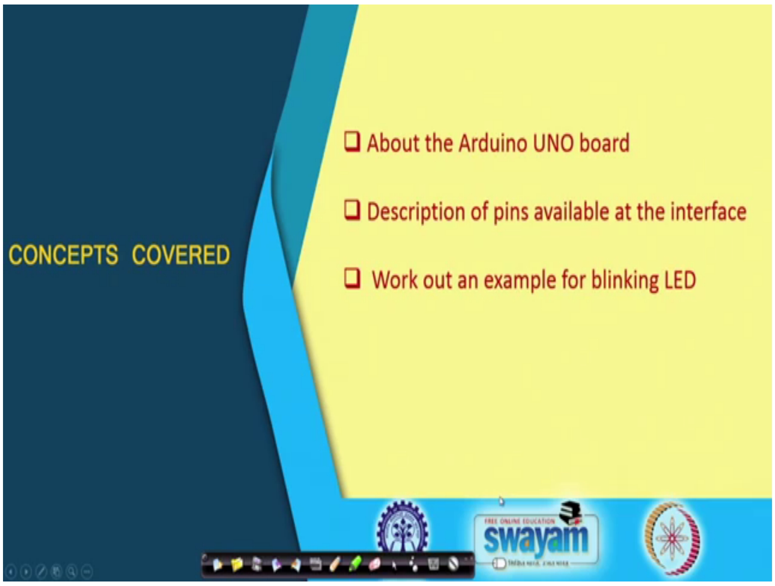
Embedded System Design with ARM
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institute of Technology, Meghalaya

Lecture – 21
Interfacing with Arduino Uno

Welcome to lecture 21. In this lecture I will be showing you how we can Interface with Arduino UNO, one of the Arduino boards very popular one. In the previous lectures I have specifically shown you how we will be using with STM board and here I will take you through the steps that are required to use Arduino UNO board, which is again fairly very straightforward. I will just show you some of the features of this board first and then what are the steps that are required to actually interface with Arduino.

(Refer Slide Time: 01:08)



CONCEPTS COVERED

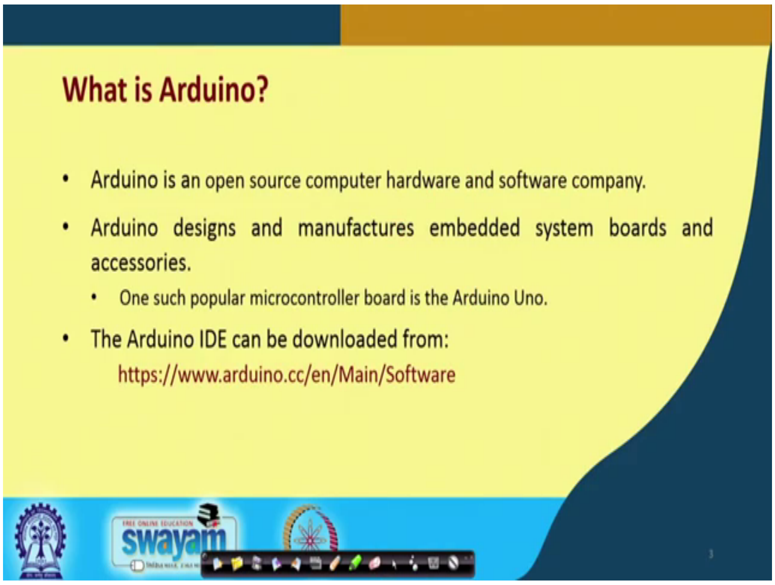
❑ About the Arduino UNO board

❑ Description of pins available at the interface

❑ Work out an example for blinking LED

As I told you, I will be talking about Arduino UNO board description about the pins that are available and also and workout example for blinking LED; although in this experiment, I will not interface it, but in subsequent slides you will find how do we interface it with the board. Today I in this lecture, I will be specifically showing you the steps that are required to use Arduino UNO board.

(Refer Slide Time: 01:41)

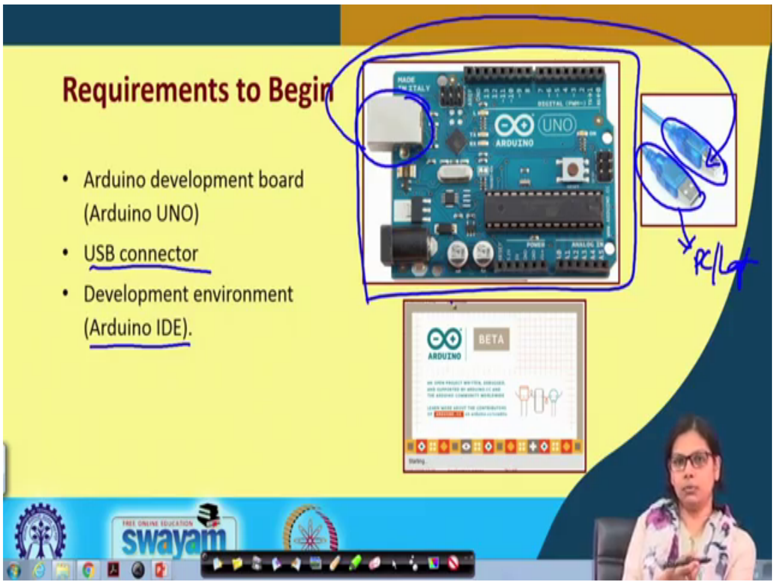


What is Arduino?

- Arduino is an open source computer hardware and software company.
- Arduino designs and manufactures embedded system boards and accessories.
  - One such popular microcontroller board is the Arduino Uno.
- The Arduino IDE can be downloaded from:
  https://www.arduino.cc/en/Main/Software

So, what is Arduino? It is an open source computer hardware and software company. It designs and manufactures various embedded system boards and accessories. And, one of the very popular board is Arduino UNO ok, which is used by student community across the world to design various small embedded system projects. This Arduino IDE can be downloaded from this particular site, you can directly go there and you can download from that particular site.

(Refer Slide Time: 02:26)

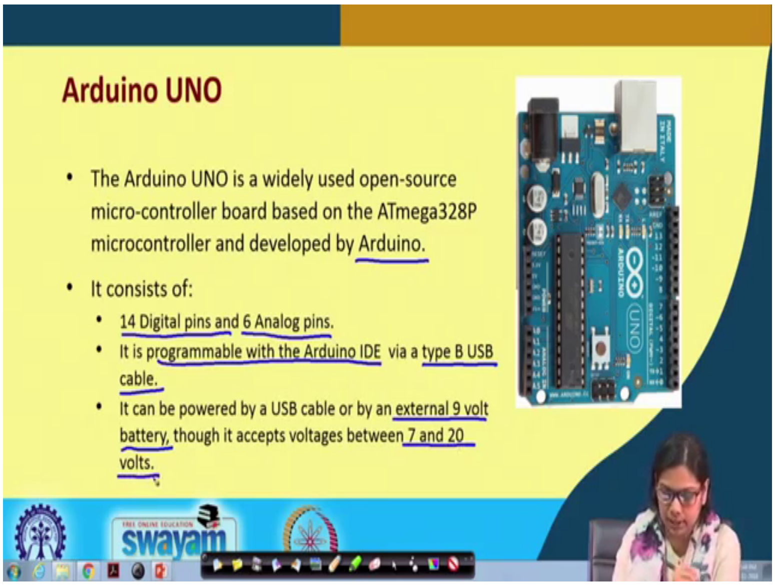


Requirements to Begin

- Arduino development board (Arduino UNO)
- USB connector
- Development environment (Arduino IDE).

So, what are the requirements? For STM32 board we have already seen: what are the requirements that are there. Now, for Arduino board again the same be this is the board in place. It depends on how do we you interface, but the most easy way is like you connect through this USB connector along with your PC. So, this jack is connected here and this jack is will be connected to your PC or laptop ok. This is how it is connected, this is the USB connector which is required and the development environment. The development environment, that are used is Arduino IDE, which looks like this which can be downloaded from the website I have already discussed ok.

(Refer Slide Time: 03:24)



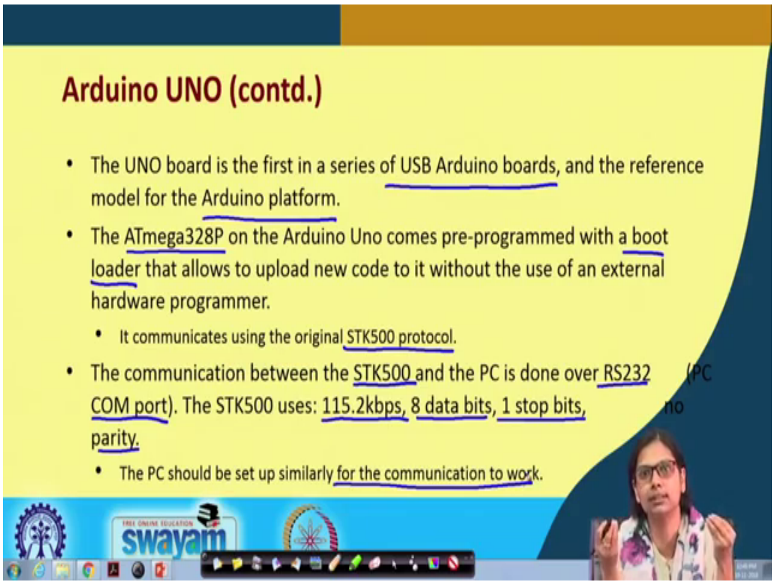
Arduino UNO

- The Arduino UNO is a widely used open-source micro-controller board based on the ATmega328P microcontroller and developed by Arduino.
- It consists of:
  - 14 Digital pins and 6 Analog pins.
  - It is programmable with the Arduino IDE via a type B USB cable.
  - It can be powered by a USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts.

So, this is Arduino UNO, which is widely used open source microcontroller board, based on ATmega 3 2 8 p microcontroller and is developed by Arduino, this company. What it consists of? We have seen now, we have already seen the STM board right in the previous lecture, how we can program. There 14 digital input output pins are available, those are all Arduino compatible pins.

So, these are the following pins that are there, 14 digital pins and 6 Analog pins and it is programmable with Arduino IDE via this type B USB cable which I have already shown. It can also be powered by USB cable or by an external 9 volt battery, let us say you have already dumped the program and you want it to run without the support of any PC anywhere. So, how do you power it? You can do that using this external line whole battery; though it except voltages between this particular range 7 to 20 volt.

(Refer Slide Time: 04:40)



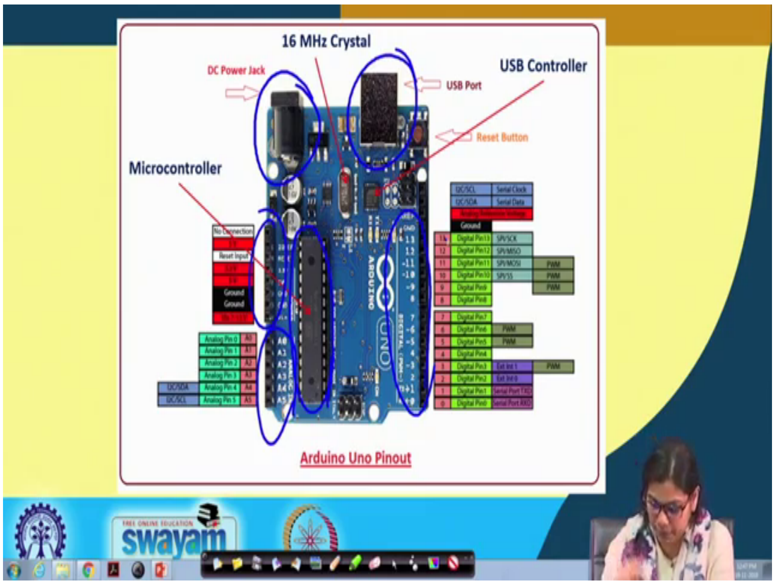
Arduino UNO (contd.)

- The UNO board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform.
- The ATmega328P on the Arduino Uno comes pre-programmed with a boot loader that allows to upload new code to it without the use of an external hardware programmer.
  - It communicates using the original STK500 protocol.
- The communication between the STK500 and the PC is done over RS232 (PC COM port). The STK500 uses: 115.2kbps, 8 data bits, 1 stop bits, no parity.
  - The PC should be set up similarly for the communication to work.

This board is the first in the series of USB Arduino board, earlier boards you need to dump it through some other means not through this USB's. But, when this USB compatible boards came in Arduino, this one Arduino UNO is the first of those kinds of board and the reference model for the Arduino platform. The ATmega328P on the Arduino UNO comes pre-programmed with a boot loader that allows to upload new code to it without the use of any external hardware programmer. So, we do not need to have any external hardware programmer; rather if when we connected this inbuilt boot loader is already there.

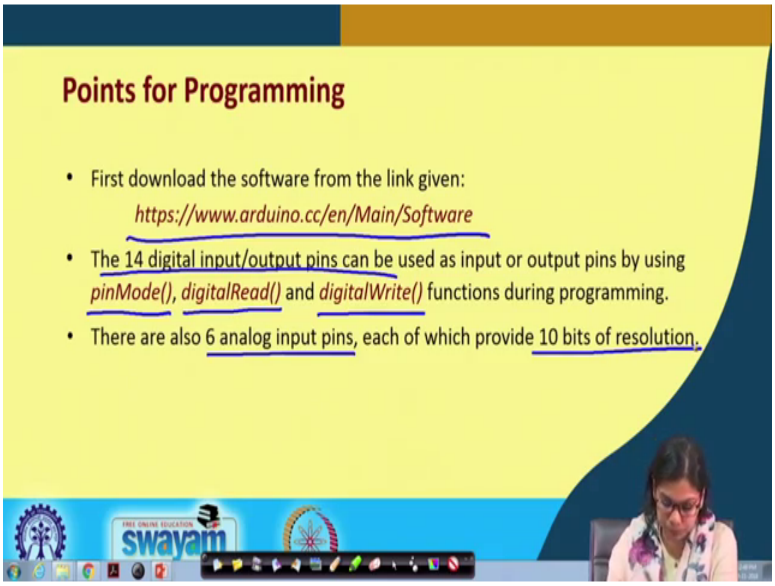
So, it communicates using the original STK500 protocol, this is a protocol that is used for this communication. The communication between this and the PC is done over this RS232 compote and this STK500 uses 115.2 kbps, 8 bit of data, 1 stop bit and parity. The PC should be set up similar for the communication to work. So, when 2 devices communicate with each other, if one device runs at a higher rate other device does not runs at that rate. So, you need to have some kind of means of communication in between them. So, this device must know at what rate this the other device is communicating. So, for that purpose this is required.

(Refer Slide Time: 00:34)



But this is the overall board, you can see, I have already discussed about this power jack, this is the USB controller USB port through which, it can be connected. This is the ATmega microcontroller that is there, this is the digital power supply pins, this is the analog pins and these are the digital input output pins. We will be looking into when we perform experiment with these particular pins, this particular board, we will be doing a couple of experiments with this board. So, there we will be specifically looking into the various aspects of these particular pins.

(Refer Slide Time: 07:24)



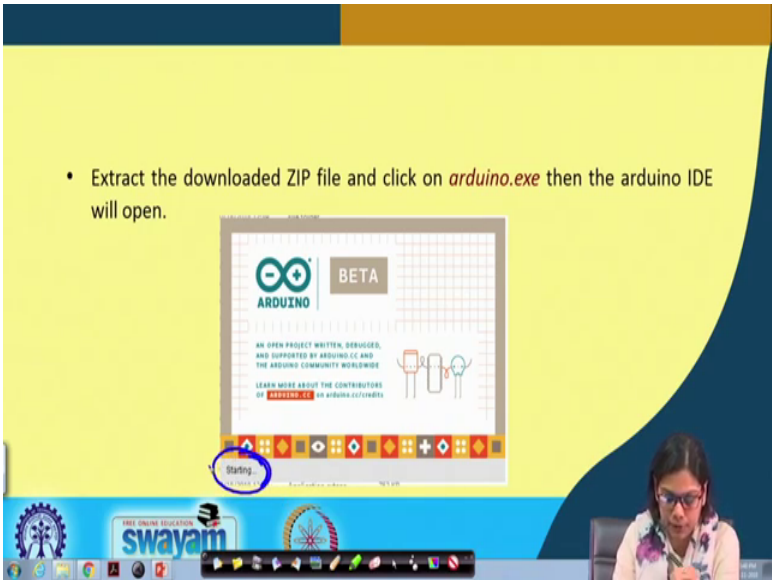
Points for Programming

- First download the software from the link given: *https://www.arduino.cc/en/Main/Software*
- The 14 digital input/output pins can be used as input or output pins by using *pinMode()*, *digitalRead()* and *digitalWrite()* functions during programming.
- There are also 6 analog input pins, each of which provide 10 bits of resolution.

So, what are the points that you need to remember for programming using this Arduino? The first and foremost thing is that you need to download the software the IDE. And for using the 14 digital input output pins, we need to know the following functions: One is spin mode, another is digital read for reading the input, digital write to write the value into the port.

So, one is the output, one is the input, functions during the programming. There are also 6 analog input pins, each of which provide 10 bits of resolution. We have already discussed about the resolution and other aspects of analog to digital converter. So, this is 10 bit resolution.

(Refer Slide Time: 08:22)

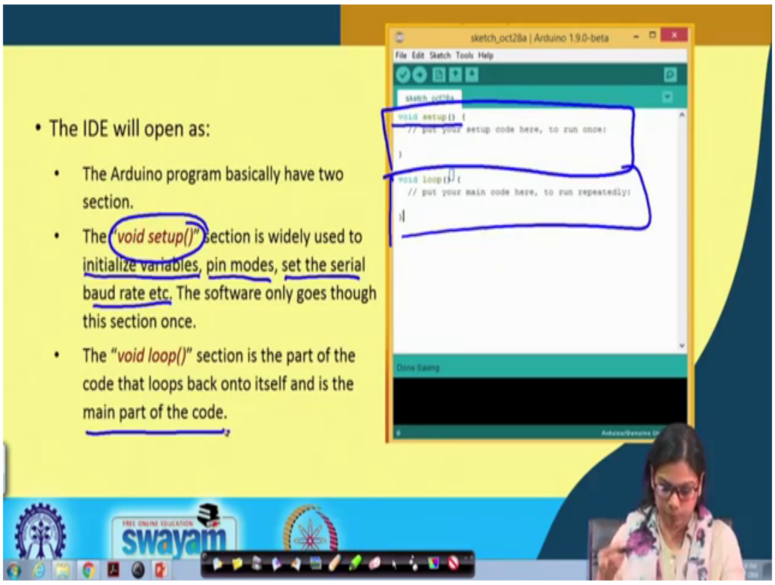


- Extract the downloaded ZIP file and click on *arduino.exe* then the arduino IDE will open.

ARDUINO
BETA
AN OPEN PROJECT WRITTEN, DEBUGGED, AND SUPPORTED BY ARDUINO.CC AND THE ARDUINO COMMUNITY WORLDWIDE
LEARN MORE ABOUT THE CONTRIBUTORS OF ARDUINO.CC on arduino.cc/credits
Starting...
swayam

Once you downloaded, extract the downloaded zip and click on this arduino.exe, then the arduino IDE will get open which looks similar to this, so it is say starting here ok. So, it will look somewhat like this ok.

(Refer Slide Time: 08: 48)



- The IDE will open as:
  - The Arduino program basically have two section.
  - The "*void setup()*" section is widely used to initialize variables, pin modes, set the serial baud rate etc. The software only goes though this 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.

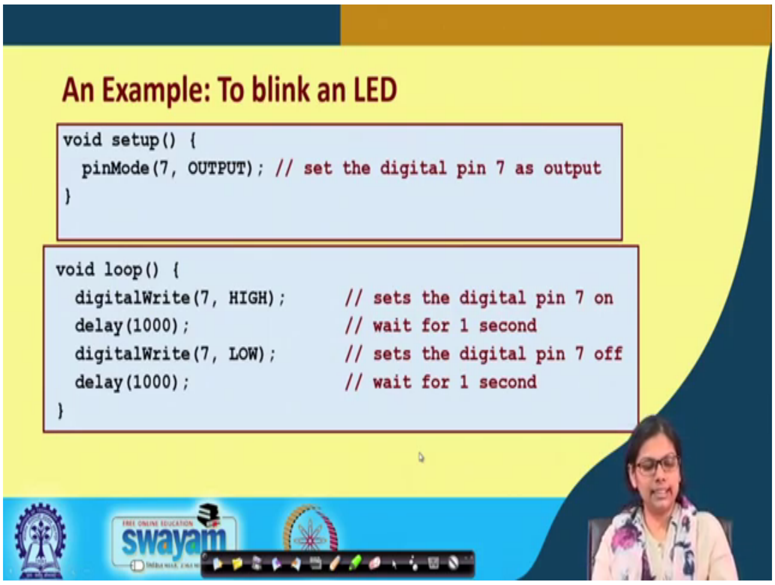
sketch_oct28a | Arduino 1.9.0-beta
File Edit Sketch Tools Help
sketch_oct28a
Done Saving

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

When you have already done with installing the IDE which is fairly very straightforward, then when you open it the IDE will open as like this, it call it a sketch. This is setup phase and this is the loop. What is the setup phase, the setup code here that we write is only run once ok. So, the setup section is widely used to initialize the variables, mention about the pin modes and set the serial baud rate etcetera, when we can use other communicating devices like the GSM or Bluetooth, you need to specify that rate here ok.

So, that must be specified, all those things must go in this void setup phase. The software only goes through this once and only once. The void loop section is the part of the code that loops back onto itself and it is the main part of the code. The code that you write, let us say that, LED will glow for some time. So, that part of the code will be put up in this void loop phase. So, this you must remember setup phase and the loop phase.

(Refer Slide Time: 10:26)



An Example: To blink an LED

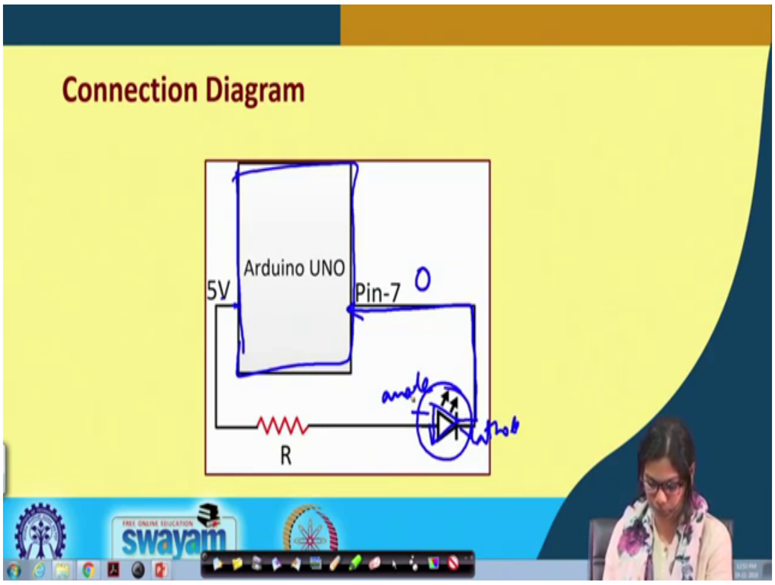
```
void setup() {
  pinMode(7, OUTPUT); // set the digital pin 7 as output
}
```

```
void loop() {
  digitalWrite(7, HIGH);      // sets the digital pin 7 on
  delay(1000);                // wait for 1 second
  digitalWrite(7, LOW);       // sets the digital pin 7 off
  delay(1000);                // wait for 1 second
}
```

Now, I will take a small example to blink an LED what we need to do? You have seen it in embed compiler we need to include a header file etcetera etcetera, we will see what is required here. But the main phases that is required here are 2; one is the set of phase. Here I am doing initializing this is basically the initialization, we are making pin mode, the 7 pin of Arduino as output. Set the digital pin 7 as output. This is what we are doing using this particular statement, pin mode 7 comma out, but you can also do it as input pin mode seven comma input ok, but this here we are specifically making it output because we want the LED to glow.

Then the void loop, what we are doing? We are using a function called digital right, where the output port that is 7 we are making it high. So, we are setting the digital pin 7 on high mode, we are making it on basically. Then we are giving a delay of one second, here it is specified in millisecond. So, 1000 milliseconds is 1 second. Then we do a digital write to the same pin 7 low and then we delay it for again the same 1000 milliseconds that is 1 second and this goes on this is a loop face, so this will go on running ok.

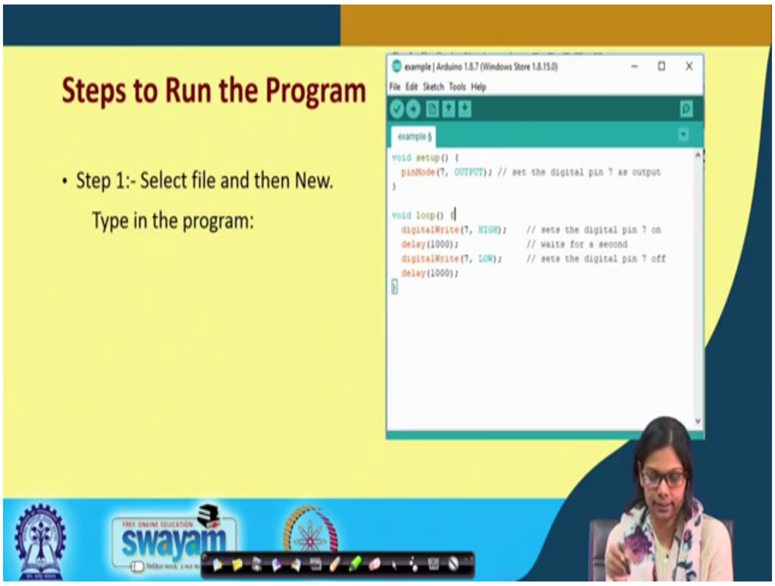
(Refer Slide Time: 12:27)



Connection Diagram

So, this is a typical pin diagram. So, we are not using the onboard LED here; rather, we are using this is the Arduino UNO board and this is the 5 volt of that Arduino and this is the LED ok, this is the anode and this is the cathode. So, the anode end through our resistance is connected to 5 volt, we can see that and the cathode end is directly connected to pin 7 ok. This is the circuit diagram, when the LED will glow as this is connected to 5 volt, when this pin will have a 0 input then only this LED really glow ok, this particular LED will glow.

(Refer Slide Time: 13:27)



Steps to Run the Program

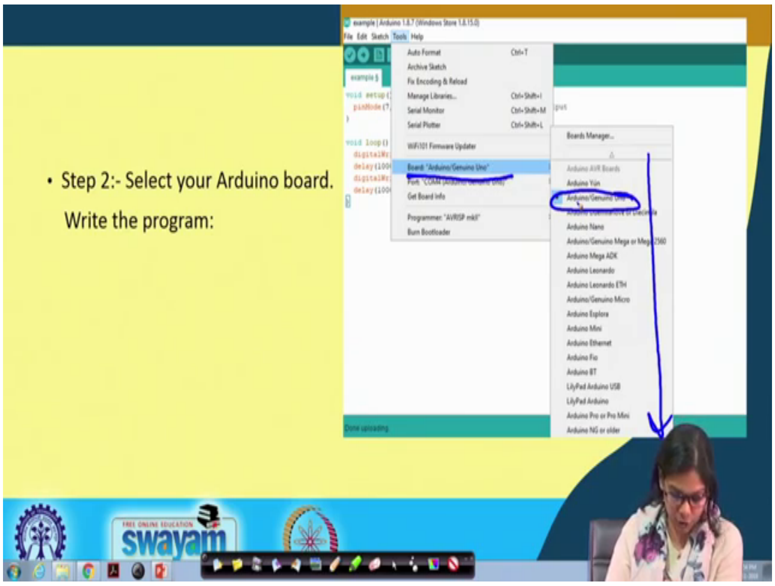
- Step 1:- Select file and then New.
  Type in the program:

example | Arduino 1.8.7 (Windows Store 1.8.15.0)
File Edit Sketch Tools Help
example §

```
void setup() {
  pinMode(7, OUTPUT); // set the digital pin 7 as output
}

void loop() {
  digitalWrite(7, HIGH);    // sets the digital pin 7 on
  delay(1000);              // waits for a second
  digitalWrite(7, LOW);     // sets the digital pin 7 off
  delay(1000);
}
```

So, what are the steps that are required here to do the needful? So, here we select this spin mode 7 as output, we write the program and then, we select the Arduino board ok, there are many Arduino board, but here we are using Arduino UNO.

(Refer Slide Time: 13:47)

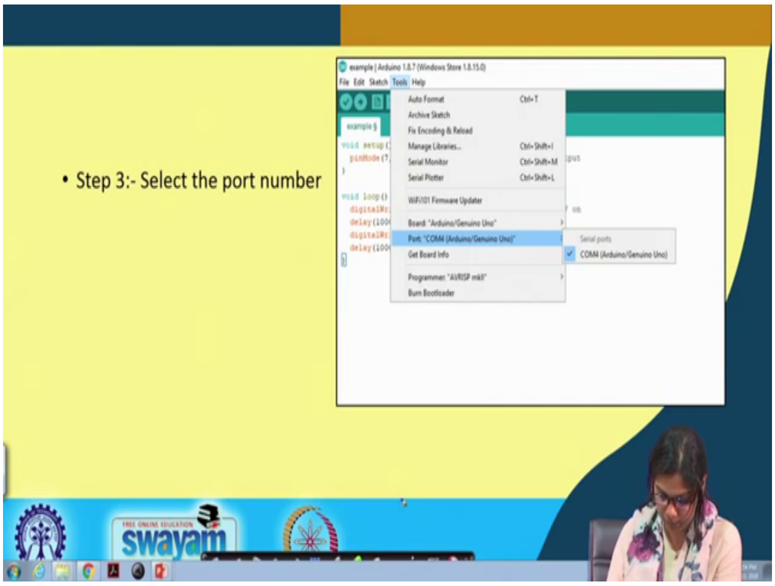


- Step 2:- Select your Arduino board.

  Write the program:

So, specifically you have to use that board from this particular list ok. You see, we have selected this particular board that is Arduino UNO ok.

(Refer Slide Time: 14:09)

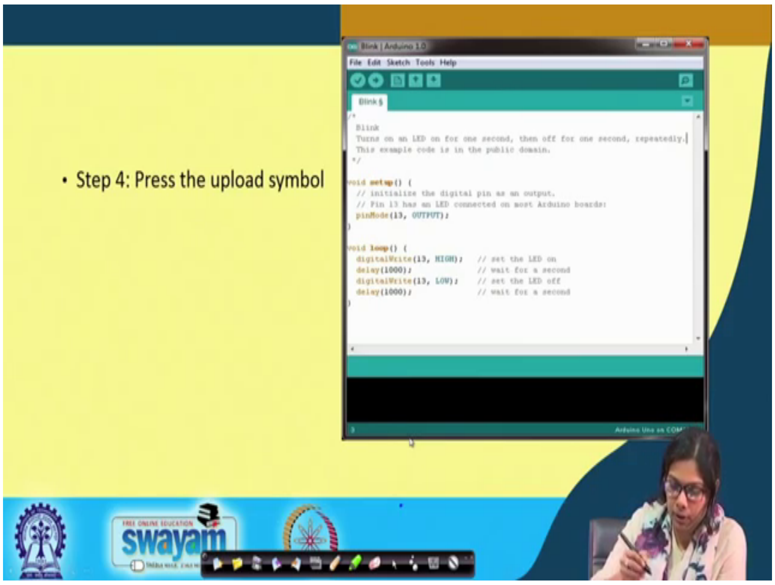


- Step 3:- Select the port number

example | Arduino 1.8.7 (Windows Store 1.8.15.0)
File Edit Sketch Tools Help
example §
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM4 (Arduino/Genuino Uno)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
✓ COM4 (Arduino/Genuino Uno)
swayam

Next select the port number; we can select this COMM for this particular UNO.

(Refer Slide Time: 14:25)

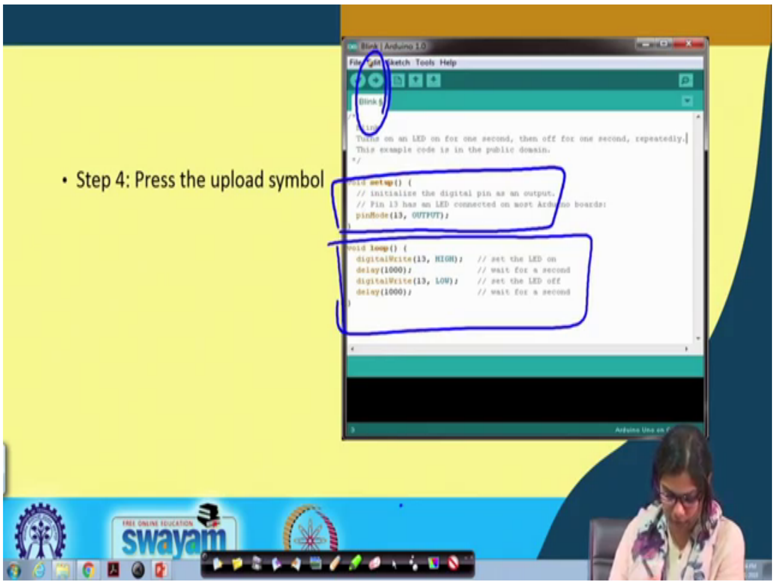


- Step 4: Press the upload symbol

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.
  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

And then we press this upload button to upload it basically, just a second.

(Refer Slide Time: 14:34)

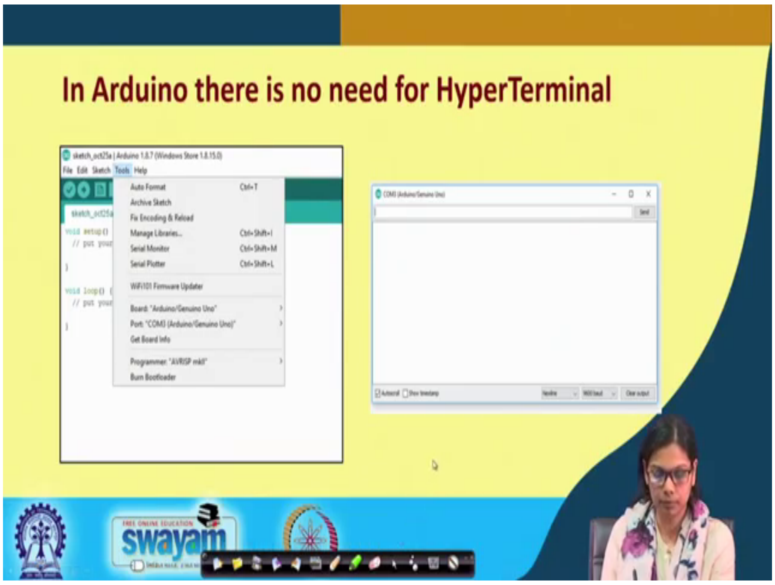


- Step 4: Press the upload symbol

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.
  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

So, this is the setup phase, this is the loop phase and we press on the upload button to upload the programming to it.

(Refer Slide Time: 14:54)



In Arduino there is no need for HyperTerminal

sketch_oct25a | Arduino 1.8.7 (Windows Store 1.8.15.0)
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM3 (Arduino/Genuino Uno)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
sketch_oct25a
COM3 (Arduino/Genuino Uno)
Send
Autoscroll
Show timestamp
Newline
9600 baud
Clear output
swayam

```
void setup() {
  // put your
}

void loop() {
  // put your
}
```

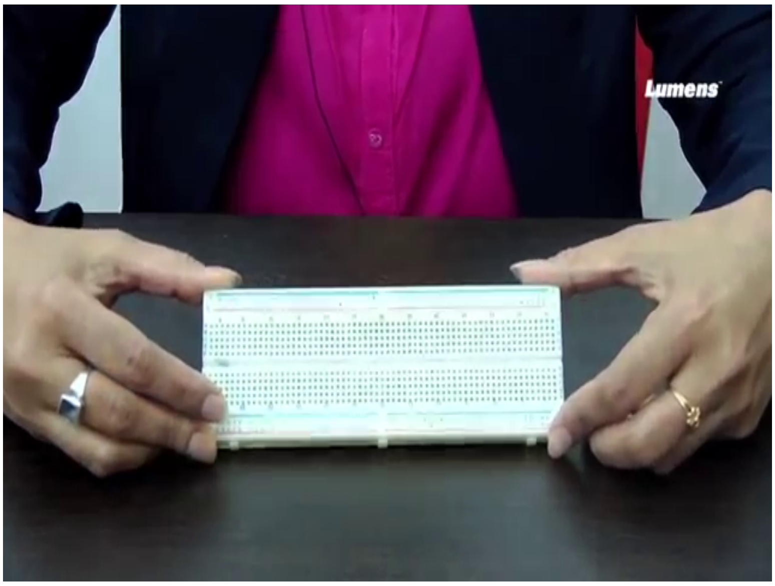
One important thing we have not talked about hyper terminal which we will be discussing in the next week, Arduino has a good feature that, it does not require any external hyper time terminal, the hyper terminal is inbuilt there. So, you go to tools and then you can click on this port COMM 3 Arduino slash gen you know, you know and you will get this. So, whatever data communication you want to see you can see it in this port.

So, we will look into that later, this is one of the feature of Arduino. So, this is all about this Arduino board, how will you interface with Arduino board, you need that ID, you need to write the code, the code goes in 2 phases and finally, you have to upload the code, it gets uploaded into the device and then it is done. So, this is another board aspect apart from STM code ok.

So, today I will be showing you now how do we connect an LED with the 2 kinds of board that I have already discussed. So, what I have discussed, so far I have discussed with you that how we will be working along with STM board and how I will be working along with Arduino board ok.

So, both the boards has got some features, which we have already seen. Now we will be right away going to the interfacing experiments, that we will be doing in this session. I have already discussed how the onboard LED will glow, great! we have already seen that. Now I will be connecting another LED using bread board with STM board, as well as with Arduino board. So, let us see how we can do this ok.

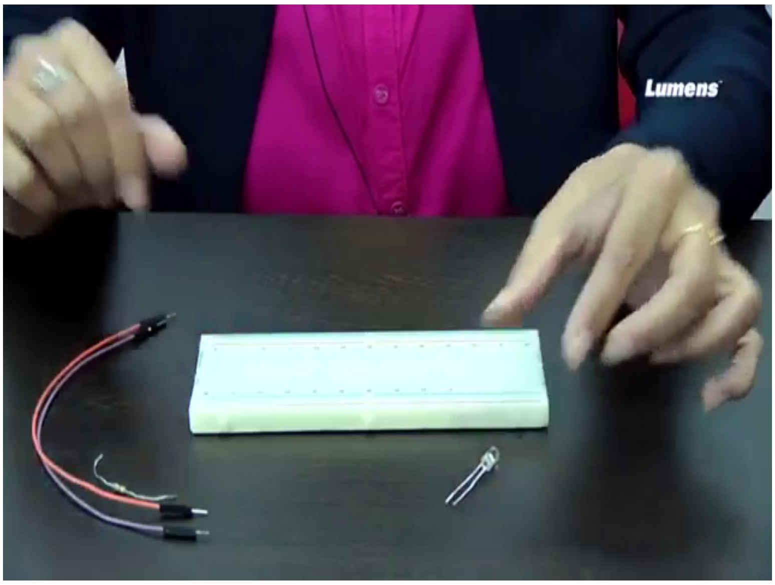
(Refer Slide Time: 17:25)



So, on the first go, I will first talk about breadboard, this is a breadboard through which we can connect various small, small interfacing devices and we can do small experiments. So, if you see how the connections are here. These lines are horizontally connected. So, if you see a breadboard, these are the lines which are horizontally connected, these are connected, this is connected all together, this is connected all together, but there is no connection between this and this, but there is a connection between these few points and these are vertically connected ok. So, there are 2 lines we can see here, these are vertically connected together.

So, from any point you can put either V CC or ground and then you can use and connect with other pins maybe which is connected here or here anywhere. Again one thing you have to remember there is a cut between here. So, this is connected, but this is not connected with this one. So, this is a separate connection, this is a separate connection. If you want to connect the whole thing you have to make aware from here till here ok. So, this is how the connection in the breadboard is. So, we will be doing the first experiment that is with LED, using this breadboard.

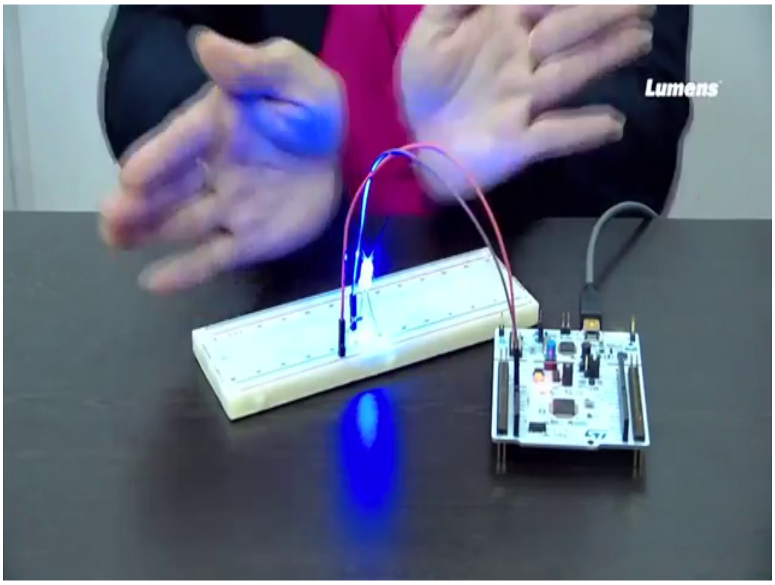
(Refer Slide Time: 19:05)



We also require some jumper wires, these are some jumper wires which is required and a resistor. Firstly, I will show you this is the LED, see this LED. If you see this LED, you can make out that one of its leg is longer, another is shortened. I already discussed this in the lecture, but what does it mean, this is the anode and this is the cathode. So, the long leg is a anode and the short leg is cathode and current always flows from anode to cathode ok.

The first thing that I will be do that, I will be doing here is that, I will connect this LED with STM board. And the first thing again I will do is first I will check whether this LED is working fine or not? How will you check that whether this LED is working fine or not? What you can do, the anode with a resistance you can connect to BCC and the cathode you can directly connect to ground and then we will see whether this LED glows or not ok.

(Refer Slide Time: 20:34)



So, let us do this experiment with STM board first. So, this is the STM board. So, this one is my anode, this one is cathode. So, I put it up like this, put it up in the breadboard like this. Then through a resistance, I connect to this end of the breadboard and from this end which is vertically connected, I will put it to BCC. So, the BCC pin here is the fourth pin from here 1, 2, 3 and 4, so I put it there. And this from this point, from this point, that is the anode, I will connect it to around and I see that the LED is glowing ok, you can see that the LED is glowing. If I take it out, it is not glowing. Now again I will put it up here, it is again glowing ok.

So, if you can understand that now you can make out that yes this LED is a working LED. Now what I will do is that, I will connect the anode with BCC, but now I will connect the ground with digital pin D2 ok. So, the digital pin D2 is here, this is D0, D1 and D2, the third pin from here, I connect it. And now what I will be doing? I will be dumping a code into this boot ok. So, for dumping the code, we have already seen what all are the steps that needs to be followed to dump the code into this particular boot so that I will be doing ok.

Now can you see one thing, what is happening? The LED is glowing for 1 second and it is off for 2 second, you see it is glowing for 1 second and it is off for 1 second and this is repeating. This is in that while loop ok. So, this is in that infinite while loop, where we have actually put up the code, where the LED was glowing for 1 second and it is off for 2. Now it is glowing for 1 second, again off for 2 second.

So, this is how you can do programming with this STM board using a single LED. You can also make different various kinds of experiment with LED, with different widths few more LED's if you connect, you can make a counter that way. So, you can put another LED here and what will be the counter? The counter will be first 0 0 then 0 1 then 1 0 and then 1 1. So, this can also repeat forever.

So, there are various kind of experiments that you can try out doing with STM board. Now what I will be doing? Now I will be doing the same experiment with Arduino board. The code I have already discussed. Now I will be doing the connection first with the Arduino board, this is Arduino UNO board. The functionalities have already discussed. Now I will connect it. The same way this connection is straightforward. Now I will connect this part with BCC and this one with pin D2 ok. Now I will again use the Arduino editor, that we have used for dumping the code ok.

So, now the code is dumped and it is also doing the same thing, it is on for one second and it is off for 2 second. The same code we have written using the Arduino board and it is doing the same thing ok. So, you see that we have done 2 experiments with same experiment basically with 2 different boards and please make sure to understand the connection. You see what connection we have made it here. These are the 2 legs, these are the 2 legs of the LED. 1 leg that is the anode is connected using this resistor, using this resistor, through the V CC it is connected this through this resistor we have connected and then from here we have put it to BCC and the cathode is directly connected to one of the digital pin that is D2 ok.

And what we are doing? We are passing through this digital pin D2 0. When we pass 0, this LED is glowing. When we are passing 1, this LED will not glow ok. So, this is how this works, this LED works ok. So, in this experiment we have connected a single LED to both the boards that is STM and with Arduino UNO. You can try out other examples connecting a few more number of LED's and then you can make a LED counter in some way or the other or you can try out with various other examples. We will really, we will be using this LED for some other things as well in some more experiments. So, we will be seeing that where we can use this particular feature of this LED.

Thank you.