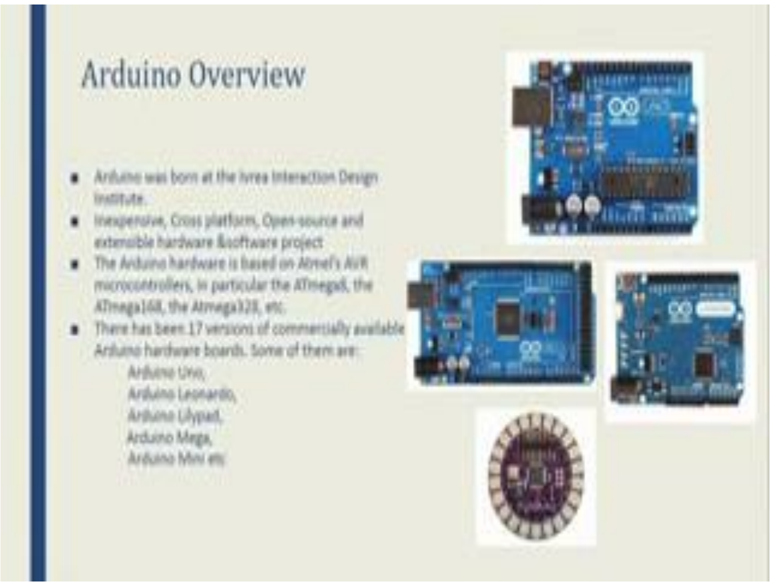
Embedded Systems Design

Prof. N. Vidya
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 19
Tutorial – V

Hello everyone. In today's tutorial we are going to learn about Arduino and very basics of it; it will be a very short tutorial. And I will show you at the end some of the examples program that will run on the Arduino board.

(Refer Slide Time: 00:42)



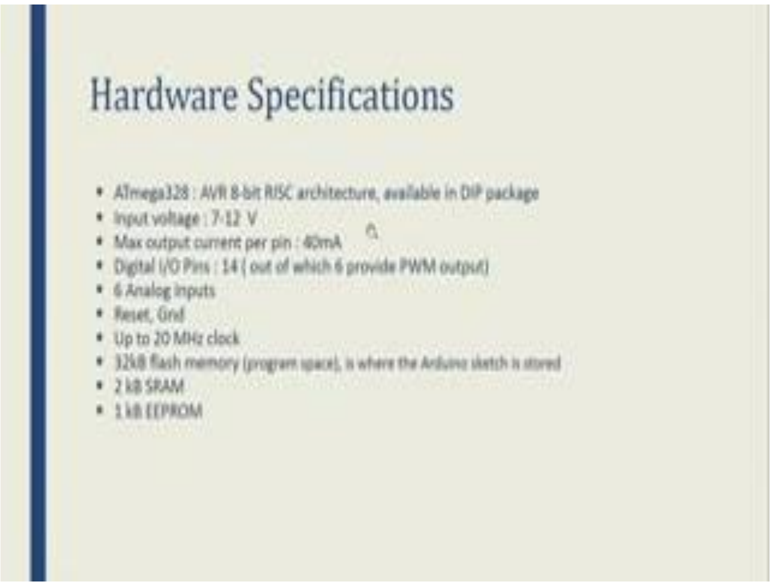
Arduino Overview

- Arduino was born at the Ivrea Interaction Design Institute.
- Inexpensive, Cross platform, Open-source and extensible hardware &software project
- The Arduino hardware is based on Atmel's AVR microcontrollers, in particular the ATmega8, the ATmega168, the Atmega328, etc.
- There has been 17 versions of commercially available Arduino hardware boards. Some of them are:
  Arduino Uno,
  Arduino Leonardo,
  Arduino Lilypad,
  Arduino Mega,
  Arduino Mini etc

So, just to give an overview Arduino was born at Ivrea Interactive Design Institute in Italy. It was developed by a team of four people in Italy; it was intended to use as a inexpensive open source extensible hardware and software project, so anyone who has very little knowledge about the electronics involved can use Arduino for their project with minimal coding and electronics knowledge. So, Arduino is a hardware based on Atmel's AVR micro controller it is in particular based on ATmega 8, ATmega 168 and 328 series.

And there are currently seventeen version of Arduino hardware that are available. So, since it is open source they are multiple vendors who have taken up the source code and schematic, and they have produced their own version of Arduino. So, the board that we have currently and the board on which I will show the demo used Arduino UNO board and it has ATmega 328 AVR micro controller. So, there are other boards like UNO, Leonardo, LilyPad, mega mini etcetera, which you can see on board. There are many more which I am mentioned, but they are been used for specific purposes.

(Refer Slide Time: 02:16)



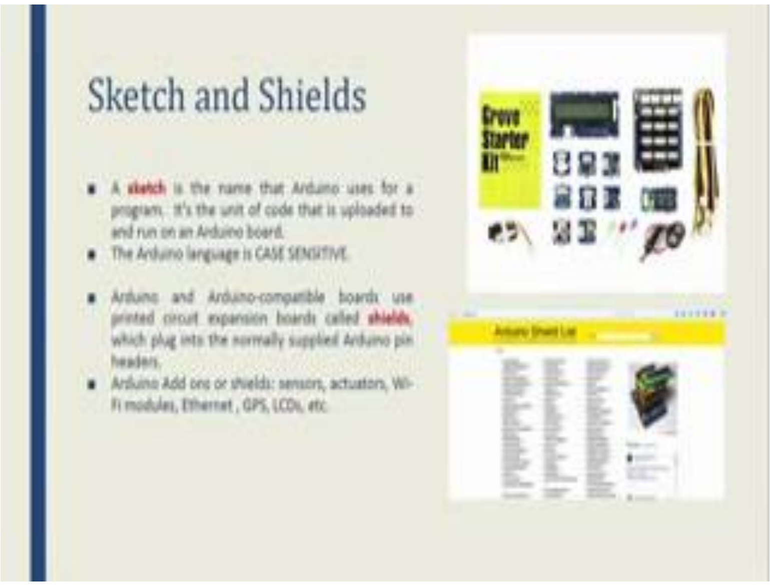
Hardware Specifications

- ATmega328 : AVR 8-bit RISC architecture, available in DIP package
- Input voltage : 7-12 V
- Max output current per pin : 40mA
- Digital I/O Pins : 14 (out of which 6 provide PWM output)
- 6 Analog inputs
- Reset, Gnd
- Up to 20 MHz clock
- 32kB flash memory (program space), is where the Arduino sketch is stored
- 2 kB SRAM
- 1 kB EEPROM

So, I will just quickly review the hardware specifications; and I have there on the Arduino board. So, we have an 8 bit RISC architecture ATmega 328 which is available in DIP package, so that we can remove it and put another input. The input voltage at the board takes 7 to 12 volts; it has a battery input like we can put an adapter and power return or we can power it on using the USB serial port. Maximum current output that each pin of the board has as 40 milliamperes, and there are total of 14 digital I O pins, which we can use and out of which 6 of them provide PWM output that is pulse width modulation.

And there are six analog inputs that are available through which we can connect different sensors and actuators. We have reset and ground; and there is a non board crystal oscillator which can provide up to 20 megahertz clock frequency there is the 32 kilobyte flash memory which is used to show the program of the Arduino board that is the space where the sketch of Arduino is stored. I will discuss about sketch in later slides. We have a 2 kb SRAM and 1 kb EEPROM which is used for storage and a programming point of view. So, this was a basic hardware description of Arduino board.

(Refer Slide Time: 03:52)



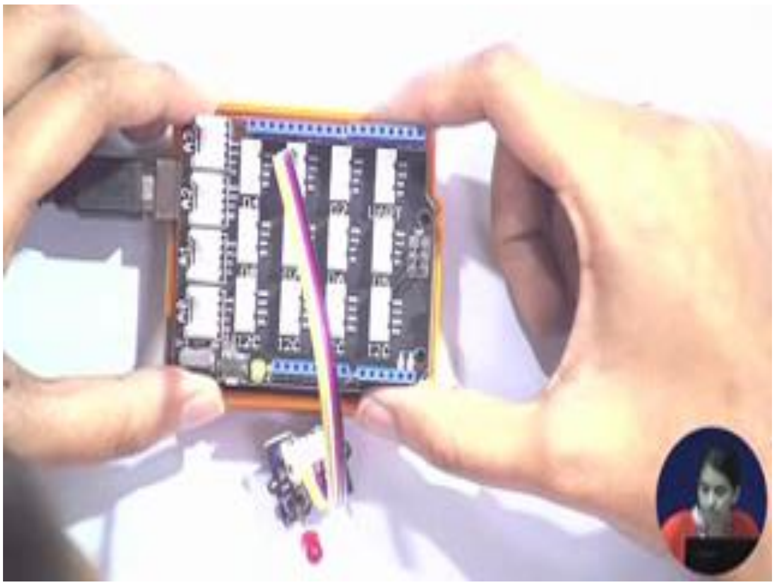
Sketch and Shields

- A **sketch** is the name that Arduino uses for a program. It's the unit of code that is uploaded to and run on an Arduino board.
- The Arduino language is CASE SENSITIVE.
- Arduino and Arduino-compatible boards use printed circuit expansion boards called **shields**, which plug into the normally supplied Arduino pin headers.
- Arduino Add ons or shields: sensors, actuators, Wi-Fi modules, Ethernet , GPS, LCDs, etc.

So, I was told we have sketch and shields that are involved with Arduino boards. So, sketch is a Arduino program that we write in Arduino based projects sketch means a program that we write which is being uploaded into the board. And the languages that we use two program like the sketches a language which is very case sensitive. And shields are the printed circuit expansion board which we use like the input and output like sensors, actuators, Ethernet modules, Wi-Fi modules LCDs etcetera are called the shields which are basically the peripherals using which we make project out of the Arduino board.

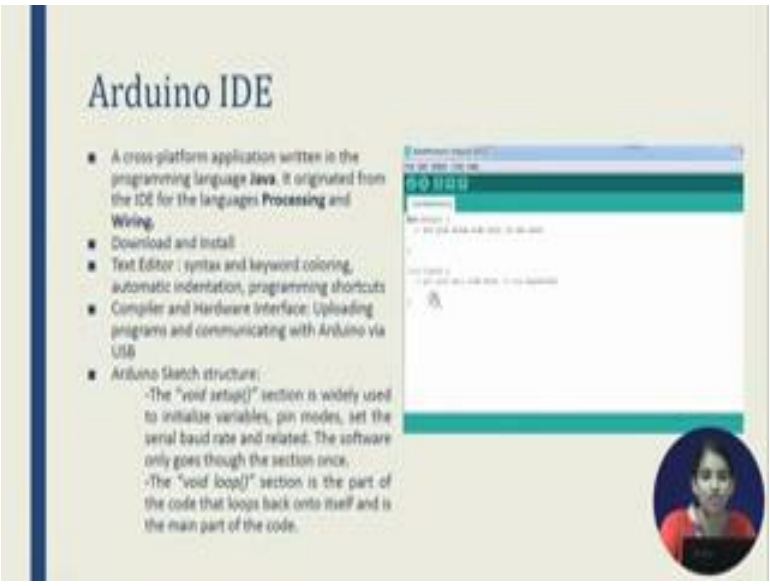
So, these are mostly Arduino based products are used in I O t kind of amendment nowadays and for DIY projects. So, in this tutorial, I have got groves starter kit which you can see here, this is a groves starter kit provided by seed studio. So, generally we have a breadboard kind of a circuit, and we connect peripherals into the Arduino directly using breadboard, but it requires basic electronics knowledge. So, this is a starter kit to give hands onto a new person who is very new to Arduino, so that we can just plug and play and just check out all the functionalities that Arduino provides.

(Refer Slide Time: 05:35)



So, we have this base shield board which brings out all the pins that are available almost of the pins not all the pin most of the pins, and we connect peripherals shields like this. This is an LED shield through which like by programming Arduino to do’s a LED blinking and etcetera kind of stuff. We can use these shields for our projects for our different functionalities. So, there are apart from this seed studio kit, there are different kinds of kits and different sheets available in the market. This is not the only one, we can we can change it according to our requirements. So, this is just the basic thing that we have in the lab. So, I got it down here. So, let me have a quick look at it.

(Refer Slide Time: 06:27)



Arduino IDE

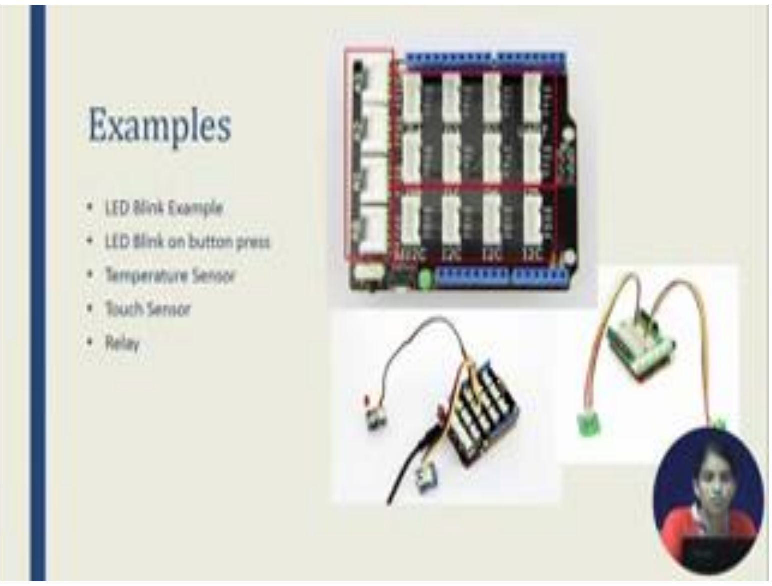
- A cross-platform application written in the programming language **Java**. It originated from the IDE for the languages **Processing** and **Wiring**.
- Download and install
- Text Editor : syntax and keyword coloring, automatic indentation, programming shortcuts
- Compiler and Hardware Interface: Uploading programs and communicating with Arduino via USB
- Arduino Sketch structure:
  - -The “void setup()” section is widely used to initialize variables, pin modes, set the serial baud rate and related. The software only goes though the section once.
  - -The “void loop()” section is the part of the code that loops back onto itself and is the main part of the code.

So, now will come onto how do we program the Arduino board, so that it does what we want it to do. So, we have an Arduino IDE which looks something like this. It is a cross platform application written in java and it was originated from IDE for the languages processing and wiring. So, wiring was developed by one of the students of the (Refer Time: 06:59) who developed Arduino project and later on it got copied and it was used inside the Arduino IDE. So, Arduino IDE is rarely available in online Arduino has its own site we can download and install it on all the Oss that are available currently, it is crossed platform. So, we can use a cross different Oss.

It has a text editor as you can see and it has syntax and keyboard coloring as we have in see automatic indentation and programming shortcuts. So, the IDE itself involve has compiler and hardware interfaces through which we can compile, verify and upload the program onto the Arduino via USB serial port. So, it has a serial port interface embedded into it. So, a basic Arduino sketch structure is what I am going to explain now. So, bare minimum structure as shown in the figure, sorry as shown in the figure consist of two main parts void setup and void loop. Void setup is used to initialize variables and pins a like pin modes that whether it is input or output; and set the serial board rate at like the serial communication protocol has a board rate associated with it.

So, board rate is a bits per second frequency. So, all those initial parameters initial setting of the code is written inside the void setup part of the sketch. In the second part - void loop part, we put the code that has to be repeated onto itself; and it is generally the main part of the code that is used in a Arduino sketch. So, this is the basic Arduino IDE.

(Refer Slide Time: 08:56)



Examples

- LED Blink Example
- LED Blink on button press
- Temperature Sensor
- Touch Sensor
- Relay

So, that was it that was the main thing that you need to know about Arduino to get started with it. So, I will try to give examples as much as possible. So, these are the examples that I planned to cover. A LED blink example, LED blinks on button press and temperature sensor. So, this is the base shield that I talked about. We have out of the 14 digital pins we have these pins are 8 to 9 pins are brought out; some of the I 2 C lines are brought out U Art pins are brought out, and these four analog pins are brought out. And you can also directly access all these Arduino pins from this blue header, so which is directly brought out.

(Refer Slide Time: 10:04)

So, I just open the IDE, so that you can. So, this is the basic Arduino IDE structure, I have already opened the blink program. We have file, edit, sketch, tools and help in the menu bar. There are examples that are already available in the IDE. So, you can see that we have in basics we have blink re-digital, re-analog voltage all kind of basic stuffs that are that are present for you to test. So, the basic built-in examples are already there in the IDE, you can check it out whenever you have the board. And sketch is the verify, upload and the programming module, where we verify the sketch code. We upload it on to the board and attach the functionality.

So, in the tool section, we have as we discussed at a different kinds of board that are available. So, before we connect a board, we need to specify which board we are using. So, in our case, we are using Arduino genuino UNO board. We have to specify the COM port that we are using for serial communication, which is COM 5 at in the present case. So, apart from the download of a download of the IDE, we have to make sure that the USB driver is install properly, so that there serial compute is audited correctly. So, once we have set up the board and the port we are to go and compile download and upload the program.

(Refer Slide Time: 12:03)

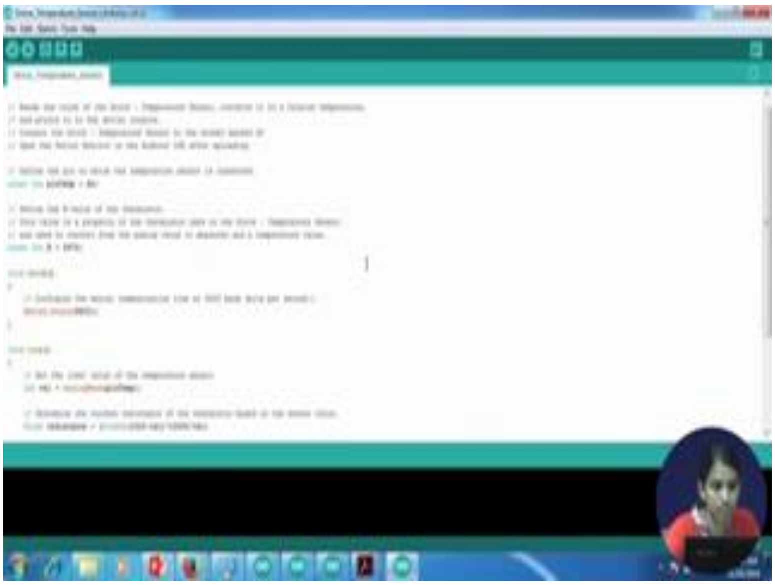
So, I will just compile this blink program to the code is compiled. As you can see in this we have two sections void set up and void loop. In void setup, we have put LED as we have mentioned LED as an output; and here we are writing high and low alternate in high and low to the LED output. To this particular pin, we are writing high and low with the delay of thousand. So, we can see the LED blinking on them. So, let us upload. So, this is connected onto the 13th this blink program operates on the 13th pin of the board. So, if we connect, so you can see that the LED is blinking as expected. So, this is the basic very basic LED blinking program that is there.

So, now, I will just use the base shield and the LED shield, and there is one more shield called button shield which we can use. So, if you can see there are some 4 to 5 shields that are available with the seed studio kit we have rotary module, we have a temperature sensor, we have light sensor, we have buzzer touch sensor, and there is a relay there is an LCD module. So, all these can be interfaced using the base shield.

So, I will just demonstrate the basic LED code using the base shield like I just the same blinking LED code using the base shield LED module. So, this is a grove LED code. This the LED is blinking, but it is not visible because of the light here. So, this is blinking according to the code that and according to the delay that we have provided; we have provided 2 millisecond delay, and the code is running in a loop, so that there is an LED that is blinking. So, in this configuration, we are using pin tree of the Arduino board which is PWM pin that is pulse with modulated pins. So, it make the width of the pulse varies according to the code, so that is how this LED behaves according to the width of the pulse LED output also varies. This is a basic grove button an LED setup, when I press the button the LED blinks.

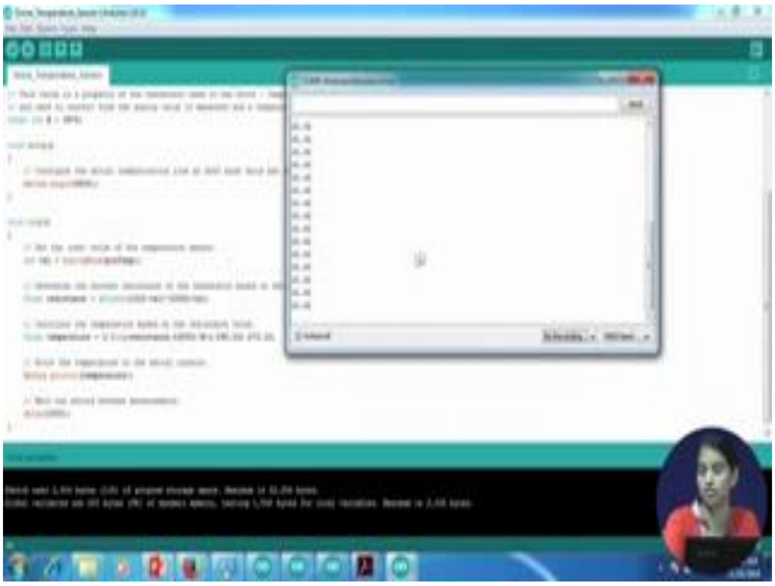
So, again the setup we have mentioned what is the input what is the output. So, we have connected button to the input d 3 and LED output is given out at d 7, so that we have mentioned in the pin mode of setup; same thing digital write and digital, digital write is used to write the read input from the button to the LED. So, we used two function digital read and digital write, so that when are there is a press in the button we get the LED output. This also would not be visible because of the light. So, I do not know if you can see, but whenever I press the button there is an LED.

(Refer Slide Time: 17:48)



I will show you the temperature sensor module. So, temperature input is an analog input. So, we connected to the analog port. We have this core where in at the a 0 connector we connect the temperature sensor shield; and we defined the serial port board rate which is 9600. The temperature is read using the analog read function; and according to the resistance of the temperature sensor, we get a temperature value based on like based on the resistance of the IC, we get the temperature value output.

(Refer Slide Time: 19:48)



So, as you can see the temperature values are printed out. So, this is the serial port output window like we have putty and hyper terminal and all, same thing, this is integrated into the IDE then Arduino IDE we have just specified in the core what is a serial port bordering. So, this is the temperature the leading that we are getting according to the code that we have written. So, these are the basic codes sketches that I have discussed. So, if you have any more concerns and any more projects that has to be done, please try to explore all this, and there are more shields available like we can have a Wi-Fi module, we can have a servomotor, we can have a wheel.

So, we can make a robot out of it we can use it for an IO t device like to sense the temperature outside and send some important data to the cloud something like that. So, there are different applications of Arduino I will just given the basic idea of how Arduino works.

(Refer Slide Time: 21:14)

So, before completing, I will just want to show you how Arduino looks. So, this if I remove the shield, this is how the Arduino board looks. The shield the base shield came over these pins, and it was available there also. And the first LED blinking that I showed was on the 13th pin and that LED also there on board. So, this is the basic Arduino board. And this is the Atmel chip that is in DIP package we have 6 analog ports here and 14 digital pins here with brown and red pins. We also have a I2C line or (Refer Time: 21:52) line through which we can control I2C peripherals. And this is a adapter input in case we are using a battery power we are using Arduino as a Starlino device we can use there is adopter and it on without the help of USB cable. And this is the 16 megahertz crystal let I mention the board has. And we have a reset push button to reset the program that is it. So, if you have any doubts, you can always get back. And in the references I have kept all the links that are useful for the development.

Thank you.