

```

ZERO_MOVE = 75
ZERO_AN = 127
BUFFER_REG = 3
PAIR_UNCONNECTED = 7.5
PAIR_CONNECTED = 12.5
MAX_AN = 255

MyPriority      // Service priority
DC              // Duty cycle (used internally)
joystickX, joystickY
bytecounter     // External variable
speed = 0
dumping = false

Function: InitMotorService(Priority)
Set MyPriority = Priority

If PWM channels and timers are correctly configured:
    Print "Motors successfully configured."

Post an ES_INIT event to this service

If post is successful:
    Return true
Else:
    Return false

Function: PostMotorService(Event)
Post Event to this service's queue
Return whether posting was successful

Function: RunMotorService(Event)
Initialize ReturnEvent as ES_NO_EVENT

Switch Event.Type:

    CASE ES_INIT:
        Set PWM channels to neutral:
            Left and Right motors = ZERO_MOVE
            Pairing servo = PAIR_UNCONNECTED
            Dumping servo = resting position

    CASE ES_NEW_KEY:
        Switch Event.Param:
            'b': joystickX = 127, joystickY = 127

```

```

        'c': joystickX = ZERO_AN, joystickY = 193
        'd': joystickX = ZERO_AN, joystickY = 225
        'e': joystickX = ZERO_AN, joystickY = 230
        'f': joystickX = ZERO_AN, joystickY = 31
        'g': joystickX = ZERO_AN, joystickY = 63
        'h': joystickX = ZERO_AN, joystickY = 123
            → Post ES_SPEED event with above values

        'i': bytecounter = 9, Event.Param = 0b0
        'j': bytecounter = 9, Event.Param = 0b1
        'n': bytecounter = 12, Event.Param = 0x01
            → Post ES_RECEIVE event with given bytecounter and
param

        'k': Set dumping servo to resting
        'l': Set dumping servo to dumping position
        'm': Set pairing servo to PAIR_CONNECTED

CASE ES_RECEIVE:
    If bytecounter == 9 and Event.Param in [0x01, 0x02]:
        Print "Connected"
        Set pairing servo to PAIR_CONNECTED

    Else if bytecounter == 12 and Event.Param in [0x01, 0x03]:
        If dumping is false:
            dumping = true
            Set dumping servo to dumping position
            Print "Dumping..."
        Else:
            dumping = false
            Set dumping servo to resting position
            Print "Not dumping..."

CASE ES_SPEED:
    If joystick is centered (within BUFFER_REG of ZERO_AN):
        Stop both motors
        Print "Motors are not moving"

    Else if joystickX ≈ MAX_AN and joystickY ≈ ZERO_AN:
        Left = 60, Right = 90 → Turning CW
        Print "Turning CW"

    Else if joystickX ≈ 0 and joystickY ≈ ZERO_AN:
        Left = 90, Right = 60 → Turning CCW
        Print "Turning CCW"

```

```

Else if joystickX  $\approx$  0 and joystickY  $\approx$  0:
    Left = 77, Right = 90  $\rightarrow$  Reverse Left
    Print "Moving backwards left"

Else if joystickX  $\approx$  MAX_AN and joystickY  $\approx$  0:
    Left = 90, Right = 77  $\rightarrow$  Reverse Right
    Print "Moving backwards right"

Else if joystickX  $\approx$  0 and joystickY  $\approx$  MAX_AN:
    Left = 73, Right = 60  $\rightarrow$  Forward Left
    Print "Moving forwards left"

Else if joystickX  $\approx$  MAX_AN and joystickY  $\approx$  MAX_AN:
    Left = 60, Right = 73  $\rightarrow$  Forward Right
    Print "Moving forwards right"

Else if joystickX  $\approx$  ZERO_AN:
    If joystickY < 32: speed = 90
    Else if < 64: speed = 86
    Else if < 96: speed = 82
    Else if < 124: speed = 78
    Else if < 162: speed = 71
    Else if < 194: speed = 67
    Else if < 226: speed = 63
    Else: speed = 60

    Set both motors to `speed`
    Print "Moving forwards/backwards at speed"

DEFAULT:
    Do nothing

Return ReturnEvent

```