

Quelle: <http://www.probonocontramalum.de/Laufendes-2016-455.html>

Solange der Taster gedrückt ist, blinken drei Lampen. Eine vierte Lampe leuchtet die ganze Zeit über, gleich ob der Taster gedrückt ist oder nicht. Die Lampen sind mit ihrer Plus-Seite an je eine Ausgangsbuchse des Arduino angeschlossen und mit ihrer Minus-Seite über einen Strombegrenzungswiderstand von 10kOhm an Masse. Der Taster ist in betätigtem Zustand leitend. Er geht von Masse an einen Eingang des Arduino. Vom Eingang führt ein Widerstand zu plus 5 Volt. Dieser Widerstand ist bereits im Arduino eingebaut und er wird durch einen Spezialbefehl in Wirkung gesetzt.

```
const int Lampe714 = 4;   const int Lampe715 = 5;   const int Lampe716 = 6;   const int Lampe717 = 7;   const int Taster848 = 8;

int Tasterstand958 = HIGH;
```

Erklärung

Lampe71 und Taster848 und Tasterstand858 sind frei wählbare Namen HIGH ist ein veränderlicher zugeordneter Wert
const=es ändert sich nicht int=es ist eine ganze Zahl = das ist der Zuordnungsbefehl ; =das ist das Endzeichen für den Befehl.
Es handelt sich um die Zuordnung von 6 weitgehend frei wählbaren Namen (blau hinterlegt) zu 6 Zahlen oder Zuständen. Davon sind die ersten fünf während des ganzen Programmablaufs unveränderlich, beim sechsten Namen kann sich der zugeordnete Wert ändern (von HIGH auf LOW= von "1" auf "null" und andersrum).

Der Arduino kennt jetzt die Mitspieler: 4 Lampen, einen Taster und den Tasterstand jeweils mit ihren Rufnamen.

```
void setup() {

pinMode(Lampe714, OUTPUT);   pinMode(Lampe715, OUTPUT);   pinMode(Lampe716, OUTPUT);   pinMode(Lampe717, OUTPUT);

pinMode(Taster848, INPUT_PULLUP);

digitalWrite(Lampe714, LOW);   digitalWrite(Lampe715, LOW);   digitalWrite(Lampe716, LOW);   digitalWrite(Lampe717, HIGH);
```

Erklärung

void setup() { = wird sind jetzt im zweiten Programmteil.
const int Lampe714 = 4; (steht im ersten Teil ganz weit oben) und pinMode(Lampe714, OUTPUT); ergeben zusammengenommen die Aussage: dem Namen "Lampe714" ist die Buchse Nummer 4 des Arduino, die auch auf dem Arduino aufgedruckt ist zugeordnet und diese Buchse wird als Ausgangsbuchse benutzt.
const int Taster848 = 8; (steht im ersten Teil ganz weit oben rechts) und pinMode(Taster848, INPUT_PULLUP); bedeuten zusammengenommen: Die Buchse Nummer 8, die Zahl 8 ist auch auf dem Arduino aufgedruckt, ist eine Eingangsbuchse. Außerdem befindet sich ein wirksamer 20kOhm-Widerstand zwischen dieser Buchse und plus 5 Volt.
digitalWrite(Lampe714, LOW); bedeutet: Die Buchse, welche dem Namen "Lampe714" zugeordnet ist, wird auf "LOW" geschaltet.
digitalWrite(Lampe717, HIGH); bedeutet: Die Buchse, welche dem Namen "Lampe717" zugeordnet ist, wird auf "HIGH" geschaltet.
Hinweis. Lampe717 wird später nicht mehr erwähnt, also brennt sie immer vor sich hin, solange das Programm läuft. Sie ist die Anzeigelampedafür, daß das Programm läuft. Auch dann, wenn die anderen Lampen aus sind, zeigt sie an, das das Programm läuft und daß der Taster zur Betätigung bereit steht.

```

}void loop() { Tasterstand958= digitalRead(Taster848);
if (Tasterstand958 == HIGH) { delay ( 5 ); }
else {
digitalWrite(Lampe714, HIGH);      digitalWrite(Lampe715, HIGH); digitalWrite(Lampe716, HIGH);
delay (1000);
digitalWrite(Lampe714, LOW);      digitalWrite(Lampe715, LOW); digitalWrite(Lampe716, LOW);
delay (1000);
}
}

```

Erklärung

`}void loop() {` ... `}` bedeutet: Anfang und Ende des dritten Programmteiles.

`Tasterstand958= digitalRead(Taster848);` bedeutet: Jetzt wird diejenige Buchse des Arduino, welche dem Namen "Taster848" zugeordnet ist, gelesen und der dort anliegende Spannungszustand wird in der Form "LOW" oder "HIGH" in den Arduino ineingenommen und dann dem Namen "Tasterzustand958" zugeordnet.

`if (Tasterstand958 == HIGH)` bedeutet: Der Arduino fragt: ist der Wert, welcher dem Namen "**Tasterstand458**" in genau diesem Augenblick zugeordnet ist, gleich mit dem Wert "**HIGH**"? Falls diese Aussage wahr ist, dann gehe zum nächsten Block `{ delay ( 5 ); }` und führe diesen aus, überspringe danach den Block "**else { ... }**" und gehe im Programm weiter.

Falls aber die obige Aussage nicht wahr ist, dann überspringe den Block `{ delay ( 5 ); }` und führe den Block "**else { ... }**" aus und gehe danach im Programm weiter.

Solange der Taster **nicht** betätigt ist, ist die Antwort: "Ja", deshalb wird Block 1 wird ausgeführt, das heißt 5 Millisekunden lang warten, dann Block2 überspringen und weitergehen, dann kommt aber schon das Endezeichen und es geht mit `Tasterstand958= digitalRead(Taster848);` von vorn los.

Solange der Taster betätigt ist, ist der dem Namen **Tasterstand 858** bei einer Abfrage der Buchse zugordnete Wert: "**LOW**", die Aussage ist falsch, deshalb wird Block 1 übersprungen und Block zwei ausgeführt.

Dieser sagt aus: Schalte drei Lampen an, warte 1000 Millisekunden, schalte die drei Lampen aus, warte 1000 Millisekunden, gehe danach weiter im Progam. Es kommt dann das Endezeichen, daraufhin geht der Arduino weider zur Zeile **Tasterstand858= digitalRead(Taster848);** usw. usf.

Weitere Erklärungen.

```
if (Prüfaussage) {Block1} else {Block2}
```

Falls die Prüfaussage wahr ist, dann führe Block1 aus, überspringe danach Block2 und gehe dann im Programmablauf weiter.

Ist aber die Prüfaussage **nicht** wahr, dann beachte Block1 nicht, führe Block2 aus und gehe dann im Programmablauf weiter. Ein Beispiel:

```
if (Tasterstand958 == HIGH) { delay ( 5 ); } else { ... "langer Inhalt" ... }
```

Zur Wahl der Namen.

Da, wo eine weitgehend freie Wahl möglich ist, wähle ich Namen aus der deutschen Sprache. Damit unterscheiden sich die Namen vom übrigen Programm, welches ja in einer Art Englisch verfaßt ist. Dadurch wird das Programm übersichtlicher

Des weiteren hänge ich an die Grundnamen eine dreistellige Ziffer an. Für Ausgangsbuchsen eine 700er-Zahl und für Eingangsbuchsen eine 800er-Zahl, für Sonstiges eine 900er-Zahl. Das ist rein willkürlich. Aber es verbessert ebenfalls die Übersicht, denn auch dies hebt sich vom übrigen Programm ab und es ist darüberhinaus in sich einheitlich. Ich kann also mehrere Lampen mit demselben Grundnamen, aber unterschiedlicher Kennzahl bezeichnen.

Unsere Schwernisverkraftungsfähigkeit.

Man darf sich durch Störungen und Mißerfolge nicht aus der Ruhe bringen lassen. Wir sind keine Informatiker, sondern Praktiker, und wir arbeiten uns durch ruhiges Überlegen und durch Ausprobieren langsam an eine Lösung heran. Im Vergleich zu einem Mathematiker oder Informatiker ist das nur ein Hunderstel dessen, was die beim Programmieren können. Aber für uns reicht es aus, wir kommen damit zurecht, wir können unsere eigenen Geräte so programmieren, daß sie ihre Arbeit erfolgreich ausführen können. Für den Anfang ist das gut genug, verbessern kann man es später immer noch.

Quelle: <http://www.probonocontramalum.de/Laufendes-2016-455.html>