- Serial Clock – SCLK (GPIO11)
- Chip Enable – CE0 (GPIO8)
- Chip Enable – CE1 (GPIO7)

(ii) SPI1

- Master Out Slave In – MOSI (GPIO20)
- Master In Slave Out – MISO (GPIO19)
- Serial Clock – SCLK (GPIO21)
- Chip Enable – CE0 (GPIO18)
- Chip Enable – CE1 (GPIO17)
- Chip Enable – CE2 (GPIO16)

(c) I2C : Inter-Integrated Circuit protocol (I2C) interface pins allow you to connect hardware modules. I2C interface allows synchronous data transfer with data and clock pins.

(i) Data : (GPIO2); Clock (GPIO3)

(ii) EEPROM Data : (GPIO0); EEPROM Clock (GPIO1)

(d) Serial : The serial interface has receive (Rx) and transmit (Tx) pins for communication with serial peripherals.

(i) TX (GPIO14)

(ii) RX (GPIO15)

6.5 Introduction to Arduino

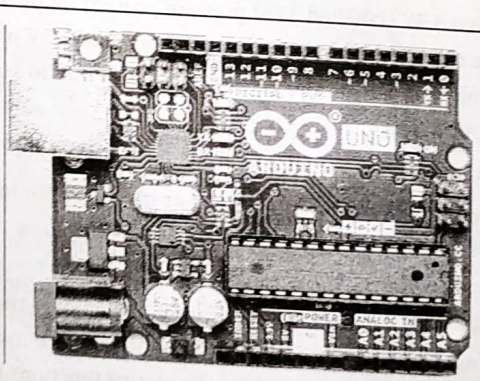


Fig. 6.5.1

- Arduino is an open-source electronics platform based on easy-to-use hardware and software. 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.
- Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

6.5.1 Interfacing of the Sensors and Actuators with Arduino

- You can interface various sensors and actuators with an Arduino board and build projects. Arduino boards could have different types of microcontrollers. The latest microcontroller is ATmega328P, which is used on most recent boards.
- The Fig. 6.5.2 shows outlines the pinout of an Arduino Uno board.

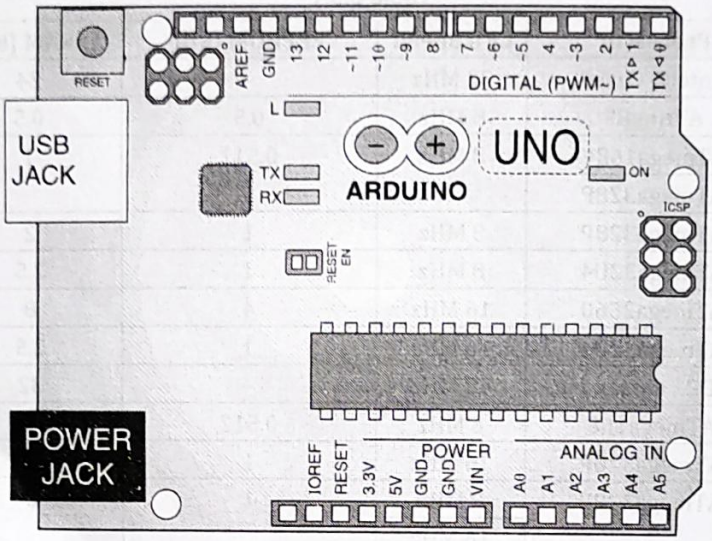


Fig. 6.5.2

The pins are as shown in the Table 6.5.2.

Table 6.5.2

Pin Name or Number	Purpose
IOREF	This provides a logic reference voltage for shields that use it. It is connected to the 5V bus.
RESET	Use to reset the Arduino board.
3.3 V	Power supply
5 V	Power supply
GND	Ground pin. In the Arduino Uno pinout, you can find 5 GND pins, which are all interconnected.
VIN	This pin is used to power the Arduino Uno board using an external power source.
A0 – A5	Analog Input Pins. The Arduino Uno has 6 analog pins, which utilize ADC (Analog to Digital converter). These pins serve as analog inputs but can also function as digital inputs or digital outputs.
AREF	Analog Reference pin
Digital Pins 2-13	Digital I/O. Pins 3,5,6,9,10,11 (marked with ~) have PWM capability. Pulse Width Modulation (PWM) is a modulation technique used to encode a message into a pulsing signal.
Digital Pins 0-1/Serial In/Out – TX/RX	These pins cannot be used for Digital I/O but for serial I/O. Serial communication is used to exchange data between the Arduino board and another serial device such as computers, displays, sensors and more. Each Arduino board has at least one serial port.

- The sensors and actuators are connected to the Arduino boards as per this pin layout.

Note : Note here that there could be a difference in the pin layout based on the board that you choose. Please read and understand the board pin layout and connection diagrams carefully before working with them else you may damage the board.

The ZigBee or ZigBee/IEEE 802.15.4 protocol is a specification created for wireless networking.

It includes hardware and software standard design for WSN (Wireless sensor network) requiring high reliability, low cost, low power, scalability and low data rate. ZigBee-style self-organizing ad-hoc digital radio networks were conceived in the 1990s, but the IEEE 802.15.4-2003/ZigBee specification was ratified on December 14, 2004. And only half year later the ZigBee Alliance announces availability of Specification 1.0 (on June 13, 2005).

A WSN consist in inexpensive wireless sensor which are capable of collecting, storing, processing environmental information, and communicating to neighbors nodes. For example, a WSN, at home can be used for light control, heating ventilation air conditioning (HVAC), security monitoring, and emergency event detection. And for industrial control can be used to improve the current manufacturing control system, detect unstable situations, control production pipelines, and so on.

ZigBee provides ultra low consumption and efficiency (thanks to the adaptable duty cycle, the low rate rates and the low coverage radio), and enable large scale networks for the WPN, making it one of the most convenient standards for this purpose.

Standard	ZigBee/IEEE 802.15.4	Bluetooth	UWB	IEEE 802.11 b/g
Working frequency	868/915 MHz, 2.4GHz	2.4 GHz	3.1 - 10.6 GHz	2.4 GHz
Range (m)	30 – 75+	10 – 30	~10	30 – 100 +
Data rate	20/40/250 kbps	1 Mbps	100+ Mbps	2 – 54 Mbps
Devices	255 – 65k	8		50 – 200
Power consumption	~1 mW	~40 – 100 mW	~80 – 300 mW	~160 mW – 600W
Cost ($US)	~2 – 5	~4 – 5	~5 – 10	~20 – 50

ZigBee provides ultra low consumption and efficiency (thanks to the adaptable duty cycle, the low rate rates and the low coverage radio), and enable large scale networks for the WPN, making it one of the most convenient standards for this purpose.

Standard	ZigBee/IEEE 802.15.4	Bluetooth	UWB	IEEE 802.11 b/g
Working frequency	868/915 MHz, 2.4GHz	2.4 GHz	3.1 - 10.6 GHz	2.4 GHz
Range (m)	30 – 75+	10 – 30	~10	30 – 100 +
Data rate	20/40/250 kbps	1 Mbps	100+ Mbps	2 – 54 Mbps
Devices	255 – 65k	8		50 – 200
Power consumption	~1 mW	~40 – 100 mW	~80 – 300 mW	~160 mW – 600W
Cost ($US)	~2 – 5	~4 – 5	~5 – 10	~20 – 50

The ZigBee Alliance

The ZigBee alliance objective is to work on the interoperability issues of IEEE 802.15.4/ZigBee protocol stacks. There are three levels of membership: Adopter, Participant, and Promoter.

- **Adopter**: Offers access to final, approved specifications, use of the Zigbee Member logo, participation in interoperability events, workshops and developers conferences.
- **Participant**: Offers full participation in all Alliance committees, work/task groups and member meetings. Participants earn voting rights in work groups and have early access to all Zigbee Alliance standards and specifications in development.
- **Promoter**: Offers automatic voting rights in all work groups, final approval rights on all standards and a seat on the Alliance Board of Directors

home using home Wi-Fi network.

(b) Gateways and backhaul network sublayer : The various access network sublayer endpoints could be connected to a common communication system called a gateway.

If you have ever done a networking connection, you must be aware that each end point device on the network has a default gateway defined through which it can communicate to an external network.

The devices on the same network (subnet) could talk to each other without the gateway. But, if the devices need to connect to other networks than its own network, then a gateway is required. For example, you can connect two phones, in close proximity, over Bluetooth and send data. But, if you need to send the data to a different phone, which is not in the local proximity, you will require to connect your phone to an external network, say a Wi-Fi, via a gateway. A common communication system organises multiple smart objects in a given area around a common gateway. The gateway communicates directly with the smart objects.

The role of the gateway is to forward the collected information through a longer-range medium (called the backhaul) to a central information processing station where the information is further processed through several applications. For example, if you want to remotely control your smart bulb at home, you would require a connection mechanism to control the bulb over cellular or Wi-Fi network via an application that could connect you to the gateway setup at your home using which you can operate the bulb.

infrastructure, supporting these legacy protocols with modern IP networks. Let's dig deeper into how these legacy serial protocols have evolved to use IP by looking specifically at DNP3 as a representative use case.

4.3.5 Distributed Network Protocol 3 (DNP3)

Distributed Network Protocol 3 (DNP3) is a set of communications protocols used between components in process automation systems. Its main use is in utilities such as electric and water companies. Like many of the other SCADA protocols, DNP3 is based on a master/slave relationship. The term master in this case refers to what is typically a powerful computer located in the control center of a utility, and a slave is a remote device with computing resources found in a location such as a substation. DNP3 refers to slaves specifically as outstations.

Outstations monitor and collect data from devices that indicate their state, such as whether a circuit breaker is on or off, and take measurements, including voltage, current, temperature, and so on. This data is then transmitted to the master when it is requested, or events and alarms can be sent in an asynchronous manner. The master also issues control commands, such as to start a motor or reset a circuit breaker and logs the incoming data. The Fig. 4.3.4 illustrates DNP3.

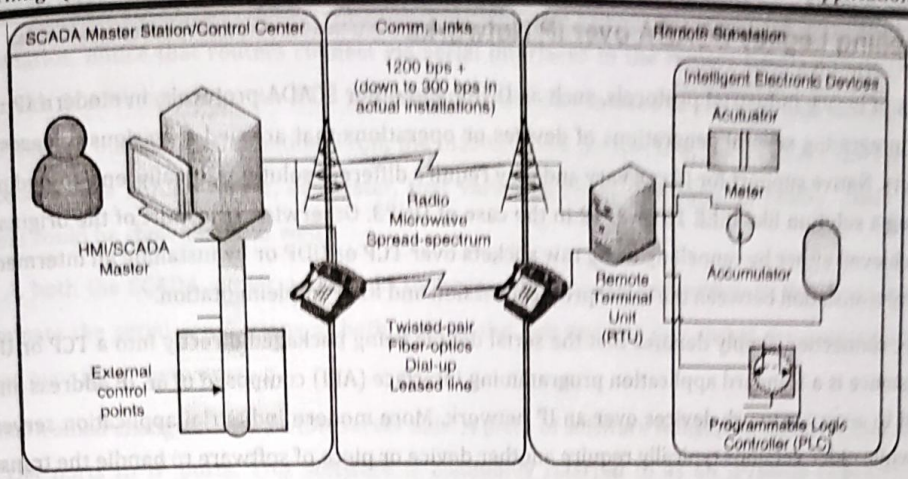


Fig. 4.3.4

The IEEE 1815-2012 specification describes how the DNP3 protocol implementation must be adapted to run either over TCP (recommended) or UDP. This specification defines connection management between the DNP3 protocol and the IP layers. The Fig. 4.3.5 shows the DNP3 connections over TCP or UDP.

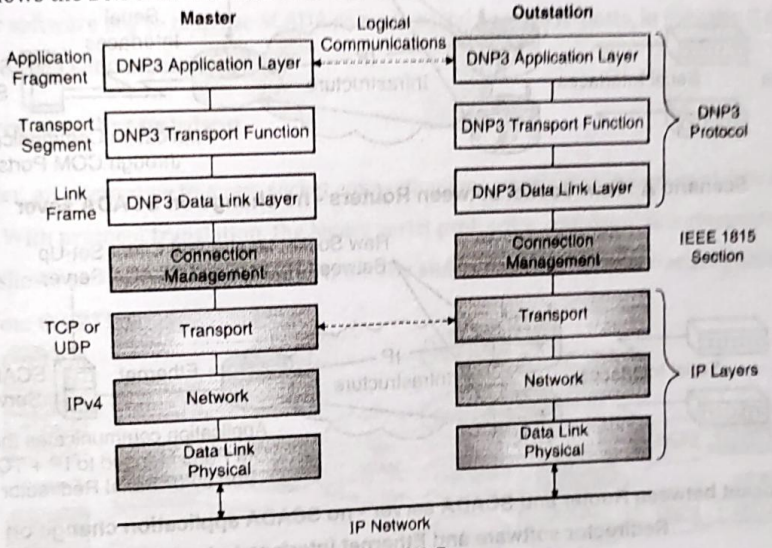


Fig. 4.3.5

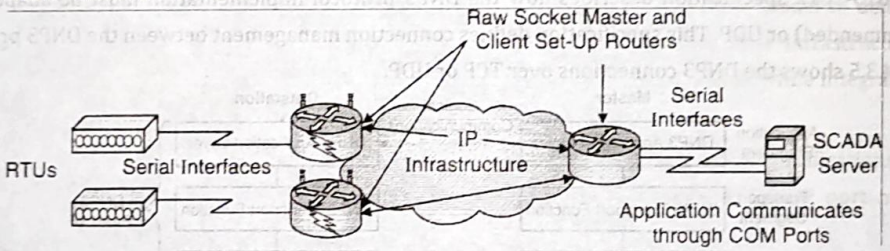
Connection management links the DNP3 layers with the IP layers in addition to the configuration parameters and methods necessary for implementing the network connection. The IP layers appear transparent to the DNP3 layers as each piece of the protocol stack in one station logically communicates with the respective part in the other. This means that the DNP3 endpoints or devices are not aware of the underlying IP transport that is occurring. The master side initiates connections by performing a TCP active open. The outstation listens for a connection request by performing a TCP passive open. Dual endpoint is defined as a process that can both listen for connection requests and perform an active open on the channel if required. Master stations may parse multiple DNP3 data link layer frames from a single UDP datagram, while DNP3 data link layer frames cannot span multiple UDP datagrams. Single or multiple connections to the master may get established while a TCP keepalive timer monitors the status of the connection.

Keepalive messages are implemented as DNP3 data link layer status requests. If a response is not received to a keepalive message, the connection is deemed broken, and the appropriate action is taken.

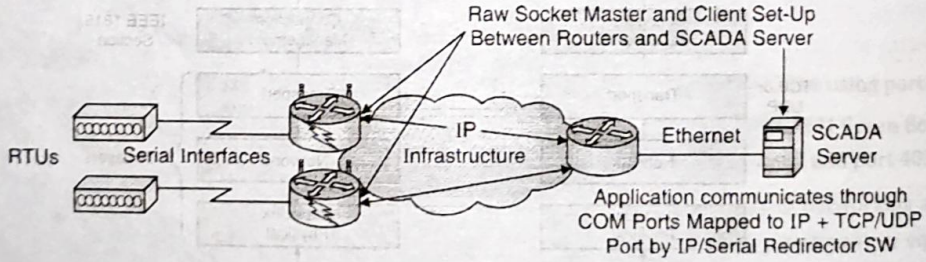
4.3.6 Tunnelling Legacy SCADA over IP Networks

Deployments of legacy industrial protocols, such as DNP3 and other SCADA protocols, in modern IP networks call for flexibility when integrating several generations of devices or operations that are tied to various releases and versions of application servers. Native support for IP can vary and may require different solutions. Ideally, end-to-end native IP support is preferred, using a solution like IEEE 1815-2012 in the case of DNP3. Otherwise, transport of the original serial protocol over IP can be achieved either by tunnelling using raw sockets over TCP or UDP or by installing an intermediate device that performs protocol translation between the serial protocol version and its IP implementation.

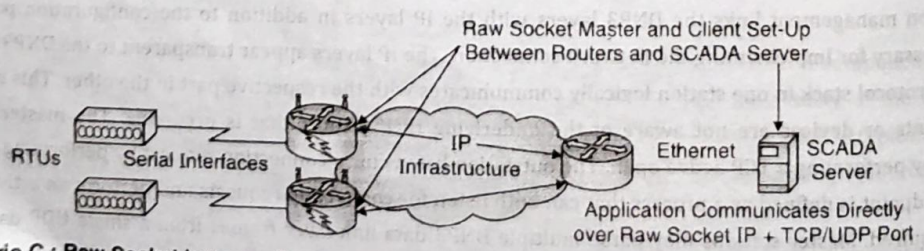
A raw socket connection simply denotes that the serial data is being packaged directly into a TCP or UDP transport. A socket in this instance is a standard application programming interface (API) composed of an IP address and a TCP or UDP port that is used to access network devices over an IP network. More modern industrial application servers may support this capability, while older versions typically require another device or piece of software to handle the transition from pure serial data to serial over IP using a raw socket. The Fig. 4.3.6 details raw socket scenarios for a legacy SCADA server trying to communicate with remote serial devices.



Scenario A : Raw Socket between Routers - no change on SCADA sever



Scenario B : Raw Socket between Router and SCADA server - no SCADA application change on server but IP/Serial Redirector software and Ethernet interface tp be added



Scenario C : Raw Socket between Router and SCADA server - SCADA application knows how to directly communicate over a Raw Socket and Ethernet interface

Fig. 4.3.6

In all the scenarios, notice that routers connect via serial interfaces to the remote terminal units (RTUs), which are often associated with SCADA networks. An RTU is a multipurpose device used to monitor and control various systems, applications, and devices managing automation. From the master/slave perspective, the RTUs are the slaves. Opposite the RTUs in each scenario is a SCADA server, or master, that varies its connection type. In reality, other legacy industrial application servers could be shown here as well.

- In Scenario A, both the SCADA server and the RTUs have a direct serial connection to their respective routers. The routers terminate the serial connections at both ends of the link and use raw socket encapsulation to transport the serial payload over the IP network.
- Scenario B has a small change on the SCADA server side. A piece of software is installed on the SCADA server that maps the serial COM ports to IP ports. This software is commonly referred to as an IP/serial redirector. The IP/serial redirector in essence terminates the serial connection of the SCADA server and converts it to a TCP/IP port using a raw socket connection.
- In Scenario C, the SCADA server supports native raw socket capability. Unlike in Scenarios A and B, where a router or IP/serial redirector software has to map the SCADA server's serial ports to IP ports, in Scenario C the SCADA server has full IP support for raw socket connections.

4.3.7 SCADA Protocol Translation

As mentioned earlier, an alternative to a raw socket connection for transporting legacy serial data across an IP network is protocol translation. With protocol translation, the legacy serial protocol is translated to a corresponding IP version. For example, the Fig. 4.3.7 shows two serially connected DNP3 RTUs and two master applications supporting DNP3 over IP that control and pull data from the RTUs.

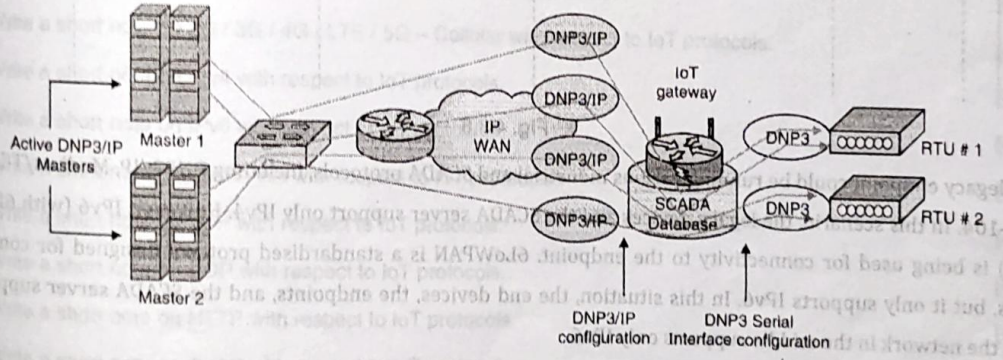


Fig. 4.3.7

The IoT gateway in the Fig. 4.3.7, performs a protocol translation function that enables communication between the RTUs and servers, despite the fact that a serial connection is present on one side and an IP connection is used on the other.

By running protocol translation, the IoT gateway connected to the RTUs is implementing a computing function close to the edge of the network. Adding computing functions close to the edge helps scale distributed intelligence in IoT networks. This can be accomplished by offering computing resources on IoT gateways or routers. Alternatively, this can also be performed directly on a node connecting multiple sensors. In either case, this is referred to as fog computing.

2.5 Smart Objects

The general dictionary meaning of the word smart is "having or showing a high degree of mental ability". Earlier, the physical things around you, such as TV, microwave, bulbs, etc., were kind of "dumb", having limited ways for you to connect with them and operate them.

In a nutshell,

✍ ***Definition :*** *Any physical object could be considered a smart objects if it allows some form of remote control, communication, and has processing capabilities.*

The Fig. 2.5.1 shows the high-level outline of a smart object.

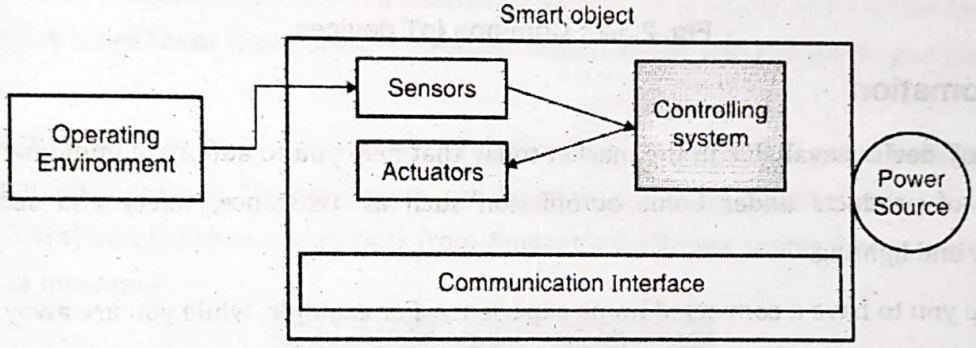


Fig. 2.5.1 : Outline of Smart Object

A smart object typically has the following components. There could be more, but these are the foundational ones.

1. **Controlling System :** The controlling system controls, manages, and operates the smart object. It typically runs a Real-Time Operating System (RTOS). RTOS provides a time-guarantee of completion for the given real-time tasks. The scheduler, in a RTOS, is designed to provide a predictable (deterministic) execution pattern.

   The controlling system could also be called as processing unit. Smart objects could have various types of controlling systems depending upon requirements, size, cost, operating environment, etc.

2. **Sensors :** Sensors get various inputs from the operating environment. A smart object may have one or more sensors depending upon the requirements. Sensors provide inputs to the controlling system based on which the controlling system may decide further processing steps.

3. **Actuators :** Actuators control the operating environment as required. They take inputs from the controlling system and perform operations in the operating environment as directed. A smart object may have one or more actuators depending upon the requirements.

4. **Communication Interface :** The ability to remotely control, manage, and operate is the desirable characteristic of a smart object. The communication interface enables a smart object to communicate with other objects in the environment as well as humans. It could be either wired or wireless. Through the communication interface, a smart object gets connected to a network.

5. **Power Source :** A smart object could be either battery operated or could be directly plugged into the regular powerline. Smart objects could have varied power requirements. Typically, it is required to have low power consumption as these devices often operate in areas where there is no powerline (hence operated on batteries) and also it could be hard for humans to easily reach the objects as they could be placed in remote areas or beyond general reach.

2.5.1 Common Smart Objects (IoT Devices)

There are several IoT devices available in the market today. Some of them are as shown in Fig. 2.5.2.

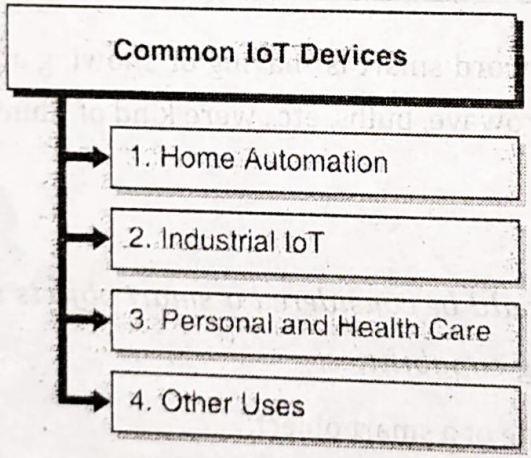


Fig. 2.5.2 : Common IoT devices

Fig. 5.2.7 : Surveillance applications

5.3 Health

You have already read about several personal and health care devices in Unit 2, Section 2.5.1(C) that could be used to continuously monitor your health and help you to stay fit.

Let's talk about some of them specifically.

5.3.1 Fitness and Health Monitoring

Wearable IoT devices allow non-invasive and continuous monitoring of physiological parameters of your body and can help in continuous health and fitness monitoring. These wearable devices can be in various forms such as belts and wrist-bands. The wearable devices form a type of wireless sensor networks called body area networks. The measurements from a number of wearable devices are continuously sent to a master node (such as a smart-phone) which then sends the data to a server or a cloud-based back-end for analysis and reporting. Health-care providers can analyse the collected health-care data to determine any health conditions or anomalies. Example of various parameters, that such fitness and health monitoring devices continuously measure, are as following.

1. Body temperature
2. Heart rate
3. Pulse oximeter oxygen saturation (SP02)
4. Blood pressure
5. Electrocardiogram (ECG)
6. Movement and steps (with accelerometers)
7. Electroencephalogram (EEG)
8. Sleeping patterns
9. Calories burned

Fig. 5.3.1 shows a simple picture illustrating fitness and health monitoring apps.

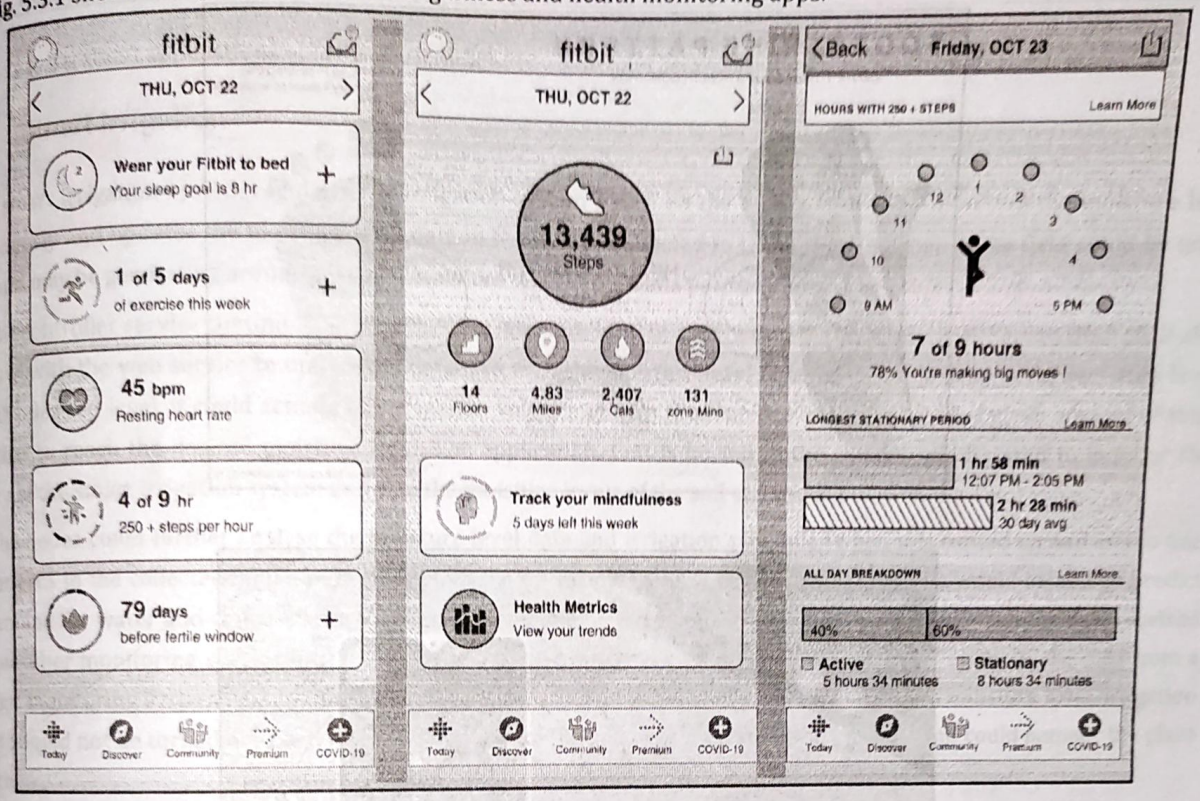


Fig. 5.3.1 : Fitness and health monitoring

5.3.2 Wearable Electronics

Wearable electronics are simpler than full-scale wearable computers. While a wearable computer has both input and output and is capable of adjusting to multiple tasks, wearable electronics are constructed with set tasks to fulfil one or more needs of a specific target group. Wearable electronics differ from mobile devices by their appearance and by being fundamentally designed to be worn on the body. A true piece of wearable electronics is also required to be worn to function, i.e. conceptually linked to the wearer's body. Some wearable devices require the user interface to be present and available all the time, meaning they are more obtrusive than devices with no input (such as the wrist unit and the chest belt of a heart-rate monitor).

Wearable electronics such as wearable gadgets (smart watches, smart glasses, wristbands, etc.) and fashion electronics (with electronics integrated in clothing and accessories, (e.g., Google Glass or Moto 360 smart watch) provide various functions and features to assist you in your daily activities and making us lead healthy lifestyles.

Smart watches that run mobile operating systems (such as Android) provide enhanced functionality beyond just timekeeping. With smart watches, the users can search the Internet, play audio/video files, make calls (with or without paired mobile phones), play games and use various kinds of mobile applications. Smart glasses allows users to take photos and record videos, get map directions, check flight status, and search the Internet by using voice commands. Smart shoes monitor the walking or running speeds and jumps with the help of embedded sensors and be paired with smart-phones to visualize the data. Smart wristbands can track the daily exercise and calories burnt. Figs. 5.3.2, 5.3.3 and 5.3.4 are a few simple pictures illustrating various wearable electronics.

Finally, you are done with several interesting topics in IoT domain! Long but enjoyable ride, isn't it? Let's conclude with a few case studies to wrap things around.

Note : I would take very simplistic examples for various case studies. The real-life IoT systems could be much more complex and may involve several other components, protocols and other functionalities that would depend on the manufacturer and the purpose you are looking to fulfill. Also, I would focus on concepts behind these case studies instead of talking about exact hardware, components, protocols and other details that may vary from solution to solution.

5.1 Home Automation

You have already got a glimpse of how IoT devices can help in home automation in Unit 1, Section 1.2.

Various IoT applications in homes

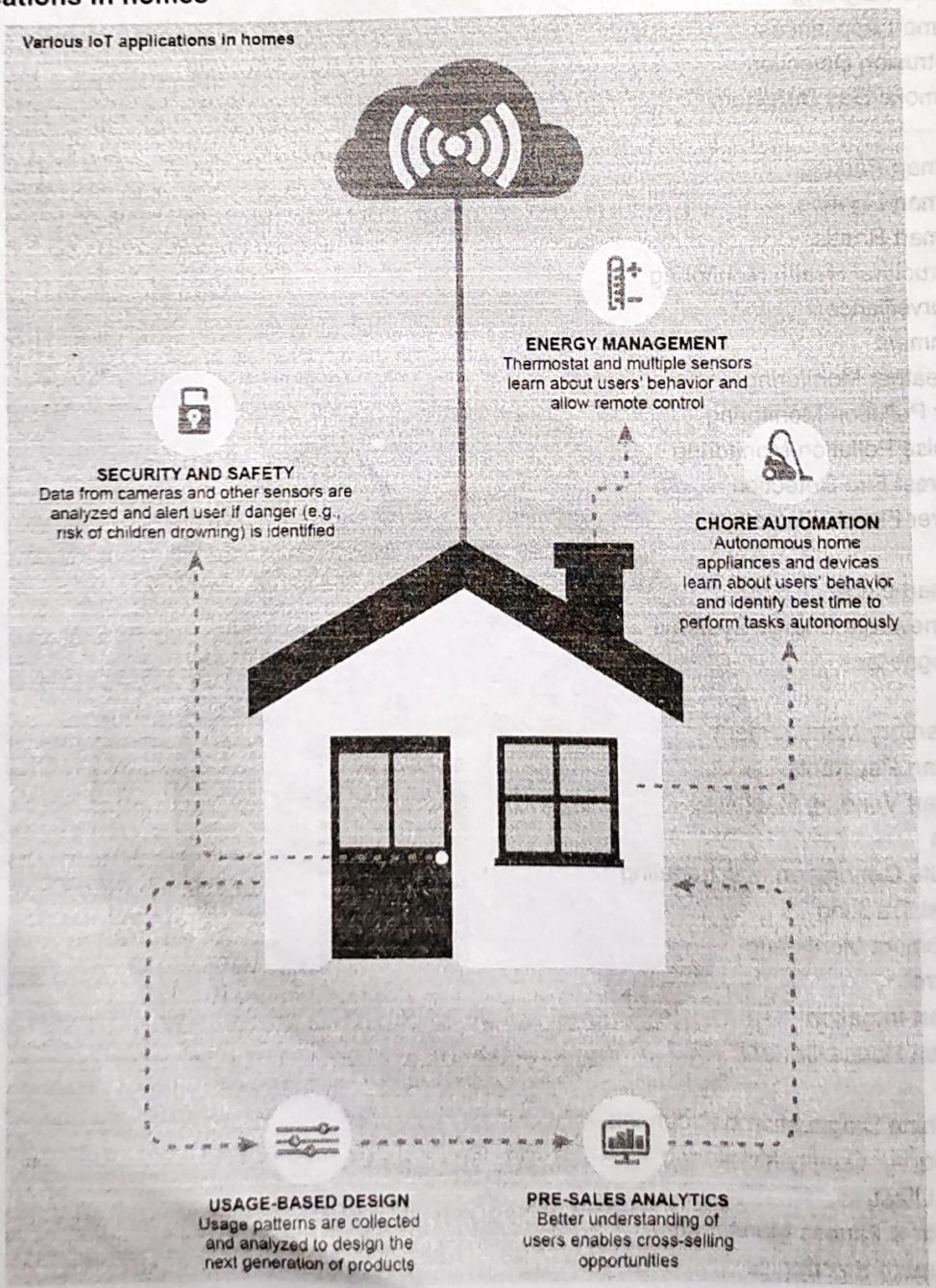


Fig. 5.1.1

You have also read about several appliances in Unit 2, Section 2.5.1 that could be used to automate several mundane tasks at home. Let's talk about some of them specifically.

5.1.1 Smart Lighting for Home

Have you left home, at times, without switching off lights? Do the lights in your home turn on only at either full power or none without depending on the illumination level required? During night, at times, have you found it hard to locate the light switch and fumbled a few times? Do you want to change the light colour based on your mood but have no way to do it? If you answered yes to any of these questions, then probably you understand what you would expect smart lights for your home to do.

Smart lighting for homes helps in saving energy by adapting the lighting to the ambient conditions and switching on/off or dimming the lights when needed. Key enabling technologies for smart lighting include solid state lighting (such as LED lights) and IP-enabled lights. For solid state lighting solutions both spectral and temporal characteristics can be configured to adapt illumination to various needs. Smart lighting solutions for home achieve energy savings by sensing the human movements and their environments and controlling the lights accordingly. Wireless-enabled and Internet connected lights can be controlled remotely from IoT applications such as a mobile or web application. Smart lights with sensors for occupancy, temperature, lux level, etc., can be configured to adapt the lighting (by changing the light intensity, colour, etc.) based on the ambient conditions sensed, in order to provide a good ambiance.

Smart lights are embedded with ambient intelligence gathered from a distributed smart wireless sensor network to optimise and control the lighting system to be more efficient and user-oriented. A solid state lighting model can be implemented based on a wireless sensor network that provides services for sensing illumination changes and dynamically adjusting luminary brightness according to user preferences.

Fig. 5.1.2 is a simple picture illustrating smart light solutions for home.

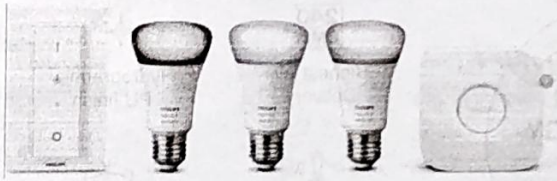


fig. 5.1.2

Smart light solutions for home typically provide the following functionalities.

1. **Be creative with 16 million colours :** You can play with light and choose from 16 million colours to instantly change the look and atmosphere of your room. You can set the scene effortlessly with one touch of a button. You can use a favourite photo and relive that special moment with splashes of light. You can also save your favourite light settings and recall them whenever you want with the tap of a finger.
2. **Wake up and go to sleep naturally :** Smart lights may help you get out of bed the way you like it, helping you start your day feeling refreshed. The light brightness increases gradually mimicking the effect of sunrise and helps you wake up naturally, instead of being woken up by the loud sound of an alarm clock.

   You can rt your day, the right way. In the evening, the relaxing warm white light helps you to unwind, relax and prepare your body for a good night's sleep.
3. **Create your ambiance with warm white to cool daylight :** You can set the right ambiance for any moment and decorate your home with warm to cool white light. You can enjoy different styles throughout the year, no matter if it's the crisp white light reminding you of a spring breeze, the warm white light of a summer sun, or the ice cool daylight of winter.

4. **Smart control, home and away :** These smart lights can be remotely operated and controlled through apps on your mobile phone. You can check if you have forgotten to switch your lights off before you left your home and switch them on if you are working late.
5. **Set timers for your convenience :** Smarts lights can make it seem like you are home when you are not, using the schedule function. You can set the lights to come on at a pre-set time, so the lights are on when you arrive home. You can even set rooms to light up at different times. You can also let the lights turn off gradually in the night, so you never have to worry whether you have left any lights on.

5.1.2 Smart Appliances

Modern homes have a number of appliances such as TVs, refrigerators, music systems, washer/dryers, etc. Managing and controlling these appliances can be cumbersome, with each appliance having its own controls or remote controls. Smart appliances make the management easier and also provide status information to the users remotely.

For example,

- Smart washer/dryers that can be controlled remotely and notify when the washing/drying cycle is complete.
- Smart thermostats allow controlling the temperature remotely and can learn the user preferences.
- Smart refrigerators can keep track of the items stored (using RFID tags) and send updates to the users when an item is low on stock.
- Smart TVs allows users to search and stream videos and movies from the Internet on a local storage drive, search TV channel schedules and fetch news, weather updates and other content from the Internet.

You can literally buy smart version of any appliance today. Alexa controlled fan as well!

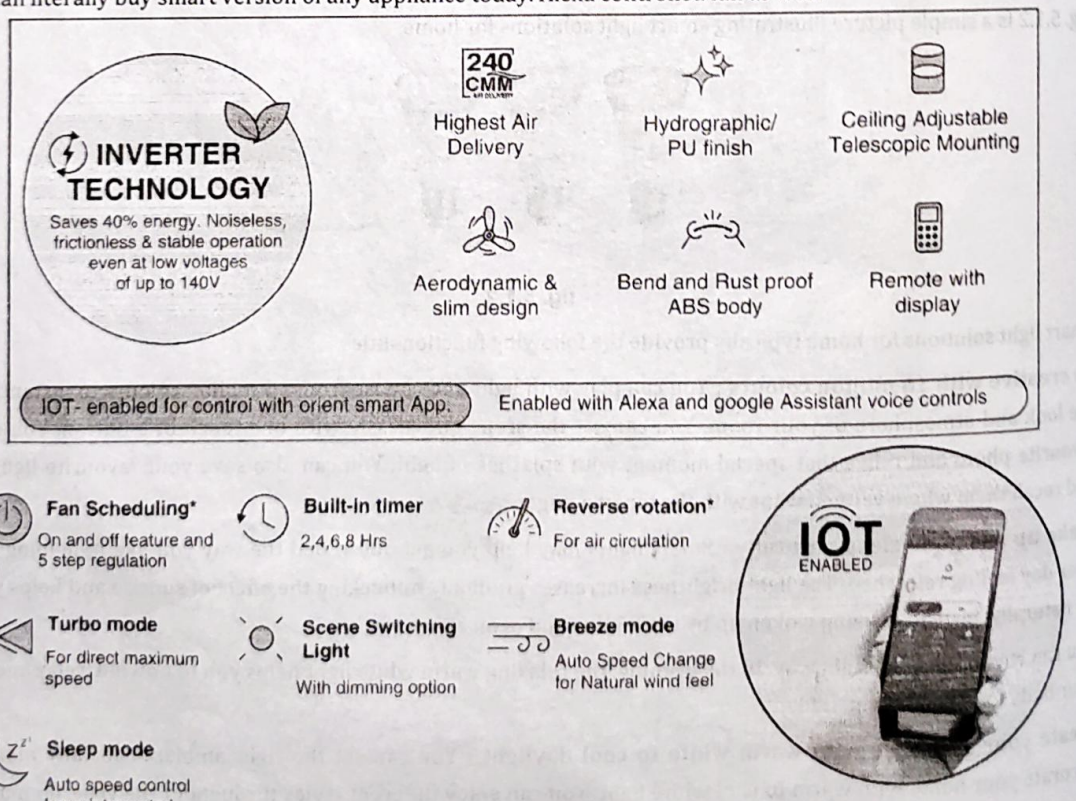


Fig. 5.1.3

5.1.3 Intrusion Detection

A home intrusion detection system could possibly detect if someone tries to break-in into your home in your absence. Home intrusion detection systems use security cameras and sensors (such as PIR sensors and door sensors) to detect intrusions and raise alerts. Alerts can be in the form of an SMS, or an email sent to the user. Advanced systems can even send detailed alerts such as an image grab or a short video clip sent as an email attachment. A cloud controlled intrusion detection system could also use location-aware services, where the geo-location of each node of a home automation system is independently detected and stored in the cloud. In the event of intrusions, the cloud services alert the accurate neighbours (who are using the home automation system) or local police. Fig. 5.1.4 illustrating home intrusion detection system.

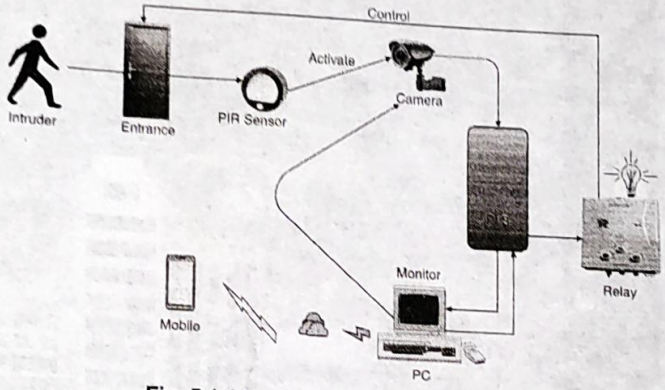


Fig. 5.1.4 : Intrusion detection system

5.1.4 Smoke / Gas Detector

Smoke detectors are installed in homes and buildings to detect smoke that is typically an early sign of fire. Smoke detectors use optical detection, ionization or air sampling techniques to detect smoke. Alerts raised by smoke detectors can be in the form of signals to a fire alarm system. Gas detectors can detect the presence of harmful gases such as Carbon Monoxide (CO), Liquid Petroleum Gas (LPG), etc. A smart smoke/gas detector can also raise alerts in human voice describing where the problem is, send or an SMS or email to the user or the local fire safety department and provide visual feedback on its status (healthy, battery-low, etc.). The Fig. 5.1.5 illustrates such a system.

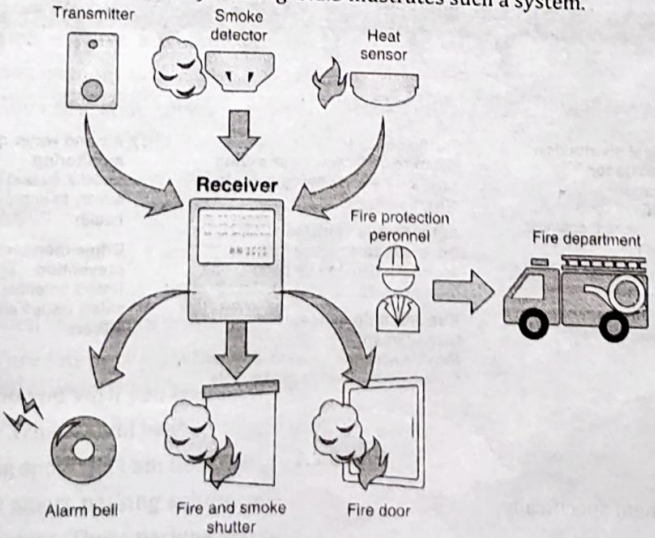


Fig. 5.1.5

Basis of	COAP	MQTT
Abbreviation	Constrained Application Protocol	Message Queuing Telemetry Transport
Communication Type	It uses Request-Response model.	It uses Publish-Subscribe model
Messaging Mode	This uses both Asynchronous and Synchronous.	This uses only Asynchronous
Transport layer protocol	This mainly uses User Datagram protocol(UDP)	This mainly uses Transmission Control protocol(TCP)
Header size	It has 4 bytes sized header	It has 2 bytes sized header
RESTful based	Yes it uses REST principles	No it does not uses REST principles
Persistence support	It does not has such support	It supports and best used for

Persistence support	It does not has such support	It supports and best used for live data communication
Message Labelling	It provides by adding labels to the messages.	It has no such feature.
Usability/Security	It is used in Utility area networks and has secured mechanism.	It is used in IoT applications and is secure
Effectiveness	Effectiveness in LNN is excellent.	Effectiveness in LNN is low.
Communication Model	Communication model is one-one.	Communication model is many-many.