BeagleY-AI Pan-Tilt Kit Quick Start

This guide uses the **Dagu RS002B Pan-Tilt Kit**, with the following specifications:

- Weight: 8g
- Torque: 1.5kg.cm @ 4.8V
- Rotation: ~180 degrees
- Speed: 0.1sec/60degrees
- Max Voltage: 6V

Kit Assembly

Assemble the servo system following the printed instructions included in the kit.

Notes

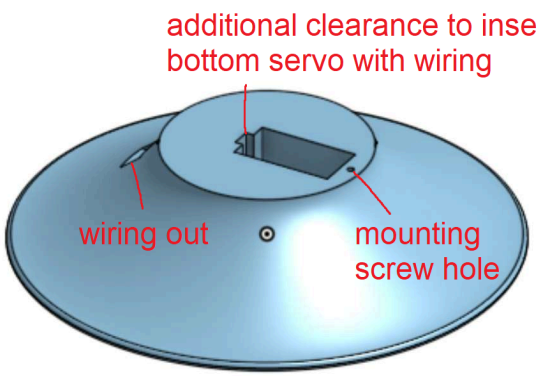
Before mounting each servo onto its metal bracket, **manually test** the minimum and maximum rotation angles. This ensures you won't block full movement after mounting, which is especially important for the tilt servo, where misalignment may require software limits to avoid damaging the servo and the completed assembly.

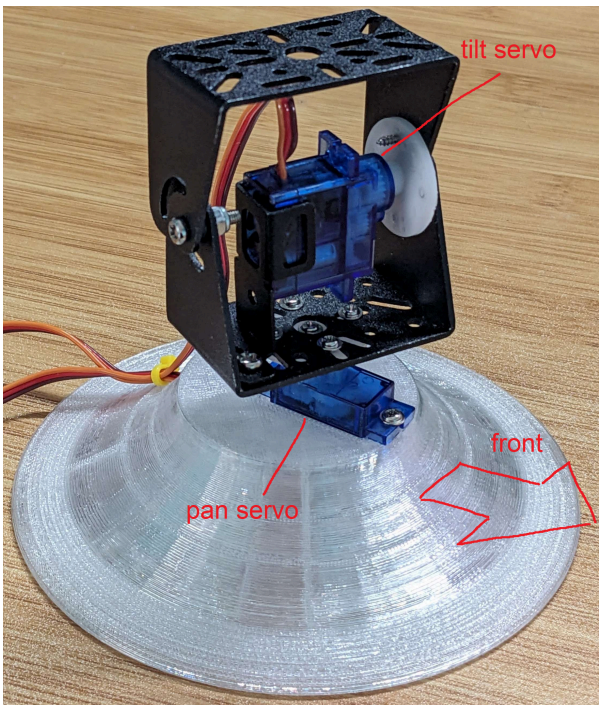
For step 4, when affixing the top (tilt) mount to the bottom (pan) mount:

- Use enough M3 washers at the rotating metal joint to provide clearance for the metal mounting screws.
- Take care to not overtighten the two screws at either end of the pivot. They should be secure, but loose enough to allow smooth motion.
- The servos apply minimal torque, just enough for a small webcam or sensors. Any unnecessary friction at the joint will cause jerky or stuttering motion, and possibly damage the servo.

Mounting Base

A stand may be designed using any CAD tool (OnShape is free to use) and 3D printed. Secure it to the base pan servo using the screw holes on the side wings.





Design tips:

- Ensure that the base is wider than the assembly with camera and other assembled components, to support the centre of gravity adequately.
- Add a slot for a weight within the base and/or rubber feet to increase stability.
- Make use of the 3d printers at SFU Makers Lab if you don't own a 3D printer

Pictures of an example design, and the 3D printed base with fully assembled pan-tilt kit are attached above

Wiring

Signal (orange wire)

For PWM control, there are only 3 free PWM-capable GPIO pins on the Zen hat:

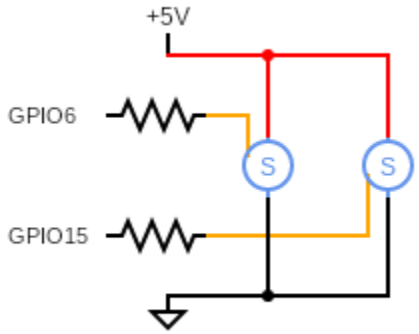
```
GPIO6  (PYMNL_3)-> PWM1_A
GPIO14 (UART_TX)-> PWM0_B
GPIO15 (UART_RX)-> PWM0_A
```

Each is on a separate PWM controller, so all 3 may be used concurrently.
Refer to the BY-AI and Zen Hat pinout for more information.

Up to 470Ωs of resistance may be connected in series along the 3.3V control line with each servo. This is optional but common practice to protect from short surges. Higher resistance may cause degraded signals due to waveforms being compressed

Power (red wire)

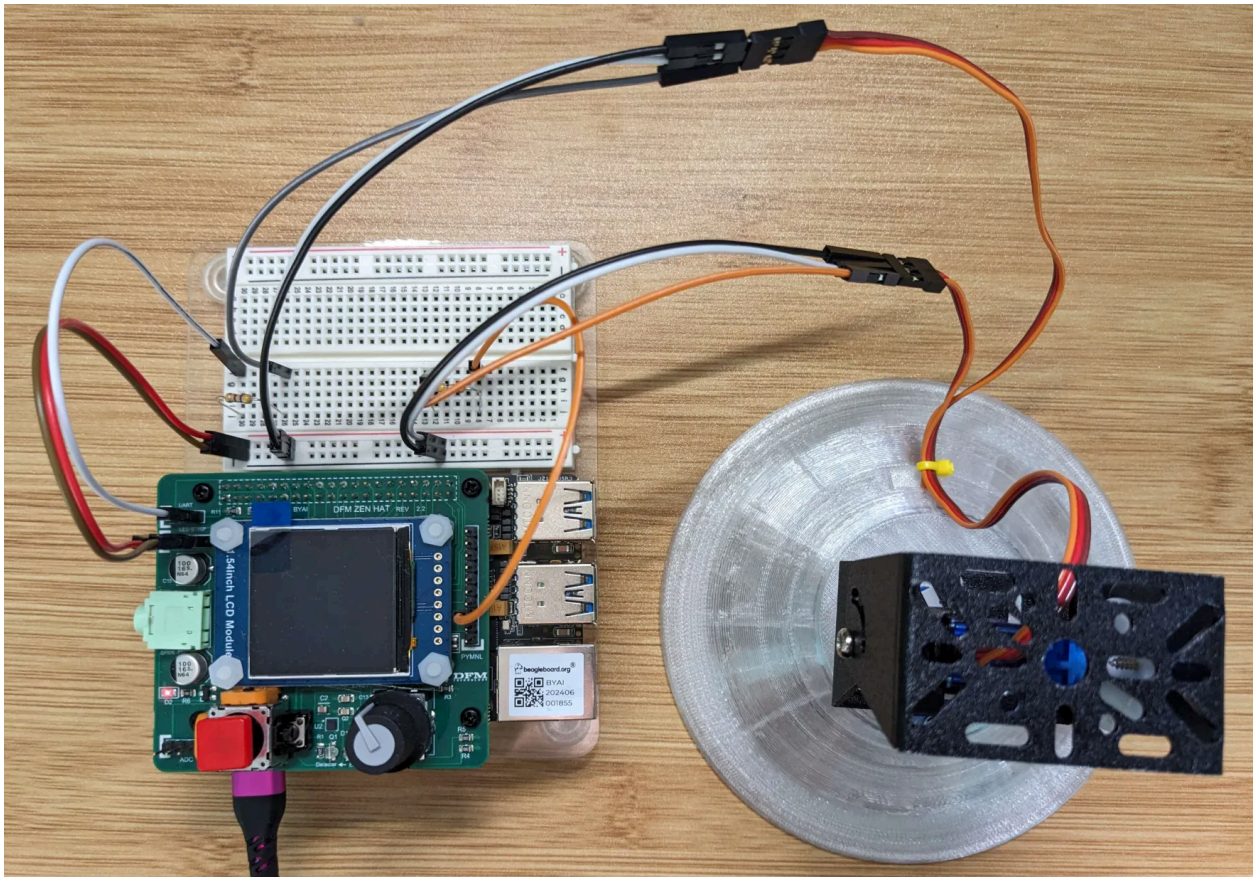
Use 5V power, **not 3.3V**, for the servos. On the Zen Hat, only one 5V pin is exposed (LED Strip Header pin 3) which must be used to power the servos. Undervolting may cause erratic motion or damage the board.



Ground (brown wire)

There are multiple ground pins on the Zen Hat. LED Strip Header pin 1 or 4 may be used for its proximity to the power.

A sample circuit diagram is depicted above, and a fully wired functioning system using GPIO6 and GPIO15 is shown below:



Usage

Firmware

Load device tree overlays:

```
$ nano /boot/firmware/extlinux/extlinux.conf
```

Append any relevant overlays, space separated, to the `fdtoverlays` line:

```
k3-am67a-beagley-ai-pwm-epwm0-gpio14.dtbo
k3-am67a-beagley-ai-pwm-epwm0-gpio15.dtbo
K3-am67a-beagley-ai-pwm-epwm1-gpio6.dtbo
```

If using **both GPIO14 and 15**, use:

```
k3-am67a-beagley-ai-pwm-epwm0-gpio15-gpio14.dtbo
```

GPIO14 & 15 are **also used by UART**, so a USB-UART Debug Probe may not be used at the same time. Double check that the following overlay is not present in the `fdtoverlays` line:

```
k3-am67a-beagley-ai-uart-ttyama0.dtbo
```

To apply firmware changes:

```
$ sudo reboot
```

Export pins

Prior to usage, ensure that the pins are configured for PWM. This must be done **upon each reboot**:

```
$ sudo beagle-pwm-export --pin hat-31 # GPIO6
$ sudo beagle-pwm-export --pin hat-8  # GPIO14
$ sudo beagle-pwm-export --pin hat-10 # GPIO15
```

If you get an error `Unknown pin name: [--pin`, ensure the overlay was added to `extlinux.conf`, and reboot

Command line testing

Most servos, including the RS002B, expect a 50 Hz signal (20 ms period) as convention, and a pulse width (duty_cycle) between 1-2 ms to represent from 0 to 180 degrees. This may vary slightly depending on boards and servos, so calibration may be needed for each servo and BY-AI used.

```
$ cd /dev/hat/pwm/
$ ls
```

Ensure that your chosen GPIO pins show up here. If not, double check that `beagle-pwm-export` was executed without error, and the correct overlays are present in `extlinux.conf`.

```
$ echo 0 > enable
```

The above may return a `-bash: echo: write error: Invalid argument` if the PWM is already disabled.

```
$ echo 0 > duty_cycle
```

Quick reminder for PWM: `0 <= duty_cycle < period`. Setting `duty_cycle` to 0 ensures that period will not be less than duty_cycle.

```
$ echo 20000000 > period # 20_000_000 ns (20 ms)
$ echo 1500000 > duty_cycle # 1 ms pulse
$ cat 1 > enable
```

This series of commands should rotate the servo to approximately 90 degrees. A duty_cycle of 1_000_000 should be ~0 degrees, and 2_000_000 should be ~180.

If the servo starts buzzing (pulsing but not moving), it has likely reached its maximum/minimum angle, or the duty_cycle is set too high/low. Set software limits in your code by testing these ranges manually first. For reference, here are the values from an example setup:

```
GPIO6 -> tilt servo: minimum=600_000, maximum=2_300_000
GPIO15 -> pan servo: minimum=500_000, maximum=2_300_000
```

Zephyr Note

If you're running Zephyr on the R5 core (e.g., to control a NeoPixel LED strip), note that after loading custom R5 firmware, the GPIO14 or 15 device files may still appear, but changes to the PWM signal will not affect the servo as expected. Ensure that `uart1` is disabled by appending the following into the in-project `.overlay` file

```
&uart1 {
    status = "disabled";
};
```

This is required due to UART being enabled by default in Zephyr's DTS.

Troubleshooting

Exporting pins returns an error:

- Double check the connected GPIO pins using Zen-Hat Schematic
- Double check the corresponding pin numbers for wired GPIO pins using BY-AI Pinout
- If the correct pin is not listed in the error message under `Possible PIN Options:`, check that the correct overlay is entered in `/boot/firmware/extlinux/extlinux.conf`, and board was rebooted to take effect

Expected GPIO folders are not listed under `/dev/hat/pwm`:

- Follow above steps for debugging exporting pins
- Try re-exporting pins using `beagle-pwm-export`

Setting PWM values returns `-bash: echo: write error: Invalid argument:`

- If writing 0 or 1 to `enable`, ignore the error, the argument is already applied
- For other cases, check that the argument is valid - `period` should be = 20_000_000, and `duty_cycle` must be between 0 and 20_000_000

Servo is not responsive but PWM files appear in GPIO folder as expected:

- Check that `cat enable` returns 1
- The servo may have reached its minimum or maximum value, try modifying `duty_cycle` to be within 1_000_000 and 2_000_000
- Double check wiring is as expected, with each wire connected to the correct pins: red -> 5V, brown -> ground, orange -> expected GPIO pin
  and that the combined resistance in the control line is not too high (above ~470 ohms)
- If using Zephyr, check the Zephyr note to prevent UART being enabled on GPIO 14 and 15

Servo is buzzing but not moving:

- Ensure that the pivot joint, or mount itself is not obstructed
- Double check the `period` is set to 20ms and `duty_cycle` is within 1 to 2ms

References

BY-AI pinout: https://pinout.beagleboard.io/pinout/pwm

Zen-Hat pinout:
https://opencoursehub.cs.sfu.ca/bfraser/grav-cms/cmpt433/guides/files_byai/ZenHatPinout.pdf
Zen-Hat schematic: https://opencoursehub.cs.sfu.ca/bfraser/solutions/433/zen-hat/HATV2.2.pdf
Base PWM Guide:
https://opencoursehub.cs.sfu.ca/bfraser/grav-cms/cmpt433/guides/files_byai/PWMGuide.pdf