Digitalelektronik 5

Schieberegister

Stefan Rothe

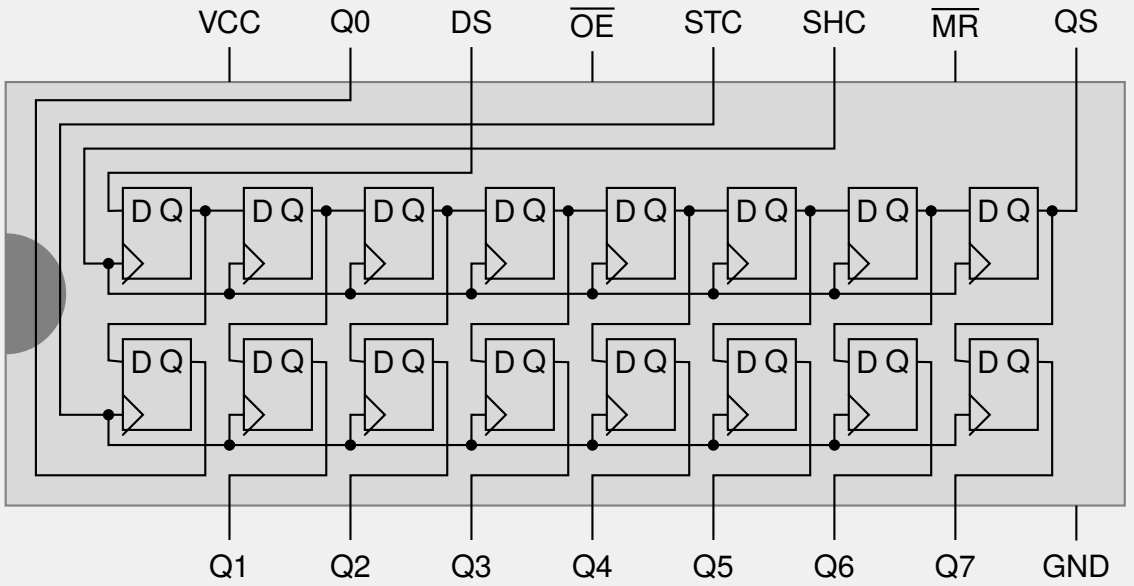
2015–04–21

Rechtliche Hinweise

Quellenangaben

Shift Register 74HC595, Quelle: Fritzing 5

74HC595: Interner Aufbau



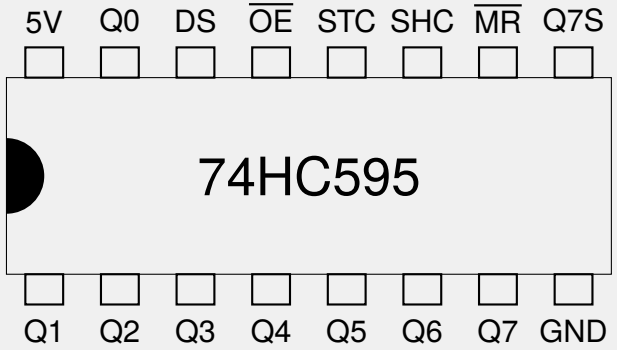
74HC595

- Standard-IC (*integrated circuit*) für Schieberegister
- Serieller Eingang mit Schiebe- und Speichertakt
- 8-bit paraleller Ausgang
- Mehrere Chips können hintereinandergehängt werden.

74HC595: Pinbelegung

Der 74HC595 hat folgende Pinbelegung:

Pin	**Bedeutung**
5V	Betriebspannung 5 V
GND	Masse (*ground*)
DS	Eingabe (*serial data input*)
SHC	Schiebetakt (*shift register clock*)
STC	Speichertakt (*storage register clock*)
$\overline{\text{MR}}$	Löschsignal (*master reset*)
$\overline{\text{OE}}$	Aktivierung der Ausgabe (*output enable*)
Q0 - Q7	Parallele Ausgabe
Q7S	Serielle Ausgabe (*serial output*)



Die überstrichenen Pins sind negativ geschaltet, d.h. ihre Funktion wird bei tiefer Spannung aktiviert.

74HC595: Funktionstabelle

SHC	STC	$\overline{OE}$	$\overline{MR}$	DS	Funktion
↑	-	0	1	0	0 in Register schieben, Ausgabe unverändert
↑	-	0	1	1	1 in Register schieben, Ausgabe unverändert
-	↑	0	1	-	Daten aus Schieberegister in Ausgabe übernehmen
↑	↑	0	1	-	Daten in Ausgabe übernehmen, dann Register schieben
-	-	0	0	-	Schieberegister löschen
-	↑	0	0	-	Schieberegister und Ausgabe löschen
-	-	1	0	-	Schieberegister löschen, Ausgabe abschalten

Arduino: Serielle Ausgabe eines Bytes (LSB)

Serielle Ausgabe eines Bytes, das tiefste Bit (*least significant bit, LSB*) zuerst:

```
void writeByteLSB(int data) {
    int mask = 1;
    for (int i = 0; i < 8; ++i) {
        digitalWrite(CLOCK_PIN, LOW);
        if (data & mask == mask) {
            digitalWrite(DATA_PIN, HIGH);
        }
        else {
            digitalWrite(DATA_PIN, LOW);
        }

        digitalWrite(CLOCK_PIN, HIGH);
        mask = mask << 1;
    }
}
```

Arduino: Serielle Ausgabe eines Bytes (MSB)

Serielle Ausgabe eines Bytes, das höchste Bit (*most significant bit, MSB*) zuerst:

```
void writeByteMSB(int data) {
    int mask = 128;
    for (int i = 0; i < 8; ++i) {
        digitalWrite(CLOCK_PIN, LOW);
        if (data & mask == mask) {
            digitalWrite(DATA_PIN, HIGH);
        }
        else {
            digitalWrite(DATA_PIN, LOW);
        }

        digitalWrite(CLOCK_PIN, HIGH);
        mask = mask >> 1;
    }
}
```

Arduino: Serielle Ausgabe eines Bytes (MSB)

Da das serielle Ausgeben eines Bytes eine sehr häufige Operation ist, definiert die Arduino-Bibliothek eine Funktion, die das übernimmt:

```
// Ausgabe eines Bytes mit dem tiefsten Bit zuerst
shiftOut(DATA_PIN, CLOCK_PIN, LSBFIRST, data);

// Ausgabe eines Bytes mit dem höchsten Bit zuerst
shiftOut(DATA_PIN, CLOCK_PIN, MSBFIRST, data);
```

Aufgabe 6: Schieberegister

a) Bauen Sie eine Schaltung, die mit einem Schieberegister acht Leuchtdioden ansteuert.

- 5V und $\overline{\text{MR}}$ auf die Betriebsspannung
- GND und $\overline{\text{OE}}$ auf die Masse
- Q0 bis Q7 auf die Anoden (Eingänge) der Leuchtdioden
- DS auf den Pin **D1** des Arduino
- SHC auf den Pin **D2** des Arduino
- STC auf den Pin **D3** des Arduino

Vergessen Sie den Vorwiderstand für die Leuchtdioden nicht!

b) Erzeugen Sie mit Hilfe von Bit-Operationen verschiedene Animationen mit den Leuchtdioden.

Aufgabe 7: Auf 100 zählen

a) Bauen Sie eine Schaltung, die mit zwei in Serie geschalteten Schieberegistern zwei 7-Segment-Anzeigen ansteuert.

Um die Schieberegister in Serie zu schalten, müssen sie den **Q7S**-Pin des ersten mit dem **DS**-Pin des zweiten Chips verbinden.

b) Schreiben Sie ein Programm, das die 7-Segment-Anzeige von 0 auf 99 zählen lässt.