**Step-by-Step Plan for Motorizing the Telescope:**

**1. Stepper Motors and Mount Setup**

**Objective:** Attach motors to the telescope to enable movement along the Right Ascension (RA) and Declination (DEC) axes.

* **Parts Needed:**
  + NEMA 17 stepper motors (2)
  + L298N motor driver (2)
  + Motor mount brackets (custom 3D printed if needed)
* **Actions:**
  + **Motor Setup:** Install NEMA 17 stepper motors on the RA and DEC axes of your telescope mount.
  + **Motor Mounts:** Use 3D printed parts or existing brackets to mount the motors securely to the telescope.
  + **Connect Motors to Driver:** Wire the stepper motors to the L298N motor drivers for controlling their movement.
* **Goal:** Ensure that you can manually control the motors to move the telescope in both RA and DEC directions. This will set up the foundation for the motorized tracking.

**2. Powering the Motors**

**Objective:** Ensure your motors and Raspberry Pi have a reliable power source.

* **Parts Needed:**
  + 12V power supply for motors
  + Power bank or adapter for Raspberry Pi
* **Actions:**
  + **Connect Power Supply:** Provide 12V power to the L298N motor drivers to power the stepper motors.
  + **Power the Raspberry Pi:** Use a power bank or 5V power adapter for the Raspberry Pi to keep everything running continuously.
* **Goal:** Make sure both the stepper motors and the Raspberry Pi are powered and functional.

**3. Raspberry Pi Setup**

**Objective:** Set up the Raspberry Pi to control the telescope and interface with the motor drivers.

* **Parts Needed:**
  + Raspberry Pi 4
  + MicroSD card (for Pi OS)
  + USB keyboard, mouse, and monitor (for setup)
* **Actions:**
  + **Install OS on Raspberry Pi:** Flash Raspberry Pi OS onto a microSD card and set up the Raspberry Pi.
  + **Install Required Libraries:** Install libraries like RPi.GPIO (for motor control) and pigpio (for precision motor control).
  + **Connect to Motor Drivers:** Wire the Raspberry Pi GPIO pins to the L298N motor drivers for stepper motor control.
* **Goal:** Ensure you can send simple commands from the Raspberry Pi to the motor drivers to control motor movements.

**4. Telescope Movement Control**

**Objective:** Write basic Python code to control the RA and DEC motors via Raspberry Pi.

* **Parts Needed:**
  + Python (installed on Raspberry Pi)
  + Motor drivers and Raspberry Pi connected
* **Actions:**
  + **Basic Code for Movement:** Write a Python script to control the stepper motors using GPIO pins. This script will send pulse signals to the L298N drivers to rotate the motors.
  + **Test Movement:** Test the movement by rotating the motors in both RA and DEC directions. Check if the telescope responds correctly to forward and backward commands.
* **Goal:** Successfully control the movement of the telescope along the RA and DEC axes using the Raspberry Pi.

**5. Telescope Calibration**

**Objective:** Calibrate the telescope to ensure precise movement and correct positioning.

* **Actions:**
  + **Set Starting Position:** Place the telescope at a known starting position (e.g., pointing at a fixed object).
  + **Move in Small Steps:** Move the telescope in small increments to check the accuracy of the movement. Adjust if needed.
* **Goal:** Ensure that the telescope can move smoothly and consistently across the sky.

**6. Camera Integration**

**Objective:** Connect and control your Canon EOS 6D Mark 2 camera to take images remotely.

* **Parts Needed:**
  + Canon EOS 6D Mark 2
  + USB cable to connect camera to Raspberry Pi
* **Actions:**
  + **Set up Camera Control:** Install gPhoto2 on the Raspberry Pi to control the camera remotely. You can use the command line to trigger the camera to take photos at specific times or intervals.
  + **Test Camera Functionality:** Take a few test photos to ensure the Raspberry Pi can trigger the camera.
* **Goal:** Successfully control the Canon EOS camera to take images using the Raspberry Pi.

**7. Telescope and Camera Control via Python**

**Objective:** Write a Python script to integrate the camera with the motorized telescope, allowing you to take images at specific angles.

* **Actions:**
  + **Combine Code:** Combine the motor control and camera control in a single Python script.
  + **Take Photos at Specific Locations:** Make the telescope point to specific coordinates (RA and DEC) and then trigger the camera to take an image.
* **Goal:** Have a working system where you can move the telescope and take photos at specific coordinates.

**8. (Optional) Remote Control**

**Objective:** Set up a simple interface to control the telescope remotely.

* **Actions:**
  + **Web Interface or SSH:** Set up a web interface (Flask) or use SSH to control the telescope from another device.
  + **Monitor Movement:** Display current telescope position and allow for manual control from the interface.
* **Goal:** Have remote control capabilities for your motorized telescope.

**9. Final Testing**

**Objective:** Test the complete motorized telescope system.

* **Actions:**
  + **Move Telescope:** Test the full range of movement for both RA and DEC axes.
  + **Capture Images:** Take a few test images to ensure the camera is working and capturing the desired objects in the sky.
* **Goal:** Ensure that the telescope is fully operational, and the camera is capturing images as expected.

**10. (Future Step) Add Computer Vision (CV)**

Once the motorized telescope and camera are working, you can start adding computer vision. At that point, you'll be ready to implement star/constellation detection, track satellites or airplanes, and overlay trajectory lines.

**Estimated Time for Completion:**

* **1-2 Weeks** for setting up motorized control, powering components, and writing basic scripts.
* **1 Week** for testing, calibration, and integration with the camera.