Unit 1: Controllers

Schools of Specialized Excellence
Delhi Board of School Education

Session Plan

Controllers

Teacher Name		**Target Grade**	**10**	**Curriculum Component**	**Applied Learning Module**

Module Title	**Introduction to embedded system**		
Week Title	**Controllers**	**Week Number**	**1**
Important Concepts	**- Explain Embedded System** **- Microcontroller vs. Microprocessor** **- Different types of microcontrollers** **- Link b/w Microcontroller and Arduino**		

Learning Standards

1. Learning embedded systems as an integral part of Electromechanical Automation
2. Listing out the Automation Projects using Embedded System
3. Fundamental knowledge of Microcontroller& Microprocessor
4. Explain the Arduino and its Relationship with the Microcontroller

Inquiry Questions

1. **What do you know about Embedded Systems?**
2. **Give me one example of Automation using Embedded System in our Day-to-day Life?**
3. **What do you know about Arduino?**

Classroom Inquiry Process	
Day 1: Introduction to the embedded system.	**Lesson Aims** **1. Learning about Embedded System** **2. Listing out the difference between Microcontroller & Microprocessor** **3. Classify the types of microcontrollers** **4. Relate the Similarities b/w microcontroller and Arduino** **Activity Title:** **1. Ice-breaking on Embedded System (10 Mins)**

2. Introduction to Embedded Systems (20 Mins)

3. Difference between Microcontroller & Microprocessor (30 Mins)

4. Different types of microcontrollers (30 Mins)

5. Link b/w microcontroller and Arduino (10 Mins)

6. Doubt Clarification / Q & A Session (10 Mins)

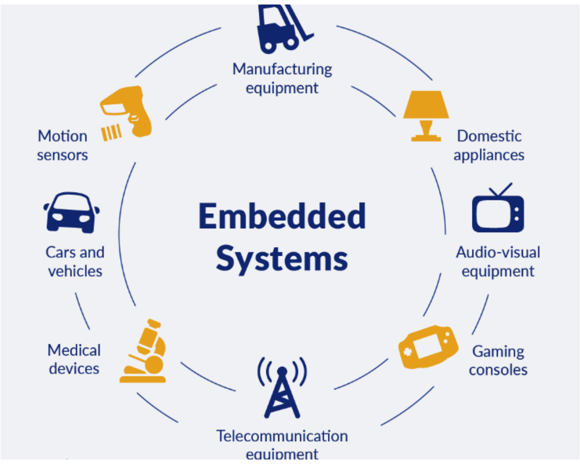
7. Instructions for taking Home Assignment (10 Mins) (Individual)

Activity Description:

1. **Ice-breaking on Embedded System:**
   At the beginning, Show them Real-Life Applications like Smart TVs, Smart Phones, Smart Remote, Automatic Washing Machines & Role of Embedded Systems in each Application. Provide a brief description of the Embedded systems & Arduino. Introduce the subject area. Explain the kind of projects that can build using Arduino.

2. **Introduction to Embedded systems:** Brief introduction on the embedded systems using an example.

   What is meant by an Embedded system?
   Embedded System is a combination of computer software and hardware which is either fixed in capability or programmable. Exercise of identifying the systems having embedded systems around us.



EXAMPLES:

Role of Embedded System in Automatic Washing Machine:

Working, Advantages & Components like Status display panel, Switches, Dials, Motor, Power supply & control unit, Inner water level sensor and solenoid valve.

Features:

1. Wash by spinning (Normal Wash/Quick Wash)
2. Rinse
3. Drying
4. Wash over by blinking / Indication
5. Each Sequence displays the process stage to User
6. In the case of Power Interruption, How the Process Resumes?

Embedded System Consists of

- Components like Sensors, Processor, Memory, Actuators
- Classifications based on Functions & Performance Requirements
- Processors
- Other Hardware
- Software for Programming, Testing
- Application like Domestic, Commercial & Industrial Electronic Projects

Activity on identifying the systems having embedded systems around us.

3. **Difference between Microcontroller & Microprocessor:**

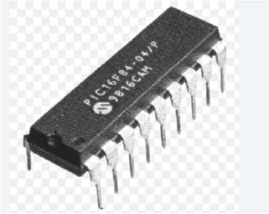
Introduce the Microcontroller and Microprocessor using live Electronic Appliances & Explain the Differences based on Features, Components, Working & applications.

Microprocessor	**Microcontroller**
Heart of the system.	Heart of the embedded system.
Externally connected with input-output components.	input-output components are embedded.
The circuit may be large depending upon usage.	The circuit is very small.
Not cost-effective.	Cost-effective.
The total consumption of power is high.	Total consumption of power is less.

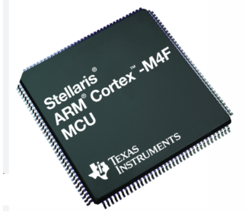
Power saving mode is not generally available.	Power saving mode is generally offered.

4. **Different types of microcontrollers:**

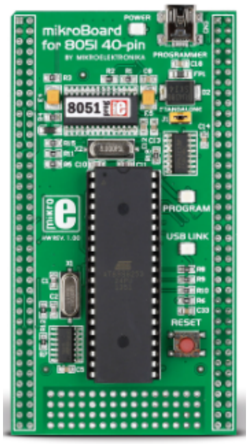
PIC- PIC Stands for Peripheral Interface Controller is a kind of microcontroller component used in the development of electronics, computer robotics, and similar devices.



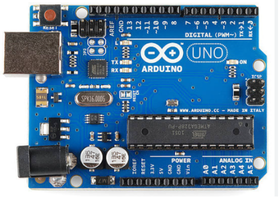
ARM- ARM stands for Advanced RISC Machine. It's the most popular Microcontroller Programming in the digital embedded system world, It is cost sensitive and high-performance device.



8051- Intel created 8051 microcontrollers in 1981. It is an 8-bit microcontroller. It's made with 40 pins DIP (Dual inline package), 4kb of ROM storage and 128 bytes of RAM storage, and 2 - 16-bit timers.



AVR- AVR stands for Alf and Vegard's RISC Processor. It was the modified Harvard architecture machine, where program and data were stored in a separate physical memory system.



MSP- MSP stands for Mixed Signal Processor. It’s the family from Texas Instruments. Built around a 16 -bit CPU, the MSP is designed for low-cost and respectively, low-power dissipation embedded statements.

5. **Link b/w Microcontroller and Arduino:**

Features:

Memory-The 8051 microcontroller has a total of 128 bytes of RAM. The Arduino Uno has only 32K bytes of flash memory and 2K bytes of SRAM. It also uses another form of memory, the EEPROM to store long-term information but is slower than SRAM.

Power Supply-The 8-bit Intel 8051 operates at a voltage between +5 volts to a maximum of 6.6 volts. The Arduino can accept between 6V and 20V (7-12V recommended) via the direct current barrel jack connector or into the Vin pin. It has built-in 5V and 3.3V regulators;

Application-

The 8051 was initially developed for use in embedded systems but later expanded to industrial control applications. They are frequently used in automatically controlled products, such as automobile engine control systems, remote controls, appliances, power tools, office machinery, and toys.

Arduino can read sensor data and control components such as lights, motors, thermostats, and garage doors. Arduino projects are frequently used for building IoT (Internet of Things) applications. It is also used for home automation systems and a wide range of innovative applications from robotics and lighting to gardening and games.

6. Doubt Clarification /Q & A Session- Do the Learning check using these Prompt Questions:

- Give one example of an Embedded System.
- What is the difference between Microcontroller & Microprocessor?
- List out the type of Microcontrollers?
- Why is Arduino called an Open-Source Platform?

7. Take Home Assignment

Make a list of different microcontrollers & Arduino Boards used in Our Day-to-day Life Electronic Appliances in your Handouts.

References

YouTube video https://youtu.be/Vt7kXpAaqGo

Day: 2 **Introduction to ICs**	**Lesson Aims** **1. Describe Embedded Systems & ICs** **2. Illustrate the Different types of IC’s**

3. Demonstrating the Real-life applications of ICs

Activity Title:

1. **Ice-breaking on Integrated Circuits (10 Mins)**
2. **Introduction to Integrated Circuits (20 Mins)**
3. **Digital Logic Gates (25 Mins)**
4. **Timer Circuit using 555 IC(25 Mins)**
5. **Voltage Regulator(20 Mins)**
6. **Doubt Clarification / Q & A Session (10 Mins)**
7. **Instructions for taking Home Assignment (10 Mins) (Individual)**

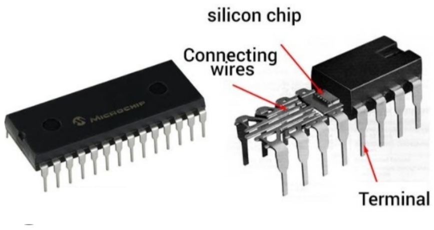
Activity Description:

1. **Ice-breaking on Integrated Circuits:** At the beginning, show them Real-Life Applications like Smart Watch, Smart Phone, Computers, and USB Drivesthat work on **Integrated Circuits**. Explain the kind of projects like Motion Detector, Infrared Remote Controls, and Voice Operated Remote that can be built using Integrated Circuits.

2. **Introduction to Integrated Circuits:** Brief introduction on the Integrated Circuits using an example. Explain How to identify the IC'sused as Timer, and Regulator.

   What is meant by Integrated Circuits?

   An integrated circuit (IC) is manufactured using silicon material and mounted in a ceramic or plastic container (known as a Chip). The basic components of an IC consist of electronic circuits for the digital gates. The various gates are interconnected inside an IC to form the required circuit.

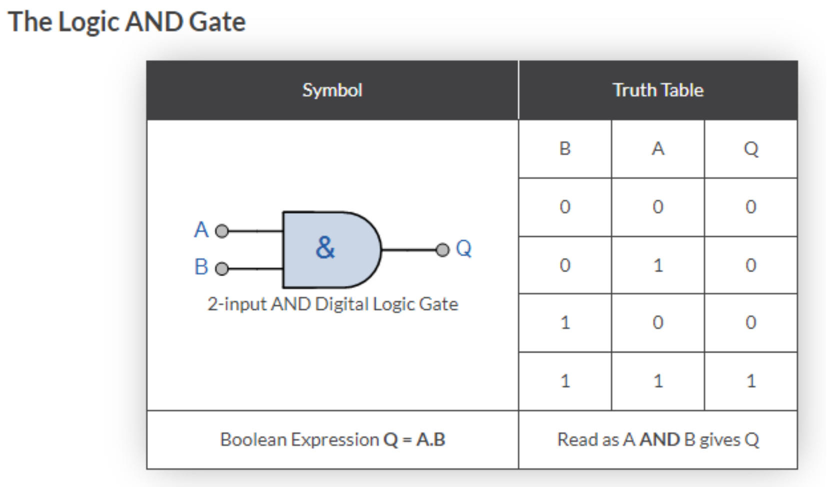


3. **Digital Logic Gates**:
   Digital electronic circuits operate with voltages of **two logic levels: Logic** Low and Logic High. The range of voltages corresponding to Logic Low is represented with '0'. Similarly, the range of voltages corresponding to Logic High is represented with '1'.
   The basic digital electronic circuit with one or more inputs and single output is called the **Logic gate**.

We can classify these Logic gates into the following three categories.

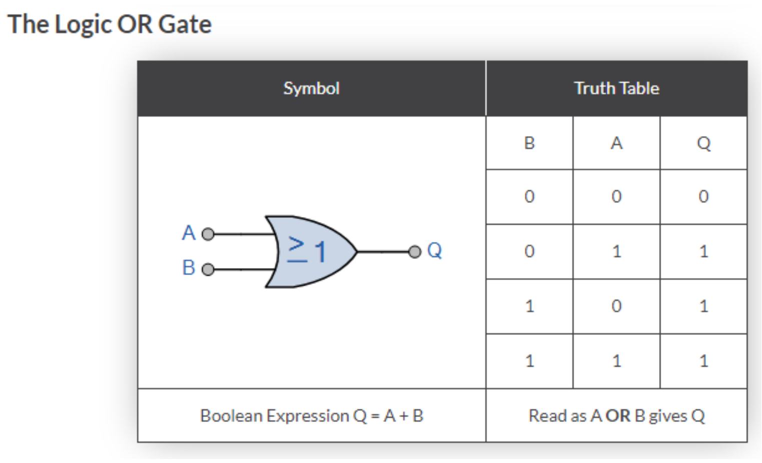
- Basic gates
- Universal gates
- Exclusive gates

Basic Gates
The basic gates are AND, OR & NOT gates.



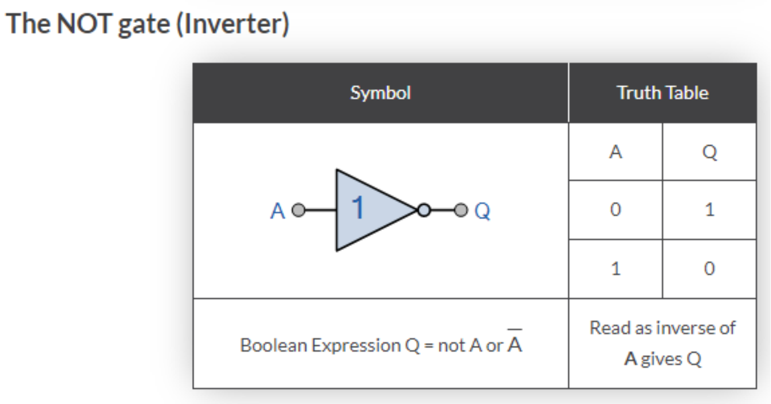
The Logic AND Gate

Symbol	Truth Table		
	B	A	Q
2-input AND Digital Logic Gate	0	0	0
	0	1	0
	1	0	0
	1	1	1
Boolean Expression **Q = A.B**	Read as A **AND** B gives Q		



The Logic OR Gate

Symbol	Truth Table		
	B	A	Q
	0	0	0
	0	1	1
	1	0	1
	1	1	1
Boolean Expression Q = A + B	Read as A **OR** B gives Q		



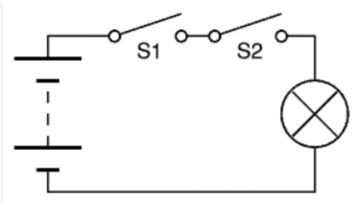
The NOT gate (Inverter)

Symbol	Truth Table	
	A	Q
	0	1
	1	0
Boolean Expression Q = not A or $\overline{A}$	Read as inverse of A gives Q	

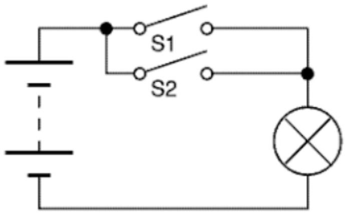
Activity:

Explain the Basic Gates Truth table by performing the Below Activity:

- Take 2 (one-way-switches) and 1 LED. Connect in Series & Verify the outcome with the Truth table of AND Function.

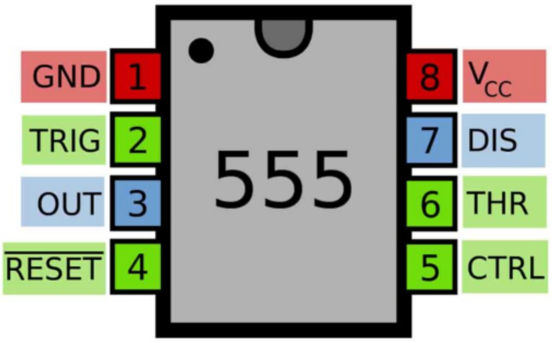


- Take 2 (one-way switches) and 1 LED. Connect in parallel and Verify the outcome with the Truth table of the OR Function.



4. **555 Timer IC:**

The 555 Timer IC got its name from the three 5KΩ resistors that are used in its voltage divider network. This IC is useful for generating accurate time delays and oscillations. This chapter explains about 555 Timer in detail.



555 Timer IC Pinout

Applications:

- For automatically turning Off mobile chargers to prevent overcharging the battery
- To turn Off reading lights automatically after the set duration
- To control the sequence of output devices one after the other after regular/irregular time periods

5. **Voltage Regulator:**

A voltage regulator is an **integrated circuit** (IC) that provides a constant fixed output voltage regardless of load or input voltage change.

	 Real-Life Applications: The battery in your car that gets charged from the alternator, the outlet in your home that provides all the electricity you desire, and the **cell phone** you likely keep on hand every minute of the day; require a specific voltage to function. Fluctuating outputs that jump from ±2V can cause inefficient operation and possibly even damage to your charging devices. **Watch this Video on Voltage Regulator Applications:** https://www.youtube.com/watch?v=h66qYRJk4Ac 6. **Doubt Clarification /Q & A Session- Do the Learning check using these Prompt Questions:** - Why is it called IC 555 timer? - List out the Countries manufacturing ICs? - List out the Application of Logic Gates in real-time? 7. **Take-Home Assignment** List down the real-life examples of Logic Gates that are used in our day-to-day life in your handouts. **References** YouTube video https://youtu.be/UWPxa6N7VvA https://www.youtube.com/watch?v=h66qYRJk4Ac
Day 3: Getting Started with Arduino	**Lesson Aims** 1. Explain the basics of Arduino 2. Describe the Pin configuration and Architecture of Arduino 3. Familiarise with Arduino Interfacing Board 4. Explain about Embedded C and Arduino platform

Activity Title:

1. **Ice-breaking on Embedded System (10 Mins)**
2. **Introduction to Arduino (15 Mins)**
3. **Pin Configuration & Architecture (30 Mins)**
4. **Familiarising with Arduino Interfacing Board(20 Mins) (demo)**
5. **Introduction to Embedded C and Arduino platform(25 Mins)**
6. **Doubt Clarification / Q & A Session (10 Mins)**
7. **Instructions for taking Home Assignment (10 Mins) (Individual)**

Activity Description:

1. **Ice-breaking on Embedded System:** Recap on previous learning on Embedded Systems, Microcontrollers & Microprocessor. Importance of programming in Arduino & Explain the kind of projects or Applications like ATM card cum Reader, Cameras, Fun City Play Card, Calculator, Digital Wristwatch can build using Arduino.

2. **Introduction to Arduino:** Brief introduction on the Arduino & Show them any Prototype of Electronic Projects like Smart Dustbin, Home Automation, Remote Control Light Switch having different types of Arduino boards**.**

Smart Dustbin

Remote Control Light Switch

Alarm Clock

Smartwatch

What is Arduino?

Arduino is an **open-source platform** based around programmable development boards that can be integrated into various simple and complex projects.

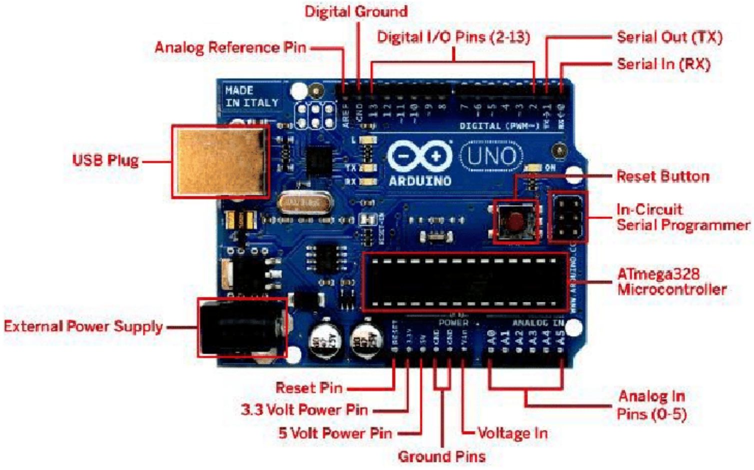
The term **open source** refers to something people can modify and share because its design is publicly accessible. The term originated in the context of software development to designate a specific approach to creating computer programs.

The Arduino family consists of different types of development boards, with the most common being the Arduino UNO.

An Arduino board contains a microcontroller which can be programmed to sense and control devices in the physical world. The microcontroller can interact with various components such as LEDs, motors and displays. Because of its flexibility and sustainability, Arduino has become a popular prototyping development board widely used worldwide.

3. **Pin Configuration & Architecture**:

Here are the components that make up an Arduino UNO R3 board and what each of their functions is:



1. Reset Button – This will restart any code that is loaded to the Arduino board

2. AREF – Stands for "Analog Reference" and is used to set an external reference voltage

3. Ground Pin – There are a few ground pins on the Arduino, and they all work the same

4. Digital Input / Output – The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as digital input pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labelled "~" can be used to generate PWM.

5. PWM – The pulse width modulation pins marked with the (~) symbol can simulate analogue output

6. USB Connection – Used for powering up your Arduino and uploading sketches

7. TX/RX – LEDs to visualise data being transmitted and received from the board

8. ATmega328p – This is the microcontroller which stores the program and processes it

9. Power LED Indicator – This LED indicates the board is connected to a power source

10. Voltage Regulator – This controls the amount of voltage going into the Arduino board

11. DC Power Barrel Jack – This is used for powering your Arduino with a power supply

12. 3.3V Pin – This pin supplies 3.3 volts of power to your projects

13. 5V Pin – This pin supplies 5 volts of power to your project

14. Ground Pins – There are a few ground pins on the Arduino, and they all work the same

15. Analog Pins – The Arduino UNO board has six analogue input pins A0 through A5. These pins can read the signal from an analogue sensor like the humidity sensor or temperature sensor and convert it into a digital value that the microprocessor can read.

4. **Familiarizing with Arduino Interfacing Board (Demo):** Present a demo bot (live project) using an Arduino board, use digital i/o pins and analogue pins, and attach sensors and actuators to demonstrate Input and output.

5. **Introduction to Embedded C and Arduino platform:**

Introduction to Embedded C:

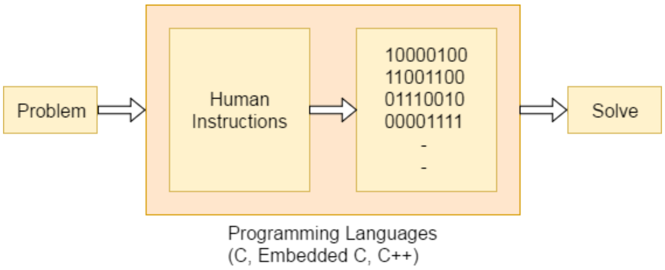
Embedded C is the most popular programming language in the software field for developing electronic gadgets. Each processor used in electronic systems is associated with embedded software.

Embedded C programming plays a key role in performing specific function by the processor. We use many electronic devices such as mobile phones, washing machines, digital cameras, etc. This all-device working is based on microcontrollers that are programmed by embedded C.

In an embedded system, programming C code is preferred over other languages. Due to the following reasons:

- Easy to understand
- High Reliability
- Portability
- Scalability

Let's see the block diagram representation of embedded system programming:

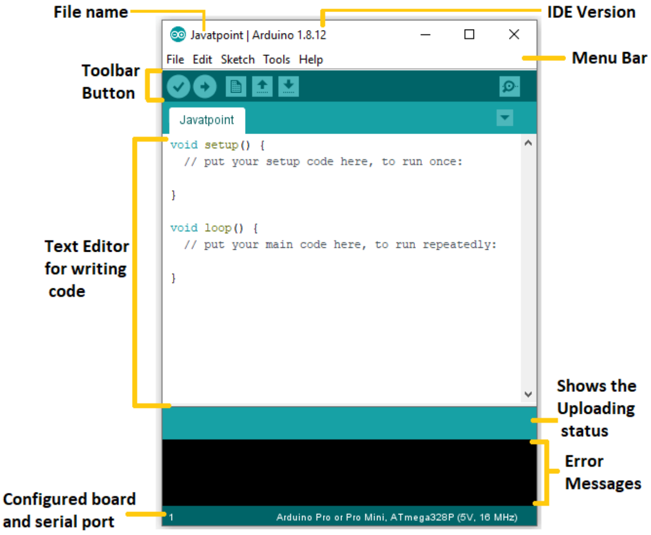


Function is a collection of statements that is used for performing a specific task and a collection of one or more functions is called a programming language. Every language consists of basic elements and grammatical rules. The C language programming is designed for functions with variables, character sets, data types, keywords, expressions, and so on that are used for writing a C program.

Arduino platform:
The Arduino IDE is open-source software which is used to write and upload code to Arduino boards. The IDE application is suitable for different

operating systems, such as **Windows, Mac OS X, and Linux**. It supports the programming languages C and C++. Here, IDE stands for **Integrated Development Environment**.

The program or code written in the Arduino IDE is often called sketching. We need to connect the Genuino and Arduino board with the IDE to upload the sketch written in the Arduino IDE software. The sketch is saved with the extension '.ino.'
The Arduino IDE will appear as **(Explain with live demo).**



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

}

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

}
```

File name
IDE Version
Javatpoint | Arduino 1.8.12
File Edit Sketch Tools Help
Menu Bar
Toolbar Button
Javatpoint
Text Editor for writing code
Shows the Uploading status
Error Messages
Configured board and serial port
1
Arduino Pro or Pro Mini, ATmega328P (5V, 16 MHz)

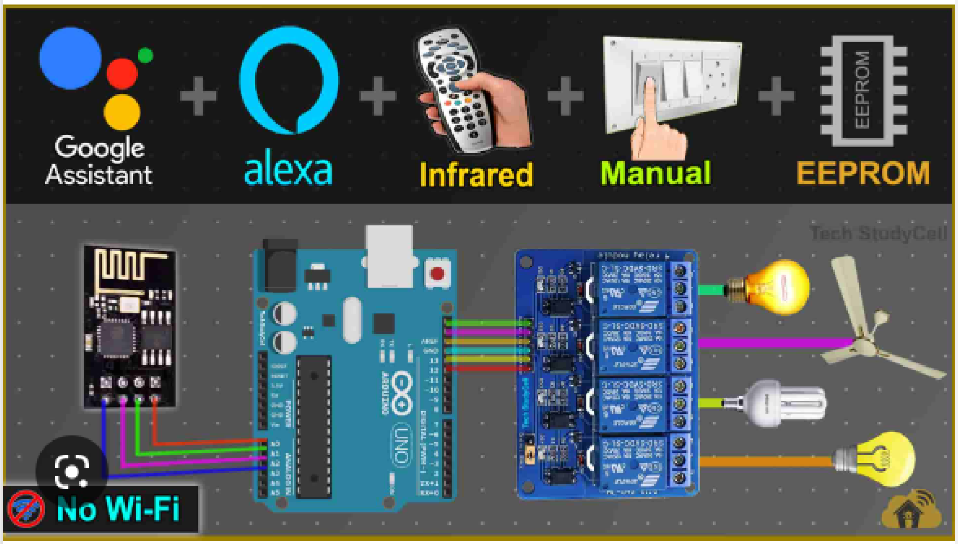
Use this link to Download Programming Software:
https://www.arduino.cc/en/software

6. Doubt Clarification /Q & A Session- Do the Learning check using these Prompt Questions:

- What is Arduino?
- Ask Questions based on describing the Selected Pin Configuration & Architecture of the Arduino Uno board.
- Where do we use Embedded C language?
- What is the full form of IDE?

7. Take Home Assignment

Explore all the features of Arduino board &IDE. / Design(own idea) project or bot using Arduino

	References - YouTube video: https://youtu.be/9cxAjRHdMVY - Watch this video to learn Arduino Projects for beginners: 15 Great Arduino Projects for beginners
Day 4: Arduino i/o Functions.	**Lesson Aims** 1. Explain the Arduino data types 2. Demonstrate the Program based on Variables and constants 3. Describe the Operators& execute any applications of Operators 4. Explain the Control Statements used in Arduino Programming with Applications. 5. Explain the Arrays & Illustrate with Sample Program. 6. Explain the Functions & Illustrate with Sample Program. **Activity Title:** 1. **Ice-breaking on Arduino& Programming (10 Mins)** 2. **Arduino data types(30 Mins)** 3. **Variables and constants(30 Mins)** 4. **Operators(30 Mins)** 5. **Doubt Clarification /Q & A Session- Do the Learning check using these Prompt Questions: (10 Mins)** 6. **Taking Home Assignment (10 Mins) (Individual)** **Activity Description:** 1. **Ice-breaking on Arduino & Programming:** At the beginning, show them the IoT Applications of Arduino like Google Assistant, Alexa, and Burglar Alarm& provide a brief description of the role of Programming Arduino.  Google Assistant alexa Infrared Manual EEPROM Tech StudyCell No Wi-Fi

2. **Arduino data types:** The data type is used for storing memory, which is used to define a variable its use.

   Void: It is used to specify the type of function. When a function does not return any value, we specify the return type as void. If there is an empty parameter list for a function, it is specified as void.

   Char: A single character can be defined as a character(char) type data type. Characters are usually 8 bits (one byte) of internal storage. The qualifier is singed or unsinged. While unsigned character values are between 0 to 255, singed characters have from -128 to 127.

   Int: Integers are whole numbers, i.e. numbers without a decimal point. A short int requires half the space of normal integer values, and a long int requires double the amount of space than normal integer values.

   Float: Data type for a floating-point number is a number that has a decimal point. Floating-point numbers often express analogue or continuous values because they have greater resolution than integers.

Watch This Video:
Fun with Data Types: https://www.youtube.com/watch?v=vOnIA9U5qmc

3. **Variables and constants:**

   A constant is a value that doesn't change with the execution of a program. A variable is an identifier that is used to store a value.

   What is a Variable?

   A variable is a data type used to store a data value. A Variable is an entity that changes during program execution. For example, when we write = 45, i.e., x can hold different values at different times, so r is a variable.

   A variable may be an integer, character, float, double, etc.
   Rules for constructing Variable Names:

- A variable name must start with the alphabet or underscore.
- A variable name can be a combination of alphabets, digits, or underscores.
- No commas, special symbols (rather than underscore [_]), or blanks are allowed within a variable name.
- It should not be a keyword.
- White spaces are not allowed.

- Upper case and lower case are significant.

4. Operators:

Its type of symbol that instructs the compiler to perform specific mathematical or logical functions.

Type of Operators:

- Arithmetic Operators
- Comparison Operators
- Boolean Operators
- Bitwise Operators
- Compound Operators

Arithmetic Operators

Show Example

Operator name	Operator simple	Description	Examp
assignment operator	=	Stores the value to the right of the equal sign in the variable to the left	A = B
addition	+	Adds two operands	A + B will give 3
subtraction	-	Subtracts the second operand from the first	A - B will give -1
multiplication	*	Multiply both operands	A * B will give 2
division	/	Divide numerator by denominator	B / A will give 2
modulo	%	Modulus Operator, and the remainder of after an integer division	B % A will give 0

Comparison Operators

Show Example

Operator name	Operator simple	Description	
equal to	==	Checks if the value of two operands is equal or not, then the condition becomes true.	(A == B) is
not equal to	!=	Checks if the value of two operands is equal or not; if values are not equal, then the condition becomes true.	(A != B) is
less than	<	Checks if the value of the left operand is less than the value of the right operand, then the condition becomes true.	(A < B) is t
greater than	>	Checks if the value of the left operand is greater than the value of the right operand; if satisfies, then the condition becomes true.	(A > B) is r
less than or equal to	<=	Checks if the value of the left operand is less than or equal to the value of the right operand, then the condition becomes true.	(A <= B) is
greater than or equal to	>=	Checks if the value of the left operand is greater than or equal to the value of the right operand; if satisfies, then the condition becomes true.	(A >= B) is

Watch This Video: https://www.youtube.com/watch?v=UUx0_s-ElSs

5. Doubt Clarification /Q & A Session- Do the Learning check using these Prompt Questions:

- List out the types of data types
- What is a variable?
- List out the types of Operators

	6. Take Home Assignment: Create a Program (Any one Application) using the arithmetic operator in your Handouts. ● Traffic Control System ● Elevator Control System for 2 Floors Watch This Video: https://www.youtube.com/watch?v=l1FxtpqNMyI : https://www.youtube.com/watch?v=u6ilYC7LeKQ **References** Website:https://www.tutorialspoint.com/arduino/index.htm Watch This Video: https://www.youtube.com/watch?v=YgQfdcc5Aa4
Day 5: Arduino Time	**Lesson Aims** 1. Explain the Arduino time function 2. Demonstrate the Program using the delay() function with Examples. 3. Explain about the delayMicroseconds() function& Illustrate with Examples 4. Describe the millis() function& Illustrate with Examples 5. Describe the micros() function& Illustrate with Examples **Activity Title:** **1. Ice-breaking on Arduino Time Delay Function / Programing (10 Mins)** **2. Types of Arduino time(10 Mins)** **3. Delay()&DelayMicroseconds function(20 Mins) (with demo)** **4. millis()&micros() function (20 Mins)** **5. Activity on Blinking LED with Arduino time commands, traffic light, ambulance light, and sound (30 Mins)** **6. Doubt Clarification / Q & A Session (10 Mins)** **7. Instructions for taking Home Assignment (20 Mins) (Individual)** **Activity Description:** 1. **Ice-breaking on Arduino Time Delay Function:** Recap on previous learning on Arduino Functions. Explain Safety Door systems in Household Appliance like Microwave Oven, Refrigerator, Front Load Washing Machine Having Arduino Primarily working with Time Delay Functions 2. **Types of Arduino time:** There are several ways to Incorporate date and time into the Program. ● Real-Time Clock (RTC) – The type of IC keeps track of the current date and time data. This IC has an In-Built battery source to keep the RTC running

even if the main power to the device gets Off. Example: Smart Phones have built-in RTC devices.

- Global Positioning System (GPS) – A GPS device communicates with satellites to determine its location anywhere in the world. It sends Signals Contains Location Coordinates, Time.
- Time Server –Computer on a network that reads the time from some reference clock and distributes it to the system connected to the network.

3. **Delay() function** accepts a single integer (or number) argument. This Function represents the time which is measured in milliseconds. It works as the program should wait until moving on to the next line of code when it Executes.

Delay Microseconds () function: The **delay Microseconds()** function accepts a single integer (or number) argument. This function represents the time and is measured in microseconds.

4. **millis() function:**

This function is used to return the number of milliseconds at the time of execution When the Arduino board begins running the current program.

micros() function:

The micros() function returns the number of microseconds from the time of Execution When the Arduino board begins running the current program.

Watch This Video:**https://www.youtube.com/watch?v=IyxY1uQyY9U**

5. Activity on Blinking LED with Arduino time commands, traffic light, ambulance light, and sound (30 Mins)

Refer The Activity Sheet: Link

- Objective
- Component Required
- Procedure
- Coding
- Final Output
- Learning Check

6. Doubt Clarification /Q & A Session- Do the Learning check using these Prompt Questions:

- What is the Difference between Delay & Delay microseconds Function?
- What is the Difference between Milli & Micro Function?

7. Take Home Assignment

Create a program (any One Application) using Arduino Time Delay functions.

- Automatic Water Tank Control System
- Automatic Washing Machine
- Automatic Two-way Traffic Control System

References

Website: https://www.tutorialspoint.com/arduino/arduino_time.htm

Videos: https://www.youtube.com/watch?v=EfYaCYVupKI