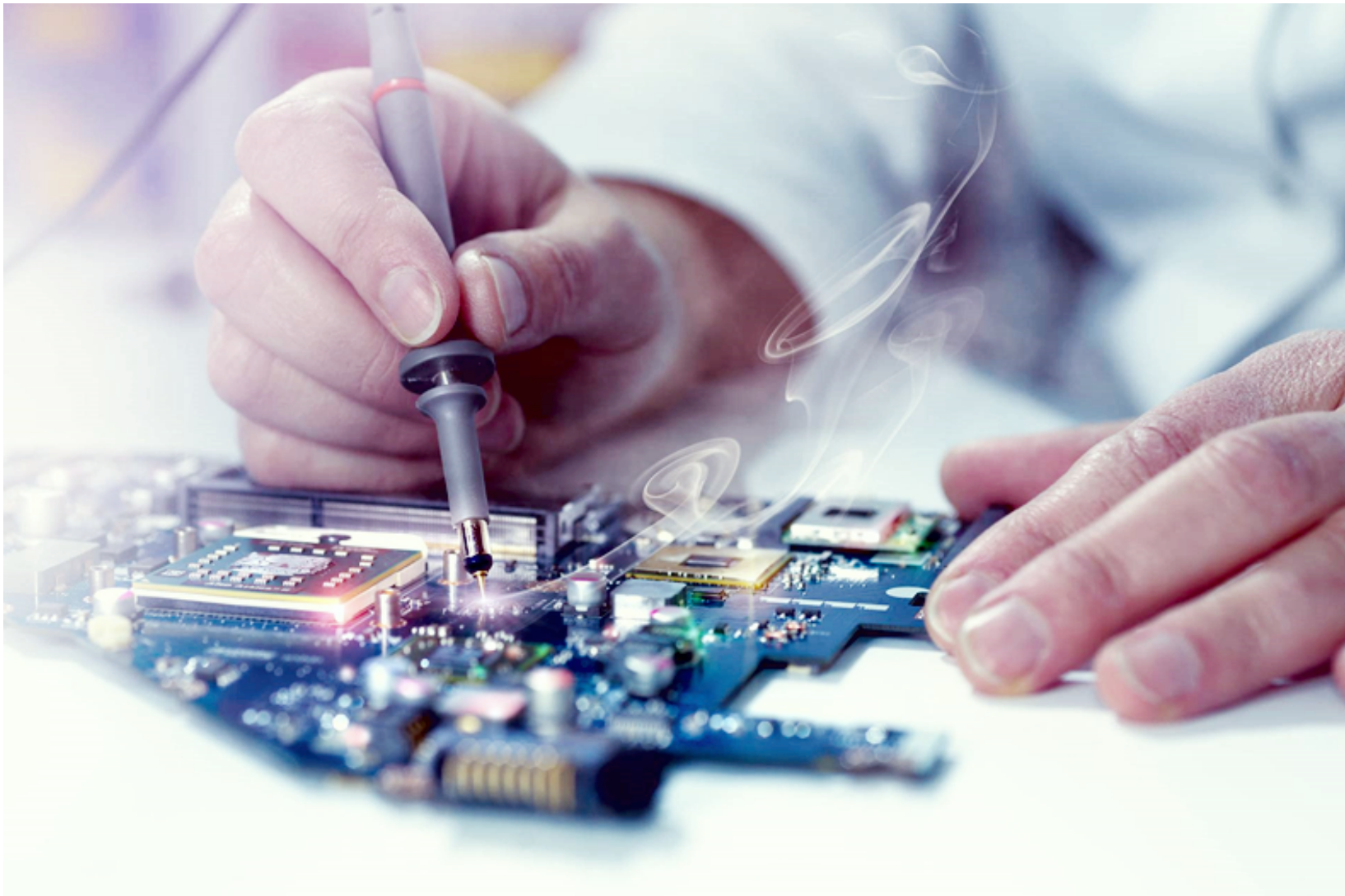


Textbook of
Internet of Things
Hardware Development
Grade - X



National Vocational & Technical Training Commission (NAVTTTC)

Textbook of
Internet of Things
Hardware Development
Grade –X



National Vocational and Technical Training Commission
H-9, Islamabad

Author: **Muhammad Umair** Lecturer, Department of Electrical, Electronics and Telecommunication Engineering, New Campus, University of Engineering & Technology, Lahore.

Reviewers:

Dr. Ahmed Mustafa

Dr. Hina Khalid

Mr. Muhammad Atif Anees

Designing: Gul Awan Printers, Blue Area, Islamabad.

Edition: Test Edition, 2022

ISBN:

Publishers: National Vocational & Technical Training Commission H-9, Islamabad.

Website: www.navttc.gov.pk,

All rights are preserved with the National Vocational and Technical Training Commission. No part of this book can be copied, translated, reproduced or used for guide books, key notes, helping books etc. without permission of NAVTTC.

PREFACE

This book is written to meet the requirements of Matric Tech to train the students in the Internet of Things (IoT) trade. This book is specific for IoT hardware development. IoT Matric Tech has been introduced first time in the history of Pakistan. Effort has been made to describe the topics related to IoT hardware development in one book. A key attempt has been made to make the book interesting and useful. All the chapters cover the basic details understandable to the students of Matric Tech. All chapters include assessments in form of MCQs, short questions and long questions.

All the topics are explained in a convenient way understandable to the students of Matric Tech. The book starts with the concepts of digital logic design and microcontrollers. A separate chapter on Projects is also included in this book. In the end, freelancing and entrepreneurship in IoT field have been described.

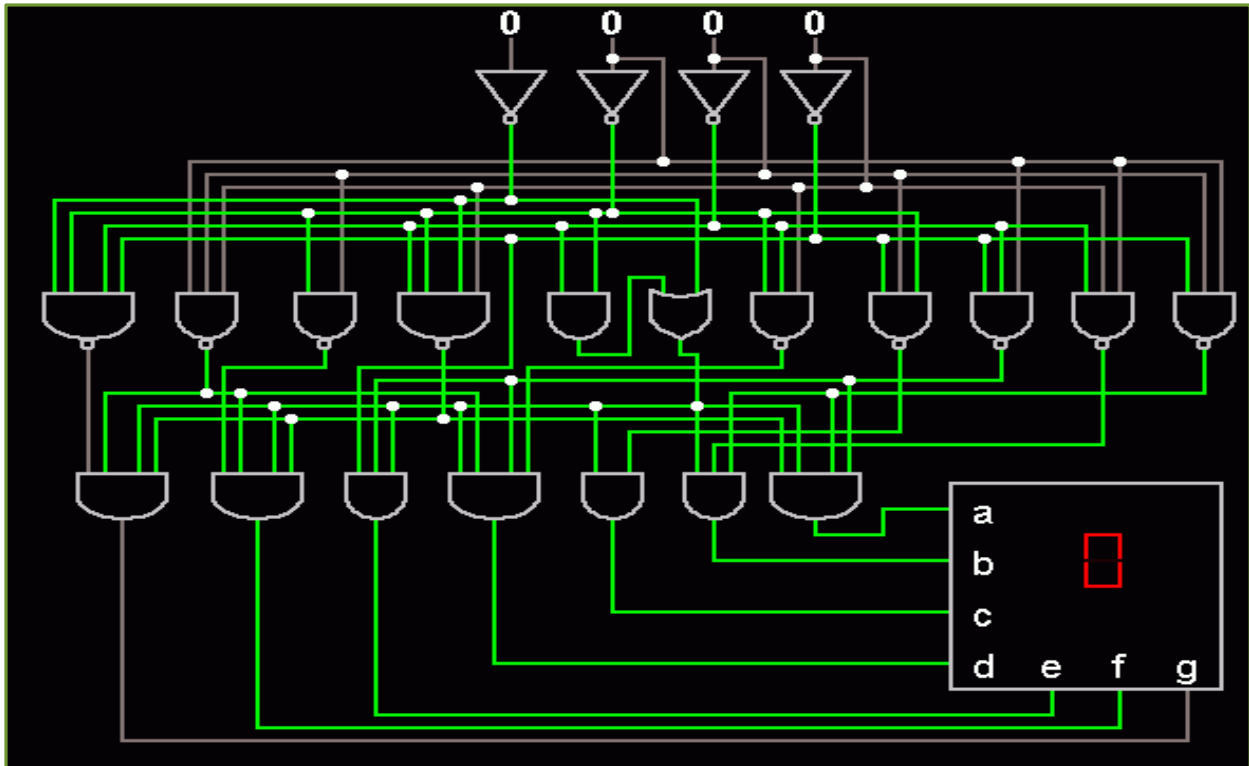
The book should be read conceptually rather than being crammed. Perform all the activities and tasks to have hands on experience on IoT. The sequence of chapters can be adjusted as per the convenience of the tutor. Make sure you never skip a pre-requisite of any chapter. Suggestion from experts, teachers and students for improvement in this book will be highly appreciated.

Executive Director
National Vocational & Technical Training Commission
(NAVTTTC)

Table of Contents

Sr. #	Chapter Name	Page #
1.	Combinational Logic Design	1
2.	Sequential Logic Design	14
3	IoT Networking and Interfacing of Sensors	25
4.	IoT Networking and Interfacing of Communication Modules	38
5	Project	56
6	Freelancing	70
7	Entrepreneurship	86

Chapter 1: Combinational Logic Design



After studying this chapter, you will be able to:

- understand the basics of digital systems.
- differentiate between analog and digital circuits.
- understand the working principle of digital gates.
- identify the symbol of logic gate, IC & logic function.
- connect LED to the output pin of IC and apply different logics and input pins.
- record & verify the output result against each given input.
- describe working of half adder.
- understand construction of half adder on the bread board using digital gates.
- construct, and test a half-adder circuit.
- record & verify the output result against each given input.
- describe working of full adder.
- understand construction of full adder on the breadboard using digital gates.
- construct, and test a full-adder circuit.
- record & verify the output result against each given input.

1.1 Digital Systems

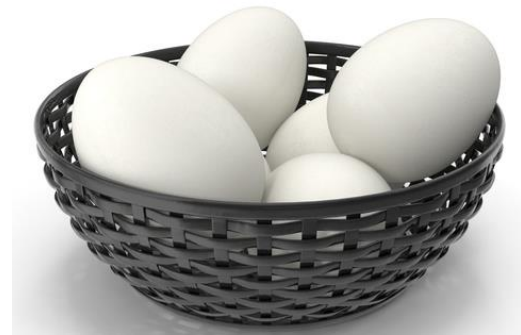
A digital system deals with discrete signals. The input and output of a digital system is a binary value which is either 0 or 1. Examples of digital systems are mobile phones, digital clocks and laptops etc.

1.2 Difference between Analog and Digital Systems

Electronics can be divided into two broad categories: analog and digital. One of the most common examples of the difference between analog and digital devices is a clock. On the analog clock, the time is represented by hands that spin around a dial and point to a location on the dial that represents the approximate time. Whereas, on a digital clock, a numeric display indicates the exact time.

Analog systems refer to circuits in which quantities such as voltage or current vary at a continuous rate. When you turn the dial of a potentiometer, for example, you change the resistance by a continuously varying rate. The resistance of the potentiometer can be any value between the minimum and maximum allowed by the pot. If you create a voltage divider by placing a fixed resistor in series with a potentiometer, the voltage at the point between the fixed resistor and the potentiometer increases or decreases smoothly as you turn the knob on the potentiometer.

In digital electronics, quantities are counted rather than measured. When you count something, you get an exact result. Whereas, when you measure something, you get an approximate result. Consider a cake recipe that calls for 2 cups of flour, 1 cup of milk, and 2 eggs. To get 2 cups of flour, you scoop some flour into a 1-cup measuring cup, pour the flour into the bowl, and then do it again. To get a cup of milk, you pour milk into a liquid measuring cup until the top of the milk lines up with the 1-cup line printed on the measuring cup and then pour the milk into the mixing bowl. To get 2 eggs, you count out 2 eggs, and add them to the mixing bowl. The



Count of eggs is a digital value



Amount of flour in a scoop is analog

measurements for flour and milk in this recipe are approximate. A teaspoon too much or too little won't affect the outcome. But the eggs are precisely counted: exactly 2. You can't have a teaspoon too many or too few eggs.

Digital circuits are more accurate because they count with complete precision. For example, you can precisely count the number of jelly beans in a jar. However, if you weigh the jar by putting it on an analog scale, your reading may be a bit imprecise because you can't always judge the exact position of the needle. Say that the needle on the scale is about halfway between 4 pounds and 5 pounds.

On the other hand, digital circuits are inherently limited in their precision because they must count in fixed units. Most digital thermometers, for example, have only one digit to the right of the decimal point. Thus, they can indicate a temperature of 98.6 or 98.7 but can't indicate 98.65.

Do you know? NOR and NAND gates are universal gates. You can create any logical Boolean expression using only NOR gates or only NAND gates.

1.3 Digital Logic Gates

A digital logic gate (also referred as “digital gate”) is a device that acts as a building block for digital circuits. It performs basic logical functions that are fundamental to digital circuits. Most electronic devices we use today contains some form of logic gates. For example, digital gates are used in technologies such as smartphones, tablets or within memory devices.

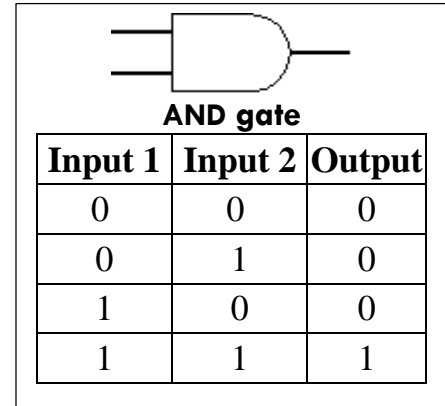
In a circuit, digital gates make decisions based on a combination of digital signals coming from its inputs. Most digital gates have two inputs and one output. Digital gates are based on Boolean algebra. At an instant, every input/output is in one of the two binary conditions, false or true. False represents 0, and true represents 1. Depending on the type of logic gate being used and the combination of inputs, the binary output will differ. A logic gate can be thought of like a light switch, wherein one position the output is off -- 0, and in another, it is on -- 1. Logic gates are commonly used in integrated circuits (IC).

1.3.1 Basic Digital Gates

There are seven basic logic gates: AND, OR, XOR, NOT, NAND, NOR, and XNOR.

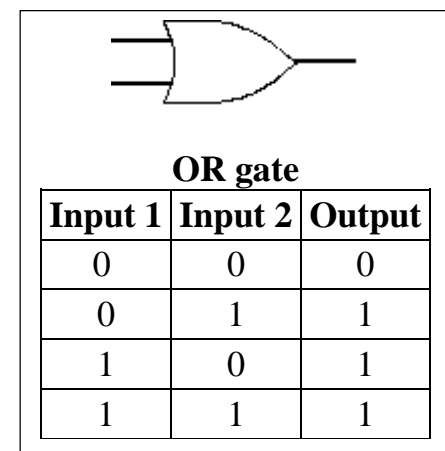
AND Gate

The *AND gate* is named so because, if 0 is called "false" and 1 is called "true," the gate acts in the same way as the logical "and" operator. The following illustration and table show the circuit symbol and logic combinations for an AND gate. (In the symbol, the input terminals are at left and the output terminal is at right.) The output is "true" when both inputs are "true." Otherwise, the output is "false." In other words, the output is 1 only when both inputs one AND two are 1.



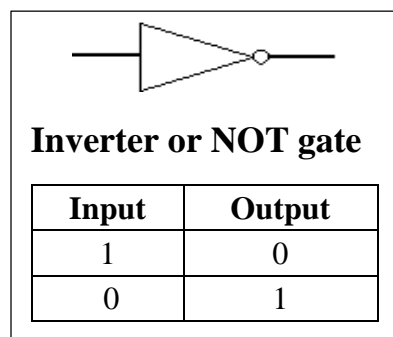
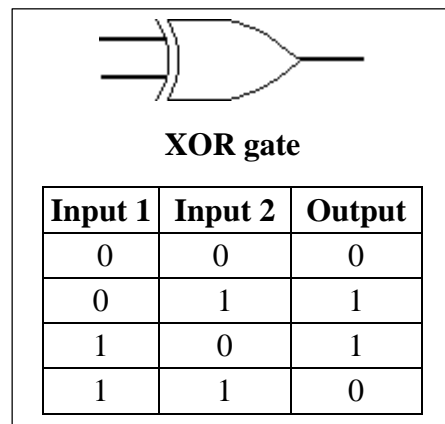
OR Gate

The *OR gate* behaves as logical inclusive "or." The output is "true" if either or both of the inputs are "true." If both inputs are "false," then the output is "false." In other words, for the output to be 1, at least input one OR two must be 1.



XOR Gate

The *XOR (exclusive-OR) gate* acts in the same way as the logical "either/or." The output is "true" if either, but not both, of the inputs are "true." The output is "false" if both inputs are "false" or if both inputs are "true." Another way of looking at this circuit is to observe that the output is 1 if the inputs are different, but 0 if the inputs are the same.

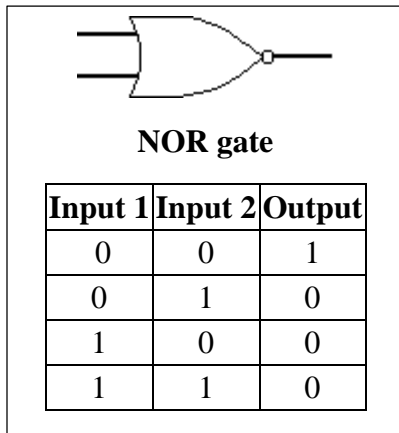
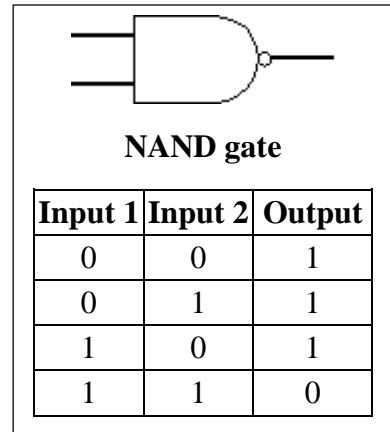


NOT Gate

A logical *inverter*, sometimes called a *NOT gate* to differentiate it from other types of electronic inverter devices, has only one input. It reverses the logic state. If the input is 1, then the output is 0. If the input is 0, then the output is 1.

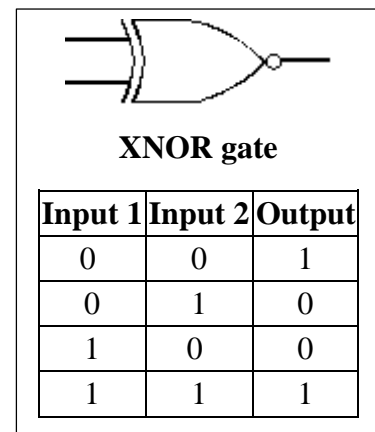
NAND Gate

The *NAND gate* operates as an AND gate followed by a NOT gate. It acts in the manner of the logical operation "and" followed by negation. The output is "false" if both inputs are "true." Otherwise, the output is "true."



NOR Gate

The *NOR gate* is a combination OR gate followed by an inverter. Its output is "true" if both inputs are "false." Otherwise, the output is "false."



XNOR Gate

The *XNOR (exclusive-NOR) gate* is a combination XOR gate followed by an inverter. Its output is "true" if the inputs are same, and "false" if the inputs are different.

Complex operations can be performed using combinations of these logic gates. In theory, there is no limit to the number of gates that can be arrayed together in a single device. But in practice, there is a limit to the number of gates that can be packed into a given physical space. Arrays of logic gates are found in digital ICs. As IC technology advances, the required physical volume for each individual logic gate decreases and digital devices of the same or smaller size become capable of performing ever-more-complicated operations at ever-increasing speeds.

1.4 Verify the Operation of Gates on Hardware

The following steps explain the procedure to verify the working of a digital gate:

- First, place the gate IC on the breadboard as shown in the figure 1.1.
- Identify the pin configuration of gate IC.
- In this example, working of NOR gate is verified.
- Switches are connected with the inputs of a particular gate.

- Output is connected to LED.
- Output of Circuit is recorded for each combination in the truth table.

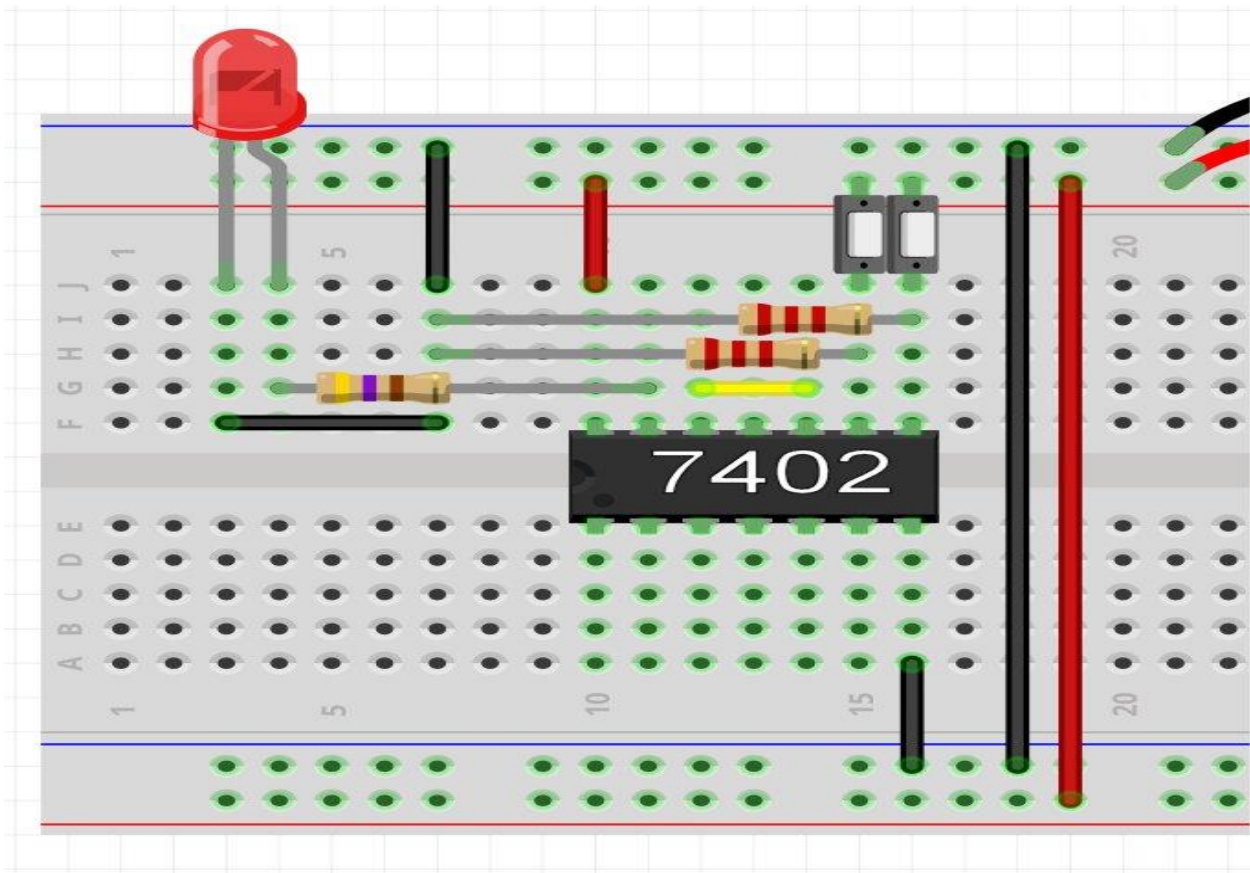


Fig. 1.1 Connection diagram of 7402

1.5 Half Adder

A half adder is used to add two single-digit binary numbers and results into a two-digit output. The adder works by combining the operations of basic logic gates, with the simplest form using only a XOR and an AND gate. This can also be converted into a circuit that only has AND, OR and NOT gates. This is especially useful since these three simpler logic gate ICs (integrated circuits) are more common and available than the XOR IC, though this might result in a bigger circuit since three different chips are used instead of just one.

Half Adder truth table

A (input)	B (input)	C (output)	S (output)
0	0	0	0
1	0	0	1
0	1	0	1
1	1	1	0

Working of Half Adder

The half adder circuit performs binary addition on two one-bit binary inputs. The output of the circuit are the sum and the carry bits.

In binary addition

$0 + 0 = 0$ (sum = 0, carry = 0)

$0 + 1 = 1$ (sum = 1, carry = 0)

$1 + 0 = 1$ (sum = 1, carry = 0)

$1 + 1 = 10$ (sum = 0, carry = 1)

The logic of this process expressed as a truth table is:

Input		Output	
A	B	Sum	Carry Out
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Note that the sum bit is equivalent to the Exclusive OR (XOR) operation: if A or B is 1 the output is 1, but if both A and B equal 0 or 1 then the output is 0. Observe that the carry bit is equivalent to the AND operation: if both A and B are 1 then the output is 1, otherwise the output is low.

The half adder circuit performs the operation by routing the A and B inputs to XOR gate and AND gate.

The half adder is implemented below (see Figures 1.2a, 1.2b and 1.2c) using 74HCxx series high-speed CMOS digital logic ICs: the SN74HC86 (7486) and SN74HC08 (7408).

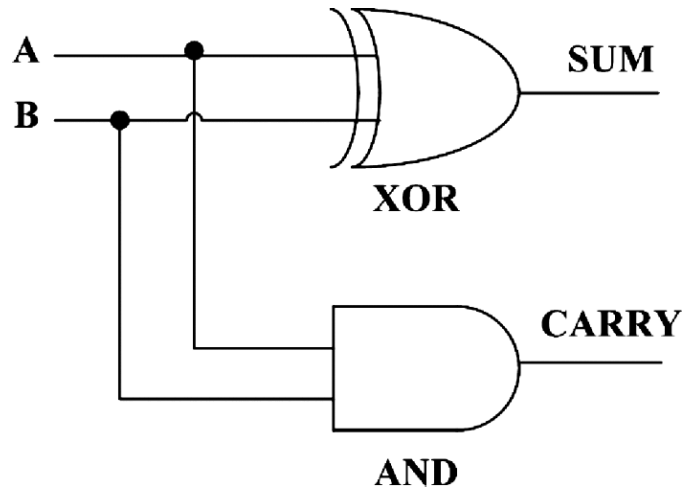


Fig. 1.2a Circuit Diagram of Half-Adder

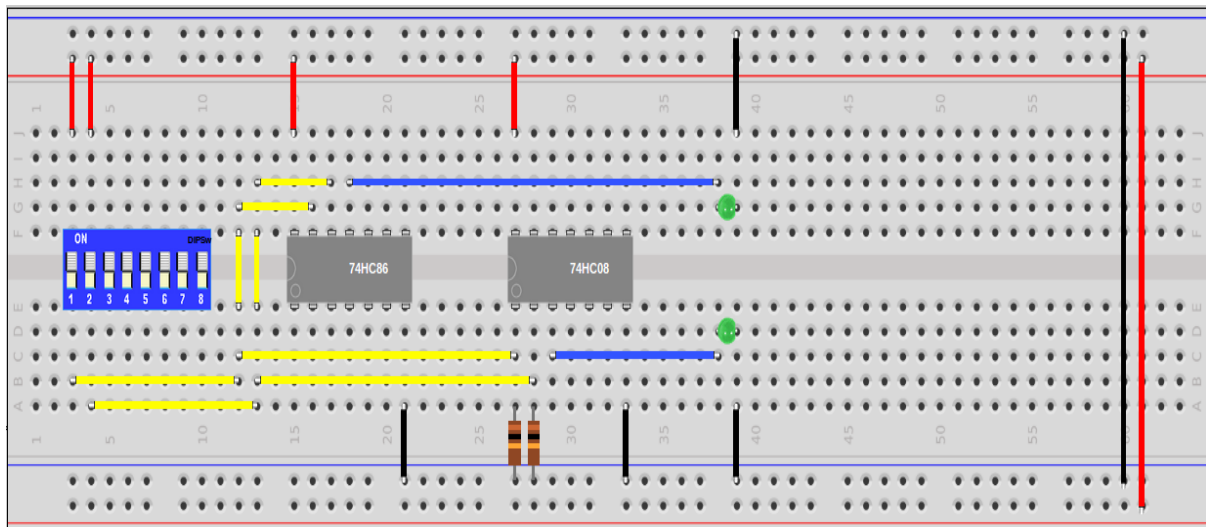


Fig. 1.2b Implementing half-adder on breadboard

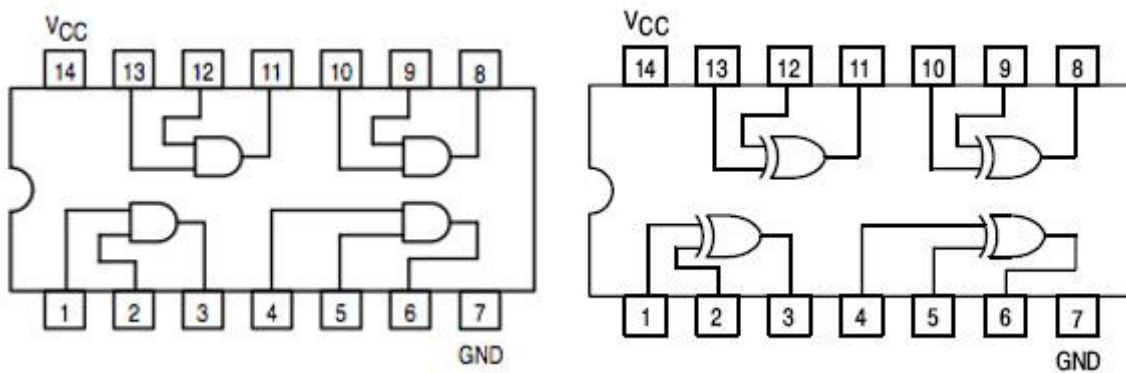


Fig. 1.2c Pin Configuration of 74HC86

1.6 Full Adder

We have seen in section 1.5 that computers use single bit binary numbers 0 and 1 for addition and create SUM and Carry-Out. There are two types of Adders: half adder and full adder. In half adder, we can add 2-bit binary numbers but we can't add carry bit in half adder along with the two binary numbers. Whereas, in full adder circuit, we can add carry-in bit along with the two binary numbers. It is named as such because putting two half adders together with the use of an OR gate results in a full adder. We can also add multiple bits binary numbers by cascading the full adder circuits which we will see later in this section. We will also use **IC 74LS283N** to practically demonstrate the full adder circuit.

Full Adder Circuit

In Full Adder Circuit we have three inputs A , B and $Carry-In$ and we will get final output SUM and $Carry-Out$. So, $A + B + CARRY IN = SUM$ and $CARRY OUT$. See Figure 1.3 below.

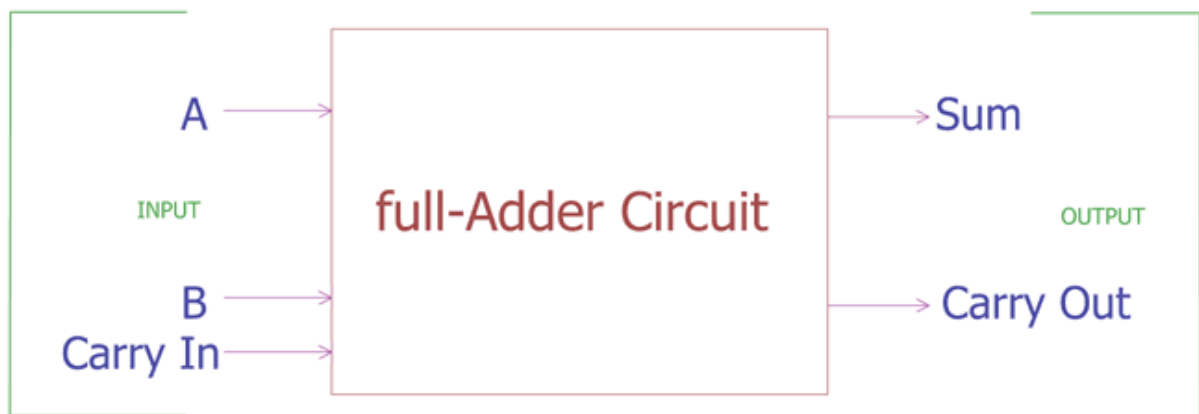


Fig. 1.3 Full Adder

We add two **half adder circuits** with an extra addition of **OR gate** and get a complete **full adder circuit**.

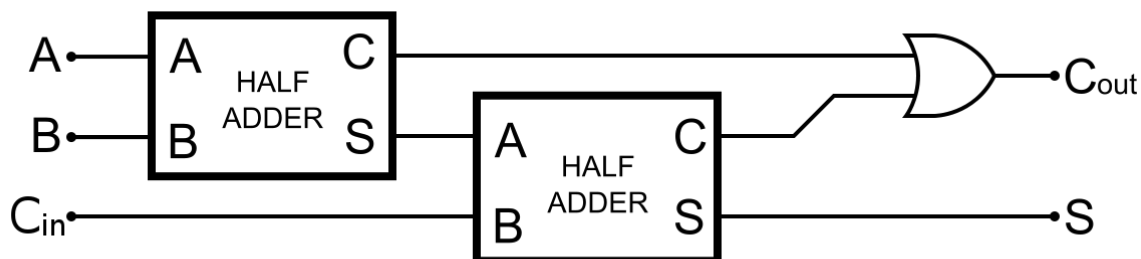


Fig. 1.4 Full Adder using two half adders

Construction of full adder circuit is shown in the above block diagram (Figure 1.4), where two half adder circuits are added together with an OR gate. The first half adder circuit has two binary inputs A and B. It produces two outputs, *SUM* and *Carry-out*. First half adder circuit's *SUM* output is further provided to the second half adder circuit's input. The *Carry-in* bit is input of second half order circuit. It will also provide *SUM*-out and *Carry-out* bit. This *SUM* output is the final output of the Full adder circuit. Whereas, the *Carry-out* of First half adder circuit and the *Carry-out* of second adder circuit is further provided into OR gate. After logic OR of two Carry outputs, we get the final carry-out of full adder circuit. The final carry-out represents the most significant bit or MSB (Most significant bit).

Truth Table of Full Adder Circuit

Full adder circuit deals with three inputs and two outputs as shown in truth table.

<i>Carry In</i>	<i>Input A</i>	<i>Input B</i>	<i>SUM</i>	<i>Carry Out</i>
0	0	0	0	0
0	1	0	1	0
0	0	1	1	0
0	1	1	0	1
1	0	0	1	0
1	1	0	0	1
1	0	1	0	1
1	1	1	1	1

We can also express the full adder circuit in Boolean expression. For *SUM*, we first XOR the A and B input then we again XOR the output with *Carry-in*. So, the Sum is $(A \text{ XOR } B) \text{ XOR } C$. We can also express it as $(A \oplus B) \oplus \text{Carry-in}$. For *Carry out*, it is $A \text{ AND } B \text{ OR } \text{Carry-in}$ $(A \text{ XOR } B)$, which is further represented by $A.B + (A \oplus B)$.

1.6.1 Hardware Implementation of Full Adder

The full adder combines two half adders and adds an OR gate to the circuit. The circuit is called a full adder because the third bit is used to bring in the carry bit from a half adder or another full adder. The ability to integrate a *carry-in* bit enables multiple full adders to be chained together to build adders that can handle 4-bit, 8-bit and larger additions.

The full adder is implemented using 74HCxx series high-speed CMOS digital logic ICs. You can follow the wiring diagram given below and verify the operation of a full adder.

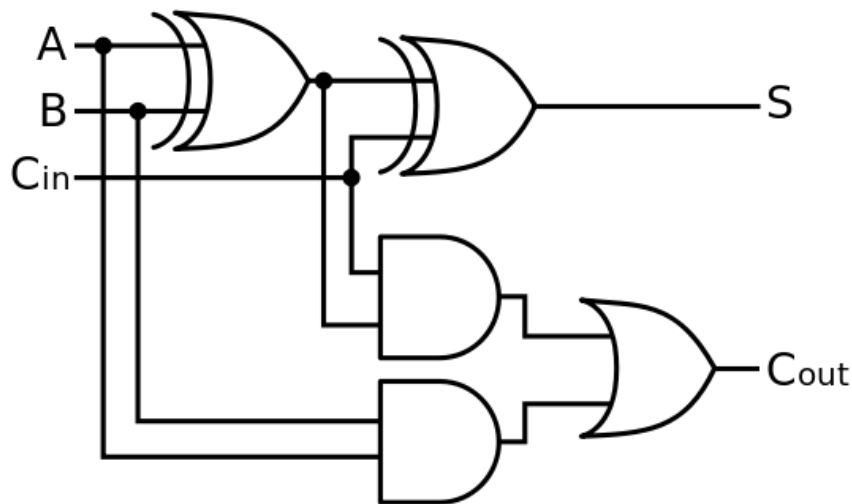


Fig. 1.5a Circuit Diagram of Full Adder

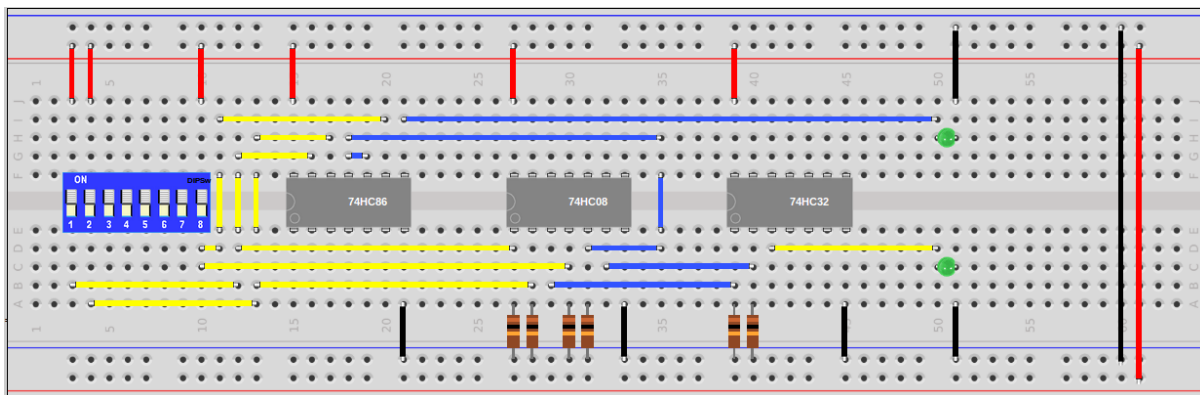


Fig. 1.5b Implementing Full Adder on breadboard

Teacher Notes: Colab is a Google Research product, which allows developers to write and execute Python code through their browser.

Key Points

- A digital system is a system which deals with discrete signal.
- The input and output of digital system is a binary value which is 0 and 1.
- The electricity that power plants generate is delivered to consumers over transmission and distribution power lines.
- Analog refers to circuits in which quantities such as voltage or current vary at a continuous rate.
- In digital electronics, quantities are counted rather than measured.
- There are seven basic logic gates: AND, OR, XOR, NOT, NAND, NOR, and XNOR.
- The XOR (exclusive-OR) gate acts in the same way as the logical "either/or." The output is "true" if either, but not both, of the inputs are "true."
- A half adder is used to add two single-digit binary numbers and results into a two-digit output.
- The half adder circuit performs binary addition on two one-bit binary inputs.
- In Full Adder Circuit we can add carry in bit along with the two binary numbers.
- In Full Adder Circuit we have three inputs A, B and Carry In and we will get final output SUM and Carry out.
- The full adder combines two half adders and adds an OR gate to the circuit.
- The ability to integrate a carry-in bit enables multiple full adders to be chained together to build adders that can handle 4-bit, 8-bit and larger additions.

Exercise

Choose the most suitable option

1. Full adder is implemented using ___ half-adders
 - a. 2
 - b. 3
 - c. 4
 - d. None of these
2. Full-adder can give maximum answer of
 - a. 2
 - b. 3
 - c. 4
 - d. None of these
3. Half-adder can give maximum answer of
 - a. 2
 - b. 3
 - c. 4
 - d. None of these
4. Which gate is used for carry in Half-ader?
 - a. OR
 - b. AND
 - c. NOR
 - d. XOR
5. Which gate is used to sum carry 1 and carry 2 in Full-ader?
 - a. OR
 - b. AND
 - c. NOR
 - d. XOR

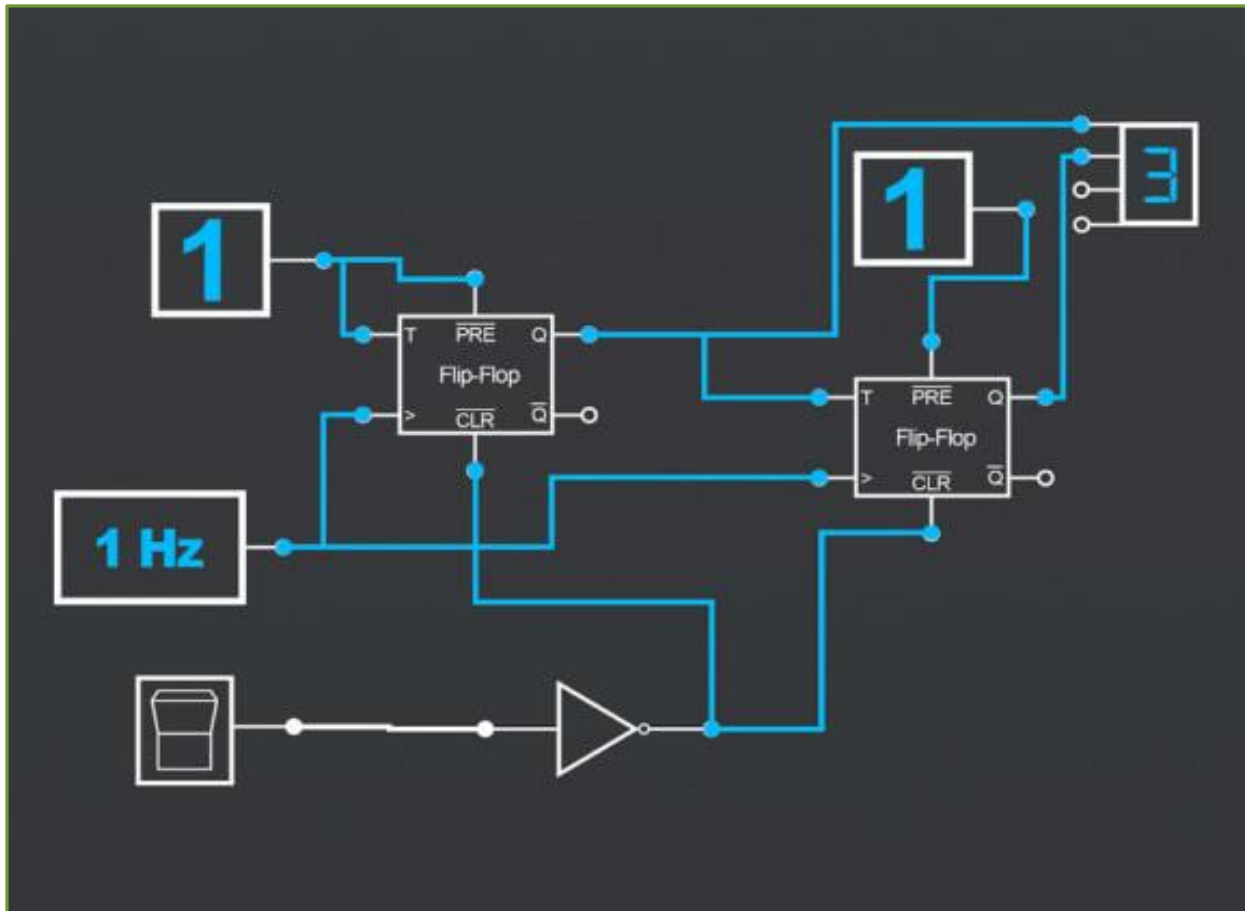
Give short answer to the following questions.

1. Define Digital Circuits.
2. Define combinational circuits.
3. Explain Digital gates with truth tables.
4. Describe Half Adder.
5. Describe Full adder.
6. How many half-adders will be required to build a 16-bit adder.

Practical Tasks

1. Verify the operation of all digital gates on hardware.
2. Verify the truth table of half-adder on actual hardware.
3. Verify the truth table of full-adder on actual hardware.

Chapter 2: Sequential Logic Design



After Studying this chapter, you will be able to:

- understand sequential logic design.
- differentiate between combinational and sequential logic design.
- differentiate between synchronous and asynchronous sequential circuits.
- introduce flipflop.
- understand basic function of flipflops.
- use flipflops in applications of digital systems.
- Understand 4-bit shift register.
- explain the working of 4-bit shift register.
- construct 4-bit shift register using D-flipflops
- record the output according to the input.

2.1 Sequential Logic Circuits

Combinational logic circuits change state depending upon the actual signals being applied to their inputs. Whereas, **Sequential Logic** circuits have some form of inherent built-in memory. Sequential logic circuits also consider their previous input state. The output state of a sequential logic circuit is a function of the following three states: the present input, the past input and/or the past output. Sequential Logic circuits remember these conditions and stay in their current state until the next clock signal (an oscillator signal between a high and a low state) changes one of the states.

The word “Sequential” means that things happen in a “sequence”, one after another. In sequential logic circuits, the clock signal determines the sequence. Simple sequential logic circuits can be constructed from standard bistable circuits such as: *Flip-flops*, *Latches* and *Counters* which can be made by simply connecting universal NAND Gates and/or NOR Gates in a particular combinational way to produce the required sequential circuit.

Sequential logic circuits are generally termed as two state or bistable devices which can have their output in one of the two basic states, a logic level “1” or a logic level “0”. They remain “latched” (hence the name latch) indefinitely in this current state or condition until some other input trigger pulse is applied which causes the bistable to change its state. Figure 2.1 shows the basic representation of sequential logic.

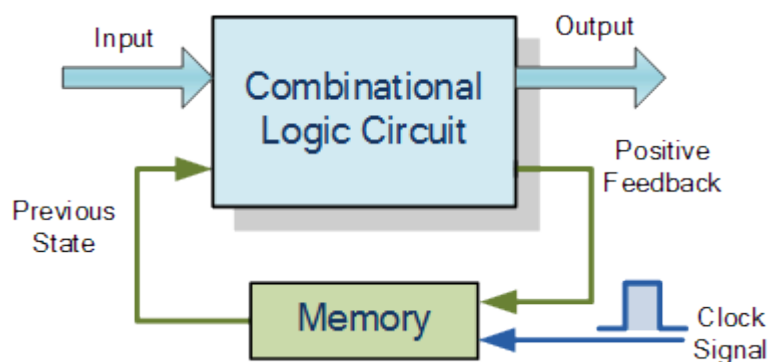


Fig. 2.1 Sequential Logic Representation

2.1.1 Classification of Sequential Logic

As standard logic gates are the building blocks of combinational circuits, bistable latches and flip-flops are the basic building blocks of sequential logic circuits. Sequential logic circuits can be constructed to produce either simple edge-triggered flip-flops or more complex sequential circuits

such as storage registers, shift registers, memory devices or counters. Sequential logic circuits can be divided into the following three main categories (see Figure 2.2):

1. Event Driven: Asynchronous circuits that change state immediately when enabled.
2. Clock Driven: Synchronous circuits that are synchronised to a specific clock signal.
3. Pulse Driven: A combination of the two that responds to triggering pulses.

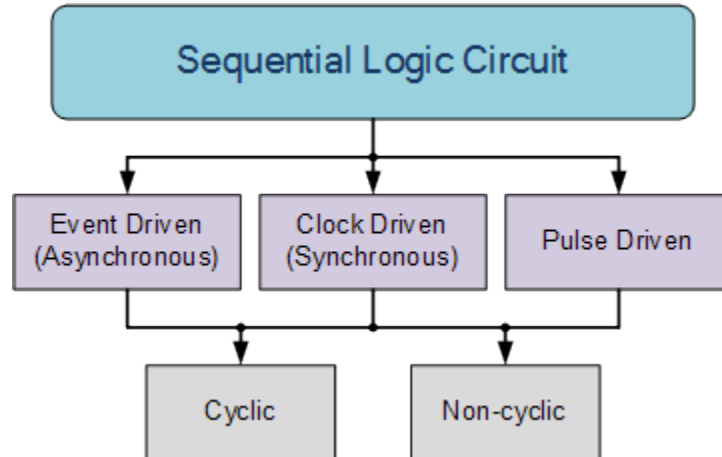


Fig. 2.2 Types of Sequential Logic Circuits

Sequential logic circuits return to their original steady state once reset and sequential circuits with loops or feedback paths are said to be “cyclic” in nature. Sequential circuits change state occur only on the application of a clock signal making it synchronous, otherwise the circuit is asynchronous and depends upon an external input. Sequential circuits rely on feedback to retain their current state. This occurs when a fraction of the output is fed back to the input.

Sequential Feedback Loop

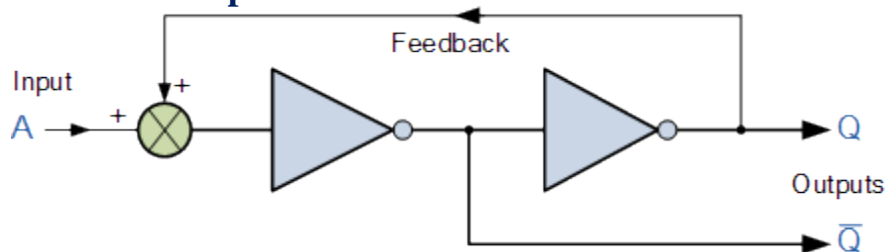


Fig. 2.3 Sequential Feedback Loop

In sequential feedback loop, two inverters or NOT gates are connected in series with the output which is fed back to the input (see Figure 2.3). This configuration never changes state because the output will always be the same, either a “1” or a “0”.

Do you know?

Sequential Circuits are used to design microprocessors and memory chips.

Point to Ponder

➤ Do microprocessors have a clock?

2.2 Flip flops

Flip-flops and latches are fundamental building blocks of digital electronics systems used in computers, communications, and many other types of systems. These are used as data storage elements. It is the basic storage element in sequential logic. A flip flop is an electronic circuit with two stable states that can be used to store binary data. The stored data can be altered by applying varying inputs.

2.2.1 Difference between a Latch and a Flip flop

The basic difference between a latch and a flip-flop is clocking mechanism. Flip Flop is edge-triggered whereas a latch is level-triggered. For instance, in the following circuit (Figure 2.4), when you set S as active, the output Q would be high and Q' will be low. This is an active-low circuit so active here means low, whereas for an active high circuit active means high.

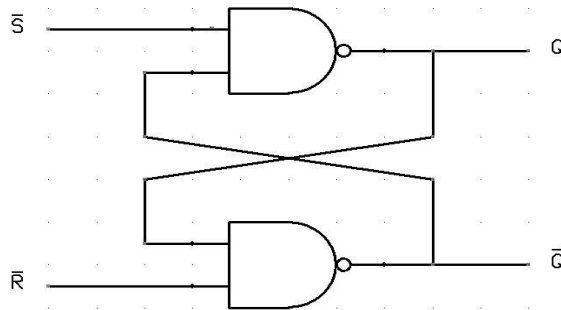


Fig. 2.4 SR Latch

A flip-flop, on the other hand, is synchronous and is also known as a gated or clocked SR latch.

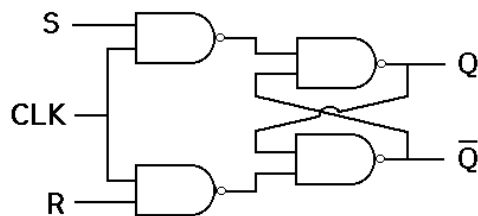


Fig. 2.5 SR Flip flop

In Figure 2.5, the output changes (i.e., the stored data changes) only when you apply an active clock signal. Otherwise, even if the S or R is active the data will not change. The following are the basic types of flip-flops:

SR Flip-flop

The most common flip-flop is SR flip-flop. This simple flip-flop circuit has a *set* input (S) and a *reset* input (R). When you set “S” as active, the output “Q” would be high and “Q’” will be low. Once the outputs are established, the wiring of the circuit is maintained until “S” or “R” go high, or power is

S	R	Q	Q'
0	0	0	1
0	1	0	1
1	0	1	0
1	1	∞	∞

turned off. The two outputs shown above are the inverse of each other. The truth table of SR Flip-Flop is highlighted below:

JK Flip-flop

In SR flip-flop, when both S and R are ‘1’, the output is undefined. JK flip-flop is an improvement of the SR flip-flop where S=R=1 is not a problem. JK flip-flop is shown in Figure 2.6.

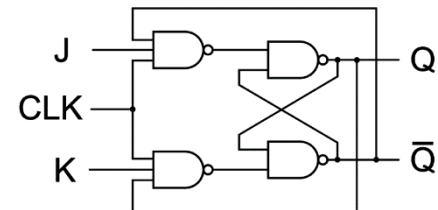


Fig. 2.6 JK Flipflop

When both J and K are ‘1’, we get an inverted version of the existing output state. If J and K data input are different (i.e., high and low) then the output Q takes the value of J at the next clock edge. If J and K are both low then no change occurs. If J and K are both high at the clock edge, then the output will toggle from one state to the other. JK Flip-Flop can function as a Set or Reset Flip-flop.

J	K	Q	Q'
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	1
0	0	1	1
0	1	1	0
1	0	1	1
1	1	1	0

D Flip-flop

D flip-flops are commonly used for counters, shift-registers and input synchronisation. D flip-flop is shown in Figure 2.7.

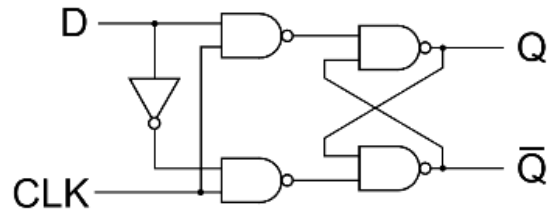


Fig. 2.7 D Flip-flop

In D-Flip-flop, the output can be only changed at the clock edge. The output will be unaffected if the input changes at other times.

Clock	D	Q	Q'
↓ » 0	0	0	1
↑ » 1	0	0	1
↓ » 0	1	0	1
↑ » 1	1	1	0

The change of state of the output is dependent on the rising edge of the clock. The output (Q) is same as the input and can only change at the rising edge of the clock.

T Flip-flop

A T flip-flop is basically a single input version of JK flip-flop. This modified form of JK flip-flop is obtained by connecting both inputs J and K together. It has only one input along with the clock input as shown in Figure 2.8.

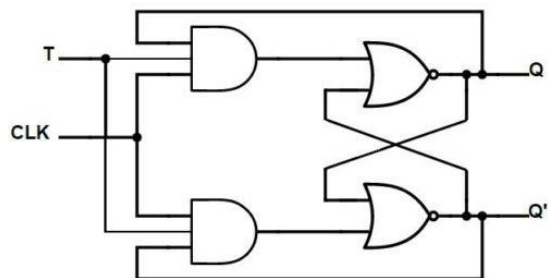


Fig. 2.8 T Flip-flop

These flip-flops are called T flip-flops because of their ability to toggle its state.

T	Q	Q (t+1)
0	0	0
1	0	1
0	1	1
1	1	0

2.2.1 Applications of Flip-flops

Flip-flops are used in the following digital electronic circuits:

- Counters
- Frequency Dividers
- Shift Registers
- Storage Registers

2.2.2 Implementing Flip-flops on Breadboard

The diagram given below presents a wiring diagram for verifying the operation of D flip-flop on a breadboard. Construct the circuit according to the diagram (Figure 2.9-2.10) and verify the operation. Use LEDs to show the output of flip-flop.

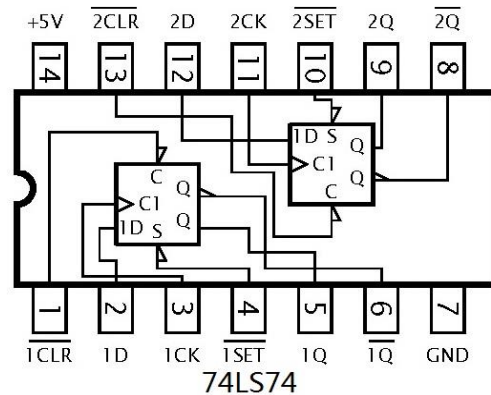


Fig. 2.9 Pin Configuration of 74LS74

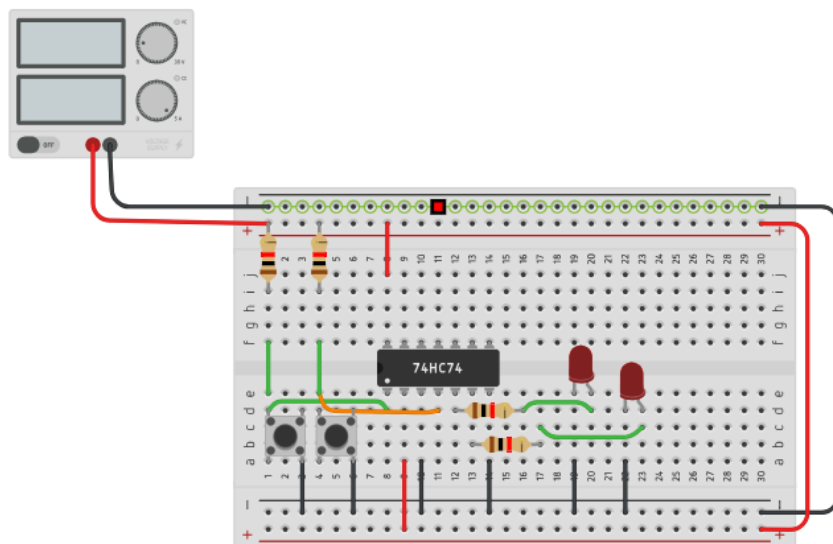


Fig. 2.10 Wiring Diagram of D Flip-flop

2.3 4-bit Shift Register

One flip-flop can store one-bit of information. Multiple flipflops are required to store multiple bits of information. The group of flip-flops, which are used to hold/store the binary data is known as **register**. A register is capable of shifting bits either towards right hand side or towards left hand

side is known as **shift register**. An ‘N’ bit shift register contains ‘N’ flip-flops. The following are the four types of shift registers based on applying inputs and accessing of outputs.

- Serial In – Serial Out shift register
- Serial In – Parallel Out shift register
- Parallel In – Serial Out shift register
- Parallel In – Parallel Out shift register

2.3.1 Serial In – Serial Out Shift Register

The shift register, which allows serial input and produces serial output is known as Serial In – Serial Out (SISO) shift register. The block diagram of a 3-bit SISO shift register is shown in Figure 2.11.

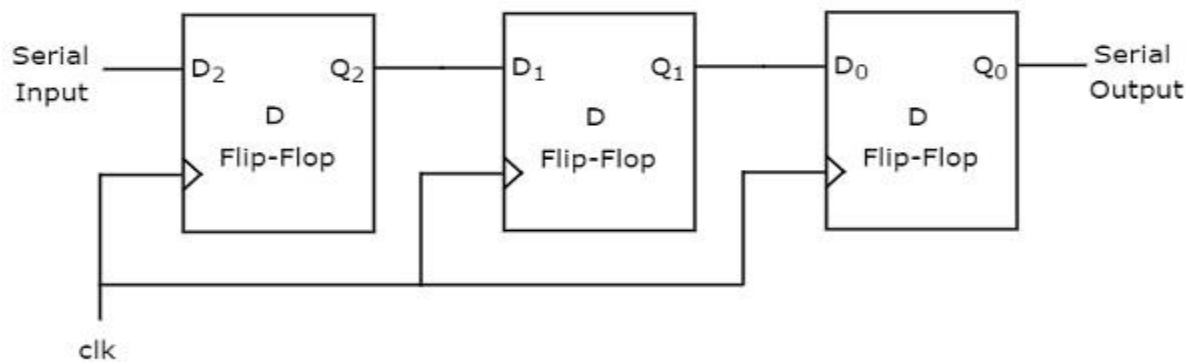


Fig. 2.11 4-bit Shift Register using D Flip-flops

This block diagram consists of three D flip-flops, which are cascaded e.g., the output of one D flip-flop is connected to the input of next D flip-flop. All these flip-flops are synchronous with each other as the same clock signal is applied to each one. In this shift register, the bits can be sent serially from the input of left most D flip-flop. Hence, this input is also called serial input. For every positive edge triggering of clock signal, the data shifts from one stage to the next. Therefore, we can receive the bits serially from the output of right most D flip-flop. Hence, this output is also called serial output.

Example

This example explains the working of a 3-bit SISO shift register by sending the binary information “011” from LSB to MSB serially at the input. Assume, initial status of D flip-flops from leftmost to rightmost is $Q_2Q_1Q_0=000$. We can understand the working of 3-bit SISO shift register from the following table:

No of positive edge of Clock	Serial Input	Q ₂	Q ₁	Q ₀
0	-	0	0	0
1	1LSB	1	0	0
2	1	1	1	0
3	0MSB	0	1	1LSB
4	-	-	0	1
5	-	-	-	0MSB

The initial status of D flip-flops in the absence of clock signal is Q₂Q₁Q₀=000. The serial output is coming from Q₀. So, the LSB 1 is received at 3rd positive edge of clock and the MSB 0 is received at 5th positive edge of clock. Therefore, the 3-bit SISO shift register requires five clock pulses to produce a valid output. Similarly, the N-bit SISO shift register requires 2N-1 clock pulses to shift 'N' bit information.

2.3.2 Implementing Shift Register on Breadboard

The wiring diagram to verify the operation of a 4-bit shift register is constructed using D flipflops on a bread board as shown in Figure 2.12. Construct the circuit according to the diagram and verify the operation. Use LEDs to show the output of the flipflop. It should be noted that a single 74HC74 IC contains two D flipflops. Therefore, only two ICs are required to build this circuit.

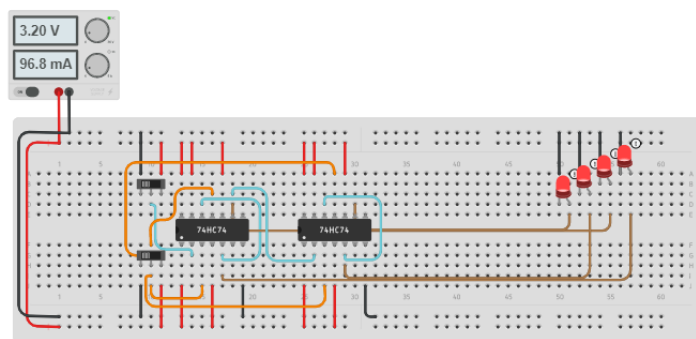


Fig. 2.12 Wiring Diagram of 4-bit Shift Register

Teacher Notes: You can also use 7-segment display instead of four LEDs.

Key Points

- Combinational logic circuits change state depending upon the actual signals being applied to their inputs at that time.
- Sequential Logic circuits have some form of inherent “Memory” built in.
- Sequential logic circuits are generally termed as two state or Bistable devices which can have their output or outputs set in one of the two basic states, a logic level “1” or a logic level “0” and will remain “latched”.
- In Sequential Logic circuits, the actual clock signal determines when things will happen next.
- A flip flop is an electronic circuit with two stable states that can be used to store binary data.
- The basic difference between a latch and a flip-flop is a gating or clocking mechanism.
- Flip Flop is edge-triggered and a latch is level triggered.
- In SR flipflop, when you Set “S” as active the output “Q” would be high and “Q’” will be low.
- The JK flip-flop is an improvement on the SR flip-flop where $S=R=1$ is not a problem.
- In D flipflop, the output (Q) is same as the input and can only change at the rising edge of the clock.
- A T flip-flop is like a JK flip-flop. These are basically a single input version of JK flip-flops.
- One flip-flop can store one-bit of information.
- The group of flip-flops, which are used to hold/store the binary data is known as register.

Exercise

Choose the most suitable option

1. Which of the following flip-flop follows input.
 - a. SR Flip-flop b. D Flip-flop c. T Flip-flop d. None of these
2. Which of the following flip-flop has undefined state?
 - a. SR Flip-flop b. D Flip-flop c. T Flip-flop d. None of these
3. Which of the following flip-flop was developed to remove undefined state of SR flip-flop?
 - a. SR Flip-flop b. D Flip-flop c. T Flip-flop d. None of these
4. Which of the following flip-flop toggles the input?
 - a. SR Flip-flop b. D Flip-flop c. T Flip-flop d. None of these
5. 4-bit shift register is made using _____ D flipflops?
 - a. 3 b. 5 c. 4 d. None of these

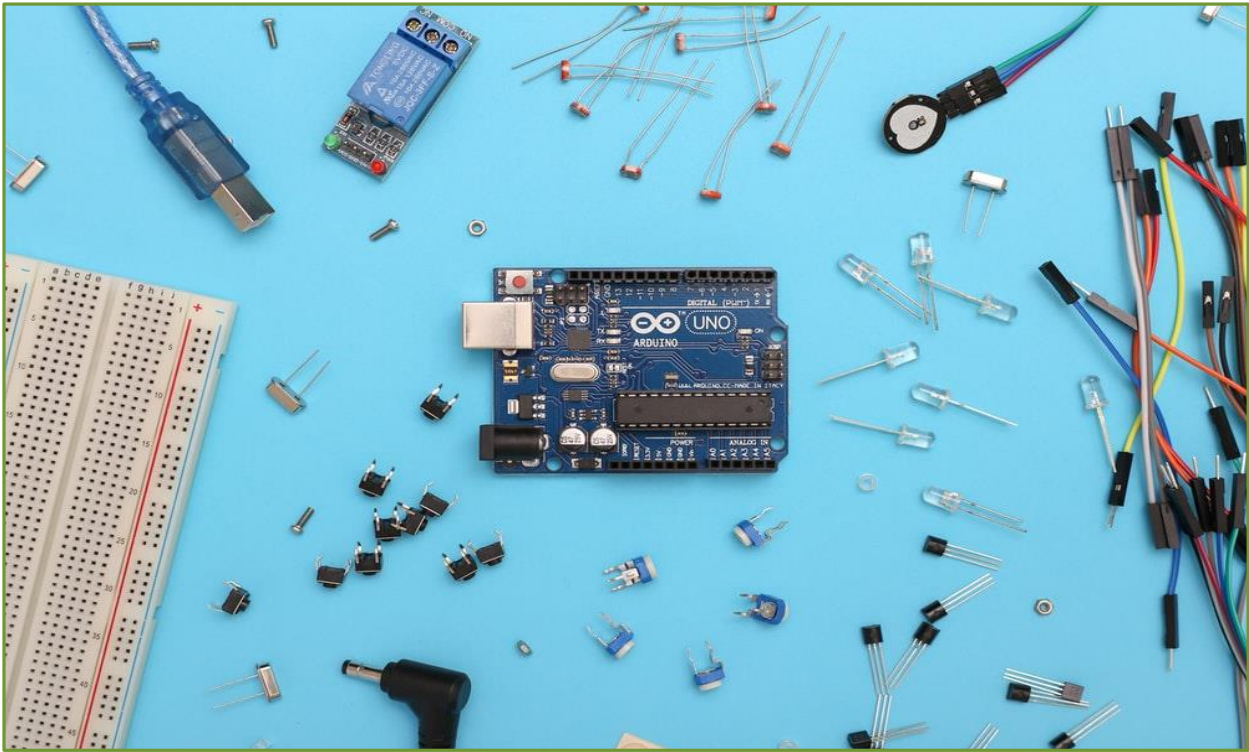
Theoretical Questions

1. Define Sequential Circuits.
2. What is the difference between sequential circuits and combinational circuits?
3. Differentiate between synchronous and asynchronous sequential circuits.
4. Define flipflops.
5. Describe the working of SR flipflop.
6. Describe the working of JK flipflop.
7. Describe the working of D flipflop.
8. Describe the working of T flipflop.
9. Describe the working of 4-bit shift register.
10. What is the difference between SR and JK flipflop.

Practical Tasks

1. Verify the operation of flipflops on actual hardware.
2. Construct a 4-bit shift register on breadboard using D flipflops. Also verify its truth table.

Chapter 3: IoT Networking and Interfacing of Sensors & Actuators



After Studying this chapter, you will be able to:

- understand interfacing process of sensors.
- differentiate between active and passive sensors.
- differentiate between analog and digital sensors.
- understand the working of temperature and humidity sensor.
- identify the communication interface of sensor.
- interface temperature and humidity sensor with Arduino.
- program Arduino to read the value of temperature and humidity sensors.
- understand actuators and relay.
- understand the working principle of a relay.
- identify control pins.
- identify normally open (NO) and normally connected (NC) pins.
- construct a relay driver circuit using transistor.
- develop an Arduino program to drive relay.

3.1 Sensors

Sensors are used to measure physical changes e.g., temperature, light and humidity etc., and convert it into a readable signal. A sensor usually contains a transducer which converts a non-electrical signal into an electrical signal. For instance, most of the temperature sensors are thermocouple-based, in which the electrical potential difference appears due to temperature difference across its terminals.

Interfacing a Sensor

Interfacing a sensor is a process of giving input from the sensor to a microcontroller in a way that the microcontroller can understand and act accordingly. Most of the sensors give output in analog form but the microcontroller or microprocessor needs input in digital form.

Classification of Sensors

Microcontrollers accept two types of inputs depending upon the type of sensor i.e., analog or digital.

Analog sensors sense external parameters (wind speed, solar radiation and light intensity etc.) and produces analog voltage as an output (refer to Figure 3.1). The output voltage may be in the range of 0 to 5V. *Logic High* is treated as “1” (3.5 to 5V) and *Logic Low* is indicated by “0” (0 to 3.5 V).

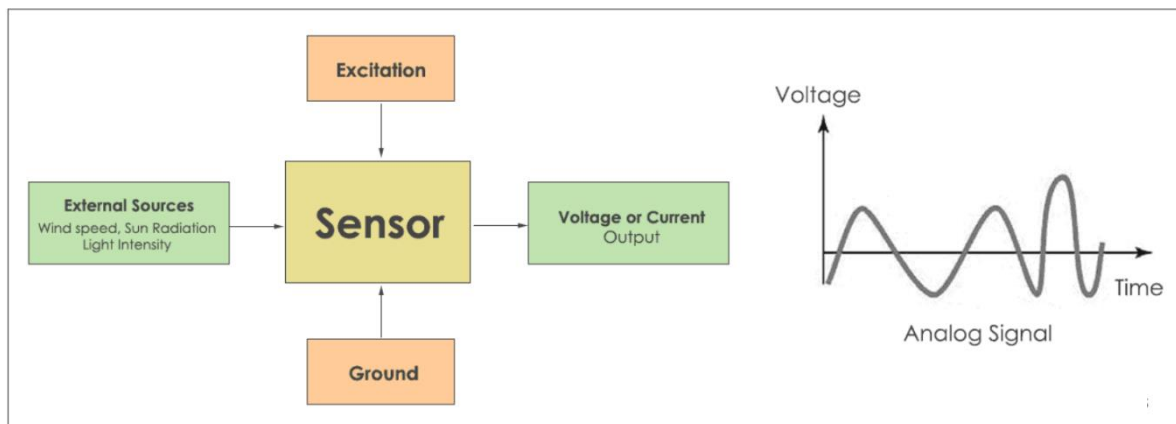


Fig. 3.1 Working of Analog Sensors

Digital Sensors produce discrete values (0 and 1's) (refer to Figure 3.2). Discrete values are often called digital (binary) signals in digital communication.

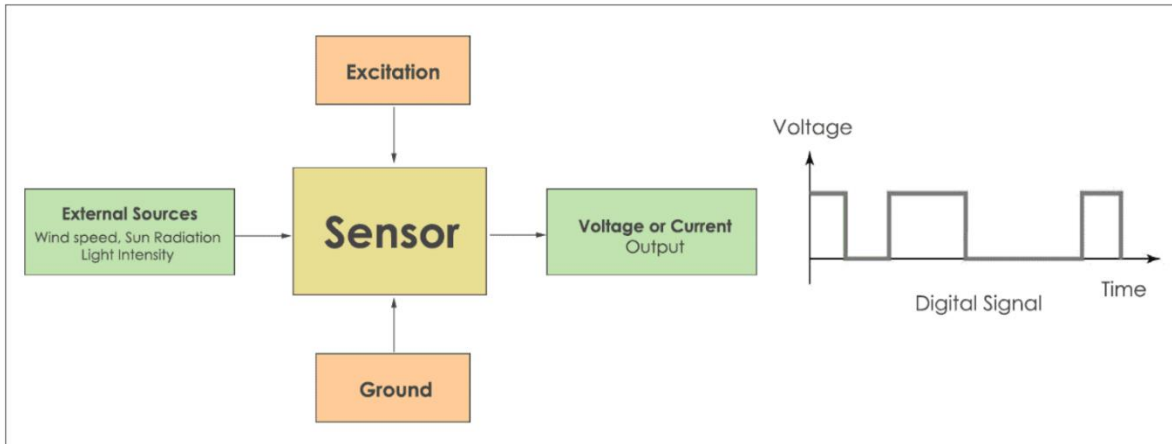


Fig. 3.2 Working of Digital Sensors

Sensor Selection (Analog vs Digital)

Analog circuits are made up of analog components like resistors, capacitors, operational amplifiers, diodes and transistors etc. Whereas, digital circuits consist of logic gates and microcontroller chips. Analog signals are much affected by external noise and create errors in the output signal. Whereas, digital signals are susceptible to noisy environments. Hence, digital sensors are preferred over analog ones.

3.2 DHT11 Sensor

DHT11 is a basic, ultra-low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and produces a digital signal on the data pin. The only downside of this sensor is you can only get new data from it once every 2 seconds.

Technical Details

- Low cost
- 3 to 5V power
- 2.5mA max current use during conversion
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50°C temperature readings $\pm 2^\circ\text{C}$ accuracy
- No more than 1 Hz sampling rate (once every second)
- Body size 15.5mm x 12mm x 5.5mm

What is Relative Humidity?

DHT11 measures *relative humidity*. Relative humidity (RH) is the amount of water vapors in air versus the saturation point of water vapors in the air. At the saturation point, water vapors start to condense and accumulate on surfaces forming dew. The relative humidity is expressed as a percentage.

Transducing Principle of DHT11

DHT11 detects water vapors by measuring the electrical resistance between two electrodes. The humidity sensing component is a moisture holding substrate with electrodes applied to the surface. When water vapors are absorbed by the substrate, ions are released by the substrate which increases the conductivity between the electrodes. The change in resistance between the two electrodes is proportional to the relative humidity. Higher relative humidity decreases the resistance between the electrodes and vice versa.

Teacher Notes: Give examples of other sensors as well.

An IC mounted on the back of the unit converts the resistance measurement to relative humidity. It also stores the calibration coefficients and controls the data signal transmission between the DHT11 and the Arduino.

DHT11 uses just one signal wire to transmit data to the Arduino. Power comes from separate 5V and ground wires. A 10K Ohm pull-up resistor is needed between the signal line and 5V line to make sure the signal level stays high.

There are two different versions of the DHT11. One type has four pins, and the other type has three pins and is mounted to a small PCB (Printed Circuit Board). The following are the pinouts for both versions:

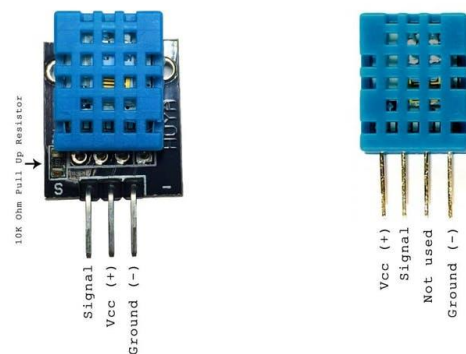


Fig. 3.3 Pin Configuration of DHT11

How to Set Up the DHT11 on an Arduino

Follow the connection diagram given below:

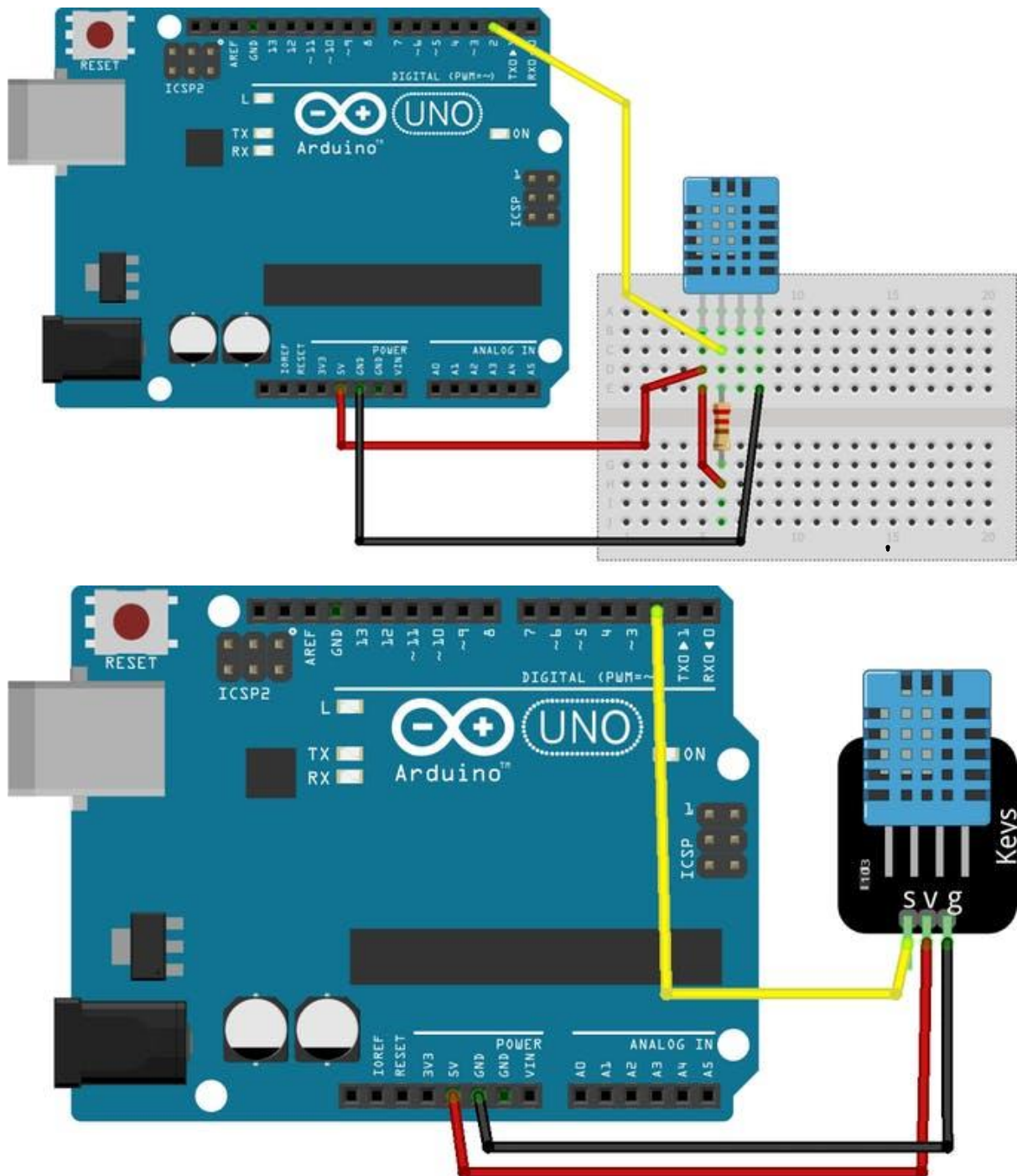


Fig. 3.4 Interfacing DHT11 with Arduino UNO

- VCC - red wire: Connect to 3.3 - 5V power. Sometime 3.3V power isn't enough in which case try 5V power.
- Data out - white or yellow wire
- Not connected
- Ground - black wire

Pin 3 is not used. You can place a 10 K ohm resistor between VCC and the data pin, to act as a medium strength pull-up on the data line. The Arduino has built-in pull-ups but they are very weak.

Preliminary Steps of Programming Arduino

It is necessary to install the DHT Sensor library in Arduino IDE to work with DHT11. It can be done through the Arduino Library Manager:

Go to: Sketch→Include Library→Manage Libraries

Enter “dht” in the search field and look through the list for the “DHT sensor library by Adafruit.” Click the “Install” button, or “Update” from an earlier version.

You will also need to install the Adafruit_Sensor library, which is also available in the Arduino Library Manager.

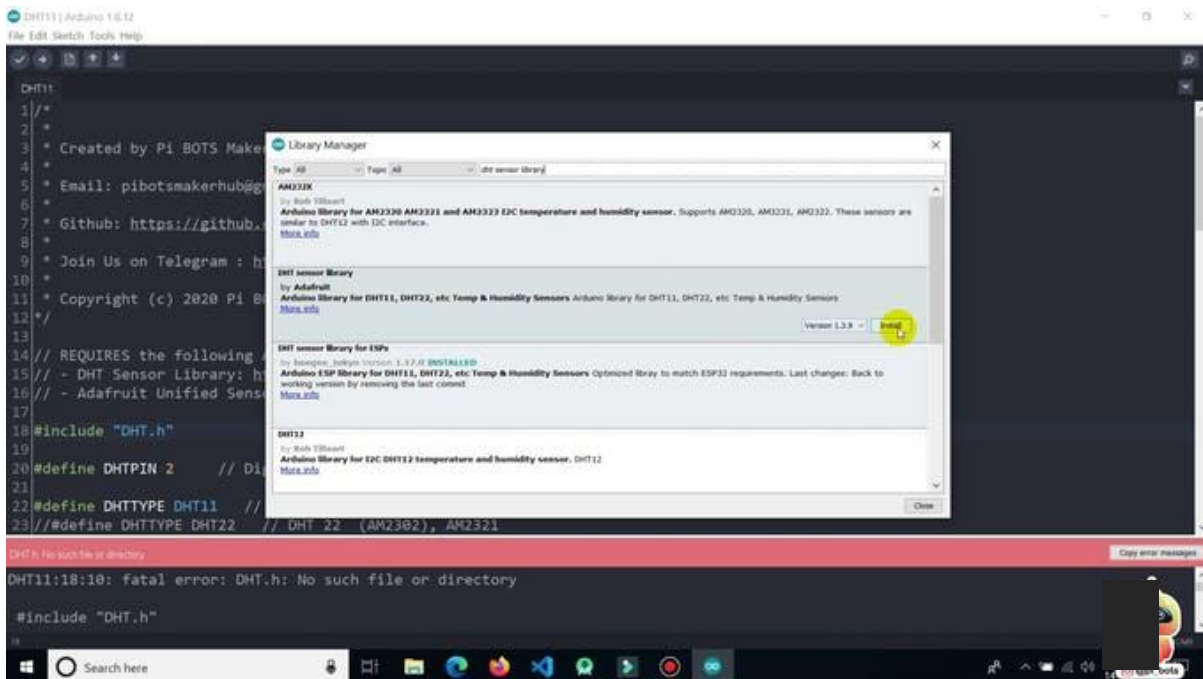


Fig. 3.5 Adding DHT11 Library to Arduino IDE

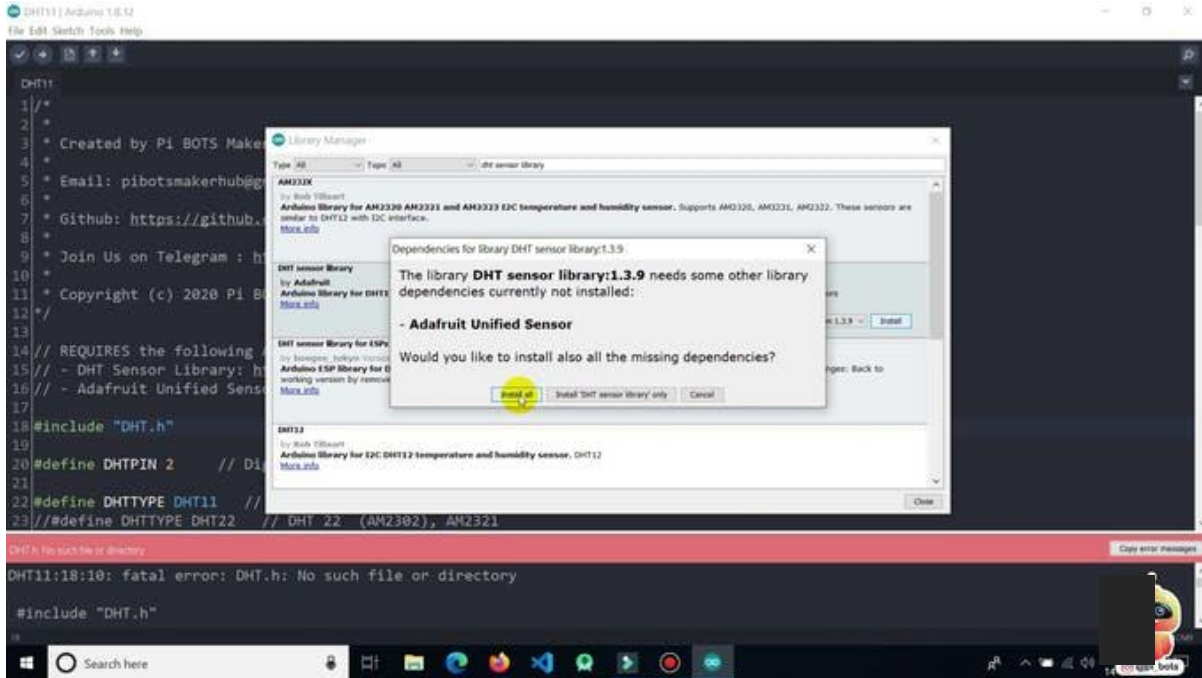


Fig. 3.6 Installing DHT11 Library

Now load Examples→DHT→DHTtester sketch.

Code Explanation

We have to include the "DHT.h" Library.

```
#include "DHT.h"
```

Then define the digital pin to which DHT11 is connected to:

```
#define DHTPIN 2 // Digital pin connected to the DHT sensor
```

Define the type of DHT Sensor:

```
#define DHTTYPE DHT11 // DHT 11
```

If you have a DHT22, write this:

```
#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
```

Define the DHT parameter to initialize DHT sensor:

```
DHT dht(DHTPIN, DHTTYPE);
```

Inside the void setup function, initialize serial communication and DHT Sensor.

```
void setup() {
  Serial.begin(9600);
  Serial.println(F("DHTxx test!"));
}
```

```
dht.begin();
}
```

Inside the void loop function, take readings.

Reading temperature or humidity takes about 250 milliseconds. Sensor readings may also be up to 2 seconds old.

```
float h = dht.readHumidity(); // read humidity
```

Read temperature as Celsius:

```
float t = dht.readTemperature(); // read temperature
```

Read temperature as Fahrenheit (isFahrenheit = true)

```
float f = dht.readTemperature(true);
```

Check if any read failed and exit early.

```
if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
}
```

Compute heat index in Fahrenheit:

```
float hif = dht.computeHeatIndex(f, h);
```

Compute heat index in Celsius (isFahreheit = false)

```
float hic = dht.computeHeatIndex(t, h, false);
```

Print the measured readings on the Serial Monitor.

```
Serial.print(F(" Humidity: "));
Serial.print(h);
Serial.print(F("% Temperature: "));
Serial.print(t);
Serial.print(F("°C "));
Serial.print(f);
Serial.print(F("°F Heat index: "));
Serial.print(hic);
Serial.print(F("°C "));
Serial.print(hif);
Serial.println(F("°F"));
```

You can use this code to measure the temperature and humidity values using the DHT11 or DHT22 sensors in any projects.

3.3 Actuators

An actuator is a device that is used to convert a control signal into mechanical motion. Electric door locks in automobiles and ailerons on aircraft are examples of actuators. Industrial plants use actuators to operate valves, dampers, fluid couplings and other devices.

Relay

A *relay* is a mechanical switch, which can be toggled on or off by energizing a coil. Relays are of different types depending on the operating principle and structural features:

- Electromagnetic Relays
- Solid State Relays
- Thermal Relays
- Power Varied Relays
- Reed Relays
- Hybrid Relays

A relay has 5 pins (refer to Figure 3.7). Two pins are used to energize the coil. Middle one is COM (common) and the rest of the two are called NO (Normally Open) and NC (Normally Close). When current flows through the coil of the relay, a magnetic field is created that causes a ferrous armature to move, either making or breaking an electrical connection. When the electromagnet is energized, the NO is on and NC is off. When the coil is de-energized the electromagnetic force disappears and the armature moves back to the original position turning on the NC contact. The closing and releasing of the contacts result in turning on and off of the circuits.

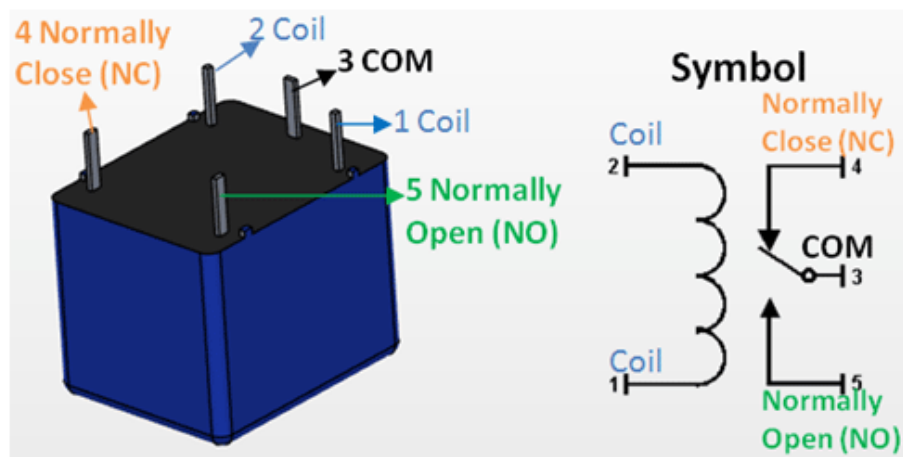


Fig. 3.7 Pin Configuration of a Relay

A microcontroller's GPIO (general purpose input/output) pins cannot handle higher power devices. A LED is easy enough, but large power items such as light bulbs, motors, pumps or fans require more complex circuitry. You can use a 5V relay to switch the 120-240V current and use the Arduino to control the relay.

A relay basically allows a relatively low voltage to easily control higher power circuits. A relay accomplishes this by using the 5V from an Arduino pin to energize the electromagnet which in turn closes an internal, physical switch to turn on or off a higher power circuit. The switching contacts of a relay are completely isolated from the coil, and the Arduino. This process is called "Electrical Isolation".

The coil of the relay needs a large current (around 150mA) to drive the relay, which an Arduino cannot provide. Therefore, we need a device to amplify the current. In the figure below (Figure 3.8), the NPN transistor 2N2222 drives the relay when the NPN junction gets saturated. Follow the following steps to drive a relay using Arduino.

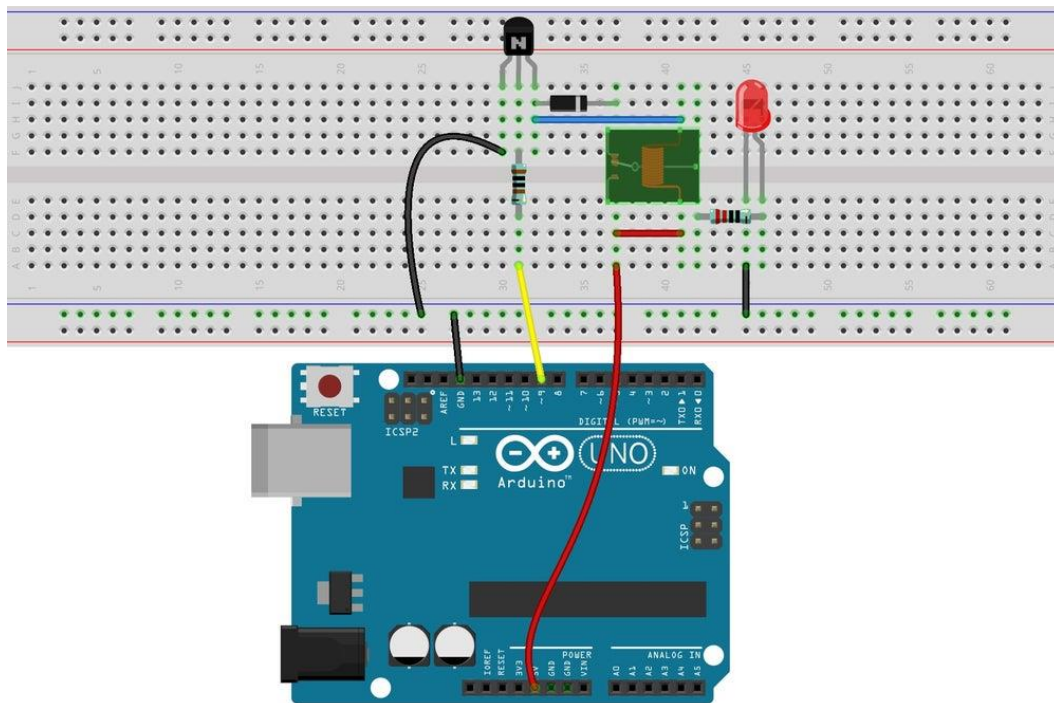


Fig. 3.8 Interfacing a Relay with Arduino UNO

1. Connect the V_{IN} and GND pins of the Arduino to the +ve and -ve rails of the breadboard.
2. Connect one of the coils pin to the +ve 5V rail of breadboard.

3. We need to connect a diode across the electromagnetic coil. The diode across the electromagnet conducts in the reverse direction when the transistor is turned off to protect against a voltage spike or the backward flow of current.
4. Connect the Collector of the NPN transistor to the 2nd pin of the coil.
5. The Emitter is connected to the -ve rail of the breadboard.
6. Finally, connect the Base of the transistor to the D2 pin of the Arduino using a 1K resistor.
7. The Circuit is complete, now we can upload the code to the Arduino to turn on or off the relay. When +5V flow through the 1K resistor to the Base of transistor, a current of about .0005 amps (500 microamps) flows and turns on the transistor. A current of about .07 amps starts flowing through the junction turning on the electromagnet. The electromagnet then pulls the switching contact and moves it to connect the COM terminal to the NO terminal.
8. Once the NO terminal is connected a lamp or any other load can be turned on.

Code

```
const int relayPin = 8; //the base of the transistor attach to
/*****/
void setup()
{
  pinMode(relayPin, OUTPUT); //initialize the
  relayPin as an output
}
/*****/
void loop()
{
  digitalWrite(relayPin, HIGH); //drive relay
closure conduction
  delay(1000); //wait for a second
  digitalWrite(relayPin, LOW); //drive the
  relay is closed off
  delay(1000); //wait for a second
```


}

/*****/

Key Points

- A sensor is used to measure the physical changes that occur in the surroundings like temperature and light and convert it into a readable signal.
- Interfacing a sensor is giving input from the sensor to the microcontroller in a way which they can understand and act accordingly.
- Analog sensor senses the external parameters (wind speed, solar radiation, light intensity etc.) and gives analog voltage as an output. The output voltage may be in the range of 0 to 5V.
- Digital Sensor produce discrete values (0 and 1's). Discrete values often called digital (binary) signals in digital communication.
- The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor.
- The DHT11 measures *relative humidity*.
- The relative humidity is the amount of water vapor in air vs. the saturation point of water vapor in the air.
- The DHT11 detects water vapor by measuring the electrical resistance between two electrodes.
- The DHT11 uses just one signal wire to transmit data to the Arduino.
- It's necessary to install the DHT Sensor library to interface DHT11 with Arduino.
- An actuator is a device that uses a form of power to convert a control signal into mechanical motion.
- A Relay is a large mechanical switch, which is toggled on or off by energizing a coil.
- This relay has 5 pins. 2 for the coil. Middle one is COM (common) and the rest of the two are called NO (Normally Open) and NC (Normally Close).

Exercise

Choose the most suitable option.

1. DHT11 can measure.
a. Temperature b. Humidity c. Pressure d. Both a and b
2. Relay has how many pins?
a. 6 b. 4 c. 5 d. 7
3. Arduino UNO needs a _____ to drive relay.
a. capacitor b. transistor c. diode d. None of these
4. DHT11 has _____ pins?
a. 3 b. 4 c. 5 d. 7
5. Delay(1000) means delay of _____ seconds?
a. 3 b. 1 c. 4 d. None of these

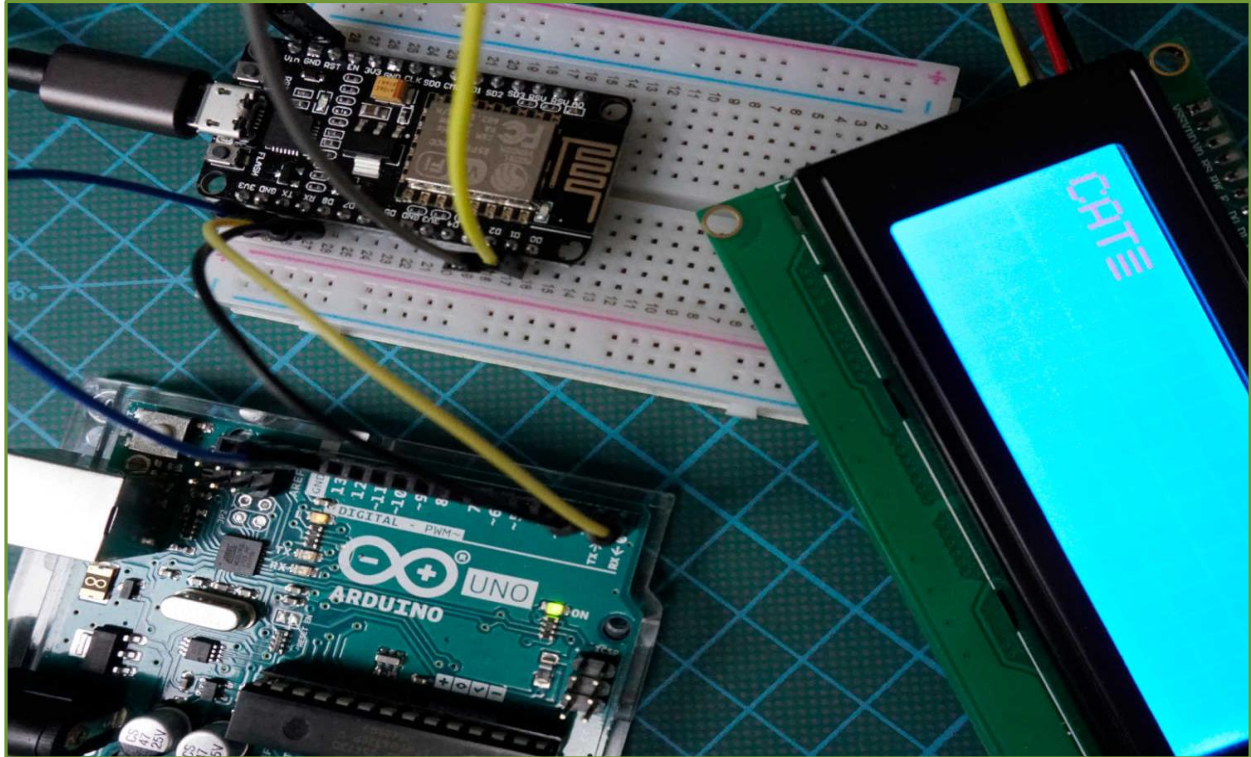
Give short answer to the following questions.

1. Define Sensors.
2. Define transducers and transducing principle.
3. Describe DHT11 sensor.
4. What is the transducing principle of DHT11?
5. What are actuators?
6. Differentiate between sensors and actuators.
7. Define relay.
8. Describe the construction of a relay.
9. What is NO and NC pins of a relay.
10. Can a microcontroller drive a relay?

Practical Tasks

1. Interface DHT11 with Arduino and record the readings of humidity.
2. Drive a relay using Arduino. You can use transistor as a switch to drive relay.

Chapter 4: IoT Networking and Interfacing of Communication Modules



After Studying this chapter, you will be able to:

- understand Wi-Fi protocol.
- understand ESP8266 module.
- connect ESP8266 with Arduino.
- program the Arduino to connect to available Wi-Fi Access Point.
- differentiate between Wi-Fi station and Wi-Fi access point.
- understand the process of configuring ESP8266 as a Wi-Fi station without using Arduino.
- write a program to connect ESP8266 to an available Wi-Fi access point.
- understand the difference between Wi-Fi and Bluetooth protocol.
- understand the difference between ESP 8266 and ESP32.
- understand the process of configuring Bluetooth in ESP32.
- develop a program to connect ESP32 to the Bluetooth of mobile phone.

4.1 Wi-Fi Protocol

Wi-Fi is based on the IEEE 802.11 family of standards and is primarily a local area networking (LAN) technology designed to provide in-building broadband coverage. Wi-Fi systems support a peak physical-layer data rate of 54 Mbps and typically provide indoor coverage over 100 feet. Wi-Fi has become the de facto standard for last mile broadband connectivity in homes, offices, and public hotspot locations.

Wi-Fi offers remarkably higher peak data rates than 3G systems. It operates over a larger 20 MHz bandwidth, but Wi-Fi systems are not designed to support high-speed mobility. One significant advantage of Wi-Fi over WiMAX and 3G is its wide availability of terminal devices.

4.2 ESP8266 Wi-Fi Module

The ESP8266 is a useful, low-cost Wi-Fi module for controlling devices over the Internet. It can work with a micro-controller, or it can be programmed to work on its own. The ESP8266 comes with factory installed firmware allowing you to control it with standard “AT commands”. You can easily create and upload your own code.

- It supports the 802.11 b/g/n protocol
- It can connect to your router and work as a client, or it can be an access point itself or both.
- It is addressable using the assigned IP.
- The “standard” version has 2 digital pins that can be used for input or output. e.g: to drive LED’s or relays. These pins can also be used for PWM (Pulse Width Modulation).
- Analog input is also available on the ESP8266 chip (ADC/TOUT) but it’s not wired up on the ESP-01.
- It can be combined with an Arduino, or it can be programmed to work on its own
- There are various tools and development environments (IDE’s) to program it.

4.3 Interfacing Wi-Fi with Arduino UNO

Wire the ESP8266 Module, the Arduino UNO, and the breadboard as shown in Figure 4.1.

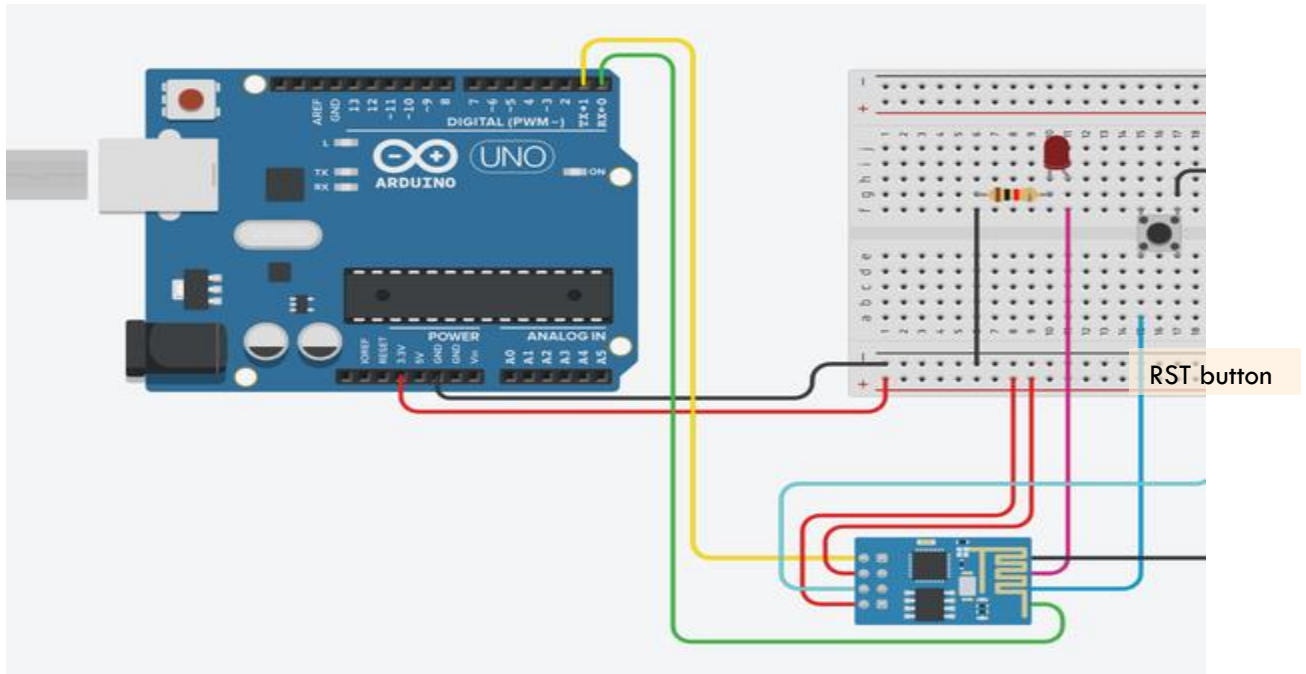


Fig. 4.1 Interfacing an ESP8266 with Arduino

The pins should be connected as follows:

ESP8266 TX --> Arduino UNO TX

ESP8266 GND --> Breadboard GND

ESP8266 CH-PD --> Breadboard 3.3V

ESP8266 GPIO2 --> Breadboard LED

ESP8266 RST --> Breadboard RST button --> Arduino UNO GND

ESP8266 GPIO0 --> Breadboard 3.3V/Breadboard FLASH button --> Arduino UNO GND

ESP8266 VCC --> Breadboard 3.3V

ESP8266 RX --> Arduino UNO RX

To flash your ESP8266 module, first check that the GPIO0 pin on the ESP8266 is connected to the FLASH button on the breadboard. If there is a program currently running on your Arduino, upload an empty project file to the Arduino to avoid interference while flashing Wi-Fi module.

Use the following code in the Arduino project and replace the SSID and PASSWORD at the top with your WiFi-network info.

```
#include <ESP8266WiFi.h>
```

```
const char* ssid = "SSID";//type your ssid
```

```
const char* password = "PASSWORD";//type your password
```

```
int ledPin = 2; // GPIO2 of ESP8266

WiFiServer server(80); // Service Port

void setup() {
  Serial.begin(115200);
  delay(10);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  // Start the server
  server.begin();
  Serial.println("Server started");
  // Print the IP address
  Serial.print("Use this URL to connect: ");
  Serial.print("http://");
  Serial.print(WiFi.localIP());
```

```
Serial.println("/");
}

void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }
  // Wait until the client sends some data
  Serial.println("new client");
  while(!client.available()){
    delay(1);
  }
  // Read the first line of the request
  String request = client.readStringUntil('\r');
  Serial.println(request);
  client.flush();
  // Match the request
  int value = LOW;
  if (request.indexOf("/LED=ON") != -1) {
    digitalWrite(ledPin, HIGH);
    value = HIGH;
  }
  if (request.indexOf("/LED=OFF") != -1){
    digitalWrite(ledPin, LOW);
    value = LOW;
  }
}
```



```

}
//Set ledPin according to the request
//digitalWrite(ledPin, value);
// Return the response
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println(""); // do not forget this one
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.print("Led pin is now: ");
if(value == HIGH) {
  client.print("On");
} else {
  client.print("Off");
}
client.println("<br><br>");
client.println("Click <a href=\"/LED=ON\">here</a> turn the LED on pin 2 ON<br>");
client.println("Click <a href=\"/LED=OFF\">here turn the LED on pin 2 OFF<br>");
client.println("</html>");
delay(1);
Serial.println("Client disconnected");
Serial.println("");
}

```

Open your serial monitor, set the baud rate to 115200. Press the RST (reset) button while holding down the FLASH button on the breadboard. You should see a blue light lightening up on the Wi-Fi module and the TX led on the Arduino. You must upload code while the TX led is lit. Now,

compile and upload the code. Record the IP printed in the serial monitor. If the flash fails, try the above steps again. If the serial monitor is printing "." repeatedly, you have entered your Wi-Fi information incorrectly.

Note: the RST button connects the RST pin on the Wi-Fi module to ground. The FLASH button connects the GPIO0 pin on the Wi-Fi module to ground. Now, for normal operation, connect GPIO0 to the 3.3V rail on the breadboard. This can be done by moving the wire connecting GPIO0 to the button over the positive rail of the breadboard. To connect to your ESP8266 module, ensure you are connected to the same network as it is.

4.4 Wi-Fi Station vs Access point

Wireless adapters, or Network Interface Controllers (NICs), are network cards with the 802.11 standard that let a machine connect to a wireless network. A station is any device that has such a card. Access points, also known as AP or hotspots, can let nearby Wi-Fi equipped stations access a wired network to which the access point is directly connected. The 802.11 standard defines two operating modes: infrastructure mode and ad hoc mode.

4.5 ESP8266 as a Wi-Fi Station

ESP8266 can communicate over Wi-Fi in two different modes. It can connect to an existing wireless hot spot, or access point, like the way you connect your phone or computer to the Internet. This is called “station” mode. In station mode the ESP8266 board can reach out to the internet and may also communicate with other devices connected to the same network, including other ESP8266 modules. The following is a simple example. Replace the values for the **ssid** and **password** variables with the name and password for your wireless network.

```
/* Connect the ESP8266 unit to an existing WiFi access point
#include <ESP8266WiFi.h>

// Replace these with your WiFi network settings
const char* ssid = "ESP8266"; //replace this with your WiFi network
name
const char* password = "ESP8266Test"; //replace this with your WiFi
network password

void setup()
{
```

```
delay(1000);
Serial.begin(115200);
WiFi.begin(ssid, password);
Serial.println();
Serial.print("Connecting");
while (WiFi.status() != WL_CONNECTED)
{
    delay(500);
    Serial.print(".");
}
Serial.println("success!");
Serial.print("IP Address is: ");
Serial.println(WiFi.localIP());
}
void loop() {
}
```

Once you load the code on your ESP8266 module, open the serial monitor and restart the module (pressing the reset button on the NodeMCU board). All the code executes in the set-up function, so it will only run once.

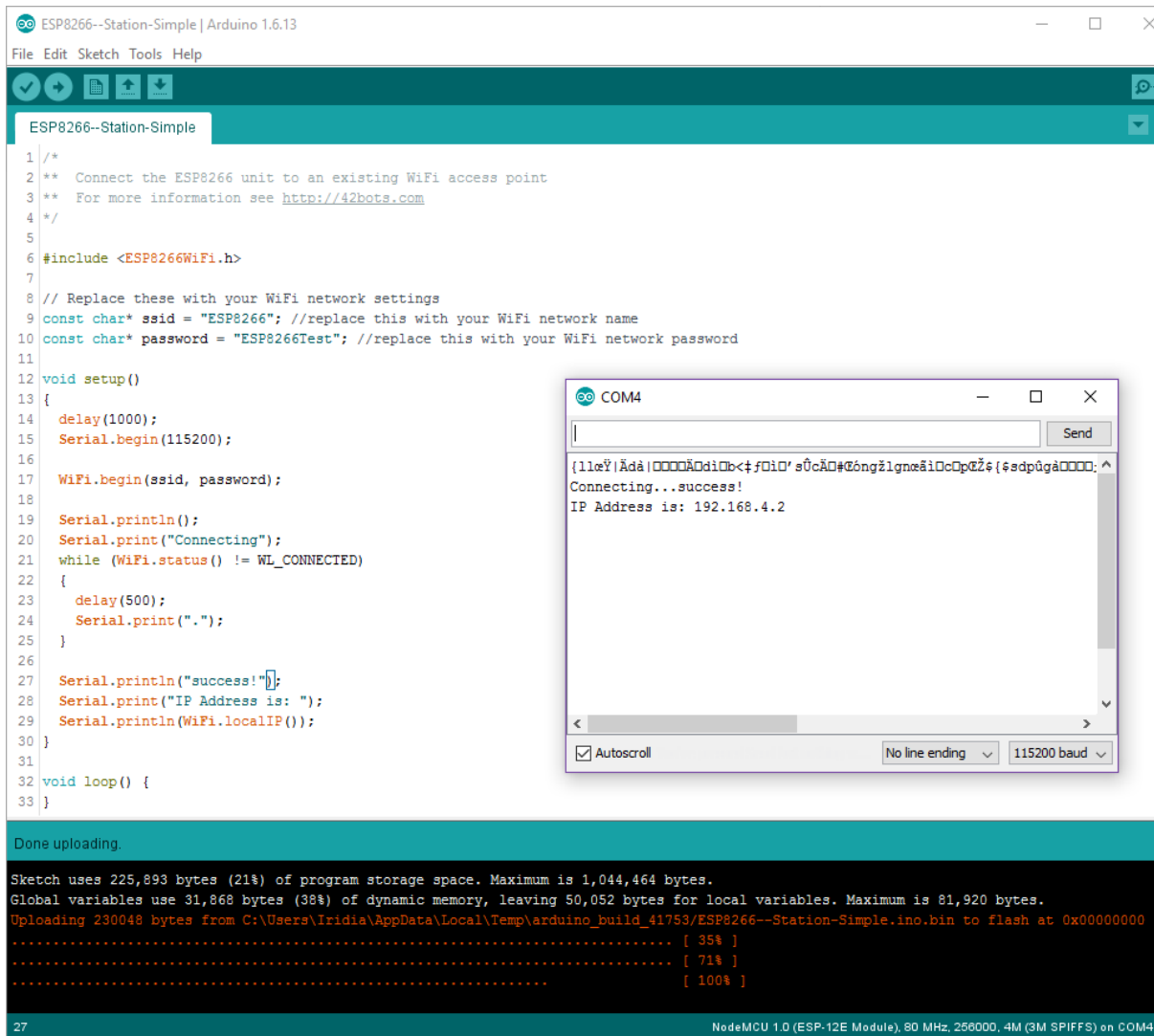


Fig. 4.2 Arduino Project to configure ESP8266 as Wi-Fi Station

4.6 Bluetooth vs Wi-Fi

Bluetooth and Wi-Fi are used for providing wireless communication through radio signals. Bluetooth is used for short-range devices for sharing information whereas Wi-Fi is used for providing high-speed web access or internet. Wi-Fi provides high information measure because the speed of web is a paramount issue.

The following are the differences between Bluetooth and Wi-Fi:

S.NO	Bluetooth	Wi-Fi
1.	Bluetooth requires Bluetooth adapter on all devices for connectivity.	Wi-Fi requires wireless adapter on all devices and wireless router for connectivity.
2.	Bluetooth consumes less power.	It consumes high power.

3.	The security of Bluetooth is low as compared to Wi-Fi.	Wi-Fi provides better security than Bluetooth.
4.	Bluetooth supports relatively limited number users.	Wi-Fi supports large number of users.
5.	The radio signal range of Bluetooth is ten meters.	Wi-Fi range is hundred meters.
6.	Bluetooth require low bandwidth.	Wi-Fi requires high bandwidth.

4.7 ESP32 vs ESP8266

The ESP32 and ESP8266 are cheap Wi-Fi modules perfectly suited for DIY (Do it yourself) projects on the Internet of Things (IoT) and Home Automation. Both chips have a 32-bit processor. The ESP32 is a dual-core 160MHz to 240MHz CPU, whereas the ESP8266 is a single-core processor that runs at 80MHz. These modules come with GPIOs that support various protocols like SPI, I2C, UART, ADC, DAC, and PWM. Both have built-in Wi-Fi. This means that you can easily control and monitor devices remotely via Wi-Fi or Bluetooth.

The ESP32 is the successor of ESP8266. It adds an extra CPU core, faster Wi-Fi, more GPIOs, and supports Bluetooth 4.2 and Bluetooth low energy. Additionally, the ESP32 comes with touch-sensitive pins that can be used to wake up the ESP32 from deep sleep, a built-in hall effect sensor, and a built-in temperature sensor.

4.8 ESP32 as a Bluetooth Device

Bluetooth operates in the unlicensed 2.4 GHz ISM (Industrial, Scientific, and Medical) frequency band. Bluetooth is a short-range wireless communication technology with range up to 100 m. Since both Wi-Fi and Bluetooth operate at the same 2.4 GHz ISM frequency, the Wi-Fi Radio and the Bluetooth Radio share the same antenna in ESP32.

ESP32 supports both the Classic Bluetooth and Bluetooth Low Energy (BLE) which can be configured with BLUEDROID Bluetooth Stack. ESP32 Bluetooth supports three types of Host Controller Interface (HCI): UART, SPI and VHCI (Virtual HCI) interfaces.

Getting Started with ESP32 Classic Bluetooth

The Classic Bluetooth is the original point-to-point network topology designed for one-to-one wireless communication between a master and a slave. Even though multiple slave devices can be connected to a single master, only one slave can be actively communicating with the master. Our Bluetooth keyboards and mouse work with Classic Bluetooth technology.

BLE or Bluetooth Low Energy is designed for low power operation and developed with IoT applications as the main target. Bluetooth Specification 4.0 added BLE functionality and is mainly used in battery operated devices like watches, audio devices, health trackers, fitness monitors and data beacons.

The BLUEDROID Bluetooth Stack communicates with Bluetooth Controller over VHCI and at the same time provides APIs for user application. Bluetooth Profiles determine the functions of each layer of the Bluetooth while the Bluetooth Protocols define message formats and procedures for data transport, link control etc. The following is a list of Classic Bluetooth Profiles and Protocols supported by BLUEDROID Bluetooth Stack of ESP32.

Classic Bluetooth Profiles

- GAP
- A2DP (SNK)
- AVRCP (CT)

Classic Bluetooth Protocols

- L2CAP
- SDP
- AVDTP
- AVCTP

ESP32 Classic Bluetooth Serial Communication

Arduino UNO and HC-05 communicate over Serial Communication. ESP32, which already has a Bluetooth Controller, also has a similar communication between the main Xtensa Processor and the Bluetooth Controller. This means that after receiving data from a Bluetooth device wirelessly, the Bluetooth controller in ESP32 transfers this data to ESP32's processor over serial communication. Similarly, to send data over Bluetooth, the processor of ESP32 transmits data to the Bluetooth controller using the serial interface. We will use this information along with a dedicated *BluetoothSerial* library to transmit and receive data. The *BluetoothSerial* library works similar to the Serial library. Some of the frequently used functions offered by *BluetoothSerial* library are:

- `begin()`
- `available()`
- `write()`
- `read()`

Let us write a simple code which transfers data between ESP32 and a Mobile Phone:

Code

Create an object of class *BluetoothSerial* and begin the communication using ‘begin()’ function. You can pass the name of the ESP32 Bluetooth Device as an argument to the ‘begin()’ function. If you leave it blank, then the default name i.e., ESP32 is used. Also initialize the normal serial communication with baud rate of 115200. Then, in the loop function, read data from *BluetoothSerial* and print it on the Serial Monitor and read data from the Serial Monitor and write it to *BluetoothSerial*. When we write data to *BluetoothSerial*, the Bluetooth Terminal App on the phone receives the data and prints it on the app. When you type data in the app and send it over Bluetooth, the *BluetoothSerial* will read this data and is printed on Serial Monitor.

```
#include "BluetoothSerial.h"

/* Check if Bluetooth configurations are enabled in the SDK */
/* If not, then you have to recompile the SDK */
#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif

BluetoothSerial SerialBT;

void setup() {
  Serial.begin(115200);
  /* If no name is given, default 'ESP32' is applied */
  /* If you want to give your own name to ESP32 Bluetooth device, then
  */
  /* specify the name as an argument
  SerialBT.begin("myESP32Bluetooth"); */
  SerialBT.begin();
  Serial.println("Bluetooth Started! Ready to pair...");
}

void loop() {
```

```

if (Serial.available())
{
  SerialBT.write(Serial.read());
}
if (SerialBT.available())
{
  Serial.write(SerialBT.read());
}
delay(20);
}

```

Uploading the Code and Testing

After uploading the code to ESP32, if you open the serial monitor of Arduino IDE, you can see the ESP32 printing some information about Bluetooth. It also displays the 'ready' message.

```

ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
config: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:10944
load:0x40080400,len:6388
entry 0x400806b4
Bluetooth Started! Ready to pair...

```

Fig. 4.3 Output of ESP32 as a Bluetooth Device

Turn on Bluetooth in your smart phone and scan for Bluetooth devices. You should see a list of ‘Paired devices’ and ‘Available devices’ and select ‘ESP32’ from the available devices.

Your mobile will ask if you want to pair with ‘ESP32’ and you select yes (or ok). There is no password. Now, open the ‘Serial Bluetooth Terminal’ app on your phone and click on the three horizontal bars on the top left corner of the screen.

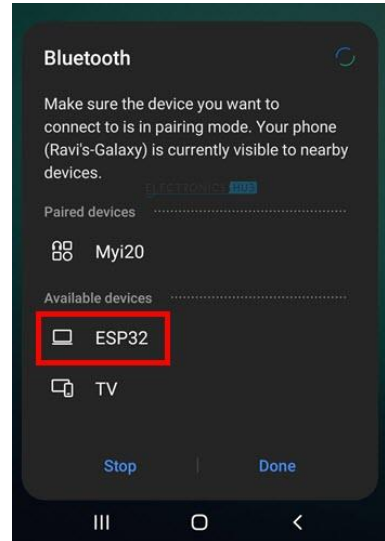
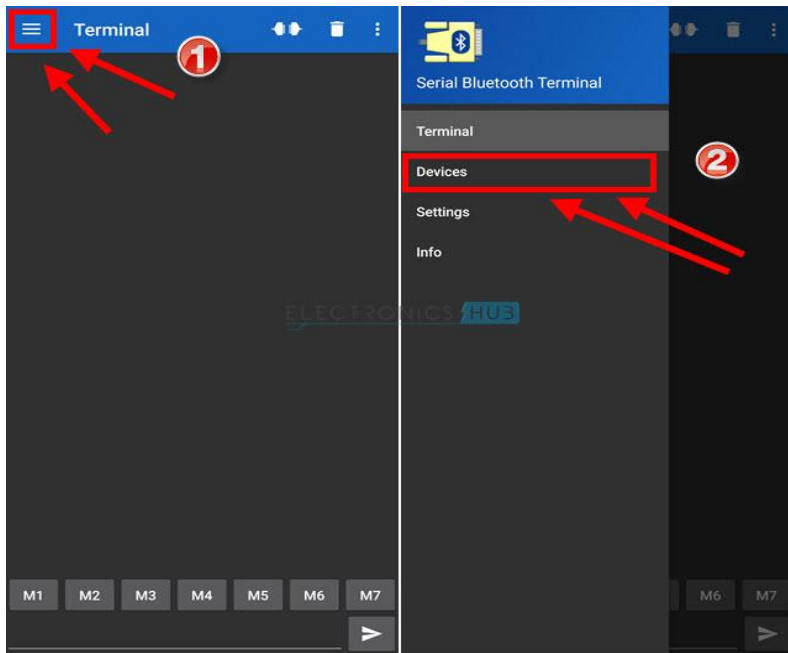


Fig. 4.4 connecting the mobile phone Bluetooth to ESP32



Select ‘Devices’ tab and select ESP32 from the list.

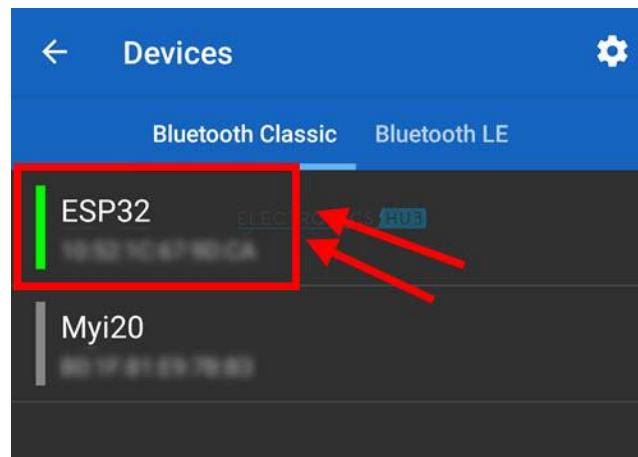


Fig. 4.6 Selecting ESP32

Now, click on the 'link' icon on the top to connect to ESP32 Bluetooth Device. The app will display the status as 'Connecting to ESP32 ...' while establishing connection. If the connection is successful, it will display 'Connected'.

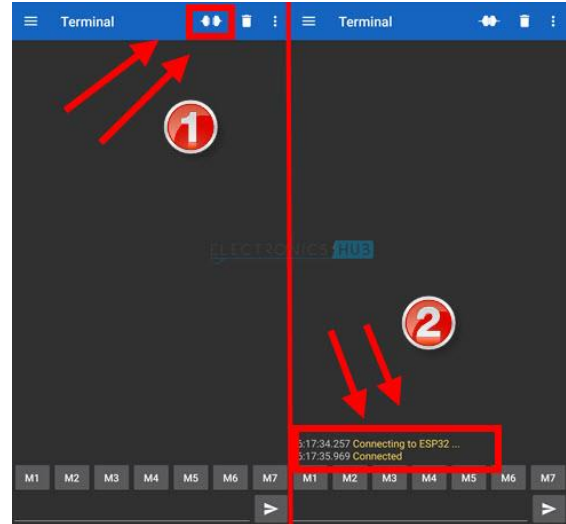
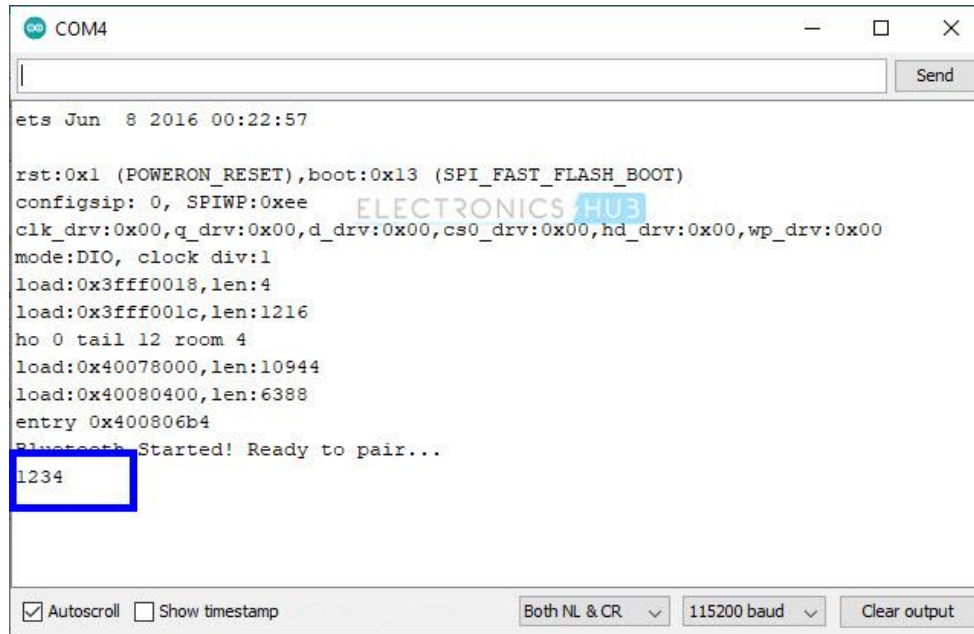


Fig. 4.7 Connecting ESP32

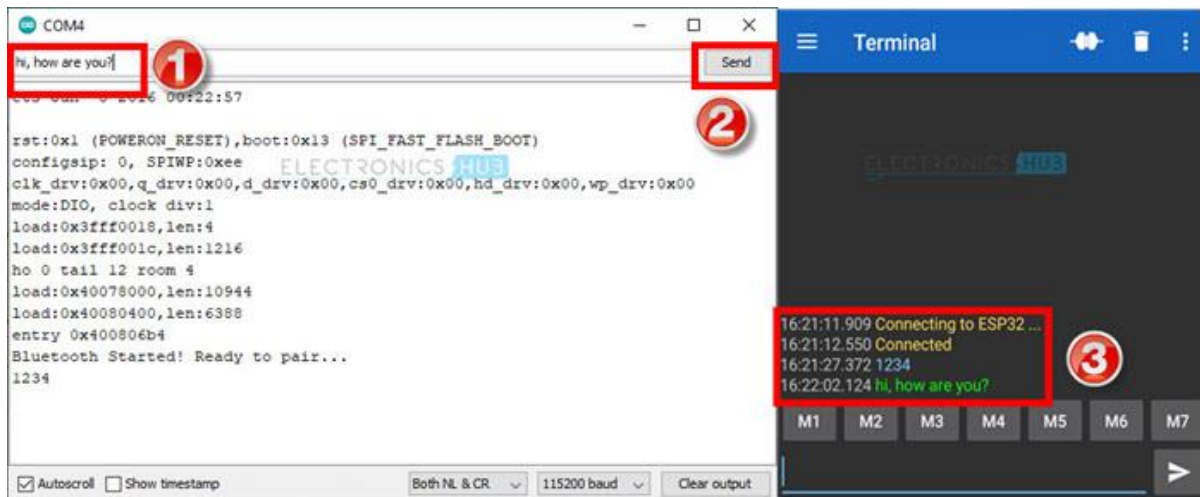


Below is a space for entering data to transfer over Bluetooth. Type something and click on the send button. The sent data is echoed back to the app. This data is sent to ESP32 over Bluetooth and is received by *Bluetooth Serial read()* function. Since we are passing this information to Serial port, you can see the data printed on the serial monitor.

Teacher Notes: You can also use different Bluetooth applications available on playstore to receive messages.



Similarly, you can send data from ESP32 to Mobile Phone. Just type some data in the serial monitor and click on send. This data is sent over Bluetooth to Mobile Phone through the BluetoothSerial write() function. The serial Bluetooth terminal app will read this data and prints it on the app.



Key Points

- Wi-Fi stands for Wireless Fidelity. It is based on the IEEE 802.11 family of standards and is primarily a local area networking (LAN) technology designed to provide in-building broadband coverage.
- Wi-Fi offers remarkably higher peak data rates than 3G systems. It operates over a larger 20 MHz bandwidth, but Wi-Fi systems are not designed to support high-speed mobility.
- The ESP8266 is a useful, cheap Wi-Fi module for controlling devices over the Internet.
- To connect to your ESP8266 module, ensure you are connected to the same network as it is.
- Wireless adapters, or network interface controllers (NICs for short), are network cards with the 802.11 standard that let a machine connect to a wireless network.
- A station is any device that has such a card. Access points, also known as AP or hotspots, can let nearby WiFi-equipped stations access a wired network to which the access point is directly connected.
- The 802.11 standard defines two operating modes: infrastructure mode and ad hoc mode.
- The most distinction between Bluetooth and Wi-Fi is that Bluetooth is actually accustomed connect short-range devices for sharing information whereas Wi-Fi is used for providing high-speed web access or internet.
- Bluetooth is a great wireless communication technology that has been popular for quite few years. Operating in the unlicensed 2.4 GHz ISM (Industrial, Scientific, and Medical) frequency band, Bluetooth is a short-range wireless communication technology with range up to 100 m.

Exercise

Choose the most suitable option

- Which Interfaces are available on ESP32?
 - Wi-Fi
 - Bluetooth
 - IR
 - Both a and b
- Which one of the following is not a standard baudrate?
 - 9600
 - 115200
 - 2600
 - None of these
- ESP8266 cannot be configured as:
 - Wi-Fi Station
 - Bluetooth
 - Wi-Fi Access Point
 - Both a and b
- ESP32 can be configured as:
 - Wi-Fi Station
 - Bluetooth
 - Wi-Fi Access Point
 - All of these
- Delay(1000) means delay of ____ seconds?
 - 3
 - 1
 - 4
 - None of these

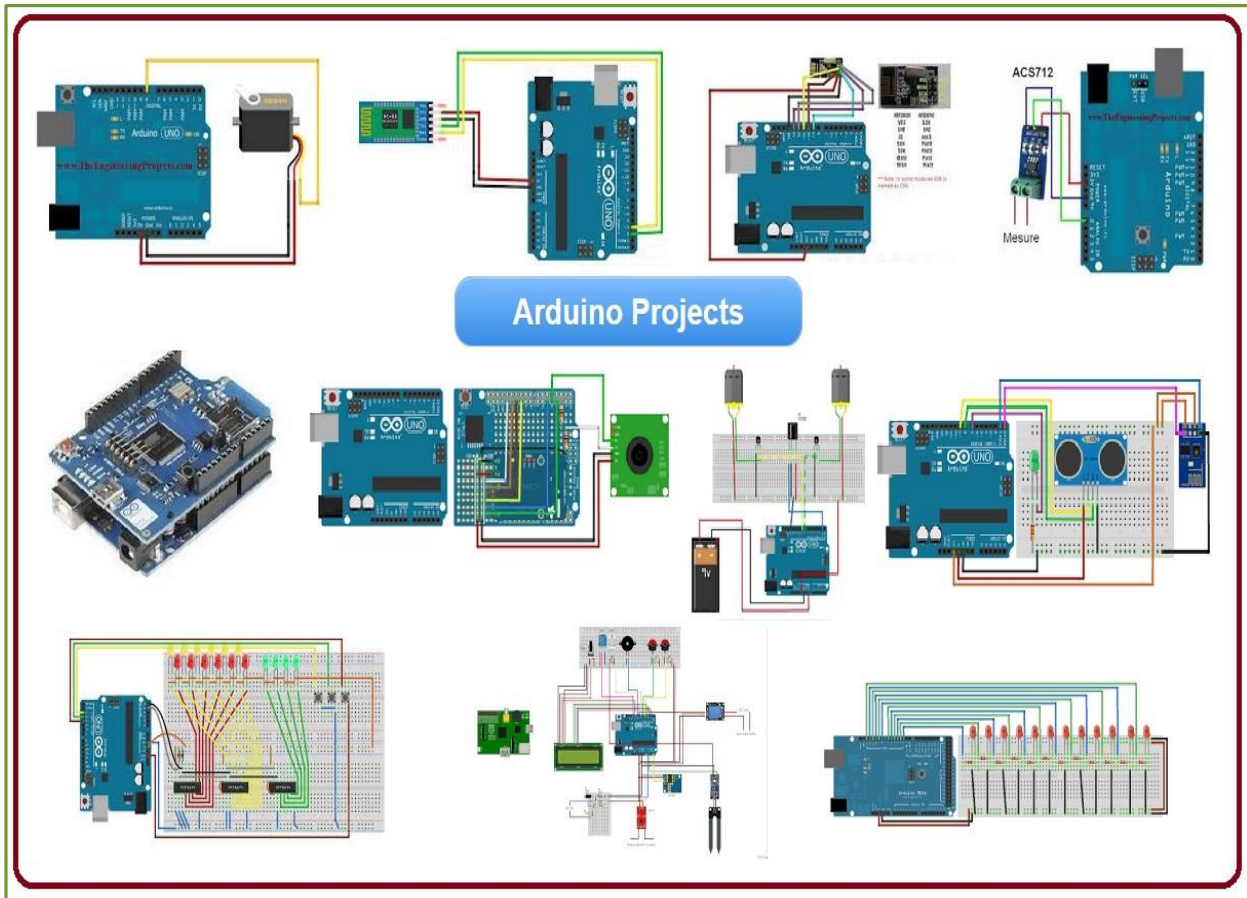
Give short answer to the following questions.

- Describe Wi-Fi Protocol.
- Describe ESP8266 module.
- What do you mean by a SoC?
- What is the difference between AP and station mode?
- What is the difference between ESP8266 and ESP32?
- What is Wi-Fi both mode?
- Describe Bluetooth protocol.
- Differentiate between Wi-Fi and Bluetooth protocol.
- What is Bluetooth Classic mode.

Practical Tasks

- Interface ESP8266 with an Arduino.
- Operate ESP8266 as an SoC in Station and Access Point mode.
- Write a code to run Wi-Fi and Bluetooth simultaneously using ESP32.

Chapter 5: Projects



After Studying this chapter, you will be able to:

- understand the working of 7-segment display.
- understand TM1637 module.
- use libraries in Arduino IDE.
- connect TM1637 with Arduino.
- display decimal numbers on TM1637 using Arduino
- understand the working of ultrasonic sensor.
- find distancing using speed and time.
- interface ultrasonic sensor with Arduino.
- record distance readings on serial monitor.

5.1 7-Segment Display

7-segment display consists of seven LEDs arranged in a rectangular fashion. Each LED is called a segment. The segments form part of a numerical digit (both Decimal and Hex) to be displayed (refer to Figure 5.1). An additional 8th LED is sometimes used within the same package thus allowing the indication of a decimal point, (DP) when two or more 7-segment displays are connected together to display numbers greater than ten.

Each LED in the display is given a positional segment (labeled LED) with one of its connection pins. These individual LED pins are labelled from *a* to *g* representing each individual LED. The other LED pins are connected together and wired to form a common pin. So, by forward biasing the appropriate pins of the LED segments in a particular order, some segments will be light, and others will be dark allowing the desired character pattern of the number to be generated on the display. This allows us to display each of the ten decimal digits 0 to 9 on the same 7-segment display. For instance, to display 7, segments *a*, *b* and *c* will be on.

The display's common pin is generally used to identify which type of 7-segment display it is. As each LED has two connecting pins, one called the "Anode" and the other called the "Cathode". Therefore, there are two types of LED 7-segment displays called: **Common Cathode (CC)** and **Common Anode (CA)**. The difference between the two displays is that the common cathode has all the cathodes of the 7-segments connected directly together and the common anode has all the anodes of the 7-segments connected together and is illuminated as follows:

1. **Common Cathode (CC):** In this display, all the cathode connections of the LED segments are joined together to logic "0" or ground. The individual segments are illuminated by application of a "HIGH", or logic "1" signal via a current limiting resistor to forward bias the individual Anode terminals (*a-g*).

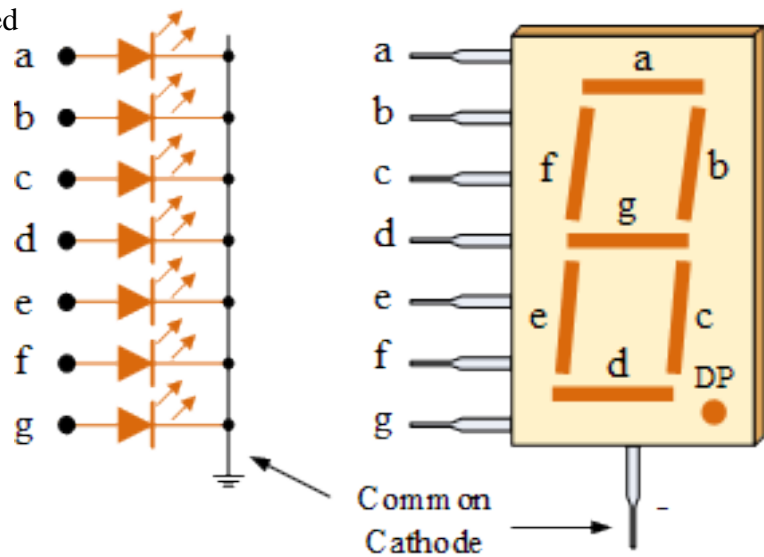


Fig. 5.1 7-Segment Display

2. Common Anode (CA): In CA topology, all the anode connections of the LED segments are joined together to logic “1”. The individual segments are illuminated by applying a ground, logic “0” or “LOW” signal via a suitable current limiting resistor to the Cathode of the particular segment (a-g).

5.2 Interfacing TM1637 with Arduino UNO

TM-1637 is a 7-segment 4-digit display (four 7-segment displays packed together) that can be easily controlled with a few wires on an Arduino. It is mainly applicable to the display drive of induction cooker, micro-wave oven and small household electrical appliances.

Instructions

1. Build the circuit according to the circuit diagram (Figure 5.2) given below:

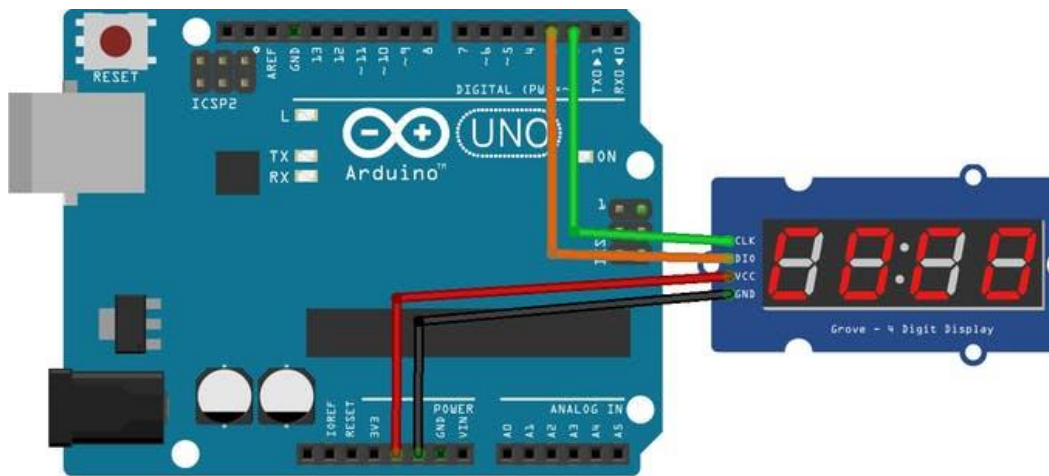


Fig. 5.2 Connecting TM1637 to Arduino

2. Install the TM1637 library in Arduino IDE by going to Sketch>Include Library>Manage Libraries. Then, type in "TM1637" and install the library labeled "Grove 4-Digit Display" by Seed Studio.

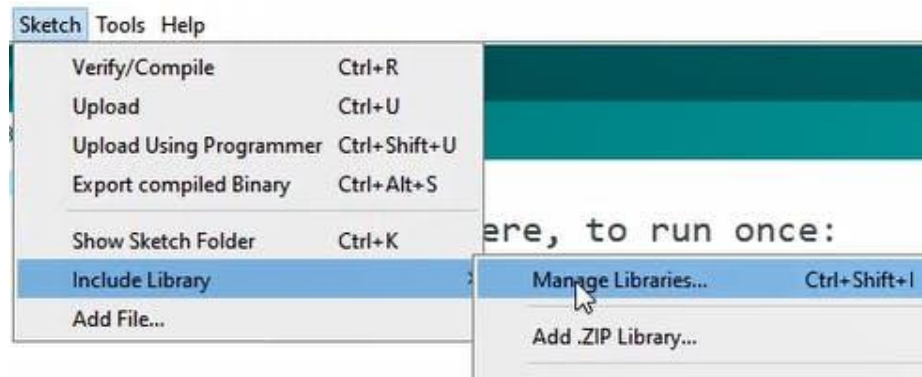




Fig. 5.3 Installing TM1637 library in Arduino UNO

3. Type in the setup code:

```
#include <TM1637.h>

int CLK = 2;
int DIO = 3;

TM1637 tm(CLK,DIO);

void setup() {
  // put your setup code here, to run once:
  tm.init();

  // set brightness; 0-7
  tm.set(2);
}
```

CLK and DIO can be set to any pin on the Arduino, they do not have to be pins 2 and 3.

To set the brightness, pass in a value between 0-7 into *tm.set()*;

4. Now you can display characters in the loop function with 2 methods:

a. *tm.display(<position>, <character>)*; will display a character on the display.

Position refers to the digit you want to change. The first is 0 and the last is 3.

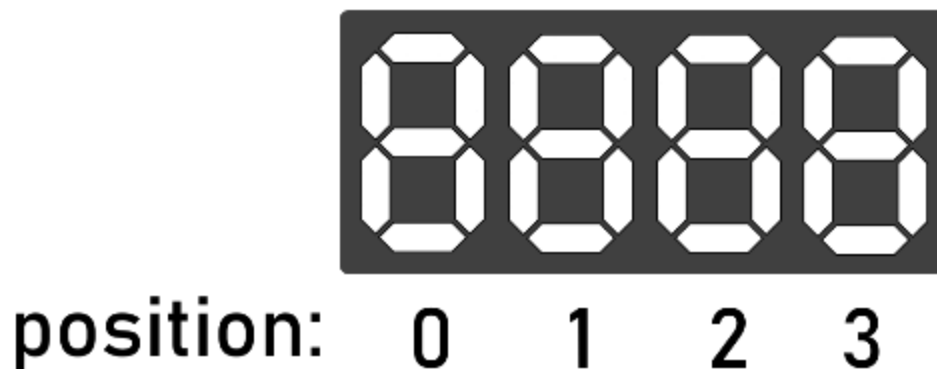


Fig. 5.4 Position in TM1637

Character refers to the character you want to display. 0-9 displays 0-9 and 10-15 displays A-F.

Input	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Output	0	1	2	3	4	5	6	7	8	9	A	b	C	d	E	F

Here is an example that displays "12:Ab":

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

  // example: "12:ab"
  // tm.display(position, character);
  tm.display(0,1);
  tm.display(1,2);
  tm.point(1);
  tm.display(2,10);
  tm.display(3,11);
}
```

5. Connect your Arduino to the computer and upload the sketch.

A Few Useful Functions

Display a number: To make displaying numbers easier, the following is a function that will display integers that you pass into it:

```
void displayNumber(int num){
  tm.display(3, num % 10);
  tm.display(2, num / 10 % 10);
  tm.display(1, num / 100 % 10);
  tm.display(0, num / 1000 % 10);
}
```

To make it display "1234" for example, you can just call `displayNumber(1234);`

Display Time: To make it display a time in the form of minutes: seconds based on the number of seconds, you can use this function:

```
void displayTime(int seconds){
  int minutes = seconds / 60;
  int seconds = seconds % 60;

  tm.point(1);
  tm.display(3, seconds % 10);
  tm.display(2, seconds / 10 % 10);
  tm.display(1, minutes % 10);
  tm.display(0, minutes / 10 % 10);
}
```

To display "15:30", you can just call `displayTime(930)`; (There are 930 seconds in 15 minutes 30 seconds).

5.3 Finding the Distance using Arduino UNO

Ultrasonic sensor HC-SR04 is a sensor that can measure distance. It emits an ultrasound at 40 000 Hz (40kHz) which travels through the air and if there is an obstacle on its path, it will bounce back to the module. Considering the travel time and the speed of the sound you can calculate the distance.

The configuration pins of HC-SR04 are VCC (1), TRIG (2), ECHO (3), and GND (4). The supply voltage of VCC is +5V and you can attach TRIG and ECHO pin to any Digital I/O in your Arduino Board.



Fig. 5.5 Ultrasonic Sensor

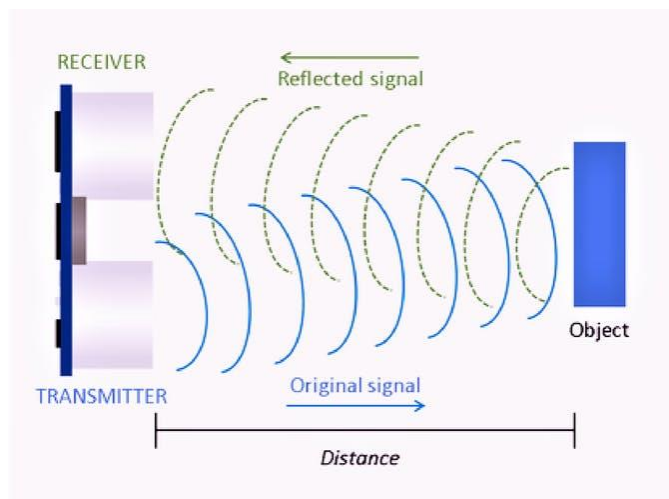


Fig. 5.6 Ultrasonic Sensor HC-SR04

The components required to make this project are:

1. Arduino UNO
2. Ultrasonic Sensor HC-SR04
3. Male to Male Jumper Wires
4. Breadboard

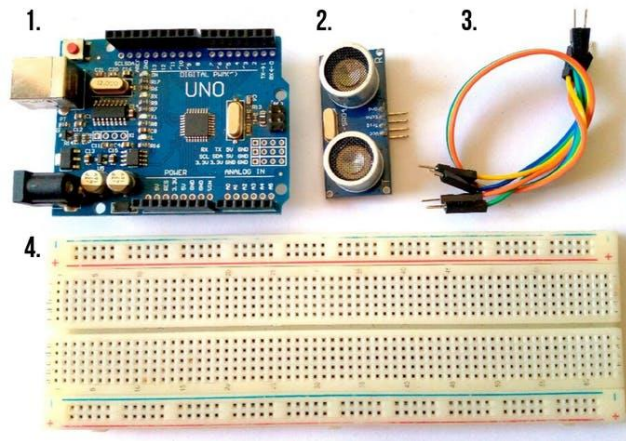
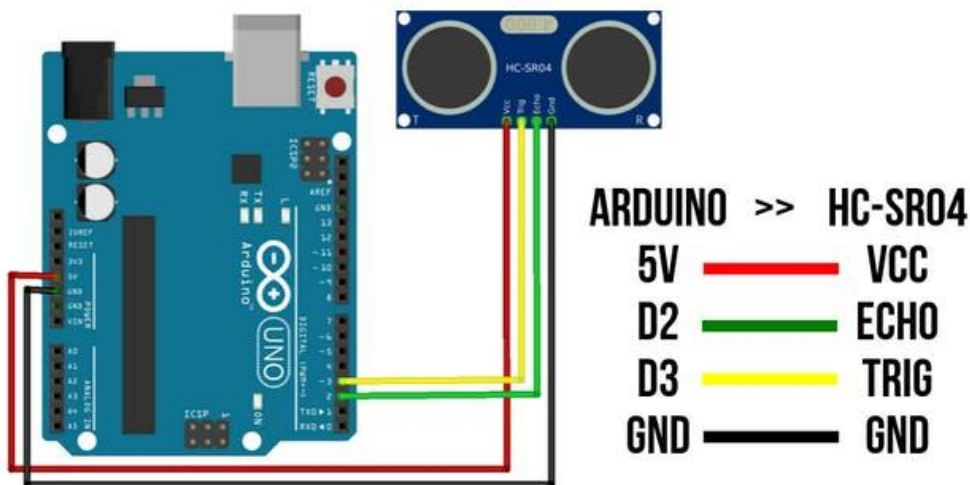


Fig. 5.7 required components to record distance using ultrasonic sensor

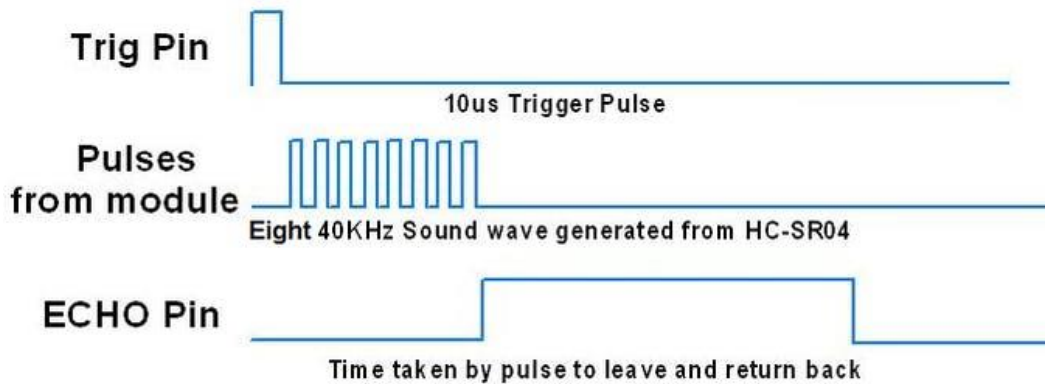
The connection of Arduino with Ultrasonic Sensor HC-SR04 is:



Connection of Arduino UNO and HC-SR04

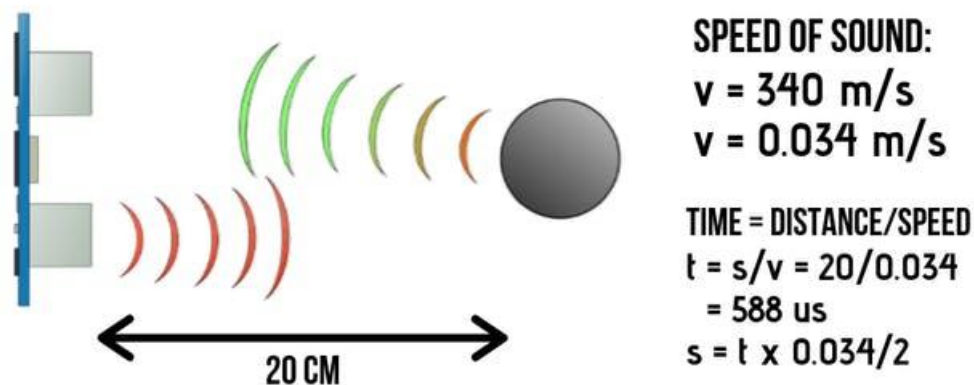
In order to generate the ultrasound waves, we need to set the *trigger pin* on a *high state* for 10 μ s. That will send out an 8-cycle sonic burst which will travel at the speed of sound and it will be received in the Echo Pin. The Echo Pin will give output in terms of time in microseconds which the sound wave travelled.

Ultrasonic HC-SR04 module Timing Diagram



Ultrasonic HC-SR04 timing diagram

For example, if the object is 20 cm away from the sensor, and the speed of the sound is 340 m/s or 0.034 cm/ μ s the sound wave will travel about 588 microseconds. But what you will get from the Echo pin will be double that number because the sound wave needs to travel forward and bounce backward. So, in order to get the distance in cm we need to multiply the received travel time value from the echo pin by 0.034 and divide it by 2.



Distance calculating

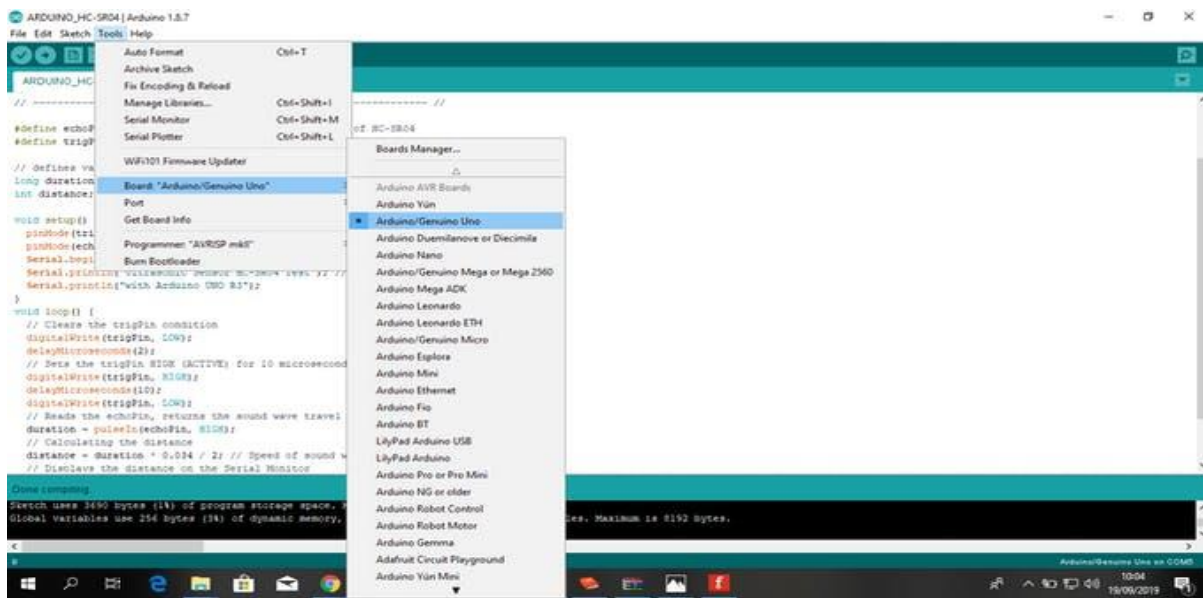
For the code, we need to define the *trigger pin* and *echo pin* that is connected to Arduino board. In this project *EchoPin* is attached to D2 and *TrigPin* to D3. Afterwards, Define variables for the distance (int) and duration (long).

In the loop function, you have to make sure that the *trigPin* is clear so we have to set that pin on a LOW state for just 2 μ s. Now for generating the ultrasound wave, we have to set

the trigPin on HIGH state for 10 μ s. Using the *pulseIn()* function, you have to read the travel time and put that value into the variable “duration”. This function has 2 parameters, the first one is the name of the echo pin and for the second one you can write either HIGH or LOW. In this case, HIGH means that the *pulseIn()* function will wait for the pin to go HIGH caused by the bounced sound wave and it will start timing, then it will wait for the pin to go LOW when the sound wave will end which will stop the timing. At the end the *pulseIn()* function will return the length of the pulse in microseconds. To calculate the distance, we multiply the duration by 0.034 and divide it by 2. Finally, values of the distance are printed on the Serial Monitor.

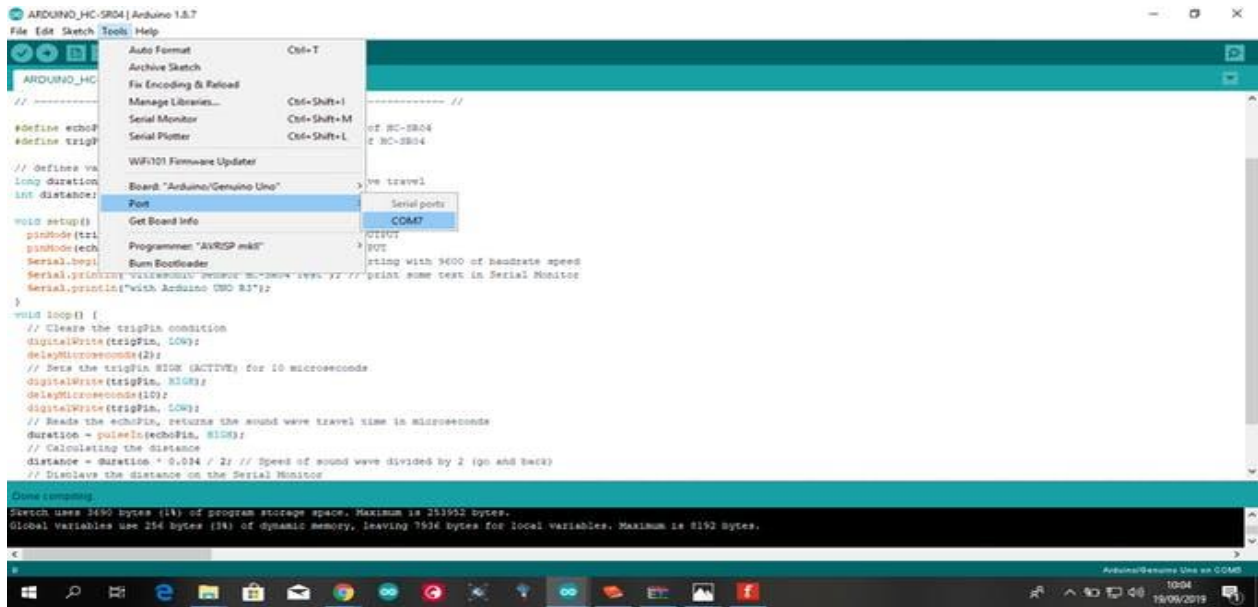
Steps

1. First do the wiring as shown in the picture.
2. Open Arduino IDE Software and write down your code, or download the code below and open it
3. Choose your own Arduino board (in this case Arduino UNO), by selecting Tools > Board > Arduino/Geniuno Uno
4. Choose your COM Port: Tools > Port > COM.
5. Upload your code by pressing Ctrl + U or Sketch > Upload
6. To display the measurement data you can use Serial Monitor by pressing Ctrl + Shift + M (make sure that the baudrate speed is 9600).



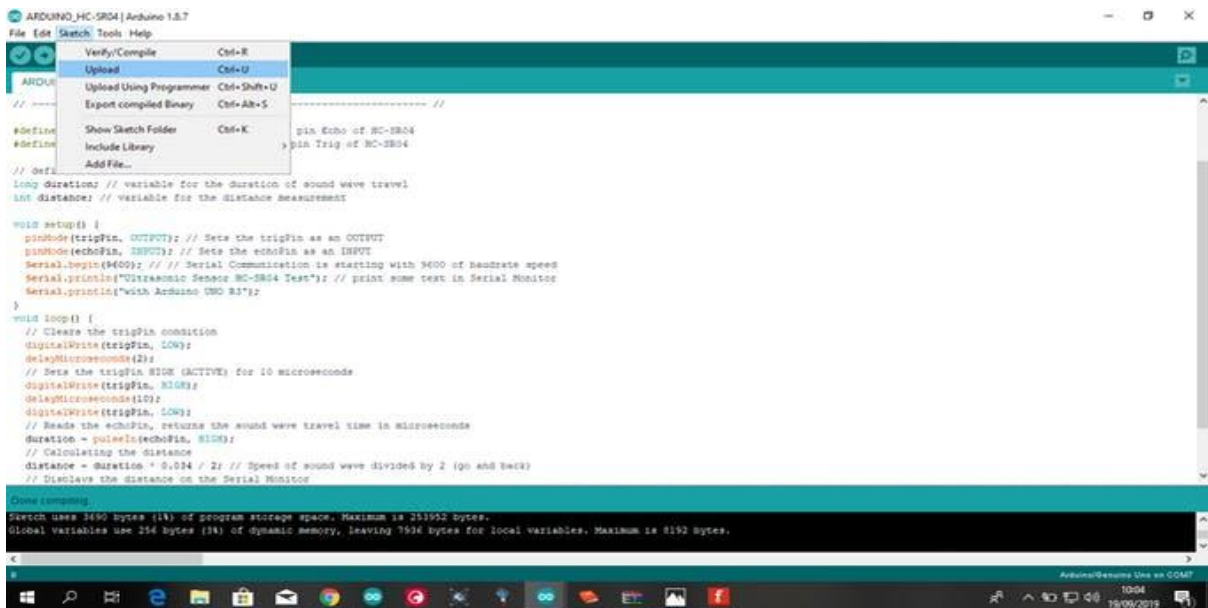
Choosing Arduino Board

Internet of Things Hardware Development



Select your own Arduino COMPort

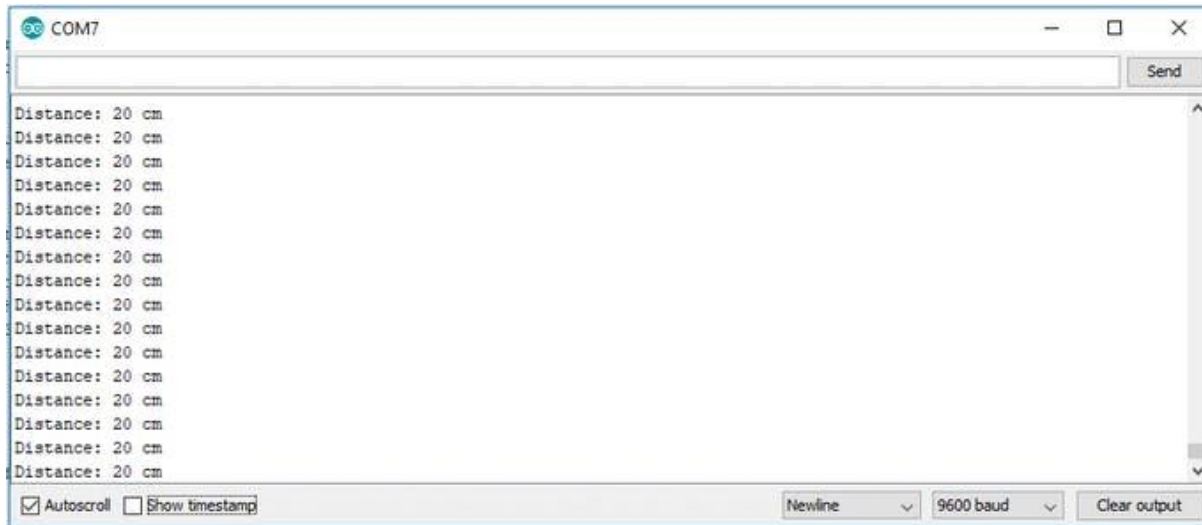
Teacher Notes: Some other libraries are also available in Arduino IDE to find Distance. Even, you can also find distance without using any library. These approaches can also be discussed with students.



Uploading the code

Results

After uploading the code, display the data on Serial Monitor. Now place an object in front of the sensor and observe the measurement.



For the consideration, you can use your manual tape meter to measure the distance and compare it with the distance on Serial Monitor. If you want to display it on LCD, you can follow the second wiring diagram and upload the code below.

```
// -----
---- //
// Arduino Ultrasonic Sensor HC-SR04
// Re-written by Arbi Abdul Jabbaar
// Using Arduino IDE 1.8.7
// Using HC-SR04 Module
// Tested on 17 September 2019
// -----
---- //

#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

// defines variables
long duration; // variable for the duration of sound wave
travel
```

```
int distance; // variable for the distance measurement

void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT
  pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT
  Serial.begin(9600); // // Serial Communication is starting
with 9600 of baudrate speed
  Serial.println("Ultrasonic Sensor HC-SR04 Test"); // print
some text in Serial Monitor
  Serial.println("with Arduino UNO R3");
}
void loop() {
  // Clears the trigPin condition
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in
microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2; // Speed of sound wave
divided by 2 (go and back)
  // Displays the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
}
```


Key Points

- The *7-segment display*, also written as “seven segment display”, consists of seven LEDs.
- There are two types of LED 7-segment display called: Common Cathode (CC) and Common Anode (CA).
- In the common cathode display, all the cathode connections of the LED segments are joined together to logic “0” or ground.
- The individual segments are illuminated by application of a “HIGH”, or logic “1” signal via a current limiting resistor to forward bias the individual Anode terminals (a-g).
- In the common anode display, all the anode connections of the LED segments are joined together to logic “1”.
- The individual segments are illuminated by applying a ground, logic “0” or “LOW” signal via a suitable current limiting resistor to the Cathode of the particular segment (a-g).
- The TM-1637 (also called the Grove 4-Digit Display by Seeed Studio) is a 7-segment 4-digit display that can be easily controlled with a few wires on an Arduino.
- Ultrasonic Sensor HC-SR04 is a sensor that can measure distance.
- It emits an ultrasound at 40 000 Hz (40kHz) which travels through the air and if there is an object or obstacle on its path it will bounce back to the module.
- In order to generate the ultrasound waves, we need to set the Trigger Pin on a High State for 10 μ s.
- For example, if the object is 20 cm away from the sensor, and the speed of the sound is 340 m/s or 0.034 cm/ μ s the sound wave will need to travel about 588 microseconds.

EXERCISE**Choose the most suitable option**

1. Which Interfaces is available on TM1637?
a. I2C b. SPI c. UART d. None of these
2. TM1637 has how many 7-segments joined together?
a. 4 b. 5 c. 2 d. None of these
3. Ultrasonic sensor is used to find:
a. Temperature b. Humidity c. Intensity of Light d. Distance
4. To display 7, which of the LEDs will be on in a 7-segment display:
a. a, b, c b. d,e,f c. g, a, b d. b, c, d
5. A 7-segment contains how many LEDs?
a. 3 b. 1 c. 4 d. 7

Give short answer to the following questions.

1. Describe 7-segment display.
2. What is the difference between Common Anode and Common Cathode 7-segment display?
3. Describe the working principle of TM1637.
4. Describe ultrasonic sensor and its transducing principle.
5. Describe HC-SR04 ultrasonic sensor and its working principle.
6. How a distance from an object is calculated using ultrasonic sensor?

Numericals

1. Interface TM1637 with Arduino.
2. Interface HC-SR04 with Arduino.

Chapter 6: Freelancing



After Studying this chapter, you will be able to:

- define freelancing and know different freelancing platforms.
- explain the terminologies related to freelancing platforms.
- set up a seller profile and add personal and professional information on your profile.
- link up social media and other professional accounts to seller profile.
- find your ideal category and services and check out the competition.
- create an appealing title for the gig and choose subcategory and tags.
- create and price gig packages and win buyers with a gig description.
- choose a suitable gig package among basic, standard and premium options.
- understanding of niche and projects lying in a niche.
- find the right project according to your niche.
- understand requirements by reading the project description and demands.
- ask questions to clarify any ambiguities.
- offer a mockup and setup a competitive fee for the project.
- review your bid proposal and submit the bid proposal.
- ask for feedback from clients and keep in touch with buyers/customers.
- use the contacts page to maintain close coordination with the potential buyers/customers.
- request customer to recommend you to other clients and work circles.
- understand payment methods for freelance platforms.
- communicate with clients using the freelance platform messaging service.
- know about use of payment protection methods to secure your reward.

6.1 Freelancing

Freelancing means working as an individual to form an independent company rather employed by someone. Freelancers are self-employed and often referred to as independent contractors. Freelancers are hired by other companies on a part time or short-term basis, but they do not receive the same compensation as full-time employees or have the same level of commitment to any company. With the rise of gig-economy, freelancing has become more popular. A study by Upwork and Freelancers Union shows that 57 million Americans freelanced in 2019.

6.1.1 Freelancing Platforms

The following are the freelancing websites to find work:

- | | |
|------------------|---------------------|
| 1. Fiverr | 2. Upwork |
| 3. Toptal | 4. LinkedIn |
| 5. Flexjobs | 6. SimplyHired |
| 7. Guru | 8. Freelancer.com |
| 9. Behance | 10. 99designs |
| 11. Dribbble | 12. People Per Hour |
| 13. ServiceScape | |

6.1.2 Buyer vs Seller Profiles

Freelancers sell their time and knowledge to complete someone's work in return of money so, the freelancers are the sellers on Fiverr. Whereas, the company or any individual or any organization who wants to get his work completed, buy the work or knowledge from the freelancers in return of money they provide, thus they are buyers.

6.1.3 Freelancing terminologies

Gig: Gig is a portfolio of your work. A Fiverr gig is a term commonly used to describe the service you sell on Fiverr. It provides you with the opportunity to showcase your unique talent and inform buyers that you are well and truly capable of completing the service you render. Since you are mostly dealing with buyers whom you have never met before, the only way they can discover your quality is through your gig.

Bid: You might have heard of the word 'auction', which is most frequent in real estate. Sellers present goods or services and offer them up for bidding. Interested buyers will then take bids on the item which is sold to the highest bidder. This is known as Ordinary Auction (also Forward

Auction). There is another auction called the Reverse Auction in which the roles of sellers and buyers are reversed. The initiator of the transaction is now the buyer. Whereas, sellers compete to win the contract. However, the buyer does not necessarily choose the bidder offering the lowest price. It is up to him to select who's awarded the job based on multiple parameters provided by sellers.

Reverse Auction (Reverse Bidding) is the principle behind project bidding on a freelance bidding site. A client will post a project on a freelance platform, then relevant freelancers will bid for the project, write a proposal to persuade the client. Client will review freelancers' proposals, portfolios, profiles, reviews and ratings to decide who gets the project. Another feature available in Reverse Bidding is *Bid Invitation*. Instead of waiting for freelancers to bid on the project, clients can actively invite their favourite freelancers to bid.

Rating: A freelancer's ratings help clients evaluate their overall performance on Freelancer.com. The higher the ratings, the more credible and trustworthy a freelancer becomes towards clients. Ratings shown on a freelancer's profile page are calculated from all projects they have worked on.

Revision: Revisions give buyers an option to reject a seller's delivery and provide sellers with more time to include buyer's feedback and any changes required based on the delivered work. Revisions are an optional feature that sellers may choose to offer to their buyers.

6.2 Creating a Seller Profile on Fiverr

Your profile is how you present yourself to the community. You should present yourself in a professional manner.

Creating a seller profile

Step 1: Personal Info

1. Fill out your full name.
2. In the **Profile Picture** section, add your picture.
3. In the **Description** section, introduce yourself to the Fiverr community.
4. In the **Languages** section, select the languages you speak and your level of fluency.
5. Click **Continue**.

Step 2: Professional Info

1. **Your Occupation:** Select your expertise. Once selected, you will see skills appear with checkboxes. You need to choose at least two.

2. **Skills:** These may be skills gained through your previous jobs, hobbies, or even everyday life. You have the option to suggest skills that are not listed. You can add up to 15 skills.
3. **Education:** Select when and where you attended college. You also have the option to suggest your diploma if it is not listed.
4. **Certifications:** Listing your honours and awards can help you stand out from other sellers. You have the option to suggest certifications.
5. **Personal Website:** Enter your URL.
6. Click **Continue**.

Step 3: Linked Accounts

1. In the **Linked Accounts** section, link your social and professional networking accounts to your Fiverr account. You may select more than one.
2. Click **Continue**.

Step 4: Account Security

1. In the **Email** field, click **Verify Now** and follow the prompts, if you have not verified your account yet.
2. In the **Phone Number** field, click **Verify Now** and continue with the process.
3. When finished, click **Continue & Create Your First Gig**.

6.3 Creating a Gig on Fiverr

On Fiverr, the service you offer and sell within the marketplace is termed as a "Gig." Your Gig is an opportunity to showcase your talent to potential customers, as well as to provide all the information they may need prior to placing an order.

To create a Gig

- Click **Switch to Selling > Gigs** after you login.
- Click **Create A New Gig**.
- On the new page that opens, you will see the message: "It seems that you don't have any active Gigs. get selling! Click **Create a new Gig**."

Step 1: Gig overview

- Write a title in the **Gig Title** field. Make sure that your Gig title is short, clear, and to the point.

- In the **Category** drop-down menu, select the appropriate category and subcategory for your Gig. Fiverr recommends categories and subcategories based on your title. To select a category, just click the relevant link.
 - *Note: Gaming is not a category. If you provide services related to Gaming, select a category related to Gaming (or define the criteria in the Gig Metadata area further down).*
- In the **Service Type** drop-down menu, choose a service type to ensure that you appear in the search results related to the category you selected.
- In the **Gig Metadata** area, select the relevant criteria for each topic.
- In the **Search Tags** field, add one to five words or phrases that best describe your Gig, then **Save & Continue**.

The screenshot shows a breadcrumb trail at the top: Overview > 2 Pricing > 3 Description & FAQ > 4 Requirements > 5 Gallery > 6 Publish. Below this is a form with the following elements:

- GIG TITLE:** A text input field containing the placeholder text "I will do something I'm really good at" and a character count "0 / 80 max". A red arrow labeled "1" points to this field.
- CATEGORY:** A dropdown menu with the text "SELECT A CATEGORY". A red arrow labeled "2" points to this dropdown.
- SEARCH TAGS:** A text input field with the text "Up to 5 terms" below it. A red arrow labeled "3" points to this field.
- Buttons:** A "Cancel" button on the left and a green "Save & Continue" button on the right. A red arrow labeled "4" points to the "Save & Continue" button.

Fig. 6.1 Fields Required to Create a Gig

Step 2: Scope and pricing

- In the **Name your package** area, write a brief title for the service you are offering.
- Describe the details of your offering.
- In the **Delivery Time** drop-down menu, select the number of days it will take for you to complete your Gig.
- Select the various items you would like to include in your service (Extras). This depends on the type of service you are offering. For example, if you are offering logo design

services, you can select options from a predefined list of items, such as the number of logos to include, the source files, or a high-resolution file.

- In the **Revisions** drop-down menu, select the number of revisions you include in your Gig.
- In the **Price** drop-down menu, select a price for your service.

The screenshot shows a 'Scope & Pricing' interface with a '3 Packages' indicator. Below the title, there is a table with three columns: BASIC, STANDARD, and PREMIUM. The table has several rows for configuring each package:

	BASIC	STANDARD	PREMIUM
Name your package	Name your package	Name your package	Name your package
Describe the details of your offering	Describe the details of your offering	Describe the details of your offering	Describe the details of your offering
Delivery Time	Delivery Time	Delivery Time	Delivery Time
Revisions	Select	Select	Select
Double-Sided	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
High Resolution	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 6.2 Pricing of a Gig

Adding Extra Services

You have the option to deliver your Gig **ExtraFast** and charge accordingly.

To set ExtraFast delivery:

In the **My Gig Extras** section, select the **ExtraFast delivery** and indicate how many days it will take you to deliver, and how much more you will charge.

*Note: If you select **ExtraFast delivery**, the amount of days for the extra fast delivery cancels out the amount of time you indicated in your extra services.*

In addition, you can offer extra services for your Gig, which will be available to the buyer during the order process. These fields are optional. To offer Extras:

- In the **My Gig Extras** area, select from the list of items available.
- If you would like to add unique customized Gig Extras, then click **Add Gig Extra**, then fill out a title, description, amount of money you will charge, and indicate how long it will take you to complete.

- *Note: For each extra provided, select the checkbox, then set your price and how long it will take you to complete.*

Shipping & Handling

Some of the services on Fiverr are delivered physically, such as arts and crafts, collectible items, and so on. If your service includes physical shipping, please be sure to select this option in your Gig:

- Select the checkbox and select how much you charge to a specific country and then to everywhere else.
- Click **Save & Continue**.

Step 3: Description and FAQs

- In the **Description** field, briefly describe your Gig (up to 1,200 characters). Be as detailed as possible so buyers can determine if your Gig meets their needs.
 - *Note: If you add a URL, only include URLs from Fiverr's approved list.*
- In the **Frequently Asked Questions** area, click **Add FAQ**.
 - *Note: Here, you can add answers to the most commonly asked questions. Your FAQs will be displayed on your Gig page.*
- Add a question you think buyers may ask, and then write your answer.
- Click **Add**.
- When finished adding your FAQs, click **Save & Continue**.

Step 4: Requirements

You can tell your buyer what you need from them to start the order. You can define instructions for your buyer as free text, a multiple-choice question, or a file upload.

- In the **REQUIREMENT** field, enter the details of your requirement.
- In the **Answer Type** field, select either **Free Text**, **Multiple Answer**, or **Attached File**.
 - *Note: If you select **Multiple Answer**, answer fields will be displayed. Complete the multiple-choice answers from which buyers must select.*
- If you require the buyer to provide the information you selected, mark the **Answer is mandatory** checkbox.
- Click **Add**.
- When you have finished defining your buyer requirements, click **Save & Continue**.

STEP 5: Gallery

- In the **Gig Photos** area, drag an image or click **Browse** and select an image. Upload up to three photos that describe or relate to your Gig. The photos can be samples of your work. Each image should be in JPEG, PNG, GIF, or BMP format, and up to 50 MB of size. The gig must have at least one image uploaded to be published. The image should be owned by you. Do not use copyrighted images or images downloaded from the internet.
- In the **Gig Video** area, drag a video or click **Browse** and select a video.
 - *Notes:*
 - *This field is mandatory for the Video & Animation category but optional for all other categories.*
 - *Once you add a video, it will be used for the thumbnail in the marketplace (in place of the Gig photos).*
- In the **Gig Audio** area, drag an audio file or click **Browse** and select a file.
 - *Notes:*
 - *Supported formats are mp3 and MPEG.*
- In the **Gig PDF** area, drag a PDF or click **Browse** and select a PDF.
- Click **Save & Continue**.

STEP 6: Publish

Publishing and Promoting Your Gig

1. Click **Publish Gig**.

Once your Gig is posted, it is published. You can promote your Gig. To promote your Gig, just click an icon and fill out the details.

2. Click **Done**.

*Note: If your Gig contains a video, its status will be pending until your Gig video is reviewed. If your Gig does not contain a video, it will be labeled **Active**.*

To edit a Gig

1. After you log in, from the **Selling** menu, click **Gigs**.
2. Find the Gig you want to edit, and from the drop-down list on the right, select **Edit**.
3. Make the changes you want to make and click **Save**.

6.4 Submitting a bid Proposal

Niche

A niche is a specialization in either a *topic* or a *content type*. A topic is “what” you write about, or you work on, and it is probably the first thing you think of about a niche. Examples include personal finance, digital marketing, and technology. Some examples are blog posts, case studies, and white papers.

What is a Bid Proposal?

A bid proposal is very specific summary explaining why you are eligible to get the job. It does more than initiating a friendly greeting to the buyer and express interest in the job. Your bid proposal allows you to state what your qualifications are and how your background would allow you to perform the specific job with excellence and accuracy.

The qualifications and skills may include:

- Skills for that job
- Education
- Past Projects
- Experience

Step# 1. Read the Project Description Carefully

If the employer does not realize you understand the project, you are not likely to win the bidding. Many employers ask for specific details that you need to be aware of. In fact, employers often include a phrase that must be included in your bid to have it considered. You should always take the time to go through the description thoroughly.

Step# 2. Use the Project Clarification Board

Use the Project Clarification Board if you have questions. Winning a project without knowing exactly what you are getting into is not a good situation for you.

Step# 3. Keep Your Bid Clear

Keep your bid clear, concise and to the point. Remember that the employer may have dozens or even hundreds of bids to consider. It is very likely that every word of every bid is not going to be read. Bids with unnecessarily long descriptions may be skipped over completely.

Step# 4. State Your Terms Clearly

Be as precise as possible in stating exactly what you provide, how much it will cost, and how long it will take to deliver. Being vague about your terms implies a lack of confidence. If you are not confident in yourself, the employer will not be, either.

Step# 5. Respond Promptly

If the employer contacts you through a private message, be sure to respond promptly. Most employers award projects within the first 24 hours of posting.

Step# 6. Provide Links to Online Examples

It is always a good idea to upload samples with your private message or provide links to online examples. Be sure that your examples are appropriate for the job and represent your best work.

Step# 7. A Word of Caution

Unless you are prepared to give your work away, any samples you provide should bear a watermark or other means of identification or at the very least your name and a statement of copyright.

Step# 8. Be Competitive with Your Pricing

It does not necessarily mean you need to be the lowest bidder. On the other hand, if you are relatively new to freelancing, you may need to establish a reputation first.

Step# 9. Do not Oversell Yourself

A little self-confidence is a good thing, but over-the-top claims probably will not impress anyone.

Step# 10. Proofread Your Bid Before You Submit it

Proofread your bid before you submit it. No matter what kind of project you are bidding on, a poorly written proposal suggests a lack of interest and poor work habits.

What to avoid when bidding

You should refrain from doing any of the following practices to avoid incurring penalties:

- Using generic proposals (bids with the exact same content) on multiple projects.
- Bidding on and accepting projects in which you do not have the required expertise.
- Bidding with a very low amount and increasing the amount significantly once awarded.
- Giving out or asking for contact information in your bid.

1. Create a Project Prototype

A **prototype** is a blueprint or sample of the whole project that you are going to start. It gives you a short overview of your result.

2. Buyer Review on Prototype

When you have done with the prototype, it is time to start getting ROI from this effort (prototype). After creating your prototype, you must send this to your client and ask him to review it. Do not forget to mention, it is just a prototype, and feel free to suggest any changes. The following is the exact script that you can use:

These are some benefits of the prototype that you're going to get;

1. Build a strong relationship with the buyer
2. Give idea, you are on the right track
3. Minimize surprises and revisions
4. Saves time and energy

3. Move to the final project

Now you have got a clear picture and exact steps, you need to take to get the outcome that you want to achieve. So, you can start proceeding towards the final work to complete the project. This project will help you to build your portfolio for future orders.

Pro Tip: While working on the final project, you've to deliver as much as possible. One thing doesn't forget to fulfil every promise that you made with the buyer in the gig description.

Step to send project file

First go to the Fiverr and open the order page of your project. Now at the bottom of the page, you will see a **Deliver Now** button on the green line.

Click on this button and it will open a pop-up. Add a zip file and project image using **upload work** button.

Next, there is a chat box where can say thanks to your buyer. You can also use a quick response. Now, just click the deliver button and your project will be delivered to the buyer. Now have patience and wait until the buyer mark this project done.

6.5 Payment Methods and Protecting a Transaction

Paper checks

Traditional approach to send payments to hired talent may be outdated, but it is still effective. Sending a check will help you avoid any service fees that come with other payment options. Paper checks come with several disadvantages. They take time to get to your hired talent and may even get lost during delivery.

PayPal

PayPal is among the most common methods clients use to pay. Opening an account is free, and their services make it easy to send invoices to freelancers. When you make a payment, the freelancer receives funds immediately in their PayPal account.

PayPal lacks the organizational tools, escrow accounts, security services, and other features that are useful for effectively tracking and paying your freelancers.

The more freelancers you work with, the more likely you will want to invest in a more comprehensive payment platform.

Direct Contracts

This is Upwork's payment solution, but keep in mind that you can use Direct Contracts even when paying professionals that don't have an Upwork account. Direct Contracts lets you enjoy the benefits of our escrow services, simple contract management, faster payments, and dispute assistance, even when working with freelancers outside of Upwork's marketplace.

Escrow service ensures your money is protected and will only be released when the hired talent has completed your work. If at any time you're dissatisfied with your contract, you can cancel the contract, so long as you do so before you approve the payment. Once canceled, your funds will return to your original payment method within seven business days.

Every Direct Contract is managed via email so that we can provide you simple and easy-to-follow instructions through every step of the process. All relevant contract details, such as the contract's name, description, price, your company name, and the freelancer's name, are delivered directly to your inbox. There are no fees associated with this service as a client, so you can easily know that you will only be paying the value of your contract.

Work marketplaces

By using work marketplaces, such as Upwork, you can securely and easily pay freelancers from around the world.

Setting up your account

You can sign up for Upwork with Google, Apple, or by entering in your work email. You will have to add a billing method to your account before you can hire any of the freelancer on our marketplace. To do so, head over to your Account Menu, select Settings, and then click on Billing Methods.

As a client, you have three billing methods available to you:

1. Credit card: Visa, Discover, Mastercard, and American Express.
2. A verified PayPal account.
3. A bank account for qualifying clients.

To avoid any account disruption in the event of a transaction failure, consider adding more than one billing method to your account.

Paying for hourly contracts

You have three options when it comes to paying hourly contracts.

1. Weekly billing: Every Monday, you will be invoiced for the previous week's hours based on your hired talent's Work Diaries. This is charged automatically to your default billing method.
2. Recurring weekly billing: This is a set, recurring weekly payment that is invoiced every week on top of your hired freelancer's submitted hours.
3. Manual payment or bonus: This is an additional payment you can deliver to your freelancer. This may be to reimburse them for certain expenses, a completed milestone, to provide overtime pay, or simply to give them a bonus for all their hard work.

Paying fixed-price contracts

Fixed-price contracts can be paid all at once or by milestone, depending on which is preferable to you as the client. By using escrow, you can easily know that your funds are sent to your hired professional *only* after your milestones are completed and approved.

Upwork Payment Protection

Upwork Payment Protection adds a level of security to payments for both hourly and fixed-price projects, allowing clients and freelancers to collaborate on projects with peace of mind.

For hourly projects:

Freelancers can opt to use the Work Diary time-tracking tool to automatically invoice for the time they worked. This enables you to quickly verify that the freelancer's invoice only includes hours properly billed to your project. This feature underpins Payment Protection as proof of activity.

Refunds and disputes

In the event that you need to request a refund, you'll have 180 days from the date of payment to get your money back through Upwork. It should be noted, however, that your hired professional can then choose to issue the full refund, a partial refund, or no refund at all. They can also choose to proactively initiate a refund without your formal request.

If a freelancer approves your refund request, it will automatically be credited back to your billing method. If the freelancer does choose to decline your request, you can dispute the charges. For hourly projects, disputes must be filed by Friday after a billing week closes.

Credit cards

Freelancer may use their own online payment solution like Freshbooks or Quickbooks. So long as the hired talent is using a payment solution that allows for credit card payments, this method will be a viable option for you.

From the freelancer's perspective, payments aren't secured before a project starts so they run the risk of not being paid. And from a client's perspective, the burden of managing multiple payments falls squarely on them. Both freelancers and clients alike stand to benefit from handling credit card transactions through more comprehensive platforms such as Upwork's direct contracts.

Electronic funds transfer

What if your freelancer doesn't want to be paid via check or an online payment solution? In situations like this, you can reliably pay them through an electronic funds transfer (EFT). This method is easy and doesn't usually come with pesky service fees. Your payment simply

travels from your bank account to the freelancer's; no credit card processing services or third-party apps necessary.

Similar to check payments, there may be a 1–2 day delay before your freelancer receives their funds, but you can trust that it will arrive safely and without any bumps along the way. If you're sending funds from one bank to another, you or your freelancer's bank will have to accommodate that transfer request, which is sometimes difficult and even costly. So, unless you or your company uses a robust payroll provider, a different payment method may be a better choice in the long-run.

Teacher Notes: Build a profile on any freelance website to give as an example to student.

Key Points

- 1 Freelancing means to work as an individual to form an independent company rather than be employed by someone else.
- 2 Freelancers are self-employed and often referred to as independent contractors.
- 3 Freelancers are hired by other companies on a part time or short-term basis, but they do not receive the same compensation as full-time employees or have the same level of commitment to any particular company.
- 4 A person who is selling his knowledge like the freelancers are doing, they are selling their time and knowledge to complete someone's work in return of money so, the freelancers are the sellers on Fiverr.
- 5 The company or any organization who wants to get his work completed, they buy the work or knowledge from the freelancers in return of money they provide.
- 6 A Fiverr gig is a term commonly used to describe the service you sell on Fiverr.
- 7 Reverse Auction (Reverse Bidding) is the principle behind Project Bidding on a freelance bidding site.
- 8 A freelancer's ratings help clients evaluate their overall performance on Freelancer.com. The higher the ratings, the more credible and trustworthy a freelancer becomes towards clients.
- 9 Revisions give buyers the option to reject a seller's delivery and provide sellers with more time to include buyer feedback and any changes required, as based on the delivered work.

Exercise

Choose the most suitable option

1. Which of the following does not apply to freelancers?
a. employed by others b. Self-employed c. hire others d. None of these
2. _____ refers to the services you sell?
a. Niche b. gig c. rating d. None of these
3. Project bidding works on the principle of:
a. Auction b. Reverse Auction c. lease d. None of these
4. Which is an unreliable payment method:
a. Paper Cash b. Escrow c. Both a, b d. None of these
5. The higher the _____, the more credible a freelancer is?
a. gig b. profile c. niche d. rating

Give short answer to the following questions

1. Define Freelancing.
2. What are the common freelancing terminologies?
3. Describe the steps to create a seller account on Fiverr.
4. What do you mean by Reverse Bidding?
5. What is a niche and how to find projects related to a niche.

Practical

1. Create a seller account on Fiverr.
2. Place a bid for a project on Fiverr.
3. Win a project related to IoT on Fiverr.

Chapter 7: Entrepreneurship



After Studying this chapter, you will be able to:

- define entrepreneurship.
- know the key concepts of entrepreneurship.
- understand main components of entrepreneurship.
- Learn different types of entrepreneurships.
- know how to identify a business opportunity.
- know how to develop a feasibility and a business Plan.
- understand about the concepts of marketing and marketing mix.
- understand 6 Ps of marketing.
- apply costing and pricing.

7.1 Entrepreneurship

Entrepreneurship is the ability and readiness to develop, organize and run a business enterprise, along with any of its uncertainties to make a profit. The most prominent example of entrepreneurship is the starting of new businesses. For instance, Facebook, eBay and Microsoft etc.

Entrepreneurship connected with land, labour, natural resources, and capital can generate a profit. The entrepreneurial vision is defined by discovery and risk-taking and is an indispensable part of a nation's capacity to succeed in an ever-changing and more competitive global marketplace.

7.1.1 Types of Entrepreneurships

Entrepreneurship is classified into the following types:

Small Business Entrepreneurship

These businesses include hairdresser, grocery store, travel agent, consultant, carpenter, plumber and electrician etc. People own their own business and hire family members or local employees. The profit earned can feed their family. It does not include making 100 million business or taking over an industry.

Scalable Start-up Entrepreneurship

This start-up entrepreneurship includes starting a business knowing that their vision can change the world. They attract investors who think and encourage people who think out of the box. The research focuses on a scalable business and experimental models. They require more venture capital to fuel and back their project or business.

Large Company Entrepreneurship

These huge companies have a defined life cycle. Most of these companies grow and sustain by offering new and innovative products that revolve around their main products. The change in technology, customer preferences, new competition etc., build pressure for large companies to create an innovative product and sell it to the new set of customers in the new market.

Social Entrepreneurship

This type of entrepreneurship focuses on producing products and services that resolve social needs and problems. This is mostly a non-profitable entrepreneurship.

7.1.2 Characteristics of Entrepreneurship

Every entrepreneur is not successful. There are definite characteristics that make entrepreneurs successful. A few of them are mentioned below:

- **Ability to take a risk:** Starting any new venture involves a considerable amount of failure risk. Therefore, an entrepreneur needs to be courageous and able to evaluate and take risks.
- **Innovation:** It should be highly innovative to generate new ideas, start a company and earn profits out of it. For instance, the launching of a product that is new to the market or a process that does the same thing but in a more efficient way.
- **Visionary and Leadership quality:** To be successful, an entrepreneur should have a clear vision of his new venture. However, to turn the idea into reality, several resources and employees are required.
- **Open-Minded:** Every circumstance in a business can be an opportunity and used for the benefit of a company. For example, Paytm recognised the gravity of demonetization and acknowledged the need for online transactions.
- **Flexible:** An entrepreneur should be flexible and open to change according to the situation.
- **Know your Product:** A company owner should know the product offerings and be aware of the latest trends in the market.

7.2 Feasibility and Business Plan

A feasibility study is done before starting a business, when you have the idea for the business, but you want to make sure it is feasible, or advisable. A business plan details how the business will operate. It assumes your feasibility study has been completed and it was determined that the idea is viable. Now you are going to spell out your financial and other objectives, the methods you plan to use to achieve them, and your proposed organizational structure.

7.2.1 Conducting a Feasibility Study

A complete competitive analysis should be conducted considering the following:

- **Product demand:** Is there a need or want for your product or service? Is the need already being met, or is there room for another product?
- **Market conditions:** Who would buy your product and where are they? Is there room/need for more products?

- **Pricing:** What do current users pay for similar products? What do you need to charge so that you will be profitable, and will consumers pay your price?
- **Risks:** What are the risks associated with your idea?
- **Probability of success:** Can you overcome risks to become profitable?

Writing a Business Plan

Business plans should include the following (as mentioned in Figure 7.1):

- **Executive summary:** Include your mission statement, products and or services, some brief information about your leadership team and key employees, as well as the location of your business.
- **Company description:** Detail the problems your business solves; its target market; its competitive advantages, compared with the competition, and anything else that makes your company superior to other.
- **Market analysis:** Perform competitive research of what other businesses are doing; their strengths and weaknesses, and how and why your business will be competitive and successful in the market.
- **Organization or management:** State the legal status of your business, such as a corporation or partnership, and include an organizational chart showing management levels and departments.
- **Service or product line:** State what you will sell or provide and describe the benefits of each.
- **Marketing and sales:** Explain in detail your marketing strategy.
- **Funding request:** If necessary, detail the amount of funding you will need for the next five years.
- **Financial projections:** This is the business's financial outlook for the next five years.

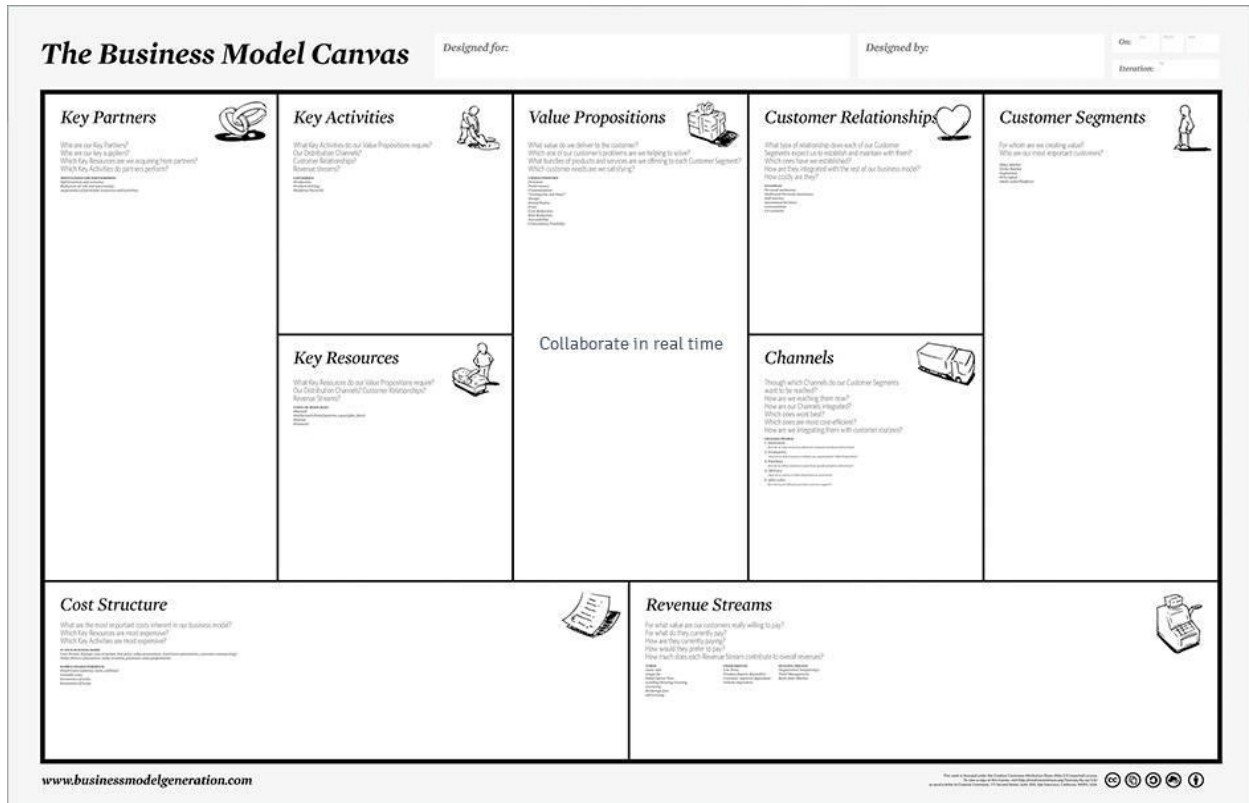


Fig. 7.1 Business Model Canvas

7.3 Marketing Mix

Marketing mix refers to the set of actions, or tactics, that a company uses to promote its brand or product in the market. The 4Ps make up a typical marketing mix - Price, Product, Promotion and Place. However, nowadays, the marketing mix increasingly includes several other Ps like Packaging, Positioning, People and even Politics.

Four Ps of Marketing

Price: It refers to the value that is being set for a product. It depends on costs of production, segment targeted, ability of the market to pay, supply - demand and a host of other direct and indirect factors.

Product: It refers to the item being sold. The product must deliver a minimum level of performance.

Place: It refers to the point of sale. In every industry, catching the consumers and making it easy for them to buy it is the main aim of a good distribution or 'place' strategy. Retailers pay a premium for the right location.

Promotion: It refers to all the activities undertaken to make the product or service known to the user and trade. This can include advertising, word of mouth, press reports, incentives, commissions, and awards to the trade.



Fig. 7.2 Four Ps of Marketing

7.4 Pricing and Costing

The price you charge your clients have a direct effect on the success of your business. The basic rules of pricing are the following:

- All prices must cover costs and profits.
- The most effective way to lower prices is to lower costs.
- Review prices frequently to assure that they reflect the dynamics of cost, market demand and profit objectives.
- Prices must be established to assure sales.

Before setting a price for your product, you must know the costs of running your business. If the price for your product or service does not cover costs, your cash flow will be cumulatively negative, you'll exhaust your financial resources, and your business will ultimately fail. To determine how much it costs to run your business, include property and/or equipment leases, loan repayments, inventory, utilities, financing costs, and salaries/wages/commissions. The costs of markdowns, shortages, employee discounts and cost of goods sold should also be included. Most important is to add profit in your calculation of costs.



Fig. 7.3 Four Ps of Marketing

Teacher Notes: Demonstrate students how to fill a business model canvas.

Key Points

- Entrepreneurship is the ability and readiness to develop, organize and run a business enterprise, along with any of its uncertainties to make a profit.
- A feasibility study is done before starting a business, when you have the idea for the business but you want to make sure it's feasible, or advisable.
- A business plan details how the business will operate.
- It assumes your feasibility study has been completed and it was determined the idea is viable.
- The marketing mix refers to the set of actions, or tactics, that a company uses to promote its brand or product in the market.
- The 4Ps make up a typical marketing mix - Price, Product, Promotion and Place.
- Price refers to the value that is put for a product.
- Product refers to the item actually being sold.
- Place refers to the point of sale.
- Promotion refers to all the activities undertaken to make the product or service known to the user and trade.
- All prices must cover costs and profits.

Before setting a price for your product, you must know the costs of running your business.

Exercise

Choose the most suitable option

1. Which of the following is a component of a business model?
a. Key partners b. Key activities c. Value Propositions d. All of these
2. There are _____ P's of marketing?
a. 3 b. 4 c. 5 d. None of these
3. Following are the P's of Marketing:
a. Product b. Pricing c. Promotion d. All of these
4. _____ refers to the set of actions that a company uses to promote its brand.
a. Marketing mix b. Promotion c. Marketing strategy d. All of these
5. _____ refers to all the activities undertaken to make the product known to the user.
a. Marketing mix b. Promotion c. Marketing strategy d. All of these

Give short answer to the following questions.

1. Define Entrepreneurship.
2. What is the difference between Feasibility plan and Business plan?
3. Define Marketing Mix.
4. What do you mean by pricing and costing.

Answer the following question in detail.

1. Describe different types of entrepreneurship.
2. Describe components of a business plan.
3. Describe 4 Ps of marketing.
4. What are the main characteristics of entrepreneurship?
5. What do you mean by a business model?

GLOSSARY

AT Commands: AT commands are instructions used to control a modem.

Auction: A public sale in which goods or property are sold to the highest bidder.

Bandwidth: The maximum amount of data transmitted over an internet connection in a given amount of time.

Baud rate: The baud rate is the rate at which information is transferred in a communication channel.

Bid: A bid is an offer made by an investor, trader, or dealer in an effort to buy an asset or to compete for a contract.

Breadboard: A breadboard, or protoboard, is a construction base for prototyping of electronics.

Clock Signal: a particular type of signal that oscillates between a high and a low state.

COM Port: A COM port is simply an I/O interface that enables the connection of a serial device to a computer.

Counters: Counters are used in digital electronics for counting purpose, they can count specific event happening in the circuit.

Demonetization: Demonetization is a form of economic intervention, where a country moves to replace one form of currency with another.

Electrode: An electrode is a solid electric conductor that carries electric current into non-metallic solids, or liquids, or gases, or plasmas, or vacuums.

Excitation Voltage: An external source of power.

Gig-economy: A gig economy is a free market system in which temporary positions are common and organizations hire independent workers for short-term commitments.

Ground: In electrical engineering, ground or earth is a reference point in an electrical circuit from which voltages are measured, a common return path for electric current.

Hex: Hexadecimal (or hex) is a base 16 system used to simplify how binary is represented.

IC: An integrated circuit or monolithic integrated circuit is a set of electronic circuits on one small flat piece (or "chip") of semiconductor material.

Life-cycle: The product life cycle is the process a product goes through from when it is first introduced into the market until it declines or is removed.

LSB: In computing, the least significant bit (LSB) is the bit position in a binary integer giving the units value, that is, determining whether the number is even or odd.

MSB: In a binary number, the bit furthest to the left is called the most significant bit (MSB).

NIC: A network interface card (NIC) is a hardware component without which a computer cannot be connected over a network.

Niche: A specialized segment of the market for a particular kind of product or service.

Potentiometer: A potentiometer is a three-terminal resistor with a sliding or rotating contact that forms an adjustable voltage divider.

Pulse: A pulse in signal processing is a rapid, transient change in the amplitude of a signal from a baseline value to a higher or lower value, followed by a rapid return to the baseline value.

PWM: Pulse-width modulation (PWM), or pulse-duration modulation (PDM), is a method of reducing the average power delivered by an electrical signal, by effectively chopping it up into discrete parts.

Scalability: Scalability is the property of a system to handle a growing amount of work by adding resources to the system.

Scoop: A utensil resembling a spoon, with a short handle and a deep bowl, used for removing dry or semi-solid substances from a container.

Serial Monitor: The Serial Monitor is an essential tool when creating projects with Arduino. It can be used as a debugging tool, testing out concepts or to communicate directly with the Arduino board.

Shipping: The transport of goods by sea or some other means.

Startup: A startup is a company that's in the initial stages of business.

Substrate: In electronics, a wafer (also called a slice or substrate) is a thin slice of semiconductor, such as a crystalline silicon (c-Si), used for the fabrication of integrated circuits and, in photovoltaics, to manufacture solar cells.

Sustainable: A sustainable business, or a green business, is an enterprise that has minimal negative impact or potentially a positive effect on the global or local environment, community, society, or economy—a business that strives to meet the triple bottom line.

Transducer: A transducer is a device that converts energy from one form to another.

Ultrasonic waves: Ultrasonic wave is defined as “inaudible sound with high frequency for human” the frequency of which generally exceeds 20 kHz.

Vcc: Vcc (Voltage Common Collector) is the higher voltage with respect to GND (ground).

Voltage Divider: A voltage divider is a simple circuit which turns a large voltage into a smaller one.

ABOUT THE AUTHOR



Muhammad Umair is a researcher at the *Sensors, Cloud and Services (SCS) Lab*, School of Computer Sciences, The University of Sydney, Australia. Muhammad Umair is also a Lecturer at the Department of Electrical, Electronics and Telecommunication Engineering, New Campus, UET Lahore. He completed his B.Sc. Electrical Engineering and M.Sc. Electrical Engineering from University of Engineering & Technology (UET) Lahore in 2014 and 2017, respectively. He has worked as a Research Officer at Internet of Things (IoT) lab at Al-Khwarizmi Institute of Computer Sciences, UET Lahore. He has also worked at Sultan Qaboos IT Research lab as a Research Officer. His survey on Social IoT platforms is the most cited survey for SIoT applications. He has designed graduate level courses on IoT.

Acknowledgements:

Special thanks to Dr. Faheem Gohar Awan for his guidance to write this book.

قومی ترانہ

پاک سر زمین شاد باد! کشورِ حسین شاد باد!
تو نشانِ عزمِ عالی شان ارضِ پاکستان
مرکزِ یقینِ شاد باد!

پاک سر زمین کا نظام قوتِ اخوتِ عوام
قوم، ملک، سلطنت پائندہ تابندہ باد!
شاد باد منزلِ مراد!

پرچمِ ستارہ و ہلال رہبرِ ترقی و کمال
ترجمانِ ماضی، شانِ حال جانِ استقبال
سایہ خدائے ذوالجلال!



National Vocational & Technical Training Commission (NAVTTC)

Plot No.38, Sector H-9/4, Kirthar Road, Islamabad.

Tel: +92-51-9207518

Website: www.navttc.gov.pk