Transducers For Instrumentation

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

Lecture - 32

Smart Sensors: Microcontroller based

Hello, welcome to the course Transducers for Instrumentation. We are discussing about smart sensors and we saw that smart sensor is nothing but a conventional sensor and some extra electronics which is in the kind of microprocessor or a microcontroller. This extra electronics is used to process the data generated by this conventional sensor locally itself so that we extract out the meaningful data from the measured, the raw data and this meaningful data is now next sent to the mesh or to the internet whatever is the destination maybe. So this smart sensor we discussed in the last lecture we have a microcontroller which is Arduino most of the time. This microcontroller can be used to receive the data from a conventional sensor and this Arduino has multiple pins which can be configured as input and output. So input pins if you specify a pin as a input that pin is expected to receive data from the sensor and provide it to the microcontroller.

If you specify a pin as output pin that pin is supposed to send out some signal to control the external circuitry that may be a sensor or maybe some interface circuitry which is controlling the sensor itself. So, we saw microcontroller earlier and we saw that microcontroller especially the Arduino one is very less expensive and easy to use. It is a open source platform where you can apply your own changes to the hardware and software itself and you can modify as per your requirement. So that is the advantage of open-source platform where you are free to choose or free to change the hardware as per your requirement and the Arduino is low cost and very limited in functionality by itself but there are multiple hardware shields available in the market which can enhance the performance of Arduino. For example there is no on-chip communication interface on Arduino, wireless communication is not available on the Arduino board itself. So external Bluetooth or Wi-Fi modules are available you can buy from the market and insert it into the slot of Arduino then Arduino can send and receive data wirelessly to internet. Similarly other hardware's for example different sensors or different relays, controllers all these things are available for Arduino to choose as per your requirement they are easily available very low cost and you can build up your system based on the Arduino the core or the heart of the project is Arduino and on top of that you can put multiple shields to achieve your specifications. So that is the case of microcontroller based smart sensor where we have Arduino which is low cost and for large number of deployments we prefer a open source platform which is easy to use and less expensive. The next kind of smart sensor which use a microprocessor based approach these smart sensors again the sensing

unit is the conventional sensor only which is generating this raw data from the measurement.

Now we have a microprocessor which is sitting there locally to this micro this sensor which is processing this data and finding out the meaningful data and transferring it to the internet or some other destination. Very famous microprocessor based hardware is the raspberry pi or we call r pi in short. Raspberry pi is not an open source platform you cannot change the hardware of raspberry pi by yourself and this is not that much low cost as Arduino is this is little bit expensive platform which is a computer is a computer in itself. Raspberry pi is limited in the performance compared to our normal microprocessors but it can be used as a standalone microprocessor you can make a small computer also out of this raspberry pi. So this raspberry pi is a microprocessor which has almost all the functionality what our normal processor has in our laptop or our desktop PC's or even most of the time in our tablets as well.

So this microprocessor based raspberry pi as we discussed is not an open source platform we are limited by changing the hardware of this board itself and the software. However raspberry pi is also very much popular amongst the students and researchers because this gives us flexibility to easily make a prototype once you have a sensor made in your lab you can easily interface it with raspberry pi and make a complete system using this smart sensor. So let us discuss little bit about what this raspberry pi is and what is the hardware specifications of this. So today we are discussing microprocessor based smart sensors and we will refer raspberry pi or sometime we call r pi in short. So we take some specification of this raspberry pi and we are referring to raspberry pi 3 model B.

So we have this as a system on a chip which is a Broadcom chip this is manufactured by Broadcom company. There is a CPU inside which is ARM Cortex 1.2 GHz is the frequency of operation and this is manufactured by ARM company. RAM or the random access memory is around 1 GB which is LPDDR 2. LPDDR is the short for low power dual data rate RAM which is 900 MHz. There is a storage on the board in the form of micro SD card. So you can have a micro SD card installed in the board. It is also available on the board. This communication is also there which is 2.4 GHz 802.11n wireless standard. There is video output from the card which is HDMI format. GPIO. There is a port 0 40 pin header. There are multiple ports available on the board. For example we have HDMI. Camera interface etc. And the cost of this board is generally expensive compared to Arduino based micro controller. So these are some of the features of Raspberry Pi or R Pi which is a microprocessor based platform. It has a SOC which is Broadcom manufactured this is the company which manufactures this IC. CPU is manufactured by ARM cortex. ARM is the company which manufactures these CPUs. The frequency of operation is 1.2 GHz. This frequency is little less compared to what our laptop generally has.

Laptop and computers they generally have like 3.5 GHz or even more. So the frequency of operation is 1.2 GHz of this Raspberry Pi. We are considering Raspberry Pi 3. The GPU is again manufactured by Broadcom company. The RAM or the Random Access Memory which is installed in this microprocessor or Raspberry Pi. This is generally 1 GHz and this is LPDDR2 which is low power DDR dual data rate. 2 is the version of this kind of memory. It operates at 900 MHz. Storage is in the form of micro SD card. So as such there is no memory available on the Raspberry Pi but there is a slot available to put a micro SD card. You can put micro SD card as per your requirements for example 16 GB or 32 GB. So this Raspberry Pi supports micro SD card slot and this micro SD card act as a memory for all the operations. So if you want to install some software or install OS so all that operation will be performed on micro SD card. Micro SD card will be providing all the memory into the Raspberry Pi. Ethernet this is available on board itself. There is a particular slot for Ethernet. You can connect the Ethernet cable directly in that port. Wireless functionality is also available in RPi which is 2.4 GHz wireless standard. Video output is available on RPi which is in the form of HDMI. So we have a HDMI slot available on RPi. We can directly connect HDMI cable to this port and the display can be taken on a monitor on a TV or other display devices. So HDMI port is there. GPIOs which is general purpose IOs.

So this RPi has 40 pin header. We have 40 pins are available on Raspberry Pi and these pins can be configured as input and output similar to the Arduino what we have the command to set a pin as a input or output. Similarly in different syntax but these GPIOs or these are also configurable as input and output in Raspberry Pi as well. So there is a 40 pin header available. There are multiple ports. For example, HDMI is there for display. 3.5 mm audio video jack which is for the putting the microphone the headphone or so. USB 2 interface is there. We can put our USB devices directly to the board and that can be detected just like our normal computer. Ethernet camera interface is also there. We can buy a camera from the market and directly connect that camera to the Raspberry Pi. Then that camera will be detected by Raspberry Pi. So these are all the functionality the hardware available on the board itself when we purchase a RPi from the market. So all these functionalities comes along on the board which are not generally there in case of Arduino.

So the cost of this Raspberry Pi board is expensive compared to Arduino because of so much of functionality and performance higher performance compared to the Arduino. So these are some hardware details. Let us discuss the interface a little bit. So in Raspberry Pi we have 17 GPIO pins. Out of those foot tip in header we have 17 GPIO pins available. Most of them have alternative functions. Two pins for UART, two for I2C and six for SPI. All 17 pins can be GPIO. They can be assigned as input as well as output.

They all support interrupts. There is internal pull-ups and pull-downs for each pin. And I2C as this I2C particular pin have on board pull-ups. These pins are 3.3 volt, not 5 volt

like Arduino. These are connected directly to inside chips. So, because this board is supporting 3.3 volt only. So sending a 5 volt signal to a pin may damage the R Pi. And the maximum permitted current draw from a 3.3 volt pin is around 50 milliamp. So these are some specification about the Raspberry Pi. There are 17 GPIO pins out of this foot tip in header. These 17 GPIO pins most have alternate functions. So these can be used as a GPIO as well or they can be used directly for other application. For example, communication with the other chips and there are multiple standards of communication. For example, UART or I2C or SPI. So all these pins, 17 pins they can be used standalone as a input output pin or they can be used for this particular applications directly. So out of these all these 17 pins can be set as a GPIO just like the Arduino. In Arduino we can set a pin as input and the output. Here also we can set these pins as input or output and all of these pins support interrupts. Interrupt is something when a system is running a particular a computation is running and a hardware sends a signal to change its course. For example, there is a change in the temperature outside. So the hardware which is the conventional sensor it will change its reading. So this will be detected as a interrupt by the microprocessor and this interrupt will be handled in the normal execution of the program. The program will detect this interrupt and based on its programming how to handle this interrupt that execution will go on. So all these 17 pins on these GPIO pins they supports all interrupts at their end and internal pull ups and pull down for each pin.

So pull up and pull down is a extra hardware connected to the pin in case of high impedance. For example, there is nothing connected at the pin outside. There's no 0 or 1 for example, no 0 voltage or 3.3 volt is applied on the pin. Then this pin is not forced to any particular voltage it can be anywhere between 0 and 3.3 volt. So to avoid this ambiguity of what the voltage is we generally connect a pull up or pull down circuitry based on our application. We want a pull up or a pull down. So this is a weak kind of pull up circuit which pulls the node to higher voltage just in case there is no voltage connected by external circuitry or internal circuitry. So pull up and pull down define the voltage of a pin in case this pin is not driven by any other hardware. So all these pull up and pull down circuitries are generally available on GPIOs to give them a specified value because not defining a voltage is very harmful for a receiver because if the input is in high impedance the receiver will detect it as a high impedance and there will be a constant burning of power at the receiver stage.

So we generally provide a pull up or a pull down circuitry to each and every pin. So this is all available there in Raspberry Pi. I2C pin has onboard pull ups. So I2C is a special kind of standard where we transfer the data in this particular format which is I2C or Inter-chip Integrated Circuit which is the kind of full form of I2C. This is a special communication scenario where we have only pull down circuits available on the hardware and the pull up is generally provided by the weak pull ups which we use for pull up and pull down. So a dedicated hardware to pull down is available but for pull up

we generally use inbuilt pull ups on the ICs which define the node to high voltage and when we want the node to bring down to 0 then we activate our regular hardware which we put as a I2C hardware. So this I2C pin has onboard pull ups. Pins are 3.3 volt not 5 volt like Arduino. So the Arduino the voltage level was 5 volt and all the input output pins were 0 or 5 volt range. Here in case of Raspberry Pi the pins are 0 or 3.3 volt. These are not 5 volt compatible. So 3.3 volt is the maximum voltage which you can apply to these pins of a Pi. They are connected directly to inside chips.

These pins are generally routed directly to the core chips which are working at 3.3 volt and sending a 5 volt to a pin may damage the R Pi by itself. So if we apply a 5 volt instead of 3.3 volt and because these pins are directly connected to the chip inside this 5 volt is enough to overstress the circuitry which is placed inside the chip and that can get damaged because of this 5 volt supply. So, we cannot connect a 5 volt sensor or any 5 volt hardware directly to a Raspberry Pi. We have to down convert this from 5 volt to 3.3 volt. There are multiple hardwares available in the market which you can directly use. On one side you apply a 5 volt signal and the other side you get a 3.3 volt signal directly from those hardwares. So these are all available for R Pi and the maximum current which you can draw from a pin is limited to 50 milliamp. We cannot put a very heavy load directly with the GPIOs because the maximum current capability is 50 milliamp. If we need higher current, for example we want to run a motor using a Raspberry Pi. Motor takes let us say 100 milliamp of current. So we need to put a buffer which provides this much of current of 100 milliamp so that this motor turns on and this buffer generally take very less current from a Raspberry Pi and generate the required current from DC power source or the external supplies. So the maximum current is limited to 50 milliamp. So the next step to use R Pi is installing and setting up the operating system for Raspberry Pi which is nothing but called the Raspbian. Which is an operating system for R Pi. So as we discussed the Raspberry or R Pi this is exactly like our normal laptop or desktop computer. This has almost all the functionality which our laptop or computer has. Similar to the operating systems what our desktop computers has Raspberry also works on the operating system unlike Arduino. Arduino there is no operating system and we simulate we kind of execute our program in our normal PC and when it works we put this program or this algorithm directly into the Arduino which keep on keep on iterating that program.

There is no operating system in Arduino. Here in Raspberry Pi we have a dedicated operating system which is called Raspbian which act as an operating system. Operating system here is needed because the operating system is the interface which connects the hardware to the software what we are going to make. For example we want to make a program for traffic light signals. So this program is directly not executable on the hardware so operating system provides an interface where this program will be converted back into a compatible language which hardware can understand. So operating system is needed for R Pi unlike Arduino and this operating system is called Raspbian. This

operating system is available from the company directly and we can download this directly from the official site. So, to do this go to official raspberry pi download site and download the Raspbian. We need to write the OS. It can be used by R Pi. We need to boot the R Pi for the first time and perform the first-time configuration setup.

This includes typically partition, password, file system etc. Then we can connect to Wi-Fi network. So, this is a brief instruction on how to start raspberry pi. So once we have the R Pi hardware with us we can go to the official R Pi site to download the version of Raspbian whatever is the latest one available on internet which we can download. We save this file this operating system file directly to the SD card which is there in the R Pi so that it can be used by raspberry pi. Then we need to boot the R Pi for the first time and perform the first configuration setup for example partition how the partition we need to make in the micro SD card, password setting or the file system what kind of file system we need to use in that micro SD card and then we can connect to Wi-Fi network. Of course these instructions are not complete because discussing the use of raspberry pi and how to use raspberry pi to make a smart sensor that is not the scope of this work. This is just for the knowledge of the viewer that we can use a raspberry pi to make a smart sensor. So, once we setup the R Pi or the raspberry pi with operating system the next step is to download the python development environment. So generally, we use python as a language to program the raspberry pi. Python is freely available on internet we can use any Linux based operating system and the python generally comes with this distribution.

Python is a language which is used for programming these raspberry pi. So I strongly encourage the viewer to learn python small programs how to make a small program in python and execute it using python and how to put this program into a raspberry pi to perform a certain operation. So the next step of this is python tool installation. Python development tool installation which typically includes libraries, install integrated development environment. Or we call IDE and we have command line terminal. So, once we setup R pi with Raspbian which is the operating system then we setup the python environment in the raspberry operating system so that we can use python as a language for programming this R pi module.

Python development tool actually contains some libraries which are needed for the execution of program. Integrated development environment which is a dedicated IDE or the environment which we use to write the program and execute it and we have a command line terminal as well so that we can directly write the command and send them to the hardware. So, python development environment is needed to program the raspberry pi. So let us discuss little bit about the python programming language. So, python is very easy computer language which is very the most many of the syntax are common to other programming languages. So, we will discuss very briefly about the python. So, control flow and data types. So, in python once a variable is given a value its type is automatically set and will not change unless we reassign two different types. Once a

variable is given value its type is set. And will not change until reassigned. And a variable can hold value of any type. Type is specified by value and not by the variable. For example, we have numerical type. integer which is 32-bit precision. And A equal to 4 is a typical assignment for integer type. The other is long which is unlimited precision. We can assign for example A equal to any value. So, this is a long type. The next is the float type which is the floating-point number. A equal to let us say 6.9758 whatever the value is. And we have complex type which is complex number. A equal to 5, 6. So this A is now 5 plus 6i. So, in python we have a variable which is given a value. So as soon as we assign a value to a variable its type is set by that assignment, and it does not change unless we reassign it to a different type.

For example, we have here the integer type which is 32-bit precision and when we have a variable for example A equal to 4 or A equal to 5. So, this assignment of A to 4 this dictates the type of variable A. Now by A equal to 4 python knows that the variable A is assigned a value of 4 and the type of this variable is an integer. Similarly we have another type is long where we assign a very long number to a variable A equal to 1000 and so on. So then the python knows that this variable is of long type. Similarly float is the floating point number when we have a number which is having a decimal places. So, A equal to 6.9758 when this number is assigned to a variable the python knows we need not to explicitly declare the variable type. Python knows by assigning this number to variable A that the variable A is of floating type in nature. The next is the complex number where if we assign A equal to 5, 6 then python knows that variable A is of complex type.

So we need not to assign when we declare the variable. However, the type is fixed now we want if we want to use the same variable as a instead of integer now we want to use as a float so we need to reassign this variable as floating type. So this is the difference. So these are the some numerical type. Next the first assignment to a variable actually created the variables type do not need to be declared. Python figures out by its own. Therefore, the assignment is equal to and comparison is. Double equal to and for numbers plus minus multiply divide etcetera. For logical are words for example and or not etcetera. So here we have these variables which need not to be defined initially and whatever is the assignment to those variable python knows by itself that this is of the type which is assigned to this variable. The next the assignment is equal to when we assign a value to a variable we put A equal to 4 but when we are conditionally checking whether these two are equal then we write double equal to which is very common practice in all the programming languages and the python follows the same similar kind of syntax as other programming languages. For example, in other programming languages well when we add two numbers or multiply two numbers we use plus or star sign for numbers and for logical operators when we are performing and operation or not operation then we generally write and in words A and D and in words so that the python knows that this is a logical operation on these two variables. So this is how the python works this discussion

is not complete python language is very big kind of very kind of involved language and I advise to study the python language from some other materials this course is not to teach the python language but we will discuss only very basics of python language how to use the python language and raspberry pi to do a smart sensor to make a smart sensor and to perform certain operations. So let us see one example based on python and we stick back again to that traffic light signal what we just saw in the Arduino case. So here we are using raspberry pi program in python. So in this case we have a raspberry pi hardware which has multiple GPIO pins and I choose let us say few pins to make this street light controller. This is for example the GND pin or the ground pin which I connect to the universal ground of external circuit.

I connect to the red LED GPIO gate. I connect to green LED and GPIO 7. I connect to the green LED. And GPIO 21 I am keeping for the external control like a button push button also. Now python program for such arrangement is something like this. traffic lights buzzer from time import sleep. Buzzer is equal to buzzer at pin 15 button is equal to button at pin 21 and lights at pin 23 is equal to traffic lights at 25, 8, 7. button wait for pass button wait for press then buzzer on then light dot green. On then sleep for one second then light yellow. On then sleep for one then light dot red. On then again sleep for one and then lights off and then buzzer off. So this is a small example of python code which can be written to control this traffic light. This has a push button which we press and when we press this button this while loop is executed. What this while loop is doing? This is turning on these red, green and yellow lights LED lights within a delay of one second because we are putting a sleep of one second. So all these functions these are actually imported from certain libraries and if we see the second line for example from time import sleep. So sleep is a very well defined function which python knows very well. Whenever we call this function sleep python knows that the program need to be halted for that many seconds whatever is the argument written in the sleep function sleep bracket one then it means for one second the program need to be halted or stayed idle for one second. This function is not readily available in all the programs whatever function you need that is available in particular libraries for example math functions are generally available in maths. Sleep function is available in time library those libraries we need to import in our program so that python knows these functions and perform the computation accordingly. So, this is a example of python program of course this is a very small program for real sensor development or a particular smart sensor we need to expertise on the python programming itself as well as the hardware connection the electronic circuitry that also that particular knowledge need to be acquired from other sources as well to make a complete smart sensor.

So, this is all for today.

Thank you.