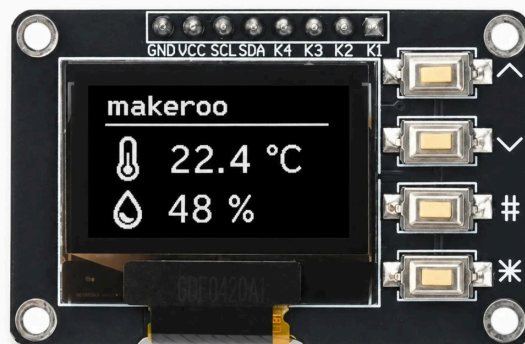


BEDIENPANEL • I2C

OLED mit 4 Tastern

0,96-Zoll-OLED und vier Taster auf einer Platine – ein fertiges Bedienpanel. Erstes Menü in rund 25 Minuten mit UNO R3.



AI Modified

AUFLÖSUNG

128 × 64 Pixel

CONTROLLER

SSD1315

BEDIENUNG

4 Taster

SCHNITTSTELLE

I2C 0x3C

01

Verkabeln

8 Leitungen zum UNO

02

Einrichten

IDE, Treiber, Bibliothek

03

Hochladen

Beispiel-Sketch

04

Bedienen

Menü mit 4 Tasten

01 Auf einen Blick

Display und Bedienung in einem Bauteil: Rechts neben dem OLED sitzen vier Taster mit $\wedge$, $\vee$, $\#$ und $*$ – hoch, runter, bestätigen, zurück. Vier Bohrungen machen das Modul zur fertigen Frontplatte.

Technische Daten

Display	OLED, weiß, 0,96 Zoll
Auflösung	128 × 64 Pixel
Controller	SSD1315 (SSD1306-kompatibel)
Schnittstelle	I2C, Adresse 0x3C
Taster	4 Stück, K1 bis K4, aktiv LOW
Versorgung	3,3 bis 5 V DC
Anschluss	8 Pins, Stiftleiste liegt bei

Lieferumfang

1 ×	0,96" OLED-Modul mit 4 Tastern (SSD1315)
1 ×	Stiftleiste (zum Selbstlöten)



SSD1315 = SSD1306

Der SSD1315 ist befehlskompatibel zum SSD1306. Die bekannte Bibliothek Adafruit SSD1306 läuft unverändert.



Taster ohne Widerstände

Die Taster schalten nach GND. Der Sketch aktiviert die internen Pull-ups des UNO – keine Zusatzbauteile nötig.

Pinbelegung

PIN	FUNKTION	SPANNUNG	HINWEIS
● GND	Masse	0 V	Auch Bezug für die Taster.
● VCC	Versorgung	3,3–5 V DC	Am UNO an 5V.
● SCL	I2C Takt	3,3 / 5 V	Am UNO an A5.
● SDA	I2C Daten	3,3 / 5 V	Am UNO an A4.
● K4	Taster *	aktiv LOW	Hier: D5, „zurück“.
● K3	Taster #	aktiv LOW	Hier: D4, „öffnen“.
● K2	Taster $\vee$	aktiv LOW	Hier: D3, „runter“.
● K1	Taster $\wedge$	aktiv LOW	Hier: D2, „hoch“.

Reihenfolge auf der Platine: GND · VCC · SCL · SDA · K4 · K3 · K2 · K1. Die Zuordnung der Taster stammt aus unseren Produktdaten.

Was du zusätzlich brauchst

UNO R3 mit USB-C

ATmega328P, CH340G-USB-Wandler
MKR-UNO-R3-013

Breadboard + Jumper-Set

400 Kontakte, 40 Dupont-Kabel
MKR-SET-BB400-JUMPER

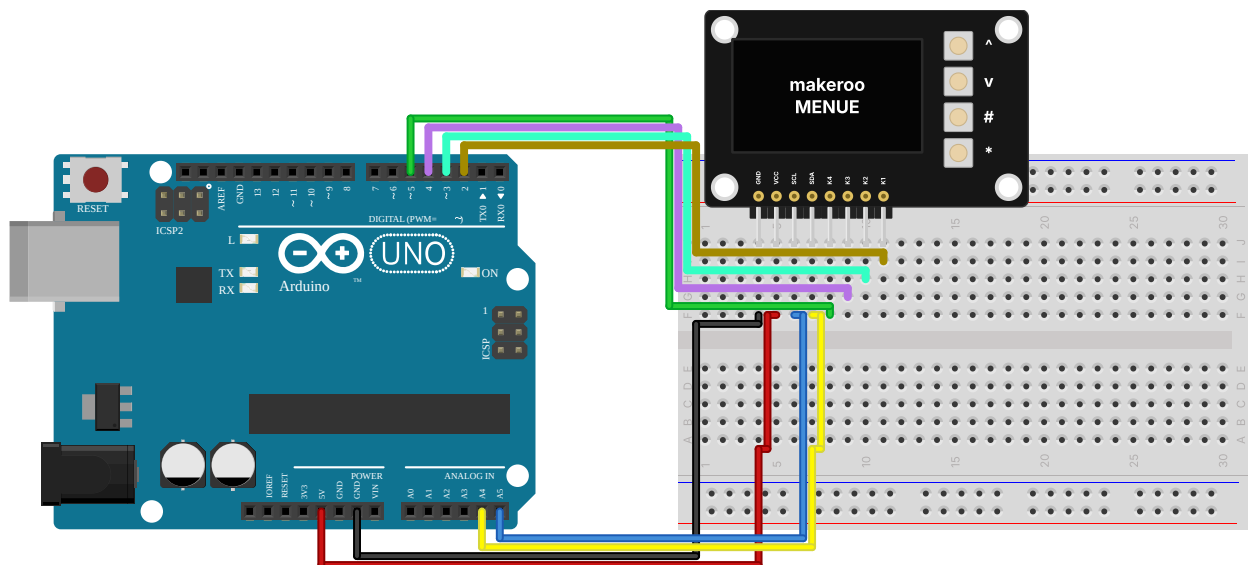
Lötkolben

Stiftleiste einlöten (z. B. TOOLCRAFT LS-30E)
MKR-LS30E-70

02 Verkabeln

Acht Leitungen: vier für das Display, je eine pro Taster.

STECKPLATTINEN-ANSICHT



Verbindungen

DISPLAY		UNO R3	LEITUNG
● GND	→	GND	Masse
● VCC	→	5V	Versorgung
● SCL	→	A5	I2C Takt
● SDA	→	A4	I2C Daten
● K4	→	D5	* zurück
● K3	→	D4	# öffnen
● K2	→	D3	v runter
● K1	→	D2	^ hoch



Stiftleiste einlöten

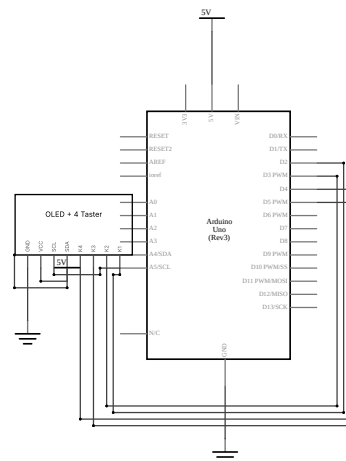
Leiste ins Steckbrett, Platine auflegen, acht Pins verlöten.



Taster vertauscht?

Reagiert „hoch“ wie „runter“? Dann die Pin-Nummern im Sketch tauschen.

SCHALTPLAN



Verdrahtung erstellt mit Fritzing

03 Einrichten & Hochladen

Schritte 1 bis 3 machst du nur ein einziges Mal – danach ist dein UNO R3 für alle weiteren Projekte startklar.

1 Arduino IDE installieren

Lade die aktuelle Arduino IDE 2 von arduino.cc/en/software herunter und installiere sie.

2 Treiber für den CH340G prüfen

Der UNO R3 nutzt den CH340G als USB-Seriell-Wandler. Aktuelle Systeme erkennen ihn meist automatisch. Taucht in Schritt 3 kein Port auf, installiere den Treiber „CH341SER“ des Chipherstellers WCH (wch-ic.com).

3 Board und Port wählen

UNO per USB-C anschließen. Dann **Werkzeuge** → **Board** → **Arduino AVR Boards** → **Arduino Uno** wählen und unter **Werkzeuge** → **Port** den neu aufgetauchten COM-Port (Windows) bzw. `/dev/ttyUSB0` (Linux) / `/dev/cu.wchusbserial...` (macOS).

4 Bibliothek installieren

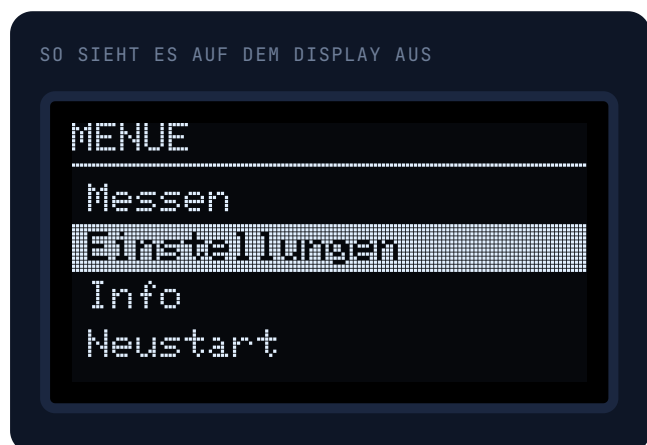
Bibliotheksverwalter öffnen (`Strg` + `Umschalt` + `I`), nach **Adafruit SSD1306** von Adafruit suchen und installieren. Abhängigkeiten (u. a. *Adafruit GFX Library*, *Adafruit BusIO*) mit **Alle installieren** bestätigen.

5 Sketch hochladen

Neuen Sketch anlegen, den Code von Seite 5 einfügen und mit dem Pfeil-Symbol hochladen (`Strg` + `U`). Unten erscheint „Hochladen abgeschlossen“.

6 Seriellen Monitor öffnen (optional)

`Strg` + `Umschalt` + `M`, unten rechts **9600 Baud**. Normalerweise bleibt er leer – nur bei einem Fehler erscheint „Display nicht gefunden!“.



i Schnelltest

Mit `↶` und `↷` wandert die Markierung durch das Menü, `#` öffnet den Eintrag, `*` geht zurück.

Nachbildung der Anzeige – berechnet aus dem Sketch auf Seite 5, Pixel für Pixel.

04

Der erste Sketch

Ein vollständiges kleines Menü: markieren, öffnen, zurück – die Grundlage für jedes Bedienpanel.

```

MAKER00_OLEDKEY_Start.ino

1 // MAKER00 · 0,96" OLED SSD1315 mit 4 Tastern · Erstes Menue
2 #include <Wire.h>
3 #include <Adafruit_GFX.h>
4 #include <Adafruit_SSD1306.h>           // passt auch fuer den SSD1315
5
6 const int K1 = 2, K2 = 3, K3 = 4, K4 = 5; // hoch, runter, #, *
7 Adafruit_SSD1306 oled(128, 64, &Wire, -1);
8 const char* const PUNKT[] = {"Messen", "Einstellungen", "Info", "Neustart"};
9 int auswahl = 0;                       // markierter Eintrag
10 int offen = -1;                        // -1 = Menue wird gezeigt
11
12 bool gedrueckt(int pin) {               // wahr nach Druck + Loslassen
13     if (digitalRead(pin) == HIGH) return false;
14     delay(20);                          // entprellen
15     while (digitalRead(pin) == LOW) {}
16     return true;
17 }
18
19 void setup() {
20     Serial.begin(9600);
21     for (int pin = K1; pin <= K4; pin++) pinMode(pin, INPUT_PULLUP);
22     if (!oled.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
23         Serial.println(F("Display nicht gefunden!"));
24         while (true) {}
25     }
26 }
27
28 void loop() {
29     if (gedrueckt(K1)) auswahl = (auswahl + 3) % 4; // hoch
30     if (gedrueckt(K2)) auswahl = (auswahl + 1) % 4; // runter
31     if (gedrueckt(K3)) offen = auswahl;           // # = oeffnen
32     if (gedrueckt(K4)) offen = -1;                // * = zurueck
33     oled.clearDisplay();
34     oled.setTextSize(1);
35     oled.setTextColor(SSD1306_WHITE);
36     oled.setCursor(0, 0);
37     oled.print(offen < 0 ? F("MENUE") : F("< zurueck mit *"));
38     oled.drawLine(0, 10, 127, 10, SSD1306_WHITE);
39     if (offen < 0) {
40         for (int i = 0; i < 4; i++) {
41             int y = 15 + i * 12;
42             if (i == auswahl) { // Auswahl invertieren
43                 oled.fillRect(0, y - 2, 128, 12, SSD1306_WHITE);
44                 oled.setTextColor(SSD1306_BLACK);
45             } else {
46                 oled.setTextColor(SSD1306_WHITE);
47             }
48             oled.setCursor(4, y);
49             oled.print(PUNKT[i]);
50         }
51     } else {
52         oled.setTextSize(2);
53         oled.setCursor(0, 24);
54         oled.print(PUNKT[offen]);
55     }
56     oled.display();
57 }

```

Zeile 6

Taster-Pins

K1 bis K4 hängen an D2 bis D5.

Zeile 12-17

Tastendruck

Wartet aufs Loslassen und entprellt – ein Druck zählt genau einmal.

Zeile 21

Pull-ups

Interne Widerstände halten die Pins auf HIGH, gedrückt sind sie LOW.

Zeile 29-32

Navigation

Hoch und runter laufen im Kreis; # öffnet, * schließt.

Zeile 42-44

Markierung

Weißer Balken mit schwarzer Schrift – invertierter Text.

Zeile 56

Anzeigen

Erst jetzt erscheint das fertige Bild.

Geprüft: kompiliert für Arduino Uno mit Adafruit SSD1306 2.5.17 (14.6 kB Flash = 46 %, 569 Byte RAM).

05

Wenn es nicht klappt

SYMPTOM	LÖSUNG
„Display nicht gefunden!“	SDA an A4 und SCL an A5? Nicht vertauscht? Mit dem I2C-Scanner (Beispiel im ESP32-Guide, läuft auch am UNO) prüfen, ob 0x3C oder 0x3D antwortet, und die Adresse im Sketch anpassen.
Tasten reagieren nicht	K1-K4 an D2-D5? GND verbunden? Ohne INPUT_PULLUP schweben die Eingänge.
Ein Druck springt mehrere Einträge	Entprellzeit erhöhen: <code>delay(20)</code> auf 40 setzen.
Bild verschoben	Nicht den SH1106-Treiber verwenden – dieses Modul braucht SSD1306/SSD1315.

06

Wie geht's weiter?

Mit ESP32 & ESPHome

VCC an **3V3**, SDA/SCL an **GPIO21/22**, K1-K4 an **GPIO32, 33, 25, 26**. In ESPHome:

```
i2c:
  sda: GPIO21
  scl: GPIO22
display:
  - platform: ssd1306_i2c
    model: "SSD1306 128x64"
    address: 0x3C
binary_sensor:
  - platform: gpio
    name: "Taste hoch"
    pin:
      number: GPIO32
      inverted: true
      mode: INPUT_PULLUP
```

Projektideen

Thermostat

Solltemperatur mit Δ v einstellen

Timer

Laufzeit wählen, mit # starten

Code-Schloss

Ziffernfolge über die Tasten eingeben

Geräte-Front

Als Deckel im 3D-Druck-Gehäuse

Passt dazu

Relais KY-019

Verbraucher schalten

[MKR-KY019-034](#)

DHT22 Temperatur & Feuchte

Klassiker für die Wetteranzeige

[MKR-DHT22-009](#)

ESP32 DevKit USB-C

WROOM-32, CH340C – 3,3-V-Logik

[MKR-ESP32-005](#)

BMP280 Luftdruck & Temperatur

Messwerte zum Anzeigen

[MKR-GY-BMP280-024](#)

Sicherheitshinweise. Elektronikmodul für Ausbildungs-, Entwicklungs- und Experimentierzwecke. Nur mit Kleinspannung betreiben, niemals an Netzspannung. Nicht geeignet für Kinder unter 14 Jahren; Nutzung nur unter Aufsicht. Elektroaltgeräte gehören nicht in den Hausmüll – Rückgabe kostenlos bei kommunalen Sammelstellen.

Verantwortlicher Wirtschaftsakteur (GPSR): Alexander Prez · makeroo · Hinterm Bahnhof 4 · 90513 Zirndorf · support@makeroo.de
Fragen? Schreib uns mit einem Foto deiner Verkabelung. Arduino ist eine Marke der Arduino SA und dient nur der Kompatibilitätsangabe.