Internet of Things

RTU EXAM **2021**

Krishna Rani | B.Tech (7th Sem) CSE | Feb 05, 2021
AITS Udaipur (Raj), Syllabus of RTU Kota

Note: Theory – All definition is Written in simple words,
So that you can write in Exams properly
Picture – We have taken all the simple pictures,
So that there is no problem at the time of Exam
Short Note - For Understand, Long Note – for Exam

Content

IOT Hardware and Software

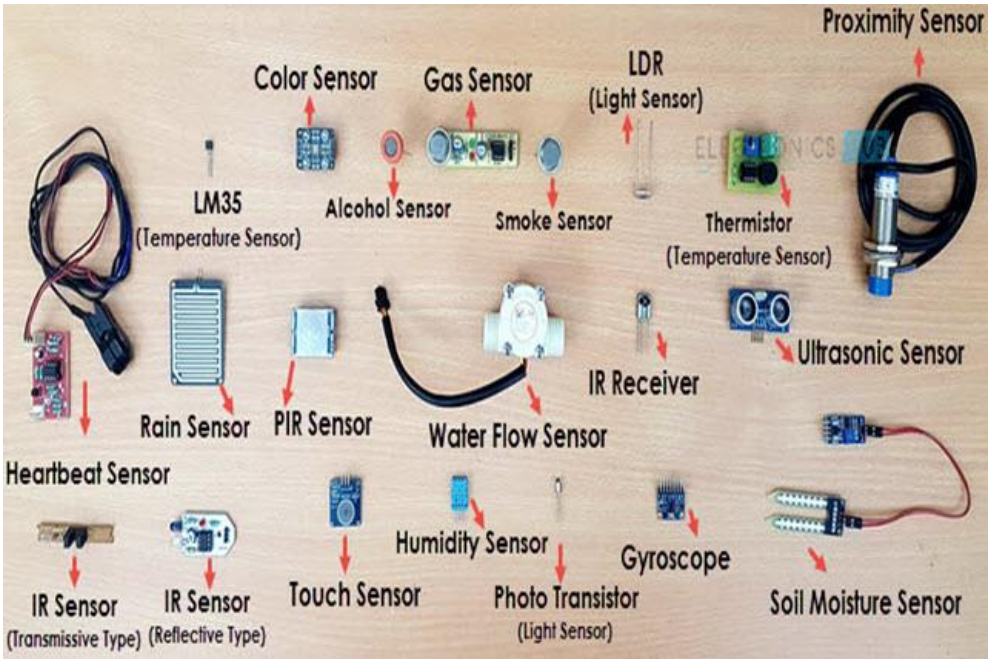
Sensor: - A **sensor** is a device that measures physical input from its environment and converts it into data that can be interpreted (व्याख्या की गई) by either a human or a machine. Most **sensors** are electronic (the data is converted into electronic data).

Key Note: All these sensors are used for measuring one of the physical properties like Temperature, Resistance, Capacitance, Conduction, Heat Transfer etc.

(2nd Definition): a sensor is a device, module, machine, or subsystem whose purpose is to detect events or changes in its environment and send the information to other electronics, frequently a computer processor. A sensor is always used with other electronics.

Type of Sensor :

1. Temperature Sensor.
2. Proximity Sensor.
3. Accelerometer.
4. IR Sensor (Infrared Sensor)
5. Pressure Sensor.
6. Light Sensor.
7. Ultrasonic Sensor.
8. Smoke, Gas and Alcohol Sensor.



This Fig. Used Only For Understanding

Block Diagram of Sensor: Sensors are used to switch currents and voltages. Every sensor has three terminals: Vcc, GND and output.

1. Vcc is used to power up the sensor (पावर देने के लिए)
2. Ground is Used to Negative reference
3. Output Terminal is used to output

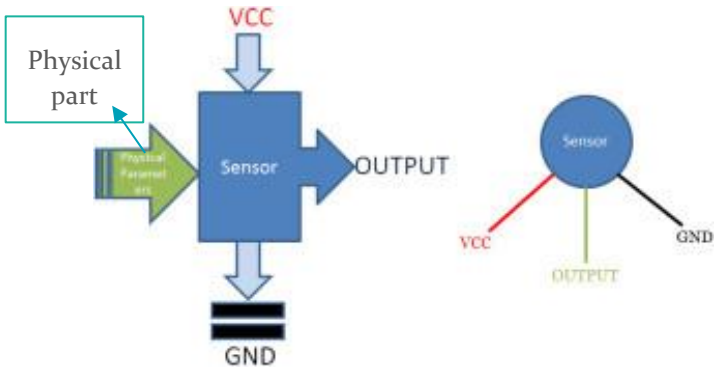


Fig. 1. block diagram

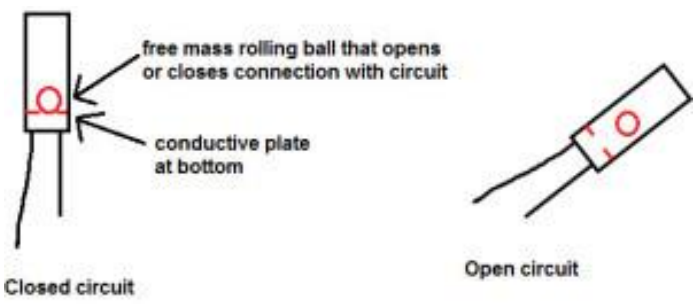


Fig. Sensor working principle

Tilt (झुकाव) Sensor Working Principle: These sensors consist of a rolling ball with a conductive plate beneath them. When the sensor gets power, the rolling ball falls to the bottom of the sensor to form an electrical connection. When the sensor is tilted, the rolling ball doesn't fall to the bottom so that the current cannot flow the two end terminals of the sensor.

इन सेंसरों में एक रोलिंग बॉल होती है, जिसके नीचे एक प्रवाहकीय प्लेट होती है। जब सेंसर को बिजली मिलती है, तो रोलिंग बॉल विद्युत कनेक्शन बनाने के लिए सेंसर के निचले भाग में आती है। जब सेंसर को झुकाया जाता है, तो रोलिंग बॉल नीचे नहीं आती है, ताकि करंट सेंसर के दो अंतिम टर्मिनलों को प्रवाहित न कर सके।

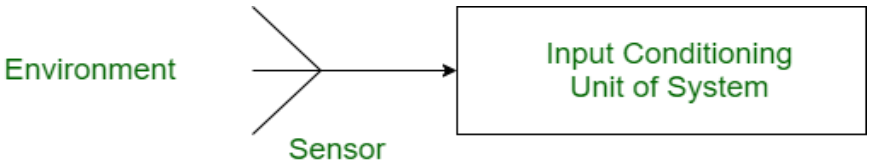
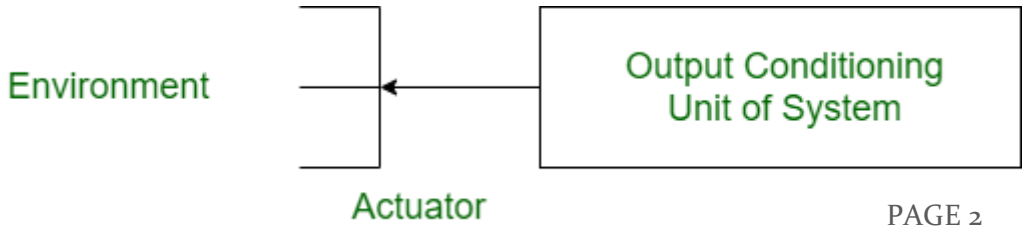


Fig . Sensor

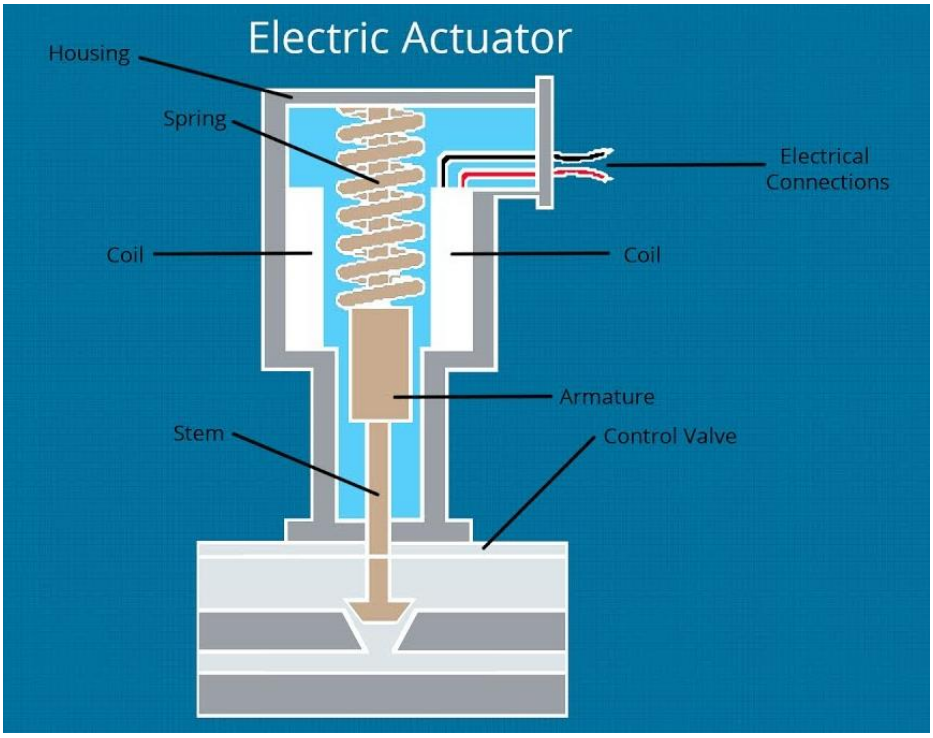
Fig . Actuator



Actuator:

Different Type of Actuator:

1. Linear Actuator (move Straight-line)
   Example. Car Seat
2. Rotary Actuator (move Circular motion)
   Example. Handwheel
3. Electric Actuator (use from electric energy to operate)
   Example. Electric Moter
4. Hydraulic Actuators (use variety liquids source of energy)
   Example. Hydraulic Motor



SENSOR	ACTUATOR
1. It converts physical characteristics into electrical signals.	1. It converts electrical signals into physical characteristics.
2. It takes input from environment.	2. It takes input from output conditioning unit of system.
3. Output is an Electric Signal.	3. Output is a Movement.
4. Sensor generated electrical signals.	4. Actuator generates heat or motion.
5. It is placed at input port of the system.	5. It is placed at output port of the system.
6. used to measure the physical quantity.	6. It is used to measure the continuous and discrete process parameters.
7. Example: Photo-voltaic cell which converts light energy into electrical energy.	7. Example: Stepper motor where electrical energy drives the motor.

Humidity (आर्द्रता, नमी) Sensor: A humidity sensor is an electronic device that measures. humidity in its environment and converts its findings into a corresponding electrical signal Relative **humidity** is calculated by comparing the live **humidity** reading at a given temperature to the maximum amount of **humidity** for air at the same Temperature.

$$AH = m/V$$

Where: AH - Absolute Humidity or RH- Relative Humidity, m - mass, V - Vapour

एक आर्द्रता सेंसर एक इलेक्ट्रॉनिक उपकरण है जो अपने वातावरण में आर्द्रता को मापता है और अपने निष्कर्षों को एक संबंधित विद्युत संकेत में परिवर्तित करता है। सापेक्ष आर्द्रता की गणना एक दिए गए तापमान पर एक ही तापमान पर हवा के लिए आर्द्रता की अधिकतम मात्रा में लाइव आर्द्रता रीडिंग की तुलना करके की जाती है

There are three basic types of humidity sensors:

1. Capacitive (Measure moisture levels using a humidity-dependent)

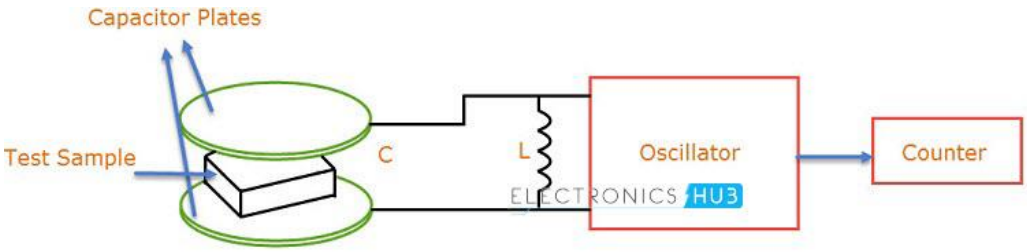
   Example (Used) : industrial and commercial environments

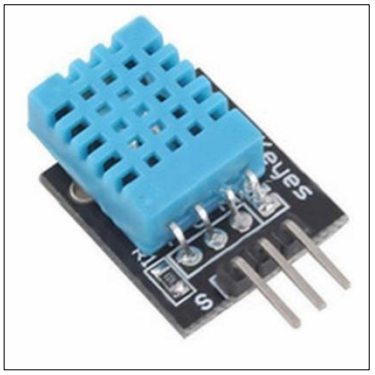
2. Resistive (These sensors can measure the electrical change in devices such as conductive polymers and treated substrates)

   Example (Used) : residential and commercial environments.

3. Thermal (use in environments that have high temperatures. They measure humidity by calibrating the difference between the thermal conductivity of dry air and that of moist air.) वे सूखी हवा की तापीय चालकता और नम हवा के बीच अंतर को मापते हुए आर्द्रता मापते हैं।

fig. Humidity Sensor





Ultrasonic Sensor: Ultrasonic transducers and ultrasonic sensors are devices that generate or sense ultrasound energy.

They can be divided into three broad categories:

1. Transmitters **{Emitter}** (emits the sound using piezoelectric crystals),

2. Receivers **{Detector}** (encounters the sound after it has travelled to and from the target) and 3. Transceivers.

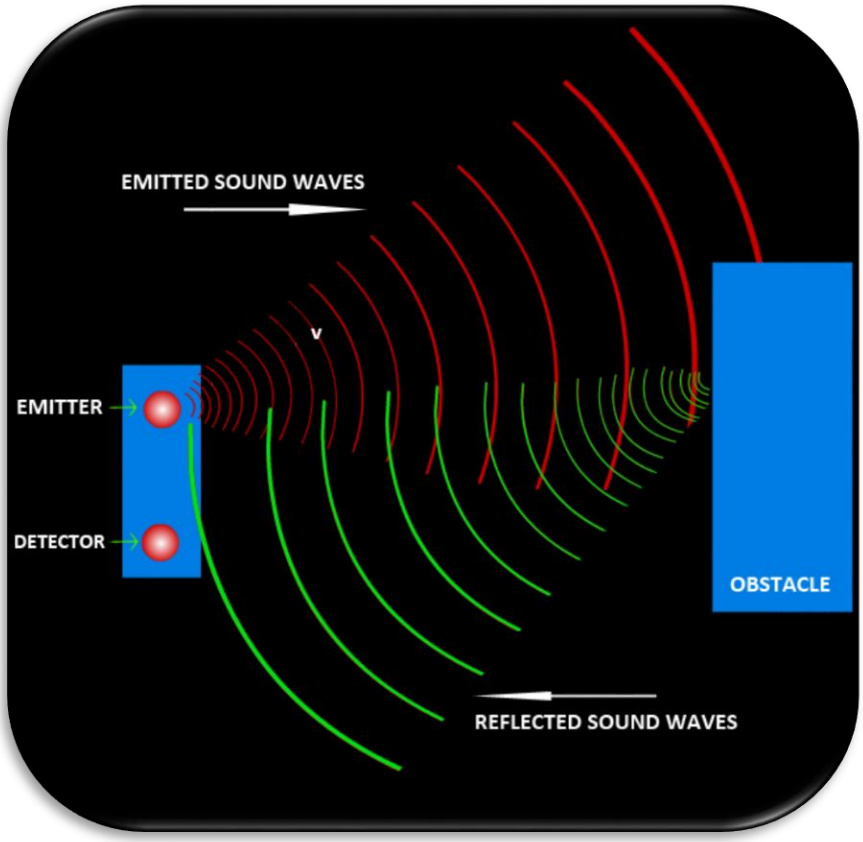
(2nd Definition) : An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear).

Calculate the distance between the sensor and the object: **D = ½ T* C**

(Where D is the Distance, T is the Time, C is the Speed of Sound ~343 meter/second.)

For example, if a scientist set up an ultrasonic sensor aimed at a box and it took 0.025 seconds for the sound to bounce back, the distance between the ultrasonic sensor and the box would be

D = 0.5 x 0.025 x 343 (or about 4.2875 meters) (fig. Working of Ultrasonic Sensor)



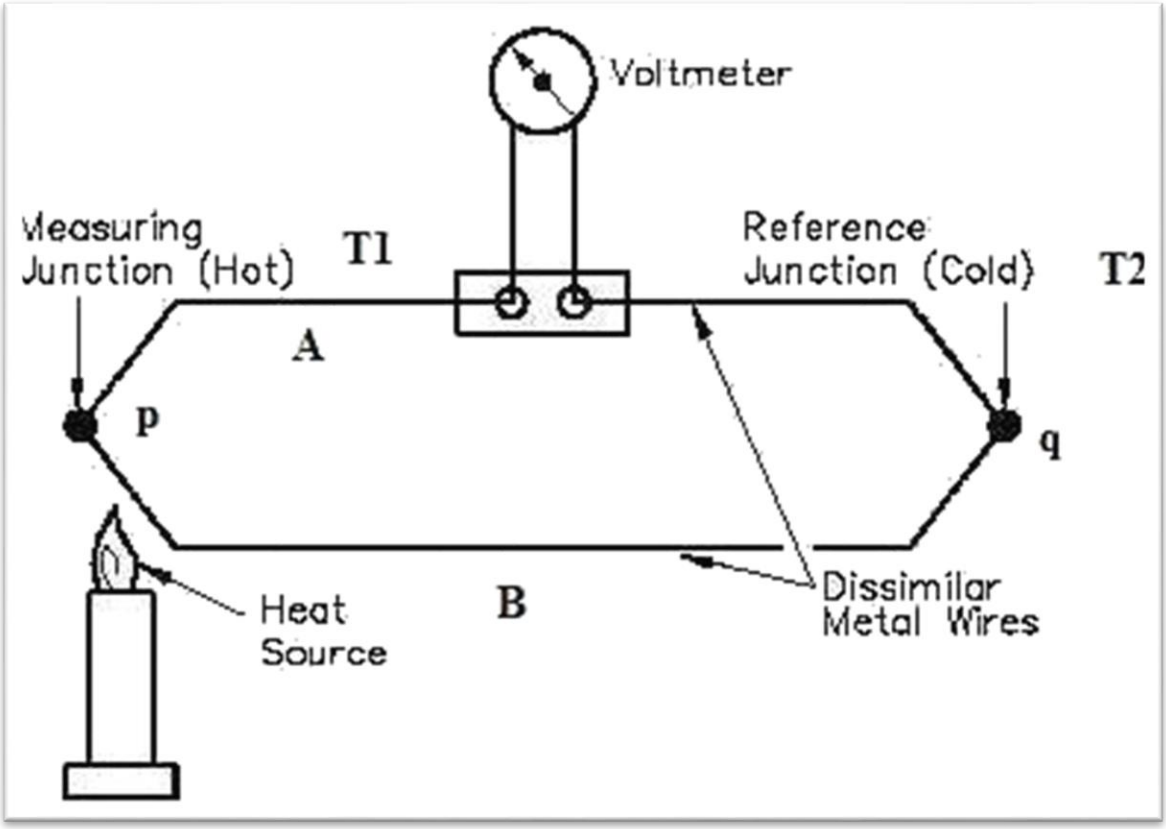
Temperature Sensor: A **temperature sensor** is a device, typically, a thermocouple or resistance **temperature** detector, that provides **Temperature** measurement in a readable form through an electrical signal. A thermometer is the most basic form of a **temperature** meter that is used to measure the degree of hotness and coolness.

(2nd Definition) : A temperature sensor is an electronic device that measures the temperature of its environment and converts the input data into electronic data to record, monitor, or signal temperature changes.

There are many different types of temperature sensors:

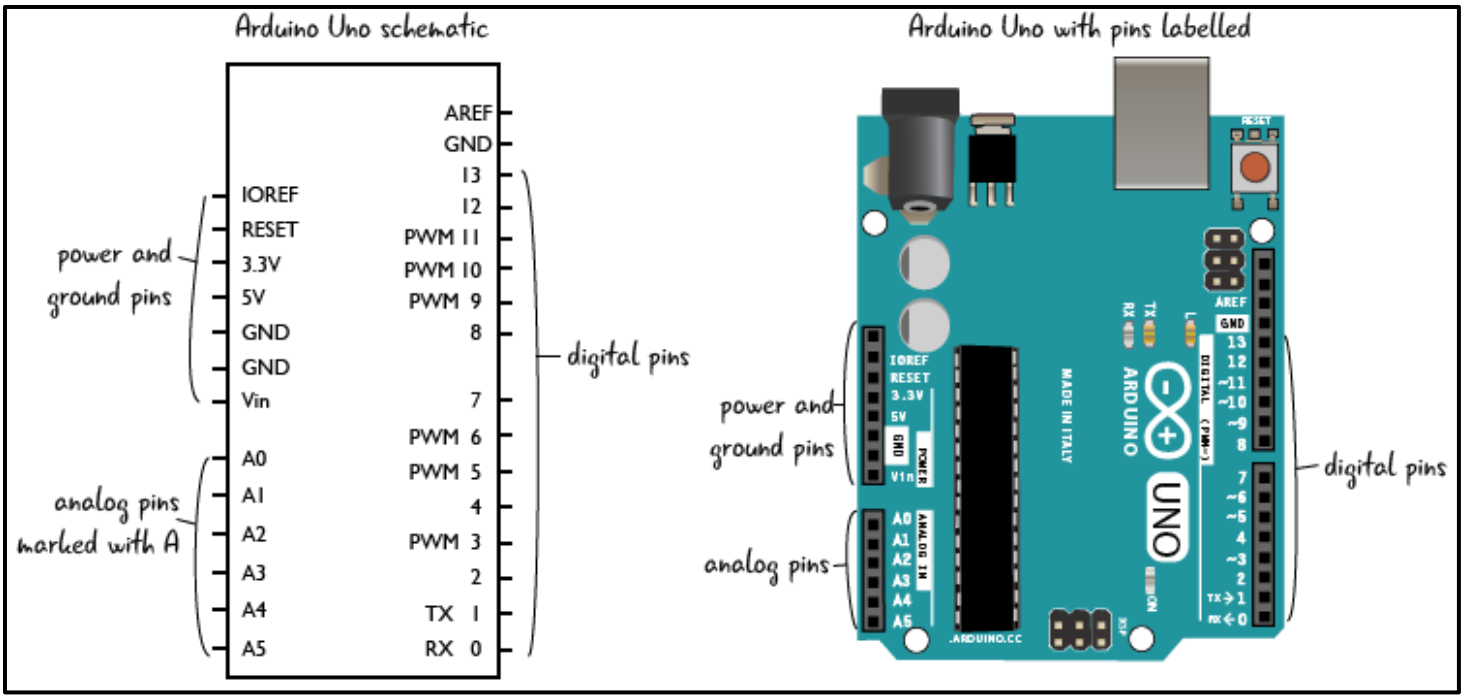
Contact temperature sensors: These types of temperature sensor are required to be in physical contact with the object being sensed and use conduction to monitor changes in temperature. They can be used to detect solids, liquids or gases over a wide range of temperatures.

Non-contact temperature sensors: These types of temperature sensor use convection and radiation to monitor changes in temperature. They can be used to detect liquids and gases,! that emit radiant energy as heat rises and cold settles to the bottom in convection currents ! (गर्मी के रूप में उज्ज्वल ऊर्जा उत्सर्जित करती है और संवहन धाराओं में ठंड नीचे तक बैठती है)



Arduino: Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

(2nd Definition): **Arduino** is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. The boards are equipped with sets of digital and analog input/output

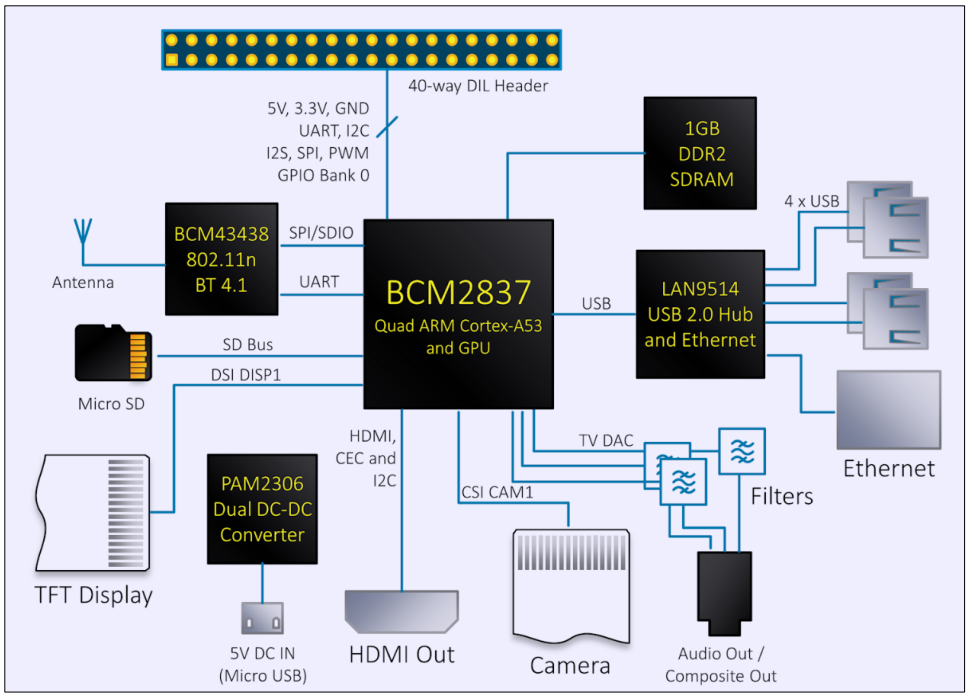


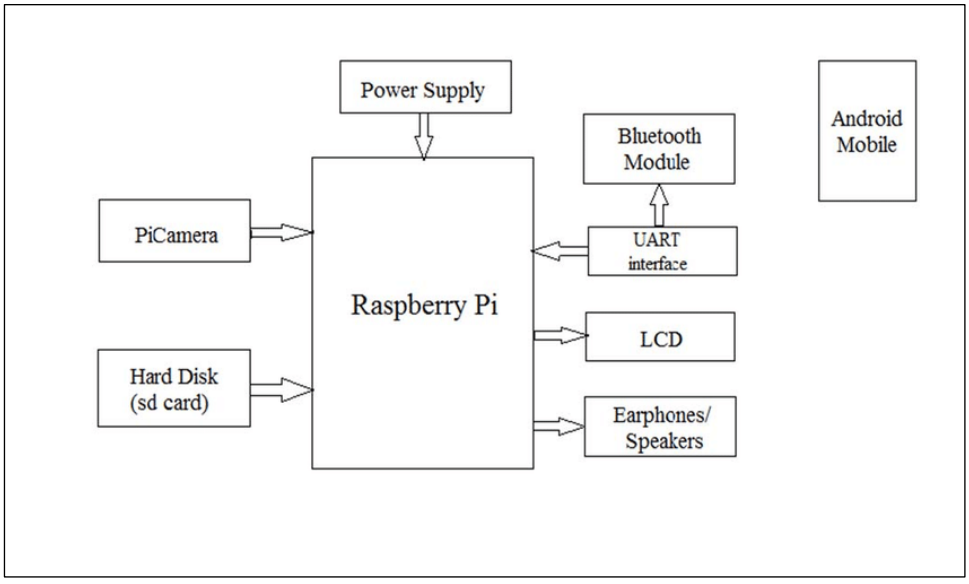
Block diagram of Temperature Sensor with Ardino Circuit and (Example of Led ON/OFF)

```
# define LED_PIN 13                    // Pin number attached to LED.

void setup() {
    pinMode(LED_PIN, OUTPUT);          // Configure pin 13 to be a digital
output.
}

void loop() {
    digitalWrite(LED_PIN, HIGH);       // Turn on the LED.
    delay(1000);                       // Wait 1 second (1000 milliseconds).
    digitalWrite(LED_PIN, LOW);        // Turn off the LED.
    delay(1000);                       // Wait 1 second.
}
```

Raspberry Pi: Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation in association with Broadcom. Early on, the Raspberry Pi project leaned towards the promotion of teaching basic computer science in schools and in developing countries.

(2nd Definition): 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.





Date introduced: 29 February 2012

Release date: 24 February 2012; 8 years ago (Original); 24 June 2019; 19 months ago (Current)

Power: 5 V; 3 A (for full power delivery to USB devices)

Storage: MicroSDHC slot, USB Mass Storage device for booting

Introductory price: : US$4 (Pi Pico); US$35 (Pi 4 2 GiB); US$55 (Pi 4 4 GiB); US$75 (Pi 4 8 GiB);

Note: 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 ...

LiteOS: Huawei LiteOS is a lightweight real-time operating system from Huawei Technologies Co., Ltd. Based on Linux, it is an open source operating system licensed under a BSD 3-clause license for IoT smart terminals. Microcontrollers of different architectures such as ARM, x86, and RISC-V are supported by the project.

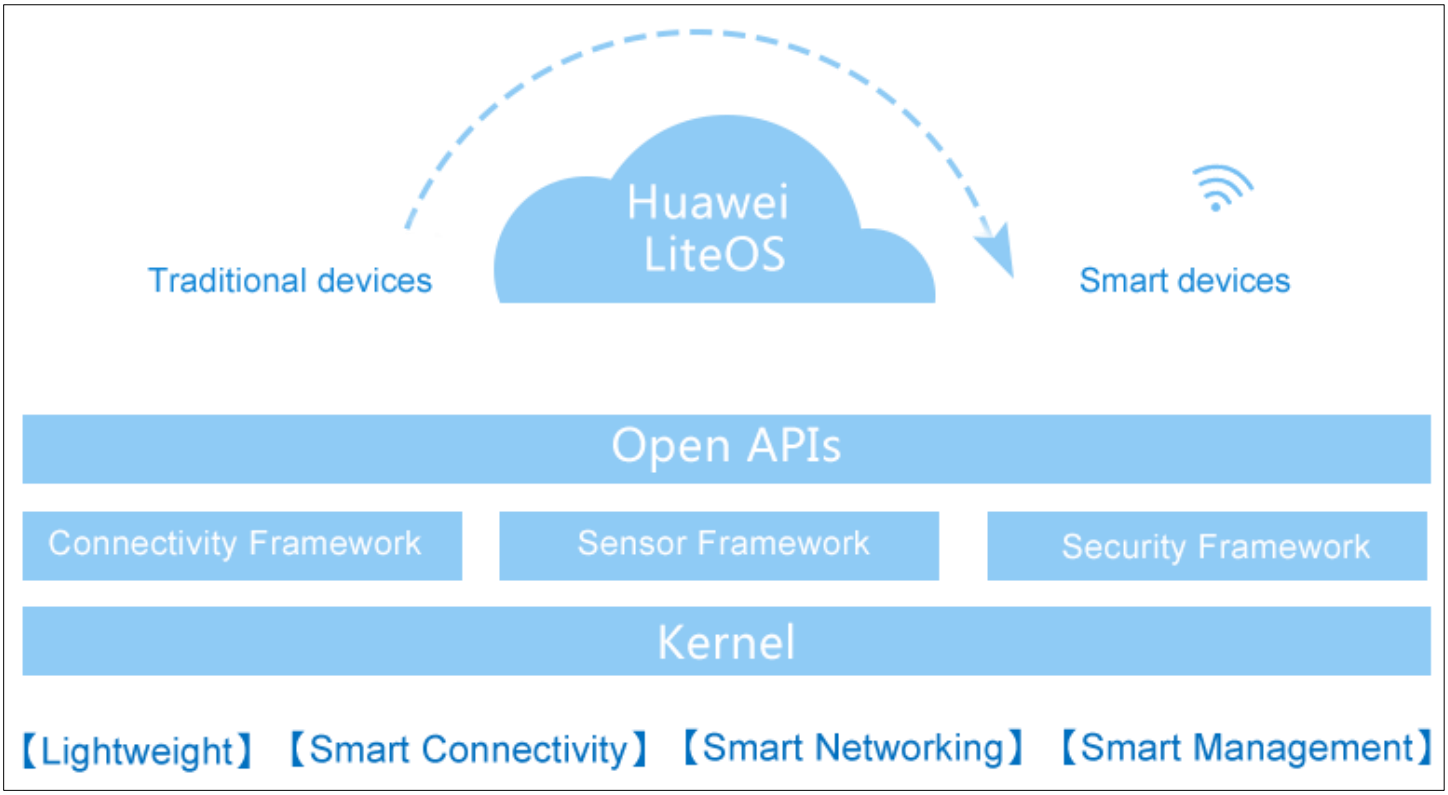
Note: ARM - Previously an acronym for **Advanced RISC Machines** and originally **Acorn RISC Machine**) is a family of reduced instruction set computing (RISC) architectures for computer processors, configured for various environments.

Source model: Open source

Developer: Huawei

Latest release: V2.1 / May 2018; 2 years ago

License: BSD 3-clause



Note: you use blank space for any missing Content

RIoTOS: RIOT is a small operating system for networked, memory-constrained systems with a focus on low-power wireless Internet of Things devices. It is open-source software, released under the GNU Lesser General Public License.

OS family: Embedded operating systems **Source model**: Open source **License**: LGPLv2

Platforms: TI MSP430, ARM7, ARM Cortex-M0-M0+-M3-M4, Atmel AVR, MIPS32r2, RISC-V

Latest release: 2018.04 / 11 May 2018; 2 years ago **Programming languages**: C, C++

(2nd Definition): **RIOT OS**, which is free and open source, is specially designed to meet the particular needs of the IoT, with features like a low memory footprint, high energy efficiency, real-time capabilities, a modular and configurable communication stack, and support for a wide range of low-power devices.

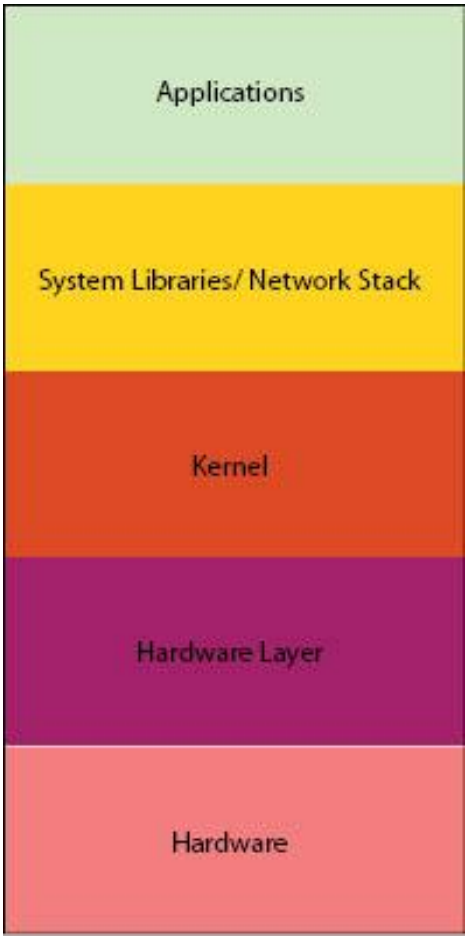


Figure: Components of the IOT

Features of RIOT

The OS is actively developed and maintained.

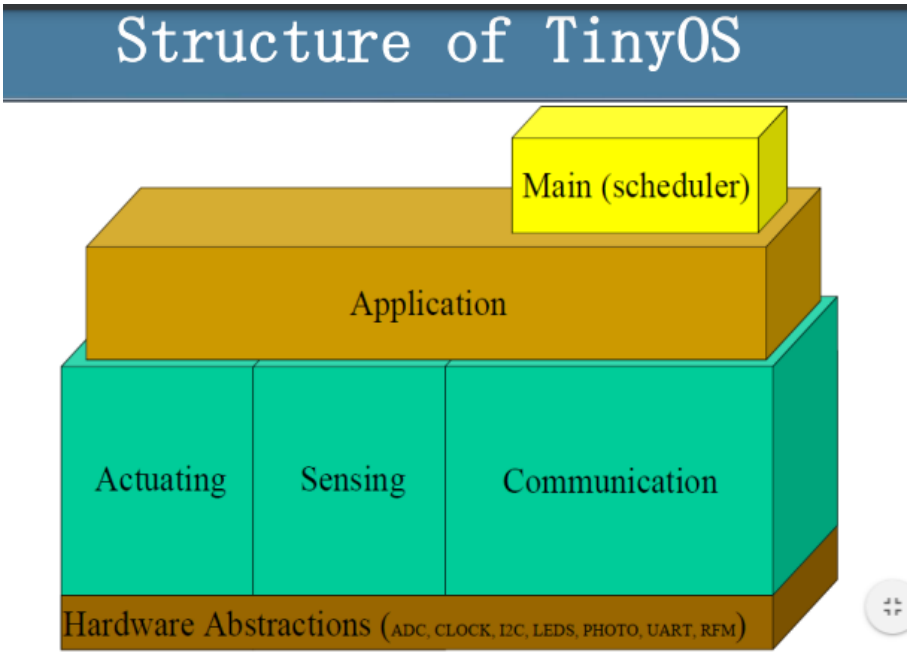
- There are no new programming environments. C or C++ can be used directly with existing tools like gcc, gdb, etc
- Less hardware dependent code
- Supports 8-,16- and 32-bit microcontroller platforms
- Energy efficieny is maintained
- Less interrupt latency, so real-time capability is ensured
- Multi-threading is enabled
- Supports the entire network stack of IoT (802.15.4 Zigbee, 6LoWPAN, ICMP6, Ipv6, RPL, CoAP, etc)
- Both static and dynamic memory allocation
- POSIX compliant (partial)

Contiki (Same as TinyOS): Contiki is an operating system for IoT that specifically targets small IoT devices with limited memory, power, bandwidth, and processing power. It uses a minimalist design while still packing the common tools of modern operating systems. It provides functionality for management of programs, processes, resources, memory, and communication.

TinyOS: TinyOS is an embedded, component-based operating system and platform for low-power wireless devices, such as those used in wireless sensor networks, smartdust, ubiquitous computing, personal area networks, building automation, and smart meters.

What is TinyOs?

1. TinyOS is a free open source operating system.
2. Designed for wireless sensor networks.
3. TinyOS began as a collaboration between University of California, Berkeley and Intel Research.
4. An embedded operating system written in nesC language.
5. It features a component based architecture.



TinyOs Models

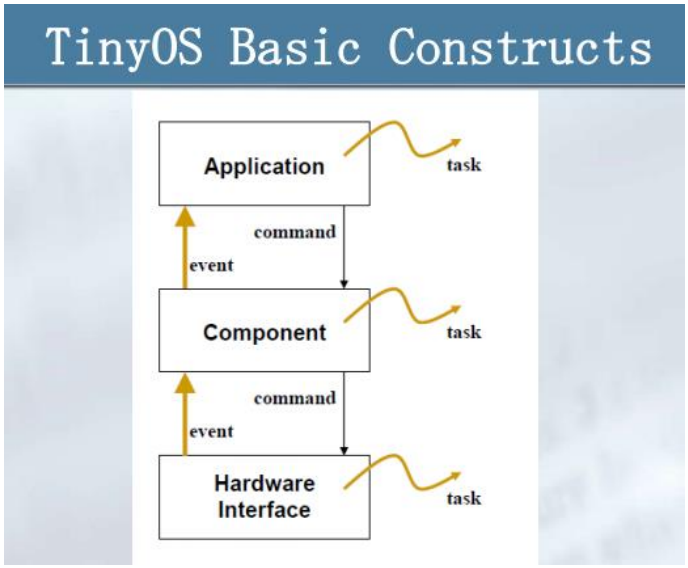
- Data Model:

1. Statics MemoryAllocation
2. Gllobal Variables
3. Local Variable

- Thread Model:

a. Power-Aware two-levels schedulling
2. Tasks
3. Events

- Programming Model:
- Component Model:
- Network Model



Architecture and Reference Model

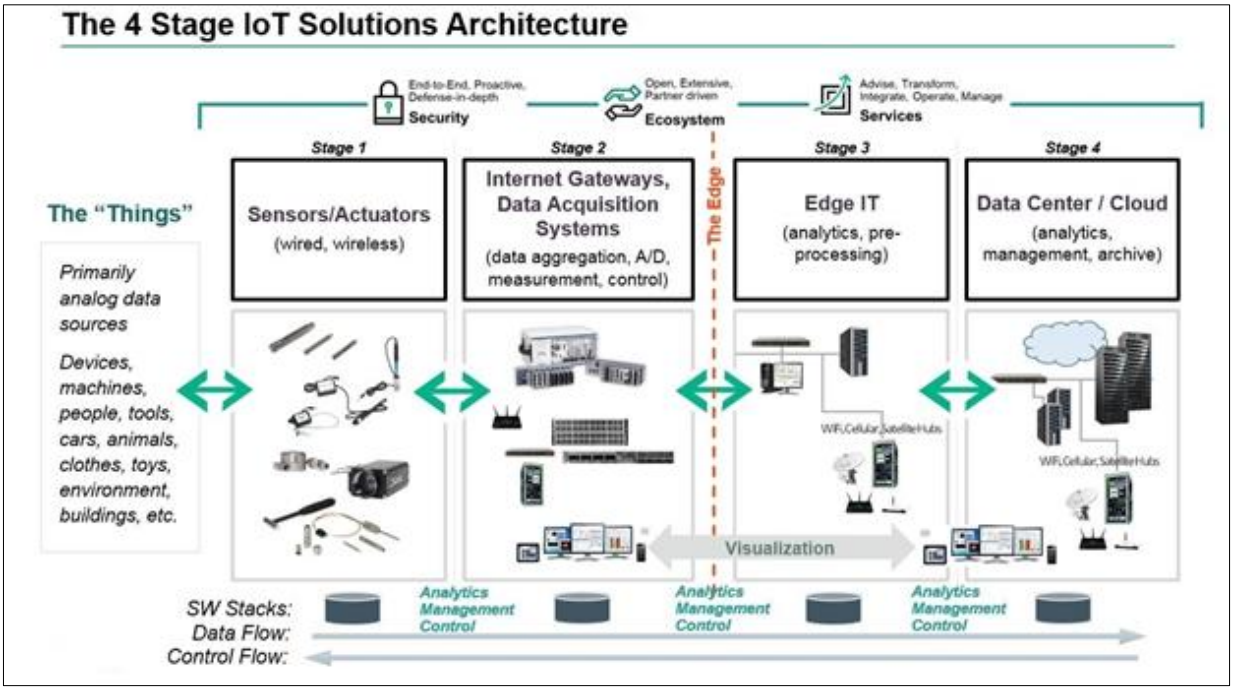
Introduction: IoT architecture is the system of numerous elements: sensors, protocols, actuators, cloud services, and layers. Given its complexity, there exist 4 stages of IoT architecture. Such a number is chosen to steadily include these various types of components into a sophisticated and unified network.

Basically, there are three IoT architecture layers: (2nd name of this option)

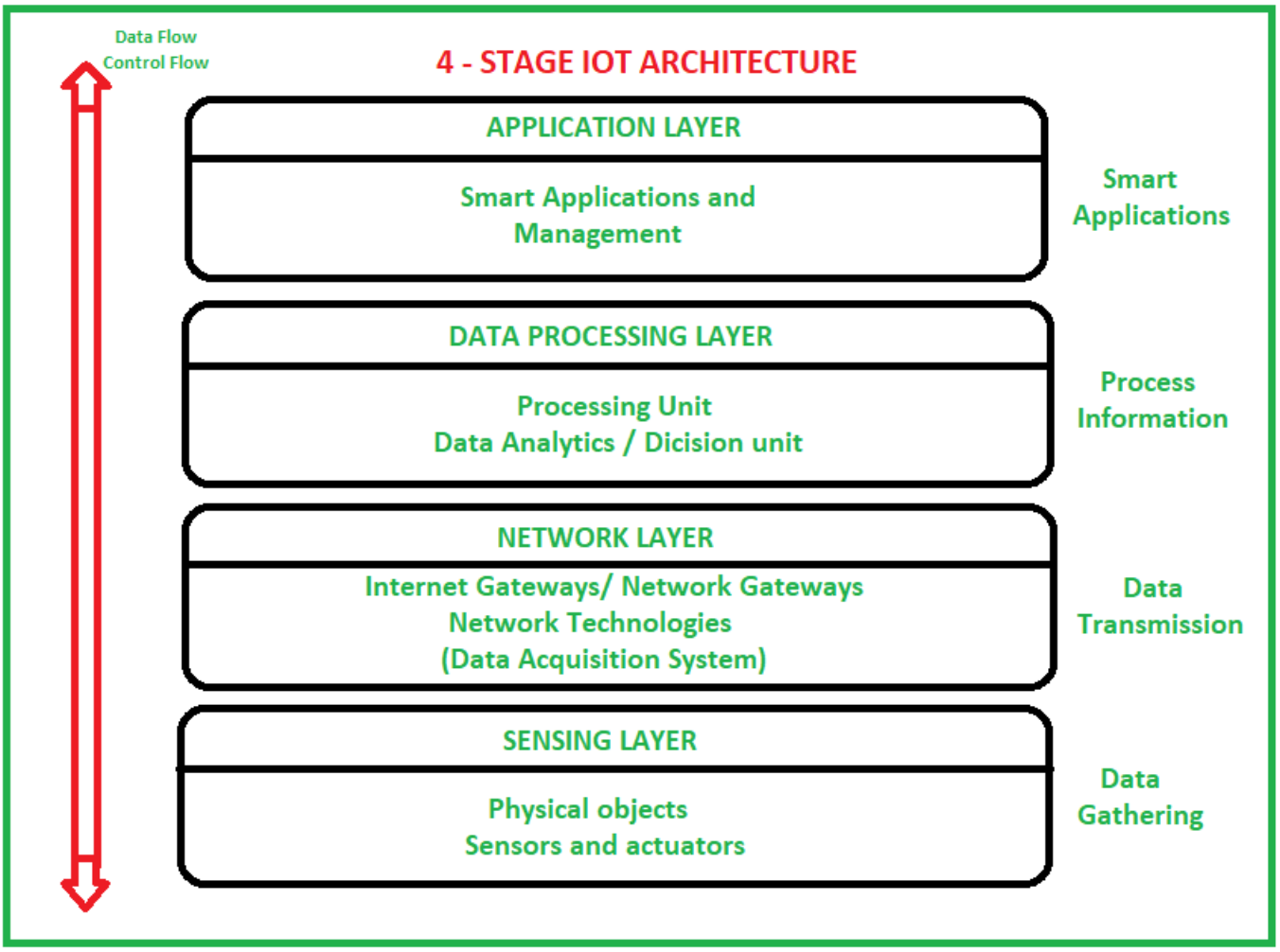
1. The client side (IoT Device Layer)
2. Operators on the server side (IoT Getaway Layer)
3. A pathway for connecting clients and operators (IoT Platform Layer)

Simple terms, the 4 Stage IoT architecture consists of (2nd name of this option)

1. Sensors and actuators (Sensing Layer)
2. Internet getaways and Data Acquisition Systems (Network Layer)
3. Edge IT (Application Layer)
4. Data center and cloud. (Data processsing)



This fig. for Understand and Exam View



(2nd figure of 4 stage IOT Architecture)

1. Sensing Layer: These Sensors or Actuators accepts data (physical, environmental parameters), processes data and emits data over network.

2. Network Layer: Internet/Network gateways, Data Acquisition System (DAS) are present in this layer. DAS performs data aggregation and conversion function (Collecting data and aggregating data then converting analog data of sensors to digital data etc).

3. Data processing Layer: This is processing unit of IoT ecosystem. Here data is analyzed and pre-processed before sending it to data center from where data is accessed by software applications

4. Application Layer: 4 stages of IoT architecture. Data centers or cloud is management stage of data where data is managed and is used by end-user applications like agriculture, health care, aerospace, farming, defense, etc

Representational State Transfer (REST) architectural style

REST: REST stands for REpresentational State Transfer and API stands for Application Program Interface. REST is a software architectural style that defines the set of rules to be used for creating web services. Web services which follow the REST architectural style are known as RESTful web services. It allows requesting systems to access and manipulate web resources by using a uniform and predefined set of rules. Interaction in REST based systems happen through Internet's Hypertext Transfer Protocol (HTTP)

REST का अर्थ है रिप्रेजेंटेटिव स्टेट ट्रांसफर और API का अर्थ एप्लीकेशन प्रोग्राम इंटरफेस है। REST एक सॉफ्टवेयर आर्किटेक्चरल शैली है, जो वेब सेवाओं को बनाने के लिए उपयोग किए जाने वाले नियमों के सेट को परिभाषित करती है। वेब सेवाएँ जो REST वास्तु शैली का अनुसरण करती हैं, उन्हें RESTful वेब सेवाओं के रूप में जाना जाता है। यह सिस्टम को नियमों के एक समान और पूर्वनिर्धारित सेट का उपयोग करके वेब संसाधनों तक पहुंचने और हेरफेर करने की अनुमति देता है। REST आधारित प्रणालियों में सहभागिता इंटरनेट के हाइपरटेक्स्ट ट्रांसफर प्रोटोकॉल (HTTP) के माध्यम से होती है।

A Restful system consists of a:

- client who requests for the resources.
- server who has the resources.

Architectural Constraints of RESTful API: There are six architectural constraints which makes any web service are listed below:

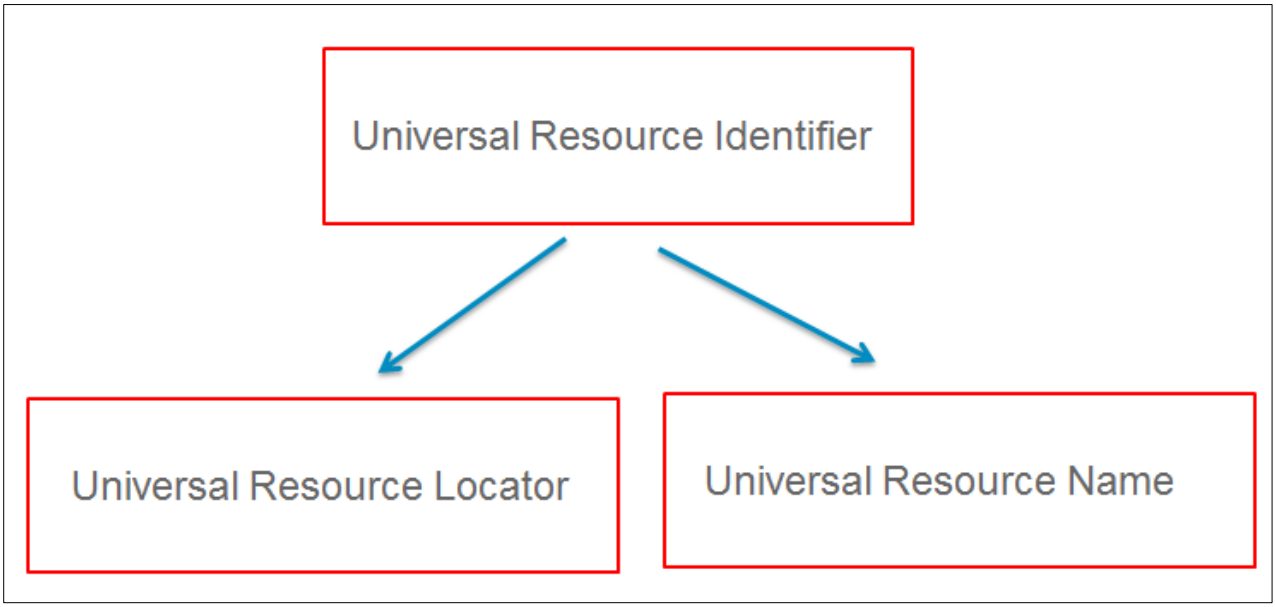
- Uniform Interface
- Stateless
- Cacheable
- Client-Server
- Layered System
- Code on Demand

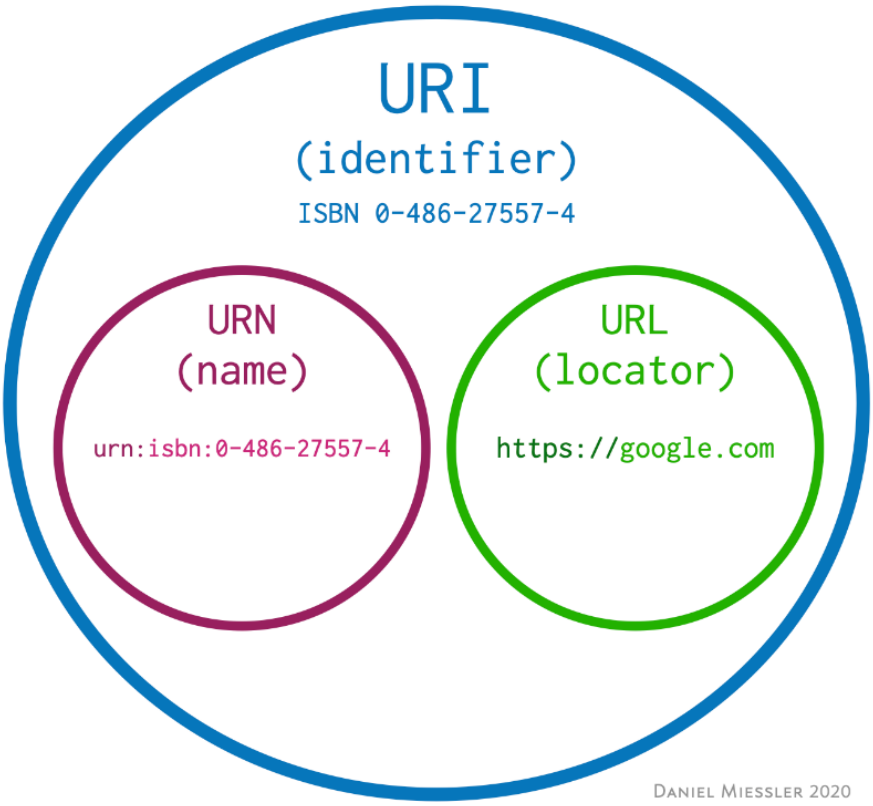
Communication between Client and Server

- Making request
- HTTP Verbs
- Header And Accept Parameters
- paths

Uniform Resource Identifiers (URIs): A Uniform Resource Identifier is a unique sequence of characters that identifies a logical or physical resource used by web technologies. URIs may be used to identify anything, including real-world objects, such as people and places, concepts, or information resources such as web pages and books.

URI Syntax: The generic form of any URI is **scheme:[//[user:password@]host[:port]][/]path[?query][#fragment]**





Scheme:

1. Authority Component: Authentication section, Registered name, Ip Address, Username, Password, symbol(@), Color, brackets{}[]

2. Fragment (HTML): Section, Article, Id

3. Query (String Attribute): Attribute value, Semicolon, question Marks

Challenges in IoT- Design challenges: Many IoT Systems are poorly designed and implemented, using diverse protocols and technologies that create complex configurations. Lack of mature IoT technologies and business processes. Limited guidance for life cycle maintenance and management of IoT devices. Limited best practices available for IoT developers

Four Challenges to be Considered When Developing IoT Devices

1. Connectivity: first concerning issue, i.e., how to connect devices to the Internet and the cloud-computing platform

For example, if you need to develop a smart home device, such as an online toaster, you may access a Wi-Fi home router or a ZigBee/Z-Wave IoT router.

2. Security & Privacy:

1. Physical Security
2. Security of Data Exchange
3. Security of Cloud Storage
4. Update

3. Flexibility & Compatibility: IoT pattern is continuously changing, you must ensure that your product can support future technologies. However, it requires you to balance between software and hardware when designing your product.

4. Data Collection & Processing

1. Supplement Existing Data
2. Analyze and Further Divide Users
3. Find Opportunities to Improve Product