

```

1  from machine import Pin
2  from machine import DEC
3
4  class Encoder:
5
6      def __init__(self, chA, chB, unit, counts_per_turn=24*75, wheel_diameter=330):
7          '''Decode output from quadrature encoder connected to pins chA, chB.
8              unit: DEC unit to use (0 ... 7).
9              counts_per_turn: Number of counts per turn of the motor drive shaft. For scaling
10                 cps to rpm.
11                 wheel_diameter: In [mm]. For scaling count to distance traveled.
12                 '''
13             self.p1 = Pin(chA, mode=Pin.IN, ...)
14             self.p2 = Pin(chB, mode=Pin.IN, ...)
15             self.cpt = counts_per_turn
16             self.wd = wheel_diameter
17             self.dec = DEC(unit, p1, p2)
18             self.count = 0
19             self.time = 0
20             self.cps = 0
21
22     def get_count(self):
23         return dec.count()
24
25     def get_distance(self):
26         '''Distance traveled in [m].'''
27         return get_count() * 3.14 * self.wd / 1000
28
29     def get_cps(self):
30         if self.count == 0:
31             self.count = dec.count()
32             self.time = time.clock()
33             return 0
34         else:
35             count = dec.count()
36             curr_time = time.clock()
37             diff = count - self.count
38             timediff = self.time - curr_time
39             self.time = curr_time
40             self.count = dec.count()
41             self.cps = diff/timediff
42             return self.cps
43
44     def get_rpm(self):
45         return self.cps/self.cpt
46
47 # Example:
48 import time
49 encA = Encoder(34, 39, 0)

```