Solutions to Raspberry Pi Limited Number of PWM Pins

Problem:

Raspberry Pi has only 2 hardware PWM pins (GPIO12 & GPIO13). If more are needed (e.g., for multiple motors or servos), alternatives are required.

Possible Solutions

1. **Software PWM (via RPi.GPIO or pigpio)**

    - Pros:
        - Unlimited PWM channels.
        - Easy to implement.
    - Cons:
        - Less precise (CPU-dependent).
        - Jitter under high load.

    - Example:
      ```python

      import RPi.GPIO as GPIO
      GPIO.setmode(GPIO.BCM)
      GPIO.setup(19, GPIO.OUT)
      pwm = GPIO.PWM(19, 1000)  # 1000Hz
      pwm.start(50)  # 50% duty

      ```

2. **Multiplexing (Time-Division PWM)**
   Method: Rapidly switch between motors.

    - Pros: Saves pins.
    - Cons: Complex code, reduced responsiveness.

3. **Use Additional Microcontrollers (Arduino via Serial/UART)**

    - Pros:
        - Offloads PWM generation.
        - More reliable than software PWM.
    - Cons:
        - Extra hardware needed.

4. PCA9685 PWM/Servo Driver

The PCA9685 is a 16-channel controller that can control 16 PWM outputs via I2C communication, allowing you to drive up to 16 servos over I2C with only 2 pins.

- **Key Features**
  - **16 independent PWM channels** with 12-bit resolution
  - **I2C interface** requiring only 2 GPIO pins (SDA, SCL)
  - Supports PWM frequencies from about 24 Hz up to 1526 Hz, with servos typically running at 50-60 Hz
  - **Chainable design**: You can chain up 62 breakouts to control up to 992 PWM outputs
  - External power supply support (5V-10V for servos)
  - Built-in oscillator eliminates timing dependencies on the Pi
- **Implementation**

  ```python
  import board
  import busio
  import adafruit_pca9685
  from adafruit_pca9685 import PCA9685
  i2c = busio.I2C(board.SCL, board.SDA)
  pca = PCA9685(i2c)
  pca.frequency = 50  # 50Hz for servos

  pca.channels[0].duty_cycle = 0x7FFF  # 50% duty cycle

  ```

- **Pros**
  - Hardware-based PWM generation
  - Frees up Pi's GPIO pins for other functions
  - Consistent timing regardless of Pi's CPU load
  - Cost-effective for multiple PWM channels
- **Cons**
  - Requires additional hardware component
  - I2C communication adds slight latency
  - Voltage limitations (typically max 6V for servo power)