



Veelzijdig weerstation

ACTUELE WEERRAPPORTEN EN WAARSCHUWINGEN

Krijgen we een onstuimige herfst of worden we verwend met een zwoele nazomer? Lekker lang op het terras kunnen zitten zou wel leuk zijn, maar misschien moeten we het doen met zo'n natte bedoening die aanhoudt tot de winter. Er is weinig aan te veranderen, maar vanaf nu weet je wél precies waar je aan toe bent, dankzij je eigen, zelfgemaakte weerstation.

■ Martijn Overman



BOODSCHAPPENLIJSTJE

- Behuizing (€ 2,-, bijvoorbeeld een pillendoosje van Kruidvat)
- Bijvoorbeeld bij www.rotor.eu:
 - Passieve piëzo-buzzer (€ 2,-)
- Bijvoorbeeld bij www.martoparts.nl:
 - NodeMCU-module (€ 9,95)
 - Oled-beeldscherm (€ 6,95)
- Bijvoorbeeld bij www.conrad.nl:
 - Diffuse leds, 6 stuks (€ 0,60)
 - Trekontlasting M10 (€ 1,-)
 - Experimenteerprintplaatje 80 × 50 mm (€ 2,50)
 - Schroefterminal of klemschroefblok 2-polig (€ 0,21)
 - Draad 0,20 mm², 2 meter (€ 0,60)
 - Usb-oplader 5 V, 1,5 A (€ 5,50)
 - Usb-kabel (€ 1,63)
- Gereedschap: soldeerbout en soldeertin, zijknijptang, stanleymes, vijl.

■ Totale kosten: **ca. € 33,-**



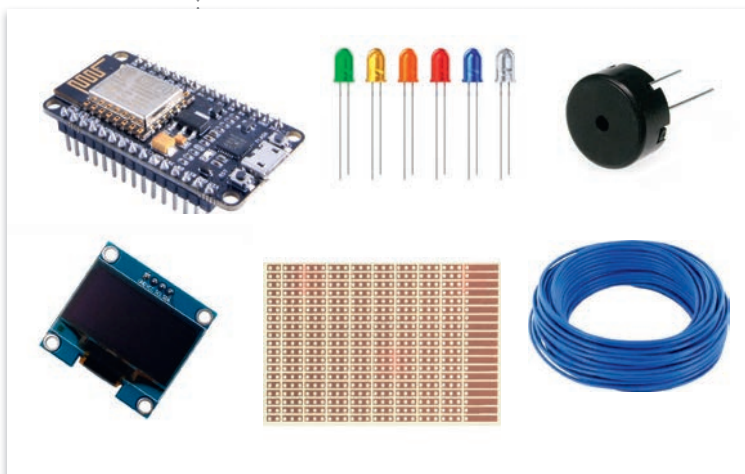
Het weerstation werkt niet met eigen sensoren en meetinstrumenten, maar haalt elke vijf minuten de nieuwste gegevens op van het dichtstbij gelegen KNMI-meetstation en van Buienradar. Het toont de actuele weersituatie op een oled-schermje en waarschuwt als het gaat regenen of vriezen. Ook bij een voor jouw regio uitgegeven weeralarm krijg je een melding. Gekleurde leds geven de weersituatie aan: bij groen is er niets aan de hand, blauw is regen, wit is vorst en bij een weeralarm brandt geel, oranje of rood.

01

Hardware

De schakeling is gebouwd rond een NodeMCU, een ontwikkelbordje dat is gebaseerd op een ESP-wifi-module. De overige onderdelen zijn een oled-beeldschermje, zes leds, drie weerstanden en een piëzo-luidspreker. Het beeldscherm en de luidspreker kunnen rechtstreeks op de module worden aangesloten, voor de leds zijn voorschakelweerstanden nodig. Normaal gesproken krijgt elke led zijn eigen weerstand, maar omdat er van groen, geel, oranje en rood altijd slechts één led tegelijk brandt (ter indicatie van een weeralarm) volstaat voor die vier leds één weerstand. Blauw en wit kunnen op ieder moment branden, dus die krijgen wel elk een eigen weerstand.

Het beeldscherm communiceert met de ESP-module



01 De bonte verzameling componenten.

door middel van een zogenoemde i2c-bus. Het voordeel daarvan is dat er slechts twee verbindingen nodig zijn voor de communicatie en twee voor de voeding, dus vier in totaal. Mede daardoor is de schakeling voor zijn veelzijdigheid erg eenvoudig van opzet.

02

Werking

We leggen even uit hoe een en ander gaat werken als je alles eenmaal af hebt. Zodra de schakeling wordt aangezet, probeert de

NodeMCU verbinding te maken met je wifi-netwerk. Als dat gelukt is, wordt achtereenvolgens informatie opgehaald over de hoeveelheid regen, het weerrapport en een eventueel weeralarm. De oorspronkelijke webpagina's van de te raadplegen webpagina's passen niet in het beperkte geheugen van de module. Door via een tussenstap (een serverscript dat draait bij ons moederbedrijf Reshift) alleen de gegevens van het voor jou relevante weerstation te tonen, is dat probleem opgelost.

Het compacte beeldscherm toont de windrichting, de windkracht, de temperatuur en een pictogram van de weersituatie. Standaard brandt de groene led. Als er regen aankomt, gaat de blauwe led aan, bij vorst brandt de witte led en bij een weeralarm brandt geel, oranje of rood. Twee piepjes geven in dit soort situaties aan dat er wat aan de hand is.

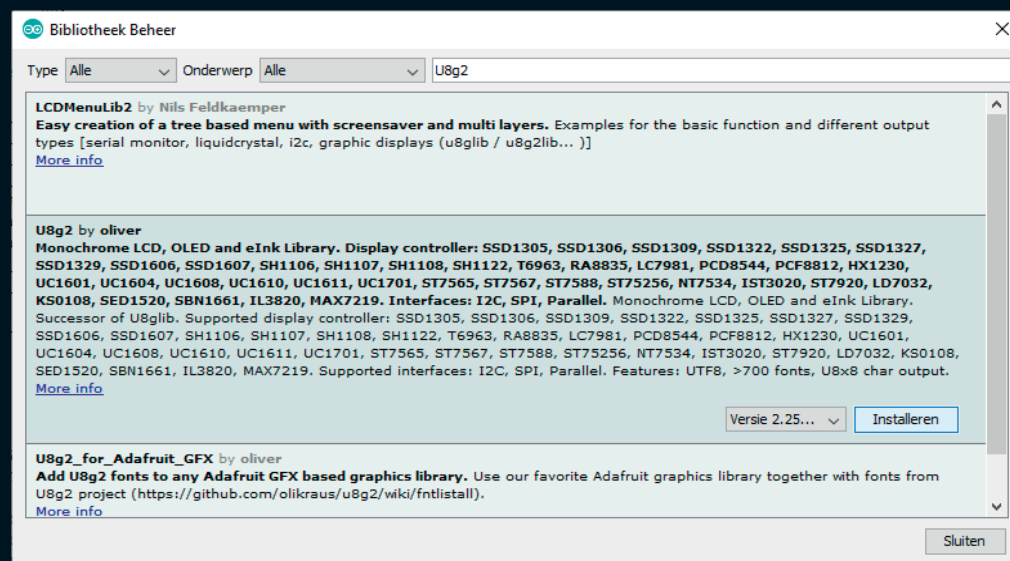
03

Uitleg bij de code

De code kun je downloaden via www.tiny.cc/weerst. We leggen hier eerst uit hoe de code werkt, voordat we er daadwerkelijk mee aan de slag gaan (vanaf paragraaf 'Code uploaden en testen'). Je kunt de code wel alvast »

OLED-BIBLIOTHEEK

Om pictogrammen en lettertypen op het beeldscherm te kunnen tonen, is een extra bibliotheek nodig. Klik in Arduino IDE op **Schets / Bibliotheek gebruiken / Bibliotheken beheren**. Typ **U8g2** bij **Filter je zoekresultaten** en klik onder het item **U8g2 by oliver** op **Installeren**. Herstart daarna Arduino IDE. Je kunt nu in je code de nieuwe bibliotheek gebruiken met de regel `#include <U8g2lib.h>`



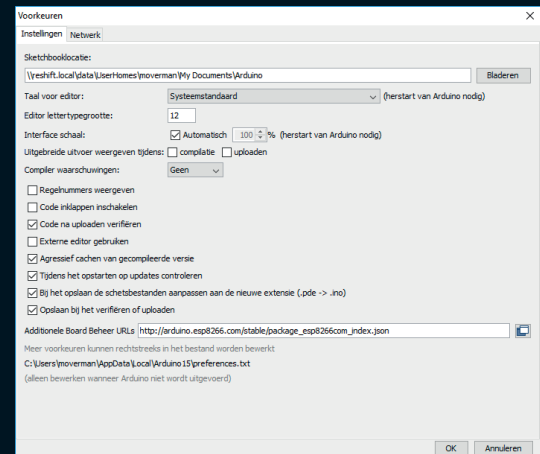
Voeg de benodigde bibliotheek toe.

ONTWIKKELOMGEVING

De ESP-module is het gemakkelijkst te programmeren met de Arduino-ontwikkelomgeving IDE. Deze kun je downloaden via www.tiny.cc/arduinoosoft. Omdat Arduino IDE niet primair voor deze module is bedoeld, zul je nog wat extra benodigde onderdelen moeten installeren. Klik daarvoor op **Bestand / Voorkeuren** en vul op het tabblad **Instellingen** bij **Additionele Board Beheer URLs** deze url in:

http://arduino.esp8266.com/stable/package_esp8266com_index.json

Voeg de module daarna toe via **Hulpmiddelen / Board / NodeMCU 1.0 (ESP-12E Module)**. Sluit de module aan via een usb-kabel en selecteer in Arduino IDE de juiste poort (**Hulpmiddelen / Poort**, kies de com-poort met het hoogste nummer). Als alles goed is gegaan, is je opstelling nu klaar om met programmeren te beginnen. Mocht de module onverhoopt niet door je pc of Mac worden herkend, dan kun je via <https://arduino.reshift.nl/hubbit.html> drivers vinden waarmee het zeker moet lukken.



Installeer de module via de optie **Additionele Board Beheer URLs**.

bekijken met Kladblok of (handiger) Notepad++.

Het programma begint met het laden van vier bibliotheken via de opdracht **include: ESP8266WiFi.h** voor alle netwerkactiviteit, **Wire.h** voor de communicatie met het beeldscherm en **U8g2.h** voor het tonen van grafische informatie op het scherm (zie kader 'Oled-bibliotheek'). Dankzij deze kant-en-klare code van derden kunnen we ons direct richten op de functionele aspecten van de schakeling. De vierde bibliotheek (**weathericons.h**) bevat de weerpictogrammen en is onderdeel van ons programma.

Het programma begint met het declareren van diverse variabelen en een lange lijst constanten. Daarvan hoeft je er maar vijf aan te passen aan jouw situatie, zie kader 'Constanten'. Via de constanten

voor de luidspreker en de leds kun je deze onderdelen in de code aanroepen met hun omschrijving in plaats van een lastig te onthouden pinnummer. Met de constante **Wachttijd** stel je in hoe vaak de weerinformatie wordt opgehaald, standaard eens per vijf minuten. **Periode** bepaalt hoever er vooruitgekeken wordt of het regent. Standaard staat hier **1**, wat staat voor vijf tot tien minuten vooruit. Hoewel het handig lijkt om hier een langere periode te kiezen, blijkt de nauwkeurigheid van de voorspelling dan sterk af te nemen.

De strings aan het einde van het lijstje constanten zijn de codes waarmee het KNMI op zijn website een weeralarm aangeeft. Hierin herken je in de waarden **red**, **orange**, **yellow** en **green** de mogelijke omstandigheden.

03 Een fragment van de code in Arduino IDE.



Onder **Setup** worden de pinnen als uitgang gedefinieerd en met het wifi-station verbonden. Tevens wordt het domein van de host voor de regenvoorspelling opgehaald en toegekend aan de variabele **raindomain**, die verderop in de code wordt omgezet in **rainhost**. Alle leds en de luidspreker gaan ten slotte beurtelings kort aan om er zeker van te zijn dat ze werken.

CONSTANTEN

Om het programma te kunnen gebruiken, zul je enkele constanten moeten aanpassen. Laat de dubbele aanhalingstekens staan, wijzig alleen wat daarbinnen staat!

De naam van je wifi-netwerk:

> `const char* ssid = "SSID"`

Het bijbehorende wifi-wachtwoord:

> `const char* password = "PASSWORD"`

Het pad naar het serverscript met de gewenste coördinaten als parameters:

> `const String rainurl = "URL"`

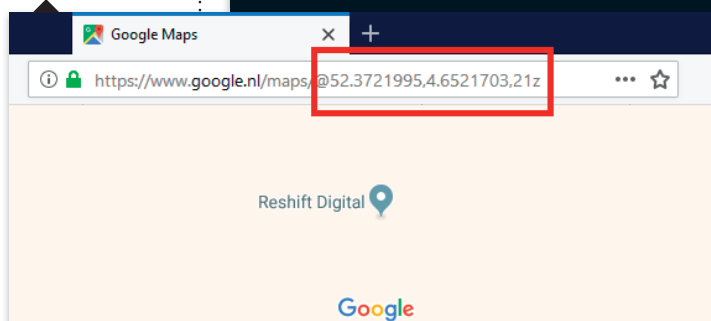
De code van het meetstation waar je het dichtst bij woont:

> `const String weerstation = "CODE"`

Het pad naar het serverscript met jouw provincie of regio als parameters:

> `const String alarmurl = "URL"`

Met Google Maps vind je snel en eenvoudig de gps-coördinaten van jouw locatie.



Je gps-coördinaten vind je eenvoudig door met je browser naar Google Maps (<https://maps.google.nl>) te gaan en in te zoomen op je straat. In de adresbalk staan de coördinaten achter het @-teken. Gebruik alleen het aantal graden en de eerste twee decimalen, in het voorbeeld dus **52.37** en **4.65** (zie afbeelding). Zie de commentaren met uitleg in de code om te zien hoe je de lijsten met meetstations en mogelijke regio's te zien krijgt.

04

Piepsignaal

Zoals eerder opgemerkt komt bij bepaalde situaties geluid uit de piëzo-luidspreker, zoals bij verwachte regen, als de temperatuur onder nul komt of bij een weeralarm. Om te voorkomen dat bijvoorbeeld bij buien elke tien minuten de pieper afgaat, is het aantal keer dat dit alarm klinkt beperkt. De duur is vastgelegd door een constante **interval** van het type **unsigned long** en staat standaard op een uur.

Zolang er nog niets heeft geklonken, gaat de pieper bij de eerste aanleiding. Vanaf dat moment geldt als voorwaarde dat er ten minste 3.600.000 milliseconden (een uur dus) verstreken moeten zijn voordat er weer wat mag piepen. De code maakt gebruik van een ingebouwde timer, die 49 dagen lang het aantal milliseconden telt vanaf het moment dat de module wordt aangezet. Daarna begint de teller weer bij nul, maar daar houdt de code rekening mee. De leds hebben met deze beperking uiteraard niets te maken en reageren direct op veranderingen.

05

Weersituatie in het scherm

De gegevens zijn afgeleid uit de xml-feed van Buienradar.nl. Dat houdt in dat de weergegevens binnen xml-tags staan, waar je deze uit moet halen. De functie **lees_tot_tag** verwerkt de feed. Deze wordt in de functie **checkweer** »



05 Op het scherm staat de temperatuur, actuele weersituatie, windsterkte en windrichting. (Dit is overigens een iets slijkere versie van ons weerstation met een speciaal ontworpen printplaatje.)

verderop in de code aangeroepen met als parameters de begin- en eindstrings van de gezochte weerinformatie. Bij de temperatuur-tag wordt bovendien nog een waarde **1** toegekend aan **vorst**, daarover straks meer. In **checkweer** staan de letterlijke tags waarbinnen de informatie staat.

De feed wordt ingelezen vanaf het gewenste weerstation en als dan de gezochte begintag is gevonden, knippen we die regel in stukken. Als **regel** bijvoorbeeld `<temperatuurGC>16.5</temperatuurGC>` is, halen we met **substring** de tekst **16.5** eruit. Die voegen we toe aan de string **weersituatie**, die verder wordt verwerkt door de functie **checkweer**. Daarbinnen gebeurt de echte magie, zoals het omzetten van de weersituatie naar het juiste pictogram en de windrichting naar de windpijl.

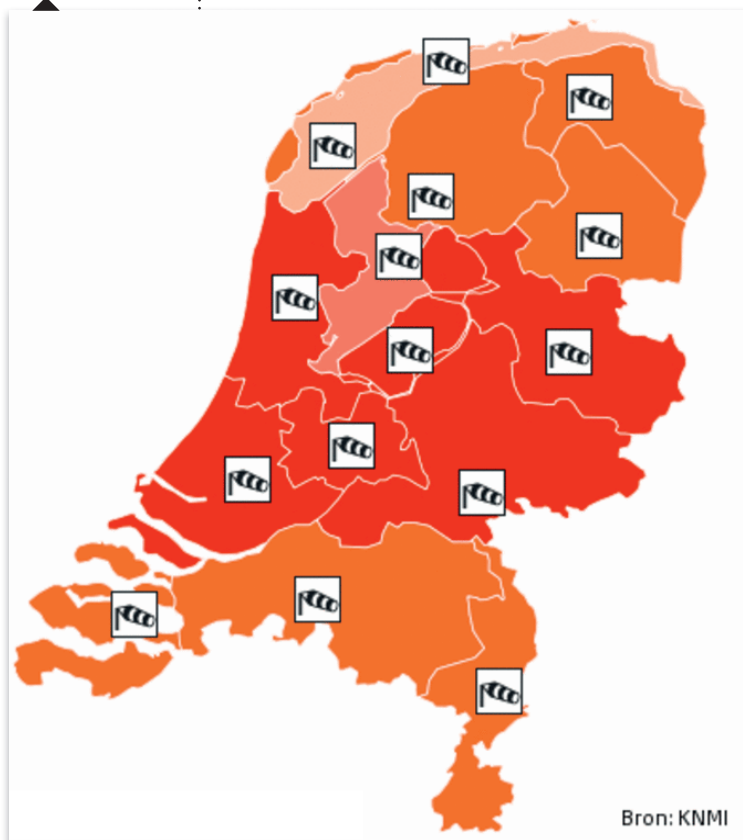
De functie eindigt met het aanroepen van **displayweather** met vier parameters: de windrichting, de temperatuur, de windkracht en het weersymbool. Aan windkracht en temperatuur voegen we respectievelijk **Bft** en **°C** toe, waarna de waarden naar het scherm worden geschreven.

06 Hopelijk blijft deze ellende ons bespaard ...

06

Vorst- en weeralarm

Tot slot laat de functie **lees_tot_tag** de blauwe led branden als het vriest, door



Bron: KNMI

te kijken of er een minteken voorkomt in de regel temperatuur. De test op de aanwezigheid daarvan vindt alleen plaats als **checkweer** de waarde **1** meegeeft aan **vorst**, dus alleen wanneer de temperatuur wordt doorgegeven. Dit voorkomt vals alarm als er elders in de tekst een minteken staat. Door booleaanse variabele **ijs** op **true** (waar) te zetten na het eerste piepsignaal, piept de luidspreker alleen aan het begin van een vorstperiode. De temperatuur moet eerst weer boven nul zijn geweest, voor er weer een vorstalarm kan klinken.

De functie **checkalarm** is redelijk rechttoe-rechtaan. Doordat de pagina is voorbereid (zie paragraaf 'Werking') volstaat het lezen van slechts één tekstregel. De aanwezigheid van de string **code_rood**, **code_oranje**, **code_geel** of **code_groen** (wat wordt getest met **indexOf**) bepaalt vervolgens welke led er brandt. Bij een weeralarm gaat de pieper en wordt de booleaanse variabele **alarm** op **true** gezet, bij code groen op **false**. Net als bij het vorstalarm voorkomt dit herhaaldelijk piepen zolang er een weeralarm geldt.

07

Regen

Omwille van de betrouwbaarheid en de flexibiliteit gebruiken we niet één enkele waarde, maar een tabel met de regenverwachting per vijf minuten voor de komende twee uur. Dat maakt het in theorie mogelijk om een lange periode vooruit te kijken, ware het niet dat de kwaliteit van die langetermijnsvoorspellingen vaak te wensen overlaat. Het bruikbaarst zijn de gegevens tot pakweg een kwartier vooruit. De praktijk leert bovendien dat de kwaliteit enorm toeneemt als er twee aaneengesloten blokken van vijf minuten allebei regen of juist droog weer voorspellen. Om die reden kijkt de functie **checkrain** zowel naar de geselecteerde periode als naar die van de regel erna (vijf minuten later). Pas als ze het 'met elkaar eens zijn', verandert de status.

De regentabel bestaat uit 24 regels en twee door het sluitteken (|, beter bekend als pipe) gescheiden kolommen. De eerste kolom bevat de hoeveelheid neerslag, de tweede de tijd. De tabel wordt ingelezen en met de functie **getValue** in twee etappes gesplitst om de regendata over te houden. Die waarde zetten we om naar een geheel getal, zodat we ermee kunnen rekenen. De hoeveelheid doet er niet toe, nul staat voor droog en alles groter dan

07 De tabel met regendata, om vijf over vier begint een regenbui die 35 minuten aanhoudt.

1	000 15:50
2	000 15:55
3	000 16:00
4	077 16:05
5	077 16:10
6	097 16:15
7	101 16:20
8	109 16:25
9	117 16:30
10	081 16:35
11	077 16:40
12	000 16:45
13	000 16:50
14	000 16:55
15	000 17:00
16	000 17:05
17	000 17:10
18	000 17:15
19	000 17:20
20	000 17:25
21	000 17:30
22	000 17:35
23	000 17:40
24	000 17:45

NIET ALLE MEETSTATIONS ZIJN BRUIKBAAR!

Helaas geven sommige meetstations niet alle gegevens door die wij gebruiken. Controleer daarom in een browser eerst of de gebruikte data aanwezig zijn. Voor meetstation De Bilt (stationcode 6260) typ je **view-source:http://arduino.reshift.nl/weermonitor.php?weerstation=6260** in de adresbalk van je browser. Zijn de data binnen de tags **<temperatuurGC>**, **<windsnelheidBF>** en **<windrichting>**, evenals de letter bij **<icoonactueel ID=** aanwezig? Pas dan is het meetstation geschikt voor onze taak. Anders zul je moeten uitwijken naar een ander nabijgelegen station.

```

1 Bron: <a href=https://buienradar.nl>Buienradar B.V.</a><br>
2 <weerstation id="6260">
3   <stationcode>6260</stationcode>
4   <stationnaam regio="Utrecht">Meetstation De Bilt</stationnaam>
5   <lat>52.10</lat>
6   <lon>5.18</lon>
7   <datum>06/13/2019 11:10:00</datum>
8   <luchtvochtigheid>68</luchtvochtigheid>
9   <temperatuurGC>16.1</temperatuurGC>
10  <windsnelheidMS>6.1</windsnelheidMS>
11  <windsnelheidBF>4</windsnelheidBF>
12  <windrichtingGR>195</windrichtingGR>
13  <windrichting>ZZW</windrichting>
14  <luchtdruk>1010.4</luchtdruk>
15  <zichtmeters>44700</zichtmeters>
16  <windstotenMS>11.2</windstotenMS>
17  <regenMMPU>-</regenMMPU>
18  <zonintensiteitM2>477</zonintensiteitM2>
19  <icoonactueel ID="c" zin="Zwaar"
20  <temperatuur10cm>16.2</temperatuur10cm>
21  <url>https://www.buienradar.nl/nederland/weerbericht/weergrafieken/6260</url>
22  <latGraden>52.17</latGraden>
23  <lonGraden>5.31</lonGraden>
24  </weerstation>

```

Zijn alle
benodigde
gegevens
aanwezig?

nul betekent regen. Eerst halen we de gekozen periode +1 op en via een aantal tussenstappen kennen we de waarde toe aan de variabele **regen**. Daarna volgt de gekozen periode zelf, die resulteert in de waarde voor **regennu**.

Zijn beide groter dan nul, dan gaat de blauwe led branden en klinkt een piepsignaal. Daarbij wordt de booleaanse variabele **nat** op **true** gezet om bij langdurige regenval herhaald piepen te voorkomen. Zijn beide variabelen gelijk aan nul, dan is het droog en is de blauwe led uit. **Nat** wordt **false**, zodat de pieper de volgende keer dat het regent weer alarm mag slaan.

08

Code uploaden en testen

De volgende stappen vereisen dat de ontwikkelomgeving Arduino IDE al is geïnstalleerd en geconfigureerd (zie kader 'Ontwikkelomgeving'). Sluit de ESP-module aan met een usb-stekker en pak het zip-bestand dat je hebt gedownload uit naar een willekeurige map op je harde schijf en open de map **esp-weerstation-grafisch**. Dubbelklik op het bestand **esp-weerstation-grafisch.ino** en wijzig de gegevens bij **ssid**, **password**, **rainurl**, **weerstation** en **alarmurl** (zie ook het kader 'Constanten').

Druk op Ctrl+Shift+M om de seriële monitor te

openen, waarmee je straks in de gaten kunt houden wat er gebeurt. Stel de snelheid van de seriële monitor in op 115200 baud. Klik op het pijltje naar rechtsboven in de ontwikkelomgeving. De code wordt nu achtereenvolgens gecontroleerd, gecompileerd en geupload naar de module.

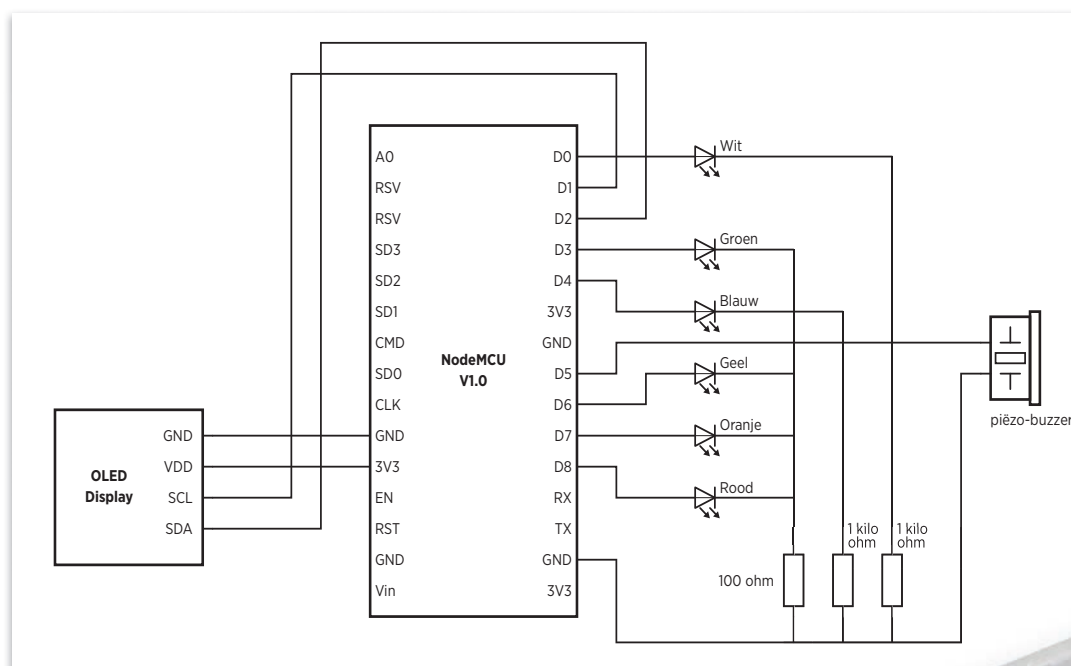
Eerst wordt de wifi-verbinding opgezet en als dat is gelukt, verwerkt de module achtereenvolgens de regendata, het weerrapport en de status van het weeralarm. Bij rustige, droge omstandigheden boven nul verschijnt het weerrapport op het scherm en brandt de groene led. Is er neerslag en/of vorst, dan branden de blauwe en/of de witte led. En bij een actief weeralarm ten slotte brandt geel, oranje of rood. Elke vijf minuten (of andere periode die je hebt ingesteld) wordt de informatie ververs, waarbij het schermpe gedurende enkele seconden uitgaat.

09

Bouwen

De componenten zijn tamelijk eenvoudig aan te sluiten. Omdat het een experimenteerprintplaat is, zul je zelf goed moeten nadenken over de plaatsing van de componenten. Druk van bovenaf telkens een component goed aan, zodat alle pootjes er aan de onderkant uitsteken. Buig dan een aantal pootjes om in tegengestelde

09 Schematische weergave van de schakeling.



richting, zodat het onderdeel aan de printplaat zit vastgeklemd. Knip de pootjes af op een lengte van ongeveer twee millimeter en herhaal deze stappen voor alle componenten. Sluit de kathodes (korte pootjes) van de groene, gele, oranje en rode led via één gezamenlijke weerstand van 100 ohm aangesloten op **GND**. De kathodes van de witte en de blauwe led zijn elk via een eigen weerstand van 1 kilo-ohm aangesloten op **GND**. Verbind de anodes (lange pootjes) van de leds als volgt met de ESP-module: wit op **D0**, groen op **D3**, blauw op **D4**, geel op **D6**, oranje op **D7** en rood op **D8**. Sluit de piëzo-luidspreker aan tussen **D5** en **GND**; en van het scherm **SDA** op **D2**, **SCL** op **D1**, **VCC** op **3V3**, en **GND** op **GND**.

Soldeer alle pootjes van de componenten en gebruik stukjes draad waarvan je de uiteinden vertint om de onderlinge verbindingen te maken volgens bovenstaand schema. Op onze website vind je via www.tiny.cc/sold tips voor solderen.

10

Afwerking

Een printplaatje van acht bij vijf centimeter blijkt precies te passen in een pillendoosje van het Kruidvat. Je zult alleen wel nog even zorgvuldig een stanleymes moeten gebruiken om een uitsparing voor de usb-stekker te maken en om een rechthoek uit te snijden zodat je het schermje kunt lezen. Het doosje is namelijk mat, wat lezen door de behuizing heen onmogelijk maakt. De randen van het gat kun je netjes afwerken met zwarte tape. Vanzelfsprekend is iedere passende behuizing bruikbaar en je zou zelfs een acryl fotolijstje kunnen gebruiken. Het mooiste is het natuurlijk als je zelf iets maakt met bijvoorbeeld een 3D-printer of freesmachine. ☒

10 Een pillendoosje van het Kruidvat past precies, maar is niet volledig transparant.

VOOR LEZERS UIT BELGIË

De code is na een kleine aanpassing ook geschikt voor onze Vlaamse lezers.

Als je bij <https://xml.weerslag.nl/xmlfeed.xml> zoekt naar het gewenste station, kun je bij **<stationcode>** het bijbehorende nummer vinden. Dat nummer gebruik je vervolgens in je programma, waarbij je **weermonitor.php** vervangt door **weerslag.php** op de volgende manier (als voorbeeld dient het weerstation van Brasschaat):

```
> const String weerstation="11162";
> const String weerurl="/weerslag.php?weerstation="+ weerstation;
```

De regendata omvatten ook die van België, dus daarvoor hoeft je alleen de juiste gps-coördinaten op te zoeken. Voor het weeralarm gebruik je de provincies Zeeland, Noord-Brabant of Limburg.