

```

/*Code für Ultraschallmodul + Leds + Summer + LCD-Bildschirm + Schrittmotor*/

/* Konstanten für Pins */

const byte TRIGGER = 6; // Pin TRIGGER

const byte ECHO = 7; // Pin ECHO

/* Konstanten für timeout */

const unsigned long Messung_TIMEOUT = 25000UL; // 25ms = 8m à 340m/s/ Schallgeschwindigkeit
in Luft mm/us */

const float SOUND_SPEED = 340.0 / 1000;


#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);


int in1 = A1; // broche A1 reliée à IN1
int in2 = A2; // broche A2 reliée à IN2
int in3 = A3; // broche A3 reliée à IN3
int in4 = A4; // broche A4 reliée à IN4
int anzahl_der_Schritte = 0;


void setup() {

    lcd.begin(16, 2);


    pinMode(8, OUTPUT); //Rote LED BUZZER
    digitalWrite(8, LOW);
    pinMode(9, OUTPUT); // Grüne LED
    digitalWrite(9, HIGH);
    pinMode(TRIGGER, OUTPUT);
    digitalWrite(TRIGGER, LOW);
    pinMode(ECHO, INPUT);

    /*Pin A1 bis A4 des Arduino definiert in output*/
    pinMode(in1, OUTPUT);

```

```

pinMode(in2, OUTPUT);
pinMode(in3, OUTPUT);
pinMode(in4, OUTPUT);
}

void loop()
{
  /* 1 Sendet einen HIGH-Impuls von 10 us auf den Trigger-Pin */
  digitalWrite(TRIGGER, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIGGER, LOW);

  /* 2 Messung der Zeit zwischen dem Senden des Impulses und seinem Echo */
  long messung = pulseIn(ECHO, HIGH, Messung_TIMEOUT);

  /* 3 Entfernungsberechnung */

  float entfernung_mm = messung / 2.0 * SOUND_SPEED;

  Serial.println("entfernung: ");
  Serial.print(entfernung_mm);
  Serial.print("mm (");

  lcd.setCursor(0,0);
  lcd.print("Entfernung in mm");
  lcd.setCursor(0,1);
  lcd.print(entfernung_mm);

  lcd.display();

  if (entfernung_mm > 600)

```

```

{
    digitalWrite(8, LOW);
    digitalWrite(9, HIGH);

}

else if (entfernung_mm < 1)
{
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
}
else
{
    digitalWrite(8, HIGH);
    digitalWrite(9, LOW);

    /*Drehung im Uhrzeigersinn :2048 Schritte = eine komplette Umdrehung der externen
Motorachse*/

    for (int Zyklus1=0; Zyklus1<2048; Zyklus1++)
    {
        switch(anzahl_der_Schritte){
        case 0:
            digitalWrite(in1, HIGH);
            digitalWrite(in2, LOW);
            digitalWrite(in3, LOW);
            digitalWrite(in4, LOW);
            break;
        case 1:
            digitalWrite(in1, LOW);
            digitalWrite(in2, HIGH);

```

```

digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
break;
case 2:
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
break;
case 3:
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
}
anzahl_der_Schritte++;
if(anzahl_der_Schritte > 3){
anzahl_der_Schritte = 0;
}

delay(2); /* Mit Delay die Repetitionszeit und damit die Motordrehzahl steuern*/
}

```

```

for (int Zyklus2=0; Zyklus2<2048; Zyklus2++)
{
switch(anzahl_der_Schritte){
case 0:
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);

```

```
digitalWrite(in4, HIGH);
break;
case 1:
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
break;
case 2:
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
break;
case 3:
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
}
anzahl_der_Schritte++;
if(anzahl_der_Schritte > 3){
anzahl_der_Schritte = 0;
}
delay(2);
}
}

delay(1000);
lcd.clear();
}
```