Experiment 1.1

Student Name: Updesh Kaur Benipal **UID:** 21ICS1021
Branch: CSE **Section/Group:** 646-B
Semester: 5th **Date of Performance:**
Subject Name: Internet of Things Lab **Subject Code:** 21CSP-344

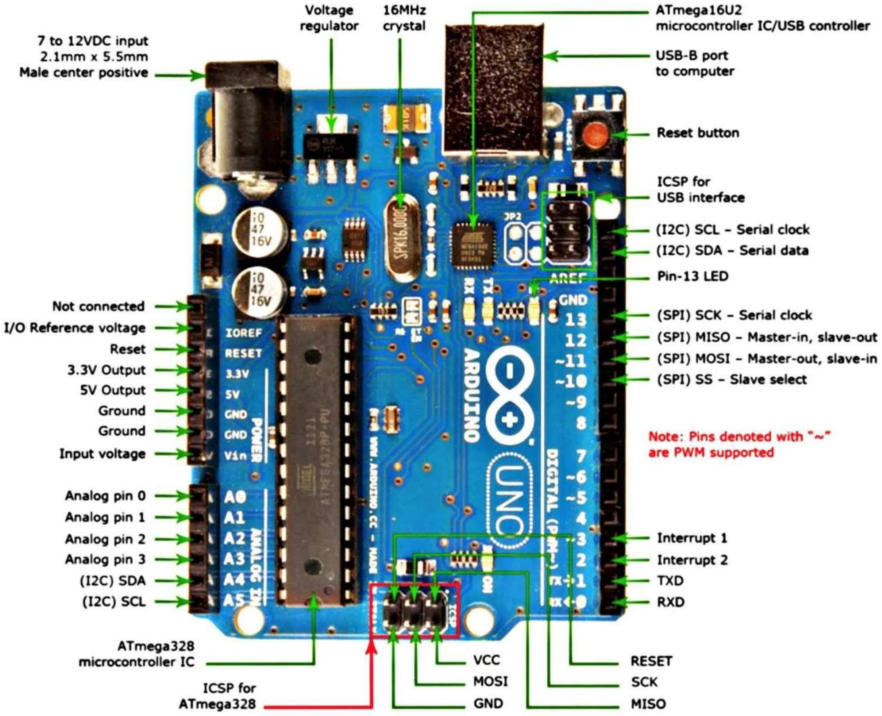
Aim: To Assemble Arduino Uno with the system and perform necessary software installation.

Objectives:

1. To study hardware and software related to IoT
2. To understand the function of Arduino Uno and other controllers.

Components Used: Arduino Uno, Connecting Cable

Diagram:



Note: Pins denoted with "~" are PWM supported

Important components:

Microcontroller (ATmega328P): This is the brain of the Arduino Uno. It's an 8-bit AVR microcontroller that runs at 16 MHz and has 32KB of flash memory for program storage, 2KBof SRAM for data storage, and 1KB of EEPROM for non-volatile data storage.

Digital I/O Pins: The Arduino Uno has 14 digital input/output pins, labeled from 0 to 13. These pins can be configured as either inputs or outputs and used to interact with digital components like LEDs, switches, sensors, etc.

Analog Input Pins: There are 6 analog input pins (labeled A0 to A5) on the board. These pins can read analog voltages from sensors or other analog devices, allowing you to measure variables like light intensity, temperature, etc.

PWM Pins: Some of the digital pins (3, 5, 6, 9, 10, and 11) are capable of generating Pulse Width Modulation (PWM) signals. PWM is used for controlling the intensity of analog- like outputs such as controlling the brightness of LEDs or the speed of motors.

Voltage Regulator: The board has a voltage regulator that can accept an input voltage ranging from 7V to 12V through the DC power jack or 5V through the USB connection. It regulates the voltage to provide a stable 5V supply for the microcontroller and other components.

Crystal Oscillator: The onboard crystal oscillator (16 MHz) provides the clock signal needed for the microcontroller to execute instructions accurately.

Reset Button: The reset button allows you to restart the microcontroller or reset your program execution.

Power LED: A power LED indicates whether the board is receiving power.

TX and RX LEDs: These LEDs indicate the communication activity between the microcontroller and your computer when using the USB connection for serial communication.

USB Connector: The USB port enables you to connect the Arduino Uno to your computer for programming and communication. It also provides power to the board.

ICSP Header: The In-Circuit Serial Programming header allows you to program the microcontroller using an external programmer if needed.

Analog Reference Pins: You can set the reference voltage for analog-to-digital conversionusing these pins. The default is the internal 5V reference, but you can use an external reference if required.

Power Pins: In addition to the regulated 5V supply, the board also provides unregulated 3.3Vand 5V pins, as well as ground (GND) pins for power and ground connections.

Serial Communication Pins (TX and RX): These pins facilitate serial communication with other devices like computers, other microcontrollers, or serial modules.

Theory and Procedure:

An Arduino is actually a micro controller-based kit. It is basically used in communications and in controlling or operating many devices. Arduino UNO board is the most popular board in the Arduino board family. It consists of two memories- Program memory and the data memory.

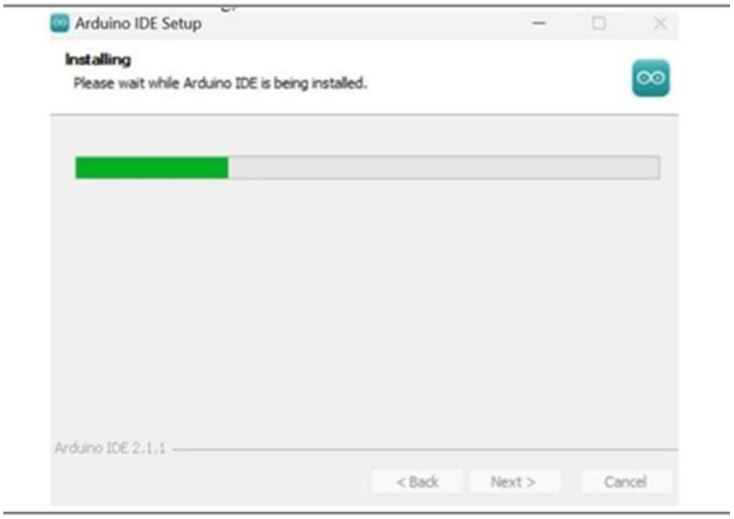
The most important advantage with Arduino is the programs can be directly loaded to the device without requiring any hardware programmer to burn the program. This is done because of the presence of the 0.5KB of Boot-loader, which allows the program to be burned into the circuit. All we have to do is to download the Arduino software and writing the code.

1. Visit **http://www.arduino.cc/en/main/software** to download the latest Arduino IDE version for your computer's operating system. There are versions for Windows, Mac, and Linux systems. At the download page, click on the "Windows Installer" option for the easiest installation.
2. Save the .exe file to your hard drive.
3. Open the .exe file.
4. Click the button to agree to the licensing agreement.
5. Select which folder to install the program to, then click "Install".
6. Wait for the program to finish installing, and then click "Close".

Configuring the Arduino IDE:
The next thing to do is to make sure the software is set up for your particular Arduino board.
Go to the "Tools" drop-down menu, and find "Board".
Another menu will appear, where you can select from a list of Arduino models. I have the Arduino Uno R3, so I chose "Arduino Uno".

Result:

Arduino Uno is successfully installed.

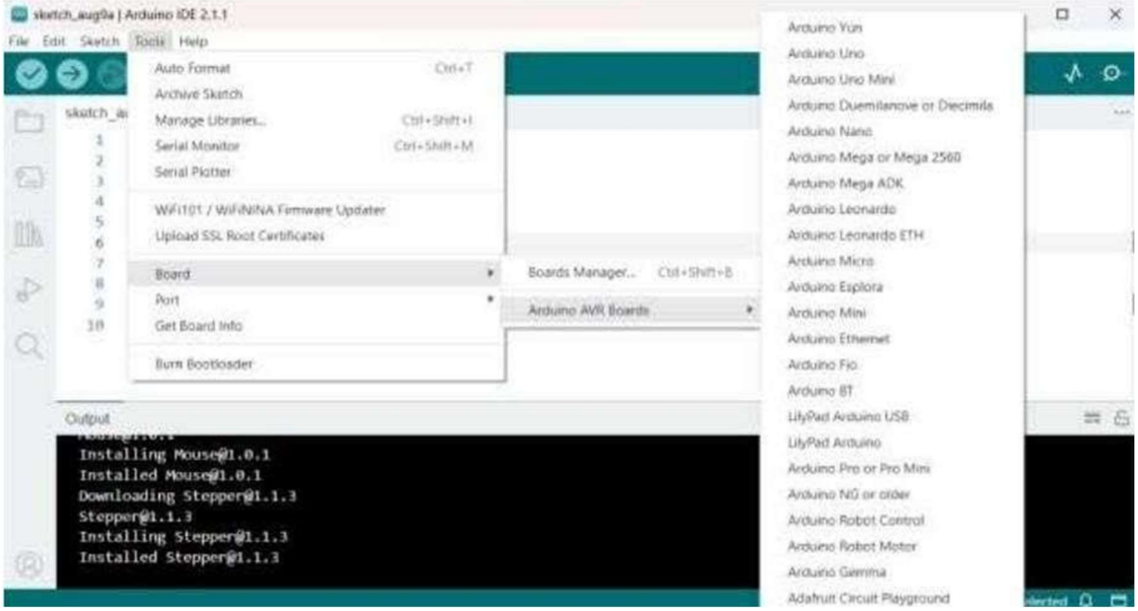


Arduino IDE Setup

Installing

Please wait while Arduino IDE is being installed.

Arduino IDE 2.1.1



sketch_aug9a | Arduino IDE 2.1.1

```
Mouse@1.0.1
Installing Mouse@1.0.1
Installed Mouse@1.0.1
Downloading Stepper@1.1.3
Stepper@1.1.3
Installing Stepper@1.1.3
Installed Stepper@1.1.3
```