

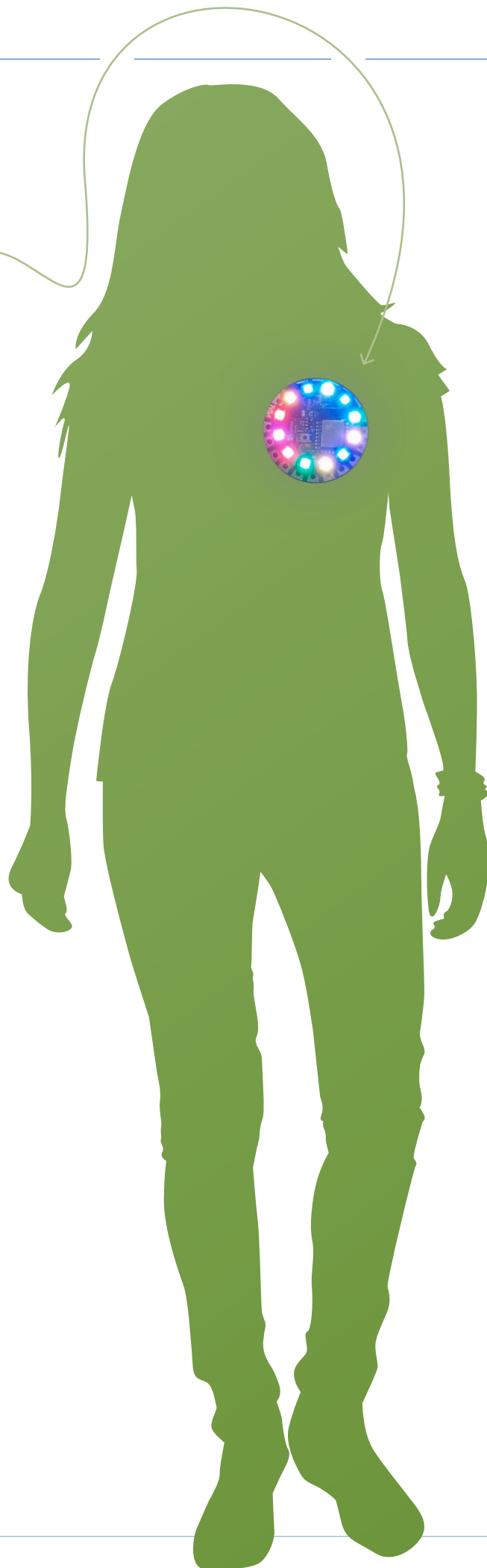
Wearable Wi-Fi Gadget

ESPHome,
uw steun en toeverlaat

Clemens Valens (Elektor)

Kent u die dozen vol printen die u bewaart voor als u ooit meer tijd hebt? Omdat ik een ESP8266-module nodig had voor mijn domotica-experimenten, wierp ik een blik in een paar van die dozen – en vond een op ESP8266 gebaseerd ‘wearable’-board dat ik een paar jaar geleden heb ontwikkeld. Niets spectaculairs, eigenlijk gewoon een ESP-12E-module met een USB/serieel-converter erop, plus een driver voor een adresseerbare WS2812-gebaseerde LED-string. Voor dit deel van onze serie heb ik dit board tot leven gewekt – uiteraard met de hulp van ESPHome!

Enkele jaren geleden vroeg een collega me om een draagbaar micro-controllerboard te ontwerpen met daarop een ESP8266-gebaseerde WiFi-module; hij zou de firmwarekant van het project doen, aangezien hij grote plannen had voor zo'n apparaat. Toen ik het prototype klaar had, probeerde hij het uit – en toen verliet hij ons. Was hij zo teleurgesteld in mijn ontwerp? Ik ben er nooit achter gekomen.



Het project werd opgegeven en het zou in de vergetelheid zijn geraakt als ik – sprong in de tijd naar een paar maanden geleden – geen dringende behoefte had gehad aan een ESP8266-module voor mijn domotica-experimenten. Terwijl ik dozen vol spullen doorzocht die op een dag van pas zouden kunnen komen, kwam ik mijn draagbare ESP8266-prototype tegen. En aangezien ik op dat moment geobserveerd was door ESPHome [1], wist ik meteen dat ik nu de tools had om eindelijk de software voor dit board te maken.

Een NodeMCU-achtige schakeling

Het schema van het board (**figuur 1**) is in feite een NodeMCU waarbij de CP2101 USB/serieel bridge van Silicon Laboratories (ook bekend als Silabs) is vervangen door de veel goedkopere FT231XS van FTDI. Er is ook een poort toegevoegd voor een adresseerbare WS2812-gebaseerde LED-string (NeoPixels als u dat liever hebt).

Omdat het draagbare gadget bijna een NodeMCU is, geldt alles wat volgt ook voor een normale NodeMCU-module. De gepresenteerde software werkt ook even goed.

De beschikbare GPIO-poorten en voeding worden naar buiten gevoerd op speciale pads met grote gaten die in een cirkel langs de rand van

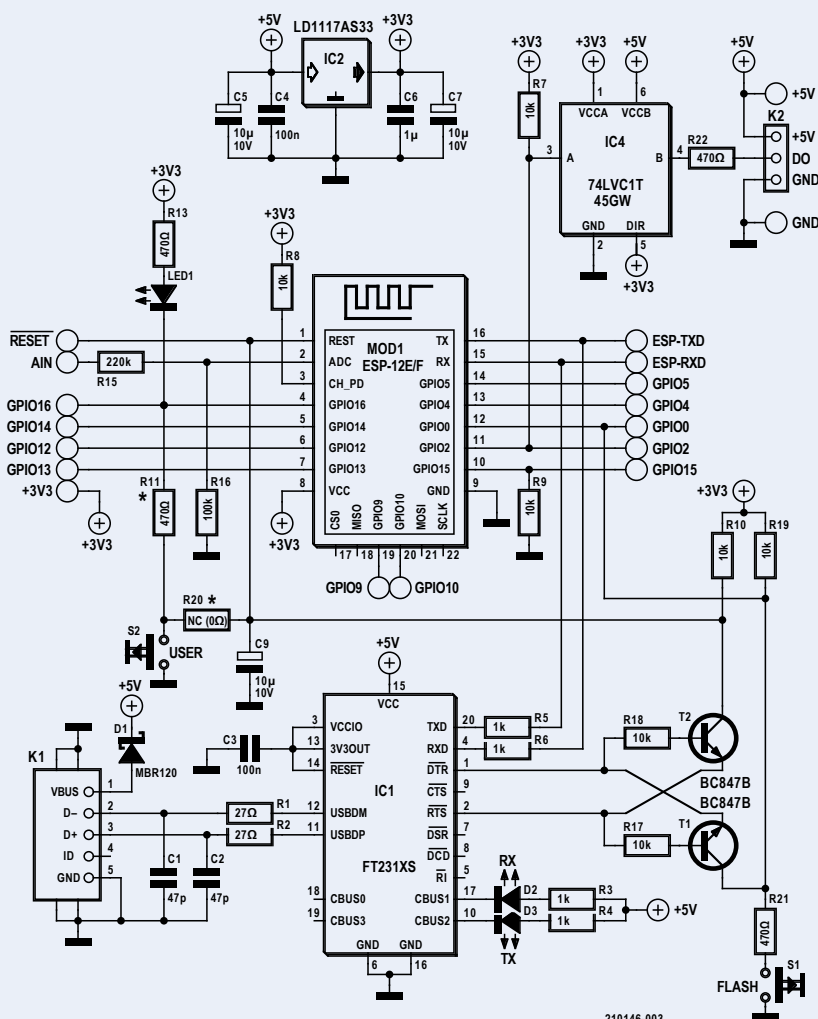
het ronde board zijn geplaatst. Deze pads zijn bedoeld voor gebruik met geleidende draad, maar u kunt er natuurlijk ook draden aan solderen of krokodillenklemmen gebruiken.

De stroom voor de schakeling wordt geleverd door de micro-USB-poort of door een externe 5V-voeding aan te sluiten op een van de 5V-pinnen. Dat is handig bij het gebruik van lange LED-strings, omdat deze meer stroom trekken dan een normale USB-poort kan leveren. Een USB-powerbank met hoge capaciteit is ook een optie. Merk op dat de LED-interface voor 5V-strings is. Merk ook op dat de 5V-ingang geen beveiliging tegen omgekeerde polariteit heeft.

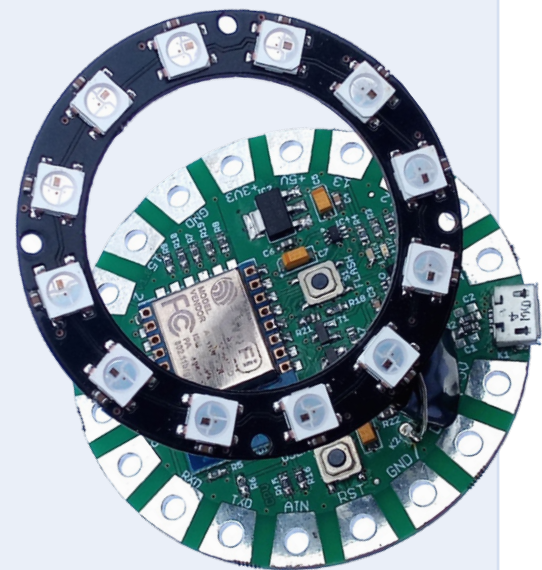
Software-ontwerp met ESPHome

Voor mijn experimenten heb ik een ringvormige string met twaalf WS2812-LED's aangesloten op poort K2, zie **figuur 2**. Natuurlijk kunt u de software voor dit board helemaal opnieuw schrijven, zoals mijn voormalige collega had gepland, en waarschijnlijk ook had moeten doen omdat er niet zoveel ESP8266-code beschikbaar was als nu, maar het adopteren van een open source-project zoals ESPHome bespaart een enorme hoeveelheid werk.

Ik heb het al eerder gezegd, en ik blijf het zeggen: met ESPHome



Figuur 1. Schema van het wearable ESP8266-board.



Figuur 2. Het prototype van het wearable WiFi-gadget, uitgerust met een 12-pixel adresseerbare RGB LED-ring.



Listing 1. Het YAML-configuratiebestand [2].

```
# Elektor 160112 Wearable ESP8266
# Configuration file for ESPHome

esphome:
  name: wearable
  platform: ESP8266
  board: nodemcu

wifi:
  ssid: "my_ssid"
  password: "my_passphrase"
  ap:
    ssid: "Wearable Fallback Hotspot"
    password: "12345678"

captive_portal:

# Enable logging
logger:

# Enable Home Assistant API
api:

# Enable Over-the-Air updates.
ota:

output:
  - platform: gpio
    id: "blue_led"
    pin:
      number: GPIO16
      inverted: True

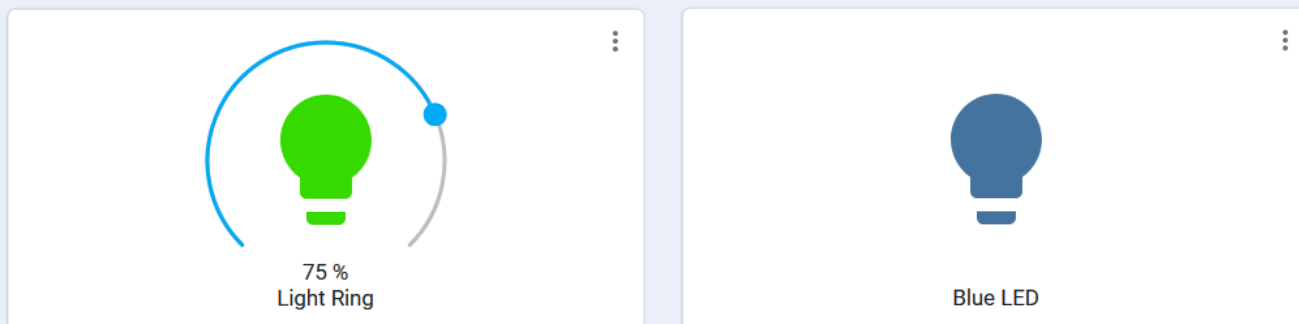
light:
  - platform: binary
    name: "Blue LED"
    output: "blue_led"
  - platform: neopixelbus
    name: "Light Ring"
    num_leds: 12
    type: GRB
    pin: GPIO2
    method: ESP8266_UART1
    effects:
      - addressable_color_wipe:
          name: "Color Wipe"
      - addressable_fireworks:
          name: "Fireworks"
      - flicker:
          name: "Flicker All"
      - addressable_flicker: # Doesn't work?
          name: "Flicker Individually"
```

```
  - addressable_rainbow:
      name: "Rainbow"
  - random:
      name: "Random All"
  - addressable_scan:
      name: "Scan"
  - strobe:
      name: "Strobe All"
  - addressable_twinkle:
      name: "Twinkle"
  - addressable_random_twinkle:
      name: "Twinkle Random"

# GPIO11 is somehow related to flash and should
not be used.
switch:
  - platform: gpio
    name: "GPIO4"
    pin: GPIO4
  - platform: gpio
    name: "GPIO5"
    pin: GPIO5
  - platform: gpio
    name: "GPIO12"
    pin: GPIO12
  - platform: gpio
    name: "GPIO14"
    pin: GPIO14
  - platform: gpio
    name: "GPIO15"
    pin: GPIO15

# Pushbutton on GPIO0.
binary_sensor:
  - platform: gpio
    name: "Flash"
    pin:
      number: GPIO0
      inverted: True

sensor:
  - platform: adc
    name: "Analog Input"
    pin: A0
    update_interval: 60s
    filters:
      - multiply: 3.2 # voltage divider is 100k/
        (220k+100k)
```



Figuur 3. Met deze kaarten in Home Assistant kunt u de LED-ring en de blauwe LED van het wearable WiFi-gadget bedienen.

kunt u in een paar minuten een *connected* applicatie maken voor de ESP8266 of ESP32 met over-the-air (OTA) programmering, fallback-hot-spot, webserver-gebruikersinterface en interfaces voor meer dan 200 apparaten. Echt waar. Zie [1] voor meer details.

ESPHome maakt gebruik van een modulaire aanpak waarbij kant-en-klare codeblokken worden gecombineerd tot een applicatie. De blokken die de applicatie nodig heeft, worden opgenomen in een configuratiebestand, het zogenaamde YAML-bestand (YAML staat voor *YAML Ain't Markup Language* en is geen taal maar een set opmaakregels voor tekstbestanden om parameters en waarden te specificeren [1]). Elk blok wordt afzonderlijk geconfigureerd, om bijvoorbeeld de GPIO-pin(nen) te specificeren die het moet gebruiken, of het communicatieprotocol of het type.

Het configuratiebestand wordt gelezen door ESPHome en omgezet in C++ code, die vervolgens wordt gecompileerd tot een uitvoerbaar bestand dat kan worden geprogrammeerd in het flashgeheugen van

de module. Als u eenmaal begrijpt hoe u een configuratiebestand moet samenstellen, kunt u aan de slag. Raadpleeg [1] voor meer details.

Het configuratiebestand nader bekeken

Het YAML-configuratiebestand (**listing 1**) begint met het verplichte *esphome*: een sectie om een naam, de gebruikte MCU (ESP8266) en het boardtype (NodeMCU) te specificeren.

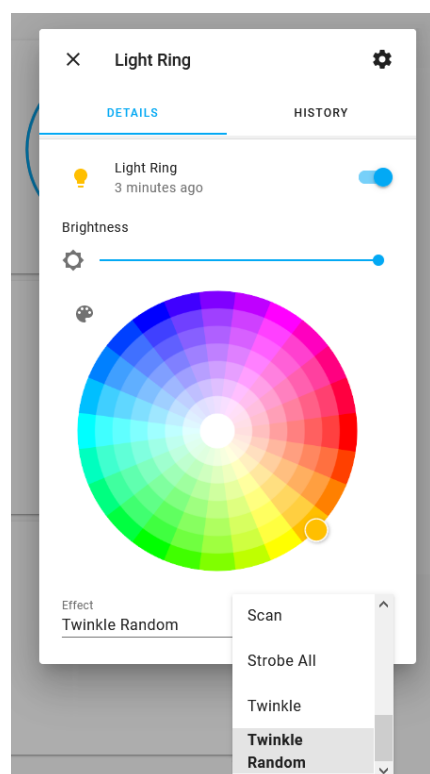
Daarna komt de WiFi-sectie om het netwerk op te geven waarmee verbinding moet worden gemaakt en een uitwijkmogelijkheid in geval van netwerkproblemen. Merk op dat de volgorde van secties niet van belang is.

Specificatie van de *logger*: deze optie activeert de status-uitvoer op de seriële poort. De *api*: -optie maakt eenvoudige integratie mogelijk met de gratis en open source besturingssoftware voor huisautomatisering Home Assistant (zie [1]). En *ota*: is voor draadloze programmering van het apparaat (vanuit bijvoorbeeld Home Assistant), wat erg praktisch is omdat het een fysieke verbinding met het apparaat overbodig maakt. Vervolgens komt de eigenlijke applicatie, te beginnen met de specificatie dat GPIO16 een uitgang moet zijn. Dat is nodig als u de LED die erop is aangesloten als *light* wilt gebruiken, en dat is interessant – aangezien lampen verschillende mogelijkheden hebben zoals schakelaars (**figuur 3**).

Lights

Als *light* heb ik de blauwe LED op GPIO16 en de LED-string gedefinieerd. Dit laatste wordt afgehandeld door het *neopixelbus*-platform, dat een paar opties heeft die moeten worden gespecificeerd, zoals de lengte van de string en de poort waarmee deze is verbonden. In dit geval is de poort GPIO2 die het gebruik van de *ESP8266_UART1* methode mogelijk maakt. Bij het specificeren van deze methode hoeft u eigenlijk niet GPIO2 op te geven omdat dit impliciet is.

Lights kunnen effecten hebben en ESPHome heeft er een paar ingebouwd die u kunt gebruiken (als uw light dit natuurlijk ondersteunt). Uiteraard kunt u ook uw eigen lichteffecten programmeren. Ik heb de meeste ingebouwde effecten voor de LED-ring gespecificeerd. Effecten kunnen parameters hebben, maar zonder deze worden standaardwaarden gebruikt. Met Home Assistant kunt u kiezen welk effect actief is (**figuur 4**). Ik vind het random twinkle-effect erg leuk.



Figuur 4. De lichteffecten die zijn aangegeven in het ESPHome YAML-configuratiebestand worden weergegeven als een lijst in Home Assistant, waaruit u die kunt kiezen die u mooi vindt.

Andere in- en uitgangen

De ongebruikte GPIO-poorten worden gedeclareerd als schakelaars, zodat u ze vanuit Home Assistant kunt in- en uitschakelen. Bij domotica is een schakelaar een apparaat dat wordt aangestuurd door het systeem (bijvoorbeeld een relais). Een schakelaar die wordt aangestuurd door de gebruiker (of bewoner) is een binaire sensor. Op GPIO0 is een drukknop aangesloten en is daarom gespecificeerd als een binaire sensor.

Ten slotte heeft het board een analoge ingang op pin A0 met een spanningsdeler ervoor (R15 en R16), wat de vermenigvuldigingsfactor van 3,2 verklaart die de gedeelde ingangsspanning weer omzet in volt. De maximale ingangsspanning is 3,3 V.

Maak er een sociaal afstandsalarm van

ESPHome staat voor automatisering en u kunt bijvoorbeeld een naderingssensor aan het board toevoegen om de kleur van de LED-ring te veranderen, afhankelijk van wat de sensor 'ziet'. Zo zou het board bijvoorbeeld een sociaal afstandsalarm kunnen zijn. Het kan ook gewoon een decoratieve elektronische speld of broche zijn; laat uw fantasie de vrije loop. Een interessante mogelijkheid hier is om Home Assistant te gebruiken. Met een paar van deze draagbare ESP8266-boards die in Home Assistant zijn geïntegreerd, kunnen mooie lichteffecten (bijvoorbeeld voor Kerstmis) worden gecreëerd. Maar hoewel Home Assistant is geoptimaliseerd voor domotica, kan het ook andere dingen doen, zoals een game besturen op het verjaardagsfeestje van uw peuter. Plak een board op elk kind en gebruik Home Assistant om teams te maken en te besturen of om te beslissen welke kandidaat een vraag mag beant-



GERELATEERDE PRODUCTEN

- > **ESP-12F, ESP8266-based Wi-Fi-module (SKU17781)**
www.elektor.nl/esp-12f-esp8266-based-wi-fi-module-160100-92
- > **NodeMCU ESP8266 microcontroller board (SKU17952)**
www.elektor.nl/nodemcu-microcontroller-board-with-esp8266-and-lua
- > **H. Henrik Skovgaard, "IoT Home Hacks with ESP8266", Elektor 2020 (SKU19158)**
www.elektor.nl/iot-home-hacks-with-esp8266

woorden of wie 'het' is bij tikkertje. Ik ben er zeker van dat met een beetje creativiteit veel leuke toepassingen kunnen worden bedacht. De ontwerpbestanden kunnen worden gedownload van [2].

210146-03

Vragen of opmerkingen?

Heeft u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de auteur via clemens.valens@elektor.com of naar de redactie via redactie@elektor.com.

Een bijdrage van

Idee, ontwerp, tekst en foto's:
Clemens Valens
Schema: **Patrick Wielders**

Redactie:
Jens Nickel, C.J. Abate
Layout: **Harmen Heida**



ONDERDELENLIJST

Weerstanden:

(alle 5%, 50 V, 0,1 W, 0603)

R20 = 0 Ω

R1, R2 = 27 Ω

R11, R13, R21 R22 = 470 Ω

R3, R4, R5, R6 = 1 k

R7, R8, R9, R10, R17, R18, R19 = 10 k

R15 = 220 kΩ

R16 = 100 k

Condensatoren:

C1, C2 = 47 p, 0603

C3, C4 = 100 n, 0603

C6 = 1 μ, 0603

C5, C7, C9 = 10 μ, 16 V, Case-A

Halfgeleiders:

D1 = MBRS540

IC4 = 74LVC1T45GW

IC1 = FT231XS

IC2 = LD1117AS33

LED1 = LED, blauw, 0603

LED2 = LED, geel, 0603

LED3 = LED, rood, 0603

T1, T2 = BC847C

Diversen:

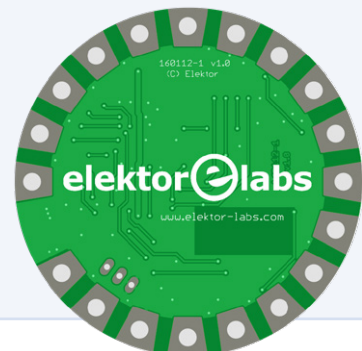
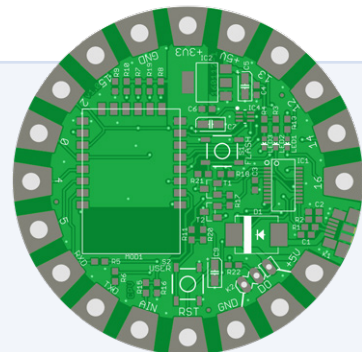
K1 = Micro-USB Type B-aansluiting,
bottom mount

K2 = 3-polige pinheader, steek 2,54 mm

S1, S2 = druktoets, 5,1x5,1 mm

MOD1 = ESP-12F

Print 160112-1



WEBLINKS

- [1] **C. Valens, "Domotica helemaal niet moeilijk", Elektor, september/oktober 2020:** www.elektormagazine.nl/200019-04
- [2] **Wearable Wi-Fi-gadget bij Elektor Labs:** www.elektormagazine.com/labs/4382

