

```
import socket, sys, signal, queue, pickle

SERVER_IP = socket.gethostbyname(socket.gethostname())
print ("IP " + SERVER_IP)
PORT = 8
CHUNK = 512 #1024

# Mic address
mic1Addr = -1
mic2Addr = -1
micID = 0

#To exit the program on a keyboard interrupt(Eventually will use this to shut down system)
def signal_handler(signal, frame):
    s.close()
    sys.exit(0)

signal.signal(signal.SIGINT, signal_handler)

def addAudioDevice(addr):
    global micID
    global mic1Addr
    global mic2Addr

    if mic1Addr == -1:
        mic1Addr = addr
        micID = 1
    elif mic2Addr == -1:
        mic2Addr = addr
        micID = 2
    elif mic1Addr == addr:
        micID = 1
    else:
        micID = 2

    #send mic id to client
    s.sendto(str.encode(str(micID)), addr)

#Make the socket
try:
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    print("socket created")
except socket.error:
    print("Could not make socket. Error: ")
    sys.exit()
#Bind to address
try:
```

```
s.bind((SERVER_IP, PORT))
print("finished socket bind")
except socket.error:
    print("Bind failed Error: ")
    sys.exit()

audioOutMic1 = queue.Queue()
audioOutMic2 = queue.Queue()

while 1:
    #Getting new data
    try:
        dataByte, addr = s.recvfrom(CHUNK)
        data = "!"
    except BlockingIOError:
        data = ""

    #decode data
    if data == "!":
        try:
            data = dataByte.decode()
        except UnicodeDecodeError:
            data = pickle.loads(dataByte)

    print ("Message: ")
    print(data)

    #Handle input data
    if(data == "Hello Server"):
        addAudioDevice(addr)
    else:
        print("receiving data:")
        print(data)
        if data[2] == "1":
            audioOutMic2.put_nowait(data)
        else:
            audioOutMic1.put_nowait(data)

    #try to send data
    if not audioOutMic1.empty():
        try:
            d = audioOutMic1.get_nowait()
            s.sendto(pickle.dumps(d), mic1Addr);
        except:
            print("Could not sent to mic1")
```

```
if not audioOutMic2.empty():
    try:
        d = audioOutMic2.get_nowait()
        s.sendto(pickle.dumps(d), mic2Addr);
    except:
        print("Could not sent to mic2")
```