

```

1  /*   Optischer_Abstandswarner       - TESTAUFBAU -           v1.21           Juli 2026
2  -----
3  Distanzmessung mit dem VL53L1X Laser Sensor bis 4 Meter.
4  Inspiriert durch: https://github.com/pololu/vl53l1x-arduino
5  Anzeige der Distanz auf einem kleinen 0,96"- SSD1306 Display.
6  11.10.2024 https://www.film-werk56.de/linux.html von Peter Klingbeil
7  17.07.2026 erweitert durch "Heimwerker Leopoldi" zur Abstandswarnung Garage
8      Test-Hardware: Arduino Nano, vl53l1x, 0,96" SSD1306 Display, Ampelmodul,
9                      LED-Stripe WS2812B (100uF, 220E), Summer 5V <40mA
10 vl.21 Texte STOP und ENDE                                     */
11
12 #define Abstand_ROT           15 // < cm | änderbare TESTaufbau werte -----
13 #define Abstand_ROT_blink     30 // < cm | Garagenabstände ausmessen -----
14 #define Abstand_GELB          40 // >= cm | und eintragen !!!-----
15 #define Abstand_GRUEN         (Abstand_GELB+1) // Modul fkt. bis 400cm
16 #define RotAusZeit             2000 // alles abschalten wenn ROT für xxx Millisek.-----
17 #define NUMPIXELS              17 // TEST: Wegen Stromversorgung über USB- Port -----
18 #define BRIGHTNESS            50 // ==== Set BRIGHTNESS to about 1/5 (max = 255) -----
19
20 #include <Wire.h>                // Funktion: Distanzauswertung in cm
21 #include "VL53L1X.h"            // Distanzanzeige: gruen, gelb, rot-blink, rot
22 #include <Arduino.h>            // Summer wenn rot-blink und rot
23 #include <Adafruit_GFX.h>       // Nach "RotAuszeit" ausschalten der Anzeigen.
24 #include <Adafruit_SSD1306.h>
25 #include <Adafruit_NeoPixel.h>
26
27 #define PIN                    6 // Digitalpin Neopixel Data
28 Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
29
30 bool flag = true;              // für blink wechsel
31 int8_t dis = 1;                // case distanz: 1= Grün, 2= Gelb, 3= Rot-Blink, 4= ROT
32
33 // LED-Ampelmodul und Summer -PIN - Definitionen
34 const int PIN_LED_GRUEN = 12;
35 const int PIN_LED_GELB  = 11;
36 const int PIN_LED_ROT   = 10;
37 const int PIN_Summer    = 2;
38
39 #define SCREEN_WIDTH 128 // OLED display width, in pixels
40 #define SCREEN_HEIGHT 64 // OLED display height, in pixels
41 #define OLED_RESET    4 // Reset pin # (or -1 if sharing Arduino reset pin)
42 #define SCREEN_ADDRESS 0x3C // datasheet Address; 0x3D for 128x64, 0x3C for 128x32
43
44 unsigned int Distance;
45 unsigned long startZeit;
46
47 VL53L1X sensor;
48 Adafruit_SSD1306 display = Adafruit_SSD1306(128, 64, &Wire);
49
50 void setup()
51 {
52     while (!Serial) {}
53     Serial.begin(9600);
54     Wire.begin();
55     Wire.setClock(400000); // use 400 kHz I2C Bus
56
57     // LED Strip vorbereiten - alle auf weiss
58     pixels.begin();
59     pixels.setBrightness(BRIGHTNESS);
60     pixels.fill(pixels.Color(100, 100, 100));
61     pixels.show(); // Befehl zum Senden der Daten an die LEDs
62     delay(500);
63
64     // LED PIN's vorbereiten
65     pinMode(PIN_LED_GRUEN, OUTPUT);
66     pinMode(PIN_LED_GELB, OUTPUT);
67     pinMode(PIN_LED_ROT, OUTPUT);
68     pinMode(PIN_Summer, OUTPUT);
69     // Ausgangszustand herstellen

```

```

70 digitalWrite(PIN_LED_GRUEN,HIGH); //evtl. weiß
71 digitalWrite(PIN_LED_GELB,LOW);
72 digitalWrite(PIN_LED_ROT,LOW);
73 digitalWrite(PIN_Summer,LOW);
74
75 sensor.setTimeout(5); //500
76 if (!sensor.init())
77 {
78     Serial.println("Fehler Initialisierung Sensor!");
79     while (1);
80 }
81
82 // Init Display
83 display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // Address 0x3C for 128x64
84 display.clearDisplay();
85
86 // Clear Display Puffer.
87 display.clearDisplay();
88 display.display();
89
90 display.setTextSize(2);
91 display.setTextColor(SSD1306_WHITE);
92 display.setCursor(0,20); //display.setCursor(0,20);
93 display.print("Laser V1.2\n");
94 display.display(); // Anzeigen
95 Serial.println("Laser Version 1.2");
96
97 sensor.setDistanceMode(VL53L1X::Long);
98 sensor.setMeasurementTimingBudget(50000);
99 sensor.startContinuous(50);
100
101 delay(1500); // Version anzeigen, delay für VL531X
102 }
103
104 void loop()
105 {
106     Distance = sensor.read();
107     Distance = Distance/10; // kill mm
108     //Serial.print(Distance);
109     if (sensor.timeoutOccurred()) { Serial.print(" TIMEOUT"); }
110
111     // Anzeige Abstand
112     display.clearDisplay();
113     display.setTextSize(4); //4
114     display.setCursor(24,20); //8,0
115     Serial.print("Abstand: ");
116
117     if (Distance >= Abstand_GRUEN) dis = 1; //Switch case Variable
118     if (Distance <= Abstand_GELB) dis = 2; //nach Abstand
119     if (Distance <= Abstand_ROT_blink) dis = 3;
120     if (Distance <= Abstand_ROT) dis = 4;
121
122     switch (dis) {
123         case 4:
124             digitalWrite(PIN_Summer,HIGH);
125             digitalWrite(PIN_LED_GRUEN,LOW);
126             digitalWrite(PIN_LED_GELB,LOW);
127             digitalWrite(PIN_LED_ROT,HIGH); // ROT
128             pixels.fill(pixels.Color(255, 0, 0));
129             pixels.show(); // Befehl zum Senden der Daten an die LEDs
130             Serial.println(Distance);
131             //display.print(Distance);
132             display.clearDisplay();
133             display.setTextSize(4); //4
134             display.setCursor(16,20); //8,0
135             display.print("STOP");
136             display.display(); // Refrech Display
137             delay(50);
138

```

```

139     if (millis() - startZeit >= RotAusZeit) {          // nach 5s ROT ausschalten
140         digitalWrite(PIN_Summer, LOW);
141         digitalWrite(PIN_LED_GRUEN, LOW);
142         digitalWrite(PIN_LED_GELB, LOW);                // Pixels und LEDs aus
143         digitalWrite(PIN_LED_ROT, LOW);                // ROT
144         pixels.clear();
145         pixels.show(); // Befehl zum Senden der Daten an die LEDs
146         display.clearDisplay();
147         display.setTextSize(4); //4
148         display.setCursor(16, 20); //8,0
149         display.print("ENDE");
150         display.display(); // Refrech Display
151         while(true) {
152             // Der Arduino tut ab hier nichts mehr
153         }
154     }
155     break;
156 case 3:
157     startZeit = millis();                                // Millis neu starten
158     if (flag) {                                          // ROT- Blinken
159         digitalWrite(PIN_Summer, HIGH);
160         digitalWrite(PIN_LED_GRUEN, LOW);
161         digitalWrite(PIN_LED_GELB, LOW);
162         digitalWrite(PIN_LED_ROT, HIGH);                // ROT an
163         pixels.fill(pixels.Color(255, 0, 0));
164         pixels.show(); // Befehl zum Senden der Daten an die LEDs
165         Serial.println(Distance);
166         display.print(Distance);
167         flag = false;
168         delay(50); }
169     else {
170         digitalWrite(PIN_Summer, LOW);
171         digitalWrite(PIN_LED_GRUEN, LOW);
172         digitalWrite(PIN_LED_GELB, LOW);                // Pixels aus
173         digitalWrite(PIN_LED_ROT, LOW); //ROT
174         pixels.clear();
175         pixels.show(); // Befehl zum Senden der Daten an die LEDs
176         Serial.println(Distance);
177         display.print(Distance);
178         flag = true;
179         delay(50); }
180     break;
181 case 2:
182     startZeit = millis();                                // Millis neu starten
183     digitalWrite(PIN_Summer, LOW);
184     digitalWrite(PIN_LED_GRUEN, LOW);                    //GELB
185     digitalWrite(PIN_LED_GELB, HIGH);
186     digitalWrite(PIN_LED_ROT, LOW);
187     pixels.fill(pixels.Color(252, 244, 0));
188     pixels.show(); // Befehl zum Senden der Daten an die LEDs
189     Serial.println(Distance);
190     display.print(Distance);
191     break;
192 case 1:
193     startZeit = millis();                                // Millis neu starten
194     digitalWrite(PIN_Summer, LOW);
195     digitalWrite(PIN_LED_GRUEN, HIGH);                    //GRUEN
196     digitalWrite(PIN_LED_GELB, LOW);
197     digitalWrite(PIN_LED_ROT, LOW);
198     pixels.fill(pixels.Color(0, 91, 9));
199     pixels.show(); // Befehl zum Senden der Daten an die LEDs
200     Serial.println(Distance);
201     display.print(Distance);
202     break;
203 }
204 display.display(); // Refrech Display
205 delay(200); // 0,2 Sekunden Pause, um die Anzeige ruhig zu halten
206 } // Programmende

```