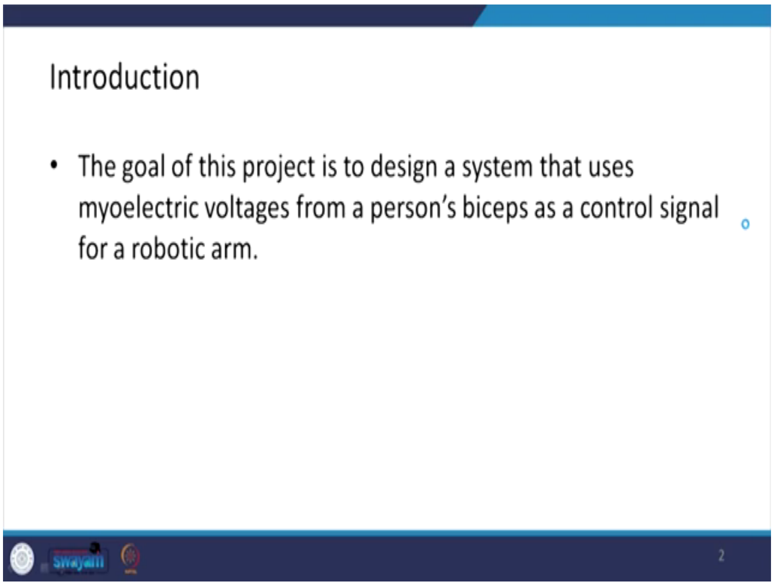
Mechatronics
Prof. Pushparaj Mani Pathak
Department of Mechanical and Industrial Engineering
Indian Institute of Technology, Roorkee

Lecture - 37
Myoelectrically Controlled Robotic Arm

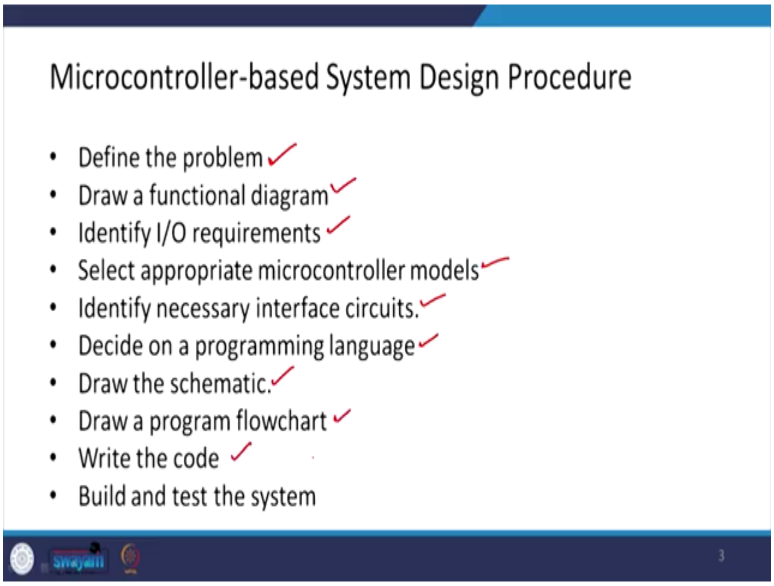
I welcome you all to today's NPTEL online certification course, lecture on Mechatronics. Today we are going to talk about Myoelectrically Controlled Robotic Arm. This example for the design of the mechatronic system, which I have taken, is based on the textbook by Alciatore and Histand, so it is a very nice realistic example. So, those who want to refer in detail can refer to that textbook; I will be giving the reference at the end of this lecture also.

(Refer Slide Time: 01:14)



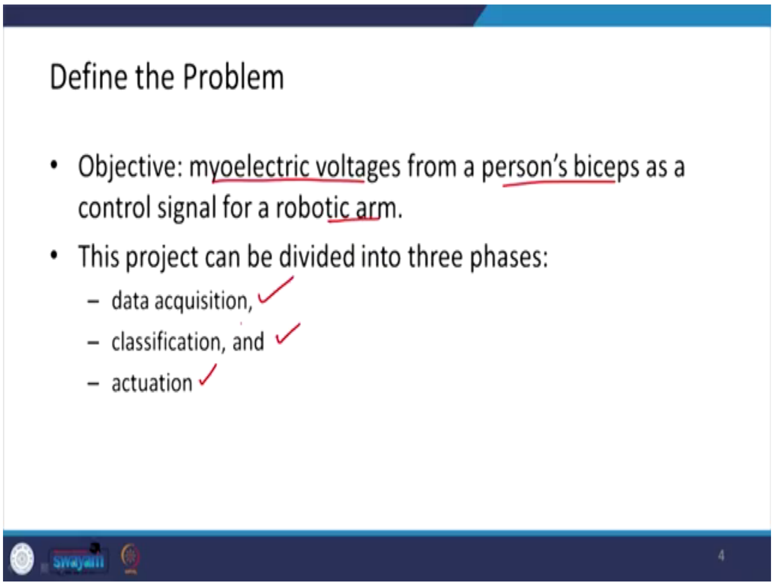
The goal of this project is to design a system that uses myoelectric voltages from a person's bicep as a control signal for a robotic arm. And you see, these myoelectric voltages are very small in amount as well as they have a lot of noise also in these signals. So, we are going to talk about all steps involved in the design of this entire mechatronic system.

(Refer Slide Time: 02:07)



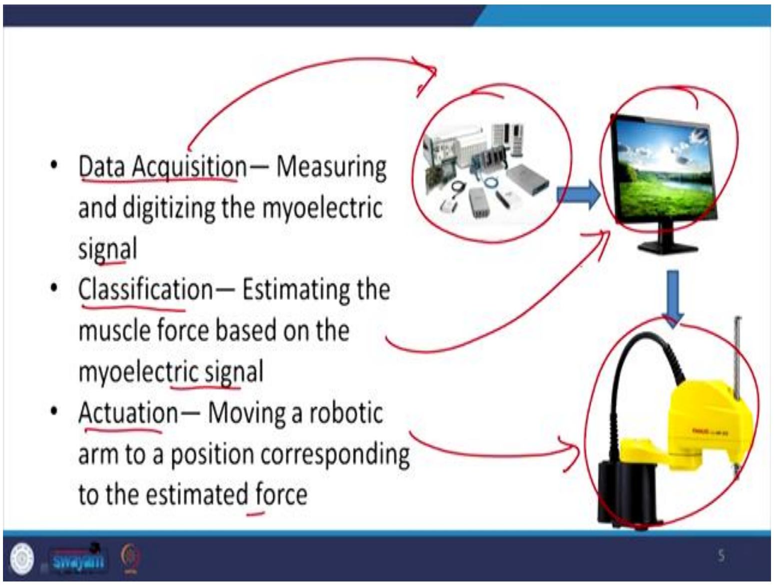
The microcontroller-based system design procedure consists of defining the problem and then drawing a functional diagram, identifying what is our input-output requirement, then selecting the appropriate microcontroller model. The identification of necessary interfacing units circuits and deciding on what programming language we want to program for the system, drawing of the schematic diagram and a program flowchart, and finally, writing a code and building a building and testing the system. So, these are the steps that are involved in the microcontroller-based system design procedure. We are going to discuss each and every step.

(Refer Slide Time: 03:11)



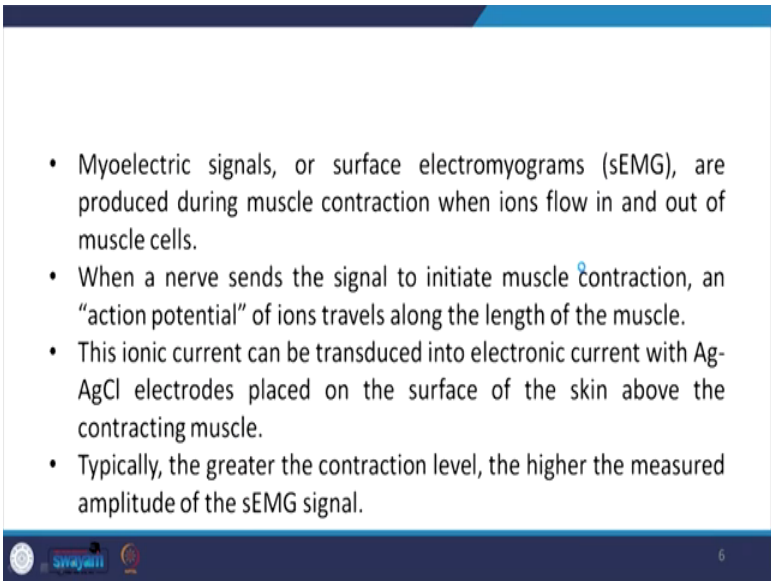
The definition of the problem is to get myoelectric voltages from a person’s bicep as a control signal for a robotic arm. And based on that signal, we want to control a robotic arm. This project can be divided into three phases; one is the acquisition of data from the person’s bicep. And then the classification of that data and then ultimately the actuation of the robotic arm. So, in these three steps, we can divide this project into these three phases.

(Refer Slide Time: 03:55)



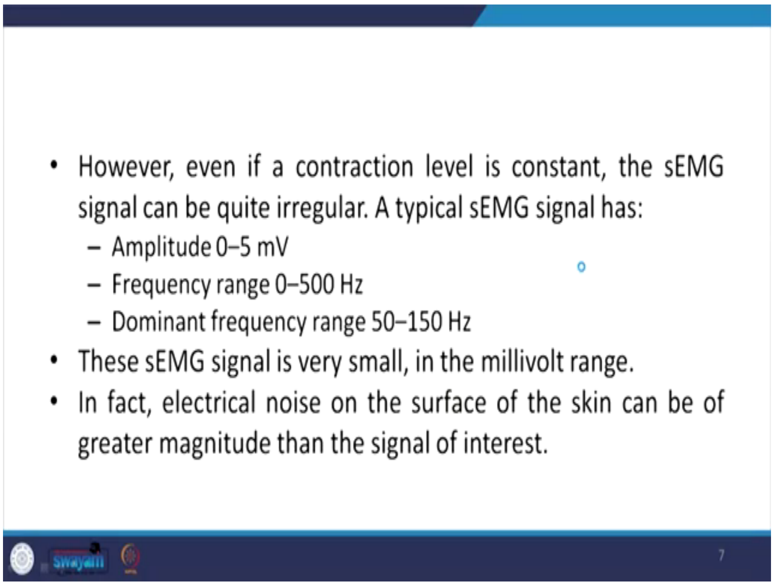
What do these phases involve? The data acquisition involves measuring and digitizing the myoelectric signal. So, first, you need to measure and then digitize that myoelectric signal. And then, classification involves estimating the muscle forces based on the myoelectric signal. And based on the myoelectric signal, we can estimate the muscle force corresponding to that signal and then ultimately the actuation that is moving a robotic arm to a position corresponding to estimated forces. So, here as you can see here, we have the data acquisition over here, and then this is your classification. And then here is the this is you’re the actuation. So, this is the symbolic representation of these steps.

(Refer Slide Time: 05:07)



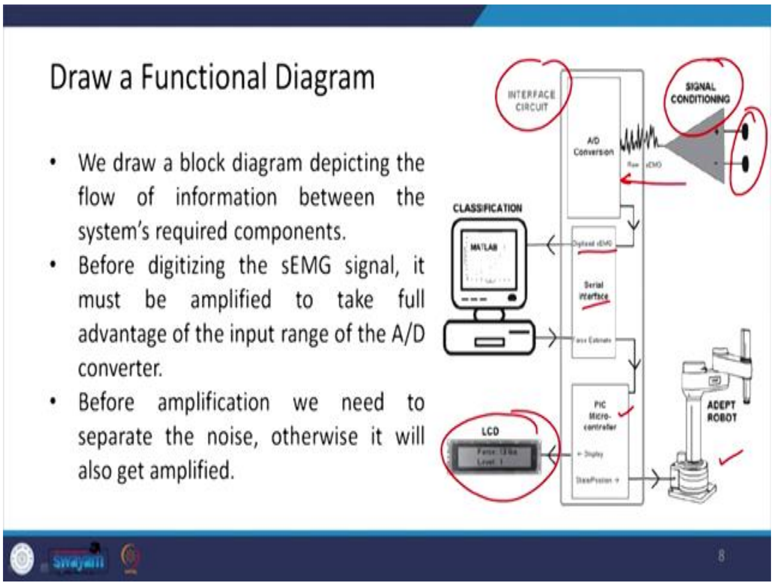
Then you see these myoelectric signals, or surface electromyograms sEMG as we call it, in short, is produced during muscle contraction when ion flow in and out of the muscle cell. And when a nerve sends the signal to initiate muscle contraction, and action potential of ions travels along the length of the muscle. And this ionic current can be transduced into an electric current with the help of Ag-AgCl electrodes placed on the surface of the skin above the contracting muscle. Typically, the greater the contraction level, the higher is the measured amplitude of the sEMG signal.

(Refer Slide Time: 06:12)



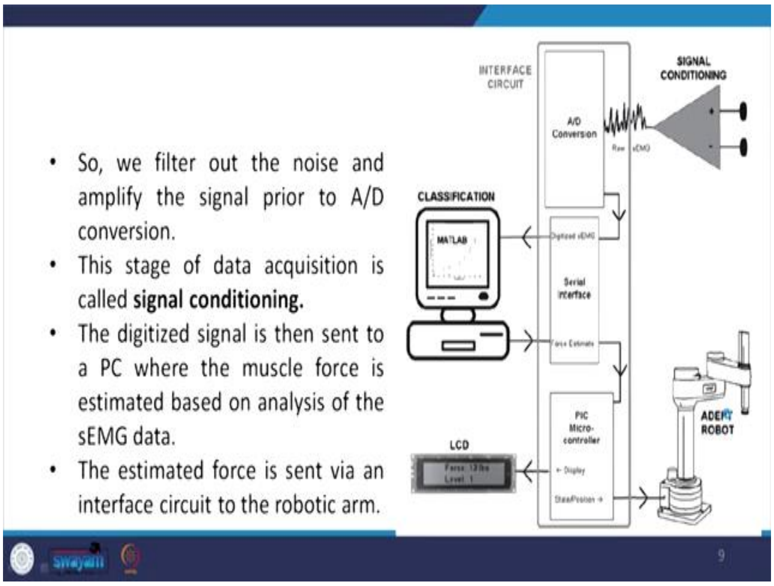
However, even if a contraction level is constant, the sEMG signal can be quite irregular, and the typical characteristic of these signals are amplitude is of the range of 0 to 5 milli Volt, you can see a very small amplitude; frequency range is 0 to 500 Hertz, and the dominant frequency range is around 50 to 150 Hertz. And as I said, these signals are very small in the millivolt range of 5 milli Volt. And in fact, electrical noise on the surface of the skin can be of greater magnitude than the signal of interest itself. Next, we look at the functional diagram.

(Refer Slide Time: 07:16)



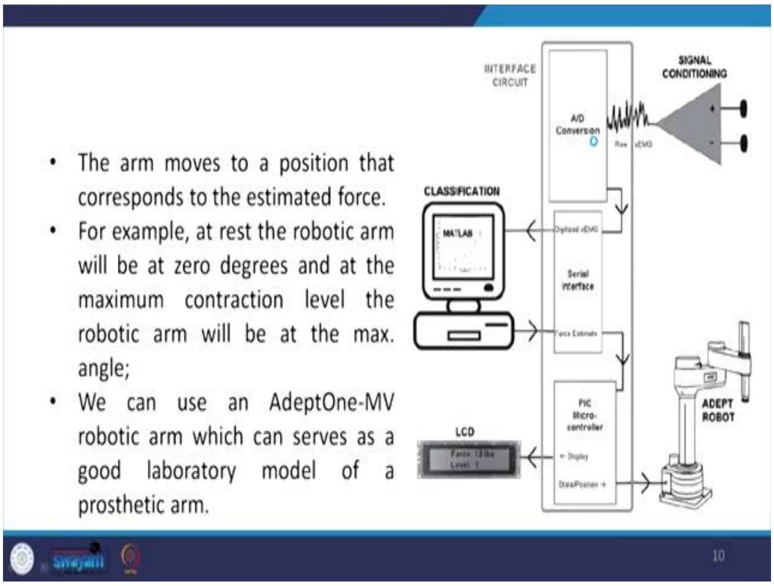
In the functional diagram, we draw a block diagram depicting the flow of information between the system's required components. And before digitizing the sEMG signal, it must be amplified to take full advantage of the input range of the analog to digital converter. And before amplification, we need to separate the noise. Otherwise, it will also be get amplified. So, here you can see that we get the signal over here; then the signal conditioning is carried out, and then we have the interfacing circuit, which consists of your A to D conversion, serial interfacing, and a microcontroller. So, the flow of signal is something like this, your signal conditioning sends the signal to A to D converter, analog to digital converter, and these digital sEMG signals are then sent to your PC for a classification purpose. And using a certain program, you can classify that and calculate the force, and this force estimation estimated is again passed through EMG signals to the PIC microcontroller. And from this PIC microcontroller, it can be sent to an Adept robot. We can also have a display over here just to display of giving a visual display of the force which is being sent to be to the robot.

(Refer Slide Time: 09:05)



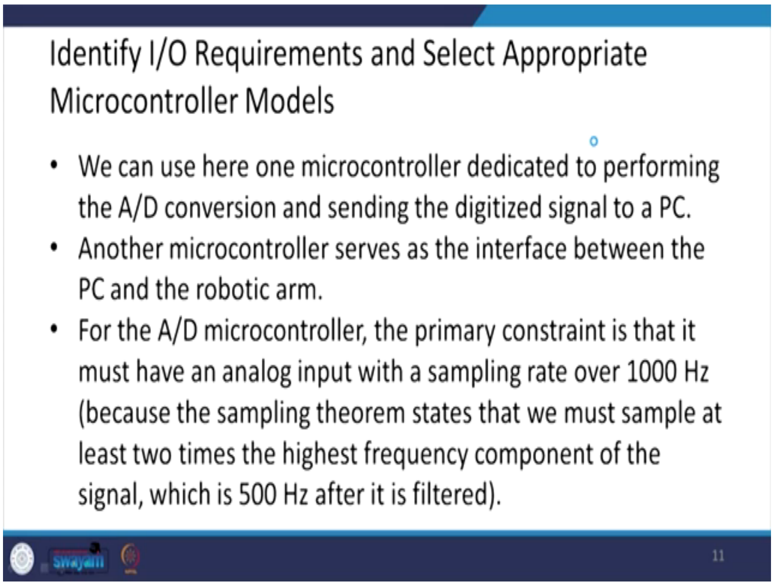
We filter out the noise and amplify the signal prior to A to D conversion over here. And if you are not doing a prefiltering, then we may have a certain issue. And this stage of data acquisition is called signal conditioning. And this digitized signal can be sent to a PC, where muscle force is estimated based on the analysis of sEMG data. This estimated force is sent via an interfacing circuit to the robotic arm. Now, the arm moves to a position that corresponds to the estimated force.

(Refer Slide Time: 10:06)



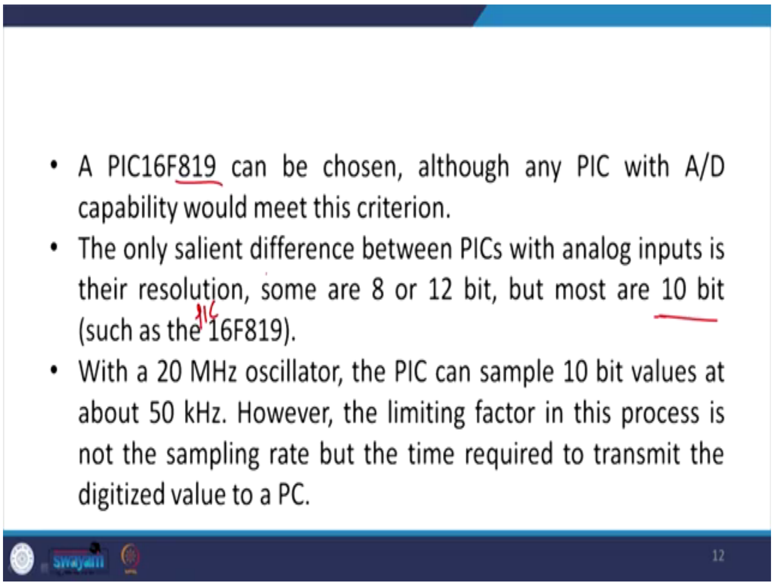
For example, at rest, the robot arm will be at zero degrees, and at the maximum contraction level, the robotic arm will be kept at the maximum angle. We can use an AdeptOne-MV robotic arm which can serve as a good laboratory model for the prosthetic arm, and its configuration is very similar to that of a prosthetic arm.

(Refer Slide Time: 10:29)



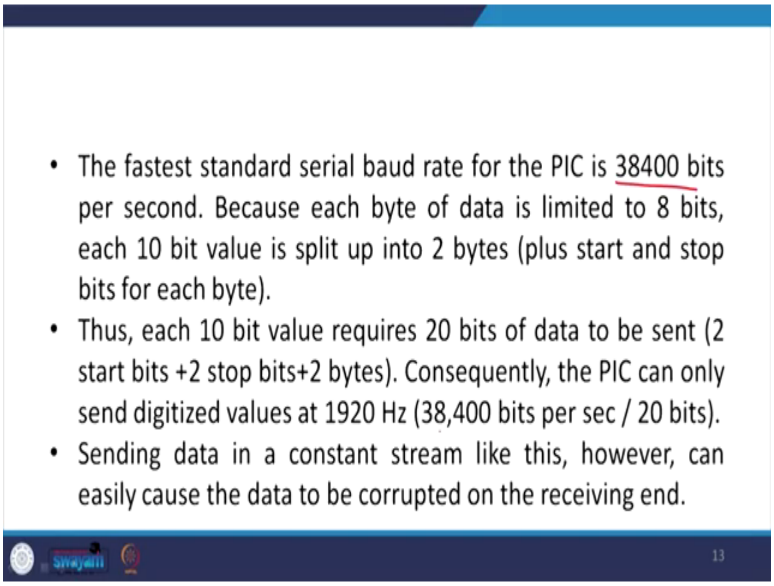
Then next is the identification of input-output requirements and selection of the appropriate microcontroller model. Here one can use a microcontroller dedicated to performing the analog to digital conversion and sending the digitized signal to a PC. Why do we require analog to digital conversion? Because the myoelectric signal is the analog signal, whereas your system will be requiring the digital signal. So, we need to have the analog to digital conversion before it is being processed. Ideally, we can use the two microcontrollers, one which can be used for analog to digital conversion, and another can serve as the interface between the PC and the robotic arm. Here we can have the two microcontrollers here; one for analog to digital conversion and another which interfaces between your system and the robotic arm. For the analog to the digital microcontroller, the primary constraint is, of course, that it must have an analog input with a sampling rate over 1000 Hertz, because the sampling theorem, as we have seen states that, we must sample at least twice of the highest frequency component of the signal, which is 500 Hertz as we have seen in the previous slides after it is a filter. If you want to sample the signal properly, the sampling rate frequency has to be 1000 Hertz.

(Refer Slide Time: 12:54)



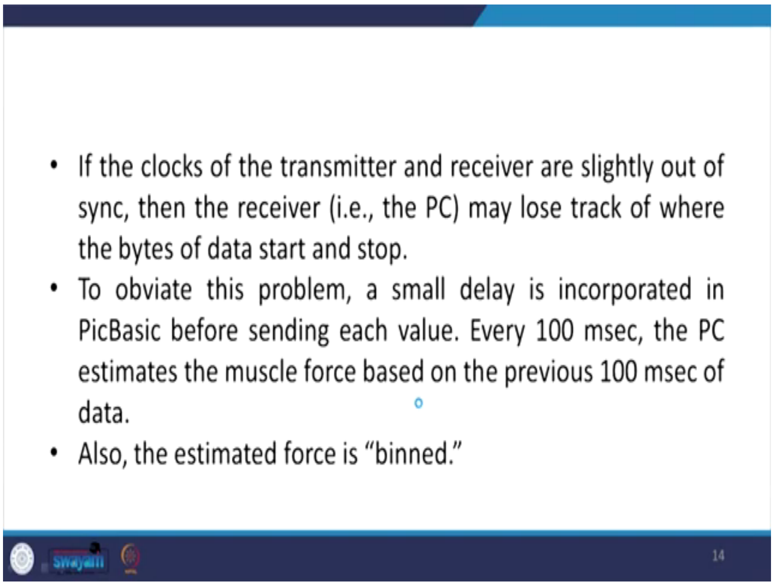
A PIC16F819 can be chosen, although any PIC with A to D capability would meet these criteria. The only salient difference between PIC with analog input is their resolution, some are 8 or 12 bit, but most are 10 bit, such as 16F819, which is PIC 16F819. Now, with a 20 MegaHertz oscillator, the PIC can sample 10 bits value at about 50 kiloHertz. However, the limiting factor in the process is not the sampling rate but the time required to transmit the digitized value to a PIC to a PC.

(Refer Slide Time: 14:12)



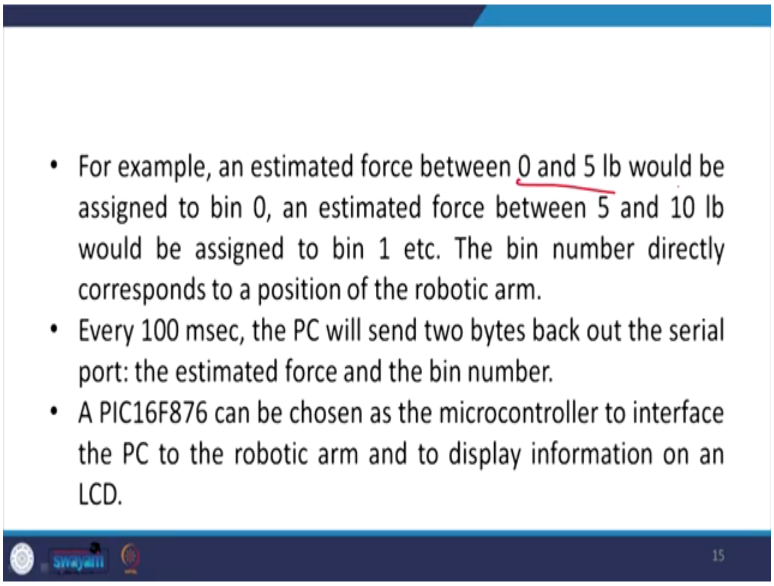
Now, the fastest standard serial baud rate for the PIC is 38400 bits per second. Because each byte of data is limited to 8 bits, each 10-bit value is split up into 2 bytes, plus the start and stop bit for each byte that has to be sent. So, for each 10-bit value requires 12 bits of data to be sent 2 for the start bit for the 2 bytes, 2 for the stop, for the 2 bytes. Consequently, the PIC can only send a digitized value at 38400 divided by 20. So, at around 1920 Hertz Now, sending data in a constant stream like this, however, can easily cause the data to be corrupted on the receiving side.

(Refer Slide Time: 14:34)



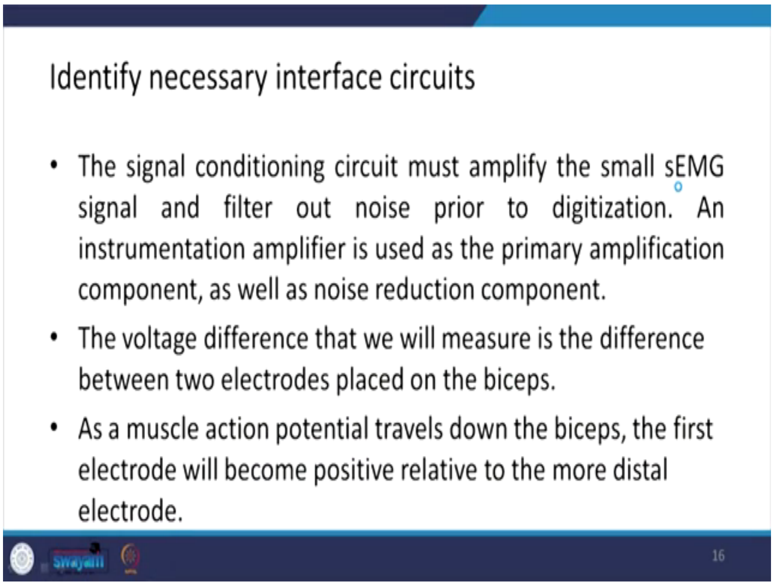
Now, if the clock of the transmitter and receiver are slightly out of synchronization, then the receiver, that is, the PC, may lose track of where the bytes of data starts and where it is stopped. So, to obviate this problem, a small delay can be introduced in the PicBasic before sending each value. For every 100 milliseconds, the PIC estimates the muscle forces based on the previous 100 milliseconds of data. So, this must delay we can introduce on that. And also, the estimated force is binned; binned means that we give the designation, that is, for an estimated force between 0 and 5 pounds would be assigned to a bin 0.

(Refer Slide Time: 15:58)



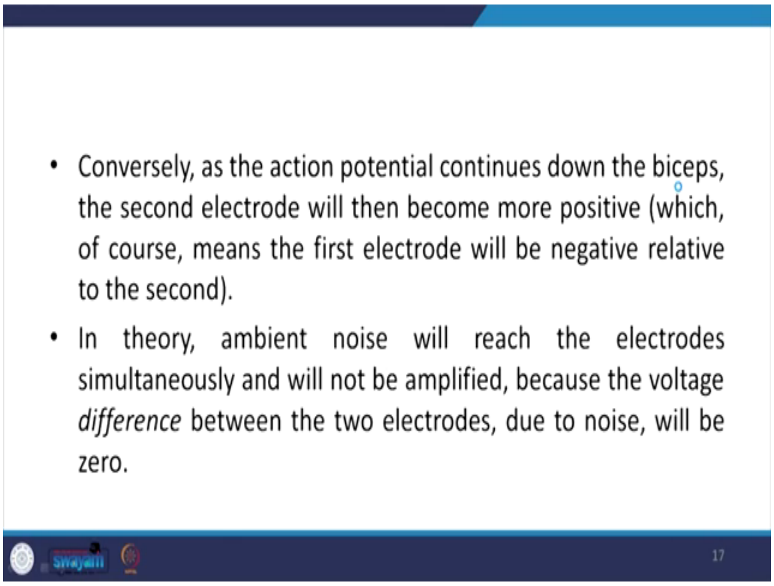
An estimated force between 5 and 10 pounds could be assigned bin one and so on. So, the bin number directly corresponds to the position of the robotic arm. Corresponding to each bin, there is going to be the position of the robotic arm. And this bin numbering is based on what amount of force is being extracted from your sEMG signal. Every 100 milliseconds, the PC will send two bytes back out the serial port, the estimated force, and the bin number. Now, the PIC16F876 can be chosen as the microcontroller to interface the PC and the robotic arm, as I told you before, and display the information on an LCD. Then next, let us look at what are the necessary interfacing circuit. The interfacing circuit unit will comprise the signal conditioning unit, and this signal conditioning circuit must amplify the signal and filter noise prior to digitization. And an instrumentation amplifier can be used for this purpose, for the primary amplification component as well as for the noise reduction component.

(Refer Slide Time: 17:20)



Now, the voltage difference that will measure is the difference between the two electrodes placed on the biceps. And as the muscle action potential travels down the bicep, the first electrode will become positive relative to the more distal electrode.

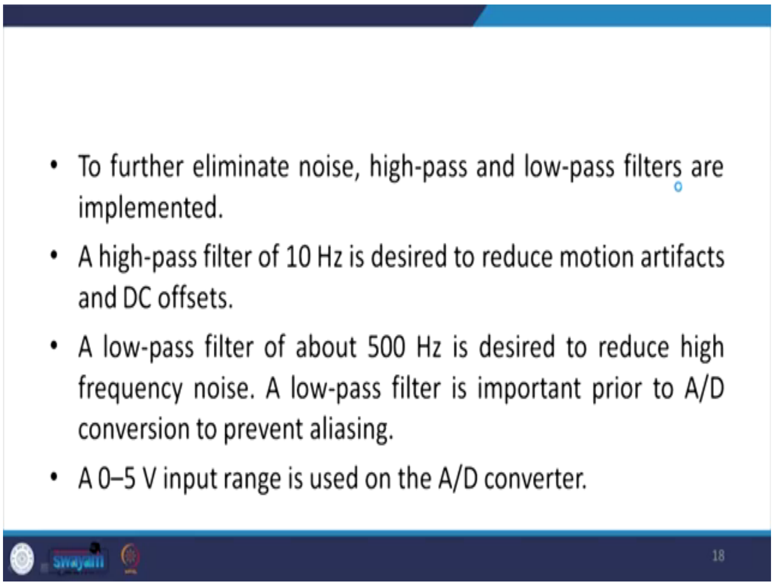
(Refer Slide Time: 17:31)



And conversely, as the action potential continues down the bicep, the second electrode will then become more positive, which of course, means that the first will be becoming negative. So, that is how it is going to be. Now, in theory, ambient noise will reach the electrode simultaneously and will not be amplified; because the voltage difference between the two electrodes due to noise will be zero.

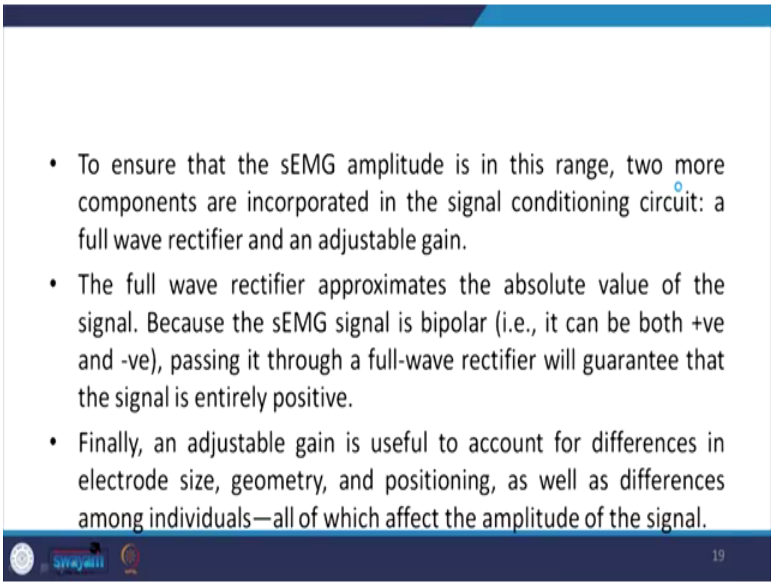
To further eliminate the noise, high pass and low pass filters are implemented. So, a high pass filter of 10 Hertz is desired to reduce motion artifacts and the DC offset voltage

(Refer Slide Time: 18:31)



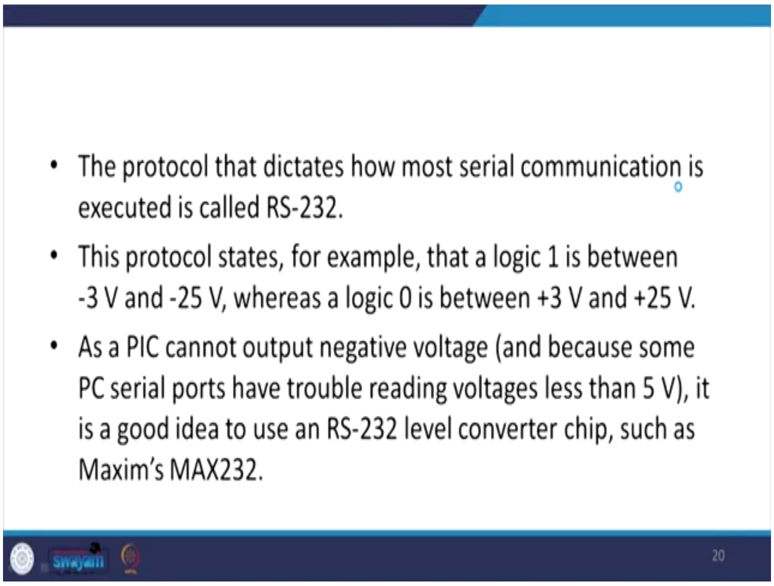
And a low pass filter of about 500 Hertz is desired to reduce high-frequency noise. So, a low pass filter is important prior to analog to digital conversion to prevent aliasing that is there. And A 0 to 5 Volt input range is used on the analog to digital converter.

(Refer Slide Time: 19:04)



Now, to ensure that sEMG amplitude is in the range, two more components are incorporated in the signal conditioning circuit these are the full-wave rectifier and an adjustable gain. Now, what does this full-wave rectifier does? A full-wave rectifier approximates the absolute value of the signal; because the sEMG signal is bipolar, it can be both positive and negative. Passing it through a full-wave rectifier will guarantee that the signal is entirely positive. And finally, an adjustable gain is useful to account for the difference in electrode size, geometry, and positioning as well as differences among individuals, all of which affect the amplitude of the signal. Now, the protocol that dictates how most serial communication is executed is RS 232. The protocol states, for example, that logic 1 is between -3 Volt to -25 Volt, whereas logic 0 is between +3 Volt and + 25 Volt.

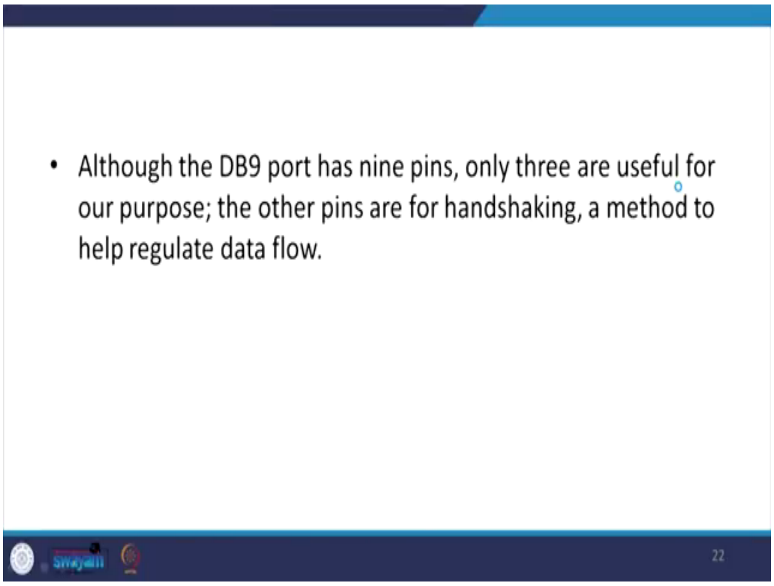
(Refer Slide Time: 20:17)



As a PIC cannot output negative voltage and because some PC serial ports have trouble reading voltages less than 5 Volt, it is a good idea to use is RS 232 level converter chip, such as a MAX 232. I am not discussing some of the details over here because of the time constant. And as I told you in the beginning, you can refer to professor Histand and Alciatore's book for better clarification. These chips convert the TTL, CMOS level signal to RS 232 level signal and vice versa, and it is important to note that they also involve the signal.

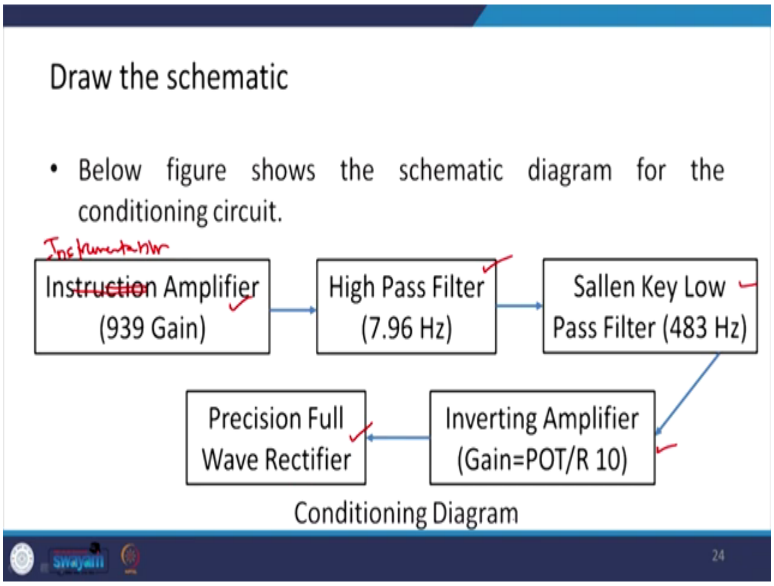
For example, given a plus 5 Volt input, it will output about minus 8 Volt, and when given 0 Volt, it outputs about plus 8 Volt. These chips are able to provide this output using a 5 Volt power supply and ground. This is made possible using a technique called charged pumping, which uses capacitors to store and boost voltage.

(Refer Slide Time: 21:44)



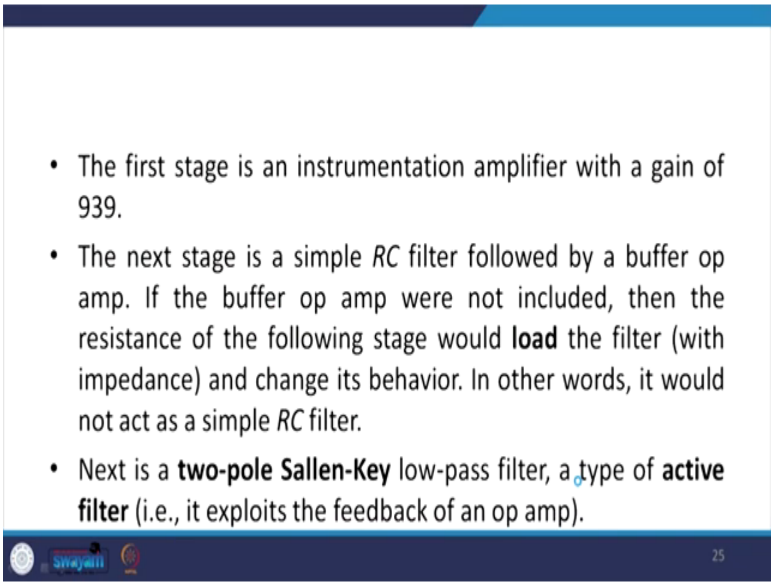
Although the DB 9 port has nine pins here, only three are useful for this purpose; the other pins are for handshaking, a method to help regulate the data flow, which I have discussed in my earlier lectures. Now, next, we focus our attention on deciding on a programming language. Since we are using the PIC microcontroller, PIC basic probe is used as the programming language. The speed and memory constants are not primary concerns here, so assembly language is not required. Also, the serial communication and the LCD interface commands provided by PIC basic pro will be quite useful, as I said in the beginning.

(Refer Slide Time: 22:51)



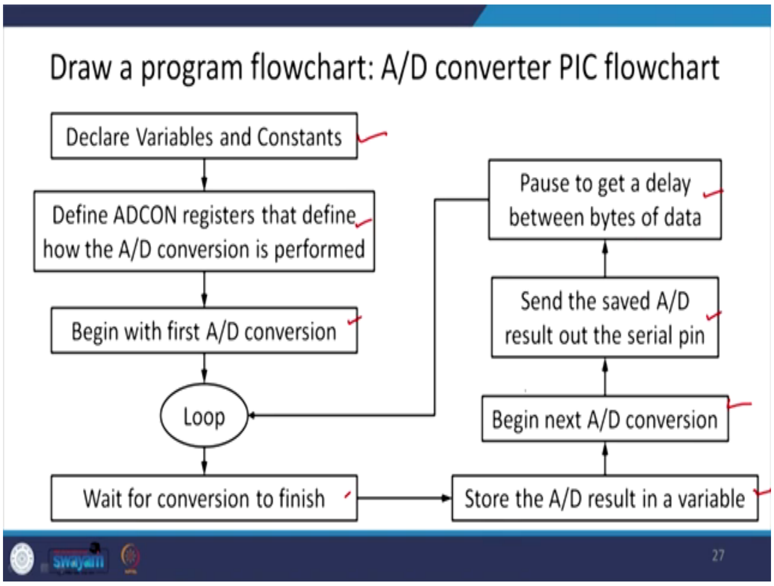
So, next is to draw the schematic. So, we show the schematic figure for the conditioning circuit. So, we have an instruction amplifier, the 939 gain, then you have a high pass filter, and then there is a Sallen key low, and then we have the inverting amplifier and then the precision full-wave rectifier. So, the next step is the draw the schematic. First goes to the instrumentation amplifier, and then it goes through the high pass filter, and then there is a low Sallen key low pass filter, and then it goes to the inverting amplifier and finally, the precision full-wave rectifier. So, this is the conditioning diagram.

(Refer Slide Time: 23:55)



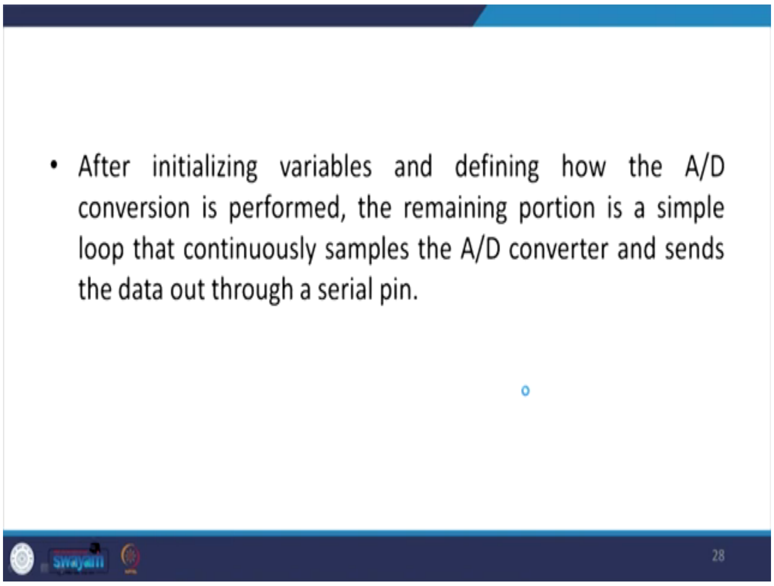
The first stage is an instrumentation amplifier with a gain of 939. And the next stage is a simple RC filter followed by a buffer op-amp. If the buffer op-amp were not included, then the resistance of the following stage will load the filter and change its behavior. So, it would not act as a simple RC filter. So, next is a two-pole Sallen key low pass filter, a type of active filter; it exploits the feedback of an operational amplifier. Now, active filters are more robust than a passive filter, such as an RC filter, and additional poles are better at attenuating unwanted frequencies. The next stage is the inverting op-amp, where the feedback register is a potentiometer used to adjust the overall gain of the system. And finally, in the last stage, there is a precision full-wave rectifier, which requires no forward bias to turn on the diodes.

(Refer Slide Time: 25:21)



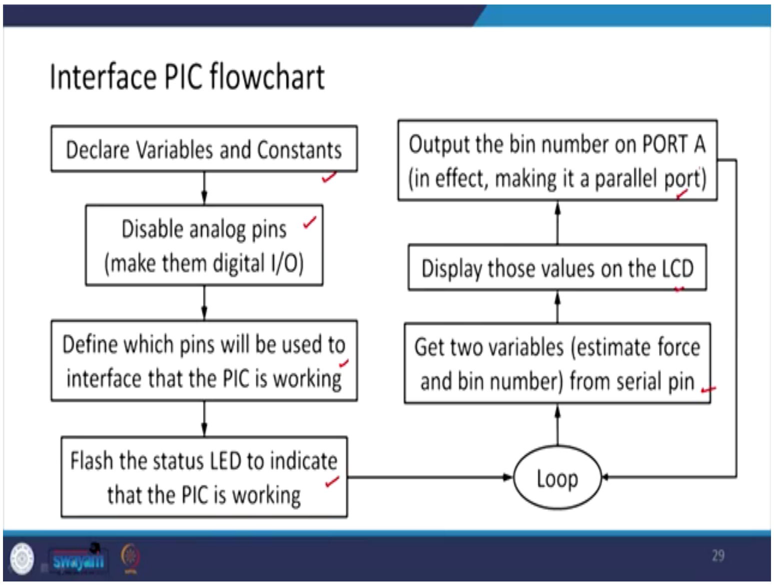
The next step is that draw a program flowchart. So, here we declare variables and constants and then define the ADCON register that defines how analog to digital conversion is performed; then begin with the first analog to digital conversion, then there is a loop, wait for the conversion to finish, store the analog to digital results in a variable, begin next A to D conversion, send the saved A to D result out the serial pin and pause to get a delay between the bytes of data as I discussed in before. And then it goes to the loop.

(Refer Slide Time: 25:58)



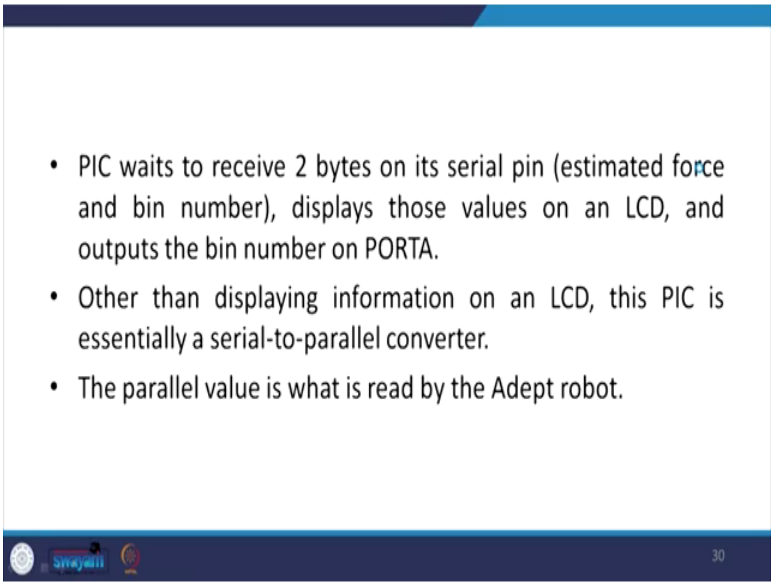
And after initializing variables and defining how the analog to digital conversion is performed, the remaining portion is a simple loop that continues continuously samples the analog to digital converter and sends the data out through the serial pins.

(Refer Slide Time: 26:23)



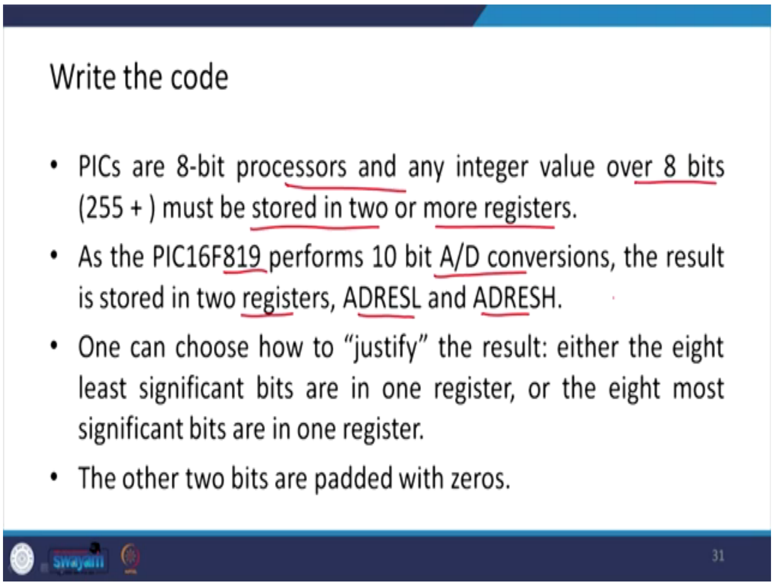
Now, interface PIC flow charts if we look at. So, declare variables and constants as is done usually, disable analog pins make them digital input-output pins, defined which pin will be used to the interface that the PIC is working. Flash the status LED to indicate that the PIC is working, and then we have a loop the get the two variable estimate force and bin number from the serial pin, display those values on the LCD, and output the bin number on PORT A in fact, making it a parallel port and then this can keep on going. So, this is the interface PIC flow chart.

(Refer Slide Time: 27:05).



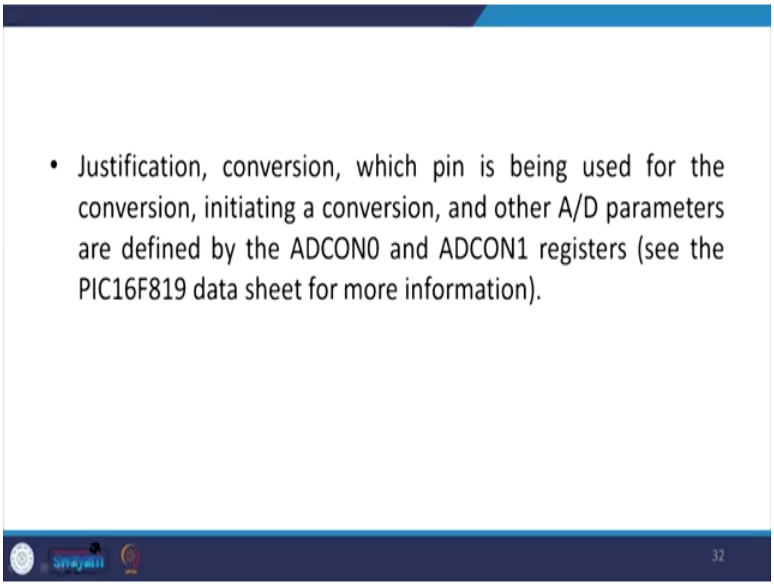
Now, PIC waits to receive 2 bytes on the serial port, estimated force, and bin number; display those values on the LCD and output the bin number on PORTA. Other than displaying information on the LCD, this PIC is essentially a serial to parallel converter. And the parallel value is what is read by the Adept robot.

(Refer Slide Time: 27:45)



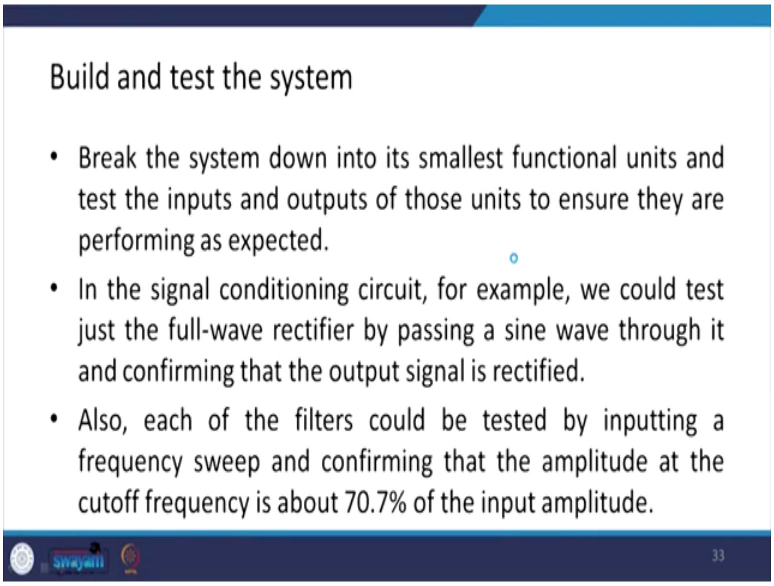
The next step is the writing of the code. So, PIC is an 8-bit processor, and any integer value over 8 bit must be stored in two or more registers. As the PIC 16F819 performs 10-bit analog to digital conversion, the result is stored in the two registers, ADRESL and ADRESHs. One can choose how to justify the results; either the eight least significant bits are in one register, or the eight most significant bits are on one register. The other two bits are padded with zeros.

(Refer Slide Time: 28:38)



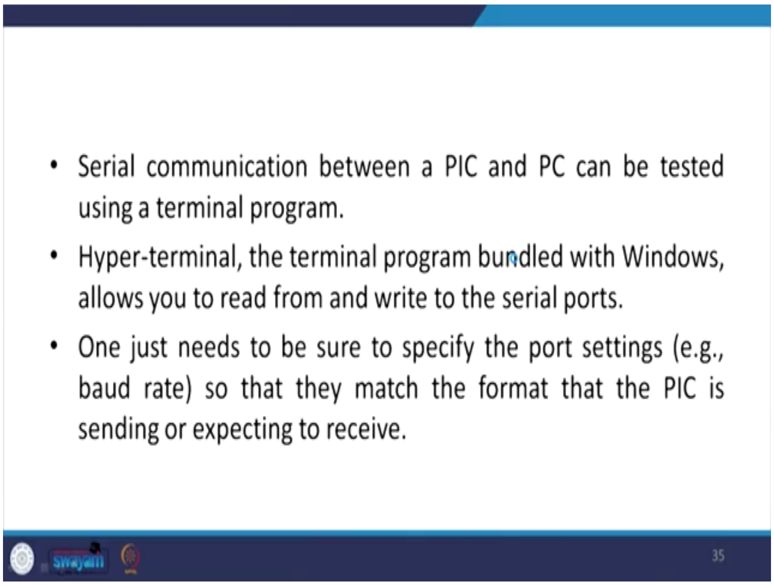
Justification, conversion, which pin is being used for the conversion, initiating a conversion, and another analog to digital parameters are defined by ADCONO and ADCON 1 register. You can see further in the data seat of PIC16F819.

(Refer Slide Time: 29:05)



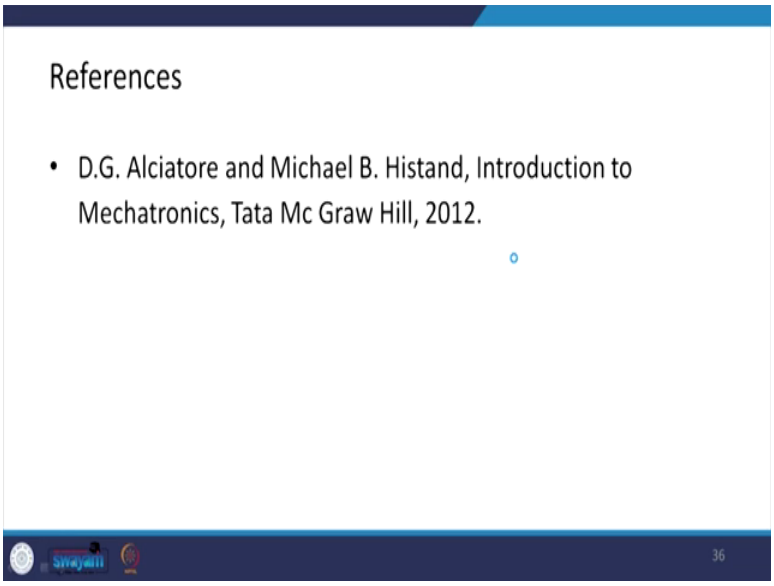
Then the last and final stage of this mechatronics design is the build and testing of the system. So, break the system down into the smallest functional units and test the input and output of those units to ensure they are performing as expected. In the signal conditioning circuit, for example, we could test just like just the full-wave rectifier by passing a sine wave through it and confirming that the output signal is getting rectified. Also, each of the filters could be tested by putting a frequency sweep and confirming that the amplitude at the cutoff frequency is about 70.7% of the input amplitude. And with any PIC, it is a good idea to include a status LED; that indicates that the PIC is on and functioning to some degree. While troubleshooting, it is also useful to flash the status LED at different points in the code to help you track the execution. And LCDs are, as I said before, extremely useful once they are functioning in debugging codes; because they can easily be used to display diagnostic messages and values before being stored in the variables.

(Refer Slide Time: 30:19)



Serial communication between PIC and PC can be tested using the terminal program. Hyper terminal, the terminal program bundled with Windows, allows you to read from and write to the serial ports. One just needs to be sure to specify the port setting, that is the such as the baud rate so that they match the format that the PIC is sending or expecting to receive.

(Refer Slide Time: 30:58)



So, this is the reference which I talk to you, that is Introduction to Mechatronics by Alciatore and Histand by Tata Mc Graw Hill 2012. There are many other editions of it also available. So, you can refer to this textbook for further details.

Thank you.