Introduction to Embedded System Design
Professor. Dhananjay V. Gadre
Netaji Subash University of Technology
Professor. Badri Subudhi
Indian Institute of Technology, Jammu
Lecture No. 15
Physical Interfacing - 1

Hello, welcome to this new session on the as part of the course on Introduction to embedded system design. In this session we are going to look at the physical interfacing of the microcontroller with various input and output devices. As you know, I am Dhananjay Gadre and I am going to conduct this session.

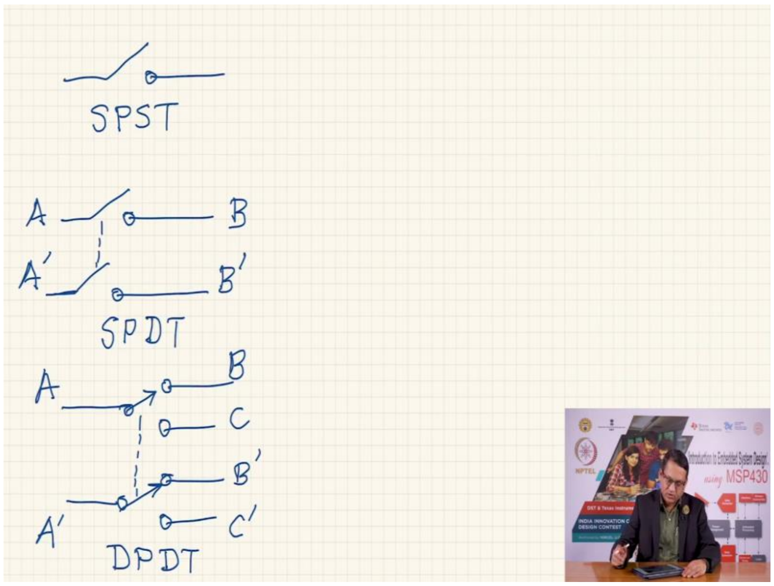
Now, if we go back to the session where we had discussed the Six Block Model one of the important blocks was the input block and input block consists of input devices through which a user would interact with the embedded system. Similarly, the embedded system would know about the environment through various sensors. In this part of this session, we are going to talk about various ways in which a user can interact, user can provide input to the microcontroller and therefore, to the embedded system.

(Refer Slide Time: 01:23)

So, what kind of Input devices we have? Well, we can use a Push Button. This one we also have Toggle Switches, switches which have two states on and off. Push Buttons also have two states except you have to keep that switch pressed to change the state of the switch. Whereas in a toggle switch, you can leave it in that position and it will remain either open or close.

Similarly, we have varieties of Toggle switches like Single Pole, Single Throw, Single Pole Double Throw, Double Pole Double Throw and we could generalize this to mean multiple pole multiple throw just to illustrate this idea what is the meaning of.

(Refer Slide Time: 02:11)

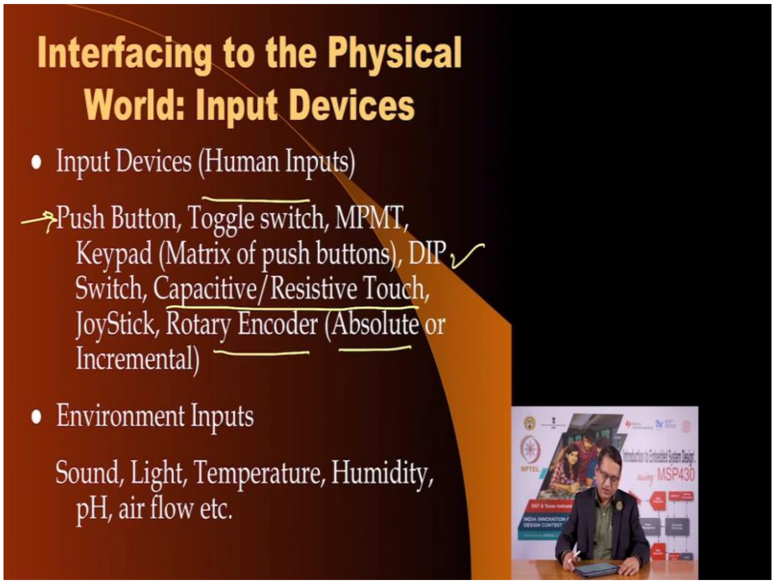


So, when I have the when I use the word single pole single throw, that means, this is a single pole, single throw. If I say single pole double throw, this is, this is SPST single pole single throw, a single pole double throw would be this and there would be two switches which will operate in parallel and to indicate that, that they are mechanically coupled, I will draw this dashed line and this would be single pole, double throw.

And I could then extend it to single double pole double throw. Now, I would have two poles and these are again mechanically coupled, which means if so, if this is A and B, I would call this A bar, A dash and B dash. So, here I can see A, B and C here it will be A dash, B dash and C dash. So, at if I look at SPDT if the switch is open, that means A and B are disconnected, then A bar and A dash and B dash will also be disconnected.

The third switch is DP double pole double throw in this if A is connected to be in one pole. Then on the second pole, A bar will be connected to B bar. If I move the switch, if I toggle the switch and A is connected to C it will be disconnected from B at that time A dash will be connected to C dash disconnecting itself from B dash. So, we have all these kinds of mechanical switches. Let us go back.

(Refer Slide Time: 04:13)

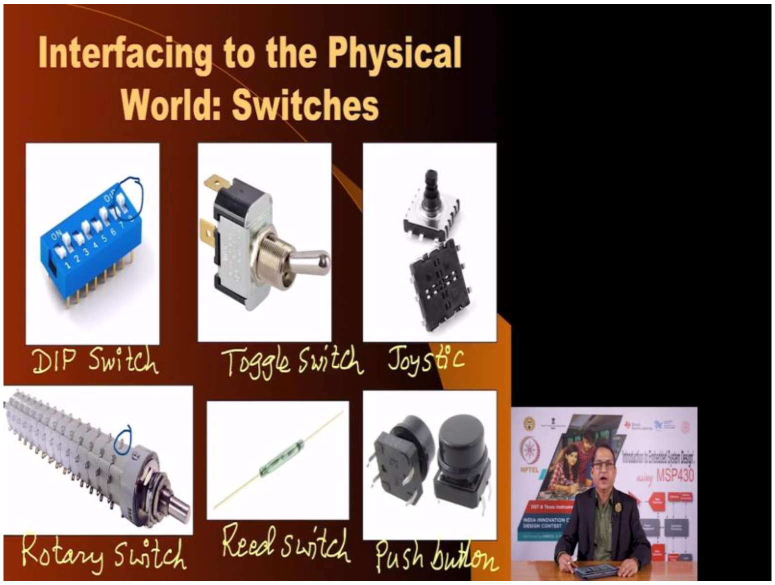


Similarly, we have a DIP switch, I will momentarily show you what is a DIP switch, we have some pictures we may interact, humans may interact with the embedded systems through Capacitive and Resistive touch. Through, a Joystick which is nothing but a combination of switches. So that it gives you up, down, left, right and maybe center position which you can engage.

There are Rotary Encoders, you may have seen especially in your oscilloscopes there are no Rotary Encoders with which you select the voltage setting or the time setting. Now, did you realize that they seem to you can move them in any direction and there does not seem to be any start or end. Similarly in audio systems, there is a dial for changing the volume or selecting a song, they do not seem to have any start or end.

So, these are examples of Rotary Encoders and usually these are the cheapeR1s are what are called as Incremental Rotary, Rotary Encoders although there is also an absolute variety of Rotary Encoders and they are quite expensive. And then apart from these Human Inputs and Environmental Inputs, such as for Sound, Light, Temperature, all the entire list that we have seen, we will consider them later.

(Refer Slide Time: 05:42)



For now, let us see what are these Switches that we are talking of? So, here is a list of picture of some Switches. This is a DIP Switch. Why is it called a DIP switch because this can be, because the footprint of this switch is like a dual inline package, a common integrated circuit package, you can insert this into a breadboard or into your circuit, solder it and then these switches are so tiny that here in this case there are 8 switches.

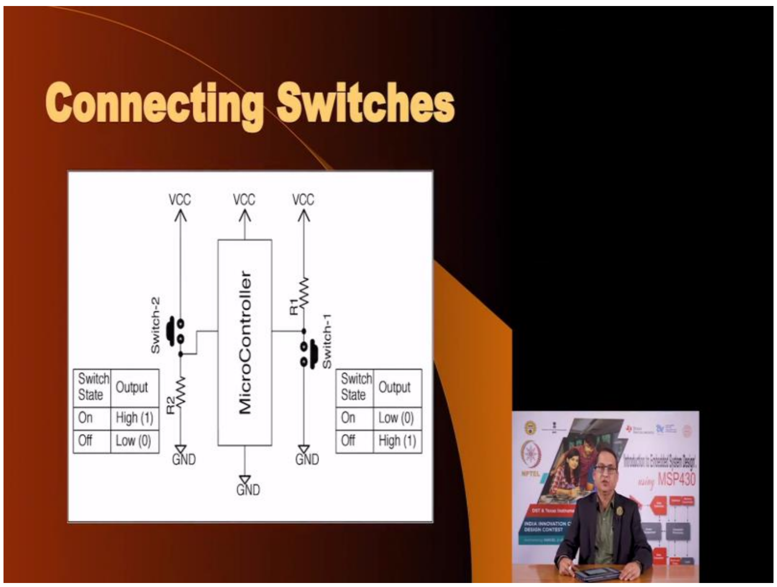
You can turn them on and off using tip of a ballpoint pen because they may be too tiny for you to engage and operate with your fingers. Then here in the second case is the Toggle Switch. This is a Toggle Switch. The third one this is a Joystick, the fourth example here, this is a Rotary Switch and as you can see, it has multiple poles and multiple throws so that you can probably move it around and it would make connections with all these lugs that you see in the picture here, these are the connections and so on.

And here the next picture is that of a Reed Switch. A Reed switch is cannot be operated just by pressing it. This is enclosed in a glass tubing and to operate it you must bring a magnetic field near it. So, you could probably hold a magnet in your hand and if you bring that magnet near the switch, the contacts of the switch will close. If you remove the magnet the contact of the switch will be opened.

And in the end here we have normal Push Button The most common input device that you would be using and connecting to your embedded system. Now, let us see how do we connect these various switches to our microcontroller, we have to realize that none of these switches provide any voltage. These are passive devices, which means they do not have a concept of high or low voltage.

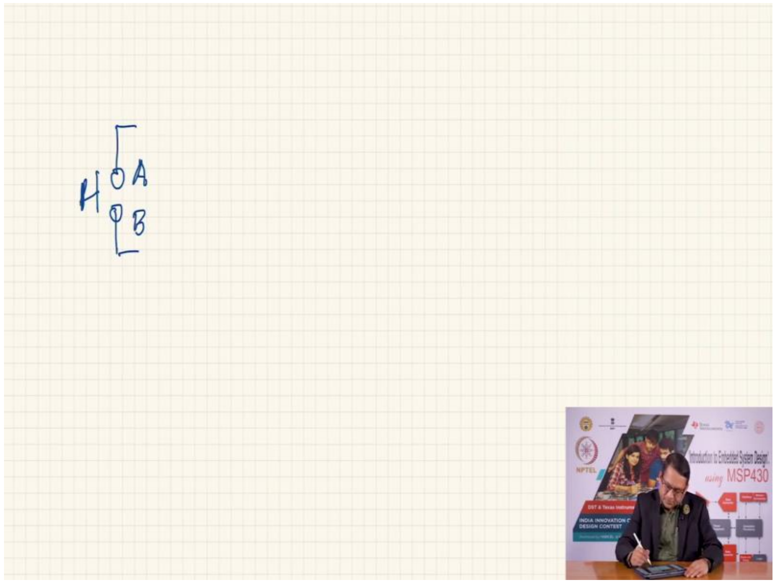
They have a concept of the state of the switch, the switch would be either open or it would be closed and it would require some external circuit connecting it to some voltage sources or maybe to ground to convert the open and close state of the switch into logic 1 and logic 0. Let us see how to do that.

(Refer Slide Time: 08:57)



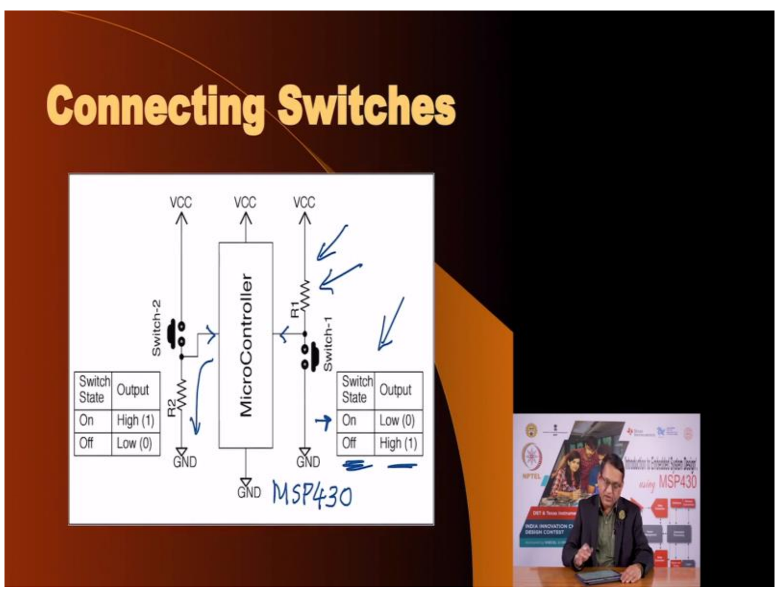
Here is a picture of it actually shown two ways to connect a push button these both these switches Switch-1 and Switch-2 are Push Button.

(Refer Slide Time: 09:10)



Nor the normal state of the I would draw the switch like this. Now, if this switch is pressed, it would short the contacts let us say A and B and if the switch is open the A and B connections would be open. And so if we connect it, we can transform we can translate the two states of the switch open and close into whatever configuration we want.

(Refer Slide Time: 09:40)



So on the left side, on the right side, you see for Switch-1 we have chosen that if the switch state is on meaning if the switch is pressed, it would I want to get an output of logic 0 when the switch is off, meaning I have not pressed the switch. I should get a logic 1, to translate to these two states of the switch we have connected external resistor this R1 and the junction of the resistor and the upper terminal of the switch is connected to input pin in this case of microcontroller.

This could be MSP430 we have considered we have started talking about MSV430 in previous lectures. So, this could be one of the port pins, there is another way to connect the switch and that is illustrated on the left side of this diagram. Now, we want to vary the output and have a different relationship between the on and off state of the switch to logic 1 and logic 0.

And now we want that when the switch is pressed, I want a logic 1 and when the switch is released, that is when the switch is not pressed, I want logic 0. Now, instead of connecting the resistor to the positive supply voltage as you see here, I have connected it to ground and the terminal the junction of the upper pin of the resistor and the lower pin of the switch are connected and this is connected to port pin.

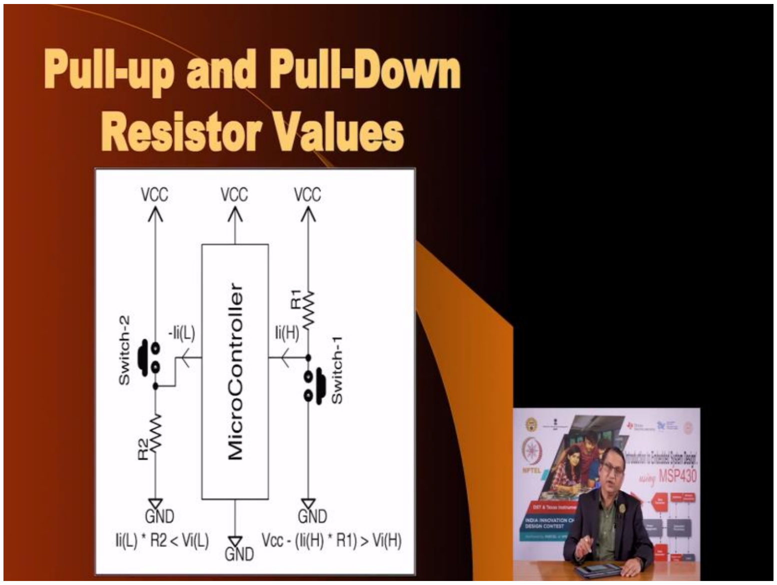
Now, what will happen when the switch is not pressed the port pin will see a logic 0 that is what we wanted when the switch is not pressed when the switch is off we want logic 0 when the switch is pressed, it connects the input pin to the VCC and you get logic 1. If you consider both these options on the right side when the switch is not pressed, the resistor R1 pulls up the input voltage to VCC and so you get 1 that is when the switch is off you get logic 1 here when the switch is off you get logic 1.

When I press the switch that input terminal input pin is now connected to ground and so you get this state. So, it will depend on you. Where you whether you want to connect the switch in the Pull Down, using the Pull Down resistor configuration, or Pull Up resistor configuration.

In many microcontrollers, they have a facility to in where the resistor whether it is a pull up resistor or pull down resistor, this is available on the port pin and we can engage whichever resistor we want, we want a pull up resistor or pull down resistor so that you do not have to connect external resistors like we have connected here. Although you may still connect external resistor just to be on the safe side.

What is the value? What are these values of these resistors it is very important you cannot just arbitrarily connect any value.

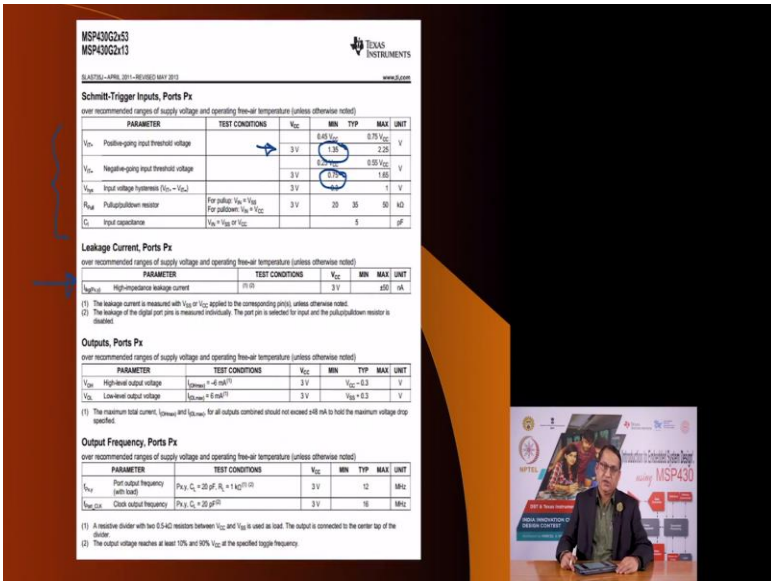
(Refer Slide Time: 12:46)



So here in this diagram, I am going to illustrate how to calculate the value of R1 or R2 in this case, pull up or pull down resistors and both these values of resistance may not be same, and I am going to take an example from the datasheet of MSP430 but we could consider other examples also so that you are well equipped to choose the values of the pull, pull up and pull down resistors whatever be the situation.

So, let us see the datasheet of MSP430 the relevant portions of the datasheet so that we can estimate the value of the pull up and pull down resistors.

(Refer Slide Time: 13:31)

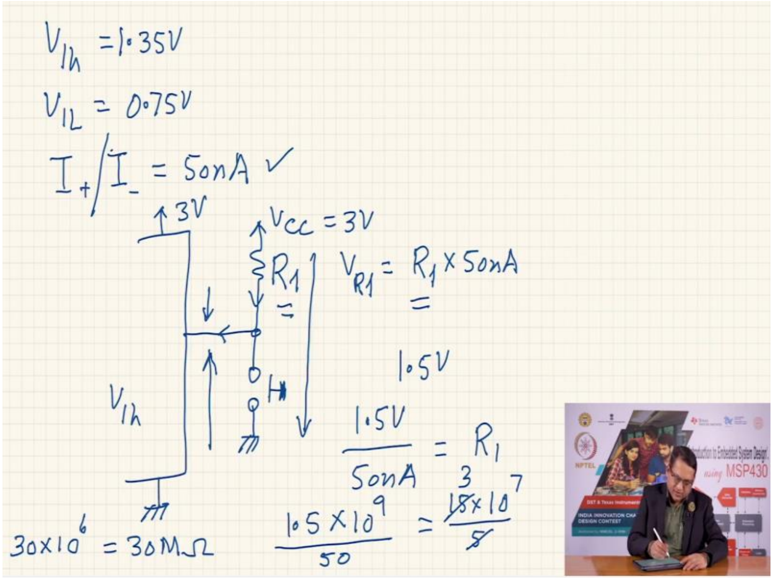


So, here is the here is a page from the datasheet. Now if you come to this part here, this talks about the logic thresholds that an MSP430 recognizes and it has options for 3 volts and we are going to use that. So if you see if the VCC is 3 volts like here, then the minimum voltage for input to be considered as logic 1 is, you see here 1.35 volts and a minimum maximum voltage when the input is to be treated as logic 0 for the same 3 volt is 0.75 volts.

So, we are going to utilize these numbers in our calculation. Similarly, when the input pin does not have a pull up or pull down resistor because we are considering here how to estimate the value we are going to program the MSB430 so, that neither the pull up resistor or the pull down resistor is engaged but still that input pin will have a current which will flow out of the pin, if the pin is grounded and if the pin is connected to VCC it will have a current going in.

And for that you see here the leakage current of a board pin is listed to be 50 nano amperes. So, we have now let us go back here, let me add a sheet.

(Refer Slide Time: 15:08)



So we have ViH, the value is 1.35 volts 1.35 volts, ViL is 0.75 volts 0.75 volts and the current I plus as well as I minus that is the current that goes into the pin or comes out to the pin is listed to be 50 nano ampere. Let us go back to this diagram. Now, let us consider the first case of Switch-1 where we have a pull up resistor. Now, here is my microcontroller. I have connected a pull up resistor and here I have connected my switch and the switch is grounded.

I will connect it to VCC since we considered a VCC of 3 volts, let me write here that VCC is 3 volt, this device is also powered with a 3 volt source 3 volt. Now, we want to estimate the value of R1 that is what is listed here. Now, when the switch is not pressed, because there is a pull up resistor, positive current will go into the pin and this current from the datasheet we have estimated to be 15 nano amperes.

And therefore, this current has to come from the supply it will lead to a voltage drop across R1 which is VR1 will be equal to the value of R1 into 50 nano ampere. Now, how much do we want this voltage to be obviously, we want this voltage to be as small as possible, but that would mean that the value of R1 has to Very small, if that value is very small, how is it going to impact the overall design that when the switch is pressed, the small value of the resistor will allow a large amount of current to go into the ground from the power supply.

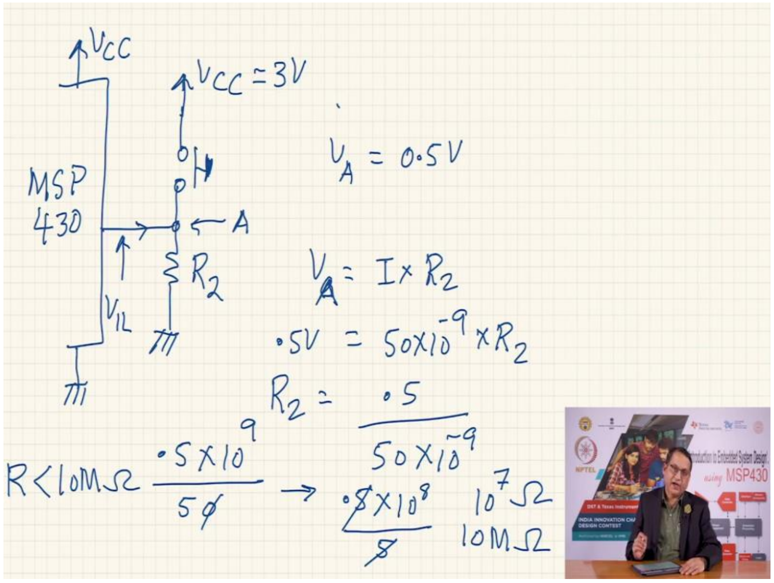
And if it is battery driven, it will discharge the battery faster. So, we would like to keep the value of R1 as high as possible, so as to minimize the current going into the ground determine when the switch is pressed, but it should not be so high that it leads to when the switch is not pressed. We want the input to be treated as ViH. So, we want the voltage at the input to be more than 1.35 volts.

Let us say that we want the voltage to be 1.5 volts because 1.5 volts is certainly more than the threshold. So, it allows me a margin of 1.5 volts on R1 and it will still leave at the input I will still have 1.5 volts which will be recognized as logic 1 and therefore, we have estimated from this that VR1 can be 1.5 volts therefore, 1.5 divided by 50 nano ampere is going to be the value of R1.

So, if we see this it is 1.5 that nano comes up it becomes into 10 raise to power 9 divided by 50. So, this will be 10 raise to power 7, 15 into 10 raise to power 7 divided by 5 and that is 3 and that therefore, the value of the register is 30 can be 30 into 10 raise to power 6, which is 30 megohm which is a very high value of resistor and in reality you can keep this value to be less than that maybe a megohm or something like that, that would still give a lot of current allow for a lot of current and it will lead to a slightly less voltage drop across R1 which is a good thing.

But, using these calculations we have estimated the upper limit the resistor value of r1 should never be more than 30 megohm it can be much less and probably a megohm or half A megohm 500 kilo ohm could be a appropriate value of the resistor. Let us now consider what will be the value of the pull down resistor if we want to use that.

(Refer Slide Time: 19:43)



So, our configuration is that we have the switch between VCC here is my switch and this is connected through a resistor to ground. This is connected to VCC here which is 3 volts this is your MSP430 just so that we are sure my resistor values now R2. Now, when the value when the switch is not pressed I would want the input here to be a voltage which is ViL or less we have estimated previously that ViL 0.75 volts.

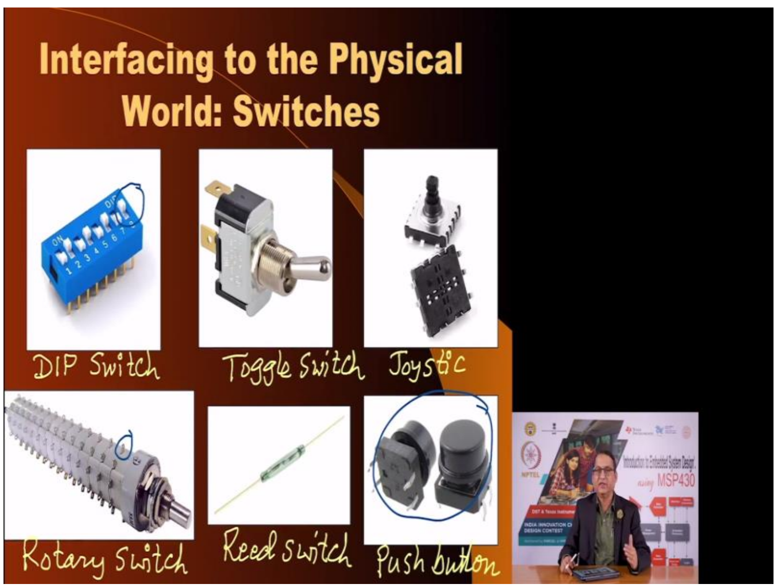
So, we would want the voltage at this terminal to be much less than ViL that is 0.75. Let us say we want the voltage at this junction let call it A. VA we want it to be say 0.5 volts and the current now because of the resistor R2 being grounded, the input current from the pin is going to flow out the amount of the value of this current is 50 nano amperes as estimated from the datasheet, and therefore 50 into 10 raise to the power 9 into R2, this is the amount of I am, this is let me write this again.

VA is equal to the current I into R2, I is 50 into 10 raised to the power minus 9 amperes into R2 VA we have estimated to be 0.5 volts and therefore the value of R2 is equal 2.5 divided by 50 into 10 raise to power minus 9, which becomes 0.5 into 10 raise to power 9 divided by 50. This cancels one of the 0s, this leads to 0.5 into 10 raise to power 8 divided by 5 and I cancel this. So, it is 0.1 into 10 raise to power 8. So, that is 10 raised to the power 7 ohms and that can be, that would be 10 megohm.

So, the value of resistor R2, R2 should be anything less than 10 megohm. So, I could again use a 4 1 megohm or 470 kilo ohm resistor, just because I am going to use that value for the pull up resistor. So, I could use an identical value it would lead to a voltage drop which is much smaller which will be lesser than 0.5 volts and therefore, I would be able to provide I would have translated the two states of the inputs that is on and off state of the input switch into logic 0 and logic 1.

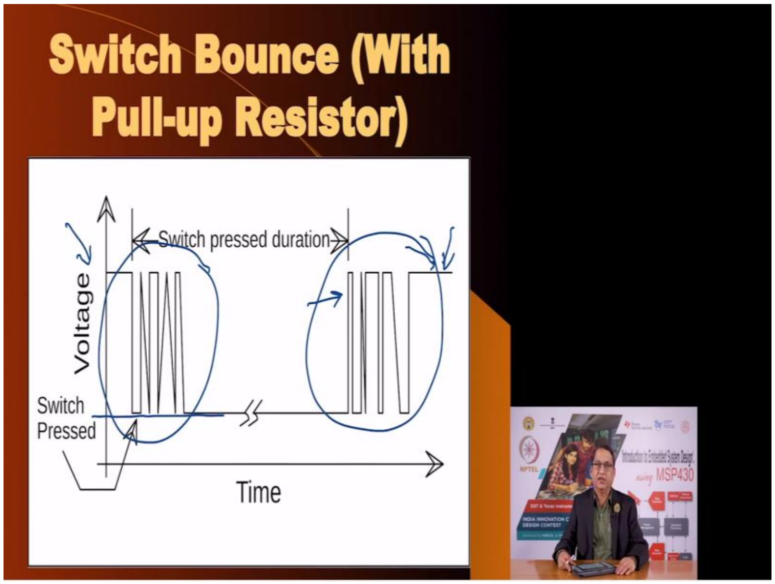
Well above the high threshold or low threshold as the case may be. And so, this is the way to calculate. In reality, if you are going to use MSP430 you can engage the internal pull up or pull down resistors and you may not use R1 and R2 at all, but as a electrical engineer, somebody who is designing embedded systems, one should know how to calculate the values of pull up and pull down resistors and this exercise I hope you have learned that.

(Refer Slide Time: 23:47)



Let us go to the next aspect. Now, the problem with Mechanical Switch such as such as this meaning this is that when you press the switch, it does not change the state instantly. In fact it bounces between the two states for a short duration of time meaning it is not completely on or off it is on off on off before settling to on state if you have pressed it and when you release again, it oscillates between these two states before settling to the off state.

(Refer Slide Time: 24:18)

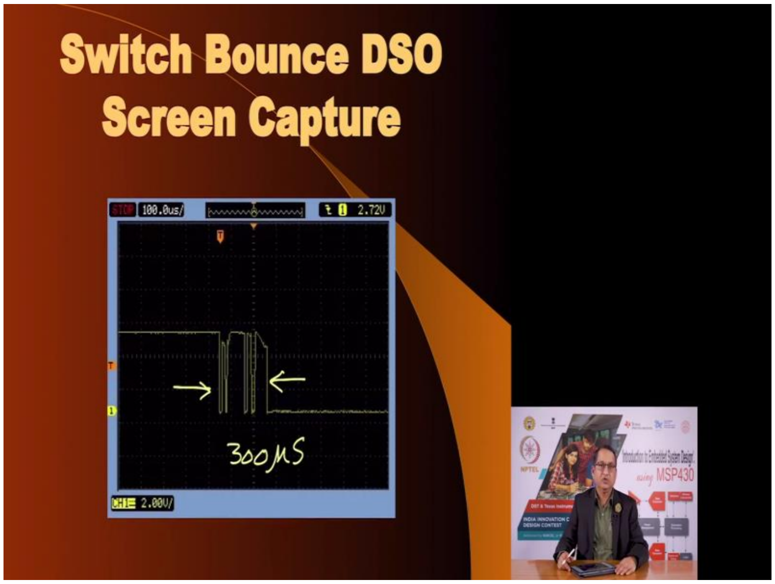


And this feature of a Mechanical Switch is called Switch Bounce. We are illustrated here that if we use a pull up resistor with a switch, then the voltage of the, at the output of that configuration would be high. This is what you are seeing here. But the moment the switch is pressed it is expected to go low. Let us say this will be the 0 volts but it does not settle to 0 instantaneously. In fact, it oscillates for some time.

Here is the time that it is actually bouncing. And then you would have pressed the switch for a certain duration of time when you release again, you have released at this point, but it oscillates for some time bounces between logic 1 and logic 0 before settling to the high state. Now, obviously, we must know, what is the duration of this bounce?

Because if this bounce the duration of this bounce is comparable to the execution rate of the microcontroller, then it each switch press will be treated as multiple switch presses and that is certainly not very desirable.

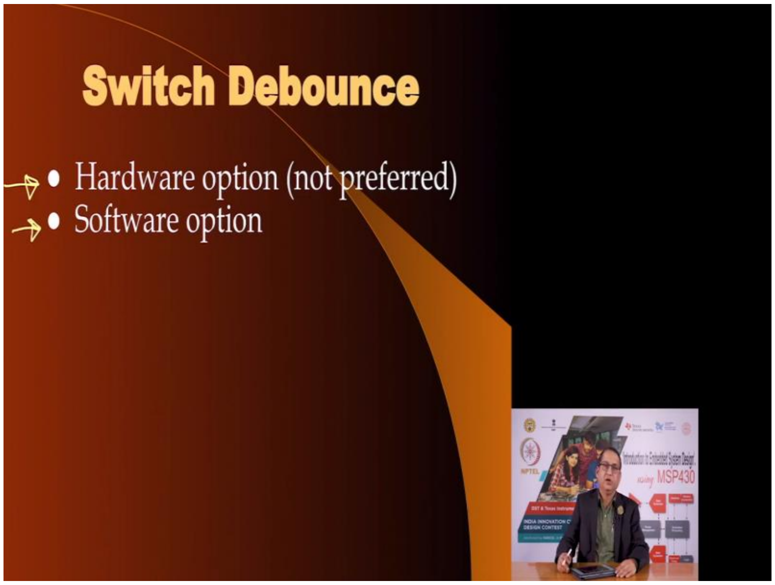
(Refer Slide Time: 25:25)



And so, here is a physical capture of a logic digital storage oscilloscope with the same pulled up resistor and you see hundred microseconds we see roughly 1, 2, 3 we have roughly if you see this as you see this, this is roughly I would say about 300 microsecond duration and certainly a microcontroller would be executing will be able to execute several instructions in these 300 microseconds

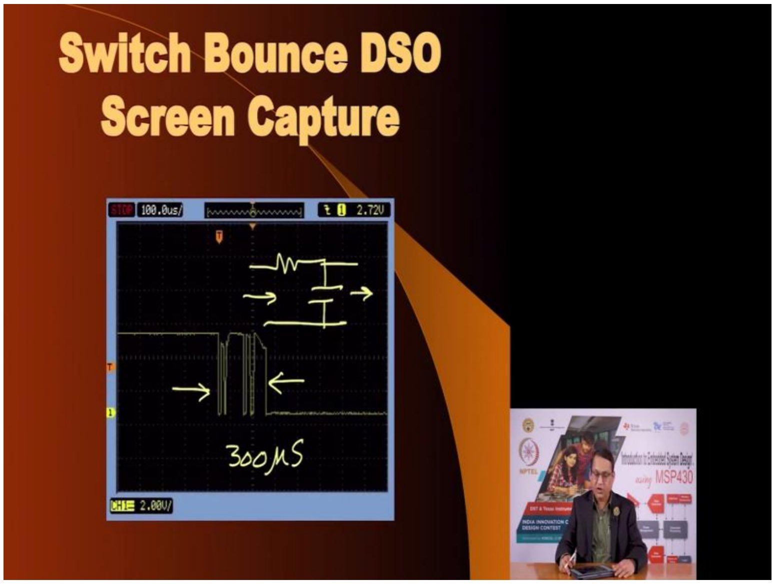
And therefore, this is not a good way to connect a switch directly to the to the microcontroller pin it would be considered when you press one switch it would be treated as the switch has been pressed multiple times even if you have pressed it once and therefore, this bounce has to be removed this activity of removing the bounce of a switch is called Switch Bounce.

(Refer Slide Time: 26:21)



And we must is called debounce and we must consider we must have some mechanism to debounce the switch. There are two options, one is Hardware debounce and the other is a Software debounce.

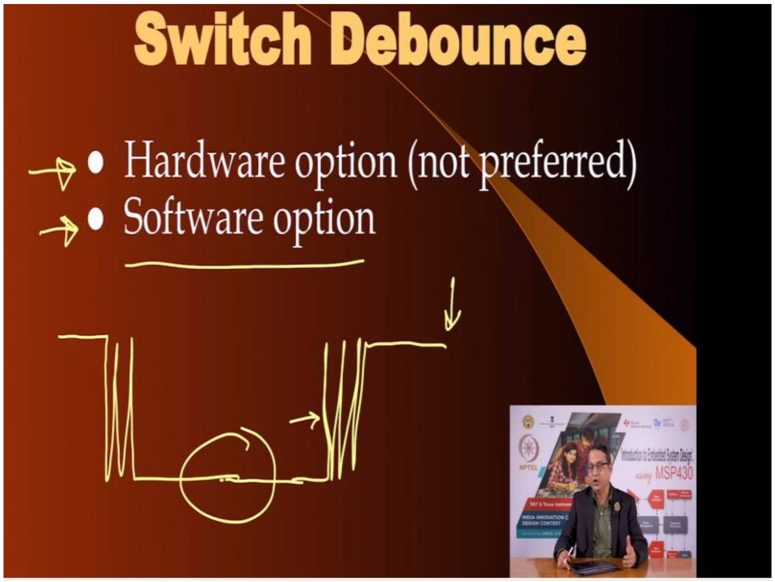
(Refer Slide Time: 26:37)



Hardware debounce would, because the these oscillations are high frequency in nature, you could apply a low pass filter and you can filter out this can be treated as noise and so, a high frequency noise and you can filter it out using a low pass filter such that, such as this you could probably this is a low pass filter this input from here and here is the output this, this filter configuration will remove this noise.

But the objective in the embedded system design is to minimize external components. So, I would not like to you would not like to engage more components because that will take a PCB space it will take up external components which will increase the cost.

(Refer Slide Time: 27:19)



So, are there other methods of D bouncing the switch? Yes, there is a method called Software debounce and what do you do in a Software debounce that when the switch, you press and you expect the switch to bounce the moment the first time you notice that the switch has been pressed, you can execute a delay sub routine and that delay sub routine can be little more than the expected debounce.

So, if we have estimated in the previous DSO capture that the bounce happened for about 300 microseconds I could easily write a delay sub routine of one millisecond that is 1000 microseconds and I will execute this delay when I come back, it would bring me somewhere here and so, this would be very nice. And now again when the, I would wait for the switch to be pressed again.

When the switch is pressed at that point of time, you would the, the output would again bounce before settling to logic 1. And the moment the first time I detect the switch has been released, I would again call this one millisecond sub routine and when I come back I would be somewhere here, which would have where the, the bounce phenomena would have finished and I would have physically I would have debounce the switch using software without any external components, which will keep the cost down.

Of course, with a little more overhead software overhead, but this is what we normally do. And let us see. Now once you have connected the switch, single switch, if you have if you want if your system requires a few switches, you can engage few pins, but as the if your requirement is of large number of switches say 10 or 20 switches, it may not be feasible to have spare 20 pins to connect 20 switches.

What do you do in that case well, we use a Switch matrix and in the next session we are going to look at how to connect a matrix of switches to a microcontroller so as to minimize the number of pins and yet be able to read every single pin every single switch even in the matrix of switches it will provide you, it will illustrate, it will show bounce and it will have to be debounced in software. So, we will consider these ideas in the next session. Thank you.