

INTERNET OF THINGS

COURSE OBJECTIVES:

- Describe what IoT is and how it works in current environment
- Recognize the factors that contributed to the emergence of IoT
- Design and program IoT devices
- Use real IoT protocols for communication

UNIT I: FUNDAMENTALS OF IoT 10

Evolution of Internet of Things, Enabling Technologies, M2M Communication, IoT World Forum (IoTWF) standardized architecture, Simplified IoT Architecture, Core IoT Functional Stack, Fog, Edge and Cloud in IoT, Functional blocks of an IoT ecosystem, Sensors, Actuators, Smart Objects and Connecting Smart Objects.

UNIT II: DESIGN AND DEVELOPMENT 10

Design Methodology, Embedded computing logic, Microcontroller, System on Chips, IoT system building blocks IoT Platform overview: Overview of IoT supported Hardware platforms such as: Raspberry pi, Arduino Board details

UNIT III: CASE STUDIES/INDUSTRIAL APPLICATIONS 10

IoT applications in home, infrastructures, buildings, security, Industries, Home appliances, other IoT electronic equipments, Industry 4.0 concepts.

COURSE OUTCOMES:

Student's will be able to

- ✓ Secure the elements of an IoT device
- ✓ Design an IoT device to work with a Cloud Computing infrastructure.
- ✓ Transfer IoT data to the cloud and in between cloud providers
- ✓ Define the infrastructure for supporting IoT deployments

Text Books:

1. IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things, David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton and Jerome Henry, Cisco Press, 2017
2. Internet of Things – A hands-on approach, Arshdeep Bahga, Vijay Madisetti, Universities Press, 2015
3. Internet of Things: Architecture, Design Principles And Applications, Rajkamal, McGraw Hill HigherEducation

REFERENCES:

1. Vijay Madisetti, Arshdeep Bahga, "Internet of Things A Hands-On- Approach", 2014.
2. Adrian McEwen, "Designing the Internet of Things", Wiley Publishers, 2013
3. Daniel Kellmerit, "The Silent Intelligence: The Internet of Things". 2013.

Unit 1

FUNDAMENTALS OF IoT

1. INTRODUCTION TO IoT

- Today the Internet has become ubiquitous, has touched almost every corner of the globe, and is affecting human life in unimaginable ways.
- We are now entering an era of even more pervasive connectivity where a very wide variety of appliances will be connected to the web.
- One year after the past edition of the Clusterbook 2012 it can be clearly stated that the Internet of Things (IoT) has reached many different players and gained further recognition. Out of the potential Internet of Things application areas, Smart Cities (and regions), Smart Car and mobility, Smart Home and assisted living, Smart Industries, Public safety, Energy & environmental protection, Agriculture and Tourism as part of a future IoT Ecosystem (Figure 1.1) have acquired high attention.



IoT Ecosystem.

- We are entering an era of the “Internet of Things” (abbreviated as IoT). There are 2 definitions: First one is defined by Verma and second by Peña-López
 1. The Internet of Things as simply an interaction between the physical and digital worlds. The digital world interacts with the physical world using a plethora of sensors and actuators.
 2. Another is the Internet of Things is defined as a paradigm in which computing and networking capabilities are embedded in any kind of conceivable object.

- We use these capabilities to query the state of the object and to change its state if possible.
- In common parlance, the Internet of Things refers to a new kind of world where almost all the devices and appliances that we use are connected to a network.
- We can use them collaboratively to achieve complex tasks that require a high degree of intelligence.
- For this intelligence and interconnection, IoT devices are equipped with embedded sensors, actuators, processors, and transceivers.
- IoT is not a single technology; rather it is an agglomeration of various technologies that work together in tandem.
- Sensors and actuators are devices, which help in interacting with the physical environment.
- The data collected by these sensors has to be stored and processed intelligently in order to derive useful inferences from it.
- Note that we broadly define the term *sensor*; a mobile phone or even a microwave oven can count as a sensor as long as it provides inputs about its current state (internal state + environment).
- An *actuator* is a device that is used to effect a change in the environment such as the temperature controller of an air conditioner.
- The storage and processing of data can be done on the edge of the network itself or in a remote server.
- If any preprocessing of data is possible, then it is typically done at either the sensor or some other proximate device.
- The processed data is then typically sent to a remote server.
- The storage and processing capabilities of an IoT object are also restricted by the resources available, which are often very constrained due to limitations of size, energy, power, and computational capability.
- As a result the main research challenge is to ensure that we get the right kind of data at the desired level of accuracy.
- Along with the challenges of data collection, and handling, there are challenges in communication as well.
- The communication between IoT devices is mainly wireless because they are generally installed at geographically dispersed locations.
- The wireless channels often have high rates of distortion and are unreliable.
- In this scenario reliably communicating data without too many retransmissions is an important problem and thus communication technologies are integral to the study of IoT devices.
- We can directly modify the physical world through actuators or we may do something virtually. For example, we can send some information to other smart things.

- The process of effecting a change in the physical world is often dependent on its state at that point of time. This is called *context awareness*. Each action is taken keeping in consideration the context because an application can behave differently in different contexts.
- For example, a person may not like messages from his office to interrupt him when he is on vacation. Sensors, actuators, compute servers, and the communication network form the core infrastructure of an IoT framework. However, there are many software aspects that need to be considered.
- First, we need a middleware that can be used to connect and manage all of these heterogeneous components. We need a lot of standardization to connect many different devices.
- The Internet of Things finds various applications in health care, fitness, education, entertainment, social life, energy conservation, environment monitoring, home automation, and transport systems.

1.2 TECHNOLOGIES INVOLVED IN IOT DEVELOPMENT: INTERNET/WEB AND NETWORKING BASICS OSI MODEL

- Networking technologies enable IoT devices to communicate with other devices, applications, and services running in the cloud.
- The internet relies on standardized protocols to ensure communication between heterogeneous devices is secure and reliable.
- Standard protocols specify rules and formats that devices use to establish and manage networks and transmit data across those networks.
- Networks are built as a “stack” of technologies. A technology such as Bluetooth LE is at the bottom of the stack.
- While others such as IPv6 technologies (which is responsible for the logical device addressing and routing of network traffic) are further up the stack. Technologies at the top of the stack are used by the applications that are running on top of those layers, such as message queuing technologies.
- This article describes widely adopted technologies and standards for IoT networking. It also provides guidance for choosing one network protocol over another. It then discusses key considerations and challenges related to networking within IoT: range, bandwidth, power usage, intermittent connectivity, interoperability, and security.

NETWORKING STANDARDS AND TECHNOLOGIES

- The Open Systems Interconnection (OSI) model is an ISO-standard abstract model is a stack of seven protocol layers.
- From the top down, they are: application, presentation, session, transport, network, data link and physical. TCP/IP, or the Internet Protocol suite, underpins the internet, and it provides a simplified concrete implementation of these layers in the OSI model.

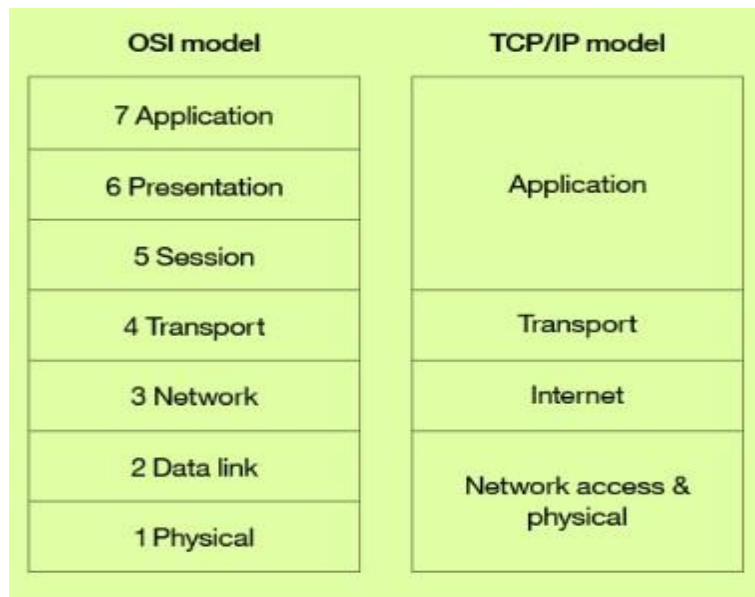


Figure 1. OSI and TCP/IP networking models

The TCP/IP model includes only four layers, merging some of the OSI model layers:

- **Network Access & Physical Layer**

This TCP/IP Layer subsumes both OSI layers 1 and 2. The physical (PHY) layer (Layer 1 of OSI) governs how each device is physically connected to the network with hardware, for example with an optic cable, wires, or radio in the case of wireless network like wifi IEEE 802.11 a/b/g/n). At the link layer (Layer 2 of OSI), devices are identified by a MAC address, and protocols at this level are concerned with physical addressing, such as how switches deliver frames to devices on the network.

- **Internet Layer**

This layer maps to the OSI Layer 3 (network layer). OSI Layer 3 relates to logical addressing. Protocols at this layer define how routers deliver packets of data

between source and destination hosts identified by IP addresses. IPv6 is commonly adopted for IoT device addressing.

- **Transport Layer**

The transport layer (Layer 4 in OSI) focuses on end-to-end communication and provides features such as reliability, congestion avoidance, and guaranteeing that packets will be delivered in the same order that they were sent. UDP (User Datagram protocol) is often adopted for IoT transport for performance reasons.

- **Application Layer**

The application layer (Layers 5, 6, and 7 in OSI) covers application-level messaging. HTTP/S is an example of an application layer protocol that is widely adopted across the internet.

Although the TCP/IP and OSI models provide you with useful abstractions for discussing networking protocols and specific technologies that implement each protocol, some protocols don't fit neatly into these layered models and are impractical. For example, the Transport Layer Security (TLS) protocol that implements encryption to ensure privacy and data integrity of network traffic can be considered to operate across OSI layers 4, 5, and 6.

NETWORK ACCESS AND PHYSICAL LAYER IOT NETWORK TECHNOLOGIES

IoT network technologies to be aware of toward the bottom of the protocol stack include cellular, Wifi, and Ethernet, as well as more specialized solutions such as LPWAN, Bluetooth Low Energy (BLE), ZigBee, NFC, and RFID.

NB-IoT is becoming the standard for LPWAN networks, according to Gartner. This IoT for All article tells more about NB-IoT.

The following are network technologies with brief descriptions of each:

- **LPWAN**

(Low Power Wide Area Network) is a category of technologies designed for low-power, long-range wireless communication. They are ideal for large-scale deployments of low-power IoT devices such as wireless sensors. LPWAN technologies include LoRa (LongRange physical layer protocol), Haystack, SigFox, LTE-M, and NB-IoT (Narrow-Band IoT).

- **Cellular**

The LPWAN NB-IoT and LTE-M standards address low-power, low-cost IoT communication options using existing cellular networks. NB-IoT is the newest of

these standards and is focused on long-range communication between large numbers of primarily indoor devices. LTE-M and NB-IoT were developed specifically for IoT, however existing cellular technologies are also frequently adopted for long-range wireless communication. While this has included 2G (GSM) in legacy devices (and currently being phased out), CDMA (also being retired or phased out), it also includes 3G, which is rapidly being phased out with several network providers retiring all 3G devices. 4G is still active and will be until 5G becomes fully available and implemented.

- **Bluetooth Low Energy (BLE)**

BLE is a low-power version of the popular Bluetooth 2.4 GHz wireless communication protocol. It is designed for short-range (no more than 100 meters) communication, typically in a star configuration, with a single primary device that controls several secondary devices. Bluetooth operates across both layers 1 (PHY) and 2 (MAC) of the OSI model. BLE is best suited to devices that transmit low volumes of data in bursts. Devices are designed to sleep and save power when they are not transmitting data. Personal IoT devices such as wearable health and fitness trackers, often use BLE.

- **ZigBee**

ZigBee operates on 2.4GHz wireless communication spectrum. It has a longer range than BLE by up to 100 meters. It also has a slightly lower data rate (250 kbps maximum compared to 270 kbps for BLE) than BLE. ZigBee is a mesh network protocol. Unlike BLE, not all devices can sleep between bursts. Much depends on their position in the mesh and whether they need to act as routers or controllers within the mesh. ZigBee was designed for building and home automation applications. Another closely related technology to ZigBee is Z-Wave, which is also based on IEEE 802.15.4. Z-Wave was designed for home automation. It has been proprietary technology, but was recently released as a public domain specification.

- **NFC**

The near field communication (NFC) protocol is used for very small range communication (up to 4 cm), such as holding an NFC card or tag next to a reader. NFC is often used for payment systems, but also useful for check-in systems and smart labels in asset tracking.

- **RFID**

RFID stands for Radio Frequency Identification. RFID tags store identifiers and data. The tags are attached to devices and read by an RFID reader. The typical range of RFID is less than a meter. RFID tags can be active, passive, or assisted passive. Passive tags are ideal for devices without batteries, as the ID is passively

read by the reader. Active tags periodically broadcast their ID, while assisted passive tags become active when RFID reader is present. **Dash7** is a communication protocol that uses active RFID that is designed to be used within Industrial IoT applications for secure long-range communication. Similar to NFC, a typical use case for RFID is tracking inventory items within retail and industrial IoT applications.

- **Wifi**

Wifi is standard wireless networking based on IEEE 802.11a/b/g/n specifications. 802.11n offers the highest data throughput, but at the cost of high-power consumption, so IoT devices might only use 802.11b or g for power conservation reasons. Although wifi is adopted within many prototype and current generation IoT devices, as longer-range and lower-power solutions become more widely available, it is likely that wifi will be superseded by lower-power alternatives.

- **Ethernet**

Widely deployed for wired connectivity within local area networks, Ethernet implements the IEEE 802.3 standard. Not all IoT devices need to be stationary wireless. For example, sensor units installed within a building automation system can use wired networking technologies like Ethernet. Power line communication (PLC), an alternative hard-wired solution, uses existing electrical wiring instead of dedicated network cables.

INTERNET LAYER IOT NETWORK TECHNOLOGIES

Internet layer technologies (OSI Layer 3) identify and route packets of data. Technologies commonly adopted for IoT are related to this layer, and include IPv6, 6LoWPAN, and RPL.

- **IPv6**

At the Internet layer, devices are identified by IP addresses. IPv6 is typically used for IoT applications over legacy IPv4 addressing. IPv4 is limited to 32-bit addresses, which only provide around 4.3 billion addresses in total, which is less than the current number of IoT devices that are connected, while IPv6 uses 128 bits, and so provides 2^{128} addresses (around 3.4×10^{38} or 340 billion billion billion) addresses. In practice, not all IoT devices need public addresses. Of the tens of billions of devices expected to connect via the IoT over the next few years, many will be deployed in private networks that use private address ranges and only communicate out to other devices or services on external networks by using gateways.

- **6LoWPAN**

The IPv6 Low Power Wireless Personal Area Network (6LoWPAN) standard allows IPv6 to be used over 802.15.4 wireless networks. 6LoWPAN is often used for wireless sensor networks, and the Thread protocol for home automation devices also runs over 6LoWPAN.

- **RPL**

The Internet Layer also covers routing. IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) is designed for routing IPv6 traffic over low-power networks like those networks implemented over 6LoWPAN. RPL (pronounced “ripple”) is designed for routing packets within constrained networks such as wireless sensor networks, where not all devices are reachable at all times and there are high or unpredictable amounts of packet loss. RPL can compute the optimal path by building up a graph of the nodes in the network based on dynamic metrics and constraints like minimizing energy consumption or latency.

APPLICATION LAYER IOT NETWORK TECHNOLOGIES

HTTP and HTTPS are ubiquitous across internet applications, which is true also within IoT, with RESTful HTTP and HTTPS interfaces widely deployed. CoAP (Constrained Application Protocol) is like a lightweight HTTP that is often used in combination with 6LoWPAN over UDP. Messaging protocols like MQTT, AMQP, and XMPP are also frequently used within IoT applications:

- **MQTT**

Message Queue Telemetry Transport (MQTT) is a publish/subscribe-based messaging protocol that was designed for use in low bandwidth situations, particularly for sensors and mobile devices on unreliable networks.

- **AMQP**

Advanced Message Queuing Protocol (AMQP) is an open standard messaging protocol that is used for message-oriented middleware. Most notably, AMQP is implemented by RabbitMQ.

- **XMPP**

The Extensible Messaging and Presence Protocol (XMPP) was originally designed for real-time human-to-human communication including instant messaging. This protocol has been adapted for machine-to-machine (M2M) communication to implement lightweight middleware and for routing XML data. XMPP is primarily used with smart appliances.

Your choice of technologies at this layer will depend on the specific application requirements of your IoT project. For example, for a budget home automation system that involves several sensors, MQTT would be a good choice as it is great for implementing messaging on devices without much storage or processing power because the protocol is simple and lightweight to implement.

IOT NETWORKING CONSIDERATIONS AND CHALLENGES

When you consider which networking technologies to adopt within your IoT application, be mindful of the following constraints:

- Range
- Bandwidth
- Power usage
- Intermittent connectivity
- Interoperability
- Security

Range

Networks can be described in terms of the distances over which data is typically transmitted by the IoT devices attached to the network:

- **PAN(Persona lArea Network)**
PAN is short-range, where distances can be measured in meters, such as a wearable fitness tracker device that communicates with an app on a cell phone over BLE.
- **LAN(Local Area Network)**
LAN is short- to medium-range, where distances can be up to hundreds of meters, such as home automation or sensors that are installed within a factory production line that communicate over wifi with a gateway device that is installed within the same building.
- **MAN (Metropolitan Area Network)**
MAN is long-range (city wide), where distances are measured up to a few kilometers, such as smart parking sensors installed throughout a city that are connected in a mesh network topology.
- **WAN (Wide Area Network)**
WAN is long-range, where distances can be measured in kilometers, such as agricultural sensors that are installed across a large farm or ranch that are used to monitor micro-climate environmental conditions across the property.

Your network should retrieve data from the IoT devices and transmit to its intended destination. Select a network protocol that matches the range is required. For example, do not choose BLE for a WAN application to operate over a range of several kilometers. If transmitting data over the required range presents a challenge, consider edge computing. Edge computing analyzes data directly from the devices rather than from a distant data center or elsewhere.

Bandwidth

Bandwidth is the amount of data that can be transmitted per unit of time. It limits the rate at which data can be collected from IoT devices and transmitted upstream. Bandwidth is affected by many factors, which include:

- The volume of data each device gathers and transmits
- The number of devices deployed
- Whether data is being sent as a constant stream or in intermittent bursts, and if any peak periods are notable

The packet size of the networking protocol should match up with the volume of data typically transmitted. It is inefficient to send packets padded with empty data. In contrast, there are overheads in splitting larger chunks of data up across too many small packets. Data transmission rates are not always symmetrical (that is, upload rates might be slower than download rates). So, if there is two-way communication between devices, data transmission needs to be factored in. Wireless and cellular networks are traditionally low bandwidth, so consider whether a wireless technology is the right choice for high-volume applications.

Consider whether all raw data must be transmitted. A possible solution is to capture less data by sampling less frequently. Thus, you'll capture fewer variables and may filter data from the device to drop insignificant data. If you aggregate the data before you transmit it, you reduce the volume of data transmitted. But this process affects flexibility and granularity in the upstream analysis. Aggregation and bursting are not always suitable for time-sensitive or latency-sensitive data. All of these techniques increase the data processing and storage requirements for the IoT device.

Power usage

Transmitting data from a device consumes power. Transmitting data over long ranges requires more power than over a short range. You must consider the power source – such as a battery, solar cell, or capacitor – of a device and its total lifecycle. A long and enduring lifecycle will not only provide greater reliability but reduce operating cost. Steps may be taken to help achieve longer power supply lifecycles. For example, to prolong the battery life, you can put the device into sleep mode whenever it is idle. Another best practice is to model the energy consumption of the device under different loads and different network conditions to ensure that the device's power supply and storage capacity matches with the power that is required to transmit the necessary data by using the networking technologies that you adopted.

Intermittent connectivity

IoT devices aren't always connected. In some cases, devices are designed to connect periodically. However, sometimes an unreliable network might cause devices to drop off due to connectivity issues. Sometimes quality of service issues, such as dealing with interference or channel contention on a wireless network using a shared spectrum. Designs should incorporate intermittent connectivity and seek any available solutions to provide uninterrupted service, should that be a critical factor for IoT landscape design.

Interoperability

Devices work with other devices, equipment, systems, and technology; they are interoperable. With so many different devices connecting to the IoT, interoperability can be a challenge. Adopting standard protocols has been a traditional approach for maintaining interoperability on the Internet. Standards are agreed upon by industry participants and avoid multiple different designs and directions. With proper standards, and participants who agree to them, incompatibility issues, hence interoperability issues may be avoided.

However, for the IoT, standardization processes sometimes struggle to keep up with innovation and change. They are written and released based on upcoming versions of standards that are still subject to change. Consider the ecosystem around the technologies: Are they widely adopted? Are they open versus proprietary? How many implementations are available?

Using these questions to plan your IoT networks help plan better interoperability for a more robust IoT network.

Security

Security is a priority. Selection of networking technologies that implement end-to-end security, including authentication, encryption, and open port protection is crucial. IEEE 802.15.4 includes a security model that provides security features that include access control, message integrity, message confidentiality, and replay protection, which are implemented by technologies based on this standard such as ZigBee.

Consider the following factors in shaping a secure and safe IoT network:

- **Authentication**

Adopt secure protocols to support authentication for devices, gateways, users, services, and applications. Consider using adopting the X.509 standard for device authentication.

- **Encryption**

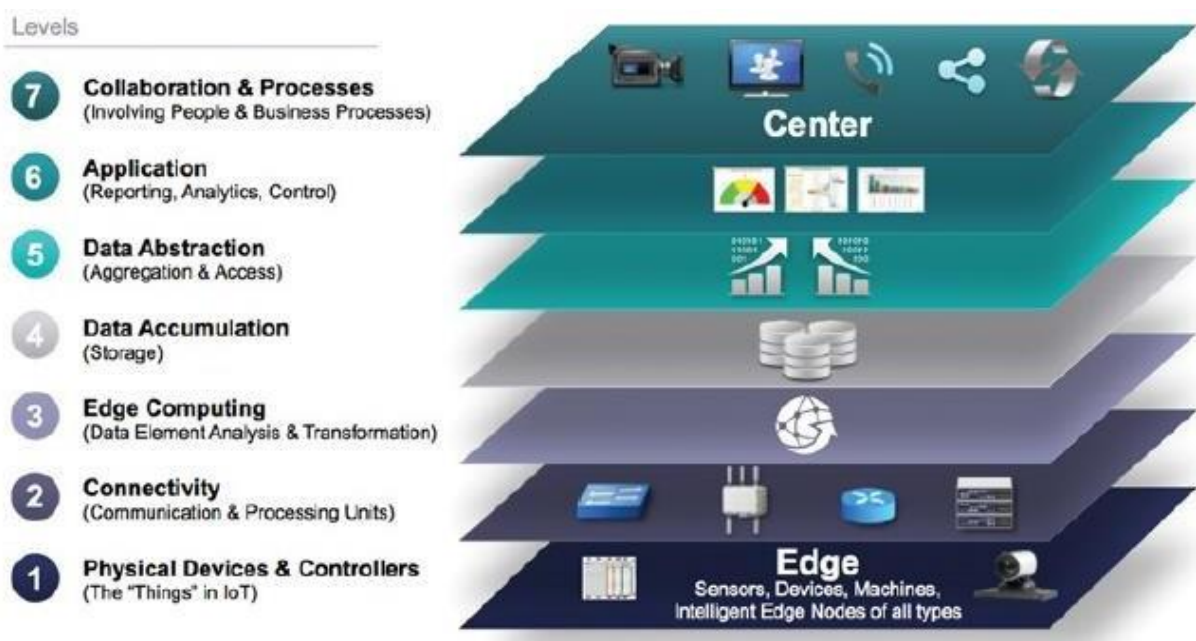
If you are using wifi, use Wireless Protected Access 2 (WPA2) for wireless network encryption. You may also adopt a Private Pre-Shared Key (PPSK) approach. To ensure privacy and data integrity for communication between applications, be sure to adopt TLS or Datagram Transport-Layer Security (DTLS), which is based on TLS, but adapted for unreliable connections that run over UDP. TLS encrypts application data and ensures its integrity.

- **Port protection**

Port protection ensures that only the ports required for communication with the gateway or upstream applications or services remain open to external connections. All other ports should be disabled or protected by firewalls. Device ports might be exposed when exploiting Universal Plug and Play (UPnP) vulnerabilities. Thus, UPnP should be disabled on the router.

The IoT World Forum (IoTWF) Standardized Architecture

In 2014 the IoTWF architectural committee (led by Cisco, IBM, Rockwell Automation, and others) published a seven-layer IoT architectural reference model. While various IoT reference models exist, the one put forth by the IoT World Forum offers a clean, simplified perspective on IoT and includes edge computing, data storage, and access. It provides a succinct way of visualizing IoT from a technical perspective. Each of the seven layers is broken down into specific functions, and security encompasses the entire model. Figure below details the IoT Reference Model published by the IoTWF.



As shown in [Figure 2-2](#), the IoT Reference Model defines a set of levels with control flowing from the center (this could be either a cloud service or a dedicated data center), to the edge,

which includes sensors, devices, machines, and other types of intelligent end nodes. In general, data travels up the stack, originating from the edge, and goes northbound to the center. Using this reference model, we are able to achieve the following:

- Decompose the IoT problem into smaller parts
- Identify different technologies at each layer and how they relate to one another
- Define a system in which different parts can be provided by different vendors
- Have a process of defining interfaces that leads to interoperability
- Define a tiered security model that is enforced at the transition points between levels

The following sections look more closely at each of the seven layers of the IoT Reference Model.

Layer 1: Physical Devices and Controllers Layer

The first layer of the IoT Reference Model is the physical devices and controllers layer. This layer is home to the “things” in the Internet of Things, including the various endpoint devices and sensors that send and receive information. The size of these “things” can range from almost microscopic sensors to giant machines in a factory. Their primary function is generating data and being capable of being queried and/or controlled over a network.

Layer 2: Connectivity Layer

In the second layer of the IoT Reference Model, the focus is on connectivity. The most important function of this IoT layer is the reliable and timely transmission of data. More specifically, this includes transmissions between Layer 1 devices and the network and between the network and information processing that occurs at Layer 3 (the edge computing layer). As you may notice, the connectivity layer encompasses all networking elements of IoT and doesn’t really distinguish between the last-mile network (the network between the sensor/endpoint and the IoT gateway, discussed later in this chapter), gateway, and backhaul networks. Functions of the connectivity layer are detailed in Figure 2-3.

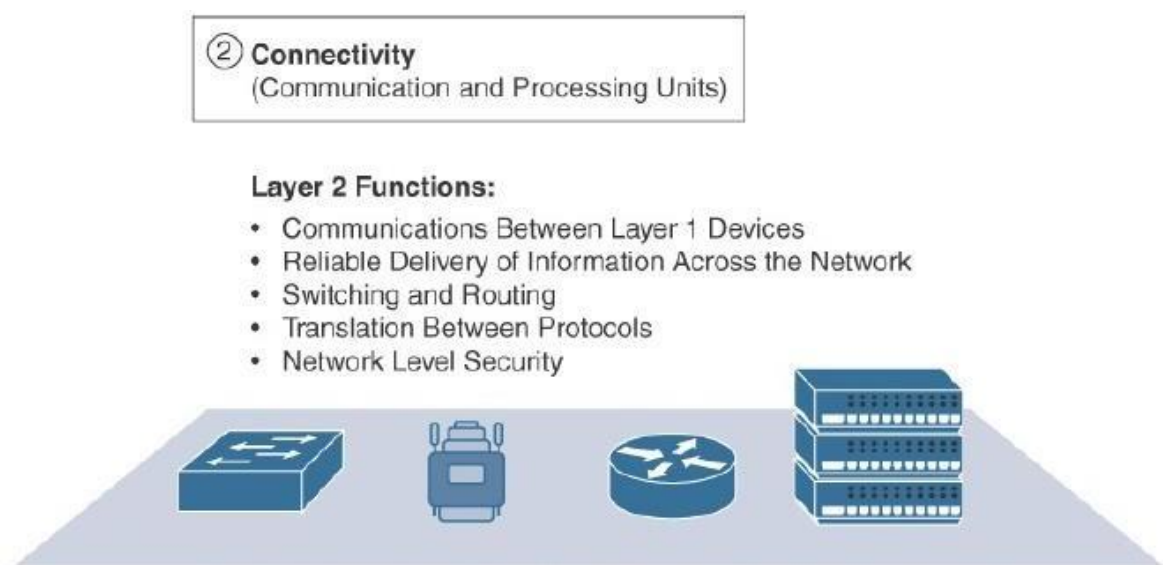


Figure 2-3 IoT Reference Model Connectivity Layer Functions

Layer 3: Edge Computing Layer

Edge computing is the role of Layer 3. Edge computing is often referred to as the “fog” layer and is discussed in the section “[Fog Computing](#),” later in this chapter. At this layer, the emphasis is on data reduction and converting network data flows into information that is ready for storage and processing by higher layers. One of the basic principles of this reference model is that information processing is initiated as early and as close to the edge of the network as possible. [Figure 2-4](#) highlights the functions handled by Layer 3 of the IoT Reference Model.

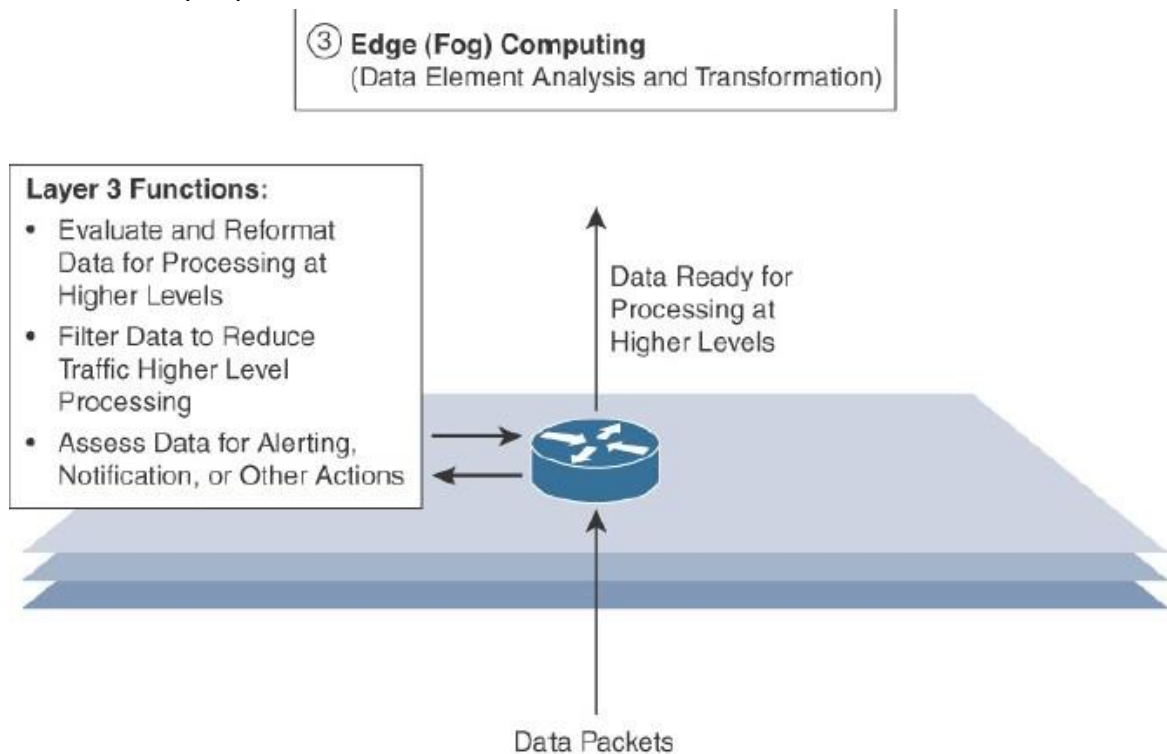


Figure 2-4 IoT Reference Model Layer 3 Functions

Another important function that occurs at Layer 3 is the evaluation of data to see if it can be filtered or aggregated before being sent to a higher layer. This also allows for data to be reformatted or decoded, making additional processing by other systems easier. Thus, a critical function is assessing the data to see if predefined thresholds are crossed and any action or alerts need to be sent.

Upper Layers: Layers 4–7

The upper layers deal with handling and processing the IoT data generated by the bottom layer. For the sake of completeness, Layers 4–7 of the IoT Reference Model are summarized in [Table 2-2](#).

IoT Reference Model Layer	Functions
Layer 4: Data accumulation layer	Captures data and stores it so it is usable by applications when necessary. Converts event-based data to query-based processing.
Layer 5: Data abstraction layer	Reconciles multiple data formats and ensures consistent semantics from various sources. Confirms that the data set is complete and consolidates data into one place or multiple data stores using virtualization.
Layer 6: Applications layer	Interprets data using software applications. Applications may monitor, control, and provide reports based on the analysis of the data.
Layer 7: Collaboration and processes layer	Consumes and shares the application information. Collaborating on and communicating IoT information often requires multiple steps, and it is what makes IoT useful. This layer can change business processes and delivers the benefits of IoT.

Table 2-2 *Summary of Layers 4–7 of the IoTWF Reference Model*

M2M Communication

Machine-to-machine communication, or M2M, is exactly as it sounds: two machines “communicating,” or exchanging data, without human interfacing or interaction. This includes serial connection, powerline connection (PLC), or wireless communications in the industrial Internet of Things (IoT). Switching over to wireless has made M2M communication much easier and enabled more applications to be connected.

In general, when someone says M2M communication, they often are referring to cellular communication for embedded devices. Examples of M2M communication in this case would be vending machines sending out inventory information or ATM machines getting authorization to dispense cash.

As businesses have realized the value of M2M, it has taken on a new name: the Internet of Things (IoT). IoT and M2M have similar promises: to fundamentally change the way the world operates. Just like IoT, M2M allows virtually any sensor to communicate, which opens up the possibility of systems monitoring themselves and automatically responding to changes in the environment, with a much reduced need for human involvement. M2M and IoT are almost synonymous—the exception is IoT (the newer term) typically refers to wireless communications, whereas M2M can refer to any two machines—wired or wireless—communicating with one another.

Traditionally, M2M focused on “industrial telematics,” which is a fancy way of explaining data transfer for some commercial benefit. But many original uses of M2M still stand today, like smart meters. Wireless M2M has been dominated by cellular since it came out in the mid-2000’s with 2G cell networks. Because of this, the cellular market has tried to brand M2M as an inherently cellular thing by offering M2M data plans. But cellular M2M is only one subsection of the market, and it shouldn’t be thought of as a cellular-only area.

How M2M Works

As previously stated, machine-to-machine communication makes the Internet of Things possible. According to *Forbes*, M2M is among the fastest-growing types of connected device technologies in the market right now, largely because M2M technologies can connect millions of devices within a single network. The range of connected devices includes anything from vending machines to medical equipment to vehicles to buildings. Virtually anything that houses sensor or control technology can be connected to some sort of wireless network.

This sounds complex, but the driving thought behind the idea is quite simple. Essentially, M2M networks are very similar to LAN or WAN networks, but are exclusively used to allow machines, sensors, and controls, to communicate. These devices feed information they collect back to other devices in the network. This process allows a human (or an intelligent control unit) to assess what is going on across the whole network and issue appropriate instructions to member devices.

M2M Applications

The possibilities in the realm of M2M can be seen in four major use cases, which we’ve detailed below:

1. MANUFACTURING

Every manufacturing environment—whether it’s food processing or general product manufacturing—relies on technology to ensure costs are managed properly and processes are executed efficiently. Automating manufacturing processes within such a fast-paced environment is expected to improve processes even more. In the manufacturing world, this could involve highly automated equipment maintenance and safety procedures.

For example, M2M tools allow business owners to be alerted on their smartphones when an important piece of equipment needs servicing, so they can address issues as quickly as they arise. Sophisticated networks of sensors connected to the Internet could even order replacement parts automatically.

2. HOME APPLIANCES

IoT already affects home appliance connectivity through platforms like Nest. However, M2M is expected to take home-based IoT to the next level. Manufacturers like LG and Samsung are already slowly unveiling smart home appliances to help ensure a higher quality of life for occupants.

For example, an M2M-capable washing machine could send alerts to the owners' smart devices once it finishes washing or drying, and a smart refrigerator could automatically order groceries from Amazon once its inventory is depleted. There are many more examples of home automation that can potentially improve quality of life for residents, including systems that allow members of the household to remotely control HVAC systems using their mobile devices. In situations where a homeowner decides to leave work early, he or she could contact the home heating system before leaving work to make sure the temperature at home will be comfortable upon arrival.

3. HEALTHCARE DEVICE MANAGEMENT

One of the biggest opportunities for M2M technology is in the realm of health care. With M2M technology, hospitals can automate processes to ensure the highest levels of treatment. Using devices that can react faster than a human healthcare professional in an emergency situation make this possible. For instance, when a patient's vital signs drop below normal, an M2M-connected life support device could automatically administer oxygen and additional care until a healthcare professional arrives on the scene. M2M also allows patients to be monitored in their own homes instead of in hospitals or care centers. For example, devices that track a frail or elderly person's normal movements can detect when he or she has had a fall and alert a healthcare worker to the situation.

4. SMART UTILITY MANAGEMENT

In the new age of energy efficiency, automation will quickly become the new normal. As energy companies look for new ways to automate the metering process, M2M comes to the rescue, helping energy companies automatically gather energy consumption data, so they can accurately bill customers. Smart meters can track how much energy a household or business uses and automatically alert the energy company, which supplants sending out an employee to read the meter or requiring the customer to provide a reading. This is even more important as utilities move toward more dynamic pricing models, charging consumers more for energy usage during peak times.

A few key analysts predict that soon, every object or device will need to be able to connect to the cloud. This is a bold but seemingly accurate statement. As more consumers, users, and business owners demand deeper connectivity, technology will need to be continually

equipped to meet the needs and challenges of tomorrow. This will empower a wide range of highly automated processes, from equipment repairs and firmware upgrades to system diagnostics, data retrieval, and analysis. Information will be delivered to users, engineers, data scientists, and key decision-makers in real time, and it will eliminate the need for guesswork.

The Value Of M2M

Growth in the M2M and IoT markets has been growing rapidly, and according to many reports, growth will continue. Strategy Analytics believes that low power, wide-area network (LPWAN) connections will grow from 11 million in 2014 to 5 billion in 2022. And IDC says the market for worldwide IoT solutions will go from \$1.9 trillion in 2013 to \$7.1 trillion in 2020.

Many big cell operators, like AT&T and Verizon, see this potential and are rolling out their own M2M platforms. Intel, PTC, and Wipro are all marketing heavily in M2M and working to take advantage of this major industry growth spurt. But there is still a great opportunity for new technology companies to engage in highly automated solutions to help streamline processes in nearly any type of industry. We're certain we'll see a huge influx of companies who begin to innovate in this area in the next five years.

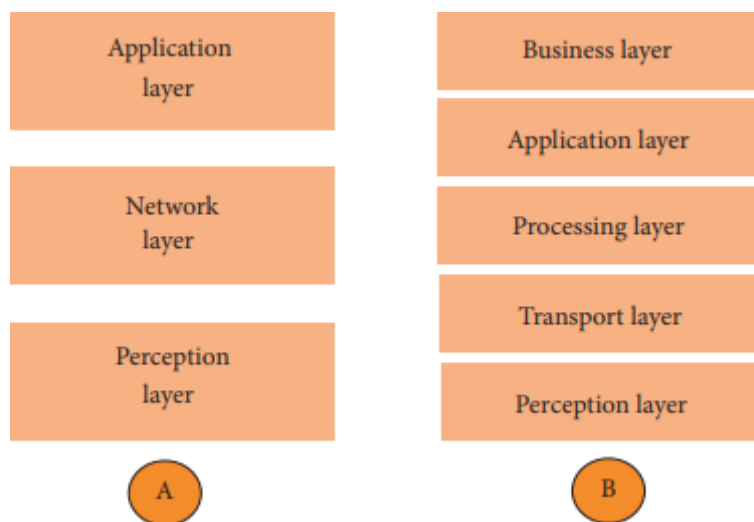
However, as the cost of M2M communication continues to decrease, companies must determine how they will create value for businesses and customers. In our mind, the opportunity and value for M2M doesn't lie in the more traditional layers of the communication world. Cell carriers and hardware manufacturers, for example, are beginning to look into full-stack offerings that enable M2M and IoT product development. We strongly believe value lies in the application side of things, and the growth in this industry will be driven by smart applications from this point forward.

Companies shouldn't think about IoT or M2M for the sake of IoT or M2M. Instead, they should focus on optimizing their business models or providing new value for their customers. For example, if you're a logistics company like FedEx or UPS, you have obvious choices for automated logistics decisions made by machines. But if you're a retailer, the transition to automation may not be as obvious. It's one thing to think of a "cool" automated process—say, creating advertising that is automatically tied to a specific customer through the use of M2M technology—but before you move forward with the process, you have to consider the value you're getting out of it. How much does it cost to implement? Will any company considering a move into the IoT space need to understand what its business model is, how it will make money, and how it will provide value for customers or internal processes.

Architecture of IoT

Figure below has three layers, namely, the perception, network, and application layers.

- (i) The perception layer is the physical layer, which has sensors for sensing and gathering information about the environment. It senses some physical parameters or identifies other smart objects in the environment.
- (ii) The network layer is responsible for connecting to other smart things, network devices, and servers. Its features are also used for transmitting and processing sensor data.
- (iii) The application layer is responsible for delivering application specific services to the user. It defines various applications in which the Internet of Things can be deployed, for example, smart homes, smart cities, and smart health.



The three-layer architecture defines the main idea of the Internet of Things, but it is not sufficient for research on IoT because research often focuses on finer aspects of the Internet of Things. That is why, we have many more layered architectures proposed in the literature. One is the five-layer architecture, which additionally includes the processing and business layers [3–6]. The five layers are perception, transport, processing, application, and business layers (see Figure 1). The role of the perception and application layers is the same as the architecture with three layers. We outline the function of the remaining three layers.

- (i) The transport layer transfers the sensor data from the perception layer to the processing layer and vice versa through networks such as wireless, 3G, LAN, Bluetooth, RFID, and NFC.
- (ii) The processing layer is also known as the middleware layer. It stores, analyzes, and processes huge amounts of data that comes from the transport layer. It can manage and provide a diverse set of services to the

lower layers. It employs many technologies such as databases, cloud computing, and big data processing modules.

- (iii) The business layer manages the whole IoT system, including applications, business and profit models, and users' privacy. The business layer is out of the scope of this paper. Hence, we do not discuss it further.

Core IoT Functional Stack

The IoT network must be designed to support its unique requirements and constraints. This section provides an overview of the full networking stack, from sensors all the way to the applications layer.

The Core IoT Functional Stack IoT networks are built around the concept of “things,” or smart objects performing functions and delivering new connected services. These objects are “smart” because they use a combination of contextual information and configured goals to perform actions. These actions can be self-contained (that is, the smart object does not rely on external systems for its actions); however, in most cases, the “thing” interacts with an external system to report information that the smart object collects, to exchange with other objects, or to interact with a management platform. In this case, the management platform can be used to process data collected from the smart object and also guide the behavior of the smart object. From an architectural standpoint, several components have to work together for an IoT network to be operational:

- “Things” layer: At this layer, the physical devices need to fit the constraints of the environment in which they are deployed while still being able to provide the information needed.
- Communications network layer: When smart objects are not self-contained, they need to communicate with an external system. In many cases, this communication uses a wireless technology. This layer has four sublayers:
 - Access network sublayer: The last mile of the IoT network is the access network. This is typically made up of wireless technologies such as 802.11ah, 802.15.4g, and LoRa. The sensors connected to the access network may also be wired.
 - Gateways and backhaul network sublayer: A common communication system organizes 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 headend central station where the information is processed. This information exchange is a Layer 7 (application) function, which is the reason this object is called a gateway. On IP networks, this gateway also forwards packets from one IP network to another, and it therefore acts as a router.
 - Network transport sublayer: For communication to be successful, network and transport layer protocols such as IP and UDP must be implemented to support the variety of devices to connect and media to use.
 - IoT network management sublayer: Additional protocols must be in place to allow the headend applications to exchange data with the sensors. Examples include CoAP and MQTT.

Application and analytics layer: At the upper layer, an application needs to process the collected data, not only to control the smart objects when necessary, but to make intelligent decision based on the information collected and, in turn, instruct the “things” or other systems to adapt to the analyzed conditions and change their behaviors or parameters. The following sections examine these elements and help you architect your IoT communication network.

Layer 1: Things: Sensors and Actuators Layer

Most IoT networks start from the object, or “thing,” that needs to be connected. From an architectural standpoint, the variety of smart object types, shapes, and needs drive the variety of IoT protocols and architectures. There are myriad ways to classify smart objects. One architectural classification could be:

- **Battery-powered or power-connected:** This classification is based on whether the object carries its own energy supply or receives continuous power from an external power source. Battery-powered things can be moved more easily than line-powered objects. However, batteries limit the lifetime and amount of energy that the object is allowed to consume, thus driving transmission range and frequency.
- **Mobile or static:** This classification is based on whether the “thing” should move or always stay at the same location. A sensor may be mobile because it is moved from one object to another (for example, a viscosity sensor moved from batch to batch in a chemical plant) or because it is attached to a moving object (for example, a location sensor on moving goods in a warehouse or factory floor). The frequency of the movement may also vary, from occasional to permanent. The range of mobility (from a few inches to miles away) often drives the possible power source.
- **Low or high reporting frequency:** This classification is based on how often the object should report monitored parameters. A rust sensor may report values once a month. A motion sensor may report acceleration several hundred times per second. Higher frequencies drive higher energy consumption, which may create constraints on the possible power source (and therefore the object mobility) and the transmission range.
- **Simple or rich data:** This classification is based on the quantity of data exchanged at each report cycle. A humidity sensor in a field may report a simple daily index value (on a binary scale from 0 to 255), while an engine sensor may report hundreds of parameters, from temperature to pressure, gas velocity, compression speed, carbon index, and many others. Richer data typically drives higher power consumption. This classification is often combined with the previous to determine the object data throughput (low throughput to high throughput). You may want to keep in mind that throughput is a combined metric. A medium-throughput object may send simple data at rather high frequency (in which case the flow structure looks

continuous), or may send rich data at rather low frequency (in which case the flow structure looks bursty).

- **Report range:** This classification is based on the distance at which the gateway is located. For example, for your fitness band to communicate with your phone, it needs to be located a few meters away at most. The assumption is that your phone needs to be at visual distance for you to consult the reported data on the phone screen. If the phone is far away, you typically do not use it, and reporting data from the band to the phone is not necessary. By contrast, a moisture sensor in the asphalt of a road may need to communicate with its reader several hundred meters or even kilometers away.
- **Object density per cell:** This classification is based on the number of smart objects (with a similar need to communicate) over a given area, connected to the same gateway. An oil pipeline may utilize a single sensor at key locations every few miles. By contrast, telescopes like the SETI Colossus telescope at the Whipple Observatory deploy hundreds, and sometimes thousands, of mirrors over a small area, each with multiple gyroscopes, gravity, and vibration sensors.

Layer 2: Communications Network Layer

Once you have determined the influence of the smart object form factor over its transmission capabilities (transmission range, data volume and frequency, sensor density and mobility), you are ready to connect the object and communicate. Compute and network assets used in IoT can be very different from those in IT environments. The difference in the physical form factors between devices used by IT and OT is obvious even to the most casual of observers. What typically drives this is the physical environment in which the devices are deployed. What may not be as inherently obvious, however, is their operational differences. The operational differences must be understood in order to apply the correct handling to secure the target assets. Temperature variances are an easily understood metric. The cause for the variance is easily attributed to external weather forces and internal operating conditions. Remote external locations, such as those associated with mineral extraction or pipeline equipment can span from the heat of the Arabian Gulf to the cold of the Alaskan North Slope. Controls near the furnaces of a steel mill obviously require heat tolerance, and controls for cold food storage require the opposite. In some cases, these controls must handle extreme fluctuations as well. These extremes can be seen within a single deployment. For example, portions of the Tehachapi, California, wind farms are located in the Mojave Desert, while others are at an altitude of 1800 m in the surrounding mountains. As you can imagine, the wide variance in temperature takes a special piece of hardware that is capable of withstanding such harsh environments. Humidity fluctuations can impact the long-term success of a system as well. Well heads residing in the delta of the Niger River

will see very different conditions from those in the middle of the Arabian Desert. In some conditions, the systems could be exposed to direct liquid contact such as may be found with outdoor wireless devices or marine condition deployments. Less obvious are the operating extremes related to kinetic forces. Shock and vibration needs vary based on the deployment scenario. In some cases, the focus is on low-amplitude but constant vibrations, as may be expected on a bushing-mounted manufacturing system. In other cases, it could be a sudden acceleration or deceleration, such as may be experienced in peak ground acceleration of an earthquake or an impact on a mobile system such as high-speed rail or heavy-duty earth moving equipment. Solid particulates can also impact the gear. Most IT environments must contend with dust build-up that can become highly concentrated due to the effect of cooling fans. In less-controlled IT environments, that phenomenon can be accelerated due to higher concentrations of particulates. A deterrent to particulate build-up is to use fanless cooling, which necessitates a higher surface area, as is the case with heat transfer fins. Hazardous location design may also cause corrosive impact to the equipment. Caustic materials can impact connections over which power or communications travel. Furthermore, they can result in reduced thermal efficiency by potentially coating the heat transfer surfaces. In some scenarios, the concern is not how the environment can impact the equipment but how the equipment can impact the environment. For example, in a scenario in which volatile gases may be present, spark suppression is a critical design criterion. There is another class of device differentiators related to the external connectivity of the device for mounting or industrial function. Device mounting is one obvious difference between OT and IT environments. While there are rack mount environments in some industrial spaces, they are more frequently found among IT type assets. Within industrial environments, many compute and communication assets are placed within an enclosed space, such as a control cabinet where they will be vertically mounted on a DIN (Deutsches Institut für Normung) rail inside. In other scenarios, the devices might be mounted horizontally directly on a wall or on a fence. In contrast to most IT-based systems, industrial compute systems often transmit their state or receive inputs from external devices through an alarm channel. These may drive an indicator light (stack lights) to display the status of a process element from afar. This same element can also receive inputs to initiate actions within the system itself. Power supplies in OT systems are also frequently different from those commonly seen on standard IT equipment. A wider range of power variations are common attributes of industrial compute components. DC power sources are also common in many environments. Given the criticality of many systems, it is often required that redundant power supplies be built into the device itself. Extraneous power supplies, especially those not inherently mounted, are frowned upon, given the potential for accidental unplugging. In some utility cases, the system must be able to handle brief power outages and still continue to operate.

- **Access Network Sublayer**

There is a direct relationship between the IoT network technology you choose and the type of connectivity topology this technology allows. Each technology was designed with a certain number of use cases in mind (what to connect, where to connect, how much data to transport at what interval and over what distance). These use cases determined the frequency band that was expected to be most suitable, the frame structure matching the expected data pattern (packet size and communication intervals), and the possible topologies that these use cases illustrate. As IoT continues to grow exponentially, you will encounter a wide variety of applications and special use cases. For each of them, an access technology will be required. IoT sometimes reuses existing access technologies whose characteristics match more or less closely the IoT use case requirements. Whereas some access technologies were developed specifically for IoT use cases, others were not. One key parameter determining the choice of access technology is the range between the smart object and the information collector. Figure 2-9 lists some access technologies you may encounter in the IoT world and the expected transmission distances.

PAN (personal area network): Scale of a few meters. This is the personal space around a person. A common wireless technology for this scale is Bluetooth.

HAN (home area network): Scale of a few tens of meters. At this scale, common wireless technologies for IoT include ZigBee and Bluetooth Low Energy (BLE).

NAN (neighborhood area network): Scale of a few hundreds of meters. The term NAN is often used to refer to a group of house units from which data is collected.

FAN (field area network): Scale of several tens of meters to several hundred meters. FAN typically refers to an outdoor area larger than a single group of house units. The FAN is often seen as “open space” (and therefore not secured and not controlled). A FAN is sometimes viewed as a group of NANs, but some verticals see the FAN as a group of HANs or a group of smaller outdoor cells. As you can see, FAN and NAN may sometimes be used interchangeably. In most cases, the vertical context is clear enough to determine the grouping hierarchy.

LAN (local area network): Scale of up to 100 m. This term is very common in networking, and it is therefore also commonly used in the IoT space when standard networking technologies (such as Ethernet or IEEE 802.11) are used. Other networking classifications, such as MAN (metropolitan area network, with a range of up to a few kilometers) and WAN (wide area network, with a range of more than a few kilometers), are also commonly used.

Network Transport Sublayer

The previous section describes a hierarchical communication architecture in which a series of smart objects report to a gateway that conveys the reported data over another medium and up to a central station. However, practical implementations are often flexible, with multiple transversal communication paths. For example, consider the case of IoT for the energy grid. Your house may have a meter that reports the energy consumption to a gateway over a wireless technology. Other houses in your neighborhood (NAN) make the same report, likely to one or several gateways. The data to be transported is small and the interval is large (for example, four times per hour), resulting in a low-mobility, lowthroughput type of data structure, with transmission distances up to a mile. Several technologies (such as 802.11ah, 802.15.4, or LPWA) can be used for this collection segment. Other neighborhoods may also connect the same way, thus forming a FAN.

IoT Network Management Sublayer

IP, TCP, and UDP bring connectivity to IoT networks. Upper-layer protocols need to take care of data transmission between the smart objects and other systems. Multiple protocols have been leveraged or created to solve IoT data communication problems. Some networks rely on a push model (that is, a sensor reports at a regular interval or based on a local trigger), whereas others rely on a pull model (that is, an application queries the sensor over the network), and multiple hybrid approaches are also possible.

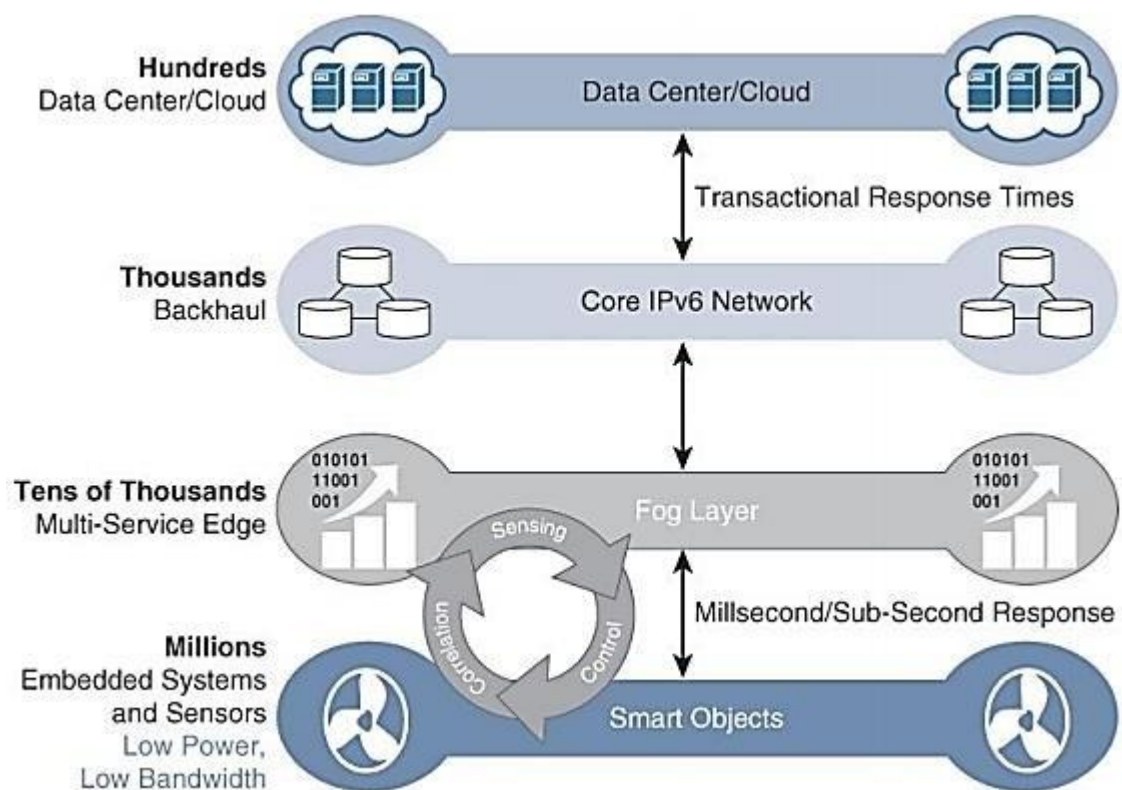
Layer 3: Applications and Analytics Layer

Once connected to a network, your smart objects exchange information with other systems. As soon as your IoT network spans more than a few sensors, the power of the Internet of Things appears in the applications that make use of the information exchanged with the smart objects. Analytics Versus Control Applications Multiple applications can help increase the efficiency of an IoT network. Each application collects data and provides a range of functions based on analyzing the collected data. It can be difficult to compare the features offered. Chapter 7, “Data and Analytics for IoT,” provides an in-depth analysis of the various application families. From an architectural standpoint, one basic classification can be as follows: Analytics application: This type of application collects data from multiple smart objects, processes the collected data, and displays information resulting from the data that was processed. The display can be about any aspect of the IoT network, from historical reports, statistics, or trends to individual system states. The important aspect is that the application processes the data to convey a view of the network that cannot be obtained from solely looking at the information displayed by a single

smart object. Control application: This type of application controls the behavior of the smart object or the behavior of an object related to the smart object. For example, a pressure sensor may be connected to a pump. A control application increases the pump speed when the connected sensor detects a drop in pressure. Control applications are very useful for controlling complex aspects of an IoT network with a logic that cannot be programmed inside a single IoT object, either because the configured changes are too complex to fit into the local system or because the configured changes rely on parameters that include elements outside the IoT object.

Fog Computing

The solution to the challenges mentioned in the previous section is to distribute data management throughout the IoT system, as close to the edge of the IP network as possible. The best-known embodiment of edge services in IoT is fog computing. Any device with computing, storage, and network connectivity can be a fog node. Examples include industrial controllers, switches, routers, embedded servers, and IoT gateways. Analyzing IoT data close to where it is collected minimizes latency, offloads gigabytes of network traffic from the core network, and keeps sensitive data inside the local network.



Fog services are typically accomplished very close to the edge device, sitting as close to the IoT endpoints as possible. One significant advantage of this is that the fog

node has contextual awareness of the sensors it is managing because of its geographic proximity to those sensors. For example, there might be a fog router on an oil derrick that is monitoring all the sensor activity at that location. Because the fog node is able to analyze information from all the sensors on that derrick, it can provide contextual analysis of the messages it is receiving and may decide to send back only the relevant information over the backhaul network to the cloud. In this way, it is performing distributed analytics such that the volume of data sent upstream is greatly reduced and is much more useful to application and analytics servers residing in the cloud.

The defining characteristic of fog computing are as follows:

Contextual location awareness and low latency: The fog node sits as close to the IoT endpoint as possible to deliver distributed computing.

Geographic distribution: In sharp contrast to the more centralized cloud, the services and applications targeted by the fog nodes demand widely distributed deployments.

Deployment near IoT endpoints: Fog nodes are typically deployed in the presence of a large number of IoT endpoints. For example, typical metering deployments often see 3000 to 4000 nodes per gateway router, which also functions as the fog computing node.

Wireless communication between the fog and the IoT endpoint: Although it is possible to connect wired nodes, the advantages of fog are greatest when dealing with a large number of endpoints, and wireless access is the easiest way to achieve such scale.

Use for real-time interactions: Important fog applications involve real-time interactions rather than batch processing. Preprocessing of data in the fog nodes allows upper-layer applications to perform batch processing on a subset of the data.

Edge Computing

Fog computing solutions are being adopted by many industries, and efforts to develop distributed applications and analytics tools are being introduced at an accelerating pace. The natural place for a fog node is in the network device that sits closest to the IoT endpoints, and these nodes are typically spread throughout an IoT network. However, in recent years, the concept of IoT computing has been pushed even further to the edge, and in some cases it now resides directly in the sensors and IoT devices.

Functional blocks of an IoT ecosystem

IoT don't exist in a void. A lone sensor isn't really good for anything, nor is a bunch of them, for that matter, unless they are all connected to one another and to platforms that generate data for further use. This is what we call an **Internet of Things (IoT) ecosystem – a broad network of connected and interdependent devices and technologies that are applied by specialists towards a specific goal**, such as the creation of a smart city.

Obviously, there are limitless applications to the IoT and therefore we can speak of endless coexisting IoT ecosystems. But if you boil what is happening in the ecosystem down to the bare essentials, you will come up with a simple schema: a **device** collects data and sends it across the **network** to a **platform** that aggregates the data for future use by the **agent**. And so we have the key components to an IoT ecosystem: devices, networks, platforms, and agents. Let's discuss them in more detail.



Four things form basic building blocks of the IoT system –sensors, processors, gateways, applications. Each of these nodes has to have its own characteristics in order to form an useful IoT system.

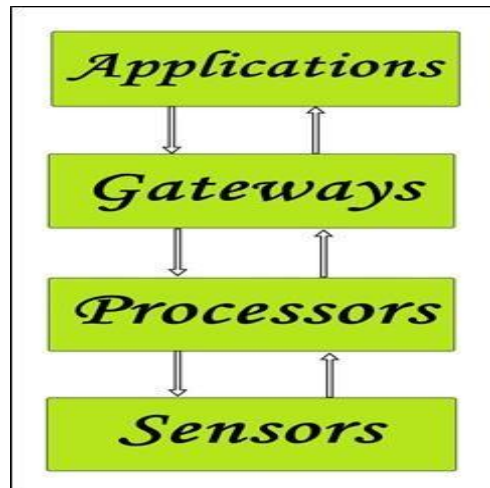


Figure 1: Simplified block diagram of the basic building blocks of the IoT

Sensors:

- These form the front end of the IoT devices. These are the so-called “Things” of the system. Their main purpose is to collect data from its surroundings (sensors) or give out data to its surrounding (actuators).
- These have to be uniquely identifiable devices with a unique IP address so that they can be easily identifiable over a large network.
- These have to be active in nature which means that they should be able to collect real-time data. These can either work on their own (autonomous in nature) or can be made to work by the user depending on their needs (user-controlled).
- Examples of sensors are gas sensor, water quality sensor, moisture sensor, etc.

Processors:

- Processors are the brain of the IoT system. Their main function is to process the data captured by the sensors and process them so as to extract the valuable data from the enormous amount of raw data collected. In a word, we can say that it gives intelligence to the data.
- Processors mostly work on real-time basis and can be easily controlled by applications. These are also responsible for securing the data – that is performing encryption and decryption of data.
- Embedded hardware devices, microcontroller, etc are the ones that process the data because they have processors attached to it.

Gateways:

- Gateways are responsible for routing the processed data and send it to proper locations for its (data) proper utilization.
- In other words, we can say that gateway helps in to and fro communication of the data. It provides network connectivity to the data. Network connectivity is essential for any IoT system to communicate.
- LAN, WAN, PAN, etc are examples of network gateways.

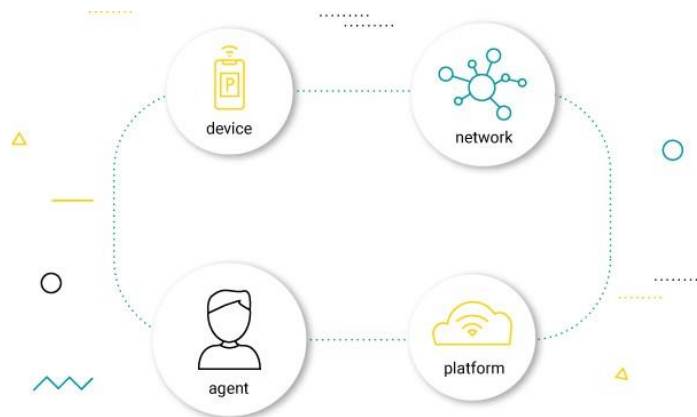
Applications:

- Applications form another end of an IoT system. Applications are essential for proper utilization of all the data collected.
- These cloud-based applications which are responsible for rendering the effective meaning to the data collected. Applications are controlled by users and are a delivery point of particular services.
- Examples of applications are home automation apps, security systems, industrial control hub, etc.

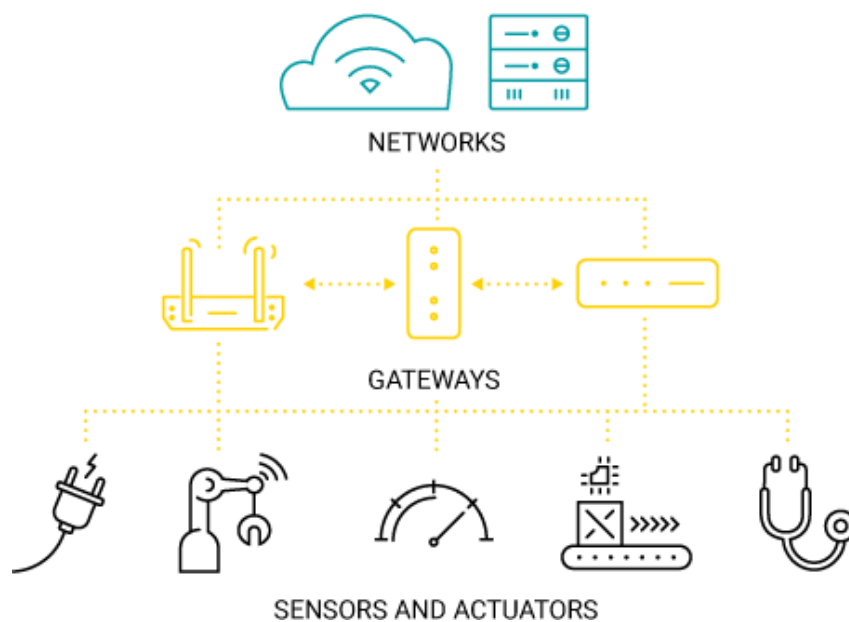
IoT devices

As we said earlier, there are many scenarios in which IoT can be employed and they all require different devices. Here, at the most basic level, we can speak of **sensors** (i.e. devices that sense things, such as temperature, motion, particles, etc.) and **actuators** (i.e. devices that act on things, such as switches or rotors).

Rarely, though, will a smart solution make do with just one type of an IoT sensor or an actuator. If you think of a smart surgical robot, for example, it will require hundreds, if not thousands, of components that measure different parameters and act accordingly. But even apparently less complicated solutions aren't truly that easy. Consider running a smart farm – for a plant to grow, it's not just a matter of measuring the humidity of the soil, but also its fertility; it's also a matter of providing proper irrigation based on insolation, and much more. So you need not just one, but many sensors and actuators that all have to work together.



When speaking of devices essential for the IoT ecosystem, one cannot forget about IoT gateways. They are a piece of hardware that is capable of “translating” and facilitating the essential connection between devices or between devices and the network and work as a kind of relay for the two. Which brings us to the next element of our puzzle...



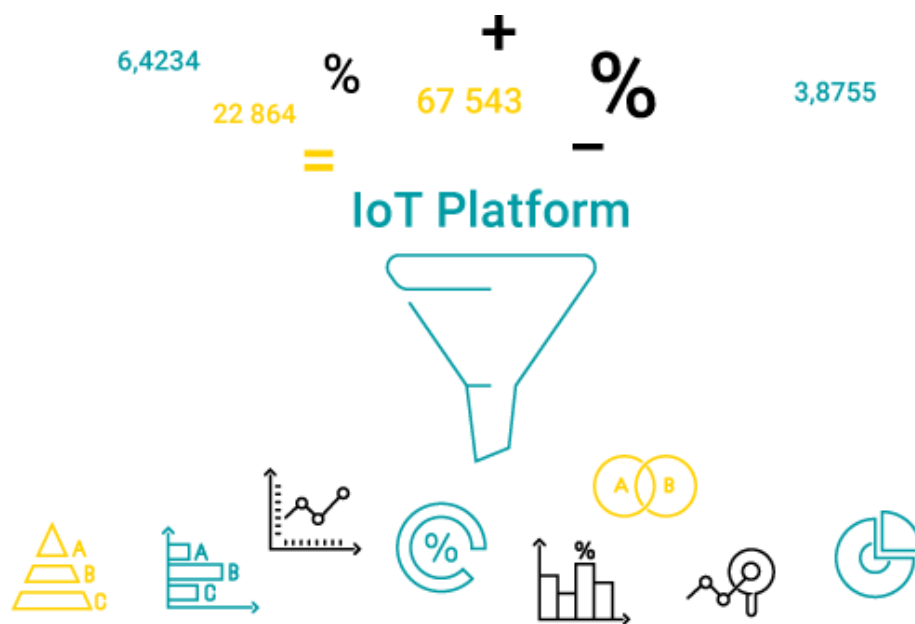
Networks

Based on what you read before, you may think: “Well, if an automatic door senses my presence and opens itself, is that IoT?” Obviously, it is not, because while that door has sensors and actuators, it is not connected to much else. And, as the name suggests, **the Internet of Things requires both things and the Internet (although there are cases of data delivery without the use of the Internet Protocol)**. Arguably, the real power of this concept lies in the connectivity.

Again, based on your deployment needs, there are plenty of different IoT connectivity options, starting with the “classics,” such as WiFi or Bluetooth, to more specialized and field-oriented technologies, such as Low-Power Wide Area Networks (LPWAN). **They all differ in range and speed of data transfer, making them more or less appropriate for particular deployments.** Consider, for example, smart cars that require both high data speed and long range and juxtapose them with the smart farms we’ve mentioned that don’t necessarily need either.

IoT platform

Whether they are in the cloud or not, **IoT platforms are always the binder for any IoT ecosystem.** They are the quiet administrators that take care of device lifecycle management, so that you don’t have to worry about them. They are also the hub that collects and aggregates the data, allowing you to make sense of it. With the variety of platforms offered on the market and the breadth of claims their providers make, **the choice of the “ideal” IoT platform for a deployment is arguably the most significant, yet also the most difficult to make.** It shouldn’t be taken lightly, as it determines whether the IoT ecosystem will thrive or wither into oblivion.



The right IoT device management platform should be versatile and adaptable, as the IoT world is very fragmented and constantly shifting and you don't want the core element of your ecosystem to become the stumbling block of your deployment. **It should also be scalable**, so that your ecosystem can grow naturally, and **secure**, so it can do so without any threats.

Agents

Agents are all the people whose actions affect the IoT ecosystem. These may be the engineers who devise IoT deployments and design the platforms, it can also be the platform operators. But probably, most importantly, it's the stakeholders, who ultimately reap the results. After all, IoT deployments aren't just art for art's sake. **These complex ecosystems are put in place for a reason: to drive efficiency and improve the quality of life. And it is the agents who decide on how to use the devices, networks and platforms to achieve these results.** This is where technology and business converge, because it's business goals that very much shape the IoT ecosystem.



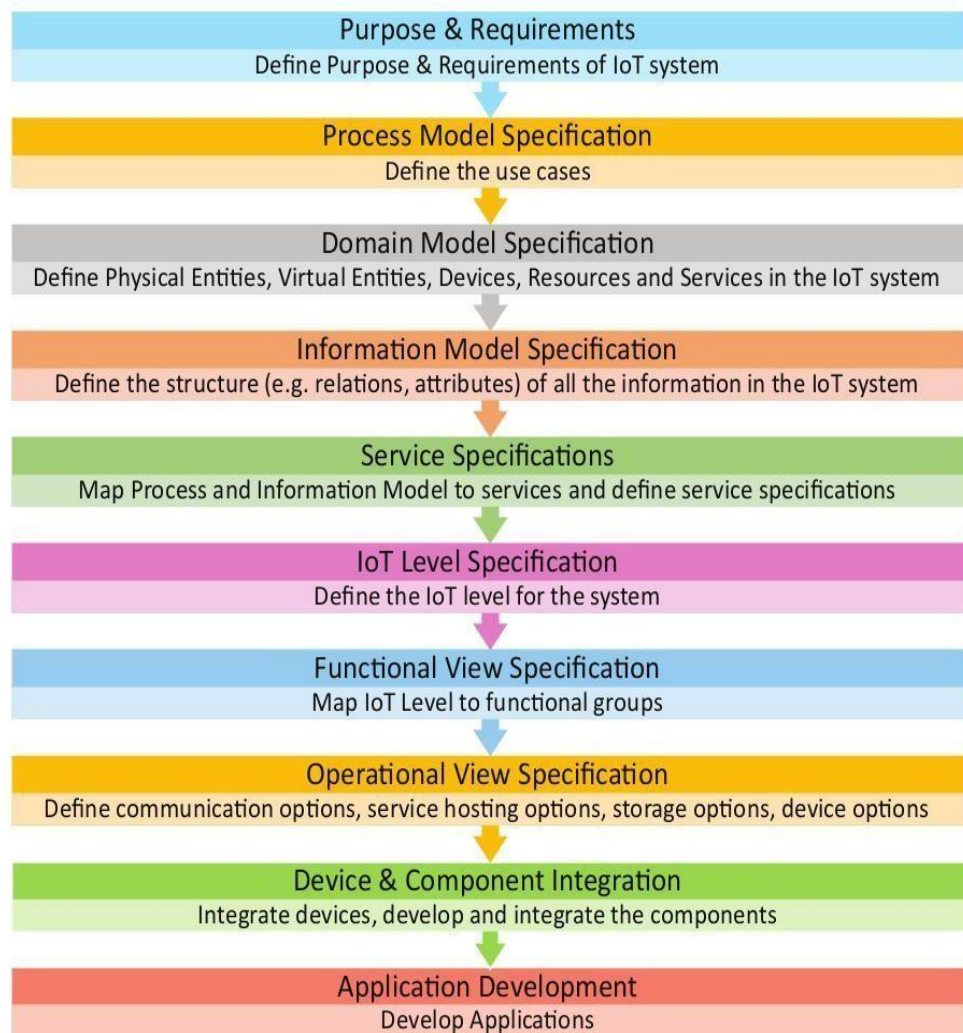
People are an essential part of this equation. Ecosystems are created by us, managed by us and, ultimately, it is our responsibility to realize their full potential. It is the devices that collect the data, but it is the people that make sense of it and put it to use. Similarly with networks and platforms, which are a necessary component of the ecosystem, but wouldn't be of much value if it weren't for the people who create and perfect them to fit their needs.

As was said, **an IoT ecosystem is a very complex concept that eludes easy classification, as its characteristics vary from deployment to deployment.** Much like our world, the IoT world comprises numerous different ecosystems that evolve and adapt. What they have in common is the idea and the people that make them happen: device manufacturers, service providers, application developers, and enterprises. **Yet in this ever changing landscape there still remains a lot of variety – the technology, represented by the devices, networks and platforms, always gets better.** This is particularly worth remembering, because the one mistake the inhabitants of an IoT ecosystem should never make, is to take it for granted. There is nothing more toxic for that landscape than stagnation and lock-in, so you should always be on the lookout for newer, better technologies that will help you flourish.

UNIT-II

DESIGN AND DEVELOPMENT

IoT Design Methodology – Steps



Step 1: Purpose & Requirements Specification • The first step in IoT system design methodology is to define the purpose and requirements of the system. In this step, the system purpose, behavior and requirements (such as data collection requirements, data analysis requirements, system management requirements, data privacy and security requirements, user interface requirements, ...) are captured.

Step 2: Process Specification • The second step in the IoT design methodology is to define the process specification. In this step, the use cases of the IoT system are formally described based on and derived from the purpose and requirement specifications.

Step 3: Domain Model Specification • The third step in the IoT design methodology is to define the Domain Model. The domain model describes the main concepts, entities and objects in the domain of IoT system to be designed. Domain model defines the attributes of the objects and relationships between objects. Domain model provides an abstract representation of the concepts, objects and entities in the IoT domain, independent of any specific technology or platform. With the domain model, the IoT system designers can get an understanding of the IoT domain for which the system is to be designed.

Step 4: Information Model Specification • The fourth step in the IoT design methodology is to define the Information Model. Information Model defines the structure of all the information in the IoT system, for example, attributes of Virtual Entities, relations, etc. Information model does not describe the specifics of how the information is represented or stored. To define the information model, we first list the Virtual Entities defined in the Domain Model. Information model adds more details to the Virtual Entities by defining their attributes and relations.

Step 5: Service Specifications • The fifth step in the IoT design methodology is to define the service specifications. Service specifications define the services in the IoT system, service types, service inputs/output, service endpoints, service schedules, service preconditions and service effects.

Step 6: IoT Level Specification • The sixth step in the IoT design methodology is to define the IoT level for the system.

Step 7: Functional View Specification • The seventh step in the IoT design methodology is to define the Functional View. The Functional View (FV) defines the functions of the IoT systems grouped into various Functional Groups (FGs). Each Functional Group either provides functionalities for interacting with instances of concepts defined in the Domain Model or provides information related to these concepts.

Step 8: Operational View Specification • The eighth step in the IoT design methodology is to define the Operational View Specifications. In this step, various options pertaining to the IoT system deployment and operation are defined, such as, service hosting options, storage options, device options, application hosting options, etc

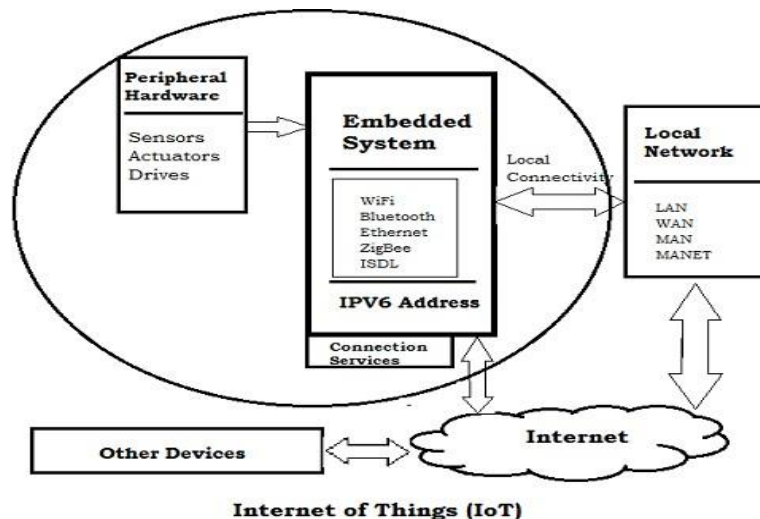
Step 9: Device & Component Integration • The ninth step in the IoT design methodology is the integration of the devices and components.

Step 10: Application Development • The final step in the IoT design methodology is to develop the IoT application.

Embedded Computing Logic

It is essential to know about the embedded devices while learning the IoT or building the projects on IoT. The embedded devices are the objects that build the unique computing system. These systems may or may not connect to the Internet.

An embedded device system generally runs as a single application. However, these devices can connect through the internet connection, and able communicate through other network devices.



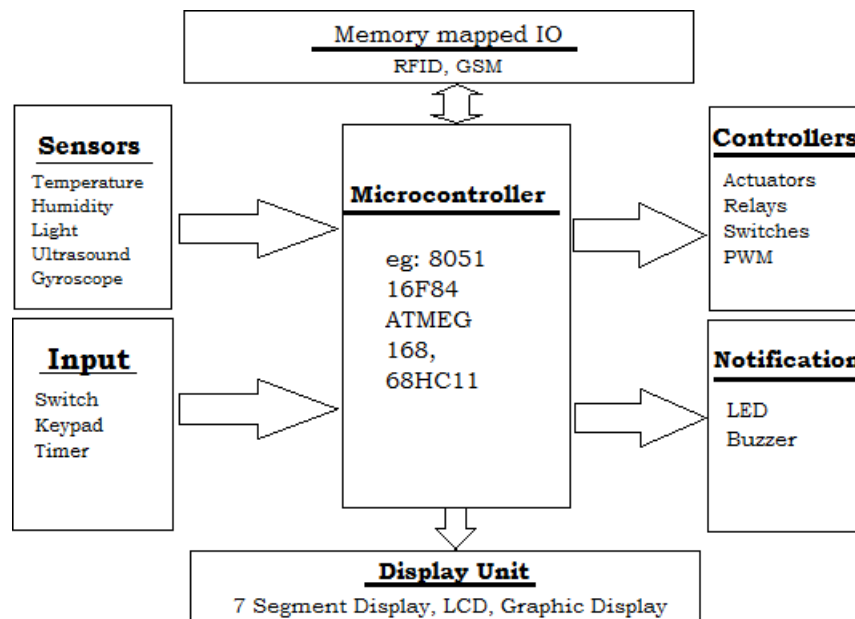
First developed in the 1960s for aerospace and the military, embedded computing systems continue to support new applications through numerous feature enhancements and cost-to-performance improvements of microcontrollers and programmable logic devices. Today, embedded computing systems control everyday devices which we don't generally think of as "computers": digital cameras, automobiles, smart watches, home appliances, and even smart garments. These embedded computing systems are commonly found in consumer, industrial, automotive, medical, commercial, and military applications.

Unlike general-purpose computers, embedded control systems are typically designed to perform specific tasks. The embedded computing system designer's task is to identify the set of components that will implement the system's functional, performance, usability, and reliability requirements, typically within tight cost and development timeline constraints. Accordingly, the selection of a microcontroller and its characteristics, including data processing capabilities, speed, peripherals, and power consumption, is one of the earliest and most critical aspects of system design.

Part of the designer's responsibility involves being aware of trends in their particular industry and taking advantage of relevant components and techniques. Let's look for examples among the top industries for microcontroller applications, the Internet of Things.

Embedded System Hardware

The embedded system can be of type microcontroller or type microprocessor. Both of these types contain an integrated circuit (IC). The essential component of the embedded system is a RISC family microcontroller like Motorola 68HC11, PIC 16F84, Atmel 8051 and many more. The most important factor that differentiates these microcontrollers with the microprocessor like 8085 is their internal read and writable memory. The essential embedded device components and system architecture are specified below.



Embedded System Software

The embedded system that uses the devices for the operating system is based on the language platform, mainly where the real-time operation would be performed. Manufacturers build embedded software in electronics, e.g., cars, telephones, modems, appliances, etc. The embedded system software can be as simple as lighting controls running using an 8-bit microcontroller. It can also be complicated software for missiles, process control systems, airplanes etc.

Microcontrollers for Embedded Computing with IoT Devices

IoT devices are meant to be inexpensive, therefore the microcontroller needs to be chosen so that its capabilities are not underutilized by the application. The microcontroller specifications that determine the best part for your application are:

- **Bit depth:** The register and data path width impacts the speed and accuracy with which microcontrollers can perform non-trivial computations.
- **Memory:** The amount of RAM and Flash in a microcontroller determines the code size and complexity the component can support at full speed. Large memories have larger die area and component cost.

- **GPIO:** These are the microcontroller pins used to connect to sensors and actuators in the system. These often share their functionality with other microcontroller peripherals, such as serial communication, A/D, and D/A converters.
- **Power consumption:** Power consumption is critically important for battery-operated devices and it typically increases with microcontroller speed and memory size.

System on Chips

System on Chip in IoT designed by Redpine Signals is discussed below. This IoT SoC supports WLAN, bluetooth and Zigbee systems on a single chip. It also supports 2.4 and 5GHz radio frequencies.

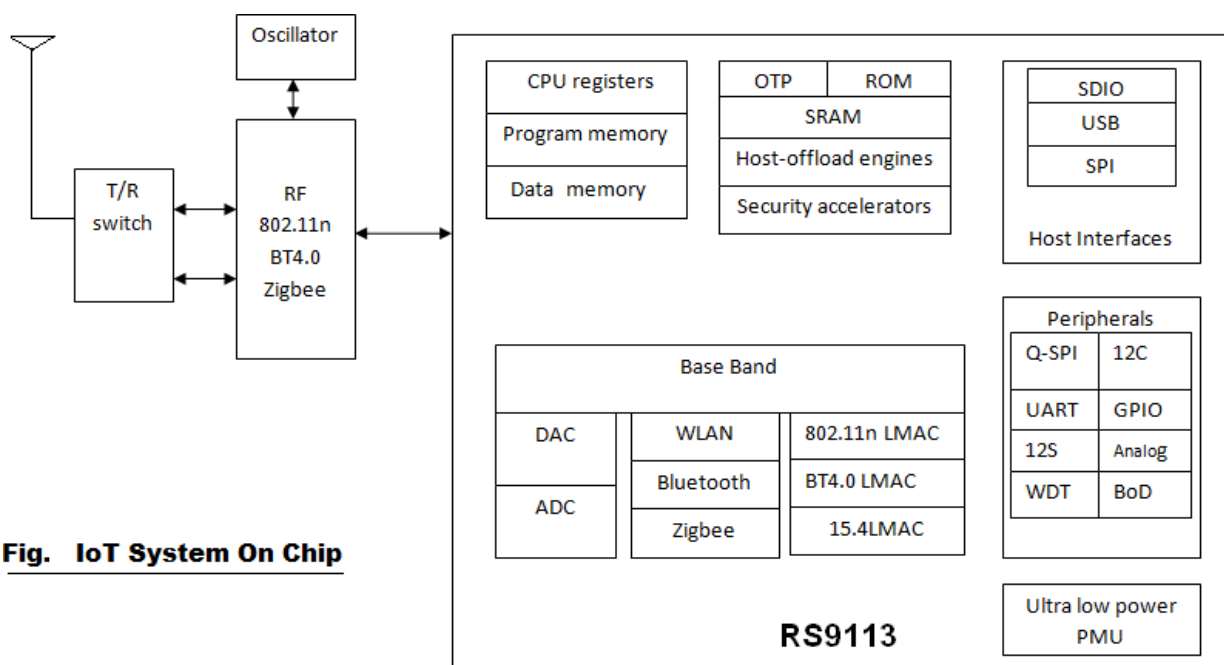


Fig. IoT System On Chip

As we know IoT is the technology which will provide communication between things, between things and people using internet and IP enabled protocols. As we have seen in IoT tutorial any IoT compliant system will have two major parts viz. front end and back end. Front end provides connectivity with physical world and consists of sensors while backend consists of processing and network connectivity interfaces.

Typical **IoT system on chip** support more than one RATs (Radio Access Technologies). It will have following modules.

- Transmit and receive switch.

- RF part mainly consists of Transmitter, receiver, oscillator and amplifiers.
- Memories i.e. Program memory, data memory to store the code and data
- Physical layer (baseband processing) either on FPGA or on processor based on complexity and latency requirement.
- MAC layer and upper protocol stacks TCP/IP etc. running on processor
- ADC and DAC to provide interface between digital baseband and analog RF portions.
- Various interfaces such as SDIO, USB, SPI etc to provide interface with the host.
- Other peripherals such as UART, I²C, GPIO, WDT etc. to use the IoT SoC for various connections.

As IoT system on chip supports multiple wireless protocols and RF hardware to support multiple frequency bands, following factors need to be carefully analyzed and to be optimized.

- Power-consumption
- Data-throughput
- Device-size
- Performance in terms of latency and other factors

Figure depicts one such IoT System on Chip model no. RS9113, which has been designed and developed by Redpine Signals recently. It supports WLAN (802.11n), Bluetooth version 4.0 and Zigbee (802.15.4-2006) in the same chip. Hence the IoT device can be connected with any of the said wireless technology based networks.

This IoT SoC (system on chip in IoT) can be used for numerous applications as mentioned below:

- Mobile
- M2M-Communication
- Real time location finding tags
- Thermostats
- Smart meters
- Wireless sensor devices
- Serial to WiFi converter
- Voice Over WiFi compliant phones
- Home automation
- Health care devices and equipments

Building Blocks Of IoT

Four things form basic building blocks of the IoT system –sensors, processors, gateways, applications. Each of these nodes has to have its own characteristics in order to form an useful IoT system.

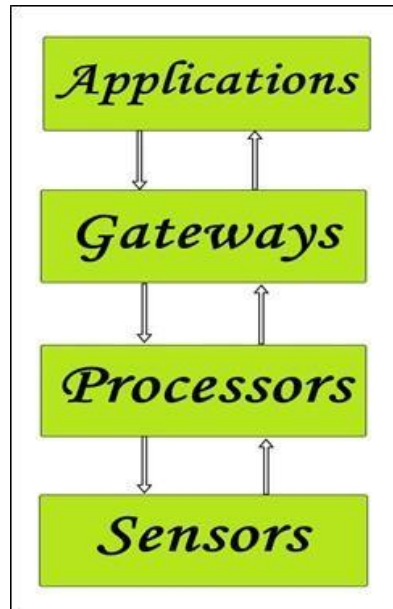


Figure: Simplified block diagram of the basic building blocks of the IoT

Sensors:

- These form the front end of the IoT devices. These are the so-called “Things” of the system. Their main purpose is to collect data from its surroundings (sensors) or give out data to its surrounding (actuators).
- These have to be uniquely identifiable devices with a unique IP address so that they can be easily identifiable over a large network.
- These have to be active in nature which means that they should be able to collect real-time data. These can either work on their own (autonomous in nature) or can be made to work by the user depending on their needs (user-controlled).
- Examples of sensors are gas sensor, water quality sensor, moisture sensor, etc.

Processors:

- Processors are the brain of the IoT system. Their main function is to process the data captured by the sensors and process them so as to extract the valuable data from the enormous amount of raw data collected. In a word, we can say that it gives intelligence to the data.
- Processors mostly work on real-time basis and can be easily controlled by applications. These are also responsible for securing the data – that is performing encryption and decryption of data.
- Embedded hardware devices, microcontroller, etc are the ones that process the data because they have processors attached to it.

Gateways:

- Gateways are responsible for routing the processed data and send it to proper locations for its (data) proper utilization.
- In other words, we can say that gateway helps in to and fro communication of the data. It provides network connectivity to the data. Network connectivity is essential for any IoT system to communicate.
- LAN, WAN, PAN, etc are examples of network gateways.

Applications:

- Applications form another end of an IoT system. Applications are essential for proper utilization of all the data collected.
- These cloud-based applications which are responsible for rendering the effective meaning to the data collected. Applications are controlled by users and are a delivery point of particular services.
- Examples of applications are home automation apps, security systems, industrial control hub, etc.

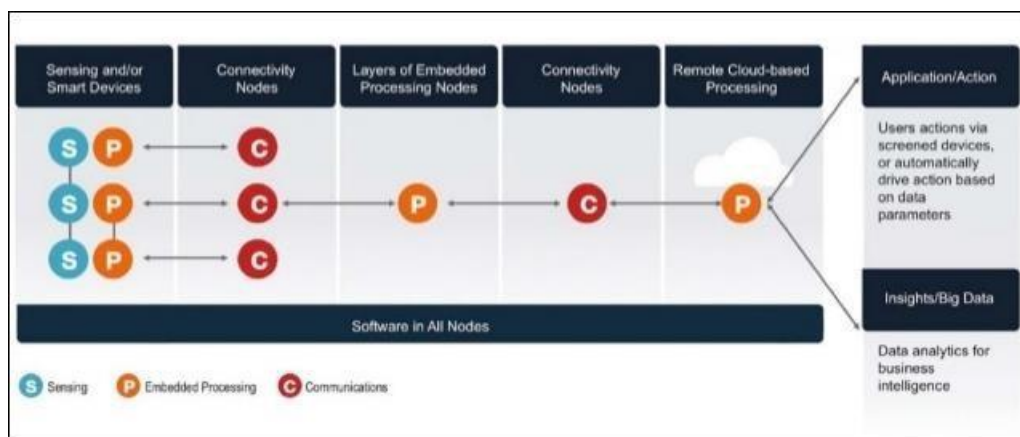


Figure : Basic building blocks of IoT

In a nutshell, from the figure we can determine that the information gathered by the sensing node (end node) is processed first then via connectivity it reaches the embedded processing nodes that can be any embedded hardware devices and are processed there as well. It then passes through the connectivity nodes again and reaches the remote cloud-based processing that can be any software and is sent to the application node for the proper applied usage of the data collected and also for data analysis via big data.

HOW IoT WORKS

How the IoT works is quite simple.

First, it acquires information with respect to basic resources (names, addresses and so on) and related attributes of objects by means of automatic identification and perception technologies such as RFID, wireless sensor and satellite positioning, in other words, the

sensors, RFID tags, and all other uniquely identifiable objects or "things" acquire real-time information (data) with the virtue of a central hub like smartphones.

Second, by virtue of many kinds of communications technologies, it integrates object-related information into the information network and realizes the intelligent indexing and integration of the information related to masses of objects by resorting to fundamental resource services (similar to the resolution, addressing and discovery of the internet).

Finally, utilizing intelligent computing technologies such as cloud computing, fuzzy recognition, data mining, and semantic analysis, it analyzes and processes the information related to masses of objects so as to eventually realize intelligent decision and control in the physical world.

Let's have a look at the following diagram.

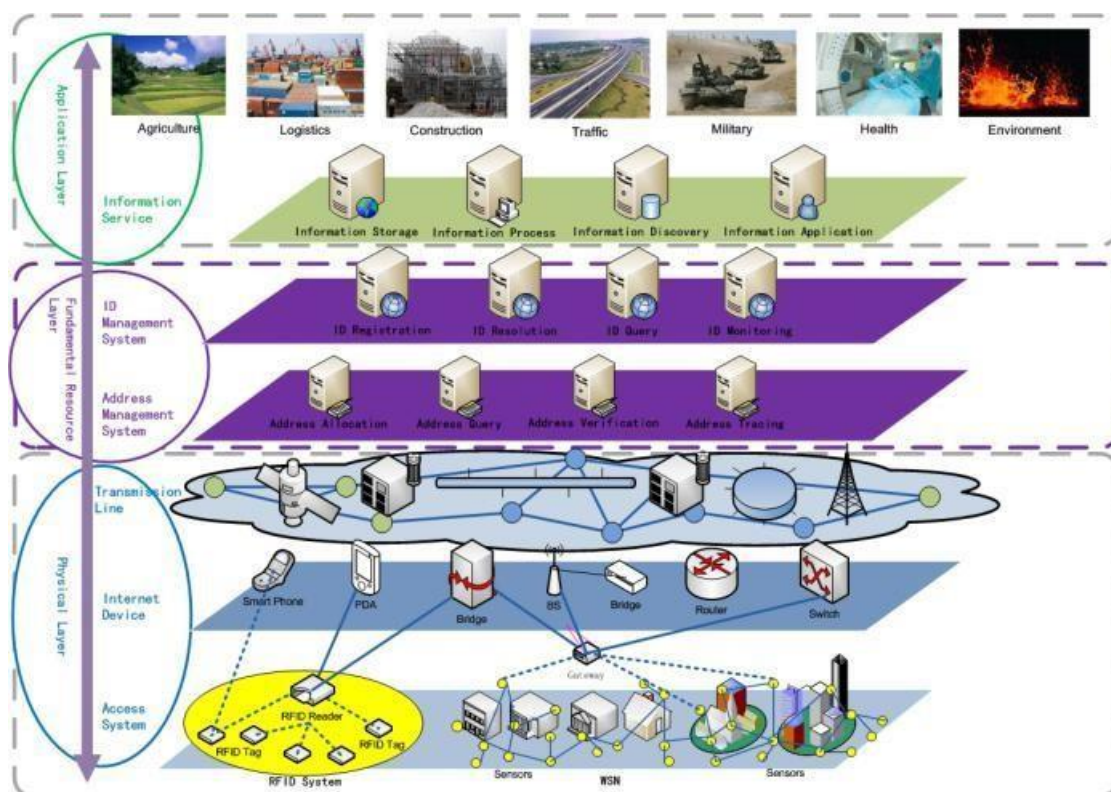
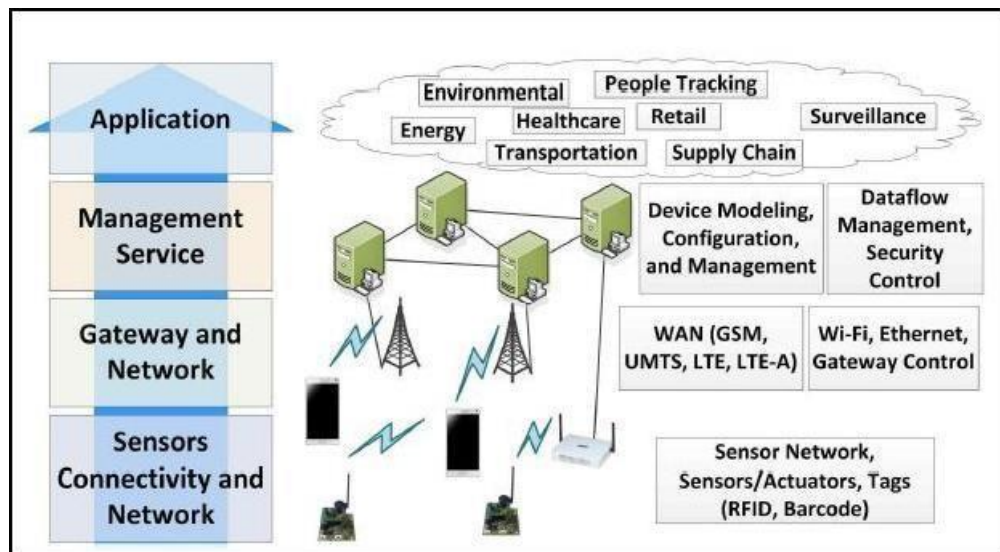


Figure : Layers of the IoT

In the Physical layer, all the data collected by the access system (uniquely identifiable "things") collect data and go to the internet devices (like smartphones). Then via transmission lines (like fiber-optic cable) it goes to the management layer where all the data is managed separately (stream analytics and data analytics) from the raw data. Then all the managed information is released to the application layer for proper utilization of the data collected.

IoT Architecture Layers

There are four major layers.



At the very bottom of IoT architecture, we start with the Sensors and Connectivity network which collects information. Then we have the Gateway and Network Layer. Above which we have the Management Service layer and then at the end, we have the application layer where the data collected are processed according to the needs of various applications.

Let's discuss the features of each of these architectural layers separately.

Sensor, Connectivity and Network Layer

- This layer consists of RFID tags, sensors (which are an essential part of an IoT system and are responsible for collecting raw data). These form the essential “things” of an IoT system.
- Sensors, RFID tags are wireless devices and form the Wireless Sensor Networks (WSN).
- Sensors are active in nature which means that real-time information is to be collected and processed.
- This layer also has the network connectivity (like WAN, PAN, etc.) which is responsible for communicating the raw data to the next layer which is the Gateway and Network Layer.
- The devices which are comprised of WSN have finite storage capacity, restricted communication bandwidth and have small processing speed.
- We have different sensors for different applications – temperature sensor for collecting temperature data, water quality for examining water quality, moisture sensor for measuring moisture content of the atmosphere or soil, etc.

As per the figure below, at the bottom of this layer, we have the tags which are the RFID tags or barcode reader, above which we have the sensors/actuators and then the communication networks.

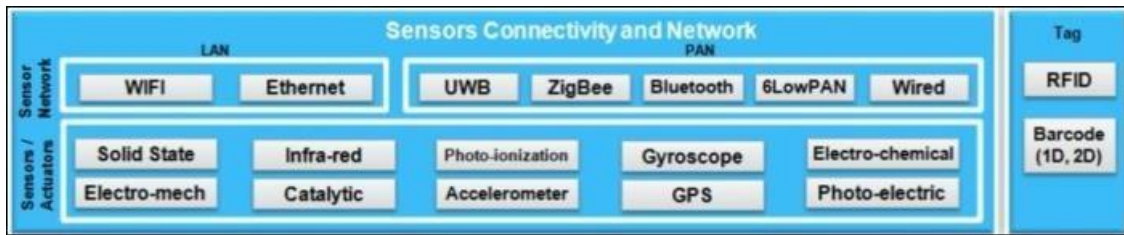


Figure : Sensor, Connectivity and Network Layer

Gateway and Network Layer

- Gateways are responsible for routing the data coming from the **Sensor, Connectivity and Network layer** and pass it to the next layer which is the **Management Service Layer**.
- This layer requires having a large storage capacity for storing the enormous amount of data collected by the sensors, RFID tags, etc. Also, this layer needs to have a consistently trusted performance in terms of public, private and hybrid networks.
- Different IoT device works on different kinds of network protocols. All these protocols are required to be assimilated into a single layer. This layer is responsible for integrating various network protocols.

From the figure below, at the bottom, we have the gateway which is comprised of the embedded OS, Signal Processors, and Modulators, Micro-Controllers etc. Above the gateway we have the Gateway Networks which are LAN(Local Area Network), WAN(Wide Area Network), etc.

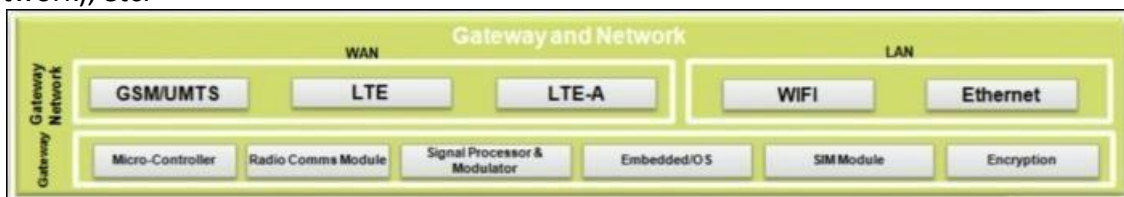


Figure : Gateway and Network Layer

Management Service Layer

- This layer is used for managing IoT services. The management Service layer is responsible for Securing Analysis of IoT devices, Analysis of Information (Stream Analytics, Data Analytics), Device Management.
- Data management is required to extract the necessary information from the enormous amount of raw data collected by the sensor devices to yield a valuable result of all the data collected. This action is performed in this layer.
- Also, a certain situation requires an immediate response to the situation. This layer helps in doing that by abstracting data, extracting information and managing the data flow.
- This layer is also responsible for data mining, text mining, service analytics, etc.

From the figure below, we can see that, management service layer has Operational Support Service (OSS) which includes Device Modeling, Device Configuration and Management and many more. Also, we have the Billing Support System (BSS) which supports billing and reporting.

Also, from the figure, we can see that there are IoT/M2M Application Services which includes Analytics Platform; Data – which is the most important part; Security which includes Access Controls, Encryption, Identity Access Management, etc. ; and then we have the Business Rule Management (BRM) and Business Process Management (BPM).



Figure : Management Service Layer

Application Layer

- Application layer forms the topmost layer of IoT architecture which is responsible for effective utilization of the data collected.
- Various IoT applications include Home Automation, E-health, E-Government, etc.
- From the figure below, we can see that there are two types of applications which are Horizontal Market which includes Fleet Management, Supply Chain, etc. and on the Sector-wise application of IoT we have energy, healthcare, transportation, etc.



Figure : Application Layer

IoT Platform

An IoT platform is a multi-layer technology that enables straightforward provisioning, management, and automation of connected devices within the Internet of Things universe. It basically connects your hardware, however diverse, to the cloud by using flexible connectivity options, enterprise-grade security mechanisms, and broad data processing powers. For developers, an IoT platform provides a set of ready-to-use features that greatly speed up development of applications for connected devices as well as take care of scalability and cross-device compatibility.

Thus, an IoT platform can be wearing different hats depending on how you look at it. It is commonly referred to as middleware when we talk about how it connects remote devices

to user applications (or other devices) and manages all the interactions between the hardware and the application layers. It is also known as a cloud enablement platform or IoT enablement platform to pinpoint its major business value, that is empowering standard devices with cloud-based applications and services. Finally, under the name of the IoT application enablement platform, it shifts the focus to being a key tool for IoT developers.

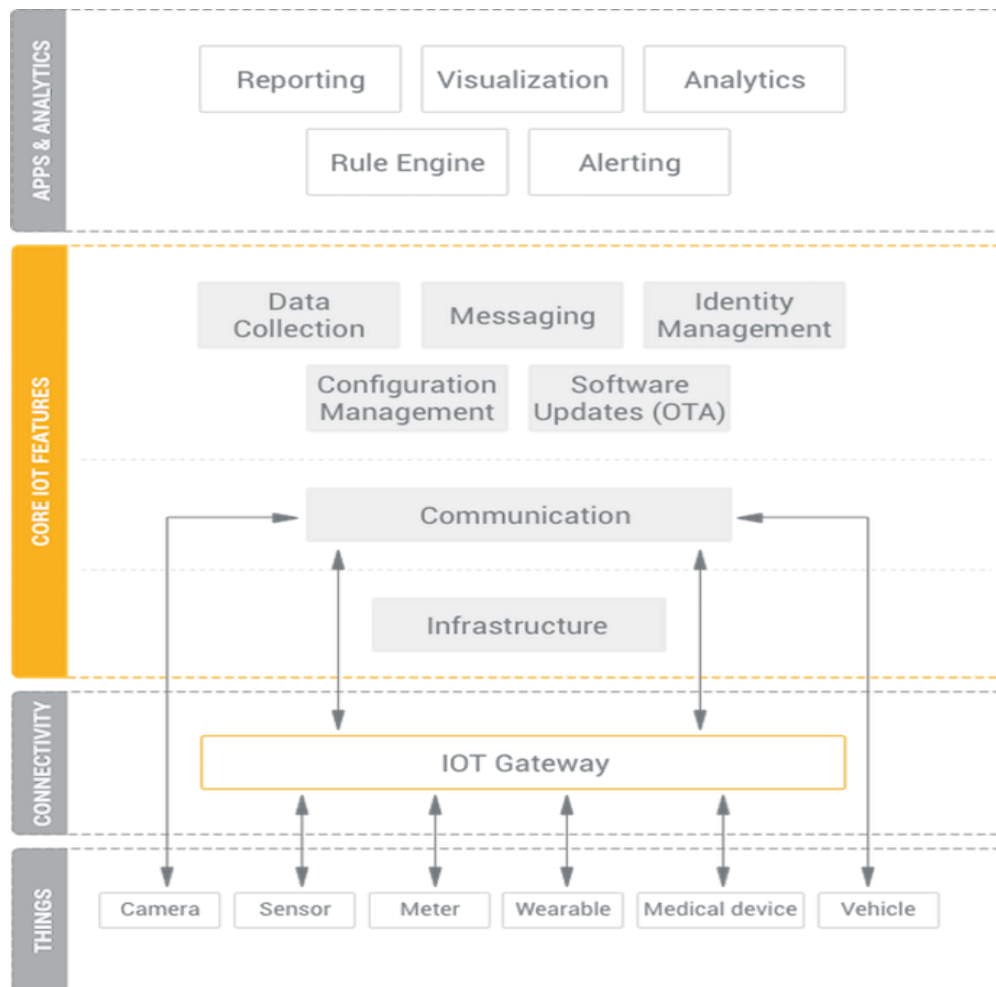
IoT platform as the middleware

IoT platforms originated in the form of IoT middleware, which purpose was to function as a mediator between the hardware and application layers. Its primary tasks included data collection from the devices over different protocols and network topologies, remote device configuration and control, device management, and over-the-air firmware updates.

To be used in real-life heterogeneous IoT ecosystems, IoT middleware is expected to support integration with almost any connected device and blend in with third-party applications used by the device. This independence from underlying hardware and overhanging software allows a single IoT platform to manage any kind of connected device in the same straightforward way.



Modern IoT platforms go further and introduce a variety of valuable features into the hardware and application layers as well. They provide components for frontend and analytics, on-device data processing, and cloud-based deployment. Some of them can handle end-to-end IoT solution implementation from the ground up.



IoT platform technology stack

In the four typical layers of the IoT stack, which are things, connectivity, core IoT features, and applications & analytics, a top-of-the-range IoT platform should provide you with the majority of IoT functionality needed for developing your connected devices and smart things.

Your devices connect to the platform, which sits in the cloud or in your on-premises data center, either directly or by using an IoT gateway. A gateway comes useful whenever your endpoints aren't capable of direct cloud communication or, for example, you need some computing power on edge. You can also use an IoT gateway to convert protocols, for example, when your endpoints are in LoRaWan network but you need them to communicate with the cloud over MQTT.

An IoT platform itself can be decomposed into several layers. At the bottom there is the infrastructure level, which is something that enables the functioning of the platform. You

can find here components for container management, internal platform messaging, orchestration of IoT solution clusters, and others.

The communication layer enables messaging for the devices; in other words, this is where devices connect to the cloud to perform different operations.

The following layer represents core IoT features provided by the platform. Among the essential ones are data collection, device management, configuration management, messaging, and OTA software updates.

Sitting on top of core IoT features, there is another layer, which is less related to data exchange between devices but rather to processing of this data in the platform. There is reporting, which allows you to generate custom reports. There is visualization for data representation in user applications. Then, there are a rule engine, analytics, and alerting for notifying you about any anomalies detected in your IoT solution.

Importantly, the best IoT platforms allow you to add your own industry-specific components and third-party applications. Without such flexibility adapting an IoT platform for a particular business scenario could bear significant extra cost and delay the solution delivery indefinitely.

Advanced IoT platforms

There are some other important criteria that differentiate IoT platforms between each other, such as scalability, customizability, ease of use, code control, integration with 3rd party software, deployment options, and the data security level.

- **Scalable (cloud native)** – advanced IoT platforms ensure elastic scalability across any number of endpoints that the client may require. This capability is taken for granted for public cloud deployments but it should be specifically put to the test in case of an on-premises deployment, including the platform's load balancing capabilities for maximized performance of the server cluster.
- **Customizable** – a crucial factor for the speed of delivery. It closely relates to flexibility of integration APIs, loose coupling of the platform's components, and source code transparency. For small-scale, undemanding IoT solutions good APIs may be enough to fly, while feature-rich, rapidly evolving IoT ecosystems usually require developers to have a greater degree of control over the entire system, its source code, integration interfaces, deployment options, data schemas, connectivity and security mechanisms, etc.
- **Secure** – data security involves encryption, comprehensive identity management, and flexible deployment. End-to-end data flow encryption, including data at rest, device authentication, user access rights management, and private cloud infrastructure for sensitive data – this is the basics of how to avoid potentially compromising breaches in your IoT solution.

Cutting across these aspects, there are two different paradigms of IoT solution cluster deployment offered by IoT platform providers: a public cloud IoT PaaS and a self-hosted private IoT cloud.

IoT cloud enablement

An IoT cloud is a pinnacle of the IoT platforms evolution. Sometimes these two terms are used interchangeably, in which case the system at hand is typically an IoT platform-as-a-service (PaaS). This type of solution allows you to rent cloud infrastructure and an IoT platform all from a single technology provider. Also, there might be ready-to-use IoT solutions (IoT cloud services) offered by the provider, built and hosted on its infrastructure. However, one important capability of a modern IoT platform consists in a private IoT cloud enablement. As opposed to public PaaS solutions located at a provider's cloud, a private IoT cloud can be hosted on any cloud infrastructure, including a private data center. This type of deployment offers much greater control over the new features development, customization, and third-party integrations. It is also advocated for stringent data security and performance requirements.

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are –

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface

(hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Here is a list of different Arduino boards available.

Arduino boards based on ATMEGA328 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Uno R3	5V	16MHz	14	6	6	1	USB via ATmega16U2
Arduino Uno R3 SMD	5V	16MHz	14	6	6	1	USB via ATmega16U2
Red Board	5V	16MHz	14	6	6	1	USB via FTDI
Arduino Pro 3.3v/8 MHz	3.3V	8MHz	14	6	6	1	FTDI-Compatible Header
Arduino Pro 5V/16MHz	5V	16MHz	14	6	6	1	FTDI-Compatible Header
Arduino mini 05	5V	16MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 3.3v/8mhz	3.3V	8MHz	14	8	6	1	FTDI-Compatible Header
Arduino Pro mini 5v/16mhz	5V	16MHz	14	8	6	1	FTDI-Compatible

							Header
Arduino Ethernet	5V	16MHz	14	6	6	1	FTDI- Compatible Header
Arduino Fio	3.3V	8MHz	14	8	6	1	FTDI- Compatible Header
LilyPad Arduino 328 main board	3.3V	8MHz	14	6	6	1	FTDI- Compatible Header
LilyPad Arduino simple board	3.3V	8MHz	9	4	5	0	FTDI- Compatible Header

Arduino boards based on ATMEGA32u4 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Leonardo	5V	16MHz	20	12	7	1	Native USB
Pro micro 5V/16MHz	5V	16MHz	14	6	6	1	Native USB
Pro micro 3.3V/8MHz	5V	16MHz	14	6	6	1	Native USB
LilyPad Arduino USB	3.3V	8MHz	14	6	6	1	Native USB

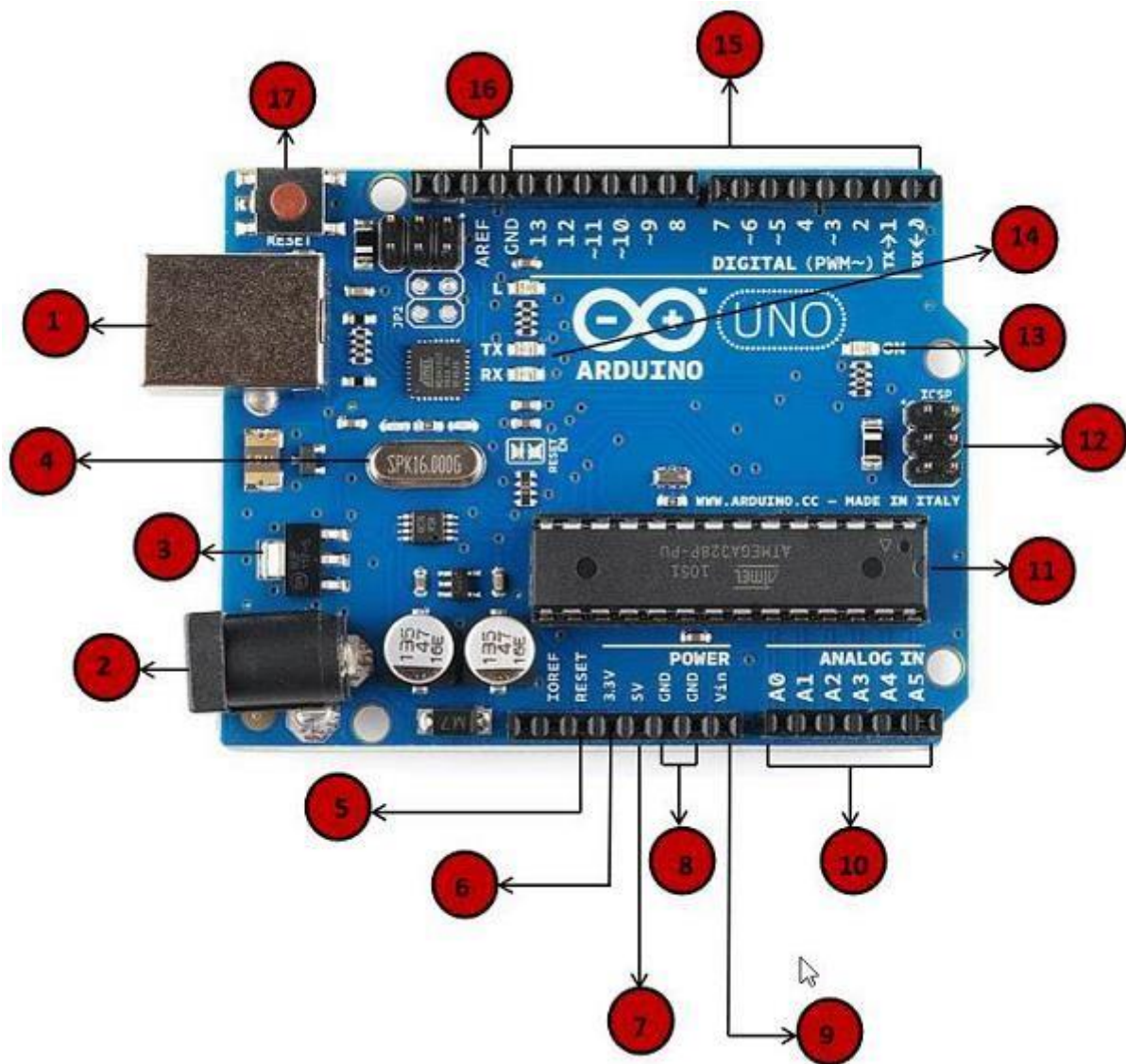
Arduino boards based on ATMEGA2560 microcontroller

Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	5V	16MHz	54	16	14	4	USB via ATmega16U2
Mega Pro 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro 5V	5V	16MHz	54	16	14	4	FTDI-Compatible Header
Mega Pro Mini 3.3V	3.3V	8MHz	54	16	14	4	FTDI-Compatible Header






Arduino boards based on AT91SAM3X8E microcontroller






Board Name	Operating Volt	Clock Speed	Digital i/o	Analog Inputs	PWM	UART	Programming Interface
Arduino Mega 2560 R3	3.3V	84MHz	54	12	12	4	USB native

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.



1	Power USB Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	Power (Barrel Jack) Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	Voltage Regulator The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

	<p>Crystal Oscillator</p> <p>The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.</p>
	<p>Arduino Reset</p> <p>You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).</p>
	<p>Pins (3.3, 5, GND, Vin)</p> <ul style="list-style-type: none"> • 3.3V (6) – Supply 3.3 output volt • 5V (7) – Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) – There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) – This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
	<p>Analog pins</p> <p>The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.</p>
	<p>Main microcontroller</p> <p>Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.</p>

	<p>ICSP pin</p> <p>Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.</p>
	<p>Power LED indicator</p> <p>This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.</p>
	<p>TX and RX LEDs</p> <p>On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.</p>
	<p>Digital I/O</p> <p>The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.</p>
	<p>AREF</p> <p>AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.</p>

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 – First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega

2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

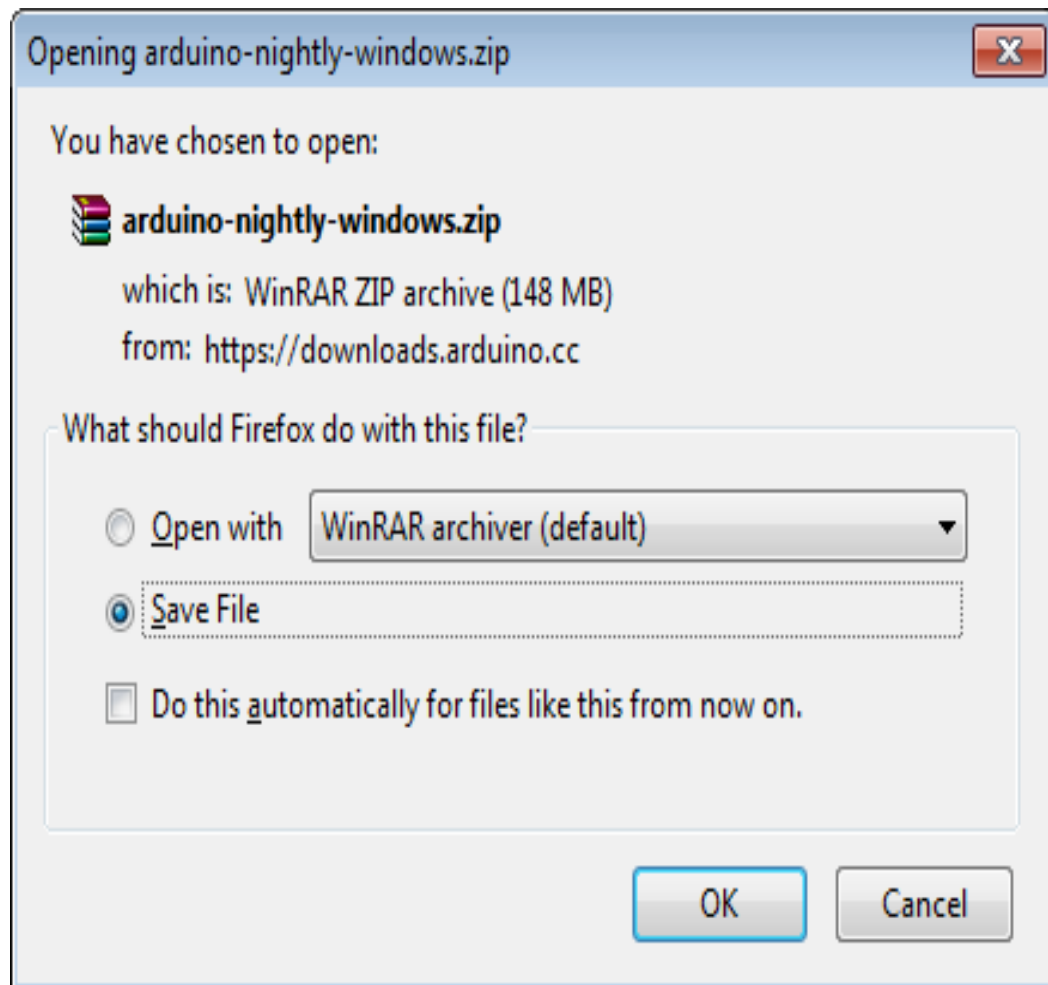


In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.



Step 2 – Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.



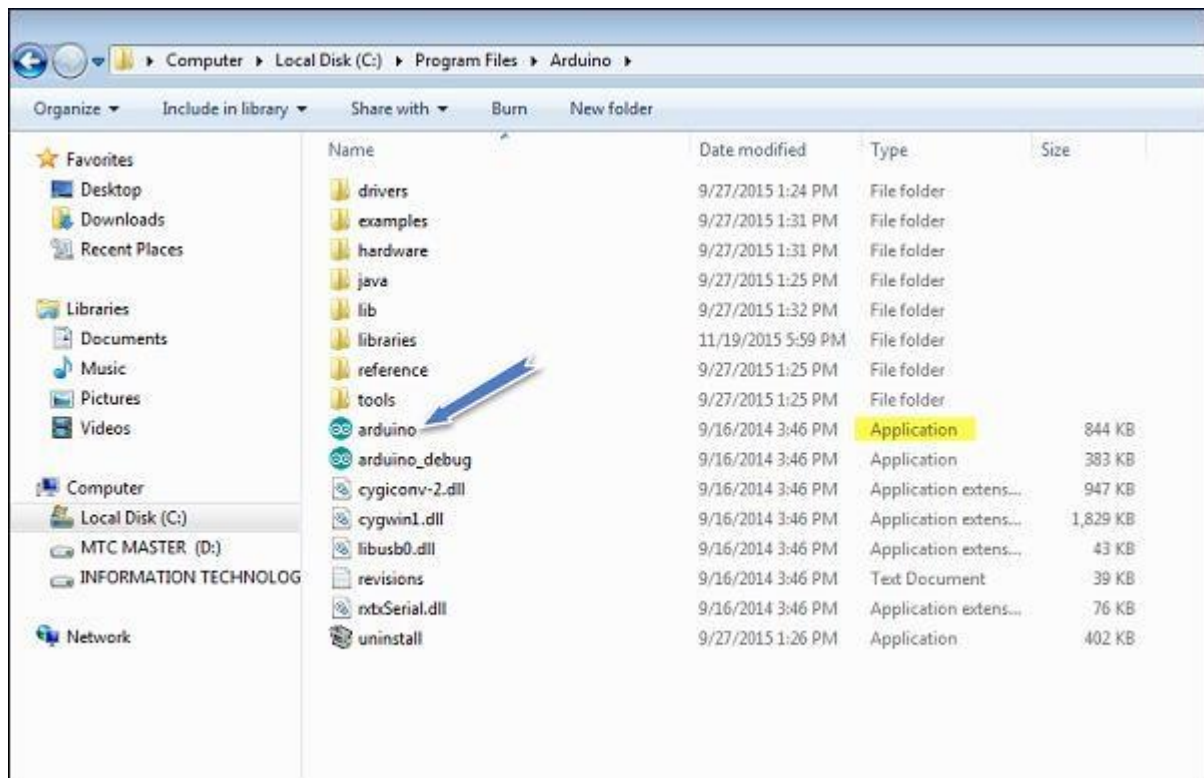
Step 3 – Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 – Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

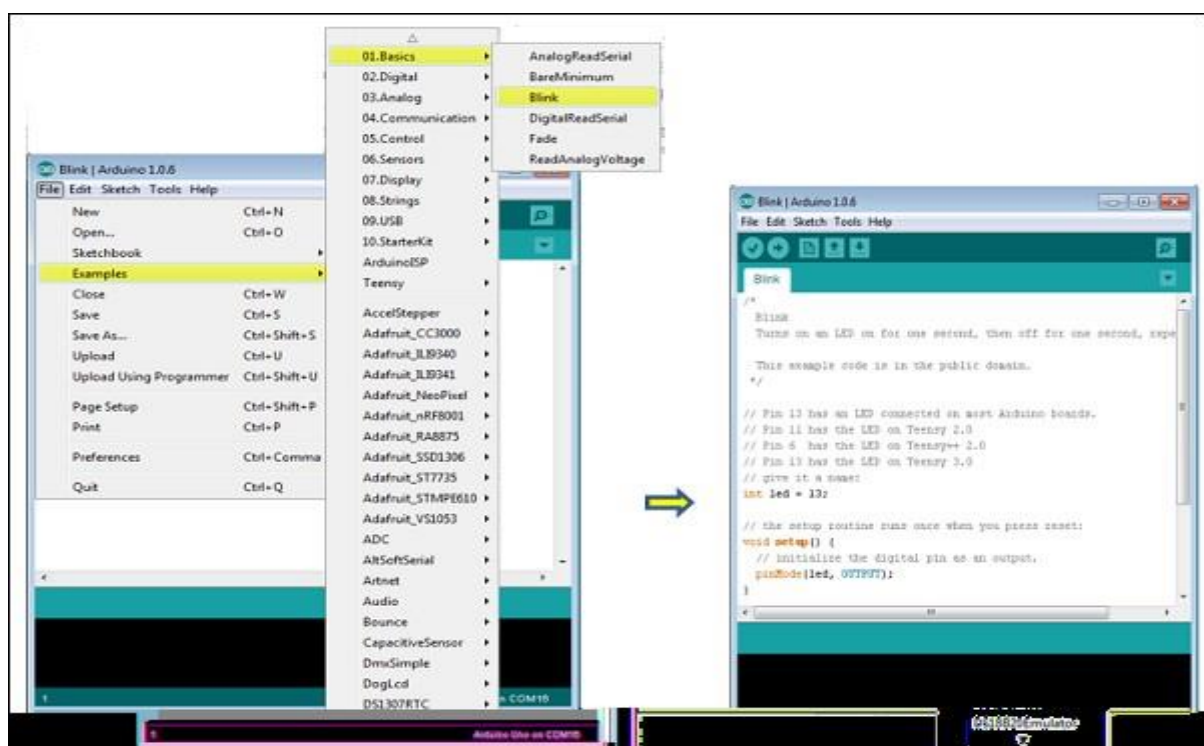


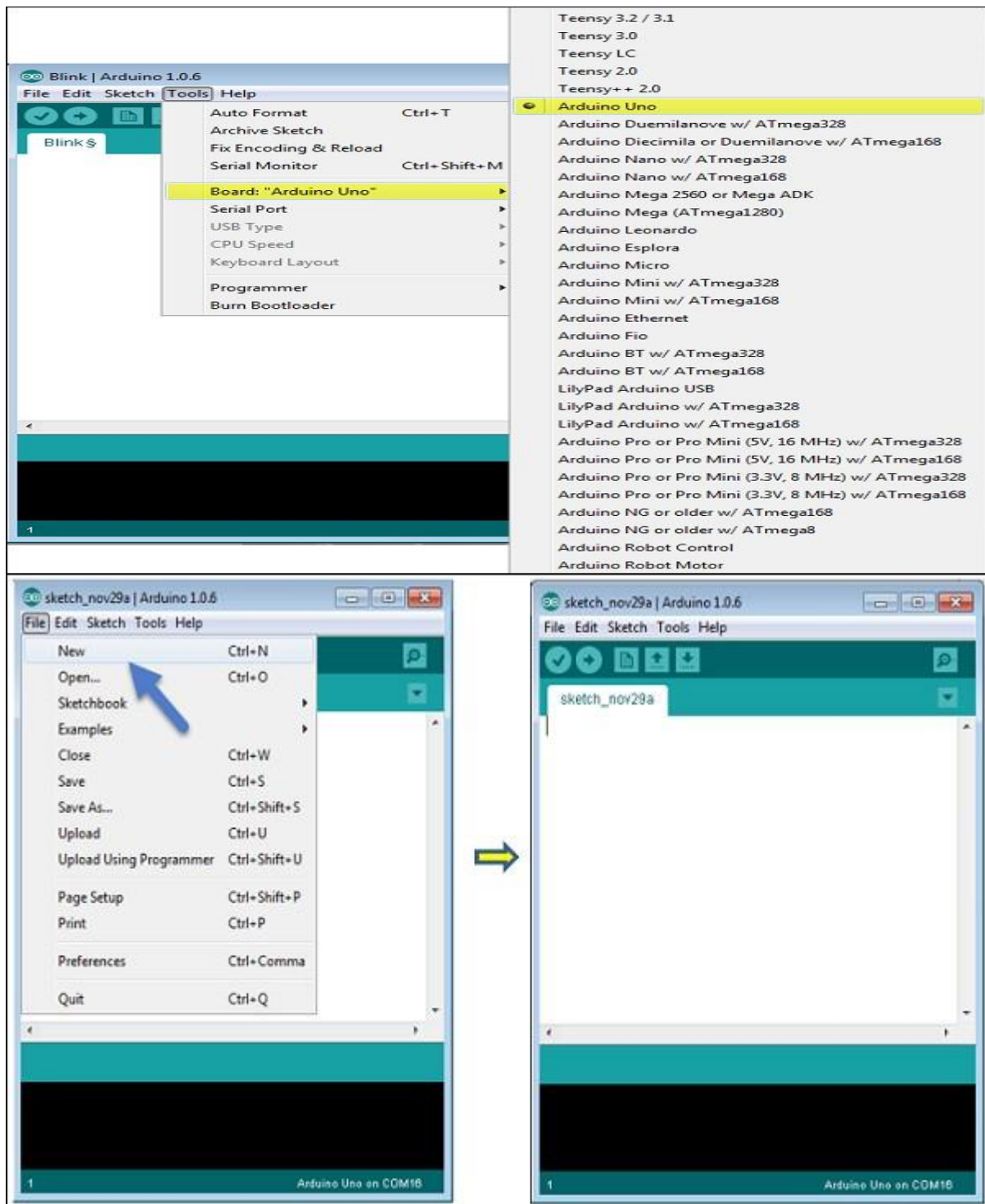
Step 5 – Open your first project.

Once the software starts, you have two options –

- Create a new project.
- Open an existing project example.

To create a new project, select File → New.





To open an existing project example, select File → Example → Basics → Blink.

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 – Select your Arduino board.

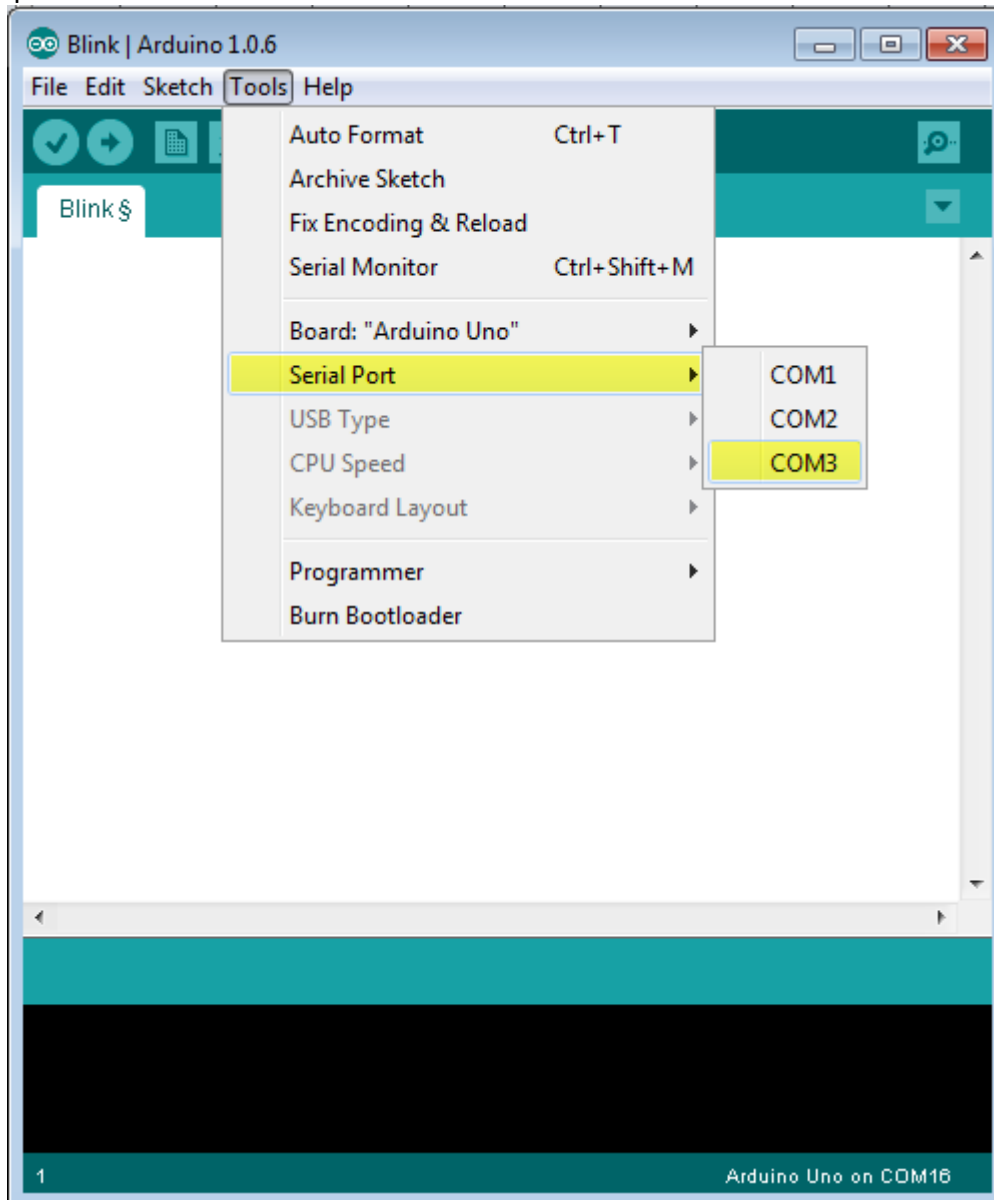
To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

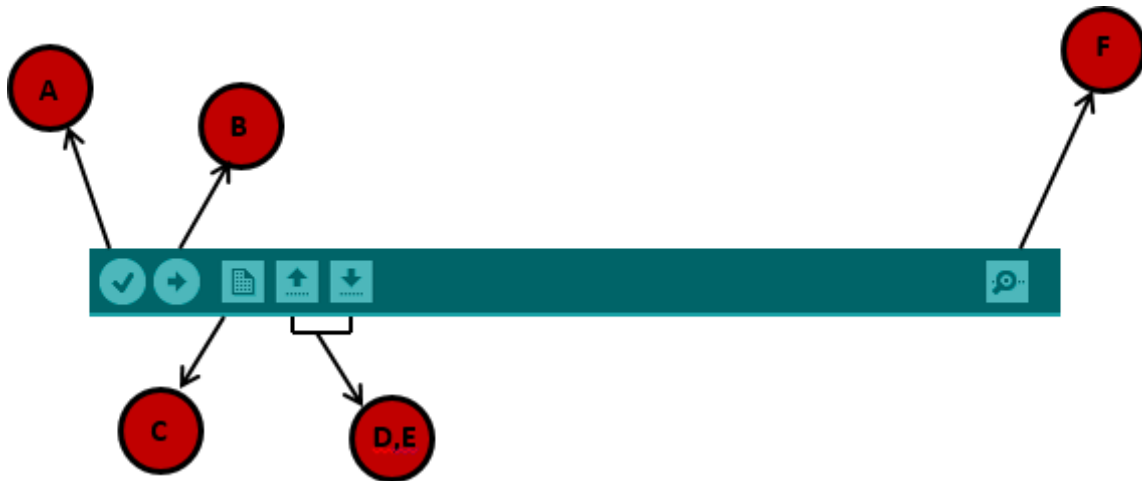
Step 7 – Select your serial port.

Select the serial device of the Arduino board. Go to **Tools → Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.



Step 8 – Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.



A – Used to check if there is any compilation error.

B – Used to upload a program to the Arduino board.

C – Shortcut used to create a new sketch.

D – Used to directly open one of the example sketch.

E – Used to save your sketch.

F – Serial monitor used to receive serial data from the board and send the serial data to the board.

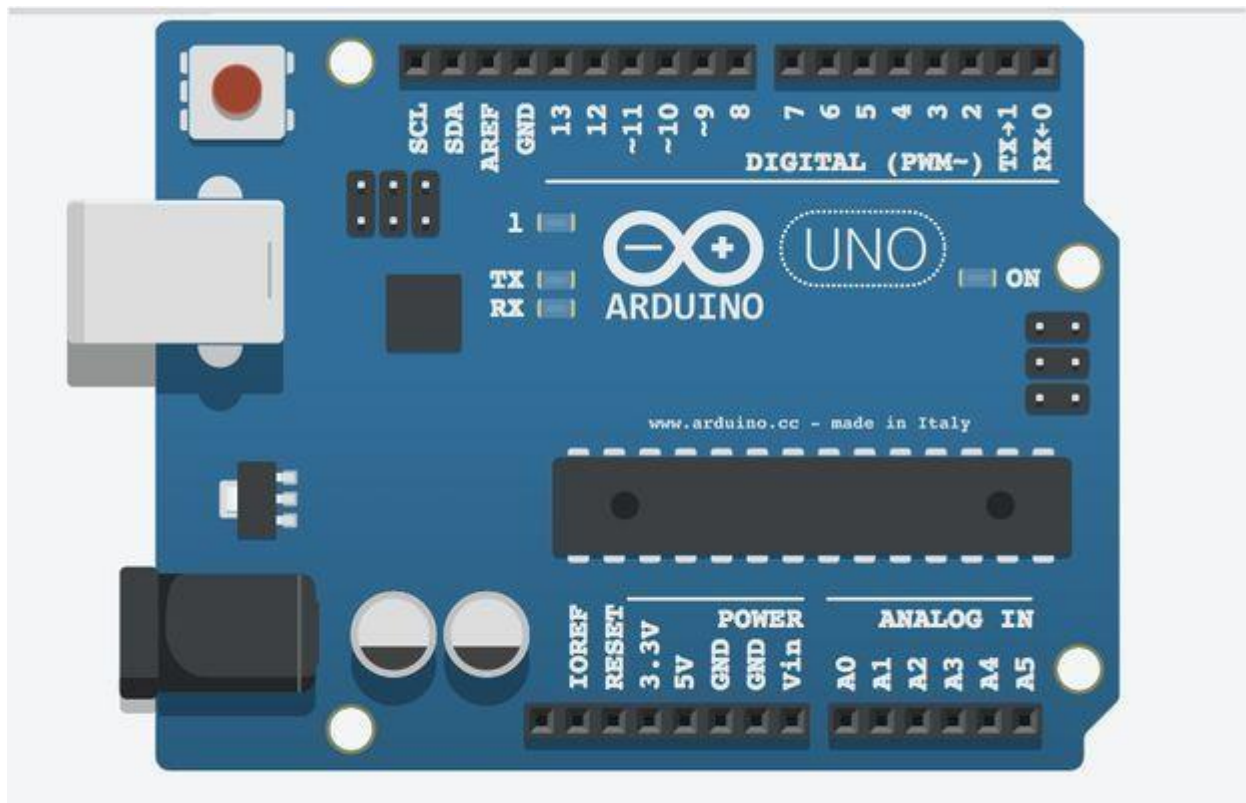
Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

IoT Platforms Overview: Arduino, Raspberry Pi

The IoT concepts imply a creation of network of various devices interacting with each other and with their environment. Interoperability and connectivity wouldn't be possible without hardware platforms that help developers solve issues such as building autonomous interactive objects or completing common infrastructure related tasks.

Let's go through the most popular IoT platforms and see how they work and benefit IoT software developers.

Arduino



The Arduino platform was created back in 2005 by the Arduino company and allows for open source prototyping and flexible software development and back-end deployment while providing significant ease of use to developers, even those with very little experience building IoT solutions.

Arduino is sensible to literally every environment by receiving source data from different external sensors and is capable to interact with other control elements over various devices, engines and drives. Arduino has a built-in micro controller that operates on the Arduino software.

Projects based on this platform can be both standalone and collaborative, i.e. realized with use of external tools and plugins. The integrated development environment (IDE) is composed of the open source code and works equally good with Mac, Linux and Windows OS. Based on a *processing* programming language, the Arduino platform seems to be created for new users and for experiments. The processing language is dedicated to visualizing and building interactive apps using animation and Java Virtual Machine (JVM) platform.

Let's note that this programming language was developed for the purpose of learning basic computer programming in a visual context. It is an absolutely free project available to every interested person. Normally, all the apps are programmed in C/C++, and are wrapped with *avr-gcc* (WinAVR in OS Windows).

Arduino offers analogue-to-digital input with a possibility of connecting light, temperature or sound sensor modules. Such sensors as *SPI* or *I2C* may also be used to cover up to 99% of these apps' market.

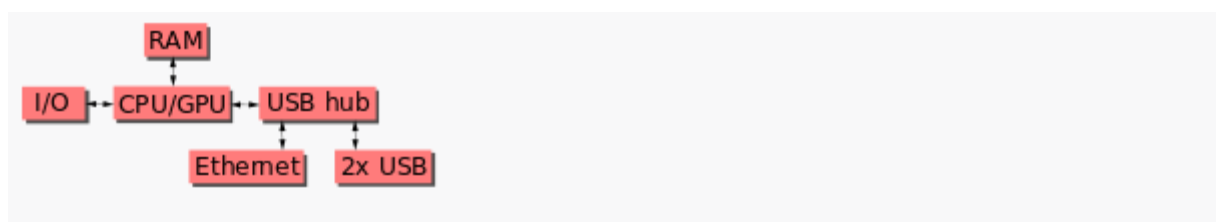
Arduino is a microcontroller (generally it is the 8-bit ATmega microcontroller), but not a mini-computer, which makes Arduino somehow limited in its features for advanced users. Arduino provides an excellent interactivity with external devices and offers a wide range of user manuals, project samples as well as a large community of users to learn from / share knowledge with.

Raspberry Pi

Raspberry Pi (/paɪ/) is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation in association with Broadcom. Early on, the Raspberry Pi project leaned towards the promotion of teaching basic computer science in schools and in developing countries. Later, the original model became far more popular than anticipated, selling outside its target market for uses such as robotics. It is now widely used in many areas, such as for weather monitoring, because of its low cost, modularity, and open design.

After the release of the second board type, the Raspberry Pi Foundation set up a new entity, named Raspberry Pi Trading, and installed Eben Upton as CEO, with the responsibility of developing technology. The Foundation was rededicated as an educational charity for promoting the teaching of basic computer science in schools and developing countries. The Raspberry Pi is one of the best-selling British computers.

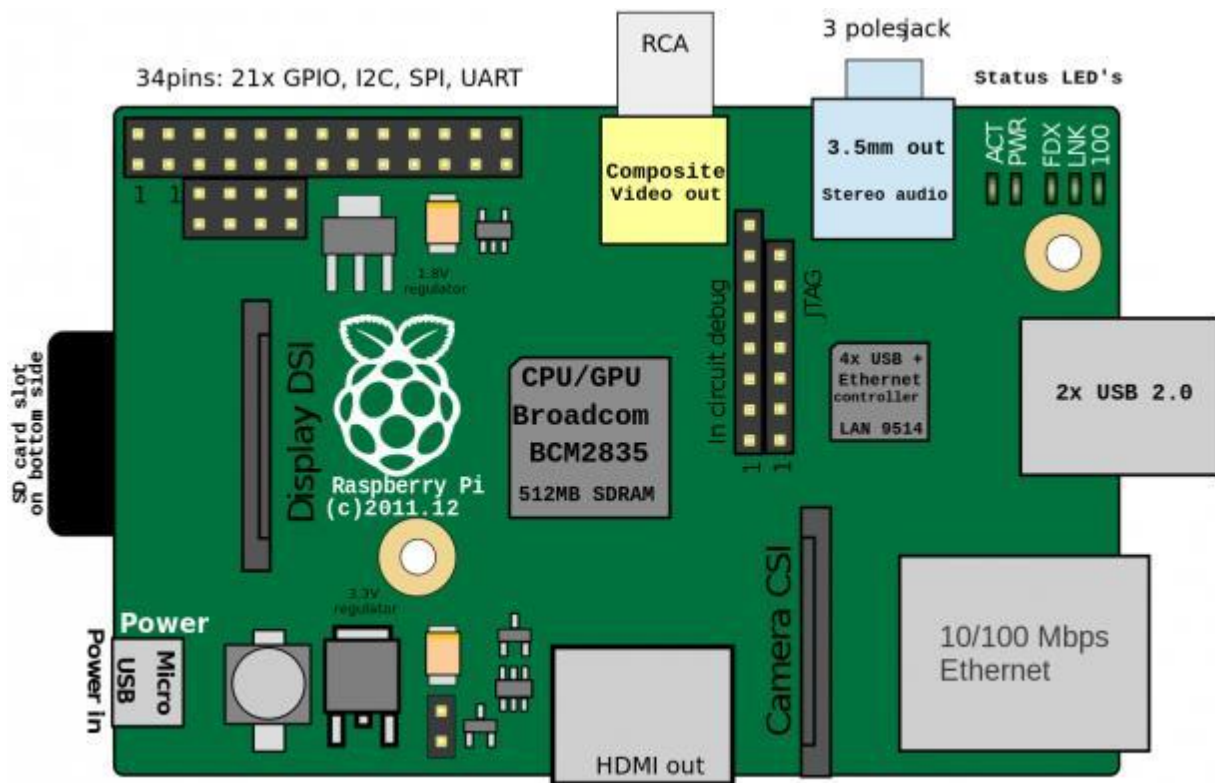
The Raspberry Pi hardware has evolved through several versions that feature variations in the type of the central processing unit, amount of memory capacity, networking support, and peripheral-device support.



This block diagram describes Model B and B+; Model A, A+, and the Pi Zero are similar, but lack the Ethernet and USB hub components. The Ethernet adapter is internally connected to an additional USB port. In Model A, A+, and the Pi Zero, the USB port is connected directly to the system on a chip (SoC). On the Pi 1 Model B+ and later models the USB/Ethernet chip contains a five-port USB hub, of which four ports are available, while the Pi 1 Model B only provides two. On the Pi Zero, the USB port is also connected directly to the SoC, but it uses a micro USB (OTG) port. Unlike all other Pi models, the 40 pin GPIO connector is omitted on the Pi Zero, with solderable through-holes only in the pin locations. The Pi Zero WH remedies this.

Processor speed ranges from 700 MHz to 1.4 GHz for the Pi 3 Model B+ or 1.5 GHz for the Pi 4; on-board memory ranges from 256 MiB to 1 GiB random-access memory (RAM), with up to 8 GiB available on the Pi 4. Secure Digital (SD) cards in MicroSDHC form factor (SDHC on early models) are used to store the operating system and program memory. The boards

have one to five USB ports. For video output, HDMI and composite video are supported, with a standard 3.5 mm tip-ring-sleeve jack for audio output. Lower-level output is provided by a number of GPIO pins, which support common protocols like I²C. The B-models have an 8P8C Ethernet port and the Pi 3, Pi 4 and Pi Zero W have on-board Wi-Fi 802.11n and Bluetooth.



Raspberry Pi is a mono-board computing platform that's as tiny as a credit card. Initially it was developed for computer science education with later on progress to wider functions.

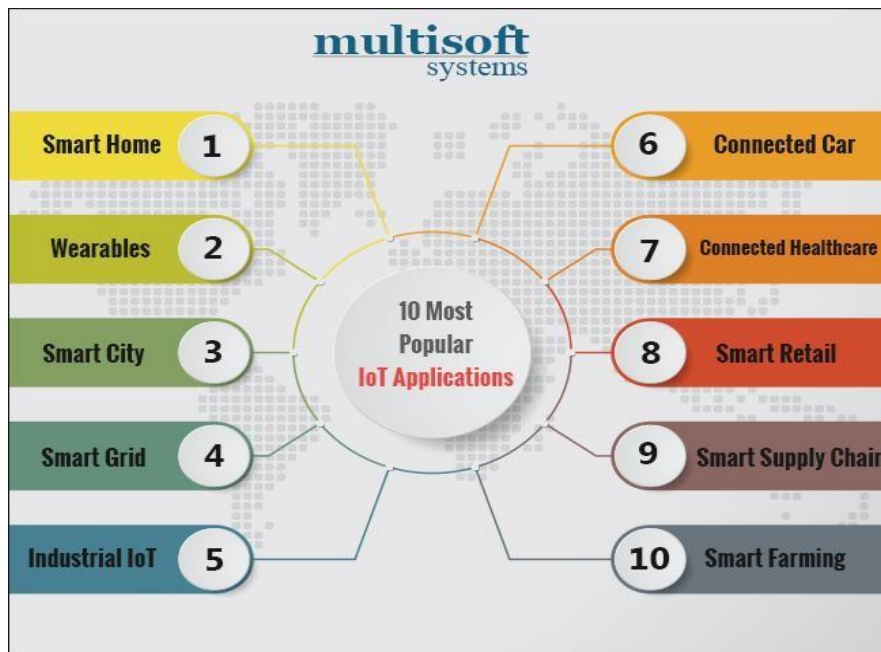
Since the inception of Raspberry, the company sold out more than 8 million items. Raspberry Pi 3 is the latest version and it is the first 64-bit computing board that also comes with built-in Wi-Fi and Bluetooth functions. According to Raspberry Pi Foundation CEO Eben Upton, *"it's been a year in the making"*. The Pi3 version is replaced with a quad-core 64-bit 1.2 GHz ARM Cortex A53 chip, 1GB of RAM, VideoCore IV graphics, Bluetooth 4.1 and 802.11n Wi-Fi. The developers claim the new architecture delivers an average 50% performance improvement over the Pi 2.

Another peculiarity of Raspberry Pi is the *GPIO* (General Purpose Input-Output), which is a low-level interface of self-operated control by input-output ports. Raspberry has it as a 40-pin connector.

Raspberry Pi uses Linux as its default operating system (OS). It's also fully Android compatible. Using the system on Windows OS is enabled through any virtualization system like *XenDesktop*. If you want to develop an application for Raspberry Pi on your computer, it

Unit 5-Case Study & IoT Applications

- IoT applications in home, infrastructures, buildings, security, Industries, Home appliances, other IoT electronic equipments.
- Industry 4.0 concepts



Case Study:

Definition:

"A case study is a research strategy and an empirical inquiry that investigates a phenomenon within its real-life context. Case studies are based on an in-depth investigation of a single individual, group or event to explore the causes of underlying principles".

One of the most promising IoT use cases is creating smarter, more efficient cities. Public energy grids can be optimized to balance workloads, predict energy surges, and distribute energy more equitably to customers. Traffic lights could be synced using IoT to adapt to traffic conditions in real-time.

IoT is the next step in the evolution of the internet and is being used in about everything you can think of.



IoT Applications:

IoT applications promise to bring immense value into our lives. With newer wireless networks, superior sensors and revolutionary computing capabilities, the **Internet of Things** could be the next frontier in the race for its share of the wallet. IoT applications are expected to equip billions of everyday objects with connectivity and intelligence. It is already being deployed extensively, few applications of IoT:

- Wearables
- Smart Home Applications
- Smart Buildings
- Smart Infrastructure
- Securities
- Health Care
- Smart Cities
- Agriculture
- Industrial Automation



IoT Applications: Smart Home, Smart Buildings and Infrastructure

IoT home automation is the ability to control domestic appliances by electronically controlled, internet-connected systems. It may include setting complex heating and lighting systems in advance and setting alarms and home security controls, all connected by a central hub and remote-controlled by a mobile app.

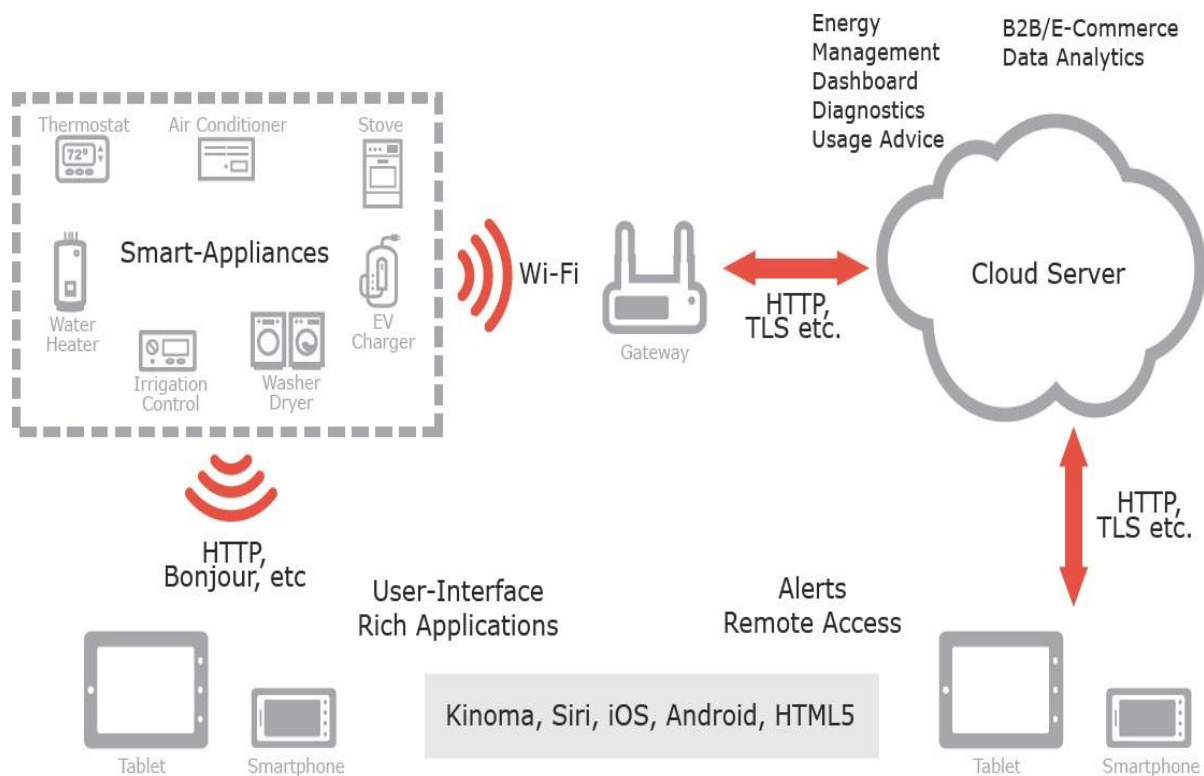


Figure .Smarthomeplatform.

The rise of Wi-Fi's role in home automation has primarily come about due to the networked nature of deployed electronics where electronic devices (TVs and AV receivers, mobile devices, etc.) have started becoming part of the home IP network and due to the increasing rate of adoption of mobile computing devices (smartphones, tablets, etc.), see above Figure.

The networking aspects are bringing online streaming services or network playback, while becoming a means to control the device functionality over the network. At the same time, mobile devices ensure that consumers have access to a portable 'controller' for the electronics connected to the network. Both types of devices can be used as gateways for IoT applications. In this context, many companies are considering building platforms that

integrate the building automation with entertainment, healthcare monitoring, energy monitoring and wireless sensor monitoring in the home and building environments.

IoT applications using sensors to collect information about the operating conditions combined with cloud hosted analytics software that analyzes disparate data points will help facility managers become far more proactive about managing buildings at peak efficiency.

Issues of building ownership (i.e., building owner, manager, or occupants) challenge integration with questions such as who pays initial system cost and who collects the benefits over time. A lack of collaboration between the subsectors of the building industry slows new technology adoption and can prevent new buildings from achieving energy, economic and environmental performance targets.

Integration of cyber physical systems both within the building and with external entities, such as the electrical grid, will require stakeholder cooperation to achieve true interoperability. As in all sectors, maintaining security will be a critical challenge to overcome.

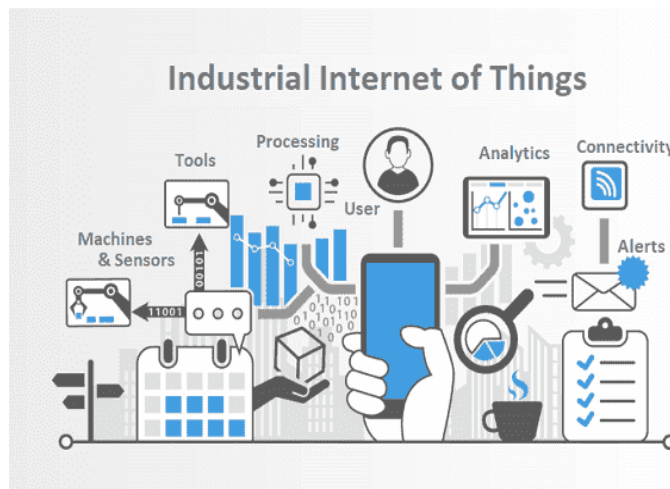
Within this field of research the exploitation of the potential of wireless sensor networks (WSNs) to facilitate intelligent energy management in buildings, which increases occupant comfort while reducing energy demand, is highly relevant.

In addition to the obvious economic and environmental gains from the introduction of such intelligent energy management in buildings other positive effects will be achieved. Not least of which is the simplification of building control; as placing monitoring, information feedback equipment and control capabilities in a single location will make a buildings' energy management system easier to handle for the building owners, building managers, maintenance crews and other users of the building. Using the Internet together with energy management systems also offers an opportunity to access a buildings' energy information and control systems from a laptop or a Smartphone placed anywhere in the world. This has a huge potential for providing the managers, owners and inhabitants of buildings with energy consumption feedback and the ability to act on that information.

In the context of the future Internet of Things, Intelligent Building Management Systems can be considered part of a much larger information system. This system is used by facilities managers in buildings to manage energy use and energy procurement and to maintain buildings systems. It is based on the infrastructure of the existing Intranets and the Internet, and therefore utilizes the same standards as other IT devices. Within this context reductions in the cost and reliability of WSNs are transforming building automation, by making the maintenance of energy efficient healthy, productive work spaces in buildings increasingly cost effective.

IoT Application in industries:

IoT in industry is a rapidly developing area. Numerous IoT research and application projects have been done by universities or in joint industry- university consortia in recent years.



Internet of things (IoT) has become part of your daily life. The “things connected to the internet” idea is continuously evolving in content, areas of applications, visions and technology. New real life and industrial projects have been done and joint future oriented industry and government initiatives such as Industry 4.0 in Germany, have been started [1]. Since Industrial production is one of the world’s biggest economic factors one of the major objectives of these initiatives is to bring the paradigms of the IoT to the factories enabling them to cope with the challenges raised by popular megatrends.

The foremost megatrends relevant for factories are globalization, progressing technological evolution, the dynamization of product life cycles, the aging work force and the shortage of resources. Central effects are the acceleration of innovation cycles and the increasing customer demand for individualized mass produces with highest quality expectations. Within the context of industrial production IoT projects and applications are developing in manufacturing, supply chain,

supervision and servicing. A major question in all projects is about the value, the benefit such application can bring to the user, to the owner or to society.

The value question is extremely pertinent in the industry: in the manufacturing industry entire factory related processes, but also in industrial applications where it comes to ensure operation of industrial installations and provide supervision, and improved life service. It is the value which such applications bring which will determine their adoption, acceptance and wide use. However, this value is very difficult to quantify and prove, and it depends on multiple aspects which are strongly application area dependent.

IoT applications form the value creation for industry and brings together expert opinions from academia, research and industry. The industrial application of IoT is multi-faceted and each of the subsections in this paper will highlight an aspect related to industrial application, discuss or show a case or the evolution and potential of a specific technology from industry application point of view. The paper is having a holistic manner to industrial challenges and requirements. Also it will refer to factory concepts and applications supported by IoT, including processes and flows taking a view on related technologies and their evolution.

IoT applications benefit and value creation in an industrial environment may have its origin in different aspects, depending on the application type. There is no value but “values” each contributing to the total benefit such as:

- Value from visibility identification, location tracking
- Value from IoT-supported safety in hard industrial environments
- Value from right information providing or collecting
- Value from improved industrial operation and flows in industry
- Value from reduced production losses
- Value from reduced energy consumption
- Value from new type of processes made possible by IoT applications
- Value from new type of maintenance and lifetime approaches
- Value enabled by smart objects, connected aspects
- Value from sustainability.

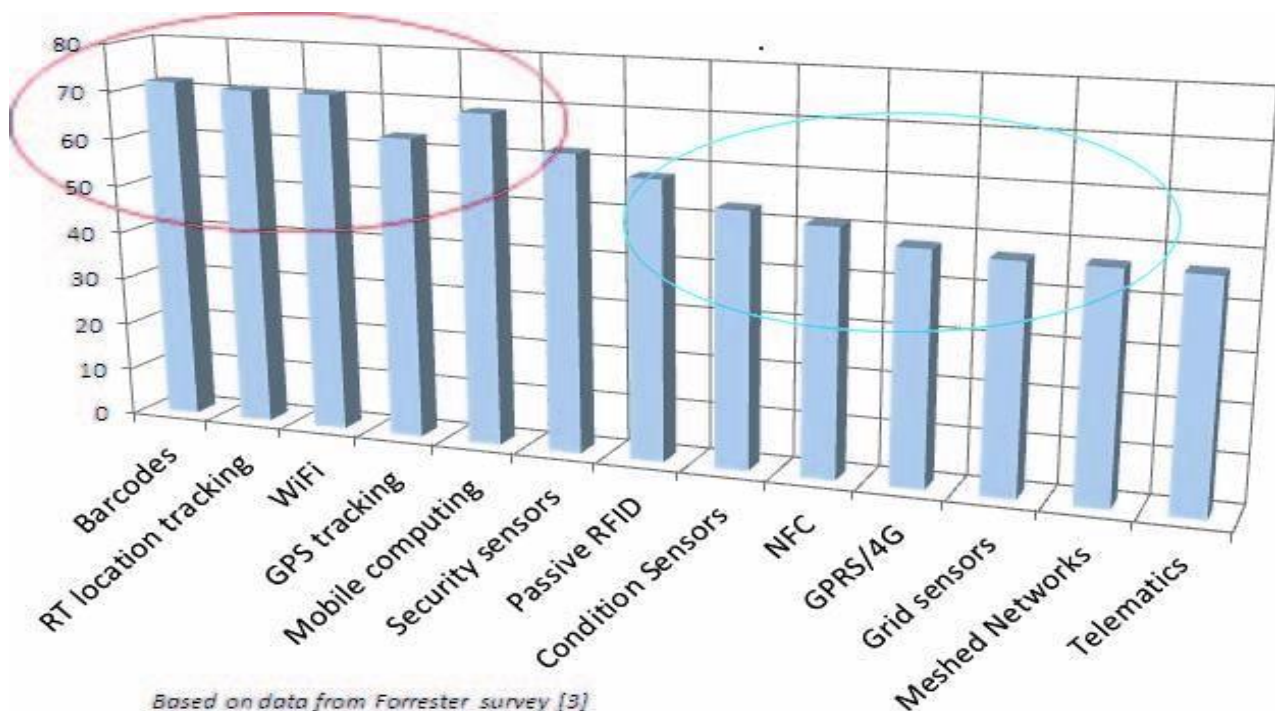


Fig. 5.2 View on very important and important perceived IoT technologies expected to bring value in applications.

The status and estimated potential of IoT applications is presented in Figure 5.3 considering three major areas: supply chain, future industry/future actory and over lifetime applications and activities such as logistics, manufacturing and service/maintenance. A strong potential and additional application is expected in industry operation and industry lifetime applications including lifetime service.

Areas	Supply chain	Industry	Lifetime
Activities	Logistics	Manufacturing	Service
IoT present Applications and Value	Many	Some	Few
IoT additional Applications Potential	Increase	Strong	Strong

Figure. Status and estimated potential of IoT applications.

IoT application requirements and capabilities:

The expectations toward IoT applications in industry are high. The capabilities they have to offer are depending strongly on the industrial area and the concrete application. For example the environment where IoT application may be used may range from clean room condition and normal ambient temperatures to heavy and dirty environment, locations with high temperatures, areas with explosion risk, areas with metallic surroundings, and corrosive environment on sea or underground.

A list of a set of industry related capabilities and requirements is presented below, without claiming completeness. The list items are related to the IoT hardware, software and to serviceability and management aspects. Comments have been added to all items to make the requirement more specific. The IoT application capabilities for industrial application should meet requirements such as:

IoT Application of sec

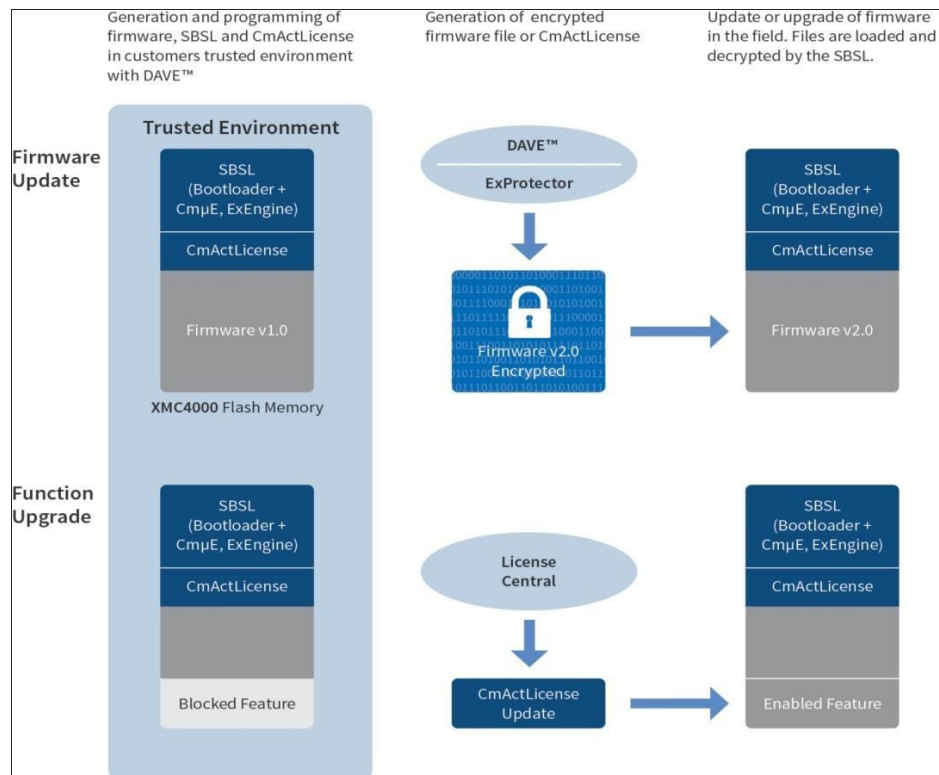
Security:

- IoT devices are connected to your desktop or laptop. Lack of security increases the risk of your personal information leaking while the data is collected and transmitted to the IoT device.
- IoT devices are connected with a consumer network. This network is also connected with other systems. So if the IoT device contains any security vulnerabilities, it can be harmful to the consumer's network. This vulnerability can attack other systems and damage them.
- Sometimes unauthorized people might exploit the security vulnerabilities to create risks to physical safety.

Privacy Risks:

- In IoT, devices are interconnected with various hardware and software, so there are obvious chances of sensitive information leaking through unauthorized manipulation.
- All the devices are transmitting the user's personal information such as name, address, date of birth, health card information, credit card detail and much more without encryption.

Though there are security and privacy concerns with IoT, it adds values to our lives by allowing us to manage our daily routine tasks remotely and automatically, and more importantly, it is a game-changer for industries.



IoT Application of home appliances:

Internet of Things is a technology that can connect to the internet without the influence of people and send information collected to users through this internet network to which they are connected. Devices in this dynamic are very common today. Many homes, companies and even public organizations benefit from this technology. Used in smart home IoT home appliances is also one of them.



A house must have smart devices to be smart. These smart devices are the building blocks of today's technology. So why are these devices and apps smart? First, these devices have their own Internet. With this internet tool, users can receive information from the device. With this internet connection, you can get a lot of information from your smart device.

This information which receives from smart devices makes safety for your living area.

Smart devices work with technological devices while making you and your home a more secure space. The biggest hero of these technological devices is microprocessors. microprocessors act as the brain for your smart device. There are sensors that allow your smart devices to be classified according to their characteristics and detect the danger or differences in your home.

There are many sensors classified by type. Motion sensors, light sensors, image detection, and processing sensors are one of them. For example, if the position of your belongings changes without your knowledge, there are motion sensors that can detect this position change. The motion sensor detects the position change and sends you information about this.

Home Appliance in Internet of Things:

Smart home systems are integrated and enable you to play an active role in every part of your home by surrounding your home. When you're not at home, but your mind stays at home, it's behind you. With smart home systems, you can intervene in your home as if you are at home and perform the necessary controls. In addition to these protection systems, smart home appliances have been making human life easier since the day it was developed.

Smart Washing Machine:



It is very important to save time in daily life. we live in a period where we have to keep up. that's where technology comes in. You can access the developed smart washing machine on your smartphone. you can monitor and control the process at the same time. This smart washing machine can also dry your laundry with the control application.

Smart Refrigerator with Internet of Things:

Internet in this kitchen which makes life easier for you and your family in the kitchen. With this internet connection, you can transmit a lot of information to your shopping list in the weather. You can also view the inside of your refrigerator with its camera technology.

Shortest Way to Dry Hair:

This time it has infrared technology. With this technology, the device is created wirelessly. Wireless shape so you can dry your hair without connecting the machine

Smart Doorbell:

The most important thing in smart home applications is known to be secure and protected home. With this smart doorbell designed for security, you can recognize people who come to your home with high quality. The night also has infrared technology added to the smart bell. This will also send the screen to you when it gets dark.



Smart Camera for Safe Home:

Control of your home is in your hands from every part. This smart camera sends records from every part of your home to your smartphone with the Internet of Things technology. Research on smart camera technology will continue for those who want a safe life.



Industry 4.0 concepts

Industry 4.0 refers to a new phase in the Industrial Revolution that focuses heavily on interconnectivity, automation, machine learning, and real-time data. Industry 4.0, also sometimes referred to as IIoT(Industrial Internet of Things) or smart manufacturing which provides physical production and operations with smart digital technology, machine learning and big data to create a more holistic and better connected ecosystem for companies that focus on manufacturing and supply chain management.

While every company and organization operating today is different, they all face a common challenge—the need for connectedness and access to real-time insights across processes, partners, products, and people. That's where Industry 4.0 comes into play. Industry 4.0 is not just about investing in new technology and tools to improve manufacturing efficiency but it's about revolutionizing the way the entire business operates and grows.

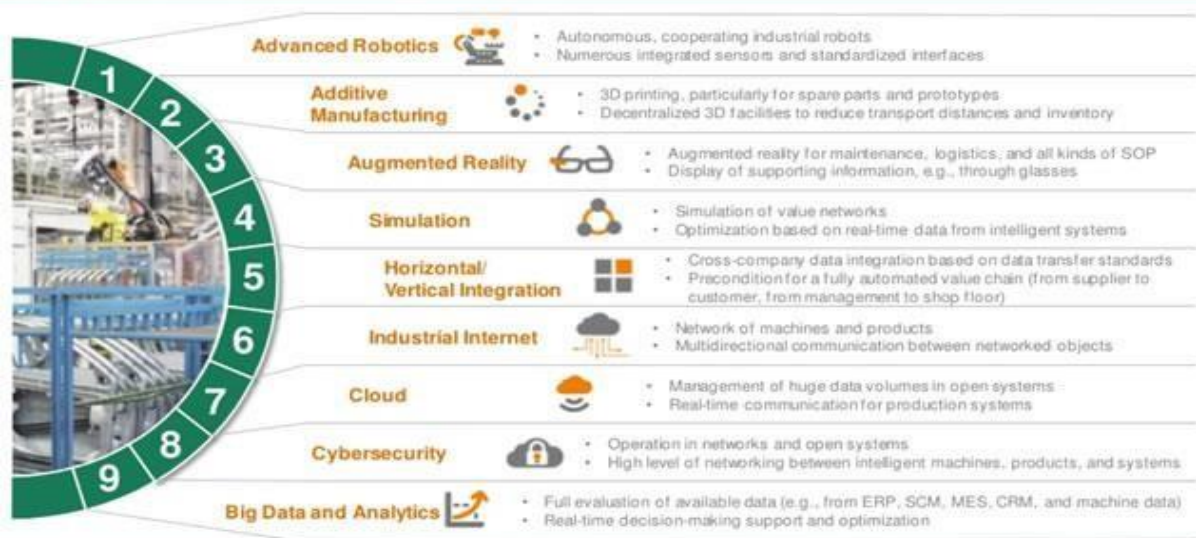
Industry 4.0 refers to the use of automation and data exchange in manufacturing. According to the Boston Consulting Group there are nine principal technologies that make up Industry 4.0: Autonomous Robots, Simulation, Horizontal and Vertical System Integration, the Industrial Internet of Things, Cybersecurity, The Cloud, Additive Manufacturing, Data and Analytics, and Augmented Reality. These technologies are used to create a “smart factory” where machines, systems, and humans communicate with each other in order to coordinate and monitor progress along the assembly line. Networked devices provide sensor data and are digitally controlled. The net effect is the ability to rapidly design, modify, create, and customize things in the real world, while lowering costs and reacting to changes in consumer preferences, demand, the supply chain and technology.

The goal is to enable autonomous decision-making processes, monitor assets and processes in real-time, and enable equally real-time connected value creation networks through early involvement of stakeholders, and vertical and horizontal integration.



Figure: Nine Technologies of Industry 4.0

Industry 4.0 refers to the convergence and application of nine digital industrial technologies



Many application examples already exist for all nine technologies

Today some companies have invested in a few of these technologies; predominantly the traditional pillars of the third platform such as cloud and Big Data / Analytics and increasingly in the Industrial Internet of Things from an integrated perspective and thus overlapping with several of these “technologies” or maybe better: sets of technologies and connected benefits.

Evolution of Industry 4.0

There are four distinct industrial revolutions that the world either has experienced or continues to experience today.

1. The First Industrial Revolution

The first industrial revolution happened between the late 1700s and early 1800s. During this period of time, manufacturing evolved from focusing on manual labor performed by people and aided by work animals to a more optimized form of labor performed by people through the use of water and steam-powered engines and other types of machine tools.

2. The Second Industrial Revolution

In the early part of the 20th century, the world entered a second industrial revolution with the introduction of steel and use of electricity in factories. The introduction of electricity enabled manufacturers to increase efficiency and helped make factory machinery more mobile. It was during this phase that mass production concepts like the assembly line were introduced as a way to boost productivity.

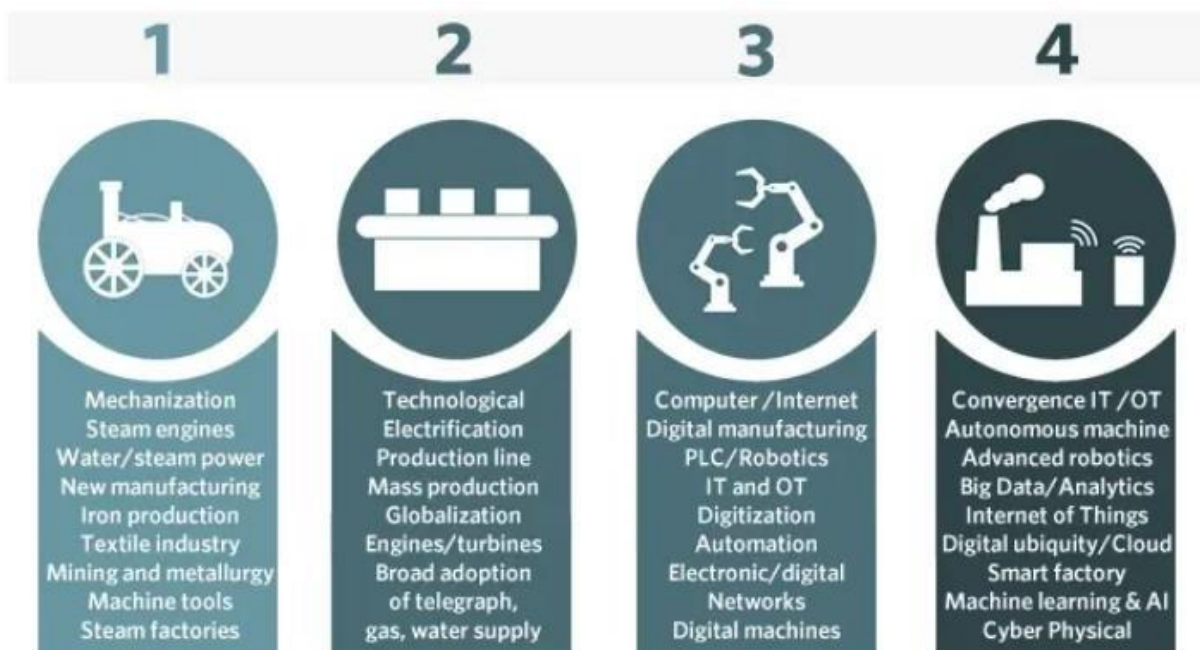
3. Third Industrial Revolution

Starting in the late 1950s, a third industrial revolution slowly began to emerge, as manufacturers began incorporating more electronic and eventually computer technology

into their factories. During this period, manufacturers began experiencing a shift that put less emphasis on analog and mechanical technology and more on digital technology and automation software.

4. Fourth Industrial Revolution[Industry 4.0]

Fourth industrial revolution has emerged known as Industry 4.0. Industry 4.0 takes the emphasis on digital technology from recent decades to a whole new level with the help of interconnectivity through the Internet of Things (IoT), access to real-time data, and the introduction of cyber-physical systems. Industry 4.0 offers a more comprehensive, interlinked and holistic approach to manufacturing. It connects physical with digital, and allows for better collaboration and access across departments, partners, vendors, product, and people. An industry 4.0 empowers business owners to control and understand every aspect of their operation, and allows them to leverage instant data to boost productivity, improve processes, and drive growth.



Industry 4.0 is often used interchangeably with the notion of the fourth industrial revolution. It is characterized among others by

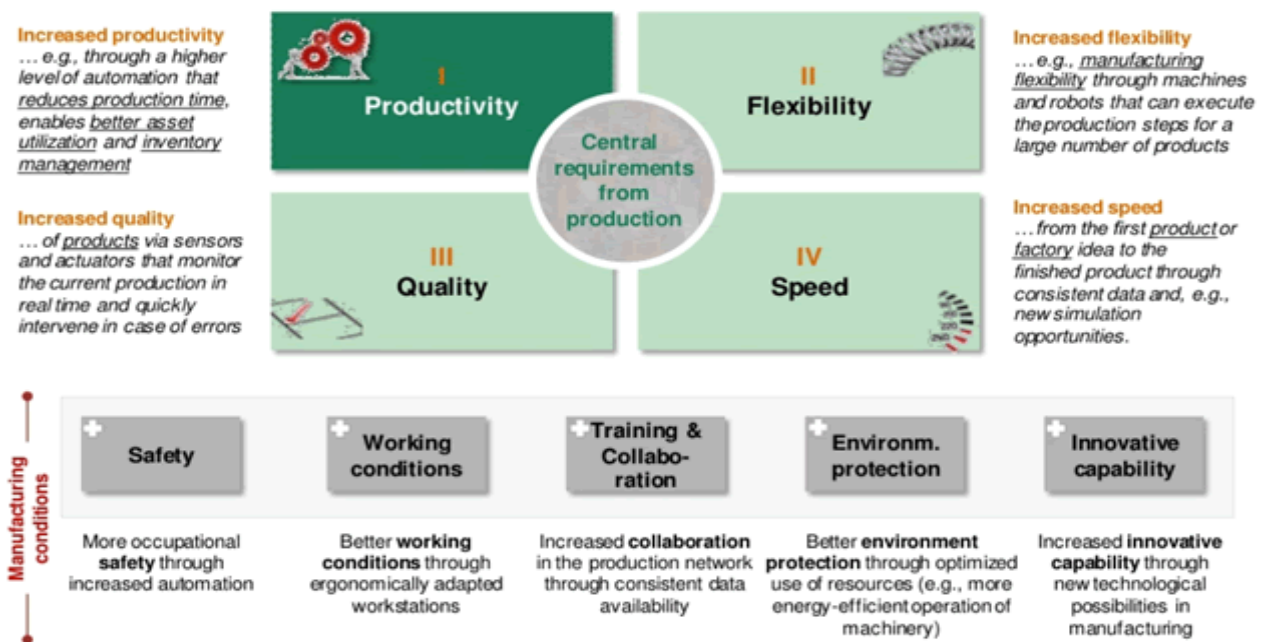
- 1) even more automation than in the third industrial revolution
- 2) the bridging of the physical and digital world through cyber-physical systems, enabled by Industrial IoT
- 3) a shift from a central industrial control system to one where smart products define the production steps
- 4) closed-loop data models and control systems and
- 5) personalization/customization of products.

Benefits of Industry 4.0

Industry 4.0 spans the entire product life cycle and supply chain, design, sales, inventory, scheduling, quality, engineering, and customer and field service. Everyone shares informed, up-to-date, relevant views of production and business processes and much richer and more timely analytics.

The essential goal of Industry 4.0 is to make manufacturing and related industries such as logistics faster, more efficient and more customer-centric, while at the same time going beyond automation and optimization and detect new business opportunities and models.

In fact, Industry 4.0 offers *multiple* benefits—enhanced productivity is just the beginning



Most of the benefits of Industry 4.0 are obviously similar to the benefits of the digital transformation of manufacturing, the usage of the IoT in manufacturing, operational and business process optimization, information-powered ecosystems of value, digital transformation overall, the Industrial Internet and many other topics on our website. Few of the key benefits of Industry 4.0 are.

1. Enhanced productivity through optimization and automation
2. Real-time data for a real-time supply chain in a real-time economy
3. Higher business continuity through advanced maintenance and monitoring possibilities
4. Better quality products: real-time monitoring, IoT-enabled quality improvement and cobots
5. Better working conditions and sustainability
6. Personalization and customization for the 'new' consumer