

Morsen mit dem Arduino

Morsecode verwendet drei Symbole, die Punkt (' · '), Strich ('-') und Pause () genannt werden, gesprochen als dit, dah (oder doh) und „Schweigen“. Die Länge eines Dit bestimmt die Geschwindigkeit, mit der gesendet werden kann, und ist die grundlegende Zeiteinheit. Dazu ein Beispiel:

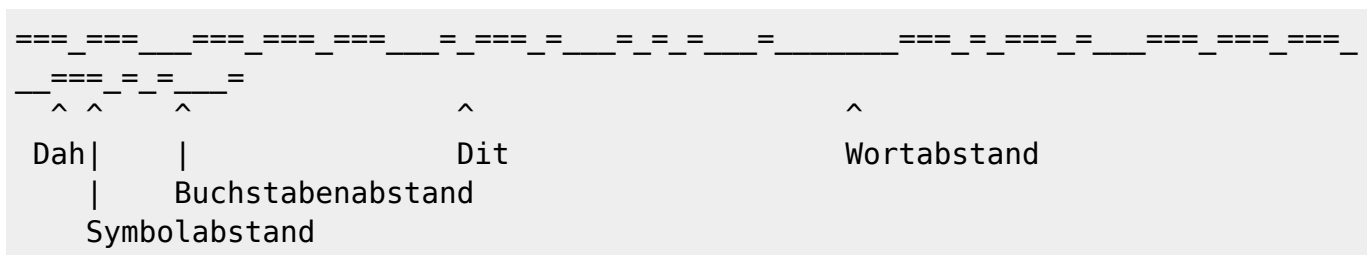
```

--  ---  ···  ···  ·    /    ···  ---  ···  ·
M   O   R   S   E  (space) C   O   D   E
    
```

Genauer gilt Folgendes:

- Ein Dah ist dreimal so lang wie ein Dit.
- Die Pause zwischen zwei gesendeten Symbolen ist ein Dit lang.
- Zwischen Buchstaben in einem Wort wird eine Pause von der Länge eines Dah (oder drei Dits) eingeschoben.
- Die Länge der Pause zwischen Wörtern entspricht sieben Dits.

Für das Beispiel „MORSE CODE“ ergibt sich dann dieses Zeitsignal (das Zeichen = bedeutet „Signal an“, das Zeichen _ bedeutet „Signal aus“):



„Gesprochen“ klingt das dann so: dahdah dahdahdah ditdahdit ditditdit dit, dahditdahdit dahdahdah dahditdit dit.

Quelle: <https://de.wikipedia.org/wiki/Morsezeichen>¹⁾

Allgemeine Form eines Unterprogramms

```

Rückgabetyt      Funktionsname      Parameterliste
int funktions_name (int x, int y) {
    do_something;
}
    
```

- **Rückgabetyt:** Gibt den Datentyp des Werts an, der zurückgegeben wird. (Siehe Beispiel 3) Wird kein Wert zurückgegeben, dann wird hier der Befehl void („leer“) verwendet. (Siehe

Beispiel 1 und 2)

- **Funktionsname:** Frei wählbarer Namen des Unterprogramms bzw. der Funktion.
- **Parameterliste:** Hier stehen die Werte, die an das Unterprogramm übergeben werden. Die Parameterliste kann auch leer bleiben.

Funktionen für "Dah" und "Dit"



(A1)

- Schließe eine LED korrekt an den Pin 8 des Arduino an.
- Übernehme das folgende Programm in die Arduino Umgebung und ergänze es so, dass es sich übersetzen lässt.
- Übertrage es auf den Arduino und führe es aus.
- Verändere das Programm so, dass es eine Botschaft aus mehreren Worten korrekt sendet. ²⁾

[morsecode.ino](#)

```
// LED-Pin
int led=FIXME;
// Laenge eines Dit in Millisekunden
int timedit=FIXME;

void setup() {
  // Pin in Ausgabemodus...
  pinMode(led,OUTPUT);
}

void loop() {
  // Lange Pause, bevors losgeht...
  digitalWrite(led,LOW);
  delay(10000);

  // Sendesequenz:
  sendDit();
  pause("dit");
  sendDah();
  pause("dit");
  sendDah()FIXME
  pause("zeichen");
  sendDah();
  pause("dit");
  sendDit();
  pause(FIXME);
```

```
}

// Eigene Funktionen
void sendDit() {
  digitalWrite(led,LOW);
  digitalWrite(led,HIGH);
  // ein "timedit" lang anlassen!
  delay(timedit);
  digitalWrite(led,LOW);
}

void sendDah() {
  digitalWrite(led,LOW);
  digitalWrite(led,HIGH);
  // drei "timedit" lang anlassen!
  delay(3*timedit);
  digitalWrite(led,LOW);
}

void pause(String dauer) {
  if ( dauer == "dit" ) {
    digitalWrite(led,LOW);
    // ein "timedit" lang anlassen!
    delay(timedit);
    digitalWrite(led,LOW);
  }
  if ( dauer == "zeichen" ) {
    digitalWrite(led,LOW);
    // drei "timedit" lang anlassen!
    delay(3*timedit);
    digitalWrite(led,LOW);
  }
  if ( dauer == "wort" ) {
    digitalWrite(led,LOW);
    // drei "timedit" lang anlassen!
    delay(7*timedit);
    digitalWrite(led,LOW);
  }
}
}
```

**(A2)**

- Ergänze dein Setup um eine weitere (rote) LED, verwende zum morsen eine grüne oder gelbe

LED. Füge die neue LED in das setup des Programms ein.

- Ergänze das Programm um eine Funktion `sendeBuchstabe` (siehe Vorlage unten). Diese Funktion soll ein Zeichen als Argument erhalten, das Sie als Morsecode ausgibt.
 - Implementiere einige Buchstaben, so dass du ein kurzes Wort ausgeben kannst.
 - Überlege dir, was bei einem Leerzeichen passieren sollte.
 - Lasse die rote LED drei Sekunden leuchten, wenn ein Buchstabe übergeben wird, für den kein Morsecode hinterlegt ist.

Achtung! Zeichenketten (Strings) werden in doppelte Anführungszeichen eingeschlossen, einzelne Zeichen (Chars) werden in einfach anführungszeichen eingeschlossen!

```
[ .... Globale variablen, Setup und so Zeug .... ]

void loop() {
  [....]

  // Sendesequenz:
  sendBuchstabe('a');
  sendBuchstabe('b');

  [....]
}

// Die Buchstabenfunktion - ein Skelett
// Es wird der Buchstabe erwartet, der gesendet werden soll
// Die anderen Funktionen werden verwendet, um den Buchstaben zu senden
// Arbeit: Die Morsecodes müssen hinterlegt werden.
void sendBuchstabe(char buchstabe) {

  switch (buchstabe) {
    case 'a':
      //Morsecode für a
      sendDit();
      pause("dit");
      sendDah();
      break;
    case 'b':
      sendDah();
      pause("dit");
      sendDit();
      pause("dit");
      sendDit();
      pause("dit");
      sendDit();
      break;
    case [....]
      [....]
  }
  // Mindestend ein dit Pause zwischen den Zeichen
  pause("dit");
}
```

```
}  
[ .... die anderen Funktionen .... ]
```

1)

Seite „Morsezeichen“. In: Wikipedia, Die freie Enzyklopädie. Bearbeitungsstand: 19. Juni 2016, 12:51 UTC. URL: <https://de.wikipedia.org/w/index.php?title=Morsezeichen&oldid=155440419> (Abgerufen: 19. Juni 2016, 16:28 UTC)

2)

Die Codetabelle für Morsecode findest du hier:

<https://de.wikipedia.org/wiki/Morsezeichen#Standard-Codetabelle>

From:

<https://wiki.qg-moessingen.de/> - QG Wiki

Permanent link:

https://wiki.qg-moessingen.de/faecher:nwt:bluetooth_steckdosen:433_mhz:morsecode:start

Last update: **14.02.2023 18:56**

