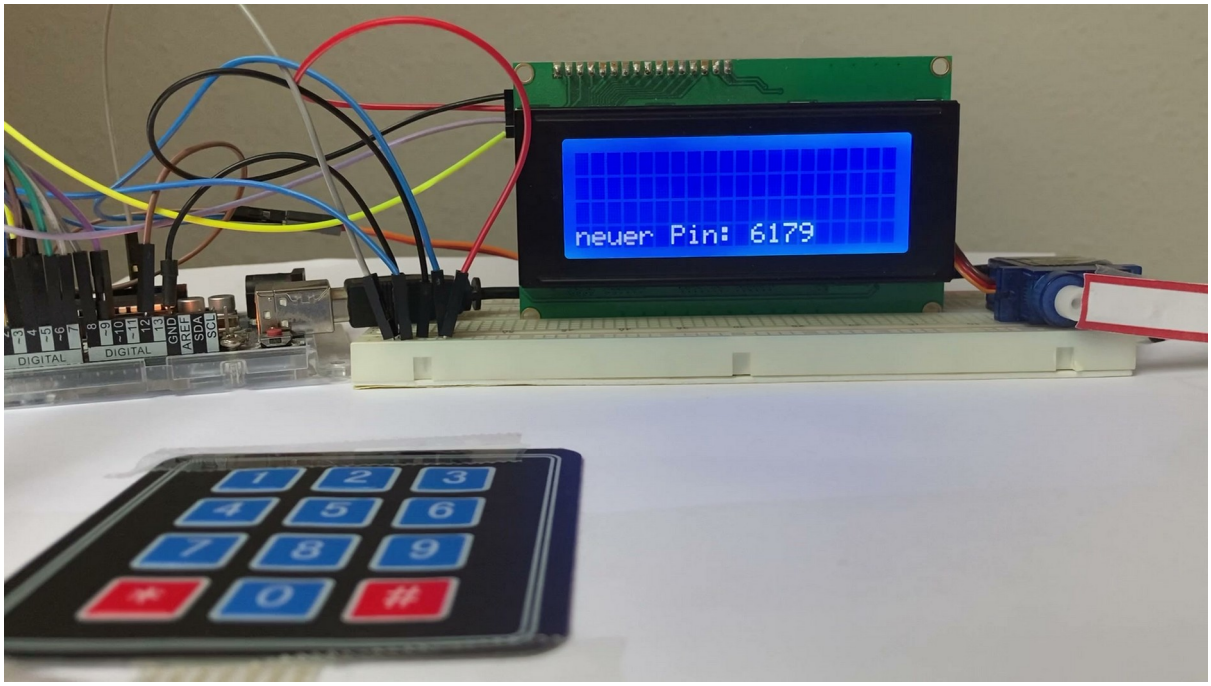
Auf einem Tastenfeld soll ein zufällig erzeugter Pin eingegeben werden. Das Programm soll diesen Pin abfragen.

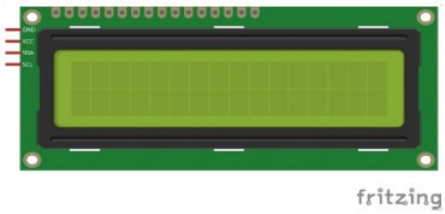
Wurde der Pin nach Druck auf die Taste # richtig eingegeben, wird auf dem LCD und im Seriellen Monitor die Meldung „korrekter Pin“ angezeigt und der Servomotor öffnet die Schranke.

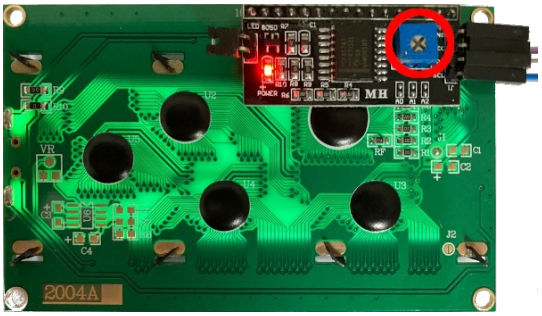


Benötigte Bauteile:

- LCD 1602
- Servomotor
- Tastenfeld 3×4
- Leitungsdrähte

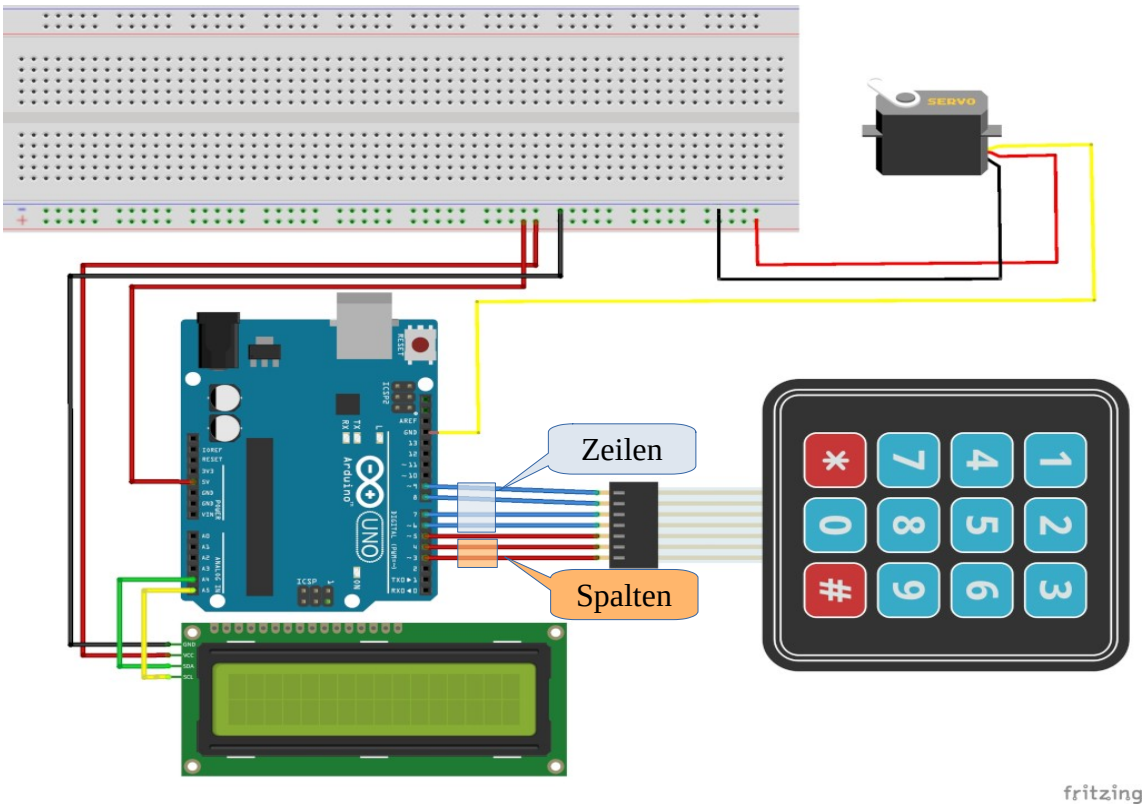
Das LCD wird an A4 und A4 oder an SDA und SCL angeschlossen.



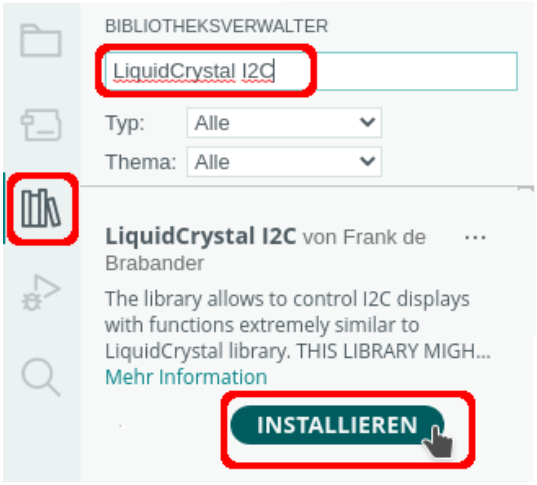


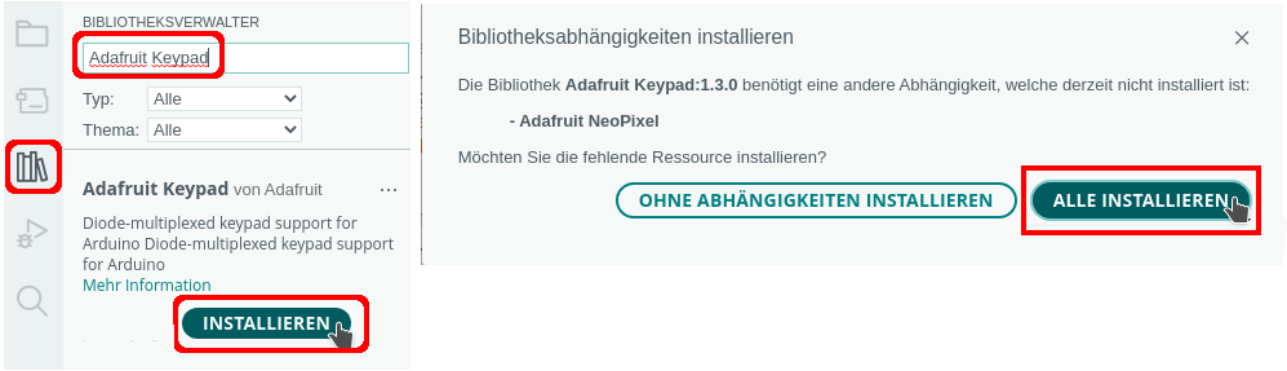
Auf der Rückseite befindet sich ein Drehpotentiometer, mit dem die Helligkeit des LCDs eingestellt wird.

Baue die Schaltung auf.



Benötigte Bibliotheken:



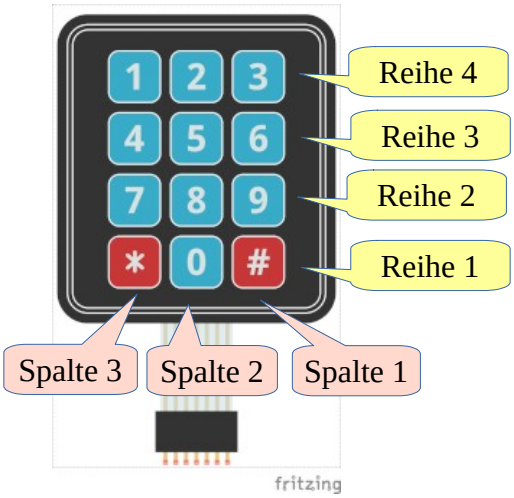


Das Tastenfeld besteht aus Reihen und Spalten, die von unten nach oben in einem **Array** angeordnet werden:

```
// Array 3 x 4
char Tasten[REIHEN][SPALTEN] =
{
   {'#', '0', '*'},
   {'9', '8', '7'},
   {'6', '5', '4'},
   {'3', '2', '1'}
};
```

Spalte 1 Spalte 2 Spalte 3

Reihe 1 Reihe 2 Reihe 3 Reihe 4



Binde die Bibliotheken ein und definiere die Variablen:

```
# include <Adafruit_Keypad.h>
# include <Servo.h>
# include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 20, 4);

// Bezeichnung des Motors
Servo Motor;

// Größe des Tastenfeldes
// 3 Spalten
const byte SPALTEN = 3;

// 4 Zeilen
const byte REIHEN = 4;
```

```
// die Ziffern/Zeichen:
// Array 3 x 4
char Tasten[REIHEN][SPALTEN] =
{
  { '#', '0', '*' },
  { '9', '8', '7' },
  { '6', '5', '4' },
  { '3', '2', '1' }
};

// die Pins für die 3 Spalten
byte SpaltenPins[SPALTEN] = { 3, 4, 5 };

// die Pins für die 4 Zeilen
byte ReihenPins[REIHEN] = { 6, 7, 8, 9 };

// TastenFeld → Name des Keypads
// -> Zuordnung der Pins zu den REIHEN und SPALTEN des Arrays
Adafruit_Keypad Tastenfeld = Adafruit_Keypad(makeKeymap(Tasten), ReihenPins,
SpaltenPins, REIHEN, SPALTEN);

String Vergleich;
String Pin;

// char-Array für die eingegebenen Zeichen
char Zeichen[5];

// Position im Array Zeichen
int Position = 0;

// Position der angezeigten Spalte LCD
int PositionSpalte = 0;

// Pin des Servos
int ServoPin = 12;
```

Der setup-Teil. Beachte die Kommentare:

```
void setup()
{
  // Pin dem Servo zuordnen
  Motor.attach(ServoPin);

  // Tastenfeld starten
  Tastenfeld.begin();

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

  Serial.begin(9600);

  // auf serielle Verbindung warten
  while (!Serial) { ; }
  delay(500);
```

```
  // nur beim Start anzeigen
  Serial.println("Bitte 4-stelligen Pin eingeben:");
  Serial.println("Pin-Eingabe mit # abschlie\u00dfen");
  Serial.println("Korrektur mit * ");

  // Zufallsgenerator starten
  randomSeed(analogRead(0));

  // neuen Pin erstellen
  neuerPin();

  // Servo in Ausgangsposition fahren
  Motor.write(0);
}
```

Der loop-Teil. Beachte die Kommentare.

```
void loop()
{
  // gedrückte Taste lesen
  Tastenfeld.tick();

  while (Tastenfeld.available())
  {
    keypadEvent Taste = Tastenfeld.read();

    // wenn die Taste losgelassen wurde
    // Wert abgefragter Taste zu char umwandeln
    if (Taste.bit.EVENT == KEY_JUST_RELEASED)
    {
      lcd.setCursor(0, 0);
      lcd.print("Pin eingeben:      ");

      // Zeilen LCD löschen
      ZeileLoeschen(1);
      ZeileLoeschen(3);

      Motor.write(0);

      // eingegebenes Zeichen an der aktuellen Position speichern
      Zeichen[Position] = (char)Taste.bit.KEY;

      // Zeichen anzeigen
      Serial.print(Zeichen[Position]);

      lcd.setCursor(PositionSpalte, 2);
      lcd.print(Zeichen[Position]);

      // nächstes Zeichen -> Position erhöhen
      // Position Spalte LCD erhöhen
      Position++;
      PositionSpalte++;
```

```
// Korrektur ASCII-Wert 42 = *
if (Taste.bit.KEY == 42)
{
  // char-Array Zeichen leeren
  for (int i = 0; i < sizeof(Zeichen); i++)
  {
    Zeichen[i] = NULL;
  }

  // Variablen zurücksetzen, Zeilen LCD löschen
  Position = 0;
  PositionSpalte = 0;
  ZeileLoeschen(0);
  ZeileLoeschen(2);
  Serial.println("\tKorrektur: ");
  lcd.setCursor(0, 0);

  lcd.print("Korrektur: ");
}

// Zeichen ASCII-Wert 35 = #
if (Taste.bit.KEY == 35)
{
  // char-Array in String umwandeln
  Vergleich = String(Zeichen);

  // letzte Zeichen des Strings sind 0 und # -> müssen entfernt werden
  Vergleich = Vergleich.substring(0, Vergleich.length() - 2);

  Serial.println();
  Serial.print(Vergleich);

  // Zeilen LCD löschen
  ZeileLoeschen(0);
  ZeileLoeschen(2);

  lcd.setCursor(0, 0);
  lcd.print("Pin: " + Vergleich + " ");

  // Eingabe mit Pin vergleichen
  if (Vergleich == Pin)
  {
    Serial.println("\korrekter Pin - Schranke \u00f6ffnen!");
    lcd.setCursor(0, 1);
    lcd.print("-> korrekter Pin!");
    delay(2000);
    lcd.setCursor(0, 1);
    lcd.print("Schranke \357ffnen!");
    Motor.write(90);
  }
```

```
        else
        {
          Serial.println("\tFalscher Pin - kein Zugriff");

          lcd.setCursor(0, 1);
          lcd.print("-> falscher Pin!");
        }

        // Variable zurücksetzen
        Position = 0;
        PositionSpalte = 0;

        // neuen Pin erstellen
        neuerPin();
      }
    }
  }
}
```

Die Funktion neuerPin() erzeugt einen neuen, zufälligen Pin und zeigt ihn an..

```
String neuerPin()
{
  // zufälligen PIN bestimmen
  Pin = "";
  Serial.println("----------------------------------------");

  for (int i = 0; i < sizeof(Zeichen) - 1; i++)
  {
    int Zahl = random(0, 10);
    Pin = Pin + String(Zahl);
  }

  // Pin anzeigen
  Serial.println("neuer Pin: " + Pin);
  lcd.setCursor(0, 3);
  lcd.print("neuer Pin: " + Pin);
  return Pin;
}
```

Die Methode ZeileLoeschen() löscht einzelne Zeilen auf dem LCD.

```
void ZeileLoeschen(int Zeile)
{
  for (int i = 0; i < 20; i ++)
  {
    lcd.setCursor(i, Zeile);

    lcd.print(" ");
  }
}
```