IoT Course Module

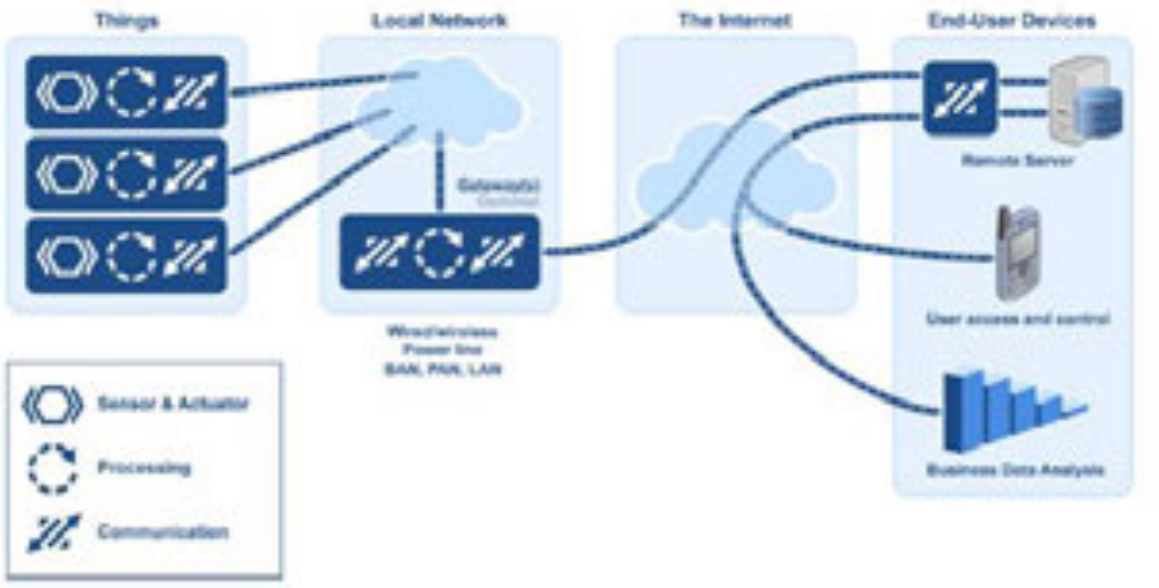
Table of Contents

Introduction

Internet of Things (IoT) is an important part of the new generation of information technologies and an important development phase in the information era. IoT is widely used in network convergence by using communications and sensing technologies, such as intelligent sensing, identification, and pervasive computing. Therefore, IoT is the third wave in the global information industry development after computers and internet.

Definition of IoT

The Internet of Things (IoT) is the network of physical devices, vehicles, home appliances and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these objects to connect and exchange data. The term IoT refers to a system of computing devices in the physical world which have been connected to the internet. These devices can either send or receive data from the internet.



Origin of IoT

Trojan Room coffee pot in 1991. At the Trojan room of the computer Laboratory in Cambridge University. Scientists went downstairs to see if the coffee was cooked, but often returned empty handed.To solve this problem, they wrote a set of programs and installed a portable camera next to the coffee pot. The camera was aimed at the coffee pot. Computer image capture technology was used to check whether the coffee was cooked at any time, eliminating the need to move up and down the stairs.

Some Important Definitions in IoT

Sensors-this is basically a device that detects events in the physical world. Sensors are dumb components that in most cases produce voltages or some other quantity, in proportion to what they are measuring. There all kinds and manners of sensors; temperature, pressure, humidity, altitude, gas, light etc. Even cameras are primarily sensors! Sensors are always attached to a microprocessor unit, forming a sensor node. A sensor node is a subsystem consisting of sensors, microcontrollers or microprocessors, power supply and some communication mechanism. The sensor collects this information from the environment, the microprocessor interprets and processes to make sense of the data and the communication mechanism transmits the data to the internet.

Actuators-Remember we said that sensors collect data from the physical world, well what if we want to control some behaviour remotely? Actuators are basically components used to move or

control a system. In the example of the ceiling fan, actuators are used to change the direction and speed of the blades accordingly.

IoT Communication Network-This refers to the mode at which these 'things' communicate to the internet. Such technologies include Ethernet, Wi-Fi, GSM, LoRaWAN, Bluetooth etc. Devices may combine several communication technologies in order to send and receive data from the internet. For example, Bluetooth cannot be used directly to connect to the internet. The sensor node might be connected to a Bluetooth gateway which is in turn connected to an Ethernet network and thus able to communicate with the internet.

IoT Cloud- This is the system that the 'things' are so interested in connecting to. The cloud basically refers to computing resources that can be accessed over the internet. Instead of storing your files on your computer, you could store them on Google Drive and access them from anywhere as long as there is an internet connection. Same scenario for IoT. Sensor nodes are collecting massive amounts of data that needs to be stored somewhere. The IoT cloud offers this somewhere. The IoT cloud is not only used for storage but applications that use this data can also run on the cloud. Back to our ceiling fan example, you could run an application (on the cloud) that assesses the temperature sent room the ceiling fan. This application could periodically check the data coming from the app and automatically adjust the speed and direction of the fan without you ever having to do anything. Another aspect of the IoT cloud is analytics. IoT devices collect huge amounts of data. We need to make sense of this data to enable us to make informed decisions from it. That is where data analytics comes into play.

Dashboard- Users may need a platform to display information collected (and analytics) and a way to control their devices. This can either through a web interface or through a mobile phone app (Android or iOS).

Application *Areas* of IoT

There are many uses in IoT as listed below:-

1. Agriculture

Farmers are using IoT to track equipment location and performance, and increasingly livestock grazing in open pastures. IoT sensors are also determining soil moisture levels to control irrigation systems and minimize water consumption. A farmer could have sensors connected around their farm to monitor and store temperature, humidity and soil moisture levels.

2. Predictive maintenance

With the use of sensors, cameras and data analytics, managers in a range of industries are able to determine when a piece of equipment will fail before it does. These IoT-enabled systems can sense warning signs, use data to create maintenance timelines and pre-emptively service equipment before problems occur.

3. Smart metering

A smart meter is an internet-capable device that measures energy, water or natural gas consumption of a building or home. Traditional meters only measure total consumption, whereas smart meters record when and how much of a resource is consumed. Power companies are deploying smart meters to monitor consumer usage and adjust prices according to the time of day and season.

Smart metering also helps utilities:

- Reduce operating expenses by managing manual operations remotely
- Improve forecasting and streamline power-consumption
- Improve customer service through profiling and segmentation
- Reduce energy theft

Simplify micro-generation monitoring and track renewable power

4. Asset tracking

The goal of asset tracking is to allow an enterprise to easily locate and monitor key assets, including along the supply chain (e.g. raw materials, final products and containers) to optimize logistics, maintain inventory levels, prevent quality issues and detect theft.

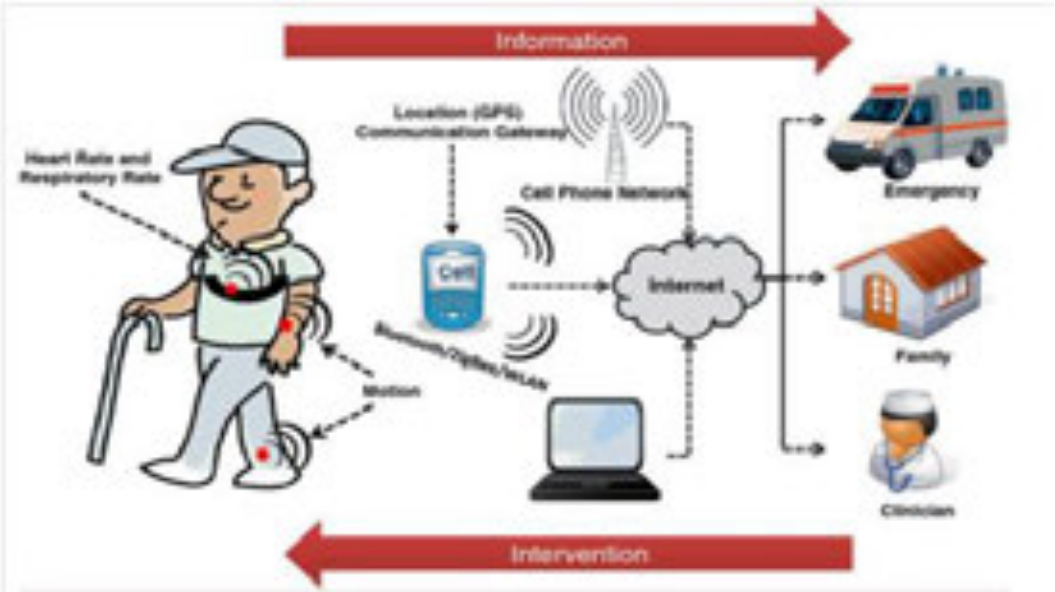
One industry that heavily relies on asset tracking is maritime shipping. On a large scale, sensors help track the location of a ship at sea, and on a smaller scale, they can provide the status and temperature of individual cargo containers. One benefit is real-time metrics on refrigerated containers; these containers must be stored at constant temperatures so perishable goods remain fresh.

5. Connected Vehicles (IoV)

These are computer-enhanced vehicles that automate many normal driving tasks – in some cases, even driving themselves. Cameras, radar and lasers are among the sensors feeding information into the differential GPS. Cameras let the car's computers see what's around it, while radar allows vehicles to see up to 100 meters away in the dark, rain, or snow. Lasers, which look like a spinning siren light, continuously scan the world around the car and provide the vehicle with a continuous, 3-D omnidirectional view of its surroundings.

6. Healthcare

IoT can be used in monitoring blood pressure and heart rate, for example, and even automatically alerting emergency personnel when they detect problematic readings—or to send help when an elderly person has fallen and can't get up.



IoT in Remote Health Monitoring

7. Homes

Connected devices include TVs, refrigerators, lights, thermostats, smoke detectors and other sensors, security systems and much more. Hotels have long pioneered using IoT for room keys. You could hook up a camera subsystem in your fridge that checks which food items are used up and sends you a text alert. Or maybe you want your umbrella to tell you whenever it's going to rain as you are about to leave your house.

8. Smart parking

Sensors can be used to sends a message when a vehicle arrives or leaves a parking spot. This can be combined with an app to find a free parking spot

9. Environment monitoring

IoT allows for the collection of sound, temperature, pollution, radiation, humidity readings. This can be used to create valuable insights in combination with geolocation.

Benefits of IoT

Communication

IoT encourages the communication between devices, also famously known as Machine-to-Machine (M2M) communication. Because of this, the physical devices are able to stay connected and hence the total transparency is available with lesser inefficiencies and greater quality.

Automation and Control

Due to physical objects getting connected and controlled digitally and centrally with wireless infrastructure, there is a large amount of automation and control in the workings. Without human

intervention, the machines are able to communicate with each other leading to faster and timely output.

Information

It is obvious that having more information helps making better decisions. Whether it is mundane decisions like needing to know what to buy at the grocery store or if your company has enough widgets and supplies, knowledge is power and more knowledge is better.

Information Monitor

The second most obvious advantage of IoT is monitoring. Knowing the exact quantity of supplies or the air quality in your home, can further provide more information that could not have previously been collected easily. For instance, knowing that you are low on milk or printer ink could save you another trip to the store in the near future. Furthermore, monitoring the expiration of products can and will improve safety.

Time

As hinted in the previous examples, the amount of time saved because of IoT could be quite large. And in today's modern life, we all could use more time.

Money

The biggest advantage of IoT is saving money. If the price of the tagging and monitoring equipment is less than the amount of money saved, then the Internet of Things will be very widely adopted. IoT fundamentally proves to be very helpful to people in their daily routines by making the appliances communicate to each other in an effective manner thereby saving and conserving energy and cost. Allowing the data to be communicated and shared between devices and then translating it into our required way, it makes our systems efficient.

Automation of daily tasks leads to better monitoring of devices

The IoT allows you to automate and control the tasks that are done on a daily basis, avoiding human intervention. Machine-to-machine communication helps to maintain transparency in the processes. It also leads to uniformity in the tasks. It can also maintain the quality of service. We can also take necessary action in case of emergencies.

Efficient and Saves Time

The machine-to-machine interaction provides better efficiency, hence; accurate results can be obtained fast. This results in saving valuable time. Instead of repeating the same tasks every day, it enables people to do other creative jobs.

Saves Money

Optimum utilization of energy and resources can be achieved by adopting this technology and keeping the devices under surveillance. We can be alerted in case of possible bottlenecks, breakdowns, and damages to the system. Hence, we can save money by using this technology.

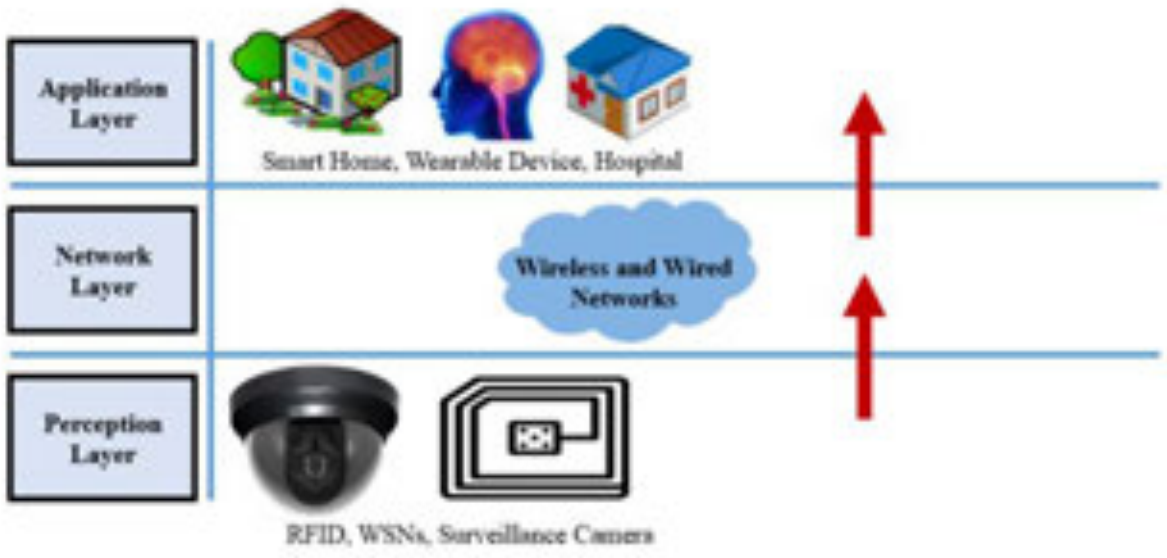
Better Quality of Life

All the applications of this technology culminate in increased comfort, convenience, and better management, thereby improving the quality of life.

IoT Architecture

Three Layer Architecture

It is a very basic architecture and fulfills the basic idea of IoT. It was proposed in the early stages of development of IoT. It has three layers. The names of these three layers are perception, network and application layer as shown in Figure below.



Perception Layer

It is also known as a sensor layer. It works like people's eyes, ears and nose. It has the responsibility to identify things and collect the information from them. There are many types of sensors attached to objects to collect information such as RFID, 2-D barcode and sensors. The sensors are chosen according to the requirement of applications. The information that is collected by these sensors can be about location, changes in the air, environment, motion, vibration, etc.

Network Layer

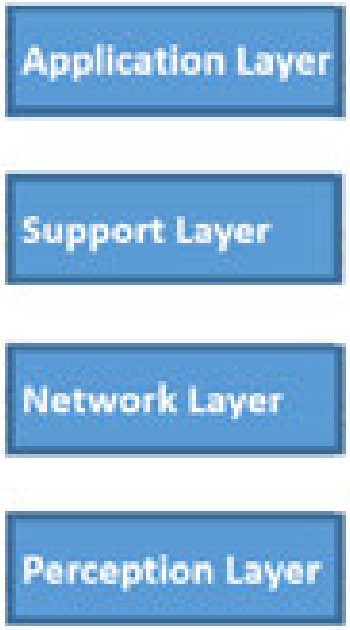
Network layer is also known as transmission layer. It acts like a bridge between perception layer and application layer. It carries and transmits the information collected from the physical objects through sensors. The medium for the transmission can be wireless or wire based. It also takes the responsibility for connecting the smart things, network devices and networks to each other.

Application Layer

Application layer defines all applications that use the IoT technology or in which IoT has deployed. The applications of IoT can be smart homes, smart cities, smart health, animal tracking, etc. It has the responsibility to provide the services to the applications. The services may be varying for each application because services depend on the information that is collected by sensors

Four Layer Architecture

The three-layer architecture was the most basic architecture. Due to continuous development in IoT, it could not fulfill all the requirements of IoT. Therefore, researchers proposed an architecture with four layers. It has three layers like the previous architecture, but it also has one more layer called a support layer. Figure below presents the four layer architecture. Application, network and perception layers have the same functionality as the three-layer architecture that has already been discussed.



Support Layer: The reason to make a fourth layer is the security in architecture of IoT. Information is sent directly to the network layer in three-layer architecture. The chances of getting threats increase due to sending information directly to the network layer. A new layer is proposed due to flaws that are available in three-layer architecture. In the four-layer architecture, information is sent to a support layer that is obtained from a perception layer. The support layer has two responsibilities. It confirms that information is sent by the authentic users and protected from threats. There are many ways to verify the users and the information. The most commonly used method is the authentication. It is implemented by using pre-shared secrets, keys and passwords. The second responsibility of the support layer is sending information to the network layer. The medium to transmit information from the support layer to network layer can be wireless and wire based.

Five Layer Architecture

The four-layer architecture played an important role in the development of IoT. There were also some issues regarding security and storage in four-layer architecture. Researchers proposed five-layer architecture to make the IoT secure. It has three layers like previous architectures whose names are perception layer, transport layer and application layer. It also has two more layers. The names of these newly proposed layers are processing layer and business layer. It is

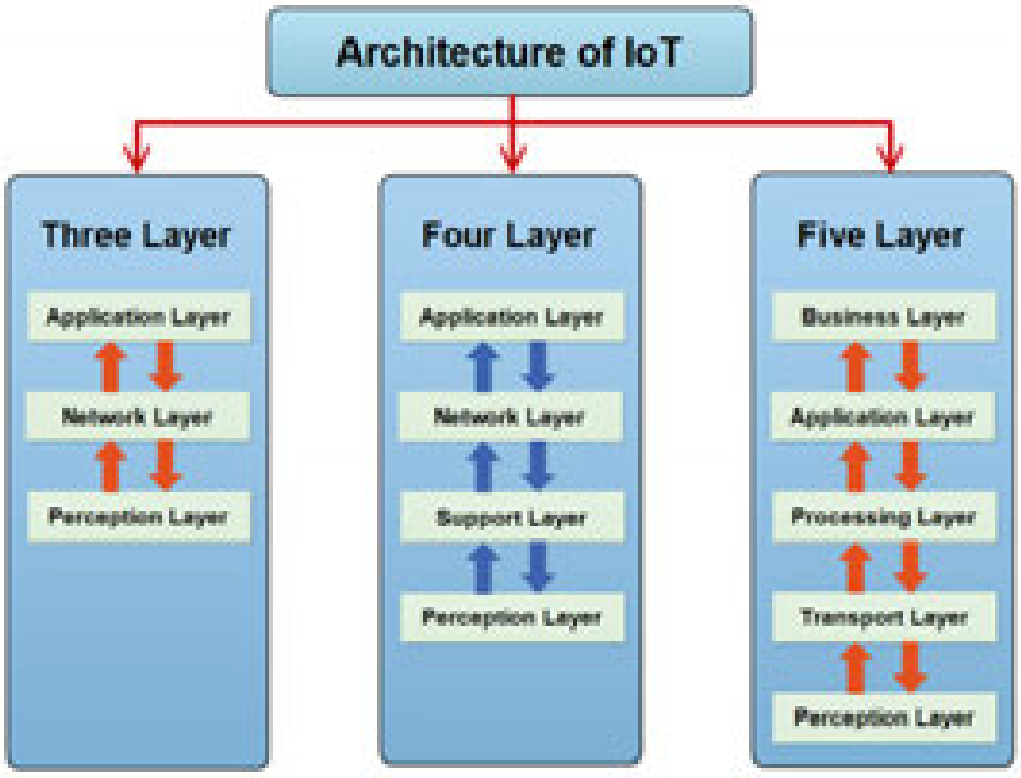
considered that the newly proposed architecture has the ability to fulfill requirements of IoT. It also has the ability to make the applications of IoT secure.

Processing Layer

The processing layer is also known as a middleware layer. It collects the information that is sent from a transport layer. It performs processing onto the collected information. It has the responsibility to eliminate extra information that has no meaning and extracts the useful information. However,it also removes the problem of big data in IoT. In big data, a large amount of information is received which can affect performance of IoT.

Business Layer

The business layer refers to an intended behavior of an application and acts like a manager of a whole system. It has responsibilities to manage and control applications, business and profits models of IoT. The user's privacy is also managed by this layer. It also has the ability to determine how information can be created, stored and changed.



IoT Protocol Stack

IoT has its own protocol stack, different from the known OSI model and TCP/IP protocol stack. IoT model protocol stack is shown in the figures below.

Layer	Protocols
Application Layer	CoAP, MQTT, XMPP, AMQP, RESTFUL, Websockets
Transport Layer	UDP, DTLS
Internet Layer	RPL, 6LoWPAN
Physical/Link Layer	IEEE 802.15 Series, IEEE 802 11 series

Application Layer Protocols

MQTT

It is a publish-subscribe lightweight messaging protocol. It is designed for situations where low impact is required and where bandwidth is limited. The publish-subscribe pattern requires a message broker. The broker is responsible for distributing messages to recipient clients.

CoAP

(Constrained Application Protocol) - it is a protocol for low power networks and nodes (bound) with data. CoAP, like HTTP, is a RESTful protocol. It is semantically aligned with HTTP and even has a bidirectional one-to-one mapping with HTTP. CoAP fully meets the needs of an extremely light protocol with a permanent connection.

XMPP

(Extensible Messaging and Presence Protocol) – it consists of a set of open protocols for real-time communication, with a wide range of applications including instant messaging, multi-party chats, voice and video calls, lightweight middleware, content syndication and routing generalized XML data. XMPP-IoT: XMPP system dedicated to IoT which aims to make the communication between machine to people and machine to machine.

AMQP

(Advanced Message Queuing Protocol) - it is an open standard protocol, at the Application level for message-oriented middleware. The AMQP definition features are message orientation, queuing, routing (including point-to-point and publish-and-subscribe), reliability and security.

Transport Layer Protocols

UDP

While TCP is used predominantly in the Internet as Transport Layer Protocol (except gaming or video streaming where User Datagram Protocol or UDP is used), most IoT scenarios are well suited for UDP. UDP is a much lighter protocol compared to TCP. UDP is a connection protocol and does not come with resiliency features of TCP, such as guaranteed packet delivery. On the other hand, UDP is much faster than TCP, the header size is much smaller than TCP – making it suitable for the constrained environment of devices and sensors. Higher level Application Layer IoT protocols like CoAP use UDP rather than TCP.

DTLS

DTLS or Datagram Transport Layer Security is a TLS/SSL counterpart that runs on UDP. The way TLS/SSL takes care of security for TCP communication, DTLS provides the same security features on UDP or Datagrams.

Internet Layer

6LoWPAN

6LoWPAN is the secret sauce that allows larger IPv6 packets to flow over 802.15.4 links that support much smaller packet sizes. 6LoWPAN is the acronym of IPv6 over Low Power Wireless Personal Area Networks. So 6LoWPAN as the name implies is an adaptation layer that allows transport of IPv6 packets over 802.15.4 links. Without 6LoWPAN IPv6, internet protocols would not work in these Low Power Wireless Personal Area Networks that uses IEEE 802.15.4. 6LoWPAN is an open standard defined under RFC 6282 by the Internet Engineering Task Force (IETF), the body that defines many of the open standards used on the internet such as UDP, TCP and HTTP to name a few. As mentioned previously, an IPv6 packet is too large to fit into a single 802.15.4 frame. What 6LoWPAN does to fit an IPv6 packet in 802.15.4 frame is -

- Fragmentation and Reassembly - It fragments the IPv6 packet and sends it through multiple smaller size packets that can fit in an 802.15.4 frame. On the other end, it reassembles the fragmented packets to re-create the IPv6 packet
- Header Compression – Additionally it also compresses the IPv6 packet header to reduce the packet size

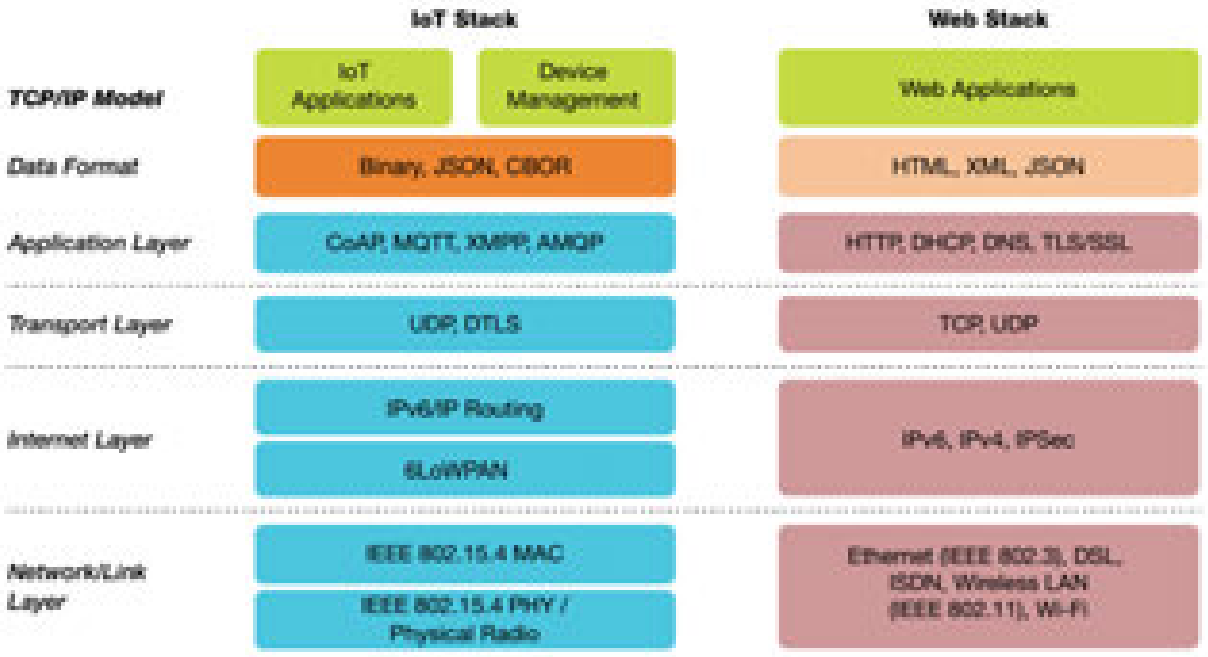
RPL

RPL stands for **Routing Protocol for Low-Power and Lossy Networks.** IPv6 routing for low power networks. RPL is a routing protocol for wireless networks with low power consumption and generally susceptible to packet loss. It is a proactive protocol based on distance vectors and operates on IEEE 802.15.

Data Link and Physical Layer Protocols

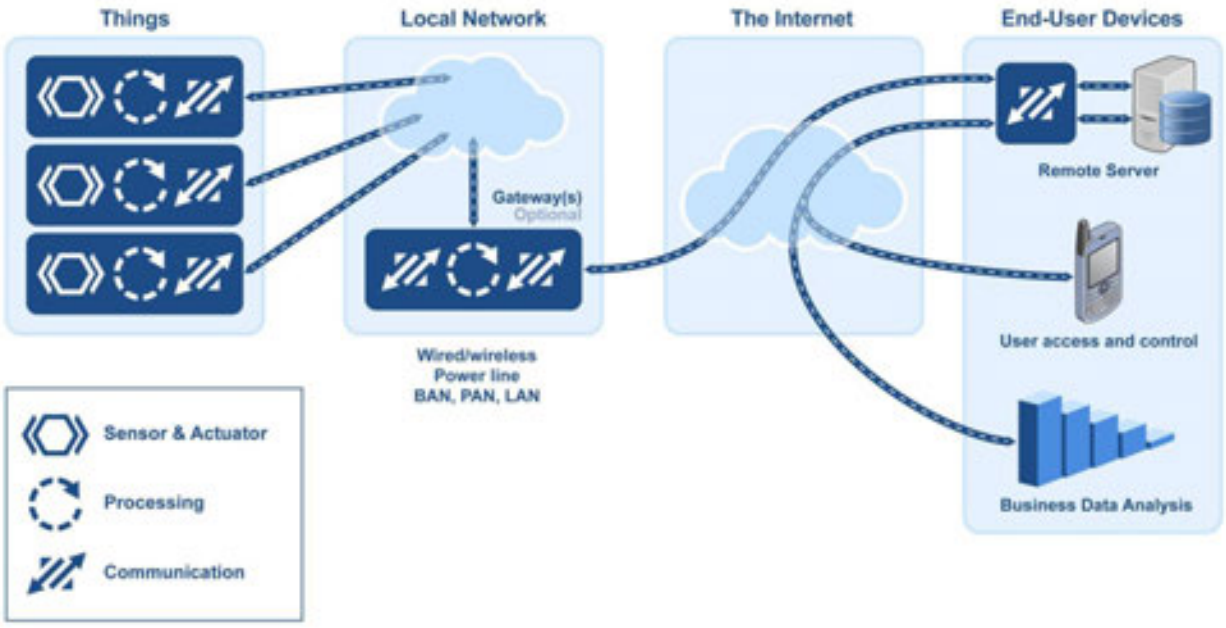
Technology	Frequency band	Range	Data rate	Battery life	Topology	Standardization	Governing body
RFID [1]	Low / High / Ultra-high	1 cm – 100 m	1 – 100 kbps	passive: N/A active: 3–5 years	P2P	open standard	no single body
NFC [14]	13.56 MHz	0.2 m	424 kbps	passive: N/A active: 3–5 years	P2P	open standard	ISO/IEC
BLE [15]	2.4 GHz	10 – 100 m	1 Mbps	months to years	P2P / Star	open standard	Bluetooth SIG
Ant [16]	2.4 GHz	30 m	1 Mbps	years	P2P / Star / Mesh / Tree	proprietary	Garmin
EnOcean [17]	sub-1 GHz	30 – 300 m	125 kbps	Self-powered (energy harvesting)	Mesh	proprietary	EnOcean Alliance
Z-Wave [18]	sub-1 GHz	40 – 200 m	100 kbps	months to years	Mesh	proprietary	Z-Wave Alliance
Insteon [19]	sub-1 GHz	30 – 50 m	37.5 kbps	months to years	Mesh	proprietary	Smartlabs
ZigBee [20]	sub-1 GHz, 2.4 GHz	10 – 100 m	250 kbps	months to years	Star / Mesh / Tree	open standard	ZigBee Alliance
MiWi [21]	sub-1 GHz, 2.4 GHz	10 – 100 m	250 kbps	months to years	Star / Mesh / Tree	proprietary	Microchip Technology
DigiMesh [22]	sub-1 GHz, 2.4 GHz	10 – 100 m	250 kbps	years	P2P mesh	proprietary	Digi International
WirelessHART [23]	sub-1 GHz, 2.4 GHz	10 – 100 m	250 kbps	years	Mesh	open standard	HART Communication Foundation
Thread [24]	sub-1 GHz, 2.4 GHz	10 – 100 m	250 kbps	months to years	Star / Mesh / Tree	open standard	Thread Group Alliance
6LowPAN [25]	sub-1 GHz, 2.4 GHz	10 – 100 m	250 kbps	months to years	Star / Mesh / Tree	open standard	IETF
Wi-Fi [26]	sub-1 GHz, 2.4 GHz, 5 GHz	100 m; 1 km	Mbps to Gbps	days to months	Star	open standard	Wi-Fi Alliance
NB-IoT [27]	450 MHz – 3.5 GHz	10 – 15 km	250 kbps	10+ years	Star	open standard	3GPP
eMTC [28]	450 MHz – 3.5 GHz	10 – 15 km	1 Mbps	10+ years	Star	open standard	3GPP
EC-GSM-IoT [29]	850 – 900 MHz, 1800 – 1900 MHz	10 – 15 km	70 – 240 kbps	10+ years	Star	open standard	3GPP
LoRaWAN [30]	sub-1 GHz	10 – 15 km	50 kbps	10+ years	Star of stars	open standard	LoRa Alliance
Symphony Link [31]	sub-1 GHz	10 – 15 km	50 kbps	10+ years	Star	proprietary	Link labs
Weightless [32]	sub-1 GHz (-N and -P), TV white space spectrum (-W)	2 – 5 km	100 kbps (-N and -P), 10 Mbps (-W)	3 – 10 years	Star	open standard	Weightless SIG
SIGFOX [33]	sub-1 GHz	10 – 50 km	100 bps	10+ years	Star	proprietary	Sigfox
DASH7 [34]	sub-1 GHz	2 – 5 km	167 kbps	10+ years	Star / Tree	open standard	Dash7 Alliance

Comparison IoT Protocol Stack and Web Protocol Stack



Major Components of an IoT Ecosystem

The following diagram depicts the major components that make up an IoT system. The for this course is on two of the parts that make the actuators, sensors and processing.



Sensors

A sensor does exactly what the name suggests. A sensor is a device that detects and responds to some type of input from the physical environment. The specific input could be light, heat, motion, moisture, pressure, or any one of a great number of other environmental phenomena. In the broadest definition, a sensor is a device, module, or subsystem whose purpose is to detect events or changes in its environment and send the information to other electronics, frequently a computer processor. A sensor is always used with other electronics, whether as simple as a light or as complex as a computer. A sensor can either be analog or digital.

There are many different types of sensors to measure all kinds of quantities.

Position - A position sensor measures the position of an object; the position of an object can either be in absolute terms or in relative terms e.g. inclinometer, proximity.

Occupancy and Motion Sensors- Occupancy sensors detect the presence of objects in a room, motion sensors detect movement. E.g. radar.

Velocity and acceleration Sensors- Detect velocity and acceleration e.g. accelerometer, gyroscope.

Force -e.g. force gauge, tactile sensor (sound sensor)

Pressure Sensor- Pressure sensors are related to force sensors; they measure force applied by liquids or gases e.g. barometer.

Flow Sensors- Detect the rate of fluid flow e.g. water meter.

Acoustic –Measure sound levels and convert that information into digital or analog signals e.g. microphone

Humidity Sensors- Detect the amount of water vapor in the air or a mass e.g. hygrometer, soil moisture sensor.

Light- detect the presence of light e.g. infrared sensor, photodetector.

Radiation Sensors- detect radiation in the environment. e.g. neutron detector.

Temperature Sensor- Measure the amount of heat or cold present in the environment. e.g. thermometer.

Chemical Sensor-measure the concentration of chemicals in a system. e.g. smoke detector.

Biosensors- detect various biological elements such as organisms, tissues, cells, enzymes, antibodies and nucleic acid.

Actuators

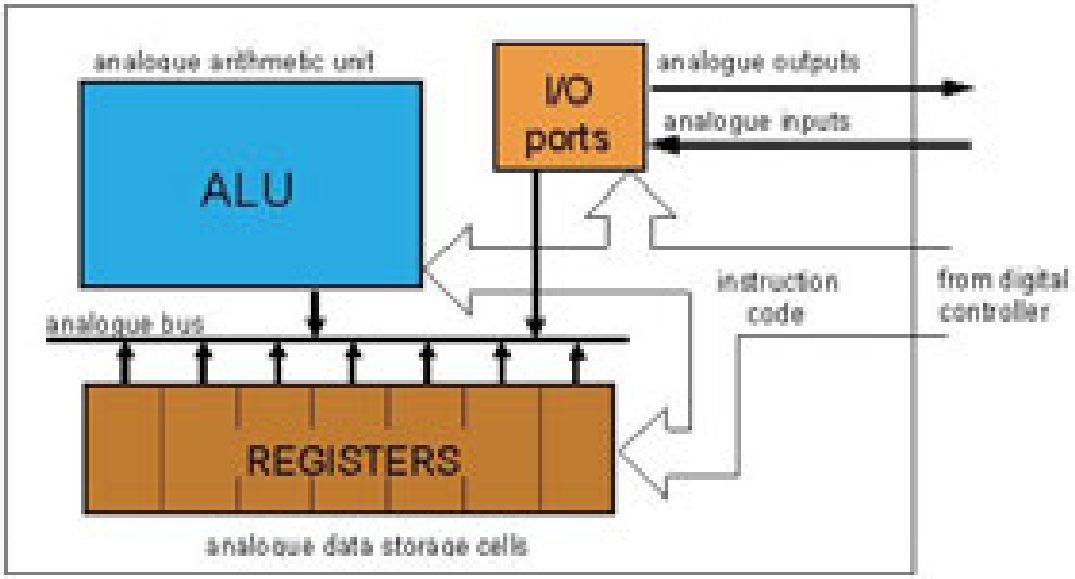
Actiators receive some type of control signal (commonly an electric signal or digital command) that triggers a physical effect such usually some type of motion or force.

Processing Unit

This is the part of the thing responsible for acquiring data, processing and analyzing sensing information received by the sensors, coordinating control signals to any actuators and controlling a variety of functions on the smart objects, including the communication and power systems. The specific type of processing unit that is used can vary greatly depending on the specific processing needs of different applications. The processing unit is of two types: microprocessors and microcontrollers. The most common is a microcontroller because of its small form factor, flexibility, programming simplicity, ubiquity, low power consumption and low cost.

Microprocessor

Microprocessor is the Central Processing Unit (CPU) of a microcomputer. It performs arithmetic and logic operations such as system control and data storage, etc. The microprocessors are configured in microchips; they are made with small transistors and other circuit elements on a solitary solid-state IC. It is abbreviated to "µP" or "uP". The first commercial microprocessor was released by Intel in November 1971 and named 4004; it was a 4-bit microprocessor. From the image of architecture of microprocessor below, it can be easily seen that it have registers and ALU as processing unit and it does not have RAM, ROM in it.



Microcontroller

A microcontroller is like a small computer on a single IC. A microcontroller comprises components like – memory, peripherals and most importantly a processor. The microcontrollers are like small computers in which a CPU, a memory unit like RAM and ROM, I/O devices, timers, counters, are incorporated in an integrated circuit, i.e. IC Microcontrollers are generally used in projects and applications that require direct control of user. As it has all the components needed in its single chip, it does not need any external circuits to do its task so microcontrollers are heavily used in embedded systems.Some examples of popular microcontrollers are 8051, AVR, PIC series of microcontrollers.

A microcontroller can be called the heart of an embedded system. Microcontroller is a compressed microcomputer manufactured to control the functions of embedded systems in office machines, robots, home appliances, motor vehicles, and a number of other gadgets. Microcontrollers are basically employed in devices that need a degree of control to be applied by the user of the device.

Microcontrollers are easily interfaced with external devices such as serial ports, ADC, DAC, Bluetooth, Wi-Fi, etc. Here, the interfacing process is faster than the microprocessor interface. In most cases, microcontrollers use the RISC or CISM architecture to perform a task on different machines. The different types of microcontrollers are:

- 8-bit microcontroller
- 16-bit microcontroller
- 32-bit microcontroller
- Built-in microcontroller

Once the microcontroller is programmed, it can operate autonomously and can perform operations or activities as and when it is requested. It is built to be self-satisfying and profitable. The microcontrollers are designed to perform particular operations that help to control particular systems. A microcroller can be abbreviated as "uC", "µC" or "MCU".

Major Components of Microcontrollers

Any electric appliance that stores, measures, displays information or calculates have microcontroller chip inside it. The basic structure of a microcontroller comprise of:-

a) CPU –Microcontrollers brain is named as CPU. CPU is the device which is employed to fetch data, decode it and at the end complete the assigned task successfully. With the help of

CPU all the components of microcontroller is connected into a single system. Instruction fetched by the programmable memory is decoded by the CPU.

b) Memory – In a microcontroller memory chip works same as **microprocessor**. Memory chip stores all programs & data. Microcontrollers are built with certain amount of ROM or RAM (EPROM, EEPROM, etc) or flash memory for the storage of program source codes.

c) Input/output ports –I/O ports are basically employed to interface or drive different appliances such as- printers, LCD's, LED's, etc.

d) Serial Ports – These ports give serial interfaces amid microcontroller & various other peripherals such as parallel port.

e) Timers –A microcontroller may be in-built with one or more timer or counters. The timers & counters control all counting & timing operations within a microcontroller. Timers are employed to count external pulses. The main operations performed by timers are- pulse generations, clock functions, frequency measuring, modulations, making oscillations, etc.

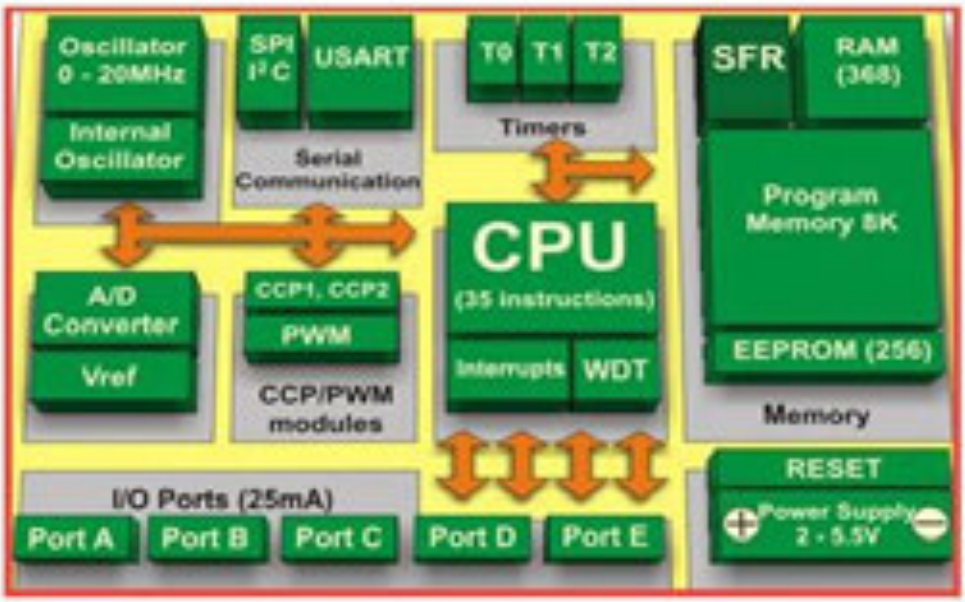
f) ADC (Analog to digital converter) –ADC is employed to convert analog signals to digital ones. The input signals need to be analog for ADC. The digital signal production can be employed for different digital applications (such as- measurement gadgets).

g) DAC (digital to analog converter) – this converter executes opposite functions that ADC perform. This device is generally employed to supervise analog appliances like- DC motors, etc.

h) Interpret Control - This controller is employed for giving delayed control for a working program. The interpret can be internal or external.

i) Special Functioning Block –Some special microcontrollers manufactured for special appliances like- space systems, robots, etc, comprise this special function block. This special block has additional ports so as to carry out some special operations.

A PIC microcontroller architecture is shown below.



How Is a Microcontroller Different from A Microprocessor?

Microprocessor has only a CPU inside them in one or few Integrated Circuits. Unlike microcontrollers it does not have RAM, ROM and other peripherals. They are dependent on external circuits of peripherals to work. Microprocessors are not made for specific task but they are required where tasks are complex like development of softwares and other applications that require high memory and where input and output are not defined. It may be called heart of a computer system. Some examples of microprocessor are Pentium, I3, and I5 etc.

The notable differences are listed below:

- Key difference in both of them is presence of external peripheral, where microcontrollers have RAM, ROM, EEPROM embedded in it while a microprocessor will have to use external circuits.
- As all the peripheral of microcontroller are on single chip it is compact while microprocessor is bulky.
- Microcontrollers are made by using complementary metal oxide semiconductor technology so they are far cheaper than microprocessors. In addition, the applications made with microcontrollers are cheaper because they need lesser external components, while the overall cost of systems made with microprocessors are high because of the high number of external components required for such systems.

- Processing speed of microcontrollers is about 8 MHz to 50 MHz, but in contrary processing speed of general microprocessors is above 1 GHz so it works much faster than microcontrollers.
- Generally microcontrollers have power saving system, like idle mode or power saving mode so overall it uses less power and also since external components are low overall consumption of power is less. While in microprocessors generally there is no power saving system and also many external components are used with it, so its power consumption is high in comparison with microcontrollers.
- Microcontrollers are compact so it makes them a favorable and efficient system for small products and applications while microprocessors are bulky so they are preferred for larger applications.
- Tasks performed by microcontrollers are limited and generally less complex. While task performed by microprocessors are software development, Game development, website, documents making etc. which are generally more complex so require more memory and speed so that's why external ROM, RAM are used with

Classification of Microprocessors

Microprocessors can be classified according to their instruction set as follows:

- **Complex Instruction Set Microprocessors:** They are also called as CISM in short and they categorize a microprocessor in which orders can be executed together along with other low level activities. It mainly performs the task of uploading, downloading and recalling data into and from the memory card. Apart from that it also does complex mathematical calculations within a single command.
- **Reduced Instruction Set Microprocessor:** This processor is also called as RISC. These kinds of chips are made according to the function in which the microprocessor can carry out small things within a particular command. In this way it completes more commands at a faster rate.
- **Superscalar Processors:** This is a processor that copies the hardware on the microprocessor for performing numerous tasks at a time. They can be used for arithmetic and as multipliers. They have several operational units and thus carry out more than a one command by constantly transmitting various instructions to the superfluous operational units inside the processor.

- **The Application Specific Integrated Circuit:** This processor is also known as ASIC. They are used for specific purposes that comprises of automotive emissions control or personal digital assistant computer. This kind of processor is made with proper specification but apart from that it can also be made using the off the shelf gears.
- **Digital Signal Multiprocessors:** Also called as DSP's, these a decoding videos or to convert the digital and video to analog and analog to digital. They need a microprocessor that is excellent in mathematical calculations. The chips of this processor are employed in SONAR, RADAR, home theaters audio gears, Mobile phones and TV set top boxes.

Now we go back to our discussion on microcontrollers and we will focus on AVR microcontrollers.

Types of Microcontrollers

Microcontrollers are divided into categories according to their memory, architecture, bits and instruction sets microcontrollers. So let's- discuss types of microcontrollers.

Bits

- 8 bits microcontroller executes logic & arithmetic operations. Examples of 8 bits micro controller is Intel 8031/8051.
- 16 bits microcontroller executes with greater accuracy and performance in contrast to 8-bit. Example of 16 bit microcontroller is Intel 8096.
- 32 bits microcontroller is employed mainly in automatically controlled appliances such as office machines, implantable medical appliances, etc. It requires 32-bit instructions to carry out any logical or arithmetic function.

Memory

- External Memory Microcontroller –When an embedded structure is built with a microcontroller which does not comprise of all the functioning blocks existing on a chip it is named as external memory microcontroller. For illustration- 8031 microcontroller does not have program memory on the chip.
- Embedded Memory Microcontroller –When an embedded structure is built with a microcontroller which comprise of all the functioning blocks existing on a chip it is named as embedded memory microcontroller. For illustration- 8051 microcontroller has all program & data memory, counters & timers, interrupts, I/O ports and therefore its embedded memory microcontroller.

Instruction Set

CISC- CISC means complex instruction set computer, it allows the user to apply 1 instruction as an alternative to many simple instructions. RISC- RISC means Reduced Instruction Set Computers. RISC reduces the operation time by shortening the clock cycle per instruction.

RISC

Reduced instruction set Computer. It is a type of microprocessor that has been designed to carry out few instructions at the same time. As instructions are few it can be executed in a less amount of time. Another advantage is the use of fewer transistors reducing its cost.

Features include:

- Demand less decoding
- Uniform instruction set
- Identical general purpose register
- Simple addressing modes
- Fewer data types in hardware

CISC

Complex instruction set computer. It's actually a CPU designed to carry out many operation in a single in a single instruction. These can be loading from and to memory and performing mathematical operation etc.

Features include:

- Complex instruction
- More number of addressing modes
- Highly Pipelined
- More data types in hardware

Memory Architecture

- Harvard Memory Architecture Microcontroller
- Princeton Memory Architecture Microcontroller

AVR Microcontrollers

AVR also known as Advanced Virtual RISC, is a customized Harvard architecture 8-bit RISC solitary chip micro-controller. It was invented in the year 1966 by Atmel. Harvard architecture signifies that program & data are amassed in different spaces and are used simultaneously. It was one of the foremost micro-controller families to employ on-chip flash memory basically for storing program, as contrasting to one-time programmable EPROM, EEPROM or ROM, utilized by other micro-controllers at the same time. Flash memory is a non-volatile (constant on power down) programmable memory.

The SRAM, Flash and EEPROM all are incorporated on a single chip, thereby eliminating the requirement of any other external memory in maximum devices. Several appliances comprise of parallel external bus alternative, so as to add extra data memory gadgets. Approximately all appliances, except TinyAVR chips comprise serial interface, which is used to link large serial Flash & EEPROMs chips.

AVR microcontrollers find many applications as embedded systems; they are also used in the Arduino line of open source board designs. And now we can begin our discussion of the Arduino Platform, but before then, we need to discuss interfacing.

Arduino and Raspberry Pi

Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Most Arduino boards are based on AVR microcontrollers. 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 , and the Arduino Software (IDE).

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.

All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

Advantages of Arduino

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux.

Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50

Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

Raspberry Pi

Raspberry Pi is a fully functional computer, a system-on-chip device (SoC), which runs on a Linux operating system specially designed for it, called Raspbian. Raspbian is the official operating system for Raspberry Pi, where other third-party operating systems like Firefox OS, Android, RISC OS, Ubuntu Mate etc. can be installed. The version for Windows 10 is also available for Pi. Like a computer, it has memory, processor, USB ports, audio output, graphic driver for HDMI output and, since it runs on Linux, most Linux software applications can be installed on it. It has several models and revisions like Raspberry Pi, Raspberry Pi 2, Raspberry Pi 2 Model B + and Raspberry Pi 3.

Raspberry Pi is an inexpensive credit-card sized computer originally developed to promote teaching computer science basics to school children [25]. Like Arduino, the Raspberry Pi encourages experimenting with its hardware configuration. The purpose behind the development of the Raspberry Pi project was to create a replacement for an increasingly complex ‘closed box’ computer that would encourage kids to code and tinker with it. The Raspberry Pi includes a processor, system memory, network interfaces, and a memory card slot, along with the ports for attaching sensors, peripherals, and other devices. A number of add-on board, including the Sensorian shield [18] and the Sense HAT (http://www.raspberrypi.org/products/sense-hat), combining several types of sensors have been developed for the Raspberry Pi. Typical setups include connecting Raspberry Pi to the same kind of peripherals that can be attached to a traditional desktop, or remotely logging in to it from another computer, in which case Raspberry Pi does not need to be connected to anything other than a power supply and a wired or wireless network. Raspberry Pi runs Linuxbased Raspbian, which includes a number of programming environments such as Python, Scratch, BlueJ, and Greenfoot [14]. The Raspberry Pi platform is a natural fit for implementing the sensing and access layers of the IoT system architecture. In the case of designing a single-device IoT system, Raspberry Pi has enough flexibility and processing power to implement the service layer by running data- and sensor-driven applications on the device. At the same time, its IO and networking capabilities would easily allow one to extend it in order to implement the interface layer to make these applications easily accessible by the users and its services accessible by remote systems.

Raspberry pi 1 Model B+

Raspberry pi 2 model B

Raspberry pi 3 model B+

Raspberry pi 3 model B

Raspberry pi zero

Raspberry pi zero W

Arduino vs Raspberry Pi

Raspberry Pi is a fully functional computer, a system-on-chip device (SoC), which runs on a Linux operating system specially designed for it, called Raspbian. Raspbian is the official operating system for Raspberry Pi, where other third-party operating systems like Firefox OS, Android, RISC OS, Ubuntu Mate etc. can be installed. The version for Windows 10 is also available for Pi. Like a computer, it has memory, processor, USB ports, audio output, graphic driver for HDMI output and, since it runs on Linux, most Linux software applications can be installed on it. It has several models and revisions like Raspberry Pi, Raspberry Pi 2, Raspberry Pi 2 Model B + and Raspberry Pi 3.

Arduino, on the other hand, is a microcontroller, which is not as powerful as Raspberry Pi and can be considered as a component on the computer. But it's a great hardware for electronics projects. It does not need OS and software applications to work, we just need to write a few lines of code to use of it. There are many Arduino boards such as Arduino UNO, Arduino PRO, Arduino MEGA, Arduino DUE etc. Although they are quite different there are some similarities.

Both were invented in European countries, as Raspberry Pi is developed by Eben Upton in the United Kingdom and Arduino is developed by Massimo Banzi in Italy. Both inventors are teachers and develop these hardware platforms as a design learning tool for their students. Raspberry Pi was introduced for the first time in the year 2012 while Arduino in 2005. A key aspect of an IOT

system will be the integration of the OT with corporate IT systems. In fact, most implementations of Industry 4.0 will need to integrate with existing SCM, PLM, MES and / or ERP systems. To achieve this level of integration, OT data will need to be filtered, aggregated and standardized for IT integration. Hence, large-scale event processing becomes a key component for the success of enterprise IT integration.

Interfacing

Embedded electronics is all about interlinking circuits (processors or other integrated circuits) to create a symbiotic system. In order for those individual circuits to swap their information, they must share a common communication protocol. Hundreds of communication protocols have been defined to achieve this data exchange, and, in general, each can be separated into one of two categories: parallel or serial.

Parallel vs. Serial

Parallel interfaces transfer multiple bits at the same time. They usually require buses of data - transmitting across eight, sixteen, or more wires. Data is transferred in huge, crashing waves of 1's and 0's.

Serial interfaces stream their data, one single bit at a time. These interfaces can operate on as little as one wire, usually never more than four.

Parallel communication certainly has its bene to implement. But it requires many more input/output (I/O) lines

Synchronous Vs Asynchronous Serial Communication

Over the years, dozens of serial protocols have been crafted to meet particular needs of embedded systems. USB (universal serial bus), and Ethernet, are a couple of the more well-known computing serial interfaces. Other very common serial interfaces include SPI, I2C and UART. Each of these serial interfaces can be sorted into one of two groups: synchronous or asynchronous.

Asynchronous means that data is transferred without support from an external clock signal. This transmission method is perfect for minimizing the required wires and I/O pins, but it does mean

we need to put some extra effort into reliably transferring and receiving data. UART is the most common form of asynchronous transfers.

UART

UART stands for Universal Asynchronous Receiver/Transmitter and is really just a fancy way of referring to a serial port. It is really easy to understand as it only requires two lines: a transmission line (TX) and a receiving line (RX).

UART transmissions begin with a start bit where the appropriate line (TX or RX) is pulled low by the sending party. Then 5 to 8 data bits are sent.

Following the data, an optional parity bit is sent, followed by 1 or 2 stop bits, where the sending module pulls the pin high.

For this protocol to work, the sender and receiver have to agree on a few things.

- How many data bits are sent with each packet (5 to 8)?
- How fast should the data be sent? This is known as the baud rate.
- Is there a parity bit after the data, and is it high or low?
- How many stop bits will be sent at the end of each transmission?

A synchronous serial interface always pairs its data line(s) with a clock signal, so all devices on a synchronous serial bus share a common clock. This makes for a more straightforward, often faster serial transfer, but it also requires at least one extra wire between communicating devices. Examples of synchronous interfaces include SPI, and I2C.

SPI

SPI stands for Serial Peripheral Interface. SPI is a common communication protocol used by many different devices. For example, SD card modules, RFID card reader modules, and 2.4 GHz wireless transmitter/receivers all use SPI to communicate with microcontrollers.

One unique benefit of SPI is the fact that data can be transferred without interruption. Any number of bits can be sent or received in a continuous stream. With I2C and UART, data is sent in packets, limited to a specific number of bits. Start and stop conditions define the beginning and end of each packet, so the data is interrupted during transmission.

Devices communicating via SPI are in a master-slave relationship. The master is the controlling device (usually a microcontroller), while the slave (usually a sensor, display, or memory chip) takes instruction from the master. The simplest configuration of SPI is a single master, single slave system, but one master can control more than one slave (more on this below).

MOSI (Master Output/Slave Input) –Line for the master to send data to the slave.

MISO (Master Input/Slave Output) –Line for the slave to send data to the master.

SCLK (Clock) –Line for the clock signal.

SS/CS (Slave Select/Chip Select) –Line for the master to select which slave to send data to.

Advantages of SPI

- No start and stop bits, so the data can be streamed continuously without interruption
- No complicated slave addressing system like I2C
- Higher data transfer rate than I2C (almost twice as fast)
- Separate MISO and MOSI lines, so data can be sent and received at the same time

Disadvantages of SPI

- Uses four wires (I2C and UARTs use two)
- No acknowledgement that the data has been successfully received (I2C has this)
- No form of error checking like the parity bit in UART
- Only allows for a single master

I2C

I2C stands for Inter-Integrated Circuitared and C", is pronounced"I two C". I2**C** is a protocol that allows one device to exchange data with one or more connected devices through the use of a single data line and clock signal. I2C combines the best features of SPI and UARTs. With I2C, you can connect multiple slaves to a single master (like SPI) and you can have multiple masters controlling single, or multiple slaves. This is really useful when you want to have more than one microcontroller logging data to a single memory card or displaying text to a single LCD.

Like UART communication, I2C only uses two wires to transmit data between devices:

SDA (Serial Data) –The line for the master and slave to send and receive data.

SCL (Serial Clock) –The line that carries the clock signal.

I2C is a serial communication protocol, so data is transferred bit by bit along a single wire (the SDA line).

Like SPI, I2C is synchronous, so the output of bits is synchronized to the sampling of bits by a clock signal shared between the master and the slave. The clock signal is always controlled by the master. Since multiple slave devices can use the same SDA line, the master needs a way to distinguish between them and talk to a single device at a time. The I2C protocol uses the concept of device addressing to coordinate traffic on the data line.

Advantages of I2C

- Only uses two wires
- Supports multiple masters and multiple slaves
- ACK/NACK bit gives confirmation that each frame is transferred successfully
- Hardware is less complicated than with UARTs
- Well known and widely used protocol

Disadvantages of I2C

- Slower data transfer rate than SPI
- The size of the data frame is limited to 8 bits
- More complicated hardware needed to implement than SPI

Communication Technologies in IoT

Bluetooth

An important short-range communications technology is of course Bluetooth, which has become very important in computing and many consumer product markets. It is expected to be key for wearable products in particular, again connecting to the IoT albeit probably via a smartphone in many cases. The new Bluetooth Low-Energy (BLE) – or Bluetooth Smart, as it is now branded – is a significant protocol for IoT applications. Importantly, while it offers similar range to Bluetooth it has been designed to offer significantly reduced power consumption.

However, Smart/BLE is not really designed for file transfer and is more suitable for small chunks of data. It has a major advantage certainly in a more personal device context over many competing technologies given its widespread integration in smartphones and many other mobile devices. According to the Bluetooth SIG, more than 90 percent of Bluetooth-enabled smartphones, including iOS, Android and Windows based models, are expected to be 'Smart Ready' by 2018.

Devices that employ Bluetooth Smart features incorporate the Bluetooth Core Specification Version 4.0 (or higher – the latest is version 4.2 announced in late 2014) with a combined basic-data-rate and low-energy core configuration for a RF transceiver, baseband and protocol stack. Importantly, version 4.2 via its Internet Protocol Support Profile will allow Bluetooth Smart sensors to access the Internet directly via 6LoWPAN connectivity (more on this below). This IP connectivity makes it possible to use existing IP infrastructure to manage Bluetooth Smart 'edge' devices. More information on Bluetooth 4.2 is available here and a wide range of Bluetooth modules are available from RS.

- Standard: Bluetooth 4.2 core specification
- Frequency: 2.4GHz (ISM)
- Range: 50-150m (Smart/BLE)
- Data Rates: 1Mbps (Smart/BLE)

Zigbee

ZigBee, like Bluetooth, has a large installed base of operation, although perhaps traditionally more in industrial settings. ZigBee PRO and ZigBee Remote Control (RF4CE), among other available ZigBee profiles, are based on the IEEE802.15.4 protocol, which is an industry-standard wireless networking technology operating at 2.4GHz targeting applications that require relatively infrequent data exchanges at low data-rates over a restricted area and within a 100m range such as in a home or building.

ZigBee/RF4CE has some significant advantages in complex systems offering low-power operation, high security, robustness and high scalability with high node counts and is well positioned to take advantage of wireless control and sensor networks in M2M and IoT applications. The latest version of ZigBee is the recently launched 3.0, which is essentially the unification of the various ZigBee wireless standards into a single standard.

- Standard: ZigBee 3.0 based on IEEE802.15.4
- Frequency: 2.4GHz
- Range: 10-100m
- Data Rates: 250kbps

Z-Wave

Z-Wave is a low-power RF communications technology that is primarily designed for home automation for products such as lamp controllers and sensors among many others. Optimized for

reliable and low-latency communication of small data packets with data rates up to 100kbit/s, it operates in the sub-1GHz band and is impervious to interference from WiFi and other wireless technologies in the 2.4-GHz range such as Bluetooth or ZigBee. It supports full mesh networks without the need for a coordinator node and is very scalable, enabling control of up to 232 devices. Z-Wave uses a simpler protocol than some others, which can enable faster and simpler development, but the only maker of chips is Sigma Designs compared to multiple sources for other wireless technologies such as ZigBee and others.

- Standard: Z-Wave Alliance ZAD12837 / ITU-T G.9959
- Frequency: 900MHz (ISM)
- Range: 30m
- Data Rates: 9.6/40/100kbit/s

6LowPAN

A key IP (Internet Protocol)-based technology is 6LowPAN (IPv6 Low-power wireless Personal Area Network). Rather than being an IoT application protocols technology like Bluetooth or ZigBee, 6LowPAN is a network protocol that defines encapsulation and header compression mechanisms. The standard has the freedom of frequency band and physical layer and can also be used across multiple communications platforms, including Ethernet, Wi-Fi, 802.15.4 and sub-1GHz ISM. A key attribute is the IPv6 (Internet Protocol version 6) stack, which has been a very important introduction in recent years to enable the IoT. IPv6 is the successor to IPv4 and offers approximately 5×10^{28}addresses for every person in the world, enabling any embedded object or device in the world to have its own unique IP address and connect to the Internet. Especially designed for home or building automation, for example, IPv6 provides a basic transport mechanism to produce complex control systems and to communicate with devices in a cost-effective manner via a low-power wireless network.

Designed to send IPv6 packets over IEEE802.15.4-based networks and implementing open IP standards including TCP, UDP, HTTP, COAP, MQTT, and websockets, the standard offers end-to-end addressable nodes, allowing a router to connect the network to IP. 6LowPAN is a mesh network that is robust, scalable and self-healing. Mesh router devices can route data destined for other devices, while hosts are able to sleep for long periods of time.

- Standard: RFC6282
- Frequency: (adapted and used over a variety of other networking media including Bluetooth Smart (2.4GHz) or ZigBee or low-power RF (sub-1GHz)
- Range: N/A
- Data Rates: N/A

WiFi

WiFi connectivity is often an obvious choice for many developers, especially given the pervasiveness of WiFi within the home environment within LANs. It requires little further

explanation except to state the obvious that clearly there is a wide existing infrastructure as well as offering fast data transfer and the ability to handle high quantities of data.

A wireless network uses radio waves to communicate with portable devices, granting them access to other connected devices and to the Internet. Depending on the specific type of wireless network you use, Wi-Fi signals travel in two distinct frequency ranges. The 802.11b and g networks use the 2.4 GHz band, while 802.11a networks use 5 GHz and 802.11n networks broadcast on both frequencies to increase throughput.

Currently, the most common WiFi standard used in homes and many businesses is 802.11n, which offers serious throughput in the range of hundreds of megabit per second, which is fine for file transfers, but may be too power-consuming for many IoT applications. A series of RF development kits designed for building WiFi-based applications are available from RS.

- Standard: Based on 802.11n (most common usage in homes today)
- Frequencies: 2.4GHz and 5GHz bands
- Range: Approximately 50m
- Data Rates: 600 Mbps maximum, but 150-200Mbps is more typical, depending on channel frequency used and number of antennas (latest 802.11-ac standard should offer 500Mbps to 1Gbps)

Cellular

Any IoT application that requires operation over longer distances can take advantage of GSM/3G/4G cellular communication capabilities. While cellular is clearly capable of sending high quantities of data, especially for 4G, the expense and also power consumption will be too high for many applications, but it can be ideal for sensor-based low-bandwidth-data projects that will send very low amounts of data over the Internet. A key product in this area is the SparqEE range of products, including the original tiny CELLv1.0 low-cost development board and a series of shield connecting boards for use with the Raspberry Pi and Arduino platforms.

- Standard: GSM/GPRS/EDGE (2G), UMTS/HSPA (3G), LTE (4G)
- Frequencies: 900/1800/1900/2100MHz
- Range: 35km max for GSM; 200km max for HSPA
- Data Rates (typical download): 35-170kps (GPRS), 120-384kbps (EDGE), 384Kbps-2Mbps (UMTS), 600kbps-10Mbps (HSPA), 3-10Mbps (LTE)

NFC

NFC (Near Field Communication) is a technology that enables simple and safe two-way interactions between electronic devices, and especially applicable for smartphones, allowing consumers to perform contactless payment transactions, access digital content and connect electronic devices. Essentially it extends the capability of contactless card technology and enables devices to share information at a distance that is less than 4cm.

- Standard: ISO/IEC 18000-3
- Frequency: 13.56MHz (ISM)
- Range: 10cm
- Data Rates: 100–420kbps

Sigfox

An alternative wide-range technology is Sigfox, which in terms of range comes between WiFi and cellular. It uses the ISM bands, which are free to use without the need to acquire licenses, to transmit data over a very narrow spectrum to and from connected objects. The idea for Sigfox is that for many M2M applications that run on a small battery and only require low levels of data transfer, then WiFi's range is too short while cellular is too expensive and also consumes too much power. Sigfox uses a technology called Ultra Narrow Band (UNB) and is only designed to handle low data-transfer speeds of 10 to 1,000 bits per second. It consumes only 50 microwatts compared to 5000 microwatts for cellular communication, or can deliver a typical stand-by time 20 years with a 2.5Ah battery while it is only 0.2 years for cellular.

Already deployed in tens of thousands of connected objects, the network is currently being rolled out in major cities across Europe, including ten cities in the UK for example. The network offers a robust, power-efficient and scalable network that can communicate with millions of battery-operated devices across areas of several square kilometres, making it suitable for various M2M applications that are expected to include smart meters, patient monitors, security devices, street lighting and environmental sensors. The Sigfox system uses silicon such as the EZRadioPro wireless transceivers from Silicon Labs, which deliver industry-leading wireless performance, extended range and ultra-low power consumption for wireless networking applications operating in the sub-1GHz band.

- Standard: Sigfox
- Frequency: 900MHz
- Range: 30-50km (rural environments), 3-10km (urban environments)
- Data Rates: 10-1000bps

LoRaWAN

LoRaWAN is a media access control (MAC) protocol for wide area networks. It is designed to allow low-powered devices to communicate with Internet-connected applications over long range wireless connections. LoRaWAN can be mapped to the second and third layer of the OSI model. It is implemented on top of LoRa or FSK modulation in industrial, scientific and medical (ISM) radio bands. The LoRaWAN protocols are defined by the LoRa Alliance and formalized in the LoRaWAN Specification which can be requested on the LoRa Alliance website.

Terminology

- **End Device, Node, Mote** - an object with an embedded low-power communication device.
- **Gateway** - antennas that receive broadcasts from End Devices and send data back to End Devices.
- **Network Server** - servers that route messages from End Devices to the right Application, and back.
- **Application** - a piece of software, running on a server.
- **Uplink Message** - a message from a Device to an Application.
- **Downlink Message** - a message from an Application to a Device

Again, similar in some respects to Sigfox and Neul, LoRaWAN targets wide-area network (WAN) applications and is designed to provide low-power WANs with features specifically needed to support low-cost mobile secure bi-directional communication in IoT, M2M and smart city and industrial applications. Optimized for low-power consumption and supporting large networks with millions and millions of devices, data rates range from 0.3 kbps to 50 kbps.

- Standard: LoRaWAN
- Frequency: Various
- Range: 2-5km (urban environment), 15km (suburban environment)
- Data Rates: 0.3-50 kbps.

Neul

Similar in concept to Sigfox and operating in the sub-1GHz band, Neul leverages very small slices of the TV White Space spectrum to deliver high scalability, high coverage, low power and low-cost wireless networks. Systems are based on the Iceni chip, which communicates using the white space radio to access the high-quality UHF spectrum, now available due to the analogue to digital TV transition. The communications technology is called Weightless, which is a new wide-area wireless networking technology designed for the IoT that largely competes against existing GPRS, 3G, CDMA and LTE WAN solutions. Data rates can be anything from a few bits per second up to 100kbps over the same single link; and devices can consume as little as 20 to 30mA from 2xAA batteries, meaning 10 to 15 years in the field.

- Standard: Neul
- Frequency: 900MHz (ISM), 458MHz (UK), 470-790MHz (White Space)
- Range: 10km
- Data Rates: Few bps up to 100kbps

Thread

A very new IP-based IPv6 networking protocol aimed at the home automation environment is Thread. Based on 6LowPAN, and also like it, it is not an IoT applications protocol like Bluetooth or ZigBee. However, from an application point of view, it is primarily designed as a complement to WiFi as it recognises that while WiFi is good for many consumer devices that it has limitations for use in a home automation setup.

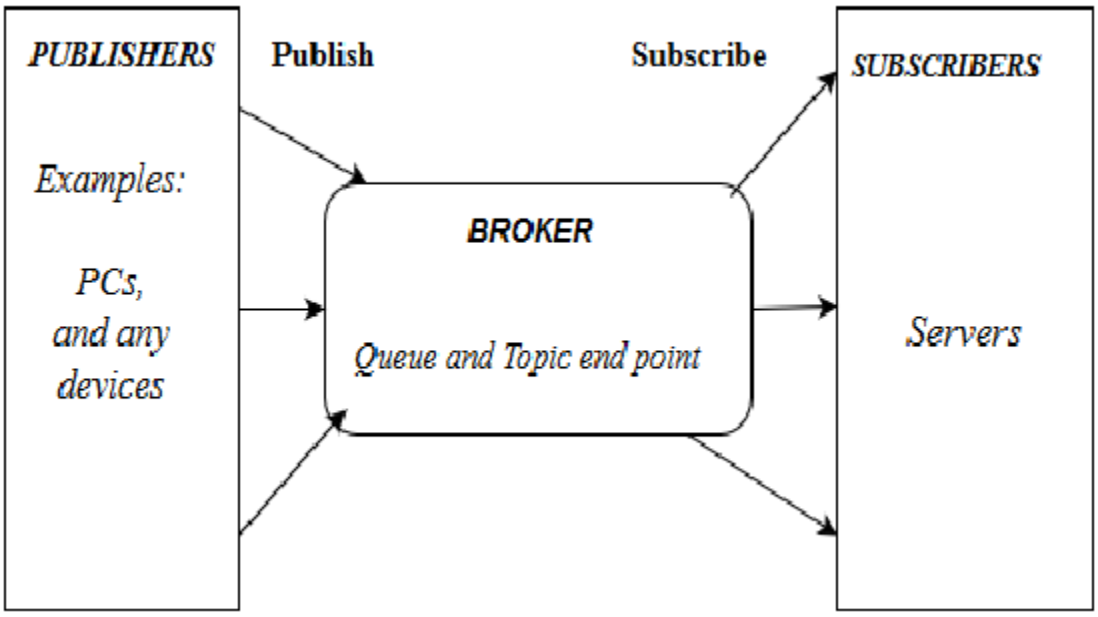
Launched in mid-2014 by the Thread Group, the royalty-free protocol is based on various standards including IEEE802.15.4 (as the wireless air-interface protocol), IPv6 and 6LoWPAN, and offers a resilient IP-based solution for the IoT. Designed to work on existing IEEE802.15.4 wireless silicon from chip vendors such as Freescale and Silicon Labs, Thread supports a mesh network using IEEE802.15.4 radio transceivers and is capable of handling up to 250 nodes with high levels of authentication and encryption. A relatively simple software upgrade should allow users to run thread on existing IEEE802.15.4-enabled devices.

- Standard: Thread, based on IEEE802.15.4 and 6LowPAN
- Frequency: 2.4GHz (ISM)
- Range: N/A
- Data Rates: N/A

IoT Application Layer Protocols

MQTT

MQTT (Message Queue Telemetry Transport) was develop by or introduce by IBM in 1999 and standardized by OASIS in 2013 to target come up with lightweight M2M communication. It is publish/subscribe protocol architecture similar to client/server protocol show in figure below. The importance of MQTT protocol is due to its simplicity and the no need of high CPU and memory usage (reason is the lightweight protocol). MQTT supports a wide range of different devices and mobile platforms. At transport layer TLS/SSL security provide to MQTT.



Show above figure there three component are there publishers, a broker and subscribers. Publishers are generally lightweight sensors that connect with a broker and send data to a broker and go back to sleep. Subscribers are IoT applications that interested in data send by sensors and also connect with a broker, so broker send interested data to subscribers. The brokers classify sensory data in topics and send them to subscribers interested in the topics. This all thing is on IoT point of view.

MQTT provide 3 option to achieve message in Quality of Services (QoS):

1. One delivery (at most):

Deliver message according to best try of the network. An acknowledgment is not required. Lowest level of QoS.

2. One delivery (at least):

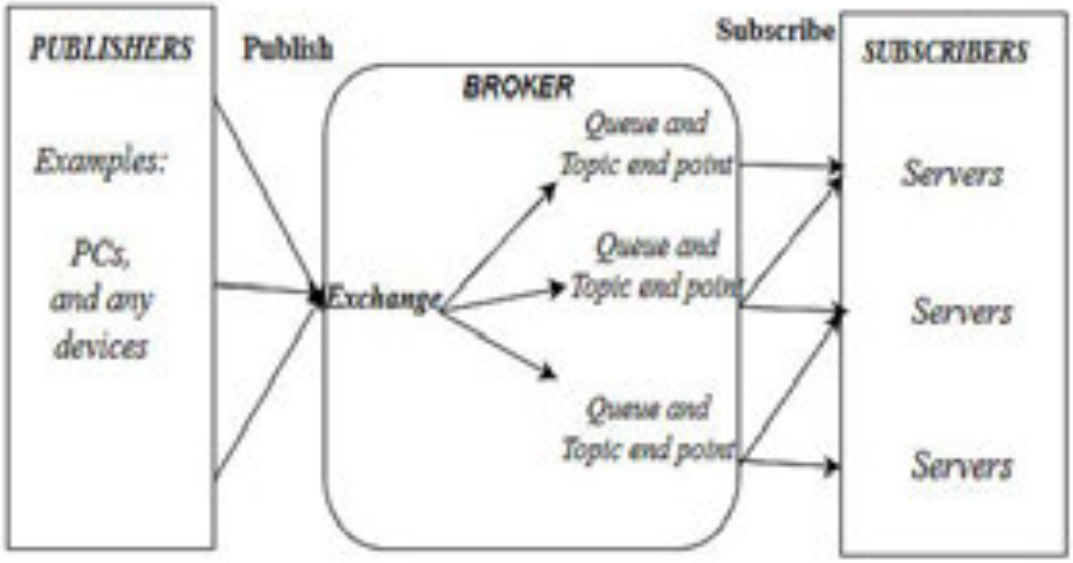
At least one message can be send and some duplicate message are there. An acknowledgment is required

3. On delivery:

Additional protocol required to ensure that one and only one message send. It is highest level of QoS.

AMQP

The Advanced Message Queuing Protocol (AMQP) is a protocol that across from the financial industry. Security is manage with the use of the TLS/SSL protocols. Its run over TCP. AMQT is follow publish/subscribe communication protocol for messaging [6]. AMQP is same like MQTT but AMQT have advantage its store data then forward it, and this feature used at when network disruptions that time ensures reliability. Show in figure below a broker divide into two part exchange and queue. Exchange responsibility to receive publishers messages and distribute to queue. Queues is based on pre-define roles and condition and it’s basically send message to subscribers who subscribe those data.



CoAP

CoAP (constrained application protocol) is used for low power and low memory embedded devices where it can be used for communication instead of HTTP. Currently there is HTTP protocol available with request/response paradigm but HTTP has many features and more footprint [5]. HTTP runs over TCP where TCP will need more resources due to three way handshake and many more complex mechanism. Now for low power embedded devices, there is no need of this heavy protocols and we can optimize it to run over TCP. As CoAP is a Restful web transfer protocol for use with constrained networks. CoAP uses client/server model of approach same as HTTP. It is designed for constrained networks with low overhead and lower footprint. Some points for CoAP that makes better protocol compared to HTTP is:

• CoAP runs over UDP (User data-gram protocol) that helps to avoid costly TCP handshake before data transmission

• CoAP protocol is only 4-byte header and provides reliable transfer and no reliable transfer as it uses four types of messages.

Show above figure, its support four types of message 1) Confirmable, 2) Non-Confirmable, 3) Acknowledgement and 4) Reset. Request/Response layer used those message and classified in 1) Piggybacked, 2) Separate response, 3) Non confirmable request and response, and communicate with each other show in below figure.

As in HTTP, CoAP utilizes GET, PUT, PUSH, DELETE messages requests to retrieve, create, update, and delete, respectively

RESTFUL Services

Representational State Transfer (RESTFUL Services) is an engineering that gives web administrations which permit correspondence and information trade between various gadgets utilizing HTTP in IoT condition. REST utilizes the HTTP strategies GET, POST, PUT, and DELETE to give an asset arranged informing framework where all activities can be performed essentially by utilizing the synchronous request/response HTTP commands. RESTful services use the secure and reliable HTTP which is the proven worldwide Internet language. It can make use of TLS/SSL for security.

WeB-Socket

The Web-Socket protocol provides two ways for communication between clients and a remote server. WebSocket provides security similar to the security model used HTTPS protocol. For browsing application layer used and web-socket work on TCP transport layer protocol, so they need to interact and communicate with host those who connect with remote. Web-Socket is a web-based protocol that works on the single TCP channel and provides full duplex communications. Web-socket starts session without publish/subscribe and request/response models like previous protocols.

XMPP

Extensible Messaging and Presence Protocol (XMPP) is a messaging protocol that was designed originally for chatting and message exchange applications. It was standardized by IETF more than a decade ago. In all application layer protocols only XMPP protocol support publish/subscribe and request/response model and it’s depend on application developers to develop application which model they us. It does not provide any quality of service guarantees and, hence, is not practical for M2M communications. XMPP is rarely used in IoT but has gained some interest for enhancing its architecture in order to support IoT applications.

DDS

Data Distribution Service (DDS) is another publish/subscribe protocol that is designed by the Object Management Group (OMG) for M2M communications. It defines two sub layers: data-centric publish- subscribe and data-local reconstruction sub layer. The first takes the responsibility of message delivery to the subscribers while the second is optional and allows a simple integration of DDS in the application layer. Publisher layer is responsible for sensory data distribution. Data writer interacts with the publishers to agree about the data and changes to be sent to the subscribers. Subscribers are the receivers of sensory data to be delivered to the IoT application. Data readers basically read the published data and deliver it to the subscribers and the topics are

basically the data that are being published. In others words, data writers and data reader take the responsibilities of the broker in the broker-based architectures.

SMQTT

An extension of MQTT is Secure MQTT (SMQTT) which uses encryption based on lightweight attribute based encryption. The main advantage of using such encryption is the broadcast encryption feature, in which one message is encrypted and delivered to multiple other nodes, which is quite common in IoT applications. In MQTT architecture exchange data between publishers and subscribers SMQTT use encryption algorithm them. DDS and SMQTT both protocols are new emerging protocols, and both are similar with MQTT in show both are upgrades of MQTT.

Comparison of Application Layer Protocols

Protocols	Transport	QoS	Architecture	Security
CoAP	UDP	YES	Request/Response	DTLS
MQTT	TCP	YES	Publish/Subscribe	TLS/SSL
XMPP	TCP	NO	Request/Response Publish/Subscribe	TLS/SSL
RESTFUL	HTTP	NO	Request/Response	HTTPS
AMQP	TCP	YES	Publish/Subscribe	TLS/SSL
Web socket	TCP	NO	Client/Server Publish/Subscribe	TLS/SSL
DDS	TCP/UDP	YES	Publish/Subscribe	TLS/SSL
SMQTT	TCP	YES	Publish/Subscribe	It have own