

Teil 3

Lektion

9

Erstelle Zeitgeber

Zusammenfassung:

In diesem Kurs lernen Sie, wie Sie einen Timer für alle Arten von Anlässen erstellen. Während des Lernprozesses lernen Sie die Unterbrechung der MCU (Micro Computer Unit), das schütteln von Schlüsseln, das dynamische Scannen von digitalen Röhren und andere verwandte Kenntnisse kennen.

benötigte Komponenten:

- (1) x Elegoo Uno R3
- (1) x 830 Steckbrett
- (1) X aktiver Summer
- (1) x NPN transistor 8050
- (3) x key
- (1) x LED lamp
- (5) x 220 Widerstand
- (1) x 10k Widerstand
- (1) x 74HC595 IC
- (1) x 4 Digit 7-Segment Display
- (2) x männlich/weiblich Kabel
- (32) x männlich/männlich Kabel



Einführung

In diesem Experiment gibt es drei Schaltflächen:

"Minuten"-Taste:

Jedes Mal, wenn Sie darauf drücken, wird die eingestellte Zeit um eine Minute verlängert. Sobald Sie die Taste drücken und gedrückt halten, wird die Zeit schnell verlängert.

"Sekunden" Taste:

Jedes Mal, wenn Sie darauf drücken, wird die eingestellte Zeit um eine Sekunde verlängert. Sobald Sie die Taste drücken und gedrückt halten, wird die Zeit schnell verlängert.

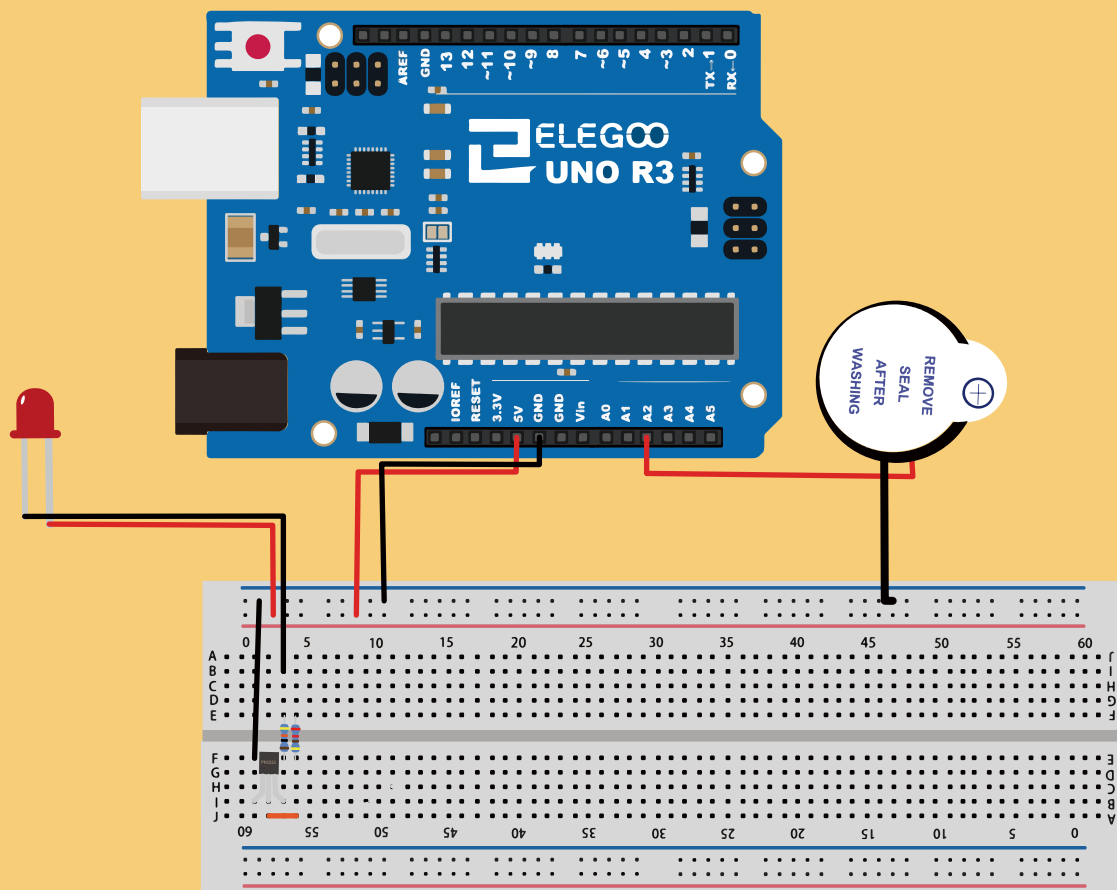
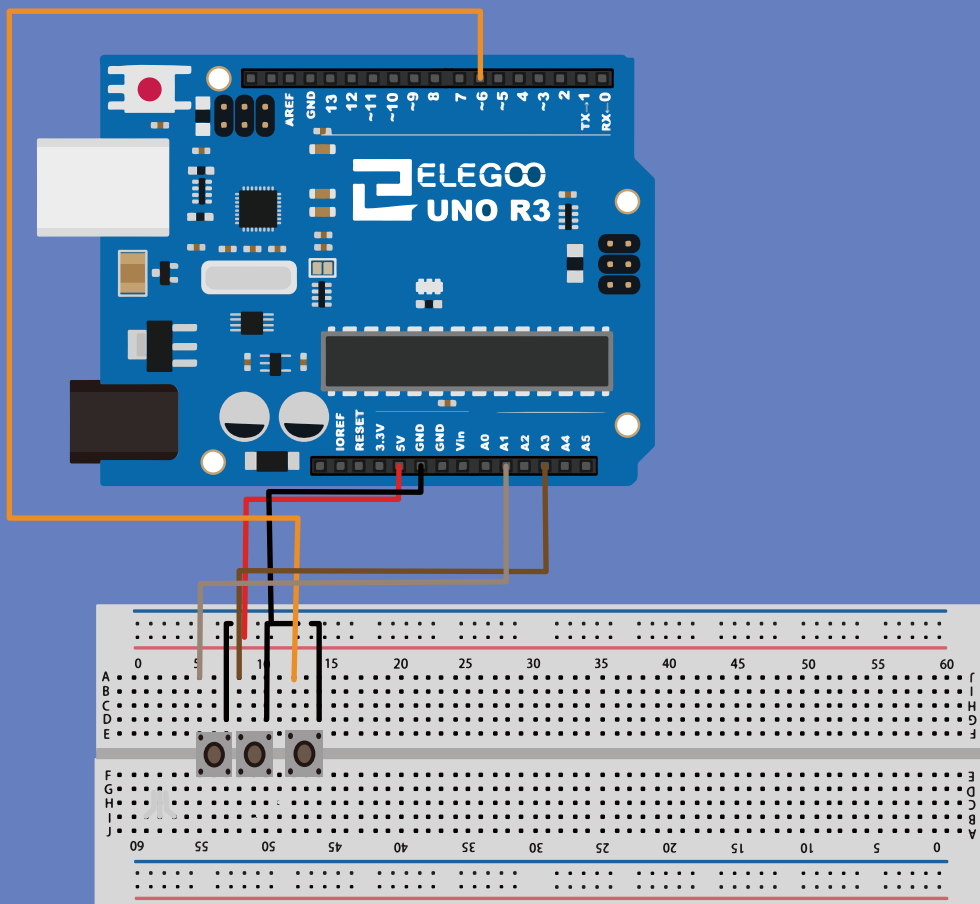
"Funktionstaste":

Nachdem Sie die Zeit eingestellt haben, sollten Sie diese Taste drücken. Wenn Sie diese drücken, wird die 4-stellige 7-Segment-Anzeige abwärts gezählt. Wenn Sie die Taste während des Abwärtszählens erneut drücken, wird die von Ihnen eingestellte Zeit gelöscht.

Die Countdown-Zeit ist eingestellt, der Summer ertönt und die LED-Lichter leuchten.

Wenn Sie zu diesem Zeitpunkt die Funktionstaste erneut drücken, wird der Ton ausgeschaltet, das Leuchten der LED wird gestoppt und die eingestellte Zeit wird ebenfalls gelöscht.





Hund Unterbrechung:

Obwohl in diesem Kurs keine Watchdog-Unterbrechung verwendet wird, müssen wir auch die Watchdog-Unterbrechung verstehen.

Der Watchdog-Timer wird hauptsächlich verwendet, um Ausfälle und Abstürze von Komponenten zu vermeiden, z. B. in einem unendlichen Zyklus. Wenn der Code in eine tote Schleife gerät, löst der Watchdog die Unterbrechung aus und setzt den Mikrocontroller zurück.

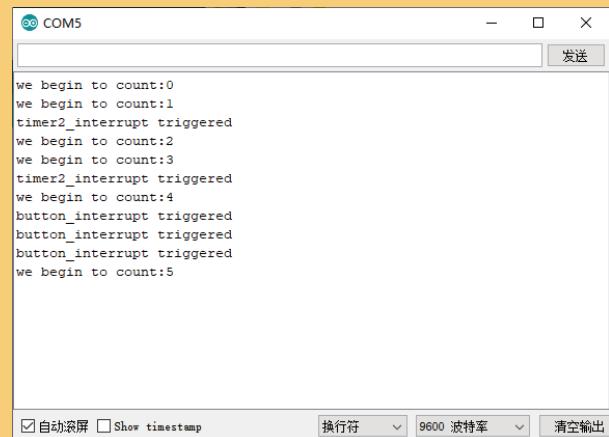
Da wir den Watchdog auf einmal pro Sekunde eingestellt haben, schlug die Auslösung aufgrund der Verzögerung von zwei Sekunden fehl, sodass die Unterbrechung des Watchdogs einmal pro Sekunde zurückgesetzt wurde. Bevor Sie das Programm ausführen, müssen Sie es zurücksetzen.

`Wdt_enable (WDTO_1S);`

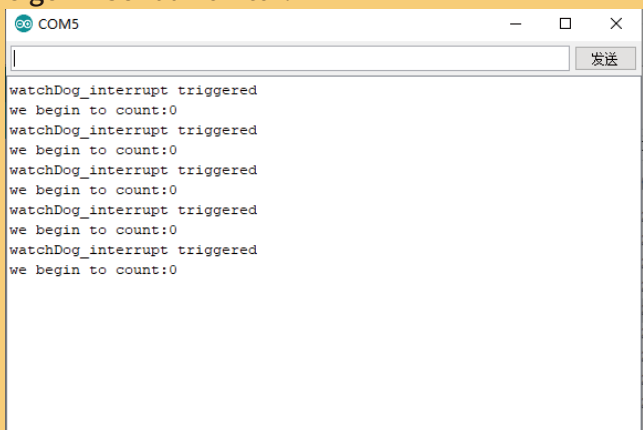
`Serial.println ("watchDog_interrupt triggered");`preceding `"/"`

Nach dem Hochladen des Sketches kommt dieses Anzeige im Serial Monitor:

```
#include "avr/wdt.h"
void setup()
{
  wdt_enable(WDTO_1S);
  Serial.println("watchDog_interrupt triggered");
}
void button_interrupt()
{
  Serial.print("we begin to count:");
  Serial.println(number++);
  delay(2000);
  wdt_reset(); //we should feed the dog avoiding it reseted.
}
```



```
COM5
we begin to count:0
we begin to count:1
timer2_interrupt triggered
we begin to count:2
we begin to count:3
timer2_interrupt triggered
we begin to count:4
button_interrupt triggered
button_interrupt triggered
button_interrupt triggered
we begin to count:5
```



```
COM5
watchDog_interrupt triggered
we begin to count:0
watchDog_interrupt triggered
we begin to count:0
watchDog_interrupt triggered
we begin to count:0
watchDog_interrupt triggered
we begin to count:0
watchDog_interrupt triggered
we begin to count:0
```

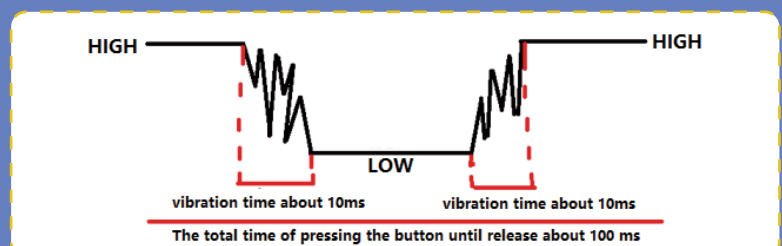
Part 2 : Eliminierung der Tastatur

Mechanischer elastischer Schalter:

Wenn der mechanische Kontakt aufgrund der elastischen Wirkung des mechanischen Kontakts getrennt und angeschlossen wird, wird ein Schlüsselschalter beim Schließen nicht sofort stabil angeschlossen und beim Trennen nicht sofort vollständig getrennt. Stattdessen führt eine Reihe von Jitters, die vom Moment des Schließens und Trennens begleitet werden, dazu, dass die Realität nicht mit dem idealen Effekt übereinstimmt, um dies nicht hervorzurufen. Das Maß für dieses Phänomen besteht darin, den Jitter der Taste zu beseitigen.

Das Jitter des mechanischen Knopfes ist die Antwort auf unser Problem im ersten Teil: Warum nur einmal drücken, zu drei Unterbrechungen führt!

Das mechanische Dithering-Schema sieht wie folgt aus:



Part 1: Unterbrechung

Eine Unterbrechung ist ein unverzichtbarer Bestandteil unseres Designs. Normalerweise wird unser Programm fortlaufend in der Schleife `loop()` ausgeführt. Sobald wir etwas Dringendes erledigen müssen, können wir einen Interrupt (die Unterbrechung) nutzen, um die dringende Angelegenheit abzuschliessen und dann zurückkommen und das ursprüngliche Programm weiter fortzuführen.

Nach der Verkabelung öffnen Sie bitte das Programm im Codeordner-demo1 und klicken Sie auf **UPLOAD**, um das Programm hochzuladen.

Öffnen Sie nach der Verkabelung das Programm im Code-Ordner demo1 und klicken Sie auf **HOCHLADEN**, um das Programm auf den Arduino hochzuladen. Stellen Sie vor dem Ausführen dieses Programms sicher, dass die Bibliothek `<PinChangeInt>` und die Bibliothek `<MTTimer2>` installiert sind oder bei Bedarf neu installiert werden. Andernfalls wird Ihr Code nicht kompiliert. Weitere Informationen zum Laden von Bibliotheksdateien finden Sie in Lektion 5 in Teil 1.

Timer2 Unterbrechung:

Führen Sie den Code aus, den Sie sehen werden:

Dies bedeutet, dass der Code, der nacheinander in `loop()` ausgeführt wird, durch eine Timer-Unterbrechung unterbrochen wird. Der Inhalt der Unterbrechungsfunktion wird zuerst ausgeführt. Erst nachdem die Unterbrechungsfunktion ausgeführt wurde, kehrt der ausgeführte Code zur Funktion `loop()` zurück.

```
#include "MsTimer2.h"
```

```
void setup()
```

```
{
```

```
  MsTimer2::set(4000, timer2_interrupt);
```

```
  MsTimer2::start();
```

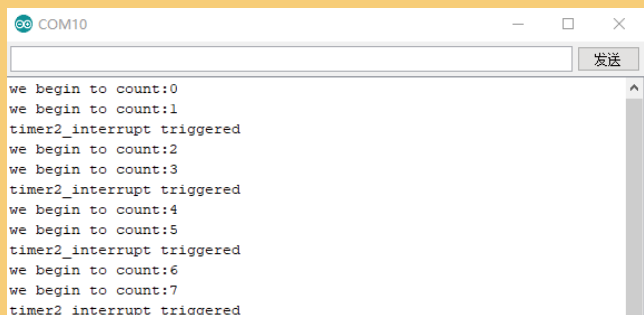
```
}
```

```
void timer2_interrupt()
```

```
{
```

```
  Serial.println("timer2_interrupt triggered");
```

```
}
```



```
COM10
we begin to count:0
we begin to count:1
timer2_interrupt triggered
we begin to count:2
we begin to count:3
timer2_interrupt triggered
we begin to count:4
we begin to count:5
timer2_interrupt triggered
we begin to count:6
we begin to count:7
timer2_interrupt triggered
```

Unterbrechung der Pin-Änderung

Sowohl der PIN-Change-Interrupt, als auch der Timer-Interrupt gehören zum Interrupt, aber der Trigger-Modus wechselt vom Timing-Trigger zum Level-Change-Trigger-Interrupt.

Die Pegeländerungen umfassen: HIGH, LOW, RISING (Umwandlung von niedrigem zu hohem Pegel), FALLING (Konvertierung von hohem zu niedrigem Pegel), CHANGE.

In diesem Fall setzen wir die Unterbrechung "FALLING", wenn die Taste gedrückt wird.

Wenn Sie die Taste drücken, sehen Sie Folgendes:

(Aber warum drücken wir nur einmal einen Knopf und haben drei Unterbrechungen? Bitte überlegen Sie selbst. Wir werden im zweiten Teil antworten.)

Bitte denken Sie sorgfältig nach und wir werden im zweiten Teil antworten.

```
#include "PinChangeInt.h"
```

```
void setup()
```

```
{
```

```
  pinMode(A3, INPUT_PULLUP);
```

```
  attachPinChangeInterrupt(A3, button_interrupt, FALLING );
```

```
}
```

```
void button_interrupt()
```

```
{
```

```
  Serial.println("button_interrupt triggered");
```

```
}
```

- Nach der Verkabelung öffnen Sie bitte das Programm im Codeordner-demo2 und klicken Sie auf UPLOAD, um das Programm hochzuladen.
- Aufgrund von Jitter müssen wir eine kurze Verzögerung hinzufügen, um den Jitter zu beseitigen.
- Die Verwendung einer Verzögerung bei der Unterbrechung führt jedoch zu einer geringeren Gesamteffizienz, was der Maschine entspricht, die im Leerlauf läuft und nichts tut.
- Wir verwenden also Millis, um die Zeit aufzuzeichnen, zu der der Interrupt zweimal ausgelöst wird, wenn die Taste zweimal gedrückt wird. In diesem Fall wird die Sinkflanke verwendet, um den Interrupt auszulösen. Gemäß der Figur beträgt das Intervall zwischen den beiden Abstiegsanten etwa 200 ms.
- Die Aufnahmezeit der zweiten Tastenunterbrechung - Die Aufnahmezeit der letzten Tastenunterbrechung > 200 bestimmt, dass der Tastendruck nicht jittert.

```
if (200 < (abs(time_buttonApin - last_time_buttonApin)))
{
  Serial.println("buttonApin have been pressed");
  last_time_buttonApin = time_buttonApin; //update the time you press
}
```

Part 3 : Dynamisches Scannen von Digitalröhren

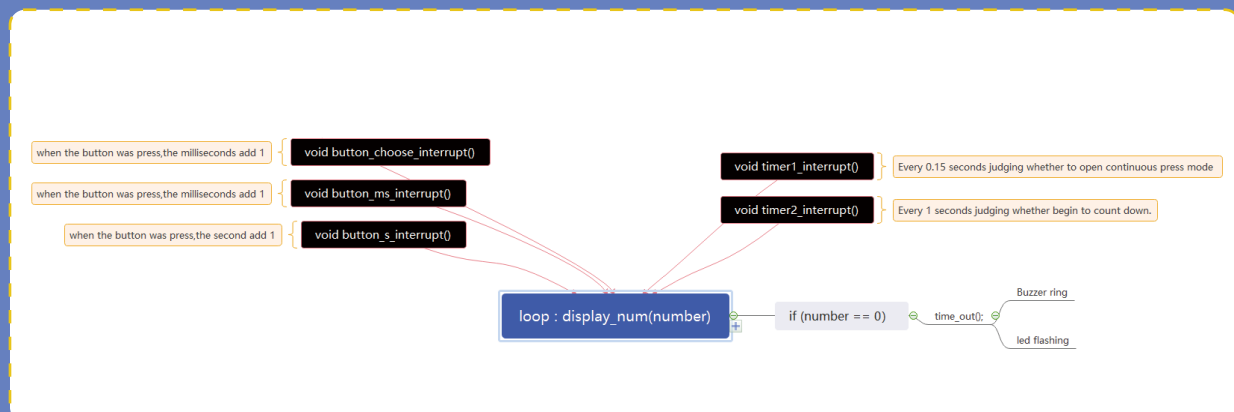
- Dynamische Scan-Anzeige: Senden Sie nacheinander Schriftcodes und die entsprechende Bitauswahl an jede digitale Röhre. Obwohl nur eine Ziffer funktioniert, können Sie nach dem Prinzip der Persistenz des Sehens alle Zahlen anzeigen, da die Scan-Geschwindigkeit jeweils so hoch ist, dass Sie kaum die Intervalle bemerken.

Demo3:

- After wiring, please open the program in the code folder- demo3 and click UPLOAD to upload the program.

Part 4 Mehrfunktionaler Timer erstellen

Gedanken:



- Öffnen Sie nach der Verkabelung das Programm im Codeordner MyTimer und klicken Sie auf HOCHLADEN, um das Programm hochzuladen.