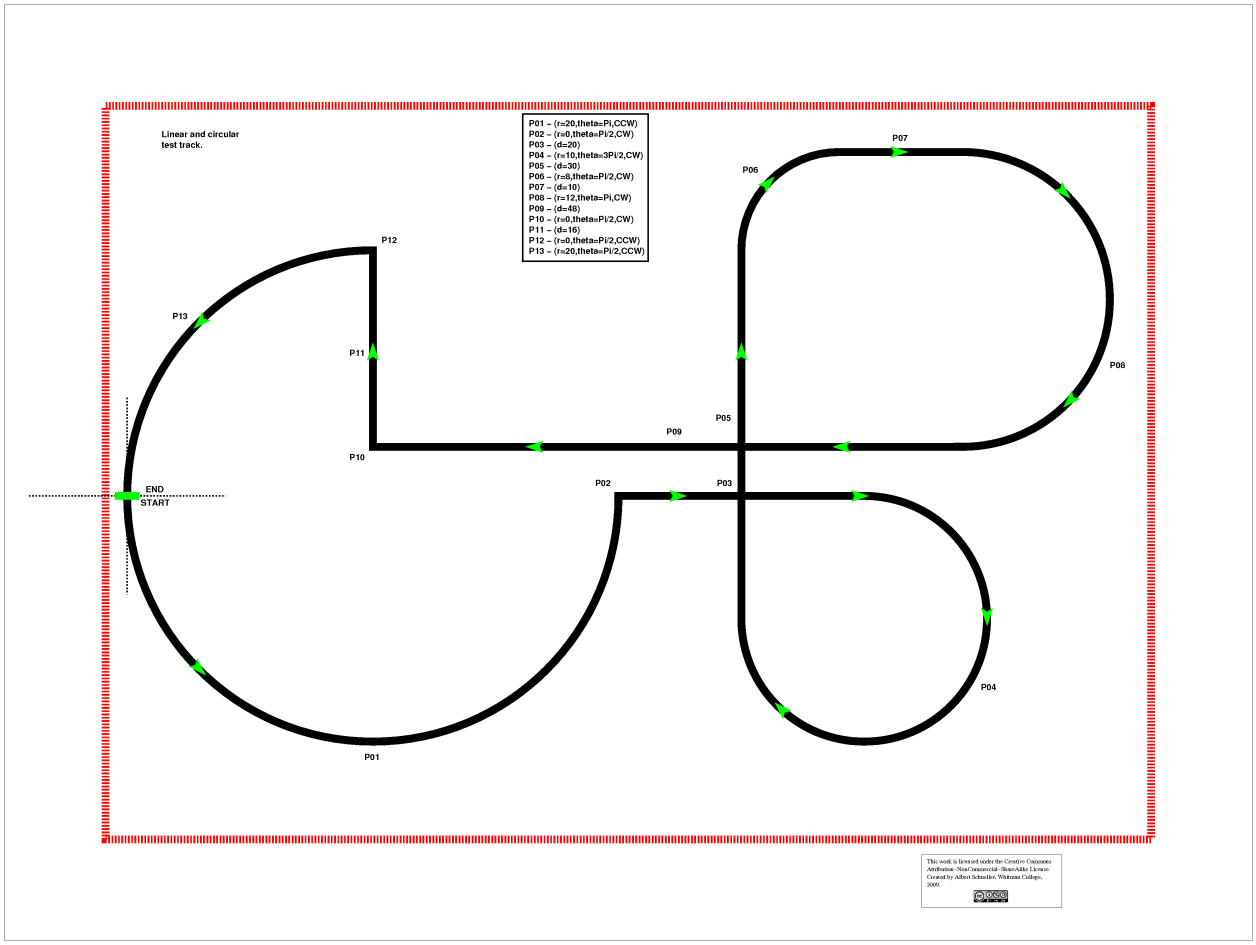
Ma204–Programming with Robots: **Homework 05**
Distributed: February 25, 2009 Due: March 6, 2009

Submit the program files to me via the upload web page[1] *described in class. Please do* **not** *print out your program, only submit it electronically.*

(1) **Programming Exercise:** Chapter 7, Exercise 2. I expect a careful write up of this problem in addition to the program. Also, the results obtained in this exercise will help with the next exercise.
(2) **Programming Exercise:** Use the information from Chapter 7 to write a program to instruct the robot to traverse the path below



The path segments are defined in centimeters and in sequential order as follows:

- P01 circular, $r = 20$, $\theta = \pi$, CCW
- P02 circular, $r = 0$, $\theta = \pi/2$, CW
- P03 linear, $d = 20$
- P04 circular, $r = 10$, $\theta = 3\pi/2$, CW
- P05 linear, $d = 30$

[1] http://carrot.whitman.edu/204/upload.html

- P06 circular, $r = 8$, $\theta = \pi/2$, CW
- P07 linear, $d = 10$
- P08 circular, $r = 12$, $\theta = \pi$, CW
- P09 linear, $d = 48$
- P10 circular, $r = 0$, $\theta = \pi/2$, CW
- P11 linear, $d = 16$
- P12 circular, $r = 0$, $\theta = \pi/2$, CCW
- P13 circular, $r = 20$, $\theta = \pi/2$, CCW

Each movement should be executed through the use of functions you write called `CircularMove()` and `LinearMove()`. Their declarations should resemble:

```
void CircularMove(float radius, float theta,
          string direction, int speed, float w,
          float ec) {
 // you fill this in
}
```

```
void LinearMove(float dist, int speed,
          float ec) {
 // you fill this in
}
```

In the first function, the arguments are self-explanatory except the `string direction`. This argument should be set to `"cw"` if the movement is clockwise and `"ccw"` if counter-clockwise. And the last two, `float w` is the trackwidth of the robot, and `float ec` is the effective circumference. The arguments to the second function are also self-explanatory. Units are centimeters and angles are in radians. (Note that RobotC recognizes the reserved word `PI` as the constant 3.1415....)

You may practice on the posterboard path available in the MathLab. Your work will be evaluated both on performance and on whether your program source code meets the above requirements. The two main performance rules are: the robot must keep the path between the drive wheels at all times (except during pivot turns); the robot must start and finish with its drive wheels inside the start/end box (as defined by the dashed alignment lines). We will test the robots in class on Friday March 6th.