Subject Notes
BTCS-602 -(IOT) Internet of Things

Unit -1

Definition-The Internet of Things, also called The Internet of Objects, refers to a wireless network between objects. By embedding short-range mobile transceivers into a wide array of additional gadgets and everyday items, enabling new forms of communication between people and things, and between things themselves. The term Internet of Things has come to describe a number of technologies and research disciplines that enable the Internet to reach out into the real world of physical objects. Things having identities and virtual personalities operating in smart spaces using intelligent interfaces to connect and communicate within social, environmental, and user contexts. From any time, any place connectivity for anyone, we will now have connectivity for anything.

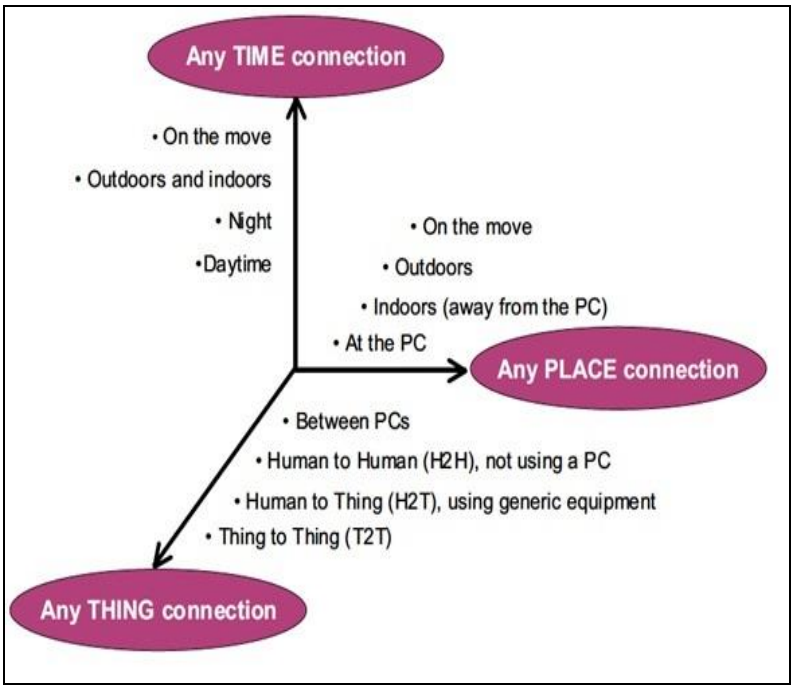


Fig. 1.1 Internet of Things Accessibility

Characteristics of IOT-The no-need-to-know in terms of the underlying details of infrastructure, applications interface with the infrastructure via the APIs.The flexibility and elasticity allows these systems to scale up and down at will utilizing the resources of all kinds CPU, storage, server capacity, load balancing, and databases The “pay as much as used and needed” type of utility computing and the “always on!, anywhere and any place” type of network-based computing. The fundamental characteristics of the IoT are as follows:

- **Interconnectivity:** With regard to the IoT, anything can be interconnected with the global information and communication infrastructure.
- **Things-related services:** The IoT is capable of providing thing-related services within the constraints of things, such as privacy protection and semantic consistency between physical things and their associated virtual things. In order to provide thing-related services within the constraints of things, both the technologies in physical world and information world will change.
- **Heterogeneity:** The devices in the IoT are heterogeneous as based on different hardware platforms and networks. They can interact with other devices or service platforms through different networks.
- **Dynamic changes:** The state of devices change dynamically, e.g., sleeping and waking up, connected and/or disconnected as well as the context of devices including location and speed. Moreover, the number of devices can change dynamically.
- **Enormous scale:** The number of devices that need to be managed and that communicate with each other will be at least an order of magnitude larger than the devices connected to the current Internet. Even more critical will be the management of the data generated and their interpretation for application purposes. This relates to semantics of data, as well as efficient data handling.

- **Safety:** As we gain benefits from the IoT, we must not forget about safety. As both the creators and recipients of the IoT, we must design for safety. This includes the safety of our personal data and the safety of our physical well-being. Securing the endpoints, the networks, and the data moving across all of it means creating a security paradigm that will scale.
- **Connectivity:** Connectivity enables network accessibility and compatibility. Accessibility is getting on a network while compatibility provides the common ability to consume and produce data.

IOT architectural Overview-IOT architecture consists of different layers of technologies supporting IOT. It serves to illustrate how various technologies relate to each other and to communicate the scalability, modularity and configuration of IOT deployments in different scenarios. The functionality of each layer is described below:

- **Smart device / sensor layer:** The lowest layer is made up of smart objects integrated with sensors. The sensors enable the interconnection of the physical and digital worlds allowing real-time information to be collected and processed. There are various types of sensors for different purposes. The sensors have the capacity to take measurements such as temperature, air quality, speed, humidity, pressure, flow, movement and electricity etc. In some cases, they may also have a degree of memory, enabling them to record a certain number of measurements. A sensor can measure the physical property and convert it into signal that can be understood by an instrument. Sensors are grouped according to their unique purpose such as environmental sensors, body sensors, home appliance sensors and vehicle telemetric sensors, etc
- **Gateways and Networks-**Massive volume of data will be produced by these tiny sensors and this requires a robust and high performance wired or wireless network infrastructure as a transport medium. Current networks, often tied with very different protocols, have been used to support machine-to-machine (M2M) networks and their applications. With demand needed to serve a wider range of IOT services and applications such as high speed transactional services, context-aware applications, etc, multiple networks with various technologies and access protocols are needed to work with each other in a heterogeneous configuration. These networks can be in the form of a private, public or hybrid models and are built to support the communication requirements for latency, bandwidth or security. Various gateways (microcontroller, microprocessor) & gateway networks (WI-FI, GSM, GPRS).
- **Management Service Layer-**The management service renders the processing of information possible through analytics, security controls, process modeling and management of devices. One of the important features of the management service layer is the business and process rule engines. IOT brings connection and interaction of objects and systems together providing information in the form of events or contextual data such as temperature of goods, current location and traffic data. Some of these events require filtering or routing to post-processing systems such as capturing of periodic sensory data, while others require response to the immediate situations such as reacting to emergencies on patient's health conditions. The rule engines support the formulation of decision logics and trigger interactive and automated processes to enable a more responsive IOT system.
- **Application Layer-**The IoT application covers "smart" environments/spaces in domains such as: Transportation, Building, City, Lifestyle, Retail, Agriculture, Factory, Supply chain, Emergency, Healthcare, User interaction, Culture and tourism, Environment and Energy.

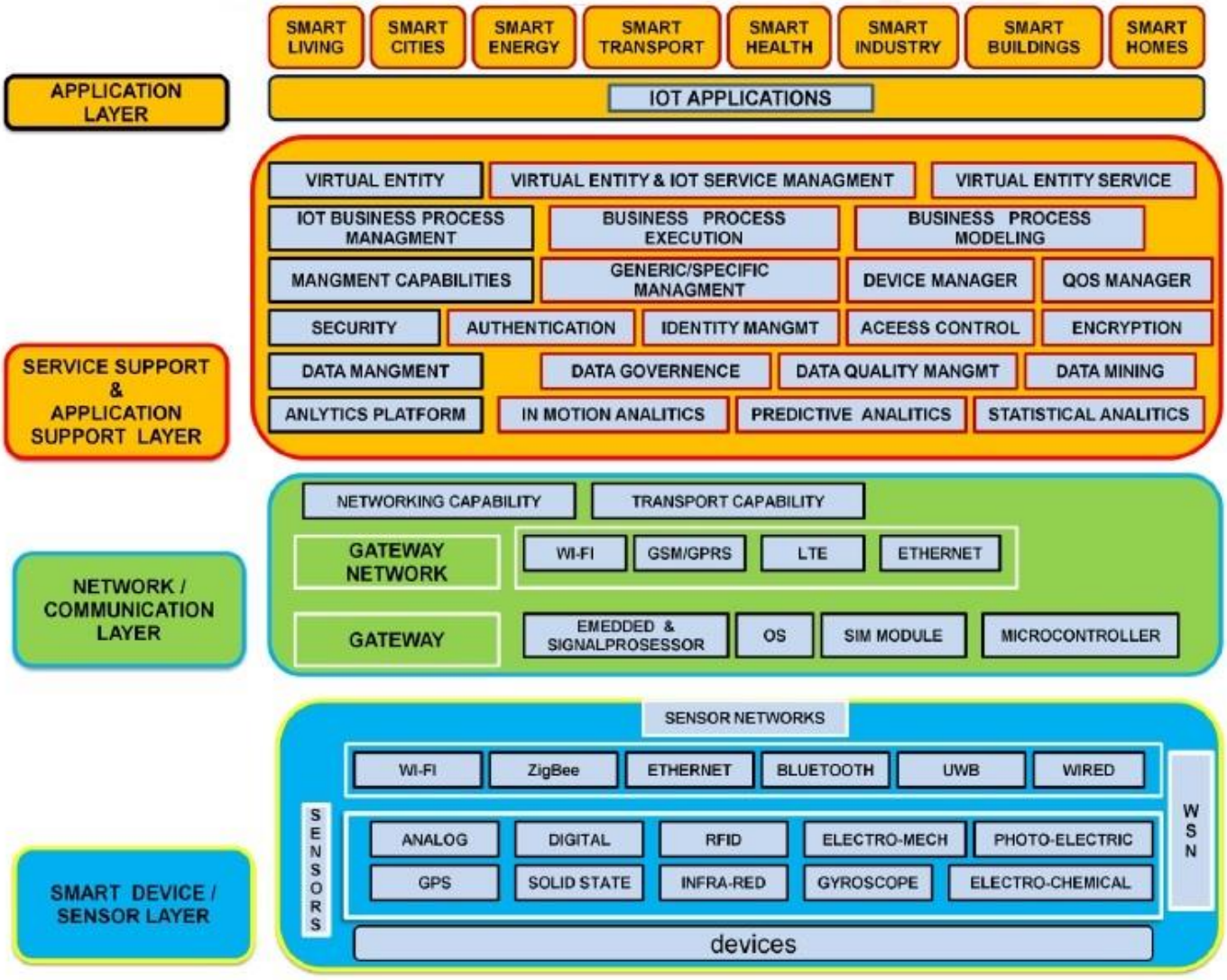


Fig. 1.2 IOT Architecture

IOT conceptual view

The main tasks of this framework are to analyze and determine the smart activities of these intelligent devices through maintaining a dynamic interconnection among those devices. The proposed framework will help to standardize IoT infrastructure so that it can receive e-services based on context information leaving the current infrastructure unchanged. The active collaboration of these heterogeneous devices and protocols can lead to future ambient computing where the maximum utilization of cloud computing will be ensured. This model is capable of logical division of physical devices placement, creation of virtual links among different domains, networks and collaborate among multiple application without any central coordination system. IaaS can afford standard functionalities to accommodate and provides access to cloud infrastructure. The service is generally offered by modern data centers maintained by giant companies and organization. It is categorized as virtualization of resources which permits a user to install and run application over virtualization layer and allows the system to be distributed, configurable and scalable.

Total infrastructure system can be categorized into 4 layers to receive context supported e-services out of raw data from the Internet of Things. These 4 layers establish a generic framework that does not alter the current network infrastructure but create an interfacing among services and entities through network virtualization.

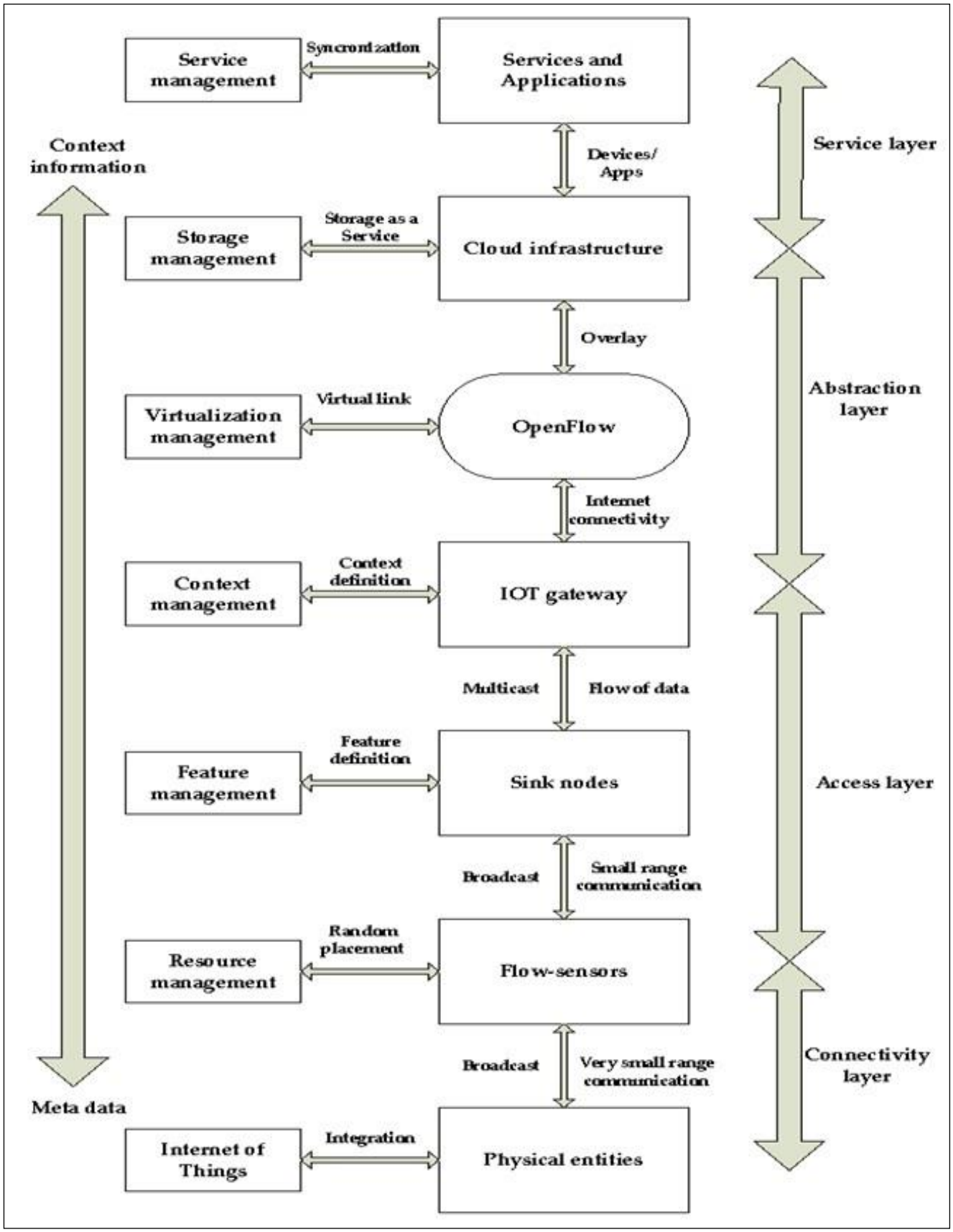


Fig. 1.3 IOT Conceptual View

1. Connectivity Layer

This layer includes all the physical devices involved in the framework and the interconnection among them. Future internet largely depends on the unification of these common objects found everywhere near us and these should be distinctly identifiable and controllable.

This layer also involves assigning of low range networking devices like sensors, actuators, RFID tags etc and resource management checks the availability of physical resources of all the devices and networks involved in the underlying infrastructure. These devices contain very limited resources and resource management ensures the maximum utilization with little overhead. It also allows sharing and distribution of information among multiple networks or single network divided into multiple domains.

2. Access Layer

Context Data will be reached to internet via IoT Gateway as captured by short range devices in form of raw data. Access layer comprises topology definition, network initiation, creation of domains etc. This layer also includes connection setup, intra-inter domain communication, scheduling, packet transmissions between flow-sensors and IoT gateway. The simulation was run later in this paper for different scenario based on this layer. Feature management contains a feature filter which accepts only acceptable context data and redundant data are rejected. Large number of sensor maintains lots of features but only a small subset of features is useful generate a context data.

Feature filter helps to reduce irrelevant data transmission, increases the data transfer rate of useful data and reduce energy and CPU consumption too. Number of features can be different based on the application requirements and context data types.

3. Abstraction Layer

One of the most important characteristics of OpenFlow is to add virtual layers with the preset layers, leaving the established infrastructure unchanged. A virtual link can be created among different networks and a common platform can be developed for various communication systems. The system is fully a centralized system from physical layer viewpoint but a distribution of service (flow visor could be utilized) could be maintained. One central system can monitor, control all sorts of traffics. It can help to achieve better band-width, reliability, robust routing, etc. which will lead to a better Quality of Services (QoS).

In a multi-hopping scenario packets are transferred via some adjacent nodes. So, nodes near to access points bears too much load in comparison to distant nodes in a downstream scenario and inactivity of these important nodes may cause the network to be collapsed. Virtual presence of sensor nodes can solve the problem where we can create a virtual link between two sensor networks through access point negotiation. So, we can design a three a three layer platform, where common platform and virtualization layer are newly added with established infrastructure. Sensors need not to be worried about reach-ability or their placement even in harsh areas. Packet could be sent to any nodes even if it is sited on different networks.

4. Service Layer

Storage management bears the idea about all sorts of unfamiliar and/or important technologies and information which can turn the system scalable and efficient. It is not only responsible for storing data but also to provide security along with it. It also allows accessing data effectively; integrating data to enhance service intelligence, analysis based on the services required and most importantly increases the storage efficiency. Storage and management layer involves data storage & system supervision, software services and business management & operations. Though they are included in one layer, the business support system resides slightly above of cloud computing service whereas Open-Flow is placed below of it as presented to include virtualizations and monitor management.

Service management combines the required services with organizational solutions and thus new generation user service becomes simplified. These forthcoming services are necessitated to be co interrelated and combined in order to meet the demand socio- economic factors such as environment analysis, safety measurement, climate management, agriculture modernization etc.

Iot Functional View-The Internet of Things concept refers to uniquely identifiable things with their virtual representations in an Internet-like structure and IoT solutions comprising a number of components such as (1) Module for interaction with local IoT devices. This module is responsible for acquisition of observations and their forwarding to remote servers for analysis and permanent storage.

(2) Module for local analysis and processing of observations acquired by IoT devices.

(3) Module for interaction with remote IoT devices, directly over the Internet. This module is responsible for acquisition of observations and their forwarding to remote servers for analysis and permanent storage. (4) Module for application specific data analysis and processing. This module is running on an application server serving all clients. It is taking requests from mobile and web clients and relevant IoT observations as input, executes appropriate data processing algorithms and generates output in terms of knowledge that is later presented to users. (5) User interface (web or mobile): visual representation of measurements in a given context (for example on a map) and interaction with the user, i.e. definition of user queries. The Designs are shown below:

Physical Design of IOT

The Internet of Things will become part of the fabric of everyday life. It will become part of our overall infrastructure just like water, electricity, telephone, TV and most recently the Internet. Whereas the current Internet typically connects full-scale computers, the Internet of Things (as part of the Future Internet) will connect everyday objects with a strong integration into the physical world.

1. Plug and Play Integration

If we look at IoT-related technology available today, there is a huge heterogeneity. It is typically deployed for very specific purposes and the configure requires significant technical knowledge and may be cumbersome. To achieve a true Internet of Things we need to move away from such small-scale, vertical application silos, towards a horizontal infrastructure on which a variety of applications can run simultaneously. This is only possible if connecting a thing to the Internet of Things becomes as simple as plugging it in and switching it on. Such plug and play functionality requires an infrastructure that supports it, starting from the networking level and going beyond it to the application level. This is closely related to the aspects discussed in the section on autonomy. On the networking level, the plug & play functionality has to enable the communication, features like the ones provided by IPv6 are in the directions to help in this process. Suitable infrastructure components have then to be discovered to enable the integration into the Internet of Things. This includes announcing the functionalities provided, such as what can be sensed or what can be actuated.

2. Infrastructure Functionality

The infrastructure needs to support applications in finding the things required. An application may run anywhere, including on the things themselves. Finding things is not limited to the start-up time of an application. Automatic adaptation is needed whenever relevant new things become available, things become unavailable or the status of things changes. The infrastructure has to support the monitoring of such changes and the adaptation that is required as a result
of the changes.

3. Semantic Modeling of Things

To reach the full potential of the Internet of Things, semantic information regarding the things, the information they can provide or the actuations they can perform need to be available. It is not sufficient to know that there is a temperature sensor or an electric motor, but it is important to know which temperature the sensor measures: the indoor temperature of a room or the temperature of the fridge, and that the electric motor can open or close the blinds or move something to a different location. As it may not be possible to provide such semantic information by simply switching on the thing, the infrastructure should make adding it easy for users. Also, it may be possible to derive semantic information, given some basic information and additional knowledge, e.g. deriving information about a room, based on the information that a certain sensor is located in the room. This should be enabled by the infrastructure.

4. Physical Location and Position

As the Internet of Things is strongly rooted in the physical world, the notion of physical location and position are very important, especially for finding things, but also for deriving knowledge. Therefore, the infrastructure has to support finding things according to location (e.g. geo-location based discovery). Taking mobility into account, localization technologies will play an important role for the Internet of Things and may become embedded into the infrastructure of the Internet of Things.

5. Security and Privacy

In addition, an infrastructure needs to provide support for security and privacy functions including identification, confidentiality, integrity, non-repudiation authentication and authorization. Here the heterogeneity and the need for interoperability among different ICT systems deployed in the infrastructure and the resource limitations of IoT devices (e.g., Nano sensors) have to be taken into account.

Logical design of Iot

The Logical design of IOT is however too abstract to be used for building directly concrete architectures. In order to implement a compliant IoT solutions, Reference Architectures must be defined, describing essential building blocks as well as design choices able to select specific constructs able to deal with converging requirements regarding functionality, performance, deployment and security, to name a few. Interfaces among different technological functional blocks should be standardized; best practices in terms of functionality and information usage need to be provided.

Existing literature provides methodologies for dealing with system architectures (hereafter called Concrete Architectures) based on Views and Perspectives. The way that the IoT-A project illustrates the Reference Architecture (RA) is through a *matrix* that provides clear technological choices in order to develop concrete architectures. To establish the contents of this matrix we need to analyze all possible functionalities, mechanisms and protocols that can be used for building any concrete IoT-related architecture and to show how interconnections could take place between selected design and technological choices. A system architect should then have a tool to make a rational selection of protocols, functional components, and architectural options, needed to build specific IoT systems.

The IoT-A project sees views as a representation of one or more structural aspects of an architecture that illustrates how the architecture addresses one or more concerns held by one or more of its stakeholders. Some typical examples for viewpoints are Functional, Information, Concurrency, Development, Deployment and Operational viewpoints. However, architectural decisions often address concerns that are common to more than one view. These concerns are often related to non-functional or quality properties.

The approach that the project is following is to define special perspectives to address these aspects of a concrete architecture, emphasizing the importance of stakeholder requirements. Therefore we are define a perspective as a collection of activities, tactics, and guidelines that are used to ensure that a system exhibits a particular set of related quality properties that require consideration across a number of the system's architectural views, where a quality property is defined as an externally visible, non-functional property of a system such as performance, security, or scalability.

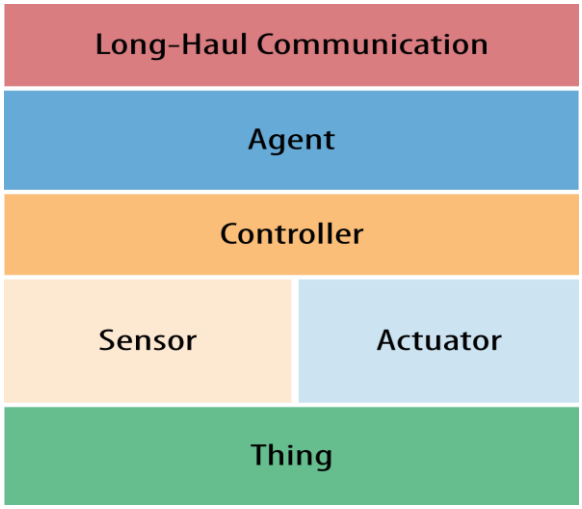


Fig. 1.4 IOT Logical View

IoT applications-Potential applications of the IoT are numerous and diverse, permeating into practically all areas of every-day life of individuals, enterprises, and society as a whole. The IoT application covers smart environments/spaces in domains such as: Transportation, Building, City, Lifestyle, Retail, Agriculture, Factory, Supply chain, Emergency, Healthcare, User interaction, Culture and tourism, Environment and Energy. Below are some of the IOT applications.

- **IOsL (Internet of smart living)-Remote Control Appliances:** Switching on and off remotely appliances to avoid accidents and save energy, Weather: Displays outdoor weather conditions such as humidity, temperature, pressure, wind speed and rain levels with ability to transmit data over long distances, Smart Home Appliances: Refrigerators with LCD screen telling what's inside, food that's about to expire, ingredients you need to buy and with all the information available on a Smartphone app. Washing machines allowing you to monitor the laundry remotely, and. Kitchen ranges with interface to a Smartphone app allowing remotely adjustable temperature control and monitoring the oven's self-cleaning feature, Safety

Monitoring: cameras, and home alarm systems making people feel safe in their daily life at home, Intrusion Detection Systems: Detection of window and door openings and violations to prevent intruders, Energy and Water Use: Energy and water supply consumption monitoring to obtain advice on how to save cost and resources.

- **IOsC (Internet of smart cities)-Structural Health:** Monitoring of vibrations and material conditions in buildings, bridges and historical monuments, Lightning: intelligent and weather adaptive lighting in street lights, Safety: Digital video monitoring, fire control management, public announcement systems, Transportation: Smart Roads and Intelligent High-ways with warning messages and diversions according to climate conditions and unexpected events like accidents or traffic jams, Smart Parking: Real-time monitoring of parking spaces availability in the city making residents able to identify and reserve the closest available spaces, Waste Management: Detection of rubbish levels in containers to optimize the trash collection routes. Garbage cans and recycle bins with RFID tags allow the sanitation staff to see when garbage has been put out.
- **IOsE (Internet of smart environment)-**Air Pollution monitoring: Control of CO2 emissions of factories, pollution emitted by cars and toxic gases generated in farms, Forest Fire Detection: Monitoring of combustion gases and preemptive fire conditions to define alert zones, Weather monitoring: weather conditions monitoring such as humidity, temperature, pressure, wind speed and rain, Earthquake Early Detection, Water Quality: Study of water suitability in rivers and the sea for eligibility in drinkable use, River Floods: Monitoring of water level variations in rivers, dams and reservoirs during rainy days, Protecting wildlife: Tracking collars utilizing GPS/GSM modules to locate and track wild animals and communicate their coordinates via SMS.
- **IOsI (Internet of smart industry)-**Explosive and Hazardous Gases: Detection of gas levels and leakages in industrial environments, surroundings of chemical factories and inside mines, Monitoring of toxic gas and oxygen levels inside chemical plants to ensure workers and goods safety, Monitoring of water, oil and gas levels in storage tanks and Cisterns, Maintenance and repair: Early predictions on equipment malfunctions and service maintenance can be automatically scheduled ahead of an actual part failure by installing sensors inside equipment to monitor and send reports.
- **IOsH (Internet of smart health)-Patients Surveillance:** Monitoring of conditions of patients inside hospitals and in old people's home, Medical Fridges: Control of conditions inside freezers storing vaccines, medicines and organic elements, Fall Detection: Assistance for elderly or disabled people living independent, Dental: Bluetooth connected toothbrush with Smartphone app analyzes the brushing uses and gives information on the brushing habits on the Smartphone for private information or for showing statistics to the dentist, Physical Activity Monitoring: Wireless sensors placed across the mattress sensing small motions, like breathing and heart rate and large motions caused by tossing and turning during sleep, providing data available through an app on the Smartphone.
- **IOsE (internet of smart energy)-Smart Grid:** Energy consumption monitoring and management, Wind Turbines/ Power house: Monitoring and analyzing the flow of energy from wind turbines & power house, and two-way communication with consumers' smart meters to analyze consumption patterns, Power Supply Controllers: Controller for AC-DC power supplies that determines required energy, and improve energy efficiency with less energy waste for power supplies related to computers, telecommunications, and consumer electronics applications, Photovoltaic Installations: Monitoring and optimization of performance in solar energy plants.
- **IOsA (internet of smart agriculture)-Green Houses:** Control micro-climate conditions to maximize the production of fruits and vegetables and its quality, Compost: Control of humidity and temperature levels in alfalfa, hay, straw, etc. to prevent fungus and other microbial contaminants, Animal Farming/Tracking: Location and identification of animals grazing in open pastures or location in big stables, Study of ventilation and air quality in farms and detection of harmful gases from excrements, Offspring Care: Control of growing conditions of the offspring in animal farms to ensure its survival and health, field Monitoring: Reducing spoilage and crop waste with better monitoring, accurate ongoing data obtaining, and management of the agriculture fields, including better control of fertilizing, electricity and watering.

Physical View-It defines different layered level component required to create the device as mentioned in diagram.

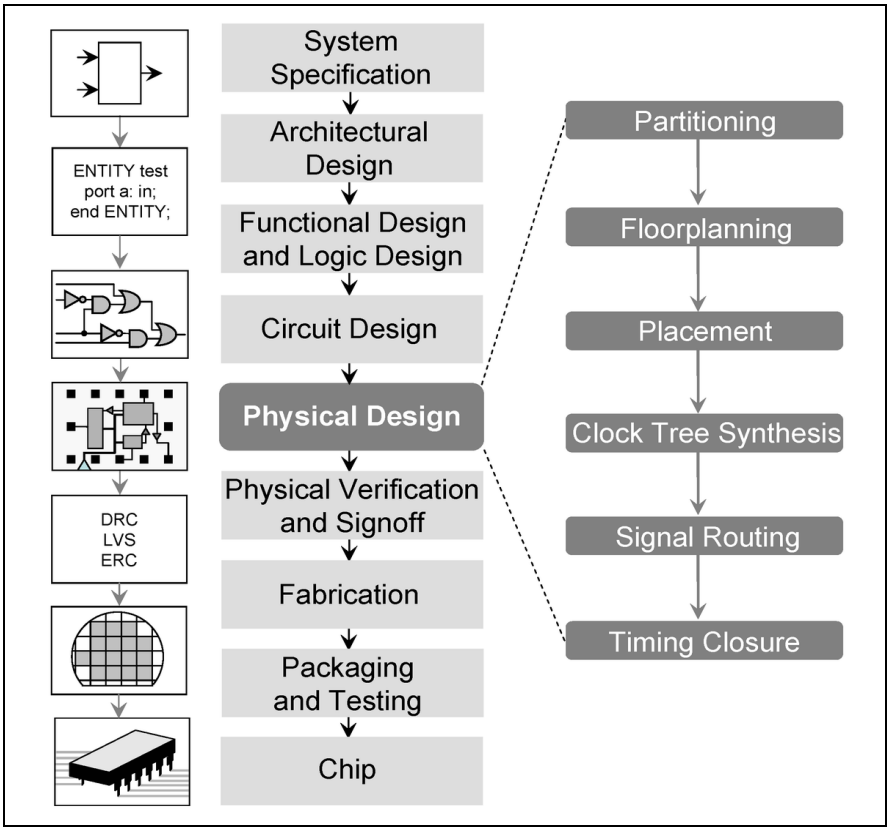


Fig. 1.5 Physical View

Machine-to-machine (M2M)

Machine to machine (M2M) is a broad label that can be used to describe any technology that enables networked devices to exchange information and perform actions without the manual assistance of humans. M2M communication is often used for remote monitoring. In product restocking, for example, a vending machine can message the distributor when a particular item is running low. M2M communication is an important aspect of warehouse management, remote control, robotics, traffic control, logistic services, supply chain management, fleet management and telemedicine. It forms the basis for a concept known as the Internet of Things (IoT). Key components of an M2M system include sensors, RFID, a Wi-Fi or cellular communications link and autonomic computing software programmed to help a networked device interpret data and make decisions. The most well-known type of M2M communication is telemetry, which has been used since the early part of the last century to transmit operational data. Pioneers in telemetric first used telephone lines and later, on radio waves -- to transmit performance measurements gathered from monitoring instruments in remote locations. The Internet and improved standards for wireless technology have expanded the role of telemetry from pure science, engineering and manufacturing to everyday use in products like home heating units, electric meters and Internet-connected appliances. Products built with M2M communication capabilities are often marketed to end users as being smart.

Fig. 2.1 M2M Communication

NFV (network function virtualization) for IOT-Utilizing NFV (network function virtualization) capabilities is one way to address the IoT network challenges providing secure network resources for IoT. Network functions virtualization (also Network function virtualization or NFV) is a network architecture concept that uses the technologies of IT virtualization to virtualizes entire classes of network node functions into building blocks that may connect, or chain together, to create communication services.NFV relies upon, but differs from, traditional server-virtualization techniques, such as those used in enterprise IT. A virtualized network function, or VNF, may consist of one or more virtual machines running different software and processes, on top of standard high-volume servers, switches and storage devices, or even cloud computing infrastructure, instead of having custom hardware appliances for each network function. For example, a virtual session border controller could be deployed to protect a network without the typical cost and complexity of obtaining and installing physical network protection units. Other examples of NFV include virtualized load balancers, firewalls, intrusion detection devices and WAN accelerators.

The NFV framework consists of three main components:

- Virtualized network functions (VNFs) are software implementations of network functions that can be deployed on a network functions virtualization infrastructure (NFVI).
- Network functions virtualization infrastructure (NFVI) is the totality of all hardware and software components that build the environment where VNFs are deployed. The NFV infrastructure can span several locations. The network providing connectivity between these locations is considered as part of the NFV infrastructure.
- Network functions virtualization management architectural framework (NFV-MANO Architectural Framework) is the collection of all functional blocks, data repositories used by these blocks, and reference points and interfaces through which these functional blocks exchange information for the purpose of managing and orchestrating NFVI and VNFs.

Relevance of NFV in IoT System-NFV can play a crucial role in achieving the goal with IoT network combining both hardware and software network features in a single virtual network. NFV helps accelerate the deployment of new services, operations, and maintenance of a network allowing high level of network optimization. It brings multiples benefits to service operators and service providers including ROI. The relevance of NFV lies with the promise of benefits across network architecture.

NFV to Enhance IoT Networking Capacity-NFV leverages couple of IT technologies to build flexible and agile IoT network such as virtualization, standard servers, and open software. It distributes intelligence throughout the IoT network enabling real time analytics and business intelligence. NFV creates menu for virtual network functions (VNFs) that includes gateways, mobile core, deep packet inspection (DPI), security, routing, and traffic management that helps delivering customized network services for IoT. Conversely IoT drives NFV opportunity for service providers too financially and technologically.

Data storage in IOT-The Internet of Things is creating an enormous amount of data. To manage, access, and make use of this data, digital storage becomes a critical factor. Data management is a broad concept referring to the architectures, practices, and procedures for proper management of the data lifecycle needs of a certain system. In the context of IoT, data management should act as a layer between the objects and devices generating the data and the applications accessing the data for analysis purposes and services. The devices themselves can be arranged into subsystems or subspaces with autonomous governance and internal hierarchical management. The functionality and data provided by these subsystems is to be made available to the IoT network, depending on the level of privacy desired by the subsystem owners.

- IoT data has distinctive characteristics that make traditional relational-based database management an obsolete solution. A massive volume of heterogeneous, streaming and geographically-dispersed real-time data will be created by millions of diverse devices periodically sending observations about certain monitored phenomena or reporting the occurrence of certain or abnormal events of interest . Periodic observations are most demanding in terms of communication overhead and storage due to their streaming and continuous nature, while events present time-strain with end-to-end response times depending on the urgency of the response required for the event. Furthermore, there is metadata that describes “Things” in addition to the data that is generated by “Things”; object identification, location, processes and services provided are an example of such data. IoT data will statically reside in fixed- or flexible-schema databases and roam the network from dynamic and mobile objects to concentration storage points. This will continue until it reaches centralized data stores. Communication, storage and process will thus be defining factors in the design of data management solutions for IoT.
- A data management framework for IoT is presented that incorporates a layered, data-centric, and federated paradigm to join the independent IoT subsystems in an adaptable, flexible, and seamless data network. In this framework, the “Things” layer is composed of all entities and subsystems that can generate data. Raw data, or simple aggregates, are then transported via a communications layer to data repositories. These data repositories are either owned by organizations or public, and they can be located at specialized servers or on the cloud. Organizations or individual users have access to these repositories via query and federation layers that process queries and analysis tasks, decide which repositories hold the needed data, and negotiate participation to acquire the data. In addition, real-time or context-aware queries are handled through the federation layer via a sources layer that seamlessly handles the discovery and engagement of data sources. The whole framework therefore allows a two-way publishing and querying of data. This allows the system to respond to the immediate data and processing requests of the end users and provides archival capabilities for later long-term analysis and exploration of value-added trends.

IoT Data Management-Traditional data management systems handle the storage, retrieval, and update of elementary data items, records and files. In the context of IoT, data management systems must summarize data online while providing storage, logging, and auditing facilities for offline analysis. This expands the concept of data management from offline storage, query processing, and transaction management operations into online-offline communication/storage dual operations. We first define the data lifecycle within the context of IoT and then outline the energy consumption profile for each of the phases in order to have a better understanding of IoT data management.

IoT Data Lifecycle-The lifecycle of data within an IoT system proceeds from data production to aggregation, transfer, optional filtering and preprocessing, and finally to storage and archiving. Querying and analysis are the end points that initiate (request) and consume data production, but data production can be set to be pushed to the IoT consuming services. Production, collection, aggregation, filtering, and some basic querying and preliminary processing functionalities are considered online, communication-intensive operations. Intensive preprocessing, long-term storage and archival and in-depth processing/analysis are considered offline storage-intensive operations.

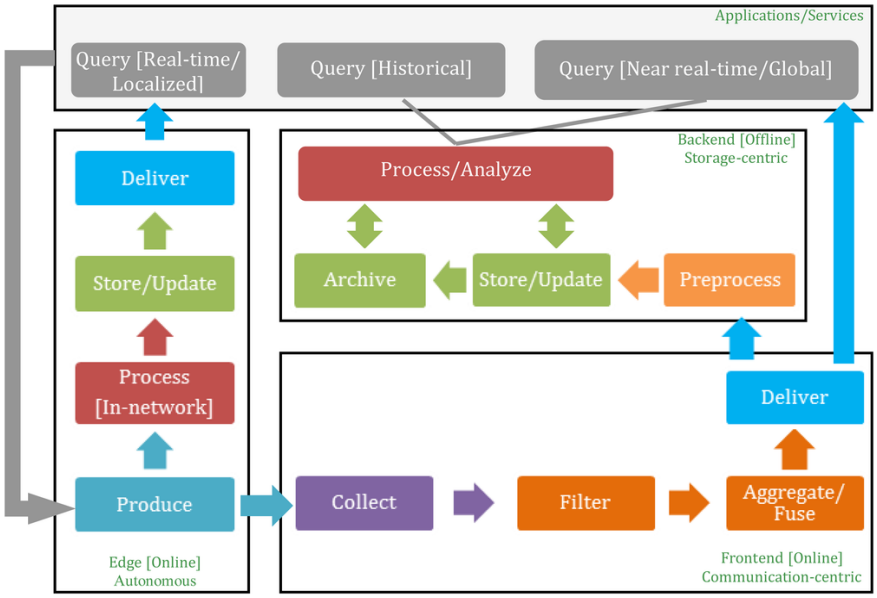


Fig 2.2 data Production

Storage operations aim at making data available on the long term for constant access/updates, while archival is concerned with read-only data. Since some IoT systems may generate, process, and store data in-network for real-time and localized services, with no need to propagate this data further up to concentration points in the system, edges that combine both processing and storage elements may exist as autonomous units in the cycle. In the following paragraphs, each of the elements in the IoT data lifecycle is explained.

- **Querying:** Data-intensive systems rely on querying as the core process to access and retrieve data. In the context of IoT, a query can be issued either to request real-time data to be collected for temporal monitoring purposes or to retrieve a certain view of the data stored within the system. The first case is typical when a (mostly localized) real-time request for data is needed. The second case represents more globalized views of data and in-depth analysis of trends and patterns.
- **Production:** Data production involves sensing and transfer of data by the “Things” within the IoT framework and reporting this data to interested parties periodically (as in a subscribe/notify model), pushing it up the network to aggregation points and subsequently to database servers, or sending it as a response triggered by queries that request the data from sensors and smart objects. Data is usually time-stamped and possibly geo-stamped, and can be in the form of simple key-value pairs, or it may contain rich audio/image/video content, with varying degrees of complexity in-between.
- **Collection:** The sensors and smart objects within the IoT may store the data for a certain time interval or report it to governing components. Data may be collected at concentration points or gateways within the network where it is further filtered and processed, and possibly fused into compact forms for efficient transmission. Wireless communication technologies such as Zigbee, Wi-Fi and cellular are used by objects to send data to collection points.
- **Aggregation/Fusion:** Transmitting all the raw data out of the network in real-time is often prohibitively expensive given the increasing data streaming rates and the limited bandwidth. Aggregation and fusion techniques deploy summarization and merging operations in real-time to compress the volume of data to be stored and transmitted.
- **Delivery:** As data is filtered, aggregated, and possibly processed either at the concentration points or at the autonomous virtual units within the IoT, the results of these processes may need to be sent further up the system, either as final responses, or for storage and in-depth analysis. Wired or wireless broadband communications may be used there to transfer data to permanent data stores.
- **Preprocessing:** IoT data will come from different sources with varying formats and structures. Data may need to be preprocessed to handle missing data, remove redundancies and integrate data from different sources into a unified schema before being committed to storage. This preprocessing is a known procedure in data mining called data cleaning. Schema integration does not imply brute-force fitting of all the data into a fixed relational (tables) schema, but rather a more abstract definition of a consistent way to access the data without having to customize access for each source's data format(s). Probabilities at different levels in the schema may be added at this phase to IoT data items in order to handle uncertainty that may be present in data or to deal with the lack of trust that may exist in data sources.

- **Storage/Update—Archiving:** This phase handles the efficient storage and organization of data as well as the continuous update of data with new information as it becomes available. Archiving refers to the offline long-term storage of data that is not immediately needed for the system's ongoing operations. The core of centralized storage is the deployment of storage structures that adapt to the various data types and the frequency of data capture. Relational database management systems are a popular choice that involves the organization of data into a table schema with predefined interrelationships and metadata for efficient retrieval at later stages. NoSQL key-value stores are gaining popularity as storage technologies for their support of big data storage with no reliance on relational schema or strong consistency requirements typical of relational database systems. Storage can also be decentralized for autonomous IoT systems, where data is kept at the objects that generate it and is not sent up the system. However, due to the limited capabilities of such objects, storage capacity remains limited in comparison to the centralized storage model.
- **Processing/Analysis:** This phase involves the ongoing retrieval and analysis operations performed and stored and archived data in order to gain insights into historical data and predict future trends, or to detect abnormalities in the data that may trigger further investigation or action. Task-specific preprocessing may be needed to filter and clean data before meaningful operations take place. When an IoT subsystem is autonomous and does not require permanent storage of its data, but rather keeps the processing and storage in the network, then in-network processing may be performed in response to real-time or localized queries.

Data Management Framework for IoT-Most of the current data management proposals are targeted to WSNs, which are only a subset of the global IoT space, and therefore do not explicitly address the more sophisticated architectural characteristics of IoT. WSNs are a mature networking paradigm whose data management solutions revolve mainly around in-network data processing and optimization. Sensors are mostly of stationary, resource-constrained nature, which does not facilitate sophisticated analysis and services. The main focus in WSN-based data management solutions is to harvest real-time data promptly for quick decision making, with limited permanent storage capacities for long-term usage. This represents only a subset of the more versatile IoT system, which aims at harnessing the data available from a variety of sources; stationary and mobile, smart and embedded, resource-constrained and resource-rich, real-time and archival. The main focus of IoT-based data management therefore extends the provisions made for WSNs to add provisions of a seamless way to tap into the volumes of heterogeneous data in order to find interesting global patterns and strategic opportunities.

IOT Cloud Based Services-As these devices start to become connected, we need a place to send, store, and process all of the information. Setting up your own in-house system isn't practical anymore. The cost of maintaining, upgrading and securing a system is just too high, and there are some great services available.

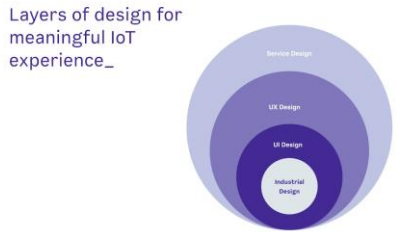
- **Amazon Web Services Iot Platform-**Amazon dominates the consumer cloud market. They were the first to really turn cloud computing into a commodity way back in 2004. Since then they've put a lot effort into innovation and building features, and probably have the most comprehensive set of tools available.
- **Microsoft Azure Iot Hub**-Microsoft is taking their Internet of Things cloud services very seriously. They have cloud storage, machine learning, and IoT services, and have even developed their own operating system for IoT devices. This means they intend to provide a complete IoT solution provider.
- **IBM Watson Iot Platform**-IBM is another IT giant trying to set itself up as an Internet of Things platform authority. They try to make their cloud services as accessible as possible to beginners with easy apps and interfaces. You can try out their sample apps to get a feel for how it all works. You can also store your data for a specified period, to get historical information from your connected devices.
- **Google Cloud Platform**-Search giant Google is also taking the Internet of Things very seriously. They claim that Cloud Platform is the best place to build IoT initiatives, taking advantage of Google's heritage of web-scale processing, analytics, and machine intelligence.

Design principles and needed capabilities

In the near future, our everyday lives will be more and more filled with intelligent, connected objects. They will appear in our homes, in our working environments and in the cities we live in as well as travel with us everywhere we go in the form of wearables, smart clothing and things we cannot even imagine right now. This development is called the internet of things, IoT.

For designers focused on designing SW services and screen based interfaces or physical products, designing IoT solutions creates totally new design challenges. IoT solutions consist of multiple elements: physical devices like sensors, actuators and interactive devices, the network connecting these devices, the data gathered from these devices

and analyzed to create a meaningful experience and last but definitely not least, the physical context in which user interacts with the solution. You need to do various types of design, from industrial product design to service and business design. All of these factors have their impact to the total UX of the IoT system and the task of designing in this context may feel quite overwhelming. To make it a little easier, I have gathered my list of the 7 most important design principles for IoT.



1. Focus on value

In the world of IoT, user research and service design are more crucial than ever. While early adopters are eager to try out new technology, many others are reluctant to take new technology into use and cautious about using it, due to not feeling confident with it. For your IoT solution to become widely adopted, you need to dig deep into users' needs in order to find out where lies a problem truly worth solving and what is the real end user value of the solution. You also need to understand what might be the barriers of adopting the new technology in general and your solution specifically. For deciding on your feature set, you need research too. The features that might be valuable and highly relevant for the tech early adopters may be uninteresting for the majority of the users and vice versa, so you need to plan carefully what features to include and in which order.

2. Take a holistic view

IoT solutions typically consist of multiple devices with different capabilities and both physical and digital touchpoints. The solution may also be provided in co-operation with multiple different service providers. It is not enough to design one of the touchpoints well, instead you need to take a holistic look across the whole system, the role of each device and service, and the conceptual model of how user understands and perceives the system. The whole system needs to work seamlessly together in order to create a meaningful experience.

3. Put safety first

As the IoT solutions are placed in the real world context, the consequences can be serious, when something goes wrong. At the same time the users of the IoT solutions may be vary of using new technology, so building trust should be one of your main design drivers. Trust is built slowly and lost easily, so you really need to make sure that every interaction with the product/service builds the trust rather than breaks it. What it means in practise? First of all, it means understanding possible error situations related to context of use, HW, SW and network as well as to user interactions and trying to prevent them. Secondly, if the error situations still occur, it means appropriately informing the user about them and helping them to recover. Secondly, it means considering data security & privacy as key elements of your design. It is really important for users to feel, that their private data is safe, their home, working environment and everyday objects cannot be hacked and their loved ones are not put at risk. Thirdly, quality assurance is critical and it should not only focus on testing the SW, but on testing the end to end system, in a real-world context.

4. Consider the context

IoT solutions exist at the crossroads of the physical and digital worlds. Commands given through digital interfaces may produce real world effects, but unlike digital commands, the actions happening in the real-world cannot necessarily be undone. In the real world context lots of unexpected things can happen and at the same time user should be able to feel safe and in control. The context places also other kind of requirements to the design. Depending on the physical context, the goal might be to minimize distraction of the user or e.g. to design devices that hold up against changing weather conditions. IoT solutions in homes, workplaces and public areas are are typically multi-user systems

and thus less personal than e.g. screen based solutions used in smartphones, which also brings into picture the social context where the solution is used and its' requirements for the design.

5. Build a strong brand

Due to the real world context of the IoT solutions, regardless of how carefully you design things and aim to build trust, something unexpected will happen at some point and your solution is somehow going to fail. In this kind of situations, it is of utmost importance, that you have built a strong brand that truly resonates with the end users. When they feel connected to your brand, they will be more forgiving about the system failures and will still keep on using your solution. While designing your brand, you must keep in mind, that trust should be a key element of the brand, one of the core brand values. This core value should also be reflected in the rest of the brand elements, like the choice of color, tone of voice, imagery etc.

6. Prototype early and often

Typically HW and SW have quite different lifespans, but as successful IoT solution needs both the HW and SW elements, the lifespans should be aligned. At the same time, IoT solutions are hard to upgrade, because once the connected object is placed somewhere, it is not so easy to replace it with a newer version, especially if the user would need to pay for the upgrade and even the software within the connected object may be hard to update due to security and privacy reasons. Due to these factors and to avoid costly hardware iterations, it's crucial to get the solution right, from the beginning of implementation. What this means from the design perspective is that prototyping and rapid iteration of both the HW and the whole solution are essential in the early stages of the project. New, more creative ways of prototyping and faking the solution are needed.

7. Use data responsibly

IoT solutions can easily generate tons of data. However, the idea is not to hoard as much data as possible, but instead to identify the data points that are needed to make the solution functional and useful. Still, the amount of data may be vast, so it's necessary for the designer to understand the possibilities of data science and how to make sense of the data. Data science provides a lot of opportunities to reduce user friction, i.e. reducing use of time, energy and attention or diminishing stress. It can be used to automate repeated context dependent decisions, to interpret intent from incomplete/inadequate input or to filter meaningful signals from noise. Understanding what data is available and how it can be used to help the user is a key element in designing successful IoT services.

What is iot design?

Iot design is the practice of gathering data of various iot systems and their interactions with a goal of creating a meaningful user experiences. Iot design takes a holistic look across the whole system, the role of each device and service, and the creates conceptual model of how user understands and perceives the entire iot system.

Why is iot design important?

For Iot solutions to become widely adopted, businesses need to dig deep into users' needs in order to find out where lies a problem truly worth solving and what is the real end user value of the iot solution. Businesses also need to understand what the barriers might be in adopting the new iot technologies in general and their iot solutions specifically.

What are some of the best practices of iot design?

Considering user safety, data security & privacy aspects and incorporating quality assurance in the iot design process are important for user trust. Prototyping early and rapid iteration of the hardware and the whole iot solution are essential. Understanding what data is available of how it can be used responsibly to help the user are key elements in successful iot design.

IoT Sensing and Actuation

Sensor Technology

A sensor is a device that detects and responds to some type of input from the physical environment. The specific input could be light, heat, motion, moisture, pressure, or any one of a great number of other environmental

phenomena. The output is generally a signal that is converted to human-readable display at the sensor location or transmitted electronically over a network for reading or further processing.
Here are a few examples of the many different types of sensors:
In a mercury-based glass thermometer, the input is temperature. The liquid contained expands and contracts in response, causing the level to be higher or lower on the marked gauge, which is human-readable.
An oxygen sensor in a car's emission control system detects the gasoline/oxygen ratio, usually through a chemical reaction that generates a voltage. A computer in the engine reads the voltage and, if the mixture is not optimal, readjusts the balance.
Motion sensors in various systems including home security lights, automatic doors and bathroom fixtures typically send out some type of energy, such as microwaves, ultrasonic waves or light beams and detect when the flow of energy is interrupted by something entering its path.
A photo sensor detects the presence of visible light, infrared transmission (IR), and/or ultraviolet (UV) energy.

Participatory Sensing-Participatory sensing is the concept of communities (or other groups of people) contributing sensory information to form a body of knowledge.
A growth in mobile devices, for example Smartphones, Tablet computers or Activity trackers, which have multiple sensors, has made participatory sensing viable in the large-scale. Participatory sensing can be used to retrieve information about the environment, weather, urban mobility, congestion as well as any other sensory information that collectively forms knowledge. Such open communication systems could pose challenges to the veracity of transmitted information. Individual sensors may require a trusted platform or hierarchical trust structures. It also includes effective incentives for participation, security, reputation and privacy.

Industrial IOT -The Industrial Internet of Things (IIoT) is the use of Internet of Things (IoT) technologies in manufacturing. Also known as the Industrial Internet, IIoT incorporates machine learning and data technology, harnessing the sensor data, machine-to-machine (M2M) communication and automation technologies that have existed in industrial settings for years. The driving philosophy behind the IIoT is that smart machines are better than humans at accurately, consistently capturing and communicating data. This data can enable companies to pick up on inefficiencies and problems sooner, saving time and money and supporting business intelligence efforts. In manufacturing specifically, IIoT holds great potential for quality control, sustainable and green practices, supply chain traceability and overall supply chain efficiency. Industrial Internet makes a connected enterprise by merging the information and operational department of the industry. Thus improving visibility, boosting operational efficiency, increases productivity and reduces the complexity of process in the industry. Industrial IoT is a transformative manufacturing strategy that helps to improve quality, safety, productivity in an industry.

Automotive IOT-With the number of networked sensors increasing across production, supply chains and products, manufacturers are beginning to tap into a new generation of systems that enables real-time, automatic interactions among machines, systems, assets and things. The pervasiveness of connected devices is finding applicability across multiple segments of manufacturing and Supply chain throughout the value chain.

- Ability to view the status of the Assets at anytime, Anywhere & Faster service response from dealer.
- By hooking equipment into the IoT, original equipment manufacturers (OEMs) or dealers could use that stream of data to adjust preventative maintenance schedules based on actual wear and be able to better optimize uptime
- Understand, monitor, predict and control process variability
- Enhance equipment and process diagnostics capabilities
- IoT helps more hands-off way to track goods and the progress of work. RFID tags and readers can play a role in this by allowing materials, locations, or tooling to essentially talk with each other.
- Faster Response time and less operations cost for machine configuration requests that could be services remotely
- Ability to view the entire population of connected products together marketing data and product trends & increased trouble shooting ability for Manufacturer's tech support
- Real-time remote monitoring of performance
- Multi site monitoring improving the operational efficiency and reducing the site downtime
- Availability of real time data for the production environment and alerts generated to the local administrators mobile phone reducing the clean room downtime
- Full manufacturing & SCM traceability
- Predictive Maintenance and quality

Actuator-An actuator is a component of a machine that is responsible for moving or controlling a mechanism or system, for example by actuating (opening or closing) a valve; in simple terms, it is a mover.
An actuator requires a control signal and a source of energy. The control signal is relatively low energy and may be electric voltage or current, pneumatic or hydraulic pressure, or even human power. The supplied main energy source may be electric current, hydraulic fluid pressure, or pneumatic (gas pressure). When the control signal is received, the actuator responds by converting the energy into mechanical motion.
An actuator is the mechanism by which a control system acts upon an environment. The control system can be simple (a fixed mechanical or electronic system), software-based (e.g. a printer driver, robot control system), a human, or any other input.

What is XaaS (anything as a service)?

"Anything as a service" (XaaS) describes a general category of services related to cloud computing and remote access. It recognizes the vast number of products, tools, and technologies that are now delivered to users as a service over the internet. Essentially, any IT function can be transformed into a service for enterprise consumption. The service is paid for in a flexible consumption model rather than as an upfront purchase or license.

What are the benefits of XaaS?

There are several benefits of XaaS: improving the expense model, speeding new apps and business processes, and shifting IT resources to higher-value projects.

Improving the expense model. With XaaS, businesses can cut costs by purchasing services from providers on a subscription basis. Before XaaS and cloud services, businesses had to buy individual products—software, hardware, servers, security, infrastructure—install them on site, and then link everything together to create networks. Now, with XaaS, businesses simply buy what they need, and pay as they go. Previous capital expenses now become operating expenses.
Speeding new apps and business processes. This model allows businesses to quickly adapt to changing market conditions with new apps or solutions. Using multitenant approaches, cloud services can provide much-needed flexibility. Resource pooling and rapid elasticity support mean that business leaders can simply add or subtract services as needed. A company can quickly access new technologies, scaling infrastructure automatically when users need innovative resources.
Shifting IT resources to higher-value projects. Increasingly, IT organizations are turning to an XaaS delivery model to streamline operations and free up resources for innovation. They are also using the benefits of XaaS to transform digitally and become more agile. In a recent survey by Deloitte, 71% of companies report that XaaS now constitutes more than half of their company's enterprise IT. XaaS provides more users with access to cutting-edge technology, democratizing innovation.

What are the disadvantages of XaaS?

XaaS has some potential drawbacks: possible downtime, performance issues, and complexity.

Possible downtime. The internet sometimes breaks, and when it does, your XaaS provider might have problems as well. With XaaS, there can be issues of internet reliability, resilience, provisioning and managing the infrastructure resources . If XaaS servers go down, users won't be able to use them. XaaS providers can guarantee services through SLAs.

Performance issues. As XaaS becomes more popular, bandwidth, latency, data storage, and retrieval times can suffer . If too many customers use the same resources, the system can slow down. Apps running in virtualized environments can also face impacts. In these complex environments, there can be integration issues, including the ongoing management and security of multiple cloud services.

Complexity impacts. Pushing technology to XaaS can relieve IT staff of day-to-day operational headaches; however, if something does go wrong, it might be harder to troubleshoot. The internal IT staff still needs to stay current on the new technology. Costs for maintaining high-performing, robust networks can increase—although the overall cost savings of XaaS models are usually much greater. Nonetheless, some companies want to retain visibility into their XaaS service provider's environment and infrastructure. In addition, an XaaS provider that gets acquired, discontinues a service, or alters its roadmap can have a profound impact on XaaS users.

What are some examples of XaaS?

Because XaaS stands for "anything as a service," the list of examples is endless. Many kinds of IT resources or services are now delivered this way. Broadly speaking, there are three categories of cloud computing models: software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS). Outside these categories, there are other examples such as disaster recovery as a service (DRaaS), communications as a service (CaaS), network as a service (NaaS), database as a service (DBaaS), storage as a service (STaaS), desktop as a service (DaaS), and monitoring as a service (MaaS). Other emerging industry examples include marketing as a service and healthcare as a service.

NetApp and XaaS

NetApp provides several XaaS options, including IaaS, IT as a service (ITaaS), STaaS, and PaaS.

IaaS. When you differentiate your hosted and managed infrastructure services, you can increase service and platform revenue, improve customer satisfaction, and turn IaaS into a profit center. You can also take advantage of new opportunities to differentiate and expand services and platform revenue, including delivering more performance and predictability from your IaaS services. Plus, NetApp® technology can enable you to offer a competitive advantage to your customers and reduce time to market for deploying IaaS solutions.

ITaaS. When your data center is in a private cloud, it takes advantage of cloud features to deliver ITaaS to internal business users. A private cloud offers characteristics similar to the public cloud but is designed for use by a single organization. These characteristics include:

- Catalog-based, on-demand service delivery
- Automated scalability and service elasticity
- Multitenancy with shared resource pools
- Metering with utility-style operating expense models
- Software-defined, centrally managed infrastructure
- Self-service lifecycle management of services

STaaS. NetApp facilitates private storage as a service in a pay-as-you-go model by partnering with various vendors, including Arrow Electronics, HPE ASE, BriteSky, DARZ, DataLink, Faction, Forsythe, Node4, Proact, Solvinity, Synoptek, and 1901 Group. NetApp also seamlessly integrates with all major cloud service providers including AWS, Google Cloud, IBM Cloud, and Microsoft Azure.

PaaS. NetApp PaaS solutions help simplify a customer's application development cycle. Our storage technologies support PaaS platforms to:

- Reduce application development complexity.
- Provide high-availability infrastructure.
- Support native multitenancy.
- Deliver webscale storage.

PaaS services built on NetApp technology enable your enterprise to adopt hybrid hosting services—and accelerate your application-deployment time.

Role of Cloud in IoT

One component that improves the success of the Internet of Things is Cloud Computing. Cloud computing enables users to perform computing tasks using services provided over the Internet. The use of the Internet of Things in conjunction with cloud technologies has become a kind of catalyst: the Internet of Things and cloud computing are now related to each other. These are true technologies of the future that will bring many benefits.

Due to the rapid growth of technology, the problem of storing, processing, and accessing large amounts of data has arisen. Great innovation relates to the mutual use of the Internet of Things and cloud technologies. In combination, it will be possible to use powerful processing of sensory data streams and new monitoring services. As an example, sensor data can be uploaded and saved using cloud computing for later use as intelligent monitoring and activation using other devices. The goal is to transform data into insights and thus drive cost-effective and productive action.

Benefits And Functions of IoT Cloud:

There are many benefits of combining these services –

1. IoT Cloud Computing provides many connectivity options, implying large network access. People use a wide range of devices to gain access to cloud computing resources: mobile

devices, tablets, laptops. This is convenient for users but creates the problem of the need for network access points.
2. Developers can use IoT cloud computing on-demand. In other words, it is a web service accessed without special permission or any help. The only requirement is Internet access.
3. Based on the request, users can scale the service according to their needs. Fast and flexible means you can expand storage space, edit software settings, and work with the number of users. Due to this characteristic, it is possible to provide deep computing power and storage.
4. Cloud Computing implies the pooling of resources. It influences increased collaboration and builds close connections between users.
5. As the number of IoT devices and automation in use grows, security concerns emerge. Cloud solutions provide companies with reliable authentication and encryption protocols.
6. Finally, IoT cloud computing is convenient because you get exactly as much from the service as you pay. This means that costs vary depending on use: the provider measures your usage statistics. A growing network of objects with IP addresses is needed to connect to the Internet and exchange data between the components of the network.

It is important to note that cloud architecture must be well-designed since reliability, security, economy, and performance optimization depends upon it. Using well-designed CI/CD pipelines, structured services, and sandboxed environments results in a secure environment and agile development.

What is the role of Cloud computing in IoT ?

1. Enables remote computing capabilities:

With a large storage capacity, IoT eliminates the dependencies on on-site infrastructure. With continued development and internet-based tech development such as the internet and devices supporting advanced cloud solutions, cloud technology has become mainstream. Packed with IoT, cloud solutions provides enterprises with the capability to access remote computing services with a single click or command.

2. Security & Privacy:

Tasks can be handled automatically with cloud tech & IoT, organizations are able to reduce security threats by a considerable amount. A cloud tech-enabled with IoT is a solution that provides preventive, detective and corrective control. With effective authentication and encryption protocols, it also provides users with strong security measures. Protocols such as biometrics in IoT products help manage as well as safeguard user identities along with data.

3. Data Integration:

Current tech developments have not only integrated IoT and cloud smoothly but also provide real-time connectivity and communication. This in turn makes the extraction of real-time information about key business processes and performing on-spot data integration with 24/7 connectivity easy. Cloud-based solutions with powerful data integration capabilities are able to handle a large amount of data generated from multiple sources along with its centralized storage, processing and analysis.

4. Minimal Hardware Dependency:

Presently, several IoT solutions offer plug-and-play hosting services that are enabled by integrating the cloud with the IoT. With cloud-enabled, IoT hosting providers need not rely on any kind of hardware or equipment to support the agility required by IoT devices. It has become easy for organizations to implement large scale IoT strategies seamlessly across platforms and move to omnichannel communication.

5. Business Continuity:

Known for their agility and reliability, cloud computing solutions are able to provide business continuity in case of any emergency, data loss or disaster. Cloud services operate via a network of data servers located in multiple geographical locations storing multiple copies of data backup. In case of any emergency, IoT based operations continue to work and data recovery becomes easy.

6. Communication Between Multiple Devices & Touchpoint:

IoT devices and services need to connect with each other and communicate to perform tasks that are enabled using cloud solutions. By supporting several robust APIs, cloud & IoT is able to interact amongst themselves and connected devices. Having a cloud supported communication helps fasten the interaction happen seamlessly.

7. Response Time & Data Processing:

Edge computing combined with IoT solutions usually shortens response time and speeds up data processing capabilities. It requires the deployment of IoT with cloud computing and edge computing solutions for maximum utilization.

Though cloud computing services can accelerate the growth of IoT, there are certain challenges in deploying these services successfully. The combination of IoT and cloud presents a few obstacles that need to be handled beforehand.

What are the challenges the Cloud & IoT brings together?

1. Large Amount of Data:

Processing a large amount of data can be tiring and overwhelming, especially with countless devices working at multiple touchpoints. This can threaten the overall performance of the application. Therefore constant monitoring of the system and data backup is advised.

2. Network and Communication Protocol:

Cloud and IoT devices involve multiple touchpoint communication using numerous protocols. Since it is an internet-dependent service, it is difficult to manage the change sometimes. Internet accessibility using wi-fi and mobile Internet can help resolve any challenges faced due to connectivity issues in such situations.

3. Sensor Network:

Sensor network allows users to process and understand the IoT environment and amplifies the benefit of IoT. But, processing larger chunks of data regularly is a major challenge faced by these networks.

Security aspects in IoT

The internet of things (IoT) is the vast network of connected physical objects (i.e., things) that exchange data with other devices and systems via the internet. While it refers to the actual devices, IoT is commonly used as an overarching term to describe a highly-distributed network that combines connectivity with sensors and lightweight

applications, which are embedded into tools and devices. These are used to exchange data with other devices, applications, and systems for everything from smart plugs and power grids to connected cars and medical devices.

Driven by low-cost computing and the cloud, IoT has become one of the most ubiquitous connected technologies with billions of instances around the world. IoT bridges the digital and physical worlds with seamless, streaming communications for everyday consumer products and complex industrial systems.

What Is IoT Security?

IoT security is an umbrella term that covers the strategies, tools, processes, systems, and methods used to protect all aspects of the internet of things. Included in IoT security is the protection of the physical components, applications, data, and network connections to ensure the availability, integrity, and confidentiality of IoT ecosystems.

Security challenges abound, because of the high volume of flaws regularly discovered in IoT systems. Robust IoT security includes all facets of protection, including hardening components, monitoring, keeping firmware updated, access management, threat response, and remediation of vulnerabilities. IoT security is critical as these systems are sprawling and vulnerable, making them a highly-targeted attack vector. Securing IoT devices from unauthorized access ensures that they do not become a gateway into other parts of the network or leak sensitive information.

IoT security vulnerabilities are found in everything from vehicles and smart grids to watches and smart home devices. For example, researchers found webcams that could be easily hacked to gain access to networks and smartwatches containing security vulnerabilities that allowed hackers to track the wearer's location and eavesdrop on conversations.

The Importance of IoT Security

IoT is widely believed to be one of the most significant security vulnerabilities that impact nearly everyone—consumers, organizations, and governments. For all of the convenience and value derived from IoT systems, the risks are unparalleled. The importance of IoT security cannot be overstated, as these devices provide cybercriminals with a vast and accessible attack surface.

IoT security provides the vital protections needed for these vulnerable devices. Developers of IoT systems are known to focus on the functionality of the devices and not on security. This amplifies the importance of IoT security and for users and IT teams to be responsible for implementing protections.

IoT Security Challenges

As noted above, IoT devices were not built with security in mind. This results in myriad IoT security challenges that can lead to disastrous situations. Unlike other technology solutions, few standards and rules are in place to direct IoT security. In addition, most people do not understand the inherent risks with IoT systems. Nor do they have any idea about the depth of IoT security challenges. Among the many IoT security issues are the following:

- Lack of visibility
  Users often deploy IoT devices without the knowledge of IT departments, which makes it impossible to have an accurate inventory of what needs to be protected and monitored.
- Limited security integration
  Because of the variety and scale of IoT devices, integrating them into security systems ranges from challenging to impossible.
- Open-source code vulnerabilities
  Firmware developed for IoT devices often includes open-source software, which is prone to bugs and vulnerabilities.
- Overwhelming data volume
  The amount of data generated by IoT devices make data oversight, management, and protection difficult.

- Poor testing
  Because most IoT developers do not prioritize security, they fail to perform effective vulnerability testing to identify weaknesses in IoT systems.
- Unpatched vulnerabilities
  Many IoT devices have unpatched vulnerabilities for many reasons, including patches not being available and difficulties accessing and installing patches.
- Vulnerable APIs
  APIs are often used as entry points to command-and-control centers from which attacks are launched, such as SQL injection, distributed denial of service (DDoS), man-in-the-middle (MITM), and breaching networks
- Weak passwords
  IoT devices are commonly shipped with default passwords that many users fail to change, giving cyber criminals easy access. In other cases, users create weak passwords that can be guessed.

Addressing IoT Security Challenges

A holistic approach is required to implement and manage IoT security effectively. It must encompass a variety of tactics and tools as well as take into consideration adjacent systems, such as networks.

Three key capabilities for a robust IoT security solution are the ability to:

1. Learn
   Take advantage of security solutions that provide network visibility to learn what the ecosystem encompasses at what the risk profiles are for each group of IoT devices.
2. Protect
   Monitor, inspect, and enforce IoT security policies commiserate with activities at different points in the infrastructure
3. Segment
   In the same way that networks are segmented, use segmentation based on policy groups and risk profiles to segment IoT systems.

Specific features required for securing IoT devices include the following:

- API security
- Broader and deep IoT device inventory
- Continuous software updates
- DNS filtering
- Education and training staff, vendors, and partners
- Encryption for data at rest and in transit
- Honeypot decoy programs
- Multi-factor authentication
- Network security
- Network traffic monitoring analysis
- Password management
- Patch management
- Security gateways
- Unauthorized IoT device scans

Enhance IoT Security to Realize Increased Benefits

IoT devices are increasingly being used by individuals and across the enterprise. They are not only here to stay, but proliferating exponentially in more and more forms. The result is increasing complexity, which hampers efforts to manage IoT systems security successfully.

IoT security challenges range from deflecting malicious insiders to defending against nation-state attacks. Because of the inherent vulnerability of IoT devices and the scale of their deployment, attacks continue to grow in scale and scope.

Securing IoT devices is well worth the investment despite the IoT security challenges. The value realized with IoT devices can only be increased with enhanced security to be on par with other technology. It will mitigate risks and increase rewards.

IoT Security Best Practices

The very first step in securing IoT is knowing what is connected. This includes using a device identification and discovery tool that automates three critical IoT security functions.

1. Automatically and continuously detects, profiles, and classifies IoT devices on the network
2. Maintains a real-time inventory of devices
3. Provides relevant risk insights for each of these asset classes by continuously monitoring across attack vectors.

By following these industry best practices for IoT security and adopting leading-edge solutions, you can understand, manage, and secure your complete asset inventory, including IoT.

Internet of Things Attacks: Deadly Dolls and Killer Cars

It's no secret: millions of IoT devices have terrible security. Yet people continue to buy them and they continue to surface in cyberattacks.

But does the internet of things pose a real threat?

What types of IoT attacks are being launched?

What vulnerabilities are being found?

Let's look at some of the nastiest threats to emerge from the land of internet-connected gadgets, widgets, and gizmos.

#1. IoT Dolls that Spy on Kids

Parents in Germany were shocked to learn a doll bought for children could be used to spy on them.

Federal Network Agency, a telecommunications watchdog in Germany, advised parents in Feb. 2016 to destroy the talking doll, called My Friend Cayla.

Cayla could connect to a smartphone via Bluetooth, giving the doll internet access. This connection allowed it to converse with children, answering simple questions such as, "What's two times two?"

Unfortunately, the IoT doll also recorded children's conversations and stored them in an online server (yikes!).

And it gets worse. The poor security of the doll's Bluetooth connection could easily allow an attacker to connect and use the toy as a spying device.

The U.S. Federal Trade Commission filed a complaint against Cayla's manufacturer, Genesis Toys, in Dec. 2016.

Here's the first paragraph of the FTC's complaint:

"This complaint concerns toys that spy. By purpose and design, these toys record and collect the private conversations of young children without any limitations on collection, use, or disclosure of this personal information. The toys subject young children to ongoing surveillance and are deployed in homes across the United States without

any meaningful data protection standards. They pose an imminent and immediate threat to the safety and security of children in the United States."

While no evidence of the doll being used in an IoT attack has surfaced, the size of the vulnerability and the potential impact on children are eye-opening.

#2. Click to Disable a Car's Brakes

Chrysler recalled 1.4 million vehicles in 2015 after security researchers demonstrated massive security gaps in the computer systems of Jeep Cherokees.

From a laptop miles away, Charlie Miller and Chris Valasek seized control of an SUV's brakes, transmission, and steering, all without physical access to the vehicle.

While a car is too large to consider a "gadget," its internet connectivity qualifies it for membership in the internet of things.

Leveraging zero-day vulnerabilities and an IoT feature that kept the car connected to a cellular network, anyone with the vehicle's IP address could connect to it, according to Wired.

After connecting, the researchers pivoted to a chip in the car's head unit and rewrote its code. This allowed them to issue commands through the car's internal computer network and control components such as the engine and brakes.

The researchers demonstrated terrifying control of the car – including the ability to disable its brakes, transmission, and engine.

Chrystler issued a patch to resolve the vulnerability and issued a recall – but when is the last time you patched a car's firmware?

#3. IoT Thermostat Held for Ransom

Security researchers not only compromised an IoT thermostat at Def Con 24, but also demonstrated how an attacker could lock the device and demand a ransom to restore functionality.

The team, Pen Test Partners, began the attack by searching for device information on the thermostat through the FCC ID Search page.

Some of the thermostat's hardware information and multiple product images were found through the search.

Upon further inspection, the hackers determined the thermostat had an **SD card port** used to customize its settings. The hackers used this feature to access the device's firmware.

Not only was the firmware easily accessed and unzipped, but it was also running in root by default, making it easy for the team to gain root access.

Even worse – after a little hacking, the team could inject malicious code into the IoT device without any additional authentication.

After creating a full-functioning version of ransomware, the attackers loaded it to the device via the SD port. The attack was successful.

The thermostat now showed a wallpaper displaying the text, "Ha! You Suck! Pay 1 Bitcoin to get control back."

#4. Teddy Bear Database Hacked

CloudPets, released by Spiral Toys, is a cute concept. The line of stuffed animals allows parents and children to record and share voice messages with each other.

For example, a traveling father can use the toy's smartphone app to send his daughter a voice message. Back at home, his daughter can listen to the message on her stuffed animal and record a response.

Similar to the Cayla doll mentioned above, the toys use Bluetooth to connect to a smartphone to gain internet access. This connection is used to send the voice recordings to a data base, where they are stored.

Unfortunately, more than 2 million of these deeply personal recordings were found unprotected in an online database, along with data on about 800,000 customer accounts.

"During the time the data was exposed, at least two security researchers, and likely malicious hackers, got their hands on it," according to Motherboard.

The exposed data included the email account used to set up a CloudPets account, the birth day and month of children, and the relationship of the account holder to the child.

Although the account passwords were encrypted, the company did not enforce ANY requirements for password strength. Entries such as "password" and "123" were allowed, which hackers could easily guess.

#5. IoT Attack Against Dyn

One of the most widely covered cyberattacks of 2016 used IoT devices to launch a massive DDoS attack.

DNS service provider Dyn estimates 100,000 endpoints flooded its architecture on Oct. 21, resulting in congestion and outages for websites such as Twitter, PayPal, Amazon, and Netflix.

To launch the attack, hackers used a botnet created with the Mirai strain of malware.

Mirai scans the web for vulnerable IoT devices, infects them, and secretly persists awaiting commands from the attacker.

Security cameras and DVRs allegedly comprise the bulk of infected devices in the Mirai botnet, although other hacked IoT devices, such as routers, are also present.

During the IoT attack against Dyn, traffic surges from the botnet are estimated to have peaked at a record-breaking 1.2 Tbps, although this is unconfirmed.

Consider the financial and reputational damage inflicted on Dyn by the attack.

Now also consider the cascading effect as services such as PayPal and Netflix were knocked offline, also hurting their reputations and bottom lines.

The destructive potential of attacks that use the internet of things is clear.

Unit -2

Programming API's (using Python/Node.js/Arduino)

Internet of Things (IoT) has been a hot buzzword for the last few years. Simply put, IoT is the concept of connecting objects to a network in order to transfer data without human-to-human or human-to-computer interaction. This simple, yet powerful concept has a wide range of applications in manufacturing, healthcare and home automation, just to name a few. IoT is an interdisciplinary field, which requires working with electronics and sensors to capture data from physical objects, computer networking for data transfer and IT skills for building meaningful applications.

In this blog post, I will introduce the building blocks for creating a simple IoT application. To do this, I will use an Arduino microcontroller with a photocell (light intensity sensor), a node.js server for capturing and transferring the data, and a cloud service called plotly to visualise the data. By the end of this tutorial, we will have a functioning IoT application that you can customise to other use-cases.

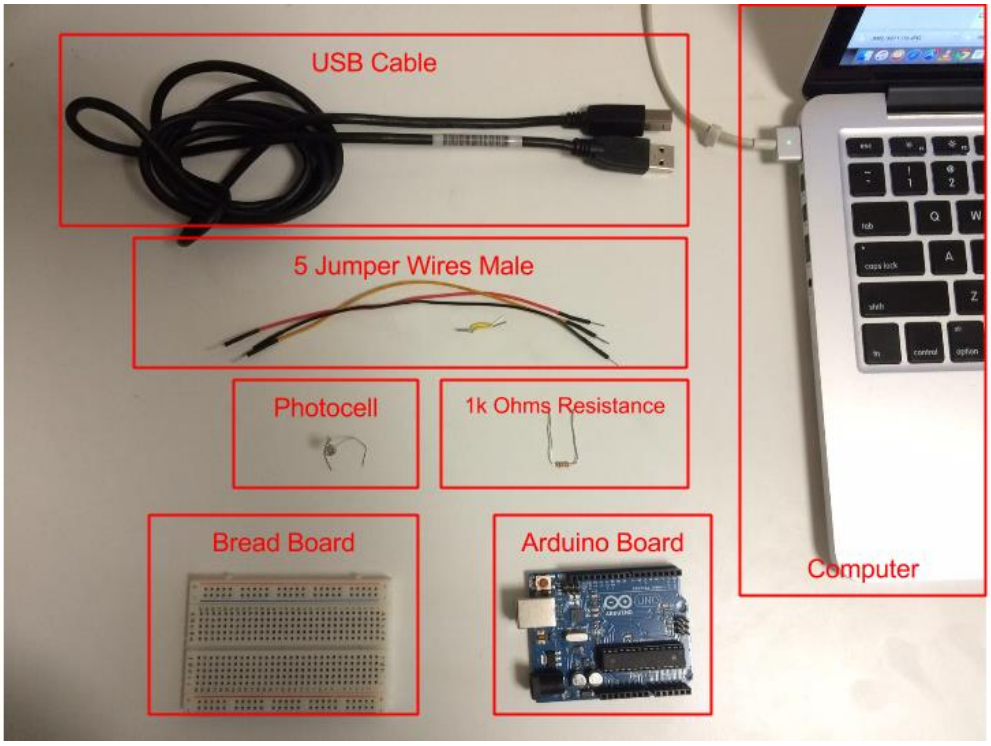
1. Use case definition and architecture

To keep things simple and focused, we will build a simple application that keeps measuring light intensities using a photocell and plots these values on a graph in real-time.

Required hardware:

- One Arduino microcontroller
- One photocell
- One 1k Ohms resistance
- One bread board
- Five jumper wires male
- One USB cable
- One computer (MAC or Windows) with internet connection

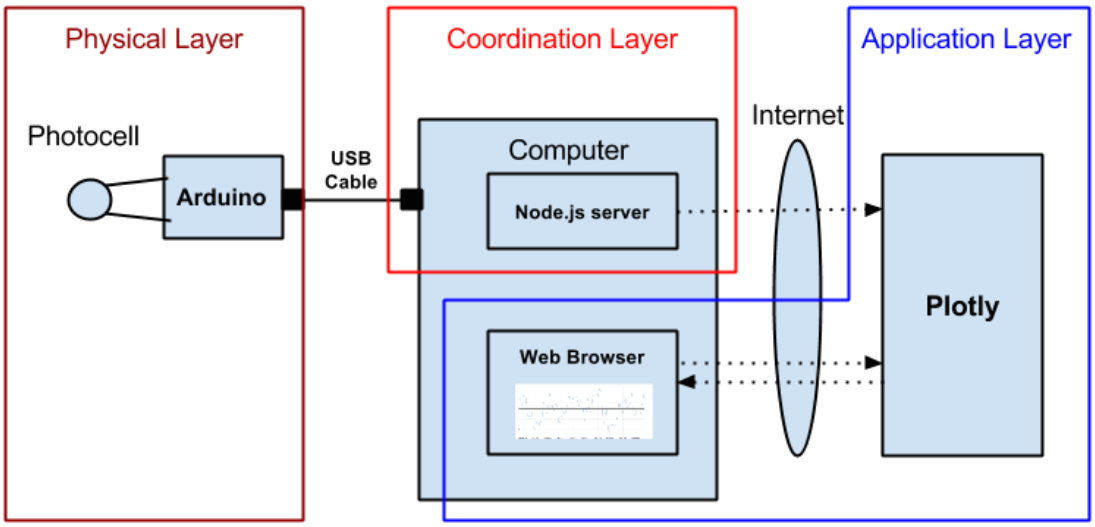
an Arduino Kit that contains an Arduino board with other electronic parts and components.



Architecture

To build this application, we need 3 main components:

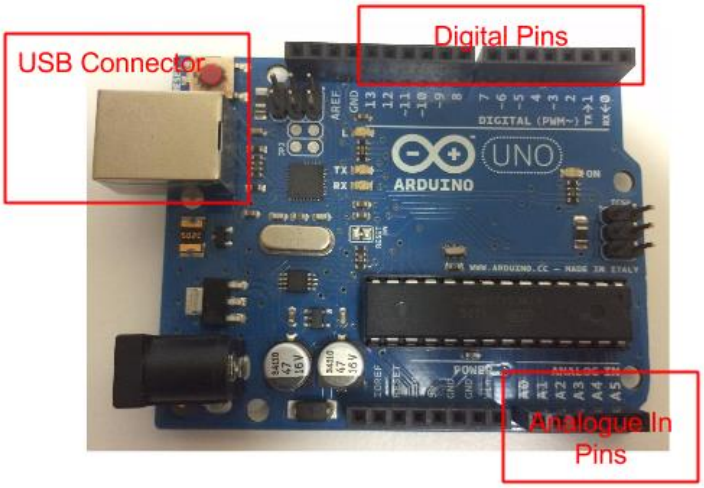
- A physical layer for capturing light intensities. We will implement this using an Arduino micro controller and a photocell.
- A coordination layer used for capturing the measurements from the physical layer, and sending the measurements to our application. We will implement this using node.js.
- An application layer for visualizing the measurements in real-time. We will implement this using a data visualization cloud service called Plotly.



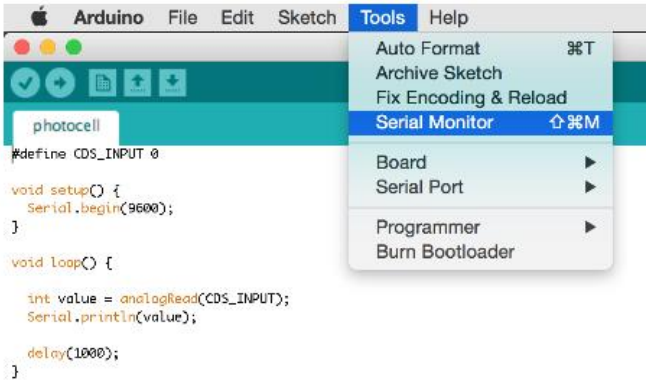
2. Microcontroller Programming 101 - A crash course in Arduino

Arduino is an open-source rapid electronic prototyping platform composed by the Arduino board (microcontroller) and the Arduino IDE (Integrated Development Environment) that runs on your computer. Arduino IDE is used to write and upload computer code to the physical board.

Arduino became very popular for electronic prototyping because of its very simple interface and low cost (under 30 USD for the board).



Arduino Board



Arduino IDE

Getting Started with Arduino

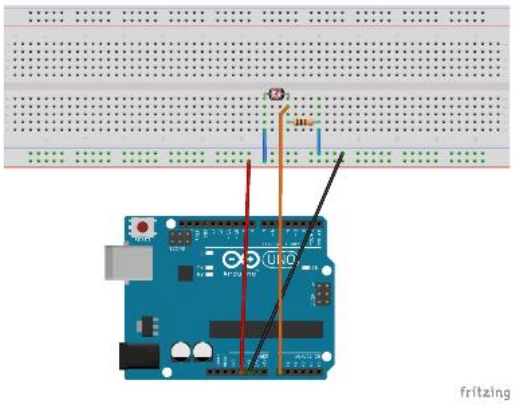
To setup your Arduino, I recommend starting with the following guides. It should take you around 10 min for the basic setup.

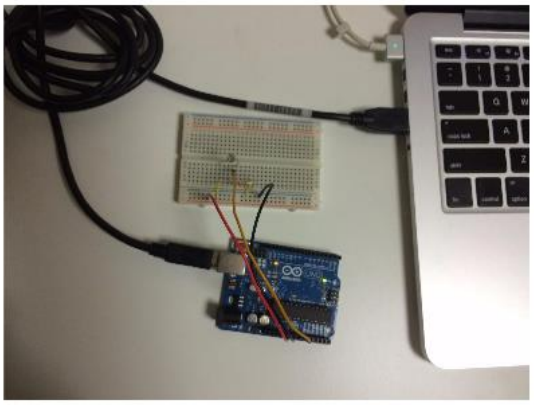
- Getting Started with Arduino on Windows
- Getting Started w/ Arduino on Mac OS X

Sensing light with Arduino and a photocell

After setting up the Arduino board and the Arduino IDE, we can start building the physical layer for capturing light intensities.

Start by connecting the electronic components (One Photocell and one 1k Ohms resistance) to the Arduino board and the bread board as shown in the picture below. Once done, connect the Arduino board to the computer using a USB cable.

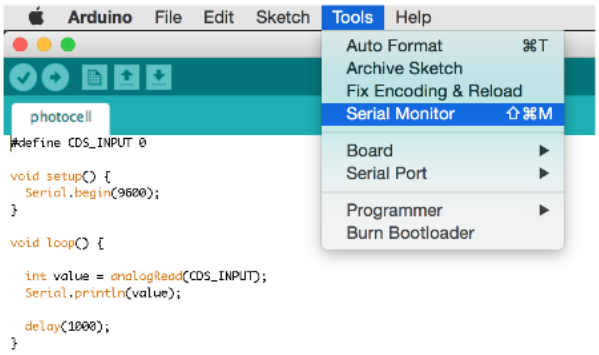


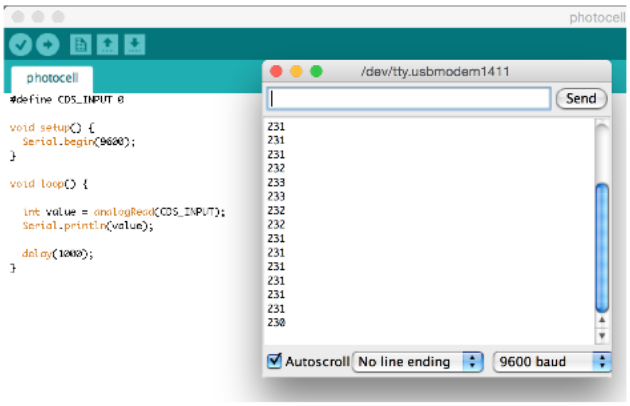


Next, open the IDE, copy the code below and upload the logic to the Arduino board. The code below implements a simple logic that configure the Arduino board to read the light intensity every second (1000 milliseconds) from the photocell (that is connected to the analogue input 0), and print the measurement to the serial monitor.

```
#define CDS_INPUT 0

void setup() {
  Serial.begin(9600);
}

void loop() {

  int value = analogRead(CDS_INPUT);
  Serial.println(value);

  delay(1000);
}
```

To see the measurements, open the serial monitor from the Arduino IDE.





3. Reading the measurements from Node.js

What is node.js?

Node.js is an open source, cross-platform runtime environment for server-side and networking applications. Node.js was originally designed as a tool for writing server programs, but it can do much more. It has a library management system called node package manager or npm that allows you to extend its functionality in many directions. To get started, download the node.js installer and install it on your machine.

Building Node.js server

In section 2, we used Arduino's serial monitor to display the measurements. In this section, we will build a node.js server that gets light intensities from Arduino and displays these measurements on the terminal.

To build this server, we need one node library called `serialport`. from your terminal, execute `npm install serialport` to install the library.

Next, create a file called `server1.js` and copy into it the code below. Make sure to put the correct port name in line 2. You can get this value from Arduino IDE > Tools > Serial Port.

```
var serialport = require('serialport');
var portName = '/dev/tty.usbmodem1411';
var sp = new serialport.SerialPort(portName, {
    baudRate: 9600,
    dataBits: 8,
    parity: 'none',
    stopBits: 1,
    flowControl: false,
    parser: serialport.parsers.readline("\r\n")
});

sp.on('data', function(input) {
    console.log(input);
});
```

To start the node.js server, from you terminal go to the folder where `server1.js` is saved, and execute `node server1.js`. You will see the measurements displayed on the terminal.

4. Sending data to Plotly

What is Plotly?

Plotly is an online analytics and data visualization tool. Plotly has a Streaming API, which makes it perfect for our use case.

Plotly account and API keys

Create a free Plotly account by going to this url.

After creating your account, go to your setting and get 3 pieces of information.

- Username
- API key
- Streaming API token

Install Plotly library for node.js

In this section, we will build a node.js server that will get the measurements from Arduino and send the data to Plotly. To do this, we need to install Plotly library for node.js by executing `npm install plotly` from the terminal.

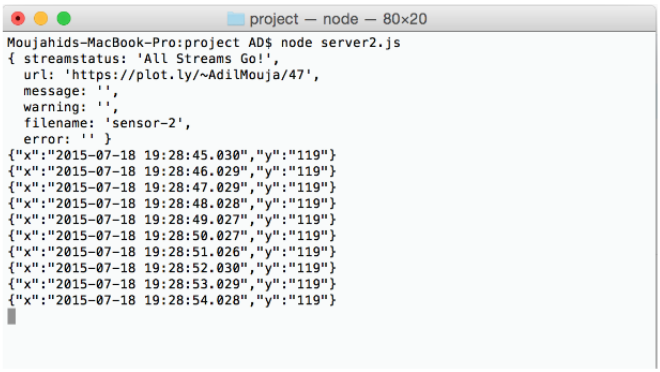
Connecting Node.js to Plotly

Create a file called server2.js and copy into it the code below. Make sure to put the correct plotly username name, API key and token in lines 2-3; and the correct Arduino port name in line 5.

```
var serialport = require('serialport'),
    plotly = require('plotly')('Plotly_UserName','Plotly_API'),
    token = 'Plotly_Token';


var portName = '/dev/tty.usbmodem1411';
var sp = new serialport.SerialPort(portName,{
    baudRate: 9600,
    dataBits: 8,
    parity: 'none',
    stopBits: 1,
    flowControl: false,
    parser: serialport.parsers.readline("\r\n")
});
```

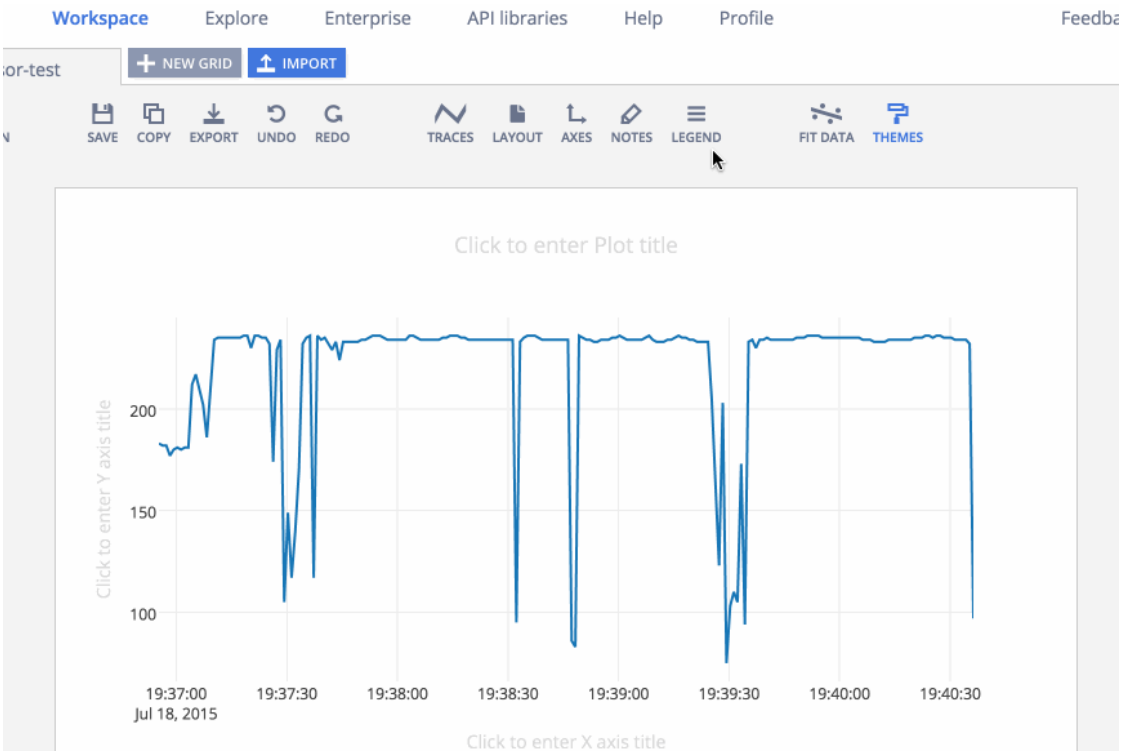
```
// helper function to get a nicely formatted date string
function getDateString() {
    var time = new Date().getTime();
    // 32400000 is (GMT+9 Japan)
    // for your timezone just multiply +/-GMT by 36000000
    var datestr = new Date(time +32400000).toISOString().replace(/T/, '
').replace(/Z/, '');
    return datestr;
}

var initdata = [{x:[], y:[], stream:{token:token, maxpoints: 500}}];
var initlayout = {fileopt : "extend", filename : "sensor-test"};

plotly.plot(initdata, initlayout, function (err, msg) {
    if (err) return console.log(err)

    console.log(msg);
    var stream = plotly.stream(token, function (err, res) {
        console.log(err, res);
    });

    sp.on('data', function(input) {
        if(isNaN(input) || input > 1023) return;

    var streamObject = JSON.stringify({ x : getDateString(), y : input });
    console.log(streamObject);
    stream.write(streamObject+'\n');
    });
});
```

To start the node.js server, from you terminal go to the folder where server2.js is saved, and execute `node server2.js`.



Viewing light intensities from Plotly

When you run server2.js code, it creates a file in Plotly called `sensor-test`. From Plotly website, click on `sensor-test` and you will be able to see a real-time graph that shows light intensities.



Here we build an end-to-end IoT application covering:

1. Getting measurements from an analogue input
2. Processing the data using node.js
3. Visualising the data using a 3rd party service

[ThingSpeak]

ThingSpeak can process HTTP requests and store and process data. Key features of the open data platform include an open API, real-time data collection, geolocation data, data processing and visualizations, device status messages and plugins. With ThingSpeak, the user can create sensor logging applications, location tracking applications, and a social network of things with status updates.

Movidius NCS:

It's an easy-to-use kit that allows you to design and implement applications such as classification and object recognition on physical products. We can simply think of Movidius NCS as a GPU (Graphics Processing Unit) running on a USB.

What is movidius neural compute stick?

The Intel Movidius Neural Compute Stick (NCS) is a tiny fanless deep-learning device that can be used to learn AI programming at the edge.

NodeMCU:

NodeMCU is an open-source Lua based firmware and development board specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.

The NodeMCU (Node MicroController Unit) is an open source software and hardware development environment that is built around a very inexpensive System-on-a-Chip (SoC) called the ESP8266. ... And, you have to program it in low-level machine instructions that can be interpreted by the chip hardware.

Is Alexa an IoT?

The term “Internet of Things” applies to any nonstandard computing device that connects to wifi and can transmit data. Well-known examples of IoT devices include smart speakers like Amazon Alexa or Google Home, smartwatches like the Apple Watch, internet-connected baby monitors, video doorbells, and even toys.

Aurdino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.

Features of Arduino Boards

Arduino Board	Processor	Analogue I/O
Arduino Uno	16Mhz ATmega328	6 input, 0 output
Arduino Due	84MHz AT91SAM3X8E	12 input, 2 output
Arduino Mega	16MHz ATmega2560	16 input, 0 output

Arduino and Raspberry pi difference?

These both teaching tools are suitable for beginners, hobbyists. The main difference between them is Arduino is microcontroller board while raspberry pi is a mini computer. Thus Arduino is just a part of raspberry pi. Raspberry Pi is good at software applications, while Arduino makes hardware projects simple.

Raspberry pi:

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python.

The Raspberry Pi operates in the open source ecosystem: it runs Linux (a variety of distributions), and its main supported operating system, Raspbian, is open source and runs a suite of open source software. The Raspberry Pi Foundation contributes to the Linux kernel and various other open source projects as well as releasing much of its own software as open source.

What operating systems can run on Raspberry Pi?

The Pi can run the official Raspbian OS, Ubuntu Mate, Snappy Ubuntu Core, the Kodi-based media centers OSMC and LibreElec, the non-Linux based Risc OS (one for fans of 1990s Acorn computers).

Raspberry Pi is the name of a series of single-board computers made by the Raspberry Pi Foundation, a UK charity that aims to educate people in computing and create easier access to computing education.

The Raspberry Pi launched in 2012, and there have been several iterations and variations released since then. The original Pi had a single-core 700MHz CPU and just 256MB RAM, and the latest model has a quad-core 1.4GHz CPU

with 1GB RAM. The main price point for Raspberry Pi has always been $35 and all models have been $35 or less, including the Pi Zero, which costs just $5.

All over the world, people use Raspberry Pis to learn programming skills, build hardware projects, do home automation, and even use them in industrial applications.

The Raspberry Pi is a very cheap computer that runs Linux, but it also provides a set of GPIO (general purpose input/output) pins that allow you to control electronic components for physical computing and explore the Internet of Things (IoT).

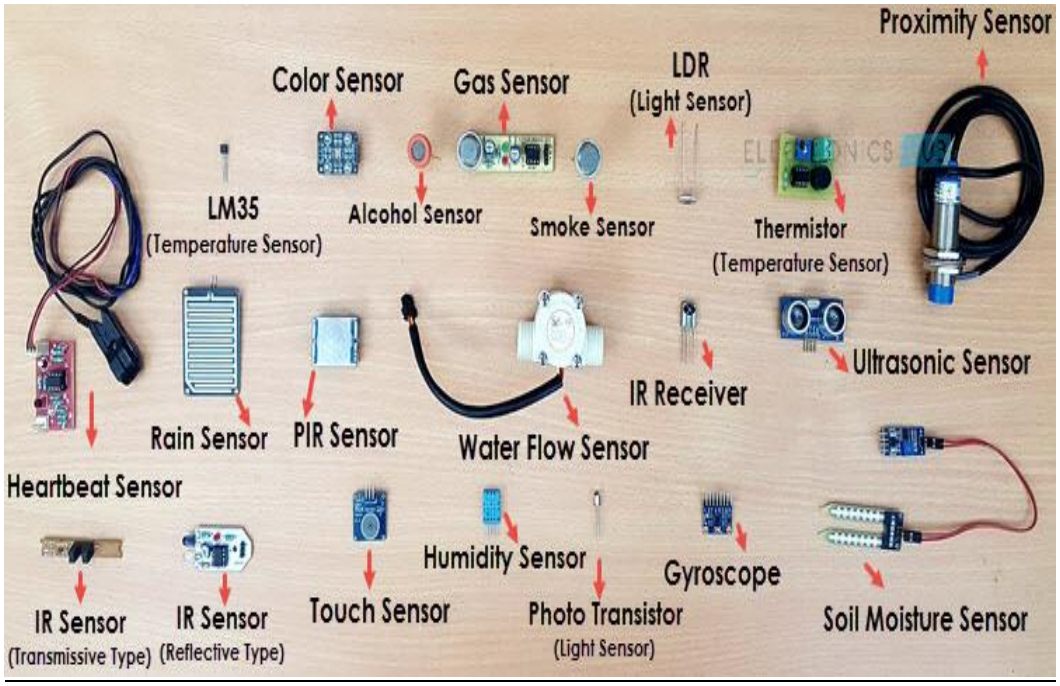
What Raspberry Pi models have been released?

There have been three generations of Raspberry Pis: Pi 1, Pi 2, and Pi 3, and there has generally been a Model A and a Model B of most generations. Model A is a cheaper variant and tends to have reduced RAM and ports like USB and Ethernet. The Pi Zero is a spinoff of the original (Pi 1) generation, made even smaller and cheaper.

IoT Sensor Types

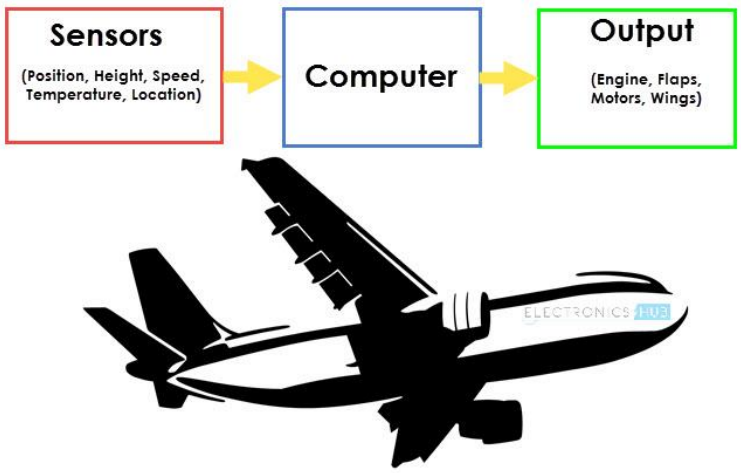
Sensors are everywhere. They're in our homes and workplaces, our shopping centers and hospitals. They're embedded in smart phones and an integral part of the Internet of Things (IoT). Sensors have been around for a long time. The first thermostat was introduced in the late 1880s and infrared sensors have been around since the late 1940s. The IoT and its counterpart, the Industrial Internet of Things (IIoT), are bringing sensor usage to a new level.

Broadly speaking, sensors are devices that detect and respond to changes in an environment. Inputs can come from a variety of sources such as light, temperature, motion and pressure. Sensors output valuable information and if they are connected to a network, they can share data with other connected devices and management systems. Sensors come in many shapes and sizes. Some are purpose-built containing many built-in individual sensors, allowing you to monitor and measure many sources of data.



Real Time Application of Sensors

The example we are talking about here is the Autopilot System in aircrafts. Almost all civilian and military aircrafts have the feature of Automatic Flight Control system or sometimes called as Autopilot.



An Automatic Flight Control System consists of several sensors for various tasks like speed control, height, position, doors, obstacle, fuel, maneuvering and many more. A Computer takes data from all these sensors and processes them by comparing them with pre-designed values.

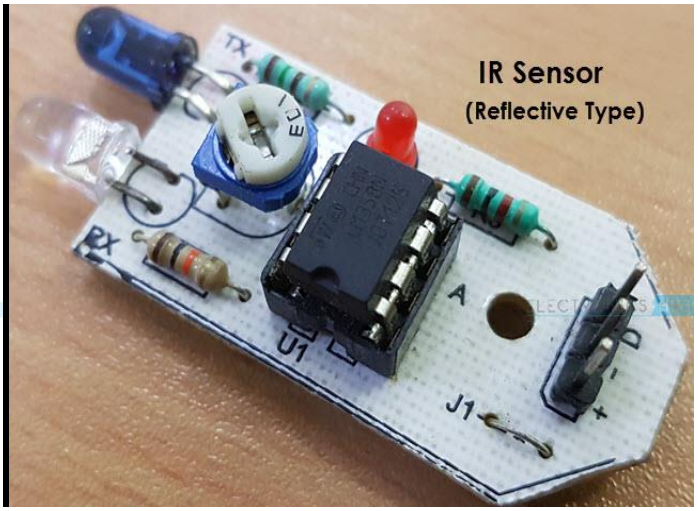
The computer then provides control signal to different parts like engines, flaps, rudders etc. that help in a smooth flight. The combination of Sensors, Computers and Mechanics makes it possible to run the plane in Autopilot Mode.

All the parameters i.e. the Sensors (which give inputs to the Computers), the Computers (the brains of the system) and the mechanics (the outputs of the system like engines and motors) are equally important in building a successful automated system.

Sensors are crucial to the operation of many of today's businesses. They can warn you of potential problems before they become big problems, allowing businesses to perform predictive maintenance and avoid costly downtime. The data from sensors can also be analyzed for trends allowing business owners to gain insight into crucial trends and make informed evidence-based decisions.

How does IoT sensor work?

In IoT applications, sensors are connected to a network (WiFi, LPWAN, cellular, etc.) over which the collected data is transmitted. The destination is usually a cloud-based service where the data is processed.



Temperature Sensors

Temperature sensors measure the amount of heat energy in a source, allowing them to detect temperature changes and convert these changes to data. Machinery used in manufacturing often requires environmental and device temperatures to be at specific levels. Similarly, within agriculture, soil temperature is a key factor for crop growth.

Humidity Sensors

These types of sensors measure the amount of water vapor in the atmosphere of air or other gases. Humidity sensors are commonly found in heating, vents and air conditioning (HVAC) systems in both industrial and residential domains. They can be found in many other areas including hospitals, and meteorology stations to report and predict weather

Pressure Sensors

A pressure sensor senses changes in gases and liquids. When the pressure changes, the sensor detects these changes, and communicates them to connected systems. Common use cases include leak testing which can be a result of decay. Pressure sensors are also useful in the manufacturing of water systems as it is easy to detect fluctuations or drops in pressure.

Proximity Sensors

Proximity sensors are used for non-contact detection of objects near the sensor. These types of sensors often emit electromagnetic fields or beams of radiation such as infrared. Proximity sensors have some interesting use cases. In retail, a proximity sensor can detect the motion between a customer and a product in which he or she is interested. The user can be notified of any discounts or special offers of products located near the sensor. Proximity sensors are also used in the parking lots of malls, stadiums and airports to indicate parking availability. They can also be used on the assembly lines of chemical, food and many other types of industries.

Some of the applications of Proximity Sensors are Mobile Phones, Cars (Parking Sensors), industries (object alignment), Ground Proximity in Aircrafts, etc. Proximity Sensor in Reverse Parking is implemented in this Project: REVERSE PARKING SENSOR CIRCUIT.

Level Sensors

Level sensors are used to detect the level of substances including liquids, powders and granular materials. Many industries including oil manufacturing, water treatment and beverage and food manufacturing factories use level sensors. Waste management systems provide a common use case as level sensors can detect the level of waste in a garbage can or dumpster.

Accelerometers

Accelerometers detect an object's acceleration i.e. the rate of change of the object's velocity with respect to time. Accelerometers can also detect changes to gravity. Use cases for accelerometers include smart pedometers and monitoring driving fleets. They can also be used as anti-theft protection alerting the system if an object that should be stationary is moved.

Gyroscope

Gyroscope sensors measure the angular rate or velocity, often defined as a measurement of speed and rotation around an axis. Use cases include automotive, such as car navigation and electronic stability control (anti-skid) systems. Additional use cases include motion sensing for video games, and camera-shake detection systems.

Gas Sensors

These types of sensors monitor and detect changes in air quality, including the presence of toxic, combustible or hazardous gasses. Industries using gas sensors include mining, oil and gas, chemical research and manufacturing. A common consumer use case is the familiar carbon dioxide detectors used in many homes.

Infrared Sensors

These types of sensors sense characteristics in their surroundings by either emitting or detecting infrared radiation. They can also measure the heat emitted by objects. Infrared sensors are used in a variety of different IoT projects including healthcare as they simplify the monitoring of blood flow and blood pressure. Televisions use infrared sensors to interpret the signals sent from a remote control. Another interesting application is that of art historians using infrared sensors to see hidden layers in paintings to help determine whether a work of art is original or fake or has been altered by a restoration process.

Optical Sensors

Optical sensors convert rays of light into electrical signals. There are many applications and use cases for optical sensors. In the auto industry, vehicles use optical sensors to recognize signs, obstacles, and other things that a driver would notice when driving or parking. Optical sensors play a big role in the development of driverless cars. Optical sensors are very common in smart phones. For example, ambient light sensors can extend battery life. Optical sensors are also used in the biomedical field including breath analysis and heart-rate monitors.

MYTHINGS IoT Sensor

The MYTHINGS Smart Sensor is a self-contained, battery-powered multi-purpose IoT sensor that allows you to capture critical data points like acceleration, temperature, humidity, pressure and GPS. The smart sensor is integrated with the MYTHINGS Library – a hardware independent, small-footprint and power-optimized library of code, featuring the MIOTY (TS-UNB) low-power wide area network protocol

---

Open Source IoT Protocol

- **Advanced Message Queuing Protocol (AMQP)**

Open standard for business messaging Internet protocol. It communicates between applications or companies seamlessly connecting systems, feeds business processes with the information and transmits instructions to attain objectives in a reliable manner. AMQP connects different aspects of Organizations, technologies, systems not available simultaneously, as well as operate at a distance in case of poor network.

- **Constrained Application Protocol (CoAP)**

https://coap.technology/

The Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained networks in the Internet of Things. The protocol is designed for machine-to-machine (M2M) applications such as smart energy and building automation.

CoAP is specified in a Standards-Track RFC. RFCs that serve as Internet Standards Documents are generated by the IETF based on an extensive technical review and quality control process. CoAP is simple enough to implement from scratch for a simple application.

- **Very Simple Control Protocol (VSCP)**

https://www.vscp.org/

While the term protocol may sound misleading, VSCP is a framework. It is a scalable, free and open solution framework for the discovery and identification of devices, configuration, autonomous device functionality, securely updating the devices — overall, a solution from the sensor to the user.

The word “Protocol” may be misleading. VSCP is much more and should probably be called a framework instead. VSCP is a scalable, a very low footprint, a free and open solution for device discovery and identification, device configuration, autonomous device functionality, secure update of device firmware. VSCP is an application level protocol making things interact using CAN, RS-232, Ethernet, TCP/IP, MQTT, 6LowPan.

- **MQTT**

(MQ Telemetry Transport or **Message Queuing Telemetry Transport**) is an open OASIS and ISO standard (ISO/IEC 20922) lightweight, publish-subscribe network protocol that transports messages between devices.)

It the standard messaging and data exchange protocol for the Internet of Things (IoT). The MQTT protocol provides a scalable and cost-efficient way to connect devices over the Internet. You can use MQTT to deliver data over the Internet in near real-time with predefined guarantees of delivery. Connecting millions of IoT devices to your business infrastructure, sending instant updates, and moving data efficiently is where MQTT truly excels.

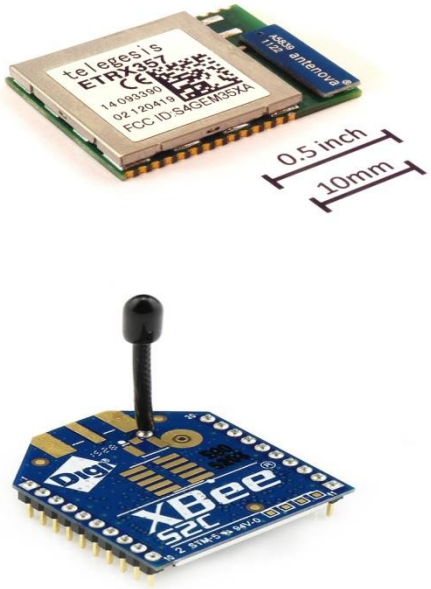
Why MQTT is used in IOT?

MQTT ensures that messages go to the correct devices during communication by utilizing topics. A topic functions the same as a file path would, and directs communication by filtering messages according to elements specified in the topic function.

MQTT is a publish/subscribe protocol that allows edge-of-network devices to publish to a broker. Clients connect to this broker, which then mediates communication between the two devices. ... When another client publishes a message on a subscribed topic, the broker forwards the message to any client that has subscribed .

- **Zigbee**

Zigbee is a wireless technology developed as an open global standard to address the unique needs of low-cost, low-power wireless IoT networks. The Zigbee standard operates on the IEEE 802.15. 4 physical radio specification and operates in unlicensed bands including 2.4 GHz, 900 MHz and 868 MHz.



Zigbee XBee Module S2C 802.15.4 2mW with Wire Antenna XB24CZ7WIT-004

Why ZigBee is better than WiFi?

ZigBee's data transfer speed is lower than WiFi's, too. It's maximum speed is just 250kbps, much lower than the lowest speed WiFi offers. ZigBee's best quality is its low power-consumption rate and battery life.

Can ZigBee connect to WiFi?

ZigBee and WiFi channels both exist in the 2.4 GHz band, existing in the exact same frequency space. When deploying both WiFi and ZigBee in the same environments, careful planning must be performed to make sure that they don't interfere with each other.

Zigbee Applications

Zigbee enables broad-based deployment of wireless networks with low-cost, low-power solutions. It provides the ability to run for years on inexpensive batteries for a host of monitoring and control applications. Smart energy/smart grid, AMR (Automatic Meter Reading), lighting controls, building automation systems, tank monitoring, HVAC control, medical devices and fleet applications are just some of the many spaces where Zigbee technology is making significant advancements.

- **6LoWPAN**

6LoWPAN is an acronym of IPv6 over Low -Power Wireless Personal Area Networks.

6LowPAN. A key IP (Internet Protocol)-based technology is 6LowPAN (IPv6 Low-power wireless Personal Area Network). Rather than being an IoT application protocols technology like Bluetooth or ZigBee, 6LowPAN is a network protocol that defines encapsulation and header compression mechanisms.

- **WiFi**

Wi-Fi is the name of a wireless networking technology that uses radio waves to provide wireless high-speed Internet and network connections. To connect to a Wi-Fi LAN, a computer must be equipped with a wireless network interface controller.

A common misconception is that the term Wi-Fi is short for "wireless fidelity," however this is not the case. Wi-Fi is simply a trademarked phrase that means IEEE 802.11x.

Essential Python Libraries for IoT Devices Coding

And the Most Effective way of learning it and Coding the Devices is in Python Language. Python is the most popular language today with usage exceeding close to 35%. It is Easy learnability, portability, a huge set of developer community support, a large set of library and packages availability, and performance of maths functions are some of the aspects making the language popular.Like many other applications, Python can also be used for IoT application development. Some of the packages you could use for Python are ;

mraa : mraa is a skeleton GPIO library for most SBCs which support Python. As there is just a library for all boards, it is easy to use. Library also provides support for communication protocols such as I2C, UART, and SPI.

What is the use of GPIO pins?

GPIO stands for General Purpose Input/Output. It's a standard interface **used** to connect microcontrollers to other electronic devices. For example, it can be **used** with sensors, diodes, displays, and System-on-Chip modules.

UART:

UART stands for Universal Asynchronous Receiver/Transmitter. It's not a communication protocol like SPI and I2C, but a physical circuit in a microcontroller, or a stand-alone IC. A UART's main purpose is to transmit and receive serial data.

Beacons:

Beacons are small, wireless transmitters that use low-energy Bluetooth technology to send signals to other smart devices nearby. They are one of the latest developments in location technology and proximity marketing.

iBeacon

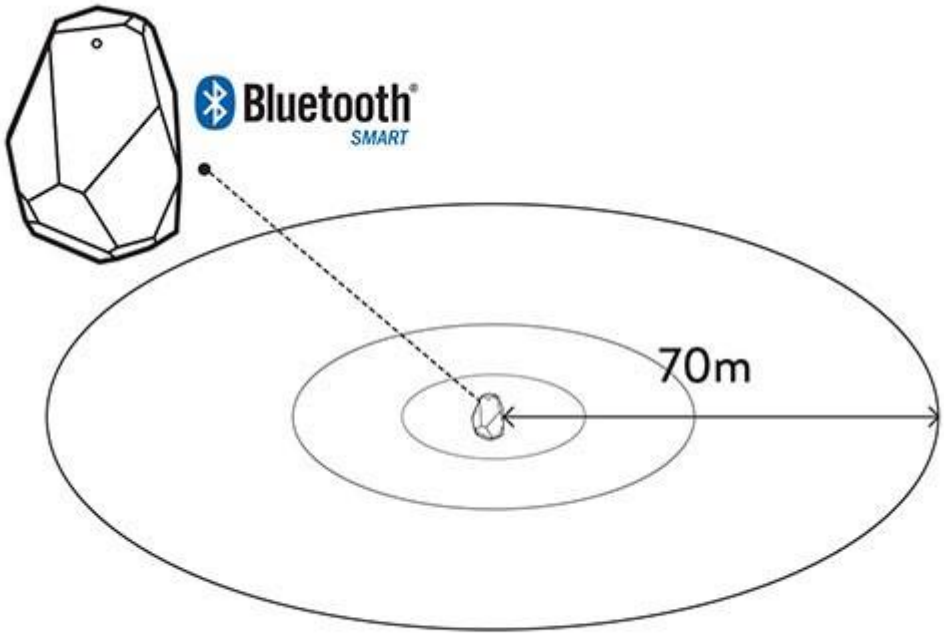
iBeacon is the name for Apple's technology standard, which allows Mobile Apps (running on both iOS and Android devices) to listen for signals from beacons in the physical world and react accordingly.

Transmit RF signals, usually Bluetooth Low Power (BLE), RFID or Wi Fi, to provide push information, location based services, or product information.

Beacons transmit in 4 ranges:

- - Immediate=inches
- - Near=3 to 8 feet
- - Far=30 to 90 feet
- - Long Range=100 feet to half a mile

New developments in power and chip technology promise to increase the range.

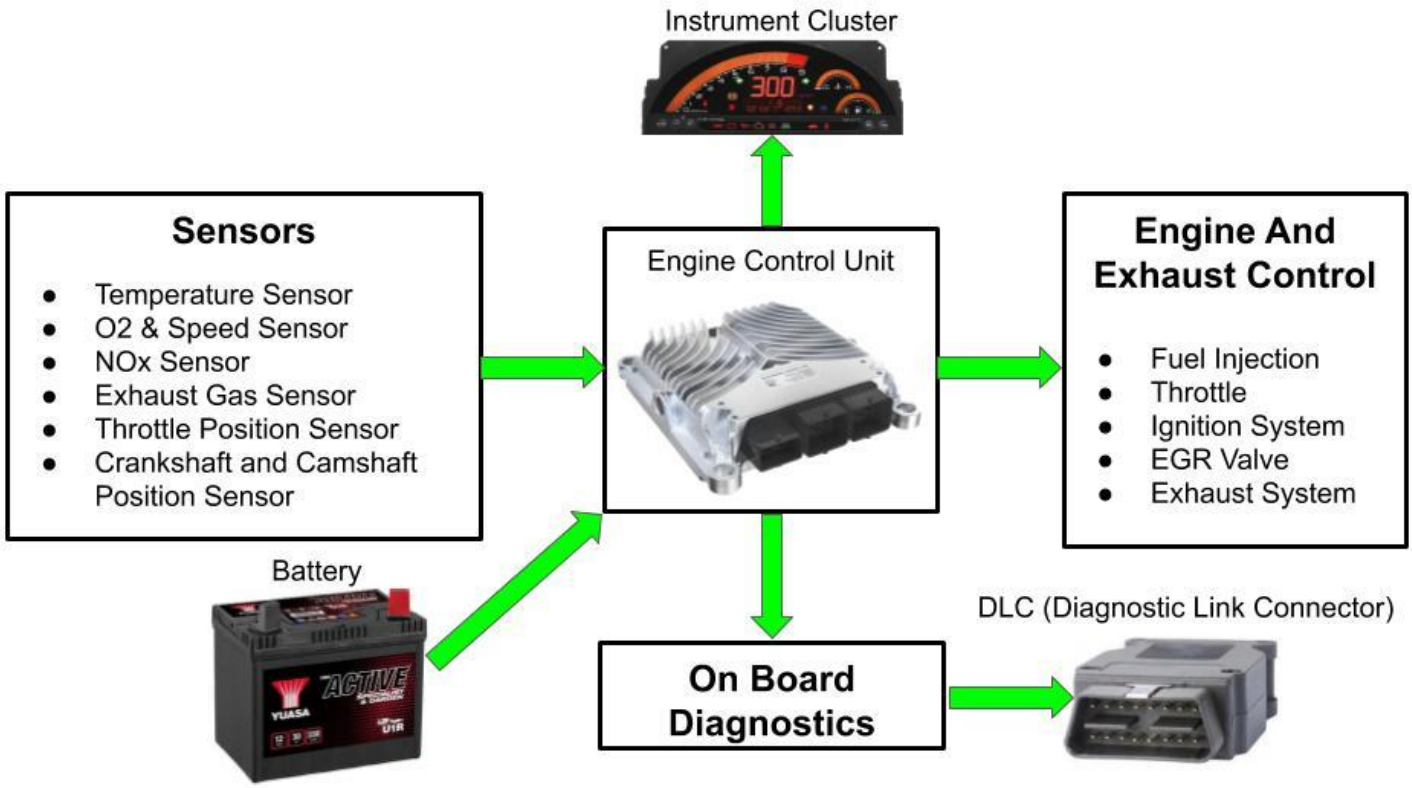

Bluetooth
SMART
70m

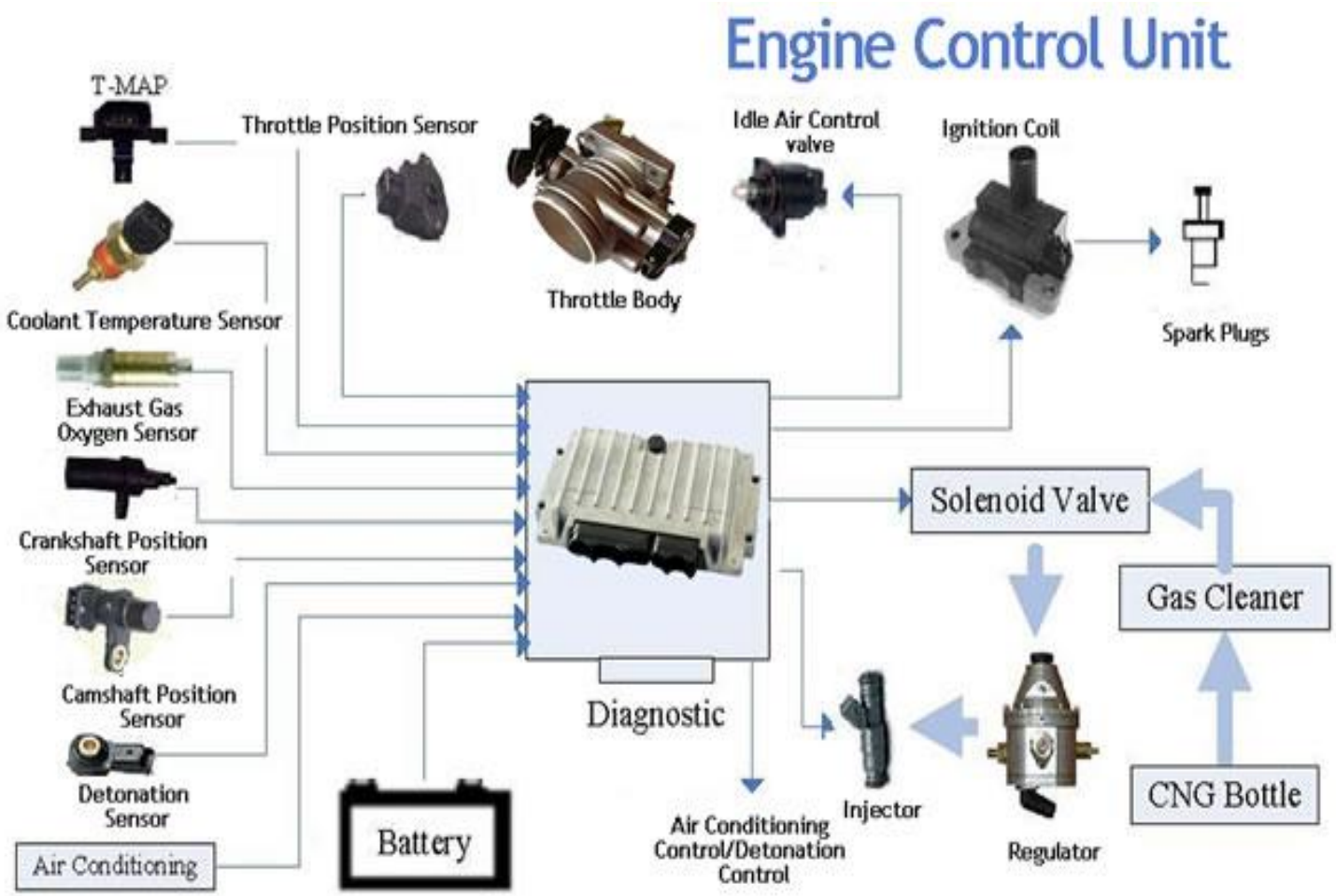
On-Board Diagnostics

OBD (On-Board Diagnostics) is the standard protocol used across most light-duty vehicles to retrieve vehicle diagnostic information. Information is generated by the engine control unit (ECU or also called engine control module) within a vehicle. It's like the vehicle's brain or computer.

What is the use of OBD?

OBD stands for On-Board Diagnostics and is a computer system inside of a vehicle that tracks and regulates a car's performance. The computer system collects information from the network of sensors inside the vehicle, which the system can then use to regulate car systems or alert the user to problems.





sockets: a package that facilitates networking over TCP/IP and UDP using Python. It provides access to Berkeley socket APIs to access the Internet

mysqldb: MySQL is the go-to relational database for most IoT developers. mysqldb is a very convenient tool to circumvents the need to execute shell commands within a Python script to read and write to a database

numpy: very similar to MatLab, numpy provides scientific computing using Python. Very easy to do array processing functions using numpy

matplotlib : The Package for data visualization

pandas : A package dedicated towards data analysis. Provides support for data handling and analysis, direct operations on local datasets and the ability to handle heterogeneous and unordered data.

opencv: Opencv is a Python port of the very successful C library for image processing. It contains high-level variants of familiar image processing functions which make photo analysis much easier

tkinter: a GUI development library. Python script can be controlled via a completely ad hoc GUI. This is extremely useful in situations such as functionality testing or repeated executions of the same code

paho-mqtt: MQTT is commonly used protocol for the Internet of Things. paho-mqtt library gives a very user-friendly version of the protocol for use with embedded systems.

List of IoT communication protocols:

- **CoAP (Constrained Application Protocol)-**Constrained Application Protocol (CoAP) is an Internet application protocol for constrained devices (defined in RFC 7228). It enables constrained devices to communicate with the wider Internet using similar protocols. CoAP is designed for use between devices on the same constrained network, between devices and general nodes on the Internet, and between devices on different constrained networks joined by the Internet. It is an application layer protocol designed for network constrained IoT devices like wireless sensor network nodes, and is often termed the lightweight version of HTTP with support for REST APIs. It can run on most devices that support UDP or a UDP analogue. It implements the REST architectural style which can be transparently mapped to HTTP. However, CoAP also provides features that go beyond HTTP such as native push notifications and group communication. While a usual HTTP header can be around 100 bytes, a CoAP standard header can be as light as just 4 bytes. Unlike MQTT, CoAP doesn't require a broker server to function.

- **Bluetooth and Bluetooth Low Energy-**While MQTT and CoAP are infrastructure-independent, which means that it doesn't matter whether you're connected to a wired or a wireless network, Bluetooth provides only wireless communication over radio frequency (2.4GHz spectrum in the ISM band) using an industry standard that was initially used to share files between mobile phones and is now powerful enough to play music (Advanced Audio Distribution Profile/A2DP), stream data, or build your next IoT device. Bluetooth, generally, is divided into three categories.
  - **Bluetooth Classic:** This is meant for high data rate applications like streaming audio wirelessly.
  - **Bluetooth Smart or Low Energy/BLE:** This is meant for low powered battery-operated devices that stream low packets of data.
  - **Bluetooth SmartReady:** These are essentially the 'hub' devices such as computers, smartphones, etc. They support both the 'classic' and 'smart' devices.

Bluetooth is a sophisticated *ad hoc* networking protocol, and is now especially designed from the ground up for IoT. It provides a stable connection and communication channel, which is extremely low profile and low powered. An obvious example is fitness trackers, which even though powered on throughout the day, can last for months on a single charge or run on a coin cell battery, all thanks to BLE (Bluetooth Low Energy). Bluetooth Classic has fixed profiles like UART over Bluetooth class and A2DP class for audio streaming. On the other hand, Bluetooth Low Energy provides GATT or Generic Attribute Profile, which allows users to define their own profile using Bluetooth, like in the case of a heart rate monitor. BLE is extremely flexible and useful in the IoT space. Bluetooth 5.0 is already out and is maturing, offering more range, more data rates and double the transmission speeds.

Message Communication Protocols for connected devices:

- **MQTT (Message Queue Telemetry Transport)-**It was created about 15 years back for monitoring remote sensor nodes, and is designed to conserve both power and memory. It is based on the 'Publish Subscribe' communication model, where a broker is responsible for relaying messages to MQTT clients. This allows multiple clients to post messages and receive updates on different topics from a central server known as the MQTT broker. This is similar to subscribing to a YouTube channel, where you get notified whenever a new video is posted.

Using MQTT, a connected device can subscribe to any number of topics hosted by an MQTT broker. Whenever a different device publishes data on any of those topics, the server sends out a message to all connected subscribers of those topics, alerting them to the new available data. It is overall a lightweight protocol that runs on embedded devices and mobile platforms, while connecting to highly scalable enterprise and Web servers over wired or wireless networks. It is useful for connections with remote embedded systems, where a small code footprint is required and/or network bandwidth is at a premium or connectivity is unpredictable. It is also ideal for mobile applications that require a small size, low power usage, minimized data packets, and efficient distribution of information to one or many receivers. It is an ISO standard (ISO/IEC PRF 20922) protocol. The good performance and reliability of MQTT

is demonstrated by Facebook Messenger, Amazon IoT (AWS-IoT), IBM Node-Red, etc—organisations that are using it to serve millions of people daily.
An MQTT-SN or MQTT sensor network allows you to use MQTT over a wireless sensor network, which is not generally a TCP/IP based model. The MQTT broker can be run locally or deployed on the cloud. It is further enhanced with features like user name/password authentication, encryption using Transport Layer Security (TLS) and Quality of Service (QoS).

SOAP-OAP (originally Simple Object Access Protocol) is a protocol specification for exchanging structured information in the implementation of web services in computer networks. Its purpose is to induce extensibility, neutrality and independence. It uses XML Information Set for its message format, and relies on application layer protocols, most often Hypertext Transfer Protocol (HTTP) or Simple Mail Transfer Protocol (SMTP), for message negotiation and transmission.
SOAP allows processes running on disparate operating systems (such as Windows and Linux) to communicate using Extensible Markup Language (XML). Since Web protocols like HTTP are installed and running on all operating systems, SOAP allows clients to invoke web services and receive responses independent of language and platforms. SOAP provides the Messaging Protocol layer of a web services protocol stack for web services. It is an XML-based protocol consisting of three parts:

- an envelope, which defines the message structure and how to process it
- a set of encoding rules for expressing instances of application-defined datatypes
- a convention for representing procedure calls and responses

SOAP has three major characteristics:

- extensibility (security and Web Services Addressing are among the extensions under development)
- neutrality (SOAP can operate over any protocol such as HTTP, SMTP, TCP, UDP, or Java Message Service)
- independence (SOAP allows for any programming model)

As an example of what SOAP procedures can do, an application can send a SOAP request to a server that has web services enabled—such as a real-estate price database—with the parameters for a search. The server then returns a SOAP response (an XML-formatted document with the resulting data), e.g., prices, location, features. Since the generated data comes in a standardized machine-parsable format, the requesting application can then integrate it directly.
The SOAP architecture consists of several layers of specifications for:

- message format
- Message Exchange Patterns (MEP)
- underlying transport protocol bindings
- message processing models
- protocol extensibility

REST-REST (Representational State Transfer) is an architectural style for developing web services. REST is popular due to its simplicity and the fact that it builds upon existing systems and features of the internet's HTTP in order to achieve its objectives, as opposed to creating new standards, frameworks and technologies.

Advantages of REST-A primary benefit of using REST, both from a client and server's perspective, is REST-based interactions happen using constructs that are familiar to anyone who is accustomed to using the internet's Hypertext Transfer Protocol (HTTP).
An example of this arrangement is REST-based interactions all communicate their status using standard HTTP status codes. So, a 404 means a requested resource wasn't found; a 401 code means the request wasn't authorized; a 200 code means everything is OK; and a 500 means there was an unrecoverable application error on the server. Similarly, details such as encryption and data transport integrity are solved not by adding new frameworks or technologies, but instead by relying on well-known Secure Sockets Layer (SSL) encryption and Transport Layer Security (TLS). So, the entire REST architecture is built upon concepts with which most developers are already familiar.

Disadvantages of REST-The benefit of REST using HTTP constructs also creates restrictions, however. Many of the limitations of HTTP likewise turn into shortcomings of the REST architectural style. For example, HTTP does not store state-based information between request-response cycles, which means REST-based applications must be stateless and any state management tasks must be performed by the client.

HTTP Restful-In REST architecture, a REST Server simply provides access to resources and the REST client accesses and presents the resources. Here each resource is identified by URIs/ Global IDs. REST uses various representations to represent a resource like Text, JSON and XML. JSON is now the most popular format being used in Web Services.
HTTP Methods
The following HTTP methods are most commonly used in REST based architecture.

- GET − Provides a read only access to a resource.
- PUT − Used to create a new resource.
- DELETE − Used to remove a resource.
- POST − Used to update an existing resource or create a new resource.
- OPTIONS − Used to get the supported operations on a resource.

Restful Web Services-A web service is a collection of open protocols and standards used for exchanging data between applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer. This interoperability (e.g., between Java and Python, or Windows and Linux applications) is due to the use of open standards. Web services based on REST Architecture are known as Restful Web Services. These web services use HTTP methods to implement the concept of REST architecture. A Restful web service usually defines a URI (Uniform Resource Identifier), which is a service that provides resource representation such as JSON and a set of HTTP Methods.

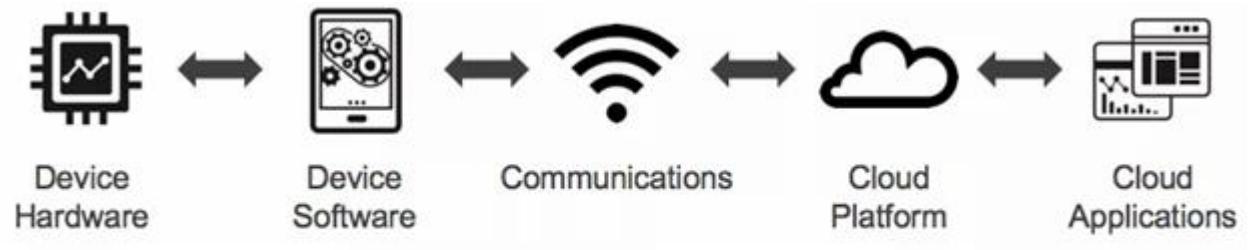
Unit -3

Solution framework for IoT applications-

The IoT decision framework provides a structured approach to create a powerful IoT product strategy. The IoT decision framework is all about the strategic decision making. The IoT Decision Framework helps us to understand the areas where we need to make decisions and ensures consistency across all of our strategic business decision, technical and more.

The IoT decision framework is much more important as the product or services communicates over networks goes through five different layers of complexity of technology.

1. Device Hardware
2. Device Software
3. Communications
4. Cloud Platform
5. Cloud Application



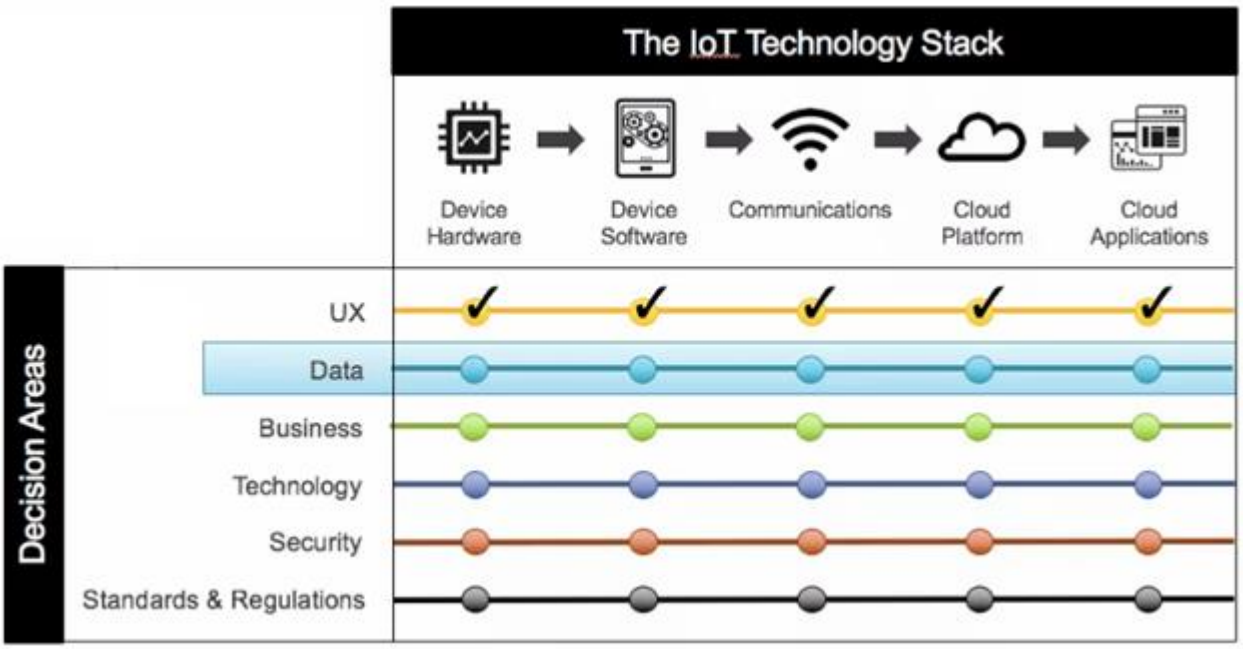
The IoT Technology Stack

Decision Area

The IoT decision framework pays attention to six key decision areas in any IoT product. These decision areas are:

1. User Experience (UX)
2. Data
3. Business
4. Technology
5. Security
6. Standards & Regulations

Each of these decision areas is evaluated at each of the IoT Technology Stack. The User Experience will be evaluated at Device Hardware, Device Software and so to provide the better user experience. Then at the next step Data Decision Area, we have to explore data considerations for all the stages of IoT Technology Stack.



Decision Area of the IoT Decision Framework

Let's see each of the Decision Area of IoT Decision Framework in detail:

1. **User Experience Decision Area:** This is the area where we concentrate about who are the users, what are their requirements and how to provide a great experience at each step of IoT stack without worrying about the technical details.
2. **Data Decision Area:** In this area, we make the overall data strategy such as the data flow over the entire IoT stack to fulfill the user's requirements.
3. **Business Decision Area:** Based on the previous decisions area, we make the decision how product or services will became financial potential. At each of the IoT Stack level are monetized about the costs of providing services.
4. **Technology Decision Area:** In this area, we work with the technology for each layer to facilitate the final solution.
5. **Security Decision Area:** After going through the implementation of technology it is important to decide and provide the security at each stage of the IoT Stack.
6. **Standards & Regulations Decision Area:** At the last stage of IoT Decision Area, we identify the standards and regulations of product or services that will affect your product at each layer of the IoT Stack.

IoT Data Acquisition System

When talking about Internet of Things (IoT), Data acquisition (DAQ) and protocols are pivotal building blocks of IoT technology. A data acquisition device helps users to make machines smarter by gathering and analyzing real-time data. IoT protocols enable it to exchange data in an organized and significant manner. IoT protocols are languages that enable interaction between sensors, devices, gateways, servers, and user applications.

IoT Integration

Integration means making independently designed applications and data work well together. IoT integration means making the mix of new IoT devices, IoT data, IoT platforms and IoT applications — combined with IT assets (business applications, legacy data, mobile, and SaaS) — work well together in the context of implementing end-to-end IoT business solutions. The IoT integration market is defined as the set of IoT integration capabilities that IoT project implementers need to successfully integrate end-to-end IoT business solutions.

Unit 4

Unstructured data storage on cloud/local server

Unstructured data is seeing exponential growth with the rise of technology solutions, eCommerce, businesses moving to the cloud, and social media. This massive growth also means that storage for the data has to be handled well. Just because it is unstructured doesn't mean that it is not practical. In fact, with the right tools, such data is a goldmine of useful information. In this article, we take a closer look at unstructured data as a whole and its relation to cloud-based storage.

What is Unstructured Data?

Unstructured data is essentially all data that doesn't fall under the purview of relational databases (RDBMS). Unstructured data is not structured via predefined data schema or models. However, it has an internal structure - it can be textual or non-textual, or human- or machine-generated, and can be stored within non-relational databases like NoSQL. Examples of unstructured data include text files, email, mobile data, social media, satellite imagery, sensor or surveillance data, communications such as chats, etc.

Unstructured data is something of a misnomer. Sure, some of this data is difficult to analyze or process, but some of the data have additional features such as metadata, making them semi-structured.

Cloud Storage of Unstructured Data

As we have seen, unstructured data can include pretty much all kinds of information. The file sizes can range from anything to a few bits and bytes to gigabytes or more. Hence, there is no one-size-fits-all approach in terms of data storage. The type of storage where the data sits depends on the capacity as well as the set input/output (I/O) requirements. So, anything from low I/O performance (NAS, cloud instance, object storage) to high-performing, massive files (distributed file, object storage).

Network-attached storage (NAS) used to be associated with single file, siloed storage of data. Not anymore. These days, scale-out NAS is able to manage high-performance, high-capacity storage of the data. But again, object storage has also grown over the years and leads in unstructured data storage. Object storage comes with many advantages, such as having unique IDs for the stored data, being high-performance, highly scalable, and easily accessed with APIs. It is no surprise that most cloud providers opt for object storage.

Cloud providers offer high-performance, scalable storage services to customers, and there is a high demand for these flexible services. Some of them come in subscription-based systems or open source, reducing the overall financial burden to enterprises and organizations.

IoT Authentication, authorization of devices

Why Is Device Authentication Necessary for the IoT?

Strong IoT device authentication is required to ensure connected devices on the IoT can be trusted to be what they purport to be. Consequently, each IoT device needs a unique identity that can be authenticated when the device attempts to connect to a gateway or central server. With this unique ID in place, IT system administrators can track each device throughout its lifecycle, communicate securely with it, and prevent it from executing harmful processes. If a device exhibits unexpected behavior, administrators can simply revoke its privileges.

IoT Authentication

IoT (Internet of Things) Authentication refers to ways to securely and conveniently access connected devices such as smart homes, autos, transportation hubs, and workplaces.

The smart device ecosystem is highly fragmented, having not yet settled on a standard for what hardware, software, and communications protocols are the dominant means to access devices. Enterprises may use RFID badges for secure entryways, while homeowners may use proprietary apps for autos and thermostats. This fragmentation causes poor usability — UX being paramount to the success of the IoT and digital transformation — and higher risk, as system fragmentation and varied settings are unsafe. In all, today's IoT security is lightweight compared to enterprise application security and the IoT's aggressive rollout despite this has created a situation where IoT authentication must catch up.

IoT authentication would benefit from a single standard onto which all device makers and solution providers deploy their technology. One solution is to settle on a single user interface (UI) such as consumer mobile devices and to authenticate based on FIDO Alliance open standards for True Keyless Authentication. This would reduce the fragmentation and an over-reliance on passwords, whose use as an authentication mechanism hinders IoT adoption by degrading usability. And, whose security is not in step with the security demands of workplaces, homes, cars, transportation hubs, and critical infrastructure.

Example:

"IoT authentication is important since the security of devices, autos, and workplaces is paramount. The risk of unauthorized management of these smart things is too great to cede that security to passwords, whose poor usability has not kept pace with the IoT just as they haven't kept pace with mobile."

The IoT methodology:

IoT Strategy Execution-This perspective looks at IoT strategy from an enterprise perspective, including IoT strategy definition, IoT opportunity identification, IoT business case and IoT programme management.

IoT Solution Delivery-This perspective looks at thev individual IoT solution and the related project. Note that it defines the interfaces to the related asset and its organization, but usually excludes design and manufacturing of the asset itself.

Requirement and Process

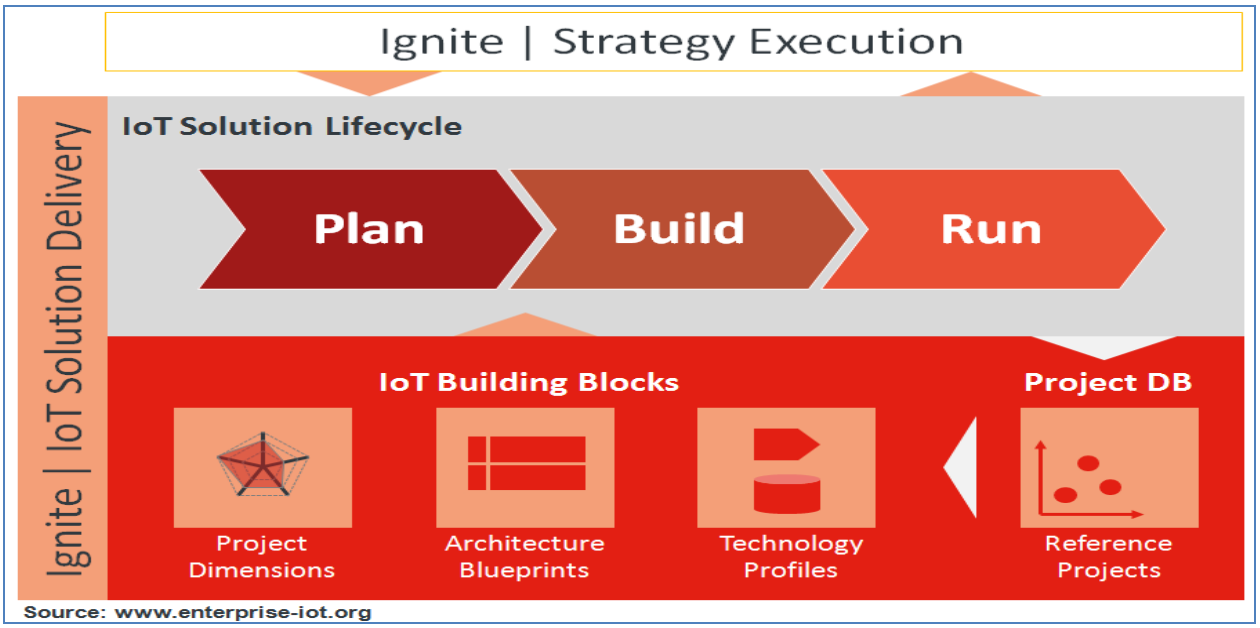


Fig. 5.1 Requirement and Process

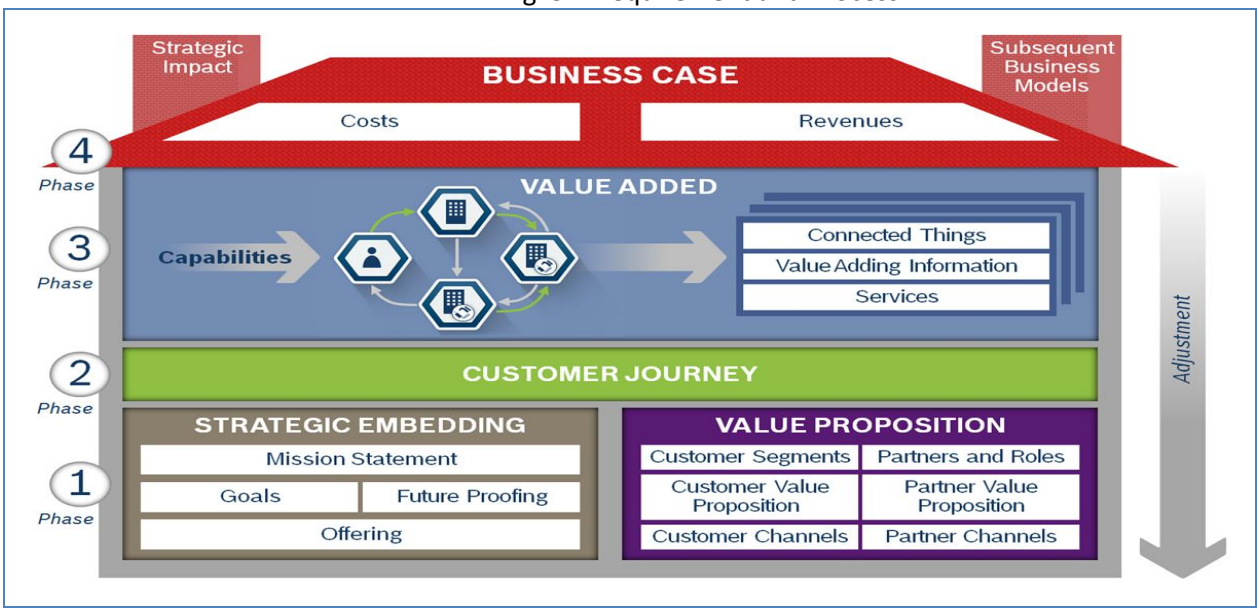


Fig .5.2 Service, Model and Functional View

IOT Privacy and security solutions-The IoT has to protect against attacks from the following categories: authentication, access control, confidentiality, integrity, and availability. Authentication involves the mutual verification of routing peers before they share route information and ensures shared data origin is accurate. In the IoT, authentication has to be strong and highly automated. Access control is the prevention of unauthorized node use, i.e. making sure nodes are not compromised. Confidentiality is the protection of information, especially when shared over a publicly accessible medium such as air for wireless. Integrity involves the protection of data and confirms no unauthorized modifications occur.

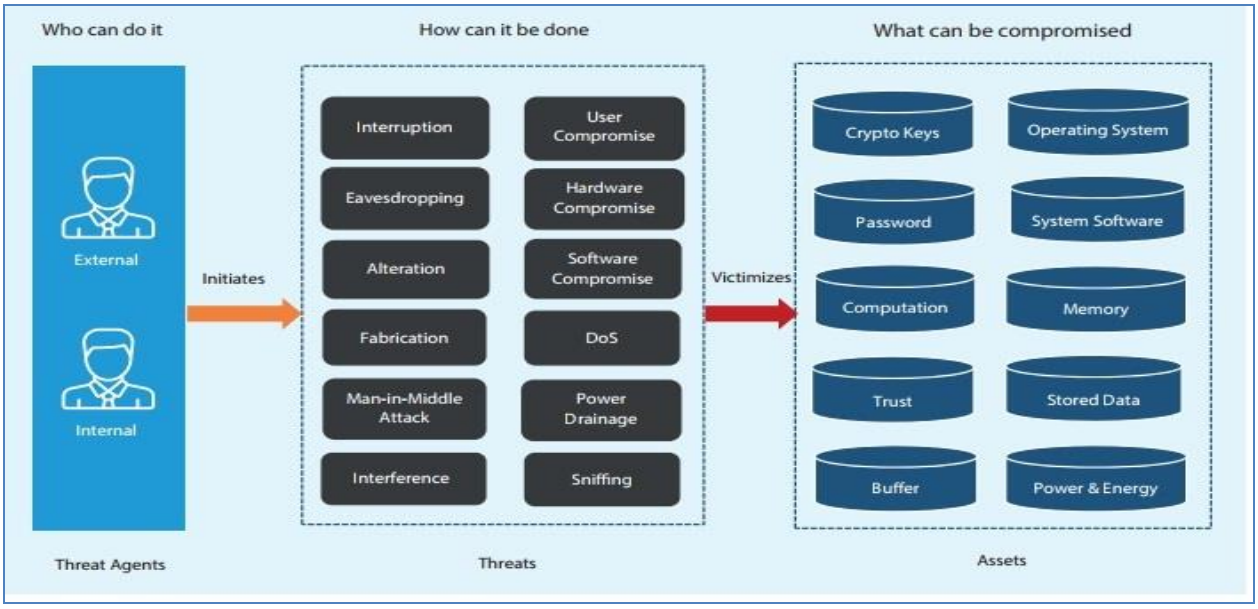


Fig .5.3 Privacy and security Architecture

A major difference between traditional Internet and the IoT is the amount of data being collected about the user. Data is collected universally in the IoT and this data can be used to build an invasive profile of the consumer. The organizations recognized three major privacy concerns: facilitation of the collection of large amounts of consumer data, using that data in ways unexpected by the consumer, and security of data. This ubiquitous data collection makes the Internet of Things a much more data driven economy. With massive quantities of continuous data, new discoveries can be made, but little to no regulation can by harmful to the consumers. Privacy issues are especially hard to discuss because, by nature, privacy is subjective. The organizations aim to promote three best practices: privacy by design, simplified consumer choice, transparency. Companies have to make an effort to build consumer protection in from the beginning.

With such an asymmetry of power between businesses and their consumers, the organizations are looking for ways to protect users against abuse of their data. The IoT, a data-driven ecosystem, requires a trust between the business and consumer that exists even now. A user shares data with a business and in return receives a service. The organizations is seeking to push businesses and companies towards built-in security and designing security into new devices. For the IoT, the data is usually passively and ubiquitously collected. As a result, the organizations believes businesses will have to earn user trust and at a data level, which means involving the user. A similar problem exists in the energy industry. A Green Button was created in order to standardize energy usage information, allow the consumers to download the information, and enlighten the users how their data is being used. Empowering and educating the consumer would help facilitate the integration of the IoT into our everyday lives.

Raspberry Pi-The Raspberry Pi is a low cost, credit-card sized **computer** that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

What's more, the Raspberry Pi has the ability to interact with the outside world, and has been used in a wide array of digital maker projects, from music machines and parent detectors to weather stations and tweeting birdhouses with infra-red cameras. We want to see the Raspberry Pi being used by kids all over the world to learn to program and understand how computers work.

Arduino devices

Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself (DIY) kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.

The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler tool chains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

IOT Case studies: smart city streetlights control & monitoring.

Introduction-Automate street lights are necessary while we are trying to survive in the era of smart world. As automation provides perfection and efficiency. In this paper we are focusing on automated street lighting, as current system is facing many problems. Here we are considering the problems which are done manually. A user has to deal with numerous problems like maintenance problem, timer problem, connectivity problem, display problem. The solution to this problems is IoT Based Street Lights, which allows This template, modified in MS Word 2007 and saved as a "Word 97-2003 Document" for the PC, provides authors with most of the formatting specifications needed for preparing electronic versions of their papers. All standard paper components have been specified for three reasons: (1) ease of use when formatting individual papers, (2) automatic compliance to electronic requirements that facilitate the concurrent or later production of electronic products, and (3) conformity of style throughout conference proceedings. Margins, column widths, line spacing, and type styles are built-in; examples of the type styles are provided throughout this document and are identified in italic type, within parentheses, following the example. Some components, such as multi-leveled equations, graphics, and tables are not prescribed, although the various table text styles are provided. The formatter will need to create these components, incorporating the applicable criteria that follow.
Street lights are one of the main city's assets which provide safe roads, inviting public areas, and enhanced security in homes, businesses, and city centers. As they use in average 40% of a city's electricity spending which leads to power consumption. Following are the issues of existing electric system. Connectivity issue-In existing system, connections of street light are done manually. As each connection requires different contractors and if any one of them is not available then it will leads to functionality problem of street lights. Timer Problem-Contractors needs to manage timer settings manually. As timer requires twelve hour of continuous electricity supply, and if in case it is not available, it will delay further timer settings. Maintenance problem-If any of the streets light gets failed or any problem occurs, it's not resolved immediately. Incorrect Readings-Sometimes exact readings are not shown on to the display. So we cannot conclude how much energy is being consumed which give rise in high billing. Streetlights are among a city's strategic assets providing safe roads, inviting public areas, and enhanced security in homes, businesses, and city centers. However they are usually very costly to operate, and they use in average 40% of a city's electricity spending. As the cost of electricity continues to rise and as wasting energy is a growing concern for public and authorities, it's becoming crucial that municipalities, highway companies and other streetlight owners deploy control systems to dim the lights at the right light level at the right time, to automatically identify lamp and electrical failures and enable real time control. Street Light Monitoring & control is an automated system designed to increase the efficiency and accuracy of an industry by automatically timed controlled switching of street lights. This project describes a new economical solution of street light control systems. The control system consists of wireless technology. Base server can control the whole city's street lights by just sending a notification using network. The main motive behind implementing this project to save energy.

Literature Survey and Current Issues

Energy efficiency using SSL-SSL is nothing but the smart street light system. The SSL system, a framework for fast, reliable, and power efficient street lamp switching based on pedestrians' location and personal desires of safety . In the developed prototype user location, detection as well as safety zone definition and announcement of other configuration information is accomplished using standard Smartphone capabilities. An application on the phone is periodically sending location and other information to the SSL server. For street lamp control, each and every lamppost is extended with a ZigBee-based radio device, receiving control information from the SSL server via multi-hop routing.

Embedded Platform for IoT applications-For embedded platforms, CoAP (Constraint Application protocol) is used for IOT applications. The main idea of this protocol is to provide a lightweight protocol for resource-oriented applications run on constrained networks. For reducing the burdens of manufacturers, we have designed our software framework for embedded system nodes to allow IoT service development with minimal efforts. As this framework supports application-layer API, which does not affect the existing codes and hides network-layer

functions, product manufacturers only need to append a simple CoAP service definition, network driver, and physical network adapter to start IoT services on nodes.

Electrical power saving using VANET-The huge amount of electrical power of many countries is consumed in lighting the streets. However, vehicles pass with very low rate in specific periods of time and parts of the streets are not occupied by vehicles over time. an efficient autonomous street lighting control and monitoring system based on the innovative technology named as Vehicular Ad-Hoc Networks (VANET) is proposed. The system can be integrated with VANET to reduce the cost and use the rich services and communication features of VANET. Huge energy can be saved without affecting the visibility and the safety of the drivers. It can extend the lifetime of the lamps. It can automatically monitor the street lighting equipment's and warn the maintenance traffic authority upon failure detection in any place of the streets.

Fully controlled street lights using Raspberry-pi and Zigbee-The Raspberry-Pi has been chosen for its low costs and for the possibility to drive also a WiMAX modem/router which allows to make the data system visible by a web site accessible by Internet also for areas very far from the city and not reached neither by the ADSL line nor by 3G signals. Intelligent lighting of the lamp, the storage of the functioning data, and their sharing by a local communication wireless mesh realized by ZigBee devices that send information to the coordinator lamp equipped with a RaspberryPi card.

System-Raspberry-Pi is used to provide interface between user and system. It is connected to wireless network and relay circuit which will pass the operational admin's message to the system. Then relay circuit operate the commands like ON Lights, OFF Lights, Alter ON, Alter OFF onto the connected array of street light. Our system includes two admins: System admin and Operational admin. System admin handles log messages and operational admin. System admin can add, delete and view operational admin. Once the operational admin added to the system by the system admin then operational admin can log in to the system.For example, operational admin choose the city and area from database to ON or OFF the street lights. And if any fault occurs in the functioning of street lights then relay circuit will send the faulty street light's IP address to the operational admin then operation admin will resolve the problem.

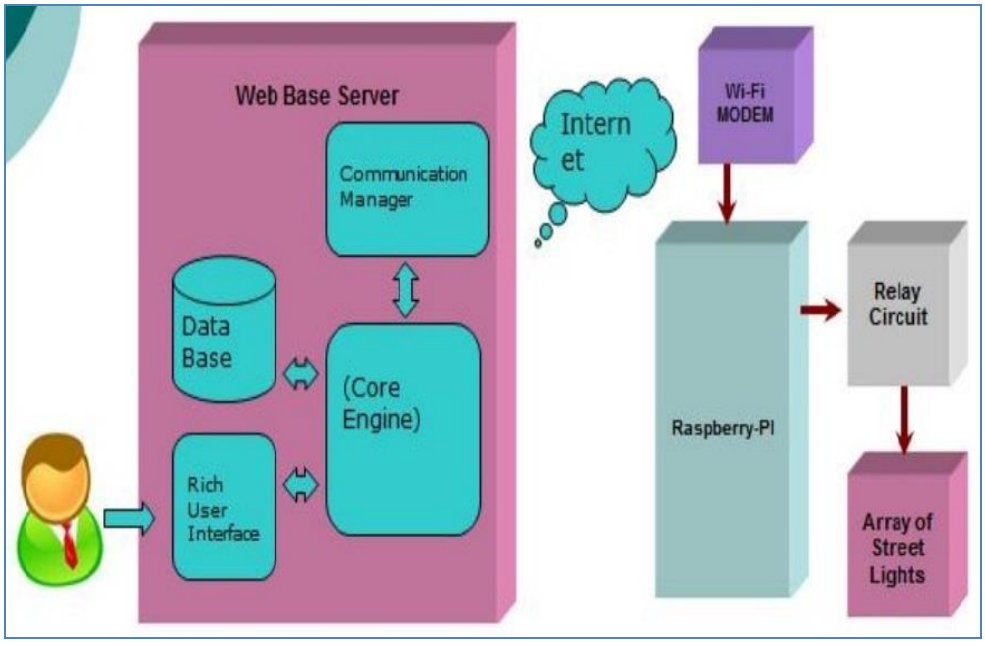


Fig 5.4: System architecture of system

IoT Development Tools

1. Arduino

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. It is both a hardware specification for interactive electronics and a set of software that includes an IDE and the Arduino programming language. The website explains that Arduino is "a tool for making computers than can sense and control more of the physical world than your desktop computer."

Project: A Smart night lamp for kids- Lights up when dark and changes color automatically.

2. Eclipse IoT Project

Have you heard of the Lua programming language yet? Eclipse is sponsoring several different projects surrounding IoT. They include application frameworks and services; open source implementations of IoT protocols and tools for working with Lua, which Eclipse is promoting as an ideal IoT programming language.

Project: The Paho project provides reliable open-source implementations of open and standard messaging protocols aimed at new, existing, and emerging applications for Machine-to-Machine (M2M) and Internet of Things (IoT).

3. Kinoma

Kinoma, a Marvell Semiconductor hardware prototyping platform encompasses three different open source projects. Kimona Create is a DIY construction kit for prototyping electronic devices. Kimona Studio is the development environment that works with Create and the Kinoma Platform Runtime. Kimona Connect is a free iOS and Android app that links smartphones and tables with IoT devices. Github

Sparkle Motion: Create an LED world map driven by global Twitter traffic data.

4. M2MLabs Mainspring

M2MLabs Mainspring is an open source application framework for building machine to machine (M2M) applications such as remote monitoring, fleet management or smart grid.Its capabilities include flexible modeling of devices, device configuration, communication between devices and applications, validation and normalization of data, long-term data storage, and data retrieval functions. It's based on Java and the Apache Cassandra NoSQL database. M2M applications can be prototyped in hours rather than weeks and finally transfered to a high performance execution environment built on top of a standard J2EE server and the highly scaleable Apache Cassandra database.

Tutorial Sample project: This tutorial covers sensor and device modeling, device creation, data retrieval and data display in the M2M platform.

5. Node-RED

A visual tool for wiring the Internet of Things i.e wiring together hardware devices, APIs and online services in new and interesting ways. Built on Node.js, Node-RED describes itself as "a visual tool for wiring the Internet of Things." It allows developers to connect devices, services and APIs together using a browser-based flow editor. It can run on **Raspberry Pi,** and more than 60,000 modules are available to extend its capabilities.

Contribute to the Node-Red IBM project or on node-red github.

Hardware

6. Arduino Yún

Arduino is an open-source electronics platform based on easy-to-use hardware and software. This microcontroller combines the ease of an Arduino-based board with Linux. It includes two processors—the **ATmega32u4** (which supports Arduino) and the **Atheros AR9331 (which runs Linux**). Other features include Wi-Fi, Ethenet support, a USB port, micro-SD card slot, three reset buttons and more.

Project: Ultrasonic Map-Maker using an Arduino Yun- Generates maps based on distance between itself and obstacles autonomously and provides visual feedback.

7. BeagleBoard

BeagleBoard offers **credit-card sized** computers that can run Android and Linux. Because they have very low power requirements, they're a good option for IoT devices. Both the hardware designs and the software they run are open source, and BeagleBoard hardware (often sold under the name **BeagleBone**) is available through a wide variety of distributors. Experiment with Linux, Android and Ubuntu and jump-start development in five minutes with the included USB cable.

Project: **Measuring Temperature with a BeagleBone Black**, learn how to connect temperature sensor to a BeagleBone Black.

8. Flutter

Flutter is a programmable processor core for electronics projects, designed for hobbysits, students, and engineers.Flutter's claim to **fame is its long range**. This Arduino-based board has a wireless transmitter that can reach more than a half mile. Plus, you don't need a router; flutter boards can communicate with each other directly. It includes 256-bit AES encryption, and it's easy to use. Github

9. LightBlue Bean Punch Through

The LightBlue Bean is a low energy Bluetooth Arduino microcontroller. Using Bluetooth 4.0, it is programmed wirelessly, runs on a coin cell battery, and is perfect for smartphone controlled projects. With Bean, you can program wirelessly from any of your devices. No more unscrewing screws and ungluing glue. Github

10. Microduino

Microduino presents the **world's smallest series of Arduino-compatible smart modules that are small**, flexible, stackable and powerful, and can be used to create a limitless amount of DIY projects. Microduino offers really small boards that are compatible with Arduino. Interactive Projects

11. OpenPicus

OpenPicus is an Italian hardware company who designs and produce Internet of Things system on modules called **Flyport**. Flyport is open hardware and the openPicus framework and IDE are open software. Its platform and hardware are open source, but its products can be used to create closed source commercial products. The company also offers its development services for hire.

12. Pinoccio

Arduino-compatible Pinnoccio boards (which the company calls **"Scouts")** connect to each other in a low-power mesh network. They include a built-in rechargeable battery that can connect **to solar panels or any USB power supply**. The organization also offers Pinoccio HQ, a GUI for monitoring the activities of the scouts, and ScoutScript, an easy-to-use scripting language for controlling the devices.

13. RasWIK

Made by a company called Ciseco, RasWIK is short for the Raspberry Pi Wireless Inventors Kit. It allows anyone with a **Raspberry Pi** to experiment with building their own Wi-Fi-connected devices. It includes **documentation for 29 different projects** or you can come up with one of your own. There is a fee for the devices, but all of the included code is open source, and you can use it to build commercial products if you choose.

14. SODAQ

Short for "Solar-Powered Data Acquistion," SADAQ offers Arduino-compatible boards with Lego-like plug-in modules. The website includes a number of tutorials, making it a suitable for beginners. And the **solar panel** makes it a good choice for logging environmental data in various locations where power and Internet connections might not be available.

15. Tessel

Tessel aims to make hardware development easier for software developers with this **JavaScript-enabled** microcontroller that plugs into any USB port. You can also connect it to additional modules to add accelerometer, ambient **light and sound, camera, Bluetooth, GPS and/or** nine other capabilities.

16. UDOO

This Arduino-compatible board can also run Android or Linux (a distribution **called UDOObuntu)** from its second processor. It boasts that it is four times as powerful as a **Raspberry Pi**. Multiple tutorials and projects are available on the website, and it also offers a "Made by UDOOers" section where people can show off their creations.

Home Automation Software

17. OpenHAB

OpenHAB lets the smart devices you already have in your home talk to one another. It's vendor- and hardware-neutral, running on any Java-enabled system. One of its goals is to allow users to add new features to their devices and combine them in new ways. It's won several awards, and it has a companion cloud computing service called **my.openHAB.**

18. The Thing System

This project includes both software components and network protocols. It promises to find all the Internet-connected things in your house and bring them together so that you can control them. It supports a long list of devices, including Nest thermostats, Samsung Smart Air Conditioners, Insteon LED Bulbs, Roku, Google Chromecast, Pebble smartwatches, Goji smart locks and much more. It's written in Node.js and can fit on a Raspberry Pi.

Middleware

19. IoTSyS

This IoT middleware provides a communication stack for smart devices. It supports multiple standards and protocols, including IPv6, oBIX, 6LoWPAN, Constrained Application Protocol and Efficient XML Interchange. Several videos on the website show how it works in action.

20. OpenIoT

The OpenIoT website explains that the project is "an open source middleware for getting information from sensor clouds, without worrying what exact sensors are used." It aims to enable cloud-based "sensing as a service," and has developed use cases for **smart agriculture, intelligent manufacturing, urban crowdsensing, smart living and smart campuses**.

Operating Systems

21. AllJoyn

Originally created by Qualcomm, this open source operating system for the Internet of Things is now sponsored by one of the most prominent IoT organizations—The AllSeen Alliance, whose members include the Linux Foundation, Microsoft, LG, Qualcomm, Sharp, Panasonic, Cisco, Symantec and many others. It includes a framework and a set of services that will allow manufacturers to create compatible devices. It's cross-platform with APIs available for Android, iOS, OS X, Linux and Windows 7.

22. Contiki

Contiki describes itself as "the open source OS for the Internet of Things." It connects low-power microcontrollers to the internet and supports standards like IPv6, 6lowpan, RPL and CoAP. Other key features include highly efficient memory allocation, full IP networking, very low power consumption, dynamic module loading and more. Supported hardware platforms include Redwire Econotags, Zolertia z1 motes, ST Microelectronics development kits and Texas Instruments chips and boards. Paid commercial support is available.

23. Raspbian

While the Raspberry Pi was intended as an educational device, many developers have begun using this credit-card-sized computer for IoT projects. The complete hardware specification is not open source, but much of the software and documentation is. Raspbian is a popular Raspberry Pi operating system that is based on the Debian distribution of Linux.

24. RIOT

RIOT bills itself as "the friendly operating system for the Internet of Things." Forked from the FeuerWhere project, RIOT debuted in 2013. It aims to be both developer- and resource-friendly. It supports multiple architectures, including MSP430, ARM7, Cortex-M0, Cortex-M3, Cortex-M4, and standard x86 PCs.

25. Spark

Spark is a distributed, cloud-based IoT operating system. The same company also offers easy-to-use hardware development kits and related products that start at just $39 (and the hardware designs are also open source). It includes a Web-based IDE, a command-line interface, support for multiple languages, and libraries for working with

many different IoT devices. It has a very active user community, and a lot of documentation and online help are available.

26. Freeboard

Freeboard aims to let users create their own dashboards for monitoring IoT deployments. The code is freely available on GitHub or you can try the service for free if you make your dashboard public. Low-priced plans are also available for those who want to keep their data private. Sample dashboards oon the site show how they can be used to track air quality, residential appliances, distillery performance or environmental conditions in a humidor.

27. Exciting Printer

Exciting offers an open source kit for experimenting with IoT printing. It makes it possible to build your own small printer and use that printer to print out information obtained from various IoT devices. For example, it could print out a list of daily reminders, the weather report, etc. And in a interesting twist, if you want to contact the project owners, you can draw a picture that will be printed on the IoT printer in their office.

Platforms and Integration Tools

28. DeviceHive

This project offers a machine-to-machine (M2M) communication framework for connecting devices to the Internet of Things. It includes easy-to-use Web-based management software for creating networks, applying security rules and monitoring devices. The website offers sample projects built with DeviceHub, and it also has a "playground" section that allows users to use DeviceHub online to see how it works.

29. Devicehub.net

Devicehub.net describes itself as "the open source backbone for the Internet of Things." It's a cloud-based service that stores IoT-related data, provides visualizations of that data and allows users to control IoT devices from a Web page. Developers have used the service to create apps that track health information, monitor the location of children, automate household appliances, track vehicle data, monitor the weather and more.

30. IoT Toolkit

The group behind this project is working on a variety of tools for integrating multiple IoT-related sensor networks and protocols. The primary project is a Smart Object API, but the group is also working on an HTTP-to-CoAP Semantic mappin , an application framework with embedded software agents and more. They also sponsr a meetup group in Silicon Valley for people who are interested in IoT development.

31. Mango

Mango bills itself as "the world's most popular open source Machine-to-Machin (M2M) software." Web-based, it supports multiple platforms. Key features include support for multiple protocols and databases, meta points, user-defined events, import/export and more.

32. Nimbits

Nimbits can store and process a specific type of data—data that has been time- or geo-stamped. A public platform as a service is available, or you can download the software and deploy it on Google App Engine, any J2EE server on Amazon EC2 or on a Raspberry Pi. It supports multiple programming languages, including Arduino, JavaScript, HTML or the Nimbits.io Java library.

33. OpenRemote

OpenRemote offers four different integration tools for home-based hobbyists, integrators, distributors, and manufacturers. It supports dozens of different existing protocols, allowing users to create nearly any kind of smart device they can imagine and control it using any device that supports Java. The platform is open source, but the company also sells a wide variety of support, ebooks and other tools to aid in the design and product development process.

34. SiteWhere

SiteWhere is an open source IoT platform. It provides a system that facilitates the ingestion, storage, processing, and integration of device data. This project provides a complete platform for managing IoT devices, gathering data and integrating that data with external systems. Github

35. ThingSpeak

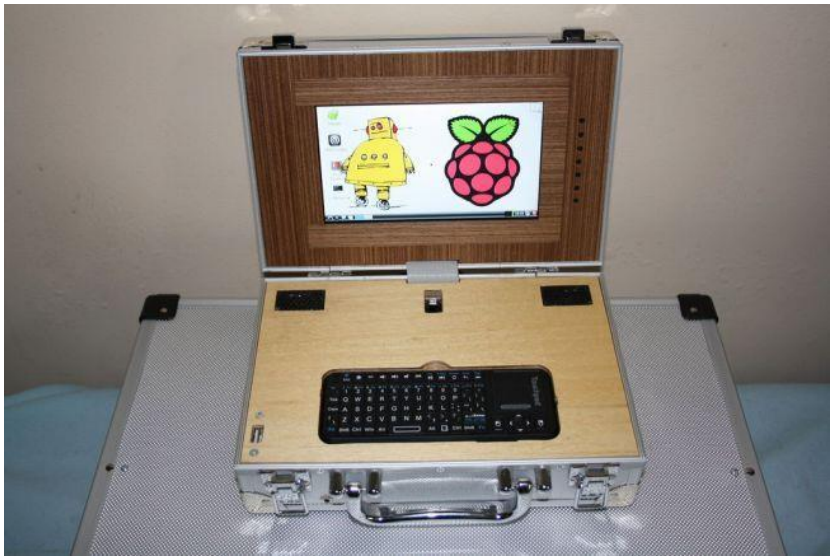
Project - RASPBERRY PI [Mini Projects]

Tech enthusiasts around the world have marvelled at the wonders of the Raspberry Pi, an affordable single-board computer which is the size of a credit card, that can be used in numerous **DIY projects**. It is an extremely flexible technology which has granted users access to a fully-fledged computing system for a few thousand rupees. Right from being a simple gateway for beginners to start coding, to being a tool that experienced, advanced individuals, can utilise to create some impressive projects and programs, the Raspberry Pi is making all of this possible. It is used to create inexpensive machines, robots, appliances, sensors and even a programming toy for children.

HOMEMADE SMARTPHONE

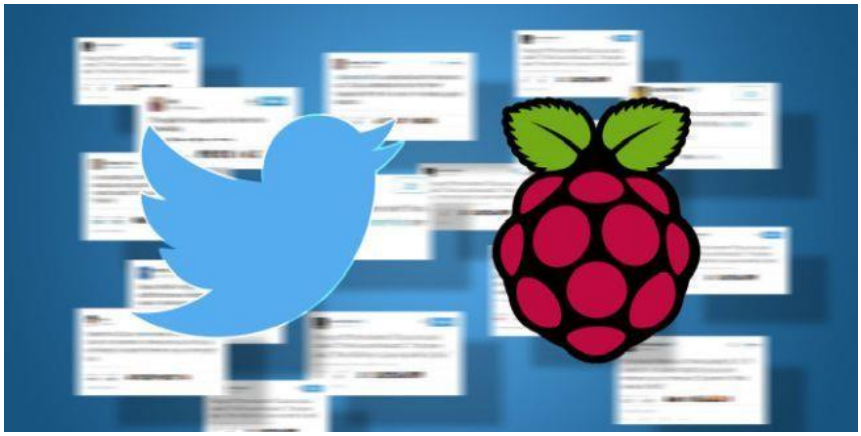
Smartphones are getting obnoxiously similar, both in design and functionalities. So, if you're bored with your smartphone devices, you can put together your very own Pi-powered phone which should be relatively cheap to build. Raspberry Pi can easily be turned into your next phone, albeit it would look a bit weird and nerdy (but we love that, right?), without a lot of tinkering. You will need to be purchase a compatible touchscreen, battery pack and a GSM device. While creating the software for the phone to run on isn't exactly easy, there are a wide variety of good tutorials online that should help you do the same.

NETBOOK



You can make your very own homemade laptop or netbook with Raspberry Pi. One such great example is the Pi-Top which is a creative kit that includes a 14-inch 1080p screen, a full-sized keyboard, and an internal cavity where you can fix your own electronic creations. **LapPi** is another creative project which is essentially a Raspberry Pi-powered laptop that is a lot cheaper than branded alternatives. Additionally, the laptop is actually in the form of a briefcase, so it is extremely portable and looks cool as well.

TWITTER BOT



You can easily turn your Raspberry Pi into a fully-operational Twitter Bot. A Twitter Bot allows users to easily manage their account, receive and send tweets, send messages, and interact with followers. Pi Twitterbot uses Python, a popular programming language, especially for DIY projects. You will have to install **Twython**, a Python module which allows interfacing with Twitter. When it comes to hardware, you will need a Raspberry Pi 2 or 3, a micro SD card, power supply and an ethernet cord or Wi-Fi dongle (Pi 3 has Wi-Fi inbuilt).

PORTABLE GAMING CONSOLE

NinTIMdo RP is a portable console which uses a 3D-printed case within which it houses a 7-inch touchscreen and console buttons. It runs on the **RetroPie emulation software**. The RetroPie OS on any Pi model can play a bunch of old **Game Boy, arcade, SNES and Atari games**. Users can build their own portable retro arcade system since there are many kits that come with controllers and cases to make this experience better. These consoles and arcade systems are powered by **PiJuice,** a battery module.

SOLAR WEATHER STATION

The Pi Solar Weather Station is powered by Raspberry Pi. It can sustain off the grid and send weather results via a wireless connection from any place. **PiJuice** is used as the power supply along with its added solar support. A sensing module is also utilised to record temperature, humidity, light levels and general gases. In order for the Pi-powered weather station to send text alerts to the users about the weather, an EFCom Pro GPRS/GSM module is incorporated to automate this process.

MULTIROOM MUSIC SYSTEM

The idea of a multiroom audio system is generally desirable for everyone. However, connected wireless speaks are not exactly cheap. But if you utilise the right software along with a couple of handy Raspberry Pis, you will get your very own inexpensive sound system that can play music in different rooms of your house at the same time. This integration will allow users to control their music players from anywhere in the house by just using a smartphone.

RADIO

Who knew that Raspberry Pi could actually do much more than computing? Turns out it can send out signals over FM airwaves as well. It is the perfect project for anyone who has ever wanted to start their own **personal radio station**. **Pirate radio is** a personal Pi-powered radio that is built using a radio receiver, an SD card, a network cable, Raspberry Pi and a fair bit of code. You can use this radio to tune into your favourite channels as well.

MICROWAVE

The Raspberry Pi Microwave made by **Nathan Broadbent** can be controlled by voice, your smartphone or the internet itself. It is better than traditional microwaves since it allows users more control over the device and has many added functionalities like voice control, tweets poster, remote access through a web interface or smartphone and even a **barcode scanner** which can procure cooking instructions from an online database.

DIGITAL/SMART MIRROR

The Magic Mirror is an extremely innovative smart mirror which houses a **one-way mirror glass** (the kind they use in police interrogation rooms) which is mounted over a flat display. This display device then outputs white text on a black background which is then seen as shown above in the image to the users. This mirror has been backed by Raspberry Pi and it can show users the time, abstract texts, weather reports and the news headlines for the day.

PRINTER

If you think traditional printers take up too much space and the cables constantly get in the way, then a wireless printer built with Raspberry Pi is a good solution. You will need to make sure that you have a connected wireless, already-set-up USB dongle for your Raspberry Pi. You will also ideally need to use a USB printer. Additionally, it is also possible to make this work with a parallel printer which is coupled with a parallel-to-USB adaptor. You will also require a USB cable connected from your printer to the Raspberry Pi. After this, the configuration process begins. There are a couple of tutorials online that should help you set up your Pi-enabled printer easily.

CUSTOMISED PICTURE FRAME

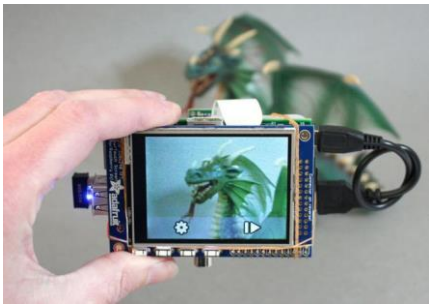
Digital picture frames are becoming increasingly common these days, however, with Raspberry Pi, you can make them customisable as well. All you need is a spare monitor or an existing digital picture frame which has a USB

connection. You can easily connect your Raspberry Pi to it with a USB-HDMI adaptor. There are scripts easily available online which can allow the Pi to automatically down pictures from **Deviant Art, Flickr and** other such websites every single day and display them as a slideshow.

THE BEETBOX

The Beetbox, which is powered by Raspberry Pi, plays some sick beats and is also made out of beets. Intrigued? It is basically an interactive drum kit which is made up of vegetables. You tap one of these beets and it creates a beat, like a real drum kit. It utilises a **capacitative touch sensor** which is connected to a **Raspberry Pi**. This sends signals to an amp inside the wooden case and produces beats. The source code for this cool DIY project along with instructions are available on **Github.**

TOUCHSCREEN CAMERA



Pi Touchscreen camera is basically a fascinating but simple 'focus and shoot' touchscreen digital camera. It houses the **Adafruit PiTFT touchscreen** and a Raspberry Pi camera board. The camera looks rather barebones, however, it is quite nifty. You can use Wi-Fi and Dropbox to automatically transfer photographs taken on this device to another computer in order to edit them, or even view them. The code is open-source so enthusiasts can customise it to do anything special they think of that other cameras are not capable of doing.

TABLET

Using Raspberry Pi, you can build simple tablets with LCD screens that can run computational tasks quite well. This is a cheap replacement for branded tablets. Users can customise the tablet as per their requirements, by choosing the size of the screen, the storage capacity and other such aspects. **RasPad** is one such example. It is a portable Raspberry Pi tablet which gives users access to all the Raspberry Pi ports and enables the creation of customised projects.

3D SCANNER

Raspberry Pi can be utilised to create 3D scanners that come in various shapes and sizes, from miniscule to life-size. These scanners utilise multiple **Raspberry Pis and Pi camera** modules to create 3-dimensional images of objects and living things. The created image can then easily be printed using a **3D printer** or you can place an order to any 3D printing company for the same. These 3D scanners are affordable alternatives to costly scanners in the market.

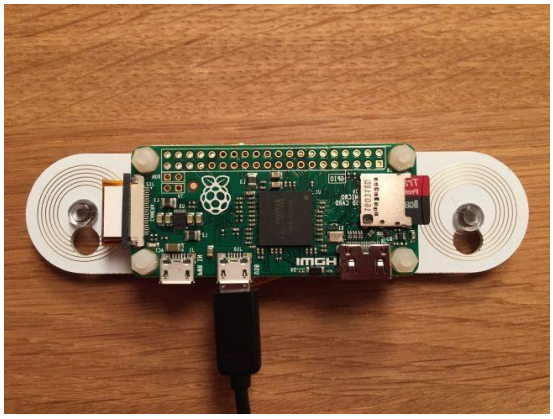
LIVE CALENDAR

Remember the times when we used to put up calendars at home at the beginning of every year? Maybe you still do. But they got obsolete over time ever since phone calendars became a thing because you could do so much more with them (reminders, events, etc). You can now use Raspberry Pi to build a digital wall-mounted calendar by utilising a spare monitor and, of course, Raspberry Pi. These calendars sync with Google Calendar and the Pi always opens this over the home **Wi-Fi by** using some clever scripting.

AI ASSISTANT

Google has partnered up with the official Raspberry Pi magazine, the **MagPi**, to create an add-on board which enables manufacturers to add voice control and AI support to their Raspberry Pi products. The folks at MagPi created a Google Home-based smart speaker powered by Raspberry Pi. They send over a cardboard kit with all the essentials to build this speaker, all you need to do is buy a Raspberry Pi board.

SECURITY CAMERA NETWORK



You can utilise Raspberry Pi to secure your home or office premises with an affordable system of Pi-powered security cameras. The network can have several cameras configured locally and over a network. You can program them to be able to capture still images as well as stream videos. Users can also set up additional features like motion detection and email notification by adding a bit of code to accomplish this.

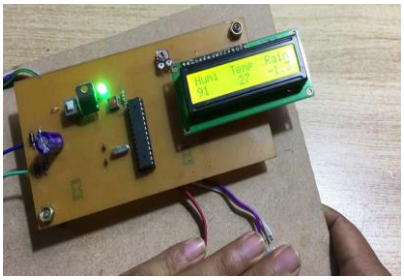
IOT Projects

Internet of Things (IoT) is a new predominant technology for this advanced world. This technology can change the lifestyle people lead. Question is what the Internet of Things is? IoT can be described as a network of physical objects connected through the internet. Physical objects could be anything that contains embedded electronics, software, sensor, etc. with the internet. Using the IP addresses, those smart objects can exchange data among the network and can make a decision. A significant number of researches is going on over the IoT trends and projects. In this article, we will talk about a few IoT project ideas based on standard IoT protocols, so that readers get the basic knowledge about the Internet of Things. These internet of things example are keen, useful, and interesting to build.

IoT Projects to Explore

Researchers are already working with advance, IoT based projects. But we will narrate here basic level IoT projects. Let's see how many readers like to drive their car using a mobile phone or control home appliances from office. Similar internet of things example we will discuss below.

1. IoT Based Weather Reporting System



At first, we are presenting a smart weather monitoring and reporting system here. To update the report manually is time-consuming. Here the necessity of automated reporting update solution arises. IoT based weather reporting system brings a solution where this system uses temperature, humidity rain sensors to monitor weather and report weather statistics online.It works constantly and sends data via microcontroller to the webserver using the WIFI internet. This system allows the user to set a threshold for a particular situation and alerts the user if weather reporting crosses the threshold value.

Important Features

- This system does not need human attention to monitor, as it is an automated system.
- It helps to collect data in tough environments like a volcano, minefield, polar zone, etc.
- Internet connection is needed to both ends.
- The future advancement of this project would be predicting weather forecast and disaster.
- To build this project, you have to know how to use Arduino.
- It uses a learn-do-review methodology.

2. Touch-Based Home Automation System

Internet of things concept works with almost every machine. But when this is all about our home appliances, IoT proposes a smart, automated system. Using IoT based automation system users can control home stuff anywhere from the world.In this article, we are talking about a touch-based home automation system based on microcontroller. It contains WIFI, inbuilt touch sensing input pins, which makes it helpful to make IoT based projects like this.

Important Features

- Adruino and ESP-32 microcontroller used in this project.
- User needs a smartphone or touchpad to control home kinds of stuff.
- Setup of ESP-32 in Adruino IDE.
- Minimum programming knowledge is needed.

3. Facial Recognition Door with Raspberry PI

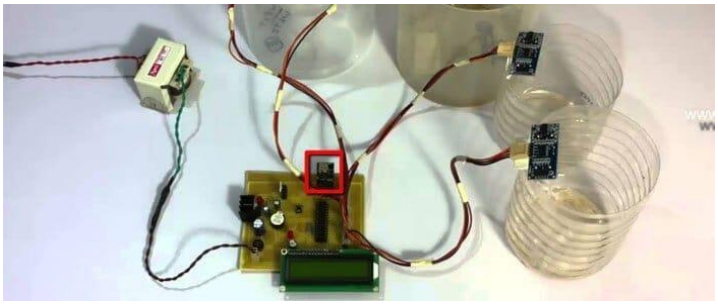
Imagine that you are standing in front of your door, and it opened up recognizing your face. This is an intelligent internet of things example. Science fiction becoming true nowadays with the hand of IoT project ideas. A smart door

secures the gateway and ensures the right person entering your home. Microsoft already made face API by their research.

Important Features

- This project built on three phases that are 1.Data gathering 2. Training recognizer 3.Facial recognition.
- Python code is used for data gathering.
- Raspberry Pi camera is used for facial detection.
- This internet of things example is not 100% accurate.

4. Liquid Level Monitoring System

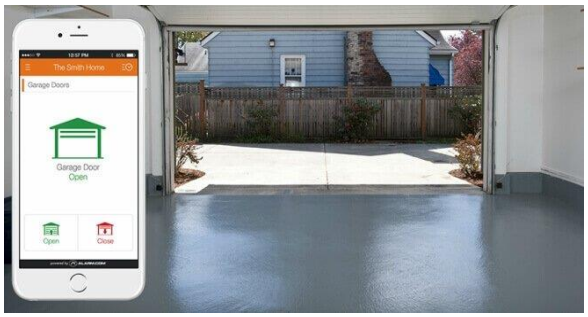


Let's talk about IoT based projects like a liquid level monitoring system. Liquid level monitoring system designed in a way that users can remotely check the level of the liquid. It has vast applications in the industrial sector where the user needs to monitor the level of liquid, whether it is below the mark of overflowing.Theft detection, usage of chemical and leak detection are some of the usages of the liquid level monitoring system. **Ultrasonic, Conductive, and float sensors** are some of the few sensors for a monitoring system.

Important Features

- The ESP8266 Wifi module is used to connect with the internet and transmit data to the required website.
- The ultrasonic float sensor will send data about the level of the liquid.
- Users will have a history of records from the website.
- It is easy to monitor from anywhere.
- In the future, this internet of things example will be developed for Bluetooth technology.

5. Smart Garage Door



Days of that bulky electronic key has gone. These days you can use your smartphone or tablet computer to open your garage replacing that clicker. Not only monitoring with a smartphone is the feature of a smart garage door using **laser and voice command is an addition**. Smart notification in run time gives an alert when it close or opened that is very much helpful for the busy families.

Important Features

- Users can add IFTT integration that can create custom commands for google assistants.

- IoT based garage door increases the security of home.
- This internet of things example is easy to setup.
- **Flask web** server used in Raspberry PI to control the garage door.

6. IoT Based Alarm Clock

We use an alarm clock to wake up in the morning. But this is the world of IoT platform-based product. A small alarm clock can give you a lot more things, not just wake up call. Imagine a morning when your alarm clock just made you wake up from the morning, turned the bathroom light on, start playing your favorite music, open the curtains. One of the gifts from the IoT based idea is a smart alarm clock. Below there are essential features of IoT based alarm clock.

Important Features

- Users can set the alarm via smartphone.
- Voice command feature can help the user even to start a video chat.
- Automatic brightness adjustment according to day or night.
- Audio amplifier volume control by voice command.
- Apache 2 server can be used for this internet of things example.

7. IoT Based Air Pollution Monitoring System

Air pollution is a common problem nowadays. Different particles in the air like led, carbon dioxide, sulfur dioxide, pollen, and mold spores are making so much air pollution. Air pollution can bring lots of diseases.

It is essential today to use a mechanism to measure air pollution in an area. Research on IoT projects brings a solution. Newly discovered IoT devices can monitor air pollution and save data to the web servers.

IoT project ideas like air pollution meter bring a solution to the existing problems like previous air pollution meter was out of memory after some time. But IoT device uses the internet and saves data to the remote web server it has now become so easy to get a log of data within an area for specific days.

Important Features

- This internet of things example can also detect flammable gas leaks.
- Particle matter detector, gas sensor, temperature, and humidity sensor are used.
- This type of project built based on Arduino Uno.
- Very helpful to detect air pollution close to the hospital or school.
- This project is cost-efficient.

8. Night Patrolling Robot

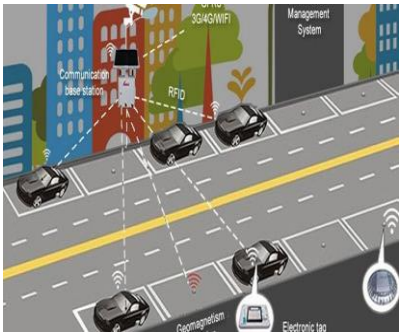
Security is a common concern for all. As most of the crime occurs at night, so the IoT project comes up with a solution that is a patrolling robot that uses a night vision camera. This robot patrol over a predefined path and detects alarming sound.

If found, it scans the area with its 360 degrees moving the camera and try to detect any human face. Then it transmits the image to the nearby user who is running this whole IoT project. The user gets the alarming notification to send by the robot.

Important Features

- IR sensor makes it happen that a robot will patrol through a defined path.
- USB camera and Raspberry Pi are connected.
- Python's language helps make its software.
- This robot can be cheaper than hiring multiple security guards.

9. Smart Parking System

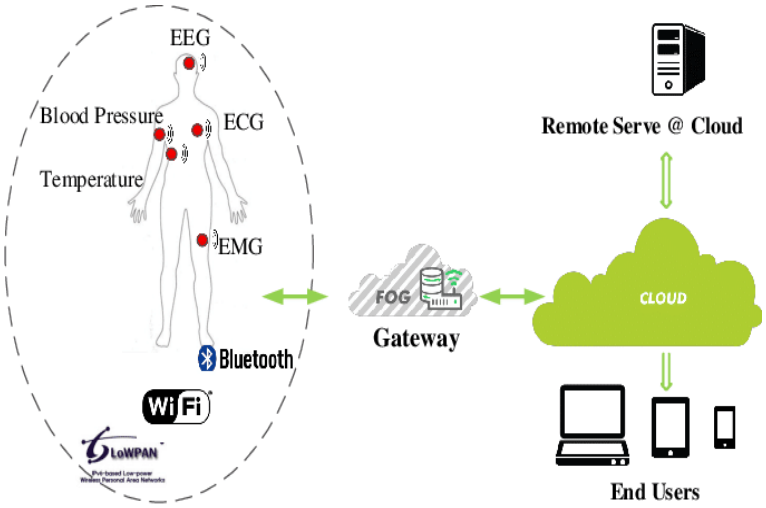


Finding parking space is a problem for the driver. Sometimes it kills a lot of time of driver to find a parking space. IoT based project smart parking system brings a solution. A major purpose of this project is to avoid unnecessary traveling by a driver for the parking area. Monitoring the whole area at the run time gives the driver an image of the entire parking area, and the user can select that free parking space.

Important Features

- Daily life problem solution by IoT based ideas.
- It uses an IR sensor for detecting free parking space.
- Illegal parking can be reduced with this internet of things example.
- This IoT based project is built on the Arduino board.
- The infrared proximity sensor can be used for finding space.

10. IoT Based Health Monitoring System



Health is the most valuable treasure of human life. With the course of time the life of people becoming so stressful that we are giving less care to our health. People are hardly going for a checkup. IoT projects like health monitoring systems can make a solution here with the devices that monitor our health regularly and send data to the doctor.

The doctor can check the current situation at any time and anywhere from the world. It is possible that sensors in the body of the patient can estimate blood pressure, sugar level, and heartbeat and immediately alarm the doctor if it is higher than normal.

Important Features

- Adruino will generate the output.
- The doctor can check the current condition of the patient using his smartphone.
- Use of firebase for run-time data.
- Communication between Adruino and android app needed.
- This internet of things example uses Bluetooth technology.

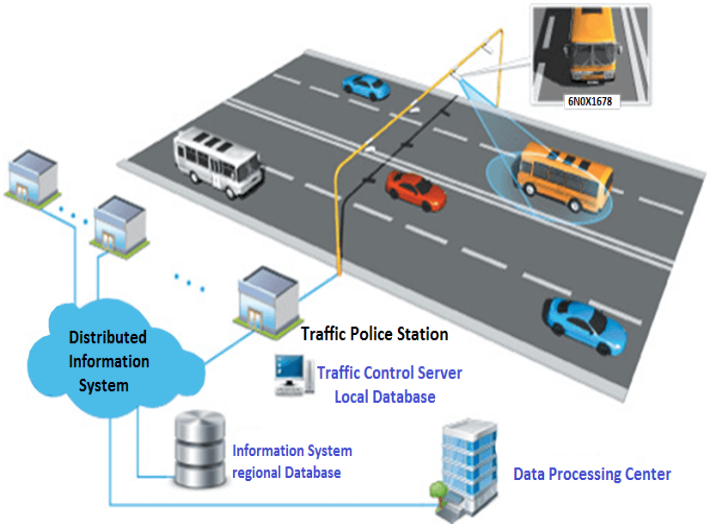
11. IoT Based Smart Water Irrigation System

Agriculture plays a vital role in a country's economy. It is difficult for the farmer to monitor the moisture level of the whole field and supply water. IoT project like smart water irrigation system proposes here an automated water irrigation system that can analyze the moisture of soil and climate condition. Users will be able to check the moisture level, and with the predefined threshold for a moisture level of soil, the power supply will be cut-off.

Important Features

- Aduino/328p microcontroller is used to control the motor that supplies water.
- Users can switch on/off the motor from the webpage.
- This internet of things example will automatically stop if it is raining.
- In future data of different sensors will be shown on the BOLT cloud in graphical form.

12. IoT Based Traffic Management System



Almost every metropolitan city faces traffic problems. As the population is increasing day by day, the necessity of a smart traffic management system is undeniable. IoT based project like smart traffic management system can reduce the traffic problem. Sometimes it is difficult for the ambulance to cross the traffic.

Counting advantage of an ambulance, this management system connects with the ambulance driver and helps to find the signal where traffic flow can be controlled dynamically. This internet of things example also monitors traffic rules violators.

Important Features

- Finding an emergency path for an emergency situation is easy.
- This IoT project can be used anywhere.
- Can identify traffic violator at night.
- It shows green light only for an ambulance, fire truck, or emergency vehicle.

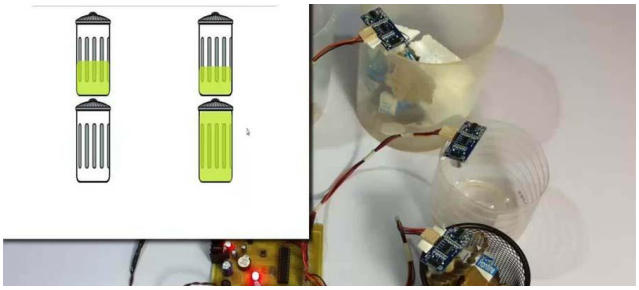
13. IoT Based Baby Monitoring System

The idea is of the smart cradle system that will help parents to monitor their infant child from a remote place. This idea comes up with a cry detecting mechanism, Live video surveillance, cloud computing data, and user interface as mobile or web version. Different sensors attached to a cradle will check the humidity or temperature of the bed.A surveillance camera on cradle will always send footage of the main IoT program. All the data will be stored in the cloud. Based on that data, a health algorithm will always check the condition of that infant and alarm parents if any unwanted situation appeared.

Important Features

- This project will reduce the pressure of parents from a monitoring infant child.
- This is a Raspberry Pi-based project.
- Software language could be Python.
- Instant app notification system.
- Analysis sound of baby and alert parents.

14. IoT Based Garbage Monitoring System



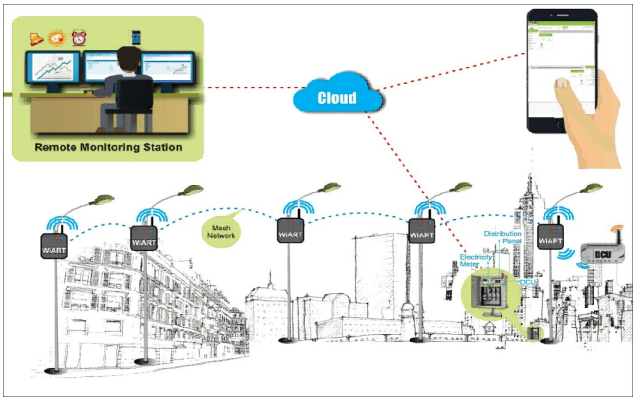
With everyday life, we produce a vast amount of garbage. We can smartly handle that garbage collection system with the help of IoT based project. The garbage monitoring system will help to clean the city more innovatively.

This IoT project idea uses an ultrasonic sensor to detect the level of garbage in each garbage bin and send those data to the main IoT program. A webpage shows a level of garbage on each garbage bin and highlights the amount of garbage on each bin. A buzzer puts on when garbage is over the limit.

Important Features

- For sending data, each bin uses Wifi Modem.
- IoT gecko web development platform used for graphical representation.
- Programming language C used in here.
- Adruino Compiler and HC-SR04 ultrasonic sensor can be used for this.

15. IoT Based Street Light Monitoring System

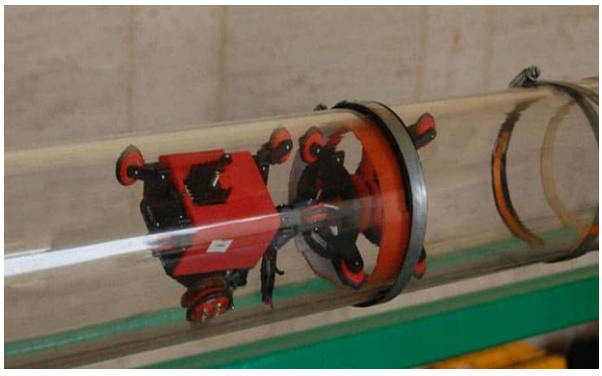


According to a study, street light consumes 19% of world energy. Most of the street light remains On although there is no one around. IoT project ideas like street light monitoring system bringing a solution here. This project consists of smart street lights that have sensors to monitor humans or vehicles around.Upon sensing the movement, the sensor sends data to the microcontroller; then it turns the street light on. If there is no movement microcontroller make the switch off. That's how this IoT project saves energy. Checking of faulty street light is another advantage of this internet of things example.

Important Features

- LDR sensors used in this project.
- Programming language C or Python can be used to do this whole project.
- Users can get data from the internet.
- Automatically shut down in day time, which saves energy.

16. Gas Pipe Leakage Detector Robot



The gas pipe is an important part of the current industries. But leakage in the gas pipes can cause fire accidents, toxicity in air and soil and serious havoc. Gas pipe leakage needs to be found very quickly to stop any massive situation.

Typically leakage found through acoustic sensor or footage from the small video camera. But IoT based project can give a better solution. A tiny robot can crawl through a gas pipe along with its interface GPS sensor which can notify the leakage location.

Important Features

- This internet of things example can detect leakage of 1-2 mm in size.
- IoT gecko platform used for getting output from the sensor.
- It can also detect leakage in a water pipe and petroleum pipe.
- It moves 3mph along the pipe
- Limitations: It needs a reasonably uniform pipe to move. But researchers are working to make it more flexible to crawl through any pipe.

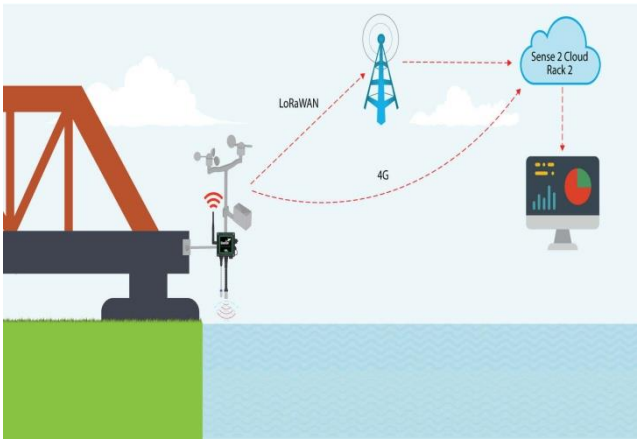
17. Smart Baggage Tracker

Although only 1% of bags lost in airline travel, it is a vast number. IoT based project like baggage tracker can innovatively help in the tourism sector. The idea is very simple. Tourist needs to use a tracker on their travel bag. This project will send its coordinate to the users' phones. This internet of things example will be handy for tourists.

Important Features

- This project uses a board called FONA.
- Its microcontroller uses its GSM module.
- It reports its position every minute.
- ThingsSpeak API is used in this project for getting the coordinate of the device.

18. IoT Based Early Flood Detection and Avoidance



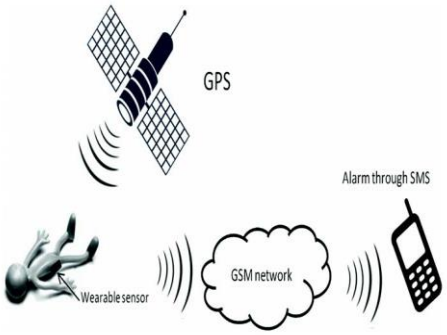
Flood is a common natural disaster. Flood causes loss of life and can destroy the economy of a country. Early flood detection can reduce the loss of life and property. From this concept, IoT proposes an idea of an early flood detection system.

This project counts various natural factors, including humidity, temperature, water level, and flow level to detect flood. The flow sensor monitors the flow of water. That result can be accessed from any IoT device from any parts of the world.

Important Features

- To detect temperature and humidity, this project uses the DHT11 sensor.
- Float sensor always checks the water level.
- A flow sensor always monitors the flow of water. It consists of a plastic valve body, a water rotor, and a hall-effect sensor.
- This IoT project also uses the HC-SR04 ultrasonic range finder distance sensor.

19. IoT Based Wheelchair Fall Detection



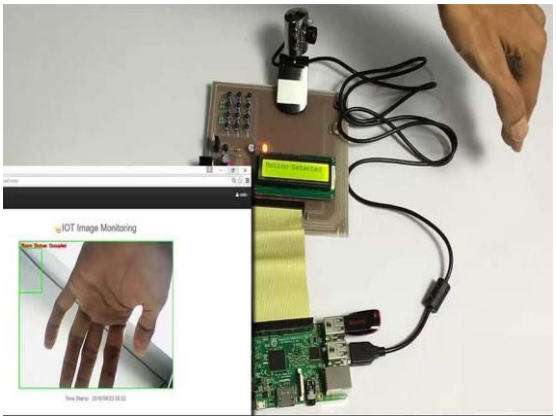
Older adults need to handle very carefully. Sometimes they can't walk, so they use a wheelchair. But older people fall from those chairs for a few reasons. It is important to help them immediately if they have fallen from a wheelchair.

This IoT based wheelchair fall detection project brings a solution here. If you are using the accelerometer and gyro-sensor on the hand of the patient or the wheelchair. Any jerk in the system will be counted as a fall from the wheelchair that will trigger the alarm.

Important Features

- If the alarm is false, then the patient can stop the alarm within a few seconds.
- IoT Gecko platform used for software building
- This IoT example is based on the Arduino compiler and ESP8266 Wifi module.
- Programming language C is used in here.

20. Smart Anti-theft System



With the rapid growth of modern civilization, security has become a prior choice for almost everyone. Everyone wants to secure their home or industry from the intruder. IoT project idea gives a solution here. When the user goes out from the house, they have to turn it on the antitheft system which will monitor all the floor, and any footstep on the floor tiles will make alert the main IoT program.

When an intruder enters the house sensor sends data to the microcontroller. The microcontroller then makes it a valid signal and moves the camera and takes a picture. Users can see that picture on his smartphone.

Important Features

- Piezo sensor used for getting movement.
- IoT gecko platform for web-based user interface
- Increases security level with immediate image capture.
- Raspberry Pi microcontroller is used for it.

Open Source IoT Protocol

- **Advanced Message Queuing Protocol (AMQP)**

Open standard for business messaging Internet protocol. It communicates between applications or companies seamlessly connecting systems, feeds business processes with the information and transmits instructions to attain objectives in a reliable manner. AMQP connects different aspects of Organizations, technologies, systems not available simultaneously, as well as operate at a distance in case of poor network.

- **Constrained Application Protocol (CoAP)**

https://coap.technology/

The Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained networks in the Internet of Things. The protocol is designed for machine-to-machine (M2M) applications such as smart energy and building automation.
CoAP is specified in a Standards-Track RFC. RFCs that serve as Internet Standards Documents are generated by the

IETF based on an extensive technical review and quality control process. CoAP is simple enough to implement from scratch for a simple application.

- **Very Simple Control Protocol (VSCP)**

https://www.vscp.org/

While the term protocol may sound misleading, VSCP is a framework. It is a scalable, free and open solution framework for the discovery and identification of devices, configuration, autonomous device functionality, securely updating the devices — overall, a solution from the sensor to the user.

The word “Protocol” may be misleading. VSCP is much more and should probably be called a framework instead. VSCP is a scalable, a very low footprint, a free and open solution for device discovery and identification, device configuration, autonomous device functionality, secure update of device firmware. VSCP is an application level protocol making things interact using CAN, RS-232, Ethernet, TCP/IP, MQTT, 6LowPan.

- **MQTT**

(MQ Telemetry Transport or **Message Queuing Telemetry Transport**) is an open OASIS and ISO standard (ISO/IEC 20922) lightweight, publish-subscribe network protocol that transports messages between devices.)

It the standard messaging and data exchange protocol for the Internet of Things (IoT). The MQTT protocol provides a scalable and cost-efficient way to connect devices over the Internet. You can use MQTT to deliver data over the Internet in near real-time with predefined guarantees of delivery. Connecting millions of IoT devices to your business infrastructure, sending instant updates, and moving data efficiently is where MQTT truly excels.

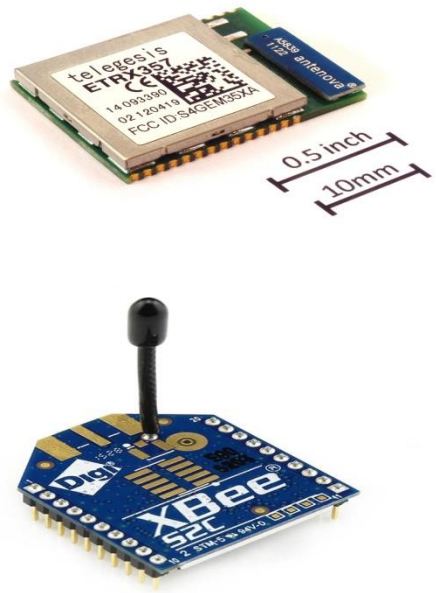
Why MQTT is used in IOT?

MQTT ensures that messages go to the correct devices during communication by utilizing topics. A topic functions the same as a file path would, and directs communication by filtering messages according to elements specified in the topic function.

MQTT is a publish/subscribe protocol that allows edge-of-network devices to publish to a broker. Clients connect to this broker, which then mediates communication between the two devices. ... When another client publishes a message on a subscribed topic, the broker forwards the message to any client that has subscribed .

- **Zigbee**

Zigbee is a wireless technology developed as an open global standard to address the unique needs of low-cost, low-power wireless IoT networks. The Zigbee standard operates on the IEEE 802.15. 4 physical radio specification and operates in unlicensed bands including 2.4 GHz, 900 MHz and 868 MHz.



Zigbee XBee Module S2C 802.15.4 2mW with Wire Antenna XB24CZ7WIT-004

Why ZigBee is better than WiFi?

ZigBee's data transfer speed is lower than WiFi's, too. It's maximum speed is just 250kbps, much lower than the lowest speed WiFi offers. ZigBee's best quality is its low power-consumption rate and battery life.

Can ZigBee connect to WiFi?

ZigBee and WiFi channels both exist in the 2.4 GHz band, existing in the exact same frequency space. When deploying both WiFi and ZigBee in the same environments, careful planning must be performed to make sure that they don't interfere with each other.

Zigbee Applications

Zigbee enables broad-based deployment of wireless networks with low-cost, low-power solutions. It provides the ability to run for years on inexpensive batteries for a host of monitoring and control applications. Smart energy/smart grid, AMR (Automatic Meter Reading), lighting controls, building automation systems, tank monitoring, HVAC control, medical devices and fleet applications are just some of the many spaces where Zigbee technology is making significant advancements.

- **6LoWPAN**

6LoWPAN is an acronym of IPv6 over Low -Power Wireless Personal Area Networks.

6LowPAN. A key IP (Internet Protocol)-based technology is 6LowPAN (IPv6 Low-power wireless Personal Area Network). Rather than being an IoT application protocols technology like Bluetooth or ZigBee, 6LowPAN is a network protocol that defines encapsulation and header compression mechanisms.

- **WiFi**

Wi-Fi is the name of a wireless networking technology that uses radio waves to provide wireless high-speed Internet and network connections. To connect to a Wi-Fi LAN, a computer must be equipped with a wireless network interface controller.

A common misconception is that the term Wi-Fi is short for "wireless fidelity," however this is not the case. Wi-Fi is simply a trademarked phrase that means IEEE 802.11x.

Funded Projects

1.(IoT Security) **Title: Lightweight Code Self-verification for Internet of Things (IoT) Devices,** funded by MeitY-NWO under Indo-Dutch Joint Research programme for ICT, (2015-2020), Project outlay: 380, 000 Euros (INR 2.6 Crores).

Summary:

The Internet of Things (IoT) is the network of physical objects (things) embedded with electronics, software, sensors and connectivity modules to enable these objects to achieve greater value and service by exchanging data with other connected devices. The IoT paradigm has been emerging for decades, with origins in factory automation, and embedded systems. The fascinating opportunities available through the usage of IoT today, could take any developing country's economy to a larger sustainable economy by reducing food wastage, reducing energy consumption etc. Along with massive deployment of IoT systems come massive problems. The operation of IoT devices in non-controlled, possibly hostile environments puts the dependability and reliability of IoT systems at stake. More specifically, adversaries may tamper with the devices, intervene their communication channels or clone the devices, to instrument (engineer) the data gathering and overall operation of IoT systems to their own interest. The key idea of this proposal is to use the advanced exploitation technique of **Return Oriented Programming (ROP),** not as an offensive mechanism, but rather as a defensive weapon, to protect programs running in hostile and possibly resource constrained IoT environments against tampering by adversaries. Specifically, our code verification approach will protect code running on IoT devices by overlapping ROP gadgets (ROP chain) within the code. These translated functions will implicitly verify the integrity of the protected code, by malfunctioning if the ROP gadgets are tampered with, effectively acting as self-verifying code.

2. (Cyber Security) Title: **Automated Detection of Security and Privacy Threats in Peer-to-Peer Networks,** funded by DeitY, Ministry of Communication & Information Technology, Govt. of India, New Delhi, March 2012, (March,2012 - June,2015), Amount: 61,95,000 INR.

Summary:

With the proliferation of P2P systems, it is critical to consider the impact of these systems on the security of an Internet environment that is already struggling from several security issues. The Indian IT act 2000 and 2008 have provisions for punishing cyber-crimes like hacking, denial of service, breach of confidentiality, publishing of fake digital certificate etc. Other developing, and developed countries have more or less similar regulations. However, most of the countries have less stringent regulations on P2P application usage. Empirical studies indicate that P2P and Web traffic together dominate today's Internet traffic. Currently, products like Snort, Sourcefire VRT etc. detect and alert policy violations for usage of P2P applications using techniques like port-based analysis, and protocol analysis. In this research, we have developed a research prototype to assess the impact of P2P traffic on perimeter security appliance, and also developed intelligent approaches to counter their impact on a large campus wide network.

---

3. (Overlay Search) Title: Efficient Peer-to-Peer Overlay Infrastructure for Secure Computing over the Internet, funded by University Grants Commission (UGC), Govt. of India, New Delhi: (2011 - 2014), Amount: 12,50,000 INR.

SUMMARY:

Peer-to-Peer (P2P) overlay networks which connect peers on top of physical networks like IP, and be it over a wired or over a wireless medium, are growing dramatically in their usage. P2P application usage has grown steadily since its inception around the year 2000. Client-server based architectures are characterized by asymmetric relationship between client and server where client queries and server responds. Contrast to that in distributed P2P systems every node acts as both a server and a client. Peer-to-Peer overlay structure is brought to popularity by the file sharing applications like Napster, Gnutella etc. A P2P overlay network is a logical network built at application layer providing connectivity, routing, and messaging amongst addressable end-points of the communication. These overlay networks have been used for file sharing (torrents), voice over P2P (Skype), and streaming media (P2Pstream) in recent times. The spring 2011 global Internet phenomena report by Canadian ISP Sandvine points to the increased percentage of P2P traffic in North American fixed access networks from 15.1% in 2009 to 18.8% in 2011. In this project, authors have tried to achieve the following two objectives that are critical in improving the performance of large scale file sharing P2P overlays:

[A].To analyze existing protocols for location management (lookup problem) in peer-to-peer overlays and propose efficient approaches to reduce overlay maintenance cost and network congestion. Under this objective, the main goal is to develop lookup algorithms to optimize the decision making process at each peer by considering the locally available information at each peer like files uploaded/downloaded, bandwidth available, etc. and global information available like type of files at other peers, congestion in the network, replica availability at other peers etc. The target overlay is this research is unstructured P2P overlays mainly file sharing overlay networks.

[B].To propose novel algorithms to safeguard against Sybil attacks in Peer-to-Peer unstructured overlay networks. Sybil attacks are a major source of concern in any collaborative activity, and particularly in a file sharing P2P network. These attacks are mounted on a sophisticated scale where the attacker stills multiple identities and uses those to disturb the collaborative work. Using Sybil attack, attacker can spoil byzantine consensus, can drop forwarding packets to the neighbor peer, can forward the packets in a wrong route etc. Our objective in this part of the project to either detect these Sybil identities or reduce their impact on the overlay services.

International Funded projects

Project #1: Investigating the Value versus Privacy Cost of the Internet of Things.

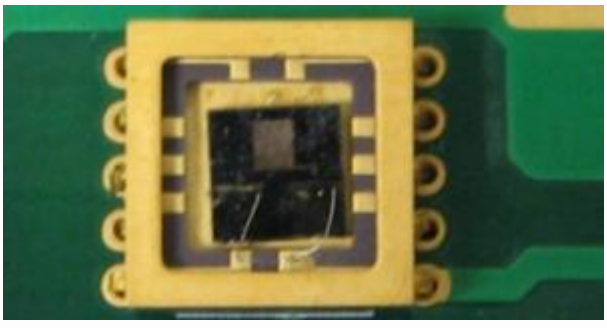
Research: By applying relevant privacy frameworks from existing research literature (e.g., Privacy Calculus, Contextual Integrity, Association of Public-Safety Communications Officials, Internet Users' Information Privacy Concerns, etc.), students will design their own experimental studies, collect data, and quantitatively analyze the data to understand the interplay between IoT data sharing practices and users' privacy perceptions and outcomes. REU students will use a modified version of the Android operating system and a mobile application developed in the faculty mentor's lab (i.e., PrivacyTrader by Dr. Turgut), to intercept the information flow between the Android devices and service providers. The students will conduct user studies that measure the actual information collected by the IoT devices, as well as the relative value users put on the services provided by the device versus the value of privacy lost through the disclosure of the information. The students will also perform de-anonymization research experiments to help test the value of anonymization measures and to develop better ones.

Project #2: IoT Device Vulnerabilities and Security

Research: IoT devices offer unparalleled functionality and convenience to both consumers and industry. As such, these devices are becoming more popular each day. However, as the number of connected devices increases, so do the security and privacy concerns. The unique properties of IoT devices and their design flow create a different breed of vulnerabilities unlike those found in general purpose computing systems. In contrast to general purpose systems, IoT developers frequently update the device's firmware to provide more sophisticated functionality. Also, due to the shorter time-to-market (TTM), IoT devices often exhibit more debugging ports, such as testing interfaces and field update vectors, which make hardware-based attacks another threat. As such, the first topic in this project for undergraduate students is to investigate the security vulnerabilities on commercial IoT devices. Device hacking demos from the Dr. Jin and Dr. Zhang's lab will be introduced, which students can use as inspiration, from both theoretical and practical aspects, to prove the ineffectiveness of current security protection methods on IoT devices. This research will further lead the students to investigate the fundamental reasons why the existing methods cannot protect IoT. Students may work on various directions which include: 1) Firmware Updating Schedule. The firmware updates happen rarely in the traditional embedded systems. However, for IoT devices, developers will update the firmware more frequently to install patches, and provide more sophisticated functionality. 2) Booting Process. The booting process becomes more vulnerable and leaves backdoors for attackers to exploit in IoT devices since security checks for booting process are rarely installed. The students will also be encouraged to perform independent studies on IoT security from the angles that are not covered from this project. This research is currently being supported by SCEEE grant 15-001: "Avalanche Effect in Cyber-Physical Systems Security Under Large-Scale Cyberattacks on Smart Devices" to which the students will contribute.

Project #3: Low Cost, Ultralow Power Sensor Design and Fabrication for IoT

Figure 1: A packaged gas sensor



Research: In this research project, students will gain research experience in the design and fabrication of ultralow power, low cost chemiresistor-based sensors. Low power consumption and low cost are critical design considerations to IoT devices such as chemiresistor-based gas sensors that are widely used for safety monitoring in homes. These sensors require a fast response time, high selectivity and low detection limit. To achieve high sensitivity, traditional sensors require extra heating elements, adding to the cost and power consumption. This leads

to the research challenge of designing and fabricating sensors based on nanoscale structures that achieve their performance targets without additional heating elements. In this project, the students will gain hands-on experience with the fabrication of a sensor using indium-doped tin oxide nanowires grown via the VLS (vapor-liquid-solid) method. The packaged device will be similar to one in Figure 2. With varying electrode configurations and processing conditions, students will study the dependence of sensor characteristics on design variables and processing conditions, compile and explain their findings in relation to the working principles of sensors.

Project #4: Energy-Efficient Computing Devices for IoT

Research: The students will perform and evaluate several key computing devices in an IoT environment. In particular, we let students to implement and experiment key components of a small synthesizable 32-bit RISC microprocessor. This embedded processor is one of the most critical computing devices in the modern IoT infrastructure. Our experimental system is currently running a live web server with an interrupt controller, UART, SRAM or DDR SDRAM controller, and Ethernet controller. This CPU executes all MIPS I(TM) user mode instructions except unaligned load and store operations. This “clean room” CPU core is implemented in VHDL with either a two or three-stage pipeline and running at 25 MHz on a Xilinx FPGA and also verified on an Altera FPGA, which now contains a bidirectional serial port, interrupt controller, and hardware timer. Students, under our supervision, will be responsible to add a DDR SDRAM controller, Ethernet MAC, or Flash interface. We will provide C and assembly code for its Real-Time Operating System, a fully preemptive RTOS supporting threads, semaphores, mutexes, message queues, timers, heaps, an interrupt manager, ANSI C library, single precision floating point library, TCP/IP protocol stack, and Web server. In short, we will provide a minimized yet complete computing IoT node to the participating students, and let them to add and modify it with the objective to understand how energy-efficient computing device is essential to the success of IoT.

Project #5: Generating Privacy and Security Threat Summary for Internet of Things

Research: This project’s primary research goal is to identify and summarize critical privacy and security threats of IoT technology through analyzing data from multiple news sources. Privacy and security are the top priorities. The project includes: (1) collect news articles from major U.S. news sources that are related to IoT privacy and security issues. The articles may include a) news releases for new IoT technologies and projects; b) media coverage of existing IoT technology vulnerabilities, e.g., media reports of IoT privacy and security issues, which have surfaced either through daily use or after malicious attacks. (2) identify and summarize privacy and security issues of the current and future IoT. Using the collected news data, students will learn to use state-of-the-art text analytics tools and investigate the emerging issues and perhaps manually summarize the cause, involved IoT technology, and consequences. Later, they will learn to use natural language processing, machine learning, and other data analytics tools to process the collected data and automatically generate text summary for the privacy and security threats. Students will perform research investigation using the extractive and compressive summarization techniques

developed by Dr. Liu and colleagues to generate condensed text summary. Future study may include exploring language generation techniques to summarize the issues into text abstracts. Junior and senior undergrad students will be involved in data collection and gain hands-on experience on natural language processing research, and become aware of the critical issues related to the next-generation IoT technology. The summarized privacy and security issues may provide useful suggestions to the IoT technology developers and government agencies.

Project #6: Protection Scenarios to Preserve Privacy and Security within IoT Use Cases in Medical Simulation

Research: Organizations must comply with legal policies governing the privacy and security of devices and information involved in IoT use. This is especially true for health care providers dealing with laws protecting privacy such as HIPAA. Integrating effective practices for IoT security and privacy require new forms of analysis of monitoring. Some activities include identifying behaviors of users that hinder or promote the application of effective IoT principles. In health care settings, users bringing their own devices pose new opportunities and threats to health care provision. The goal of this project is to mitigate risk and inform effective practice to integrate IoT in a variety of health care use cases such as in a medical office, in-patient and out-patient hospital setting, surgical theatre, and military combat battlefield. The students will have the opportunity to conduct research for integrating secure and private IoT applications across a variety of use cases based on actual projects conducted in the Mixed Emerging Technology Integration Lab (METIL) at the UCF Institute for Simulation and Training. Medical simulation scenarios will be modeled using virtual environments based on real use cases that incorporate 3D visualization of human decisions and IoT interactions.

Project #7: Modeling Social Network Structures and their Dynamic Evolutions with User-Generated Data from IoT

Research: With user-generated data (UGD) from the IoT, such as GPS locations and their contacts, the involved undergraduates will develop new algorithms and systems to derive the social network structures and study how these structures evolve over time with continuously incoming streams of UGD. The blossom of IoT devices like smartphones and wearable devices, as well as the popularity of various apps on these devices, make it possible for us to study the social network structures underlying the uses of IoT and reason about the social and economic factors accounting for the formation, evolution and termination of these structures. This will deepen our understanding of many computational concepts and principles characterizing the users' behaviors and intentions, thereby allowing companies to better serve these users. In this research, the involved students will have opportunity to sharpen their knowledge on graph theory, and perform research on deriving, tracking and predicting the latent social structures, such as communities and circles, over time. For example, the students will develop new models and algorithms to study how the social communities form and evolve by modeling and analyzing their GPS routings and communication patterns with their contacts. This research will reveal the dynamic interactions between the social communities that will evolve over time, splitting into smaller communities or merging into bigger ones. The students will base their modeling of the social structures on the UGD generated from IoT devices. Our previous

projects have collected such UGD to build computational models for social network analysis. This will serve as a starting point for the students to start their research explorations. However, the undergraduates will need to tailor and clean the various genres of UGD to fit into the new research problems. The students will also design new mechanisms and scenarios to collect the UGD upon the IoT. For this purpose, this project will be coordinated with the other project(s), developing new apps to boost the users' involvements as well as addressing their privacy concerns. The new scenarios will help reveal the hidden social network structures previously unknown to us from new dimensions of UGD measuring the users' interactions.

Project #8: Internet of Hospital Things (IoHT): Communicating to Facilitate Healing

Research: This project is related to the application of IoT principles to devices in a hospital. Today's hospitals need to respond to patient needs, frequently under chaotic circumstances, while simultaneously keeping devices, drugs, and people meticulously organized to protect and prepare for patients. Medical devices, ranging from the mundane (e.g., a bed) to the complex (e.g., Doppler ultrasound devices, infusion pumps, and X-Ray machines), are surprisingly easily misplaced and difficult to monitor for preparedness (e.g., cleaning) and maintenance. We generally benefit from advances in devices and procedures, yet the advances are often accompanied by complexity that can increase the risk of human mistakes. While various real-time location tracking technologies are available commercially, most devices are typically isolated from each other, the hospital, and the patient. In the IoHT, defibrillators could be queried for their battery health and calibration status, infusion pumps could be remotely programmed for medication concentrations and flow rates, and patient-worn IoHT devices would estimate patient stress/pain and make medication adjustments. We seek to develop and test a proof-of-concept IoHT module that supports a common device interface protocol (e.g., IrDA) for one or more devices (e.g., a defibrillator or infusion pump), along with software implementing a general protocol and user interface that supports remote monitoring of device functions, perhaps including device location. We will test the modules using medical devices used at the College of Nursing for training purposes. We will work with technology specialists at Florida Hospital (FH) on specific device and design choices. (FH already tracks Stryker beds, infusion pumps, and even nurses.)

Project #9: Investigating User Benefits and Risks Associated with IoT use,

Research: Users adopt wearable devices despite privacy and security concerns because the technology affords some level of value. Personal benefits may include enhanced self-awareness, positive behavior change, and cost savings. A number of such wearable IoT devices also allow end users to connect with other users to create synergistic social benefits, such as a sense of belonging, friendship, support, and even competition. For example, Fitbit users are able to track their steps to meet their personal fitness goals, but leveraging Fitbit's social networking platform allows Fitbit users to also benchmark their progress with their friends, providing additional support and motivation. Many IoT devices have social plug-ins to external social platforms, such as Facebook, that allow users to disseminate their personal information to a broader network of others. It is important to understand users' perceptions on why it may

be valuable to share certain types of information so that the benefits of wearable IoT devices can be optimized. Yet, as these devices become more and more socially integrated, the privacy implications become even greater. Even when using wearable IoT devices for their intended purposes, users may share personal information accidentally or with unintentional, negative consequences. The goal of this project is to understand the social benefits and privacy risks faced by users when using devices for their intended purposes. Using qualitative techniques (e.g., semi-structured interviews, think-aloud cognitive walkthroughs with users of privacy management IoT interfaces, or web-based survey or diary studies), students will have the opportunity to conduct an in-depth, user-focused analysis to better understand the potential social benefits and privacy implications for IoT end users given the complex information sharing practices between wearable IoT devices and various social networking platforms. From this analysis, students will identify important emerging themes, suggest new features, and pinpoint potential opportunities for interface redesign that could enhance the benefits of IoT and wearable devices and/or minimize privacy risks for users.

Project #10: Innovative IoT Applications in Scientific Research and Consumer Market

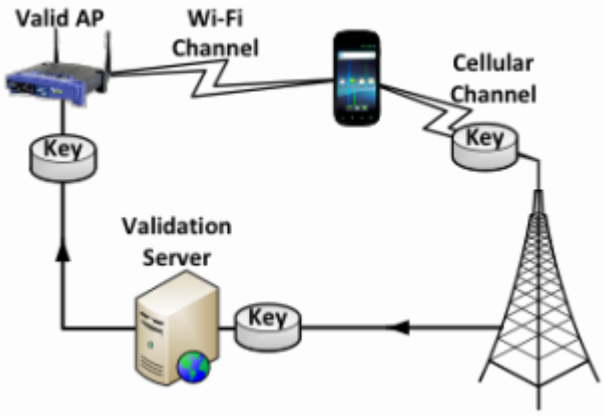
Figure 2: A Sample Home Automation System Support by Smart Power Grid

Research: The development of IoT devices provides opportunities for innovative applications that would either significantly reduce the cost or achieve much higher performance. Supported by the REU Site, undergraduate students will have the chance to investigate new IoT applications to solve real-world problems. The research tasks will be divided into two main directions to help improve quality-of-life and to help preserve our natural environment. Towards the first direction, students will do research in improving the intelligence of home automation systems where different types of IoT devices will be included. Students will be encouraged to investigate and test various topologies for IoT device deployment and to design and test data communication protocols for efficient communication bandwidth optimization. Figure 1 shows a prototype where smart devices are leveraged to track and dynamically adjust the power consumption of a smart house. The research topics here include the data analysis method development, data communication, and smart house security. Meanwhile, towards the second direction, students will consider biology-related applications that currently rely on existing, less-connected embedded systems. As one leading example, students will work with graduate students from the PIs' groups to develop a sophisticated and responsive bio-acoustic system for ecology research. The students will begin by gaining

an understanding of the IoT devices, field recording challenges, and recognition algorithms through building a passive acoustic recording system. Next, the students will develop an active prototype with high computation power so that the developed system can recognize and extract bioacoustics signals from background noise.

Project #11: Security and Privacy of the Communication Channels among IoT Devices

Figure 3: Dual channel validation to defend against fake access point attack



Research: One major reason why IoT devices become increasingly popular is that they enable users to remotely control other IoT devices, or remotely obtain valuable information. Thus the data communication channel between IoT devices and data collectors, or between IoT devices and the Internet, is the critical component for all IoT applications. However, because of the limited computing, storage or battery resources of most IoT devices, and because of the inherent insecurity of wireless communication channels (WiFi, ZigBee, Bluetooth, etc.), the security and privacy of IoT communication channels is a major concern for users and could slow down the success deployment of IoT applications. In this project, we will educate students with practical skills and cutting-edge research techniques on security issues associated with IoT device communication channels, especially WiFi, since it is the dominant communication mechanism for most IoT devices. In Dr. Zou's lab, several easy-to-deploy prototypes have been developed to conduct and defend against fake WiFi Access Point attack (see Figure 3), **Evil Twin attack**, disassociation attack, and parallel on-line password dictionary attack. Students will have hands-on experience in setting up these prototypes, learning how to initiate those well-known WiFi attacks and more importantly, how to defend IoT devices against those attacks accordingly. By learning and having hands-on experience in related research work, the students will be guided to conduct their own research in proposing and brainstorming research ideas on defending against attacks in current and future IoT systems. Different IoT devices have very distinctive data-link layer traffic patterns. Even though WiFi traffic of most IoT devices could be encrypted, the encryption happens on the network layer or higher. Thus by eavesdropping WiFi traffic, attackers can figure out what IoT devices (even brand names) a home or a hotspot has, which poses a serious privacy concern for consumers. In this project, students will first learn how to conduct research on using suitable statistical data analysis tools to fingerprint IoT devices, and

learn how to conduct MAC address randomization, and propose new mechanisms, to defend against this fingerprinting attack.

Securing Embedded Software

Today, we don't have security analysis that can find rounding errors and other critical bugs. Not a big deal for a webserver, but a huge deal for IoT devices found in cars, automobiles, and planes. We are research and developing new tools to help identify such problems and releasing our tools as open source is github.

Toward a Smarthome IoT Infrastructure Free of Privacy Leaks and Software Vulnerabilities

The software components of home IoT ecosystems are often constructed from off-the-shelf components, like messaging libraries. This project examines to what extent use of such components introduces risks to users' data, including due improper configuration of and vulnerabilities in the off-the-shelf components. The project will both empirically quantify these risks and develop tools and guidance to avoid them.

IoT Device Privacy and Security Nutrition Labels

Our ongoing project is to design a privacy and security label for IoT devices based on inputs from privacy and security experts and follow-up user studies with average consumers. The label, which may appear online or on physical product packaging, will contain information about key privacy and security factors such as the types of data the device is collecting, the purpose of data collection, who the data is being shared with, the security update lifetime, the average response time to patch a found vulnerability, and whether or not the device is receiving signed and critical automatic updates. The labels are designed to educate consumers to make more informed IoT-related purchase decisions, allow for product comparisons based on privacy and security properties, and promote device manufacturer accountability for privacy and security.

Wireless Physical Layer Security

This project aims to develop an authentication paradigm for extremely low-power wireless devices, the vast majority of connected devices in the Internet of Things. We aim to do so by learning unique hardware-specific imperfections that these radios inevitably manifest in the signals they transmit. The project aims to leverage these imperfections to tackle a wide range of security and privacy challenges in low-power networks.

Third-Party Network Traffic Attribution and Cross-Device User Tracking for IoT and Web

Not long after the invention of the cookie, online advertisers realized technical features of the web could facilitate fine-grained observation of user behavior. By the time mobile “apps” debuted there was an extensive industry ready and willing to apply web tracking techniques to a new medium. The purpose of this study is to investigate if tracking techniques from web and mobile are migrating into the IoT space. We will seek to determine what third-parties, if any, are capable of tracking users across web, mobile, and IoT, and what policies, if any, regulate the collection and processing of IoT data.

Privacy-preserving Inference and Decision-Making with IoT Data

In the age of Internet-of-things (IoT) and edge computing, various data-collection mechanisms are constantly collecting rich data about the environment. While these data are an essential component of smart decision-making and inference, they can reveal sensitive information about individuals and violate their privacy. Commercial adoption of data analytics systems will be constrained by these privacy issues, in both regulatory domain (e.g., the EU General Data Protection Regulation, GDPR) and from the users' trust perspective. This project aims to enable statistical inference and learning systems without compromising individual privacy. To achieve this goal we will pursue: i) the design of data-collection mechanisms that protect individual privacy, while still providing useful information about the system as a whole; ii) provably optimal techniques to combine information collected from heterogeneous sources; iii) algorithms to sequentially obtain measurements in order to minimize the cost of data collection.

Privacy Preserving Data Analytics using Secure Multi-Party Computation

IoT devices collect a significant amount of data and this is expected to go up even further. There is a need to develop data analytics techniques which can respect the privacy constraints. In this project, we will investigate privacy preserving data analytics by using the cryptographic technique of secure multi-party computation (MPC).

Flipping the Cloud: Managing and Protecting IoT Interactions among Mutually Distrusting Stakeholders at the Network Edge

The future of IoT software will involve complicated interactions among multiple stakeholders, including software developers, hardware manufacturers, infrastructure providers, network operators, users, and regulators, many of which involve user data. However, according to many current and future visions, users have very little control over the data they create. This project focus on how to change this by allowing users to directly influence or even control the data that relates to them, including data ownership, privacy, and sharing; at the same time, our approach can maintain value provided to other stakeholders. Our approach builds on core technologies of edge computing, hardware security, and regulation of information.

IoTHub for Managing and Securing Devices in the Home

We are building an IoTHub that will make it easy for everyday consumers to manage and secure IoT devices in the context of homes. Think of this hub like a smart WiFi router for IoT devices, which also offers services and functionality to help with managing and securing IoT devices, especially low-end devices that have minimal computational and networking capabilities.

Do-It-Yourself-Locally: An IoT Architecture for Localized Data Control for Privacy and Security

Most IoT devices these days come vertically integrated with the manufacturers proprietary backend services, raising serious privacy concerns since users have to cede complete control over their sensitive data and implicitly trust the manufacturer without much transparency on how their data will be used. To protect users' privacy without compromising the functionality of the current IoT ecosystems, we propose a new clean-slate IoT architecture -- DIYL -- that safely extends a local IoT hub's data control to generic cloud platforms. In DIYL, IoT apps execute either locally or are securely offloaded to a DIYL supported generic cloud platform (e.g. Amazon AWS) using primitives that provide secure execution and data privacy, completely in the user's control.

Analysis of Security-Relevant Configuration Options in IoT Infrastructure

The objective of this project is to secure modern cyber-physical systems and internet of things (IoT) devices that are built on layers of reusable software components and infrastructure by understanding, modeling, and offering decision support regarding the impact of configuration options and their interactions on the functionality, performance, energy consumption, and attack surface of the system.

Crowdsourced Smart Cities

The vision of applying computing and communication technologies to enhance life in our cities is fundamentally appealing. Pervasive sensing and computing can alert us to imminent dangers, particularly with respect to the movement of vehicles and pedestrians in and around crowded streets. Signaling systems can integrate knowledge of city-scale traffic congestion. Self-driving vehicles can borrow from and contribute to a city-scale information collaborative. Achieving this vision will require significant coordination among the creators of sensors, actuators, and application-level software systems.

Cities will invest in such smart infrastructure if and only if they are convinced that the value can be realized. Investment by technology providers in creation of the infrastructure depends to a large degree on their belief in a broad and ready market. To accelerate innovation, this stalemate must be broken.

The Usable Privacy Policy Project-Towards Effective Web Privacy Notice and Choice

Natural language privacy policies have become the de facto standard to address expectations of “notice and choice” on the Web. However, users generally do not read these policies and those who do struggle to understand them. Initiatives, such as P3P and Do Not Track aimed to address this problem by developing machine-readable formats to convey a website's data practices. However, many website operators are reluctant to embrace such approaches.

In the Usable Privacy Policy Project, we build on recent advances in natural language processing (NLP), privacy preference modeling, crowdsourcing, and privacy interface design in order to develop a practical framework based on a website's existing natural language privacy policy that empowers users to more meaningfully control their privacy, without requiring additional cooperation from website operators.

The Personalized Privacy Assistant Project

The Internet of Things (IoT) and Big Data are making it impractical for people to keep up with the many different ways in which their data can potentially be collected and processed. What is needed is a new, more scalable paradigm that empowers users to regain appropriate control over their data. We envision personalized privacy assistants as intelligent agents capable of learning the privacy preferences of their users over time, semi-automatically configuring many settings, and making many privacy decisions on their behalf. Through targeted interactions, privacy assistants will help their users better appreciate the ramifications associated with the processing of their data, and empower them to control such processing in an intuitive and effective manner. This includes selectively alerting users about practices they may not feel comfortable with, confirming with users privacy settings the assistants are not sure how to configure, refining models of their user’s preferences over time, and occasionally nudging users to carefully (re)consider the implications of some of their privacy decisions. Ultimately, these assistants will learn our preferences and help us more effectively manage our privacy settings across a wide range of devices and environments without the need for frequent interruptions.

The Internet of Things (IoT) Privacy Infrastructure (Patent Pending)

With the emergence of IoT and a data-centric economy, where a growing number of products, services and business processes rely on the collection and processing of user data, people are increasingly confronted to an unmanageable number of privacy decisions. While there is ample evidence that people care about their privacy, research shows that they are simply overwhelmed by the amount of information they would have to read and settings they are expected to configure. Our team has been developing and piloting personalized privacy assistants, namely intelligent assistants capable of learning the privacy preferences of their users over time to selectively inform them about data collection and use practices they would want to know about and to help them discover and configure available settings.

This involves the development of an infrastructure and protocols to help privacy assistants discover relevant IoT resources, relevant elements of their privacy policies and any available privacy settings. This also includes developing models of people's privacy preferences and expectations, including notification preferences as they pertain to a growing collection of IoT scenarios.

GymCam: Detecting, Recognizing and Tracking Simultaneous Exercises in Unconstrained Scenes

Worn sensors are popular for automatically tracking exercises. However, a wearable is usually attached to one part of the body, tracks only that location, and thus is inadequate for capturing a wide range of exercises, especially when other limbs are involved. Cameras, on the other hand, can fully track a user's body, but suffer from noise and occlusion. We present GymCam, a camera-based system for automatically detecting, recognizing and tracking multiple people and exercises simultaneously in unconstrained environments without any user intervention. We collected data in a varsity gym, correctly segmenting exercises from other activities with an accuracy of 84.6%, recognizing the type of exercise at 93.6% accuracy, and counting the number of repetitions to within ± 1.7 on average. GymCam advances the field of real-time exercise tracking by filling some crucial gaps, such as tracking whole body motion, handling occlusion, and enabling single-point sensing for a multitude of users

Ubicoustics

Despite sound being a rich source of information, computing devices with microphones do not leverage audio to glean useful insights about their physical and social context. For example, a smart speaker sitting on a kitchen countertop cannot figure out if it is in a kitchen, let alone know what a user is doing in a kitchen—a missed opportunity. In this work, we describe a novel, real-time, sound-based activity recognition system. We start by taking an existing, state-of-the-art sound labeling model, which we then tune to classes of interest by drawing data from professional sound effect libraries traditionally used in the entertainment industry. These well-labeled and high-quality sounds are the perfect atomic unit for data augmentation, including amplitude, reverb, and mixing, allowing us to exponentially grow our tuning data in realistic ways. We quantify the performance of our approach across a range of environments and device categories and show that microphone-equipped computing devices already have the requisite capability to unlock real-time activity recognition comparable to human accuracy.

<u>IoT Project Ideas for Beginners</u>

1. Led Blinking Project

Project Idea – The basic project for beginners is that they can play around with led lights with Python. We can automatically turn on and off a series of led lights for better visual effects.

2. Motor Speed Control

Project Idea – Another project for a beginner is that they can control the DC motor speed and direction using raspberry pi. You can control the speed of a fan according to the weather outside. This will also improve your knowledge of robotics.

3. Weather Reporting System using IoT

Project Idea – A weather reporting system is simple in which we use sensors to measure the temperature, humidity, and rain. They will display the measures in real-time. You can also remotely send this information anywhere.

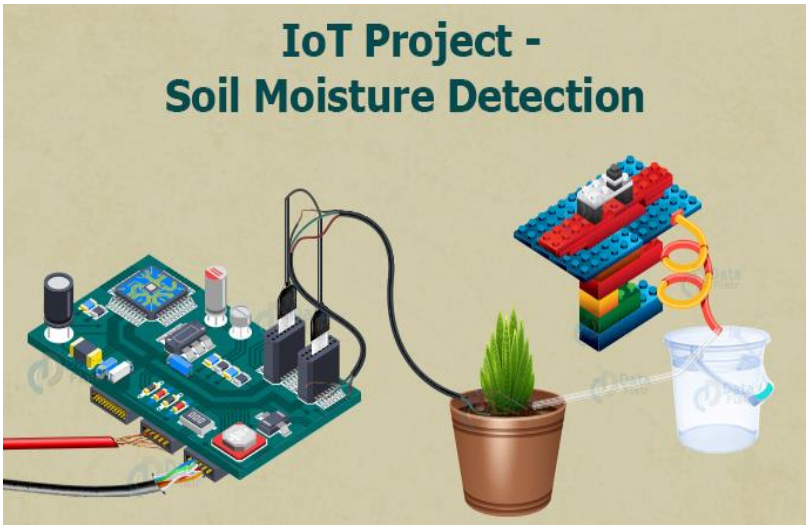
4. IoT Based Liquid Level Monitoring System

Project Idea – By monitoring the liquid level of a system you can have many applications like alerting when a liquid tank is full. We can build a smart system that can fill a tank with liquid without overflowing.

5. Led Light Lamp

Project Idea – You can build your own IoT based light lamp. By using an app from your phone you can remotely control the light intensity and change the color to soothe your eyes. It should support multiple colors.

6. Soil Moisture Detection



Project Idea – If you keep forgetting to water your plants then don't worry you can automate the watering process with Python. Soil moisture sensors can be used to detect the moisture in the soil and you can then water the plants according to the soil moisture.

7. Fire Detection System

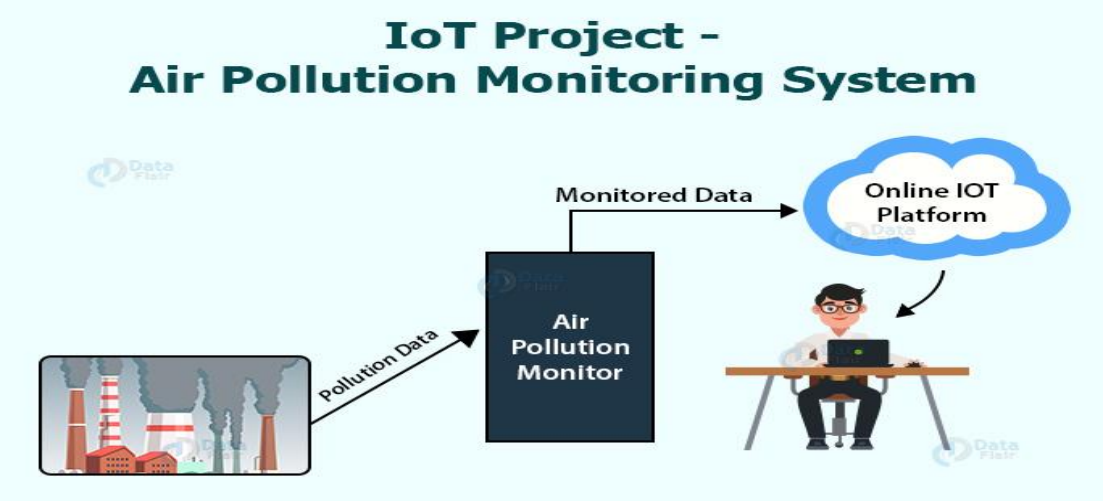
Project Idea – Fire spreads at a quick rate and it is necessary to take action as soon as possible. You can build a fire detection system that will alert the owner of the building, home, etc and will also report directly to the fire department so that immediate actions can be taken.

Intermediate IoT Project Ideas

1. Remote Control Car

Project Idea – You can build your own remote-controlled car. The car can turn around, move forward and backward. You can remotely access the car and drive around through sensors and signals.

2. IoT Based Air Pollution Monitoring System



Project Idea – An air pollution monitoring system is a great project to help monitor the different particles present in air like led, carbon dioxide, sulphur dioxide, etc that are responsible for air pollution. So you can monitor and store all the data on the web servers to check the pollution statistics remotely.

3. IoT Based Smart Mirror

Project Idea – Raspberry pi can be used to make a smart mirror. This mirror not only shows your face but can also display real-time, weather, notes, etc. It uses a large display so you can use any old LCD or LED monitor tv and convert it into a smart mirror.

4. Server Room Temperature Monitoring System

Project Idea – Servers are continuously working day and night without any break. It is a great help to have a temperature monitoring system on the server so that we can remotely see the statistics of the server room.

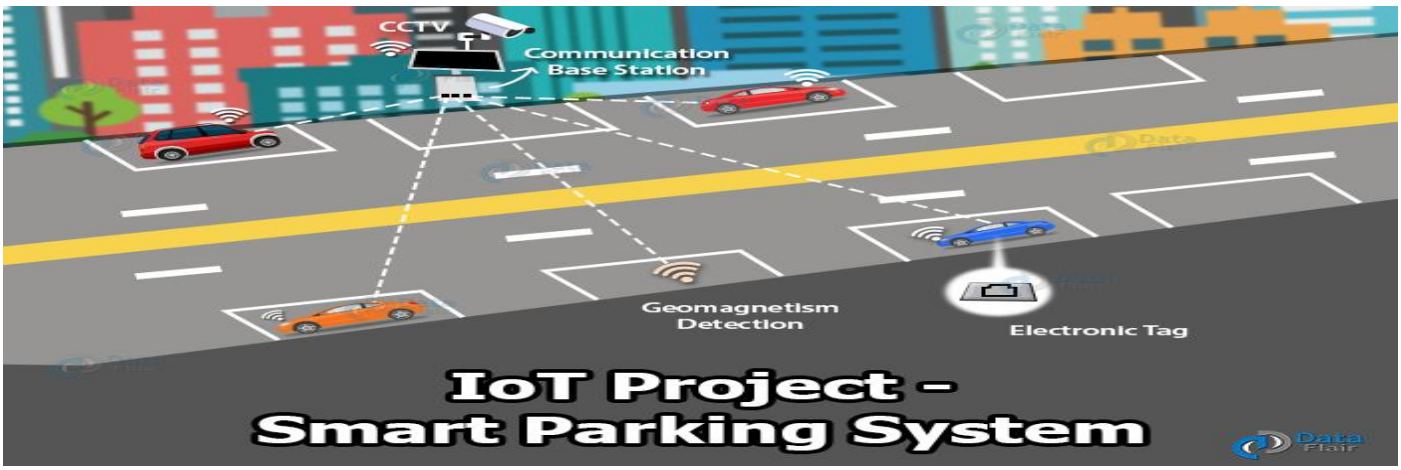
5. Self-Balancing Robot

Project Idea – A self-balancing robot has gyroscopic sensors that are used to balance the robot that does not have the symmetry to stand upright. Many hoverboards are being built using these self-balancing techniques.

6. Automatic Drawing Machine

Project Idea – A nice IoT project is to build an automatic drawing machine in which, through the program, we control the arms of the machine containing a pen. By using a drawing algorithm it will draw the points on the canvas.

7. Smart Parking System using IoT

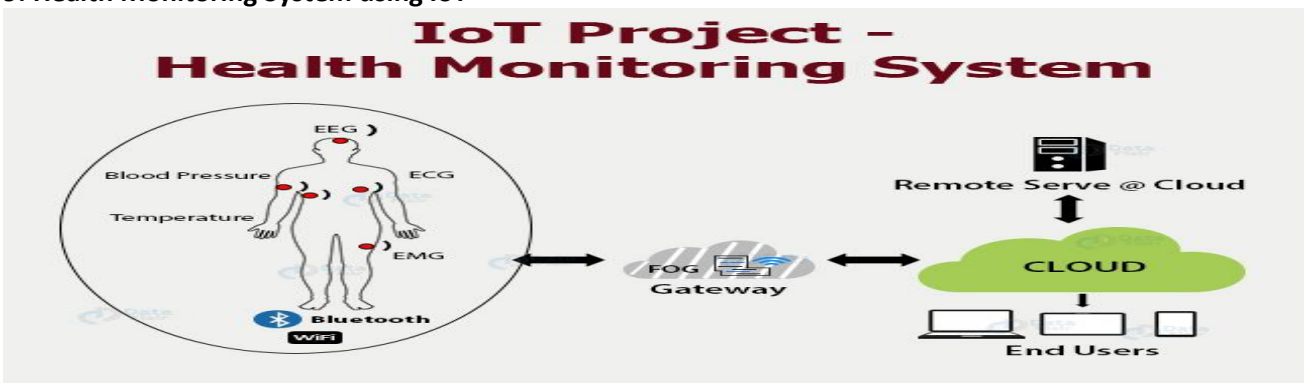


Project Idea – It's hard to find a parking space. A smart parking system is a solution to this problem. To build this we need to fit sensors at each parking space and they will communicate to the communication station that will display which parking space is empty.

8. Surveillance Camera using IoT

Project Idea – You can build your own surveillance cameras using raspberry pi and a camera. This can be used to monitor the baby or children's room and you can also monitor the home from your smartphones with just a few taps. It is a lot cheaper than the market price of surveillance cameras.

9. Health Monitoring System using IoT



Project Idea – The health monitoring system consists of several small sensors located around your body to continuously measure the statistics of the body by calculating ECG, EEG, blood pressure, temperature, etc. This information can be sent and stored on the cloud.

Advanced IoT Project Ideas

1. IoT Based Home Automation System

Project Idea – The home automation system is a big project. You can automate most of your home appliances like fans, lights, tv, door, music system, etc. You can send signals from your smartphone and control all the devices remotely.

2. A robot that can "see"

Project Idea – We can build a robot that can move and also detect the objects coming on its way by using deep learning object detection techniques. For this, we will need to install **TensorFlow** on a raspberry pi model which will take input from the camera and perform detection on the image.

3. Night Patrolling Robot

Project Idea – The night patrolling robot is an automated robot car that has a night vision camera and it contains sensors to detect noises. We can use the camera to detect any human faces and sounds to report back to the user.

As most of the theft cases occur at night, we can set a predefined path for the robot to cover all the angles of the neighbourhood.

4. Facial Recognition Door Unlock

Project Idea – A good idea project is to make a door locking system that will only open when the authorized person tries to open the door. You need to implement a facial recognition system in Python and then if the person exists in the database then we give him entry inside the door.

5. IoT Smart City

Project Idea – The internet of things can be used to build a smart city in which all the places in a smart city are interconnected with each other with IoT components for efficient usage of resources. It can have tracking abilities, capturing air pollution data, traffic management, and parking systems, smart waste system, automatic water supply in houses and gardens, etc. There is no limit to the things we can include in a smart city.

6. Automatic Coffee Maker

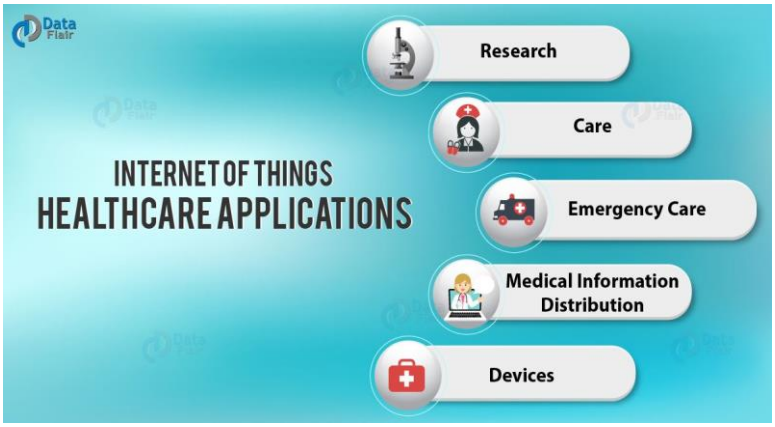
Project Idea – Connect the coffee maker with the internet by using raspberry pi and then you can remotely turn on the coffee maker anytime, set the time and also turn off the coffee maker. Then you will get your coffee ready when you reach home.

7. Raspberry pi Drone

Project Idea – Drones are useful in a lot of ways. They can carry small packages and be controlled from a long distance. Drones are used a lot in making cinematic videos and photography. Use your ideas to build a drone using raspberry pi and Python.

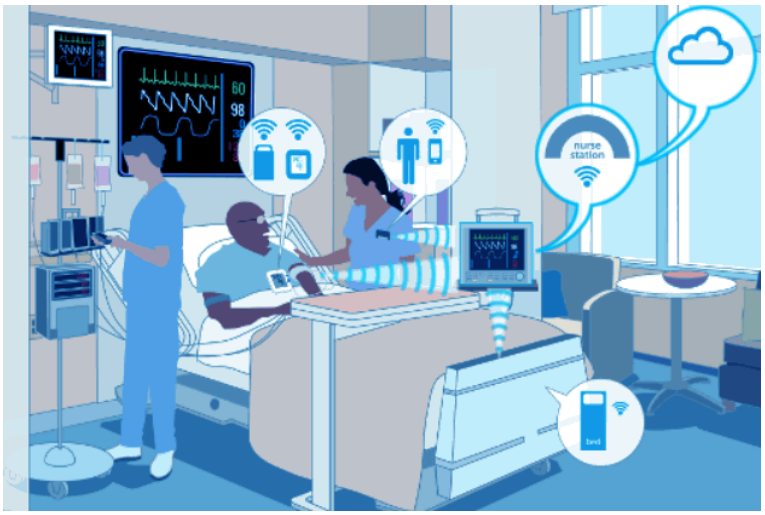
Healthcare IOT projects

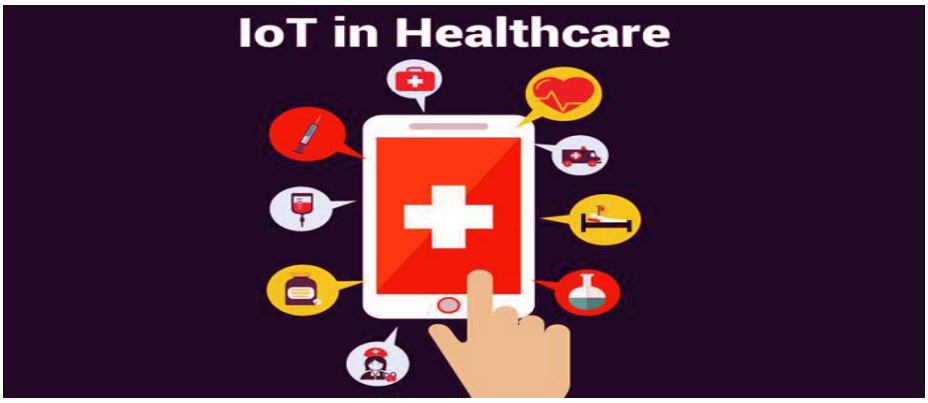
1. IoT in healthcare and IoT Application.



2. IoT Applications in Healthcare

The current technology in healthcare and a general practice of medicine gets enhanced with the IoT system. Professionals reach is expanding within a facility. The diverse data collected from large sets of real-world cases increases both the accuracy and size of medical data. The precision of medical care delivery is also improved by incorporating more sophisticated technologies in the healthcare system.



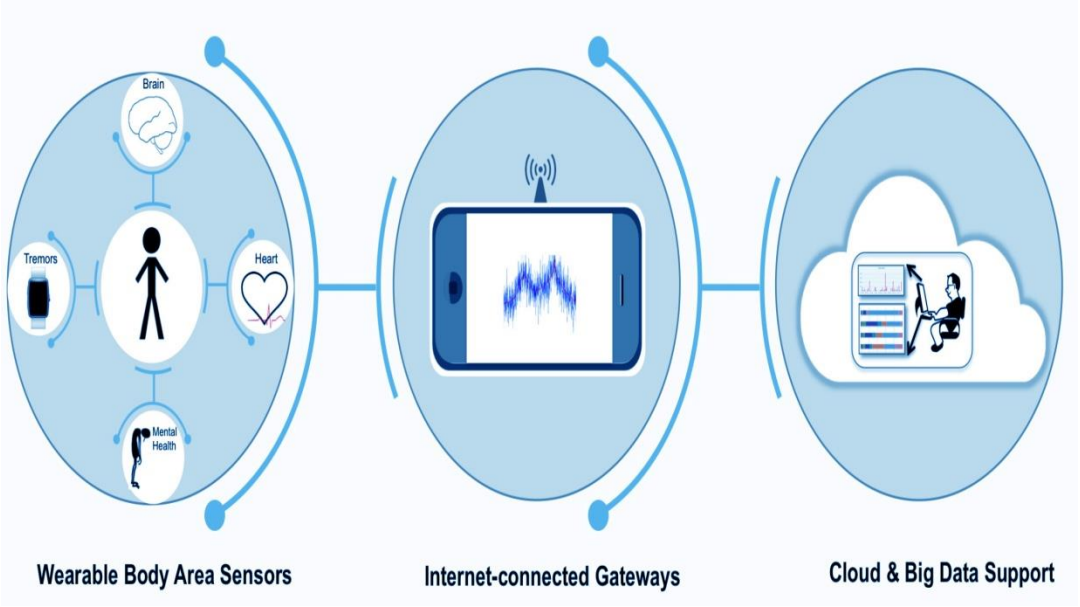


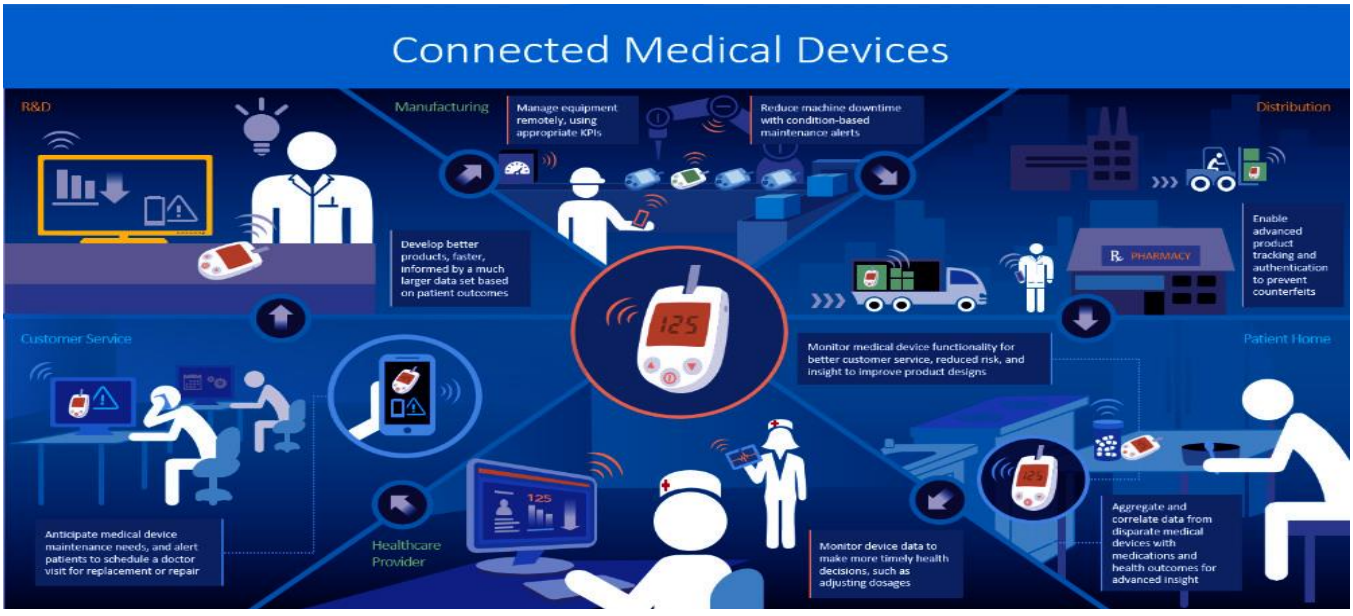
a. Research

The resources that current medical research uses lack critical real-world information. It mostly uses leftovers, controlled environments and volunteers for medical examination. IoT opens ways to a sea of valuable data and information through analysis, real-time field data, and testing.
IoT can deliver data that is far superior to standard analytics through making use of instruments that are capable of performing potential research. As a result, IoT helps in healthcare by providing more practical and reliable data, which yields better solutions and discovery of issues that were previously unknown, that's why research is one of the most important IoT applications in healthcare.

b. Devices

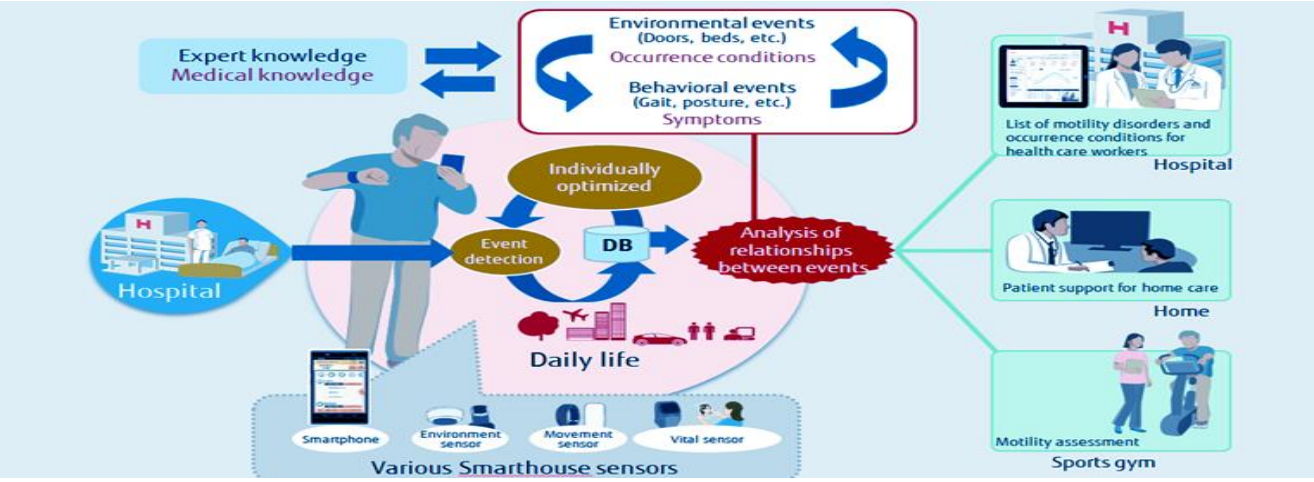
Even current devices are improving in their power, precision, and availability; they still offer fewer benefits and qualities that an IoT system offers. IoT has the potential to unlock existing technology, and lead us towards better healthcare and medical device solutions.
IoT tries and fills gaps between the way we deliver healthcare and the equipment by creating a system rather than just tools. It then detects flaws and reveals patterns and missing elements in healthcare and suggests improvements.





c. Care

IoT empowers healthcare professionals to use their knowledge and training in a better way to solve problems. It helps them utilize better data and equipment that in turn supports more precise and swift actions. IoT allows in the professional development of healthcare professionals because they practically exercise their talent rather than spending time on administrative tasks.



d. Medical Information Distribution

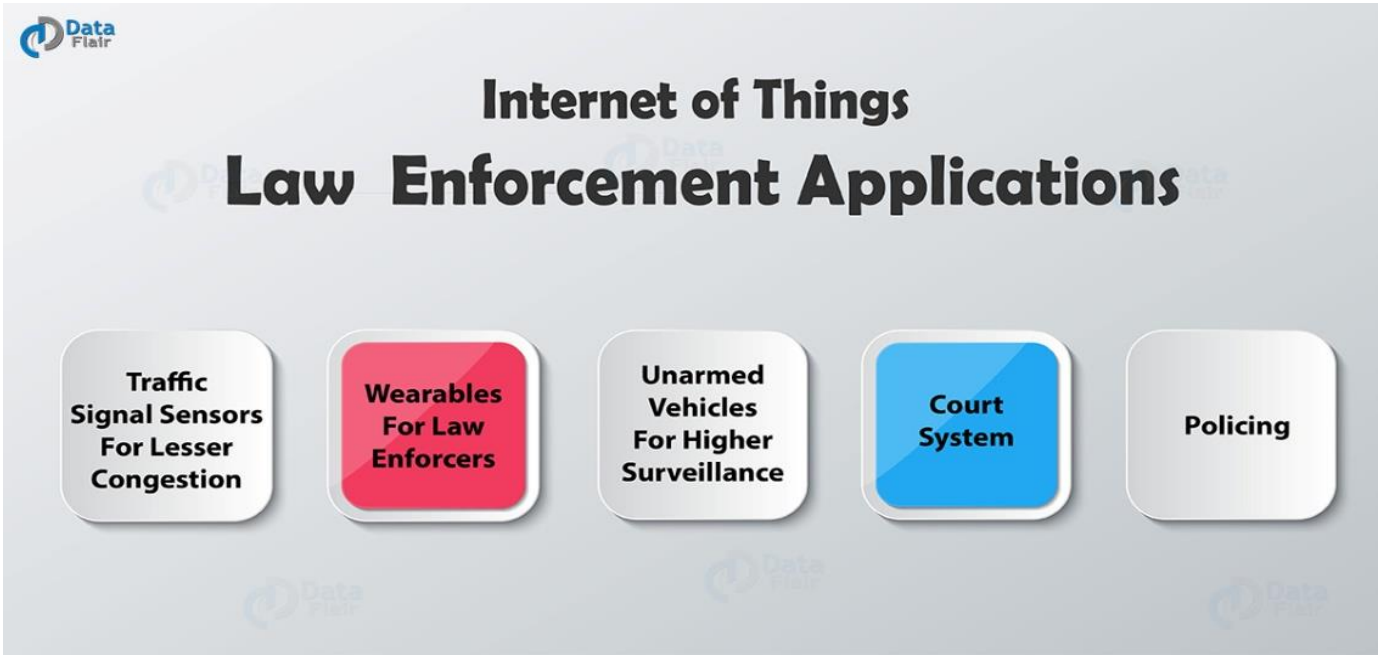
This is a most prominent innovation of IoT applications in healthcare, the distribution of accurate and current information to patients remains one of the most challenging concerns of medical care. IoT devices not only improve health in the daily lives of individuals but also facilities and professional practice.
IoT systems take healthcare out of facilities like hospitals and allow intrusive care into the office, home or social space. They empower and enable individuals to cater to their own health, and allow healthcare providers to deliver better care to patients. As a result, this has resulted and paved way for fewer accidents that usually result from miscommunication, improved patient satisfaction, and better preventive care.

e. Emergency Care

The emergency support services have always had the problem of suffering from limited resources and getting disconnected with the base facility. The advanced automation and analytics of IoT cater to this problem in the healthcare sector. An emergency can be analyzed from a far distance or rather miles away. The providers get access to the patient profiles way before their arrival because of which they can deliver essential care to the patients on time. In this way, associated losses are reduced, and emergency health care is improved.

IoT Law Enforcement Applications – Internet of Things Safety

1. **IoT Law Enforcement Applications.**



2. **IoT Law Enforcement Applications**

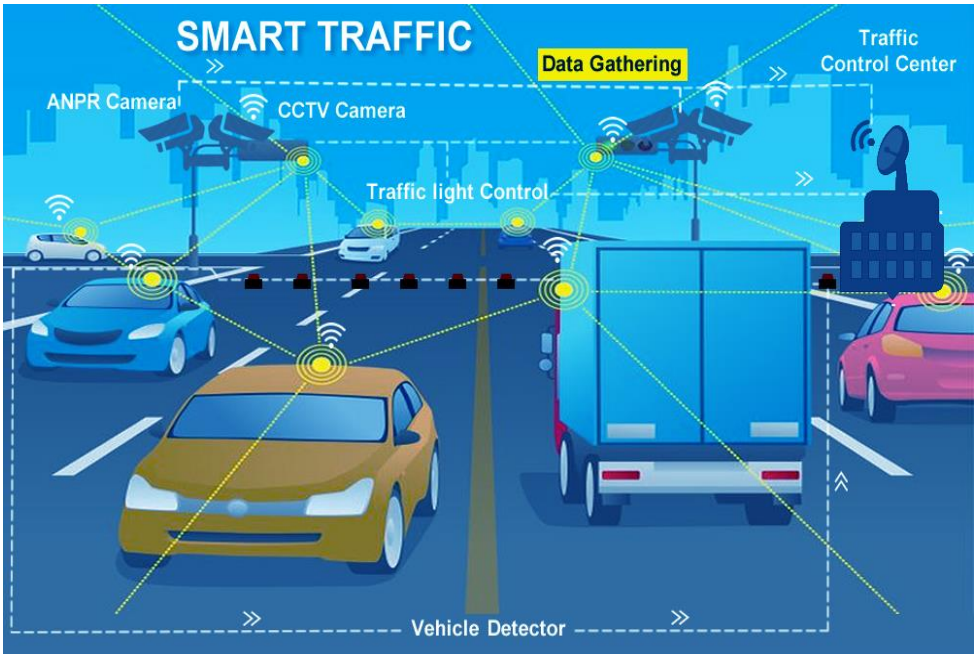
To improve human existence, regulation enforcement in an area that plays a critical function in securing people and making sure that they're under protection. Policing is critical because criminal charges are increasing throughout the globe. In their assignment to limit crime rates, law enforcement government is leveraging technologies to make sure that their employees are upgrading themselves with technology to be better capable to perform their duties. No

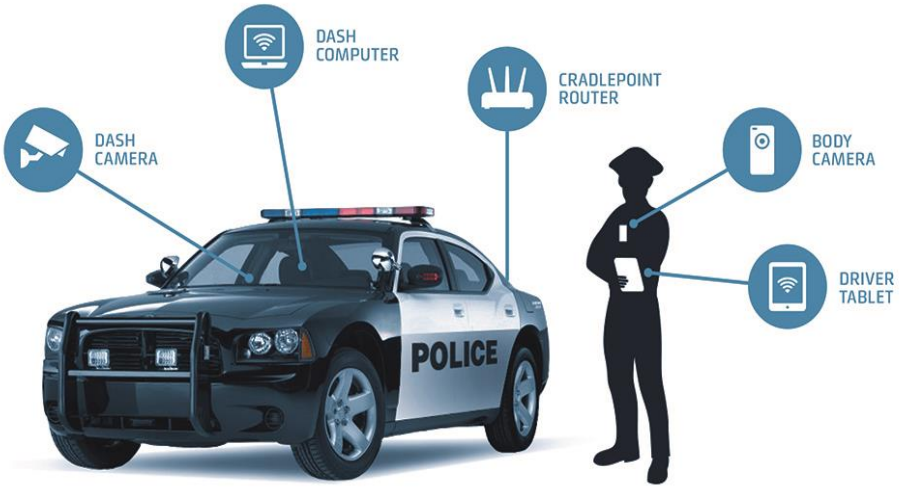
wonder, packages of IoT in law enforcement are being searched to enhance the current repute of law enforcement government.

IoT being a dynamic era that has brought numerous ameliorations at some point in several industries, is now reshaping the field of law enforcement. Right here are some applications of IoT in law enforcement that could assist human beings and authorities to improve the cutting-edge repute of law enforcement.

a. Traffic single sensors for lesser congestion

Smart site visitor signals are becoming a fashion across numerous international locations and have proved useful in reducing injuries and improving site visitor management. According to a research, 30% of accidents are caused because of loss of parking spaces. This congestion no longer only results in delays in human beings reaching their destination but also effects on fuel wastage. With IoT, traffic indicators rework into clever indicators which can help people and criminal authorities in a plethora of approaches consisting of notifying authorities about congestion in a specific area and what precautionary measures are required. With clever sensors, parking woes reduce rather as those devices do now not want batteries that demand to be modified after a few months and can continue operating on the energy they have for several years. Also, these sensors can notify the customers of any available parking spots too.



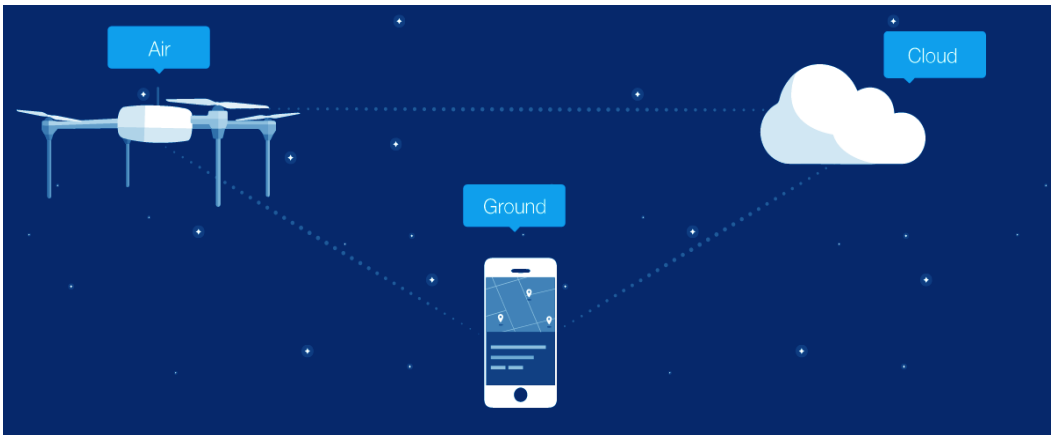


b. Wearables for law enforcers

Currently, smart wearable took into consideration to be the most up to date developments amongst human beings. With the elevated utilization of clever devices, law enforcement authorities realized their importance in monitoring crimes and decreasing their frequency. With the availability of smartwatches, a government can right away talk with their server rooms. The smartwatches additionally consist of a pedometer, a heart charge sensor, and other equipment to decide if an officer has received adequate sleep and maintained stress tiers.

c. Unarmed vehicles for higher surveillance

There are situations where physically reaching an area becomes tough because the areas can also pose a risk to the lives of government authorities. To counter such circumstances and gain higher surveillance, IoT proves to be useful. Drones can assist authorities to reach a remote vicinity without being present there physically.



Drones can hover above a specific vicinity that requires surveillance and behave as an additional pair of eyes when equipped with distinctive styles of cameras. The utility of cameras make tracking smoothly for authorities and ensure that crime fee of their region subsides. Other than cameras, drones have several sensors that help law enforcement authorities to improve surveillance.

Regulation enforcement groups have to consciously focus on locating methods via which they could teach their personnel in accepting and leveraging those technologies into their challenge to make certain things secure and productive.

d. Court system

Present day courtroom systems make use of traditional generation and assets. They usually do no longer exploit contemporary analytics or automation outside of minor legal obligations. IoT brings advanced analytics, better evidence, and optimized processes to court systems which boost up tactics, take away immoderate processes, manage corruption, reduce expenses, and improve pride.

Within the crook courtroom device, this could result in an extra effective and truthful system. In habitual court offerings, it introduces automation similar to that of commonplace authorities office offerings; for example, IoT can automate forming an LLC.

IoT mixed with new rules can get rid of lawyers from many commonplace criminal tasks or reduce the want for their involvement. This reduces costs and hurries up many procedures which frequently require months of traversing criminal procedures and forms.

e. Policing

Law enforcement can be hard. IoT acts as a device of regulation enforcement that helps reduce exertions and decisions on people through better information, statistics sharing, and superior automation. IoT systems shave expenses with the aid of reducing human labor in certain regions along with positive traffic violations.

IoT aids in developing better answers to problems by way of the usage of generation in the area of pressure. For instance, mild in-man or woman investigations of suspicious activities may be replaced with a far-flung statement, logged footage of violations, and digital ticketing. It also reduces corruption with the aid of casting off human control and opinion for a few violations.

Machine Learning Tools

Machine learning is an astonishing technology, if you use it in a correct way. How fascinating it would be to build a machine that behaves like a human being to a great extent. Mastering machine learning tools will let you play with the data, train your models, discover new methods, and create your own algorithms.

Machine learning comes with an extensive collection of ML tools, platforms, and software. Moreover, ML technology is evolving continuously. Out of a pile of machine learning tools, you need to choose any of them to gain expertise.

1. Knime

Knime is an open-source machine learning tool that is based on GUI. The best thing about Knime is, it doesn't require any knowledge of programming. One can still avail of the facilities provided by Knime. It is generally used for data relevant purposes. For example, data manipulation, data mining, etc.

Moreover, it processes data by creating different various workflows and then execute them. It comes with repositories that are full of different nodes. These nodes are then brought into the Knime portal. And finally, a workflow of nodes is created and executed.

2. Accord.net

Accord.net is a computational machine learning framework. It comes with an image as well as audio packages. Such packages assist in training the models and in creating interactive applications. For example, audition, computer vision, etc.As .net is present in the name of the tool, the base library of this framework is C# language. Accord libraries are very much useful in testing as well as manipulating audio files.

3. Scikit-Learn

Scikit-Learn is an open-source machine learning package. It is a unified platform as it is used for multiple purposes. It assists in regression, clustering, classification, dimensionality reduction, and preprocessing. Scikit-Learn is built on top of the three main Python libraries viz. NumPy, Matplotlib, and SciPy. Along with this, it will also help you with testing as well as training your models.

4. TensorFlow

TensorFlow is an open-source framework that comes in handy for large-scale as well as numerical ML. It is a blender of machine learning as well as neural network models. Moreover, it is also a good friend of Python.The most prominent feature of TensorFlow is, it runs on CPU and GPU as well. Natural language processing, Image classification are the ones who implement this tool.

5. Weka

Welcoming the next ML tool, Weka. It is also open-source software. One can access it through a graphical user interface. The software is very user-friendly. The application of this tool is in research and teaching. Along with this, Weka lets you access other machine learning tools as well. For example, R, Scikit-learn, etc.

6. Pytorch

Pytorch is a deep learning framework. It is very fast as well as flexible to use. This is because Pytorch has a good command over the GPU. It is one of the most important tools of machine learning because it is used in the most vital aspects of ML which includes building deep neural networks and tensor calculations.

Pytorch is completely based on Python. Along with this, it is the best alternative to NumPy.

7. RapidMiner

RapidMiner is a piece of good news for the non-programmers. It is a data science platform and has a very amazing interface. RapidMiner is platform-independent as it works on cross-platform operating systems.

With the help of this tool, one can use their own data as well as test their own models. Its interface is very user-friendly. You only drag and drop. This is the major reason why it is beneficial for non-programmers as well.

8. Google Cloud AutoML

The objective of Google cloud AutoML is to make artificial intelligence accessible to everyone. What Google Cloud AutoML does is, it provides the models which are pre-trained to the users in order to create various services. For example, text recognition, speech recognition, etc.Google Cloud AutoML became very much popular among companies. As the companies want to apply artificial intelligence in every sector of the industry but they have been facing difficulties in doing so because there is a lack of skilled AI persons in the market.

9. Jupyter Notebook

Jupyter notebook is one of the most widely used machine learning tools among all. It is a very fast processing as well as an efficient platform. Moreover, it supports three languages viz. Julia, R, Python.

Thus the name of Jupyter is formed by the combination of these three programming languages. Jupyter Notebook allows the user to store and share the live code in the form of notebooks. One can also access it through a GUI. For example, winpython navigator, anaconda navigator, etc.

10. Apache Mahout

Mahout is launched by Apache which is an open-source platform based on Hadoop. It is generally used for machine learning and data mining. Techniques such as regression, classification, and clustering became possible with Mahout. Along with this, it also makes use of math-based functions such as vectors, etc.

11. Azure machine learning studio

Azure machine learning studio is launched by Microsoft. Just like, Google's Cloud AutoML, this is Microsoft's product which provides machine learning services to the users. Azure machine learning studio is a very easy way to form connections of modules and datasets.

Along with this, Azure also aims to provide AI facilities to the user. Just like TensorFlow, it also works on CPU and GPU.

12. MLLIB

Like Mahout, MLLIB is also a product of Apache Spark. It is used for regression, feature extraction, classification, filtering, etc. It also often called Spark MLLIB. MLLIB comes with very good speed as well as efficiency.

13. Orange3

Orange3 is a data mining software which is the latest version of the Orange software. Orange3 assists in preprocessing, data visualization, and other data-related stuff. One can access Orange3 through the Anaconda Navigator. It is really very helpful in Python programming. Along with this, it can also be a great user interface.

14. IBM Watson

IBM Watson is a web interface that is given by IBM for using Watson. Watson is a human interaction Q and A system which is based on Natural Language processing. Watson is applied in various fields such as automated learning, information extraction, etc.

IBM Watson is generally used for research and testing purposes. Its objective is to offer a human-like experience to the users.

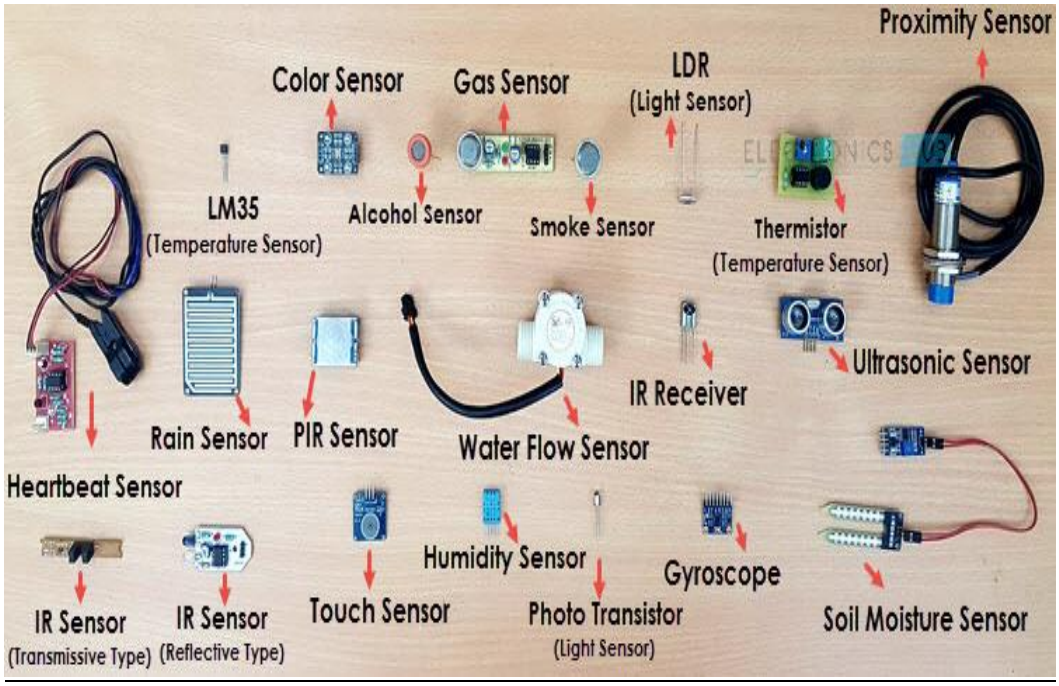
15. Pylearn2

Pylearn2 is a machine learning library that is built on top of Theano. Therefore, there are many functions that are similar between them. Along with this, it can perform math calculations. Pylearn2 is also capable of running on the CPU and GPU as well. Before getting to Pylearn2, you must be familiar with Theano.

IoT Sensor Types

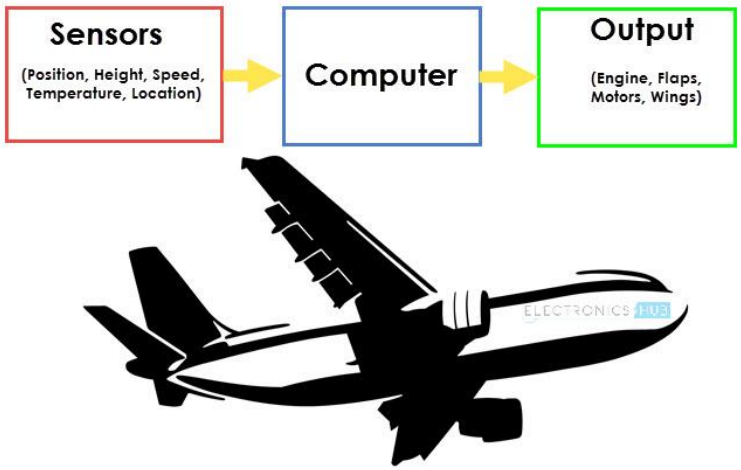
Sensors are everywhere. They're in our homes and workplaces, our shopping centers and hospitals. They're embedded in smart phones and an integral part of the Internet of Things (IoT). Sensors have been around for a long time. The first thermostat was introduced in the late 1880s and infrared sensors have been around since the late 1940s. The IoT and its counterpart, the Industrial Internet of Things (IIoT), are bringing sensor usage to a new level.

Broadly speaking, sensors are devices that detect and respond to changes in an environment. Inputs can come from a variety of sources such as light, temperature, motion and pressure. Sensors output valuable information and if they are connected to a network, they can share data with other connected devices and management systems. Sensors come in many shapes and sizes. Some are purpose-built containing many built-in individual sensors, allowing you to monitor and measure many sources of data.



Real Time Application of Sensors

The example we are talking about here is the Autopilot System in aircrafts. Almost all civilian and military aircrafts have the feature of Automatic Flight Control system or sometimes called as Autopilot.



An Automatic Flight Control System consists of several sensors for various tasks like speed control, height, position, doors, obstacle, fuel, maneuvering and many more. A Computer takes data from all these sensors and processes them by comparing them with pre-designed values.

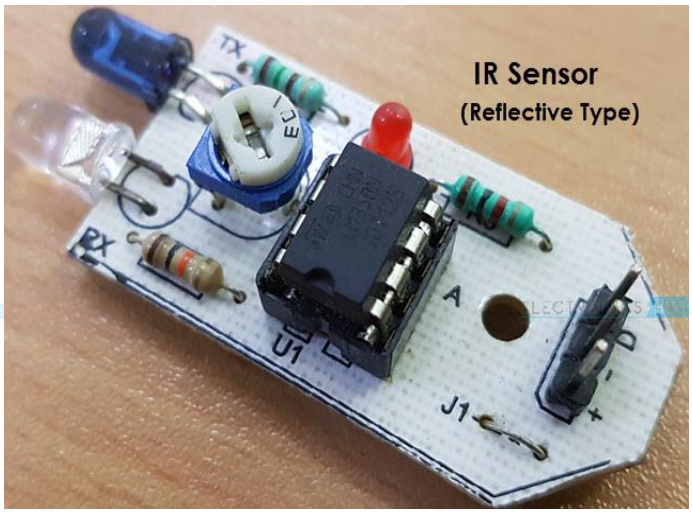
The computer then provides control signal to different parts like engines, flaps, rudders etc. that help in a smooth flight. The combination of Sensors, Computers and Mechanics makes it possible to run the plane in Autopilot Mode.

All the parameters i.e. the Sensors (which give inputs to the Computers), the Computers (the brains of the system) and the mechanics (the outputs of the system like engines and motors) are equally important in building a successful automated system.

Sensors are crucial to the operation of many of today's businesses. They can warn you of potential problems before they become big problems, allowing businesses to perform predictive maintenance and avoid costly downtime. The data from sensors can also be analyzed for trends allowing business owners to gain insight into crucial trends and make informed evidence-based decisions.

How does IoT sensor work?

In IoT applications, sensors are connected to a network (WiFi, LPWAN, cellular, etc.) over which the collected data is transmitted. The destination is usually a cloud-based service where the data is processed.



Ultrasonic Sensor

Temperature Sensors

Temperature sensors measure the amount of heat energy in a source, allowing them to detect temperature changes and convert these changes to data. Machinery used in manufacturing often requires environmental and device temperatures to be at specific levels. Similarly, within agriculture, soil temperature is a key factor for crop growth.

Humidity Sensors

These types of sensors measure the amount of water vapor in the atmosphere of air or other gases. Humidity sensors are commonly found in heating, vents and air conditioning (HVAC) systems in both industrial and residential domains. They can be found in many other areas including hospitals, and meteorology stations to report and predict weather

Pressure Sensors

A pressure sensor senses changes in gases and liquids. When the pressure changes, the sensor detects these changes, and communicates them to connected systems. Common use cases include leak testing which can be a result of decay. Pressure sensors are also useful in the manufacturing of water systems as it is easy to detect fluctuations or drops in pressure.

Proximity Sensors

Proximity sensors are used for non-contact detection of objects near the sensor. These types of sensors often emit electromagnetic fields or beams of radiation such as infrared. Proximity sensors have some interesting use cases. In retail, a proximity sensor can detect the motion between a customer and a product in which he or she is interested. The user can be notified of any discounts or special offers of products located near the sensor. Proximity sensors are also used in the parking lots of malls, stadiums and airports to indicate parking availability. They can also be used on the assembly lines of chemical, food and many other types of industries.

Some of the applications of Proximity Sensors are Mobile Phones, Cars (Parking Sensors), industries (object alignment), Ground Proximity in Aircrafts, etc. Proximity Sensor in Reverse Parking is implemented in this Project: REVERSE PARKING SENSOR CIRCUIT.

Level Sensors

Level sensors are used to detect the level of substances including liquids, powders and granular materials. Many industries including oil manufacturing, water treatment and beverage and food manufacturing factories use level sensors. Waste management systems provide a common use case as level sensors can detect the level of waste in a garbage can or dumpster.

Accelerometers

Accelerometers detect an object's acceleration i.e. the rate of change of the object's velocity with respect to time. Accelerometers can also detect changes to gravity. Use cases for accelerometers include smart pedometers and monitoring driving fleets. They can also be used as anti-theft protection alerting the system if an object that should be stationary is moved.

Gyroscope

Gyroscope sensors measure the angular rate or velocity, often defined as a measurement of speed and rotation around an axis. Use cases include automotive, such as car navigation and electronic stability control (anti-skid) systems. Additional use cases include motion sensing for video games, and camera-shake detection systems.

Gas Sensors

These types of sensors monitor and detect changes in air quality, including the presence of toxic, combustible or hazardous gasses. Industries using gas sensors include mining, oil and gas, chemical research and manufacturing. A common consumer use case is the familiar carbon dioxide detectors used in many homes.

Infrared Sensors

These types of sensors sense characteristics in their surroundings by either emitting or detecting infrared radiation. They can also measure the heat emitted by objects. Infrared sensors are used in a variety of different IoT projects including healthcare as they simplify the monitoring of blood flow and blood pressure. Televisions use infrared sensors to interpret the signals sent from a remote control. Another interesting application is that of art historians using infrared sensors to see hidden layers in paintings to help determine whether a work of art is original or fake or has been altered by a restoration process.

Optical Sensors

Optical sensors convert rays of light into electrical signals. There are many applications and use cases for optical sensors. In the auto industry, vehicles use optical sensors to recognize signs, obstacles, and other things that a driver would notice when driving or parking. Optical sensors play a big role in the development of driverless cars. Optical sensors are very common in smart phones. For example, ambient light sensors can extend battery life. Optical sensors are also used in the biomedical field including breath analysis and heart-rate monitors.

MYTHINGS IoT Sensor

The MYTHINGS Smart Sensor is a self-contained, battery-powered multi-purpose IoT sensor that allows you to capture critical data points like acceleration, temperature, humidity, pressure and GPS. The smart sensor is integrated with the MYTHINGS Library – a hardware independent, small-footprint and power-optimized library of code, featuring the MIOTY (TS-UNB) low-power wide area network protocol

MODULE - 5

IOT physical Devices and Endpoints - Arduino UNO

Introduction to Arduino

Arduino is an open-source advancement prototying platform which depends on simple to-utilize equipment and programming.

Arduino can read inputs - such as detecting the power & light, events triggered by a button or a twitter message and can respond into a yield.

The Arduino is a small computer that you can program to read information from the world around you and to send commands to the outside world.

- Arduino is a tiny computer that you can connect to electrical circuits. This makes it easy to read inputs - and control outputs - Send a command to the outside.

Why Arduino ?

Arduino is an open source product, software/hardware which is accessible and flexible to customers.

Arduino is flexible because of offering variety of digital and analog pins, SPI and PWM outputs.

Arduino is easy to use, connected to computer via a USB and communicates using Serial protocol.

Arduino has growing online community where lots of source code is available for use.

Arduino is Cross-platform, which can work on Windows, Mac or Linux platforms.

Arduino follows simple, clear programming environment as C language.

Which Arduino ?

There are hundreds of "Arduino boards" available in the market serving every kind of purpose. Among all we almost focus on popular Arduino UNO which is used in almost 99% of projects use.

→ Some of the Boards from Arduino family are given below

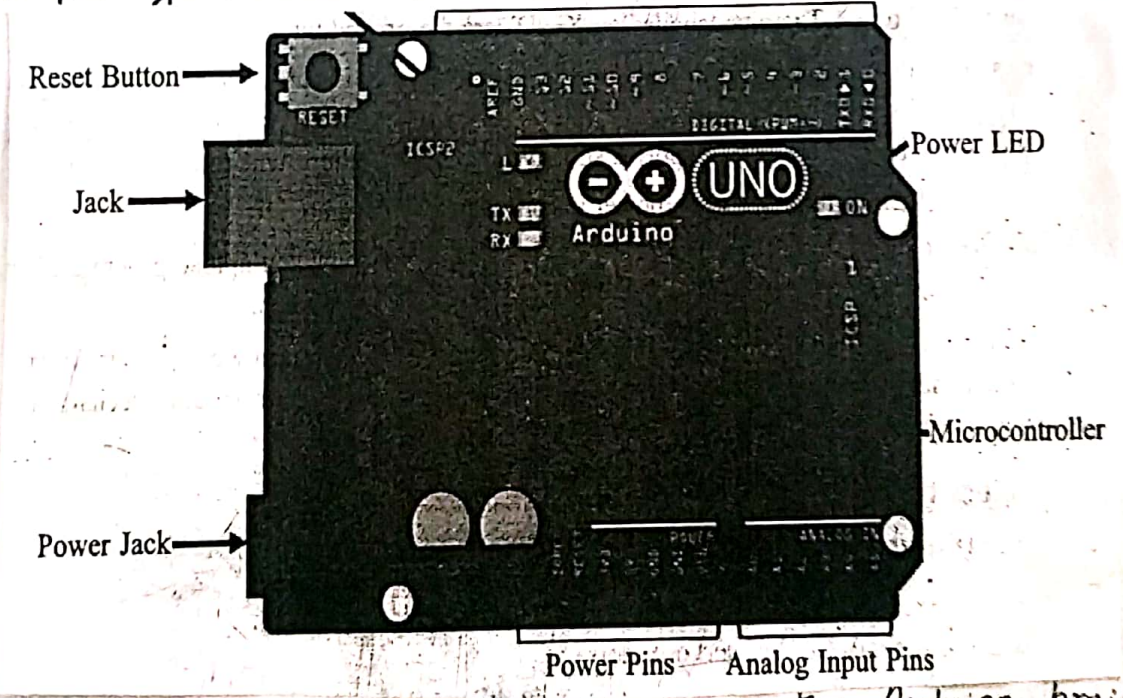
Arduino Mega is a big sister to the UNO with more memory and pins with a different chip the ATmega2560.

Flora is an Arduino compatible from Adafruit which is a round wearable which can be sewed into clothing.

The Arduino MKR1000 is a little like an Arduino Micro but has a more powerful 32-bit ATSAM ARM chip and built-in Wifi.

Arduino Micro is bit smaller with a chip Atmega 32u4 that can act like a keyboard or mouse.

Exploring Arduino UNO Learning Board



* Microcontroller : The ATmega328p is the Arduino brain. Everything on the Arduino board is meant to support this microcontroller.
* Digital pins : Arduino has 14 digital pins, labeled from 0 to 13 that can act as inputs or outputs.

* PWM pins : These are digital pins marked with a ~ (pins 11, 10, 9, 6, 5 and 3). PWM stands for "pulse width modulation" and allows to make digital pins output "fake" varying amounts of voltage.
* TX and RX pins : digital pins 0 and 1. The T stands for "transmit" and the R for "receive".
* LED attached to digital pin 13 : This is useful for an easy debugging of the Arduino sketches.
* Analog pins : The analog pins are labeled from A0 to A5 and are most often used to read analog sensors.
* Power pins : The Aurdino has 3.3V or 5V supply, which is really useful since most components require 3.3V or 5V.
* Reset button : When you press that button, the program that is currently being run in your Arduino will start from the beginning.
* Power ON LED : will be on since power is applied to the Arduino.
* USB jack : Connecting a male USB A to male USB B cable is how you upload programs from your computer to your Arduino board.
* Power jack : The power jack is where you connect a component to power up your Arduino.

Things that Arduino can do

Motion Sensor : It allows you detect movement

Light Sensor : this allows you to "measure" the quantity of light in the outside world.

Humidity and temperature Sensor : this is used to measure the humidity and temperature.

Ultrasonic Sensor : this Sensor allows to determine the distance to an object through Sonar.

Installing the Software (ARDUINO IDE)

The Arduino IDE (Integrated Development Environment) is where you develop your programs that will tell your Arduino what to do.

To download your Arduino IDE, browse on the following link https://www.arduino.cc/en/Main/Software.

Select which Operating System you're using and download it.

Fundamentals of Arduino Programming

1) Structure

The structure of Arduino programming contains of two parts as shown below

```
void setup()
{
  Statement(s);
}
void loop()
{
  Statement();
}
```

2)
```
void setup()
  void loop()
{
  digitalWrite(pin, HIGH);
  delay(10000);
  digitalWrite(pin, LOW);
  delay(10000);
}
```

3) Functions

A function is a piece of code that has a name and set of statements executed when function is called.

Functions are declared by its type followed with name of a function.

Syntax : type functionName (parameters)
{
Statement(s);
}

4) { } Curly braces

They define beginning and end of function.

5) Semicolon

It is used to end a statement and separate elements of a program.

Syntax : int x=14;

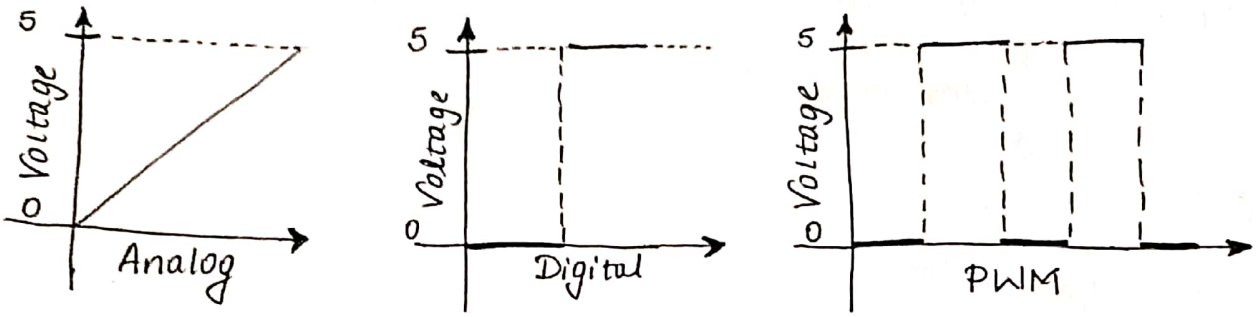
Differences between Analog, Digital and PWM Pins

In analog pins, you have unlimited possible states between 0 and 1023, This allows you to read sensor values for example, with a light sensor, if it is very dark, you'll read 1023, if it is very bright you'll read 0 If there is a brightness between dark and very bright you'll read a value between 0 and 1023.

In digital pins, you have just two possible states, which are on or off. These can also be referred as High or Low, 1 or 0 and 5V or 0V. For example, if an LED is on, then, its state is high or 1 or 5V. If it is off, you'll have Low, or 0 or 0V.

PWM pins are digital pins, so they output either 0 or 5V. However these pins can output "fake" intermedia voltage values between 0 and 5V, because they can perform "Pulse Width Modulation" (PWM). PWM allows to "simulate" varying levels of power by oscillating the output voltage of the Arduino.

The below figure shows the representation of Analog, Digital and PWM pins of Arduino.



IOT Physical Devices and Endpoints : RaspberryPi

Introduction to RaspberryPi

The RaspberryPi is a series of credit card sized single-board computers developed in the United Kingdom by Raspberrypi Foundation to promote the teaching of basic computer science in School and developing Countries.

The original model became far more popular than anticipated, selling outside its target market for uses such as robotics. It does not include peripherals and cases. However, some accessories have been included in several official and unofficial bundles.

The Organisation behind the Raspberry Pi consists of two arms. The first two models were developed by the Raspberry Pi Foundation. After the Pi Model B was released, the Foundation setup Raspberry Pi Trading, with Eben Upton as CEO, to develop the third model the B+.

"Why Raspberry Pi?" - Inexpensive, Cross-platform, Simple, clear programming environment, Open Source and extensible Software and Open Source and extensible hardware.

Exploring The Raspberrypi Learning Board

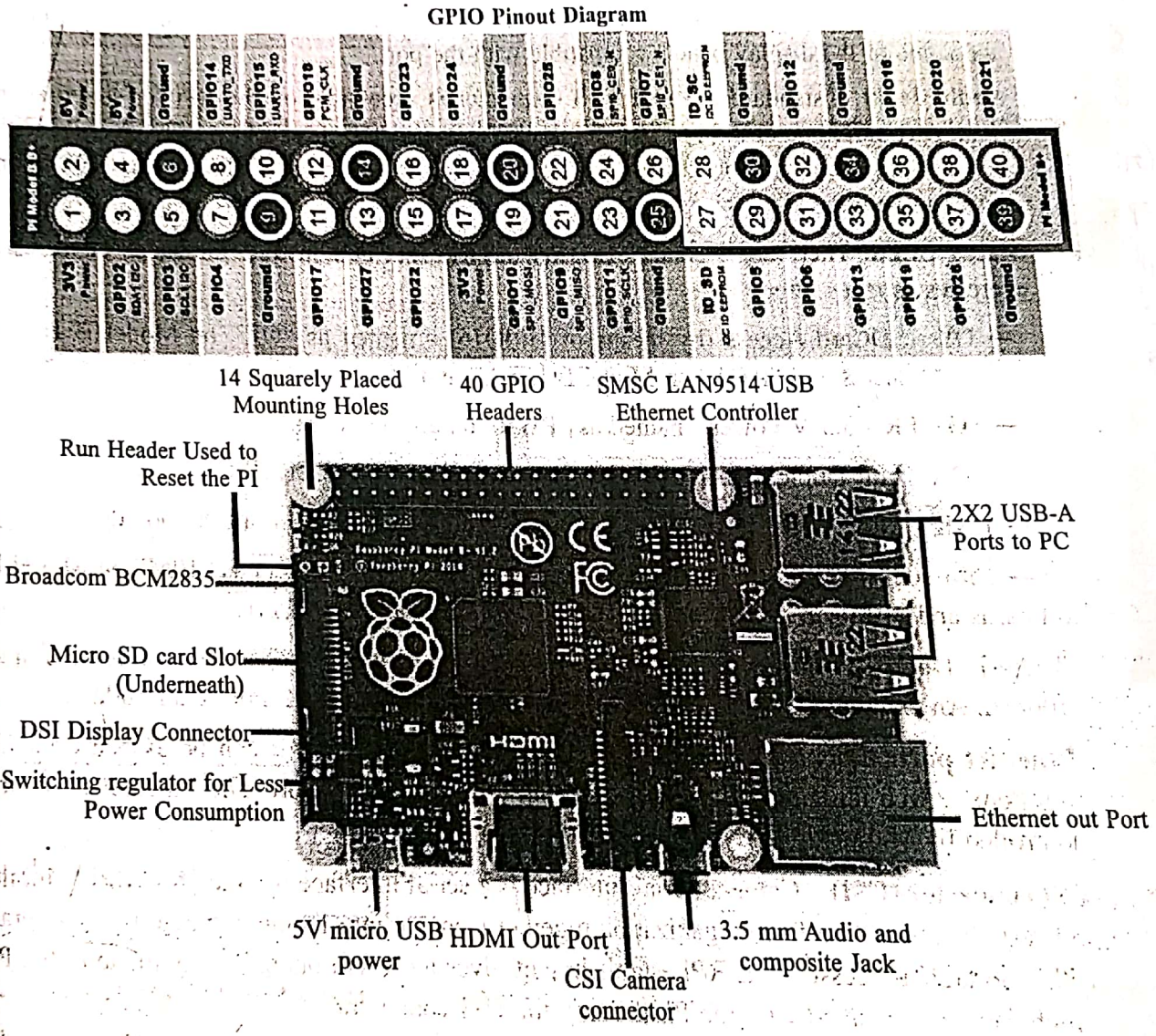


Figure 8-1: Raspberry Pi2 Model B and its GPIO

Processor : The Broadcom BCM2835 SoC used in the first generation Raspberrypi is somewhat equivalent to the chip used in first generation smart phones, which includes a 700 MHz ARM 1176JZF-S processor, Video Core IV graphics processing Unit (GPU) and RAM. This has a level 1 cache of 16KB and a level 2 cache of 128KB.

Power Source : The recommended and easiest way to power the Raspberrypi is Via the Micro USB port on the side of the unit.

SD Card : The Raspberry Pi does not have any locally available storage accessible. The working framework is stacked on a SD card which is embedded on the SD card space on the Raspberry Pi.

GPIO (General Purpose Input Output) : GPIO is a non specific pins on a coordinated circuit to know is an input or output pin which can be controlled by the client at run time. GPIO pins have no exceptional reason characterized, and go unused as a matter of course.

DSI Display X : The Raspberrypi Connector S2 is a display Serial interface (DSI) for connecting a liquid crystal display (LCD) panel using a 15-pin ribbon cable.

Audio Jack : A standard 3.5mm TRS connector is accessible on the RPi for stereo sound yield. Any earphone or 3.5mm sound link can be associated straightforwardly.

Ethernet Port : It is accessible on Model B and B+. It can be associated with a System or web utilizing a standard LAN Link on the Ethernet port.

CSI connector(CSI) : Camera Serial Interface is a Serial interface outlined by MIPI (Mobile Industry Processor Interface) organization together went for interfacing computerized cameras with a portable Processor.

JTAG headers : JTAG is an acronym for 'Join Test Action Group', an association that began back in the mid 1980's to address test point get to issues on PCB with Surface mount gadgets.

Description of System on Chip (SoC)

A System on a chip (SoC) is an integrated Circuit (IC) that co-ordinates all parts of a PC or other electronic framework into a Solitary chip.

It might contained advanced, simple, blended flag, and regularly radio-recurrence works - all on a Solitary chip Substrate. SoCs are exceptionally regular in the portable gadgets advertise in view of their low power utilization. A run of the mill application is in the range of implanted frameworks.

An SoC comprises of:

* A microcontroller, chip or DSP core(s). Some SoCs - called multiprocessor framework on chip (MPSoC) - incorporate more than one processor center.
* Memory pieces including a choice of ROM, RAM, EEPROM and Streak memory.
* Timing Sources including oscillators and Stage bolted circles.
* Simple interfaces including ADCs and DACs.
* Voltage controllers and power administration circuits.

Raspberry Pi interfaces

Raspberry Pi has Serial, SPI and I2C interfaces as shown in the figure of Raspberry Pi Learning board.

* Serial : The Serial interface on Raspberry Pi has receive (rx) and transmit (Tx) pins for Communication With Serial peripherals.
* SPI : Serial Peripheral Interface (SPI) is a Synchronous Serial data Used for communicating with one or more peripheral devices.

* I2C : The I2C interface pins on Raspberry Pi allow you to connect hardware modules. I2C interface allows Synchronous data transfer with just two pins - SDA (data line) and SCL (clock ~~line~~ line).

Raspberry Operating Systems

Various operating systems can be installed on Raspberry through SD cards. Most use a MicroSD slot located on the bottom of the board.

The Raspberrypi primarily uses Raspbian, a Debian-based Linux operating System.

Operating Systems (not Linux based)

- RISC OS Pi
- FreeBSD
- NetBSD
- Plan 9 from Bell Labs and Inferno
- Windows 10 IOT Core - a no cost edition of Windows 10 offered by Microsoft that runs natively on the RaspberryPi 2.

Operating Systems (Linux based)

- Xbian - using Kodi open source digital media center
- openSUSE
- Raspberry Pi Fedora remix
- Pidora, another fedora Remix optimised for Raspberry Pi
- Gentoo Linux
- Diet Pi
- CentOS \ Open Wrt
- Kali Linux
- Ark OS
- Kano OS
- Nard SDK

Media center operating systems

- OSMC
- OpenELEC
- LibreELEC
- Xbian
- Rasplex

Audio operating Systems

- Volumio
- Pimusicbox
- Runeaudio
- moOdeaudio

Recalbox

- Happi Game Center
- Lakka
- ChameleonPi
- Piplay

Operating System Setup On RaspberryPi

Preinstalled NOOBS operating System is already available in many authorized as well as independent Seller, there are many other operating system for RaspberryPi in the market like NOOBS, Raspbian and third party Operating Systems are also available like UBUNTU MATE, OSMC, RISC OS etc. TO setup an operating System we need a SD card with minimum capacity of 8GB.

Formatting SD card

Format the SD card before copying NOOBS onto it. To do this -

- Download SD formatter 4.0 from SD Association Website for either Windows or Mac.

- Follow the instructions to install the Software
- Insert the SD card into the computer or laptops SD card reader and make a note of the drive letter allocated to it.
- In SD formatter, Select the drive letter the SD card is and format it.

OS Installation

Follow the Step to install operating System in SD card

- Go to Raspberry Pi foundation website and click on DOWNLOAD Section.
- Click on NOOBS, then click on "Download zip" button under NOOBS and Select a folder to Save this Zip file.
- Extract all the files from ZIP.
- Once SD card has been formatted, drag all the files in the extracted NOOBS folder and drop them onto the SD card drive.
- The necessary file will then be transferred to the SD card.
- When this process has finished, Safely remove the SD card and insert it into the RaspberryPi.

First Boot

- Plug in the Keyboard, mouse, and monitor cables.
- Now plug the USB cable into the RaspberryPi
- Now Raspberrypi will boot, and a window will appear with a list of different operating System.
- Raspbian will then run through its installation Process.

Programming RaspberryPi With Python

RaspberryPi runs Linux and supports python out of the box. Henceforth you can run any python program that runs on a normal computer. However it is the general purpose input/output capability provided by the GPIO pins on Raspberry Pi that makes it useful device for Internet of things.

Simple python Programs on RaspberryPi:

Program	Code
1. Print hello world	`print ("hello world")`

2. Program to add two numbers	`a = 1.2`
	`b = 5.3`
	`Sum = float(a) + float(b)`
	`Print ("the sum of {0} and {1} is {2}":`
	`format (a,b,sum))`

3. Program to print fibonacci Series	`a,b = 0,1`
	`While b < 200 :`
	`Print(b)`
	`a,b = b, a+b`

4. Program to display calender of given month of the year	`import calender`
	`yy = 2017`
	`mm = 11`
	`print(calender.month(yy.mm))`

5. Program to find the ip address of raspberrypi	`import urllib`
	`import re`
	`point("we will try to open this url, in order to get ip address")`
	`url = http : // checkip . dyndns.org`
	`point(url)`