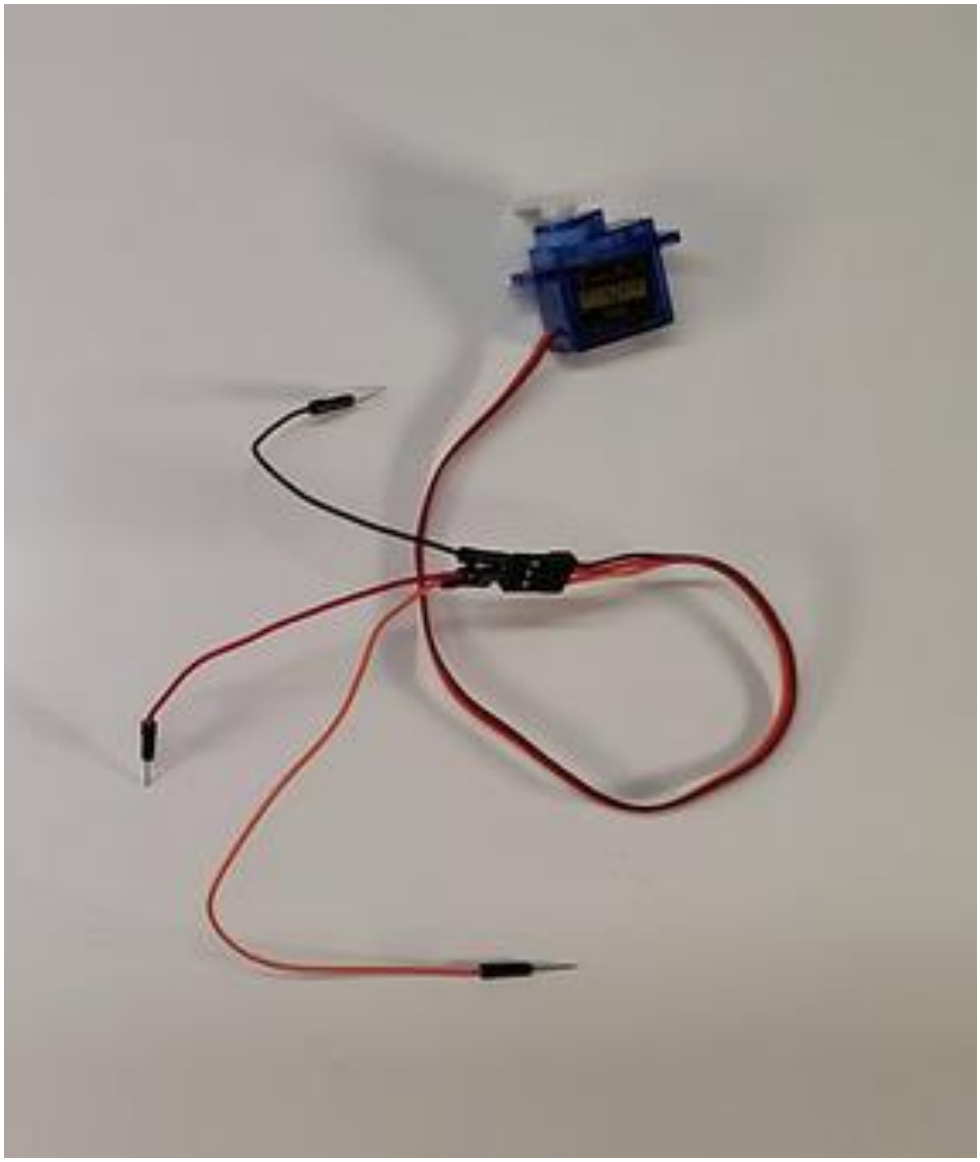
Connecting Servo Motor to BeagleBone Green for Movement Along Two Axes

by Parmveer Dayal, Jashanraj Singh Gosain, and Oliver Yalcin Wells

Created: April 18, 2024



Part 1: Required Parts

- SG90 Servo Motor
- BeagleBone Green (BBG)
- Breadboard
- Wires
- 2x 470 ohm Resistors

Part 2: Pulse-Width Modulation (PWM)

PWM is an efficient technique used for controlling power in electronics by generating digital wave forms. Digital wave form includes two main components: Period and Duty Cycle.

- Period: The duration of one complete cycle of the waveform
- Duty Cycle: The percentage of the cycle where the signal is high (or low, depending on configuration)

The exact values selected for period and duty cycle are dependent on what your goal is for your motor.

Listed below are the BBG's PWM channels:

Zen Cape Use	PWM Channel	BBB Pin	Linux Path[1]	Notes
Buzzer	PWM-0A	P9-22	/dev/bone/pwm/**0/a/**	These two share a PWM timer: Period must be the same; duty cycle is independently.
unused	PWM-0B	P9-21	/dev/bone/pwm/**0/b**	
Blue LED	PWM-1A	P9-14	/dev/bone/pwm/**1/a**	These two share a PWM timer.
Red LED	PWM-1B	P9-16	/dev/bone/pwm/**1/b/**	
Green LED	PWM-2A	P8-19	/dev/bone/pwm/**2/a**	These two share a PWM timer.
unused	PWM-2B	P8-13	/dev/bone/pwm/**2/b**	

Figure 1: BBG's PWM Channels
from: https://opencoursehub.cs.sfu.ca/bfraser/grav-cms/cmpt433/guides/files/PWMGuide.pdf

We chose to use PWM-2A (P8-19) and PWM-2B (P8-13) for our two motors. However, you can choose to use other ones from this list if you would like.

Part 3: Circuit Wiring

To establish a connection, first you need to connect the motor to the BBG itself. The circuit diagram below shows how to connect the servo to the BBG.

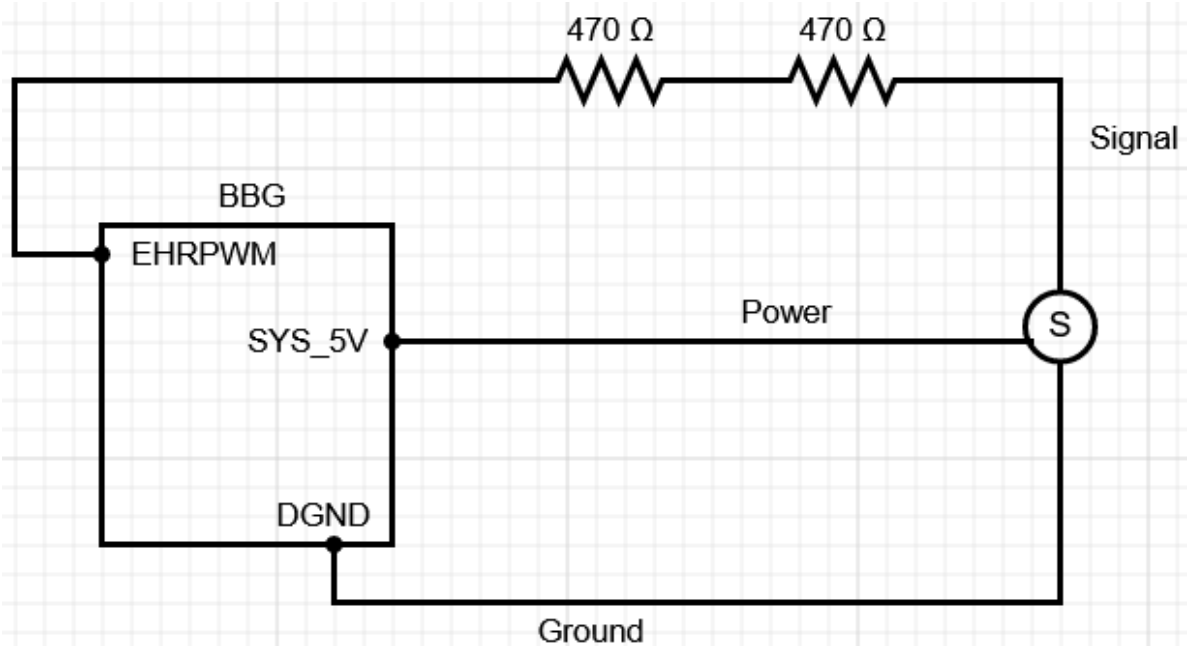


Figure 2: Wiring of the motor and BBG

Each servo motor is made up of three wires: Signal, Power, and Ground. The ground wire connects to a DGND pin, the power wire connects to a SYS_5V pin, and the signal wire connects to a EHRPWM pin. Below is a figure with all the pins available to choose from:

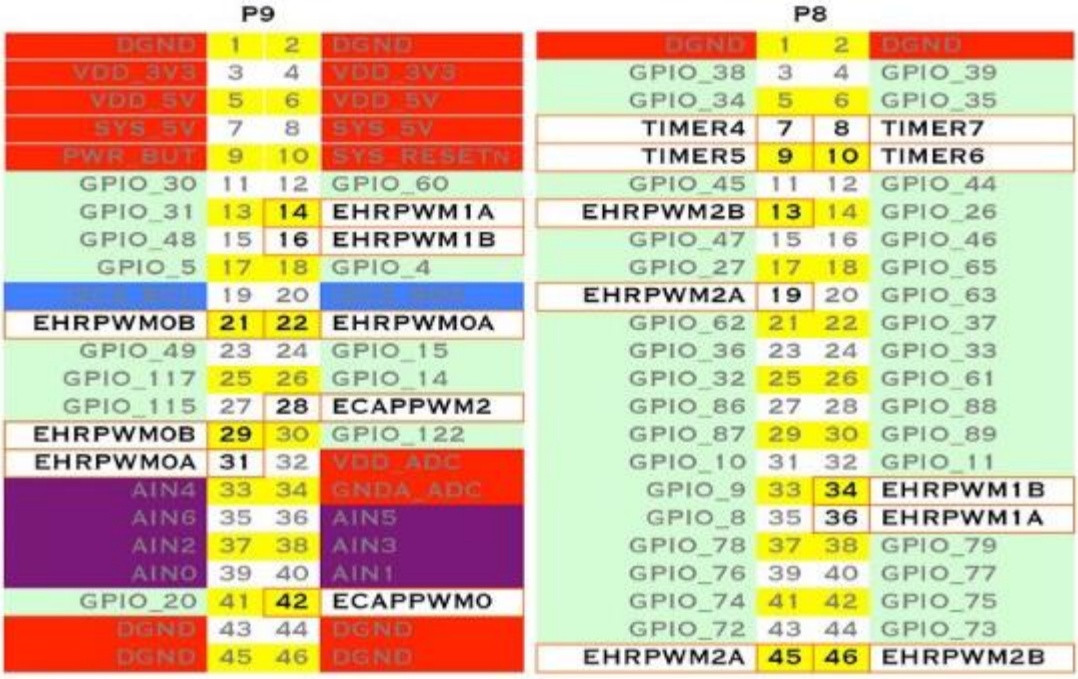


Figure 3: Available PWM pins for BBG
from: https://www.mouser.com/datasheet/2/744/Seeed_102010027-1217531.pdf

Any of the available DGND and EHRPWM pins can be selected, however, SYS_5V is limited to only pins P9-7 and P9-8. For our project, the wiring was:

- Connecting motor 1 (bottom) to P8-13 (since using PWM-2B)
- Connecting motor 2 (up) to P8-19 (since using PWM-2A)
- 5V power connected to P9-7 and P9-8 (SYS 5V pins)
- Connect ground wires to P9-1 and P9-2 (DGND pins)

Part 4: Motor Commands

1. SSH into the target machine (BBG)

2. Configure the pins corresponding to the targeted PWM channels. In this case since we are using channels PWM-2A and PWM-2B, we will configure pins P8-19 and P8-13.
   - (bbg)$ sudo config-pin p8_19 pwm
   - (bbg)$ sudo config-pin p8_13 pwm

3. View PWM files:
   - (bbg)$ ls /dev/bone/pwm/2/a
     capture duty_cycle enable period polarity power uevent
   - (bbg)$ ls /dev/bone/pwm/2/b
     capture duty_cycle enable period polarity power uevent

4. Navigate to the correct PWM channel file (2A for example) and set the period (time is in ns):
   - (bbg)$ cd /dev/bone/pwm/2/a
   - (bbg)$ echo 10000000 > period

5. Set the duty cycle time (ns):
   - (bbg)$ echo 5000000 > duty_cycle

6. Finally, enable the pin and the motor will move:
   - (bbg)$ echo 1 > enable

7. To turn off the motor, simply turn enable back to 0:
   - (bbg)$ echo 0 > enable

Part 5: Troubleshooting

- If the motor is not moving after adjusting the period and duty cycle times, make sure the wiring is correct and you have correctly selected and configured an available pin.
- We recommend adjusting the period and duty cycle values by finding the 0/180/360 degree angles first, as it will give you a better idea of how the motor works.
- "Invalid argument" when trying to enable the pin means you have not yet set the period and duty cycle values correctly. Go back and do steps 4 and 5 again (ensure you are in the correct directory as well).
- If motor is still not working after all these steps, try using a different PWM chip as suggested in this guide (https://opencoursehub.cs.sfu.ca/bfraser/grav-cms/cmpt433/links/files/2022-student-howtos/ServoPWMGuide_SG90_9gMicroServo.pdf). Disclaimer: changing chips did not work for us as we were only able to get it to work as described in our guide above.

A note on finding the correct duty cycle: The value will depend on the specific servo motors themselves. The most challenging part is figuring out when the servo motor duty_cycle is 0/180/360 degrees (we recommend figuring out these angles first). To use correct angles one must figure out the corresponding values.