

```

import time
import board
import pwmio
from adafruit_motor import servo
from adafruit_circuitplayground import cp

# Setup the PWM output for the servo on pin A1
pwm = pwmio.PWMOut(board.A1, duty_cycle=0, frequency=50)
my_servo = servo.Servo(pwm)

# Initialize the servo position
position = 0
my_servo.angle = position

# Indicate that the script is ready
cp.pixels.fill((255, 0, 0)) # Red indicates idle state
cp.pixels.show()
print("Press button A to start or repeat the action.")
print("Press button B to start or repeat the action.")

while True:
    # Wait for button A press
    if cp.button_a:
        # Indicate the start
        cp.pixels.fill((0, 0, 255)) # Blue indicates the action is starting
        cp.pixels.show()

        # Move the servo 18 degrees every second for 10 seconds
        position = 0 # Reset position
        my_servo.angle = position
        for i in range(10):
            position += 18
            if position > 180: # Ensure the position doesn't exceed the servo limit
                position = 0
            my_servo.angle = position
            time.sleep(1)

        # Light up all the CPX LEDs to green
        cp.pixels.fill((0, 255, 0)) # Green indicates the action is done
        cp.pixels.show()

        # Wait a moment to prevent double triggering
        time.sleep(0.5)

```

```
# Reset to idle state if button A is not pressed
cp.pixels.fill((255, 0, 0)) # Red indicates idle state
cp.pixels.show()
time.sleep(0.1)

# Wait for button B press
if cp.button_b:
    # Indicate the start
    cp.pixels.fill((0, 0, 255)) # Blue indicates the action is starting
    cp.pixels.show()

    # Move the servo 9 degrees every second for 20 seconds
    position = 0 # Reset position
    my_servo.angle = position
    for i in range(20):
        position += 9
        if position > 180: # Ensure the position doesn't exceed the servo limit
            position = 0
        my_servo.angle = position
        time.sleep(1)

    # Light up all the CPX LEDs to green
    cp.pixels.fill((0, 255, 0)) # Green indicates the action is done
    cp.pixels.show()

    # Wait a moment to prevent double triggering
    time.sleep(0.5)

# Reset to idle state if button B is not pressed
cp.pixels.fill((255, 0, 0)) # Red indicates idle state
cp.pixels.show()
time.sleep(0.1)
```