

Erhalte eine Nachricht aufs Smartphone, wenn die Post da ist



Gehst du ab und an vergeblich zum Briefkasten? Das ist jetzt vorbei! Dieses Projekt sendet dir eine Telegram-Nachricht aufs Smartphone, sobald der Briefträger die Post gebracht hat.

Hierfür verwenden wir einen Wemos D1 Mini und einen PIR-Bewegungsmelder. Du kannst aber natürlich auch einen anderen ESP8266 verwenden. Ebenso kannst du zum Beispiel ein Radarmodul verwenden, um die Bewegung im Briefkasten festzustellen.

Da wir im Briefkasten keine Steckdose haben, verwenden wir eine Powerbank für die Stromversorgung und versetzen den Microcontroller die meiste Zeit in Tiefschlaf. **Eine Bedingung gibt es allerdings:** Der D1 Mini bzw. ESP8266 muss sich mit deinem WLAN verbinden können – dieses sollte also bis zu deinem Briefkasten reichen.

Fortgeschrittene

1 – 2 Stunden

ca. 15€

Für dieses Projekt benötigst du (Mengen s. Beschreibung):

Angebot



[AZDelivery 1 x ESP32 D1 Mini NodeMCU WiFi Modul ESP 32 Bluetooth Internet Entwicklungsboard | kompatibel mit Arduino und kompatibel mit wemos d1 Mini | EIN kostenloses E-Book ist ebenfalls enthalten](#)

10,99 €

Angebot



[AZDelivery HC-SR501 PIR Bewegungssensor Bewegungsmelde Modul kompatibel mit Arduino und Raspberry Pi inklusive E-Book!](#)

□ PIR Modul zur Bewegungserkennung; □ Betriebsspannung 5V. Ideal für...

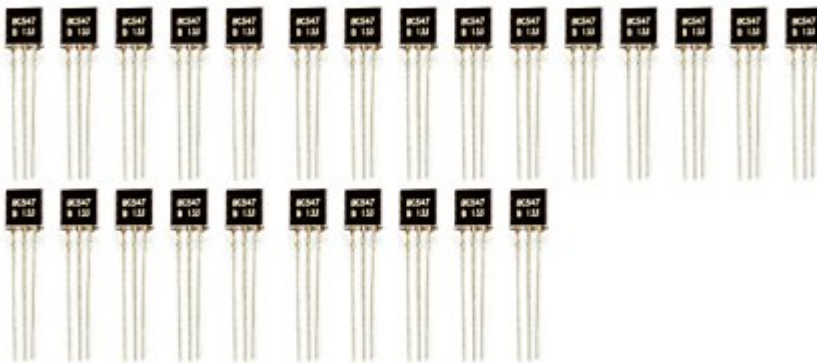
6,99 €



[Haobase 100x 1N4007 DO-41 Gleichrichter Diode 1A 1000V](#)

Sperrspannung: 1000 V.; Nennstrom: 1A.; Diodengröße: Ca. 4,5 mm (L) x 2,5 mm (D)...

4,90 €



[MissBirdler 25Stück BC547 TO-92 NPN 45V 0.1A Transistor für Arduino Raspberry Pi Prototyping](#)

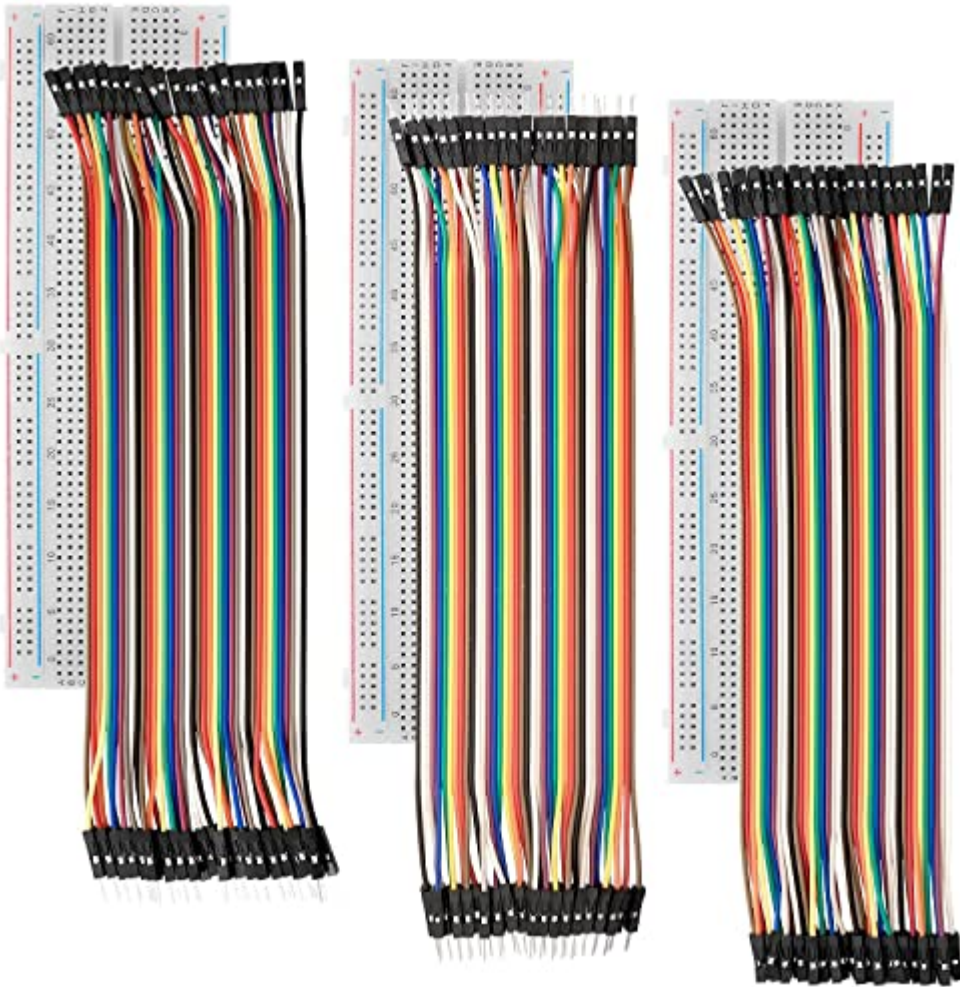
UNTERHALTUNGSELEKTRONIK; MissBirdler

4,99 €



Widerstand 1 K Ohm, 20 Stück, Metallschicht 0.6W 1% Metallfilm
Widerstände

3,30 €



[AZDelivery Breadboard Kit - 3X Jumper Wire m2m/f2m/f2f + 3er Set MB102 Breadboard kompatibel mit Raspberry Pi inklusive E-Book!](#)

□ Steckbrett für schnellen Aufbau elektronischer Schaltungen mit 830 Kontakten.; □...

10,99 €

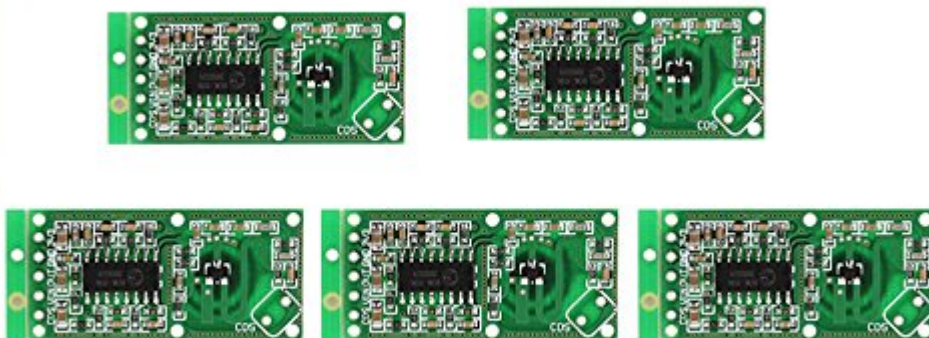
Alternativ kannst du auch einen NodeMCU ESP8266 und/oder ein Radarmodul verwenden:



[AZDelivery NodeMCU Amica Modul V2 ESP8266 ESP-12F WiFi - Node MCU ESP 8266 WiFi Development Board mit CP2102 kompatibel mit Arduino - inklusive Installationsanleitung als E-Book](#)

□ Maße (LxBxH): 48 x 26 x 13 mm

8,49 €



[WINGONEER 5PCS Mikrowellen-Radarsensor RCWL-0516 Schalter-Modul menschlicher Induktions-Brett-Detektor](#)

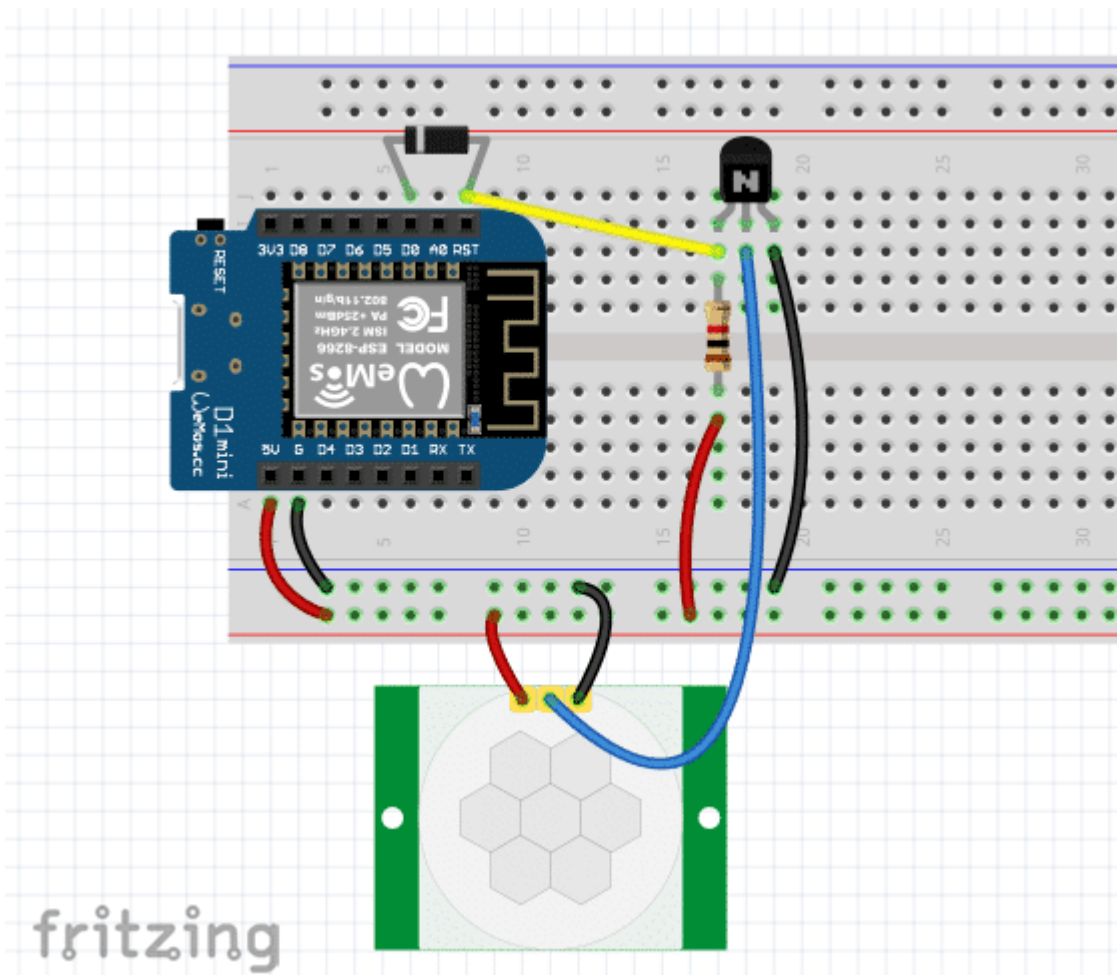
10,67 €

Vorbereitungen

Ein paar Dinge musst du eventuell vorab erledigen – wie, erfährst du in diesen Tutorials: [Richte dir zunächst einen Telegram-Bot ein](#), um Nachrichten vom D1 Mini auf deinem Smartphone erhalten zu können. Außerdem musst du deinen [D1 Mini bzw. ESP8266 in deiner Arduino IDE verfügbar machen](#), um ihn von dort aus programmieren zu können.

Das Projekt aufbauen

Montiere zunächst alle Teile wie folgt auf deinem Breadboard.



Aufbau auf dem Breadboard

Schauen wir zunächst auf die Verbindung der Pins **D0** und **RST**: Hiermit kannst du deinen Microcontroller aufwecken, wenn von D0 ein Signal an RST gesendet wird. In diesem Projekt soll

dieses Signal jedoch vom Bewegungssensor kommen. Damit es also nicht fälschlicherweise doch vom Pin **D0** gesendet wird, verwenden wir eine Diode, die Strom von diesem Pin blockiert (achte beim Einbau auf die Richtung des weißen Rings). **Falls du keine Diode zur Hand hast, sollte das Projekt behelfsweise auch mit einer einfachen Drahtbrücke funktionieren.**

Noch ein Hinweis: Solange die beiden Pins miteinander verbunden sind, kannst du keinen Sketch hochladen. Hierfür musst du die Verbindung zunächst trennen.

Das Wecksignal für den D1 Mini

Du weckst deinen Microcontroller also über ein HIGH am Reset-Pin auf. Hierfür benötigst du allerdings einen eindeutigen Wechsel von LOW auf HIGH. Wenn du deinen PIR-Sensor direkt an den Reset-Pin anschließt, erhältst du kein korrektes Signal, was zu mehreren Resets hintereinander führen kann – oder einfach nicht funktioniert, wie in unseren Tests.

In Anlehnung an [dieses Tutorial](#) verbauen wir einen **NPN-Transistor (BC547)**, der uns dabei hilft, ein **eindeutiges Signal vom PIR-Sensor an den Reset-Pin des D1 Mini zu übertragen**. Sollte das bei dir trotzdem zu ungewollten Neustarts (also Aufwachen) führen, wird im verlinkten Tutorial auch noch ein Aufbau mit einem Kondensator beschrieben, der dir weiterhelfen könnte.

Mehr über NPN-Transistoren erfährst du in [diesem Artikel](#). [In diesem Tutorial erfährst du mehr über den Bewegungssensor HC-SR501.](#)

Der Sketch

Kommen wir also zum Code. Zunächst benötigst du ein paar Bibliotheken, die du zu Beginn des Sketchs einbindest. Falls noch nicht geschehen, installiere im Bibliotheksmanager die jeweils neueste Version.

```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
```

Anschließend folgen einige Definitionen und Konstanten für deine Zugangsdaten zum WLAN und zu deinem Telegram Bot. Zur Erinnerung: Wie du die benötigten Daten für Telegram erhältst, erfährst du im oben genannten Tutorial.

```
//Deine WLAN-Zugangsdaten
const char* ssid = "NETWORK";
const char* password = "PASSWORD";

//Dein Bot-Token
#define botToken "BOT-TOKEN" // den Bot-Token bekommst du vom
Botfather)

//Deine User ID
#define userID "USER-ID"
```

Im letzten Schritt vor der Setup-Funktion erstellst du eine Instanz von **WiFiClientSecure** und initialisierst mit dieser die Verbindung zum **Telegram-Bot**.

```
WiFiClientSecure client;
UniversalTelegramBot bot(botToken, client);
```

Die Setup-Funktion

Im Sketch für dieses Projekt benötigst du nur die Setup-Funktion, die nach dem Start des Wemos D1 Mini abläuft. Hier startest du die Internet-Verbindung und sendest die Nachricht, dass die Post da ist, an dein Smartphone (also an deinen Telegram-Bot). Zuletzt legst du deinen Microcontroller wieder schlafen.

Sobald er ein neues Signal von deinem Bewegungsmelder erhält, wacht er wieder auf und durchläuft das Setup erneut. **Wie du siehst, benötigst du also keinen Loop.**

Zuerst konfigurierst du den **client**, den du oben erzeugt hast, startest den Seriellen Monitor und gibst dort die Nachricht aus, dass dein D1 Mini aufgeweckt wurde:

```
client.setInsecure();  
Serial.begin(115200);  
Serial.println("Awake");
```

Anschließend stellst du die Verbindung zu deinem WLAN-Netzwerk her:

```
Serial.print("Connecting to WiFi: ");  
Serial.println(ssid);
```

```
WiFi.mode(WIFI_STA);  
WiFi.begin(ssid, password);
```

```
while (WiFi.status() != WL_CONNECTED) {  
    Serial.print(".");  
    delay(500);  
}
```

```
Serial.println("");  
Serial.println("Connected!");
```

Zuletzt sendest du eine Nachricht an deinen Telegram-Bot mit der Funktion **bot.sendMessage()** – wartest 2 Sekunden und legst den D1 Mini wieder schlafen:

```
bot.sendMessage(userID, "Your mail has arrived!", "");  
  
delay(2000);  
Serial.println("Going to sleep");  
ESP.deepSleep(0);
```

Für den Tiefschlaf verwendest du die Funktion **ESP.deepSleep()** mit dem Parameter **0**. Diese Null sorgt dafür, dass dein Microcontroller **ohne zeitliches Limit** in den Schlaf versetzt wird – das heißt er wird erst wieder durch ein Signal vom

Bewegungssensor aufgeweckt.

Und das war es auch schon – eigentlich ganz einfach. Hier nun der gesamte Sketch zum Rauskopieren.

```
//Libraries
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>

//WiFi Credentials
const char* ssid = "NETWORK";
const char* password = "PASSWORD";

//Telegram Bot Credentials
#define botToken "BOT-TOKEN"
#define userID "USER-ID"

WiFiClientSecure client;
UniversalTelegramBot bot(botToken, client);

void setup() {
  client.setInsecure();
  Serial.begin(115200);
  Serial.println("Awake");

  //Connecting to WiFi
  Serial.print("Connecting to WiFi: ");
  Serial.println(ssid);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  Serial.println("");
  Serial.println("Connected!");

  bot.sendMessage(userID, "Your mail has arrived!", "");
```

```
    delay(2000);  
    Serial.println("Going to sleep");  
    ESP.deepSleep(0);  
}  
  
void loop() {  
}
```

Wie geht es weiter?

Du hast nun einen Post-Melder mit dem Bewegungssensor HC-SR501. Je nachdem, wie dein Briefkasten beschaffen ist, kann ein Radar-Modul die bessere Wahl sein.

Vielleicht fallen dir aber auch noch ganz andere Lösungen ein, um zu erkennen, dass dein Briefkasten geöffnet wurde. Nerdiger und ambitionierter wäre zum Beispiel eine Variante, die KI nutzt. Auf Pollux Labs kannst du die Grundlagen von [Künstlicher Intelligenz auf Microcontrollern](#) lernen.