Nicholas Wyrwas

CS-350-13290-M01

Mr. Eric Gregori

07/03/2025

2-4 Milestone One: Pulse Width Modulation Lab

The first observation I made while experimenting with pulse width modulation is observing the frequency when the LED I was connected to actually started to visibly blink. This was usually around 30 Hz, as when the frequency is set to 60 Hz or above, the LED appears to be constantly on due to the persistence of vision. However, if I continue to lower the frequency step-by-step to 50 Hz, 40 Hz, and finally 30 Hz, I can begin to see individual flashes of the LED on and off. The blinking becomes clearly visible at around 10 to 20 Hz, which would be an appropriate frequency for visibly blinking an LED in a human context.

When I varied the duty cycle of a PWM signal, the brightness of the LED changed as a result. Starting from a 50% duty cycle, the LED was at a medium brightness level. When I decreased it to around 40% or lower, it became noticeably dimmer. As I continued to lower the duty cycle, once it was at around 20–30%, the LED’s brightness was significantly reduced, but still visible. This illustrates how PWM can be used to finely control the intensity of an output, such as an LED, without changing the voltage applied to it.

For the LED to fade smoothly, the loop in the script gradually increased or decreased the duty cycle by 5 every 0.1 seconds, allowing the LED to appear to fade in brightness. However, this would not be perceived as entirely smooth by the human eye. To make the transition appear smoother, I could reduce the increment size to 1 or 2 to have more granular steps in the fading process. Additionally, I could decrease the time between duty cycle updates from 0.1 seconds to 0.05 seconds, giving the illusion of more continuous motion by displaying more steps during the transition.

The function for setting the PWM frequency on a GPIO line is GPIO.PWM(pin, frequency), where pin is the number of the GPIO pin being used and frequency is the desired frequency in hertz. This creates a PWM instance with the specified frequency. The duty cycle is then set using the ChangeDutyCycle(value) method of the PWM object, where value is the desired duty cycle percentage for the signal.