ACDS Lecture Series

Lecture - 18

CSIR

Internet of Things (IoT)

G. N. Sastry and Team

ADVANCED COMPUTATION AND DATA SCIENCES (ACDS) DIVISION

CSIR-North East Institute of Science and Technology, Jorhat, Assam, India

18. Internet of Things

Defining IoT

Physical object + Controller, sensors and actuators + Internet = Internet of Things

Kevin Ashton
©twitter

History

- The concept of IoT was first given by Kevin Ashton, co-founder Auto-ID lab, MIT in 1999.
- It was formally proposed by International Telecommunication Union (ITU) in 2005.

Characteristics

- Intelligence
- Dynamic nature
- Sensing
- Security
- Connectivity
- Enormous scale
- Heterogeneity

18.1.4 Time series

- **1999-** Neil Gershenfeld gave IoT principles in his book titled "When Things Start to Think" MIT Auto-ID Lab, originally founded by Kevin Ashton, David Brock and Sanjay Sarma, helped to develope the Electronic Product Code.
- **2000-** LG announced its first Internet of refrigerator plans.
- **2002-** The Ambient Orb created by David Rose and others in a spin-off from the MIT Media Lab is released into wild with NY Times Magazine naming it as one of the Ideas of Year .
- **2003-2004-** RFID is deployed on a massive scale by the US Department of Defense in their Savi program and Wal-Mart in the commercial world .
- **2005-** The UN's International Telecommunications Union (ITU) published its first report on the Internet of Things topic .

18.1.4 Time series

- **2008-** Recognition by the EU and the First European IoT conference is held Companies launched the IPSO Alliance to promote the use of IP in networks of “Smart Objects” and to enable the Internet of Things. The IoT was born according to Cisco’s Business Solutions Group US National Intelligence Council listed the IoT as one of the 6 “Disruptive Civil Technologies” with potential impacts on US interests out to 2025.
- **2009**- IBM presented “”Smart Planet”, which aims to embed sensors in several physical objects.
- **2010-** Chinese Premier Wen Jiabao calls the IoT a key industry for China and has plans to make major investments in Internet of Things.
- **2011-** IPv6 public launch-The new protocol allows for 340, 282, 366, 920, 938, 463, 463, 374, 607, 431,768, 211, 456 (2128) addresses.

18.1.5 Requirements for seamless IoT commuting

Hardware
- composed of sensors, actuators and embedded communication hardware

Middleware
- on demand storage and computing tools for data analytics with cloud and Big Data Analytics

Presentation
- easy to understand visualization and interpretation tools that can be designed for the different applications.

18.21 Physical design of IoT

Things in IoT

An embedded computing device that transmits and receives information over a network.

Components of ‘Things’ in IoT

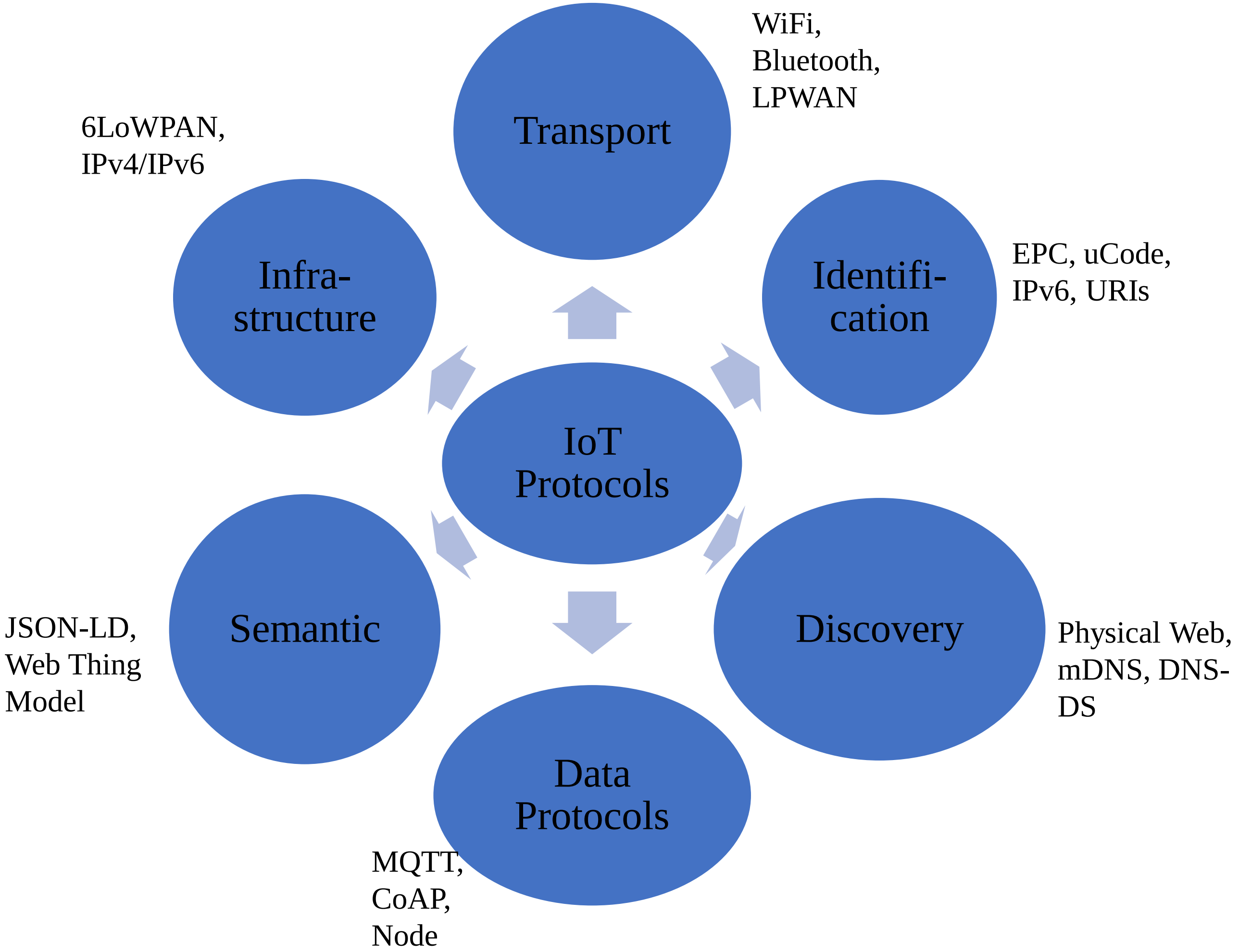
Microcontroller: Engines of countless sensors and automated factory machinery

Sensor: Ways to get information into your device

Actuator: Outputs for the device

18.2.2 IoT Protocols

IoT communication protocols are modes of communication that protect and ensure optimum security to the data being exchanged between connected devices.



18.2.2.1 Infrastructure

- IPv6/ IPv4
  - Developed by Internet Engineering Task Force (IETF) in 1998.
  - Internet Layer protocol for packet-switched internet working and provides end-to-end datagram transmission across multiple IP networks.
  - IPv4 considered the “backbone of modern internet ”and IPv6 considered the “next generation of the internet”
  - IPv4 is a connectionless protocol used in packet-switched layer networks, such as Ethernet.

IPv4 Datagram Header

<table>
<tr><th>Version
4 bits</th><th>HLEN
4 bits</th><th colspan="2">Type of service
8 bits</th><th>Total length
16 bits</th><td>32 bits = 4 bytes</td></tr>
<tr><td>Identification
16 bits</td><td>0(Null bit)
1 bit</td><td>DF(Don’t fragment)
1 bit</td><td>MF (More fragment)
1 bit</td><td>Fragment offset
13 bits</td><td>4 bytes</td></tr>
<tr><td>Time to leave
8 bits</td><td>Protocol
8 bits</td><td colspan="3">Header checksum
16 bits</td><td>4 bytes</td></tr>
<tr><td colspan="5">Source IP
32 bits</td><td>4 bytes</td></tr>
<tr><td colspan="5">Destination IP
32 bits</td><td>4 bytes</td></tr>
<tr><td colspan="5">Option
0 to 40 bytes</td><td></td></tr>
<tr><td colspan="5">Data/ Padding
32 bits</td><td>DATA = 20 bytes to 65,536 bytes</td></tr>
</table>

20 bytes

Max header = 60 bytes
Min header = 20B

18.2.2.1 Infrastructure

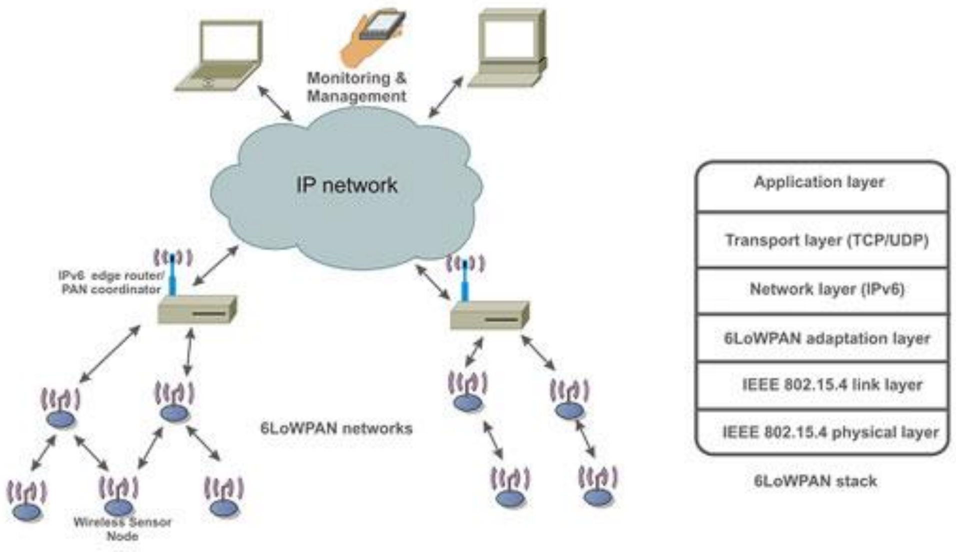
IPv6 Datagram Header

Fixed header

Version **4 bits**	**Priority/Traffic Class** **8 bits**	**Flow Label** **20 bits**
Payload length 16 bits	Next header 8 bits	Hop Limit 8 bits
Source Address 128 bits		
Destination Address 128 bits		
Extension Header		

6LoWPAN

- An acronym of IPv6 over Low power Wireless Personal Area Networks.
- 6LoWPAN provides the upper layer system for use with low power wireless communications for IoT and M2M, originally intended for 802.15.4, it is now used with many other wireless standards.
- This protocol operates only in the 2.4 GHz frequency range with 250 kbps transfer rate.
- Application areas: General automation, home automation, smart grid, industrial monitoring.



18.2.2.2 Discovery

- **Multicast Domain Name System (mDNS)**
  - Resolves host names to IP addresses within small networks that do not include a local name server.
  - mDNS is a joint effort by participants of the IETF Zero Configuration Networking (zeroconf) and DNS Extensions (dnsext) working groups.
  - mDNS uses domain names with the .local suffix, example http://esp8266.local.
  - An IP address can be searched in Arduino using mDNS.

ESP8266 Arduino Core supports mDNS:

```
#include <ESP8266WiFi.h>        // Include the Wi-Fi library

#include <ESP8266WiFiMulti.h>   // Include the Wi-Fi-Multi
library

#include <ESP8266mDNS.h>        // Include the mDNS library

ESP8266WiFiMulti wifiMulti;     // Create an instance of the
ESP8266WiFiMulti class, called 'wifiMulti'

void setup() {

Serial.begin(115200);         // Start the Serial communication to
send messages to the computer

delay(10);

Serial.println('\n');

wifiMulti.addAP("ssid_from_AP_1", "your_password_for_AP_1");
// add Wi-Fi networks you want to connect to

wifiMulti.addAP("ssid_from_AP_2", "your_password_for_AP_2");

wifiMulti.addAP("ssid_from_AP_3", "your_password_for_AP_3");

Serial.println("Connecting ...");
  int i = 0;
  while (wifiMulti.run() != WL_CONNECTED) { // Wait for the Wi-Fi
to connect: scan for Wi-Fi networks, and connect to the strongest of
the networks above
    delay(1000);
    Serial.print(++i); Serial.print(' ');
  }
  Serial.println('\n');
  Serial.print("Connected to ");
  Serial.println(WiFi.SSID());              // Tell us what network we're
connected to
  Serial.print("IP address:\t");
  Serial.println(WiFi.localIP());           // Send the IP address of the
ESP8266 to the computer

  if (!MDNS.begin("esp8266")) {             // Start the mDNS responder
for esp8266.local
    Serial.println("Error setting up MDNS responder!");
  }
  Serial.println("mDNS responder started");
}

void loop() { }
```

18.2.2.2 Discovery

Upload it and open ping again. Try to ping to *esp8266.local:*

- **user@computername**:~$ ping esp8266.local
- PING esp8266.local (10.92.237.128) 56(84) bytes of data.
- 64 bytes from 10.92.237.128: icmp_seq=1 ttl=128 time=5.68 ms
- 64 bytes from 10.92.237.128: icmp_seq=2 ttl=128 time=3.41 ms
- 64 bytes from 10.92.237.128: icmp_seq=3 ttl=128 time=2.55 ms
- 64 bytes from 10.92.237.128: icmp_seq=4 ttl=128 time=2.19 ms
- 64 bytes from 10.92.237.128: icmp_seq=5 ttl=128 time=2.29 ms
- 64 bytes from 10.92.237.128: icmp_seq=6 ttl=128 time=2.74 ms
- ^C
- --- esp8266.local ping statistics ---
- 6 packets transmitted, 6 received, 0% packet loss, time 5007ms
- rtt min/avg/max/mdev = 2.190/3.148/5.687/1.202 ms
- ** ping will automatically find the IP address of the ESP for you.

18.2.2.2 Discovery

Physical Web

- Enables you to see a list of URLs being broadcast by objects in the environment around you with a Bluetooth Low Energy (BLE) beacon.
- Limitations: you can only send secure web pages (HTTPS), and the URL is limited to 17 characters because of Bluetooth packet size limitations.
- Steps:
  - 1. Get beacons. On supported Android devices, you can use Beacon Toy to transform your phone into an Eddystone beacon. Otherwise, choose from a variety of beacon manufacturers.
  - 2. Configure beacons. You’ll have to select which URLs you’d like to broadcast (browsers such as Chrome and Nearby Notifications only support HTTPS) and how far and often you want your beacons to broadcast.
  - 3. Deploy. Place your beacons in a physical space. Anyone who passes by with a Physical Web-compatible service will see your URL.

18.2.2.2 Discovery

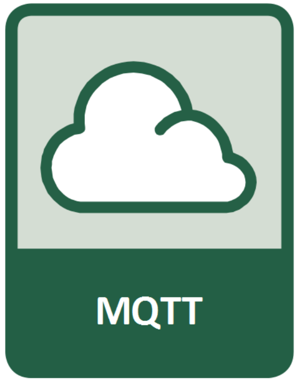
Physical Web from a Raspberry Pi/Linux Machine

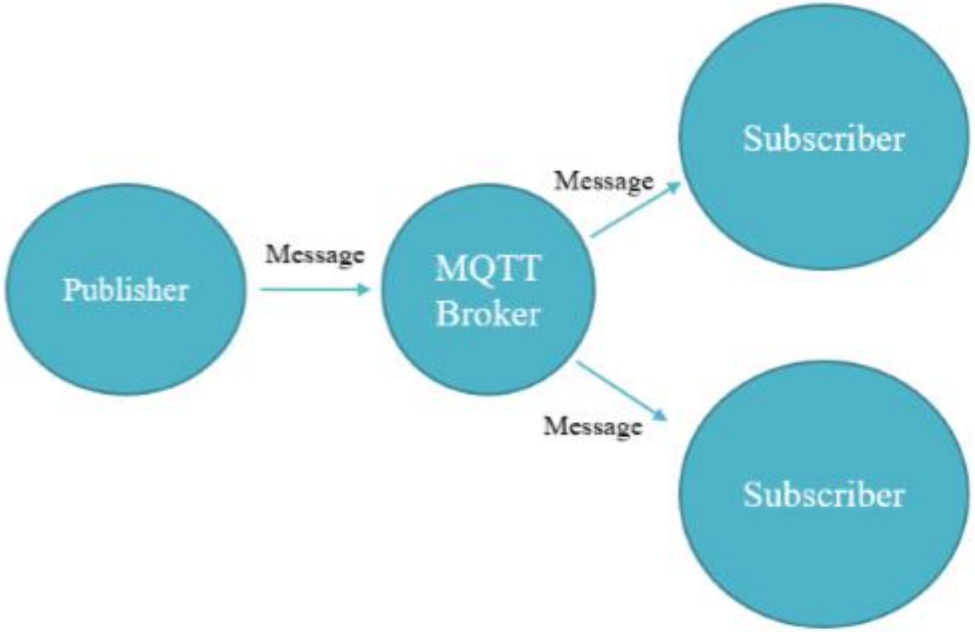
- #Make sure you have python3, pip, and bluez installed. This can be achieved by the following command
- sudo apt-get install bluez python3 python3-pip
- #Then, install the PyBeacon python package:
- sudo pip install PyBeacon
- #You can now start broadcasting your Physical Web URL by running this command:
- sudo PyBeacon -u https://medium.com/@urish
- #You can stop broadcasting the URL at any point by running:
- sudo PyBeacon -t

18.2.2.3 Data Protocols

Message Queuing Telemetry Transport (MQTT)

- Enables a publish/subscribe messaging model in an extremely lightweight way.
- It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium.
- It helps minimize the resource requirements for IoT device, and can handle unreliable networks.
- It runs on large networks of small devices that need to be monitored from a back-end server.





18.2.2.3 Data Protocols

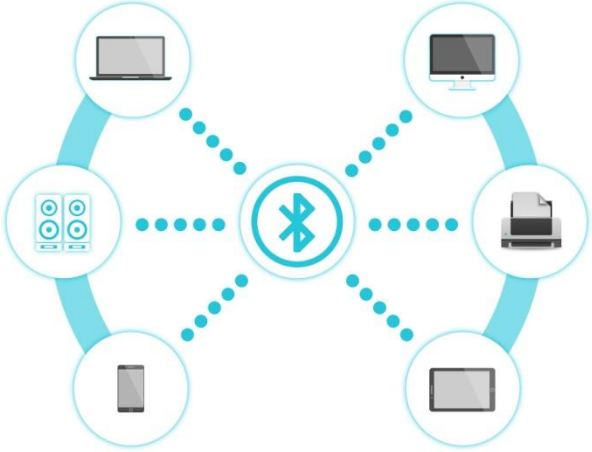
Websocket

- Developed as part of the HTML5 initiative as WebSocket JavaScript interface.
- A full-duplex single socket connection over which messages can be sent between client and server.
- The WebSocket standard simplifies much of the complexity around bi-directional web communication and connection management.
- Disadvantage: often too data heavy for IoT applications

18.2.2.4 Communication/Transport Layer

Bluetooth

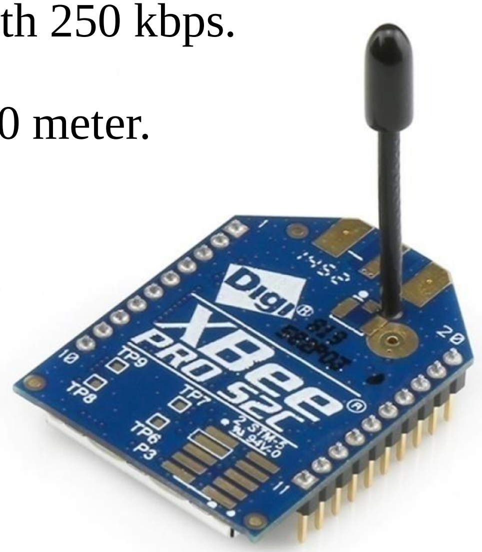
- Developed by Ericson Mobile Communication in 1994.
- Works in the 2.4 GHz ISM band and uses frequency hopping.
- With a data rate up to 3 Mbps and maximum range of 100m.
- Each application type using Bluetooth has its own profile.
- This Piconet is capable of 2 - 8 devices at a time for data sharing, and that data may be text, picture, video and sound.



18.2.2.4 Communication/Transport Layer

ZigBee

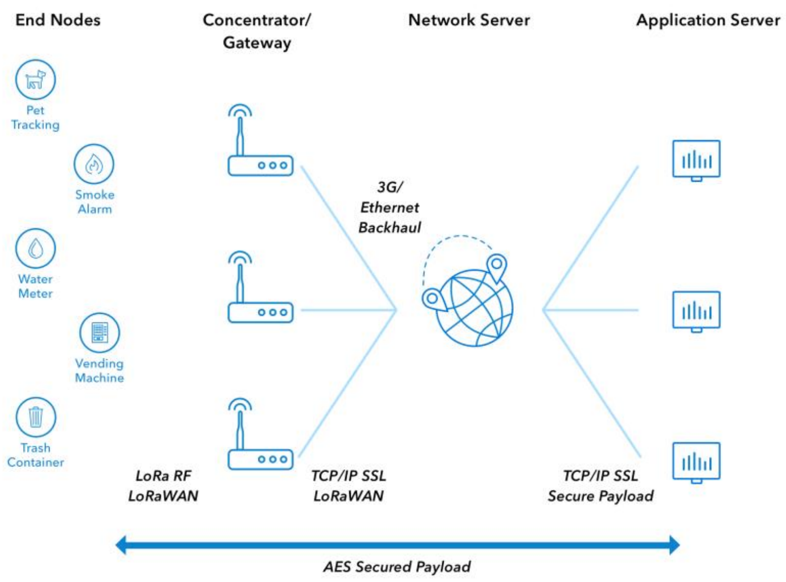
- Developed for enhancing the features of wireless sensor networks
- ZigBee technology was created by the ZigBee Alliance, founded in the year 2001.
- Characteristics of ZigBee are low cost, low data rate, relatively short transmission range, scalability, reliability, flexible protocol design
- It uses the 802.15.4 standard and operates in the 2.4 GHz frequency range with 250 kbps.
- The maximum number of nodes in the network is 1024 with a range up to 200 meter.
- ZigBee can use 128 bit AES encryption.



18.2.2.4 Communication/Transport Layer

Low Range Wide Area Network (LoRaWAN)

- Network protocol intended for wireless battery operated Things in regional, national or global network.
- It is designed to allow low-powered devices to communicate with Internet-connected applications over long range wireless connections.
- Suitable uses- low range, low power, low bandwidth, coverage everywhere, secure.
- Unsuitable uses- realtime data, phone calls, controlling lights in the house, sending photos etc.



18.2.2.5 Semantic

- **Internet of Things Data Base (IoTDB):**
  - IoTDB is a specialized database management system for time series data generated by a network of IoT devices with low computational power.
  - Data in IoTDB is stored in TsFile, a file format designed for accessing, compressing, and storing time series data.
  - IoTDB is a project started in 2017 by Prof. Jianmin Wang’s group in the School of Software of Tsinghua University and China’s National Engineering Laboratory for Big Data Software.
- To use IoTDB, you need to have:

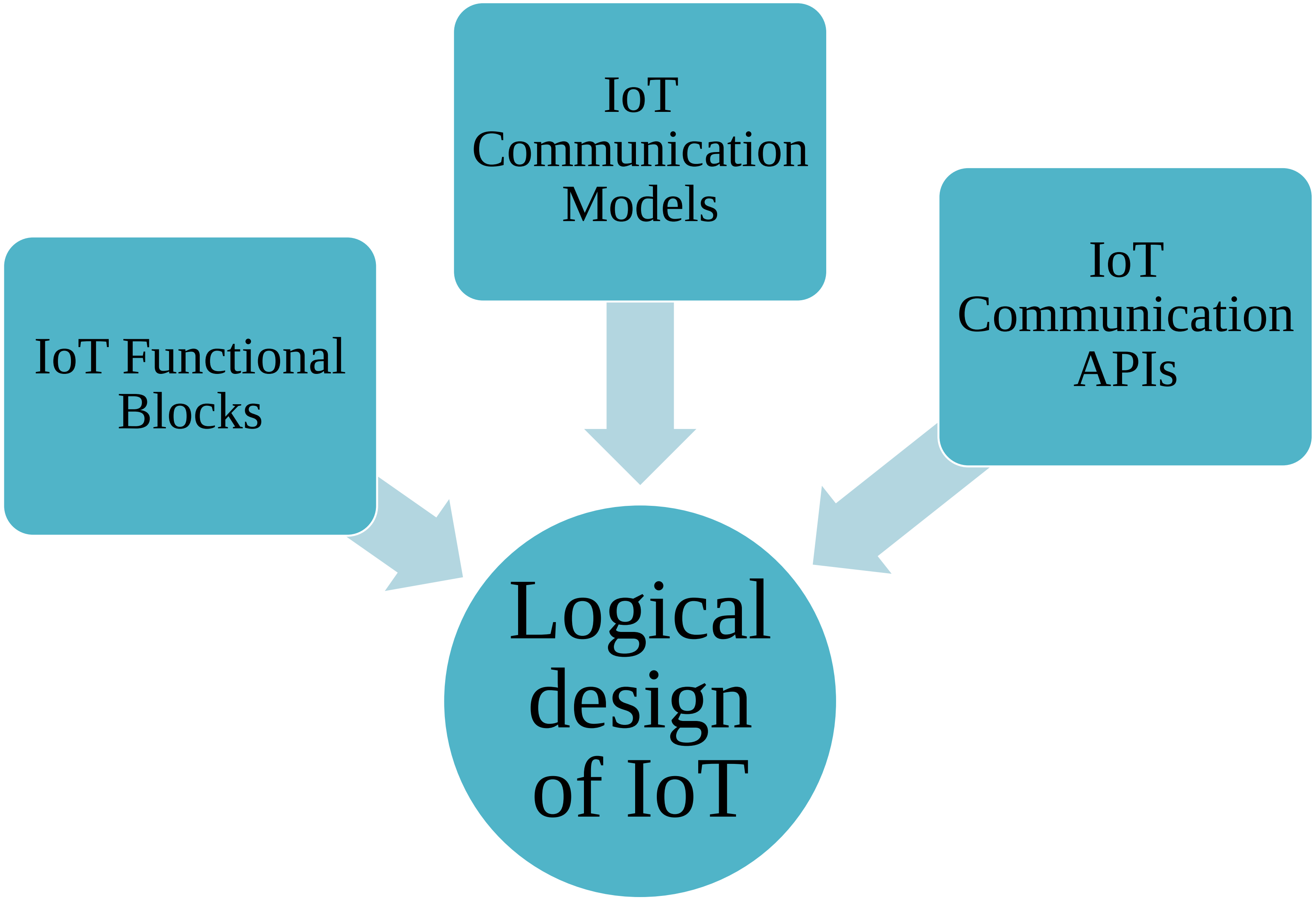
1. Java >= 1.8 (1.8, 11, and 13 are verified. Please make sure the environment path has been set accordingly).
2. Maven >= 3.1 (If you want to compile and install IoTDB from source code).
3. Set the max open files num as 65535 to avoid "too many open files" error.

18.2.2.5 Semantic

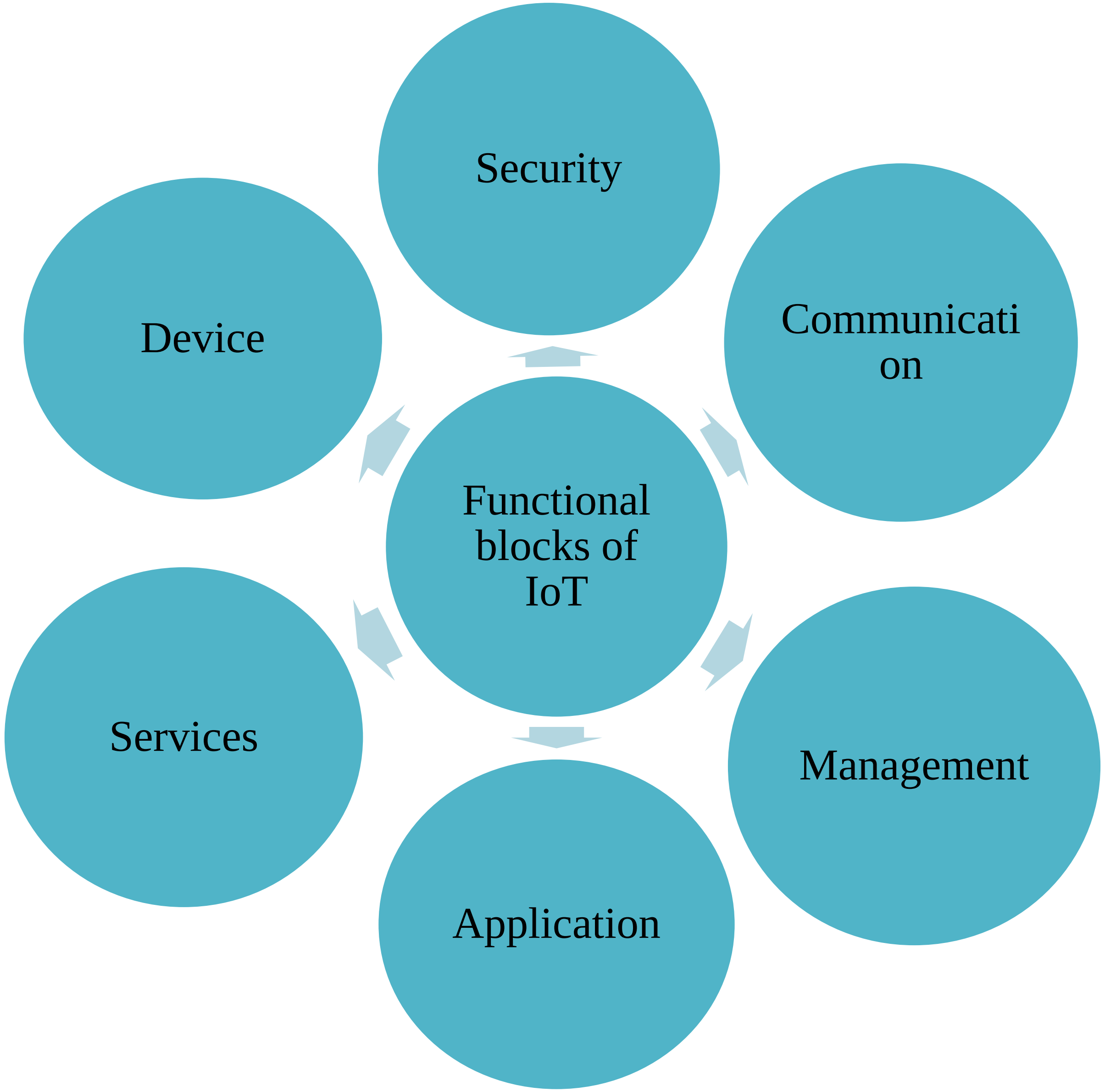
Sensor Model Language (SensorML)

- A set of standard models and XML schema as defined in the Sensor Web Enablement method by the Open Geospatial Consortium for describing sensor systems and processes.
- This includes sensors and actuators as well as computational processes applied pre- and post-measurement.
- The main objective is to enable interoperability.

18.3 Logical design of IoT

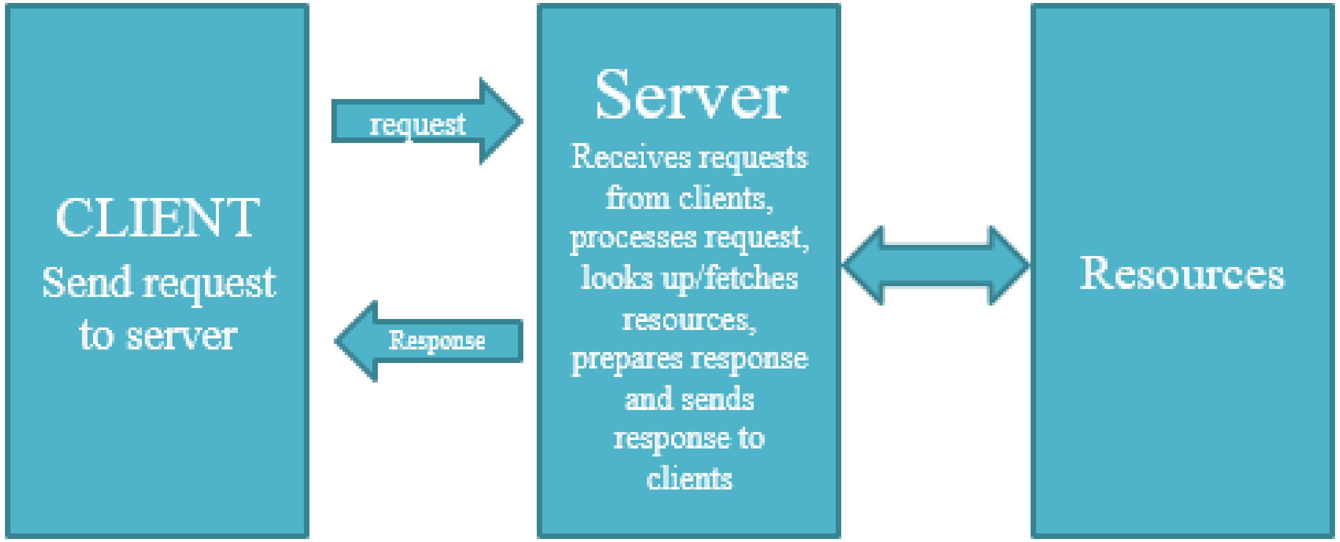


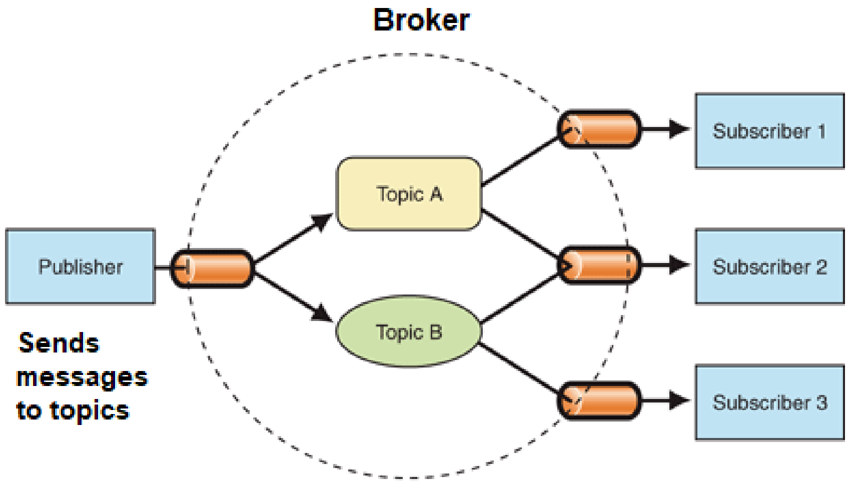
18.3.1 IoT Functional Blocks



18.3.2 IoT Communication models

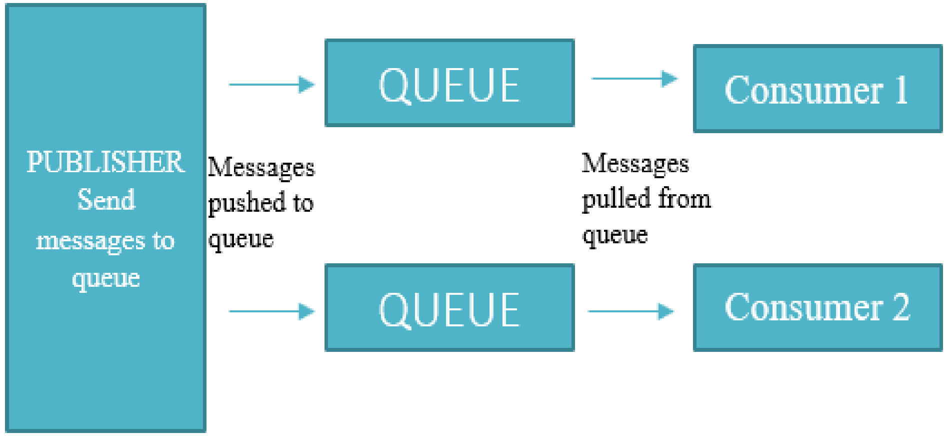
Request Response Model



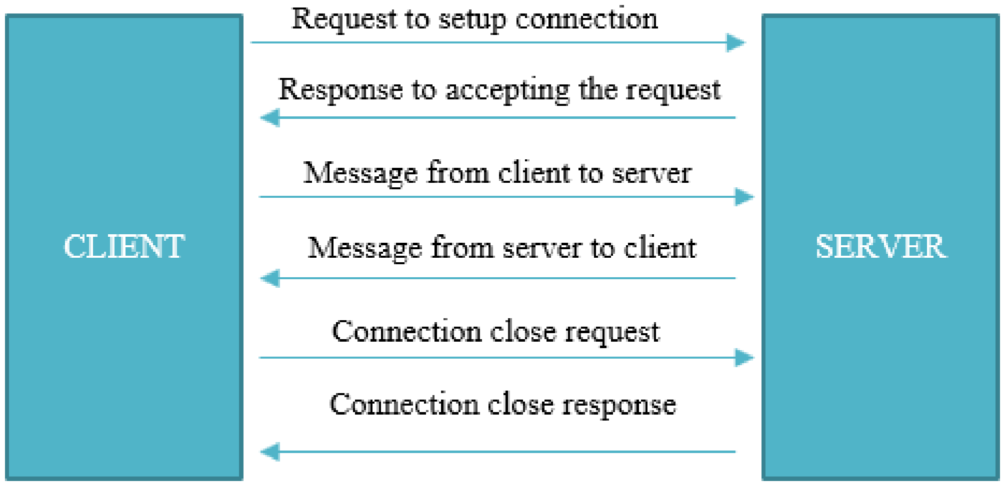


18.3.2 IoT Communication models

Push-Pull Model

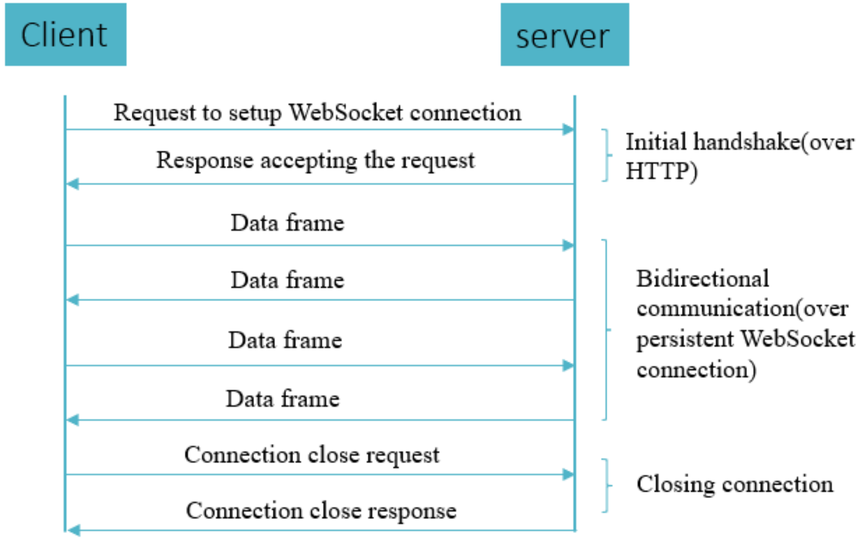


Exclusive pair Model



18.3.3 IoT communication APIs

WebSocket-based Communication APIs: This API follows the Exclusive Pair Model. It allows bi-directional full duplex communication between clients and servers.

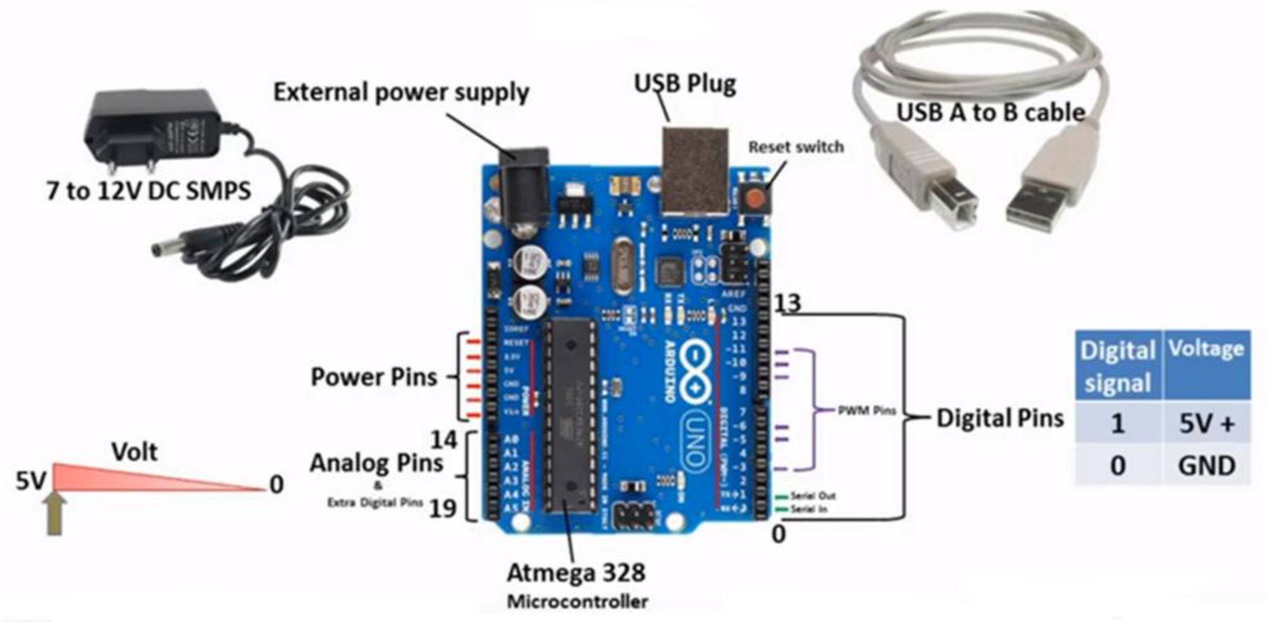


REST-based Communication APIs: Representational state transfer (REST) are principles used in the web design services and web APIs. It follows the request-Response Model

18.4 Arduino

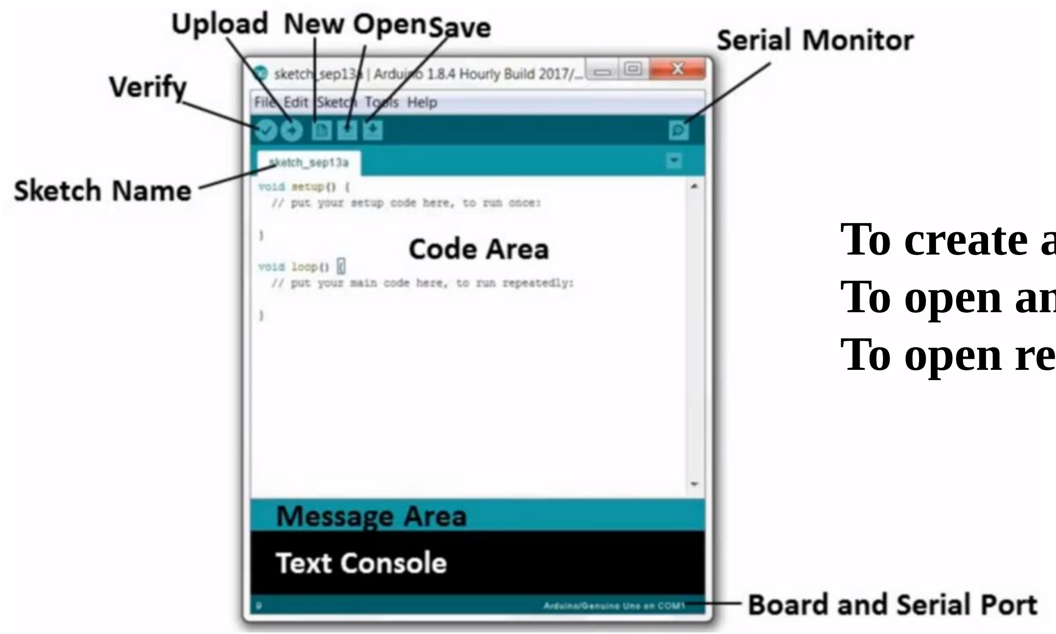
Arduino is an open-source microcontroller development board.

- Developed by IDII(Interaction Design Institute Ivrea), Northern Italy, in 2005
- Consists of physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE.
- The Uno is one of the more popular boards in the Arduino family and a great choice for beginners



18.4.1 Arduino IDE

- Arduino IDE is an open source software used to program the Arduino Board.
- Arduino IDE freely downloadable from the Arduino official website and installed into PC.
- The software IDE will help you change the behavior of the microcontroller according to our needs.



To create a new sketch: File -> New
To open an existing sketch: File -> open
To open ready to use sketch: File -> Examples -> select any program

18.4.2 Developing on the Arduino

Programming language: Example using C++

- **Blinking a single LED**
- // Pin 13 has an LED connected on most Arduino boards.
- // give it a name:
- int led = 13;
- // the setup routine runs once when you press reset:
- void setup() {
- // initialize the digital pin as an output.

pinMode(led, OUTPUT);
}
// the loop routine runs over and over again forever:
void loop() {
digitalWrite(led, HIGH); // turn the LED on
delay(1000); // wait for a second
digitalWrite(led, LOW); // turn the LED off
delay(1000); // wait for a second
}

18.4.3 Case Study

The Good Night Lamp

Internet-connected table/bedside lamp.

Botanicalls

Monitoring kits placed in plant pots.

BakerTweet

Twitter designed for use in a bakery.

18.5 Raspberry Pi

- Series of credit-card-sized single-board computers dedicated for educational purposes
- Power Modes:
  - Run Mode
  - Standby Mode
  - Dormant Mode
  - Shutdown Mode
- One of the great things about the Raspberry Pi is that it is very flexible and there's no single way to use it. For example, it can be used for general purpose computing, learning to program or integrate it with electronics projects.

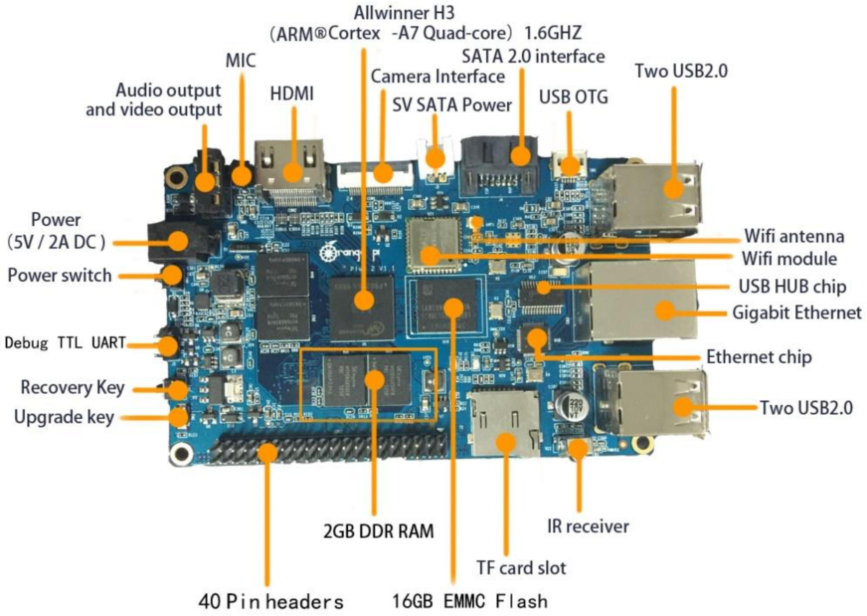
Eben Upton, Founder, Raspberry Pi

18.5 Raspberry Pi

- Why Raspberry Pi?
  - Raspbian has a desktop environment similar to Windows and Mac called Lightweight X11 Desktop Environment (LXDE)
  - It comes pre-installed with software useful for writing codes.
  - The operating system has been tailored to run on the Raspberry Pi.
  - There is wide spread community support for the operating system.

18.5.1 Models of Raspberry Pi

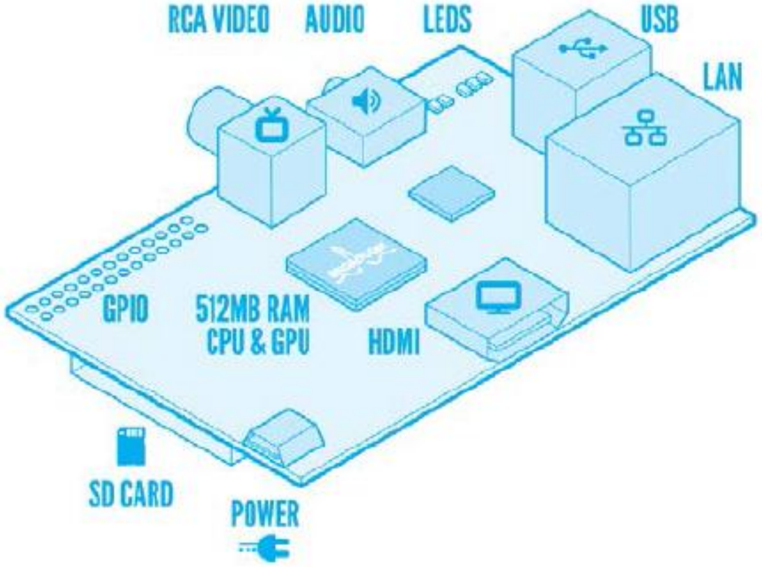
Key features	Raspberry pi 3 model B	Raspberry pi 2 model B	Raspberry pi zero
RAM	1GB SDRAM	1GB SDRAM	512MB SDRAM
CPU	Quad cortex A53@1.2GHz	Quad cortex A53@900MHz	ARM@1GHz
GPU	400 MHz video core IV	250 MHz video core IV	250 MHz video core IV
Ethernet	10/100	10/100	None
Wireless	802.11/Bluetooth 4.0	None	None
Video output	HDMI/Composite	HDMI/Composite	HDMI/Composite
GPIO	40	40	40



18.5.2 Operating system of Raspberry Pi

Operating system on Linux distribution:

- **Raspbian:** Released by the Raspbian Pi Foundation, Raspbian is a distro based on Debian. This is the default "official" distribution and is certainly a good choice for general work with a Pi.
- **Occidentalis:** This is Adafruit's customized Raspbian. Unlike Raspbian, the distribution assumes that you will use it "headless"—not connected to keyboard and monitor—so you can connect to it remotely by default. (Raspbian requires a brief configuration stage first.)



For IoT work, it is recommended to use Adafruit distro.

18.5.3 Raspberry Pi OS Installation

1. Download preferable distro from www.raspberrypi.org
2. Extract the zip file
3. Unpack it on the SD card(Note that some SD cards don’t work well with the Pi, use class 10 cards)
4. Place the SD card in the Raspberry Pi SD card slot
5. At this point, the Pi may boot up if you have enough power to it from the USB
6. You can communicate with the Pi just as you’d communicate with any computer
7. Use following command to log in to the Pi

   > For Linux and Mac $ ssh root@raspberrypi.local

   > For Windows (www.chiark.greenend.org.uk/~sgtatham/putty)
8. After you connect to the device, you can develop a software application for it as easily as you can for any Linux computer.

18.5.3 Library Installation

Install Adafruit DHT sensor library in Raspberry pi

Get the clone from GIT

git clone https://github.com/adafruit/Adafruit_Python_DHT.git

Go to folder Adafruit_Python_DHT

cd Adafruit_Python_DHT

Install the library

sudo python setup.py.install

18.5.3 Raspberry Pi OS Installation

Programming language: Example using Python

Blinking lights example:

```
import RPi.GPIO as GPIO
from time import sleep
GPIO.setmode(GPIO.BOARD) # set the numbering scheme to be the
# same as on the board
GPIO.setup(8, GPIO.OUT) # set the GPIO pin 8 to output mode
led = False
GPIO.output(8, led) # initiate the LED to off
while 1:
GPIO.output(8, led)
led = not led # toggle the LED status on/off for the next
# iteration
sleep(10) # sleep for one second
```

18.5.4 Specification

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- 1GB, 2GB or 4GB LPDDR4-3200 SDRAM (depending on model)
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- Gigabit Ethernet
- 2 USB 3.0 ports; 2 USB 2.0 ports.
- Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
- 2 × micro-HDMI ports (up to 4kp60 supported)
- 2-lane MIPI DSI display port
- 2-lane MIPI CSI camera port
- 4-pole stereo audio and composite video port
- H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
- OpenGL ES 3.0 graphics
- Micro-SD card slot for loading operating system and data storage
- 5V DC via USB-C connector (minimum 3A*)
- 5V DC via GPIO header (minimum 3A*)
- Power over Ethernet (PoE) enabled (requires separate PoE HAT)
- Operating temperature: 0 – 50 degrees C ambient
- Consumes about 260mA of current at 5.0 V.

* A good quality 2.5A power supply can be used if downstream USB peripherals consume less than 500mA in total.

18.5.5 Advantages and disadvantages

Advantages of Raspberry Pi	Disadvantages of Raspberry Pi
very large working memory (RAM) and expandable memory to store the data (SD storage)	does not have a real-time clock (RTC) with a backup battery.
works on processor which supports a large set of instructions	always boots from an SD card.
operates at speeds from 700 MHz to 1000 MHz.	most Linux distributions are still a bit picky about their hardware, so it should be first checked whether flavor of Linux supports particular device
has support for USB 2.0	doesn't have built-in an Analog to Digital converter
Built-in with WiFi and Bluetooth Adapters	necessary to provide a power supply that can provide enough current to power the device plus any connected peripherals
possible to form an expandable system with various electronic components (sensors and electronic circuits)	external storage devices can be used but can't be used to boot the Raspberry Pi

18.5.6 Arduino vs Raspberry Pi

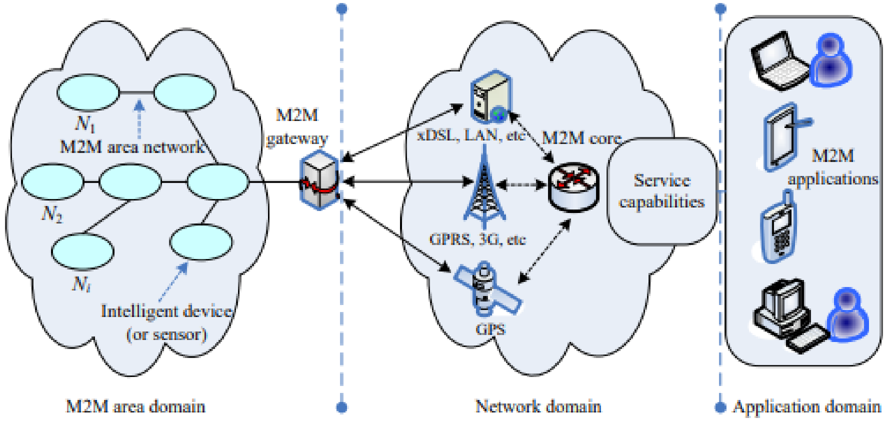
	Arduino Due	Raspberry Pi Model B
CPU Speed	84 MHz	700 MHz ARM11
GPU	None	Broadcom Dual-Core VideoCore IV Media Co-Processor
RAM	96KB	512MB
Storage	512KB	SD card (4GB +)
OS	Bootloader	Various Linus distributions, other operating systems available
Connections	54 GPIO pins 12 PWM outputs 4 UARTs SPI bus I²C bus USB 16U2 + native host 12 analogue inputs (ADC) 2 analogue outputs (DAC)	8 GPIO pins 1 PWM output 1 UART SPI bus with two chip selects I²C bus 2 USB host sockets Ethernet HDMI out Component video and audio out

18.6 Advanced approaches of IoT

18.6.1 M2M: Machine to Machine

- Communications between computers, embedded processors, smart sensors, actuators, and mobile devices without or with limited human intervention.

- M2M architecture domains



- Application Domains

Domain	Applications
Security	Surveillance applications, alarms, object/people tracking
Transportation	Fleet management, emission control, toll payment, road safety related to activities in TC ITS
Heathcare	Related to e-health and personal security
Utilities	Measurement, provisioning, and billing of utilities, such as oil, water, electricity, heat, and others
Manufacturing	Production chain monitoring and automation
Supply and Provisioning	Freight supply, distribution monitoring, and vending machines
Facility Management	Home, building, and campus automation

18.6.1 IoT vs M2M

	IoT	M2M
Communication	Uses both wired and wireless communication	Uses wired or wireless communication.
Internet connection	Requires an active internet connection	Does not necessarily require an internet connection
Relies on	Internet connectivity and cloud	Cellular or the wired network
Scalability	Cloud based makes IoT more scalable	Less scalable

18.6.2 Wireless Sensor Networks (WSN) for IoT

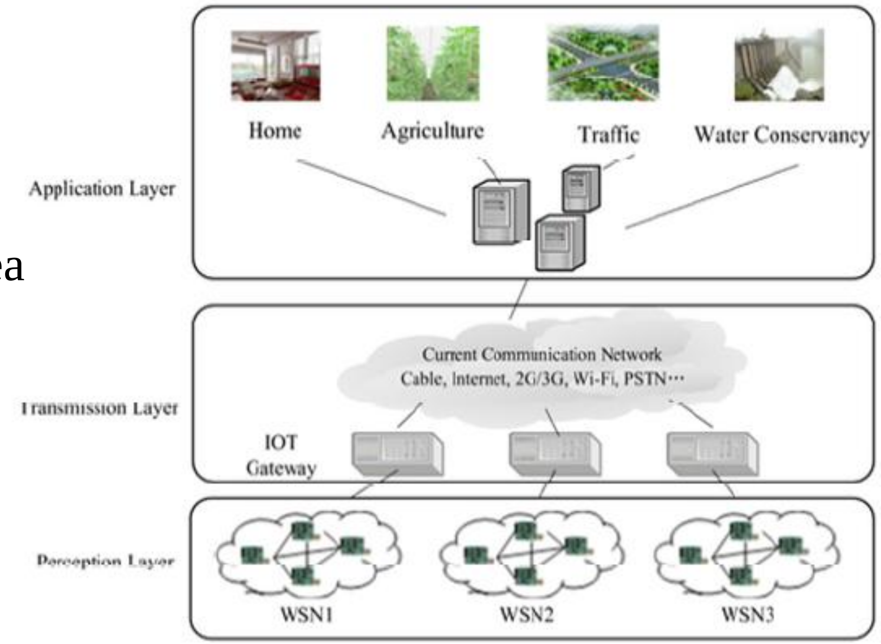
- Wireless Sensor Networks (WSNs) are connecting things to the Internet through a gateway that interfaces the WSN to the Internet.
- To enable the integration of WNS in the IoT, there are two key points that should be added to the relevant protocols: First, the IPv6/IPv4 over Low power Wireless Personal Area Networks (6LoWPAN) protocol should be implemented and deployed in Wireless Sensor Networks (WSNs); Second, Machine to Machine communications (M2M) protocols need to be standardized.
- WSN Key Characteristics
    - Sensor nodes
    - Wireless connectivity technology such as Zigbee, Bluetooth or WiFi
    - A Network Manager
    - A Cloud Platform

18.6.2 WSN-based IoT Application Architecture

1. Perception layer: collect and process the data from the physical world, which consists of two parts: the sensor device and wireless sensor networks

2. Transmission layer: the system aims to transfer data in a large area or long distance. The data collected from perception layer can be transferred successfully to remote destination.

3. Application layer: Data processing and services providing are two major purposes of the application layer.

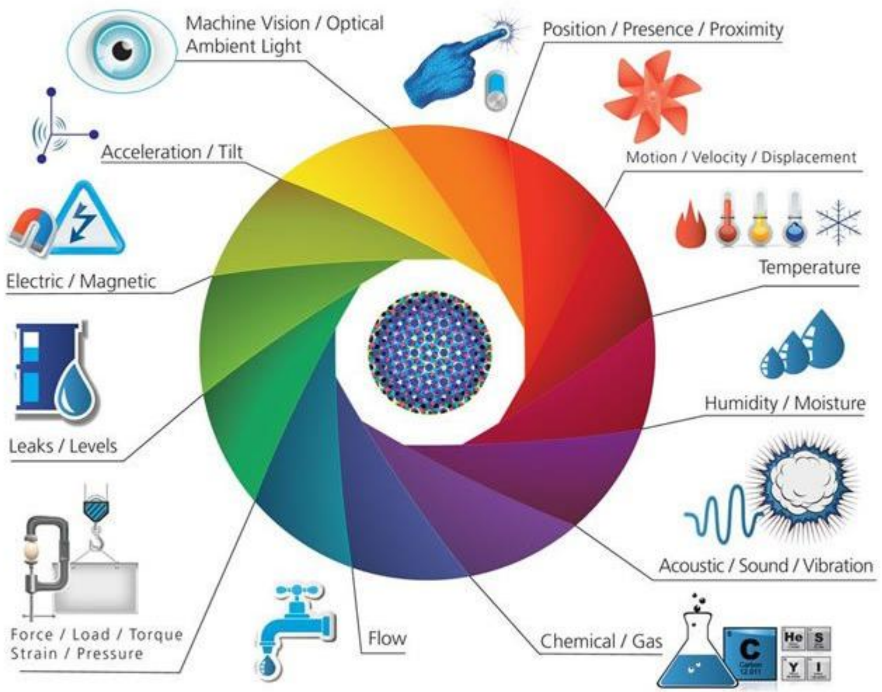


Typical IoT application architecture

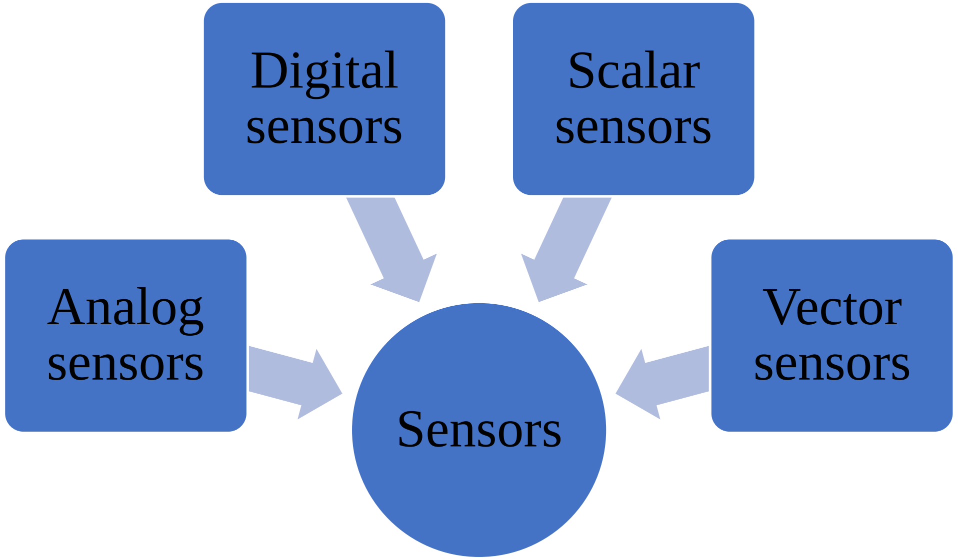
18.6.2 WSN Advantages and disadvantages

Advantages of WSN	Disadvantages of WSN
Avoids a lot of wiring	Lower speed compared to wired network
Can accommodate new devices at any time	More complex to configure
Flexible and can go through physical partitions	Easily distracted by various elements like Bluetooth
Can be accessed through a centralized monitor	High installation cost

18.6.3 IoT Sensors

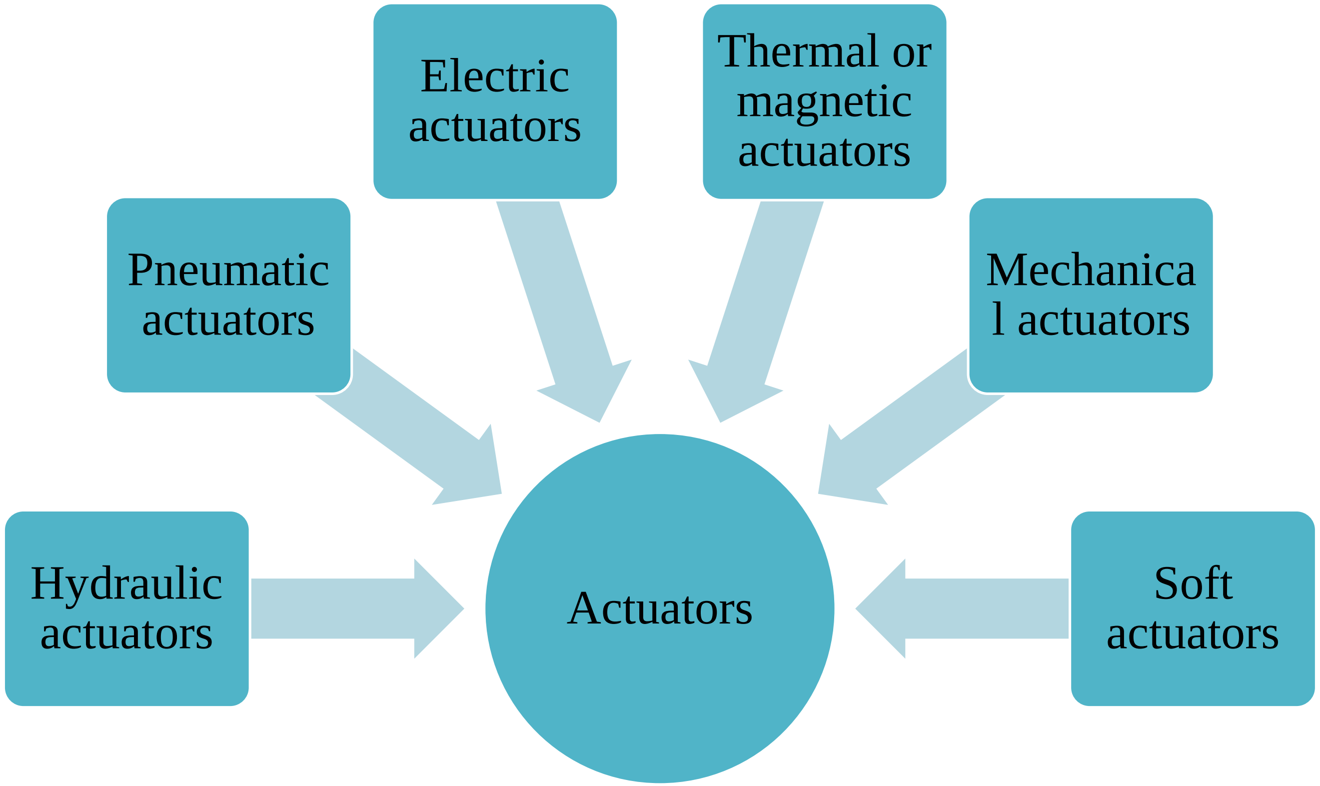


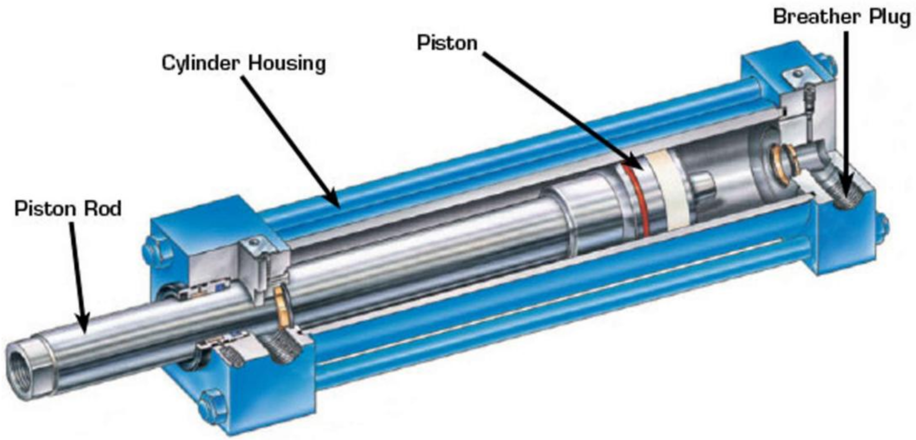
- A device that detects and responds to some type of input from the physical environment
- The main purpose of sensors is to collect data from the surrounding environment.
- Whereas sensors sense and send, actuators act and activate. The actuator gets a signal and sets in motion what it needs to set in motion in order to act upon/within an environment.



18.6.4 IoT Actuators

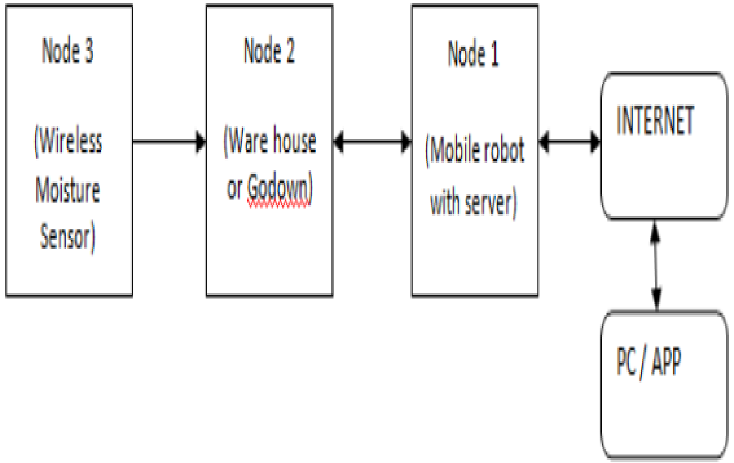
It is a component of the machine or system that moves or controls the mechanism or the system.

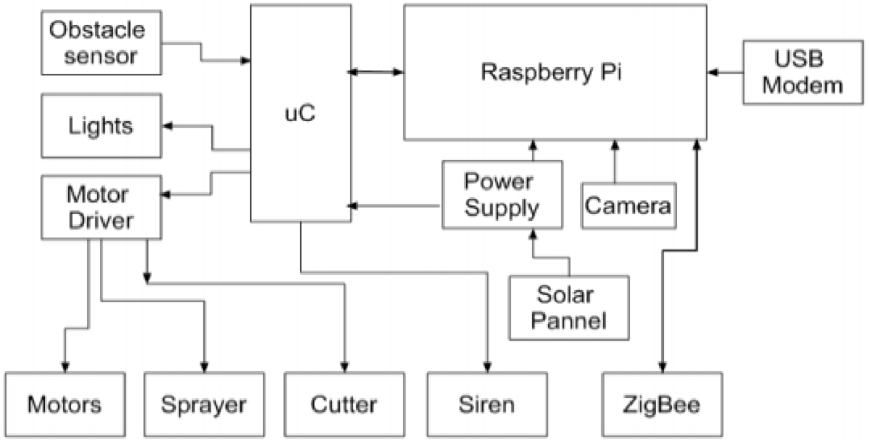




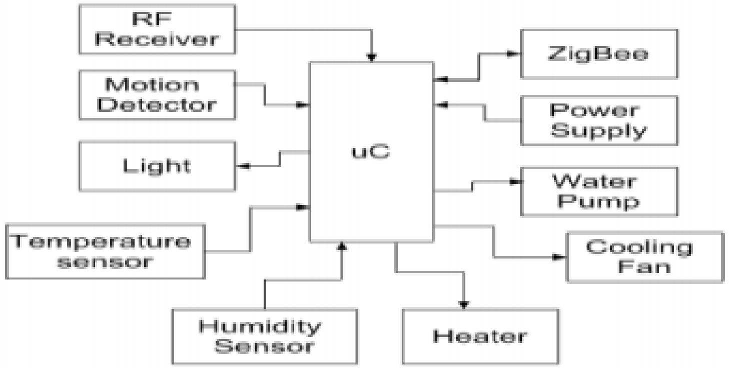
IoT based Smart Agriculture

The sensors and microcontrollers of 3 Nodes are interfaced with Raspberry Pi and wireless communication is achieved between various nodes

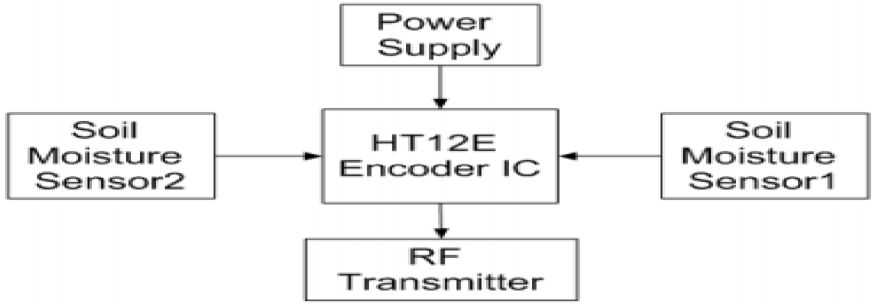




Node 1 : Mobile Robot



Node 2: Godown/Warehouse



Node 3 : Wireless Moisture Sensor Node

18.6.6 IoT Physical Servers and Cloud Offerings

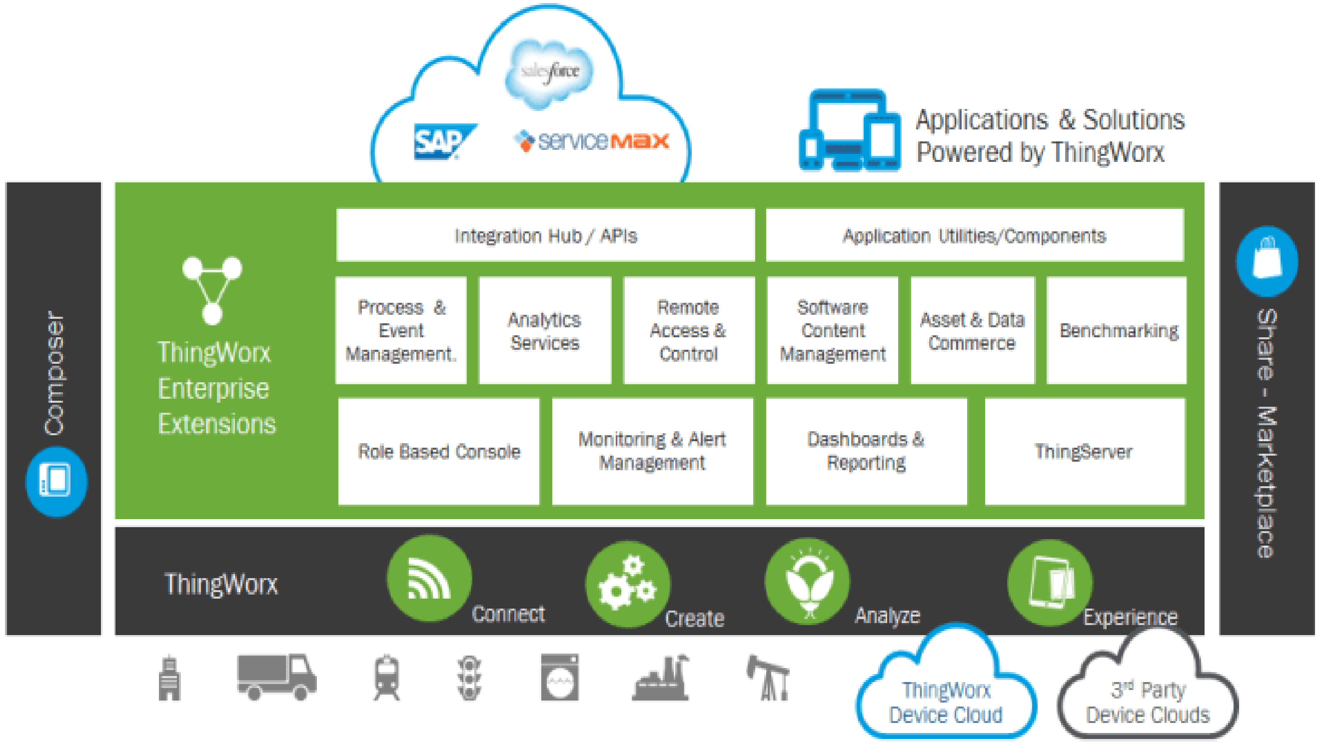
Cloud: The cloud is a huge, interconnected network of powerful servers that performs services for businesses and for people.

Top IoT clouds:

1. **Thingworx 8 IoT Platform**

Features

- Easy connectivity with electronic devices, like sensors and RFIDs
- You can work remotely once you are done with the setup
- Pre-built widgets for the dashboard
- Remove Complexity of the project
- Integrated machine learning

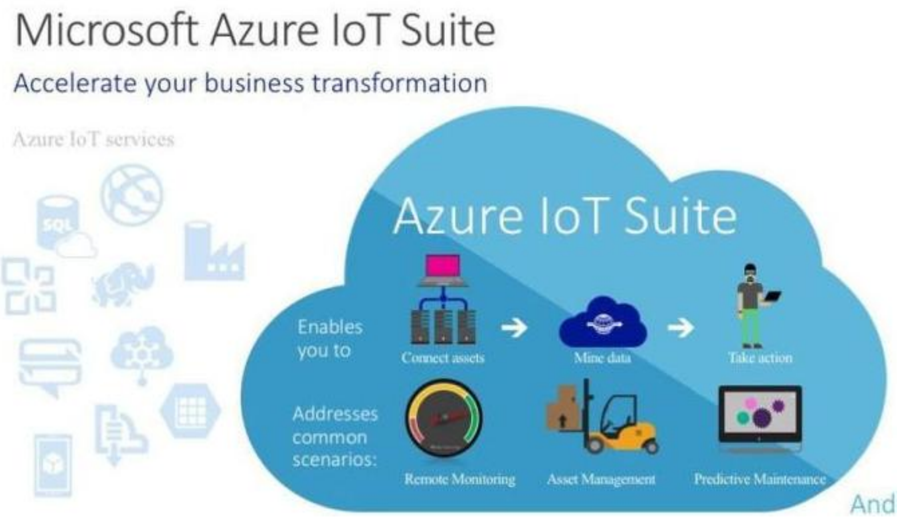


18.6.6 IoT Physical Servers and Cloud Offerings

2. **Microsoft Azure IoT Suite**

Features:

- Easy Device Registry.
- Rich Integration with SAP, Salesforce, Oracle, WebSphere, etc
- Dashboards and visualization
- Real-time streaming



3. **Google Cloud's IoT Platform**

Features:

- Provides huge storage
- Cuts cost for server maintenance
- Business through a fully protected, intelligent, and responsive IoT data
- Efficient and scalable
- Analyze big data

Google Cloud Platform

18.6.6 IoT Physical Servers and Cloud Offerings

4. IBM Watson IoT Platform

Features:

- Real-time data exchange
- Secure Communication
- Cognitive systems
- Recently added data sensor and weather data service

5. AWS IoT Platform

Features:

- Device management
- Secure gateway for devices
- Authentication and encryption
- Device shadow

18.6.7 Challenges of IoT

Development challenges

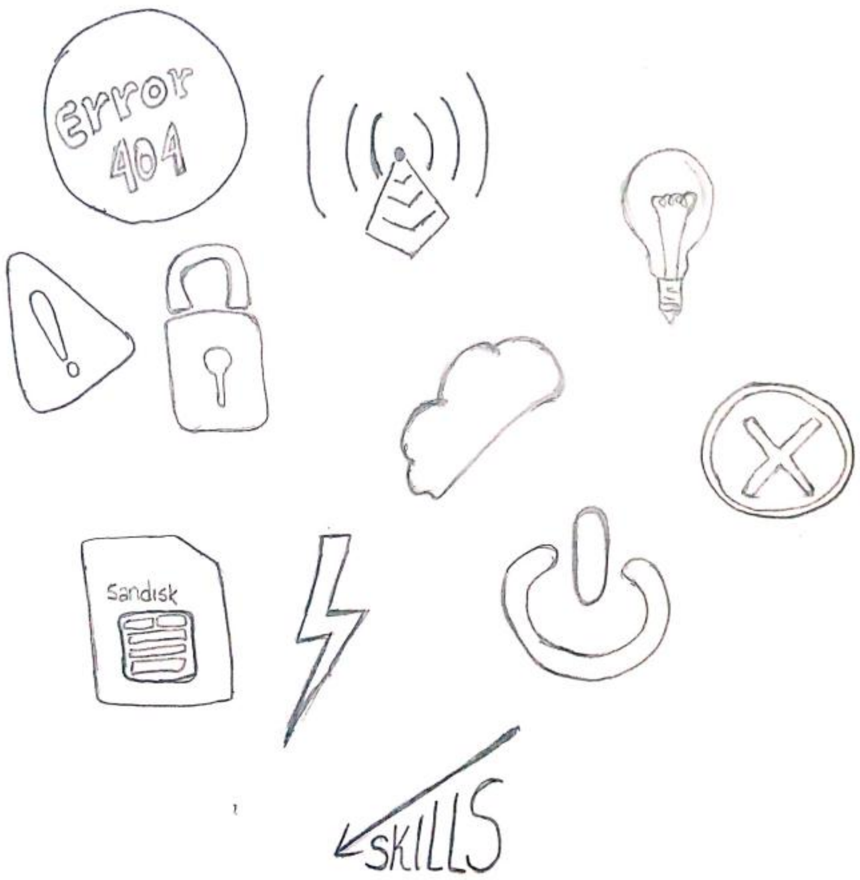
- Security and privacy
- Connectivity
- Cross-Platform compatibility(Hardware and devices)
- Data collection and processing
- Lack of skill set

Security challenges

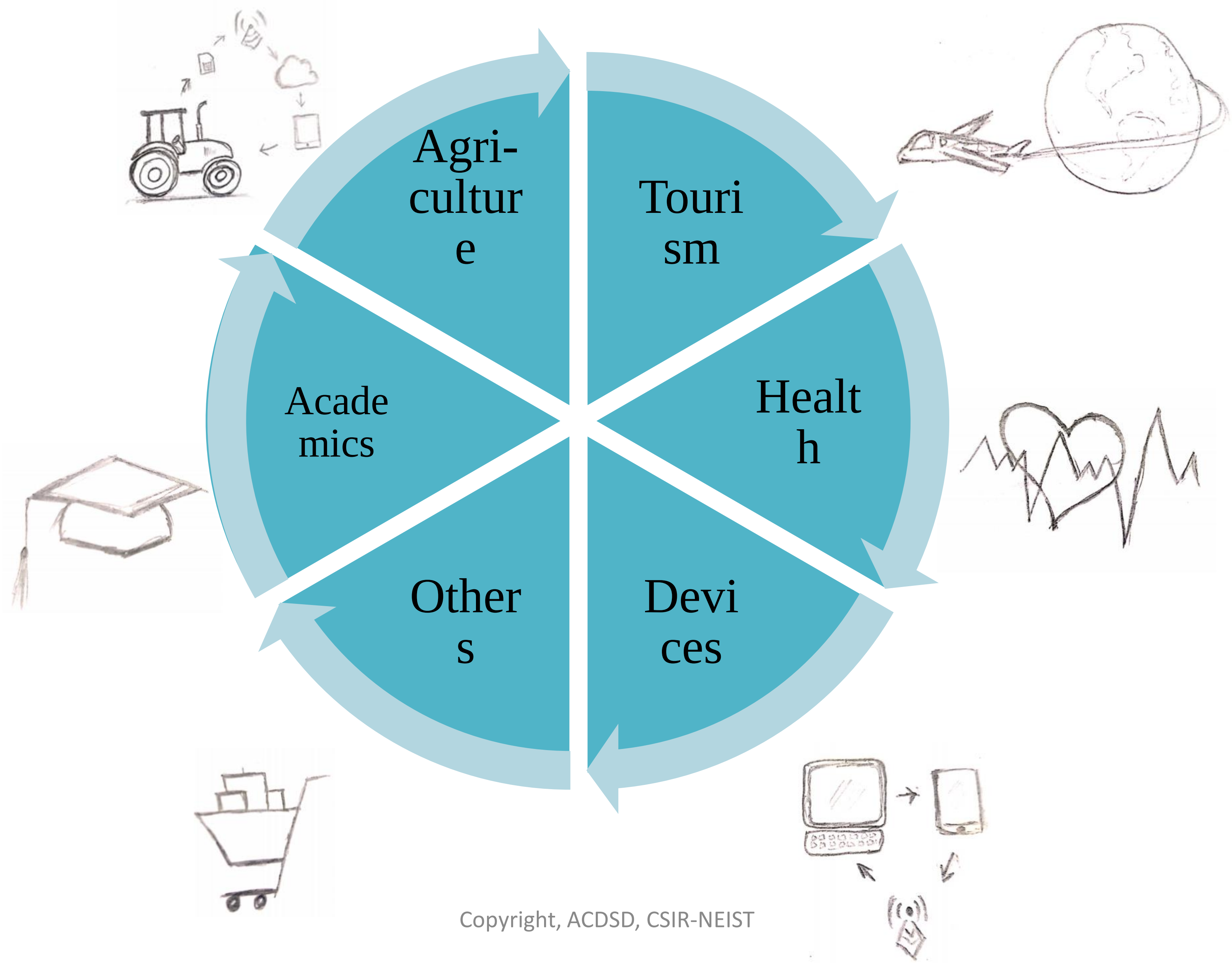
- Data exchange security
- Physical security
- Cloud Storage Security
- Privacy updates

Design challenges

- Stability of network
- Power management
- Network failover and memory management
- OS optimization and tuning
- Non-functional requirements



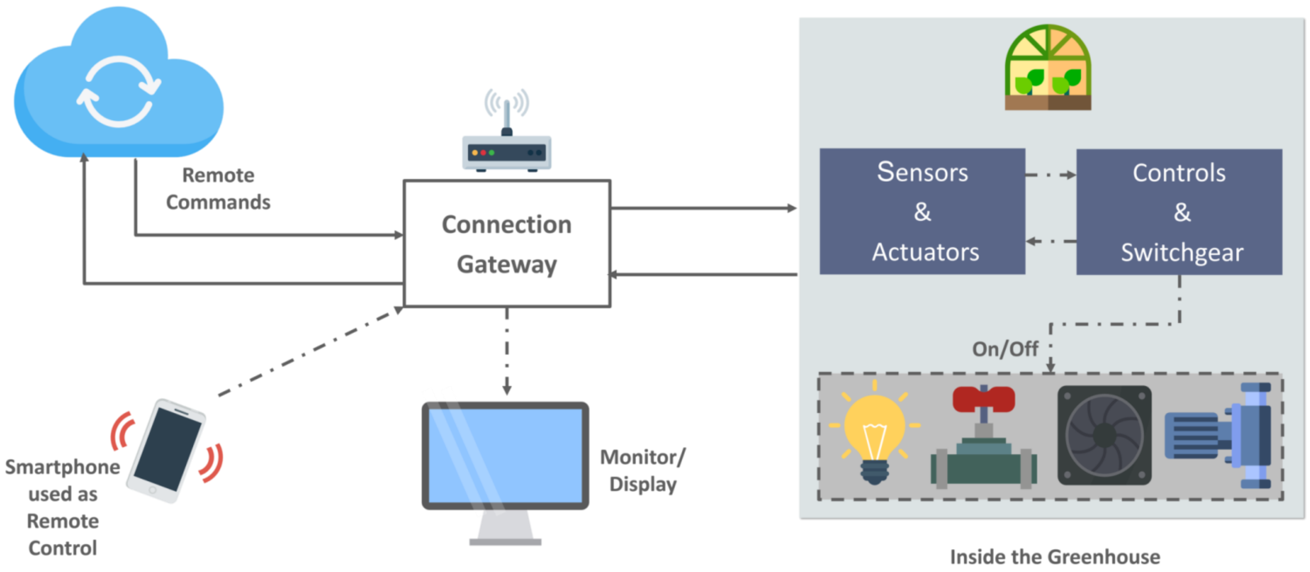
18.6.8 Domain specific applications of IoT



18.6.8 Domain specific applications of IoT

Application in agriculture

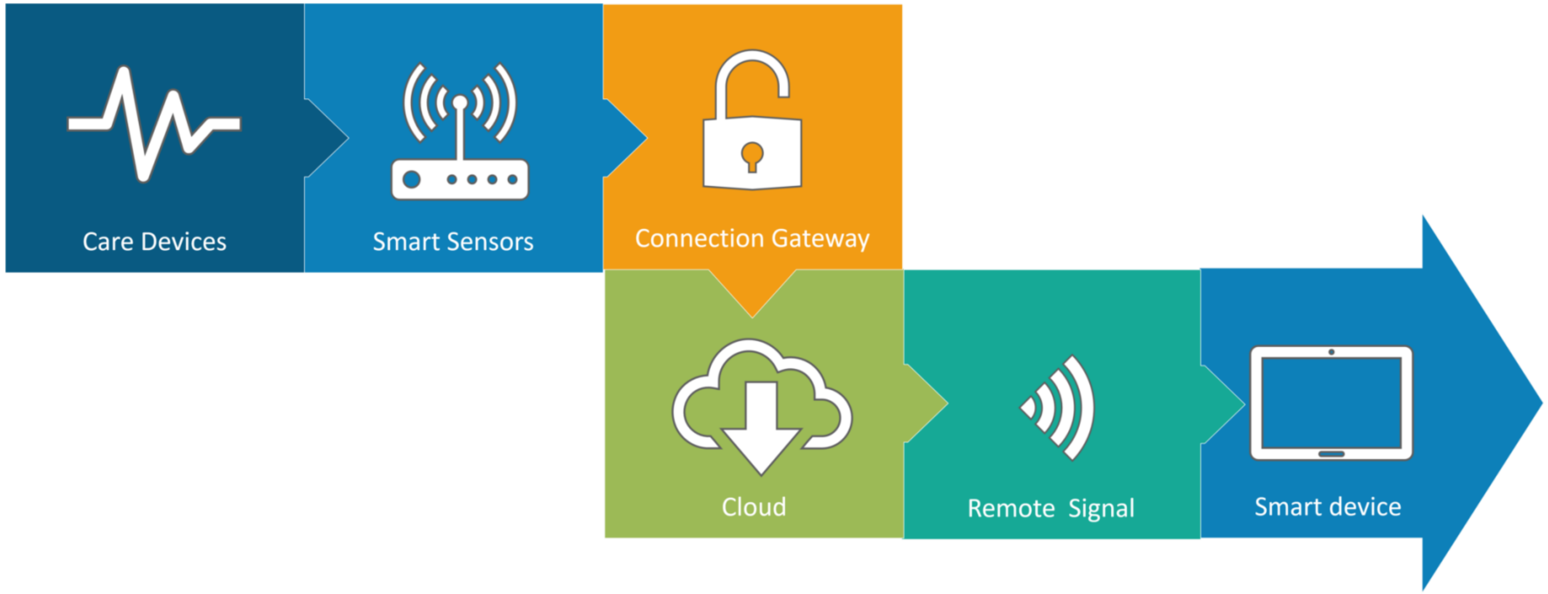
Smart Greenhouse: Greenhouses are embedded with sensors which measure different parameters according to the plant requirement and send it to the cloud. It then processes the data and applies a control action.



18.6.8 Domain specific applications of IoT

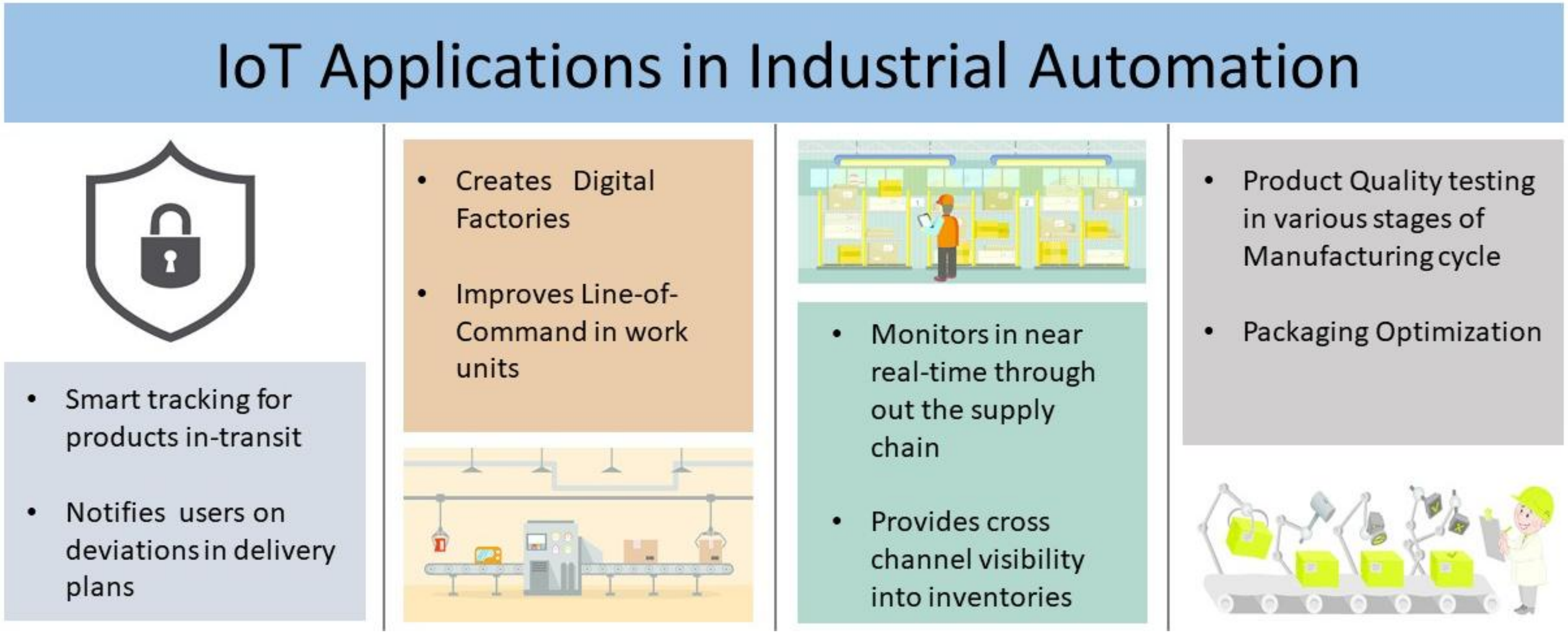
Application in Healthcare

- Healthcare domain are classified into tracking of objects, staff and patients, identification and authentication of people, automatic data collection and sensing.
- IoT applications can turn reactive medical-based systems into proactive wellness-based systems.



Application in industry

- Industrial Internet is the new buzz in the industrial sector, also termed as Industrial Internet of Things (IIoT)
- It is empowering industrial engineering with sensors, software and big data analytics to create brilliant machines.



18.6.8 Domain specific applications of IoT

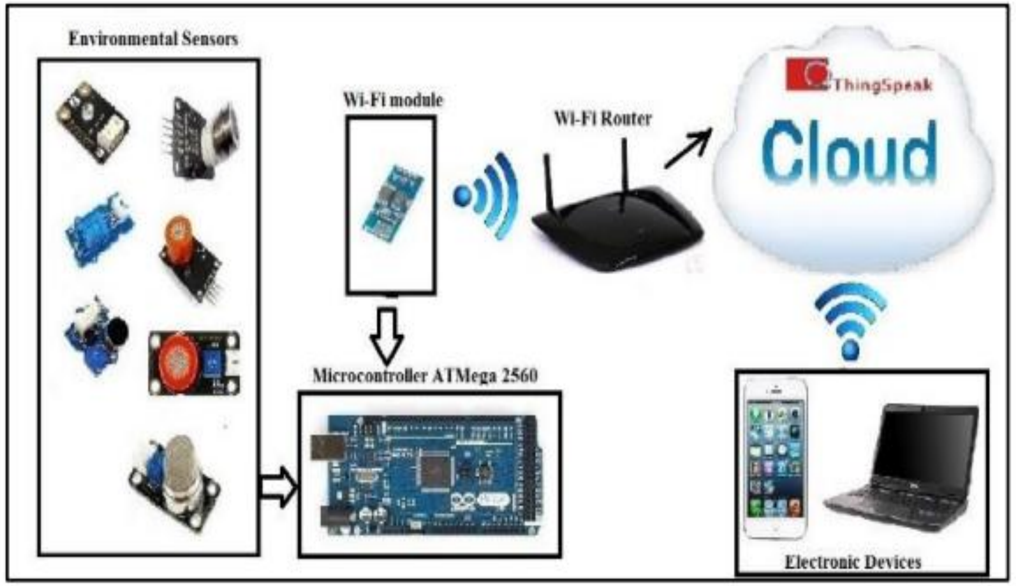
Application in academics

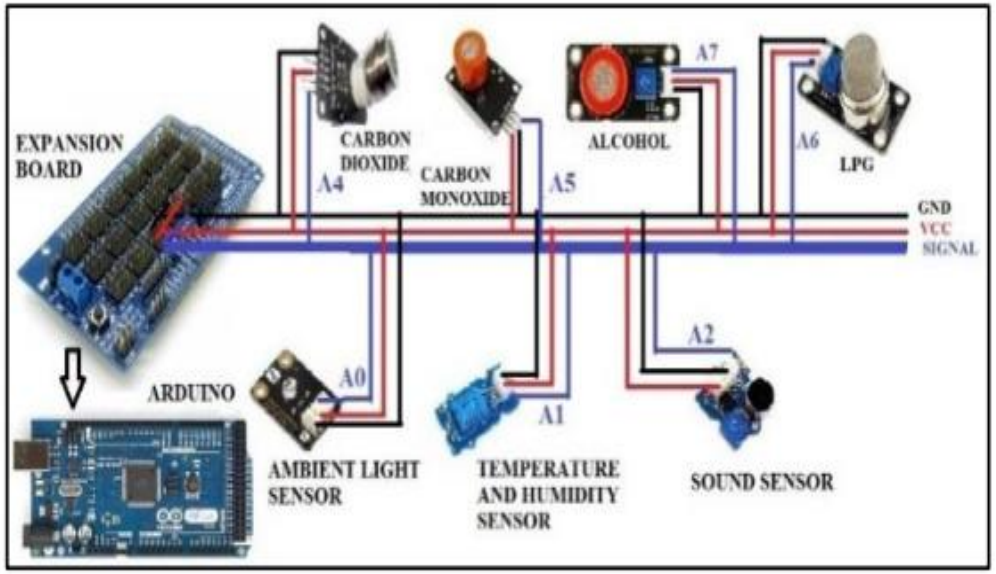
IoT in academics is very prominent. Following are the list of some applications in IoT

- Classroom technologies: Projectors, IP streamed audios, computer presentation integration
- IP connected laboratory
- Safety: IP video surveillance, fire alarm and life safety systems, security alarms etc

Application in environmental monitoring

Ambient environmental parameters, such as, CO, CO_2, alcohol, temperature, humidity, sound etc, are selected and connected to sensors. All environmental sensors act as input and connected to Arduino ATMega 2560.





18.6.8 Domain specific applications of IoT

IoT Applications in Biological and Chemical Sciences

IEEE JOURNAL ON EMERGING AND SELECTED TOPICS IN CIRCUITS AND SYSTEMS, VOL. 3, NO. 1, MARCH 2013

Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications

Mihai T. Lazarescu

2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing

Bringing IoT and Cloud Computing towards Pervasive Healthcare

Charalampos Doukas
Dept. of Information & Communication Systems Engineering
University of the Aegean
Samos, Greece
doukas@aegean.gr

Ilias Maglogiannis
Dept. of Computer Science & Biomedical Informatics
University of Central Greece
Lamia, Greece
imaglo@ucg.gr

Internet of Things (IoT)on Bio-Technology

Himadri Nath Saha[1], Supratim Auddy[1], Subrata Pal[1], Shubham Kumar[1], Subhadeep Jasu[1], Rocky Singh[1], Rakhee Singh[1], Swarnadeep Banerjee[1], Priyanshu Sharan[1], Ankita Maity[2]

[1]Deptt. Of Computer Science & Engineering

[2]Deptt. Of Information Technology

[1]Institute of Engineering & Management, Kolkata

[2]Netaji Subhas Engineering College, Kolkata

Vol.8/No.1 (2016) INTERNETWORKING INDONESIA JOURNAL

Ambient Environmental Quality Monitoring Using IoT Sensor Network

Arko Djajadi, *Member, IEEE*, Michael Wijanarko

Worked out Examples

Exercise 1. Nodes A, B and C in Figure periodically collect and send temperature samples to the remote sink. The transmission phase is managed through a dynamic clustering approach which work as follows: two nodes send their samples to the cluster head which then takes the average out of all the sample (two received + one obtained locally) and sends a single packet to the SINK. The cluster head role is assigned in a round robin fashion starting from node A (node A, then B, then C, then A, etc.) (when cluster head is C, B sends its message directly to C, and vice versa not through A). Find the network lifetime (time to the first death) with the following parameter set:

- energy required to operate the TX/RX circuitry E_c = 6[μJ/packet],
- energy required to support sufficient transmission output power $E_{tx}(d) = k \times d^2$ [nJ/packet], being k=120 [nJ/packet/m2]
- energy for taking the average of 3 samples E_p= 4 [uJ]
- initial energy budget E_b=122[μJ] for all the three nodes

Solution:

The energy consumed by the three nodes in one collection round is:

$$E_C = E_B = 2E_c+E_{tx}(5m)+E_{tx}(10m)+2E_c+E_p+E_c+E_{tx}(\sqrt{125}m) = 61[\mu J]$$

$$E_A = 2E_c + 2E_{tx}(5m) + 2E_c + E_p + E_c + E_{tx}(10m) = 52[\mu J]$$

Two full rounds of data collection can be performed.

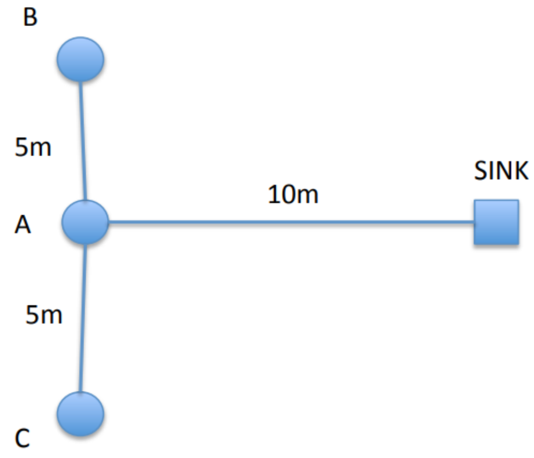


Fig: Reference topology

Exercise 2: A linear wireless sensor network is composed of 5 nodes. Each node is d=5 m away from its closest neighbor. Assuming that: (i) the energy required to operate the TX/RX circuitry, Ec=50 [nJ/bit], the energy required to support sufficient transmission output power Etx(d) = kd2 [nJ/bit], being k=1 [nJ/bit/m2], (iii) packets of b=2000 [bits], tell if its more energy convenient to use direct transmission (from A to E) or minimum per-transmission energy routing (A-B-C-D-E)

Solution

The total consumed energy under direct transmission (A-E) is:

$$E_{direct} = E_c b + E_{tx}(4d)b + E_c b = 1[mJ]$$

The total consumed energy under minimum per-transmission energy routing (A-B-C-D-E) is:

$$E_{minenergy} = 4b(2Ec + Etx(d)) = 1[mJ]$$

Worked out Examples

Exercise 3. Sensor node 1 and sensor node 2 are equipped with cameras and collect images with size I=12.8[kbyte]. The two sensors have to deliver the images to sensor 3 by using packets whose length is L=128[byte]. Assuming that: the energy required to operate the TX/RX circuitry is Ec=6 [uJ/packet], the energy required to support sufficient transmission output power Etx(d) = kd2 [nJ/packet], being k=120 [nJ/packet/m2], find the total energy consumption (energy consumed by sensor 1, sensor 2 and sensor 3) to deliver one single image each in the following two cases:

- sensor 1 and sensor 2 send directly the images to the sink.
- sensor 1 sends the image to sensor 2, sensor 2 sends to sensor 3 its own image and a compressed version of sensor 1s image (compression ratio 0.1, that is, the compressed image has size .1 x I). In this case the energy required by sensor 2 to compress the image is Ep= 0.1 [µJ] for each packet of the original uncompressed image.

Solution:

The uncompressed image requires N = I/L = 100 packets to be delivered; the compressed image requires 0.1N/L = 10 packets to be delivered.

In Case 1, the energy consumed by the three sensor nodes is:

$E_1 = 100[E_c + E_{tx}(10[m])]$

$E_2 = 100[E_c + E_{tx}(5[m])]$

$E_3 = 200E_c$ The total energy is, therefore,

E_1 tot $= 400E_c + 100E_{tx}(10[m]) + 100E_{tx}(5[m]) = 800[\mu J] + 1200[\mu J] + 300[\mu J] = 2.3[mJ]$

In Case 2, the energy consumed by the three sensor nodes is:

$E_1 = 100[E_c + E_{tx}(5[m])]$

$E_2 = 110[E_c + E_{tx}(5[m])] + 100Ep$

$E_3 = 110E_c$

The total energy is, therefore,

E_1 tot $= 320E_c + 210E_{tx}(5[m]) + 100E_p = 640[\mu J] + 630[\mu J] + 10[\mu J] = 1.28[mJ]$

Exercise 4. Sensor nodes 1, 2 and 3 run the SPARE MAC protocol, they are all in range and they have one slot each in the Data Sub-Frame. Sensor 1 and 3 have traffic towards sensor 2 characterized by a Poisson point process with intensity 1= 2[packet/frame] 3= 1[packet/frame], respectively. Find out the probability that the transmissions of sensor 1 and sensor 3 do collide at sensor 2.

Solution:

The collision probability is the probability that both sensor 1 and sensor 3 have at least one packet ready for transmission towards sensor 2,

that is,

$$P=(1-e^{-\lambda 1})(1-e\ \lambda 2) = 0.546$$

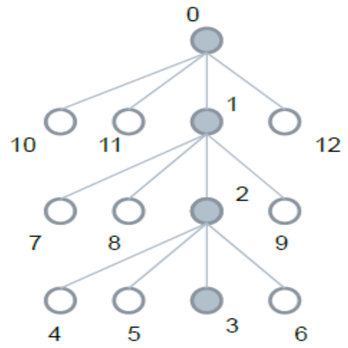
Exercise 5 A PAN Coordinator has 16 available short addresses to be assigned to ZigBee associated devices. Assuming that: the number of ZigBee routers per level in the tree is R_m=1, the number of ZigBee devices per level in the tree is D_m=3, find out the maximum tree depth and design an address distribution schedule.

Solution

Each level in the tree has R_m + D_m=4 devices and thus requires 4 addresses. The total number of addresses which are available for devices (routers and end devices) is 15 (one address out of the original 16 ones is assigned by default to the PAN coordinator). Thus, the total number of "full" (composed of 4 devices) levels which can be supported by the tree is:

$$L_m = \left\lfloor \frac{15}{4} \right\rfloor = 3.$$

A feasible address assignment is reported in Figure



Exercise 6. Find out the Expected Transmission Time (ETT) for the two wireless links in the figure assuming packet size l=128[byte]. Each link is labeled with the corresponding packet error probability, p, the nominal data rate, r, and the propagation delay, t. Assume negligible size for the acknowledgements and a repetition time-out equal to the round trip time for the link.

Solution:

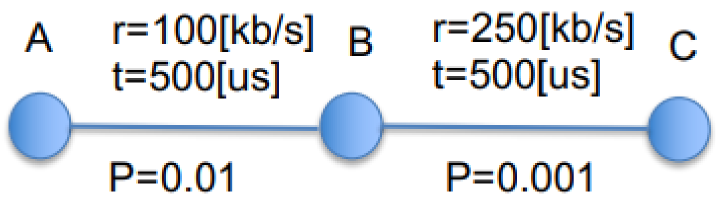
The total time for transmitting a packet through link i and receive the corresponding ACK is: $T_i = \frac{1}{ri} + 2t$.

Thus we have,

T_1=11.24[ms] and

T_2=5.096[ms].

The ETX for the two links is:



ET XAB = $\frac{1}{1-0.01}$ = 1.01 and ET XAB = $\frac{1}{1-0.001}$ = 1.001.

The ETT for the two links is then: ET T_{AB} = ET $X_{AB}T_1$ and ET T_{BC} = ET $X_{BC}T_2$

Exercise 7: A RFID collision arbitration system is based on multi-frame dynamic frame ALOHA. Find out the efficiency of the collision arbitration process if the initial number o tags is N=3 and the initial frame size is r=2.

Solution

The efficiency is defined as $\eta = \frac{N}{L_N}$, where L_N is the average resolution time with a initial tag population of N tags. L_3 can be calculated by applying the recursive formula studied during classes. Namely,

$$L_3 = 2 + P(S = 0)L_3 + P(S = 1)L_2 + P(S = 2)L_1;$$

where $P(S = 2) = 0$, $P(S = 0) = 14$ and $P(S = 1) = \frac{3}{4}$

Similarly, L_2 can be written as:

$$L_2 = 2 + P(S = 0)L_2 + P(S = 1)L_1;$$

where $P(S = 0) = 1\ 2$, $P(S = 1) = 0$. Solving for L_2 and substituting the obtained valued back in the formula of L_3 we get $L_3 = \frac{20}{3}$

The efficiency is finally $\eta = \frac{9}{20}$.

Exercise 8: A Dynamic Frame ALOHA system is used to arbitrate 4 tags. What is the average throughput after the first two frames of the arbitration process knowing that the respective frame lengths are r_1=2, r_2=2?

Solution:

After the first frame either 0 or 1 tag are resolved. The probability to solve one tag after the first frame is $2\frac{4}{3}(\frac{1}{2})^4 = \frac{1}{2}$. If one tag is resolved after the first frame, 3 tags are left; either 1 or 0 tags can be resolved in this case with probability $\frac{3}{4}$ and $\frac{1}{4}$ respectively. If zero tags are resolved after the first frame, the initial 4 tags are left, thus with probability $\frac{1}{2}$ one tag is resolved after the second frame.

In summary, the average number of tags solved after 2 frames under the given conditions is:

$$E[S] = \frac{1}{2}(1 + \frac{3}{4}) + \frac{1}{2}\frac{1}{2}1 = \frac{9}{8}$$

Exercise 9. If value at HLEN in IPv4 is 1101. Find the size of option and padding field.

Solution:

HLEN value = 1101= 13 Bytes

Total No. of Bytes in the Header = 13x 4 = 52 Bytes

The first 20 Bytes are the main header

Therefore, options + Padding field = 32 Bytes

Worked out Examples

Exercise 10. Nodes A, B, C and D in the figure run MQTT protocol and they are all subscribed to topic1. Assuming that node D issues at time t=0 a PUBLISH message on topic1 find the time of arrival of such PUBLISH message at nodes A, B and C. Each link in the figure is characterized by the nominal data rate (r), and the propagation delay (t). The PUBLISH message has size L=40[byte].

Solution:

$$T_{D-broker} = L/100[b/s] + 1[ms] = 3.201[s]$$

$$T_{broker-A} = L/20[kb/s] + 10[ms] = 26[ms]$$

$$T_{broker-B} = L/5[kb/s] + 5[ms] = 69[ms]$$

$$T_{broker-C} = L/10[kb/s] + 5[ms] = 37[ms]$$

$$T_A = T_{D-broker} + T_{broker-A} = 3.227[s]$$

$$T_B = T_{D-broker} + T_{broker-B} = 3.27[s]$$

$$T_C = T_{D-broker} + T_{broker-C} = 3.238[s]$$

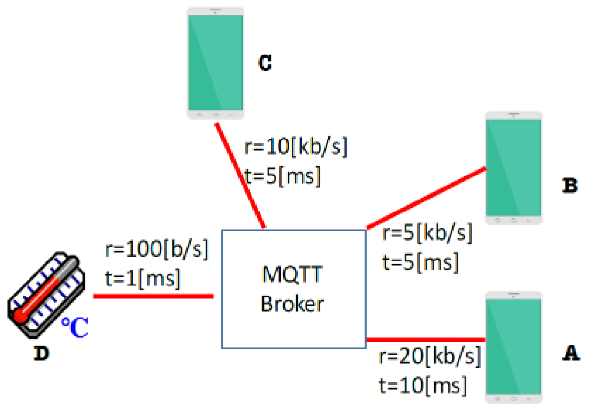


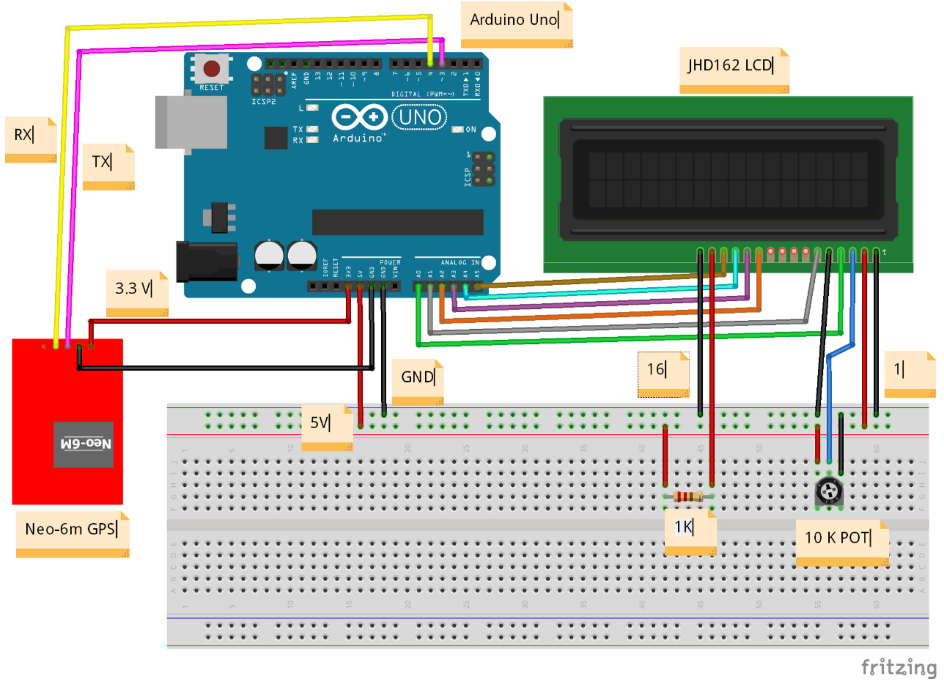
Fig: Reference topology

Worked out Examples

Exercise 11. You have the following components – 1k Ohm resistor, 10k potentiometer, USB cable type A/B, S1216R GPS module with external GPS antenna and an 16x2 LCD display. Run a code using Arduino UNO to show GPS interface.

Solution

```
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
#include <TinyGPS.h>
//long   lat,lon; // create variable for latitude and longitude object
float lat = 28.5458,lon = 77.1703; // create variable for latitude and longitude object
SoftwareSerial gpsSerial(3,4);//rx,tx
LiquidCrystal lcd(A0,A1,A2,A3,A4,A5);
TinyGPS gps; // create gps object
void setup(){
Serial.begin(9600); // connect serial
//Serial.println("The GPS Received Signal:");
gpsSerial.begin(9600); // connect gps sensor
lcd.begin(16,2);
}

void loop(){
   while(gpsSerial.available()){ // check for gps data
   if(gps.encode(gpsSerial.read()))// encode gps data
   {
gps.f_get_position(&lat,&lon); // get latitude and longitude
  // display position
  lcd.clear();
  lcd.setCursor(1,0);
  lcd.print("GPS Signal");
  //Serial.print("Position: ");
  //Serial.print("Latitude:");
  //Serial.print(lat,6);
  //Serial.print(";");
  //Serial.print("Longitude:");
  //Serial.println(lon,6);
  lcd.setCursor(1,0);
  lcd.print("LAT:");
  lcd.setCursor(5,0);
  lcd.print(lat);
  //Serial.print(lat);
  //Serial.print(" ");

  lcd.setCursor(0,1);
  lcd.print(",LON:");
  lcd.setCursor(5,1);
  lcd.print(lon);

 }
}

String latitude = String(lat,6);
  String longitude = String(lon,6);
Serial.println(latitude+";"+longitude);
delay(1000);
}
```

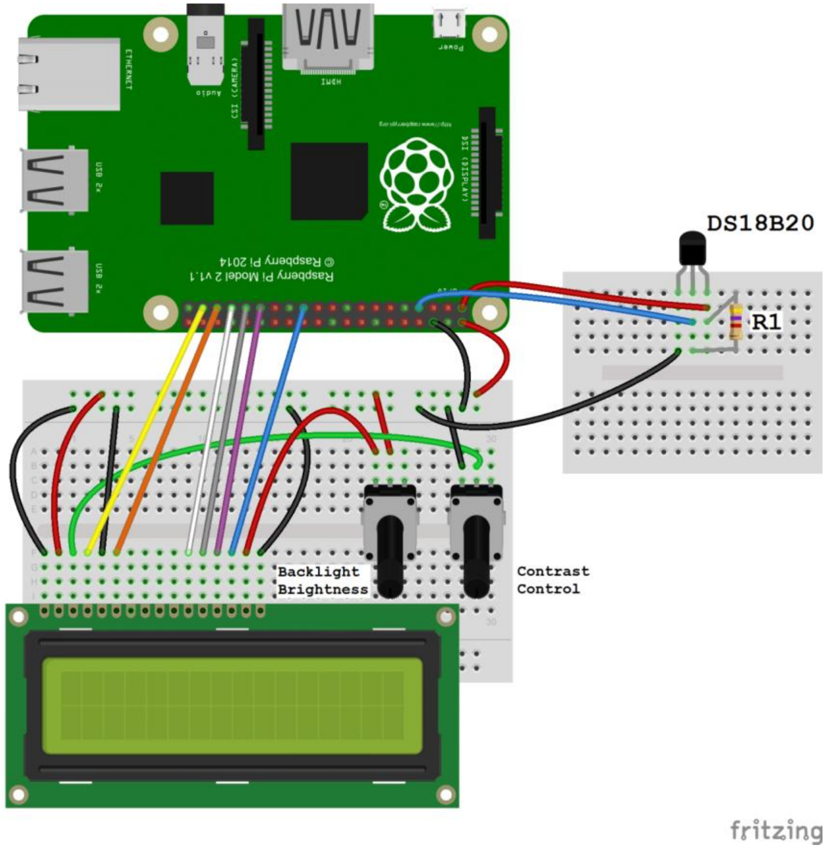


Exercise 12. You are provided with the following components- Raspberry Pi, SD card, monitor, HDMI cable, keyboard, mouse, power supply, 1 red and blue LED, 2 1K resistors, push button, jumper wires, breadboard, buzzer and LM35 temperature sensor. Run a code to process the input.

Solution

This example shows how to get an analog input from GPIO pins and process the input. An infinite loop runs over the sensors which records a temperature every second.

```
import os
import glob
import time

os.system('modprobe w1-gpio')
os.system('modprobe w1-therm')

base_dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0]
device_file = device_folder + '/w1_slave'

def read_temp_raw():
   f = open(device_file, 'r')
   lines = f.readlines()
   f.close()
   return lines

def read_temp():
   lines = read_temp_raw()
   while lines[0].strip()[-3:] != 'YES':
      time.sleep(0.2)
      lines = read_temp_raw()
   equals_pos = lines[1].find('t=')
   if equals_pos != -1:
      temp_string = lines[1][equals_pos+2:]
      temp_c = float(temp_string) / 1000.0
      temp_f = temp_c * 9.0 / 5.0 + 32.0
      return temp_c, temp_f

while True:
                print(read_temp())
                time.sleep(1)
```



Exercise 13. Create a database table for EMPLOYEE using MySQLbd, a logical design using Python.

Solution:

```
#!/usr/bin/python

import MySQLdb

# Open database connection
db = MySQLdb.connect("localhost","testuser","test123","TESTDB" )

# prepare a cursor object using cursor() method
cursor = db.cursor()

# Drop table if it already exist using execute() method.
cursor.execute("DROP TABLE IF EXISTS EMPLOYEE")

# Create table as per requirement
sql = """CREATE TABLE EMPLOYEE (
        FIRST_NAME  CHAR(20) NOT NULL,
        LAST_NAME  CHAR(20),
        AGE INT,
        SEX CHAR(1),
        INCOME FLOAT )"""

cursor.execute(sql)

# disconnect from server
db.close()
```

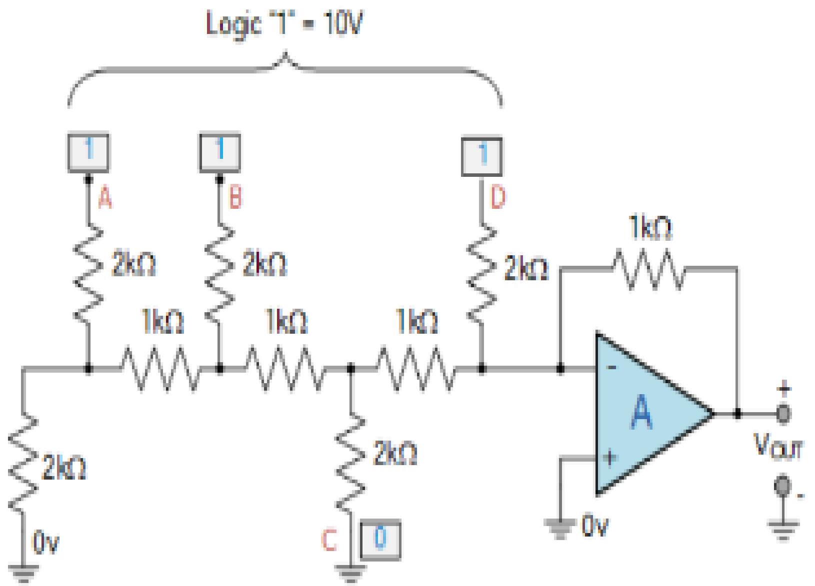
Exercise 14. Find all the output voltages of a weighted resistors D/A converter circuit of inputs A, B, C and D, if 0 = 0V and 1 = 10V.

Solution

$$V_{OUT} = \frac{1V_A + 2V_B + 4V_C + 8V_D}{2^n}$$

$$V_{OUT} = \frac{1\times10 + 2\times10 + 4\times0 + 8\times10}{16}$$

$$V_{OUT} = \frac{110}{16} = 6.875 \text{ Volts}$$

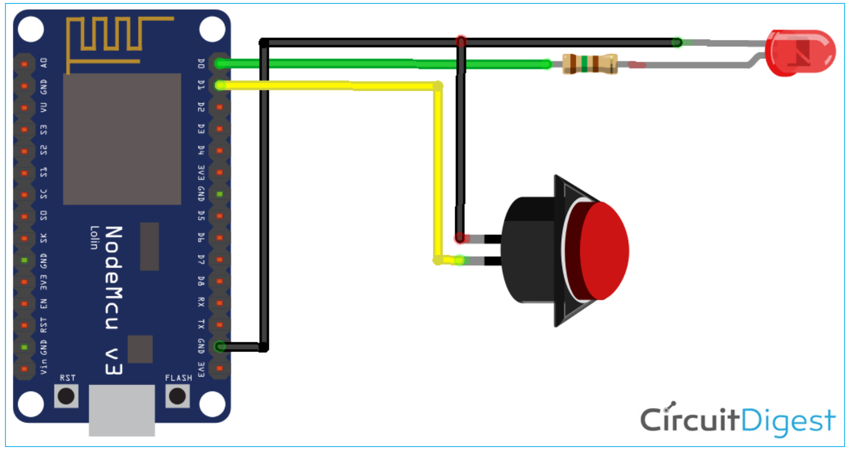


Worked out Examples

Exercise 15. What is the sum after the following loop terminates?

```
sum = 0
idea = 0
while sum >5:
item = 1
sum = item
if sum >4: break
print (sum)
```

Exercise 16. Performing MQTT communication with ESP8266/NodeMCU using Arduino IDE.



Solution

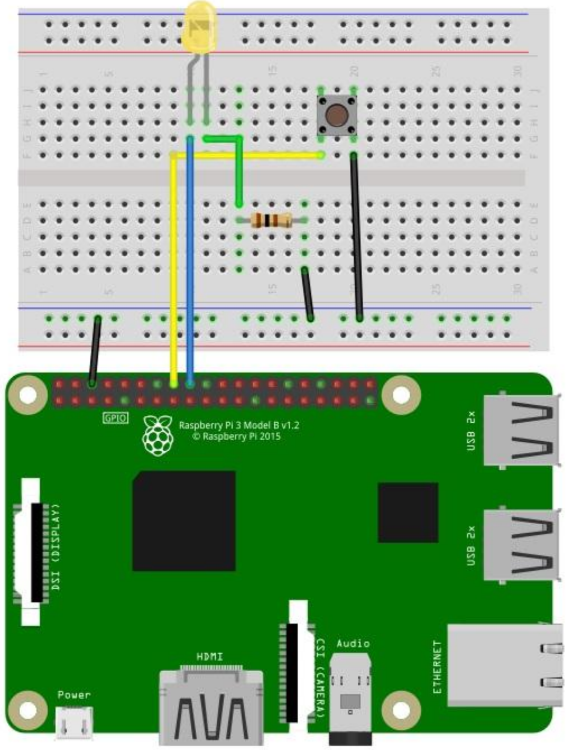
```
#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#define LED D0

const char* ssid = "admin";

const char* password =  "12345678";

const char* mqtt_server =
"mqtt.eclipse.org";

const int mqtt_port = 1883;

WiFiClient espClient;

PubSubClient client(espClient);

void setup()

{

 pinMode(LED, OUTPUT);

 pinMode(D1,INPUT_PULLUP);

 Serial.begin(115200);

 WiFi.begin(ssid, password);

 while (WiFi.status() != WL_CONNECTED)

 {
delay(500);
  Serial.println("Connecting to WiFi..");
 }
 Serial.print("Connected to WiFi :");
 Serial.println(WiFi.SSID());
 client.setServer(mqtt_server, mqtt_port);
 client.setCallback(MQTTcallback);
 while (!client.connected())
 {
  Serial.println("Connecting to MQTT...");
  if (client.connect("ESP8266"))
  {
   Serial.println("connected");
  }
  else
  {
   Serial.print("failed with state ");
   Serial.println(client.state());
   delay(2000);
  }
 }
 client.subscribe("esp/test");
}
void MQTTcallback(char* topic, byte* payload, unsigned int length)
{
 Serial.print("Message received in topic: ");
 Serial.println(topic);
 Serial.print("Message:");
 String message;
 for (int i = 0; i < length; i++)
 {
  message = message + (char)payload[i];
 }
 Serial.print(message);
 if (message == "on")
 {
  digitalWrite(LED, HIGH);
 }
 else if (message == "off")
 {
  digitalWrite(LED, LOW);
 }
 Serial.println();
 Serial.println("-----------------------");
}
void loop()
{
 if(digitalRead(D1)==0)
 {
 client.publish("esp/test1", "Hello from ESP8266");
 delay(1000);
 }
 else;
 client.loop();
}
```

Exercise 17. Interface a Push Button with Raspberry Pi.

Solution

This example shows how to get input from GPIO pins and process the state of the LEDS

```
import RPi.GPIO as GPIO
import time
button = 16
led = 18
def setup():
GPIO.setmode(GPIO.BOARD)
GPIO.setup(button, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(led, GPIO.OUT)
def loop():
while True:
button_state = GPIO.input(button)
if button_state == False:
GPIO.output(led, True)
print('Button Pressed...')
while GPIO.input(button) == False:
time.sleep(0.2)
else:
GPIO.output(led, False)
def endprogram():
GPIO.output(led, False)
GPIO.cleanup()
if __name__ == '__main__':
setup()
try:
loop()
except KeyboardInterrupt:
print 'keyboard interrupt detected'
endprogram()
```



Exercise 18. An IEEE 802.15.4 mote wants to access the wireless channel. Assuming that the probability of finding the channel busy at each sensing event is p=0.05, find the probability, Q, that the mote is not able to access the channel within the first two backoff/sensing rounds.

Solution

The backoff procedure starts with: NB=0, CW=2, BE=2.

At the first attempt, the accessing station waits for Rand(0, 2^BE-1) backoff periods and then senses the channel.

If the channel is sensed free for CW consecutive backoff periods, the station accesses the channel.

Otherwise, NB=1, CW=2, BE:=BE+1, and go back to 1.

Lets dene the backoff success probability, Psucc, as the probability that the channel is sensed free in all the CW backoff periods. That is,

$P_{succ} = (1 - p)^2 = 0.9$

The probability that the backoff procedure ends at the *i - th* attempt is:

$P_{succ}(i) = P_{succ}(1 - P_{succ})^{(i-1)}$

The require probability that the backoff procedure "fails" for the first two attempts is:

$P = 1 - P_{succ}(1)$ □ $P_{succ}(2) = 0.01$

Exercise 20. The nodes in the Figure 3.1 exchange packets of L=128[byte] which are acknowledged with ACKs of the same size. The figure reports for each wireless link the corresponding link capacity C and Expected Transmission Count (ETX). Find the Expected Transmission Time (ETT) metric for all the wireless links (assume negligible propagation delay); find the shortest path from node A to node B using the ETT as routing metric.

Solution

The Round Trip Time (RTT) for each link can be calculated as RTT = 2 x $\frac{L}{C}$

Where, C is the capacity of the specific link.

The ETT can then be calculated as ETT = ETX RTT.

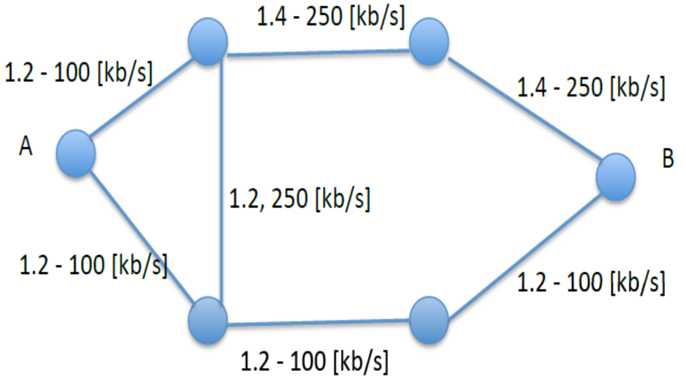


Fig: Reference topology

The following Figure reports the ETT values for all the links and the corresponding shortest path between A and B.

Exercise 19. A sensor node runs the IEEE 802.15.4 Carrier Sensing Multiple Access (CSMA); what is the average backoff time after the third failed attempt to access the channel? (the node has tried to access the channel for 3 times and the channel was busy)

Solution

After the third failed attempt, the sensor draws the random backoff time in the window [0, 24-1]. On average, it will have to wait 7.5 backoff periods.

Exercises

1. A wireless link between two sensor nodes is characterized by the following attenuation (in dB) L = 70+32log(d), where d is the link length (expressed in km). Assuming that the receiver sensitivity is P_{min} = -95[dBm] , find the minimum required transmitted power which allows reception if the link length is d = 500[m].

2. A sensor node performs channel access according to the CSMA/CA scheme of the IEEE 802.15.4 standard. Assuming that the probability of finding the channel busy is p=0.05 at each back off period, what is the probability that the sensor node does actually access the channel within the first two tries.

3. A sensor node performs channel access according to the CSMA/CA scheme of the IEEE 802.15.4 standard. Assuming that the probability of finding the channel busy is p=0.1 at each back off period, find the probability that the sensor node does actually access the channel within the first two tries.

4. A COAP client issues a CONFIRMABLE message. Assuming that the packet error rate is p=0.2 and the MAX-RETRANSMIT parameter is 2, find the probability that the CONFIRMABLE message in the end goes through.

5. A wireless link is characterized by a Bit Error Rate (BER), p=0.001. Assuming that transmissions on the link use packets of length L=128[byte] which are acknowledged with ACK with size L=8[byte], find an estimate for the Expected Transmission Count (ETX) of the link.

6. A ZigBee network is characterized by the following parameters: L_m=3 (three layers), R_m= 2 (number of zigbee routers), D_m = 2 (number of end devices). Tell how many addresses are needed to support all the devices in the network and plot a consistent address assignment scheme.

7. A Dynamic Frame ALOHA system estimates the current backlog of unresolved tags to be 2. What is the single-frame throughput assuming that the system optimally set the frame length ACCORDING TO THE BACKLOG ESTIMATE and knowing that the real number of unresolved tags is 4.

8. A RFID system based on Dynamic Frame ALOHA is composed of 2 tags. Assuming that the initial frame size is r=4, find the overall collision resolution efficiency, (assume that after the first frame, the frame size is correctly set to the current backlog size).

9. Under the same network topology as Exercise 2, find out the number of sensor nodes for which the direct transmission consumes the same amount of energy as the min per-transmission energy routing.

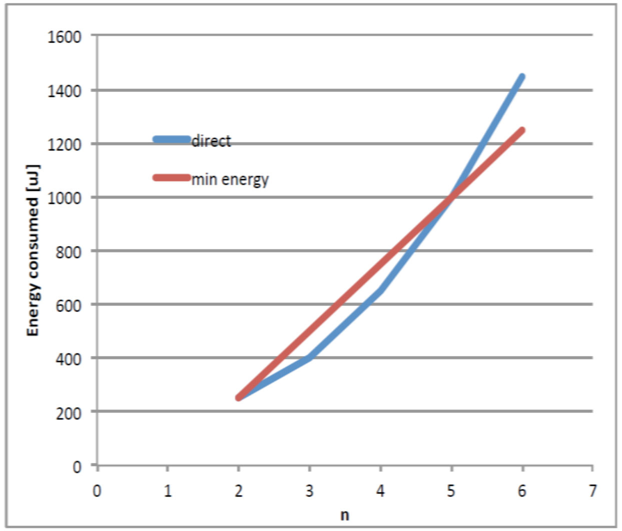


Figure: Energy consumption.

10. In an IPv4 datagram, the M bit is 0, the value of HLEN is 5, the value of total length is 200, and the offset value is 200. What is the number of the first byte and number of the last byte in this datagram? Is this the last fragment, the first fragment, or a middle fragment?

Exercises

11. A packet has arrived in which the offset value is 100, the value of HLEN is 5, and the value of the total length field is 100. What are the numbers of the first byte and the last byte?

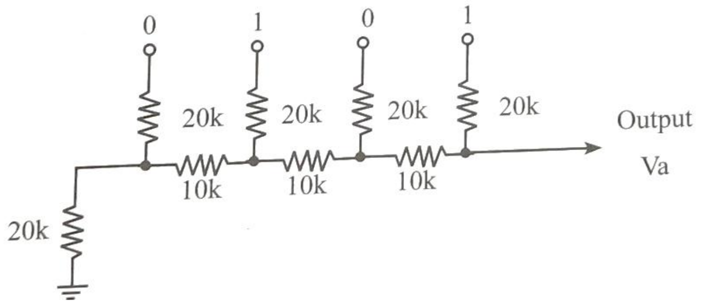
12. Consider a simplified time slotted MAC protocol, where each host always has data to send and transmits with probability p = 0.2 in every slot. There is no back-off and one frame can be transmitted in one slot. If more than one host transmits in the same slot, then the transmissions are unsuccessful due to collision. What is the maximum number of hosts which this protocol can support, if each host has to be provided a minimum throughput of 0.16 frames per time slot?

13. Nodes A and B are connected by a 100 Mbps Ethernet segment with 6 µs propagation delay. Suppose A and B send frames at t=0 and frames get collided. After first collision A draws k = 0 and B draws k = 1. If jam signal is ignored and timeout time is 1 RTT then at what time A's packet get completely delivered to B assuming packet size 100 bits.

14. A 4 port bridge connects the station of 10 Mbps Ethernet LAN. If the average data rate of each station is 1.25 Mbps then how many stations are there?

15. An IP router with a Maximum Transmission Unit (MTU) of 1500 bytes has received an IP packet of size 4404 bytes with an IP header of length 20 bytes. What will be the values of the relevant fields in the header of the third IP fragment generated by the router for this packet?

16. Given below is the circuit of binary ladder, with applied digital signals (4 bit). Calculate the analog output voltage at the given input combination, if 0 = 0V and 1 = 16V



17. What will be displayed for the following code?

```
x, y = 1, 2
x, y = y, x
print (x, y)
```

18. What is the sum after the following loop terminates?

```
sum = 0
idea = 0
while sum >5:
item = 1
sum = item
if sum >4: continue
print (sum) (refer exercise)
```

19. Find the codes to create a record of EMPLOYEE executing a SQL INSERT statement. (refer example 13)

20. Find the codes to run automatic lights with light sensors in Arduino UNO using the following components- 1x LED, 1x 220Ω resistor, 1x photoresistor, 1x 10kΩ resistor, jumper wires and breadboard.

References

1. Zhu, Q.; Wang, R.; Chen, Q.; Liu, Y.; Qin, W. *IOT Gateway: Bridging Wireless Sensor Networks into Internet of Things*, IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, **2010**, DOI: 10.1109/EUC.2010.58.

   The paper describes an IOT Gateway system based on Zigbee and GPRS protocols according to the typical IOT application scenarios and requirements from telecom operators, presented the data transmission between wireless sensor networks and mobile communication networks, protocol conversion of different sensor network protocols, and control functionalities for sensor networks, and finally gave an implementation of prototyping system and system validation.

2. Saha, H.N.; Auddy, S.; Pal, S.; Kumar, S.; Singh, R.; Singh, R.; Banerjee, S.; Sharan, P.; Maity, A. *Internet of Things (IoT)on Bio-Technology*, IEEE, Industrial Automation and Electromechanical Engineering Conference, **2017**, DOI: 10.1109/IEMECON.2017.8079624.

   The paper proposes some smart applications using the software and hardware of IoT.

References

3. Doukas, C.; Maglogiannis, I. *Bringing IoT and Cloud Computing towards Pervasive Healthcare*, IEEE, International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, **2012**, DOI: 10.1109/IMIS.2012.26.

   This paper presents a platform based on Cloud Computing for management of mobile and wearable healthcare sensors, demonstrating this way the IoT paradigm applied on pervasive healthcare.

4. Ferdoush, S.; Li, X. *Wireless Sensor Network System Design using Raspberry Pi and Arduino for Environmental Monitoring Applications*, Procedia Computer Science, **2014**, 34, 103 – 110, DOI: 10.1016/j.procs.2014.07.059 .

   The paper describes a WSNS design developed using various hardware available. The system developed is low-cost and highly scalable both in terms of the type of sensors and the number of sensor nodes, which makes it well suited for a wide variety of applications related to environmental monitoring.

5. Zhang, Z.; Cho, M.; Wang, C.; Hsu, C.; Chen, C.; Shieh, S. *IoT Security: Ongoing Challenges and Research Opportunities*, IEEE, International Conference on Service-Oriented Computing and Applications, **2014**, DOI: 10.1109/SOCA.2014.58.

   The paper presents a detailed information about the current security problems related to IoT.

References

6. Vujovic, V.; Maksimovic, M. *Raspberry Pi as a Wireless Sensor Node: Performances and Constraints*, MIPRO, Croatia, **2014**, 1013-1018.

The paper presents Raspberry Pi's abilities to be used as WSN node and SensorWeb node.

7. Gondchawar, N.; Kawitka, R. S. *IoT based Smart Agriculture*, International Journal of Advanced Research in Computer and Communication Engineering. Vol. 5, Issue 6, **2016**, DOI: 10.17148/IJARCCE.2016.56188.

The use of interfacing sensors, Wi-Fi or ZigBee module, camera and actuators with micro-controller and Raspberry Pi for use in agricultural sector has been provided in this paper.

8. Khalil, N.; Abid, M.R.; Benhaddou, D.; Gerndt, M. *Wireless Sensors Networks for Internet of Things*, IEEE, International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP) Symposium on Public Internet of Things, 1-6, **2014**, DOI: 10.3837/ISSNIP.2012.02.002.

This paper presents the integration of wireless sensor networks into IoT, and shed further light on the subtleties of such integration.

References

1. Jose, J. *Internet of Things (IoT)*, Khanna Publishing, **2018.** ★ ★ ★ ★

   A very simply written book for easy understanding. This book covers most basic topics of IoT with concentration on applications. The book has great reviews in online portals and is priced at a very reasonable amount. The minus point of this book is that it does not have exercises that can be solved by the user. Other than that it is a great book that can be owned by the lab.

2. Khan, J. Y.; Yuce. M. R. *Internet of Things (IoT): Systems and Applications*,Jenny Stanford Publishing, **2019**. ★★★

   Excellent book for IoT beginners. Information on IoT hardware, IoT M2M and IoT protocols such as LoRaWAN has been extracted from this book.

3. McEwen, A.; Cassimally, H. *Designing Internet of Things(IoT)*,Wiley Publishing, **2014.** ★ ★ ★ ★

   This book is very systematically arranged. Most information about Arduino, Raspberry Pi and Internet Protocol (in these slides) has been extracted from this book. Various examples and case study has been presented very well in this book. The book covers all necessary topics of Internet of Things. The contents of all chapters have been very well summarized. (The lab already has a printed copy of this book).

References

4. Priyadarshini, A. *Internet of Things : Applications And Challenges In Technology And Standardization*, NIST, **2013**. ★

    Information of applications of IoT has been extracted from this book.

5. Cesana, M. *Internet of things: Exercises*. **2016.** ★ ★

    This is a very handy book for enhancing the IoT numerical solving skills. All exercises has been extracted from this book.

THANK YOU