CMR
INSTITUTE OF
TECHNOLOGY

USN | | | | | | | | | | |

Internal Assessment Test 2– Feb 2023

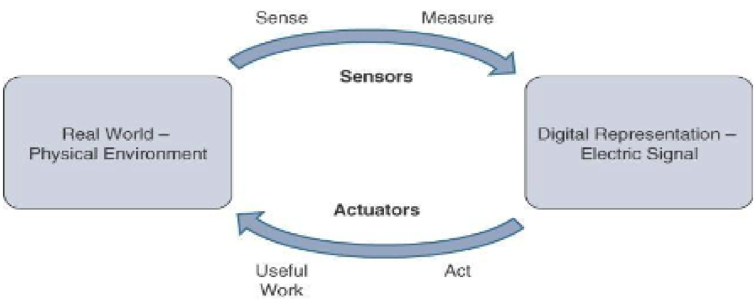
Sub:	Internet of Things					Sub Code:	20MCA32	Branch:	MCA
Date:	06\02\2023	Duration:	90 min's	Max Marks:	50	Sem	III		

Q1. Define sensors and actuators; explain how actuators and sensors interact with physical world with the neat diagram. Classify actuators based on energy type.

A sensor measures some physical quantity and converts that measurement reading into a digital representation.

Sensors are designed to sense and measure practically any measurable variable in the physical world. They convert their measurements (typically analog) into electric signals or digital representations that can be consumed by an intelligent agent (a device or a human).

Actuators, on the others hand, receive some type of control signal (commonly an electric signal or digital command) that triggers a physical effect, usually some type of motion, force,



and so on.

Type	Examples
Mechanical actuators	Lever, screw jack, hand crank
Electrical actuators	Thyristor, biopolar transistor, diode
Electromechanical actuators	AC motor, DC motor, step motor
Electromagnetic actuators	Electromagnet, linear solenoid
Hydraulic and pneumatic actuators	Hydraulic cylinder, pneumatic cylinder, piston, pressure control valves, air motors
Smart material actuators (includes thermal and magnetic actuators)	Shape memory alloy (SMA), ion exchange fluid, magnetorestrictive material, bimetallic strip, piezoelectric bimorph
Micro- and nanoactuators	Electrostatic motor, microvalve, comb drive

Q2. List and explain different type of sensors with an example.

A sensor measures some physical quantity and converts that measurement reading into a digital representation.

Sensors are designed to sense and measure practically any measurable variable in the physical world. They convert their measurements (typically analog) into electric signals or digital representations that can be consumed by an intelligent agent (a device or a human).

Sensor Types	Description	Examples
Position	A position sensor measures the position of an object; the position measurement can be either in absolute terms (absolute position sensor) or in relative terms (displacement sensor). Position sensors can be linear, angular, or multi-axis.	Potentiometer, inclinometer, proximity sensor
Occupancy and motion	Occupancy sensors detect the presence of people and animals in a surveillance area, while motion sensors detect movement of people and objects. The difference between the two is that occupancy sensors generate a signal even when a person is stationary, whereas motion sensors do not.	Electric eye, radar
Velocity and acceleration	Velocity (speed of motion) sensors may be linear or angular, indicating how fast an object moves along a straight line or how fast it rotates. Acceleration sensors measure changes in velocity.	Accelerometer, gyroscope
Force	Force sensors detect whether a physical force is applied and whether the magnitude of force is beyond a threshold.	Force gauge, viscometer, tactile sensor (touch sensor)
Pressure	Pressure sensors are related to force sensors, measuring force applied by liquids or gases. Pressure is measured in terms of force per unit area.	Barometer, Bourdon gauge, piezometer
Flow	Flow sensors detect the rate of fluid flow. They measure the volume (mass flow) or rate (flow velocity) of fluid that has passed through a system in a given period of time.	Anemometer, mass flow sensor, water meter

Q3. Define smart object and explain the characteristics, Also provide the definition for SANET? Explain the advantages and disadvantages of it.

Smart Objects

Smart objects are, quite simply, the building blocks of IoT. They are what transform everyday objects into a network of intelligent objects that are able to learn from and interact with their environment in a meaningful way. A *smart object*, is a device that has, at a minimum, the following four defining characteristics

- **Processing Unit:** A smart object has some type of processing unit for acquiring data, processing and analyzing sensing information received by the sensor(s), coordinating control signals to any actuators, and controlling a variety of functions on the smart object, including the communication and power systems.
- **Sensor(s) and /or actuator(s):** A smart object is capable of interacting with the physical world through sensors and actuators. A smart object does not need to contain

  both sensors and actuators. In fact, a smart object can contain one or multiple sensors and/or actuators, depending upon the application.
- **Communication Device:** The communication unit is responsible for connecting a smart object with other smart objects and the outside world (via the network). Communication devices for smart objects can be either wired or wireless.
- **Power Source:** Smart objects have components that need to be powered. Interestingly, the most significant power consumption usually comes from the communication unit of a smart object.

Sensor Networks:

- A sensor/actuator network (SANET), as the name suggests, is a network of sensors that sense and measure their environment and/or actuators that act on their environment.
- The sensors and/or actuators in a SANET are capable of communicating and cooperating in a productive manner.
- SANETs offer highly coordinated sensing and actuation capabilities.
- Smart homes are a type of SANET that display this coordination between distributed sensors and actuators.
- For example, smart homes can have temperature sensors that are strategically networked with heating, ventilation, and air-conditioning (HVAC) actuators. When a sensor detects a specified temperature, this can trigger an actuator to take action and heat or cool the home as needed.

The following are some advantages and disadvantages that a wireless-based solution offers:

Advantages:

- Greater deployment flexibility (especially in extreme environments or hard-to-reach places)
- Simpler scaling to a large number of nodes
- Lower implementation costs
- Easier long-term maintenance
- Effortless introduction of new sensor/actuator nodes
- Better equipped to handle dynamic/rapid topology changes

Disadvantages:

- Potentially less secure (for example, hijacked access points)
- Typically, lower transmission speeds
- Greater level of impact/influence by environment

Q4. List and explain any 15 sensors present in smart phones with its purpose

.

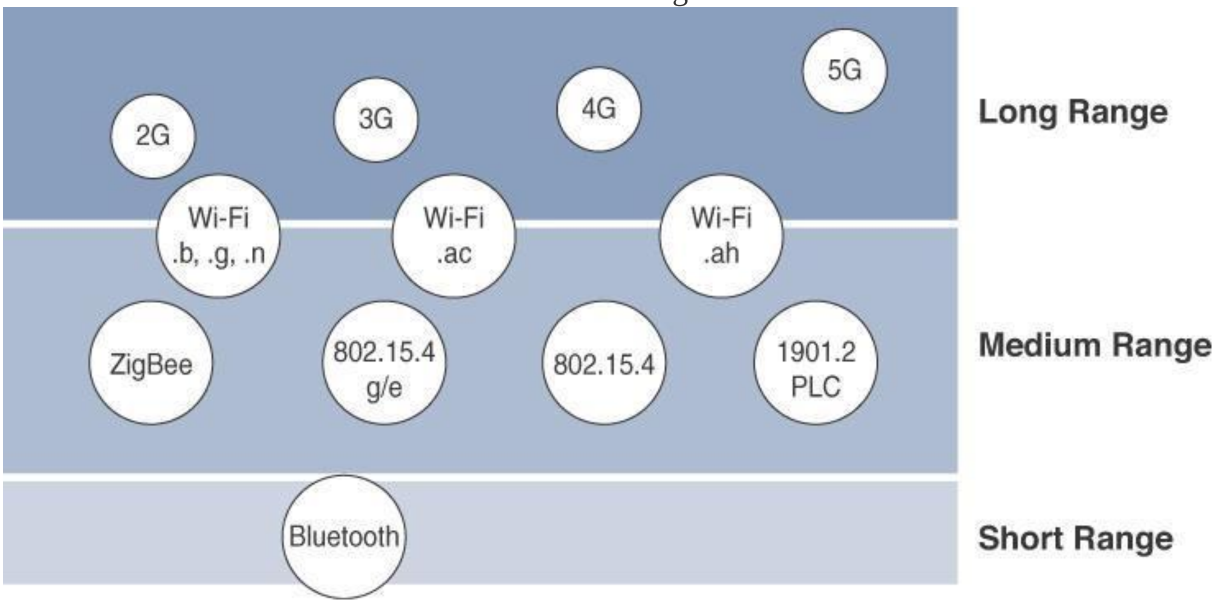
Explain any 15 from the following

Figure 3-2 *Sensors in a Smart Phone*

Q5) Explain the communication criteria's for connecting smart objects in detail

The characteristics and attributes considered when selecting and dealing with connecting smart objects are

1. **Range:** It defines how far does the signal need to be propagated? That is, what will be the area of coverage for a selected wireless technology? The below figure 2.4 shows the range considered



- **Short Range:**
  - The classical wired example is a serial cable.
  - Wireless short-range technologies are often considered as an alternative to a serial cable, supporting tens of meters of maximum distance between two devices.
  - Examples of short-range wireless technologies are IEEE 802.15.1 Bluetooth and IEEE 802.15.7 Visible Light Communications (VLC).
  - These short-range communication methods are found in only a minority of IoT installations.
- **Medium Range:**

    - In the range of tens to hundreds of meters, many specifications and implementations are available.
    - The maximum distance is generally less than 1 mile between two devices.
- **Long Range**:
    - Distances greater than 1 mile between two devices require long-range technologies. Wireless examples are cellular (2G, 3G, 4G) and some applications of outdoor IEEE 802.11 Wi-Fi and Low-Power Wide-Area (LPWA) technologies.
    - LPWA communications have the ability to communicate over a large area without consuming much power.
    - These technologies are therefore ideal for battery-powered IoT sensors.
    - Found mainly in industrial networks, IEEE 802.3 over optical fiber and IEEE 1901 Broadband Power Line Communications are classified as long range but are not really considered IoT access technologies.

2. **Frequency Bands:**

- Radio spectrum is regulated by countries and/or organizations, such as the International Telecommunication Union (ITU) and the Federal Communications Commission (FCC).
- These groups define the regulations and transmission requirements for various frequency bands.
- For example, portions of the spectrum are allocated to types of telecommunications such as radio, television, military, and so on.
- Focusing on IoT access technologies, the frequency bands leveraged by wireless communications are split between licensed and unlicensed bands.
- Licensed spectrum is generally applicable to IoT long-range access technologies and allocated to communications infrastructures deployed by services providers, public services (for example, first responders, military), broadcasters, and utilities.
- The ITU has also defined unlicensed spectrum for the industrial, scientific, and medical (ISM) portions of the radio bands.
- These frequencies are used in many communications technologies for short-range devices (SRDs).
- Unlicensed means that no guarantees or protections are offered in the ISM bands for device communications.

Power Consumption:

- Battery-powered nodes bring much more flexibility to IoT devices.
- These nodes are often classified by the required lifetimes of their batteries.
- A powered node has a direct connection to a power source, and communications are usually not limited by power consumption criteria.
- IoT wireless access technologies must address the needs of low power consumption and connectivity for battery-powered nodes.
- This has led to the evolution of a new wireless environment known as Low-Power Wide-Area (LPWA).

Topology

- Among the access technologies available for connecting IoT devices, three main topology schemes are dominant: star, mesh, and peer-to-peer.
- For long-range and short-range technologies, a star topology is prevalent, as seen with cellular, LPWA, and Bluetooth networks.
- Star topologies utilize a single central base station or controller to allow communications with endpoints.
- For medium-range technologies, a star, peer-to-peer, or mesh topology is common.
- Peer-to-peer topologies allow any device to communicate with any other device as long as they are in range of each other.

- Peer-to-peer topologies enable more complex formations, such as a mesh networking topology.

Constrained Devices:

Constrained nodes have limited resources that impact their networking feature set and capabilities.

Class	Definition
Class 0	This class of nodes is severely constrained, with less than 10 KB of memory and less than 100 KB of Flash processing and storage capability. These nodes are typically battery powered. They do not have the resources required to directly implement an IP stack and associated security mechanisms. An example of a Class 0 node is a push button that sends 1 byte of information when changing its status. This class is particularly well suited to leveraging new unlicensed LPWA wireless technology.
Class 1	While greater than Class 0, the processing and code space characteristics (approximately 10 KB RAM and approximately 100 KB Flash) of Class 1 are still lower than expected for a complete IP stack implementation. They cannot easily communicate with nodes employing a full IP stack. However, these nodes can implement an optimized stack specifically designed for constrained nodes, such as Constrained Application Protocol (CoAP). This allows Class 1 nodes to engage in meaningful conversations with the network without the help of a gateway, and provides support for the necessary security functions. Environmental sensors are an example of Class 1 nodes.
Class 2	Class 2 nodes are characterized by running full implementations of an IP stack on embedded devices. They contain more than 50 KB of memory and 250 KB of Flash, so they can be fully integrated in IP networks. A smart power meter is an example of a Class 2 node.

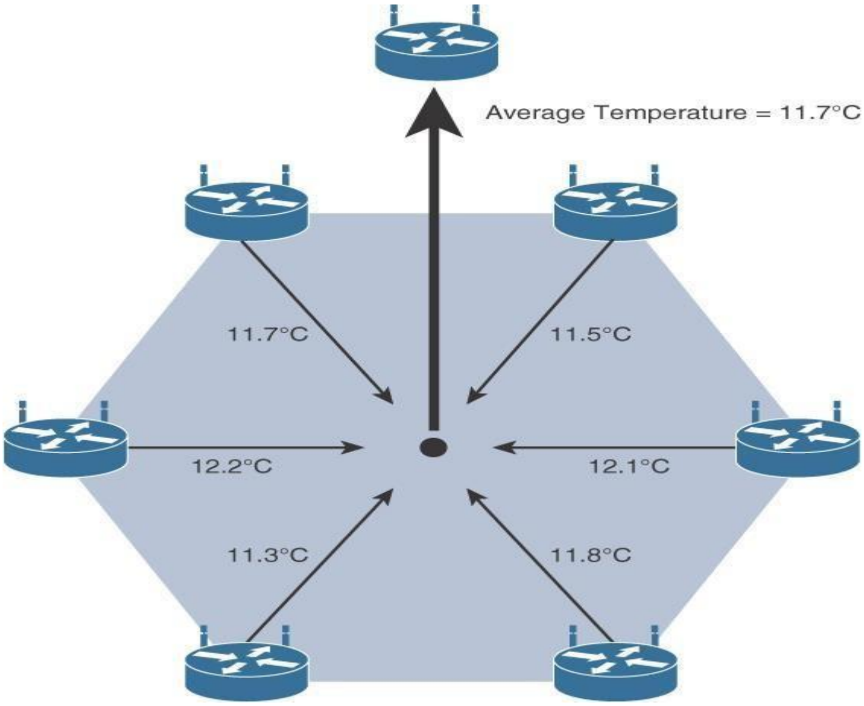
Q6) List out the limitations of the smart objects in WSNs and explain the data aggregation in WSN with a neat diagram.

Wireless Sensor Networks (WSNs)

Wireless sensor networks are made up of wirelessly connected smart objects, which are sometimes referred to as *motes*. The following are some of the most significant limitations of the smart objects in WSNs:

- Limited processing power
- Limited memory
- Lossy communication
- Limited transmission speeds
- Limited power

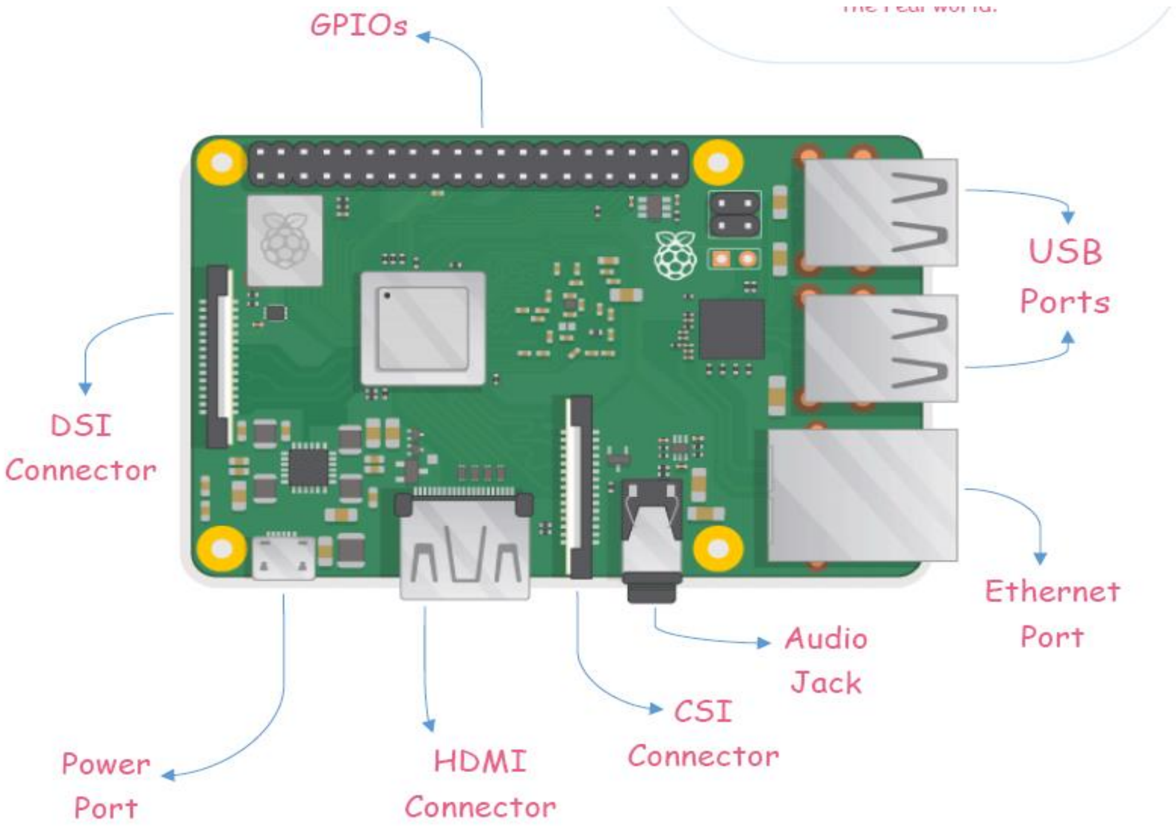
These limitations greatly influence how WSNs are designed, deployed, and utilized. Figure 2.3 below shows an example of such a data aggregation function in a WSN where temperature readings from a logical grouping of temperature sensors are aggregated as an average temperature reading.



These data aggregation techniques are helpful in reducing the amount of overall traffic (and energy) in WSNs with very large numbers of deployed smart objects. Wirelessly connected smart objects generally have one of the following two communication patterns:

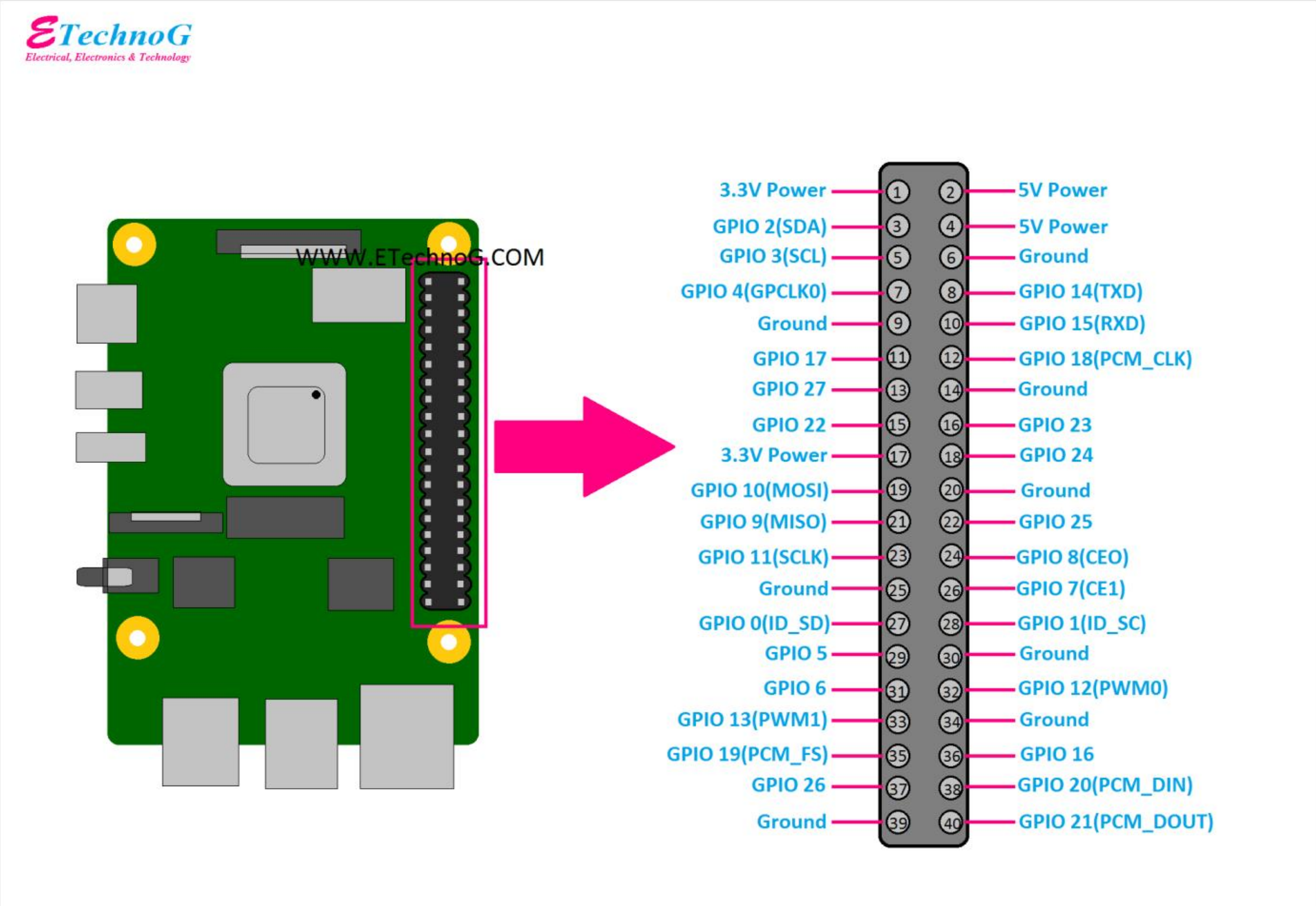
- **Event-driven:** Transmission of sensory information is triggered only when a smart object detects a particular event or predetermined threshold.
- **Periodic:** Transmission of sensory information occurs only at periodic intervals.

Q7) With a neat diagram, Explain Raspberry Pi learning board with pin configuration



- GPIOs- Connect devices to interact with the real world, for instance, sensors, LEDs, Motors etc.

➢ USB Port- to connect a mouse, a keyboard or other peripherals.
➢ Ethernet Port- to connect to the internet using an Ethernet cable.
➢ Audio Jack- to connect an audio device.
➢ CSI Connector- to connect a camera with a CSI(Camera Serial Interface) ribbon.
➢ HDMI Connector- to connect a monitor or TV.
➢ Power Port- to power up your Pi.
➢ DSI Connector- to connect DSI compatible Display.



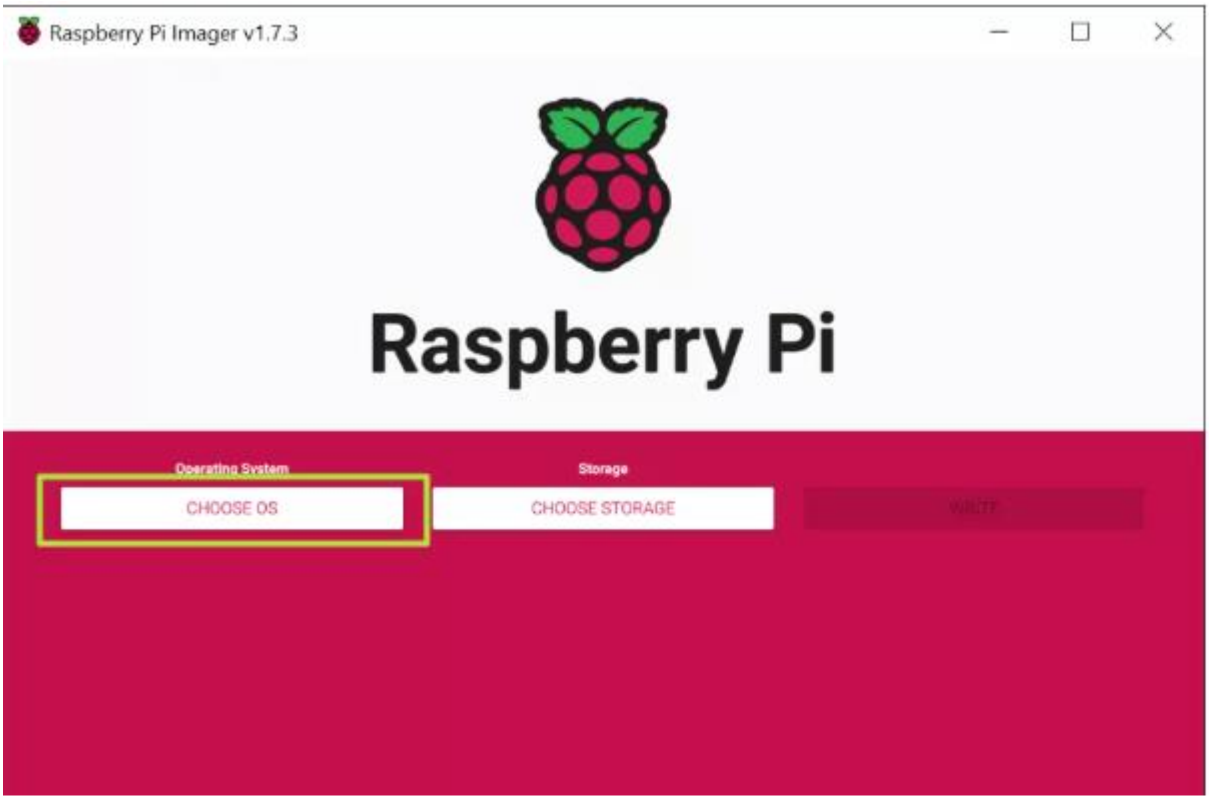
Raspberry Pi 4 Pinout Diagram

Q8) Write a note on i.) OS configuration on raspberry Pi ii.)Architecture of smart city

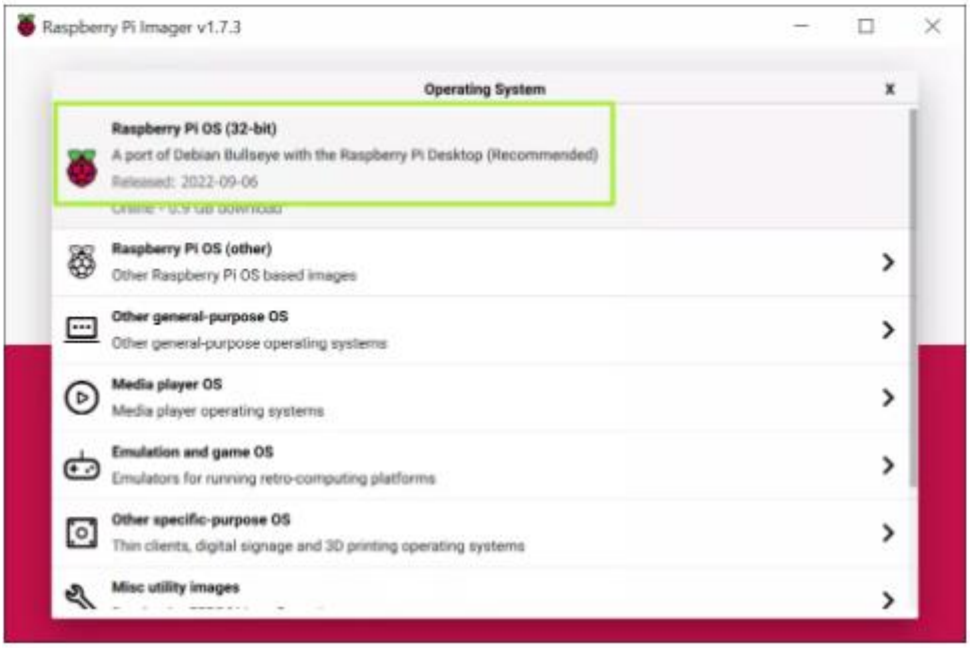
i. OS configuration on raspberry Pi

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

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



4. **Select Raspberry Pi OS (32-bit)** from the OS menu (there are other choices, but for most uses, 32-bit is the best).



(Image credit: Tom's Hardware)

4. **Click Choose storage** and **pick the SD card** you're using.

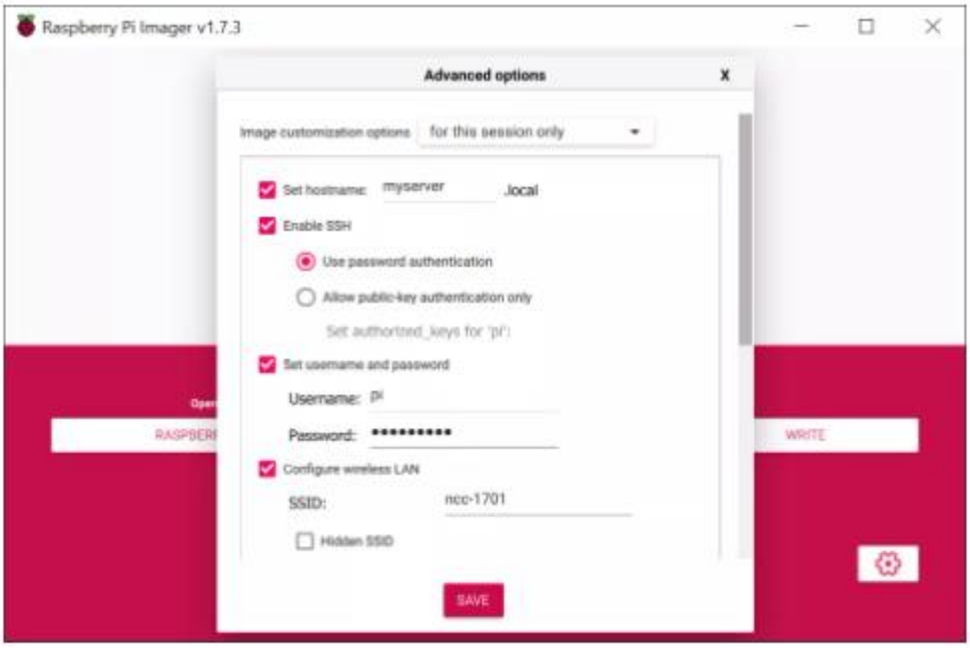
(Image credit: Tom's Hardware)

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

(Image credit: Tom's Hardware)

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

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



(Image credit: Tom's Hardware)

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

Booting Your Raspberry Pi for the First Time

After you're done writing the Raspberry Pi OS to a microSD card, it's time for the moment of truth.

1. **Insert the microSD card** into the Raspberry Pi.

2. **Connect the Raspberry Pi** to a monitor, keyboard and mouse.

3. **Connect an Ethernet cable** if you plan to use wired Internet.

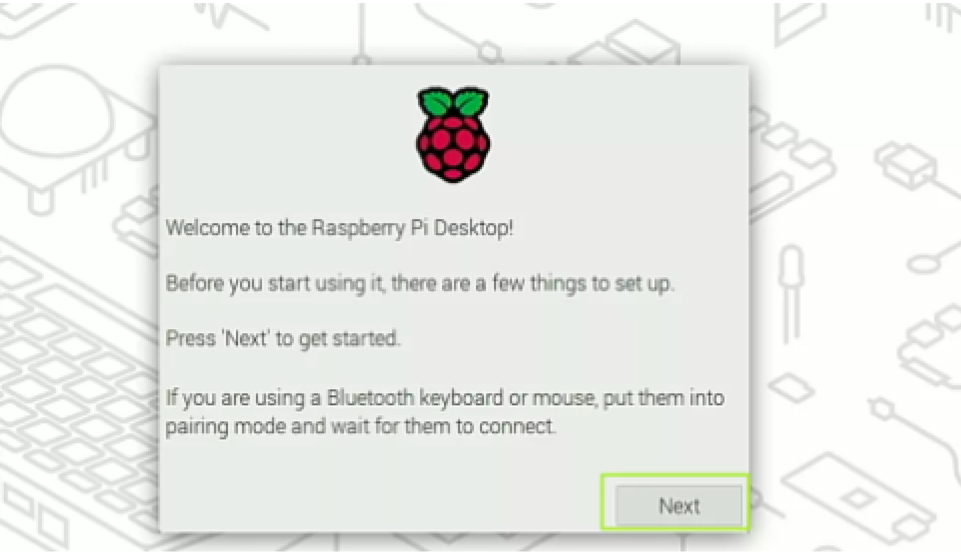
4. **Plug the Pi in** to power it on.

If you had used the Raspberry Pi Imager settings to create a username and password, you'll be able to go straight into the desktop environment, but if not, you will get a setup wizard.

Using the Raspberry Pi First-Time Setup WIzard

If you chose a username and password in Raspberry Pi Imager settings, before writing the microSD card, you will get the desktop on first boot. But, if you did not, you'll be prompted to create a username and password and enter all the network credentials by a setup wizard on first boot. If that happens, follow these steps to finish setting up your Raspberry Pi.

1. **Click Next** on the dialog box.



ii.)Architecture of smart city

What Is a Smart City?

There are a range of definitions for what a smart city actually is, and no single consensus on the essential criteria.

"A smart city is a municipality that uses information and communication technologies (ICT) to increase operational efficiency, share information with the public and improve both the quality of government services and citizen welfare," says the Internet of Things Agenda site.

The consultancy Deloitte notes that a "360-degree smart city looks across every aspect of a city's operations and uses technology to improve outcomes. The digital infrastructure of a smart city sets the stage for a network of partnerships all focused on one goal: Creating a smarter city. The result? An urban center that not only leverages technology to improve its own operations but connects with citizens, businesses, and nonprofits in new ways."

The firm says that quality of life, economic competitiveness and sustainability are the three key differentiators for smart cities.

Bee Smart City, a smart city network and community, believes that for a municipality to "become a truly smart city or community," they need to "advance in six key strategic action fields: Smart Government, Smart Economy, Smart Environment, Smart Living, Smart Mobility and Smart People."

Others focus on the outcomes of using an IoT architecture and connected technologies.

"Our definition of smart cities is around how you become efficient at optimizing certain technologies or operations or infrastructures," Gartner Analyst Bettina Tratz-Ryan tells TechRepublic. "How you can start to share outcome or best practices with each other and generate not just best practices but generate citizen outcome or context. The contextual services where you don't only look at a citizen but you look at a person with individual needs or business groups with very specific needs. That constitutes a smart city."

Smart City Examples Leveraging IoT Architecture

IoT architecture serves as the foundation for smart city use cases. Cities deploy an array of sensors to collect data on everything from whether a parking spot is empty to how many people are passing by a certain point.

That data, when transported through the network layer and subsequently analyzed, can inform whether an application shows a resident that the spot is empty or can lead to a streetlight dimming or becoming brighter based upon nearby pedestrian traffic.

"Municipal governments are leveraging cellular and Low Power Wide Area (LPWAN) wireless technologies to connect and improve infrastructure, efficiency, convenience, and quality of life for residents and visitors alike," infrastructure and defense conglomerate Thales notes.

For example, the company notes, "connected traffic lights receive data from sensors and cars adjusting light cadence and timing to respond to real-time traffic, reducing road congestion."

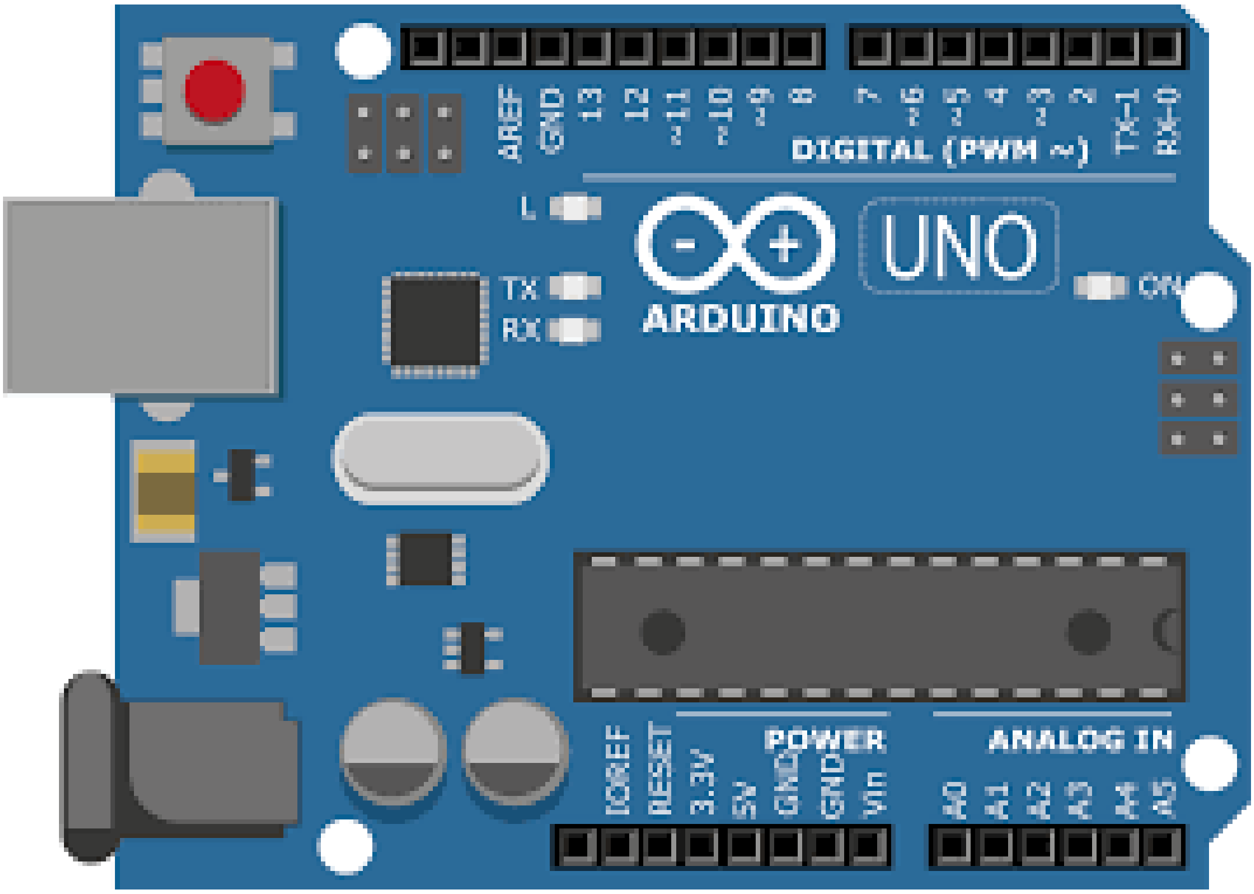
Smart parking is a visible example of the entire architecture at work, says Esmeralda Swartz, vice president of strategy, marketing and communications at network technology vendor Ericsson.

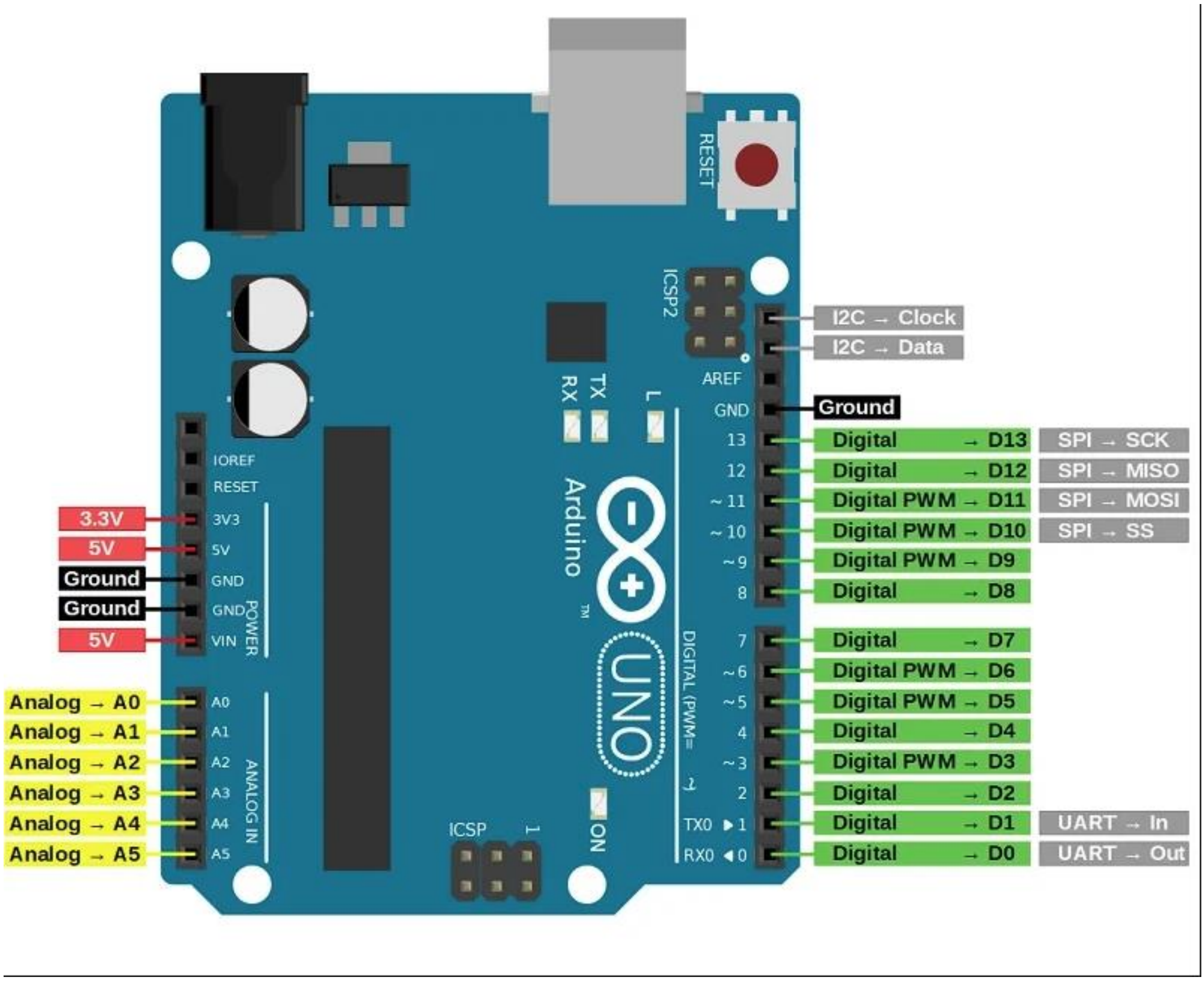
"If you think about one of the biggest drivers of traffic congestion, it is people, whether residents or visitors, driving around looking for an open parking spot," Swartz tells TechRepublic. "Now through a mobile app, through sensors that are deployed on parking spots, you know exactly where [to go] and you don't have to search around and try to find an open parking spot. It's those simple things we take for granted for improving interaction with common city services."

Another example Swartz points to is waste management. "Think about sensor technology applied to a smart waste receptacle," she says. "It knows when the trash is basically hitting the middle of the container. It compresses it down and when it gets to the top and is full, it notifies the city sanitation department that it's time to collect the trash."

Without the IoT architecture to collect, funnel and analyze data, none of these applications and their benefits would be possible.

Q9) With a neat diagram, Explain Aurdino UNO board with neat diagram





Q10) Write a note on fundamentals of Aurdino Programming

i.Structure

Sketch

- A sketch is a program written with the Arduino IDE.[57] Sketches are saved on the development computer as text files with the file extension .ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension .pde.

- A minimal Arduino C/C++ program consist of only two functions:
  - setup(): This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch.
  - loop(): After setup() function exits (ends), the loop() function is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.

ii.Functions

Digital I/O

a. pinMode()
b. digitalRead()
c. digitalWrite()

Analog I/O

- analogRead()
- analogWrite()

iii.Data types

iv.Flow control statements

```
if and if ..else Statement
if (expression)
{
statement;
}
if (expression)
{
do_this;
}
else
{
do_that;
}
```

v.Variables

A variable is a name of the memory location. It is used to store data. Its value can be changed, and it can be reused many times.
It is a way to represent memory location through symbol so that it can be easily identified.
Let's see the syntax to declare a variable:

1. type variable_list;

The example of declaring the variable is given below:

1. int a;
2. float b;
3. char c;