

# OMICRON ROBOT CONTROL PROGRAM

## Global Variable Initialization:

- Define motor speed setting variable.
  - Set initial value to 0.
- Define update interval counters.
  - Update rate for “heartbeat” LED.
  - Update rate for digital pots on motor speed controllers.
  - Update rate for behavioral maneuvers.

## End of Variable Initialization

## Setup():

- Define digital pin names.
- Define analog pin names (digital pot settings).
- Define pin functions (INPUT or OUTPUT).
- Set initial output pin states (HIGH or LOW).
- Reset bumper latches.
  - Check that raw switch inputs are open.
  - Cycle each latch reset output pin HIGH, LOW, HIGH.
- Ensure everything (motors) starts from “Off”.
  - “Spin” digital pots to zero.
- Initialize serial monitor.
- Await “Start” trigger command (front or rear sensor hit). Beep every ten secs.
  - Pause a second then start moving.
  - Move away from “obstacle” (forward if rear sensor, backward if front sensor).
- End Await Loop

## End of Setup

*NOTE: Use random function and read an open input to mix up the behavioral responses so they are not so predictable to give the robot the appearance of being alive.*

## Main Loop():

- Start wandering behavior:
  - Move forward, stop.
  - Randomize Left versus Right response (read open input as HIGH or LOW).
  - Randomly select (Spin/veer/pivot), L or R, veer forward L or R, stop.
  - Randomly select (Spin/veer/pivot) L or R, veer forward L or R, stop.
  - Repeat above behavior until a sensor changes it.
  - If bump sensors tripped:
    - Move away from obstacles it hits.
    - Change orientation/direction.
    - Move off in the new direction.
    - Reset bumper latches.
      - Check that each switch input is open then
        - Cycle each latch reset output pin HIGH, LOW, HIGH.
    - If any front + rear sensor together:
      - STOP. Beep every ten seconds. Await assistance.
    - End STOP loop

# OMICRON ROBOT CONTROL PROGRAM

- End bump sensors Loop
- If PINGs detect something:
  - Avoid hitting obstacles if possible.
    - Randomly chose to Veer, pivot, or spin L or R
    - Move straight forward
  - End PING loop
- Monitor PIRs when stopped and do something “interesting”.
  - Mounted on diagonal panels
  - Move toward PIR responses?
- Monitor IR feelers and avoid “cliffs”
  - Back away
- End Wandering Loop

**End of Main Loop - Loop back to Main Loop()**

## **Functions (called as needed):**

Pass (Maneuver# (1-13, LM spd, RM spd, maneuver exec time) to generic motor control function:

1. Sit and scan sensors
  - a. LM = OFF = RM
2. Move Straight, Forward
  - a. LM = RM == FWD
3. Veer Forward, Left
  - a. LM < RM == FWD
4. Pivot Forward, Left
  - a. LM = OFF, RM = FWD
5. Spin Counterclockwise (Turn to Left)
  - a. LM = REV == RM = FWD
6. Veer Forward, Right
  - a. LM, FWD > RM, FWD
7. Pivot Forward, Right
  - a. LM = FWD, RM = OFF
8. Spin Clockwise (Turn to Right)
  - a. LM = FWD == RM = REV
9. Move Straight, Backward
  - a. LM = RM == REV
10. Veer Backward, Left
  - a. LM = REV < RM = REV
11. Pivot Backward, Left
  - a. LM = OFF, RM = REV
12. Veer Backward, Right
  - a. LM = REV > RM = REV
13. Pivot Backward, Right
  - a. LM = REV, RM = OFF

**End Of File**