

```

import Diablo

class MotorController():

    #contains i2caddresses given to class constructor
    i2caddresses = []
    #motor speed limits set in class constructor
    speedLimit = 0
    reverseLimit = 0

    DIABLO1 = None
    DIABLO2 = None
    DIABLO3 = None

    def Init(self, diablo1address, speedlimit, rspeedlimit, diablo2address = None, diablo3address =
None):
        self.diablos = []
        self.speedLimit = speedlimit
        self.reverseLimit = rspeedlimit
        self.DIABLO1 = Diablo.Diablo()
        self.DIABLO1.i2cAddress = diablo1address
        self.DIABLO1.Init()
        self.i2caddresses.append(diablo1address)
        self.activestraightmotor = 0
        self.diablos.append(self.DIABLO1)

        #if given second i2caddress, attempt to init diablo at that address
        if diablo2address != None:
            self.DIABLO2 = Diablo.Diablo()
            self.DIABLO2.i2cAddress = diablo2address
            self.DIABLO2.Init()
            self.diablos.append(self.DIABLO2)

        if diablo3address != None:
            self.DIABLO3 = Diablo.Diablo()
            self.DIABLO3.i2cAddress = diablo3address
            self.DIABLO3.Init()
            self.diablos.append(self.DIABLO3)

    def GetListOfDiablos(self):
        return self.diablos

    def DriveLeftSide(self, speed):
        speed = speed*-1
        self.DIABLO1.SetMotor1(speed)
        self.DIABLO1.SetMotor2(speed)

    def DriveRightSide(self, speed):
        self.DIABLO3.SetMotor1(speed)
        self.DIABLO3.SetMotor2(speed)

```

```
def MoveStraightMotorForward(self, speed):
    if self.activestraightmotor == 0:
        self.DIABLO2.SetMotor1(speed)
    elif self.activestraightmotor == 1:
        self.DIABLO2.SetMotor2(speed)
```

```
def MoveStraightMotorBackward(self, speed):
    speed = speed * -1
    if self.activestraightmotor == 0:
        self.DIABLO2.SetMotor1(speed)
    elif self.activestraightmotor == 1:
        self.DIABLO2.SetMotor2(speed)
```

```
def ChangeStraightMotor(self):
    if self.activestraightmotor == 0:
        self.activestraightmotor = 1
```