```
import bge, GameLogic, mathutils, socket
from math import radians

 def main():

        cont = bge.logic.getCurrentController()
        own = cont.owner
        Mat_orig = own.localOrientation

if not 'init' in own:

    print('Encoder Multiplex Init')

        own['init'] = 1
        own['UDP_IP'] = " "
        own['UDP_PORT'] = 7505
        own['Timeout'] = 0.01
        own['UDPdata'] = 0
        own['UDPPreamble'] = 0

        GameLogic.globalDict["SerialData"] = ('0 0 0')
        GameLogic.globalDict['sock'] = socket.socket( socket.AF_INET, socket.SOCK_DGRAM )
        GameLogic.globalDict['sock'].bind((own['UDP_IP'],own['UDP_PORT']))
        GameLogic.globalDict['sock'].settimeout(0.01)
        GameLogic.globalDict["EncoderReceiverState"] = 1

 GameLogic.globalDict["SerialData"], addr = GameLogic.globalDict['sock'].recvfrom(1024)

   if int(GameLogic.globalDict["EncoderReceiverState"]) == 1:

pitch = float(str(GameLogic.globalDict["SerialData"].decode()).split()[0])
roll = float(str(GameLogic.globalDict["SerialData"].decode()).split()[1])

    if float(pitch or roll) < 11:

        m_cor = mathutils.Matrix.Rotation(radians(90.0), 3,'X')
        m_rotY = mathutils.Matrix.Rotation(radians(pitch), 3,'Y')
        m_rotX = mathutils.Matrix.Rotation(radians(roll), 3,'X')
        m_rot = m_rotY*m_rotX
        own.orientation = m_rot*m_cor

  else:

    print('TopArmAxisEncoder 0 FeedBack Down')

main()
```