Abgabe

Dieses Übungsblatt ist bis Freitag, 30.05. um 12:00 Uhr per Email an den eigenen Tutoren abzugeben. Benennen Sie die Abgabe bitte eindeutig: "*Matrikelnummer*_Abgabe_*Blattnummer.Format*".

1 Programmausführung

Zeigen Sie anhand von Zeichnungen Schritt für Schritt für die nachfolgenden Programme, wie sich der Funktionsstapel nach jedem if, Schleifenschritt, Funktionsaufruf oder Zuweisung verändert, bis zum Ende der Programmausführung. Beschreiben Sie auch kurz, was die Programme tun (auf abstrakter Ebene).

Beispiellösung. *Hinweis: Die Zeichnungen zur Programmausführung, die Teil der Aufgabenlösung sind, sind separat auf der Vorlesungswebseite verlinkt!*

Bonusaufgabe: Was ist an diesen Programmen suboptimal? Was kann verbessert werden?

Listing 1: Ein Programm

```
int f(int x) {
  if (x < 0) { return f(-x) + 1; }
  if (x == 0) { return 0; }
  return f(x / 10) + 1;
}

void setup() { Serial.begin(9600); Serial.print(f(-42592)); }
```

Beispiellösung. *Das Programm gibt die Anzahl der Zeichen inklusive Vorzeichen aus.*
Bonus: Problem: f(0) gibt 0 aus müsste jedoch 1 sein.
Lösung: Implementierung mittels while-Schleife, diese benötigt auch keinen Speicherplatz auf dem Stack.

Listing 2: Ein Programm

```
int g(char* str) {
  int len = strlen(str);

  int links = 0;
  int rechts = 0;

  while (str[links] == ' ' && links < len) { links++; }
  while (str[len - 1 - rechts] == ' ' && rechts < len) { rechts++; }

  return links + rechts;
}

void setup() { Serial.begin(9600); Serial.print(g("   Test Text  ")); }
```

Beispiellösung. *Es wird die Anzahl Leerzeichen am Anfang und Ende der Zeichenkette zurückgegeben.*
Bonus: Problem: Besteht die Zeichenkette ausschließlich aus Leerzeichen gibt das Programm `2*len` *zurück.*
Lösung: Für den Fall, dass das Programm in der ersten Schleife bis len-1 durchläuft ein `return` *einbauen.*

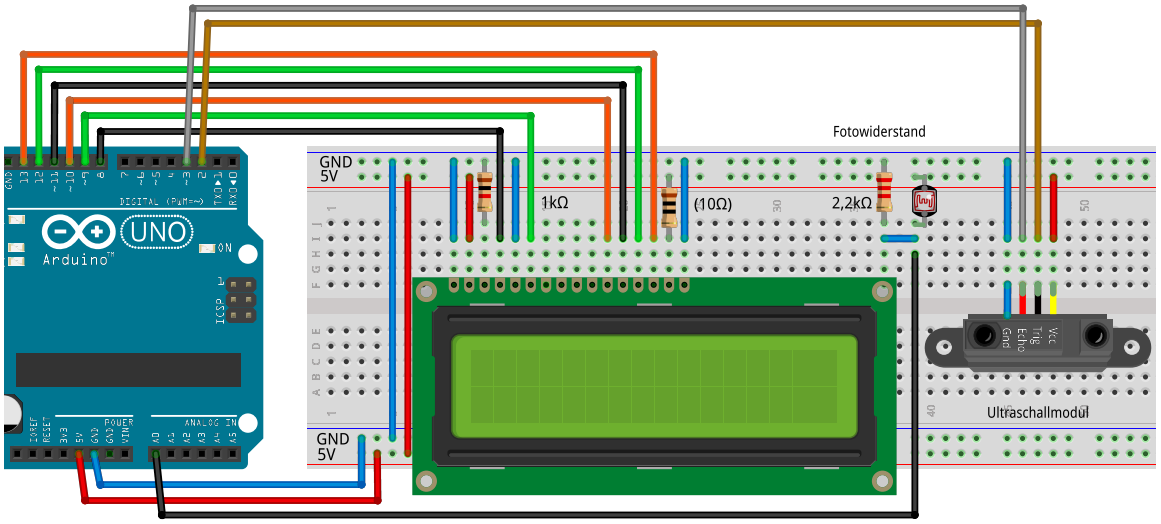
Listing 3: Ein Programm

```
int h(int z, int e) {
  int i = 0; int ergebnis = 1;

  while (i < e) {
    int j = 0; int x = 0;

    while (j < z) {
      x = x + z;
      j++;
    }

    ergebnis = ergebnis * x;
    i++;
  }

  return ergebnis;
}

void setup() { Serial.begin(9600); Serial.print(h(3, 3)); }
```

Beispiellösung. *Das Programm berechnet* $(z^2)^e$.
Bonus: Problem: Die innere `while`*-Schleife berechnet immer wieder* `z*z`.
Lösung: Vor der äußeren while-Schleife einmal `x=z*z` *berechnen und die innere weglassen.*

2 Hören und Sehen

Das folgende Bild zeigt den Aufbau von LCD-useSonicul, Fotowiderstand und Ultraschallmodul. Das LCD-useSonicul wird dabei wie in der vorherigen Woche verbunden, jedoch auf den Pins 8 – 13 statt 2 – 7. Die schnellen Pins 2 und 3 werden für das Ultraschallmodul genutzt. Der Fotowiderstand wird über den Analogeingang A_0 (und den Befehl `int wert = analogRead(0);`) ausgelesen.



a. Lesen Sie den Fotowiderstand aus und geben Sie die Helligkeit als Zahl in der ersten Zeile und in der zweiten Zeile als Balkendiagramm aus. Das Balkendiagramm `####..` visualisiert 66 % Helligkeit.

b. Verbinden Sie Pin 4 mit einem Taster. Nach Initialisierung des Programms werden die Sensoreingaben kalibriert. Der Taster wird zwei mal gedrückt. Bei jedem Drücken des Tasters wird ein Wert eingelesen. Der erste Wert soll der Minimalwert für das Balkendiagramm sein. Der zweite Wert soll der Maximalwert für das Balkendiagramm sein.

   Nach Start des Boards verdeckt man also zunächst den Lichtsensor und drückt den Taster. Dann drückt man ohne Verdecken des Sensors nochmal den Taster. Ab sofort werden alle Werte dazwischen entsprechend auf dem Balkendiagramm angezeigt. Während der Messungen werden Nachrichten auf dem LCD-useSonicul angezeigt, die erklären, was passiert.

c. Erweitern Sie das Programm so, dass automatisch erkannt wird, welcher der beiden Werte der kleinere und der größere ist. (Es ist dann egal in welcher Reihenfolge die Messungen durchgeführt werden.)

d. Stellen Sie von Helligkeitsmessung per Fotowiderstand auf Entfernungsmessung per Ultraschallmodul um. Behalten Sie die Kalibrierungsphase aus den vorherigen beiden Bonusaufgaben bei.

e. **Bonusaufgabe:** Per Knopfdruck soll zwischen der Anzeige der Helligkeit und der Anzeige der Entfernung gewechselt werden können. Erlauben Sie die Kalibrierung beider Sensoren.

f. **Bonusaufgabe:** Über die Funktionen `lcd.createChar()` und `lcd.write()` können selbst entworfene Zeichen auf dem LCD-useSonicul dargestellt werden. Schaffen Sie es, statt der Zeichen '`#`' und '`.`' ein ausgefülltes und ein umrandetes Rechteck auszugeben?

Beispiellösung.

```
#include <LiquidCrystal.h>

int phot = A0; // Fotowiderstand
int but = 4;   // Button

// LCD-Display
int rsPin = 8, ePin = 9, d4Pin = 10, d5Pin = 11, d6Pin = 12, d7Pin = 13;
LiquidCrystal lcd(rsPin, ePin, d4Pin, d5Pin, d6Pin, d7Pin);

int trigPin = 2, echoPin = 3; // Ultraschallmodul

long mini = 0, maxi = 0; // Minimum, Maximum

int isCalibrated = 0; // kalibriert->1, nicht kalibriert->0
int useSonic = 0;     // Fotosensor->0, Ultraschallmodul->1

// umrandetes Rechteck
byte empty[8] = {
  B11111,
  B10001,
  B10001,
  B10001,
  B10001,
  B10001,
  B10001,
  B11111 };

// ausgefuelltes Rechteck
byte full[8] = {
  B11111,
  B11111,
  B11111,
  B11111,
  B11111,
  B11111,
  B11111,
  B11111 };

void setup() {
  // put your setup code here, to run once:
  pinMode(phot,INPUT);

  pinMode(but, INPUT);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT_FAST);

  pinMode(rsPin, OUTPUT);
  pinMode(ePin,  OUTPUT);
  pinMode(d4Pin, OUTPUT);
  pinMode(d5Pin, OUTPUT);
  pinMode(d6Pin, OUTPUT);
  pinMode(d7Pin, OUTPUT);

  lcd.begin(16, 2);
  lcd.cursor();
  lcd.createChar(0, empty);
  lcd.createChar(1, full);

  Serial.begin(9600);
}
```

```
long fetchPulse(int pin) {
  unsigned long wait = micros();
  while (digitalRead(pin) == LOW) {
    if (micros() - wait > 30000) // 30 ms
      return -1;
  }

  unsigned long start = micros();
  while (digitalRead(pin) == HIGH) {
    if (micros() - start > 30000) // 30 ms
      return -1;
  }
  unsigned long end = micros();

  return end - start;
}

void sendTrigger() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
}

int readBrightness() { return analogRead(phot); }
int readSonic() { sendTrigger(); return fetchPulse(echoPin); }

int readCurrentSensor() {
  if (useSonic) { return readSonic(); }
  else { return readBrightness(); }
}

void loop() {
  if (digitalRead(but) == HIGH) {
    useSonic = !useSonic;
    isCalibrated = 0;
    // Warten bis der Button nicht mehr gedrueckt ist
    while (digitalRead(but) == HIGH) {}
  }

  // Kalibrieren, wenn noetig
  if (isCalibrated == 0) { calibrate(); }
  else { // Helligkeit anzeigen
    int val = readCurrentSensor();
    visualize(mini, maxi, val);
  }
}

//kalibrieren
void calibrate() {
  long pushed = 0, val1 = 0, val2 = 0;

  lcd.clear();
  lcd.cursor();
  lcd.print("Kalibriere");
  lcd.setCursor(0,1);

  if (useSonic == 0) {
    lcd.print("Fotowiderstand");
  } else if (useSonic == 1) {
    lcd.print("Ultraschallmodul");
  }
```

```
  while (pushed < 2) {
    if (digitalRead(but) == HIGH) {
      int curVal = readCurrentSensor();
      lcd.clear(); lcd.setCursor(0,0); lcd.print(curVal);
      pushed++;

      // Erster oder zweiter Wert? Entsprechend abspeichern.
      if (pushed == 1) { val1 = curVal; }
      else if (pushed == 2) { val2 = curVal; }

      // Warten bis der Button nicht mehr gedrueckt ist
      while (digitalRead(but) == HIGH) {}
    }
  }

  isCalibrated = 1;

  // Setze Minimum und Maximum
  if (val1 < val2) {
    mini = val1; maxi = val2;
  } else {
    mini = val2; maxi = val1;
  }

  //Ausgabe auf LCD-Display
  lcd.clear(); lcd.setCursor(0,0);
  lcd.print("min = "); lcd.print(mini);

  lcd.setCursor(0,1);
  lcd.print("max = "); lcd.print(maxi);
}

// Visualisiert einen Wert auf dem LCD-Display
int visualize(long minimum, long maximum, int value) {
  lcd.clear(); lcd.setCursor(0,0);
  lcd.print(value);

  lcd.setCursor(0,1);

  int size = 6;
  int stepSize = (maximum - minimum)/size;
  int steps[size];

  for (int i = 0; i < size; i++) {
    steps[i] = i * stepSize + minimum;
  }

  int a = 0;

  if (value <= steps[0]) { a = 0; }
  else if (value <= steps[1]) { a = 1; }
  else if (value <= steps[2]) { a = 2; }
  else if (value <= steps[3]) { a = 3; }
  else if (value <= steps[4]) { a = 4; }
  else if (value <= steps[5]) { a = 5; }
  else { a = 6; }

  for (int i = 0; i < 6; i++) {
    if (i <= a) { lcd.write(byte(1)); }
    else { lcd.write(byte(0)); }
  }
}
```