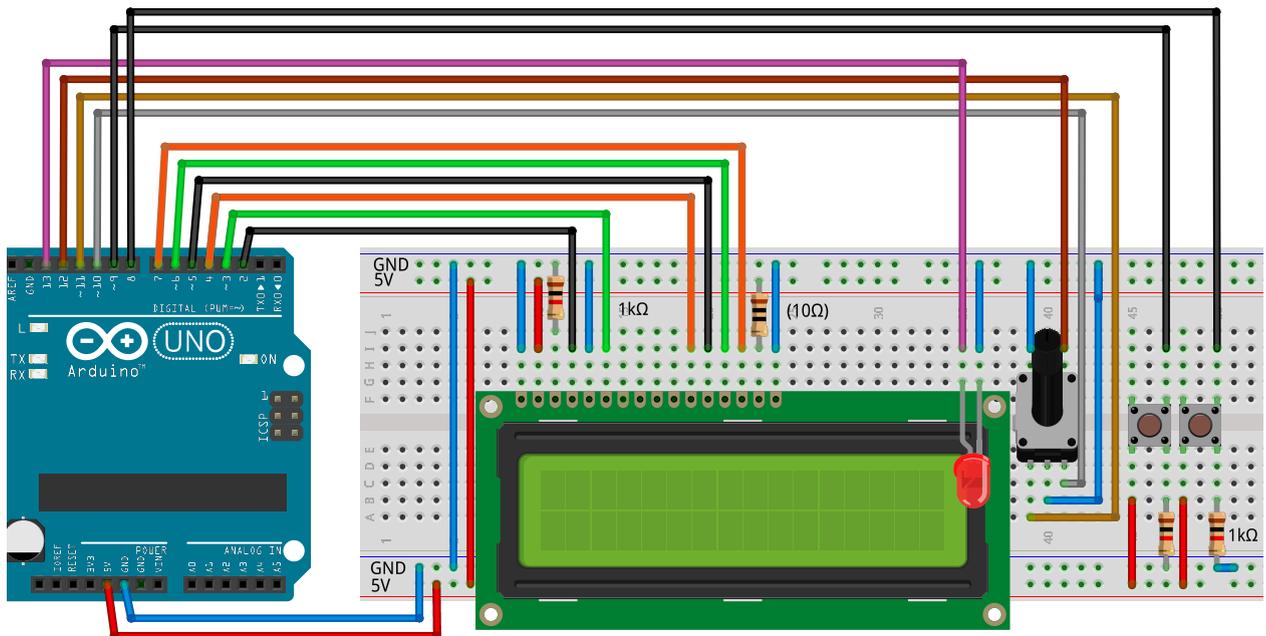


## Abgabe

Dieses Übungsblatt ist bis Freitag, 23.05. um 12:00 Uhr per Email an den eigenen Tutoren abzugeben. Benennen Sie die Abgabe bitte eindeutig: *“Matrikelnummer\_Abgabe\_Blattnummer.Format”*.

## 1 Schaltung für einen Getränkeautomaten

Das folgende Bild zeigt den Aufbau inklusive zweier Push-Buttons mit Pull-Down-Widerstand. Die Verbindung des 15. Pins des LCD-Modules mit dem 10- $\Omega$ -Widerstand (in der Zeichnung in Klammern) aktiviert die Hintergrundbeleuchtung und ist optional. Alle anderen Widerstände betragen jeweils 1 k $\Omega$ . Der Drehimpulsgeber hat insgesamt 5 Pins, die verbunden werden müssen. Nach Anlöten der Stiftleisten kann das LCD-Modul direkt ins Breadboard gesteckt werden. Wir werden einen Lötservice anbieten.



## 2 Bezahlvorgang

Implementieren Sie auf dem Sketch der Vorlesungswebseite aufbauend Zahlungsfunktionalität:

- Nach Auswahl eines Getränks durch Klicken des Drehimpulsgebers soll der Bezahlvorgang beginnen.
- Stellen Sie während des Bezahlvorgangs den Namens des ausgewählten Getränks, den bereits gezahlten Betrag und den noch zu zahlenden Betrag dar.
- Die Bezahlung des Getränks wird durch Drücken der zwei Taster simuliert. Der linke Taster entspricht dem Einwurf von 0,50€, der rechte Taster dem Einwurf von 1€.
- Nach erfolgter Bezahlung leuchtet die rote LED. Nach Bestätigung durch Klicken des Drehimpulsgebers kehrt der Automat zurück ins Hauptmenü.

## Beispiellösung.

```
#include <LiquidCrystal.h>

int DRINKS = 3;
char *drink_name[] = { "Wasser", "Limo", "Bier" };
double drink_price[] = { 1.00, 1.50, 2.00 };

// LCD-Anzeige
int rsPin = 2;
int ePin = 3;
int d4Pin = 4;
int d5Pin = 5;
int d6Pin = 6;
int d7Pin = 7;

LiquidCrystal lcd(rsPin, ePin, d4Pin, d5Pin, d6Pin, d7Pin);

int button50Cent = 8;
int button100Cent = 9;

// Schalter
int turnLeftPin = 10;
int turnRightPin = 11;
int selectPin = 12;

int led = 13;

void print_drinks() {
    char buffer[200];
    strcpy(buffer, "");

    for (int i = 0; i < DRINKS; i++)
    {
        if (i > 0)
            strcat(buffer, " ");
        strcat(buffer, drink_name[i]);
    }
    lcd.print(buffer);
}

void print_prices() {
    int x = 0;
    for (int i = 0; i < DRINKS; i++) {
        char buffer[100];

        lcd.setCursor(x, 1);
        sprintf(buffer, "%.2f", drink_price[i]);
        Serial.println(buffer);
        lcd.print(buffer);
        x += strlen(drink_name[i]) + 1;
    }
}
```

```
void show_selection(int drink) {
    int x = 0;

    for (int i = 0; i < drink; i++) {
        x += strlen(drink_name[i]);
        x += strlen(" ");
    }
    lcd.setCursor(x, 0);
    lcd.blink();
}

// Getraenk auswählen und dessen Nummer zuruckgeben
int choose_drink() {
    int current_selection = 0;
    unsigned long last_select = millis();

    show_selection(current_selection);

    while (1) {
        if (millis() - last_select > 500) {

            // Drehen nach rechts
            if (digitalRead(turnRightPin) == LOW) {
                if (current_selection < DRINKS - 1) {
                    current_selection++;
                }
                show_selection(current_selection);
                last_select = millis();
            }

            // Drehen nach links
            if (digitalRead(turnLeftPin) == LOW) {
                if (current_selection > 0) {
                    current_selection--;
                }
                show_selection(current_selection);
                last_select = millis();
            }

            // Drucken
            if (digitalRead(selectPin) == LOW) {
                lcd.noBlink();
                return current_selection;
            }
        }
    }
}

void displayPayment(double toPay, double hasPaid) {
    lcd.setCursor(0, 1);
    lcd.print(toPay);
    lcd.print(" - ");
    lcd.print(hasPaid);
}
```

```
void pay_for_drink(int selection) {
    double toPay = drink_price[selection];    double hasPaid = 0.0;
    unsigned long last_select = millis();

    lcd.clear();  lcd.home();
    lcd.print(drink_name[selection]);
    displayPayment(toPay, hasPaid);

    while (toPay > 0.0) {
        if (millis() - last_select > 250) {
            if (digitalRead(button50Cent) == HIGH) {
                hasPaid += 0.5;
                toPay -= 0.5;
                last_select = millis();
            }

            if (digitalRead(button100Cent) == HIGH && toPay >= 1.0) {
                hasPaid += 1.0;
                toPay -= 1.0;
                last_select = millis();
            }
        }
        displayPayment(toPay, hasPaid);
    }
}

void dispense_drink(int selection) {
    digitalWrite(led, HIGH);
    while (digitalRead(selectPin) == HIGH) {
        // Wait until selection button has been pressed.
    }
    digitalWrite(13, LOW);
}

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

    pinMode(turnLeftPin,  INPUT_PULLUP);
    pinMode(turnRightPin, INPUT_PULLUP);
    pinMode(selectPin,    INPUT_PULLUP);

    pinMode(button50Cent, INPUT);
    pinMode(button100Cent, INPUT);
    pinMode(led, OUTPUT);

    lcd.begin(16, 2);
    lcd.cursor();

    Serial.begin(9600);
}
```

```
void loop() {  
    // put your main code here, to run repeatedly:  
    lcd.clear();  
    print_drinks();  
    print_prices();  
    int selection = choose_drink();  
  
    // Weitere Funktionen, etwa:  
    pay_for_drink(selection);  
    dispense_drink(selection);  
}
```

### 3 Fehlersuche

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

Listing 1: Ein fehlerhaftes Programm

```
// Gewuenshtes Verhalten: Den String an sich selbst haengen  
void setup() {  
    char[] str = {'a', 'b', 'c', 'd'};  
    strcat(str, str);  
}
```

#### Beispiellösung.

```
void setup() {  
    char str[10] = {'a', 'b', 'c', 'd'};  
    strcat(str, str);  
}
```

Listing 2: Ein fehlerhaftes Programm

```
int taster = 2;
int led = 3;
int tasterGedrueckt = 0;

// Gewuenshtes Verhalten: LED leuchtet, wenn Taster
// zwischen einschl. 6x und einschl. 9x gedrueckt
void loop() {
    tasterGedrueckt = digitalRead(taster);

    if (tasterGedrueckt = 1) { zaehler++; }

    if (5 < zaehler < 10) {
        digitalWrite(led, HIGH);
    } else {
        digitalWrite(led, LOW);
    }
}
```

### Beispiellösung.

```
int taster = 2;
int led = 3;
int zaehler = 0;
unsigned long last_input = millis();

void loop() {
    // Taster entprellen
    if (digitalRead(taster) = HIGH && millis() - last_input > 50) {
        zaehler++;
        last_input = millis();
    }

    if (5 < zaehler && zaehler < 10) {
        digitalWrite(led, HIGH);
    } else {
        digitalWrite(led, LOW);
    }
}
```

Listing 3: Ein fehlerhaftes Programm

```
int led = 3;
int naechsterWechsel = 0;

void setup() { naechsterWechsel = millis + 300; }

// Gewuenshtes Verhalten: LED blinkt ohne Einsatz von delay()
void loop() {
  if (millis > naechsterWechsel) {
    digitalWrite(!digitalRead(led));
    naechsterWechsel = millis + 300;
  }
}
```

### Beispiellösung.

```
int led = 3;
int naechsterWechsel = 0;
int ledLeuchtet = HIGH;

void setup() {
  pinMode(led, OUTPUT);
  naechsterWechsel = millis() + 300;
}

void loop() {
  if (millis() > naechsterWechsel) {
    digitalWrite(led, ledLeuchtet);
    ledLeuchtet = !ledLeuchtet;
    naechsterWechsel = millis() + 300;
  }
}
```

Listing 4: Ein fehlerhaftes Programm

```
int taster = 2;
int led = 3;

// Gewuenshtes Verhalten: LED leuchtet, wenn Taster gedruickt
void loop() {
  if (taster == HIGH) {
    led = HIGH;
  } else {
    led = LOW;
  }
}
```

### Beispiellösung.

```
int taster = 2;
int led = 3;

void loop() { digitalWrite(led, digitalRead(taster)); }
```

Listing 5: Ein fehlerhaftes Programm

```
int ledLeuchtet = 0;
int naechsterWechsel = 0;

// Gewuenshtes Verhalten: LED blinkt ohne Einsatz von delay()
void loop() {
  while (millis() <= naechsterWechsel) {
    if (millis() > naechsterWechsel) {
      digitalWrite(led, !ledLeuchtet);
      naechsterWechsel = millis() + 300;
    }
  }
}
```

### Beispiellösung.

```
int ledLeuchtet = 0;
int naechsterWechsel = 0;
int led = 3;

void loop() {
  if (millis() > naechsterWechsel) {
    digitalWrite(led, ledLeuchtet);
    ledLeuchtet = !ledLeuchtet;
    naechsterWechsel = millis() + 300;
  }
}
```

Listing 6: Ein fehlerhaftes Programm

```
int leds[] = {7, 8, 9};

// Gewuenshtes Verhalten: Pins 7-9 werden als OUTPUT initialisiert
// Zu initialisierende Pins sind in einem Array abgelegt
void setup() {
  int i = 7;
  while (i < 9) {
    pinMode(leds[i++], OUTPUT);
  }
}
```

### Beispiellösung.

```
int leds[] = {7, 8, 9};

void setup() {
  int i = 0;
  while (i < 3) {
    pinMode(leds[i++], OUTPUT);
  }
}
```