```
import socket
import sys
import time
import RPi.GPIO as GPIO
from Adafruit_PWM_Servo_Driver import PWM

def setMotors(ls, rs, fs, bs, ld, rd, fd, bd, a, c):

    #ls = left speed
    #rs = right speed
    #fs = front speed
    #bs = back speed

    #ld = left direction
    #rd = right direction
    #fd = front direction
    #bd = back direction

    #a = amp limit
    #Used to reduce current below 25A when three motors are in use

    #c = constant for amp limit

    if a == 0:
       ls = int(ls * 1.15)
       rs = int(rs * 1.15)
       fs = int(fs * 1.15)
       bs = int(bs * 1.15)

    if a == 1:
       ls = int(ls * c)
       rs = int(rs * c)
       fs = int(fs * c)
       bs = int(bs * c)

    if ld == 0:
         pwm.setPWM(0, 0, 387)
    if ld == 1:
         pwm.setPWM(0, 0, ls + 387)
    if ld == 2:
        pwm.setPWM(0, 0, 387 - ls)

    if rd == 0:
         pwm.setPWM(1, 0, 387)
    if rd == 1:
        pwm.setPWM(1, 0, rs + 387)
    if rd == 2:
        pwm.setPWM(1, 0, 387 - rs)

    if fd == 0:
```

```
         pwm.setPWM(2, 0, 387)
    if fd == 1:
        pwm.setPWM(2, 0, fs + 387)
    if fd == 2:
        pwm.setPWM(2, 0, 387 - fs)

    if bd == 0:
         pwm.setPWM(3, 0, 387)
    if bd == 1:
        pwm.setPWM(3, 0, bs + 387)
    if bd == 2:
        pwm.setPWM(3, 0, 387 - bs)



GPIO.setwarnings(False)
pwm = PWM(0x40)

#1100us ~~> 272 Full Power Reverse
#1500us ~~> 387 Stopped
#1900us ~~> 502 Full Power Forward


pwm.setPWMFreq(int(60 * 0.9))

pwm.setPWM(0, 0, 387)
pwm.setPWM(1, 0, 387)
pwm.setPWM(2, 0, 387)
pwm.setPWM(3, 0, 387)

time.sleep(3)

print("ESC Initialized")


HOST = ''
PORT = 8886

frontDirection = 0
backDirection = 0
leftDirection = 0
rightDirection = 0
frontSpeed = 0
backSpeed = 0
leftSpeed = 0
rightSpeed = 0
clawState = 0
ampLimit = 0

msg = ''
```

```
try:
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    print("Socket Created")
except (socket.error, msg):
    print('Failed to create socket. Error Code : ' + str(msg[0]) +
'Message' + msg[1])
    sys.exit()


try:
    s.bind((HOST, PORT))
except (socket.error, msg):
    print('Bind Failed. Error Code : ' + str(msg[0]) + 'Message' +
msg[1])
    sys.exit()


print('Socket Bind Complete')


try:
    while 1:
        d = s.recvfrom(1024)
        data = d[0]
        addr = d[1]

        if not data:
            break

        reply = ('')
        s.sendto(reply, addr)
        print('Message[' + addr[0] + ':' + str(addr[1]) + '] - ' +
data.strip())

        #Client Data Processing

        msg = data.strip()

        leftSpeed = int(msg[0] + msg[1] + msg[2])
        leftDirection = int(msg[3])
        rightSpeed =  int(msg[4] + msg[5] + msg[6])
        rightDirection = int(msg[7])
        frontSpeed = int(msg[8] + msg[9] + msg[10])
        frontDirection = int(msg[11])
        backSpeed = int(msg[12] + msg[13] + msg[14])
        backDirection = int(msg[15])
        #clawState = int(msg[16]) #Claw Data, no claw implemented
yet!!
        ampLimit = int(msg[17])
```

```
        #T100 Motor Output

        setMotors(leftSpeed, rightSpeed, frontSpeed, backSpeed,
                  leftDirection, rightDirection, frontDirection,
backDirection,
                  ampLimit, 0.80)



except KeyboardInterrupt:
    print("User Cancelled")

finally:
    s.close()
    pwm.setPWM(0, 0, 0)
    pwm.setPWM(1, 0, 0)
    pwm.setPWM(2, 0, 0)
    pwm.setPWM(3, 0, 0)
    quit()
```