

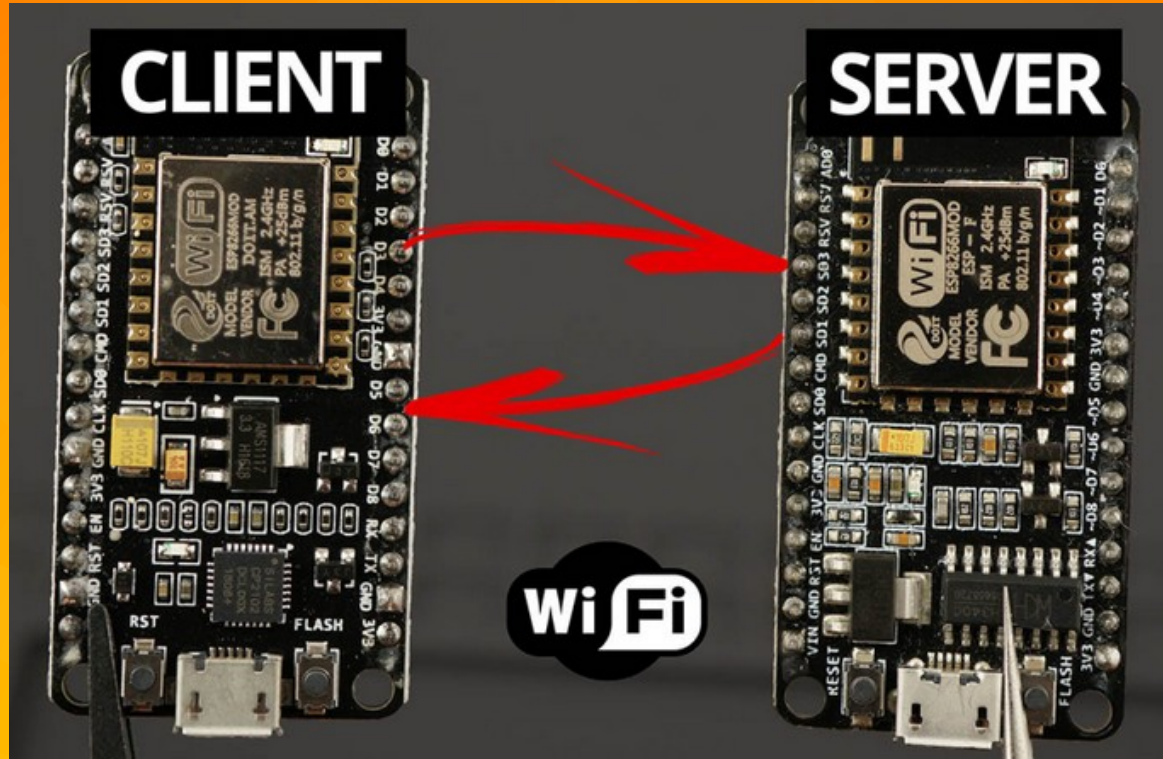
De Chatbot

- 1: Kan je er gebruik van maken in de Hobby?
- 2: Kan je programmeren zonder kennis programmeren?
- 3: Welke Chatbot kan je gebruiken?
- 4: De bekendste:
 - Chatgpt, Claude, Copilot, Gemenie
 - Al chat, Poe, Perplexity

ESP8266 Modules

- OM twee modules met elkaar laten praten, moeten ze geprogrammeerd worden.
- De taal die gekozen is voor de ARDUINO.
- Voor de CHAT-BOT is POE gebruikt.
- Voordeel ingeven en uitlezen is in NEDERLANDS
- Elke opdracht blijft bewaard, voor eventueel later.
- Je kan zelf bewaarde opdrachten verwijderen uit POE

ESP8266 Modules



De tekst om twee esp8266 aan elkaar te koppelen via wifi en met elkaar praten

- Verbinden van twee esp8266 via wifi
-
- Zender esp8266 initialiseren
- In de zender esp8266 zijn D0 en D1 de schakelaars.
- lees de schakelaars uit
- D0 en D1 = laag is cijfer = 0
- D0 = laag en D1 = hoog is cijfer = 1
- D0 = hoog en D1 = laag is cijfer = 2
- D0 = hoog en D1 = hoog is cijfer = 3
- Save de waarde in de variabele, bewaar en cijfer

- controle loop
- // contact dender
- wacht 0.2sec
- lees de schakelaars uit
- is cijfer == bewaar, dan einde loop
- anders lees opnieuw de schakelaars uit
- end loop
-
- als de verbinding naar ontvang esp8266 goed is
- verzend cijfer naar ontvanger esp8266
- Wacht 1sec
-

- Ontvanger esp8266 initialiseren
- cijfer_off = 0
- loop test ontvangen cijfer op:
- Is cijfer <0 of is cijfer >3, dan geen correct cijfer
- Led D4 op ontvanger esp8266 reset
- cijfer_off = 1
- Wacht opnieuw om cijfer te ontvangen
- is cijfer 0, 1, 2, of 3 dan is cijfer goed
- cijfer_off = 0
- end loop

- als cijfer_off = 0
- Verander de volgende pinnen
- case 0
- $\text{pin16} = 1, \text{pin5} = 0, \text{pin4} = 0, \text{pin0} = 0$
-
- case 1
- $\text{pin16} = 0, \text{pin5} = 1, \text{pin4} = 0, \text{pin0} = 0$
-
- case 2
- $\text{pin16} = 0, \text{pin5} = 0, \text{pin4} = 1, \text{pin0} = 0$
-

- case 3
- $\text{pin16} = 0, \text{pin5} = 0, \text{pin4} = 0, \text{pin0} = 1$
- einde cijfer_off
-
- Led D4 op ontvanger esp8266 aanzetten
-
- ontvanger esp8266 wachten op de volgende opdracht
-
- Maak er een schets voor Arduino van

Compileren met Arduino IDE

- Copy de gegenereerde tekst Zender ESP8266 in de IDE
- Compileer de tekst file en geen error save voor later
- Copy de gegenereerde tekst Ontvanger ESP8266 in de IDE
- Compileer de tekst file en geen error save voor later

Zender ESP8266

- `#include <ESP8266WiFi.h>`
-
- `const char* ssid = "your_SSID"; // Zet je Wi-Fi SSID hier`
- `const char* password = "your_PASSWORD"; // Zet je Wi-Fi wachtwoord hier`
- `WiFiClient client;`
-
- `const int switchD0 = D0;`
- `const int switchD1 = D1;`
- `int bewaar, cijfer;`
-

- `void setup() {`
- `pinMode(switchD0, INPUT);`
- `pinMode(switchD1, INPUT);`
- `Serial.begin(115200);`
- `WiFi.begin(ssid, password);`
- `while (WiFi.status() != WL_CONNECTED) {`
- `delay(1000);`
- `Serial.println("Verbinden met Wi-Fi...");`
- `}`
- `Serial.println("Verbonden met Wi-Fi.");`
- `}`

- void loop() {
- // Lees de schakelaars uit
- cijfer = (digitalRead(switchD0) << 1) |
digitalRead(switchD1);
-
- // Controle loop
- bewaar = cijfer;
- delay(200);
-

- while (true) {
- cijfer = (digitalRead(switchD0) << 1) |
digitalRead(switchD1);
- if (cijfer == bewaar) {
- break; // Einde loop
- }
- }
- }
-

- `// Verzend cijfer naar ontvanger`
- `if (client.connect("receiver_IP", 80)) { // Zet het IP-adres van de ontvanger hier`
- `client.print(cijfer);`
- `client.stop();`
- `}`
- `delay(1000);`
- `}`

Ontvanger ESP8266

- `#include <ESP8266WiFi.h>`
- `#include <ESP8266WebServer.h>`
-
- `const char* ssid = "your_SSID"; // Zet je Wi-Fi SSID hier`
- `const char* password = "your_PASSWORD"; // Zet je Wi-Fi wachtwoord hier`
- `ESP8266WebServer server(80);`
-
- `int cijfer_off = 0;`

- `void setup() {`
- `pinMode(16, OUTPUT);`
- `pinMode(5, OUTPUT);`
- `pinMode(4, OUTPUT);`
- `pinMode(0, OUTPUT);`
- `pinMode(2, OUTPUT); // LED D4`
- `digitalWrite(2, LOW); // LED uit`
- `}`

- Serial.begin(115200);
- WiFi.begin(ssid, password);
- while (WiFi.status() != WL_CONNECTED) {
- delay(1000);
- Serial.println("Verbinden met Wi-Fi...");
- }
- Serial.println("Verbonden met Wi-Fi.");
-
- server.on("/", handleClient);
- server.begin();
- }
-

- `void loop() {`
- `server.handleClient();`
- `}`
-
- `void handleClient() {`
- `String cijfer = server.arg(0); // Ontvang het cijfer`
- `int cijferInt = cijfer.toInt();`
-

- if (cijferInt < 0 || cijferInt > 3) {
- cijfer_off = 1; // Ongeldig cijfer
- digitalWrite(2, LOW); // LED uit
- } else {
- cijfer_off = 0; // Geldig cijfer
- digitalWrite(2, HIGH); // LED aan
- switch (cijferInt) {
- case 0:
- digitalWrite(16, HIGH);
- digitalWrite(5, LOW);
- digitalWrite(4, LOW);
- digitalWrite(0, LOW);
- break;

- case 1:
 - digitalWrite(16, LOW);
 - digitalWrite(5, HIGH);
 - digitalWrite(4, LOW);
 - digitalWrite(0, LOW);
 - break;
- case 2:
 - digitalWrite(16, LOW);
 - digitalWrite(5, LOW);
 - digitalWrite(4, HIGH);
 - digitalWrite(0, LOW);
 - break;

- case 3:
- `digitalWrite(16, LOW);`
- `digitalWrite(5, LOW);`
- `digitalWrite(4, LOW);`
- `digitalWrite(0, HIGH);`
- `break;`
- `}`
- `}`
- `server.send(200, "text/plain", "Received: " + cijfer); //`
Bevestiging naar client
- `}`

Error op lossen

- Compilattion error ' D4' was not declared in this scope
- Copy plakken in een Chatbot en dit is het antwoord Chatbot

```
const int pin = D4; // or  
const int pin = 2; // GPIO2 corresponds to D4
```

2. For Arduino Uno/Nano:

cpp

```
const int pin = 4; // Use the actual pin number
```

D4 veranderen in 2 en de error is opgelost

Uitleg van de:

Zender ESP8266

1. De zender maakt verbinding met het WiFi-netwerk dat door de ontvanger wordt gecreëerd.
2. Bij het opstarten wordt de initiële schakelaarstand gelezen en verzonden.
3. In de hoofdloop wordt elke seconde de schakelaarstand gecontroleerd.
4. Als er een verandering is, wordt het nieuwe cijfer naar de ontvanger verzonden via een HTTP GET-verzoek.

Uitleg van de:

Ontvanger ESP8266

1. De ontvanger creëert een WiFi-netwerk als access point.
2. Er wordt een webserver opgezet om GET-verzoeken te ontvangen.
3. Wanneer een verzoek wordt ontvangen, wordt de juiste GPIO-pin ingesteld op basis van het ontvangen cijfer.

Programmeren ESP8266

- Via een USB naar serieel adapter
- Zoals een FTDI-adapter CP2102, CH340
- VCC: verbinden met 3.3V, **geen 5V**
- Gnd: verbinden met Gnd
- TX adapter: verbinden met RX van de ESP8266
- RX adapter: verbinden met TX van de ESP8266
- GPIO0:m verbinden met de Gnd
Alleen tijdens programmeren

- ESP8266 in flash mode zetten:
Reset knop gebruiken of kort de voeding onderbreken
- Arduino juist instellen Com-poort, Upload-snelheid
- De software word via de seriële poort naar de ESP8266
- gestuurd
- Ontkoppel GPIO0 van aarde en reset ESP8266
- Als alles goed is gegaan werk de ESP8266 zoals bedoeld
- Zijn er nog vragen?
-

Verkorte Antenne 40Mtr

- De spoel is 4uH
- Frequentie is 7Mhz
- In CHATGPT in geven voor de berekening:
- Verticale antenne voor 40Mtr, verkort door een spoel
- van 4uH bereken de antenne

- 1. Berekening van de volledige 1/4-golf antenne
-
- Een $\lambda/4$ -antenne zonder verkorting heeft een theoretische lengte van:
- $L_{\text{quarter}} = 3004 \times 7 = 10,71$ meter
- $L_{\text{quarter}} = 4 \times 7300 = 10,71$ meter
-
- Maar omdat we de antenne verkorten met een spoel,
- wordt de fysieke lengte kleiner.

- 2. Berekening van de inductieve verkorting
-
- De reactantie van een 4 μH spoel op 7 MHz is:
- $X_L = 2\pi fL = 2\pi \times 7 \times 10^6 \times 4 \times 10^{-6}$
- $X_L = 175,9\Omega$
-
- De spoel compenseert een elektrisch lengteverschil en
- verkort de antenne effectief.

- 3. Opsplitsing van de antenne
-
- Je wilt de spoel plaatsen op $1/3$ van de totale antennehoogte.
- We noemen de totale (verkorte) lengte L en splitsen dit als volgt:
-
- Onderste deel vóór de spoel: $L_1=1/3L$
- Bovenste deel na de spoel: $L_2=2/3$
-
- De spoel moet ervoor zorgen dat de elektrische lengte
- van L_1+L_2 gelijk is aan de originele 10,71 meter.

• 4. Schatting van de effectieve lengte

-
- Een spoel op 1/3 hoogte maakt dat de antenne effectief rond 30-35% korter.
- Dit geeft een schatting:
- Lverkort $\approx 7,5$ tot $8,0$ meter
-
- Met de opsplitsing:
 - Onderste sectie vóór de spoel: $L1 \approx 2,5 - 2,7$ meter
 - Bovenste sectie na de spoel: $L2 \approx 5,0 - 5,3$ meter
- De spoel vervangt het ontbrekende stuk om
- elektrisch de volle $10,71$ meter te simuleren.

- Conclusie
-
- Een verkorte 40M verticale antenne met een 4 μ H spoel
- op 1/3 van de hoogte heeft een fysieke lengte van
- ongeveer 7,5 - 8 meter.
- De spoel wordt rond 2,5 - 2,7 meter hoogte geplaatst.
- Zijn er nog vragen?

- Dank voor Uw aandacht,
- Henk de Ronde (PA0JMD)