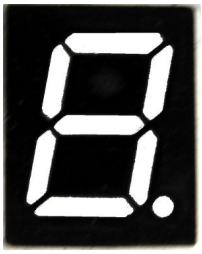
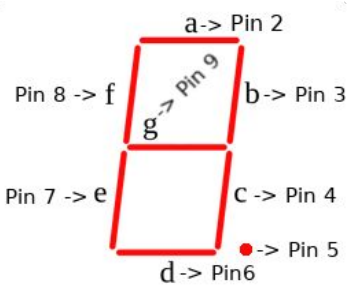
Countdown mit einer einstelligen 7-Segment-Anzeige

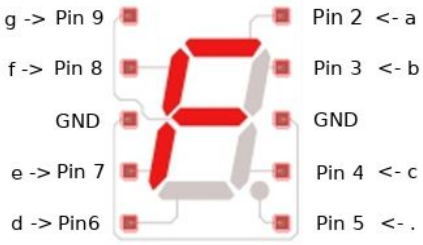


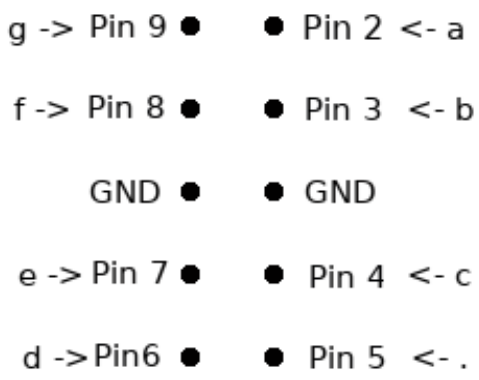
Die 7-Segment-Anzeige besteht aus sieben horizontal und vertikal verlaufenden Segmenten und einem Punkt in der rechten unteren Ecke, die einzeln angesteuert werden. Es lassen sich alle Zahlen und eine Reihe von Buchstaben darstellen.

0123456789



Die Segmente sind von a bis g gekennzeichnet. Jedes Segment muss mit einem Pin des Arduinos verbunden werden.





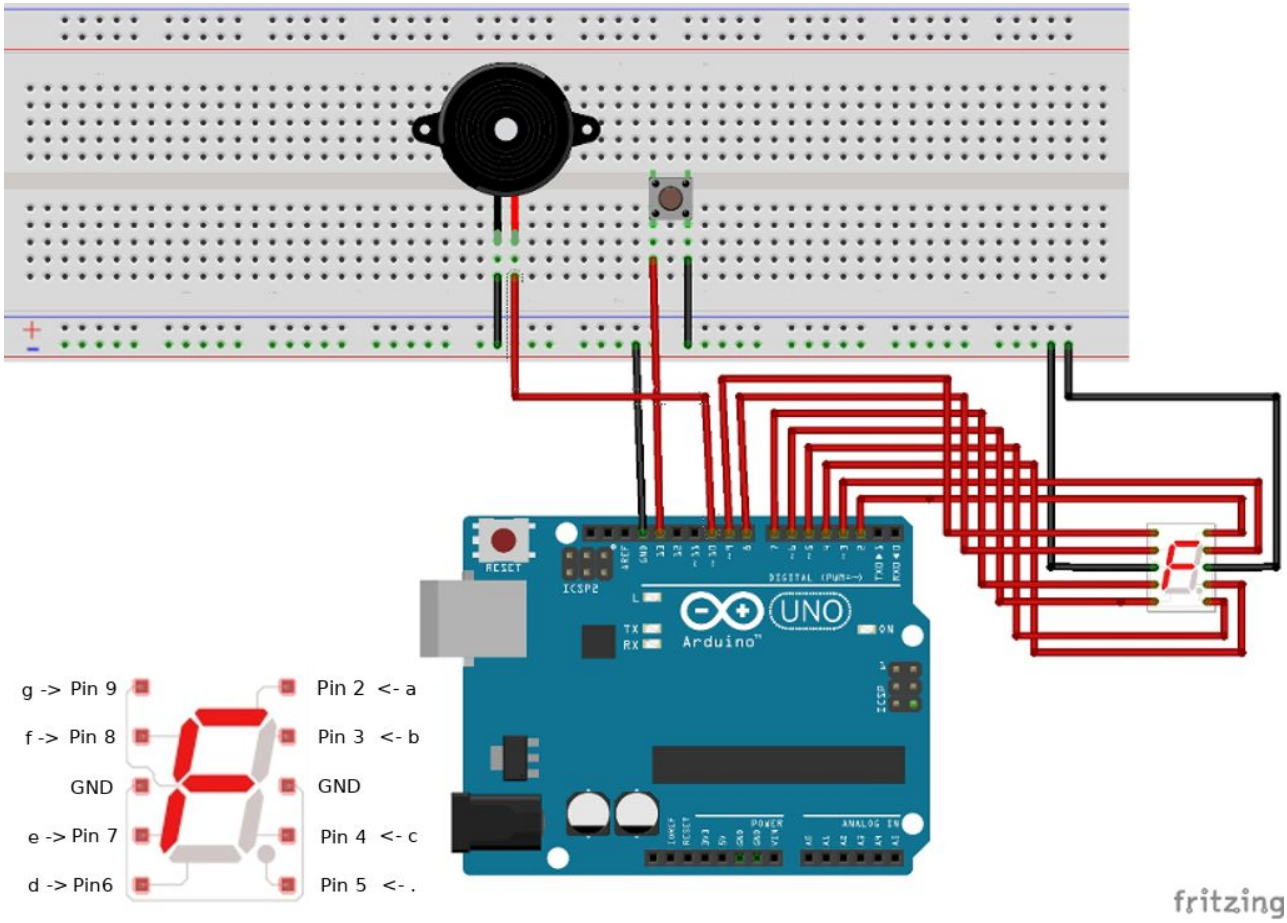
Es gibt die 7-Segmente-Anzeige in zwei Ausführungen:
entweder - (Common Cathode → GND) oder + (Common Anode → 5V).
Die verwendete Version kannst du durch einfaches Umstecken (GND/5V) herausfinden.

Das Programm soll nach einem Tasterdruck von 9 bis 0 herunterzählen. Zu jedem Wechsel der Zahl wird ein kurzer Ton wiedergegeben. Der Ablauf des Countdowns wird zusätzlich mit dem Abspielen einer zufällig erzeugten Melodie ergänzt.

Benötigte Bauteile:

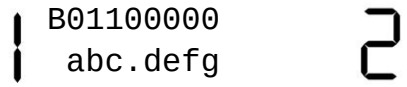
- Taster
- Einstellige 7-Segment-Anzeige
- Lautsprecher
- Leitungsdrähte

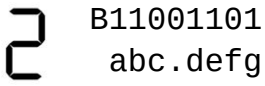
Baue die Schaltung auf.



Die Zahlen, die dargestellt werden sollen, werden als Binärwert notiert. Eine 1 steht für Segment einschalten, eine 0 zeigt das Segment nicht an. Die Reihenfolge der Ziffern entspricht der Reihenfolge der Pins. Die erste Ziffer schaltet Pin 2, die zweite Pin 3 und die letzte Pin 9.

Beispiele:





Lege die Variablen fest.

```
byte Zahlen[10] =
{
  B11101011, // 9
  B11101111, // 8
  B11100000, // 7
  B10101111, // 6
  B10101011, // 5
  B01100011, // 4
  B11101001, // 3
  B11001101, // 2
  B01100000, // 1
  B11101110, // 0
};
```

```
int TASTER = 13;
int LAUTSPRECHER = 10;

// der Frequenzbereich der Töne
int Minimum = 500;
int Maximum = 2000;
```

Der setup-Teil. Beachte die Kommentare.

```
void setup()
{
  // Pins auf OUTPUT setzen
  for (int i = 2; i <= 9; i++)
  {
    pinMode(i, OUTPUT);
  }

  pinMode(TASTER, INPUT_PULLUP);

  // Zufallsgenerator starten
  randomSeed(analogRead(0));

  // A (B11100111) anzeigen
  ZahlZeigen(B11100111);
}
```

Der loop-Teil. Beachte die Kommentare.

```
void loop()
{
  int TasterLesen = digitalRead(SENSOR);
  if (TasterLesen == LOW)
  {
    for (int i = 0; i <= sizeof(Zahlen) - 1; i++)
    {
      // aktuelles Array i (Zahlen als Folge von Bits)
      // an Methode ZahlZeigen übergeben
      ZahlZeigen(Zahlen[i]);
      tone(LAUTSPRECHER, 1000, 10);
      delay(1000);
    }

    // Countdown abgelaufen zufällige Tonfolge spielen
    for (int i = 0; i < 10; i++)
    {
      tone(LAUTSPRECHER, random(Minimum, Maximum), 500);
      delay(200);
    }
    // A (B11100111) anzeigen
    ZahlZeigen(B11100111);
  }
}
```

Im loop-Teil wird die Methode `ZahlZeigen()` aufgerufen. Als Parameter wird ihr ein Element des Arrays Zahlen – eine der Binärwerte für die Zahl 9 bis 0 – übergeben.

```
void ZahlZeigen(byte ArrayZahl)
{
  // Bits des Arrays ArrayZahl prüfen
  // von Pin 2 bis Pin 9 durchlaufen
  for (int i = 2; i <= 9; i++)
  {
    /*
      vergleicht das Byte ArrayZahl mit dem Byte B10000000
      befindet sich an beiden Positionen eine 1
      das Ergebnis der Prüfung ist also nicht 0
      -> Segment einschalten
      ist eine der Positionen eine 0
      das Ergebnis der Prüfung ist 0
      -> Segment ausschalten
      1 Bit nach links schieben -> nächstes Bit prüfen
      nach 8 Durchläufen sind alle Segmente (Pins) richtig geschaltet
    */
    if ((ArrayZahl & B10000000) != 0) digitalWrite(i, HIGH);
    else  digitalWrite(i, LOW);
    ArrayZahl = ArrayZahl << 1;
  }
}
```