



## JEPPIAAR INSTITUTE OF TECHNOLOGY

(An Autonomous Institution)

Self-Belief | Self Discipline | Self Respect

Kunnam, Sunguvachatram, Sriperumbudur-631604



## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### CS3691– EMBEDDED AND IoT LABORATORY

**NAME** :

**REG NO.** :

**YEAR** :

**SEMESTER** :

**BRANCH** :

# **JEPPIAAR INSTITUTE OF TECHNOLOGY**

**SELF BELIEF | SELF DISCIPLINE | SELF RESPECT**

**KUNNAM, SUNGUVARCHATRAM, SRIPERUMBUDUR, CHENNAI - 631 604.**



## **BONAFIDE CERTIFICATE**

This is a certified Bonafide Record Work of Mr./Ms. \_\_\_\_\_

Register No. \_\_\_\_\_ submitted for the Anna University Practical Examination held on \_\_\_\_\_ in **CS3691 – EMBEDDED AND IoT LABORATORY** during the year 2025-2026.

Signature of the Lab In-charge

Head of the Department

**Internal Examiner**

**External Examiner**

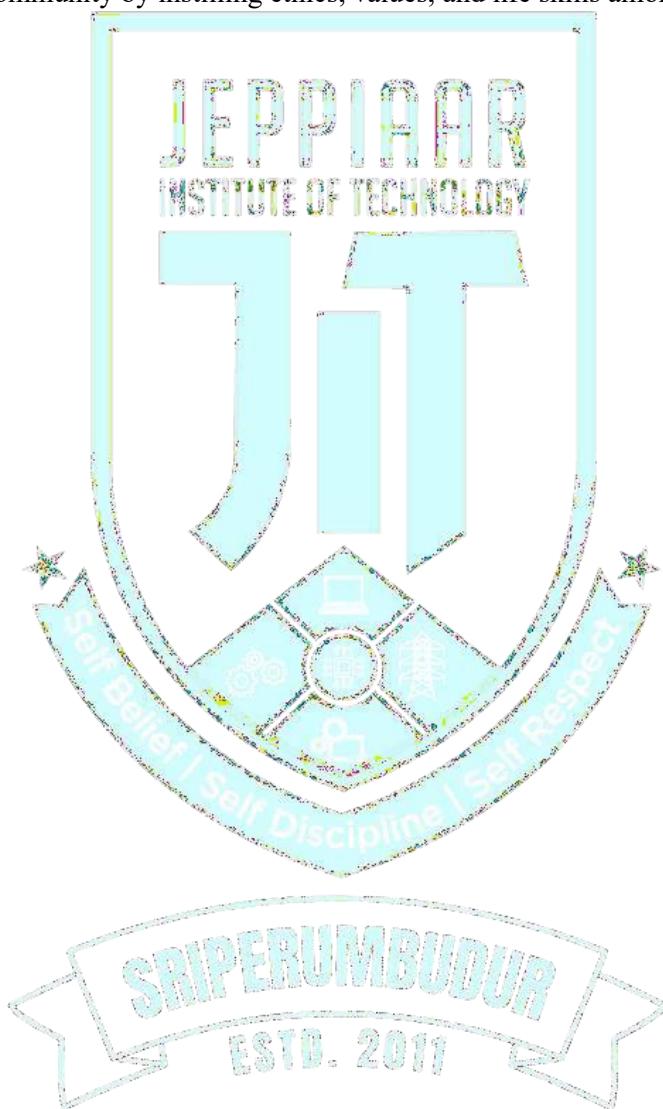
Date \_\_\_\_\_

## **VISION OF THE INSTITUTION**

Jeppiaar Institute of Technology aspires to provide technical education in futuristic technologies with the perspective of innovative, industrial, and social applications for the betterment of humanity.

## **MISSION OF THE INSTITUTION**

- To produce competent and disciplined high-quality professionals with the practical skills necessary to excel as innovative professionals and entrepreneurs for the benefit of society.
- To improve the quality of education through excellence in teaching and learning, research, leadership, and by promoting the principles of scientific analysis, and creative thinking.
- To provide excellent infrastructure, serene, and stimulating environment that is most conducive to learning.
- To strive for productive partnership between the Industry and the Institute for research and development in the emerging fields and creating opportunities for employability.
- To serve the global community by instilling ethics, values, and life skills among the students needed to enrich their lives.



## **VISION OF THE DEPARTMENT**

To impart futuristic technological education, innovation and collaborative research in the field of Computer Science Engineering and develop Quality Professional for the improvement of the society and industry.

## **MISSION OF THE DEPARTMENT**

**M 1:** Devise students as Professionally competent and disciplined engineers for the benefit of the country's development.

**M 2:** Produce excellent to adopt latest technologies, industry-institute interaction and encouraging research activities.

**M 3:** Provide multidisciplinary technical skills to pursue research activities, higher studies, entrepreneurship and perpetual learning.

**M 4:** Enrich students with professional integrity and ethical standards to handle social challenges successfully in their life.

## **PEO's OF THE DEPARTMENT**

- To support students with substantial knowledge for developing and resolving mathematical, scientific and engineering problems.
- To provide students with adequate training and opportunities to work as a collaborator with informative and administrative qualities.
- To motivate students for extensive learning to prepare them for graduate studies, R&D and competitive exams.
- To cater students with industrial exposure in an endeavour to succeed in the emerging cutting-edge technologies.
- To shape students with principled values and to follow the code of ethics in social and professional life.

## **KNOWLEDGE AND ATTITUDE PROFILE(WK)**

**WK1.** A systematic, theory-based understanding of the natural sciences applicable to the discipline and awareness of relevant social sciences.

**WK2.** Conceptually-based mathematics, numerical analysis, data analysis, statistics and formal aspects of computer and information science to support detailed analysis and modelling applicable to the discipline.

**WK3.** A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline.

**WK4.** Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline; much is at the forefront of the discipline.

**WK5.** Knowledge including efficient resource use, environmental impacts, whole-life cost, re-use of resources, net zero carbon, and similar concepts, that supports engineering design and operations in a practice area.

**WK6.** Knowledge of engineering practice (technology) in the practice areas in the engineering discipline.

**WK7.** Knowledge of the role of engineering in society and identified issues in engineering practice in the discipline, such as the professional responsibility of an engineer to public safety and sustainable development.

**WK8.** Engagement with selected knowledge in the current research literature of the discipline, awareness of the power of critical thinking and creative approaches to evaluate emerging issues.

**WK9.** Ethics, inclusive behavior and conduct. Knowledge of professional ethics, responsibilities, and norms of engineering practices. Awareness of the need for diversity by reason of ethnicity, gender, age, physical ability etc. with mutual understanding and respect, and of inclusive attitudes.

## **PROGRAM OUTCOMES (PO'S):**

Engineering Graduates will be able to:

- 1. Engineering knowledge:** (K3) Apply the knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.
- 2. Problem analysis:** (K4) Identify, formulate, review research literature, and analyze complex engineering problem reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)
- 3. Design/development of Solutions:** (K4) Design creative solutions for complex engineering problems and design/develop system/components/processes to meet the specified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society, and environment as required. (WK5)
- 4. Conduct investigations of Complex problems:** (K5) Conduct investigations of Complex engineering problems using research-based knowledge including design of experiments, modelling, analysis and interpretation of data to provide valid conclusions. (WK8)
- 5. Engineering Tool Usage:** (K3, K5, K6) Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling recognizing their limitations to solve complex engineering problems.
- 6. The Engineer and The World:** (A3) Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5 and WK7).
- 7. Ethics:** (A3) Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national and international laws. (WK9)
- 8. Individual and Collaborative team work:** (A3) Function effectively as an individual, and as a member or leader in diverse/multidisciplinary teams.
- 9. Communication:** (A3) Communicate effectively and inclusively within the engineering community and society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, considering cultural, language, and learning differences.
- 10. Project management and finance:** (A3) Apply knowledge and understanding of the engineering and management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multi-disciplinary environments.
- 11. Life-long learning:** (A2) Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new emerging technologies and iii) critical

thinking in the broadest context of technological change.(WK8)

## **PROGRAM SPECIFIC OUTCOMES**

Engineering Graduates will be able to:

**PSO 1:** Analyse, design, and implement quality software by applying fundamental and programming concepts of Computer Science and Engineering

**PSO 2:** Design and develop solutions for scientific, business and real time applications through analytical, logical and problems solving skills.

**PSO 3:** Provide efficient solutions for industrial and society needs with acquired knowledge through emerging technical skills.

## **LIST OF EXPERIMENTS**

1. Write 8051 Assembly Language experiments using simulator.
2. Test data transfer between registers and memory.
3. Perform ALU operations.
4. Write Basic and arithmetic Programs Using Embedded C.
5. Introduction to Arduino platform and programming
6. Explore different communication methods with IOT devices (Zigbee, GSM, Bluetooth)
7. Introduction to Raspberry PI platform and python programming
8. Interfacing sensors with Raspberry PI
9. Communicate between Arduino and Raspberry PI using any wireless medium
10. Setup a cloud platform to log the data
11. Log Data using Raspberry PI and upload to the cloud platform
12. Design an IOT based system

## TABLE OF CONTENTS


## **INTRODUCTION**

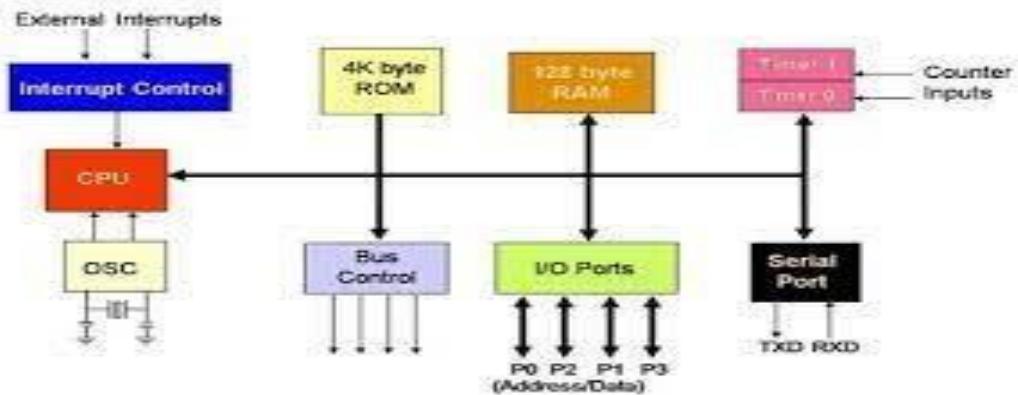
Earlier to Microcontrollers, Microprocessors were greatly used for each and every purpose. Microprocessors were containing ALU, general purpose register, stackpointer, program counter, clock counter and so many other features which the today's Microcontroller also possesses. But the difference between them exists with respect to the number of instructions, access times, size, reliability, PCB size and so on. Microprocessor contains large instruction set called as CISC processor whereas Microcontroller contains less number of instructions and is called as RISC processor. The access time is less in case of microcontrollers compared to microprocessors and the PCB size reduces in case of microcontrollers.

There are many versions of microcontrollers 8051, 8052, 8751, AT8951 from Atmel Corporation and many more. In this manual we will study about the 8051 architectures, its features, programming and interfacing.

MCS 8051 is an 8-bit single chip microcontroller with many built-in functions and is the core for all MCS-51 devices.

The main features of the 8051 cores are:

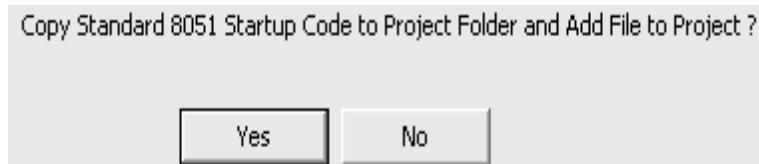
- ♣ Operates with single Power Supply +5V.
- ♣ 8-bit CPU optimized for control applications.
- ♣ 16-bit program counter (PC) and 16-bit data pointer (DPTR).
- ♣ 8-bit program status word (PSW).
- ♣ 8-bit stack pointer (SP).
- ♣ 4K Bytes of On-Chip Program Memory (Internal ROM or EPROM).
- ♣ 128 bytes of On-Chip Data Memory (Internal RAM):
  - ◆ Four Register Banks, each containing 8 registers (R0 to R7) [Total 32 reg]
  - ◆ 16-bytes of bit addressable memory.
  - ◆ 80 bytes of general-purpose data memory (Scratch Pad Area).
- ♣ Special Function Registers (SFR) to configure/operate microcontroller.
- ♣ 32 bit bi-directional I/O Lines (4 ports P0 to P3).
- ♣ Two 16-bit timers/counters (T0 and T1).
- ♣ Full duplex UART (Universal Asynchronous Receiver/Transmitter).
- ♣ On-Chip oscillator and clock circuitry.



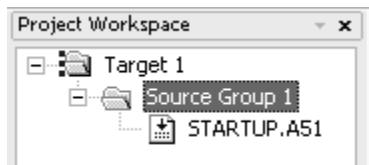
### STEPS TO CREATE AND COMPILE Keil µVision-3/4 PROJECT:



1. Double Click on the icon on the desktop.
2. Close any previous projects that were opened using – **Project -> Close**.
3. Start **Project – New Project**, and select the CPU from the device database (Database-Atmel-AT89C51ED2 or AT89C51RD2 as per the board). On clicking ‘OK’, the following option is displayed. Choose ‘No’.



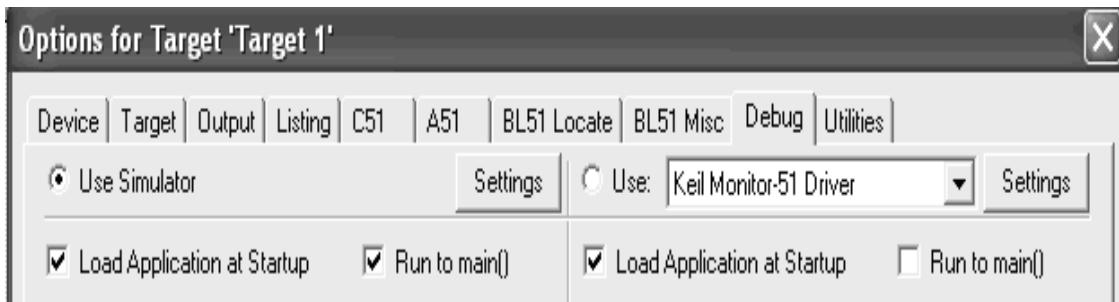
4. Create a source file (using **File->New**), type in the assembly or C program and save this (filename.asm/filename.c) and add this source file to the project using either one of the following two methods. (i) **Project->Manage->Components, Environment Books->addfiles-> browse to the required file -> OK**  
“OR” ii) right click on the Source Group in the Project Window and the **Add Files to Group** option.



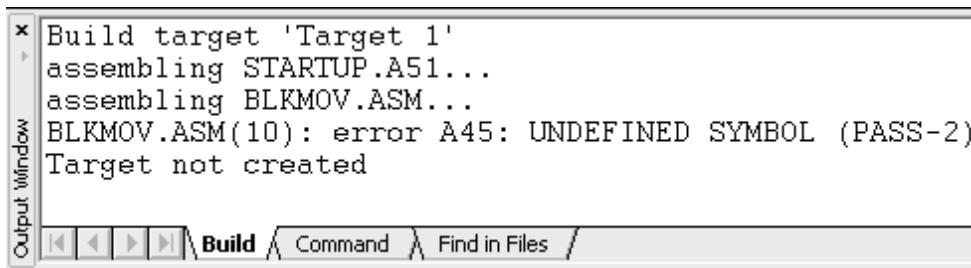
5. Set the Target options using –> **Project – Options for Target** opens the **Options for Target – Target** configuration dialog. Set the Xtal(Crystal frequency) frequency

as 11.0592 MHz, and also the **Options for Target**

- **Debug – use either Simulator / Keil Monitor- 51 driver.**



6. If **Keil Monitor- 51 driver** is used click on **Settings** -> COM Port settings select the COM Port to which the board is connected and select the baud rate as 19200 or 9600 (recommended). Enable **Serial Interrupt** option if the user application is not using on-chip UART, to stop program execution.
7. Build the project; using **Project -> Build Project**. μVision translates all the user application and links. Any errors in the code are indicated by – “Target not created” in the Build window, along with the error line. Debug the errors. After an error free, to build go to Debug mode.



8. Now user can enter into **Debug** mode with **Debug- Start / Stop Debug session** dialog. Or by clicking in the icon.
9. The program is run using the **Debug-Run** command & halted using **Debug-Stop Running**.

Also the (reset, run, halt) icons can be used. Additional icons are .

10. If it is an interface program the outputs can be seen on the LCD, CRO, motor, led status, etc. If it is a part-A program, the appropriate memory window is opened using View -> memory window (for data RAM & XRAM locations), Watch window (for timer program), serial window, etc.

11. **Note:** To access data RAM area type address as D: 0020h. Similarly to access the DPTR region (XRAM-present on chip in AT89C51ED2) say 9000h location type in X: 09000H.

Ex no:

## ARITHMETIC OPERATIONS USING SIMULATOR

Date:

### AIM:

To write a program for performing addition operations using 8051microcontroller using keil simulator.

### APPARATUS REQUIRED:

1. Computer
2. Keil micro vision software 3/4

### ALGORITHM:

1. Open the keil micro vision 4 software
2. Select ne project and build the new project
3. Write the assembly program for addition operations
4. Debug the program

### PROGRAM :

```
Mov r0, #34h  
Mov r1, #12h  
Mov r2, #0dch  
Mov r3, #0feh  
Clr c  
Mov a, r0  
Add a, r2  
Mov 22h, a  
Mov a, r1  
Addc a, r3  
Mov 21h, a  
Mov ooh, c  
end
```

**RESULT :**

Thus program for performing addition operations using 8051microcontroller using keil simulator is successfully completed.

Ex no:

## **DATA TRANSFER BETWEEN REGISTERS AND MEMORY**

Date:

### **AIM:**

To Write an assembly language program to transfer the data from one location to another

### **APPARATUS REQUIRED:**

1. Computer
2. Keil micro vision software 3/4

### **ALGORITHM:**

1. Open – keil micro vision 5 – new micro vision project – once open the project, dialog box will be open – set the microcontroller AT89C51 and click ok – it ask whether STARTUP is ok – click yes – on a project window – target will be generated.
2. Once target is generated – go to new – file – design a name for it with extension.asm and click ok - right click on target – source group – right click –add existing files – ask file name – choose file name.asm – ok.
3. Do the program for data transfer in a work place. Once program is done – click save
4. GOTO – project – build target – on command window – error is acknowledged program is successfully compiler.

### **PROGRAM :**

```
Org 000h
Mov r0, #30h
Mov r1, #50h
Mov r3, #05h
BACK : mov a , @r0
Mov @r1 , a
Inc r0inc r1
Djnz r3, BACK
end
```

### Before Execution:

	Address: d30h
D:0x30:	01 02 03 04 05 00 00 00 00 00 00
D:0x3A:	00 00 00 00 00 00 01 02 03 04
D:0x44:	05 00 00 00 00 00 00 00 00 00 00
D:0x4E:	00 00 00 00 00 00 00 00 00 00 00
D:0x50:	00 00 00 00 00 00 00 00 00 00 00

### After Execution:

	Address: d30h
D:0x30:	01 02 03 04 05 00 00 00 00 00 00
D:0x3A:	00 00 00 00 00 00 01 02 03 04
D:0x44:	05 00 00 00 00 00 00 00 00 00 00
D:0x4E:	00 00 00 00 00 00 00 00 00 00 00
D:0x50:	00 00 00 00 00 00 00 00 00 00 00

### Result:

Thus the program for data transfer is verified and executed successfully.

## ARITHMETIC OPERATIONS

Ex no:

Date:

**AIM:**

To Write an assembly language program to perform addition operation

**APPARATUS REQUIRED:**

1. Computer
2. Keil micro vision software 3/4

**ALGORITHM:**

1. Open – keil micro vision 5 – new micro vision project – once open the project ,dialog box will be open – set the microcontroller AT89C51 and click ok – it ask whether STARTUP is ok – click yes – on a project window – target will be generated.
2. Once target is generated – go to new – file – design a name for it with extension.asm and click ok - right click on target – source group – right click –add existing files – ask file name – choose file name.asm – ok.
3. Do the program for data transfer in a work place. Once program is done – click save
4. GOTO – project – build target – on command window – error is acknowledged program is successfully compiler.

**PROGRAM :**

INSTRUCTIONS	COMMANDS
MOV A, #33H	Move the immediate data of 33h into accumulator A
MOV B, #11H	Move the immediate data of 11h into B
ADD A ,B	Add the values of B with values in accumulator.
END	Halt the program

**RESULT :**

Thus , the program for addition operation is verified and executed successfully

## ARITHMETIC OPERATIONS

Ex no:

Date:

**AIM:**

To Write an assembly language program to perform subtraction operation

**APPARATUS REQUIRED:**

1. Computer
2. Keil micro vision software 3/4

**ALGORITHM:**

1. Open – keil micro vision 5 – new micro vision project – once open the project, dialog box will be open – set the microcontroller AT89C51 and click ok – it ask whether STARTUP is ok – click yes – on a project window – target will be generated.
2. Once target is generated – go to new – file – design a name for it with extension.asm and click ok - right click on target – source group – right click –add existing files – ask file name – choose file name.asm – ok.
3. Do the program for data transfer in a work place. Once program is done – click save
4. GOTO – project – build target – on command window – error is acknowledged program is successfully compiler.

**PROGRAM :**

INSTRUCTIONS	COMMANDS
MOV A, #22H	Move the immediate data of 22h into accumulator A
MOV B, #11H	Move the immediate data of 11h into B
SBB A ,B	Subtract the values of B with values in accumulator.
END	Halt the program

**RESULT :**

Thus , the program for subtraction operation is verified and executed successfully

## ARITHMETIC OPERATIONS

Ex no:

Date:

**AIM:**

To Write an assembly language program to perform multiplication operation

**APPARATUS REQUIRED:**

1. Computer
2. Keil micro vision software 3/4

**ALGORITHM:**

1. Open – keil micro vision 5 – new micro vision project – once open the project, dialog box will be open – set the microcontroller AT89C51 and click ok – it ask whether STARTUP is ok – click yes – on a project window – target will be generated.
2. Once target is generated – go to new – file – design a name for it with extension.asm and click ok - right click on target – source group – right click –add existing files – ask file name – choose file name.asm – ok.
3. Do the program for data transfer in a work place. Once program is done – click save
4. GOTO – project – build target – on command window – error is acknowledged program is successfully compiler.

**PROGRAM :**

INSTRUCTIONS	COMMANDS
MOV A, #22H	Move the immediate data of 22h into accumulator A
MOV B, #11H	Move the immediate data of 11h into B
MUL A ,B	Multiply the values of B with values in accumulator.
END	Halt the program

**RESULT :**

Thus , the program for multiplication operation is verified and executed successfully

## ARITHMETIC OPERATIONS

Ex no:

Date:

**AIM:**

To Write an assembly language program to perform division operation

**APPARATUS REQUIRED:**

1. Computer
2. Keil micro vision software 3/4

**ALGORITHM:**

1. Open – keil micro vision 5 – new micro vision project – once open the project ,dialog box will be open – set the microcontroller AT89C51 and click ok – it ask whether STARTUP is ok – click yes – on a project window – target will be generated.
2. Once target is generated – go to new – file – design a name for it with extension.asm and click ok - right click on target – source group – right click –add existing files – ask file name – choose file name.asm – ok.
3. Do the program for data transfer in a work place. Once program is done – click save
4. GOTO – project – build target – on command window – error is acknowledged program is successfully compiler.

**PROGRAM :**

INSTRUCTIONS	COMMANDS
MOV A, #22H	Move the immediate data of 22h into accumulator A
MOV B, #11H	Move the immediate data of 11h into B
DIV A ,B	Multiply the values of B with values in accumulator.
END	Halt the program

**RESULT :**

Thus , the program for division operation is verified and executed successfully

## LOGICAL OPERATIONS

Ex no:

Date:

**AIM:**

To Write an assembly language program to perform AND logical operation

**APPARATUS REQUIRED:**

1. Computer
2. Keil micro vision software 3/4

**ALGORITHM:**

1. Open – keil micro vision 5 – new micro vision project – once open the project ,dialog box will be open – set the microcontroller AT89C51 and click ok – it ask whether STARTUP is ok – click yes – on a project window – target will be generated.
2. Once target is generated – go to new – file – design a name for it with extension.asm and click ok - right click on target – source group – right click –add existing files – ask file name – choose file name.asm – ok.
3. Do the program for data transfer in a work place. Once program is done – click save
4. GOTO – project – build target – on command window – error is acknowledged program is successfully compiler.

**PROGRAM :**

INSTRUCTIONS	COMMANDS
MOV A, #22H	Move the immediate data of 22h into accumulator A
MOV B, #11H	Move the immediate data of 11h into B
ANL A ,B	AND the values of B with values in accumulator.
MOV 46,A	Move the values of A into memory location 46
END	Halt the program

**RESULT :**

Thus , the program for AND operation is verified and executed successfully

## LOGICAL OPERATIONS

Ex no:

Date:

**AIM:**

To Write an assembly language program to perform OR logical operation

**APPARATUS REQUIRED:**

1. Computer
2. Keil micro vision software 3/4

**ALGORITHM:**

1. Open – keil micro vision 5 – new micro vision project – once open the project, dialog box will be open – set the microcontroller AT89C51 and click ok – it ask whether STARTUP is ok – click yes – on a project window – target will be generated.
2. Once target is generated – go to new – file – design a name for it with extension.asm and click ok - right click on target – source group – right click –add existing files – ask file name – choose file name.asm – ok.
3. Do the program for data transfer in a work place. Once program is done – click save
4. GOTO – project – build target – on command window – error is acknowledged program is successfully compiler.

**PROGRAM :**

INSTRUCTIONS	COMMANDS
MOV A, #22H	Move the immediate data of 22h into accumulator A
MOV B, #11H	Move the immediate data of 11h into B
ORLL A ,B	OR the values of B with values in accumulator.
MOV 47,A	Move the values of A into memory location 47
END	Halt the program

**RESULT :**

Thus , the program for OR operation is verified and executed successfully

## ARITHMETIC OPERATIONS USING EMBEDDED C

Ex no:

Date:

**AIM:**

To Write an assembly language program to perform addition operation using embedded c.

**APPARATUS REQUIRED:**

1. Computer
2. Keil micro vision software 3/4

**ALGORITHM:**

1. Open – keil micro vision 5 – new micro vision project – once open the project, dialog box will be open – set the microcontroller AT89C51 and click ok – it ask whether STARTUP is ok – click yes – on a project window – target will be generated.
2. Once target is generated – go to new – file – design a name for it with extension.asm and click ok - right click on target – source group – right click –add existing files – ask file name – choose file name.asm – ok.
3. Do the program for data transfer in a work place. Once program is done – click save
4. GOTO – project – build target – on command window – error is acknowledged program is successfully compiler.

**PROGRAM :**

```
#include <reg51.h>
Void main (void)
{
Unsigned char x,y,z;
X=0x34;
Y=0x99;
P1 = 0x00;
Z=x+y;
P1 = z;
}
```

**RESULT :**

Thus, the program for addition operation using embedded c is verified and executed successfully

## SUBTRACTION

Ex no:

Date:

### AIM:

To Write an assembly language program to perform subtraction operation using embedded c.

### APPARATUS REQUIRED:

1. Computer
2. Keil micro vision software 3/4

### ALGORITHM:

1. Open – keil micro vision 5 – new micro vision project – once open the project, dialog box will be open – set the microcontroller AT89C51 and click ok – it ask whether STARTUP is ok – click yes – on a project window – target will be generated.
2. Once target is generated – go to new – file – design a name for it with extension.asm and click ok - right click on target – source group – right click –add existing files – ask file name – choose file name.asm – ok.
3. Do the program for data transfer in a work place. Once program is done – click save
4. GOTO – project – build target – on command window – error is acknowledged program is successfully compiler.

### PROGRAM :

```
#include <reg51.h>
Void main (void)
{
Unsigned char x,y,z;
X=0x34;
Y=0x99;
P1 = 0x00;
Z=x - y;
P1 = z;
}
```

**RESULT :**

Thus , the program for subtraction operation using embedded c is verified and executed successfully

## DIVISION

Ex no:

Date:

**AIM:**

To Write an assembly language program to perform division operation using embedded c.

**APPARATUS REQUIRED:**

1. Computer
2. Keil micro vision software 3/4

**ALGORITHM:**

1. Open – keil micro vision 5 – new micro vision project – once open the project, dialog box will be open – set the microcontroller AT89C51 and click ok – it ask whether STARTUP is ok – click yes – on a project window – target will be generated.
2. Once target is generated – go to new – file – design a name for it with extension.asm and click ok - right click on target – source group – right click –add existing files – ask file name – choose file name.asm – ok.
3. Do the program for data transfer in a work place. Once program is done – click save
4. GOTO – project – build target – on command window – error is acknowledged program is successfully compiler.

**PROGRAM :**

```
#include <reg51.h>
Void main (void)
{
Unsigned char x,y,z;
X=0x34;
Y=0x99;
P1 = 0x00;
Z=x/y;
P1 = z;
}
```

**RESULT :**

Thus , the program for division operation using embedded c is verified and executed successfully

## MULTIPLICATION

Ex no:

Date:

### AIM:

To Write an assembly language program to perform multiplication operation using embedded c.

### APPARATUS REQUIRED:

1. Computer
2. Keil micro vision software 3/4

### ALGORITHM:

1. Open – keil micro vision 5 – new micro vision project – once open the project ,dialog box will be open – set the microcontroller AT89C51 and click ok – it ask whether STARTUP is ok – click yes – on a project window – target will be generated.
2. Once target is generated – go to new – file – design a name for it with extension.asm and click ok - right click on target – source group – right click –add existing files – ask file name – choose file name.asm – ok.
3. Do the program for data transfer in a work place. Once program is done – click save
4. GOTO – project – build target – on command window – error is acknowledged program is successfully compiler.

### PROGRAM :

```
#include <reg51.h>
Void main (void)
{
Unsigned char x,y,z;
X=0x34;
Y=0x99;
P1 = 0x00;
Z=x*y;
P1 = z;
}
```

**RESULT :**

Thus , the program for multiplication operation using embedded c is verified and executed successfully

**Ex No:**

## INTRODUCTION ARDUINO PROGRAMMING ENVIRONMENT (IDE)

**Date :**

### AIM:

To study the basics of Arduino Uno board and Arduino IDE 2.0 software.

### Hardware & SOFTWARE TOOLS REQUIRED:

S.No.	Hardware & Software Requirements	Quantity
1	Arduino IDE 2.0	1
2	Arduino Uno Board	1

### INTRODUCTION TO ARDUINO:

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

The screenshot shows the official Arduino website's download section. At the top, there's a navigation bar with links for HOME, BUY, SOFTWARE (which is highlighted in blue), PRODUCTS, LEARNING, FORUM, SUPPORT, and BLOG. To the right of the navigation are a search icon, a user icon, and a SIGN IN link. Below the navigation, the text "Download the Arduino IDE" is centered. On the left, there's a large circular logo featuring a white infinity symbol with a minus sign on the left and a plus sign on the right, all set against a teal background. To the right of the logo, the text "ARDUINO 1.8.2" is displayed, followed by a detailed description of the software: "The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions." On the far right, there's a teal sidebar with links for "Windows Installer", "Windows ZIP file for non admin install", "Windows app" (with a Get button), "Mac OS X 10.7 Lion or newer", "Linux 32 bits", "Linux 64 bits", "Linux ARM", and links for "Release Notes", "Source Code", and "Checksums (sha512)". At the bottom of the page, there are two sections: "ARDUINO SOFTWARE HOURLY BUILDS" on the left and "ARDUINO 1.0.6 / 1.5.x / 1.6.x PREVIOUS RELEASES" on the right, each with their respective download links.

## Software-Installation

- Windows  
[arduino.cc/windows](http://arduino.cc/windows)  
Installation for: Windows 7, Vista, e XP
- Mac OS X [arduino.cc/mac](http://arduino.cc/mac)  
Installation for: OS X 10.7 or newer
- Linux  
[arduino.cc/linux](http://arduino.cc/linux)

## Communication with Arduino

- Launch the Arduino IDE (double click)

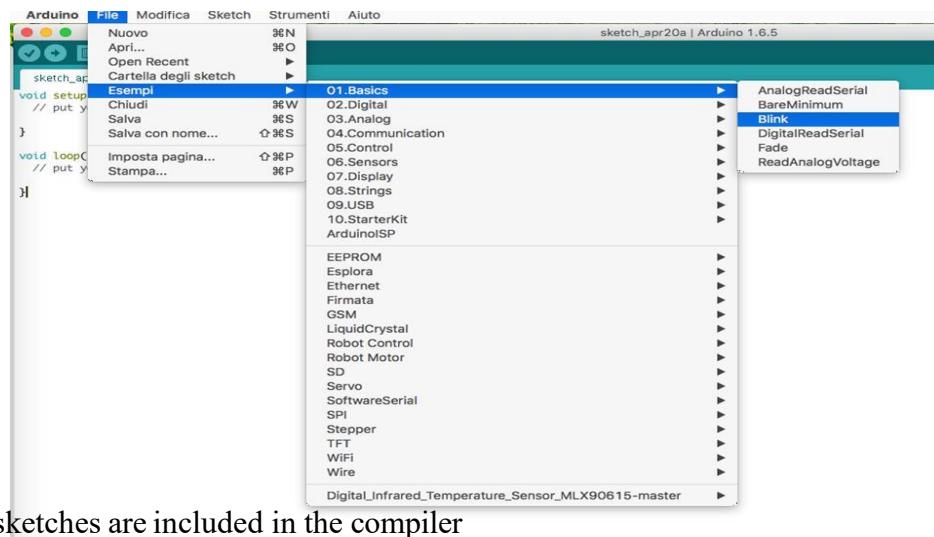


## Arduino Program Development

- Based on C++ without 80% of the instructions.
- A handful of new commands.
- Programs are called 'sketches'.
- Sketches need two functions:

- void setup()
- void loop()
- setup() runs first and once.
- loop() runs over and over, until power is lost or a new sketch is loaded.

Open the sketch



Numerous sample sketches are included in the compiler

- Located under File, Examples
- Once a sketch is written, it is uploaded by clicking on File, Upload, or by pressing <Ctrl> U

Open the Blink sketch

```

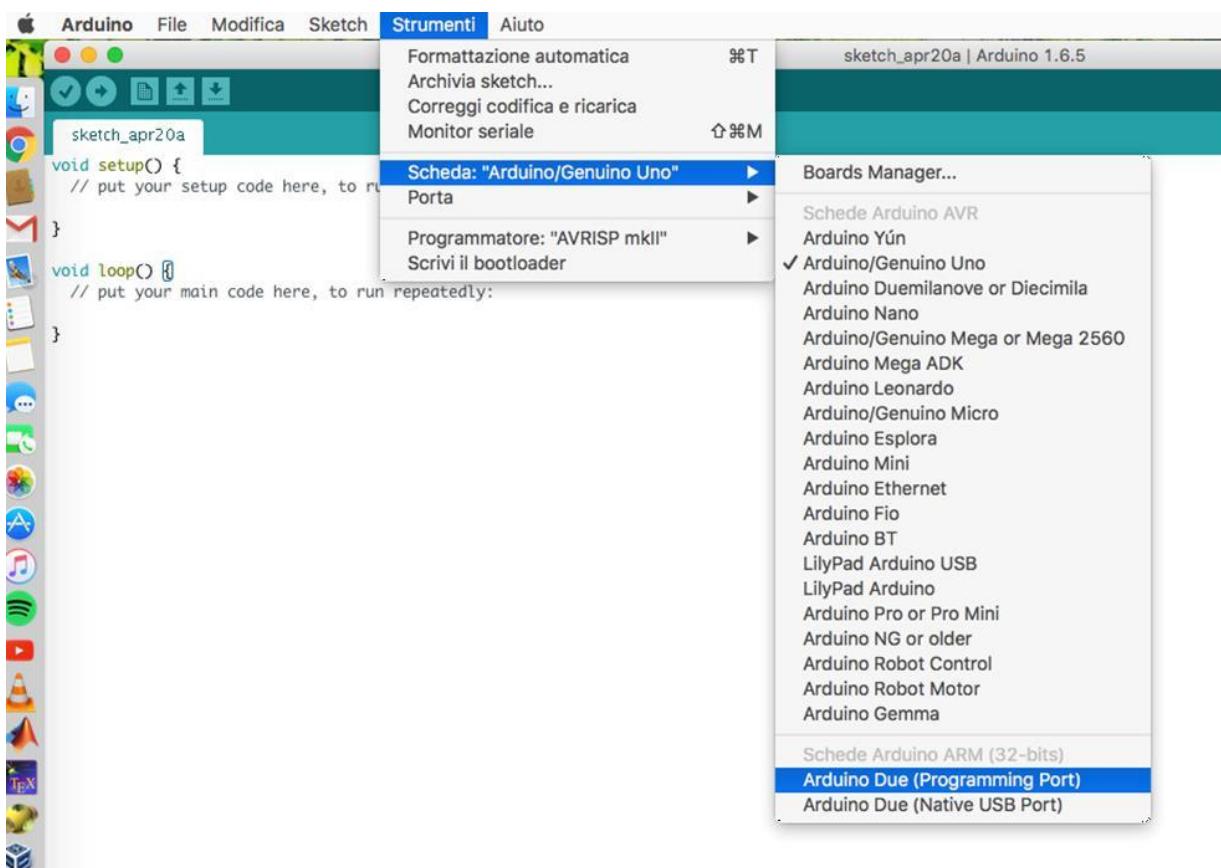
 00 Blink | Arduino 1.0.1
 File Edit Sketch Tools Help
 Blink
 Blink
 Turns on an LED on for one second, then off for one second, repeatedly.
 This example code is in the public domain.
 */
 // Pin 13 has an LED connected on most Arduino boards.
 // give it a name:
 int led = 13;
 // the setup routine runs once when you press reset:
 void setup() {
 // initialize the digital pin as an output:
 pinMode(led, OUTPUT);
 }
 // the loop routine runs over and over again forever:
 void loop() {
 digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
 delay(1000); // wait for a second
 digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
 delay(1000); // wait for a second
 }

 Save Canceled.

```

The screenshot shows the Arduino IDE with the 'Blink' sketch open in the code editor. The code is a standard 'Blink' example for an Arduino board, which turns an LED on and off at a 1-second interval. The Arduino Uno is connected to COM7.

## Select the Board



## Select the Serial Port



- **Mac:**

You can indifferently choose between /dev/tty.usbmodemXXXXXX or /dev/cu.usbmodemXXXXXX

- **Windows:**

- There are one or more COM ports:
  - choose the one with the higher number if it does not work try with the other proposals
- The connection to the serial port is reported in the code window in bottom right

```
Blink | Arduino 1.0.5

Upload

Blink
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
*/

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);     // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}

14 Arduino Uno on /dev/tty.usbmodem24161
```

## RESULT:

**EX NO:**

**DATE :**

## **EXPLORE DIFFERENT COMMUNICATION METHODS WITH IOT DEVICES (ZIGBEE, GSM, BLUETOOTH)**

### Wi-Fi

Wi-Fi (Wireless Fidelity) is the most popular **IOT communication protocols** for wireless local area network (WLAN) that utilizes the IEEE 802.11 standard through 2.4 GHz UHF and 5 GHz ISM frequencies. Wi-Fi provides Internet access to devices that are within the range of about 20 - 40 meters from the source. It has a data rate upto 600 Mbps maximum, depending on channel frequency used and the number of antennas. In embedded systems, ESP series controllers from Espressif are popular for building IoT based Applications. **ESP32** and **ESP8266** are the most commonly use wifi modules for embedded applications. You can find various projects based on ESP32 and ESP8266 by following the link.

In terms of using the Wi-Fi protocol for IOT, there are some pros & cons to be considered. The infrastructure or device cost for Wi-Fi is low & deployment is easy but the power consumption is high and the Wi-Fi range is quite moderate. So, the Wi-Fi may not be the best choice for all types of IOT applications but it can be used for applications like Home Automation.

There are many development boards available that allow people to build IOT applications using Wi-Fi. The most popular ones are the Raspberry Pi and Node MCU. These boards allow people to build IOT prototypes and also can be used for small real-time applications. Likewise is the Marvell Avastar 88W8997 SoC, which follows the Wi-Fi's IEEE 802.11n standard. The chip has applications like wearables, wireless audio & smart home.

## Bluetooth

Bluetooth is a technology used for exchanging data wirelessly over short distances and preferred over various **IOT network protocols**. It uses short-wavelength UHF radio waves of frequency ranging from 2.4 to 2.485 GHz in the ISM band. The Bluetooth technology has 3 different versions based on its applications:

- **Bluetooth:** The Bluetooth that is used in devices for communication has many applications in IOT/M2M devices nowadays. It is a technology using which two devices can communicate and share data wirelessly. It operates at 2.4GHz ISM band and the data is split in packets before sending and then is shared using any one of the designated 79 channels operating at 1 MHz of bandwidth.
- **BLE (Bluetooth 4.0, Bluetooth Low Energy):** The BLE has a single main difference from Bluetooth that it consumes low power. With that, it makes the product of low cost & more long-lasting than Bluetooth.
- **iBeacon:** It is a simplified communication technique used by Apple and is completely based on Bluetooth technology. The Bluetooth 4.0 transmits an ID called UUID for each user and makes it easy to communicate between iPhone users.



Bluetooth has many applications, such as in telephones, tablets, media players, robotics systems, etc. The range of Bluetooth technology is between 50 – 150 meters and the data is being shared at a maximum data rate of 1 Mbps.

After launching the BLE protocol, there have been many new applications developed using Bluetooth in the field of IOT. They fall under the category of low-cost consumer products and Smart-Building applications. Like Wi-Fi, **Bluetooth also has a module Bluetooth HC-05 that can be interfaced with development boards like Arduino or Raspberry Pi to build DIY projects.** When it comes to Real-time applications, Marvell's Avastar 88W8977 comes with

Bluetooth v4.2 and has features like high speed, mesh networking for IOT. Another product, M5600 is a wireless pressure transducer with a Bluetooth v4.0 embedded in it.

## Zigbee

ZigBee is another IoT wireless protocols has features similar to the the Bluetooth technology. But it follows the IEEE 802.15.4 standard and is a high-level communication protocol. It has some advantages similar to Bluetooth i.e. low-power consumption, robustness, high security, and high scalability.

Zigbee offers a range of about 10 – 100 meters maximum and data rate to transfer data between communicated devices is around 250 Kbps. It has a large number of applications in technologies like M2M & IOT.



Having limitations in regards to data rate, range, and power consumption, Zigbee is only appropriate for Small-Scale Wireless applications. Though having some limitations, **it provides a 128-bit AES encryption and is giving a big hand in making secure communication for Home automation & small Industrial applications.** Zigbee too has its DIY module named **XBee & XBee Pro** which can be interfaced with Arduino or Raspberry Pi boards to make simple projects or application prototypes.

The company Develco has made products using Zigbee technologies like Sensors, gateways, meter interfaces, smart plugs, smart relays, etc which all work on the Zigbee wireless Mesh network, consuming low power and free from external interferences. Another company, Datanet has Zigbee based products which are used in real-time applications already, like the DNL910 & DNL920.

Ex no:

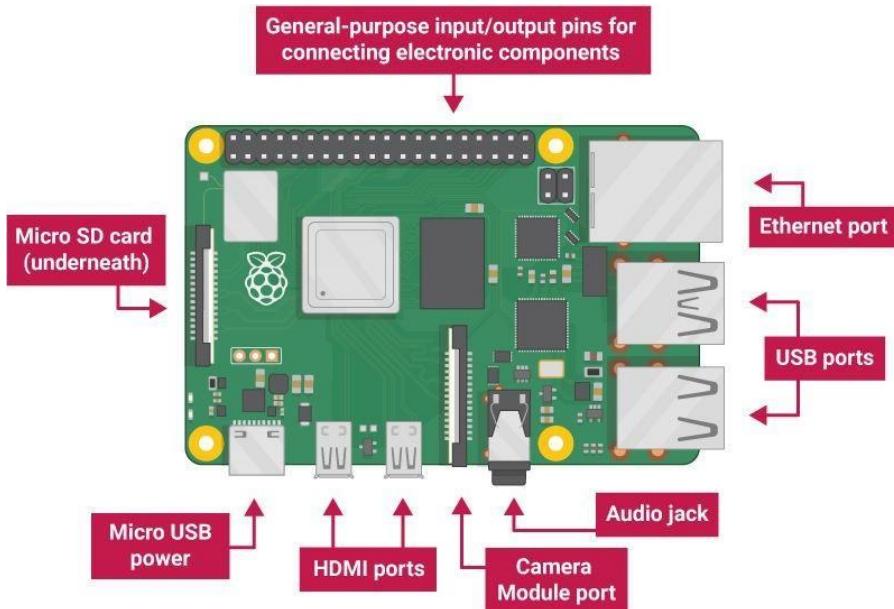
## INTRODUCTION TO RASPBERRY PI PLATFORM AND PYTHON PROGRAMMING

Date :

### Raspberry Pi

You are going to take a first look at Raspberry Pi! You should have a Raspberry Pi computer in front of you for this. The computer shouldn't be connected to anything yet.

- o Look at your Raspberry Pi. Can you find all the things labelled on the diagram?



- USB ports** — these are used to connect a mouse and keyboard. You can also connect other components, such as a USB drive.
  - SD card slot** — you can slot the SD card in here. This is where the operating system software and your files are stored.
  - Ethernet port** — this is used to connect Raspberry Pi to a network with a cable. Raspberry Pi can also connect to a network via wireless LAN.
  - Audio jack** — you can connect headphones or speakers here.
- 
- HDMI port** — this is where you connect the monitor (or projector) that you are using to display the output from the Raspberry Pi. If your monitor has speakers, you can also use them to hear sound.
  - Micro USB power connector** — this is where you connect a power supply. You should always do this last, after you have connected all your other components.
  - GPIO ports** — these allow you to connect electronic components such as LEDs and buttons

to Raspberry Pi.

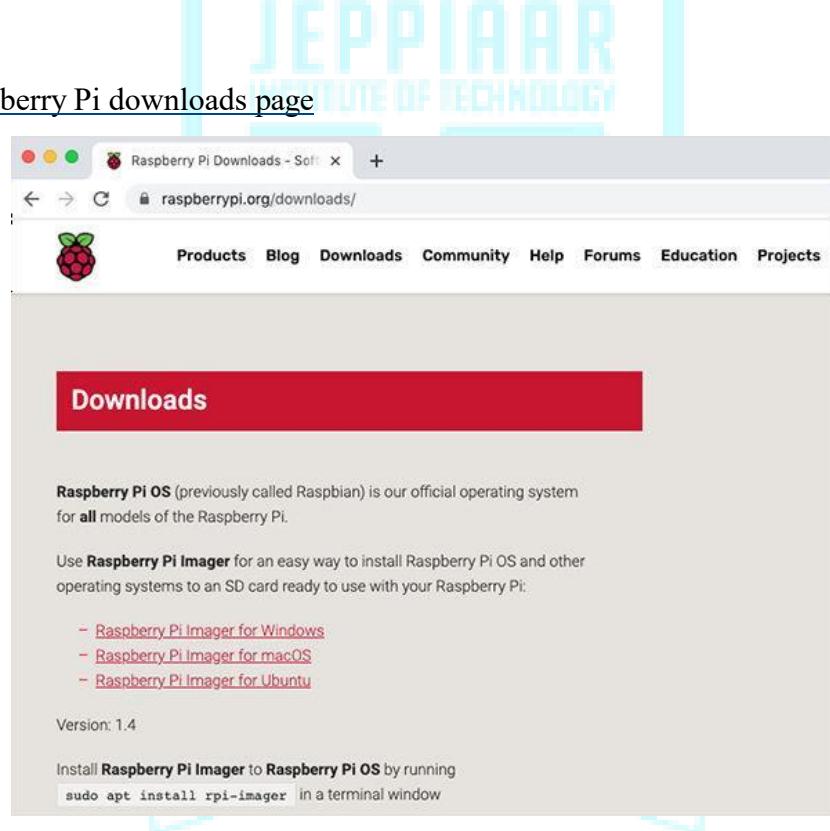
## Set up your SD card

If you have an SD card that doesn't have the Raspberry Pi OS operating system on it yet, or if you want to reset your Raspberry Pi, you can easily install Raspberry Pi OS yourself. To do so, you need a computer that has an SD card port — most laptop and desktop computers have one.

The Raspberry Pi OS operating system via the Raspberry Pi Imager

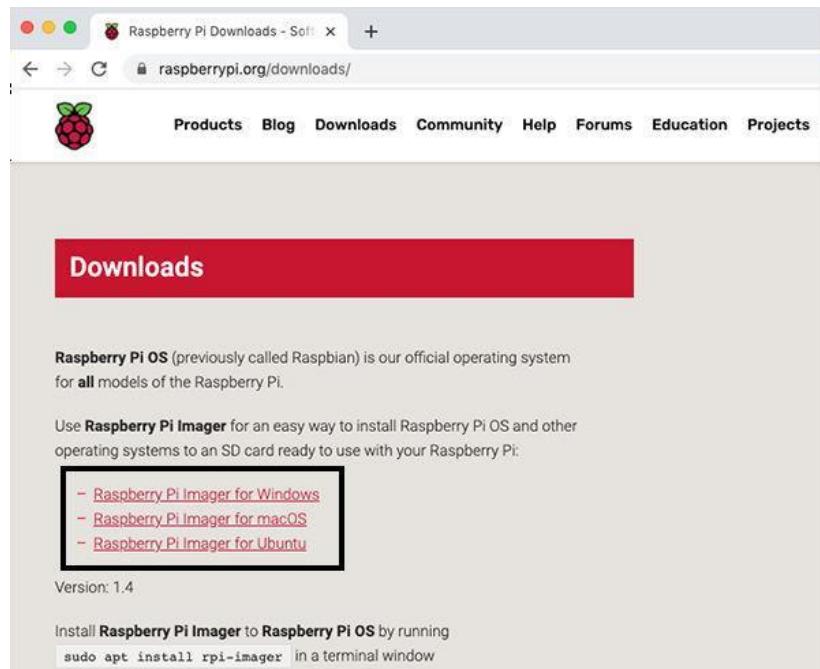
Using the Raspberry Pi Imager is the easiest way to install Raspberry Pi OS on your SD card. Note: More advanced users looking to install a particular operating system should use this guide to [installing operating system images](#).

*Download and launch the Raspberry Pi Imager*

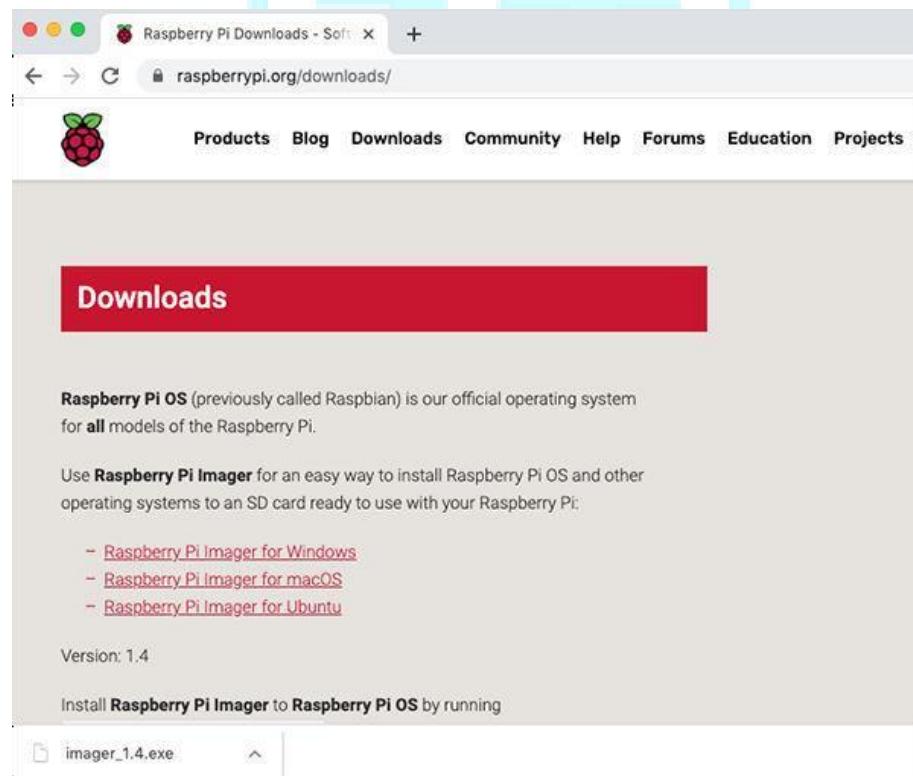


- o Visit the [Raspberry Pi downloads page](#)

- o Click on the link for the Raspberry Pi Imager that matches your operating system



- o When the download finishes, click it to launch the installer



## Using the Raspberry Pi Imager

Anything that's stored on the SD card will be overwritten during formatting. If your SD card currently has any files on it, e.g. from an older version of Raspberry Pi OS, you may wish to back up these files first to prevent you from permanently losing them.

When you launch the installer, your operating system may try to block you from running it. For example, on Windows I receive the following message:



- If this pops up, click on **More info** and then **Run anyway**
- Follow the instructions to install and run the Raspberry Pi Imager
- Insert your SD card into the computer or laptop SD card slot
- In the Raspberry Pi Imager, select the OS that you want to install and the SD card you would like to install it on

Note: You will need to be connected to the internet the first time for the the Raspberry Pi Imager to download the OS that you choose. That OS will then be stored for future offline use. Being online for later uses means that the Raspberry Pi imager will always give you the latest version.

## Raspberry Pi Imager

### Operating System

#### Raspberry Pi OS (32-bit)

A port of Debian with the Raspberry Pi Desktop (Recommended)  
Released: 2020-08-20  
Online - 1.1 GB download



#### Raspberry Pi OS (other)

Other Raspberry Pi OS based images

X



#### LibreELEC

A Kodi Entertainment Center distribution

>



#### Ubuntu

Choose from Ubuntu Core and Server images

>



#### RetroPie

Turn your Raspberry Pi into a retro gaming machine

>



## Raspberry Pi Imager

### SD Card

X



#### APPLE SSD SM0256G Media - 251.0 GB



#### AppleAPFSMedia - 250.8 GB

Mounted as /Volumes/diskoGogorra



#### WD Elements 25A3 Media - 8001.5 GB

Mounted as /Volumes/Mai Gudana



#### WD Elements 25A3 Media - 8001.5 GB

Mounted as /Volumes/Flux Capacitor

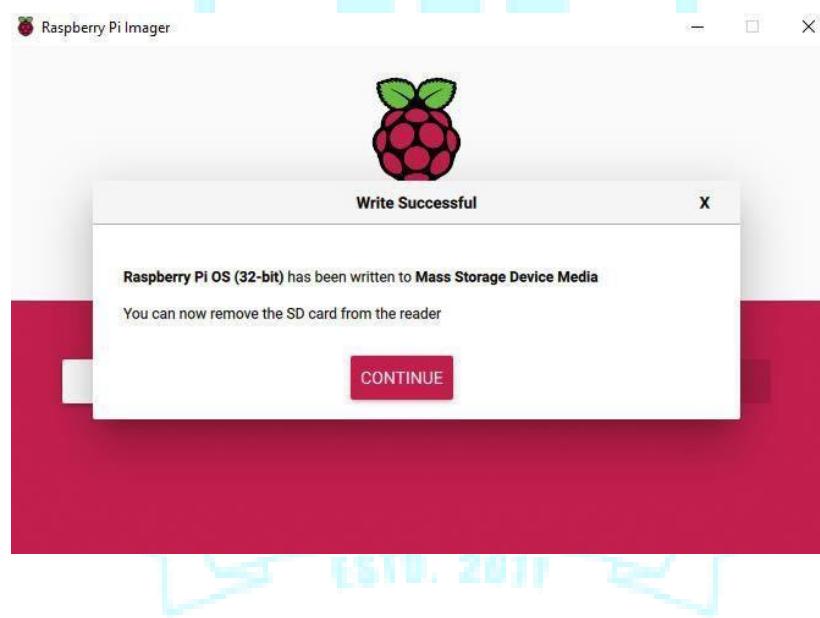


#### Mass Storage Device Media - 16.0 GB

Mounted as /Volumes/boot



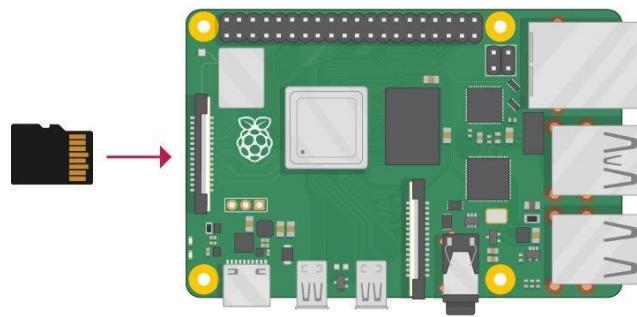
- Then simply click the **WRITE** button
- Wait for the Raspberry Pi Imager to finish writing
- Once you get the following message, you can eject your SD card



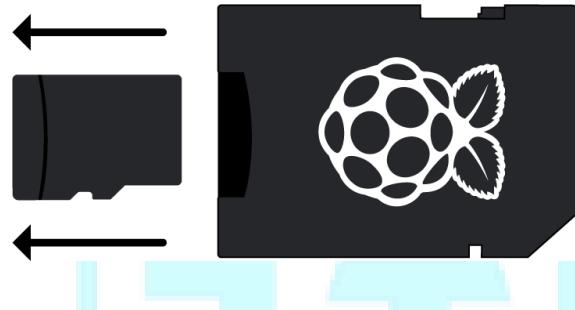
### Connect your Raspberry Pi

Let's connect up your Raspberry Pi and get it running.

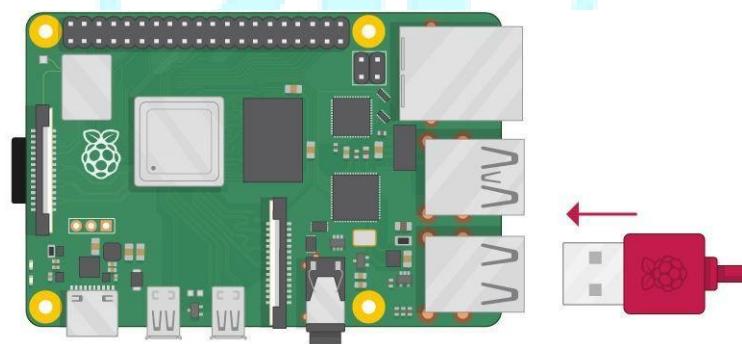
- o Check the slot on the underside of your Raspberry Pi to see whether an SD card is inside. If no SD card is there, then insert an SD card with Raspbian installed (via NOOBS).



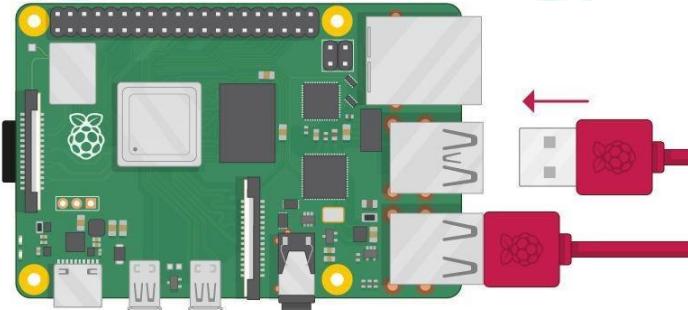
Note: Many microSD cards come inside a larger adapter — you can slide the smaller card out using the lip at the bottom.



- o Find the USB connector end of your mouse's cable, and connect the mouse to a USB port on your Raspberry Pi (it doesn't matter which port you use).



- o Connect the keyboard in the same way.

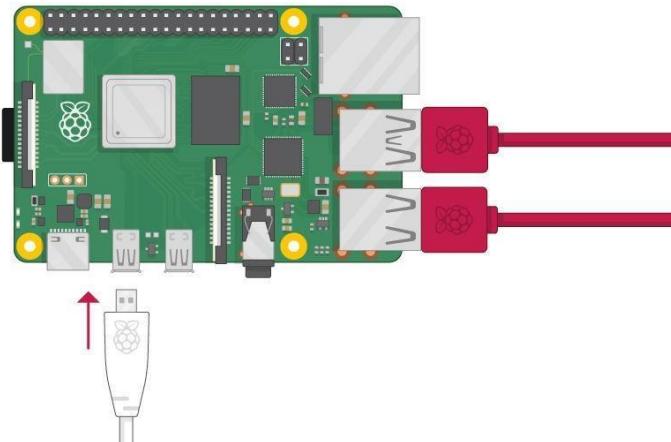


- o Make sure your screen is plugged into a wall socket and switched on.
- o Look at the HDMI port(s) on your Raspberry Pi — notice that they have a flat side on top.

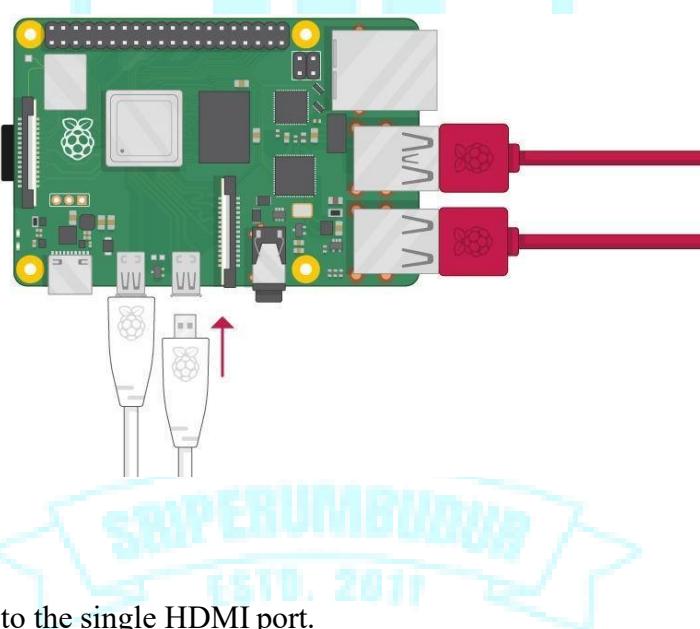
- o Use a cable to connect the screen to the Raspberry Pi's HDMI port — use an adapter if necessary.

#### Raspberry Pi 4

Connect your screen to the first of Raspberry Pi 4's HDMI ports, labelled HDMI0.

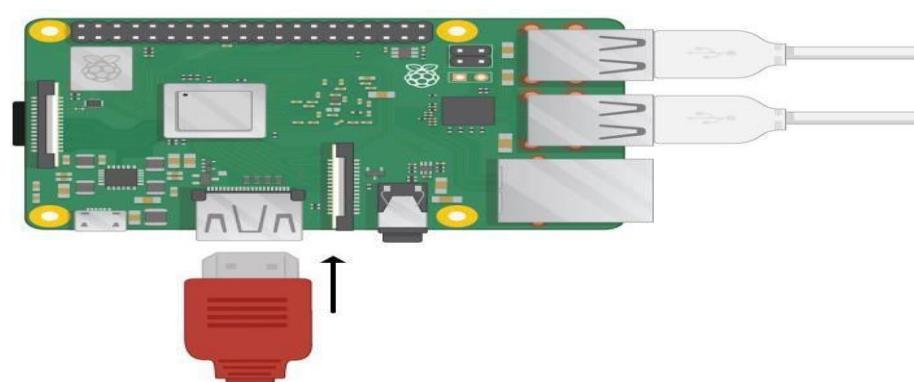


You could connect an optional second screen in the same way.



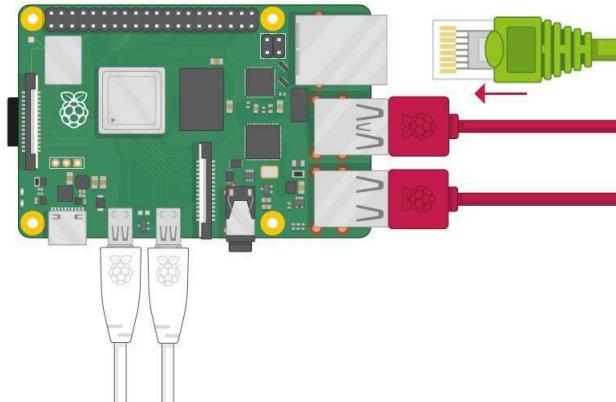
Raspberry Pi 1, 2, 3

Connect your screen to the single HDMI port.

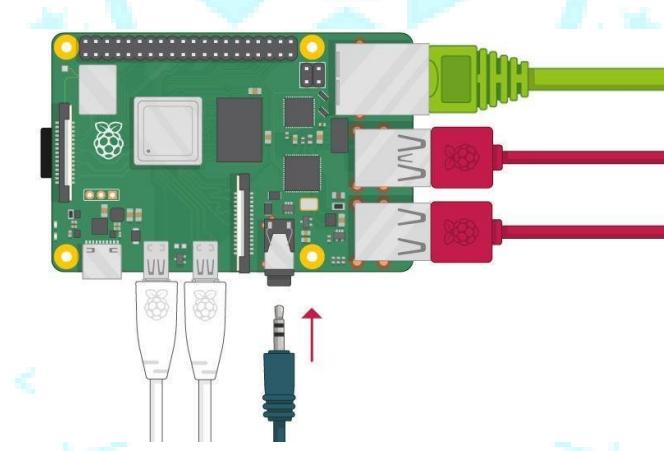


Note: nothing will display on the screen, because the Raspberry Pi is not running yet.

- o If you want to connect the Pi to the internet via Ethernet, use an Ethernet cable to connect the Ethernet port on the Raspberry Pi to an Ethernet socket on the wall or on your internet router. You don't need to do this if you want to use wireless connectivity, or if you don't want to connect to the internet.



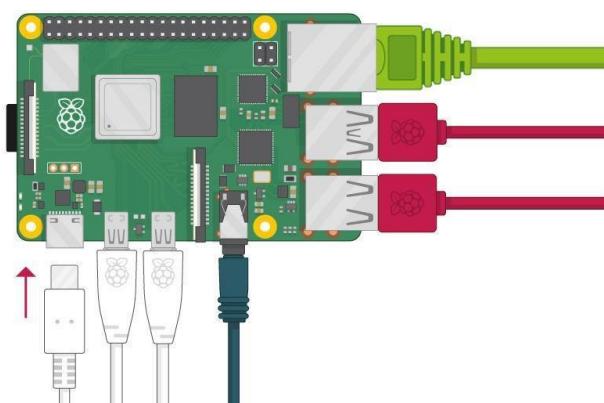
- o If your screen has speakers, your Raspberry Pi can play sound through these. Or you could connect headphones or speakers to the audio port.



Start up your Raspberry Pi

Your Raspberry Pi doesn't have a power switch. As soon as you connect it to a power outlet, it will turn on.

- o Plug the power supply into a socket and connect it to your Raspberry Pi's power port.



You should see a red LED light up on the Raspberry Pi, which indicates that Raspberry Pi is connected to power. As it starts up (this is also called booting), you will see raspberries appear in the top left-hand corner of your screen.



After a few seconds the Raspberry Pi OS desktop will appear.



### *Finish the setup*

When you start your Raspberry Pi for the first time, the Welcome to Raspberry Pi application will pop up and guide you through the initial setup.



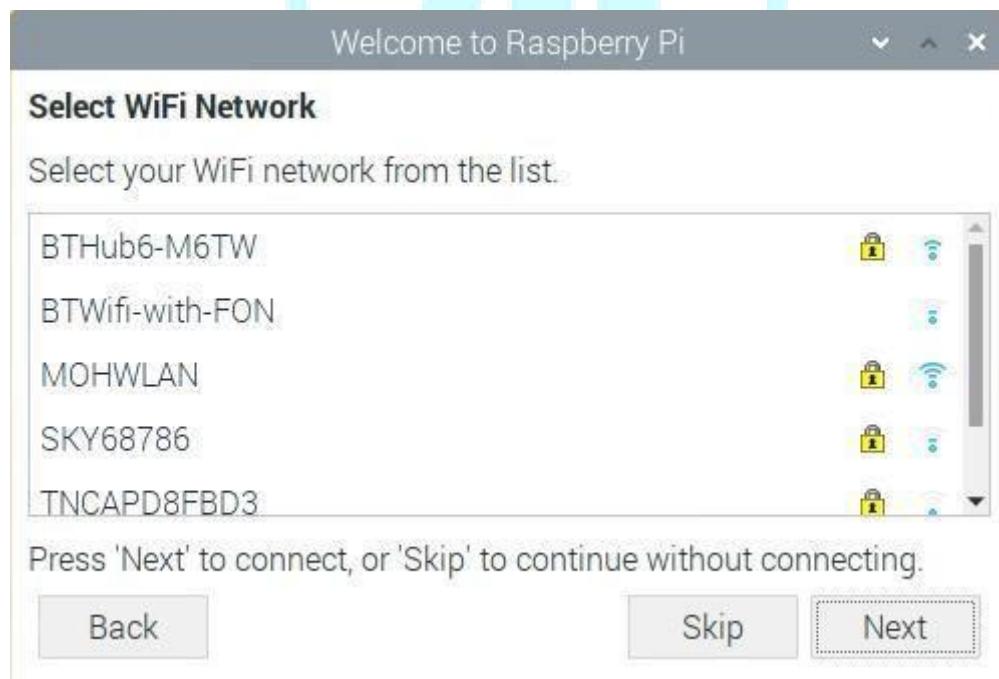
- Click Next to start the setup.
- Set your Country, Language, and Timezone, then click Next again.



- o Enter a new password for your Raspberry Pi and click Next.

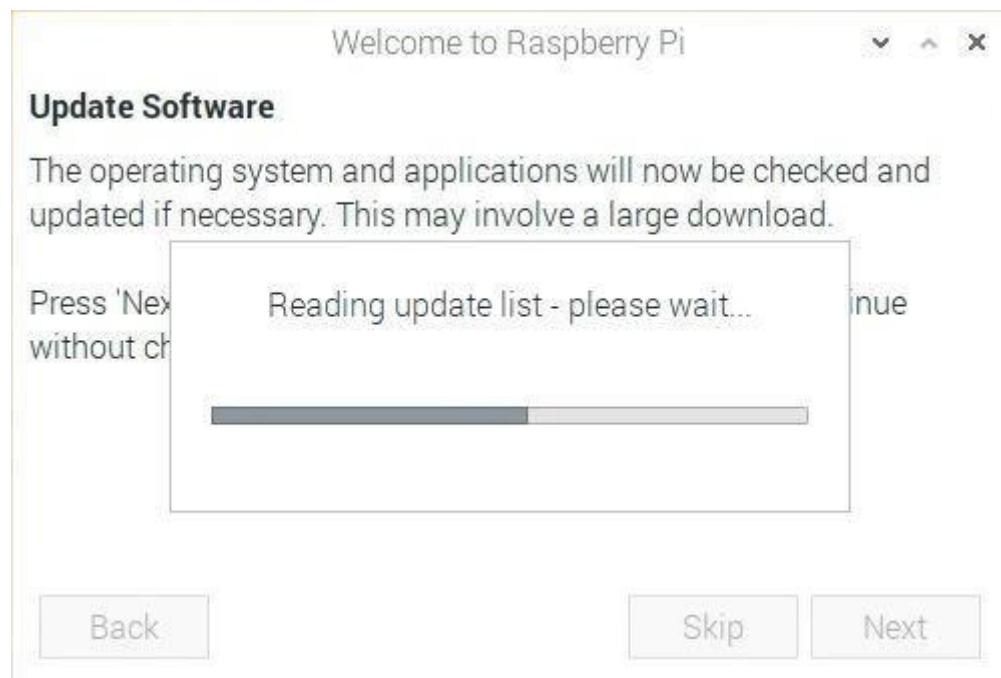


- o Connect to your WiFi network by selecting its name, entering the password, and clicking Next.



**Note:** if your Raspberry Pi model doesn't have wireless connectivity, you won't see this screen.

- o Click Next let the wizard check for updates to Raspbian and install them (this might take a little while).



- o Click Done or Reboot to finish the setup.

Note: you will only need to reboot if that's necessary to complete an update.

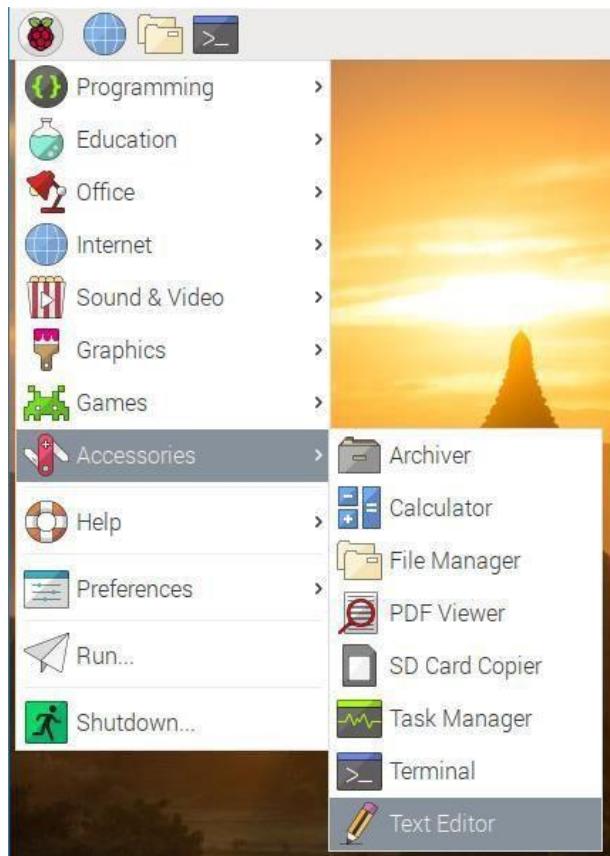


## A tour of Raspberry Pi

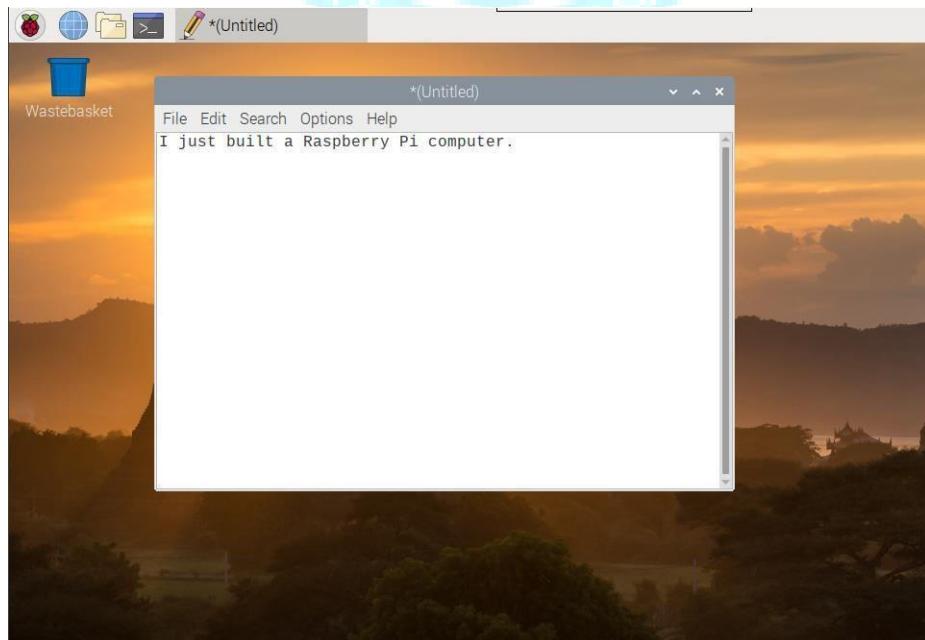
Now it's time to take a tour of your Raspberry Pi.

- o Do you see the raspberry symbol in the top left-hand corner? That's where you access the

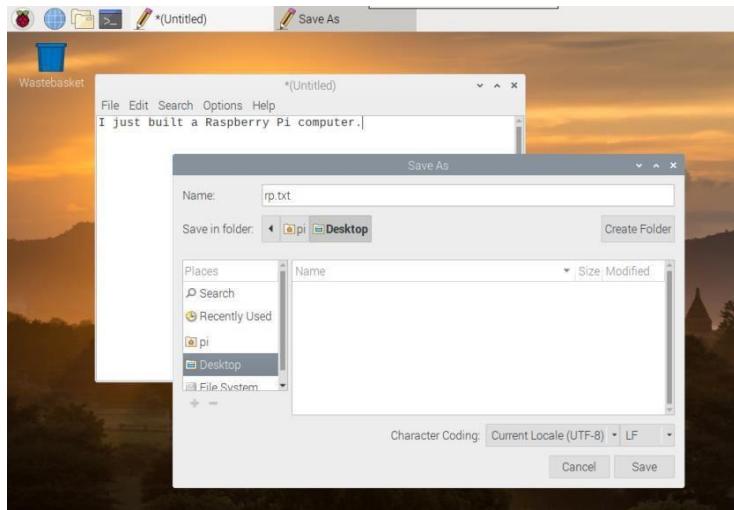
- menu: click on it to find lots of applications.
- o Click on Accessories, and then click on Text Editor.



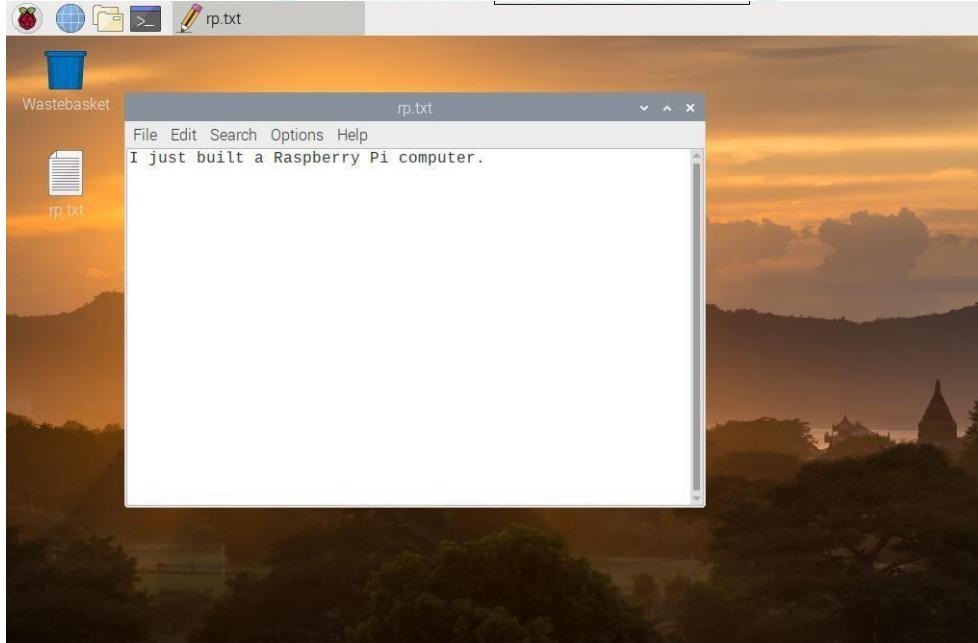
- o Type I just built a Raspberry Pi computer in the window that appears.



- o Click on File, then choose Save, and then click on Desktop and save the file as .

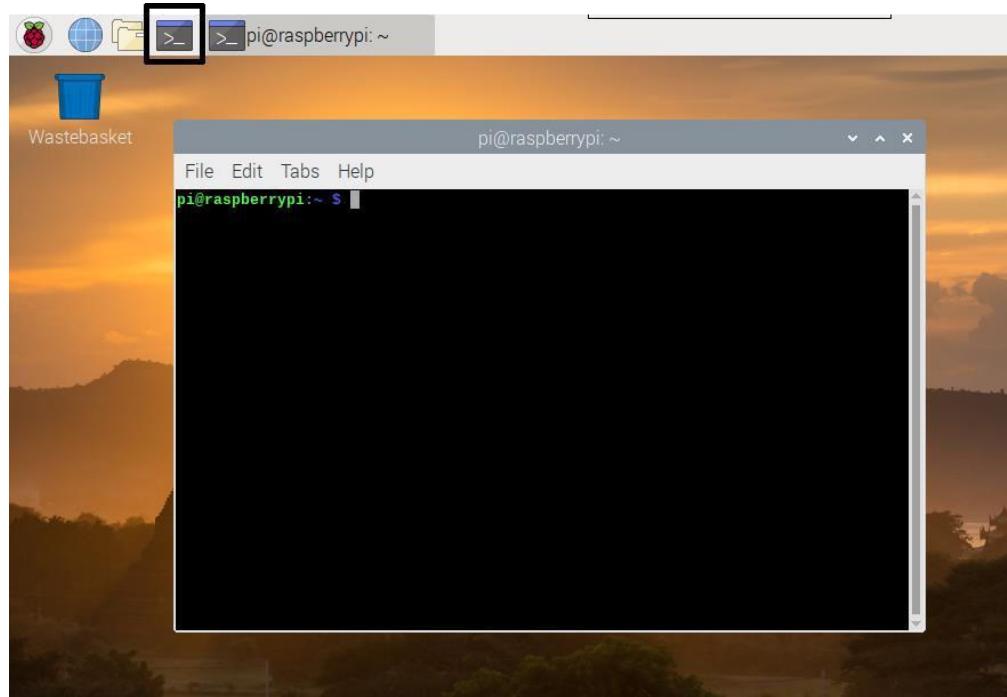


- o You should see an icon named rp.txt appear on the desktop.



Your file has been saved to your Raspberry Pi's SD card.

- Close the text editor by clicking the X in the top right-hand corner of the window.
- Return to the menu, click on Shutdown, and then click on Reboot.
- When Raspberry Pi has rebooted, your text file should still be there on the desktop.
- Raspberry Pi runs a version of an operating system called Linux (Windows andmacOS are other operating systems). This operating system allows you to make things happen by typing in commands instead of clicking on menu options. To try this out, click on the Terminal symbol at the top of the screen:



- o In the window that appears, type:

```
ls
```

and then press Enter on the keyboard.

You can now see a list of the files and folders in your directory.

- o Now type this command to change directory to the Desktop:

```
cd Desktop
```

You have to press the Enter key after every command.

Then type:

```
ls
```

Can you see the text file you created?

- o Close the terminal window by clicking on the X.
- o Now drag to the Wastebasket on the desktop so the Raspberry Pi will be tidy for the next person using it.



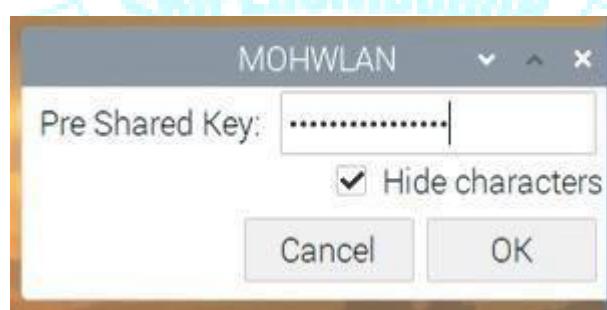
## Browsing the web

You might want to connect your Raspberry Pi to the internet. If you didn't plug in an ethernet cable or connect to a WiFi network during the setup, then you can connect now.

- o Click the icon with red crosses in the top right-hand corner of the screen, and select your network from the drop-down menu. You may need to ask an adult which network you should choose.



- o Type in the password for your wireless network, or ask an adult to type it for you, then click OK.



- o When your Pi is connected to the internet, you will see a wireless LAN symbol instead of the red crosses.

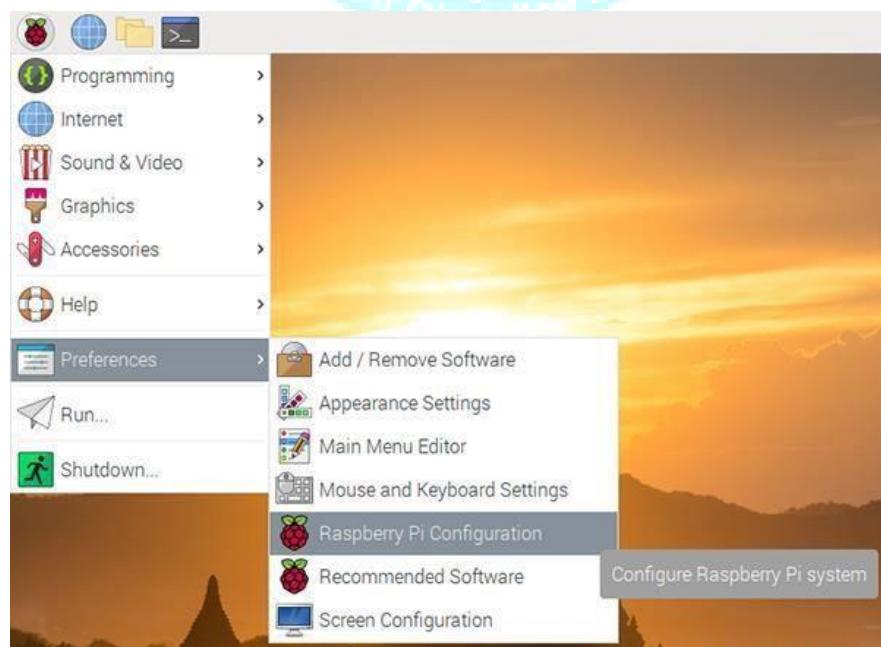


- o Click the web browser icon and search for .



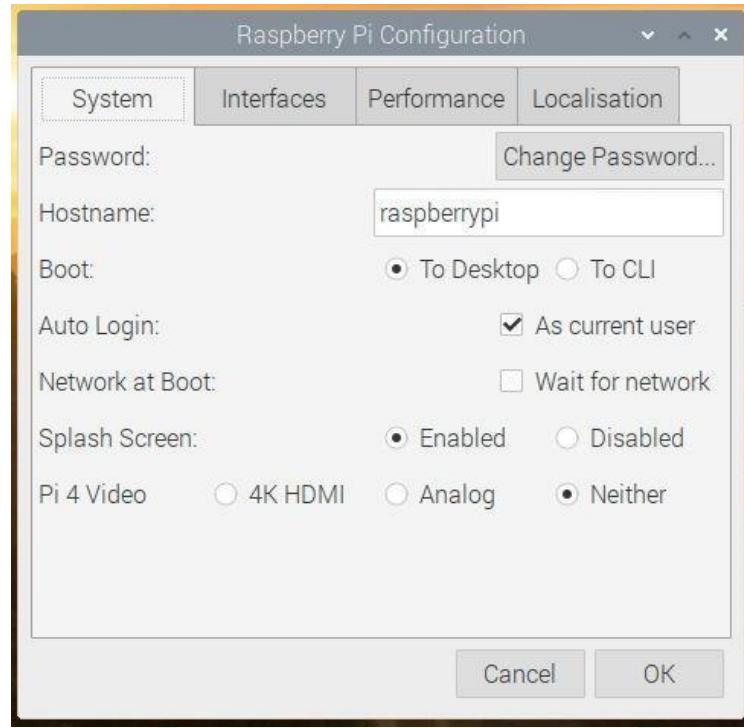
## Configuring your Raspberry Pi

You can control most of your Raspberry Pi's settings, such as the password, through the Raspberry Pi Configuration application found in Preferences on the menu.



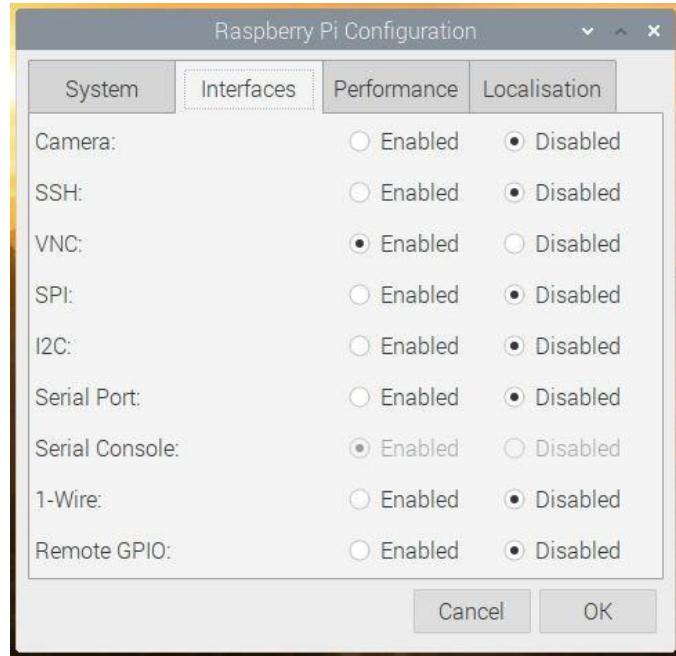
### □ System

In this tab you can change basic system settings of your Raspberry Pi.



- Password — set the password of the **pi** user (it is a good idea to change the password from the factory default ‘raspberry’)
- Boot — select to show the Desktop or CLI (command line interface) when your Raspberry Pi starts
- Auto Login — enabling this option will make the Raspberry Pi automatically log in whenever it starts
  
- Network at Boot — selecting this option will cause your Raspberry Pi to wait until a network connection is available before starting
- Splash Screen — choose whether or not to show the splash (startup) screen when your Raspberry Pi boots
- Interfaces

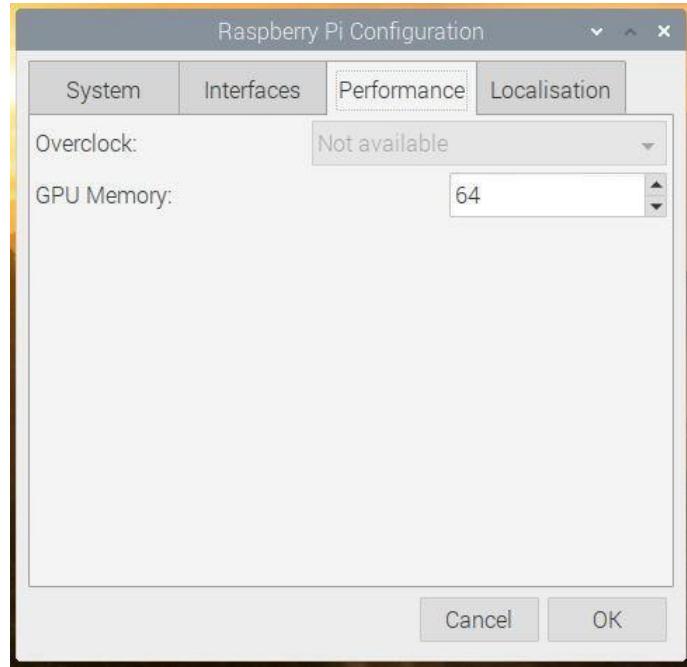
You can link devices and components to your Raspberry Pi using a lot of different types of connections. The Interfaces tab is where you turn these different connections on or off, so that your Raspberry Pi recognises that you've linked something to it via a particular type of connection.



- Camera — enable the [Raspberry Pi Camera Module](#)
- SSH — allow remote access to your Raspberry Pi from another computer using SSH
- VNC — allow remote access to the Raspberry Pi Desktop from another computer using VNC
- SPI — enable the SPI GPIO pins
- I2C — enable the I2C GPIO pins
- Serial — enable the Serial (Rx, Tx) GPIO pins
- 1-Wire — enable the 1-Wire GPIO pin.
- Remote GPIO — allow access to your Raspberry Pi's GPIO pins from another computer
- Performance

If you need to do so for a particular project you want to work on, you can change the performance settings of your Raspberry Pi in this tab.

Warning: Changing your Raspberry Pi's performance settings may result in it behaving erratically or not working.



- Overclock — change the CPU speed and voltage to increase performance
- GPU Memory — change the allocation of memory given to the GPU
- Localisation



This tab allows you to change your Raspberry Pi settings to be specific to a country or location.

- Locale — set the language, country, and character set used by your Raspberry Pi
- Timezone — set the time zone
- Keyboard — change your keyboard layout
- WiFi Country — set the WiFi country code

**Ex no :**

## **INTERFACING OF LED**

**Date :**

**Aim :**

write a program to blink LED using Arduino

**Apparatus required :**

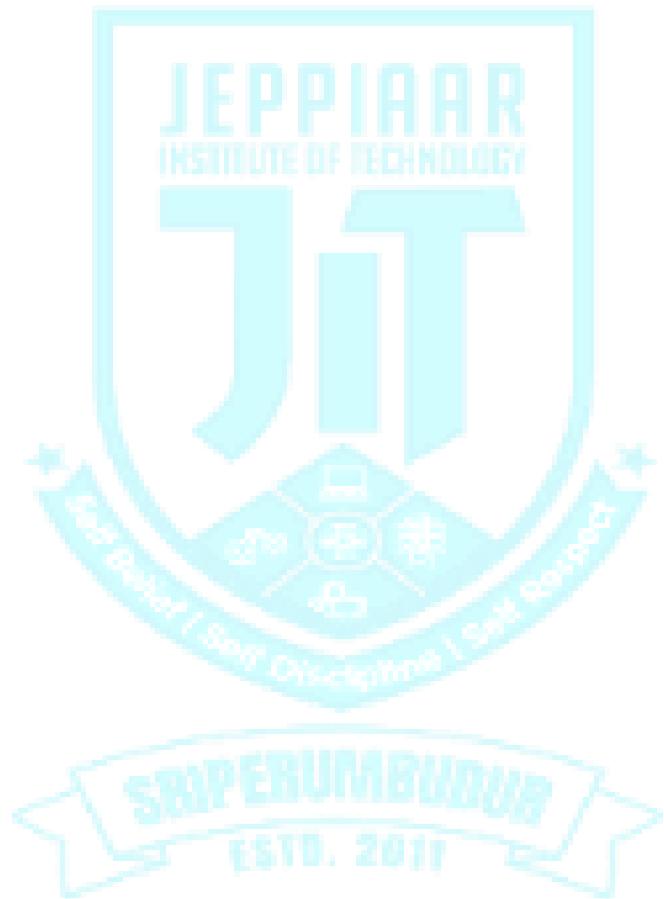
1. Computer
2. Arduino software

**Procedure:**

1. Connect the Arduino UNO board to the system make sure all the connections . and install the Arduino IDE 2.11 in the system
2. Open Arduino IDE 2.11 - tools-board- Arduino UNO
3. Go to file –new program – save – compile –run
4. Now,view the LED blinking in the Arduino UNO board ,it shows LED on for one second and off for one second

**Program :**

```
Void setup ()  
{  
Pinmode(LED _BUILTIN,OUTPUT);  
}  
Void loop()  
{  
Digitalwrite(LED _BUILTIN, HIGH);  
Delay(1000)  
Digitalwrite(LED _BUILTIN, LOW);  
Delay(1000)  
}
```



### **Result :**

Thus the interfacing LED is successfully completed.

**Ex no:**

## **WIRELESS COMMUNICATION USING AURDINO**

**Date :**

**Aim :**

Write a program to using wireless communication medium in Arduino

**Apparatus required :**

1. Computer
2. Arduino software

**Procedure:**

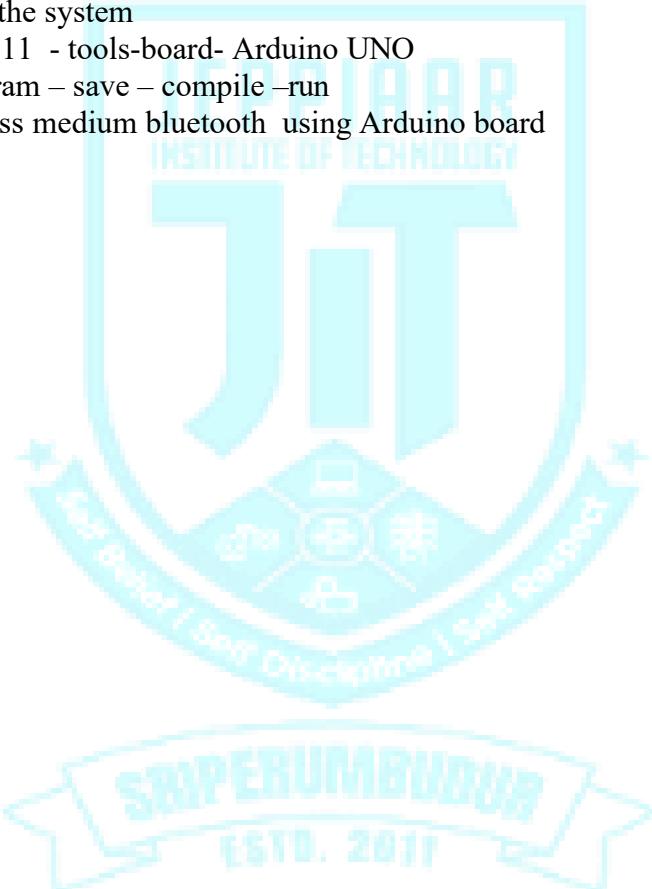
1. Connect the Arduino UNO board to the system make sure all the connections . and install the Arduino IDE 2.11 in the system
2. Open Arduino IDE 2.11 - tools-board- Arduino UNO
3. Go to file –new program – save – compile –run
4. Now, view the wireless medium bluetooth using Arduino board

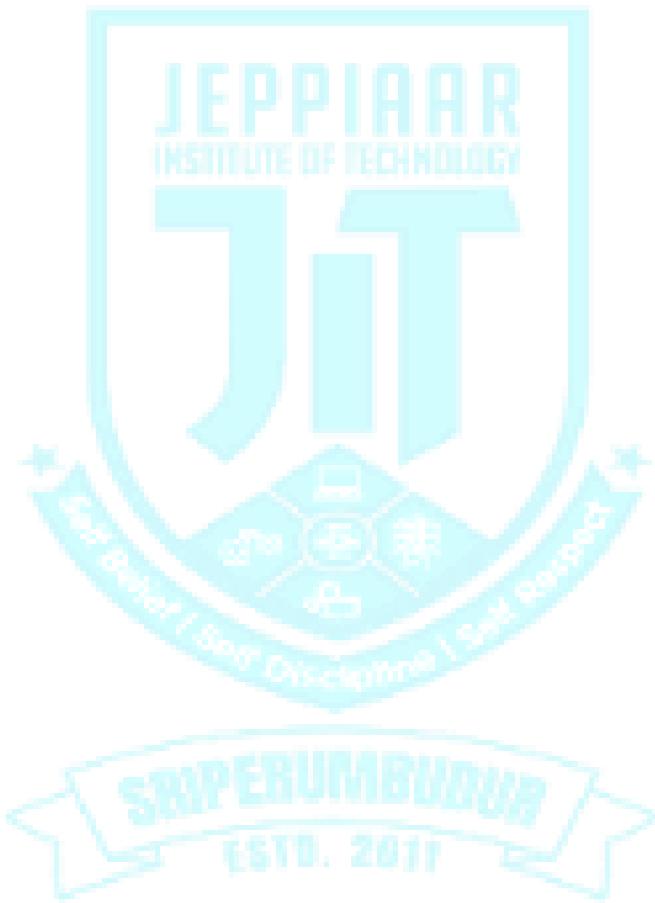
**Program**

```
Void setup()
{
Pinmode(12,OUTPUT);
Serial.begin(9400);

}

Void loop
{
Char inchar = (char)serial.read();
If (inchar =='A')
{
Digitalwrite (12, HIGH);
Serial printin (inchar);
}
elseif(inchar =='D')
Digitalwrite (12,LOW);
Serial printin (inchar);
}
```





**Result :**

Thus the wireless communication using Arduino is successfully completed.

Ex no :

## SETUP A CLOUD PLATFORM TO LOG THE DATA

Date :

Aim:

Building and hosting a simple website(static/dynamic) on the device and make it accessible online. There is a need to install server(eg: Apache) and thereby host the website.

### *Setting up an Apache Web Server on a Raspberry Pi*

Apache is a popular web server application you can install on the Raspberry Pi to allow it to serve web pages.

On its own, Apache can serve HTML files over HTTP, and with additional modules can serve dynamic web pages using scripting languages such as PHP.

#### *Install Apache*

First, update the available packages by typing the following command into the Terminal:

```
sudo apt update
```

Then, install the apache2 package with this command:

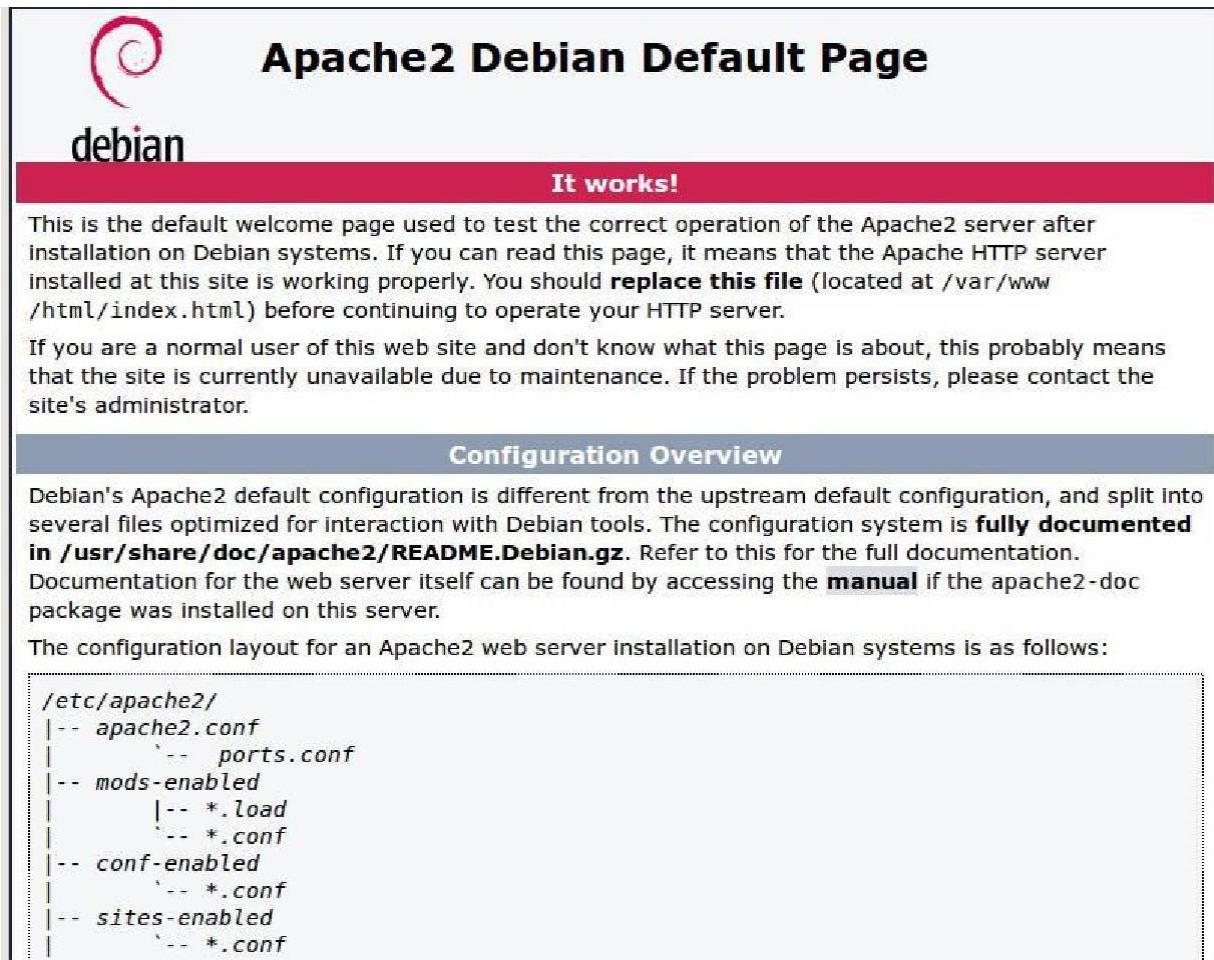
```
sudo apt install apache2 -y
```

#### *Test the web server*

By default, Apache puts a test HTML file in the web folder. This default web page is served when you browse to <http://localhost/> on the Pi itself, or <http://192.168.1.10> (whatever the

Pi's IP address is) from another computer on the network. To find the Pi's IP address, type `hostname -I` at the command line (or read more about finding your [IP address](#)).

Browse to the default web page either on the Pi or from another computer on the network and you should see the following:



The screenshot shows a web browser window titled "Apache2 Debian Default Page". The header features the Debian logo and the text "Apache2 Debian Default Page". Below the header is a red bar containing the text "It works!". The main content area contains the following text:  
**This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.**  
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

### Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   '-- ports.conf
|-- mods-enabled
|   '-- *.load
|   '-- *.conf
|-- conf-enabled
|   '-- *.conf
|-- sites-enabled
|   '-- *.conf
```

This means you have Apache working!

Changing the default web page

This default web page is just an HTML file on the filesystem. It is located at `/var/www/html/index.html`.

Navigate to this directory in a terminal window and have a look at what's inside:

```
cd  
/var/www/html
```

This will show you:

```
total 12  
drwxr-xr-x 2 root root 4096 Jan 8 01:29 .  
drwxr-xr-x 12 root root 4096 Jan 8 01:28 ..  
-rw-r--r-- 1 root root 177 Jan 8 01:29 index.html
```

*Additional - install PHP*

To allow your Apache server to process PHP files, you'll need to install the latest version of PHP and the PHP module for Apache. Type the following command to install these:

```
sudo apt install php libapache2-mod-php -y
```

Now remove the `index.html` file:

```
sudo rm index.html
```

and create the file `index.php`:

```
sudo nano index.php
```

Put some PHP content in it:

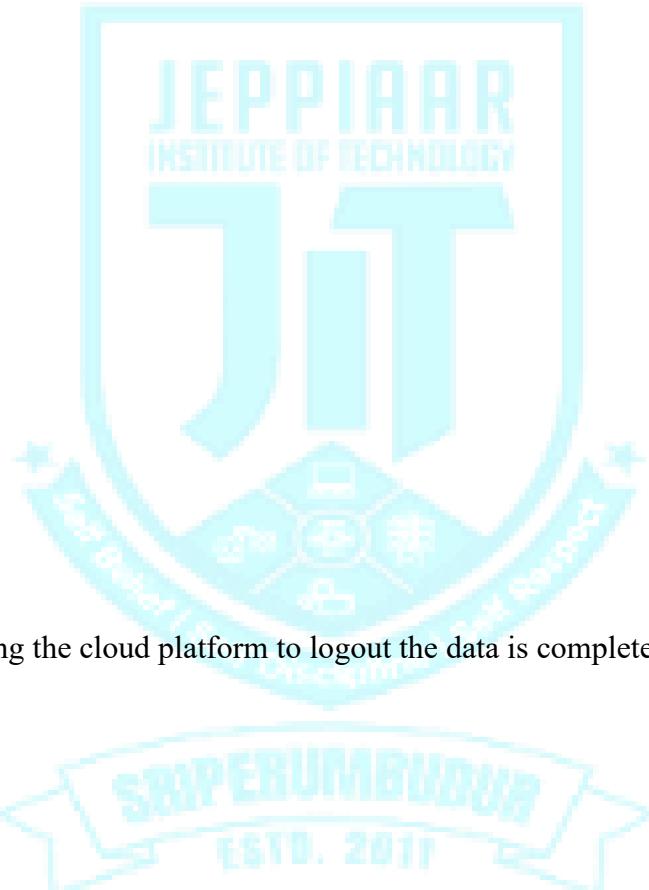
```
<?php echo "hello world"; ?>
```

Now save and refresh your browser. You should see "hello world". This is not dynamic but still served by PHP. Try something dynamic:

```
<?php echo date('Y-m-d H:i:s'); ?>
```

or show your PHP info:

```
<?php phpinfo(); ?>
```



Result :

Thus the setting the cloud platform to logout the data is completed successfully.

**Ex no:** LOG DATA USING RASPBERRY PI AND UPLOAD TO THE CLOUD PLATFORM

**Date :**

**Aim :**

Interfacing the regular usb webcam with the device and turn it into fully functional IPwebcam & test the functionality.

## Using a standard USB webcam

Rather than using the Raspberry Pi camera module, you can use a standard USBwebcam to take pictures and video on the Raspberry Pi.

Note that the quality and configurability of the camera module is highly superior to a standard USB webcam.

Install fswebcam

First, install the **fswebcam** package:

```
sudo apt install fswebcam
```

Add your user to **video** group

If you are not using the default **pi** user account, you need to add your username to

the **vide** group, otherwise you will see 'permission denied' errors.

```
sudo usermod -a -G video <username>
```

To check that the user has been added to the group correctly, use the **groups** command.

Basic usage

Enter the command

**fswebcam** followed by a filename and a picture will be

taken using the webcam, and saved to the filename specified:

```
fswebcam image.jpg
```

This command will show the following information:

```
--- Opening /dev/video0...
Trying source module
v4l2...
/dev/video0 opened.
No input was specified, using the first.
Adjusting resolution from 384x288 to
352x288.
--- Capturing frame...
Corrupt JPEG data: 2 extraneous bytes before marker
```



Note the small default resolution used, and the presence of a banner showing the timestamp.

Specify resolution

The webcam used in this example has a resolution of

so to specify the

resolution I want the image to be taken at, use the

-r flag:

```
fswebcam -r 1280x720 image2.jpg
```

This command will show the following information:

```
--- Opening /dev/video0...
```

```
Trying source module v4l2...
```

```
/dev/video0 opened.
```

```
No input was specified, using the first.
```

```
--- Capturing frame...
```

```
Corrupt JPEG data: 1 extraneous bytes before marker
```

```
0xd5Captured frame in 0.00 seconds.
```

```
--- Processing captured image...
```



Picture now taken at the full resolution of the webcam, with the banner present.

Specify no banner

Now add the  --no- flag:

```
fswebcam -r 1280x720 --no-banner image3.jpg
```

which shows the following information:

```
--- Opening /dev/video0...
Trying source module
v4l2...
/dev/video0 opened.
No input was specified, using the first.
--- Capturing frame...
Corrupt JPEG data: 2 extraneous bytes before marker
0xd6Captured frame in 0.00 seconds.

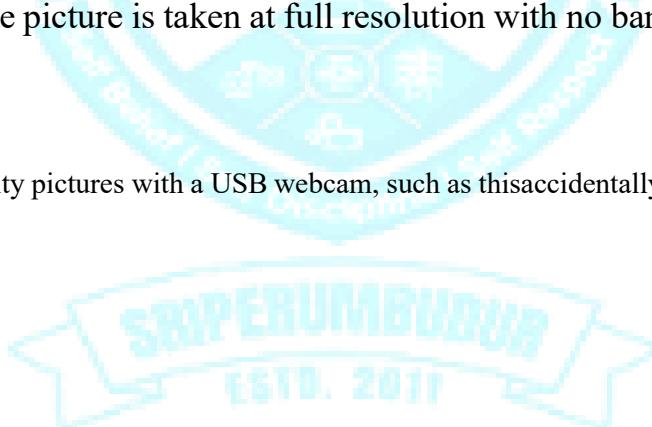
--- Processing captured
```



Now the picture is taken at full resolution with no banner.

### Bad Pictures

You may experience poor quality pictures with a USB webcam, such as this accidentally artistic piece:





Some webcams are more reliable than others, but this sort of issue may occur with poor quality webcams. If the problem persists, ensure your system is [up to date](#). Also try other webcams, but you'll get the best performance from the RaspberryPi [camera module](#).

#### Bash script

You can write a Bash script which takes a picture with the webcam. The script below saves the images in the `/home/pi/web` directory, so create the `webcam` subdirectory first with:

```
mkdir webcam
```

To create a script, open up your editor of choice and write the following example code:

```
#!/bin/bash  
DATE=$(date +"%Y-%m-%d_%H%M")  
fswebcam -r 1280x720 --no-banner /home/pi/webcam/$DATE.jpg
```

This script will take a picture and name the file with a timestamp. Say we saved it

as `webcam.sh`, we would first make the file executable:

```
chmod +x webcam.sh
```

Then run with:

```
./webcam.sh
```

Which would run the commands in the file and give the usual output:

```
--- Opening /dev/video0...  
Trying source module  
v4l2...  
  
/dev/video0 opened.  
No input was specified, using the first.  
--- Capturing frame...  
Corrupt JPEG data: 2 extraneous bytes before marker
```

```
Disabling banner.  
Writing JPEG image to '/home/pi/webcam/2013-06-07_2338.jpg'.
```

Time-lapse using cron

You can use `cron` to schedule taking a picture at a given interval, such as every minute to capture a time-lapse.

First open the cron table for editing:

```
crontab -e
```

This will either ask which editor you would like to use, or open in your default editor. Once you have the file open in an editor, add the following line to schedule taking a picture every minute (referring to the Bash script from above):

```
* * * * * /home/pi/webcam.sh 2>&1
```

Save and exit and you should see the message:

```
crontab: installing new crontab
```

Ensure your script does not save each picture taken with the same filename. This will overwrite the picture each time.

**Result :**

Thus, the Log Data using Raspberry PI and upload to the cloud platform is completed successfully.

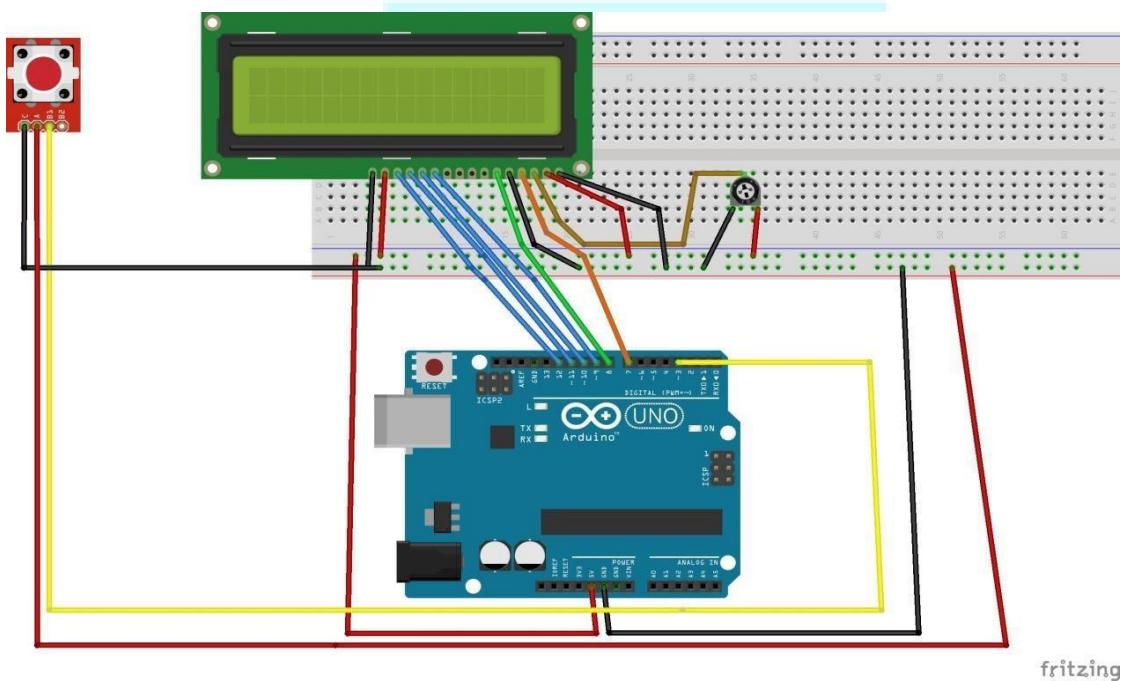
Ex no :

## **DESIGN AN IoT SYSTEMS**

Date :

## Aim:

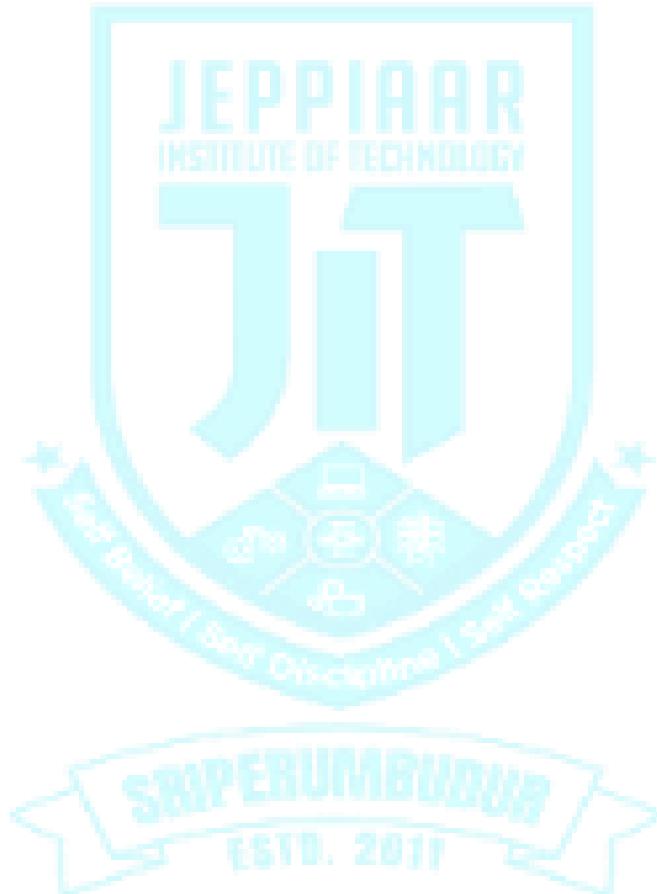
Instead of using the conventional dice, generate a random value similar to dice value and display the same using a 16X2 LCD. A possible extension could be to provide the user with option of selecting single or double dice game.



```
#include <LiquidCrystal.h>
long randNumber;
int Led = 13; //define LED port
int Shock = 2; //define shock port
int val;//define digital variable val
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(7, 8, 9, 10, 11, 12 );
byte customChar[] = {
    B00000,
    B00000,
    B11111,
    B11001,
```

```
B10101,  
B10011,  
  
B11111,  
B00000  
};  
void setup()  
{  
  
lcd.begin(16, 2);  
lcd.createChar(0, customChar);  
lcd.home();  
pinMode(Led, OUTPUT); //define LED as a output port  
randomSeed(analogRead(0));  
pinMode(Shock, INPUT); //define shock sensor as a output port  
lcd.write(byte( 0));  
lcd.print("Digital dice");  
lcd.write(byte( 0));  
delay(1000);  
}  
  
void loop()  
{  
  
val = digitalRead(Shock); //read the value of the digital interface 3 assigned to val  
if (val == LOW) //when the shock sensor have signal do the following  
{  
lcd.clear();  
lcd.print("Rolling dice...");  
delay(4000);  
lcd.clear();  
lcd.setCursor(0, 0);  
randNumber = random(1,7);  
lcd.print("Dice 1 = ");  
lcd.print(randNumber);  
lcd.setCursor(0, 1);  
randNumber = random(1,7);
```

```
lcd.print("Dice 2 = ");  
lcd.print(randNumber);  
  
}  
  
delay(150);  
}
```



### **Result :**

Thus the IoT based deicing systems is designed successfully.