AUTODESK®
TINKERCAD®

Register

Programmieren Sie eine LED-Lichtshow

Circuit design 8 LED Chase (digit | Circuit design Copy of 8 LED Cha | Circuit design Example Arduino L | Circuit design LED Chase with Re

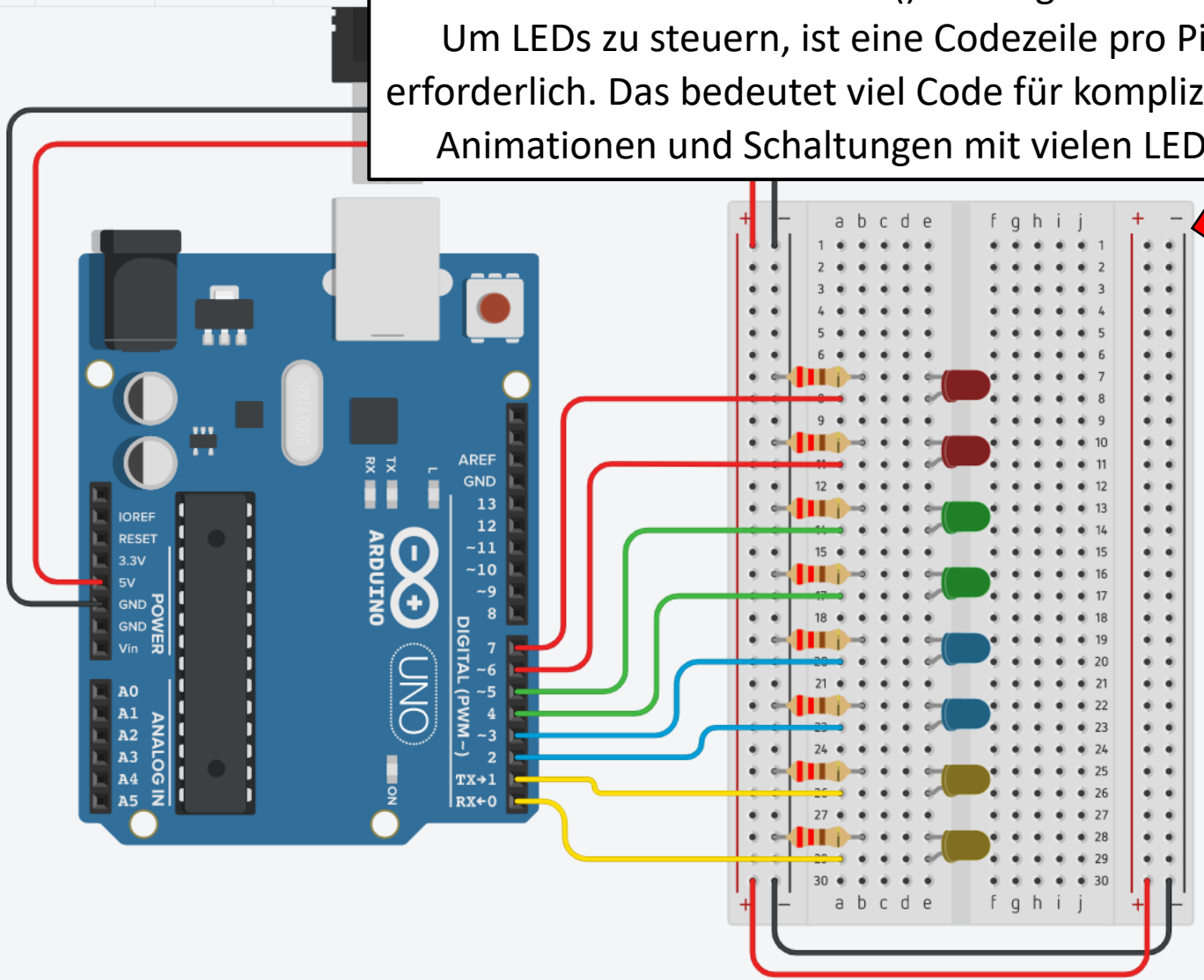
tinkercad.com/things/dcoobTV5YGM-stunning-fyyran/editel?tenant=circuits

8 LED Chase (digitalWrite)

All changes saved

Code | Start Simulation | Export | Share

Verwenden von PinModus() und digitalWrite()
Um LEDs zu steuern, ist eine Codezeile pro Pin erforderlich. Das bedeutet viel Code für komplizierte Animationen und Schaltungen mit vielen LEDs.



Text | 1 (Arduino Uno R3)

```
// C++ code
//

int t = 100; // declare a variable to use for delay

void setup()
{
  // setup code that only runs once
  // set pins 0-7 as outputs
  pinMode(0, OUTPUT);
  pinMode(1, OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
}

void loop()
{
  // code that loops forever
  // note that each LED requires its own line of code
  // turn pin 0 on, others low
  digitalWrite(0, HIGH);
  digitalWrite(1, LOW);
  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  delay(t); // Wait for t millisecond(s)
  // turn pin 1 on, others low
  digitalWrite(0, LOW);
  digitalWrite(1, HIGH);
  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
```

Serial Monitor

Circuit design 8 LED Chase (digit | Circuit design Copy of 8 LED Cha | Circuit design Example Arduino L | Circuit design LED Chase with Re

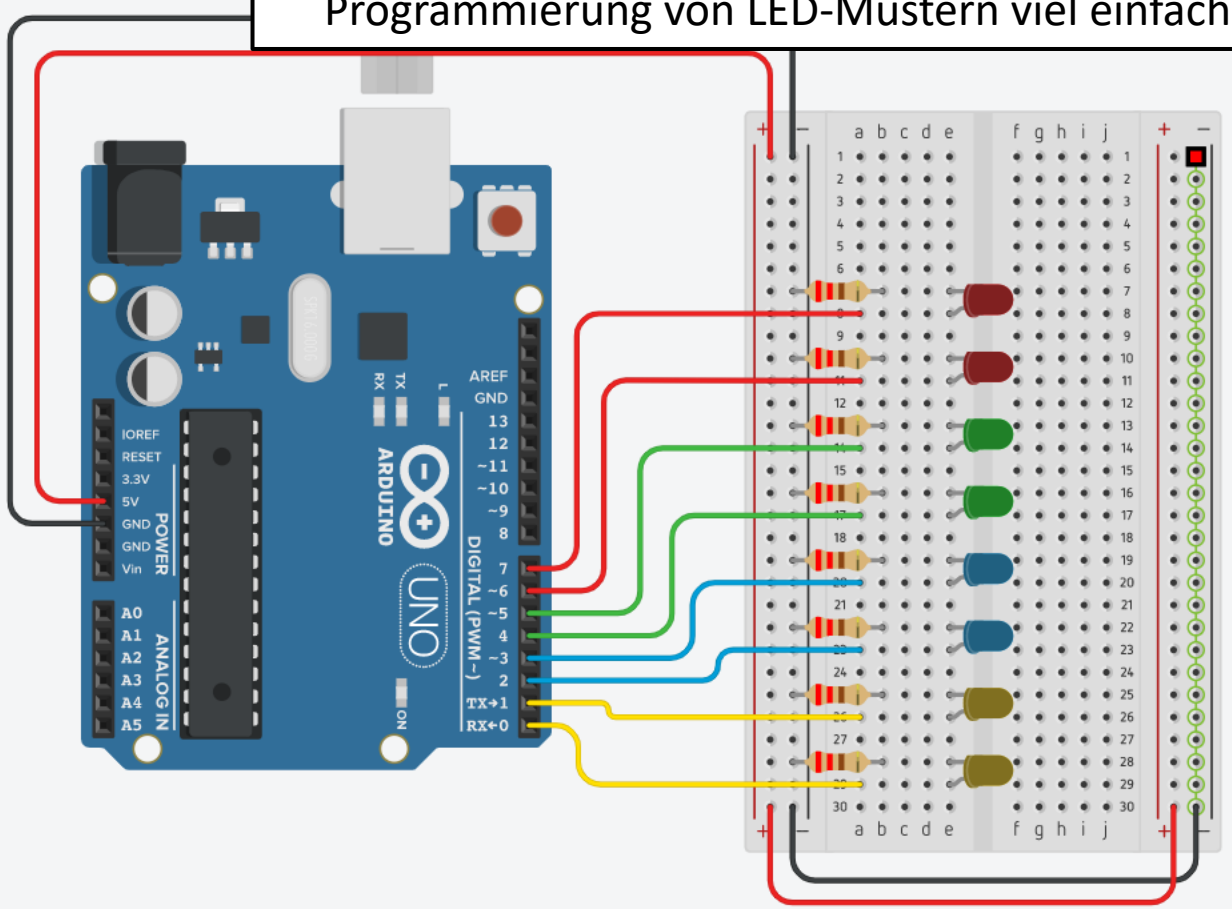
tinkercad.com/things/heTik1InbRL-copy-of-8-led-chase-digitalwrite/editel?tenant=circuits

8 LED Chase (registers)

All changes saved

Code | Start Simulation | Export | Share

Sie können verwenden **registriert** um 8 LEDs mit einer einzigen Codezeile zu steuern! Dies macht die Programmierung von LED-Mustern viel einfacher.

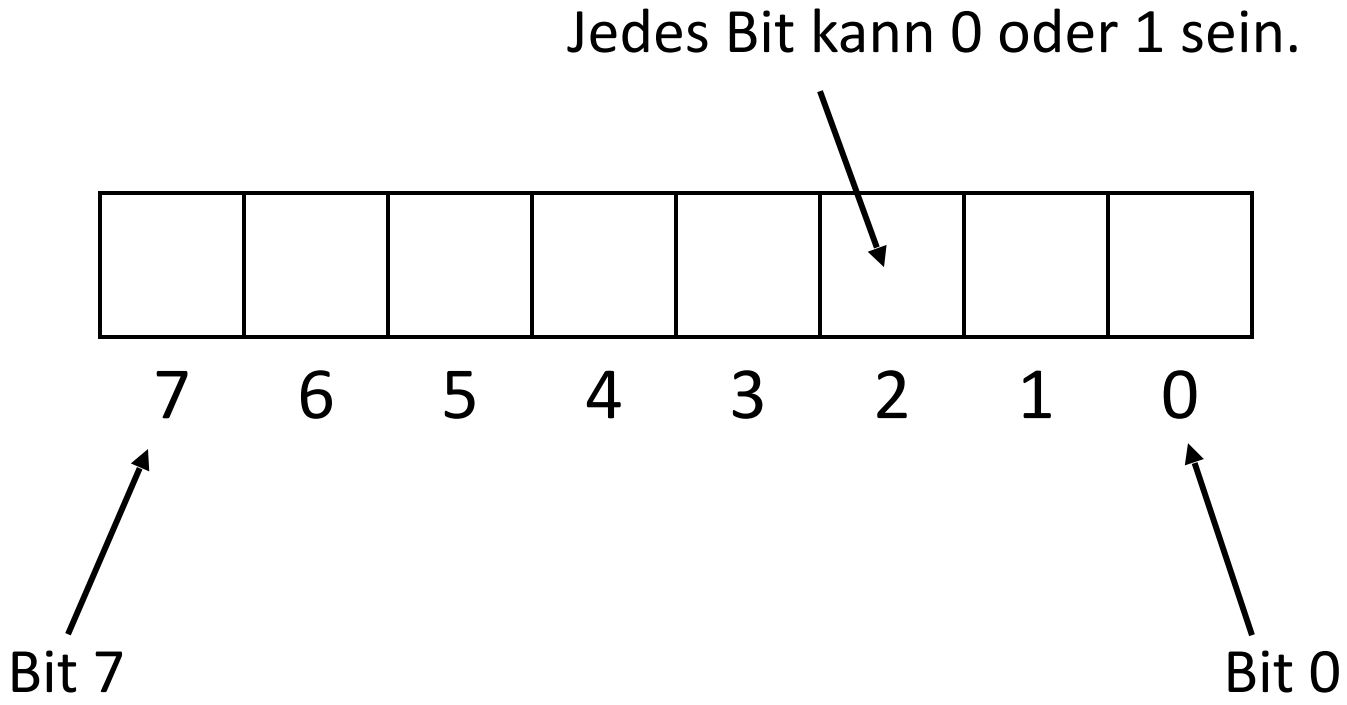


Text | 1 (Arduino Uno R3)

```
// C++ code
//

int t = 100;  // declare a variable to use for the delay

void setup()
{
  // setup code that only runs once
  DDRD = 0b11111111; // set all pins on port D (pins 0-7) as output
}

void loop()
{
  // code that loops forever
  // control all 8 LEDs with one line of code!
  // count the bits from right to left, starting at 0
  // bit #: 76543210
  PORTD = 0b00000001;  // turn pin 0 on, others off
  delay(t);            // Wait for t milliseconds
  PORTD = 0b00000010;  // turn pin 1 on, others off
  delay(t);            // Wait for t milliseconds
  PORTD = 0b00000100;  // turn pin 2 on, others off
  delay(t);            // Wait for t milliseconds
  PORTD = 0b00001000;  // turn pin 3 on, others off
  delay(t);            // Wait for t milliseconds
  PORTD = 0b00010000;  // turn pin 4 on, others off
  delay(t);            // Wait for t milliseconds
  PORTD = 0b00100000;  // turn pin 5 on, others off
  delay(t);            // Wait for t milliseconds
  PORTD = 0b01000000;  // turn pin 6 on, others off
  delay(t);            // Wait for t millisecond(s)
  PORTD = 0b10000000;  // turn pin 7 on, others off
  delay(t);            // Wait for t millisecond(s)
}
```

Serial Monitor

Jedes Register hat 8 Bits:



Zählen Sie die Bits von rechts nach links, beginnend bei 0.

Die Bits im Register entsprechen physischen Pins auf dem Arduino:

Bit 7
Bit 6
Bit 5
Bit 4
Bit 3
Bit 2
Bit 1
Bit 0

Über die Pins 8-13 a sprechen wir später.)

Die **DDRD** registrieren ersetzt die **PinModus()** Befehl.
Das Setzen eines Bits auf 1 setzt den Pin als Ausgang.
Das Setzen eines Bits auf 0 setzt den Pin als Eingang.

Zum Beispiel:

```
DDRD = 0b11111111;
```

Diese Codezeile setzt die Pins 0-7 als Ausgänge.

Hinweis: Das „0b“ teilt dem Arduino mit, dass diese Zahl binär ist.

Die **PORTD** registrieren ersetzt die **digitalWrite()** Befehl.
Das Setzen eines Bits auf 1 setzt den Pin HIGH.
Das Setzen eines Bits auf 0 setzt den Pin auf LOW.

Zum Beispiel:

```
PORTD = 0b10000000;
```

Diese Codezeile setzt Pin 7 auf High und alle anderen Pins auf Low.

Sie können jetzt acht Codezeilen in eine einzige Zeile verkleinern!

```
PinModus(0, AUSGANG);
PinModus(1, AUSGANG);
PinModus(2, AUSGANG);
PinModus(3, AUSGANG);
PinModus(4, AUSGANG);
PinModus(5, AUSGANG);
PinModus(6, AUSGANG);
PinModus(7, AUSGANG);
```

```
DDRD = 0b11111111;
```

```
digitalWrite(0, HOCH);
digitalWrite(1, NIEDRIG);
digitalWrite(2, NIEDRIG);
digitalWrite(3, NIEDRIG);
digitalWrite(4, NIEDRIG);
digitalWrite(5, NIEDRIG);
digitalWrite(6, NIEDRIG);
digitalWrite(7, NIEDRIG);
```

```
PORTD = 0b10000000;
```

Aber was ist mit den Arduino-Pins 8-13?

?

Sie werden von der **DDRB** und **PORTB** registriert.

DDRB: Pin-Modus einstellen (Eingang oder Ausgang)

PORTB: Setzen Sie den Pin hoch oder niedrig, wenn es sich um einen Ausgang handelt

Aber Vorsicht – die Bitnummern in den Registern stimmen nicht mit den Arduino-Pin-Nummern überein! Dies kann verwirrend sein, daher ist es einfacher, mit den Pins 0-7 zu arbeiten, wenn Sie zum ersten Mal die Verwendung von Registern erlernen.

Bit 5
Bit 4
Bit 3
Bit 2
Bit 1
Bit 0

Zum Beispiel würden diese beiden Codezeilen alle Pins als Ausgänge setzen und dann Pin 13 (Bit 5) auf High setzen:

```
DDRB = 0b00111111;
PORTB = 0b00100000;
```