

LEDs mit dem Schieberegister 74HC595 steuern



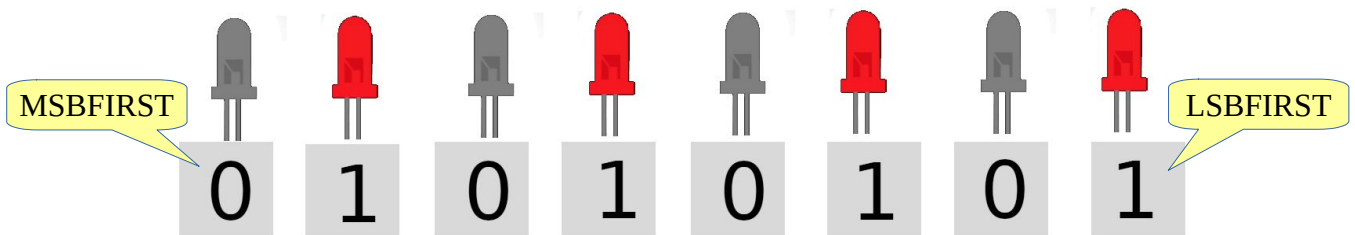
Das Schieberegister 74HC595 verfügt über acht Ausgänge, die mit nur drei Datenleitungen angesteuert werden. Für jeden Ausgang wird ein Signal in Form einer „0“ = LOW, oder „1“ = HIGH gesendet. Insgesamt sind acht Bits notwendig.

Beispiele:

1111111 → alle LEDs einschalten
 0000000 → alle LEDs ausschalten
 1010101 → jede zweite LED ausschalten
 1000001 → nur die beiden äußeren LEDs einschalten

Der erste Wert wird in den Speicher geschrieben.

Sobald der nächste Wert geschrieben wird, werden alle vorherigen Werte eine Position weiter geschoben.



Das Schieberegister besteht aus zwei Teilen:

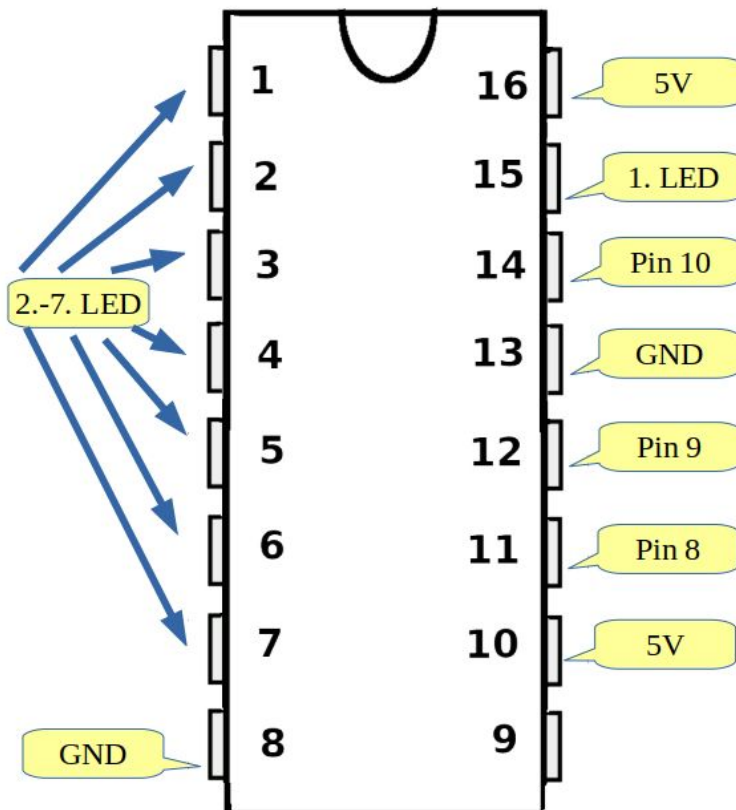
dem Schieberegister:

die einzelnen Bits werden durch den Seriellen Eingang (Pin 14 des Schieberegisters) zuerst hier hinein geschoben, es wird noch kein Signal an die Ausgangspins geschickt, mit MSBFIRST (links beginnen) LSBFIRST (rechts beginnen) kann der Start des „Schiebevorgangs“ bestimmt werden

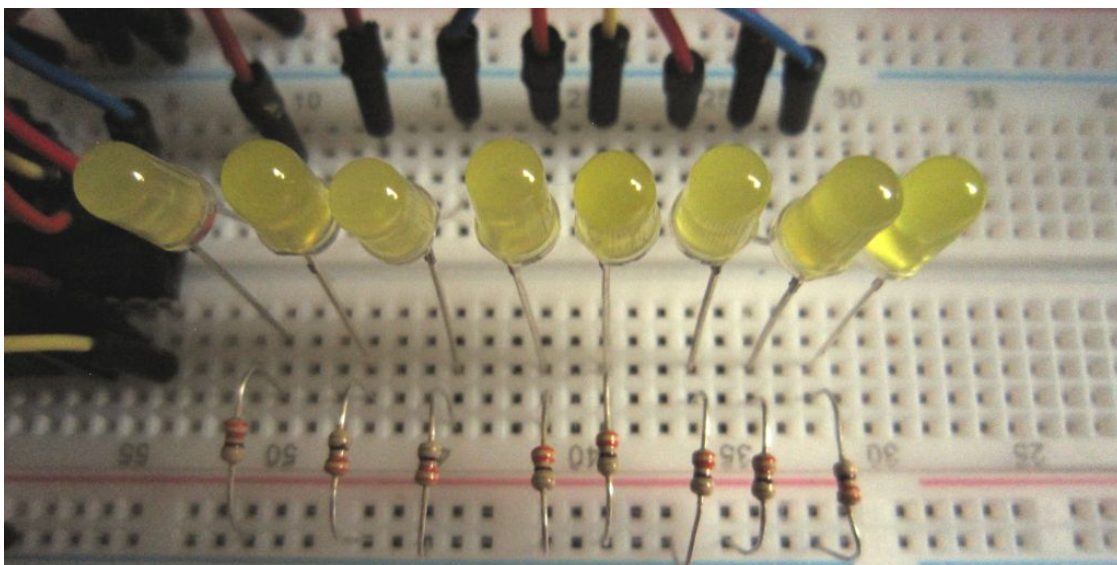
dem Ausgangsregister:

erst wenn alle Bits an ihren Platz geschoben wurden, werden durch das Schreiben von HIGH auf den Pin 12 die Daten in das Ausgangsregister übernommen.

Die Pinbelegung:



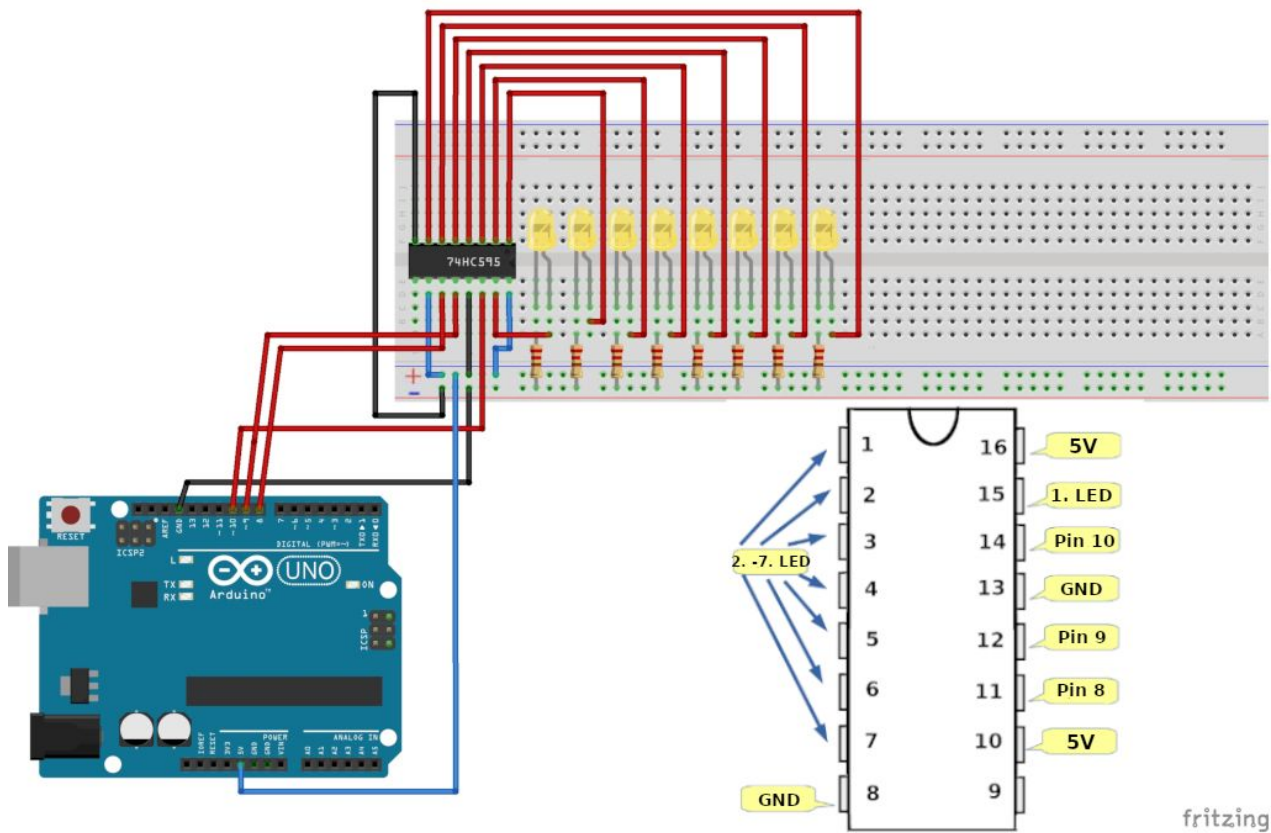
Die Anordnung der LEDs:



Benötigte Bauteile:

- 8 LED
- Schieberegister 74HC595
- 8 Widerstände $> 100 \Omega$
- Leitungsdrähte

Baue die Schaltung auf.



fritzing

Für die Programmierung brauchst du noch Informationen über verschiedene Manipulationen von Bits.

Verschieben von Bits nach rechts :

	1	1	1	1	1	1	1
LED = LED >> 1	0	1	1	1	1	1	1
LED = LED >> 1	0	0	1	1	1	1	1
LED = LED >> 1	0	0	0	1	1	1	1
LED = LED >> 1	0	0	0	0	1	1	1

B markiert eine Binärzahl

LED = B1111111;

LED = LED >> 1;

Alle Bits werden um eine Position nach rechts verschoben. Das letzte Bit auf der rechten Seite fällt heraus, auf der linken Seite wird jeweils eine 0 eingefügt.

Im Programm sieht das so aus:

```
// LSBFIRST -> Bits von rechts in das Register schieben
LED = B11111111;

/*
  8-mal schieben
  bis alle Bits herausgeschoben sind
  und eine 0 vorne hinzugefügt wurde
  Stand am Ende:
  B00000000
*/
for (int i = 0; i <= 7; i++)
{
  // Ausgaberegister ausschalten
  digitalWrite(AusgabePin, 0);

  // Bits in das Schieberegister schieben
  shiftOut(EingangPin, BitsSchieben, LSBFIRST, LED);

  // Ausgaberegister einschalten
  digitalWrite(AusgabePin, 1);
  delay(200);

  // 1 Bit nach rechts schieben
  LED = LED >> 1;
}
```

Verschieben der Bits nach links:

1 1 1 1 1 1 1 1

LED = LED << 1;

1 1 1 1 1 1 1 0

LED = LED << 1

Alle Bits werden um eine Position nach links verschoben.

1 1 1 1 1 1 0 0

LED = LED << 1

Das letzte Bit auf der linken Seite fällt heraus, auf der

1 1 1 1 1 0 0 0

LED = LED << 1

rechten Seite wird jeweils eine 0 eingefügt.

1 1 1 1 0 0 0 0

LED = LED << 1

Im Programm sieht das so aus:

```
LED = B11111111;

for (int i = 0; i <= 7; i++)
{
  // Ausgaberegister ausschalten
  digitalWrite(AusgabePin, 0);

  // Bits von rechts in das Register schieben
  shiftOut(EingangPin, BitsSchieben, LSBFIRST, LED);

  // Ausgaberegister einschalten
  digitalWrite(AusgabePin, 1);
  delay(200);

  // 1 Bit nach links schieben
  LED = LED << 1;
}
```

Wert der Bits umkehren:

1 0 1 0 1 0 1 0

LED = B10101010;

0 1 0 1 0 1 0 1

LED = ~ LED;

aus 0 wird 1, aus 1 wird 0

Im Programm sieht das so aus:

```
LED = B10101010;

for (int i = 0; i <= 6; i++)
{
  digitalWrite(AusgabePin, 0);
  shiftOut(EingangPin, BitsSchieben, MSBFIRST, LED);
  digitalWrite(AusgabePin, 1);
  delay(200);
  LED = ~ LED;
}
```

Definiere die Variablen für das Schieberegister und das Array für das LED-Muster.

```
int BitsSchieben = 8;
int AusgabePin = 9;
int EingangPin = 10;

// Array LEDs leuchten von außen nach innen und zurück
byte AussenNachInnen[] =
{
    B10000001, B01000010, B00100100, B00011000,
    B00100100, B01000010, B10000001
};
```

Im setup-Teil werden die Ausgänge definiert:

```
void setup()
{
    pinMode(BitsSchieben, OUTPUT);
    pinMode(AusgabePin, OUTPUT);
    pinMode(EingangPin, OUTPUT);
}
```



Der loop-Teil. Beachte die Kommentare.

```
void loop()
{
    /*
    Lauflicht vorwärts
    LSBFIRST Bits von rechts ins Register schieben
    dann jeweils 1 Bit nach rechts schieben
    Startpunkt hängt davon ab, an welchen Pins die
    1. Led und die weiteren LEDs gesteckt sind
    */
    byte LED = B10000000;

    // die letzte LED bleibt an
    for (int i = 0; i <= 6; i++)
    {
        // Ausgaberegister ausschalten
        digitalWrite(AusgabePin, 0);

        // Bits von rechts in das Register schieben
        shiftOut(EingangPin, BitsSchieben, LSBFIRST, LED);

        // Ausgaberegister einschalten
        digitalWrite(AusgabePin, 1);
        delay(200);
    }
}
```

```
// 1 Bit nach rechts schieben
LED = LED >> 1;
}

/*
durch die Verschiebung ist der aktuelle Stand:
LED = B00000001
Lauflicht rückwärts -> 1 Bit nach links schieben
*/
for (int i = 0; i <= 7; i++)
{
// Ausgaberegister ausschalten
digitalWrite(AusgabePin, 0);

// Bits von rechts in das Register schieben
shiftOut(EingangPin, BitsSchieben, LSBFIRST, LED);

// Ausgaberegister einschalten
digitalWrite(AusgabePin, 1);
delay(200);

// 1 Bit nach links schieben
LED = LED << 1;
}

// Lauflicht 2 LEDs vorwärts
LED = B11000000;

for (int i = 0; i <= 2; i++)
{
// Ausgaberegister ausschalten
digitalWrite(AusgabePin, 0);

// LSBFIRST -> Bits von rechts in das Register schieben
shiftOut(EingangPin, BitsSchieben, LSBFIRST, LED);

// Ausgaberegister einschalten
digitalWrite(AusgabePin, 1);
delay(200);

// 2 Bits nach rechts schieben
LED = LED >> 2;
}

// Lauflicht 2 LEDs rückwärts
for (int i = 0; i <= 3; i++)
{
// Ausgaberegister ausschalten
digitalWrite(AusgabePin, 0);
```

```
// Bits von links in das Register schieben
shiftOut(EingangPin, BitsSchieben, LSBFIRST, LED);

// Ausgaberegister einschalten
digitalWrite(AusgabePin, 1);
delay(200);

// 2 Bits nach links schieben
LED = LED << 2;
}

// Lauflicht -> LEDs gehen nach und nach aus
// LSBFIRST -> Bits von rechts in das Register schieben
LED = B11111111;

for (int i = 0; i <= 7; i++)
{
    // Ausgaberegister ausschalten
    digitalWrite(AusgabePin, 0);

    // Bits in das Schiebregister schieben
    shiftOut(EingangPin, BitsSchieben, LSBFIRST, LED);

    // Ausgaberegister einschalten
    digitalWrite(AusgabePin, 1);
    delay(200);

    // 1 Bit nach rechts schieben
    LED = LED >> 1;
}

// LSBFIRST -> Bits von rechts in das Register schieben
LED = B11111111;
for (int i = 0; i <= 7; i++)
{
    // Ausgaberegister ausschalten
    digitalWrite(AusgabePin, 0);

    // Bits in das Schiebregister schieben
    shiftOut(EingangPin, BitsSchieben, LSBFIRST, LED);

    // Ausgaberegister einschalten
    digitalWrite(AusgabePin, 1);
    delay(200);

    // 1 Bit nach links schieben
    LED = LED << 1;
}
```



```
// LEDs leuchten von außen nach innen und zurück
// Array AussenNachInnen
for (int i = 0; i <= 6; i++)
{
    digitalWrite(AusgabePin, 0);
    shiftOut(EingangPin, BitsSchieben, MSBFIRST, AussenNachInnen[i]);
    digitalWrite(AusgabePin, 1);
    delay(200);
}

// alle LEDs blinken
for (int i = 0; i <= 6; i++)
{
    digitalWrite(AusgabePin, 0);
    shiftOut(EingangPin, BitsSchieben, MSBFIRST, LED);
    digitalWrite(AusgabePin, 1);
    delay(200);

    // alle Werte umkehren
    LED = ~ LED;
}

// MSBFIRST -> Bits von links in das Register schieben
// jede zweite LED im Wechsel
LED = B10101010;

for (int i = 0; i <= 6; i++)
{
    digitalWrite(AusgabePin, 0);
    shiftOut(EingangPin, BitsSchieben, MSBFIRST, LED);
    digitalWrite(AusgabePin, 1);
    delay(200);

    // alle Werte umkehren
    LED = ~ LED;
}

// jeweils 2 LEDs im Wechsel
LED = B11001100;

for (int i = 0; i <= 6; i++)
{
    digitalWrite(AusgabePin, 0);
    shiftOut(EingangPin, BitsSchieben, MSBFIRST, LED);
    digitalWrite(AusgabePin, 1);
    delay(200);

    // alle Werte umkehren
    LED = ~ LED;
}
```

```

// äußere LEDs/innere LEDs im Wechsel
LED = B11000011;

for (int i = 0; i <= 6; i++)
{
  digitalWrite(AusgabePin, 0);
  shiftOut(EingangPin, BitsSchieben, MSBFIRST, LED);
  digitalWrite(AusgabePin, 1);
  delay(200);

  // alle Werte umkehren
  LED = ~ LED;
}
}

```



Ändere die Richtung der Lauflichter durch Setzen des Startbits.



Erstelle eigene Muster von blinkenden LEDs.