CatFeeder Deployment Guide:
0 to 1 Setup

Cat Feeder Group

Summary

This guide is prepared by the BME450W SP25 Cat Feeder Capstone Group at Penn State University, working in collaboration with AstroLabe LLC. It outlines how to configure a Raspberry Pi to host and run the remote-controlled CatFeeder system, which includes:

- A Python-based backend for controlling the stepper motor, speaker, and camera
- A Next.js-based frontend interface for remote interaction
- Systemd automation to ensure the app stack runs at boot
- A Wi-Fi hotspot mode for deployment without internet

The guide is intended to be used for deployment and support of the CatFeeder in environments with or without active Wi-Fi networks.

—

1 Hardware Setup: Raspberry Pi, Control Board, and Stepper Motor

Goal: Connect the stepper motor to the Raspberry Pi using GPIO pins and a control board for motorized food dispensing.

1. Components Required

- Raspberry Pi 5
- Stepper motor
- Motor driver board
- Jumper wires
- External 5V–12V power supply
- ArduCam Pi Camera
- USB Speaker

2. GPIO Pin Assignments (BCM numbering)

GPIO Pin	Physical Pin
GPIO17	Pin 11
GPIO18	Pin 12
GPIO22	Pin 15
GPIO23	Pin 16

3. Wiring Example with ULN2003 Driver

- GPIO17 → IN1
- GPIO18 → IN2
- GPIO22 → IN3
- GPIO23 → IN4
- GND → GND
- 5V → VCC (or use external power)

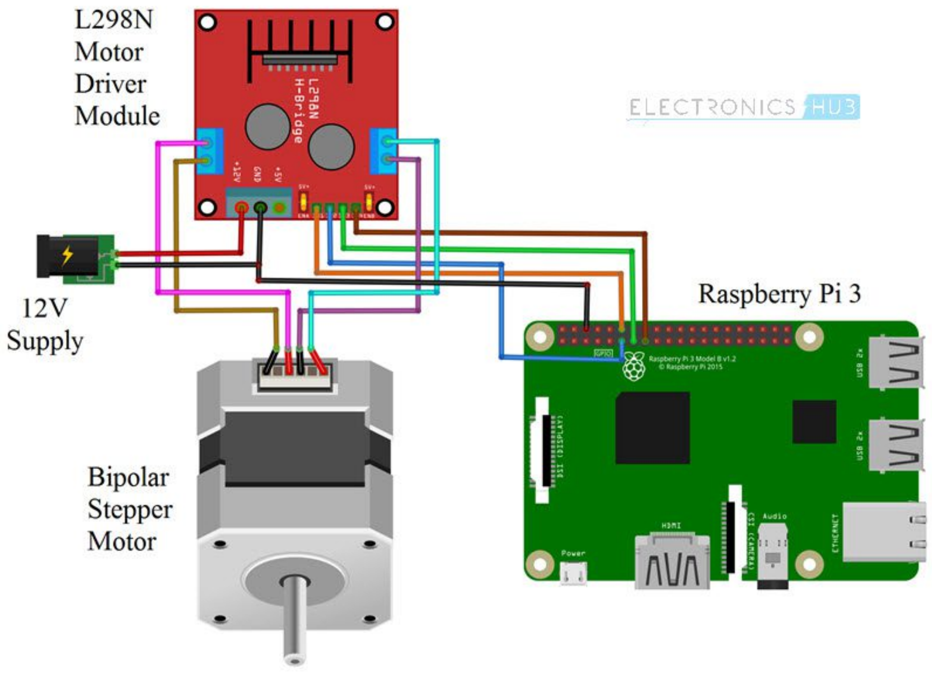

Figure 1: Wiring diagram: Raspberry Pi to stepper motor controller using GPIO pins

4. Example Python Script to Drive Stepper Motor (After flashing and SSH, See Sction 2

This Python script uses GPIO pins 17, 18, 22, and 23 to drive a stepper motor using a full-step sequence. Adjust the number of steps and delay as needed to rotate the feeder correctly.

```python
import RPi.GPIO as GPIO
import time

# Pin Definitions
IN1 = 17
IN2 = 18
IN3 = 22
IN4 = 23

# Stepper motor sequence (full-step)
sequence = [
    [1, 0, 0, 1],
    [1, 0, 0, 0],
    [1, 1, 0, 0],
    [0, 1, 0, 0],
    [0, 1, 1, 0],
    [0, 0, 1, 0],
    [0, 0, 1, 1],
    [0, 0, 0, 1]
]

def setup_gpio():
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(IN1, GPIO.OUT)
    GPIO.setup(IN2, GPIO.OUT)
    GPIO.setup(IN3, GPIO.OUT)
    GPIO.setup(IN4, GPIO.OUT)

def cleanup_gpio():
    GPIO.cleanup()

def step_motor(step: int):
    GPIO.output(IN1, sequence[step][0])
    GPIO.output(IN2, sequence[step][1])
    GPIO.output(IN3, sequence[step][2])
    GPIO.output(IN4, sequence[step][3])

def rotate_motor(steps: int, delay: float):
    for _ in range(steps):
        for step in range(8):
            step_motor(step)
            time.sleep(delay)

if __name__ == "__main__":
    setup_gpio()
    try:
        print("Rotating motor...")
        rotate_motor(50, 0.01)  # Adjust steps and delay as needed
    except KeyboardInterrupt:
        pass
    finally:
        cleanup_gpio()
```

2 Flash Raspberry Pi Bookworm OS (64-bit Lite) No Desktop

Goal: Set up your SD card with a minimal OS for headless use.

1. Download Raspberry Pi Imager: `https://www.raspberrypi.com/software`
2. Choose OS: `Raspberry Pi Bookworm OS Lite (64-bit) No Desktop`
3. Click the gear icon and configure:
   - Enable SSH
   - Set username/password
   - Configure Wi-Fi

   Set Specific Login Credentials

   - Username: `CatFeeder`
   - Password: Create your own (or `raspberry` if not changed)
4. Write the image and insert the SD card into the Pi

Connecting to the Raspberry Pi via SSH

After booting the Pi, you can connect remotely using SSH.

Method 1: Using Hostname (Recommended)

If Avahi/mDNS is working, use the following command from your computer:

```
ssh CatFeeder@raspberrypi.local
```

Note: This assumes:

- The Pi was set up with SSH enabled via Raspberry Pi Imager
- The hostname is `raspberrypi`
- Both devices are on the same network

Method 2: Using the Pi's IP Address

If the '.local' hostname doesn't resolve:

1. Run:
   ```
   arp -a
   ```
2. Or check your Wi-Fi router's admin dashboard for connected devices
3. SSH using:
   ```
   ssh CatFeeder@192.168.X.X
   ```

Resolving SSH Host Key Mismatch

If you previously connected to a Pi with the same hostname or IP, and re-flashed it, you may see an error like:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

Fix: Remove the old host key entry:

```
ssh-keygen -R raspberrypi.local
# or if using IP:
ssh-keygen -R 192.168.X.X
```

Then try connecting again:

```
ssh CatFeeder@raspberrypi.local
```

3 Initial Boot and System Update

Goal: Access your Pi and update all system packages.

```
ssh CatFeeder@raspberrypi.local
sudo apt update && sudo apt upgrade -y
```

4 Install Required Dependencies

Goal: Install Node.js, Git, Python, and Avahi for app hosting and device discovery.

```
# Node.js and npm
curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash -
sudo apt install -y nodejs

# Python, Git, and Avahi
sudo apt install -y python3 python3-venv git avahi-daemon
sudo systemctl enable avahi-daemon
sudo systemctl start avahi-daemon
```

5 Clone Application Repositories

Goal: Clone your app source code to the Pi.

```
cd /home
sudo mkdir CatFeeder && sudo chown $USER CatFeeder
cd CatFeeder

git clone https://github.com/Codakshay/CatFeederCapstone.git
```

6 Set Up Python Virtual Environments

Goal: Prepare isolated Python environments for your backend apps.

```
# Camera app
cd /home/CatFeeder/camera-app
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
deactivate

# Speaker app
cd /home/CatFeeder/speaker
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
deactivate
```

7 Set Up Next.js Frontend

Goal: Install frontend dependencies and build the production app.

```
cd /home/CatFeeder/catfeeder
npm install
npm run build
```

8 Create a Startup Script

Goal: Start all apps (camera, speaker, and frontend) from one script.
File: `/home/CatFeeder/start-catfeeder.sh`

```
#!/bin/bash

# Start camera app
source /home/CatFeeder/camera-app/venv/bin/activate
python3 /home/CatFeeder/camera-app/app.py &
deactivate

# Start speaker app
source /home/CatFeeder/speaker/venv/bin/activate
python3 /home/CatFeeder/speaker/app.py &
deactivate

# Start frontend
cd /home/CatFeeder/catfeeder
npm start
```

```
chmod +x /home/CatFeeder/start-catfeeder.sh
```

9 Create a Systemd Service

Goal: Run your app stack automatically at boot.

File: /etc/systemd/system/catfeeder.service

```
[Unit]
Description=CatFeeder Full Stack Service
After=network-online.target NetworkManager-wait-online.service
Requires=NetworkManager-wait-online.service
Wants=network-online.target

[Service]
User=CatFeeder
Group=CatFeeder
WorkingDirectory=/home/CatFeeder
ExecStart=/home/CatFeeder/start-catfeeder.sh
Restart=always
RestartSec=5
Environment=NODE_ENV=production
Environment=PATH=/home/CatFeeder/.nvm/versions/node/v22.14.0/bin:/usr/bin
   ↪ :/usr/local/bin
Environment=HOST=0.0.0.0
Environment=PORT=3000

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

```
sudo systemctl daemon-reexec
sudo systemctl daemon-reload
sudo systemctl enable catfeeder.service
sudo systemctl start catfeeder.service
journalctl -u catfeeder.service -f
```

10 Set Up Wi-Fi Hotspot (Access Point Mode)

Goal: Allow other devices to connect to the Pi when no internet is available.

Create Hotspot Connection

```
\subsection*{1. Create Hotspot Connection}

Use the following commands to configure the Raspberry Pi as a Wi-Fi access
   ↪  point named \texttt{ClaudeNet}. This allows devices to connect to
   ↪ the Pi even in the absence of a home Wi-Fi network.

\begin{lstlisting}[language=bash]
# Create the Wi-Fi hotspot connection
sudo nmcli connection add type wifi ifname wlan0 con-name CatFeederHotspot
   ↪  autoconnect yes ssid ClaudeNet

# Set mode to Access Point and band
sudo nmcli connection modify CatFeederHotspot 802-11-wireless.mode ap
sudo nmcli connection modify CatFeederHotspot 802-11-wireless.band bg

# Set WPA2 security and password
sudo nmcli connection modify CatFeederHotspot wifi-sec.key-mgmt wpa-psk
sudo nmcli connection modify CatFeederHotspot wifi-sec.psk "
   ↪ StrongPassword123"

# Enable IP sharing (NAT + DHCP)
sudo nmcli connection modify CatFeederHotspot ipv4.method shared

# Activate the hotspot
sudo nmcli connection up CatFeederHotspot
```

Enable Hotspot at Boot

```
nmcli connection modify CatFeederAP connection.autoconnect yes
nmcli connection up CatFeederAP
```

11 Switch Between Hotspot and Wi-Fi Client

Goal: Switch from hotspot to home Wi-Fi or vice versa.

To Connect to Regular Wi-Fi

```
nmcli dev wifi list
nmcli dev wifi connect YourSSID password YourWiFiPassword
nmcli connection modify YourSSID connection.autoconnect yes
nmcli connection modify CatFeederAP connection.autoconnect no
```

To Re-enable Hotspot

```
nmcli connection down YourSSID
nmcli connection up CatFeederAP
```

12 Final Reboot Check

Goal: Verify startup services and network behavior.

```
sudo reboot

# After reboot:
nmcli connection show --active
systemctl status catfeeder.service
ip a
```