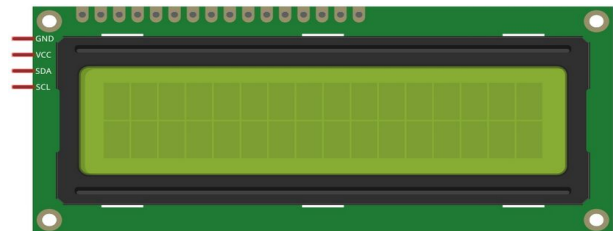


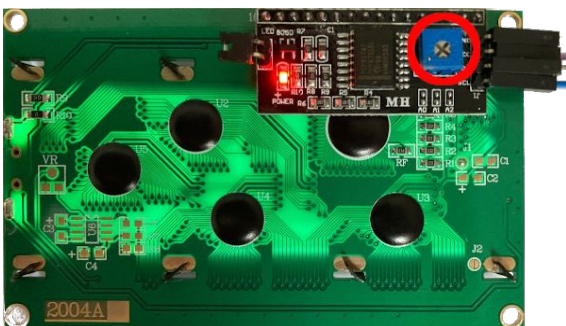
Multiplikations-Taschenrechner

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:




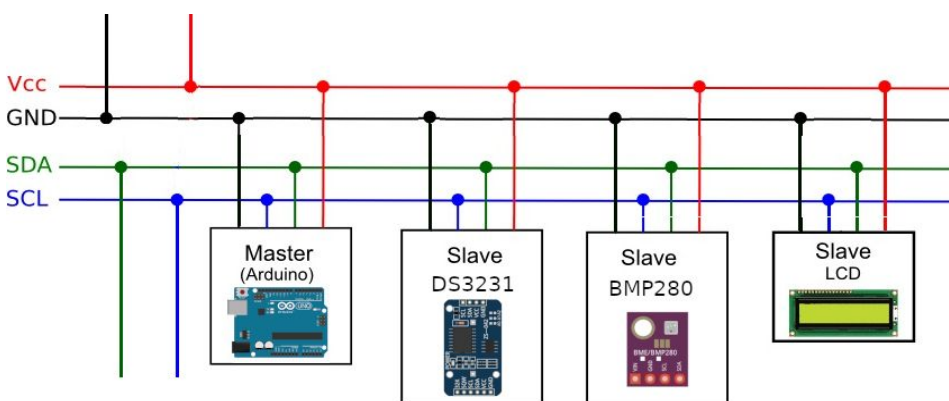
fritzing



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.

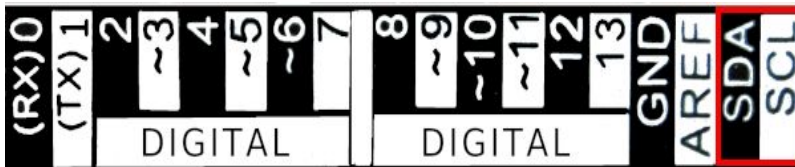
 Der I²C-Bus (Inter Integrated Circuit) wurde ursprünglich von Philips entwickelt, er sollte die Kommunikation mit einem Master (dem Arduino) und den verschiedenen Bauelementen (den Slaves) ermöglichen.



Quelle: <http://prometec.org/displays/the-i2c-bus> (eigene Bearbeitung)

Der I²C-Bus kommt mit zwei Datenleitungen aus:

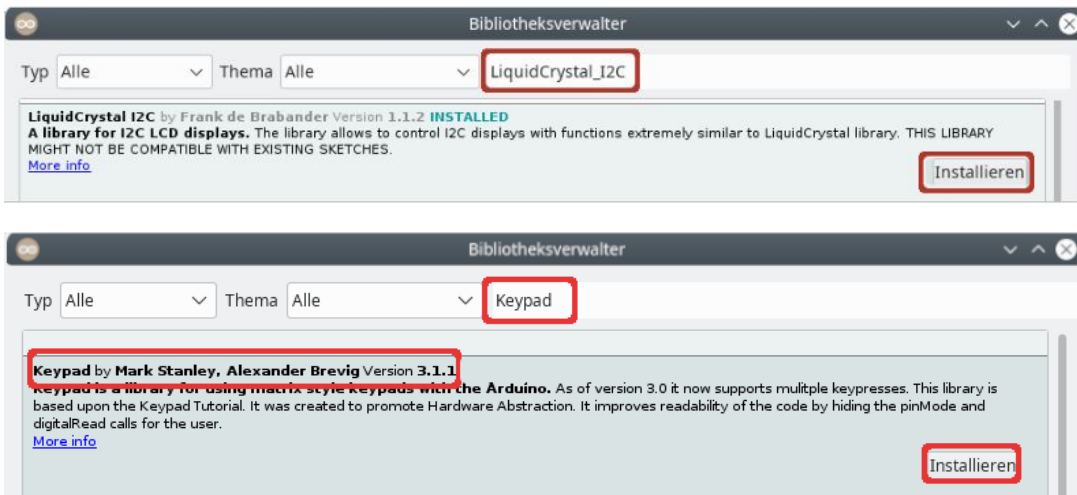
- die Taktleitung SCL (Serial Clock) → A5
- die Datenleitung SDA (Serial Data) → A4



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

Benötigte Bibliotheken:

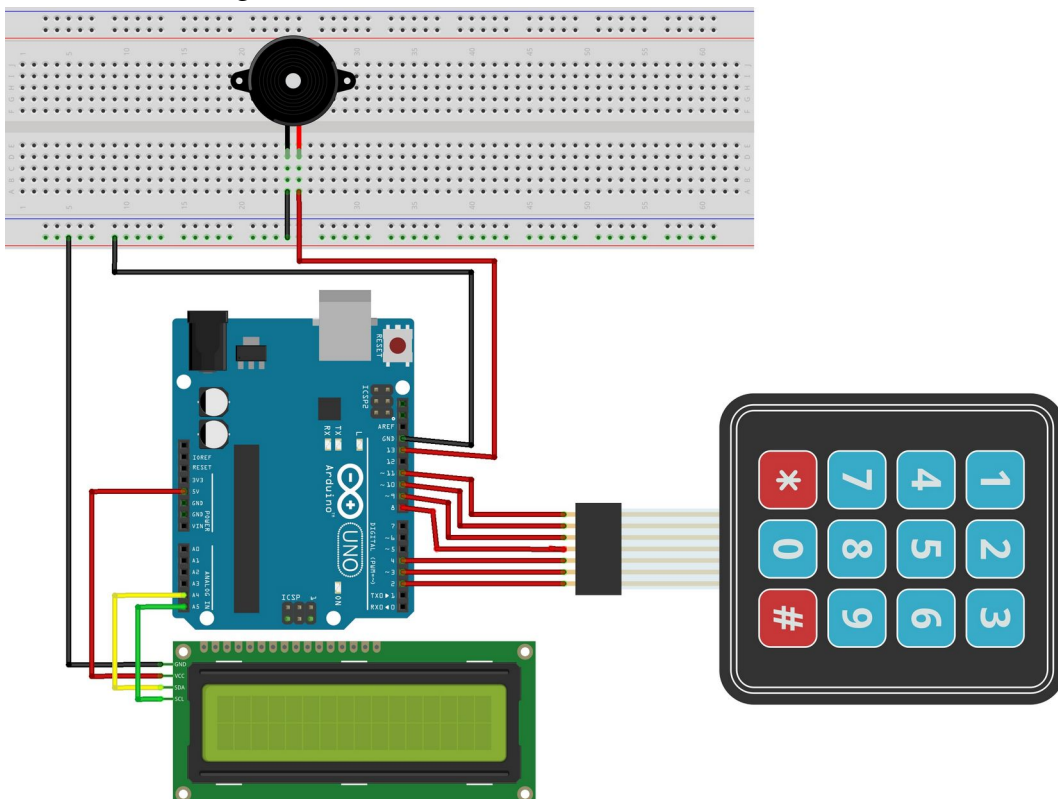
Sketch → Bibliothek einbinden → Bibliotheken verwalten



Benötigte Bauteile:

- ➔ LCD 1602
- ➔ Lautsprecher
- ➔ Tastenpad 3×4
- ➔ Leitungsdrähte

Baue die Schaltung auf.



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

# include <Keypad.h>
/*
  Größe des Tastenfeldes
  Arrays sind statisch,, deshalb müssen SPALTEN/REIHEN als statisch (const)
  definiert werden
  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] = {2, 3, 4};

// die Pins für die 4 Zeilen
byte ReihenPins[REIHEN] = {8, 9, 10, 11};
char gedruckteTaste;

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

0x27 ist die ID des I²C-Busses, 20 bestimmt die Anzahl der Zeichen pro Zeile, 4 nennt die Anzahl der Zeilen.

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

Jetzt kann die eigentliche Programmierung beginnen.

Lege zusätzliche Variable fest:

```
// setzt die Position der Spalte
int Position = 0;

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

```
// temporäre Speicherung der gedrückten Tasten
String Eingabe;
char gedruckteTaste;
int LAUTSPRECHER = 13;

void setup()
{
    Serial.begin(9600);
    // LCD einschalten
    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);

    // gedrückte Taste abfragen
    gedruckteTaste = TastenFeld.getKey();

    // wenn eine Taste gedrückt wurde
    if (gedruckteTaste)
    {
        tone(LAUTSPRECHER, 1000);
        delay(50);
        noTone(LAUTSPRECHER);

        /* wenn eine Aufgabe gerechnet 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);
        }

        // Ausgabe eine Position weiter
        Position ++;
    }
}

```

```
// # soll nicht angezeigt werden
if (gedrueckteTaste != '#') lcd.print(gedrueckteTaste);

// gedrueckteTaste dem String Eingabe hinzufügen
Eingabe = Eingabe + gedrueckteTaste;
// oder Kurzform:
// Eingabe += gedrueckteTaste;

// # gedrückt → Rechnung ausführen
if (gedrueckteTaste == '#')
{
    // letztes Zeichen ist # → muss entfernt werden
    Rechnung = Eingabe.substring(0, Eingabe.length() - 1);

    // Position des Malzeichens bestimmen
    // Zählung beginnt mit 0
    int MalZeichen = Eingabe.indexOf('*');
    // erster Faktor: von 0 bis zur Position des Malzeichens
    String FaktorEins = Eingabe.substring(0, MalZeichen);

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

    /* Produkt berechnen
       damit dürfen die beiden Faktoren 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();
    Eingabe = "";
}
}
```

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

```
void RechnungAusgeben()
{
    // Ausgabe Serieller Monitor
    Serial.println(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();

    // wenn die Länge der Rechnung > 20
    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 < 20 Zeichen ...
    else lcd.print(Rechnung);
}
```