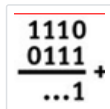


Topcase001

dinsdag 13 oktober 2020 20:09



Deel 1 Alusta systeem ombouwen (Wemos klaarmaken en inbouwen)

Wat gaan we doen:

- 1: in iedere roosterunit gaan we een klein Arduino printje plaatsen (ESP8266 Wemos D1) die geeft een IR infrarood signaal en daarom hoeven we ook niet aan de roosterprint te sleutelen.
- 2: we maken van een RaspberryPi een router En zetten er ook Domoticz op (beide gratis van internet) en daarmee kunnen we de roostersunits helemaal zelf bedienen, automatisch a.d.h. van de CO2 waardes of handmatig.
- 3: we passen de bestaande MV aan en maken gebruik van het bestaande Flow-sensor kastje (blauwe kastje) waarin we een Sonoff plaatsen die ook geschakeld gaat worden door ons systeem. Keurig en veilig dus.
- 4: de Vent-o-Staat heb je alleen nog maar nodig voor de voeding en we schakelen de bediening uit.

Je kunt dit project per roosterunit doen, terwijl het oude systeem gewoon nog draait (als die nog draait 😊) en ben je bij het laatste rooster aangekomen dan kun je de MV overnemen met een Sonoff en dan ben je van die blauwe flow-sensor af. Een mooie bijkomstigheid is dat als er in de oude situatie 1 unit kapot is, het hele bestaande systeem niet werkt (piepen) en in ons geval blijven de andere roosters nog gewoon werken! (zonder piepen)

Het is erg leuk als je een duimpje geeft als je dit een leuk project vindt. Tot nu toe zie ik weinig duimpjes en vraag mij af of ik dit doe voor vier mensen? Is het dan wel handig om het op deze manier te doen?

Hier kun je een klein filmpje zien hoe het werkt (kwaliteit is niet zo goed maar het idee is duidelijk). Als je gaat douchen/koken druk je op een wandknop of digitaal, tablet etc, en gaan de, (in mijn geval) de vier roosters open en na de tijd die je zelf ingegeven hebt, gaat de ventilator uit en gaan de vier roosters weer dicht. Alleen waar de CO2 te hoog is in een bepaalde ruimte, blijft het rooster open totdat het PPM gehalte weer op peil is en gaat die rooster alsnog dicht. Alles geheel automatisch.

[YouTube: Voorbeeld Alusta](#)

Benodigde materialen:

- 1: Wemos D1 mini. (net zoveel als je roosters hebt + het aantal CO2 meters) Kost een paar euro per stuk (China?) Dus als je vier rooster hebt en drie CO2 meters, heb je 7 ESP8266 Wemos D1 nodig.
- 2: IR led, zelfde hoeveelheid als je roosters die heb je al voor €0,25 per stuk (China?)
- 3: Weerstand 180 Ohm, zelfde hoeveelheid als je roosters
- 4: Wat krimpkous
- 5: Wat draadjes met pinnetjes (als je met Arduino werkt heb je er vast een paar liggen)
- 6: Raspberrypi vanaf versie 3 met WIFI, SD kaartje en goede voeding! Voldoende Amp! Belangrijk!

We beginnen: de ESP8266 Wemos D1 gereed maken

We gaan aan 1 poot van de IR (heeft twee pootjes en we kunnen moeilijk doen dat het een Anode heeft en een Kathode, maar als hij niet werkt hoeft je hem alleen maar om te draaien) een weerstand van 180 Ohm met draadje solderen en doe daar wel een krimpkous over heen.

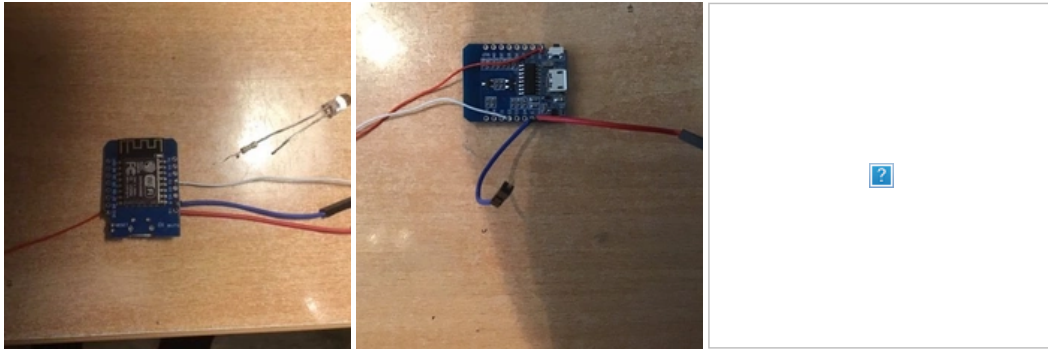
De andere poot van de IR voorzien we ook van een draadje en daaroverheen ook weer krimpkous en deze beide draadjes gaan we aan de Wemos solderen. Een aan de 3.3 V en de ander aan D2 Geen paniek ik doe overall een voorbeeld foto bij. Ook hoeft je geen ster in het solderen te zijn. Google is je vriend 😊

De Wemos moet wel gevoed worden en daarom solderen we er twee draadjes aan met aan de uiteinde een pinnetje die we later zo in de unit prikken.

Rood = plus en die soldeer je aan de 5V

Blauw = min en die soldeer je aan de G = ground (zie voorbeeld foto)

Dat is alles voor 1 rooster!



Noot! Voor alles geldt hier: als je aanbevelingen, verbeteringen etc hebt, graag doorgeven. Ik ben geen programmeur en hoewel het goed werkt is het "gelikter maken" ook leuk en leerzaam voor anderen.

Dan wordt het tijd om het programma in de Wemos te zetten. Ik ga er even van uit dat je daar al wat van weet en anders wat info gaat zoeken op het internet. Onderstaande schets kopiëren en plakken in de Arduino IDE, compileren en updaten in de esp8266. Als je dit gedaan hebt moet je wel het MAC en IP noteren die je later nodig hebt om deze in de router te zetten (statisch) Mocht je op problemen stuiten gaan we daar later natuurlijk op in, maar in dit geval even zo laten anders wordt het een te groot document (het is zo al groot). Misschien ga ik verder op een eigen website en kan ik het uitgebreider doen en groter publiek bereiken. Maar vooralsnog op deze manier.

/* IRremoteESP8266: IRsendDemo - demonstrates sending IR codes with IRsend.

Version 1.1 January, 2019

Based on Ken Shirriff's IRsendDemo Version 0.1 July, 2009,

Copyright 2009 Ken Shirriff, <http://arcfn.com>

Aansluitingen: zie beschrijving op Tweaker

Bewerkt door Kees Krijnberg t.b.v. Alusta roosterunit zonder in het rooster te hoeven solderen of te desolderen.

Ondanks dat het goed werkt graag hoor ik verbeteringen om het geheel netter te maken. Wat is overbodig?, wat is teveel?, waar moet wat staan? etc.

*/

#include <Wire.h> // hier kunnen waarschijnlijk nog het e.e.a. weggelaten worden

#include <ESP8266WiFi.h>

#include <IRremoteESP8266.h>

#include <ESP8266DNS.h>

#include <WiFiUdp.h>

#include <Arduino.h>

#include <IRsend.h>

const char * domoticz_server = "10.3.141.1"; // Raspap server waar Domoticz op staat

int port = 8080;

//int idx = 1086; // idx uit Domoticz, als app aangemaakt is

WiFiServer server(80);

WiFiClient client; // Wifi status

// zelfstandige Router: www.raspap.com Deze kan draaien alleen in huis en/of van buitenaf benaderbaar met de telefoon tablet etc.

//const char* ssid = "raspi-webgui"; //connect to SSID raspap Zie beschrijving raspap

//const char* password = "ChangeMe"; //wachtwoord raspap (wel veranderen als je naar buiten gaat)

const uint16_t klrLed = 4; // voor de Wemos ESP8266 (D2)

IRsend irsend(klrLed);

```

void setup()
{
  Serial.begin(9600, SERIAL_8N1);
  Serial.println("Booting");
  irsend.begin();
  WiFi.mode(WIFI_STA);
  WiFi.hostname("Wemos1"); // Geef de juiste naam. In router te zien
  WiFi.begin("raspi-webgui", "ChangeMe"); // Verbinden met netwerk

  while (WiFi.waitForConnectResult() != WL_CONNECTED) {
    Serial.println("Connection Failed! Rebooting...");
    delay(5000);
    ESP.restart();
  }
  Serial.println("Is verbonden met WIFI"); // Verbonden in seriële monitor
  Serial.print("SSID: "); // Access point (SSID) in seriële monitor
  Serial.println(WiFi.SSID());
  Serial.print("IP: "); // Ip adres in seriële monitor (nodig om in raspap statisch
  // te maken)
  Serial.println(WiFi.localIP());
  Serial.print("ESP MAC: "); // MAC adres in seriële monitor (nodig om in raspap
  // statisch te maken)
  Serial.println(WiFi.macAddress());
  delay(500);

  Serial.print("Start server..."); // Start server
  server.begin();
  Serial.println("OK!");
}

void loop() {
  WiFiClient client = server.available(); // Controle of de client verbonden is
  if (!client) {
    return;
  }
  Serial.println("new client"); // Wacht totdat de cliënt data zend
  while (!client.available()) {
    delay(1);
  }
  String request = client.readStringUntil('\r'); // Lees de informatie van het verzoek
  Serial.println(request);
  client.flush();

  if (request.indexOf("rechts=AAN") != -1) // Rooster open
  {
    irsend.sendRC5(0xFFFF, 12); // Alusta IR code voor open
    Serial.print(" Alusta code OPEN"); // tekst in seriële monitor (is niet perse nodig)
    // delay(5000);
  }
  if (request.indexOf("rechts=UIT") != -1) // Rooster stop
  {
    irsend.sendRC5(0x7FFF, 12); // Alusta IR code nogmaals Open
    Serial.print(" Alusta code nogmaals OPEN = UIT");
    // delay(2000);
  }
  if (request.indexOf("links=AAN") != -1) // Rooster Dicht
  {
    irsend.sendRC5(0x7FE, 12); // Alusta IR code dicht
    Serial.print(" Alusta code DICHT");
    // delay(5000);
  }
  if (request.indexOf("links=UIT") != -1) // Rooster stop
  {
    irsend.sendRC5(0xFFE, 12); // Alusta IR code nogmaals dicht
    Serial.print(" Alusta code nogmaals DICHT UIT");
    // delay(2000);
  }
  delay(1); // Voor de rust
  Serial.println("Client disconnected");
}

```

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

```
/*
```

```
De HEX codes van de afstandsbediening zijn:
```

```
open = 0xFFFF nogmaals = uit = 0xFF
```

```
dicht = 0x7FE nogmaals = uit = 0xFFE
```

```
auto = 0x7FD nogmaals = uit = 0xFFD
```

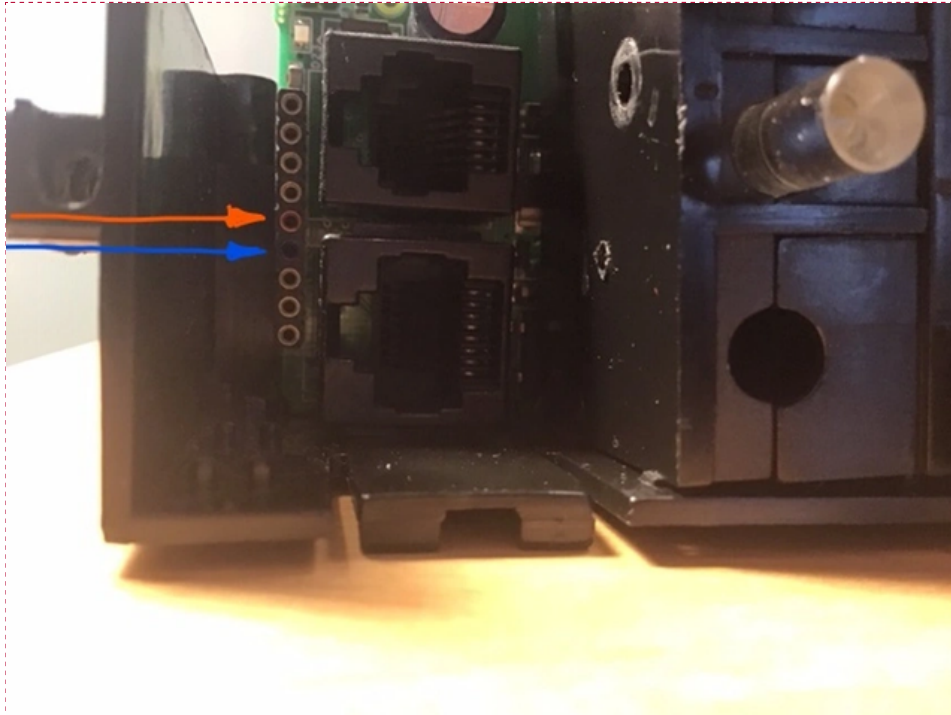
```
prog. = 0x7FC nogmaals = uit = 0xFFC
```

```
*/
```

```
}
```

```
*****
```

Als je het programma in de Wemos D1 hebt gezet ben je klaar en kun je hem in de roosterunit plaatsen.
Zie voorbeeld foto's.

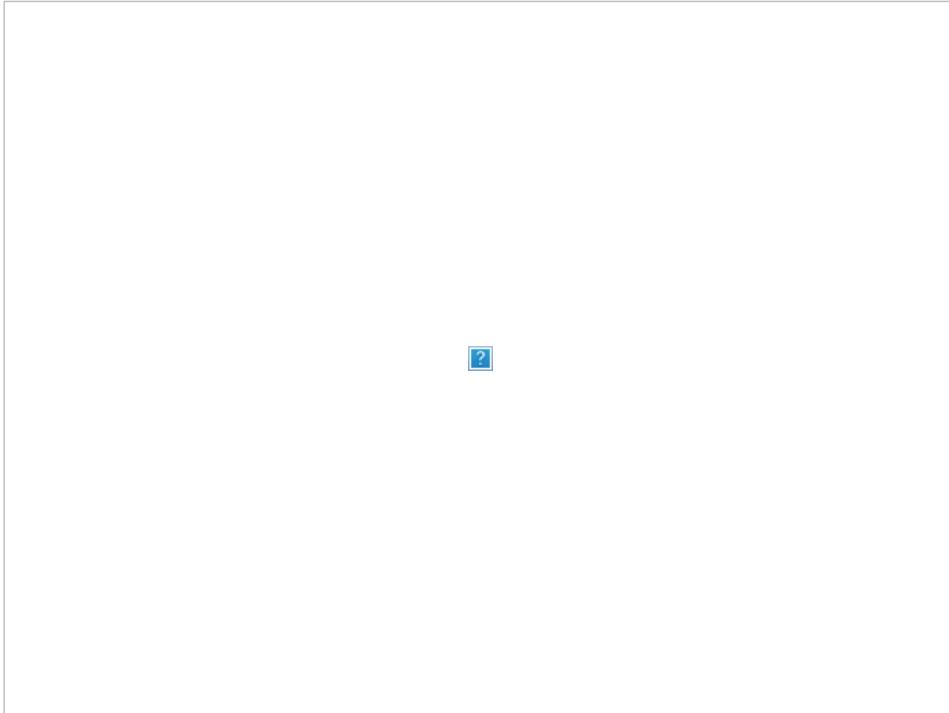


Aan de linkerkant zie je een strip waarin je pinnetjes kunt steken. Op (4) komt de min(blauw) en (5) de rode. Ik heb ze voor de foto gekleurd met een stift en als je goed kijkt kun je de kleuren zien.

Zorg voor isolatie tussen de strip en het Wemos bordje. Niet helemaal inpakken want dan kan het bordje zijn warmte niet kwijt. Dus simpel het zakje waarin het bordje zit tussen de strip en het bordje doen.

Nu moet het onderste lensje uit de unit getrokken worden. Dat kan zonder dat er iets kapot gaat. Het lijkt of het ergens aan vast zit, maar dat is een rubber ringetje die van het lensje afstroopt. Dat ringetje zit daar omdat het lensje er niet zo uit kan vallen. Dus gewoon met een tangetje eruit trekken. In het gat wat dan vrijkomt gaat de IR led in. De Wemos kun je nu in de Roosterunit plaatsen.

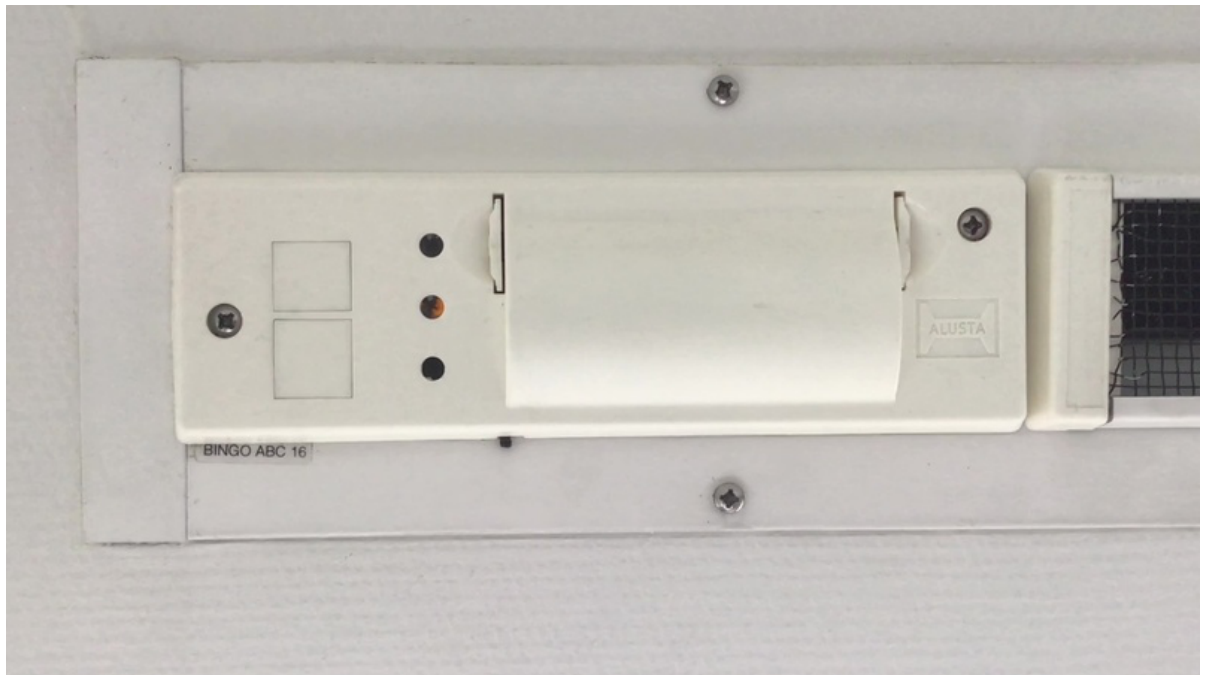
Let wel op! Doe isolatie!! tussen de strip en de Wemos zodat er geen sluiting kan komen. Niet inpakken anders kan hij zijn warmte niet kwijt!



Nu werkte de IR led op het breadboard wel, maar als de IR led te dicht op de ontvangende IR komt, werkt het niet of krijgt hij verkeerde codes door. Niet schrikken. Het is een fenomeen wat ik niet ken, maar ik kwam er achter dat als deze twee te dicht op elkaar zitten het niet goed werkt. Van het antistatische zakje heb ik een stukje afgeknipt en er dubbel voor de lens gedaan en dat werkt goed. Het signaal is te sterk anders of? Wie weet mag het zeggen. Zoals het nu zit werkt het ook erg goed en geen enkel fout of mis signaal. Op de foto heb ik het laten zitten, zodat je kunt zien wat ik gedaan heb.



Knip een klein stukje uit de voorklep om ruimte te maken voor de draadjes en het kapje kan er op. Klaar is Kees. Ik zelf programmeer de esp's over the air (OTA) en kan zo later eventuele modificaties programmeren vanaf de laptop en hoeft ze er niet uit te halen.



Oke, super dat je dit gedaan hebt!

1e unit heb je klaar. Je kunt hem alleen nog niet aansturen omdat de Wemos D1 in wil loggen op een router en die router heb je nog niet draaien. Zodra je die hebt draaien (zie deel 2) wordt het leuk want je kunt dan al het rooster vanaf je computer of telefoon open en dicht zetten (zelf in het nog bestaande draaiende systeem)

Nu is het tijd om een RaspberryPi te gaan kopen of je hebt er nog 1 liggen. Deze moet natuurlijk wel Wifi aan boord hebben dus je moet een RaspberryPi vanaf versie 3 hebben of hoger. Een SD kaartje om de programma's op te zetten. Als je niet op de centen hoeft te letten kun je meer Mb nemen en/of een hogere versie. Ook het SD kaartje is best belangrijk minimaal 16 Mb, categorie (10) <https://www.budget-hardware.nl> heb ik ontzettende goede service en levert tegen Chinese prijzen. Soms krijg ik meer dan besteld echt waar.



RaspberryPi model 3B + Dit is de computer die het hele Vent-O-staat systeem over neemt en doet wat JIJ wilt en niet andersom 😊

[nieuws: Raspberry Pi Foundation brengt krachtiger 3B+-model met 5GHz-wifi uit](#)

Heb je inmiddels de RaspberryPi? Mooi, dan gaan we nu de SD kaart in orde maken en er wat programma's op zetten. Als je dit voor het eerst doet kost dat wel even wat tijd, maar het is ook een leuke bezigheid en