

Projektset – Wetterstation

Einleitung

Du möchtest das Klima an einem Ort überwachen? Dann wird es Zeit, dass du deine eigene Wetterstation baust. Bei diesem Aufbau kannst du die Temperatur und Luftfeuchtigkeit mit einem DHT-11 Sensor messen. Außerdem wird mit einem Fotowiderstand die Helligkeit gemessen.

Das reicht dir nicht? Kein Problem. Mit dem MQ-135 Sensor kannst du außerdem Messwerte zur Luftqualität erfassen.

Inhalt

Das Projektset enthält folgende Teile:

- Funduino UNO R3
- Montageplatte mit Breadboard
- 40x Breadboardkabel M/W – 10cm
- 40x Breadboardkabel M/M – 10cm
- 40x Breadboardkabel M/M – 20cm
- Sensor - DHT 11
- Sensor - MQ-135
- Display - I2C LCD Modul 16x02
- Fotowiderstand
- LED Blau

Helligkeit messen

Erklärung

Als erstes starten wir damit, die Helligkeit mit dem Fotowiderstand zu messen. Eine blaue LED soll leuchten, sobald es dunkel wird.

Der Mikrocontroller soll über einen Fotowiderstand auslesen, wie hell es ist. Dazu nutzt man ein einfaches physikalisches Prinzip. Wenn in einem Stromkreis zwei Verbraucher hintereinander angeschlossen sind (Reihenschaltung), dann „teilt“ sie sich auch die gemeinsam anliegende Spannung.

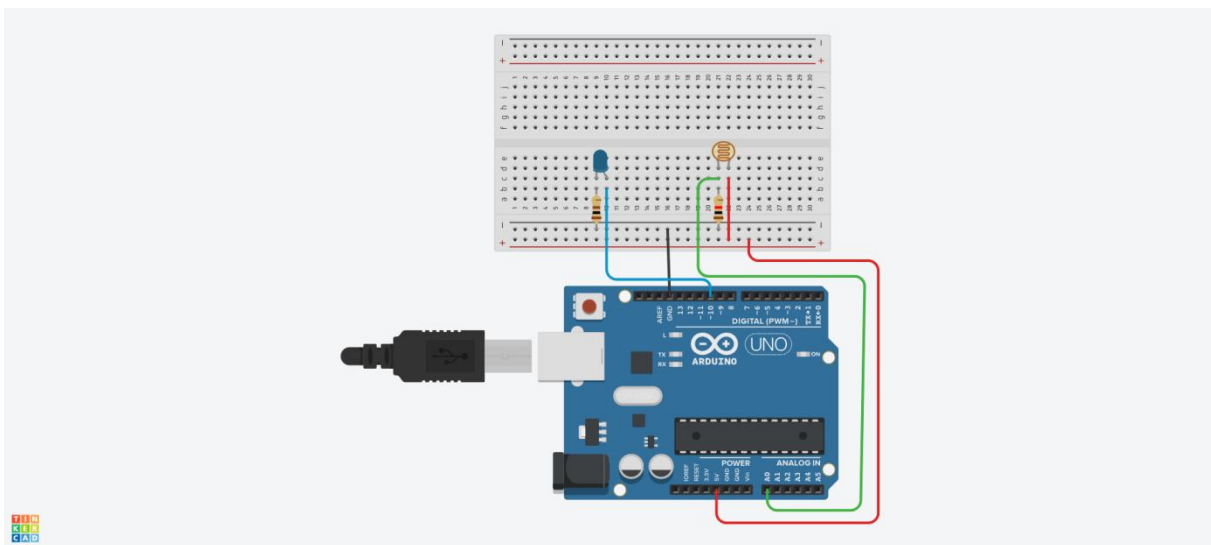
Der Fotowiderstand ändert seinen Widerstand in Abhängigkeit der Lichtstärke. Diesen Effekt nutzt man, um anhand der an ihr anliegenden Spannung einen Wert für Helligkeit bzw. Dunkelheit in Form von verschiedenen Spannungen abzulesen. Damit man hier überhaupt eine Spannungsteilung erzeugen kann, schließt man den Fotowiderstand und einen

Widerstand (1 – 10 K Ohm, je nach verwendetem Fotowiderstand. Der Widerstand sollte einen ähnlichen Widerstandswert wie der Fotowiderstand haben) in Reihe an und verbindet sie mit 5 Volt und der „Erdung“ (Ground / GND) – siehe Aufbau.

Das Mikrocontroller-Board ist in der Lage, analoge Signale (Spannung) zu messen und diese zu verarbeiten. Dies geschieht mit den analogen Eingängen auf dem Board. Dieser wandelt den gemessenen Spannungswert in eine Zahl um, die dann weiter verarbeitet werden kann. 0 Volt entspricht dabei der Zahl 0 und der höchste Messwert 5 Volt entspricht der Zahl 1023 (0 bis 1023 entspricht 1024 Zahlen = 10 Bit).

Schaltskizze

Hier findest du eine Skizze für den Aufbau:



Sobald du die Verkabelung abgeschlossen hast, solltest du diese erneut anhand der Skizze überprüfen. Anschließend kannst du mit der Programmierung fortfahren.

Der Programmcode

```
int eingang= A0; //Das Wort „eingang“ steht jetzt für den Wert
„A0“ (Bezeichnung vom Analogport 0)

int LED = 10; //Das Wort „LED“ steht jetzt für den Wert 10

int sensorWert = 0; //Variable für den Sensorwert mit 0 als Startwert

void setup()//Hier beginnt das Setup.
{
Serial.begin(9600); //Die Kommunikation mit dem seriellen Port wird gestartet.
Das benötigt man, um sich den tatsächlich ausgelesenen Wert später im serial
monitor anzeigen zu lassen.
```

```
pinMode (LED, OUTPUT); //Der Pin mit der LED (Pin 10) ist jetzt ein Ausgang
//Der analoge Pin muss nicht definiert werden.
}

void loop()
{
  //Mit dieser Klammer wird der Loop-Teil geöffnet.
  sensorWert =analogRead(eingang); //Die Spannung an dem Fotowiderstand auslesen
  und unter der Variable „sensorWert“ abspeichern.
  Serial.print("Sensorwert = " ); //Ausgabe am Serial-Monitor: Das Wort
  „Sensorwert: „
  Serial.println(sensorWert); //Ausgabe am Serial-Monitor. Mit dem Befehl
  Serial.print wird der Sensorwert des Fotowiderstandes in Form einer Zahl
  zwischen 0 und 1023 an den serial monitor gesendet.

  if (sensorWert > 512 ) //Wenn der Sensorwert über 512 beträgt...
  {
    digitalWrite(LED, HIGH); //...soll die LED leuchten...
  }
  else //andernfalls...
  {
    digitalWrite(LED, LOW); //....soll sie nicht leuchten.
  }
  delay (50); //Eine kurze Pause, in der die LED an oder aus ist
} //Mit dieser letzten Klammer wird der Loop-Teil geschlossen.
```

Temperatur und Luftfeuchtigkeit messen

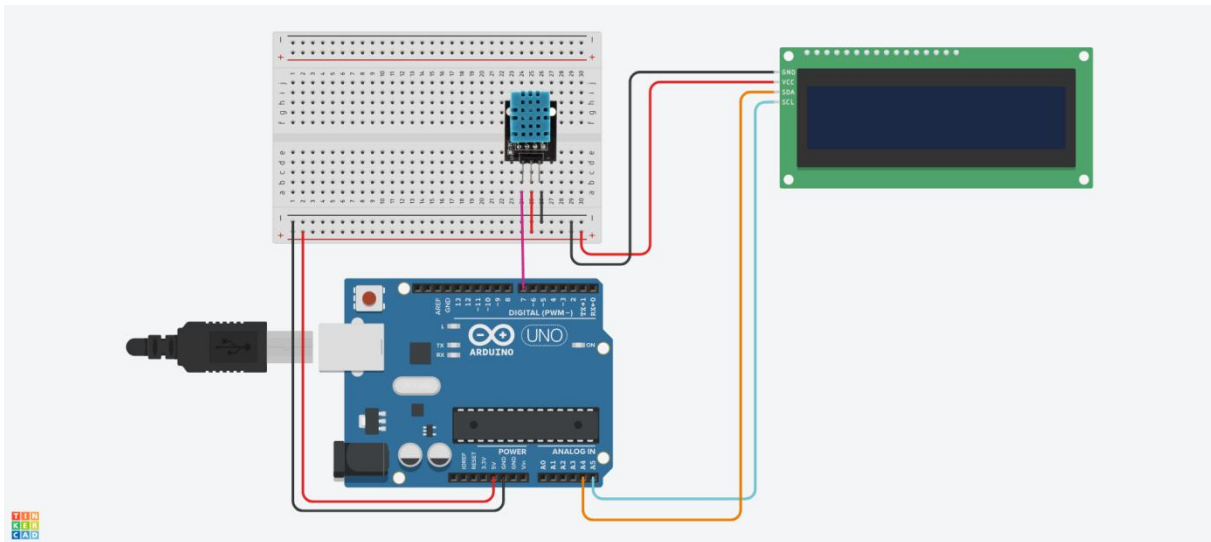
Erklärung

Jetzt wollen wir die Temperatur und Luftfeuchtigkeit messen. Die Messwerte sollen auf einem LCD dargestellt werden.

Der DHT11 Sensor misst im Luftfeuchtigkeitsbereich von ca. 20-80 % (5 % Genauigkeit) und im Temperaturbereich von ca. 0-50 °C (ca. 2 °C Genauigkeit).

Schaltskizze

Eine Skizze für den Aufbau findest du hier:



Sobald du die Verkabelung abgeschlossen hast, solltest du diese erneut anhand der Skizze überprüfen. Anschließend kannst du mit der Programmierung fortfahren.

Programmierung

Die Programmierung erfolgt in der Arduino IDE. Einen Beispielsketch findest du hier:

```
#include <DHT.h>; // Vorher hinzugefügte DHT Bibliothek einbinden
#include <LiquidCrystal_I2C.h> // Vorher hinzugefügte LiquidCrystal_I2C
Bibliothek einbinden. Diese Bibliothek ist von Marco Schwartz und Frank de
Brabander.

#include <Wire.h> // Wire Bibliothek einbinden

LiquidCrystal_I2C lcd(0x27,16,2); // Hier wird festgelegt um was für einen
Display es sich handelt.
```

```
#define DHTPIN 7 // an Pin 7 wird der DHT11 angeschlossen
#define DHTTYPE DHT11 // Hier wird definiert dass der DHT11 angeschlossen
wird
DHT dht(DHTPIN, DHTTYPE);
// Hier werden die Variablen definiert
int feuchtigkeit;
int temperatur;

void setup()
{
  Serial.begin(9600);
  Serial.println("Temperatur und Feuchtigkeit Sensor");
  dht.begin();
  lcd.init(); //Im Setup wird das LCD gestartet
  lcd.backlight(); //Hintergrundbeleuchtung einschalten (lcd.noBacklight();
  schaltet die Beleuchtung aus).
}

void loop()
{
  // Die Daten werden ausgelesen und es werden Variablen eingesetzt für
  Temperatur (temperatur) und Feuchtigkeit (feuchtigkeit).
  feuchtigkeit = dht.readHumidity();
  temperatur = dht.readTemperature();

  //Im seriellen Monitor soll der Text und die dazugehörigen Werte angezeigt
  werden
  Serial.print("Feuchtigkeit: ");
  Serial.print(feuchtigkeit);
  Serial.print(" %, Temp: ");
  Serial.print(temperatur);
  Serial.println(" ° Celsius");
```

```
//Hier wird die Position des ersten Zeichens festgelegt. In diesem Fall  
bedeutet (0,0) das erste Zeichen in der ersten Zeile.
```

```
lcd.setCursor(0, 0);
```

```
lcd.print("Temperatur: ");
```

```
lcd.print(temperatur);
```

```
lcd.print("C");
```

```
// In diesem Fall bedeutet (0,1) das erste Zeichen in der zweiten Zeile
```

```
lcd.setCursor(0, 1);
```

```
lcd.print("Feuchtigkeit:");
```

```
lcd.print(feuchtigkeit);
```

```
lcd.print("%");
```

```
delay(1000); // Pause 1 Sek.
```

```
}
```

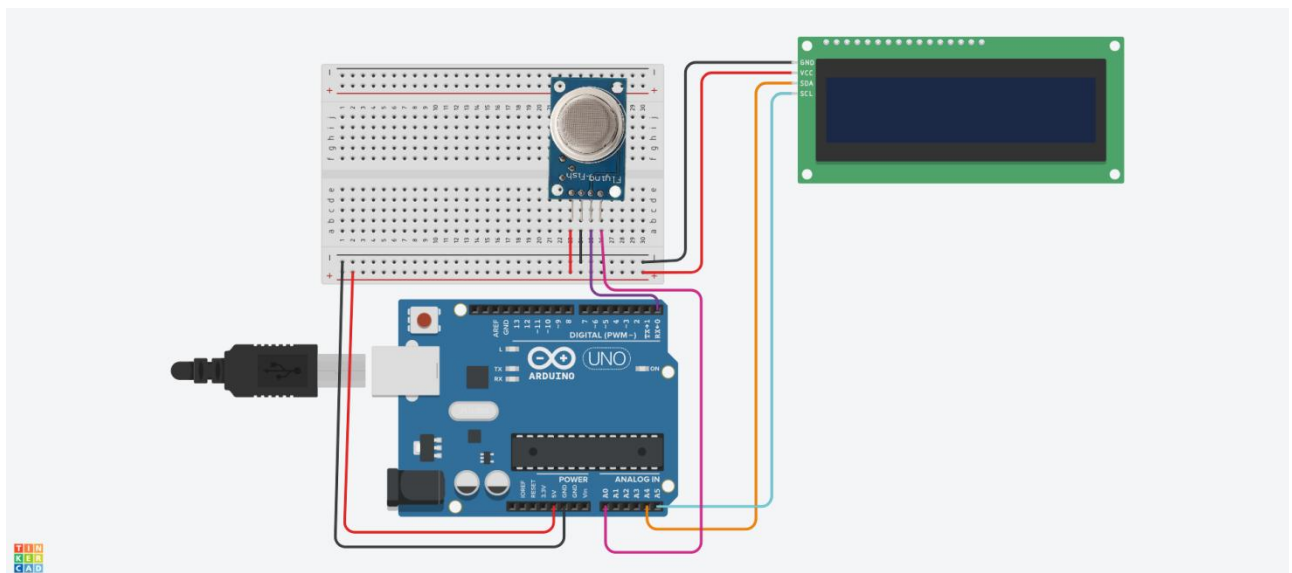
Luftqualität messen

Erklärung

Der MQ-135 Sensor ist ein Modul, welches die Konzentration verschiedener Schadstoffe in der Luft erfasst. Der Sensor gibt die erfassten Schadstoffwerte als analogen Wert aus. Diese können im Anschluss im seriellen Monitor ausgegeben oder anderweitig verarbeitet werden. Wichtig ist, dass dieser Sensor zuerst kalibriert werden muss, da die erfassten Werte ansonsten nutzlos sind. Eine genauere Anleitung für die Kalibrierung findest du auf funduino.de unter Anleitung Nr. 27. Wir werden die Luftqualität anhand eines glücklichen oder traurigen Smiley auf dem LCD darstellen.

Schaltskizze

Hier findest du eine Skizze für die Verkabelung:



Der Programmcode

```
#include <LiquidCrystal_I2C.h> // Vorher hinzugefügte LiquidCrystal_I2C
Bibliothek einbinden

#include <Wire.h> // Wire Bibliothek einbinden

LiquidCrystal_I2C lcd(0x27,16,2); // Hier wird festgelegt um was für einen
Display es sich handelt.
```

```
byte gluecklich[8] = { // Hier wird der glückliche Smiley erstellt
    B00000,
    B00000,
    B01010,
    B00000,
    B10001,
    B01110,
    B00000,
    B00000
};

byte traurig[8] = { // Hier wird der traurige Smiley erstellt
    B00000,
    B00000,
    B01010,
    B00000,
    B01110,
    B10001,
    B00000,
    B00000
};

// Hier werden die Variablen definiert
int eingang= A0;
int Luftqualitaet;

void setup(){
    Serial.begin(9600);
    pinMode( 1 , INPUT);
    Serial.println("Temperatur und Feuchtigkeit Sensor");
    lcd.init(); //Im Setup wird das LCD gestartet
```



```
lcd.backlight(); //Hintergrundbeleuchtung einschalten (lcd.noBacklight();
schaltet die Beleuchtung aus).

lcd.createChar(1, gluecklich); // Der glückliche Smiley hat jetzt die Nummer 1
lcd.createChar(0, traurig); //Der traurige Smiley hat jetzt die Nummer 0
}

void loop()
{
Luftqualitaet = analogRead(1); // Zuerst wird der Wert am Pin A0 ausgelesen...

Serial.print("Luftqualitaet: ");
Serial.println(Luftqualitaet, DEC);          //... und anschließend im
seriellen Monitor als Dezimalzahl ausgegeben.

Serial.println();

//Hier wird die Position des ersten Zeichens festgelegt. In diesem Fall
bedeutet (0,0) das erste Zeichen in der ersten Zeile.

lcd.setCursor(0, 0);
lcd.print("Quali:");
if (Luftqualitaet < 50)
{
lcd.write(byte(1));
}
if (Luftqualitaet > 50)
{
lcd.write(byte(0));
}

delay(1000); // Pause 1 Sek.
}
```

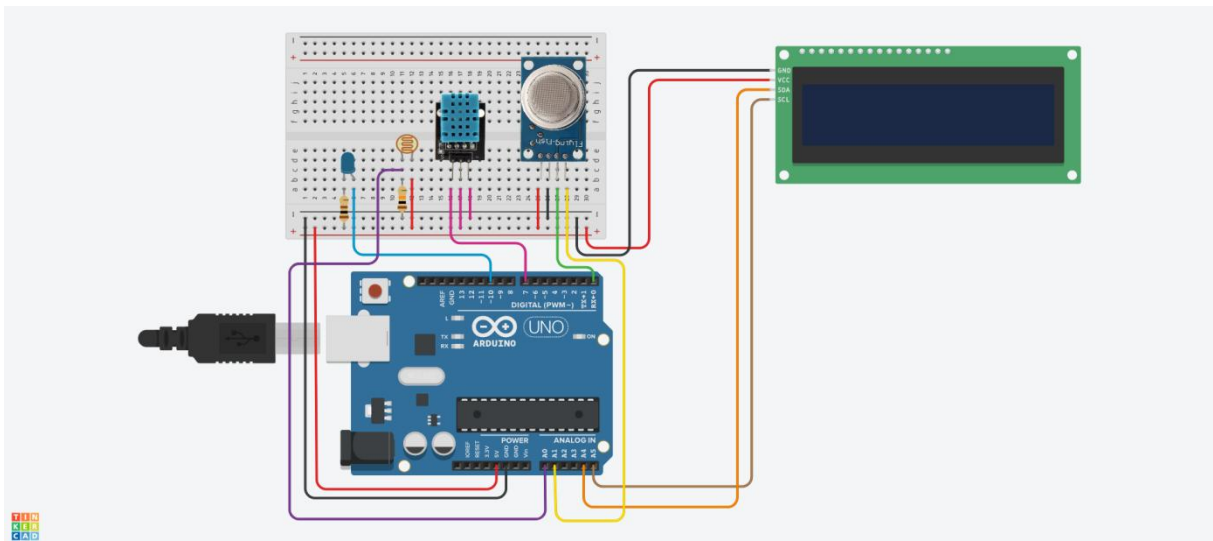
Die Wetterstation

Erklärung

Jetzt kombinieren wir alle Module miteinander und erhalten unsere Wetterstation. Die Messwerte von dem DHT11 und dem MQ-135 Sensor werden auf dem LCD dargestellt. Die Werte des DHT 11 werden als zahlen dargestellt, während die blaue LED weiterhin leuchtet, sobald es dunkel wird.

Schaltskizze

Hier findest du eine Skizze für die Verkabelung:



Der Programmcode

```
#include <DHT.h>; // Vorher hinzugefügte DHT Bibliothek einbinden
#include <LiquidCrystal_I2C.h> // Vorher hinzugefügte LiquidCrystal_I2C
Bibliothek einbinden
#include <Wire.h> // Wire Bibliothek einbinden

LiquidCrystal_I2C lcd(0x27,16,2); // Hier wird festgelegt um was für einen
Display es sich handelt.

#define DHTPIN 7 // an Pin 7 wird der DHT11 angeschlossen
#define DHTTYPE DHT11 // Hier wird definiert dass der DHT11 angeschlossen
wird
DHT dht(DHTPIN, DHTTYPE);
```

```
byte gluecklich[8] = { // Hier wird der glückliche Smiley erstellt
    B00000,
    B00000,
    B01010,
    B00000,
    B10001,
    B01110,
    B00000,
    B00000
};
```

```
byte traurig[8] = { // Hier wird der traurige Smiley erstellt
    B00000,
    B00000,
    B01010,
    B00000,
    B01110,
    B10001,
    B00000,
    B00000
};
```

```
// Hier werden die Variablen definiert
```

```
int feuchtigkeit;
int temperatur;
int eingang= A0;
int LEDblau = 10;
int Helligkeit = 0;
int Luftqualitaet;
```

```
void setup()
{
  Serial.begin(9600);

  pinMode (LEDblau, OUTPUT); //Der Pin mit der LED (Pin 10) ist jetzt ein
  Ausgang

  pinMode( 1 , INPUT);

  Serial.println("Temperatur und Feuchtigkeit Sensor");

  dht.begin();

  lcd.init(); //Im Setup wird das LCD gestartet

  lcd.backlight(); //Hintergrundbeleuchtung einschalten (lcd.noBacklight();
  schaltet die Beleuchtung aus).

  lcd.createChar(1, gluecklich); // Der glückliche Smiley hat jetzt die Nummer 1
  lcd.createChar(0, traurig); //Der traurige Smiley hat jetzt die Nummer 0
}

void loop()
{

  Luftqualitaet = analogRead(1); // Zuerst wird der Wert am Pin A0 ausgelesen...

  Serial.print("Der erfasste Wert lautet : ");
  Serial.println(Luftqualitaet, DEC); //... und anschließend im seriellen
  Monitor als Dezimalzahl ausgegeben.

  Serial.println();

  Helligkeit =analogRead(eingang); //Die Spannung an dem Fotowiderstand auslesen
  und unter der Variable „sensorWert“ abspeichern.
```

```
Serial.print("Helligkeit = " ); //Ausgabe am Serial-Monitor: Das Wort
„Sensorwert: „

Serial.println(Helligkeit); //Ausgabe am Serial-Monitor. Mit dem Befehl
Serial.print wird der Sensorwert des Fotowiderstandes in Form einer Zahl
zwischen 0 und 1023 an den serial monitor gesendet.

if (Helligkeit < 512 ) //Wenn der Sensorwert unter 512 beträgt...
{
digitalWrite(LEDblau, HIGH); //...soll die LED leuchten...
}

if (Helligkeit > 512 )
{
digitalWrite(LEDblau, LOW); //...soll sie nicht leuchten.
}

// Die Daten werden ausgelesen und es werden Variablen eigesetzt für
Temperatur (temperatur) und Feuchtigkeit (feuchtigkeit).
feuchtigkeit = dht.readHumidity();
temperatur = dht.readTemperature();

//Im seriellen Monitor soll der Text und die dazugehörigen Werte angezeigt
werden
Serial.print("Feucht: ");
Serial.print(feuchtigkeit);
Serial.print(" %, Temp: ");
Serial.print(temperatur);
Serial.println(" ° Celsius");

//Hier wird die Position des ersten Zeichens festgelegt. In diesem Fall
bedeutet (0,0) das erste Zeichen in der ersten Zeile.

lcd.setCursor(0, 0);
lcd.print("Temp: ");
lcd.print(temperatur);
```

```
lcd.print("C");

// In diesem Fall bedeutet (0,1) das erste Zeichen in der zweiten Zeile
lcd.setCursor(0, 1);
lcd.print("Feuch:");
lcd.print(feuchtigkeit);
lcd.print("%");

// In diesem Fall bedeutet (10,1) das zehnte Zeichen in der zweiten Zeile
lcd.setCursor(10, 1);
lcd.print("Qual:");

if (Luftqualitaet < 50)
{
    lcd.write(byte(1));
}

    if (Luftqualitaet > 50)
{
    lcd.write(byte(0));
}

    delay(1000); // Pause 1 Sek.
}
```