# Guide on the Lynxmotion Pan & Tilt Kit

**Overview**:
This guide will focus on setting up the Lynxmotion Pan and Tilt kit. It will be based on the previous semester's guide, but instead, it will focus on applying the kit to the BeagleY-AI board instead of the previous Beaglebone Green.

---

**Hardware Setup**:
The Lynxmotion Pan and Tilt kit consists of 2 HS 422 servo motors, each composed of 3 wires: Ground (Black), Power (Red), and Signal (Yellow). The BeagleY-AI board consists of various pins that are available for each wire. However, if you also utilise the Zen Hat, those pins will most likely be occupied or redirected to the Zen Hat. The HS 422 servo motors require access to a 5V power pin. Based on the Zen hat schematics, the only accessible one is from the NeoPixel header. Regarding the ground, the Zen hat has multiple accessible ground pins from each of the various headers. For simplicity and accessibility, the pin used for the guide is next to the 5V pin on the Neopixel header.

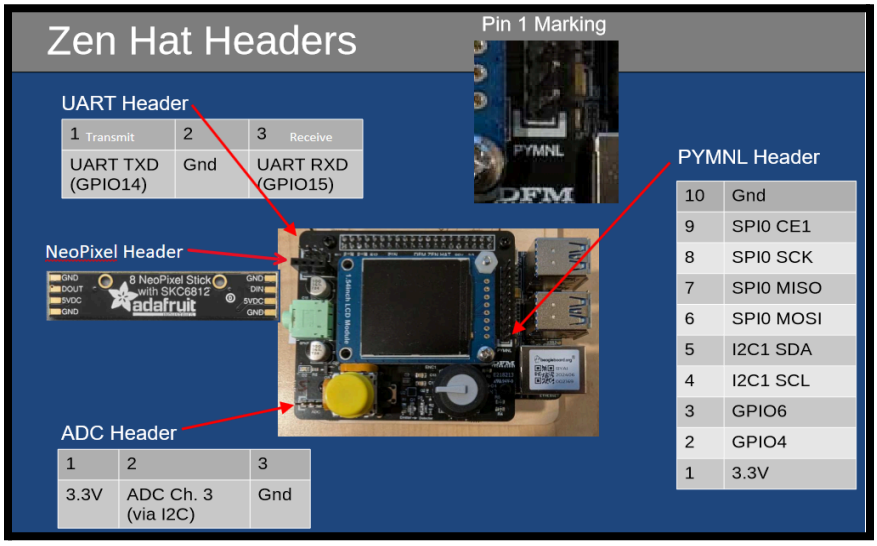


*Figure 1: Electronics Slide Deck – Slide 31 (with added Neopixel details on the left)*

We will utilise PWM, Pulse-width modulation, regarding the signal wire, to control the motor's movement/rotations. PWM generates digital waveforms by using a combination of period and duty cycles. For further details on PWM, refer to the PWM guide on the course page. The Zen hat has selected accessible GPIO pins to connect with the signal wire. For the walkthrough, we will be using GPIO pins 6 and 15. However, each GPIO pin requires 3.3V, and the motor uses 5V, so 2 470Ω resistors will be needed for each GPIO-to-motor connection. Below are visualisations of the hardware setup and circuit diagram.

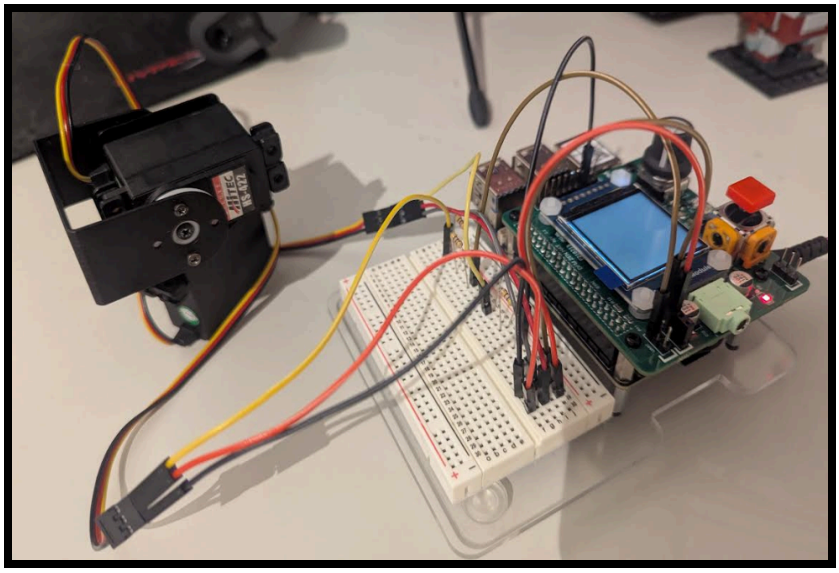

*Figure 2: Visualization of the complete circuit*

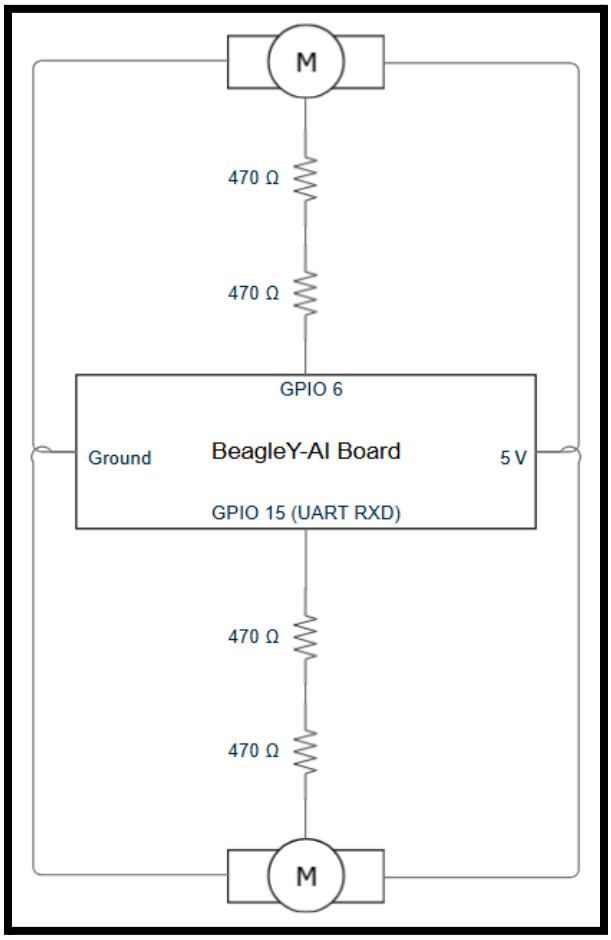


*Figure 3: Circuit Diagram*

**Software Setup**:

To initialise and make the GPIO pins accessible, you must first configure them for PWM usage.

- SSH into the Beagle-Y AI board

- Edit the extlinux.conf file to tell Linux to dedicate the pins to PWM

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

- Add the following bolded green text to ftdoverlays under microSD (default).
  - Note, this should be on the same line as fdtoverlays.
  - If there are already existing overlays, you can add multiple “fdtoverlay” lines or list them all on the same line.

```
label microSD (default)
kernel /Image
append console=ttyS2,115200n8 root=/dev/mmcblk1p3 ro ...
fdtdir /
fdt /ti/k3-am67a-beagley-ai.dtb
fdtoverlays ... /overlays/k3-am67a-beagley-ai-pwm-epwm1-gpio6.dtbo
/overlays/k3-am67a-beagley-ai-pwm-epwm0-gpio15.dtbo
initrd /initrd.img
```

- Reboot the board

```
byai $  sudo reboot
```

- Note: whenever you reboot the board, the HAT pin PWM symlink pin must be configured each time. Run the following commands to configure each pin using its HAT pins. To determine which HAT pin matches with your GPIO pin, use the Zen Hat Pinout or the BeagleY-AI Pinout

```
byai $  sudo beagle-pwm-export --pin hat-31 // For GPIO pin 6
byai $  sudo beagle-pwm-export --pin hat-10 // For GPIO pin 15
```

---

**Controlling the Servo Motors**

After initialising the GPIO pins, they are available to configure. The PWM files for each GPIO pin should have the same contents.

```
byai $ ls /dev/hat/pwm/gpio6
capture duty_cycle enable period polarity power uevent
```

```
byai $ ls /dev/hat/pwm/gpio15
capture duty_cycle enable period polarity power uevent
```

The main files we will manipulate are ***duty_cycle***, ***enable,*** and ***period***.

- ***duty_cycle*** will control how fast/much the motor turns based on the signal's height during each PWM cycle.
- ***enable*** will be used to turn the motor on and off.
- ***period*** is used to set the PWM cycle, and the upper limits for the duty cycle, which will determine how finely we control the motor's movement.

Before manipulating the duty_cycle, we need to ensure the period is set first to provide an upper bound on the duty_cycle. For our purposes, the period value we will use is 20000000 (20 million). After we set the duty_cycle, we enable the motor

```
byai $  cd /dev/hat/pwm/gpio15
byai $  echo 0 > duty_cycle
byai $  echo 20000000 > period
byai $  echo 1200000 > duty_cycle
byai $  echo 1 > enable
byai $  echo 0 > enable     // To turn it off
```

Some helpful duty_cycle values to try out:

```
byai $  echo 2500000 > duty_cycle  // Moves it to end
byai $  echo 630000  > duty_cycle  // Moves it to the opposite end
byai $  echo 1200000 > duty_cycle  // Moves it to the center
```

---

**Troubleshooting**

- If you get errors when "***sudo beagle-pwm-export…***", ensure you have edited the **extlinux.conf** file correctly and ensure you provided an available GPIO overlay.
  - To check all the possible overlay options:

    ```
    byai $ ls /boot/firmware/overlays/*beagley-ai-pwm*
    ```
- If any **GPIO#** folders aren't showing in "***/dev/hat/pwm/***", ensure you provided the correct GPIO configuration in extlinux.conf file properly, and you correctly configured the correct HAT pin PWM symlink pin (can double check on the Zen Hat Pin layout).
  - In the worst case, you can use other GPIO pins from the listed overlay options.
- If the motor isn't moving, ensure the wiring is correct and use the proper pins for ground, 5V power and GPIO.
- If you get "***invalid argument***" errors when setting the duty_cycle, it's most likely because your value is greater than the set period value.

**References**


- MG995R Servo Motor Guide by Felix Lionardo and Ryan Ly: https://opencoursehub.cs.sfu.ca/bfraser/grav-cms/cmpt433/links/files/2024-student-howtos/MG995RServoMotorGuide.pdf
- Lynxmotion Pan and Tilt Kit: https://ca.robotshop.com/products/lynxmotion-pan-and-tilt-kit-aluminium2
- PWM Guide for BeagleY-AI by Dr.Brian: https://opencoursehub.cs.sfu.ca/bfraser/grav-cms/cmpt433/guides/files_byai/PWMGuide.pdf
- ZenHat Pinout: https://opencoursehub.cs.sfu.ca/bfraser/grav-cms/cmpt433/guides/files_byai/ZenHatPinout.pdf
- BeagleY-AI pinout: https://pinout.beagleboard.io/


**Copyright Notice**