```python
from machine import Pin, DEC, PWM
from drv8833 import DRV8833
import time


class Encoder:

    def __init__(self, chA, chB, unit, counts_per_turn=24*75, wheel_diameter=330):
        '''Decode output from quadrature encoder connected to pins chA, chB.
        unit: DEC unit to use (0 ... 7).
        counts_per_turn: Number of counts per turn of the motor drive shaft. For scaling cps to rpm.
        wheel_diameter: In [mm]. For scaling count to distance traveled.
        '''
        self.p1 = Pin(chA, mode=Pin.IN)
        self.p2 = Pin(chB, mode=Pin.IN)
        self.cpt = counts_per_turn
        self.dia = wheel_diameter
        self.dec = DEC(unit, self.p1, self.p2)
        self.count = self.dec.count()
        self.time = time.time()
        self.cps = 0

    def get_count(self):
        return self.dec.count()

    def get_distance(self):
        return get_count() / self.cpt * 3.14 * self.dia / 1000

    def get_cps(self):
        count = self.dec.count()
        curr_time = time.time()
        diff = count - self.count
        timediff = curr_time - self.time
        self.time = curr_time
        self.count = self.dec.count()
        self.cps = diff/timediff
        return self.cps

    def get_rpm(self):
        return self.get_cps()/self.cpt * 60

    def clear_count(self):
        # modify to match the variable names used in your code:
        self.dec.clear()
        self.count = self.dec.count()
        self.time = time.time()
```