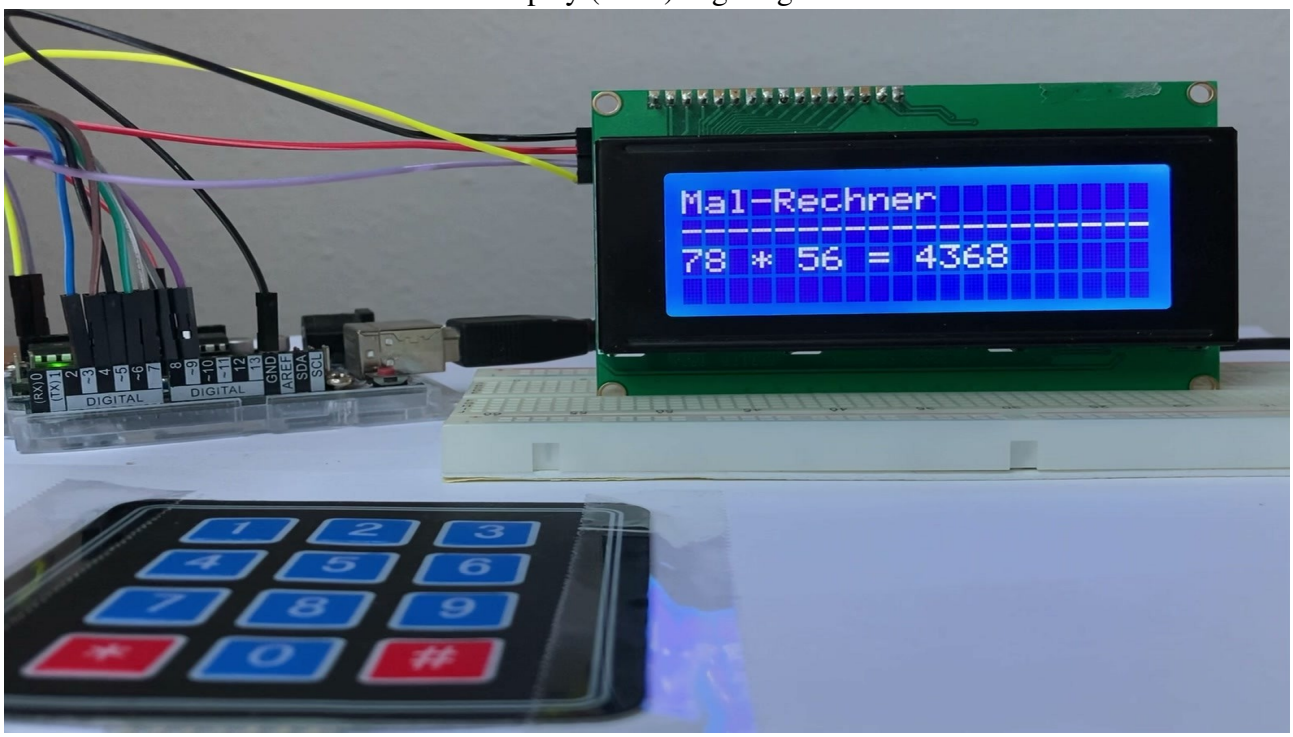
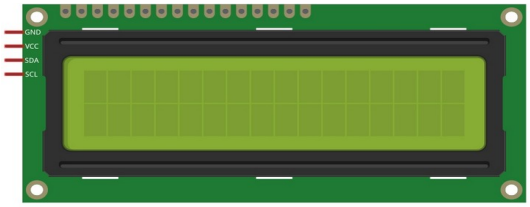
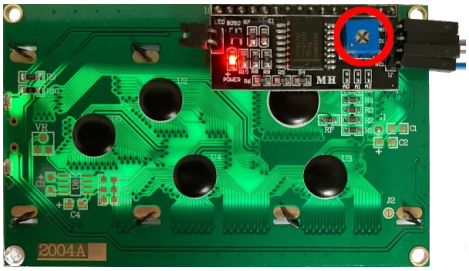
Das Tastenfeld soll dazu verwendet werden um Multiplikationen durchzuführen und das Ergebnis soll im Seriellen Monitor und im LC-Display (LCD) angezeigt werden.



Schließe das LCD an:



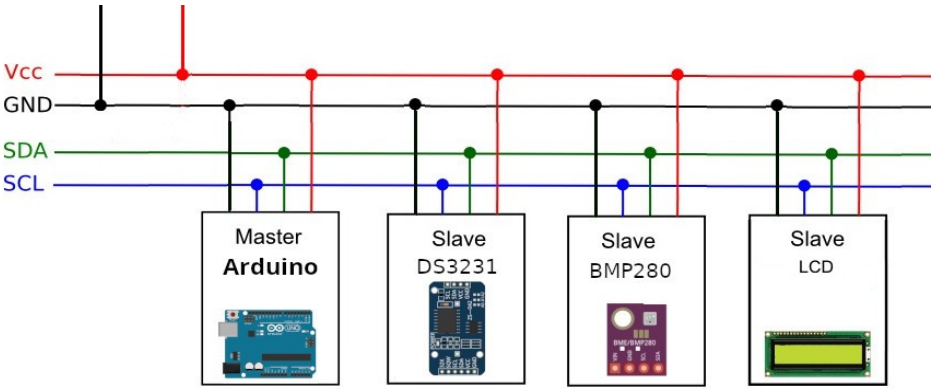
fritzing



Auf der Rückseite befindet sich ein Potentiometer mit dem die Helligkeit eingestellt werden kann.

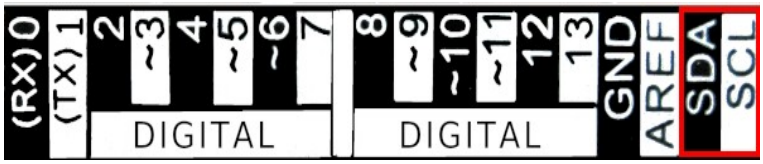
Normalerweise wäre eine komplexe Verkabelung zum Betrieb eines LCDs nötig. Der I²C-Bus regelt über einen eigenen Mikroprozessor die Kommunikation der Datenleitungen untereinander.
Es werden deshalb nur vier Anschlüsse benötigt.

Der I²C-Bus (Inter Integrated Circuit) wurde ursprünglich von Philips entwickelt, er sollte die Kommunikation mit einem Master (dem Arduino) und den verschiedenen Bauelementen (den Slaves) ermöglichen.



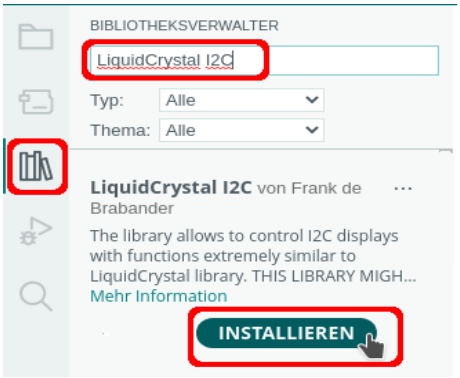
Der I²C-Bus kommt mit zwei Datenleitungen aus:

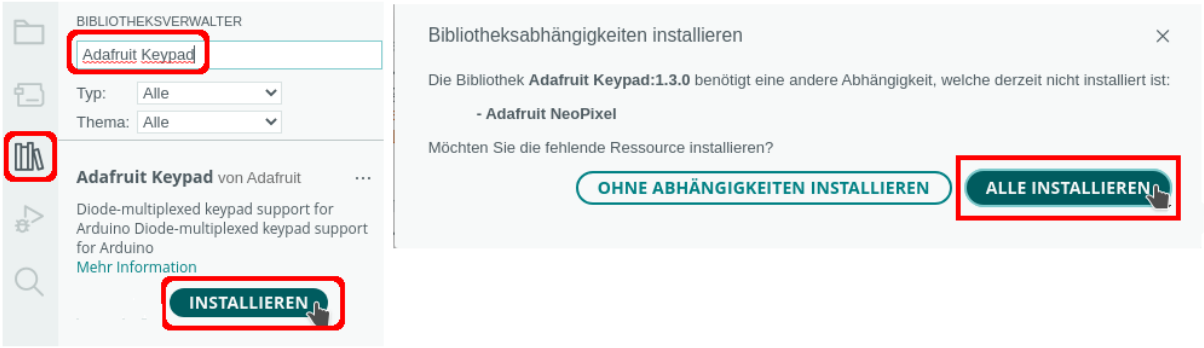
- die Taktleitung SCL (Serial Clock) → A5
- die Datenleitung SDA (Serial Data) → A4



Statt A4 (SDA) und A5 (SCL) kannst du auch die mit SCL und SDA beschrifteten Pins verwenden.

Benötigte Bibliotheken:

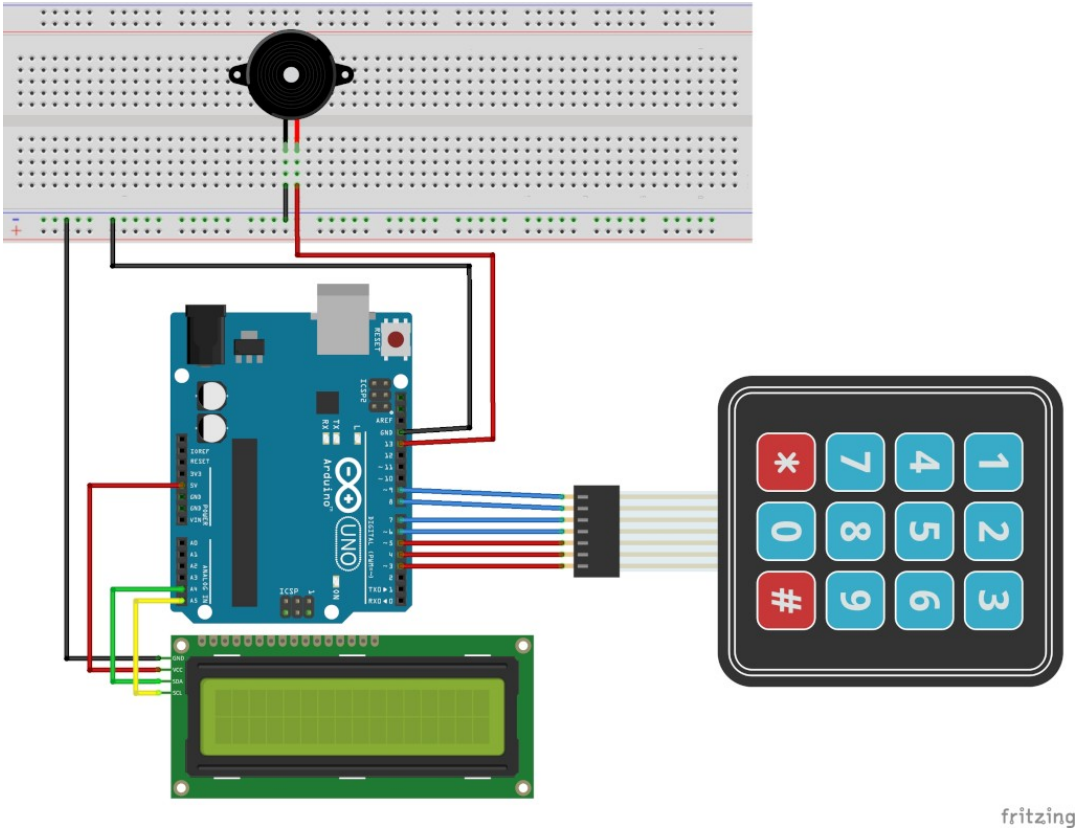




Benötigte Bauteile:

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

Baue die Schaltung auf.

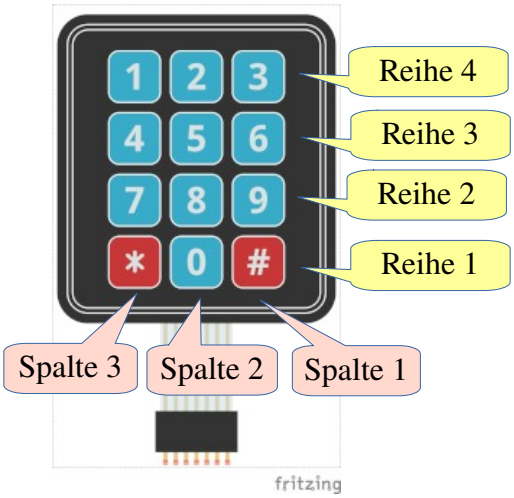


Das Tastenfeld besteht aus Reihen und Spalten, die von unten nach oben in einem **Array** angeordnet werden:

```
// 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 benötigten Bibliotheken ein und definiere die Variablen. Beachte die Kommentare.

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

int LAUTSPRECHER = 13;

// 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);

// LCD initialisieren
LiquidCrystal_I2C lcd(0x27, 20, 4);

// setzt die Position der Spalte auf dem LCD
int PositionSpalte = 0;
```

```
// char-Array für die eingegebenen Zeichen
char Zeichen[50];

// Position des eingegebenen Zeichens im char-Array
int Position = 0;

// beinhaltet die vollständige Rechnung
String Rechnung;
```

Wenn dir die ID des verwendeten Displays nicht bekannt ist, kannst du sie mit folgendem **Programm** herausfinden.

Der setup-Teil. Beachte die Kommentare.

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

  // Tastenfeld starten
  Tastenfeld.begin();

  // LCD einschalten, Startbildschirm zeigen
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Mal-Rechner");
  lcd.setCursor(0, 1);
  lcd.print("--------------------");
  lcd.setCursor(0, 2);
}
```

Der loop-Teil: Beachte die Kommentare.

```
void loop()
{
  String GleichZeichen = " = ";
  lcd.setCursor(Position, 2);

  // Tastenfeld abfragen
  Tastenfeld.tick();

  while (Tastenfeld.available())
  {
    /*
      wenn eine Aufgabe ausgegeben wurde
      LCD löschen, Anzeige neu aufbauen
    */
    if (Rechnung != "")
    {
      Rechnung = "";
      lcd.clear();
      Position = 0;
      lcd.setCursor(0, 0);
      lcd.print("Mal-Rechner");
```

```
    lcd.setCursor(0, 1);
    lcd.print("--------------------");
    lcd.setCursor(0, 2);

    // char-Array Zeichen leeren
    for (int i = 0; i < sizeof(Zeichen); i ++)
    {
      Zeichen[i] = NULL;
    }

    Position = 0;
    PositionSpalte = 0;
  }

  // gedrückte Taste lesen
  keypadEvent Taste = Tastenfeld.read();

  // wenn die Taste losgelassen wurde
  // Wert abgefragter Taste zu char umwandeln
  if (Taste.bit.EVENT == KEY_JUST_RELEASED)
  {
    // eingegebenes Zeichen an der aktuellen Position speichern
    Zeichen[Position] = (char)Taste.bit.KEY;

    // Ton ausgeben
    tone(LAUTSPRECHER, 1000);
    delay(50);
    noTone(LAUTSPRECHER);

    // Zeichen anzeigen
    // # soll nicht angezeigt werden
    if (Taste.bit.KEY != 35)
    {
      Serial.print(Zeichen[Position]);
      lcd.print(Zeichen[Position]);
    }

    // nächstes Zeichen -> Position erhöhen
    Position ++;

    // Ausgabe LCD eine Position weiter
    PositionSpalte ++;

    // # (ASCII-Code = 35) gedrückt → Rechnung ausführen
    if (Taste.bit.KEY == 35)
    {
      // char-Array in String umwandeln
      Rechnung = String(Zeichen);

      // letztes Zeichen ist # → muss entfernt werden
      Rechnung = Rechnung.substring(0, Position - 1);

      // Position des Malzeichens bestimmen
      // Zählung beginnt mit 0
      int MalZeichen = Rechnung.indexOf('*');
```

```
        // erster Faktor: von 0 bis zur Position des Malzeichens
        String FaktorEins = Rechnung.substring(0, MalZeichen);

        /*
          zweiter Faktor: von der Position hinter dem Malzeichen
          bis zum vorletzten Zeichen
          das letzte Zeichen ist # → soll entfernt werden
        */
        String FaktorZwei = Rechnung.substring(MalZeichen + 1, Rechnung.length());

        /*
           Produkt berechnen -> die beiden Faktoren dürfen nicht größer als 32768 sein
           long Produkt = FaktorEins.toInt() * FaktorZwei.toInt();
           Zahlen nach long umwandeln
        */
        long ErsterFaktor = FaktorEins.toInt();
        long ZweiterFaktor = FaktorZwei.toInt();
        long Produkt =  ErsterFaktor * ZweiterFaktor;

        // String Rechnung "zusammenbauen"
        Rechnung = FaktorEins + " * " + FaktorZwei + GleichZeichen + String(Produkt);

        // Rechnung anzeigen
        RechnungAusgeben();
      }
    }
  }
}
```

Ergänze das Programm mit der Methode **RechnungAusgeben().**

```
void RechnungAusgeben()
{
  // Ausgabe Serieller Monitor
  Serial.println("\n" + Rechnung);

  // Ausgabe LCD
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.noCursor();
  lcd.print("Mal-Rechner");
  lcd.setCursor(0, 1);

  lcd.print("--------------------");
  lcd.setCursor(0, 2);

  /* Länge der Rechnung bestimmen
     das LCD hat nur 20 Zeichen
     wenn es mehr als 20 sind → Zeilenumbruch beim =-Zeichen
  */
  int Laenge = Rechnung.length();
```

```
  if (Laenge > 20)
  {
    // Position des =-Zeichen bestimmen
    int GleichZeichen = Rechnung.indexOf('=');

    // Rechnung bis =-Zeichen anzeigen
    lcd.print(Rechnung.substring(0, GleichZeichen + 1));

    // Zeilenumbruch
    lcd.setCursor(0, 3);
    lcd.print(Rechnung.substring(GleichZeichen + 2, Rechnung.length()));
  }

  // wenn Laenge < 20 Zeichen
  else lcd.print(Rechnung);
}
```