

Using the Sequencer

As you may know, Lisp sends commands for motion and other controls to the microcontroller via a message that is passed through the Nav and Control program. Each command type (e.g. go forward, turn, etc.) is represented by a number. Values of less than 100 represent basic motions which are passed on to the microcontroller with little processing. Values of 100 or greater are for more complex commands (e.g. “go to the kitchen”, in an optimistic sense) which require the Nav and Control program to read sensors and command the microcontroller to accomplish the high level task.

The Sequencer provides a convenient means to create scripts defining such complex tasks. The Sequencer is a simple state machine for which the function of each state is defined in C++ code in the Nav and Control program; and the sequence of states is defined in a text file. This implementation allows tasks such as “go forward following a wall” to be defined once in the Nav and Control code, and for that task to be called by any task list that needs it.

Some examples of available tasks are:

- Go straight for a specified distance
- Turn for a specified angle change
- Stop forward and/or turn motion
- Wait for a specified time
- Terminate sequencer mode.

Hence, the robot could have a high level command to move in a square pattern with a list that says:

- Go straight for 1000 mm.
- Turn right 90 degrees
- Go straight for 1000 mm.
- Turn right 90 degrees
- Go straight for 1000 mm.
- Turn right 90 degrees
- Go straight for 1000 mm.
- Turn right 90 degrees
- Terminate sequencer mode

The command list is (unfortunately) written as a numerical list. Each line has 12 parameters, each of which is an integer number. (It might be nice to write a program to generate and edit these lists in a more user friendly manner someday)

For all lines, the first parameter is a label allowing the sequence to have different starting points and for branching to occur between states.

The second parameter defines the particular state to be executed and the remaining 10 parameters define the operation of that state.

The length of the list is primarily determined by the most complex command that is to be executed. This is a command to specify the robot start a forward motion and a turn motion simultaneously. This command is essentially the same format as the primitive command sent by the Lisp software to start a motion. An example follows:

```
100  0    1    200  400  2000  0    0    0    0    0    0
```

The first parameter (100) is the label number.

The second (0) is the mode number commanding the robot to go straight forward.

The next four parameters command the forward motion mode, an acceleration of 200 mm/sec², a top speed of 400 mm/sec and a distance of 2000 mm before stopping.

The next four parameters are provisions for a similar turn command with turn mode number, acceleration, speed and distance. This is for a future mode which does forward motion and turn simultaneously. The next to last parameter (0) is to specify options and the final parameter is for a label number which is for an optional label number which the state may choose to branch to depending on logic performed in the state.

Hence, the example of the square pattern above would be:

```

100  0    1    200  400  1000  0    0    0    0    0    0
0    1    0    0    0    0    1    100  500  900  0    0
0    0    1    200  400  1000  0    0    0    0    0    0
0    1    0    0    0    0    1    100  500  900  0    0
0    0    1    200  400  1000  0    0    0    0    0    0
0    1    0    0    0    0    1    100  500  900  0    0
0    0    1    200  400  1000  0    0    0    0    0    0
0    1    0    0    0    0    1    100  500  900  0    0
0    8    0    0    0    0    0    0    0    0    0    0

```

Where the final line, “0 8.....” is the exit sequence mode command.

General notes:

The label number must always be supplied. It is generally specified as zero if not used. It can vary from 0 to 2000000000.

Label numbers (other than zero) should not be reused. The label is searched for starting from the top of the list. Hence, a second use will not be found.

The second parameter is the number of the command mode. This number is used in a case statement in the Nav and Control code to execute the proper command. (need code to handle the error if a non-existent command mode is specified)

The remaining parameters are variable and depend on the command mode.

Any line starting with a ‘;’ is a comment line. ALL lines that are not command lines must start with a ‘;’

Specific modes:

Notes: parameter numbers below start with the label called zero.

The term “deg*10” means the measurement is in tenths of a degree. E.g. 900 is 90 degrees. Rate and accel numbers are always positive unless noted otherwise.

0: Go straight for a specified distance

param 2 should be either ????

param 3 is the acceleration/deceleration in mm./sec^2

param 4 is the maximum speed to be used in mm/sec.

Param 5 is the distance to travel in mm. (negative values mean reverse)

The robot will decelerate to a stop at the specified distance and exit the mode.

The remaining parameters are unused.

Example:

```

0    0    1    200  400  -1000  0    0    0    0    0    0

```

says to backup 1000 mm at a max speed of 400 mm/sec with accel/dec of 200 mm/sec^2

- 1: Turn to a specified angle**
 params 2 thru 5 are unused
 param 6 should be either ???
 param 7 is the acceleration/deceleration in deg*10/sec².
 Param 8 is the maximum turn rate in deg*10/sec.
 Param 9 is the distance to turn in deg*10. Positive numbers mean turn right, negative is left.
 The robot will decelerate to a stop at the specified angle and exit the mode
 The remaining parameters are unused.

Example:

0 1 0 0 0 0 1 600 900 -450 0 0
 says to turn left 45 degrees at a max rate of 90 deg/sec with accel/decel of 60 deg/sec²

- 2: Go straight forever**
 parameters are the same as case 0 above.
 The difference is that the mode exits immediately rather than waiting for command completion.
 This allows other commands to be executed during the forward motion.
 Param 5, distance, will be a very large value, e.g. +/- 999999 to go “forever”, however, any desired distance can be specified.

Example:

0 2 1 200 400 999999 0 0 0 0 0 0
 says to go forward “forever” at a max speed of 400 mm/sec with accel/decel of 200 mm/sec² and to exit this mode and go to the next sequential step immediately.

- 3: Turn forever**
 parameters are the same as case 1 above.
 The difference is that the mode exits immediately rather than waiting for command completion.
 This allows other commands to be executed during the turn.
 Param 9, distance, will be a very large value, e.g. +/- 999999, to turn “forever”, however, any desired distance can be specified.

Example:

0 3 0 0 0 0 1 600 900 999999 0 0
 says to turn right “forever” at a max rate of 90 deg/sec with accel/decel of 60 deg/sec² and to exit this mode and go to the next sequential step immediately.

- 4: Stop forward motion**
 param 2 should be either ??? specifies type of stop
 param 3 is the deceleration in mm./sec²
 The remaining parameters are unused.

Example:

0 4 ? 200 0 0 0 0 0 0 0 0
 says to stop forward (or reverse) motion at a deceleration of 200 mm/sec² and to exit when the stop is completed

- 5: Stop turn motion**
 param 6 should be either ??? specifies type of stop
 param 7 is the deceleration in mm./sec²
 The other parameters are unused.

Example:

0 5 0 0 0 0 ? 900 0 0 0 0
 says to stop turning with deceleration of 90 deg/sec² and to exit when the stop is completed

6: Stop forward and turn motion
param 2 should be either ??? specifies type of stop
param 3 is the deceleration in mm./sec²
param 6 should be either ??? specifies type of stop
param 7 is the deceleration in mm./sec²
The other parameters are unused.

Example:

0 6 ? 200 0 0 ? 900 0 0 0 0
says to forward motion with a deceleration of 200 mm/sec² and to stop turning with deceleration of 90 deg/sec² and to exit when the stop is completed.

7: Wait for a specified time in milliseconds
param 2 is the time to wait in milliseconds
The other parameters are not used.

Example:

0 7 2000 0 0 0 0 0 0 0 0 0
says to wait for two seconds before continuing to the next step.

8: Terminate sequence mode
All other parameters are unused. Sequence mode will exit and report completion to Lisp.

Example:

0 8 0 0 0 0 0 0 0 0 0 0
says to exit the sequence and wait for the next command from Lisp

50: Set an RC servo to a specified angle
param 2 specifies the RC servo output number
Param 3 is the maximum turn rate in deg*10/sec.
Param 4 is the angle to turn in deg*10. Positive numbers mean right of center , negative is left.
The remaining parameters are not used.

Example:

0 50 2 600 900 0 0 0 0 0 0 0
says to command servo number 2 to turn to the 90 degree clockwise position at 60 deg/sec.

51: Set an RC servo to scan between two angles
param 2 specifies the RC servo output number
Param 3 is the maximum turn rate in deg*10/sec.
Param 4 is the most clockwise angle to turn to in deg*10. Positive numbers mean right of center , negative is left.
Param 5 is the most CCW angle to turn to in deg*10.
The remaining parameters are not used.
This scanning will continue until another RC command is received (e.g. mode 50).

Example:

0 50 2 600 450 -450 0 0 0 0 0 0
says to command servo number 2 to scan back and forth between 45 degrees left and right of center at 60 deg/sec.