

Abgabe

Dieses Übungsblatt ist bis Freitag, 02.05. um 12:00 per E-Mail an den eigenen Tutoren abzugeben.
Benennung beispielsweise “\$Matrikelnummer_Abgabe_\$Blattnummer.\$Format”.

1 Erste Schritte mit dem Galileo

Um überhaupt produktiv arbeiten zu können, sind eine Reihe grundlegender Vorbereitungen notwendig:

1. Treten Sie in Kontakt mit den Mitgliedern Ihrer Gruppe. Die E-Mail-Adressen und Namen Ihrer Mitstreiter werden Ihnen mitgeteilt.
2. Vereinbaren Sie einen Termin mit Ihren Gruppenmitgliedern, an dem Sie Ihr Board gemeinsam am Lehrstuhl abholen. Pfand pro Gruppe: 50 EUR. Lernen Sie sich untereinander kennen.
3. Um das Board zu nutzen, muss es zunächst per Netzteil an den Strom und dann per USB an den Rechner angeschlossen werden. Weitere Informationen in Foliensatz 1 von der Webseite.
4. Unter Windows müssen Sie möglicherweise zusätzliche Treiber installieren. Weitere Informationen auf der Webseite.
5. Installieren Sie die Galileo-Entwicklungsumgebung. Links und Informationen auf der Webseite.

Folgendes Programm bringt die interne LED eines Galileo-Boards zum Leuchten. Führen Sie das Programm auf Ihrem Galileo-Board aus.

```
int led = 13;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
}
```

Das Programm bringt die LED Ihres Galileos zum Dauerleuchten. Was müssen sie ändern um die LED zum Blinken zu bringen?

Wandeln Sie das Programm so ab, dass jeweils eines der folgenden Blinkmuster durchgegeben wird:

- a. 3 Sekunden an, 3 Sekunden aus
- b. 1 Sekunde an, 3 Sekunden aus
- c. 3 Sekunden an, 1 Sekunde aus

Beispiellösung.

```
int led = 13;
int delay_on = 1000;
int delay_off = 1000;

/*
  a) delay_on = 3000, delay_off = 3000
  b) delay_on = 1000, delay_off = 3000
  c) delay_on = 3000, delay_off = 1000
*/

void setup()
{
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
  delay(delay_on);
  digitalWrite(led, LOW);
  delay(delay_off);
}
```

2 Morsecode

Auf der Webseite finden Sie den Sketch “Morse.ino” aus der Vorlesung vom 22.04.

1. Wandeln Sie diesen Sketch durch Ergänzen und Aufrufen eigener Funktionen so ab, dass er die Nachricht “Hallo Welt” morsen kann.
2. **Bonusaufgabe:** Wie lässt sich der Sketch so abwandeln, dass er einen Nachrichtenteil (z.B. “Hallo”) auf einer LED und einen anderen Nachrichtenteil (z.B. “Welt”) auf einer anderen LED anzeigt?
3. Im Sketch existiert eine rekursiv definierte Funktion zum Morsen der einzelnen Ziffern einer Dezimalzahl. Implementieren Sie analog eine Funktion:
 - Die eine Zahl als Parameter bekommt.
 - Die die Binärrepräsentation der angegebenen Zahl morst.
 - Dabei soll eine 0 als *Dit* und eine 1 als *Dah* gemorst werden.

Beispiellösung.

```
int led = 13; // led to use for output (13 is built-in)

int dit_delay = 500; // length of a dit in ms
int dah_delay = dit_delay * 3; // length of a dah in ms

// dit() and dah() already include dit_delay
int letter_delay = dah_delay - dit_delay;

// letters already include letter delay
int word_delay = dit_delay * 7 - letter_delay;

void dit() {
  Serial.print(".");

  // send a dit
  digitalWrite(led, HIGH);
  delay(dit_delay);

  digitalWrite(led, LOW);
  delay(dit_delay);
}

void dah() {
  Serial.print("-");

  // send a dah
  digitalWrite(led, HIGH);
  delay(dah_delay);

  digitalWrite(led, LOW);
  delay(dit_delay);
}
```

```
void pause_letter() {
    Serial.println("");
    delay(letter_delay);
}

void pause_word() {
    Serial.println("");
    delay(word_delay);
}

void morse_S() {
    dit();
    dit();
    dit();
    pause_letter();
}

void morse_I() {
    dit();
    dit();
    pause_letter();
}

void morse_N() {
    dah();
    dit();
    pause_letter();
}

void morse_K() {
    dah();
    dit();
    dah();
    pause_letter();
}

void morse_SINK() {
    morse_S();
    morse_I();
    morse_N();
    morse_K();
    pause_word();
}

// send n in morse code
void morse_digit(int n) {
    Serial.print("morse_digit(");
    Serial.print(n);
    Serial.println(")...");
}
```

```
if (n == 0) {
    dah();
    dah();
    dah();
    dah();
    dah();
}
if (n == 1) {
    dit();
    dah();
    dah();
    dah();
    dah();
}
if (n == 2) {
    dit();
    dit();
    dah();
    dah();
    dah();
}
if (n == 3) {
    dit();
    dit();
    dit();
    dah();
    dah();
}
if (n == 4) {
    dit();
    dit();
    dit();
    dit();
    dah();
}
if (n == 5) {
    dit();
    dit();
    dit();
    dit();
    dit();
}
if (n == 6) {
    dah();
    dit();
    dit();
    dit();
    dit();
}
if (n == 7) {
    dah();
```

```
        dah();
        dit();
        dit();
        dit();
    }
    if (n == 8) {
        dah();
        dah();
        dah();
        dit();
        dit();
    }
    if (n == 9) {
        dah();
        dah();
        dah();
        dah();
        dit();
    }
    pause_letter();

    Serial.print("morse_digit(");
    Serial.print(n);
    Serial.println(")...done");
}

void morse_number(int n) {
    Serial.print("morse_number(");
    Serial.print(n);
    Serial.println(")...");

    if (n >= 10) {
        morse_number(n / 10);
    }
    morse_digit(n % 10);

    Serial.print("morse_number(");
    Serial.print(n);
    Serial.println(")...done");
}

void morse_H() {
    dit();
    dit();
    dit();
    dit();
    pause_letter();
}

void morse_A() {
```

```
    dit();
    dah();
    pause_letter();
}

void morse_L() {
    dit();
    dah();
    dit();
    dit();
    pause_letter();
}

void morse_O() {
    dah();
    dah();
    dah();
    pause_letter();
}

void morse_W() {
    dit();
    dah();
    dah();
    pause_letter();
}

void morse_E() {
    dit();
    pause_letter();
}

void morse_T() {
    dah();
    pause_letter();
}

void morse_HALLO() {
    morse_H();
    morse_A();
    morse_L();
    morse_L();
    morse_O();
    pause_word();
}

void morse_WELT() {
    morse_W();
    morse_E();
    morse_L();
    morse_T();
}
```

```
    pause_word();
}

void morse_binary(int n) {
    if (n >= 2) {
        morse_binary(n / 2);
    }
    if (n % 2 == 0) {
        dit();
        Serial.print(0);
    } else {
        dah();
        Serial.print(1);
    }
}

void setup() {
    // put your setup code here, to run once:

    // Configure LED as output
    pinMode(led, OUTPUT);

    // Use serial port as output
    Serial.begin(9600);
}

void loop() {
    morse_HALLO();
    led = 12;
    morse_WELT();
    led = 13;
    morse_binary(5024);
}
```


3 Fehlersuche

In den folgenden (jeweils in sich vollständigen) Programme findet sich jeweils mindestens ein Fehler. Finden Sie so viele Fehler, wie möglich.

- Was passiert beim Übersetzen dieser Programme?
- Gab es Fehlermeldungen? Wenn ja, was bedeuten sie?
- Falls es keine Fehlermeldungen gab: Verhält das Programm sich korrekt?
- Wie lassen sich die Fehler korrigieren?

Listing 1: Ein fehlerhaftes Programm

```
int led_Red := 12;
int led_Green := 12+1;

void setup {
  pin_mode(led_red; OUTPUT);
  pin_mode(led_green; OUTPUT);
}

void loop {
  digital_write(led_red; HIGH);
  digital_write(led_green; LOW);
}
```

Beispiellösung. *Das Programm hat folgende Fehler:*

- inkonsistente Variablennamen
- der Zuweisungsoperator ist '=', nicht ':='
- Funktionsparameter werden mit einem Komma statt Semikolon getrennt
- die Funktionen `pinMode` und `digitalWrite` sind falsch geschrieben

Eine korrigierte Version des Programms könnte etwa wie folgt aussehen:

```
int led_red = 12;
int led_green = 12+1;

void setup {
  pinMode(led_red, OUTPUT);
  pinMode(led_green, OUTPUT);
}

void loop {
  digitalWrite(led_red, HIGH);
  digitalWrite(led_green, LOW);
}
```

Mit dem Programm würde die rote LED dauerhaft leuchten und die grüne aus bleiben.

Listing 2: Ein anderes fehlerhaftes Programm

```
loop() {  
    digitalWrite(13, HI)  
    delay(1.0s)  
    digitalWrite(13, LO)  
}
```

Beispiellösung. Das Programm hat folgende Fehler:

- die *setup()* Funktion fehlt
- der *pinMode* fuer Pin 13 muss auf *OUTPUT* gesetzt werden
- der Rückgabetyt (*void*) fehlt bei der *loop()* Funktion
- die Konstanten *HIGH* und *LOW* sind falsch geschrieben
- der Parameter fuer die Funktion *delay* ist ein Integer
- Semikolons fehlen am Ende der Zeile

Eine korrigierte Version des Programms könnte etwa wie folgt aussehen:

```
void setup() {  
    pinMode(13, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
}
```

Mit dem Programm würde die eingebaute LED dauerhaft leuchten, falls sie blinken sollte müsste man noch ein weiteres *delay()* einbauen.