Hardware Roadmap

**Phase 0 – Gather & sanity-check the parts (½ day)**

| **Item** | **What to look for** | **Why you need it** |
| --- | --- | --- |
| Raspberry Pi 3 B (or 3 B+) in clear case | HDMI bootscreen, healthy 5 V rail | The brain |
| Boe-Bot chassis | Two Parallax continuous-rotation servos already bolted to frame | Mobile base |
| Tiny white breadboard on green carrier | Makes servo + sensor wiring painless |  |
| 2× long black **IR LED/receiver pairs** on front standoffs | Optional line/obstacle sensors |  |
| BASIC Stamp board (likely “Board of Education”) | We’ll remove or re-use |  |
| Power sources | • USB power bank (≥2 A) • 4×AA holder (if present) | Separate rails for Pi & servos |
| Jump wires, small Phillips screwdriver, side cutters | — | Tools |

Do a *dry fit*: verify each servo’s 3-pin header is labelled **S (white/yellow)** / **+ (red)** / **– (black)**.

**Phase 1 – Free the chassis (30 min)**

1. **Remove the BASIC Stamp**  
   *Unplug its 9 V battery clip, then back out the four screws.*  
   Keep the board in a static bag—you might want it later for another project.
2. **Check the servo leads**  
   They should exit up through the breadboard carrier. If they point downward, reroute them so the white/yellow signal wires reach the Pi easily.
3. **Tighten wheel screws**  
   Ensure the wheels don’t wobble; loose hubs cause drift you’ll blame on algorithms later!

**Phase 2 – Power architecture (45 min)**

**Option A – One USB power bank, step-up for servos**

*(simplest if you only have a power bank)*

1. **USB-C/µ-B → Pi** as usual.
2. Add a cheap **USB booster board** that outputs 6 V from the same bank → servo “+”.
3. Tie booster **GND** to Pi **GND** so PWM signals share a reference.

**Option B – Two supplies (rock-solid)**

* Pi runs from the power bank.
* Servos run from 4×AA NiMH pack (≈5.2 V fresh).
* **Must** connect AA pack **GND** to Pi **GND**.

**Tip:** stick a small toggle switch in-line with the servo pack so you can cut motor power without yanking cables.

**Phase 3 – Mounting the Pi on the chassis (20 min)**

* Fastest: industrial Velcro on the clear case lid → top plate.
* Neater: 4× ½-inch M3 standoffs through the acrylic, nuts on underside.
* Leave the CSI ribbon port facing forward if you’ll add a camera.

**Phase 4 – Signal wiring (1 hour)**

| **Connection** | **Pi Pin (BCM)** | **Breadboard row** |
| --- | --- | --- |
| **Left servo signal** | GPIO 18 | White/yellow wire |
| **Right servo signal** | GPIO 19 | White/yellow wire |
| **IR-LED/receiver left OUT** | GPIO 23 | (if you plan to use) |
| **IR-LED/receiver right OUT** | GPIO 24 | (optional) |
| **Shared GND** | Any Pi ground (pin 6, 9, 14, 20…) | Black servo wires & sensor GND |
| **Servo V+ rail** | From booster or AA pack | Red servo wires |

*Use the Pi’s 3.3 V logic directly—continuous-rotation servos accept it.*

**Phase 5 – Quick-and-dirty validation (40 min)**

1. **Boot the Pi** (headless SSH is fine).
2. sudo pigpiod to launch the precise PWM daemon.
3. Create servo\_test.py exactly as we did earlier and run it:

bash

CopyEdit

python3 servo\_test.py

*Expect the robot to roll ~20 cm forward then back.*

1. If one wheel spins backward: swap left.value sign or detach the wheel and spin it 180 °.
2. If nothing moves, double-check:
   * Booster or AA pack actually on
   * GND common between Pi and servo rail
   * GPIO numbers match the code.

**Phase 6 – Optional sensors & camera (1 hour)**

**IR pairs**

* Unscrew each black cylindrical board → note which pin is **OUT** (label on PCB).
* Connect OUT to GPIO 23/24; Vcc to servo + (5 V is fine); GND already common.
* gpiozero.DigitalInputDevice(23) should read **1 (open)** and go **0** when pointed at a close object or black tape.

**Pi Camera**

* Mount on LEGO tower or a foam-core upright zip-tied to front standoff.
* Plug ribbon into CSI port; enable with sudo raspi-config.
* libcamera-hello should show live video.

**Milestone check**

✅ Servos respond to Pi PWM  
✅ Chassis moves without brown-outs  
✅ At least one sensor or camera feeds data to Python

You’re now ready to jump into **software integration**: PID line following, tag detection, graph building, etc.