

```

#include<LiquidCrystal.h>
/* Create object named lcd of the class LiquidCrystal */
LiquidCrystal lcd(13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3);    /* For 8-bit mode */
//LiquidCrystal lcd(13, 12, 11, 6, 5, 4, 3);    /* For 4-bit mode */

unsigned char Character1[8] = { 0x04, 0x1F, 0x11, 0x11, 0x1F, 0x1F, 0x1F, 0x1F };    /* Custom Character 1 */
unsigned char Character2[8] = { 0x01, 0x03, 0x07, 0x1F, 0x1F, 0x07, 0x03, 0x01 };    /* Custom Character 2 */

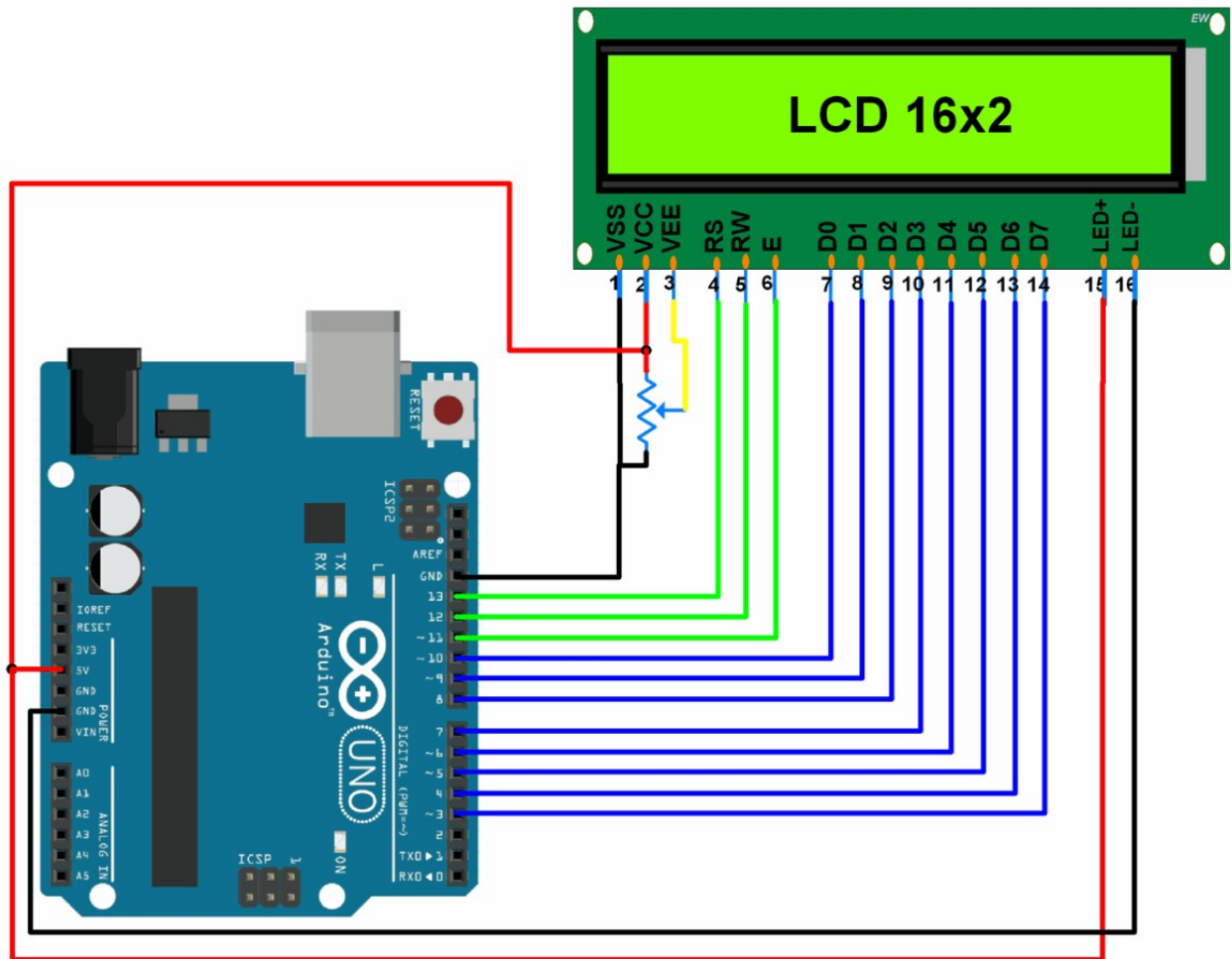
void setup() {
    lcd.begin(16,2);    /* Initialize 16x2 LCD */
    lcd.clear();    /* Clear the LCD */
    lcd.createChar(0, Character1);    /* Generate custom character */
    lcd.createChar(1, Character2);
}

void loop() {
    lcd.setCursor(0,0);    /* Set cursor to column 0 row 0 */
    lcd.print("Hello!!!!");    /* Print data on display */
    lcd.setCursor(0,1);
    lcd.write(byte(0));    /* Write a character to display */
    lcd.write(1);
}

```

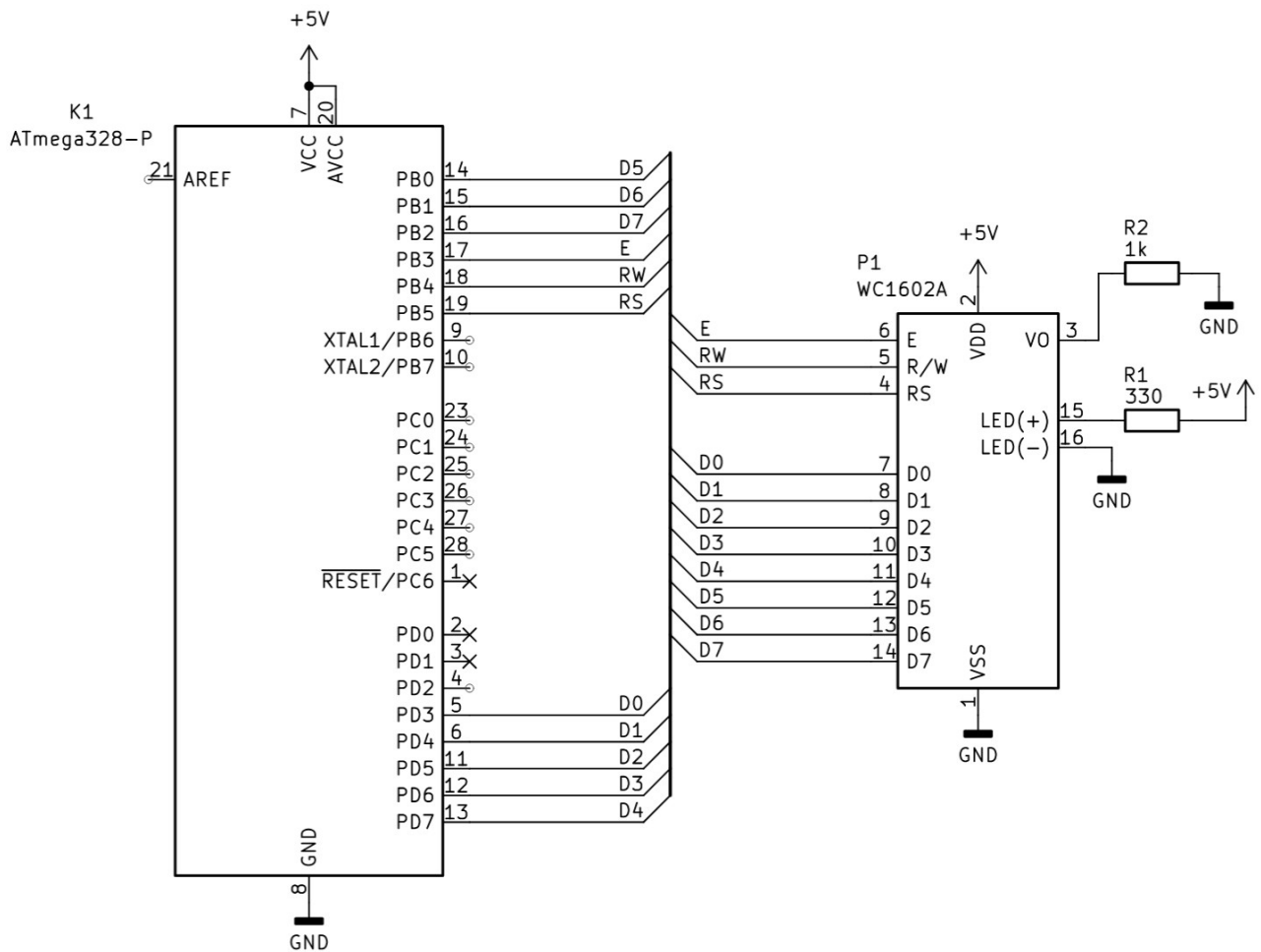
Figure 1: Quelle: <https://www.electronicwings.com/sensors-modules/lcd-16x2-display-module>

Die LiquidCrystal-Bibliothek wird verwendet, um LCD-Displays mit HD44780 mit einem Arduino zu steuern. Zu Beginn müssen die Pins beim Erstellen des Objektes übergeben werden. Dann wird das Display geleert und die eigenen Buchstaben erstellt. Die Buchstaben werden in 6x8 Bit Array gespeichert. Das heißt jeder Pixel hat einen Bit. In der Hauptschleife wird der Cursor auf die erste Stelle in der ersten Zeile gestellt. Danach wird „Hello!!!“ geschrieben, standardmäßig wandert der Cursor mit dem Geschriebenen. Anschließend wird auf die nächste Zeile gewechselt. Die eigenen Buchstaben werden geschrieben. Es würde auch reichen diesen Code nur einmal auszuführen.



1602A LCD Display Anschlüsse:

VSS	GND
VCC	5V
RS	0 = Befehlsregister, 1 = Datenregister
RW	0 = Schreiben, 1 = Lesen
E	Enable, Freigabe- und Taktleitung
DBx	Datenleitung
LED+	5V
LED-	GND



Ich bin davon ausgegangen dies mit „Verdrahtungsplan“ gemeint ist. Doch wahrscheinlich ist das folgende Bild damit gemeint:



```

/*
Pinbelegung:
| | LCD | | Arduino | | MC
VSS      GND
VDD      5V
V0       5V -> 1 kOhm
RS       13
RW       12
E        11
D0       10
D1       9
D2       8
D3       7
D4       6
D5       5
D6       4
D7       3
A        5V -> 330 Ohm
K        GND
*/

```

```

1  /*
2  TI_70_LCD_8Bit_Testaufbau
3
4  Pinbelegung:
5
6  | LCD | Arduino | MC
7  VSS   GND
8  VDD   5V
9  V0    5V -> 1 kOhm
10 RS    13          PB5
11 RW    12          PB4
12 E     11          PB3
13 D0    10          PB2
14 D1    9           PB1
15 D2    8           PB0
16 D3    7           PD7
17 D4    6           PD6
18 D5    5           PD5
19 D6    4           PD4
20 D7    3           PD3
21 A     5V -> 330 Ohm
22 K     GND
23 */
24
25 #include<LiquidCrystal.h>
26 /* Create object named lcd of the class LiquidCrystal */
27 LiquidCrystal lcd(13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3); /* For 8-bit mode */
28 //LiquidCrystal lcd(13, 12, 11, 6, 5, 4, 3); /* For 4-bit mode */
29
30 unsigned char Character1[8] = { 0x04, 0x1F, 0x11, 0x11, 0x1F, 0x1F, 0x1F, 0x1F }; /* Custom Character 1 */
31 unsigned char Character2[8] = { 0x01, 0x03, 0x07, 0x1F, 0x1F, 0x07, 0x03, 0x01 }; /* Custom Character 2 */
32
33 void setup() {
34     Serial.begin(9600);
35     Serial.println("LCD Display Test");
36
37     lcd.begin(16,2); /* Initialize 16x2 LCD */
38     lcd.clear(); /* Clear the LCD */
39     lcd.createChar(0, Character1); /* Generate custom character */
40     lcd.createChar(1, Character2);
41     Serial.println("LCD wurde initialisiert");
42
43     lcd.setCursor(0,0); /* Set cursor to column 0 row 0 */
44     lcd.print("Hello!!!!"); /* Print data on display */
45     Serial.println("Hello!!!!");
46     lcd.setCursor(0,1);
47     lcd.write(byte(0)); /* Write a character to display */
48     lcd.write(1);
49 }
50
51 void loop() {
52 }

```

Output Serial Monitor x

Message (Enter to send message to 'Arduino Uno' on '/dev/ttyUSB0')

```

LCD Display Test
LCD wurde initialisiert
Hello!!!!

```