Transducers For Instrumentation

Prof. Ankur Gupta
Centre for Applied Research in Electronics (CARE)
Indian Institute of Technology, Delhi

Lecture - 31
Smart Sensors: Microcontroller based.

Hello, welcome to the course Transducers for Instrumentation. We are discussing smart sensors and last lecture we discussed that smart sensor is nothing but the conventional sensor which is now interfaced with external electronic circuitry, microprocessor and microcontrollers which actually processing this data generated by conventional sensor and the processing is done at locally itself making it as a sensor node and this data which is already processed is sent to remote location to some other sensor mesh or over the internet. This smart sensor now they have this external circuitry which is based on microcontrollers or microprocessor. There are two types and we discussed one microcontroller which is very much used is Arduino and the microprocessor which widely used is the Raspberry Pi or R Pi we call it. So today we are going to discuss this microcontroller based smart sensors and we use Arduino generally for this application. So Arduino is a open source hardware it means you can modify the hardware itself as per your own requirements and the software which is used to program these Arduinos we have to implement certain algorithm or so that is also a open source you can modify you can change it you can use it any form to suit your applications.

So Arduino is a open source platform most of the time used with for a low cost sensors because the cost of Arduino is very less compared to microcontroller based hardware such as R Pi and there are plenty of interfaces board available for example in the Arduino itself we do not have much of a functionality but external hardware's are available which you can use as per your requirement. For example you need a pressure sensor so these pressure sensors are available as a standalone unit. You can buy it and plug it with the Arduino to measure the pressure. Generally other types of hardware's are available for these Arduino boards. It is an open source and very flexible easy to use hardware and the software programming is also not very complex this can sense the environment by making use of the external sensor circuitry that may be a conventional sensor or may be a ready made hardware available in the market and generally on off light or there are many kind of motor and actuator hardware's you can make using Arduino.

So let us discuss today what we can how we can use a Arduino board to make a smart sensor. So we discussed microcontroller based smart sensor. Arduino has become the standard microcontroller. And researchers to its ease of use, low cost and plenty of interface boards. This is an open-source electronics. Flexible, easy to use hardware and software. This Arduino can sense the environment by receiving input from a variety of sensors. It can affect its surroundings. By controlling light, motor or power. Or other actuators. This microcontroller on the board is programmed using, number one is the Arduino programming language. And based on the wiring. So here we have this microcontroller Arduino. This is an open source electronics which can sense the environment by receiving the inputs from a variety of sensors. You can attach a temperature sensor, pressure sensor or many other kind of sensors which are readily available in market or you can develop these sensors in house and you can hardware's you can make using Arduino. So let us discuss today what we can how we can use a Arduino board to make a smart sensor. So we discuss microcontroller based smart sensor. Arduino has become the standard microcontroller. And researchers to its ease of use, low cost and plenty of interface boards. This is an open source electronics.

Flexible, easy to use hardware and software. This Arduino can sense the environment by receiving input from a variety of sensors. It can affect its surroundings. By controlling light, motor or power. Or other actuators. This microcontroller on the board is programmed using, number one is the Arduino programming language. And based on the wiring. So here we have this microcontroller Arduino. This is an open source electronics which can sense the environment by receiving the inputs from a variety of sensors. You can attach a temperature sensor, pressure sensor or many other kind of sensors which are readily available in market or you can develop these sensors in house and you can connect directly to these microcontroller which is very easy to program and interface with the sensor. It can affects its surrounding by controlling certain input output pins which you can use to actuate the other hardware. And this microcontroller you can program it in two parts. One is the hardware circuitry which pin you connect to which sensor. This is one way of making a complete circuit and then based on this wiring you can program the microcontroller which is sitting in this Arduino microcontroller. There is a microcontroller placed on this board that can be programmed using a programming language which is open source again for this Arduino and the programs generally we write for Arduino they are called sketches. So these are the basic features of this microcontroller. Let's discuss how is the power compatibility of these Arduino boards. So the power pins are as follows. So we have let's say a Arduino board. In this Arduino board generally we have a USB connector.

We have a power connector as well. And then we have multiple pins on the side. And some of these pins are 3.3 volt and some are 5 volt. So some of the pins are 3.3 volt output. Some pins are 5 volt output. And some pins are input signals. So here we have this Arduino board which has a USB connector where you can connect a USB cable coming from your computer. Directly you can connect this Arduino to your computer using this USB cable and the power will be drawn from this USB connector itself which is 5 volt power supply. So the Arduino board runs on 5 volt power supply and all the other pins are 5 volt logic level. Some pins are 3.3 volt output. This 3.3 volt is generated on board in Arduino using external power circuitry. This 3.3 volt output you can use as a power supply to a very small sensors if the need arises otherwise all the pins and the logic levels in Arduino are 5 volt logic level. So the 3.3 volt signal which is coming in this need to be step up to 5 volt so that it is compatible with Arduino board. So all the hardware circuitry which we need to use for Arduino should be 5 volt range. So the V in supply, the input voltage to the Arduino. Vold when it is using an external power source.

We can supply voltage. Through this pin. The next is 5 volt in the regulated power supply to power. The microcontroller and other components. This can come either from V in via regulator or from the power supply or supplied by USB or another regulated 5 volt supply. The 3.3 volt in a 3.3 volt supply. generated by on board chip. The maximum current drawn is 50 milli amp and GND is the ground pins. So these are some details about the Arduino hardware where we have multiple pins and the Arduino as a microcontroller works at 5 volt supplies. Though it has a regulator on chip which can convert 5 volt to 3.3 volt and this 3.3 volt supply is available on board and can be used to interface some of the sensors. However, the maximum drawn current is only 50 milli amp because the regulator is limited by this current only. So we cannot draw a high current from this pin or we cannot attach a heavy load directly to this pin which is running at 3.3 volt. If the need arises to connect a heavy load we have to use certain buffer circuitries which boost up this signal.

So this 3.3 volt supply is all present and ground is the universal ground which should be connected to whole of the external circuitries and 5 volt supply is always available on this Arduino. It can be derived directly from the USB which we are connecting it to connect to the computer or from a power connector which we connect through a adapter or it can be directly given as a 5 volt to this volt. So these are some basics of these power pins of Arduino. Let us look at a little bit about the basic programming of Arduino. How do we do this programming for Arduino? So the basic programming. We need to declare all the variables at the top. Generally, we have two loops, one is the setup and one is the loop. These are the two statements which are there in all Arduino sketches. That are a part of every Arduino sketch.

The first one is the setup. It is called once when the sketch starts. It is a place to do setup task like setting the pin mode or initializing libraries. For example, this is how a setup a sketch starts. We define a void setup and this loop starts and then we write our statements and then this loop closes. So this is how a sketch starts. The second function is the loop function. And the important part of every sketch. And we need to include both functions in the sketch. Even if we do not need them for anything. This is how this void loop function is. We write void loop function. It starts and then we write our statements and we close the bracket. So here we have two basic functions. One is the setup, one is the loop. Both are important and we need to include them in our program or sketch. Even if we do not use them but we still need to use both of them setup and loop.

The difference between setup and loop is the setup is called only once and this is the program where the sketch will start and end and within this function all the programming will happen. Now if we need a repetitive measurement of programming that is done in the loop function. This function is written separately. This loop function is for example we have a sensor and this is a temperature sensor. We want to measure the temperature every one minute. So setup is the program which is setting up all the parameters. For example the pin numbers with which the sensor is connected. The libraries whatever we use for math or for sensing. So that is done in the setup itself. Then every second, every minute we need to take the measurement of temperature. This is a repetitive process. So this measurement is actually done in the loop statement. Every one minute setup will call the loop function and when the loop function is called it will perform the measurement of temperature from a sensor and gives it reading back to the setup function where it will calculate all the output and then wait for one minute or so and again call the loop function. So setup function is the one which is called only once and the program execute in there and the loop function is a repetitive process which we need to perform repetitively in the program. So there are two main functions.

First is the variables. We have certain variables in the programming language to perform the task. So let us discuss variables. You can use variables in a similar way as they are used in math or physics or even any programming language. As they are used in math or physics. Or in any other programming language. All variables have to be declared before they are used. And optionally set an initial value. Means initializing the variables. So the example is. int LED pin equal to 13. The other example is. integer var equal to value. So we have certain variables in the programming language. And these variables for Arduino it can be in a similar way that they are used in math or physics or even any other programming language. Similar way we can use in Arduino. All variables need to be declared first before we use them in the program. However this is optional to put a initial value or initializing the variable. So for example we have int LED pin equal to 13. So here we are initializing a variable which is LED pin. This is the variable name. And the type of variable is given here int which is integer for short. So LED pin is a variable which is of type integer. And the value we are assigning to this is equal to 13. Means LED pin wherever we will call in the program will represent a value of 13 assigned to it. In another example we have a var which is the variable name which is type integer again.

And the value is equal to val the value. This val we can declare later on or by successive measurements it can come up. So here we are not initializing the variable by its value but we have to declare the variable and its type which is integer here. So the basic programming is the basic programming includes multiple functions such as the digital write function includes a value on a pin. The possible values are low or 0 volt or high which is 5 volt. For example, digital write 13, high and digital write 11. So here we have the basic programming which has a function like digital write. Digital write is a function which outputs the value to a particular pin. We have to specify the pin number in digital write as well as the value. The values are possible of two types only. One is the low which is 0 volt and the other one is high which is 5 volt because the Arduino runs at 5 volt so all the digital high is 5 volt. Now the example of this digital write is digital write 13, high means digital write will enable this pin 13 which is the pin number of in that Arduino board. The pin number 13 will be applied a digital high which is 5 volt so the pin number 13 will have a voltage of 5 volt. In the other example digital write 11, low the pin number 11 will be pulled down to 0 volt because this is specified as low voltage. The next is the delay function. The delay causes the Arduino to wait for the specified number of milliseconds. Before continuing on the next line. So, for example. Delay 1000. This statement creates a delay of 1 second. So we have the another function which is very useful is the delay function. This function whenever it is called it delays the programming of Arduino for that number whatever is written in inside this braces. The example is delay 1000 means whatever is the number given in this brackets that number of milliseconds the program halts it waits for these many milliseconds before going to the next line.

So delay 1000 means 1000 milliseconds it will wait to go to the next line. 1000 milliseconds means is the 1 second time. So after 1 second the next line will be executed. So these are some basic functions we will not cover all the programming how to do Arduino programming and all the sensors connected to it that is a separate course by itself. So we will just focus on basic understanding of Arduino and how it works and what is the basic functionalities. There are lots of other functions as well which are equally important for Arduino. So other useful functions for example the pin mode. This set a pin as input or output. Digital write. That is a digital pin high or low. The third is digital read. This read digital pins state. Next is analog read. This reads an analog pin. Analog write. This write an analog PWM value. The other is delay. This waits for an amount of time in milliseconds. The other function is millis which gets the current time. So there are some other functions which are useful for making a Arduino sketch. The one is the pin mode. This sets the pin as input or output. So as we know Arduino has multiple pins which can be configured as input or output. We can use the same pin to receive the signal as well as to send the signal. So pin mode is the function which is used to set the pin as input or output. If we set the pin as input the external voltage whatever is applied to that pin that will be detected by Arduino. If we set the pin as output then the value will be written by Arduino on that pin and the external circuit is going to receive that value.

So pin mode is the function to assign the value whether it is the pin will be used as a input pin or the output pin. Digital write is the function which is used to enable that pin high or low. So when we write digital write high it will apply a 5 volt signal and when we write digital write low it means 0 volt will be applied through that pin. Similarly digital read is a function which is used to read the value from that pin if that pin is assigned as a receiver or the input type then the external circuit is applying a voltage on this pin and

Arduino is going to read whether this is a 0 volt applied from the external circuit or it is a 5 volt applied on that pin. So these are digital write and digital read means the value can be only 0 or 5 Volt. Similarly we have analog read. Analog read is the function which is used to read the analog value because the sensor which is generally connected to the Arduino they give the output as a analog signal. Analog signal not necessarily need to have 0 or only 5 volt value. The value or the voltage can be in between as well so that is a analog signal. Analog read is the function used to detect how much is the voltage at the pin supplied by the sensor and this voltage will be used for further calculations or the programming.

So analog read is to use to read the analog signal from that pin. Similarly analog write is the function which is used to apply analog value to the pin. Delay as we just discussed last slide. Delay is a function which waits for the program for these many milliseconds whatever we specified in that bracket. For example, if we specified delay 10 the program is going to halt or to wait for 10 milliseconds before going to the next line. Delay 1000 means the 1000 milliseconds or the value is 1 second. The program will wait for 1 second before going to the next level. Millie's is the function which is used to get the current value or the current time in the program. So these are some basic functions which are used for programming in the Arduino sketches. Let us see a program how a program in Arduino look like and how do we make a small programming Arduino sketch or Arduino IDE.

So we take a very small example of traffic light control. So in this program we first initialize some variables which is int. LED red is 13. integer LED green is equal to 11 and integer type LED yellow is equal to 12. So these are the three variables we are defining LED red is equal to 13. These are nothing but the pin numbers what we are going to use. The pin number 13 will be connected to the red light. Pin number 11 will be connected to the green light and pin number 12 will be connected to the yellow light. So we have these three lights for traffic light control. Now we write the setup loop on the setup function void setup and in this we declare pin mode LED red, output because we need to control all the pins from Arduino so all the three pins will be in the output mode. And here we close the bracket. So this is the setup statement in the program. Before this we have defined three variables LED red which is assigned the value 13, LED green which is integer type variable assigned to value 11 and LED yellow which is again an integer type initialized to value 12. Now we need to have two functions in the Arduino sketch. One is the setup function and one is the loop function. Setup function is the function which we generally used only for once to set up all the constraints in the program. Here the setup the constraints we are making pin mode LED is output. So the pin mode what the pin the pin number 13 is used as output pin for throughout this program pin number 11 which is LED green this is also output type pin and yellow which is connected to the yellow pin this is pin is also a output type. All the three pins are of output type. Next is the loop statement. And we write in the same program void loop and we write digital write LED green void comma high. Then we put a delay of 5000 milliseconds it means 5 seconds. Then we write digital write LED green low and digital write LED green low. LED yellow as high. Then we put a delay of further 5000 milliseconds and again write digital write LED yellow as low and digital write LED red as high.

Then we further put delay of 5000. And then we write digital write LED red as low. And then we close the brackets. So this is the void loop which will be executed repeatedly and what we are doing here the first statement is digital write LED green high. So the green light the pin which is associated to this green light will be turned high it means the 5 volt will be applied to this pin and this 5 volt will be applied for 5 second because the next statement is delay 5000 milliseconds. So 5000 millisecond means the 5 second delay program is going to wait to execute next instruction. So for 5 second the green light will be at 5 volt or green light will be high. After 5 second digital write LED green low. So the light will be turned low the green light the voltage applied will be 0 now and in the next statement immediately the LED yellow high. So yellow pin the pin which is connected to the yellow light will be applied a 5 volt signal and further this program is going to halt for again 5 second before executing the next instruction. So the yellow light will be turned on for 5 second. Next after 5 second the yellow light will be turned low and the red light digital write LED red comma high the red light will be turned on high and the program again will wait for 5 seconds. So the red light will be turned on for 5 second and then the red light will be turned low and this loop will execute itself forever. So if we see from outside first the green light will turn on after 5 second it will turn off the yellow light will turn on after 5 second the yellow light will turn on turn off and the red light will turn on and again it will be remain red light for 5 second it will turn off after this and again green light will turn on. So this is the basic structure of a Arduino program and some basic functions we discussed which are used to make this Arduino sketch. Of course this is not enough to make all the Arduino sketches and all functional programs which are used for actual programming of Arduino, other sources can be used to learn the Arduino programming.

This is all for today.

Thank you.