IoT Enabling Technologies : Hardware

Dr. Bibhas Ghoshal

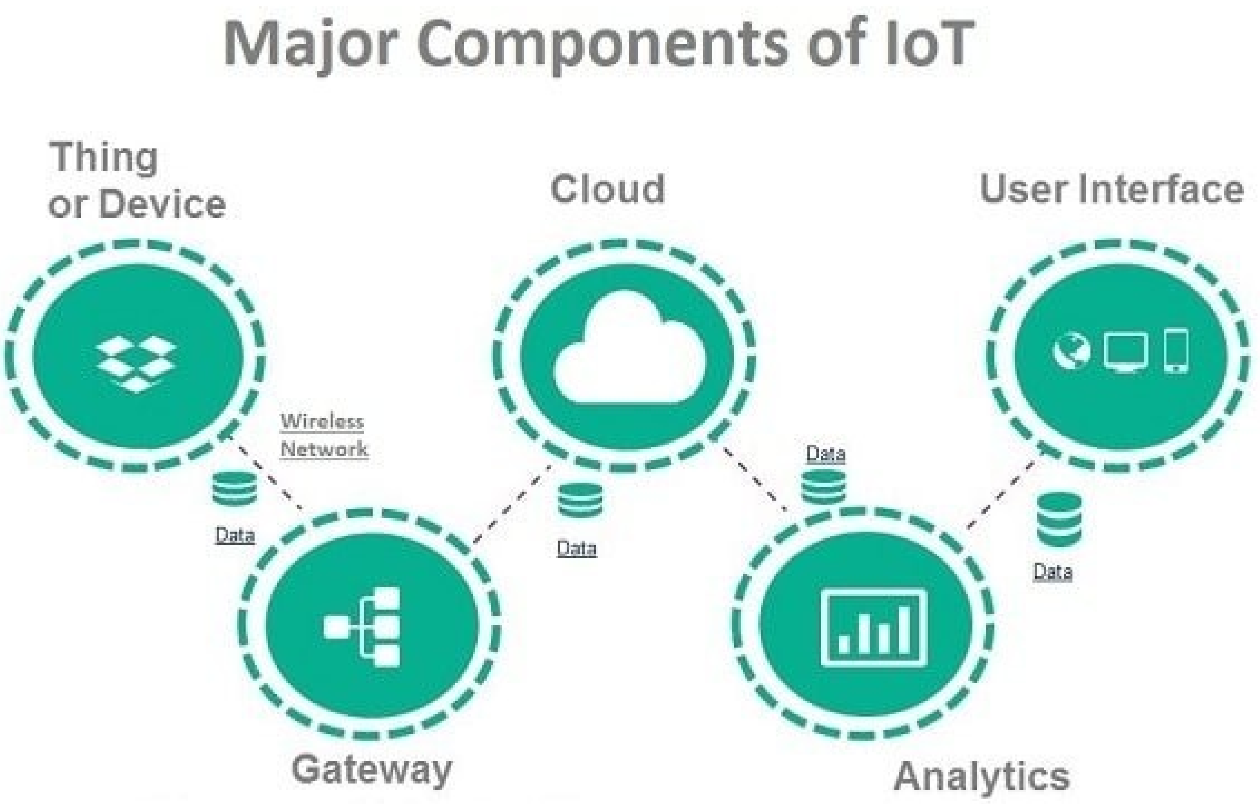
IIIT Allahabad

IoT Devices

- Sensors – Types, Use and Calibration, Sensor arrays
- Sensing Circuits – Smart sensors
- Wireless sensor networks
- Actuators – Types and Use
- Embedded Devices – Microprocessors for IoT
- Battery and Power Management
- IoT Development Platforms
- Interfacing sensors to microcontrollers

IoT System

•



Source: Internet

IoT System Component : Things

Physical things exist in the physical world and are capable of being sensed, actuated and connected.

Examples : industrial robots, goods and electrical equipment

Virtual things exist in the information world and are capable of being stored, processed and accessed.

Examples : multimedia content and application software

Source: Recommendation **ITU-T Y.2060**

Capturing the Real World

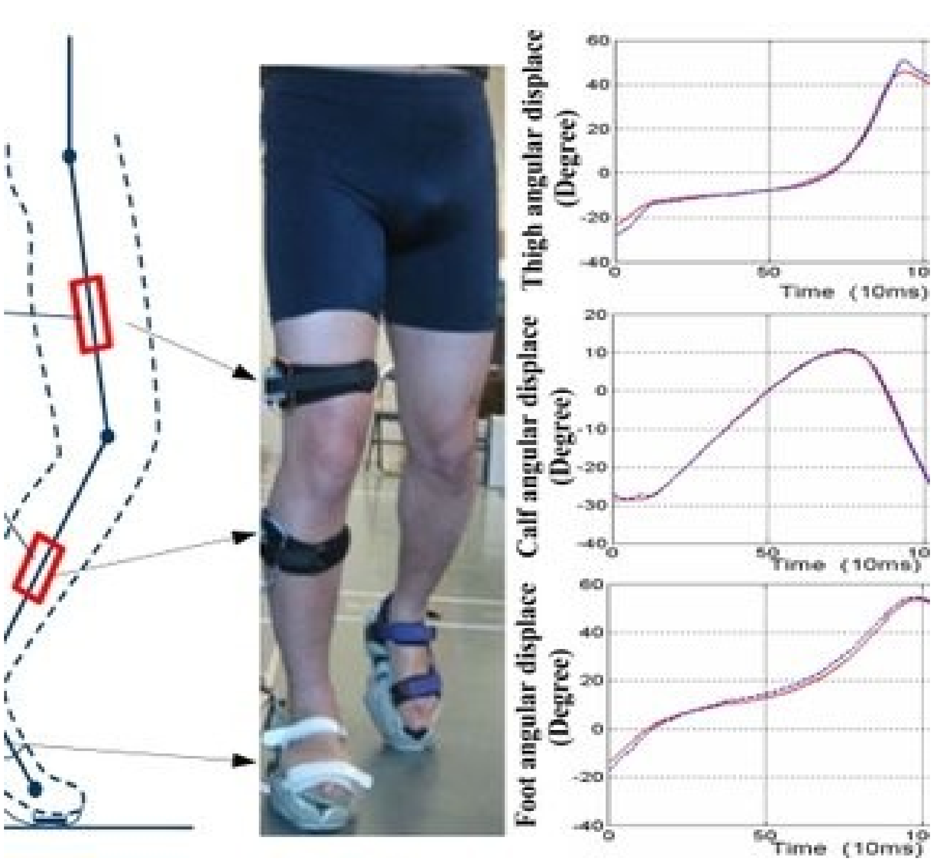
Physical things help in capturing the real world - System states and immediate surroundings

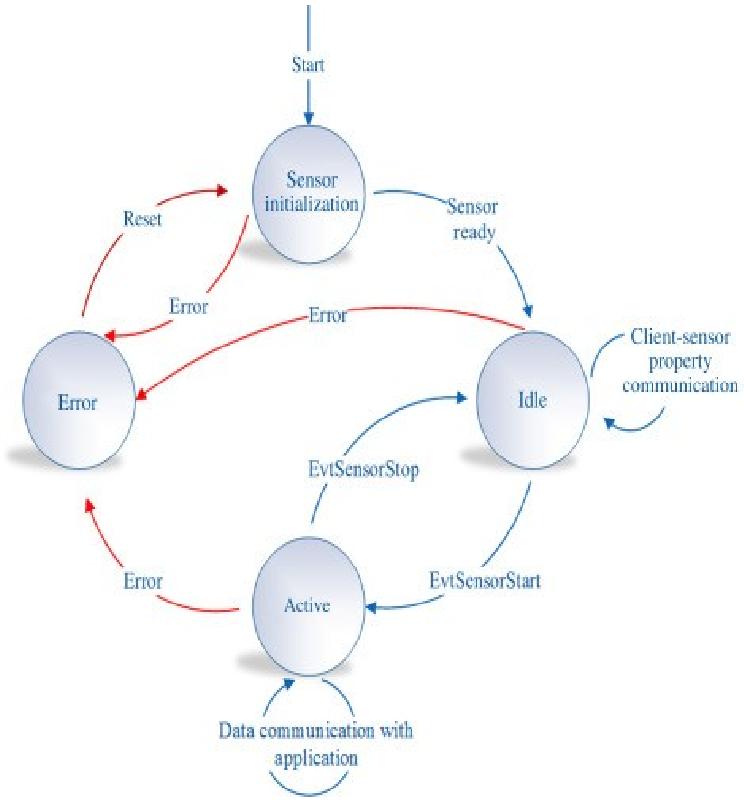
Q: How to capture system states and surroundings?
A : Sensors

Q: How to monitor machines?
A : Sensors

Q: How to monitor humans?
A : Mobile Phone (sensor hub)

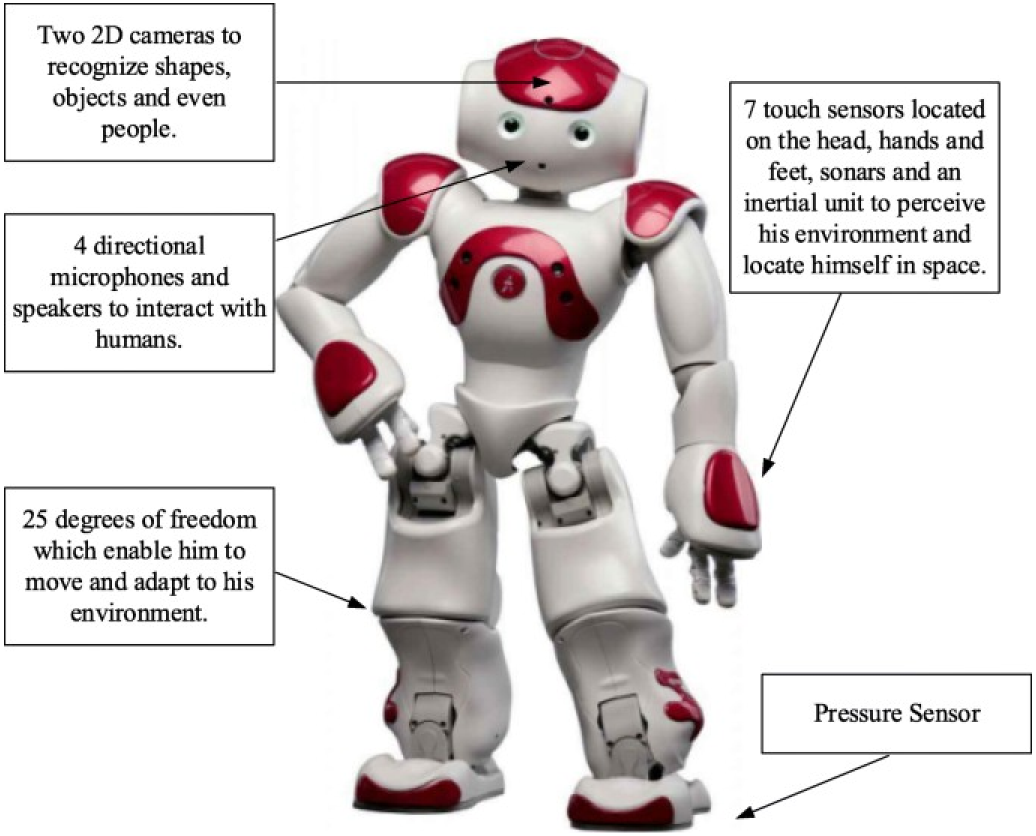
Capturing System States Through Sensors



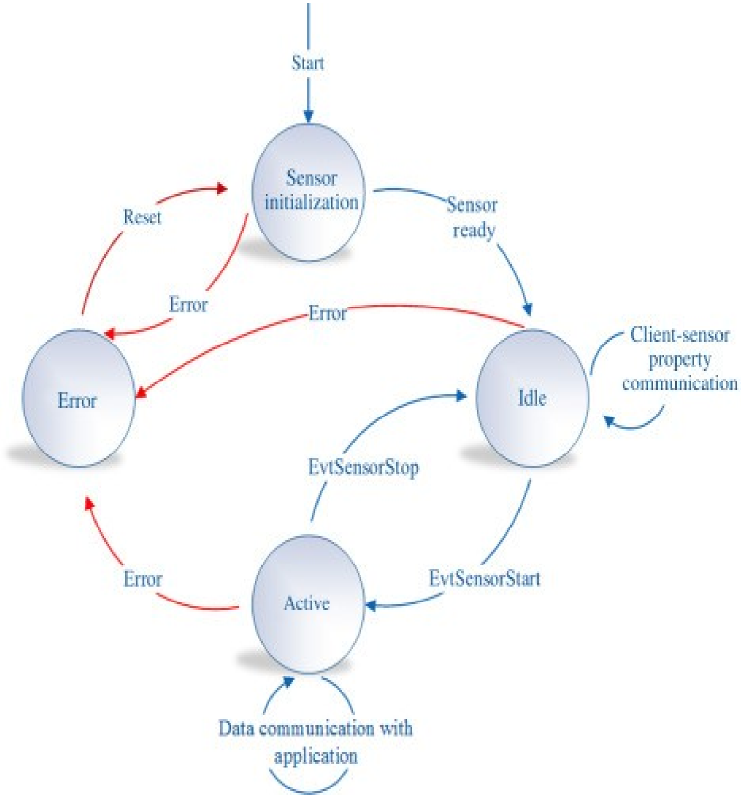


Ref : Gait Analysis Using Wearable Sensors, Weijun Tao et al., Sensors 12(2):2255-83, DOI: 10.3390/s120202255
Source : PubMed

Monitor Machines Through Sensors



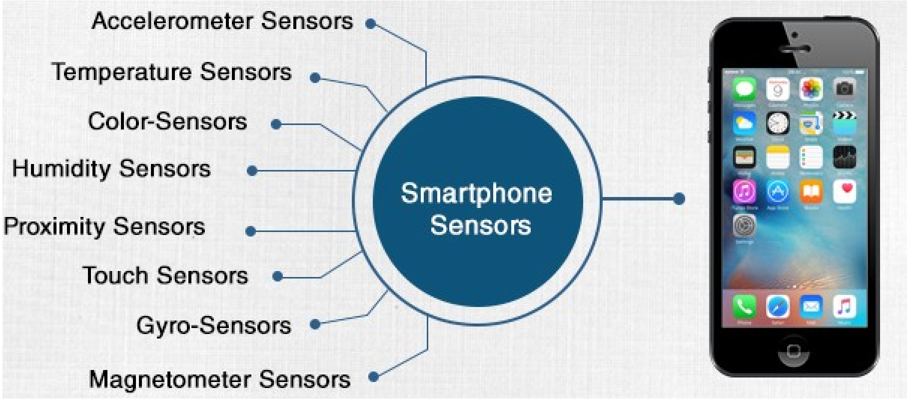
NAO Robot



Monitor Humans Through Sensors

Parental Control through mobile

Spying through mobile apps



Mobile Phone : A Sensor Hub

What are Sensors?

Physical things help in capturing the real world - System states and immediate surroundings

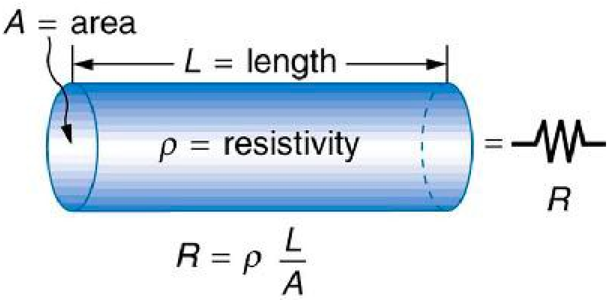
Q: How to capture system states and surroundings?
A : Sensors

Q: How to monitor machines?
A : Sensors

Q: How to monitor humans?
A : Mobile Phone (sensor hub)

Sensors

- Sensors measure or identify physical quantities
- Senses change in physical parameters
- Sensors change physical properties to electrical signals

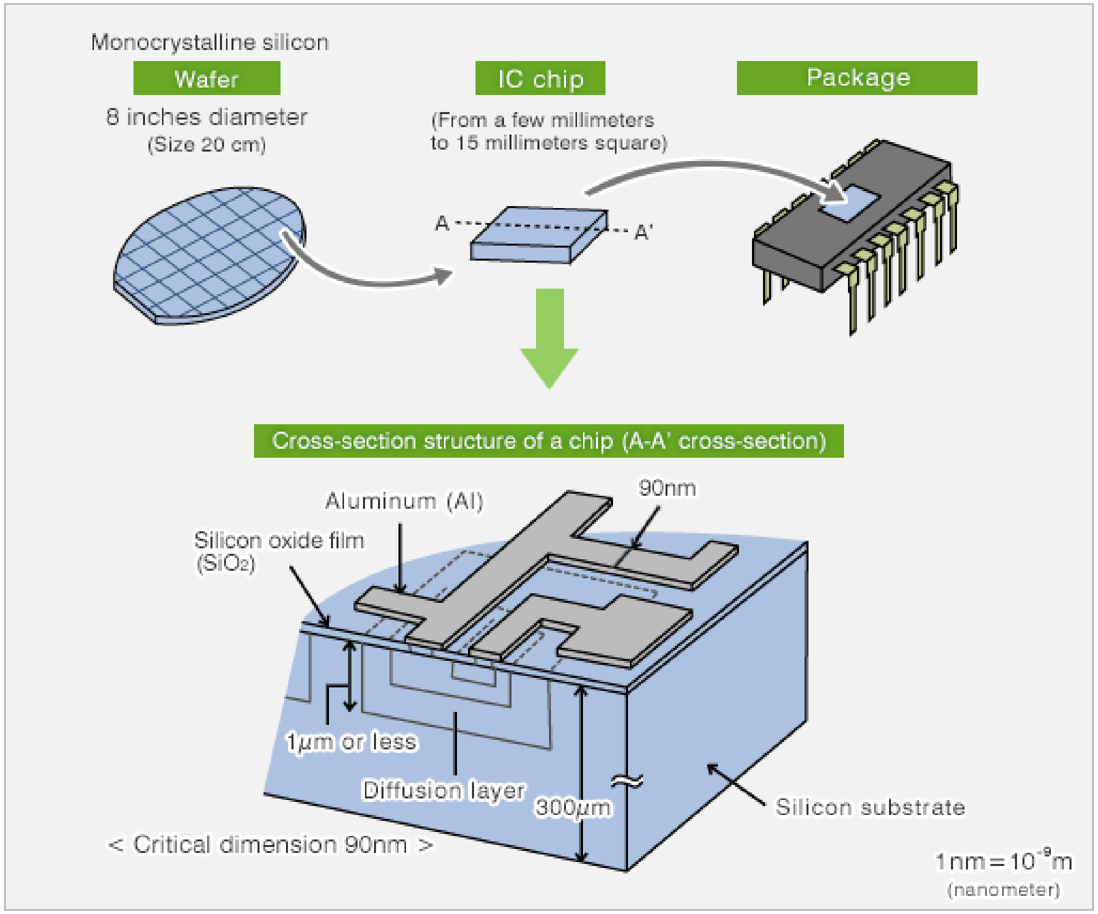


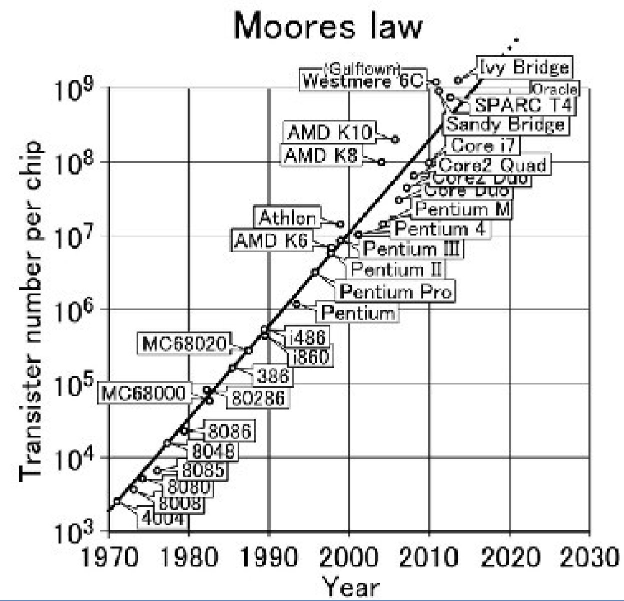
<u>Working of a pressure sensor</u>

- Pressure results in change of shape
- Change of shape results in change of resistance
- Change of resistance results in change in electrical signals

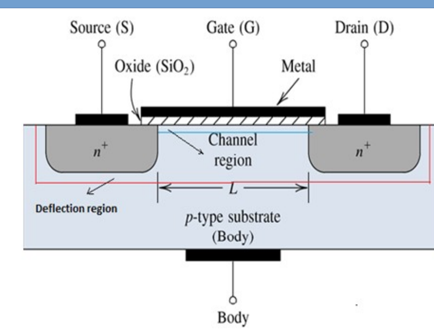
Enabling Technology of Sensors

Semiconductor Technology

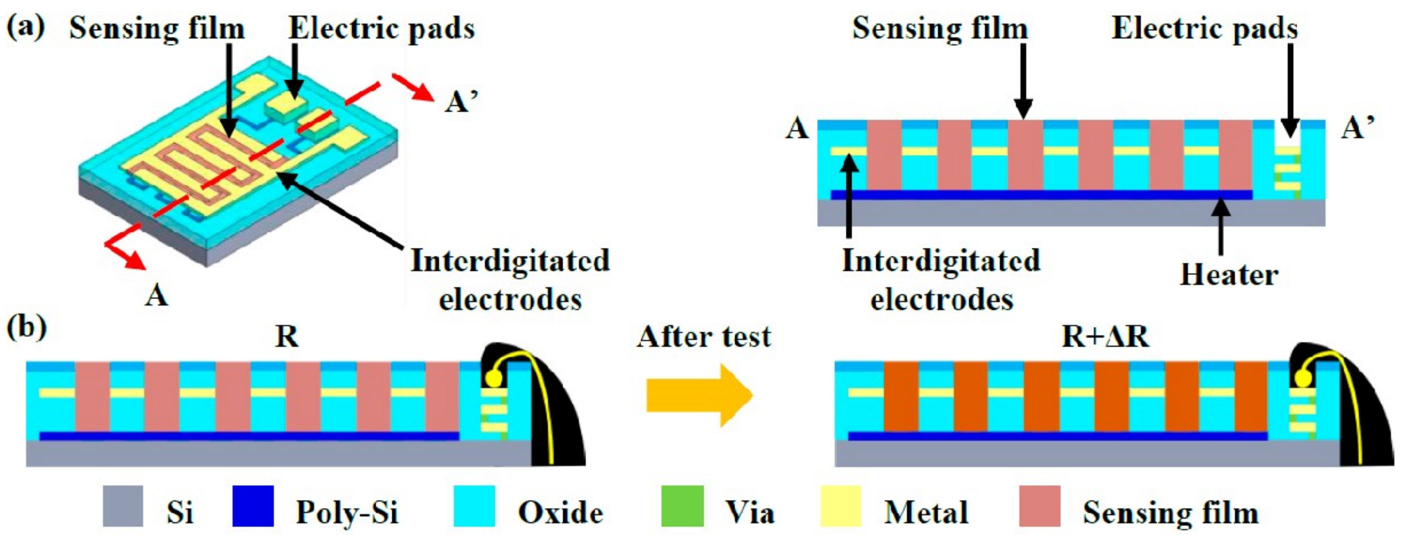




Technology node ≡ Min Feature Size

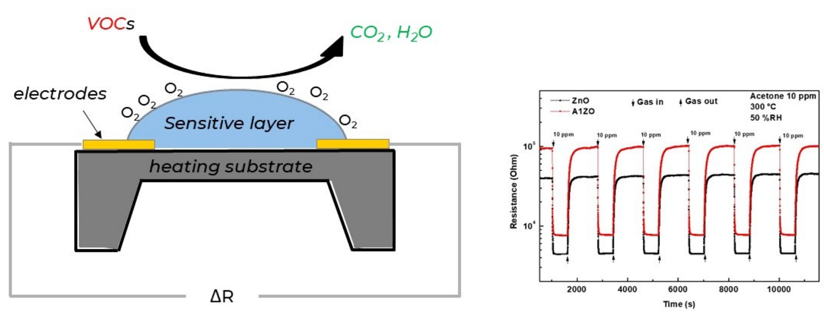


Fabrication of Semiconductor based Sensors



Fabrication of MOS type Gas Sensors

MOS type Gas Sensor



Operating Principle of a MOS type Gas Sensor

Enabling Technology of Sensors

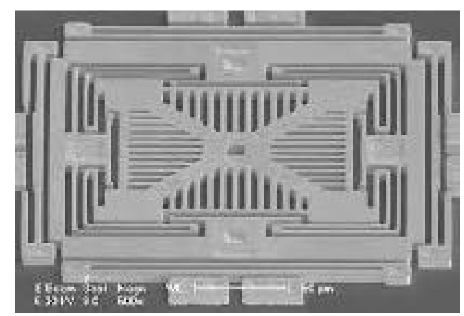
<u>Micro Electro Mechanical Systems (MEMS) Technology</u>

- Miniaturized mechanical and electro-mechanical elements
- Moving structures fabricated on a Silicon substrate
- Made using techniques of micro-fabrication
- MEMS fabrication is similar to CMOS IC fabrication except :

1. Mechanical Properties
2. Feature Size
3. Unconventional materials

Micro motor

Gyroscope

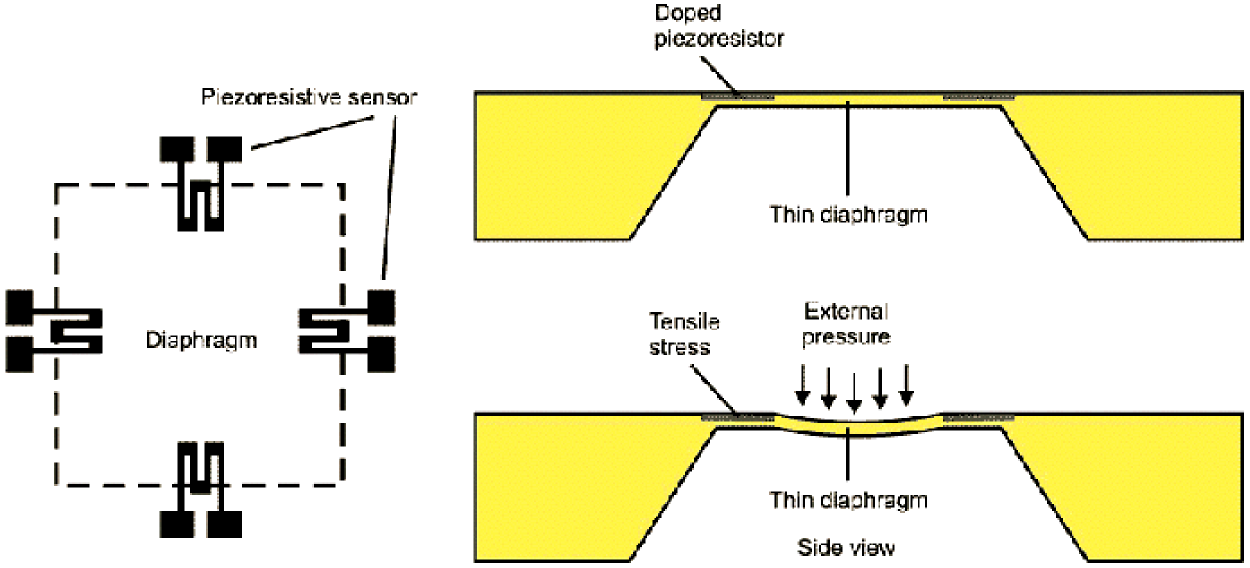


Accelerometer

Ref: CS664 IoT System Design Course,
Instructor : Prof. Amey Karkare, IIT Kanpur

Different Kinds of MEMS Sensors

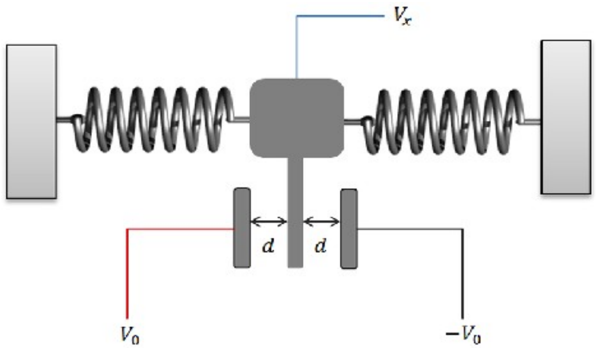
- MEMS Inertial sensors
- MEMS Pressure sensors
- MEMS Gas sensors
- MEMS Humidity and Temperature sensors
- MEMS chemical sensors
- MEMS Bio sensors

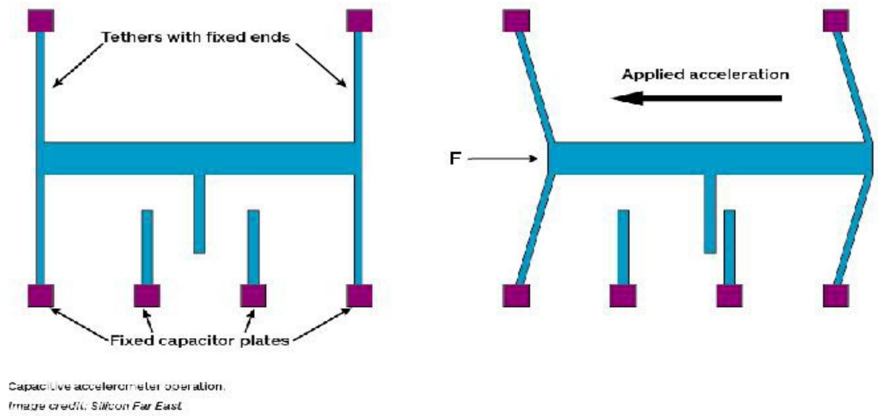


MEMS Pressure sensors

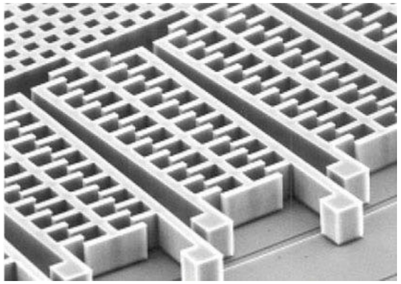
Ref: CS664 IoT System Design Course,
Instructor : Prof. Amey Karkare, IIT Kanpur

MEMS Accelerometer

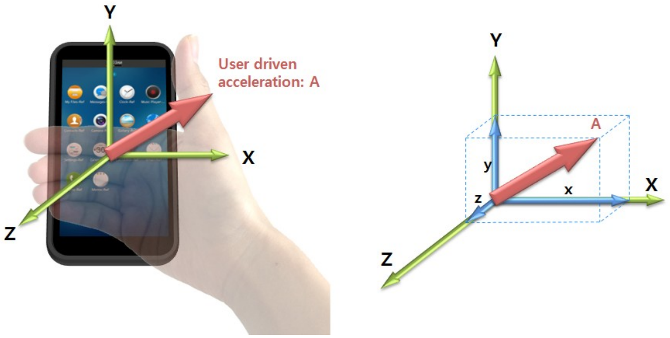




Acceleration – Displacement – Electrical Signal



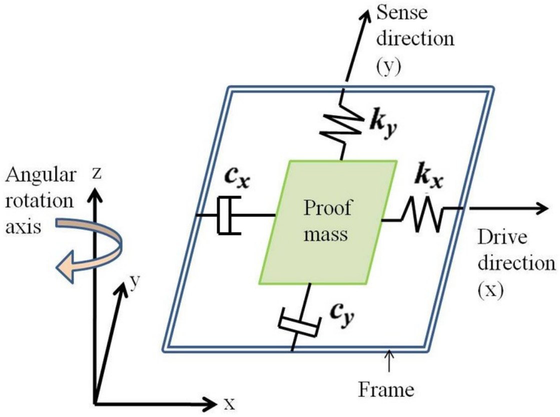
MEMS Accelerometer



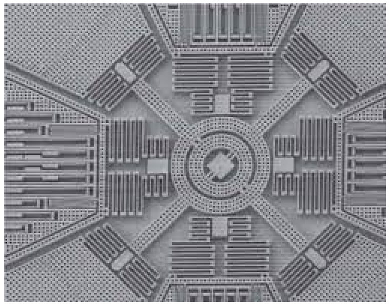
Motion Sensing Using MEMS Accelerometer

Source : Internet

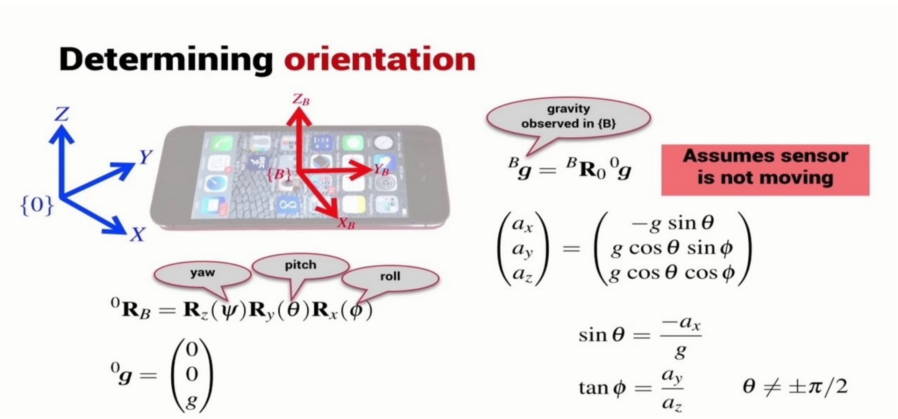
MEMS Gyroscope



Angular Displacement – Electrical Signal



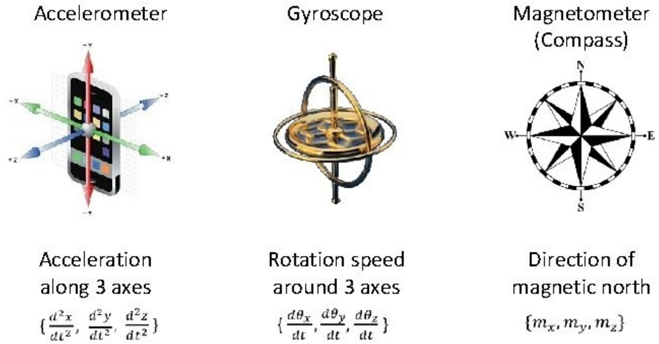
MEMS Gyroscope



Source : Internet

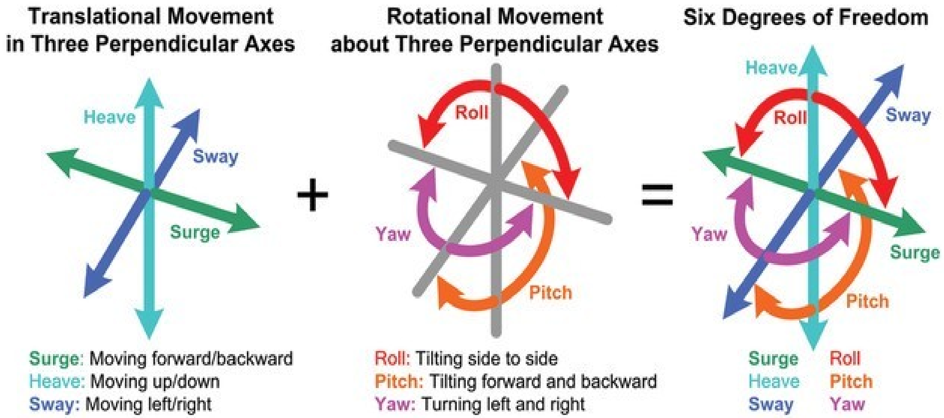
Motion Sensing with Accelerometer and Gyroscope

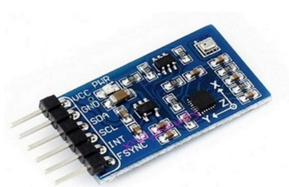
IMU (Inertial Measurement Unit)

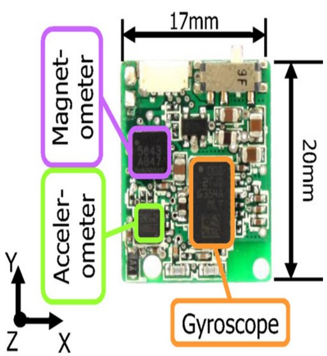


Popular since they are inexpensive, small, and power efficient

Can be embedded inside any object to enable intelligence







IMU unit

MPU – 6050
6-axis IMU
3-axis accelerometers
3 axis gyroscope
Motion processor

MPU – 9250
9-axis IMU
3-axis accelerometers
3 axis gyroscope
3-axis magnetometer
Motion processor

Source : Internet

Some common Sensors

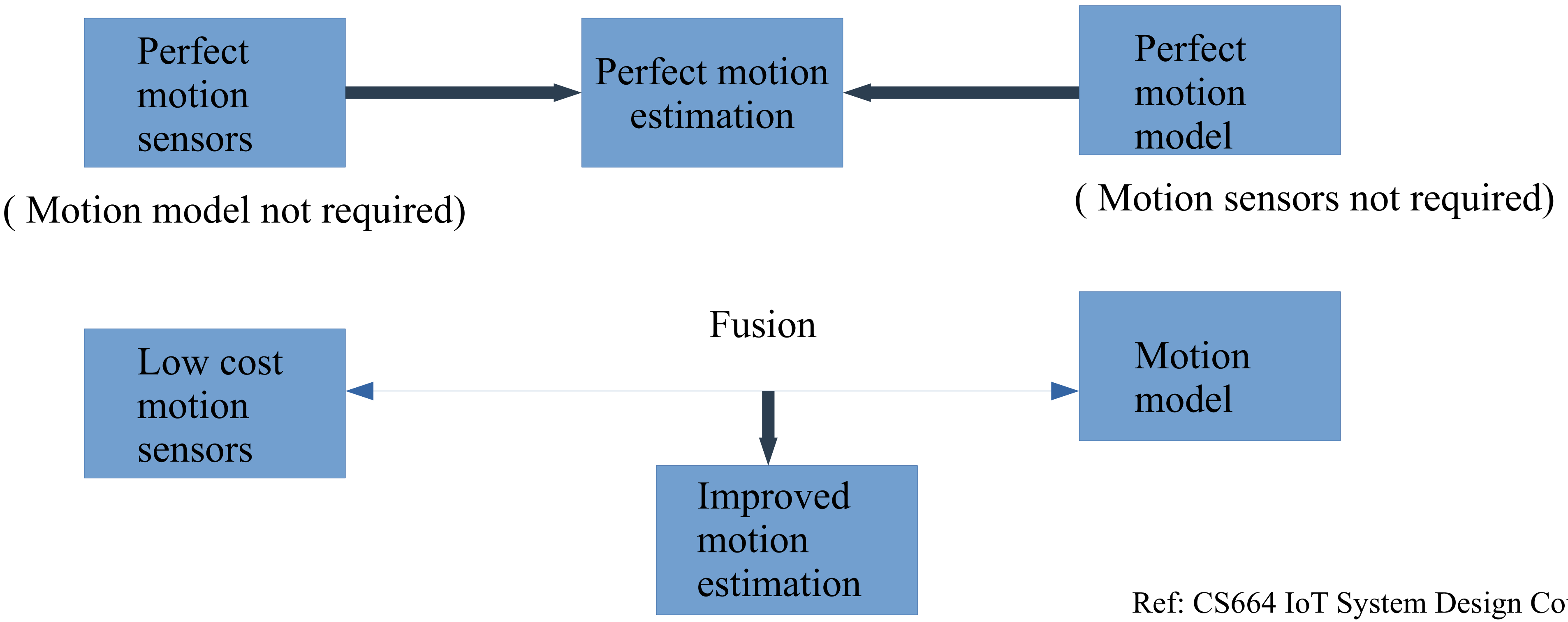
Source : Internet

Working with low cost Sensors

Problem : Noise and Non-aligned response

Solution : Fusing sensors with computation models

Example : Handling low cost motion sensors



Ref: CS664 IoT System Design Course,
Instructor : Prof. Amey Karkare, IIT Kanpur

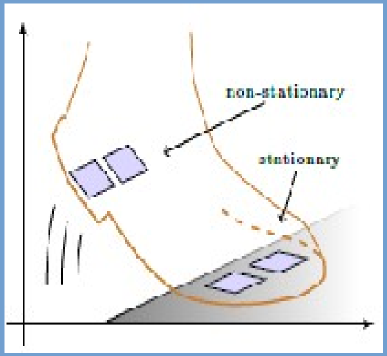
Multi Sensor System

Classical Multi Sensor System

Pressure Sensor Motion Sensor

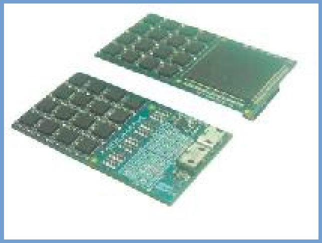
Fusing Data from different Sensors improves performance

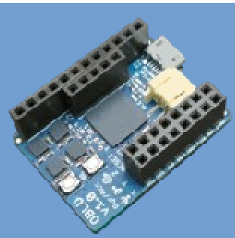
Multi Sensor Joint System



Multiple no-rigidly connect IMUs result in dynamic diversity

Co-located Multi Sensor System

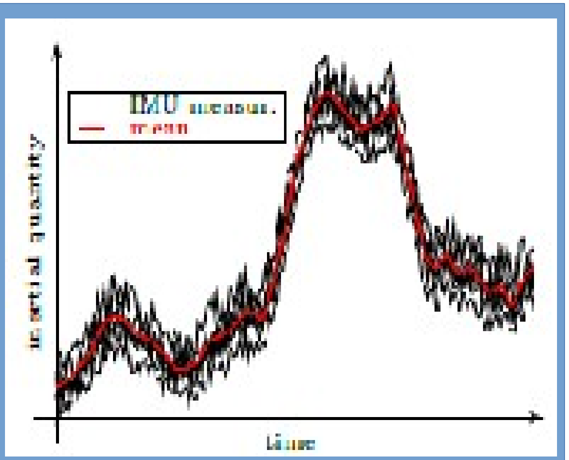




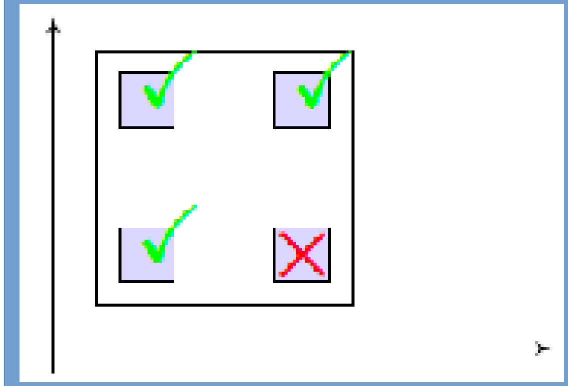
An array of well placed IMUs provides new opportunities for sensing

Ref: CS664 IoT System Design Course,
Instructor : Prof. Amey Karkare, IIT Kanpur

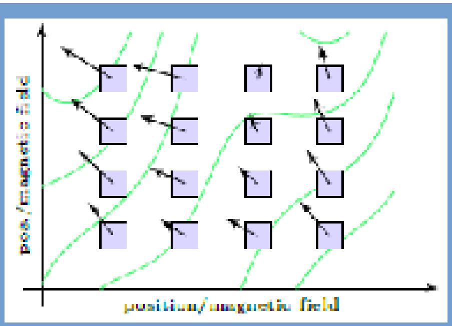
Multi Sensor System : Advantages

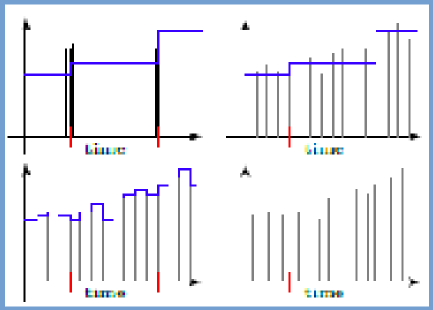


Noise
Averaging



Fault Detection and
Isolation





Spatial and temporal diversity

Ref: CS664 IoT System Design Course,
Instructor : Prof. Amey Karkare, IIT Kanpur

Multi Sensor System

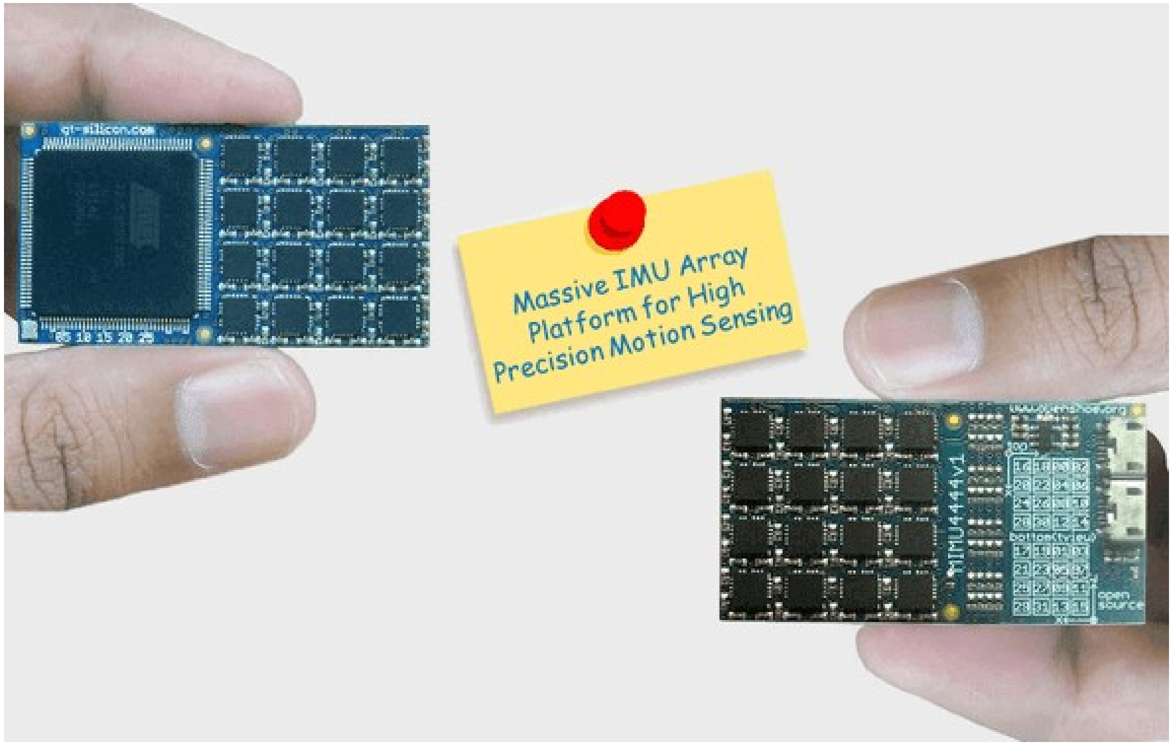
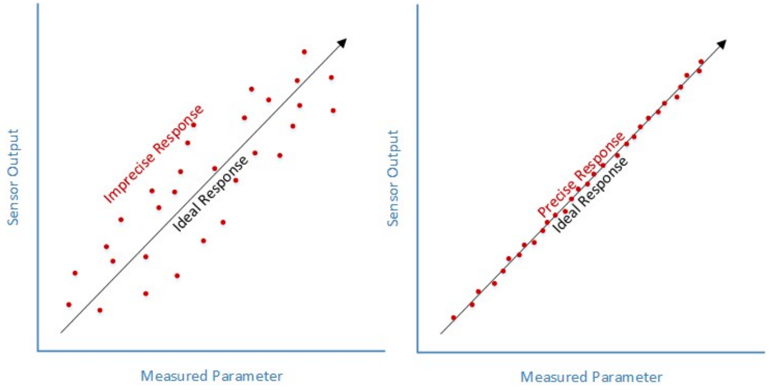


Figure Source : Inertial Elements

Calibration of Sensors

Q: What is sensor calibration?

A: Sensor calibration is an adjustment or set of adjustments performed on a sensor or instrument to make that instrument function as accurately, or error free, as possible.



Q: Why do we need to calibrate sensors ?

A: Randomness in nature gets manifested in the sensors during fabrication and packaging which lead to random errors in sensors

Error in sensor measurement :

i. **Error due to improper zero reference – sensed value drifts over time due to environment or operating conditions after prolonged use**

ii. **Error due to shift in sensor's range**

iii. **Error due to mechanical wear or damage**

Calibration of Sensors

Q: How do we calibrate?

A: Standard References

i. Calibrated sensor – a sensor known to be accurate is used to make reference readings for comparison.

ii. Standard physical reference – reasonable accurate physical standards used as standard reference.

Rangefinders – Rulers or sticks   Temperature sensors – Boiling water is 100°C at sea level;

Accelerometers – Gravity is 1G on surface of earth (z axis accelerometers will sense "g" when placed Horizontally; x and y axis sense "g" when placed vertically. All three axes will measure "g" when placed at certain angle

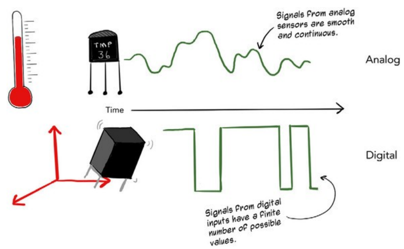
Sensor parameters are compared with any standard reference to find the error.

$$P_{calibrated} = P_{measured} * k + b \; ; \; k = \text{Gain and } b = \text{Bias}$$

The error in any measured parameter can be modelled as gain and bias

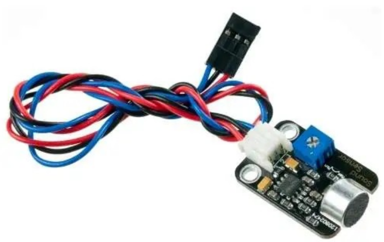
The process of correcting sensor output using gain and bias is called calibration compensation

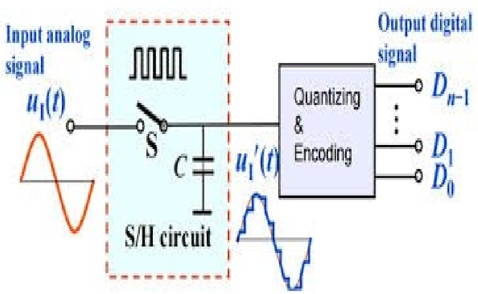
Type of Sensors : Analog and Digital Sensors



• **Digital sensor :** electronic or electrochemical sensor, where data is digitally converted and transmitted. Sensors are often used for analytical measurements, e.g. the measurement of chemical and physical properties of liquids.

• **Analog sensor :** can measure over continuous range, simpler setup, more noise prone, needs an analog to digital converter for computer interpretation





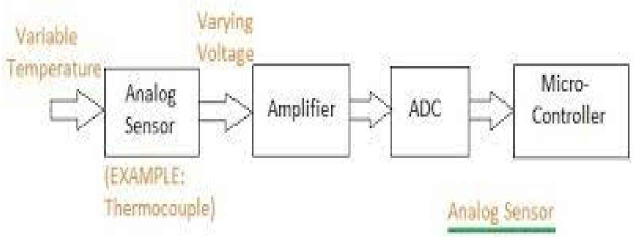
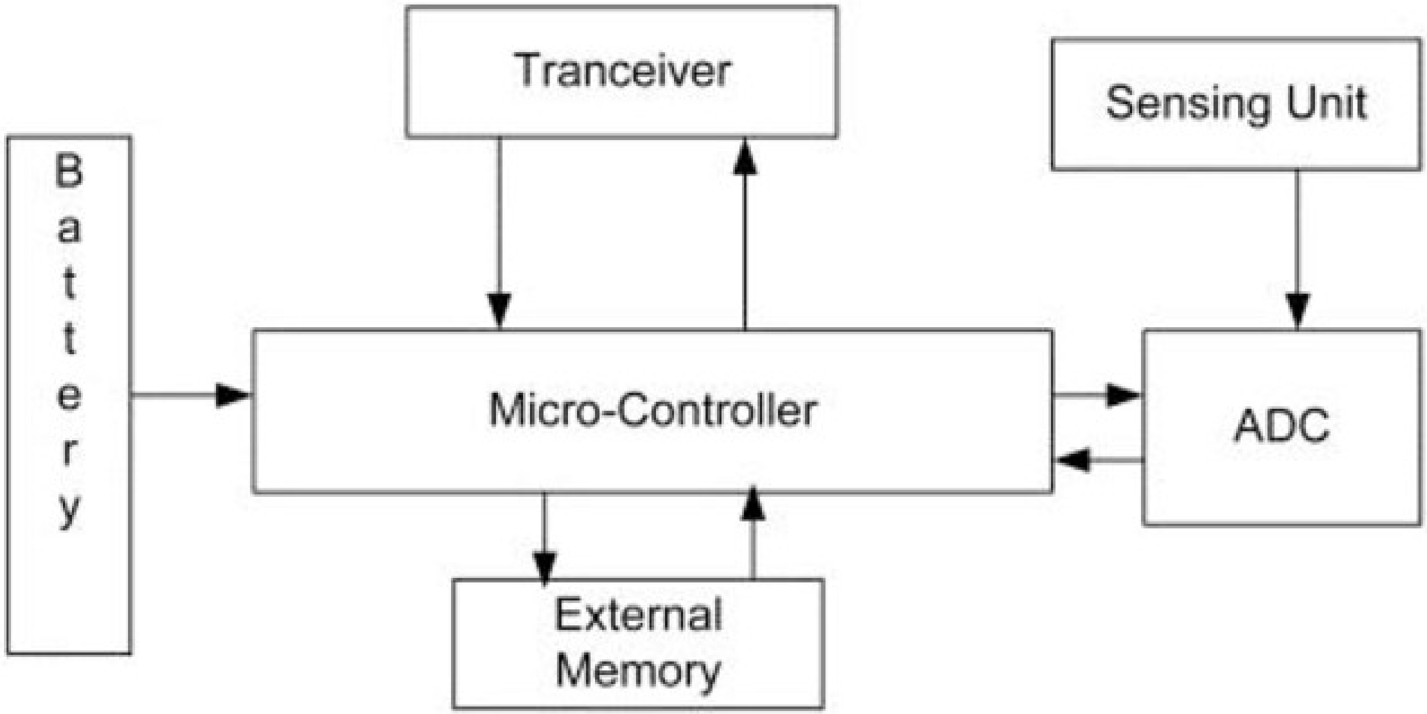


Figure Source : Internet

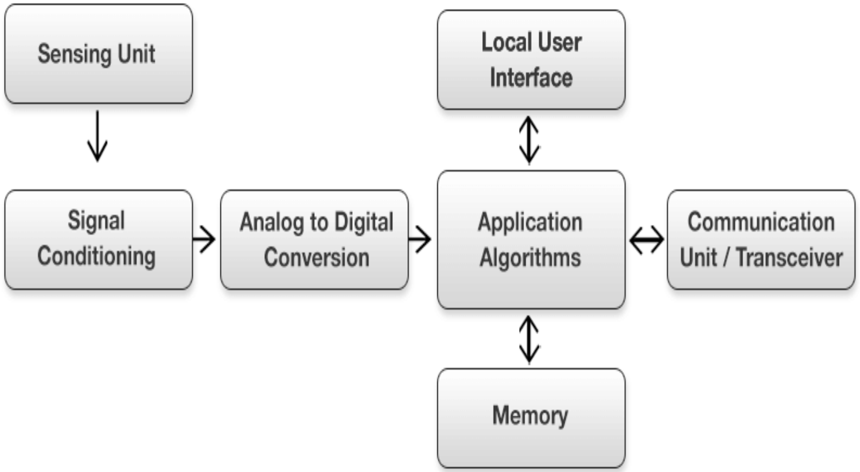
Sensor Node



Tranceiver
Sensing Unit
B
a
t
t
e
r
y
Micro-Controller
ADC
External
Memory

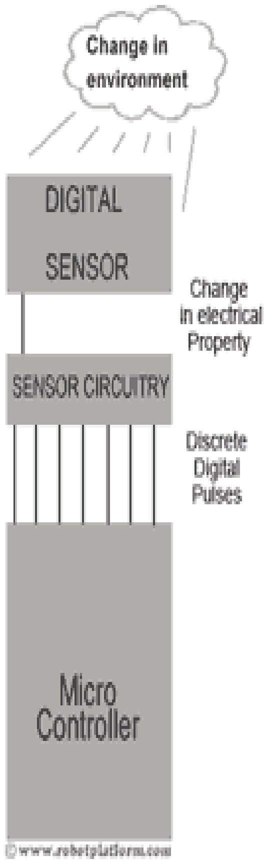
Smart Sensor

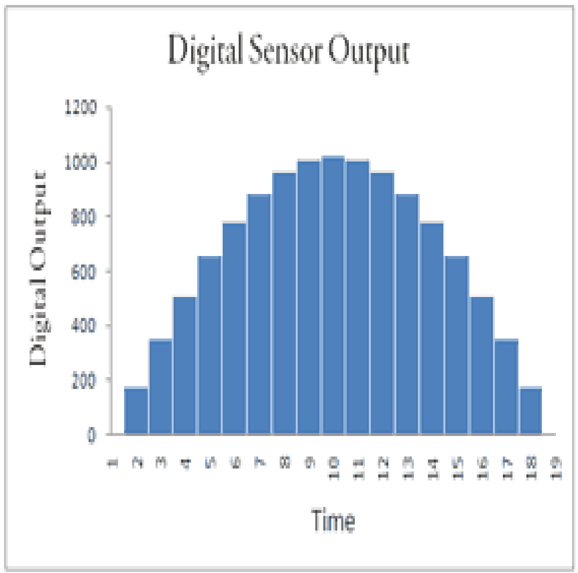
- integrated electronics performing Data conversion, bidirectional communication, decisions and logical operations
- built-in integrated circuit (microcontroller, and sensor) which provides physical parameter as output on connecting it to a supply voltage and programming it.

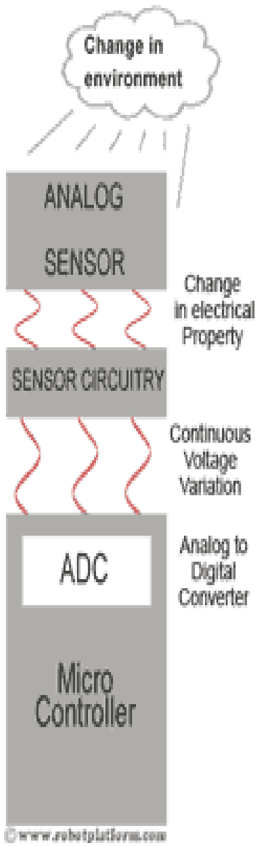


A smart sensor for temperature gives output as hex-digit - 10 UART serial bits according to the degree celsius.
For ex. 01100100 is obtained for 100 degree Celsius considering the sensor has been calibrated

Analog and Digital Sensors







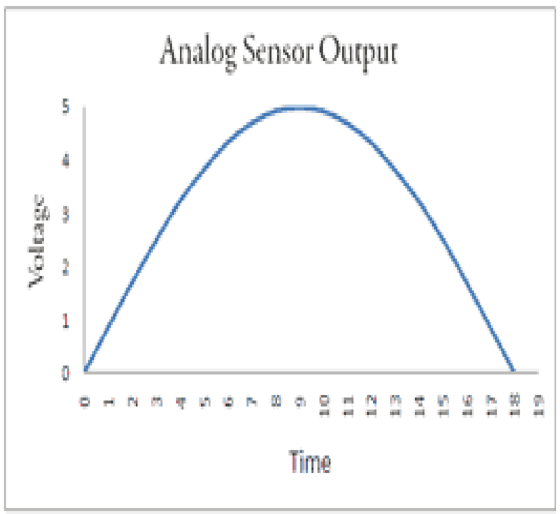


Figure Source : Internet

IoT Sensor Data Acquisition System

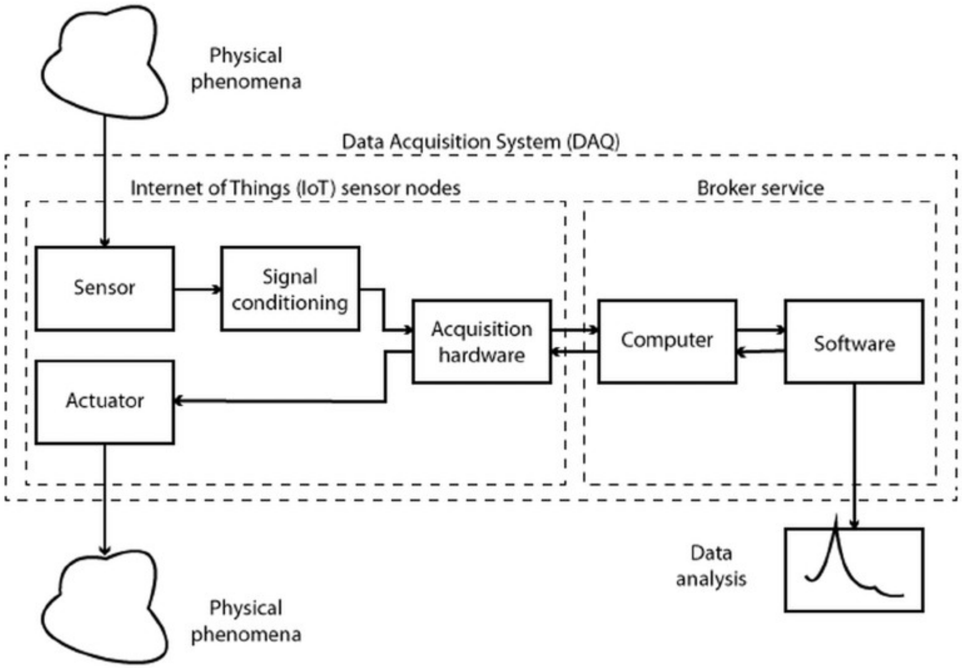


Figure Source : Internet

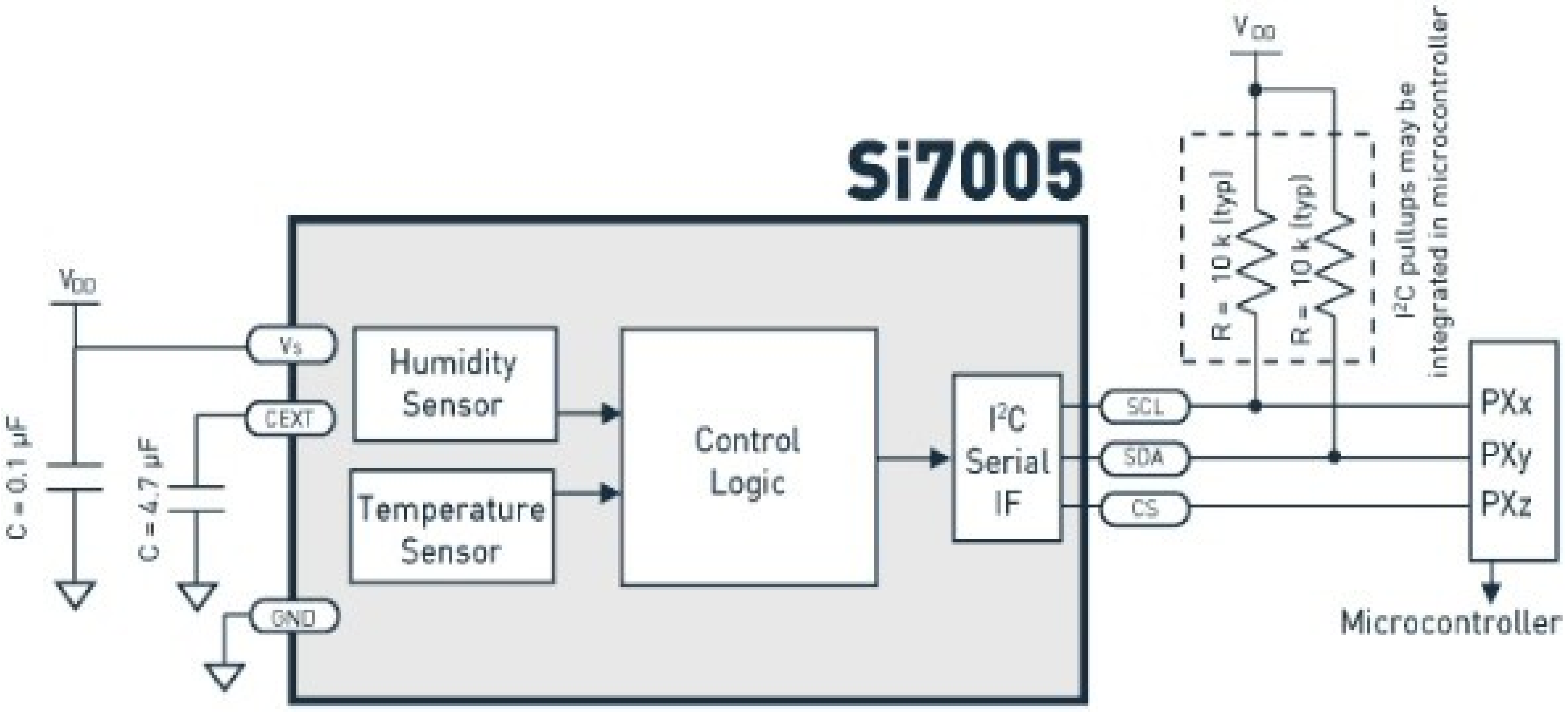
Signal Conditioning + Acquisition hardware = Sensing circuit

Sensing Circuit input receives output of sensor/transducer

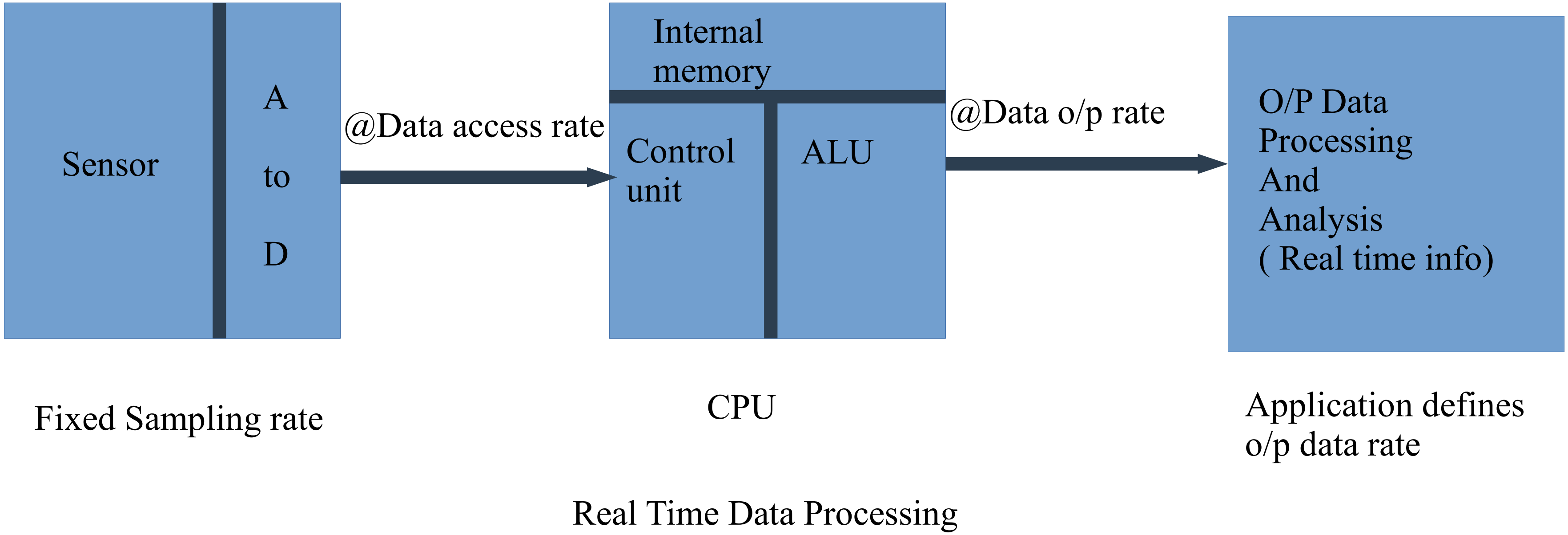
Sensing Circuit output variation is according to the variation in physical condition

Sensing Circuit receives energy in the form of variation in currents, voltages, phase angle or frequencies

Digital Sensor Circuit

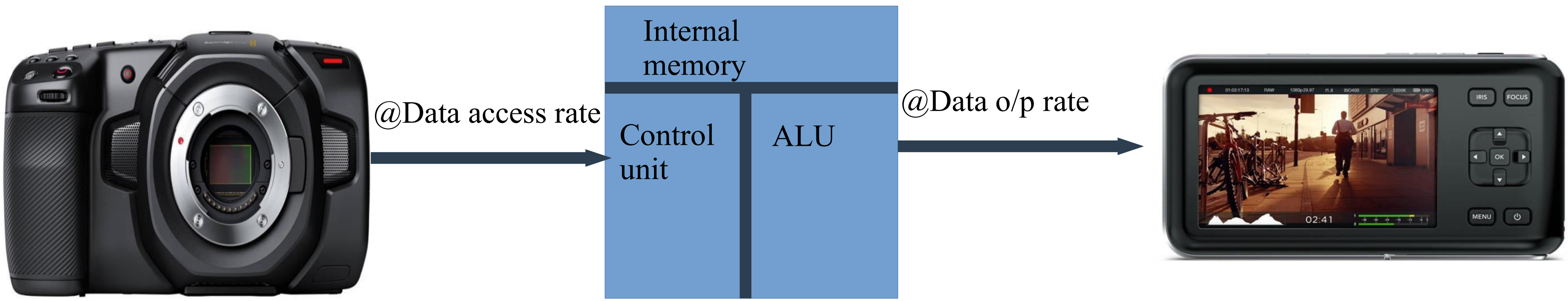


IoT Sensor Node



Ref: CS664 IoT System Design Course,
Instructor : Prof. Amey Karkare, IIT Kanpur

Real Time Sensor Data Processing : Camera Display System



Camera

CPU

Real Time Display

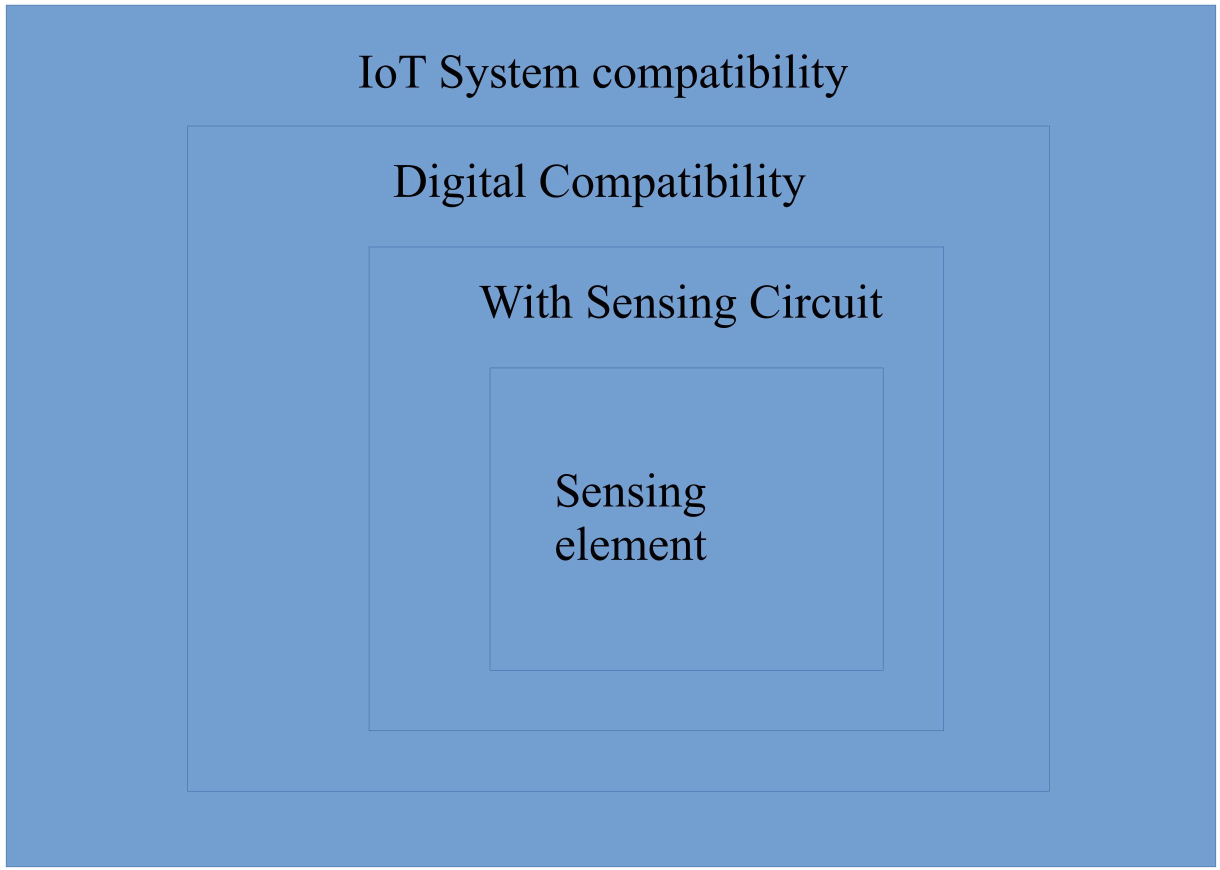
Real Time Data Processing

Ref: CS664 IoT System Design Course,
Instructor : Prof. Amey Karkare, IIT Kanpur

IoT System Component : Sensor Node

IoT System compatibility
- Computation
- Power mgmt. (battery)
- Wireless
- Data comm. Protocols

Digital Compatibility
- Analog to Digital
- data transfer protocols

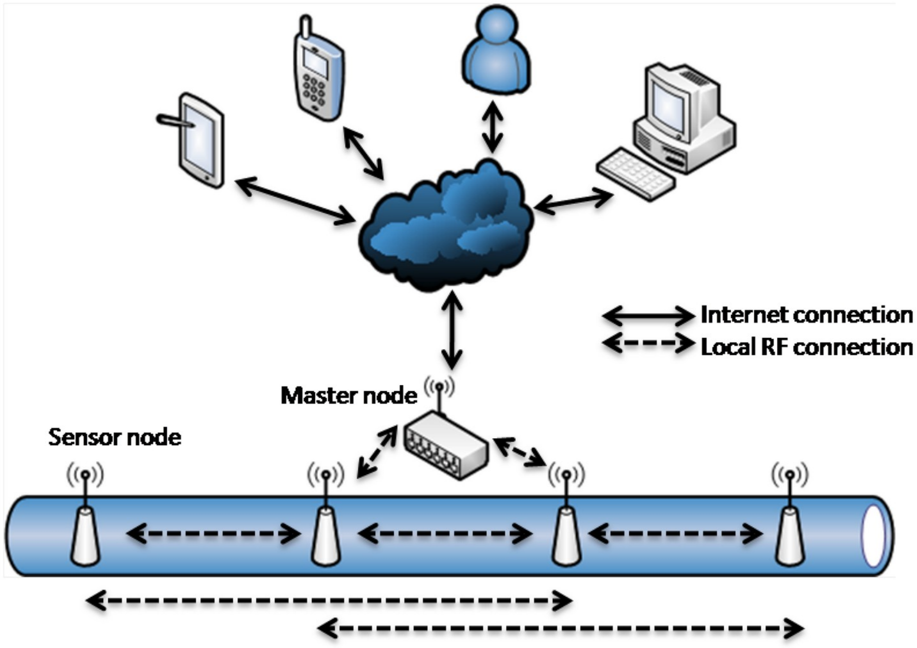


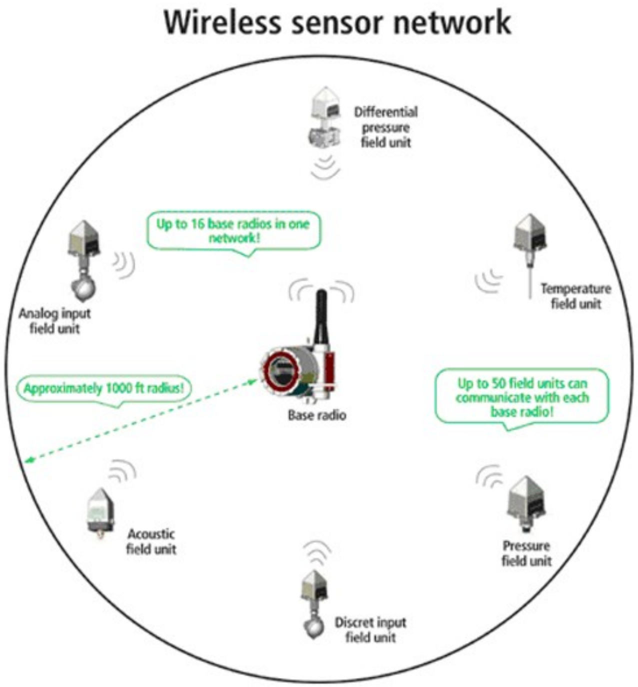
Sensing Circuit
- excite electrically
- amplification

Sensing Element
- responds to physical world

Ref: CS664 IoT System Design Course, Instructor : Prof. Amey Karkare, IIT Kanpur

Wireless Sensor Network (WSN) : Smart Sensor Network





Network of sensor nodes which connect wirelessly
Nodes have capability of computation, data compaction, aggregation and analysis, communication and networking. Each node has independent computing power and capability to send and receive responses, data forward and routing capabilities

Actuator

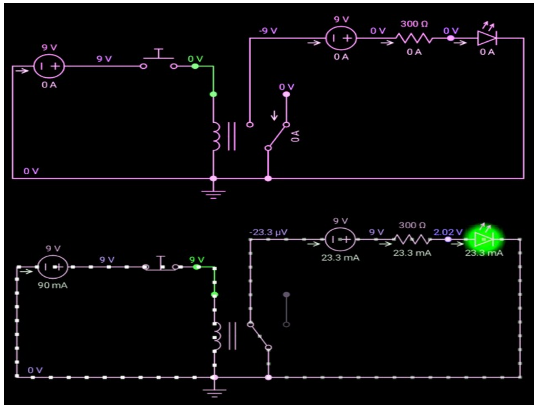
A device that takes the actions as per the input command, pulse, state (1/0), set of 1's and 0's or control signal. It converts electrical energy into physical action

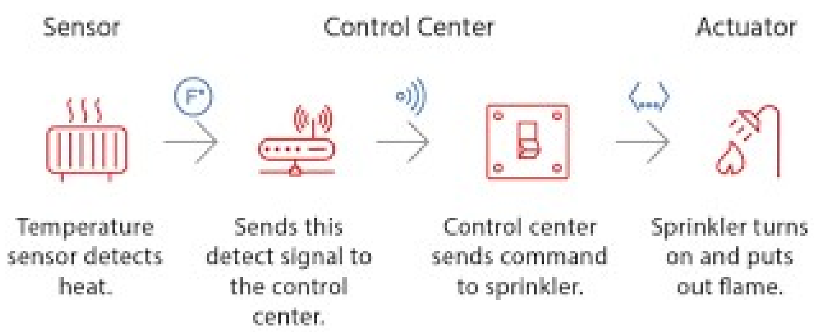
Example :

1. Motor (dc / ac)

2. Piezoelectric vibrator - piezoelectric crystals when applied varying electric voltages at input generate vibrations

3. Relay Switch : An electronic switch can be controlled by 1/0 from the port pin of microcontroller. A relay switch makes mechanical contact when input magnetizes with a control circuit and pulls a lever to make the contact





Power Management

1. Voltage Supply / Power Source
2. Imperfections

3. Power Rating
4. Real Life design challenges

5. Efficiency
6. Voltage regulator

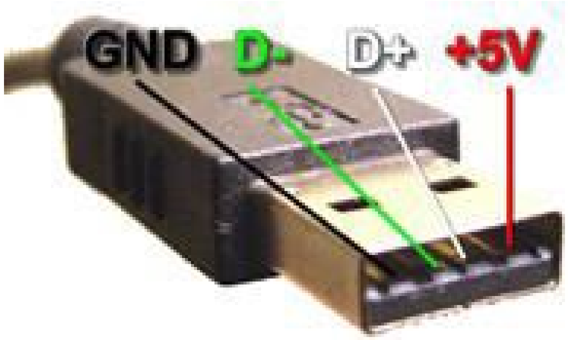
Power handling in Electronic Circuits

Typical power source for electronic gadgets : AC mains, batteries, USB (it a connector)

USB Port as Power Source :

Fixed voltage – 5V DC

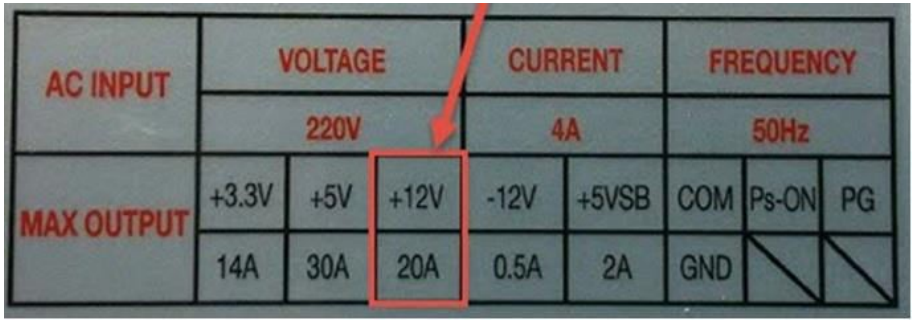
- Maximum current limit of a USB port (which also supports USB data transfer)
  - 100 mA
  - 500 mA
- Cellphone chargers use USB connectors
- Those are just physical connectors for charging only
- There is no USB data transfer support in chargers
- Charger circuit controls the charging current
- Charging current is chosen as per battery specifications



Power handling in Electronic Circuits

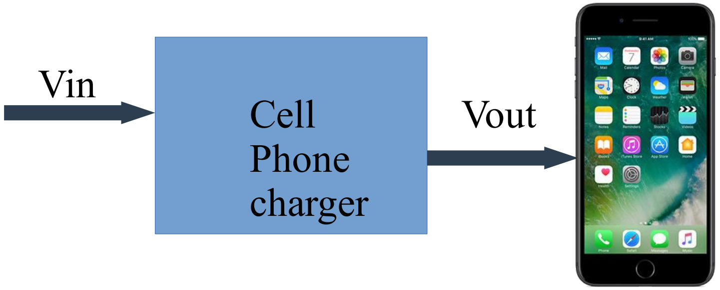
1. AC to Dc conversion (cell phone charger)
2. SMPS – accepts 220V and delivers +5V, +12V etc.

3. Power Rating of SMPS is important



AC INPUT	VOLTAGE			CURRENT		FREQUENCY		
	220V			4A		50Hz		
MAX OUTPUT	+3.3V	+5V	+12V	-12V	+5VSB	COM	Ps-ON	PG
	14A	30A	20A	0.5A	2A	GND		

4. power consumption :



5. Voltage Regulator : provides voltage at certain level (1.8V, 2.5V etc.), maintains output despite Variation in input

Type of batteries

Commonly used rechargeable batteries

– **Li-ion / Li-Poly (Lithium ion / Lithium polymer)**

– **Pb-Acid (Lead Acid)**

– **NiCd (Nickle Cadmium)**

– **NiMH (Nickle Metal Hydride)**

- **Different chemistries, different terminal voltages**

Li-ion / Li-Poly: most popular for portable and wearable IoT

– Highest energy density

– Low maintenance

– Ease of handling

Li-ion battery

Typical terminal voltage of a unit cell (3.7V)

- Battery capacity (milli-Amp-Hours or mAH or C)
- Charging current
  - Recommended C/2 for best performance
  - Charging time with C/2 is ~2 hours
- Fast charging (2C, max limit)
  - Typical is 2C
  - Max limit of charge current
  - Must not be used on regular basis

Cylindrical

Prismatic

Coin cell

Flexible Li-ion battery for wearables

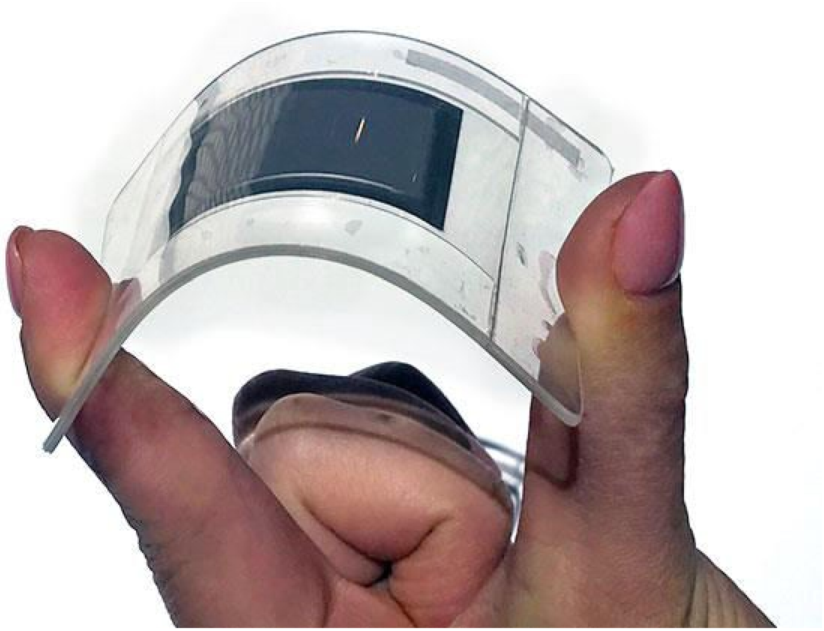


Image source- http://spectrum.ieee.org/tech-talk/consumer-electronics/portable-devices/ces-2017-panasonic-shows-off-bendable-lithiumion-battery-for-iot-wearables

Li-ion battery Unpacked

Electrodes

Insulator

Image source- internet

Need for Computing

1. Sensors convert real world entities to electrical signals

2. Real world generates raw data that has to be interpreted to extract meaningful information

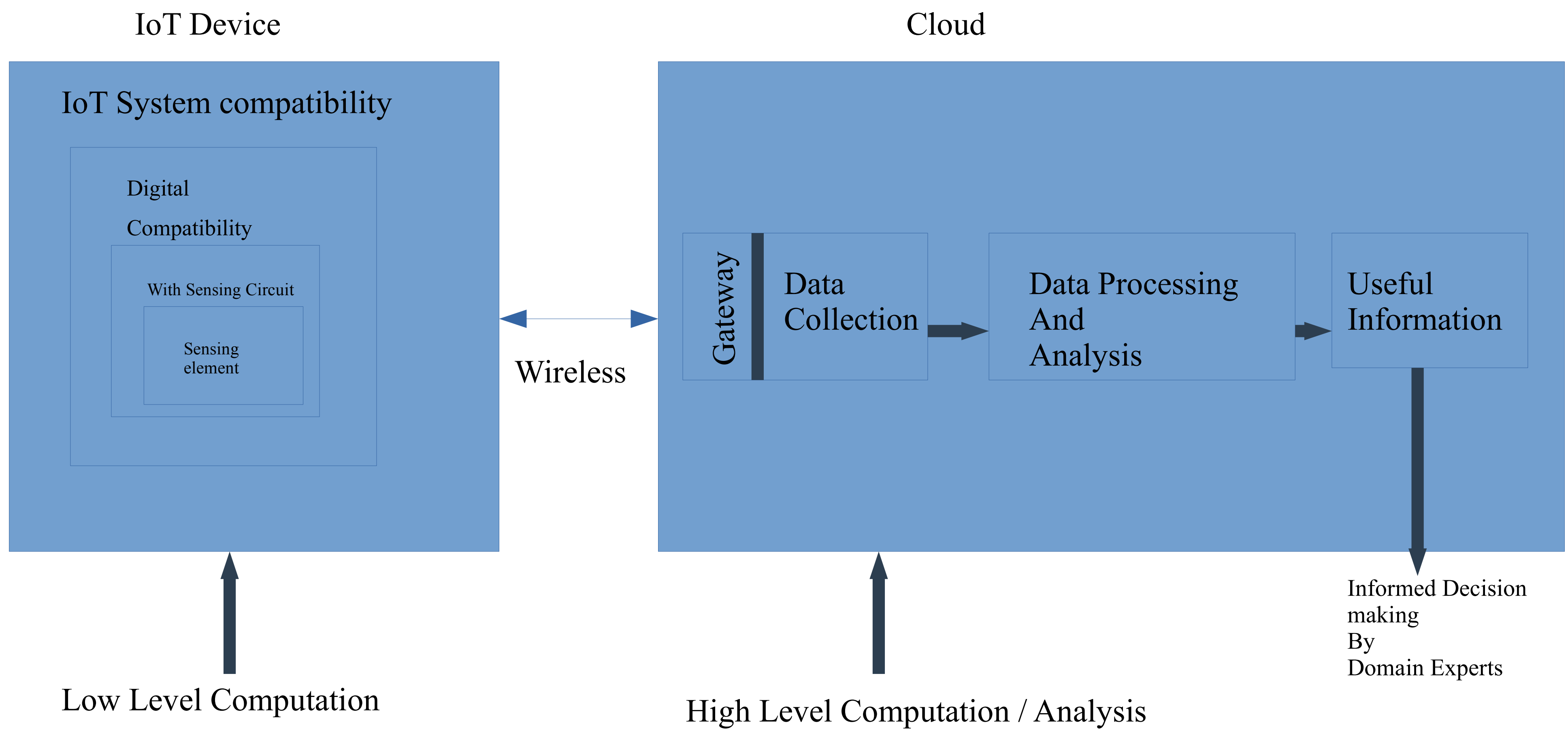
3. There is a strong need to store, transport and sort data

4. There is a strong need to process data and extract information

5. Right information means informed decision making

6. Informed decision means better life !!!

IoT System : Distribution of Computation

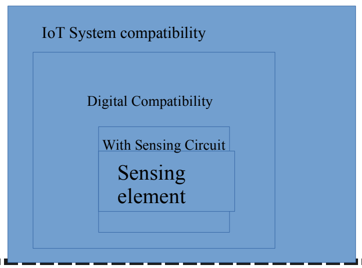


Desirable
- Increased capabilities at local node
- Reduced requirements of connectivity
- Providing backend with high level information
- Simple data interface

Ref: CS664 IoT System Design Course,
Instructor : Prof. Amey Karkare, IIT Kanpur

Types of IoT System : Distribution of Computation

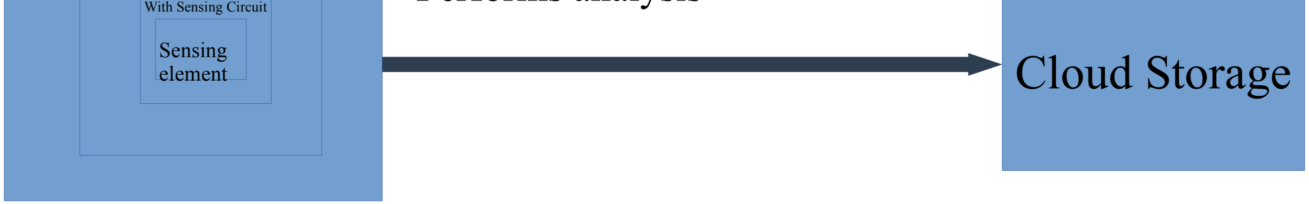
IoT Level 1



Monitoring node, Performs storage and analysis

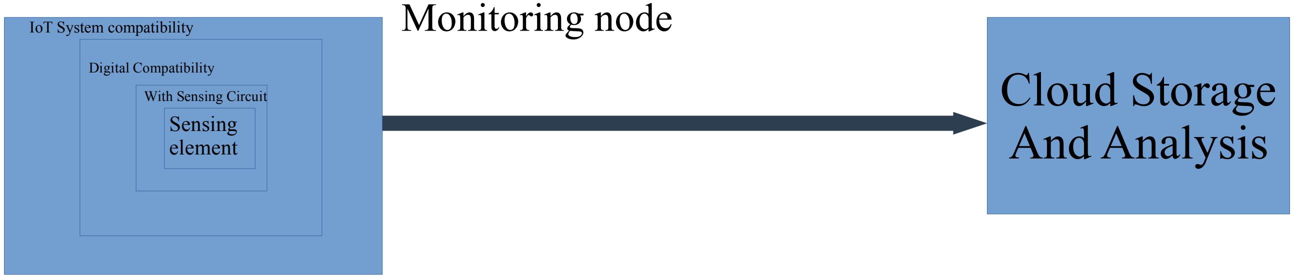
IoT Level 2

Monitoring node, Performs analysis

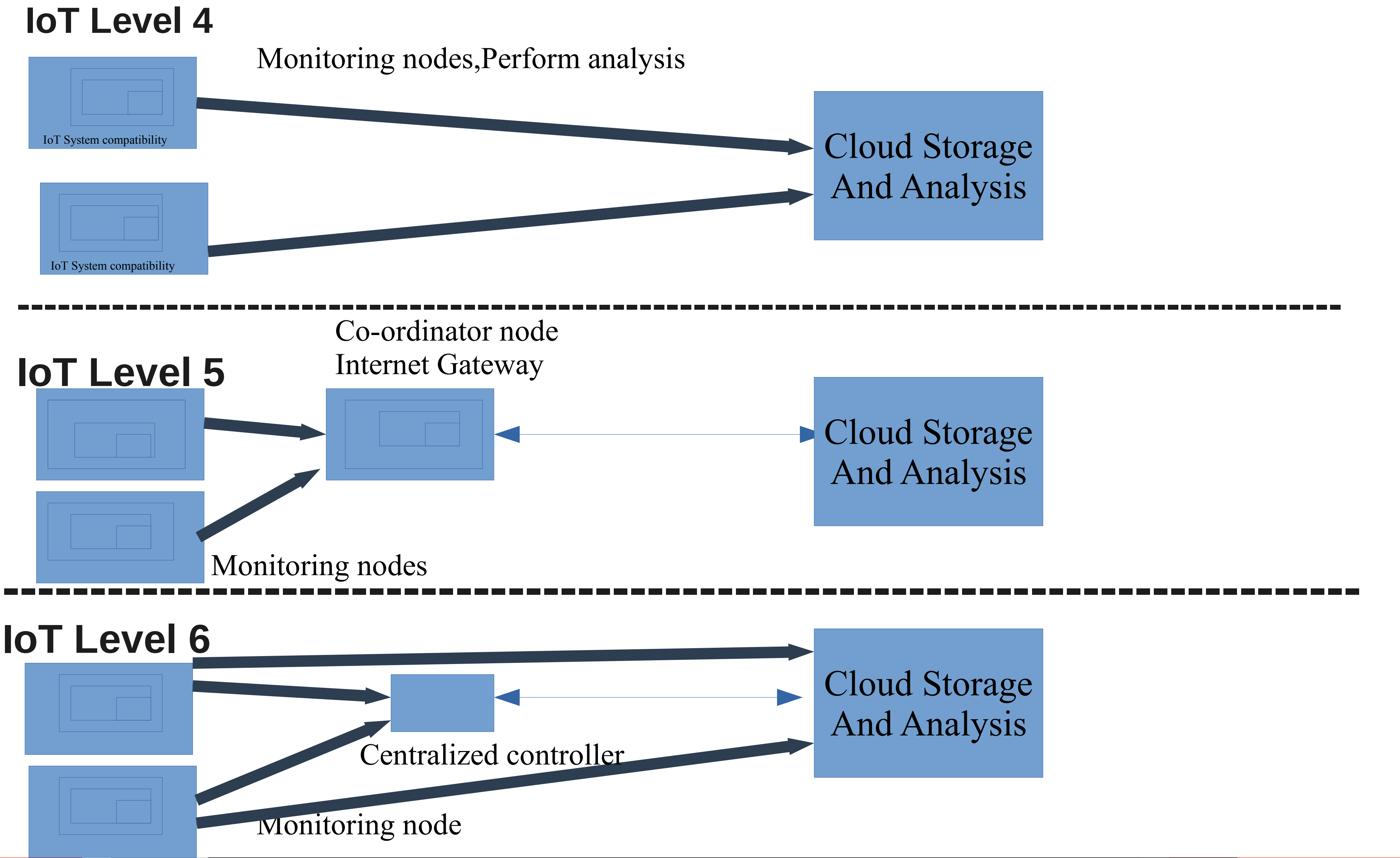


IoT Level 3

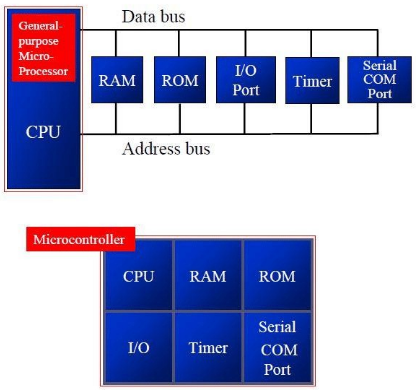
Monitoring node



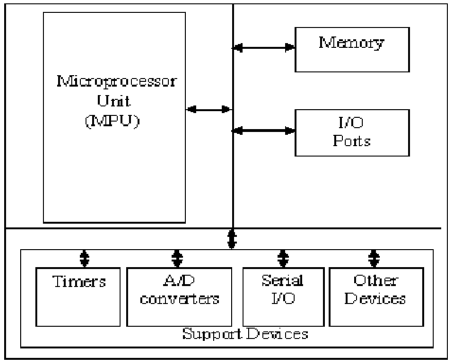
Types of IoT System : Distribution of Computation



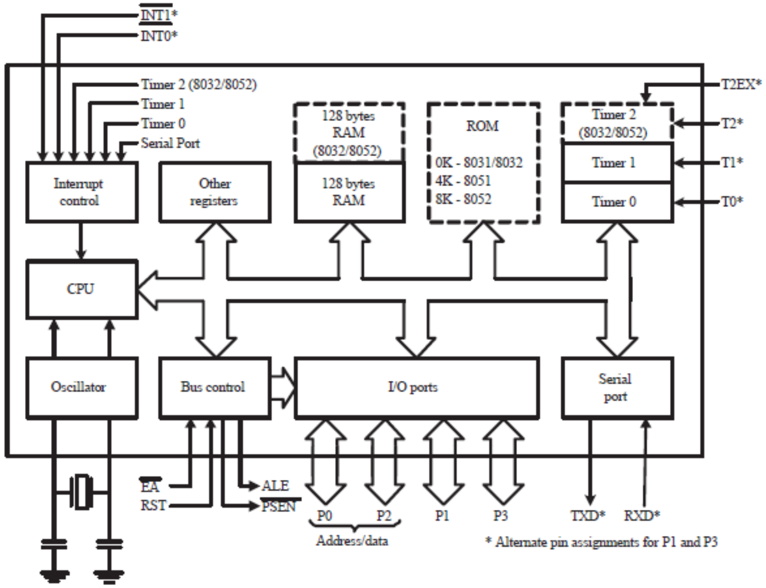
Microcontroller



Microprocessor Vs. Microcontroller



Block Diagram of Microcontroller showing its components



Block Diagram of 8051 Microcontroller

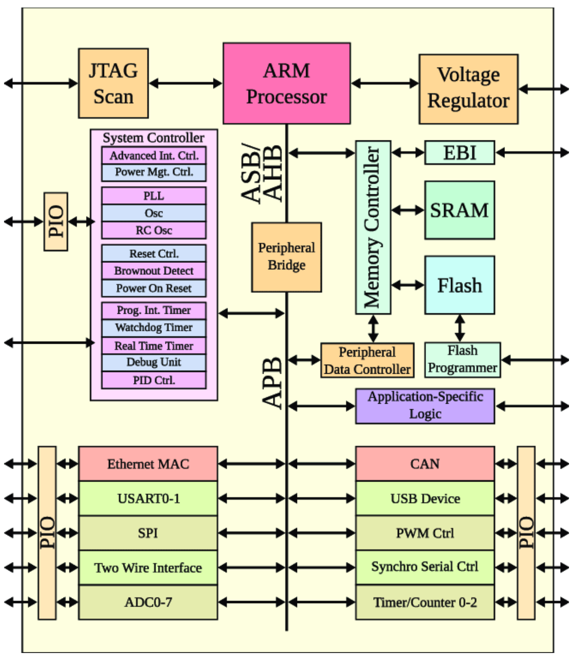
System-on-Chip

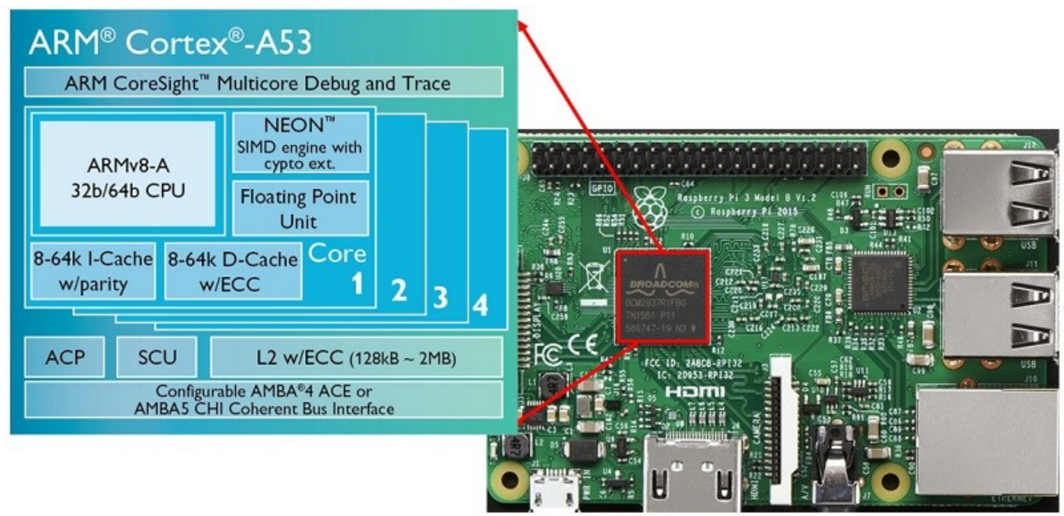
- A system on a chip is an integrated circuit that integrates multiple processors, hardware units, analog circuits and the embedded software

- Microcontroller unit with SD card for embedded software and OS software that enables use of the chip distinctly for a particular purpose

- Example : ARM Cortex, ATmega328

- Microcontroller components : Processor, Internal RAM, Internal Flash and Firmware, Timers, Programmable I/O Ports, General purpose I/Os, Serial I/O Ports, PWM, ADC, Communication Network Interfaces

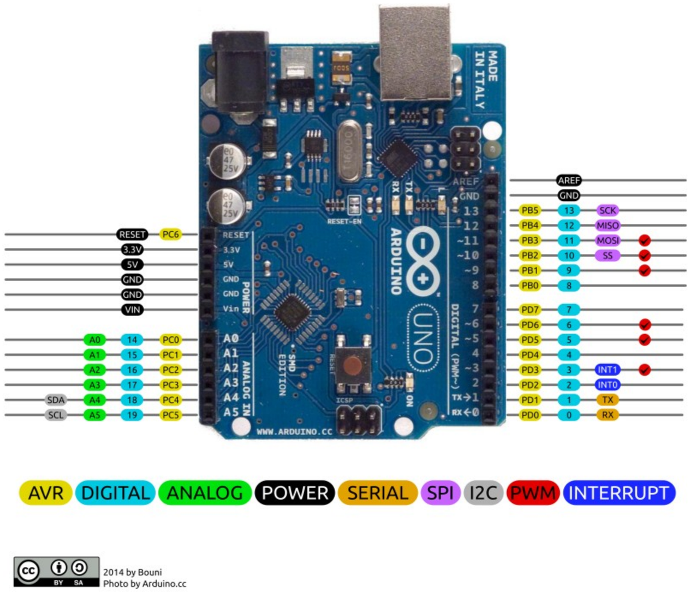


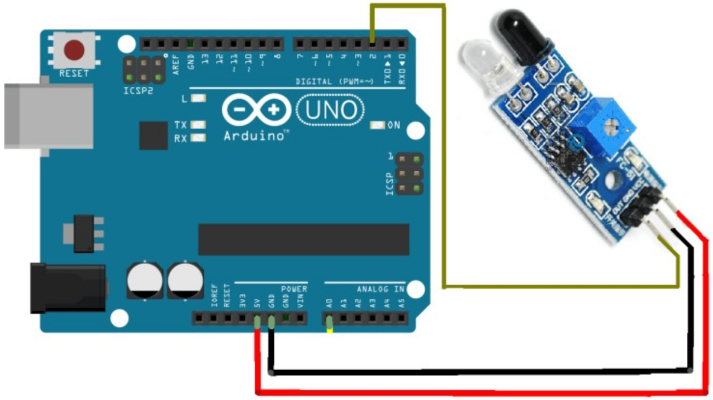


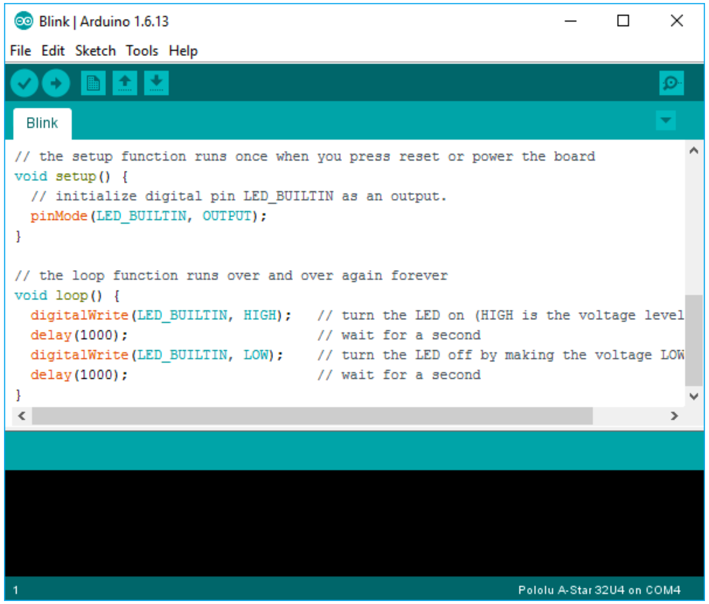
Raspberry Pi Board hosting the ARM Cortex A-53 System-on-Chip

Common IoT Computing Platforms : Arduino

Arduino Uno SMD Pinout

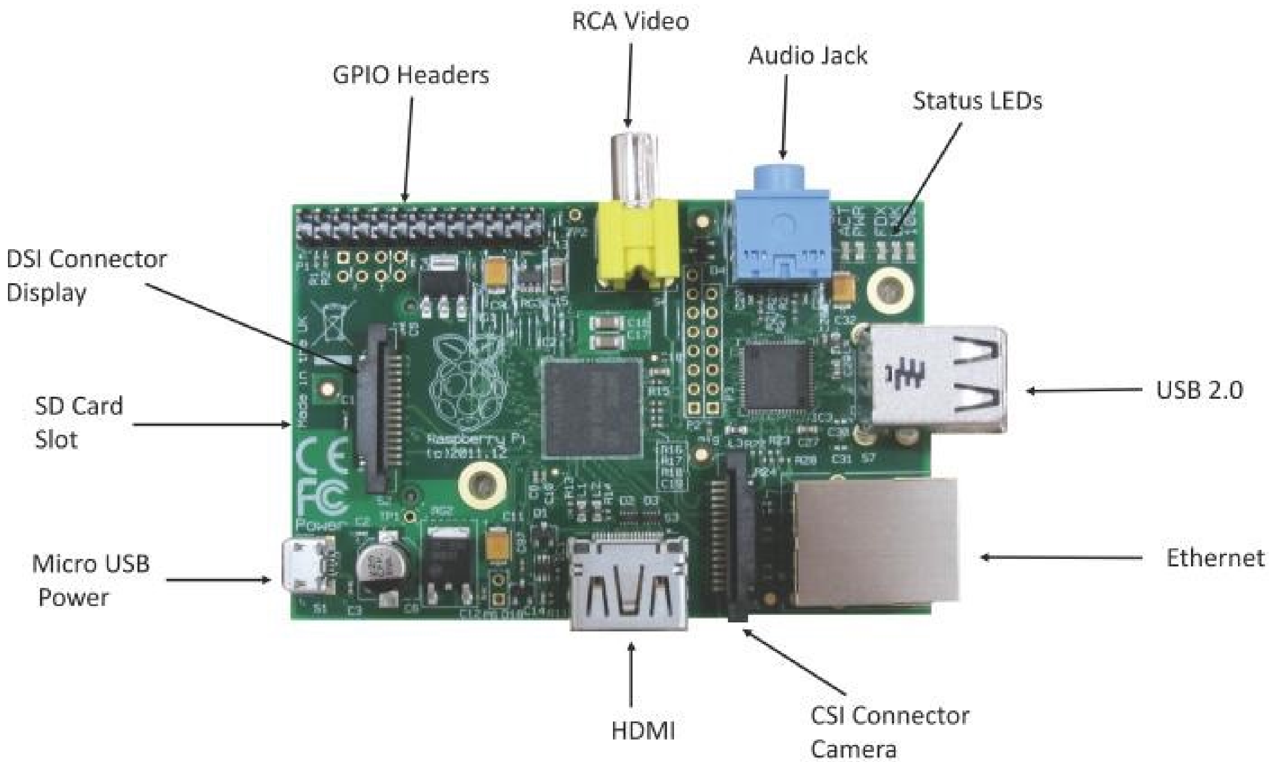




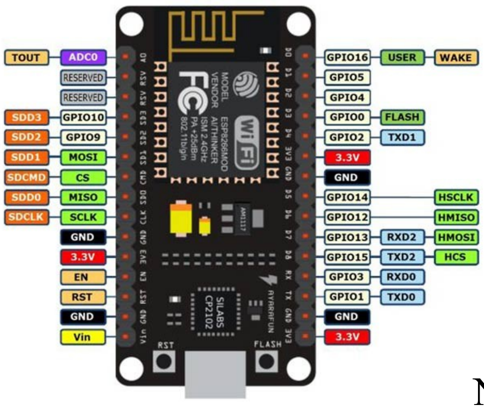


Common IoT Computinng Platforms : Raspberry Pi

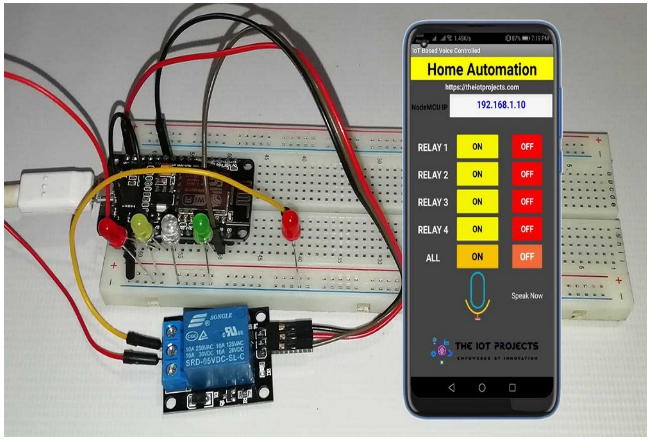
Low cost mini computer, allows interfacing sensors though GPIOs, runs Raspbian OS (a Liux variant), supports Python



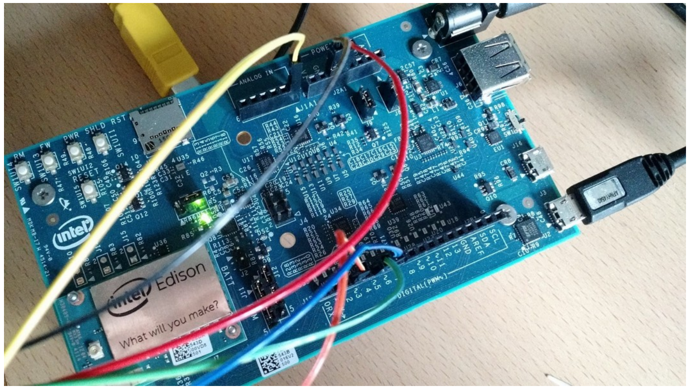
Other IoT Computing Platforms



Node MCU



Beagle board



Intel Edison Board

Interfacing Sensors to Microcontrollers

The process of connecting devices together so that they can exchange is called interfacing

In order for these devices to swap their information, they must share a common communication protocol.

Communication protocols are of two types :

Parallel - multi line channel with each line capable of transmitting several bits of data simultaneously.
usually require buses of data transmitting across eight, sixteen, or more wires
data is transferred as streams of 0's and 1's

Serial - stream their data, one single bit at a time.
operate as little as one wire, usually never more than four, Simple wiring,
serial interface cables can be longer than parallel interface cables since less crosstalk among conductors

Most hardware interfaces are serial interfaces sacrificing potential speed in parallel.

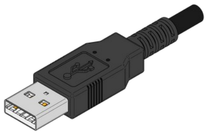
Serial interfaces generally use multiple wires to control the flow and timing of binary information along the primary data wire.

Each type of hardware interface defines a method of communicating between a peripheral and the central processor

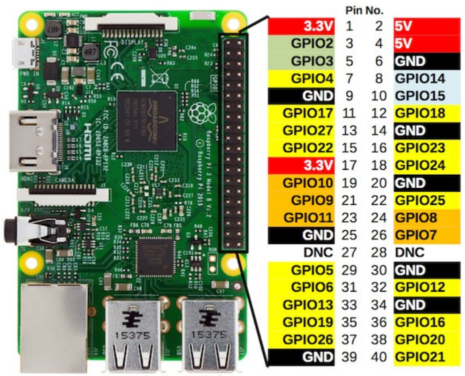
Interfacing Sensors to Microcontrollers

IoT hardware platforms use a number of common interfaces. Sensor and actuator modules can support one or more of these interfaces:

1. Universal Serial Bus (USB) - a technology that allows a person to connect an electronic device to a microcontroller. It is a fast serial bus.

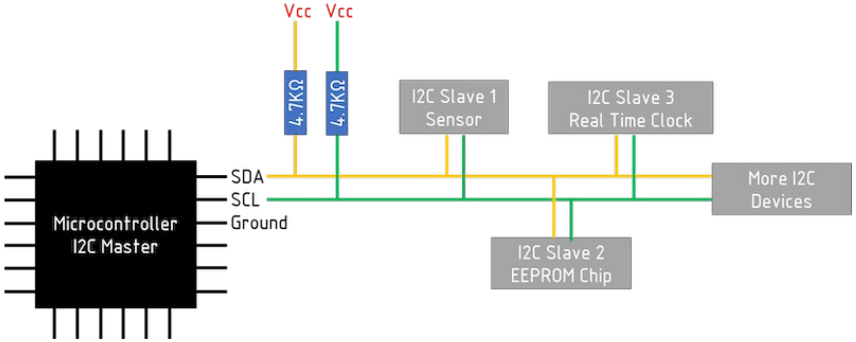


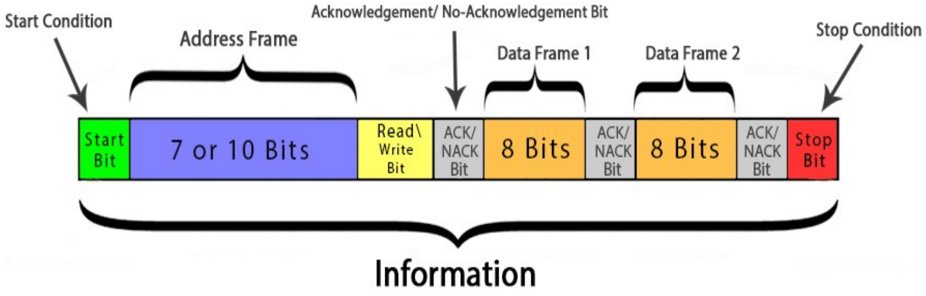
2. General-purpose input/output pins (GPIO) - generic pin on an integrated circuit or computer board whose behaviour (whether it is an input or output pin) is controllable by the user at run time. GPIO pins have no predefined purpose, and go unused by default. GPIO pins can be designed to carry digital or analog signals, and digital pins have only two states: HIGH or LOW.



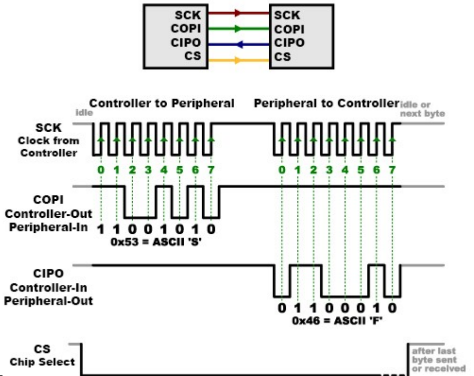
Interfacing Sensors to Microcontrollers

3. Inter-Integrated Circuit serial bus (I2C) - uses a protocol that enables multiple modules to be assigned a discrete address on the bus. I2C is sometimes pronounced “I two C”, “I-I-C”, or “I squared C”. I has two wires, a clock and data wire.

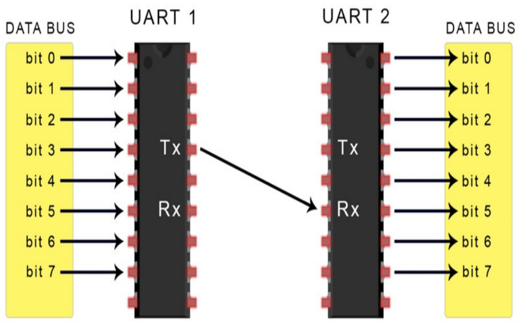


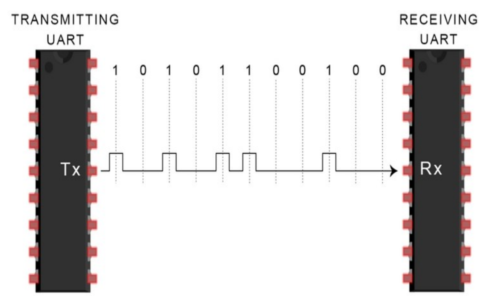


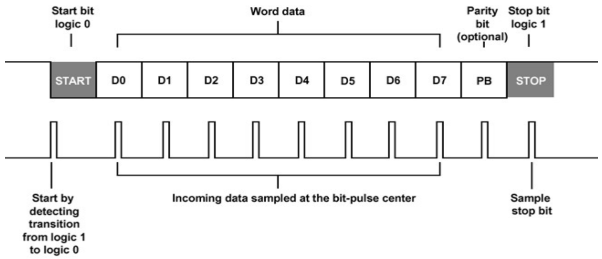
4. Serial Peripheral Interface/Interchange (SPI) - Bus devices employ a master-slave architecture, with a single master and full-duplex communication.



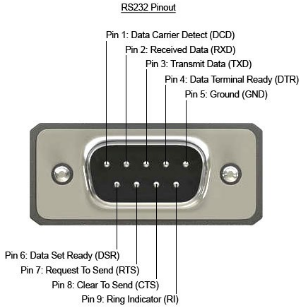
5. Universal Asynchronous Receiver/Transmitter (UART) - it is not a communication protocol like SPI and I2C, but a physical circuit in a microcontroller, or a stand-alone IC**.** devices translate data between serial and parallel forms at the point where the data is acted on by the processor. UART is required when serial data must be laid out in memory in a parallel fashion.

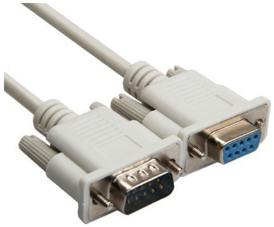


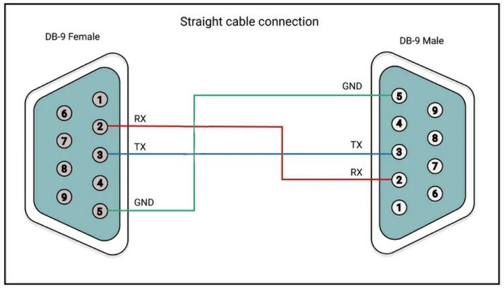




6. Recommended Standard 232(RS232) - is used for obtaining communication between the computer and circuit in order to transfer data

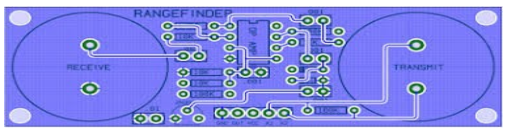


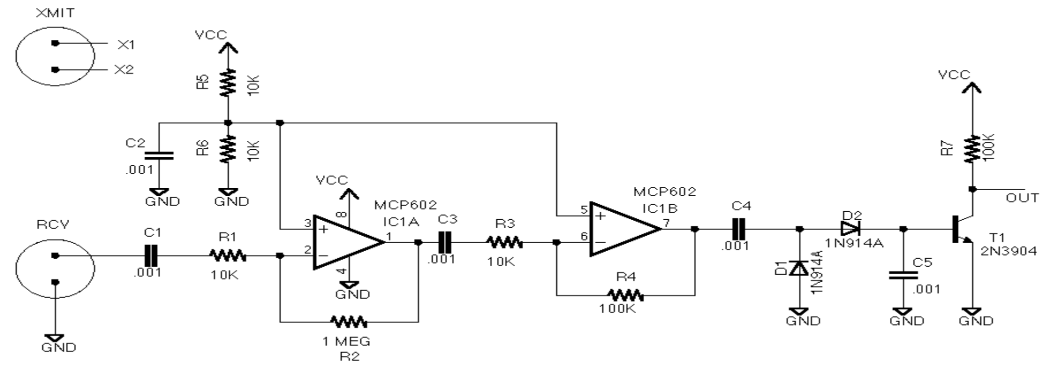


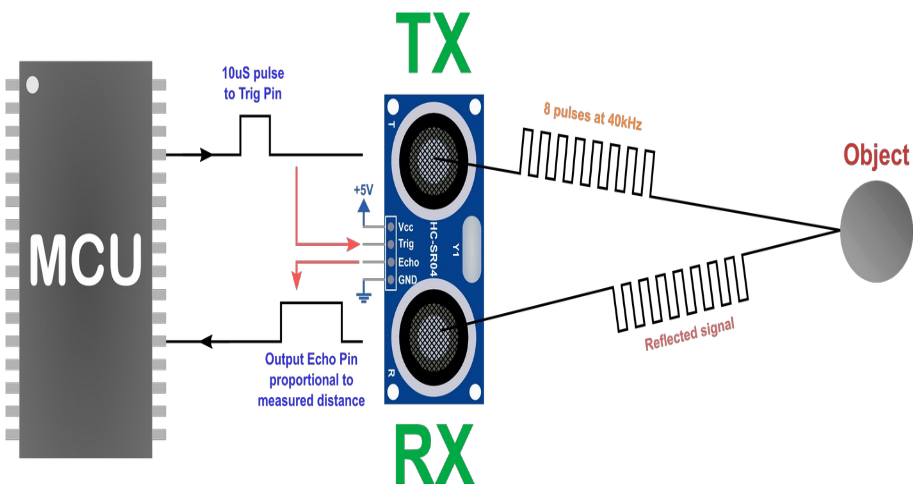


- **Interfacing Sonar Range finder to MCU**

Only two inputs are required : INIT : Start transmitting; ECHO – Return signal

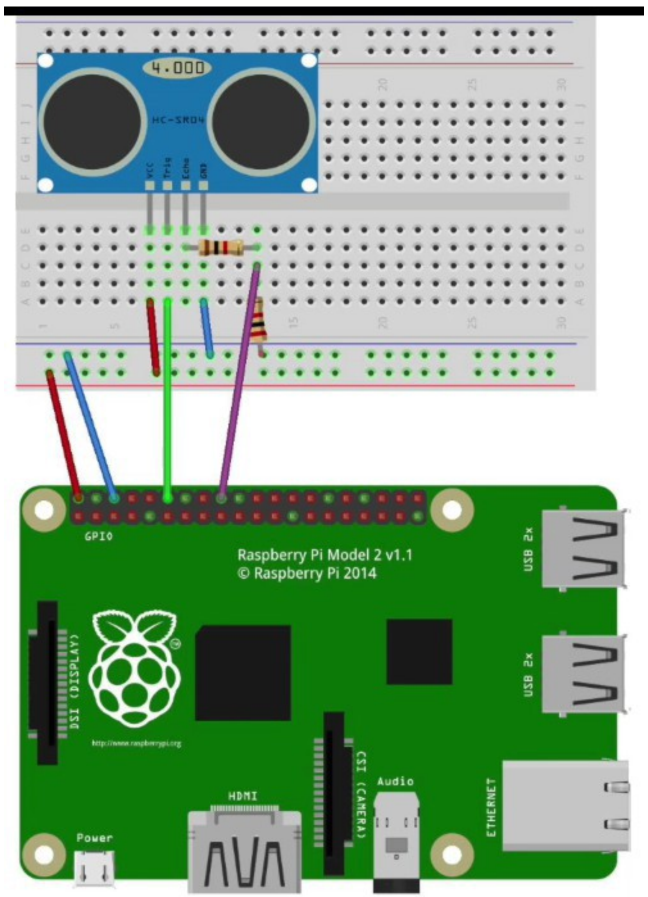







```
// Pseudo-code  showing the interfacing technique
   // Setup
          Set the Trig pin as output
          Set the Echo pin as input
          Set the baud rate for serial communication
  // Execution
Loop()
{
        //Sets the trigPin on HIGH state for 10 micro seconds
            Write Trig pin  = ‘ 1’ ; delay 10microsec; Write Trig pin = ‘ 0’ ;
        // Reads the echoPin, returns the sound wave travel time in microseconds

            Read the echo pin and record the duration of the pulse received;

        // calculate the distance from the duration of the echo pulse and print

            Distance  =  (duration of the echo pulse * 34300 ) / 2;
            Print Distance on the serial monitor;
            Insert a suitable delay;
 }
```

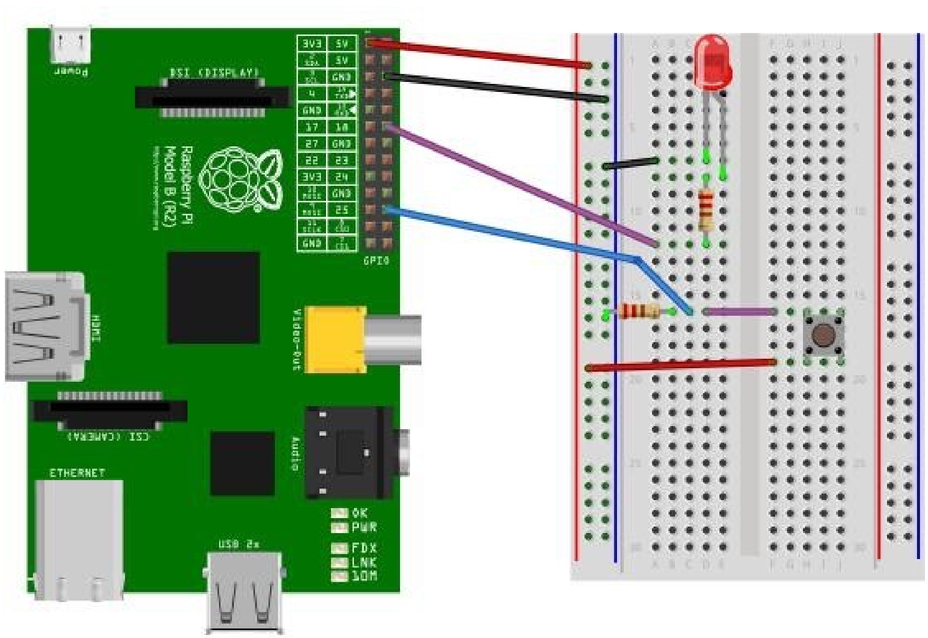
Interface Sonar Range Finder with Raspberry Pi




```
import RPi.GPIO as GPIO
import time, signal, sys
GPIO.setmode(GPIO.BCM)
pinTrigger = 18
pinEcho = 24

def close(signal, frame):
        print("\nTurning off ultrasonic detection...\n")
        GPIO.cleanup()
        sys.exit(0)

signal.signal(signal.SIGINT, close)
GPIO.setup(pinTrigger, GPIO.OUT)
GPIO.setup(pinEcho, GPIO.IN)

while True:
        # set Trigger to HIGH
        GPIO.output(pinTrigger, True)
        # set Trigger after 0.01ms to LOW
        time.sleep(0.00001)
```

```
GPIO.output(pinTrigger, False)

startTime = time.time()
stopTime = time.time()

# save start time
while 0 == GPIO.input(pinEcho):
        startTime = time.time()

# save time of arrival
while 1 == GPIO.input(pinEcho):
        stopTime = time.time()

TimeElapsed = stopTime - startTime
distance = (TimeElapsed * 34300) / 2

print ("Distance: %.1f cm" % distance)
time.sleep(1)
```

Interfacing devices to 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()
```