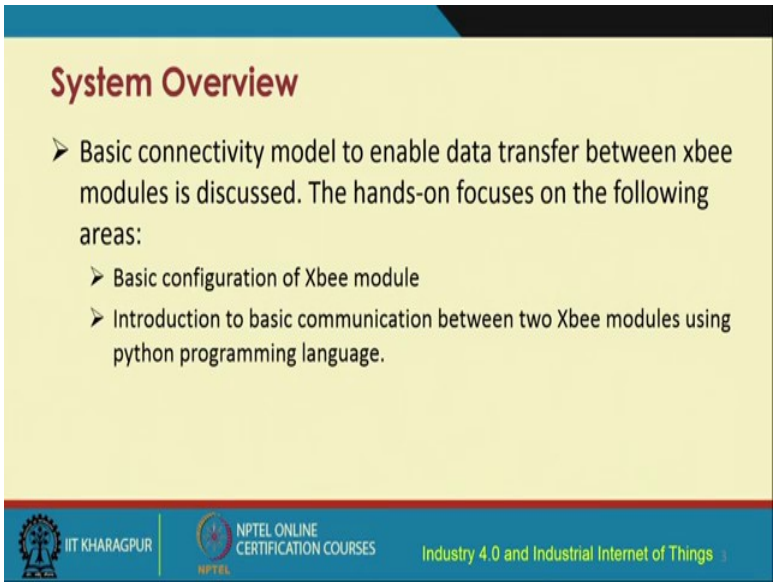
**Introduction to Industry 4.0 and Industrial Internet of Things**

**Prof. Sudip Misra**
**Department of Computer Science and Engineering**
**Indian Institute of Technology, Kharagpur**

**Lecture – 32**
**Key Enablers of Industrial IoT: Connectivity – Part 5**

So, this is going to be the last lecture in the series on connectivity for IIoT. So here, basically I am going to give you a demo of the ZigBee. So, before I do so, let me just try to go through some of these basics of ZigBee how it works and how you can configure ZigBee for working. And thereafter, I am going to show you this ZigBee in ZigBee communication in action. So, basically we will have a ZigBee transmitter and a ZigBee receiver. And I am going to show you how this transmitter sends and the receiver receives and basically shows the sent information from the transmitter.

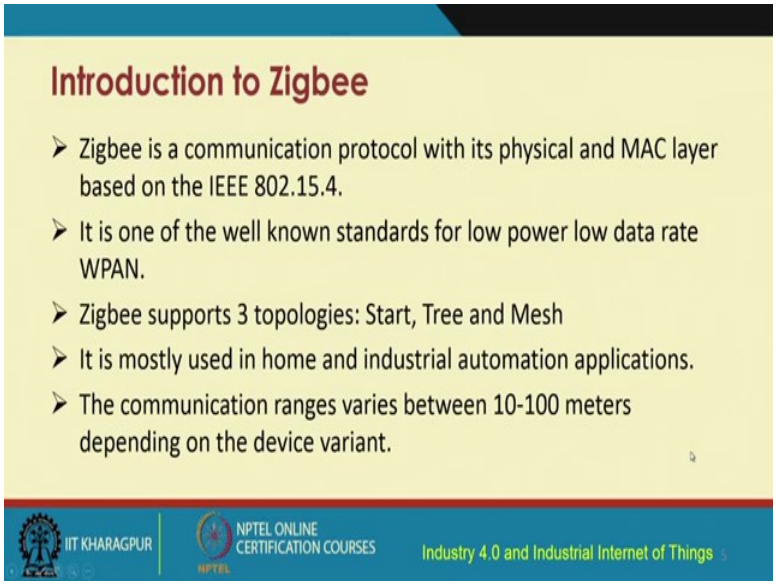
(Refer Slide Time: 01:03)




So, for ZigBee connectivity, the basic connectivity model that will be used for data transfer between the ZigBee modules is discussed. And particularly we are using the Xbee for doing it. And basically we are going to focus over here, this Xbee, which is basically also pronounced as ZigBee, but is a product from one of these companies a specific company, specific product which follows the ZigBee protocol. This Xbee module and its configuration I am going to show and thereafter I am going to introduce

the basic communication between 2 Xbee modules, using python programming language, that I am going to do next.

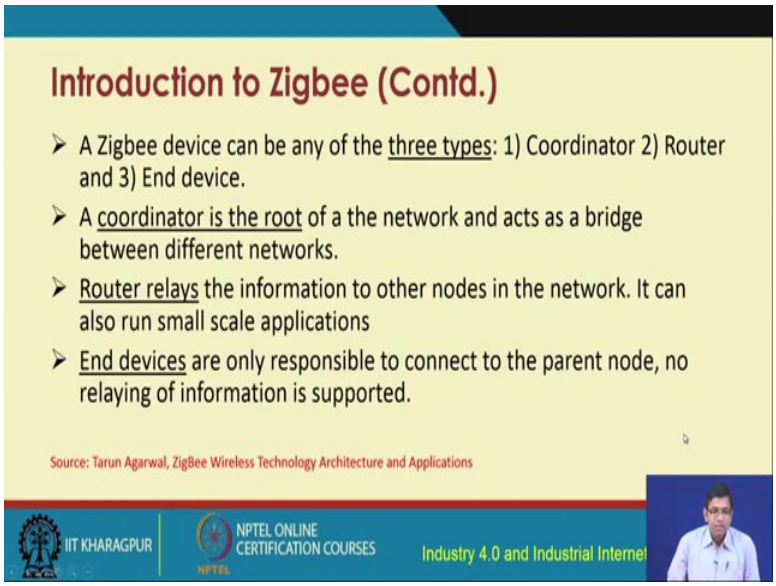
(Refer Slide Time: 01:52)




So, just a recap, ZigBee is a communication protocol that is used for mid range communication and its it follows the specification of IEEE 802.15.4. And it is one of the very well known protocols for IoT, for whether it is for home based applications or for industry based applications, ZigBee is a very popular technology. And it is very popular for low data rate, low power communication in wireless personal area networks. There are different topologies that are supported by ZigBee, star topology, tree topology, and mesh topology. Particularly, the star and the mesh topologies are the most common and are widely used for home and industrial automation applications.

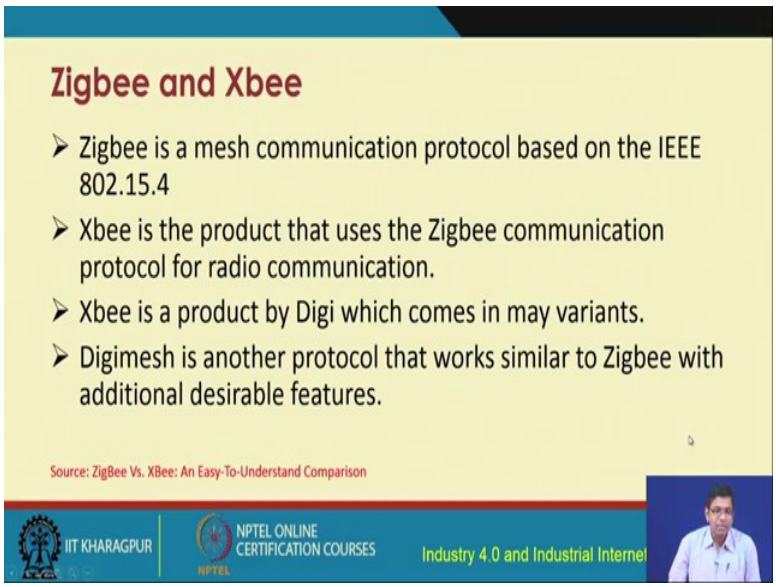
The communication range in ZigBee varies from 10 to 100 meters. It becomes more, if that the receiver and the transmitter are in the line of sight of each other.

(Refer Slide Time: 02:52)




So, a ZigBee module basically can be of 3 types. There can be a coordinator, a router, and an end device. The coordinator is the root of the network and it acts as the bridge between these different networks. The router basically relays the information to other nodes in the network, and the end devices are only responsible to connect to the parent node, and no relaying of information is done by the end devices.

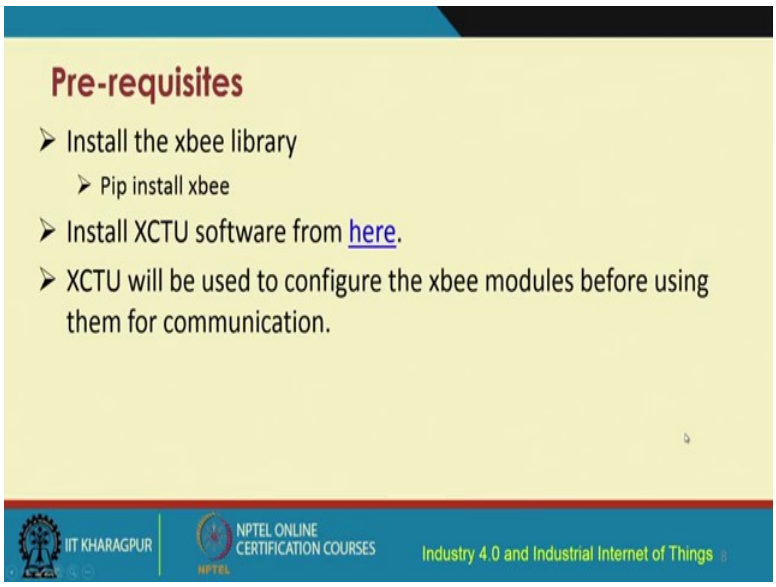
(Refer Slide Time: 03:18)




So, these are the 3 different types of devices that are supported by Zigbee.

So, ZigBee and Xbee, although these terms are used interchangeably, but they are different. So, ZigBee protocol is a mesh communication protocol which is based on IEEE 802.15.4 standard specifications. Whereas, Xbee is a product from this from this from the company Digi, and it comes in different variants. It is available commercially in different variants. So, Digi-mesh is a protocol that basically is similar to Zigbee, but has certain additional features. So, Digi-mesh is basically supported by Xbee and it has certain features that are similar to ZigBee, but certain additional desirable features are also their Digi-mesh.

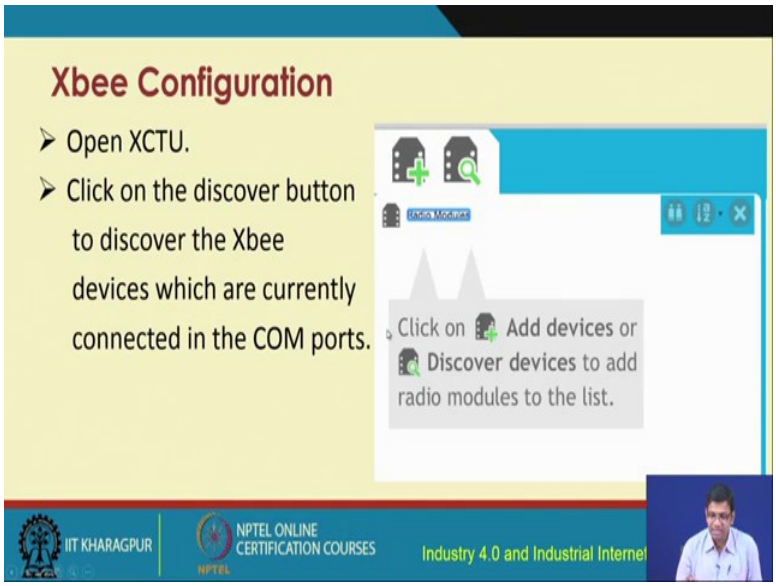
(Refer Slide Time: 04:07)




So, these are the prerequisites. Initially, you have to install the Xbee library, using this command pip install, Xbee and then install the XCTU software from this particular link that is given. So, you can click on this particular link, and you would be taken to that. From this link you would be able to get the download and the installation of this software XCTU and that is required to be installed even before you can do this configuration.

So, XCTU will be used to configure the Xbee modules before they can be used for communication; that means the ZigBee center and the transmitters could be used.

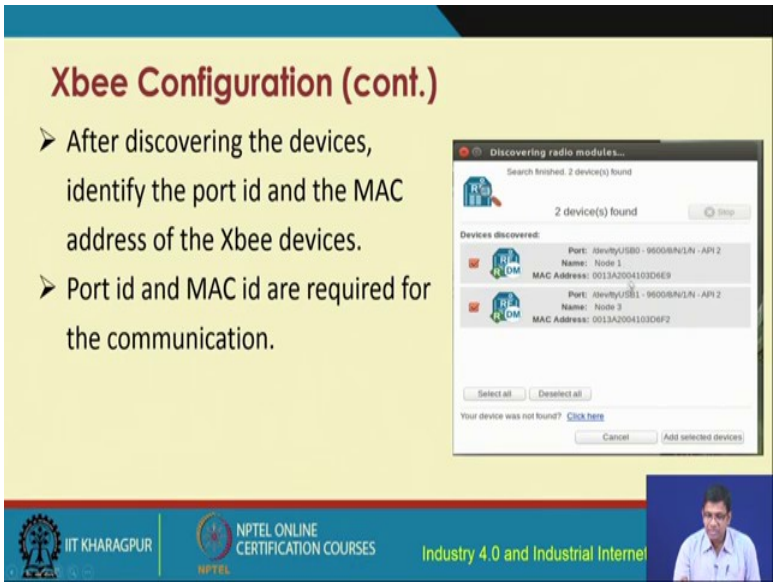
(Refer Slide Time: 04:45)




So, for configuration, after you have installed XCTU you open the XCTU window like this and then you click on the discover button to discover the Xbee. So, basically once you do that, you would be able to add the different devices and discover devices. For adding devices, this is the button that you will be using and for discovering this is basically the button that you are going to use.

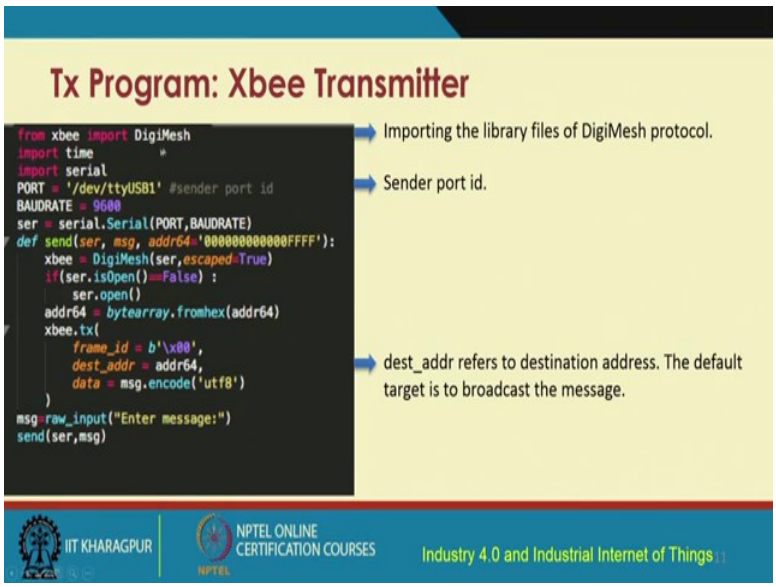
So, once you click on this particular button it has discovered different modules and this is this module that has been discovered and has been found. So, there could be many different other modules, that could be discovered through this discovery process.

(Refer Slide Time: 05:26)



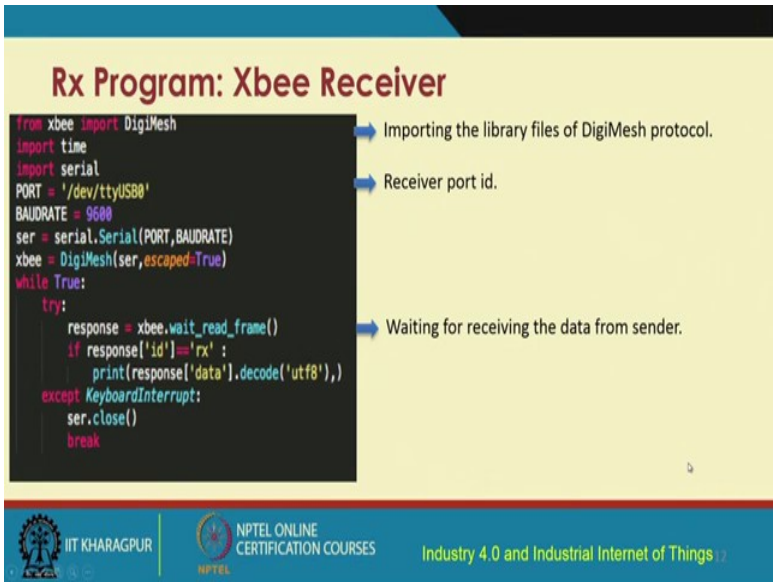

So, after discovering that this devices, it is required to identify the port id and the MAC address of the Xbee devices. And this port id and the MAC id are required for this communication. And this is quite obvious and here as you can see based on these different devices that are discovered the corresponding port id and the MAC id, will have to be assigned, will have to be found out. And the names of these devices will also have to be assigned and so, here the names are node 1 and node 3, but you could change, you could over here, you could change the names of these devices.

(Refer Slide Time: 06:01)



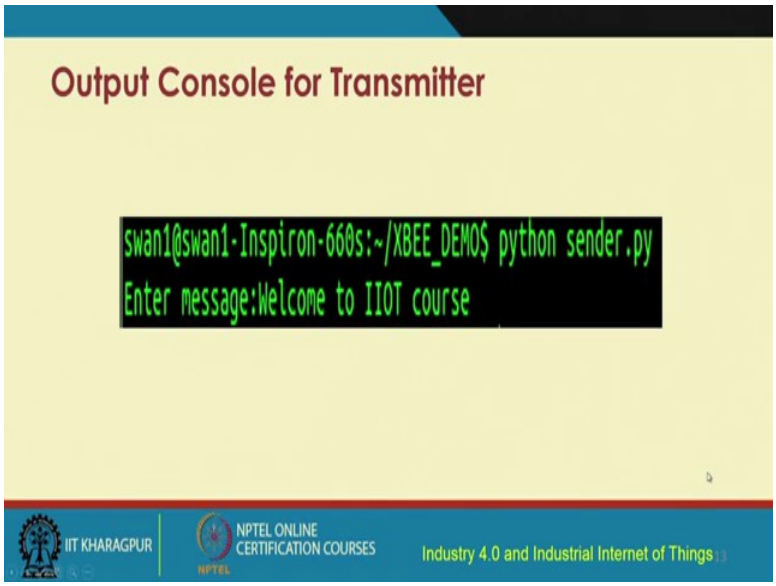
So, in that for the transmitter program, for Xbee this is basically a screenshot for the transmitter. this part basically talks about importing the Digi-Mesh library. This is the Digi-Mesh library, the sender port id is sent is set basically through this particular comment line. And for the destination address this destination address is set through this particular line of code in python and the default target is the broadcast mode of communication.

(Refer Slide Time: 06:40)



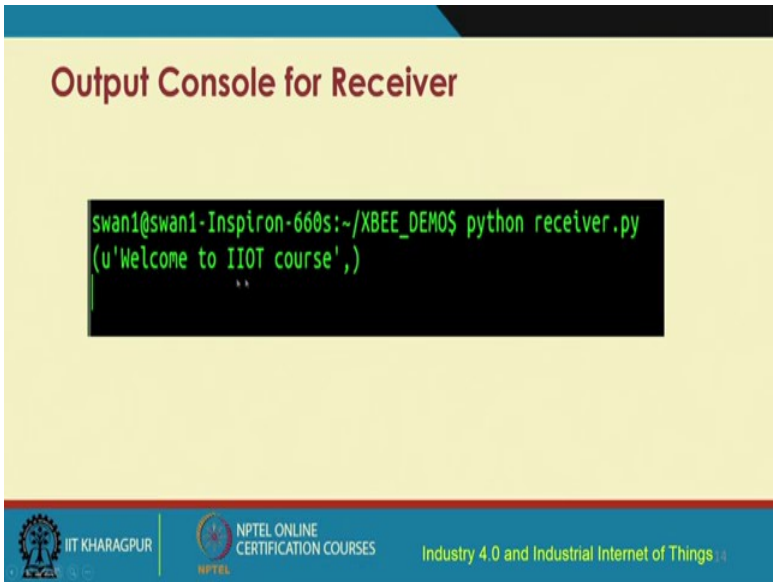

This is the receiver side program. again you import the Digi-Mesh and because this Digi-Mesh library is the backbone the enabler for supporting this communication. So, it is very important to imp import for the transmitter as well as the receiver Xbee modules, it is imp it is important to import this Digi-Mesh library. And then the port is set the receiver port id is set over here and this try block if you look at it is looking for, it is waiting for the receiving of the data from the that is sent from the sender. So, it is waiting for reading the frame and.

(Refer Slide Time: 07:24)




So, this is basically this block and this is the python code and so, output console for the transmitter is going to look like this. So, you can enter the message that you want to send from the transmitter. Let us say that we type in welcome to IIoT course.

(Refer Slide Time: 07:37)

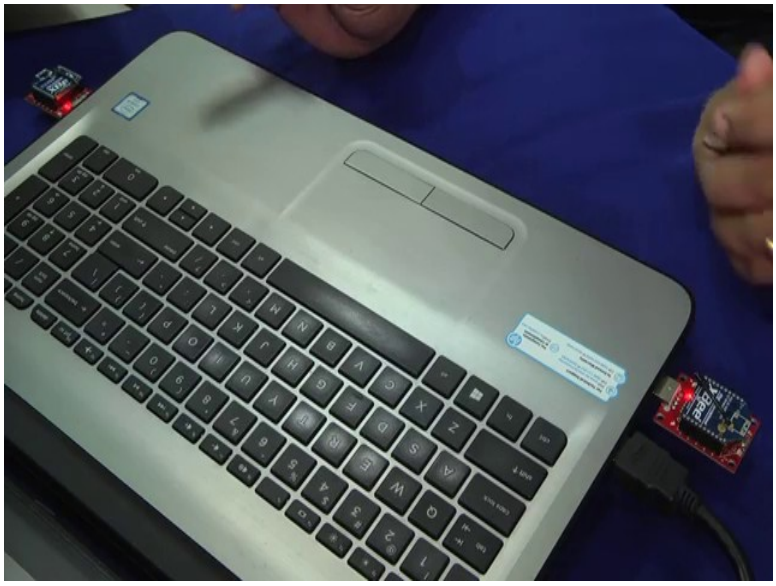



And then at the receiver, you are going to receive this message ‘welcome to IIoT course’ and u stands for unique code. And so let us not worry about it and this is basically this message that is received at the receiver end.

(Refer Slide Time: 07:49)


So, with this, we come to an end of this particular lecture we have seen the configuration of ZigBee, and the sender and the transmitter module configuration, their corresponding code, and how the data can be sent from the transmitter to that receiver. So, now, I am going to show you whatever I have explained so far, the XCTU configuration and also this python code for the ZigBee transmitter and the receiver, I am going to show it in action.

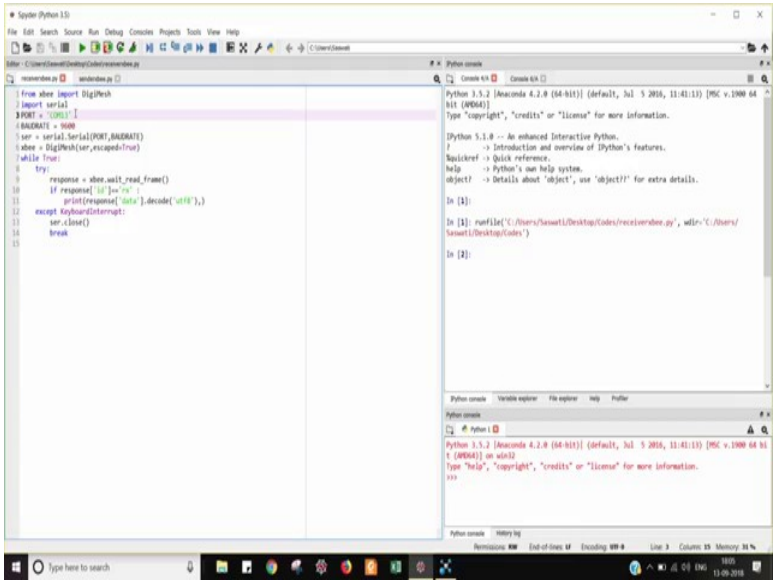
(Refer Slide Time: 08:27)



So, as you can see over here we have these two ZigBee modules. this one, is the transmitter a ZigBee module, and this is the receiver ZigBee module, which have been com connected to the laptop, because ZigBee modules can connect over USB. So, for convenience we have connected over laptop.

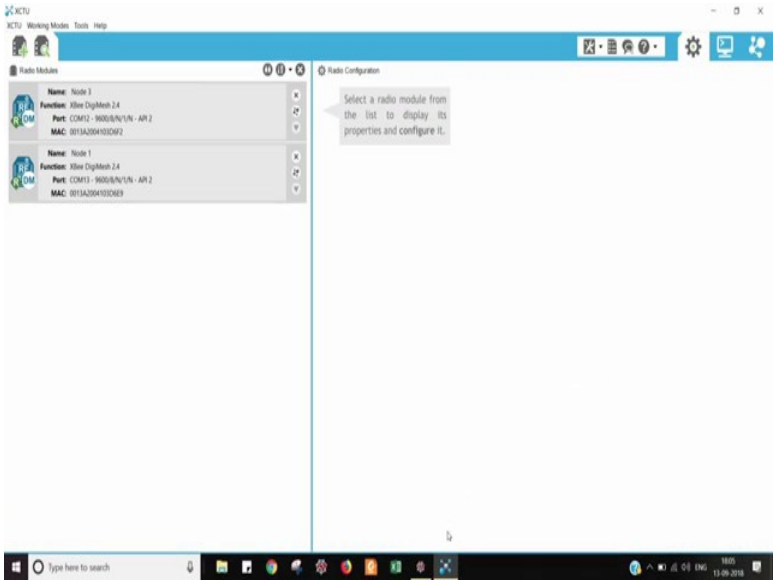
So, this is the transmitter module and this is the receiver module and we will show that how this communication is taking place. Right?

(Refer Slide Time: 08:55)



So, we have already written the code that will be required for it, but before that let me show you. let me show you, this XCTU configuration windows.

(Refer Slide Time: 09:03)




So, this is basically this XCTU configuration window. And if you recall, that in the slide I had shown you that you will have to first configure these different nodes regarding, which port they are going to work, which one will be the sender port, which will be the receiver port, what will be the names of these different nodes? The MAC ids and all these things will have to be configured. Many other configure configurations you can do, but those can be done in the XCTU configuration screen.

So, after you have done that basically, you can then transmit the data and then also receive it. So, this is basically the receiver code in python we have this receiver code receiver Xbee dot p y. and this is basically there is a while loop trying to read the data frame as I was showing you in the slide earlier. And it is trying to look for whether any there is any data that has been sent and this is this imported Digi-Mesh library this is required as I told you already.

And so, let us execute this one over here, and thereafter let us go to the other code which is the sender x b dot pi. this is the other code here, also we have imported the Digi-Mesh library and we have set the port and also we try to form this message that is going to be sent. this message is sent through this command send acr mes msg, and msg is this message that is built and is being sent. So, after this we go to this console to see that how it looks like from the sender side.

So, that was the receiver one and this is the sender console. We execute over here, we run, and we send a message. So we can type in any message, let us say, hello world, hello world, and then we send it. So, it is being sent from the transmitter side. We send it, and we go back, and we can see whether it has been received or, not. So, as we can see over here, it has been received. Hello world has been received at the receiver side. This is the receiver console and that was the sender console. From the sender console, it was sent and from the receiver console, we are able to receive this.

So, you can try it out yourselves, and see whether things are working for you or, not. It is very simple and there are lot of open source code available for supporting ZigBee programming and so on. particularly with python and you can try out this particular setup that I have demoed you today.

(Refer Slide Time: 11:50)


So, these are these references that you could go through and if you are interested. You could go through any of these references, and I would encourage you to try out this particular setup, that I have just explained. And try to send some data from the transmitter and see whether it has been correctly received at the receiver or not, because that is the basic building block for ZigBee communication.

Thank you.