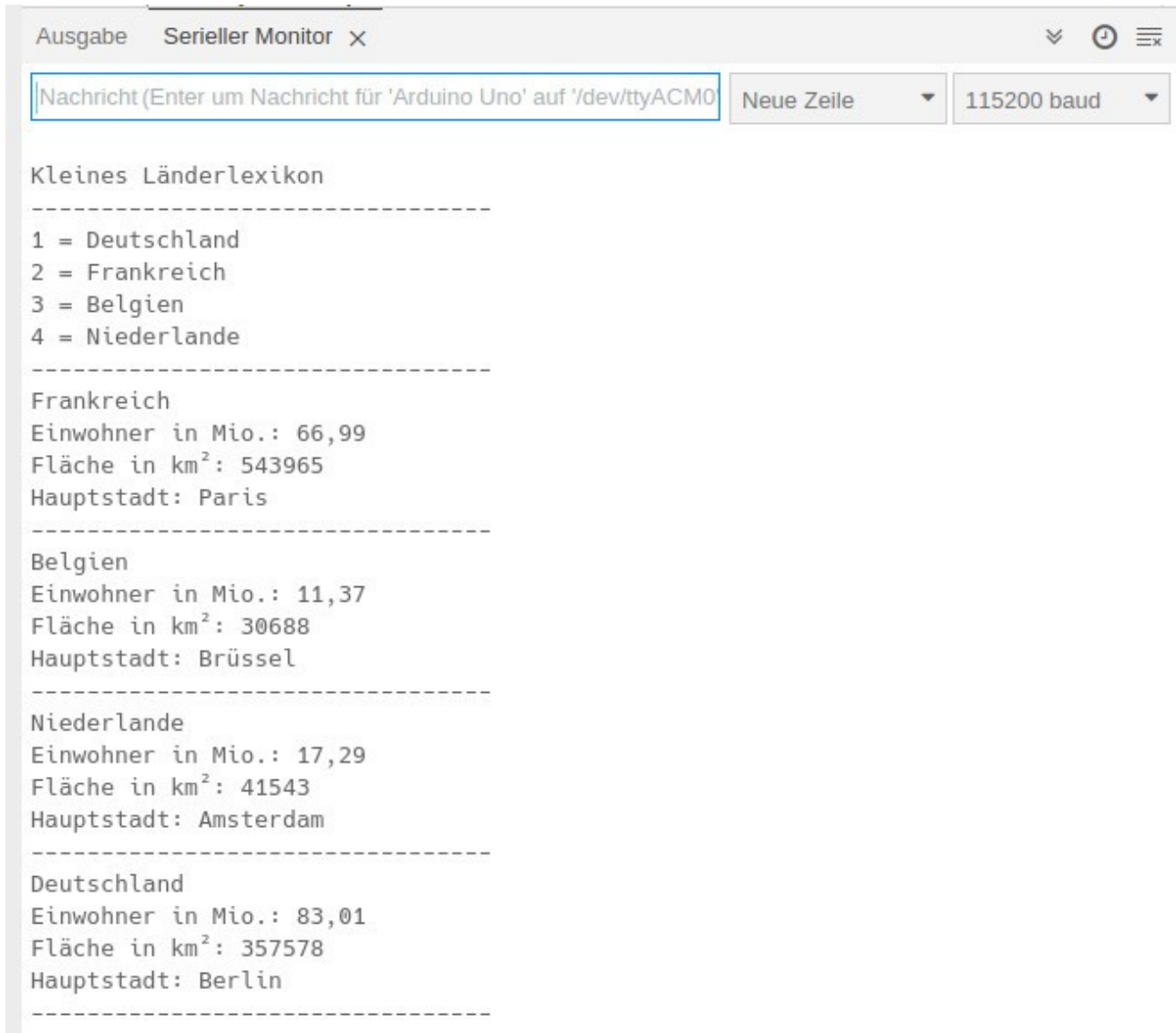


Das Programm soll nach einem Druck auf eine der Tasten des Tastenfelds Informationen zu verschiedenen Ländern im Seriellen Monitor oder auf einem LCD anzeigen.

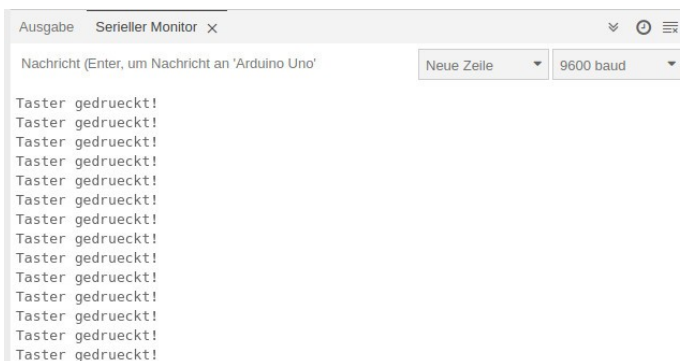


Bei Programmen, die einen Taster verwenden, taucht immer das Problem auf, dass ein längerer Druck auf den Taster die gewünschte Aktion mehrfach ausführt. Selbst ein kurzes delay() kann das Problem nicht lösen, sondern höchsten abmildern. Dieses Phänomen wird „Prellen“ genannt.

Das kleine Beispielprogramm verdeutlicht das Problem:

```
void setup()
{
  pinMode(5, INPUT_PULLUP);
  Serial.begin(9600);
}

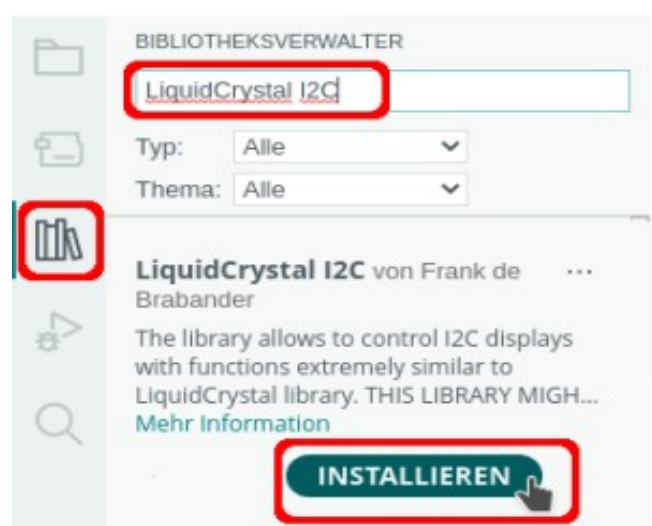
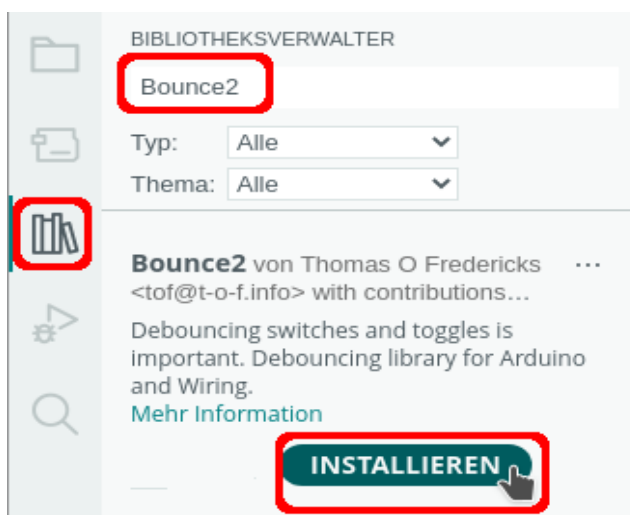
void loop()
{
  int TasterStatus = digitalRead(5);
  if (TasterStatus == LOW)
  {
    delay(200);
    Serial.println("Taster gedrueckt!");
  }
}
```



Ein längerer Druck auf den Taster produziert die mehrfache Ausgabe des Textes.

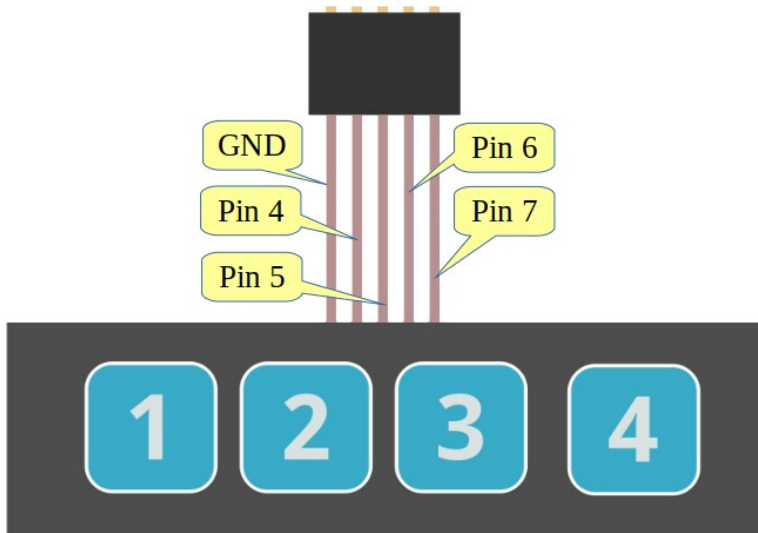
Die Bibliothek Bounce2 löst genau dieses Problem:

Benötigte Bibliotheken:

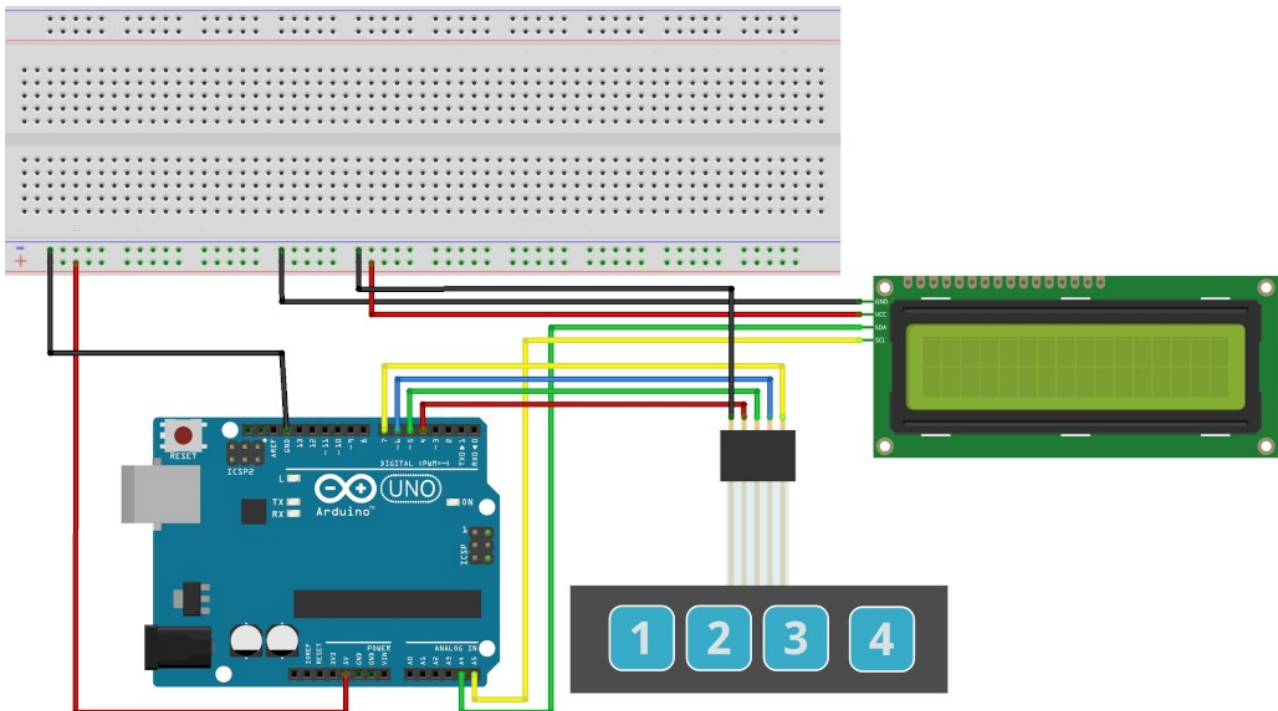


Benötigte Bauteile:

- ➡ LCD 1602
- ➡ Tastenfeld 1x4
- ➡ Leitungsdrähte



Baue die Schaltung auf.



fritzing



Du kannst außerdem mit einem kleinen Programm testen, ob die Tasten 1 bis 4 den richtigen Pins zugeordnet wurden. Die gedrückte Taste wird im Seriellen Monitor angezeigt.

```
int TASTER_EINS = 5; // Taste 1
int TASTER_ZWEI = 4; // Taste 2
int TASTER_DREI = 7; // Taste 3
int TASTER_VIER = 6; // Taste 4

void setup()
{
  Serial.begin(9600);

  // Vorwiderstand aktivieren
  pinMode(TASTER_EINS, INPUT_PULLUP);
  pinMode(TASTER_ZWEI, INPUT_PULLUP);
  pinMode(TASTER_DREI, INPUT_PULLUP);
  pinMode(TASTER_VIER, INPUT_PULLUP);
}

void loop()
{
  if (digitalRead(TASTER_EINS) == LOW)
  {
    delay(100);
    Serial.println("Taste 1 an Pin " + String(TASTER_EINS));
  }

  if (digitalRead(TASTER_ZWEI) == LOW)
  {
    delay(100);
    Serial.println("Taste 2 an Pin " + String(TASTER_ZWEI));
  }

  if (digitalRead(TASTER_DREI) == LOW)
  {
    delay(100);
    Serial.println("Taste 3 an Pin " + String(TASTER_DREI));
  }
  if (digitalRead(TASTER_VIER) == LOW)
  {
    delay(100);
    Serial.println("Taste 4 an Pin " + String(TASTER_VIER));
  }
}
```

Binde die benötigten Bibliotheken ein, definiere die struct, die Taster und die Zuordnung der Tasten zur Bibliothek Bounce2.



Arrays erweisen sich als geeignet, wenn eine Vielzahl von Daten eines gleichen Variablentyps verarbeitet werden sollen.

Strukturen (struct) sind hingegen eine Sammlung und Beschreibung verschiedener Daten.

Sie können beliebige Datentypen verwenden.

Die Struktur ist die Vorlage für die Definition der konkreten Daten.

Im Beispiel werden verschiedene Informationen zu den einzelnen Städten mit Hilfe unterschiedlicher Datentypen zusammengefasst.

Beispiel Definition als einzelne struct

Die Vorlage:

```
struct Laenderdaten
{
    String Name;
    float Einwohner;
    long Groesse;
    String Hauptstadt;
};
```

Definition der Daten:

```
Laenderdaten Deutschland =
{
    "Deutschland",
    83.01,
    357578,
    "Berlin"
};
```

```
Laenderdaten Frankreich =
{
    "Frankreich",
    66.99,
    543965,
    "Paris"
};
```

Der Zugriff auf die Elemente einer struct erfolgt über die Punkt-Schreibweise.

Land.Name → String Name

Landname.Einwohner → float Einwohner

StadtName.Einwohner → long Groesse

StadtName.Vorwahl → String Hauptstadt

```
# include <Bounce2.h>
# include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);

struct Laenderdaten
{
    String Name;
    float Einwohner;
    long Groesse;
    String Hauptstadt;
};

Laenderdaten Deutschland =
{
    "Deutschland",
    83.01,
    357578,
    "Berlin"
};

Laenderdaten Frankreich =
{
    "Frankreich",
    66.99,
    543965,
    "Paris"
};

Laenderdaten Belgien =
{
    "Belgien",
    11.37,
    30688,
    "Br\365ssel"
};

Laenderdaten Niederlande =
{
    "Niederlande",
    17.29,
    41543,
    "Amsterdam"
};

/*
    Anschlüsse von links nach rechts
    GND D4 D5 D7 D6
*/
int TASTER1 = 5; // Taste 1
int TASTER2 = 4; // Taste 2
int TASTER3 = 7; // Taste 3
int TASTER4 = 6; // Taste 4
```

```
// Bibliothek Bounce2
// "Prellverhinderer" für die Tasten starten
Bounce Land1 = Bounce();
Bounce Land2 = Bounce();
Bounce Land3 = Bounce();
Bounce Land4 = Bounce();
```

Im setup-Teil werden die Instanzen der Taster zugeordnet, das LCD-Display gestartet und der Vorwiderstand der Taster aktiviert.

```
void setup()
{
    // Instanzen des Objekts Bounce für jede Taste zuordnen
    // Zeitintervall einstellen
    Land1.attach(TASTER1);
    Land1.interval(20);

    Land2.attach(TASTER2);
    Land2.interval(20);

    Land3.attach(TASTER3);
    Land3.interval(20);

    Land4.attach(TASTER4);
    Land4.interval(20);

    // INPUT_PULLUP -> Eingangspegel auf HIGH setzen
    pinMode(TASTER1, INPUT_PULLUP);
    pinMode(TASTER2, INPUT_PULLUP);
    pinMode(TASTER3, INPUT_PULLUP);
    pinMode(TASTER4, INPUT_PULLUP);

    // LCD starten
    lcd.init();
    lcd.backlight();

    // Info über den Programmablauf
    lcd.setCursor(0, 0);
    lcd.print("1 = Deutschland");
    lcd.setCursor(0, 1);
    lcd.print("2 = Frankreich");
    lcd.setCursor(0, 2);
    lcd.print("3 = Belgien");
    lcd.setCursor(0, 3);
    lcd.print("4 = Niederlande");
}
```

Der loop-Teil:

```
void loop()
{
    // Taste 1 gedrückt
    if (Land1.update())
    {
        if (Land1.read() == LOW)
        {
            ZeigeDaten(Deutschland);
        }
    }

    // Taste 2 gedrückt
    if (Land2.update())
    {
        if (Land2.read() == LOW)
        {
            ZeigeDaten(Frankreich);
        }
    }

    // Taste 3 gedrückt
    if (Land3.update())
    {
        if (Land3.read() == LOW)
        {
            ZeigeDaten(Belgien);
        }
    }

    // Taste 4 gedrückt
    if (Land4.update())
    {
        if (Land4.read() == LOW)
        {
            ZeigeDaten(Niederlande);
        }
    }
}
```

Der Funktion ZeigeDaten wird die Struktur Laenderdaten übergeben, die Variable Land enthält das Element, das im Aufruf der Methode ZeigeDaten übergeben wurde.

```
void ZeigeDaten(struct Laenderdaten Land)
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(Land.Name);

    // bei der Einwohnerzahl . durch , ersetzen
    String Einwohner = String(Land.Einwohner);
    Einwohner.replace(".", ",");
    lcd.setCursor(0, 1);

    lcd.print(Einwohner + " Millionen");
}
```

```
lcd.setCursor(0, 2);

lcd.print("Fläche " + String(Land.Groesse) + " qkm");
lcd.setCursor(0, 3);

lcd.print("Hauptstadt ");
lcd.print(Land.Hauptstadt);
lcd.setCursor(0, 0);
}
```

Hartmut Waller (hartmut-waller.info/arduino-blog) Letzte Änderung: 10.05.24