

# BrainMoveAJM: Installatiegids

---

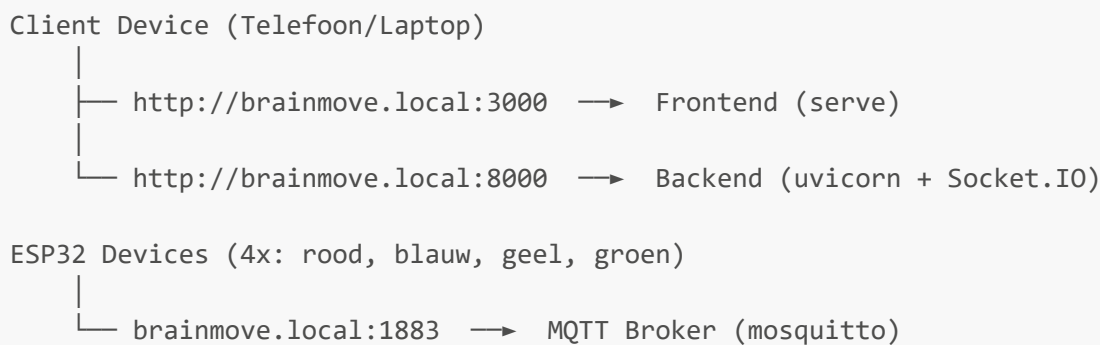
**Snelle Start:** Wil je alle stappen overslaan? Vraag een kant-en-klare SD-kaart image aan! Neem contact op via [arne.vandeputte@student.howest.be](mailto:arne.vandeputte@student.howest.be), [jonathan.matthys@student.howest.be](mailto:jonathan.matthys@student.howest.be), [michiel.gekiere@student.howest.be](mailto:michiel.gekiere@student.howest.be) om de vooraf geconfigureerde RPi image te ontvangen. Je hoeft dan alleen de image te flashen en de ESP32's te programmeren (Fase 10).

---

## Samenvatting Project

- **Hostnaam:** BrainMoveAJM
  - **Gebruiker:** bmajm
  - **Wachtwoord:** secureajm5!
  - **OS:** Debian Trixie (via RPi Imager)
  - **Netwerk:** Hotspot 10.42.0.1, Ethernet 192.168.137.50
  - **Frontend URL:** <http://brainmove.local:3000>
  - **Backend URL:** <http://brainmove.local:8000>
  - **SSID:** BrainMoveG1
  - **Communicatie:** MQTT (Mosquitto broker)
- 

## Architectuur Overzicht



**Opmerking:** Deze setup gebruikt GEEN nginx. Frontend en backend worden direct benaderd op hun respectievelijke poorten voor minimale latency bij real-time Socket.IO communicatie.

---

## Fase 1: SD Kaart Flashen (Op PC)

*Doel: Het besturingssysteem correct instellen voordat we de Pi aanraken.*

1. Download en installeer **Raspberry Pi Imager** op je PC.
2. Steek de MicroSD-kaart in je PC.
3. Open Imager en kies:
  - **Device:** Raspberry Pi 5.

- **OS:** Selecteer de **Debian Trixie** image.
- **Storage:** Selecteer je SD-kaart.

4. Klik op **Next** en kies **EDIT SETTINGS** (OS Customisation):

- **General:**
  - Hostname: **BrainMoveAJM**
  - Username: **bmajm**
  - Password: **secureajm5!**
  - Wireless LAN: **Uitvinken** (We gebruiken eerst de kabel).
- **Services:**
  - Vink "Enable SSH" aan -> "Use password authentication".

5. Klik op **SAVE** en **YES** om te flashen.

---

## Fase 2: Fysieke Aansluiting

*Doel: Hardware correct aansluiten. Let op de volgorde!*

1. Haal de SD-kaart uit de PC en steek deze in de Raspberry Pi.
2. Sluit de **Ethernetkabel** aan tussen de RPi en je Windows PC.
3. ⚠ **WACHT:** Sluit de stroom (USB-C) nog **niet** aan.

---

## Fase 3: Windows Netwerk Configureren (ICS)

*Doel: Internet delen van Windows naar de Pi via de kabel.*

1. Open op Windows: **ncpa.cpl** (via Windows-toets + R).
2. Rechtermuisknop op je **WiFi-adapter** (jouw internetbron) -> **Eigenschappen** -> **Delen**.
3. Vink aan: "Andere netwerkgebruikers toestaan verbinding te maken...".
4. Selecteer bij "Thuisnetwerkverbinding" de **Ethernet-adapter** van de Pi.
  - Windows geeft de ethernetpoort nu automatisch IP **192.168.137.1**.

---

## Fase 4: Eerste Boot & Updates

*Doel: Systeem starten en basissoftware installeren.*

1. **Stroom Aan:** Sluit nu de USB-C voeding aan. Wacht 2 minuten.
2. Open PowerShell op Windows en log in:

```
ssh bmajm@192.168.137.50
# Wachtwoord: secureajm5!
```

(Werkt het IP niet? Probeer `ssh bmajm@BrainMoveAJM.Local`)

### 3. Statisch IP vastzetten op de Pi:

```
sudo nmcli con mod "Wired connection 1" ipv4.addresses 192.168.137.50/24  
ipv4.gateway 192.168.137.1 ipv4.dns 8.8.8.8 ipv4.method manual  
sudo nmcli con down "Wired connection 1" && sudo nmcli con up "Wired  
connection 1"
```

### 4. Software installeren:

```
# Update pakketlijsten  
sudo apt update && sudo apt full-upgrade -y  
  
# Installeer systeemtools en Python tools  
sudo apt install -y git python3-venv python3-full tree mosquitto mosquitto-  
clients  
  
# Installeer Node.js 20 (voor frontend)  
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -  
sudo apt install -y nodejs
```

---

## Fase 5: Mosquitto MQTT Broker Configureren

*Doel: MQTT broker instellen voor communicatie met ESP32 apparaten.*

#### 1. Configuratie bestand aanmaken:

```
sudo nano /etc/mosquitto/conf.d/brainmove.conf
```

#### 2. Plak deze inhoud:

```
# BrainMove MQTT Broker Configuration  
listener 1883  
allow_anonymous true  
persistence true  
persistence_location /var/lib/mosquitto/  
max_queued_messages 1000  
log_dest file /var/log/mosquitto/mosquitto.log  
connection_messages true
```

#### 3. Mosquitto herstarten en inschakelen:

```
sudo systemctl restart mosquitto  
sudo systemctl enable mosquitto
```

#### 4. Test de broker:

```
# In terminal 1: subscribe
mosquitto_sub -h localhost -t "test/#" &

# In terminal 2: publish
mosquitto_pub -h localhost -t "test/hello" -m "MQTT werkt!"

# Je zou "MQTT werkt!" moeten zien
```

---

## Fase 6: Project Ophalen & Hotspot

*Doel: Code downloaden en WiFi netwerk (BrainMoveG1) maken.*

### 1. Clone Repository:

```
mkdir -p ~/BrainMove && cd ~/BrainMove
# Vervang <URL> met de juiste repository URL
git clone <JOUW_REPO_URL> BrainMoveG1
```

### 2. Hotspot Instellen: Controleer met *ip link* welke interface je dongle is (meestal *wlan1* bij RPi5).

```
sudo raspi-config nonint do_wifi_country BE

# Hotspot aanmaken
sudo nmcli con add type wifi ifname wlan1 con-name "BrainMoveG1" autoconnect
yes ssid "BrainMoveG1" mode ap ipv4.method shared wifi-sec.key-mgmt wpa-psk
wifi-sec.psk "YOUR_SECURE_PASSWORD"

sudo nmcli connection up "BrainMoveG1"
```

### 3. Controleer hotspot status:

```
nmcli connection show --active | grep BrainMoveG1
```

---

## Fase 7: Backend Installatie (met venv)

*Doel: Virtuele omgeving opzetten en dependencies installeren.*

### 1. Ga naar de map en maak de omgeving:

```
cd ~/BrainMove/BrainMoveG1
python3 -m venv venv
source venv/bin/activate
```

## 2. Installeer packages:

```
cd backend
pip install -r requirements.txt
```

## 3. Zet environment variabelen:

```
cp .env.example .env
```

De `.env` bevat:

```
# MQTT Configuration
MQTT_BROKER_HOST=localhost
MQTT_BROKER_PORT=1883
MQTT_KEEPALIVE=60
MQTT_RECONNECT_INTERVAL=5

# Network
HOST_IP=10.42.0.1
```

---

# Fase 8: Frontend Build

*Doel: Website bouwen voor productie.*

## 1. Controleer API configuratie:

```
cat ~/BrainMove/BrainMoveG1/frontend/.env.production
```

Moet bevatten:

```
VITE_API_BASE_URL=http://brainmove.local:8000
```

## 2. Builden:

```
cd ~/BrainMove/BrainMoveG1/frontend
npm install
npm run build
sudo npm install -g serve
```

---

## Fase 9: Systemd Services (Autostart)

*Doel: Alles automatisch laten starten bij boot.*

### 1. Backend Service:

```
sudo nano /etc/systemd/system/brainmove-backend.service
```

Plak:

```
[Unit]
Description=BrainMove Backend
After=network.target mosquitto.service
Wants=mosquitto.service

[Service]
User=bmajm
WorkingDirectory=/home/bmajm/BrainMove/BrainMoveG1/backend
Environment="PYTHONPATH=/home/bmajm/BrainMove/BrainMoveG1/backend:/home/bmajm/BrainMove/BrainMoveG1/backend/src"
ExecStart=/home/bmajm/BrainMove/BrainMoveG1/venv/bin/uvicorn src.main:app --host 0.0.0.0 --port 8000
Restart=always
RestartSec=5

[Install]
WantedBy=multi-user.target
```

### 2. Frontend Service:

```
sudo nano /etc/systemd/system/brainmove-frontend.service
```

Plak:

```
[Unit]
Description=BrainMove Frontend
After=network.target brainmove-backend.service
Requires=brainmove-backend.service
```

```
[Service]
User=bmajm
WorkingDirectory=/home/bmajm/BrainMove/BrainMoveG1/frontend
ExecStart=/usr/bin/npx serve -s dist -l 3000
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
```

### 3. Services activeren en starten:

```
sudo systemctl daemon-reload
sudo systemctl enable mosquitto brainmove-backend.service brainmove-frontend.service
sudo systemctl start brainmove-backend.service brainmove-frontend.service
```

### 4. Status controleren:

```
sudo systemctl status brainmove-backend.service
sudo systemctl status brainmove-frontend.service
sudo systemctl status mosquitto
```

---

## Fase 10: ESP32 Flashen

*Doel: ESP32 apparaten programmeren met MQTT firmware.*

De ESP32 firmware staat in `esp32/brainmove_mqtt/brainmove_mqtt.ino`.

### Belangrijke instellingen in de code:

```
// WiFi credentials
const char* WIFI_SSID = "BrainMoveG1";
const char* WIFI_PASSWORD = "bmSecure5!";

// MQTT broker (Raspberry Pi)
const char* MQTT_HOST = "brainmove.local";
const int MQTT_PORT = 1883;

// Device color (uniek per ESP32)
const char* DEVICE_COLOR = "rood"; // of: blauw, geel, groen
```

### Flash instructies:

1. Open Arduino IDE of PlatformIO
  2. Selecteer ESP32 board
  3. Pas `DEVICE_COLOR` aan voor elke ESP32
  4. Upload de sketch
- 

## Fase 11: Custom Domein Instellen (Optioneel)

Doel: `brainmove.local` gebruiken in plaats van het IP-adres.

Dit is optioneel - het IP-adres blijft altijd werken als fallback.

### 1. dnsmasq configuratie aanmaken:

```
sudo mkdir -p /etc/NetworkManager/dnsmasq.d  
sudo nano /etc/NetworkManager/dnsmasq.d/brainmove.conf
```

### 2. Plak deze inhoud:

```
address=/brainmove.local/10.42.0.1  
address=/brainmove/10.42.0.1
```

### 3. NetworkManager herstarten:

```
sudo systemctl restart NetworkManager
```

### 4. Frontend updaten:

```
# Update .env.production  
nano ~/BrainMove/BrainMoveG1/frontend/.env.production
```

Verander naar:

```
VITE_API_BASE_URL=http://brainmove.local:8000
```

### 5. Frontend rebuilden:

```
cd ~/BrainMove/BrainMoveG1/frontend  
npm run build
```



## 6. Services herstarten:

```
./scripts/restart_all.sh
```

**Resultaat:** Webapp bereikbaar via <http://brainmove.local:3000>

Voor uitgebreide troubleshooting, zie [dnsmasq-setup-guide.md](#)

---

## Gebruikers Toegang (QR Codes)

Print deze twee QR-codes uit voor op het apparaat:

### 1. Sticker 1: "Verbind met WiFi"

- Code: `WiFi:T:WPA;S:BrainMoveG1;P:bmSecure5!;;`

### 2. Sticker 2: "Speel het spel"

- Code: `http://brainmove.local:3000`

---

## Troubleshooting

### Backend start niet

```
# Bekijk logs
sudo journalctl -u brainmove-backend.service -f

# Test handmatig
cd ~/BrainMove/BrainMoveG1/backend
source ../venv/bin/activate
uvicorn src.main:app --host 0.0.0.0 --port 8000
```

### ESP32 verbindt niet met MQTT

```
# Monitor MQTT berichten
mosquitto_sub -h localhost -t "bm/#" -v

# Controleer of broker draait
sudo systemctl status mosquitto
```

### Frontend laadt niet

```
# Bekijk logs
sudo journalctl -u brainmove-frontend.service -f
```

```
# Test handmatig
cd ~/BrainMove/BrainMoveG1/frontend
npx serve -s dist -l 3000
```

## Hotspot werkt niet

```
# Status bekijken
nmcli connection show

# Hotspot herstarten
sudo nmcli connection down BrainMoveG1
sudo nmcli connection up BrainMoveG1
```

---

## Quick Reference

Service	URL/Poort	Doel
Frontend	<a href="http://brainmove.local:3000">http://brainmove.local:3000</a>	Webapp
Backend API	<a href="http://brainmove.local:8000">http://brainmove.local:8000</a>	REST API
API Docs	<a href="http://brainmove.local:8000/docs">http://brainmove.local:8000/docs</a>	Swagger documentatie
MQTT Broker	<a href="http://brainmove.local:1883">brainmove.local:1883</a>	ESP32 communicatie
Hotspot	<a href="#">BrainMoveG1</a>	WiFi netwerk