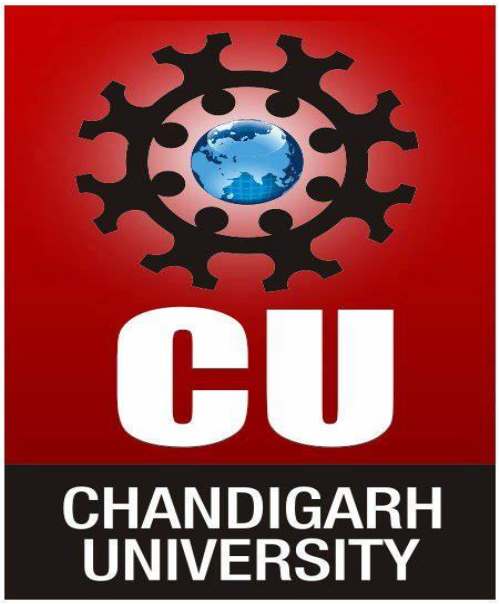
CHANDIGARH UNIVERSITY

UNIVERSITY INSTITUTE OF ENGINEERING



Lab Manual

Subject Name: Internet of Things (IoT) Lab

Subject Code: 20CSP-358

BATCH: B.E. CSE (2020-2024)

B.E. III Year – VI Semester

Even Semester: February-June, 2023

Department of Computer Science & Engineering

Gharuan, Mohali

CONTENTS

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 student`s profound future growth.

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

3. PROGRAM EDUCATIONAL OBJECTIVES (PEO)

The statements of PEOs (revised from 2022) are given below:

PEO 1. Graduates of the Computer Science and Engineering can contribute to the Nation‟s growth through their ability to solve diverse and complex computer science & engineering problems across a broad range of application areas.

PEO 2. Graduates of the Computer Science and Engineering can be successful professionals, designing and implementing Products & Services of global standards in the field of Computer Science & Engineering, becoming entrepreneurs, pursuing higher studies& research.

PEO 3. Graduates of the Computer Science and Engineering Program can be able to adapt to changing scenario of dynamic technology with an ability to solve larger societal problems using logical and flexible approach in decision-making.

Consistency of the PEOs with Mission of the Department Mission of the department –PEO matrix

PEO	**M1**	**M2**	**M3**	**M4**	**M5**
Statement	H	M	L	H	L
PEO 1					
PEO 2	M	M	H	M	L
PEO 3	L	L	M	L	H

4. PROGRAM OUTCOMES

Program Outcomes are adopted from the outcomes defined by the National Board of Accreditation of India, which is the permanent signatory of the Washington Accord. Program outcomes are defined to ensure the holistic development of students.

Engineering Graduates will be able to:

PO 1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO 2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO 3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO 4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO 5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO 6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO 7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO 8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO 9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO 10. Communication: Communicate effectively on complex engineering activities with the engineering community and with 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.

PO 11. Project management and finance: Demonstrate knowledge and understanding of the 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.

PO 12. 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 of 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 as follows:

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 of computing 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 appropriateto the program`s discipline.
SO 6. Apply computer science theory and software development fundamentals to produce computing-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.

6. PROGRAM SPECIFIC OUTCOMES (PSOs)

A Graduate of Computer Science and Engineering Program will be able:

PSO 1. To acquire proficiency in developing and implementing efficient solutions using emerging technologies, platforms and free and open-source software (FOSS).

PSO 2. To gain critical understanding of hardware and software tools catering to the contemporary needs of IT industry.

7. COURSE OBJECTIVE INTERNET OF THINGS LAB (20CSP-358)

	Course Objectives
1	To study hardware and software related to IoT.
2	To understand the functions of Node MCU, Arduino Uno and Raspberry Pi.
3	To grasp knowledge about interfacing using non-wired connection.

8. COURSE OUTCOMES INTERNET OF THINGS LAB (20CSP-358)

	Course Outcomes
1	Analyze the components of IoT system.
2	Testing of model on IoT based Simulation.
3	Illustrate real time application using Node MCU/Arduino Uno/Raspberry Pi.
4	Develop an interface between controller and sensor to capture real time data.
5	Design an application to control actuators using wireless connectivity.

9. SYLLABUS (AS APPROVED IN BOS)

Chandigarh University, Gharuan

Subject Code 20CSP-358	**Internet of Things Lab**	**L**	**T**	**P**	**C**
	Total Contact Hours: 48Hours	**0**	**0**	**2**	**1**
	Common to all Specializations of CSE 3rd Year				
	Prerequisite: Knowledge of basic electronics and computer science principles and skills, at a level sufficient to interface sensors with controllers and basic understanding of logics.				

Marks-100	
Internal-60	External-40

Unit	Course Outcome
1	Analyze the components of IoT system.
2	Testing of model on IoT based Simulation.
3	Illustrate real time application using Node MCU/Arduino Uno/Raspberry Pi.
4	Develop an interface between controller and sensor to capture real time data.
5	Design an application to control actuators using wireless connectivity.

List of Experiments

UNIT-I

1. Familiarization with Arduino/Raspberry Pi hardware and perform necessary software installation.

2. Identification of different sensors used in IoT applications.

3. Demonstration of Autodesk Tinkercad Simulation Platform.

4. Program to interface the Arduino/Raspberry Pi with LED and blinking application.

UNIT-II

5. To measure the distance of an object using an ultrasonic sensor.

6. Interfacing of Arduino/Raspberry Pi with temperature and humidity sensor with real time application.

7. To display data generated by sensor on LCD using Arduino/Raspberry Pi.

UNIT-III

8. Interfacing Air Quality Sensor (MQ135) and display data on LCD.

9. Real time application of controlling actuators through Bluetooth application using Arduino.

10. Study the Implementation of Zigbee Protocol using Raspberry Pi/Arduino.

10. CO/PO/PSO MAPPING

CO vs PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	NA	3	NA	NA	NA	NA	NA	NA	2	2	NA
CO2	3	3	3	2	3	NA	NA	NA	NA	NA	NA	2	2	NA
CO3	3	3	3	2	3	NA	NA	NA	NA	NA	NA	1	1	NA
CO4	3	3	3	2	3	1	1	NA	NA	NA	NA	2	2	NA
CO5	3	3	3	3	3	1	1	NA	1	NA	NA	3	3	NA
Target	3	3	3	2.25	3	1	1	NA	1	NA	NA	2	2	NA

11. LIST OF EXPERIMENTS (MAPPED WITH COS)

Sr No.	Experiment Name	Mapped with CO Number(s)
1	Familiarization with Arduino/Raspberry Pi hardware and perform necessary software installation.	**CO1**
2	Identification of different sensors used in IoT applications.	**CO1**
3	Demonstration of Autodesk Tinkercad Simulation Platform.	**CO2**
4	Program to interface the Arduino/Raspberry Pi with LED and blinking application.	**CO3**
5	To measure the distance of an object using an ultrasonic sensor.	**CO3**
6	Interfacing of Arduino/Raspberry Pi with temperature and humidity sensor with real time application.	**CO4**
7	To display data generated by sensor on LCD using Arduino/Raspberry Pi.	**CO4**
8	Interfacing Air Quality Sensor (MQ135) and display data on LCD.	**CO4**
9	Real time application of controlling actuators through Bluetooth application using Arduino.	**CO5**
10	Study the Implementation of Zigbee Protocol using Raspberry Pi/Arduino.	**CO5**

Code of Ethics

Any misuse or abuse of computer equipment, programs or data will result in termination of computer privileges and may lead to disciplinary action.

This includes:

1. Entering or exploring any area of the computer or network other than the program for the course and the home directory where the student`s data files are located. This includes attempting to explore the Internet without permission.

2. Attempting to log on with any other ID other than their own

3. Attempting to destroy or modify programs, records or data belonging to the university or another user.

4. Accessing or modifying any desktop icons or files without instructor's permission.

5. Computer vandalism, which includes writing on any equipment, taking mouse balls, writing on mouse pads or copyholders, or removing anything from the classroom that belongs here.

Vandalism also includes changing computer configurations or attempting to & "hack" into any system that is not part of the curriculum.

6. Unauthorized use of computing resources for unauthorized purposes.

7. Accessing or copying programs, records or data belonging to the school or another user without permission.

8. Attempting to breach the security of another user`s account or deprive another user of access to computing resources.

9. Using the university`s computing resources for personal or financial gain.

10. Transporting copies of the university`s programs, records or data to another person or site without written authorization.

11. Cheating or computer vandalism results in referral to Admin. Cheating includes copying or using other student`s work or copying or using other student`s files.

12. Computer mentioned above is also considered as controller.

13. Any hardware damage/missing in any case during the practical time concerned batch will be held responsible for the same and fine will be imposed as per the cost of the damaged/missing hardware.

12.

MANUAL TO CONDUCT EACH EXPERIMENT

Experiment 1

Aim: Familiarization with Arduino/Raspberry Pi hardware and perform necessary software installation.

Objectives:

1. To study hardware and software related to IoT

2. To understand the function of Node MCU, Arduino Uno and Raspberry Pi.

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.

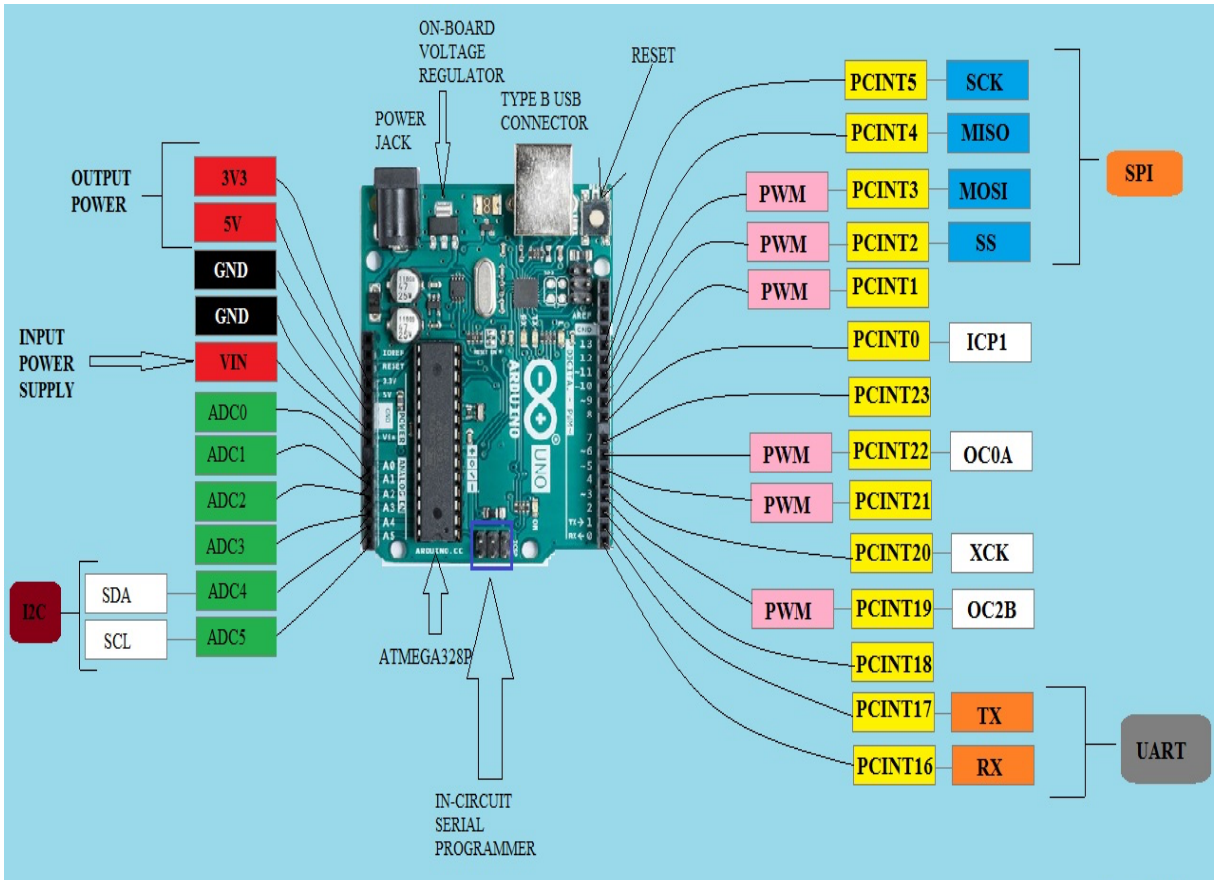


Figure: Diagram of Arduino Board

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

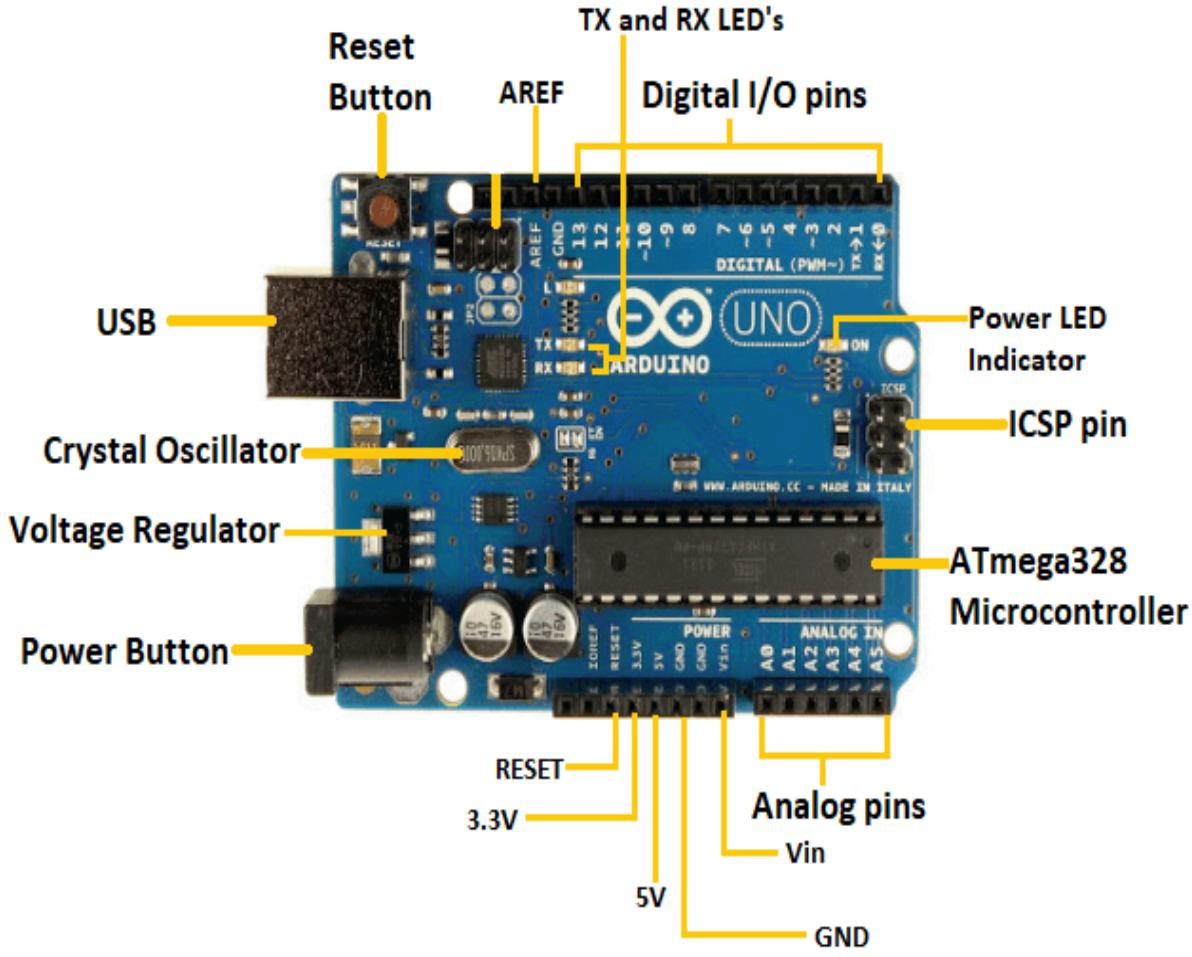


Figure: Arduino Uno

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.

RASPERRY PI

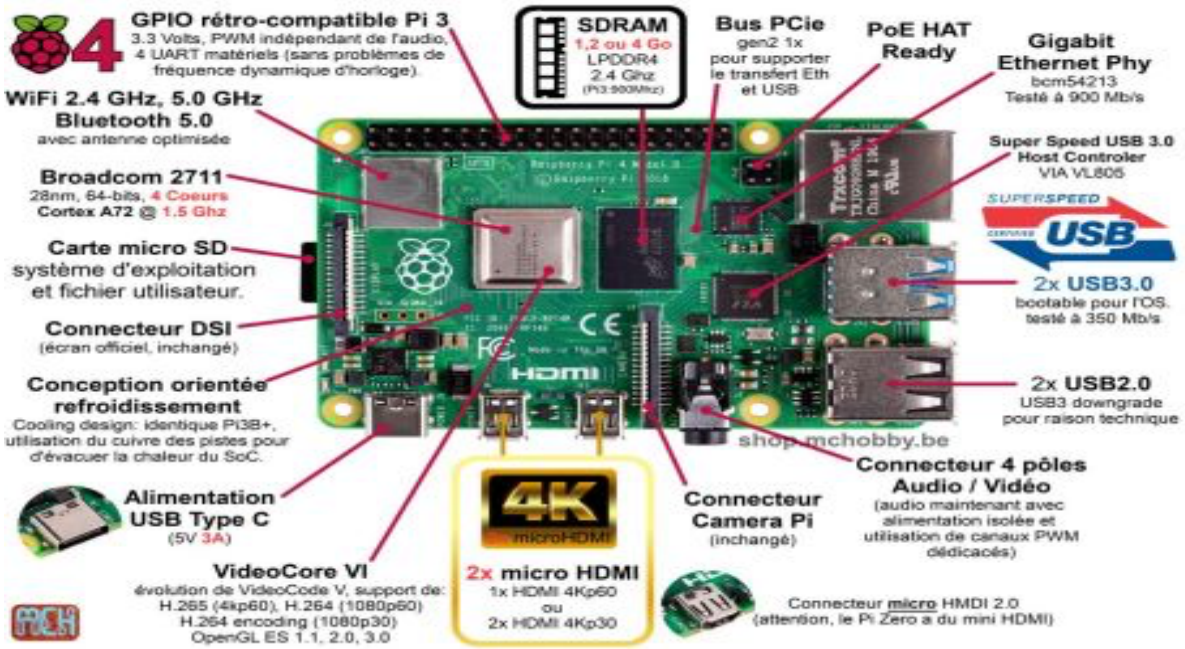


Figure: Raspberry Pi

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.

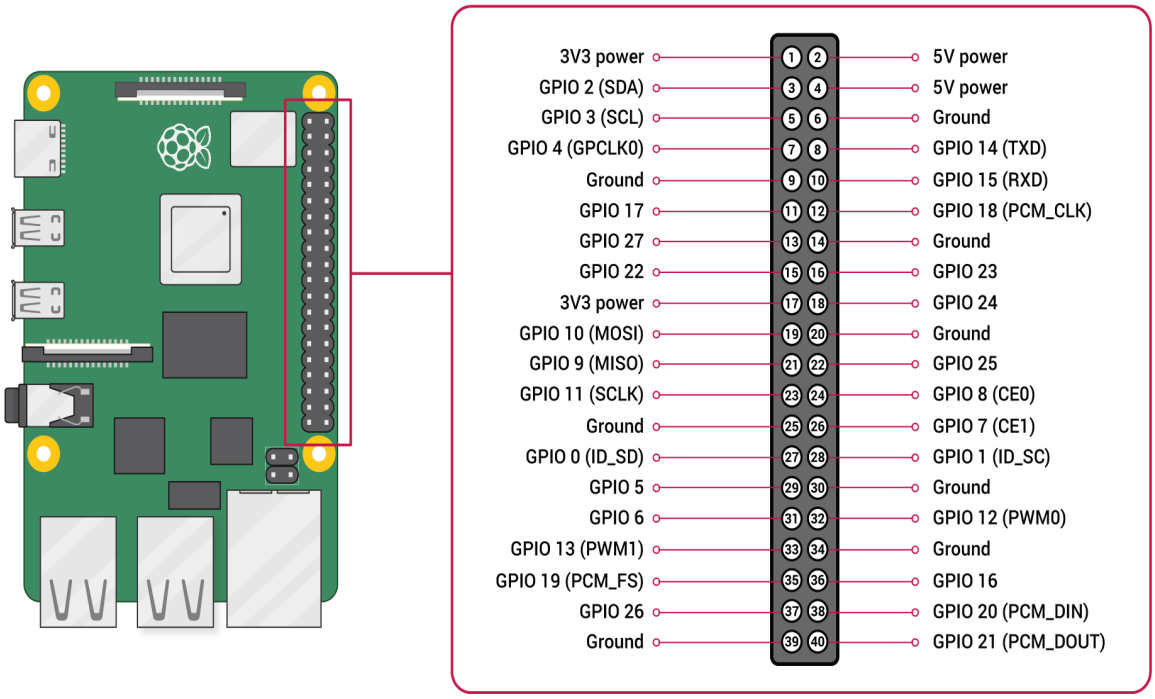


Figure: Raspberry Pi Pin Description

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

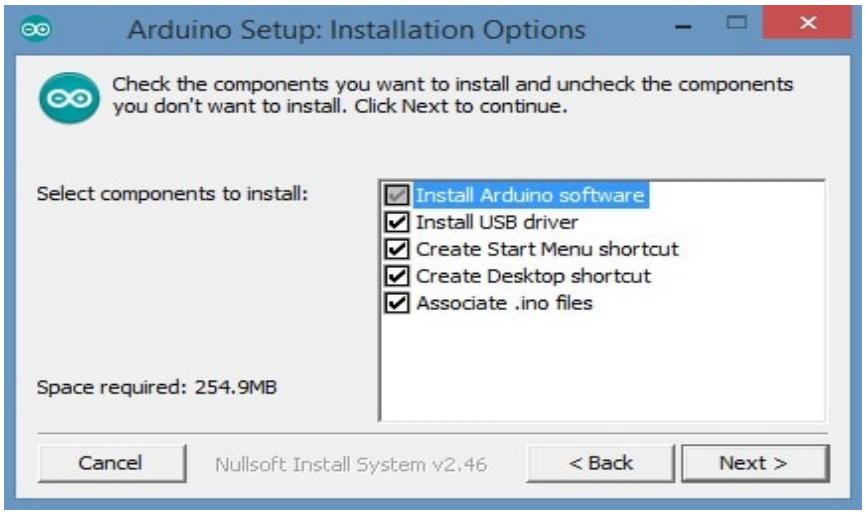
1. Visit http://www.arduino.cc/en/main/software to download the latest Arduino IDE version for your computer's operating system. There are versions for Windows, Mac, and Linux systems. At the download page, click on the "Windows Installer" option for the easiest installation.

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

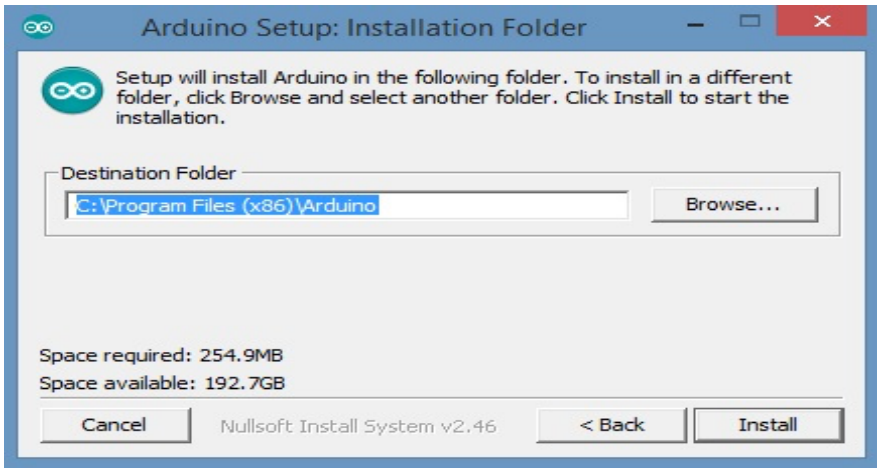
3. Open the .exe file.

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

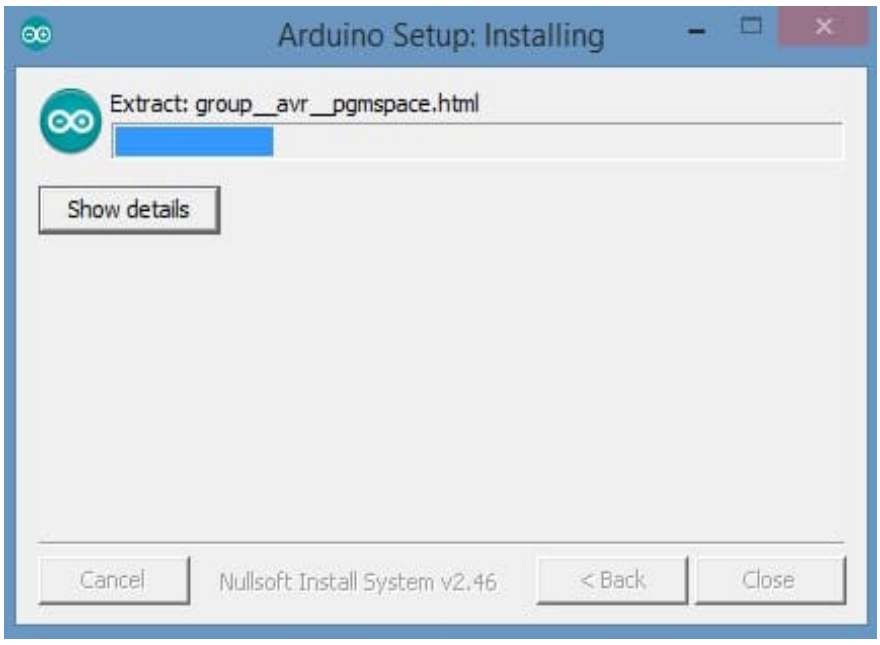
5. Decide which components to install, then click "Next":



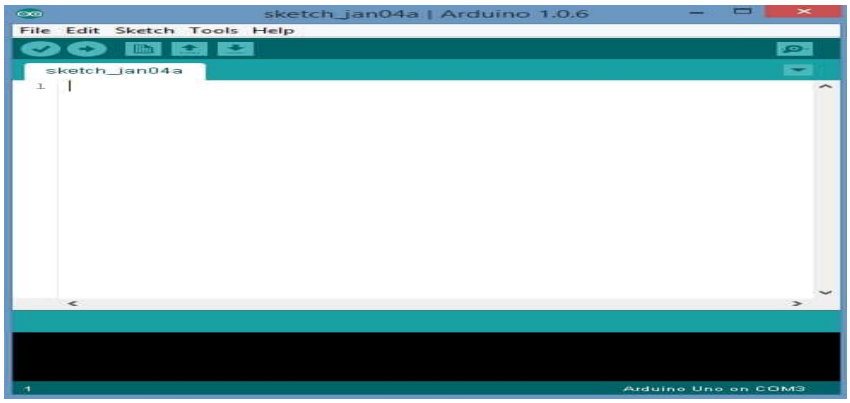
6. Select which folder to install the program to, then click “Install”:



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

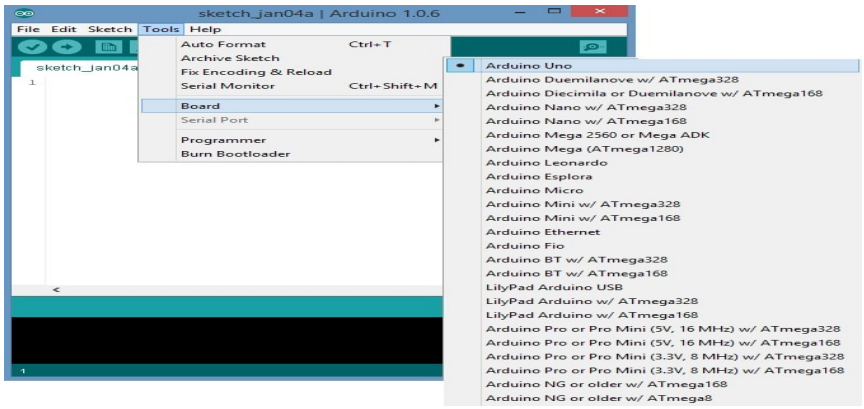


8. Now find the Arduino shortcut on your Desktop and click on it. The IDE will open up and you’ll see the code editor:



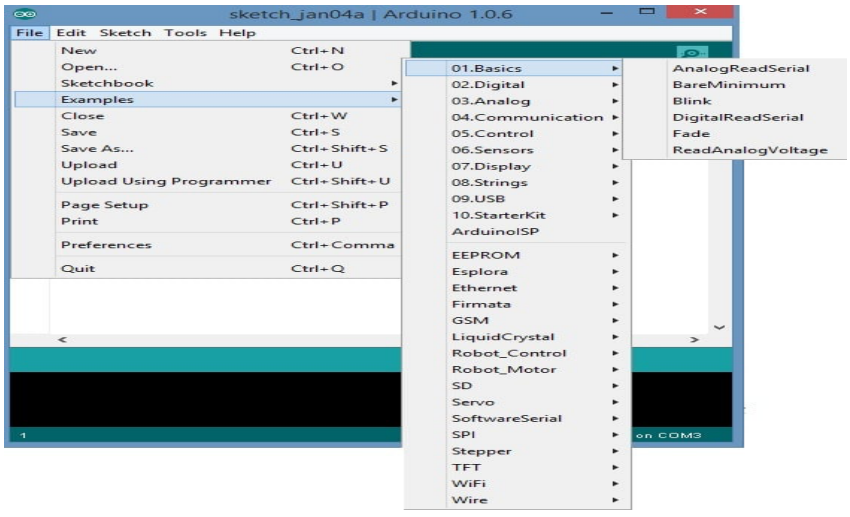
CONFIGURING THE ARDUINO IDE

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.

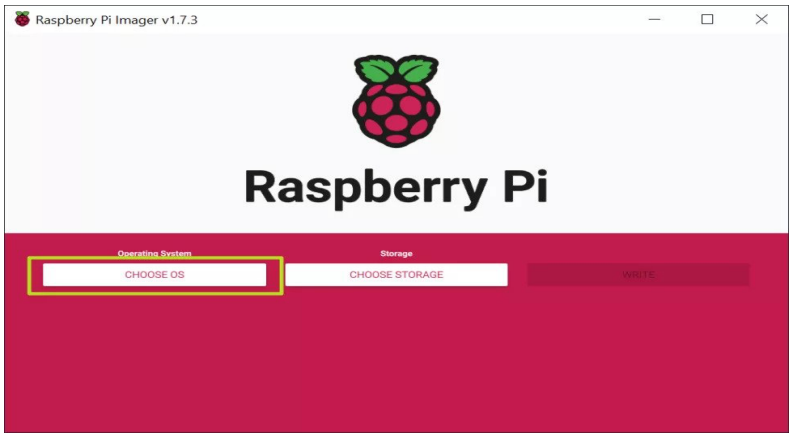
Downloading and Installing Raspberry Pi OS

Once you have all the components you need, use the following steps to create the boot disk you will need to set up your Raspberry Pi. These steps should work on a using a Windows, Mac or Linux-based PC (we tried this on Windows, but it should be the same on all three).

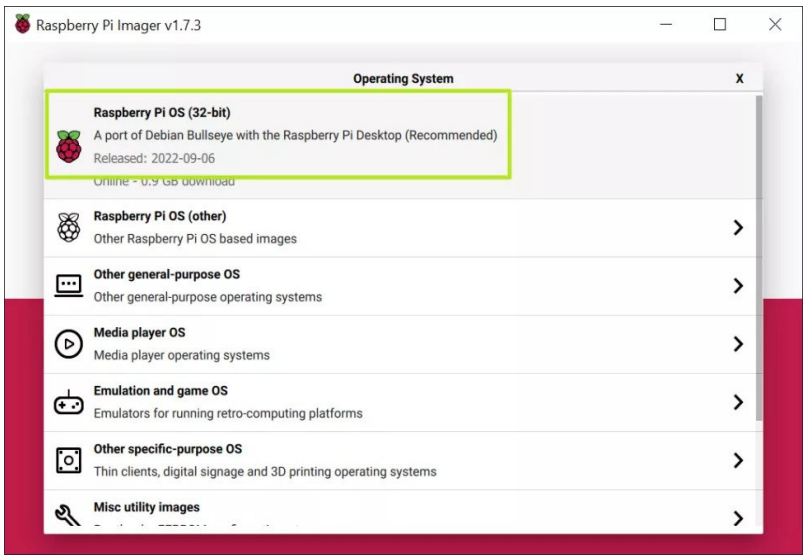
1. **Insert a microSD card/reader** into your computer.

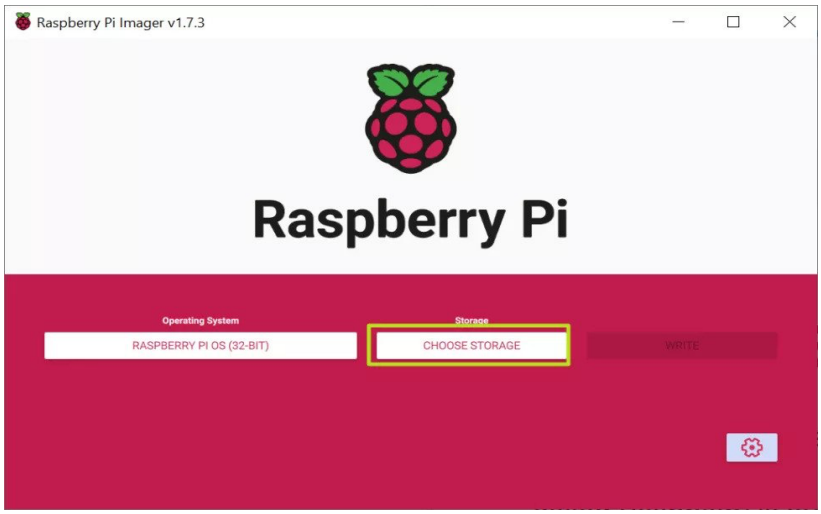
2. **Download and install the official Raspberry Pi Imager.** Available for Windows, macOS or Linux, this app will both download and install the latest Raspberry Pi OS. There are other ways to do this, namely by downloading a Raspberry Pi OS image file and then using a third-party app to "burn it," but the Imager makes it easier.

3. **Click Choose OS.**



4. **Select Raspberry Pi OS (32-bit)** from the OS menu (there are other choices, but for most uses, 32-bit is the best) and **Click Choose storage** and **pick the SD card** you're using.

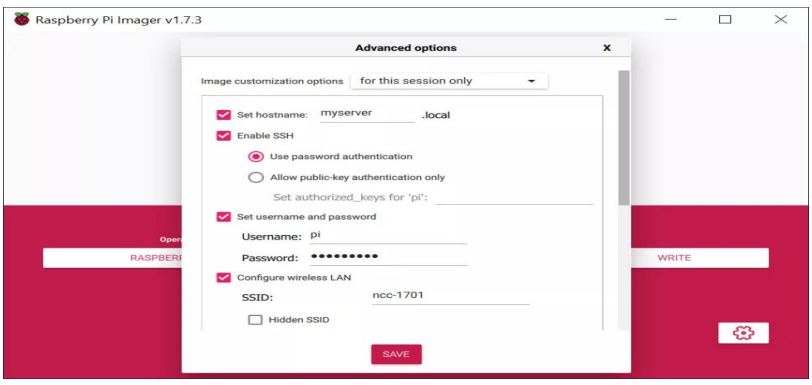




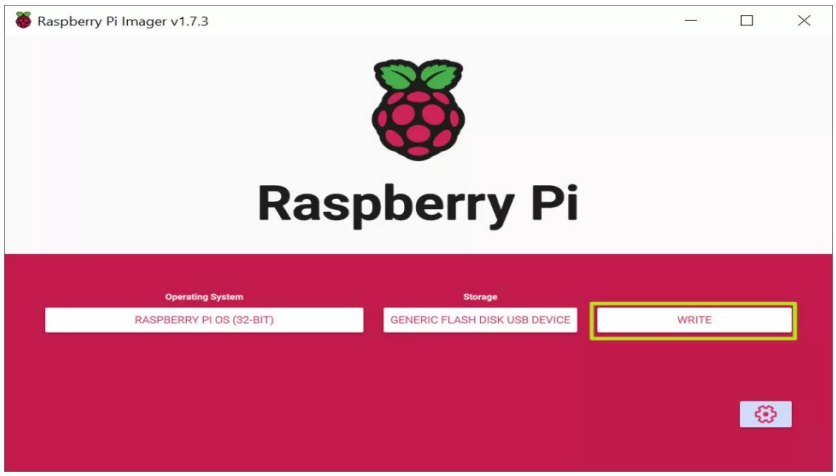
5. **Click the settings button** or hit CTRL + SHIFT + X to enter settings.

6. **Fill in settings fields** as follows and then **hit Save**. All of these fields are technically optional, but highly recommended so that can get your Raspberry Pi set up and online as soon as you boot it. If you don't set a username and password here, you'll have to go through a setup wizard that asks you to create them on first boot.

- **Set hostname**: the name of your Pi. It could be "raspberrypi" or anything you like.
- **Enable SSH**: Allow SSH connections to the Pi. Recommended.
- **Use password authentication / public key:** method of logging in via SSH
- **Set username and password:** Pick the username and password you'll use for the Pi
- **Configure wireless LAN: s**et the SSID and password of Wi-FI network
- **Wireless LAN country:** If you're setting up Wi-Fi, you must choose this.
- **Set locale settings:** Configure keyboard layout and time zone (probably chosen correctly by fault)



7. **Click Write.** The app will now take a few minutes to download the OS and write to your card.



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: Identification of different sensors used in IoT applications.

Objectives:

1. To study hardware related to IoT.

2. To understand and identify different sensors used in IoT.

Sensors:
The sensors are defined as a machine, module, or a device that detect changes in the environment. The sensors transfer those changes to the electronic devices in the form of a signal. A sensor and electronic devices always work together. The output signal is easily readable by humans. Nowadays, Sensors are used in daily lives. For example, controlling the brightness of the lamp by touching its base, etc. The use of sensors is expanding with new technologies. The sensor is a device, which is made up of Single Crystal Silicon. It is considered as a widely used semiconductor material. It has superior mechanical stability, machinability, etc. It can also combine electronics and sensing elements on the same substrate. The sensors are used to measure the physical quantities, such as pressure, temperature, sound, humidity, and light, etc. An example of sensors is Fire Alarm; a detector present on the fire alarm detects the smoke or heat. The signal generated from the detector is sent to the alarming system, which produces an alert in the form of alarm. The types of detectors are smoke detectors, heat detectors, carbon monoxide detectors, multi-sensors detectors, etc.

TABLE 1. IOT APPLICATIONS AND TYPES OF SENSORS USED IN IT

IoT Applications	Type of Sensors
Smart City	Velocity, Light, Accelerometer, Position, Temperature, Proximity, Humidity, Pressure, Infrared
Smart Environment	Light, Temperature, Humidity, Chemical, Gyroscope, Bio Sensors, Chemicals, Accelerometer, Optical
Smart Water	Temperature, Humidity, Occupancy, Water Quality
Smart Building	Light, Accelerometer, Chemical, Gyroscope, Magneto
Smart Health	Light, Gyroscope, Biosensors, Chemicals, Magneto, Accelerometer, Pressure
Smart Home	Light, Gyroscope, Biosensors, Chemicals, Magneto, Accelerometer, Temperature, Proximity, Position, Infrared
Smart Transport	Gyroscope, Pressure, chemicals, Magneto, Accelerometer, Temperature, Motion, Infrared
Smart Security	Light, Gyroscope, Chemical, Magneto, Accelerometer, Temperature, Infrared
Smart Agriculture	Temperature, Humidity, Water Quality, Chemical, Proximity, Position

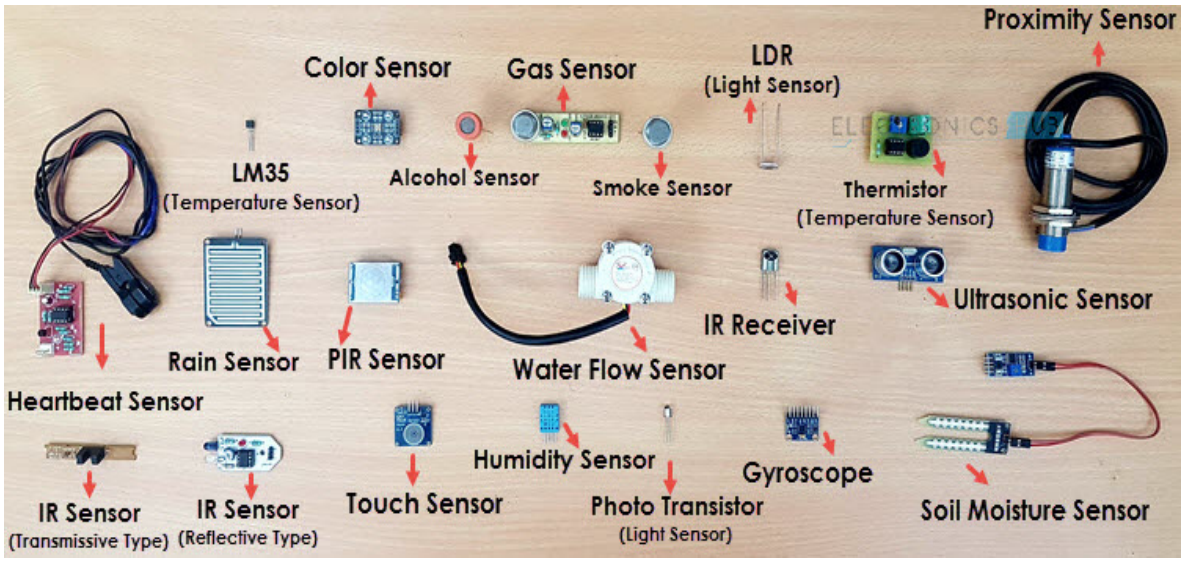


Figure: Different type of Sensors

1. Temperature Sensors

By definition, "A device, used to measure amount of heat energy that allows to detect a physical change in temperature from a particular source and converts the data for a device or user, is known as a Temperature Sensor." These sensors have been deployed for a long time in a variety of devices. However, with the emergence of IoT, they have found more room to be present in an even greater number of devices. Only a couple of years ago, their uses mostly included A/C control, refrigerators and similar devices used for environmental control. However, with the advent of the IoT world, they have found their role in manufacturing processes, agriculture and health industry. In the manufacturing process, many machines require specific environment temperature, as well as device temperature. With this kind of measurement, the manufacturing process can always remain optimal. On the other hand, in agriculture, the temperature of soil is crucial for crop growth. This helps with the production of plants, maximizing the output.

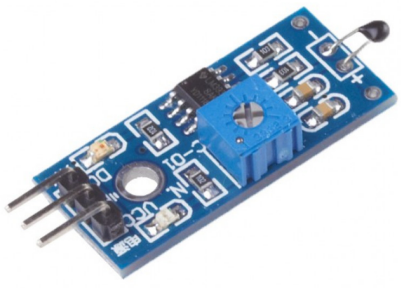


Figure: Temperature Sensor

Followed are some sub-categories of Temperature Sensors:

- Thermocouples: These are voltage devices that indicate temperature measuring with a change in voltage. As temperature goes up, the output voltage of the thermocouple rises.

• Resistor temperature detectors (RTD): The resistance of the device is directly proportional to the temperature, increase in a positive direction when the temperature rises resistance going up.
• Thermistors: It is a temperature sensitive resistor that changes its physical resistance with the change in temperature.
• IC (Semiconductor): They are linear devices where the conductivity of the semiconductor increases linearly and it takes advantage of the variable resistance properties of semiconductor materials. It can provide a direct temperature reading in digital form, especially at low temperatures.
• Infrared sensors: It detects temperature by intercepting a portion of emitted infrared energy of the object or substance, and sensing its intensity, can be used to measure temperature of solids and liquids only, not possible to use it on gases because of their transparent nature.

2. Proximity Sensor

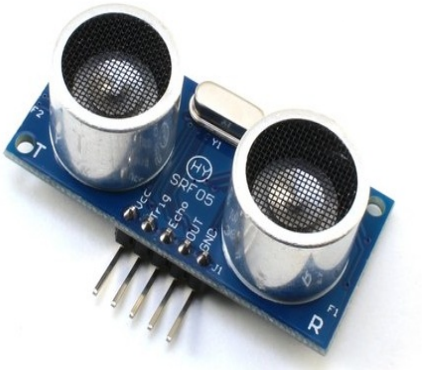


Figure: Proximity Sensor

A device that detects the presence or absence of a nearby object, or properties of that object, and converts it into signal which can be easily read by user or a simple electronic instrument without getting in contact with them. Proximity sensors are largely used in the retail industry, as they can detect motion and the correlation between the customer and product they might be interested in.

A user is immediately notified of discounts and special offers of nearby products. Another big and quite an old use-case are vehicles. You are reversing your car and are alarmed about an obstacle while taking reverse, that's the work of proximity sensor.

They are also used for parking availability in places such as malls, stadiums or airports.

Following are some of the Proximity Sensors sub-categorized:

• Inductive Sensors: Inductive proximity sensors are used for non-contact detection to find out the presence of metallic objects using electromagnetic field or a beam of electromagnetic radiation. It can operate at higher speeds than mechanical switches and also seems more reliable because of its robustness.
• Capacitive Sensors: Capacitive proximity sensors can detect both metallic as well as non-metallic targets. Nearly all other materials are dielectric different from air. It can be used to sense very small objects through a large portion of target. So, generally used in difficult and complicated applications.

• Photoelectric Sensors: Photoelectric sensor is made up of light-sensitive parts and uses a beam of light to detect the presence or absence of an object. It is an ideal alternative of inductive sensors. And used for long distance sensing or to sense non-metal object.
• Ultrasonic Sensors: Ultrasonic sensors are also used to detect the presence or to measure the distance of targets similar to radar or sonar. This makes a reliable solution for harsh and demanding conditions.

2. Pressure sensor

Figure: Pressure Sensor

A pressure sensor is a device that senses pressure and converts it into an electric signal. Here, the amount depends upon the level of pressure applied. There are plenty of devices that rely on liquid or other forms of pressure. These sensors make it possible to create IoT systems that monitor systems and devices that are pressure propelled. With any deviation from standard pressure range, the device notifies the system administrator about any problems that should be fixed. Deployment of these sensors is not only very useful in manufacturing, but also in the maintenance of whole water systems and heating systems, as it is easy to detect any fluctuation or drops in pressure.

4. Water Quality Sensor

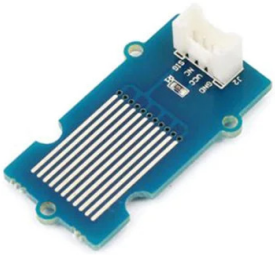


Figure: Water Sensor

Water quality sensors are used to detect the water quality and Ion monitoring primarily in water distribution systems. Water is practically used everywhere. These sensors play an important role as they monitor the quality of water for different purposes. They are used in a variety of industries.

Following is a list of the most common kind of water sensors in use:

- Chlorine Residual Sensor: It measures chlorine residual (i.e. free chlorine, mono chloramine & total chlorine) in water and most widely used as disinfectant because of its efficiency.
- Total Organic Carbon Sensor: TOC sensor is used to measure organic element in water.
- Turbidity Sensor: Turbidity sensors measure suspended solids in water; typically it is used in river and stream gaging, wastewater and effluent measurement.
- Conductivity Sensor: Conductivity measurements are carried out in industrial processes primarily to obtain information on total ionic concentrations (i.e. dissolved compounds) in water solutions.
- pH Sensor: It is used to measure the pH level in the dissolved water, which indicates how acidic or basic (alkaline) it is.
- Oxygen-Reduction Potential Sensor: The ORP measurement provides insights into the level of oxidation/reduction reactions occurring in the solution.

5. Chemical Sensor

Chemical sensors are applied in a number of different industries. Their goal is to indicate changes in liquid or to find out air chemical changes. They play an important role in bigger cities, where it is necessary to track changes and protect the population. Main use cases of chemical sensors can be found in Industrial environmental monitoring and process control, intentionally or accidentally released harmful chemical detection, explosive and radioactive detection, recycling processes on Space Station, pharma industries and laboratory etc.

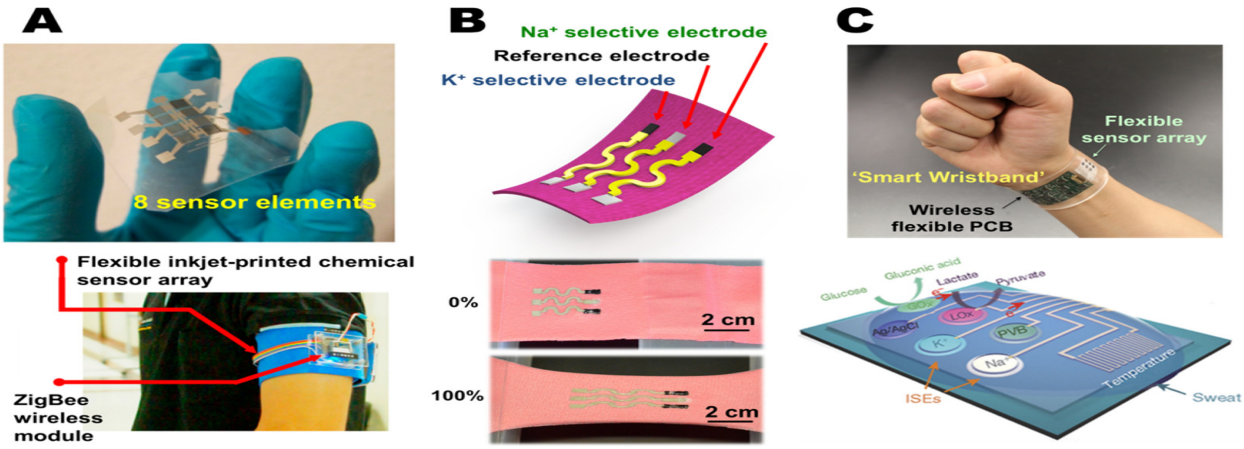


Figure: Chemical Sensors

Following are most common kind of chemical sensors in use:

- Chemical field-effect transistor
- Chemiresistor
- Electrochemical gas sensor
- Fluorescent chloride sensor
- Hydrogen sulfide sensor
- Nondispersive infrared sensor
- pH glass electrode
- Potentiometric sensor
- Zinc oxide nanorod sensor

6. Gas Sensor

Figure: Gas Sensors

Gas sensors are similar to the chemical ones, but are specifically used to monitor changes of the air quality and detect the presence of various gases. Like chemical sensors, they are used in numerous industries such as manufacturing, agriculture and health and used for air quality monitoring, detection of toxic or combustible gas, hazardous gas monitoring in coal mines, oil & gas industries, chemical laboratory research, manufacturing – paints, plastics, rubber, pharmaceutical & petrochemical etc.

Following are some common Gas sensors:

- Carbon dioxide sensor
- Breathalyzer
- Carbon monoxide detector
- Catalytic bead sensor
- Hydrogen sensor
- Air pollution sensor
- Nitrogen oxide sensor
- Oxygen sensor
- Ozone monitor
- Electrochemical gas sensor
- Gas detector
- Hygrometer

7. Smoke sensor

Figure: Smoke Sensor

A smoke sensor is a device that senses smoke (airborne particulates & gases), and it's level.

They have been in use for a long period of time. However, with the development of IoT, they are now even more effective, as they are plugged into a system that immediately notifies the user about any problem that occurs in different industries.

Smoke sensors are extensively used by the manufacturing industry, HVAC, buildings, and accommodation infra to detect fire and gas incidences. This serves to protect people working in dangerous environments, as the whole system is much more effective in comparison to the older ones.

Common Types of Smoke Sensors
Smoke sensors detect the presence of Smoke, Gases and Flame surrounding their field. It can be detected either optically or by the physical process or by the use of both the methods.

- Optical smoke sensor (Photoelectric): Optical smoke sensor used the light scatter principle trigger to occupants.
- Ionization smoke sensor: Ionization smoke sensor works on the principle of ionization, kind of chemistry to detect molecules causing a trigger alarm.

8. IR Sensors

Figure: IR Sensors

An infrared sensor is a sensor that is used to sense certain characteristics of its surroundings by either emitting or detecting infrared radiation. It is also capable of measuring the heat being emitted by objects. They are now used in a variety of IoT projects built by IoT development companies, especially in Healthcare as they make monitoring blood flow and blood pressure simple. They are even used in a wide array of regular smart devices such as smart watches and smartphones as well.

Other common use includes home appliances & remote control, breath analysis, Infrared vision (i.e. visualize heat leaks in electronics, monitor blood flow, art historians to see under layers of paint), wearable electronics, optical communication, non-contact based temperature measurements, automotive blind-angle detection. Their usage does not end there; they are also a great tool for ensuring high-level security in your home. Also, their application includes environment checks, as they can detect a variety of chemicals and heat leaks. They are going to play an important role in the smart home industry, as they have a wide range of applications.

9. Level Sensors

A sensor, which is used to determine the level or amount of fluids, liquids or other substances that flow in an open or closed system, is called Level sensor.

Like IR sensors, level sensors are present in a wide array of industries. They are primarily known for measuring fuel levels, but they are also used in businesses that work with liquid materials. For example, the recycling industry, as well as the juice and alcohol industry relies on these sensors to measure the number of liquid assets in their possession.

The best use cases of the level sensor are fuel gauging & liquid levels in open or closed containers, sea level monitoring & Tsunami warning, water reservoirs, medical equipment, compressors, hydraulic reservoirs, machine tools, beverage and pharmaceutical processing, high or low-level detection, etc.

Figure: Level Sensors

This helps better streamline their businesses, as sensors collect all the important data at all times. With the use of these sensors, any product manager can precisely see how much liquid is ready to be distributed and whether the manufacturing should be stepped up.

There are two basic level measurement types:

- Point level sensors: Point level sensors usually detect the particular specific level and respond to the user if the sensing object is above or below that level. It is integrated into single device to get an alarm or trigger
- Continuous level Sensor: Continuous level sensors measure liquid or dry material levels within a specified range and provide outputs, which continuously indicate the level. The best example of it is fuel level display in the vehicle.

10. Image Sensors

Image sensors are instruments, which are used to convert optical images into electronic signals for displaying or storing files electronically. The major use of image sensors is found in digital cameras & modules, medical imaging and night vision equipment, thermal imaging devices, radar, sonar, media houses, and Biometric & IRIS devices.

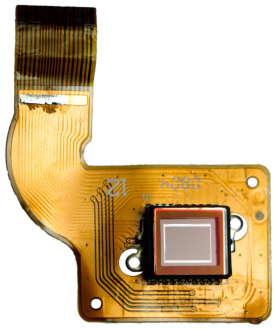


Figure: Image Sensor

Two main types of sensors are used in:

- CCD (charge-coupled device), and
- CMOS (complementary metal-oxide semiconductor) imagers.

Although each type of sensor uses different technology to capture images, both CCD and CMOS imagers use metal-oxide semiconductors, having the same degree of sensitivity to light, and no inherent quality difference.

An average consumer would think that this is a regular camera, but even though this is not far from the truth, image sensors are connected to a wide range of different devices, making their functionality much better.

One of the best-known uses includes the car industry, in which imagery plays a very important role and a automotive software development company would heavily rely on it to bring automation. With these sensors, the system can recognize signs, obstacles and many other things that a driver would generally notice on the road. They play a very important role in the IoT industry, as they directly affect the progress of driverless cars.

They are also implemented in improved security systems, where images help capture details about the perpetrator. In the retail industry, these sensors serve to collect data about customers, helping businesses get a better insight into who is actually visiting their store, race, gender, age are only some of the useful parameters that retail owners get by using these IoT sensors.

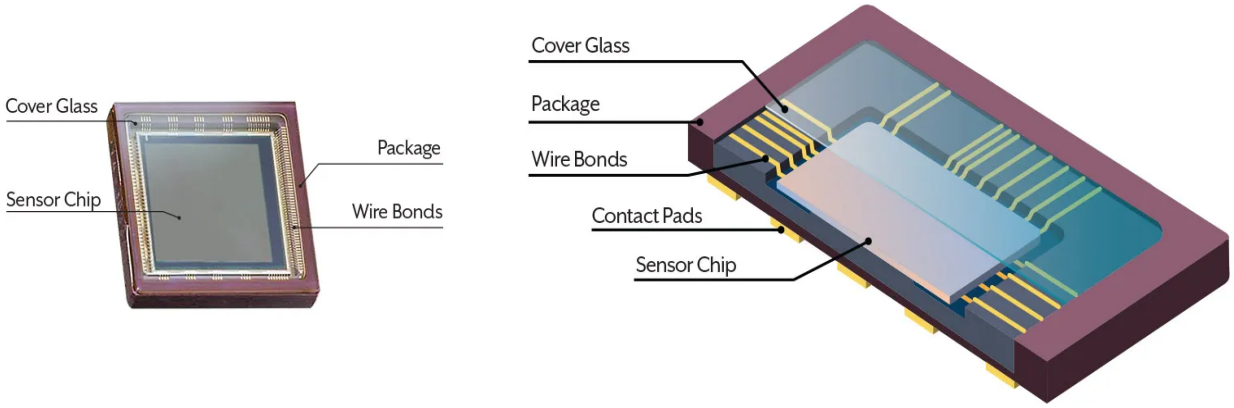


Figure: Description of Image Sensor

11. Motion Detection Sensors

A motion detector is an electronic device which is used to detect the physical movement (motion) in a given area and it transforms motion into an electric signal; motion of any object or motion of human beings

Motion detection plays an important role in the security industry. Businesses utilize these sensors in areas where no movement should be detected at all times, and it is easy to notice anybody's presence with these sensors installed.

A software development firm or IoT company will use these sensors primarily for intrusion detection systems, automatic door control, boom barrier, smart camera (i.e motion-based capture/video recording), toll plaza, automatic parking systems, automated sinks/toilet flusher, hand dryers, energy management systems (i.e. Automated Lighting, AC, Fan, Appliances Control), etc.

Figure: Motion Sensors

On the other hand, these sensors can also decipher different types of movements, making them useful in some industries where a customer can communicate with the system by waving a hand or by performing a similar action. For example, someone can wave to a sensor in the retail store to request assistance with making the right purchase decision.

Even though their primary use is correlated with the security industry, as technology advances, the number of possible applications of these sensors is only going to grow.

Following are key motion sensor types widely used:

- Passive Infrared (PIR): It Detects body heat (infrared energy) and the most widely used motion sensor in home security systems.
- Ultrasonic: Sends out pulses of ultrasonic waves and measures the reflection off a moving object by tracking the speed of sound waves.
- Microwave: Sends out radio wave pulses and measures the reflection off a moving object. They cover a larger area than infrared & ultrasonic sensors, but they are vulnerable to electrical interference and more expensive.

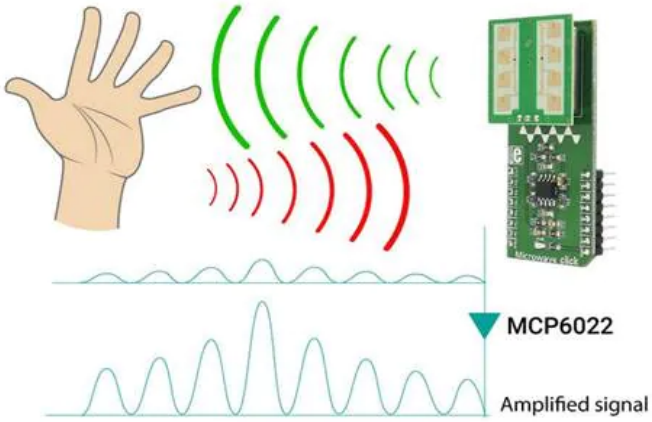


Figure: Application of Motion Sensor

12. Accelerometer Sensors

An accelerometer is a transducer that is used to measure the physical or measurable acceleration experienced by an object due to inertial forces and converts the mechanical motion into an electrical output. It is defined as rate of change of velocity with respect to time

These sensors are now present in millions of devices, such as smartphones. Their uses involve the detection of vibrations, tilting, and acceleration in general. This is great for monitoring your driving fleet, or using a smart pedometer.

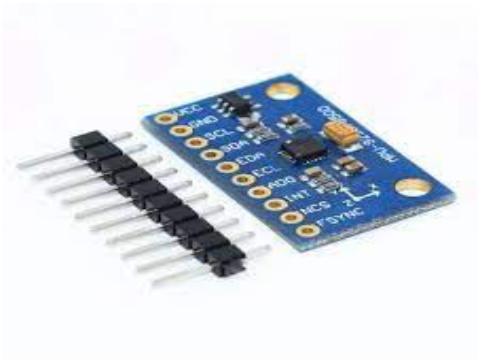


Figure: Accelerometer Sensor

In some instances, it is used as a form of anti-theft protection, as the sensor can send an alert through the system if an object that should remain stationary is moved.

They are widely used in cellular & media devices, vibration measurement, automotive control and detection, free fall detection, aircraft and aviation industries, movement detection, sports academy/athletes behavior monitoring, consumer electronics, industrial & construction sites etc.

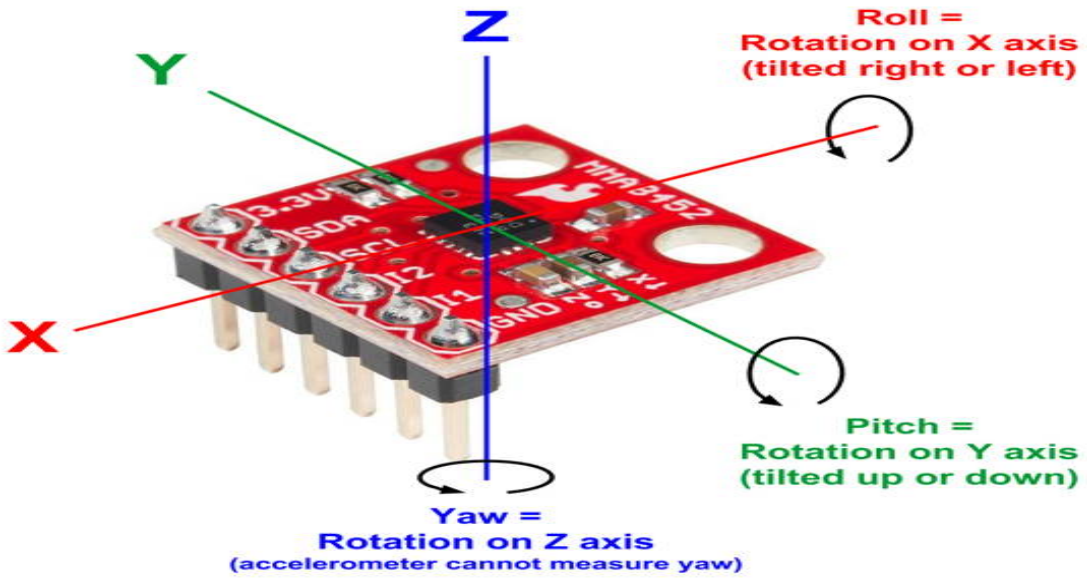


Figure: Axis Diagram of Accelerometer Sensor

There are various kinds of accelerometers and following are few mainly used in IoT projects:

- Hall-effect accelerometers: Hall-effect accelerometers are using Hall principle to measure the acceleration, it measures the voltage variations caused by changes in a magnetic field around them.
- Capacitive accelerometers: Capacitive accelerometers sensing output voltage dependents on the distance between two planar surfaces. Capacitive accelerometers are also less prone to noise and variation with temperature.
- Piezoelectric accelerometers: Piezoelectric sensing principle is working on the piezoelectric effect. Piezo-film based accelerometers are best used to measure vibration, shock, and pressure.

Each accelerometer sensing technology has its own advantages and compromises. Before selecting, it's important to understand the basic differences of the various types and the test requirements.

13. Gyroscope Sensors

A sensor or device, which is used to measure the angular rate or angular velocity, is known as Gyro sensors, Angular velocity is simply defined as a measurement of speed of rotation around an axis. It is a device used primarily for navigation and measurement of angular and rotational velocity in 3-axis directions. The most important application is monitoring the orientation of an object.

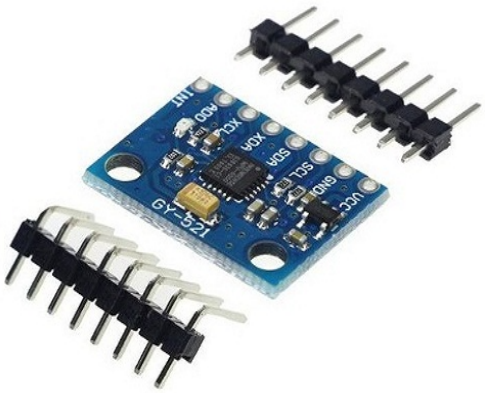


Figure: Gyroscope Sensors

Their main applications are in car navigation systems, game controllers, cellular & camera devices, consumer electronics, robotics control, drone & RC control helicopter or UAV control, vehicle control/ADAS and many more.

There are several different kinds of gyro sensors, which are selected by their working mechanism, output type, power, sensing range and environmental conditions.

- Rotary (classical) gyroscopes
- Vibrating Structure Gyroscope
- Optical Gyroscopes
- MEMS (micro-electro-mechanical systems) Gyroscopes

These sensors are always combined with accelerometers. The use of these two sensors simply provides more feedback to the system. With gyroscopic sensors installed, many devices can help athletes improve the efficiency of their movements, as they gain access to the athlete's movement during sports activities.

This is only one example of its application, however, as the role of this sensor is to detect rotation or twist, its application is crucial for the automation of some manufacturing processes.

14. Humidity Sensors

Humidity is defined as the amount of water vapor in an atmosphere of air or other gases. The most commonly used terms are "Relative Humidity (RH)

These sensors usually follow the use of temperature sensors, as many manufacturing processes require perfect working conditions. Through measuring humidity, you can ensure that the whole process runs smoothly, and when there is any sudden change, action can be taken immediately, as sensors detect the change almost instantaneously.

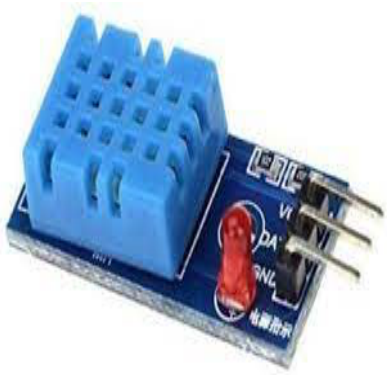


Figure: Humidity Sensor

Their applications and use can be found in Industrial & residential domain for heating, ventilating, and air conditioning systems control. They can also be found in Automotive, museums, industrial spaces and greenhouses, meteorology stations, paint and coatings industries, hospitals & pharma industries to protect medicines

15. Optical Sensors

A sensor, which measures the physical quantity of light rays and convert it into electrical signal that can be easily readable by user or an electronic instrument/device is called optical sensor.

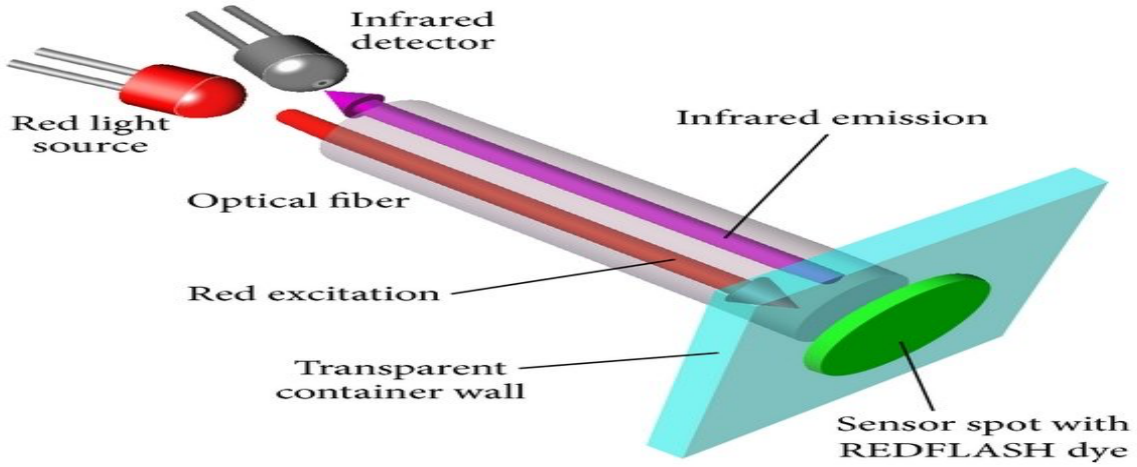


Figure: Optical Sensor

IoT experts love optical sensors, as they are practical for measuring different things simultaneously. The technology behind this sensor allows it to monitor electromagnetic energy, which includes, electricity, light and so on.

Due to this fact, these sensors have found use in healthcare, environment monitoring, energy, aerospace and many more industries. With their presence oil companies, pharmaceutical companies and mining companies are in a much better position to track environmental changes while keeping their employees safe.

Their main use can be found in ambient light detection, digital optical switches, optical fibers communications, due to electrical isolation best suited for oil and gas applications, civil and transportation fields, high speed network systems, elevator door control, assembly line part counters and safety systems.

Following are key type of optical sensors:

- Photo detector: It uses light sensitive semiconductor materials like photocells, photodiodes or phototransistors to work as photo detector
- Fiber Optics: Fibers optics carry no current, so its immune to electrical & electromagnetics interference and even in damaged condition no sparking or shock hazard happens.
- Pyrometer: It estimates the temperature of an object by sensing the color of the light and Objects radiate light according to their temperature and produce same colors at same temperature.
- Proximity & Infrared: Proximity use light to sense objects nearby and infrared are used

where visible light would be inconvenient.

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-3

Aim: Demonstration of Autodesk Tinkercad Simulation Platform.

Objective:

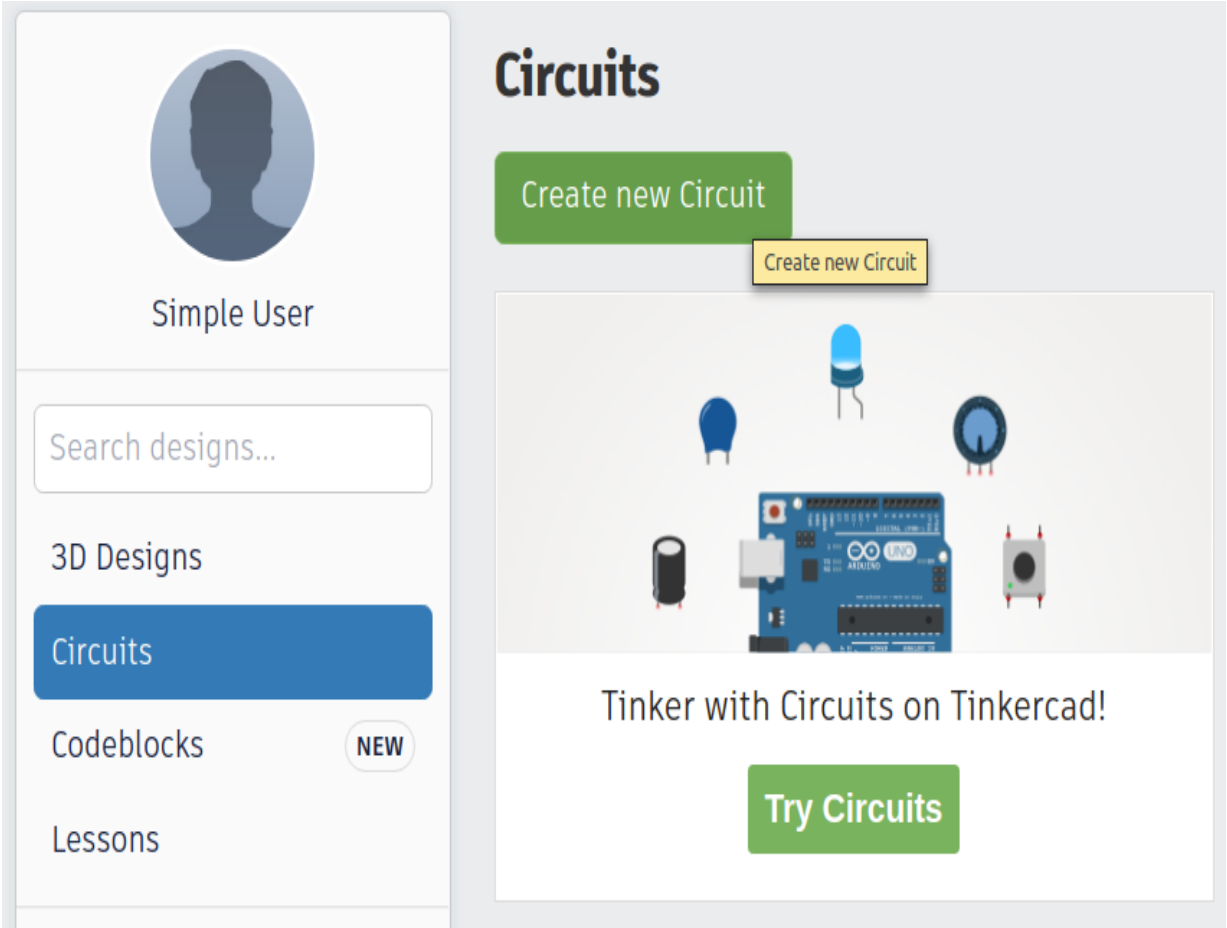
1. Learn about IoT based simulations.

2. Testing and model in IoT based simulation platform.

Introduction of Tinkercad

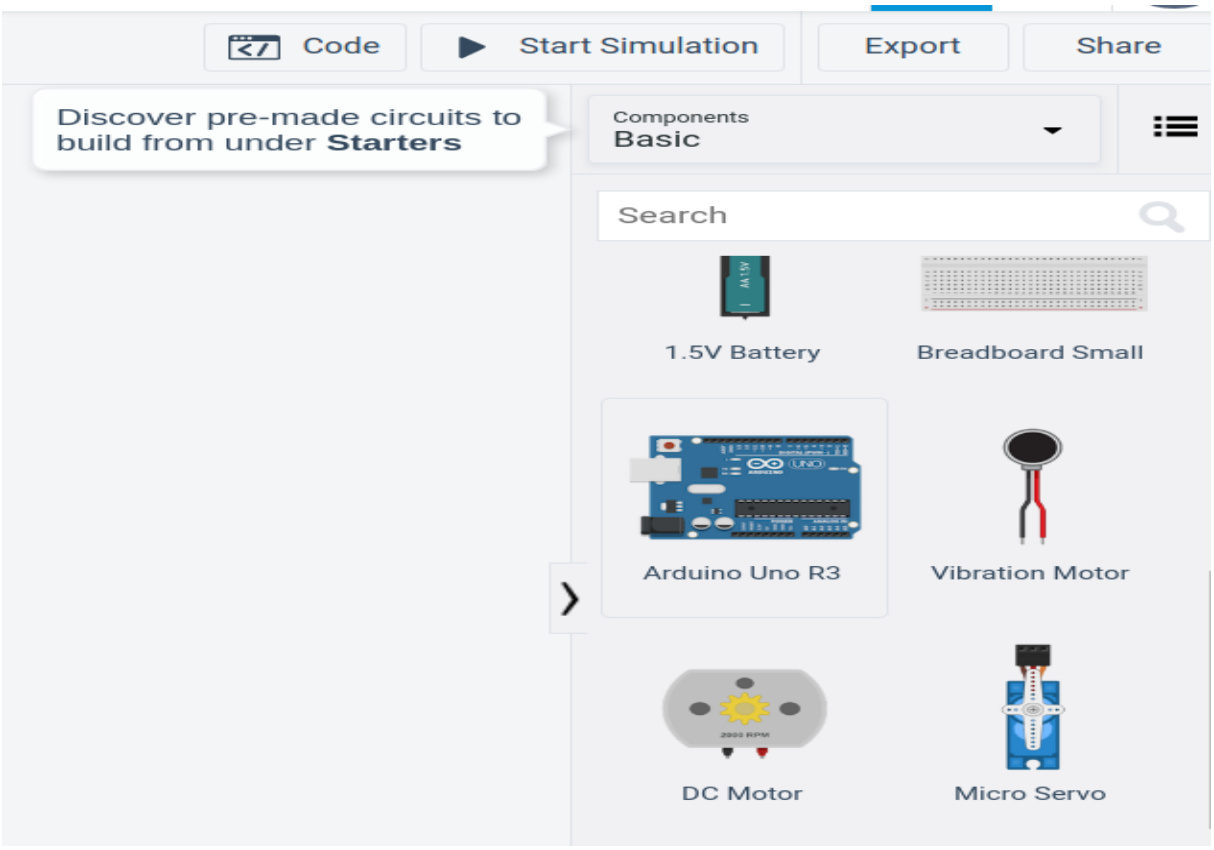
Tinkercad - https://www.tinkercad.com is an excellent tool that allows you to simulate Arduino-based systems (and a lot more). You can (perhaps you SHOULD) simulate all exercises and even your own designs before trying them on real hardware. It also allows you to do programming using blocks. You can 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.

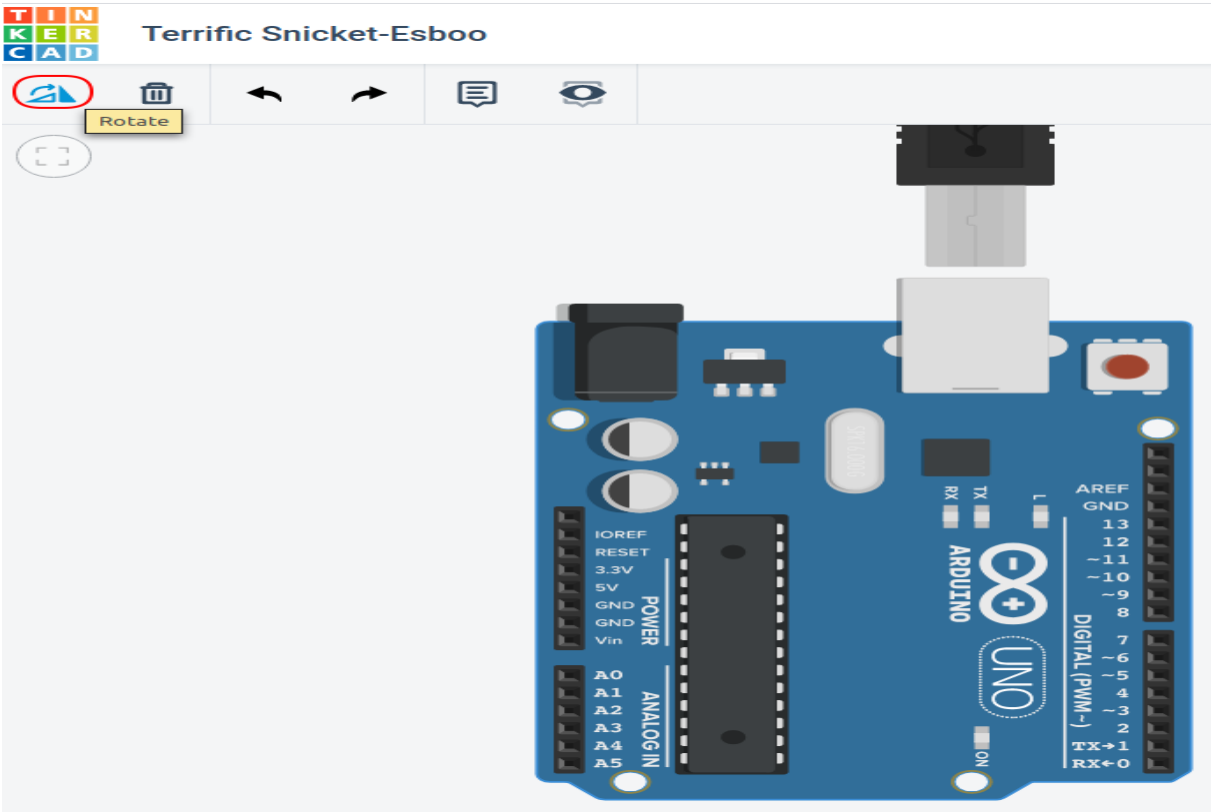


Hardware

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.



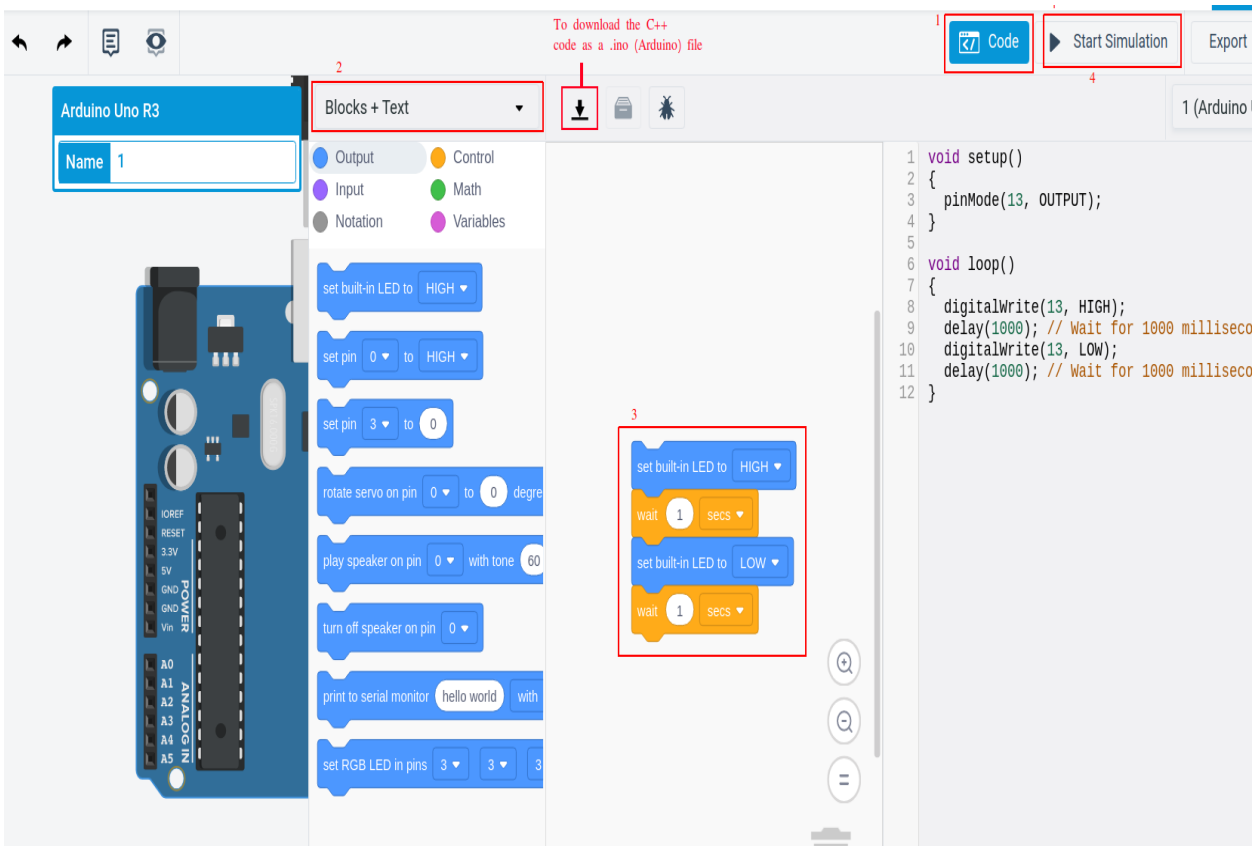
You can add more components and wire them up as desired. Clicking on the lead of a component allows you to start a connecting wire from there. Clicking on a wire allows you to change its color.

Programming and Simulation

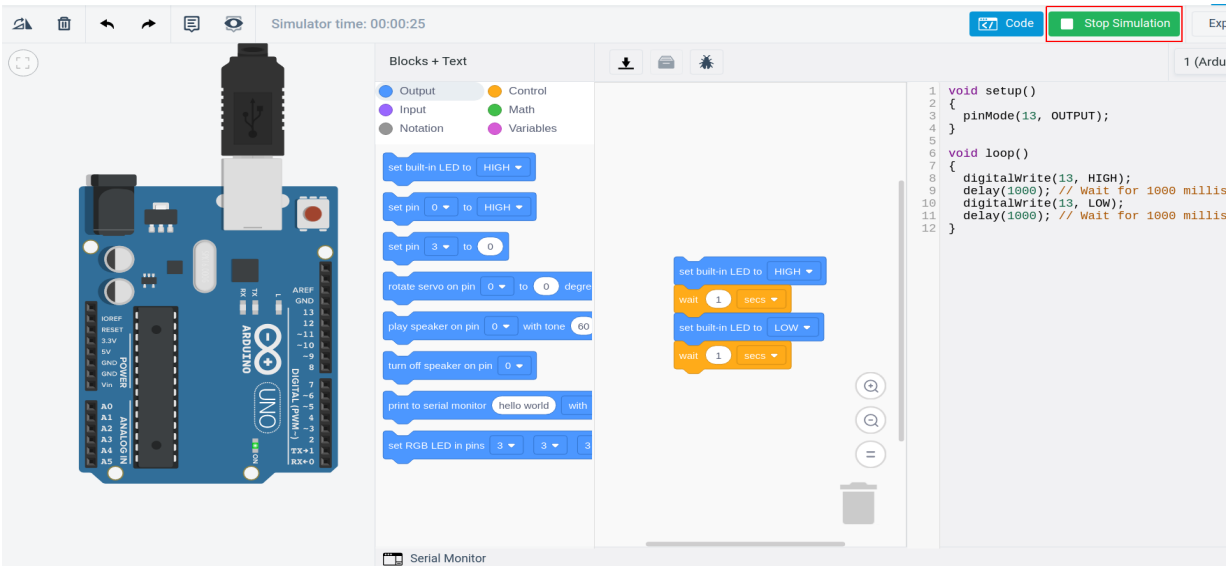
To program the Arduino,

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).

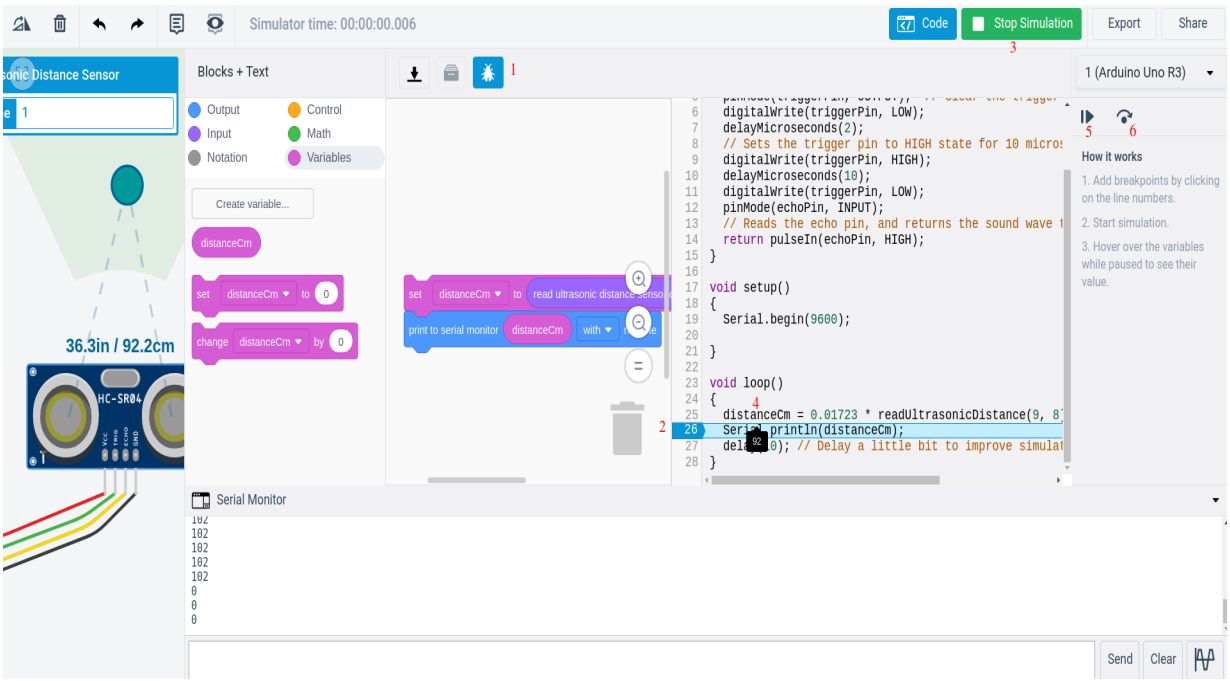


You have to click Stop Simulation to stop the simulation before you can modify your program and/or hardware connections.



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.

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 4

Aim: Program to interface the Arduino/Raspberry Pi with LED and blinking application.

Objectives:

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

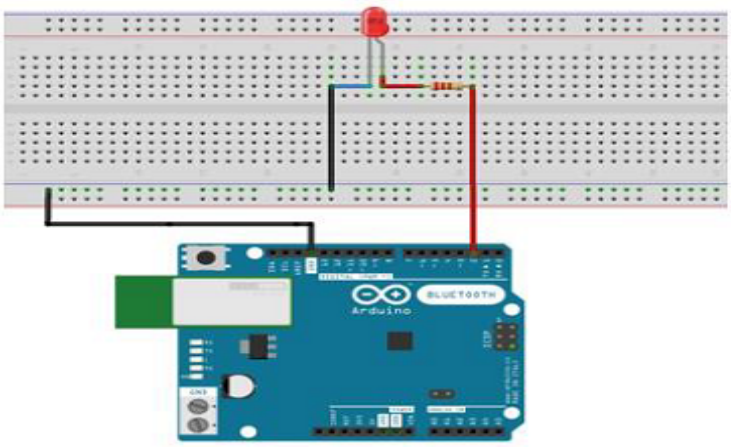
Components Required:

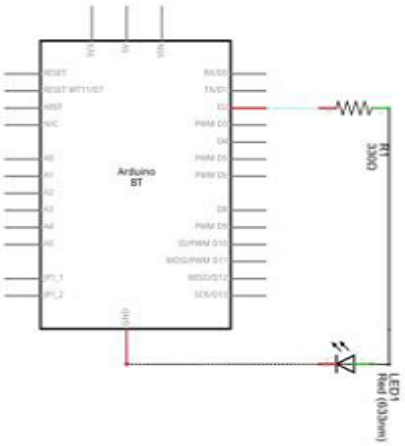
- 1 × Breadboard
- 1 × Arduino Uno R3
- 1 × LED
- 1 × 330Ω Resistor
- 2 × Jumper

Procedure

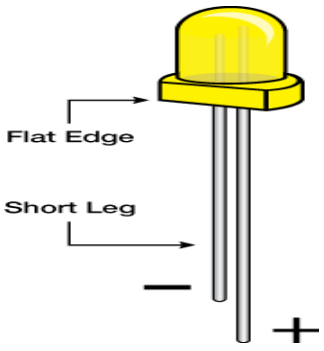
LEDs are small, powerful lights that are used in many different applications. To start, we will work on blinking an LED, the Hello World of microcontrollers. It is as simple as turning a light on and off. Establishing this important baseline will give you a solid foundation as we work towards experiments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

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

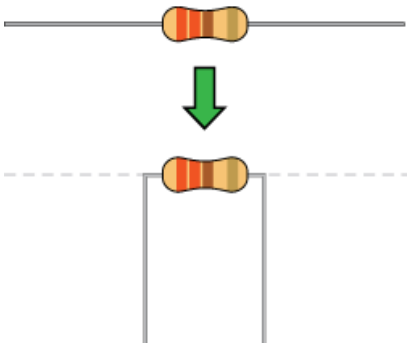




Note − To find out the polarity of an LED, look at it closely. The shorter of the two legs, towards the flat edge of the bulb indicates the negative terminal.

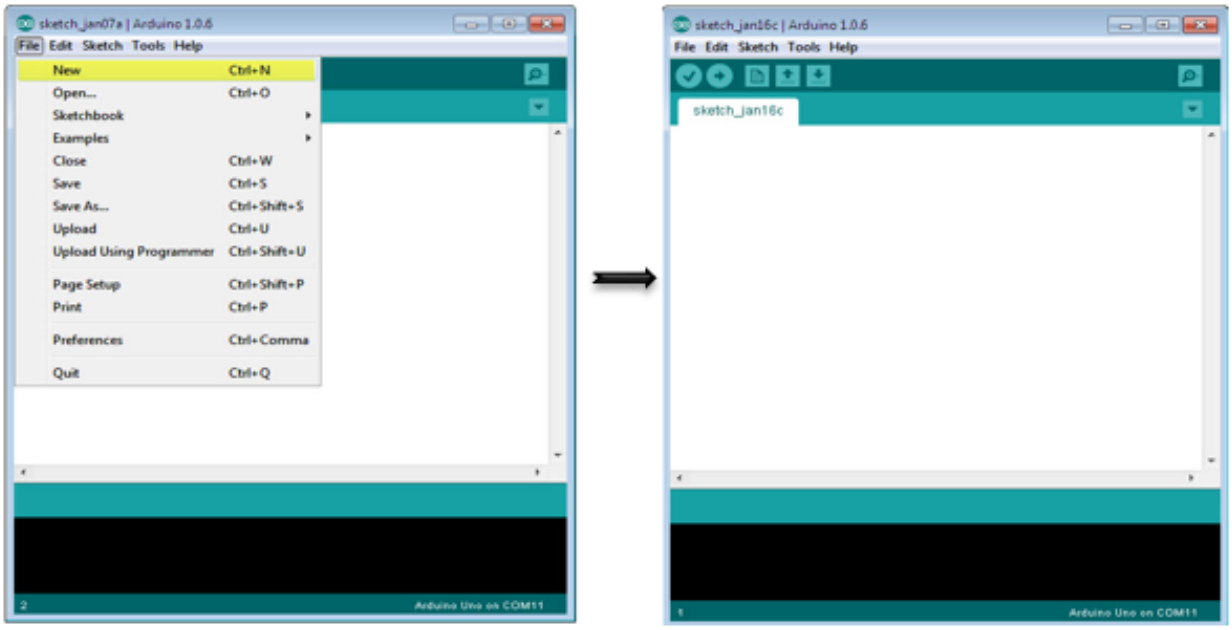


Components like resistors need to have their terminals bent into 90° angles in order to fit the breadboard sockets properly. You can also cut the terminals shorter.



Sketch

Open the Arduino IDE software on your computer. Coding in the Arduino language will control your circuit. Open the new sketch File by clicking New.



Arduino Code

```
/*
Blink    Turns on an LED on for one second, then off for one second, repeatedly.
*/
// the setup function runs once when you press reset or power the board
void setup() {  // initialize digital pin 13 as an output.
pinMode(2, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
digitalWrite(2, HIGH); // turn the LED on (HIGH is the voltage level)
delay(1000); // wait for a second
digitalWrite(2, LOW); // turn the LED off by making the voltage LOW    delay(1000); // wait for a second
}
```

Code to Note

pinMode(2, OUTPUT) − Before you can use one of Arduino's pins, you need to tell Arduino Uno R3 whether it is an INPUT or OUTPUT. We use a built-in "function" called pinMode() to do this.

digitalWrite(2, HIGH) − When you are using a pin as an OUTPUT, you can command it to be HIGH (output 5 volts), or LOW (output 0 volts).

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.

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 5

Aim: To measure the distance of an object using an ultrasonic sensor.

Components 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:

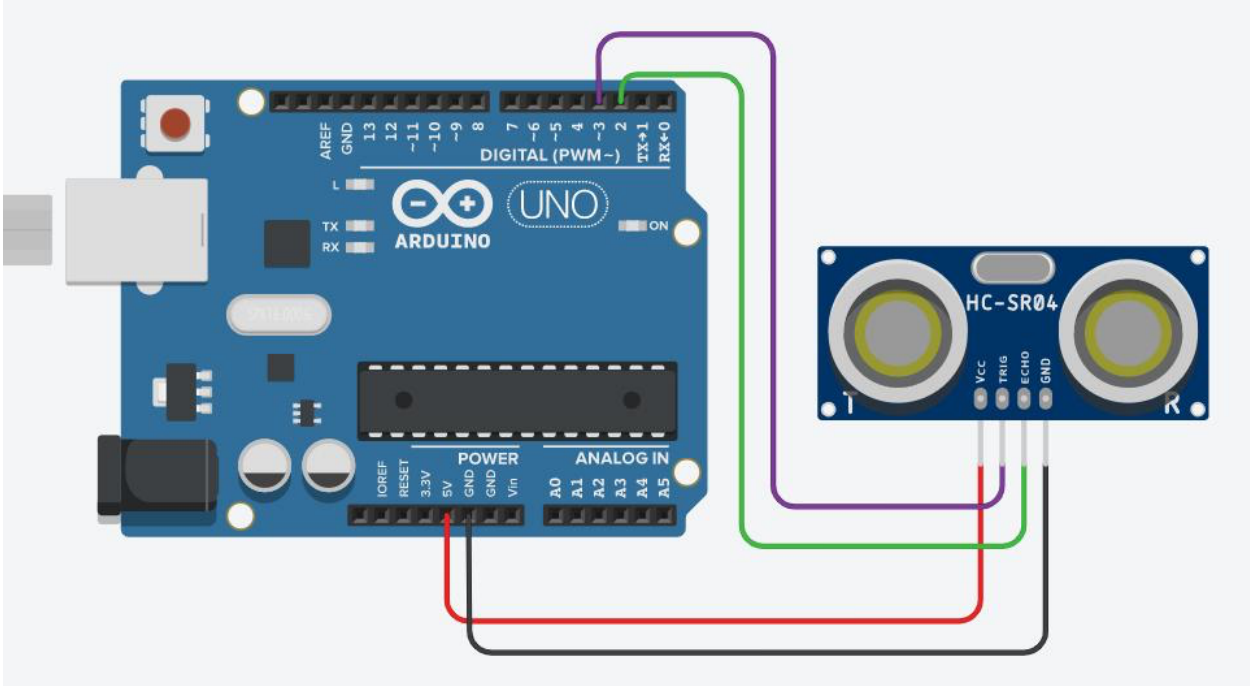


Figure: Ultrasonic Distance measurement circuit

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);
}
```

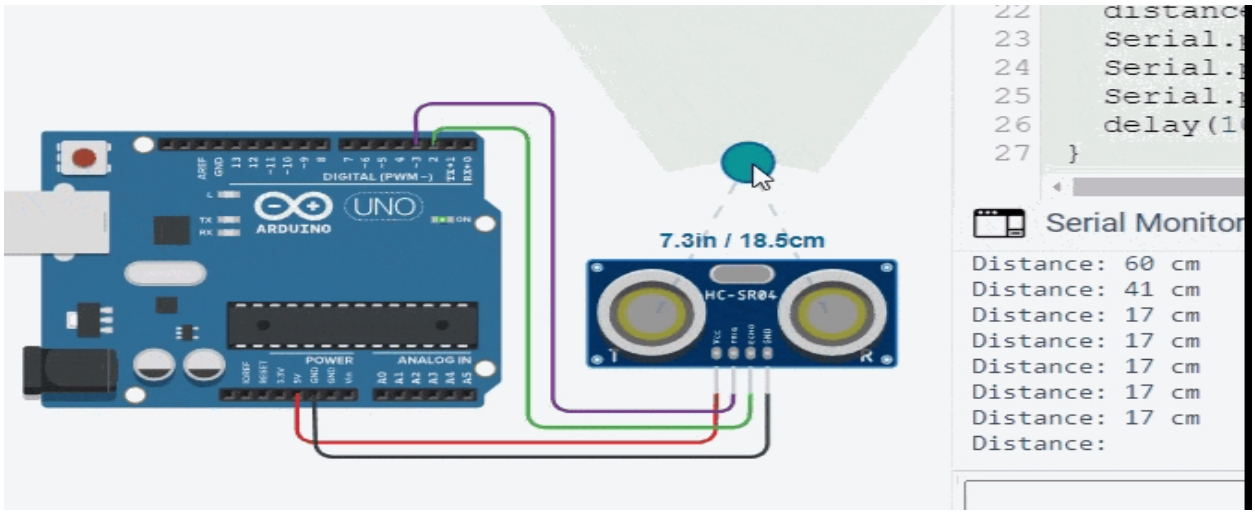


Figure: Distance measurement

Output in External Display device (LCD Display):

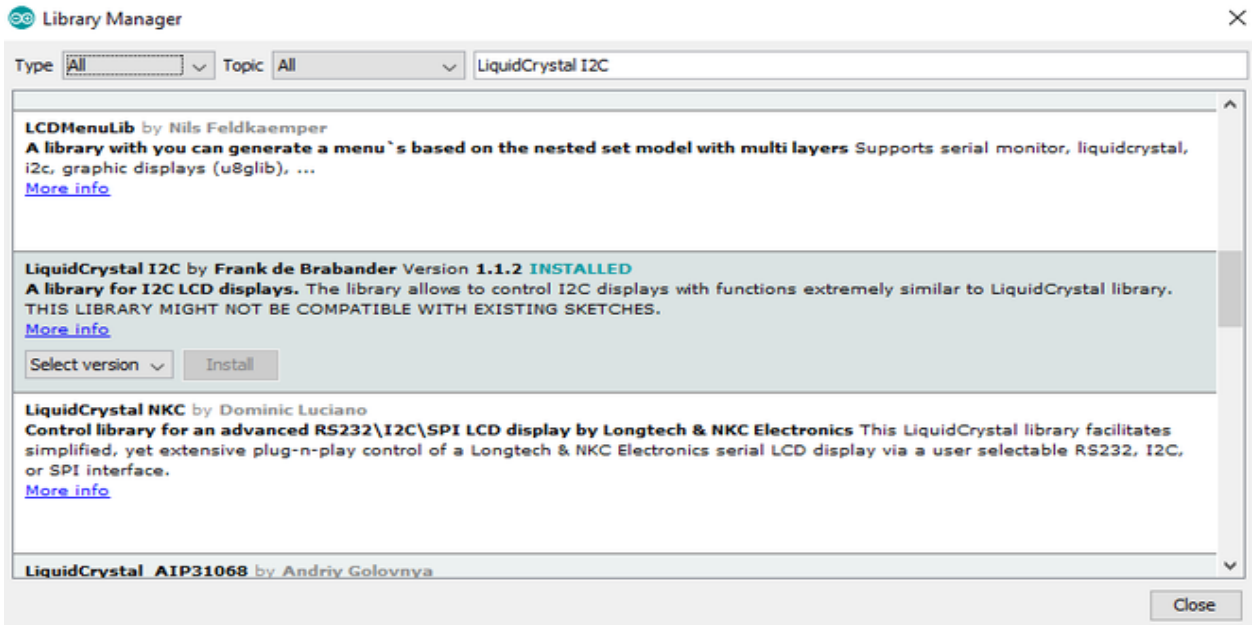
To monitor the output without a PC, we have to interface display devices to monitor the output. Here, we are using "a 16×2 LCD display with I2C communication.

To interface with the LCD display, we need to install the supporting library to the Arduino IDE.

Steps to Interface LCD display:

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.



Library Installation

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.

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C
   lcd(0x20, 16, 2); // Format => (ADDRESS,Width,Height )

void setup()
{

   lcd.init(); // Initialize the lcd
   lcd.backlight(); // Turn on the Backlight

   // ....
}
```

Note: For ADDRESS, Try “0x27” or “0x3F”. If those addresses are not working. Refer manufacturer page.

Or

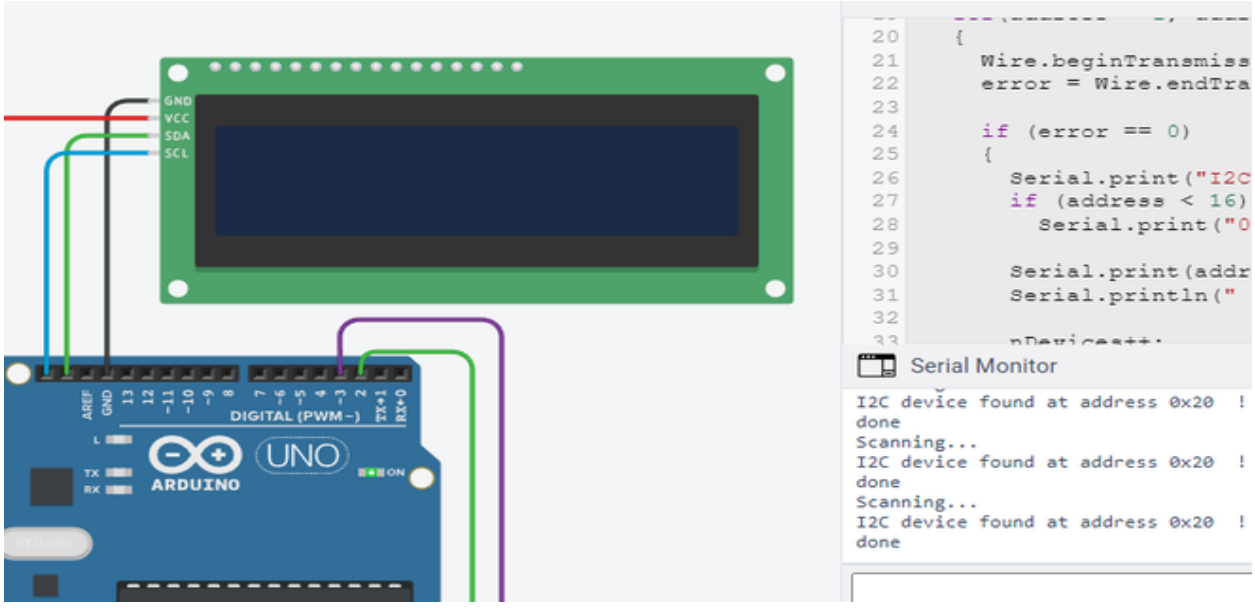
To find the address, Run the following code to find the address.

Arduino code for I2C device Address Finding:

```
// I2C address scanner program
#include <Wire.h>

void setup()
{
   Wire.begin();
   Serial.begin(9600);
   Serial.println("I2C Scanner");
}

void loop()
{
   byte error, address;
   int nDevices;

   Serial.println("Scanning...");
```

```
nDevices = 0;
   for (address = 1; address < 127; address++) {
      Wire.beginTransmission(address);
      error = Wire.endTransmission();

      if (error == 0) {
         Serial.print("I2C device found at address 0x");
         if (address < 16)
            Serial.print("0");

         Serial.print(address, HEX);
         Serial.println("  !");

         nDevices++;
      }
      else if (error == 4) {
         Serial.print("Unknown error at address 0x");
         if (address < 16)
            Serial.print("0");

         Serial.println(address, HEX);
      }
   }

   if (nDevices == 0)
      Serial.println("No I2C devices found");
   else
      Serial.println("done");

   delay(5000); // wait 5 seconds for next scan
}
```

Serial monitor Output:

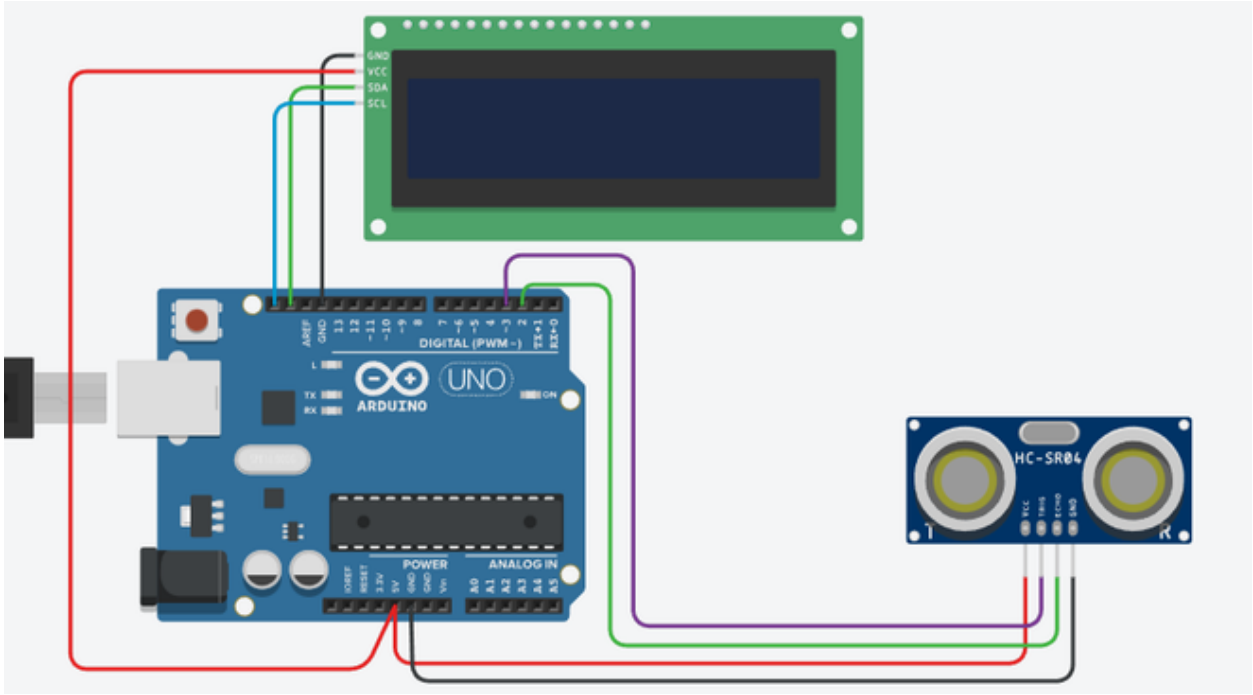


LCD Address Finding

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 6

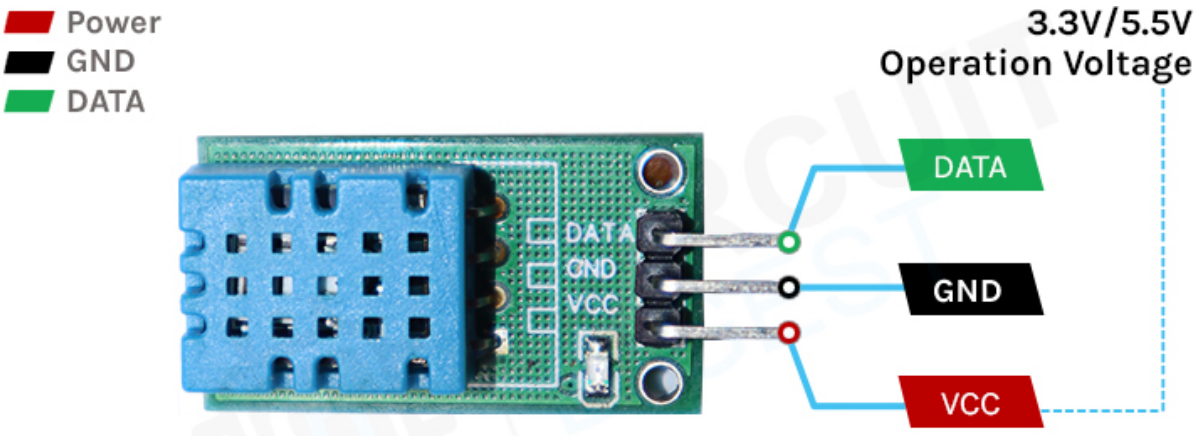
Aim: Interfacing of Arduino/Raspberry Pi with temperature and humidity sensor with real time application.

Components Required:

- Raspberry Pi3 Model B`s with installed Raspbian or Arduino Board
- 8 GB microSD cards
- Internet connection (Wired or Wireless) to access Pi Desktop
- VNC client on a wired or wireless device
- Breadboard
- Jumper Wires
- DH11 Temperature and Humidity Sensor
- USB or 5V Power Supply

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:

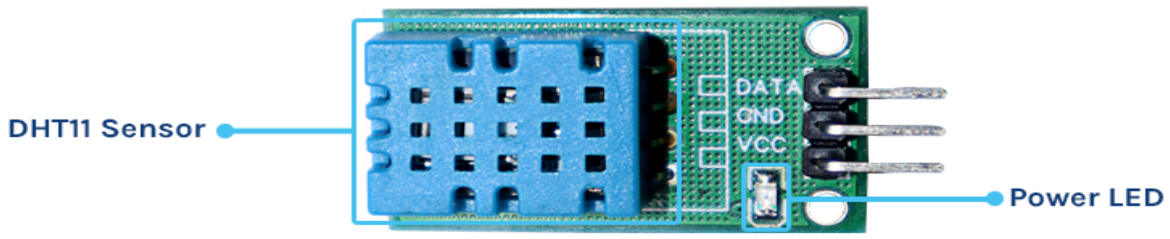


DATA Data pin for 1-wire communication.

GND Ground Connected to Ground pin of the Arduino.

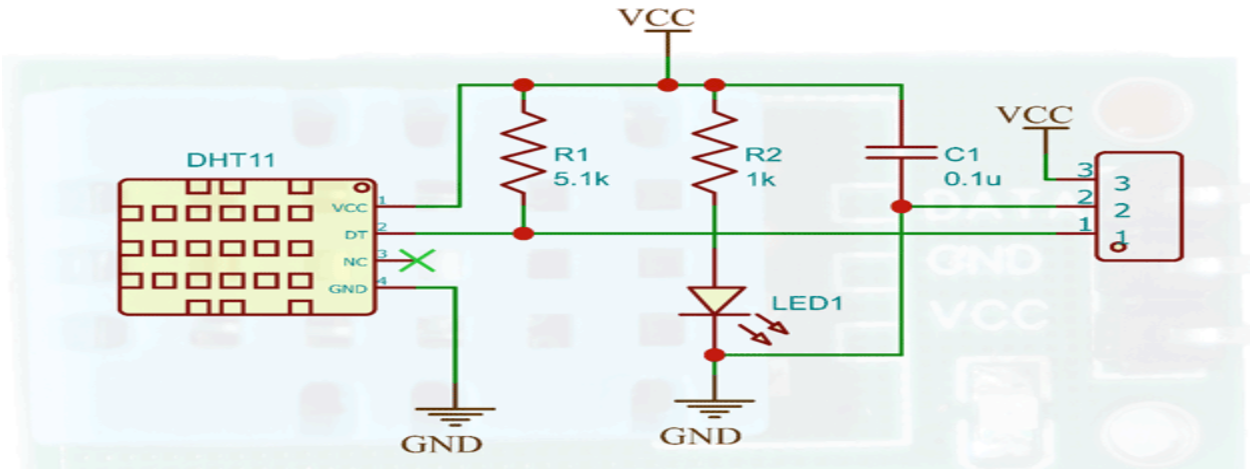
VCC Provides power for the module, Connect to the 5V pin of the Arduino.

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.



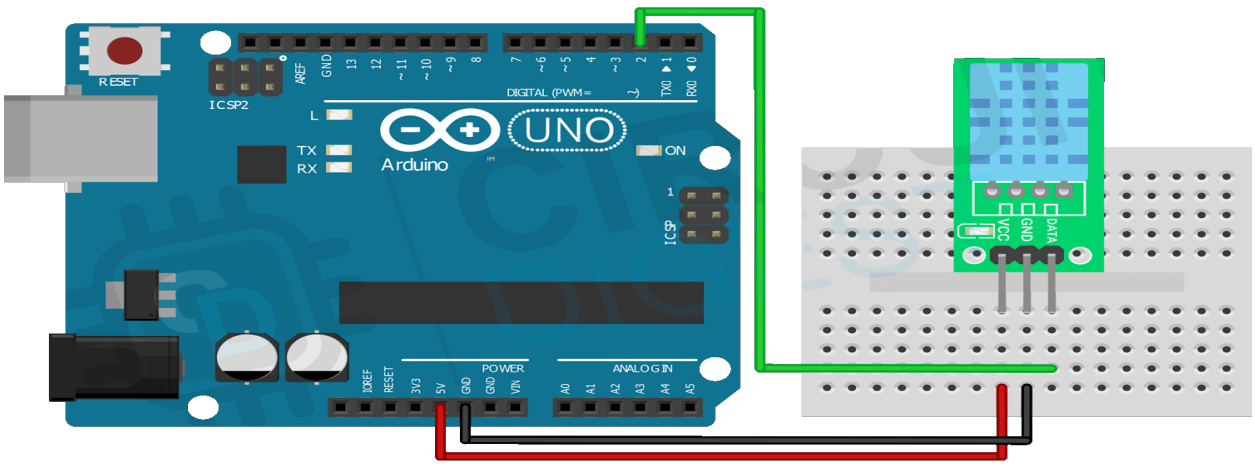
DHT11 Module Circuit Diagram

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.

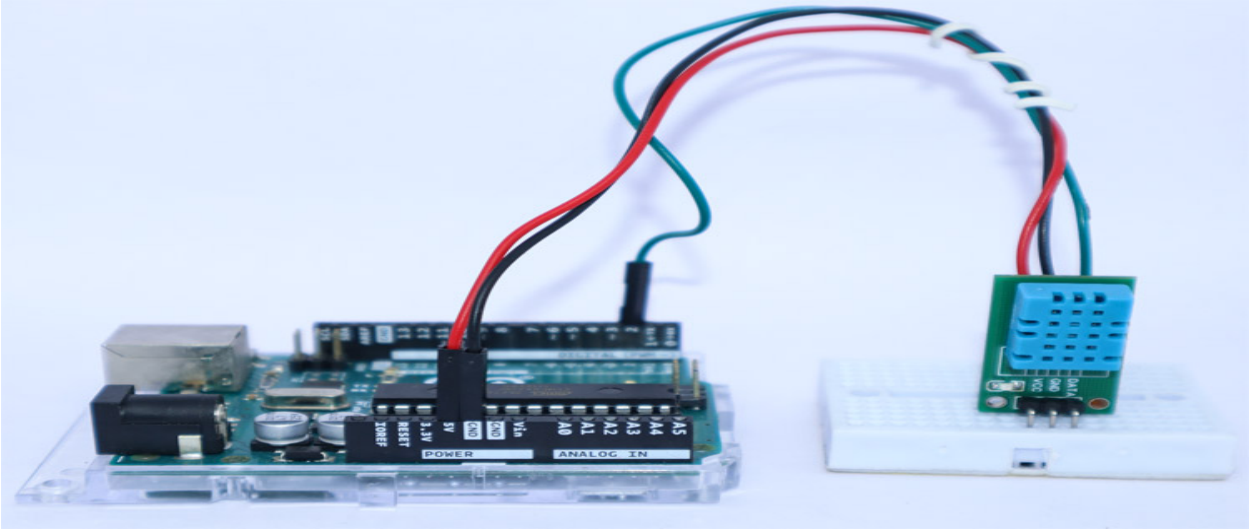


Circuit Diagram for Interfacing DHT11 Sensor with Arduino

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.

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 display data generated by sensor on LCD using Arduino/Raspberry Pi.

Components Required:

1. 1x Raspberry PI (Raspberry Pi B+ with rasbian wheezy)
2. 1x Arduino (I'm using the Arduino UNO)
3. 1x LCD 16x2

About LCD

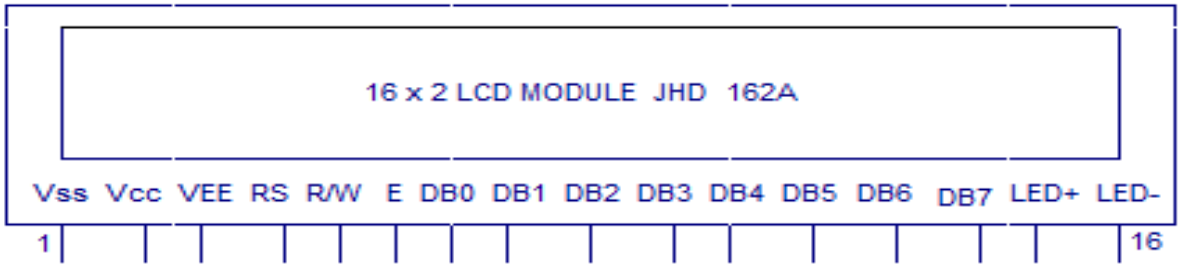
A Liquid Crystal Display commonly abbreviated, as LCD is basically a display unit built using Liquid Crystal technology. When we build real life/real world electronics based projects, we need a medium/device to display output values and messages. The most basic form of electronic display available is seven-segment display, which has its own limitations. The next best available option is Liquid Crystal Displays, which comes in different size specifications. Out of all available LCD modules in market, the most commonly used one is 16×2 LCD Module, which can display 32 ASCII, characters in 2 lines (16 characters in 1 line). Other commonly used LCD displays are 20×4 Character LCD, Nokia 5110 LCD module, 128×64 Graphical LCD Display and 2.4 inch TFT Touch screen LCD display.

Interfacing 16×2 LCD to Arduino uno

LCD modules form a very important part in many arduino based embedded system designs. So the knowledge on interfacing LCD module to arduino is very essential in designing embedded systems. Interfacing an Arduino to 16×2 LCD. JHD162A is the LCD module used here. JHD162A is a 16×2 LCD module based on the HD44780 driver from Hitachi. The JHD162A has 16 pins and can be operated in 4-bit mode (using only 4 data lines) or 8-bit mode (using all 8 data lines). Here we are using the LCD module in 4-bit mode. First, I we will see how to display a plain text messages on the LCD module using arduino and then we design a useful application using LCD and arduino – a digital thermometer. Before going in to the details of the project, let's have a look at the JHD162A LCD module.

16×2 LCD Module Pin Out Diagram

The JHD162A LCD module has 16 pins and can be operated in 4-bit mode or 8-bit mode. Here we are using the LCD module in 4-bit mode. Before going in to the details of the project, let's have a look at the JHD162A LCD module. The schematic of a JHD162A LCD pin diagram is given below.



Pin1 (Vss): Ground pin of the LCD module.

Pin2 (Vcc): Power to LCD module (+5V supply is given to this pin)

Pin3 (VEE): Contrast adjustment pin. This is done by connecting the ends of a 10K potentiometer to +5V and ground and then connecting the slider pin to the VEE pin. The voltage at the VEE pin defines the contrast. The normal setting is between 0.4 and 0.9V.

Pin4 (RS): Register select pin. The JHD162A has two registers namely command register and data register. Logic HIGH at RS pin selects data register and logic LOW at RS pin selects command register. If we make the RS pin HIGH and feed an input to the data lines (DB0 to DB7), this input will be treated as data to display on LCD screen. If we make the RS pin LOW and feed an input to the data lines, then this will be treated as a command (a command to be written to LCD controller – like positioning cursor or clear screen or scroll).

Pin5 (R/W): Read/Write modes. This pin is used for selecting between read and write modes. Logic HIGH at this pin activates read mode and logic LOW at this pin activates write mode.

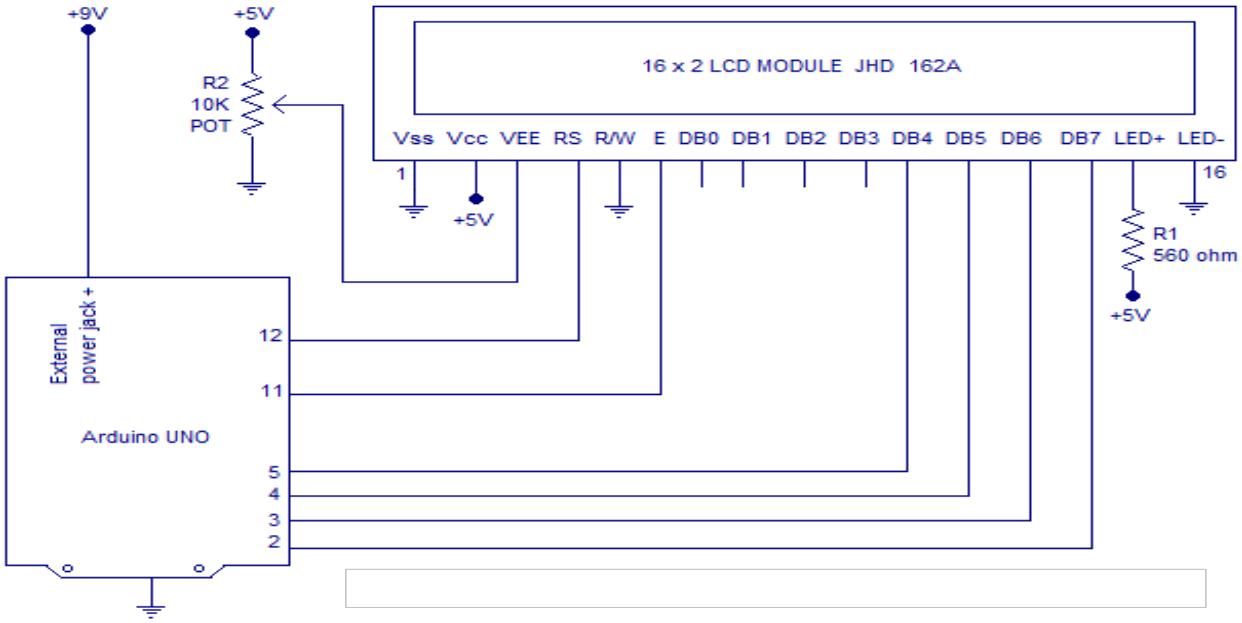
Pin6 (E): This pin is meant for enabling the LCD module. A HIGH to LOW signal at this pin will enable the module.

Pin 7(DB0) to Pin14 (DB7): These are data pins. The commands and data are fed to the LCD module though these pins.

Pin 15(LED+): Anode of the back light LED. When operated on 5V, a 560-ohm resistor should be connected in series to this pin. In arduino based projects the backlight LED can be powered from the 3.3V source on the arduino board.

Pin 16(LED-): Cathode of the back light LED.

Circuit diagram – Arduino to 16×2 LCD Module



RS pin of the LCD module is connected to digital pin 12 of the arduino. R/W pin of the LCD is grounded. Enable pin of the LCD module is connected to digital pin 11 of the arduino. In this project, the LCD module and arduino are interfaced in the 4-bit mode. This means only four of the digital input lines (DB4 to DB7) of the LCD are used. This method is very simple, requires less connections and you can almost utilize the full potential of the LCD module. Digital lines DB4, DB5, DB6 and DB7 are interfaced to digital pins 5, 4, 3 and 2 of the Arduino. The 10K potentiometer is used for adjusting the contrast of the display. 560-ohm resistor R1 limits the current through the back light LED. The arduino can be powered through the external power jack provided on the board. +5V required in some other parts of the circuit can be tapped from the 5V source on the arduino board. The arduino can be also powered from the PC through the USB port.

Program – Arduino to LCD

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  // sets the interfacing pins
void setup()
{
lcd.begin(16, 2);  // initializes the 16x2 LCD
}
void loop()
{
lcd.setCursor(0,0);          //sets the cursor at row 0 column 0
lcd.print("16x2 LCD MODULE"); // prints 16x2 LCD MODULE
lcd.setCursor(2,1);          //sets the cursor at row 1 column 2
lcd.print("HELLO WORLD");     // prints HELLO WORLD
}
```

About the program

To facilitate communication between Arduino and LCD module, we make use of a built in library in Arduino <LiquidCrystal.h> – which is written for LCD modules making use of the Hitachi HD44780 chipset (or a compatible chipset). This library can handle both 4 bit mode and 8 bit mode wiring of LCD.

Other Important aspects of Program

LiquidCrystal lcd() – is a constructor used to declare a variable of its type. Here 'lcd' is the variable declared using the constructor and is used to invoke methods defined inside the library LiquidCrystal.h (Example – lcd.print(); lcd.setCursor() and other methods)

lcd.begin() – is called to initialize the lcd screen and to pass the dimension of lcd screen (columns, rows) as parameters of the invoked method.

Program for scrolling the LCD screen using Arduino.

A simple program for scrolling a text message on the LCD screen using arduino is shown here. This is done using the "**scroll()**" method defined inside LiquidCrystal.h library. For example the method "lcd.scrollDisplayRight()" will scroll the display to right and the method"lcd.scrollDisplayLeft()" will scroll the display to left. A "for" loop is used for selecting the number of positions to scroll at a time. In the program shown below, it is chosen to be 2 because the text to be displayed is comparatively long. For shorter texts more number of positions must be scrolled at a time to get a smooth display.

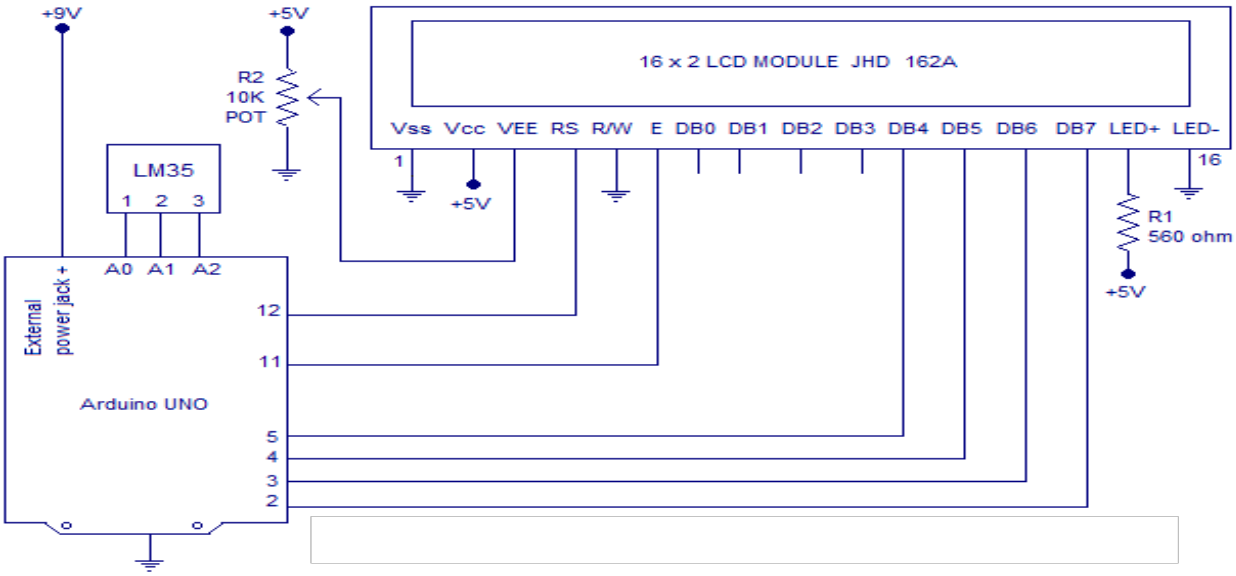
```
#include <LiquidCrystal.h>
int pos=0; // variable to hold cursor position
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup()
{
lcd.begin(16, 2);                    //initializes 16x2 LCD
lcd.print("16x2 LCD MODULE & ARDUINO-UNO");   //text to display
}
void loop()
{
for(pos=0; pos<2; pos++)
{
lcd.scrollDisplayLeft();    //scrolls display left by two positions
}
delay(800);            //sets the speed at which display moves
}
```

Note: This interfacing tutorial is good enough to learn interfacing of other dimensions of line LCD screens (say 8×1, 8×2, 16×1, 16×3, 16×4, 20×4, 20×2, 40×2, 40×4 etc) which are manufactured using the Hitachi HD44780 chipset. The pin configuration of all the line LCD screens (based on HD44780 chipset) is very same. This means the same circuit diagram is enough to interface other size lcd screens to arduino. We have explained more about this in the section given below – on 20×4 LCD to Arduino Interfacing

Digital thermometer with LCD display using Arduino

This is just a practical implementation of the interfacing of LCD and Arduino. A simple digital thermometer using arduino and 3-digit seven-segment display had been already published here. You can find that article here: Digital thermometer using arduino. Read this article before attempting the LCD version. LM35 is the temperature sensor used in this project. It is a three terminal linear analog temperature sensor. The output voltage of the LM35 increases 10mV per °C rise in temperature and the range is from -55°C to +155°C. The circuit diagram of the LCD thermometer using arduino is shown in the figure below.

Circuit diagram: LCD thermometer

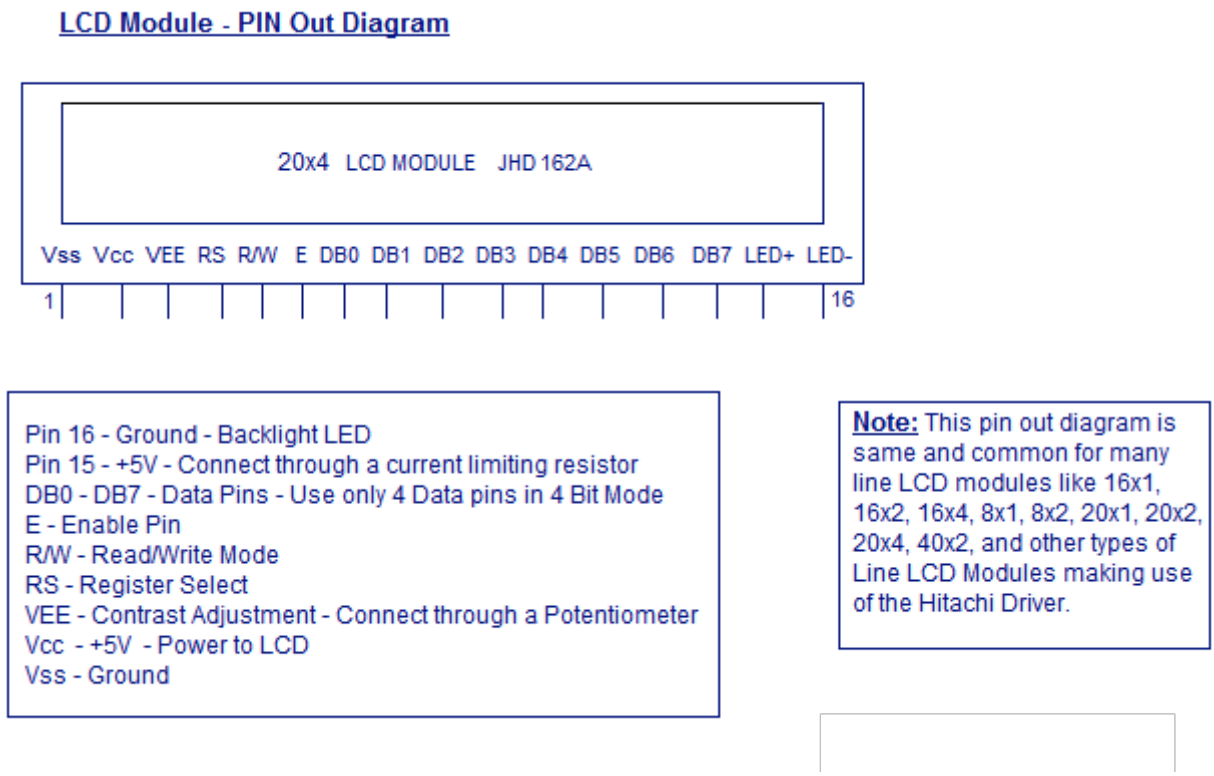


The LM35 temperature sensor is interfaced to the analog input pins of the arduino. Vcc pin (pin 1) of the LM35 is connected to A0 pin of the arduino. Output pin (pin 2) of the LM35 is connected to A1 pin of the arduino. GND pin (pin 3) of the LM35 is connected to A2 pin of the arduino.

Program: LCD thermometer

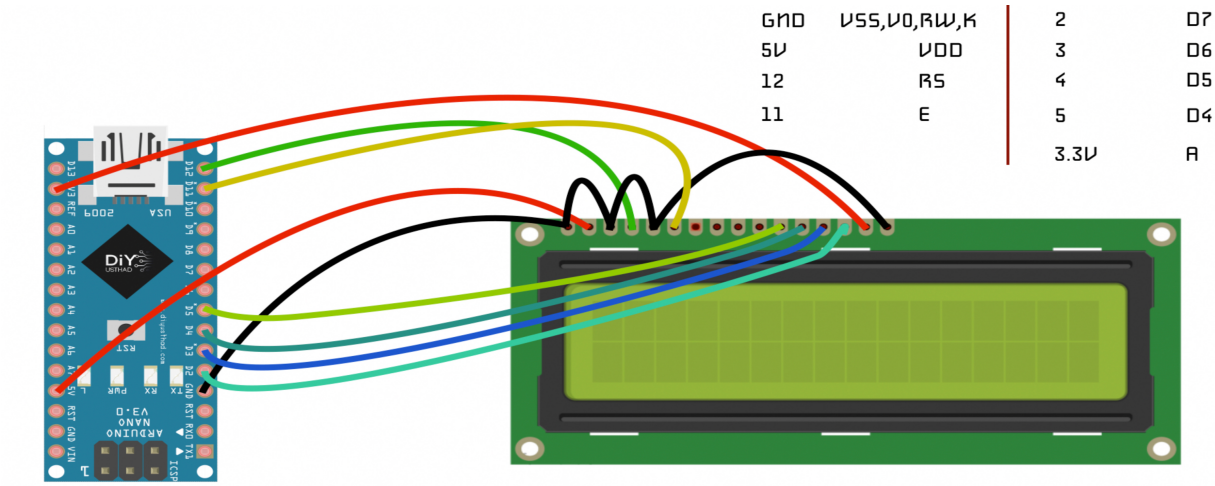
```
#include <LiquidCrystal.h>
int vcc=A0;
int sensor=A1;
int gnd=A2;
float temp;
float tempf;

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup()
{
pinMode(vcc,OUTPUT);
pinMode(sensor,INPUT);
pinMode(gnd,OUTPUT);
digitalWrite(vcc,HIGH);   // Vcc for LM35
digitalWrite(gnd,LOW);    // Ground for LM35
lcd.begin(16, 2);         // initializes the 16x2 LCD
lcd.setCursor(2,0);       // sets the cursor at column 2 row 0
lcd.print("TEMPERATURE"); // prints temperature
}
void loop()
{
 temp=analogRead(sensor); // reads the sensor output
 temp=temp*5;             // converts the sensor reading to temperature
 temp=temp/10;            // adds the decimal point
 tempf=(temp*1.8)+32;     // converts to Fahrenheit

 lcd.setCursor(0,1);   // sets cursor at column 0 row 1
 lcd.print(temp);      // prints temperature in degree Celsius
 lcd.print((char)223); // prints degree sign
 lcd.print("C");       // prints letter c
 lcd.setCursor(8,1);   // sets cursor at column 8 row 1
 lcd.print(tempf);     // prints temperature in degree Fahrenheit
 lcd.print((char)223); // prints degree sign
 lcd.print("F");       // prints letter F
 delay(1000);          // 1 second delay
}
```

Interfacing Arduino and 20×4 LCD Module



The 20×4 LCD module pin out diagram is very much same as the 16×2 LCD module pin out diagram. It is same with the number of pins, order of pins and the purpose of pins. So the interfacing circuit diagram is also very same as the 16×2 LCD module with Arduino.

Note: - The only changes you might need to make in the circuit diagram is with the current limiting resistor connected to backlight LED at pin 15 and with the potentiometer setting connected to VEE (the contrast levels of 16×2 and 20×4 modules might vary with a select potentiometer value). Rest all is very similar to interfacing a 16×2 LCD to Arduino.



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: Interfacing Air Quality Sensor (MQ135), displays data on LCD

Component Required:

1. Arduino Uno R3

2. MQ 135 AirQuality 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.

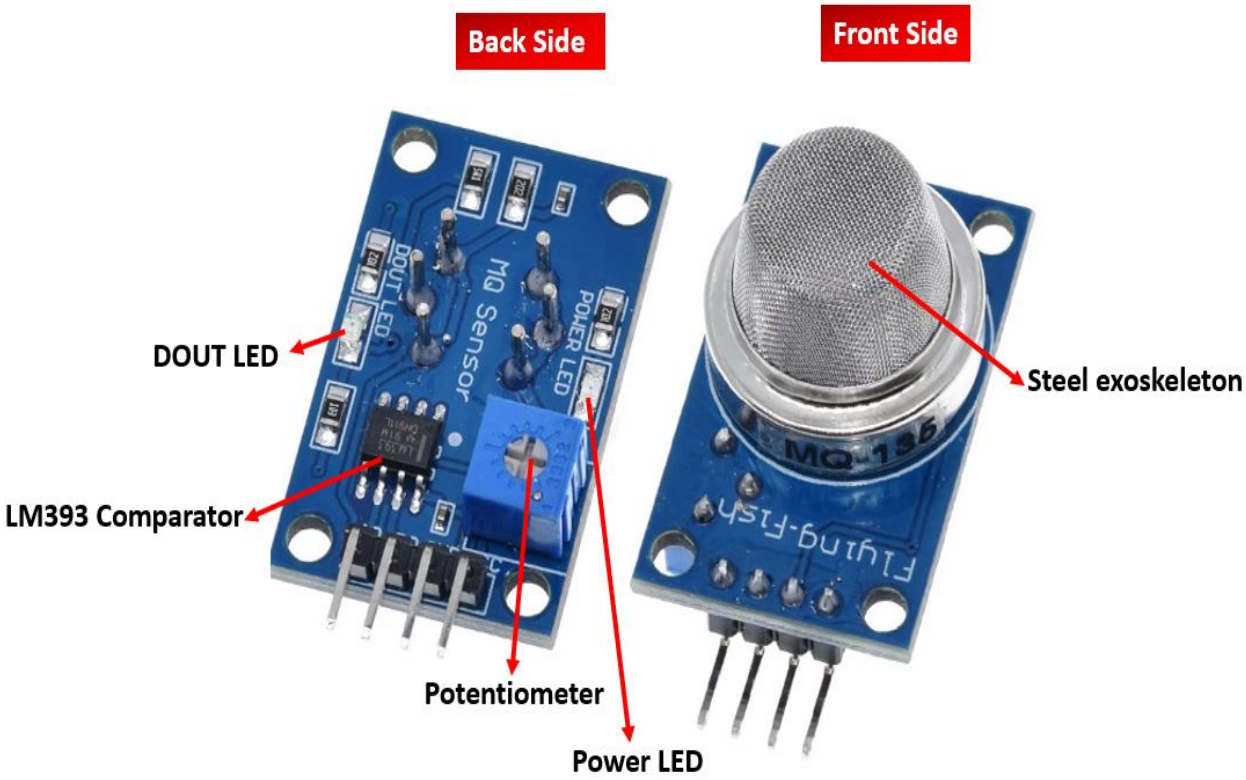


Figure: MQ 135 Sensor

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

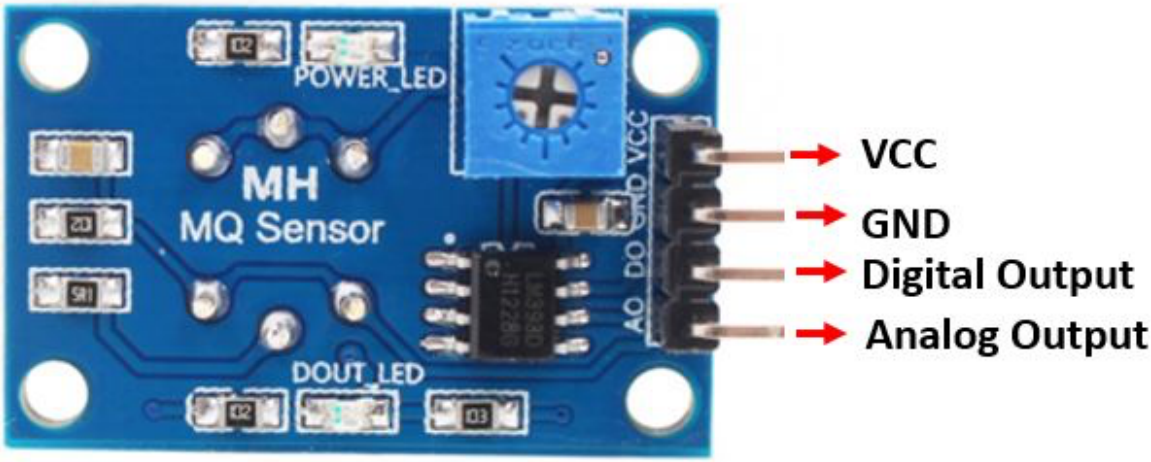


Figure: MQ-135 Sensor Pin out

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.

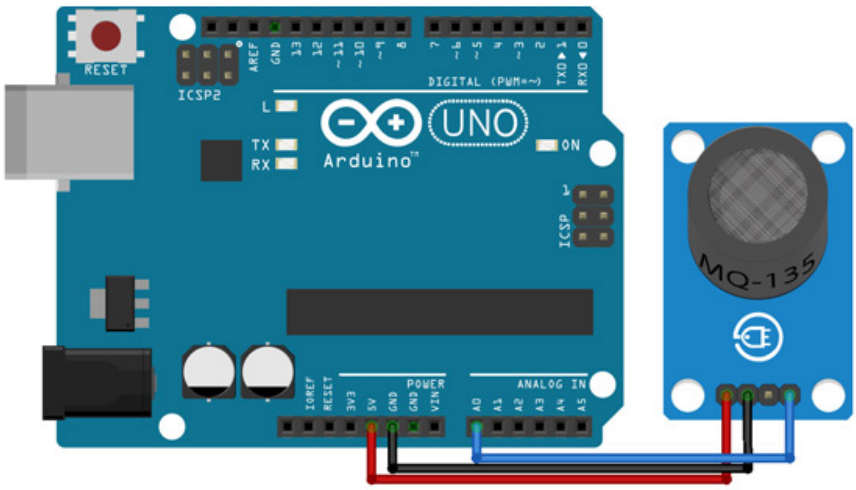


Figure: Connections of MQ135 with Arduino

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

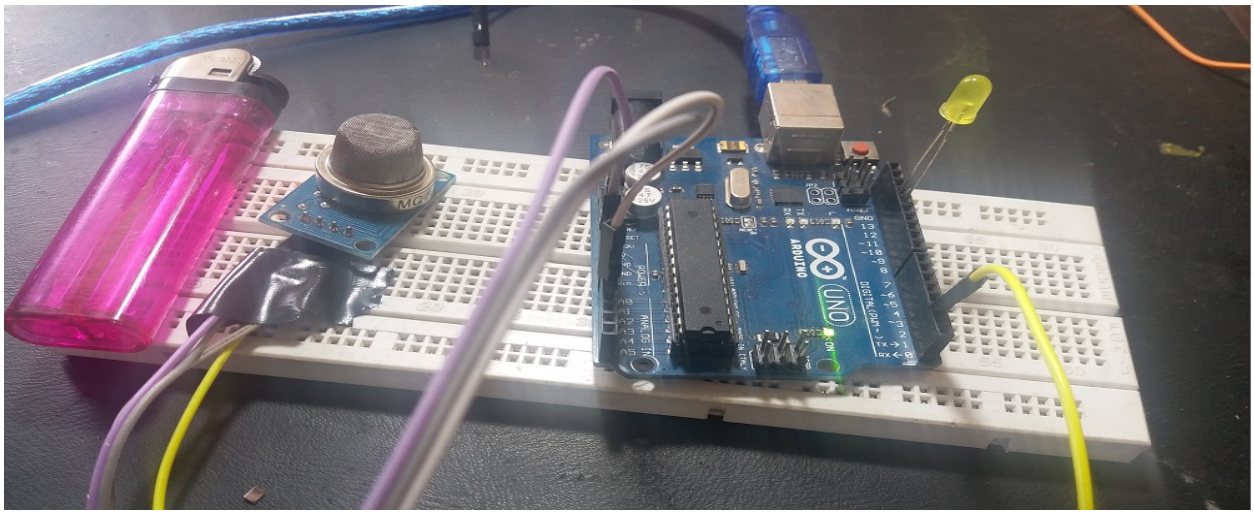


Figure: Interfacing of MQ135 with Arduino

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.

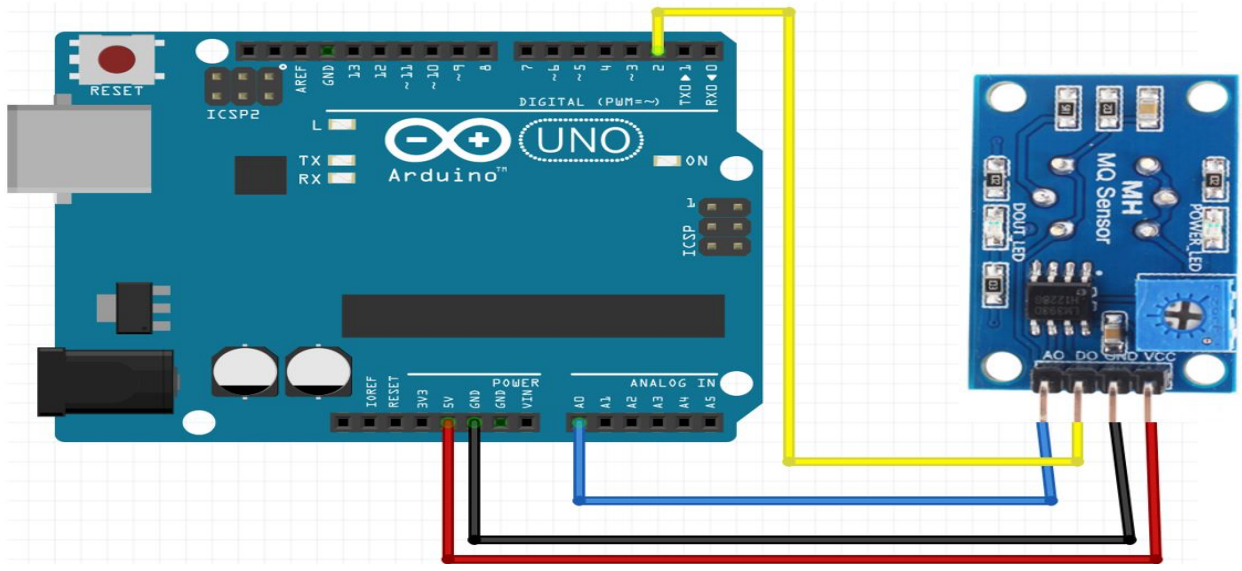


Figure: Arduino UNO with MQ-135 Module using both digital and analog outputs

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.

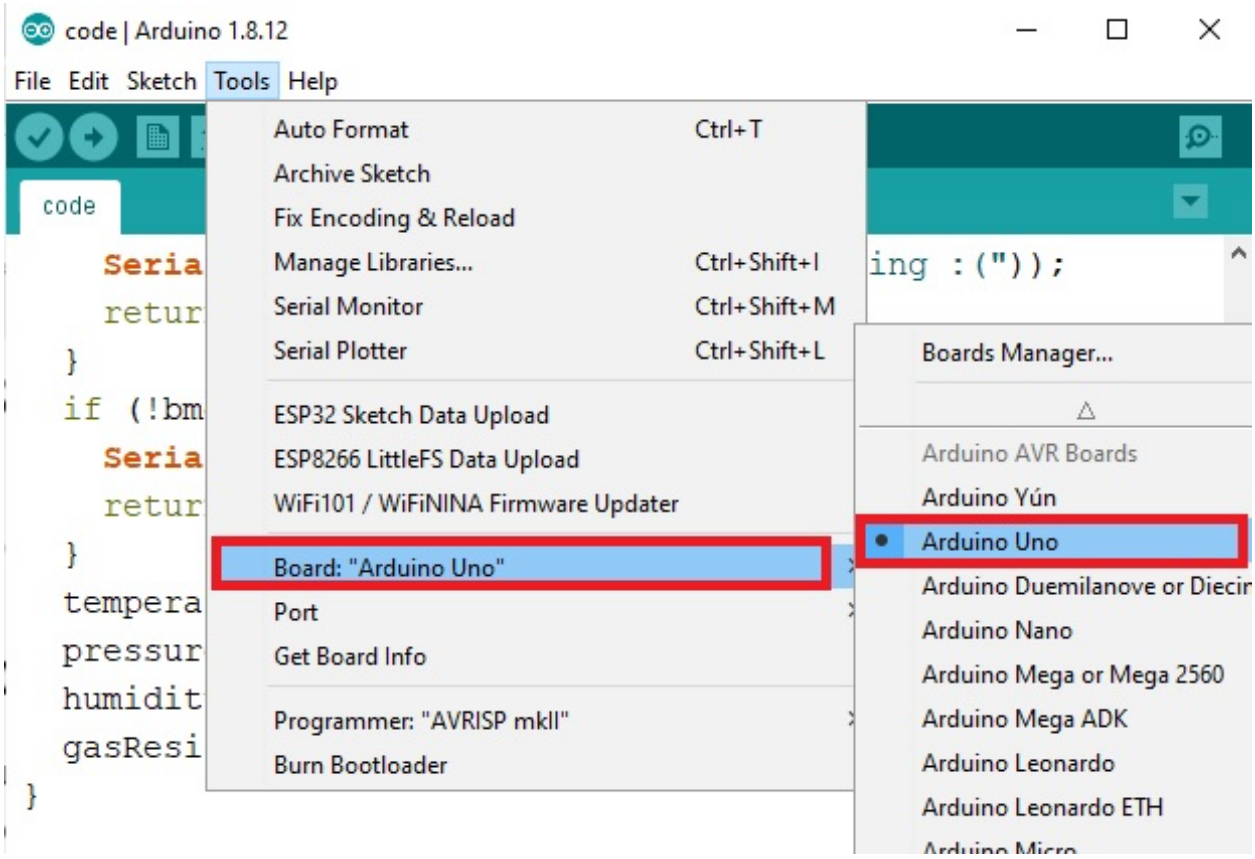
```
void setup()
{
 Serial.begin(9600); // sets the serial port to 9600
 pinMode(13, OUTPUT);
 pinMode(2, INPUT);
}
```

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 9

Aim: Real Time application of controlling actuators through Bluetooth application using Arduino.

Components Required:

8 Male/Male Jumper Wires
1 HC-05 Bluetooth Module
1 (5 mm) LED: Red
1 Arduino UNO
1 Resistor 1k ohm

Apps and platforms:
1 Arduino IDE
1 MIT App Inventor

Step 1 Here is what you need to control Led's with Bluetooth:

- Arduino
- HC-05 Bluetooth module
- Solder less breadboard
- 3 Led's
- 3 220Ω resistors
- Wires
- Most importantly your phone and a downloaded Bluetooth app (Arduino Bluetooth Controller, which offers many different features)

Step 2: Circuit

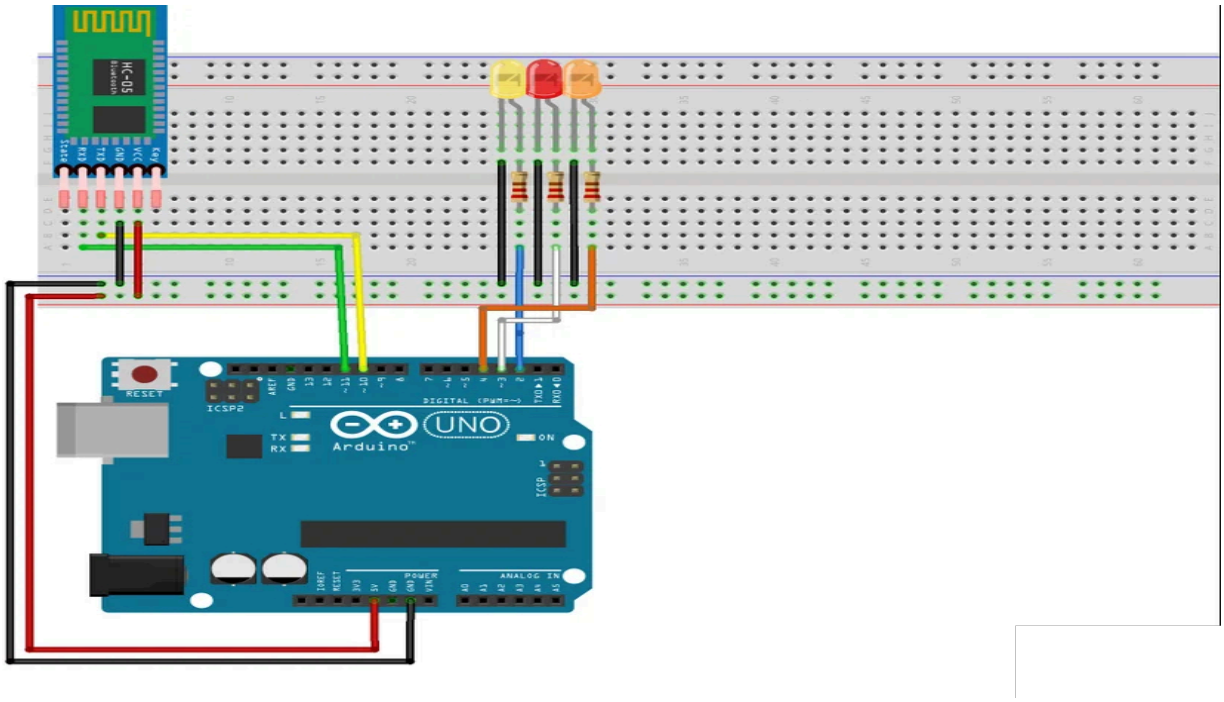


Figure: Circuit Diagram of HC05 interfacing with Ardunio

Bluetooth module connection:

- Connect the BT module's Rx pin to pin 11 on the Arduino
- Connect the BT module's Tx pin to pin 10 on the Arduino
- Connect up the Gnd and Vcc (5v) to the Arduino

Led's connection

- Connect all the cathodes (short pin) of the led to Gnd
- Connect each anode to a 220Ω resistor
- Connect a resistor to Arduino pin 2,3 and 4

If the led on the Bluetooth Module is blinking quickly then it is ready to pair to your phone, if not then check your connections

Code:

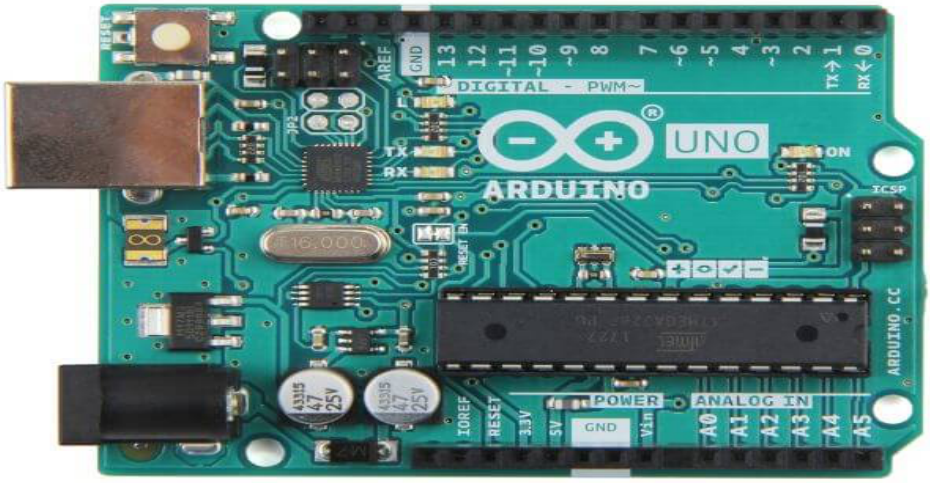
```
/*Code to control an LED via bluetooth through a phone app.
Project by MukeshArvindh. Code by MukeshArvindh.*/

/*If you are going to copy and paste the code, then do not forget
to delete the void setup() and void loop() function from the
sketch you are using before doing so,as the functions already
exist in this sketch. Copy-pasting the comments will not
cause any changes in the outcome or code.*/

/*Note:This code has been compiled and checked multiple times, and has proven to be accurate. The product also works as intended.*/

//Bluetooth uses serial communication. So, we use many serial functions //in this sketch.
const int LED = 5;
/*Declaring that there is an LED on pin 5 of the arduino board. We use const as we will not change this. You don't have to name it LED. You
can even put your name instead.*/
char switchstate;
/*declaring that there is a variable called switchstate, which will
hold a character value. This is due to programming of the app, which
will send a text value to arduino. If we use 'int' instead of
'char' the code will not work properly.*/
void setup() {//Here the code only runs once.
Serial.begin(9600);
/*To start serial communication at a rate of 9600 bits per second. This is the default rate anyways.*/
pinMode(LED, OUTPUT);
//Declaring that the LED is an output.
}
void loop() {//This code repeats. This is our main code.
while(Serial.available()>0){
//code to be executed only when Serial.available()>0
```

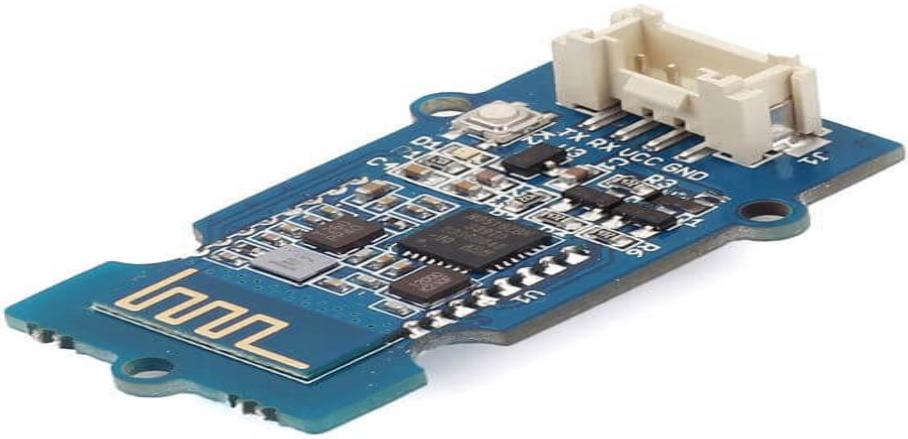
```
34 /*Serial.available>0 is to check if   there is any reading from the
35 HC-05 Bluetooth module.*/
36 switchstate = Serial.read();
37 /*The   character we had declared earlier is now being assigned a value- 38 the value of
whatever Serial.read() is.*/
39 //Serial.read() is to read the value coming from   app.
40 Serial.print(switchstate);
41 //This will print the value onto the Serial   monitor.
42 Serial.print("\
43 ");
44 //This moves to the next line after every   new line printed.
45 delay(15);
46 /*Gives a break of 15 milliseconds. Delay is   for human eye, and for
47 speed of some computers, as some will crash at high speeds.*/
48 if(switchstate   == '1'){//Checking if the value from app is '1'
49 digitalWrite(5, HIGH);
50 //If   it is, write the component on pin 5(LED) high.
51 }
52 else if(switchstate == '0'){//Else,   if the vaue from app is '0',
53 digitalWrite(5, LOW);//Write the component on pin   5(LED) low.
54 }
55 }
56 }
```

About the interfacing of Arduino & Bluetooth

Let's get started right away with the Arduino Bluetooth Tutorial, where I'll provide all you need to know about pairing Arduino with Bluetooth. Ranging from hardware configurations to programming and using it with your iPhone or Android devices!

1. **Arduino Bluetooth:**





Flexible and packed with high Bluetooth transmission speed, the Grove – Blueseeed LE – Dual Model (HM13) uses a CSR dual-mode Bluetooth chip, with the ARM architecture single chip that supports AT instructions.

This allows users like to have control over the serial baud rate, equipment name, and pairing password!

2. Hardware configurations:

Step 1: Connect the Grove – Blueseeed – Dual model (HM13) to a Grove port on the Grove – Base Shield via the Grove cable
Step 2: Plug Grove – Base Shield into your Arduino board
Step 3: Connect your Arduino to PC via USB cable

3. Understanding the software:
Conventions
In EDR mode, only the slave can be configured while either master or slave can be in BLE mode.

Factory default setting:EDR Name HMSoft, Slave role, PinCode 1234
BLE Name HMSoft, Slave role, PinCode 000000
Baud: 115200, N, 8, 1;

AT Command format: Uppercase AT command format. string format, without any other symbol. (e.g. \r or \n).

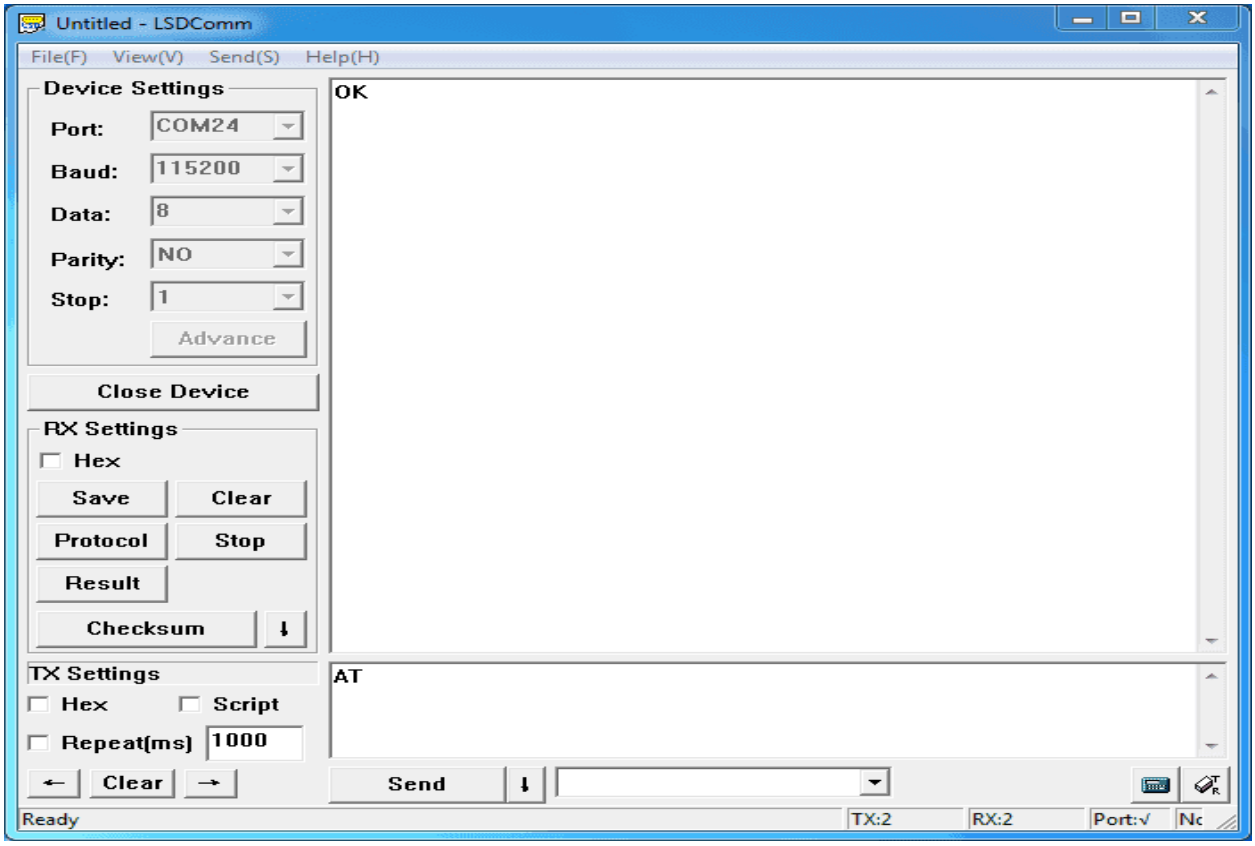
Any incorrect command would get no response.

4. How to program on the Arduino Bluetooth Module

After understanding the software configurations, here's how to configure Bluetooth with a PC. For hardware connection, do refer to the "Hardware configurations" section. You'll find that the flashing blue LED on the module illustrates no connection is set up

Step 1: Open a serial terminal and set Baud Rate: 115200, Databits: 8, Stopbits: 1, and no flow control like above

Step 2: Send "AT" to Bluetooth with the serial terminal to check if you receive an "OK" The Bluetooth only respond AT commands either when: No connection is set up All commands were seen as string and sent out
*You can distinguish the above status in step 2 through LED indications.



Here are some useful configurations that can be sent:

Test serial connection, send "AT", will return "OK".
Restore factory settings, send "AT+RENEW", return "OK+RENEW".
Reset baud rate of serial port, send "AT+BAUD2", return "OK+Set:2".
Enable authentication, send "AT+AUTH1", return "OK+Set:1".
Reset the Bluetooth, send "AT+RESET", return "OK+RESET".
Query firmware version, send "AT+VERS?", return "OK+Get:HMSoftV217".
Query MAC of EDR, send "AT+ADDE?", return "OK+Get:000E0E002074".
Query MAC of BLE, send "AT+ADDB?", return "OK+Get:000E0B002074".
Set the name of EDR, send "AT+NAMEHM-13-EDR", return "OK+Set:HM-13-EDR".
Set the name of BLE, send "AT+NAMEHM-13-BLE", return "OK+Set:HM-13-BLE".
Set the password of EDR, send "AT+PINE123451", return "OK+Set:123451".
Set the password of BLE, send "AT+PINB123451", return "OK+Set:123451".
Enable discovery and connectable, send "AT+SCAN0", return "OK+Set:0".
Enable notify information of connection, send "AT+NOTI1", return "OK+Set:1".
Notify information include address, send "AT+NOTP1", return "OK+Set:1".
Enable user key, send "AT+PIO01", return "OK+Set:1".
Set to Central mode, send "AT+ROLB1", return "AT+ROLB1".
Set to Peripheral mode, send "AT+ROLB0", return "AT+ROLB0".

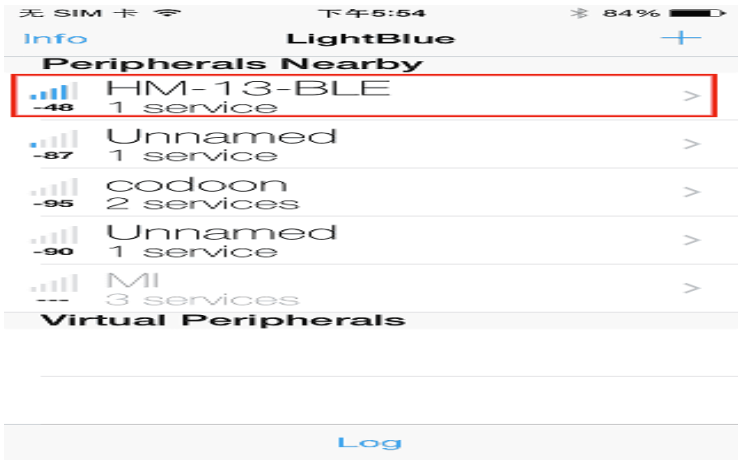
We used two Bluetooth that were connected with the PC, with one set as central while the other as Peripheral. Several seconds later, they find each other, and the LED stops flashing connected!

5. How to pair Arduino Bluetooth Module with iPhone and Andriod

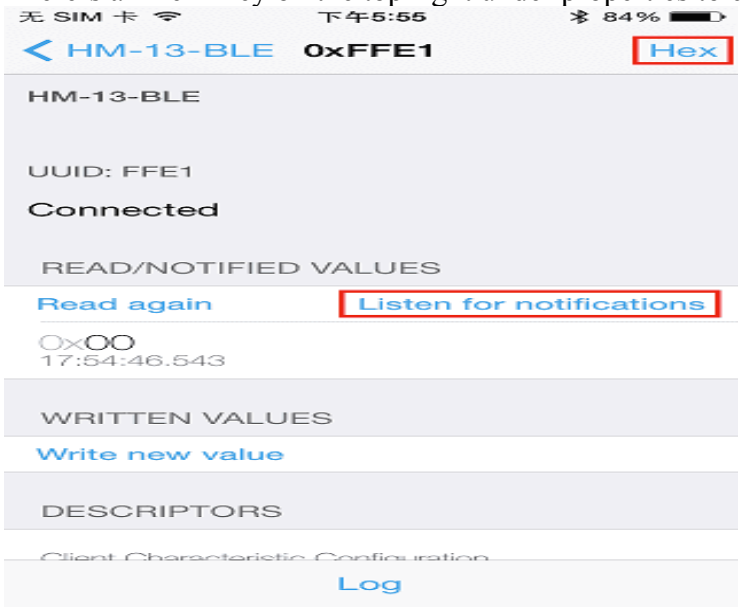
Since the Grove – Blueseeed – Dual model (HM13) have two protocol: Bluetooth EDR (Enhanced Data Rate) and Bluetooth Low Energy (BLE), it can communicate with either Andriod or iPhones!
For this part of the tutorial, we'll use an iPhone to demonstrate how you can interact with Bluetooth!

*Note: The tutorial below is run on an older version of ios but it should still work the same

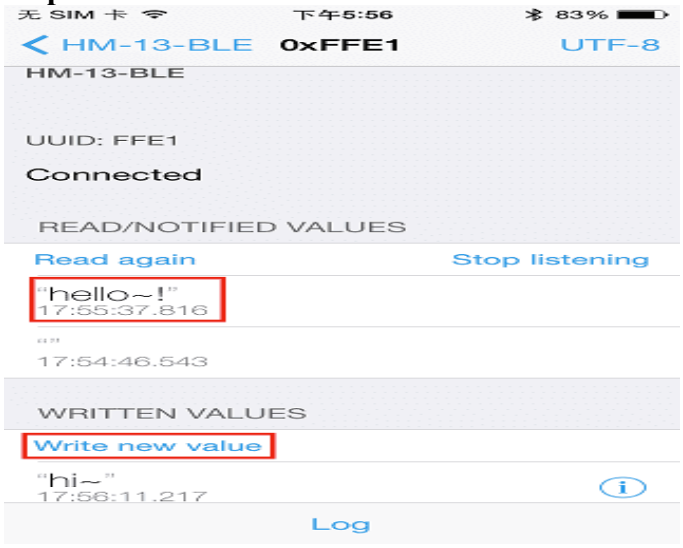
Step 1: Power the Bluetooth and configure it as a Peripheral role
Step 2: Search Light Blue in the App Store and install it
Step 3: Launch the app, and connect to "HM-13-BLE"



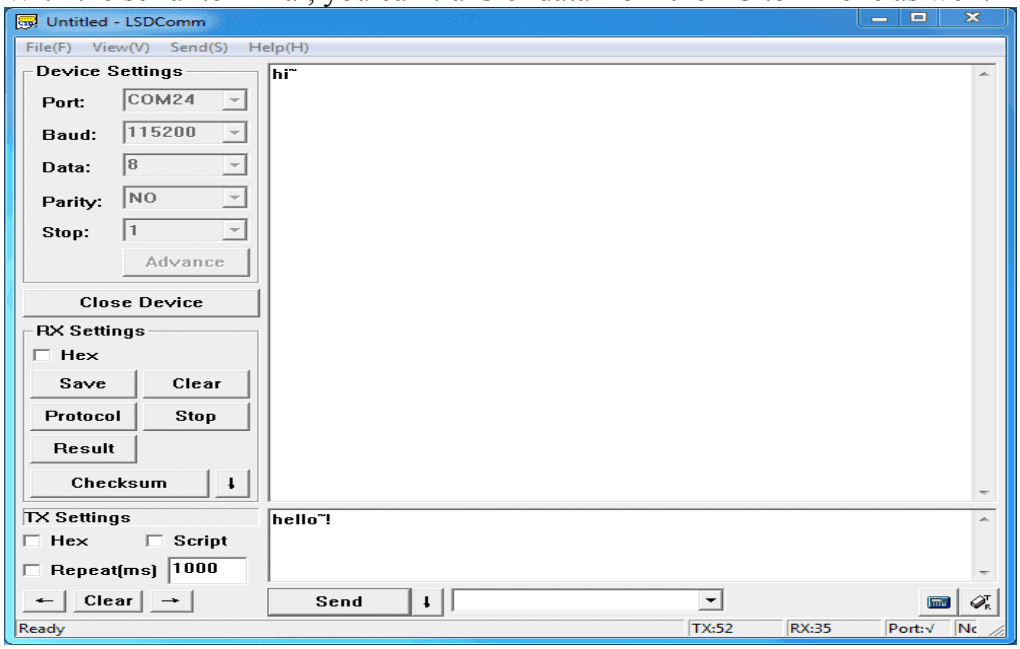
Step 4: Touch on properties and hit "listen for notifications" to enable data receiving There's a "Hex" key on the top right under properties to change data format as well



Step 5: Hit “Write new value” and write some words to start sending data to the PC



With the serial terminal, you can transfer data from the PC to iPhone as well:



6. Bluetooth Data transmission guide between two Arduino boards

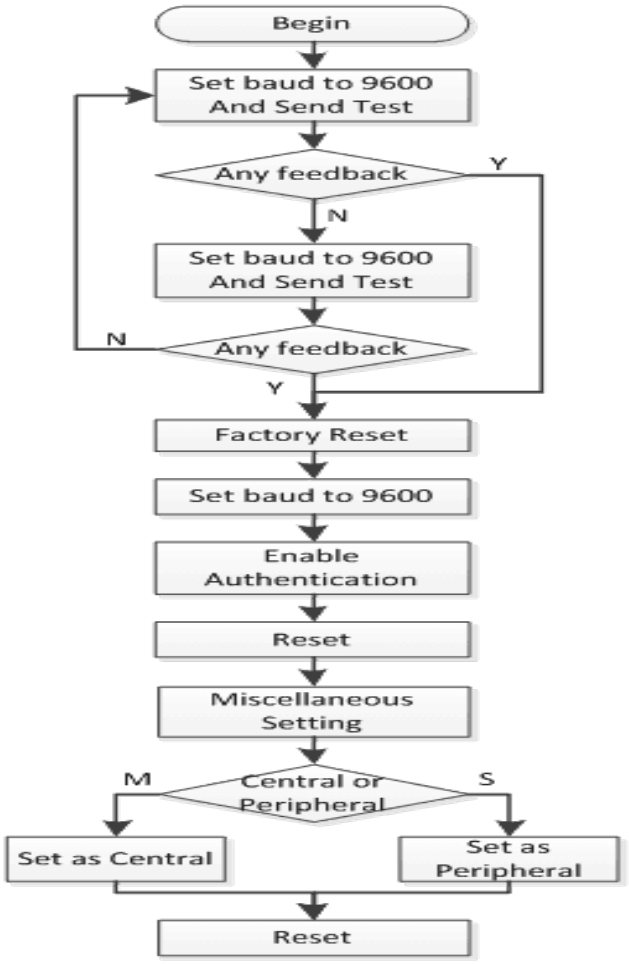
Now after all the above steps, are you ready to code? In this final section, we’ll use two Arduino Uno and a pair of Bluetooth modules to get started!

Step 1: Set up the connection mentioned in the hardware configurations section

Step 2: Assign the Bluetooth to the Central role by modifying the text to “#define MASTER 1”

The program of Central and Peripheral use the same code but there’s a difference in the micro define at the beginning of the program

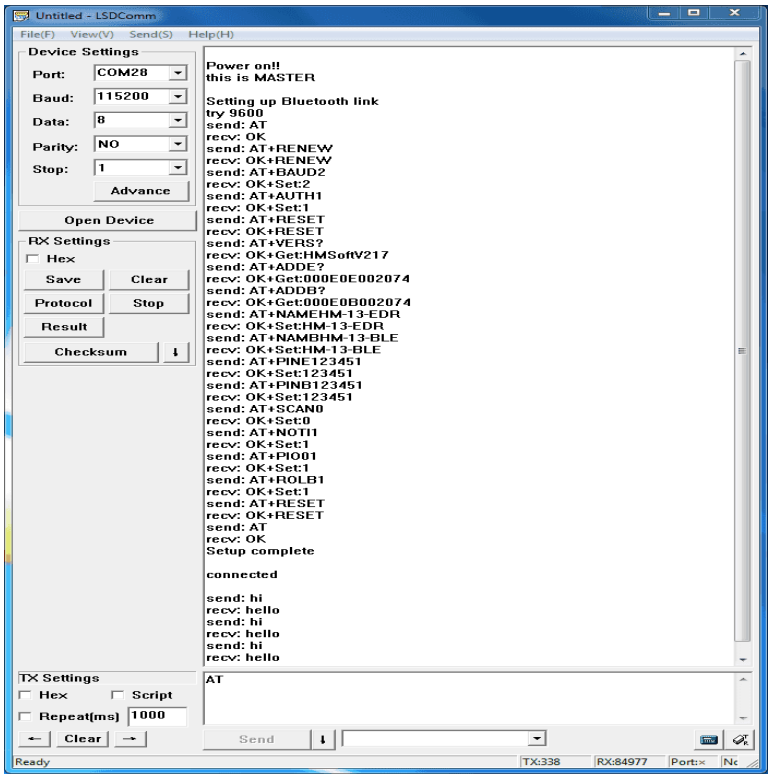
Step 3: Follow the flow chart below for initialization of the program

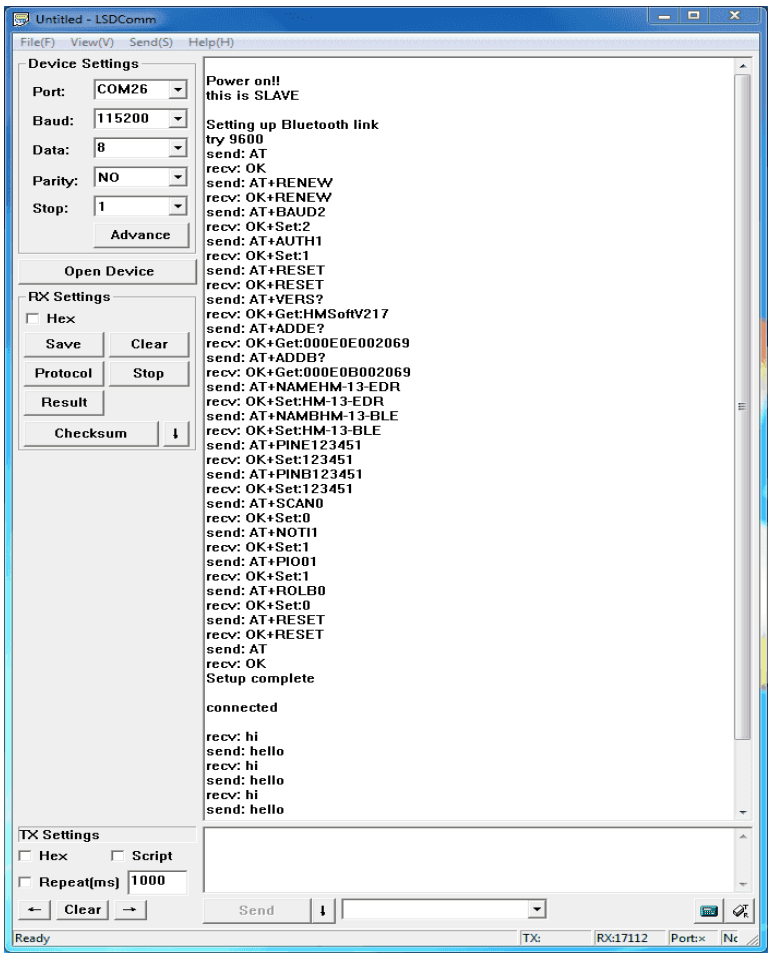


After the initialization, the Central and Peripheral will do different things; Central will send a message to the Peripheral interval and print what's received from the Peripheral while the Peripheral only responds to the central

Step 4: Download the test code and open HM-13_SW.ino with Arduino IDE, compile and download to Arduino Uno
Remember to configure the Bluetooth to the different role by modifying the macro at the beginning

Step 5: After the program is downloaded, open two serial terminal windows and wait for the Bluetooth connection
A connection is indicated by: LEDs on the Bluetooth modules will flash for a few seconds, stop flashing, and kept on
According to the program written, the Central will now send a message to the Peripheral continually and get feedback every time


Untitled - LSDComm
File(F) View(V) Send(S) Help(H)
Device Settings
Port: COM28
Baud: 115200
Data: 8
Parity: NO
Stop: 1
Advance
Open Device
RX Settings
Hex
Save
Clear
Protocol
Stop
Result
Checksum
Power on!!
this is MASTER
Setting up Bluetooth link
try 9600
send: AT
recv: OK
send: AT+RENEW
recv: OK+RENEW
send: AT+BAUD2
recv: OK+Set:2
send: AT+AUTH1
recv: OK+Set:1
send: AT+RESET
recv: OK+RESET
send: AT+VERS?
recv: OK+Get:HMSoftV217
send: AT+ADDE?
recv: OK+Get:000E0E002074
send: AT+ADDB?
recv: OK+Get:000E0B002074
send: AT+NAMEHM-13-EDR
recv: OK+Set:HM-13-EDR
send: AT+NAMBHM-13-BLE
recv: OK+Set:HM-13-BLE
send: AT+PINE123451
recv: OK+Set:123451
send: AT+PINB123451
recv: OK+Set:123451
send: AT+SCAN0
recv: OK+Set:0
send: AT+NOTI1
recv: OK+Set:1
send: AT+PIO01
recv: OK+Set:1
send: AT+ROLB1
recv: OK+Set:1
send: AT+RESET
recv: OK+RESET
send: AT
recv: OK
Setup complete
connected
send: hi
recv: hello
send: hi
recv: hello
send: hi
recv: hello
TX Settings
Hex
Script
Repeat(ms) 1000
Clear
Send
AT
Ready
TX:338
RX:84977
Port:×

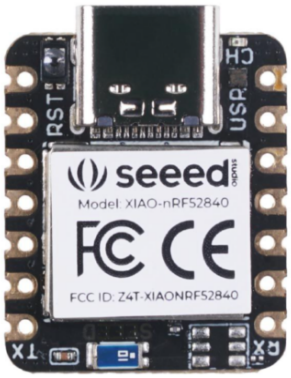

Untitled - LSDComm
File(F) View(V) Send(S) Help(H)
Device Settings
Port: COM26
Baud: 115200
Data: 8
Parity: NO
Stop: 1
Advance
Open Device
RX Settings
Hex
Save
Clear
Protocol
Stop
Result
Checksum
Power on!!
this is SLAVE
Setting up Bluetooth link
try 9600
send: AT
recv: OK
send: AT+RENEW
recv: OK+RENEW
send: AT+BAUD2
recv: OK+Set:2
send: AT+AUTH1
recv: OK+Set:1
send: AT+RESET
recv: OK+RESET
send: AT+VERS?
recv: OK+Get:HMSoftV217
send: AT+ADDE?
recv: OK+Get:000E0E002069
send: AT+ADDB?
recv: OK+Get:000E0B002069
send: AT+NAMEHM-13-EDR
recv: OK+Set:HM-13-EDR
send: AT+NAMBHM-13-BLE
recv: OK+Set:HM-13-BLE
send: AT+PINE123451
recv: OK+Set:123451
send: AT+PINB123451
recv: OK+Set:123451
send: AT+SCAN0
recv: OK+Set:0
send: AT+NOTI1
recv: OK+Set:1
send: AT+PIO01
recv: OK+Set:1
send: AT+ROLB0
recv: OK+Set:0
send: AT+RESET
recv: OK+RESET
send: AT
recv: OK
Setup complete
connected
recv: hi
send: hello
recv: hi
send: hello
recv: hi
send: hello
TX Settings
Hex
Script
Repeat(ms) 1000
Clear
Send
Ready
TX:
RX:17112
Port:×

Arduino-supported MCU with Bluetooth 5.0

As the first wireless product in the Seeed XIAO family, Seeed XIAO BLE & BLE Sense has equipped a powerful Nordic nRF52840 MCU which is designed in **a Bluetooth 5.0 module**, built around a 32-bit ARM® Cortex™-M4 CPU with Floating-Point Unit(FPU) operating at 64Mhz.

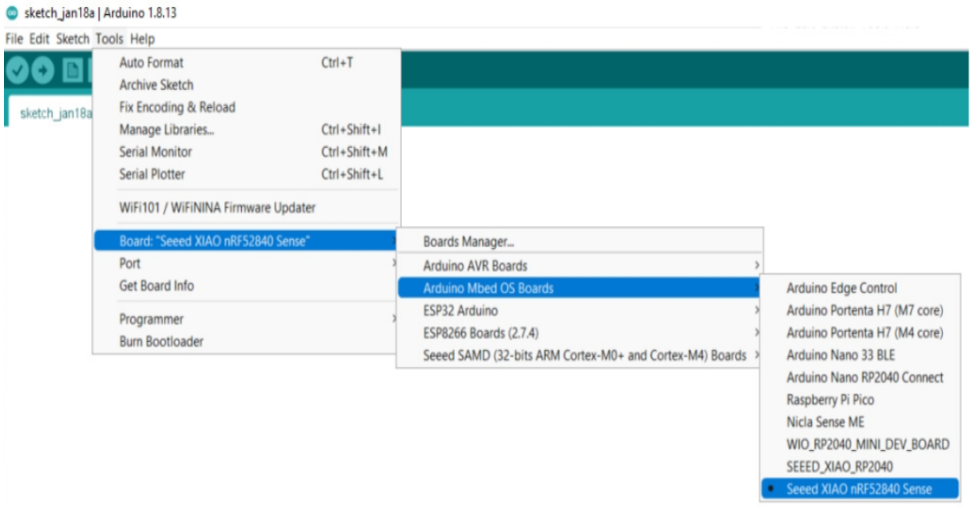
Seeed XIAO BLE nRF52840
-Supports Arduino / MicroPython -Bluetooth5.0 with Onboard Antenna

Seeed XIAO BLE nRF52840 Sense
– TinyML/TensorFlow Lite- IMU / Microphone – Bluetooth5.0



Here are features and specifications:

Powerful CPU: Nordic nRF52840, ARM® Cortex™-M4 32-bit processor with FPU operating at 64 MHz
Wireless capabilities: **Bluetooth 5.0**, NFC, and ZigBee module with onboard antenna
Ultra-small size: 21 x 17.5mm, Seeed Xiao series classic form-factor for wearable devices
Ultra-low sleep power: 5 μA, deep sleep model
Battery charging chip: BQ25101 chip supported lithium battery charge management
Rich interface: 1x Reset button, Ix UART, 1x IIC, 1x SPI, 1x NFC, 1x SWD, 11x GPIO, 6x ADC, 1x Three-in-one LED,1x User LED
Onboard 2 MB flash
Onboard PDM microphone and 6-axis IMU **(only for XIAO BLE nRF52840 Sense)**
Single-sided components, surface mounting design
Support Arduino/ MicroPython/ CircuitPython
Bluetooth usage tutorial on XIAO BLE (Sense)
If you “originate” from the Arduino ecosystem, you can still use your Arduino IDE and many libraries to program the XIAO BLE (Sense).
XIAO BLE and XIAO BLE Sense both support Bluetooth connectivity.



Viva Voce:

1. The frequency band of Bluetooth radio is around _____
2. Define baseband
3. Explain LMP.
4. List benefits of Bluetooth technology.
5. Define network topology.
6. The single piconet formed by _____
7. The multiple piconets formed by _____
8. Explain Zigbee.
9. Define Bluetooth.
10. How many channels do Bluetooth consists?
11. The architecture of Bluetooth is called _____
12. A _____ node is a node from which data is being sent.
13. In which node the data is being received?
14. How many nodes do piconet consists of?
15. What is the advantage of using Bluetooth technology?
16. How many layers are present in the Bluetooth technology?
17. _____ is a combination of baseband layer and link manager
18. What was the range of Bluetooth?
19. The power consumption in Bluetooth is _____
20. What is the standard form of L2CAP?
21. _____ are the functions of logical link control adaptation layer.
22. What is the standard form of ISM?
23. How many firms does the Bluetooth frame format have?
24. How many bits does the access code have in Bluetooth frame format?
25. How many bits does the header have in Bluetooth frame format?
26. The access code in Bluetooth frame format divided into how many sections?
27. How many bits do the preamble in an access code consists of?
28. The IEEE 802.11 b and 802.11 g wireless standard frequency range is around _____
29. When was Bluetooth version 1.0 is released?
30. The Bluetooth version 5 supports IOT devices True or False.
31. What are the disadvantages of using Bluetooth?

Experiment 10

Aim: Study the Implementation of Zigbee Protocol using Raspberry Pi/Arduino.

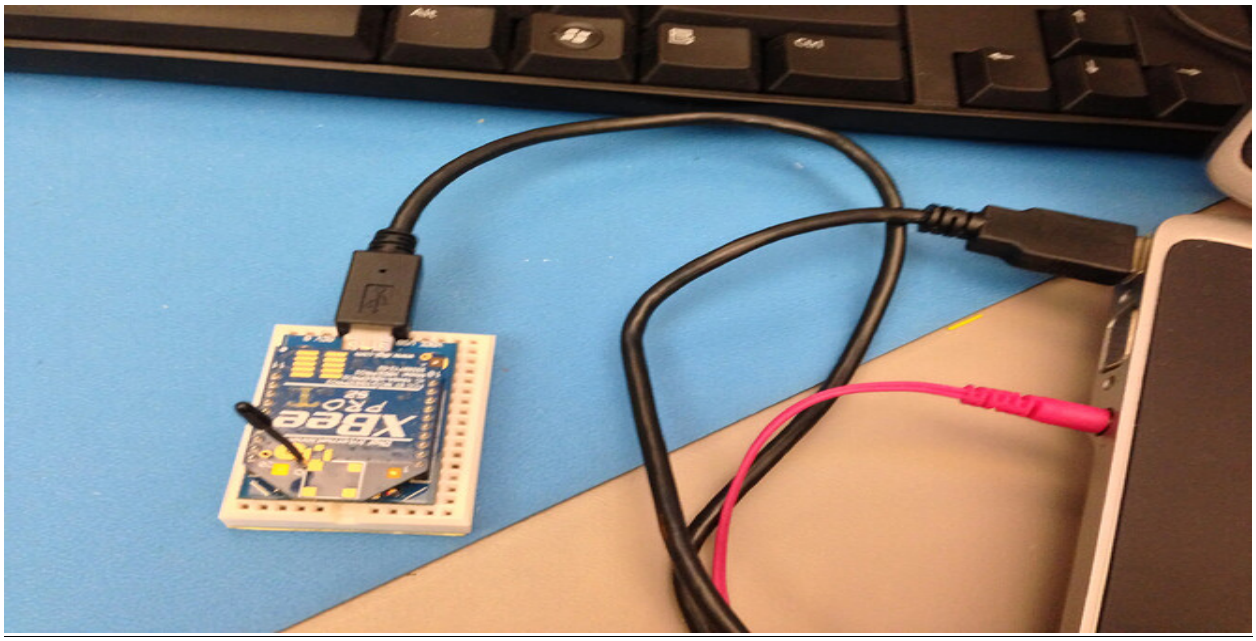
Introduction
XBee wireless transceivers provide a quick, easy way to add wireless communication to any system. This page will outline how to set up two XBee Pro Series 2 transceivers for communication with each other.

Hardware
2 XBee Pro S2 Transceiver
2 UART to USB adapter board
1 USB Cord

XBee Wireless Communication Setup

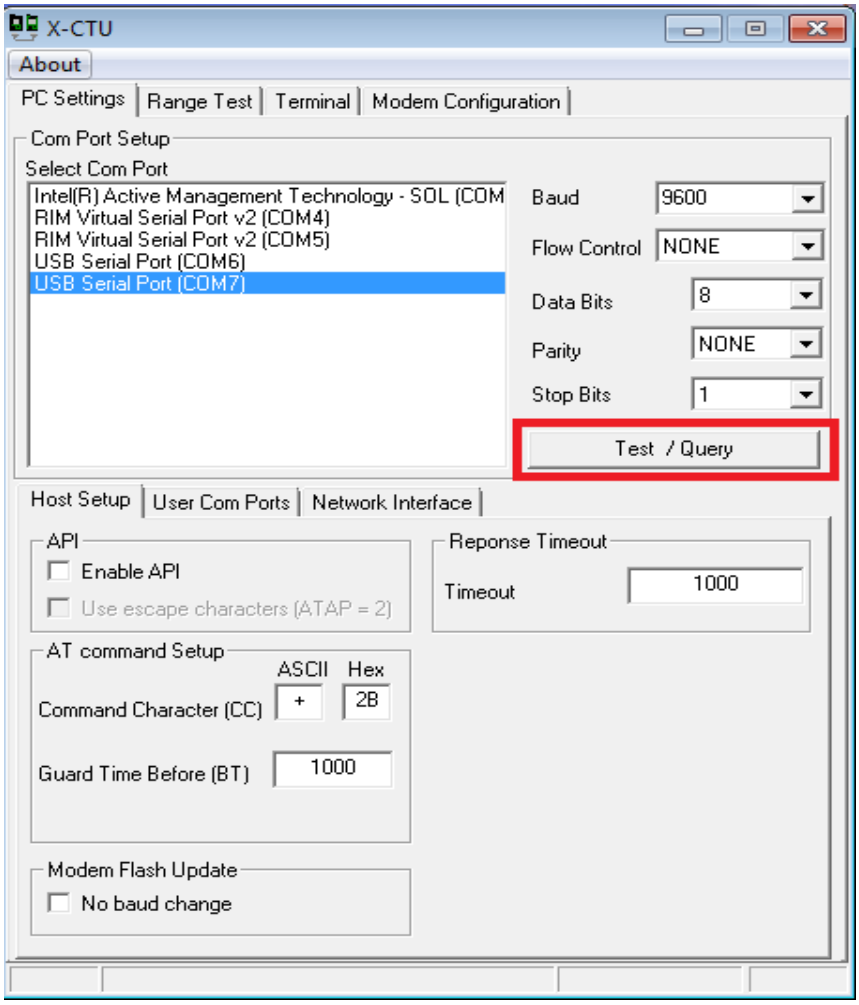
Step 1: Download X-CTU Software
The X-CTU software is free to download and provides a simple interface to configure and update your XBee transceivers. With this software firmware updates are a breeze and configuration is simple. The software can be downloaded from Digi's website.

Step 2: Put together your XBee breakout board
The XBee transceivers have a 2mm pin spacing, which does not allow them to be plugged into a standard 0.1-inch breadboard. There are, however, several different breakout boards available that allow the transceiver to be inserted into a breadboard. The various adapter boards also allow for connection through USB or serial to your computer. The breakout board used here has a UART to USB conversion circuit and allows the XBee to be connected to the computer and X-CTU software easily. By following the link above in the Hardware list, the data sheet with step-by-step instructions for putting together the adapter board can be found. After your adapter board is assembled you can plug your XBee into it and then connect it to your computer.

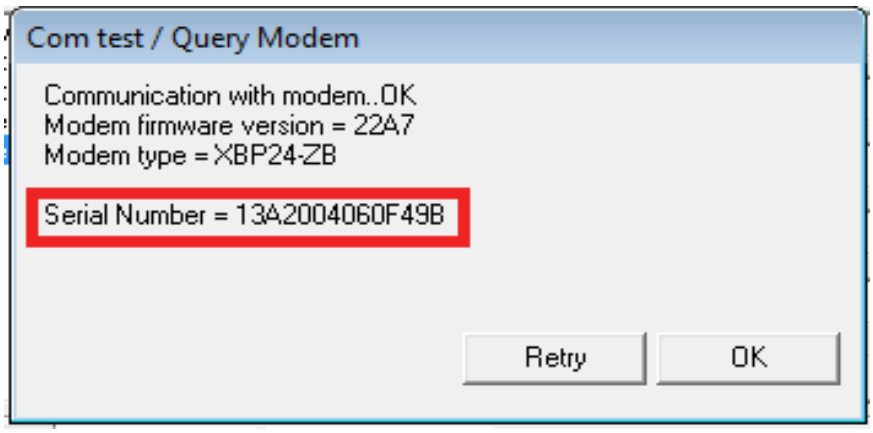


Step 3: Configure 1st XBee as a coordinator

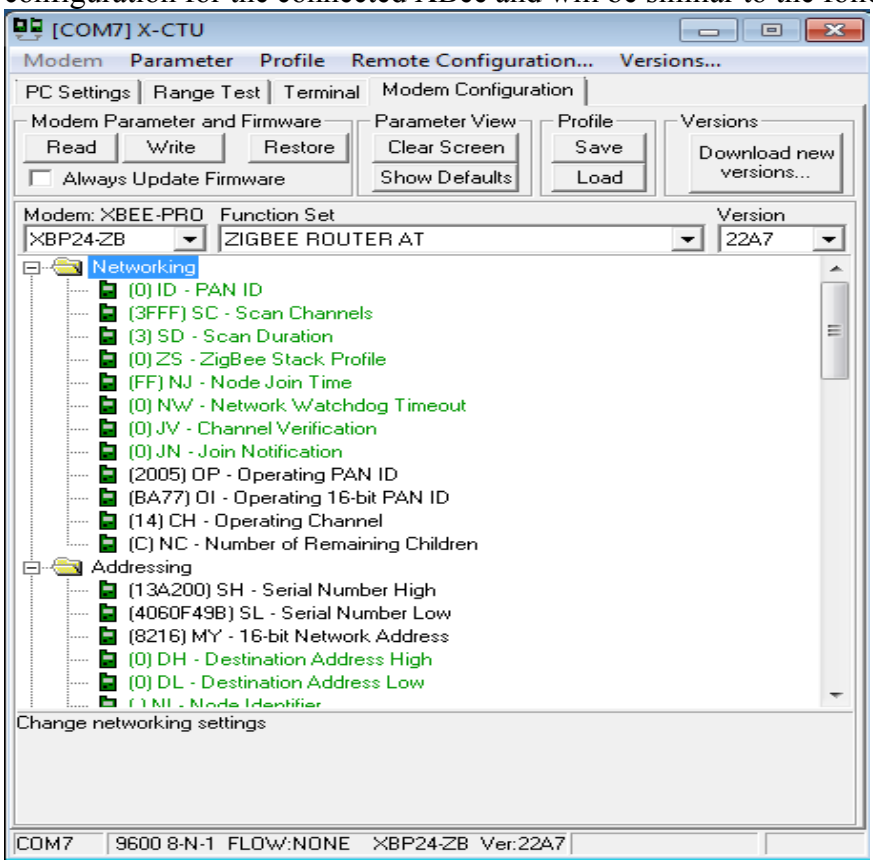
When opening the X-CTU software you should see a window like that shown. After selecting the proper COM port click the Test/Query button.



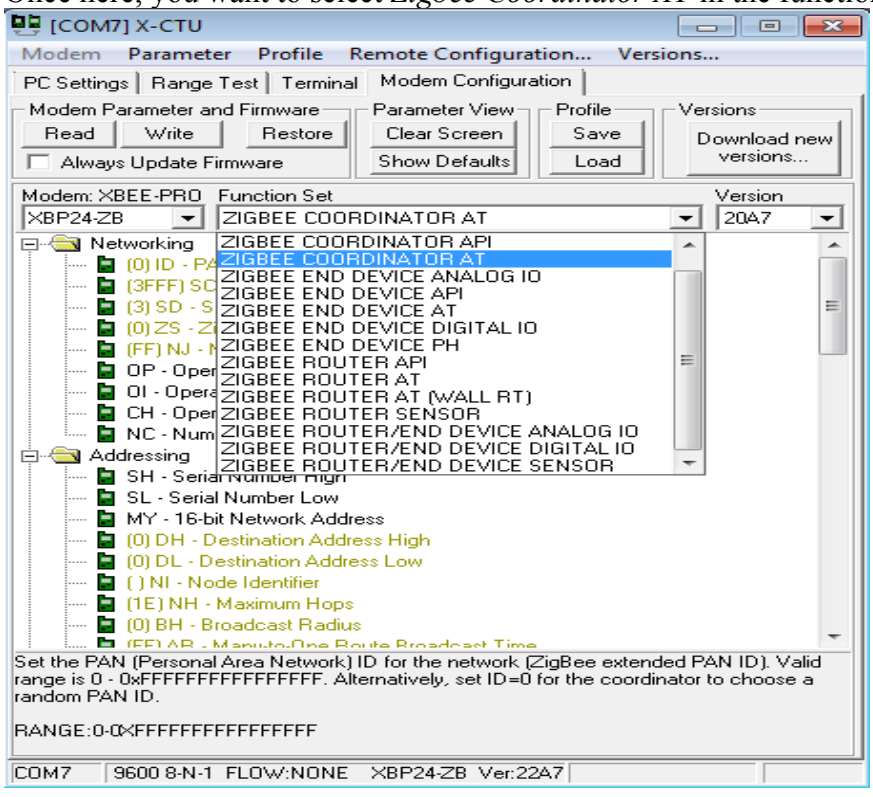
After selecting the Test/Query button, you should see a dialogue box like the one below. You will want to record the serial number shown, as you will need it in a couple minutes.



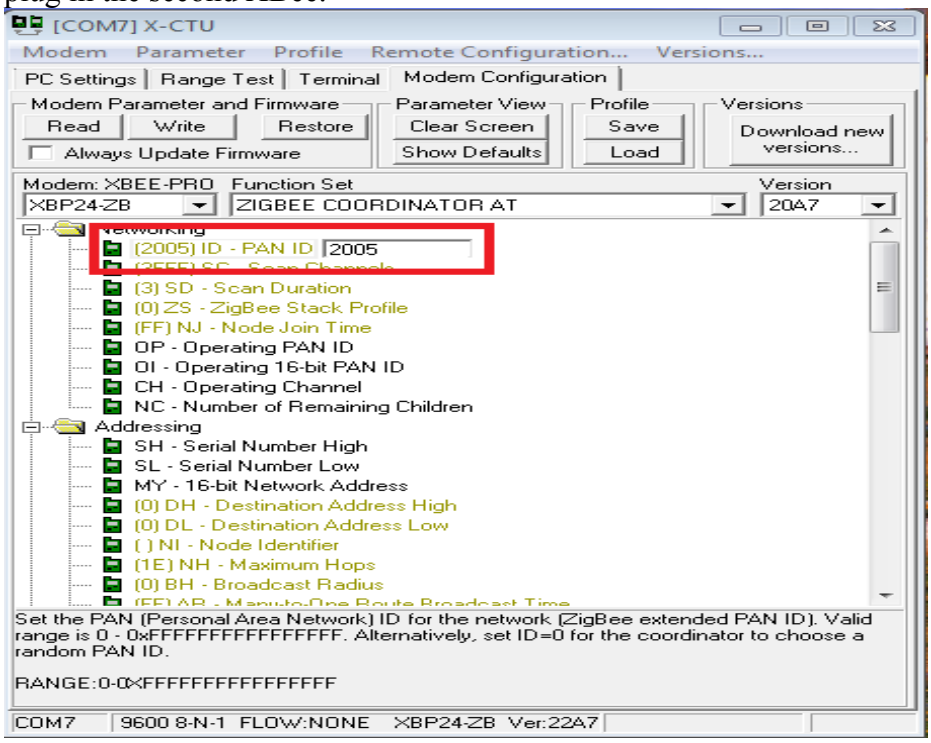
After recording the serial number you can click OK. Next, select the Modem Configuration tab at the top of the window. Once here, select the read button. This will bring up the current configuration for the connected XBee and will be similar to the following:



Once here, you want to select *Zigbee Coordinator AT* in the function set drop down menu.

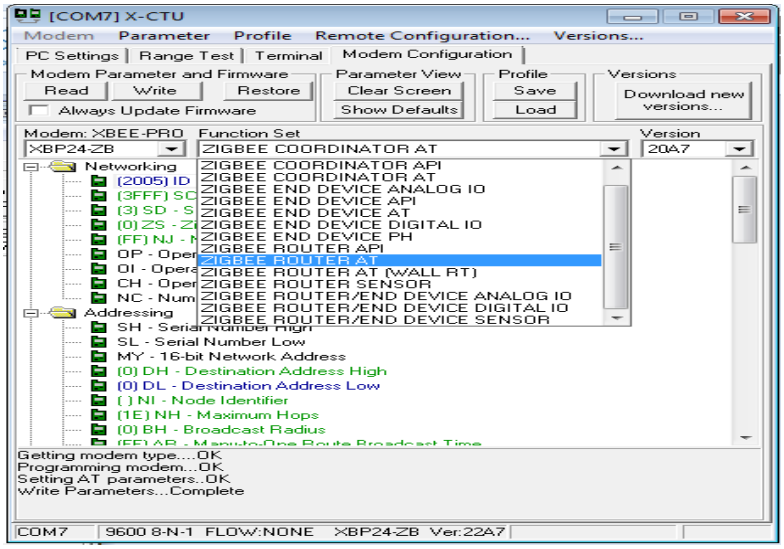


After selecting the coordinator function set, you will need to set the PAN ID. This can be any four-digit number and allows the XBees to distinguish between modules in their network and those from other networks. Once you have added the wanted PAN ID click the "Write" button. This will update and configure the XBee. Once this is done you can disconnect the XBee and plug in the second XBee.

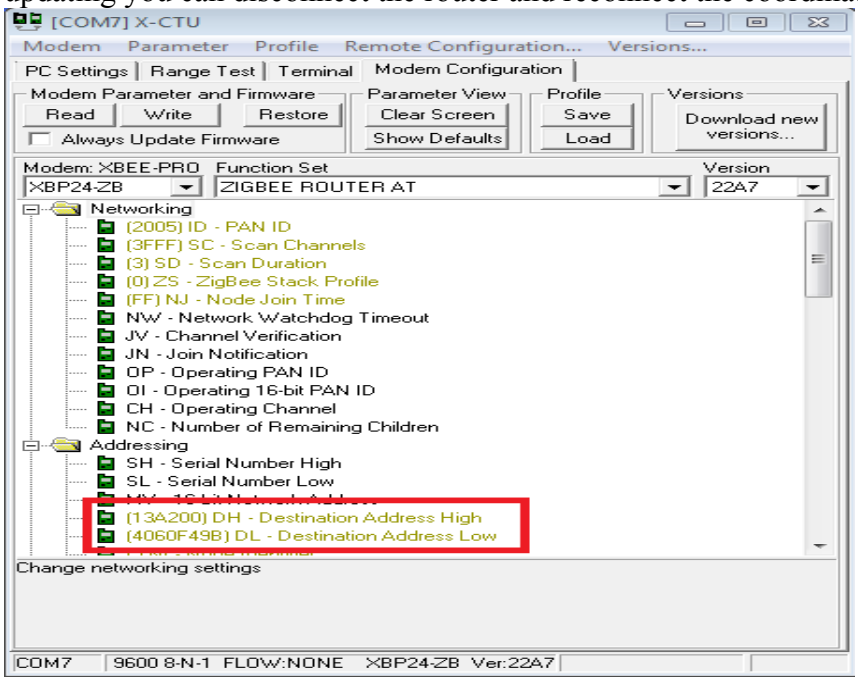


Step 4: Configure 2nd XBee as Router

To configure the 2nd XBee, you will follow the same process as for the coordinator with one difference. In the PC settings tab, again, click the "Test/Query" button and record the serial number. Then in the *Modem configuration* tab, click the read button to load the current configuration of the XBee and set the PAN ID to the same ID used for the coordinator. The only change will be the function set you choose. For the second XBee we will set this as *Zigbee Router AT*.



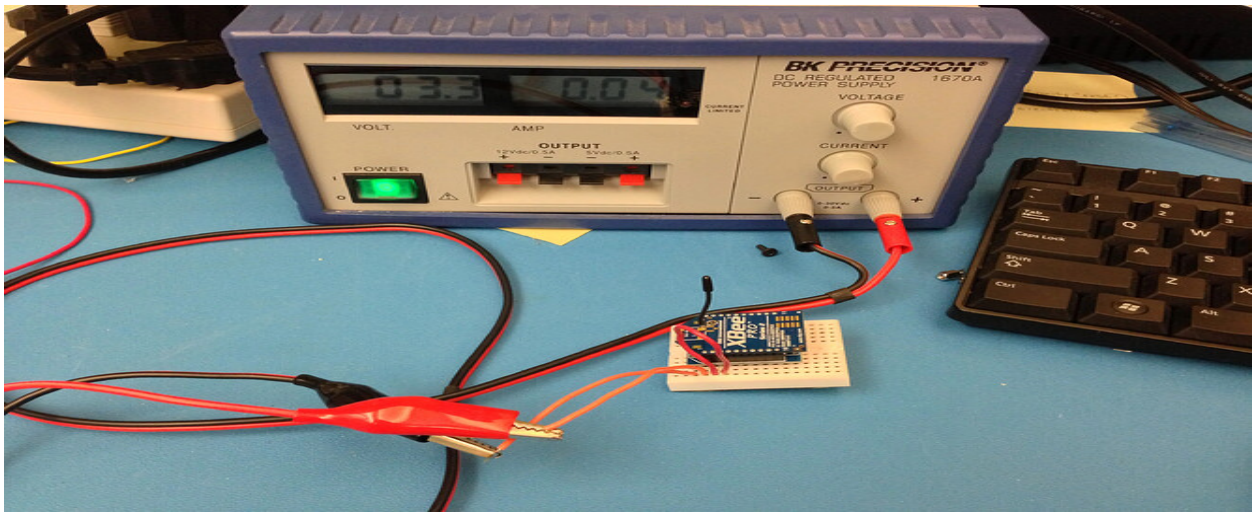
After setting the PAN ID and the function set for the router you will put the recorded serial numbers to use. Enter the first 6 digits of the coordinators serial number into the *Destination Address High* field and the rest of the serial number into the *Destination Address Low* field. Then select the "Write" button to update the configuration settings for the router. Once this is done updating you can disconnect the router and reconnect the coordinator to the computer.



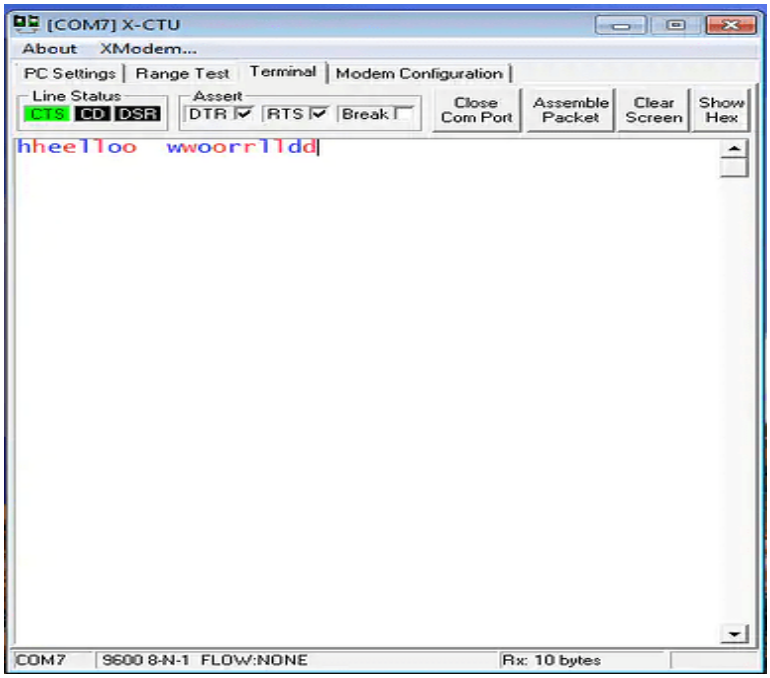
After re-connecting the coordinator to the computer. You will again go into the Modem Configuration tab and click the read button. You will then want to set the Destination Address to the serial number of the Router XBee in the same manner as you just did for the Router. Once done you will again click the write button to update the coordinator XBee settings. After the write process is complete you are ready to use your XBees and communicate wirelessly

Step 5: Test the configuration

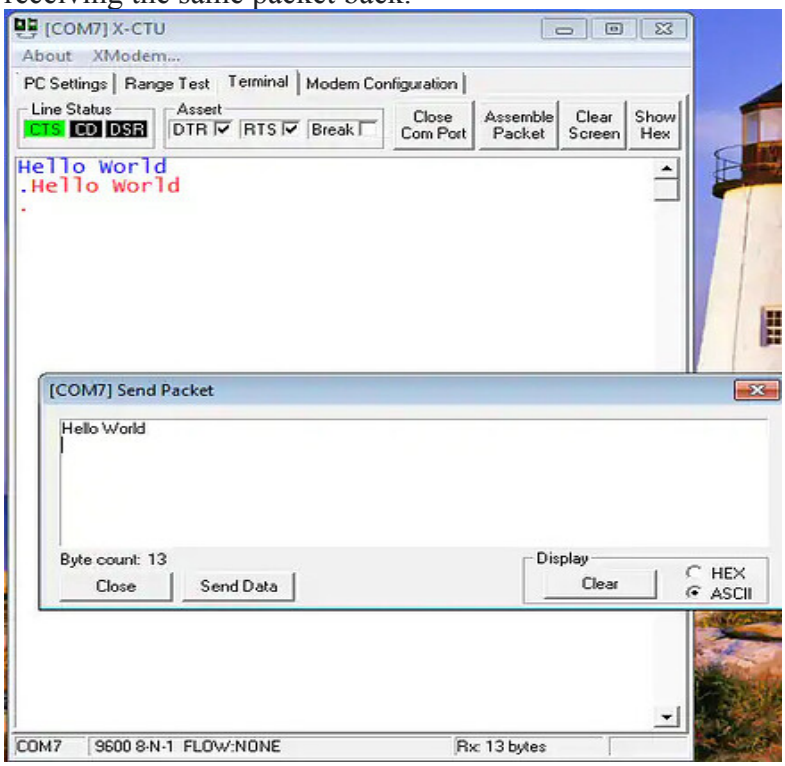
A simple test can be done to be sure the two XBees are communicating properly. You can connect either one of the XBees to the computer. Then, connect the second XBee to 3.3V power and connect the Dout and Din pins together. This will cause the XBee to automatically retransmit any data it receives.



When you have both XBees connected go to the *Terminal* tab in the X-CTU window. Whatever you type in the terminal window will appear in blue font and whatever is received will appear in red font. If the XBees are configured correctly every character you type should be mirrored in red.



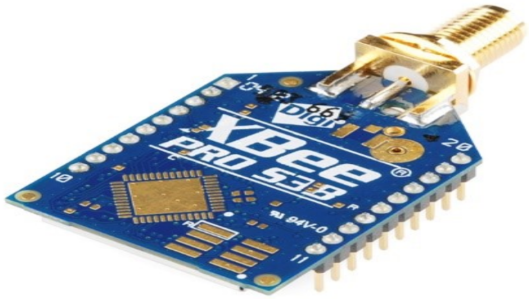
When typing single characters, you should see a screen similar to the one above. To send strings of data you can assemble a data packet. To do this click the "Assemble Packet" button and type the wanted string into the box then click send data. This will send the entire packet before receiving the same packet back.



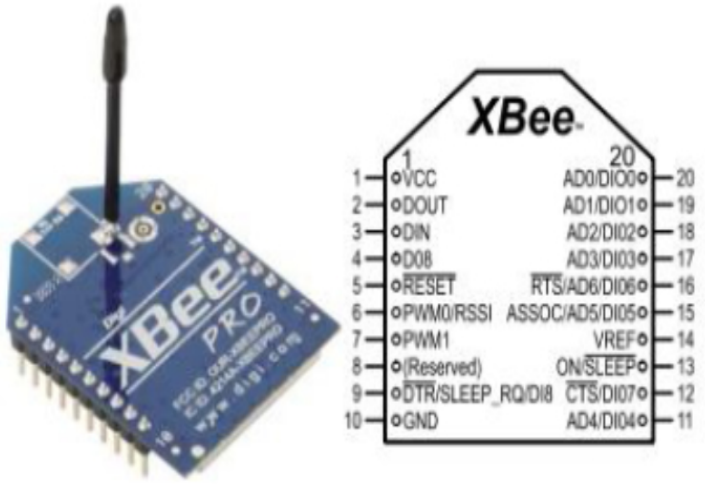
If everything you type is being reflected back in red, you are successfully transmitting and receiving with your XBees!! You are now ready to use them however you wish. You can connect them to any microcontroller and transmit data through the UART peripheral or you can connect each XBee to a different computer and have a chat application. There are many more possibilities that are now at your fingertips with your working XBees.

About Zigbee

Zigbee is a wireless communication protocol targeted for battery-powered devices (it has both low power and low cost). It generally operates in the 2.4GHz range (although there are geographic variations), and supports data ranges from 20 to 250 kbits/s. The transmission distance though, is small compared to the likes of LoRa. It is 10 to 100 m, whereas LoRa can transmit over a few kilometers. Another thing to note is that Zigbee communication doesn't work very well if there is no line of sight between transmitter and receiver. Even minor obstacles have been observed to significantly degrade the communication. Keep these limitations in mind when using Zigbee. You may want to look out for other options if your application can't meet these constraints. In order to make Zigbee work with Arduino, we will use the XBee module.



These works with UART and therefore, it is fairly easy to interface them with Arduino. It is important to look at the pinout of XBee though, to understand which are the UART pins –



The DOUT and DIN pins in the figure above are the UART pins (TX and RX). They can be connected to two digital pins of Arduino (if you plan to use SoftwareSerial), or else to pins 0 and 1 of Arduino respectively (if you plan to use HW Serial). Please note that you won't be able to read print statements from the Arduino on the Serial Monitor if you use Hardware Serial for Zigbee interface.

Configuring the XBee modules

The XBee modules (transmitter and receiver) need to be configured using the X-CTU Software. It can be downloaded from XTCU6.3, this software is provided by DigiKey, and they have given a detailed configuration guide. Therefore, there is no point of me reinventing the wheel here.

There's another one by Sparkfun that is adapted to the newer version of the X-CTU software.

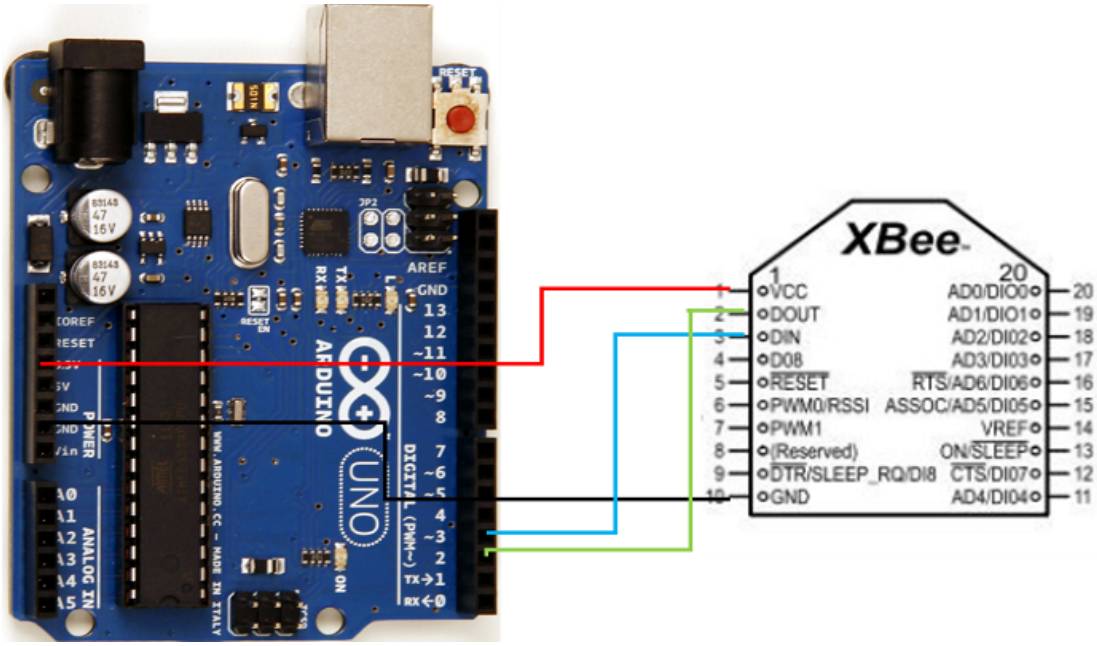
Please note that the two XBee modules that intend to communicate with each other should belong to the same series.

Here are a few things to note about the configuration −

- You will need a breakout board or an Explorer with a USB to UART converter for this configuration.
- The PAN ID (Personal Area Network ID) has to be the same for the devices that want to communicate with each other.
- One module needs to be set as the transmitter and the other as the receiver (this is determined by the CE field).
- Note the baud rate that you set. This will be used in the Arduino code, when configuring the Serial communication with XBee.

Circuit Diagram

Once your XBee is configured, you can connect it to the Arduino via the breakout/Explorer board. In that case, the pinout will be slightly different depending on which board/ Explorer you are using. Here we will assume you are connecting the XBee directly to Arduino Uno, in which case, the connections will be –



As you can see, we have connected Vcc to 3.3V on Arduino, GND to GND, DOUT (TX) to pin 2, which will act as RX on the Arduino, and DIN (RX) to pin 3, which will act as TX on the Arduino.

The connections will be similar on the receiving side as well. If you have an on-board antenna, that's good, else you'll have to connect an antenna to the UFL connector.

Code

The code is quite straightforward. If you are using a board other than Arduino Uno, all digital pins may not support Software Serial.

On the transmitting side, the code will be −

```
#include <SoftwareSerial.h>
SoftwareSerial xbeeSerial(2,3); //RX, TX

void setup() {
        Serial.begin(9600);
        xbeeSerial.begin(9600);
}

void loop() {
        if(Serial.available() > 0){
        char input = Serial.read();
        xbeeSerial.print(input);
        }
}
```

As you can see, whatever the user on the Serial Monitor sends is sent to the XBee module, and it will be received on the receiving side. The code for the receiving side is −

```
#include <SoftwareSerial.h>
SoftwareSerial xbeeSerial(2,3); //RX, TX

void setup() {
        Serial.begin(9600);
        xbeeSerial.begin(9600);
}

void loop() {
        if(xbeeSerial.available() > 0){
        char input = xbeeSerial.read();
        Serial.print(input);
        }
 }
```

Over here, whatever is received from XBee is forwarded to the Serial Monitor. Thus, when testing out the combined system, whatever you type on the Serial Monitor on the transmitter side should be printed on the Serial Monitor on the receiver side.

Viva Voce:

1. Explain Zigbee.
2. Write advantages of Zigbee.
3. When was Zigbee comes into the picture.
4. In which sector Zigbee can be implemented.
5. Explain mesh networking.
6. Define topologies.
7. How many USB Ports are there in Raspberry Pi.
8. Explain WLAN.
9. Define VLAN.
10. What is network node.
11. List Zigbee Alliance.
12. Difference between Zigbee and Bluetooth.
13. Define the range of Zigbee.
14. List some shortcomings of Zigbee.
15. Explain MAC Layer.
16. Define IEEE802.15.4.
17. Explain WIfI.
18. Define LoRa.
19. Define Zigbee Coordinator Device.
20. Explain Zigbee Router.
21. Explain Zigbee End Devices.
22. List applications of Zigbee.
23. Mention how many frequency channels are supported in zigbee in different PHY versions.
24. Explain the different components, which form Zigbee network or system.
25. What is Zigbee RF4CE version?
26. What is Zigbee 6LoWPAN version?
27. Explain Zigbee network architecture and how it is established?
28. Explain block diagram of Zigbee physical layer modules.
29. Explain Zigbee frame structure and explain different fields.
30. Explain Zigbee routing protocol.
31. Compare Zigbee with WiFi.