Klair Dashboard

Software Setup Guide

From Blank SD Card to Working Dashboard

Version 1.0 | December 2025

Contents

1. Prerequisites

What You Need

- A computer (Mac, Windows, or Linux) to flash the SD card
- MicroSD card (32GB or larger recommended)
- MicroSD card reader
- WiFi network name and password
- Raspberry Pi Imager software (free download)

Download Raspberry Pi Imager

Download from: https://www.raspberrypi.com/software/

Available for macOS, Windows, and Ubuntu.

2. Flash Raspberry Pi OS

1. Insert your MicroSD card into your computer's card reader.
2. Open Raspberry Pi Imager.
3. **Click 'Choose Device'** → Select 'Raspberry Pi 4'.
4. **Click 'Choose OS'** → Select 'Raspberry Pi OS (64-bit)' under 'Raspberry Pi OS (other)'.

ℹ️ **Why 64-bit?** Better performance with Node.js and modern software. Your Pi 4 fully supports it.

5. **Click 'Choose Storage'** → Select your MicroSD card.
6. **Click the gear icon (⚙️) or 'Edit Settings'** to open advanced options.

Configure Advanced Options (Important!)

In the OS Customisation menu, configure the following:

General Tab

Setting	Value
Set hostname	`klair-dashboard`
Set username	`parminder` (or your preferred username)
Set password	Your secure password
Configure WiFi	Enter your WiFi SSID and password
WiFi country	`GB` (United Kingdom)
Set locale	Timezone: Europe/London, Keyboard: gb

Services Tab

- **Enable SSH:** ✅ Check this box
- **Use password authentication:** ✅ Select this option

✓ **Tip:** Enabling SSH lets you configure the Pi remotely from your MacBook without needing a separate keyboard/mouse.

7. **Click 'Save'** to save your settings.
8. **Click 'Write'** and confirm. This will erase the SD card and write the OS.
9. Wait for the write and verification to complete (5-15 minutes).
10. Eject the SD card safely when prompted.

3. First Boot Configuration

1. Insert the MicroSD card into your Raspberry Pi 4.
2. Connect the display (if assembled) or an HDMI monitor.
3. Connect power to the Raspberry Pi.
4. Wait 2-3 minutes for first boot to complete.

Connect via SSH (Recommended)

From your MacBook terminal:

```
ssh parminder@klair-dashboard.local
```

If hostname doesn't resolve, find the IP from your router or use:

```
ping klair-dashboard.local
```

Then connect with:

```
ssh parminder@<IP_ADDRESS>
```

ℹ️ First connection: You'll be asked to verify the host fingerprint. Type 'yes' to continue.

4. System Updates & Dependencies

Once connected via SSH, run the following commands:

Update the System

```
sudo apt update && sudo apt upgrade -y
```

This may take 5-10 minutes. Let it complete fully.

Install Essential Packages

```
sudo apt install -y git curl wget build-essential
```

Install Chromium (for Kiosk Mode)

```
sudo apt install -y chromium-browser unclutter
```

unclutter hides the mouse cursor when not in use—perfect for a dashboard display.

5. Install Node.js

We'll use NodeSource to install Node.js 20 LTS (recommended for Next.js).

Install Node.js 20

```
# Download and run NodeSource setup script

curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -



# Install Node.js

sudo apt install -y nodejs
```

Verify Installation

```
node --version   # Should show v20.x.x

npm --version    # Should show 10.x.x
```

Install pnpm (Optional but Recommended)

```
sudo npm install -g pnpm
```

6. Install Python Hardware Libraries

Python is pre-installed on Raspberry Pi OS. We need to add libraries for GPIO and NeoPixel control.

Install pip (if not present)

```
sudo apt install -y python3-pip python3-venv
```

Install GPIO Library

```
sudo apt install -y python3-rpi.gpio
```

Install NeoPixel Libraries

```
sudo pip3 install rpi_ws281x adafruit-circuitpython-neopixel --break-system-packages
```

Install Flask (for Hardware API)

```
sudo pip3 install flask flask-cors --break-system-packages
```

⚠️ Note: The --break-system-packages flag is required on Raspberry Pi OS Bookworm. It's safe for these hardware libraries.

7. Configure GPIO & UART

Enable UART (Optional - for sensor configuration)

If you want to configure the mmWave sensor's sensitivity via UART:

```
sudo raspi-config
```

Navigate to:

- Interface Options → Serial Port
- Would you like a login shell over serial? → **No**
- Would you like serial port hardware enabled? → **Yes**
- Finish and reboot when prompted

Add User to GPIO Group

```
sudo usermod -aG gpio $USER
```

Log out and back in for this to take effect.

8. Clone & Setup Dashboard Project

Create Project Directory

```
mkdir -p ~/projects

cd ~/projects
```

Clone the Dashboard Repository

```
git clone https://github.com/YOUR_USERNAME/klair-dashboard.git

cd klair-dashboard
```

ℹ️ Note: Replace with your actual repository URL once the dashboard is developed and pushed to GitHub.

Install Dependencies

```
npm install

# or if using pnpm:

pnpm install
```

Configure Environment Variables

Create a .env.local file with your API keys:

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

nano .env.local
```

Fill in your API keys:

```
OPENWEATHERMAP_API_KEY=your_key_here

GOOGLE_CLIENT_ID=your_client_id

GOOGLE_CLIENT_SECRET=your_client_secret

JELLYFIN_URL=http://100.103.162.79:8096

JELLYFIN_API_KEY=your_jellyfin_key

HOME_ASSISTANT_URL=http://100.103.162.79:8123

HOME_ASSISTANT_TOKEN=your_ha_token
```

Build the Production Version

```
npm run build
```

Test the Dashboard

```
npm run start
```

Open a browser and visit `http://localhost:3000` to verify it works.

9. Configure Kiosk Mode (Auto-Start)

Kiosk mode makes the Pi boot directly into the dashboard in full-screen, with no desktop visible.

Step 1: Create Dashboard Service

Create a systemd service to run the Next.js server:

```
sudo nano /etc/systemd/system/klair-dashboard.service
```

Paste the following content:

```
[Unit]

Description=Klair Dashboard

After=network.target


[Service]

Type=simple

User=parminder

WorkingDirectory=/home/parminder/projects/klair-dashboard

ExecStart=/usr/bin/npm run start

Restart=on-failure

RestartSec=10

Environment=NODE_ENV=production

Environment=PORT=3000


[Install]

WantedBy=multi-user.target
```

Enable and start the service:

```
sudo systemctl enable klair-dashboard

sudo systemctl start klair-dashboard
```

Step 2: Create Hardware Sidecar Service

Create a service for the Python hardware controller:

```
sudo nano /etc/systemd/system/klair-hardware.service
```

Paste the following:

```
[Unit]

Description=Klair Hardware Controller

After=network.target


[Service]

Type=simple

User=root

WorkingDirectory=/home/parminder/projects/klair-dashboard/hardware

ExecStart=/usr/bin/python3 server.py

Restart=on-failure

RestartSec=10


[Install]

WantedBy=multi-user.target
```

Enable and start:

```
sudo systemctl enable klair-hardware

sudo systemctl start klair-hardware
```

⚠️ Note: The hardware service runs as root because NeoPixel requires direct hardware access.

Step 3: Configure Auto-Start Browser (Kiosk)

Create an autostart script for Chromium in kiosk mode:

```
mkdir -p ~/.config/autostart

nano ~/.config/autostart/kiosk.desktop
```

Paste the following:

```
[Desktop Entry]

Type=Application

Name=Klair Dashboard Kiosk

Exec=/home/parminder/kiosk.sh

X-GNOME-Autostart-enabled=true
```

Create the kiosk launch script:

```
nano ~/kiosk.sh
```

Paste:

```
#!/bin/bash


# Wait for dashboard server to be ready

sleep 10


# Disable screen blanking

xset s off

xset -dpms

xset s noblank


# Hide cursor when idle

unclutter -idle 0.5 -root &


# Launch Chromium in kiosk mode

chromium-browser \
```

```
  --kiosk \

  --noerrdialogs \

  --disable-infobars \

  --disable-session-crashed-bubble \

  --disable-restore-session-state \

  --no-first-run \

  --start-fullscreen \

  --autoplay-policy=no-user-gesture-required \

  http://localhost:3000
```

Make it executable:

```
chmod +x ~/kiosk.sh
```

Step 4: Disable Screen Blanking Permanently

```
sudo nano /etc/lightdm/lightdm.conf
```

Find or add under [Seat:*]:

```
xserver-command=X -s 0 -dpms
```

10. Display Configuration

Touch Display 2 Settings

The Raspberry Pi Touch Display 2 should work out of the box. If you need to adjust settings:

```
sudo nano /boot/firmware/config.txt
```

Useful Display Settings

Setting	Effect
`lcd_rotate=2`	Rotate display 180° (if upside down)
`display_lcd_rotate=2`	Alternative rotation method
`disable_splash=1`	Disable rainbow splash on boot

Adjust Brightness (Software)

The Touch Display 2 supports software brightness control:

```
# Set brightness (0-255)

echo 128 | sudo tee /sys/class/backlight/*/brightness



# Read current brightness

cat /sys/class/backlight/*/brightness
```

The dashboard can control this programmatically based on presence detection.

11. Final Checklist

Before rebooting into kiosk mode, verify everything:

Services Running

```
sudo systemctl status klair-dashboard

sudo systemctl status klair-hardware
```

Both should show 'active (running)'.

Hardware Test

- mmWave sensor detects presence (test script from hardware guide)
- NeoPixel LEDs light up (test script from hardware guide)
- Touch screen responds to input

Network & APIs

- Pi connected to WiFi
- Can reach N100 server (ping 100.103.162.79)
- Weather API returns data
- RSS feeds loading

Final Reboot

```
sudo reboot
```

The Pi should boot directly into the dashboard in full-screen kiosk mode.

✓ Success! If everything is working, your dashboard is ready. The Pi will automatically start the dashboard on every boot.

Appendix A: Useful Commands

Service Management

Command	Purpose
`sudo systemctl restart klair-dashboard`	Restart dashboard server
`sudo systemctl stop klair-dashboard`	Stop dashboard server
`journalctl -u klair-dashboard -f`	View dashboard logs (live)
`journalctl -u klair-hardware -f`	View hardware logs (live)

Exit Kiosk Mode

To exit kiosk mode and access the desktop:

- SSH into the Pi from another computer
- Run: `pkill chromium`
- Or press Alt+F4 on a connected keyboard

Update Dashboard

```
cd ~/projects/klair-dashboard

git pull

npm install

npm run build

sudo systemctl restart klair-dashboard
```

Appendix B: Troubleshooting

Dashboard Won't Start

- Check logs: `journalctl -u klair-dashboard -n 50`
- Verify Node.js: `node --version`
- Check if port 3000 is in use: `sudo lsof -i :3000`

Can't SSH After Kiosk Setup

- SSH should still work—the kiosk only affects the display
- Try: `ssh parminder@klair-dashboard.local`
- If that fails, connect a keyboard and press Ctrl+Alt+F2 for a terminal

Screen Goes Blank

- Screen saver may have activated despite settings
- SSH in and run: `DISPLAY=:0 xset s off -dpms`
- Check lightdm config was saved correctly

WiFi Disconnects

- Disable WiFi power management:

```
sudo nano /etc/NetworkManager/conf.d/default-wifi-powersave-on.conf
# Set wifi.powersave = 2 (disable)
```

Out of Memory

- Chromium can be memory-hungry. Add swap:

```
sudo dphys-swapfile swapoff
sudo nano /etc/dphys-swapfile
# Set CONF_SWAPSIZE=1024
sudo dphys-swapfile setup
sudo dphys-swapfile swapon
```

— End of Software Setup Guide —