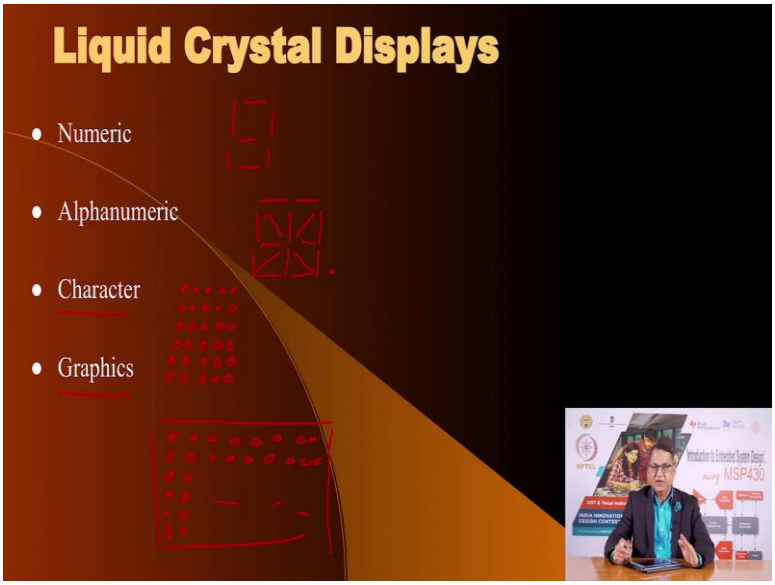
Introduction to Embedded System Design
Professor Dhananjay V. Gadre
Netaji Subhas University of Technology, New Delhi
Lecture 28
Interfacing Liquid Crystal Displays (LCD)

Hello and welcome to a new session for this ongoing online course on introduction to embedded system design. I am your instructor Dhananjay Gadre and in this session I will be talking about how to connect additional output devices for human interaction namely the liquid crystal display.

So, liquid crystal displays are commonly called as LCD outputs and we are going to talk about how to connect them to our favourite MSV430 micro-controller. Now, you would have seen LCD displays commonly around you in many of the gadgets that you use on a regular basis. For example, on your calculator, on your specially the AC remote and so on. And these are common output devices because they allow very low power operation.

(Refer Slide Time: 01:20)



These liquid crystal displays come in various types. The simplest one are called numeric displays and these are the ones which are used in calculators. The numeric displays will allow you to display information in seven segment format like you have seen for seven segment displays which use LEDs but you have same format for LCD also. So, you would have seen this on calculators.

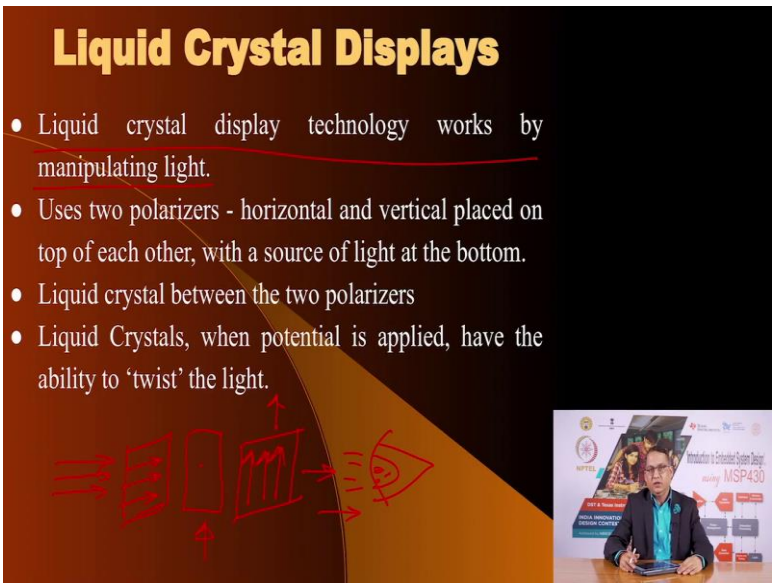
Similarly, the seven segment display is not able to display a variety of information and so you have another type of output configuration which is called alphanumeric display which is similar to the numeric display except it has additional segments and typical alphanumeric display will look like this. Something like this.

This allows you to display alphabetical characters with little more clarity than is possible with a numeric displays. Then the third type of liquid crystal displays which are common are called character LCD displays in which you can display ASCII characters in the form of a 5 by 7 or 5 by 8 matrix of LCD elements, liquid crystal elements. Something like this.

Now, on such a matrix you can display almost a large number of alphabets, numbers, special characters and so on and so forth and that is why these are called character displays. And the most common of the liquid crystal display is what is called as graphics. In which row and column full of pixels are available that you can use to create whatever graphical or numeric or alphabetical information that you would like to display. So, these are the common LCD displays that are available.

We will now investigate how are LCD displays different from LED displays and the basic difference is in a LED display emits light. So, it has a source of light namely the light emitting diode. In the case of LCD we do not have the mechanism to generate light in a way that a LED does. But LCD allows you to manipulate light.

(Refer Slide Time: 04:23)



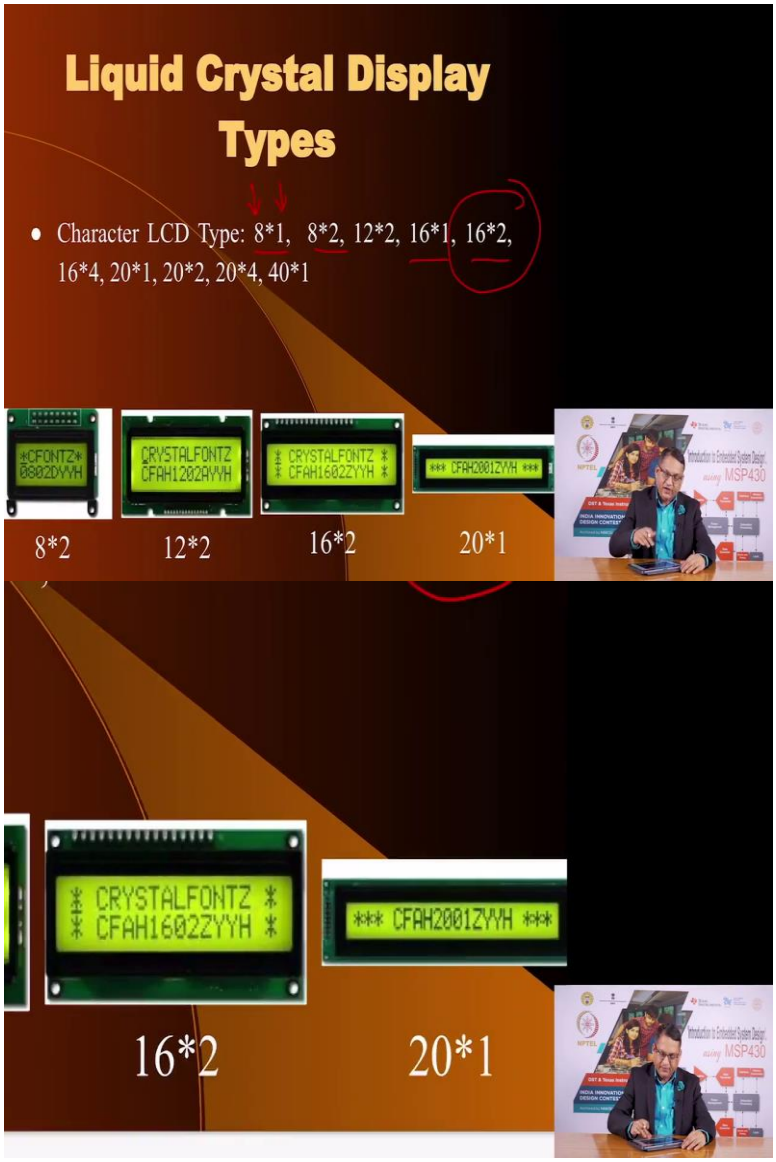
So, LCD liquid crystal display works by manipulating light. What does it consist of? It consist of several layers of structures. Initially a source of light, the source of light can be ambient light as well as light from a LED source, followed by a polarizer in which it polarizes the incoming light let us say in the horizontal direction.

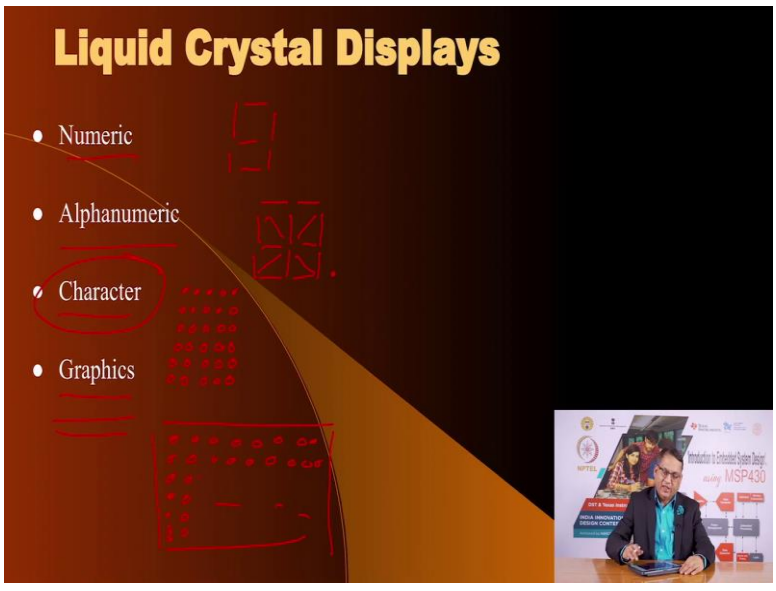
Then you have the liquid crystal in between and then you have another polarizer where the polarization happen in a vertical direction and then of course here is the human eye which is able to see what is reaching the human eye. Now, how does it work? If the 2 polarizers, polarize a light in such a way that one is polarizing it in horizontal direction and the other offers polarization in the vertical direction no light from the source would reach the human eye.

But the intervening liquid crystal has a great ability that once you apply potential to it, it has the ability to twist the light meaning it has the ability to change direction of the light which reaches this the crystal and change its direction and if it is aligned to the vertical direction then the light will come out and you can see it, the human eye can see it.

And so LCD is very different from LED. In LED you generate light, whereas in LCD you manipulate existing light. So, this is the basic difference. Because it is not generating any light, it is able to operate at much low power levels as compared to LED display and that is the reason why LCDs are so common in everyday gadgets for sharing information for human interaction.

(Refer Slide Time: 06:35)





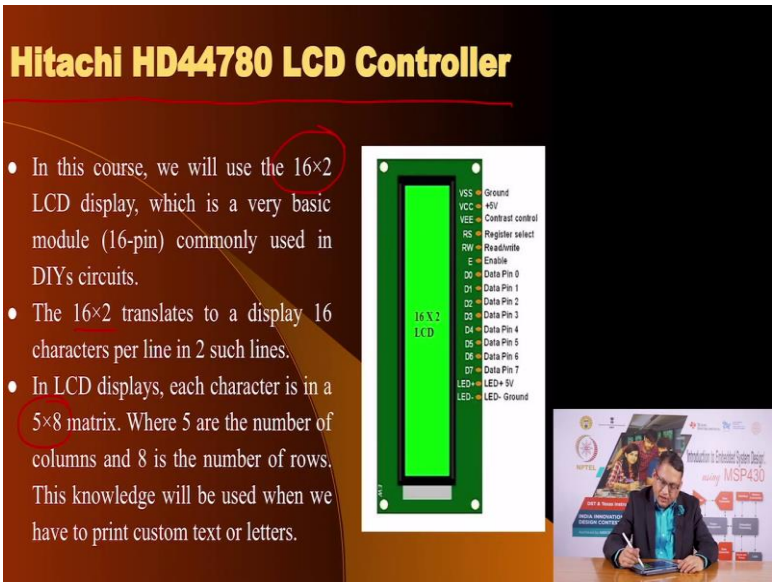
These liquid crystal displays specially the so called character LCD displays, we are not going to talk about numeric and alphanumeric because numeric displays and alphanumeric displays do not have the ability to share too much variety of information.

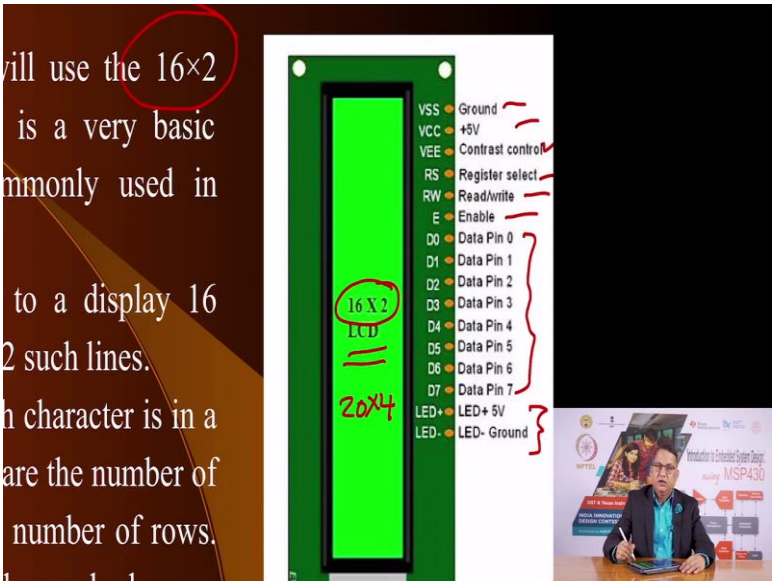
We are going to concentrate on character LCD displays in this lecture and I will mention what all graphic displays, LCD displays you can have. In liquid crystal display of the character type you get it in the form of what is called as 8 cross 1 or 8 cross 2 and these refer to the number of characters in the first instance and the second information is the how many lines you have.

So, when I say that I have a 8 cross 1 LCD character display, it means I have 1 line and this line has 8 characters. Each character can be any ASCII value, any ASCII character you can display. Similarly, 8 cross 2 would be 2 lines of 8 characters. 16 cross 1, 16 cross 2 these are the most in fact 16 cross 2 is the most common LCD display that you can find and in this exercise we are going to use a 16 cross 2 display.

Here is an example of a variety of this. This is a 8 cross 2, next to it is a 12 cross 2, then you have a 16 cross 2 and you have a 20 cross 1. You may have seen this at various locations and we are going to use this in our experiment.

(Refer Slide Time: 08:13)





Now, one could create a mechanism for connecting such a display to a micro-controller and the most common interface for connecting such a display to a micro-controller is what is called as Hitachi HD44780 LCD Controller. This has become a defector standard for character LCD displays which spans the entire range from 8 cross 1 to 20 cross 4.

It has a common interface as we see here. This is the interface that it requires certain pins for sending information and reading information from the display for providing power supply to this display, for providing power supply for the backlight. The source of light which we are able to stop or allow to come out of the LCD that is the light we were talking of initially and this has been available for a very long time. In this display the one we are going to use in our experiment, we are going to use a 16 cross 2 LCD display.

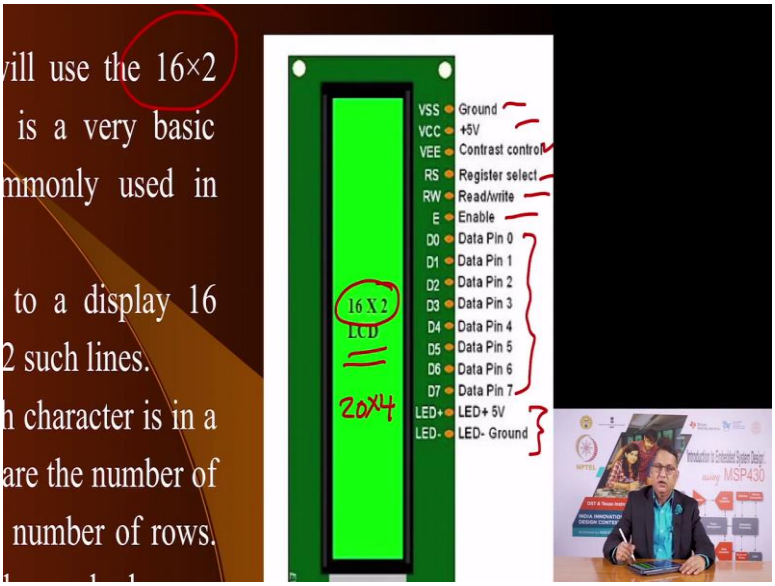
This is the most common display that is available and therefore the cheapest that you can find. The 16 cross 2 translates to 2 lines and each of the line allows you to display 16 characters. Each character can be displayed in a 5 by 8 matrix which means that a 16 cross 2 LCD, how many pixel it has? Let us calculate that.

(Refer Slide Time: 09:52)

A 16 cross 2 LCD where each of the displays, each of the character is 5 cross 8, that means 40 pixels. Therefore, the total number of pixels on a 16 cross 2 LCD is 32 into 40 and that is a large number of pixels that you have. Now, obviously to be able to turn control each of these pixels is equal to controlling so many LEDs that we have seen in a previous lecture and the only way to achieve this is by way of multiplexing except the responsibility of multiplexing is not on the micro-controller.

The LCD has resident micro-controller or a single purpose computer which is doing that job on our behalf. All we have to do is send information to the controller that what do you want to display and this information, this interface that we are talking about basically allows an external controller such as MSP430 to communicate with the resident single purpose computer on the LCD display, convey information to it so that controller is able to multiplex all these LCD pixels to display whatever information that the user may want to provide.

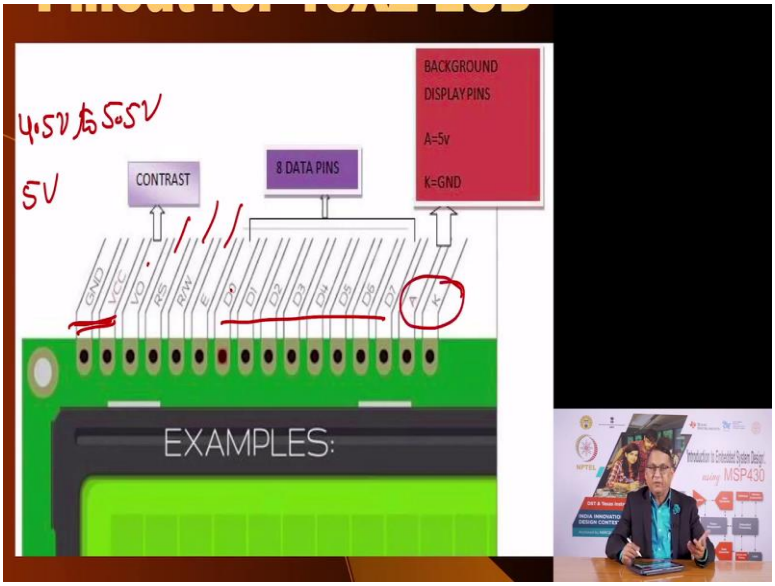
(Refer Slide Time: 11:30)



And the Hitachi controller HD44780 offers a unique and uniform interface which has these 8 data lines, some control lines like this and this and a power supply here, a power supply for the backlight and a contrast control. Using a 16 pin interface any micro-controller such as MSP430 or any other micro-controller can display information in a uniform fashion and it does not matter that you have a 16 cross 2 LCD, you can also have a 20 cross 4 LCD.

Even then the interface to the LCD will remain the same. That is the beauty of this Hitachi HD44780 LCD controller. Let us go through the pin out of this LCD controller. The most important part is what is the power supply required.

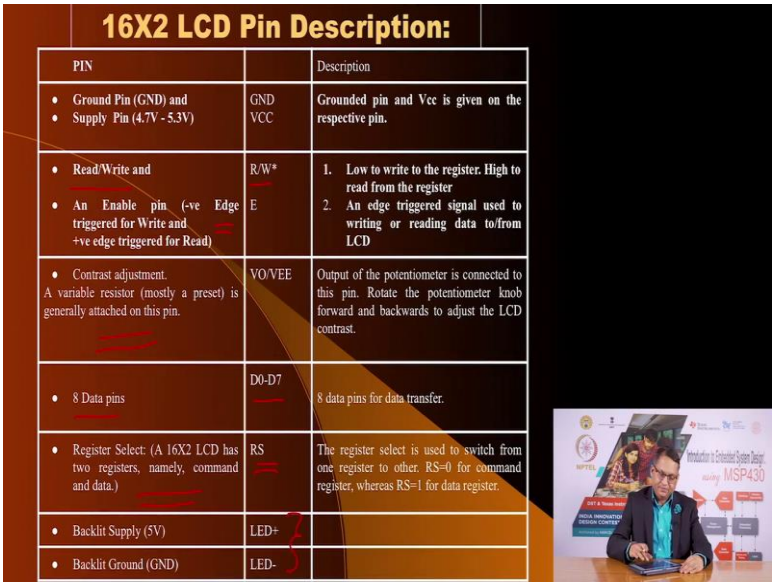
(Refer Slide Time: 12:26)



The power supply that you require is here ground and VCC and this is the commonly available LCDs required 5 volt. Of course, 5 volt does not mean 5 volt. Usually it is 5 volt plus minus 10 percent which means you can have a supply voltage from 4.5 volts to 5.5 volts. Apart from that you need to set the contrast and I will share the circuit that you have to connect to be able to adjust the contrast of the LCD. Apart from that you have data pins D0 to D7 as you see here and control signals which is read write enable and register select.

These are the 3 control pins that you require for communicating with the common LCD display and then you have 2 LEDs, 2 pins of LED, anode and cathode which provide the backlight to this LCD. These LCDs are available in many colours like green colour with black segments or any other colour and you can choose whatever is available for you.

(Refer Slide Time: 13:40)

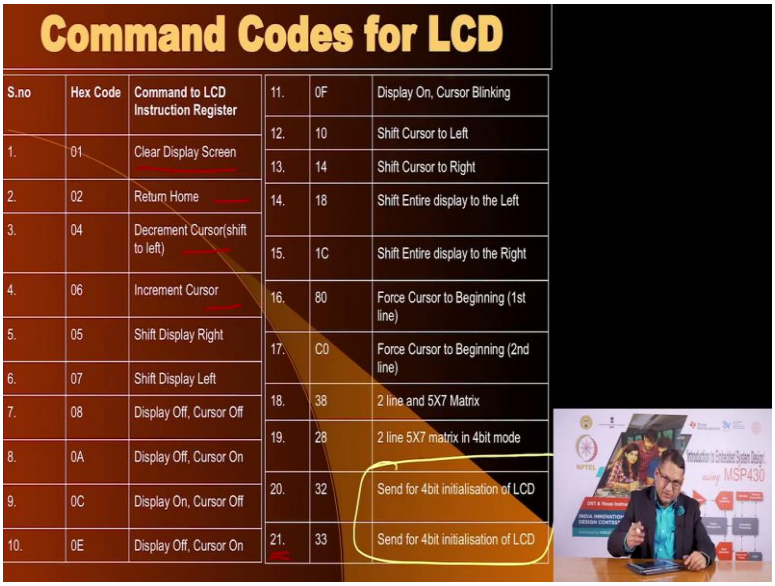


Now, these are the description of the pins. You have read write pin. If the read write pin is 1 that means you want to read internal register. If you set this pin to 0 that means you want to write into that register. You also have a enable pin which is like a chip select.

You have to keep this signal low to be able to a negative edge on this signal which indicates to the LCD controller that you are writing something. Then you have a contrast adjustment on one of the pins. You have 8 data pins D0 to D7 and you have a register select.

You have 2 registers. One is for command and the other is for data. So, if this value is 0 that means you are sending information to the command register. If this value is 1 that means you are sending information to the data register and apart from that you have 2 pins which are the anode and cathode of the backlight LED.

(Refer Slide Time: 14:41)

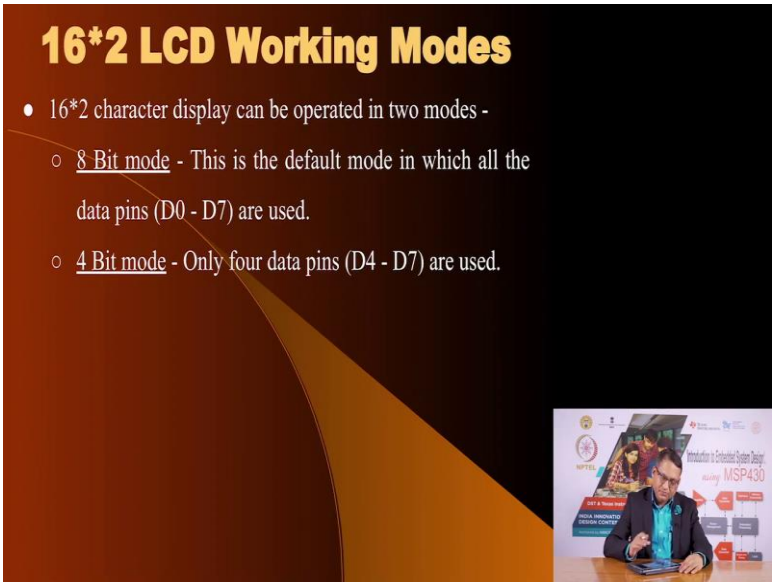


Now, to create a uniform interface the Hitachi LCD controller offers a set of commands and any controller can send these commands and the Hitachi controller will react appropriately. At this point I would like you to take this LCD that you have in your hand and flip it around and see what is behind that LCD.

You will find 2 black dots and those black dots are what are called as chip on boards. Underneath those black dots are ICs which are the controllers and the memory for this LCD controller. So, these are the commands. What commands you send. You can send a command to clear the display. You can send a command to return the home. Home means the initial position of the cursor. You can increment or decrement the cursor. You can go forward or you can go backward and so on.

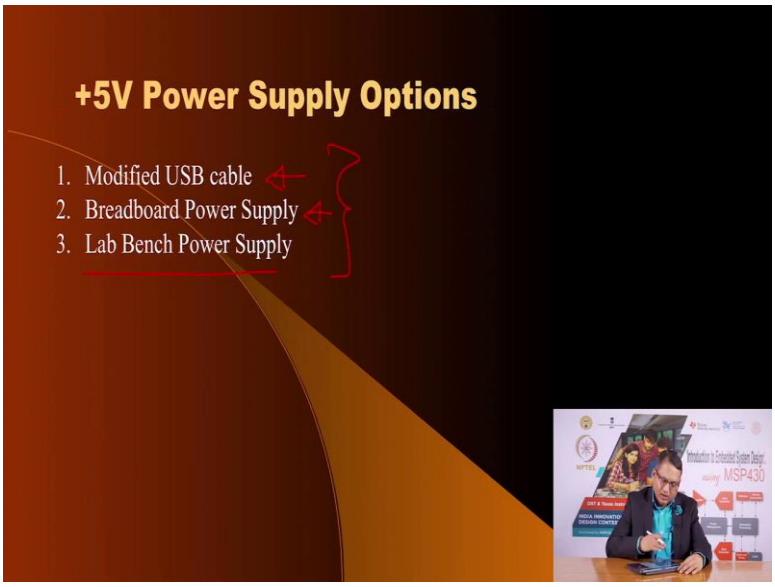
There are about 20-25 commands. Here as you see 21 commands which allow you to interact with the LCD controller. Some of these commands we will see when we do an exercise. Let us go towards that.

(Refer Slide Time: 15:56)



Now, one of the problems with the LCD 16 cross 2 or any other character LCD is that it works at 5 volts.

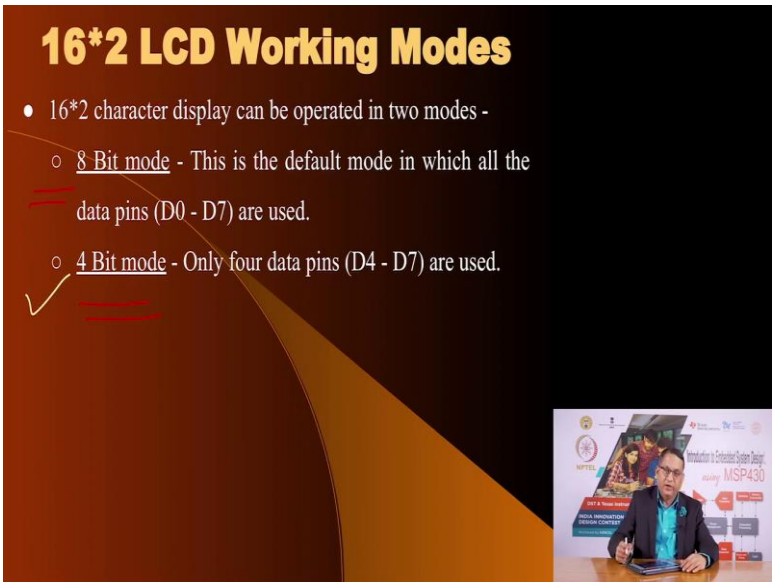
(Refer Slide Time: 16:08)



Here, it works at 5 volts and so we have this issue that our MSP430 lunch box operates at 3.3 volts but the LCD works at 5 volts. Now, how do you provide 5 volts? So, in this lecture we have offered you several ways of creating 5 volts supply so that you can power your LCD and the first method is to use a modified USB cable that you may have or if you search the internet you can get access to what is called as a breadboard power supply, small module that will fit into the power supply pins, the

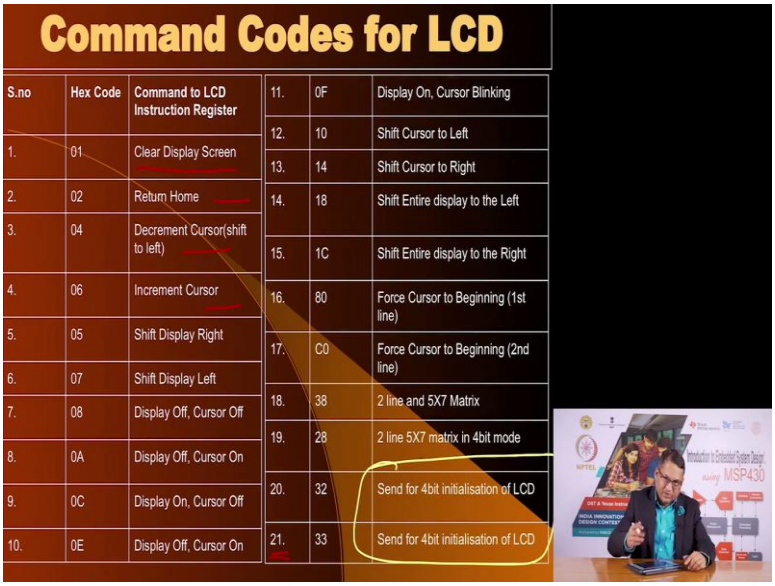
side pins of a breadboard easily or if you have access to a lab power supply, you can set it to 5 volts and you can use it to power your LCD also. I come to that in a brief moment but let us go back to the interface of the LCD.

(Refer Slide Time: 17:04)



Now, because you have 8 pins for interaction the LCD, the controller inside the LCD expects you to send commands over these 8 pins. But often times we do not have 8 pins. The Hitachi interface is very benevolent in that sense that it will adjust to send you information on 4 bits also. Instead of 8 bits you can communicate with LCD controller on 4 of the 8 pins and that is called the 4 bit mode. The default mode is 8 bit mode but if you like, if you do not have pins select the 4 bit mode.

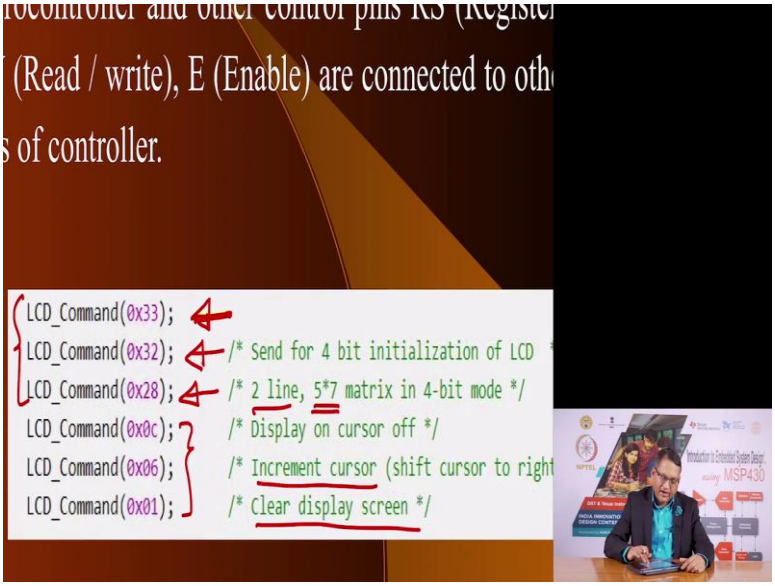
(Refer Slide Time: 17:43)



In fact one of the commands as you see here, these 2 commands allow you to tell the, communicate to the LCD, that no we do not have 8 pins to communicate to you and we are going to send information on 4 bits also. So, you have to send these commands and we will see briefly how you send these commands.

So, you have a 8 bit mode and a 4 bit mode. Now, in the 4 bit mode the actual information is being send, you want to send 8 bit of information, but because the communication connection only has 4 pins, you have to send information on those pins twice so that you can send entire 8 bit of information. So, you have to tell the controller that we are going to send 4 bits and then another 4 bits, the lower level and the higher level and the way to do that, here is the sequence of instructions.

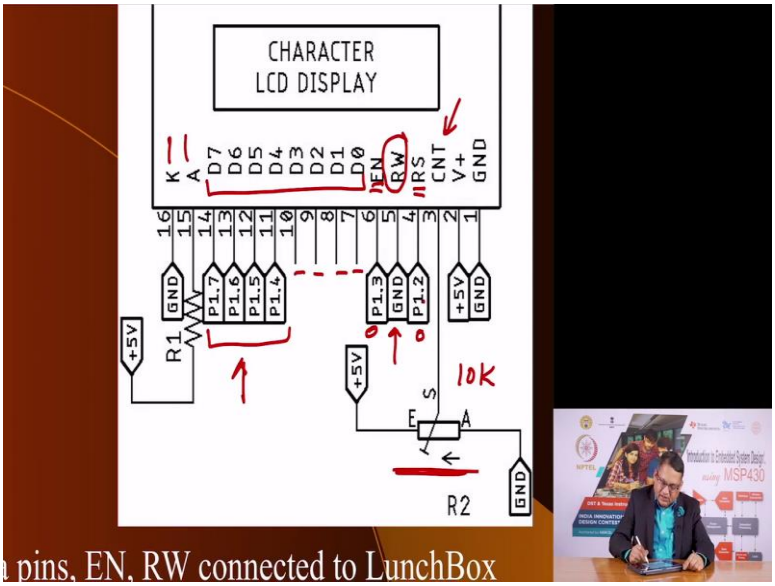
(Refer Slide Time: 18:38)



You send a information like this and say you send a value called 33. 33 is in hexadecimal. Then you send 32 and then you send 28. When you send these 3 bytes it conveys to the LCD controller that you want to communicate with the LCD controller in a 4 bit mode and you want to have 2 lines and each of the line will have character which are defined as 5 by 7 matrix.

And then these are additional commands where you are turning the cursor off and you want the increment cursor automatically and you want to clear the display screen that any previous information is cleared off. But the most important is that you want to send these 3 bytes of information to the controller but you do not have 8 bits to send, 8 wires to send. You are sending it in chunks of 4 bits each. So, we will see when we look at the code for doing this how we achieve that.

(Refer Slide Time: 19:46)



These are the connections you will find on a character LCD. You have anode and cathode which are for the backlight LED. You have this 8 data lines and if you so wish you can choose not to use 4 of the lines. So, you see these 4 lines D0 to D7, D0 to D3 are not being used and you can only use 4 pins and the numbers that you see here are, they refer to the 4 pins of MSP430.

So, we are going to connect P1.4 to P1.7 to send data. We are going to use P1.3 for the enable signal. We are going to use P1.2 for the register select and we also have read and write and we have permanently grounded it. What does it mean? That we permanently want to write into the LCD. We do not want to read any information back. That saves you 1 pin.

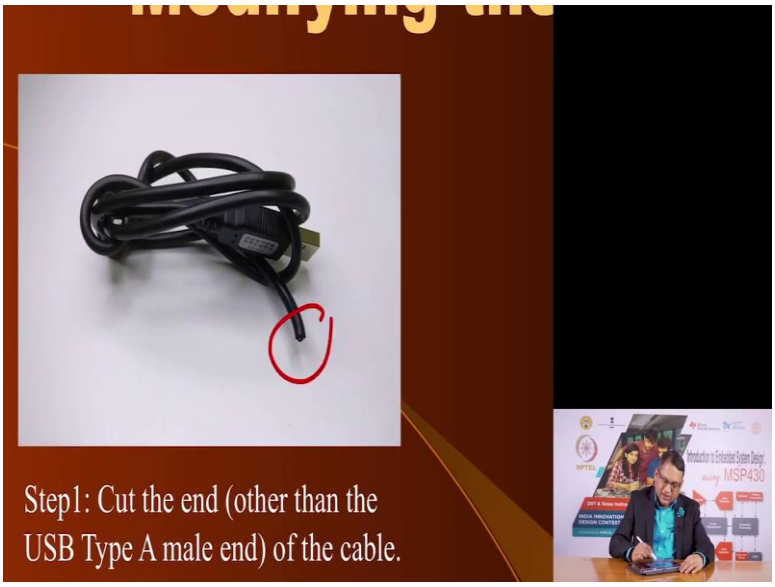
So, with this you are able to using just 6 pins - 4 for the data and 2 for the control signals. This one and this one with 6 pins available on your micro-controller you are able to control a LCD. Apart from that you need a contrast which is achieved by having a 10 kilo ohm pre-set where you adjust the pre-set to give you the contrast that you want then you leave it there and this is the supply voltage, 5 volts but of course 5 volts plus minus 10 percent.

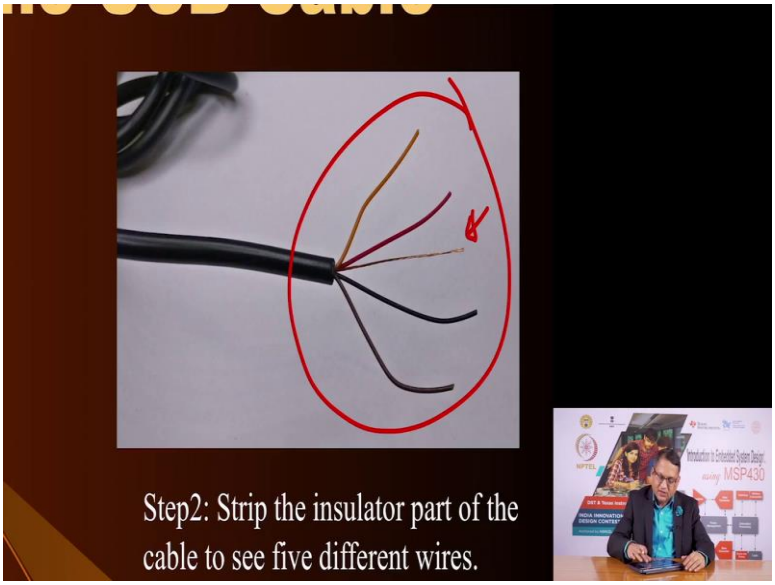
These are the connections that you require, that you need to make and connect it to your micro-controller in this case MSP430 lunch box to be able to send information and display that information. Now, coming back to the power supply requirements.

Since our MSP430 lunch box does not have 5 volt available easily, we have to find a mechanism to provide 5 volts.

One option is that you take any USB cable whether it is a micro-USB or a mini-USB output, plug it into your power bank or onto your laptop and on the other side you can cut it and select appropriate wires to provide 5 volts and in this PPT we are going to show you how we can do that.

(Refer Slide Time: 22:03)

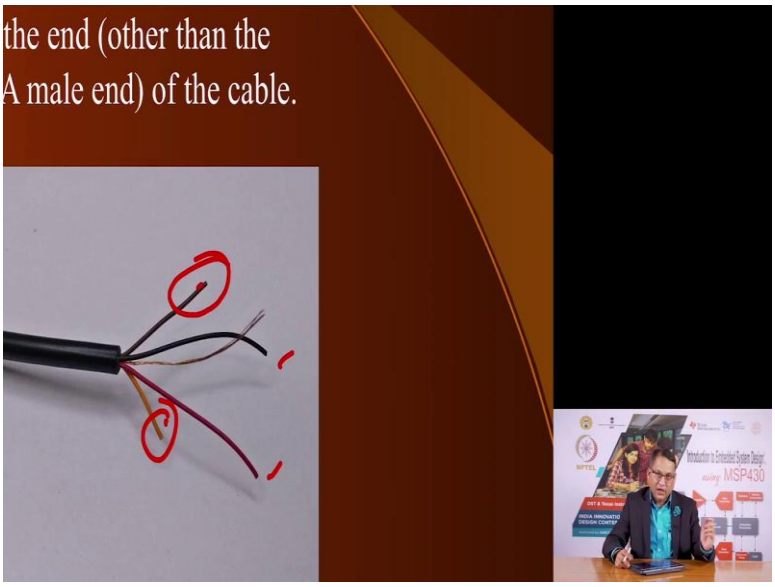




So here is the, you take a USB cable like this. Cut one side off. This is the cut side. Then you strip it of and you will find the insulator side there are 4 wires. The 2 of the wires are yellow and black, red and black and that is the power supply plus 5 and

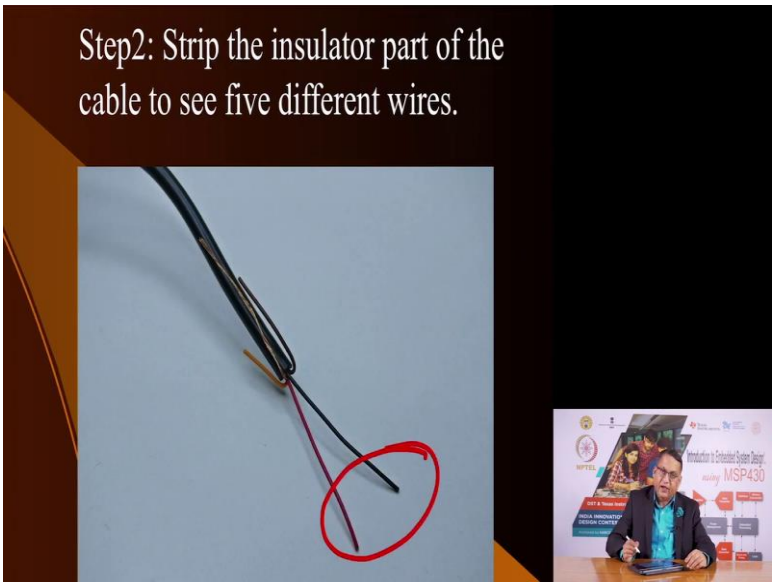
ground and then you have a copper wire which is like the shield and then you have 2 more wires. Those are for communicate data plus and data minus. We do not need them in our application so we are going to cut them off.

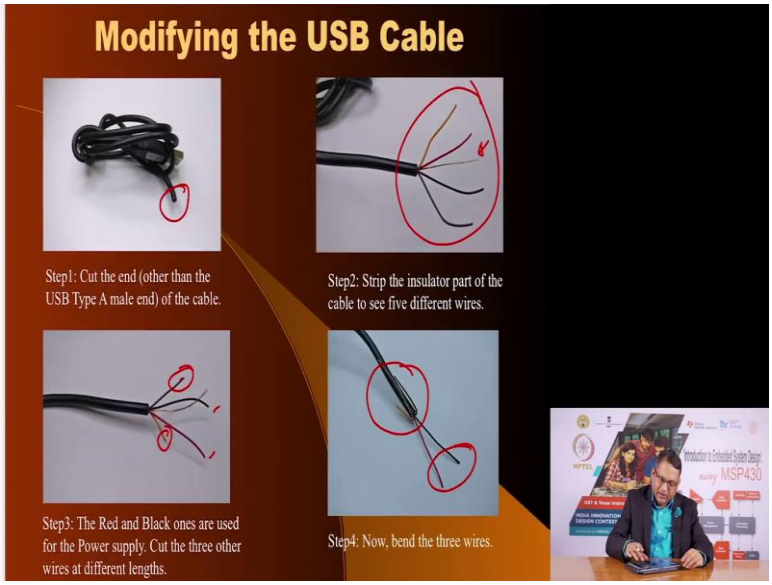
(Refer Slide Time: 22:35)

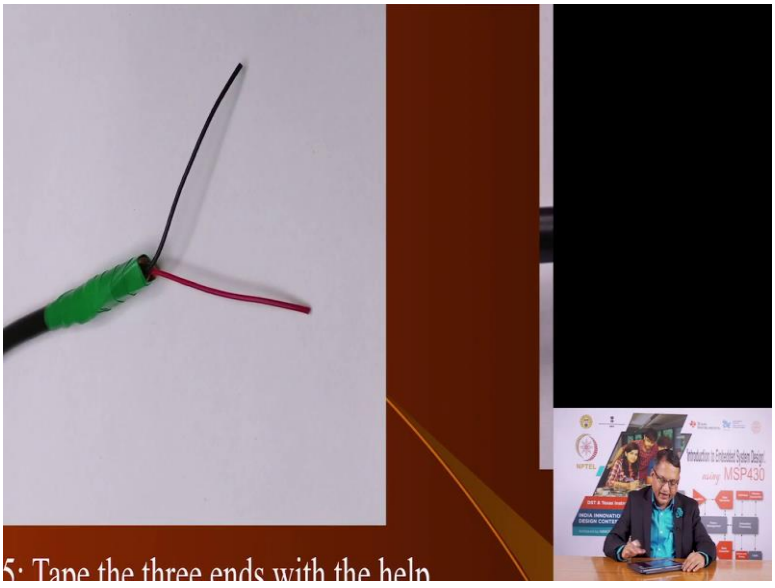


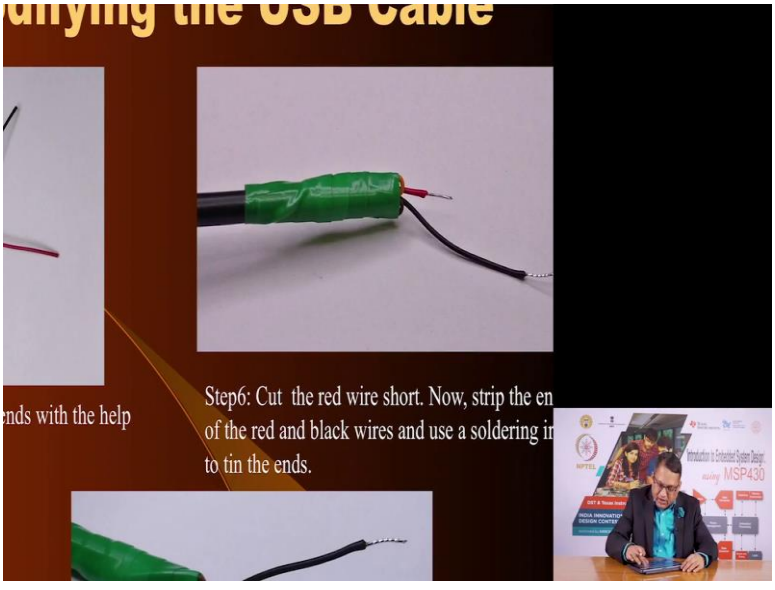
Then you cut them off at different length. The black and red wire you keep of the maximum length and the other 2 wire this one and this one you cut at different lengths because you do not want to short them because that may give false information to the cable where you are connected it.

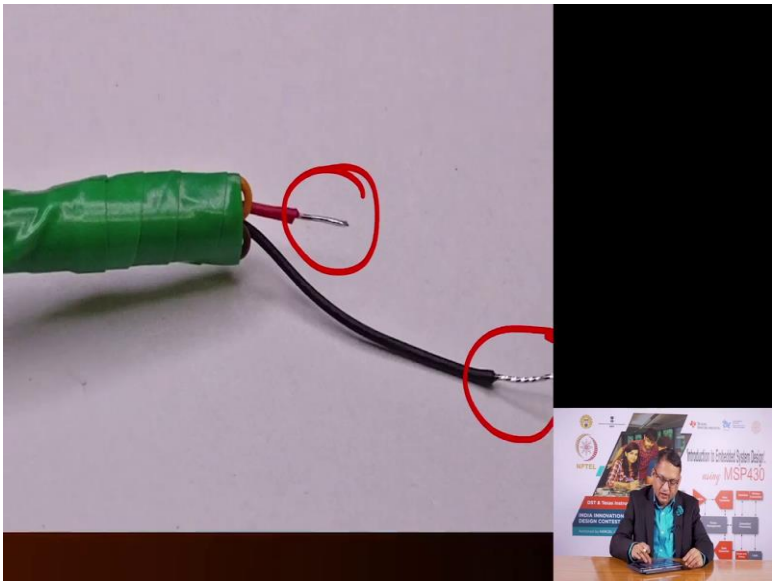
(Refer Slide Time: 22:58)

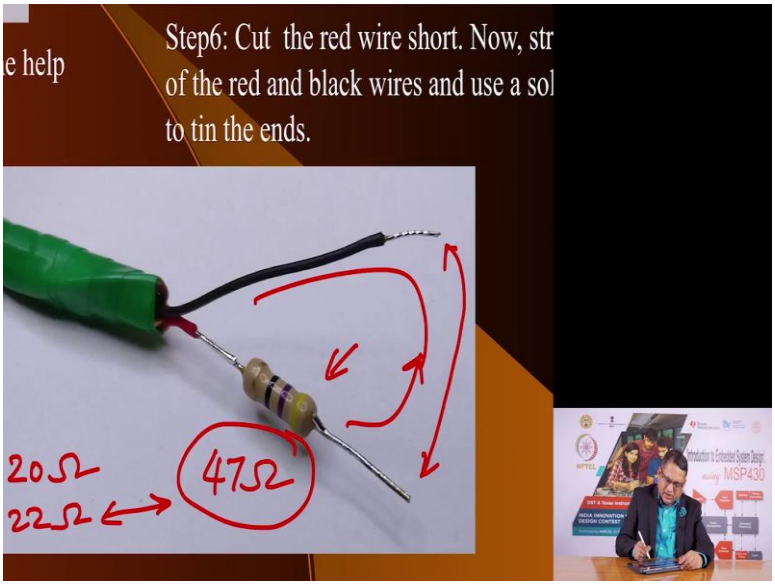



Modifying the USB Cable
Step1: Cut the end (other than the USB Type A male end) of the cable.
Step2: Strip the insulator part of the cable to see five different wires.
Step3: The Red and Black ones are used for the Power supply. Cut the three other wires at different lengths.
Step4: Now, bend the three wires.


5: Tape the three ends with the help


nds with the help
Step6: Cut the red wire short. Now, strip the en
of the red and black wires and use a soldering ir
to tin the ends.





Then leaving the black and red cable you bend the rest of the 3 cables and then you tape them off like we see here. Use a insulating tape. Tape it off. Now, these wires can be cut at different lengths. Why? Because we will see, we do not want them to be of equal length. They may short with each other and they may create problems where the USB interface.

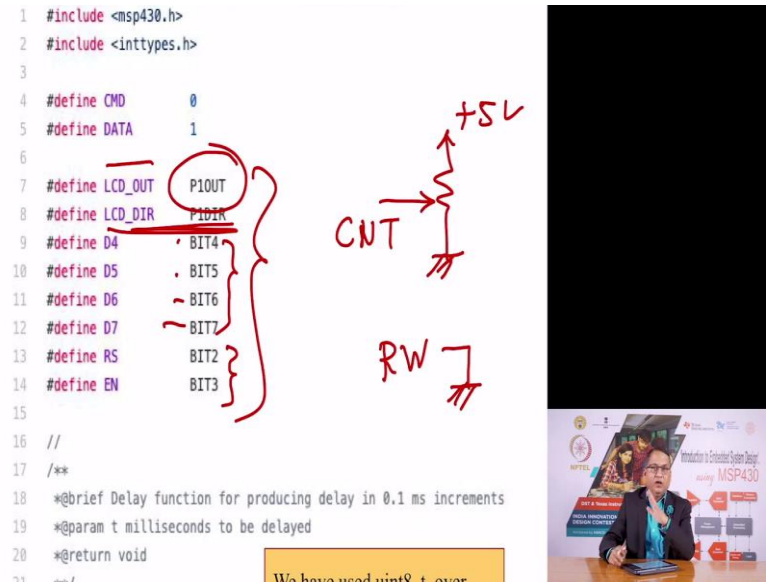
So, you cut them at different levels. Those cut wires you tilt using solder wire. You apply solder onto it so that the breads of the wire are all joined together. Then I suggest you use a resistor on the positive wire.

In this case this is the 47 ohm resistor. You can choose any resistor up to 20 ohms. The reason is if inadvertently you short these 2 pins the resistor will limit the amount

of current that can flow in this circuit and the safe limit for USB normal USBs 500 milli amperes. Till 500 milli amperes it will not damage the USB port. So, we have put a 47 ohm resistor that means we are allowing up to 100 milli amperes of current and this is suitable and sufficient for our LCD.

If you have a lesser value resistor up to 20 ohm that is suitable. So, you can choose a 22 standard ohm up to 47 ohm resistor like this and this once you solder on the 2 wires it provides you 5 volts that you can use safely to power your LCD. Now, here we are looking at the hello LCD code. This is the code we are going to write on our code composer studio. We are going to build and compile it an we are going to download it in our lunch box and we will see here which connections we have made to the LCD.

(Refer Slide Time: 24:57)

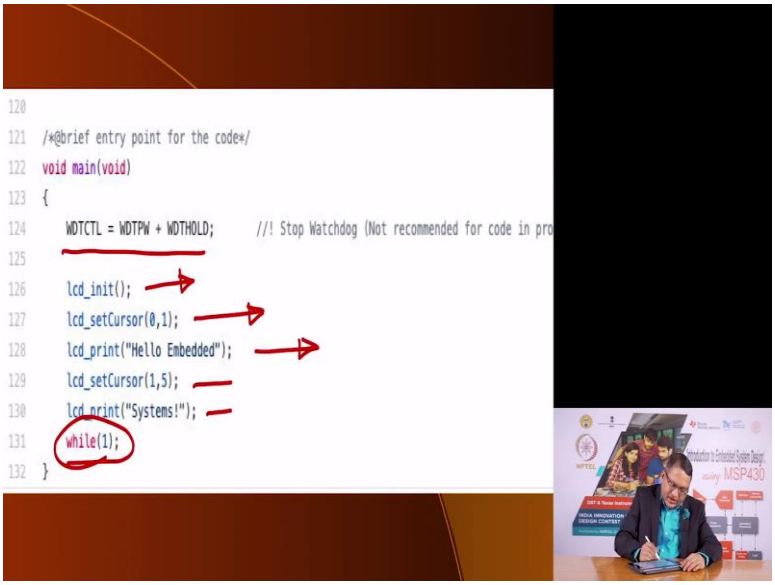


Let us go through those connections. So, these are the pin outs. The LCD output we are going to take out of P1. So, we are taking it P1. The direction is set here that they will all be output and these are the 4 bits of data. As I mentioned we are going to interface the LCD in the 4 bit mode.

So, we have 4 bits - bit 7, 6, 5 and 4 and the resistor select and enable pin are going to be derived from P1.2 and P1.3. Now, we are going to come back here later. Let us go back to our main program. So basically, what you need to do is using this pin out, you need to connect your MSP430 micro-controller to the connections of the LCD that we have seen earlier.

You need to also connect the 5 volt and ground. You need to connect the potentiometer between VCC that is plus 5 volt and ground and the centre of the potentiometer or the pre-set you need to connect to the contrast pin of your LCD. Apart from that these 4 data pins, the register select and enable pin you need to connect to the port 1 pins and the other pin that we mentioned, RW pin you need to connect to ground. Once you make those connections on your LCD which you can insert in your breadboard, you are ready to go. Let us see the main program.

(Refer Slide Time: 26:52)



So, the main program starts here. The main program is actually very simple but that is because most of the functionality has been outsourced to in the form of functions. The first thing that we do the moment we enter into the program is that we disable the watchdog timer. We do not want the watchdog timer on our MSP430 to interrupt us.

Then we call a subroutine to initialize the LCD and then we set the cursor to the home position that is 0, 1 which is the top left pixel is our home position and then we want the LCD to print Hello Embedded on the first line and on the second line we want to print Systems and then we are going to wait here in this, doing nothing. That is the main code. Now, let us go through each of the functions and see what it does. Let us see what LCD in it does.

(Refer Slide Time: 27:39)

So, LCD in it what it does is, it initializes the port 1 pins to be output and it sets the direction as output and it defies them as output. This is what it does and it makes all the pins equal to 0. The register select is 0. The enable is 0 and all the data pins D4 to D7 are set to 0. Then it calls a delay of certain amount which we will see what it does and then it writes a value called LCD write here. Let me erase this. It calls a function called LCD write and in that LCD write there are 2 variables, 2 parameters.

One is number called 33 and then it say this number should be treated as command. Now, you remember the LCD is able to accept either command or data using the register select function RS. If we are sending a command, that value has to be 1, if you are sending a data that value has to be 0, you are basically conveying to the LCD

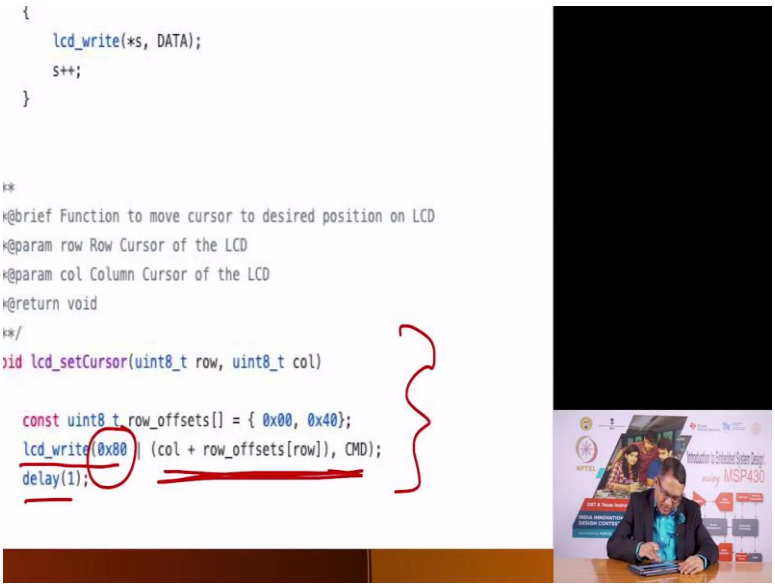
write function that this value, this number 033 0x33x should be sent as a command. Then you are delaying for some more time and you are sending another command with a value 32. Then you are delaying for a lesser period of time.

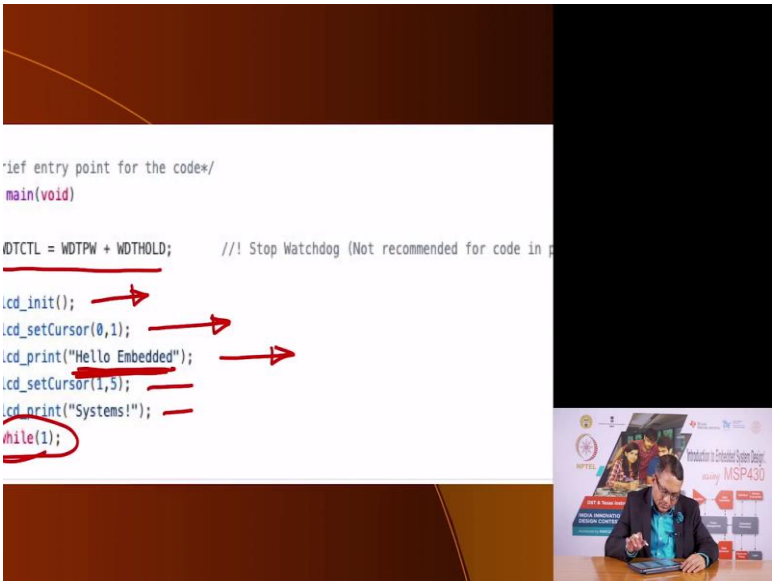
Now, you are sending these 4 commands basically as you can correlate with the commands that you can send to the LCD. This is allowing you to initialize the LCD in 4 bit mode, the 2 lines. You want the cursor display to be on, you want the cursor to be off and you want the cursor not to blink and then you clear the screen using this command.

After some delay you are setting the cursor for auto increment meaning when you write one character, you want the cursor to go to the next character and initially you want to set the cursor to 0,0. This means in your display, this is 1 line, this is your second line, you want it to go here.

This is the initial location. Once you have initialized the LCD you are executing the second command where you are saying set the LCD cursor to 0, 1.

(Refer Slide Time: 29:54)

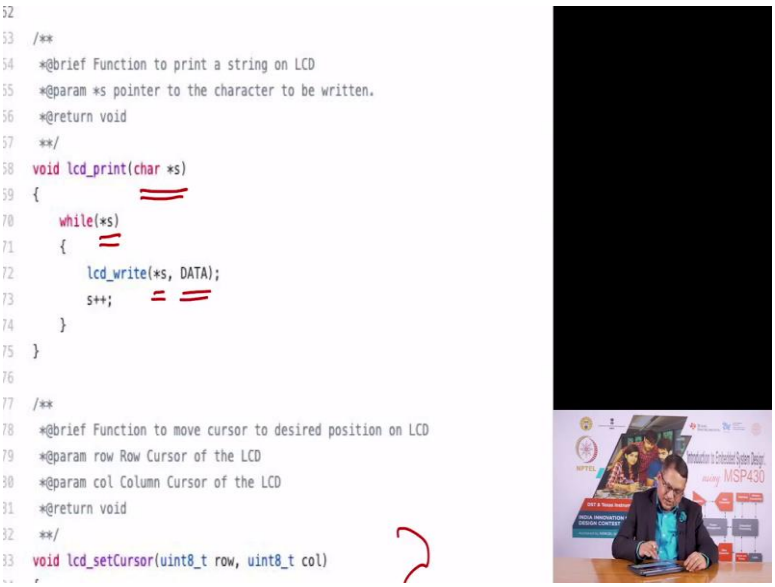


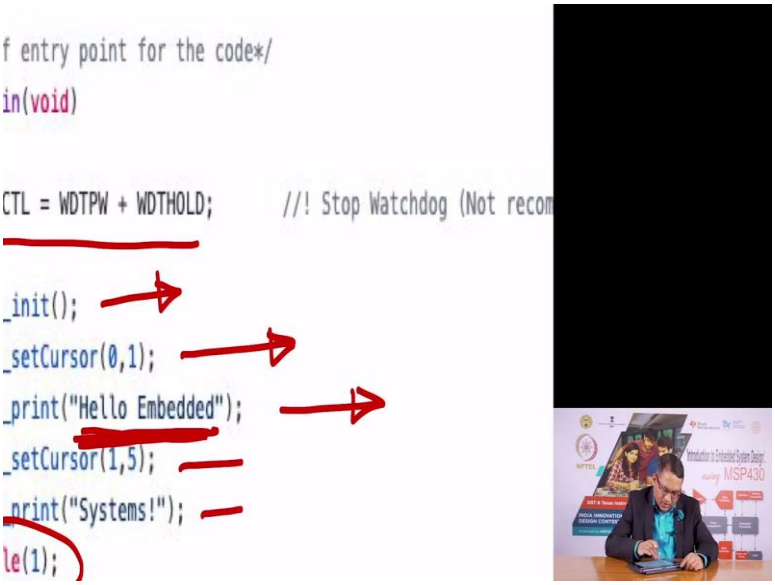


So, let us see what that command is LCD cursor. Here you are sending some information which uses the same LCD write. You are sending this information and you are sending the offset of the values to the LCD and then you are delaying it for some time.

Let us go back to the program. Then you are saying Hello Embedded. You want to print this information. You want to print this characters on to this LCD. Let us go back to the LCD print. What it does?

(Refer Slide Time: 30:29)

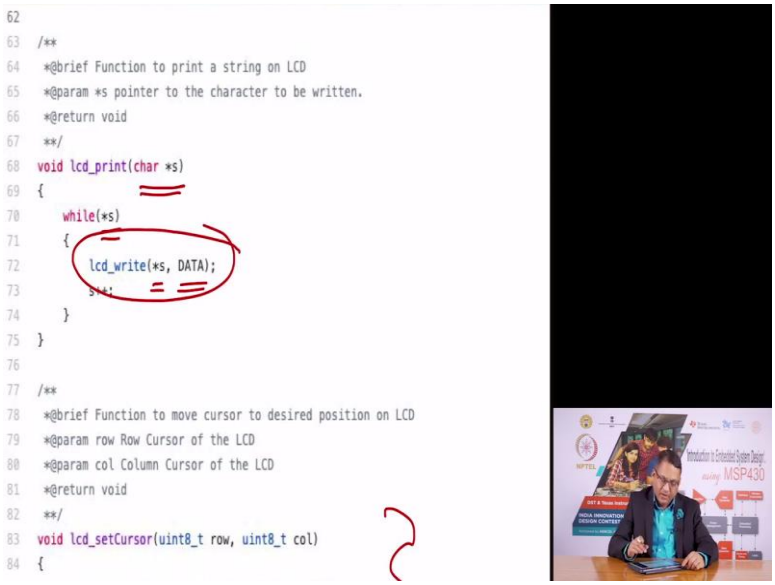




Now you are passing a array to the LCD print and you are passing its pointer. So, till the pointer does not give you a null because when the pointer ends, the value is null. Till the null point, null value is received, you are sending information and you are saying that this information should be treated as data. What does it consist of?

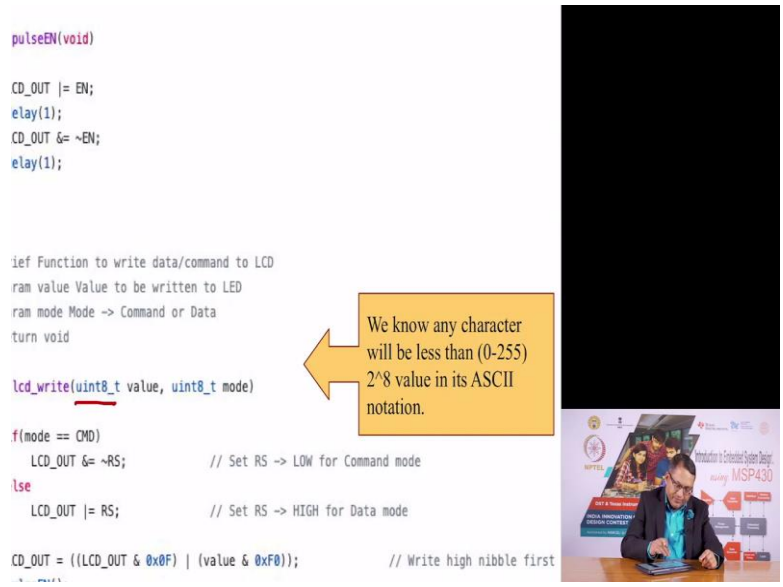
This consists of the characters under the apostrophes Hello Embedded. So, you are sending one byte at a time to the LCD write function here.

(Refer Slide Time: 31:12)



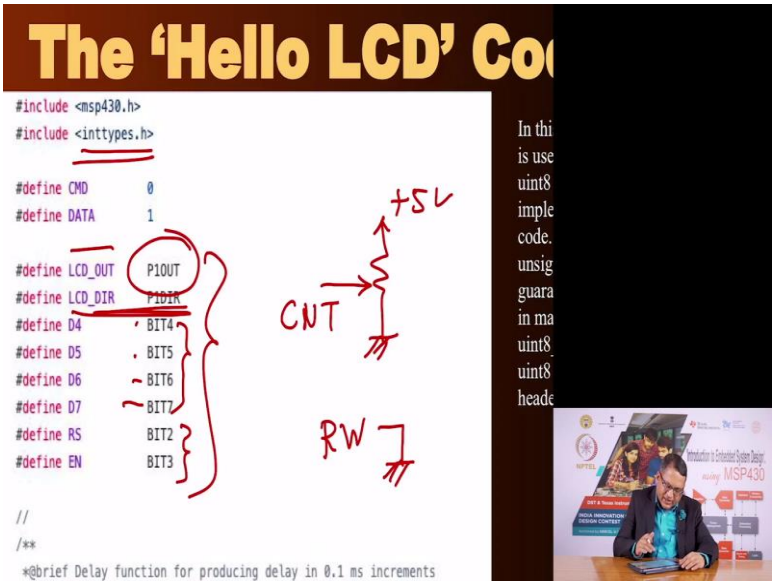
So, let us go and evaluate the LCD write function. Here is the LCD write function here.

(Refer Slide Time: 31:29)



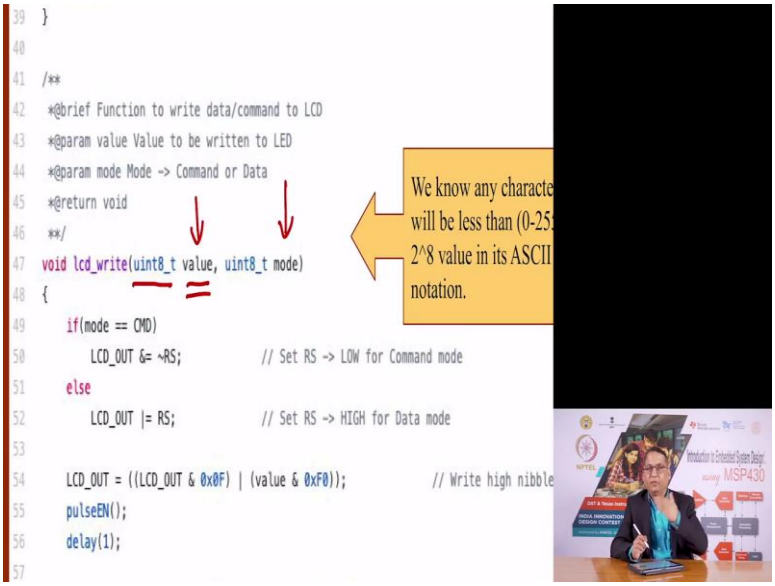
In the LCD write, before that let me mention here that in our program we have chosen to use a variable of the type uint8_t. This means unsigned integer of a size of 8 bits which is not very different from unsigned character but this is not available in your normal MSP430 header file. For that you have to include a different header file and that is this int types.

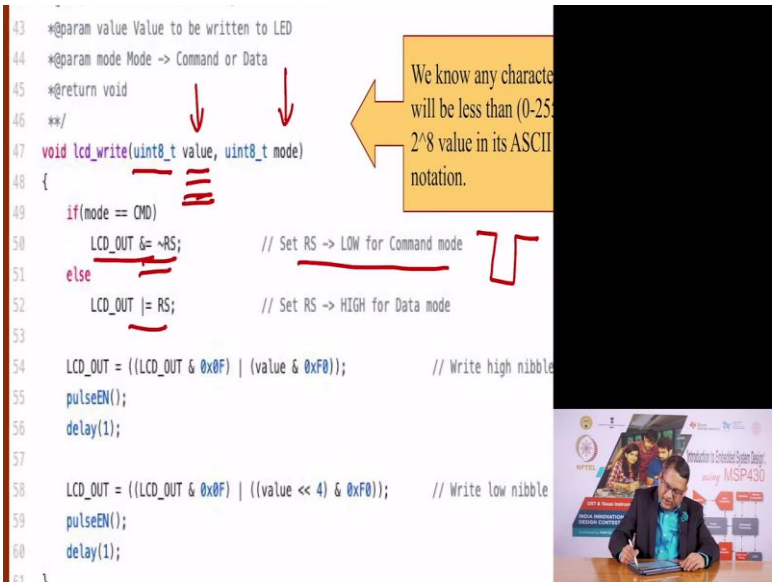
(Refer Slide Time: 31:52)

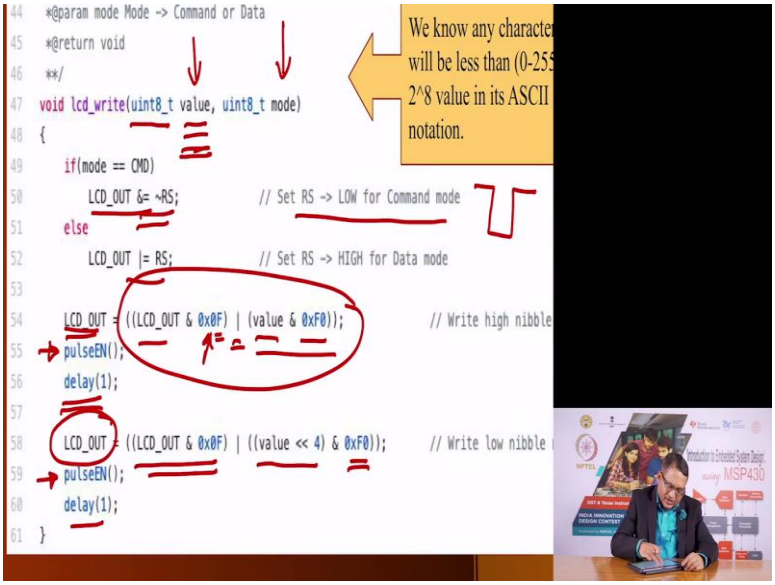


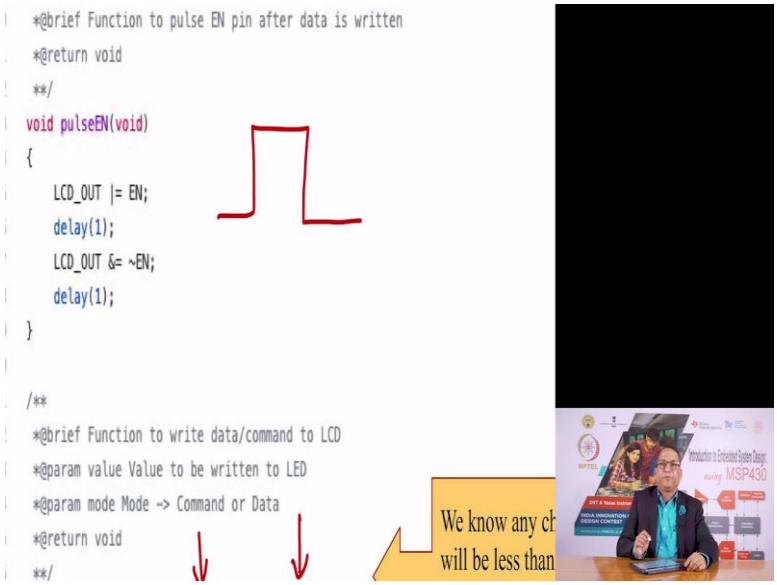
In int types there is a definition of how to invoke uint8 kind of variables. So, we have used that.

(Refer Slide Time: 32:04)









So, LCD write requires 2 values. One is the value and the other is what is the mode meaning in what mode should these values be sent to the LCD. So, this mode can be either command or it can be data and whatever value is there it will be set. But this value will be 8 bits. But as we know we are not able to send 8 bits of data from the controller to the LCD. We are sending 4 bits at a time.

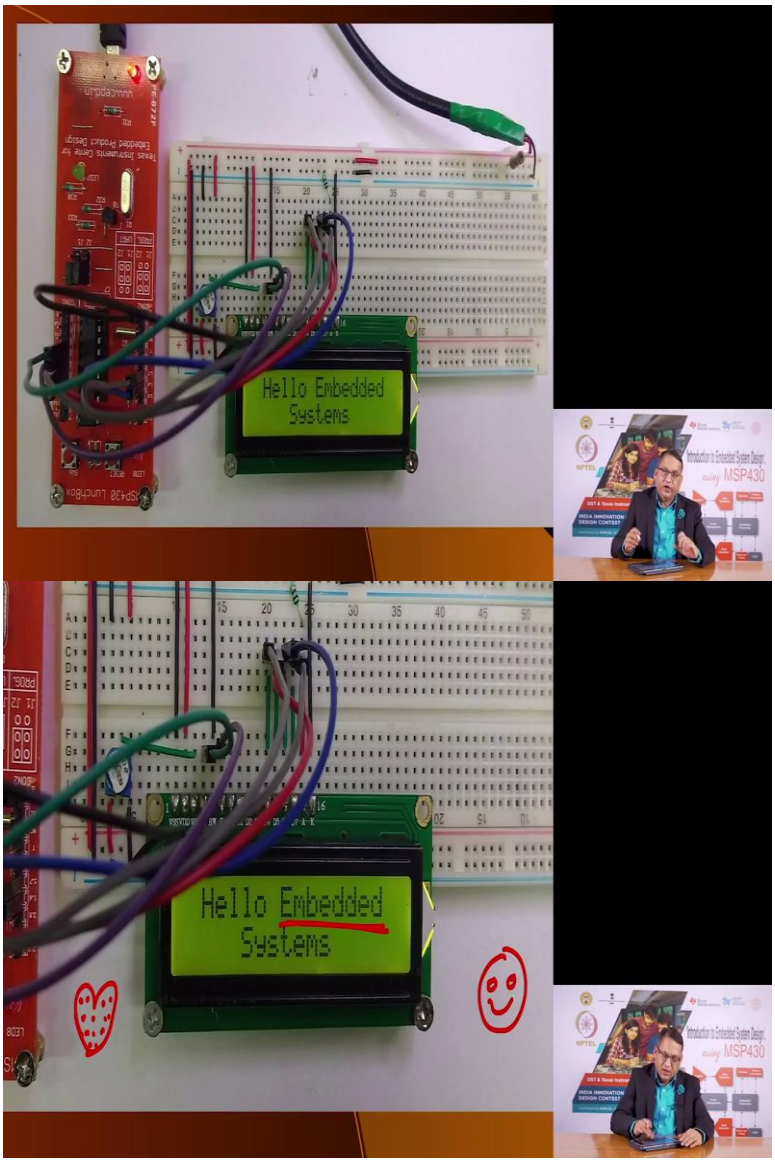
So, this LCD write function will actually split the 8 bits of data that you get here and split it into nibbles and send one at a time. Let us see how it does that. So, first of all it says that it going to generate a low command on RS for command mode. So, it is going to generate a pulse like this, making this using this command. It makes RS 0 and using this makes RS I.

Then it is going to output the 2 values which is there in the value here. You are going to take the upper nibble by ending it with F0 and the original LCD out value you are retaining the lower values but the upper values you have made 0 by ending it with 0F and that value is being odd with this information. And then you provide this value onto the port 1 pins that is LCD out. Then you pulse the EN pin which makes the LCD controller receive this information. Then you delay for some time.

Now, you are still left with sending the lower nibble. The value in the lower nibble is shifted to the upper nibbles and with F0 so that the other bits are 0 and same function you operate here and create a value to be send to port 1 and once you send it to the port 1, you pulse the signal on EN and then you delay and you go back and what does pulse EN does? It does nothing but it creates EN equal to 1 and then it makes 0.

So, these are the sequences of operations that LCD needs to, the controller needs to perform so as to send information to the LCD. So, I recommend that you download this code, build and compile and download it into your MSP430 lunch box.

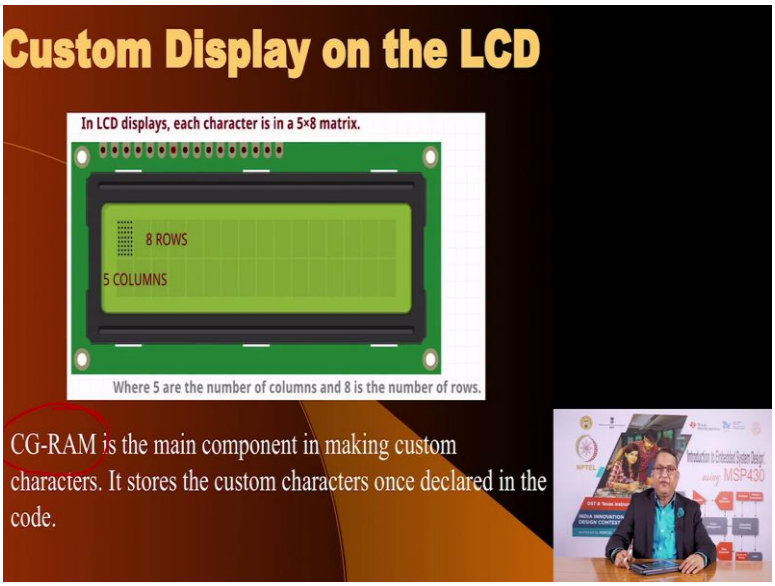
(Refer Slide Time: 34:45)



Make the connections to the LCD as discussed and this is what it would appear that you have made appropriate connections to the LCD and when you run the program you will see that the LCD prints this message. So, this is one part. Now the LCD is capable of creating custom characters also. These are all built in ASCII characters that it is able to print. But what if you wanted to print a smiley?

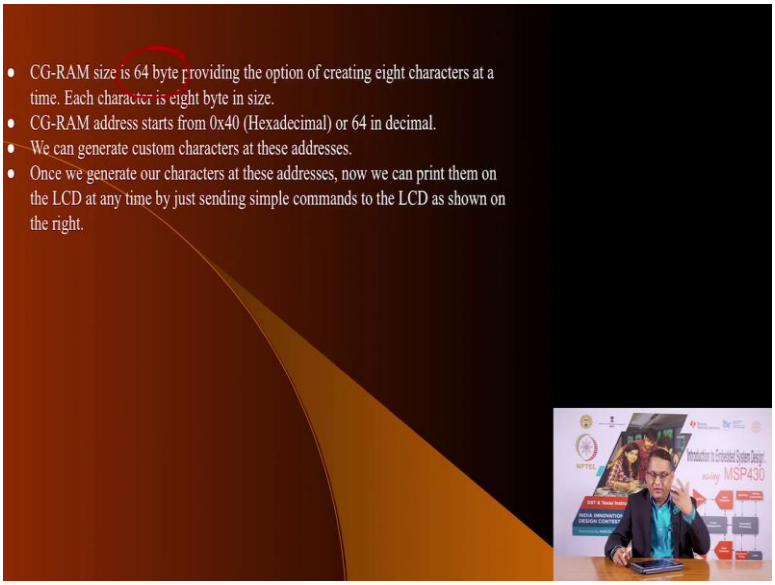
Something like this or a heart. Something like this. It is possible to print the heart like this by creating a pixel character and store this information in the RAM that the LCD has and then invoke them so that you can print these characters.

(Refer Slide Time: 35:32)



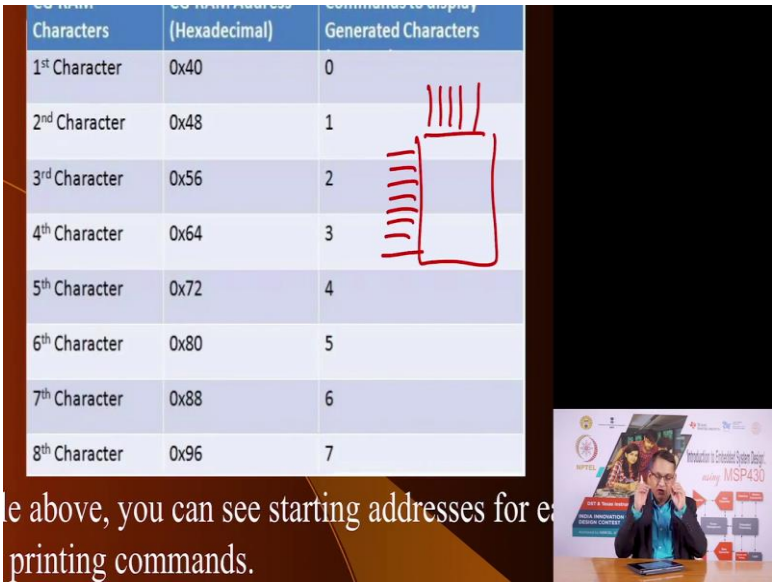
So, the second part of this code we are going to show, how we can create custom characters, store them in the memory of the LCD and then invoke them and tell the LCD controller to display them.

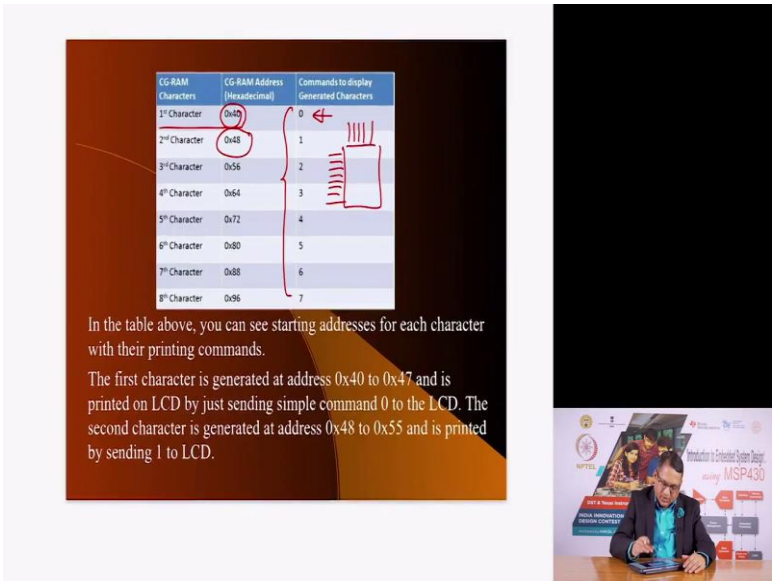
(Refer Slide Time: 35:57)



The information can be stored in a RAM that is available on the LCD that is called custom graphics RAM and that information that size is you are getting 64 bytes for each character you are getting 8 bytes.

(Refer Slide Time: 36:07)

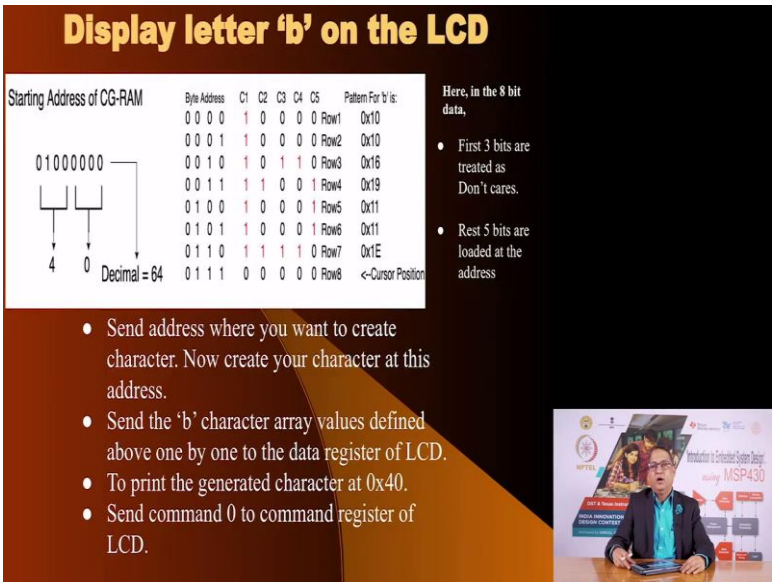




Out of these 8 bytes as you see here you can have 8 characters. Each character requires 8 bytes. Out of these 8 bytes you are ignoring 3 bits because your character is 5 columns and 8 rows. So, you can specify into this 5 by 8 or you can think of it as a 8 by 8 RAM which of the bits have to be 1 and once you write that information you can then invoke them later on and the first character is stored at address 40 and its address is 0. The second character is stored in these are RAM addresses. 40, 48 as you see they are all 8 bytes apart.

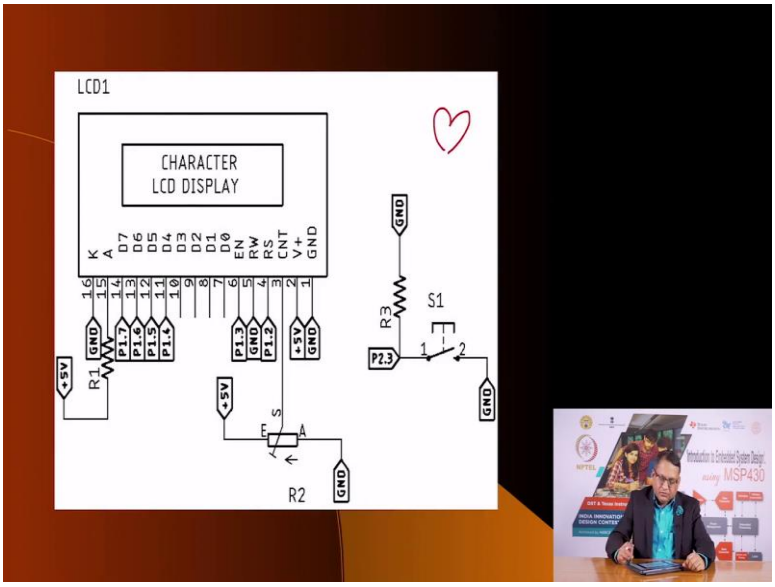
So, you can write information into these RAM locations and once you have written you can invoke them by sending information to the LCD that you want to display the characters stored at address 0 or 1 or 2 and so on.

(Refer Slide Time: 37:07)



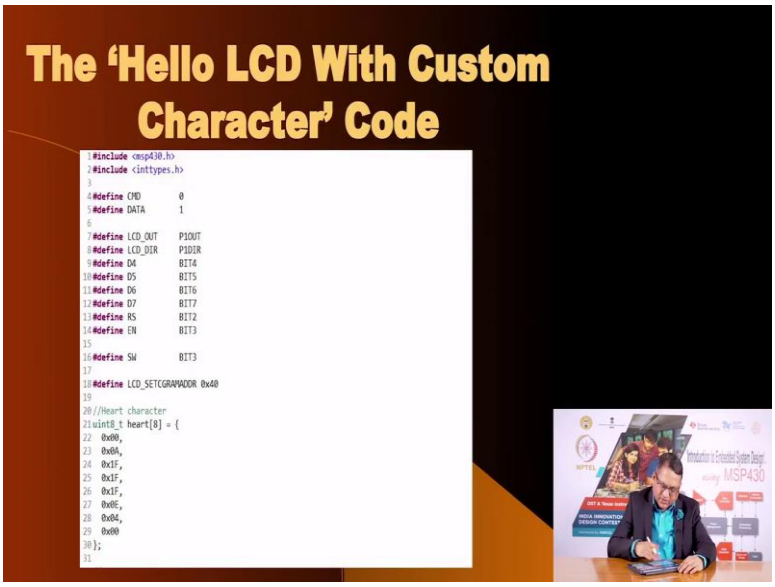
So, I strongly recommend that you go through this code. It is not very different from what we have discussed earlier.

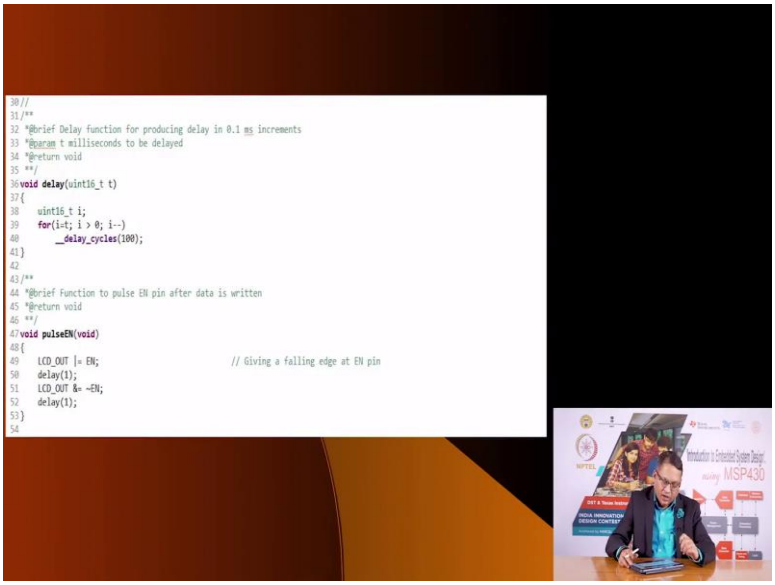
(Refer Slide Time: 37:11)



The only addition is that we have stored custom characters. In fact, we have stored a hearty. We have stored a heart like this in the CG RAM and you can invoke it and display this into your display.

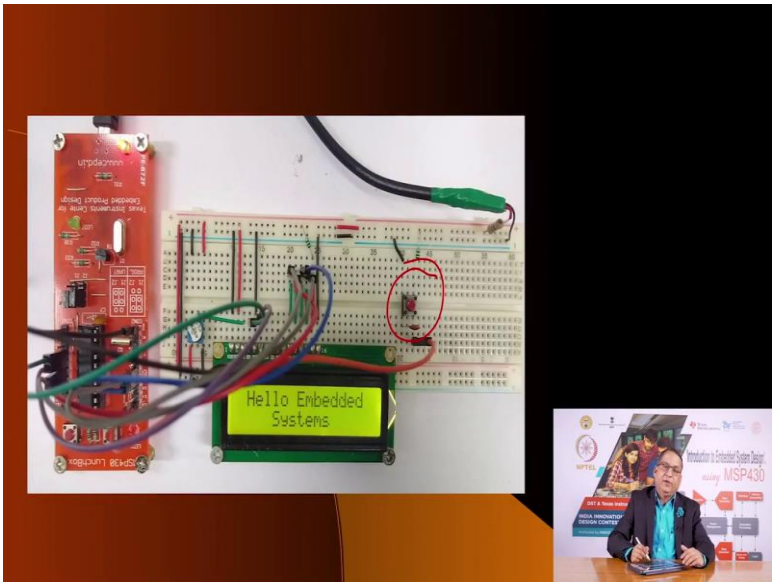
(Refer Slide Time: 37:31)





So, I strongly recommend that you go through this. The codes are very well documented. Reading it will tell you what we are doing.

(Refer Slide Time: 37:43)



And once you connect the connections are not very different. Other than the LCD you only need to connect a switch here. So when you press the switch, when you download the code you will see the same hello embedded systems, when you press the switch you will see that 2 hearts appear around before and after the systems. When you press the switch again they will disappear and so on and so forth.

So, I strongly recommend that you go through that code. That code is not very difficult. That code is not very different from the earlier code. The only addition is to add custom characters and the code is quite readable. I recommend that you go through it and use that template for creating your own custom characters and use it whenever your application demands so. So, with this we are at the end of this lecture where we have illustrated how we can connect character LCDs to your projects using MSP430 and we will see how we use it in our future projects. Thank you. See you.