

**Transducers For Instrumentation**  
**Prof. Ankur Gupta**  
**Centre for Applied Research in Electronics (CARE)**  
**Indian Institute of Technology, Delhi**  
**Lecture - 34**

**Smart Sensors: Data Acquisition: Analog to Digital Conversion**

Hello, welcome to the course Transducers for Instrumentation. Last lecture we were discussing about the signal conditioning unit, and we discussed that for a smart sensor we have a conventional sensor and in extra electronics which process the signal to bridge the gap between these two we need external hardware or the extra hardware which provide necessary conditions so that this the data generated by the sensor is suitable to apply to these microcontrollers or microprocessors for processing. So in that bridging circuit we have mainly two components, one is the signal conditioning unit we last discussed in the last lecture. The first is the signal conditioning unit where we have bridge completion amplification and filtering unit and the second is data acquisition unit where we saw we have three modules in that sample and hold circuit, analog to digital converter and multiplexer if you needed. So, in last lecture signal conditioning unit we saw that we have bridge completion and then we have amplification. For amplification we use any kind of linear amplifier which boost the signal generated by the transducer or a sensor.

This signal is basically in milli volt range and amplifier amplifies it to a volt range typically up to 3.3 volt or 5 volt depending upon the microcontroller. After this amplification because noise also get amplified, we apply a filter and we saw four types of filter low pass filter, high pass filter, band pass filter and band reject filter or band stop filter. So, depending upon our requirements we can filter out the unnecessary noise from the signal and we need to find out what is our signal frequency in and around of that signal frequency we need a filter which has a pass band located in that range and the rest of the frequencies should be eliminated.

So that was the work of a filter in signal conditioning unit. So, after a signal conditioning unit we get a signal which is now amplified in a volt range in typically 1 to 2 volt of range, and which is free of noise now. But the signal achieved at this point after signal conditioning, this is still an analog signal. Analog means it is continuous in time. So, this signal is difficult to process using our microcontrollers or microprocessors because these hardwares these controllers they can process the data which is in digital form more efficiently.

So our analog signal now it is amplified but still not suitable to apply directly to a microcontroller or a microprocessor. We most of the time convert this analog signal into a digital signal and that part is done using data acquisition unit. In data acquisition we have primarily these three components one is the sample and hold circuit. What this

circuit does it samples the analog signal in time and then hold the data for a limited amount of time so that during this time even if the analog signal is changing the output of a sample and hold is stuck only to that value wherever we have take the sample and now this sample will be converted into a digital form using analog to digital converter. Sample and hold we use because the next stage analog to digital converter it takes amount of time to convert this data into a digital form.

So, during that time this data need to be stable it should not be changing with time. However, the analog signal which is coming after signal conditioning unit that is continuously changing. So, we need a sample and hold circuit which samples the data fast enough and sample this data and hold it for certain time. So this circuit is called sample and hold circuit. Then this is followed by analog to digital converter where a corresponding code will be generated from this hold signal which is depending upon the level of this data or the voltage of this data a particular code will be generated that corresponds to the value of analog signal and this data or this code is in digital form 011101 typically like that.

This data now is ready to be processed using any microcontroller or microprocessor. So this is the complete flow of making a smart sensor from the conventional sensor to the microcontroller or microprocessor. So, we today we are going to discuss the data acquisition part which is this block. So we discuss the data acquisition and in this we discuss basically first two things one is the analog signal. These analog signals that vary continuously throughout a defined range.

So these are the analog signals that vary continuously throughout a defined range. These are representative of noise. Many physical quantities, for example temperature and velocity or pressure. This represents generally a voltage or a current level. However, on the other hand we have digital signals.

These are the signals that take on specific values only. These types of signals are required for digital logic operations. And these are also representative of physical quantities by a series of binary numbers. So this is the data acquisition part. So here we have two types of signals.

One is the analog signal and the other is the digital signal. Analog signals that are continuous throughout the range which assumes all the values in between for example if analog signal is changing from 0 to 5 volt all the intermediate voltage levels are being occupied by this analog signal at some point voltage will be 1.1-volt, 1.2 volt, 1.23 volt all the values are allowed for an analog signal.

So, this is continuous in time. This represents many physical quantities, for example we have a temperature sensor. So based on the temperature it generates a output voltage or a current it can be any of them. So this voltage it produce let us say 1 milli volt. This

corresponds to the physical quantity which is the temperature here and when this value changes it means the physical quantity is changing the temperature is changing because of that this signal generated is changing.

So, this is an analog type of signal which is produced mostly by sensors. On the other hand, we have a digital signal that signals take on only specific values. For example, a digital signal can have only 0 volt or it can have a 5 volt if we are talking about 5 volt compatibility. So 0 volt is called 0 most of the time and 5 volt is called 1 the binary logic level 1. So this is not like the voltage is 1 volt.

Voltage can be anything, but the high voltage is represented by a binary 1. So, if we say binary value 0101 it means the signal is changing from 0 to 5 volt then 0 to 5 volts like that. So, these voltage levels are defined by the microcontroller we use. If let us say we are using a Arduino based microcontroller which is 5 volt input output so this high level is 5 volt. If we are using a raspberry pi which is 3.3-volt microprocessor based hardware so then our high level should be 3.3 volt should not be 5 volt. So this high voltage should be compatible to our next electronics which is going to follow this stage of data acquisition. Now these 0 and 1 which is now binary numbers they also represent the physical quantities. For example a temperature is there which is detected by our temperature sensor.

Interestingly it generates 1 milli volt of signal which is analog in nature but if we convert this signal into a corresponding binary code let us say that comes out to be 001. If the temperature increases so that the output voltage also increases this sensor which this sensor produces. This voltage increase will give the corresponding increase in the binary number so this 001 becomes 100 or so. So this binary number is also a representative of physical quantities however the form of representation is different. Earlier it was a analog signal now we have a digital signal means the code the binary code where all the low and high values are only specified values 0 or 5 volt and the code which is 101011 this code represents the physical quantity not the actual value which is 5 volt.

So 5 volt does not represent the physical quantity but the code actually represents the physical quantity this is the difference between the analog signal processing and our digital signal processing. Now we have certain advantages of digital representation. So, the first one is the values are limited to specific discrete segments. So, the second one is the values are limited to specific discrete segments. Next, it is not subject to the distortions just like analog signals.

These are easy to reproduce. And these are easy to copy or stored. So, these are some of the advantages of digital representation. The very first one is the values are limited to specific discrete segments only. So, we need not to deal with all the continuous values of

analog signal we are only handling the discrete values of voltages and based on that we are processing our signal.

So, we are losing some accuracy in some sense but this is now very easy to implement certain algorithms or do this digital processing. So, these values are very limited so we can handle them very easily. The second the advantage of this binary representation or the digital signal is these are not subject to distortion or just like in analog signal. For example, we have analog signal which is continuous in range if there is a certain distortion or some noise comes in that will distort the signal which is continuous now and we do not have any mean to detect whether this is a spike in the signal or this spike is caused by some external noise. So, we cannot detect later on whether signal is has certain noise or the signal is like this itself.

So, this is the problem with analog signals however in digital signal we know that the voltage levels are only 0 or 5 volt anything in between is we can kind of discard that voltage. So, signal can assume only 0 volt or 5 volt no voltage is allowed in between. So, when we receive the signal and we find out that voltage is not actually 5 volt it is 4.5 volt slightly lesser than 5 volt. So, we can safely say that this has happened because of some noise and we need to restore it back to 5 volt.

So, at the later stage we can easily restore this error what is caused by some external noise or some other factors. So, these digital signals these are not subject to distortion or error just like in analog signals. So, these are easy to reproduce. So, when we transmit these data this digital signals because of some loss or because of some noise if the signal magnitude goes down instead of 5 volt we receive the data as 4.5 volt then we can put a inverter or some external circuitry which can restore this data back to 5 volt.

So, this 4.5 volt becomes again 5 volt without losing any information. So, this is easy to reproduce. The next is easy to copy or stored. So, we have multiple medias to store digital data for example flash drive or hard disk drive. So, these all the devices work on storing on the digital data which is in the form of 0 or 1 or we can say the 0 volt or the high voltage.

So, all these medias are compatible with the digital data. So, these are some advantages of digital representation. Now let us discuss about how we convert this analog signal into a digital signal. So, the very first block is sample and hold circuit.

So, we discuss sample and hold. So, a sample is an instantaneous measurement of an analog voltage. So, sample and the sampling frequency is the number of samples taken per unit time. So, for example we have a data which is analog data now coming in x axis is time and y axis we have voltage input and we have a data which is something like this. So, this is an analog signal. So, what we do we have a kind of sampler which take the sample of this data at certain instances only.

So, this is time and this is the sampler voltage and now we have a sampling frequency applied to this like this and the actual hardware for this is very simple. So, we have a switch connected here and then a capacitor. This is the output  $V_{out}$  and this is the voltage  $V_i$  which is generated by some sensor or some external circuitry. This is  $V_i$  and this point and  $V_s$  actually controls this switch. So, we see  $V_s$  is nothing but a train of pulses which has a certain frequency it is called sampling frequency and when this frequency is applied on this switch this is the switch when there is a high voltage applied this switch will turn close this whole thing will be conducting and when there is a 0 volt applied on  $V_s$  this switch will be open.

So, when this switch is closed whatever is the voltage at  $V_i$  that will pass through this switch and reach  $V_0$ ,  $V_0$  becomes  $V_i$ . So, if we plot the  $V_{out}$  graph as well in the same graph. So, these are the instances when the switch is going to be closed and the path will be conducted and the switch will remain on till we apply this pulse and as soon as the pulse goes down the switch turns off. So, this green dotted line shows where the switch is going to be turned on and  $V$  output becomes  $V_i$ . So, if we plot  $V_i$  in this graph when the switch turns on here  $V$  output becomes  $V_i$  and the  $V$  output will follow  $V_i$  if the  $V_i$  is changing.

So, we can plot like this the  $V_i$   $V$  output is  $V_i$  is changing very similar to  $V_i$  here  $V_i$  is going up so  $V$  output also going up. This is  $V$  output this is the time axis. Now when the switch this sampling frequency or  $V_s$  goes low at that point whatever is the voltage remained at  $V_0$  that will be kept  $V_0$  because the capacitor is there which is going to hold this voltage across this capacitor and  $V_{out}$  is going to hold the voltage whatever is applied by  $V_i$  and the duration for which this output will be hold that is the time till the next pulse arrive which is the sampling frequency. So, if we see this  $V$  output graph this is going to hold this value till the next pulse comes in which is the next green line and when this comes again  $V$  output is going to follow  $V$  input which is something like this and when that pulse goes back again this again goes down it remains same and the pulse goes down. So, this voltage will be retained at constant value.

So, this is called sampling by using this pulse train and when the pulse is not there the voltage is actually hold that is why the hold circuit is called sample and hold circuit. So, if we plot the  $V$  output for the whole pulse of train it will look something like this. So, this is a typical signal which is going to be produced by this sample and hold and we can see this does not represent exactly the analog signal but it is very similar to analog signal. So, if we see the analog signal first goes up and then goes down and again goes up we can see this whole sample and hold circuit after this we also have this orange color output which is also again increasing first then decreasing and then increasing it again though now the values are not kind of like continuous in analog signal we take the sample of analog signal for a certain amount of time and then hold this measured value for certain

amount of time which is called hold time. So, we have now two things one is sample time and one is the hold time and the sampling frequency which is defined by our requirement.

So, if we keep sampling frequency high enough then we can reproduce this analog signal using this sample and hold circuit if we have very high sampling frequency our output signal will be very close to the analog signal if we have a low sampling frequency then we lose out much of a information in sample and hold. So, this is how a sample and holds work and this is a very simple circuit where we have a capacitor which is used to store this data or used to hold the value of voltage and this switch it can be a mosfet switch where we connect source and drain to one VI and V output and the gate is controlled by VS signal which turns this mosfet on and off and this is a very simple circuit of sample and hold. Now the accuracy of this digital representation. It depends on sampling frequency and quantization. Where this quantization is? The number of bits used to represent an analog voltage as a digital number.

And the resolution is the analog step size. For example, we have certain key parameters such as full scale voltage range which is  $R$  number of bits which is  $B$  number of quantization level which is  $2$  to the power  $B$  and we have quantization width which is  $Q$  equal to  $R$  upon  $2$  to the power  $B$ . So, in last slide we discussed sample and hold circuit which produce the discrete samples of a analog signal. So, these samples are now discrete in time and discrete in the voltage levels because we take a sample and we hold the sample for a certain amount of time so that this value is not changed now. Now this sample and hold circuit produce this output which now need to be converted into a digital signal. This is done by analog to digital converters or A to B we call it in short.

And this conversion now it actually depends on the sampling frequency and quantization. So, sampling frequency we saw in last slide when we take the sample that depends on how many samples we collect in unit time that is the sampling frequency and if we collect more samples it means we are very much close to the original data. If we collect less samples it means we are quite away from the actual data and we introduce certain amount of errors in our conversion. So, our digital representation the accuracy of that depends on the sampling frequency how fast we sample and what are the levels we consider for quantization.

So, for example we have a full range of 0 to 5 volt. So, the signal can assume any voltage between 0 to 5 volt if it is an analog signal. But when we convert it into digital we define certain levels the signal can assume only those levels and if the signal falls anywhere in between those levels this will be truncated to the higher value or the lower value depending upon us. So, any signal voltage for example, let us say we have a 5 volt range 0 to 5 volt and we define only 5 levels 1 volt, 2 volt, 3 volt, 5 volt, 4 volt and 5 volt. Any signal cannot assume any voltage between 2 and 3 volt.

So, if my sampled signal comes out 2.7 volt then we truncate it to 3 volt. So, that this signal does not fall in between this voltage. So, once I fix my quantization levels my signal can only assume only those quantization levels it cannot be in between them. So, this quantization levels depends on how many bits I use for representation. So, for example I have a range of 5 volt I want to put levels of like 8 levels in this full 5 volt range then I need 3 bits to represent all these levels because 2 to the power 3 is equal to 8.

If I want to split this whole range into 8 discrete levels I need 3 bits of 3 bits of binary data to represent all these levels. So, 1 sample will correspond to 3 bits of binary data. Now if I have a 5 volt range I just split into 8 levels then my signal cannot assume anything in between it will be truncated to the higher or lower value then the error is more. If I increase more quantization levels I if I want to have let us say 16 levels then my reproduced signal is very close to my analog signal. So, my accuracy is better, but to represent these 16 levels I need to have now 4 bits.

So, for each sample the data which will be converted to that will have 4 bits of binary data. So, my size of data actually increases if I go for more quantization, but my accuracy improves. So, there are certain key parameters to convert this data now to a digital signal. So, we have this full scale range.

So, let us see a graph where it can be represented better. This is time axis and on the y axis we have these levels which are in voltage. This complete voltage range is called R from 0 to maximum voltage whatever is allowed. Now I have multiple levels let us say this is 1 level, this is 2 level, this is 3 level, this is 4 level, this is 5, this is 6, this is 7. So, we have total now 8 levels. The first one is this which I can say 0 0 0 then I can write 0 0 1, 0 1 0 and 0 1 1.

This is 1 0 0, 1 0 1, 1 1 0 and 1 1 1. So, these are all 8 levels corresponding to 3 bits. So, here we have B equal to 3 bits and the number of levels equal to 2 to power B which is equal to 2 to the power 3 or 8. So, we have first level, second level, third, fourth, five, six, seven and eight. So, these are the 8 levels. Now if a signal comes in for example, sample and hold circuit generates a signal which comes out like this, any arbitrary signal.

Now if you want to convert it into a binary signal it can be easily coded. So at this point the value of the signal is close to this 0 signal. So here the code is 0 0 0 then here 0 0 1 then here it is 0 1 1 then it is 1 0 0 then it is 0 1 1 here it is 1 1 0 then this is 0 1 1 here it is 0 1 0 and 0 0 1 and back to 0 0 0. So if we want to represent this signal in terms of binary data we can easily say the signal can be represented as 0 0 0 then 0 0 1, 0 1 1, 1 0 0, 0 1 1, 1 1 0, 0 1 1, 0 1 0, 0 0 1 and 0 0 0. So this signal which is generated by the sample and hold circuit can now represented by this stream of binary data which is in the form of 0 and 1 and this 0 and 1 is logical 0 and 1 it can be in the voltage range it can be 0 or 5 volt where this 0 represent a 0 volt and 1 represent a 5 volt or 3.3 volt. This voltage

level depends on the microcontroller or microprocessor we are using for Arduino or that controller this 1 volt 1 logic level is 5 volt and for raspberry pi this 1 logic level is 3.3 volt. So this is how we can convert a signal which is generated by a sensor which was of very small value first we do a signal conditioning on that and then by data acquisition or this external hardware which converts first it to sample and hold and then we use ADC to convert it into a digital stream which is shown here. So this signal which is arbitrary signal now is converted into a digital form.

So, this is all for today.

Thank you.