

ENGR 6928: Computer Aided Engineering

Tutorial- Setting up MatLab to achieve LAN/WLAN data communication

1. Establishing communication between two Matlab applications running on the same machine:

Network communication setup for Matlab programs running on the same machine using TCP protocol.

Machine1	Local IP	Irrelevant	Machine2	Local IP	Irrelevant
Application X	Local Port	9000	Application Y	Local Port	Any
	Remote IP	0.0.0.0		Remote IP	127.0.0.1
	Remote Port	Any		Remote Port	9000
	Protocol	TCP /Server		Protocol	TCP/Client

Run 2 matlab applications on the same machine. The following code would configure one matlab application as a server and the other as a client.

Server Application script	Client Application script
<pre>clear clc close all t = tcpip('0.0.0.0',4012,'NetworkRole', 'server'); fopen(t) while(1) if (t.BytesAvailable>0) data = fscanf(t) end end fclose(t)</pre>	<pre>clear clc instrreset % reset any connections of tis app if open close all t = tcpip('127.0.0.1',4012,'NetworkRole', 'client'); fopen(t) pause(1) fprintf(t,'testing') fclose(t)</pre>

Network communication setup for Matlab programs running on the same machine using the UDP protocol.

Machine1	Local IP	192.168.1.1	Machine2	Local IP	192.168.1.8
Application X	Local Port	9000	Application Y	Local Port	12345
	Remote IP	127.0.0.1		Remote IP	127.0.0.1
	Remote Port	12345		Remote Port	9000
	Protocol	UDP		Protocol	UDP

2. Using the DSP system toolbox for UDP communication:

The above script is only possible with the Matlab instrument control toolbox which is currently not available @ MUN labs. If you are to implement this in the computers at the lab., Your only option is to use the DSP systems toolbox.

receiver Application script	Sender Application script
<pre>clear clc close all hudpr = dsp.UDPReceiver('RemoteIPAddress','0.0.0 .0','LocalIPPort',25000,'MessageDataType ','uint8') string=[]; count=1; eof=0; while(~eof) dataReceived = step(hudpr); if strcmp(char(dataReceived),'!') eof=1; %! signified the end else string=[string char(dataReceived)]; %fill the string count=count+1; end end string %display the string release(hudpr)</pre>	<pre>clear clc close all hudps = dsp.UDPSEnder('LocalIPPortSource', 'Auto','RemoteIPAddress','127.0.0.1 ','RemoteIPPort',25000) data = uint8('testing...!'); %convert the string to uint8 for i=1:size(data,2) step(hudps,data(i)) %send the numbers 1 by 1 end release(hudps)</pre>

Notice that here the string is first converted to unsigned 8 bit integers, at the sender. At the receiver the data is then converted back to characters. This is because the 'MessageDataType' argument of the UDPReceiver function does not allow char as a data type.

3. Setting up a personal access point:

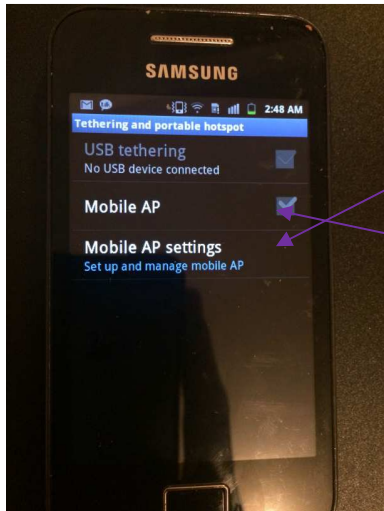
For communicating between two separate machines connected to a LAN, the windows firewall settings must be turned off. So it is not recommended to perform the connections that follows, while connected to the internet.

The recommended approach is to connect to a personal access point which is not connected to the internet, which would allow to perform the connections without unnecessary security risks.

This capability is there on some laptops. And is there on Android devices. iOS devices tend to connect to the internet through a data network prior to setting up a hotspot, so this would still create security

issues. Following are some screenshots which guide you through setting up a access point on an Android device.

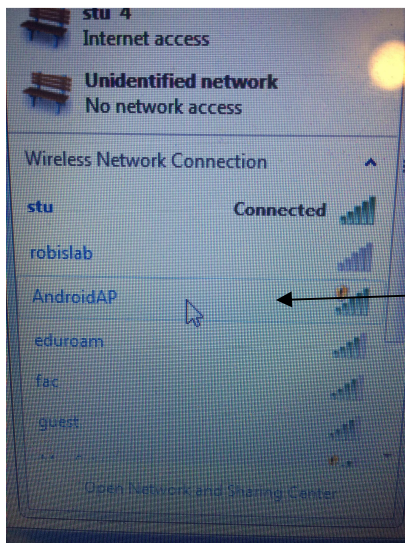
a) go to settings> Wireless & Networks > Tethering & portable hotspot. And turn on mobile AP.



Set the access point name and pass key.

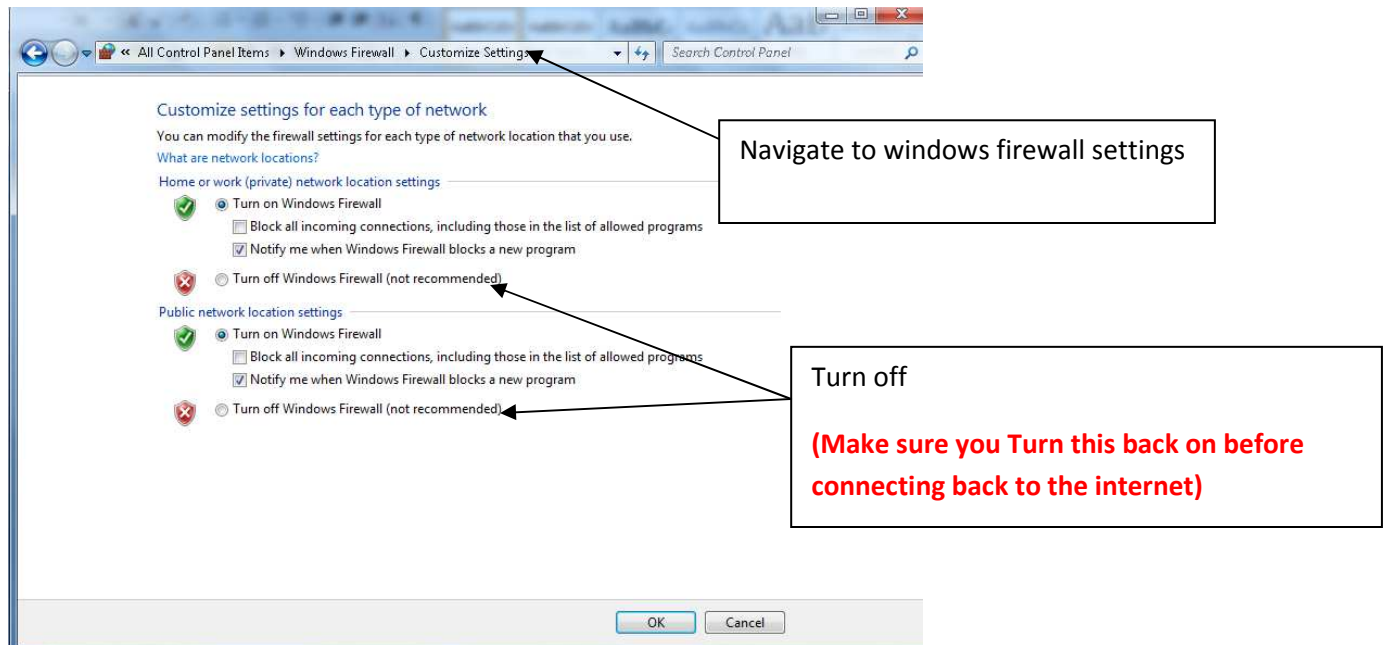
Enable Access point

b) Each machine can connect to this mobile access point.

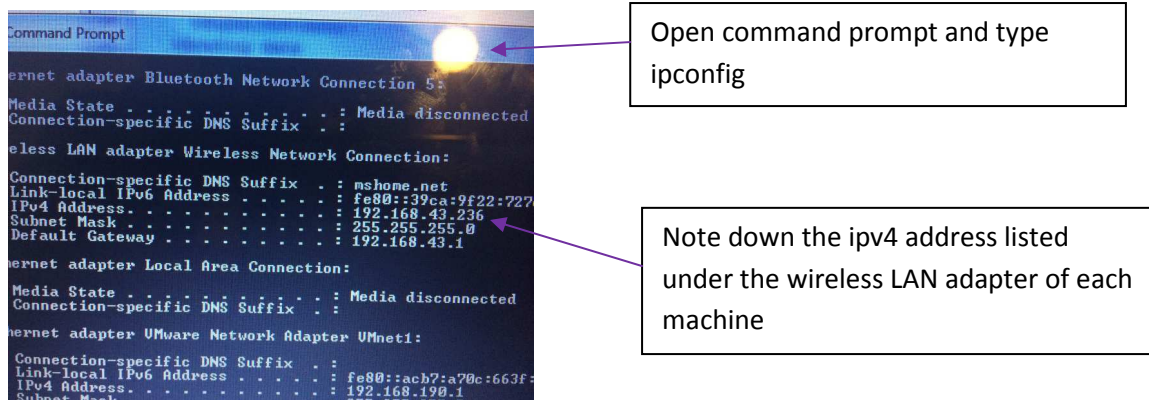


Connect to the access point which you created, using the passkey you set above

c) Turnoff windows firewall of each machine (be sure to turn it back on before connecting to the internet).



d) Note down the IP addresses of the machines, by using the ipconfig command typed on the command prompt.



For this example assume the IP address of machine 1 is :192.168.43.236 and the address of machine 2 is: 192.163.43.237.

4. Settings for communications between two different machines(hosts):

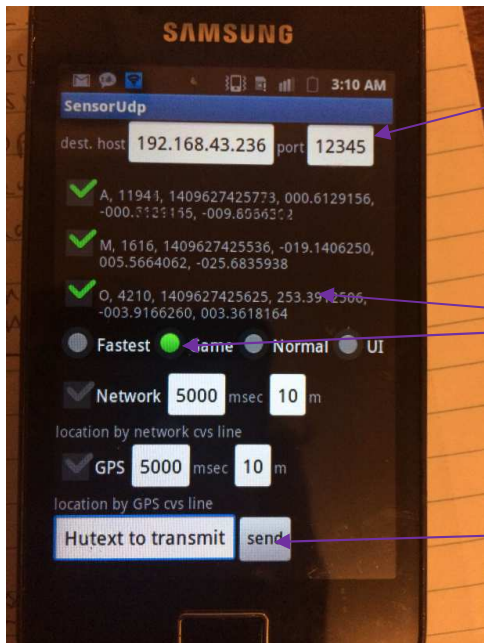
a) Network communication setup for Matlab programs on 2 different machines using TCP protocol.

Machine1	Local IP	192.168.43.236	Machine2	Local IP	192.163.43.237
Application X	Local Port	9000	Application Y	Local Port	Any
	Remote IP	0.0.0.0		Remote IP	192.163.43.236
	Remote Port	Any		Remote Port	9000
	Protocol	TCP /Server		Protocol	TCP/Client

Run Matlab on two machines connected to a mobile access point. An example of the configuration for communication is given above.

b) Network communication setup for UDP data reception from the sensor UDP app.

The sensorUDP app available for free, allows to send data to a UDP port. Following is an example communication setting to use to receive data from this App to matlab.



Set the IP address and the PORT of the Matlab script running on your computer

Enable the data strings you want to send and the speed.

Type some text here and press send to initiate communication

Machine1 MatLab	Local IP	192.168.43.236	Android device SensorUDP App	Local IP	(irrelevant)
	Local Port	12345		Local Port	(irrelevant)
	Remote IP	(irrelevant)		Remote IP	192.168.43.236
	Remote Port	(irrelevant)		Remote Port	12345
	Protocol	UDP		Protocol	UDP

Following is a n example Matlab script which would plot the received data. Only the O (orientation) data is enabled at the App.

```
instrreset
clear
clc
%Creating UDP object
UDPComIn=udp('0.0.0.0','LocalPort',12345);
set(UDPComIn,'DatagramTerminateMode','off')
%Creating bar plot for sensor visualization
figure
sensorbar=bar([0,0,0]);
fopen(UDPComIn);
longestLag=0;
while 1
    tic
```

```

csvdata=fscanf(UDPComIn);
scandata=textscan(csvdata,'%s %f %f %f %f %f','Delimiter',' ');
data=[scandata{4},scandata{5},scandata{6}];
if size(data,1)==1
    data(1)=data(1);
    set(sensorbar,'YData',data(1,:))
    axis([0.5,3.5,-180,180])
    pause(0.00001)
end
clc
disp('Data received:')
disp(data)
disp('Longest Lag:')
disp(longestLag)

t=toc;
longestLag=max(t,longestLag);
end
%Closing UDP communication
fclose(UDPComIn);
%Deleting UDP communication
delete(UDPComIn)
%CodeEnd-

```