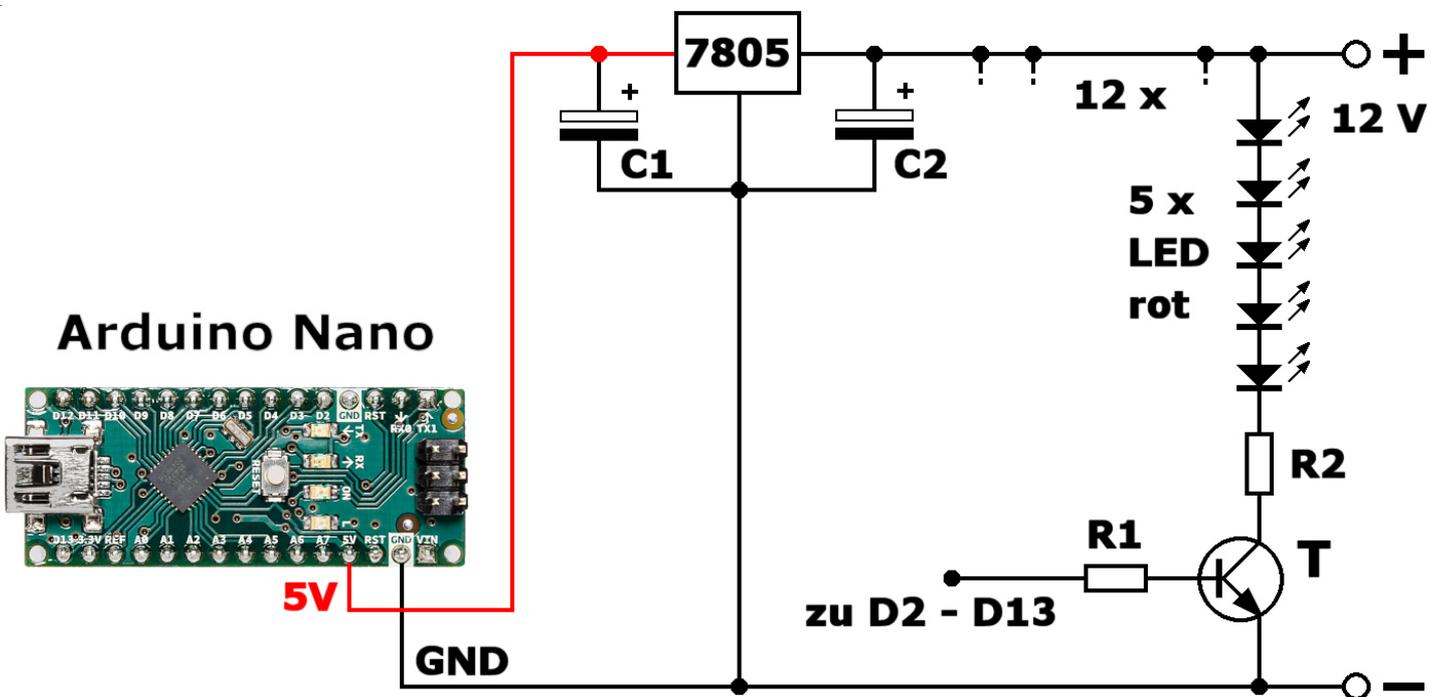
void showbitmuster() {
  for (int i = 0; i < 12; i++) {
    digitalWrite(port[i], bitmuster[zeilenposition][i]);
  }
  delay(bitmuster[zeilenposition][12]); // Wartezeit
}

void loop() {
  showbitmuster();
  zeilenposition++;
  if (zeilenposition == zeilenmax) zeilenposition = 0;
}
```

Diese Schaltung für den Arduino Nano zeigt einen Typ **Blinker**, wie man ihn von einigen Autos kennt. Er kann individuell angepasst bzw. auch noch erweitert werden. Das Bitmuster wird schrittweise ausgegeben, wobei wegen der besseren Übersicht die Verweilzeit unmittelbar neben dem Bitmuster eingetragen wird. Bei Änderungen ist lediglich darauf zu achten, dass die richtige Zeilenanzahl (const int zeilenmax = 13;) der aufeinanderfolgenden Bitmuster eingetragen werden muss. In diesem Fall sind es 13 Zeilen. Mit geringem Aufwand ist es z.B. auch möglich, eine **Ampelschaltung** einer kompletten Kreuzung mit Fußgängerübergang zu gestalten. Sogar Links-Abbiegespuren gehen sich mit den hier verwendeten 12 Ausgängen des Arduino Nano aus.

Schaltungsvorschlag zur Ansteuerung mehrerer LED pro Ausgang

Für den Arduino Nano gibt es mehrere Möglichkeiten der Stromversorgung, wenn er nicht über USB an den Computer angeschlossen ist. Pro Ausgang darf der Arduino nicht mit mehr als **40 mA** belastet werden, die Summe aller Ausgänge darf allerdings auch **200 mA** nicht überschreiten, wobei der Nano selbst etwa **20 mA** benötigt. Werden die LED direkt an die Ausgänge des Arduino D2 bis D13 angeschlossen, so wird jeweils ein Vorwiderstand mit **270 Ohm** verwendet. Damit ist man mit 12 LED absolut auf der sicheren Seite. Die folgende Abbildung zeigt eine Variante, wie man einen Arduino mit nahezu jeder Menge an LED beschalten kann. Die Dimensionierung mit je 5 roten LED pro Ausgang bei einer stabilisierten Betriebsspannung von 12 Volt wäre hier: T = BC 547B, R1 = 10 kOhm, R2 = 150 Ohm, C1 + C2 jeweils 100 μF und der Festspannungsregler 7805. Damit versorgt man den Arduino mit 5 Volt, während man in der Peripherie jede Menge an Leuchtdioden betreiben kann. Nähere Informationen zur Beschaltung und Verwendung anderer Transistoren, (BC 337-25, BD 139-16, mehr LED), gibt es in meinem Skriptum „Ultrahelle Leuchtdioden und ihre praktische Anwendung“ und in meinem Buch „Elektronik mit Herz“.



Wird keine elektronisch stabilisierte Betriebsspannung benützt, so empfiehlt sich die Verwendung von **Konstantstromquellen** für die einzelnen LED-Reihen. Für den Transistor T wird hier in beiden Fällen wieder der BC 547B eingesetzt, für den Widerstand R2 empfiehlt sich ein Wert mit 33 Ohm.

Werden **LED-Ketten parallel** geschaltet, so sollte pro Kette je ein (kleiner) Vorwiderstand vorgeschaltet werden. Die übrigen Bauelemente sind dann entsprechend dem Strom anzupassen. Grundsätzlich ist es möglich, auf diese Weise mehrere 100 Leuchtdioden anzusteuern, sofern man über eine entsprechende Stromversorgung verfügt.