Al-Mustaqbal University
Department of Biomedical Engineering
Class: fourth
Subject: micrcontroller

Lecture- 1: Arduino programming language

1. Introduction to Arduino software

Arduino is an open-source electronics platform based on easy-to-use hardware and software. The software environment is called the **Arduino IDE (Integrated Development Environment)**. It enables programmers and hobbyists to write code, upload it to microcontroller boards, and develop a wide variety of electronic projects.

The **Arduino IDE** is the official software used to write and upload code to the Arduino boards. It's free and runs on Windows, macOS, and Linux. The software allows users to write programs in a simplified version of **C/C++,** which can be directly uploaded to the hardware.

- **Key Features:**
  - **Text editor** for writing code (sketches).
  - **Verify/Compile** button to check the code for errors.
  - **Upload** button to transfer the code to an Arduino board via a USB cable.
  - **Serial monitor** to view data sent from the Arduino to your computer in real-time.

The code written for Arduino is called a **sketch**.

---

2. Arduino Programming Language

The Arduino platform uses a simplified version of **C++**, with built-in libraries to make it easier to interface with the hardware. The structure of an Arduino program typically contains two main functions:

- **setup()**: This function runs once when you start the Arduino. It's used to set initial configurations, such as pin modes (input or output).
- **loop()**: This function runs repeatedly after setup() has been executed. It's the main body of the code that continues running, enabling continuous operations.

- Example Code (Blink an LED):

```
void setup() {
  pinMode(13, OUTPUT); // Set pin 13 as output
}
void loop() {
  digitalWrite(13, HIGH); // Turn the LED on
  delay(1000);            // Wait for 1 second
  digitalWrite(13, LOW);  // Turn the LED off
  delay(1000);            // Wait for 1 second
}
```

This simple sketch blinks an LED connected to pin 13 of the Arduino board.

3. How to Install Arduino IDE

- **Step 1**: Download the latest version of Arduino IDE from the official Arduino website.
- **Step 2**: Follow the installation steps based on your operating system (Windows, macOS, Linux).
- **Step 3**: Connect your Arduino board to the computer using a USB cable.
- **Step 4**: Launch the IDE, select the appropriate **board** and **port** from the "Tools" menu.
- **Step 5**: You’re ready to start programming and uploading sketches to your Arduino board.

4. Components of Arduino IDE

a. Editor Panel

- Where you write and modify your sketches. It includes syntax highlighting for better code readability.

b. Toolbar

- Buttons for common actions like verifying code, uploading to the board, and opening new files.

c. Serial Monitor

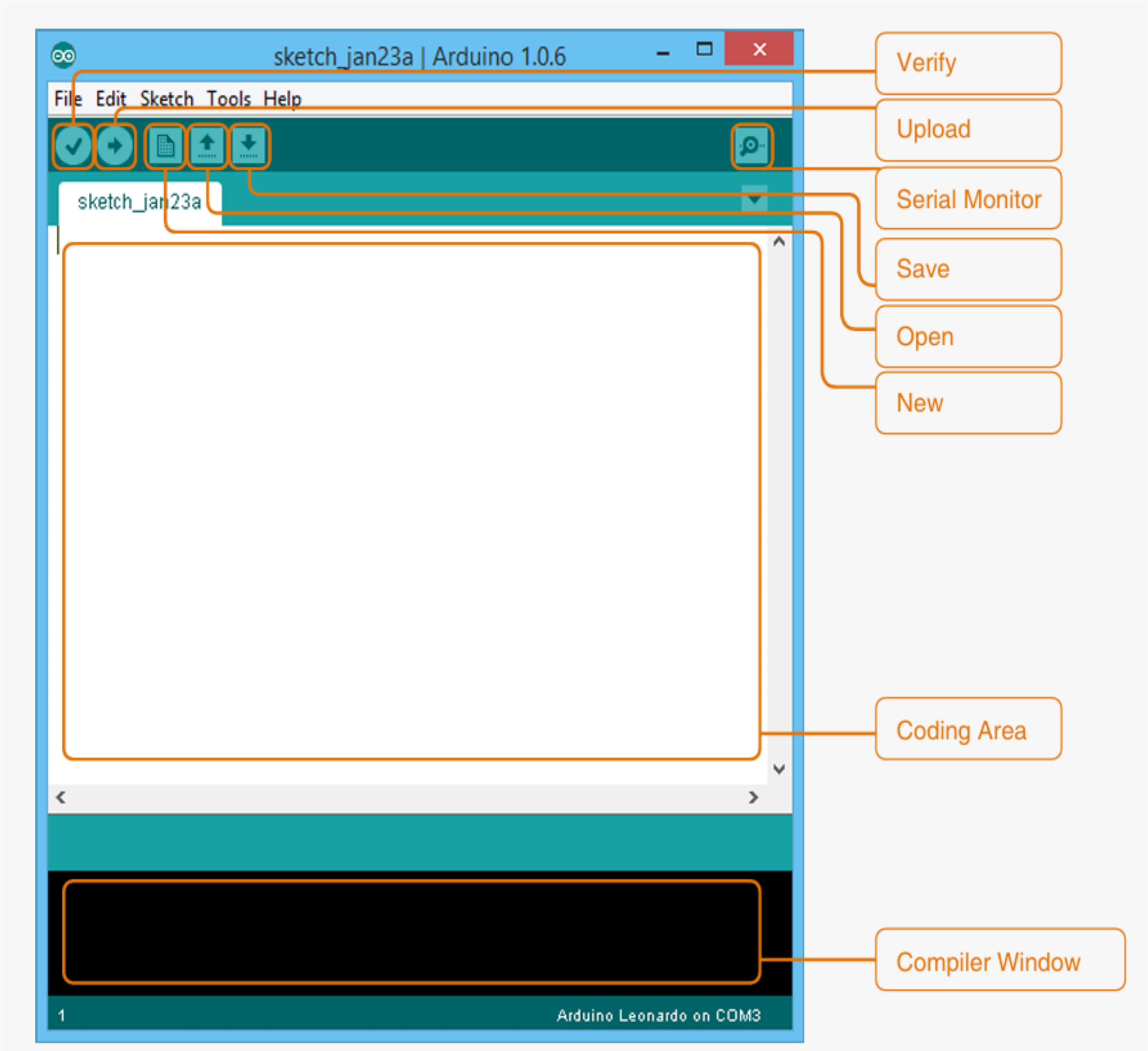
- Allows real-time communication between the Arduino and your computer. It's often used for debugging and checking sensor values or sending data between your computer and the board.

d. Board and Port Selection

- You need to select the specific Arduino board you are using (e.g., Arduino Uno, Mega) and the correct USB port that the board is connected to.

e. Library Manager

- Arduino comes with a rich set of libraries that extend functionality, such as working with sensors, motors, displays, and networking. You can add new libraries or manage the existing ones.



5. Uploading Code to Arduino

Once you've written your sketch, the next step is to upload it to the Arduino board.

Steps to Upload:

1. Click the **Verify** button to check for any errors in your code.
2. If no errors are found, click **Upload**. The code is compiled and sent to the microcontroller.
3. Once uploaded, the Arduino board will automatically start executing the program.

6. Using Libraries in Arduino IDE

Libraries are pre-written code that makes it easy to use various hardware components such as sensors, displays, or communication modules. For example, the **LiquidCrystal** library can be used to control LCD screens.

Steps to Include a Library:

1. Go to **Sketch** > **Include Library** > **Manage Libraries**.
2. Search for the library you need (e.g., LiquidCrystal for LCDs).
3. Click **Install**.
4. You can now include the library in your code with #include <LiquidCrystal.h>.

7. Debugging in Arduino IDE

Although Arduino IDE does not have an advanced debugging tool, you can use the **Serial Monitor** to help debug by printing variable values at different points in the program.

Example:

```
void setup() {
  Serial.begin(9600); // Initialize Serial communication at 9600 baud
}
void loop() {
  int sensorValue = analogRead(A0); // Read sensor value
  Serial.println(sensorValue);     // Print sensor value to monitor
  delay(1000);                     // Wait 1 second
}
```

This will print the sensor values from analog pin A0 every second.

8. Advanced Features

a. External Libraries

You can create your own libraries or add external ones to expand the functionality.

b. Real-Time Programming with Interrupts

Interrupts allow you to stop the normal flow of code and execute a function when an external event (such as a button press) happens.

c. Multiple Arduino Boards

Arduino IDE supports multiple boards like the **Arduino Uno**, **Mega**, **Nano**, and others. Advanced users can work with multiple boards for complex projects.

9. Applications of Arduino

Arduino is widely used in:

- **Home automation** (e.g., controlling lights, fans, or security systems).
- **Robotics** (e.g., building robots with motors, sensors, and controllers).
- **IoT projects** (e.g., connecting sensors and devices to the cloud).
- **Wearable electronics** (e.g., smart watches or health monitoring devices).