

BD-1 Animatronic Electronics, Mechanics & Wiring Guide (v4)

This version includes electronics, wiring, BOM, diagrams, test script, mechanical mounting guide, calibration, and systemd auto-start.

Mechanical Mounting & Calibration

Head Mechanics (3 DoF)

Yaw (left/right twist) via MG996R at the neck base. Tilt (forward/back nod) via MG996R. Cant (side lean) via MG996R or MG90S. Mount servos in a stacked joint order: base yaw → tilt yoke → cant axis.

Antennas (2 x SG90)

Each antenna pivots with a small hinge and is driven by an SG90 servo via a linkage rod. Travel $\pm 20\text{--}30^\circ$ is sufficient. O-rings or friction washers help prevent jitter.

Stim Drawer (1 x MG90S)

Drawer can be actuated via a rack & pinion or a pushrod linkage. Use a small neodymium magnet to keep it closed when not actuated.

Horn Orientation & Neutral

Always set servos to 90° in software before mounting horns. Fit horns so the joints are mechanically centered at neutral. Tighten screws with threadlocker.

Travel Limits

Increment servo angles in software to find safe limits. Record min/max and keep $\sim 5\text{--}10^\circ$ margin. Store in a Python dict to clamp future moves.

Calibration Procedure

1. Run a script to set all channels to 90° before installing horns.
2. Install horns so joints are neutral.
3. Find travel limits per axis by slowly testing angles.
4. Record limits and clamp in software.
5. Test LEDs and audio separately before full integration.

```
from adafruit_servokit import ServoKit
kit = ServoKit(channels=16)
for ch in [0,1,2,3,4,5,6]:
    kit.servo[ch].angle = 90
print("All channels set to 90°")
```

Auto-Start on Boot (systemd)

You can configure BD-1 to auto-start on boot using a systemd service.

Directory Layout

```
/home/pi/bd1/  
■ bd1.py  
■ sounds/bd1_sound.mp3  
■ requirements.txt  
■ logs/
```

systemd Service File

```
[Unit]  
Description=BD-1 Control  
After=network-online.target sound.target  
Wants=network-online.target  
  
[Service]  
Type=simple  
User=pi  
WorkingDirectory=/home/pi/bd1  
Environment=PYTHONUNBUFFERED=1  
ExecStart=/home/pi/bd1/venv/bin/python /home/pi/bd1/bd1.py  
Restart=always  
RestartSec=3  
StandardOutput=append:/home/pi/bd1/logs/bd1.out.log  
StandardError=append:/home/pi/bd1/logs/bd1.err.log  
  
[Install]  
WantedBy=multi-user.target
```

Example bd1.py Skeleton

```
from adafruit_servokit import ServoKit
import board, neopixel, time, pygame, signal, sys

LIMITS = {
    0:(55,125), # head_tilt
    1:(60,120), # head_cant
    2:(60,120), # head_yaw
    3:(60,120), # antenna L
    4:(60,120), # antenna R
    5:(60,100), # drawer
}

kit = ServoKit(channels=16)
pixels = neopixel.NeoPixel(board.D18, 10, auto_write=False, brightness=0.5)
pygame.mixer.init()

def clamp(ch, ang):
    lo, hi = LIMITS.get(ch, (0,180))
    return max(lo, min(hi, ang))

def center_all():
    for ch in LIMITS.keys():
        kit.servo[ch].angle = clamp(ch, 90)

def set_back_and_eyes(back=(0,0,255), eye=(0,0,255)):
    for i in range(8): pixels[i] = back
    pixels[8] = eye; pixels[9] = eye
    pixels.show()

def cleanup(*_):
    pixels.fill((0,0,0)); pixels.show()
    center_all()
    sys.exit(0)

signal.signal(signal.SIGTERM, cleanup)
signal.signal(signal.SIGINT, cleanup)

if __name__ == "__main__":
    center_all()
    set_back_and_eyes()
    while True:
        time.sleep(1)
```