

```

import time
import XboxController
import socket
import sys

#timer values, helps the speed of the UDP data
start = 0
current = 0
timerlogic = True

#Xbox Controller Values

rTrigger = 0
lTrigger = 0
LB = 0
RB = 0
xValueL = 0
yValueL = 0
xValueR = 0
yValueR = 0
A = 0
B = 0

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

ID = 0
msg = ''
rightTurn = False #Gives priority to forward/reverse controls
ampLimit = 0
conv = 1.190476190476191

def speedToString(speed):
    if (speed == 100):
        return str(speed)
    if (speed < 100):
        if (speed > 10):
            return ('0' + str(speed))
        if (speed == 10):

```

```
        return ('010')
    if (speed < 10):
        return ('00' + str(speed))
    else:
        return ('000')
```

```
def controlCallback(xboxControlId, value):
    #time.sleep(0.01)

    global rTrigger
    global lTrigger
    global LB
    global RB
    global xValueL
    global yValueL
    global xValueR
    global yValueR
    global A
    global B
    global X

    global frontDirection
    global backDirection
    global leftDirection
    global rightDirection
    global frontSpeed
    global backSpeed
    global leftSpeed
    global rightSpeed
    global clawState
    global ampLimit

    global msg
    global ID
    global rightTurn
    global conv

    global timerlogic

    ID = xboxControlId

    if ((int(round((int(value * 1000)), -1)) / 10) == (int(value *
100))):
        timerlogic = True
```

```
#Xbox Input Storage
```

```
if (ID == 4):  
    if value > 0:  
        rTrigger = 100  
    else:  
        rTrigger = 0  
if (ID == 5):  
    if value > 0:  
        lTrigger = 100  
    else:  
        lTrigger = 0  
if (ID == 10):  
    LB = value * 100  
if (ID == 11):  
    RB = value * 100  
  
if (ID == 0):  
    xValueL = int(value * 100)  
if (ID == 1):  
    yValueL = int(value * 100)  
if (ID == 2):  
    xValueR = int(value * 100)  
if (ID == 3):  
    yValueR = int(value * 100)  
  
if (ID == 6):  
    A = value  
if (ID == 7):  
    B = value
```

```
#Left and Right Motor Logic
```

```
if ((xValueL < 15 and xValueL > -15) and (yValueL > 15)):  
    leftDirection = 1  
    rightDirection = 1  
    leftSpeed = int((yValueL - 16) * conv)  
    rightSpeed = int((yValueL - 16) * conv)  
    rightTurn = False  
  
if ((xValueL < 15 and xValueL > -15) and (yValueL < -15)):  
    leftDirection = 2  
    rightDirection = 2  
    leftSpeed = int((yValueL + 16) * conv) * -1  
    rightSpeed = int((yValueL + 16) * conv) * -1  
    rightTurn = False  
  
if ((yValueL < 15 and yValueL > -15) and (xValueL > 15)):
```

```

leftDirection = 1
rightDirection = 0
leftSpeed = int((xValueL - 16) * conv)
rightSpeed = 0
rightTurn = False

if ((yValueL < 15 and yValueL > -15) and (xValueL < -15)):
    leftDirection = 0
    rightDirection = 1
    leftSpeed = 0
    rightSpeed = int((xValueL + 16) * conv) * -1
    rightTurn = False

if (yValueL > 15 and xValueL > 15):
    leftDirection = 1
    rightDirecton = 1
    leftSpeed = 100
    rightSpeed = int((yValueL - 16) * conv)
    rightTurn = False

if (yValueL < -15 and xValueL > 15):
    leftDirection = 2
    rightDirecton = 2
    leftSpeed = 100
    rightSpeed = int((yValueL + 16) * conv) * -1
    rightTurn = False

if (yValueL < -15 and xValueL < -15):
    leftDirection = 2
    rightDirecton = 2
    leftSpeed = int((yValueL + 16) * conv) * -1
    rightSpeed = 100
    rightTurn = False

if (yValueL > 15 and xValueL < -15):
    leftDirection = 1
    rightDirecton = 1
    leftSpeed = int((yValueL - 16) * conv)
    rightSpeed = 100
    rightTurn = False

if ((yValueL > -15 and yValueL < 15) and (xValueL > -15 and
xValueL < 15)):
    leftDirection = 0
    rightDirection = 0
    leftSpeed = 0
    rightSpeed = 0

```

```

rightTurn = True

if((xValueR > 15) and (rightTurn == True)):
    leftDirection = 1
    rightDirection = 2
    leftSpeed = int((xValueR - 16) * conv)
    rightSpeed = int((xValueR - 16) * conv)

if((xValueR < -15) and (rightTurn == True)):
    leftDirection = 2
    rightDirection = 1
    leftSpeed = int((xValueR + 16) * conv) * -1
    rightSpeed = int((xValueR + 16) * conv) * -1

#Back Motor Logic

if (LB > 0 and lTrigger == 0):
    backDirection = 1
    backSpeed = LB
if (LB == 0 and lTrigger > 0):
    backDirection = 2
    backSpeed = lTrigger
if (LB == 0 and lTrigger == 0):
    backDirection = 0
    backSpeed = 0
if (LB > 0 and lTrigger > 0):
    backDirection = 0
    backSpeed = 0

#Front Motor Logic

if (RB > 0 and rTrigger == 0):
    frontDirection = 1
    frontSpeed = RB
if (RB == 0 and rTrigger > 0):
    frontDirection = 2
    frontSpeed = rTrigger
if (RB == 0 and rTrigger == 0):
    frontDirection = 0
    frontSpeed = 0
if (RB > 0 and rTrigger > 0):
    frontDirection = 0
    frontSpeed = 0

#Claw Logic

if (A == 1 and B == 0):

```

```

        clawState = 1
    if (A == 0 and B == 1):
        clawState = 2
    if (A == 1 and B == 1):
        clawState = 0
    if (A == 0 and B == 0):
        clawState = 0

#Amp Limiter Logic

ampLimit = 0

    if (((frontDirection + backDirection) > 0) and ((leftDirection +
rightDirection) > 0)):
        ampLimit = 1

#Data Compiler

msg = (speedToString(leftSpeed) + str(leftDirection) +
        speedToString(rightSpeed) + str(rightDirection) +
        speedToString(frontSpeed) + str(frontDirection) +
        speedToString(backSpeed) + str(backDirection) +
        str(clawState) + str(ampLimit))

if (timerlogic == True):
    try:
        s.sendto(msg, (host, port))
        print(msg)
        d = s.recvfrom(1024)
        reply = d[0]
        addr = d[1]

        print('Server reply: ' + reply)
        timerlogic = False

    except (socket.error, msg):
        print('Error Code: ' + str(msg[0]) + 'Message ' + msg[1])
        sys.exit()

xboxCont = XboxController.XboxController(
    controlCallback,
    joystickNo = 0,
    deadzone = 0.1,

```

```
    scale = 1,
    invertYAxis = True)

try:
    #Setup UDP Client

    host = '192.168.0.7'
    port = 8886

    try:
        s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        print('Connected')
    except socket.error:
        print('Failed to create socket')
        sys.exit()

    #Startup Xbox Controller
    xboxCont.start()
    print("Xbox Controller Running")

    #Main loop
    while True:
        time.sleep(0.05)

except KeyboardInterrupt:
    print(" User Cancelled")

finally:
    xboxCont.stop()
    quit()
```