Arduino Einführung Teil 10

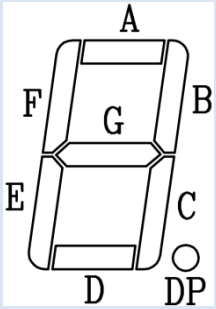
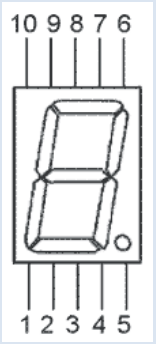
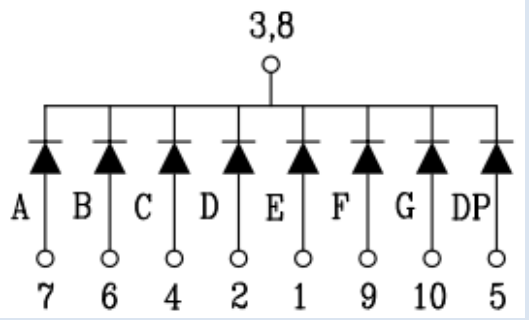
7-Segment LED mit Schieberegister

Wie gehe ich am besten vor?

1. Was will ich machen?
2. Bauteile
3. Aufbau 7-Segment Anzeige I und II
4. Aufbau Schieberegister I, II und III
5. Überlegungen zum Schieberegister SN74HC595N
6. Schaltplan 7-Segment Anzeige
7. Vorversuch „5161AS“
8. Vorversuch „74HC595“
9. Vorversuch Schieberegister „74HC595“ I und II
10. Sketch Versuch 1 Countdown
11. Sketch Versuch 2 Schlange

Was will ich machen?	
Vorversuch 7-Segment Anzeige „5161AS“	Zuordnung der 7-Segment-Kontakte zu den Segmenten A bis G und DP.
Vorversuch Schieberegister „74HC595“	Muster programmieren.
Versuch 1: Countdown	Arduino-Sketch mit 1fachem Array shiftOut()-Funktion Die Ziffern 0 bis 9 sollen im Sekundenrythmus herunter gezählt werden.
Versuch 2: Schlange	Arduino-Sketch mit 1fachem Array shiftOut()-Funktion Die Segmente werden durchlaufen

Bauteile?	
Vorversuch 7-Segment Anzeige	ALLNET-Bausatz: Steckbrett 7-Segment Anzeige „5161AS“ 330 Ω Widerstände Spannungsquelle < 5V
Vorversuch Schieberegister	ALLNET-Bausatz: Steckbrett 7-Segment Anzeige „5161AS“ 330 Ω Widerstände Schieberegister SN74HC595
Versuche 1 und 2	Bauteile s.o.

Aufbau 7-Segment Anzeige I	
Datenblatt	TAIWAN OASIS LED DATA SHEET: TOS-5161AS
PINs	LED-PINS: +5V; gemeinsame Kathode/GND
Segmente	 PIN-Belegung  Zuordnung Segmente und PINs 
Vorwiderstand	Bei 5V und 12 mA ca. 330 Ω
Funktion	Über die PINs 1, 2, 4, 5, 6, 7, 9, 10 werden die Segmente angesteuert. An den PINs 3 oder 8 liegt GND.

Aufbau 7-Segment Anzeige II

Wahrheitstabelle Ziffern (ohne DP)

Zeichen	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

Auswahl von Kombinationen (möglich sind 128)

Hello

Open

Close

Play

Load

No Disc

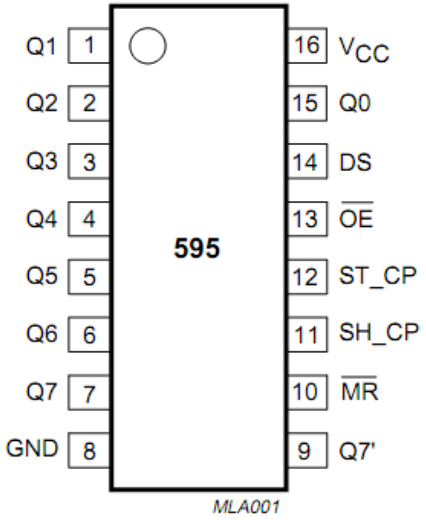
Un Disc

-OFF-

Aufbau Schieberegister I

Datenblatt	NXP Semiconductors DATA SHEET: 74HC595	
Anwendung	Wandelt serielle Daten in parallele Daten um:	8-Bit serielle Eingabe in 8-Bit parallele Ausgabe

Pin-Belegung

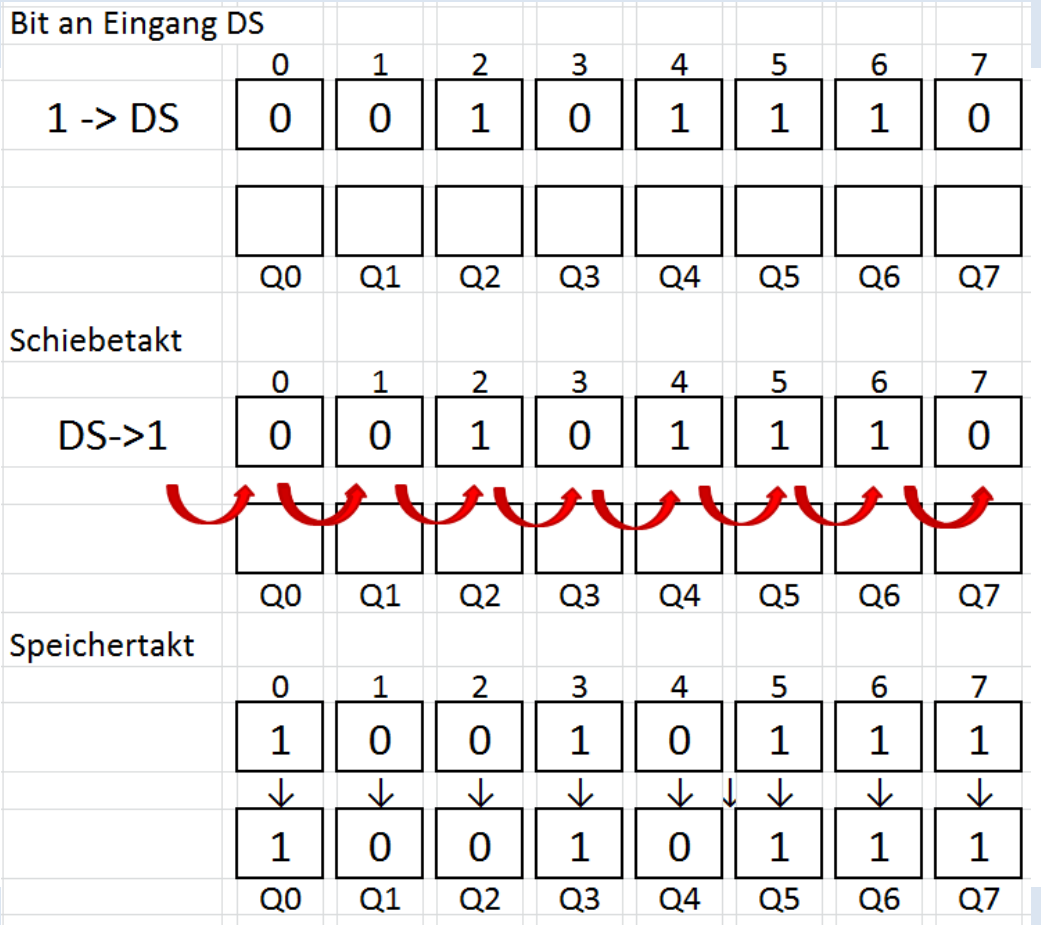


Symbol	Pin	Beschreibung
Q1, Q2, Q3, Q4, Q5, Q6, Q7	1, 2, 3, 4, 5, 6, 7	Parallele Ausgänge Q1 bis Q7
GND	8	Masse (0 V)
Q7‘	9	Serieller Datenausgang
MR	10	Master Reset
SH_CP	11	Schiebetakt (shift clock)
ST_CP	12	Speichertakt (store clock)
OE	13	Ausgangssteuerung
DS	14	Serieller Dateneingang
Q0	15	Paralleler Ausgang Q0
VCC	16	Betriebsspannung

Aufbau Schieberegister II			
Beschaltung			
Symbol	Pin	Beschreibung	Beschaltung
Q1, Q2, Q3, Q4, Q5, Q6, Q7	1, 2, 3, 4, 5, 6, 7	Parallele Ausgänge Q1 bis Q7	-> Vorwiderstand -> Segment
GND	8	Masse	Arduino GND
Q7‘	9	Serieller Datenausgang	unbenutzt
MR	10	Master Reset	auf HIGH (Arduino +5V)
SH_CP	11	Schiebetakt (shift clock)	Arduino PIN 11
ST_CP	12	Speichertakt (store clock)	Arduino PIN 10
OE	13	Ausgangssteuerung	auf LOW (Arduino GND)
DS	14	Serieller Dateneingang	Arduino PIN 12
Q0	15	Paralleler Ausgang Q0	-> Vorwiderstand -> Segment
VCC	16	Betriebsspannung	Arduino +5V

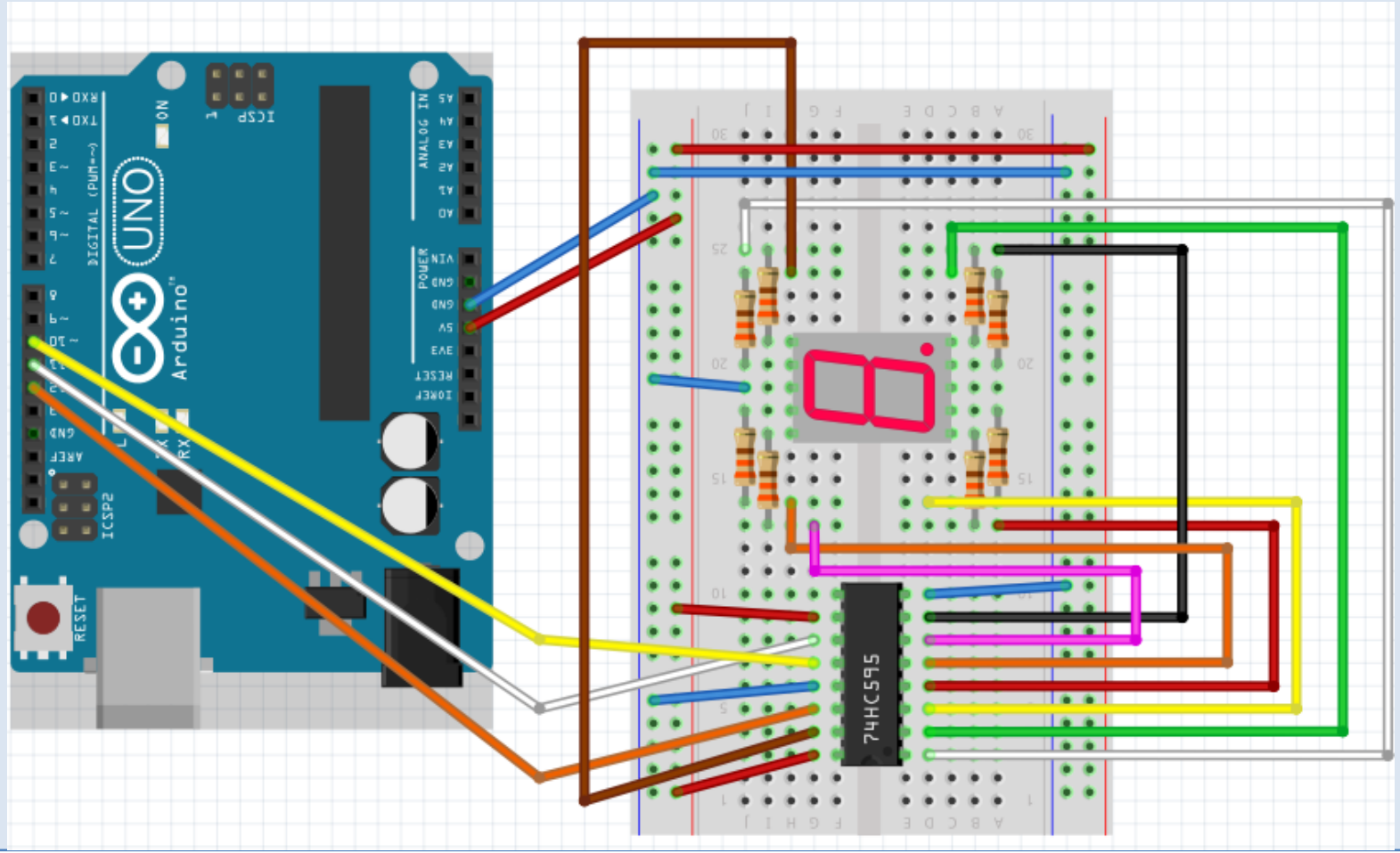
Aufbau Schieberegister III

Funktion Schieberegister



Überlegungen zum Schieberegister SN74HC595N	
Ziel	Anstelle von 8 Arduino-Pins werden nur 3 Arduino-Pins benötigt. Mit einem Schieberegister lassen sich 8 Ausgänge steuern.
Algorithmus für Schieberegister	1. Schiebetaktflanke und Speichertaktflanke können auf HIGH oder LOW stehen 2. Auf den Dateneingang (DS) den gewünschten Wert legen: HIGH oder LOW 3. Einen Schiebetakt ausführen (SH_CP), dazu den PIN zuerst auf LOW dann auf HIGH setzen. 4. Sind alle 8 Bit im Schieberegister dann einen Speichertakt ausführen (ST_CP), dazu den PIN zuerst auf LOW dann auf HIGH setzen.
	Der Wert im Speicherregister bleibt bis zum nächsten Speichertakt erhalten.
	Der Wert im Schieberegister bleibt bis zum nächsten Schiebetakt erhalten.
Programmierung	Da die 8 Ausgänge gleichzeitig gesetzt werden, müssen die gewünschten Zustände immer in einem Byte vorliegen.
	Um das Schieberegister mit 8 Bit zu füllen, bietet die Arduino-IDE die Funktion:
	„shiftOut(dataPin, clockPin, bitOrder, value) „ an.

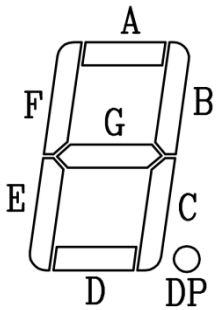
Schaltplan 7-Segment Anzeige

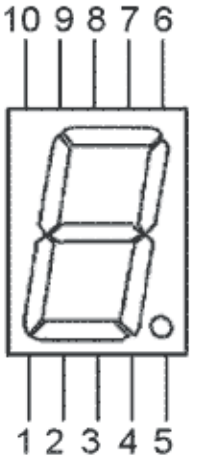


Vorversuch „5161AS“

Aufgabe

Verbinde Kontakt 3 oder 8 über einen geeigneten Widerstand mit Masse.
Verbinde nacheinander die Kontakte 1, 2, 4, 5, 6, 7, 9, 10 mit ca. 5 V.
Notiere unten die Zuordnungen.





A	B	C	D	E	F	G	DP

1	2	4	5	6	7	9	10

7-Segment an Schieberegister

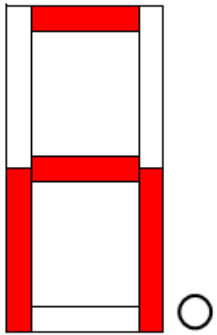
7-Segment	E	D	C	DP	B	A	F	G
7-Segment	1	2	4	5	6	7	9	10
74HC595	Q4	Q3	Q2	Q7	Q1	Q0	Q5	Q6

Vorversuch „74HC595“

Beschreibung
ALL_7Segment_0.ino

74HC595	PIN		Arduino	PIN
DS	14	Dateneingang	dataPin	12
SH_CP	11	Schiebetakt	clockPin	11
ST_CP	12	Speichertakt	latchPin	10

Muster anzeigen



Werte des Musters bzw. Schieberegisters

A	B	C	D	E	F	G	DP
1	0	1	0	1	0	1	0
Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7

Vorversuch Schieberegister „74HC595“ I

```
// ALL_7Segment_0.ino
// functions.ino (zweiter TAB)
// PIN verbunden mit DS, Eingang am  Schieberegister
int dataPin = 12;
// PIN verbunden mit SH_CP (SHIFT CLOCK)
int clockPin = 11;
// PIN verbunden mit ST_CP (STORE CLOCK)
int latchPin = 10;

void setup() {
  //Pins als OUTPUT
  pinMode(latchPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(dataPin, OUTPUT);

  // Es soll nachfolgendes Muster angezeigt werden:
  //              A  B  C  D  E  F  G  DP
  // Muster { 1, 0, 1, 0, 1, 0, 1, 0 }
  // LSBBIT des Musters zuerst an dataPin, also das ganz rechts stehende Bit
  digitalWrite(12, 0);       // Segment DP
  // Schiebtakt: Schieberegister um eine Stelle verschieben
  // Funktion schiebetakt() siehe nachfolgende Folie
  schiebetakt();
  digitalWrite(12, 1);       // Segment G
  schiebetakt();
  digitalWrite(12, 0);       // Segment F
  schiebetakt();
  digitalWrite(12, 1);       // Segment E
  schiebetakt();
  digitalWrite(12, 0);       // Segment D
  schiebetakt();
  digitalWrite(12, 1);       // Segment C
  schiebetakt();
  digitalWrite(12, 0);       // Segment B
  schiebetakt();
  digitalWrite(12, 1);       // Segment A
  schiebetakt();

  // Speichertakt: Schieberegister nach Speicherregister kopieren
  // die Ausgänge Q0 bis Q7 damit setzen, das Muster leuchtet auf
  // Funktion speichertakt() siehe nachfolgende Folie
  speichertakt();
}

void loop() {
}
```

Vorversuch Schieberegister „74HC595“ II

```
// „functions.ino“ in einem zweiten TAB der Arduino-IDE

void schiebetakt(){
  // Schieberegister (SH_CP-Takt) um eine Stelle weiterschieben
  digitalWrite(clockPin, LOW);
  digitalWrite(clockPin, HIGH);
}

void speichertakt(){
  // Schieberegister nach Speicherregister (ST_CP-Takt) kopieren
  digitalWrite(latchPin, LOW);
  digitalWrite(latchPin, HIGH);
}
```

Sketch Versuch 1 Countdown

```
// ALL_7Segment_1.ino und functions.ino (zweiter TAB)
// Countdown
int dataPin = 12;
int clockPin = 11;
int latchPin = 10;

// 1-dimensionales Array für das definieren der Ziffern 0 bis 9; 10 Ziffern mit je 8 Segmenten
byte dataSegment[10] = { B11111100, B01100000, B11011010, B11110010, B01100110,          // 0, 1, 2, 3, 4
                         B10110110, B10111110, B11100000, B11111110, B11110110 };        // 5, 6, 7, 8, 9
void setup() {
  //Pins als OUTPUT
  pinMode(latchPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
}

void loop() {
  // Schleife durchläuft die Ziffern 9 bis 0; der Index eines Arrays beginnt mit „0“
  for (int number = 9; number >= 0; number--){

    // LSBFIRST bedeuted, dass das Byte von rechts nach links gelesen wird
    shiftOut(dataPin, clockPin, LSBFIRST, dataSegment[number]);

    // Schieberegister nach Speicherregister
    speichertakt();
    delay(1000);
  }
}
```

Sketch Versuch 2 Schlange

```
// ALL_7Segment_2.ino und functions.ino (zweiter TAB)
// Schlange
int dataPin = 12;
int clockPin = 11;
int latchPin = 10;

//  1-dimensionales Array für:
//   A              B             C             D             E             F             G
//   B10000000, B01000000, B00100000, B00010000, B00001000, B00000100, B00000010

byte dataSegment[] = { B10000000, B10000000 | B01000000, B01000000 | B00000010, B00000010 | B00001000,
                       B00001000 | B00010000, B00010000 | B00100000, B00100000 | B00000010, B00000010 | B00000100, B00000100 };
void setup() {
  pinMode(latchPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
}

void loop() {
  for (int seg = 0; seg <= sizeof(dataSegment) - 1; seg++){

    shiftOut(dataPin, clockPin, LSBFIRST, dataSegment[seg]);
    // Schieberegister nach Speicherregister
    speichertakt();
    delay(1000);
  }
}
```