A Micromite (MKII) Robot Control Language

   IMHO, any robot operating system should, at a minimum, meet three criteria.

   1. Critical features (e.g., motor control) should run in the background, eliminating dependence on interrupts.

   2. The robot operating system should be programmable, i.e., parameters can be changed, new commands added, old commands deleted in situ.

   3. The robot command syntax should be simple, easily remembered, and universally applicable.

   The Micromite MMBasic meets these criteria. The fact that key ROS commands are embedded in the software (e.g., PWM, IR, Distance, Counters, and Servo), simplifying program structure, makes the Micromite a near perfect system for hobby robotics.

   My approach to programming, using MMBasic, is to mimic Forth by defining all components, even complex maneuvers, as subroutines. In effect, this adds commands - the robot control language - to the basic core and simplifies debugging. The commands described in this document apply to two-wheeled robot platforms using dc motors and H-bridge controllers as well as a servo-operated ping. Provision for IR-remote operation is also included

   The current version of the program includes 13 global variables, 38 subroutines, and occupies 6% of program memory.

The Kernel consists of all those commands which include elements specific to the robot platform. In this case, the Actobotics ActoBitty2 platform.

START is executed first. It sets direction pins as outputs, defines constants, explicitly defines & initializes variables, and centers the servo.

The RCL constants (units = inches) are:

 wd (wheel diam.) = 2   *wc* (wheel circum.) = 6.28
 rotd (rotational diam.) = 4.63 *rotc* = 14.5
 trnd (turn diam.) = 9.25   *trnc* = 29

Set Direction Pins Left & Right Wheels to Forward or
Backward - **LDF** or **LDB**   **RDF** or **RDB**

Set Wheel Velocities These 4 commands set the left (*lduty*)
and right (*rduty*) wheel duties (0-100%) which determine the
robot speeds. If necessary, using lduty as a reference,
rduty may be altered to match wheel speeds.
The parameter **cspd** stores the current lduty, *ipsec* (inches-
per-sec) a calibration factor relating distance to time
traveled, and *dadj* a correction factor for turns.

5 Volt Motor Power: **S1** (~10 rpm) or **S2** (~20 rpm)
9 Volt Motor Power: **S3** (~30 rpm) or **S4** (~45 rpm)

GO *lduty,rduty* - Enables the user to experiment with wheel
velocity combinations not covered by the **S3-S4** commands.

Basic Motion Commands Nonautomonous (drone) modes for robot
control.

  F - moves forward   **B** - moves backward

  RR - rotates right   **RL** - rotates left

  TR - turns right   **TL** - turns left

  S - stops motion

Defined Motion Commands These commands enable the robot to
move or turn a specified number of inches or degrees.

GF *x* - go forward   *or* **GB** *x* - go backward *x* inches

GRR *x* - rotate right or **GRL** *x* - rotate left *x* degrees

GTR *x* - turn right   or **GTL** *x* - turn left *x* degrees

Special Motion Commands

RU *dt* - Ramps speed up to maximum set by **Sx.**
        (*dt* = no. ms step width - determines rate)
RD *dt* - Ramps speed down from maximum set by **Sx.**

VR - Robot veers right 5 degrees.

VL - Robot veers left 5 degrees.

GSTEP *x,dt* - Emulates a step motor. Robot moves *x* number of steps with distance determined by *dt* ms.

zsno - Resets step number (*sno*) to 0.

SetStep *x* - Sets motion type *x*, 0 - 5, which will be stepped, e.g., 0 is **F**, 5 is **TL**, etc.

Accessories

PING(*x***)** - A function which activates the ultrasonic distance measurement. *x* is a dummy variable set to 0.

SERV x - Moves the servo *x* degrees: 1 to 90 turns ping to right, -1 to -90 turns ping to left or 0 centers the ping. Turns servo off when action is completed.

pars - A utility which displays the ping distance measurement, servo angle, duties, and current rpm.

IR Remote Operation

NOOP - A dummy command which does nothing.

IRGO - Relates IR remote keycodes to specific robot control commands.

IROPEN - Activates IR remote operation to control robot. Exit by pressing mute on the remote control.

```
Keys  1 - F   2 - B   3 - RR  4 - RL  5 - TR  6 - TL
      7 - RU  8 - RD  9 - VR  0 - VL  Power - S
      VolUp - S1  CHUp - S2  VolDn - S3  ChDn - S4
      Mute - Exit
```


Maneuvers - At this time, only one example has been programmed.

AVOID1 - If the robot is approaching close to an object, it stops, moves back, then looks to the right and left of the object. It moves in the direction of the largest

measured distance.

AUTO1 *dt* - Activates the AVOID1 manuever, taking a ping
reading every 200 ms. The maneuver times out at dt seconds.