Robots I - Week 7

Turbo Motors

Read the Docs

You can learn a lot about what the RasPi Robot supports by reading the project page:

https://github.com/simonmonk/raspirobotboard3

Or even the source code to library:

https://github.com/simonmonk/raspirobotboard3/blob/master/python/rrb3.py

Turbo Motors

When reading through the documentation, one of the things you'll notice is that we've been running our motors at half power. In most weeks, we've been using the "high level" interface, using code such as:

```
rr.forward()
```

This code will run at run both motors at half speed indefinitely. The RasPi Robot also has a "low level" interface that is a little harder to use, but provides greater "fine grained" motor control capability for our robot. Today we will look at the "set_motors" function.

```
rr.set_motors(LEFT_SPEED, LEFT_DIRECTION, RIGHT_SPEED, RIGHT_DIRECTION)
```

So, in code, `rr.forward()` is the same as `rr.set_motors(0.5, 0, 0.5, 0)`. This means to run out motors in full speed, we will have the command `rr.set_motors(1, 0, 1, 0)`.

Using the same code from previous weeks, we are going to introduce speed keys and refactor our main loop to take advantage of the fine grain API.

First, we are going to introduce two new constants

```
SLOWER = 46
FASTER = 44
```

These are set to the key codes for the “>” (FASTER) and “<” (SLOWER) keys. Prior to our main loop, we will have three variables that keep track of the state of the left and right motors as well as the fine grain speed.

```
left_direction = 0
speed = 0.5
right_direction = 0
```

By default, our robot will operate at half speed going in the forward direction (forward is 0). We change around our while true loop to capture how we want these variables to be set. Then we make a single “set_motors” function call. We will again use our rangefinder to automatically halt the robot if it is too close to an obstacle, however, feel free to turn this off it gets in the way of progress. Here is our finished code:

```
import os
import sys
from time import *
from rrb3 import *
import threading

FORWARD = 56
LEFT = 54
RIGHT = 52
REVERSE = 50
STOP = 53
SLOWER = 46
FASTER = 44

os.system("stty raw -echo")

robot = RRB3(6,6)

def worker():
   while True:
      if robot.get_distance() < 10:
         robot.stop()
      time.sleep(0.2)

t = threading.Thread(target=worker)
t.start()

left_direction = 0
speed = 0.5
right_direction = 0

os.system("stty raw -echo")
```

```
while True:
   r = ord(sys.stdin.read(1))
   is_set = True
   print r
   if r == FORWARD:
      left_direction = 0
      right_direction = 0
   elif r == LEFT:
      left_direction = 0
   elif r == RIGHT:
      left_direction = 1
      right_direction = 0
   elif r == REVERSE:
      left_direction = 1
      right_direction = 1
   elif r == FASTER:
      speed = speed + 0.1
      if speed > 1.0:
         speed = 1.0
   elif r == SLOWER:
      speed = speed - 0.1
      if speed < 0.0
         speed = 0.0
   else:
      is_set = False

   if is_set:
      robot.set_motors(speed, left_direction, speed, right_direction)
   else:
      robot.stop()
```

Race Time

Once you have your code tested “on the blocks”, take it for a spin around the room. Once we get enough robots moving around it will be time for a race!