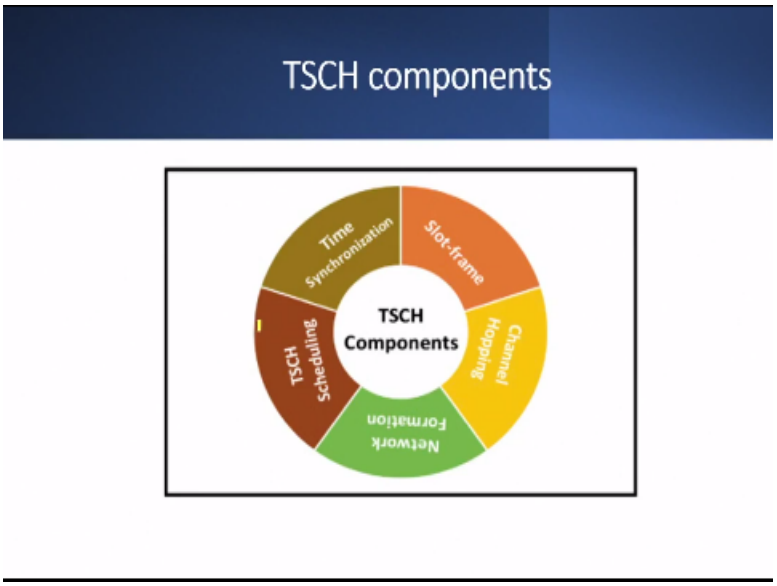
Design for Internet of Things
Prof. T V Prabhakar
Department of Electronic Systems Engineering
Indian Institute of Science-Bengaluru

Lecture - 45
IEEE 802.15.4e – 03

What is the demo that we have here?

(Refer Slide Time: 00:36)



Let me point you to the most important part of the TSCH components essentially include time synchronization. Why is that required? As I mentioned to you, you want to get deterministic access to the network. That means, if you are a Iot industrial IoT sensor node and you have to communicate this data to your parent so that parent in turn can give it to another parent and so on and reach the aggregation node.

Depending on how deep you are away from the aggregation node you need that access in deterministic time. That means, when you are transmitting the receiver should be ready to get you right, your parent should be able to get you. And when the parent is transmitting, its parent should be ready to get it so that synchronization is required.

Not only synchronization is required also that time over which deterministic time over which the data packet has to go has also got to be fixed. Therefore, one major challenge that you have to solve if you are building large TSCH networks from

scratch is this problem of time synchronization, okay. Then there is this issue of scheduling. In scheduling, you may want to look at priority of packets.

Some packets need more priority as compared to other type of packets. So whenever you have prioritization that is required scheduling will definitely have to come in place. The third important challenge indeed is the network formation. In network formation essentially what we are saying is that suppose you want to build such a large network, how do new nodes join the network?

How do nodes exit from the network? How is slots allotted to it? How can you ensure that this whole thing is ultra-low power because after you get a slot you may want to sleep. And if you sleep and your clock is down or perhaps running at an extremely low frequency you must be able to get up and get back to synchronization of the complete system, right?

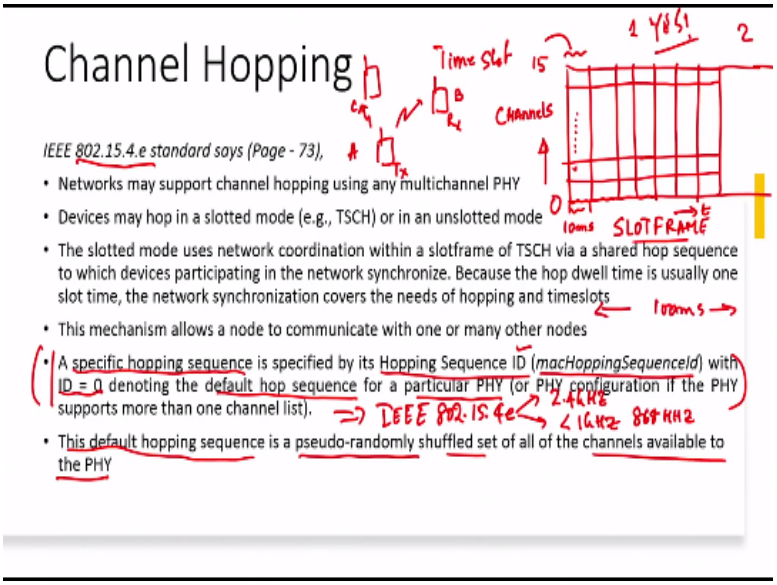
Joining of nodes, exiting of nodes, making groups all of that comes under network formation. And of course, the most important thing is to battle other interferences coming in the same frequency band. Essentially, if you are talking about 2.4 gigahertz band, that band is highly crowded, right? You have Bluetooth, you have Wi-Fi, you have microwave ovens, you have other type of radios which are all in the 2.4 gigahertz.

So you want to ensure that you, that somehow periodically you should be able to hop channels so that you can battle interference on a given narrow band and move on to another band over which you can do a transmission. So all of that essentially means that you should also implement channel hopping. And of course, deciding how many nodes are required in your network actually will determine what is known as the slot frame size, okay?

So that is another challenge, how to decide and how to determine the slot frame size and so on is what is another challenge. Today, we cannot do all of it. And demonstration has to be limited to one specific module in this and that indeed is the

channel hopping part of the demonstration. Alright, so what actually it means is the following.

(Refer Slide Time: 03:48)



The standard if you look right, the standard on page 73 of the standard, if you pick, download the standard and look at it, this is what is written in the standard. Networks may support channel hopping using any multichannel PHY. And devices may hop in a slotted mode or in unslotted mode. Essentially we are looking at slotted mode. So it is indeed going to be TSCH kind of network.

Standard does not differentiate though, but we have done an implementation. We need this implementation because we want to do it for the industrial environments. Now if you look at slotted mode, it uses network coordination within a slot frame of TSCH via shared hop sequence to which devices participating in the network synchronize.

Because the hop dwell time is usually one slot time, the network synchronization covers the needs of hopping and time slot. So let us look at what the standard is saying in a high level view, get a high level view of it. This is what is called the slot frame. So I will write it here, slot frame. And this slot frame repeats. This can be 1, the same thing repeats from here onwards, okay.

This will become 2. No space so I just leave it there, this is 2 okay. So just let us elaborate 1. What essentially you are doing is you are dividing it into, oh it is not so equal, but kindly pardon my drawing and this essentially is what is known as a slot time. This is a slot time or also it is called time slot, okay. It is also called time slot. Let us call it time slot, time slot and slot time.

Usually one slot time he says. But anyway, you can use it either interchangeably, time slot or slot time. Essentially it is divided into several slot, time slots. So you are slicing the slot frame into, this is time right? This axis is time. So you are just dividing it into small chunks of time. Now what is this division? Why is this so long here? This is so long because you have number of channels 0 to 15 channels you have.

So nodes essentially can remain on, can start with one base channel and they can start hopping anywhere here. They can keep helping and they can communicate. They can be used, they can be using two time slots in a slot frame or they may go on one time slot in this slot frame and go to another time slot in the next frame; that is also possible. So I think several possibilities exist here. But we will take a very simple case, all right.

So the channels on the x axis, channels right this is channels. So when you have to divide it into channels, you have to put a line like this, right? You have to put it here. So I will put dot dot dot dot so that I do not have to worry about making things even. Just leave it like this. So essentially he is saying the dwell time, this is the dwell time. You take any square here that is the dwell time of the channel, right?

So the dwell time is usually one slot time that is essentially this one, right? Any slot you can take and then mark it like that. The network synchronization covers the needs of hopping and time slots. So essentially as it is progressing in time, you should be able to take care of synchronization as well as the needs of hopping and time slots. So essentially that is what it is saying.

The mechanism allows a node to communicate with one or many other nodes, fantastic. So if you take a sensor node like this and it is speaking on a given within a given slot frame, it is talking on a particular time slot, obviously it is communicating to another node, which is in the reception mode, okay. This is in the transmit mode. Exactly in the same time slot, the receiver is in this node A and this node B.

B is in reception mode. It is possible that there is another node C, the whole network is synchronized, there is possible that a node C is also listening to what A is transmitting. So that is what it is saying here. It is saying that the mechanism allows a node to communicate with one or more other nodes, okay. Now here is the key. A specific hopping sequence, how should it hop?

In which channel should it be to begin with? And in which channel should it be at a later point in time is the question. So that is essentially hopping. The hopping sequence is specified by its hopping sequence ID and that is mentioned in the standard as this macHoppingSequence ID with ID equal to 0 denoting the default hop sequence for a particular PHY, okay.

So we will get to that as we go along because this is there is the standard desk talk about what is the specific hopping sequence. Because you have to battle interference, so to keep changing channels and you have to hop from one channel to the other across slot frames. And therefore, you must know the sequence. So essentially the standard is talking about how it should be done.

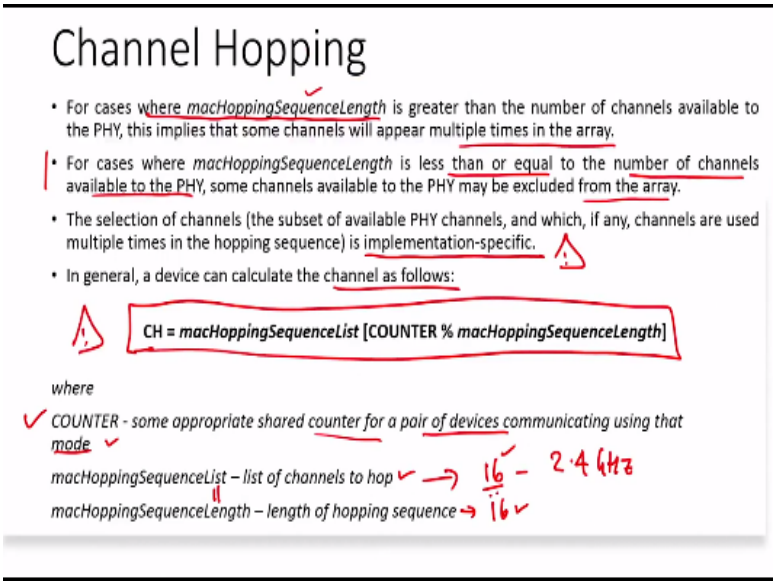
It may not tell you any implementation detail, but will tell you what should be done in order to meet the requirements of the standard, okay. So essentially, this is the 15.4 e standard. This default hopping sequence is a pseudo-random shuffled set of all the channels available to the PHY. So what essentially it is saying is that there might be a PHY.

See essentially the problem why this is a little cryptic is also to understand from the following manner that if you look at IEEE 802.15.4e, the standard operates in several

frequencies. One of the frequencies it operates is 2.4 gigahertz. It is also possible that it will operate in the sub 1 gigahertz range which is the 868 I think megahertz range, never mind. All it simply means is that for 868 megahertz it may be different range of channels which are available.

But for 2.4 which is the most popular you get 15 channels 16 channels 0 to 15. So can I now call this picture that I drew specific to 2.4 gigahertz? The big answer is yes. This structure which I described is for 2.4 gigahertz, right? So that is the key point. All right, so now we will move on to the next slide.

(Refer Slide Time: 11:12)



And now here it says cases where macHoppingSequenceLength is greater than the number of channels available to the PHY, this implies that some channels will appear multiple times in the array. This is one case. It is actually standard takes care of all kinds of situations where you have a number of channels which are available in the PHY.

Greater number of channels this one this sequence length is greater than the number of channels you have a case. For cases where macHoppingSequenceLength is less than or equal to the number of channels available to the PHY, some channels available to the may be excluded from the array, okay. This is also possible, okay. All these cases have to be handled, because this is a standard.

And we have to read the standard carefully, interpret it and then implement it. I will not go into great detail because this is not a course on standard implementation. This is something that we wanted to show you demonstrate to you about how channel hopping actually is working. The selection of channels, the subset of channels available, PHY channels and which if any channels are used in multiple times in the hopping sequence is implementation specific.

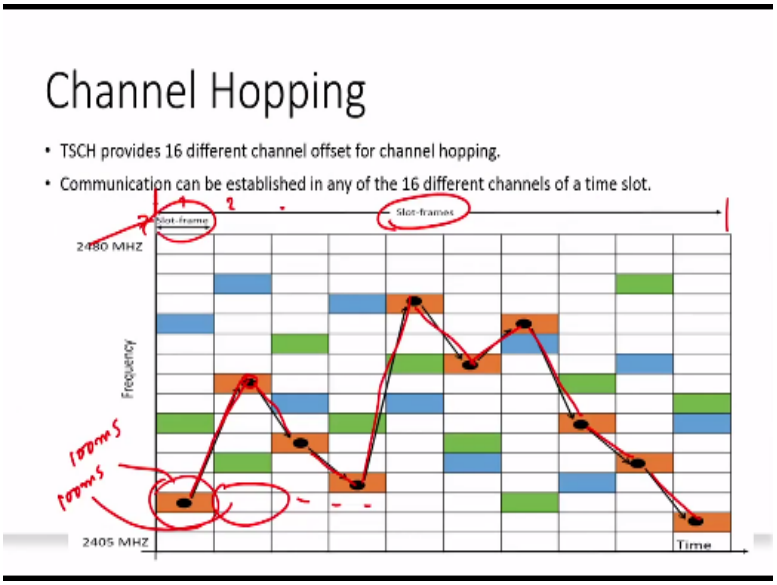
So that is it. Everything by any standard IEEE document will not detail anything related to implementation. It will only tell you what you should do and how you want to do is up to you. So the how is never mentioned in a standard. In general a device can calculate the channel as follows. So here is the basic equation, okay.

This equation says use the macHoppingSequenceList and you basically start with some number counter and then you do a mod of the macHoppingSequenceLength essentially. That is what you will end up with your next channel that you want to, you have to hop to. Now that he mentioned here clearly. What is counter? Some appropriate shared counter for a pair of devices communicating using that mode, done.

MacHoppingSequenceList, list of channels to hop. In our case it might be 16 because in the 2.4 gigahertz range, for 2.4 gigahertz range, you know that it will give you 0 to 15 which is 16. Usually, macHoppingSequenceList is usually equal to the macHoppingSequenceLength, okay.

And the sequence list is 16, I mentioned to you already and therefore, macHoppingSequenceLength is the length of the hopping sequence is also 16. So we have more or less fixed this. There may be special cases which you may want to consider. That is really out of scope at the moment.

(Refer Slide Time: 14:02)



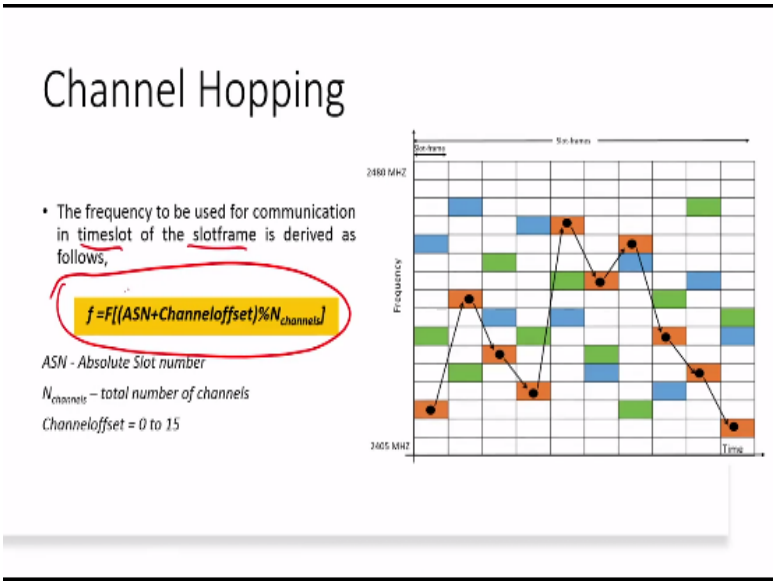
Now you look at this beautiful picture here. x axis you can see that first of all look at the picture and understand all that I wrote earlier, it is coming back here. You have a slot frame, it starting from here. And then it is so these are so many slot frames, right? These are all one here, two here and so on. So he is talking about so many slot frames, which essentially lead you to what is known as a slot frame, right?

This is what we are talking about. And this is the slot frame. And see this picture is interesting, right? This picture is in because if you can think about this whole slot frame as one here, okay. And if you take this each is as 10 milliseconds, take this as 10 milliseconds. Take this as 10 milliseconds. 10, 10 millisecond slots. So that is 100 milliseconds. So you can talk about 100 milliseconds as a possible slot frame.

So think about this as 100 milliseconds. This is the next 100 milliseconds, okay. Think about this as the next 100 milliseconds and so on. Now see the beauty. What is actually happening? In the first slot frame, it is in some channel and then it moves on. In the next slot frame it has gone to some other channel. You can see y axis is frequency, x axis is time. It has gone to some other channel.

So it goes up, goes like this. And this sequence of hopping is fixed, right? Because everybody has a rule on how to hop and that is given by this expression, this expression gives you the hopping sequence, right?

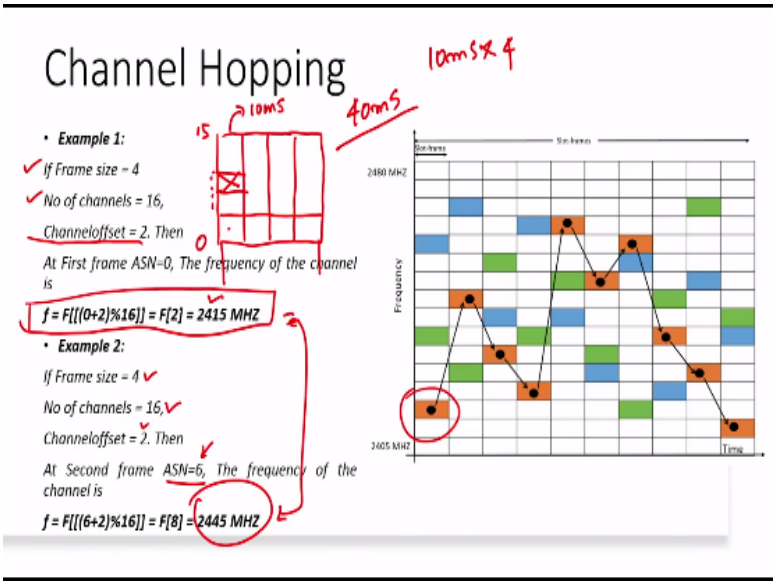
(Refer Slide Time: 16:07)



So and this is what you have to implement as closely as possible. In fact, you can see now what is the f to settle down? Here you are, you are essentially saying f is absolute slot number plus the channel offset modulo the number of channels, right? So that is essentially what it is. If you look at n channels is the total number of channels and then the channel offset is 16 and this is a function of that.

So the frequency to be used for communication in time slot of a slot frame is derived as follows. You can see this is a very simple expression out there.

(Refer Slide Time: 16:44)



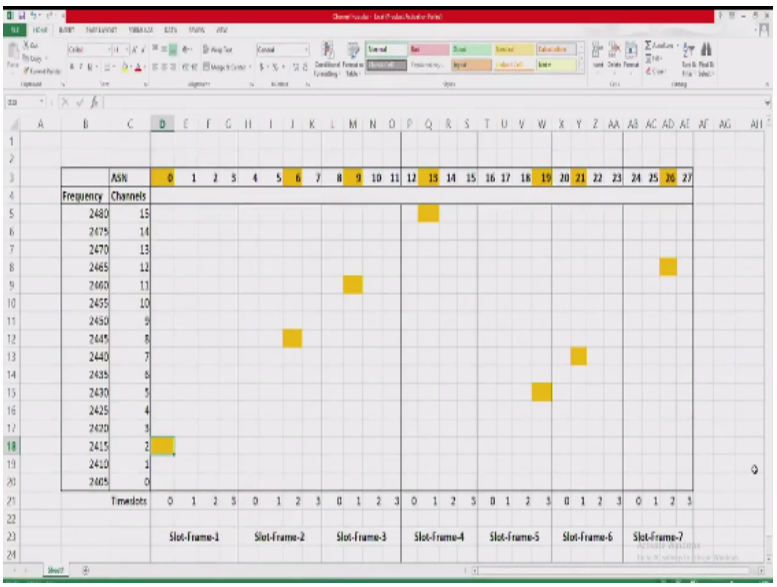
So let us take a very practical example which was built by one of our colleagues Ms. Sahana and we will see how that demonstration runs. Just to give you an overview of the example, think of a case where the slot frame size is 40 milliseconds, okay. The overall slot frame is 40 milliseconds. There are 4 time slots of 10 milliseconds okay, there are 4 time slots of 10 milliseconds.

So 1, 2, 3, 4 these are the 4 and a node can hop anywhere from 0 to 15. That means it can take any channel starting with from 0 to 15 it can take any channel. And thereafter it should do have a proper sequence of hopping, okay. So that is essentially what we have written here. Slot frame is 4, number of channels is 16, channel offset is 2. Then at first frame ASN is equal to 0.

The frequency of the channel is given by this same expression which we mentioned to you previously. This is the same one that you find here, okay. You apply this expression, you will get this frequency. So it will hop to a channel in which this is the frequency of the, of that particular channel. Similarly, if you take another example, continuing the frame size is 4, the number of channels being 16, now the channel offset was 2.

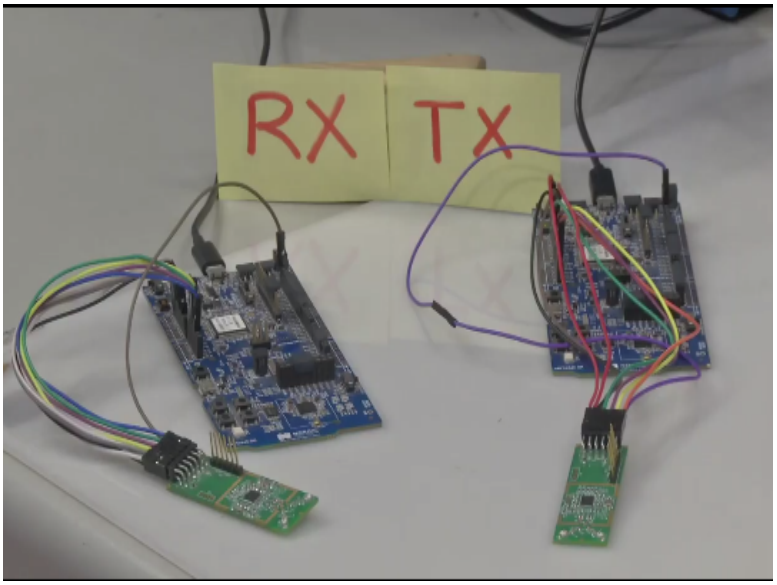
Then at the second frame, this is like continuing from the previous frame to the next frame, the second frame will give you ASN of 6. And corresponding to that ASN of 6, if you apply the same expression that we show here right, you apply the same expression here, you will find that it will go and sit on a channel which is 2445 megahertz. It is good to see a demonstration of this.

(Refer Slide Time: 18:50)



And therefore let us shift to this screen which essentially has the picture that I show here. But before you go on to that picture, it is important to note how we are going to demonstrate this complete system. For that I will draw your attention to this picture here.

(Refer Slide Time: 19:06)



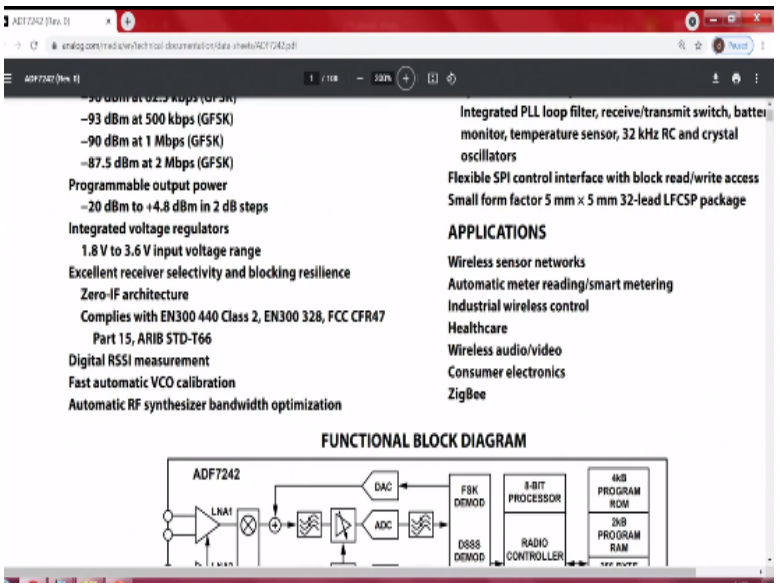
Here what you see is a transmit node which is here and a receiver node which is here. You see two pieces of hardware on either side. What you see here is essentially the Nordic 52840 controller. We are only using its software, okay. We are only using its controller option. We are not using any radio option. We are not using any other function except the controller part.

Of course you can connect ADC, you can use sensors and so on and so forth. But what is interesting is, we are building our own 15.4 e TSCH stack, which requires you to build a full state machine of the MAC. And that MAC is actually running on this controller and for that MAC the specific hardware PHY layer is this, what you see here okay, this board.

This board is essentially a another chip which essentially provides you only the PHY layer capability. So this MAC talks to this PHY layer through these wires of course, and what is good is the complete software implementation of the TSCH MAC implementation. Similarly, we have a receiver which is again commercial PHY layer chip and to that is connected via these wires call it to this microcontroller.

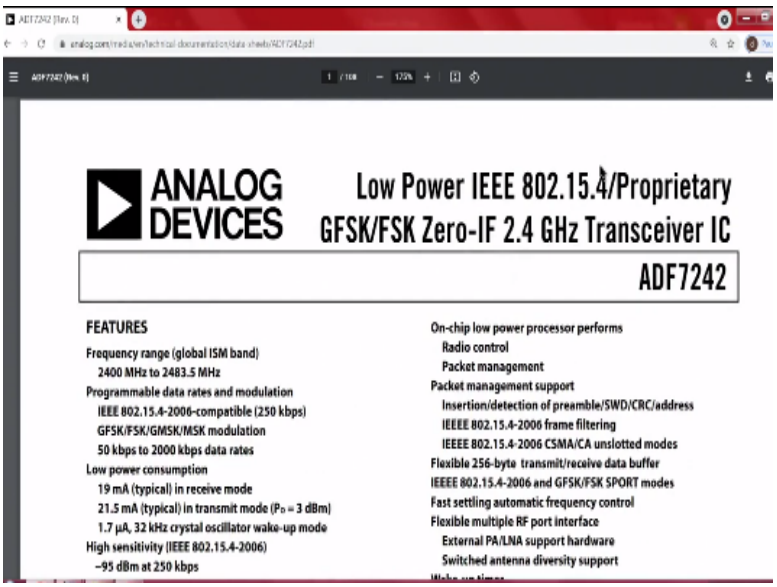
And again we are using 52840 as the controller of option. Now question is what is this chip? Why did we choose this paradigm? I will show you why we chose this paradigm.

(Refer Slide Time: 20:59)



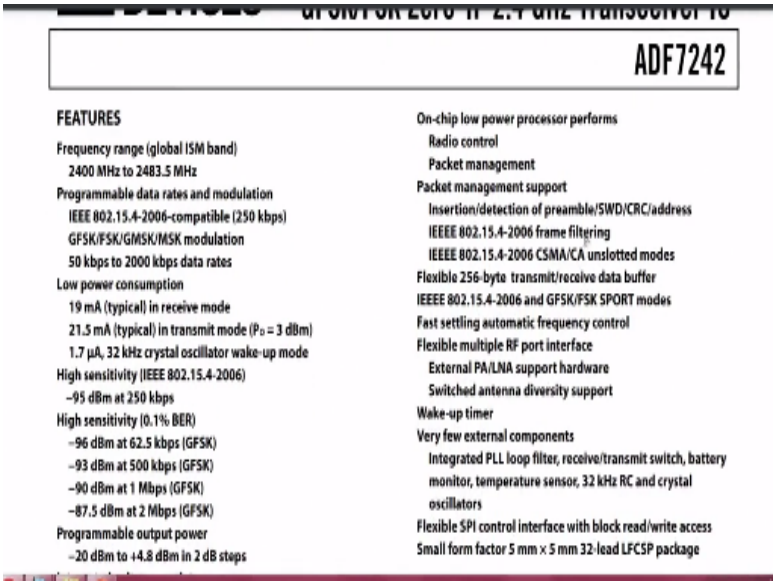
The reason is this chip actually is very generic, is very generic okay. And what is this chip?

(Refer Slide Time: 21:07)



This is ADF7242 low power IEEE 802.15.4 proprietary GFSK/FSK IF 2.4 gigahertz transceiver IC. Just nothing but a transceiver IC, okay.

(Refer Slide Time: 21:23)



And it has some features and what are those features. You can see that it very well supports the range of frequencies that TSCH requires, programmable data rates. This is a generic or a general PHY layer chip. This is to experiment our own MAC folks that we are building, right?

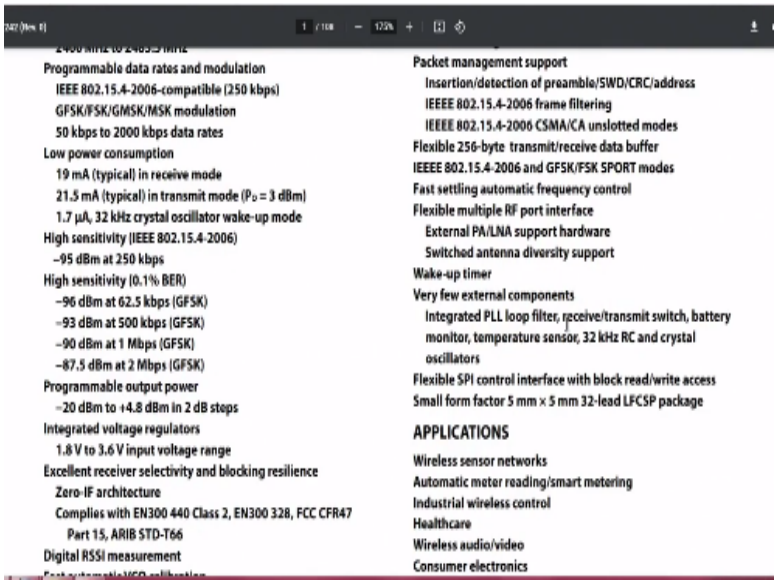
And it has a sensitivity. We discussed this sensitivity in great detail of -95 dBm and it has high sensitivity of with measurement of 0.1% BER and it tells you the different modulation and coding schemes and the associated sensitivities. And of course the

power amplifier can be programmed from -20 dBm to +4.8 dBm in steps of 2 dB. It has integrated voltage regulators, okay.

It has excellent receiver selectivity and blocking resilience. It has on chip low power processor platforms. It has packet management support. It has flexible 256 byte transmit and receive buffer, which means the MAC can hand over the data to this chip and the chip can manage it all by itself because it has independent 256 byte transmit and receive buffer. I suppose it has 128 into 2. 128 for transmit and 128 for receive.

So essentially giving you 256. So for transmit it is 128 and for receive it is 128, okay.

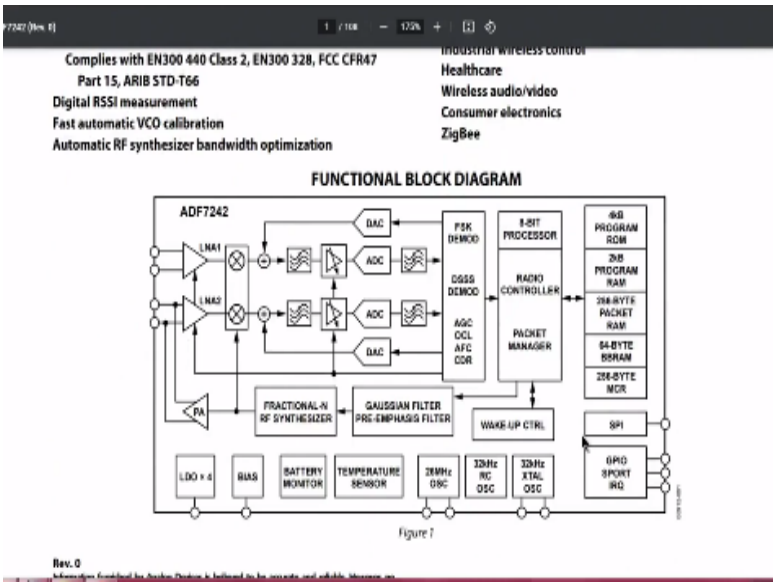
(Refer Slide Time: 23:00)



Now you can see that a flexible multiple RF port support. In other words, if you do not like the transmission power that this guy is pumping out, you can use this chip and apply by an external power amplifier or a low noise amplifier and improve the sensitivity by simply interfacing external components to this chip, okay. And it has a wakeup timer, very few external components and so on.

And it provides you SPI interface between the controller and this particular chip. And therefore, it does provide you good amount of block read and write access. And it has indicated several applications; wireless, sensor networks, industrial wireless, health care and so on and so forth.

(Refer Slide Time: 23:48)



Now that is not the point right folks? So many times I mentioned to you about the transmit and receive chain. But now you can see there is a super opportunity to understand what exactly this whole chain and how exactly it works. I am sure you recognize that in the towards the transmission side there is the DAC, right. And on the reception side, there is the ADC. So you can see these two blocks, right.

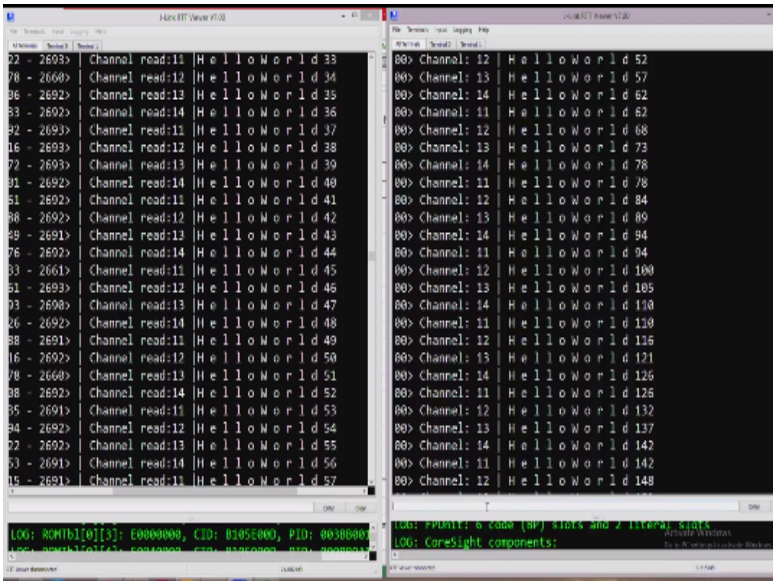
And of course, whatever comes from the low noise amplifier passes through, you know filters. Does all the, passes through the local oscillator, then basically demodulates and down converts the signal and then gives it to the ADC and you get it into digital form here and so on it processes. Similarly, if you want to do a transmission, you would pass it through upconversion block.

And then get back an analog signal, upconverted signal, and then you would pass it through the power amplifier and then it would simply go out onto the air interface. So this picture sort of highlights everything that is around the ADF 7242 chip. And there are other you know advantages of using this chip because it gives you low dropout regulation, then battery monitoring and other sensors on board like temperature sensor and all that.

So it is quite feature rich. So the whole idea of this demonstration is to show you that we can take any controller of interest. It need not have been the Nordic 840 system, any controller, and implement the full state machine of the standard so that you will be able to build your own stack and optimize your own stack for your specific applications. So that is the idea.

Industrial settings may actually warrant you to build you know fast joining and tailor made systems yet interoperate or become interoperable with commercial systems. Alright. So having given you this background, let us now shift to the actual demonstration, which essentially is this picture here.

(Refer Slide Time: 26:14)



What you see on the screen here on the left side is the transmitter and what you see on the right side is the receiver. Let us pause once and then expand this picture so that we will be able to show you what exactly is happening there, okay. Now if you scroll you will see that the demonstration is basically the transmitter is pushing out a data payload which contains hello world, right?

It contains this information hello world. Look very carefully, you will see that 11, 12, 13, 14. Then again 11, 12, 13, 14 channels are used. In other words, first time it sends the packet in 11. Then it shifts to 12, then to 13 and then to 14 and so on, right? That

is what is happening on the left hand side. And on the right hand side, you see that it is receiving on 12, 13, 14. 11 got knocked out for some reason.

Then 12, 13, 14. Maybe it got clipped when we did a pause of the screen. But look carefully. 11, 12, 13, 14; 11, 12, 13, 14; 11, 12, 13, 14 and so on, it seems to be okay at the moment. But sometimes you may also have a missing packet on a particular channel. So it is actually hopping and yet transmitting the hello world, which is part of the payload. It will be nice to see with a good graphic to understand how this channel hopping works.

So let us shift to this channel. To begin with what you see is on the extreme left begins on channel 2415 and this is the basically the ASN 0, which I explained to you right in the picture. Now what happens is after the first slot frame, so essentially what is the frequency for channel offset 2? It is 2415 and then it is ASN 0. Let me put back this picture so that it connects back quite well to you.

So you see here, you have 16 channels, channel offset is 2. Then ASN is 0 and then it corresponds to 2415, which is showing here. Let us move the cursor to 2415. Now you can see it is matching with this, 2415. What happens in the next second frame okay, when it moves on to the next slot frame, see what happens. Now look at this screen here, it has gone from 2 it has settled down in ASN 6, okay.

That is ASN 6 and for that ASN 6, the frequency is 2445 and the channel is 8, okay. So you can take a very simple example. This is 2, this is 6. If you add them you get 8 and if you take the modulo it is still 8. Modulo of 16 it is still the 8. So it settles down on this 8, on this channel and that is corresponding to 2445, alright? Now if you do the same operation in the next slot frame, you come here.

You see that it is now gone to ASN 9 and to ASN 9 you get 2460. You see carefully from channel offset continues to remain at 2. 2 goes to 9. That means it becomes 11. And if it is 11 the corresponding frequency is 2460. This cycle repeats, it goes up and

then of course, at the end it has to roll back. So you see, you can actually see it going down.

So in other words, you can also plot a picture, which is very similar to what I show you here, right? Going up, going down, going up and going down and so on. This is how it is able to battle interference by shifting from one channel to the other channel, yet giving you deterministic access to the channel.

And extremely useful, and this is an extremely useful technique in the industrial environments where there can be lot of wireless congestion due to other radios in the vicinity. And thus implementation of 15.4e, its deeper understanding is a very important requirement. Thank you very much.