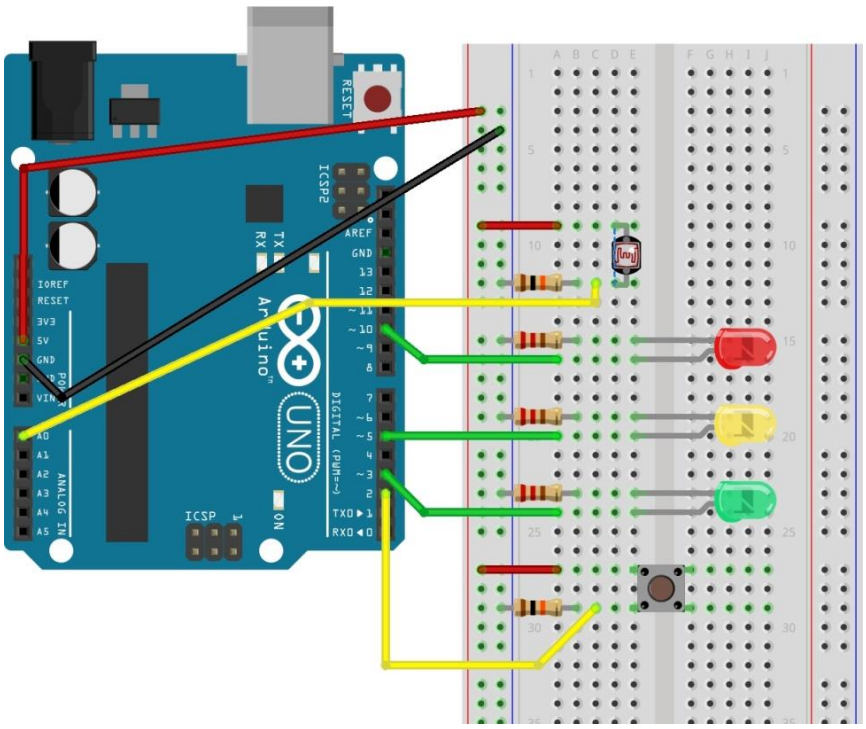
Arbeiten in der Arduino-Programmierumgebung/Lösungen

Abbildung der verwendeten Schaltung:



Aufgabe 1) Hinweis zu häufigen Fehlerquellen beim Abschreiben des Programmcodes:

Beim Codieren in einer Sprache der C-Sprachfamilie sind zu Beginn folgende zwei Fehlertypen häufig („üblich“):

- Vergessen des Strichpunktes (Semikolon) **;** am Ende einer Befehlszeile;
- Fehlen von schließenden, **}** (manchmal auch von öffnenden, **{**) Klammern, sodass der durch die beiden Klammern festgelegte Programmblock im Code nicht definiert ist.

Aufgabe 2)

```
int redLedPin = 10;
int yellowLedPin = 5;
int greenLedPin = 3;

void setup() {
  pinMode(redLedPin,OUTPUT);
  pinMode(yellowLedPin,OUTPUT);
  pinMode(greenLedPin,OUTPUT);
  //set inital LED-states
  digitalWrite(redLedPin,HIGH);
  digitalWrite(yellowLedPin,LOW);
  digitalWrite(greenLedPin,LOW);
}
```

```
void loop() {
  delay(5000);
  digitalWrite(yellowLedPin,HIGH);
  digitalWrite(redLedPin,LOW);
  delay(2000);
  digitalWrite(greenLedPin,HIGH);
  digitalWrite(yellowLedPin,LOW);
  delay(5000);
  digitalWrite(greenLedPin,LOW);
  digitalWrite(yellowLedPin,HIGH);
  delay(2000);
  digitalWrite(yellowLedPin,LOW);
  digitalWrite(redLedPin,HIGH);
}
```

Aufgabe 3)

```
void loop() {
  delay(5000);
  digitalWrite(yellowLedPin,HIGH);
  digitalWrite(redLedPin,LOW);
  delay(2000);
  digitalWrite(greenLedPin,HIGH);
  digitalWrite(yellowLedPin,LOW);
  delay(5000);
  digitalWrite(greenLedPin,LOW);
  delay(1000);
  digitalWrite(greenLedPin,HIGH);
  delay(1000);
  digitalWrite(greenLedPin,LOW);
  delay(1000);
  digitalWrite(greenLedPin,HIGH);
  delay(1000);
  digitalWrite(greenLedPin,LOW);
  delay(1000);
  digitalWrite(greenLedPin,HIGH);
  delay(1000);
  digitalWrite(greenLedPin,LOW);
  digitalWrite(yellowLedPin,HIGH);
  delay(2000);
  digitalWrite(yellowLedPin,LOW);
  digitalWrite(redLedPin,HIGH);
}
```

Aufgabe 4) Lösung mit roter LED als „Morsesignalgeber“:

```
int ledPin = 10;

void setup() {
  pinMode(ledPin, OUTPUT);

}

void loop() {
  // S
  digitalWrite(ledPin,HIGH);
  delay(200);
  digitalWrite(ledPin,LOW);
  delay(200);
  digitalWrite(ledPin,HIGH);
  delay(200);
  digitalWrite(ledPin,LOW);
  delay(200);
  digitalWrite(ledPin,HIGH);
  delay(200);
  digitalWrite(ledPin,LOW);
  delay(200);
  // O
  digitalWrite(ledPin,HIGH);
  delay(600);
  digitalWrite(ledPin,LOW);
  delay(200);
  digitalWrite(ledPin,HIGH);
  delay(600);
  digitalWrite(ledPin,LOW);
  delay(200);
  digitalWrite(ledPin,HIGH);
  delay(600);
  digitalWrite(ledPin,LOW);
  delay(200);
  // S
  digitalWrite(ledPin,HIGH);
  delay(200);
  digitalWrite(ledPin,LOW);
  delay(200);
  digitalWrite(ledPin,HIGH);
  delay(200);
  digitalWrite(ledPin,LOW);
  delay(200);
  digitalWrite(ledPin,HIGH);
  delay(200);
  digitalWrite(ledPin,LOW);
  delay(5000);
}
```

Hinweis: Die Verzögerungen zur Unterscheidung des langen und des kurzen Morsesignals können natürlich von den hier gewählten Werten (0,6 s, 0,2 s) abweichen…

Aufgabe 5

a) Mögliche Fehlinterpretationen des Morsecode dreimal „kurz“, – dreimal „lang“ – dreimal „kurz“ (Auswahl):

„kurz“ – zweimal „kurz“ + „lang“ – zweimal „lang“ – dreimal „kurz“:	EUMS
„kurz“ – „kurz“ – „kurz“ – „lang“ – „lang“ – „lang“ – „kurz“ – „kurz“ – „kurz“:	EEETTTEEE
zweimal „kurz“ – „kurz“ + zweimal „lang“ + „kurz“ – zweimal „kurz“:	UPI

(viele mehr sind möglich)

b) Lösung mit gelber LED als „Zeichenende-Anzeige“ (**`charPin`** als Variablenname für den Pin/Steckkontakt, über den das Ende des Zeichens (engl. **char**acter) angezeigt wird):

```
int ledPin = 10;
int charPin = 5;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(charPin, OUTPUT);

}

void loop() {
  // S
  digitalWrite(ledPin,HIGH);
  delay(200);
  digitalWrite(ledPin,LOW);
  delay(200);
  digitalWrite(ledPin,HIGH);
  delay(200);
  digitalWrite(ledPin,LOW);
  delay(200);
  digitalWrite(ledPin,HIGH);
  delay(200);
  digitalWrite(ledPin,LOW);
  delay(200);
  digitalWrite(charPin,HIGH);
  delay(600);
  digitalWrite(charPin,LOW);
  delay(200);

  // O
  digitalWrite(ledPin,HIGH);
  delay(600);
  digitalWrite(ledPin,LOW);
  delay(200);
  digitalWrite(ledPin,HIGH);
  delay(600);
  digitalWrite(ledPin,LOW);
  delay(200);
  digitalWrite(ledPin,HIGH);
  delay(600);
  digitalWrite(ledPin,LOW);
  delay(200);
  digitalWrite(charPin,HIGH);
  delay(600);
  digitalWrite(charPin,LOW);
  delay(200);

  // S
  digitalWrite(ledPin,HIGH);
  delay(200);
  digitalWrite(ledPin,LOW);
  delay(200);
  digitalWrite(ledPin,HIGH);
  delay(200);
  digitalWrite(ledPin,LOW);
  delay(200);
  digitalWrite(ledPin,HIGH);
  delay(200);
  digitalWrite(ledPin,LOW);
  delay(200);
  digitalWrite(charPin,HIGH);
  delay(600);
  digitalWrite(charPin,LOW);
  delay(5000);
}
```

Aufgabe 6) Erweiterung der Lösung zu Aufgabe 5 mit der gelben LED als „Zeichende-Anzeige“ und der grünen LED als „Wortende-Anzeige“:

```
int ledPin = 10;
int charPin = 5;
int wordPin = 3;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(charPin, OUTPUT);
  pinMode(wordPin, OUTPUT);

}

void loop() {
  // S
  digitalWrite(ledPin,HIGH);
  delay(200);

        ...

  digitalWrite(ledPin,LOW);
  delay(200);
  digitalWrite(charPin,HIGH);
  delay(600);
  digitalWrite(charPin,LOW);
  delay(200);
  // S
  digitalWrite(ledPin,HIGH);
  delay(200);
  digitalWrite(ledPin,LOW);
  delay(200);
  digitalWrite(ledPin,HIGH);
  delay(200);
  digitalWrite(ledPin,LOW);
  delay(200);
  digitalWrite(ledPin,HIGH);
  delay(200);
  digitalWrite(ledPin,LOW);
  delay(200);
  digitalWrite(wordPin,HIGH);
  delay(600);
  digitalWrite(wordPin,LOW);
  delay(5000);
}
```

Aufgabe 7)

a) Vervollständige den nachfolgenden Text, indem Du jeweils den passenden Begriff bzw. die passenden Worte an der dafür vorgesehenen Stelle einsetzt:

Bei der Variablendeklaration wird für **die Variable** ein passender Speicherbereich reserviert. In diesem Speicherbereich ist zunächst **ein unbekanntes Datum** gespeichert. Der Inhalt eines **Speicherbereichs** lässt sich als Folge von **Nullen** und/oder **Einsen** darstellen.

b) Die grüne Leuchtdiode wird in der verwendeten Schaltung über den Steckkontakt mit der Nummer ~3 angesteuert. Dies soll im Programm mit Hilfe einer Variablen namens **greenLedPin** erfolgen. Dazu wird mit dem Befehl `int greenLedPin;` ein passender Speicherbereich reserviert – wir sprechen von einer **Variablendeklaration**.

Nach dieser Variablendeklaration steht im reservierten Speicherbereich (als Bitmuster codiert)

❑ immer die Zahl 0.

❑ immer die Zahl 3.

☒ irgendeine Zahl, man kann nicht sagen, welche.

❑ gar nichts, weil der Speicher noch leer ist.

c) Mit dem Befehl `greenLedPin = 3;` wird in den für die Variable namens **greenLedPin** reservierten Speicherbereich die Zahl 3 geschrieben.

Beim Ausführen des Befehls `digitalWrite(greenLedPin, HIGH);` wird dieser Wert im Speicher gelesen.

Nach dem Lesen dieses Wertes steht in dem für die Variable namens **greenLedPin** reservierten Speicherbereich

❑ noch immer die Zahl 3.

☒ die Zahl 0.

❑ irgendeine Zahl, man kann nicht sagen, welche.

❑ gar nichts, weil der Speicherinhalt gelesen wurde und der Speicher nun leer ist.

Bonusaufgabe

…die Umwandlung der Bitmuster in die entsprechende ganze Zahl erfolgt nach dem beispielhaft angegebenen Schema...