

LCD Modul 84x48

Nokia 5110

Sehr geehrter Kunde,

vielen Dank, dass Sie sich für unser Produkt entschieden haben.

Im Folgenden haben wir aufgelistet, was bei der Inbetriebnahme zu beachten ist:

Verwendung mit einem Arduino

Schritt 1 – Anschließen des Displays

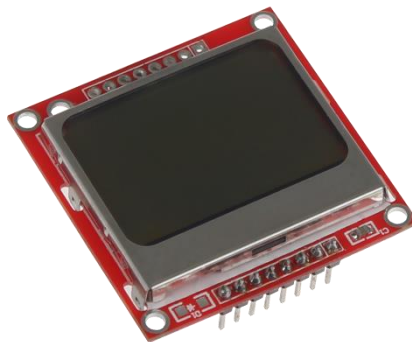


Bild 1: 84x48 Pixel LCD-Modul

Schließen Sie das Display, wie im folgenden Bild 2, bzw. in der folgenden Tabelle 1, zu sehen, an die PINs des Arduinos an.

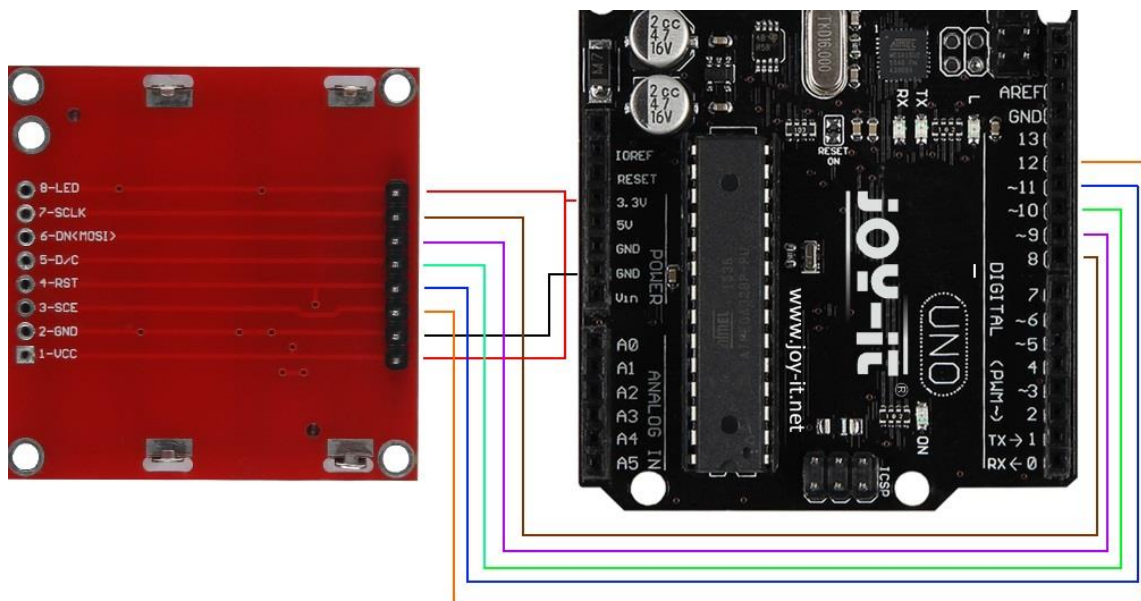


Bild 2: Verbindung zwischen Arduino und LCD-Display

LCD-Modul PIN	Arduino PIN
8 LED	3,3 V
7 SCLK	8
6 DN <MOSI>	9
5 D/C	10
4 RST	11
3 SCE	12
2 GND	GND
1 VCC	3,3V

Tabelle 1: PIN-Verbindung zwischen Arduino und LCD-Modul

Schritt 2 – Installation des LCD-Displays

Mit folgendem Codebeispiel, können Sie das LCD-Modul mit einem Arduino verwenden.
Bitte übertragen sie **alle** Code-Teile auf ihren Arduino.

In der **void loop()** können Sie mit Hilfe von **LCDString** einen von Ihnen gewünschten Text auf dem Display ausgeben.

```
//Zuweisung der PIN-Belegung
#define PIN_SCE 12
#define PIN_RESET 11
#define PIN_DC 10
#define PIN_SDIN 9
#define PIN_SCLK 8

#define LCD_COMMAND 0
#define LCD_DATA 1

//Einstellen der Bildschirmgröße (84x48 Pixel)
#define LCD_X 84
#define LCD_Y 48
```

Code 1: Teil 1 des LCD-Quellcodes

```
//Hexadezimalwerte zur Pixelzuweisung von Zeichen und Buchstaben
```

```
static const byte ASCII[][5] = {
  {0x00, 0x00, 0x00, 0x00, 0x00} // 20
  ,{0x00, 0x00, 0x5f, 0x00, 0x00} // 21 !
  ,{0x00, 0x07, 0x00, 0x07, 0x00} // 22 "
  ,{0x14, 0x7f, 0x14, 0x7f, 0x14} // 23 #
  ,{0x24, 0x2a, 0x7f, 0x2a, 0x12} // 24 $
  ,{0x23, 0x13, 0x08, 0x64, 0x62} // 25 %
  ,{0x36, 0x49, 0x55, 0x22, 0x50} // 26 &
  ,{0x00, 0x05, 0x03, 0x00, 0x00} // 27 '
  ,{0x00, 0x1c, 0x22, 0x41, 0x00} // 28 (
  ,{0x00, 0x41, 0x22, 0x1c, 0x00} // 29 )
  ,{0x14, 0x08, 0x3e, 0x08, 0x14} // 2a *
  ,{0x08, 0x08, 0x3e, 0x08, 0x08} // 2b +
  ,{0x00, 0x50, 0x30, 0x00, 0x00} // 2c ,
  ,{0x08, 0x08, 0x08, 0x08, 0x08} // 2d -
  ,{0x00, 0x60, 0x60, 0x00, 0x00} // 2e .
  ,{0x20, 0x10, 0x08, 0x04, 0x02} // 2f /
  ,{0x3e, 0x51, 0x49, 0x45, 0x3e} // 30 0
  ,{0x00, 0x42, 0x7f, 0x40, 0x00} // 31 1
  ,{0x42, 0x61, 0x51, 0x49, 0x46} // 32 2
  ,{0x21, 0x41, 0x45, 0x4b, 0x31} // 33 3
  ,{0x18, 0x14, 0x12, 0x7f, 0x10} // 34 4
  ,{0x27, 0x45, 0x45, 0x45, 0x39} // 35 5
  ,{0x3c, 0x4a, 0x49, 0x49, 0x30} // 36 6
  ,{0x01, 0x71, 0x09, 0x05, 0x03} // 37 7
  ,{0x36, 0x49, 0x49, 0x49, 0x36} // 38 8
  ,{0x06, 0x49, 0x49, 0x29, 0x1e} // 39 9
  ,{0x00, 0x36, 0x36, 0x00, 0x00} // 3a :
  ,{0x00, 0x56, 0x36, 0x00, 0x00} // 3b ;
  ,{0x08, 0x14, 0x22, 0x41, 0x00} // 3c <
  ,{0x14, 0x14, 0x14, 0x14, 0x14} // 3d =
  ,{0x00, 0x41, 0x22, 0x14, 0x08} // 3e >
  ,{0x02, 0x01, 0x51, 0x09, 0x06} // 3f ?
  ,{0x32, 0x49, 0x79, 0x41, 0x3e} // 40 @
  ,{0x7e, 0x11, 0x11, 0x11, 0x7e} // 41 A
  ,{0x7f, 0x49, 0x49, 0x49, 0x36} // 42 B
  ,{0x3e, 0x41, 0x41, 0x41, 0x22} // 43 C
  ,{0x7f, 0x41, 0x41, 0x22, 0x1c} // 44 D
  ,{0x7f, 0x49, 0x49, 0x49, 0x41} // 45 E
  ,{0x7f, 0x09, 0x09, 0x09, 0x01} // 46 F
  ,{0x3e, 0x41, 0x49, 0x49, 0x7a} // 47 G
  ,{0x7f, 0x08, 0x08, 0x08, 0x7f} // 48 H
  ,{0x00, 0x41, 0x7f, 0x41, 0x00} // 49 I
  ,{0x20, 0x40, 0x41, 0x3f, 0x01} // 4a J
```

```

, {0x7f, 0x08, 0x14, 0x22, 0x41} // 4b K
, {0x7f, 0x40, 0x40, 0x40, 0x40} // 4c L
, {0x7f, 0x02, 0x0c, 0x02, 0x7f} // 4d M
, {0x7f, 0x04, 0x08, 0x10, 0x7f} // 4e N
, {0x3e, 0x41, 0x41, 0x41, 0x3e} // 4f O
, {0x7f, 0x09, 0x09, 0x09, 0x06} // 50 P
, {0x3e, 0x41, 0x51, 0x21, 0x5e} // 51 Q
, {0x7f, 0x09, 0x19, 0x29, 0x46} // 52 R
, {0x46, 0x49, 0x49, 0x49, 0x31} // 53 S
, {0x01, 0x01, 0x7f, 0x01, 0x01} // 54 T
, {0x3f, 0x40, 0x40, 0x40, 0x3f} // 55 U
, {0x1f, 0x20, 0x40, 0x20, 0x1f} // 56 V
, {0x3f, 0x40, 0x38, 0x40, 0x3f} // 57 W
, {0x63, 0x14, 0x08, 0x14, 0x63} // 58 X
, {0x07, 0x08, 0x70, 0x08, 0x07} // 59 Y
, {0x61, 0x51, 0x49, 0x45, 0x43} // 5a Z
, {0x00, 0x7f, 0x41, 0x41, 0x00} // 5b [
, {0x02, 0x04, 0x08, 0x10, 0x20} // 5c BACKSLASH
, {0x00, 0x41, 0x41, 0x7f, 0x00} // 5d ]
, {0x04, 0x02, 0x01, 0x02, 0x04} // 5e ^
, {0x40, 0x40, 0x40, 0x40, 0x40} // 5f _
, {0x00, 0x01, 0x02, 0x04, 0x00} // 60 `
, {0x20, 0x54, 0x54, 0x54, 0x78} // 61 a
, {0x7f, 0x48, 0x44, 0x44, 0x38} // 62 b
, {0x38, 0x44, 0x44, 0x44, 0x20} // 63 c
, {0x38, 0x44, 0x44, 0x48, 0x7f} // 64 d
, {0x38, 0x54, 0x54, 0x54, 0x18} // 65 e
, {0x08, 0x7e, 0x09, 0x01, 0x02} // 66 f
, {0x0c, 0x52, 0x52, 0x52, 0x3e} // 67 g
, {0x7f, 0x08, 0x04, 0x04, 0x78} // 68 h
, {0x00, 0x44, 0x7d, 0x40, 0x00} // 69 i
, {0x20, 0x40, 0x44, 0x3d, 0x00} // 6a j
, {0x7f, 0x10, 0x28, 0x44, 0x00} // 6b k
, {0x00, 0x41, 0x7f, 0x40, 0x00} // 6c l
, {0x7c, 0x04, 0x18, 0x04, 0x78} // 6d m
, {0x7c, 0x08, 0x04, 0x04, 0x78} // 6e n
, {0x38, 0x44, 0x44, 0x44, 0x38} // 6f o
, {0x7c, 0x14, 0x14, 0x14, 0x08} // 70 p
, {0x08, 0x14, 0x14, 0x18, 0x7c} // 71 q
, {0x7c, 0x08, 0x04, 0x04, 0x08} // 72 r
, {0x48, 0x54, 0x54, 0x54, 0x20} // 73 s
, {0x04, 0x3f, 0x44, 0x40, 0x20} // 74 t
, {0x3c, 0x40, 0x40, 0x20, 0x7c} // 75 u
, {0x1c, 0x20, 0x40, 0x20, 0x1c} // 76 v
, {0x3c, 0x40, 0x30, 0x40, 0x3c} // 77 w
, {0x44, 0x28, 0x10, 0x28, 0x44} // 78 x
, {0x0c, 0x50, 0x50, 0x50, 0x3c} // 79 y
, {0x44, 0x64, 0x54, 0x4c, 0x44} // 7a z
, {0x00, 0x08, 0x36, 0x41, 0x00} // 7b {
, {0x00, 0x00, 0x7f, 0x00, 0x00} // 7c |
, {0x00, 0x41, 0x36, 0x08, 0x00} // 7d }
, {0x10, 0x08, 0x08, 0x10, 0x08} // 7e ~
, {0x78, 0x46, 0x41, 0x46, 0x78} // 7f DEL
};

```

```
//Initialisierung des Displays
void setup() {
  LCDInit();
}

void loop() {
  LCDClear();
  LCDString("joy-IT ");
  LCDString("84x48");
  LCDString(" LED-Modul");
  delay(1000);
}

//Bewege Zeiger zur angegebenen Position
void positionXY(int x, int y) {
  LCDWrite(0, 0x80 | x);
  LCDWrite(0, 0x40 | y);
}

//Nimmt einen einzelnen Character entgegen und sucht in der Tabelle
//nach dem passenden Hexadezimal-Wert
void LCDCharacter(char character) {
  LCDWrite(LCD_DATA, 0x00);
  for (int index = 0 ; index < 5 ; index++)
  {
    LCDWrite(LCD_DATA, ASCII[character - 0x20][index]);
  }
  LCDWrite(LCD_DATA, 0x00);
}

//Nimmt die Eingabe entgegen und leitet sie
//Buchstabe für Buchstabe weiter
void LCDString(char *characters) {
  while (*characters)
  {
    LCDCharacter(*characters++);
  }
}

//setzt alle Eingaben auf dem Display zurück
void LCDClear() {
  for (int index = 0 ; index < (LCD_X * LCD_Y / 8) ; index++)
  {
    LCDWrite(LCD_DATA, 0x00);
  }
  positionXY(0, 0);
}
```

Code 4: Teil 4 des LCD-Quellcodes

```
//Initialisierung des Displays
void LCDInit() {

    pinMode(PIN_SCE, OUTPUT);
    pinMode(PIN_RESET, OUTPUT);
    pinMode(PIN_DC, OUTPUT);
    pinMode(PIN_SDIN, OUTPUT);
    pinMode(PIN_SCLK, OUTPUT);

    digitalWrite(PIN_RESET, LOW);
    digitalWrite(PIN_RESET, HIGH);

    LCDWrite(LCD_COMMAND, 0x21);
    LCDWrite(LCD_COMMAND, 0xB0);
    LCDWrite(LCD_COMMAND, 0x04);
    LCDWrite(LCD_COMMAND, 0x14);
    LCDWrite(LCD_COMMAND, 0x20);
    LCDWrite(LCD_COMMAND, 0x0C);
}

void LCDWrite(byte data_or_command, byte data) {
    digitalWrite(PIN_DC, data_or_command);
    digitalWrite(PIN_SCE, LOW);
    shiftOut(PIN_SDIN, PIN_SCLK, MSBFIRST, data);
    digitalWrite(PIN_SCE, HIGH);
}
```

Code 5: Teil 5 des LCD-Quellcodes

Verwendung mit einem Raspberry Pi

Schritt 1 – Anschließen des Displays

Schließen Sie das Display, wie im folgenden Bild 3, bzw. in der folgenden Tabelle 2, zu sehen, an die PINs des Raspberry Pis an.

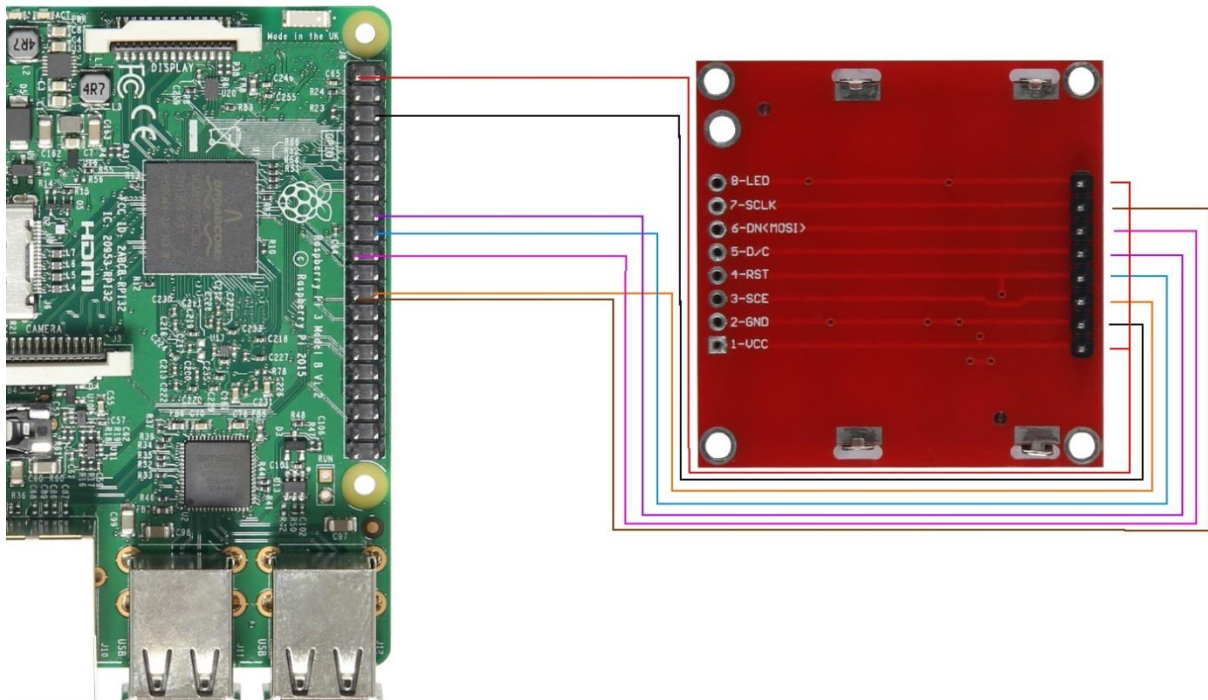


Bild 3: Verbindung zwischen Raspberry Pi und LCD-Modul

Raspberry Pi PIN	LCD Modul PIN
PIN 1 (3.3 V)	8 LED
PIN 23 (BCM 11 / SCLK)	7 SCLK
PIN 19 (MCM 10 / MOSI)	6 DN <MOSI>
PIN 16 (BCM 23)	5 D/C
PIN 18 (BCM 24)	4 RST
PIN 24 (BCM 8)	3 SCC
PIN 6 (GND)	2 GND
PIN 1 (3.3 V)	1 VCC

Tabelle 2: PIN-Verbindung zwischen Raspberry und LCD-Modul

Schritt 2 – Installation der Software

Sollten Sie bereits ein aktuelles Raspbian-System auf Ihrem Raspberry verwenden, so können Sie diesen Schritt überspringen und sofort mit Schritt 3 fortfahren.

Installieren Sie auf Ihre SD-Karte mit Hilfe des „Win32 Disk Imager“-Programms das aktuelle Raspbian Image, welches Sie unter dem folgenden [Link](#) zum Download finden.

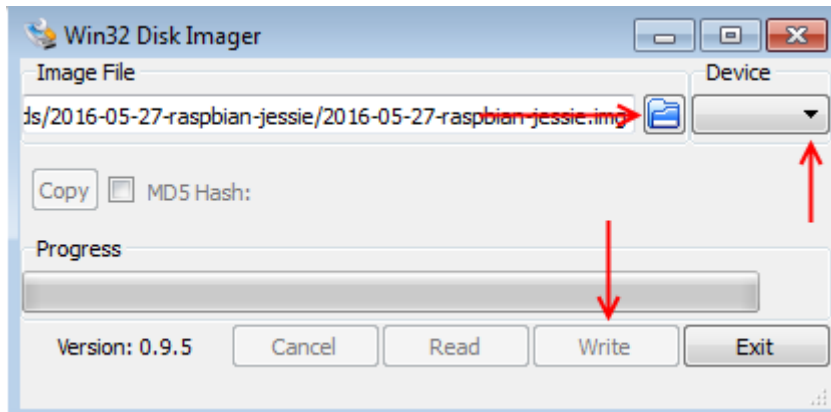


Bild 4: Screenshot des Win32 Disk Imagers

Schritt 3 – Installation der Bibliotheken

Sobald Sie die Installation abgeschlossen und das System gestartet haben, öffnen Sie die Terminal-Konsole und führen Sie folgende Kommandos aus:

```
sudo apt-get install python-pip python-dev build-essential  
sudo pip install RPi.GPIO
```

Terminal 1: Installation der GPIO Bibliothek

```
sudo apt-get install python-imaging
```

Terminal 2: Installation der Python Bibliothek

Schritt 4 – Installation des LCD-Displays

Um Ihnen eine möglichst einfache und schnelle Installation des Displays zu ermöglichen, beziehen wir uns im Folgenden auf eine von [Adafruit](#) erstellte Bibliothek. Diese können Sie, nach dem Sie die in Schritt 3 erforderlichen GPIO- und Python-Bibliothek installiert haben, wie folgt installieren.

Öffnen Sie die Terminal-Konsole und führen Sie folgende Kommandos aus:

```
sudo apt-get install git
git clone https://github.com/adafruit/Adafruit_Nokia_LCD.git
cd Adafruit_Nokia_LCD
sudo python setup.py install
```

Terminal 3: LCD Installation

Die Installation des Displays ist nun abgeschlossen.

Rufen Sie den **examples** Ordner auf, so können Sie auf verschiedene Beispiele zugreifen und so das Display ansteuern.

Natürlich können Sie auch die Beispiele nach Ihren Wünschen bearbeiten und so die Anzeige nach eigenen Vorstellungen konfigurieren.

```
cd examples
```

Terminal 4: Öffnen des examples Ordners

```
sudo python animate.py
```

Terminal 5: Ausführen des Animations-Beispiels

```
sudo python image.py
```

Terminal 6: Ausführen des Bild-Beispiels

```
sudo python shapes.py
```

Terminal 7: Ausführen des Formen-Beispiels