

```
"""
```

```
Created on Thu Jan 11 21:19:40 2018
```

```
@author: mecha10, JGrillo, TGoehring, TPeterson
```

```
"""
```

```
import pyb
```

```
class MotorDriver:
```

```
    ''' This class implements a motor driver for the  
    ME405 board.
```

```
    either MotorDriver(3, pyb.Pin.board.PA10, pyb.Pin.board.PB4, pyb.Pin.  
        MotorDriver(5, pyb.Pin.board.PC1, pyb.Pin.board.PA0, pyb.Pin.  
    ...
```

```
def __init__(self, timer, EN_Pin, Pin_1, Pin_2):
```

```
    ''' Creates a motor driver by initializing GPIO  
    pins and turning the motor off for safety. '''
```

```
    print ('Creating a motor driver')
```

```
    ## Set Pin PA10 to as open-drain output with pull up resistors
```

```
    self.EN_Pin=pyb.Pin(EN_Pin,pyb.Pin.OUT_OD, pull=pyb.Pin.PULL_UP)
```

```
    ## Set Pin PB4 as push-pull with the correct alternate function (
```

```
    self.Pin_1=pyb.Pin(Pin_1, pyb.Pin.AF_PP,af=2)
```

```
    ## Set Pin PB5 as push-pull with the correct alternate function (
```

```
    self.Pin_2=pyb.Pin(Pin_2, pyb.Pin.AF_PP,af=2)
```

```
    self.timer= pyb.Timer(timer, freq=20000)
```

```
    self.ch1 = self.timer.channel(1, pyb.Timer.PWM, pin=self.Pin_1) #
```

```
    self.ch2 = self.timer.channel(2, pyb.Timer.PWM, pin=self.Pin_2) #
```

```
    self.EN_Pin.low()
```

```
    # Set P
```

```
    self.Pin_1.low()
```

```
    self.Pin_2.low()
```

```
def set_duty_cycle (self, level):
```

```
    ''' This method sets the duty cycle to be sent  
    to the motor to the given level. Positive values  
    cause torque in one direction, negative values  
    in the opposite direction.
```

```
    @param level A signed integer holding the duty cycle of the volta  
    '''
```

```
    if (level >= 0):
```

```
        self.ch1.pulse_width_percent(0)
```

```
        self.ch2.pulse_width_percent(level)
```

```
    else:
```

```
        self.ch2.pulse_width_percent(0)
```

```
        self.ch1.pulse_width_percent(-level)
```

```
    self.EN_Pin.high()
```

