

Draft Study Material

Telecom Technician – IoT Device/ Systems

(Job Role)



(Qualification Pack: Ref. Id. TEL/Q6210)

Sector: Telecom

Textbook for Class XII



PSS CENTRAL INSTITUTE OF VOCATIONAL EDUCATION

(a constituent unit of NCERT, under Ministry of Education, Government of India)

Shyamla Hills, Bhopal- 462 002, M.P., India

<https://www.psscive.ac.in>

No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior permission of the publisher.

Preface

Vocational Education is a dynamic and evolving field, and ensuring that every student has access to quality learning materials is of paramount importance. The journey of the PSS Central Institute of Vocational Education (PSSCIVE) toward producing comprehensive and inclusive study material is rigorous and time-consuming, requiring thorough research, expert consultation, and publication by the National Council of Educational Research and Training (NCERT). However, the absence of finalized study material should not impede the educational progress of our students. In response to this necessity, we present the draft study material, a provisional yet comprehensive guide, designed to bridge the gap between teaching and learning, until the official version of the study material is made available by the NCERT. The draft study material provides a structured and accessible set of materials for teachers and students to utilize in the interim period. The content is aligned with the prescribed curriculum to ensure that students remain on track with their learning objectives.

The contents of the modules are curated to provide continuity in education and maintain the momentum of teaching-learning in vocational education. It encompasses essential concepts and skills aligned with the curriculum and educational standards. We extend our gratitude to the academicians, vocational educators, subject matter experts, industry experts, academic consultants, and all other people who contributed their expertise and insights to the creation of the draft study material.

Teachers are encouraged to use the draft modules of the study material as a guide and supplement their teaching with additional resources and activities that cater to their students' unique learning styles and needs. Collaboration and feedback are vital; therefore, we welcome suggestions for improvement, especially by the teachers, in improving upon the content of the study material.

This material is copyrighted and should not be printed without the permission of the NCERT-PSSCIVE.

Deepak Paliwal
(Joint Director)
PSSCIVE, Bhopal

Date: 16 April 2026

STUDY MATERIAL DEVELOPMENT COMMITTEE

Members

Prof Dr Prakash Khanale, Senior Academic Coordinator, Yashwantrao Chavan Maharashtra Open University, Nashik., Maharashtra and Ex. Head Department of Computer Science, DSM College, Parbhani, Maharashtra
Mrs. Rita Jain, Professor in Electronic and Communication, LNCT, Bhopal and Director (COI IOT), Workspace, Bhopal

Member Coordinator

Prof Dr Deepak D. Shudhalwar, Professor (CSE), Head, Department of Engineering and Technology, PSSCIVE, NCERT, Bhopal, Madhya Pradesh

CONTENTS

Unit 1. Setup and Configuration of IoT Devices

Chapter 1. Installation of Arduino IDE and Raspberry Pi

1.1 Arduino

1.1.1 Arduino IDE Installation

1.1.2 Connection of Arduino Board to Computer for Data Transfer

1.2 Raspberry Pi

1.2.1 Connection of Raspberry Pi board to Computer for Data Transfer

1.3 Checking Connectivity

1.4 Managing the communication hurdles

Chapter 2. Introduction to Arduino Programming

2.1 Program Structure

2.2 Libraries

2.3 Arduino Software Tools

2.4 Workflow in Arduino IDE

2.5 Simulation tools

2.5.1 Steps for simulation of an Arduino

Chapter 3. Initialize and Configure Nodes, Gateways, and Control Edge Appliances

3.1 IoT Device Installation

3.1.1 Unpack the Camera

3.1.2 Set Up the Basic Camera Hardware

3.1.3 Ensure the Power Supply and Cable Connection,

3.1.4 Set Up the Internet Connection of Camera Device

3.1.5 Installation of edge devices

3.2 Testing of network connectivity

3.3 Software Execution

Chapter 4. Establish Communication and Connectivity between devices

4.1 Types of USB Data Transfer

4.2 Data Transfer Modes

4.2.1 Simplex

4.2.2 Half Duplex mode

4.2.3 Full Duplex Mode

4.3 Controlling the Data Transfer

4.4 Data Transfer Using the Indicators

4.5 Comparison of data transfer techniques over various networks

4.6 Causes of Data Transfer Failure

4.7 Connecting to the Network Remotely

4.8 Connecting to Different Short- Range Wireless Networks

Chapter 5. Authentication and Authorization Mechanism in IoT

5.1 Authenticating the Edge Devices

5.1.1 Password-based authentication

5.1.2 Token-based authentication

5.1.3 Biometric authentication

5.1.4 Multi-factor authentication

5.2 Security Challenges in Authentication

5.3 Authorization of the Edge Devices

5.4 Installation of Access Control System

5.5 Access Control System Architecture for IoT

5.5.1 Distributed Architecture

5.5.2 Centralized Architecture

5.6 Control Access Using Security

5.7 Securing Wireless Connection

5.8 Check for Malware and DDoS Attacks- Malware Attacks, DDoS Attacks

Unit 2. Level 1 Troubleshooting of IoT Devices

Chapter 6 Testing and checking Connectivity

6.1 Levels of IoT

6.2 IoT testing challenges

6.3 Testing of IoT Systems

6.3.1 Usability testing

6.3.2 Compatibility testing

6.3.3 Regulatory testing

6.3.4 Security and privacy testing

6.3.5 Upgrade testing

6.3.6 Performance and real-time testing

6.3.7 End-user application testing

6.3.8 Connectivity testing

6.3.9 Functional Testing

6.4 Setting Up Components for IoT Testing

6.4.1 Understand the device

6.4.2 Prepare the device

6.4.3 Set up the server

6.4.4 Enable connections

6.5 IoT Testing

6.5.1 Performing Manual IoT Testing

6.5.2 Performing Automatic IoT Testing

6.6 IoT Testing Tools

6.6.1 software tools

6.6.2 hardware tools

6.7 Testing of IoT devices

6.7.1 Testing of Sensors

6.7.2 Testing of Actuators

6.8 Verifying Network Connectivity

6.8.1 Event Viewer

6.8.2 Hardware components

6.8.3 Local connectivity

6.9 IoT connectivity

6.9.1 Requirements for good IoT Connectivity

6.9.2 Importance of Choosing the Right Connectivity Technology

6.9.3 Connecting the Internet of Things- Trade-off between Power Consumption, Range, and Bandwidth

6.10 Testing of Connectivity Modules

6.11 End -to- End Testing of IoT network

Chapter 7 Testing storage, power supply and internet connectivity of IoT boards

7.1 Testing on-board Memory Storage

7.1.1 Testing the Storage in Arduino board

7.2.2 Testing the Storage in Raspberry pi board

7.2 Testing the On-board Power Supply

7.2.1 Testing the On-board Power Supply in Arduino board

7.2.2 Testing the On-board Power Supply in Raspberry pi board

Chapter 8 Data transfer between Devices

8.1 Testing the Communication between Devices

8.1.1 Test for Node and Gateway Software

8.1.2 Checking Communication Link Performance Matrix

8.2 Data Transfer from Gateway to Server

8.2.1 Checking Connectivity

8.2.2 Setting Connectivity Credentials

Unit 3. Organisational Processes and Standards

Chapter 9 Organisational Processes and Standards

9.1 Importance of organisational processes

9.2 Elements of organisational Process

9.2.1 Unity

9.2.2 Teamwork

9.2.3 Delegation of Work

9.3 Steps of Organisational Process

9.3.1 Analyse the Target

9.3.2 Create a Strategy

9.3.3 Categorise the Work

9.3.4 Assign and Give Authorization

9.3.5 Co-ordinate and Maintain Relationships

9.4 Common Organisational Structure

Chapter 10 Project Management concepts and applications

10.1 Project Management

10.1 Project Initiation

10.2 Project Planning

10.3 Project Execution

10.4 Project Monitoring and Controlling

10.5 Project Closing

10.2 Project Handling

10.3 IoT based Projects Examples

10.3.1 Project on Humidity and Temperature Sensing Device using IoT

10.3.2 Project on Air Pollution Measuring Device Using IoT

Chapter 11 Record Maintenance and Documentation

11.1 Record Maintenance

11.1.1 Storing of Records

11.1.2 Importance of Storing Records

11.1.3 Recording Performance

11.1.4 Methods to Record Performance

11.2 Documentation-Document Form, Document Processing

11.3 Role of a Telecom Technician (IoT Devices and systems) in Record Maintenance and documentation

Unit 1. Setup and Configuration of IoT Devices

The term IoT, or Internet of Things, refers to the technology that facilitates communication between devices and the cloud, as well as between the devices themselves. It is the the network of physical objects that are embedded with sensors, software, and other technologies for connecting and exchanging data with other devices and systems over the internet. Currently there are more than 20 billion connected IoT devices today.

IoT device management involves remotely registering, configuring, provisioning, maintaining and monitoring connected devices from a centralized platform that can access with an internet connection from anywhere on any device.

Arduino IDE and Raspberry Pi are the building blocks of IoT that helps to create projects. In this unit you will understand to setup and configure IoT Devices. It demonstrates to install and set up Arduino IDE and Raspberry Pi board, how to how to connect and configure different IoT devices, like sensors and control appliances.

Chapter 1. Installation of Arduino IDE and Raspberry Pi

Arduino is a microcontroller board that can run single tasks repeatedly unless additional hardware is added. The Raspberry Pi is a single-board computer with its powerful processor. With the Raspberry Pi, you can run a wide range of applications, including the popular Arduino IDE (integrated development environment). You can use it to program microcontrollers connected to Raspberry Pi computer via USB to create interactive electronic projects.

In this chapter you will learn to install Arduino IDE....

<https://docs.arduino.cc/software/ide-v2/tutorials/getting-started-ide-v2/>

1.1 Arduino

Arduino is open-source platform based on easy-to-use hardware and software designed to make electronics accessible to everyone, regardless of their expertise. It combines user-friendly hardware and software to empower individuals to bring their creative ideas to life through interactive electronic projects.

Arduino boards are able to read inputs – light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

At its core, Arduino consists of two main components – *hardware* and *software*. The hardware component is a programmable circuit board containing a microcontroller, which serves as the brain of the project. This microcontroller can interact with various sensors, motors, and other electronic components, enabling users to create a wide range of devices and gadgets.

Overview of the Arduino IDE

The Arduino Integrated Development Environment (IDE) is a software tool that simplifies the process of writing, compiling, and uploading code to the Arduino board. The IDE provides the user friendly platform for programming, making it easy for users to learn and experiment with coding.

The 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 Arduino IDE software is used to write and send code to Arduino boards. It works on different systems like Linux, Mac OS X, and Windows. The IDE supports programming languages like C and C++. In the Arduino language, the programs or code are called *sketches*. To upload a sketch created in the Arduino IDE software, it must connect to the Arduino board.

Writing Sketches

Programs written using Arduino Software (IDE) are called ***sketches***. These sketches are written in the text editor and are saved with the file extension ***.ino***. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

Versions of Arduino IDE

Arduino IDE 1 is the classic offline editor. The Arduino IDE 2 is an improvement of the classic IDE, with increased performance, improved user interface and many new features, such as autocompletion, a built-in debugger and syncing sketches with Arduino Cloud.

The Arduino IDE 2.0 is open source and its source code is hosted on GitHub. Arduino IDE 2.3.2 is the new major release of the Arduino IDE. It is faster and more powerful. In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

Arduino IDE 1.8.19 is the open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It can be used with any Arduino board.

The previous versions, Arduino IDE 1.8.18, Arduino 1.0.x and Arduino 1.5.x beta are also available for download.

Downloading Arduino IDE 2

It is possible to download and install the desired version of Arduino IDE for various operating systems such as Windows, Linux and Mac. So first download the desired version of Arduino IDE for the desired operating system from its official website <https://www.arduino.cc/en/software>. **Installation of Arduino IDE 2 on Windows**

To install the Arduino IDE 2 on a Windows computer, first download the Arduino IDE 2 from its official website. The software will get downloaded in the **Downloads** folder of the Windows as shown in Figure 1.1.

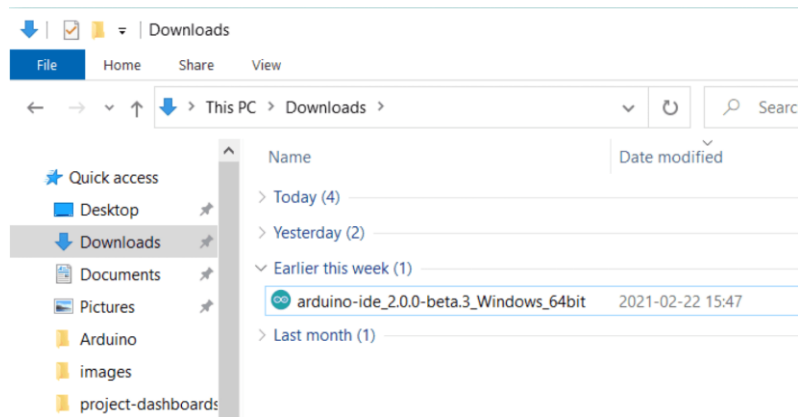


Fig. 1.1 Downloading Arduino IDE 2 on Windows

Once the download is complete, start to install the software. Practical Activity 1.1 illustrate the process of installation of *Arduino IDE 2 on Windows*. If the operating system shows a warning during the driver installation process, allow it to proceed.

Practical Activity 1.1. Demonstrate the Installation of Arduino IDE 2 in Windows OS

Material Required

Computer with the Arduino IDE 2 software downloaded in Windows, internet connectivity

Procedure

First download the Arduino IDE 2 for Windows operating systems from its official website <https://www.arduino.cc/en/software>.

After downloading the Arduino IDE 2 software downloaded in Windows, follow the following steps to install the Arduino IDE 2 under Windows.

Step 1. First open the folder where the *Arduino IDE 2* software downloaded in the computer. To install the software run the *Arduino IDE Setup*. The installation window will appear as shown in Figure 1.2. Click on **I Agree** button to proceed with Licence Agreement as shown in Figure 1.2.

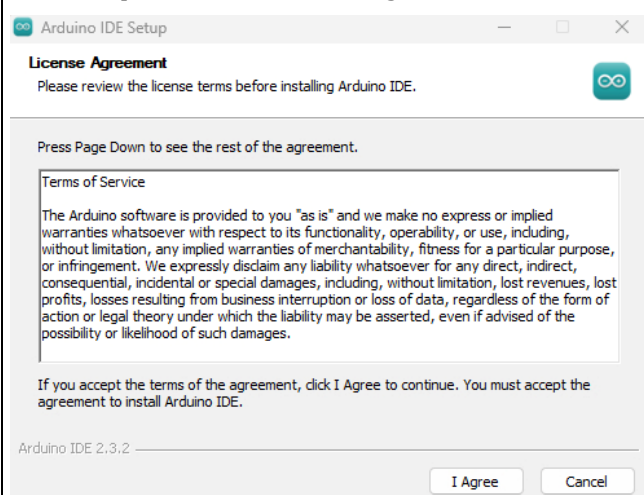


Fig. 1.2: Arduino IDE Setup – Licence Agreement

Step 2. Then proceed further to choose the installation option, as shown in Figure 1.3.

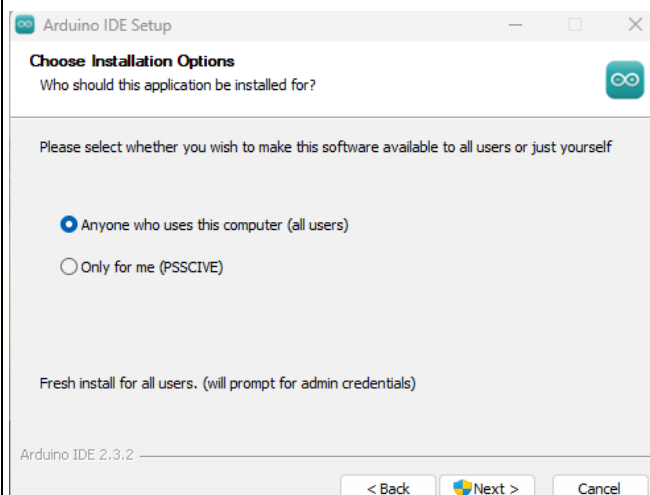


Fig. 1.3: Arduino IDE Setup – Installation Options

Step 3. The next step is to choose the location for installation as shown in Figure 1.4.

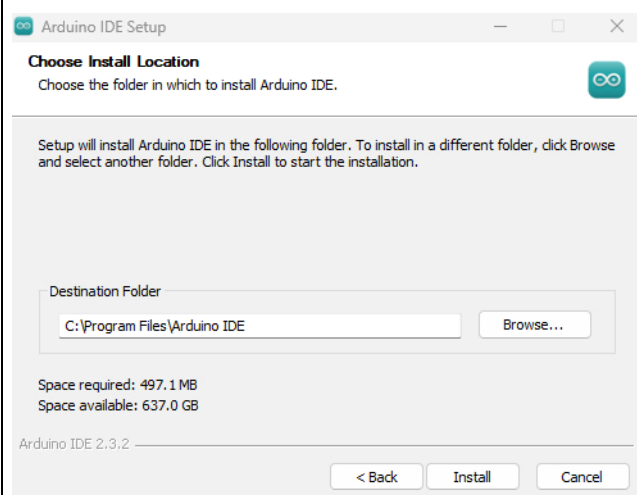


Fig. 1.4: Arduino IDE Setup – Installation Folder

Step 4. Now setup starts to install the software as shown in Figure 1.5 shows.

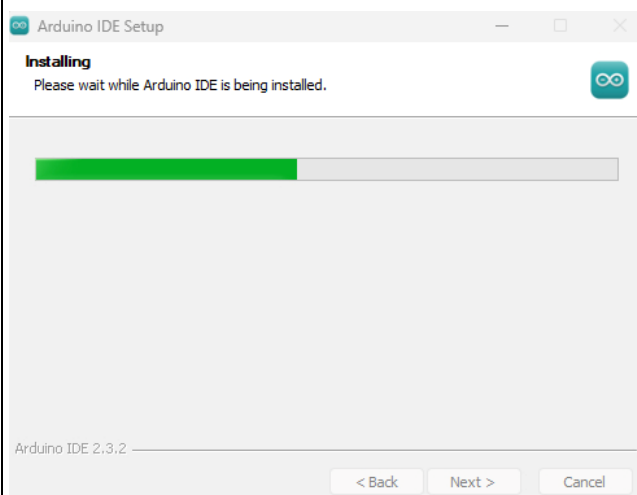


Fig.1.5: Installation progress window

Step 4. After finishing the installing process, click on the Finish button as shown in Figure 1.6 to complete the installation process.

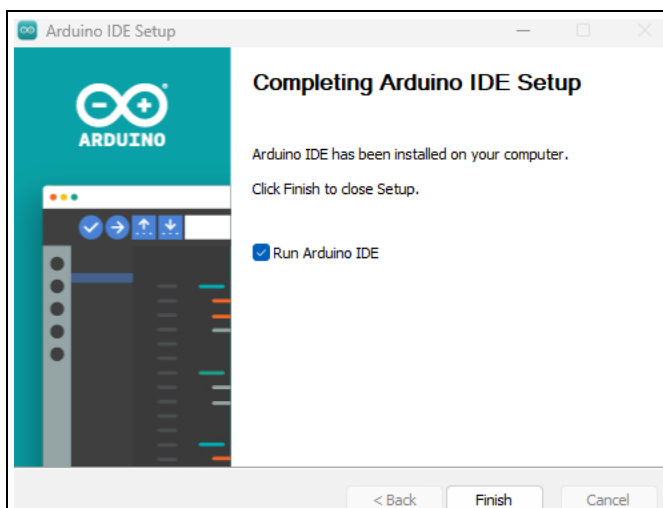


Fig. 1.6: Installation window

Step 5. After completing the installation processes, launch the Arduino IDE. The Arduino window will appear as shown in Figure 1.7.

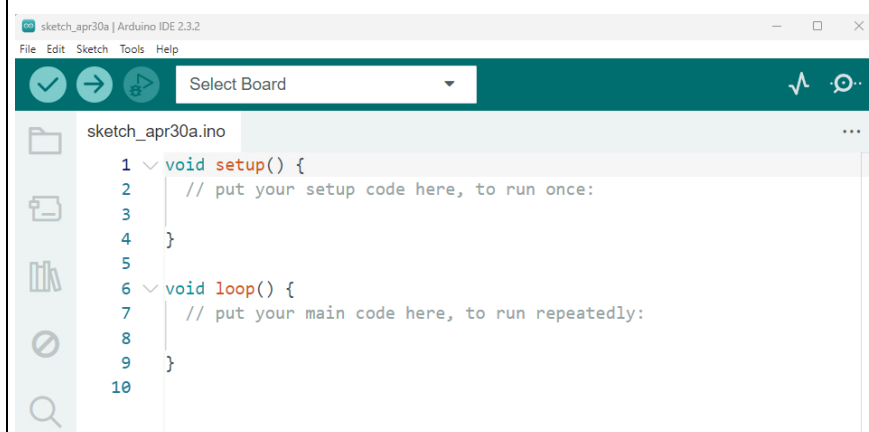


Fig. 1.7: Arduino IDE Window

Installing Arduino IDE 2 on Linux

The Practical Activity 1.2 illustrates to install Arduino IDE 2 on Ubuntu Linux.

Practical Activity 1.2. Demonstrate the Installation of Arduino IDE 2 in Ubuntu Linux

Material Required

Computer with Ubuntu Linux and internet connectivity

Procedure

Step 1. To install the Arduino IDE 2 on Linux, first download the **AppImage 64 bits (X86-64)** from the Arduino Software page. It is downloaded in the **Downloads** folder as shown in Figure 1.8.

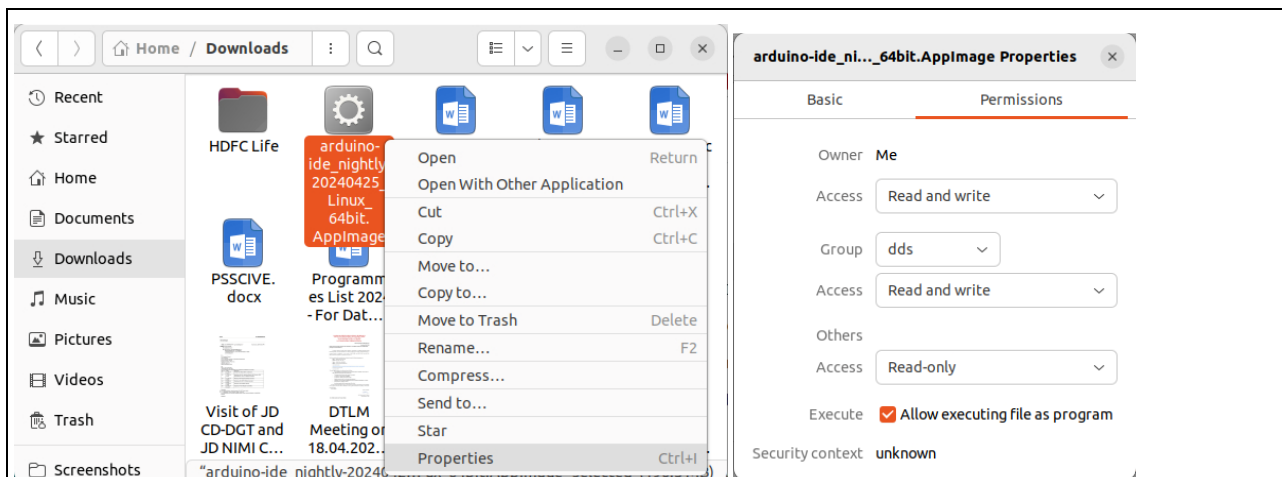


Fig. 1.8 : Downloaded AppImage 64 bits (X86-64) from the Arduino Software page.

Step 2. Allow the execute permission to this file. For that, right-click the file > choose **Properties** > select **Permissions** tab > tick the **Allow executing file as program** box.

Step 3. Next, double click the file to launch the Arduino IDE 2 on your Linux machine. In case you cannot run the AppImage file, make sure that FUSE is installed on your system.

Step 4. In Ubuntu Linux issue the following command to install FUSE.

```
sudo add-apt-repository universe
```

```
sudo apt install libfuse2
```

Step 5. You can now double click the file to launch the Arduino IDE 2 on your Linux machine.

Click on *Agree* button and proceed to install (Figure 1.9).

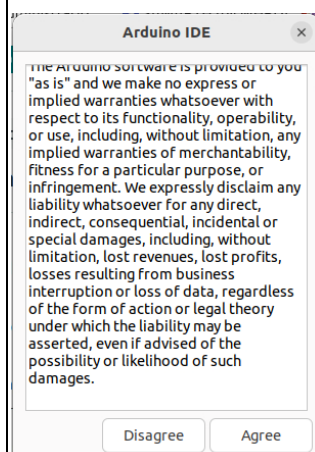


Fig. 1.9

After processing the installation files the Arduino IDE 2 will be launched as shown in Figure 1.10.

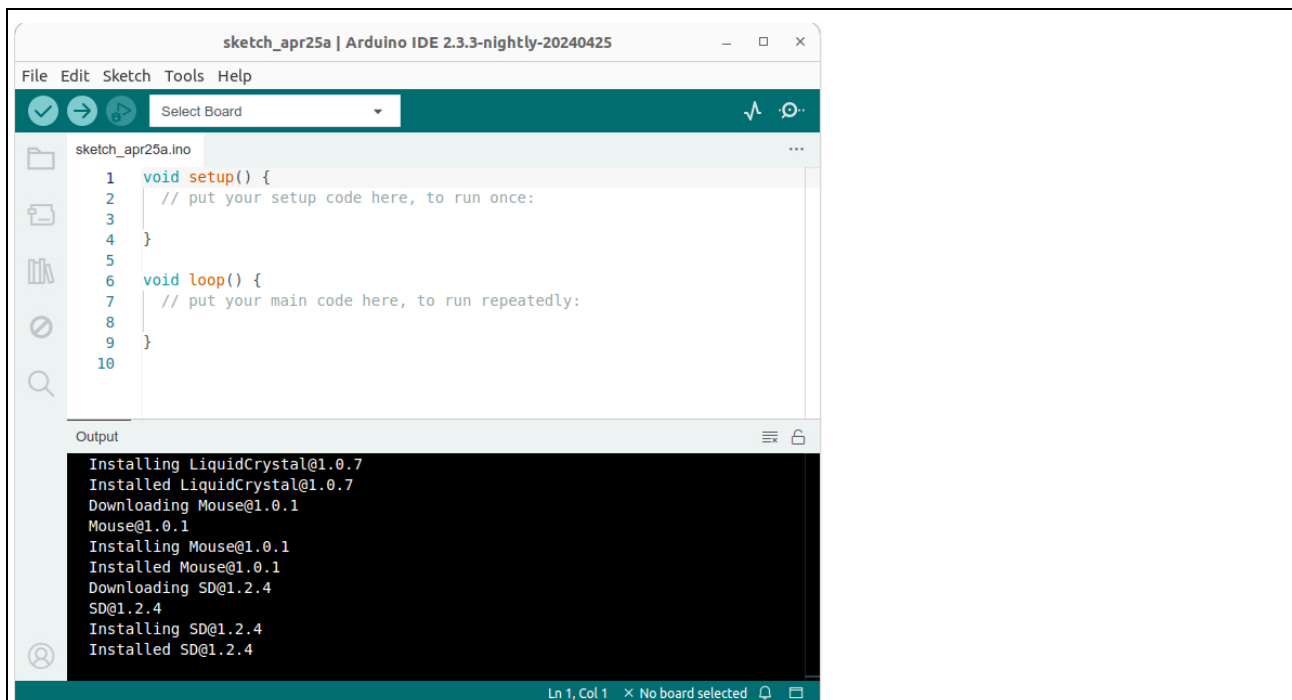


Fig. 1.10: Arduino IDE 2

Overview and Features of Arduino IDE 2

The Arduino IDE 2 features with a new sidebar, making the most commonly used tools more accessible. The different parts of this Arduino IDE comprising Menu Bar, Toolbar are shown in Figure 1.11.



Fig. 1.11 : Different parts of Arduino IDE 2

Verify/Upload – compile and upload the code to Arduino Board.

Select Board & Port – detected Arduino boards automatically show up here, along with the port number.

Sketchbook – here you will find all of your sketches locally stored on your computer. Additionally, you can sync with the Arduino Cloud, and also obtain your sketches from the online environment.

Boards Manager – browse through Arduino & third party packages that can be installed. For example, using a MKR WiFi 1010 board requires the Arduino SAMD Boards package installed.

Library Manager – browse through thousands of Arduino libraries, made by Arduino & its community.

Debugger – test and debug programs in real time.

Search – search for keywords in your code.

Open Serial Monitor – opens the Serial Monitor tool, as a new tab in the console.

Major components and Features of Arduino IDE 2

The Arduino IDE 2 comes with many features. It is possible to install libraries directly, sync sketches with Arduino Cloud, debug sketches and much more. The various components of Arduino IDE 2 are explained below.

Sketchbook

The Arduino program code is called as sketch and the program files are stored in a sketchbook. Arduino sketches are saved as **.ino** files, and must be stored in a folder of the exact name. For example, a sketch named `my_sketch.ino` must be stored in a folder named `my_sketch`. Typically, the sketches are saved in a folder named **Arduino** in **Documents** folder. To access the sketchbook, click on the **folder icon** located in the sidebar as shown in Figure 1.12.

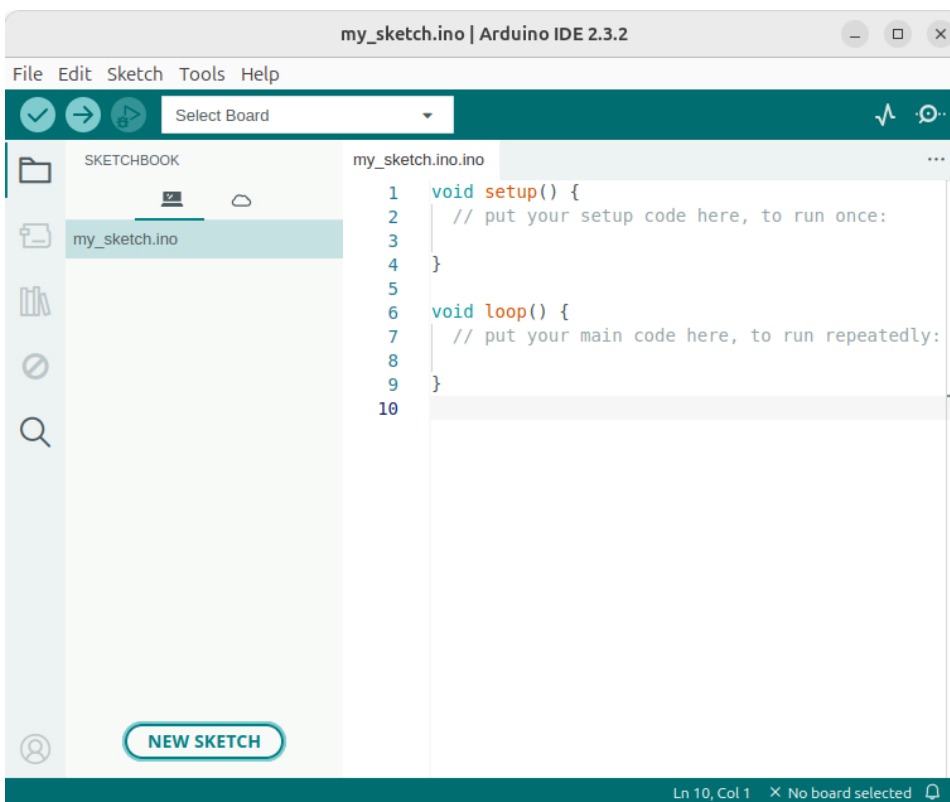


Fig. 1.12: Arduino Sketchbook

Board Manager

The Boards Manager, allows to browse and install board packages. A board package contains the "instructions" for compiling code to the boards that are included in the board package. There are several Arduino board packages available, such as **avr**, **samd**, **megaavr** and more.

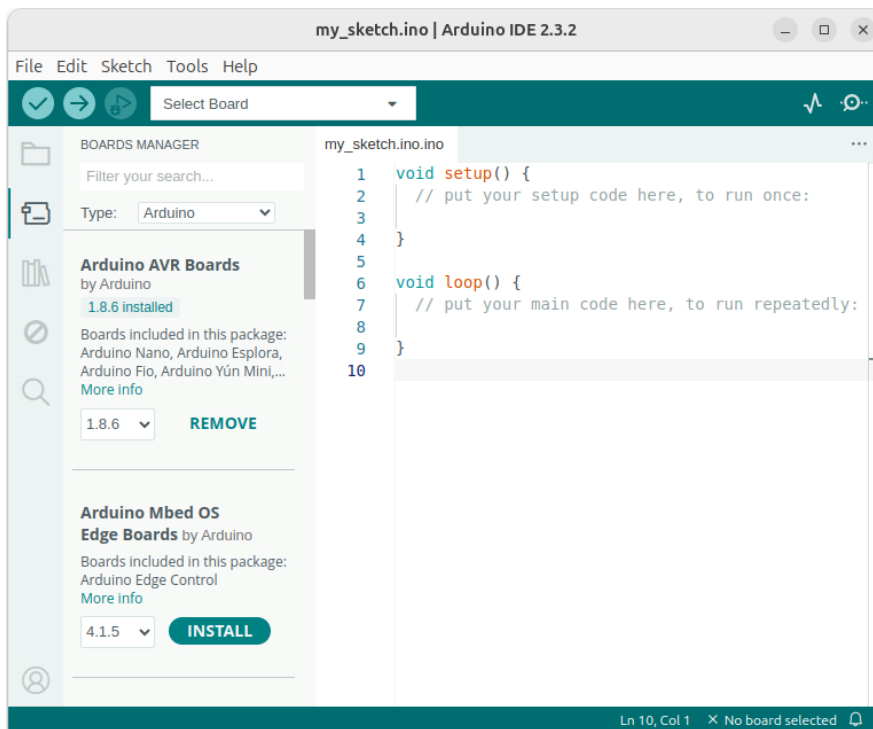


Fig. 1.13: Boards Manager

Library Manager

The Library Manager, allows to browse and install thousands of libraries. Libraries are extensions of the Arduino API, and makes it easier to. For example control a servo motor, read specific sensors, or use a Wi-Fi module.



Fig. 1.14: Library Manager

Serial Monitor

The Serial Monitor is a tool that allows to view data streaming from board. For example the `Serial.print()` command. This tool is located in a separate window, but is now integrated with the editor. This makes it easy to have multiple instances running at the same time on your computer.

Serial Plotter

The Serial Plotter tool is great for visualizing data using graphs, and to monitor for example peaks in voltage. You can monitor several variables simultaneously, with options to enable only certain types.



Fig. 1.15: Serial Plotter

Autocompletion

This is the important feature of Aurdino IDE 2. When writing code, this is useful to understand more about the elements of the Arduino API. You always need to select your board for autocompletion to work.

Remote Sketchbook

The Remote Sketchbook feature lets you sync sketches from your Arduino Cloud sketchbook with your local computer. To enable this feature, login to your Arduino Cloud account.

Example Sketches

An important part of the Arduino Documentation are the example sketches that come bundled with libraries. They will show examples of the functions used in practice, illustrating the intended use and features of a library.

Libraries that come bundled as a part of a boards package may also include libraries, and those libraries include example sketches.

To open the example sketches bundled in either the libraries you have installed manually or that come bundled in board packages, navigate to **File > Examples** and search the library in the list that appears.

From here, you can navigate to **File > Examples > Digital > BlinkWithoutDelay >** and upload the sketch to your board.

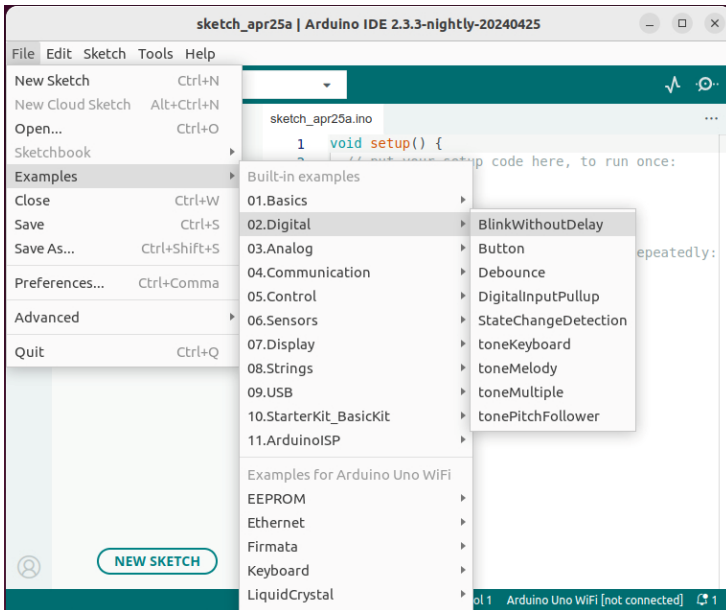


Fig. 1.16: Example sketches



Fig. 1.17: Example BlinkwithoutDelay

Debugging

The debugger tool is used to test and **debug** programs, hence the name. It can be used to navigate through execution of a program in a controlled manner.

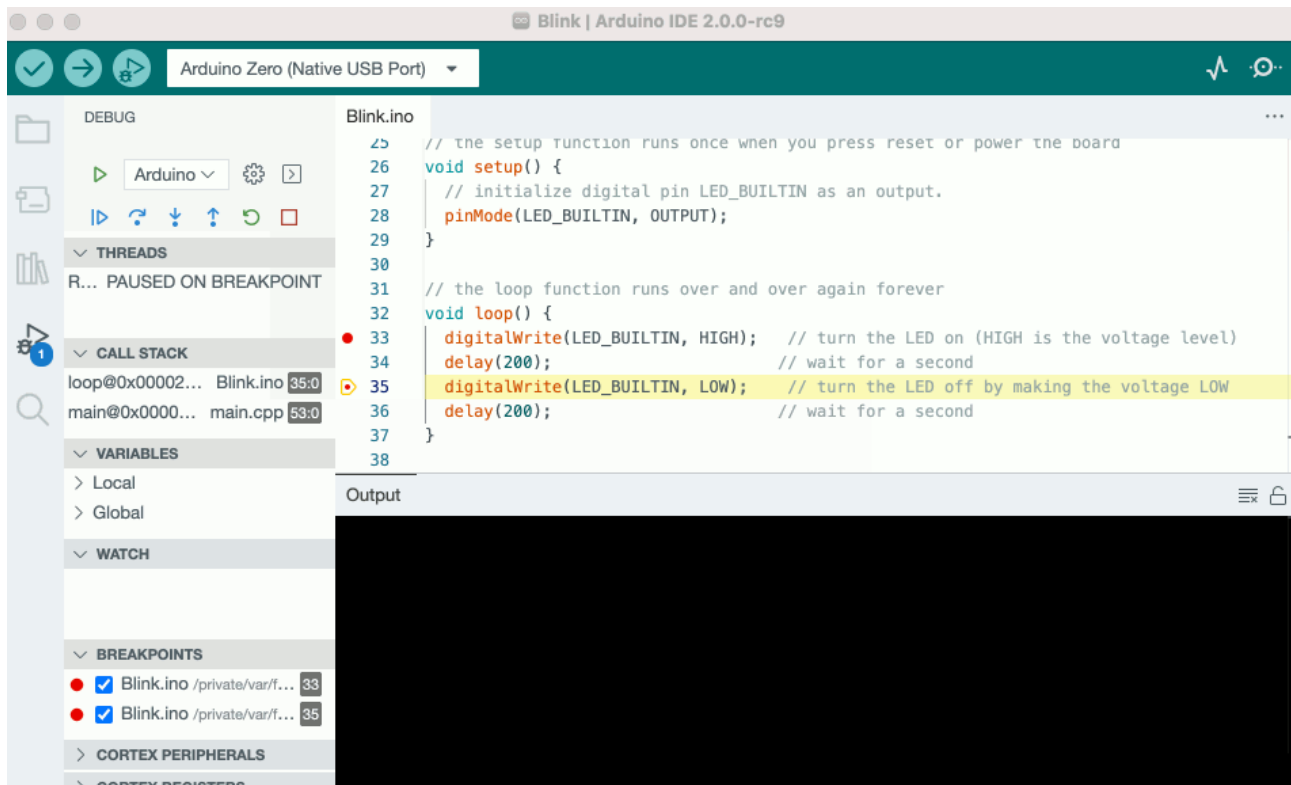


Fig. 1.18: Debugging

UPLOADING A SKETCH WITH THE ARDUINO IDE 2

In the Arduino environment, to execute the Arduino program or *sketches* has to be uploaded to Arduino boards, then compile and execute the code. Let us see how to select a board connected to your computer, and how to upload a sketch to that board, using the Arduino IDE 2.

There are two main tools when uploading a sketch to a board: **verify** and **upload**. The verify tool simply goes through your sketch, checks for errors and compiles it. The upload tool upload the sketches to the board after finishing the compilation of the code.

A good practice is to use the verifying tool before attempting to upload anything. This is a quick way of spotting any errors in the code and to fix them before actually uploading the code.

The process of uploading a Sketch is illustrated in Practical Activity 1.3.

Practical Activity 1.3. Demonstrate to upload a Sketch

Material Required

Computer with Arduino IDE installed and internet connectivity.

Procedure

Step 1. Open the Arduino IDE 2.

Step 2. Observe the editor toolbar at the top as shown in Figure 1.19. At the very left, there is a **checkmark** and an **arrow pointing right**. The checkmark is used to **verify**, and the arrow is used to **upload**.

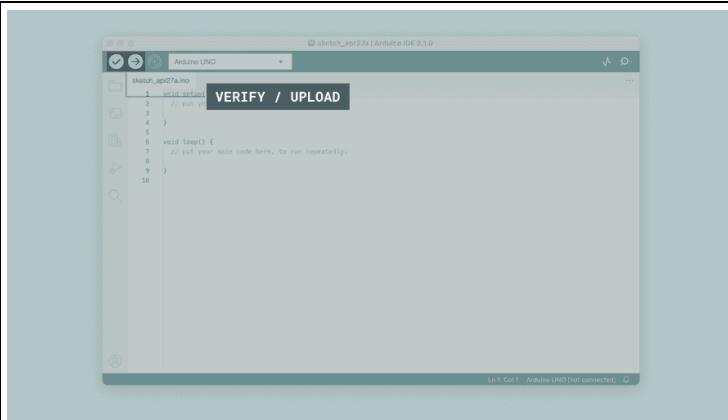


Fig. 1.19 Verify and Upload buttons.

Step 3. Before verifying and uploading the code, always check the selected board from drop down menu as shown in Figure 1.20. In most cases, it will display Arduino boards that is connected to your computer. If your board is not automatically detected, you can either press **“Select other board and port...”** in the drop-down and follow the instructions, or go to **Tools > Board** and **Tools > Port** in the toolbar menu to select the board and port manually.

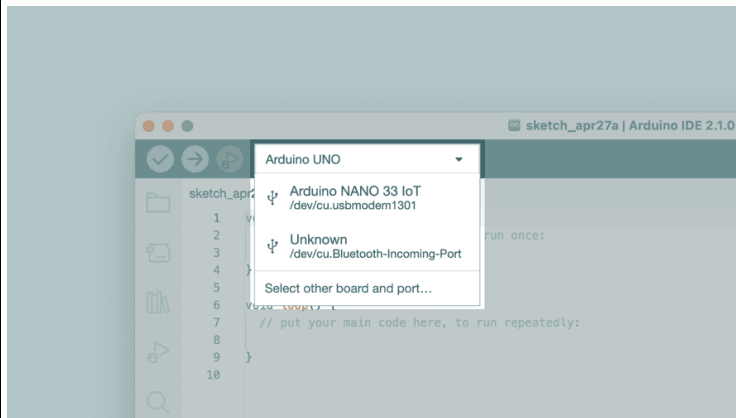


Fig. 1.20 Selecting the board.

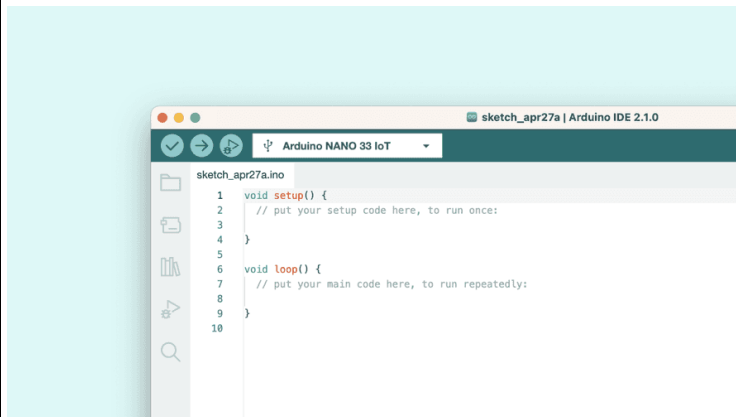


Fig. 1.21 Showing the selected board Arduino NANO 33 IoT

Step 4. So connect your board to the computer and select it from the drop down menu. You'll know that there is a connection to the board when the board name appears in **bold**.

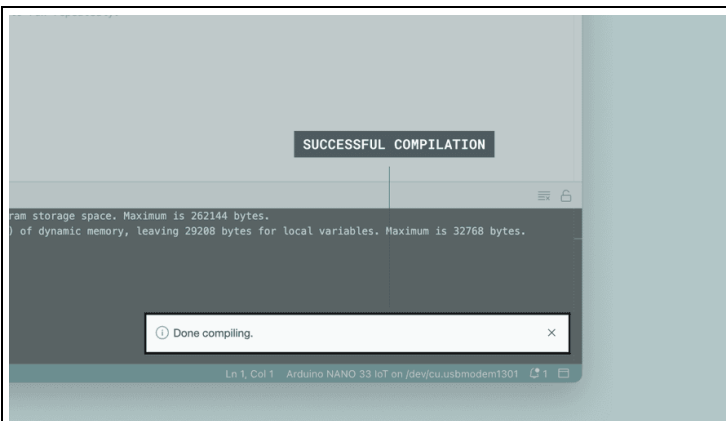


Fig. 1.22 Successful compilation printed in the console.

Step 5. Click on the verify tool (checkmark). Since we are verifying an empty sketch, we can be sure it is going to compile. After a few seconds, we can see the result of the action in the console (black box in the bottom).

Step 6. With the board selected, we are good to go! Click on the **upload** button, and it will start uploading the sketch to the board.

Step 7. When it is finished, a notification pops up in the bottom right of your IDE window. Of course, sometimes there are some complications when uploading, and these errors will be listed here as well.

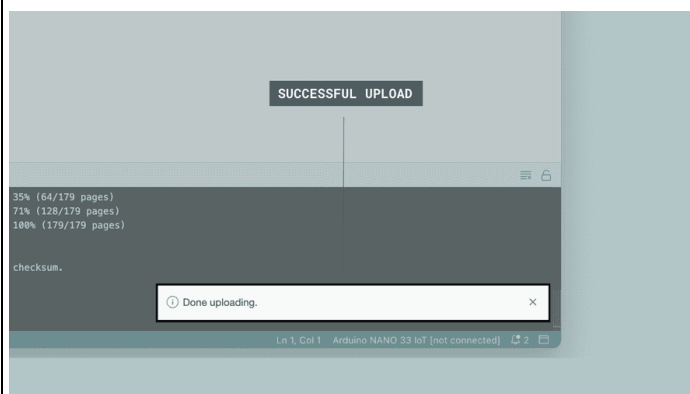


Fig. 1.23 Successful upload printed in the console.

You have now uploaded a sketch to your Arduino board!

The Board Manager

The board manager is a tool that is used to install different board packages on your local computer to use Arduino boards.

Board package

A board package is written and designed for specific microcontrollers. Arduino offers several different types of boards, and these boards may also have different types of microcontrollers. While different microcontrollers accomplish tasks in similar ways, the way code is compiled, pins are mapped, and what features are available is tailor-made to the silicon itself, meaning that this will also be microcontroller-specific.

A board package act as a layer between all of this microcontroller-specific jargon. The board package translates it into the Arduino API so that you can program any of the microcontrollers in the Arduino ecosystem in the same way.

For example, an Arduino UNO has an **ATmega328P**, which uses the **AVR package**, while an Arduino Nano 33 IoT has a **SAMD21** microcontroller, where we need to use the **SAMD package**. However, regardless of what microcontroller is on the board we are using, `digitalWrite(LED_BUILTIN, HIGH)` will turn on the built-in LED, and `analogRead(A0)` will read the analog pin 0 and check for a voltage.

Selecting the Board Package to Install

When you plug in a board in to your computer through USB port, and it has not the appropriate board package installed, the IDE will automatically prompt to install it. There are various boards and the appropriate board

packages are designed for that boards. The detail informaton about the boards and board packages are given below.

AVR boards package

The **AVR boards package** is based on the **Arduino AVR Core**, and includes the following boards:

- UNO R3
- UNO R3 SMD
- UNO Mini Limited Edition
- Leonardo
- Micro
- Nano
- Mega 2560

The **AVR board package** comes pre-installed when you download the Arduino IDE, means you need not to install it. You can, however, still find it in the board manager if you want to change what version of the board package you have installed. For source code and reporting issues, visit the official GitHub repository of **ArduinoCore-avr**.

MegaAVR Boards package

The **MegaAVR Boards package** is based on the **Arduino AVR Core**, and includes the following boards:

- UNO WiFi Rev2
- Nano Every

You can install this package in the board manager by searching for "**Arduino MegaAVR Boards**", as illustrated in Practical Activity 1.4. For source code and reporting issues, visit the official GitHub repository of **ArduinoCore-megaavr**.

UNO R4 Boards package

The **UNO R4 Boards package** is based on the **Arduino Renesas Core**, and includes the following boards:

- UNO R4 Minima
- UNO R4 WiFi

You can install this package in the board manager by searching for "**Arduino UNO R4 Boards**", as illustrated in Practical Activity 1.4. For source code and reporting issues, visit the official GitHub repository of **ArduinoCore-renesas**.

SAM Boards package

The **SAM Boards package** is based on the **Arduino SAM Core**, and includes the boards: Due.

You can install this package in the board manager by searching for "**Arduino SAM Boards**", as illustrated in Practical Activity 1.4. For source code and reporting issues, visit the official GitHub repository of **ArduinoCore-sam**.

SAMD Boards package

The **SAMD Boards package** is based on the **Arduino SAMD Core**, and includes the following boards:

- Zero
- Nano 33 IoT
- MKR 1000 WiFi
- MKR Zero
- MKR WiFi 1010
- MKR FOX 1200
- MKR WAN 1300
- MKR WAN 1310
- MKR GSM 1400
- MKR NB 1500

- MKR Vidor 4000

You can install this package in the board manager by searching for "**Arduino SAMD Boards**", as illustrated in Practical Activity 1.4. For source code and reporting issues, visit the official GitHub repository of *ArduinoCore-samd*.

Mbed OS GIGA Boards

The **Mbed OS GIGA Boards** package is based on the **Arduino Mbed Core**, and includes the following boards:

- GIGA R1 WiFi

You can install this package in the board manager by searching for "**Arduino Mbed OS GIGA Boards**", as illustrated in Practical Activity 1.4. For source code and reporting issues, visit the official GitHub repository of *ArduinoCore-mbed*.

For source code and reporting issues, please visit the official GitHub repository:

- ArduinoCore-mbed

Mbed OS Nano Boards

The **Mbed OS Nano Boards** package is based on the **Arduino Mbed Core**, and includes the following boards:

- Nano RP2040 Connect
- Nano 33 BLE
- Nano 33 BLE Rev2
- Nano 33 BLE Sense
- Nano 33 BLE Sense Rev2

You can install this package in the board manager by searching for "**Arduino Mbed OS Nano Boards**", as illustrated in Practical Activity 1.4. For source code and reporting issues, visit the official GitHub repository of *ArduinoCore-mbed*.

Mbed OS Portenta Boards

The **Mbed OS Portenta Boards** package is based on the **Arduino Mbed Core**, and includes the following boards:

- Portenta H7
- Portenta H7-lite
- Portenta H7-lite-connected
- Portenta X8

You can install this package in the board manager by searching for "**Arduino Mbed OS Portenta Boards**", as illustrated in Practical Activity 1.4. For source code and reporting issues, visit the official GitHub repository of *ArduinoCore-mbed*.

Mbed OS Nicla Boards

The **Mbed OS Nicla Boards** package is based on the **Arduino Mbed Core**, and includes the following boards:

- Nicla Sense ME
- Nicla Vision
- Nicla Voice

You can install this package in the board manager by searching for "**Arduino Mbed OS Nicla Boards**", as illustrated in Practical Activity 1.4. For source code and reporting issues, visit the official GitHub repository of *ArduinoCore-mbed*.

Mbed OS Edge boards

The **Mbed OS Edge Boards** package is based on the **Arduino Mbed Core**, and includes the following boards:

- Edge Control

You can install this package in the board manager by searching for "**Arduino Mbed OS Edge Boards**", as illustrated in Practical Activity 1.4. For source code and reporting issues, visit the official GitHub repository of *ArduinoCore-mbed*.

Renesas Portenta Boards

The **Renesas Portenta Boards** package is based on the **Arduino Renesas Core**, and includes the following boards:

- Portenta C33

You can install this package in the board manager by searching for "**Arduino Renesas Portenta Boards**", as illustrated in Practical Activity 1.4. For source code and reporting issues, visit the official GitHub repository of **ArduinoCore-renesas**.

ESP32 Boards

The **ESP32 Boards** package is based on the **Arduino-esp32 Core**, and includes the following boards:

- Nano ESP32

You can install this package in the board manager by searching for "**Arduino ESP32 Boards**", as illustrated in Practical Activity 1.2. For source code and reporting issues, visit the official GitHub repository of **ArduinoCore-esp32**.

Installing a Board Package in the Arduino IDE 2

To use a specific board, it is essential to install a specific board package. To install the package for the board, your computer should be ready with Arduino IDE 2 installed. The Practical Activity 1.4 illustrates to install the appropriate package for the board.

Practical Activity 1.4. Demonstrate to install a Board Package in the Arduino IDE 2

Material Required

A computer with Arduino IDE 2 installed and internet connection.

Procedure

Step 1. Open the Arduino IDE 2.

Step 2. With the editor open, let's take a look at the left column. Here, we can see a couple of icons. Let's click the on the **Arduino board** icon.

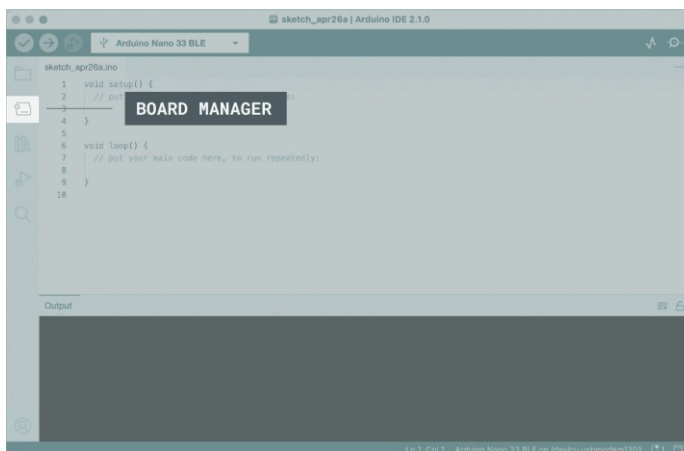


Fig. 1.24

Step 3. A list will now appear of all available packages. Now let's say we are using an **Nano 33 BLE** board, and we want to install the board package. Simply enter the name in the search field, and the right package (Mbed OS Nano) will appear, where the Nano 33 BLE features in the description. Click on the **"INSTALL"** button.

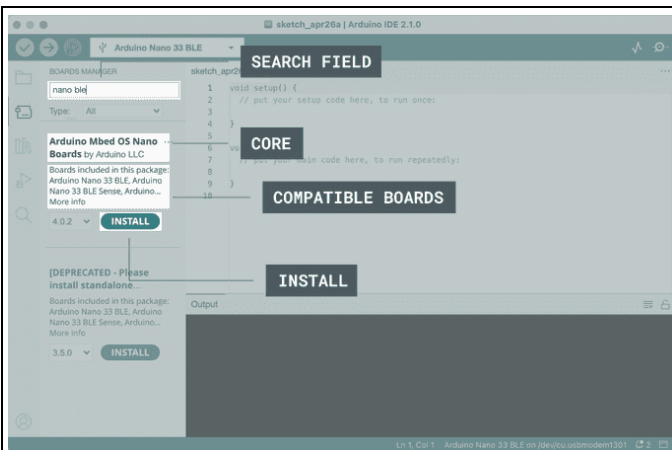


Fig. 1.25

Step 4. This will begin the installation process, which usually take few moments.



Fig. 1.26

Step 5. When it is finished, we can take a look at the core in the boards manager column, where it should say "INSTALLED", as well as noting which version you have installed on your machine.

You have now successfully downloaded and installed a board package on your machine, and you can start using your Arduino board.

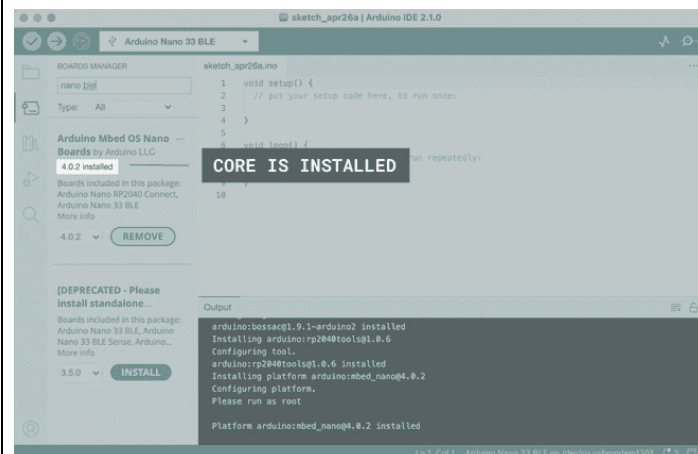


Fig. 1.25

Changing version

In some specific use cases it may require to downgrade the version of a board package that you have on your machine. The Arduino IDE has built in support to change the versions with a breeze. The process to change the installed version of a boards package is illustrated in Practical Activity 1.3.

Practical Activity 1.5. Demonstrate to change the installed version of a boards package

Material Required

A computer with Arduino IDE 2 installed with board package and internet connection.

Procedure

Step 1. Open the Board Manager.



Fig. 1.26

Step 2. Search for the name of the package you want to change.



Fig. 1.27

Step 3. Select the desired version in the drop down menu.



Fig. 1.28

Step 4. Press "INSTALL".

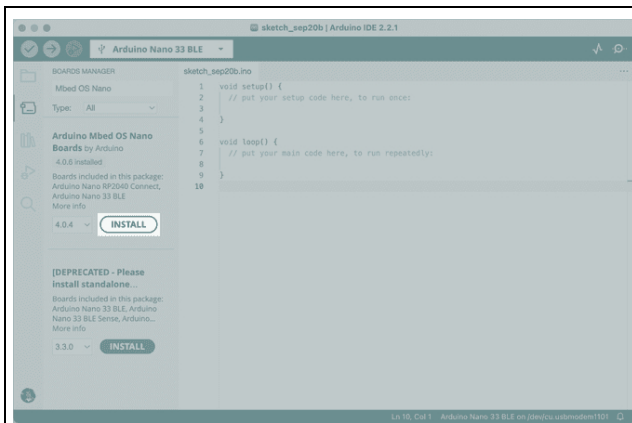


Fig. 1.29

Step 5. To make sure that the process was successful, cross check what version is installed by looking at this little tag right underneath the package name.

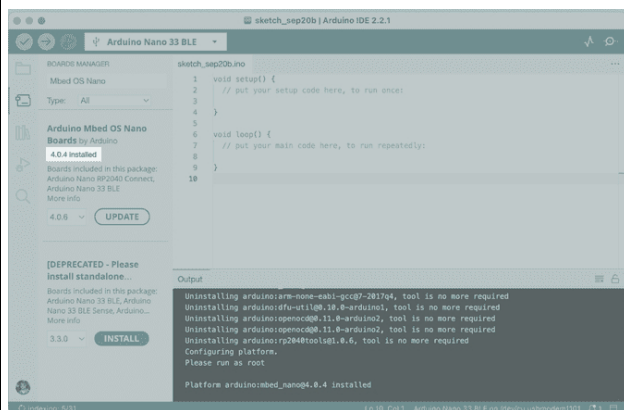


Fig. 1.30

Working with Arduino Board

Open Arduino Environment Software and Select the Arduino board type under **Tools > Boards > Arduino UNO**. The screenshot as shown in Figure 1.10 (a) shows the identification of the board. Selected ESP 8266 board in Arduino IDE is shown in Figure 1.10 (b).

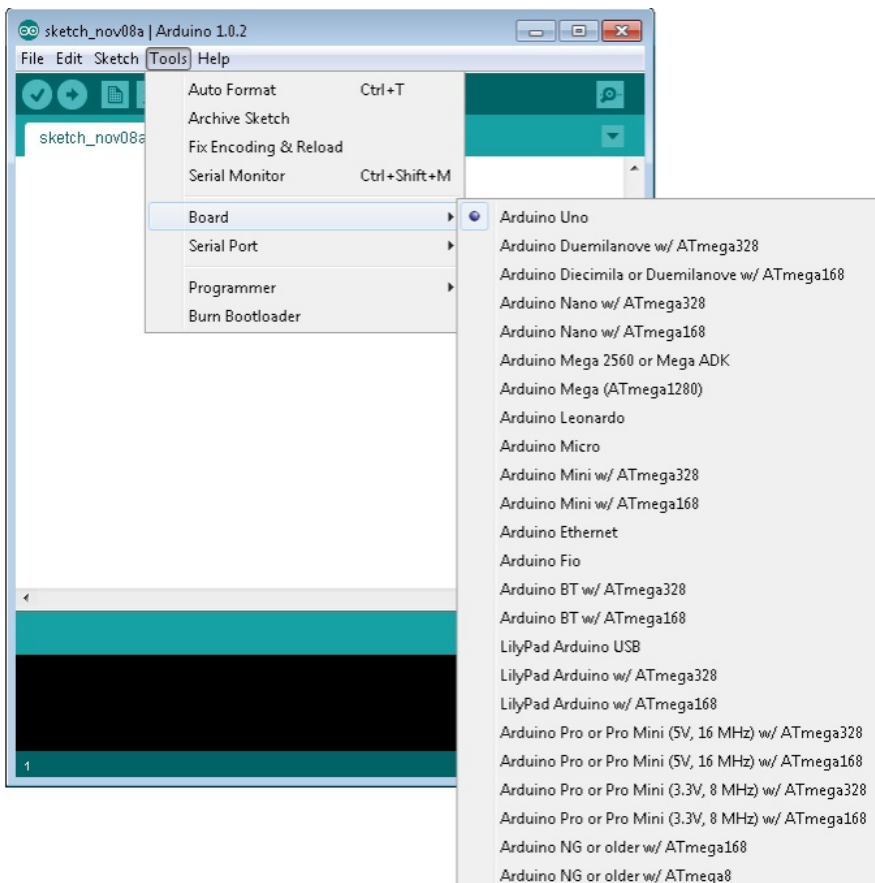


Fig.1.31 (a) : Identifying the board

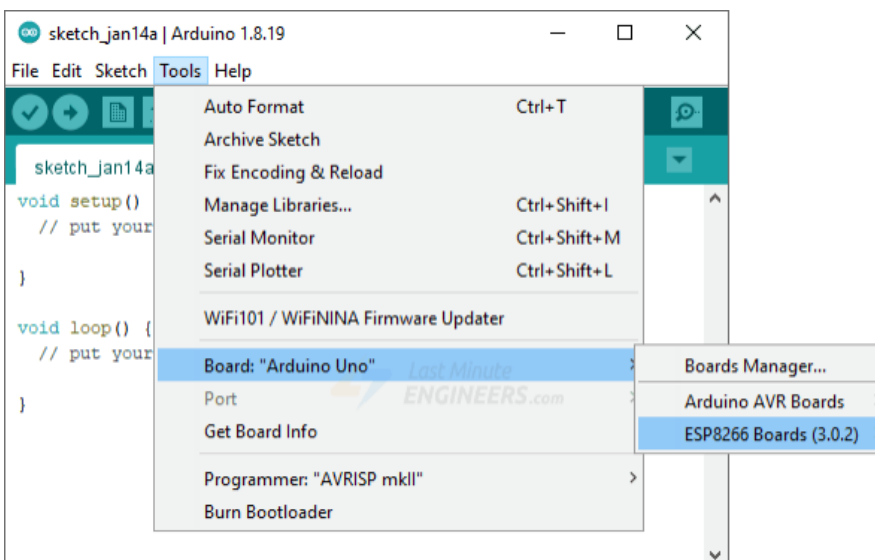


Fig. 1.31 (b) : Selecting ESP 8266 board

Then, choose the serial port under **Tools** > **Serial Port** > **COM 4** (COM # depending on what COM port is free during setup) as shown in Figure 1.32.

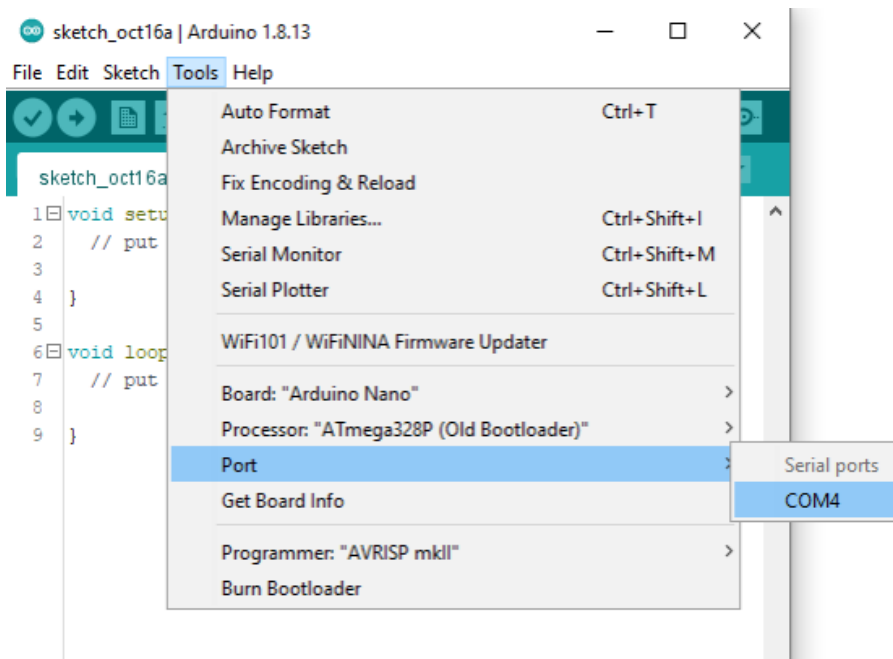


Fig. 1.32 : Selecting the serial port

Now open an example code by clicking **File > Examples > Basics > Blink** as shown in Figure 1.33.

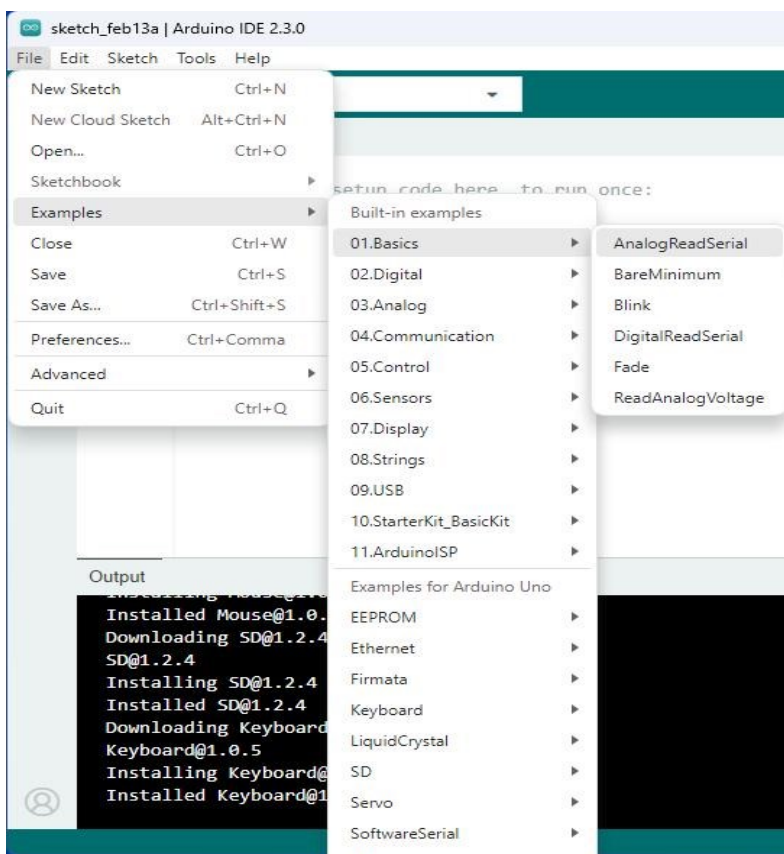


Fig. 1.33 : Opening a code

A screen will appear as shown in Figure 1.34.

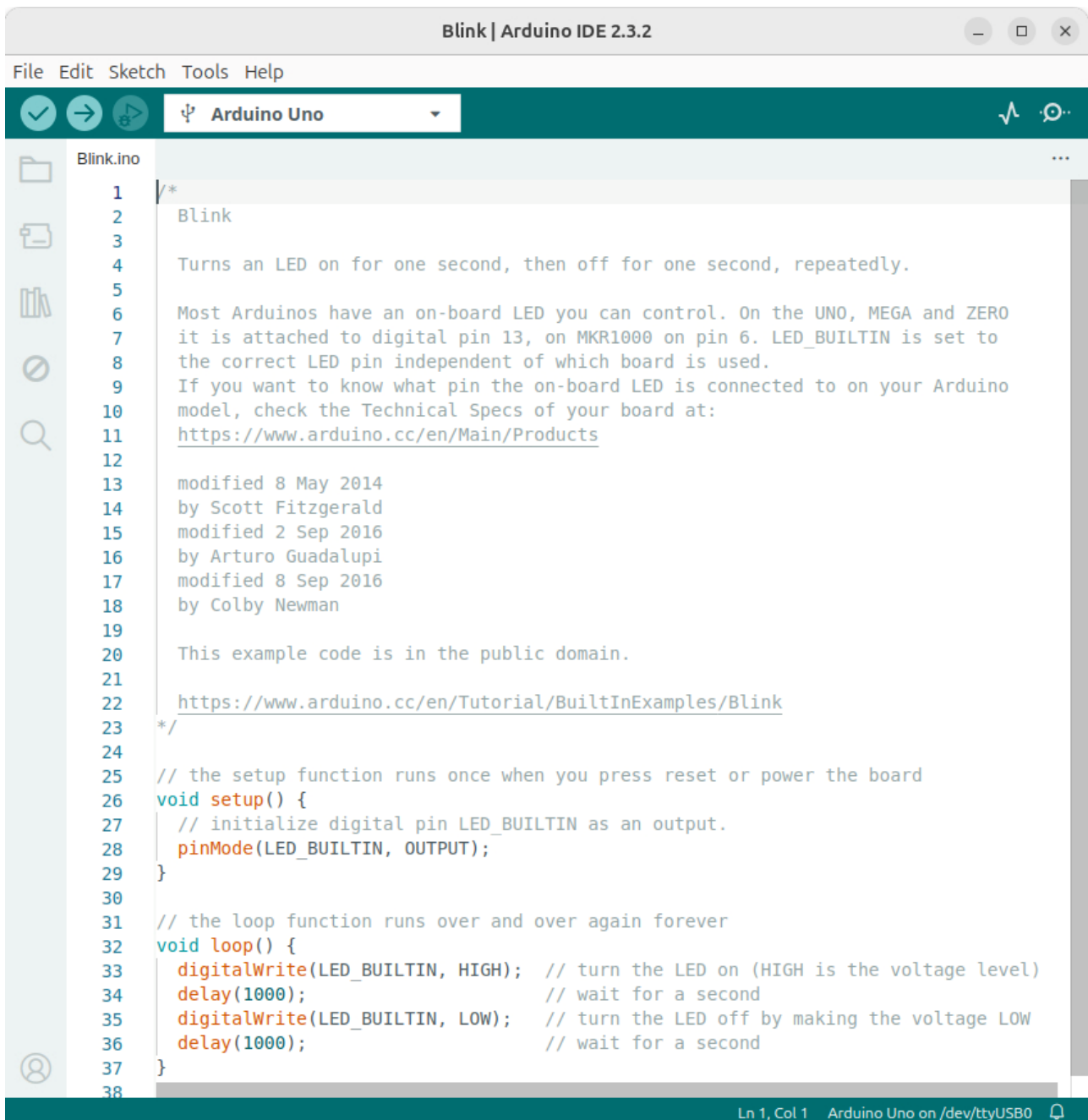


Fig. 1.34 : Screenshot of the code Blink.ino

Let's see the code of Blink . Ino in Arduino software.

```

Blink.Ino
/*
This is a comment at the top of a program, it will not be recognized as code.
Very good to add an explanation of what your code does here.

This sketch shows how to read a vaLue from a sensor connected to pin A1, print
it out in the Serial Monitor, and turn on an LED connected to pin number 2 if a
conditional is met.
*/

int sensorPin = A1; //define pin A1 (analog pin)
int ledPin = 2; //define pin 2 (digital pin)
int sensorValue; //create variable for storing readings

```

```

//void setup is for configurations on start up
void setup()
{
  Serial.begin(9600); //initializes serial communication
  pinMode(ledPin, OUTPUT); //define ledPin as an output
}

void loop()
{
  sensorValue = analogRead(sensorPin); // do a sensor read
  Serial.print("Sensor value is: "); //print a message to
  Serial.println(sensorValue); //print the value to the serial monitor

  //check if sensorValue is below 200
  if(sensorValue < 200)
  {
    digitalWrite(ledPin, HIGH); //if it is, turn on the LED
  }

  //if sensorValue is above 200, turn off the LED
  else
  {
    digitalWrite(ledPin, LOW);
  }
}

```

Then, click on the upload button to upload the code to Arduino. Wait till the “*Done Uploading*” message status appears. Figure 1.35 shows a screenshot for uploading code.

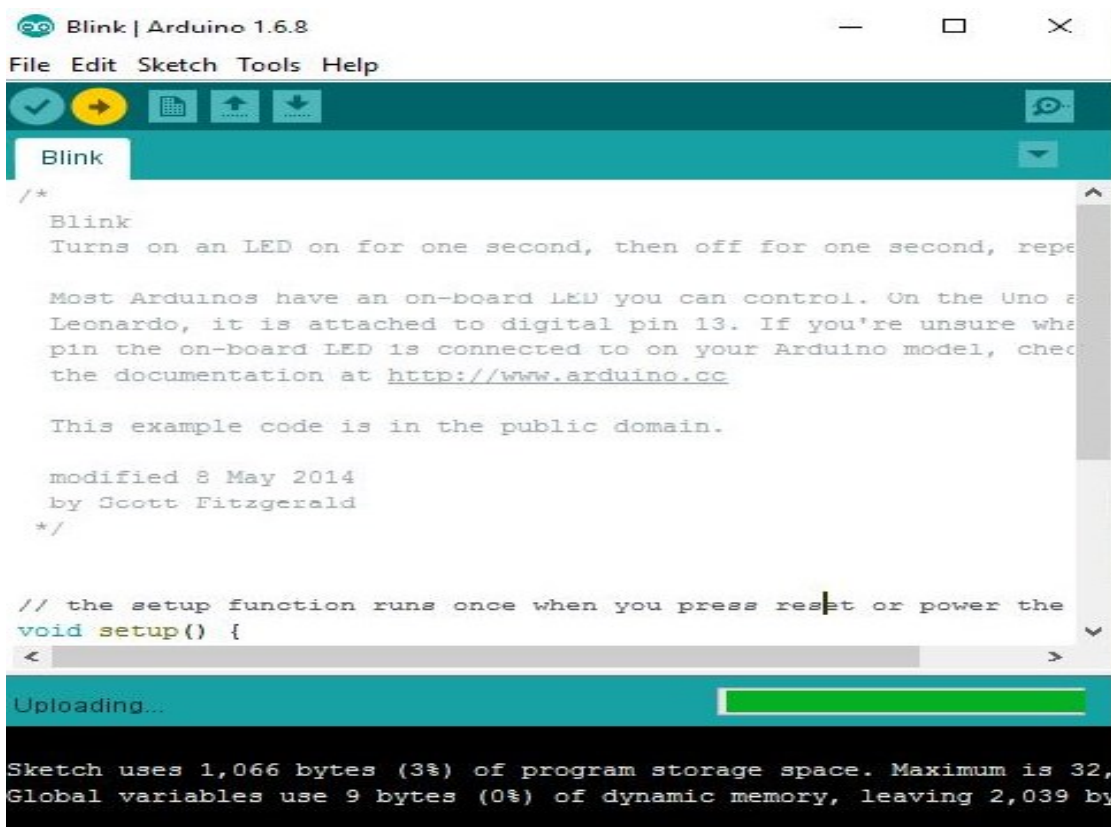


Fig. 1.35: Uploading a code

```

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

Most Arduinos have an on-board LED you can control. On the Uno a Leonardo, it is attached to digital pin 13. If you're unsure what pin the on-board LED is connected to on your Arduino model, check the documentation at http://www.arduino.cc

This example code is in the public domain.

modified 8 May 2014
by Scott Fitzgerald
*/

// the setup function runs once when you press reset or power the
void setup() {
  <
  Uploading.....
  Sketch uses 1,066 bytes (3%) of program storage space. Maximum is 32, Global variables use 9 bytes (0%) of
  dynamic memory, leaving 2,039 by
  >

```

>>>>>>

1.1.2 Connection of Arduino Board to Computer for Data Transfer

To establish a connection between an Arduino board and a computer, the following software and hardware components are required:

Hardware components required are *Arduino Uno Board* and *USB AB Cable*.

The required software are : Any one operating System: *Windows 10/8/7/XP, macOS*, or *Linux* and *Arduino IDE* (Software for compiling Arduino programs).

For data transfer, a laptop or computer must be linked to the microcontroller board. Installing the necessary software on the PC or laptop is essential for establishing this connection. The software program on the microcontroller retrieves sensor data, which is then transmitted to the PC via the USB cable. During this process, data is updated every second. Setting up the Arduino's hardware is necessary when using it for the first time.

Practical activity 1.1. Demonstrate to Connect Arduino Board to Computer for uploading the sketch (program).

Material Required

Arduino Board, USB B cable, Computer/laptop with Arduino IDE

Procedure

Step 1. Get all the equipment ready. Now connect the narrow end of the USB AB cable to Arduino and the other end of the cable to the USB port of the PC. This USB B cable will help to upload the program from the PC to the Arduino board. It is depicted in Figure 1.15.



Fig. 1.15: USB cable connection with PC

Step 2. Once the Arduino board is plugged-in, a LED will start blinking which indicates that the Arduino board is functional and it is connected to the PC. To check if the computer and the Arduino are communicating, The Figure 1.16 and Figure 1.17 shows connection of the USB plug to the board.

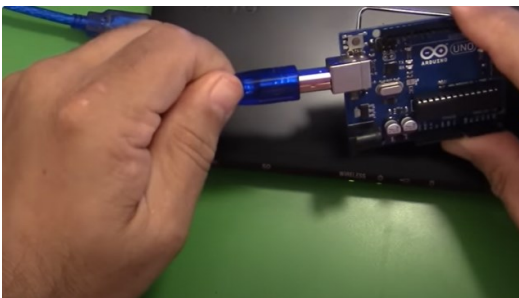


Fig.1.16: Connecting the USB plug to the board



Fig.1.17: LED blinking on the Arduino board after connecting the USB cable

Step 3. Check the blinking of the LED. This LED, called an onboard LED, is used to check the functioning of the board whether it is working or not. Figure 1.17 shows the complete set up with blinking LED.



Fig. 1.18: Arduino board Set up

1.2 Raspberry Pi

The Raspberry Pi board is a credit card-sized single-board computer, and is a revolutionary tool for DIY electronics and programming. It is developed by the Raspberry Pi Foundation. It offers an affordable and compact solution for a wide range of projects. With its versatile GPIO pins and powerful processor, it has become a go to choose for enthusiasts exploring computing, coding, and electronics. Various parts of the Raspberry Pi board are shown in Figure 1.19.

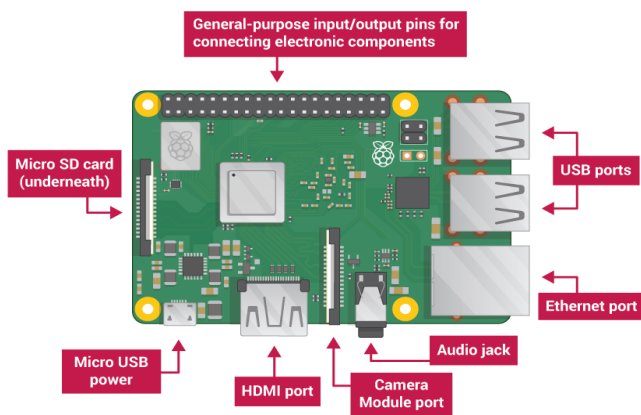


Fig. 1.19: Raspberry Pi Board

The various components of board are as follows:

USB ports — It is Universal serial port, where you can connect device such as hardware, mouse as well as attach user flash drive.

SD card slot — This is slot to insert SD card, in which the operating system software and data files are stored.

Ethernet port — This is the network port. Raspberry Pi board can connect to the network using a cable through this port. Raspberry Pi can also connect to a network via wireless LAN.

Audio jack — Headphones or speakers can be connected through this.

HDMI port — This port is used to connect the monitor (or projector) that is used to display the output from the Raspberry Pi.

Micro USB power connector — This port is used to connect a power supply. It should always connect at the end, after connecting all other components.

GPIO ports (General Purpose Input/output ports) — These allow to connect electronic components such as LEDs and buttons to Raspberry Pi.

1.2.1 Connection of Raspberry Pi board to Computer for Data Transfer

Connecting a Raspberry Pi board to a computer for data transfer is essential for smooth communication between the two devices. This connection enables us to make the most of the computing power and features of both the Raspberry Pi and the computer for different tasks. It makes it easy to share data, allowing to gather, handle, and send information effectively. Moreover, connecting the Raspberry Pi to a computer creates chances for teamwork on projects, programming, and sharing resources, boosting the overall usefulness and adaptability of computing setup.

To connect a Raspberry Pi board with a computer following are the required hardware and software:

1. Raspberry Pi board
2. MicroUSB to wall power block
3. MicroUSB to USB power cable;
4. Ethernet cable;
5. USB to Ethernet dongle;
6. MicroSD card

Practical Activity 1.2. Demonstrate to Setup Raspberry Pi Board for uploading the program code

Materials needed

Raspberry Pi boards, microSD cards, keyboards, HDMI cables, Ethernet cables, and power supplies.

Procedure

Step1. To start off, place the SD card into the SD card slot on the Raspberry Pi board. The cards and the wires should be connected to the proper port and pins. Identify the port and pin by observing pin configuration. Figure 1.20 shows placing the SD card.

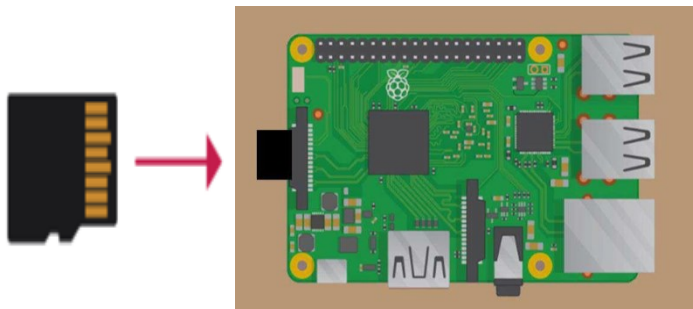


Fig.1.20: Placing the SD card

Step 2. Then connect the keyboard and the mouse via the USB ports on the Raspberry Pi as shown in Figure 1.21.

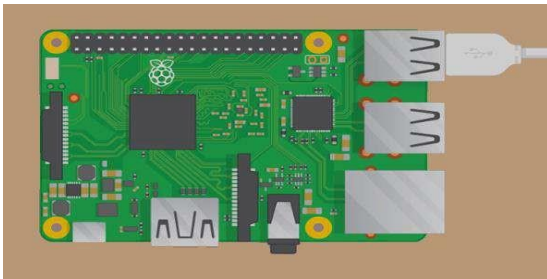


Fig. 1.21: Plugging cables to the USB ports

Step 3. Turn on the device and choose the appropriate input medium such as HDMI 1, DVI, and so on. Then, connect HDMI cable from Raspberry Pi to the device so that the status can be displayed on the monitor of the device. Figure 1.22 shows HDMI cables plugged in the board.

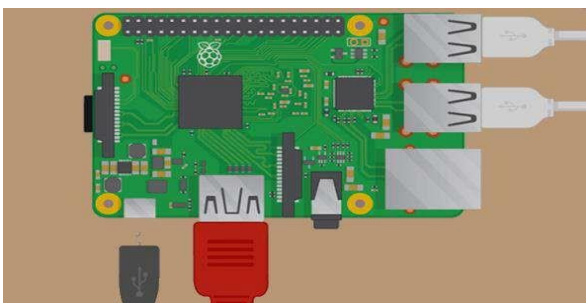


Fig. 1.22: Plugging HDMI cables to the board

Step 4. For internet connectivity with Raspberry Pi, plug in Ethernet cable into the Ethernet port, for connecting the Wi-Fi dongle to the USB ports. In the case of Raspberry Pi3, there is an onboard port for this.

Step 5. When all the cables are plugged, and the SD card is inserted correctly, connect the micro USB power

supply in the socket. This power boots up the Raspberry Pi as shown in Figure 1.23.



Fig.1.23: Connection set up of board

Step 6. LED light is visible on the Raspberry Pi, which indicates that the Raspberry Pi is connected to power. As it starts up (also called as booting), the raspberries appear in the top left-hand corner of the screen of the computer, as shown in Figure 1.24.

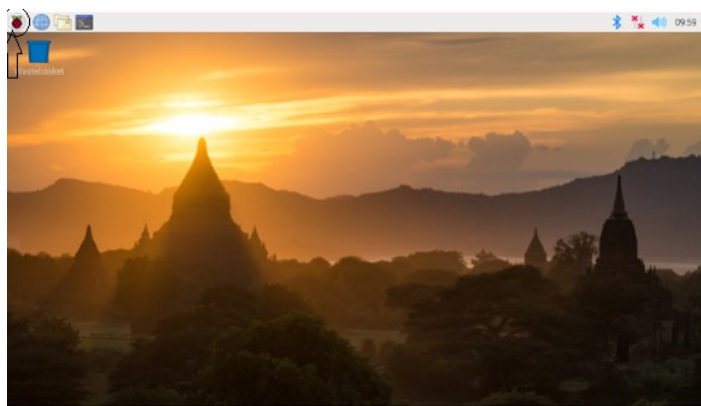


Fig.1.24: Raspberry Pi OS desktop

1.3 Checking Connectivity

Ethernet, LAN, or Wi-Fi connection should be used to connect the boards to the network. The connection should then be tested for proper operation. Steps to test the network connectivity of the device are as follows.

Step 1. Connect the device to the network

Step 2. Select Test Management Network and enter.

Step 3. Type the IP address or another DNS hostname.

Step 4. To run the test, press the Enter key.

Practical Activity 1.3. Demonstrate to test the Ethernet Connectivity in Arduino Uno board.

Materials Needed

Arduino Uno board, Ethernet shield (e.g., W5100 or W5500), Ethernet cable, Power source for the Arduino board

Procedure

Step 1. Prepare the Arduino Board

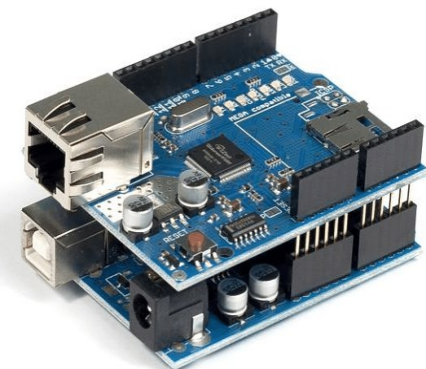
Ensure that the Arduino board is powered off before proceeding. If there's already an existing shield mounted on the Arduino board, carefully remove it to make space for the Ethernet shield.

Step 2. Attach the Ethernet Shield



Align the pins of the Ethernet shield with the headers on the Arduino board.

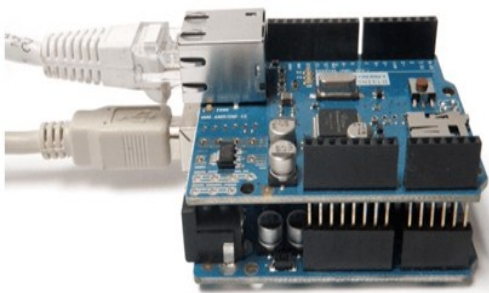
Gently press down the Ethernet shield to firmly attach it to the Arduino board. Ensure that all pins are properly aligned and seated.



Step 3. Connect Ethernet Cable

Plug- in one end of the Ethernet cable into the Ethernet port on the Ethernet shield.

Connect the other end of the Ethernet cable to the available Ethernet port on router, switch, or network hub.



Step 4. Power Up the Arduino Board

Connect the Arduino board to a power source, either via USB or an external power supply.

Ensure that the power source provides sufficient voltage and current to power both the Arduino board and the Ethernet shield.

Step 5. Verify Connection:

Once the Arduino board is powered up, you can verify the Ethernet connection by checking the status LEDs on the Ethernet shield.

Most Ethernet shields have indicator LEDs that show the status of the Ethernet connection, such as link, activity, and speed.

A solid green LED indicates a successful connection to the network.

1.4 Managing the communication hurdles

The device connections to the monitoring system could encounter some issues. The technician should fix any issues with the devices' and gateways' communication. The following are some of the possible causes of the communication barriers:

- There is no direct line of sight to the devices.
- The devices are out of range of the network connection.

- Ethernet is not properly connected.
- No network connection.

The technician should verify that each connection has been made correctly. It must make sure that the devices are installed correctly and that the connectivity range is appropriate. It should also be verified that the hardware and software are compatible.

Summary

- The chapter covers the installation of Arduino IDE and Raspberry Pi, essential tools in programming. Installation involves downloading the software, setting up components, and connecting peripherals.
- In this chapter the knowledge of coding and uploading in Arduino IDE is given in detail with examples and activities.
- It offers an overview of the Raspberry Pi board, a compact single-board computer used for various projects.
- Overall this chapter provides required knowledge for learning and implementation in electronics and programming.

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. Arduino is? (a) A type of computer (b) An open-source electronics platform (c) A type of microchip (d) A software development tool
2. Which of the following components is essential for programming Arduino boards? (a) Microcontroller (b) Capacitor (c) Resistor (d) Diode
3. IDE for in the context of Arduino stands for? (a) Integrated Design Environment (b) Integrated Development Environment (c) International Development Engine (d) Intelligent Design Environment
4. How is the code transferred to Arduino boards? (a) Via Wi-Fi (b) Using a special cable (c) Through a Bluetooth connection (d) Using a USB cable
5. Which programming languages are supported by the Arduino IDE? (a) C and Java (b) Python and CSS (c) C and C++ (d) HTML and CSS
6. Arduino IDE software is available to download for? (a) <https://www.arduino.com/software> (b) <https://www.arduino.cc/en/software> (c) <https://www.arduino.org/download> (d) <https://www.arduino.net/ide>
7. Which component is necessary to connect an Arduino board to a computer for data transfer? (a) USB AB cable (b) HDMI cable (c) Ethernet cable (d) VGA cable
8. What is the purpose of the onboard LED on an Arduino board? (a) To indicate that the board is functional and connected (b) To provide light in dark environments (c) To indicate the battery level (d) To signal errors in the code
9. Raspberry Pi is? (a) A type of fruit (b) An open-source electronics platform (c) A type of microchip (d) A software development tool
10. The main feature of Raspberry Pi is? (a) Its ability to control LEDs (b) Its low cost and compact size (c) Its compatibility with Windows only (d) Its use of proprietary software

B. Fill in the blanks

1. Arduino hardware is a programmable circuit board called a ____.
2. Arduino software is an IDE Integrated Development Environment through which developers_____ the code to the microcontroller.
3. Arduino eliminates the need for hardware; instead, it uses a simple _____to load code onto the Arduino board.
4. The Arduino IDE for the operating system can be downloaded from the website _____.
5. Select the Arduino board type under Tools > _____.
6. Choose the serial port under Tools >_____.
7. The microcontroller's software program retrieves _____, which is then sent via USB cable to the PC

8. To start off, place the SD card into the SD card slot on the Raspberry Pi board. The cards and the wires should be connected to the proper _____.
9. The devices' connections to the monitoring system could encounter some issues. The technician should fix any issues with _____.
10. Once the Arduino board is plugged-in, a LED will start_____ which indicates that the Arduino board is functional and it is connected to the PC.

C. State true or False for the following

1. The Arduino IDE is necessary to connect a microcontroller board to a computer.
2. Raspberry Pi boards have GPIO ports that allow connecting electronic components such as LEDs and buttons.
3. The serial port selection is important to establish communication between the Arduino and the computer.
4. The Raspberry Pi board requires an SD card for proper booting.
5. The Ethernet port on Raspberry Pi is used to connect it to a network wirelessly.
6. The SD card slot on Raspberry Pi is used to store the operating system software and data files.
7. Network connectivity can be tested by selecting "Test Management Network" and entering an IP address.
8. A direct line of sight is not important for effective communication between devices and gateways.
9. Arduino is an open source platform.
10. Arduino shields are physical barriers used to protect the microcontroller from external interference.

D. Short Answer Type Questions

1. What is Arduino, and what does it consist of?
2. Describe the role of Arduino software (IDE) in programming Arduino boards.
3. How does Arduino eliminate the need for separate hardware to load code onto the board?
4. Explain the file name format for the sketches created in the Arduino IDE software.
5. Describe the process of installing the Arduino IDE software.
6. What steps are involved in setting up an Arduino board for data transfer to a computer?
7. What are the essential components of a Raspberry Pi board, and what are their functions?
8. How does Raspberry Pi connect to a network for internet connectivity?
9. Explain the steps involved in setting up a Raspberry Pi board for booting up.
10. What are the methods for testing network connectivity in microcontroller boards?

Chapter 2. Introduction to Arduino Programming

The Arduino API, or the "Arduino Programming Language", consists of several functions, variables and structures based on the C/C++ language.

The Arduino API can be divided into three main parts: functions, variables, and structure as follows:

Functions – Functions are for controlling the Arduino board and performing computations. For example, to read or write a state to a digital pin, map a value, or use serial communication.

Variables – The Arduino constants, data types, and conversions. E.g. int, Boolean, array.

Structure – The elements of the Arduino (C++) code, such as

sketch (loop(), setup())

control structure (if, else, while, for)

arithmetic operators (multiplication, addition, subtraction)

comparison operators, such as == (equal to), != (not equal to), > (greater than).

2.1 Program Structure

In the Arduino project, a program is referred to as a "sketch". A sketch is essentially a file to compose a program. It is identified by the .ino extension and is consistently housed in a folder sharing the same name. This folder may encompass supplementary files, like a header file, which can be incorporated into a sketch.

Every Arduino program must include a minimum of two functions: void setup() and void loop(). The use of *void* indicates that these functions do not return any values upon execution.

The function void setup() is executed only once when the Arduino is powered on. In this function, we define initial settings such as configuring the mode of a pin (as either input or output), setting the baud rate for serial communication, or initializing libraries.

On the other hand, void loop() is where we write the code that will be executed repeatedly. This can include tasks like toggling a lamp based on an input, or performing sensor readings at regular intervals.

While these two functions are mandatory in an Arduino sketch, additional functions can be added to accommodate longer programs and enhance code organization.

Example Sketch

An example of a standard Arduino sketch, which contains some popular Arduino programming elements is given below in Figure 2.1.

```
1 /*
2 This is a comment at the top of a program,
3 it will not be recognized as code. Very good
4 to add an explanation of what your code does here.
5
6
7 This sketch shows how to read a value from a
8 sensor connected to pin A1, print it out in
9 the Serial Monitor, and turn on an LED connected
10 to pin number 2 if a conditional is met.
11 */
12
13 int sensorPin = A1; //define pin A1 (analog pin)
14 int ledPin = 2; //define pin 2 (digital pin)
15 int sensorValue; //create variable for storing readings
16
17 //void setup is for configurations on start up
18 void setup() {
```

```

19 Serial.begin(9600); //initialize serial communication
20 pinMode(ledPin, OUTPUT); //define ledPin as an output
21 }
22
23 void loop() {
24     sensorValue = analogRead(sensorPin); // do a sensor repeatedly.
25
26 Serial.print("Sensor value is: "); //print a message to
27 Serial.println(sensorValue); //print the value to the s
28
29 //check if sensor Value is below 200
30 if(sensorValue< 200) {
31     digitalWrite(ledPin, HIGH); //if it is, turn on the
32
33 }
34 //if sensor Value is above 200, turn off the LED
35 else{
36     digitalWrite(ledPin, LOW);
37 }

```

Fig. 2.1: Standard Arduino sketch

Some of the most common elements in the standard Arduino API are described as following:

a) General

setup (): The setup () function is where you put your code to run once.

Loop (): The loop () function is where your main program is stored. It will run repeatedly as long as your board is powered. The screenshot is shown below in Figure 2.2.

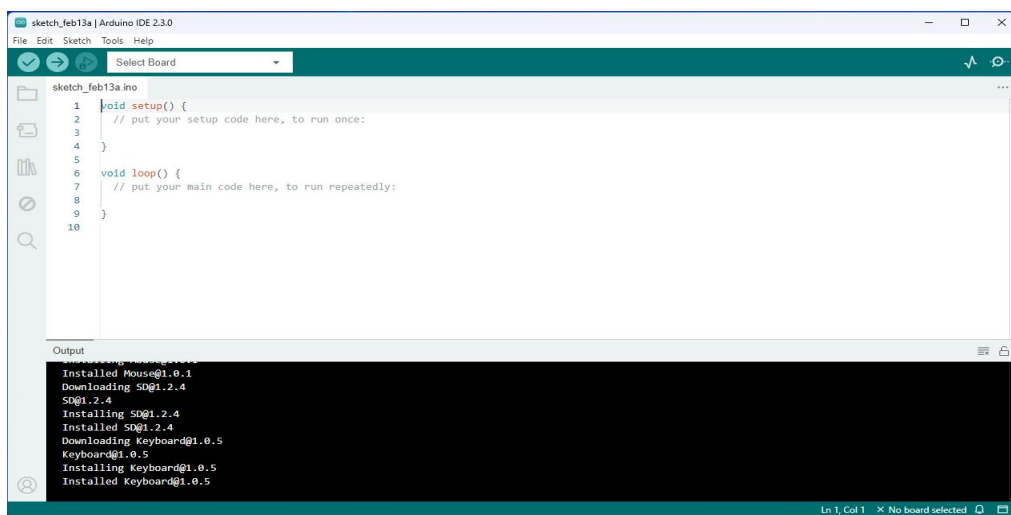


Fig. 2.2: setup() and loop() functions

delay(): The delay() function pauses the program for a set number of milliseconds as shown in the below code.

```

1 void loop() {
2
3 digitalWrite(LED, HIGH); //turn on an LED
4 delay(1000); //as program is paused, with the LED on
5 digitalWrite(LED, LOW); //program is unpaused, and the LED
6 delay(1000); //program is paused, with the LED off
7
8 }

```

The delay() function is an incredibly useful function, and you will find it in almost all examples.

b) Functions

custom functions that either just execute code and return to the program, or that return a result.

Example of a void function that does not return is shown in code below.

```

1 int x;
2
3 void loop() {
4     thisFunction(); //execute the function
5 }
6
7 void thisFunction() {
8     x++; //increase x by 1 each time function is run.
9 }

```

Example of a type int function that returns a value is shown in code below.

```

1 int value;
2
3 void setup(){
4
5 }
6
7 void loop() {
8     value = returnFunction();
9 }
10
11 int returnFunction() {
12     int returnValue = 5 + 2;
13     return returnValue;
14 }

```

c) Variable Definition

Variables can either be created locally or globally. Variables that are defined in the loop() are considered local, and variables defined at the top of your sketch are considered global. The following code shows how local and global variables are declared.

```

1 int sensorReading = x; //global variable
2
3 void setup() {
4
5 }
6
7 void loop() {
8     int sensorReading = x; //local variable
9 }

```

d) Data Types

There are several data types available for use. The most common data types are:

```

1 bool
2 byte
3 char
4 double
5 float
6 int
7 long
8 short
9 String

```

e) Serial Communication

Serial Communication is essential to Arduino programming, as it is the easiest way to know what goes on your board. For this, the Serial class is used.

Serial.begin(): Initializes serial communication between board & computer. This is defined in the void setup() function, where you also specify baud rate (speed of communication). It is shown in the code below.

```

1 void setup() {
2   Serial.begin(9600);
3 }

```

Serial.print(): Prints data to the serial port, which can be viewed in the Arduino IDE.

Serial Monitor tool as shown below in programming code.

```

1 void loop() {
2   Serial.print();
3 }

```

Serial.read(): Reads the incoming serial data as in programming code.

```

1 void loop() {
2   int incomingByte = Serial.read();
3 }

```

f) GPIO / Pin Management

Configuring, controlling, and reading the state of a digital/analog pin on an Arduino.

pinMode(): Configures a digital pin to behave as an input or output. It is configured inside the void setup () function as shown in programming code.

```

1 pinMode(pin, INPUT); //configures pin as an input

```

```
2 pinMode(pin, OUTPUT); //configures pin as an output
3 pinMode(pin, INPUT_PULLUP); //enables the internal pull-up r
```

digitalRead(): Reads the state of a digital pin. It is used to for example detect a button click. It is show in programming code.

```
1 int state = digitalRead(pin); //store the state in the "stat
```

DigitalWrite(): Writes a high or low state to a digital pin.It is used to switch on or off a component as shown below in Programming code.

```
1 digitalWrite(pin, HIGH); // writes a high (1) state to a pin
2 digitalWrite(pin, LOW); // writes a Low (0) state to a pin
```

AnalogRead(): Reads the voltage of an analog pin, and returns a value between 0-1023 (10-bit resolution). It is used to read analog components as given in programming code.

```
1 sensorValue = analongRead (A1); //stores reading of A1 in "ser
```

AnalogWrite(): Writes a value between 0-255 (8-bit resolution). Used for dimming lights or setting the speed of a motor. Also referred to as PWM, or Pulse Width Modulation. It is shown below in programming code.

```
1 analogWrite(pin, value); //write a range between 0-255 to a
```

g) Conditionals

Conditionals are some of the most popular elements in any program. In Arduino, a typical conditional consists of an if and else statement as shown below in programming code.

```
1 if(variable == true){
2   //do something
3 }
4 else {
5   //do something else
6 }
```

h) Loops / Iterations

The for and while loops are commonly used in programs, to execute a block of code for a set number of times, or while a condition is met.A basic use of a while loop to execute a block of code while variable is true as shown in programming code.

```
1 while (variable == true) {
2   //do something
3 }
```

A basic use of a for loop is to execute a block of code a custom number of times (in this case, 10).It is shown in programming code.

```
1 for (int x = 0; x < 10; x++) {
2   //do something 10 times
3 }
```

To break out of a loop, you can use break. In the snippet below, if a condition is met (variable is true), we break out of the loop as shown below in programming code.

```
1 for (int x = 0; x < 10; x++) {
2   if(variable == true) {
3     break;
4   }
```

2.2 Libraries

Libraries simplify the utilization of complex code for various tasks such as sensor readings, motor control, or establishing an Internet connection in Arduino projects. By installing a library, incorporating it at the outset of your code, and leveraging its functions, you can avoid the need to manually craft extensive code for these functionalities. All Arduino libraries are accessible without charge and are open for use by anyone. It is imperative to include a library at the commencement of your code to harness its capabilities, as demonstrated in the example below:

```
# include <Library.h>
```

2.3 Arduino Software Tools

The software tools of the Arduino ecosystem are another essential component. An Integrated Development Environment (IDE), is commonly called the Arduino platform. You must write a program, translate it into machine code, and then upload the updated program to your board to program it. Everything is made possible by the Arduino IDE, from writing the first line of code to having it run on the microcontroller of the Arduino board. To handle all of your code development, you can use an online version of the program or application, which you can download. At present three version of Arduino IDEs are available:

- Arduino IDE 1.8.x (classic)
- Arduino IDE 2 (new)
- Arduino Web Editor (online)

2.4 Workflow in Arduino IDE

To upload code to an Arduino board using the IDE, follow these steps:

Install your board: This involves installing the appropriate "package" for your board directly in the IDE. Without the package, your board cannot be utilized. The installation process is swift and straightforward.

Create a new sketch – A sketch serves as the primary program file, where a set of instructions can be written intended for execution on the microcontroller.

Compile the sketch – The code written doesn't directly resemble what gets uploaded to the Arduino. Compilation involves error-checking and the conversion of code into a binary file (comprising 1s and 0s). If any issues arise, they will be displayed in the error console.

Upload the sketch – Once the compilation is successful, the code can be uploaded to the board. During this step, physically connect the board to the computer and select the appropriate serial port.

Serial Monitor (optional) – For most Arduino projects, understanding the board's activities is crucial. The Serial Monitor tool, available in all IDEs, facilitates the transmission of data from your board to your computer, providing insights into its functioning.

Practical Activity 2.1. Write a basic Arduino program that blinks an LED to gain hands-on experience in writing, compiling, and transferring code to an Arduino microcontroller.

Materials Required

Arduino board, Computer with Arduino IDE installed, USB cable for Arduino, Components for a simple project (e.g., LEDs, resistors, jumper wires)

Procedure

Step 1. Connect the resistor in series with the LED on the breadboard.

Step 2. Connect the left leg of the resistor (connected in series with red LED) to the digital output pin of the UNO board, i.e., PIN 8.

Step 3. Connect the negative/shorter terminal (Cathode) of the LED to the GND pin of the UNO board using the wire, as shown below in Figure 2.15.

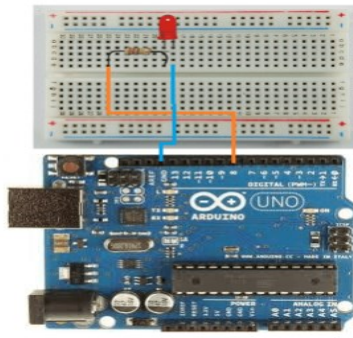


Fig. 2.15: Connection of board with the components

Step 4. Connect the USB cable.

Step 5. Open the IDE and enter the following code.

```
void setup
{
pinMode (8, OUTPUT); //to set the OUTPUT mode of pin number 8.
}
Void loop ()
{
digitalWrite (8, HIGH);
delay(1000); //1 second = 1 x 1000 milliseconds
digitalWrite (8, Low);
delay(500); //0.5 second = 0.5 x 1000 milliseconds
}
```

(delay duration can be modified according to choose or as per the requirements)

Make sure that the code is free of errors.

Step 6. Select the board and serial port in the Arduino IDE.

Step 7. Upload the sketch or code on the board as shown in figure 2.16.

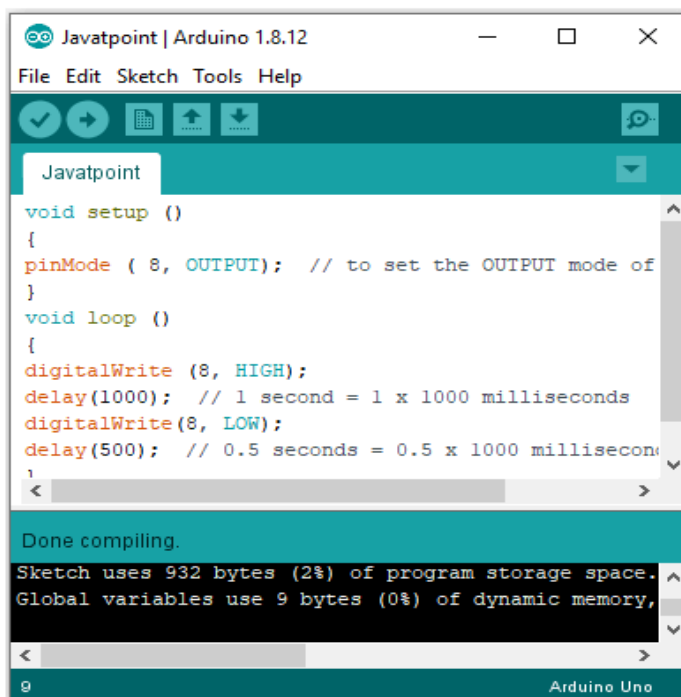


Fig. 2.16: Arduino Sketch

Step 8. Click on the Verify button to compile the code.

Step 9. The RX and TX LED on the board will light up after the successful uploading of the code.

Step 10. Observe that the LED on the breadboard will dim and light for the specified duration as mentioned in Arduino sketch.

Practical Activity 2.2. Write an Arduino program that connects Arduino board with IR Sensor and output can be seen on an LED connected to the board.

Materials needed

Arduino Uno board, IR sensor, LED and Resistor Kit, breadboard, Jumper wires, USB cable type A/B

Procedure

The parts of the IR motion sensor module are given in Figure 2.17.

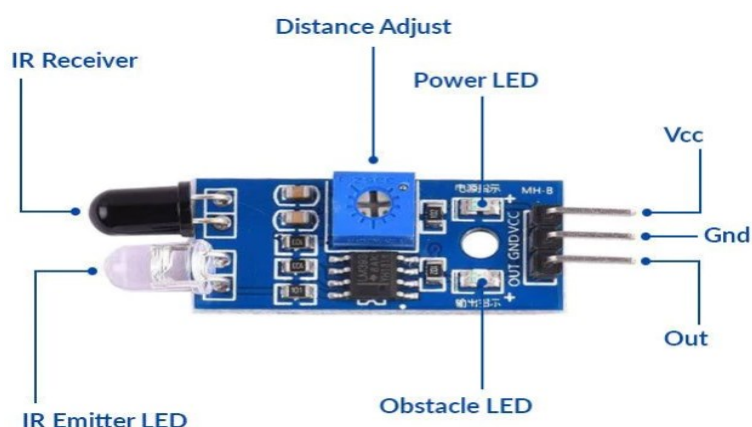


Fig. 2.17: IR Motion Sensor Module – Parts

Step 1. Write the following sketch in Arduino IDE for Arduino with IR Sensor

```
int SensorPin = 10;
int OutputPin = 13;
void setup() {
  pinMode(OutputPin, OUTPUT);
  pinMode(SensorPin, INPUT);
  Serial.begin(9600);
}
void loop() {
  int SensorValue = digitalRead(SensorPin);
  Serial.print("SensorPin Value: ");
  Serial.println(SensorValue);
  delay(1000);
  if (SensorValue==LOW){ // LOW MEANS Object Detected
    digitalWrite(OutputPin, HIGH);
  }
  Else
  {
    digitalWrite(OutputPin, LOW);
  }
}
```

Step 2. Connect IR Sensor' output pinto to digital pin no.2 of Arduino UNO board.

Step 3. Connect LED from digital pin 13 to ground through a 220 ohm resistor.

These steps are illustrated in Figure 2.18.

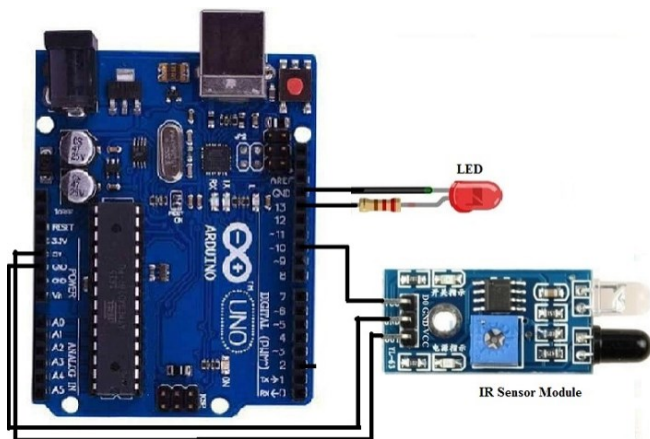


Fig. 2.18: Arduino with IR Sensor

Step 4. Observe the LED glow. It should turn ON, when an object comes in the range of the IR Sensor. If it is not shown the desired output, ensure the circuit has been properly assembled, and verified and uploaded the code to your board.

Practical Activity 2.3. Demonstate to set up Arduino IDE for the ESP 8266 Node mcu board

Materials needed

ESP8266Nodemcu board, LED,180 Ohm Resistor. Bread board, Jumper wires, USB cable

Procedure

Step 1. Collect all the required components as shown in Figure 2.19.

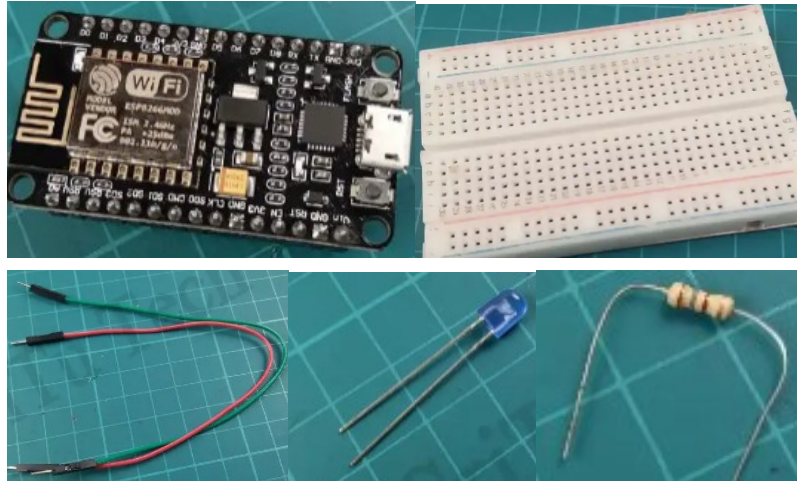


Fig 2.19: ESP8266Nodemcu board, breadboard, Jumper wires, LED and 180 Ohm Resistor.

Step 2. Next, connect these components as shown in schematic of Figure 2.20.

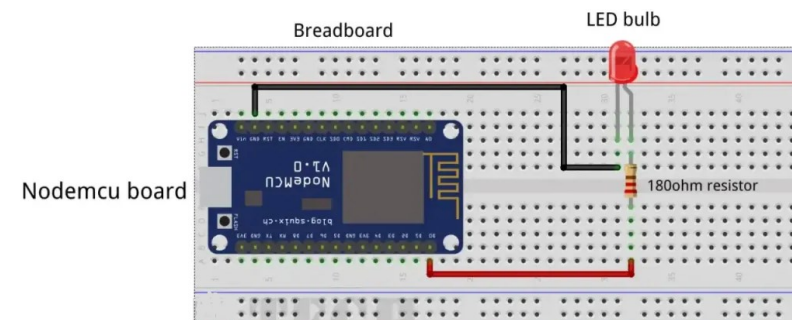


Fig. 2.20: Schematic diagram of connection

Step 3. After connecting the components, the circuit will look like as shown in Figure 2.21.

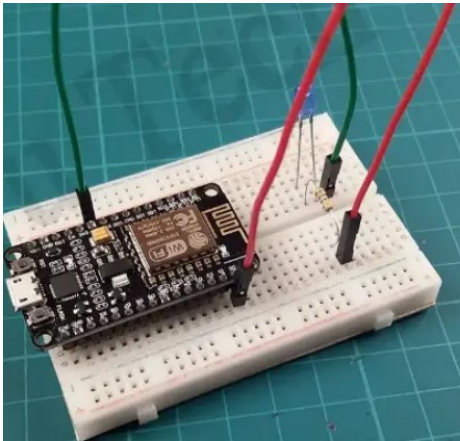


Fig. 2.21: Connection of all the components

Step 4. Connect this circuit to the computer using a USB cable.

Step 5. Open Arduino IDE and write a simple LED bulb blink code.

```
void setup() {  
  pinMode(D0, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(D0, HIGH);  
  delay(1000);  
  digitalWrite(D0, LOW);  
  delay(1000);  
}
```

Step 6. Now select the board and port in Arduino IDE as shown in screenshot of Figure 2.22.

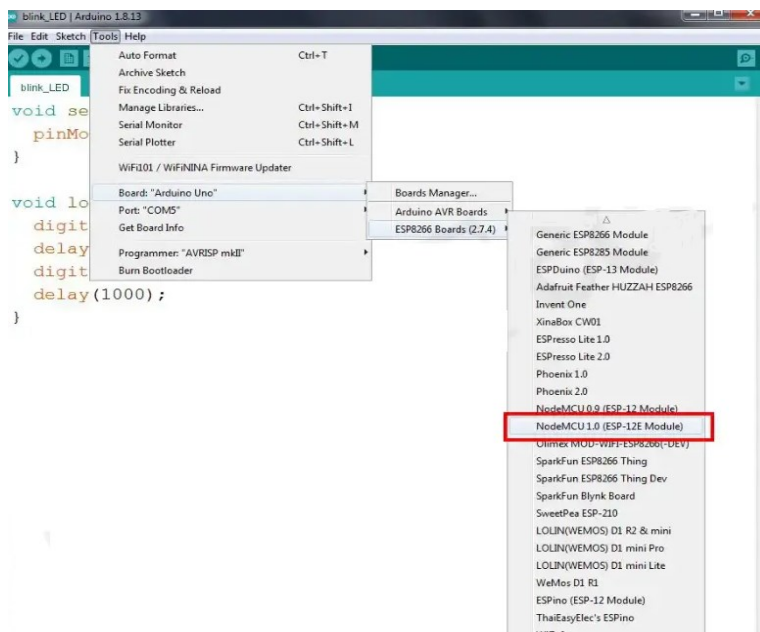


Fig. 2.22: Board selection window

Step 7. Now, click the upload button as shown in the screenshot in Figure 2.23.

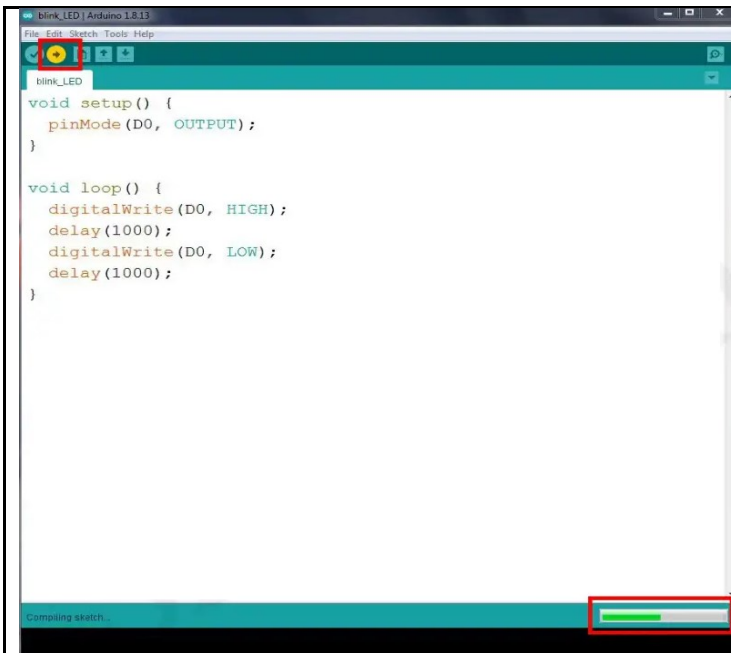


Fig. 2.23: Uploading window

Step 8. As the code is uploaded successfully, the LED starts to blink as shown in Figure 2.24 (a) and (b).

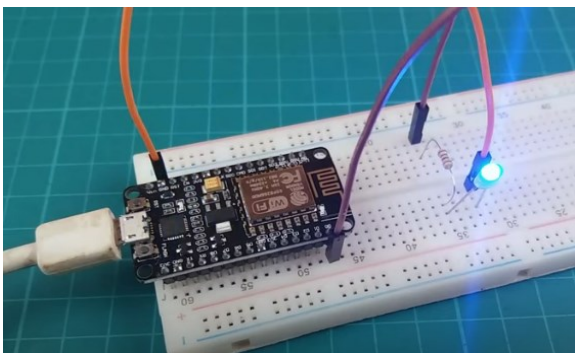


Fig. 2.24:(a) LED blinking (ON)

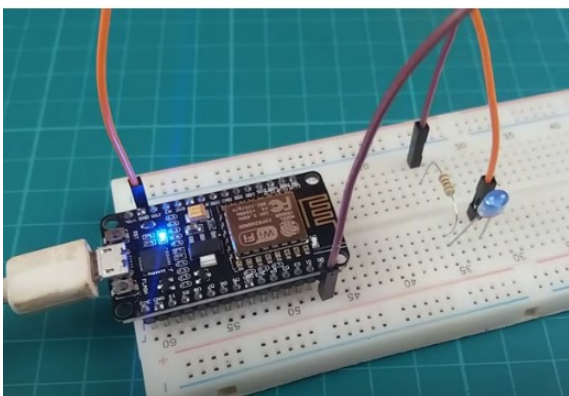


Fig. 2.24: (b) LED blinking (OFF)

2.5 Simulation tools

Simulation is a very powerful tool to understand how the device will work after implementing it in real life. Before starting any project, it is a good practice to simulate it and virtually verify that it works or not. Proteus is a simulation-based software. By using this software Arduino can be simulated.

2.5.1 Steps for simulation of an Arduino

The steps for simulation of an Arduino are as follows:

Step 1. Open the design software for simulation and place the components of the circuit as shown in Figure 2.25.

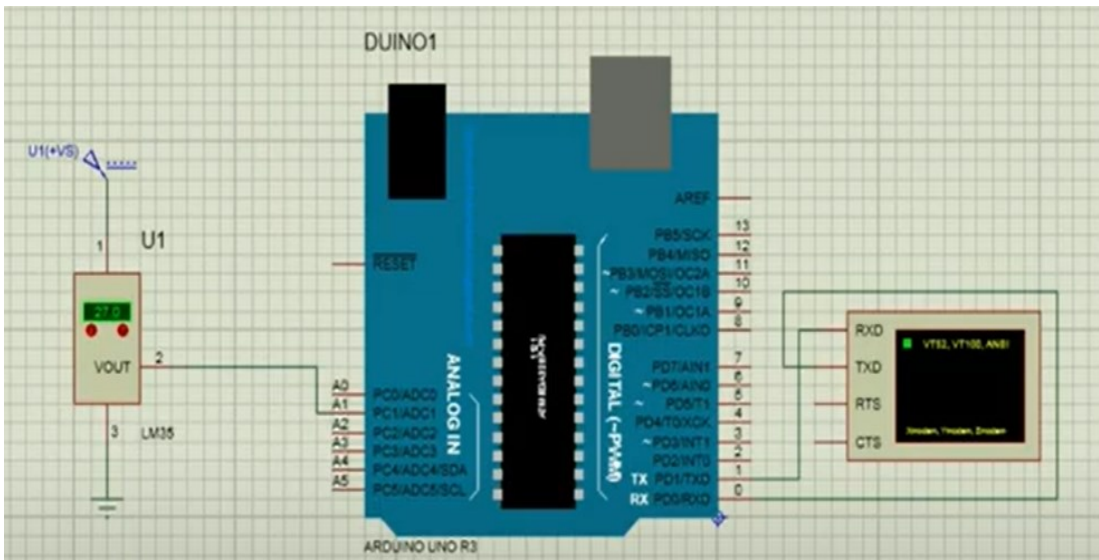


Fig. 2.25: Designing the circuit

Step 2. Save the design and open the code in the application for the Arduino coding as shown in Figure 2.26.

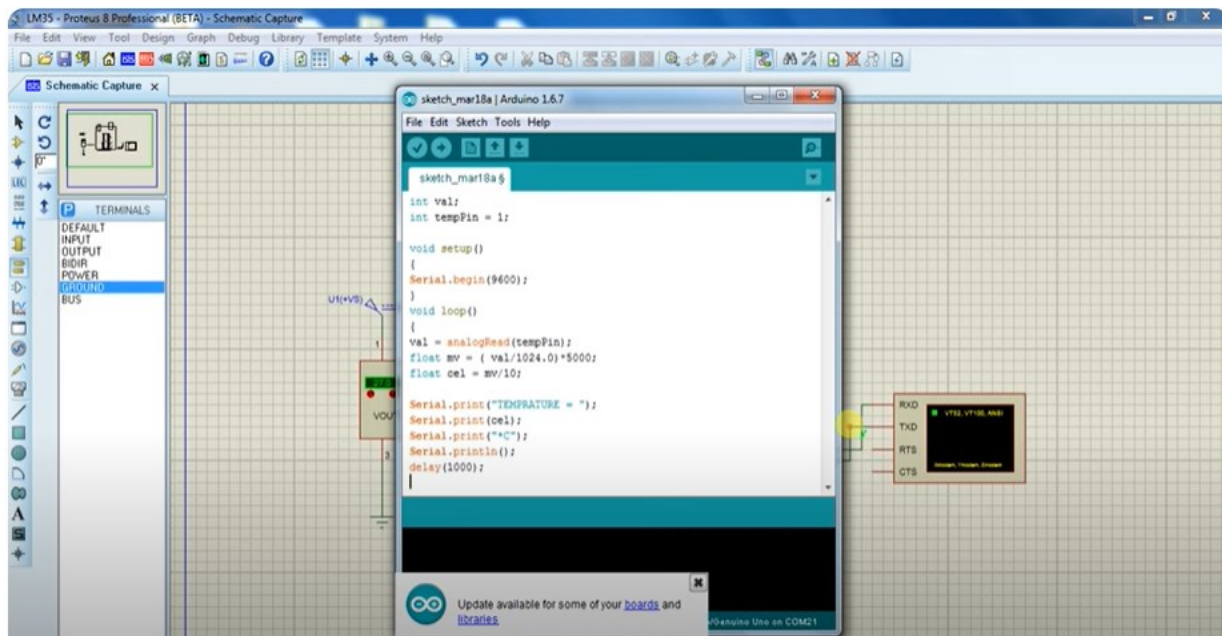


Fig. 2.26: Arduino code

Step 3. Compile the code and save it as shown in Figure 2.27.

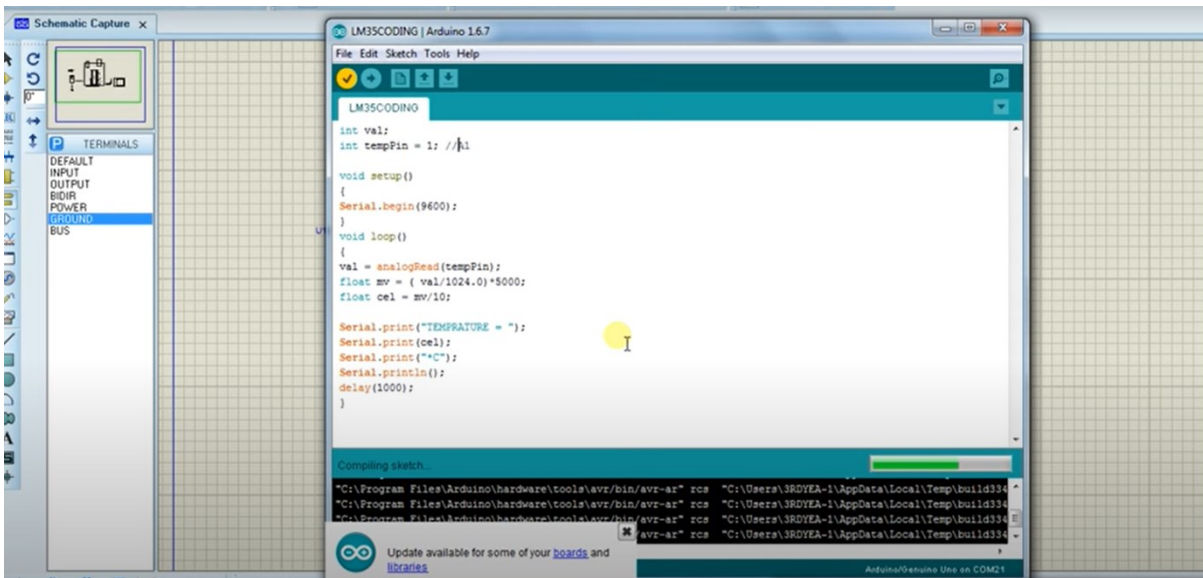


Fig. 2.27: Compiling the code

Step 4. Double-click the Arduino board. The Edit component window appears, as shown in Figure 2.28.

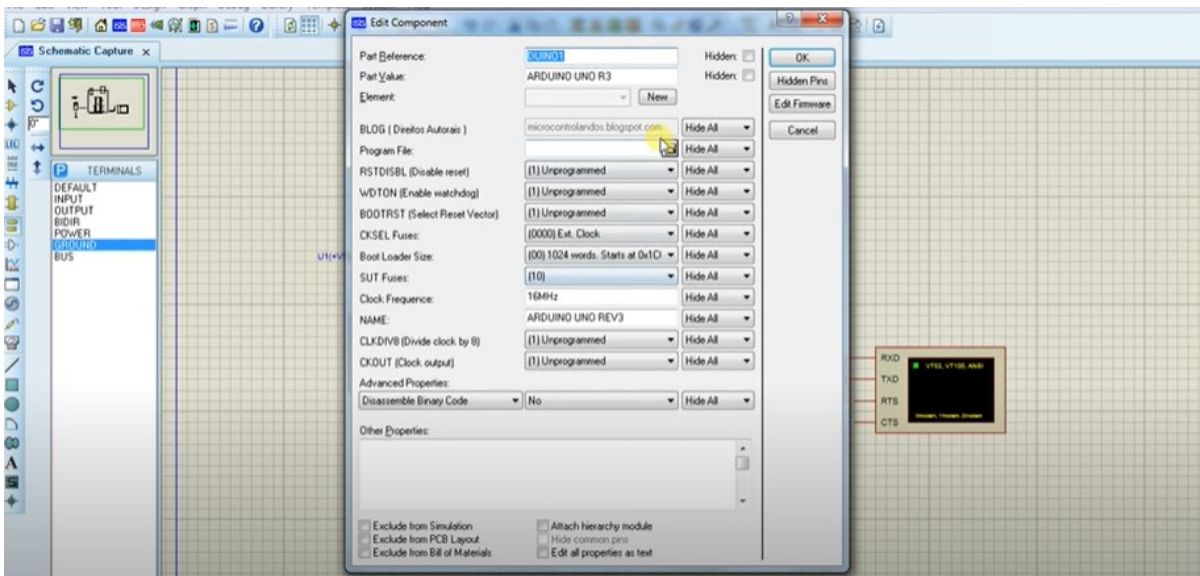


Fig. 2.28: Edit component window

Step 5. Browse the Arduino code and run. A virtual terminal with output will appear, as shown in Figure 2.29.

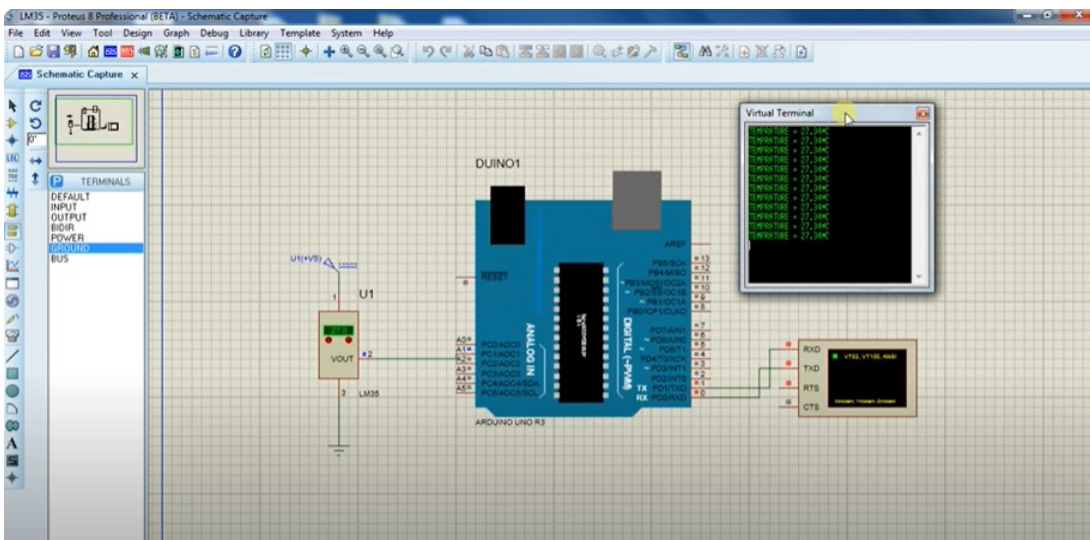


Fig. 2.29: Virtual terminal window with output

Summary

- The content introduces Arduino programming, covering its key elements and tools.
- It explains the structure of an Arduino sketch, the role of functions like `void setup()` and `void loop()`, and common programming elements such as functions, variables, and data types.
- It emphasizes the importance of libraries in extending Arduino's capabilities and outlines the process of uploading code using the Arduino IDE.
- Additionally, various examples of programming on Arduino IDE have been given in this chapter to provide better understanding.
- Also, chapter highlights the role of simulation tools like Proteus in verifying Arduino projects.

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. Arduino API consist of (a) Functions, variables, and structures (b) Functions, strings, and arrays (c) Variables, loops, and conditions (d) Functions, data types, and libraries
2. Which function is executed only once when the Arduino is powered on? (a) `loop()` (b) `setup()` (c) `delay()` (d) `pinMode()`
3. What is the purpose of the `void loop()` function? (a) To configure initial settings (b) To execute code repeatedly (c) To define custom functions (d) To read analog pins
4. Which Arduino function is used to pause the program for a set number of milliseconds? (a) `digitalWrite()` (b) `analogRead()` (c) `delay()` (d) `pinMode()`
5. Which data type is used to store integer values in Arduino? (a) float (b) Boolean (c) int (d) char
6. Which Arduino function is used to initialize serial communication between the board and the computer? (a) `Serial.begin()` (b) `Serial.print()` (c) `Serial.read()` (d) `Serial.available()`
7. What is the purpose of the `pinMode()` function in Arduino? (a) Reads the state of a digital pin (b) Writes a value to a digital pin (c) Configures a digital pin as input or output (d) Reads the voltage of an analog pin
8. Which Arduino statement is used to execute a block of code repeatedly while a condition is true? (a) if (b) else (c) while (d) for
9. What is the role of libraries in Arduino programming? (a) To simplify complex code for various tasks (b) To define local and global variables (c) To initialize serial communication (d) To execute code only once
10. Which step is essential for uploading code to an Arduino board using the IDE? (a) Creating a new sketch (b) Compiling the sketch (c) Saving the code as a binary file (d) Connecting the board to the computer

B. Fill in the blanks

1. In the Arduino project, a program is referred to as a "_____".
2. Every Arduino program must include at least two functions: `void _____()` and `void _____()`.
3. The function `void loop()` is where we write the code that will be _____.
4. Conditionals like _____ and _____ are popular elements in Arduino programming.
5. Variables defined within the `loop()` function are considered _____, while variables defined at the top of the sketch are considered _____.
6. `Serial.print()` is used to _____ data to the serial port, which can be viewed in the Arduino IDE _____ tool.
7. The `pinMode()` function configures a digital pin to behave as either an _____ or an _____.
8. The `analogRead()` function reads the _____ of an analog pin and returns a value between 0 and _____.
9. To simulate an Arduino project, one can use software tools like _____.

10. The `digitalWrite()` function writes a _____ or _____ state to a digital pin, used to switch on or off a component.

C. State true or False for the following

1. Arduino sketches are identified by the `.ard` extension and are consistently housed in a folder sharing the same name.
2. The Arduino API includes functions, variables, and structures based on the Python language.
3. The `void loop()` function in Arduino runs repeatedly as long as the board is powered, executing the main program code.
4. The function `void setup()` in Arduino is executed only once when the board is powered on, where initial settings are defined.
5. Arduino IDE is the only software tool available for writing and uploading code to Arduino boards.
6. The Serial Monitor tool in Arduino IDE is used to compile the code and upload it to the Arduino board.
7. Libraries in Arduino programming can only be installed from paid subscription services.
8. The `delay()` function in Arduino pauses the program execution for a set number of seconds.
9. Proteus is a simulation-based software used for simulating Raspberry Pi projects.
10. The `delay()` function can be used to control the speed of a motor connected to an Arduino board.

D. Short Answer Type Questions

1. What are the three main parts of the Arduino API?
2. Describe the purpose of the `void setup()` function in Arduino.
3. Explain the role of the `void loop()` function in an Arduino sketch.
4. How does the `pinMode()` function contribute to Arduino programming?
5. What is the significance of the `delay()` function in Arduino sketches?
6. Briefly describe the purpose of libraries in Arduino programming.
7. How does the Serial Monitor tool in Arduino IDE aid in debugging?
8. What are the steps involved in uploading code to an Arduino board using the IDE?
9. Explain the importance of including libraries at the beginning of Arduino code.
10. How is simulation useful in Arduino projects, and what software tool is commonly used for simulation?

Setting up and configuring nodes, gateways, and control edge appliances is essential for building effective IoT systems. Nodes collect data, gateways facilitate communication, and control edge appliances process data at the network edge. Proper setup ensures smooth operation and secure connectivity in IoT networks. This guide explores the key steps involved in initializing and configuring these components, enabling users to establish reliable and scalable IoT infrastructures.

Setting up a username and password on routers and IoT devices is known as initializing a node and gateway. The technician must finish the full installation of the IoT setup before initializing the nodes and gateways. This involves working with IoT devices, such as setting up a router, an IoT camera, and a functional Internet connection. For instance, using the procedures depicted in Figure 3.1, an IoT camera can be set up alongside the IoT framework, and then, finally, a node gateway can be initialized.

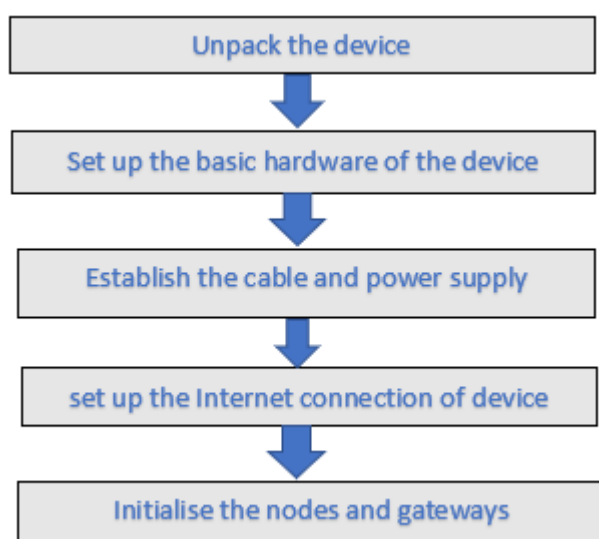


Fig. 3.1: Prerequisite steps for an IoT device installation

3.1 IoT Device Installation

In the process of installation of IoT devices like cameras, motion sensors, an IoT camera, certain steps need to be performed.

3.1.1 Unpack the Camera

- The technician will receive the camera and its accessories in the package as part of the installation kit.
- Before unpacking, compare the installation kit's specifications to the order copy.
- Using the appropriate tool (a utility knife) remove the packaging.
- Verify that the package contains a complete and undamaged device along with all of its accessories.
- Dispose of the packaging waste according to the operating guidelines. Install the fundamental camera hardware.

3.1.2 Set Up the Basic Camera Hardware

Once unpacked, the foundational hardware requires setup, including mounting the camera and its stand. The following steps illustrate the basic hardware setup procedures:

- Slide the camera onto the stand to connect it.
- Mount the screws and secure them tightly.

The Figure 3.2 shows the camera hardware assembly.

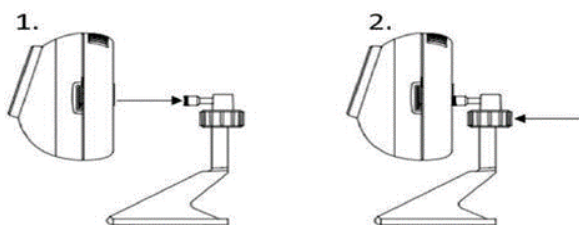


Fig. 3.2: Camera hardware setup

if the camera needs to be mounted on a wall then following actions must be taken:

- Locate suitable location on the wall where the camera can adequately cover the surveillance area.
- Next, draw pilot holes on the camera's bottom stand and drill them.
- Insert the wall anchor into the holes, then use screws to fasten the camera's bottom stand.
- Fasten the camera to the stand.

Figure 3.3 shows the mounting of a camera on the wall.

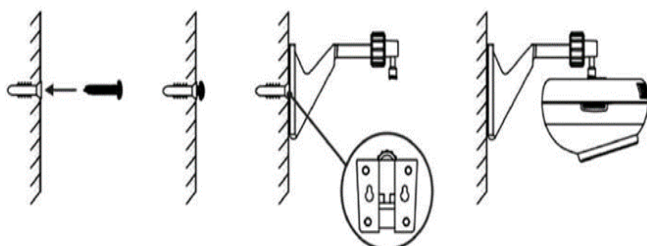


Fig. 3.3: Mounting of the camera on a wall

3.1.3 Ensure the Power Supply and Cable Connection

The connection of the power cable and power supply must be completed next. To finish the cable connection, perform the following actions:

Verify the power adaptor's specifications. The power adaptor that is typically included has a power output of 12 V/1 A. The power labeling on an adaptor is displayed in Figure 3.4.



Fig. 3.4: Specifications for power adapters

Connect the power adaptor end to the power supply port on the camera's back. Next, attach the power supply plug to a power supply port that is close as shown in Figure 3.5.

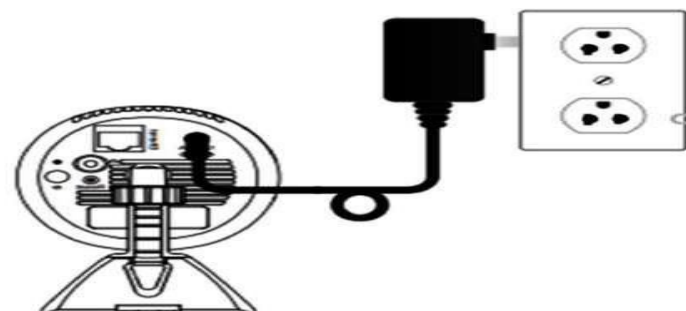


Fig. 3.5: Connection for the power adaptor

3.1.4 Set Up the Internet Connection of Camera Device

For the IoT camera to operate via wireless internet using Wi-Fi, it must connect to the router. Depending on the customer's needs or site requirements, there may already be a pre-installed wireless internet connection available, or the technician may need to install a router to enable wireless connectivity.

If a pre-installed wireless internet connection is accessible then following steps should be carried out to complete a wireless setup automatically

Step 1. Verify that the router device's wireless Internet connection is functioning properly.

Step 2. Ascertain that the camera is positioned within the router's coverage area.

Step 3. For 1 to 5 seconds, press the WPS (Wi-Fi Protected Setup) button on the router or gateway. so that the WPS LED begins to flash.

Step 4. Next, press the WPS button on the camera's back to make the WiFi LED start blinking and turn green.

Figure 3.6 shows the steps of the automatic connection of router and camera:



Fig. 3.6: Automatic router and camera connection

If the automatic connection fails, establish a manual connection using a utility program like EnViewer that was included on the CD that came with the IoT camera's packaging. EnViewer Finder by EnGenius is being used in this illustration. The procedure to manually connect the camera is as follows:

Step 1. Turn on the laptop or desktop, wait until the system starts up, and check that the window opens and runs.

Step 2. Run the utility tool disc by inserting it into the disc drive. The following are the steps to run the disc:

- Click the start button to launch "My Computer."
- Select "Run the disc as administrator" by performing a right-click on the disk drive.
- To finish installing, adhere to the installation instructions.

Step 3. To use the tool after completing the installation, follow these steps:

- Access the desktop of the computer. Pressing "window" and "R" on the keyboard will open and run the window.
- Type the name of the tool and press "Enter".

Step 4. From the list provided in the tool window, select the camera linked to the router network. If the camera isn't showing up in the list, click Refresh to see it. The device will display the hostname, IP (in this case, the camera's IP is assumed to be 192.168.0.101), and version in the list's camera row.

- The screenshot in Figure 3.7 shows the camera listed in the utility tool.



Fig. 3.7: Screenshot showing the camera listed in the utility tool

Select the camera from the list, enter the username and the password in the column given on the top right corner, and then click next for the configuration shown in Figure 3.7. The default username and password are given in the camera packaging or in the user manual. Figure 3.8 shows the username and password on a camera packing.



Fig. 3.8: Username and password on camera packing

Click the network tab given in the options in the utility tool and select the mode of connecting the camera with the network. Figure 3.9 shows the modes of network connection – DHCP, Manual, PPPoE.



Fig. 3.9: Screenshot showing the modes of network connection

Dynamic Host Configuration Protocol (DHCP) – DHCP allows to connect to the network without providing any IP. The camera automatically requests the IP from the router.

Manual – In this option, IP address is to be provided in the TCP/IP section. Make sure that the IP is not used by some other network. Figure 3.10 shows the manual network setup wizard.

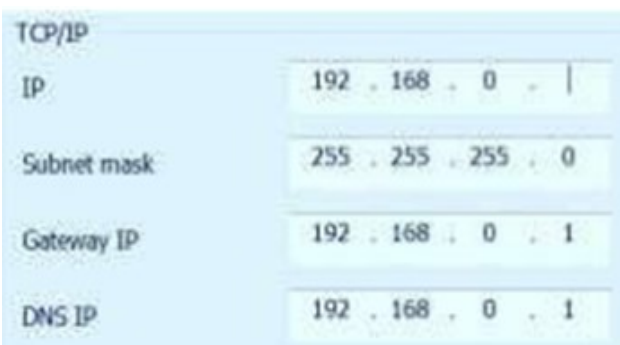


Fig. 3.10: Manual network setup wizard

Point-to-Point Protocol over Ethernet (PPPoE) – The camera and modem can be directly connected when the Internet service is using the PPPoE Internet protocol. This calls for getting in touch with the service provider to

get the necessary information details. In the same PPPoE protocol setting on the router, only the username and password are required as shown in Figure 3.11.

Fig. 3.11: Setting up the PPPoE protocol

The camera will be successfully added to the wireless Internet connection after doing this. Depending on the needs, the technician is given a router in the configuration kit, whether a wireless internet connection is already present on the site or whether a router device needs to be installed to establish a wireless internet connection.

3.1.5 Installation of edge devices

Edge devices in IoT refer to hardware components deployed at the network edge to collect, process, and transmit data from IoT sensors and devices. Examples include gateways, edge servers, routers, and computing nodes. These devices enable real-time data processing, reduce latency, conserve bandwidth, and enhance security by processing data closer to its source. They play a crucial role in enabling faster insights, improved efficiency, and enhanced privacy in IoT deployments.

A) Installation of the Router

Following steps has to be followed by the technician to install a router for a wireless internet connection at site.

1. Unpacking the router device.
2. Connecting the ethernet wire with router.
3. Configuring the router device Initialising the router device.

1. Unpacking the Router Device

The technician may get the router from the Internet Service Provider (ISP) or the manufacturer, based on the requirement of the job. Match the specification of the router with the requirement of the camera. The table 3.1 lists some of the specifications that need to be checked with the camera:

Table 3.1 Specifications to be checked

Interface	RJ45
Wi-Fi	802.11 b/g
Secured access	WPA/WPA2-PSK and WEP
Data transmission rate	Up to 54 Mbps

Unpack the router, then check the accessories and devices for any damage. The following list shows the items provided in the packaging:

- Router device
- RJ 45 cable
- Power adaptor
- Product manual

2. Connecting the Ethernet and Power Cable with Router

After unpacking, place the router at a position which is suitable for a clear line of sight with the devices, such as a camera as per the example, that need to be connected. After positioning the router, the Ethernet and the power cable of the router need to be connected to the power source using the following steps:

Step 1. Connect the power cable end to the back of the router and the adapter plug to a power socket. Ensure that the power LED indicator light on the router turns ON. Figure 3.12 shows the LED indicator on a router.



Fig. 3.12: Router LED indicator

Step 2. Connect the Ethernet cable given in the kit to the back of the router in the WAN port and the other end of the cable to the modem installed at the site. Figure 3.13 shows the power and Ethernet cable connection with the router.

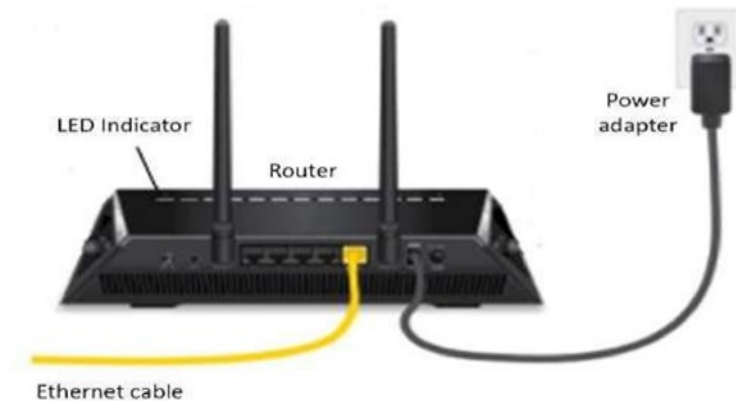


Fig. 3.13: Power and Ethernet cable connection with router

3. Configuring the Router Device

To configure the router, a technician can connect the router with a desktop/laptop through a LAN wire or with the help of the IP address. For this following step need to be performed:

Step 1. Open a web browser and in the address bar enter the IP address for the router which is given in the product manual. Suppose 192.168.0.1 is the IP address of the router, then enter it in the browser as shown in Figure 3.14.

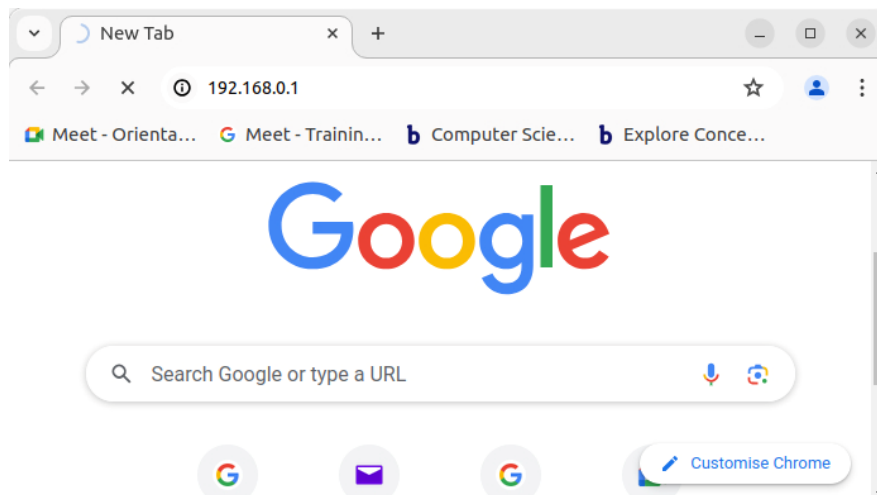


Fig. 3.14: IP address for router configuration – IoT Devices/Systems

Step 2. Then, enter the default username and password given in the router manual or within the packing in the window that opens. Figure 3.15 shows the window:

Fig. 3.15: Window for router configuration

After completing the hardware setup of the IoT device and connecting it to the Internet, the final step is to initialize the node devices and the gateway routers. The initialization of gateways and nodes means configuring the network settings, that is, the TCP/IP configuration setting, securing the network, and commissioning the IoT device framework.

Node Initialization

Setting a login and password for the system default user is necessary when using a node or device for the first time. Initialization of nodes (IoT devices) is another name for this action. The system will become secure and configured as a result. As an illustration, initialization through network video recorder (NVR) software is mentioned for a camera setup using NVR, in which the NVR device is initialized with the router. A typical NVR device is shown in Figure 3.16.



Fig. 3.16: Front and back view of an NVR

Practical Activity 3.1. Demonstrate to establish the connection between an NVR device, a camera, and a router.

Material Required

NVR device, camera, router, ethernet cable

Procedure

Step 1. Unpack the NVR device and connect the device to the router using an Ethernet cable. Figure 3.17 shows the connection of IoT camera, a router, and an NVR device setup:

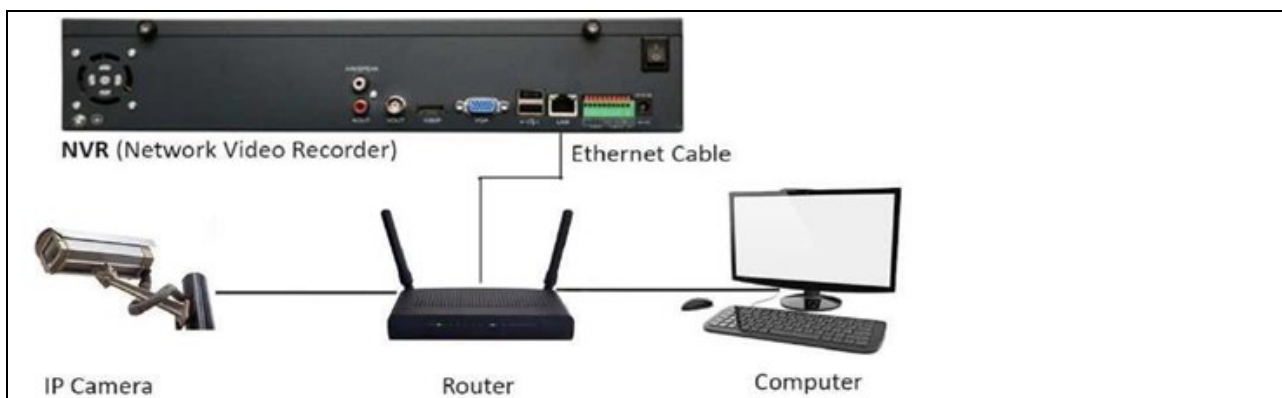


Fig. 3.17: NVR device connection with router and camera

Step 2. Power ON the NVR device and let it boot into the initialization interface. The power indicator of an NVR device is displayed in Figure 3.18.

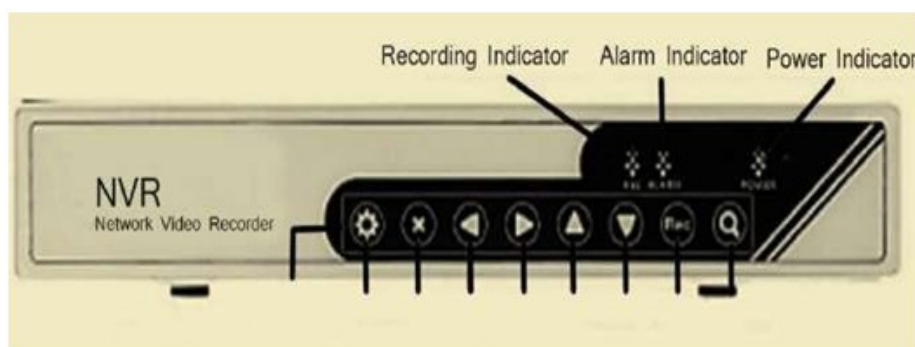


Fig. 3.18: NVR device indicators

Step 3. Either the NVR device comes with software, or it can be downloaded from the Internet. For example, the NVR software for Genius Vision camera setup can be downloaded from the link:

https://geniusvision.net/community/GeniusVisionNVRCommunitySetup_v960.exe

Step 4. Install and activate the NVR software on a computer system linked to the same network as the IP camera and NVR device.

Step 5. Configure the default login details for the camera. Figure 3.19 shows the device login screen of NVR software on a computer system.



Fig. 3.19: Device login screen

Step 6. For initialization of the camera device, connected through the same LAN, launch the interface on the desktop and check the connected device list.

Figure 3.20 shows the screenshot of the list of devices connected to the network.



Fig. 3.20: Screenshot of the list of devices connected to the network

Step 7. Select the uninitialized camera device from the list and enter the initializing interface by clicking on the “Initialize” button. Figure 3.21 shows the initializing interface.



Fig. 3.21: Initializing interface

Step 8. Select the devices and click “Initialize” and set the initialization parameters. Figure 3.22 shows the setting up the password of the device.



Fig. 3.22: Setting up of password of the device

Step 9. Figure 3.23 shows the detected devices after initialization.



Fig. 3.23: Detected devices after initialization

Gateway Initialization

To establish an Internet connection between the network provider and the gateway, which is the router installed at the location, the gateway or router must first be initialized.

Gateway initialization is the vital process of setting up a gateway device to enable communication between different networks. It involves configuring hardware, installing software, assigning network addresses, setting up routing, implementing security measures, and enabling monitoring. Through thorough testing and validation, the gateway ensures reliable and secure network operation.

Practical Activity 3.2. Demonstrate to initialize a gateway or router.

Material required

D-Link router, computer, Ethernet cable

Procedure

Step 1. Attach the power adapter to the router's back panel.

Step 2. Connect the computer and router with an Ethernet cable. Figure 3.24 represents the window that will open on the screen.

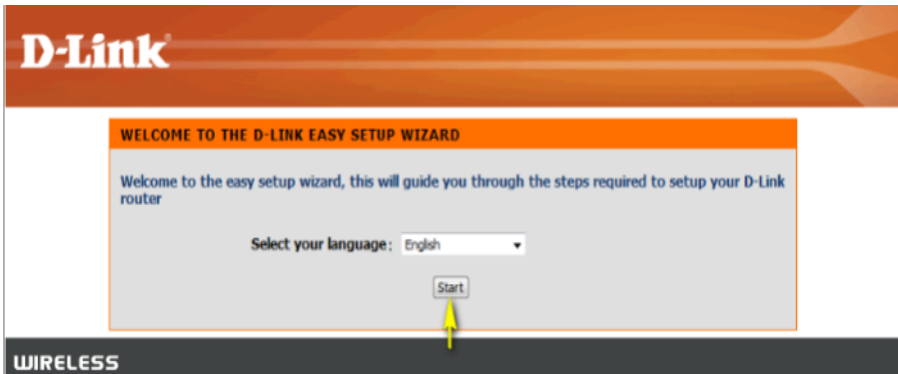


Fig. 3.24: Router setup wizard

Step 3. Configure the Internet connection and set up a password. Figure 3.25 shown the configuration of Internet connection and password.

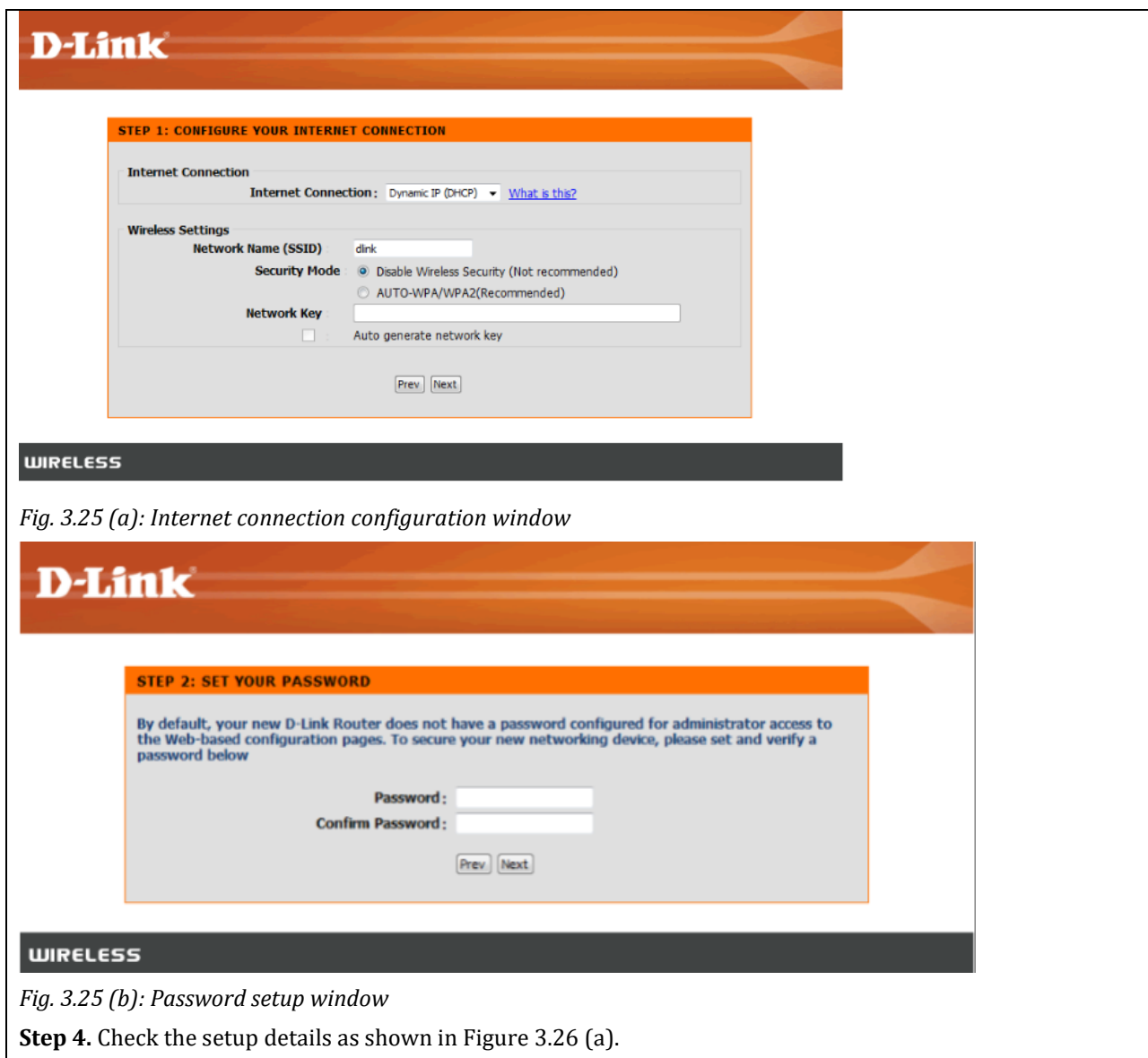


Fig. 3.25 (a): Internet connection configuration window

Fig. 3.25 (b): Password setup window

Step 4. Check the setup details as shown in Figure 3.26 (a).

Product Page : DIR-605L
Hardware Version : Ax
Firmware Version : 1.12

D-Link

CURRENT NETWORK SETTING

The current network settings and the connection status are displayed below. If you want to reconfigure your wireless settings, please click the "Configure" button. You can also enter advanced settings by clicking "Manual Setup".

Internet Settings

Internet Connection : Dynamic IP (DHCP)
Status : Connected

Wireless Settings

Network Name (SSID) : dlink
Status : Encryption

Security : Auto (WPA or WPA2) - Personal

Network Key : 123456789

Device Info

User Name : admin

Password :

mydlink Account

You have not activated mydlink service.
Status : Not Connected

☐ Save my network settings

Cancel
Manual Setup
Save

WIRELESS

Fig. 3.26 (a): Router configuration

Step 5. Open the Internet browser and type the default gateway address on the address bar. Login to the account and the router details' window will open. Figure 3.26 (b), (c), (d), (e) shows the screenshots of router settings.

Product Page: DIR-855

Hardware Version: A2

Firmware Version: 1.12

DIR-855

SETUP

ADVANCED

TOOLS

STATUS

SUPPORT

INTERNET

WIRELESS SETTINGS

NETWORK SETTINGS

USB SETTINGS

WIRELESS

Use this section to configure the wireless settings for your D-Link Router. Please note that changes made on this section may also need to be duplicated on your Wireless Client.

Save Settings

Don't Save Settings

WIRELESS NETWORK SETTINGS

Wireless Band : 2.4GHz Band

Enable Wireless : ☒

Always

New Schedule

Wireless Network Name : My Wifi Router (Also called the SSID)

802.11 Mode : Mixed 802.11n, 802.11g and 802.11b

Enable Auto Channel Scan : ☒

Wireless Channel : 2.437 GHz - CH 6

Transmission Rate : Best (automatic) (Mbit/s)

Channel Width : 20 MHz

Visibility Status : ☒ Visible ☐ Invisible

Helpful Hints...

Changing your Wireless Network Name is the first step in securing your wireless network. Change it to a familiar name that does not contain any personal information.

Enable Auto Channel Scan the router can select the best possible channel for your wireless network to operate on.

Enabling Hidden Mode is another way to secure your network. With this option enabled, no wireless clients will be able to see your wireless network when they scan to see what's available. For your wireless devices to connect to your router, you will need to manually enter the Wireless Network Name on each device.

Fig.3.26 (b): Router settings

Product Page: DIR-855

Hardware Version: A2

Firmware Version: 1.12

DIR-855

SETUP

ADVANCED

TOOLS

STATUS

SUPPORT

VIRTUAL SERVER

PORT FORWARDING

APPLICATION RULES

QOS ENGINE

NETWORK FILTER

ACCESS CONTROL

WEBSITE FILTER

INBOUND FILTER

FIREWALL SETTINGS

ROUTING

ADVANCED WIRELESS

WISH

WI-FI PROTECTED SETUP

ADVANCED NETWORK

GUEST ZONE

PORT FORWARDING

This option is used to open multiple ports or a range of ports in your router and redirect data through those ports to a single PC on your network. This feature allows you to enter ports in various formats including, Port Ranges (100-150), Individual Ports (80, 68, 888), or Mixed (1020-5000, 689). This option is only applicable to the INTERNET session.

Save Settings

Don't Save Settings

24 -- PORT FORWARDING RULES

		Ports to Open		
☐	Name	<< Application Name	TCP	Schedule Always
	IP Address	<< Computer Name	UDP	Inbound Filter Allow All
☐	Name	<< Application Name	TCP	Schedule Always
	IP Address	<< Computer Name	UDP	Inbound Filter Allow All
☐	Name	<< Application Name	TCP	Schedule Always
	IP Address	<< Computer Name	UDP	Inbound Filter Allow All
☐	Name	<< Application Name	TCP	Schedule Always
	IP Address	<< Computer Name	UDP	Inbound Filter Allow All

Helpful Hints...

Check the **Application Name** drop down menu for a list of predefined applications. If you select one of the predefined applications, click the arrow button next to the drop down menu to fill out the corresponding field.

You can select a computer from the list of DHCP clients in the **Computer Name** drop down menu, or you can manually enter the IP address of the LAN computer to which you would like to open the specified port.

Select a schedule for when the rule will be enabled. If you do not see the schedule you need in the list of schedules, go to the **Tools** → **Schedules** screen and create a new schedule.

You can enter ports in various formats:
Range (50-100)
Individual (80, 68, 888)
Mixed (1020-5000, 689)

More...

Fig. 3.26 (c): Router settings

Telecom Technician IoT Devices/ System (TEL-Q6210) Grade-XII

Page 60 / 184

DIR-855	SETUP	ADVANCED	TOOLS	STATUS
DEVICE INFO LOGS STATISTICS INTERNET SESSIONS WIRELESS WISH SESSIONS	DEVICE INFORMATION All of your Internet and network connection details are displayed on this page. The firmware version is also displayed here.			
	GENERAL Time : Saturday, January 31, 2004 11:14:11 AM Firmware Version : 1.12, 2008/09/10			
	WAN Connection Type : DHCP Client QoS Engine : Active Cable Status : Disconnected Network Status : Disconnected Connection Up Time : N/A Renew Release MAC Address : 00:21:91:19:03:45 IP Address : 0.0.0.0 Subnet Mask : 0.0.0.0 Default Gateway : 0.0.0.0 Primary DNS Server : 0.0.0.0			

Fig. 3.26 (d): Router settings

Product Page: DIR-855		Hardware Version: A2 Firmware Version: 1.12	
D-Link			
DIR-855 ADMIN TIME SYSLOG EMAIL SETTINGS SYSTEM FIRMWARE DYNAMIC DNS SYSTEM CHECK SCHEDULES	ADMINISTRATOR SETTINGS The 'admin' and 'user' accounts can access the management interface. The admin has read/write access and can change passwords, while the user has read-only access. By default there is no password configured. It is highly recommended that you create a password to keep your router secure. Save Settings Don't Save Settings		Helpful Hints... For security reasons, it is recommended that you change the password for the Admin and User accounts. Be sure to write down the new and passwords to avoid having to reset the router in case they are forgotten. Enabling Remote Management, allows you or others to change the router configuration from a computer on the Internet. Choose a port to open for remote management. Select a filter that controls access as needed for this admin port. If you do not see the filter you need in the list of filters, go to the Advanced → Inbound Filter screen and create a new filter. More...
	ADMIN PASSWORD Please enter the same password into both boxes, for confirmation. Password : Verify Password :		
	USER PASSWORD Please enter the same password into both boxes, for confirmation. Password : Verify Password :		
	SYSTEM NAME Gateway Name : D-Link Systems DIR-855		
	ADMINISTRATION Enable HTTPS Server : ☐ Enable Remote Management : ☐ Remote Admin Port : 8080 Use HTTPS : ☐ Remote Admin Inbound Filter : Allow All Details : Allow All		
WIRELESS			
Copyright © 2004-2008 D-Link Systems, Inc.			

Fig. 3.26 (e): Router settings

Step 6. After initializing the node device which is an IoT camera in the above-given case and the gateway which is a D Link router in the above case, the technician should check the connectivity of the node devices to the gateways.

3.2 Testing of network connectivity

After establishing the network connection, it should be tested for proper functioning. The network connectivity can be tested by **ipconfig** and **ping** command. Follow the steps below to test the network connectivity.

Step 1. Enter the command **ipconfig** on the command line and press enter. If the screen shows the connection details, it means that there is no problem with the network connection.

Step 2. In the same way enter the **ping** command followed by <gateway address> to test. If it executes, it shows that there is no loss of packets and the router is working well.

Testing the network connectivity in Ubuntu Linux is shown in Figure ...

Practical Activity 3.3. Demonstrate to check the connection of the devices to the gateway.

Material required

Router, computer, Ethernet cable, sensor device

Procedure

Step 1. Launch the IoT device software installed on the laptop or desktop and open the default web server page. Figure 3.27 (a) and (b) show the screenshot of the 6LoWPAN Border router details along with the system information, sensors connected, network status, and so on:



Fig. 3.27 (a): Internet settings

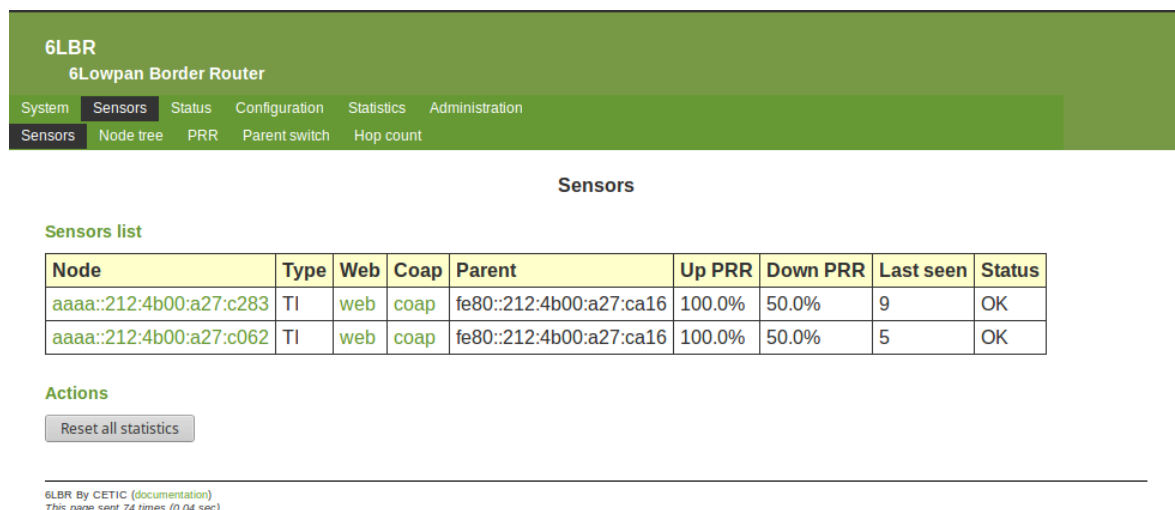


Fig. 3.27 (b): List of sensors connected

Step 2. Check that the status of each device and sensor is "OK". This means that the devices are connected to the gateway or the router.

Practical Activity 3.4. Demonstrate to connect a physical device to the gateway and to test the

connectivity

Material required

Microsoft Azure suite installed computer, router, Ethernet cable, sensor device

Procedure

Step 1. To do so in the Microsoft Azure suite, click on the devices to check the device parameters. Figure 3.28 shows the connected devices in the Microsoft Azure window.

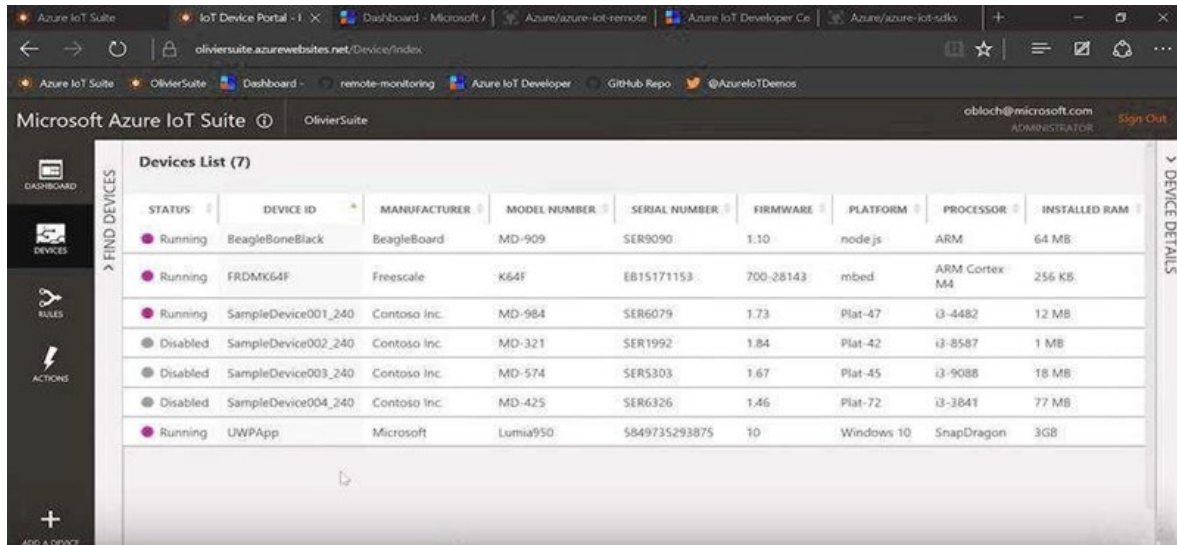


Fig. 3.28: Microsoft Azure window

Step 2. Click on “Add New” as shown in Figure 3.29.

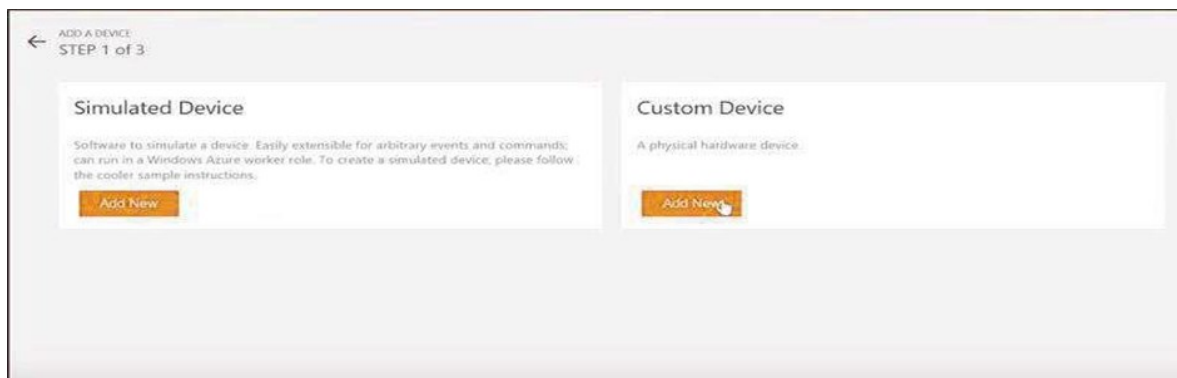


Fig. 3.29: Add new device

Step 3. Define the device ID as shown in Figure 3.30.

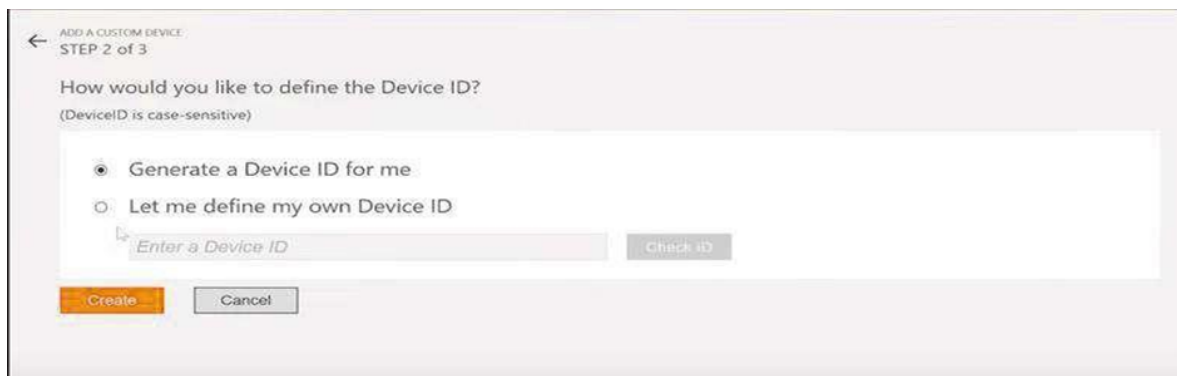


Fig. 3.30: Defining device ID

Step 4. Configure the device ID and IoT hub name as shown in Figure 3.31.

ADD A CUSTOM DEVICE
STEP 3 of 3

Copy credentials into the configuration file on the device

Device ID:

IoT Hub Hostname:

Device Key:

Fig. 3.31: Configuring the device ID and IoT hub name

Step 5. Set the command for the device as shown in Figure 3.32.

COMMAND

MESSAGE

Command History

COMMAND NAME	RESULT	VALUES SENT	TIME CREATED	TIME UPDATED
No data available in table				

Fig. 3.32: Setting command for the device

Step 6. Check whether the device is functioning according to the command as shown in Figure 3.33.

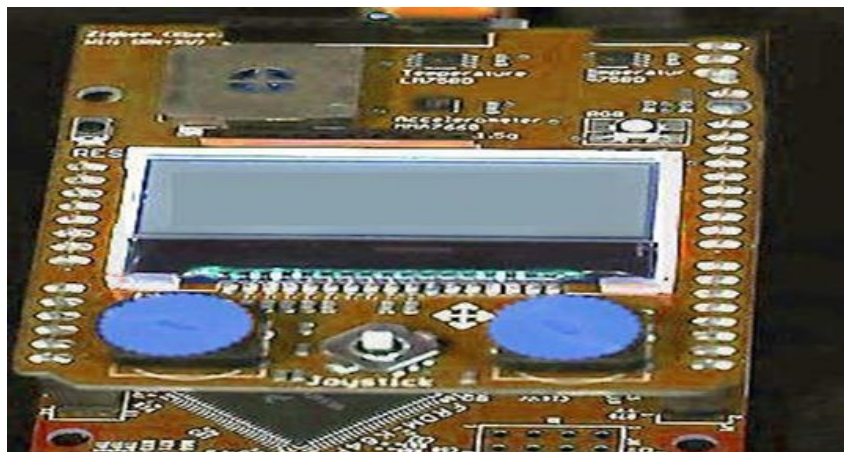


Fig. 3.33: Checking whether the device is functioning according to the command

3.3 Software Execution

The software is executed after the node devices and gateway (routers) have been initialized. The software is run in the final step, after which the device is joined to the communication channel.

Consider an IoT camera device that requires a connection to a ZigBee communication channel.

The following steps should be performed for connecting the device and executing the software

Step 1. Connect a USB cable to the ZigBee coordinator on the camera device and the computer system. Figure 3.34 shows the connection:

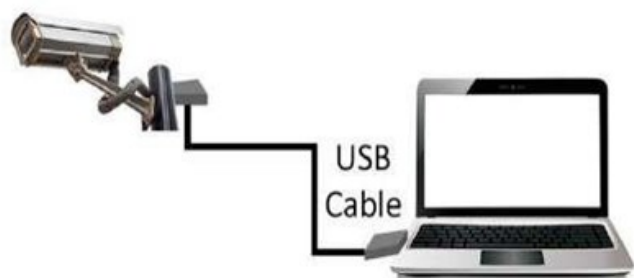


Fig. 3.34: Connection between camera and laptop with a USB cable

Step 2. Virtual COM port drivers are required for the ZigBee Mesh module; which can be downloaded through the link for Windows:

<http://www.ftdichip.com/Drivers/VCP.htm>

Step 3. Download the National Control Device (NCD) Base Station software to access the module; which will help to test devices and access them. The link for the software is given as: <http://ncd-base-station.software.informer.com/>

Step 4. Run the Base Station software for setting up the device. Figure 3.35 shows the selection window for setting up Base Station.

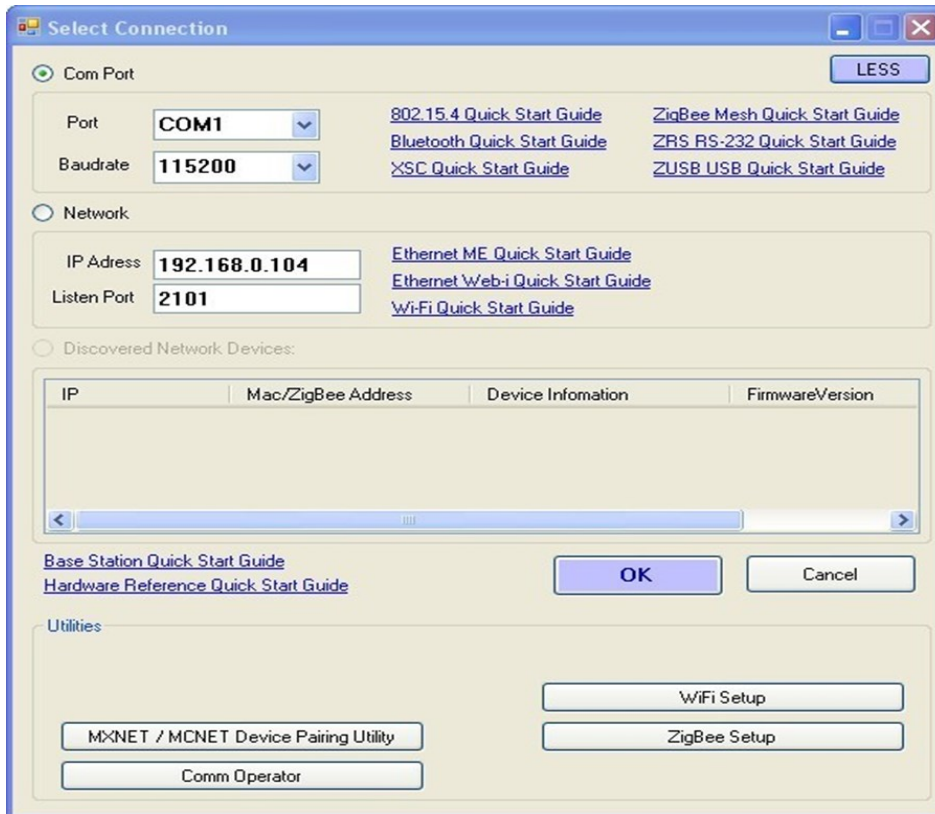


Fig. 3.35: Selection window for setting up Base Station

- Run Base Station.
- Select appropriate Com Port for ZigMo (ZigBee Coordinator).
- Click ZigBee Setup.
- Click Refresh.
- The progress bar will show for searching the device status.

Select the device in the list. Figure 3.36 shows the device selection window.

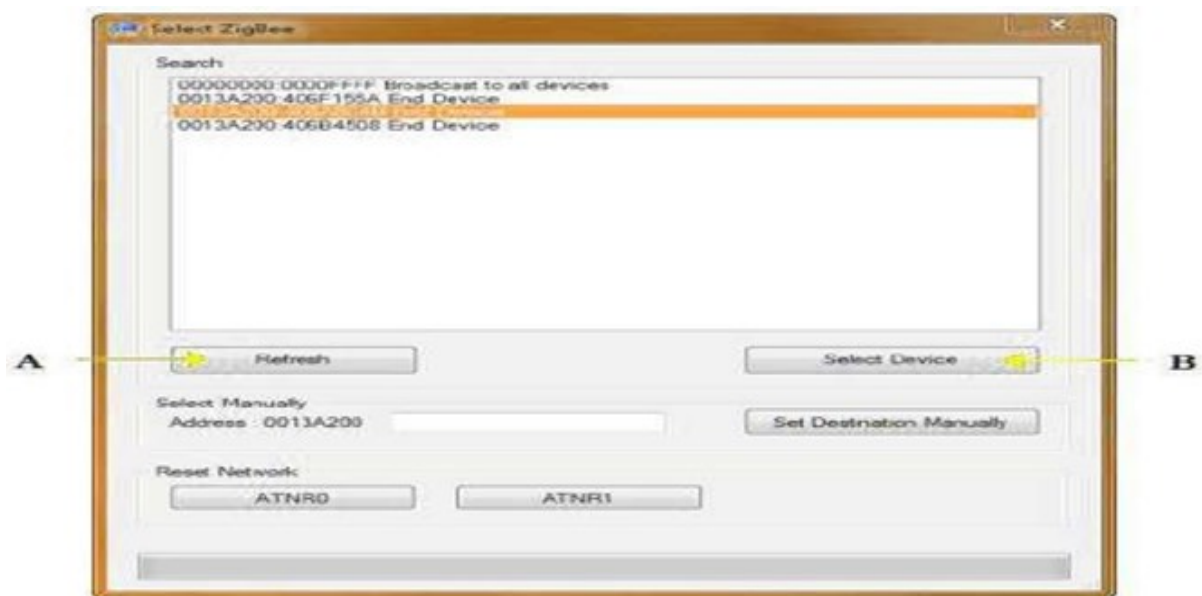


Fig. 3.36: Device selection window

- Click select device which is marked as (B) in the above screenshot.
- Progress bar will indicate that the device is loading.
- Click 'OK' in the alert box which appears.
- Close the Base Station window after the device is ready to use. Figure 3.37 shows the NCD configuration window.

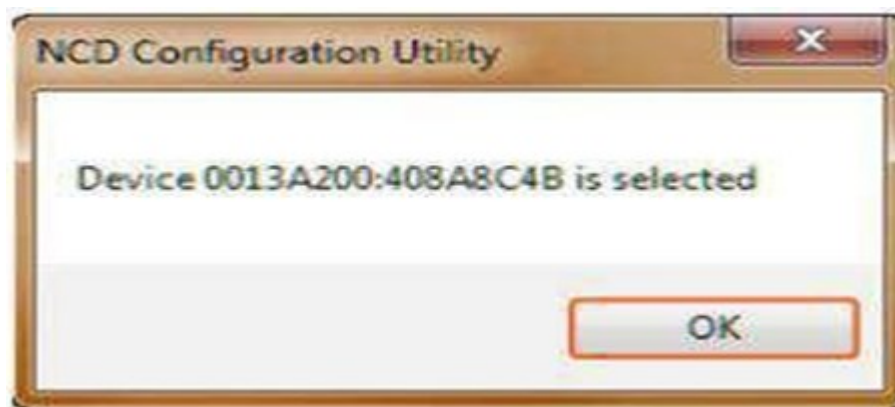


Fig. 3.37: NCD configuration window

5. Run the Base Station software to control the relays, read the A/D inputs, or start communicating with the remote device as follows:

- Run the Base Station software
- Now, select the appropriate Com Port on the opening '**Select Connection**' window
- Click **OK**

6. Then, the Base Station software will generate a list of commands which the controller can process.

7. Now, select the command and use the device. Figure 3.38 shows the command option window.

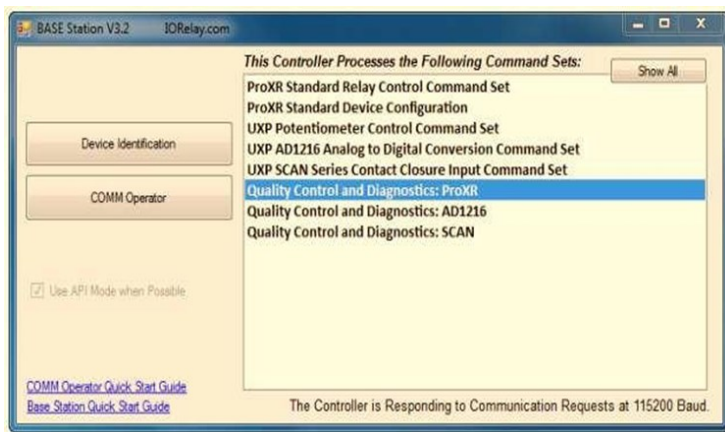


Fig. 3.38: Command option window

Summary

- This chapter covers the essential steps for setting up and configuring nodes, gateways, and control edge appliances in an IoT system.
- It emphasizes the importance of proper prerequisites, such as having routers, IoT devices, and internet connections in place.
- The process of installing an IoT camera and initializing nodes and gateways is given in detailed form in this chapter.
- Additionally, it outlines how to test network connectivity and execute necessary software for devices.
- This chapter provides a comprehensive guide for establishing a strong foundation in an IoT network.

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. What is the first step in initializing a node or gateway in an IoT setup? (a) Setting up a username and password (b) Unpacking the device (c) Connecting the power supply (d) Configuring the Internet connection
2. Which tool is recommended for manually connecting the camera in case automatic connection fails? (a) EnViewer Finder (b) NCD Base Station (c) Genius Vision NVR (d) ZigBee Coordinator
3. What is the purpose of mounting the camera on a wall? (a) To improve Wi-Fi signal strength (b) To enhance video quality (c) To cover a specific surveillance area adequately (d) To reduce power consumption
4. How can a technician verify that a camera is positioned within the router's coverage area? (a) Check the camera's packaging (b) Press the WPS button on the camera (c) Use EnViewer Finder tool (d) Compare installation kit specifications
5. What is the function of the WPS (Wi-Fi Protected Setup) button during camera setup? (a) Connect the camera to the power supply (b) Establish a wireless Internet connection (c) Initialize the camera's hardware (d) Mount the camera on a wall
6. Which mode of network connection requires providing an IP address manually? (a) DHCP (b) Manual (c) PPPoE (d) WPS
7. What is the purpose of unpacking the router device? (a) Matching specifications with the camera (b) Connecting the Ethernet cable (c) Checking for accessories and damage (d) Configuring the Internet connection
8. How can a technician access the router settings for configuration? (a) Open a web browser and enter the router's IP address (b) Run the NCD Base Station software (c) Connect the router with an Ethernet cable (d) Insert the utility tool disc
9. What does node initialization involve in an IoT setup? (a) Configuring network settings and securing the network (b) Connecting the power supply and cable (c) Mounting the camera on a wall (d) Unpacking the router
10. Which step follows the initialization of a node device in the provided content? (a) Testing network connectivity (b) Unpacking the router device (c) Configuring the router device (d) Installing the router

B. Fill in the blanks

1. To finish the cable connection, join the power adaptor end to the power supply port on the camera's _____.
2. The IoT camera must be connected to the router in order for it to function over a wireless Internet connection using _____.
3. If the automatic connection fails, establish a manual connection using a utility program like _____ that was included on the CD that came with the IoT camera's packaging.
4. To configure the router, a technician can connect the router with a desktop/laptop through a LAN wire or with the help of the _____ address.
5. To establish an Internet connection between the network provider and the gateway, which is the router installed at the location, the gateway or router must first be _____.
6. The software is executed after the node devices and gateway (routers) have been initialized. The software is run in the final step, after which the device is joined to the _____ channel.
7. After completing the hardware setup of the IoT device and connecting it with the Internet, the final step is to initialise _____.
8. Unpack the NVR device and connect the device to the router using an _____.
9. Connect a USB cable to the ZigBee coordinator _____ and the computer system.
10. Consider an IoT camera device that requires a connection to a _____.

C. State true or False for the following

1. Setting up a username and password on routers and IoT devices is known as initializing a node and gateway.
2. Unpacking the IoT camera is the first step in the installation process.
3. Mounting the camera and the camera stand are part of the fundamental hardware setup.
4. The power adaptor typically included in the packaging has a power output of 12 V/1 A.
5. The IoT camera must be connected to the router for it to function over a wireless Internet connection using Wi-Fi.
6. If the automatic connection fails, a technician can establish a manual connection using a utility program like EnViewer.
7. Unpacking the router is not necessary if the technician receives it from the Internet service provider (ISP).
8. Connecting the power cable end to the back of the router and the adapter plug to a power socket is part of the router installation process.
9. To configure the router, a technician can connect it to a desktop/laptop through a LAN wire or with the help of the IP address.
10. The final step after initializing the node devices and gateways is to execute the software, after which the device is joined to the communication channel.

D. Short Answer Type Questions

1. What is the purpose of initializing a node or gateway in an IoT setup?
2. Describe the steps involved in setting up the basic hardware of an IoT camera.
3. How can a technician connect the IoT camera to a wireless Internet connection using Wi-Fi?
4. What should be done if the automatic connection of the camera to the router fails?
5. What are the steps involved in unpacking and setting up a router for a wireless Internet connection?
6. How can a technician access the router settings for configuration?
7. What is the importance of node initialization in an IoT setup?
8. How does the technician test the network connectivity after initializing the node devices and gateways?
9. Explain the process of executing the software after initializing the node devices and gateways.
10. How does a technician connect a USB cable to the ZigBee coordinator on the camera device for software execution?

Chapter 4. Establish Communication and Connectivity between devices

Establishing communication and connectivity between devices is a fundamental aspect of data transfer in various technological applications. The process involves creating a reliable link between devices to facilitate the exchange of information.

Devices need a physical means of connection, such as cables or wireless technologies (e.g., Wi-Fi, Bluetooth, NFC, or Infrared). The choice of the connection method depends on factors like data transfer speed, range, and the nature of the devices.

Modern motherboards, single-board computers, embedded microcontroller boards, and nearly every digital peripheral—from standard computer peripherals like keyboards, mice, and joysticks to smart devices like cameras, flash drives, smartphones, and tablets—all have USB ports. USB is the standard and must-have interface.

4.1 Types of USB Data Transfer

Data transmission from one device to another is referred to as data transfer. There are four types of transfer modes or types which can be used for data transfer via USB as shown in Figure 4.1.

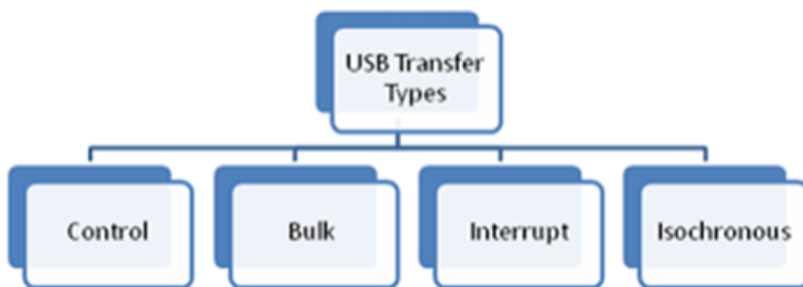


Fig.4.1: Types of USB Data Transfer

Control Transfer

Control transfer is driven by rules about the content of the data to be transferred. It is used to allocate USB addresses, exchange device details and configure devices. All the devices use control transfer along with the other ones.

Bulk Transfer

Digital cameras, printers, scanners, and other image input, printing, and storage devices are necessary for the lossless transfer of large amounts of data. Highly reliable data transfers are made possible by bulk transfer. The availability of buses determines the transfer rate. Therefore, it is not appropriate for applications that need to strictly control the data transfer rate.

Interrupt transfer

This is typically used by input devices like keyboards and mouse that are used as human interfaces. The input signals that are detected are treated as interrupt requests. In case of USB, the host starts the processes as interrupt requests. In order to avoid agitating the user, the host periodically "polls" the input device to display keyboard inputs on the screen. "Interrupt transfer" is the process by which the host periodically transmits the data.

Isochronous Transfer

The audio and video devices require real-time data transfer. Such devices must be able to periodically transfer a certain amount of data. For the purpose of carrying out data transfers, USB divides time into frames. The main idea is to maintain a steady flow of time while transferring a fixed amount of data over each time interval.

It is depicted with examples in Figure 4.2.

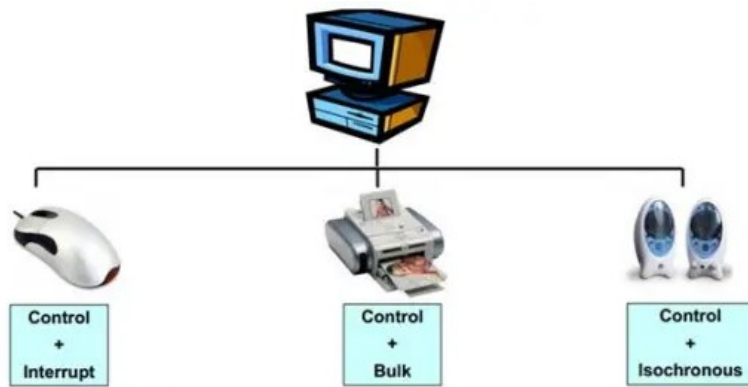


Fig.4.2: Example of USB Data Transfer

4.2 Data Transfer Modes

Communication technology deals with the mode of transfer of data. Mode refers to the direction of data flow over the network. There are three types of modes as follows:

4.2.1 Simplex

In simplex mode, the sender is able to send data but not receive it. It is a form of unidirectional communication in which only one direction is involved. Examples of this mode include the keyboard and conventional monitors. The direction of data transfer is shown in Figure 4.3.

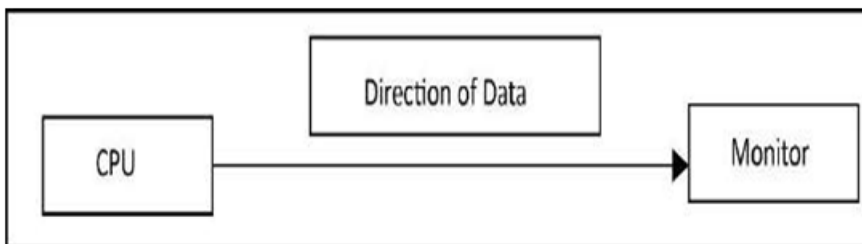


Fig.4.3: Simplex mode

4.2.2 Half Duplex mode

The sender can send and receive data one at a time while in half-duplex mode. It is a form of two-way directional communication, but it can only be used by one person at once. The Walkie-Talkie is an illustration of this type of transmission because it sends each message one at a time but in both directions. It is depicted in Figure 4.4.

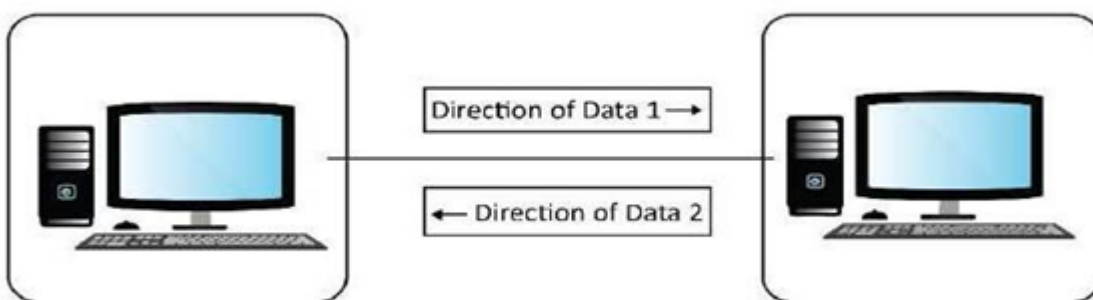


Fig.4.4: Half duplex mode

4.2.3 Full Duplex Mode

In Full-duplex mode, the Sender can send the data and also receive the data simultaneously. It is two-way directional communication that can happen simultaneously. An example of this kind of transmission is the Telephone Network, where communication happens simultaneously as shown in Figure 4.5.

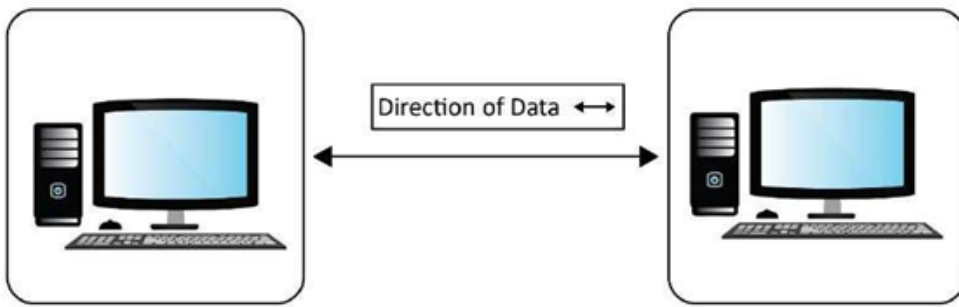


Fig.4.5: Full Duplex Mode

4.3 Controlling the Data Transfer

Data transfer can be controlled by changing the baud rate of the Arduino or the Raspberry Pi. Figure 4.6 shows the baud rate of a setup.

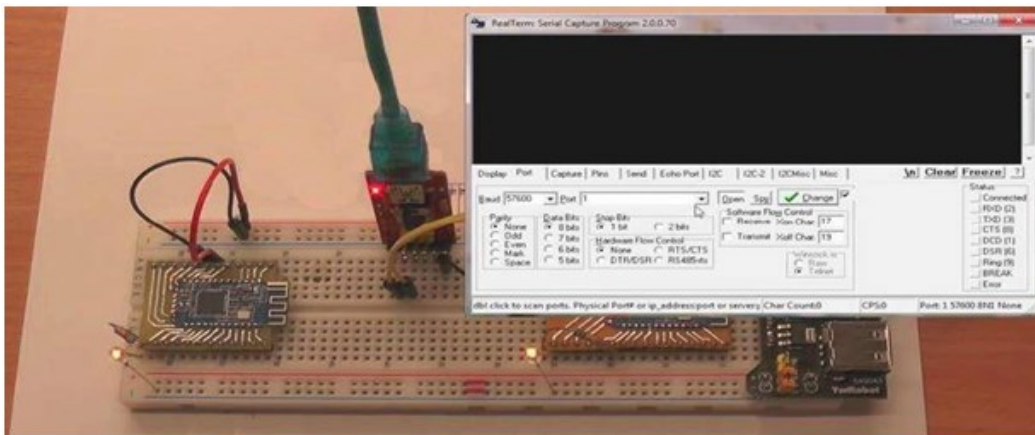


Fig. 4.6: Baud rate of a setup

To control the data transfer rate for a router, follows the steps:

Step 1. Open the router settings by entering the username and the password.

Step 2. Visit the “Guest Network” settings and enter the details. For controlling the transfer rate, the bandwidth needs to be set. Figure 4.7 shows the wireless settings for TP-Link.



Fig. 4.7: Wireless settings for TP-Link

4.4 Data Transfer Using the Indicators

The lights on devices, including routers, gateways, and other similar equipment, assist in determining the status of the equipment's operation. According to the system test manual, all data transfer and communication using appropriate LED indications or on screen I/O streams are inspected. The various indicators on a device are listed in table 4.1.

Table 4.1 Different indicators on the device

	<i>Solid</i>	<i>Blinking</i>	<i>Off</i>
Battery	Battery is good	The battery is not good	No Battery
Wi-Fi	Wireless network enabled	Connection not stable due to traffic on the network	Network failure or disabled
Ethernet	Device connected to the Ethernet port	Data is being transmitted over Ethernet link	Device not connected to the Ethernet port
Online	Internet available	-----	No Internet available
Upstream (US)	Yellow/ Green: Connected to the Internet	Not connected to the Internet	-----
Downstream (DS)	Yellow / Green: Connected to the Internet	Not connected to the Internet	-----
Power	AC power is available	-----	There is no AC power

4.5 Comparison of data transfer techniques over various networks

The mechanism and techniques of data transfer depend on the type of network.

The table 4.2 lists the comparison between the data transfer over various networks:

Table 4.2 Comparison between the data transfer over various networks

	<i>ZigBee</i>	<i>Bluetooth LE</i>	<i>Z-Wave</i>	<i>NFC</i>	<i>Wi-Fi</i>
IEEE Specification	802.15.4	802.15.1	ITU -T	ISO 13157	802.11 a/b/g
Frequency Band	868/915 MHz; 2.4 GHz	2.4 – 2.5 GHz	908.42 MHz	13.56 MHz	2.4 GHz; 5 GHz
Network Type	WPAN	WPAN	WPAN	P2P	WPAN
Power Consumption (mA)	40	12.5	2.5	50	116
Nominal Range (m)	10	50	30	.05	100
Max. Signal Rate	250 kbps	305 kbps	40-100 kbps	424 kbps	54 Mbps

To check the network speed and data transfer rates, there are many tools available. By selecting "Ethernet" and then "Properties" in Windows OS, it can be verified. An Ethernet network's status is displayed as shown in Figure 4.8.

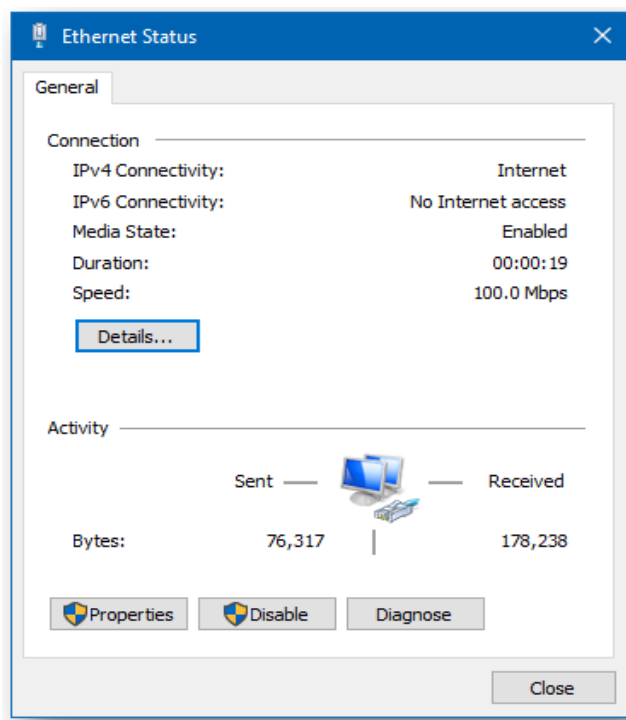


Fig. 4.8: Network status

4.6 Causes of Data Transfer Failure

Data transfer failure can occur due to bad network connection or improper connection and configuration. The following table lists some scenarios of data transfer failures:

Issue	Cause / Solution
It takes time to move data from a computer to the cloud longer than anticipated.	The computer is missing a gigabit Ethernet port. adapter. Gigabit is not supported by the hardware or the network. Wi-Fi local network overhead or line of sight is present between devices. A network hub rather than a switch is being used. Data protection has an impact on data transfer speed. malware protection, antivirus software, etc. A sizable multi-terabyte data set is present. The USB port or power source the device is using is malfunctioning. or a low-quality USB cable.
The device is not in range of the network.	Connect the computer to the network using an Ethernet cable. A wired connection is preferable to a direct one; pertaining to efficiency and transfer rates. If an Ethernet connection is possible, disable the Wi-Fi network.

Performance problems are caused by faulty network hardware. Inconsistent network speeds are the result.	A USB 3.0 or higher port along with a high-quality USB cable should be used. Faulty and outdated networking devices should not be used.
Network hubs can result in network packet collisions, auto-negotiation mismatches, and packet drops.	Network hubs should be replaced with Gigabit switches.
Outdated firmware can affect network performance.	Network equipment firmware must be updated and outdated hardware should be replaced.
Data transfer over 2.4 and 5 GHz Wi-Fi bands are much slower than wired Ethernet.	Direct connection using Ethernet cable is better than the wireless one. If an Ethernet connection is not possible, connect using the 5 GHz band.
Third-party applications contribute to network traffic by scanning and downloading content.	Ensure third party applications are not indexing or virus scanning.
Backup functions consume CPU, memory, and disk access resources and decrease data transfer rates.	Stop the backup functions while transferring content.

4.7 Connecting to the Network Remotely

An IoT framework must have external connectivity in order to give remote users access to the central site. A connectivity strategy that enables both site-to-site and remote client connectivity must be implemented for this. The gateway and the neighbourhood Wi-Fi router or 3G/4G connectivity options can be connected.

Practical Activity 4.1. Demonstrate to access a router or any device remotely.

Material Required

TP link wireless router, computer

Procedure

Step 1. Open the web browser and enter the router address. The default address is 192.168.0.1. For any device, such as a surveillance camera, enter the device's IP address. Figure 4.9 shows the default IP address at the address bar:

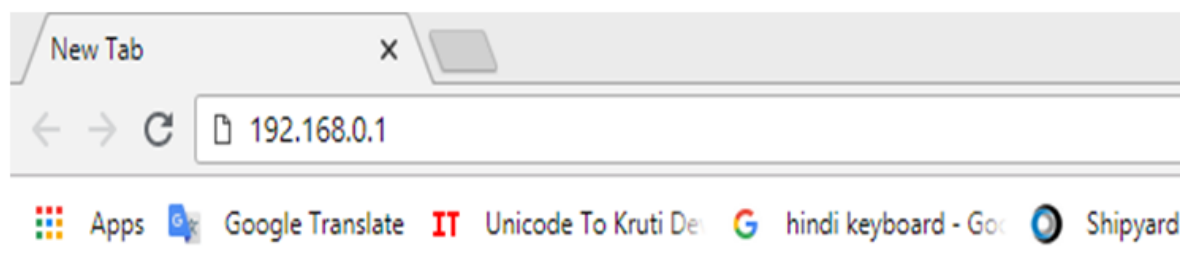


Fig. 4.9: IP address at address bar

Step 2. An authentication window will appear. Enter the user ID and the password in the window as shown Figure 4.10.

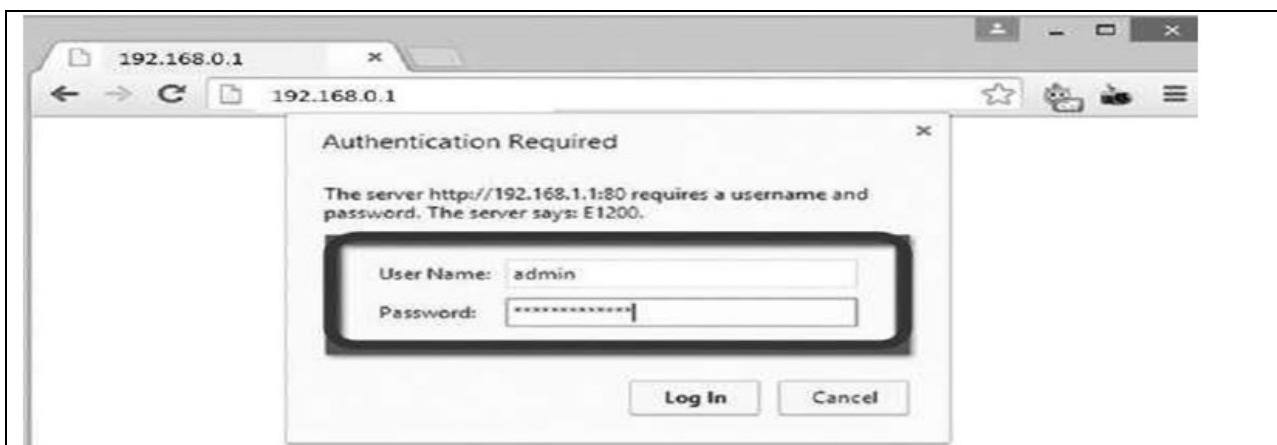


Fig. 4.10: Authentication window

Step 3. Check the network status and the details on the window as shown Figure 4.11.

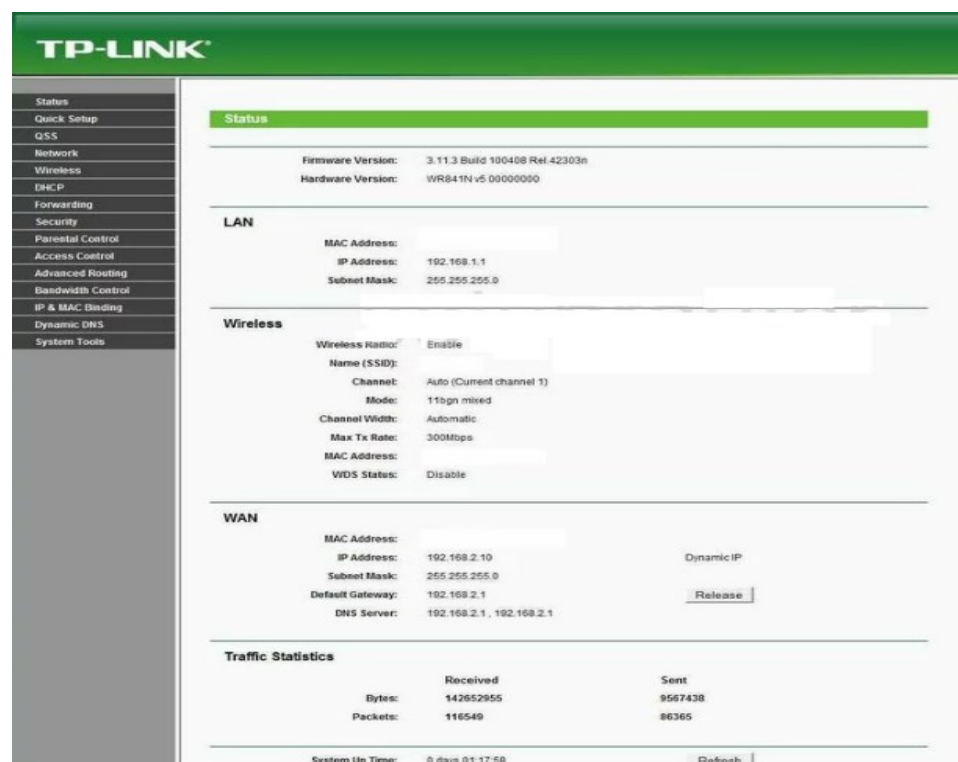


Fig. 4.11: Network status window

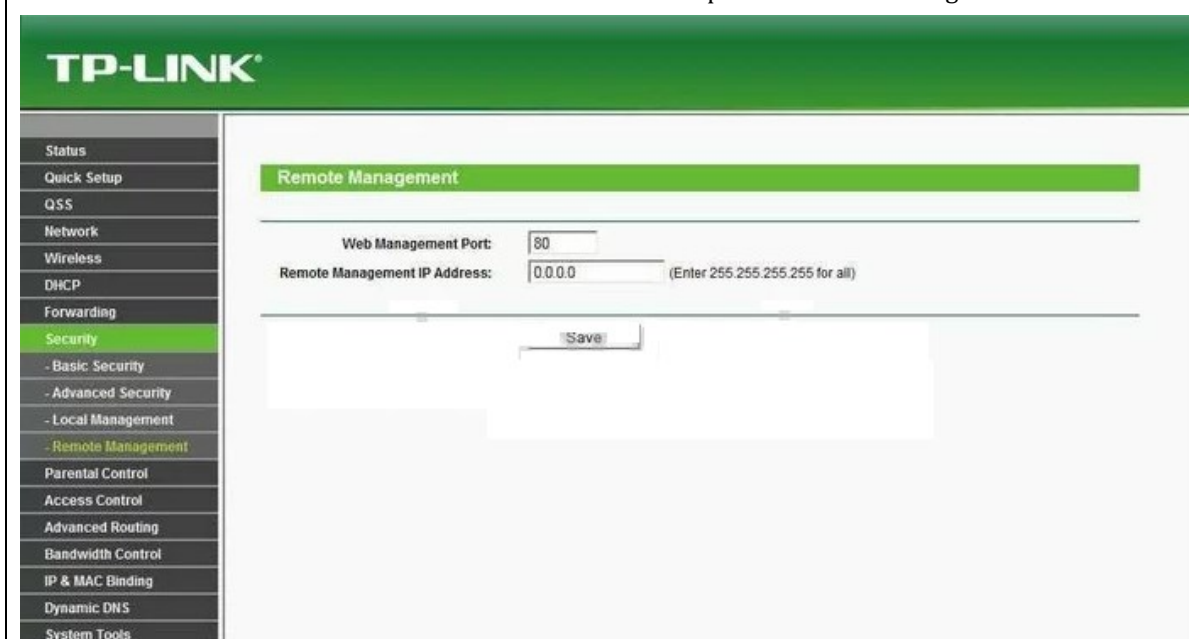
Step 4. Enter the security settings details and enable the security option suitable, as shown in Figure 4.12.



The image shows the TP-LINK web interface for Basic Security settings. The left sidebar contains a menu with options: Status, Quick Setup, QSS, Network, Wireless, DHCP, Forwarding, Security (highlighted), Parental Control, Access Control, Advanced Routing, Bandwidth Control, IP & MAC Binding, Dynamic DNS, and System Tools. Under the Security menu, there are sub-options: Basic Security (highlighted), Advanced Security, Local Management, and Remote Management. The main content area is titled 'Basic Security' and contains three sections: Firewall, VPN, and ALG. The Firewall section has a 'SPI Firewall' toggle set to 'Enable'. The VPN section has three toggles: 'PPTP Passthrough', 'L2TP Passthrough', and 'IPSec Passthrough', all set to 'Enable'. The ALG section has three toggles: 'FTP ALG', 'TFTP ALG', and 'H323 ALG', all set to 'Enable'. A 'Save' button is located at the bottom right of the main content area.

Fig. 4.12: Basic security settings

Step 5. Enter the details such as port and remote IP address in the “Remote Management” under security. The IP address will be 255.255.255.255 for the device to make it public as shown in Figure 4.13.



The image shows the TP-LINK web interface for Remote Management settings. The left sidebar is the same as in Figure 4.12, but the 'Remote Management' sub-option under Security is highlighted. The main content area is titled 'Remote Management' and contains two input fields: 'Web Management Port' with a value of '80' and 'Remote Management IP Address' with a value of '0.0.0.0'. A note next to the IP address field says '(Enter 255.255.255.255 for all)'. A 'Save' button is located at the bottom right of the main content area.

Fig. 4.13: Remote management settings for public access

Step 6. The IP address will be 0.0.0.0, for the device to deny any access, as shown in Figure 4.14.

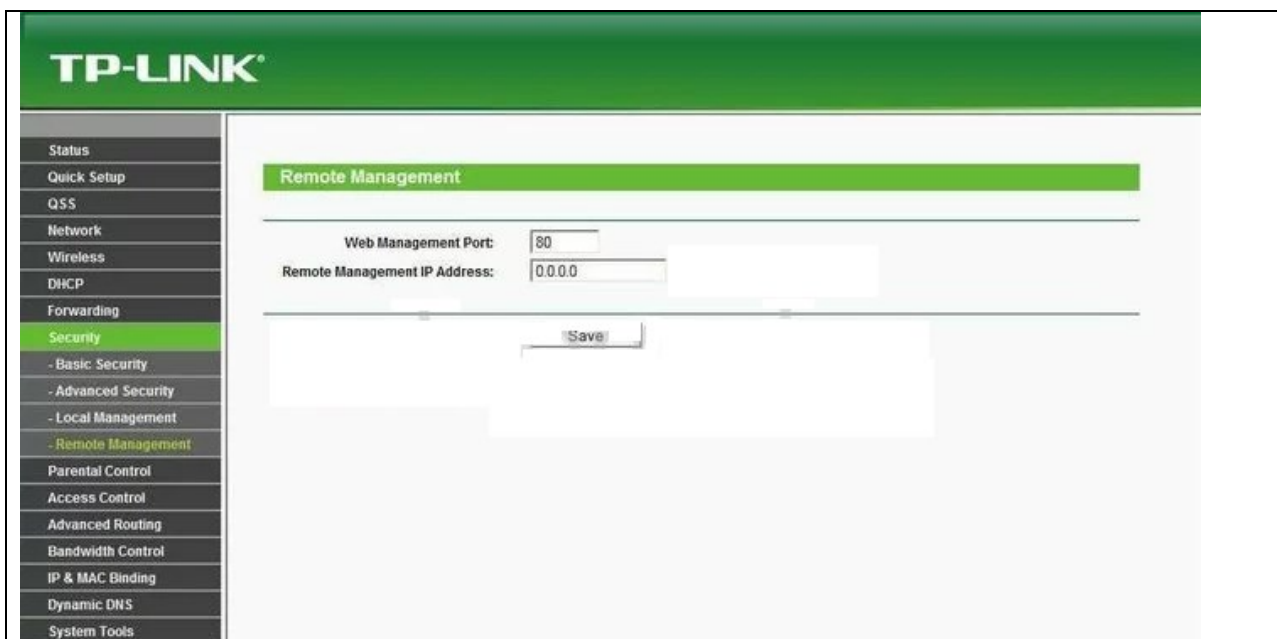


Fig. 4.14: Remote management settings for denying any access

Step 7. For a surveillance camera, the IP address must be entered in the address bar. The configuration such as network details and port numbers can be assigned to the device remotely. Figure 4.15 shows accessing the camera remotely.

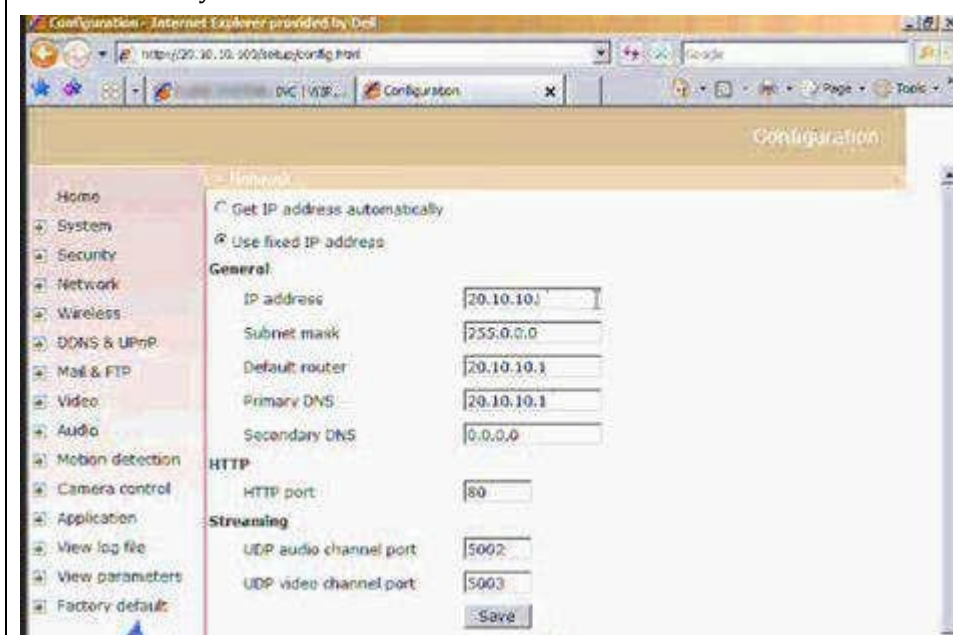


Fig. 4.15: Accessing the camera remotely

4.8 Connecting to Different Short- Range Wireless Networks

The devices can be connected to different wireless networks within the range. The wireless network may be Bluetooth, NFC, wireless router, ZigBee and so on.

Practical Activity 4.2. Demonstrate to establish a Bluetooth connection with a computer using the USB Bluetooth device on Raspbian OS.

Material Required

Raspbian OS installed computer, Bluetooth device

Procedure

Step 1. Connect a Bluetooth module to the Raspberry Pi and open the interface for the Raspberry Pi in the system. Figure 4.16 shows the interface window.

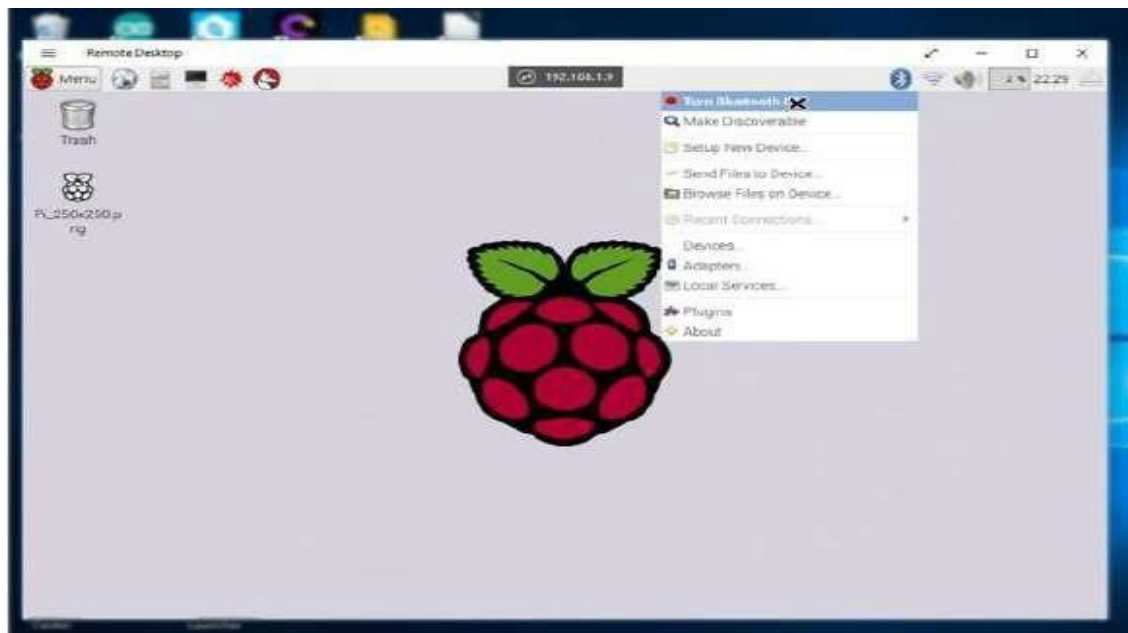


Fig. 4.16: Raspberry Pi interface window

Step 2. Open the Bluetooth in the system and turn it on. The Figure 4.17 shows turning the Bluetooth on:

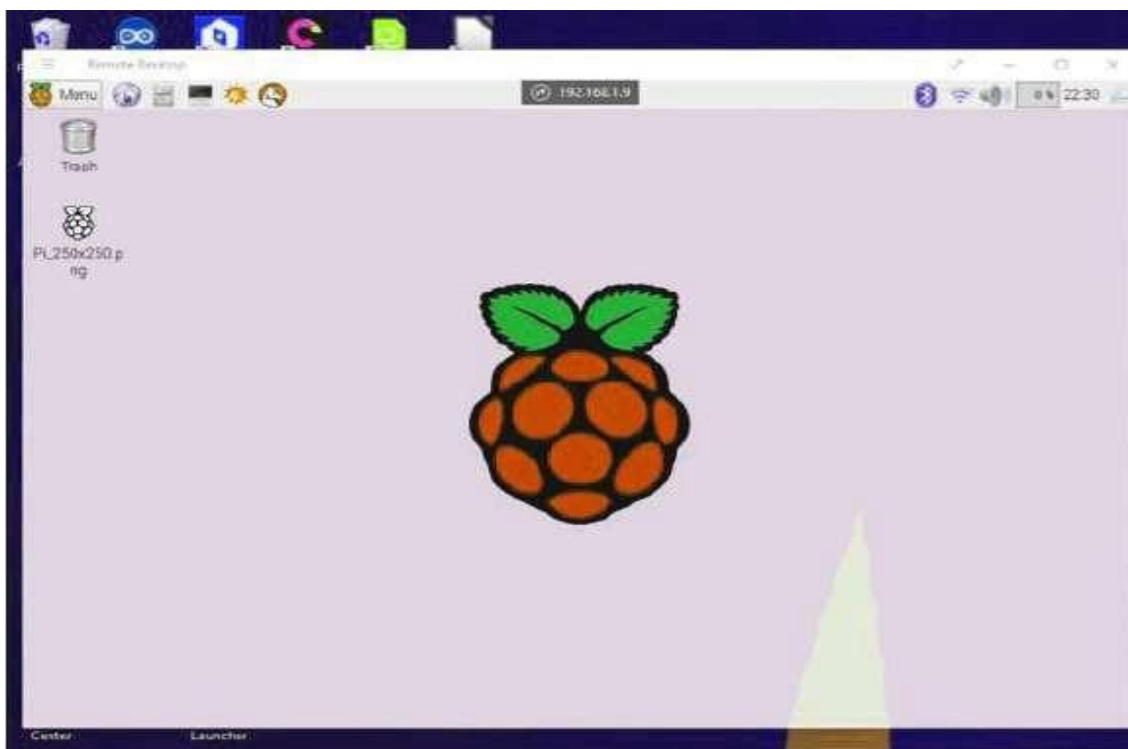


Fig. 4.17: Turning Bluetooth on

Step 3. Open the devices window and search for Bluetooth devices as shown in Figure 4.18.

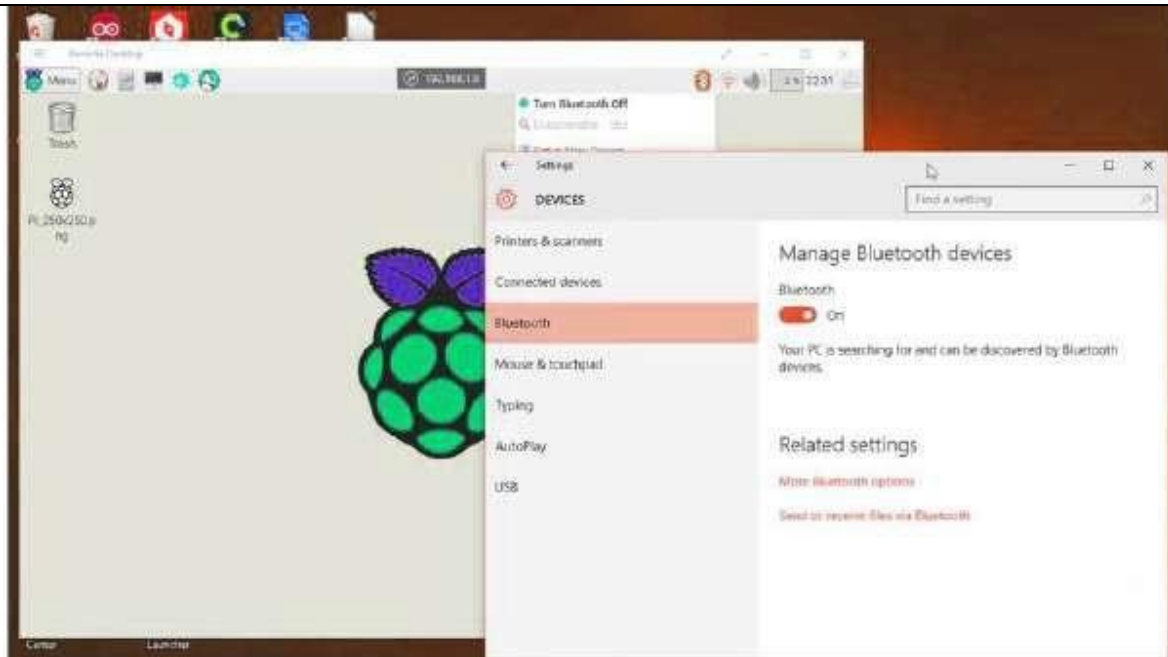


Fig. 4.18: Searching for Bluetooth devices

Step 4. Check for the Raspberry Pi and pair the devices as shown in Figure 4.19.

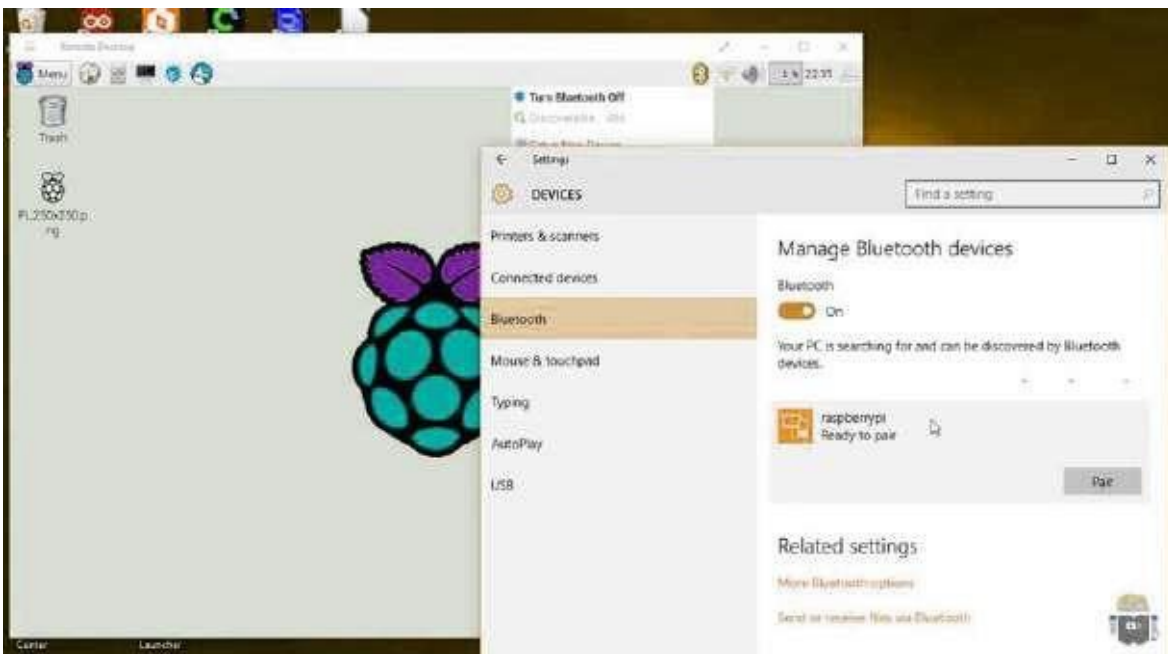


Fig. 4.19: Pairing the Bluetooth devices

Step 5. Check whether the passcodes are matching and confirm the pairing as shown in Figure 4.20.



Fig. 4.20: Checking the passcodes

Step 6. Click on the “Send/Receive files” for file sharing as shown in Figure 4.21.

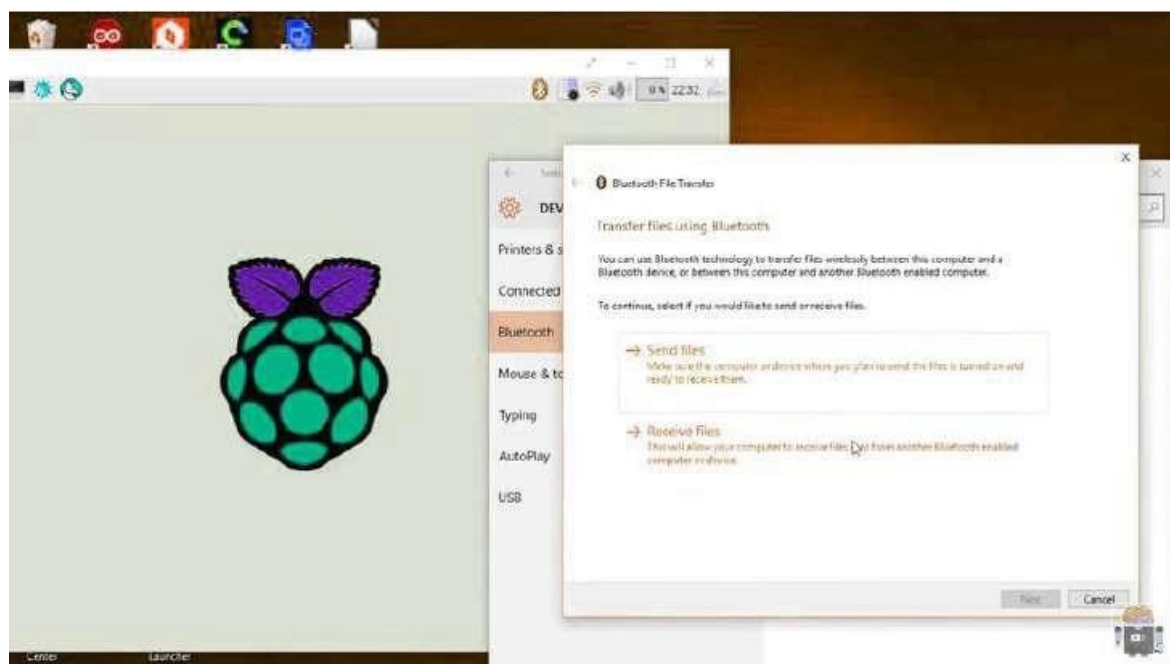


Fig. 4.21: Sharing the files

Practical Activity 4.2. Demonstrate the different data transfer modes and troubleshoot common data transfer issues.

Materials Required

Two laptops/desktops with Ethernet ports, Ethernet cables, Router

Step 1. Setting Up Devices:

- Connect one laptop (Device A) to the router using an Ethernet cable and ensure it has internet access.
- Connect the second laptop (Device B) to the router using another Ethernet cable.

Step 2. Exploring Data Transfer Modes:

Test Data Transfer Modes:

- a. Test Simplex Mode: Using Device A, send a file to Device B, but don't allow Device B to send anything back.
- b. Test Half Duplex Mode: Allow Device B to reply after receiving a file from Device A.
- c. Test Full Duplex Mode: Simultaneously send and receive files between Device A and Device B.

Step 4. Troubleshooting Data Transfer:

- a. Introduce common issues like loose Ethernet connections, outdated drivers, or interference for both wired and Bluetooth connections.
- b. Have participants troubleshoot and resolve these issues.

Step 5. Testing Remote Access:

- a. Access the router settings of Device A from Device B by entering the router's IP address in a web browser.
- b. Enable or disable remote management settings and observe the impact on access.

Summary

- This chapter covers data transfer types (Interrupt, Bulk, Isochronous, Control), modes (Simplex, Half Duplex, Full Duplex), and controlling transfer rates.
- It explains device indicators and troubleshooting.
- Additionally, it touches on remote network access and connecting via Bluetooth and ZigBee for IoT communication.

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. What type of data transfer is typically used by input devices like keyboards and mouse? (a) Bulk Transfer (b) Isochronous Transfer (c) Control Transfer (d) Interrupt Transfer
2. Which mode of data transfer allows the sender to send data and receive it one at a time? (a) Simplex (b) Half Duplex (c) Full Duplex (d) None of these
3. Which type of data transfer is used for audio and video devices requiring real-time data transfer? (a) Bulk Transfer (b) Isochronous Transfer (c) Control Transfer (d) Interrupt Transfer
4. What is the purpose of Control Transfer? (a) To transfer large amounts of data (b) To allocate USB addresses and configure devices (c) For lossless transfer of data (d) To periodically transfer a fixed amount of data
5. Which indicator on a device indicates that the battery is not good? (a) Solid (b) Blinking (c) Off (d) red
6. In which mode of transfer is communication simultaneous in both directions? (a) Simplex (b) Half Duplex (c) Full Duplex (d) Both b and c
7. What can cause data transfer failure due to bad network connection? (a) Faulty network hardware (b) Outdated firmware (c) Third-party applications (d) All of the above
8. How can an IoT framework provide remote user access to the central site? (a) Through gateway and neighbourhood Wi-Fi router (b) Through Ethernet connection (c) Through direct cable connection (d)
9. What is the purpose of connecting ZigBee modules to Arduino? (a) To transfer large amounts of data (b) To establish short-range wireless communication (c) To control data transfer modes (d)
10. Which network type is suitable for WPAN (Wireless Personal Area Network)? (a) ZigBee (d) Wi-Fi (c) NFC (d) All of the above

B. Fill in the blanks

1. Data transmission from one device to another is referred to as _____.
2. Digital cameras, printers, scanners, and other image input, printing, and storage devices require _____ for the lossless transfer of large amounts of data.
3. Control transfer is driven by rules about the content of the data to be transferred and is used to allocate _____.
4. In simplex mode, the sender is able to _____ data but not receive it.
5. In half-duplex mode, the sender can send and receive data _____ at a time.

6. In full-duplex mode, the sender can send and receive data _____.
7. The mechanism and techniques of data transfer depend on the _____ of the network.
8. Data transfer failure can occur due to bad network connection or improper connection and configuration. One common cause is the absence of a _____.
9. The devices can be connected to different wireless networks within the range, such as _____.
10. Control transfer is driven by rules about the _____ to be transferred.

C. State true or False for the following

1. Simplex mode allows data to be sent in both directions simultaneously.
2. Full Duplex mode allows simultaneous two-way communication.
3. Baud rate is a crucial factor in controlling data transfer for Arduino and Raspberry Pi.
4. Data transfer over 2.4 and 5 GHz Wi-Fi bands is generally faster than wired Ethernet.
5. Backup functions do not impact CPU, memory, and disk access resources during data transfer.
6. An IoT framework must have external connectivity to provide remote user access.
7. To access a router remotely, the user must enter the default address 192.168.0.1 in the web browser.
8. Bluetooth can be used to connect a Raspberry Pi to another system.
9. ZigBee modules need to be connected to an Arduino to establish a ZigBee network.
10. Port parameters play a crucial role in connecting ZigBee modules.

D. Short Answer Type Questions

1. What are the four types of data transfer?
2. Which types of devices typically use Interrupt Transfer for data transmission?
3. What kind of devices benefit from Bulk Transfer for data transfer?
4. Why is Isochronous Transfer important for audio and video devices?
5. Provide an example of a device that operates in Full Duplex mode.
6. How can the data transfer rate for a router be controlled?
7. What is the significance of the baud rate in controlling data transfer?
8. What are the steps involved in accessing a router or device remotely?
9. How can a Raspberry Pi be connected to another system via Bluetooth?
10. What is the purpose of installing ZigBee module software in the system?

As the Internet of Things (IoT) becomes integral to businesses, security emerges as a significant concern. Over half of future main business processes will involve some IoT component, heightening the importance of IoT system security, even for seemingly irrelevant data coverage. The choice of edge devices in IoT settings depends on the use case; for instance, in the telecom industry, it might be a cell tower, while in the automotive sector, an engine sensor serves as an edge device. These devices collect and transmit data to the data center or cloud for processing and analysis, enhancing real-time IoT analytics through edge computing. Local data processing at the edge minimizes latency and ensures that sensitive information at the network's edge is secure and authenticated.

5.1 Authenticating the Edge Devices

Securing edge devices is super important for making sure that the whole IoT system works smoothly. To make sure only authorized people or devices can access them, the edge should use more than just passwords – things like fingerprints or a mix of security measures can help. When the right credentials match up, it means the access is confirmed. For strong security in IoT edge devices, each device should have a special certificate identity for network interactions. Keeping edge devices safe from IoT security problems needs careful attention, hard work, and good planning. Since the edge is a key spot where data comes together, it's crucial to control it with strict security rules and structures. The Figure 5.1 shows the procedure of authenticating edge devices in IoT.

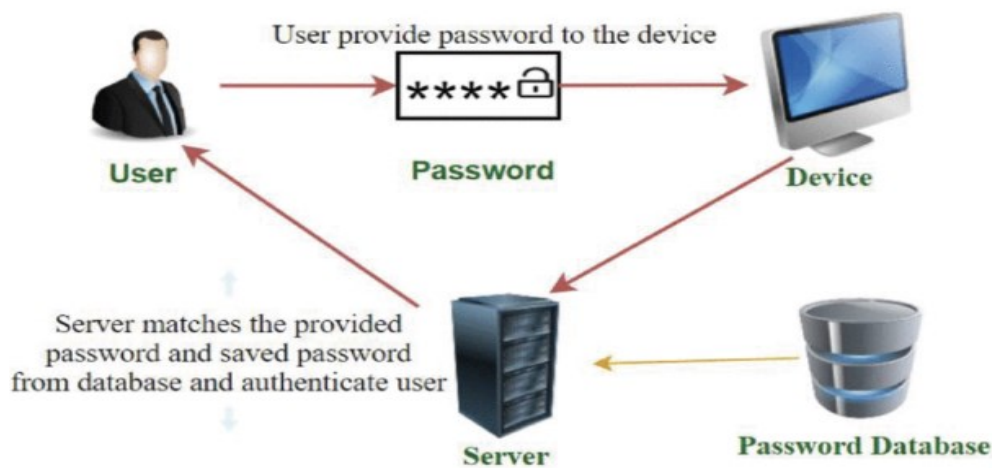


Fig. 5.1: Procedure for authenticating edge devices in IoT

Authenticating each device poses a significant challenge. Authentication is the process of confirming the identity of users and devices before granting access to a network or information system. Radio Frequency Identification (RFID) is a valuable tool for entity identification, using electromagnetic induction and wave propagation to distinguish objects. Various security mechanisms have been proposed, including cryptographic, password-based, biometric, token-based, and multi-factor authentication.

5.1.1 Password-based authentication

It is commonly used for device or user verification. Users create a unique ID and password, stored in a database, and only with a matching ID and password can access be granted as shown in Figure 5.1.

5.1.2 Token-based authentication

It involves generating a token to identify a user or device. In soft token-based authentication, a one-time password (OTP) is sent to the registered device or user and stored in the database for matching. In hard token-based authentication, a physical card or device with verification information is created. It is illustrated in Figure 5.2.

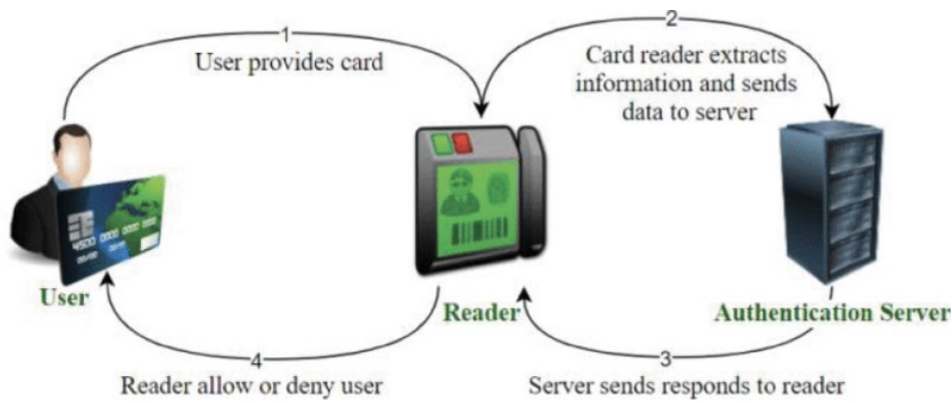


Fig. 5.2: Token-based authentication

5.1.3 Biometric authentication

It relies on human biological features. A scanner collects unique biological data from a user, comparing it with previously collected data in the database. This method can use fingerprint, face, iris, retina, hand, voice authentication, or a combination of these to maintain data uniqueness. It is shown in Figure 5.3.

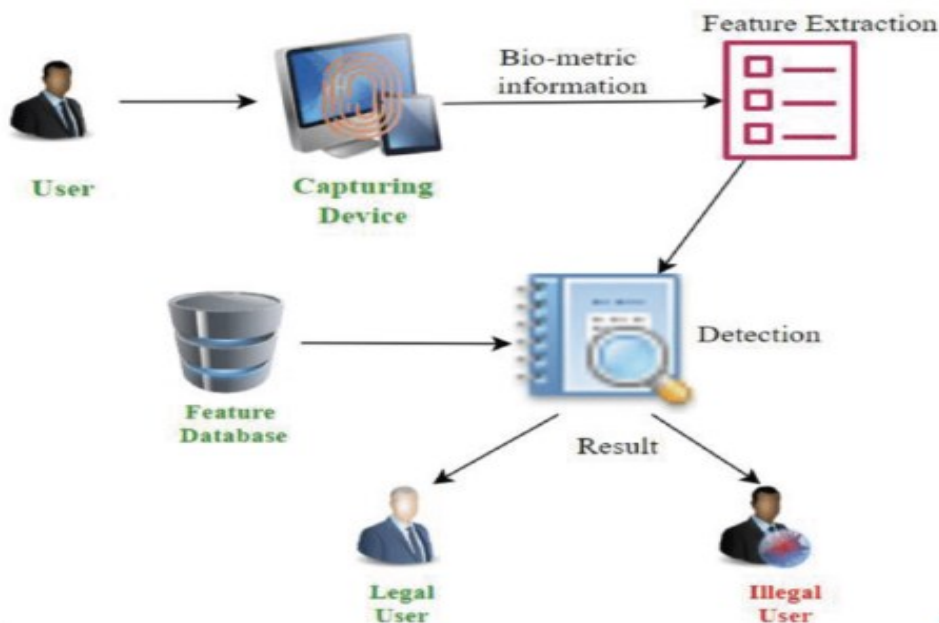


Fig. 5.3: Biometric authentication

5.1.4 Multi-factor authentication

It combines two or more ways to identify the user or device identity.

5.2 Security Check List for Edge Devices in IoT

The following is the security checklist for edge devices:

- Identify what devices are there on the network
- Create separate security profiles for all devices
- Have a device policy framed and enforced
- Change default passwords and have new passwords in place
- Have IT and Operational Technology (OT) work in cooperation
- Identify and know where secured data is stored for processing

It is recommended to detect devices on the network and perform continuous operation control of external devices. It is also required to have knowledge of which edge devices are on the network, what data they collect and share, and what risks they face.

5.3 Security Challenges in Authentication

- IoT security risks arise from edge devices trusting connected sources without verification.

- Blind trust among devices sharing data in IoT poses significant security concerns.
- Edge devices, making various devices smart and connected, increase the risk of threats in IoT.
- The IoT security industry is in its early stages, similar to the early days of the PC era with growing internet connectivity.
- Many IoT products currently lack sufficient security measures as manufacturers prioritize speedy market delivery over comprehensive security.

5.4 Authorization of the Edge Devices

Delegating control and authority is critical to achieving the fundamental principle of security. Edge devices can access data and resources within their allowed limits if it is architecturally possible. This means that the devices would have some kind of permission by configuration and by architecture used. Authorization here would mean role-based access control (RBAC) and signing certificate privileges to top security operations.

Access control allows limited access to specific groups, people, or access levels. This ensures easy and safe access to the site for users. Smart locks, keypads, card readers, and other related devices are commonly used to control access. Any malicious activity in the access system area will trigger an alarm and create a detailed notice of the driver of administration. Devices are configured for different operating conditions, sensitivities, specifications, and permissions in the control software. The control software is used as the entire controller framework. Dual control can be configured via a mobile app for mobile devices notifications can be received in real time. Figure 5.4 shows a mobile app connected to a home alarm system.

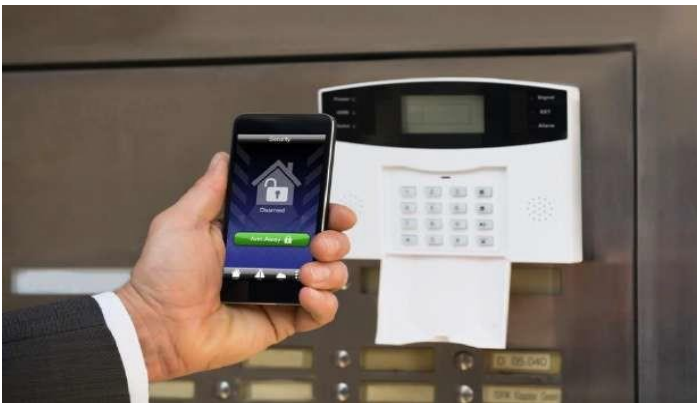


Fig. 5.4: A mobile app connected to a home alarm system

Each authorized mobile device must use its own unique IP address for acquisition access to the main controller to operate the security system. Also, the main server has a unique IP address to establish communication among access control system components.

5.5 Installation of Access Control System

Managing an access control system can be a complex task. To install the system the technician must know the following tasks:

- Addition of a new base station to the control system
- Keep the system secure
- Identification of users and devices
- Solve installation problems
- Ensures servers are up to date
- Make sure proper firewalls and the latest security patches are installed

5.6 Access Control System Architecture for IoT

Access control for IoT can be implemented in two ways as follows:

5.6.1 Distributed Architecture

In a distributed architecture, the control server offers access tokens to the users, so that they can have direct access to the IoT devices. These tokens serve as digital credentials, allowing users direct access to IoT devices. The tokens contain user identity and permissions, ensuring secure interaction while managing access throughout their lifecycle. Figure 5.5 shows the distributed architecture.

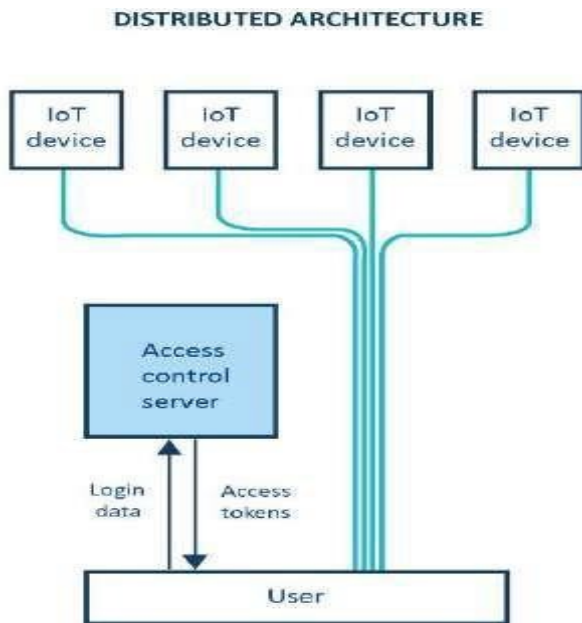


Fig. 5.5: Distributed architecture

5.6.2 Centralized Architecture

In a centralized architecture, users access cloud-based servers to interact with IoT devices. These servers authorize user requests and relay data between users and devices. They also handle data processing, scalability, security, and management, offering a centralized solution for IoT communication. Figure 5.6 shows the centralized architecture.

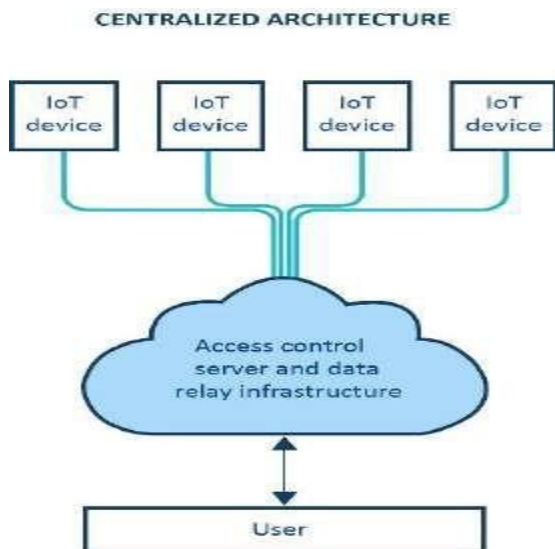


Fig. 5.6: Centralized architecture

5.7 Control Access Using Security

The access control system provides security to the organisation by offering access only to the authorised people. Figure 5.7(a) and (b) shows two examples of granting and denying access to a person.



Fig. 5.7 (a): Access granted

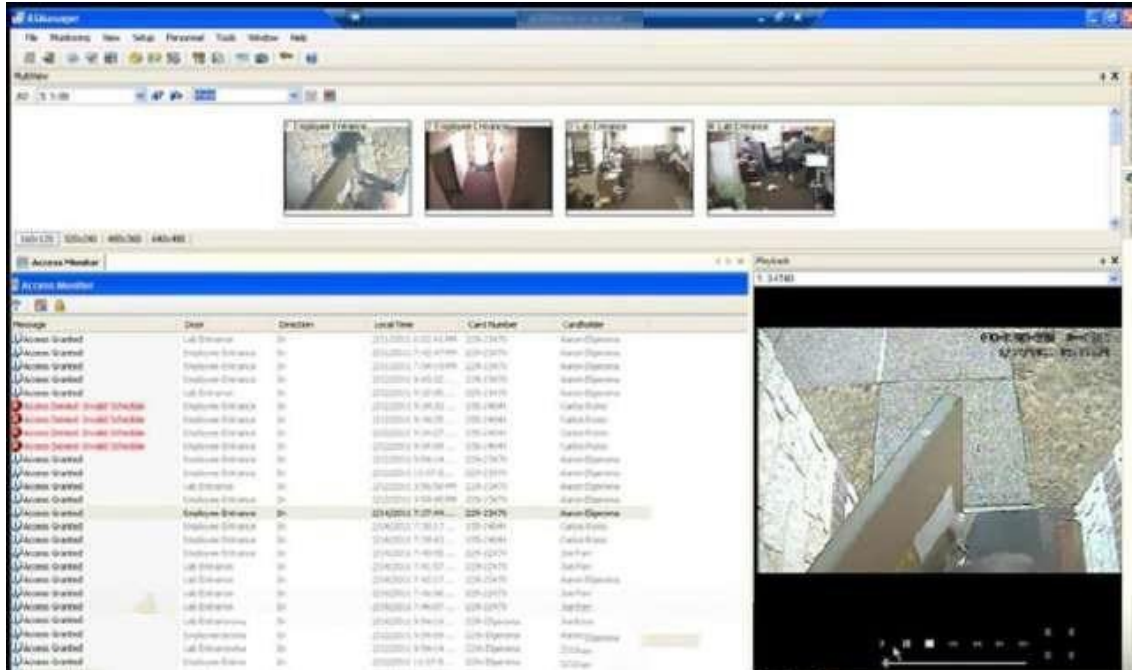


Fig. 5.7 (b): Access denied

It is of high priority to ensure the following precautions:

- Eliminate risk with security tools
- Track everything with the software interface
- Set the remote access rules restriction options properly
- Use advanced authentication method for restricting the access to unauthorised users
- Set email connection notifications

To set up the software interface, the following steps should be followed:

Step 1. Install the software and open it. Enter the login details as shown in Figure 5.8.



Fig. 5.8: Software login window

Step 2. Edit the controller settings by entering the network details and product details. Figure 5.9 shows the edit controller window.

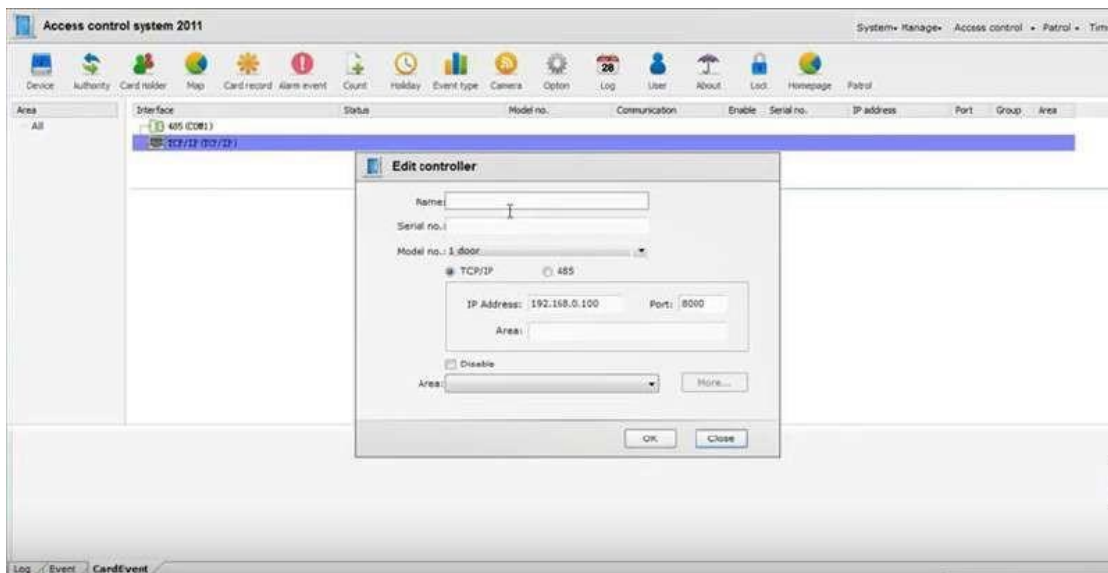


Fig. 5.9: Edit controller window

Step 3. Edit the device or machine settings. Figure 5.10 shows the device settings window.

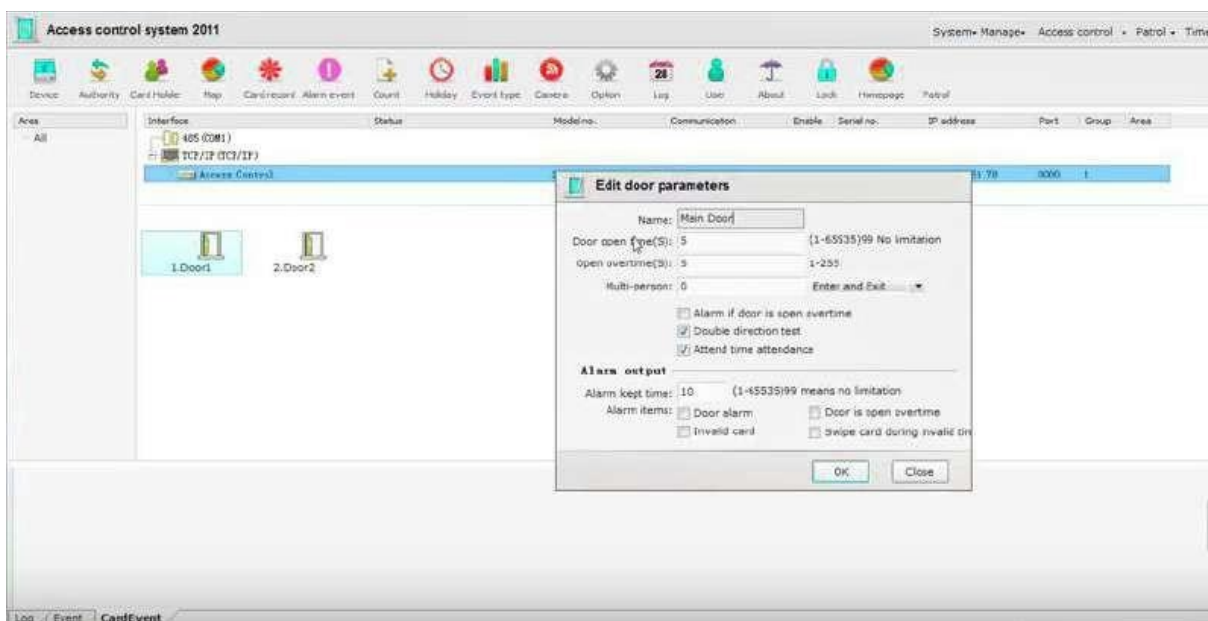


Fig. 5.10: Device settings window

Step 4. Create the time zone settings as per requirement. Figure 5.11 shows the time settings window.

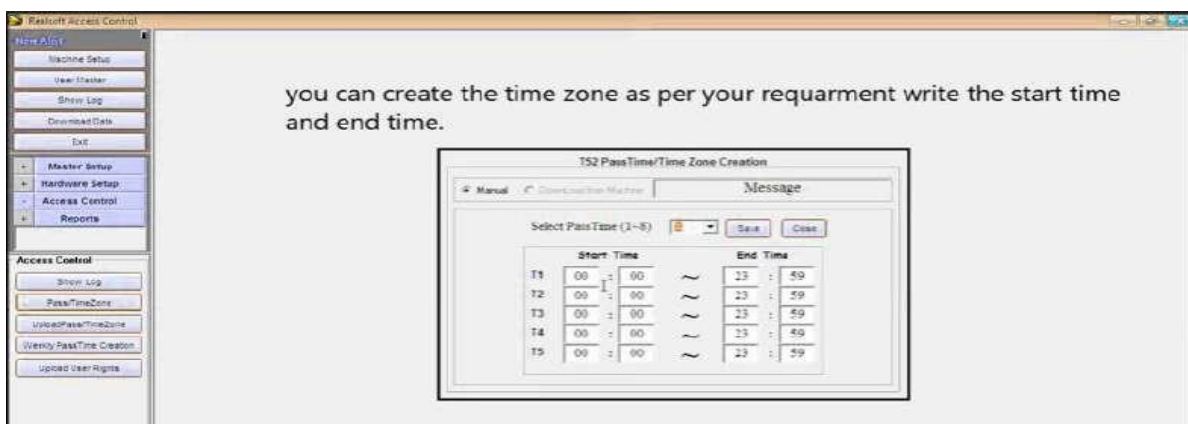


Fig. 5.11: Time settings window

Step 5. Fill the employee records. Figure 5.12 shows the employee records window.

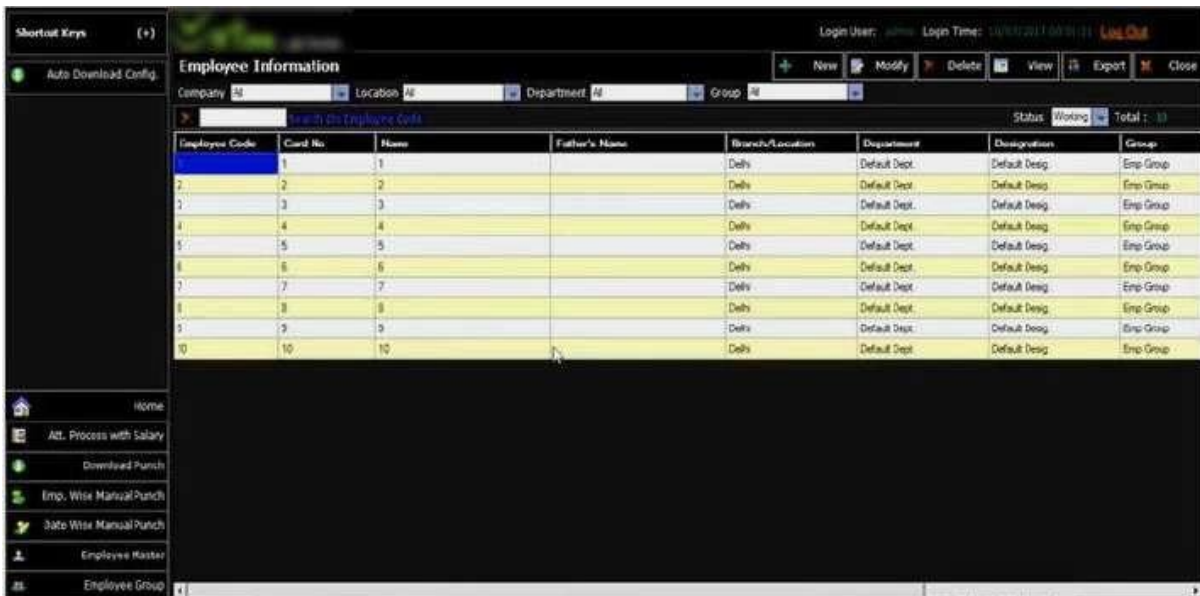


Fig. 5.12: Employee details window

Step 6. Click on a row to create or update the employee details. Figure 5.13 shows editing the employee records window.

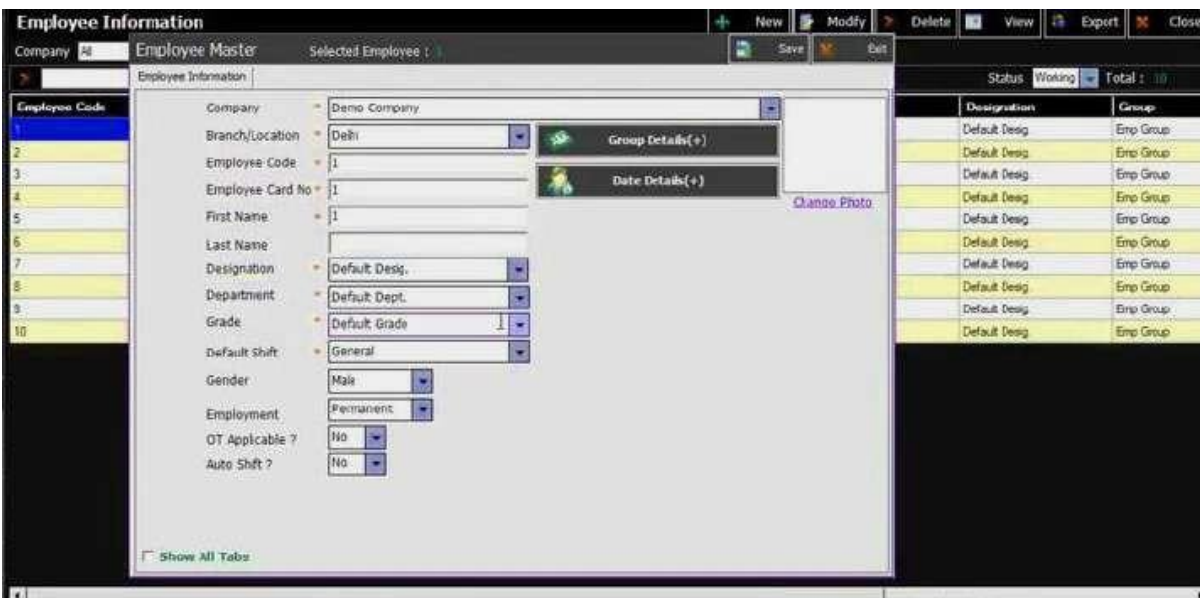


Fig. 5.13: Editing employee records window

Step 7. Connect the device and select the time zone and upload it to the machine as shown in Figure 5.14.

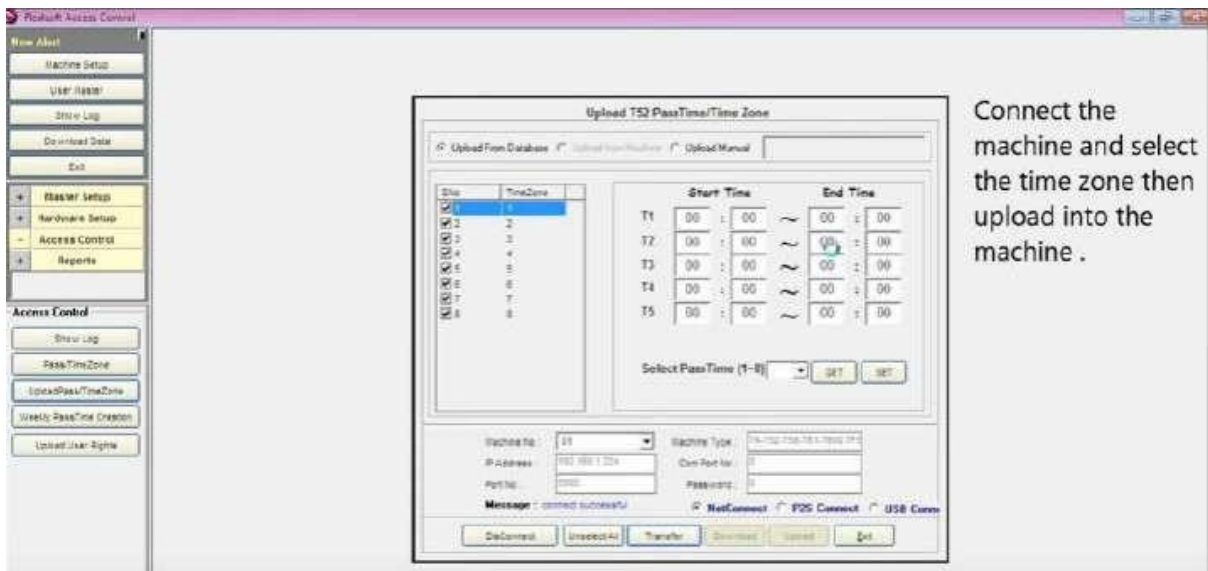


Fig. 5.14: Uploading time zone to the device

Step 8. Upload the user rights to the device as shown in Figure 5.15.

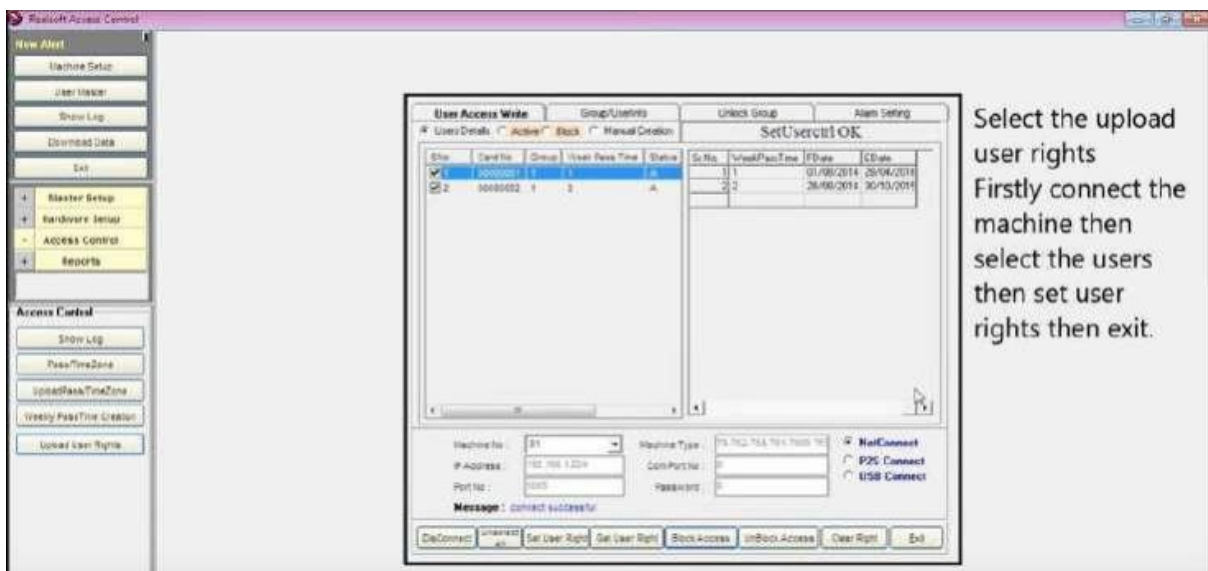


Fig. 5.15: Uploading user rights to the device

Step 9. Generate report as shown in Figure 5.16.

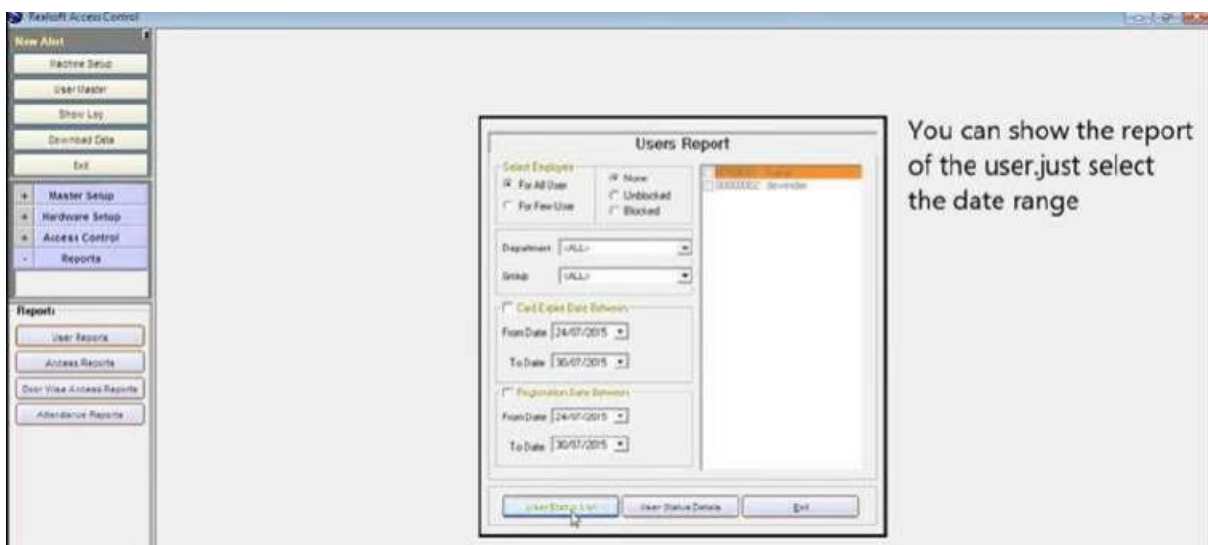


Fig. 5.16: Generating report

5.8 Securing Wireless Connection

Securing the network connection is very important. There are some ways to secure the wireless connection. The following default settings of the router must be changed:

- Router password must be changed
- IP address and the subnet mask should be updated
- Remote management should be disabled

Figure 5.17 shows changing the default settings.

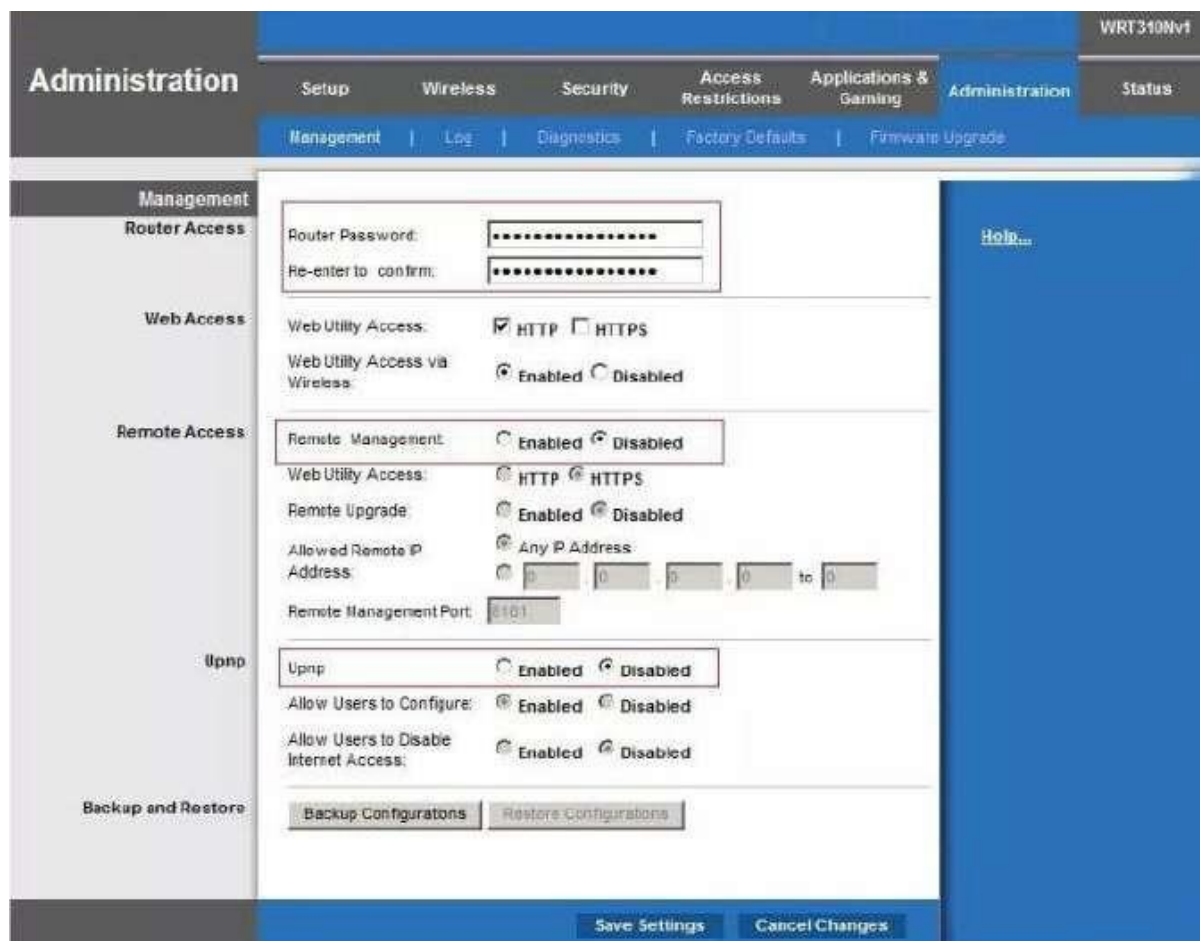


Fig. 5.17: Changing the router default settings

The default Service Set Identifier (SSID), which denotes the name of the network, must be changed and the broadcasting name option should be disabled. Figure 5.18 shows changing the wireless settings.

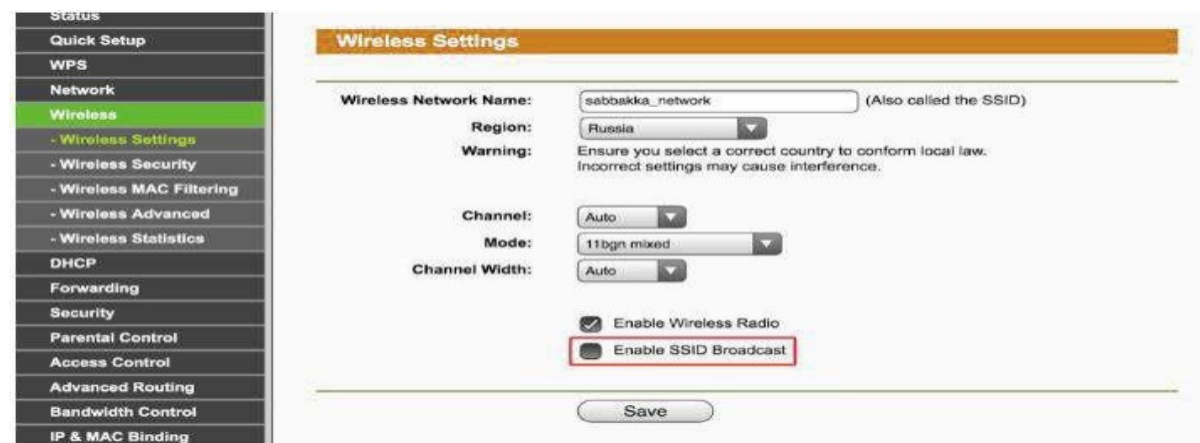


Fig. 5.18: Changing the wireless settings

Reliable encryption standard must be selected. Figure 5.19 shows the WAP2 selected as encryption method.

Wireless Security Settings

☐ Disable Security

☒ **WPA/WPA2 - Personal(Recommended)**

Version: WPA2-Personal

Encryption: AES

Password: 9N4MY_4Tqz_2048
(You can enter ASCII characters between 8 and 63 or Hexadecimal characters between 8 and 64.)

Group Key Update Period: 0 Seconds (Keep it default if you are not sure, minimum is 30, 0 means no update)

☐ WPA/WPA2 - Enterprise

Version: Automatic

Encryption: Automatic

Radius Server IP:

Radius Port: 1812 (1-65535, 0 stands for default port 1812)

Radius Password:

Group Key Update Period: 30 (in second, minimum is 30, 0 means no update)

☐ WEP

Type: Automatic

WEP Key Format: Hexadecimal

Key Selected	WEP Key (Password)	Key Type
Key 1: ☒		Disabled
Key 2: ☐		Disabled
Key 3: ☐		Disabled
Key 4: ☐		Disabled

We do not recommend using the WEP encryption if the device operates in 802.11n mode due to the fact that WEP is not supported by 802.11n specification.

Save

Fig. 5.19: WAP2 selected as encryption method

Router firewall must be enabled and the firmware must be updated. Figure 5.20 shows the upgrading of firmware.

Firmware Upgrade

File: файл не выбран

Firmware Version: 3.17.2 Build 140610 Rel.65031n

Hardware Version: MR3020 v1 00000000

Fig. 5.20: Upgrading of firmware

5.9 Check for Malware and DDoS Attacks

5.9.1 Malware Attacks

Many IoT devices become soft targets and often victims of malware because of poor security. Most of the IoT malware targets the devices that are not PC embedded and does not have any advanced security features. The following is a list of some symptoms that indicate that the device is malware affected:

- The system is slowing down or crashes.

- Annoying advertisements or unusual error messages are displayed.
- Pop-up messages and unusual messages show unexpectedly.
- There is an increase in Internet traffic.
- The browser homepage changes and the security solution is disabled.
- The control Panel cannot be accessed.
- Unfamiliar icons are visible on the desktop.

The malware can be removed by performing the following actions:

- Deleting temporary files
- Running a malware scan
- Using a good antivirus software

5.9.2 DDoS Attacks

DDoS is an attack in which a group of systems targets a single target, causing denial of service for the users of the victim. The most common symptom in a DDoS attack is the flooding of the incoming packets to the victim. The steps for detecting DDoS in Windows are as follows:

Step 1. Select start -> Select Run -> Type “cmd” -> Select OK.

Step 2. In the command prompt type netstat – ano - netstat.txt and then press Enter key.

The NETSTAT command displays the current TCP/IP network connections and protocol statistics in a system. The command is as follows:

netstat -ano

a – Shows the connections and listening ports.

n – Shows addresses and port numbers.

o – Shows the process ID related to each connection.

Step 3. Check the total number of connection at port 80 using netstat -ano | find /i /c“80” command.

Step 4. Check for the IP addresses with the maximum number of connections.

Step 5. Block the access of such IP

Summary

- The chapter explores authentication and authorization mechanisms in IoT, stressing the importance of securing edge devices and employing methods like password, token, biometric, and multi-factor authentication.
- It outlines security checklists, challenges, and measures for access control systems.
- The chapter provides the methods to secure wireless connections and detect malware and DDoS attacks.
- Overall, the chapter emphasized on the critical role of robust security practices in IoT environments.

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. What is the primary concern as IoT integrates into businesses? (a) Speed of data transmission (b) Security (c) Cost-effectiveness (d) Device compatibility
2. Which of the following is NOT considered an edge device? (a) Cell tower (b) Engine sensor (c) Cloud server (d) Smart thermostat
3. What is the purpose of authenticating edge devices in IoT? (a) To ensure efficient data transmission (b) To confirm the identity of users and devices (c) To regulate internet access (d) To prevent hardware malfunctions
4. Which authentication method relies on unique biological features? (a) Password-based authentication (b) Token-based authentication (c) Biometric authentication (d) Multi-factor authentication
5. What does RBAC stand for in the context of authorization? (a) Remote Biometric Access Control (b) Role-Based Access Control (c) Resource-Based Authorization Code (d) Restricted Biometric Access Center

6. Which architecture offers access tokens directly to users in IoT? (a) Distributed architecture (b) Centralized architecture (c) Decentralized architecture (d) Hybrid architecture
7. What is the primary purpose of securing wireless connections in IoT? (a) To increase data transmission speed (b) To reduce energy consumption (c) To prevent unauthorized access and attacks (d) To enhance device compatibility
8. What are common symptoms of malware-infected IoT devices? (a) Decreased internet speed (b) Unusual error messages (c) Reduced energy consumption (d) Improved device performance
9. What does DDoS stand for? (a) Distributed Denial of Service (b) Device Data Overload System (c) Digital Data Output Service (d) Directed Device Operation System
10. How can DDoS attacks be detected in Windows systems? (a) By running a malware scan (b) By deleting temporary files (c) By checking network connections using the NETSTAT command (d) By upgrading firmware and enabling firewalls

B. Fill in the blanks

1. Authenticating edge devices in IoT involves confirming the _____ of users and devices before granting access to a network or information system.
2. Multi-factor authentication combines two or more ways to identify the _____ or device identity.
3. Access control allows limited access to specific _____, people, or access levels to ensure easy and safe access to the site for users.
4. Securing the network connection is crucial in IoT, and changing the default settings of the router, such as updating the _____ and the subnet mask, is recommended.
5. Malware attacks on IoT devices can be identified through symptoms such as system _____ or crashes, unusual error messages, and increased internet traffic.
6. In token-based authentication, a one-time password (OTP) is sent to the registered device or user and stored in the database for _____.
7. Distributed architecture in access control for IoT involves the control server offering access tokens to users, allowing them to have direct access to _____.
8. _____ authentication relies on human biological features like fingerprint, face, iris, or voice authentication to confirm identity.
9. The steps for detecting DDoS attacks in Windows include using the NETSTAT command to display current TCP/IP network connections and protocol statistics, followed by checking for the total number of connections at port _____.
10. Access control systems provide security by offering access only to _____ individuals or devices.

C. State true or False for the following

1. In IoT settings, edge devices collect and transmit data to the data center or cloud for processing and analysis, enhancing real-time IoT analytics through edge computing.
2. Password-based authentication involves generating a token to identify a user or device.
3. Biometric authentication relies on human biological features such as fingerprint, face, iris, or voice authentication.
4. The centralized architecture in access control for IoT involves users having direct access to the IoT devices.
5. DDoS attacks involve flooding the incoming packets to a single target, causing denial of service for the users of the victim.
6. Multi-factor authentication combines two or more ways to identify the user or device identity.
7. Edge devices, such as engine sensors in the automotive sector, serve as edge devices that collect and transmit data in IoT settings.
8. In token-based authentication, a physical card or device with verification information is created for soft token-based authentication.
9. Access control allows unlimited access to specific groups, people, or access levels.

10. Malware attacks on IoT devices often target devices with advanced security features.

D. Short Answer Type Questions

1. How does RFID contribute to entity identification in IoT?
2. Describe the procedure for authenticating edge devices in IoT.
3. What is the importance of role-based access control (RBAC) in authorization mechanisms?
4. Explain the difference between distributed and centralized architecture in access control systems for IoT.
5. How does multi-factor authentication enhance security in IoT systems?
6. What are some security challenges faced in authenticating edge devices?
7. Describe the process of token-based authentication in IoT.
8. What are the key components of the security checklist for edge devices in IoT?
9. How can access control systems contribute to securing wireless connections in IoT environments?
10. What steps can be taken to detect and mitigate malware attacks in IoT devices?

UNIT 2. LEVEL 1 TROUBLESHOOTING OF IOT DEVICES

Introduction

This unit provides individuals with foundational skills for diagnosing and resolving issues within Internet of Things (IoT) networks and devices. It emphasizes practical troubleshooting techniques essential for maintaining optimal IoT system performance. It includes testing connectivity between devices, evaluating storage and power supply, ensuring smooth communication, and verifying data transfer. This makes students able to learn the expertise needed to address a wide range of IoT-related challenges.

The unit explores the critical aspect of testing connectivity within IoT systems, using a medical healthcare tracking system as a prime example. It outlines the levels of IoT architecture, ranging from device and network levels to application services and user interfaces. The chapter highlights the challenges faced by testers due to the intricate interplay between hardware and software components, the need for real-time communication, and the complexities of data acquisition. It also discusses various testing methodologies and tools to address these challenges effectively, along with specific procedures for testing sensors, actuators, and connectivity modules.

Furthermore, the unit underscores the importance of verifying network connectivity and delves into the technical requirements for achieving optimal IoT connectivity. It concludes with insights into end-to-end testing methodologies aimed at ensuring the reliability and functionality of interconnected devices within an IoT network. By providing a comprehensive overview of IoT testing, this chapter equips readers with the knowledge and tools necessary to navigate the complexities of ensuring seamless connectivity and performance in IoT systems.

Next, the unit discusses the testing of storage, power supply, and internet connectivity for IoT boards, focusing on Arduino and Raspberry Pi. It covers methods for testing on-board memory storage, including Arduino's non-volatile memory and SD card options, and commands for assessing Raspberry Pi's disk usage. The chapter also provides steps for testing the on-board power supply, with practical activities for hands-on verification of storage space and power supply performance.

The unit covers the fundamental concept of the data transfer between devices, covering communication testing, performance metrics assessment, connectivity checks, and device security. It explores steps for verifying communication between nodes and gateways, evaluating metrics like MTU and latency, and setting up connectivity credentials. Additionally, it discusses device authentication methods such as user ID/password and OTP authentication.

Chapter 6. Testing devices and checking Connectivity

Let's look at an example of a medical healthcare tracking system where the device records information about a patient's health, heart rate, and fluid intake and reports the results to the doctors. The information is stored in the system and is viewable at any time as needed. Based on the information, doctors can start patients on medication and fluid supplements. Any computer or mobile device that is connected to the medical device has the ability to remotely trigger this. This can be understood from Figure 6.1.

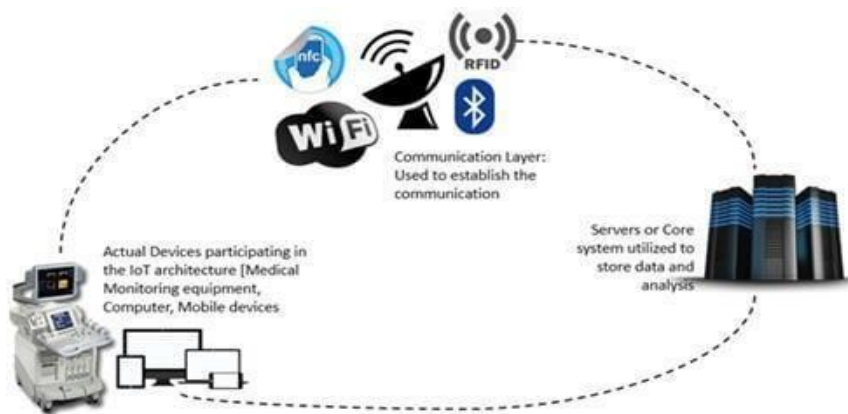


Fig. 6.1: IoT system

In the developing world around us, IoT is a growing market and has lots of opportunities. IoT-enabled gadgets, smart device applications, and communication modules play a vital role in studying and evaluating the performance and behaviour of various IoT services. Poor design of IoT-enabled devices and services can hamper the correct functioning of the application and in turn, negatively affect the end-user experience.

6.1 Levels of IoT

The architectural elements of the Internet of Things vary based on applications of use. Depending upon this fact, different levels are defined for the IoT system from Level 1 to 5, which also helps to explain the different characteristics of IoT. It is described as follows:

Level 1 – It is IoT Device Level that includes all IoT sensors and actuators.

Level 2 – It is the network Level of IoT that includes all IoT network devices including IoT gateways, routers, and switches.

Level 3 – It is IoT Application Services Platform Level which includes the key management software functions to enable the overall management of IoT devices and network. It also includes main functions connecting the device and network levels with the application layer.

Level 4 – This is IoT Application Level includes all applications operating in the IoT network

An example of an air conditioner can be referred to understand the IoT levels. This example covers the working of IoT systems up to level 4 as shown in Figure 6.2.

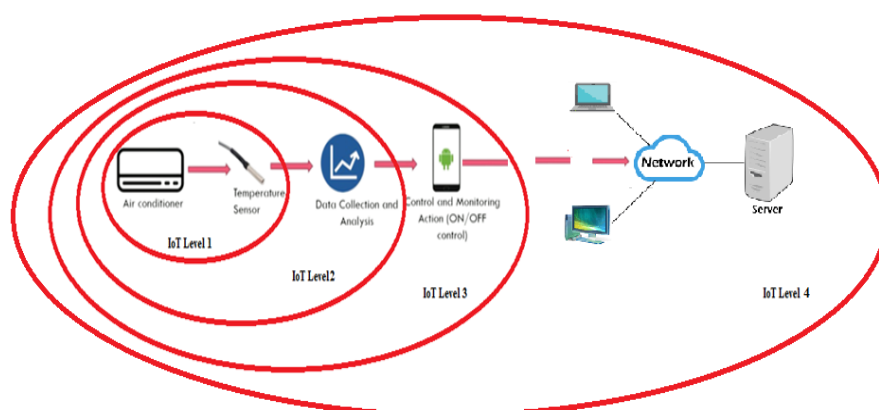


Fig. 6.2: Example of IoT levels in air conditioner

Level 1 of IoT

At its simplest, Level 1 of IoT involves a basic setup comprising a sensor or device, such as a temperature or pressure sensor, responsible for sensing data. This data is then stored locally and subjected to analysis. Monitoring and control functions are accessible via a mobile app. Level 1 is suited for straightforward applications with minimal or no complexity, characterized by relatively small datasets that don't qualify as big data. All control operations are facilitated through internet connectivity. Physical devices form the backbone of connections at this level, making it a foundational aspect of IoT characteristics.

6.2 IoT Testing Challenges

Technicians face many challenges in the testing of IoT systems.

Hardware-Software Mesh

IoT is a system of interconnected hardware and software components. The hardware components of the system, such as sensors, communication gateways, and other components, are just as important as the software programs. Functionality testing alone cannot fully certify a system to be compliant with standards. In terms of the environment, data transfer, and other factors, hardware and software are always dependent on one another. Therefore, compared to testing a generic system like a software or hardware component, it becomes a laborious task.

Device Interaction Module

Since this architecture connects various sets of hardware and software, real-time or nearly real-time communication between them is required. The testing team faces difficulties with security, backward compatibility, and upgrade issues when they both integrate with one another.

Real-time Data Testing

For a system like this, it has already been established that regulatory testing and pilot testing are required; however, obtaining the necessary data to conduct an IoT device pilot test is extremely difficult. It is very challenging for the technician to be on the testing team, to get through regulatory checkpoints, or to get the system deployed in the pilot. If, as in the example, the system is connected to healthcare, the step becomes even more difficult. For the testing team, this continues to be a significant challenge.

User Interface (UI)

Devices running on all platforms—Android, Windows, Linux, and iOS—are covered by IoT. It is practically impossible to test it on every device, but it is possible to test it on some of them. It is impossible to ignore the chance that a simulated or non-owned device will be used to access the user interface. It's a difficult challenge to overcome.

Network Availability

As the IoT is all about data being communicated at constantly faster speeds, a network connection is essential. It is necessary to test the IoT architecture across a range of network speeds and connectivity. Virtual network simulators are primarily used to change the network load, connectivity, stability, and other parameters in order to test this.

6.3 Testing of IoT Systems

IoT systems are prone to a range of errors due to the scattered data received from sensors, inventory, machines, equipment and tools, edge devices, people, and production logs. Because of their distinct features and intricate architecture, IoT systems require a variety of tests to be conducted on every part of the system. A few tests are suggested to make sure that IoT applications are up to the mark in terms of scalability, performance, and security. These tests are as follows:

1. Usability testing
2. Compatibility testing
3. Regulatory testing
4. Security and privacy testing
5. Upgrade testing
6. Performance and real-time testing
7. End-user application testing
8. Connectivity testing

9. Functional Testing

6.3.1 Usability Testing

Ensuring the usability of each device is necessary, with a focus on portability for seamless integration into various segments and locations. The equipment should possess intelligence to not only deliver notifications but also relay error messages and warnings efficiently. Additionally, the system must offer logging capabilities to provide clarity to end-users, with provisions for storing events in a database if necessary. Proper handling of notifications and display functionalities on devices, whether computers or mobile devices, is very important. Thorough testing should encompass usability aspects such as data display, processing, and the execution of job tasks from the devices. The notifications should be shown and handling of the display should be done properly in the devices [computers/mobile devices].

6.3.2 Compatibility Testing

Compatibility testing is essential for IoT systems due to their intricate design. IoT compatibility testing requires evaluating things like different operating system versions, and browser types along with their corresponding versions, device generations, and connection protocols (such as Bluetooth 2.0 and 3.0).

6.3.3 Regulatory Testing

Because the Internet of Things is made up of several protocols, it must go through several regulatory and compliance checkpoints. It is crucial to make sure that a situation like this doesn't happen, when the product passes every testing phase but fails the final compliance checklist (regulatory body testing). It is preferable to get experience complying with regulations at the outset of the development cycle. The testing checklist should include the same item. This guarantees the product's certification for the regulatory checklist as well.

6.3.4 Security and privacy Testing

The primary security issue with IoT is its data-centric design, where all the devices/systems connected operate based on the data that is available. There is always a possibility that data being transmitted between devices might be viewed or read while it is being transferred. Testing-wise, it should be determined whether data is encrypted or secured while it is moved between devices. Make sure that any user interface (UI) that exists has password security.

6.3.5 Upgrade Testing

The IoT is a combination of a variety of hardware, firmware, operating systems, devices, networking layers, and protocols. When an upgrade is carried out, whether it's for the system or any of the components, careful regression testing should be done and a strategy should be adopted in order to resolve upgrade-related problems.

6.3.6 Performance and real-time Testing

The testers must make sure the IoT system is scalable based on the system's performance. Testing should be done by technicians on 2-10 testing platforms at a time and it should be made sure that the information spreads to 10-20 devices. The technicians' role as testers is to ensure that the system functions the same even though the additional data spreads. The monitoring tool that shows system power and usage should also be tested by the technician's temperature, usage, and so forth.

6.3.7 End-user application Testing

This testing is essential when it comes to the Internet of Things. Only laboratory testing ensures that the system or product functions as intended. Still, this could have a disastrous backlash if put in real-time situations with real-time steps. The system is tested in the real world with a small number of users during the pilot phase. They make use of the application and provide system feedback.

6.3.8 Connectivity Testing

Since the solution is based on the Internet, connectivity is essential. The system must always be accessible and have flawless connectivity with the parties involved. When the connection is up and operating, maintaining connectivity, sending data, and receiving job tasks from the devices should be effortless.

The scenario of a connection failure is the other condition that has to be tested. That doesn't matter how reliable the network and system are, there's a chance that the system will not function. The technician should test the offline conditions since they are testers as well. There must be a notification when the system is unavailable over the network. which can instruct the user or the technician to fix any problem. Conversely, the system must have a

feature that allows it to store all of the data during the time spent offline. Once the system comes online, all that data should get propagated. Data loss should not be there in any condition.

6.3.9 Functional Testing

Every IoT device serves a distinct function within the IoT system, necessitating thorough testing to ensure efficient operation. To achieve this, various testing methodologies are employed:

- **Unit Testing** – Developers create and test code for individual modules or devices within the IoT system to verify their functionality.
- **Integration Testing** – Modules are integrated to assess how they function collectively. This is particularly important in systems involving multiple devices to ensure seamless communication between them.
- **End-to-End Testing** – The entire IoT system is tested from start to finish to simulate real-world user interactions and ensure comprehensive functionality.
- **Smoke Testing** – This involves checking the stability of the IoT system and devices to ensure they are capable of undergoing further testing.
- **Regression Testing** – After implementing new modules or updates, regression testing is performed to ensure that existing components continue to function as expected, preventing any unintended consequences.
- **Interface Testing** – The graphical user interface of the IoT system is verified to ensure it meets user requirements and specifications, facilitating user interaction and control.

Functional testing is paramount for IoT systems, ensuring that core functions operate reliably to enable proper system functionality.

The complete IoT testing framework is shown in Figure 6.3.

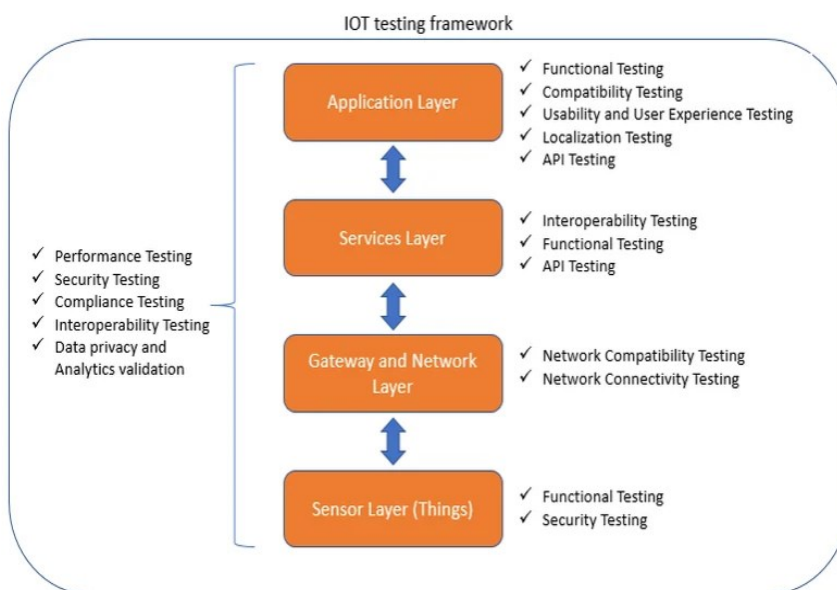


Fig. 6.3 IoT Testing Framework

6.4 Setting Up Components for IoT Testing

Now, the system is ready to perform the actual IoT testing using the device. For this, there is a need to set up the components as it would be done by the user when they receive it. It can be done using following steps:

6.4.1 Understand the device

The testers in charge should have full knowledge of the device and how it works. The team can bring new testers or real users to understand the device's ease of use and basic information.

6.4.2 Prepare the device

The device should be new (not the one used for testing) and installed at the same place for which it is intended. For instance, if it is a smartwatch, it should be worn first. If it is a waterproof CCTV camera, it should be installed in the open with an actual DVR to understand the real use case working.

6.4.3 Set up the server

Next, we need a working server with no issues. This will keep us focused on just the device faults. The server should also be tested for connections and compatibility once before to ensure the connecting device works with the server.

6.4.5 Set up the application

If the device comes with a mobile application (or any other platform application), it should have been tested before so that the focus can be directed toward the IoT device. The application should be available on the application hosting platforms and ready to be downloaded and used like a real end-user.

6.4.5 Enable connections

Finally, an established connection is ready to be connected to the device. For instance, if Wi-Fi is required, it should be enabled, and its working should be verified before establishing an IoT connection. If Bluetooth is required, devices should be brought into the range of IoT devices. If multiple connections are allowed, there should be numerous compatible devices. It is also advisable to bring the devices that should not be connected. This ensures compatibility testing.

6.5 IoT Testing

IoT testing is essential for ensuring the reliability and security of interconnected devices. It involves assessing aspects like interoperability, data integrity, and network connectivity. Through rigorous testing, organizations can mitigate risks and enhance user experience in IoT solutions. It can accomplish by following two methods:

6.5.1 Performing Manual IoT Testing

Manual testing of IoT devices excludes the use of automation tools or frameworks, relying solely on a team of testers equipped with manual testing tools. The initial phase involves setting up all necessary components manually, including frameworks, tools, servers, applications, and physical devices.

Firstly, the server is established and ensured to be operational. Then, it is configured for the IoT device, ensuring proper connectivity and accurate error logging. Multiple devices are required to install the application and verify its functionality, involving repetitive tasks across various devices. Subsequently, the IoT device is connected to the application for further testing.

However, this initial setup process is time-consuming and detracts testers' attention from other critical tasks, potentially causing delays in production cycles. Therefore, while manual testing can verify usability and basic functionality, relying solely on it can negatively impact organizations, particularly considering the high reliance on IoT systems and their susceptibility to security vulnerabilities.

Manual testing may cover security aspects, but it may not be as effective in thwarting sophisticated attacks compared to automated methods. Hence, to ensure the best product release, a combination of manual and automated IoT testing is recommended.

6.5.2 Performing Automatic IoT Testing

When it comes to testing IoT devices, what really matters is how well they work, not how many features they have. Even if an IoT device has just a few features, those features need to work perfectly all the time. Think about something like a tire pressure monitor – it has to give accurate readings, or it could be dangerous.

Using automated testing for IoT devices is helpful because it makes sure everything is right and the device is secure by running automatic checks. But automated testing for IoT isn't as advanced as it is for other things like websites. IoT devices have become super important in our lives, but they weren't always tested as thoroughly in the past. So, testing tools and methods for IoT devices are still catching up.

So, when it comes to testing IoT devices, it's important to use both manual and automated testing. Some things, like checking the sound of a speaker, need to be done manually. Other tasks, such as ensuring the device's safety and functionality, can be automated. For the most dependable testing, it's ideal to combine both manual and automated methods.

6.6 IoT Testing Tools

There are various tools that are used during the testing of IoT systems which are as follows:

6.6.1 Software Tools

A) Wireshark

Wireshark is a free and open-source network traffic analysis tool compatible with Windows, Mac, Unix, and Linux systems, captures and interprets data packets traversing various network interfaces like Ethernet, LAN, or SDRs. As a packet sniffer or network protocol analyzer, it provides IT professionals and cybersecurity teams with valuable insights into network activities.

Formerly known as Ethereal, Wireshark employs display filters to translate captured packets into readable formats, facilitating the identification of network security issues and potential cyber threats. Operating in 'promiscuous mode,' it allows users to analyze traffic regardless of its destination, like a fly on a wall watching office activity as a silent observer.

However, the accessibility of Wireshark also poses security risks, as cybercriminals can exploit it for malicious purposes. To mitigate such threats, security measures should be implemented.

Wireshark finds extensive use in diverse sectors, including governments, educational institutions, and technology firms. Typical applications range from diagnosing slow internet connections, troubleshooting latency issues, and detecting malicious network activities to tracing VoIP calls and intercepting Man-in-the-Middle attacks.

By rendering and translating traffic in real-time, Wireshark simplifies packet analysis, enabling swift resolution of detected issues and prevention of service disruptions or data breaches.

B) tcpdump

Unlike Wireshark, tcpdump operates without a graphical user interface (GUI). It is a command-line utility designed to display TCP/IP and other packets transmitted or received across a network.

Launched from the command line, tcpdump serves as a packet analyzer, intercepting and presenting packets created or received by the host computer. Compatible with Linux and most UNIX-type operating systems, it provides insight into network traffic.

Despite its name, tcpdump can capture non-TCP traffic, including UDP and ICMP. Its widespread availability establishes it as the default standard for capturing network traffic, bundled with many BSD, Linux, and macOS distributions, with a version also installable on Windows systems. Its extensive history ensures ample online and text-based resources for learning.

A basic application of tcpdump is examining all firewall-seen traffic on a network interface, with the option to adjust detail levels if defaults are insufficient. The command syntax for this task is **tcpdump -i<interface name>**. For instance, to monitor traffic on the internal (physical) interface named eth-s1p3 on a Nokia device, the command **tcpdump -i eth-s1p3** would suffice.

C) Shodan

Shodan serves as an IoT search engine, scanning internet-connected systems such as servers, routers, printers and indexing their exposed services like web servers, SSH, RTSP.

With Shodan, it's possible to discover vulnerable IoT devices by gathering their digital banners. These banners hold important device details like operating system, IP address, open ports, serial number, hardware specifications, geographic location, ISP, and owner information. Hackers may exploit this data to uncover vulnerabilities, like outdated software, or find devices with default login credentials.

Moreover, Shodan enables targeted searches for devices based on specific criteria, like open ports. For instance, querying "port:22" would yield devices with port 22 open. It facilitates the discovery of various connected devices like webcams, traffic signals, and more.

Additionally, Shodan provides statistics on devices, including their location, open ports, and operating systems. This wealth of information aids in creating maps pinpointing the precise locations of target devices.

While these technical details may seem insignificant to ordinary users, they hold significant value for hackers, enabling potential exploitation and harm

6.6.2 Hardware Tools

A) Joint Test Action Group (JTAG)

JTAG (Joint Test Action Group) is a well-established IEEE 1149.1 standard that was developed in the year 1980 to solve the manufacturing issues that occurred within electronic boards or printed circuit boards. This IEEE standard simply defines how the circuitry of a computer is tested to confirm whether it is working correctly after the procedure of manufacturing. On the circuit boards, tests are performed to check the solder joints. It is shown in Figure 6.4.



Fig. 6.4: JTAG

Joint Test Action Group provides a pins-out view for testers with every IC pad which helps in identifying any faults within a circuit board. Once this protocol is interfaced to a chip, this can attach a probe to the chip by allowing a developer to control the chip as well as its connections with other chips. The interface with the Joint Test Action group can also be used by Developers for copying the firmware to non-volatile memory in an electronic device. This is similar to a debugger in PC applications. This helps in debugging the target platform code and shows the variables step by step.

B) Digital Storage Oscilloscope

Digital Storage Oscilloscopes (DSOs) are indispensable tools in networking for analyzing and troubleshooting electrical signals. They enable engineers to visualize signal waveforms, assess timing characteristics, and ensure the integrity of data transmission across network devices. By capturing and storing digital data,

DSOs allow for in-depth analysis of network communication protocols, helping diagnose issues such as signal distortions, noise interference, and latency problems. In essence, DSOs serve as vital instruments for maintaining and optimizing network performance in various industries, including telecommunications, IT, and electronics.

This is used to check various events with time stamps, glitches in power supply and signal integrity checks. It is shown in Figure 6.5.



Fig. 6.5: Digital Storage Oscilloscope

C) Multi meter

A multi meter is a test tool, mostly used in electronics. A digital multi meter is a test tool used to measure two or more electrical values—principally voltage (volts), current (amps) and resistance (ohms). It is a standard diagnostic tool for technicians in the electrical and electronic industries as shown in Figure 6.6.

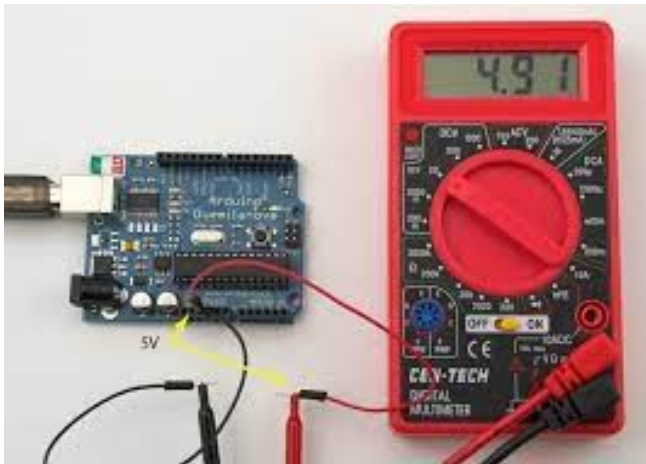


Fig. 6.6: Multi meter connected with Arduino board

6.7 Testing of IoT devices

6.7.1 Testing of Sensors

A) Proximity Sensor

Proximity sensors come in various types, such as inductive, capacitive, photoelectric, safety gratings, and hall sensors. Despite their differences, they all work similarly. They detect objects within their range and trigger a circuit to turn on or off. It is shown in Figure 6.7.



Fig. 6.7: Proximity sensor

These sensors find wide application across different fields. They're used for tasks like controlling liquid levels, counting products, distinguishing widths, detecting speeds, recognizing holes, sensing automatic doors, identifying color marks, operating manipulators, and providing safety in potentially hazardous environments. Furthermore, photoelectric sensors, which utilize infrared rays, can also serve as anti-theft alarms in banks, warehouses, stores, and other places where security is essential.

Practical Activity 6.1. Demonstrate to test proximity sensor using a multi meter.

Material Required

Proximity sensor, multimeter, 24VDC supply

Procedure

Step 1. Collect all the required components as shown in Figure 6.7.



Fig.6.7: Proximity sensor and multimeter

Step 2. Set the multi meter to the DC voltage mode.

Step 3. Use a switching power supply to provide power to the proximity switch.

Step 4. Connect the brown wire from the proximity switch to the positive terminal of the DC power supply.

Step 5. Connect the blue wire from the proximity switch to the negative terminal of the DC power supply.

Step 6. Bring the object to be measured close to the proximity switch.

Step 7. Check the output of the black wire from the proximity switch.

Step 8. If the output is positive voltage, the sensor is a PNP type. If it's negative voltage, it's an NPN type. It is shown in Figure 6.8.

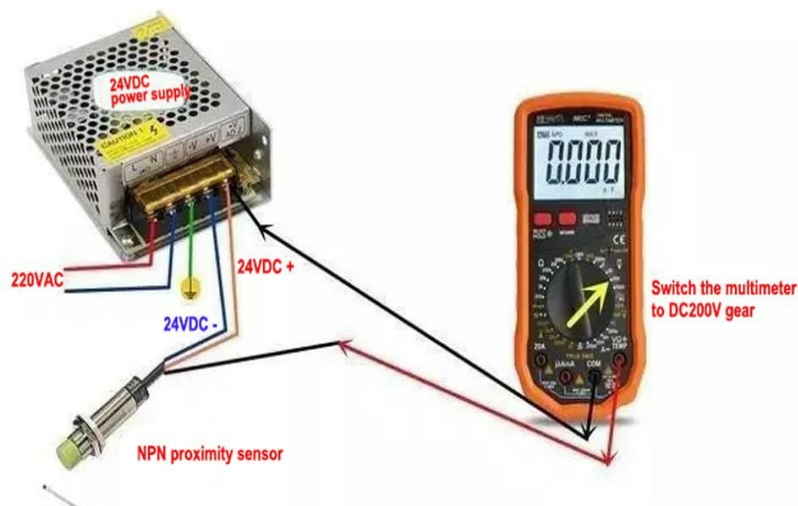


Fig. 6.8: Test arrangement for Proximity sensor

Step 9. If there's no output on the black wire, it indicates that the proximity switch is damaged.

B) Temperature Sensor

A temperature sensor is a device that measures the temperature of its surroundings and converts it into an electrical signal. It's used in various industries for monitoring, control, and safety purposes. Temperature sensors come in different types, such as RTDs, thermocouples, thermistors, and semiconductor-based sensors, offering versatility and accuracy in temperature measurement.

Testing of temperature sensors is quite simple. Here the example of TMP36 sensor is taken to understand the procedure.

Practical Activity 6.2. Demonstrate to test temperature sensor TMP36 using a multi meter.

Material Required

TMP36 sensor, multi meter, connecting wires, AA batteries

Procedure

Step 1. Connect a power supply that provides 2.7-5.5V (like 2-4 AA batteries) to the pins of TMP36 sensor as shown in Figure 6.9.

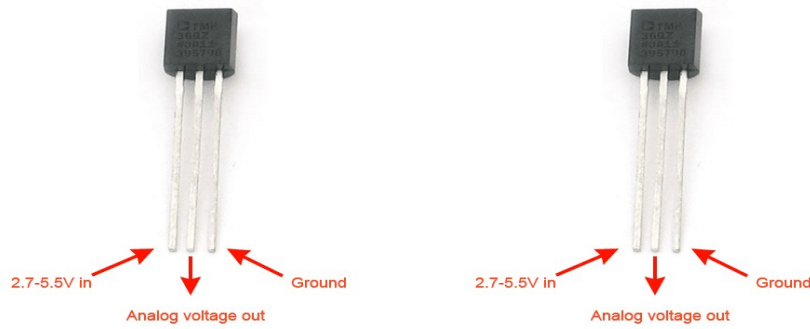


Fig. 6.9: TMP36 sensor pin connections

Step 2. Make sure the ground is connected to pin 3 (the right pin), and power is connected to pin 1 (the left pin).

Step 3. Now, connect your multi meter in DC voltage mode to ground and the remaining pin 2 (the middle pin).

Step 4. As the TMP36 sensor is used here and the temperature is around room temperature (25°C), the reading on the multi meter should be about 0.75 V. However, if an LM35 sensor is used, the voltage will be around 0.25V. It is shown in Figure 6.10.



Fig. 6.10: TMP36 sensor testing at room temperature

The sensor is indicating that the temperature is 26.3°C also known as 79.3°F .

Step 5. Now, the voltage range can be changed by pressing the plastic case of the sensor with fingers, a rise in temperature/voltage can be seen as shown in Figure 6.11.



Fig. 6.11: TMP36 sensor testing at high temperature

By putting fingers on the sensor, heating it up a little, the temperature reading is now 29.7°C / 85.5°F .

Step 6. Next, by touching the sensor with an ice cube (preferably in a plastic bag so it doesn't get water on your circuit) a drop in the temperature/voltage can be seen as depicted in Figure 6.12.



Fig. 6.12: TMP36 sensor testing at low temperature

As an ice-cube is pressed against the sensor, temperature drops down to 18.6°C / 65.5°F.

6.7.1 Testing of Actuators

Servo motor

The servo motor is a very popular and useful tool. It works as an actuator in the circuits. Testing servo motors for IoT involves ensuring compatibility with IoT platforms, responsiveness to remote commands, and accuracy in executing movements based on real-time sensor data. It is shown in Figure 6.13.



Fig. 6.13: Servo motor

For testing of servo motor, perform the following steps:

Practical Activity 6.3. Demonstrate to test a servo motor connected with an Arduino board using a push button switch.

Materials Required

Servo motor, Arduino board, push button switch, jumper wires

Procedure

In this activity, a servo motor is connected with an Arduino board using a push button switch. The servo motor should act accordingly with the application of the switch.

Step 1. Connect the black wire from pin 1 of the switch (on the left-hand side) to the GND pin on the Arduino.

Step 2. Connect the yellow wire from pin 2 of the switch (on the right-hand side) to pin 8 on the Arduino. It is depicted in Figure 6.14.

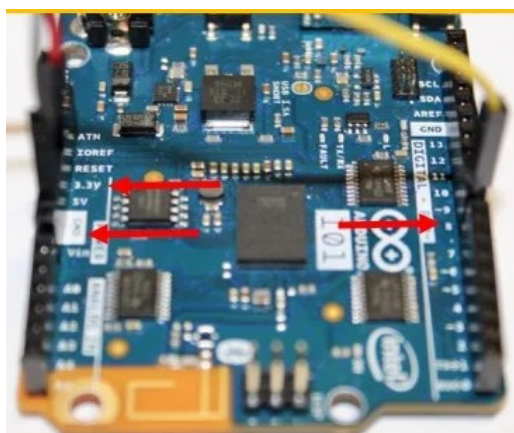


Fig. 6.14: Connection on Arduino board

Step 3. Attach the 10kΩ resistor (brown-black-orange) to pin 2 of the switches. Connect the other terminal of the resistor to the 3.3V pin on the Arduino as shown in Figure 6.15 and 6.16.

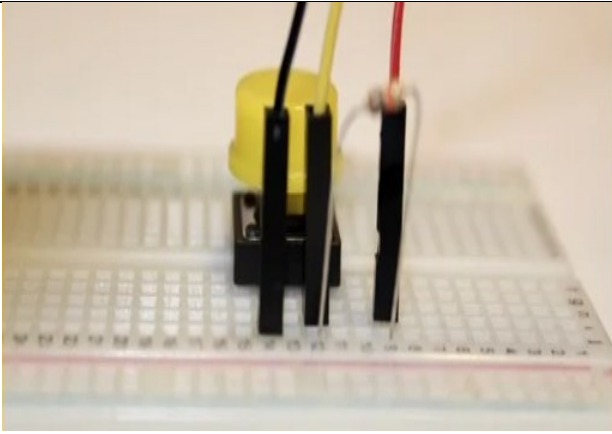


Fig. 6.15: Connection of switch

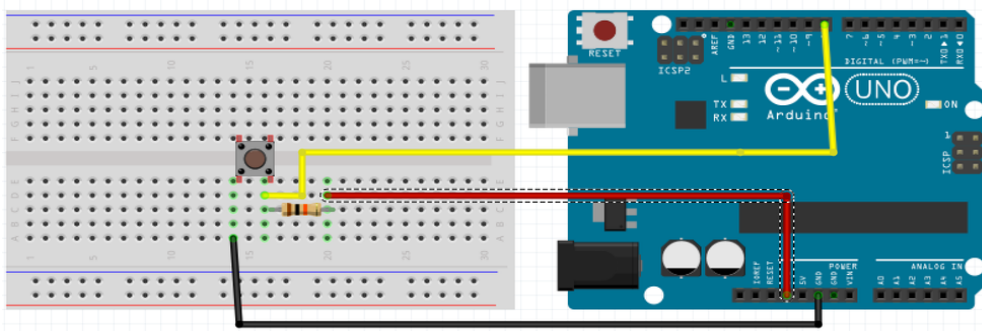


Fig. 6.16: Connection of switch and Arduino board

Step 4. The Blue wire is connected from the signal pin of the servo motor to pin 3 of the Arduino.

Step 5. The Red wire is connected from the signal pin of the servo motor to the 5V pin of the Arduino.

Step 6. The Black wire is connected from the ground pin of the servo motor to the GND pin of the Arduino.

It is illustrated in Figure 6.17 and 6.18.

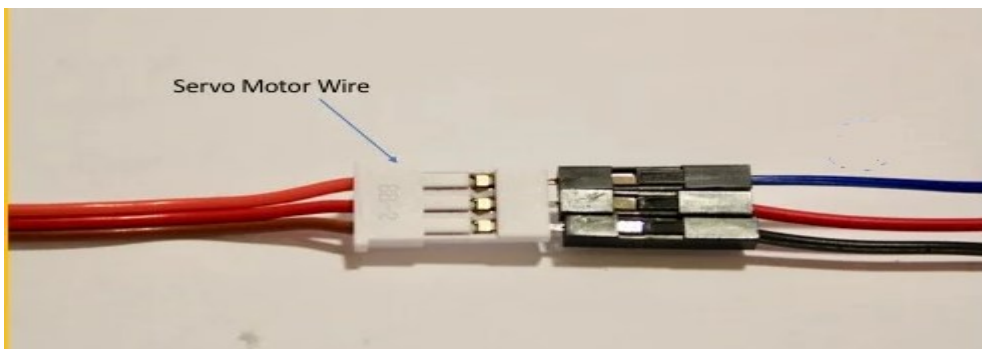


Fig. 6.17(a): Servo motor wires connection

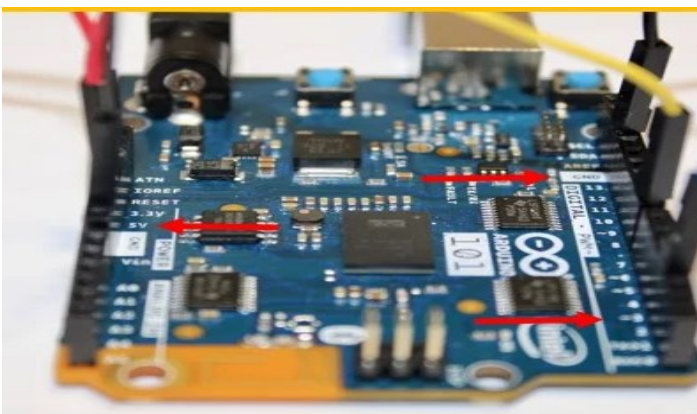


Fig. 6.17(b): Servo motor wires connection pin in Arduino

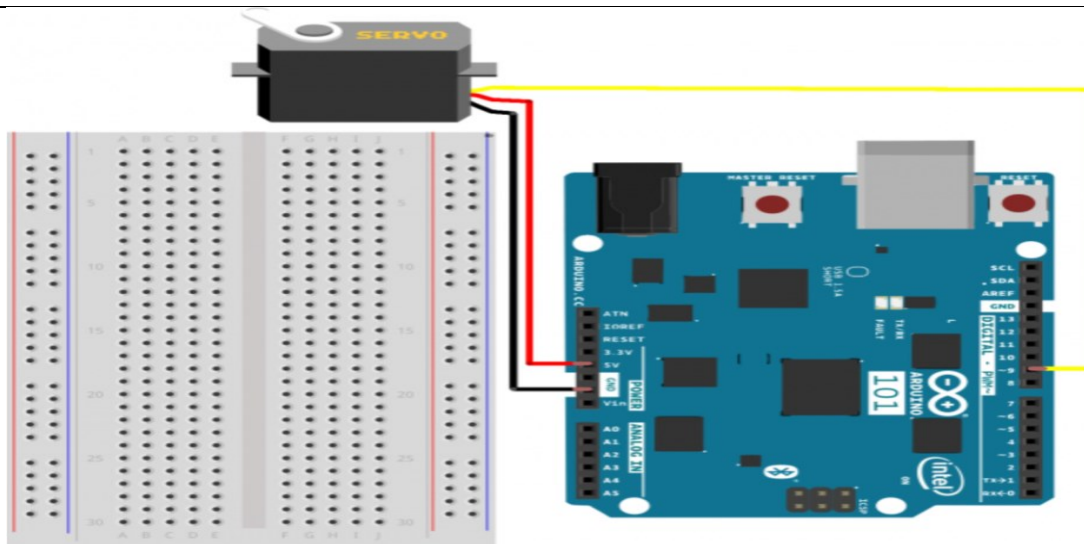


Fig. 6.18: Connection of servo motor and Arduino board

Step 7. The switch is also connected on the breadboard. The Arduino has multiple GNDs that are all common.

Step 8. Push the button and watch the servo motor moving to a different angle each time.

Step 9. Test the working of servo motor. If the movement changes with the application of the switch, motor is working.

The complete set up is shown in Figure 6.19.

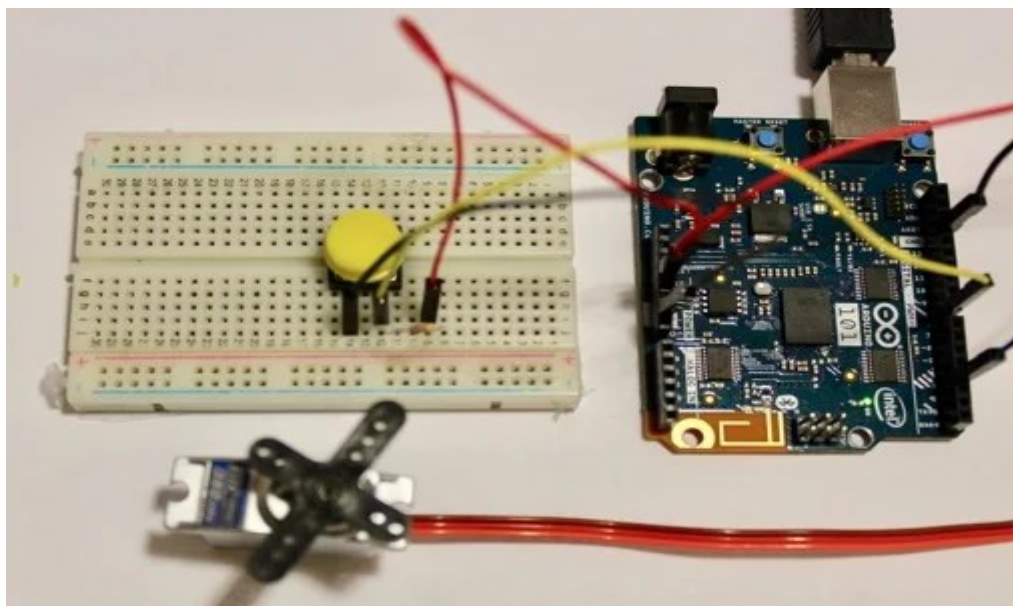


Fig. 6.19: Set up for testing of servo motor

6.10 Testing of Connectivity Modules

The following parameters should be considered to check the working of an on-board Wi-Fi or a 3G, 4G connectivity module at the nodes:

- Range
- Bandwidth
- Intermittent connectivity
- Security

Range

The range, which is the typical data transmission distance for Internet of Things devices connected to a network, can be used to describe the network. A software tool on the wireless devices, such as green, yellow, or red-light

indicators, tracks the transmission strength and signal strength. When the indication is green, the signal is strong and excellent. As one moves farther away from the router or access point, the signal strength drops and the indicator turns yellow. When the signal strength is at its lowest, the indication turns red.

Bandwidth

The rate at which data can be gathered from IoT devices and transmitted upstream is constrained by bandwidth, which is the amount of data that can be transmitted in a given amount of time. The following elements must be considered:

- The amount of data that each device generates
- The number of devices deployed in a network
- The mode of data transmission—a continuous stream or intermittent bursts, depending on how much bandwidth is available to handle peak times.

Intermittent Connectivity

First, it's necessary to determine if the problem is with the wireless or wired network. The following actions can be taken to accomplish this:

- Constantly ping both the router and Google's public DNS (8.8.8.8) from the laptop's terminals.
- There must be a problem with the wireless connection if packet loss is detected in both IPs.
- The wired connection should be having problems if the packet loss is limited to 8.8.8.8.

Following the identification of the wire/wireless connection issue, the following actions ought to be taken:

- Verify the setup.
- Incorrect IP address, subnet mask, and gateway configuration can cause syncing failure, which will complicate testing. Add Internet and DNS access via router to avoid any further issue.

Security

Since security should always come first, make sure the networking technologies you choose include end-to-end security features like encryption, authentication, and open port protection. For instance, the security model included in IEEE 802.15.4 offers features like replay protection, message integrity, message confidentiality, and access control. ZigBee and other technologies that are based on this standard are examples of these technologies.

6.11 End-to-End Testing of IoT network

Testing the reliability and functionality of an IoT network is crucial to ensure seamless operation and data integrity. For instance, consider a smart home system consisting of various IoT devices such as smart lights, thermostats, and security cameras all connected to a central hub. In testing this network, engineers would simulate real-world scenarios to verify that each device can communicate effectively with the hub and with each other. They would test scenarios like turning on/off lights remotely, adjusting thermostat settings from a mobile app, and receiving alerts from security cameras.

Testing of an IoT network can be examined by end to end testing of the system. Here an example of working of a heat pump using an IoT network is taken to understand it better. This works in following steps:

Step1. Install the required application on the mobile phone from the Play Store.

Step2. Connect the app to the appliance using the internet. Here wi-fi is used to connect it.

Step 3. Give the control commands from the mobile app to the heat pump using wi-fi connection.

Step 4. Now collect the data from the sensor used in the heat pump. It is shown as live feedback in the figure.

Step 5. If data received from the pump is accurate and the pump operates in accordance with the commands, it means the IoT network functions properly.

It is illustrated in the following Figure 6.32 for a better understanding.

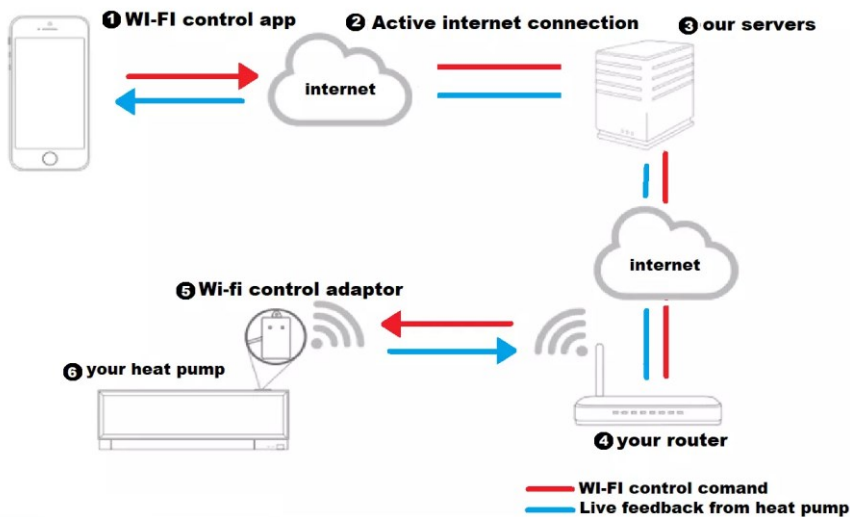


Fig. 6.32: End to end testing of IoT network

With an internet-connected phone, we can control every aspect of our appliance, including temperature control and self-cleaning. It is made possible by an IoT system that uses an appliance to carry out the necessary tasks.

Practical Activity 6.5. Demonstrate to check the connectivity of ESP8266 NodeMCU to an LCD display unit.

Material required

1. I2C LCD Display Module PCF8574
2. 16×2 LCD Display
3. Connection Wires
4. Single row female connector
5. ESP 8266 NodeMCU

In IoT, LCD is required to monitor errors and connection related issues. I2C based display interface uses only two IO lines.

Procedure

Step 1. Connect the GND pin on the LCD display to one of the GND pins on the NodeMCU.

Step 2. Connect the VCC pin on the LCD display to the VIN pin on the NodeMCU.

Step 3. Connect the SDA pin on the LCD display to the D2 pin on the NodeMCU.

Step 4. Connect the SDL pin on the LCD display to the D1 pin on the NodeMCU. It is as shown in Figure 6.33.

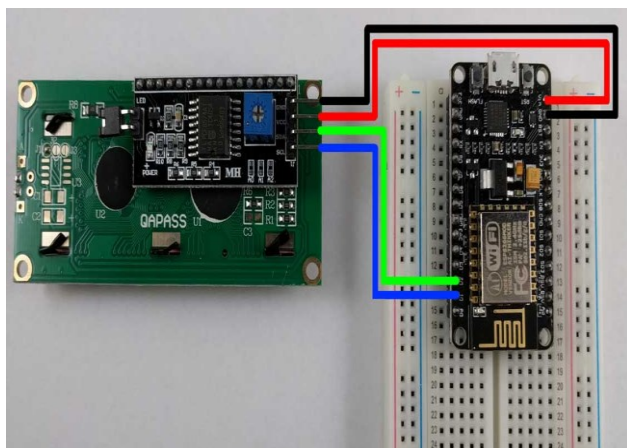


Fig. 6.33: Connection of ESP8266 NodeMCU to an LCD display unit

Step 5. Write and upload following sketch in Arduino IDE for ESP8266 and I2C LCD Interfacing

```
#include <LiquidCrystal_I2C.h>
```

```
// Construct an LCD object and pass it the
```

```

// I2C address, width (in characters) and
// height (in characters). Depending on the
// Actual device, the IC2 address may change.
LiquidCrystal_I2C lcd(0x3F, 16, 2);
void setup() {
  // The begin call takes the width and height. This
  // Should match the number provided to the constructor.
  lcd.begin(16,2);
  lcd.init();
  // Turn on the backlight.
  lcd.backlight();
  // Move the cursor characters to the right and
  // zero characters down (line 1).
  lcd.setCursor(5, 0);
  // Print HELLO to the screen, starting at 5,0.
  lcd.print("HELLO");
  // Move the cursor to the next line and print
  // WORLD.
  lcd.setCursor(5, 1);
  lcd.print("WORLD");
}
void loop() {
}

```

Step 6. The LCD display works by first moving the cursor to where you want to start and then printing some characters. In this example, HELLO and WORLD to be centered on each line. For "HELLO", the cursor needed to be 5 characters from the right and zero characters down, so it is moved to (5, 0). For "WORLD", it is needed to be 5 characters to the right and one character down, so it is moved to (5, 1). It is shown below in Figure 6.34.

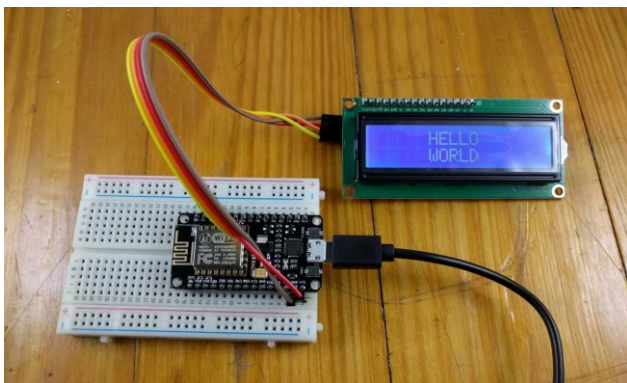


Fig. 6.34: LCD display "HELLO WORLD"

Result: IoT network is connected and tested successfully and the output is shown on display as desired.

Summary

- The chapter focuses on IoT testing using a medical healthcare tracking system as an example.
- It explores IoT levels, testing challenges, methodologies including manual and automated testing, and relevant tools.
- Testing of sensors, actuators, and network connectivity is discussed in detail, emphasizing the critical importance of verifying IoT connectivity for seamless operation.

- The chapter concludes with a discussion on end-to-end testing of an IoT network, demonstrated through an example scenario involving the control of a heat pump.

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. What is the primary purpose of the medical healthcare tracking system described in the chapter? (a) To record information about a patient's health (b) To monitor temperature levels in air conditioners (c) To test IoT applications for performance and behaviour (d) To evaluate the security of IoT-enabled devices
2. Which level of IoT involves simple applications with limited complexity and controls through the Internet? (a) Level 1 (b) Level 2 (c) Level 3 (d) Level 4
3. What is the main security concern with IoT systems according to the chapter? (a) Data encryption during transmission (b) Password security for user interfaces (c) Protection against hardware malfunctions (d) Compatibility with different operating systems
4. Which tool is recommended for monitoring traffic in the interface and source/destination host addresses in IoT systems? (a) Tcpdump (b) Wireshark (c) JTAG Dongle (d) Digital Storage Oscilloscope
5. How can the range of an IoT network be characterized? (a) By the volume of data generated by devices (b) By the strength of the Wi-Fi signal (c) By the number of devices deployed (d) By the type of network cables used
6. Which level of IoT involves the use of physical devices for connections? (a) Level 1 (b) Level 2 (c) Level 3 (d) Level 4
7. Which type of testing is crucial for ensuring IoT systems comply with regulatory and compliance requirements? (a) Usability testing (b) Regulatory testing (c) Compatibility testing (d) Performance and real-time testing
8. What is the primary challenge faced by technicians in testing IoT systems due to their interconnected hardware and software components? (a) Device Interaction Module (b) Hardware-Software Mesh (c) Real-time Data Testing (d) User Interface (UI) Testing
9. Which hardware tools are used in IoT testing? (a) Digital Storage Oscilloscope (b) Wireshark (c) Shodan (d) None of these
10. Which is the IoT application level? (a) Level 1 (b) Level 2 (c) Level 3 (d) Level 4

B. Fill in the blanks

1. Usability testing focuses on ensuring that devices are _____ and can be conveniently moved into different segments and areas.
2. _____ testing is essential for IoT systems due to their intricate design.
3. Hardware components in an IoT system, such as _____ and communication gateways, are just as important as the software programs.
4. The primary function of an IoT gateway is to extend connectivity to nodes, allowing them to connect to the _____.
5. _____ characterizes the distances over which IoT devices transmit data in a network.
6. Bandwidth limits the rate at which _____ can be collected from IoT devices and transmitted upstream.
7. Intermittent connectivity refers to the ability of IoT devices to maintain a constant and uninterrupted connection to the _____.
8. Wireshark is an open-source application used to monitor _____ in the interface, source/destination host addresses, etc.
9. Range, _____ are parameters used to check the working of an on-board Wi-Fi or a 3G, 4G connectivity module at the nodes.
10. LPWANs send small data amounts, allowing for _____ with mile-range coverage.

C. State true or False for the following

1. In Level 1 of IoT, data is stored locally, and monitoring and control can be done through a mobile app.

2. Intermittent connectivity refers to the ability of IoT devices to connect and disconnect from a network at regular intervals.
3. In Level 1 of IoT, data analysis is not a part of the process.
4. Usability testing focuses on ensuring that devices are portable and can be conveniently moved into different segments and areas.
5. JTAG is a tool used for troubleshooting network issues in IoT testing by monitoring traffic and packet details.
6. Regulatory testing is not necessary for IoT systems as they are exempt from compliance requirements.
7. Hardware components in an IoT system, such as sensors and communication gateways, are equally important as software programs.
8. The primary function of an IoT gateway is to act as a router for nodes to connect directly to the Internet.
9. Bandwidth limits the rate at which data can be collected from IoT devices and transmitted upstream.
10. Wireshark is primarily used for testing the range of IoT devices in a network.

D. Short Answer Type Questions

1. What is the purpose of Level 1 in IoT architecture?
2. Why is usability testing important for IoT devices?
3. Why is compatibility testing essential for IoT systems?
4. What is the significance of regulatory testing in IoT?
5. Why are hardware components crucial in IoT systems?
6. What is the primary function of an IoT gateway?
7. How does bandwidth impact data collection in IoT?
8. What does intermittent connectivity refer to in IoT?
9. What is the role of Wireshark in monitoring network traffic?
10. Name one parameter used to check the working of on-board connectivity modules in IoT nodes.

Chapter 7. Testing storage, power supply and internet connectivity of IoT boards

Testing the storage, power supply, and internet connectivity of IoT boards is crucial for ensuring their reliability and functionality. This involves evaluating storage capacity and speed, checking power compatibility and consumption, and verifying seamless connectivity to networks like Wi-Fi or Ethernet. These tests are essential for validating the performance of IoT boards in diverse applications.

7.1 Testing on-board Memory Storage

Testing on-board memory storage in IoT boards verifies the capacity, speed, and reliability of storage components like flash memory or microSD cards. This includes assessing read/write speeds, data integrity, and compatibility with file systems. Thorough testing ensures efficient and reliable data storage for diverse IoT applications.

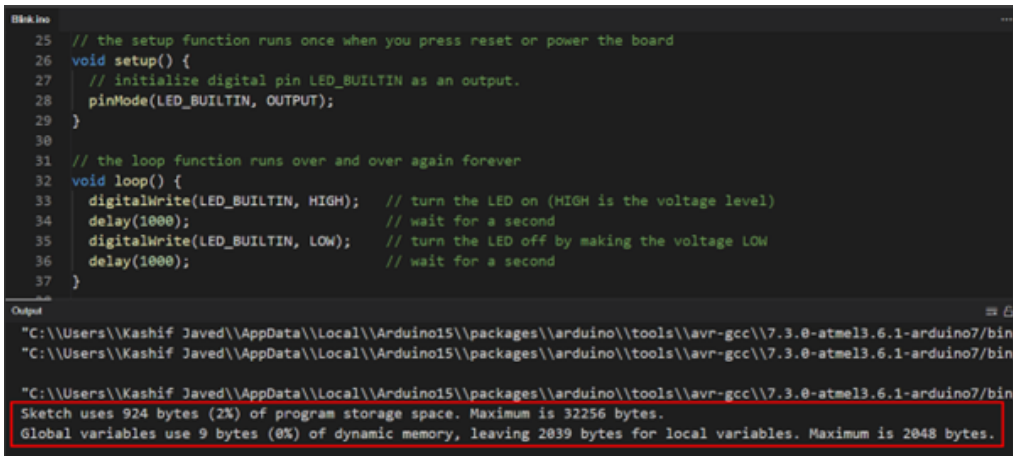
7.1.1 Testing the Storage in Arduino board

Arduino boards do not have on-board storage devices. There are two ways to store data locally for the Arduino as follows:

Local Storage Options for the Arduino

A) It can be stored in a special form of non-volatile memory.

Once the code is uploaded it gives memory status in Arduino board as shown in Figure 7.1.



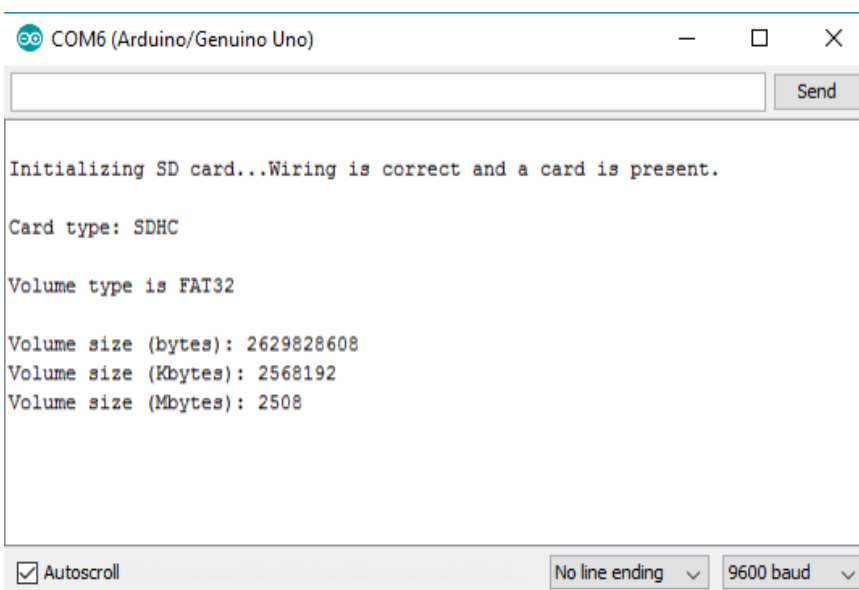
```
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27   // initialize digital pin LED_BUILTIN as an output.
28   pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
34   delay(1000); // wait for a second
35   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
36   delay(1000); // wait for a second
37 }
```

Output

```
"C:\\Users\\Kashif Javed\\AppData\\Local\\Arduino15\\packages\\arduino\\tools\\avr-gcc\\7.3.0-atmel3.6.1-arduino7/bin/
"C:\\Users\\Kashif Javed\\AppData\\Local\\Arduino15\\packages\\arduino\\tools\\avr-gcc\\7.3.0-atmel3.6.1-arduino7/bin/
"C:\\Users\\Kashif Javed\\AppData\\Local\\Arduino15\\packages\\arduino\\tools\\avr-gcc\\7.3.0-atmel3.6.1-arduino7/bin/
Sketch uses 924 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2048 bytes for local variables. Maximum is 2048 bytes.
```

Fig.7.1: Non-volatile memory status in Arduino

B) It can be stored on an SD card, hosted via either a special SD card shield or an Ethernet shield (most Ethernet shields have a built-in SD card drive). It is depicted in Figure 7.2.



```
COM6 (Arduino/Genuino Uno)
```

Send

```
Initializing SD card...Wiring is correct and a card is present.

Card type: SDHC

Volume type is FAT32

Volume size (bytes): 2629828608
Volume size (Kbytes): 2568192
Volume size (Mbytes): 2508
```

☒ Autoscroll No line ending 9600 baud

Fig. 7.2: SD card Memory status in Arduino

7.1.2 Testing the Storage in Raspberry pi board

Raspberry Pi boards come with a Secure Digital (SD) drive and can accept USB-based storage devices where data can be stored. The steps that could be used to check the on-board memory storage card for storing node data in Raspberry Pi are as follows:

Step 1. Open a new command line/terminal session as shown in Figure 7.3.

```
ncert@raspberrypi: ~
File Edit Tabs Help
ncert@raspberrypi:~ $
```

Fig. 7.3: New command line

Step 2. Run the command “Df” and press enter. This will display the total disk usage for the SD card in columns as shown in Figure 7.4.

```
ncert@raspberrypi:~ $ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev            3186080         0   3186080   0% /dev
tmpfs           645852         932    644920   1% /run
/dev/sda1       99574896 6758140  87712424   8% /
tmpfs           3229244         0   3229244   0% /dev/shm
tmpfs           5120          4      5116   1% /run/lock
tmpfs           645848        36    645812   1% /run/user/1000
ncert@raspberrypi:~ $
```

Fig.7.4: Total disk usage

Step 3. To make it more readable, add the -h flag (Df-h), which will add G and M units for gigabytes and megabytes as shown in Figure 7.5.

```
ncert@raspberrypi:~ $ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev            3186080         0   3186080   0% /dev
tmpfs           645852         932    644920   1% /run
/dev/sda1       99574896 6758140  87712424   8% /
tmpfs           3229244         0   3229244   0% /dev/shm
tmpfs           5120          4      5116   1% /run/lock
tmpfs           645848        36    645812   1% /run/user/1000
ncert@raspberrypi:~ $ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            3.1G     0   3.1G   0% /dev
tmpfs           631M   932K  630M   1% /run
/dev/sda1       95G    6.5G   84G    8% /
tmpfs           3.1G     0   3.1G   0% /dev/shm
tmpfs           5.0M   4.0K  5.0M   1% /run/lock
tmpfs           631M   36K   631M   1% /run/user/1000
ncert@raspberrypi:~ $
```

Fig. 7.5: Total disk usage

7.2 Testing the On-board Power Supply

Testing the on-board power supply of IoT boards ensures they can consistently provide the required voltage and current levels under various conditions. This includes verifying voltage accuracy, current stability, and the PSU's ability to handle transient loads and voltage fluctuations. Adherence to safety standards and efficient thermal performance are also assessed to ensure reliable operation in diverse environments.

7.2.1 Testing the On-board Power Supply in Arduino board

A) Measure Vin

Vin is the voltage of the power supply. The USB supplies a nominal 5V. It is shown in Figure 7.6 as 4.43V.

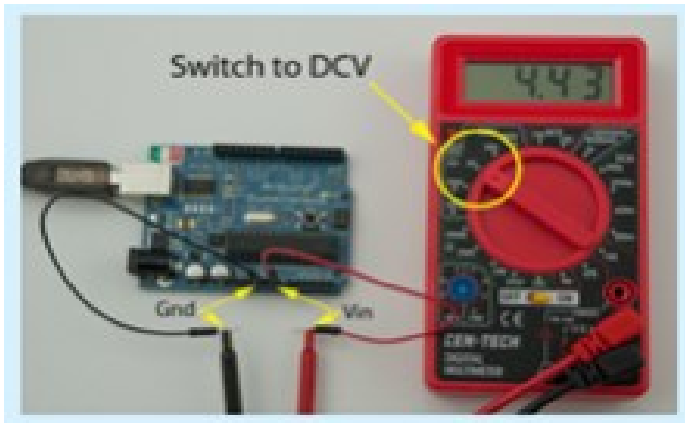


Fig. 7.6: Measure Vin in Arduino board

B) Change power source and measure Vin

Now a 7V DC power supply is plugged into the power jack of the Arduino. It is showing 7.04V as given in Figure 7.7.

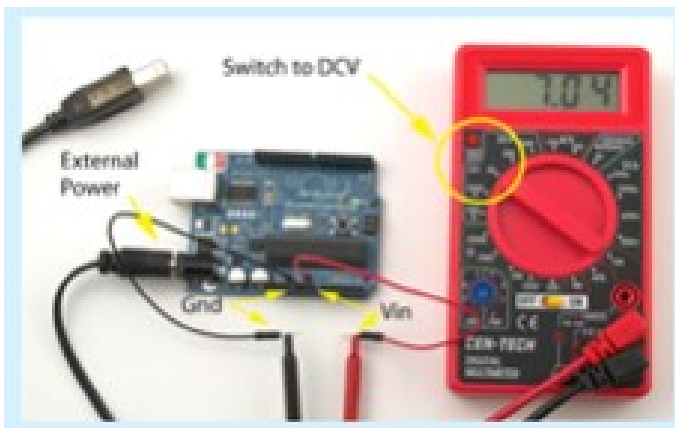


Fig. 7.7: Change power source and measure Vin in Arduino board

C) Check Voltage at 5V Power Pin

The on-board voltage regulator maintains the voltage on the 5V pin at about 5V. The measured voltage is close to 5V target as shown in Figure 7.8.



Fig. 7.8: Change power source and measure Vin in Arduino board

D) Check Voltage at 3V3 Pin

When Arduino board is connected with the computer through the USB cable, it shows an on-board voltage regulator that gives the output of 3.3V as shown in the Figure 7.9.

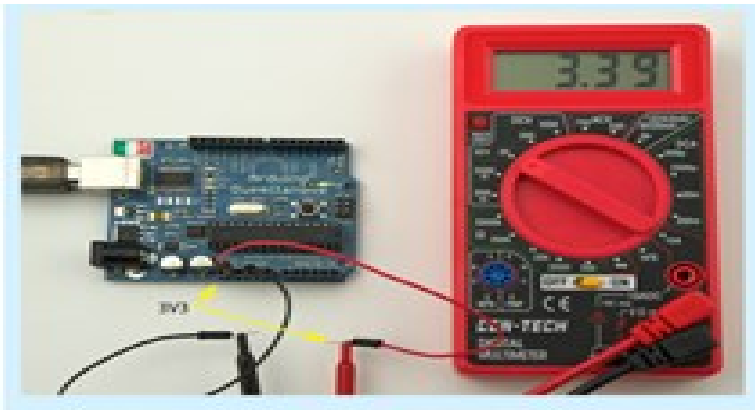


Fig. 7.9: Check Voltage at 3V3 Pin

7.2.2 Testing the On-board Power Supply in Raspberry pi board

The Raspberry Pi is powered by a micro USB supply with a voltage of +5.1V. The amount of current (mA) it requires depends on connected peripherals. Model A can use as little as 500mA when no peripherals are connected, while model B usually needs between 700 and 1000mA. It can draw up to 1 amp of power at its maximum, and if more is needed, connect a USB drive to an externally powered USB hub. Power consumption increases with use of interfaces. GPIO pins have limits: 16mA per pin and 50mA in total. Keyboard and mouse draw 100mA to 1000mA, HDMI port uses 50mA, and the camera module uses 250mA. Ensure the power supply matches the devices' power ratings. Check voltage across the Raspberry Pi to confirm it's between 4.75V and 5.25V. Use a multi meter on TP1 and TP2 for measurement, as shown in Figure 7.10.

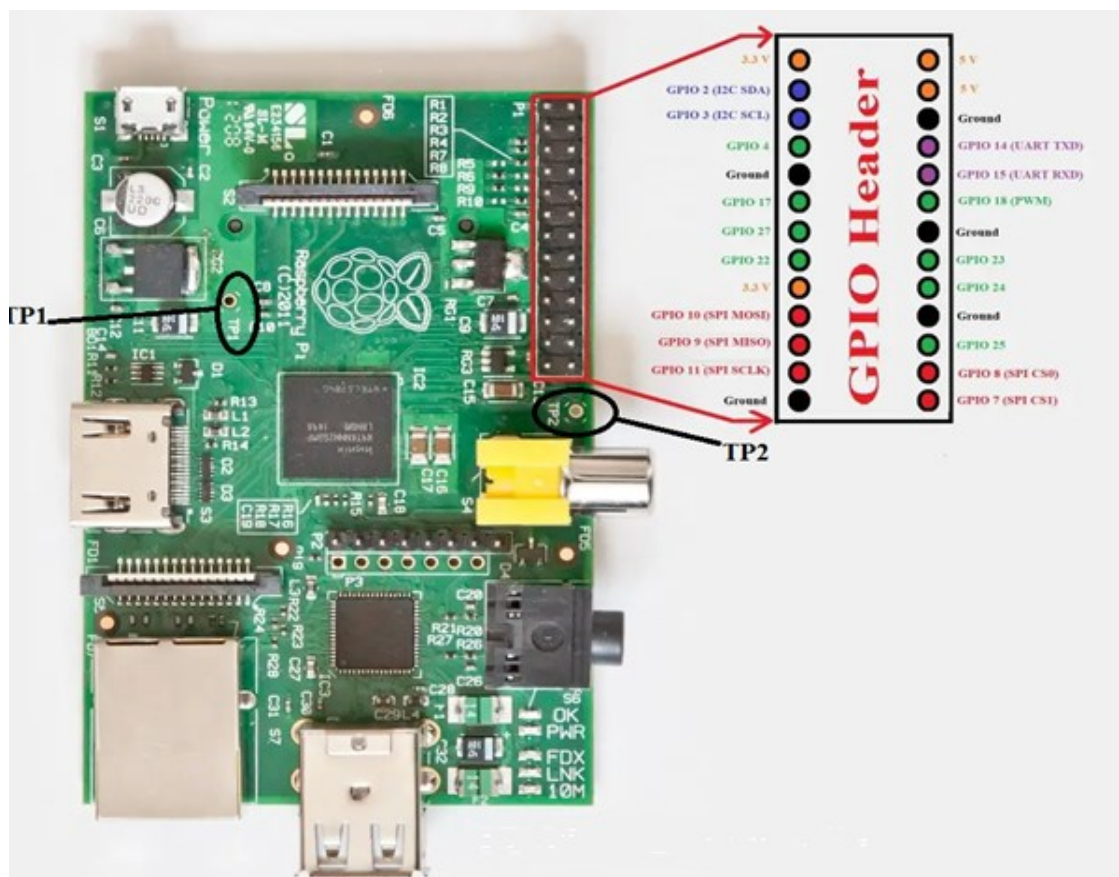


Fig. 7.10: TP1 and TP2 points on Raspberry Pi board

Check the power supply at different hardware configurations as shown below in table 7.1.

Table 7.1: Power supply in Raspberry pi board

Settings of Raspberry pi	Power Supply
--------------------------	--------------

Normal Mode + WLAN + LAN	4.76V
Normal Mode + WLAN	4.80V
Normal Mode + LAN + USB Keyboard + Mouse	4.83V
Normal Mode + LAN	4.87V
Normal / Idle Mode	4.91V
Power Down Mode	5V

Practical Activity 7.1. Demonstrate to verify the available storage space on the on-board memory storage card of a Raspberry Pi.

Material Required

Raspbian OS installed PC/laptop, SD card

Procedure

Step 1. Open a new command line/terminal session on the Raspberry Pi as shown in screenshot in Figure 7.11.

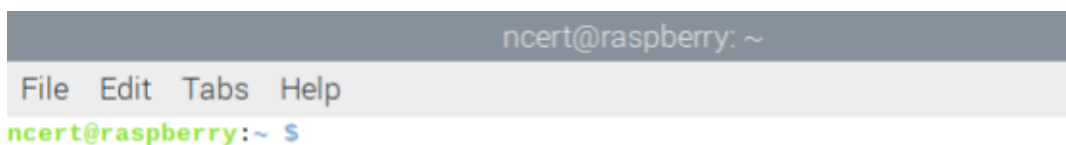


Fig. 7.11: New command line

Step 2. Type the command "df" and press Enter. The window will appear as shown in Figure 7.12.

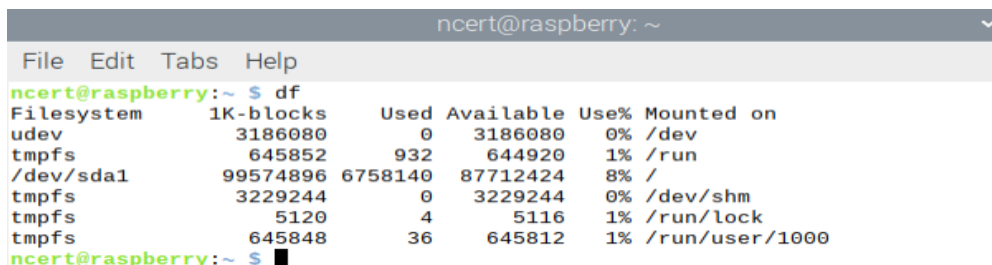


Fig. 7.12: command "df"

Step 3. Review the output to find the information related to the on-board memory storage. This will display the total disk usage for the SD card.

Step 4. To make the output more readable, add the "-h" flag to the command (e.g., "df -h"). This will display sizes in gigabytes and megabytes as shown in Figure 7.13.

```

ncert@raspberrypi: ~
File Edit Tabs Help

ncert@raspberrypi:~$ df
Filesystem      1K-blocks      Used Available Use% Mounted on
udev            3186080         0    3186080   0% /dev
tmpfs           645852         932     644920   1% /run
/dev/sda1       99574896 6758140    87712424   8% /
tmpfs           3229244         0    3229244   0% /dev/shm
tmpfs           5120           4        5116   1% /run/lock
tmpfs           645848         36     645812   1% /run/user/1000

ncert@raspberrypi:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            3.1G     0   3.1G   0% /dev
tmpfs           631M  932K  630M   1% /run
/dev/sda1       95G   6.5G   84G    8% /
tmpfs           3.1G     0   3.1G   0% /dev/shm
tmpfs           5.0M   4.0K   5.0M   1% /run/lock
tmpfs           631M   36K   631M   1% /run/user/1000
ncert@raspberrypi:~$

```

Fig. 7.13: add the "-h" flag to the command

Step 5. Note down the available storage space for future reference and check the size of memory block given in Gigabytes and Megabytes. Like 3186080 converted into 3.1G and 645852 converted into 631M.

Practical Activity 7.2. Demonstrate to test the on-board power supply of a Raspberry Pi under various hardware configurations.

Material Required

Raspberry Pi board set up, multi meter

Procedure

Step 1. Set up the Raspberry Pi with the desired hardware configurations (e.g., WLAN, LAN, USB peripherals).

Step 2. Connect a multi meter to measure voltage across TP1 and TP2 on the Raspberry Pi board.

Fig. 7.14: Raspberry Pi Set up

Step 3. Power on the Raspberry Pi and observe the voltage reading on the multi meter.

Step 4. Record the voltage readings for different hardware configurations (e.g., normal mode with WLAN and LAN, normal mode with WLAN only, etc.).

Step 5. Analyze the voltage readings to ensure they fall within the range of 4.75V to 5.25V, confirming the power supply's proper operation as shown in table.

Settings of Raspberry pi	Power Supply

Normal Mode + WLAN + LAN	4.76V	
Normal Mode + WLAN	4.80V	
Normal Mode + LAN + USB Keyboard + Mouse	4.83V	
Normal Mode + LAN	4.87V	
Normal / Idle Mode	4.91V	
Power Down Mode	5V	

Summary

- The chapter delves into essential aspects of IoT device functionality, namely storage, power supply, and internet connectivity.
- For Arduino, the chapter outlines options for local storage, highlighting different methods and considerations.
- Methods for checking on-board memory storage for Raspberry Pi using commands are explained, emphasizing the importance of data integrity and reliability.
- Insights into the significance of a proper power supply for IoT devices are provided, including considerations for power consumption in various hardware configurations.
- Additionally, the chapter covers basic techniques for checking internet connectivity of boards, ensuring seamless communication and data transmission.
- This chapter serves as a fundamental guide for managing these key aspects of IoT device operation, offering insights and practical tips for ensuring optimal performance and reliability.

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. Which type of storage device does an Arduino board have? (a) On-board memory (b) Secure Digital (SD) drive (c) USB-based storage (d) None of the above
2. What command is used to display the total disk usage for the SD card in a Raspberry Pi? (a) ls (b) cd (c) df (d) sd
3. How can you make the disk usage output more readable in Raspberry Pi? (a) Add -r flag (b) Add -h flag (c) Add -d flag (d) Add -m flag
4. What is the voltage requirement for powering a Raspberry Pi? (a) +3.3V (b) +5.1V (c) +7.5V (d) +12V
5. How much current (in mA) can a GPIO pin draw safely on a Raspberry Pi? (a) 16mA (b) 32mA (c) 50mA (d) 100mA
6. What should be used to measure the voltage across the Raspberry Pi? (a) Multi meter on TP1 and TP2 (b) USB cable (c) Ethernet cable (d) HDMI cable
7. In which mode does a Raspberry Pi use the most power? (a) Normal Mode + WLAN + LAN (b) Normal Mode + WLAN (c) Normal Mode + LAN + USB Keyboard + Mouse (d) Power Down Mode
8. Where can TP1 and TP2 be located on a Raspberry Pi board? (a) On the side edge (b) Near the GPIO pins (c) On the bottom surface (d) In the center of the board
9. Which tool is used to troubleshoot network issues in Raspberry Pi? (a) Wireshark (b) PuTTY (c) WinSCP (d) FileZilla
10. What command is used to open a new command line/terminal session? (a) start (b) run (c) cmd (d) terminal

B. Fill in the blanks

1. Raspberry Pi boards come with a _____ and can accept USB-based storage devices where data can be stored.

2. The model A can use as little as _____ mA when no peripherals are connected, while the model B usually uses between _____ and _____ mA.
3. One GPIO pin can only safely draw _____ mA; however, the GPIO pins as a whole can safely draw _____ mA.
4. A keyboard and mouse can draw as little as _____ mA or as much as _____ mA.
5. An HDMI port uses _____ mA and the camera module _____ mA.
6. To check the voltage across the Raspberry Pi, use a multi meter on _____ and _____.
7. Check the voltage across the Raspberry Pi in various configurations to confirm that the power supply is operating between _____ and _____ V.
8. To check the on-board memory storage card for storing node data in Raspberry Pi, open a new command line/terminal session and run the command_____.
9. The Raspberry Pi can use up to _____of power at its maximum.
10. Power consumption _____ with use of interfaces.

C. State true or False for the following

1. Arduino boards have built-in storage devices.
2. TP1 and TP2 are used to measure the board's voltage on a Raspberry Pi.
3. The model A Raspberry Pi uses between 700 and 1000mA when no peripherals are connected.
4. Data can be stored on an SD card connected to an Arduino using a special SD card shield.
5. Arduino boards can accept USB-based storage devices for data storage.
6. The Raspberry Pi requires more power when its GPIO pins are used intensively.
7. The Arduino has built-in non-volatile memory for data storage.
8. The Raspberry Pi's power supply should operate between 4.75 and 5.25 V.
9. The Raspberry Pi can draw up to 1 amp of power at its maximum.
10. One GPIO pin can safely draw up to 50mA.

D. Short Answer Type Questions

1. How can you check the on-board memory storage card on a Raspberry Pi?
2. What are the two ways to store data locally for an Arduino?
3. What is the voltage requirement for powering a Raspberry Pi?
4. How much current (mA) does a Raspberry Pi model A use when no peripherals are connected?
5. How much power can the Raspberry Pi draw at its maximum?
6. What is the purpose of TP1 and TP2 on a Raspberry Pi board?
7. In what voltage range should the power supply for a Raspberry Pi operate?
8. What should be used if a USB drive needs more than one amp of power to function?
9. What is the function of GPIO pins on a Raspberry Pi?
10. How can you make the "df" command display disk usage in gigabytes and megabytes for better readability?

Data transfer between devices is essential for the functioning of the Internet of Things (IoT). It involves the exchange of data gathered from sensors, actuators, and user interactions, enabling real-time monitoring, automation, and decision-making. This process is crucial for various applications, including smart homes, industrial automation, and healthcare. The efficiency, reliability, and security of data transfer mechanisms are vital for successful IoT deployments.

8.1 Testing the Communication between Devices

8.1.1 Test for Node and Gateway Software

After the final setup of nodes and gateways, the technician needs to check the communication between the devices by running the software at the nodes and gateways. The following are the steps performed in loading the software and testing the communication between the devices:

A) Start the Node

Click on the Start Node on the SNA Node operations toolbar to start the local node. Then, to start the node, specify the configuration on it. Select the files that are configured and click open. After some time, the node will start and establish all the defined links.

B) Check that the Links are Active

As the nodes start, active nodes will appear with names that were specified earlier. To check if the links are active, perform the following steps:

1. Locate the host resource icon which is on the left side of the SNA node operation window.
Select the "+" icon to expand the list of resources.
2. Click the connections (for CPI-C and APPC configurations, click peer connections).
3. Click on the details icon on the toolbar, to get the details about the displayed links.

C) Establish a Session

To establish a session in a configuration, the clients have to establish a session through the gateway to the host. The configuration specified by the client needs to tell the gateway's network address information, such as the token ring address for the gateway's token ring card. For AnyNet socket over the SNA, use the following ping application to reach another socket over SNA node in the network:

D) Ping IP address

If this comes successfully, then a test frame will be received which will indicate that packets were transmitted to the remote node and then they return. Ping application or any other socket application can also be used to reach the local node from another socket over SNA node which is in the same network. Then, run the application on that node only after specifying the local nodes Any Net socket over the SNA IP address.

8.1.2 Checking Communication Link Performance Matrix

Performance Matrix Parameter

Parameters which affect the performance of node and gateway connections are called Performance Matrix parameters. These are as follows:

- MTU
- Data loss
- Delay

A) MTU (Maximum transmission unit)

In computer networking, the MTU is the size of the largest network layer protocol data unit that can be communicated in a single network transaction. Fixed MTU parameters usually appear in association with a communications interface or standard. Some systems may decide MTU at connection time.

Steps to check the best value of MTU for the Internet

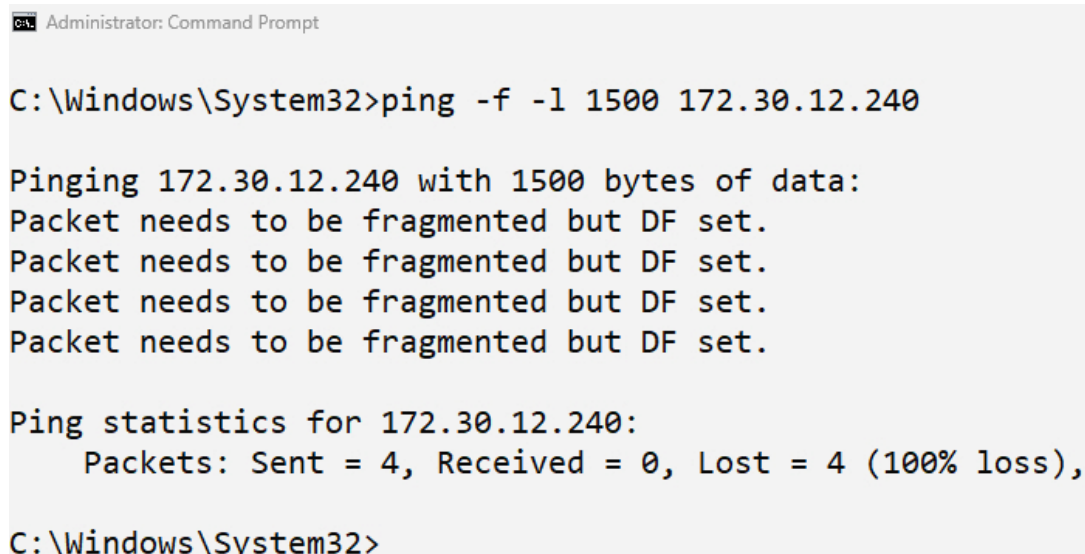
These are given as follows:

Step 1. Open the command prompt window. Give input command as follows:

"ping -f -l xxxx172.30.12.139".

Note: xxxx stands for the value of MTU. The best value of MTU then needs to be attained according to the results.

Step 2. If the result "packet needs to be fragmented but DF set." appears, it means that the value is too big. The screenshot in Figure 8.1 shows checking of MTU.



```
Administrator: Command Prompt

C:\Windows\System32>ping -f -l 1500 172.30.12.240

Pinging 172.30.12.240 with 1500 bytes of data:
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.

Ping statistics for 172.30.12.240:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

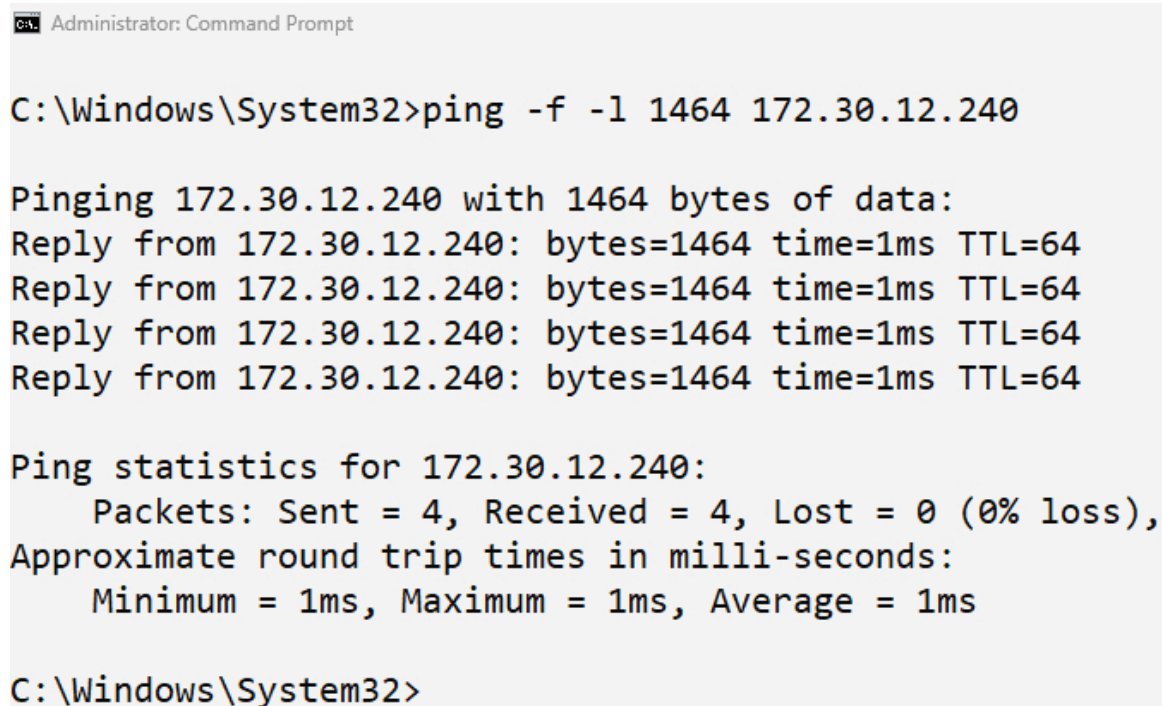
C:\Windows\System32>
```

Fig. 8.1: Checking MTU

Step 3. Decrease the value and test again. If the result comes out as follows, it means that the value is suitable.

"Reply from 172.30.12.139 : bytes=xxxx time=xxxms TTL=xxx"

It is shown in Figure 8.2.



```
Administrator: Command Prompt

C:\Windows\System32>ping -f -l 1464 172.30.12.240

Pinging 172.30.12.240 with 1464 bytes of data:
Reply from 172.30.12.240: bytes=1464 time=1ms TTL=64
Reply from 172.30.12.240: bytes=1464 time=1ms TTL=64
Reply from 172.30.12.240: bytes=1464 time=1ms TTL=64
Reply from 172.30.12.240: bytes=1464 time=1ms TTL=64

Ping statistics for 172.30.12.240:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms

C:\Windows\System32>
```

Fig. 8.2: Unfragmented packet

Step 4. To find the best value of MTU, increase the value and test again until the best value is found, which will not cause the packet to be fragmented. The following screenshot shows unfragmented packets:

Step 5. Since the reality value of MTU should include 20 bytes for IP header and 8 bytes for ICMP header, it needs to add 28 bytes into the result which is obtained from the previous step.

B) Data Loss

In information systems, data loss is a type of condition where information is lost due to mistakes or carelessness during processing, transmission, or storage. Information systems use disaster recovery and backup tools and procedures to stop data loss and recover lost data.

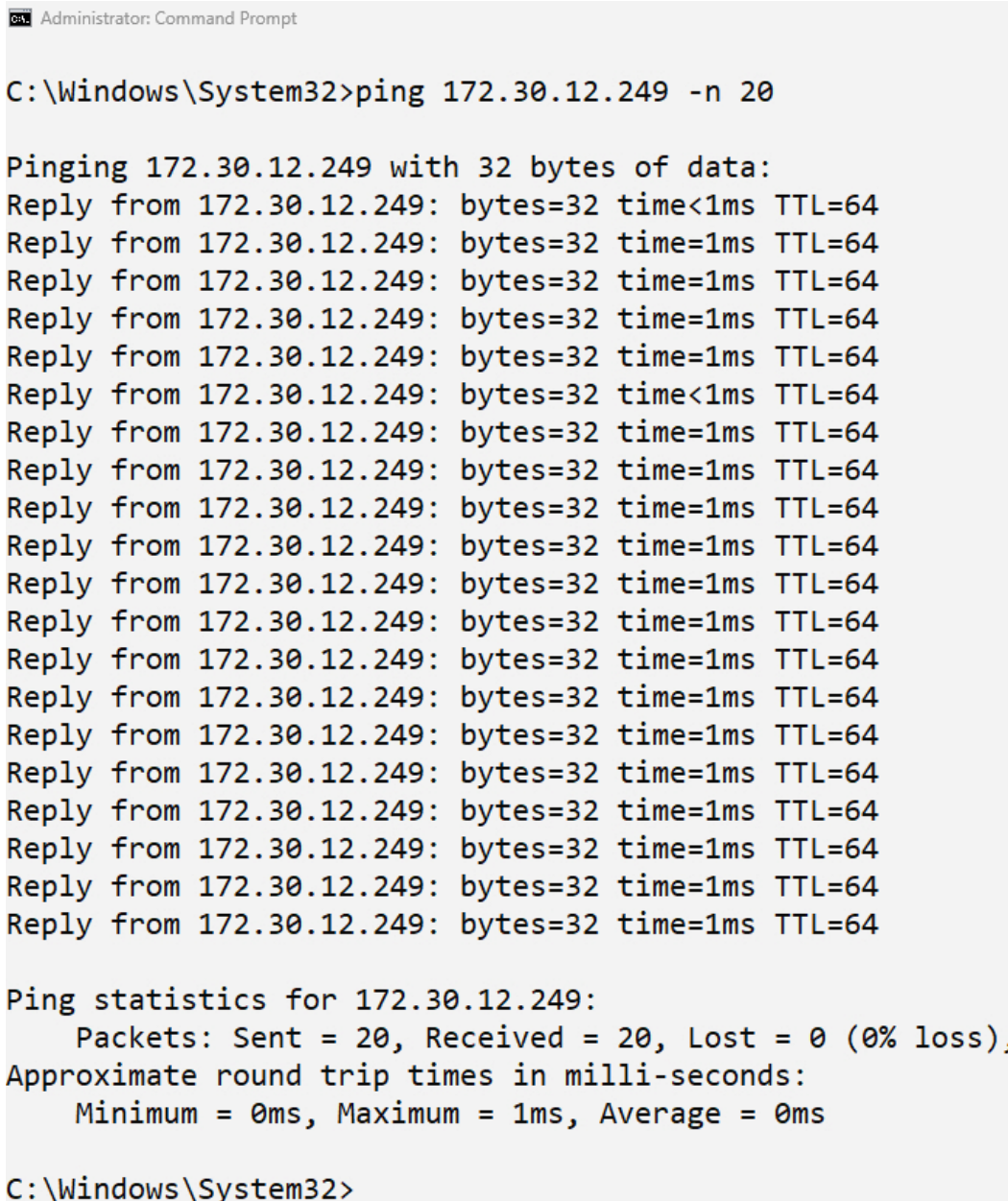
Steps to check Data Loss

Step 1. Begin the packet loss test. Open Windows menu to locate the command processor.

Step 2. Identify the IP address of the gateway.

Step 3. Now, ping the IP of the gateway. The screenshot in Figure 8.3 shows the pinging gateway IP:

Fig. 8.3: Pinging gateway IP



```
Administrator: Command Prompt

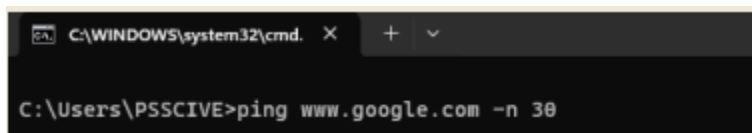
C:\Windows\System32>ping 172.30.12.249 -n 20

Pinging 172.30.12.249 with 32 bytes of data:
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time=1ms TTL=64
Reply from 172.30.12.249: bytes=32 time=1ms TTL=64
Reply from 172.30.12.249: bytes=32 time=1ms TTL=64
Reply from 172.30.12.249: bytes=32 time=1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time=1ms TTL=64
Reply from 172.30.12.249: bytes=32 time=1ms TTL=64
Reply from 172.30.12.249: bytes=32 time=1ms TTL=64
Reply from 172.30.12.249: bytes=32 time=1ms TTL=64
Reply from 172.30.12.249: bytes=32 time=1ms TTL=64
Reply from 172.30.12.249: bytes=32 time=1ms TTL=64
Reply from 172.30.12.249: bytes=32 time=1ms TTL=64
Reply from 172.30.12.249: bytes=32 time=1ms TTL=64
Reply from 172.30.12.249: bytes=32 time=1ms TTL=64
Reply from 172.30.12.249: bytes=32 time=1ms TTL=64
Reply from 172.30.12.249: bytes=32 time=1ms TTL=64
Reply from 172.30.12.249: bytes=32 time=1ms TTL=64
Reply from 172.30.12.249: bytes=32 time=1ms TTL=64

Ping statistics for 172.30.12.249:
    Packets: Sent = 20, Received = 20, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Windows\System32>
```

Step 4. To test the packet loss, ping a website from the network; for example, google.com. The screenshot in Figure 8.4 shows pinging of google.com.



```
C:\WINDOWS\system32\cmd. X + v

C:\Users\PSSCIVE>ping www.google.com -n 30
```

Fig. 8.4: Pinging google.com

Step 5. It will provide the network latency reported as round-trip time (RTT) in milliseconds (ms). The screenshot in Figure 8.5 shows round trip time.

```
C:\Windows\System32>ping google.com -n 20

Pinging google.com [216.58.196.78] with 32 bytes of data:
Reply from 216.58.196.78: bytes=32 time=26ms TTL=115
Reply from 216.58.196.78: bytes=32 time=25ms TTL=115
Reply from 216.58.196.78: bytes=32 time=25ms TTL=115
Reply from 216.58.196.78: bytes=32 time=25ms TTL=115
Reply from 216.58.196.78: bytes=32 time=25ms TTL=115
Reply from 216.58.196.78: bytes=32 time=25ms TTL=115
Reply from 216.58.196.78: bytes=32 time=25ms TTL=115
Reply from 216.58.196.78: bytes=32 time=26ms TTL=115
Reply from 216.58.196.78: bytes=32 time=25ms TTL=115
Reply from 216.58.196.78: bytes=32 time=25ms TTL=115
Reply from 216.58.196.78: bytes=32 time=29ms TTL=115
Reply from 216.58.196.78: bytes=32 time=26ms TTL=115
Reply from 216.58.196.78: bytes=32 time=26ms TTL=115
Reply from 216.58.196.78: bytes=32 time=25ms TTL=115
Reply from 216.58.196.78: bytes=32 time=25ms TTL=115
Reply from 216.58.196.78: bytes=32 time=25ms TTL=115
Reply from 216.58.196.78: bytes=32 time=26ms TTL=115
Reply from 216.58.196.78: bytes=32 time=25ms TTL=115
Reply from 216.58.196.78: bytes=32 time=25ms TTL=115

Ping statistics for 216.58.196.78:
    Packets: Sent = 20, Received = 20, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 25ms, Maximum = 29ms, Average = 25ms

C:\Windows\System32>
```

Fig. 8.5: Round trip time

C) Delay

The time taken for a bit of data to move from one node or endpoint to another across a network is known as its delay. Usually, it is expressed in fractions or multiples of seconds. Depending on the location of the particular pair of communicating nodes, delays can vary slightly.

Steps for test Delay in Network

Step 1. Enter the commands for testing network and Internet latency directly at the command line.

Step 2. Run a Ping loopback test. The Ping loopback test will test the computer's connection to verify that there are no local hardware problems causing the network or Internet latency issue.

Step 3. Type "Ping 127.0.0.1 -n 20". This IP address is the same for nearly all built in network connections. The "-n 20" extension will send 20 packets of data before terminating the test as shown in the screenshot of Figure 8.6.

```
C:\Windows\System32>ping 127.0.0.1 -n 20
```

```
Pinging 127.0.0.1 with 32 bytes of data:
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Ping statistics for 127.0.0.1:
```

```
    Packets: Sent = 20, Received = 20, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\Windows\System32>
```

Fig. 8.6: Ping loopback test

Step 4. View the statistics. The time it took for the packet of data to travel locally should be less than 5ms and there should be zero packet loss as shown in Figure 8.7.

```
Ping statistics for 127.0.0.1:
    Packets: Sent = 20, Received = 20, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Fig. 8.7: Ping Statistics

Step 5. Run Ping to a remote server. Now it has been verified that the local port is working, ping remote servers to test the latency. It is depicted in the screenshot of Figure 8.8.

```
C:\WINDOWS\system32\cmd. X + v

C:\Users\PSSCIVE>ping 172.30.12.249

Pinging 172.30.12.249 with 32 bytes of data:
Reply from 172.30.12.249: bytes=32 time=2ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64

Ping statistics for 172.30.12.249:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

C:\Users\PSSCIVE>
```

Fig. 8.8: Ping to a remote server

Step 6. Normal latency varies by the type of connection from 5ms - 40ms for cable modem, 10ms- 70ms for DSL, 100ms - 220ms for dial-up and 200ms – 600ms for cellular. The distance to the remote server also adds to latency.

Step 7. Type “Ping” followed by the IP address or site URL to be pinged and press Enter as shown in the screenshot of Figure 8.9.

```
C:\WINDOWS\system32\cmd. X + v

C:\Users\PSSCIVE>ping 127.0.0.1

Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\PSSCIVE>
```

Fig. 8.9: Ping to an IP address or site URL

Step 8. View the report. As the test pings the remote address, it will report back the results; the final number after the “time = “is the time it took, in milliseconds, for the packet to travel to the remote site and back to the computer.

Step 9. Run Traceroute test. The traceroute test will show the path that the data travels from the computer to the remote server and any delay in that path. This can be helpful in determining the source of network or Internet delays.

Step 10. Type “tracert” followed by the IP address or site URL to route and press Enter as shown in screenshot of Figure 8.10.

```

C:\Users\PSSCIVE>tracert www.google.com

Tracing route to www.google.com [142.250.193.36]
over a maximum of 30 hops:

  1    1 ms    <1 ms    <1 ms    172.30.12.249
  2    3 ms    <1 ms    <1 ms    14.139.54.177
  3    *      *      *      Request timed out.
  4   19 ms   22 ms   21 ms   10.131.10.17
  5   21 ms   21 ms   23 ms   10.255.238.1
  6   18 ms   18 ms   18 ms   10.152.7.214
  7   44 ms   45 ms   44 ms   72.14.204.62
  8   34 ms   35 ms   34 ms   142.251.49.177
  9   43 ms   38 ms   39 ms   108.170.248.170
 10   33 ms   34 ms   33 ms   216.239.48.65
 11   76 ms   77 ms   74 ms   142.250.230.117
 12   41 ms   39 ms   40 ms   142.251.255.57
 13   68 ms   77 ms   80 ms   142.251.54.91
 14   36 ms   37 ms   35 ms   dell1s15-in-f4.1e100.net [142.250.193.36]

Trace complete.

C:\Users\PSSCIVE>

```

Fig. 8.10: Run Traceroute test

Step 11. View the results. As the test traces the path, it will display each address along the way and the time it took for a data packet to travel and acknowledge receipt for each “hop” along the path. The more “hops” or other devices the data packet needs to route through, the more delay will be experienced.

Practical Activity 8.1. Demonstrate to check MTU value.

Step 1. Open the command prompt window.

Step 2. Identify the IP address of the system.

```

C:\Windows\system32\cmd.exe

C:\Users\Computer Center>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::b28d:5c6:accc:819c%14
    IPv4 Address. . . . . : 172.30.12.130
    Subnet Mask . . . . . : 255.255.254.0
    Default Gateway . . . . . : fe80::c6e9:aff:fe48:3fd5%14
                                fe80::5ad5:6eff:fe6d:8674%14
                                fe80::c6e9:aff:fee7:b22b%14
                                172.30.12.249

C:\Users\Computer Center>

```

Step 3. Input the command "ping -f -l 1800 172.30.12.130" (1800 is the value of MTU).

Step 4. If the result shows successful transmission, note down the value as suitable.

C:\Windows\system32\cmd.exe

C:\Users\Computer Center>ping -f -l 1800 172.30.12.130

Pinging 172.30.12.130 with 1800 bytes of data:

Reply from 172.30.12.130: bytes=1800 time<1ms TTL=128

Reply from 172.30.12.130: bytes=1800 time<1ms TTL=128

Reply from 172.30.12.130: bytes=1800 time<1ms TTL=128

Reply from 172.30.12.130: bytes=1800 time<1ms TTL=128

Ping statistics for 172.30.12.130:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Computer Center>

Step 5. If "packet needs to be fragmented but DF set." appears, decrease the value of MTU and test again.

Practical Activity 8.2. Demonstrate to check the data loss in the network.

Step 1. Begin the packet loss test by opening the command prompt.

Step 2. Identify the IP address of the gateway.

C:\Windows\system32\cmd.exe

C:\Users\Computer Center>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . :

Link-local IPv6 Address : fe80::b28d:5c6:accc:819c%14

IPv4 Address. : 172.30.12.130

Subnet Mask : 255.255.254.0

Default Gateway : fe80::c6e9:aff:fe48:3fd5%14

fe80::5ad5:6eff:fe6d:8674%14

fe80::c6e9:aff:fee7:b22b%14

172.30.12.249

C:\Users\Computer Center>

Step 3. Ping the IP address of the gateway.

C:\Windows\system32\cmd.exe

```
C:\Users\Computer Center>ping 172.30.12.249
```

```
Pinging 172.30.12.249 with 32 bytes of data:
```

```
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
```

```
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
```

```
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
```

```
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
```

```
Ping statistics for 172.30.12.249:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\Users\Computer Center>
```

Step 4. Observe the results for network latency (round trip time) and packet loss.

Practical Activity 8.3. Demonstrate to test the delay in the Network.

Step1. Run a Ping loopback test with the command "Ping 127.0.0.1 -n 20" to ensure no local hardware problems.

C:\Windows\system32\cmd.exe

```
C:\Users\Computer Center>ping 127.0.0.1 -n 20
```

```
Pinging 127.0.0.1 with 32 bytes of data:
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Ping statistics for 127.0.0.1:
```

```
    Packets: Sent = 20, Received = 20, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\Users\Computer Center>
```

Step 2. View the statistics for round trip time and packet loss

```
C:\Windows\system32\cmd.exe
C:\Users\Computer Center>ping 172.30.12.249 -n 20

Pinging 172.30.12.249 with 32 bytes of data:
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64
Reply from 172.30.12.249: bytes=32 time<1ms TTL=64

Ping statistics for 172.30.12.249:
    Packets: Sent = 20, Received = 20, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Computer Center>
```

```
C:\Windows\system32\cmd.exe

C:\Users\Computer Center>tracert www.sciencedirect.com

Tracing route to www.sciencedirect.com [162.159.137.70]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms    172.30.12.249
  2    <1 ms    <1 ms    <1 ms    14.139.54.177
  3    *        *        *        Request timed out.
  4    *        *        *        Request timed out.
  5    *        *        *        Request timed out.
  6    14 ms    16 ms    14 ms    10.152.7.38
  7    39 ms    39 ms    39 ms    115.247.69.85
  8    *        *        *        Request timed out.
  9    41 ms    41 ms    40 ms    49.44.187.43
 10   43 ms    44 ms    43 ms    162.159.137.70

Trace complete.

C:\Users\Computer Center>ping -f -l 5000 172.30.12.130

Pinging 172.30.12.130 with 5000 bytes of data:
Reply from 172.30.12.130: bytes=5000 time<1ms TTL=128
Reply from 172.30.12.130: bytes=5000 time<1ms TTL=128
Reply from 172.30.12.130: bytes=5000 time<1ms TTL=128
Reply from 172.30.12.130: bytes=5000 time<1ms TTL=128

Ping statistics for 172.30.12.130:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:

        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Computer Center>
```

Page 132 / 184

The connection between the gateway and the server can be done by performing a few basic steps and checking the ports. The steps performed in checking the data transfer between the gateway and the server are as follows:

- Ensure that the gateway and the security management server are connected properly.
- Check whether the server and the gateway are using the same SIC activation key.
- Ensure that the date and the time of the operating system is accurate. If the security management server and the remote gateways are in two different time zones then the remote gateway needs certification validation.
- Ensure that the connection between the security management server and the remote gateway are allowed, if the security management server is behind any other gateway.
- Check whether the security management server's IP address and name are present in the host file on the gateway.
- If the IP address of the security management server undergoes static network address translation (NAT) by the local security gateway, then add a public IP address of the security management server to the host file on the remote gateway to resolve to its hostname.
- Then, restart the Check point daemon (CPD) with the commands as follows: **#cpwd admin stop-name CPD-path "\$CPDIR/bin/cpd_admin" -command "cpd_admin stop"**
#cpwd_admin start -name CPD -path "\$CPDIR/bin/cpd" -command "cpd"
- Using the command line of the gateway, type: fwunloadlocal. This will remove the security policy from the security gateway server and allow all the traffic through it.

After basic troubleshooting, the technician should perform the basic connectivity check to make sure the gateway and the server are connected properly.

8.2.1 Checking Connectivity

Before performing the connectivity check, make sure that the SIC ports are open. The SIC ports are:

- **Port 18209** – this port is used for communication between the VPN-1/Firewall-1 Module and the certificate authority (status, issue, revoke).
- **Port 18210** – This port is used to pull certificates from the ICA.
- **Port 18211** – This port is used by the cpd daemon on the module to receive the certificate (when clicked on the initialize in the policy editor).

To check if the SIC is listening to its network ports on the checkpoint device (can be a security gateway server or security management server), the following commands are used:

A) For Windows:

Open the CMD and execute:

>netstat -na

The output should be:

TCP 0.0.0.0:135 0.0.0.0:0 LISTENING

as shown in the screenshot in Figure 8.11.

```
C:\Users\Computer Center>netsat -na
'netsat' is not recognized as an internal or external command,
operable program or batch file.
```

```
C:\Users\Computer Center>netstat -na
```

Active Connections

Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:5040	0.0.0.0:0	LISTENING
TCP	0.0.0.0:7680	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49664	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49665	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49666	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49667	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49668	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49669	0.0.0.0:0	LISTENING
TCP	172.30.12.23:139	0.0.0.0:0	LISTENING
TCP	172.30.12.23:51872	23.206.173.72:443	CLOSE_WAIT
TCP	172.30.12.23:51873	23.206.173.72:443	CLOSE_WAIT
TCP	172.30.12.23:51877	152.199.43.62:443	CLOSE_WAIT
TCP	172.30.12.23:51894	20.198.119.143:443	ESTABLISHED
TCP	172.30.12.23:51896	142.250.183.110:443	ESTABLISHED
TCP	172.30.12.23:51898	216.58.203.35:443	ESTABLISHED
TCP	172.30.12.23:51902	142.250.183.106:443	TIME_WAIT
TCP	172.30.12.23:51905	216.239.32.116:443	TIME_WAIT
TCP	172.30.12.23:51908	142.250.192.74:443	ESTABLISHED
TCP	172.30.12.23:51909	142.250.183.1:443	ESTABLISHED
TCP	172.30.12.23:51910	142.251.42.46:443	ESTABLISHED
TCP	172.30.12.23:51911	142.250.67.131:443	ESTABLISHED
TCP	172.30.12.23:51912	74.125.68.113:443	ESTABLISHED
TCP	172.30.12.23:51915	172.217.194.84:443	ESTABLISHED
TCP	172.30.12.23:51917	142.250.194.99:443	ESTABLISHED
TCP	172.30.12.23:51919	157.240.16.52:443	ESTABLISHED
TCP	172.30.12.23:51920	172.253.118.188:5228	ESTABLISHED
TCP	172.30.12.23:51923	142.250.77.238:443	ESTABLISHED
TCP	172.30.12.23:51924	8.8.8.8:443	ESTABLISHED
TCP	172.30.12.23:51925	20.54.24.231:443	ESTABLISHED
TCP	172.30.12.23:51926	172.217.167.37:443	ESTABLISHED
TCP	172.30.12.23:51927	142.250.194.238:443	ESTABLISHED
TCP	172.30.12.23:51928	142.250.195.10:443	ESTABLISHED
TCP	172.30.12.23:51929	142.250.195.10:443	ESTABLISHED
TCP	172.30.12.23:51930	142.250.193.238:443	ESTABLISHED
TCP	172.30.12.23:51931	142.250.183.196:443	ESTABLISHED
TCP	172.30.12.23:51932	18.202.12.139:443	ESTABLISHED
TCP	172.30.12.23:51933	23.63.109.27:443	ESTABLISHED
TCP	172.30.12.23:51934	142.250.206.138:443	ESTABLISHED
TCP	172.30.12.23:51935	142.250.206.138:443	ESTABLISHED
TCP	172.30.12.23:51936	142.251.39.3:443	ESTABLISHED
TCP	172.30.12.23:51937	142.251.39.3:443	ESTABLISHED
TCP	[::]:135	[::]:0	LISTENING
TCP	[::]:445	[::]:0	LISTENING
TCP	[::]:7680	[::]:0	LISTENING
TCP	[::]:49664	[::]:0	LISTENING
TCP	[::]:49665	[::]:0	LISTENING
TCP	[::]:49666	[::]:0	LISTENING
TCP	[::]:49667	[::]:0	LISTENING
TCP	[::]:49668	[::]:0	LISTENING
TCP	[::]:49669	[::]:0	LISTENING

Fig. 8.11: Checking connectivity in windows

Now, for checking connectivity with a specific port, open the CMD and execute:

>netstat -na | findstr 135

As shown in screenshot of Figure 8.12.

```
C:\Users\Computer Center>netstat -na|findstr 135
TCP    0.0.0.0:135          0.0.0.0:0           LISTENING
TCP    [::]:135             [::]:0              LISTENING

C:\Users\Computer Center>
```

Fig. 8.12: Checking for port in windows

B) For Linux

Open the CMD and execute:

<netstat -na

The output should be:

tcp 0 0.0.0.0:111 0.0.0.0:* LISTEN

It is shown in the screenshot of Figure 8.13.

```
File Edit View Search Terminal Help
[rhcsa@backup-source ~]$ clear

[rhcsa@backup-source ~]$ netstat -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:111             0.0.0.0:*              LISTEN
tcp        0      0 192.168.122.1:53        0.0.0.0:*              LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*              LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*              LISTEN
tcp        0      0 127.0.0.1:25            0.0.0.0:*              LISTEN
tcp6       0      0 :::111                  :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::1:631                :::*                    LISTEN
tcp6       0      0 :::1:25                  :::*                    LISTEN
udp        0      0 192.168.122.1:53        0.0.0.0:*              LISTEN
udp        0      0 0.0.0.0:67              0.0.0.0:*              LISTEN
udp        0      0 0.0.0.0:111             0.0.0.0:*              LISTEN
udp        0      0 0.0.0.0:856             0.0.0.0:*              LISTEN
udp        0      0 0.0.0.0:5353            0.0.0.0:*              LISTEN
udp        0      0 0.0.0.0:42339           0.0.0.0:*              LISTEN
udp6       0      0 :::111                  :::*                    LISTEN
udp6       0      0 :::856                  :::*                    LISTEN
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node  Path
unix   2      [ ACC ]     STREAM    LISTENING     7678    /run/systemd/journal/stdout
unix   2      [ ACC ]     STREAM    LISTENING    30579    /tmp/.ICE-unix/1996
```

Fig. 8.13: Checking for port in Linux

Now, for checking connectivity with a specific port, open the CMD and execute:

< netstat -na | grep 80

This command will show all the ports containing this number as shown in screenshot of Figure 8.14.

```

File Edit View Search Terminal Help
[rhcsa@backup-source ~]$ netstat -na | grep 80
unix 2      [ ACC ]     STREAM     LISTENING   25480       private/proxywrite
unix 3      [  ]       STREAM     CONNECTED   32580
unix 3      [  ]       STREAM     CONNECTED   18090
unix 3      [  ]       STREAM     CONNECTED   30480       @/tmp/dbus-huri4X3ZlI
unix 3      [  ]       STREAM     CONNECTED   19801       /run/systemd/journal/stdout
unix 3      [  ]       STREAM     CONNECTED   33808
unix 3      [  ]       STREAM     CONNECTED   32280
unix 3      [  ]       STREAM     CONNECTED   19800
unix 3      [  ]       STREAM     CONNECTED   18091       /run/systemd/journal/stdout
unix 3      [  ]       STREAM     CONNECTED   33809
unix 2      [  ]       DGRAM      CONNECTED   18034
unix 3      [  ]       STREAM     CONNECTED   31580       @/tmp/dbus-sukx4w9Cdx
[rhcsa@backup-source ~]$

```

Fig. 8.14: Checking for port in Linux

The output should be:

TCP 0.0.0.0:18090 CONNECTED

8.2.2 Setting Connectivity Credentials

A) Securing Devices

After completion of the installation of IoT devices with all the connection tests done, the last step is to secure the entire network and the device.

Device security ensures that only a trusted set of devices are used to access the network and the devices which can break the trust or application can be blocked from sending any type of command to the network or accessing any data. A device simulator program that demonstrates the security mechanism is discussed as follows:

- User ID/Password authentication
- OTP authentication
- Server-unique authentication
- Message payload authentication

The screenshot of Figure 8.15 shows a device simulator



Fig. 8.15: Device simulator

B) Authenticating with a User Name and Password

The MQTT protocol provides username and password field in the CONNECT message for the authentication of any device. A client should send a user name and password to connect to any MQTT broker.

The user name is a Unicode Transformation Format 8-bit (UTF-8) encoded string and the password is binary data, each of which has a 65535-byte max. The MQTT protocol which does not encrypt the user name or password unless any transportation encryption is used, is sent in clear text format. The screenshot of Figure 8.16 shows sample coding for username and password in MQTT:

```

1  try {
2      MqttClient securedClient = new MqttClient(broker, clientId, persistence);
3      MqttConnectOptions connOpts = new MqttConnectOptions();
4      connOpts.setCleanSession(true);
5      connOpts.setUserName(userName);
6      connOpts.setPassword(password.toCharArray());
7      System.out.println("Connecting to broker: "+broker);
8      securedClient.connect(connOpts);
9      System.out.println("Connected");
10 } catch(MqttException me) {
11     System.out.println("reason "+me.getReasonCode());
12     System.out.println("msg "+me.getMessage());
13     System.out.println("loc "+me.getLocalizedMessage());
14     System.out.println("cause "+me.getCause());
15     System.out.println("excep "+me);
16     me.printStackTrace();
17 }

```

Fig. 8.16: User name and password fields in MQTT

C) Authenticating with OTP Authentication

In MQTT's provided authentication mechanism, the IoT application needs to implement additional security to identify any device which tries to connect. An OTP based authentication approach is developed for these situations. This is useful for protecting the device from improper use by eliminating any kind of risk occurring due to unauthorised users gaining access to it.

When the OTP authentication is enabled, the device sends an OTP request to the IoT broker application after start-up by using normal MQTT messaging. The Figure 8.17 shows the flow diagram of the OTP authentication system:

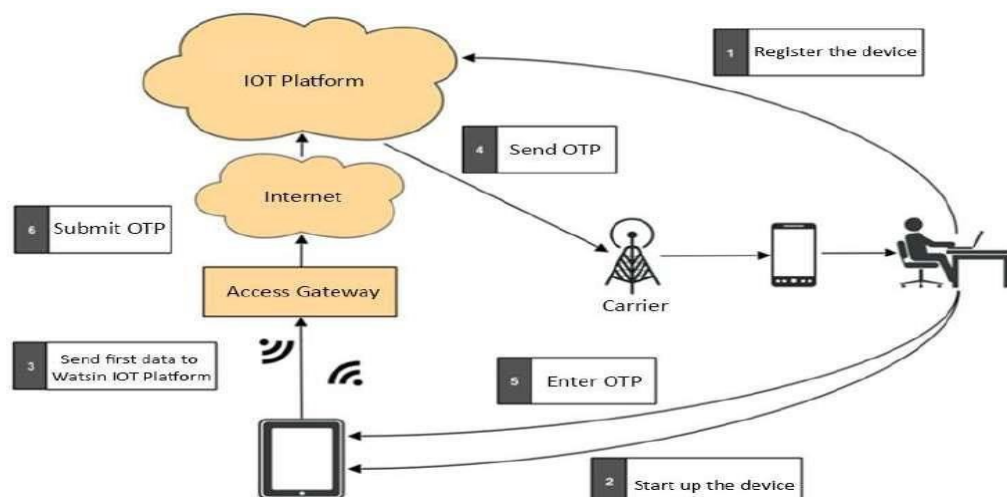


Fig. 8.17: OTP authentication system

A code shows how OTP authentication can be turned ON or OFF by using any of the devices' properties. The code is shown in the screenshot of Figure 8.18.

```

1  // Create the request for OTP
2  JSONObject idObj1 = new JSONObject();
3  try {
4      idObj1.put("event", "server_otp_request");
5      idObj1.put("deviceId", deviceId);
6  } catch (JSONException e1) {
7      System.out.println("Exception occurred");
8      e1.printStackTrace();
9  }
10 new SendMessageToServer("server_otp_request", idObj1).start();
11 System.out.println("otp request sent....");
12 }

```

Fig. 8.18: Code for OTP authentication in MQTT

The IoT app then generates an OTP, which is sent to the device owner separately by sending the notification to the device. The screenshot of Figure 8.19 shows the OTP generation code in MQTT:

```

1  otp = IOTSecurityUtil.generateOTP();
2
3  JSONObject jsonObj = new JSONObject();
4  try {
5      jsonObj.put("cmd", "server_otp_response");
6      jsonObj.put("otp", otp);
7      jsonObj.put("appid", strAppId);
8      jsonObj.put("time",
9          new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new Date()));
10
11     // Server starts a timer of 5 mins during which the
12     // OTP is valid.
13     task = new TimeOutTask();
14     t = new Timer();
15     t.schedule(task, 300000000L);
16
17 } catch (JSONException e) {
18     e.printStackTrace();
19 }
20 System.out.println("Sending otp - " + otp);
21
22 // Publish command to one specific device
23 // iot-2/type/<type-id>/id/<device-id>/cmd/<cmd-id>/fmt/<format-id>
24 new SendMessageToDevice(strDeviceId, "server_otp_response", jsonObj)
25     .start();

```

Fig. 8.19: Code for OTP generation in MQTT

After the OTP is generated, the OTP is entered into the device which sends it to the broker app. This app validates the OTP sent by the device and sends a success/failure (in case of incorrect OTP or timeout situation) message to the device. Then the device can retry OTP authentication based on a retry count from the configuration.

If the OTP authentication is not successful even after a retry, the application shuts down. Otherwise, if OTP authentication is not enabled, the device will skip OTP authentication after start up.

The screenshot in Figure 8.20 shows the code for validation of OTP authentication.

```

1  if (receivedOTP.equals(otp)) {
2      if (task.isTimedOut) {
3          // User took more than 100 seconds and hence the OTP is invalid
4          System.out.println("Time out!");
5          otpValidated = false;
6          otpTimeOut = true;
7      } else {
8          System.out.println("OTP validated..");
9          otpValidated = true;
10         otpTimeOut = false;
11     }
12 } else {
13     System.out.println("Incorrect OTP..");
14     otpValidated = false;
15     otpTimeOut = false;
16 }
17
18 JSONObject otpRespObj = new JSONObject();
19 try {
20     otpRespObj.put("cmd", "server_otp_validate");
21     otpRespObj.put("isOTPValid", String.valueOf(otpValidated));
22     otpRespObj.put("isTimeOut", String.valueOf(otpTimeOut));
23     otpRespObj.put("appid", strAppId);
24     otpRespObj.put("time", new SimpleDateFormat(
25         "yyyy-MM-dd HH:mm:ss").format(new Date()));
26 } catch (JSONException e) {
27     e.printStackTrace();
28 }
29 System.out.println("Result of OTP validation - " + otpValidated);
30 // Publish command to one specific device
31 new SendMessageToDevice(strDeviceId, "server_otp_validate",
32     otpRespObj).start();
33
34 }

```

Fig. 8.20: Code for OTP validation in MQTT

Practical Activity 8.4. Demonstrate to find the port 18211 and 127 using netstat command in windows.

Step 1. Open the command prompt and execute

<netstat -na

```

C:\Windows\system32\cmd.exe

C:\Users\Computer Center>netsat -na
'netsat' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Computer Center>netstat -na
Active Connections

```

Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:5040	0.0.0.0:0	LISTENING
TCP	0.0.0.0:7680	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49664	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49665	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49666	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49667	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49668	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49669	0.0.0.0:0	LISTENING
TCP	172.30.12.23:139	0.0.0.0:0	LISTENING
TCP	172.30.12.23:51872	23.206.173.72:443	CLOSE_WAIT
TCP	172.30.12.23:51873	23.206.173.72:443	CLOSE_WAIT
TCP	172.30.12.23:51877	152.199.43.62:443	CLOSE_WAIT
TCP	172.30.12.23:51894	20.198.119.143:443	ESTABLISHED
TCP	172.30.12.23:51896	142.250.183.110:443	ESTABLISHED
TCP	172.30.12.23:51898	216.58.203.35:443	ESTABLISHED
TCP	172.30.12.23:51902	142.250.183.106:443	TIME_WAIT
TCP	172.30.12.23:51905	216.239.32.116:443	TIME_WAIT
TCP	172.30.12.23:51908	142.250.192.74:443	ESTABLISHED
TCP	172.30.12.23:51909	142.250.183.1:443	ESTABLISHED
TCP	172.30.12.23:51910	142.251.42.46:443	ESTABLISHED
TCP	172.30.12.23:51911	142.250.67.131:443	ESTABLISHED
TCP	172.30.12.23:51912	74.125.68.113:443	ESTABLISHED
TCP	172.30.12.23:51915	172.217.194.84:443	ESTABLISHED
TCP	172.30.12.23:51917	142.250.194.99:443	ESTABLISHED
TCP	172.30.12.23:51919	157.240.16.52:443	ESTABLISHED
TCP	172.30.12.23:51920	172.253.118.188:5228	ESTABLISHED
TCP	172.30.12.23:51923	142.250.77.238:443	ESTABLISHED
TCP	172.30.12.23:51924	8.8.8.8:443	ESTABLISHED
TCP	172.30.12.23:51925	20.54.24.231:443	ESTABLISHED
TCP	172.30.12.23:51926	172.217.167.37:443	ESTABLISHED
TCP	172.30.12.23:51927	142.250.194.238:443	ESTABLISHED
TCP	172.30.12.23:51928	142.250.195.10:443	ESTABLISHED
TCP	172.30.12.23:51929	142.250.195.10:443	ESTABLISHED
TCP	172.30.12.23:51930	142.250.193.238:443	ESTABLISHED
TCP	172.30.12.23:51931	142.250.183.106:443	ESTABLISHED
TCP	172.30.12.23:51932	18.202.12.139:443	ESTABLISHED
TCP	172.30.12.23:51933	23.63.109.27:443	ESTABLISHED
TCP	172.30.12.23:51934	142.250.206.138:443	ESTABLISHED
TCP	172.30.12.23:51935	142.250.206.138:443	ESTABLISHED
TCP	172.30.12.23:51936	142.251.39.3:443	ESTABLISHED
TCP	172.30.12.23:51937	142.251.39.3:443	ESTABLISHED
TCP	[::]:135	[::]:0	LISTENING
TCP	[::]:445	[::]:0	LISTENING
TCP	[::]:7680	[::]:0	LISTENING
TCP	[::]:49664	[::]:0	LISTENING
TCP	[::]:49665	[::]:0	LISTENING
TCP	[::]:49666	[::]:0	LISTENING
TCP	[::]:49667	[::]:0	LISTENING
TCP	[::]:49668	[::]:0	LISTENING
TCP	[::]:49669	[::]:0	LISTENING

Step 2. Find the port using command:

<netstat -na | findstr 18211

```

C:\Windows\system32\cmd.exe

C:\Users\Computer Center>netstat -na|findstr 18211

C:\Users\Computer Center>

```

Step 3. Find the port using command:

<netstat -na | findstr 127

```

C:\Windows\system32\cmd.exe

C:\Users\Computer Center>netstat -na|findstr 127
UDP    127.0.0.1:1900      *:*
UDP    127.0.0.1:54891     *:*
UDP    127.0.0.1:62638    *:*

```

Practical Activity 8.5. Demonstrate to find the port 88, 32124 1235 using netstat command in Linux.

Material Required

Computer System with Linux Operating System installed.

Step 1. Open the command prompt and execute the **netstat -na** command.

```

dds@psscive: ~
dds@psscive:~$ netstat -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp      0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp      0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp      0      0 172.30.13.245:46724     57.144.125.32:443       ESTABLISHED
tcp      0      0 172.30.13.245:44882     57.144.125.32:443       ESTABLISHED
tcp      0      0 172.30.13.245:44086     34.107.243.93:443       ESTABLISHED
tcp      0      0 172.30.13.245:32990     34.117.188.166:443      TIME_WAIT
tcp      0      0 172.30.13.245:41990     117.198.143.35:443      ESTABLISHED
tcp      0      0 172.30.13.245:43524     34.36.165.17:443        TIME_WAIT
tcp      0      0 172.30.13.245:33488     172.217.194.188:5228    ESTABLISHED
tcp      0      0 172.30.13.245:51386     57.144.125.32:443       ESTABLISHED
tcp      0      0 172.30.13.245:43348     142.250.66.4:443        ESTABLISHED
tcp6     0      0 :::1716                 :::*                     LISTEN
tcp6     0      0 :::1:631                 :::*                     LISTEN
udp      0      0 0.0.0.0:39398           0.0.0.0:*               *
udp      0      0 127.0.0.53:53           0.0.0.0:*               *
udp      0      0 172.30.13.245:68        172.30.12.249:67        ESTABLISHED
udp      0      0 172.30.13.149:68        172.30.12.249:67        ESTABLISHED
udp      0      0 0.0.0.0:631             0.0.0.0:*               *
udp      0      0 224.0.0.251:5353        0.0.0.0:*               *
udp      0      0 224.0.0.251:5353        0.0.0.0:*               *
udp      0      0 0.0.0.0:5353            0.0.0.0:*               *
udp6     0      0 :::41071                 :::*                     *
udp6     0      0 fe80::8b40:1382:a10:546 :::*                     *
udp6     0      0 fe80::f290:3288:3d5:546 :::*                     *
udp6     0      0 :::5353                  :::*                     *
udp6     0      0 :::1716                   :::*                     *
raw6     0      0 :::58                    :::*                     7
raw6     0      0 :::58                    :::*                     7

Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State      I-Node  Path
unix   3      [ ]         STREAM     CONNECTED  33309    /run/dbus/system_bus_socket
unix   3      [ ]         STREAM     CONNECTED  388678   /run/dbus/system_bus_socket
unix   3      [ ]         STREAM     CONNECTED  35339    /run/user/1000/at-spi/bus
unix   3      [ ]         STREAM     CONNECTED  38203    /run/systemd/journal/stdout
unix   3      [ ]         STREAM     CONNECTED  30372    /run/systemd/journal/stdout
unix   3      [ ]         STREAM     CONNECTED  39022    /run/systemd/journal/stdout
unix   3      [ ]         STREAM     CONNECTED  38971    /run/systemd/journal/stdout
unix   3      [ ]         STREAM     CONNECTED  372751   /run/dbus/system_bus_socket
unix   3      [ ]         STREAM     CONNECTED  195268   /run/dbus/system_bus_socket

```

Step 2. Filter the result to find the port 88, 32124 and 1235 using the grep filter as shown below.

```

dds@psscive:~$ netstat -na | grep 88
tcp        0      0 172.30.13.245:44882  57.144.125.32:443    ESTABLISHED
tcp        0      0 172.30.13.245:33488  172.217.194.188:5228 ESTABLISHED
udp6       0      0 fe80::f290:3288:3d5:546 :::*
unix  3      [ ]          STREAM  CONNECTED  388678
unix  3      [ ]          STREAM  CONNECTED  407888 /run/user/1000/gvfsd/socket-t7Ggph4o
unix  3      [ ]          STREAM  CONNECTED  88857 /run/user/1000/wayland-0
unix  3      [ ]          STREAM  CONNECTED  388879 /run/user/1000/wayland-0
unix  3      [ ]          STREAM  CONNECTED  193388 @/tmp/.X11-unix/X0
unix  3      [ ]          STREAM  CONNECTED  388856
unix  3      [ ]          STREAM  CONNECTED  285288
unix  3      [ ]          STREAM  CONNECTED  37288 /run/user/1000/bus
unix  3      [ ]          STREAM  CONNECTED  388880
unix  3      [ ]          STREAM  CONNECTED  34088 /run/systemd/journal/stdout
unix  3      [ ]          STREAM  CONNECTED  388672
unix  3      [ ]          STREAM  CONNECTED  88840 /run/systemd/journal/stdout
unix  3      [ ]          STREAM  CONNECTED  33588 /run/user/1000/bus
unix  3      [ ]          STREAM  CONNECTED  86881 /run/user/1000/pulse/native
unix  3      [ ]          STREAM  CONNECTED  388677 /run/user/1000/bus
unix  3      [ ]          STREAM  CONNECTED  88839 /run/systemd/journal/stdout
unix  3      [ ]          STREAM  CONNECTED  84188 @/home/dds/.cache/ibus/dbus-CoH6IkWS
unix  3      [ ]          STREAM  CONNECTED  38893
unix  3      [ ]          STREAM  CONNECTED  392880
unix  3      [ ]          STREAM  CONNECTED  388673
unix  3      [ ]          STREAM  CONNECTED  88863 /run/dbus/system_bus_socket
unix  3      [ ]          STREAM  CONNECTED  388674 /run/user/1000/bus
unix  3      [ ]          STREAM  CONNECTED  388855
unix  3      [ ]          STREAM  CONNECTED  401688 /run/user/1000/bus
unix  3      [ ]          STREAM  CONNECTED  38894
unix  3      [ ]          STREAM  CONNECTED  38988
unix  3      [ ]          STREAM  CONNECTED  88849 /run/user/1000/wayland-0
unix  3      [ ]          STREAM  CONNECTED  32388 /run/systemd/journal/stdout
dds@psscive:~$ netstat -na | grep 32124
dds@psscive:~$ netstat -na | grep 1235
dds@psscive:~$

```

Summary

- The chapter emphasizes effective communication between nodes and gateways in a network.
- It outlines steps to initiate local nodes, validate active links, and establish sessions for seamless communication.
- Key parameters affecting communication performance, including MTU, Data Loss, and Delay, are introduced and discussed.
- Practical instructions are provided to optimize MTU, assess data loss, and measure network latency, ensuring reliability.
- The chapter also covers secure data transfer between gateways and servers, addressing troubleshooting and connection verification.
- Device security is highlighted, with authentication methods such as user ID/password and OTP explained.
- Clear steps for implementing OTP authentication for enhanced security are provided, ensuring safe data transmission in the IoT network.

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. What does MTU stand for in computer networking? (a) Maximum Transmission Unit (b) Medium Transfer Unit (c) Minimal Transmission Unit (d) Maximum Transfer Unit

2. How do you determine the best value of MTU for the Internet? (a) By running a ping test with different values (b) By checking the data loss percentage (c) By measuring the network latency (d) By performing a traceroute test
3. How is data loss checked in the network? (a) By expanding the list of resources (b) By running a ping test to a website (c) By testing network and Internet latency (d) By clicking on the detail's icon
4. What is the purpose of the Ping loopback test? (a) To test the connection to a remote server (b) To verify there are no local hardware problems causing latency (c) To check for active links between nodes (d) To establish a session through the gateway
5. What does the Traceroute test reveal about the data's path? (a) The time it took for a data packet to travel (b) The number of devices the data packet needs to route through (c) The IP address of the gateway (d) The best value of MTU for the Internet
6. What is the purpose of OTP authentication in MQTT-based IoT applications? (a) To verify the connection between gateway and server (b) To ensure data integrity during transfer (c) To add an extra layer of device authentication (d) To encrypt user names and passwords
7. When OTP authentication is enabled, what does the device send to the IoT broker application after start up? (a) Activation key (b) OTP request (c) User name and password (d) Certificate
8. What is the maximum size of the password in MQTT? (a) 128 bytes (b) 256 bytes (c) 64 bytes (d) 65535 bytes
9. What is the purpose of checking SIC ports before performing connectivity checks? (a) To ensure data integrity (b) To verify the connection between gateway and server (c) To confirm that the necessary ports are open for communication (d) To encrypt user names and password
10. Which authentication method involves sending a one-time password to the device owner for verification (a) User ID/Password authentication (b) OTP authentication (c) Server-unique authentication (d) Message payload authentication

B. Fill in the blanks

1. To establish a session in a configuration, the clients have to establish a session through the gateway to the _____.
2. To check Data Loss, begin the packet loss test and ping the _____ of the gateway.
3. The time taken for a bit of data to move from one node or endpoint to another across a network is known as its _____.
4. The traceroute test will show the _____ that the data travels from the computer to the remote server and any delay in that path.
5. To check if the links are active, locate the host resource icon and select the " _____ " icon to expand the list of resources.
6. The user name is a Unicode Transformation Format 8-bit (UTF-8) encoded string and the password is _____.
7. MQTT protocol does not encrypt the user name or password unless any _____ encryption is used.
8. If OTP authentication is not enabled, the device will skip OTP authentication after _____.
9. OTP stands for _____.
10. The IoT app then generates an OTP, which is sent to _____ separately by sending the notification to the device.

C. State true or False for the following

1. Active nodes will appear with names that were specified earlier as the nodes start.
2. The MTU value should be increased until it causes the packet to be fragmented.
3. Delay in network communication is expressed in fractions or multiples of milliseconds.
4. Ping application can be used to test communication between nodes and gateways.
5. The traceroute test shows the path that data travels from the remote server to the computer.
6. If OTP authentication fails after a retry, the application continues running.

7. When OTP authentication is enabled, the device sends an OTP request before shutdown.
8. The OTP generation code is sent to the device owner for validation.
9. If OTP authentication is not enabled, the device will skip OTP authentication after start up.
10. Device security ensures that any device can access the network without restrictions.

D. Short Answer Type Questions

1. What is MTU in computer networking, and how can you determine the best value of MTU for the Internet?
2. How can data loss be checked in information systems, and what steps are involved in the process?
3. Define delay in network communication and explain why it can vary depending on the location of the communicating nodes.
4. What is the purpose of running a Ping loopback test, and what does it verify?
5. How can you determine the best value of MTU for the Internet?
6. What information is displayed during the traceroute test, and how is it useful in troubleshooting network delays?
7. Describe the process of OTP authentication in MQTT, including OTP generation and validation.
8. What is the role of user name and password authentication in MQTT?
9. When is OTP authentication useful? How does it work to protect devices?
10. What happens if OTP authentication is not successful even after a retry in an IoT application?

UNIT 3. ORGANISATIONAL PROCESSES AND STANDARDS

Introduction

This unit covers fundamental aspects of organizational processes and standards, project management concepts and applications, and record maintenance and documentation. Mastering these components is essential for technicians operating in the dynamic realm of IoT devices.

This unit will explore the significance of organizational processes, their key elements, and the steps for effective execution.

Additionally, the concept of project management methodologies and the complexities of project implementation in the IoT sector is also discussed here. Some common IoT projects are taken as an example in this unit to understand the implementation procedure.

Furthermore, this unit covers critical role of record maintenance and documentation focusing on methods, formats, and qualities necessary for efficient record management in this field.

Chapter 9. Organizational Processes and Standards

Organizational processes refer to the structured series of steps or activities that are designed to achieve specific objectives within an organization. These processes outline how tasks are to be performed, including the sequence of actions, the responsibilities of individuals or teams involved, and the resources required. Organizational processes can encompass various areas such as production, operations, marketing, sales, finance, human resources, and more. They are crucial for ensuring efficiency, consistency, quality, and adherence to standards within the organization. Examples of organizational processes include project management processes, procurement processes, quality control processes, and employee onboarding processes.

Organizational Processes and Standards are like the instruction manual for a well-functioning business. Just as a manual tells you how to assemble a piece of furniture step by step, these processes and standards guide employees on how to carry out their tasks effectively. They provide a clear roadmap for how things should be done, ensuring consistency and quality across the board. By following these guidelines, teams can work efficiently together, minimizing errors and maximizing productivity. In essence, they serve as the backbone of the organization, ensuring smooth operations and successful outcomes.

The overall productivity and efficiency of the organization may suffer if organizational processes are poorly planned. Whether they are technicians, supervisors, or managers, the primary goal of an organizational process is to increase productivity and efficiency in a group of people. These groups have their own role and responsibility at work.

9.1 Importance of Organisational Processes

Organizational processes are essential for efficient operations and consistent results in any company. They provide structure, clarity, and efficiency by guiding how tasks are performed, reducing errors, and promoting teamwork. These processes ensure quality, adaptability, and compliance with regulations, ultimately contributing to the company's success. Several points that emphasize the importance of processes are as follows:

Consistency – Organizational processes ensure that tasks are completed consistently, following the same steps each time.

Efficiency – They help streamline workflows, saving time and resources by eliminating unnecessary steps or redundancies.

Quality Assurance – By defining standards and procedures, processes ensure that products or services meet quality benchmarks.

Clarity – Clear processes provide employees with guidelines on how to perform their tasks, reducing confusion and errors.

Team Collaboration – Standardized processes facilitate collaboration among team members, as everyone understands their roles and responsibilities.

Adaptability – Processes can be updated or modified to accommodate changes in the organization or industry, ensuring flexibility and responsiveness.

Risk Reduction – By identifying potential risks and implementing controls, processes help mitigate errors or failures.

Customer Satisfaction – Consistent processes lead to consistent outcomes, enhancing customer satisfaction and loyalty.

Performance Monitoring – Processes often include metrics or indicators for measuring performance, allowing for continuous improvement.

Compliance – They ensure that the organization adheres to regulatory requirements or industry standards, minimizing legal risks.

9.2 Elements of Organisational Process

There are three important elements of an organisational process. These are-unity, teamwork and delegation of work. These are explained below in detail.

9.2.1 Unity

Unity within an organization refers to the collective effort and alignment of individuals towards a common goal or purpose. Steps to reach desired goals are decided by the manager and conveyed to the employees. People follow common rules, policy, and strategies. It can be understood by following components:

Shared Goals – In an organization, unity is fostered by establishing shared goals that all members strive to achieve. These goals are often set by upper management and communicated across the organization.

Common Rules and Policies – Unity is reinforced through the establishment and enforcement of common rules, policies, and procedures that govern behaviour and decision-making within the organization. These guidelines ensure consistency and coherence in actions.

Strategic Alignment – To maintain unity, it's essential for all members of the organization to be aligned with the overarching strategies and objectives. This alignment ensures that everyone is working towards the same vision and mission.

9.2.2 Teamwork

In an organisation, different people work in a team to reach a common purpose. People with different skill sets put in their individual efforts to attain success. Team leaders and other high authorities encourage and guide the employees. Teamwork is the collaborative effort of individuals with diverse skills and perspectives working together towards a common purpose. It can be achieved by following key components as given here:

Collaborative Environment – Organizations foster a collaborative environment where individuals from different departments or functions come together to achieve shared objectives. This involves effective communication, mutual respect, and support among team members.

Utilization of Diverse Skill Sets – Teamwork allows organizations to leverage the diverse skill sets and expertise of their employees. By pooling together varied talents and knowledge, teams can tackle complex challenges more effectively.

Leadership and Guidance – Effective teamwork relies on strong leadership and guidance from team leaders and higher authorities. These leaders facilitate collaboration, provide direction, and empower team members to excel in their roles.

9.2.3 Delegation of Work

In an organisation, people are classified based on their skill sets and put in a group. For example, the Human Resources team recruits and supports the employees. The Technical team works on technical-related issues.

Delegation involves assigning tasks and responsibilities to individuals or teams based on their skills and expertise. Here's how it works within an organizational context:

Skill-based Classification – Organizations classify employees into different groups or departments based on their skills, knowledge, and expertise. This allows for more efficient allocation of tasks and resources.

Specialization – Delegation of work enables employees to focus on tasks that align with their strengths and competencies. For example, the HR team handles recruitment and employee support, while the technical team addresses technical issues.

Effective Utilization of Resources – Delegation ensures that tasks are assigned to individuals or teams best equipped to handle them, thereby maximizing productivity and efficiency within the organization.

9.3 Steps of Organisational Process

An organisational process has five major steps as shown in Figure 9.1.



Fig. 9.1: Organisational process

The steps can be described as follows:

9.3.1 Analyse the Target

- This initial step involves recognizing and categorizing the activities necessary to achieve the organizational goal.
- Higher authorities, such as managers and team leaders, play a crucial role in planning and strategizing these actions.
- Analysis includes assessing the current state, identifying potential obstacles or challenges, and determining the resources required for successful execution.

9.3.2 Create a Strategy

- Once the target is analyzed, managers and team lead collaborate to devise strategies aimed at achieving the organizational goal effectively and efficiently.
- Strategies may involve determining the sequence of activities, allocating resources, setting timelines, and identifying key performance indicators (KPIs) to measure progress.
- The strategy should be aligned with the organization's overall mission and vision, considering any external factors that may impact its implementation.

9.3.3 Categorise the Work

- In this step, activities and tasks identified in the strategy are categorized based on their importance, urgency, and dependencies.
- Categorization helps prioritize tasks and organize them into logical sequences, ensuring that resources are allocated efficiently and timelines are met.
- This stage finalizes the workflow and framework of the work to be carried out by individuals or teams within the organization.

9.3.4 Assign and Give Authorization

- Once the work is categorized, the next step is to assign tasks to individuals or teams and grant them the necessary authorization to carry out their responsibilities.
- Assignments should be based on each individual's skills, expertise, and capacity to ensure optimal performance and resource utilization.
- Authorization ensures that employees have the authority and autonomy to execute their tasks effectively, while also holding them accountable for their actions.

9.3.5 Coordinate and Maintain Relationships

- Coordination and relationship management are crucial for ensuring smooth execution of tasks and fostering a positive work environment.
- Effective coordination involves aligning activities, resources, and timelines across different teams or departments to avoid bottlenecks and delays.
- Maintaining healthy relationships with employees is essential for morale, productivity, and retention. Organizations should respect their employees, provide support, and encourage open communication and collaboration.
- By promoting a culture of cooperation and mutual respect, organizations can enhance teamwork, creativity, and overall organizational performance.

9.4 Common Organisational Structure

In organizations, individuals are grouped into departments based on their skills, educational qualifications, and work experience. This division, known as departmentalization, allows for the efficient allocation of resources and encourages specialization. For example, departments like marketing, finance, human resources, and operations each house employees with specific expertise relevant to their functions. This structure enables teams to collaborate effectively on tasks that align with their respective skill sets, contributing to overall organizational efficiency.

Managers play a crucial role in overseeing departmental activities and ensuring alignment with organizational goals. They work closely with employees across different functions, providing guidance, direction, and support. For instance, a marketing manager may collaborate with graphic designers, copywriters, and market researchers to develop and implement marketing strategies. This collaboration ensures that tasks are completed cohesively and in line with the organization's objectives.

Effective collaboration between managers and employees, such as technicians, is essential for achieving operational excellence. Technicians bring specialized technical skills to their departments, contributing to the execution of tasks and the resolution of operational challenges. Their expertise is critical in maintaining the quality and efficiency of processes within their respective domains. By fostering collaboration between managers and employees across various functions, organizations can leverage diverse skill sets to drive innovation, productivity, and ultimately, success. The layout of a general organisational structure is as shown in Figure 9.2.

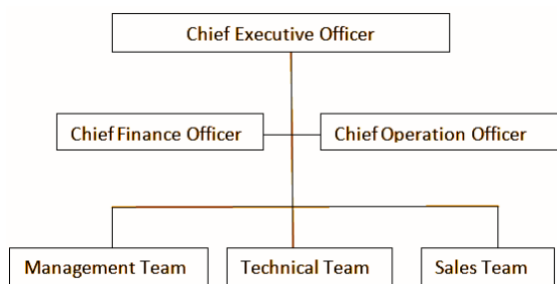


Fig. 9.2: Organisational Structure

It is an example of a three-layer general organisational structure that includes the following levels:

Top-Level Management

Chief Executive Officer (CEO) – Responsible for setting the overall vision, mission, and strategic direction of the company.

Middle Level Management

Chief Operation Officer (COO) – Oversees day-to-day operations and ensures that business activities align with the company's goals.

Chief Finance Officer (CFO) – Manages financial activities, including budgeting, forecasting, financial reporting, and financial strategy development.

Frontline Employees

Management Team

- Depending on the size and structure of the organization, the management team may include executive and senior management roles such as Chief Marketing Officer (CMO), Chief Human Resources Officer (CHRO), Chief Legal Officer (CLO), and Chief Strategy Officer (CSO).
- These executives oversee specific functional areas within the company and collaborate with other members of the management team to achieve organizational goals.

Technical Team

The technical team within an organization is responsible for designing, developing, implementing, and maintaining the technology infrastructure, systems, and solutions that support the company's operations and objectives. Technical team include team Leader/Manager, software engineers, developers, systems administrators, network Engineers, architects, quality assurance, test Engineers, etc.

Sales Team

The sales team within an organization is responsible for generating revenue by selling products or services to customers. This team includes sales Representatives, sales managers, sales support team, etc.

Practical Activity 9.1. Demonstrate the organizational process steps to IoT device installation and configuration

Procedure

According to the five steps of implementation of organisational process, it can be demonstrate as follows:

Step 1. Analysing the Target

Initiation

The company receives a service request from a homeowner, who wants to upgrade his home with smart devices to improve energy efficiency. The homeowner specifically requests the installation of a smart thermostat.

Step 2. Create a Strategy

Planning

A project manager in the company reviews this request and assigns the task to a certified IoT technician.

Step 3. Categorise the Work

IoT technician conducts a preliminary assessment of home to ensure compatibility with the smart thermostat. He creates a detailed installation plan, including a checklist of required materials and tasks.

Step 4. Assign and Give Authorisation

The project manager schedules an appointment with the homeowner for the installation and coordinates with the logistics team to ensure timely delivery of equipment.

Step 5. Co-ordinate and Have Perfect Relationship**Execution**

On the scheduled day, an IoT technician arrives at customer's home with the necessary tools and equipment. He introduces himself as a representative of the company and discusses the installation process with homeowner.

He inspects and determines the best location for the smart thermostat. He carefully installs the device, following manufacturer guidelines and ensuring proper wiring connections.

After installation, technicians configure the smart thermostat to preferences, such as temperature settings and Wi-Fi connectivity. He demonstrates how to use the accompanying mobile app for remote control and monitoring.

Monitoring and Controlling:

IoT technicians conduct thorough tests to verify the functionality of the smart thermostat, ensuring it accurately controls the home's temperature and connects seamlessly to homeowner's Wi-Fi network.

He provides a detailed demonstration of the thermostat's features and functionality, answering any questions and addressing concerns.

He encourages homeowners to reach out to company's customer support team for assistance with troubleshooting or additional guidance.

Closure

Upon successful completion of the installation, the technician hands over a certificate of installation and warranty documentation for the smart thermostat.

The project manager follows up with homeowner to gather feedback on his experience with company's services and ensures his satisfaction.

Company updates its records with details of the installation, including photos and documentation, for future reference and quality assurance purposes.

Summary

- This chapter highlights the significance of adhering to organizational procedures, especially for technicians dealing with IoT devices.
- It emphasizes teamwork, efficient task allocation, and protocol adherence.
- The chapter introduces key elements like Unity, Team Work, and Delegation of Work, along with five crucial steps in an effective organizational process.
- Additionally, it presents a common organizational structure based on skills and qualifications, emphasizing collaborative efforts towards shared goals.

CHECK YOUR PROGRESS**A. Multiple Choice Questions**

1. What is one of the key elements of organizational processes? (a) Unity (b) Independence (c) Competition (d) Isolation
2. Why is it important for a technician handling IoT devices to adhere to organizational procedures? (a) It boosts personal reputation (b) It ensures job security (c) It increases technical skills (d) It leads to promotions
3. What are the three common basic components of an organizational process? (a) Unity, Collaboration, Competition (b) Unity, Team Work, Delegation of Work (c) Individualism, Innovation, Efficiency (d) Strategy, Coordination, Communication
4. How many steps are necessary in an effective organizational process? (a) 3 (b) 5 (c) 7 (d) 10
5. In the context of organizational structure, how are individuals typically grouped based on? (a) Randomly (b) Alphabetically (c) Based on Skill Set and Qualifications (d) Based on Age.

6. What is the primary goal of an organizational process? (a) To increase productivity and efficiency (b) To maximize profits (c) To encourage individual competition (d) To reduce teamwork
7. What is the significance of removing redundant tasks and responsibilities in organizational processes? (a) It increases errors (b) It reduces errors (c) It has no impact on errors (d) It leads to conflict among team members
8. In an organizational structure, how are individuals typically grouped based on their skill set? (a) Based on their favorite tasks (b) Based on their personal preferences (c) Based on their experience and qualifications (d) Based on their age
9. What is the importance of acknowledging each person's unique talents and abilities within an organization? (a) It has no impact on motivation (b) It decreases mental well-being (c) It increases mental well-being and motivation (d) It leads to conflict among team members
10. What is the role of delegation of work in an organizational process? (a) It randomly assigns tasks to team members (b) It classifies people based on their skill sets and assigns tasks accordingly (c) It reduces the need for teamwork (d) It hinders productivity

B. Fill in the blanks

1. Every person who works is a part of an organization, so they must _____ by and follow any procedures established by the organization.
2. Understanding, classifying, and allocating work to the most qualified employee for the job are all part of an _____ process.
3. An organizational process has three common basic components: _____, Team Work, and Delegation of Work.
4. The delegation of work involves classifying individuals based on their _____ and assigning them to specific groups.
5. Removing redundant tasks and responsibilities in organizational processes helps _____ errors while distributing tasks.
6. An organizational process has _____ major steps to bring effectiveness.
7. The primary goal of an organizational process is to increase _____ and _____ in a group of people.
8. Effective organizational processes play a pivotal role in ensuring _____ and _____.
9. Strategies are created by the _____ to achieve a goal effectively and efficiently.
10. In an organisation, _____ work in a team to reach a _____ purpose.

C. State true or False for the following

1. Every individual within an organization must adhere to established procedures.
2. Well-designed organizational processes have no impact on the overall productivity and efficiency of an organization.
3. Delegation of Work involves classifying individuals based on their skill sets and assigning them to specific groups.
4. The primary goal of an organizational process is to decrease productivity and efficiency in a group of people.
5. Removing redundant tasks and responsibilities in organizational processes increases errors while distributing tasks.
6. Unity in an organization emphasizes the collective pursuit of a common goal, guided by the manager's decisions and communicated to employees through established policies and strategies.
7. Effective organizational processes play a pivotal role in ensuring productivity and efficiency within an organization.
8. An organizational process involves three common basic components: Unity, Team Work, and Collaboration.
9. The delegation of work involves randomly assigning tasks to team members without considering their skill sets.

10. In an organization, individuals are typically grouped based on their personal preferences and hobbies.

D. Short Answer Type Questions

1. What is the primary goal of an organizational process?
2. What are the three common basic components of an organizational process?
3. How does delegation of work contribute to the effectiveness of an organizational process?
4. How does well-designed organizational processes impact the productivity and efficiency of an organization?
5. What are the five key steps involved in an effective organizational process, as outlined in Chapter 9?
6. How does adherence to established procedures benefit a technician handling IoT devices, as mentioned in the chapter.
7. Briefly explain the role of teamwork in achieving common objectives within an organization.
8. How does the classification of individuals in an organizational structure contribute to achieving common goals?
9. Briefly explain the importance of removing redundant tasks and responsibilities in organizational processes.
10. Provide a brief explanation of the significance of acknowledging each person's unique talents and abilities within an organization.

Each work process is integrated into a single system through a project. Multifunctional units collaborate to create a single solution to all issues in the expanding telecom sector. As a result, team integration is crucial to the success of IoT projects. IoT projects require complex programming and tool development, as well as strategic thinking and decision-making.

10.1 Project Management

Project management is a discipline that involves planning, organizing, and overseeing the successful completion of a project. A project is a temporary endeavour with a specific goal, a defined beginning and end, and usually constrained by factors such as time, cost, and resources.

Effective project management is crucial for organizations to achieve their objectives, deliver products or services, and adapt to the dynamic business environment.

Every successful project goes through the same five phases: initiation, planning, execution, monitoring and controlling, and closing as shown in Figure 10.1.

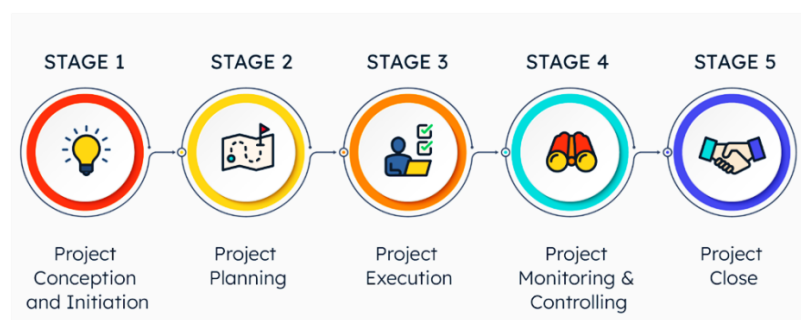


Fig. 10.1: Project Management Phases

The step-by-step progress of any project in these phases can be described as follows:

10.1.1 Project Initiation

This phase defines the project's purpose, scope, and objectives. It works on identifying stakeholders and understanding their needs and expectations. In this phase a feasibility analysis is conducted to assess the project's viability.

10.1.2 Project Planning

In this phase a detailed project plan is developed that includes tasks, timelines, resources, and budget. Here potential risks are identified and risk management strategies are created accordingly. The effective allocation of resources and establishing communication channels are also a part of the planning phase.

10.1.3 Project Execution

This phase includes implementation of the project plan and coordination of people and resources to ensure the alignment with the plan. It also works on responding to any sudden changes in the plan and unforeseen challenges promptly.

10.1.4 Project Monitoring and Controlling

In this phase rigorous monitoring and controlling of the project progress is done to ensure its completion on time. This phase regularly measures and tracks project performance against the established plan. By identifying and addressing issues, risks, and changes as they arise this phase ensures that the project stays on track and meets its objectives.

10.1.5 Project Closing

This phase of project management works on formalizing project completion and obtaining acceptance from stakeholders. Here the project's success is evaluated with releasing of project resources, documenting final reports, and closing contracts.

10.2 Project Handling

When a technician is involved in Internet of Things (IoT) projects, there are specific parameters and considerations that need to be addressed to ensure successful project handling. IoT projects involve the integration of various

devices, sensors, and connectivity to create a network of smart, interconnected systems. For a technician involved in IoT project, it is required to check some parameters related to project handling given as follows:

Cost – Technician should analyse project cost prior to starting the project. It includes the cost of the tools, devices, and implementation methods used in the project.

Tools and Equipment – Tools and equipment used in the project should be effective and easily available. If possible, technicians should use preferred brands that have a high-performance record in the market.

Device Compatibility – Ensure compatibility and interoperability among different IoT devices and sensors. Verify that the devices adhere to industry standards and protocols for seamless integration.

Connectivity – Assess and implement appropriate communication protocols (e.g., MQTT, CoAP) based on project requirements. Consider the factors such as bandwidth, latency, and reliability when choosing connectivity solutions.

Security – Implement robust security measures to protect IoT devices and data from unauthorized access. Use encryption, authentication, and authorization mechanisms to secure communication channels.

Data Management – Define data collection, storage, and processing mechanisms. Establish the protocols for handling large volumes of data generated by IoT devices.

Power Consumption – Optimize power consumption for IoT devices, especially for battery-operated sensors. Implement energy-efficient communication protocols and low-power modes for low power consumption.

Scalability – Design the IoT solution to be scalable, allowing for the addition of new devices and sensors as needed. Consider the potential growth of the IoT network and plan for scalability.

Remote Monitoring and Management – Enable remote monitoring and management capabilities for IoT devices. Implement tools for troubleshooting, software updates, and configuration changes remotely.

Interoperability – Ensure that the IoT devices can seamlessly interact with other systems and platforms. Consider the use of open standards and APIs to facilitate interoperability.

10.3 Project Implementation Process

IoT projects are complex and require many activities and planning to be done. The step wise flow of a typical project implementation process can be designed as follows:

Step 1. Performing detailed analysis on the project – Conducting an in-depth analysis of the project involves scrutinizing its components, methodologies, and outcomes to gain comprehensive insights. This detailed examination enables a thorough understanding of the project's intricacies and contributes to informed decision-making for further advancements.

Step 2. Conducting a meeting with supervisor/seniors to plan a strategy – Engaging in a meeting with supervisors or seniors is essential to collaboratively strategize and outline a cohesive plan, ensuring alignment with organizational goals and effective project execution. This proactive approach fosters clear communication and enhances coordination for successful outcomes.

Step 3. Designing the steps and assigning timelines for the project – Designing the project steps and assigning timelines is a pivotal phase, laying the foundation for structured implementation. This meticulous planning ensures a systematic progression, fostering efficiency and timely completion of each project component.

Step 4. Ask for help from others if needed – Seeking assistance when needed is a strength, not a weakness. It fosters collaboration, accelerates problem-solving, and contributes to a more successful and well-rounded outcome.

Step 5. Assigning tasks to the peer/junior technicians – Delegating tasks to peer or junior technicians involves thoughtful assignment based on strengths and skills, fostering a collaborative environment for efficient project execution. Clear communication and defined responsibilities contribute to a streamlined workflow, ensuring successful completion of each aspect.

Step 6. Managing the work process – Effectively managing the work process requires vigilant oversight, ensuring tasks align with project objectives and timelines. Regular assessments and adjustments contribute to a smooth workflow, optimizing productivity and achieving project goals.

Step 7. Successfully implementing the plan to complete the project – The successful implementation of the plan hinges on meticulous execution, adherence to timelines, and proactive issue resolution. Clear communication and agile adjustments further contribute to achieving project milestones and overall success.

10.4 IoT based Projects Examples

10.4.1 Project on Humidity and Temperature Sensing Device using IoT

Project Title

Prepare an IoT project on a humidity and temperature sensing device which obtains information of the surrounding environment and uploads the data in a cloud network.

Application

This device can be installed in cold storage to obtain the humidity and temperature readings anytime from anywhere. This would help in analysing the change in temperature and humidity and its effect on the stored material.

Objective

The objective is to set up IoT hardware for obtaining humidity and temperature information from a sensor which will be analysed on cloud platforms through wireless Internet.

Requirements

- DHT-11 sensor
- Thing Speak platform
- Arduino MCU
- ESP8266 Wi-Fi module

Block Diagram of the Setup

The Figure 10.2 shows a block diagram for the complete IoT framework for the humidity and temperature sensing device.

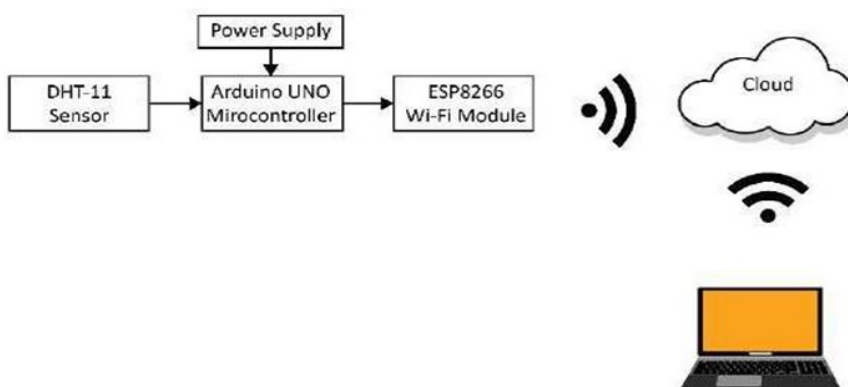


Fig. 10.2: Block diagram for the setup

Circuit Diagram

Circuit diagram for the project is shown in Figure 10.3. It is built around Arduino MCU, DHT11 sensor and ESP8266 Wi-Fi module.

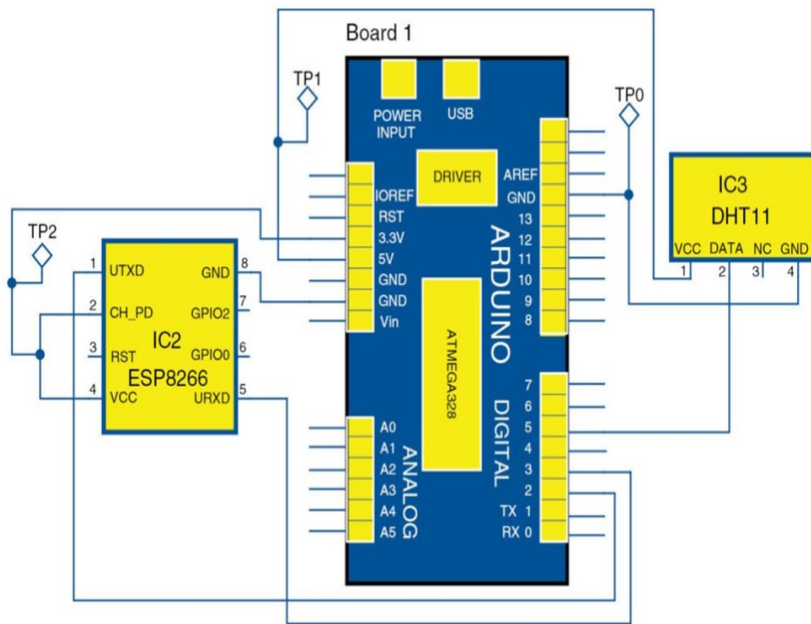


Fig. 10.3: Circuit diagram of humidity and temperature sensing device using IoT

Basic Working of the Circuit

The DHT11 sensor senses the temperature and humidity of the surrounding and then it sends the data to digital pin 5 of the Arduino MCU. From Arduino MCU, the temperature and humidity values are sent to the cloud. These values are sent at regular intervals of time through the ESP8266 Wi-Fi module. Then from the cloud, humidity and temperature values can be seen graphically on Thing Speak platform from anywhere in the world.

Construction of Project

Step 1. Login to the Think Speak by registering. Use the following link for the same: <http://thingspeak.com>

Step 2. After completing the registration, login to the account and create a new channel by providing humidity and temperature in the fields. The following screenshot in Figure 10.4. shows the same.

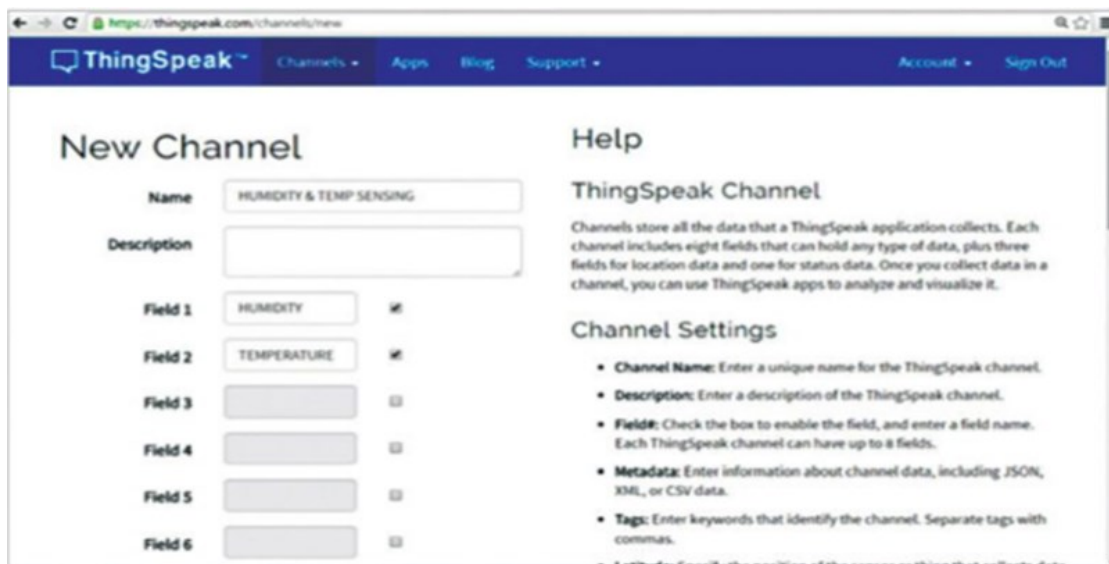


Fig. 10.4: New channel on Thing Speak platform

Step 3. After creating a new channel, it will create the following two API keys:

- Write API key
- Read API key

Step 4. Replace the given line in the program with write API key:

`String apiKey = "NTIM1RXET6YVUVWF ";`

Step 5. Now, provide the Wi-Fi host name and password of the network in the two lines given in the program (IoT.ino):

```
StringHost_Name="Jonah";  
String Password = "2569696";
```

Step 6. Now, the program is verified with the Wi-Fi setup.

Step 7. It uses the DHT library. If the DHT library is not present in your Arduino folder, download it from <https://github.com/adafruit/DHT-sensor-library>. To import the DHT library in Arduino IDE, select Sketch→ Import library→ Add library→ Select the library that you have downloaded.

Step 8. Compile the sketch/program provided and upload it to the Arduino MCU through Arduino IDE.

Step 9. Ensure that the Wi-Fi modem and the Internet connection in your PC/smartphone are working properly.

Step 10. After uploading the sketch, the temperature and humidity value can be taken by logging in to the Think Speak account and it can be seen graphically in the Private View window, as shown in following Figure 10.5.



Fig.10.5: Graphical view of humidity and temperature on Thing Speak platform

10.4.2 Project on Air Pollution Measuring Device Using IoT

Project Title

Prepare an IoT project for air pollution measuring devices which obtains information about air pollution through any smartphone.

Application

This device can be installed in public places like parks and local markets so that it can daily access the pollution, temperature and humidity parameters of the environment. This information can be accessed by people through their smartphones so that they get the information about the pollution level of any area they want while sitting in their homes.

Objective

The objective of this project is to design an IoT-enabled air pollution meter for monitoring air pollution directly on the smart phone through the Blynk application and Arduino board. Blynk stands as an Internet of Things (IoT) platform that enables the control of devices like Arduino, Raspberry Pi, and similar ones over the Internet. In this particular project, Blynk serves as the conduit for a digital dashboard on the smartphone, presenting real-time air quality readings for immediate surroundings.

Requirements

- Arduino board with Arduino shield
- A smartphone with latest android version
- Blynk application available for android phones
- Voltage regulators 7805 (IC1 and IC2),
- Temperature and humidity module DHT11

- Gas sensorMQ135 and PM2.5/PM10 sensor

Keynote:

About Blynk- It can control hardware remotely. It can display sensor data, store and visualise it, among other cool things. There are three major components in the platform as given below:

Blynk app

It allows to create amazing interfaces for various projects using various widgets.

Blynk server

It is responsible for all communication between the smartphone and hardware. You can use Blynk Cloud or run your private Blynk server locally. It is open source, and can handle thousands of devices. It can also be launched on Raspberry Pi.

Blynk libraries

Libraries are available for all popular hardware platforms. The libraries enable communication with the server, and process all incoming and outgoing commands.

When you press a button on the Blynk app on the phone, the signal travels to the Blynk Cloud, where it finds its way to the hardware. It works the same way in the opposite direction, and everything happens in a blink of an eye. The conceptual block diagram as shown in Figure 10.6.

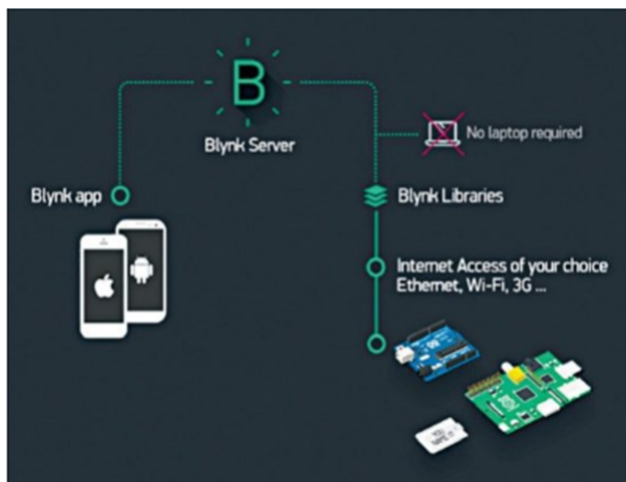


Fig.10.6: Connection of devices using Blynk app

Circuit Diagram

The components in the set-up are connected with each other as follows:

- The main part is the Arduino Uno board with an Arduino shield.
- The voltage regulators 7805 (IC1 and IC2) are connected with CON7.
- The temperature and humidity module DHT11 is connected to CON3.
- The output of the gas sensor is connected to analogue input pin A3 of the Arduino Uno board through CON2. The PM2.5/PM10 sensor is connected to CON1 and this sensor is connected to the UART port (TX and RX) of the Arduino Uno board.
- Connector CON4 is used for connecting 230V AC mains which drives the lamp connected to CON5 and fan connected to CON6.
- A 12V battery is connected to CON7, which is regulated to 5V using 7805 regulators (IC1 and IC2).

The circuit diagram of the project is given below in Figure 10.7.

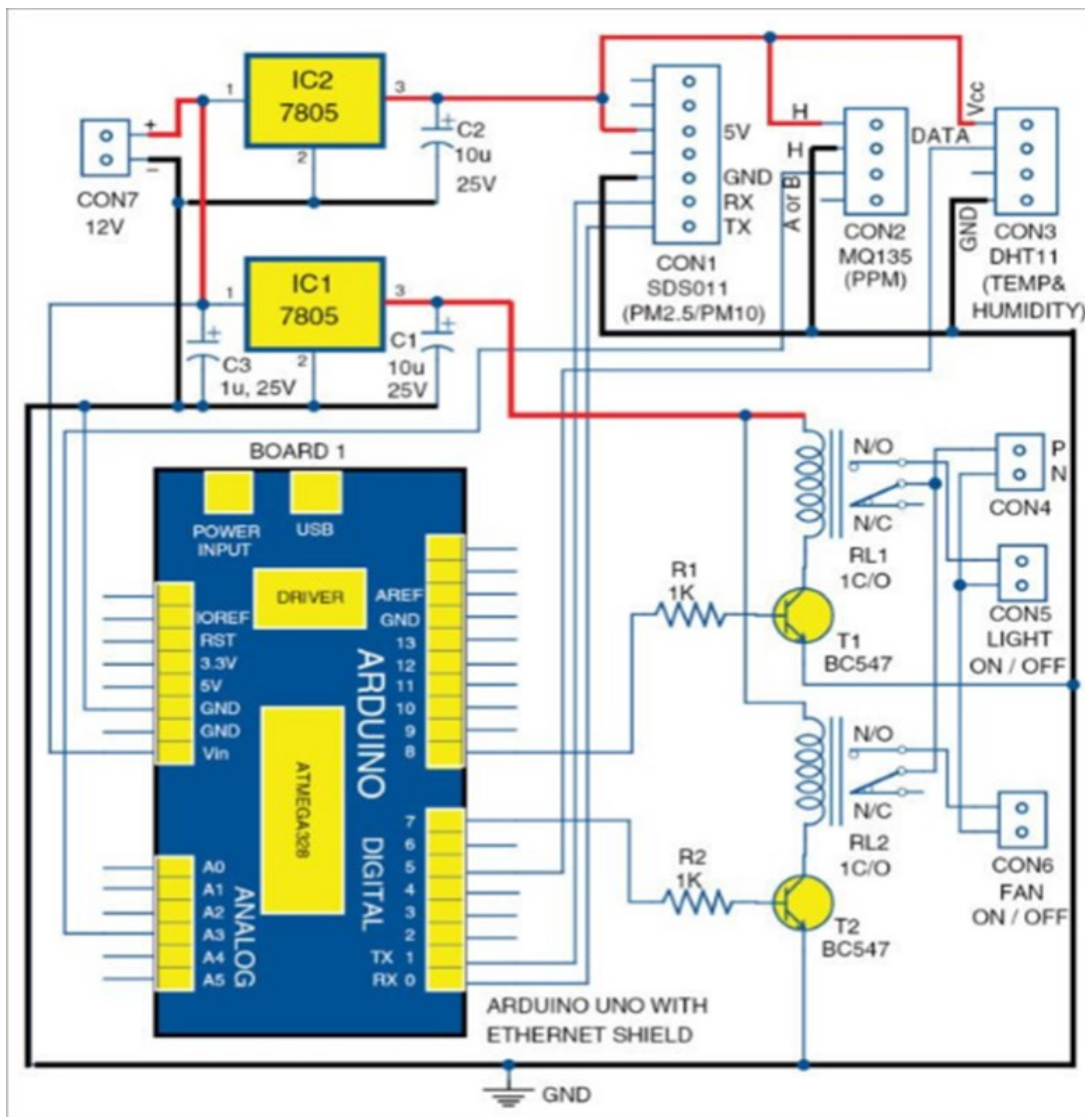


Fig.10.7: Circuit diagram for the air pollution monitoring IoT set-up

Basic Working of the Circuit

The basic working of this project can be described in terms of the functions of components used in the circuit as follows:

Controller and Sensors

Arduino Uno is used as a control and measuring unit, and Arduino Uno Ethernet shield for connection to the Blynk Cloud. The sensors include nova PM sensor SDS011, gas sensor MQ135, and temperature and humidity sensor DHT11.

The PM2.5/PM10 (particulate matter) sensor connected across CON1 uses the principle of laser scattering in the air, and can detect suspended particulate matter concentration ranging from 0.3 to 10 microns. Data collected by the sensor is stable and reliable. The SDS011 sensor is connected to the UART port (TX and RX) of the Arduino Uno board.

Sensitive material of the sensor is tin-dioxide, whose conductivity increases with the concentration of gas. Change in conductivity is converted into output voltage signal, which varies corresponding to the concentration of combustible gas. Gas sensor MQ135 is highly sensitive to ammonia, sulphide and benzene steams, smoke and other harmful gases. It is a low-cost sensor, suitable for different applications. Output of the gas sensor is connected to analogue input pin A3 of Arduino Uno board through connector CON2.

Temperature and humidity sensor (DHT11) is a composite sensor containing calibrated digital signal outputs of temperature and humidity. Connected to connector CON3, it includes a resistive-type humidity measurement component and an NTC temperature-measurement device. Its output pin is connected to digital pin 5 of Arduino Uno board. It is a relatively inexpensive sensor for its performance.

Additional alarm indication and control through relays RL1 and RL2 have been provided. The two appliances (light and fan) can be connected to connectors CON5 and CON6, but more can be connected as per requirement. Connector CON4 is used for connecting 230V AC mains to drive the lamp and fan connected at CON5 and CON6, respectively.

A 12V battery or supply is connected to connector CON7, which is regulated to 5V using 7805 regulators (IC1 and IC2). When the system is powered on, the device takes one minute to preheat the gas sensor before it is ready for operation. The whole sensor circuit can be powered by a 9V or 12V adaptor or battery. Here, IC1 and IC2 have been used for driving various parts of the circuit.

Smartphone and Digital dashboard

Android phones are used as a smart-meter. Blynk app is a well-designed interface builder. It works on iOS and Android, and provides easy-to-use widgets for the LCD, push buttons, on/off buttons, LED indication, RTC and more. Using Blynk digital dashboard features in a smartphone helps reduce the cost of the project because traditional LCD display, mechanical buttons, LEDs, RTC and so on are not required. The digital dashboard or smart meter display is shown in Figure 10.8 given below.

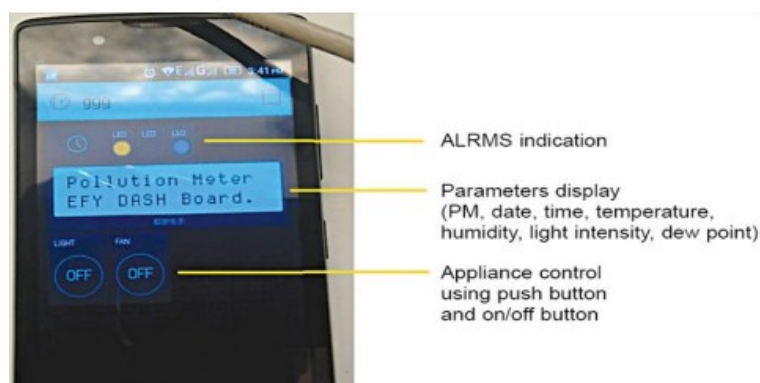


Fig. 10.8: Digital dashboard on Android phone

Arduino Ethernet shield allows you to easily connect Arduino to the Internet. This shield enables Arduino to send and receive data from anywhere in the world, with an Internet connection.

Construction of Project

Step 1. Mount the Ethernet shield on the Arduino Uno board and then connect Board1 to a computer by using a USB cable.

Step 2. In Arduino sketch ethernet client.ino, change the IP address with the system's IP in IP Address.

Step 3. Compile and upload the code into Arduino Uno from Arduino IDE.

Step 4. Now, compile ethernet server.ino sketch and upload it to Arduino Uno board. Change the IP address with the systems IP in IP Address.

Step 5. Navigate to Ethernet shield's IP address, Arduino will respond through HTML browser and will be ready to accept input values from analogue pins (A0 through A5) of Board1.

Step 6. Connect the android mobile with the Wi-Fi.

Step 7. Download and install Blynk app from Google Play Store.

Step 8. Create a new Blynk account. The following Figure 10.9 shows the screen for creating a new Blynk account.

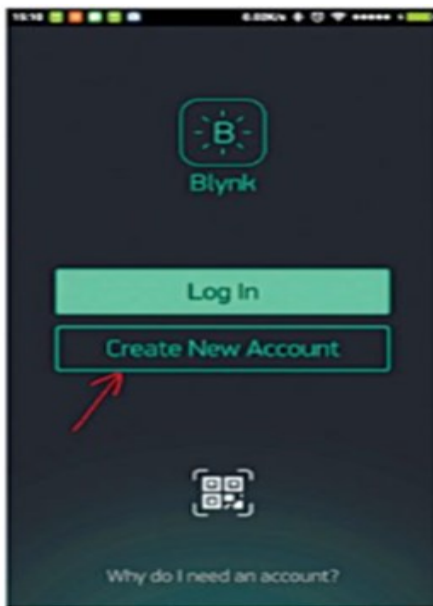


Fig. 10.9: Creating a new account in Blynk

Step 9. After logging into your account, start by creating a new project by giving it a name as shown in following Figure 10.10 for creating a new project:



Fig. 10.10: Creating a new project window

Step 10. Select the hardware model which is used. Select Arduino Uno in this case. The following image of Figure 10.11 shows the screenshot for selecting hardware.



Fig. 10.11: Selecting the hardware

Step 11. Click the email button to send a token to the email address for registration. This token will be used in `auth[] = "your token"` in `pollution.ino` file. The following Figure 10.12 shows the token generation screen of the Blynk:

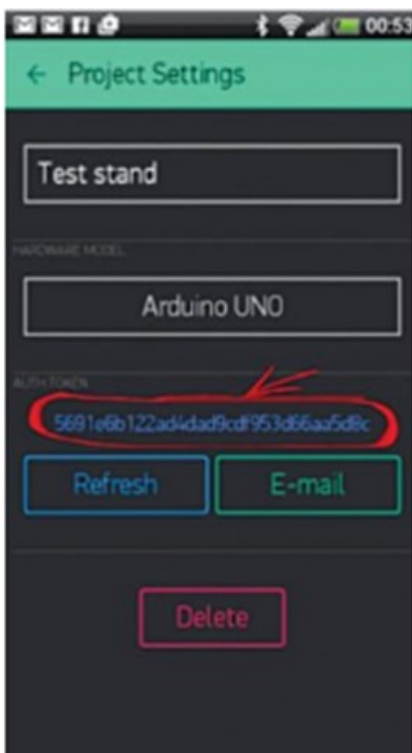


Fig. 10.12: Typical authorisation token

Step 12. Download the Blynk library (Blynk_v0.3.4.zip) and include the library in Arduino IDE. Then compile and upload `pollution.ino` code to Arduino board. The link for the library file is given below:

<https://github.com/blynkkk/blynk-library/releases/tag/v0.3.4>

Step 13. After completing the above steps press Create in the app. The following image in Figure 10.13 shows the screen for the same.

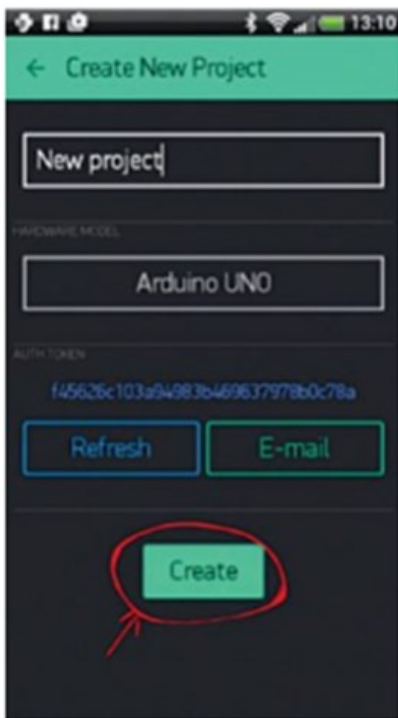


Fig. 10.13: Pressing Create button

Step 14. Tap anywhere on the screen to open the widget box and add widgets as shown in Figure 10.14.

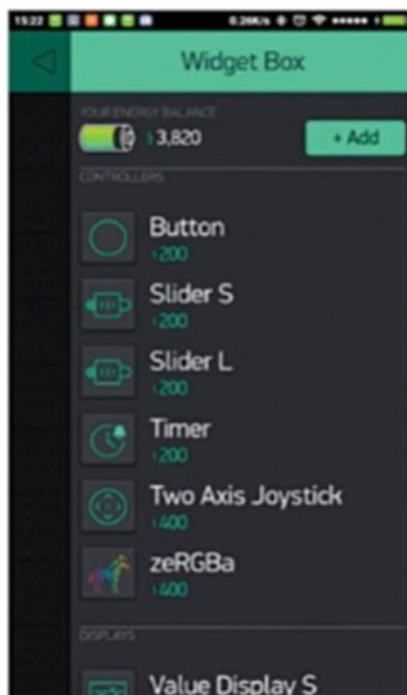


Fig. 10.14: Widget box window

Step 15. Add the LCD, LED, on/off switch, pushbuttons and RTC widgets. The Figure 10.15 shows the screen for the LCD screen widget.



Fig.10.15: LCD widget setting

Step 16. Now, run the project by pressing Play. This will switch the display from edit to play mode where interaction with the hardware can be performed. While in play mode, it is not possible to drag or set up new widgets. Press Stop and get back to edit mode. The figure 10.16 shows the play screen and play mode on the app.

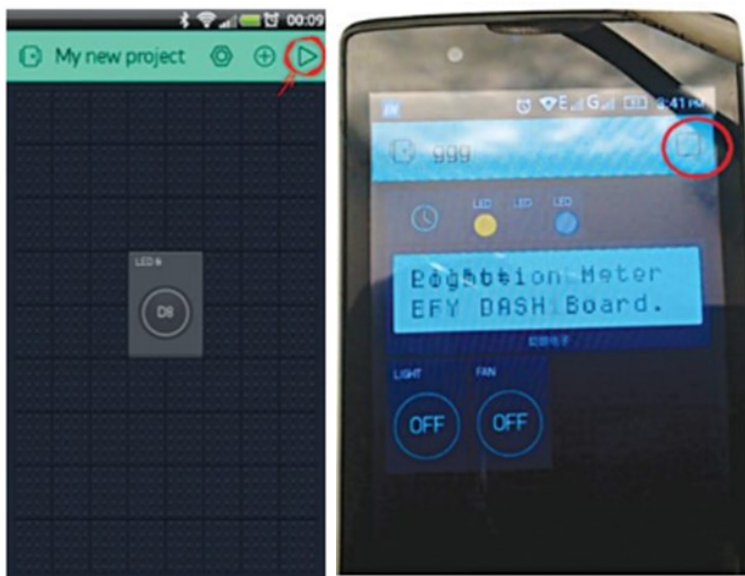


Fig. 10.16: (a) Play button (b) play mode

Practical Activity 10.1. Project to control the LEDs over the Internet using IoT applications

Hardware needed

ESP8266 Nodemcu, 4 LEDs, Jumper wires, breadboard, USB cable

Software needed

Arduino IDE with BlynkLibrary, smart phone with Blynk app

Procedure

Step 1. Connect the ESP 8266 on breadboard as shown in Figure 10.17.

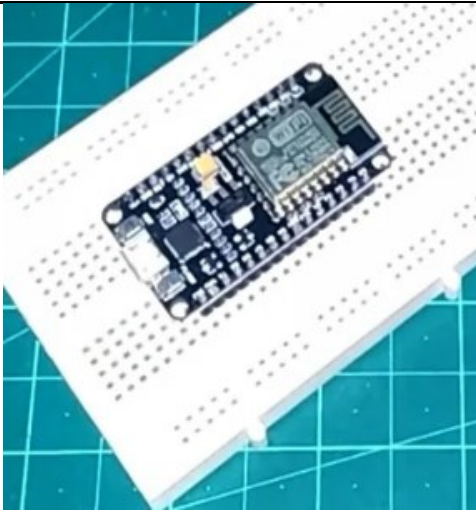


Fig. 10.17: Connection of nodemcu on breadboard

Step 2. Connect 4 LEDs to nodemcu as shown in Figure 10.18.



Fig. 10.18: connection of LEDs on breadboard

Step 3. The positive leg of the LED will be connected to a digital pin to the nodemcu.

Step 4. The Gnd pin of LED will be connected to Gnd of Nodemcu. So 4 LEDs are connected to D0, D1, D2 & D3 pin of Nodemcu as shown in Figure 10.19.

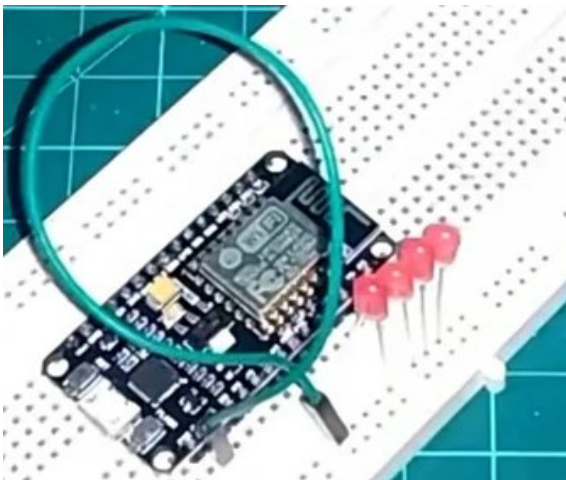


Fig. 10.19: Connection of nodemcu with LEDs

Step 5. Check for blynk in library manager of Arduino IDE and install it if it is not available.

Step 6. On the smartphone download Blynk app and login/register on the app then click on a new project. Name the Project & select your board as NodeMCU/ESP8266 and click create to create the Project and it will send a auth token to your email id which we can use later in the code as given in previous section 10.4.2.

Step 7. Then in the code go to widget section and select the button widget as here 4 buttons are selected to control 4 pins/LEDs of nodemcu. It is given in Figure 10.20(a) and (b).

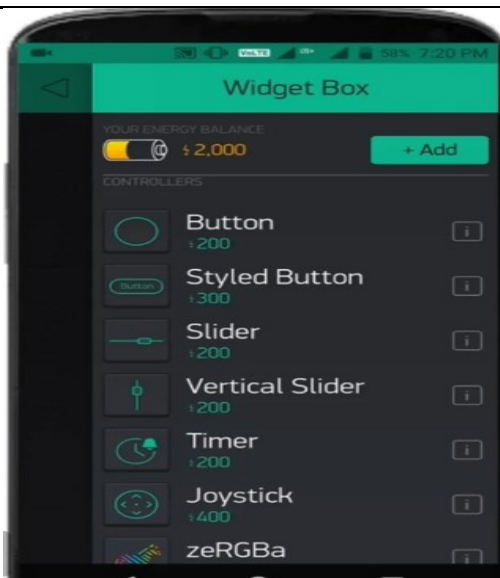


Fig. 10.20(a): widget section in Blynk app

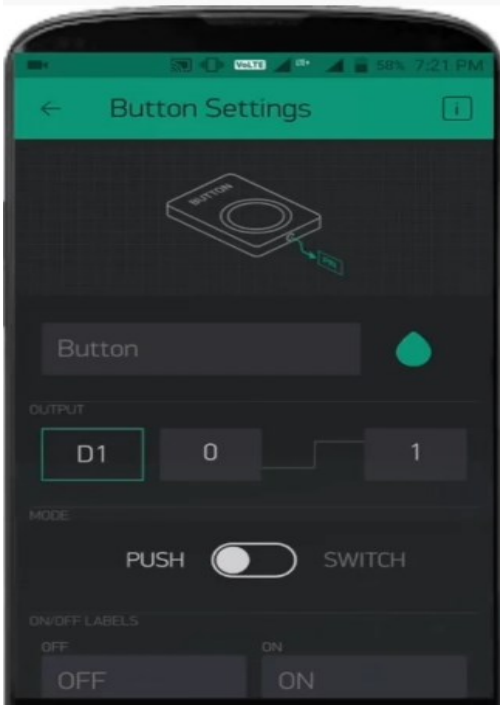


Fig. 10.20(b): widget section in Blynk app

Step 8. After selecting a button widget click on that widget and select the pin for particular button as here selected D0, D1, D2, D3 pins for all 4 buttons which means they will be used to control these 4 pins on nodemcu, as shown in Figure 10.21.



Fig. 10.21: Button settings in Blynk app

Step 9. After installing the library in Arduino IDE go to Examples > Blynk > Boards wifi > Esp8266 standalone, as shown in Figure 10.22.

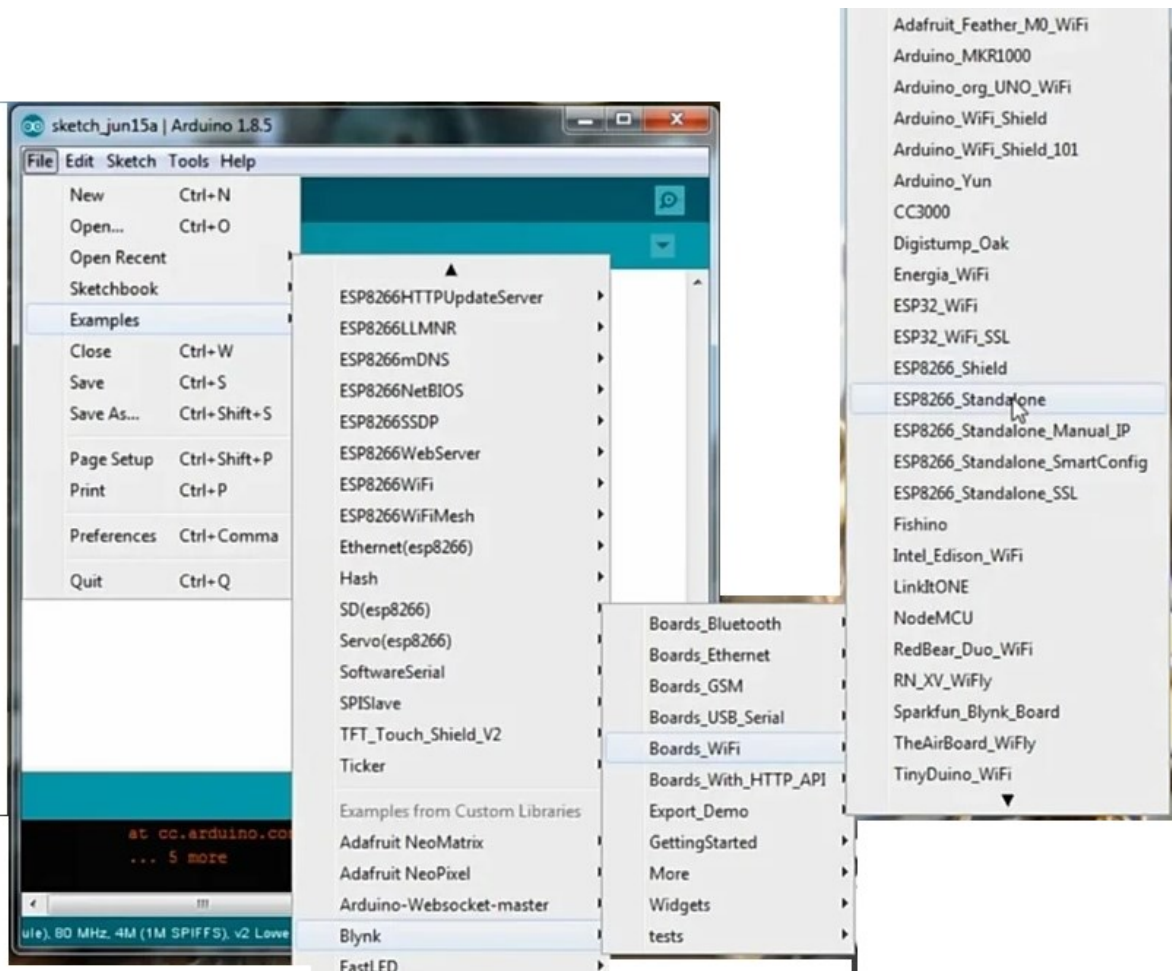


Fig. 10.22: Arduino IDE window

Step 10. Open the code Then in the code section put your auth code and put your Wi-Fi credentials (make sure your Wi-Fi has internet access) and then upload the code to your esp8266 as shown in Figure 10.23 (a), (b) and (c).

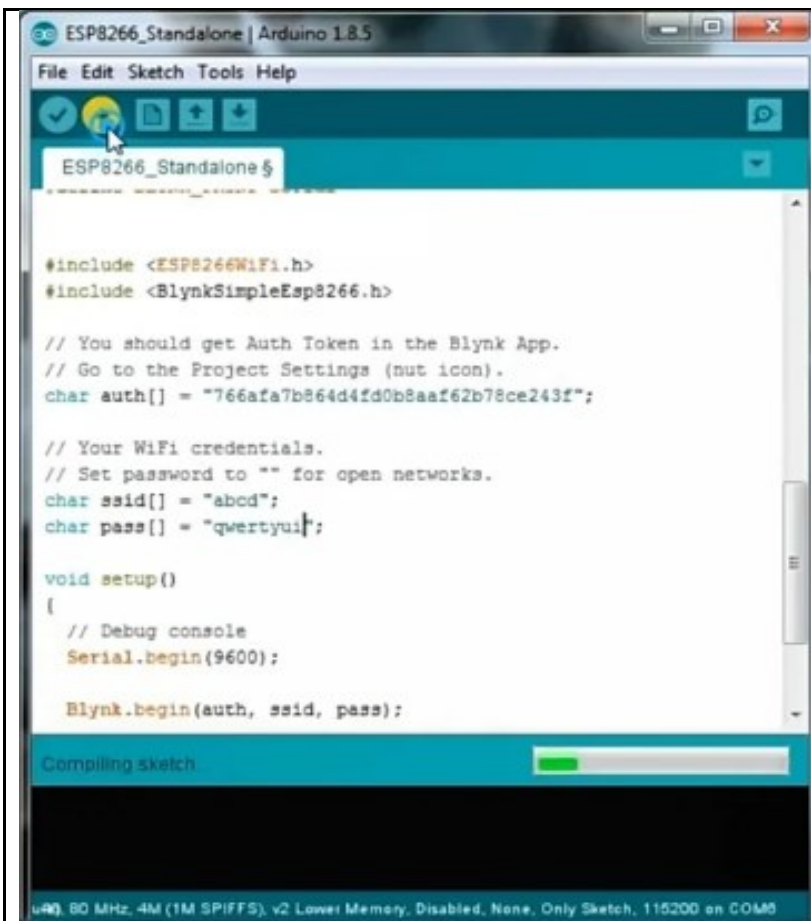


Fig. 10.23 (a): Arduino code

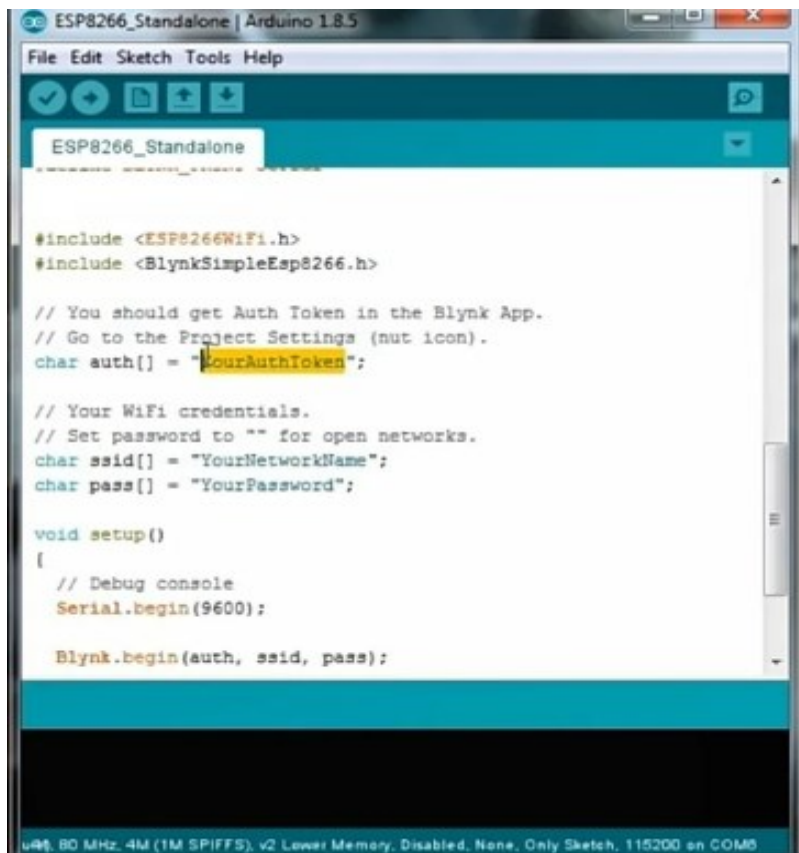


Fig. 10.23 (b): Arduino code

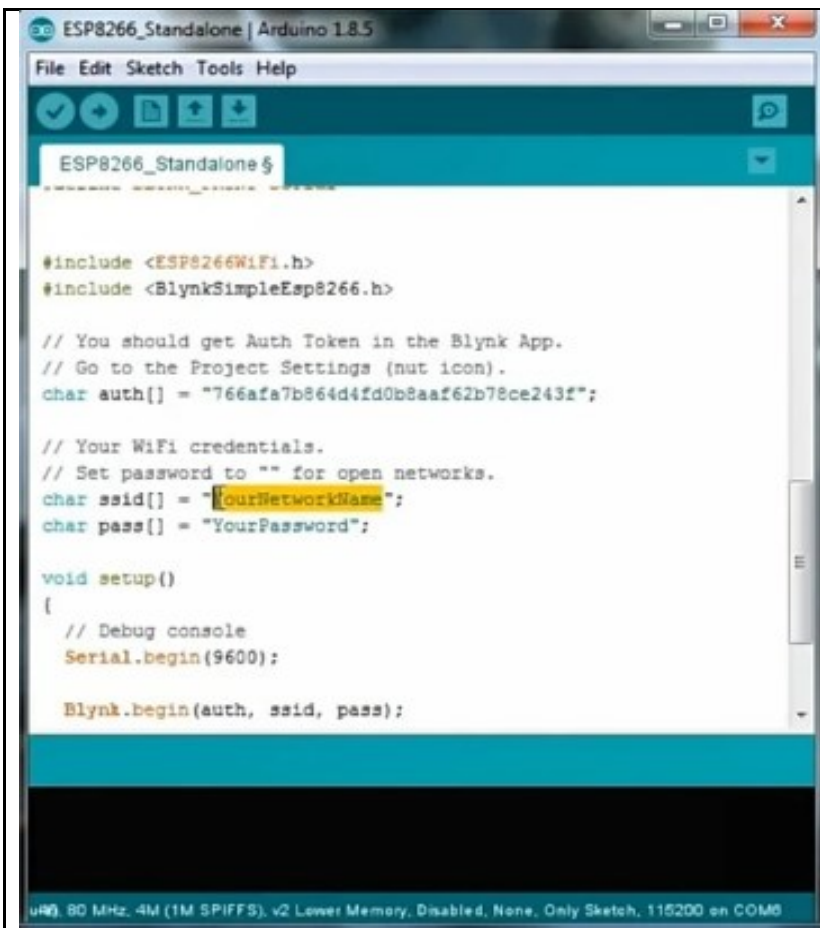


Fig. 10.23 (c): Arduino code

Step 11. Now to test it, plug the power to the Nodemcu and click on the small play button on the Blynk app as shown in Figure 10.24.

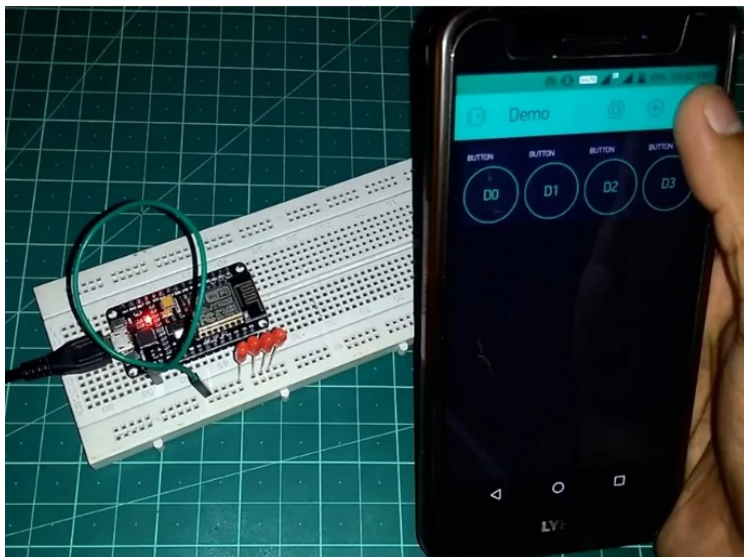


Fig. 10.24: node mcu connected with Blynk app

Step 12. Next, if we click on any of the button LED will be turned on for that particular button as here corresponding LEDs are ON for shown particular buttons. It is shown below for different buttons in Figure 10.25(a), (b) and (c).

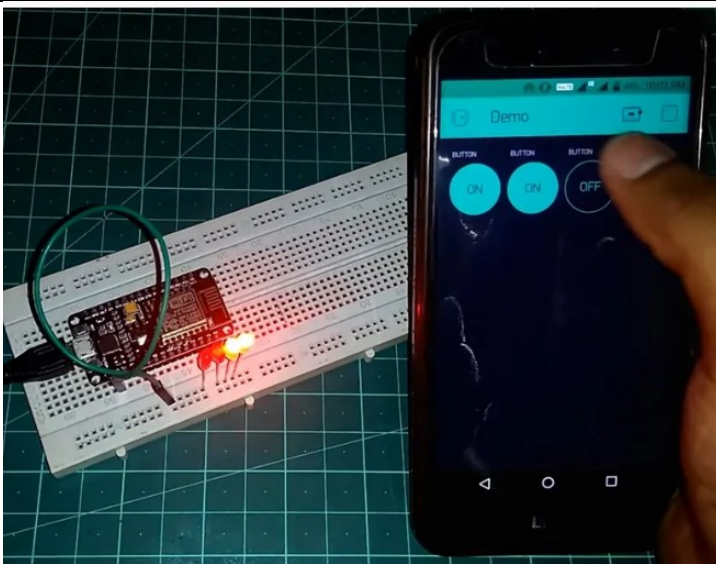


Fig. 10.25 (a): Two LEDs glow with D0 and D1 ON

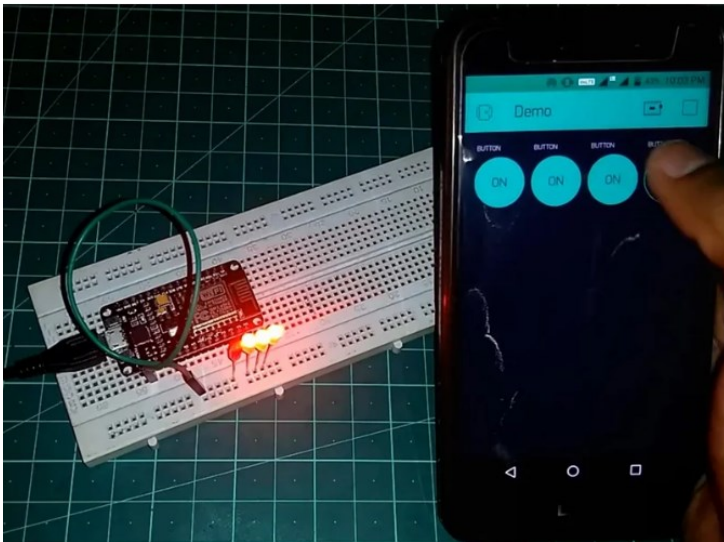


Fig. 10.25 (b): Three LEDs glow with D0, D1 and D2 ON

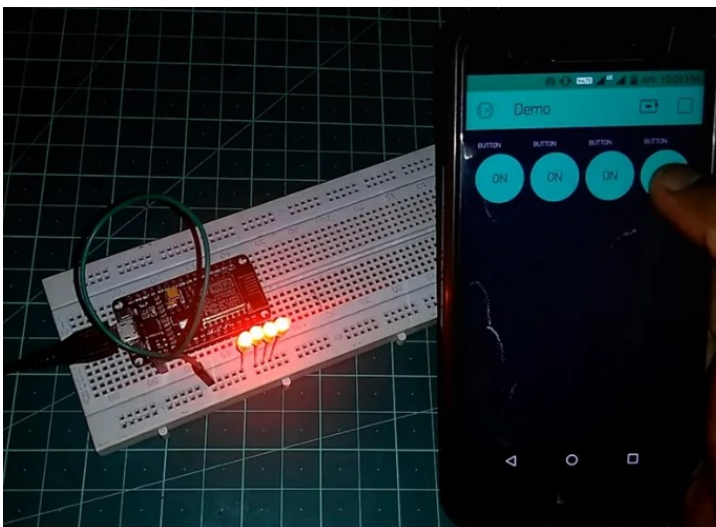


Fig. 10.25 (c): Four LEDs glow with D0, D1, D2 and D3 ON

Summary

- This chapter covers the basic knowledge of project management. It gives an overview of project handling parameters and outlines the steps in a typical project implementation process,
- It focuses on managing IoT projects in the telecom sector.
- Additionally, this chapter covers the practical implementation of IoT projects for better understanding.

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. What is emphasized as crucial for the success of IoT projects in the expanding telecom sector? (a) Technological expertise (b) Team integration (c) Financial resources (d) Marketing strategies
2. What is the key factor required for managing IoT projects according to the chapter? (a) Complex programming (b) Effective communication (c) Multifunctional units (d) Budget allocation
3. What is necessary for the implementation of IoT projects? (a) Budget allocation (b) Meticulous planning and execution (c) Rapid decision-making (d) Minimal team collaboration
4. What is the primary focus of this chapter in relation to IoT projects? (a) Integration of various work processes (b) Marketing strategies for IoT products (c) Telecom sector regulations (d) Consumer preferences in IoT
5. What role do multifunctional units play in IoT projects? (a) They handle budget allocation (b) They collaborate to create solutions (c) They oversee marketing strategies (d) They specialize in programming
6. Which of the following is NOT mentioned as a requirement for success in IoT projects? (a) Strategic decision-making (b) Team integration (c) Basic programming skills (d) Complex tool development
7. What is one of the challenges associated with IoT projects? (a) Minimal need for team collaboration (b) Simple programming requirements (c) Complexity and planning demands (d) Limited use of specialized tools
8. What is the overarching focus of IoT projects? (a) Managing financial resources (b) Ensuring regulatory compliance (c) Handling project integration and implementation (d) Promoting IoT products in the market
9. What is the primary purpose of integrating work processes into a single system through a project? (a) To reduce the need for specialized tools (b) To create a unified solution for telecom sector challenges (c) To eliminate the need for team collaboration (d) To minimize the complexity of IoT projects
10. What skills and attributes are highlighted as essential for handling IoT projects? (a) Strategic thinking, decision-making, and technical expertise (b) Budget allocation and financial planning (c) Minimal team collaboration and basic programming skills (d) Regulatory compliance and legal knowledge

B. Fill in the blanks

1. Each work process is integrated into a single system through a _____.
2. Project management is a discipline that involves _____ the successful completion of a project.
3. As a technician, one would be required to check some parameters related to project _____.
4. The success of IoT projects is contingent upon an effective team _____.
5. Multifunctional units collaborate to create a single solution to all issues in the expanding _____ sector.
6. The DHT11 sensor senses the _____ of the surrounding _____.
7. IoT projects require complex programming and _____ development, as well as strategic thinking and decision-making.
8. IoT projects are inherently complex and require meticulous planning and _____.
9. _____ allows to create amazing interfaces for various projects using various widgets.
10. Every successful project goes through the same five phases: initiation, _____, monitoring and controlling, and closing.

C. State true or False for the following

1. IoT projects require only technical expertise and do not involve strategic thinking and decision-making.
2. The success of IoT projects is independent of effective team integration.

3. IoT projects in the expanding telecom sector do not demand complex programming or tool development.
4. Multifunctional units in IoT projects do not collaborate to create a unified solution for challenges in the telecom sector.
5. The integration of various work processes into a single system is not crucial.
6. IoT projects do not require careful planning and execution.
7. Strategic thinking, decision-making, and technical expertise are essential skills for handling IoT.
8. Complex programming, tool development, and strategic decision-making are essential for the success of IoT projects.
9. The integration of various work processes is achieved through a project.
10. As a technician, it is required to check some parameters related to project handling is crucial to the success of IoT projects.

D. Short Answer Type Questions

1. Explain why team integration is emphasized as crucial for the success of IoT projects in the expanding telecom sector.
2. What are the key requirements for managing IoT projects effectively?
3. Describe the parameters related to project handling as mentioned in Section 10.2.
4. Why do IoT projects are inherently complex and require meticulous planning and execution?
5. Briefly explain the significance of the project implementation process.
6. How do multifunctional units collaborate to address challenges in the expanding telecom sector?
7. What role does a project play in integrating various work processes into a single system?
8. Provide examples of the skills and attributes highlighted as essential for handling IoT projects.
9. Why is effective team integration considered crucial for the success of IoT projects?
10. What does this chapter suggest about the complexity of programming and tool development required for IoT projects in the telecom sector?

Chapter 11. Record Maintenance and Documentation

IoT is all about integrating hardware, software, infrastructure, and network. This information is stored in a database or in a cloud platform. Most of the information that is generally stored by a technician includes plans, client data, tools, and strategies. It is important for a record system to be simple, consistent, and easily available to make it more efficient.

11.1 Record Maintenance

11.1.1 Storing of Records

Record storing is a crucial part of record maintenance. There are two types of methods used for storing records- electronic method and manual method. Technicians can use any method according to availability and requirement. These methods can be compared as shown in the following table 11.1.

Table 11.1: Methods to store records

<i>Electronic Method</i>	<i>Manual Method</i>
It is a method which uses computer-based techniques to store any information.	It is a traditional method.
In this method records can be stored in the format of Word files, Excel files, etc.	In this method the stationary items like paper, pen, files, etc. are used to store the information.
It is a very simple, effective and smart method of storing records.	It is cost effective.
It can be stored in soft and hard copies. There are very less chances of losses of records in this method.	This is less secure as it is prone to losses of records.

11.1.2 Importance of Storing Records

Maintaining correct records plays a significant role in the following ways:

- To schedule the work and make service-related appointments
- To store the operational process, product and technology
- To provide reference to previous work to curb possible errors in the future
- To check the billing and other costs involved during any project
- To check the profit earned by any particular project Handbook
- To reduce complications occurring at the time of fixing
- To identify and rectify the drawbacks
- To remove confusion during work by providing reference material
- To keep business and the process on track

11.1.3 Recording Performance

Each organization should have a proper communication system with the clients and the technicians to resolve any errors and to improve the quality of the service. There are tools available to alert technicians in case of an error. After the delivery of the service, as a technician, it is important to schedule a maintenance and performance check. This helps in preventing the future occurrence of errors. At times, some errors at the technician's end may cause a monetary loss to the client. Such errors should be avoided to uphold the client's trust and the quality of the service.

11.1.4 Methods to Record Performance

A technician should follow certain methods to record the performance of service. These can be listed as below:

Make Use of Real-time Data – Real-time sensors can be used at the client's premises to monitor them constantly. An updated data feed from sensors can be used to check the quality and performance of the service. A presentation or graphic representation from the data feed can be used, if required.

Make Use of Automated Alerts – Alert systems can be used for checking the operational conditions of services that need significant attention, for example, Internet downtime and temperature change. It must be ensured that

the alert system is connected to the mobile of the person who will need an alert in case of an error in any operational conditions. Technicians should reach the client's site in case of any operational problem, preferably on an immediate basis.

Ensure Automated Ticket – Technician can set a system or tool that can create and send the tickets automatically. Ticket system eliminates middle men which requires cost and time. Technicians must make sure that the ticket system is easily accessible to clients.

Forecast the Problems or Errors – Technician must forecast the problems and solve them on a regular basis to ensure reduction of errors and downtime in the very first place. Efficient Tools like proximity manager can be used to handle the errors. These tools often suggest possible ways to resolve a problem by analysing past history and help in fetching the solution.

11.2 Documentation

Nowadays, laptops, computers, and mobile devices play a vital role in documentation. So, it is mandatory for every technician to be proficient in operating smart phones and computers. Technicians should document the following information:

- Service request from client
- Strategy used and followed
- First visit date
- Product used and their cost
- Errors
- Services
- Bills
- Service operation visits

11.2.1 Documentation Form

Tools are available to capture the above information and document them automatically. Any document should have information about the work process, the tools used, the involved technician, and the client. The general format used in documentation of the project is given below in table 11.2.

Table 11.2: Documentation Format

Documentation Form	
Client's name	
Project plan and strategy	
Tools and product information	
Duration of the project	
Start date	
Technician information	
Work process information	
Test and feedback information	
Error report	
Customer service and query information	

11.2.2 Document Processing

Document processing is a procedure that is used to convert all information into digital form. The flow of processing a document as shown in Figure 11.1.

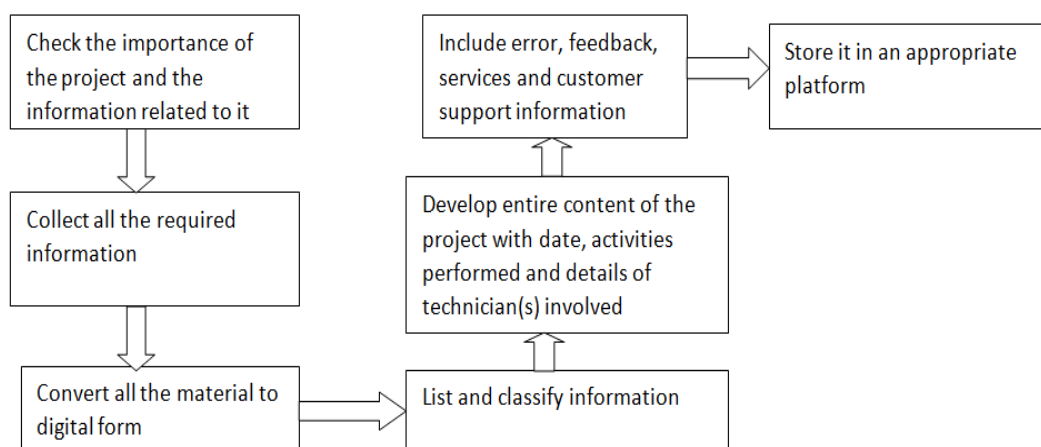


Fig. 11.1: Document processing

Tools and devices that can be used in document processing are:

- Laptop or Desktop
- Scanner
- Software Tools

11.3 Role of a Telecom Technician (IoT Devices and systems) in Record Maintenance and documentation

Advanced technologies, smarter infrastructure, and automation tools have increased the need for IoT services in all organizations. If an organization has derived work strategy, development, and technologies, there is a need for IoT solutions to enable their work plans and operating models. IoT services can help in designing a work plan for solution development and delivery. IoT can create a cloud platform that can connect all the operations in an organization including infrastructure, hardware, and software tools. Records are maintained in IoT services by technicians on high-security systems. So, technicians must be skilled to maintain such records.

The Role of a Telecom Technician (IoT Devices and systems) in Record maintenance and documentation can be described as follows:

- Installation of efficient and high-quality devices.
- Scheduled visits to a client site to check the performance of the device(s).
- Aware about the use of automation tools, alarm systems in case of emergency.
- Knowledge of all the tools and sources related to maintenance
- Able to make decisions appropriate for maintaining and processing records
- Awareness about legal and privacy statements
- Continuous tracking of all the day-to-day activities
- Able to protect the documents and be aware of the possible threats
- Efficient to furnish the required records
- Knowledge of classification between important and unimportant information

Practical Activity 11.1. Create a sample of documentation format for IoT device installation.

Sample of documentation form is given as follows:

Documentation Form	
Client's name	XXXXX

Project plan and strategy	Install IoT temperature monitoring devices in the manufacturing plant to enhance process efficiency and product quality. Strategy – Deploy temperature sensors at key production stages and integrate them with the existing manufacturing control system.
Tools and product information	Tools – Screwdrivers, drill, Ethernet cables, laptop for configuration. Product – Temperature Sensors (Model: TS-200), Quantity: 20 units.
Duration of the project	2 weeks
Start date	xxxxxx
Technician information	xxxxxx
Work process information	Week 1 • Site survey and sensor placement planning. • Configuration of sensors and integration with the control system. Week 2 • Physical installation of sensors at designated locations. • Testing of sensor functionality and data transmission. • Final integration and system calibration.
Test and feedback information	Testing – Conducted daily temperature checks using manual and automated methods. Feedback: Client expressed satisfaction with sensor accuracy and system reliability.
Error report	Issue – Connectivity problem with one sensor at the packaging area. Resolution – Reconfigured network settings and replaced Ethernet cable, issue resolved.
Customer service and query information	Customer Service Contact: xxxxx Queries – Client requested additional training on system operation,

Summary

- This chapter emphasizes the essential role of managing information in IoT services.
- It highlights the integration of hardware, software, and network data, emphasizing the need for accessible record systems.
- The chapter introduces electronic and manual record storage methods and underscores the significance of recording performance for service quality and client trust.

CHECK YOUR PROGRESS

A. Multiple Choice Questions

1. What is the primary focus of this chapter (a) Integrating hardware and software in IoT services (b) Managing and documenting information in IoT services (c) Analysing performance data in IoT services (d) Designing cloud platforms for IoT services
2. According to the chapter, where is information in IoT services typically stored? (a) Cloud platforms (b) Paper documents (c) Filing cabinets (d) Local hard drives
3. What are the two methods mentioned for storing records in IoT services? (a) Electronic and manual (b) Paper and digital (c) Cloud-based and on-premise (d) Visual and audio

4. What is the importance of recording performance, as highlighted in Chapter 11? (a) To increase costs in IoT projects (b) To maintain client trust and enhance service quality (c) To decrease the role of technicians in IoT services (d) To eliminate the need for documentation
5. What tasks are mentioned as part of the role of a technician in IoT services? (a) Managing financial records (b) Installing devices and conducting performance checks (c) Conducting market research (d) Developing software for IoT applications
6. What is the primary role of a technician in IoT services? (a) Managing financial transactions (b) Installing and checking device performance (c) Conducting marketing campaigns (d) Drafting legal documents
7. What are the qualities mentioned for a person appointed to maintain high-security records in IoT services? (a) Proficiency in cooking and gardening (b) Understanding of maintenance tools and legal awareness (c) Expertise in fashion design and photography (d) Knowledge of art history and literature
8. What is the purpose of using automated alerts in IoT services? (a) To monitor the weather conditions (b) To check for internet connectivity (c) To promptly identify and address operational issues (d) To receive news updates
9. What types of information should technicians document in IoT services? (a) Personal hobbies and interests (b) Service requests, errors, product details, and customer interactions (c) Favourite movies and TV shows (d) Sports preferences and achievements
10. Why is it important to use real-time data in IoT services? (a) To predict the stock market (b) To constantly monitor and assess operational conditions (c) To track wildlife migration patterns (d) To plan a vacation itinerary

B. Fill in the blanks

1. It highlights the integration of _____ emphasizing the need for accessible record systems.
2. There are two methods for storing records: electronic and _____.
3. Each organization should have a proper communication system with the clients and the technicians to resolve any errors and to _____.
4. The technician's role in IoT services includes tasks like device installation, conducting _____ checks, and utilizing automation tools.
5. Proficiency in mobile devices and computers is vital for effective _____.
6. Key information such as service requests, strategies, product details, errors, bills, and customer interactions should be documented in a structured _____.
7. IoT can create a cloud platform that can connect all the operations in an organization including infrastructure.
8. Records are maintained in IoT services by _____ on high-security systems.
9. Any document should have information about the _____ and the client.
10. Document processing is a procedure used to convert all information into _____.

C. State true or False for the following

1. Information in IoT services is typically stored only in physical filing cabinets.
2. There are two methods for storing records: electronic and manual.
3. Recording performance is not emphasized as important for service quality and client trust in IoT services, according to the chapter.
4. The technician's role in IoT services includes tasks like device installation, conducting performance checks, and utilizing automation tools.
5. Proficiency in mobile devices and computers is not considered important for effective documentation in IoT services.
6. Key information like service requests, strategies, and errors should not be documented in a structured format.
7. Document processing involves converting information into digital form, which can be done using tools like laptops, scanners, and software tools.

8. The person appointed to maintain high-security records in IoT services does not need to understand maintenance tools and legal awareness.
9. The general format used in documentation does not include information about the client's name, project plan, and strategy, as mentioned in the chapter.
10. There is no need to structure information in documentation for IoT services.

D. Short Answer Type Questions

1. What is the primary focus of this chapter?
2. What are the two methods mentioned for storing records in IoT services?
3. Why is recording performance emphasized as important in IoT services?
4. What are some key tasks included in the role of a technician in IoT services?
5. Why is proficiency in mobile devices and computers considered crucial for effective documentation in IoT services?
6. What types of information should technicians document in IoT services, as outlined in the chapter?
7. What is the purpose of document processing?
8. What qualities should a person appointed for high-security record maintenance possess?
9. What information is typically included in the general format used for documentation in IoT services?
10. What tools and devices are recommended for document processing, as suggested in the chapter?