UNIT IV Developing Internet of Things

4.1 IoT Platforms Design Methodology

• Purpose & Requirements Specification

• Process Specification

• Domain Model Specification

• Information Model Specification

• Service Specifications

• IoT Level Specification

• Functional View Specification

• Operational View Specification

• Device & Component Integration

• Application Development

4.1.1 PURPOSE &REQUIREMENTS SPECIFICATION

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

• Applying this to our example of a smart home automation system, the purpose and requirements for the system may be described as follows:

• Purpose : A home automation system that allows controlling of the lights in a home remotely using a web application.

• Behavior : The home automation system should have auto and manual modes. In auto mode, the system measures the light level in the room and switches on the light when it gets dark. In manual mode, the system provides the option of manually and remotely switching on/off the light.

• System Management Requirement : The system should provide remote monitoring and control functions.

• Data Analysis Requirement : The system should perform local analysis of the data.

• Application Deployment Requirement : The application should be deployed locally on the device, but should be accessible remotely.

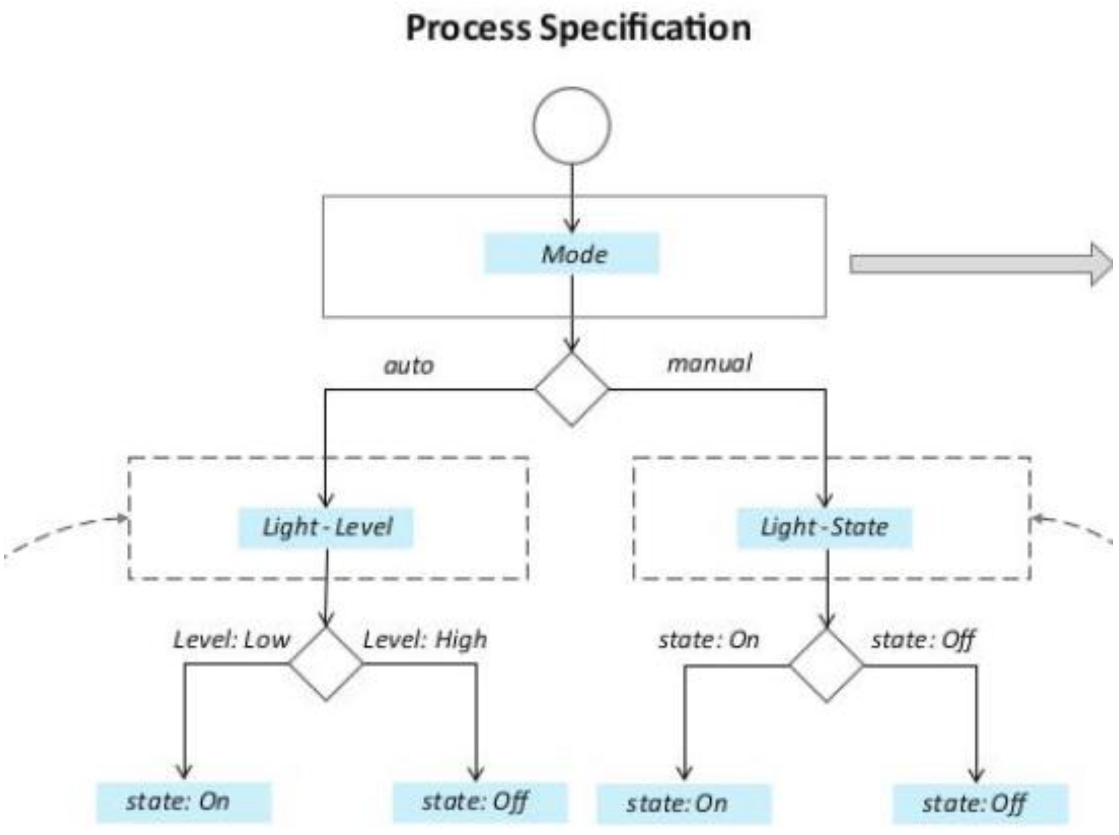
• Security Requirement : The system should have basic user authentication capability.

4.1.2 PROCESS SPECIFICATION

The second step in the IoT design methodology is to define the process specification. In this step, the usecases of the IoT system are formally described based on and derived from the purpose and requirement specifications. Define the process with the help of use cases

• The use cases are formally described based on Purpose & requirement specification In this

From the process specification and information model we identify the states and attributes. For each state and attributes we define a service. These services either change a state or attribute values and are retrieve the current values. For example, the mode service sets mode to auto or manual or retrieves the current mode. The state services sets the light appliances state to ON or OFF. In the auto mode, the controller service monitors the light level in auto mode and switches the light ON or OFF and updates the status in the status database. In manual mode the controller service retrieves the current state from the database and switches the light ON or OFF. The figure deriving services from process specification and information model for home automation IoT system.



4.1.3 DOMAIN MODEL SPECIFICATION

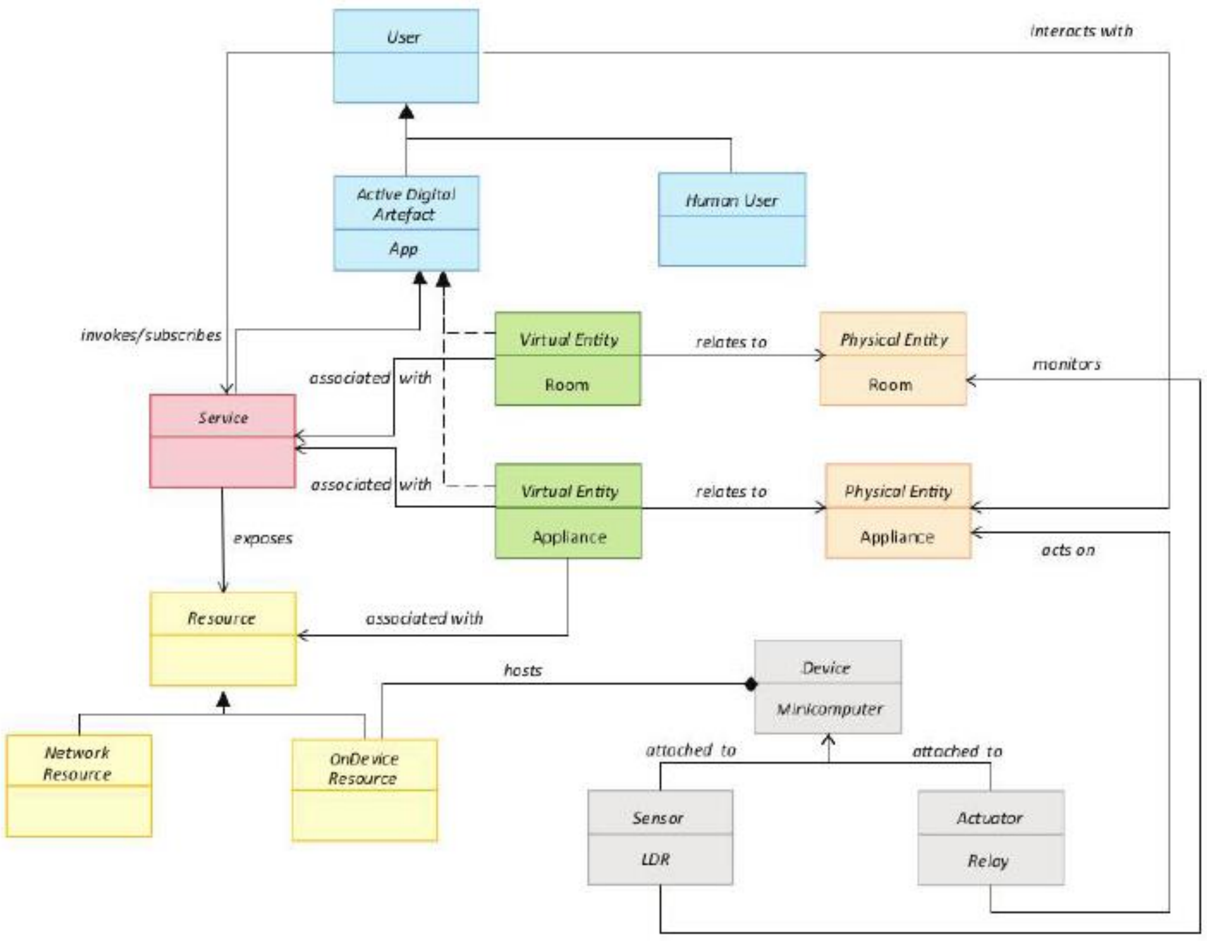
Describes the main concepts, entities and objects in the • The third step in the IoT design methodology is to define the Domain Model. The domain model describes the main concepts, entities and objects in the domain of IoT system to be designed. Domain model defines the attributes of the objects and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

• It defines the attributes of the objects and relationships between them

• Entities , Objects and Concepts include the following : Physical entity, Virtual entity , Device, Resource, Service

• Physical Entity:

– Discreet identifiable entity in physical environment



– For eg. Pump, motor, LCD

– The IoT System provides the information about the physical entity (using sensors) or performs actuation upon the Physical entity(like switching a motor on etc.)

– In smart irrigation example, there are three Physical entities involved :

• Soil (whose moisture content is to be monitored)

• Motor (to be controlled)

• Pump (To be controlled)

• Virtual Entity:

– Representation of physical entity in digital world

– For each physical entity there is a virtual entity

• Device:

– Medium for interactions between Physical and Virtual Entities.

– Devices (Sensors) are used to gather information from the physical entities

– Devices are used to identify Physical entities (Using Tags)

– In Smart Irrigation System, device is soil moisture sensor and buzzer as well as the actuator (relay switch) attached to it.

In smart irrigation system there are three services :

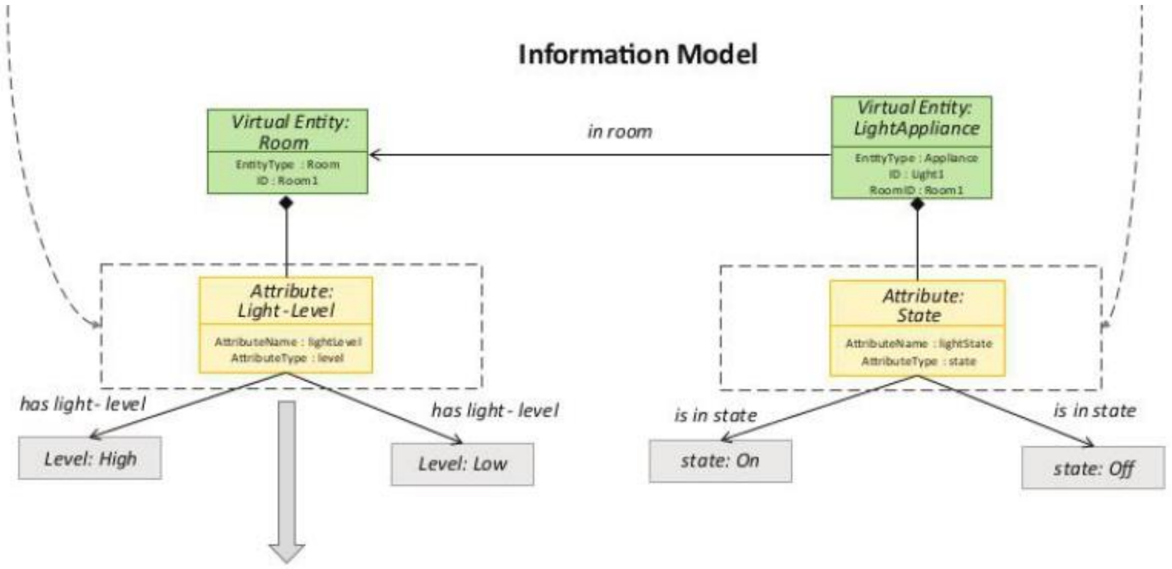
– A service that sets the signal to low/ high depending upon the threshold value

– A service that sets the motor state on/off

– A controller service that runs and monitors the threshold value of the moisture and switches the state of motor on/off depending upon it. When threshold value is not crossed the controller retrieves the motor status from database and switches the motor on/off.

4.1.4 INFORMATION MODEL SPECIFICATION

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



Defines the structure of all the information in the IoT system (such as attributes, relations etc.)

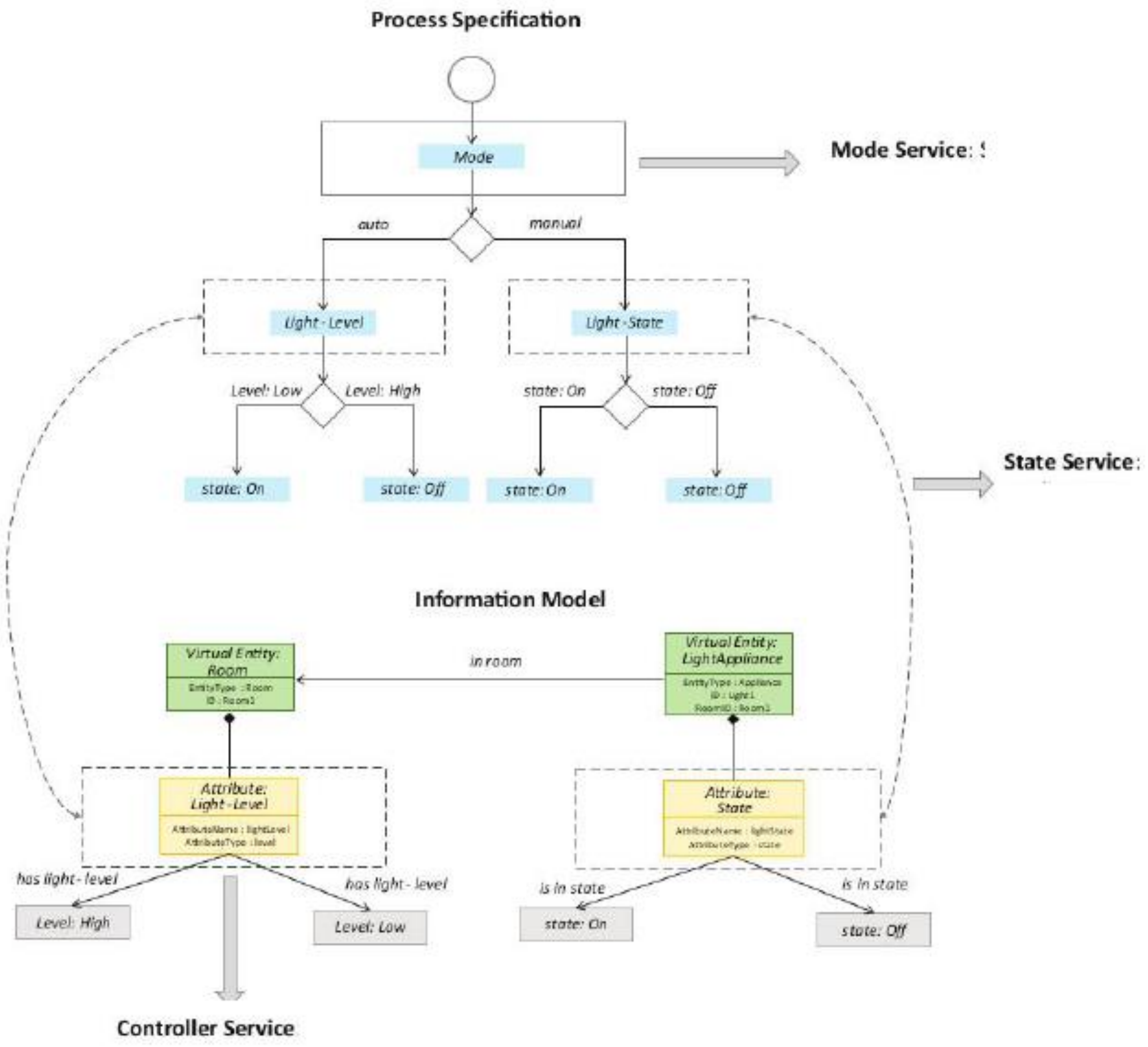
• It does not describe the specifics of how the information is represented or stored.

• This adds more information to the Virtual entities by defining their attributes and relations

• I: e, Draw Class diagram

4.1.5 SERVICE SPECIFICATIONS

The fifth step in the IoT design methodology is to define the service specifications. Service specifications define the services in the IoT system, service types, service inputs/output, service endpoints, service schedules, service preconditions and service effects. The following figures shows deriving the services from the process specification and information model for the home automation IoT system.



Define the services in IoT System, service types, service inputs/outputs, service endpoints, service schedules, service preconditions and service effects

• Services can be controller service, Threshold service, state service for smart irrigation system

• These services either change the state/attribute values or retrieve the current vlues.

• For eg.

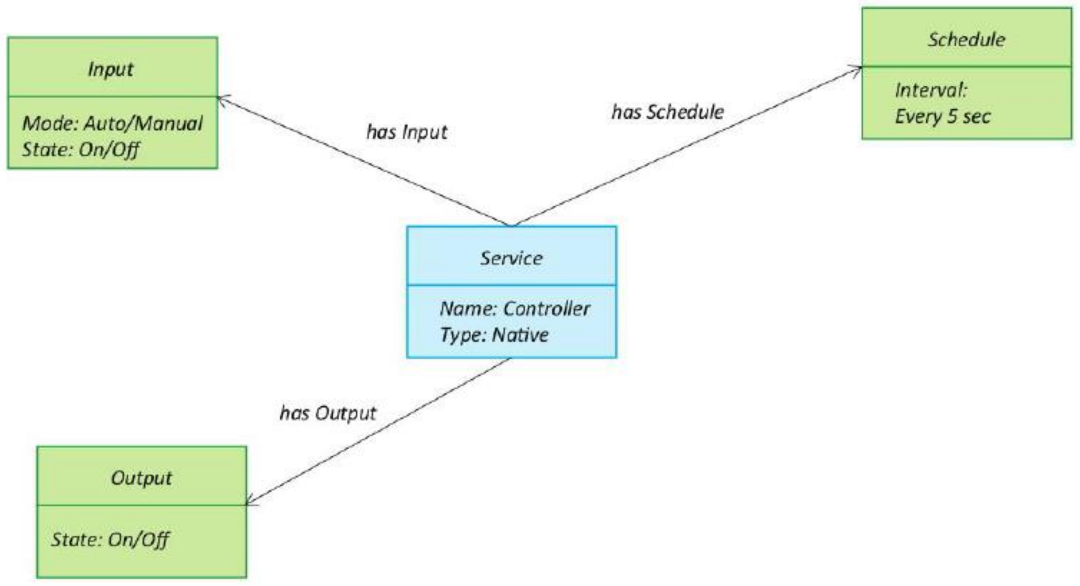
– Threshold service sets signal to high or low depending upon the soil moisture value.

– State service sets the motor state : on or off

– Controller service monitors the threshold value as well as the motor state and switches the motor on/off and updates the status in the database

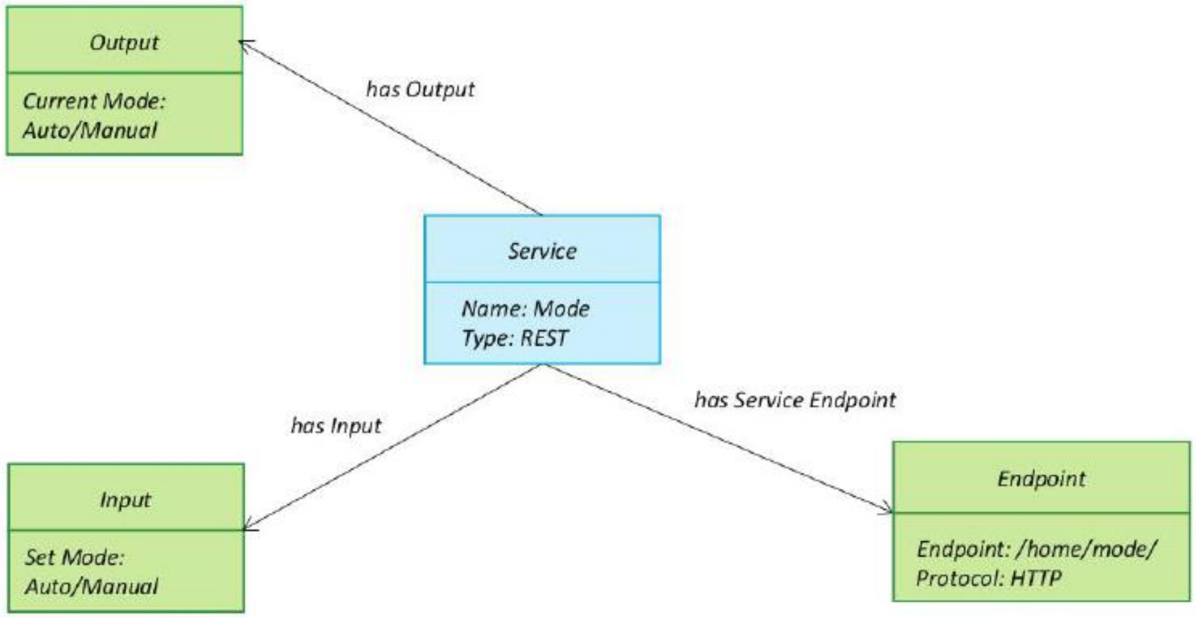
Controller Service:

In the auto mode, the controller service monitors the light level in auto mode and switches the light ON or OFF and updates the status in the status database. In manual mode the controller service retrieves the current state from the database and switches the light ON or OFF

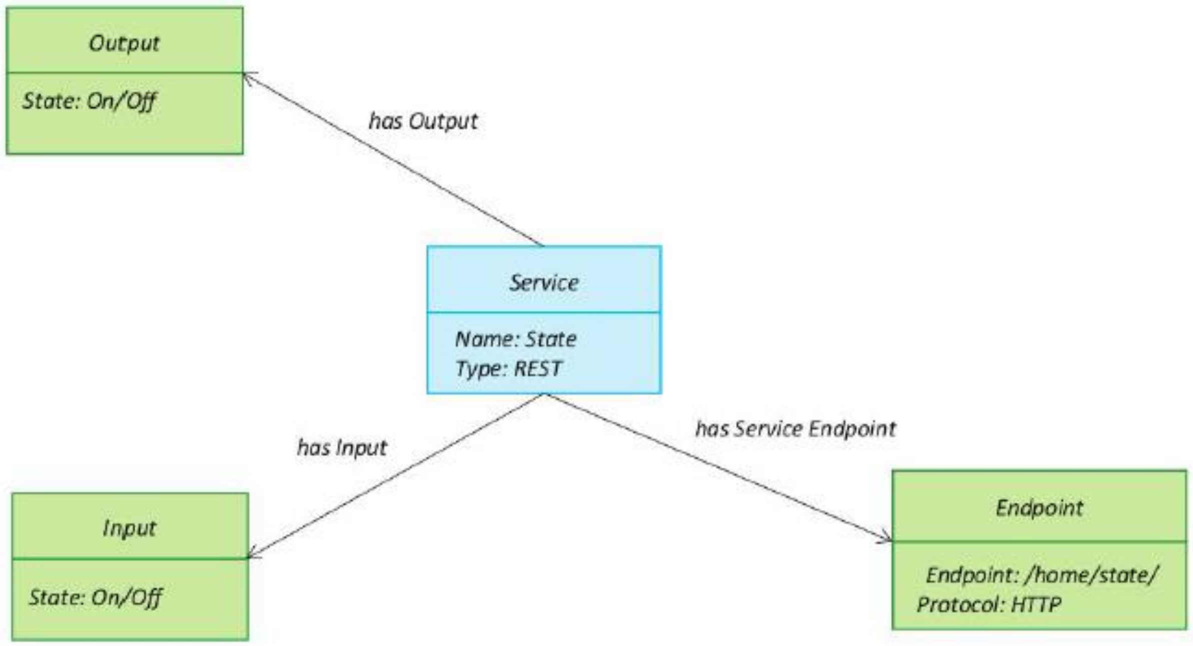


Mode Service: Sets mode to auto or manual or retrieves the current mode

State Service : Sets the light appliances state to ON or OFF.

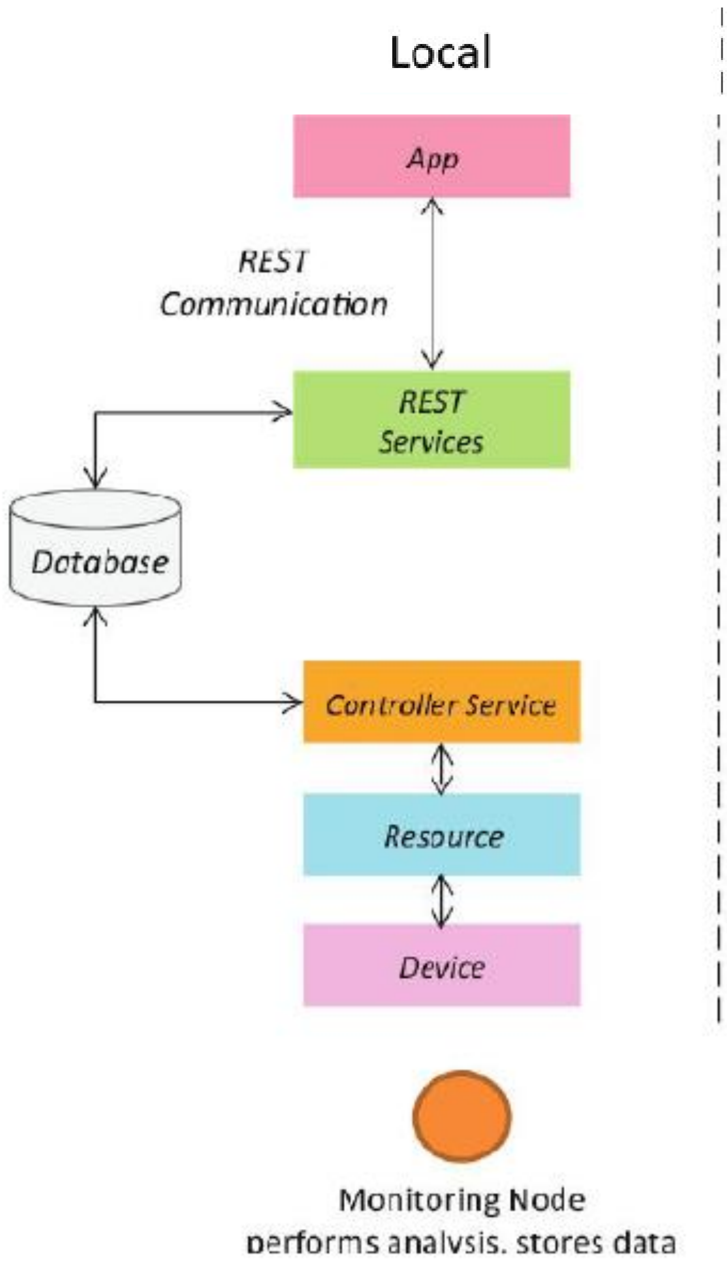


State Service: Sets the light appliances state to ON or OFF.



4.1.6 IOT LEVEL SPECIFICATION

The sixth step in the IoT design methodology is to define the IoT level
for the system. There are 6 IoT deployment levels. The following diagram shows deployment level of the home automation system of Level 1.



4.1.7 Functional View Specifications

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

The Functional Groups(FG) included in a Functional View include:

Device: the device FG contains devices for monitoring and control. In the home automation example, the device FG includes a single board minicomputer, a light sensor and relay switch (actuator)

Communication : The communication FG handles the communication for the IoT system. The communication FG includes the communication protocols that form the backbone of IoT systems and enable network connectivity. The communication API home automation example is a REST based APIs.

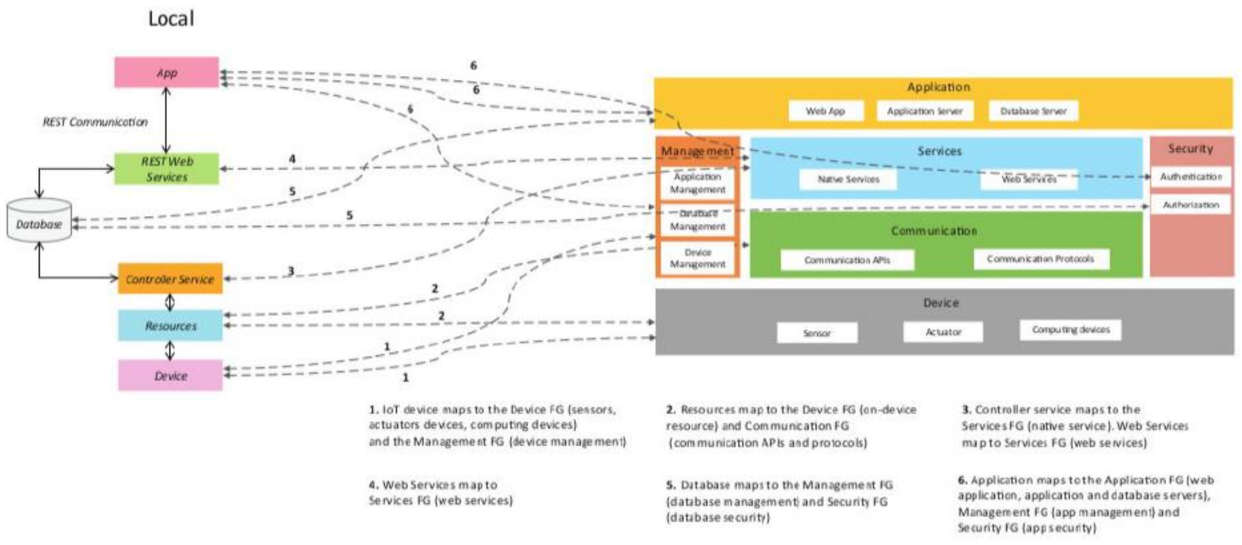
Services :

The service FG includes various services involved in the IoT system such as services for device monitoring , device control services , data publishing services and services for device discovery. In home automation example, there are two REST services (mode and state) and one native service (controller service).

Management: the management FG includes all functionalities that are needed to configure and manage IoT System.

Security: the security FG includes security mechanisms for the IoT system such as authentication, authorization , data security etc.,

Application :the application FG includes applications that provide an interface to the users to control and monitor various aspects of the IoT system. Applications also allow users to view the system status and the processed data.



1. IoT device maps to the Device FG (sensors, actuators devices, computing devices) and the Management FG (device management)

2. Resources map to the Device FG (on-device resource) and Communication FG (communication APIs and protocols)

3. Controller service maps to the Services FG (native service). Web Services map to Services FG (web services)

4. Web Services map to Services FG (web services)

5. Database maps to the Management FG (database management) and Security FG (database security)

6. Application maps to the Application FG (web application, application and database servers), Management FG (app management) and Security FG (app security)

4.1.8 OPERATIONAL VIEW SPECIFICATIONS

The eighth step in the IoT design methodology is to define the Operational View Specifications.
In this step, various options pertaining to the IoT system deployment and operation are defined, such as, service hosting options, storage options, device options, application hosting options, etc
The following figure shows an example of mapping functional groups to operational view specifications for IoT home automation system. Operational view specifications for the home automation example are as follows.

Devices:

Computing Device , Raspberry PI, Light dependent resistor (sensor), really switch (actuator)

Communication APIs

REST APIs

Communication Protocols:

Link Layer-802.11, Network Layer-IPv4/IPv6, Transport Layer –TCP, Application Layer- HTTP

Services:

i)Controller Service- Hosted on device, Implemented Python and run as a native service.
ii)Mode Service-REST-ful web service, hosted on device, implemented with DJango – REST frame work.
iii) State Service- RESTful REST-ful web service, hosted on device, implemented with Django – REST frame work.

Application:

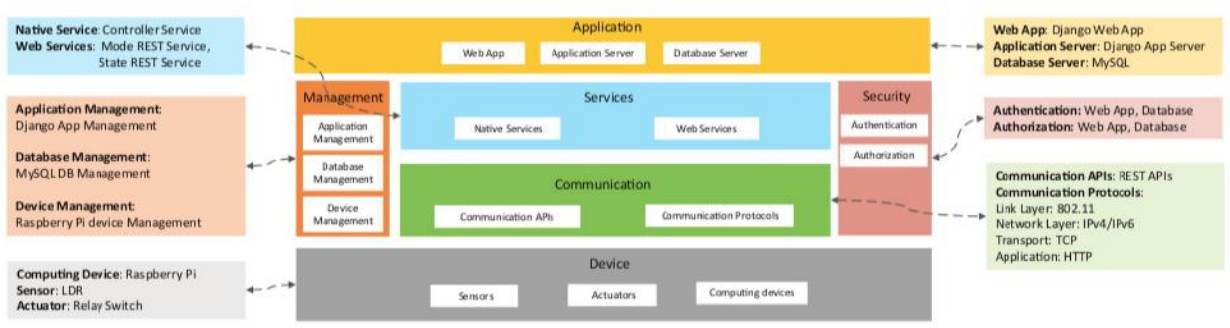
i) Web Application- Django web application ,
ii)Application Server- Django App server
iii) Database Server-MySQL

Security:

i) Authentication- Web App, Database
ii) Authorization- Web App, Database

Management:

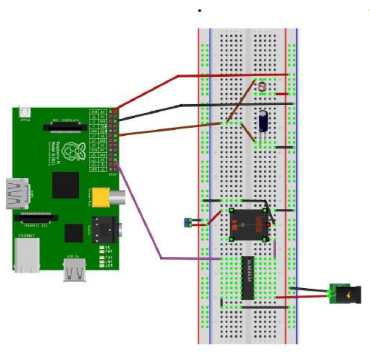
i) Application Management-Django App Management
ii) Database Management- My SQL, DB Management
iii)Device Management- Raspberry Pi Device Management.



4.1..9 DEVICE & COMPONENT INTEGRATION

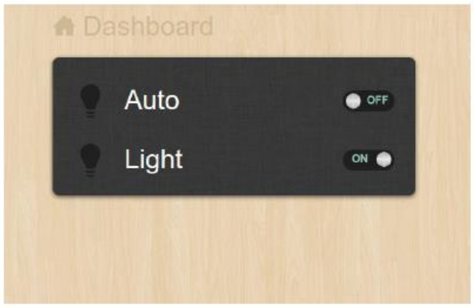
The ninth step in the IoT design methodology is the integration of the devices and components.

The following figure shows the schematic diagram of the IoT Home automation System. The devices and component used in this example are Raspberry Pi mini computer, LDR sensor and relay switch actuators.



4.1.10APPLICATION DEVELOPMENT

The final step in the IoT design methodology is to develop the IoT application. The following figure shows the screenshot of the home automation web application. The application has controls for the mode (auto ON or auto OFF) and the light ON or OFF. In the auto mode, the IoT system controls the light appliance automatically based on the lightning conditions in the room. When auto mode is enable, the light control in the application is disabled and reflects the current state of the light. When the auto mode is disabled, the light control is enabled and is used for manually controlling the light.



• Auto

• Controls the light appliance automatically based on the lighting conditions in the room

Light

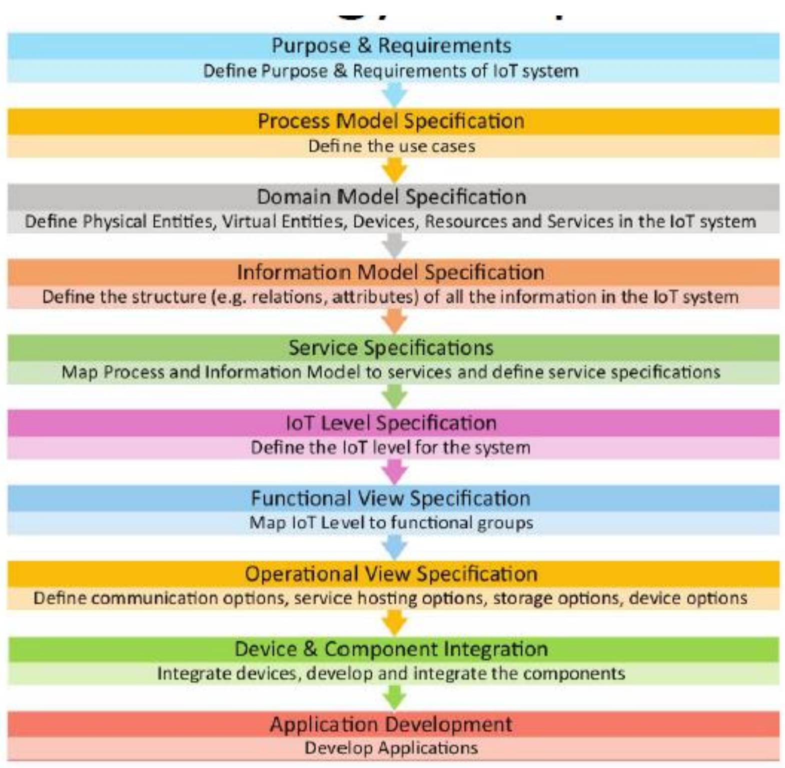
• When Auto mode is off, it is used for manually controlling the light appliance.

• When Auto mode is on, it reflects the current state of the light appliance.

4.2 Case Study: IOT Based Weather Monitoring System

Introduction

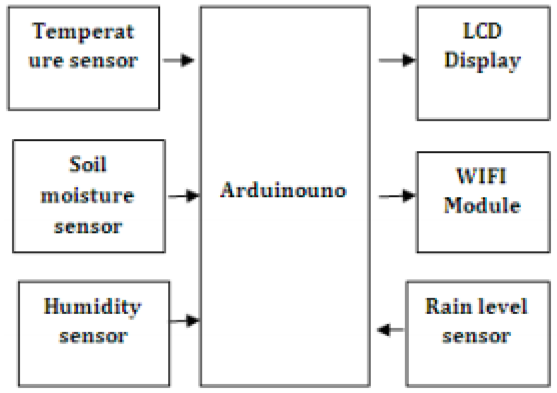
IoT system comprises of multiple components and deployment tier.


Purpose & Requirements
Define Purpose & Requirements of IoT system
Process Model Specification
Define the use cases
Domain Model Specification
Define Physical Entities, Virtual Entities, Devices, Resources and Services in the IoT system
Information Model Specification
Define the structure (e.g. relations, attributes) of all the information in the IoT system
Service Specifications
Map Process and Information Model to services and define service specifications
IoT Level Specification
Define the IoT level for the system
Functional View Specification
Map IoT Level to functional groups
Operational View Specification
Define communication options, service hosting options, storage options, device options
Device & Component Integration
Integrate devices, develop and integrate the components
Application Development
Develop Applications

Climatic change and environmental monitoring have received much attention recently. Man wants to stay updated about the latest weather conditions of any place like a college campus or any other particular building. Since the world is changing so fast so there should be the weather stations.

In IOT enabled weather monitoring system project, Arduino Uno measures four weather parameters using four respective sensors. These sensors are temperature sensor, humidity sensor, moisture sensor and rain level sensor. These four sensors are directly connected to Arduino Uno. Arduino Uno has inbuilt Analog to digital converter. Arduino calculates and displays these weather parameters on LCD display. Then it sends these parameters to Internet using IOT techniques. The process of sending data to the internet using Wi-Fi is repeated after constant time intervals. Then the user needs to visit a particular website to view this weather data. The project connects and stores the data on a web server. Hence user gets Live reporting of weather conditions. Internet connectivity or Internet connection with Wi-Fi is compulsory in this IOT based weather monitoring reporting system

SYSTEM DESIGN:



Rain level sensor: The rain sensor module is an easy tool for rain detection. It can be used as a switch when raindrop falls through the raining board and also for measuring rainfall intensity

Temperature & Humidity Sensor: This DHT11 Temperature and Humidity Sensor features digital signal output .It is integrated with a high-performance 8-bit microcontroller

Soil Moisture Sensor: Soil moisture sensors measure the contents in soil. Since the direct gravimetric measurement of free soil moisture requires removing, drying, and weighting of a sample, soil moisture sensors measure the volumetric water content

WiFi Module: The Arduino Uno WiFi is an Arduino Uno with an integrated WiFi module. The board is based on the ATmega328P with an ESP8266 WiFi Module integrated. T

LCD Display: A Liquid Crystal Display commonly abbreviated as LCD is basically a display unit built using Liquid Crystal technology.

ADVANTAGES:

- IOT weather mentoring system project using Arduino Uno is fully automated.
- It does not require any human attention.
- We can get prior alert of weather conditions
- The low cost and efforts are less in this system Accuracy is high.
- Self Protection
- Smart way to monitor Environment Efficient

APPLICATIONS:

- The weather forecasting plays very important role in the field of agriculture.
- It is also helpful at places like volcano and rain forests.
- It is quite difficult for a human being to stay for longer time at such places.

4.3 IoT Physical Devices and Endpoints: IoT device

- A "Thing" in Internet of Things (IoT) can be any object that has a unique identifier and which can send/receive data (including user data) over a network (e.g., smart phone, smartTV, computer, refrigerator, car, etc.).
- IoT devices are connected to the Internet and send information about themselves or about their surroundings (e.g. information sensed by the connected sensors) over a network (to other devices or servers/storage) or allow actuation upon the physical entities/environment around them remotely.

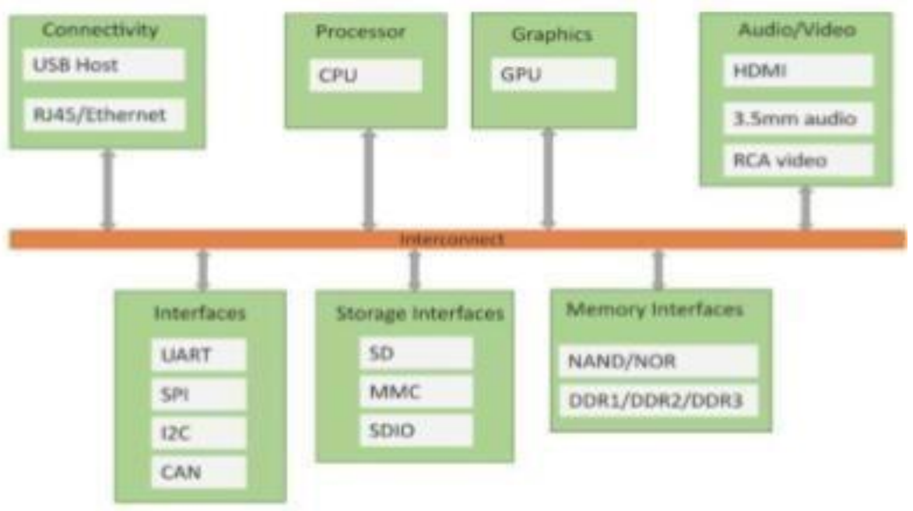
IoT Device Examples

- A home automation device that allows remotely monitoring the status of appliances and controlling the appliances.
- An industrial machine which sends information abouts its operation and health monitoring data to a server.
- A car which sends information about its location to a cloud-based service.
- A wireless-enabled wearable device that measures data about a person such as the number of steps walked and sends the data to a cloud-based service.

4.4 Basic Building Blocks of an IoT Device.

- **Sensing:** Sensors can be either on-board the IoT device or attached to the device . IoT device can collect various types of information from the on board or attached sensors such as temperature, humidity, light intensity, etc
- **Actuation:** IoT devices can have various types of actuators attached that allow taking actions upon the physical entities in the vicinity of the device.
  Example: A Relay switch connected to an IoT device can turn an appliance on/off based on the commands sent to the device.
- **Communication:** Communication modules are responsible for sending collected data to other devices or cloud-based servers/storage and receiving data from other devices and commands from remote applications.
- **Analysis & Processing:** Analysis and processing modules are responsible for making sense of the collected data

Block Diagram of an IoT Device



Expansions

- USB Host-Universal Serial Bus Host
- RJ 45/Ethernet- Component /Port
- CPU- Central Processing Unit

- GPU- Graphical Processor Unit
- HDMI-High-Definition Multimedia Interface Splitter
- RCA Video-Radio Corporation of America Community
- UART- Universal Asynchronous Receiver Transmitter
- SPI-Serial Peripheral Interface
- I2C-Inter Integrated Circuit bus
- CAN-Controller Area Network
- SD-Secondary Storage
- MMC-Multimedia Memory Cards.
- SDIO-Secure Digital Input Output
- NAND/ NOR- Logic Gates
- DDR1/DDR2/DDR3-Double Data Rate

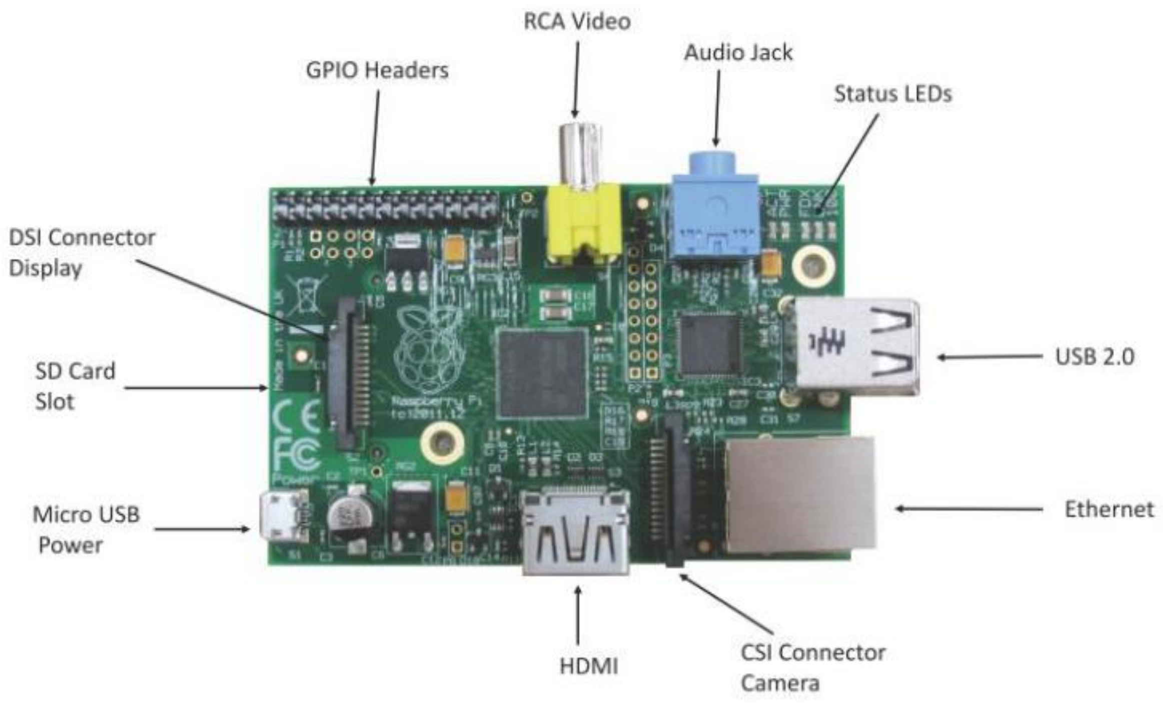
4.5 Raspberry Pi

Raspberry Pi is a low-cost mini-computer with the physical size of a credit card.

• Raspberry Pi runs various flavors of Linux and can perform almost all tasks that a normal desktop computer can do.

• Raspberry Pi also allows interfacing sensors and actuators through the general purpose I/O pins.

• Since Raspberry Pi runs Linux operating system, it supports Python "out of the box".



4.6 Linux on Raspberry Pi

• Raspbian

• Raspbian Linux is a Debian Wheezy port optimized for Raspberry Pi.

• Arch

• Arch is an Arch Linux port for AMD devices.

• Pidora

• Pidora Linux is a Fedora Linux optimized for Raspberry Pi.

• RaspBMC

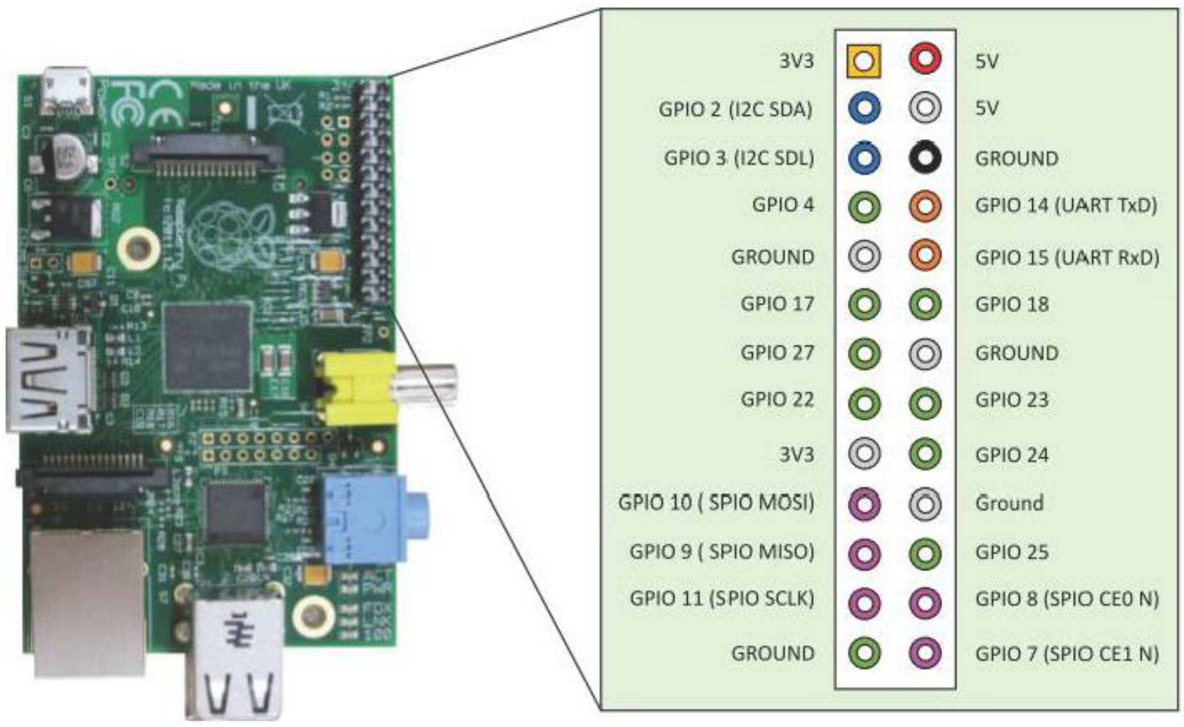
• RaspBMC is an XBMC media-center distribution for Raspberry Pi.

• OpenELEC

• OpenELEC is a fast and user-friendly XBMC media-center distribution.

• RISC OS

• RISC OS is a very fast and compact operating system.



4.7 Raspberry Pi Interfaces

Serial

• The serial interface on Raspberry Pi has receive (Rx) and transmit (Tx) pins for communication with serial peripherals.

• SPI

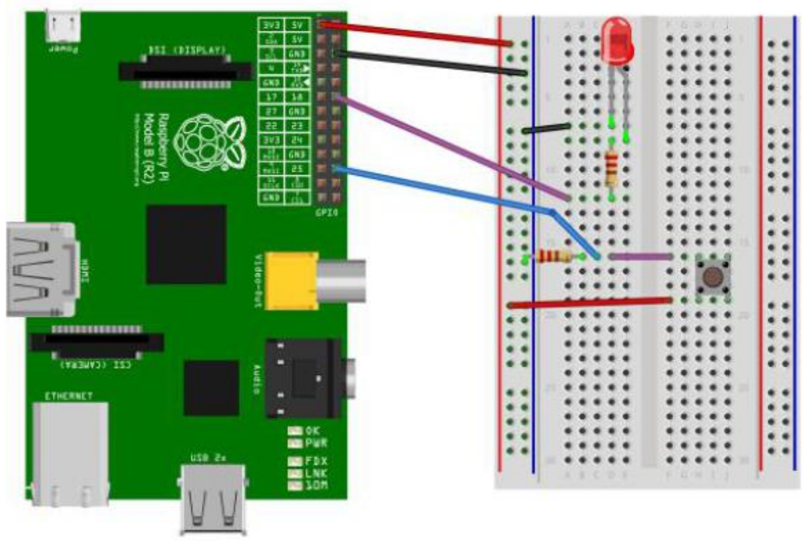
• Serial Peripheral Interface (SPI) is a synchronous serial data protocol used for communicating with one or more peripheral devices.

• 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).

Raspberry Pi Example: Interfacing LED and switch with Raspberry Pi

```
from time import sleep
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
#Switch Pin
GPIO.setup(25, GPIO.IN)
#LED Pin
GPIO.setup(18, GPIO.OUT)
state=false
def toggleLED(pin):
state = not state
GPIO.output(pin, state)
while True:
try:
if (GPIO.input(25) == True):
toggleLED(pin)
sleep(.01)
except KeyboardInterrupt:
exit()
```



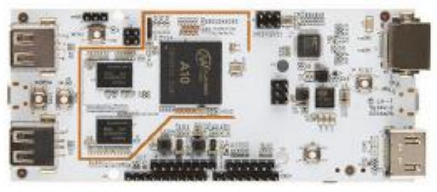
4.8 Programming Raspberry Pi with Python

Raspberry Pi is a low-cost mini-computer with the physical size of a credit card. Raspberry Pi runs various flavors of Linux and can perform almost all tasks that a normal desktop computer can do. Raspberry Pi also allows interfacing sensors and actuators through the general purpose

I/O pins. Since Raspberry Pi runs Linux operating system, it supports Python "out of the box". Raspberry Pi is a low-cost mini-computer with the physical size of a credit card. Raspberry Pi runs various flavors of Linux and can perform almost all tasks that a normal desktop computer can do. Raspberry Pi also allows interfacing sensors and actuators through the general purpose I/O pins. Since Raspberry Pi runs Linux operating system, it supports Python "out of the box".

4.9 Other IoT Device

• pcDuino • BeagleBone Black • Cubieboard



Part A-Multiple Choice Questions

[Separately discussed]

Part B- 7 Marks

1. Discuss in detail about Raspberry Pi Interfaces
2. Describe the basic building blocks of an IoT Device with diagram
3. Discuss in detail about Linux on Raspberry Pi

Part C- 16 Marks

1. Write about the Process Specification of IoT Design Methodology
2. Write about the Process Specification of IoT System for Weather Monitoring