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

@author: JasonGrillo
"""

import encoder
import micropython

class Encoder_Task:
    ''' This defines the task function method for an encoder. The encoder
        passes its data via a shared variable with another task.
        To create an instance of this task class (example):
            # create encoder position shared variable
            pan_position = task_share.Share ('i', thread_protect = False,
                                             name = "Share_0_pan_positi
            # create encoder 1 task object
            pan_encoder = Encoder_Task(pan_position, 4,
                                       pyb.Pin.board.PB6, pyb.Pin.board.PB7
            # create task1 function
            task1 = cotask.Task (pan_encoder.enc_fun(), name = 'Task_1',
                         period = 2, profile = True, trace = False)
            # append task1 to list of scheduled tasks
            cotask.task_list.append (task1)
    '''

    def __init__(self, pan_position, timer, pin1, pin2):
        ''' Construct an encoder task function by initilizing any shared
            variables and initialize the encoder object
            @param pan_position The shared variable between tasks that co
            @param timer The Encoder's timer channel
            @param pin1 The Encoder's first pin, Pin A
            @param pin2 The Encoder's second pin, Pin B
        '''
        self.pan_position = pan_position
        self.Encoder = encoder.Encoder(timer, pin1, pin2)

    def enc_fun(self):
        ''' Defines the task function method for an Encoder object.
        '''

        STATE_0 = micropython.const(0)
        STATE_1 = micropython.const(1)

        self.state = STATE_0
```

```
while True:
    ## STATE 0: ZERO REFERENCE
    if self.state == STATE_0:
        self.Encoder.zero_encoder()
        self.state = STATE_1

    ## STATE 1: UPDATING
    elif self.state == STATE_1:
        # Read encoder and update the shared variable
        self.pan_position.put(self.Encoder.read_encoder())
    yield (self.state)
```