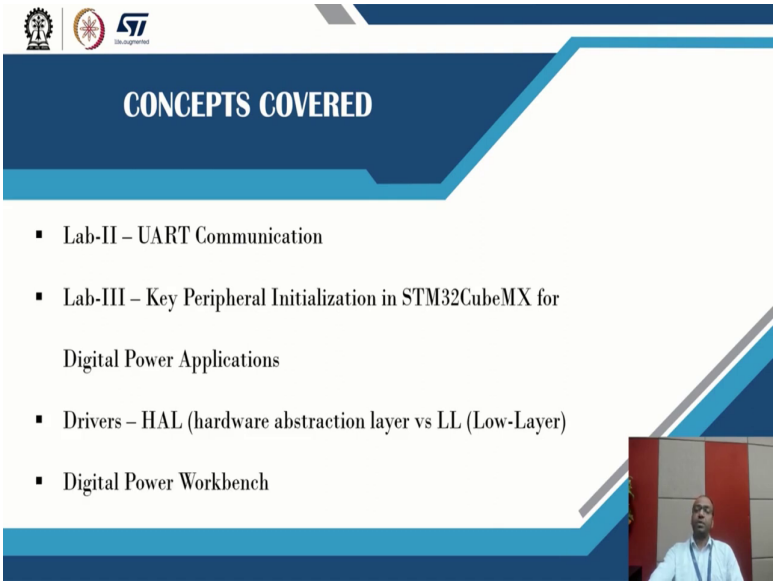
**Digital Control in Switched Mode Power Converters and FPGA-based Prototyping**

**Prof. Santanu Kapat**
**Department of Electrical Engineering**
**Indian Institute of Technology, Kharagpur**

**Module - 09**
**Digital Control Implementation using Microcontroller**
**Lecture - 83**
**Getting started with STM32CubeMX (Part - II)**

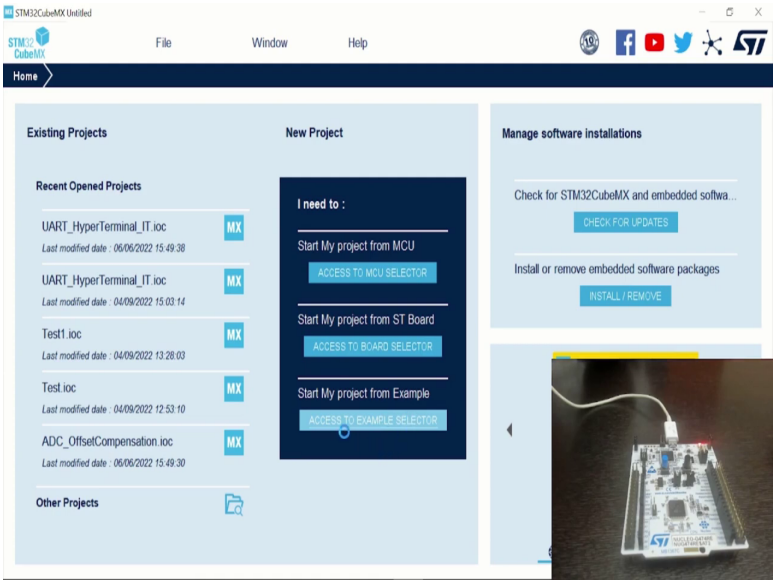
Welcome, all to the new lecture on Digital Control in Switch Mode Power Converters and FPGA-based Prototyping course. This lecture will be in continuation of the last lecture and it will again focus on STM32CubeMX.

(Refer to Slide Time: 00:43)




In the last lecture, we discussed the example of GPIO toggling in this lecture we will be starting with the example of UART communication and it will be demonstrated over the STM32 G4 Nucleo board. Then we will be covering the key peripherals initialization required for digital power applications mainly the ADCs and the timers. Then we will be focusing on the advanced setting where we will be covering the difference between the HAL layer and the lower level layer. And in the end, we will be focusing on the digital power workbench.

(Refer Slide Time: 01:16)



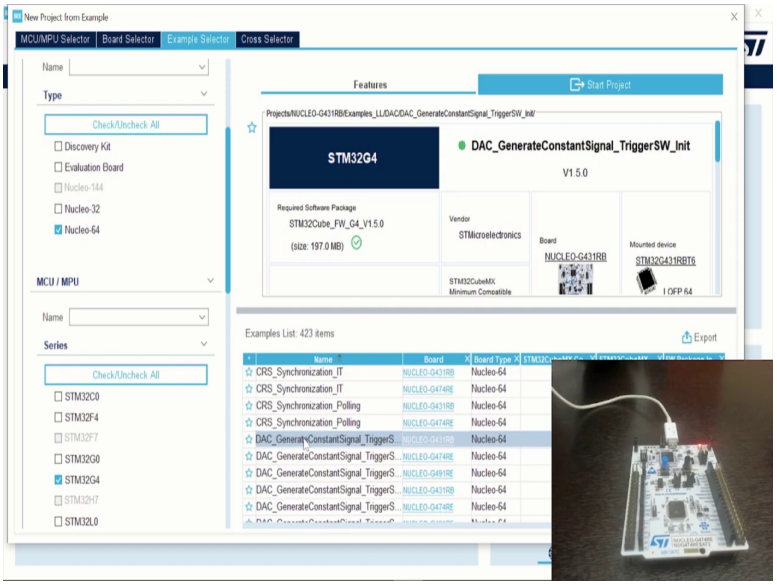

Let us start with the UART communication example here we are using a hyper terminal where the Nucleo board will be exchanging the data. We can directly go to the repository and open that particular example, but if we are not aware of the examples available we can go to the access to example selector.

(Refer Slide Time: 01:38)


The beauty of this option is here on a single screen we can see all the firmware examples available for that particular Nucleo board or the discovery board that we are having. So, we can select the STM32 G4 microcontroller. So, all these are the examples particularly for the

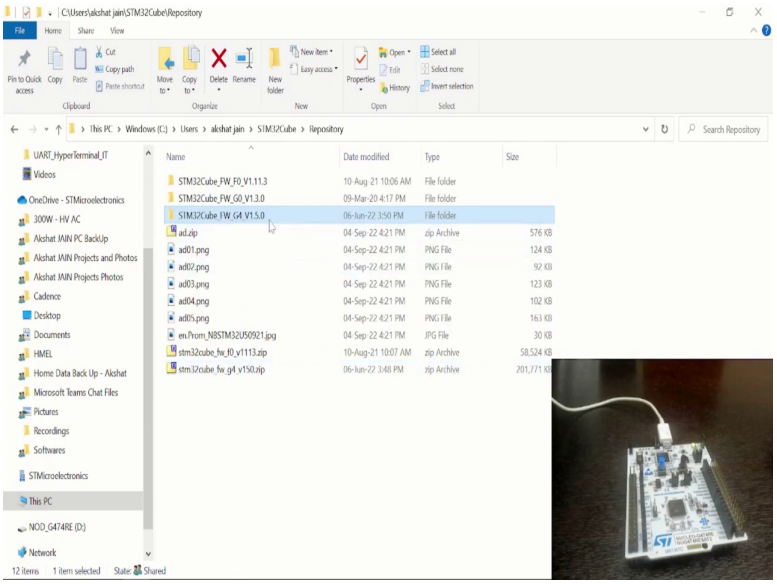
G4 microcontroller these are the ADCs comparator examples. So, we can select any of these examples depending on our requirements.

(Refer Slide Time: 02:07)



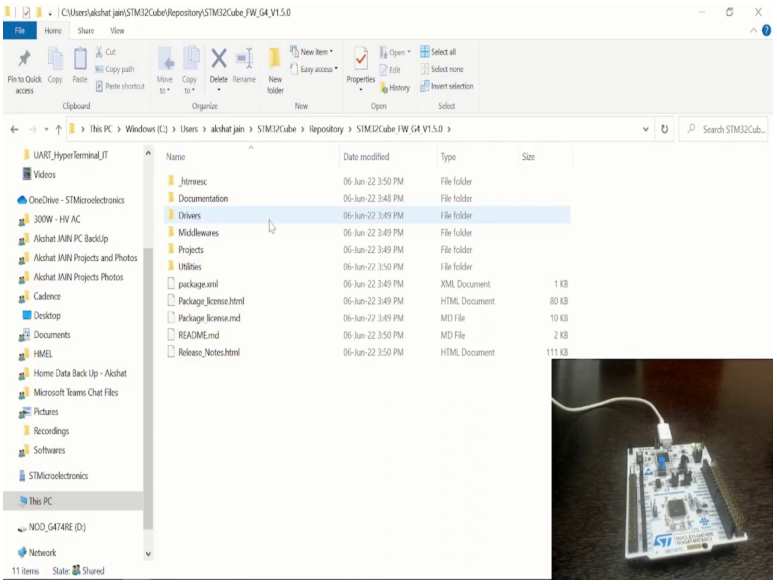

So, corresponding the ioc file will be open also the firmware project is already there. So, we need not generate the code again and again unless we require some specific change. So, let us go to the repository directly.

(Refer Slide Time: 02:23)



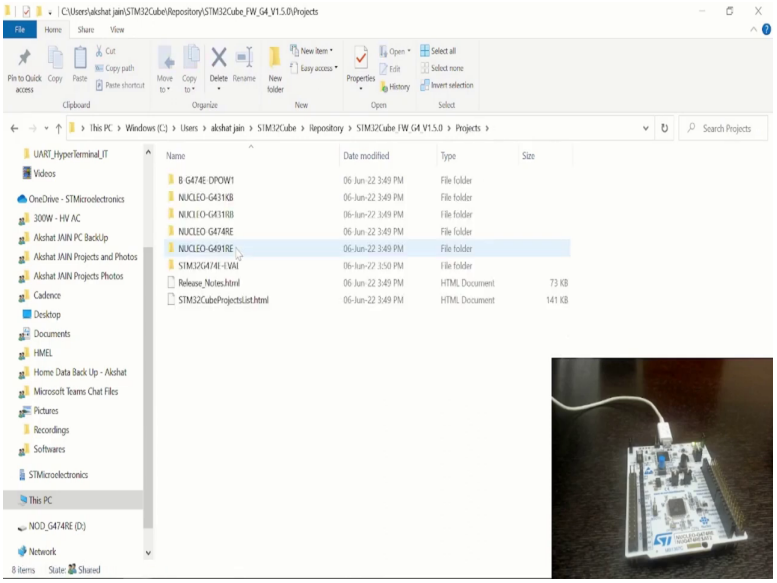

So, this is my repository I will be going into G4 version 1.5.0.

(Refer Slide Time: 02:29)



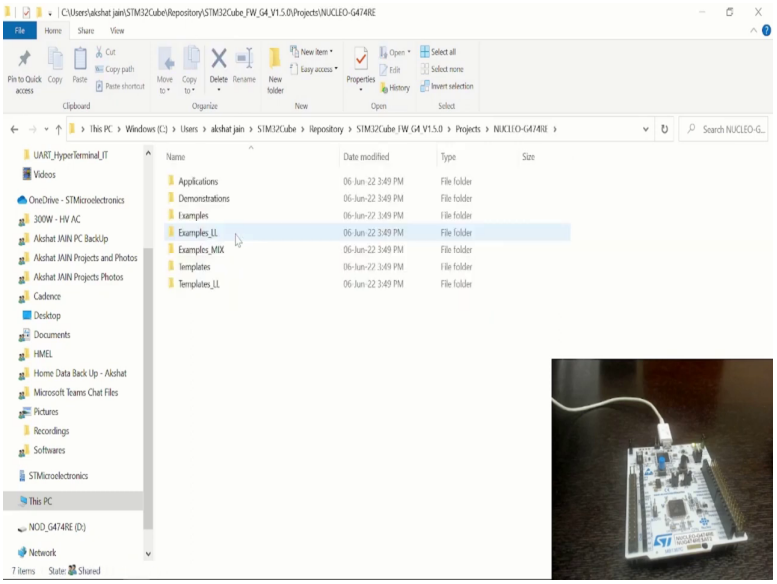
In this, I will be going on to the projects.

(Refer Slide Time: 02:32)

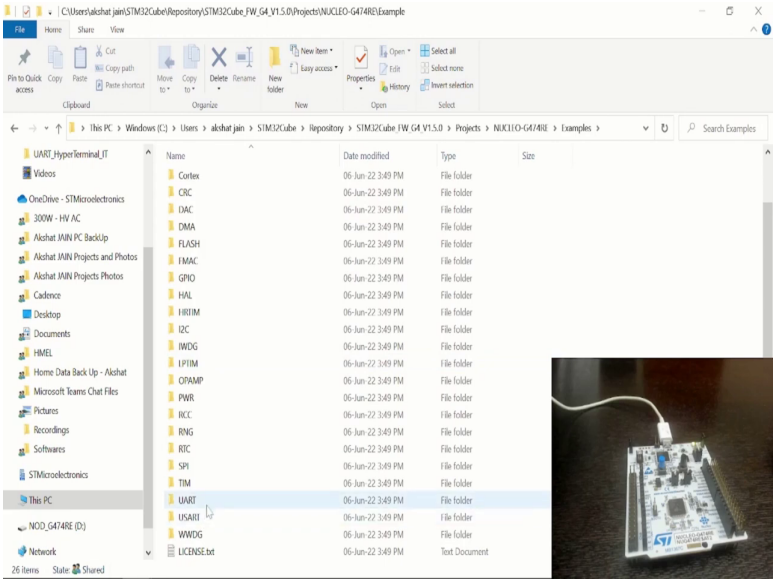


Then I am using NUCLEO G4 74RE86.

(Refer Slide Time: 02:36)

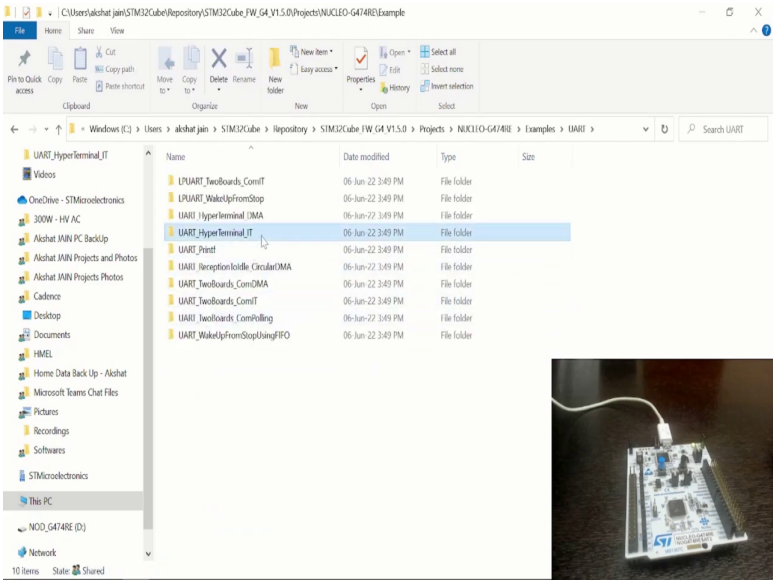



(Refer Slide Time: 02:39)



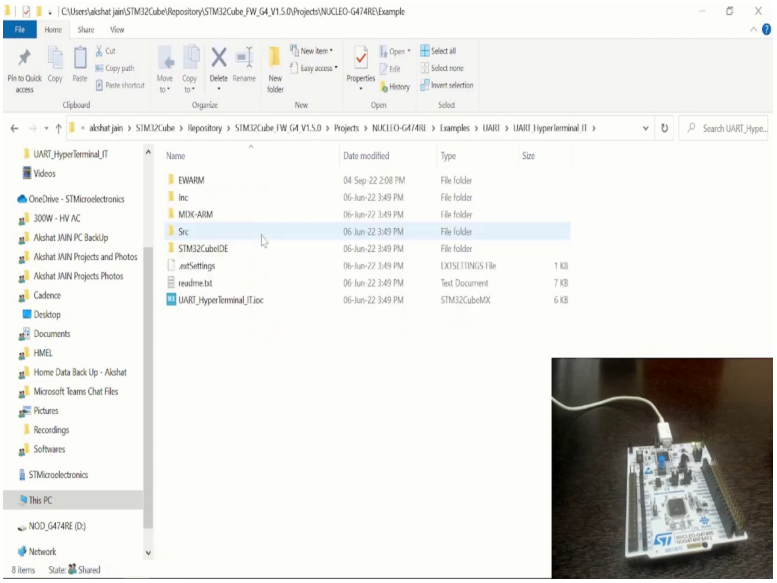

In the examples, I will be going to your peripheral.

(Refer Slide Time: 02:43)



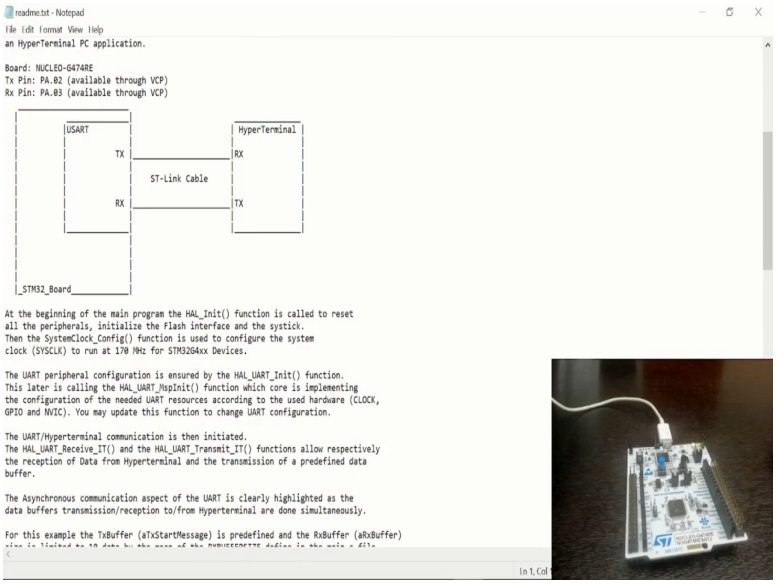

And within that, I will be selecting a hyper terminal interrupt example.

(Refer Slide Time: 02:46)



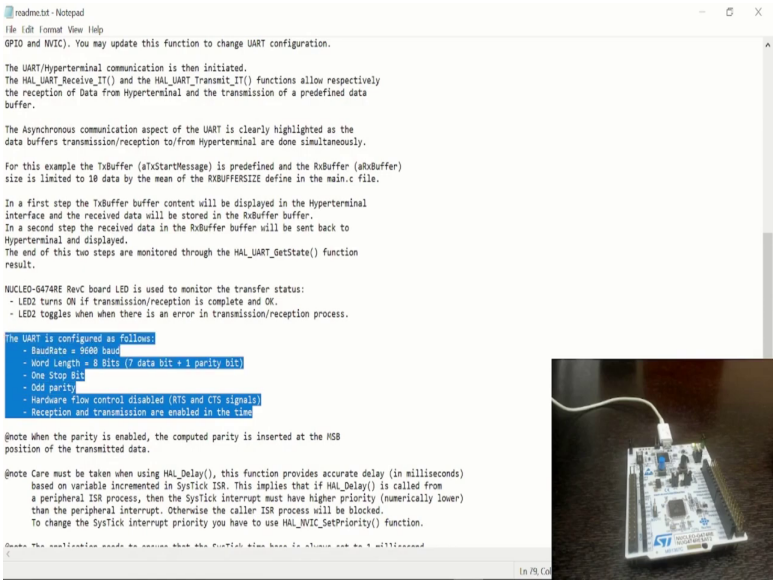

Before jumping on to the project directly I would request every user to go through the read me dot text file, before exploring any of the examples.

(Refer Slide Time: 02:58)



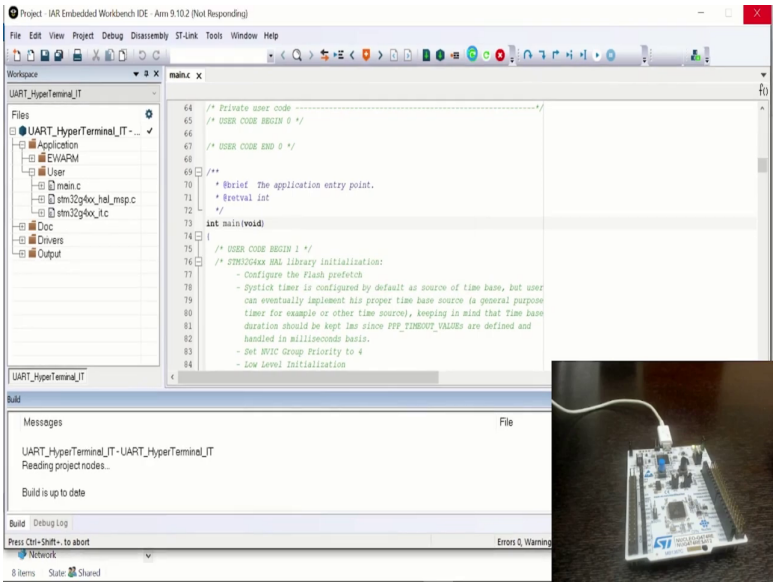

It tells what all pins are configured how exactly they are configured and also the basic details that what the end user can expect that basically, the results from that particular example.

(Refer Slide Time: 03:17)



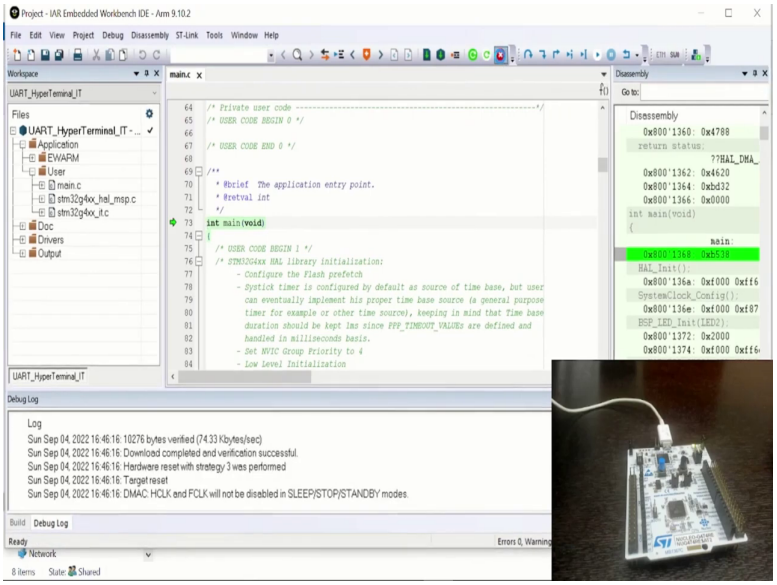

Also in some examples, some settings need to be done like in this the hyper terminal settings that we need to do is given here. So, it is beneficial to go through the read me dot text file for any of the examples used.

(Refer Slide Time: 03:37)

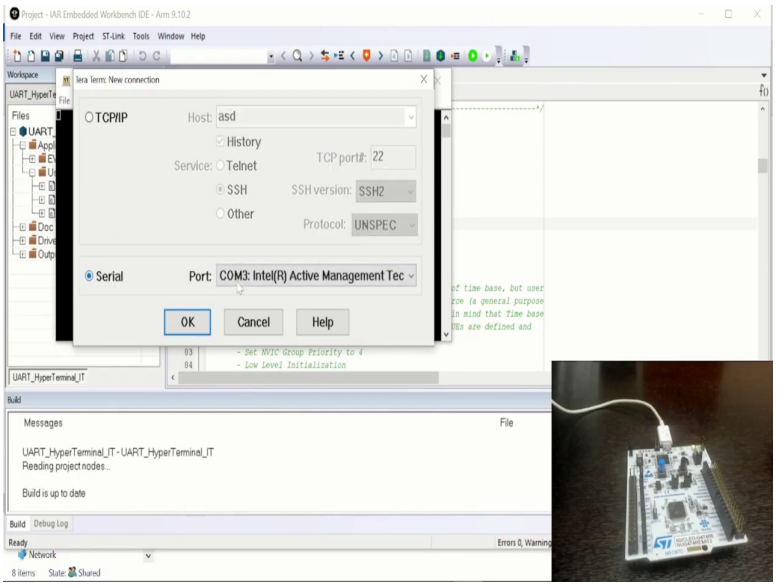


So, let me copy the path of the project and open this in IAR. So, I am fleshing this code into my Nucleo board, it will take a couple of seconds. So, the firmware is less. So, I am stopping the debugging here.

(Refer Slide Time: 04:16)

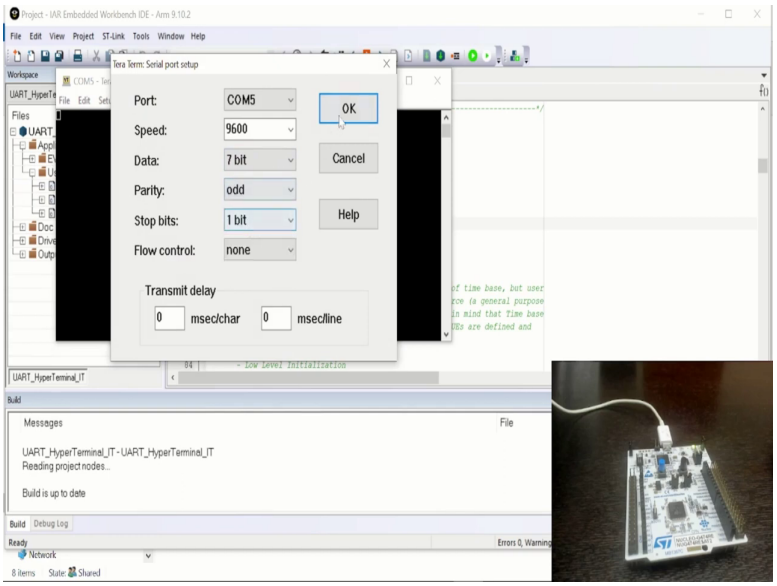


(Refer Slide Time: 04:33)

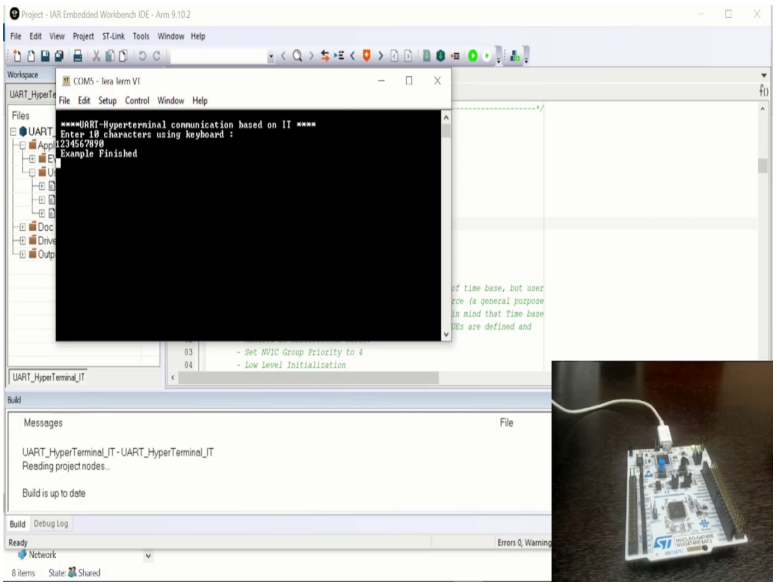



Now, for the hyper terminal, I am using the terraform tool and you can use any of the hyper terminal tools, but we just need to be aware of the settings that we need to do.

(Refer Slide Time: 04:45)



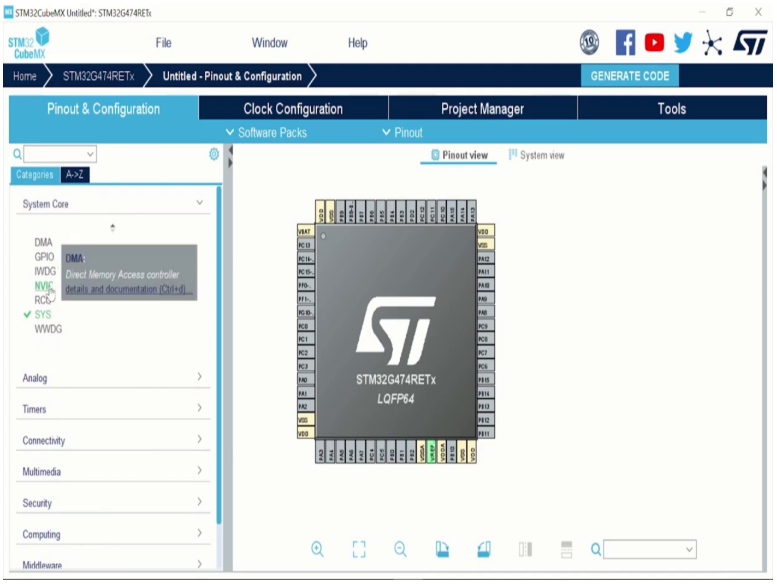
(Refer Slide Time: 04:54)




The settings are present in the read me dot text file and also in the main dot c file of your project, so I have done the setting. So, if you can see the Nucleo board I am just pressing the reset button of this and you can see my hyper terminal window it shows hyper terminal communication based on interrupt entering 10 characters using keyboards

So, I am typing just 1, 2, 3, 4, 5, 6, 7, 8, 9, and 0. So, as soon as I entered it says example finished and whatever data I have entered it has displayed. So, we have completed the hyper-terminal interrupt example. So, the idea behind of showcasing this particular example and similarly there are several other examples that we can use, and on top of it, we can build our application depending on the requirement or the configuration required. Now, we will be switching to the key peripherals that are required within digital power applications.

(Refer Slide Time: 05:55)

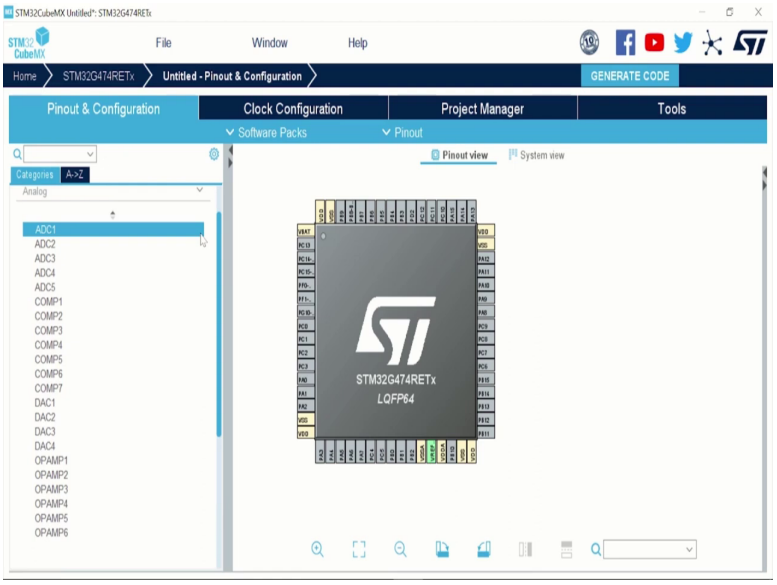



Digital power applications are very much timing critical several analog signals need to be sensed and they will be processed within the microcontroller and accordingly, the right signal or the right PWM needs to be generated to switch on or off our MOSFETs. Along with that in our hardware there are several comparator and op amps to have various comparison sensing with appropriate gain.

For a digital power application, we need to make sure that the MCU clock is fast enough and that the firmware or the code is written inside it should be highly optimized it can be done by using lower-level libraries. So, that every command can be executed in very less clock cycles.

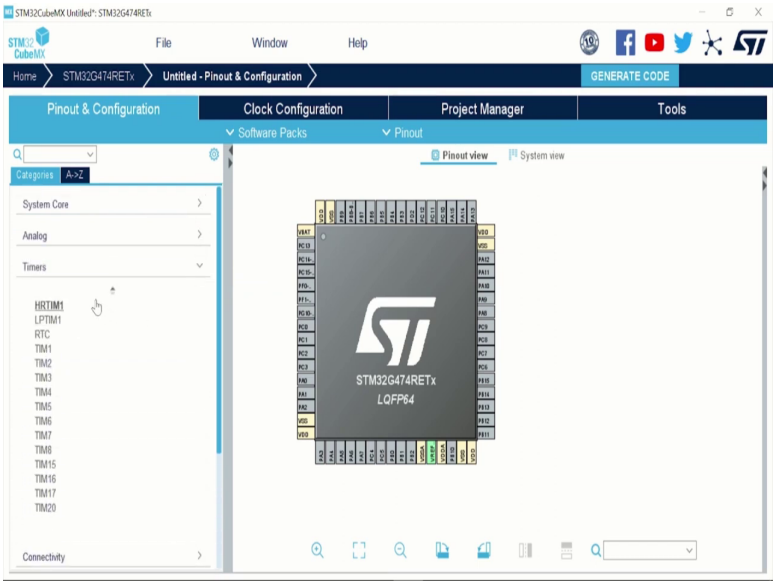
On the screen we can see the STM32 G4 microcontroller which contains all the peripherals required for any digital power applications, on the left side we see all the peripherals it includes; let us text one by one. In the system core, we have already discussed that it includes the GPIO settings, the interrupt settings, and the debugger settings.

(Refer Slide Time: 07:09)




The most important is the analog sections which include the ADC comparator DAC and the OPAMPs the advantages of having the OPAMPs and the comparator inside are not only in terms of cost but also because the space is optimized and we can adjust the various gains inside.

(Refer Slide Time: 07:25)

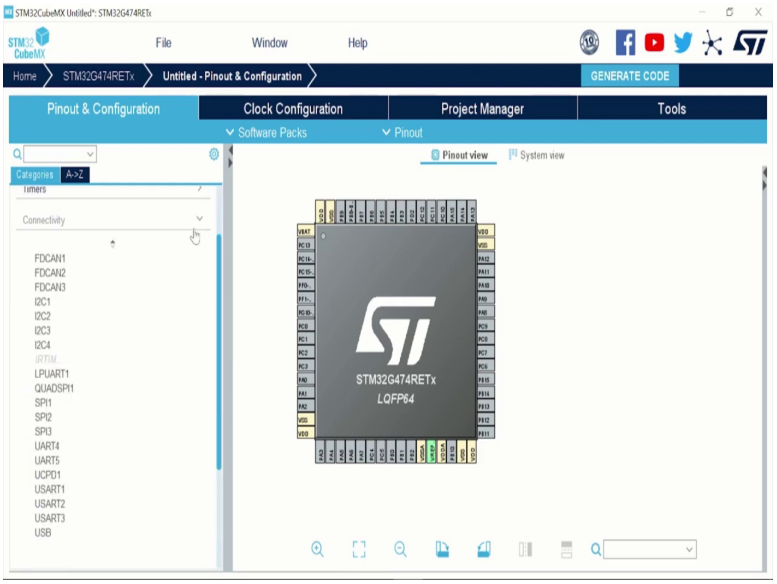



In the timer section, we have high-resolution timers, advanced timers, low-power timers, and several other timers. The high-resolution timer is having a resolution of 184 picoseconds,

where it can be configured and interlinked with several other peripherals and some external events to make it work as an independent state machine.

So, when we say statement that digital power application is very time critical. So, these high-resolution timers and their interlinking with external events are very much helpful.

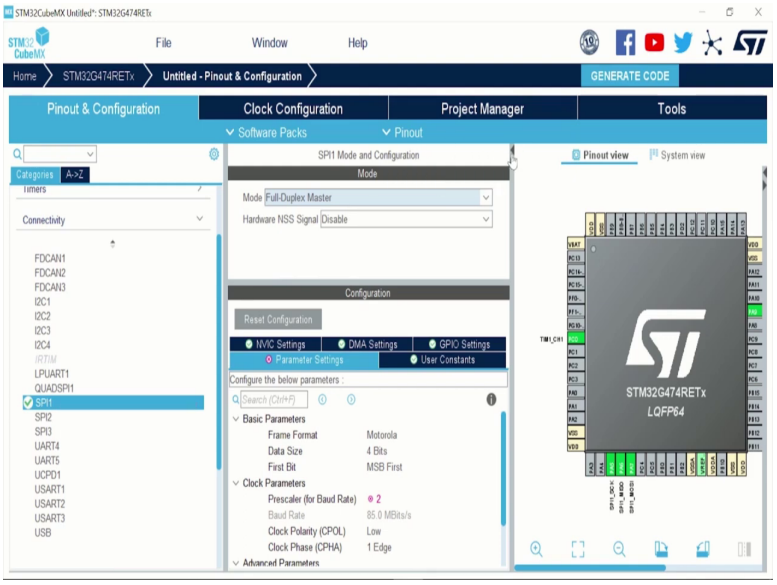
(Refer Slide Time: 08:00)




Apart from that there is the connectivity in which we have the UART SPI I 2C. So, these communication channels can be used as per the requirement, apart from that we can use the middleware computing security as per our user applications though they are not very much required in the digital power applications, but if required we can use it accordingly.

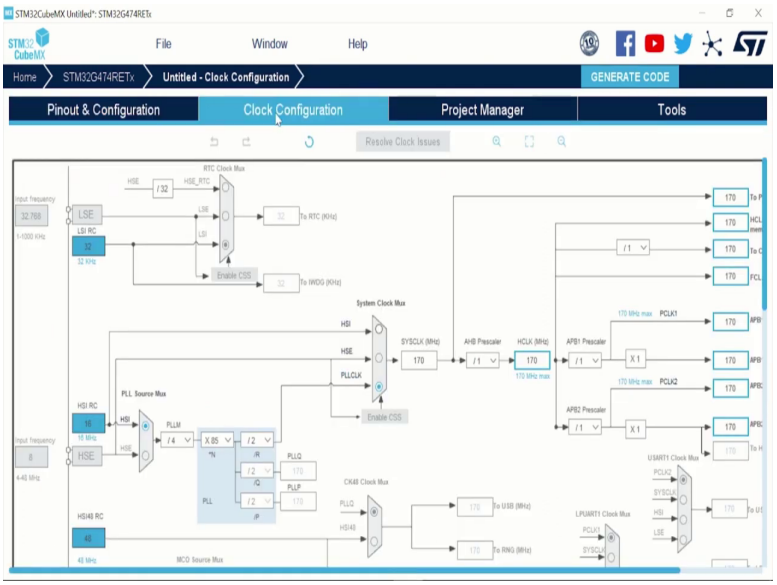
Time and again we highlight that digital power applications are very much time is critical. So, there are some advanced settings in which we can have the best optimizations in terms of code. So, let us go through those settings.

(Refer Slide Time: 08:38)



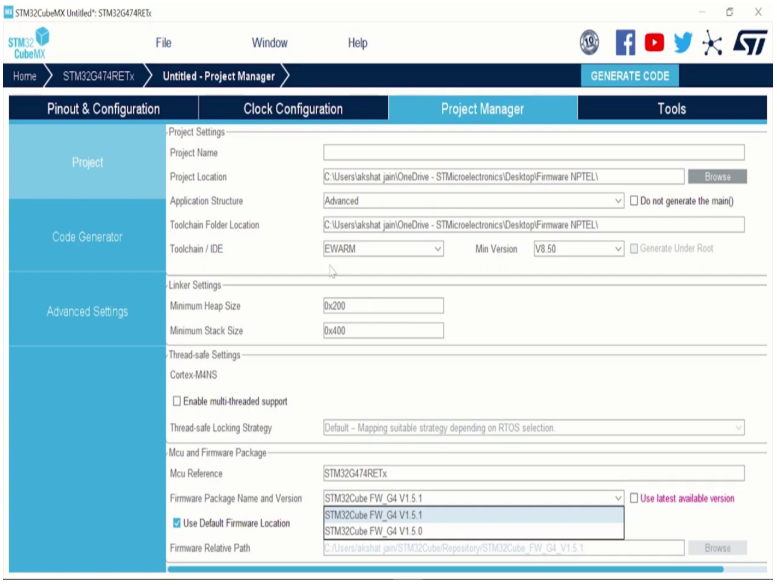

Before we enter into those settings let us configure a few of the pins with different peripherals to explain better. So, just let me configure one of the pins as the timer, and let us take one SPI. So now, we have three peripherals SPI GPIO, and the timer.

(Refer Slide Time: 09:08)



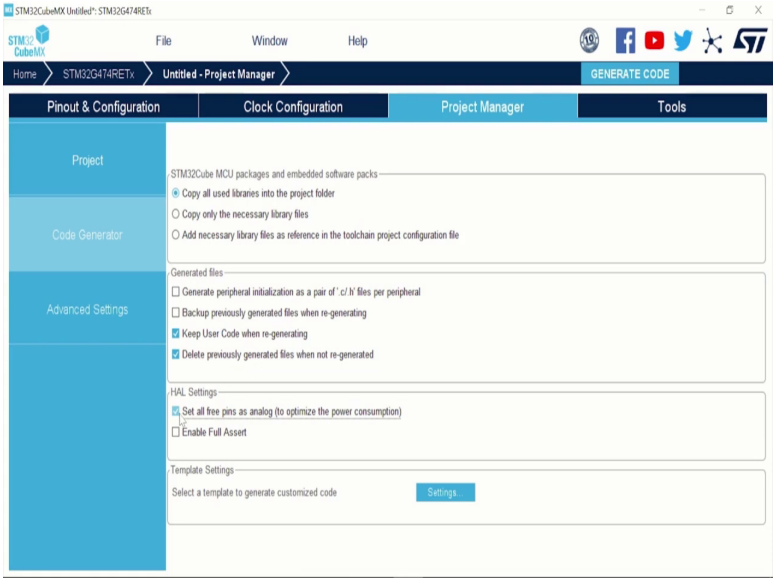

So, on the dark loop, we are on the pin and configuration page. So, this is the second one is the clock and configuration which we have already discussed in the last section.

(Refer Slide Time: 09:15)




The third is the project manager option. So, by default, we see this page where we can enter the project name location, and other things, and in terms of the firmware package that we want to use the package we have already downloaded, we will be able to see here. But just in case we always want to use the latest version, so we can check this option. Also, the stack size that can if you want to change can be updated here, normally it takes a default setting of the microcontroller that we have selected.

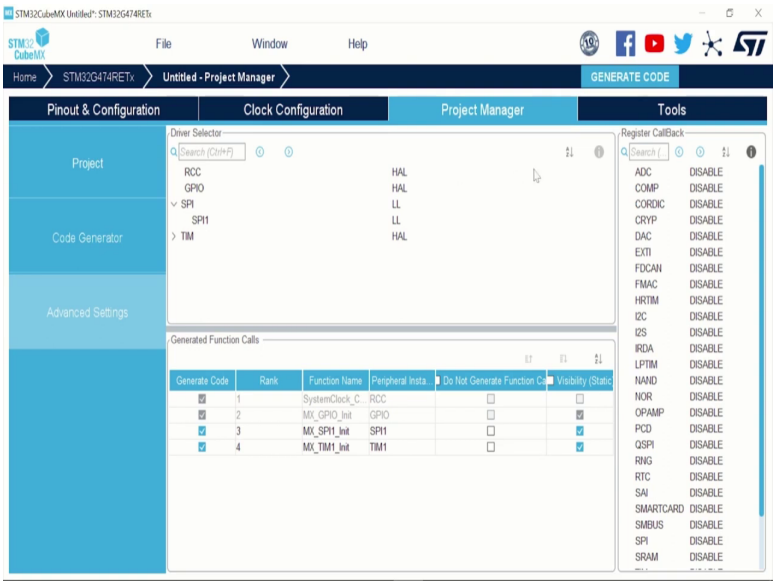
(Refer Slide Time: 09:48)



The second option is the code generator. So, in this, there are again several options depending on the user's requirement by default all the library files have been added to the project folder, but in case there is some space constraint we can use the copy only the necessary library files, and similarly, other options can be chosen.

The third option is very important, so if we select this option set all pins all free pins analog. So, all the unused pins will be configured as analog if we select this option. So, this will be optimized in terms of power.

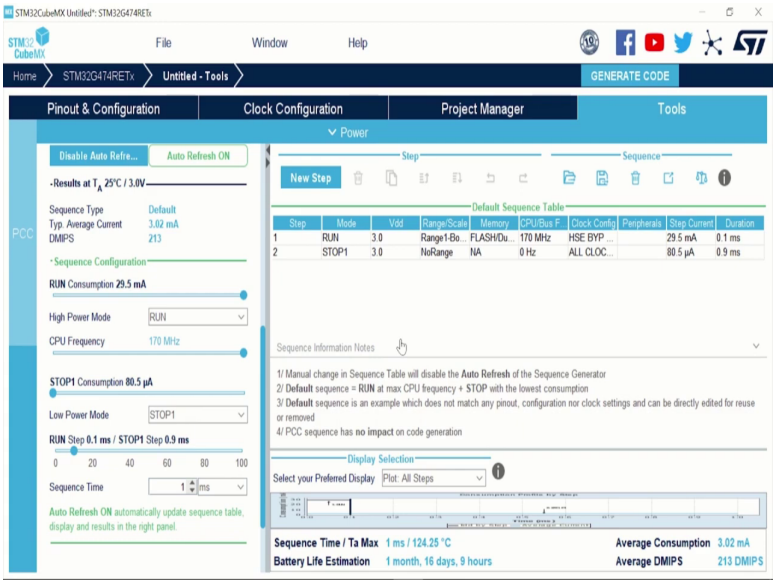
(Refer Slide Time: 10:28)




While if you go on to the advanced setting this is again a very important section in this there are options for selecting the drivers. So, let us take an example of SPI by default hardware abstraction layer is selected as a driver, but if I want to use the lower-level libraries. So, the lower libraries are much more optimized we can directly write the registers it takes fewer clock cycles of the microcontroller to execute the commands.

So, this is very important for digital power applications. I would suggest we can start with the HAL layer to have a better understanding then as we have more understanding of the STM Cube MX store and more understanding of the firmware. Then we can select the lower-level libraries and optimize the code and accordingly we can see the difference.

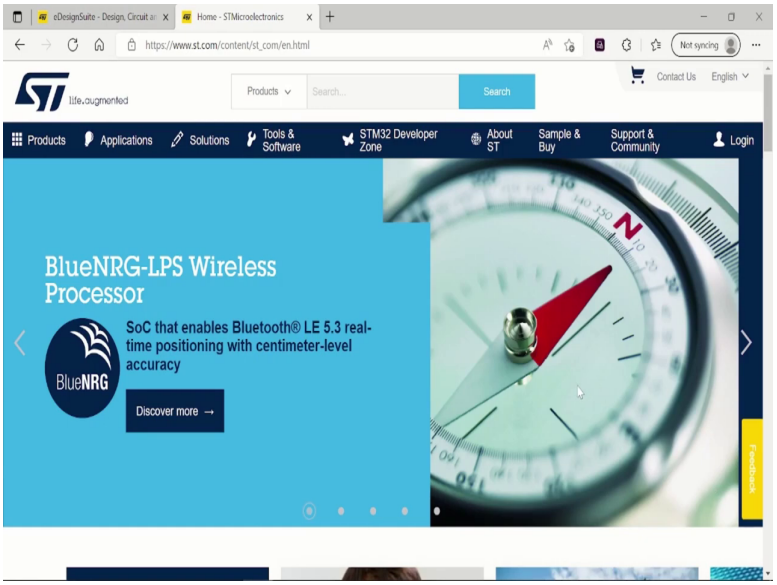
(Refer Slide Time: 11:23)




The last section is the tools here we can see the power consumption and other things as we have not configured all the peripherals, but if you can configure them as per the requirement we can see what sort of power consumption and the stop consumption different modes, what sort of the microcontroller behavior will be there.

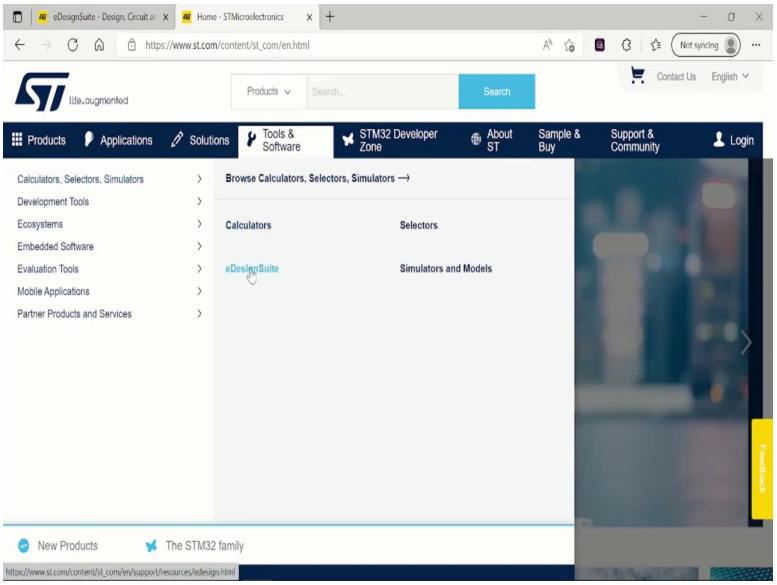
So, we can explore this, but we have to make sure that all the pins of the microcontrollers are configured as per our requirements. Then this tool page is very much important for us.

(Refer Slide Time: 12:01)



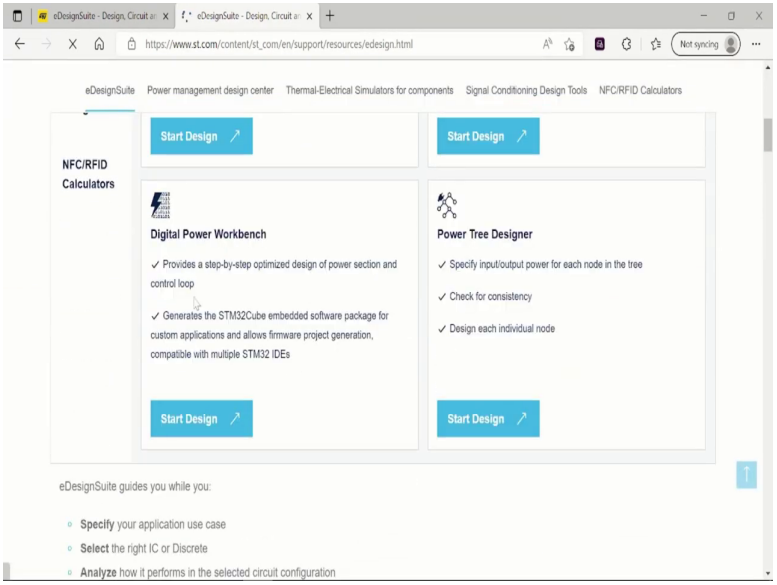
Based on the STM32CubeMX tool there is another power powerful tool from ST microelectronics which is a digital power workbench.

(Refer Slide Time: 12:12)



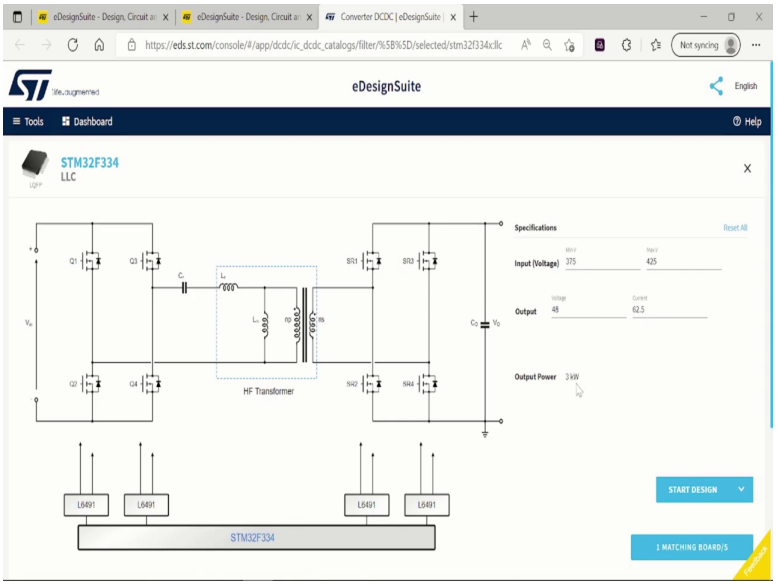

So, I am on the st website, if I go to the tools and software section and within the first section I could see the eDesign Suite.

(Refer Slide Time: 12:21)



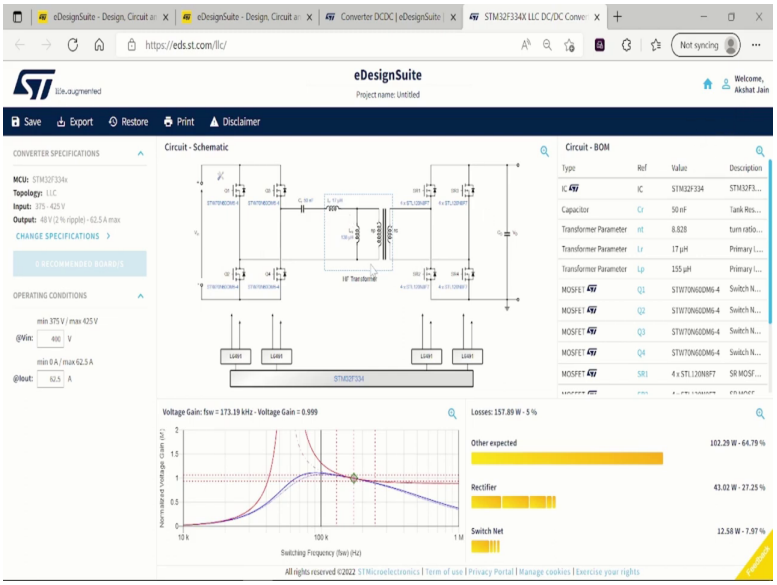

In eDesign suit, if you scroll down we can see the digital power workbench.

(Refer Slide Time: 12:29)



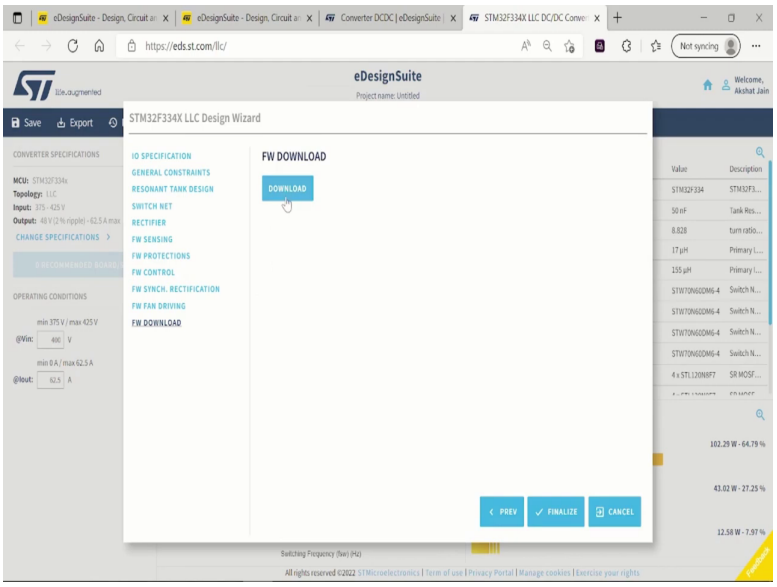

We can click on star design this is a very much new tool from st microelectronics right, now we have the full bridge LLC converter with synchronous rectification; it is based on STM32F334. More and more topologies and microcontrollers are being added. So, without changing any specifications it is a 3-kilowatt configuration I just click on start design.

(Refer Slide Time: 12:58)



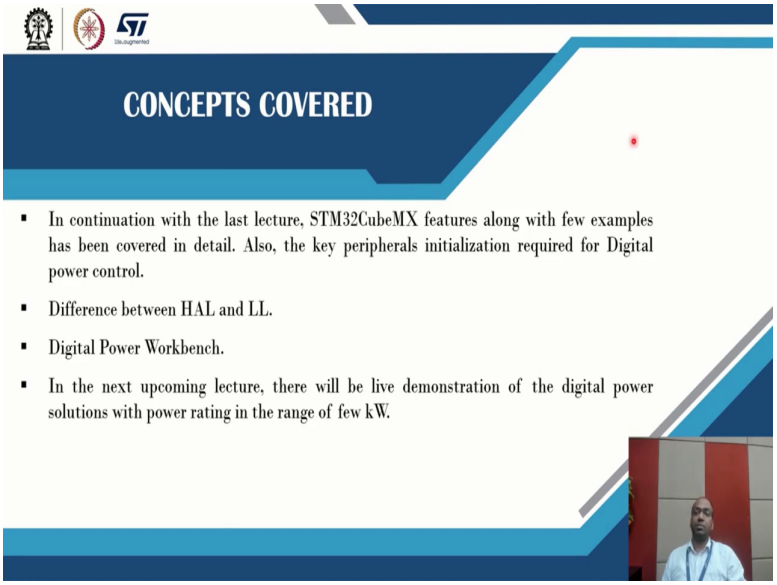

So, it has already taken my login, but if you want to do the login and make some changes in the configuration settings or you want to update the power rating, or some other MOSFETs you want to explore you can just do the login. And the most important thing is when we go to the specifications the last option is the firmware download.

(Refer Slide Time: 13:16)




So, here we can download the firmware. So, it will be downloading your complete firmware package along with the STM Cube MX file, there also you can do certain changes and make your application work.

(Refer Slide Time: 13:37)




Concepts covered in the last lecture and the ongoing lecture we have covered the features of the STM Cube MX software tool along with some examples, also we have discussed key peripherals required for the digital power application.

Then we discussed the advanced settings particularly the HAL layer and the lower layer differences. Then we also touched on the digital power workbench particularly for the LLC application while in the upcoming lecture, there will be live demonstrations of digital power solutions with a power rating in the range of a few kilowatts.

Thank you see you in the next lecture.