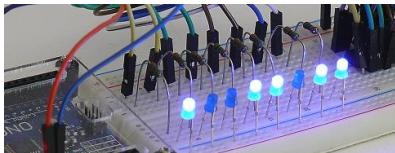


Einführung des Shift-Registers

1. Die seriell-parallel-Umsetzung ist eine ganz zentrale Aktion für die Programmierung. Nur so ist es möglich z.B. mit nur 3 Datenpins des Arduino beliebig viele Ziffern, später auch Grafiken, in ein Display zu transportieren. Öffne dazu die separate PDF: „74HC595 8 Bit Schieberegister mit Latch“.



a) Der **shiftregister-sketch-1** befindet sich auf dem Stick. Teste ihn und nimm Veränderungen dran vor.

b) Erzeuge damit ein schnelles „Lauflicht“. Hier sollten nur wenige zusammenhängende Bits aktiv sein, z.B. **0b10000000** (das ist eine 8-stellige Binärzahl).

c) Vereinfache den Sketch, so dass ein im Array vorgegebenes Bitmuster nach dem Hochladen des Sketches sofort von den LEDs angezeigt wird. Dazu muss man das Register nicht löschen und die Übergabe an das Latch muss nur noch erfolgen nachdemüberlege selbst!

2. Spielen mit dem Shift-Register

Jetzt wird's komfortabel: **shiftOut()** und andere Bequemlichkeiten.

Lade den **shiftregister-sketch-2** und teste ihn.

a) Verändere ihn so, dass rückwärts gezählt wird.

b) Erweitere den Sketch so, dass im seriellen Monitor zusätzlich der Wert der Variablen **counter** erst als Binärzahl und ca. 5-10 Leerzeichen dahinter als Dezimalzahl angegeben wird. **New Line** nicht vergessen!
Binäre Anzeige: `Serial.print(counter, BIN);`
Dezimale Anzeige: `Serial.print(counter)`

3. Lauflichter und Ping-Pong

Lade vom Stick den **shiftregister-sketch-3**. Hier werden einzelne Bits gesetzt. Wir deuten den Sketch gemeinsam.

a) Erweitere den Sketch so, dass das Bitschieben erst von links nach rechts und danach von rechts nach links verläuft. Verändere die Laufgeschwindigkeit, der Effekt soll möglichst „wirkungsvoll“ sein. Mit **bitSet()** und **shiftOut()** ist das relativ einfach realisierbar.

b) **Zusatz:** Erweitere nun den Sketch so, dass synchron mit dem Lauflicht kurze Töne mit an- bzw. absteigender Frequenz, je nach Laufrichtung, zu hören sind. Das klingt ganz lustig!

Integrierte Zifferanzeige

4. Die erste größere Library (TM 1637)

Nun verwenden wir das Anzeige-Modul mit vier Stück 7-Segment Anzeigen und verlassen uns auf die Library.

Übersicht der wichtigsten Anweisungen:

<code>display.begin();</code>	Initialisierung
<code>display.setBacklight(100);</code>	Einstellen der Helligkeit
<code>display.print(„HALLO“);</code>	Text schreiben (naja...)
<code>display.clear();</code>	löscht den Inhalt
<code>display.blink();</code>	erzeugt Blinkereffekt

a) Das Display soll den Inhalt eines Zählers von Null bis Hundert anzeigen. Öffne das Programm „TM1637-einfache-Anwendung“, ändere Einstellungen und teste.

b) Das Display soll dein Geburtsdatum (MM,JJ) anzeigen: Nacheinander 5 Sekunden hell, dann 5 Sekunden blinkend und schließlich 5 Sekunden dunkler. Danach soll das Datum (oder „HALLO“) erscheinen.

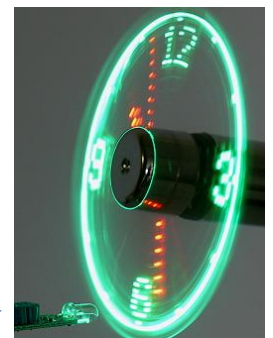


c) Der Sketch „Snake“ soll mindestens 20 Runden alleine ablaufen, ohne Vorspann.

d) Nur eventuell: Finde den Sketch „Display a clock“, in der Library, stelle die Uhr so ein, dass sie als Stoppuhr ab 00:00 läuft, ersetze „true“ durch „false“ in der Display-Anweisung im Loop und beobachte, was sich ändert.... einfach damit etwas spielen aber maximal 5 Minuten!

5. Rotierende Lichtquellen

Wir verwenden die **USB Fan Clock** als rotierende Lichtquelle und positionieren das Licht-Sensor-Modul auf einem Stativ vor dem Rand der Flügel. Der digitale Sensorausgang wird z.B. mit PIN2 des Arduino verbunden. Mit Hilfe des Seriellen Plotters lässt sich nun die Schwelle der Lichtempfindlichkeit so optimieren



(am Poti drehen), dass fast eine Rechteckkurve zu sehen ist. Wenn der Sketch, den wir besprechen, richtig läuft, kannst du die Drehfrequenz (Anzahl der Drehungen pro Sekunde) anzeigen lassen. Verwende dazu das Anzeige-Modul aus Aufgabe 4.

Mit Hilfe eines Stroboskops können wir das Ergebnis überprüfen.