Unit– IV
IoT Physical Device & Cloud Platforms

4.1 IoT device: Building Blocks of an IoT device.
4.2 Exemplary Physical Device: Raspberry Pi, About the Raspberry Pi board, Linux on Raspberry Pi, Configuring Raspberry Pi, Raspberry Pi interfaces: Serial, SPI, I2C, and Programming Raspberry Pi with Python.
4.3 Other IoT Devices: pcDulno, BeagleBone Black, Cubieboard.
4.4 Role of cloud in IoT, Cloud based IoT platforms and other open- source platforms.
4.5 Introduction to cloud storage models & communication APIs, WAMP.
4.6 Web server for IoT, Cloud for IoT.

IoT DEVICES:

IoT devices are pieces of hardware, such as sensors, actuators, gadgets, appliances, or machines, that are programmed for certain applications and can transmit data over the internet or other networks.

Some examples of IoT devices are listed below:

- A home automation device that allows remotely monitoring the status of appliances and controlling the appliances.
- An industrial machine which sends information about 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 the number of steps walked and sends the data to a cloud-based service.

Basic building blocks of an IoT Device:

An IoT device can consist of a number of modules based on functional attributes. such as:

- Sensing: Sensors can either on-board the IoT device or 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. The sensed information can communicated either to other devices or cloud-based servers/storage.
- Actuation: IoT devices can have various types of actuators attached that allow taking actions upon the physical entities in the vicinity of the device. For 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.

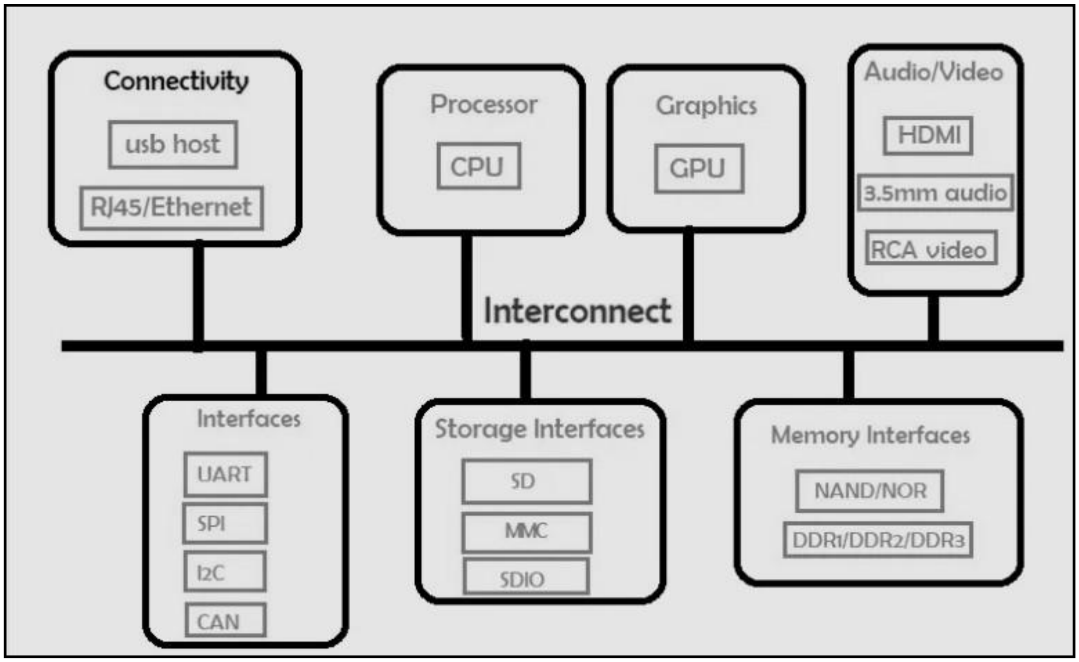
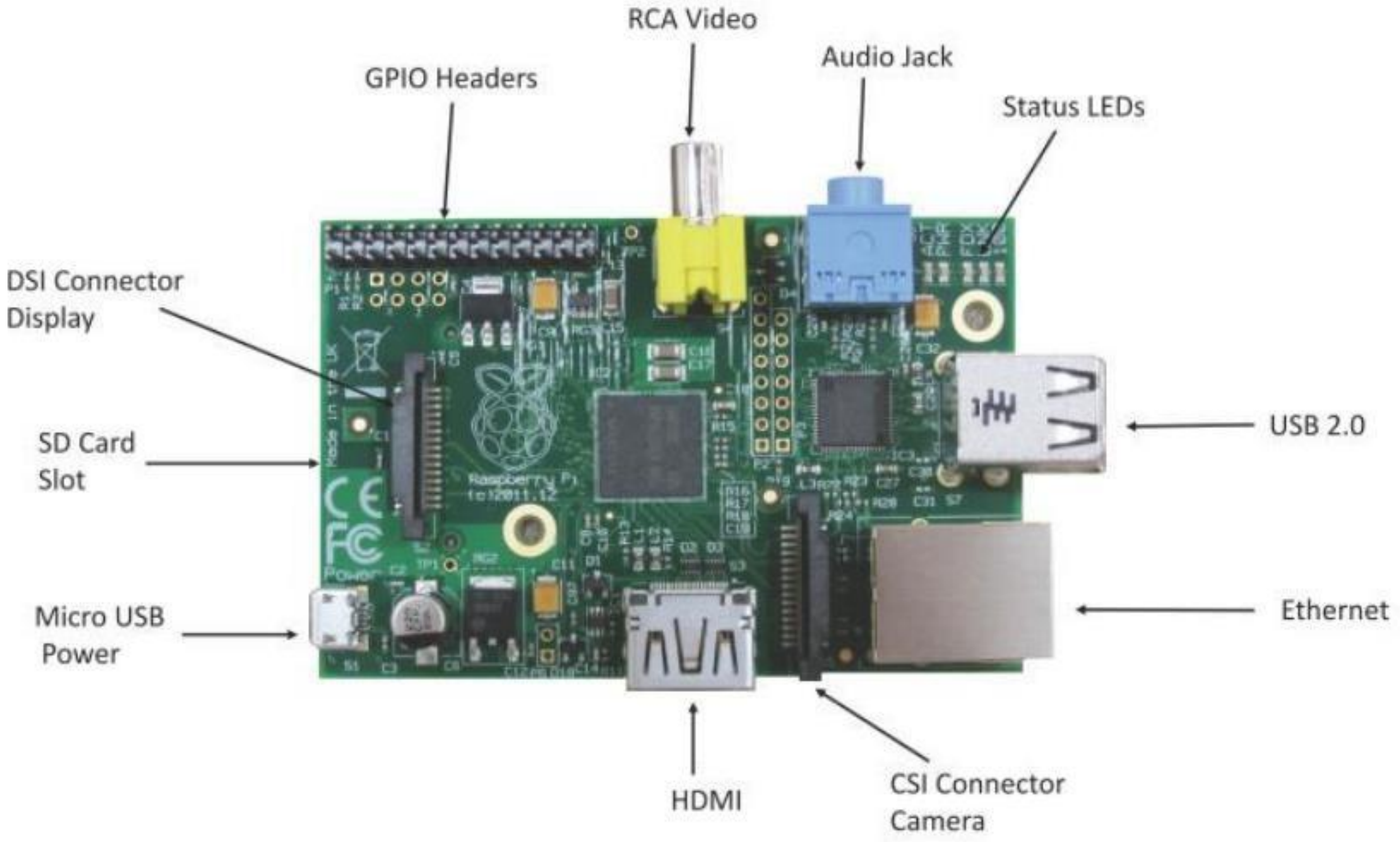


Fig: **Block Diagram of an IoT device**

Exemplary Device: RaspberryPi

- 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".

About the Raspberry Pi board:



- Processor & RAM : Raspberry Pi is based on an ARM Processor.
- USB ports : Raspberry Pi comes with two USB 2.0 ports. The USB ports on Raspberry Pi can provide a current upto 100mA.
- Ethernet Ports : Raspberry Pi comes with a standard RJ45 Ethemet port.
- HDMI Output : The HDMI port on Raspberry Pi provides both video and audio output.
- Composite Video Output : Raspberry Pi comes with a composite video output with an RCA jack.
- Audio Output : Raspberry Pi has a 3.5mm audio output jack. This audio jack is used for providing audio output to old televisions along with the RCA jack for video.
- GPIO Pins : Raspberry Pi comes with a number of general purpose input/ouput pins.
- Display Serial Interface (DSI) : DSI interface can be used to connect an LCD panel to RaspbetTY Pi.
- Camera Serial Interface (CSI) : The CSI interface can be used to connect a camera module to Raspberry Pi.
- Status LEDs : Raspberry Pi has five status LEDs.
  SD Card Slot : Raspberry Pi does not have a built in operating system and storage.
- Power Input: Raspberry Pi has a micro-USB connector for power input.

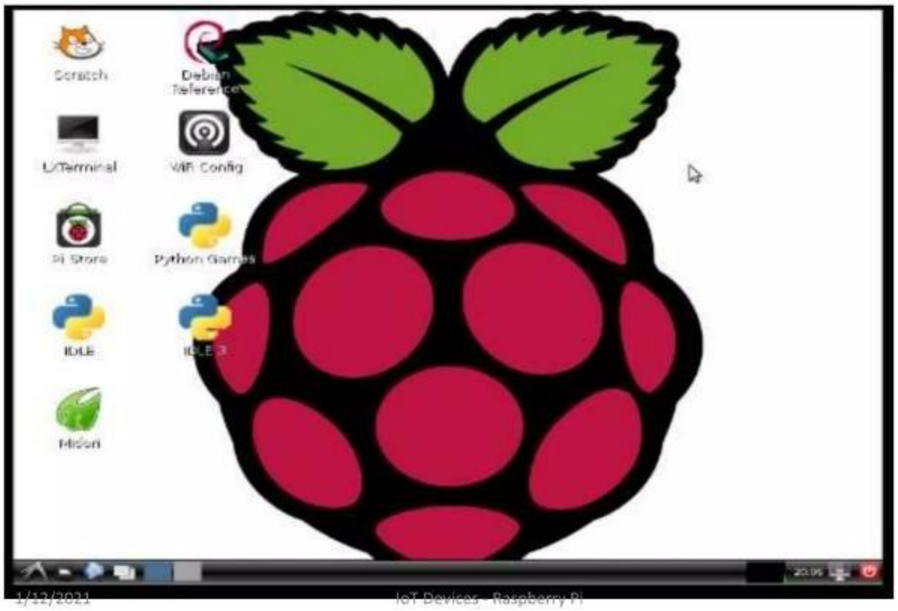
Linux on Raspberry Pi

Raspberry Pi supports various flavours of Linux including:

- **Raspbian** Linux is a Debian Wheezy port optimized for Raspberry Pi.
- **Arch** is an Arch Linux port for AMD devices.

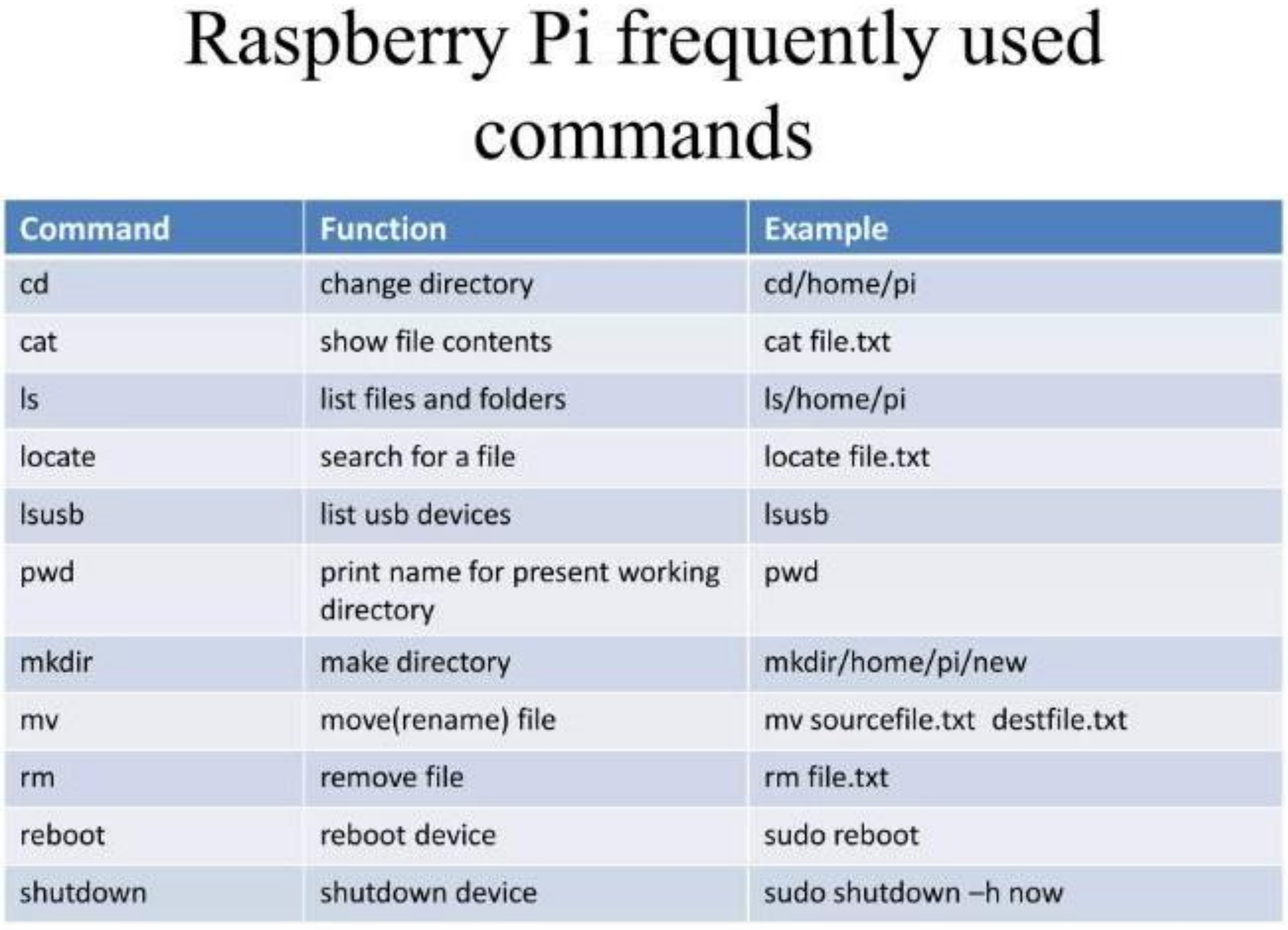
- **Pidora** Linux is a Fedora Linux optimized for Raspberry Pi.
- **RaspBMC** is an XBMC media-center distribution for Raspberry Pi.
- **OpenELEC** is a fast and user-friendly XBMC media-center distribution.
- **RISC OS** is a very fast and compact operating system.

Rasbian Linux Desktop



1/12/2021 IoT Devices - Raspberry Pi

Raspberry frequently used commands:



Raspberry Pi frequently used commands

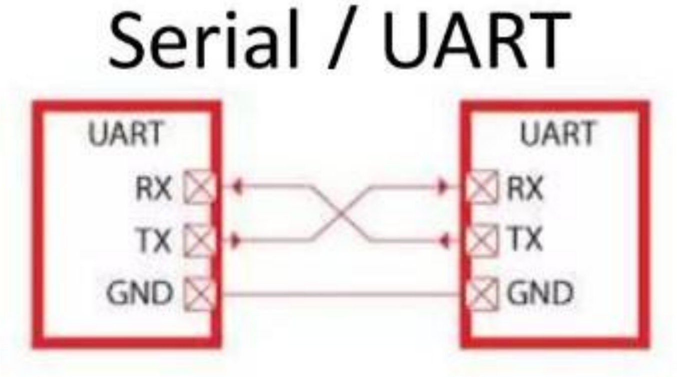
Command	Function	Example
cd	change directory	cd/home/pi
cat	show file contents	cat file.txt
ls	list files and folders	ls/home/pi
locate	search for a file	locate file.txt
lsusb	list usb devices	lsusb
pwd	print name for present working directory	pwd
mkdir	make directory	mkdir/home/pi/new
mv	move(rename) file	mv sourcefile.txt destfile.txt
rm	remove file	rm file.txt
reboot	reboot device	sudo reboot
shutdown	shutdown device	sudo shutdown –h now

Raspberry Pi Interfaces

- Serial Interface / UART Interface

— Universal Asynchronous Receiver and Transmitter(UART)

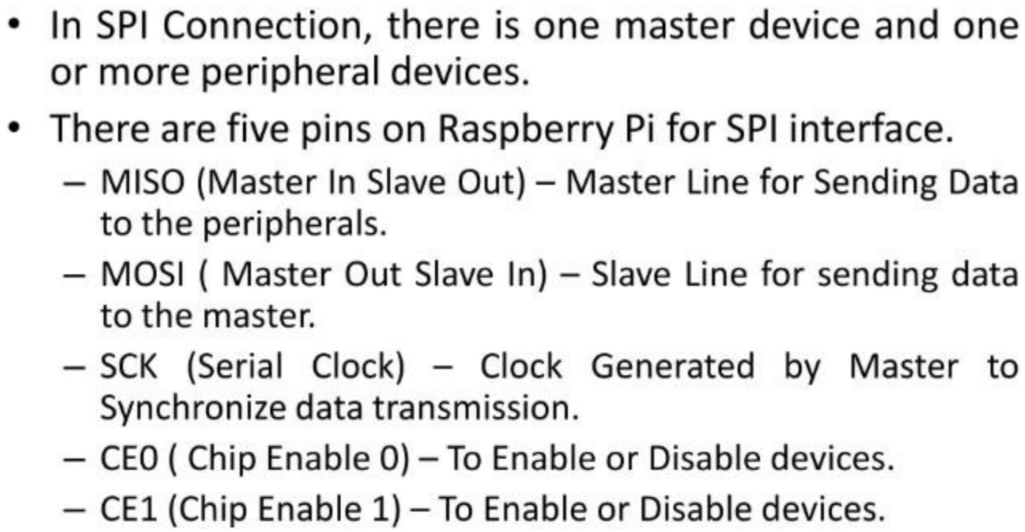
➢ The serial interface on Raspberry Pi has receive (Rx) and transmit (Tx) pins for communication with serial peripherals.



- SPI

— Serial Peripheral Interface (SPI)

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



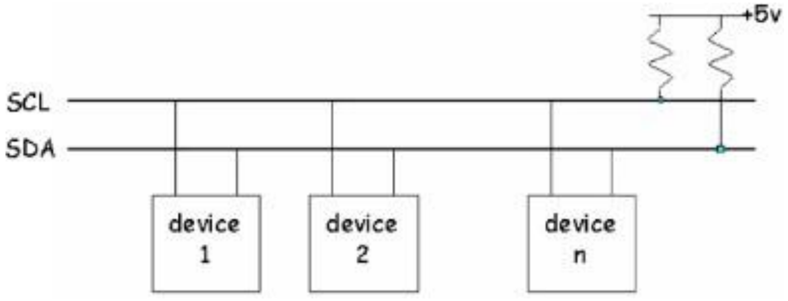
- In SPI Connection, there is one master device and one or more peripheral devices.
- There are five pins on Raspberry Pi for SPI interface.
  - – MISO (Master In Slave Out) – Master Line for Sending Data to the peripherals.
  - – MOSI (Master Out Slave In) – Slave Line for sending data to the master.
  - – SCK (Serial Clock) – Clock Generated by Master to Synchronize data transmission.
  - – CE0 (Chip Enable 0) – To Enable or Disable devices.
  - – CE1 (Chip Enable 1) – To Enable or Disable devices.

- 12C

— Inter-lntegrated Circuits (12C)

➢ 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)



Programming Raspberry Pi with Python.

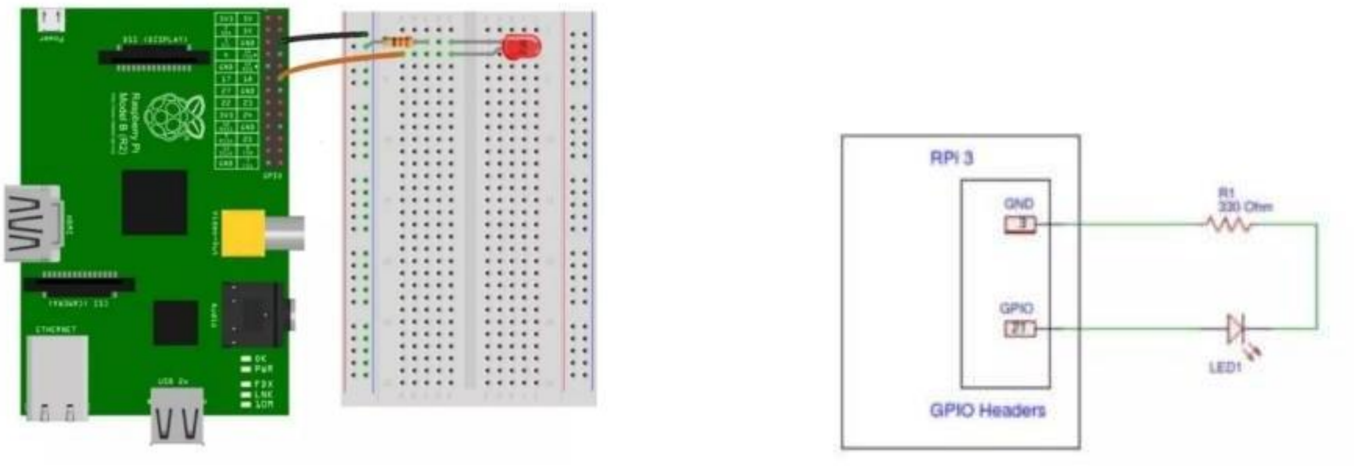
Python programming with Raspberry PI with focus of

— Interfacing external gadgets
— Controlling output
— Reading input from pins

- GPIO pins on Raspberry Pi that makes it useful device for IOT.
- We can interface a wide variety of sensors and actuators with Raspberry Pi using the GPIO pins and SPI, 12C and Serial Interfaces.
- Input from the sensors connected to Raspberry Pi can be processed and various actions can be taken, for

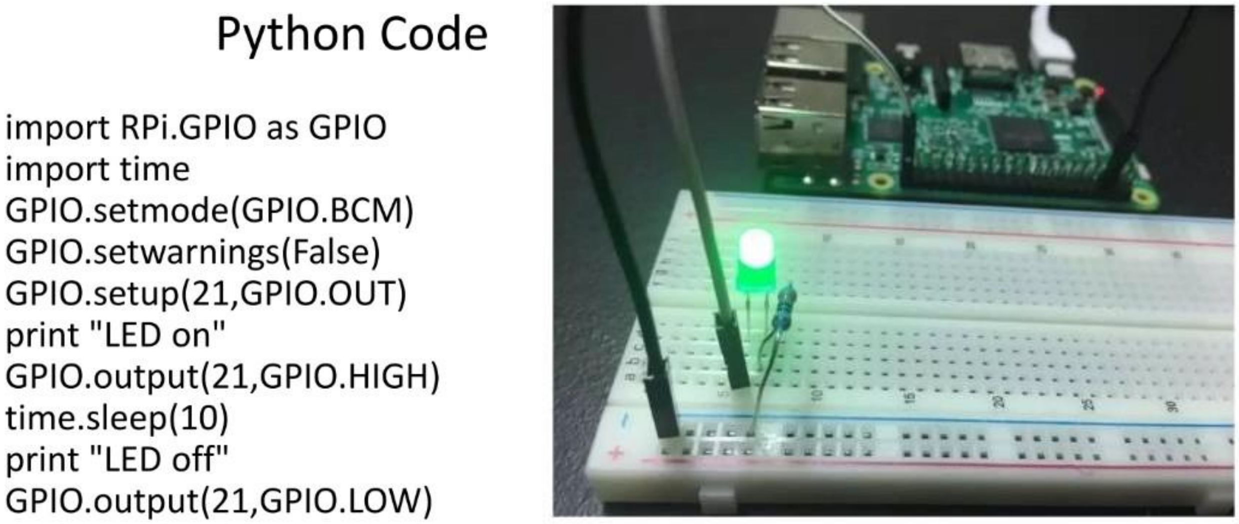
  — Instance,
  — Sending data to a server
  — Sending an email
  — Triggering a relay switch

Controlling LED with Raspberry Pi



Components Required the following components to connect the circuit.

- Raspberry Pi
- LED
- Resistor - 330 ohm
- Breadboard
- 2 Male-Female Jumper Wires

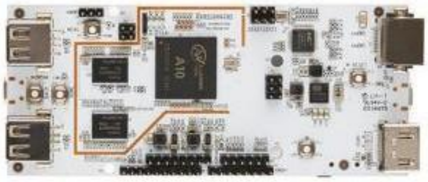


Python Code

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(21,GPIO.OUT)
print "LED on"
GPIO.output(21,GPIO.HIGH)
time.sleep(10)
print "LED off"
GPIO.output(21,GPIO.LOW)
```

Other IoT Devices

- PcDuino
- BeagleBone Black
- Cubieboard



pcDuino

pcDuino [1051 is an Arduino-pin compatible single board mini-computer that comes with a 1 GHz **ARM Cortex-A8 processor**. pcDuino is a high performance and cost effective device. That runs PC like OS such as Ubuntu and Android ICS. Like Raspberry Pi, it has HDMI video/audio interface .It support various programming languages such as C, C++, Java and Python.

BeagleBone Black

It is similar to Raspberry Pi, but more powerful device. It comes with 1 GHz **ARM Cortex-A8 processor** and supports both Linux and Android operating systems. Like Raspberry Pi, it has HDMI video/audio interface, USB and Ethernet ports.

Cubieboard

Cubieboard is powered by a dual core ARM Cortex A7 processor and has a range of input/output interfaces including USB, IR, serial, Ethernet, SATA, and 96 pin extended interface. The board supports both Linux and Android operating systems

Role of Cloud in IoT.

- Provides remote processing power:
  The cloud has such a vast storage that it takes away dependencies on on-premise infrastructure.
- Provides security and privacy:

Cloud has made IoT more secure with preventive, detective and corrective controls.

- Removes entry barrier for hosting providers:
  With the cloud, most hosting providers can allow their clients a ready-to-roll model, removing entry barriers for them.
- Facilitates inter-device communication:
  Cloud acts as a bridge in the form of a mediator or communication facilitator when it comes to IoT.

Cloud based IoT platforms:

- Amazon Web Services IoT
- IBM Watson IoT Platform
- Microsoft Azure IoT Hub
- Google Cloud IoT
- Oracle Integrated Cloud for IoT
- SAP Cloud Platform for the Internet of Things
- Cisco Jasper Control Center
- PTC ThingWorx Industrial IoT Platform
- Salesforce IoT
- Xively

Amazon Web Services IoT -

- AWS IoT provides the cloud services that connect your IoT devices to other devices and AWS cloud services. AWS IoT provides device software that can help you integrate your IoT devices into AWS IoT-based solutions. If your devices can connect to AWS IoT, AWS IoT can connect them to the cloud services that AWS provides.
- AWS IoT Core can support billions of devices and trillions of messages, and can process and route those messages to AWS endpoints and to other devices reliably and securely.

IBM Watson IoT Platform -

- IBM Watson IoT Platform is a managed, cloud-hosted service designed to make it simple to derive value from your Internet of Things devices.

Microsoft Azure IoT Hub -

- Azure IoT Hub provides a cloud-hosted solution back end to connect virtually any device. Extend your solution from the cloud to the edge with per-device authentication, built-in device management and scaled provisioning. Security-enhanced communication channel for sending and receiving data from IoT devices.
- You can connect millions of devices and their backend solutions reliably and securely. Almost any device can be connected to an IoT hub.

Google Cloud IoT –

- Google Cloud Internet of Things (IoT) Core is a fully managed service for securely connecting and managing IoT devices, from a few to millions. Ingest data from connected devices and build rich applications that integrate with the other big data services of Google Cloud Platform.

- Using Google Cloud technologies, combined with partners, our customers can customize specific IoT solutions to meet their unique needs.

open- source IoT platforms:

OpenRemote:

- integration of different asset types,
- support of standard protocols such as HTTP or MQTT as well as specific protocols like KNX for connecting IoT devices,
- a customizable manager interface for the automation, monitoring, and control of processes, apps, and devices,
- a mobile app so that you can connect to your phone services and push notifications,

ThingsBoard:

- IoT device management with monitoring and control mechanisms,
- scalability with the capability to orchestrate multiple devices at the same time, across the entire IoT ecosystem
- enables users to create and manage alerts for connected devices (e.g. in the event of a disconnect or inactivity), other assets, and customers with real-time alarm monitoring.
- extending default functionality with customizable rule chains, widgets, and transport implementations,
- multi-tenancy

Thinger.io:

We can connect any device, the most typical being Arduino, ESP8266, Raspberry Pi, and Intel Edison. You simply install the server in your own cloud and use the open-source libraries for integrating the IoT devices.

MainFlux:

Mainflux provides the complete infrastructure and middleware to execute:

- Device management
- Data aggregation & data management
- Connectivity & message routing
- IoT application enablement
- Analytics

Introduction to cloud storage models & communication APIs

Cloud computing is a transformative computing paradigm that involves delivering applications and services over the internet. NIST defines cloud computing as- Cloud computing is a model for enabling ubiquitous, convenient. On-demand network access to a shared pool of configurable computing resources (eg.. networks, servers. storage. applications. and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

WAMP – AutoBahn for IoT
Web Application Messaging Protocol

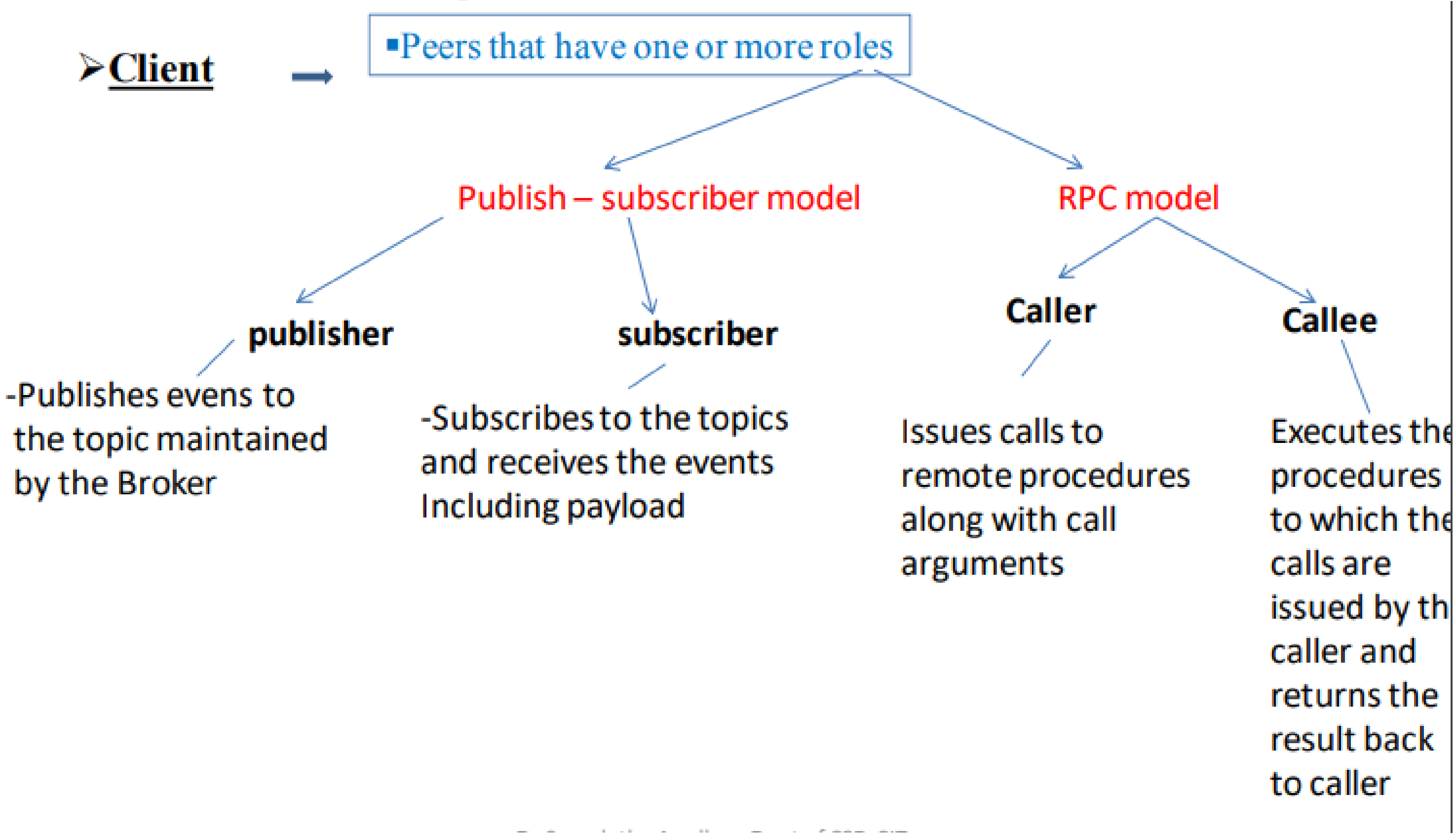
❖**Key concepts of WAMP are:**

➢**Transport**

▪Connects two peers
▪Default transport is WebSocket
▪Support message based reliable bi –directional communication

- **Session**
  - Conversation between two peers
  - Runs over a Transport

- **Client** ➡



▪Peers that have one or more roles

Publish – subscriber model

RPC model

publisher

-Publishes evens to the topic maintained by the Broker

subscriber

-Subscribes to the topics and receives the events Including payload

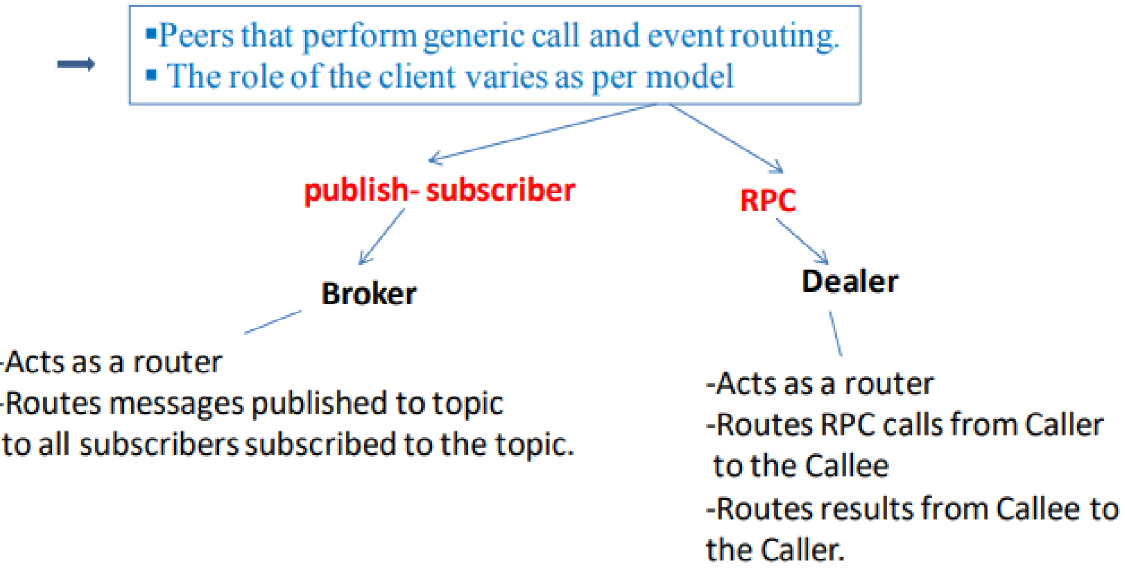
Caller

Issues calls to remote procedures along with call arguments

Callee

Executes the procedures to which the calls are issued by th caller and returns the result back to caller

- **Router** ➡



▪Peers that perform generic call and event routing.
▪ The role of the client varies as per model

publish- subscriber

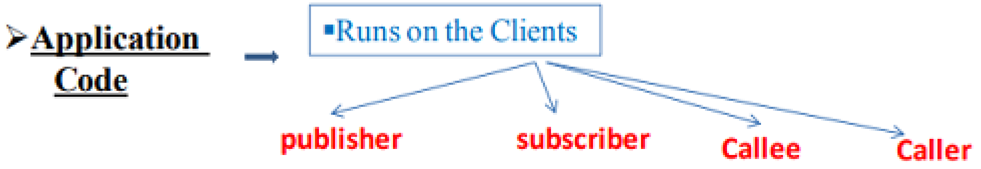
RPC

Broker

-Acts as a router
-Routes messages published to topic to all subscribers subscribed to the topic.

Dealer

-Acts as a router
-Routes RPC calls from Caller to the Callee
-Routes results from Callee to the Caller.



- **Application Code** →

▪Runs on the Clients

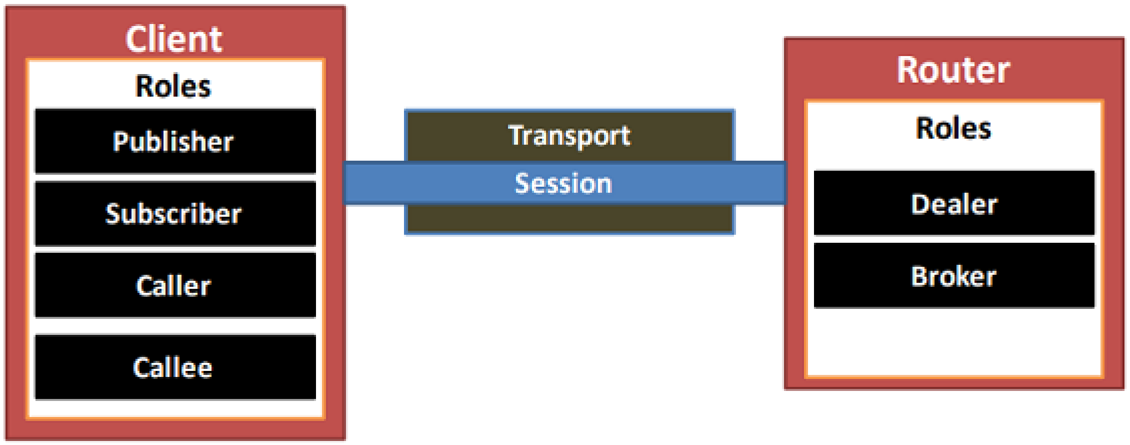
publisher

subscriber

Callee

Caller

WAMP Session between Client and Router:



Client

Roles

Publisher

Subscriber

Caller

Callee

Transport

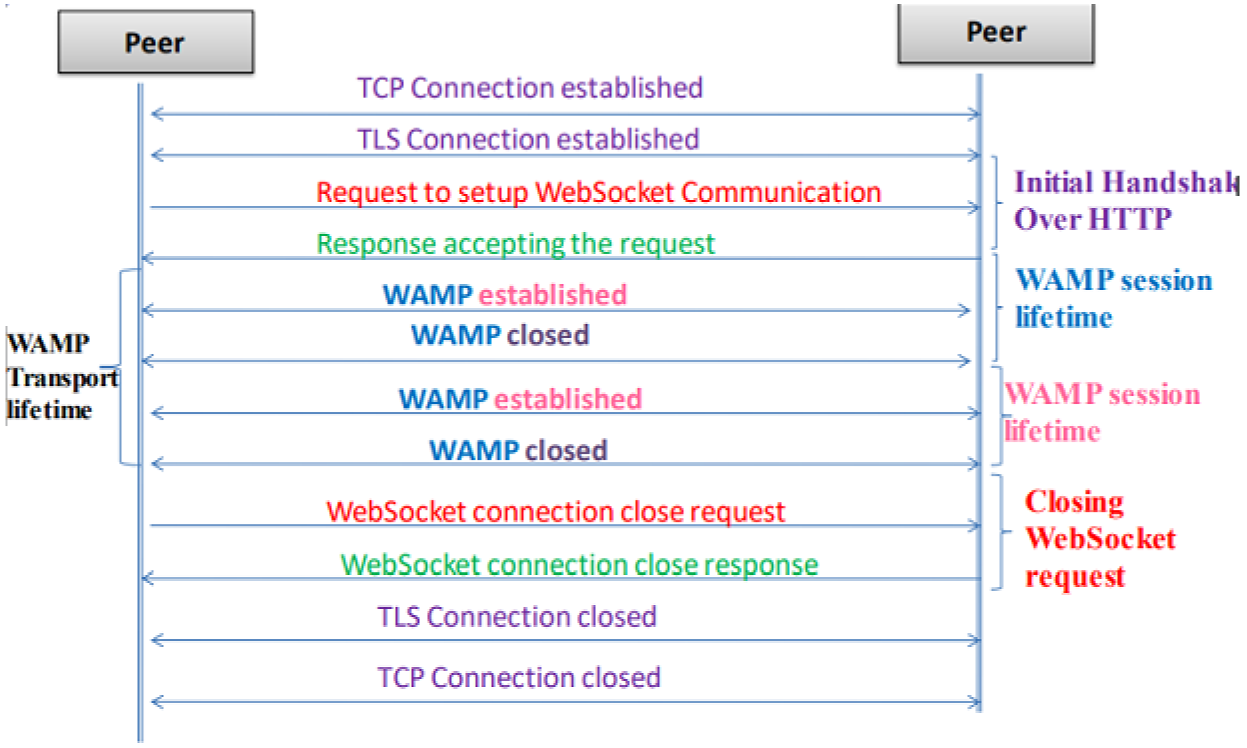
Session

Router

Roles

Dealer

Broker



Peer

Peer

TCP Connection established

TLS Connection established

Request to setup WebSocket Communication

Response accepting the request

WAMP established

WAMP closed

WAMP established

WAMP closed

WebSocket connection close request

WebSocket connection close response

TLS Connection closed

TCP Connection closed

Initial Handshak
Over HTTP

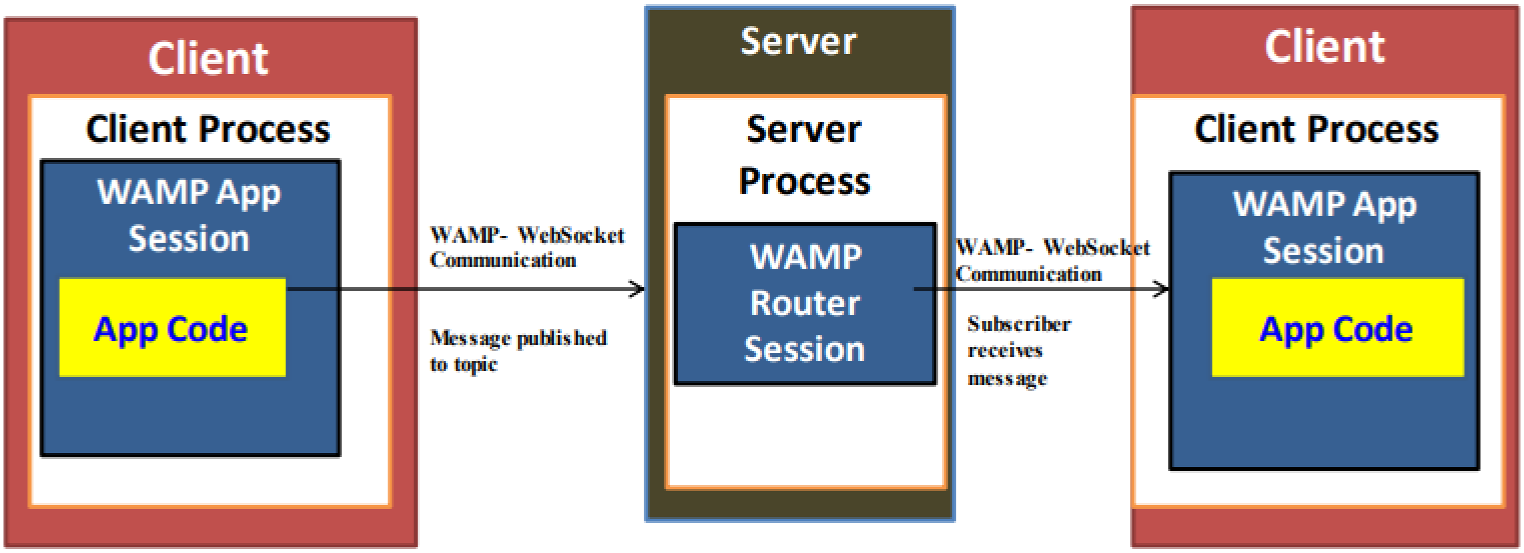
WAMP session
lifetime

WAMP session
lifetime

Closing
WebSocket
request

WAMP
Transport
lifetime

Publish – Subscribe messaging WAMP AutoBahn :



Cloud for IoT

An IoT cloud is a massive network that supports IoT devices and applications. This includes the underlying infrastructure, servers and storage, needed for real-time operations and processing. An IoT cloud also includes the services and standards necessary for connecting, managing, and securing different IoT devices and applications.

Use of Cloud IoT cloud-based service

- The service provides for the data collection, data points, messages and calculation objects.
- The service also provisions for the generation and communication of alerts, triggers and feeds to the user.
- A user is an application or service. The user obtains responses or feeds from the cloud service.

Xively:

Xively(formerly known as Cosm and Pachube) is an Internet of Things (IoT) platform owned by Google

- Pachube platform: for data capture in real-time over the Internet
- Cosm: a changed domain name, where using a concept of console, one can monitor the feeds
- Xively is the latest domain name
- A commercial Platform as a Servive(PaaS) for the IoT/M2M
- A data aggregator and data mining website often integrated into the Web of Things
- An IoT PaaS for services and business services.

Xively PaaS services

- Data visualisation for data of connected sensors to IoT devices.
- Graphical plots of collected data.
- Generates alerts.
- Access to historical data
- Generates feeds which can be real-world objects of own or others.

Xively HTTP based APIs

- Easy to implement on device hardware acting as clients to Xively web services
- APIs connect to the web service and send data.
- APIs provides services for logging, sharing and displaying sensor data of all

Xively Support

- The platform supports the REST, WebSockets and MQTT protocols and connects the devices to Xively Cloud Services
- Native SDKs for Android, Arduino, ARM mbed, Java, PHP, Ruby, and Python languages
- Developers can use the workflow of prototyping, deployment and management through the tools provided at Xively.