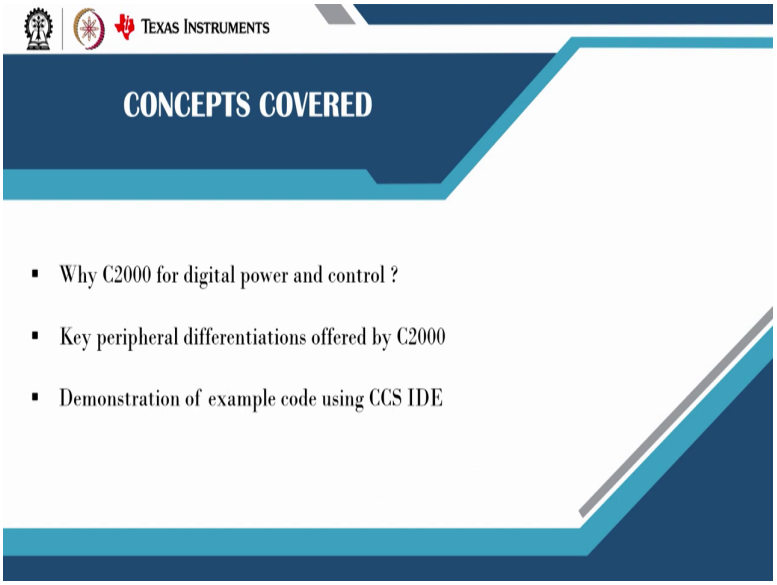
Digital Control in Switched Mode Power Converters and FPGA-based Prototyping

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

Module - 09
Digital Control Implementation using Microcontrollers
Lecture - 88
Texas Instruments C2000 key peripheral differentiations

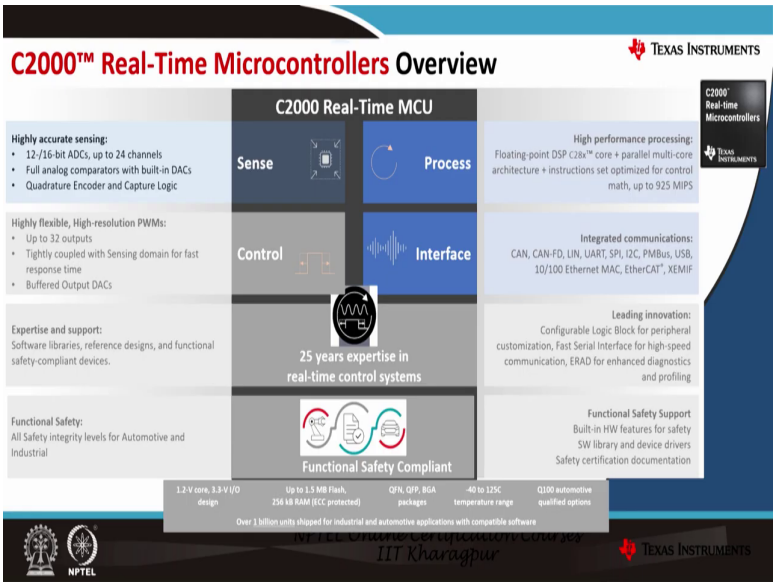
Hello and welcome to the NPTEL online certification course a course on Digital Control in SMPCs and FPGA-based prototyping. I am Aditya Dholakia from Texas instruments and module 9 Digital Control Implementation using Microcontrollers. Today in lecture 88 we are going to study Texas Instruments C2000 key peripheral differentiations.

(Refer Slide Time: 00:45)



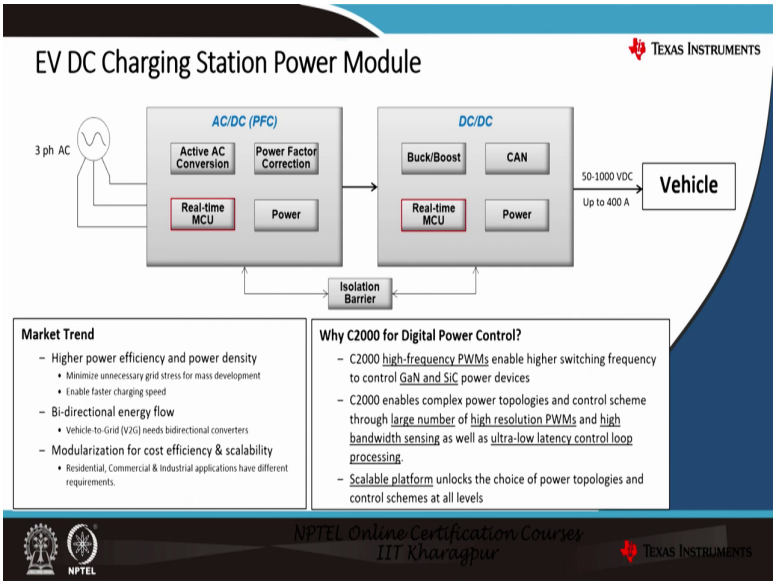
So, the concepts covered will be, why do we need a C2000 for digital power as well as a control? What are the key peripheral differentiations offered by the C2000 device? And lastly, we will give a brief overview of what is the demonstration of example code using CSS IDE.

(Refer Slide Time: 01:05)



So, in the previous lecture, we went through the processing part of the processing capabilities of a C2000 real-time microcontroller. In this lecture, we will go through the sensing as well as the control part.

(Refer Slide Time: 01:17)



Starting with one of the applications which is an electric vehicle DC charging station power module. What it looks like is something like this where a 3-phase AC grid voltage is connected to an AC DC which is the power factor character unit that ultimately feeds its voltage into the DC-DC unit. Lastly, there is a vehicle that gets charged because of it.

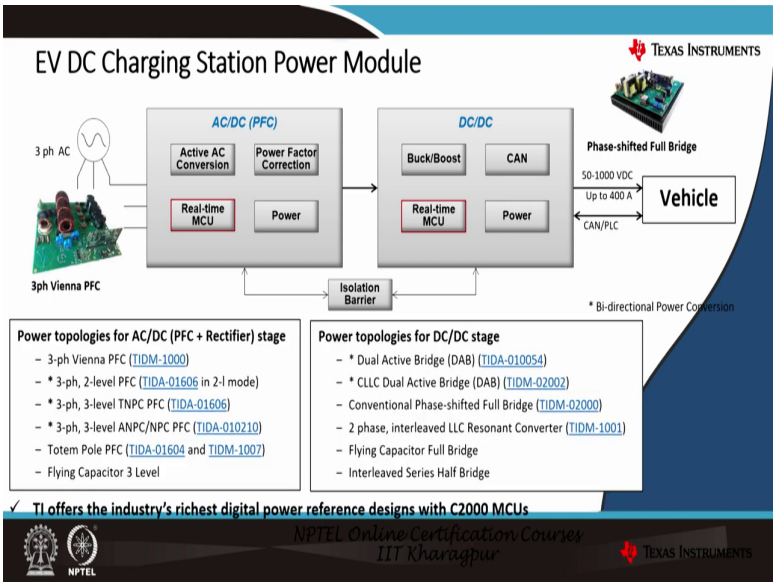
There is an isolation barrier in between as well now two real-time MCUs are coming into the picture over here. One is in the AC DC stage, the second is in the DC-DC stage. The current market trend is highlighted in the bottom picture. So, the first one that you see here is the higher power efficiency and power density. This is needed because it will minimize the unnecessary grid stress for mass development as well as if you have a higher power density it will enable fast charging speed if you have the same size.

Moreover, power efficiency will reduce losses as well. Additionally, you need to have bi-directional energy flow back for the vehicle to grid charging. Lastly, we have modularization for cost efficiency and scalability. That is residential, commercial, and industrial applications will have their different requirement. So, that is why you need modularization. Now why C2000 for digital power control? So, the first point is that for higher power efficiency and power density, we have high-frequency PWMs which enable high switching frequency to control the gallium nitride as well as the silicon power devices.

C2000 also enables complex power topologies and control schemes through a large number of high-resolution PWMs and high bandwidth sensing which enables ultra-low latency control loop processing. That is a very fast IDC as well as fast processing which ultimately will give you an ultra-low latency control loop.

The platform is also scalable if you have an older version of the device and if you want to port it to a newer one or if you want to play around with the device the platform is scalable where you can port the code from one device to the other with minimal or no changes. This unlocks the choice of power topologies as well as control schemes at all levels.

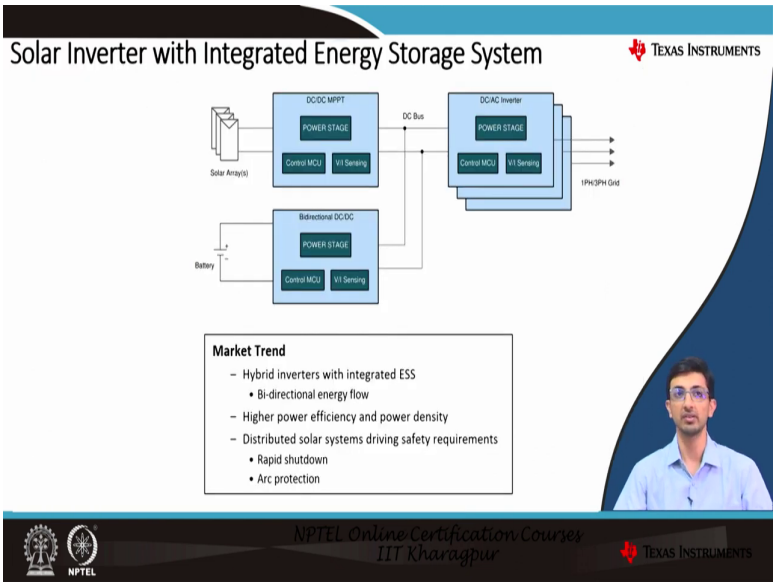
(Refer Slide Time: 03:23)



So, just to give a brief overview of what all DI designs are available for the power topologies for AC DC as well as the power topologies for the DC-DC stage here is a picture. So, in the AC DC that is the PFC plus rectifiers stage we have 3 phase Vienna rectifiers PFC we have 3 phase 2-level PFC there is a TNPC PFC stage an ANPC or NPC PFC stage, a totem pole PFC stage which we will discuss in the coming design as well and a flying capacitor 3 level.

In the DC-DC power topologies, we have a dual active bridge, a CLLC dual active bridge a conventional PSFB that is a conventional phase shifted full bridge a 2 phase interleaved LLC resonant converter, a flying capacitor full bridge, and an interleaved series half bridge. These designs are provided free of cost on TI dot com. You can quickly check out and jump-start your design development. The software is also available on DI com. Specifically in the digital power SDK as well as in the motor controllers SDK.

(Refer Slide Time: 04:19)



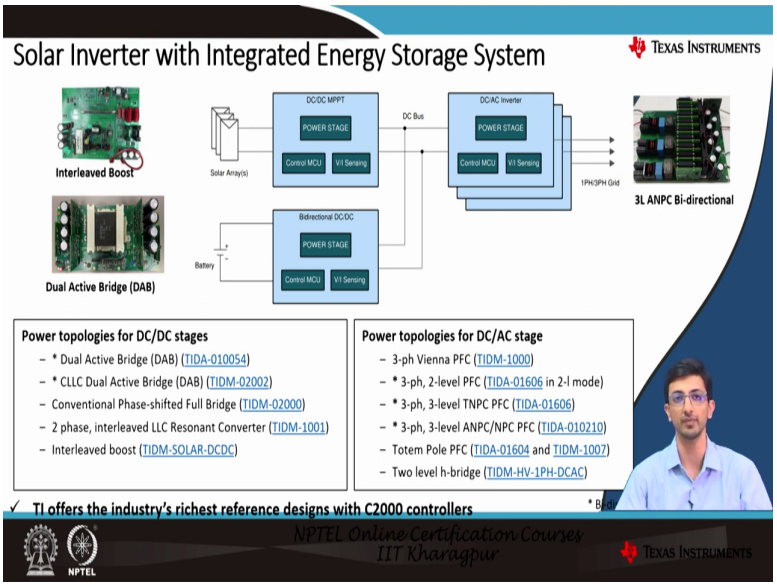
Next, the second application is the solar inverter with an integrated energy storage system. So, in this application, the solar arrays are connected to the DC-DC stage with the maximum power point tracking that is the MPPT which ultimately feeds into the DC AC inverter stage. Also, we require an integrated energy storage system along with this. So, we have a battery pack and for that, we have a bi-directional DC-DC stage connected to the DC bus.

So, the market trend here looks something like this where a hybrid inverter is there with integrated ESS which is an energy storage system. This also requires a bi-directional energy flow. Same way as compared to the previous example this also demands a high power efficiency and power density. Also, the distributed solar systems that are our rooftop solar systems drive safety requirements as well.

So, the primary safety requirement is that if you have let us suppose a fault condition. So, you need an immediate rapid shutdown. And in such scenarios where there is a chance of physical damage to the system and there can be an arc introduced, we have a quick requirement for arc detection as well as quenching.

So, arc protection is also a critical feature highlighted in the market trend. These all market trends are addressable by the C2000 devices.

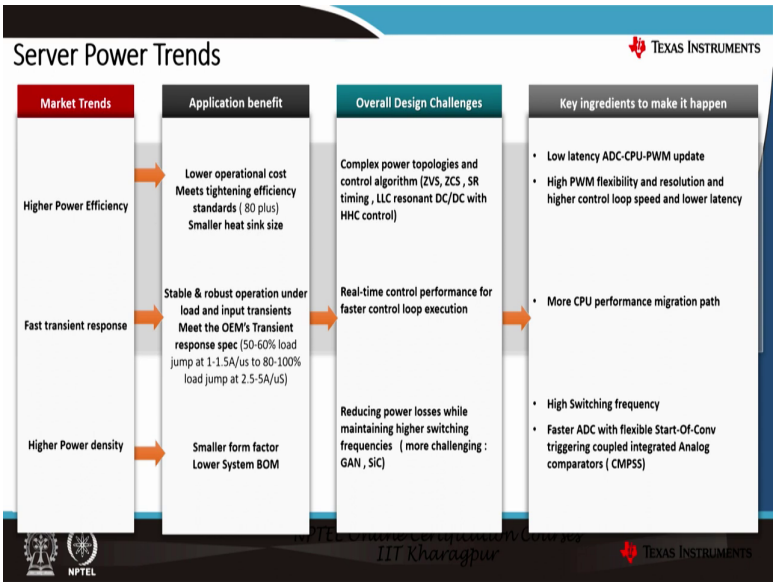
(Refer Slide Time: 05:34)



So, coming back again to the device topologies that are provided for a DC-DC stage as well as the DC AC stage this will look pretty much the same thing because most of our topologies involve a bi-directional power flow. So, the DC-DC stage I have highlighted earlier as well the dual active bridge the CLLC dual active bridge, and so on.

And the same way for the DC AC stage we have a 3-phase Vienna PFC 3 phase, 2 level PFC, a TNPC stage, an ANPC stage, a totem pole PFC and a 2-level edge h bridge. TI also offers the industry's richest reference designs with C2000 controllers. All these reference designs are available with the C2000 digital power SDK which you can quickly take a look at.

(Refer Slide Time: 06:14)



Now moving on to the server power trends. So, the market trends again over here demand higher power efficiency, fast transient response, and higher power density as well. And higher power efficiency the application benefits look something like this. Like it has lower operation cost it specifically meets tightening efficiency standards which you need to have more than 80 and since your efficiency will be higher you will have a lower power loss and that is why you will need a smaller heat sink.

And the fast transient response. So, for that, you need the application benefit will look like it will have a stable and robust operation under load as well as an input transient same way as compared to the higher power efficiency part this also will meet the OEM's spec. And lastly, the higher power density where we have a smaller form factor available with a lower system BOM.

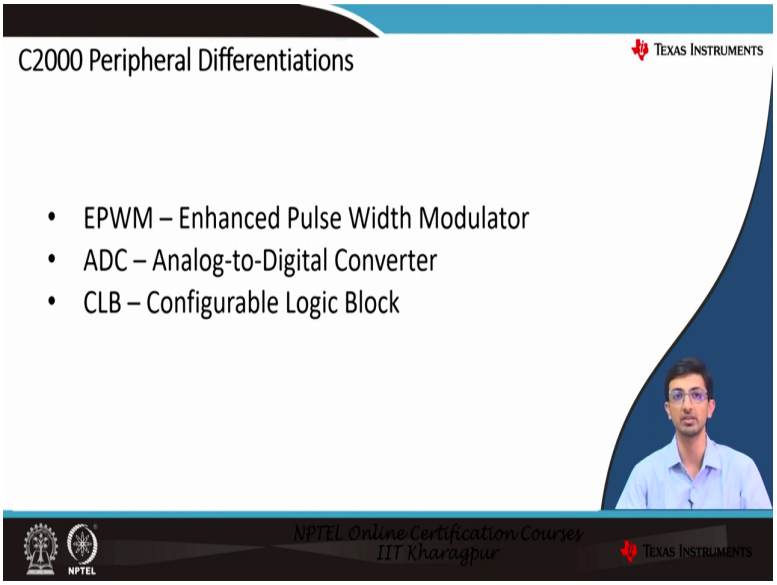
What are the overall design challenges if we talk about the market trends? So, the first design challenge that comes up is the complex power topology along with the control algorithm that is like we want to incorporate ZVS ZCS etcetera. So, for that what are the key ingredients to make it happen within the C2000? So, we have the low latency ADC to CPU to PWM update. What I mean by this is that the sensing part is taken care of by the ADC then the processing will be taken care of inside the CPU whereas, the PWM will be updated.

So, this whole process is taken care of at a very fast speed which ultimately results in very low latency. Then there is high PWM flexibility resolution and higher control loop speed and

lower latency which enables the overall design challenge of complex power topology easily achievable. The second design challenge is the real-time control performance for faster control loop execution. So, for that, we have coprocessors along with the CPU which frees up the CPU bandwidth such as CLA, TMU, FPU etcetera.

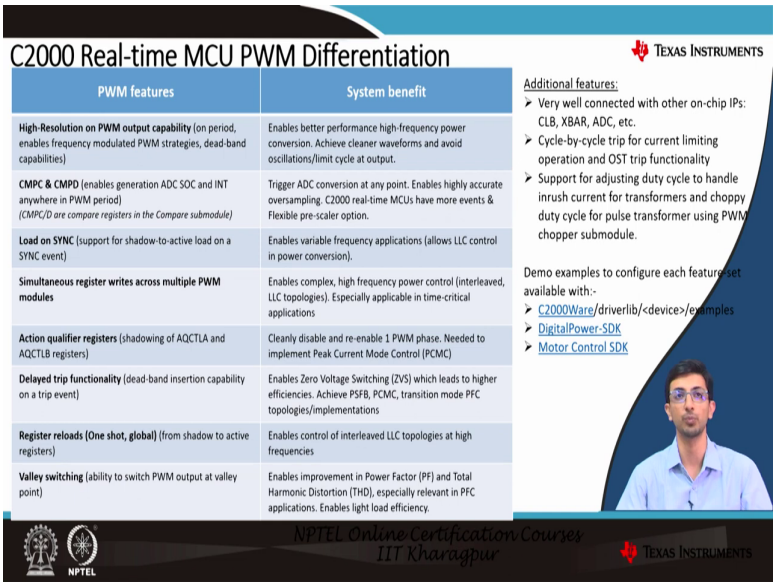
And that is why the CPU will be able to give better performance. Lastly we need to achieve higher switching frequency specifically for GAN and silicon carbide devices. So, the C2000 device can switch at a very high frequency and also it has a faster ADC with the flexible start of conversion triggering that could be coupled with an integrated internal analog comparator. That is an integrated internal CMPSS module.

(Refer Slide Time: 08:25)



So, now moving on to some of the key peripheral differentiations that C2000 offers. I would not be touching upon all the peripherals in detail, but specifically talking about the sensing as well as the processing part sensing as well as the control part. We have the EPWM, ADC as well as CLB. That is the enhanced pulse width modulator module the analog to digital converter as well as the configurable logic block.

(Refer Slide Time: 08:48)



So, starting with the ePWM module which is the enhanced pulse width modulator module. We have the first on the top left that you see the high resolution on the PWM output capability. This is a high-resolution feature over the conventional PWM which adds specific use cases when you are operating at a very higher frequency of more than 1000 kilowatt and so on.

These specific details about when and where their high resolution can be utilized can be checked out in the specific device data sheet. This enables better performance for high-frequency power conversions, achieves cleaner waveforms, and avoids limit cycle oscillations at the output. Then the second PWM feature that I would like to talk about is the comparator C and the comparator D module

So, these CMPC and CMPD are some comparators that enable the generation of ADC SOC event that is the start of conversion event as well as the interrupt generation event anywhere in the PWM period. The third point is the load on the sync event. So, the load on sync is basically that a CPU generates a sync event whenever a sync event is generated we have support for the shadow to active load on a synchronous event.

So, you have simultaneous register writes across multiple PWM modules. So, this particular application can be utilized in variable frequency applications such as LLC control and LLC control topologies etcetera. Then there is the action qualifier register where we have shadowing as well. So, the action qualifier register is capable of cleanly disabling and

re-enable one PWM phase specifically needed to implement peak current mode control applications. Then there is delayed trip functionality.

So, the delayed trip functionality is basically that you can delay the trip event when needed. So, this enables the ZVS operation which leads to higher efficiencies. The applications include the PSFB, PCMC that is the peak current mode control, then transition mode PFC topologies etcetera.

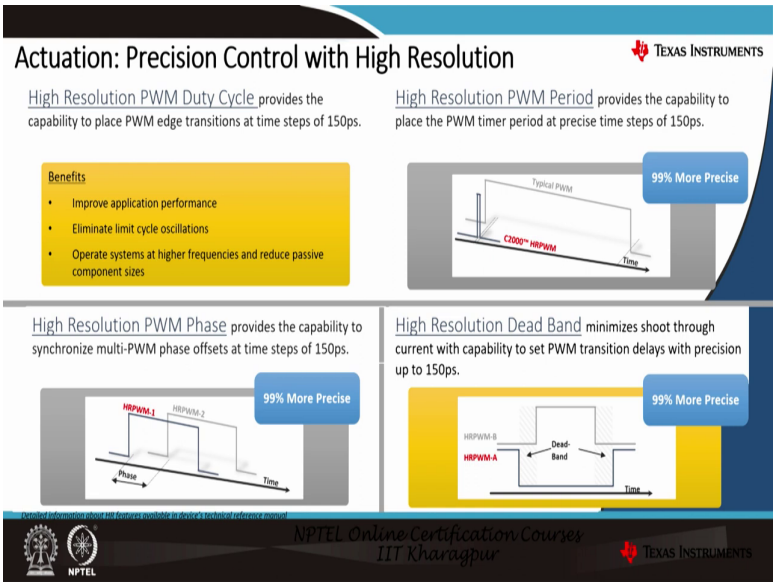
The register reloads are also possible if you want to only load registers from shadow to act in a single PWM module you can use the one-shot register reload. And if you want to operate at a global level that is an all-PWM module that is also possible using the register lead load option at a global level.

This enables control of interleaved LLC topologies at higher frequencies. Lastly, that is a valley switching feature that can switch the PWM output exactly at the valley point or late as desired by the user. So, this again enables improvement in the power factor, and the total harmonic distortion, and especially this is relevant in the power factor character applications. Specifically at the light load efficiency conditions.

Beyond these features, there are some additional features as well. Since PWM is a very critical IP it is very well connected with other on-chip IPs such as the CLB that is the configurable logic block, the cross bars, ADC etcetera. There are also features like a cycle-by-cycle trip for current limiting operation and one short trip functionality for fault detection purposes. Then the support for adjusting the duty cycle to handle rush current for transformers and choppy duty cycle for pulse transformers using the PWM chopper submodule.

All these features that are discussed for that the demo examples are available to configure each feature set available that is in C2000 where with driver lib device and examples folder moreover you can check out digital power SDK as well as motor control SDK.

(Refer Slide Time: 12:01)

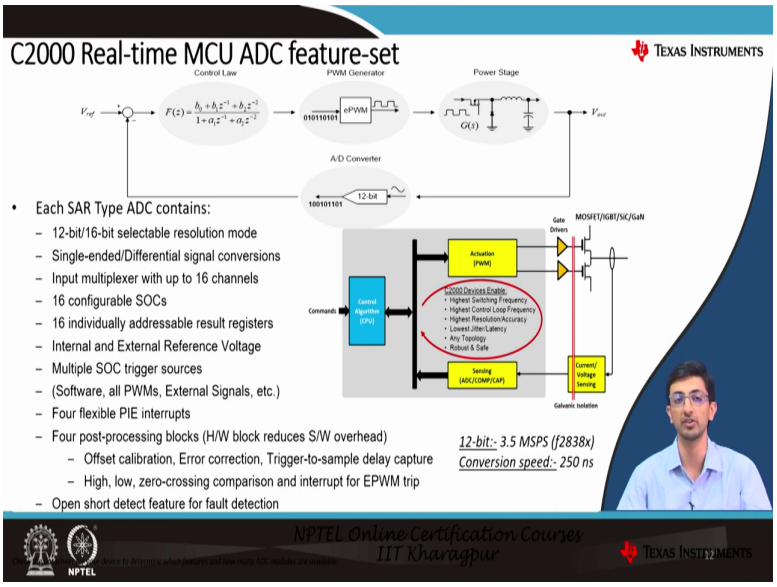


Now, specifically talking about the high-resolution modules. So, high resolution is a module that enables you to improve application performance, eliminate limit cycle oscillations and it operates the system at a very high frequency, and reduce the passive component sizes.

It can provide the to place the PWM h transition at a time step of as minute as 150 picoseconds. Now typically this use case is used at the PWM duty cycle. So, if you want to have a very specific duty cycle that is where hr PWM can come in place.

If you want to have a very specific PWM period or a PWM phase where you are operating at a multi-phase multi-PWM phase level or you want a very specific dead band that will be ultimately used in minimizing shoot through that is when the high-resolution feature will be used. Just take care that this is this resolution is provided up to as much as low as 150 picoseconds.

(Refer Slide Time: 12:56)



Then we come down to the ADC which is the analog-digital comparator. So, TI C2000 ADC is a SAR type that is a successive approximation type ADC. There are 12 to 16-bit selectable resolutions, a single-ended or differential-ended differential signal conversion single-ended is compatible with both 12-bit as well as 16-bit resolutions whereas, the differential signal conversion is compatible only with the 16-bit resolution mode.

We have an input multiplexer with up to 16 channels per ADC module. And for every 16 channels, we have 16 configurable SOCs as well. Again for every 16 SOCs, we have configurable addressable registers as well. There is an option to select whether you want to go for an internal reference voltage or you want to use an external reference voltage.

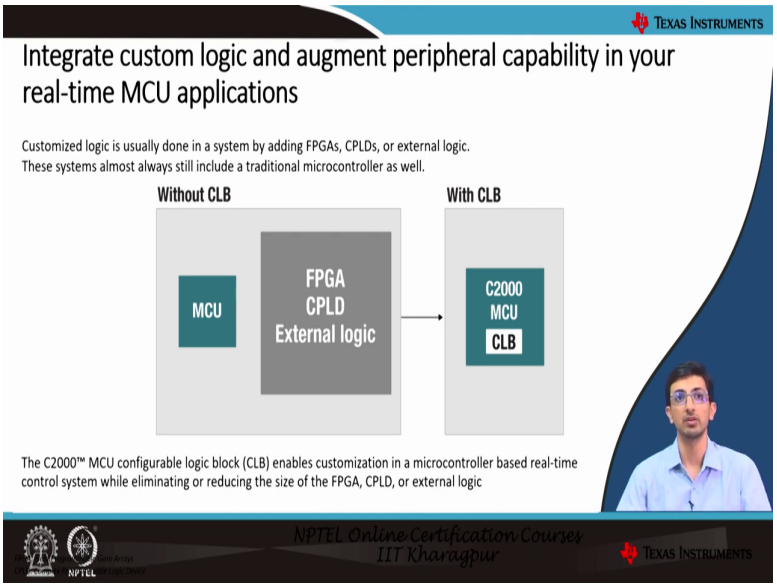
The internal reference voltage can also be used within the chip multiple. SOC triggers that are the start of conversion triggers for the ADC are possible in this. That is you can use either software all the PWMs are capable to trigger the ADC. You can even use external triggers to trigger the ADC.

ADC also has good connectivity with the peripheral interrupt expander block which is the PIE block which is the interrupt block and you can generate interrupts using this PIE block on ADC events on specific ADC events. Then there are four post-processing blocks as well which we call PPB blocks these are hardware blocks to reduce the software overhead. The use case of this PPB block is the red is capable to have an offset calibration, and error correction that triggers sample delay capture, it can even detect high, low, and zero crossing

comparison and it can also generate an interrupt for PWM trip in such cases where you want to have a zero crossing or a high low comparison.

Lastly, this ADC also has an open shot detect feature where you can even detect whether the pin that you have connected for ADC configuration is open or short to some other external source. Specifically if talking about data speed this 12-bit ADC for an f2838x device is somewhere around 3.5 million samples per second and the conversion speed somewhere comes somewhere around 250 nanoseconds. Now, this value will vary from device to device and all the device-specific information can be captured in the device datasheet.

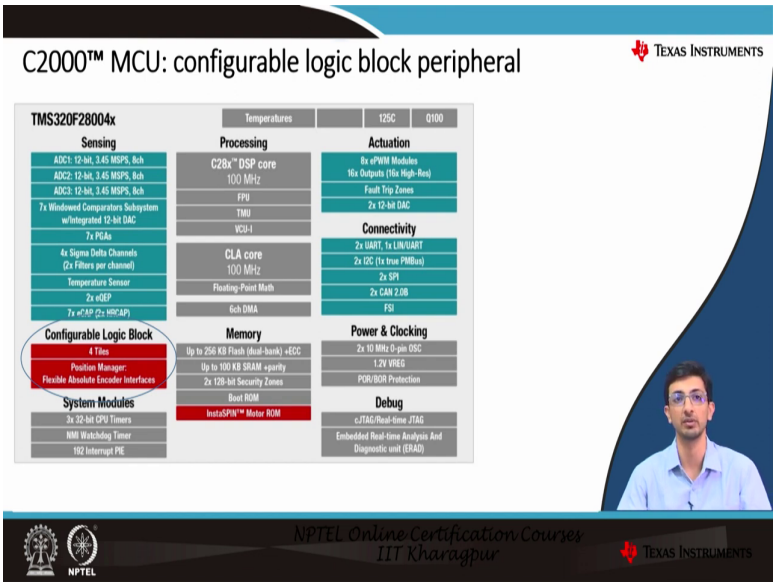
(Refer Slide Time: 15:07)



Now, coming on to the third part which is the CLB part. So, there is a customizable logic block that is available which we call a configurable logic block. This customized logic is usually done in our system by adding FPGAs or CLPD or some external logic, but these systems most always still include a traditional microcontroller as well. That is the image that you see around the left-hand side. The C2000 microcontroller unit configurable logic block enables customization and a microcontroller-based real-time control system while eliminating.

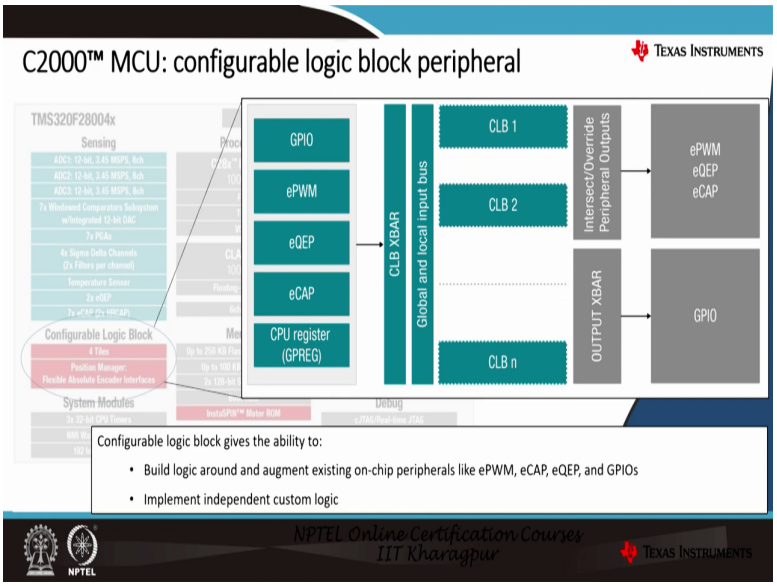
So, now you do not need an external FPGA or an external CLPD CPLD for this purpose, but in case of any minor logic that you want to implement that you can implement using the CLB block available within the C2000 device. So, you eliminate the size you will reduce the size of your whole device by eliminating FPGA and CBD.

(Refer Slide Time: 15:56)



So, taking a sample use case of TMS320F28004x we have four configurable logic blocks as highlighted in this image.

(Refer Slide Time: 16:04)



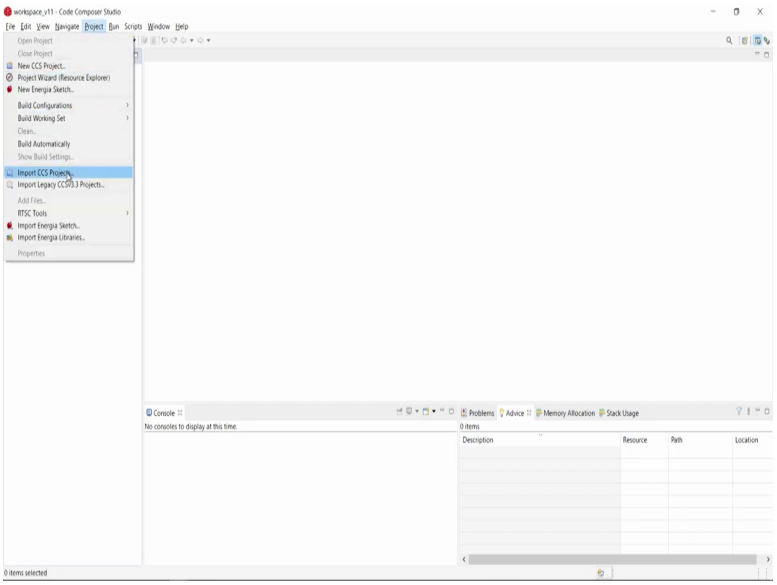
Now this configurable logic block gives the ability to build logic around and augment existing the peripherals like GPIO, ePWM, eQEP that is the quadrature in quadruples, the capture pulse etcetera. All these peripherals specifically PWM capture and QEP can be connected to different CLB tiles using something called a CLB crossbar. This crossbar is capable to connect all these IPs to the CLB.

Now, the user can implement his custom independent logic similar to FPGA and do whatever operations he wants, and based on the decisions he wants to take you can then intersect or override peripheral outputs to the ePWM, QEP, and eCAP as well as if you want to give the output back to the GPIO that is also possible using the output crossbars.

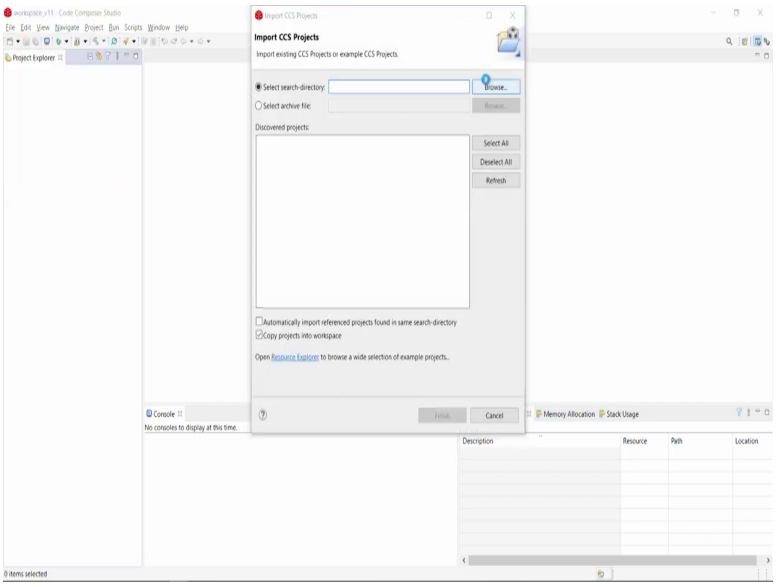
(Refer Slide Time: 16:47)

Now, we would like to show a small demonstration of how a CSS or a C2000 ware environment is utilized I will particularly show an ADC ePWM example, ok.

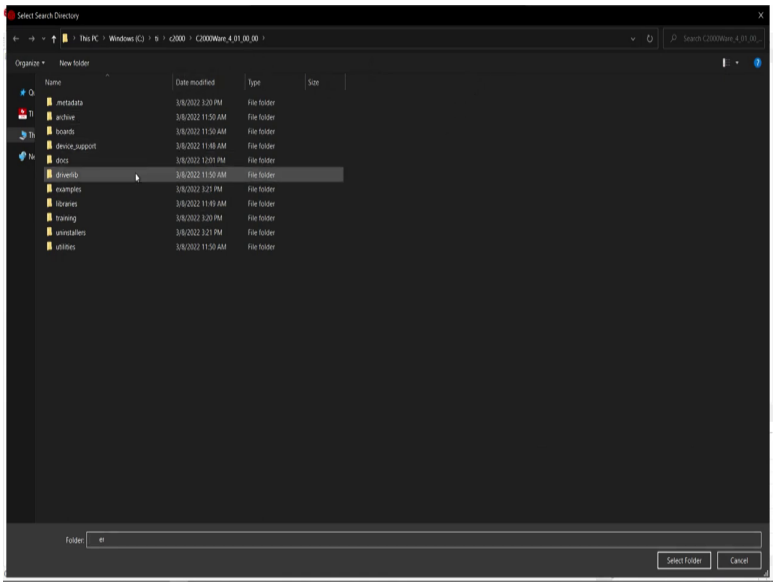
(Refer Slide Time: 16:59)



(Refer Slide Time: 17:08)

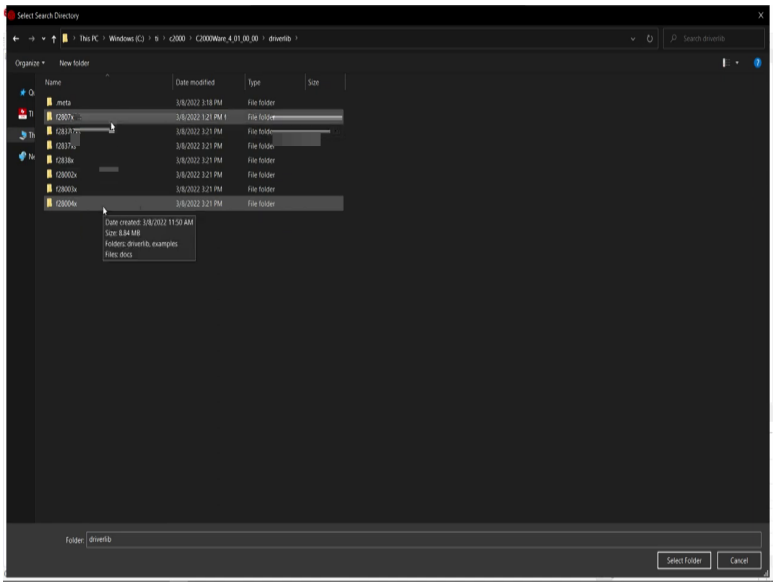


(Refer Slide Time: 17:13)

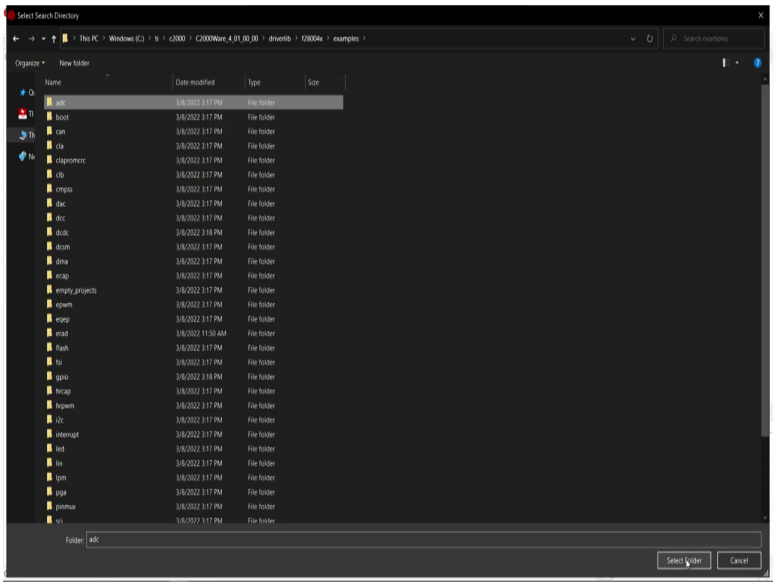


So, we start the demonstration and for that, you open the CCS window which is the code composer studio window, you can go to the project menu click on import CSS project click on browse and this will lead you to the latest version of C2000 ware that you have installed.

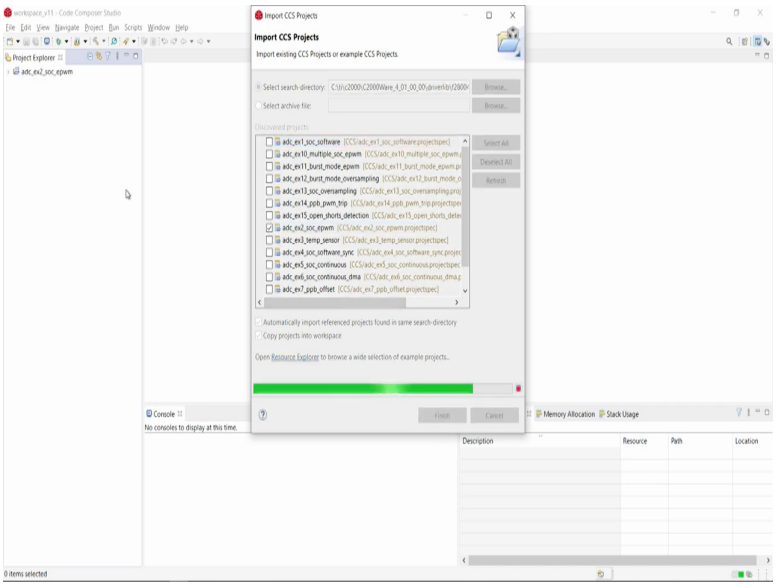
(Refer Slide Time: 17:15)



(Refer Slide Time: 17:22)

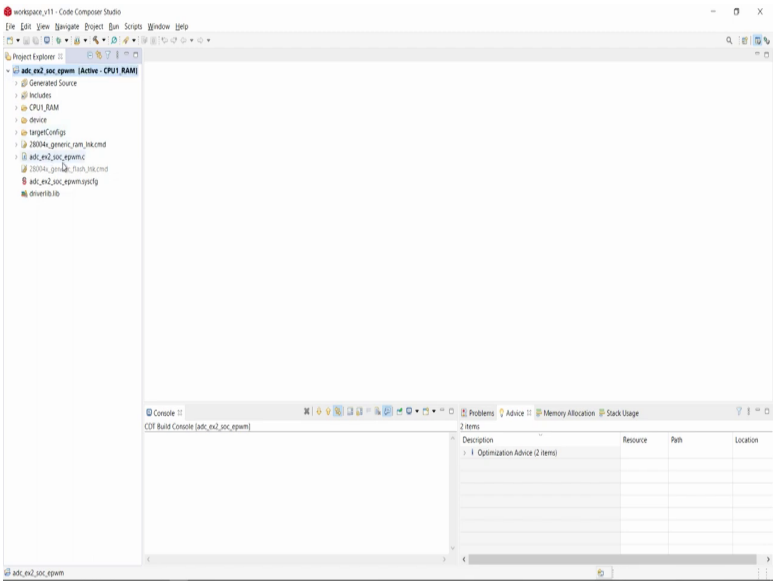


(Refer Slide Time: 17:24)



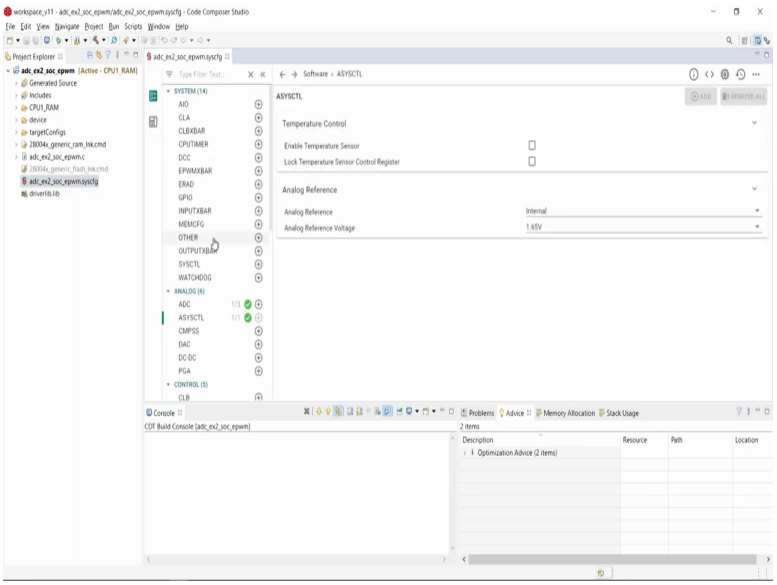
Click on driver lib click on the device that you have connected click on examples and select the example that you want to run in this case I want to run an ADC example. So, I have selected ADC and selected adc folder I will run adc soc epwm example which is example number two. Click on the example and click on finish. This will import the example into your CCS environment. You can open this example by clicking on the drop-down symbol available here.

(Refer Slide Time: 17:40)

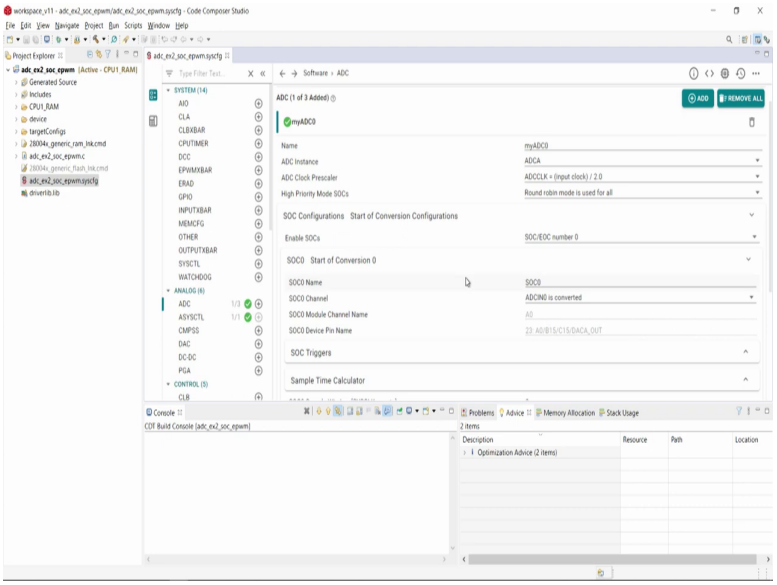


There are two main files available one is the main file that is the source file and the second one is the sysconfig file the main file contains all the initialization configuration whereas, the sys config file is a GUI where you can do all the initialization file initialization configurations.

(Refer Slide Time: 17:52)



(Refer Slide Time: 17:57)

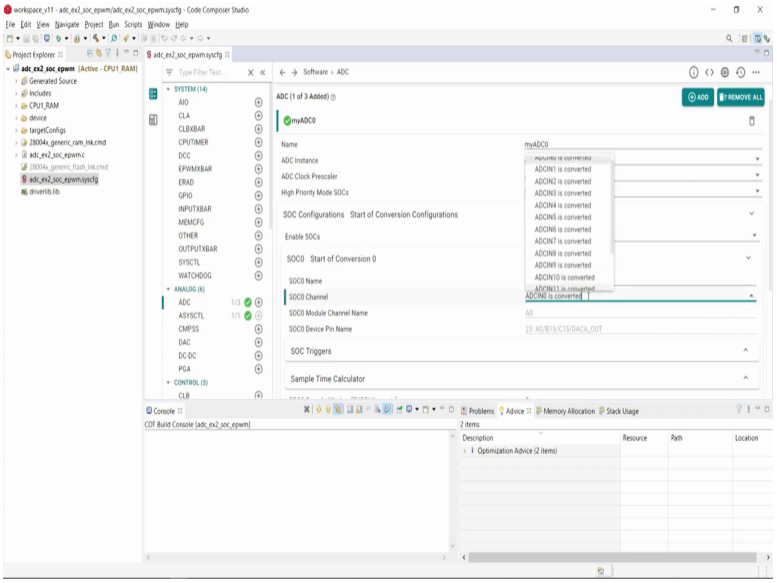


So, first of all, I will open the sysconfig file where all the initialization is done. So, you can see here the ADC is already enabled with the sys control as well. You can see all the basic ADC configurations done over here such as the pre scalar if you want to go for the high

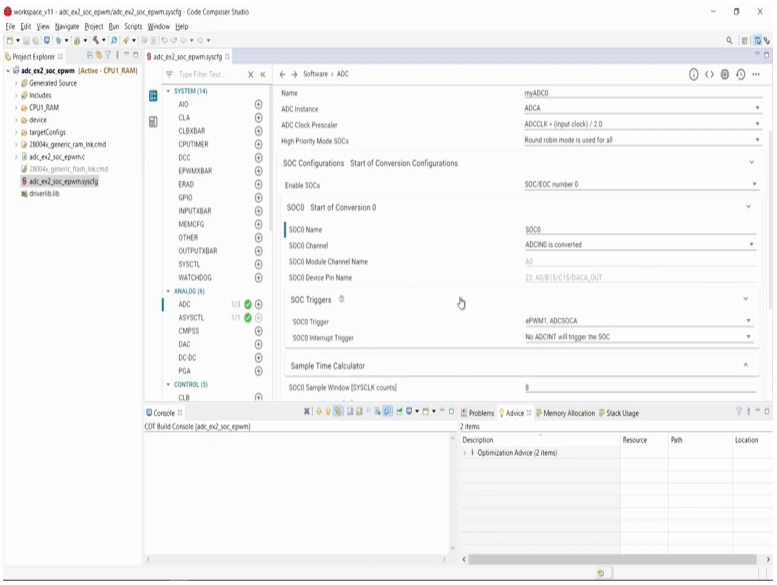
priority mode then you can enable that as well all the start of conversion can be enabled over here.

In this particular example, we have enabled the SOC number 0 enabled. So, specifically for SOC0 all the configurations can be done inside the SOC0 tab that is its name can be changed by which channel is configured you can select all the channels that are from 0 to 15 that is 16 channels.

(Refer Slide Time: 18:34)

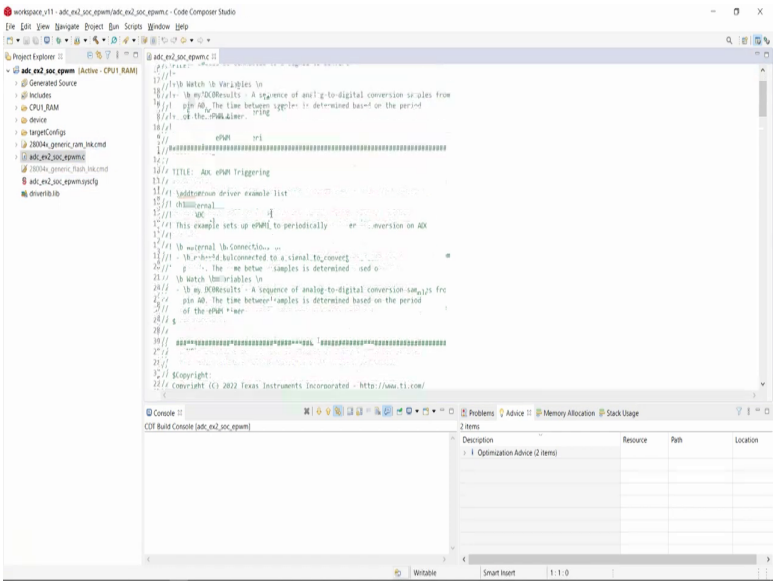


(Refer Slide Time: 18:42)

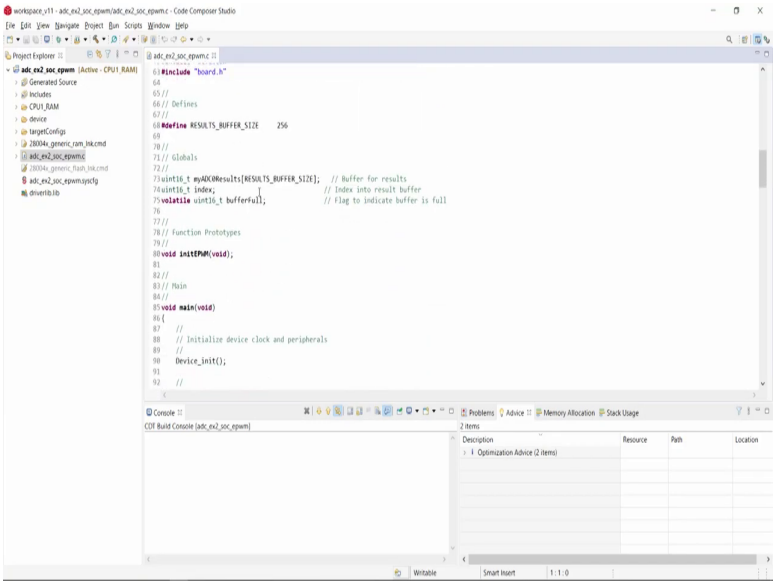


So, in this particular case, we have selected ADC in 0 for conversion. Lastly, we will have the trigger configuration that also can be enabled. So, in this particular case, our trigger is ePWM1 ADCSOCA. Once you are done with the configuration you can save this file and close it.

(Refer Slide Time: 18:54)

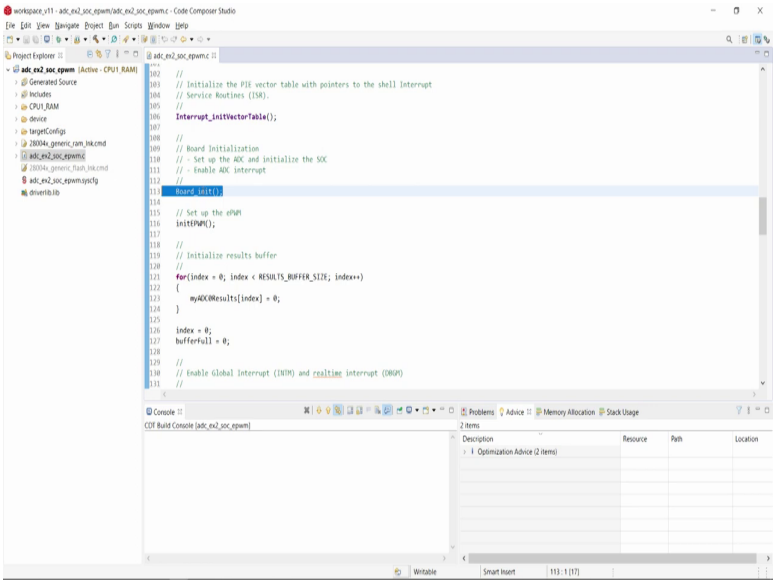


(Refer Slide Time: 18:58)

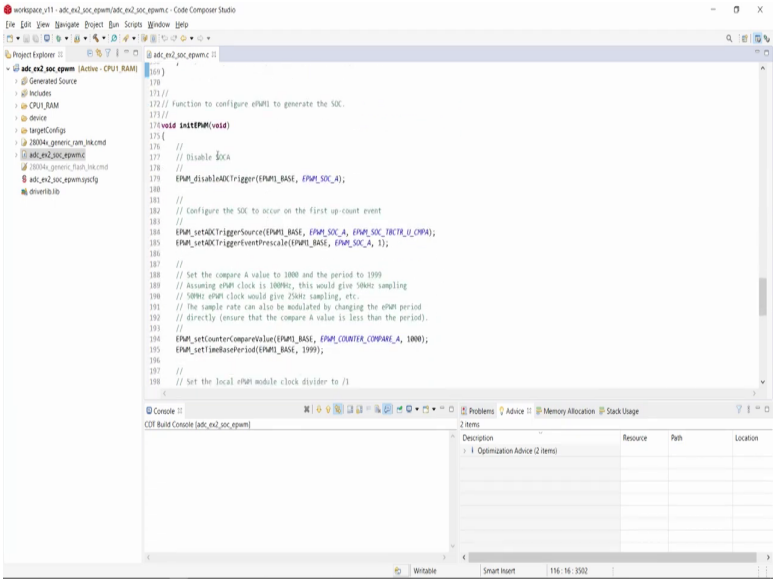


In the main coming to the main file we have the initial introduction for the example then some initialization parts are done they are the device initialization. Then there is one file and one function called a board under the board to underscore the init function.

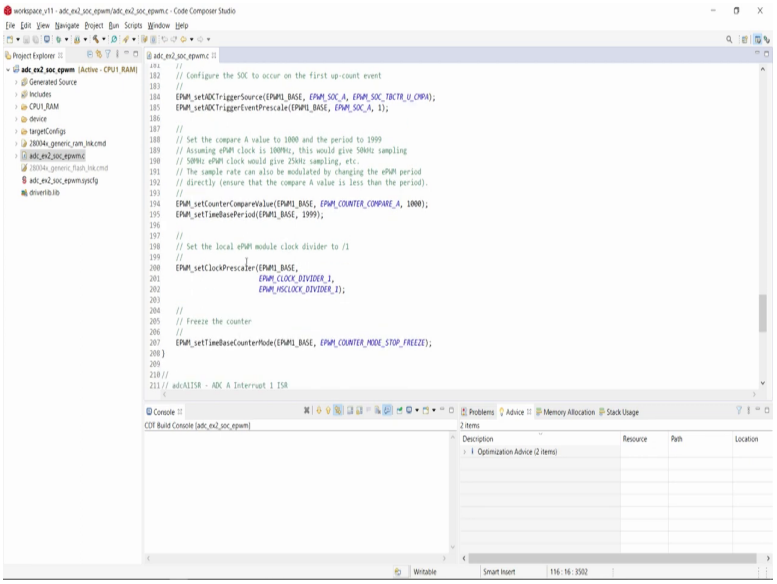
(Refer Slide Time: 19:05)



(Refer Slide Time: 19:19)

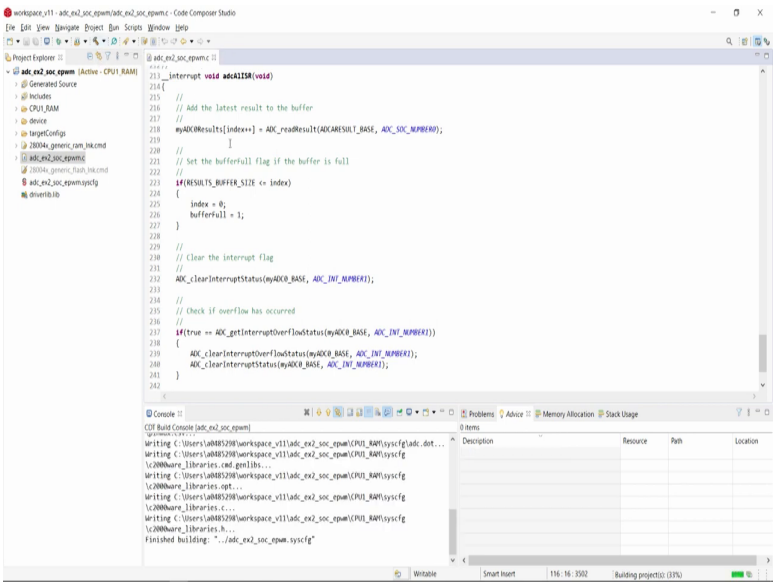


(Refer Slide Time: 19:22)



So, this particular function has all the functions that are developed using the sys config file. Then we move on to the init PWM function which sets up the PWM for this particular example. Coming down we have the INIT PWM function where we have all the ADC configurations done for the PWM. Lastly, we have the ADC A1 ISR where we read the results.

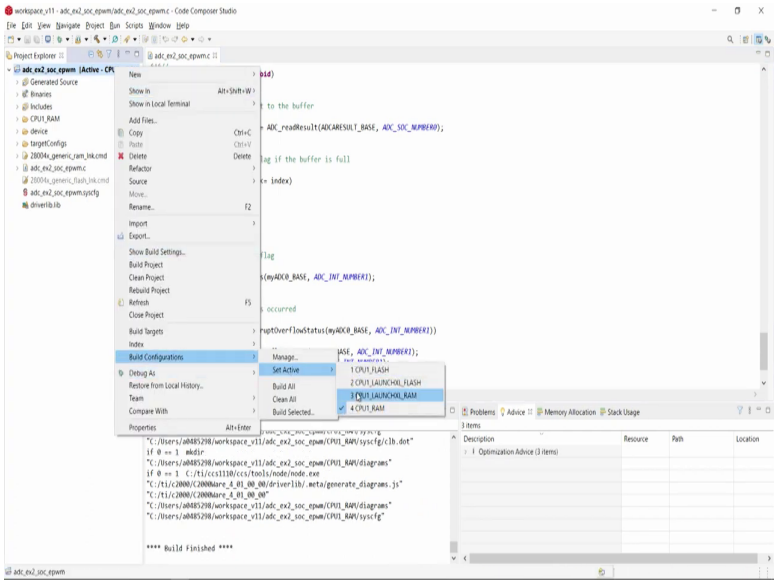
(Refer Slide Time: 19:25)



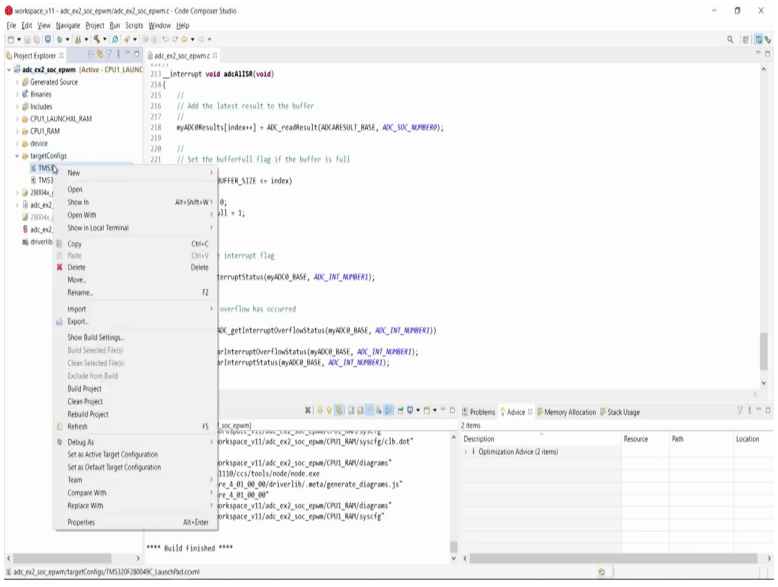
So, we have a 256-sized buffer where we will read the results for the ADC conversions. The first step once we are done with the configuration is to build the code. So, for that, I will click

on the symbol and build the code. Make sure that you have selected the appropriate build configuration. So, in my case, I have a TMS320f280049c launch pad connected to my device.

(Refer Slide Time: 19:50)

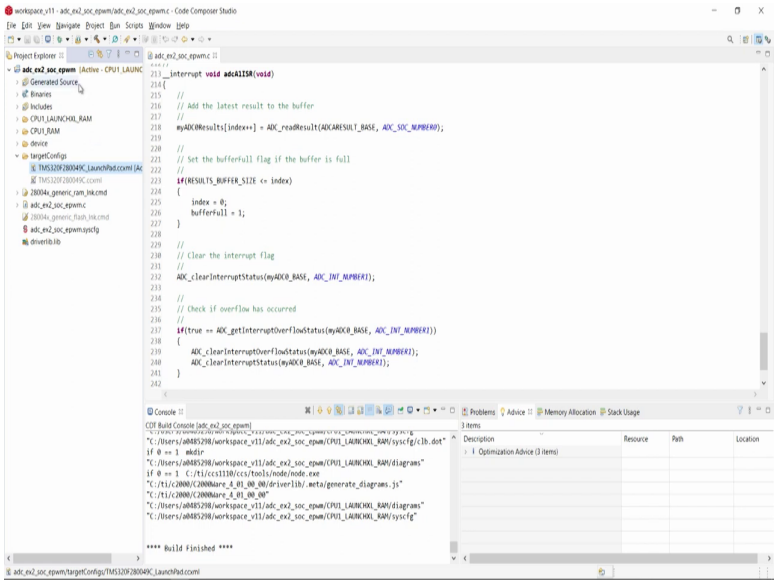


(Refer Slide Time: 20:03)



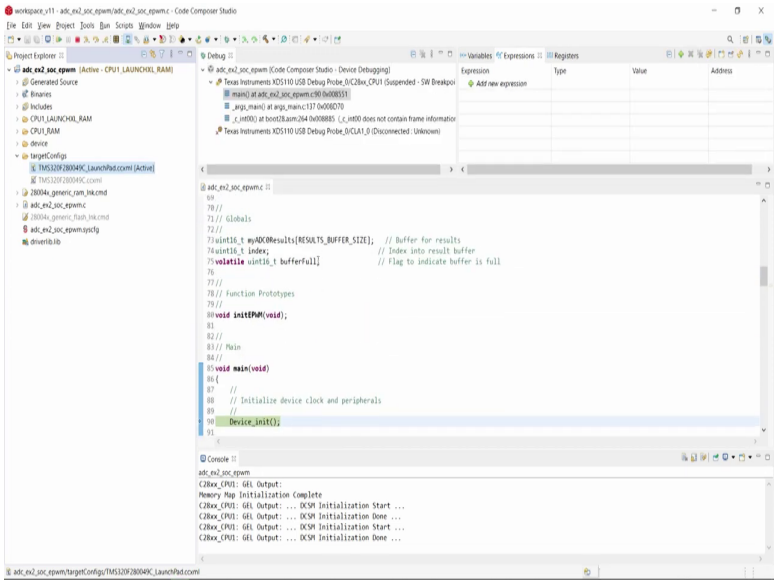
So, based on that the active configuration to be selected will be CPU1 launch excel RAM and for that the target configuration to be selected is TMS320f280049c underscore launch pad dot cc XML. The simple target configuration without the launch pad tag is for the control card hardware. Once you are done building with the example code once the build finish shows up you are now ready to debug the code on the hardware.

(Refer Slide Time: 20:25)

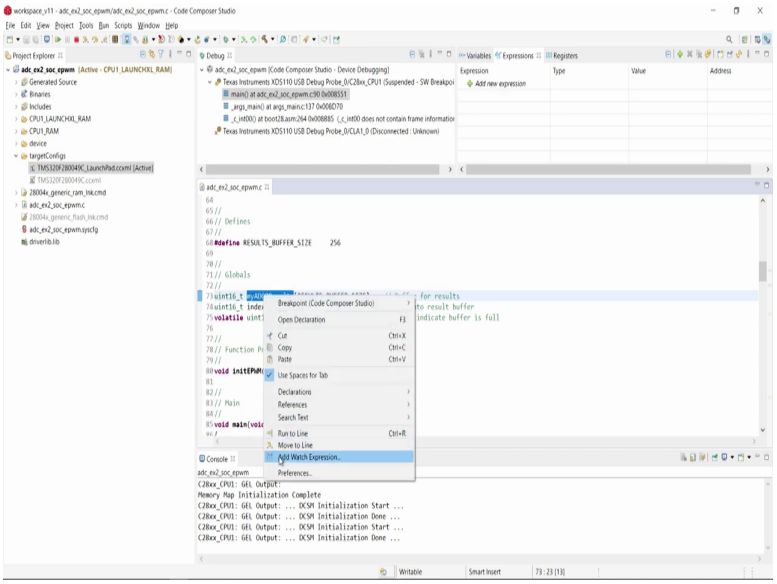


But in case this example would have any error it could have been shown in the problem tab. Since this example is clean I will now go ahead and debug the code.

(Refer Slide Time: 20:45)

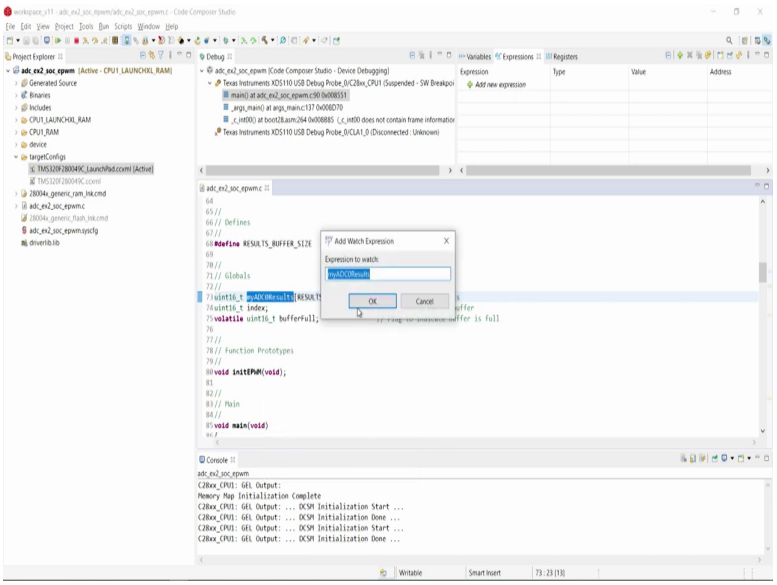


(Refer Slide Time: 20:58)



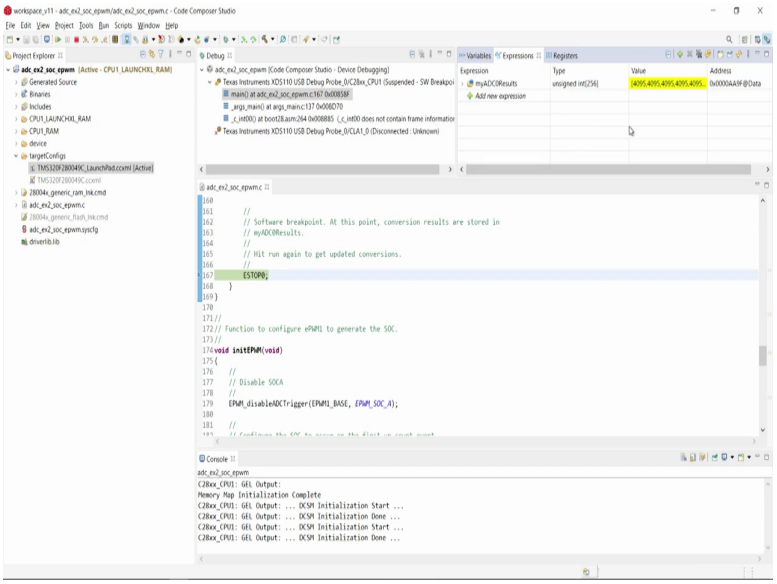
So, once we have the code debugged on the hardware what you can do is that we have the ADC result in ADC0 result in this variable holding is going to hold all the results for our ADC. So, I will add this particular variable to my watch expressions.

(Refer Slide Time: 21:00)



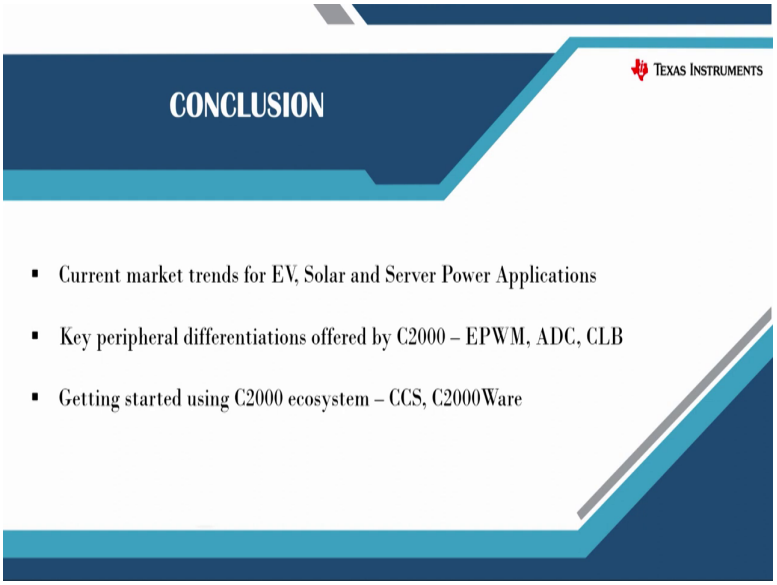
So, I can right-click on that select add watch expressions, and click on ok. This will add my ADC result in the 0 variable into the expressions window. I can now run the code to see whether I am getting my desired results or not.

(Refer Slide Time: 21:14)



So, since I have connected a 3.3-volt source to my ADC input which is a channel ADC 0 I am getting 4096 for the 12-bit ADC configuration. That is all I wanted to cover for this demonstration.

(Refer Slide Time: 21:28)



So, as a part of the conclusion of this lecture, we studied the current market trends for EV, solar as well as server power applications. Also, we went through some of the key peripheral differentiations offered by C2000 that is in EPWM, ADC as well as CLB and lastly we also

went through getting started using the C2000 ecosystem that is in CCS and C2000. I hope you enjoyed the video.

Thank you.