

Willkommen!

Und herzlichen Dank für den Kauf unseres Regentropfen Sensors!
Auf den folgenden Seiten gehen wir mit dir gemeinsam die ersten Schritte von der Einrichtung bis zur Ausgabe der Werte durch.

Viel Spaß!



Der Regentropfensensor misst die Leitfähigkeit der Oberfläche mithilfe elektrischer Spannung und schaltet bei einem einstellbaren Schwellenwert (LM393, Poti). Der Sensor besteht aus zwei Modulen, dem Messmodul und dem Schaltmodul. Bereits ein kleiner Tropfen reicht um die Veränderung des Wertes im Analog-Monitor, bzw -Plotter zu erkennen. Je mehr Regentropfen sich auf dem Sensor befinden desto höher wird der analoge Ausgabewert.

Die wichtigsten Informationen in Kürze

» **Abmessungen:** 55mm x 40mm x 1,2mm
31mm x 15mm x 7mm

» **Verbindung:**

3.3 - 5 V	VCC
GND	Masse
D0	DigitalOut
A0	AnalogOut

» **Temperaturbereich:** -65 - 150 °C

» Programmierung über Digital Pin, Analog Pin

Auf den nächsten Seiten findest du Informationen zur

» **Einrichtung der Hardware**

und eine Anleitung für

» **das Auslesen der Sensordaten.**

Diese Anleitung setzt voraus, dass du weißt, wie du Sketche auf einen Arduino hochlädst und den Serial Monitor verwendest!

Diese Anleitung setzt voraus, dass du weißt wie man einen Raspberry in Betrieb nimmt und die GPIOs nutzt. Nähere Informationen findest du im E-Book für den RaspberryPi

Alle Links im Überblick

LM393:

- » Datenblatt: <http://www.ti.com/lit/ds/symlink/lm393-n.pdf>
- » Bibliothek (Rpi): <https://github.com/WiringPi/WiringPi>

Programmieroberflächen:

- » Arduino IDE: <https://www.arduino.cc/en/Main/Software>
- » Web-Editor: <https://create.arduino.cc/editor>
- » Arduino-Erweiterung für SublimeText:
<https://github.com/Robot-Will/Stino>
- » Arduino-Erweiterung "Visual Micro" für Atmel Studio
oder Microsoft Visual Studio:
<http://www.visualmicro.com/page/Arduino-for-Atmel-Studio.aspx>

Arduino Tutorials, Beispiele, Referenz, Community:

- » <https://www.arduino.cc/en/Tutorial/HomePage>
- » <https://www.arduino.cc/en/Reference/HomePage>

Interessantes von AZ-Delivery

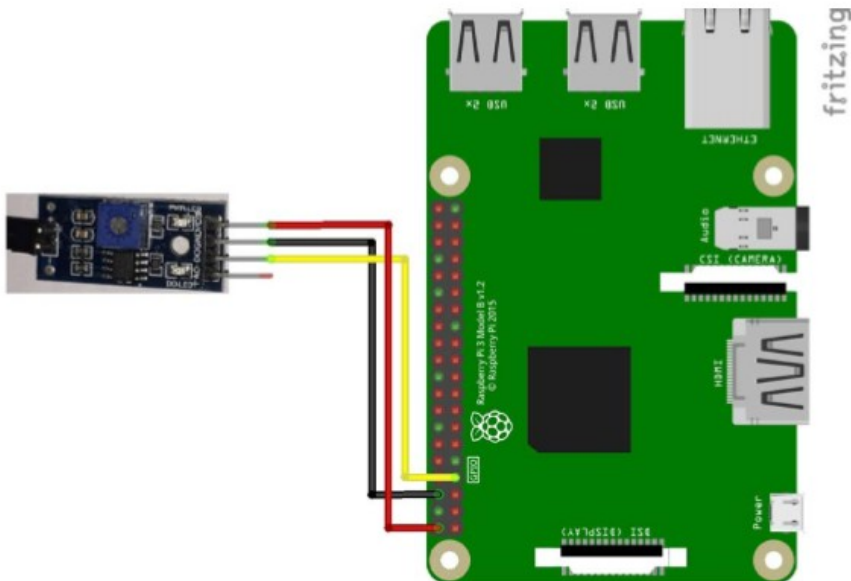
- » Arduino Zubehör:
<https://az-delivery.de/collections/arduino-zubehor>
- » AZ-Delivery G+Community:
<https://plus.google.com/communities/115110265322509467732>
- » AZ-Delivery auf Facebook:
<https://www.facebook.com/AZDeliveryShop/>

Einrichtung der Moduls

Dank dem verbauten Lm393 ist dieser Sensor sehr Controllerfreundlich, er kann mit 3.3 – 5V betrieben werden. Auf dem Schaltmodul kann man mithilfe des Potentiometers den Schwellenwert für den Schaltkontakt D0 einstellen, dieser Pin liefert ein High/Low Signal. Jeder Digitale Pin am Arduino kann verwendet werden. A0 gibt den Messwert mit einem Wert von 0 – 1023 aus, Sie können jeden Analog Pin am Arduino benutzen.

Aufgrund des Messprinzips dieses Sensors empfehlen wir im Praxiseinsatz jede Messung bei einem Zeitabstand von mind. 5 Minuten auszuführen, da ansonsten galvanische Prozesse den Sensor zerstören.

Anschlussbeispiel am Digital-Pin Raspberry:



Beispielcode am RaspberryPI:

installation der GPIO-Bibliothek

```
sudo apt-get install git
cd ~
git clone git://git
cd wiringPi
./build
gpio -v
gpio readall
```

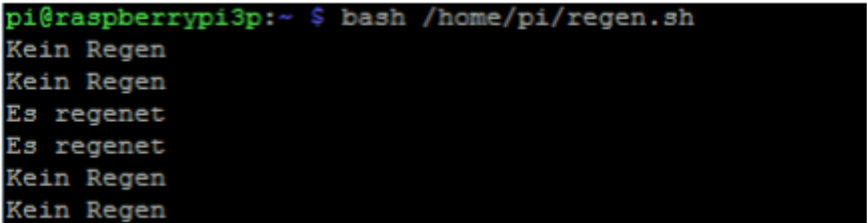
```
pi@raspberrypi3p:~/wiringPi $ gpio readall
```

+-----Pi 3+-----+												
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM		
		3.3v			1	2		5v				
2	8	SDA.1	IN	1	3	4		5v				
3	9	SCL.1	IN	1	5	6		0v				
4	7	GPIO. 7	IN	1	7	8	0	IN	TxD	15	14	
		0v			9	10	1	IN	RxD	16	15	
17	0	GPIO. 0	IN	0	11	12	0	IN	GPIO. 1	1	18	
27	2	GPIO. 2	IN	0	13	14		0v				
22	3	GPIO. 3	IN	0	15	16	0	IN	GPIO. 4	4	23	
		3.3v			17	18	0	IN	GPIO. 5	5	24	
10	12	MOSI	IN	0	19	20		0v				
9	13	MISO	IN	0	21	22	0	IN	GPIO. 6	6	25	
11	14	SCLK	IN	0	23	24	1	IN	CE0	10	8	
		0v			25	26	1	IN	CE1	11	7	
0	30	SDA.0	IN	1	27	28	1	IN	SCL.0	31	1	
5	21	GPIO.21	IN	1	29	30		0v				
6	22	GPIO.22	IN	1	31	32	0	IN	GPIO.26	26	12	
13	23	GPIO.23	IN	0	33	34		0v				
19	24	GPIO.24	IN	0	35	36	0	IN	GPIO.27	27	16	
26	25	GPIO.25	IN	0	37	38	0	IN	GPIO.28	28	20	
		0v			39	40	0	IN	GPIO.29	29	21	
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM		
+-----Pi 3+-----+												

Weiter geht es mit der Abfrage der Daten:

```
cd ~  
touch feuchte.sh  
nano feuchte.sh  
  
#!/bin/sh  
gpio -g mode 4 in  
while true do result="$( gpio -g read 4 )"  
if [ "$result" = "0" ]; then  
echo "Es regnet"  
fi if [ "$result" = "1" ]; then  
echo "kein Regen"  
fi sleep 2  
done  
  
bash home/pi/feuchte.sh
```

Die Ausgabe:

A terminal window with a black background and green text. The prompt is 'pi@raspberrypi3p:~' followed by '\$ bash /home/pi/regen.sh'. The output consists of six lines: 'Kein Regen', 'Kein Regen', 'Es regnet', 'Es regnet', 'Kein Regen', and 'Kein Regen'.

```
pi@raspberrypi3p:~ $ bash /home/pi/regen.sh  
Kein Regen  
Kein Regen  
Es regnet  
Es regnet  
Kein Regen  
Kein Regen
```

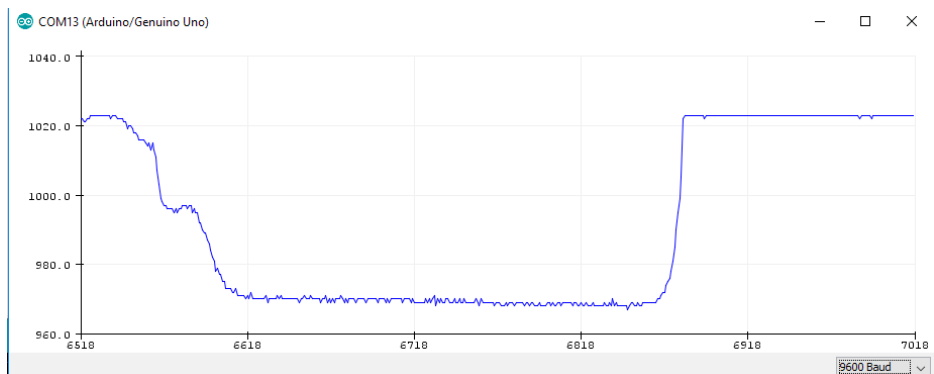
Anschlussbeispiel am Analog-Pin des Arduino:

Wir verbinden VCC mit 5V, GND mit GND und A0 mit A3 am Uno

der BeispielCode sieht so aus:

```
void setup() {  
  // initialize serial communication at 9600 bits per second:  
  Serial.begin(9600);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  // read the input on analog pin 3:  
  int sensorValue = analogRead(A3);  
  // print out the value you read:  
  Serial.println(sensorValue);  
  delay(1);    // delay in between reads for stability  
}
```

Den ausgegebenen Analog-Wert können wir am Seriellen Monitor oder Seriellen Plotter darstellen.



Du hast es geschafft! Herzlichen Glückwunsch!

Ab jetzt heißt es lernen und ausprobieren. Du weißt nun, wie du Regen mit einem Controller erkennen kannst. Jetzt kannst du versuchen, die Werte praktisch einzusetzen – vielleicht für eine automatische Jalousie? Diesen und noch mehr Hardware findest du natürlich in deinem Online-Shop auf:

<https://az-delivery.de>

Viel Spaß!

Impressum

<https://az-delivery.de/pages/about-us>