Department of Computer Science & Engineering

SUBJECT: Internet of Things Architecture and its Protocl Lab
(22CSP-329/22ITP-329)

BATCH: B.E. CSE (2022-2026)

B.E. III Year – V Semester

ACADEMIC SESSION (JULY-DEC 2024)

Lab Manual

Chandigarh University

Gharuan, Mohali

Prepared By: Parveen Badoni Verified By: HOD Sign:

CONTENTS

IoT Lab Safety/General Guidelines

1. After every hardware connections and beforeturning on Power Supply, make sure you get permission from your lab supervisor.
2. Variable voltage equipment should never be changed or modified in any way.
3. Always turn off a voltage power supply when you are attaching it.
4. Whenever you can, avoid using extension cords.
5. Always tie back hair that is chin-length or longer.
6. Make sure that loose clothing or dangling jewelry is secured, or avoid wearing it in the first place.
7. Never wear sandals or other open-toed shoes in the lab. Footwear should always cover the foot completely.
8. When working with Bunsen burners, lighted splints, matches, etc., acrylic nails are not allowed.
9. Always keep your work area(s) tidy and clean.
10. Make sure that all eye wash stations, emergency showers, fire extinguishers, and exits are always unobstructed and accessible.
11. Only materials you require for your work should be kept in your work area. Everything else should be stored safely out of the way.
12. Only lightweight items should be stored on top of cabinets; heavier items should always be kept at the bottom.
13. Any equipment that requires airflow or ventilation to prevent overheating should always be kept clear.
14. Before leaving the lab or eating, always wash your hands.
15. After performing an experiment, you should always wash your hands with soap and water.
16. When using lab equipment and chemicals, be sure to keep your hands away from your body, mouth, eyes, and face.
17. Ensure you are fully aware of your facility's/building's evacuation procedures.
18. Know emergency phone numbers to use to call for help in case of an emergency.
19. Make sure you are aware of where your lab's exits and fire alarms are located.
20. If there is a fire drill, be sure to turn off all electrical equipment and close all containers.
21. Do not chew gum, drink, or eat while working in the lab.
22. Never use lab equipment that you are not approved or trained by your supervisor to operate.
23. If an instrument or piece of equipment fails during use, or isn't operating properly, report the issue to a lab in charge right away. Never try to repair an equipment problem on your own.
24. Never leave an ongoing experiment unattended.
25. Make sure you always follow the proper procedures for disposing lab waste.
26. Report all injuries, accidents, and broken equipment or glass right away, even if the incident seems small or unimportant.
27. If you notice any unsafe conditions in the lab, let your supervisor know as soon as possible.

Associate Director
Department of Computer Science & Engineering
Chandigarh University
Gharuan, Mohali, Punjab (INDIA)

1. University-Vision and Mission

Vision:

To be globally recognized as a Centre of Excellence for Research, Innovation, Entrepreneurship and disseminating knowledge by providing inspirational learning to produce professional leaders for serving the society

Mission:

Providing world-class infrastructure, renowned academicians and ideal environment for Research, Innovation, Consultancy and Entrepreneurship relevant to the society.

Offering programs & courses in consonance with National policies for nation building and meeting global challenges.

Designing Curriculum to match International standards needs of Industry, civil society and for inculcation of traits of Creative Thinking and Critical Analysis as well as Human and Ethical values.

Ensuring students delight by meeting their aspirations through blended learning, corporate mentoring, professional grooming, flexible curriculum and healthy atmosphere based on co- curricular and extra-curricular activities.

Creating a scientific, transparent and objective examination/evaluation system to ensure an ideal certification.

Establishing strategic relationships with leading National and International corporates and universities for academic as well as research collaborations.

Contributing for creation of healthy, vibrant and sustainable society by involving in Institutional Social Responsibility (ISR) activities like rural development, welfare of senior citizens, women empowerment, community service, health and hygiene awareness and environmental protection.

2. Department-Vision and Mission

Vision:

To be recognized as a leading Computer Science and Engineering department through effective teaching practices and excellence in research and innovation for creating competent professionals with ethics, values and entrepreneurial attitude to deliver service to society and to meet the current industry standards at the global level.

Mission:

M1: To provide practical knowledge using state-of-the-art technological support for the experiential learning of our students.

M2: To provide industry recommended curriculum and transparent assessment for quality learning experiences.

M3: To create global linkages for interdisciplinary collaborative learning and research.

M4: To nurture advanced learning platform for research and innovation for students' profound future growth.

M5: To inculcate leadership qualities and strong ethical values through value based education.

3. Program Educational Objectives (PEOs)

The Program Educational Objectives of the Computer Science & Engineering undergraduate program are for graduates to achieve the following, within few years of graduation. The graduates of Computer Science & Engineering Program will

PEO 1: To produce computer science graduate engineers with an ability to comprehend, understand and analyze real life problems for providing sustainable solutions in the light of disruptive technologies.

PEO 2: To inculcate life-long learning skills in graduates preparing them for work in changing environments and multidisciplinary teams in order to enhance their capability being globally employable.

PEO 3: To instill leadership qualities in graduates with a sense of confidence, professionalism and ethical attitude to produce professional leaders for serving the society.

PEO 4: To make the graduates adaptable to changing career opportunities who have the potential to excel in industry/ public sector/ higher studies or entrepreneurship exhibiting global competitiveness.

.

4. Program Outcomes (POs)

PO1: Engineering knowledge: Apply the knowledge of Mathematics, Science, Engineering fundamentals and computer science fundamental and strategies which have the solution of complex computer science engineering problems.

PO2: Problem analysis: Identify, formulate, research literature, and analyze complex computer science engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex database and software engineering problems and design system components or processes that meet the specified needs with appropriate considerations for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of software engineering & networking based experiments, analysis and Interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern Computer science engineering and IT tools including prediction and modeling to complex database or software engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess social, health, safety, legal and cultural issues and the consequent responsibilities relevant to the Professional Computer Science & Engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional computer science and engineering solutions in social and environmental contexts, and demonstrate the knowledge of, and need for sustainable development goals.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of computer science engineering practice

PO9: Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex computer science engineering activities with the engineering community like CSI society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the computer science engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life- long learning in the broadest context to technological change.

5. Student Outcomes

The Bachelor of Engineering is a programme offered by the Department of Computer Science & Engineering in accordance with the Student Outcome of Computing Accreditation Commission (CAC) and Engineering Accreditation Commission (EAC) of ABET. The Student Outcomes are asfollows:

Student Outcomes according to Computing Accreditation Commission (CAC)

SO 1. Analyze a complex computing problem and apply principles of computing and otherrelevant disciplines to identify solutions.
SO 2. Design, implement and evaluate a computing-based solution to meet a given set ofcomputing requirements in the context of the program's discipline.
SO 3. Communicate effectively in a variety of professional contexts.
SO 4. Recognize professional responsibilities and make informed judgments in computingpractice based on legal and ethical principles.
SO 5. Function effectively as a member or leader of a team engaged in activities appropriate tothe program's discipline.
SO 6. Apply computer science theory and software development fundamentals to producecomputing-based solutions.

Student Outcomes according to Engineering Accreditation Commission (EAC)

SO 1. An ability to identify, formulates, and solve complex engineering problems by applying principles of engineering, science, and mathematics
SO 2. An ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as a global, cultural, social, environmental, and economic factor
SO 3. An ability to communicate effectively with a range of audiences
SO 4. An ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts
SO 5. An ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives.
SO 6. An ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions.
SO 7. An ability to acquire and apply new knowledge as needed, using appropriate learning strategies.

Program Specific Outcomes

PSO1. To acquire proficiency in developing and implementing efficient solutions using emerging technologies, platforms and Free and Open-Source Software (FOSS).

PSO2. To gain critical understanding of hardware and software tools catering to the contemporary needs of the IT industry.

Code of Ethics

I. To uphold the highest standards of integrity, responsible behavior, and ethical conduct in professional activities.

1. To hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment;

2. To improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems;

3. To avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist;

4. To avoid unlawful conduct in professional activities, and to reject bribery in all its forms;

5. To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest and realistic in stating claims or estimates based on available data, and to credit properly the contributions of others;

6. To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;

II. To treat all persons fairly and with respect, to not engage in harassment or discrimination, and to avoid injuring others.

7. To treat all persons fairly and with respect, and to not engage in discrimination based on characteristics such as race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression;

8. To not engage in harassment of any kind, including sexual harassment or bullying behavior;

9. To avoid injuring others, their property, reputation, or employment by false or malicious actions, rumors or any other verbal or physical abuses;

III. To strive to ensure this code is upheld by colleagues and co-workers.

10. To support colleagues and co-workers in following this code of ethics, to strive to ensure the code is upheld, and to not retaliate against individuals reporting a violation.

7. COURSE OBJECTIVE IoT Architecture & its Protocols Lab

(22CSP-329/22ITP-329)

1	To Explore the architecture of IoT systems, including edge computing and networking protocols like MQTT, CoAP, or HTTP.
2	Assemble the various components of IoT including sensors, actuators, controllers using communication protocols and data processing.
3	Hands on experience through various lab experiments on IoT devices, platforms and tools.

8. COURSE OUTCOMES

INTERNET OF THINGS ARCHITECTURE & ITS PROTOCOLS LAB

(22CSP-329/22ITP-329)

	Course Outcomes
1	Classification of various components of IoT system.
2	Design and deploy IoT models on Simulation based tools.
3	Formulate real time IoT data using Arduino Uno and sensors.
4	Develop an interface between controller and sensor to capture real time data.
5	Evaluation of IoT Applications in various platforms.

9. CO/PO/PSO MAPPING

Course Outcome	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	-	3	-	-	-	-	-	-	2	2	-
CO2	3	3	3	2	3	-	-	-	-	-	-	2	2	-
CO3	3	3	3	2	3	-	-	-	-	-	-	1	1	-
CO4	3	3	3	2	3	1	1	-	-	-	-	2	2	-
CO5	3	3	3	2	3	1	1	-	1	-	-	3	3	-
AVG	3	3	3	1.8	3	1	1	-	1	-	-	2	2	-

10. SYLLABUS (AS APPROVED IN BOS)

Chandigarh University, Gharuan

		L	T	P	C
22CSP 329/22ITP 329	**IoT Architecture and its Protocols Lab** Total Contact Hours: 30 Hours For CSE/IT 5th Semester **Prerequisite:** Familiarity with electronics and basic hardware components is necessary for building IoT devices. Understanding concepts like circuitry, sensors, actuators, microcontrollers, and hardware interfaces (such as GPIO) will help you design and connect devices effectively. Some experience with programming. Programming is a valuable skill for social network analysis, as it allows you to develop your own tools and algorithms. A fundamental understanding of networking protocols, such as TCP/IP, DHCP, DNS, and HTTP, is crucial for IoT development. **Co-requisite:** Knowledge of IoT protocols and standards such as MQTT (Message Queuing Telemetry Transport), CoAP (Constrained Application Protocol), Zigbee, Z-Wave, and Bluetooth Low Energy (BLE). Understanding these protocols will help you establish communication between IoT devices and gateways. Understanding of languages commonly used in IoT projects, such as Python, C, C++, Java, or JavaScript.	0	0	2	1
Max Marks-100					
Internal-60	External-40				

A. Course Description:

Internet of Things (IoT) is presently a hot technology worldwide. Government, academia, and industry are involved in different aspects of research, implementation, and business with IoT. IoT cuts across different application domain verticals ranging from civilian to defense sectors. These domains include agriculture, space, healthcare, manufacturing, construction, water, and mining, which are presently transitioning their legacy infrastructure to support IoT. Today it is possible to envision pervasive connectivity, storage, and computation, which, in turn, gives rise to building different IoT solutions. IoT-based applications such as innovative shopping system, infrastructure management in both urban and rural areas, remote health monitoring and emergency notification systems, and transportation systems, are gradually relying on IoT based systems. Therefore, it is very important to learn the fundamentals of this emerging technology.

B. Course Objectives:

1. To Explore the architecture of IoT systems, including edge computing and networking protocols like MQTT, CoAP, or HTTP.
2. Assemble the various components of IoT including sensors, actuators, controllers using communication protocols and data processing.
3. Hands on experience through various lab experiments on IoT devices, platforms and tools.

C. Course Outcomes:

1. Classification of various components of IoT system.
2. Design and deploy IoT models on Simulation based tools.
3. Formulate real time IoT data using Arduino Uno and sensors.
4. Develop an interface between controller and sensor to capture real time data.
5. Evaluation of IoT Applications in various platforms.

D. Lab Syllabus

List of Experiments

Sr No.	Experiment Name	Mapped with CO Number(s)
1	To connect Arduino Uno controller to a computer system/laptop and complete the essential software setup.	CO1
2	To Design LCD interfacing on WOKWI or Any other IoT Simulation Platform.	CO2
3	To Develop a smart traffic light management system with the help of IoT.	CO4
4	To Formulate distance of an object using an ultrasonic sensor.	CO3
5	To Design a weather station by checking Air quality of an environment with the help of IoT.	CO4
6	To Investigate real-time relationship between humidity and temperature in IoT.	CO3
7	To Assemble and Controlling of actuators using Arduino Uno.	CO4
8	To Create a Smart door lock system using RFID.	CO4
9	To develop an IoT-based system to optimize agricultural practices.	CO5
10	Lab based Mini Project	CO5

BEYOND SYLLABUS

1. Case studies of IoT in Healthcare and propose any IoT model for the healthcare sector. (CO-5)
2. Create an IoT system to predict equipment failures and schedule maintenance. (CO-4)
3. Monitor and manage energy consumption in real-time. (CO-5)
4. Design and implement a smart home system using IoT devices. (CO-4)

E. Assessment Pattern - Internal and External

The performance of students is evaluated as follows:

	Theory	
Components	**Continuous Internal Assessment (CAE)**	**Semester End Examination (SEE)**
Marks	60	40
Total Marks	100	

Internal Evaluation Component

Sr. No.	Type of Assessment	Weightage of actual conduct	Frequency of Task	Final Weightage in Internal Assessment	Remarks
1	Conduct	12 Marks per Practical	1 per practical	60 Marks per course	
2	Report	08 Marks per Practical	1 per practical		
3	Viva- Voce	10 Marks per Experiment	1 per Experiment		

CO-PO Mapping

Course Outcome	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	-	3	-	-	-	-	-	-	2	2	-
CO2	3	3	3	2	3	-	-	-	-	-	-	2	2	-
CO3	3	3	3	2	3	-	-	-	-	-	-	1	1	-
CO4	3	3	3	2	3	1	1	-	-	-	-	2	2	-
CO5	3	3	3	2	3	1	1	-	1	-	-	3	3	-
AVG	3	3	3	1.8	3	1	1	-	1	-	-	2	2	-

TEXT BOOKS

1. Practical Internet of Things for Beginners: IoT Projects with Realsense, Azure, Arduino, and Intel Edison Paperback – Import, July 2020 by Pooja Baraskar, Apress Publication, 1st Edition.
2. The IoT Hacker's Handbook A Practical Guide to Hacking the Internet of Things by Adity Gupta, Apress Publication 2021
3. Internet of Things: A Hands-On Approach by Arsheep Bahga, Vijay Madisetti, Orient Blackswan Private Limited, New Delhi Publication 2021
4. 21 IOT Experiments by Yashavant Kanetakr and Shrirang Korde, BPB Publications, 2018
5. Visual Inference for IoT Systems: A Practical Approach by Delia Velasco-Montero, Jorge Fernández-Berni, Angel Rodriguez-Vazquez, Springer 2022

REFERENCE BOOKS

1. Building Arduino Projects for the Internet of Things: Experiments with Real-World Applications by Adeel Javed, Apress Publications 2016
2. Practical Internet of Things Security: Beat IoT security threats by strengthening your security strategy and posture against IoT vulnerabilities by Brian Russell, Drew Van Duren, Packt Publishing 2016
3. A Practical Guide for IoT Solution Architects: Architecting secure, agile, economical, highly available, well performing IoT ecosystems: 1 (Internet of Things - IoT Architecture) Independently Published by Dr. Mehmet Yildiz 2019
4. Programming Arduino with LabVIEW by Marco Schwartz, Oliver Manickum, PACKT Publication 2015
5. New Arduino Communication Projects using MATLAB and Sensors: Simple technical approach by Anbazhagan k, Ambika parameswari k, Kindle Edition 2019

LIST OF EXPERIMENTS (MAPPED WITH COs)

Sr No.	Experiment Name	Mapped with CO Number(s)
1	To connect Arduino Uno controller to a computer system/laptop and complete the essential software setup.	**CO1**
2	To Design LCD interfacing on WOKWI or Any other IoT Simulation Platform.	**CO2**
3	To Develop a smart traffic light management system with the help of IoT.	**CO4**
4	To Formulate distance of an object using an ultrasonic sensor.	**CO3**
5	To Design a weather station by checking Air quality of an environment with the help of IoT.	**CO4**
6	To Investigate real-time relationship between humidity and temperature in IoT.	**CO3**
7	To Assemble and Controlling of actuators using Arduino Uno.	**CO4**
8	To Create a Smart door lock system using RFID.	**CO4**
9	To develop an IoT-based system to optimize agricultural practices.	**CO5**
10	Lab based Mini Project	**CO5**

MANUAL TO CONDUCT EACH EXPERIMENT

Experiment 1

Aim: To connect Arduino Uno controller to a computer system/laptop and complete the essential software setup.

Objectives:

1. To study hardware and software related to IoT

2. To understand the function of Arduino Uno and other controllers.

Hardware Required:

1. Arduino Uno
2. Connecting Cable

Arduino Board:

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.

In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduino's have majority of these components in common.

It consists of two memories- Program memory and the data memory.

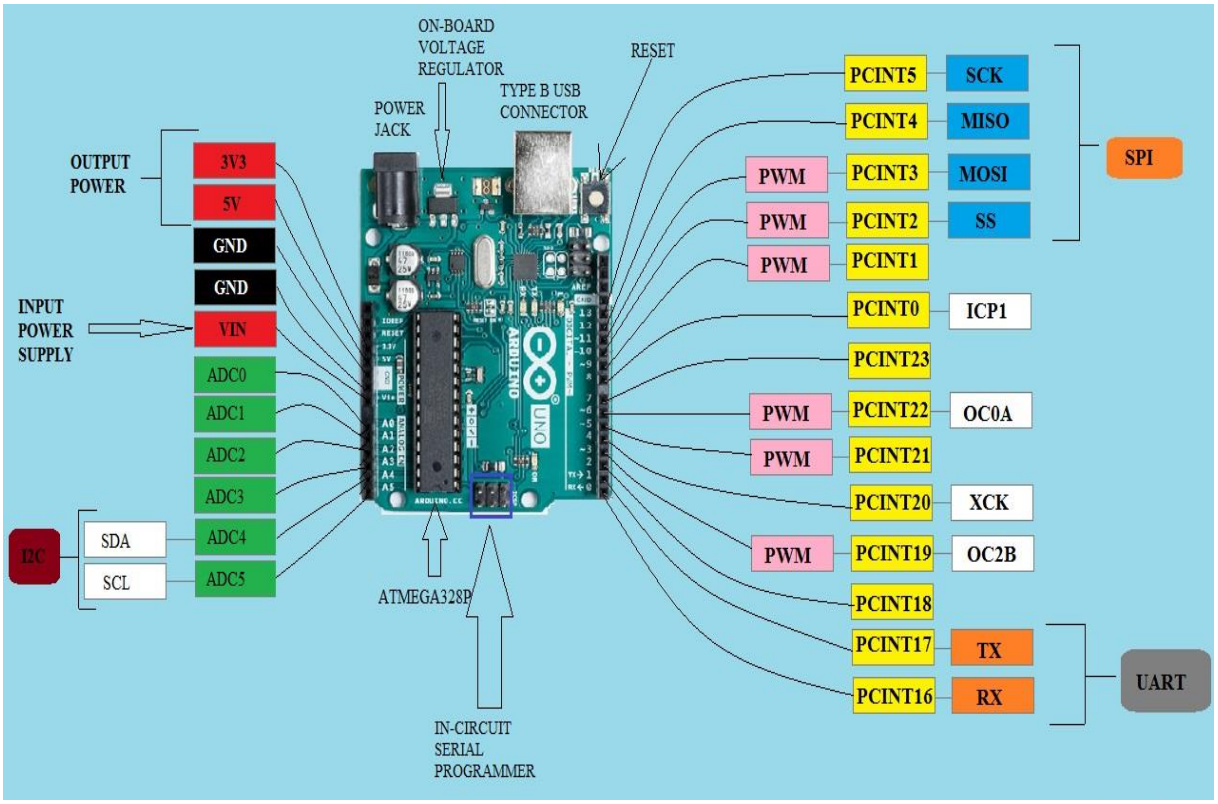


Figure: Diagram of Arduino Board

The code is stored in the flash program memory, whereas the data is stored in the data memory.

Arduino Uno consists of 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button

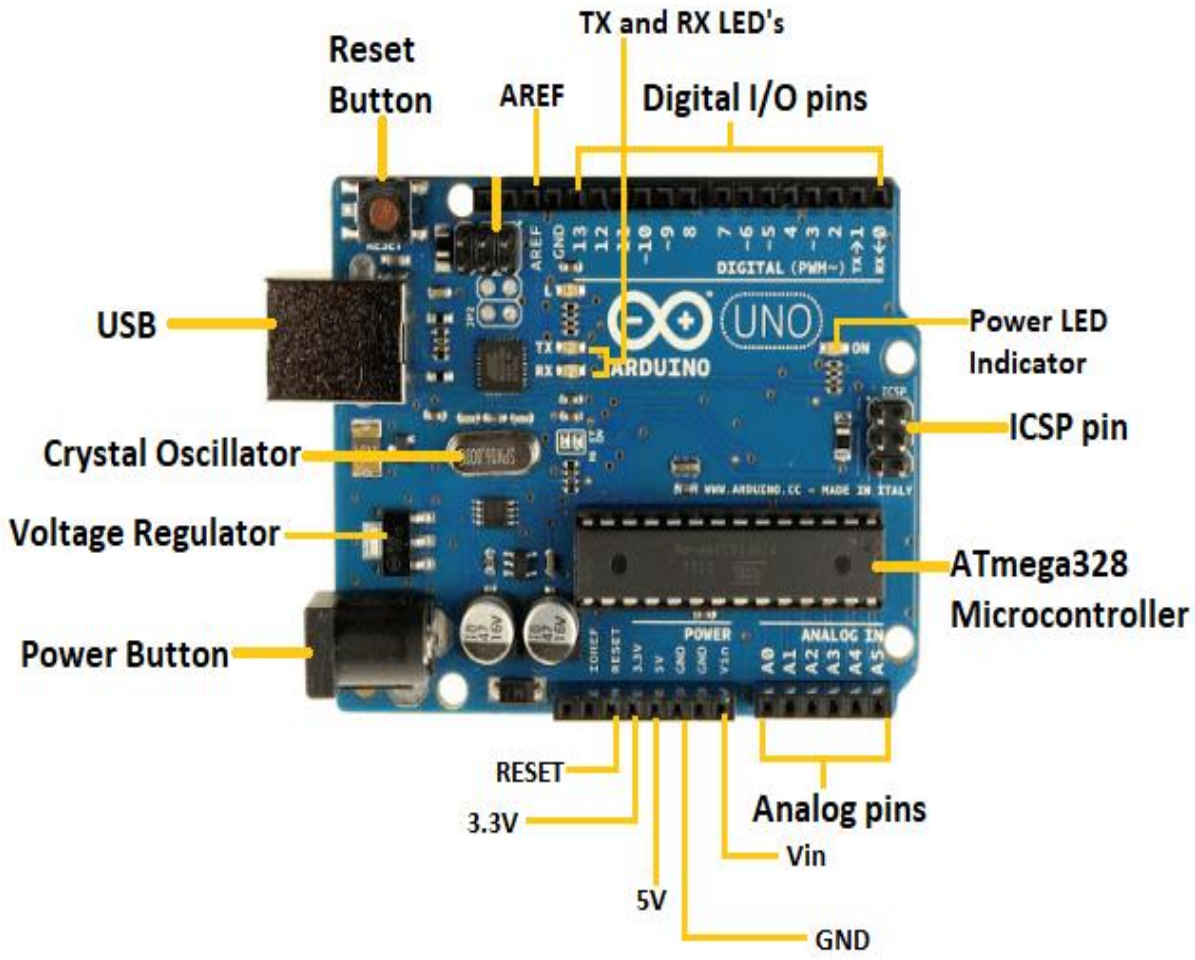


Figure: Arduino Uno

1. Power USB Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).

2. Power (Barrel Jack) Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).

3. Voltage Regulator The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

4. Crystal Oscillator The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.

5,17.Arduino Reset You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).

6,7,8,9.Pins (3.3, 5, GND, Vin)

• 3.3V (6) − Supply 3.3 output volt

• 5V (7) − Supply 5 output volt

• Most of the components used with Arduino board works fine with 3.3 volt and 5 volt.

• GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit.

• Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.

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

11. Main micro controller Each Arduino board has its own micro controller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The micro controllers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.

12. ICSP pin Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.

13. Power LED indicator This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.

14. TX and RX LEDs On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

15. Digital I/O

• 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 input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.

6.AREF

AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins

Program an Arduino:

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.

The Arduino tool window consists of the toolbar with the buttons like verify, upload, new, open, save, serial monitor.

It also consists of a text editor to write the code, a message area which displays the feedback like showing the errors, the text console which displays the output and a series of menus like the File, Edit, Tools menu.

Steps to program an Arduino

1. Programs written in Arduino are known as sketches. A basic sketch consists of 3 parts

a. Declaration of Variables

b. Initialisation: It is written in the setup () function.

c. Control code: It is written in the loop () function.

2. The sketch is saved with .ino extension. Any operations like verifying, opening a sketch, saving a sketch can be done using the buttons on the toolbar or using the tool menu.

3. The sketch should be stored in the sketchbook directory.

4. Chose the proper board from the tools menu and the serial port numbers.

5. Click on the upload button or chose upload from the tools menu. Thus the code is uploaded by the boot loader onto the micro controller.

Basic Adruino functions are:

1. digitalRead(pin): Reads the digital value at the given pin.

2. digitalWrite(pin, value): Writes the digital value to the given pin.

3. pinMode(pin, mode): Sets the pin to input or output mode.

4. analogRead(pin): Reads and returns the value.

5. analogWrite(pin, value): Writes the value to that pin.

6. serial.begin(baud rate): Sets the beginning of serial communication by setting the bit rate.

Advantages of Arduino Board

1. It is inexpensive

2. It comes with an open source hardware feature which enables users to develop their own kit using already available one as a reference source.

3. The Arduino software is compatible with all types of operating systems like Windows, Linux, and Macintosh etc.

4. It also comes with open source software feature which enables experienced software developers to use the Arduino code to merge with the existing programming language libraries and can be extended and modified.

5. It is easy to use for beginners.

6. We can develop an Arduino based project which can be completely stand alone or projects which involve direct communication with the software loaded in the computer.

7. It comes with an easy provision of connecting with the CPU of the computer using serial communication over USB as it contains built in power and reset circuitry.

Interfaces:

UART Peripheral:

1. A UART (Universal Asynchronous Receiver/Transmitter) is a serial interface.

2. It has only one UART module.

3. The pins (RX, TX) of the UART are connected to a USB-to-UART converter circuit and also connected to pin0 and pin1 in the digital header.

SPI Peripheral:

1. The SPI (Serial Peripheral Interface) is another serial interface. It has only one SPI module.

TWI:

1. The I2C or Two Wire Interface is an interface consisting of only two wires, serial data, and a serial clock: SDA, SCL.

2. You can reach these pins from the last two pins in the digital header or pin4 and pin5 in the analog header.

Other Controllers:
RASPERRY PI

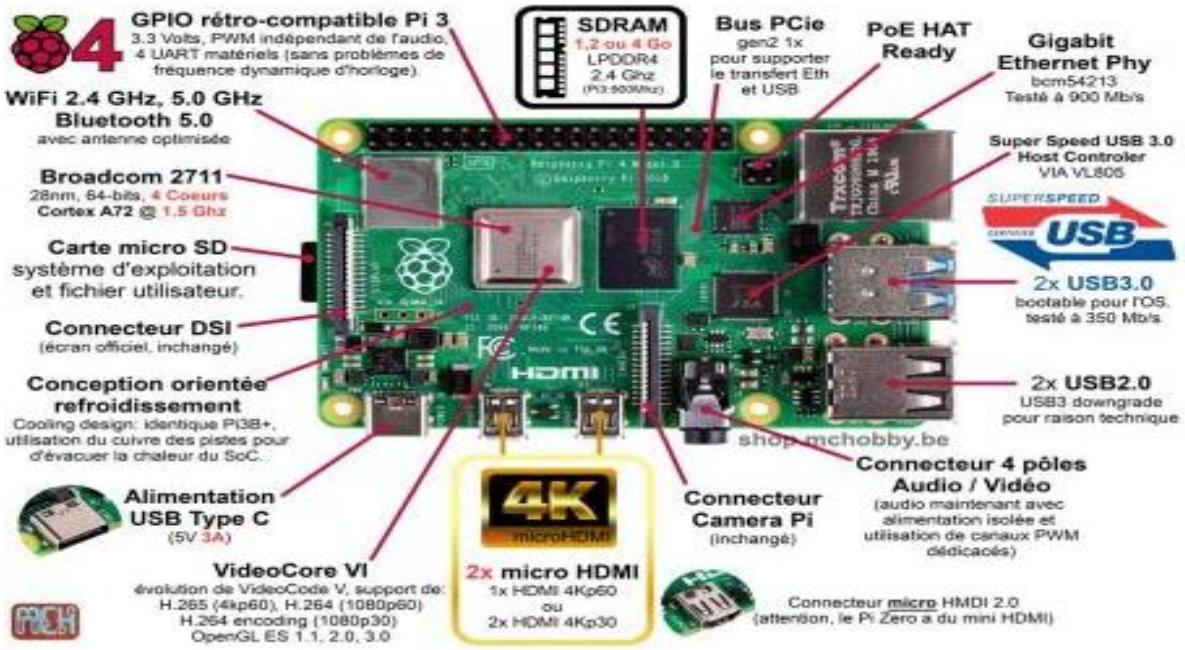


Figure: Raspberry Pi

1. The Raspberry Pi is a very cheap computer that runs Linux, but it also provides a set of GPIO (general purpose input/output) pins that allow you to control electronic components for physical computing and explore the Internet of Things (IoT).

2. Raspberry Pi was basically introduced in 2006.

3. It is particularly designed for educational use and intended for Python.

4. A Raspberry Pi is of small size i.e., of a credit card sized single board computer, which is developed in the United Kingdom (U.K) by a foundation called Raspberry Pi.

5. There have been three generations of Raspberry Pis: Pi 1, Pi 2, and Pi 3

6. The first generation of Raspberry (Pi 1) was released in the year 2012, which has two types of models namely model A and model B.

7. Raspberry Pi can be plugged into a TV, computer monitor, and it uses a standard keyboard and mouse.

8. It is user friendly as can be handled by all the age groups.

9. It does everything you would expect a desktop computer to do like word-processing, browsing the internet spreadsheets, playing games to playing high definition videos.

10. All models feature on a broadcom system on a chip (SOC), which includes chip graphics processing unit GPU (a Video Core IV), an ARM compatible and CPU.

11. The CPU speed ranges from 700 MHz to 1.2 GHz for the Pi 3 and on board memory range from 256 MB to 1 GB RAM.

12. An operating system is stored in the secured digital SD cards and program memory in either the MicroSDHC or SDHC sizes.

13. Most boards have one to four USB slots, composite video output, HDMI and a 3.5 mm phone jack for audio. Some models have WiFi and Bluetooth.

14. Several generations of Raspberry Pis have been released.

15. All models feature a Broadcom system on a chip (SoC) with an integrated ARM compatible central processing unit (CPU) and on-chip graphics-processing unit (GPU).

16. Processor speed ranges from 700 MHz to 1.4 GHz for the Pi 3 Model B+ or 1.5 GHz for the Pi 4; on-board memory ranges from 256 MB to 1 GB with up to 4 GB available on the Pi 4 random-access memory (RAM).

17. Secure Digital (SD) cards in MicroSDHC form factor (SDHC on early models) are used to store the operating system and program memory.

18. The boards have one to five USB ports. For video output, HDMI and composite video are supported, with a standard 3.5 mm tip-ring-sleeve jack for audio output.

19. Lower-level output is provided by a number of GPIO pins, which support common protocols like I²C. The B-models have an 8P8C Ethernet port and the Pi 3 and Pi Zero W have on-board Wi-Fi and Bluetooth.

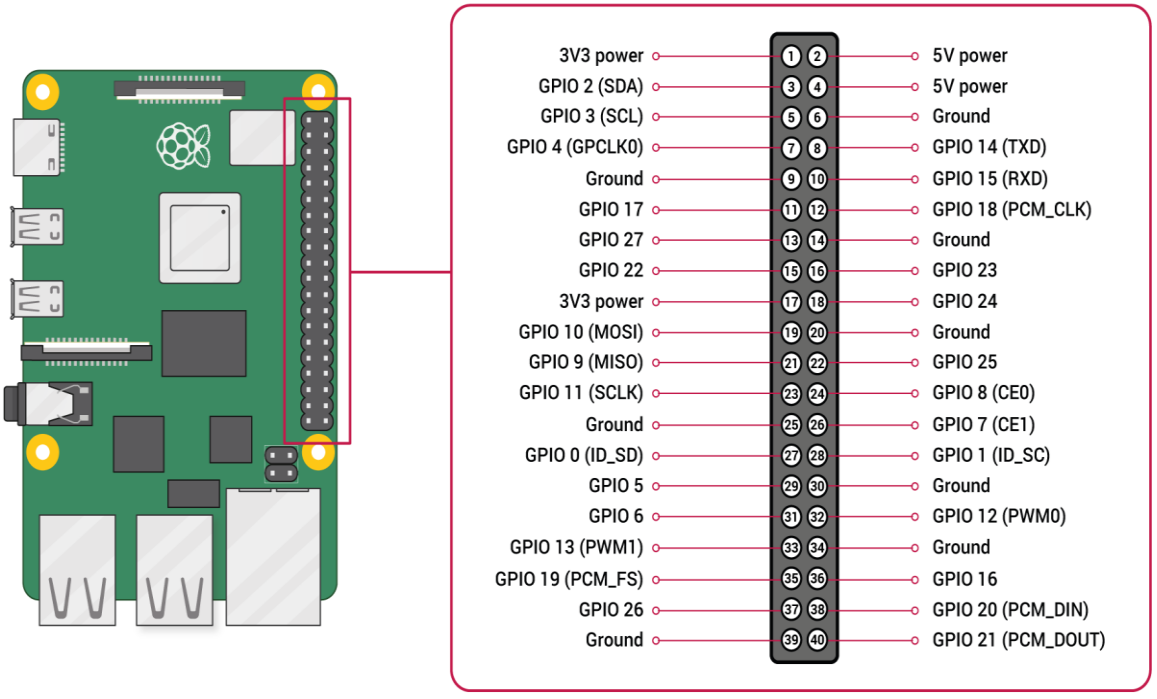


Figure: Raspberry Pi Pin Description

Components and Peripherals:
Voltages: Two 5V pins and two 3V3 pins are present on the board, as well as a number of ground pins (0V). The remaining pins are all general purpose 3V3 pins

A GPIO pin designated as an output pin can be set to high (3V3) or low (0V). A GPIO pin designated as an input pin can be read as high (3V3) or low (0V).

Processor & RAM: Raspberry based on ARM11 processor. Latest version supports 700MHz processor and 512MB SDRAM. The Central processing unit is the brain of the raspberry pi board and that is responsible for carrying out the instructions of the computer through logical and mathematical operations.

Ethernet: The Ethernet port of the raspberry pi is the main gateway for communicating with additional devices. The raspberry pi Ethernet port is used to plug your home router to access the Internet.

USB Ports: It has 2 USB ports. USB port provides current upto 100mA. For connecting devices that draw current more than 100mA, an external USB powered hub is required.

Ethernet Port: It has standard RJ45 Ethernet port. Connect Ethernet cable or USB wifi adapter to provide internet connectivity.

HDMI Output: It supports both audio and video output. Connect raspberry Pi to monitor using HDMI cable.

Composite video Output: Raspberry comes with a composite video output with an RCA jack that supports both PAL and NTSC video output.

Audio Output: It has 3.5mm audio output jack. This audio jack is used for providing audio output to old television along with RCA jack for video.

GPIO Pins: It has a number of general-purpose input/output pins. These pins are used to connect other electronic components. For example, you can connect it to the temperature sensor to transmit digital data.

Display Serial Interface (DSI): DSI interface are used to connect an LCD panel to Raspberry PI.

Cameral Serial Interface (CSI): CSI interface are used to connect a camera module to Raspberry PI.

SD Card slot: Raspberry does not have built in OS and storage. Plug in an SD card loaded with Linux to SD card slot.

Power Input: Raspberry has a micro USP connector for power input.

Memory: The raspberry pi model A board is designed with 256MB of SDRAM and model B is designed with 51MB.Raspberry pi is a small size PC compare with other PCs. The normal PCs RAM memory is available in gigabytes. But in raspberry pi board, the RAM memory is available more than 256MB or 512MB

Status LEDs: Raspberry has 5 status LEDs.

1. ACT SD card Access

2. PWR 3.3V power is present

3. FDX Full duplex LAN Connected

4. LNK Link/Network Activity

5. 100 100 Mbit LAN connected

Raspberry PI Interfaces:
It supports SPI, serial and I2C interfaces for data transfer.

Serial : Serial Interface on Raspberry has receive(Rx) and Transmit(Tx) pins for communication with serial peripherals.

SPI: Serial Peripheral Interface (SPI) is a synchronous serial data protocol used for communicating with one or more peripheral devices. In an SPI connection, there is one master device and one or more peripheral devices.

There are 5 pins Raspberry for SPI interface.

- MISO(Master In Slave Out): Master line for sending data to the peripherals.
- MOSI(Master Out Slave In): Slave Line for sending data to the master.
- SCK(Serial Clock): Clock generated by master to synchronize data transmission.
- CE0(Chip Enable 0): To enable or disable devices.
- CE1(Chip Enable 1): To enable or disable devices.

I2C: I2C Interface pins are used to connect hardware modules. I2C interface allows synchronous data transfer with two pins: SDA(data line) and SCL (Clock Line)

Features of Raspberry PPI

1. Where the system processing is huge. They can process high end programs for applications like Weather Station, Cloud server, gaming console etc. With 1.2GHz clock speed and 1 GB RAM RASPBERRY PI can perform all those advanced functions.

2. RASPBERRY PI 3 has wireless LAN and Bluetooth facility by which you can setup WIFI HOTSPOT for internet connectivity.

3. RASPBERRY PI had dedicated port for connecting touch LCD display which is a feature that completely omits the need of monitor.

4. RASPBERRY PI also has dedicated camera port so one can connect camera without any hassle to the PI board.

5. RASPBERRY PI also has PWM outputs for application use.

6. It supports HD steaming

Applications

1. Hobby projects.

2. Low cost PC/tablet/laptop

3. IoT applications

4. Media center

5. Robotics/Industrial/Home automation

6. Server/cloud server

7. Print server

8. Security monitoring

9. Web camera

10. Gaming

11. Wireless access point

INSTALLING THE ARDUINO IDE

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.

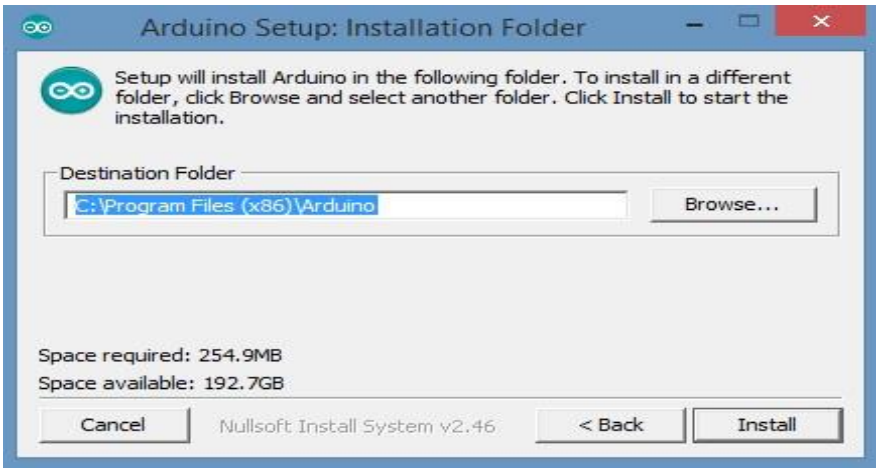
2. Save the .exe file to your hard drive.

3. Open the .exe file.

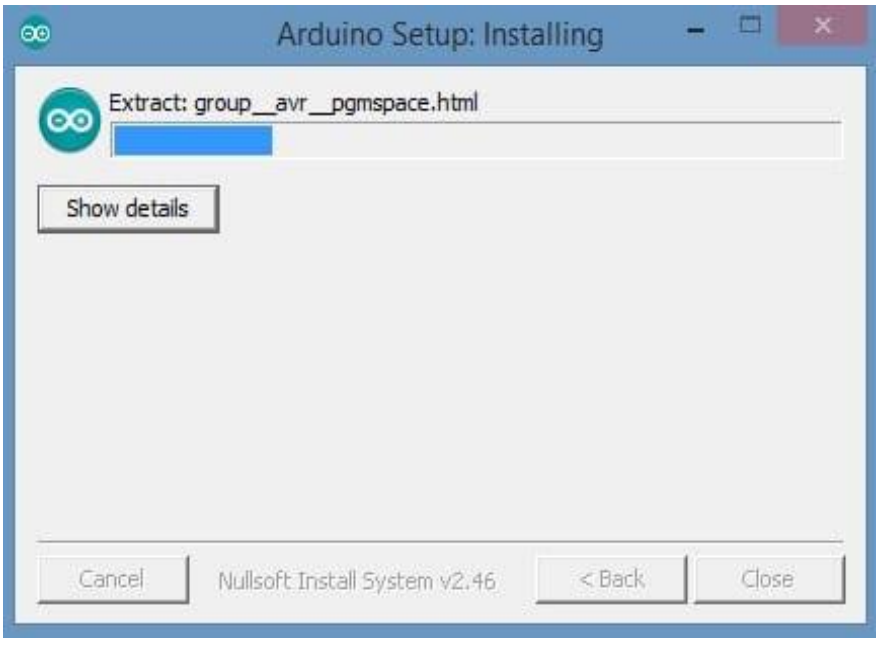
4. Click the button to agree to the licensing agreement:

5. Decide which components to install, then click "Next":

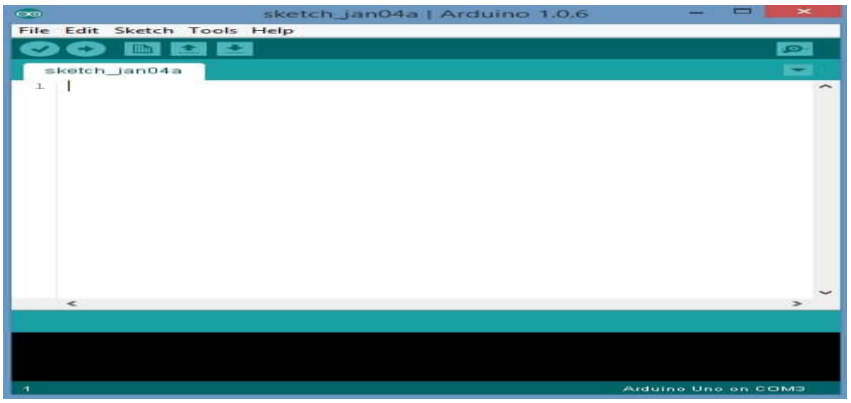
6. Select which folder to install the program to, then click “Install”:



7. Wait for the program to finish installing, and then click “Close”:

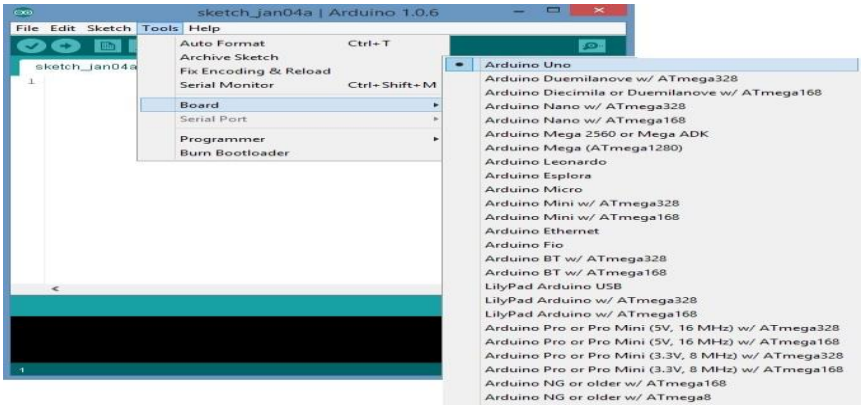


8. Now find the Arduino shortcut on your Desktop and click on it. The IDE will open up and you’ll see the code editor:



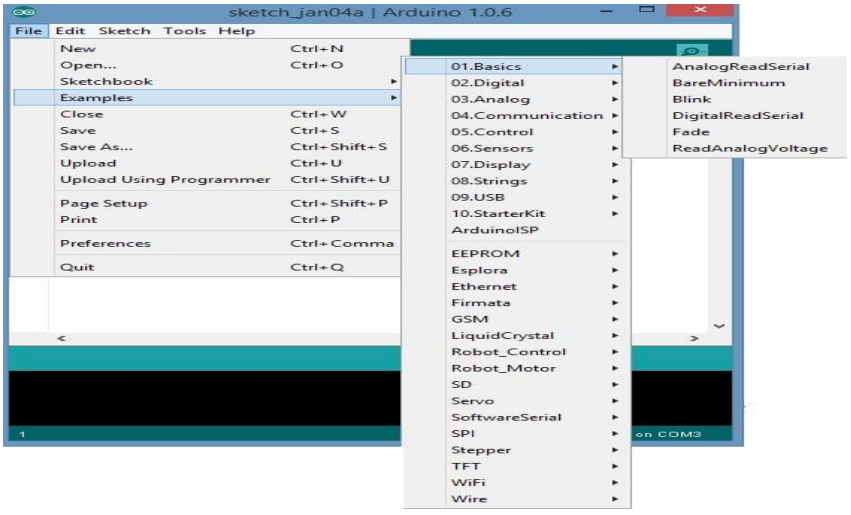
CONFIGURING THE ARDUINO IDE

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".



EXPLORING THE ARDUINO IDE

If you want, take a minute to browse through the different menus in the IDE. There is a good variety of example programs that come with the IDE in the "Examples" menu. These will help you get started with your Arduino right away without having to do lots of research:



EXPERIMENTING WITH THE ARDUINO

Play around with the example programs and try changing parts of the code to see what happens. But if you want to learn programming as a skill, it's best not to rely too much on these examples in your projects. You'll learn much more by experimenting and writing your own code from scratch.

Viva Voce:

1) Explain Raspberry Pi.

2) Explain the working of Raspberry Pi.

3) List some interesting projects can be made with Raspberry Pi.

4) Point down the uses of Raspberry Pi in IoT.

5) Define different components of a Raspberry Pi Board.

6) Discuss about NOOBS Software.

7) Deliberate the use of GPIO Pin in Raspberry Pi Boards.

8) Can Raspberry Pi be used as a server.

9) Discuss the language used by Raspberry Pi.

10) Difference between Raspberry Pi and Arduino.

11) With an example discuss the best-fit use of Raspberry Pi.

12) Deliberate on confront of overheating problem in Raspberry Pi.

13) Explain how can you measure power consumption used by Raspberry Pi.

14) List THE generations of Raspberry Pi available.

15) Difference between Raspberry pi version 3 and 4.

16) Define Arduino.

17) Discuss the stable version of Arduino Software.

18) Name the developer of Arduino.

19) With proper application discuss the use of Arduino.

20) Deliberate about IDE toolbar of Arduino.

21) Discuss the use of operator in Arduino.

22) Define the role of Sketch in Arduino.

23) Explain software structure functions.

24) Name the function used to find the length of a string.

25) List some advantages of Arduino.

26) Explain three important parts of Arduino.

27) Define the role of libraries in Arduino.

28) In which language Arduino software was written.

29) Explain the function of time in Arduino.

30) Explain the process of converting a string to upper case.

Experiment-2

Aim: To Design LCD interfacing on WOKWI or Any other IoT Simulation Platform.

Objective:

1. Learn about IoT based simulations.

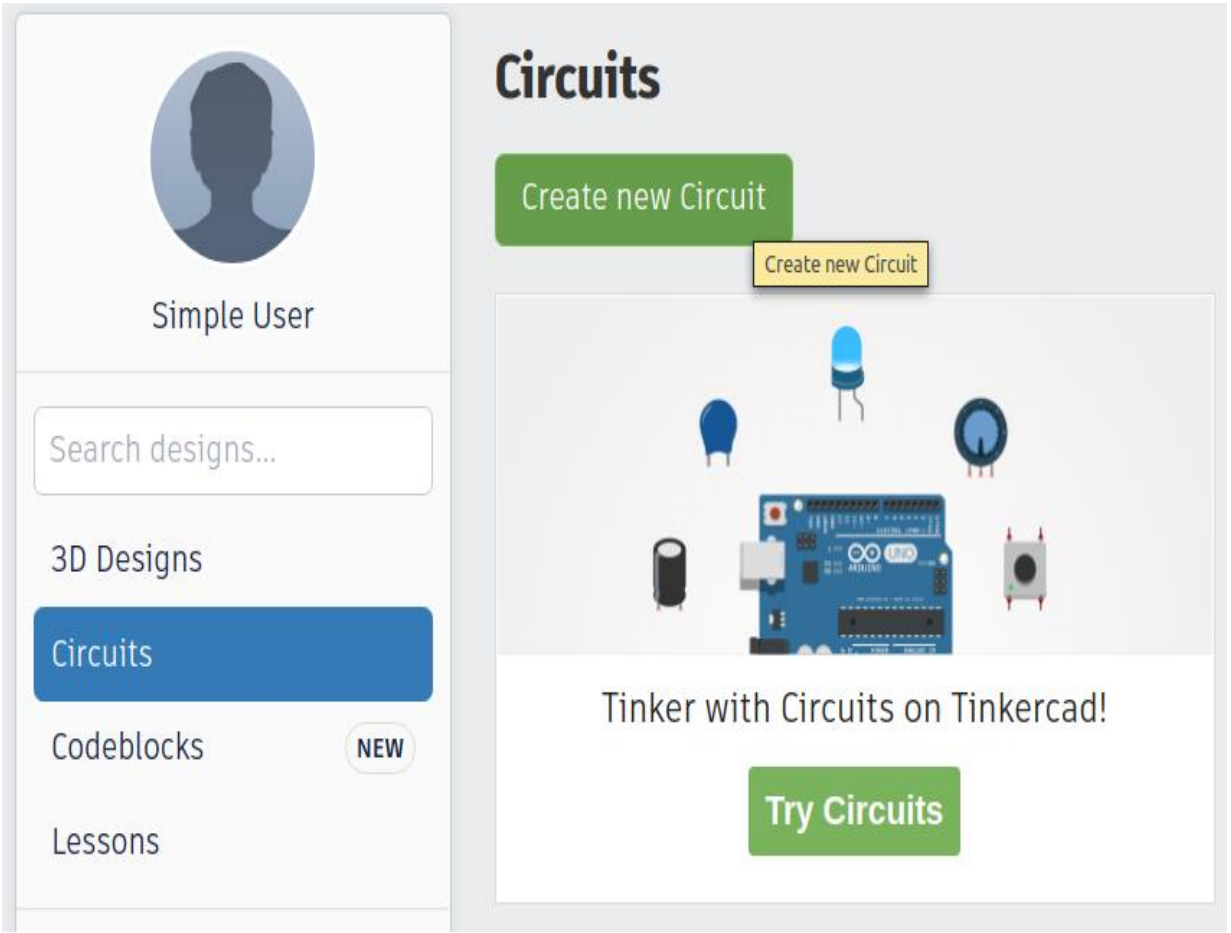
2. Testing and model in IoT based simulation platform.

Software: Tinker Cad Simulation

Introduction of Tinkercad

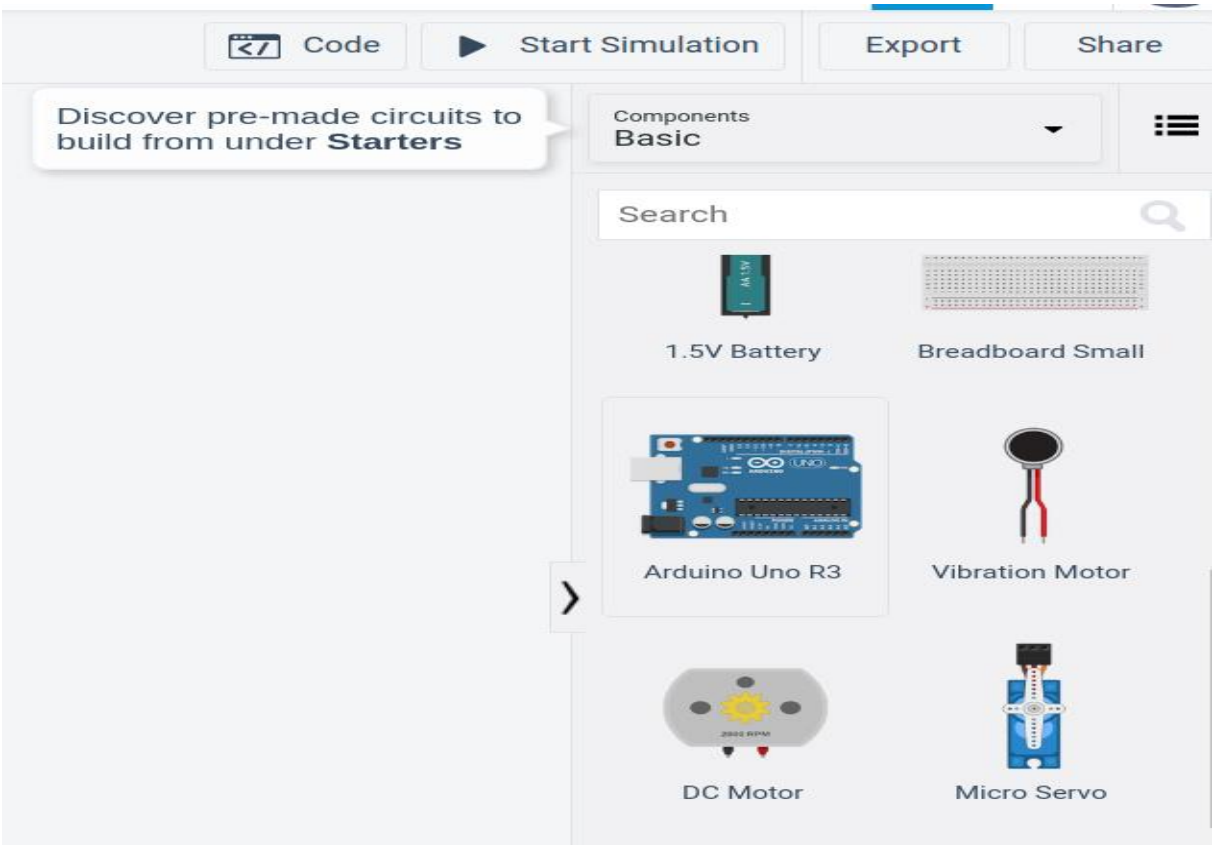
Tinkercad - https://www.tinkercad.com is an excellent tool that allows you to simulate Arduino-based systems (and a lot more). You can (perhaps you SHOULD) simulate all exercises and even your own designs before trying them on real hardware. It also allows you to do programming using blocks. You can download / copy-paste the generated code later into Arduino IDE to program the real Arduino board, rather than having to write it from scratch.

Create a new personal account on Tinkercad website (you can also use your Google account to log in). Then select Circuits on the left pane, and click Create new Circuit.

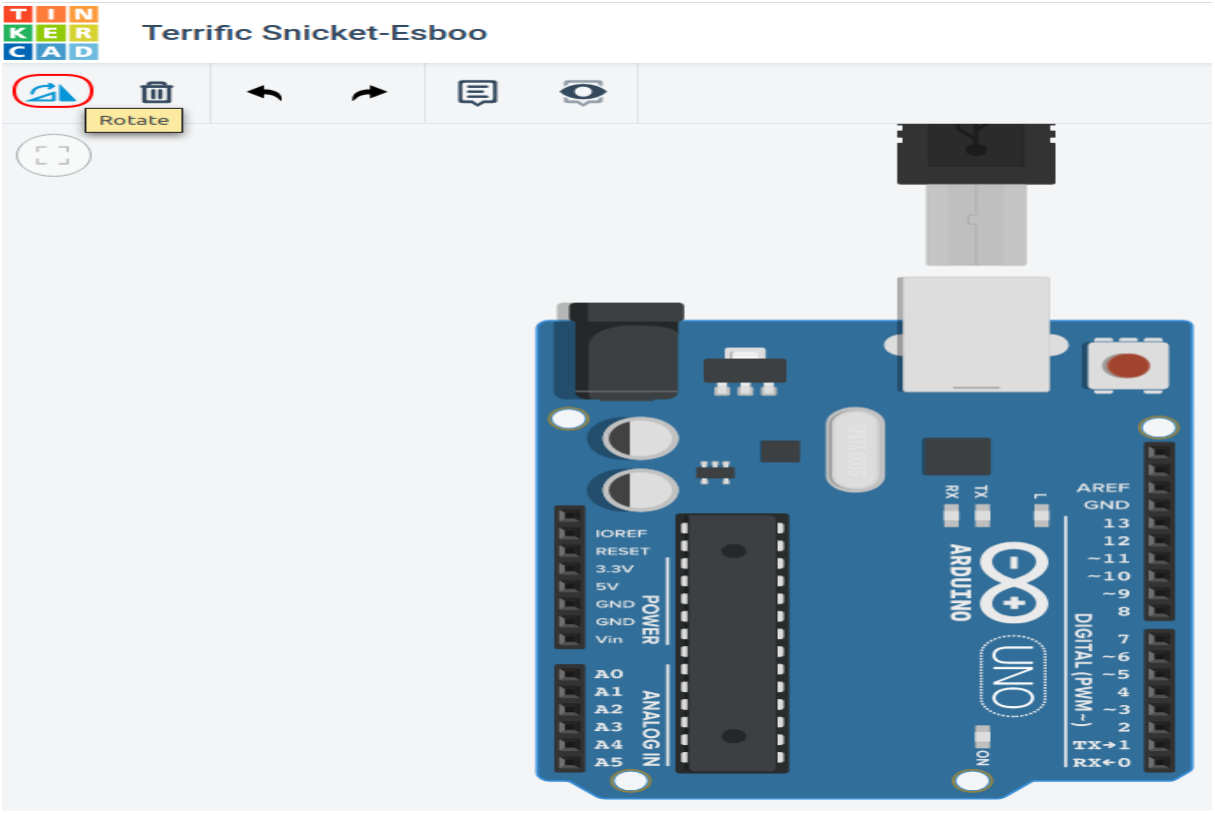


Hardware

In Components Basic, you can select Arduino Uno R3.



You can rotate it to portrait mode if you wish, which will allow more space for other components to be added.



You can add more components and wire them up as desired. Clicking on the lead of a component allows you to start a connecting wire from there. Clicking on a wire allows you to change its color.

Step-by-step guide for designing an LCD interfacing circuit using the WOKWI or TinkerCad simulation platforms:

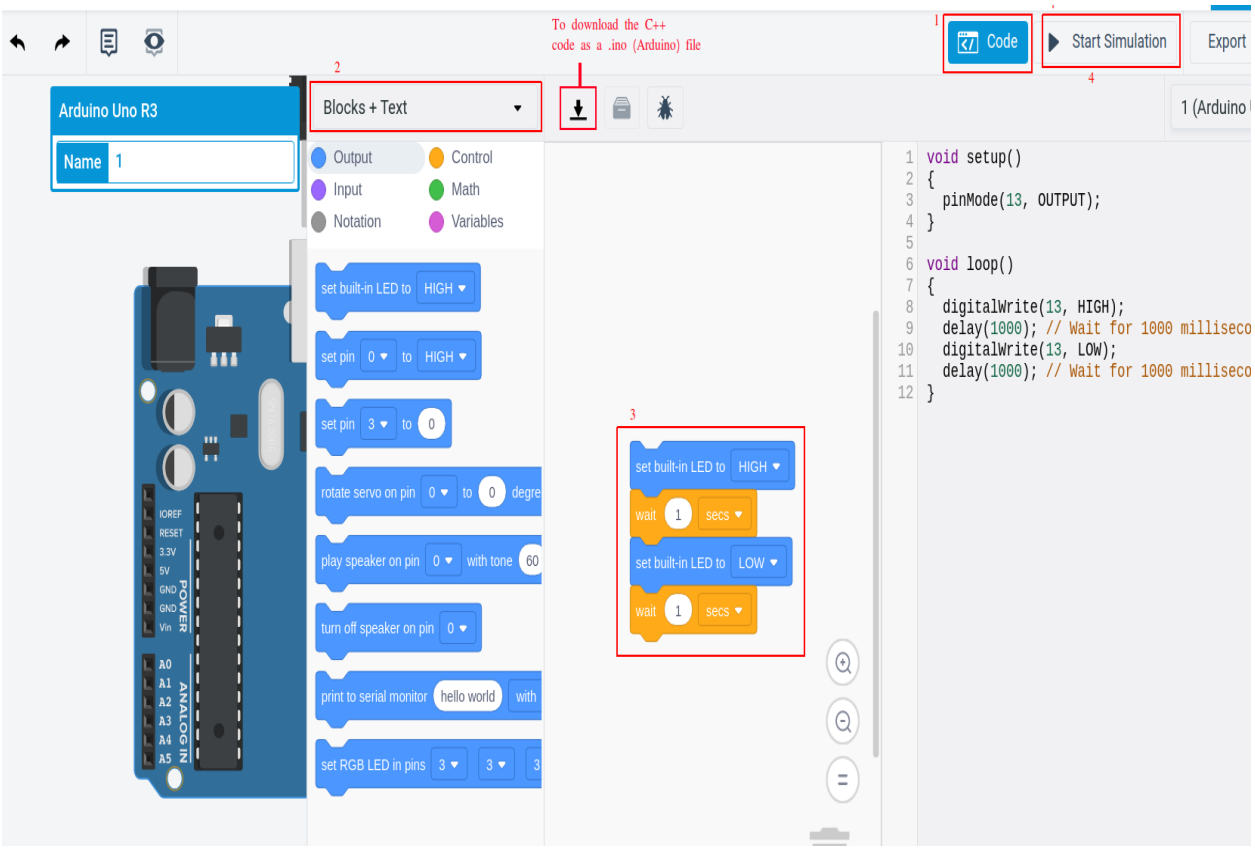
1. Open the WOKWI or TinkerCad simulation platform in your web browser and create a new project.
2. Select the components needed for LCD interfacing, including an Arduino board, LCD module, and necessary resistors and wires. These components should be available in the platform's component library.
3. Place the Arduino board on the workspace and connect it to the power and ground rails.
4. Connect the LCD module to the Arduino board. Typically, the LCD module will have 16 pins. Connect the appropriate pins of the LCD module to the corresponding pins of the Arduino board. Refer to the datasheet or documentation of the LCD module for the pinout details.
5. Add the necessary resistors to the circuit. The LCD module usually requires a potentiometer or a series of resistors to control the contrast of the display. Connect these components according to the LCD module's datasheet.
6. Use jumper wires to establish the connections between the Arduino board, LCD module, and resistors. Pay attention to the pin configurations and ensure that the connections are correct.
7. Write a program for the Arduino to control the LCD module. Depending on your requirements, you can display text, numbers, or custom characters on the LCD. The programming language used in Arduino is based on C/C++. If you're new to Arduino programming, you can find plenty of examples and tutorials online.
8. Upload the program to the Arduino board using the simulation platform's interface. This will allow you to test the LCD interfacing circuit virtually.
9. Run the simulation to see the output on the LCD module. Make sure the connections and code are correct. If you encounter any issues, debug your circuit and program accordingly.

Programming and Simulation

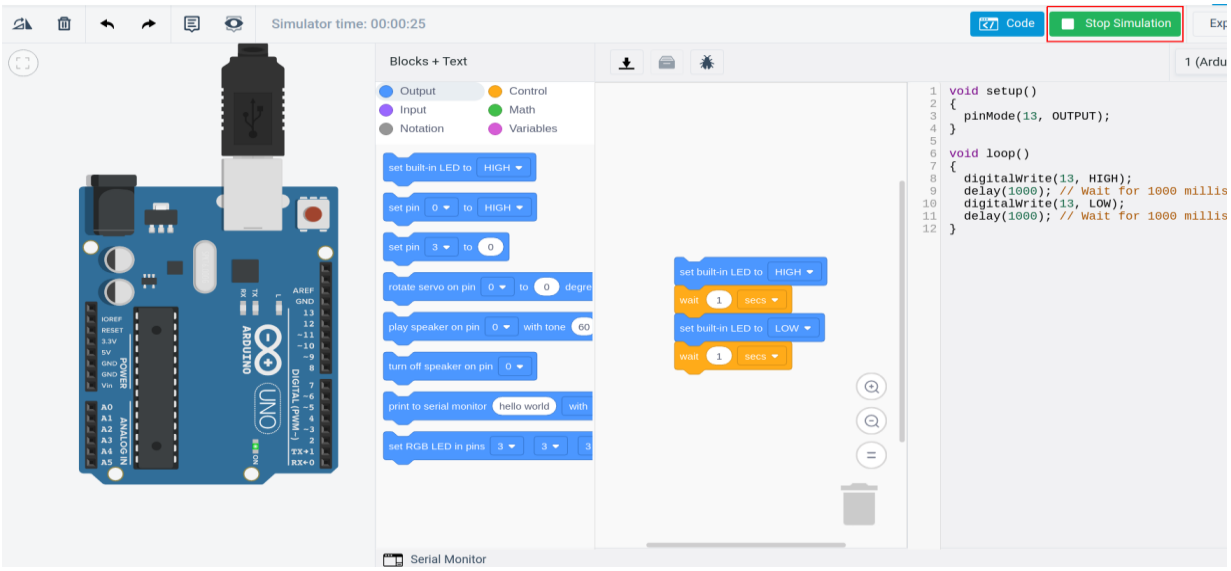
To program the Arduino,

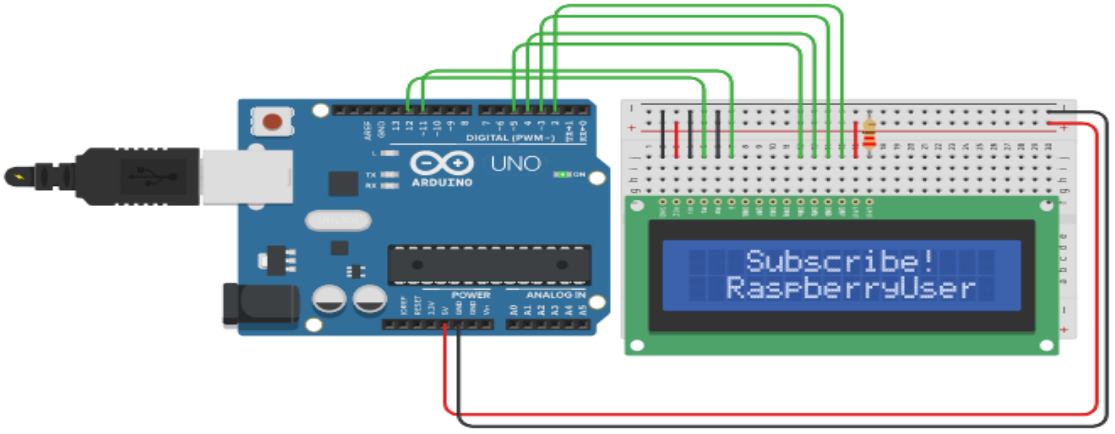
1. Click on Code
2. You can choose Blocks or Blocks+Text or Text*. For beginners, it is recommended to use Blocks + Text.
    a. This allows you to see the C++ code generated corresponding to your blocks.
    b. You can copy this code later into Arduino IDE to program the real Arduino, rather than having to write it from scratch.
    c. You can also download the code as an Arduino-compatible .ino file.
3. You can code by selecting the blocks and connecting them appropriately.
4. You can start the simulation by clicking Start Simulation.

*Note: You can go between Blocks and Blocks+Text anytime. You can go from Blocks / Blocks+Text to Text, but you can't go back from Text to either of the other two (converting blocks to text is easy, converting text to blocks is computationally non-trivial).



You have to click Stop Simulation to stop the simulation before you can modify your program and/or hardware connections.





Code:

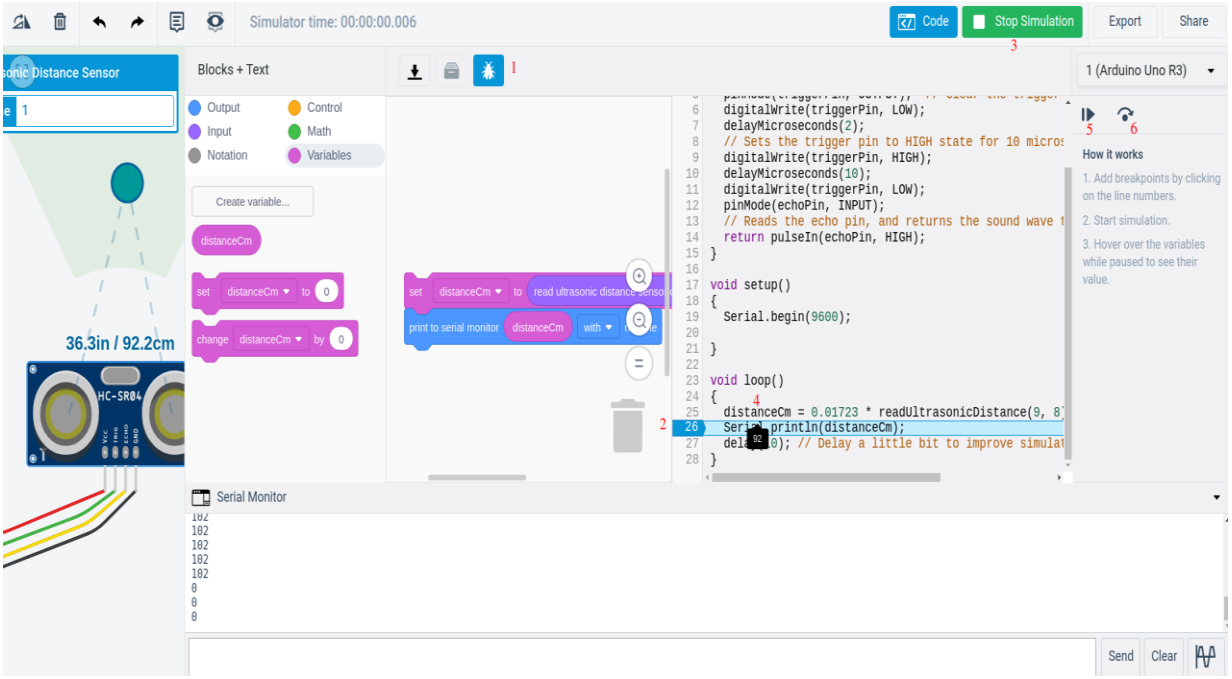
```
#include<LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup()
{
 lcd.begin(16, 2);
}

void loop()
{
  lcd.setCursor(0,0);
  lcd.print("   Subscribe!");
  lcd.setCursor(2,1);
  lcd.print("RaspberryUser");
  }
```

Debugging (Advanced)

More often than not, the code written by a programmer does not work as expected the very first time he/she runs it. We need to find out the logical flaws in our code and fix them before we are able to achieve full functionality. Figuring out flaws usually boil down to inspecting variable values at various points in our code, and comparing it with the expected values at those points based on the program logic and data inputs. The usual way Arduino programmers do it is by printing out the variable values to Serial console. Tinkercard allows for debugging without having printing the values you want to inspect through Serial. The example below shows debugging of the Ultrasonic Distance Sensor example.



1. Press the Debugger button.
2. Select the line(s) where you want the execution is to be paused. Such a line where you wish to pause execution is called a *breakpoint*.
3. Click Start Simulation.
4. Hover over the variable values you want to inspect, and determine if the values are along the expected lines. If not, there is something wrong, and use your logic to determine what could be wrong.
5. You can press the Resume execution button to run until the next breakpoint.
6. You can also step line by line by clicking the Step Over Next Function button.

Program:

```
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  // Set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print Hello World! to the LCD.
  lcd.print("Hello World!");
}

void loop() {
  // The location of line one is 0
  // The location of line two is 1
  // set the cursor to column 0, line 1 the print the second line
  lcd.setCursor(0, 1);
  lcd.print("Seconds ");
  lcd.print(millis() / 1000);
}
```

Viva Voce:

1. Explain how might Internet Address (IPv6) affect the development and implementation of the Internet of Things.
2. List the applications of IoT in Law enforcement.
3. Define IoT GE-PREDIX.
4. Should the customers be worried about security and protection issues considering the measure of information Internet of Things (IoT) gathers?
5. Explain what might be the effect of the Internet of Things hands on market, do you see any conceivable activity cuts.
6. Discuss distinctive parts where the Internet of Things can really enhance the present procedures.
7. Discuss the impacts of Internet of Things (IoT) have on the Energy Sector.
8. Deliberate the impacts of Internet of Things (IoT) have on the Agriculture Sector.
9. State the term Thingful mean in IoT.
10. Define GPIO.
11. Define the network's importance in IoT.
12. Explain by the term Internet of Things Gateway.
13. Distinguish between IoT and IIoT.
14. Name the various CAN Frames in IoT.
15. Define Thermocouple sensor.
16. Mention suitable databases for IoT.
17. List the importance of the Internet of Everything.
18. Explain how to run Raspberry pi in headless mode.
19. Discuss the available models in Raspberry Pi.
20. Discuss the impacts will the Internet of Things (IoT) have on the Transportation Sector.
21. Name the programming language supported by MongoDB.
22. Can Node MCU act as a web server.
23. How does IoT influence the development of smart cities.
24. What impacts will the Internet of Things (IoT) have on the Health Care Sector?
25. Explain IoT Thingworx.
26. List the dangers and difficulties that we ought to know about with regards to the Internet of Everything?
27. Explain the various IoT communication models.
28. List out the various IoT Protocol layers stack.
29. List the applications of IoT.
30. Explain Industry 4.0

Experiment 3

Aim: To Develop a smart traffic light management system with the help of IoT.

Objectives:

1. Learn about interfacing.
2. Learn about IoT programming.

Hardware Required:

- 1 × Breadboard
- 1 × Arduino Uno R3
- 3 × LEDs (Red, Yellow, Green)
- 3 × 220Ω Resistor
- 3 × Jumper

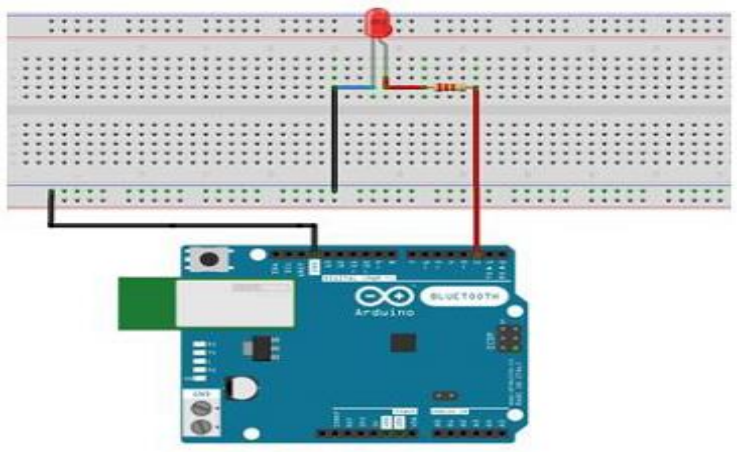
Introduction:

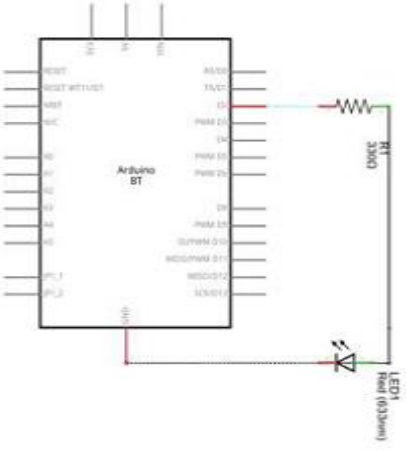
Nowadays, everyone prefers a personal vehicle. Hence, the number of vehicles on the road is increasing continuously, which results in traffic jams. Traffic light controller helps to manage the traffic and to maintain proper traffic management. These systems are placed at the intersections of the road or at the crossings to avoid congestions and accidents. The systems indicate to the driver by using different colors of light. Therefore it is simple to avoid congestion at the intersections.

Procedure

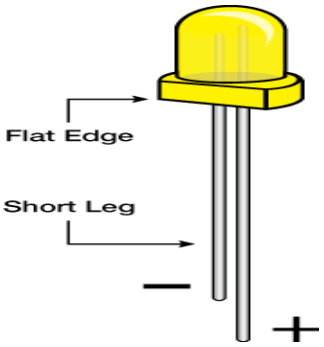
LEDs are small, powerful lights that are used in many different applications. To start, we will work on blinking an LED, the Hello World of microcontrollers. It is as simple as turning a light on and off. Establishing this important baseline will give you a solid foundation as we work towards experiments that are more complex.

Follow the circuit diagram and hook up the components on the breadboard as shown in the image given below.

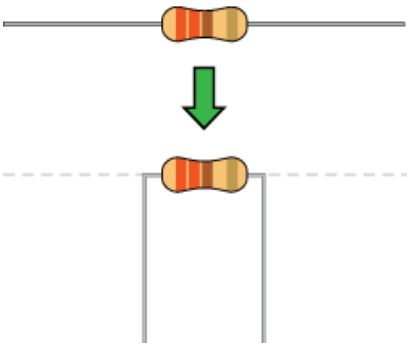




Note − To find out the polarity of an LED, look at it closely. The shorter of the two legs, towards the flat edge of the bulb indicates the negative terminal.



Components like resistors need to have their terminals bent into 90° angles in order to fit the breadboard sockets properly. You can also cut the terminals shorter.



Roads without any supervision or guidance can lead to traffic conflicts and accidents. Traffic signals are required for an orderly flow of traffic. A traffic signal is used as an instructing device that indicates the road user to act as per the displayed sign.
Traffic lights allow everyone to cross the intersection point one by one, reducing conflicts between vehicles entering intersection points from different directions. It provides road safety, also helps to solve traffic in simple manners.

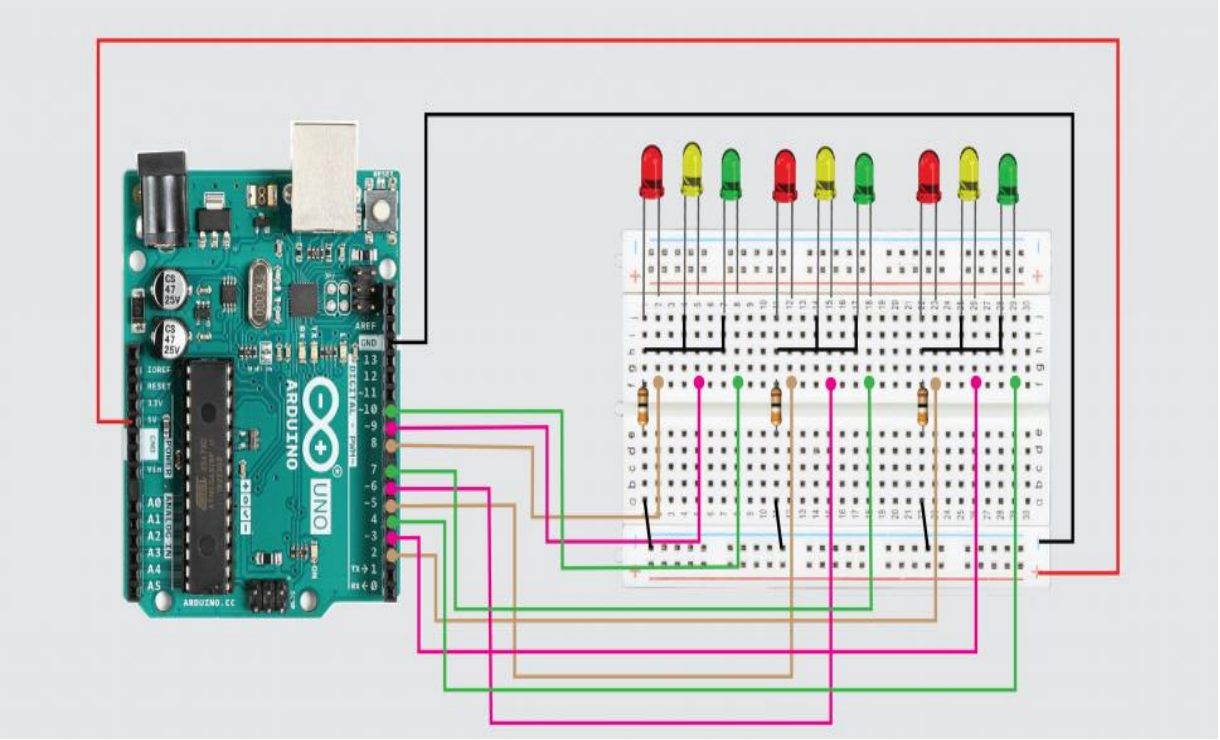
There are different colors in traffic lights. Each light has a meaning, and these lights tell drivers what to do.

- Red light ON- A driver should stop.
- Yellow light ON- A driver has to slow down and be ready to stop.
- Green light ON- A driver can start driving or keep driving.

Connections:

This is the circuit diagram for the traffic light controller by using Arduino.

- Connect LEDs on the breadboard as Red, Yellow, Green, respectively.
- Connect the negative terminal of the LED and connect the 220 Ohm resistor in series.
- Connect these negative terminals to the ground.
- Connect the positive terminal of the LEDs to the pins 2 to 10, respectively.
- Power the breadboard by using 5V and GND on the Arduino.

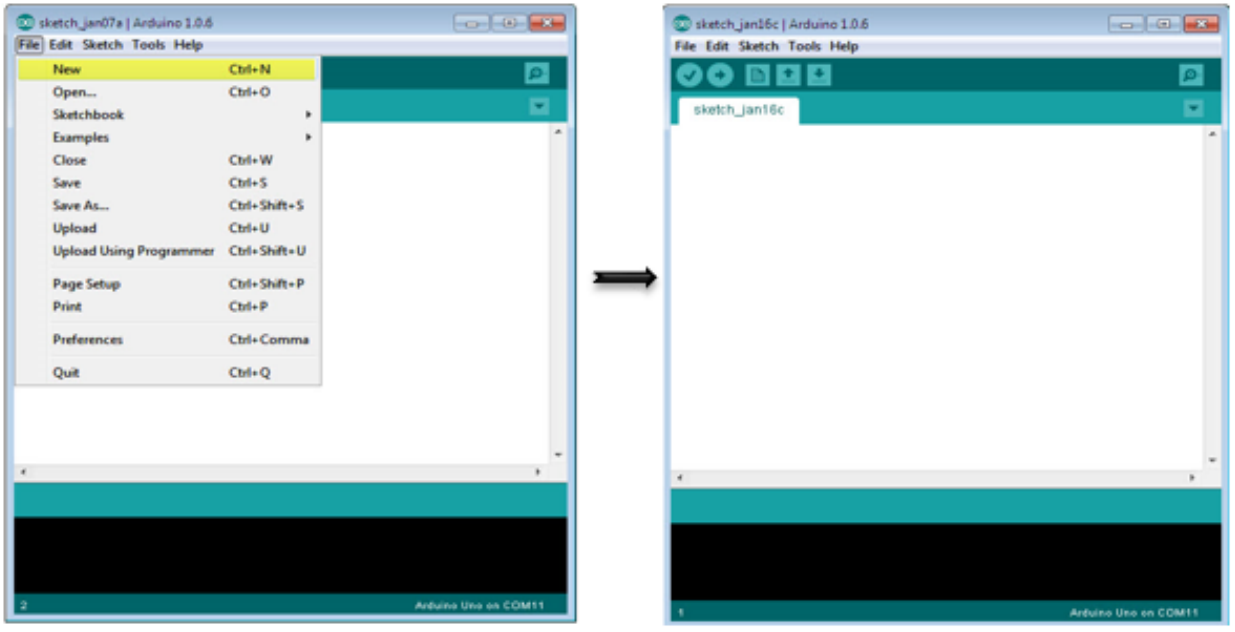


The logic of This Traffic Light System Using Arduino Code

The code for this traffic light controller is simple and easy to understand. We have exhibited traffic lights for a three-way road. The LEDs will be glowing in a particular sequence to form an actual traffic light controller system.

At a time, two red LEDs will glow, and one Green LED will be ON. Also, two yellow LEDs will be ON for one second. Yellow LED will Glow on each transition between RED to GREEN. In short first RED LED will glow for 5 Seconds, then YELLOW for 1 second, and then Green LED will ON for 5 seconds.

In the void setup of the code, we have defined the pins for the LEDs as outputs from 2 to 10. In the void loop section, we have defined the functions to turn LEDs ON and OFF into the sequence.



Code

```
void setup() {
// configure the output pins
pinMode(2,OUTPUT);
pinMode(3,OUTPUT);
pinMode(4,OUTPUT);
pinMode(5,OUTPUT);
pinMode(6,OUTPUT);
pinMode(7,OUTPUT);
pinMode(8,OUTPUT);
pinMode(9,OUTPUT);
pinMode(10,OUTPUT);
} void loop()
{
digitalWrite(2,1); //enables the 1st set of signals
digitalWrite(7,1);
digitalWrite(10,1);
digitalWrite(4,0);
digitalWrite(3,0);
digitalWrite(6,0);
digitalWrite(8,0);
digitalWrite(9,0);
digitalWrite(5,0);
delay(5000);
digitalWrite(3,1); //enables the yellow lights
digitalWrite(6,1);
digitalWrite(2,0);
digitalWrite(7,0);
delay(1000);
digitalWrite(4,1); //enables the 2nd set of signals
digitalWrite(5,1);
digitalWrite(10,1);
digitalWrite(2,0);
digitalWrite(3,0);
digitalWrite(6,0);
digitalWrite(8,0);
digitalWrite(9,0);
digitalWrite(7,0);
delay(5000);
digitalWrite(9,1); //enables the yellow lights
digitalWrite(6,1);
digitalWrite(10,0);
digitalWrite(5,0);
digitalWrite(4,0);
delay(1000);
digitalWrite(8,1); //enables the 3rd set of signals
digitalWrite(4,1);
digitalWrite(7,1);
digitalWrite(2,0);
digitalWrite(3,0);
digitalWrite(5,0);
digitalWrite(6,0);
digitalWrite(9,0);
```

```
digitalWrite(10,0);
delay(5000);
digitalWrite(9,1); //enables the yellow lights
digitalWrite(3,1);
digitalWrite(7,0);
digitalWrite(8,0);
digitalWrite(4,0);
delay(1000);
}
```

Result

You should see your LED turn on and off. If the required output is not seen, make sure you have assembled the circuit correctly, and verified and uploaded the code to your board. This traffic light controller includes a crosswalk signaling system. The traffic light controller in this system can be implemented practically, and it could be expanded.

Viva Voce:

1. Name the latest Raspberry Pi release.
2. List the wireless communications boards present in Raspberry Pi.
3. List applications of PWM in IoT.
4. List sensor and actuator are used to control home appliances from any IoT devices in wired mode.
5. Give the examples of the MEMS sensor.
6. Explain the IoT Cloud.
7. Define OSI Model in IoT.
8. Discuss ZigBee protocol.
9. List the sensors used in Agriculture.
10. Discuss about GainSpan's GS2000 Protocol for the Internet of Things (IoT).
11. Answer the abbreviation of MQTT.
12. How many GPIO pins are there in the latest Raspberry Pi.
13. List a few operating systems, which support Raspberry Pi.
14. Explain BLE in the Internet of Things.
15. Define library in Arduino in IoT.
16. Explain IoT GE Predix.
17. Explain how the Internet of Things Impact Our Daily Lives?
18. Define Big Data and how it is related to IoT.
19. Explain the role of publishers in IoT.
20. Discuss the commonly used protocols in IoT.
21. Explain IoT Contiki.
22. List the Industries that are using IoT currently.
23. Explain a live case example of IoT.
24. Discuss Bluegiga APX4 convention for the Internet of Things (IoT)?
25. Explain how can I run a Raspberry Pi without a display.
26. Name the founder of IoT.
27. List the sensors used in robotics.
28. Define IIoT.
29. Define Alexa.
30. Explain the LoRa in IoT.

Experiment 4

Aim: To Formulate distance of an object using an ultrasonic sensor.

Hardware Required:

1. Arduino Uno R3 board
2. Ultrasonic sensor (HC-SR04)
3. 16×2 LCD I2C Display
4. Jumper Wires

Arduino:
It is an open-source electronics platform. It consists ATmega328 8-bit Micro controller. It can be able to read inputs from different sensors & we can send instructions to the micro controller in the Arduino. It provides Arduino IDE to write code & connect the hardware devices like Arduino boards & sensors.

Ultrasonic Sensor:
An ultrasonic Sensor is a device used to measure the distance between the sensor and an object without physical contact. This device works based on time-to-distance conversion.

Working Principle of Ultrasonic Sensor:
Ultrasonic sensors measure distance by sending and receiving the ultrasonic wave. The ultrasonic sensor has a sender to emit the ultrasonic waves and a receiver to receive the ultrasonic waves. The transmitted ultrasonic wave travels through the air and is reflected by hitting the Object. Arduino calculates the time taken by the ultrasonic pulse wave to reach the receiver from the sender.

We know that the speed of sound in air is nearly 344 m/s,

So, the known parameters are time and speed (constant). Using these parameters, we can calculate the distance traveled by the sound wave.

Formula: Distance = Speed * Time

In the code, the "duration" variable stores the time taken by the sound wave traveling from the emitter to the receiver. That is double the time to reach the object, whereas the sensor returns the total time including sender to object and object to receiver. Then, the time taken to reach the object is half of the time taken to reach the receiver.

so we can write the expression as,

Distance = Speed of Sound in Air * (Time Taken / 2)

Note: Speed of sound in air = 344 m/s.

Circuit Diagram:

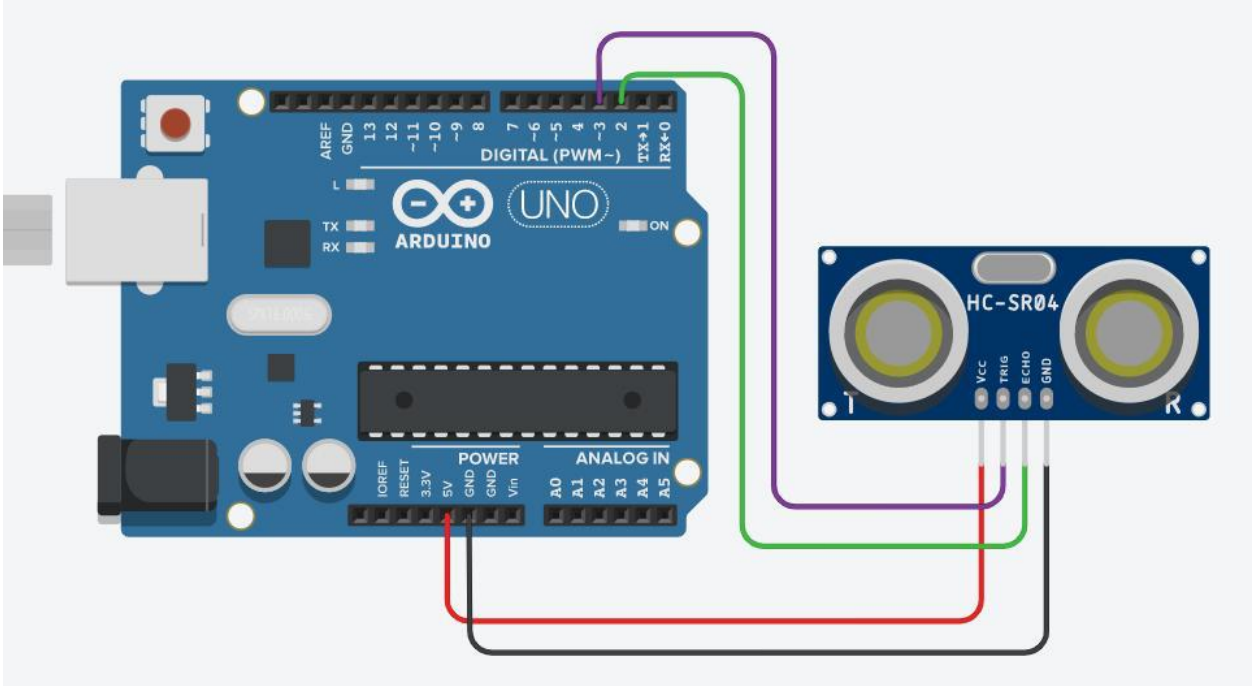


Figure: Ultrasonic Distance measurement circuit

Setup:

1. Connect the Echo pin of the sensor to the D2 pin of the Arduino.
2. Connect the Trig pin of the sensor to the D3 pin of the Arduino.
3. Navigate to Tools and select board and port.
4. Verify and compile the code, then upload the code to the Arduino Uno R3 board.
5. Monitor the output in the Serial monitor (Set the baud rate as 9600). To open Serial monitor Tools>Serial Monitor or (Ctrl+Shift+M).

Arduino Code (Output in Serial monitor):

The following code will show the output on the serial monitor of Arduino software with a baud rate of 9600.

```
#define echoPin                                 \
   2 // attach pin D2 Arduino to pin Echo of HC-SR04
#define trigPin                                  \
   3 // attach pin D3 Arduino to pin Trig of HC-SR04                                  \


long duration; // Variable to store time taken to the pulse
          // to reach receiver

int distance; // Variable to store distance calculated using
          // formula

void setup()
{
```

```
    pinMode(trigPin,
            OUTPUT); // Sets the trigPin as an OUTPUT
    pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT

    // Serial Communication is starting with 9600 of
    // baudrate speed
    Serial.begin(9600);

    // The text to be printed in serial monitor
    Serial.println(
       "Distance measurement using Arduino Uno.");
    delay(500);
}

void loop()
{
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2); // wait for 2 ms to avoid
                          // collision in serial monitor

    digitalWrite(
       trigPin,
       HIGH); // turn on the Trigger to generate pulse
    delayMicroseconds(
       10); // keep the trigger "ON" for 10 ms to generate
            // pulse for 10 ms.

    digitalWrite(trigPin,
                 LOW); // Turn off the pulse trigger to stop
                       // pulse generation

    // If pulse reached the receiver echoPin
    // become high Then pulseIn() returns the
    // time taken by the pulse to reach the
    // receiver

    duration = pulseIn(echoPin, HIGH);
    distance
       = duration * 0.0344 / 2; // Expression to calculate
                                // distance using time

    Serial.print("Distance: ");
    Serial.print(
       distance); // Print the output in serial monitor
    Serial.println(" cm");
    delay(100);
}
```

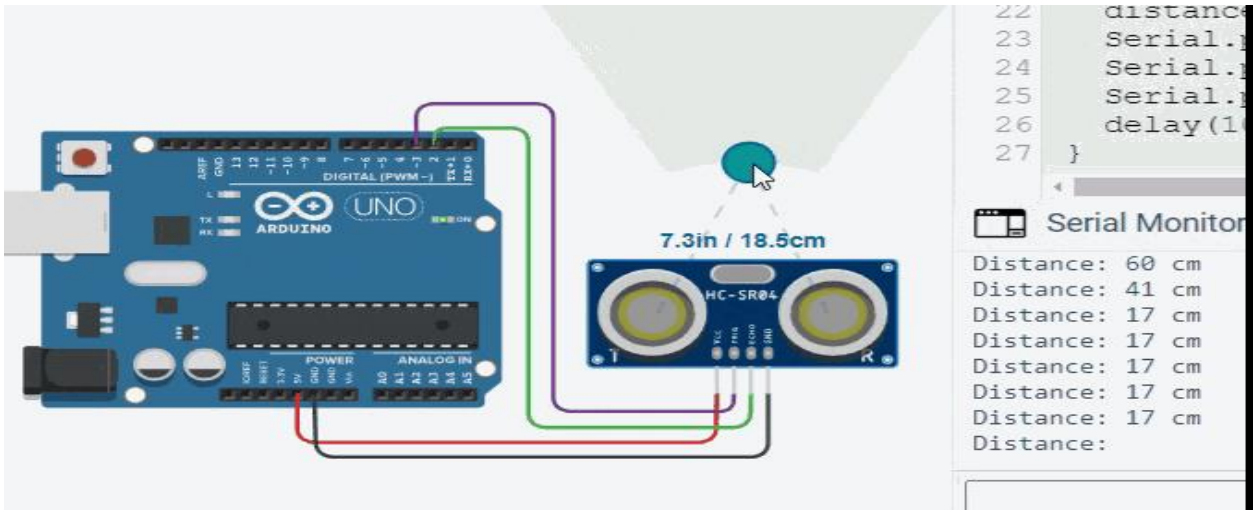


Figure: Distance measurement

Library Installation

1. Install driver library for Liquid Crystal Display.

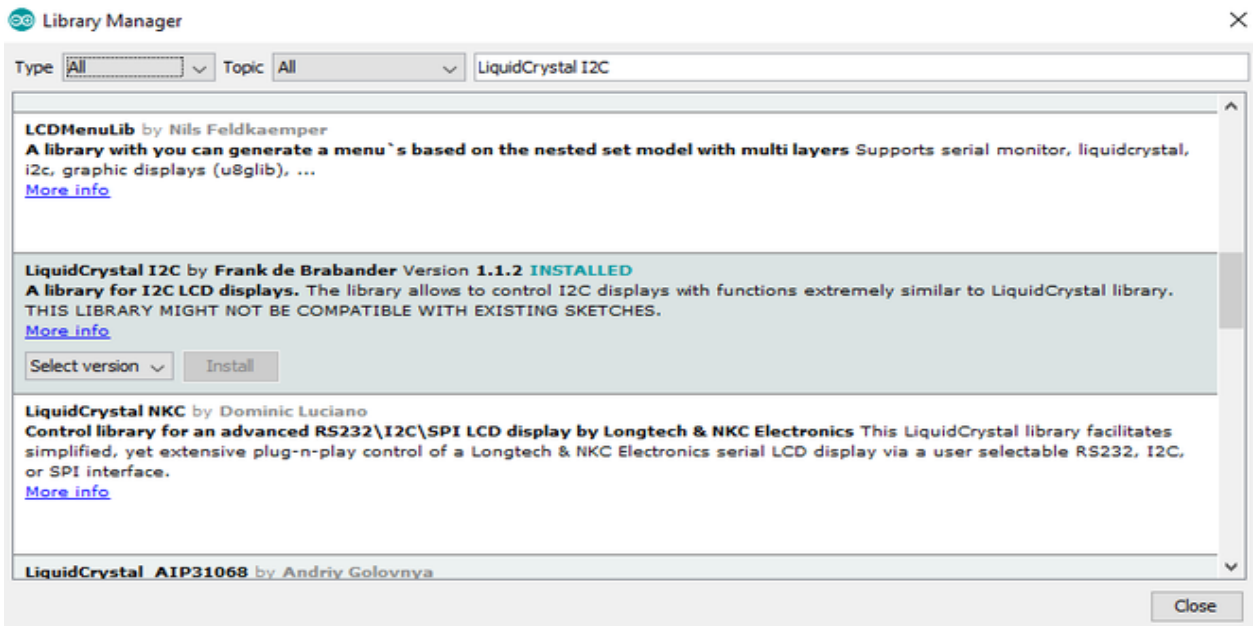
- Navigate **Tools>Library Manager** (or) Enter (**Ctrl+Shift+I**) to open library manager.
- Search for “LiquidCrystal I2C” and install the “LiquidCrystal I2C” library.

2. Import the header file “LiquidCrystal_I2C.h” in the code.

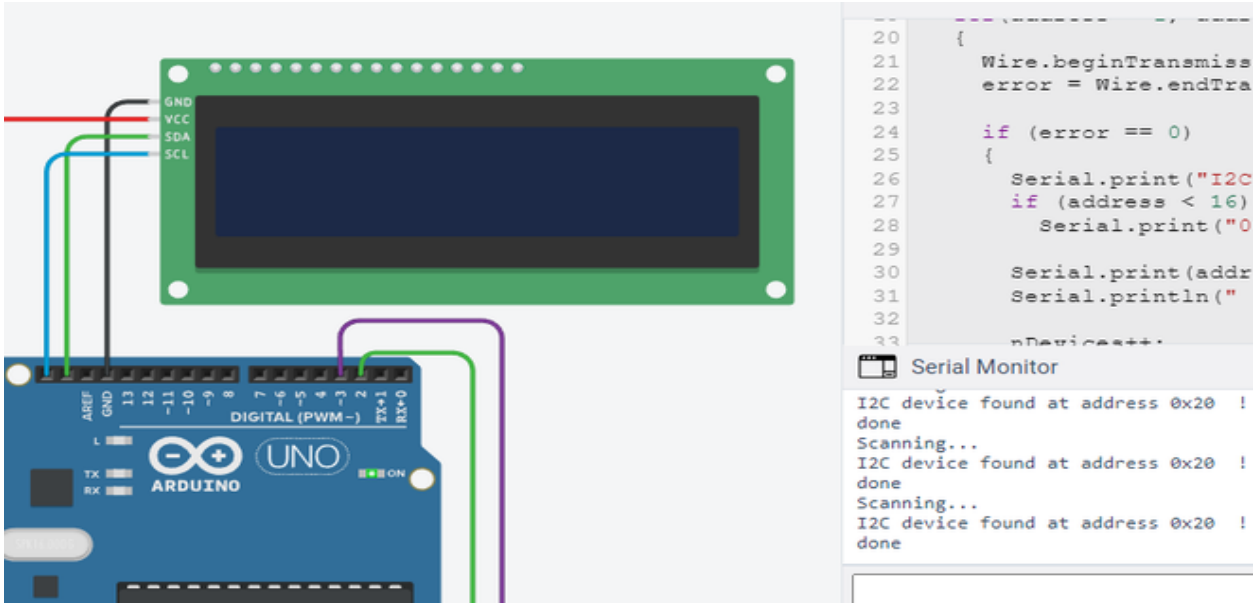
3. Connect the SDA pin of an LCD display to the SDA pin of the Arduino Board and the SCL pin of an LCD display to the SCL of the Arduino Board.

4. Connect VCC to 5V pin and GND to GND pin.

5. Include the below code to define the display device.



Serial monitor Output:

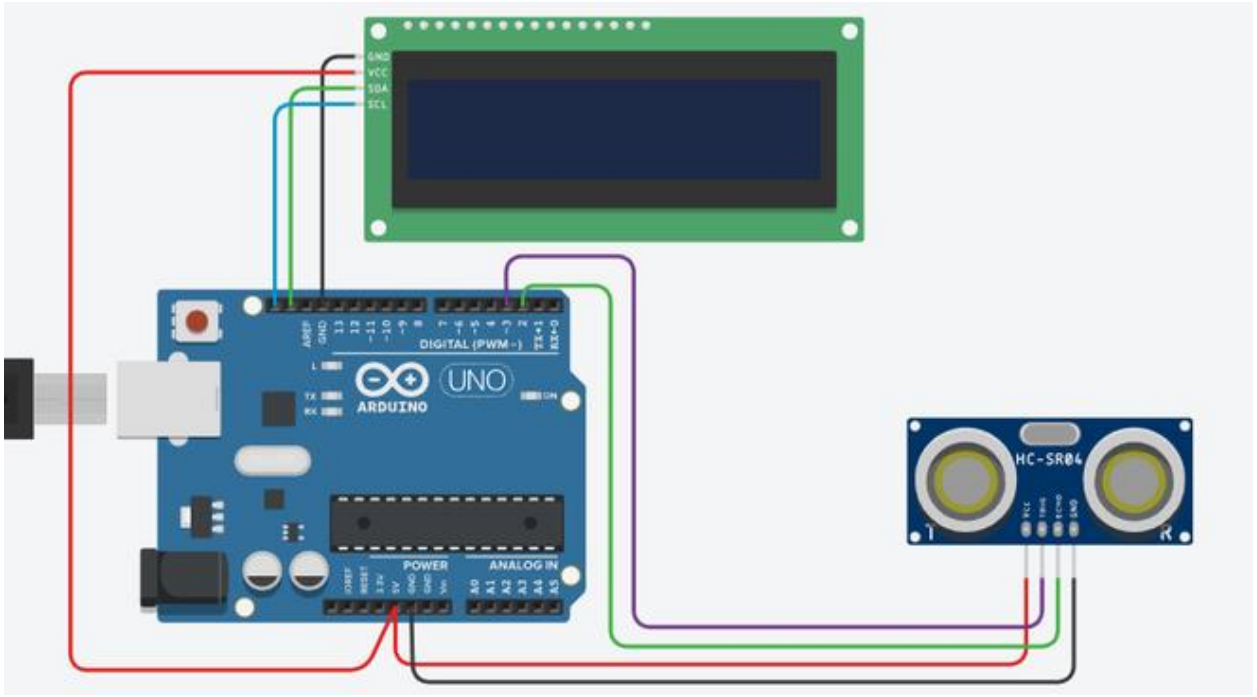


LCD Address Finding

Here I2C Address is “0x20”, So replace the address with “0x20” in the code.

(Ensure that you have connected the LCD display device as per the instruction while running this program in the Arduino).

Circuit Diagram with LCD display:



Applications of Ultrasonic Distance Measurement:

- Used in RADAR system.
- To measure distance without physical contact with measuring instruments.
- Used in object detection for security purposes.

Viva Voce:

1) What is Arduino? What are the features of Arduino Uno? Explain following functions of Arduino with example: analogRead(), analogWrite().
2) What is Raspberry Pi? What are its potential applications?
3) Explain interfacing of LED with Raspberry Pi.
4) Draw and explain Arduino architecture.
5) Draw the block diagram of Raspberry Pi and explain components of Raspberry Pi in brief.
6) Write features of latest version of Raspberry Pi.
7) Explain pinMode(), digitalRead() and digitalWrite () functions of Arduino.
8) Write Raspberry Pi code to change the brightness of LED.
9) What is PWM? Explain usage of PWM pins with Example.
10) Write Raspberry Pi code to blink LED ON and OFF .
11) Write arduino code to transmit "Hello World – Code to demonstrate BT Communication" string on serial monitor using Bluetooth.
12) Explain characteristics of the IoT. Explain characteristics of the IoT.
13) Define IoT & Explain various levels of IoT.
14) List challenges while opting for IoT & cloud computing. How can we overcome them?
15) List any five IoT applications. Explain any one in details.
16) Give difference between Wireless Sensor Network (WSN) & IoT
17) With a neat sketch, explain IoT stack with appropriate examples for each layer.
18) Explain IOT challenges.
19) What is M2M and IoT?
20) What do you mean by enabling technologies of IOT? Explain about each 1 of them in brief.
21) Define following terms: IoT, Sensor, WSN.
22) List down components of IoT system.
23) Explain IoT and cyber physical system.
24) Explain web of things.
25) Explain sensor and micro controller.
26) List down components of IoT system.
27) Difference between Micro controller and Microprocessor.
28) What is ARM? Explain special feature of ARM processor.
29) What is heartbeat and gas sensor?
30) Explain specification of sensor.

Experiment 5

Aim: To Design a weather station by checking Air quality of an environment with the help of IoT.

Hardware Required:

1. Arduino Uno R3

2. MQ 135 Air Quality Sensor Module

3. Male to Female Jumper Wire

4. Software: Arduino IDE

About Air Quality Sensor:

MQ-135 sensor belongs to the MQ series that are used to detect different gasses present in the air. The MQ-135 sensor is used to detect gases such as NH3,NOx, alcohol, Benzene, smoke,CO2 ,etc. steel exoskeleton houses a sensing device within the gas sensor module.

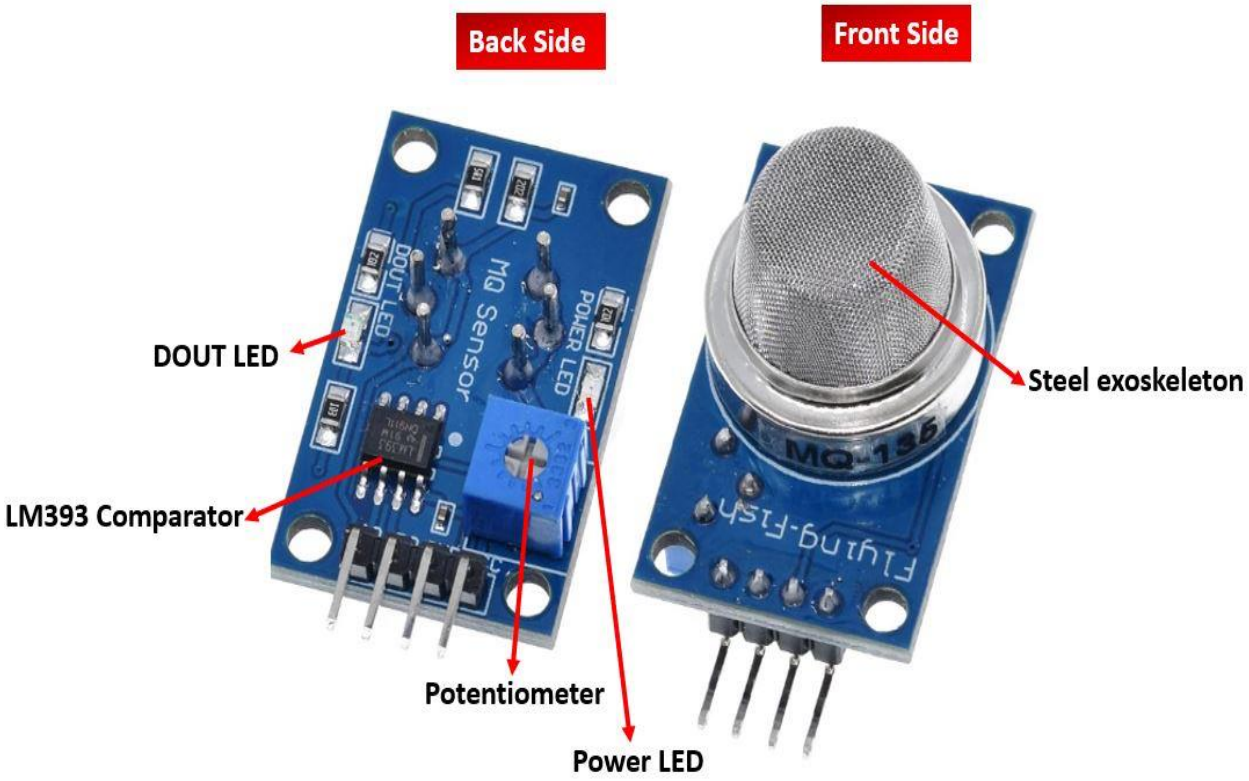


Figure: MQ 135 Sensor

Specifications

The table below shows some key specifications of the MQ-135 sensor module:

Feature	Description
Operating Voltage	2.5-5.0V
Detecting Concentration	10ppm-300ppm for NH3 10ppm-1000ppm for Benzene 10ppm-300ppm for Alcohol
Load Resistance	Adjustable
Heater Resistance	33Ω ± 5%
Heater Consumption	less than 800mW
Operating Temperature	-10 to 45°C

Pin out

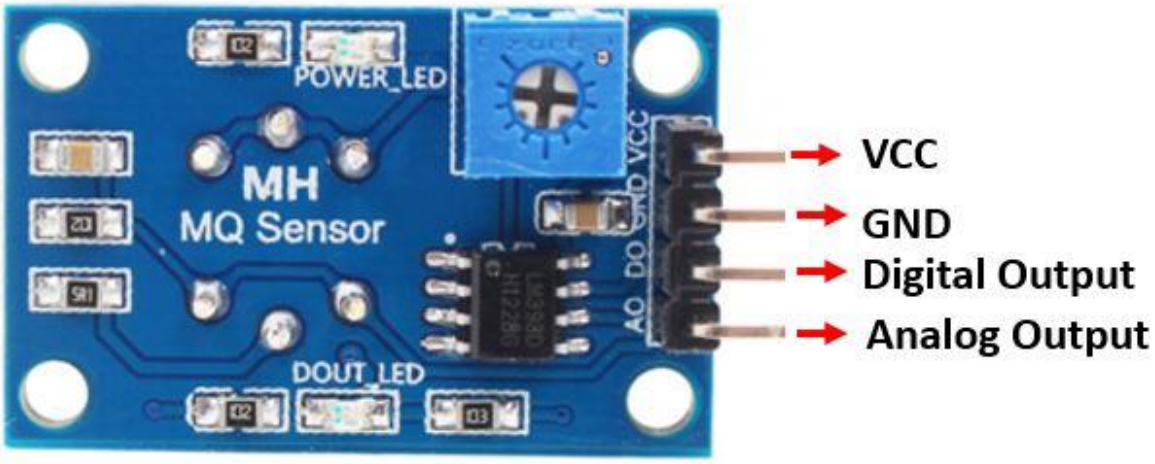


Figure: MQ-135 Sensor Pin out

This sensor has 4 pins:

- 5V: Module power supply – 5 V
- GND: Ground
- DOUT: Digital output
- AOUT: Analog output

Circuit

The following circuit shows how you should connect Arduino to MQ-135 module. Connect wires accordingly.

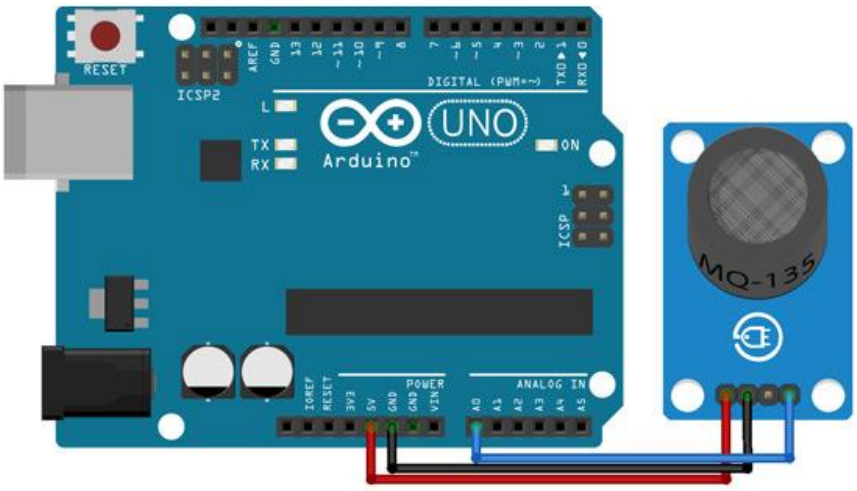


Figure: Connections of MQ135 with Arduino

The MQ-135 sensor module consists of four pins namely VCC, GND, DO, and DO. The table below gives a brief description of them.

Pin	Description
VCC	Positive power supply pin that powers up the sensor module.
GND	Reference potential pin.
AO	Analog output pin. It generates a signal proportional to the concentration of gas vapors coming in contact with the sensor.
DO	Digital Output pin. It also produces a digital signal whose limit can be set using the in-built potentiometer.

Interfacing MQ-135 Gas Sensor with Arduino

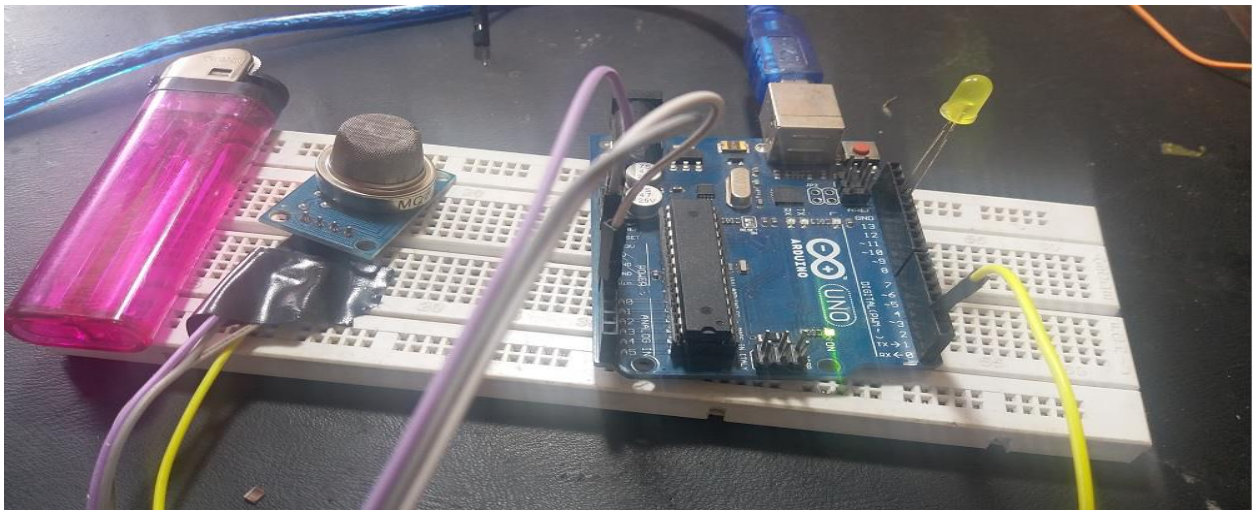


Figure: Interfacing of MQ135 with Arduino

The table below shows the connections you need to make between the MQ3 sensor module and Arduino using both the analog output and the digital output pins of the sensor.

MQ-135 Module	**Arduino**
VCC	5V
GND	GND
AO	A0
DO	Pin 2

Connect MQ-135 sensor's VCC pin with 5V terminal of Arduino UNO. This will power up the sensor. Additionally, we will connect the analog pin AO with A0 and DO with Pin 2 of Arduino UNO. Both the devices will be commonly grounded. Follow the connection diagram below, to connect your devices accordingly.

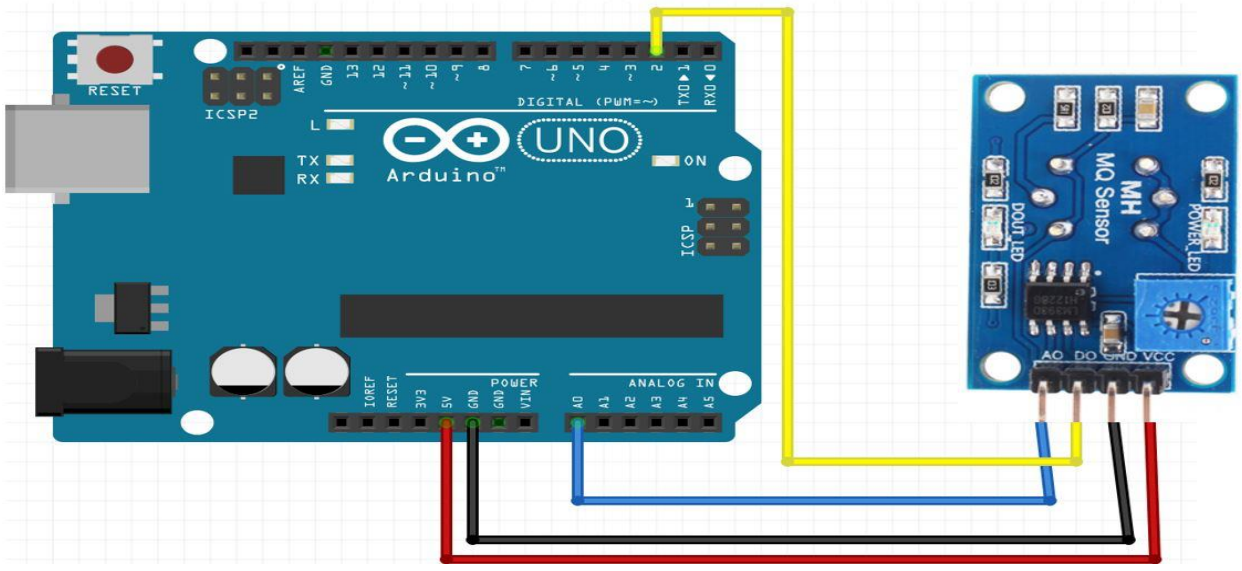


Figure: Arduino UNO with MQ-135 Module using both digital and analog outputs

MQ-135 Gas Detection Arduino Sketch

Open your Arduino IDE and go to **File > New**. Copy the code below in that file.

This sketch will read both the analog and digital outputs of the sensor. If the analog output is greater than 400 then an LED connected at Arduino pin 2 will turn ON. Otherwise, turn the LED OFF and print both the analog and digital output readings on the serial monitor.

```
int sensorValue;
int digitalValue;

void setup()
{
  Serial.begin(9600); // sets the serial port to 9600
  pinMode(13, OUTPUT);
  pinMode(2, INPUT);
}

void loop()
{
 sensorValue = analogRead(0); // read analog input pin 0
 digitalValue = digitalRead(2);
 if (sensorValue > 400)
{
 digitalWrite(13, HIGH);
}
else
   digitalWrite(13, LOW);
Serial.println(sensorValue, DEC); // prints the value read
Serial.println(digitalValue, DEC);
delay(1000); // wait 100ms for next reading
}
```

How the Code Works

Create two int variables to hold the analog and digital output readings.

```
int sensorValue;
int digitalValue;
```

Inside the setup() function, we will open the serial communication at a baud rate of 9600. Then configure pin2 connected with the DO pin of the sensor as an input and pin13 connected with the LED's anode pin as an output.

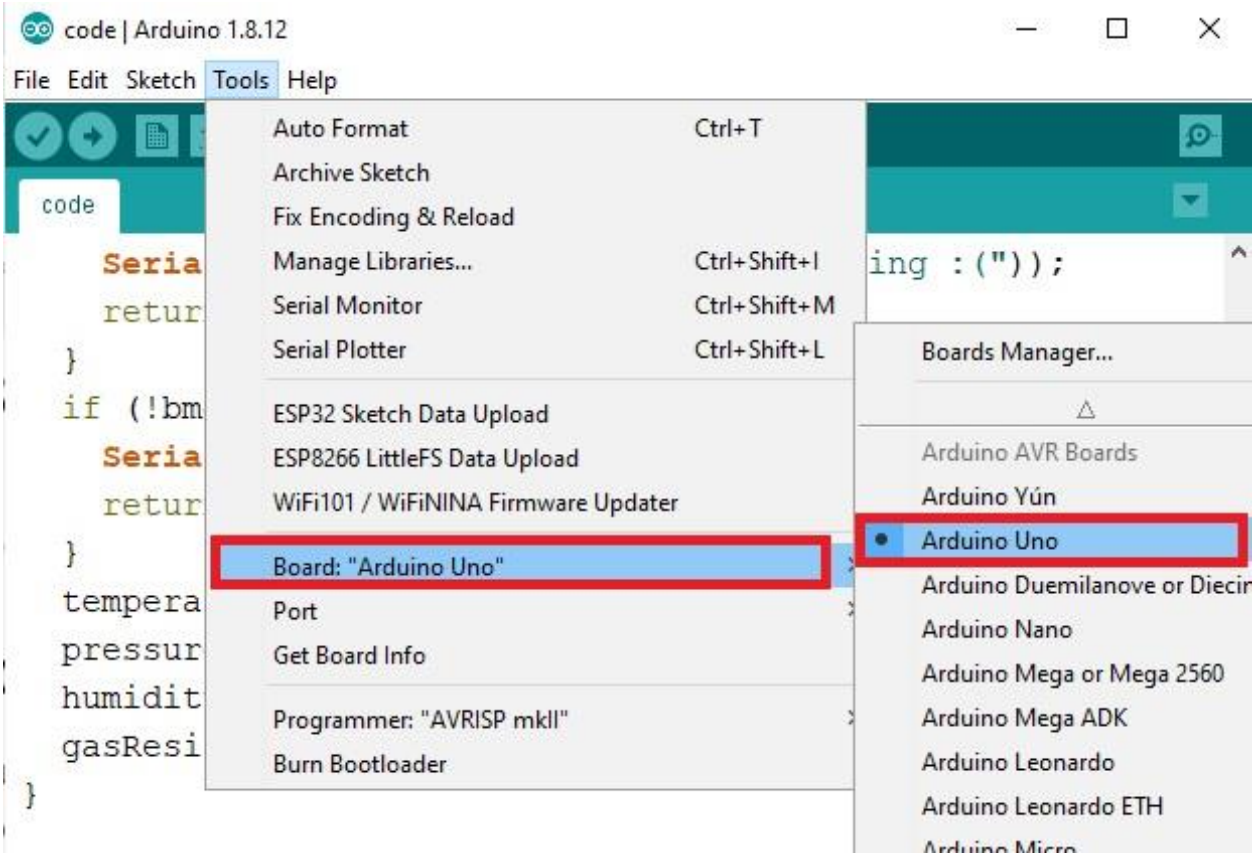
```
void setup()
{
 Serial.begin(9600); // sets the serial port to 9600
 pinMode(13, OUTPUT);
 pinMode(2, INPUT);
}
```

In the infinite loop(), we will use analogRead() on the A0 pin and save the value in ‘sensorValue.’ Likewise, we will read the digital output on pin2 using digitalRead() and save the value in ‘digitalValue.’ Next, using an if-else statement we will check if the analog reading is greater than 400 or not. If it is then turn the LED ON. Otherwise, leave the LED OFF and display the current analog and digital readings in the serial monitor.

```
void loop()
{
 sensorValue = analogRead(0); // read analog input pin 0
 digitalValue = digitalRead(2);
 if (sensorValue > 400)
{
 digitalWrite(13, HIGH);
 }
 else
 digitalWrite(13, LOW);
 Serial.println(sensorValue, DEC); // prints the value read
 Serial.println(digitalValue, DEC); delay(1000); // wait 100ms for next reading
}
```

Demonstration

To see the demonstration of the above code, upload the code to Arduino. But, before uploading code, make sure to select the Arduino board from Tools > Board and also select the correct COM port to which the Arduino board is connected from Tools > Port.



On the serial monitor, you can see the values of the analog pin being detected. Currently, in my case, they are around about 150, which indicates normal air.

- Normal air returns approximately 100-150
- Alcohol returns approximately 700
- Lighter gas returns approximately 750

Viva Voce:

1. What are mostly used IoT protocols.
2. Explain IoT sensor types.
3. Explain IoT actuators types.
4. Difference between Sensors and Actuators.
5. Which sensor is LM35.
6. How many pins does temperature sensor have?
7. Which sensor is used for monitor electric distribution and switching equipment temperature?
8. Electric motor protection has which sensor?
9. Which type of temperature sensor is placed in Integrated Circuits?
10. Which sensor is linear and low accuracy?
11. What is the use of the GSR Sensor?
12. What is the use of the MQ2 Gas Sensor?
13. What will happen if we supply a voltage of 25V to the Vcc of the MQ2 Gas Sensor?
14. If the PPM concentration of a gas that is being detected by the MQ2 Gas Sensor increases what will happen to its Analog Output pin?
15. What is the name of the mesh surrounding the MQ2 Gas Sensor?
16. What is the use of the LDR Sensor?
17. What is the full form of the LDR Sensor?
18. How many pins are present in the LDR Sensor?
19. What kind of input does the DC pin accept?
20. What is the use of the Nokia 5110 GDM?
21. What will happen if we start using the Nokia 5110 GDM without resetting?
22. What is the use of the OV7670 Camera Module?
23. How many pins are present in the OV7670 Camera Module?
24. What is the resolution of the OV7670 Camera Module?
25. What is the type of waves that the ESP8266 WiFi Module detects?
26. What is the role of the MISO pin in the RFID Module?
27. What is the use of the RFID Module?
28. Which frequency does the RFID Module operate in?
29. What is the maximum read range of the RFID Module?
30. What is the use of the ESP8266 WiFi Module?
31. What kind of device is the ESP8266 WiFi Module?
32. What is the type of waves that the ESP8266 WiFi Module detects?

Experiment 6

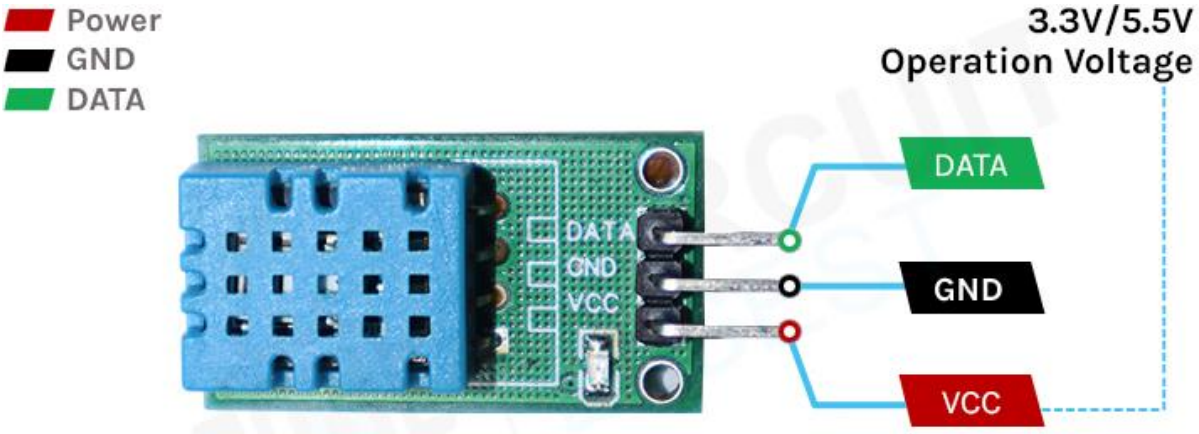
Aim: To Investigate real-time relationship between humidity and temperature in IoT.

Hardware Required:

- Arduino Board
- Breadboard
- Jumper Wires
- DH11 Temperature and Humidity Sensor

About DH11 Sensor:
DHT11 Module features a temperature & humidity sensor complex with a calibrated digital signal output. The exclusive digital-signal-acquisition technique and temperature & humidity sensing technology ensure high reliability and excellent long-term stability. This sensor includes an NTC for temperature measurement and a resistive-type humidity measurement component for humidity measurement. These are connected to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability, and cost-effectiveness.

DHT11 Module Pin out
The DHT11 module has a total of 3 pins. In which two are for power and one is for communication. The pin out of a DHT11 Sensor module is as follows:

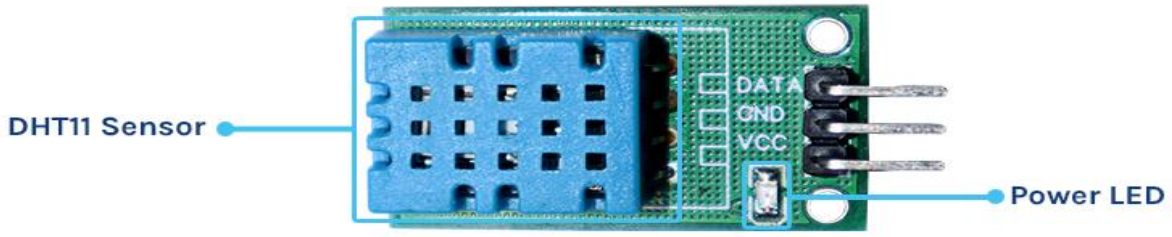


DATA Data pin for 1-wire communication.

GND Ground Connected to Ground pin of the Arduino.

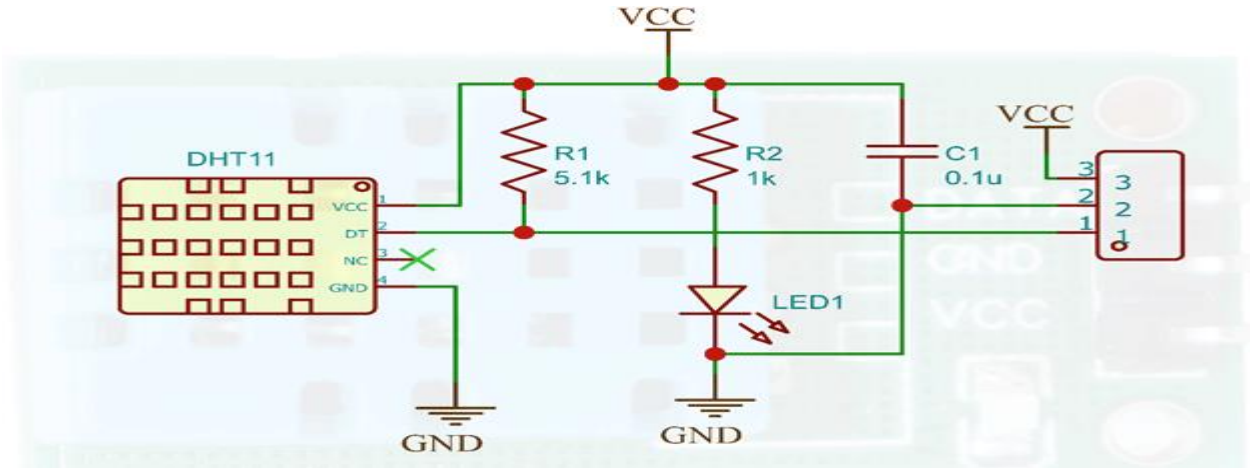
VCC Provides power for the module, Connect to the 5V pin of the Arduino.

DHT11 Module Parts
The DHT11 module has only a very low number of parts that includes the DHT11, pull-up resistor, bypass capacitor, and power led with a current limiting resistor.



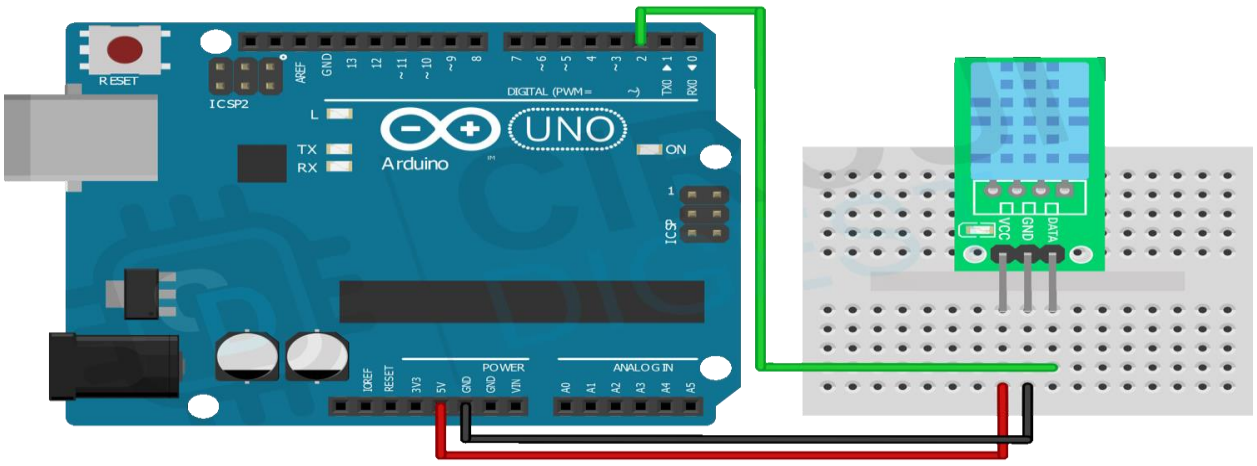
DHT11 Module Circuit Diagram

The schematic diagram for the DHT11 module is given below. As mentioned earlier, the board has a very low components count. The VCC and GND are directly connected to the DHT11 and a pull-up resistor is added to the DATA pin. Sufficient filtering is provided with the tantalum and multilayer capacitors. An LED with a current limit resistor is used as a power indicator.

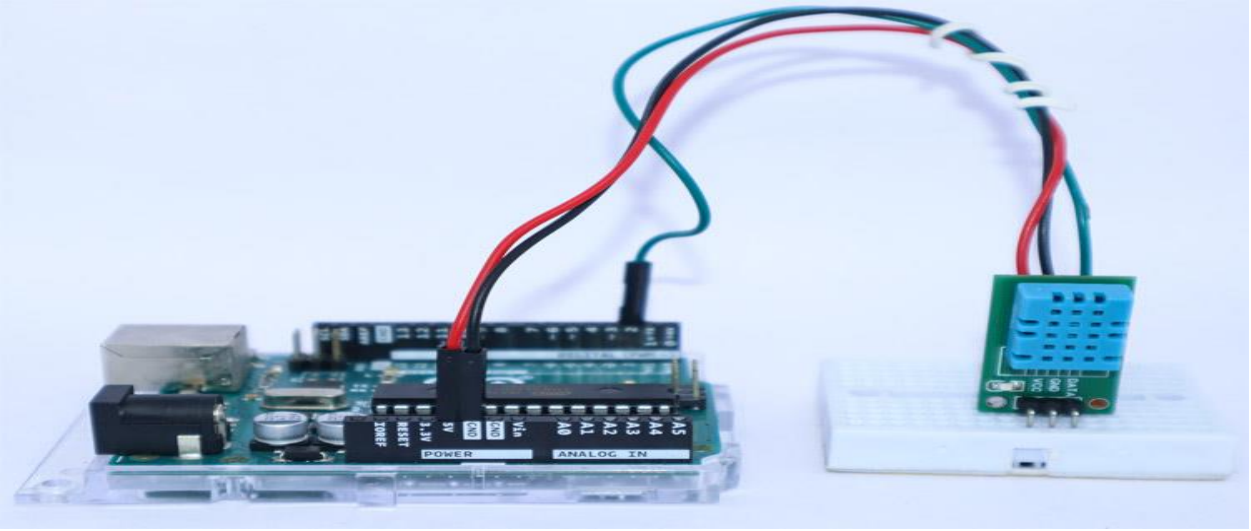


Circuit Diagram for Interfacing DHT11 Sensor with Arduino

Now that we have completely understood how a DHT11 Sensor works, we can connect all the required wires to Arduino and write the code to get all the data out from the sensor. The following image shows the circuit diagram for interfacing the DHT11 sensor module with Arduino.



Connections are pretty simple and only require three wires. Connect the VCC and GND of the module to the 5V and GND pins of the Arduino. Then connect the DATA pin to the Arduino's digital pin 2. We communicate with DHT11 through this pin.



Arduino DHT11 Code for Interfacing the Sensor Module

Now let's look at the code for interfacing the DHT11 sensor. For that first install the Adafruit's DHT sensor library and Adafruit Unified Sensor Driver through the library manager. Then create a blank sketch and paste the code at the end of this article into it.

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#define DHTTYPE    DHT11     // DHT 11 #define DHTPIN 2
DHT_Unified dht(DHTPIN, DHTTYPE);
uint32_t delayMS;
```

In the starting, we have included all the necessary libraries and defined the sensor type as DHT11 and the sensor pin as digital pin 2. And then created an instance for the DHT library. Also created a variable to declare the minimum delay.

```
void setup() {
 Serial.begin(9600);
 dht.begin();
 sensor_t sensor;
 delayMS = sensor.min_delay / 1000;
}
```

In the **setup** function, we have initialized serial communication and the DHT library. Then the minimum delay for the data refresh.

```
void loop()
{
 sensors_event_t event;
 dht.temperature().getEvent(&event);
 Serial.print(F("Temperature: "));
 Serial.print(event.temperature);
 Serial.println(F("°C"));
 dht.humidity().getEvent(&event);
```

```
 Serial.print(F("Humidity: "));
 Serial.print(event.relative_humidity);
 Serial.println(F("%"));
 delay(delayMS);
}
```

In the **loop** function, we have created an event and using this event the temperature and humidity data is read from the DHT11 sensor. Then this value is printed to the serial monitor.

Result
You will see the temperature and humidity display on serial port monitor which is updated every 2 seconds. Analyze their relation.

Viva Voce:

1. What is ESP8266?
2. Which sensor is LM35?
3. How many pins does temperature sensor have?
4. Monnit temperature sensor is used for what?
5. Name any Wireless battery-free sensor?
6. Which sensor is used for monitor electric distribution and switching equipment temperature?
7. Electric motor protection has which sensor?
8. What is DS18B20?
9. What is STSC1?
10. Line of sight of wireless sensor tag is _____
11. A _____ is thermally sensitive resistor that exhibits a large change in resistance.
12. _____ measures temperature by correlating the resistance of the RTD with temperature.
13. _____ consists of two different metals connected at two points.
14. Which type of temperature sensor is placed in Integrated Circuits?
15. Which sensor is linear and low accuracy?
16. What is the use of the MPU 6050 Sensor?
17. How many pins are present in the MPU6050 Sensor?
18. What is the use of the AD0 pin?
19. What will happen if we supply a voltage of 25V to the Vcc of the MPU 6050 Sensor?
20. Write the formulae give us the angular velocity in the MPU 6050 along the X axis?
21. What is the optimum current that is required to operate the MPU 6050 Sensor?
22. What kind of sensor is the MPU 6050 Sensor?
23. What is the datatype of the output given by the temperature module on the MPU 6050?
24. What is the formula for calculating the temperature in degrees celsius from the raw data gathered by the temperature module of the MPU 6050?
25. Mention applications of IoT.
26. List advantages of IoT.
27. List sensors can be used in IIoT.
28. Name the sensor used to measure heart beats.
29. Difference between sensors and actuators.
30. Difference between Arduino and Raspberry Pi.

Experiment 7

Aim: To Assemble and Controlling of actuators using Arduino Uno.

Hardware Required:

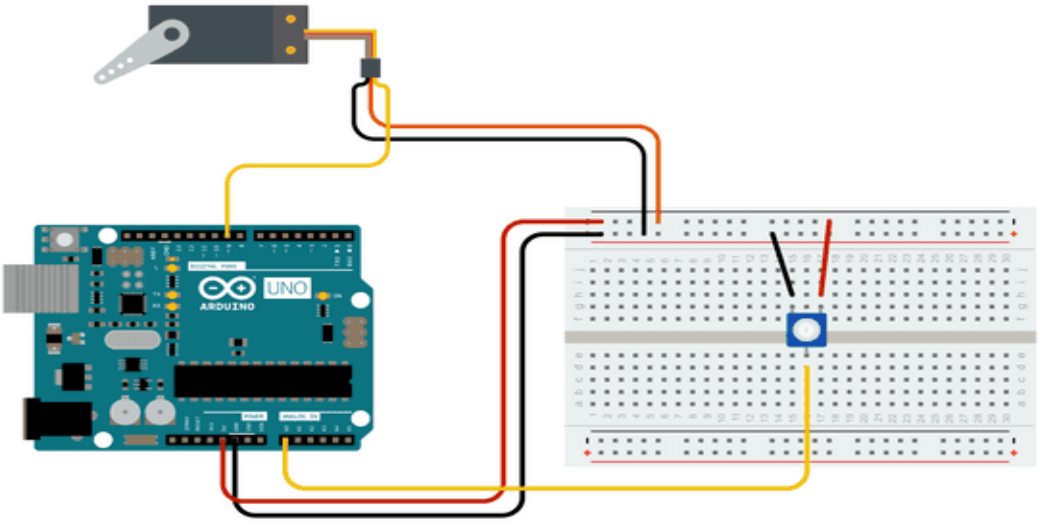
1. 1x Arduino
2. 1x LED
3. 1x Motor
4. 1x Buzzer

Actuators

Servo Motors: Servomotors have three wires: power, ground, and signal. The power wire is typically red, and should be connected to the 5V pin on the Arduino board. The ground wire is typically black or brown and should be connected to a ground pin on the board. The signal pin is typically yellow or orange and should be connected to PWM pin on the board. In these examples, it is pin number 9.

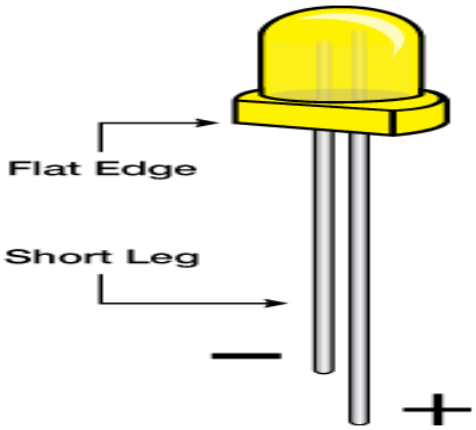
Knob Circuit

For the **Knob** example, wire the potentiometer so that its two outer pins are connected to power (+5V) and ground, and its middle pin is connected to Ao on the board. Then, connect the servomotor to +5V, GND and pin 9.



LED: LEDs (light-emitting diodes) are small, bright, power-efficient lights commonly used in electronic products.

An LED light is a polarized part, meaning it has to be connected to a circuit in a certain way to work properly. Specifically, each LED has a positive leg and a negative leg. These can be identified visually by length: the negative leg has been made shorter.



Buzzer: The buzzer is a sounding device that can convert audio signals into sound signals. It is usually powered by DC voltage. It is widely used in alarms, computers, printers and other electronic products as sound devices. It is mainly divided into piezoelectric buzzer and electromagnetic buzzer, represented by the letter "H" or "HA" in the circuit. According to different designs and uses, the buzzer can emit various sounds such as music, siren, buzzer, alarm, and electric bell.

Code:

```
// Pin definitions

const int ledPin1 = 2;  // Connect first LED to digital pin 2

const int ledPin2 = 3;  // Connect second LED to digital pin 3

const int motorPin = 4; // Connect motor to digital pin 4

void setup() {

pinMode(ledPin1, OUTPUT); // Set LED pin as output

pinMode(ledPin2, OUTPUT); // Set LED pin as output

pinMode(motorPin, OUTPUT); // Set motor pin as output }

void loop() {

// Turn on LED 1 and motor
```

```
digitalWrite(ledPin1, HIGH);

digitalWrite(motorPin, HIGH);

delay(1000); // Wait for 1 second

// Turn off LED 1 and motor

digitalWrite(ledPin1, LOW);

digitalWrite(motorPin, LOW);

delay(1000); // Wait for 1 second

// Turn on LED 2

digitalWrite(ledPin2, HIGH);

delay(1000); // Wait for 1 second

// Turn off LED 2

digitalWrite(ledPin2, LOW);

delay(1000); // Wait for 1 second

}
```

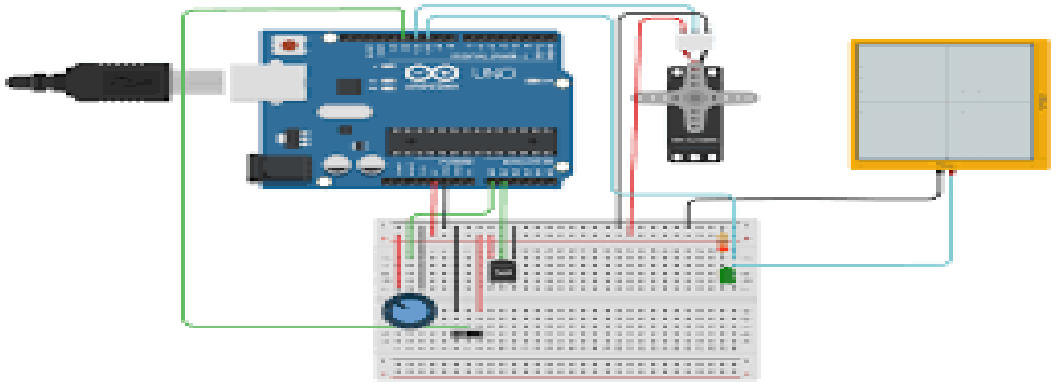
About the program

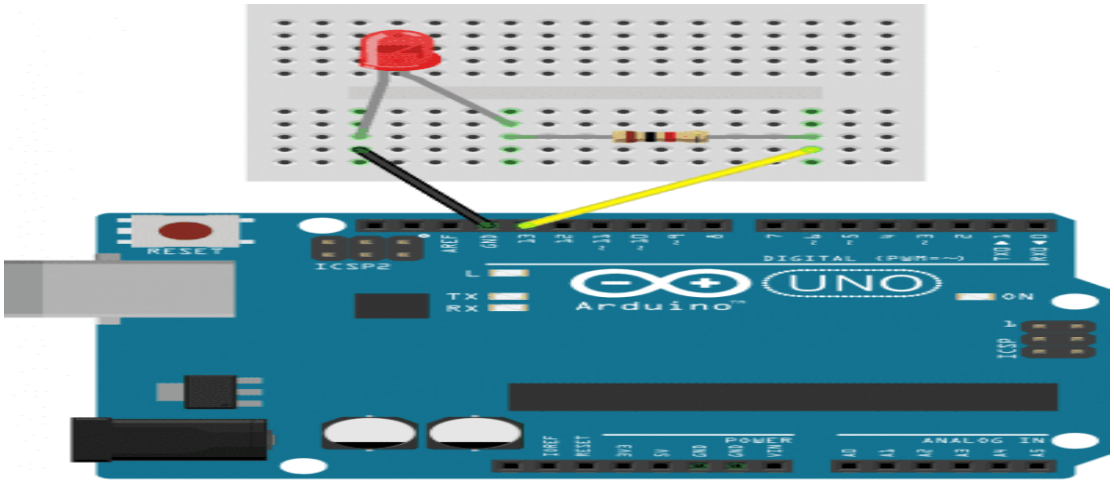
We have three actuators: two LEDs connected to digital pins 2 and 3, and a motor connected to digital pin 4. The **pinMode()** function is used to set these pins as outputs in the **setup()** function.

The **loop()** function continuously repeats the following steps:

1. Turns on LED 1 and the motor by setting the corresponding pins to **HIGH**.
2. Waits for 1 second using **delay(1000)** function.
3. Turns off LED 1 and the motor by setting the corresponding pins to **LOW**.
4. Waits for 1 second.
5. Turns on LED 2.
6. Waits for 1 second.
7. Turns off LED 2.
8. Waits for 1 second.

Connections:





Result:
Two or more actuators can be connected together with Arduino and programmed to performed action.

Viva Voce:

1. What is the ZigBee protocol?
2. What is meant by the library in Arduino in IoT?
3. Meaning of Sketch in Arduino in IoT?
4. What is the application of IoT in Environmental Monitoring?
5. What is IoT ThingWorx?
6. Explain IoT GE Predix?
7. What is IoT Contiki?
8. What is the Bluegiga APx4 Protocol for Internet of Things (IoT)?
9. What role does the network play in the Internet of Everything?
10. How Wireless Communications might affect the Development and Implementation of the Internet of Things (IoT)?
11. How does the Internet of Everything relate to the Internet of Things?
12. What is the difference between the Internet of Things (IoT) and the Sensor Business?
13. What impacts will the Internet of Things (IoT) have on Economic Growth?
14. Why will the Internet of Things (IoT) be successful in the coming years?
15. What impacts will the Internet of Things (IoT) have on the Health Care Sector?
16. What are the main Social and Cultural Impacts of the Internet Of Things (IoT)?
17. Will IoT actually work over the Internet or will it have its own dedicated wide area network?
18. What is the use of BLE in IoT?

19. What are GPIO Pins?
20. What is the latest Raspberry Pi release?
21. How many GPIO pins are there in Raspberry Pi?
22. What are Interrupts in Arduino?
23. List a few operating systems that Raspberry Pi supports?
24. How do you run Raspberry pi in headless mode?
25. What are the available wireless communications boards present in Raspberry Pi?
26. What Python libraries used in Raspberry Pi to control GPIO pins?
27. Can node JS be used in Raspberry Pi to control GPIO pins?
28. What is the syntax to read analog and digital data from a sensor in Arduino?
29. Difference between IIoT and IoT?
30. Examples of MEMS sensor?
31. What will happen in terms of Jobs Losses And Skills As IoT Makes Devices And Robots More Intelligent?

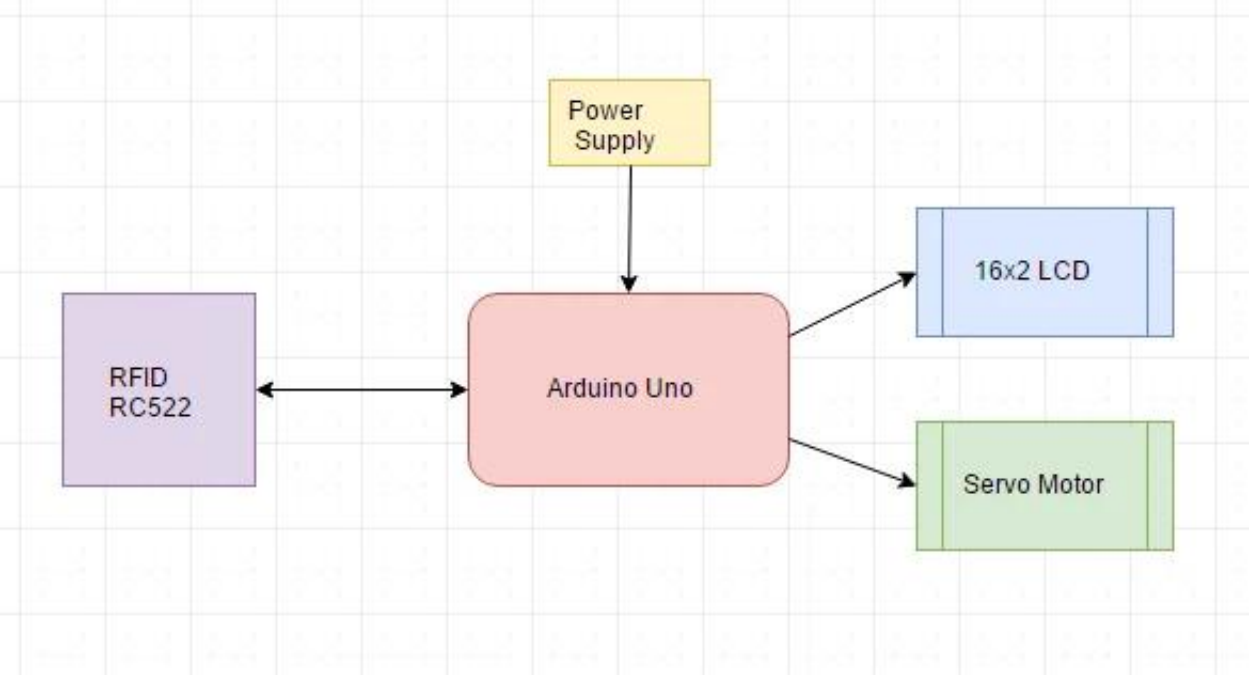
Experiment: 8

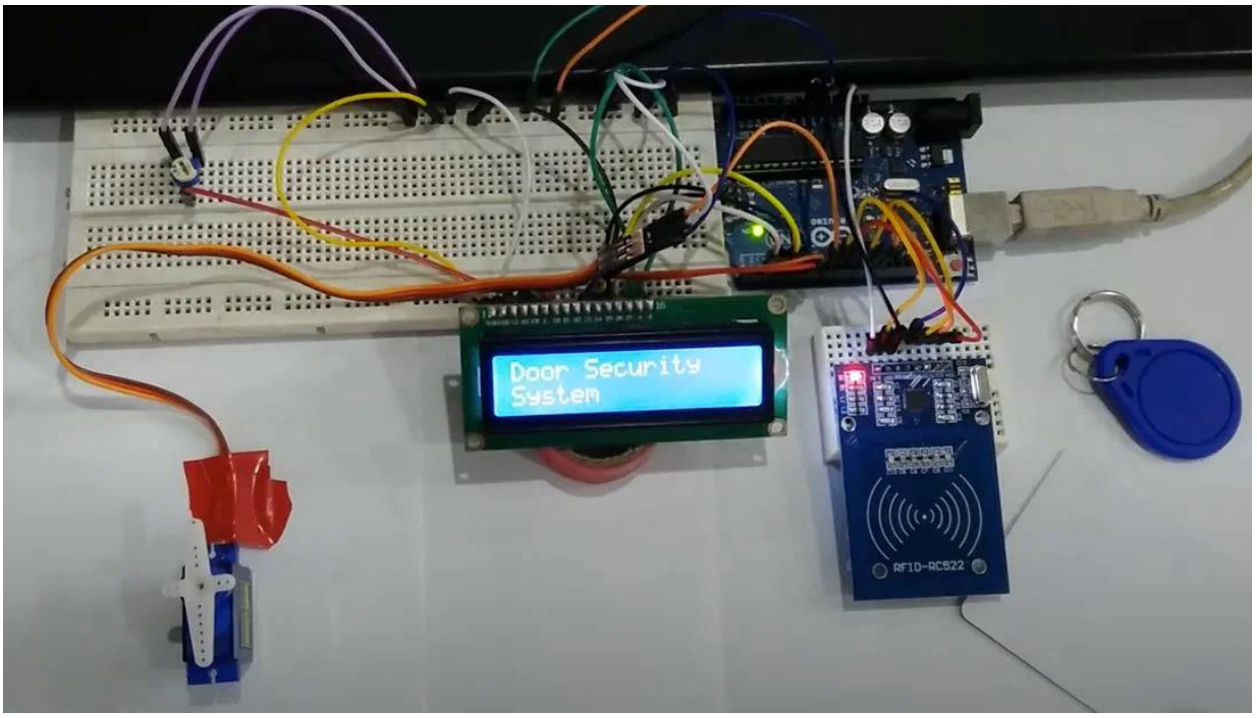
Aim: To Create a Smart door lock system-using RFID.

Hardware Required:
Arduino Uno
RFID
LCD
I2C module
Servomotor
Door lock
Jumper Wires

A door security system is used to stop unauthorized people to get access to a room or a building. Such system is very useful if you have important data, information, and research stored in a computer or in a bundle of files. This system is very useful in the Military, Research Centers, Banks, Server rooms, Jails, Government offices, and Nuclear Plants. It prevents large destruction and loss.

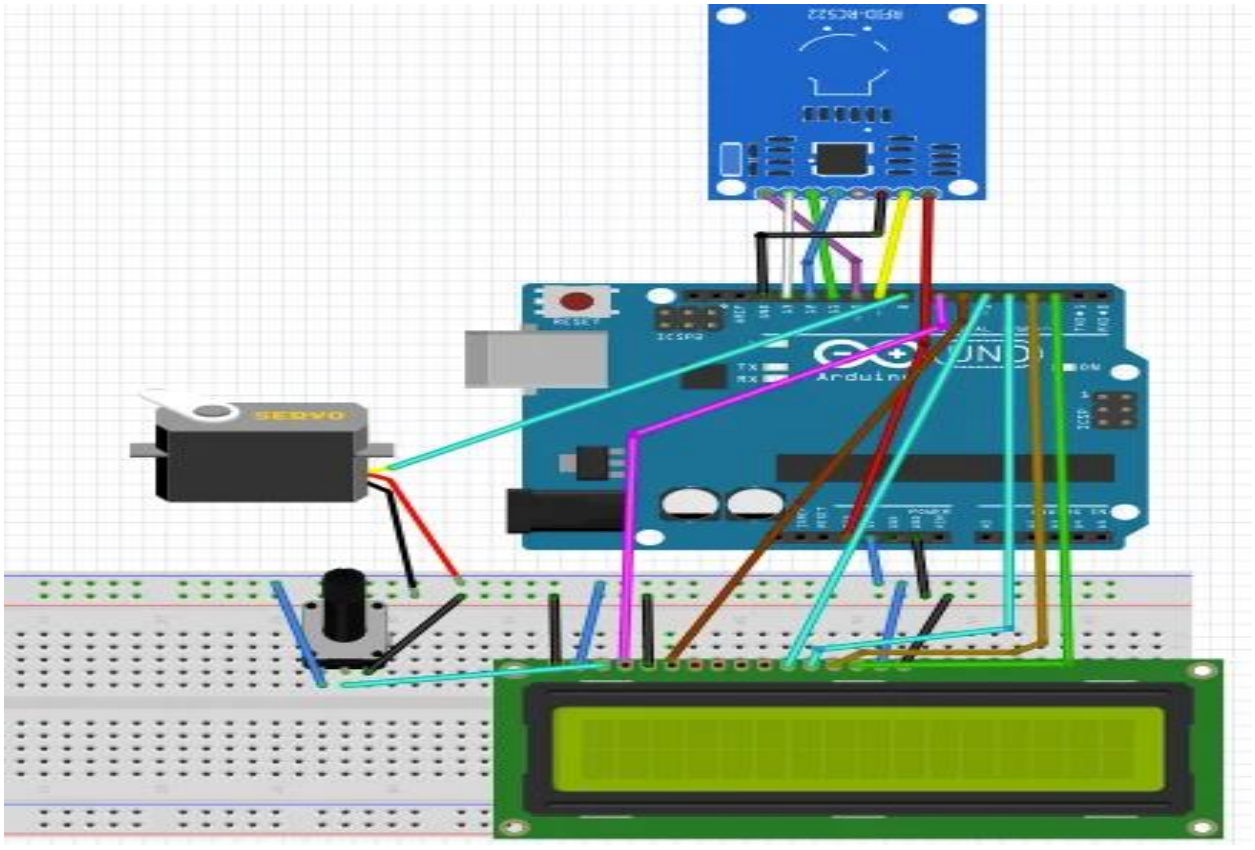
1. **Arduino Uno**: We will use Arduino due to its easy use. It also provides several digital pins to interface with the servomotor and RFID module at the same time. It is very friendly when you prototyping any project.
2. **RFID RC522:** This module is based on MFRC522 IC which is used in contactless communication. It is used both as a reader and a writer.
3. **SG-90 Servo Motor**: SG90 is a low-cost and high output power servo motor. In this project, servo motor is used to indicate that the door is opening and closing. It is easy to use and program and also provides precise movement. Learn more about servo at servo motor interfacing with Arduino
4. **16×2 LCD:** 16×2 LCD is used to display 16 text/characters in two lines. It has a total of 16 pins. A 10K ohm potentiometer is connected with pin three to set the contrast of the LCD. It also contains a backlight LED. In this project, **LCD** is used to show messages.
5. Jumper wires





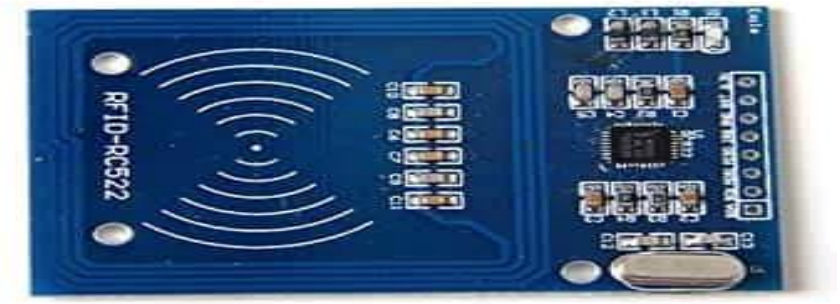
Connection Diagram of RFID Door Security System

This section will explain the connections of the various components specified above with Arduino Uno to form the door security system.



Arduino with RC522

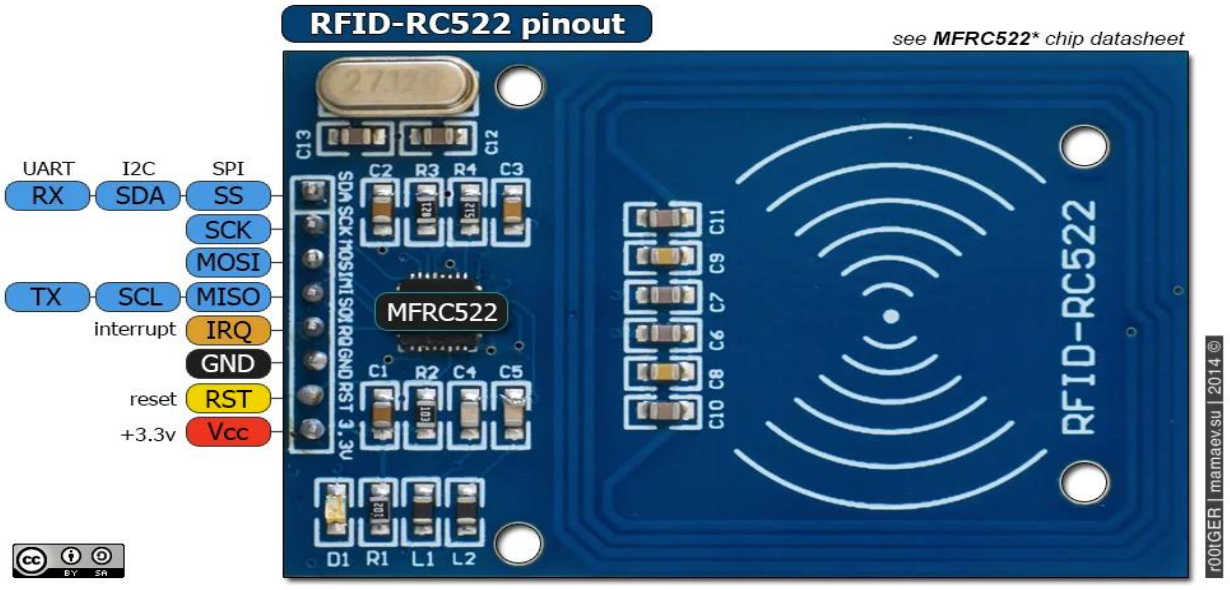
RC522 is a Multi-communication RFID Module for Arduino and Microcontrollers. The RC522 is known as MFRC-522 due to its NFX semiconductor microcontroller. The module allows the developers to interface it with any other SPI, I2C, and UART based microcontrollers. It comes with an RFID card tag and key fob consisting of 1KB of memory.



The RC522 module works on 13.56 MHz frequency and it can act as a reader and write for UID/RFID cards. The RFID cards communicate with the module at a short distance with radio frequency due to the mutual induction technique. In most of the security and commercial products, the module is effective because the errors and issues with RFID Tags are detectable by it. In this module, there are only two kinds of pins. So, the first one is power and the second one is the communication pins. Therefore, the device may have its microcontroller chip on itself but it only makes it to works as an RFID. The onboard microcontroller won't make the module a stand-alone device.

RC522 Pinout

All the pins of MFRC/RC522 RFID card Reader are:



The table below shows the eight pins found on the RC522 RFID module. You can also view their descriptions.

Pin	Description
VCC	The power pins are VCC. In some versions of RC522, this pin is denoted by 3V3 on the module instead of VCC.
RST	This is the reset pin for the module. Therefore, it resets the device in case of an error when a device isn't giving any response.
GND	Ground helps to make the common ground with every external device, e.g. power Supply or microcontroller.
IRQ	The device can go into sleep mode to save power. So, the IRQ helps to wake it.
MISO	This pin connects with the microcontroller for SPI communication. However, it transfers the data from module to the microcontroller. The MISO pin is also useable for other functions instead of SPI. It can also interface with I2C for clock pulse and UART Serial for Data transfer from the module.
MOSI	MOSI is the data input pin for RFID module in SPI communication
SCK	The SCK pins help to send the clock pulse in SPI communications.
SS	The SS pin is a chip enable pin in SPI communication. Therefore, it receives the signal when Master (Arduino) must perform SPI communication. The SS pin in RFID is useable as a second pin (SDA) for I2C communication. It also receives data during UART communication.

Connections of RC522 Module with Arduino

RC522 RFID Reader Module	Arduino
VCC	3.3V
RST	Pin 9
GND	GND
IRQ	Not connected
MISO	Pin 12
MOSI	Pin 11
SCK	Pin 13
SDA	Pin 10

Arduino with 16×2 LCD

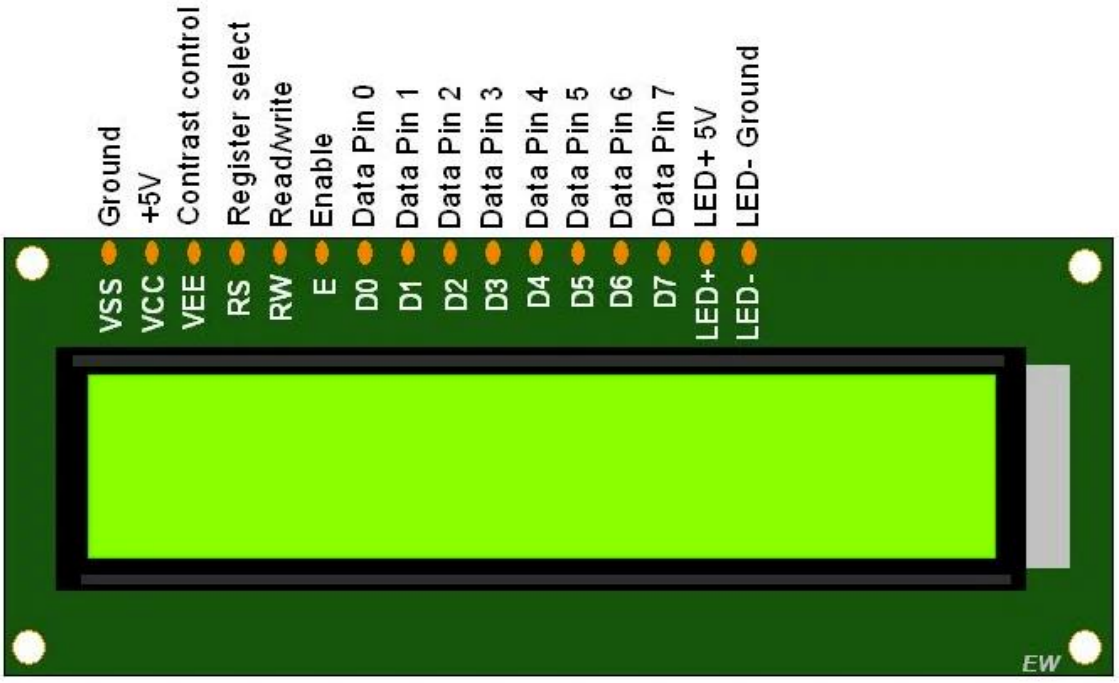
We will use a 16×2 LCD in our project to display different messages. To connect a 16×2 LCD with Arduino we will require an additional 10k potentiometer as well.

We have a dedicated tutorial regarding interfacing 16×2 LCD display with Arduino with some example sketches. Have a look at it before proceeding further for a better understanding of the LCD.

There are two types of pins on the whole 16×2 LCD module. Some pins are used to send to 16×2 LCD and some are command pins. In other words, every pin has a role in controlling a single pixel on the display.16 x 2 LCD has sixteen columns and two rows. That means, it can display sixteen characters per row and it has two such rows.

Pinout

The diagram shows the pin configuration of a 16×2 LCD display. It has sixteen pins.



- D0 – D7: Pin number 7-14 are data bus lines that are used to send data from Arduino which you want to display on LCD. With these 8 data lines, data can be transferred either in an 8-bit format or in a 4-bit format. In a 4-bit format, only upper four bits (D4-D7) are used to send data from Arduino to LCD. The full byte is transmitted in two successive transmissions. A 4-bit format is used to save GPIO pins of Arduino. Because fewer GPIO pins of Arduino will be required to transfer data.
- Contrast Select (VEE): It will help to control the contrast of PIXELS according to the 16X2 LCD light.
- RS: This pin is known as a register select pin. It helps to toggle the command/data register.
- R/W: The signal on this pin will decide whether it is going to read from LCD or write on it.
- EN: Enable pin will help to transfer the instruction from the data pins and another command pin to the LCD. It act as permission to internal registers.
- VSS: It’s a ground pin for common grounds.
- VCC: The power pin will use for voltage input to the 16X2 LCD.

Arduino connections with 16×2 LCD

We are using the following connections as described below. Refer to the schematic diagram to have a clearer idea of the connections.

16×2 LCD	Arduino
D4 – D7	Pin 5,4,3,2
Enable	Pin 6
RS	Pin 7
RW	GND
VEE	10k POT (Middle Leg)
VSS	GND
VCC	+5V
LED+	+5V
LED-	GND

Arduino with Servo Motor

SG90 is a low-cost and high output power servo motor. It can rotate up to 180 degrees and each step can be of maximum of 90 degrees. Moreover, it is small enough that it can easily fit into your robotics ARM or obstacle avoidance robotics projects. On top of that, it requires only one output pulse signal to control its movement.

The following figure shows the pinout diagram of SG90 servo motor. It consists of three pins only such as PWM, ground and Vcc pin. Brown, orange and red wires are GND, Vcc and PWM pins respectively. Details and functionality of each pin is listed in the next section.

Pin Configuration Details

Vcc and ground, as their name suggests, are the power supply pins which are used to a power servo motor. Only 5 volts of power signal is required to power this motor. Mostly microcontrollers or development boards have onboard 5 volts supply which we can use to power SG90 servos.

This table briefly describes all three pins:

Wire of motor	Possible colors of each wire
VCC pin	Red
GND pin	Black, or brown
Control Signal / PWM pin	Yellow, orange, or white

Connections of Servomotor with Arduino are as follows:

Servo Motor	Arduino
VCC pin	5V
GND pin	GND
Control Signal	Pin 8

Door Security System Using RFID RC522

When the user brings the token near the module it detects the token and reads its value. If the value is same as defined in the code then Access Granted message is shown on LCD and it opens the door for the user. After some delay, which is also, defined in the code the door automatically closes. This process occurs every time the user wants to access it. There can be multiple users but each user has it unique token.

Arduino Code

Open your Arduino IDE and go to **File** > **New** to open a new file. Copy the code given below in that file and save it.

```
#include <deprecated.h>
#include <MFRC522.h>
#include <MFRC522Debug.h>
#include <MFRC522Extended.h>
#include <MFRC522Hack.h>
#include <require_cpp11.h>
#include<RFID.h>
#include <SPI.h>
#include <Servo.h>
#include <LiquidCrystal.h>

LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

Servo myservo; // create servo object to control a servo

#define val0 117 // chnage the value according to token no
#define val1 134
int pos = 0;
/* Define the DIO used for the SDA (SS) and RST (reset) pins. */
#define SDA_DIO 10
#define RESET_DIO 9
RFID RC522(SDA_DIO, RESET_DIO);

void setup()
{
  myservo.attach(8);
  Serial.begin(9600);
  /* Enable the SPI interface */
  SPI.begin();
  /* Initialise the RFID reader */
  RC522.init();
  lcd.begin(16, 2);
}
void loop()
{
  byte i = 0;
  int ID;
  if (RC522.isCard())
  {
    RC522.readCardSerial();
    Serial.println(RC522.serNum[i], DEC);
    Serial.println("Card detected:");
    ID = RC522.serNum[0];
  }
  delay(500);
  if ( val0 == ID)
  { lcd.clear();
    lcd.print("Access Granted");
    lcd.setCursor(0, 1);
    lcd.print("Welcome Haris");
    delay(1800);
    lcd.clear();
    lcd.print("Door Opening");
```

```
  delay(1500);
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    delay(15); // waits 15ms for the servo to reach the position
  }
  delay(5000);
  lcd.clear();
  lcd.print("Door Closing");
  delay(1500);
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    delay(15); // waits 15ms for the servo to reach the position
  }
 }
 else
 {
  lcd.clear();
  lcd.print("Door Security ");
  lcd.setCursor(0, 1);
  lcd.print("System ");
 }
 if ( val1 == ID)
 { lcd.clear();
  lcd.print("Access Granted");
  lcd.setCursor(0, 1);
  lcd.print("Welcome Bilal");
  delay(1800);
  lcd.clear();
  lcd.print("Door Opening");
  delay(1500);
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    delay(15); // waits 15ms for the servo to reach the position
  }
  delay(5000);
  lcd.clear();
  lcd.print("Door Closing");
  delay(1500);
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    delay(15); // waits 15ms for the servo to reach the position
  }
 }
 else
 {
  lcd.clear();
  lcd.print("Door Security ");
  lcd.setCursor(0, 1);
  lcd.print("System ");
 }
}
```

Click on the upload button to upload the code to the board.
Bring the RFID tag or card close to the RC522 module. If access is granted, the LCD will display relevant messages and the servo motor will start moving to indicate that the door is opening. After few seconds, the servo motor goes back to its initial position indicating the door is closing.

Viva Voce:

1. List the fundamental components of IoT. 2. Explain Pulse Width Modulation.
3. List the challenges of IoT.
4. List mostly used IoT protocols.
5. Explain WSN.
6. Define Zigbee.
7. Explain Z-Wave.
8. Discuss MQTT.
9. List the various types of antennas designed for IoT devices.
10. Explain the types of IoT.
11. Define Asset Tracking.
12. Difference between the IoT network and Wireless Sensor Network.
13. Define Shodan.
14. Define Arduino.
15. Explain IoT test approaches.
16. Explain Python libraries used in Raspberry Pi to control GPIO pins.
17. Difference between the IoT network and Wireless Sensor Network.
18. Explain the types of testing in IoT.
19. Explain the default operating system of Raspberry Pi, May I use any other operating systems. 20. List the popular software IDEs are using in IoT.
21. List the popular hardware prototypes are using in IoT.
22. List Bluetooth Low Energy (BLE) Protocol for Internet of Things (IoT).
23. State the differences between Arduino and Raspberry pi.
24. List the most used sensors types in IoT.
25. Explain the hardware communication interfaces present in the Arduino board.
26. Explain how you insert bulk data in MongoDB, and can write the query for that.
27. Define Bluegiga APX4 protocol for an Internet of Things.
28. Discuss the economic impacts of the increased application of IoT.
29. With example give the top five Machine-to-Machine (M2M) applications in the world.
30. Difference between a wireless sensor network (WSN) and the Internet of Things (IoT) network.

Experiment 9

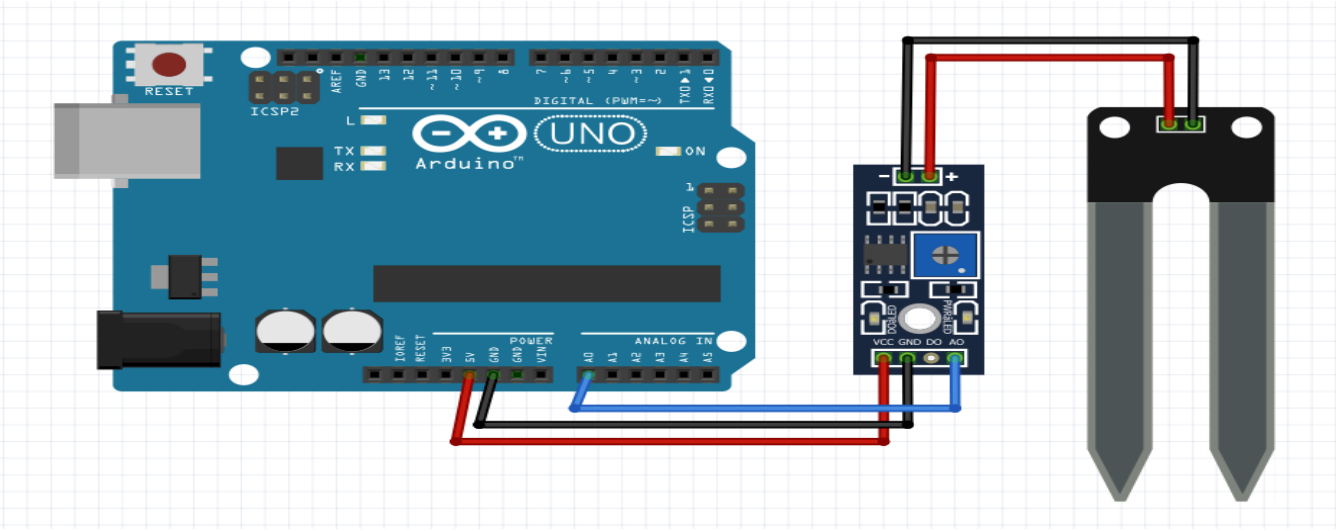
Aim: Case studies of IoT in Healthcare and propose any IoT model for the healthcare sector.

Materials Required:

1. Arduino Uno board
2. Soil moisture sensor
3. Breadboard
4. Jumper wires
5. USB cable for Arduino
6. Computer with Arduino IDE installed

Theory:

Soil moisture sensors measure the volumetric water content in soil. They typically consist of two probes that allow the current to pass through the soil. More water in the soil results in lower resistance and higher conductivity between the probes.



Procedure:

Step 1: Setup the Arduino and Sensor

Gather Components: Ensure you have all the necessary components.

Connect the Soil Moisture Sensor to the Arduino:

VCC (Power): Connect the VCC pin of the soil moisture sensor to the 5V pin on the Arduino.

GND (Ground): Connect the GND pin of the soil moisture sensor to a GND pin on the Arduino.

Analog Output (A0): Connect the A0 pin of the soil moisture sensor to an analog input pin (e.g., A0) on the Arduino.

Step 2: Write the Arduino Code

Open the Arduino IDE: Launch the Arduino IDE on your computer.

Write the Code: Enter the following code into the Arduino IDE:

```
// Define the pin connected to the soil moisture sensor
const int sensorPin = A0;
void setup()
{
// Initialize serial communication at 9600 bits per second
Serial.begin(9600);
}
void loop()
{
// Read the value from the soil moisture sensor
int sensorValue = analogRead(sensorPin);
// Print the sensor value to the serial monitor
Serial.print("Soil Moisture Value: ");
Serial.println(sensorValue);
// Wait for a second before taking another reading
delay(1000);
}
```

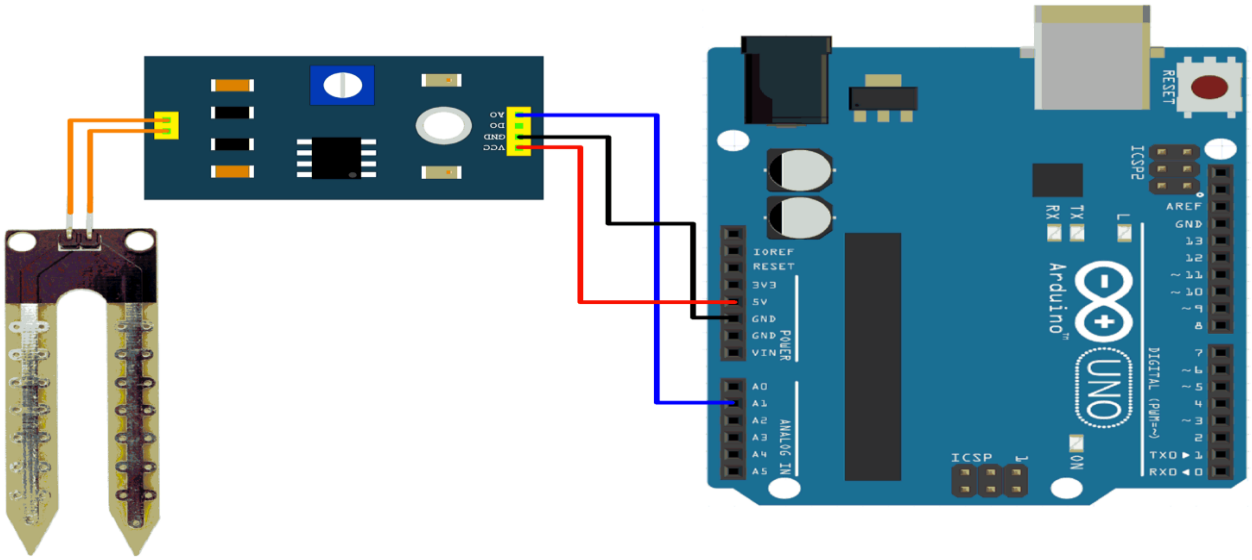
3. Upload the Code: Connect the Arduino to your computer using the USB cable and upload the code to the Arduino board.

Step 3: Testing and Observation
Open Serial Monitor: After uploading the code, open the Serial Monitor from the Arduino IDE (Tools > Serial Monitor) to see the sensor readings.
Take Readings: Observe the sensor values as you place the sensor in different soil conditions:
Dry Soil: Note the sensor value in dry soil.
Wet Soil: Note the sensor value in wet soil.
Record Data: Record the sensor values for different soil moisture levels.

Step 4: Analyzing the Data
Interpret Sensor Values: Higher sensor values typically indicate lower moisture levels (dry soil), while lower sensor values indicate higher moisture levels (wet soil).
Calibrate if Necessary: Based on your observations, you might need to calibrate the sensor readings for accurate moisture level interpretation.

Troubleshooting:
No Sensor Readings: Check all connections and ensure the sensor is properly connected to the Arduino.
Inconsistent Readings: Ensure the sensor probes are clean and fully inserted into the soil.
Extensions:
Integrate an LCD to display moisture levels.
Connect a relay module to automate an irrigation system based on moisture levels.
Use data logging to record moisture levels over time for analysis.

Circuit Diagram:
Here's a simple circuit diagram to illustrate the connections:



Conclusion:

By completing this experiment, you have successfully interfaced a soil moisture sensor with an Arduino board, measured soil moisture levels, and interpreted the sensor readings. This knowledge can be applied to automate irrigation systems in agriculture.

Viva voce:

1. Explain the role of pulse Sensor.
2. How can you explain Smart hospital?
3. Explain the role of IoMT.
4. How can heartbeat sensor benefited in healthcare sector.
5. Explain the role of Wireless Communications for IoT.
6. What parameters are required to be noted in healthcare sector?
7. How smart monitoring of patient can be helpful.
8. List sensors, which can be used in healthcare sector.
9. Explain how you will check the quality of hospitals.
10. List some healthcare sensors.
11. What are the features of influxDB.
12. Define telemedicine.
13. Discuss importance of hygiene in hospitals.
14. Briefly discuss the role of technology in healthcare.
15. Discuss where to place sensors in a hospital.
16. Explain MQTT.
17. Explain Zigbee.
18. Define SHODAN.
19. Explain Sharding.
20. On which principle does thermocouple works.
21. Explain about replication.
22. Discuss the role of Sketch in IDE.
23. List organizations that support IoT.
24. Explain the layers of OSI.
25. What is DB Command?
26. Define library in Ardunio IDE.

27. What is the standard form of ISM?
28. How many firms does the Bluetooth frame format have?
29. How many bits does the access code have in Bluetooth frame format?
30. How many bits does the header have in Bluetooth frame format?

Experiment 10

Aim: Lab based mini projects

Designing a complex mini-project in cloud computing involves selecting a specific problem or application and implementing it using cloud services and technologies.

MAX TEAM SIZE: 5

MAX MARKS: 30 MARKS

TOPICS: WILL BE SHARED IN THE LAB

Project Details (IoT)		
Sr. No	**Title of Project**	**Description**
1	Autonomous Drone System	Autonomous drones are unmanned aerial vehicles (UAVs) that operate using Artificial Intelligence (AI)-powered navigation and operational software, and do not require a human pilot. From taking off and landing to carrying out aerial site inspections and surveying, these aircrafts complete tasks and make decisions on their own.
2	Industry 4.0 based Projects	Industry 4.0 is signalling a change in the traditional manufacturing landscape. Also known as the Fourth Industrial Revolution, Industry 4.0 encompasses three technological trends driving this transformation: connectivity, intelligence and flexible automation. Industry 4.0 converges IT (Information Technology) and OT (Operational Technology), to create a cyber-physical environment.
3	Smart Parking System	Various devices and processes form the structure of smart parking, acting as parking space detectors. On the one hand, the deployment of sensors and/or cameras, which record and process data and images to provide real-time traffic occupancy data for the area we are heading to. An IoT cloud-based system, on the other hand, allows these devices to be connected and the data to be centralized. The data are then analyzed using big data in order to calculate the availability of on-street parking spaces or spaces in public and private parking facilities.
4	IoT based Home Delivery System	home automation has been projected to target wide array applications for the new digital consumer. some of the areas where consumers can expect to see home automation led iot-enabled connectivity are: lighting control, hvac, lawn/gardening management, smart home appliances, improved home safety and security, home air quality and water quality monitoring, natural language- based voice assistants, better infotainment delivery, ai-driven digital experiences, smart switches, smart locks and smart energy meters.
5	IoT based Rovers	Rovers are important for conducting in-situ scientific analysis of objectives that are separated by many meters to tens of kilometers. Current mobility designs are complex, using many wheels or legs. They are open to mechanical failure caused

		by the harsh environment on Mars. This thesis describes IOT Based Solar Rover, a four wheeled rover capable of traversing rough terrain using an efficient high degree of mobility suspension system. The primary mechanical feature of the IOT Based Rover design is its drive train simplicity, which is accomplished by using only two motors for mobility.
6	IoT for Smart Cities	The Internet of Things (IoT) shall be able to incorporate transparently and seamlessly a large number of different and heterogeneous end systems, while providing open access to selected subsets of data for the development of a plethora of digital services. Building a general architecture for the IoT is hence a very complex task, mainly because of the extremely large variety of devices, link layer technologies, and services that may be involved in such a system.
7	IoT based web controlled home automation	In Web controlled Home Automation project, any microcontroller can be used for performing all the operations. It will communicate with ESP8266 Wi-Fi module to send and receive data from the Adafruit server and take action accordingly to turn ON/OFF relay or load and displaying the status of loads over LCD. We have used 16x2 LCD display for displaying the status of connected AC appliances.
8	IoT based security system with voice messages	In this age of technology, security is one of the key issues. Also, face recognition is a significant part of the purpose of security and surveillance. Our goal is to explore the implementation of a Raspberry Pi-based face recognition system using conventional face detection and recognition techniques for resolving security issues. For face detection and recognition, the Eigen face method is used, and to implement the steps of this method.
9	IoT based surveillance camera with motion	The Internet of Things (IoT) combines the idea of technology providing value to our daily lives. One major category in which technology benefits us is the concept of security and privacy. Smartphones can act as a security alert system as the smartphone is the most widely-used smart device. Recently, the use of Artificial Intelligence (AI)-integrated smart IoT devices has significantly increased as well.
10	Heart beat monitoring over internet	In this project we are going to make a Heart Beat Detection and Monitoring System using Arduino that will detect the heart beat using the Pulse Sensor and will show the readings in BPM (Beats Per Minute) on the LCD connected to it. It will also send the readings to ThingSpeak server using the Wi-Fi module ESP8266, so that Heart Beats can be monitored from anywhere in the world over the internet. ThingSpeak is a great source for displaying the data online and you can access the data from ThingSpeak at any time and at any place.
11	IoT based Air Pollution Monitoring System	IoT Based Air Pollution Monitoring System in which we will monitor the Air Quality over a webserver using internet and will trigger a alarm when the air quality goes down beyond a certain level, means when there are sufficient amount of harmful gases are present in the air like CO2, smoke, alcohol, benzene and NH3. It will show the air quality in PPM on the LCD and as well as on webpage so that we can monitor it very easily.
12	IoT based Indicators	Using relevant historical data to extract the corresponding factors such as industry scale, marketing demand, technology innovation in different hierarchical levels, and quantifies these factors. Making use of the Analytic Hierarchy Process (AHP), we build the indicator framework of IOT industry growth. The empirical results show that: these factors, namely IOT industry scale, innovation level and

		technology progress, all promote IOT industry growth relevantly.
13	IoT based notice boards	In IOT based Web Controlled Notice Board, Internet is employed to wirelessly send the message from Browser to the display. A local web server is created, this could be a global server over net. At the PIC microcontroller, LED matrix is used to display message and flask for receiving the message over network.
14	Industrial Automation and Control	As the world is digitalizing at a drastic rate, industries across several sectors are opting for industrial automation. Industrial automation products refer to the use of control systems, such as information technologies, robots, and tools for handling different types of machinery and processes used in several industries. Therefore, automation products are important as they effectively decrease production costs and improve the quality of products, leading to better profits. Industrial automation is a product of digitalization that has significantly improved the industrial sector. It makes processes efficient by leveraging robots, computers, and other control systems to supervise various intricate tasks. Industrial automation helps reduce costs, improve productivity, enhance communication, better equipment supervision, and excel the scope of traceability.
15	Role of IoT in Agriculture 4.0	The use of IoT in Agriculture 4.0 enables field management systems to connect data gleaned in real-time from GPS-equipped drones, satellites, sensors, and other advanced tools. Systems based on IoT can automatically adapt to weather changes and plan irrigation accordingly.
16	IoT based smart clothing	Smart wearables or IoT-enabled garments are electronic devices embedded in clothing and designed to be worn near, on or within the body to provide intelligent services that may be linked into a larger smart system via the use of communication protocols.
17	Indoor Farming with the help of IoT.	Smart indoor farms are essentially indoor farms that rely on IoT devices. This farm is a closed structure with IoT sensors like humidity sensors, temperature sensors, soil moisture sensors, and actuators that remotely control plants' irrigation.
18	Face Recognition AI Robot	Recently, face recognition became a key element in social cognition which is used in various applications including human–robot interaction (HRI), pedestrian identification, and surveillance systems. Deep convolutional neural networks (CNNs) have achieved notable progress in recognizing faces. However, achieving accurate and real-time face recognition is still a challenging problem, especially in unconstrained environments due to occlusion, lighting conditions, and the diversity in head poses.
19	Smart Wheelchair	Smart Wheelchair is known as a Power Wheelchair that is integrated into multiple sensors, assistive technology, and computers that give the user with a disability such as impairment, handicaps, and permanent injury, the required mobility to move freely and safely. These types of wheelchairs are gradually replacing the traditional wheelchairs; however, their expensive costs are preventing a large size of disabled people from having one. According to the organization of World Health (WHO), only 5 to 15% out of 70 million disabled people have access to wheelchairs. Therefore, we need to offer a cost-effective Smart that not only minimized the cost but also provides plenty of features that use the latest components and technologies.

20	Brain Controlled EEG	Brain-computer interface (BCI) based on electroencephalogram (EEG) signals can provide a way for human to communicate with the outside world. This approach is independent of the body's peripheral nerves and muscle tissue.
21	Low Cost IoT Robot	There are many types of robots, from simple toy cars to advanced industrial robots. The Wi-Fi-controlled IoT robot described here uses NodeMCU and Blynk app. It can be controlled wirelessly using any Wi-Fi enabled Android smartphone.
22	IoT Enabled Pollution Meter	IoT enabled air pollution meter to monitor air quality on your smartphone using Blynk application and Arduino board. Blynk is an Internet of Things (IoT) platform to control Arduino, Raspberry Pi and the like over the Internet. In this project Blynk provides a digital dashboard on your smartphone that displays real-time air quality readings for the immediate surroundings. Blynk is not meant for a specific board or shield. It will get you online and ready for the IoT, irrespective of whether Arduino or Raspberry Pi is linked to the Internet over Wi- Fi, Ethernet or an ESP8266 chip.
23	Level based Security with the help of IoT	The Internet of Things (IoT) has been one of the fastest developing technology trends in recent years. According to IoT Analytics, by 2025, there will likely be more than 27 billion connected devices in the world. However, increasing security concerns like software vulnerabilities and cyberattacks can make many customers refrain from using IoT devices. Such Internet of Things security problems are especially significant for organizations that operate in healthcare, finance, manufacturing, logistics, retail, and other industries that have already started adopting IoT systems.
24	IoT based Precision Farming	Smart farming is a development that has emphasized information and communication technology used in machinery, equipment, and sensors in network-based hi-tech farm supervision cycles. Innovative technologies, the Internet of Things (IoT), and cloud computing are anticipated to inspire growth and initiate the use of robots and artificial intelligence in farming. Such ground-breaking deviations are unsettling current agriculture approaches, while also presenting a range of challenges.
25	IoT based cleaning of river project	To address the issue of marine debris floating on water, this project proposes a design of an autonomous water cleaning robot that can detect and collect garbage from the water bodies thus can clean the water bodies effectively. One of our objectives is to reduce the cost so that it is feasible to implement our solution on a large scale.
26	IoT based vertical farming	IoT-based systems in vertical farming can monitor and control various environmental factors, such as temperature, humidity, and light levels, to optimise growing conditions and increase productivity. These systems can also automate certain aspects of the growing process, such as irrigation and nutrient delivery, saving time and labour costs.
27	Data logging of Brain Parameters using IoT	Data exchange and automation using Internet of things (IoT) is a rapidly growing technology. It includes sensors, cyber systems – the things in IoT and cloud computing. To reach humans in real time, cyber systems communicate collaboratively at each stage over the Internet. The advances in Internet innovation have made possible techniques for the conveyance of healthcare. Networking infrastructure and common access can encourage sharing of patient data and clinical information making the Internet a perfect tool for remote patient

		observing applications.
28	Password creation with the help of brain nerves using IoT	As security becomes a strong factor in daily activities, finding secure ways to unlock machines and smartphones is a challenge due to hardware limitations and the high risk of hacking. Considering the level of security and privacy in the digital world, attackers tend to be one step ahead. The BCI measures brain changes and extracts relevant bio-features from each subject using non- invasive electroencephalogram (EEG) tests. The proposed system allows users to gain access to their devices using brain waves (bypass) instead of inserting their password manually (normal path), which saves the user time and upgrades the level of privacy as no physical actions are required during this process.
29	Smart Waste Collection Management System	Smart waste management refers to any system that uses technology to make trash collection more efficient, cost-effective and environmentally friendly. Most of these systems are equipped with the Internet of Things (IoT), a monitoring technology that collects and tracks real-time data, to help optimize waste collection and spur future innovation.
30	Self Driving Car using IoT	Self-driving cars work using the Internet of Things (IoT). IoT technology enables cars to connect wirelessly to a cloud system. It shares information about the road the car is driving on as well as the information about the car when it is moving. These IoT systems collect a huge amount of data about navigation, traffic, roads, and more.