SERVO CHANNEL MAPPING

```
DEFINE dictionary "servos":
   leg 1 → hip:0, knee:1       #channels #
   leg 2 → hip:2, knee:3
   leg 3 → hip:4, knee:5
   leg 4 → hip:6, knee:7
   leg 5 → hip:8, knee:9
   leg 6 → hip:10, knee:11
```

BODY GROUPING

```
DEFINE LEFT_SIDE_LEGS  =[4, 5, 6]        to rotate left # left leg back, right leg push forward
DEFINE RIGHT_SIDE_LEGS =  [1, 2, 3]

DEFINE TRIPOD GROUP A = [1, 5, 3]
DEFINE TRIPOD GROUP B = [2, 6, 4]
```

SERVO ANGLE CONSTANTS

```
DEFINE HIP_CENTER = 90
DEFINE HIP_FORWARD = 130        // Swing leg forward
DEFINE HIP_BACKWARD = 50       // Pull leg backward

DEFINE KNEE_UP = 40              // Lift leg
DEFINE KNEE_DOWN = 90          // Plant leg
```

ANGLE → PWM FUNCTION

```
FUNCTION set_angle(channel, angle):    #12 bits duty cycle
   CLAMP angle between 0 and 180

   CONVERT angle to duty_cycle range:
      0° maps to duty = 102                #duty = (pulse_width_ms / 20 ms) * 4096   0.5ms
      180° maps to duty = 512                                                          2.5ms

   SEND duty_cycle to PCA9685(channel)
END FUNCTION
```

BASIC LEG CONTROL

```
def set_leg_pos(leg_number, hip_angle, knee_angle):
```

```
    1. Get servo channels from mapping
    hip_channel  = servos[leg_number]["hip"]
    knee_channel = servos[leg_number]["knee"]

    2. Mirror right-side legs
    if leg_number in RIGHT_SIDE_LEGS:
        hip_angle  = 180 - hip_angle      # fix hip direction
        knee_angle = 180 - knee_angle     # fix knee direction

    3. Send to PCA9685
    set_angle(hip_channel, hip_angle)
    set_angle(knee_channel, knee_angle)
```

DETERMINE HIP TARGET BASED ON DIRECTION

Turning LEFT = left legs go backward + right legs go forward

```
FUNCTION get_hip_angle(leg_number, direction):

    IF direction == "forward":
        RETURN HIP_FORWARD

    IF direction == "backward":
        RETURN HIP_BACKWARD

    IF direction == "left":
        IF leg_number is in LEFT_SIDE_LEGS:
            RETURN HIP_BACKWARD
        ELSE:
            RETURN HIP_FORWARD

    IF direction == "right":
        IF leg_number is in RIGHT_SIDE_LEGS:
            RETURN HIP_BACKWARD
        ELSE:
            RETURN HIP_FORWARD

    RETURN HIP_CENTER
END FUNCTION
```

CORE TRIPOD STEP

```
FUNCTION step_logic(lift_group, push_group, direction):

    // 1. LIFT air legs
    FOR each leg in lift_group:
        MOVE knee to KNEE_UP
    WAIT small delay

    // 2. MOVE both groups simultaneously
    FOR each leg in lift_group:
        hip_target = get_hip_angle(leg, direction)
        MOVE hip servo to hip_target

    FOR each leg in push_group:
        IF direction == "forward":
            MOVE hip servo to HIP_BACKWARD   (push)
        ELSE IF direction == "backward":
            MOVE hip servo to HIP_FORWARD    (pull)
        ELSE IF direction == "left":
            LEFT legs → HIP_FORWARD
            RIGHT legs → HIP_BACKWARD
        ELSE IF direction == "right":
            RIGHT legs → HIP_FORWARD
            LEFT legs → HIP_BACKWARD

    WAIT movement delay

    // 3. DROP air legs back
    FOR each leg in lift_group:
        MOVE knee to KNEE_DOWN
    WAIT small delay

END FUNCTION
```

STAND

```
FUNCTION stand():
    FOR each leg from 1 to 6:
        MOVE hip to HIP_CENTER
        MOVE knee to KNEE_DOWN
    WAIT half second
```

END FUNCTION

WALK (Forward / Backward / Left / Right)

```
FUNCTION walk(direction, steps):
   FOR count in 1 to steps:
      CALL step_logic(GROUP A, GROUP B, direction)
      CALL step_logic(GROUP B, GROUP A, direction)
   CALL stand()
END FUNCTION
```

CALIBRATION MODE

```
FUNCTION calibrate_servos():
   FOR each leg:
      MOVE hip to 90
      MOVE knee to 90
   PRINT "Attach legs straight now"
END FUNCTION
```

LOOP { main}

```
CALL stand()

WAIT 1 second

CALL walk("forward", 5 steps)
WAIT 1 second

CALL walk("backward", 5 steps)
WAIT 1 second

CALL walk("left", 5 steps)    // rotate left
WAIT 1 second

CALL walk("right", 5 steps)   // rotate right
WAIT 1 second

CALL stand()
```