

1) The PWM interface of the GPIO pins allows for analog frequencies to be sent from the Pi to the device. Why would the Pi not allow reading analog frequencies on a GPIO pin interface?

The Pi does not have a built-in analog-to-digital converter and all of its GPIO pins are digital pins. This means that the Pi would not be able to read analog signals or analog frequencies directly from its GPIO pins. The digital pins can only read high or low states and not the continuous voltage values that analog signals give off. If the Pi allowed reading analog frequencies using its existing GPIO interface the values would be incorrect. Therefore, that is why the PWM interface is used as the solution to this problem. The PWM interface switches the digital pin on and off rapidly which creates a signal that mimics analog voltages over time.

2) Motors have diodes in the circuit to prevent the flow of electricity in a direction that is not desired to protect the circuit. What might we need to be able to do to allow the basic DC motor to operate forward and backwards? How would you propose to address that need in software, ignoring the hardware requirements?

The current configuration for the breadboard and circuit uses a single transistor and diode, which only allows for one-directional movement. The transistor acts as a switch, enabling PWM to control the speed of the motor but not its direction. To allow the motor to operate both forward and backward, we would need an H-bridge circuit that uses four transistors controlled by two GPIO pins. This H-bridge configuration allows voltage to be applied in either direction across the motor.

In software, we can set one GPIO pin to high and the other to low to make the current flow in one direction, spinning the motor forward. Swapping the states of these same pins (setting the first pin low and the second high) reverses the direction of the current and in turn the motor direction. Using PWM on these GPIO pins would allow us to control both the speed and direction of the motor.

Source: (<https://docs.onion.io/omega2-maker-kit/maker-kit-servo-h-bridge.html>)