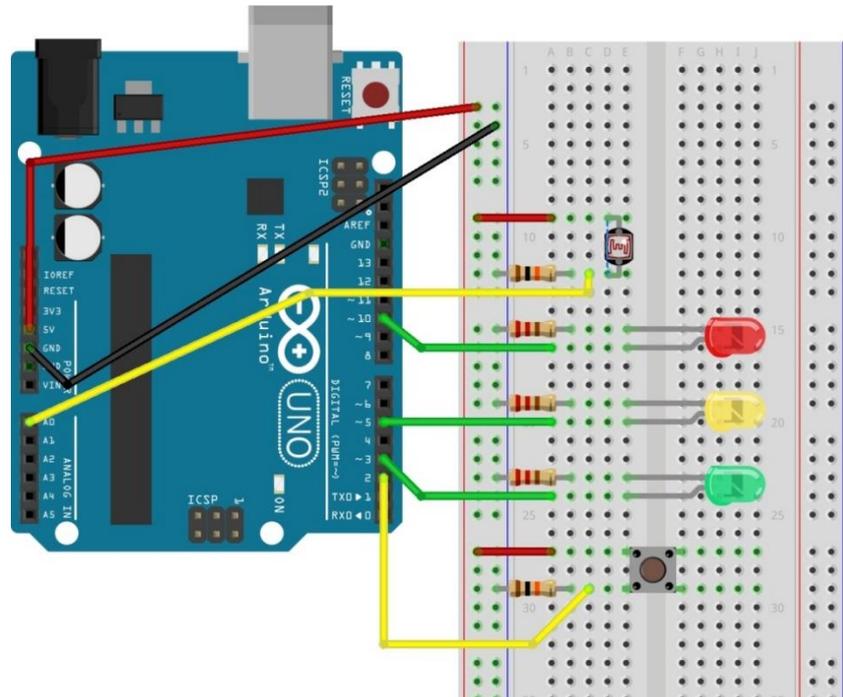


## Arduino – Wiederholungen effizient programmieren

Du verwendest bei diesem Arbeitsblatt weiterhin die Arduino-Steckbrett-Kombination, auf der die bekannte Schaltung zur Programmierung einer Ampelsteuerung oder eines Morseapparats vorbereitet ist:



**Aufgabe 1)** In der Informationsdatei namens [ST\\_I\\_07\\_Arduino\\_Wiederholungen](#) ist erklärt, wie Befehlsblöcke, die mehrmals hintereinander codiert sind, mit Hilfe einer `while`-Schleife übersichtlicher geschrieben werden können. Studiere diese Informationsdatei und verändere sodann die bereits codierten Befehle

```
void greenLed3Blink(),
void yellowLed3Blink(),
void redLed3Blink() und
void led3Blink(int pin)
```

von Aufgabenblatt [ST\\_AA\\_06](#) so, dass das dreimalige Aus- und Einschalten der Leuchtdiode jeweils mit Hilfe einer solchen `while`-Schleife codiert wird. Teste diese »neuen« Befehle.

**Aufgabe 2)** Das nachfolgend abgebildete Codefragment ist so programmiert, dass die Programmschleife fünfmal durchlaufen wird:

Mit Hilfe der Zählvariable `counter` wird dabei von 1 bis 5 mitgezählt.

Auf der nächsten Seite findest du andere „Mitzählvarianten“, damit die Programmschleife fünfmal durchlaufen werden soll. Überlege jeweils, ob dies auch tatsächlich der Fall ist – nutze allenfalls die Modellvorstellung zu Variablen im Zusammenhang mit Zählschleifen!

```
int counter = 1;
...
while(counter <= 5){
    delay(1000);
    digitalWrite(pin,HIGH);
    delay(1000);
    digitalWrite(pin,LOW);
    counter = counter + 1;
}
...
```

```

int counter = 0;
...
while(counter <= 4){
  delay(1000);
  digitalWrite(pin,HIGH);
  delay(1000);
  digitalWrite(pin,LOW);
  counter = counter + 1;
}
...

int counter = 3;
...
while(counter <= 7){
  delay(1000);
  digitalWrite(pin,HIGH);
  delay(1000);
  digitalWrite(pin,LOW);
  counter = counter + 1;
}
...

int counter = 5;
...
while(counter > 0){
  delay(1000);
  digitalWrite(pin,HIGH);
  delay(1000);
  digitalWrite(pin,LOW);
  counter = counter - 1;
}
...

int counter = 0;
...
while(counter < 5){
  delay(1000);
  digitalWrite(pin,HIGH);
  delay(1000);
  digitalWrite(pin,LOW);
  counter = counter + 1;
}
...

int counter = 5;
...
while(counter >= 0){
  delay(1000);
  digitalWrite(pin,HIGH);
  delay(1000);
  digitalWrite(pin,LOW);
  counter = counter - 1;
}
...

int counter = 5;
...
while(counter > 0){
  delay(1000);
  digitalWrite(pin,HIGH);
  delay(1000);
  digitalWrite(pin,LOW);
  counter = counter + 1;
}
...

```

**Aufgabe 3)** Wir Menschen sind es gewohnt, beim Zählen mit der Zahl Eins zu beginnen. Es mag an dieser Gewohnheit liegen, dass in der vorherigen Aufgabe 2) von all' den Möglichkeiten des Mitzählens die zu Beginn angegebene Variante jene ist, die am leichtesten durchschaubar ist:

„...beginne bei Eins und zähle so lange immer wieder Eins dazu, bis du Fünf erreicht hast...“

```

int counter = 1;
...
while(counter <= 5){
  delay(1000);
  digitalWrite(pin,HIGH);
  delay(1000);
  digitalWrite(pin,LOW);
  counter = counter + 1;
}
...

```

Selbst programmierte Befehle, die die Leuchtdiode, die an dem durch die Variable `pin` bezeichneten Steckkontakt angeschlossen ist, vier-, fünf- oder sechsmal aufleuchten lassen, haben bei Verwendung dieser Variante dann folgenden Code:

```

void Led4Blink(int pin) {
  int counter;

  counter = 1;
  digitalWrite(pin, LOW);
  delay(1000);
  while(counter <= 4){
    digitalWrite(pin,HIGH);
    delay(1000);
    digitalWrite(pin,LOW);
    delay(1000);
    counter++;
  }
}

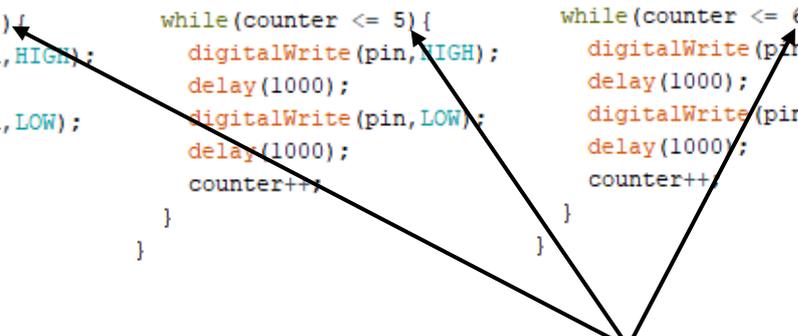
void Led5Blink(int pin) {
  int counter;

  counter = 1;
  digitalWrite(pin, LOW);
  delay(1000);
  while(counter <= 5){
    digitalWrite(pin,HIGH);
    delay(1000);
    digitalWrite(pin,LOW);
    delay(1000);
    counter++;
  }
}

void Led6Blink(int pin) {
  int counter;

  counter = 1;
  digitalWrite(pin, LOW);
  delay(1000);
  while(counter <= 6){
    digitalWrite(pin,HIGH);
    delay(1000);
    digitalWrite(pin,LOW);
    delay(1000);
    counter++;
  }
}

```



Offenbar unterscheidet sich der Befehlscode dieser drei Befehle nur an einer einzigen Stelle – nämlich an jener, an der festgelegt wird, wie weit mitgezählt werden soll!

Vermutlich erinnerst du dich: Ganz ähnlich war es in Aufgabe 3 des vorhergehenden Aufgabenblattes [ST\\_AA\\_06](#) – dort haben wir erkannt, dass wir nicht für jede der drei Leuchtdioden einen eigenen Befehl codieren müssen, wenn wir dem Befehl die Nummer des Steckkontakts, über den die jeweilige Leuchtdiode mit dem Arduino-Board verbunden ist, als „Zusatzinformation“ mitgeben, z.B. so:

```
void Led6Blink(int pin) .
```



Diese „Zusatzinformation“, die in der runden Klammer hinter dem Befehlsnamen angegeben werden muss, wird auch als **Parameter** dieses Befehls bezeichnet und wird wie eine Variable mit einem **Parameternamen** und **vorangestelltem Typbezeichner** (z.B. `int` für eine ganze Zahl) festgelegt. Wenn ein Befehl mehrere Parameter benötigt, müssen diese in der runden Klammer jeweils durch Komma voneinander getrennt werden.

So kann ein Befehl `void ledBlink(int pin, int times)` codiert werden, der die an dem durch `pin` festgelegten Steckkontakt angeschlossene Leuchtdiode so oft blinken lässt, wie durch den **Parameter** namens `times` festgelegt ist.

Codiere diesen Befehl und teste ihn mit der „Arduino-Ampelschaltung“.

**Aufgabe 4)** In Aufgabe 6 vom Aufgabenblatt [ST\\_AA\\_06](#) hast du Befehle zum Morsen einiger Buchstaben codiert. Analysiere den geschriebenen Code und modifiziere ihn an jenen Stellen, wo Befehls-teile zweimal oder öfter wiederholt werden, durch Verwendung von `while`-Schleifen.