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

@author: JasonGrillo
"""

import BNO055
import pyb
import micropython

class IMU_Task:
    ''' This defines the task function method for an IMU. The IMU
        passes its data via a shared variable with another task.
        To create an instance of this task class (example):
            # create run shared variable
            Run = task_share.Share('i', thread_protect = False,
                                   name = "Run_Intertask_Comm_Variable")
            # create IMU position shared variable
            IMU_position = task_share.Share ('i', thread_protect = False,
                                               name = "IMU_position")
            # create IMU 1 task object
            IMU_1 = IMU_Task(Run, IMU_position, 4,
                                     pyb.Pin.board.PB6, pyb.Pin.board.PB7
            # create task1 function
            task1 = cotask.Task (IMU_1.IMU_fun, name = 'Task_1', priority
                         period = 10, profile = True, trace = False)
            # append task1 to list of scheduled tasks
            cotask.task_list.append (task1)
    '''

    def __init__(self, tilt_angle):
        ''' Construct an IMU task function by initilizing any shared
            variables and initialize the IMU object
            @param tilt_angle The shared variable between tasks that cont
        '''
        self.tilt_angle = tilt_angle
        self.imu = BNO055.bno055 (pyb.I2C (1, pyb.I2C.MASTER, baudrate =

    def IMU_fun(self):
        ''' Defines the task function method for an IMU object.
        '''

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

        self.state = STATE_0
        while True:
            ## STATE 0: Initialize State Machine
            if self.state == STATE_0:
                # Calibrate the IMU against the hardstop
                # ... must be against hardstop upon system boot
                self.imu.zero_Euler_vals()
                self.state = STATE_1

            ## STATE 1: Get IMU Values
            elif self.state == STATE_1:
                # Read IMU and update the shared variable with Euler pitc
                self.tilt_angle.put(self.imu.get_euler_roll())

            yield (self.state)
```