

Blinken auf dem Zeitstrahl.

Eine Lampe ist dauernd an. Eine zweite Lampe blinkt auf dem Zeitstrahl.

Hinweis: Die Lampen sind mit ihrer Plus-Seite an eine Ausgangsbuchse des Arduino angeschlossen und mit ihrer Minus-Seite über einen Strombegrenzungswiderstand von 10kOhm an Masse.

Näheres siehe unter: <http://www.probonocontramalum.de/Laufendes-2016-455.html>

```
const int Lampe714 = 5;    const int Lampe717 = 7;
int Lampenspannung934 = LOW;
unsigned long Alter_Zeitwert977 = 0 ;

const long Leuchtdauer945 = 300;

void setup() {
pinMode(Lampe714, OUTPUT);
pinMode(Lampe717, OUTPUT);  digitalWrite(Lampe717, HIGH);
} void loop() {

unsigned long Neuer_Zeitwert988 = millis();

if (Neuer_Zeitwert988- Alter_Zeitwert977 >= Leuchtdauer945 )
{
  Alter_Zeitwert977 = Neuer_Zeitwert988 ;

  if ( Lampenspannung934 == LOW) { Lampenspannung934 = HIGH; }
  else { Lampenspannung934 = LOW; }

digitalWrite(Lampe714, Lampenspannung934);
}
}
```

Erklärungen.

a)

```
Teil1 void setup() { Teil2 } void loop() { Teil3 }
```

Das ist der Aufbau des Gesamtprogramms.

b)

```
const int Lampe714 = 5; const int Lampe717 = 7;
```

const heißt konstant **int** heißt: eine ganze Zahl **=** ist das Zuordnungszeichen **Das Semikolon** ist das Abschlußzeichen. Dem frei wählbaren Namen "Lampe714" wird die Zahl 5 zugeordnet

c)

```
int Lampenspannung934 = LOW ;
```

Dem freiwählbaren Namen "Lampenspannung934" wird der Wert "LOW" zugeordnet. Dieser kann übrigens später in den Wert "HIGH" umgewandelt werden. Das entspricht "null" und "eins".
int heißt: der Wert ist ganzzahlig

d)

```
unsigned long Alter_Zeitwert977 = 0 ;
```

Dem freiwählbaren Namen "Alter_Zeitwert977" wird der Wert "null" zugeordnet. Dieser kann übrigens später in andere Zahlen umgewandelt werden.
unsigned long heißt: der Wert ist ganzzahlig, er ist eine positive Zahl und er kann sehr groß sein. Das wird deshalb gemacht, weil später Millisekunden gezählt werden und dabei große Zahlen zustandekommen. Der Arduino bekommt dies hier schon vorangekündigt und legt sich entsprechend Speicherplatz dafür zurecht.

d)

```
const long Leuchtdauer945 = 300; ;
```

Dem frei wählbaren Namen "Leuchtdauer945" wird der Wert "300" zugeordnet.
const bedeutet daß der zugeordnete Wert unveränderlich ist.
long bedeutet, daß es eine große Zahl sein kann und das es ganzzahlig ist.

e)

```
void setup() {
```

Hier beginnt der zweite Teil des Gesamtprogramms.

f)

```
pinMode(Lampe717, OUTPUT); digitalWrite(Lampe717, HIGH);
```

links: Diejenige Buchse des Arduino, welche dem frei wählbaren Namen "Lampe717" zugeordnet ist (siehe auch in Teil1 des Gesamtprogramms) wird eine Ausgangsbuchse sein.
rechts: Setze diese Buchse auf HIGH (das sind plus 5Volt). Wenn später kein Gegenbefehl kommt, wird die zugehörige Lampe die ganze Zeit durchgehend leuchten.

g)

```
} void loop() {
```

Hier beginnt der dritte Teil des Gesamtprogramms.

f)

```
unsigned long Neuer_Zeitwert988 = millis() ;
```

millis() bedeutet: Arduino, lies deine eingebaute innere Uhr ab, welche die Zeit in fortlaufenden Millisekunden anzeigt, und merke dir den Zahlenwert.
Die ganze Zeile bedeutet: Lies die innere Uhr ab und ordne den abgelesenen Zahlenwert dem frei wählbaren Namen "**Neuer_Zeitwert988**" zu.

h)

```
if (Neuer_Zeitwert988 - Alter_Zeitwert977 >= Leuchtdauer945) { ... }
```

Diese Zeile bedeutet: Wenn die Prüfaussage **nicht** wahr ist, dann überspringe den in geschweiften Klammern stehenden Text und gehe dann im Programm weiter. Hier heißt das: Du kommst dann an das Programmendezeichen } und von da aus gehst du an deine Arbeit und führst den dritten Programmteil immer und immer wieder aus.

i)

```
{  
Alter_Zeitwert977 = Neuer_Zeitwert988 ;    "... (hier steht noch weiterer Text)..."  
}
```

Das sind die geschweiften Klammern aus Punkt h). Deren Inhalt kommt dann zum Tragen, wenn die Prüfaussage aus Punkt h) **wahr** ist. Der frei wählbare Name bekommt einen neuen Zahlenwert zugeordnet. Und zwar bekommt der Name "Alter_Zeitwert977" ab sofort denjenigen Zahlenwert, welchen derzeit der Name "Neuer_Zeitwert988" zugeordnet hat.

j)

```
{  
    "... (hier steht noch weiterer Text)..."  
    if (Lampenspannung934 == LOW) { Lampenspannung934 = HIGH; }  
    else { Lampenspannung934 = LOW; }  
    "... (hier steht noch weiterer Text)..."  
}
```

Auch das sind die geschweiften Klammern aus Punkt h).

Hier ist die Prüfaussage: "Der dem frei wählbaren Namen zugeordnete Wert einerseits und der Wert "LOW" andererseits sind gleich." Ist diese Aussage **wahr**, dann ordne dem frei wählbaren Namen "**Lampenspannung934**" den Wert "**HIGH**" zu und überspringe dann den "else"-Block und mache im Programm weiter.

Ist aber die Prüfaussage **nicht wahr**, dann überspringe den Inhalt der unmittelbaren folgenden geschweiften Klammern und mache im "else"-Block weiter, führe den aus und gehe dann im aufgeschriebenen Programm weiter.

Zusammengenommen bedeuten diese Zeilen: War der bisherige Zustand "LOW", dann mache daraus "HIGH" und umgekehrt: war der bisherige Zustand **nicht** "LOW" (=also war er "HIGH"), dann setze auf "LOW"

j)

```
{  
    "... (hier steht noch weiterer Text)..."    digitalWrite(Lampe714, Lampenspannung934);  
}
```

Auch das sind die geschweiften Klammern aus Punkt h).

Diese Zeile bedeutet: ordne derjenigen Buchse des Arduino, die mit dem frei wählbaren Namen "**Lampe714**" bezeichnet wird, einen Wert zu. Und zwar genau denjenigen Wert, der in diesem Augenblick dem frei wählbaren Namen "**Lampenspannung934**" zugeordnet ist. Dieser Wert kann nur "HIGH" oder "LOW" sein. Das geht daraus hervor, daß viel weiter vorn diese Buchse als "OUTPUT" beschaltet wurde, und ein solcher kann nur die Werte "HIGH" und "LOW" annehmen. **digitalWrite** bedeutet, daß die Buchse eingeschaltet oder ausgeschaltet wird.

```
}
```