```
"""
Created on Fri Feb  9 23:53:47 2018

@author: JasonGrillo
"""
import motor
import controller
import micropython

class Motor_Task:
    ''' This defines the task function method for a motor. The motor
        utilizes shared data from an encoder to know where it is.
        To create an instance of this task class (example):
            # have run shared variable declared
            Run = task_share.Share('i', thread_protect = False,
                                   name = "Run_Intertask_Comm_Variable")
            # have encoder position shared variable declared
            enc_1_position = task_share.Share ('i', thread_protect = Fals
                                               name = "Share_0_enc_1_posi
            # create motor 1 task object
            Motor_1 = motor_task_func.Motor_Task(Run, enc_1_position, 4,
                                     pyb.Pin.board.PB6, pyb.Pin.board.PB7
            # create task2 function, adjust parameters for implementation
            task2 = cotask.Task (Motor_1.enc_fun(), name = 'Task_2', prio
                         period = 2, profile = True, trace = False)
            # append task2 to list of scheduled tasks
            cotask.task_list.append (task2)
    '''

    def __init__(self, position, coordinate, timer, EN_Pin, pin1, pin2, K
        ''' Construct an encoder task function by initilizing any shared
            variables and initialize the encoder object
            @param position The shared variable between tasks that contai
            @param coordinate The desired coordinate to which to move the
            @param timer The Motor's timer channel
            @param EN_pin The Motor's __?__ pin
            @param pin1 The Motor's first pin, Pin A
            @param pin2 The Motor's second pin, Pin B
            @param Kp The Motor Controller's proportional gain
            @param Ki The Motor Controller's integral gain
            @param Setpoint Where the motor is desired to go
            @param saturation The anti wind up saturation limit
        '''
        self.position = position
        self.coordinate = coordinate
        self.Motor = motor.MotorDriver(timer, EN_Pin, pin1, pin2)
```

```
        self.Controller = controller.Controller(Kp, Ki, Kd, saturation)

    def mot_fun(self):
        ''' Defines the task function method for a Motor object.
        '''
        STATE_0 = micropython.const (0)
        STATE_1 = micropython.const (1)

        self.state = STATE_0

        while True:
            ## STATE 0: STOPPED
            if self.state == STATE_0:
                # Stop motor
                self.Motor.set_duty_cycle(0)
                self.state = STATE_1

            ## STATE 1: CONTROLLING MOTOR
            elif self.state == STATE_1:
                if self.coordinate.any():
                    print('new coordinate to move to!')
                    self.Controller.clear_controller()
                    self.Controller.set_setpoint(self.coordinate.get())
                # Use controller object to get appropriate duty cycle for
                self.Duty_Cycle = self.Controller.repeatedly(self.positio
                # Set duty cycle to motor
                self.Motor.set_duty_cycle(self.Duty_Cycle)

            yield(self.state)
```