JECRC Foundation

JAIPUR ENGINEERING COLLEGE AND RESEARCH CENTRE

JAIPUR ENGINEERING COLLEGE AND RESEARCH CENTRE

Year & Sem – IV year & VIII Sem
Subject – Internet of Things & 8IT4-01
Unit – II

UNIT-II

IoT Hardware and Software

Vision

To establish outcome based excellence in teaching, learning and commitment to support IT Industry.

Mission

To provide outcome based education.

To provide fundamental & Intellectual knowledge with essential skills to meet current and future need of IT Industry across the globe.

To inculcate the philosophy of continuous learning, ethical values & Social Responsibility.

Course outcomes (CO)

CO1: Understand the revolution of internet in field of cloud, wireless network, embedded system and mobile devices.
CO2: Apply IOT design concepts in various dimensions implementing software and hardware.
CO3: Analyze various M2M and IOT architectures.
CO4: Design and develop various applications in IOT.

Mapping of CO & PO

Subject	Code	L/T/P	CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
IOT	8IT4-01	L	Understand the revolution of internet in field of cloud, wireless network, embedded system and mobile devices.	H	H	M	M	M	M	-	--	M	M	M	H	H	H
		L	Apply IOT design concepts in various dimensions implementing software and hardware.	H	H	H	H	H	M	-	--	L	M	M	H	H	H
		L	Analyze various M2M and IoT architectures.	H	M	M	L	M	M	-	L	M	L	L	H	M	M
		L	Design and develop various applications in IOT.	H	M	H	H	H	M	M	L	H	H	H	H	H	H

IV Year- VII & VIII Semester: B. Tech. (Information Technology)

8IT4-01: Internet of Things

Credit: 3 **Max. Marks: 150(IA:30, ETE:120)**

3L+0T+0P **End Term Exam: 3 Hours**

SN	Contents	Hours
1	**Introduction:** Objective, scope and outcome of the course.	01
2	**Introduction to IoT:** Definition and characteristics of IoT, Design of IOT: Physical design of IOT, Logical Design of IOT- Functional Blocks, communication models, communication APIs, IOT enabling Technologies- Wireless Sensor Networks, Cloud computing, big data analytics, embedded systems. IOT Levels and deployment templates.	08
3	**IoT Hardware and Software:** Sensor and actuator, Humidity sensors, Ultrasonic sensor, Temperature Sensor, Arduino, Raspberry Pi, LiteOS, RIoTOS, Contiki OS, Tiny OS.	07

WHAT ARE WE GOING TO LEARN !!

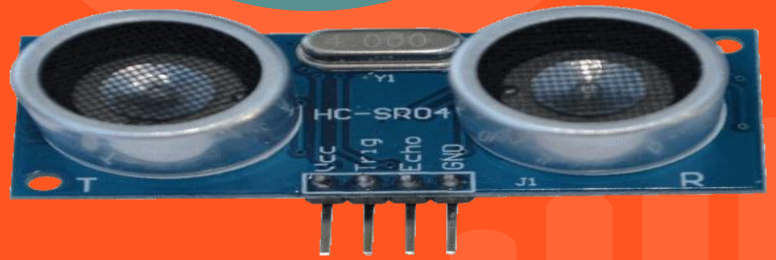
- COMPARISON between **Transducers Sensors And Actuators**.
- Brief description About **Sensors, Types of Sensors, Classifications** .
- Actuators and it’s working.
- COMPUTER PROCESS CONTROL SYSTEM.
- Analog To Digital Convertor.
- Sampling ,Quantization, Encoding.

Transducer

Any device that convert one form of energy to another.

Sensors

Devices that measures physical quantities and convert them into signals which can be read by instruments



Actuators

Devices that actuates or moves something.More specifically, they converts energy into motion or mechanical energy

Transducer

Three types of transducers: light bulb, microphone, and electric motors

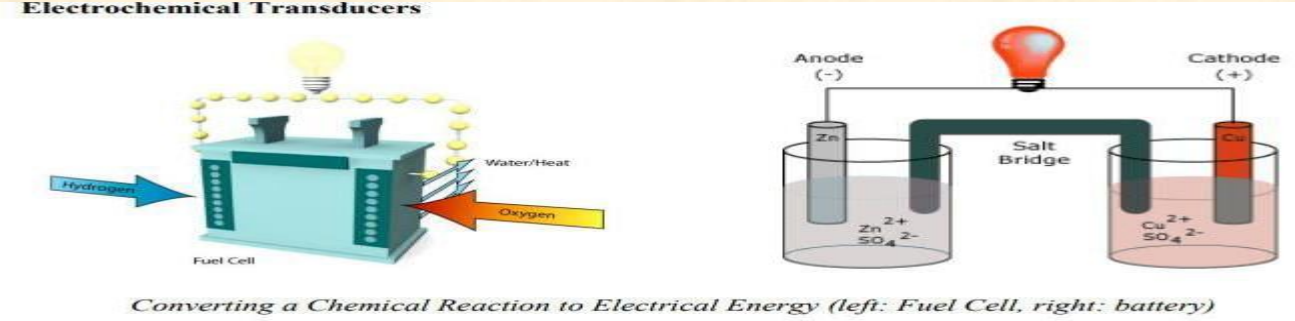
A transducer is any device which converts one form of energy into another. Examples of common transducers include the following:

- A microphone converts sound into electrical impulses and a loudspeaker converts electrical impulses into sound (i.e., sound energy to electrical energy and vice versa).
- A solar cell converts light into electricity and a thermocouple converts thermal energy into electrical energy.
- An incandescent light bulb produces light by passing a current through a filament. Thus, a light bulb is a transducer for converting electrical energy into optical energy.
- An electric motor is a transducer for conversion of electricity into mechanical energy or motion.

Basic Concept of Transducer

A transducer is defined as a substance or a device that converts (or transfers) an input energy into a different output energy.

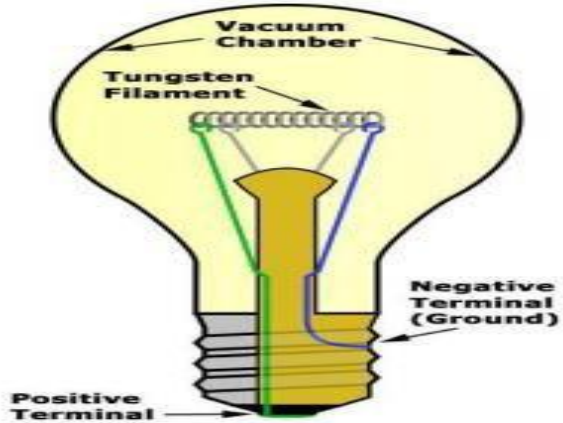
Different types of Transducer:



Converting a Chemical Reaction to Electrical Energy (left: Fuel Cell, right: battery)

Contd…

The Incandescent Light Bulb (a transducer)



Schematic of an incandescent light bulb. A power source's positive and negative terminals are connected to the bulb's terminals. An electrical current flows through the filament. The resistance of the tungsten filament converts the electrical energy into heat and light.

SENSORS

Sensors

Human beings are equipped with 5 different types of sensors.

Detects
Light

Detects
Sound

Detects
Certain Chemicals

Detects
Pressure & Temperature

Eyes detect light energy, ears detect acoustic energy, a tongue and a nose detect certain chemicals, and skin detects pressures and temperatures. The eyes, ears, tongue, nose, and skin receive these signals then send messages to the brain which outputs a response. For example, when you touch a hot plate, it is your brain that tells you it is hot, not your skin.

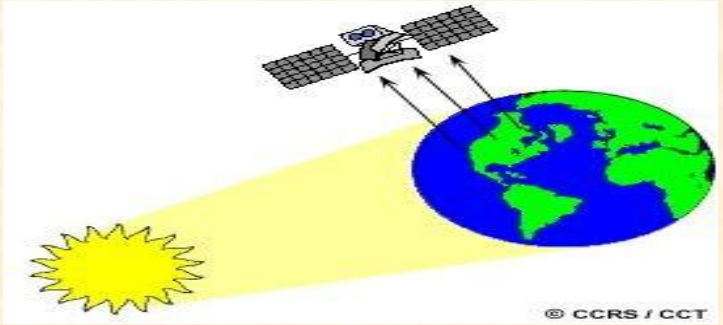
Classification of Sensors

In **passive sensing**, sensor measures the energy that is naturally available, such as thermal infrared, surface emissions.

In **active sensing,** sensors provides energy on their own as a source of illumination. The energy reflected by the target is detected and measured.

Active vs. Passive Sensors

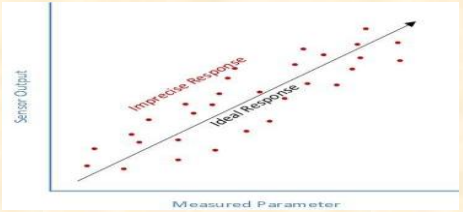
Active Sensor

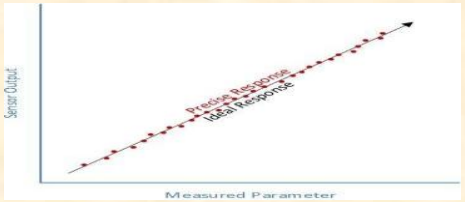


Passive Sensor

What makes a goodsensor?

- **Precision:** An ideal sensor produces same output for same input. It is affected by noise and hysteresis.
- **Resolution:** The ability to detect small changes in the measuring parameter
- **Accuracy:** ‘It is the combination of precision, resolution and calibration.’





Calibration of Sensors

Most sensors are **not ideal** and are often affected by **surrounding noise**. For a color sensor, this could be ambient light, and specular distributions.

If a sensor is known to be accurate, it can be used to make comparison with reference readings. This is usually done with respect to certain standard physical references, such as for a rangefinder we may use a ruler for calibration.

Each sensor has a ‘characteristic curve’ that defines the sensor’s response to an input. The **calibration process** maps the sensor’s response to an ideal linear response

Contd…

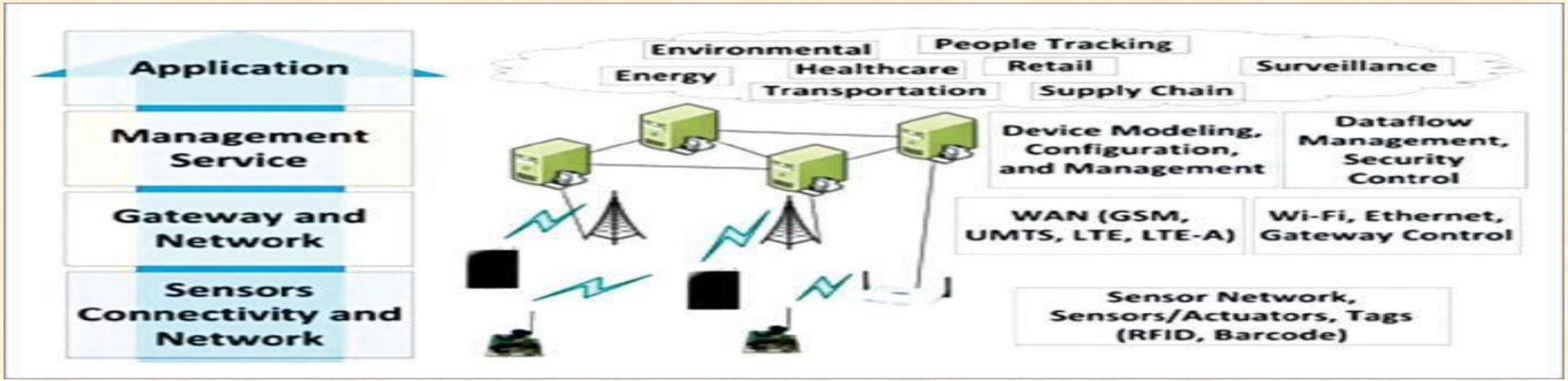
Purpose

The main purpose of sensors is to collect data from the surrounding environment. Sensors, or ‘things’ of the IoT system, form the front end. These are connected directly or indirectly to IoT networks after signal conversion and processing.

How Sensor Works in IOT?

An IOT System consists of sensors/devices which “talk” to the cloud through some kind of connectivity.

Once the data gets to the cloud, Software processes it then might decide to perform some action , such as sending an alert or automatically adjusting the sensor/devices
without the need of the user.



Characteristic Curve of Sensor

Suppose the output of a sensor for some physical quantity *x(t)* is given by *f(x(t))*:

- **Linear Model** $f(x(t)) = ax(t)$ $a \in \mathbb{R}$

, where

- **Affine Model** $f(x(t)) = ax(t) + b$ $a \in \mathbb{R}$ $b \in \mathbb{R}$

, where ,

Often, ‘a’ is called the proportionality constant, which gives an idea of the sensitivity of the sensor, and ‘b’ denotes the bias.

Note: **The sensitivity of a sensor is ratio of output value to measured quantity.**

Sensor's Operating Range

If the operating range of a sensor is *(L, H)*,

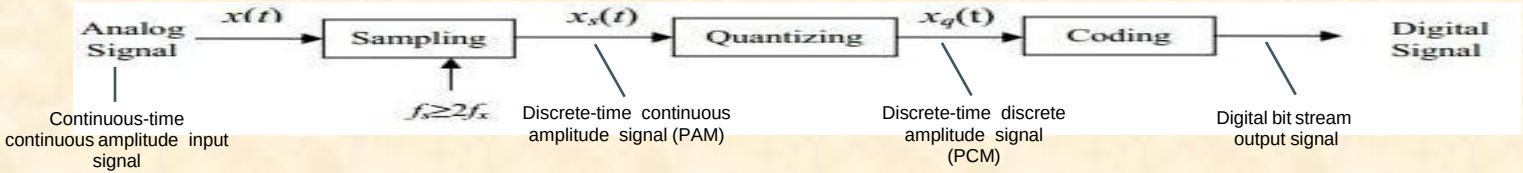
$$f(x(t)) = \begin{cases} ax(t) + b & \text{if } L \leq x(t) \leq H \\ aH + b & \text{if } x(t) > H \\ aL + b & \text{if } x(t) < L, \end{cases}$$

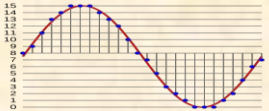
To get an idea of how precise the measurements of a sensor can be, one defines its **precision 'p'** as the smallest difference between two distinguishable sensor readings of the physical quantity.

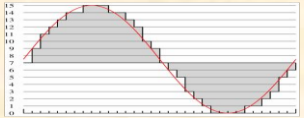
Sampling and Quantisation

The **process of the discretization** of the domain of the signal being measured is called sampling, whereas quantization refers to the **discretisation of the range**.

Pulse Code Modulator





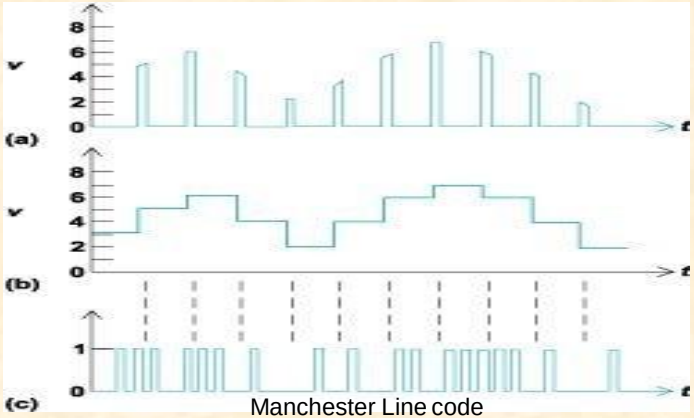


Sampling and Quantisation

SAMPLING: Evaluating the input signal at discrete units of time, say *0, T, 2T, ... nT.*

QUANTIZING: Provides discretized values to the input on basis of a finite number of thresholding conditions

ENCODING: Transforms the digital data into a digital signal, comprising of bits 0111011…, on basis of various schemes.



Manchester Line code

Sampling and Quantisation

- If the **sampling rate** isn't high, one can end up with different signals(aliases) during reconstruction, that fit the same set of sample points. This is called aliasing, and is undesirable. For best sampling, the sampling rate must be >= 2 times the frequency of the signal. **(Nyquist Shannon Sampling Theorem)**
- In the case of quantisation, selection of fewer **levels of discretisation** can lead to progressive loss of spatial detail. Also, contours(artificial boundaries) can start appearing due to sudden changes in intensity. For audio signals, this can be heard as noise/distortions.

VARIETIES OF SENSORS

Acoustic Sensors
Geophone Hydrophone
Microphone

Automotive Sensors Air flow meter Speedometer
Hall-Effect Sensor Air- Fuel Ratio meter

Electric Current Sensors
Hall Probe Magnetometer
Current sensor Voltage Detector

Proximity Sensor
Infrared sensor
Ultrasonic sensor

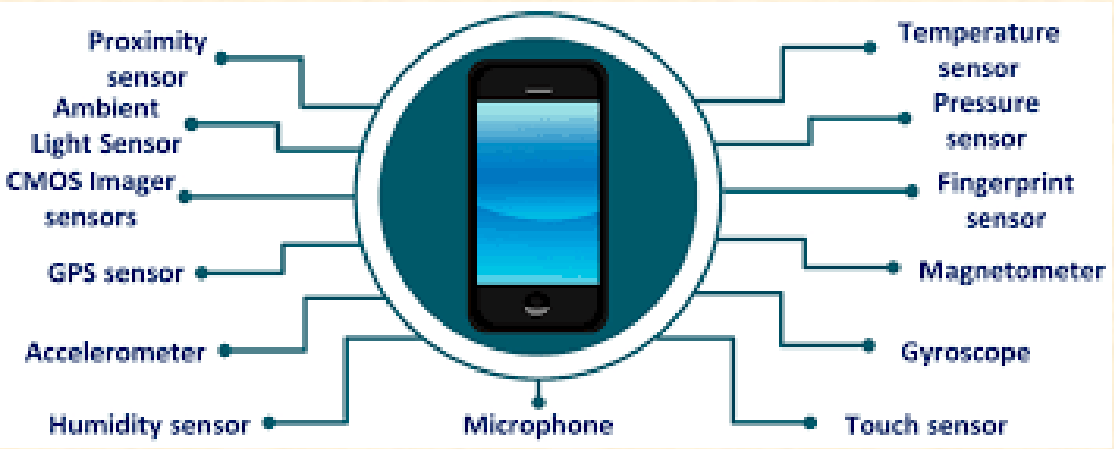
Navigation Instruments
LIDAR
Gyroscope Rotary Encoder
Odometer Tachometer

Optical Sensor
Photodiode Infrared sensor Camera

Sensor Types:

1. Temperature Sensor
2. Humidity Sensor
3. Ultrasonic Sensor
4. Motion Sensor
5. Pressure Sensor
6. Gas Sensor
7. Smoke Sensor etc.
8. Camera
9. Inertial Measurement Unit
10. Photo-resistors
11. Infrared Sensor
12. Flex Sensors
13. Ultrasonic Sensor
14. Rotary Encoder
15. Touch Sensor
16. Thermocouple

Different Types of Sensors

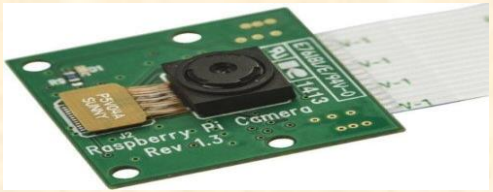

Proximity sensor
Ambient Light Sensor
CMOS Imager sensors
GPS sensor
Accelerometer
Humidity sensor
Microphone
Temperature sensor
Pressure sensor
Fingerprint sensor
Magnetometer
Gyroscope
Touch sensor

1. Camera

Vision processing requires a lot of RAM, and even low resolution cameras may give lots of data, parsing through which can be difficult.

Cameras draw in around 0.1 A current, the current rating of the USB hub to which they are attached must be checked.

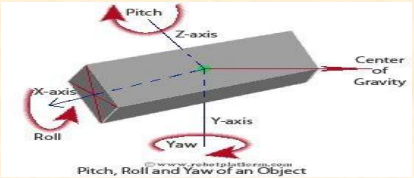
Raspberry Pi Camera

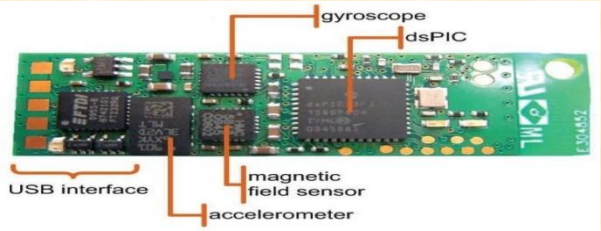


Advamotion

2. Inertial Measurement Unit

- Consists of three sensors:
  - **Accelerometer:** Used to measure inertial acceleration
  - **Gyroscope :**Measures angular velocity about defined axis
  - **Magnetometer :** Can be used along with gyroscope to get better estimates of robot's orientation (i.e. roll, pitch, yaw)

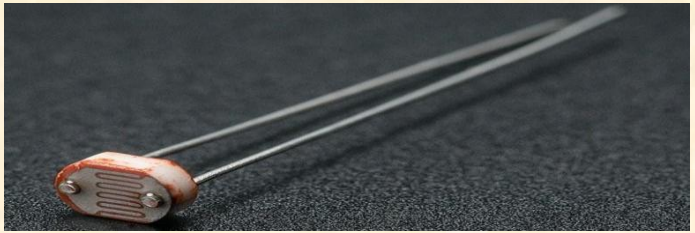




3. Photo-resistors

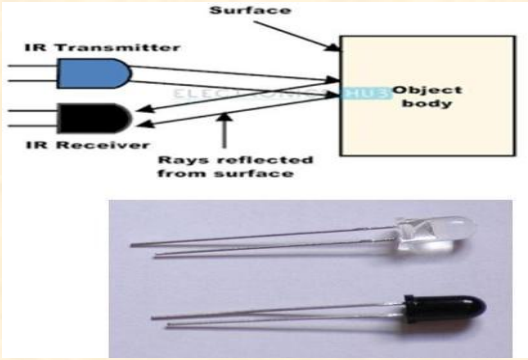
Light sensitive resistors whose resistance decreases as the intensity of light they are exposed to increases. They are made of high resistance semiconductor material.

When light hits the device, the photons give electrons energy. This makes them jump into the conductive band and thereby conduct electricity.



4. Infrared Sensor

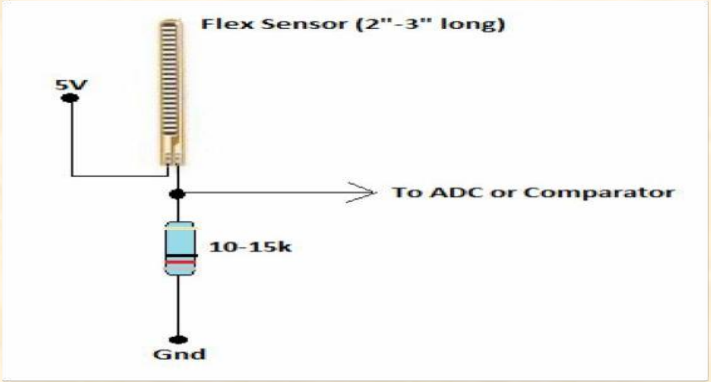
- IR led is led that emits light in IR region and can't be seen by the eyes.
- Photodiode is a type of diode which works in reverse bias and its resistance is changed when subjected to change in light intensity.
- They are used for colour detection etc.



5. Flex Sensors

Measure the amount of deflection caused by bending, also called bend sensors.

The bending must occur around a radius of curvature, as by some angle at a point isn't effective and if done by more than 90 deg., may permanently damage the sensor.



6. Ultrasonic Sensor

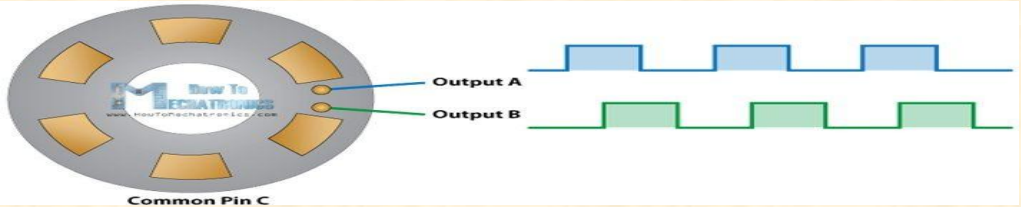
These are commonly used for obstacle detection.

Works on principle similar to that of Sonar which consists of time of flight,the Doppler effect and the attenuation of sound waves.

7. Rotary Encoder

They convert the angular position of a shaft or axle to a
analog / digital code.

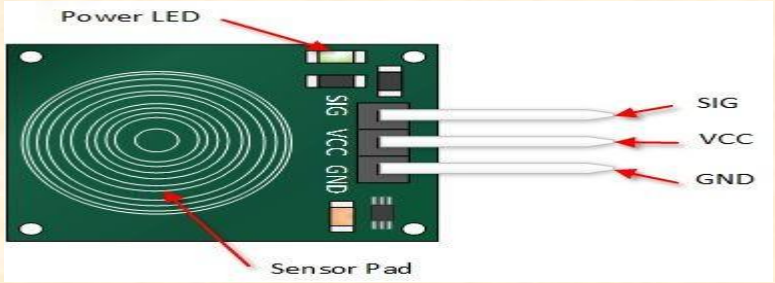
They may represent the value in **absolute** or **incremental terms**. The advantage of absolute encoders is that they maintain the information of the position even when power is removed, and this is available immediately on its application.



8. Touch Sensor

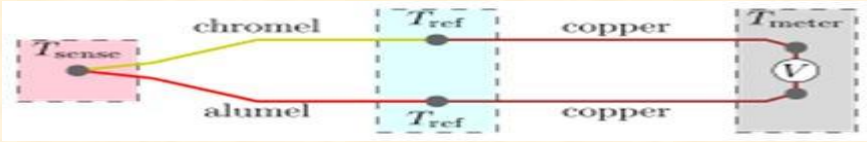
Touch sensors can be defined as switches that are activated by the touch.

Examples includes capacitance touch switch, resistance touch switch, and piezo touch switch.



9.Thermocouple

- Converts thermal energy into electrical energy and is used to measure temperature.
- When two dissimilar metal wires are connected at one end forming a junction, and that junction is heated, a voltage is generated across the junction .



Humidity Sensors

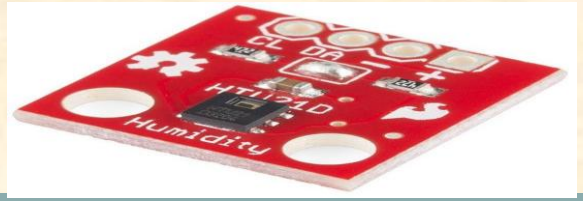
Humidity is defined as the amount of water present in the surrounding air.

This water content in the air is a key factor in the wellness of mankind.

For example, we will feel comfortable even if the temperature is 00C with less humidity i.e. the air is dry.

But if the temperature is 100C and the humidity is high i.e. the water content of air is high, then we will feel quite uncomfortable.

The units for measurement humidity is RH (relative humidity), D/F PT (Dew/frost point) & PPM (parts per million).



Humidity Sensors

sensing, measuring, monitoring and controlling humidity is a very important task.

Some of the important areas of application for sensing, measuring and controlling Humidity are mentioned below:

- Domestic
- Industrial
- Agriculture
- Electronics
- Semiconductor
- Medical

Important Terms Related to Humidity

Relative Humidity(RH): is the amount of moisture in the air compared to what the air can hold at that temperature.

RH: How much moisture is in the air (%). When the stmosphere is saturated with Moisture the related Humidity is 100 %

Moisture: means water content of any material or substance. But practically, the term Moisture refers to the water content in solids and liquids.

Dew Point: is the temperature at which the relative humidity is 100 %

When air temperature reaches the dew point temperature:

- 100 % relative Humidity
- Condensation or precipitation occurs

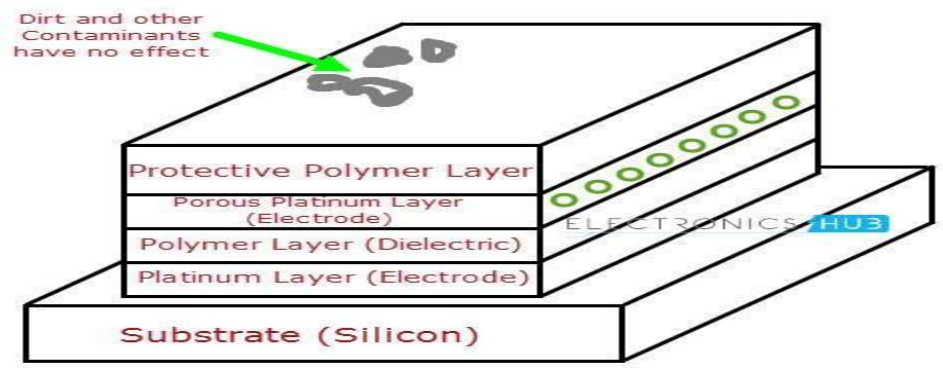
Classification of Humidity Sensors

- **Capacitive Relative Humidity Sensors**
- **Resistive Humidity Sensors (Electrical Conductivity Sensors)**
- **Thermal Conductivity Humidity Sensors**

Capacitive Relative Humidity Sensors

In Capacitive Relative Humidity (RH) Sensors, the electrical permittivity of the dielectric material changes with change in humidity.

Working of Capacitive RH Sensors



Advantages of Capacitive Humidity Sensors

•The output voltage is near linear.
•They provide stable results over long usage.
•Can detect wide range of RH.

Disadvantages of Capacitive Humidity Sensors

•The distance from the sensor and signalling circuit is very limited.

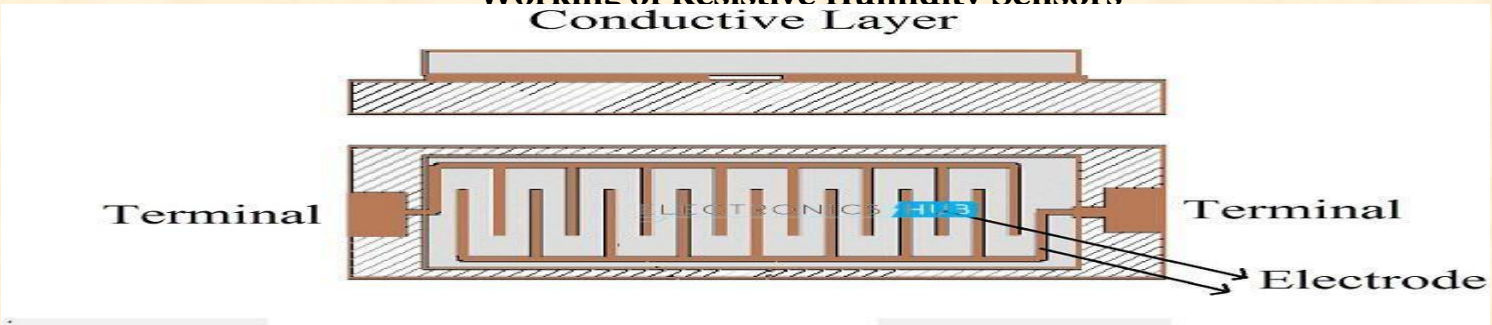
Applications of Capacitive Humidity Sensors

Capacitive Humidity Sensors are used in a wide range of applications including but not limited to:
•HVAC Systems
•Printers and Fax Machines
•Weather Stations
•Automobiles
•Food Processing
•Refrigerators, Ovens and Dryers

Resistive Humidity Sensors (Electrical Conductivity Sensors)

Resistive Humidity Sensors are another important type of Humidity Sensors that measure the resistance (impedance) or electrical conductivity. The principle behind resistive humidity sensors is the fact that the conductivity in non – metallic conductors is dependent on their water content.

Working of Resistive Humidity Sensors



Advantages of Resistive Humidity Sensors

• Low cost

•Small Size

•The distance between the sensor and signal circuit can be large (suitable for remote operations).

• Highly interchangeable as there are no calibration standards.

Disadvantages of Resistive Humidity Sensors

•Resistive Humidity Sensors are sensitive to chemical vapors and other contaminants

•The output readings may shift if used with water soluble products.

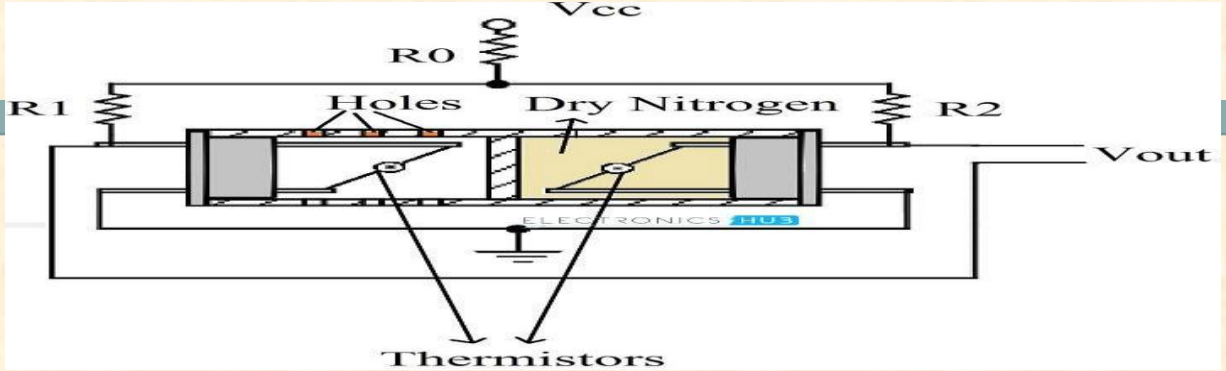
Applications of Resistive Humidity Sensors

Resistive or Electrical Conductive Humidity sensors are low cost sensors with relatively small size. They are often used in several industrial, domestic or residential and commercial applications.

Thermal Conductivity Humidity Sensors

- Thermal Conductivity Humidity Sensors are also known as Absolute Humidity (AH) Sensors as they measure the Absolute Humidity.
- Absolute Humidity doesn't take temperature in to account but it changes with temperature and pressure.
- Thermal Conductivity Humidity Sensors measure the thermal conductivity of both dry air as well as air with water vapor. The difference between the individual thermal conductivities can be related to absolute humidity.

Working Principle



Advantages of Thermal Conductivity Humidity Sensors

•Suitable for high temperature environments and high corrosive situations.

•Very durable

•Higher resolution compared to other types

Disadvantage of Thermal Conductivity Humidity Sensors

Exposure to any gas with thermal properties different than Nitrogen might affect reading measurement.

Applications of Thermal Conductivity Humidity Sensors

•Drying kilns

•Pharmaceutical plants

•Owens

•Clothes dryers and drying machines

Important Considerations when Selecting a Humidity Sensor

The following are some of the factor that must be taken into consideration when selecting a Humidity Sensor.

Accuracy of the sensor.

Calibration – requirements and methods Size of the sensor

Cost of the sensor and cost of replacement

Output repeatability

Circuit complexity Resistance to contamination

Reliability of the sensor

Temperature Sensor

The temperature sensor is used to detect the heat energy which is produced from an object otherwise nearby area. These sensors are applicable for the Internet of Things (IoT), which includes from manufacturing to farming. The main role of these sensors in manufacturing is for temperature monitoring of machines. Similarly, in the agriculture field, these sensors are used to monitor the temperature of plants, soil, and water.

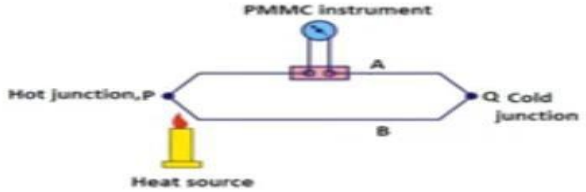
fig: Temperature Sensor Major Applications: mainly includes refrigerators, AC's etc.

Types of Temperature Sensors

- Thermocouple Sensor
- Infrared Sensors
- Thermistor Sensor
- Resistance Temperature Detector
- Thermometer

Thermocouple Sensor

A Temperature Sensor is the instrumentation equipment which is used to measure temperature or heat on the operating machine part. Temperature sensing is performed by equipment called Thermocouple. A thermocouple is a temperature-measuring device consisting of two dissimilar conductors that contact each other at one or more points. It produces a voltage when the temperature of one of the points differs from the reference temperature at other parts of the circuit.



IR Sensors

Infrared sensors are mainly used to measure the heat which is produced by objects. These sensors are used in the various applications of IoT like healthcare for monitoring the flow of blood, BP, etc.
These sensors are used in smartphones for controlling, wearable devices for detecting the amount of light, detection of blind-spot within vehicles, etc.

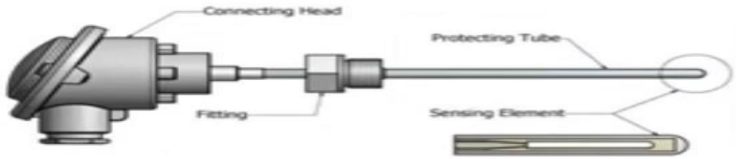
Fig: Infrared sensor

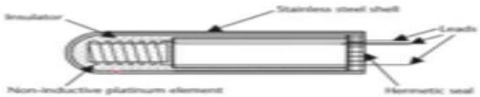
Thermistor Sensor .

The thermistor sensor is a type of sensor. This type of sensors is used mostly in the human thermometers. If there is a change in the temperature, then the electrical current or resistance also changes. The thermistor is prepared by using the semiconductor materials with a resistivity which is especially sensitive to temperature. The resistance of a thermistor decreases with increasing temperature so that when the temperature changes, the resistance change is predictable.

Resistance Temperature Detector

These are the temperature sensors with a resistor that changes the resistive value simultaneously with temperature changes. The RTDs are used in a wide temperature range from -500C to 5000C for thin film and for the wire wound variety the range is from the +2000C to 8500C. The thin layer of platinum on a substrate is present on the thin film RTD element. A new pattern is created which provide the electrical circuit and it is trimmed to give a specific resistan





Thermometer:

It is a device which is used to measure the temperature of any glass solids or liquids. In this type mercy or alcohol is used in a tube whose volume is changed by changing the temperature. Its volume is directly proportional to temperature.

ACTUATORS

TYPES OF ACTUATORS

In a robot, actuators are used in order to produce some mechanical movement.

Electric

Electro-mechanical devices which allow movement through use of electrically controlled systems of gears

DC Motor

Hydraulic

Transforms energy stored in reservoirs into mechanical energy by means of suitable pumps

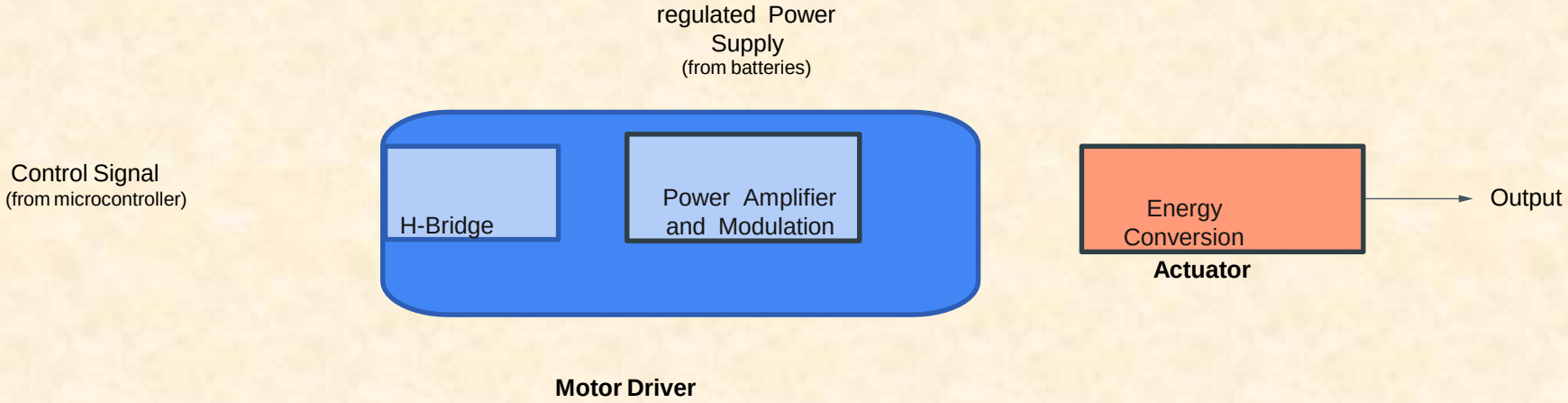
Water Pump by Tefulong Ltd.

Pneumatic

Uses pneumatic energy provided by air compressor and transforms it into mechanical energy by means of pistons or turbines

Pneumatic cylinder by Janatics Ltd.

ACTUATOR FUNCTIONAL DIAGRAM



MOTOR DRIVER

- Microcontrollers, typically, have current rating of 5-10 mA, while motors draw a supply of 150mA. This means motors can’t be directly connected to microcontroller.

- For electromechanical actuators, following motor drivers are often used:

  - **Simple DC Motors:** L298, L293
  - **Servo Motors:** Already have power cable and different control cable
  - **Stepper Motors:** L/R Driver Circuit, Chopper Drive

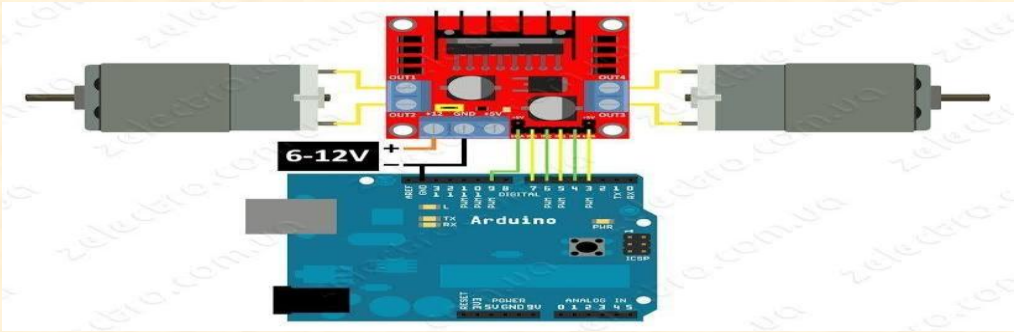
L298N Stepper Motor Driver Controller

L298 DUAL H-BRIDGE IC

Allows to independently control two DC motors up to 2 A each in both directions.

Power consumption for logical part 0-36 mA

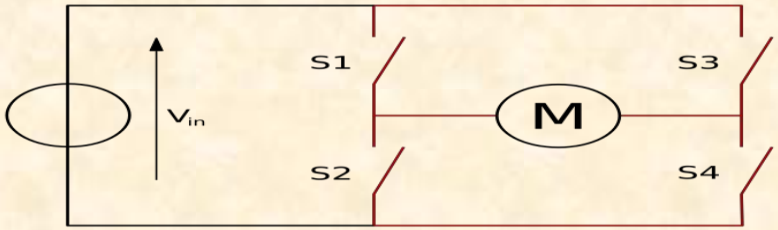
- Requires protective diodes against back e.m.f. externally



Connections to L298 Dual H-Bridge 2A

H - BRIDGE

It is an electronic circuit used to apply voltage across a load in either direction on basis of input from a microcontroller

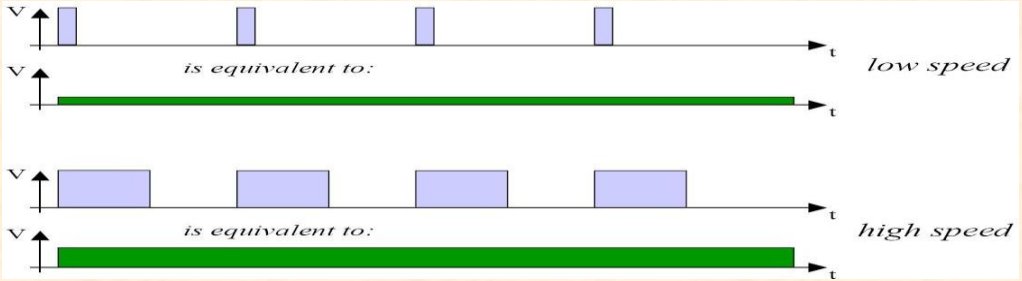


S1	S2	S3	S4	Result
1	0	0	1	Motor moves right
0	1	1	0	Motor moves left
0	0	0	0	Motor coasts
0	1	0	1	Motor brakes
1	0	1	0	Motor brakes
1	1	0	0	Short circuit
0	0	1	1	Short circuit
1	1	1	1	Short circuit

SPEED CONTROL USING PWM

- Pulse Width Modulation (PWM) is scheme in which duty cycle of square wave output

- Voltage seen by the load is directly proportional to the unregulated source voltage



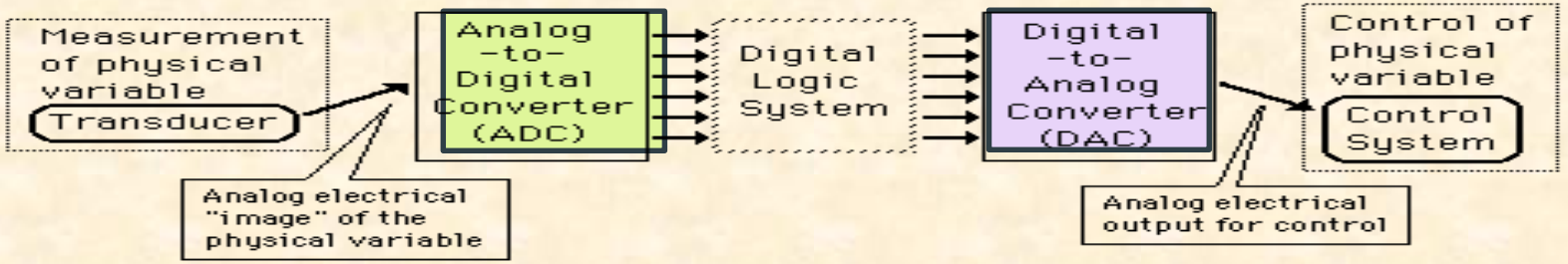
Components of a System Hardware

Data Handling Systems

Both data about the physical world and control signals sent to interact with the physical world are typically "analog" or continuously varying quantities.

In order to use the power of digital electronics, one must convert from analog to digital form on the experimental measurement end and convert from digital to analog form on the control or output end of a laboratory system.

Data Collection after Control



Source: http://hyperphysics.phy-astr.gsu.edu/hbase/hph.html

LiteOS

•LiteOS is a real-time operating system (RTOS).

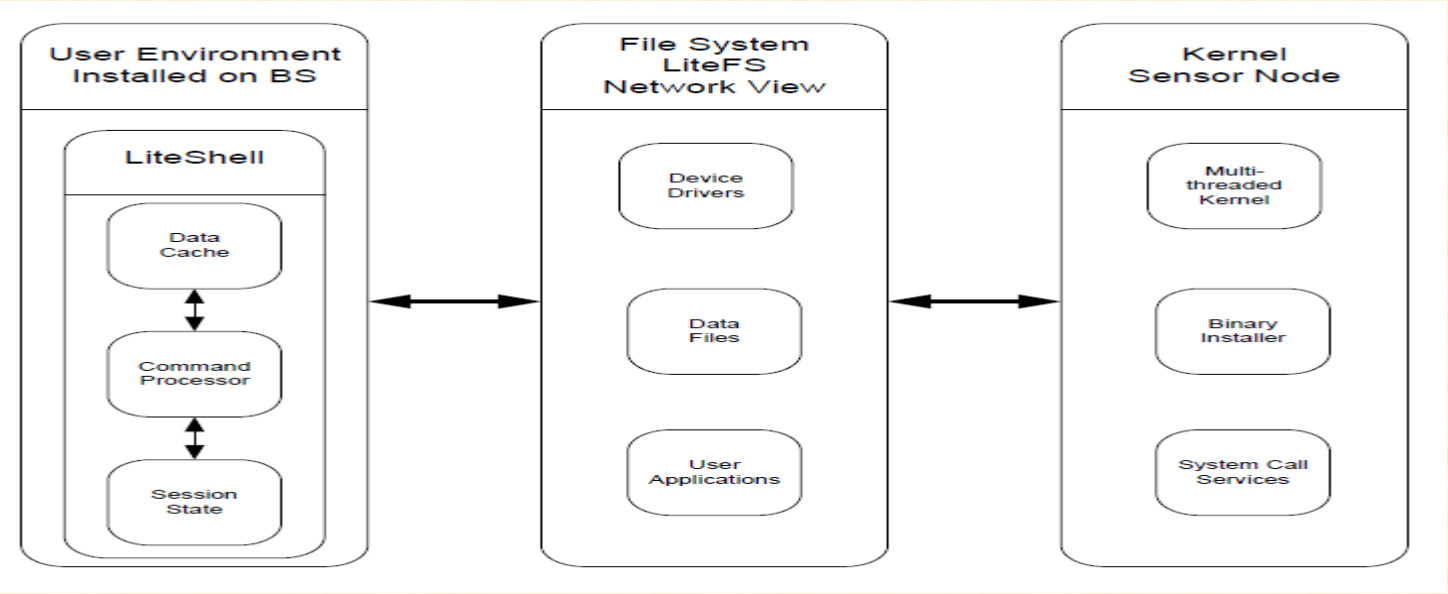
•LiteOS is a Unix-like operating system that fits on memory-constrained sensor nodes.

•This operating system allows users to operate wireless sensor networks like operating Unix, which is easier for people with adequate Unix background.

•LiteOS provides a familiar programming environment based on Unix, threads, and C.

•It follows a hybrid programming model that allows both event-driven and thread-driven programming

LiteOS Architecture



LiteOS

Programming Model:

•LiteOS is a multitasking OS and it supports multithreading.

•processes run applications as separate threads.

•LiteOS also provides support for event handling. Application programmers can register event handlers using a callback facility provided by LiteOS.

•To avoid potential race conditions, LiteOS provides atomic_start() and atomic_end() functions. Whenever shared data among different threads is accessed or modified, it is highly recommend to use these functions.

LiteOS

Scheduling:

- LiteOS provides an implementation of Round Robin scheduling and Priority-based scheduling.
- The tasks run to completion or until they request a resource that is not currently available.
- When a task requires a resource that is not available, the task enables interrupts and goes to sleep mode.
- Once the required resource becomes available, the appropriate interrupt is signaled and the task resumes it execution from where it had left. When a task completes its operation it leaves the kernel.
- When there are no active tasks in the system, the sensor node goes to sleep mode. Before going to sleep mode the node enables its interrupts so that it can wake up at the proper event or time.

LiteOS

Memory Protection and Management

- Inside the kernel, LiteOS supports dynamic memory allocation through the use of C-like malloc and free functions.
- User applications can use these APIs to allocate and de-allocate memory at run-time.
- Dynamic memory grows in the opposite direction of the LiteOS stack.
- The LiteOS kernel compiles separately from the application, therefore the address space is not shared between the kernel and the application. Similarly, each user application has its separate address space. Processes and Kernel memory safety is enforced through separate address spaces.

LiteOS

Communication Protocol Support

- LiteOS provides communication support in the form of files
- LiteOS creates a file corresponding to each device on the sensor node.
- Whenever there is some data that needs to be sent, the data is placed into the radio file and is afterward wirelessly transmitted.
- At the network layer LiteOS supports geographical forwarding.
- Each node contains a table that can only hold 12 entries.

LiteOS

Resource Sharing

LiteOS suggest the use of APIs provided for synchronization whenever a thread wants to access resources that are shared by multiple threads. The LiteOS documentation does not provide any detail on how system resources are shared among multiple executing threads.

LiteOS

Support for Real-Time Applications

•LiteOS does not provide any implementation of networking protocols that support real-time multimedia applications.

•It provides a priority-based process scheduling algorithm but once a process is scheduled it runs to completion

•This can result in a missed deadline of a higher priority process that enters the ready queue once a low priority process has been scheduled.

LiteOS

Additional Features

•Lite File System (LiteFS)

LiteOS provides support for a hierarchical file system called LiteFS. LiteFS provides support for both files and directories. LiteFS is partitioned in three modules. It keeps open file descriptors, memory allocation bit vectors, and information about flash memory layout in RAM.

Simulation Support

LiteOS supports wireless sensor networks simulations through AVRORA.

Language Support

LiteOS supports application development in the LiteC++ language.

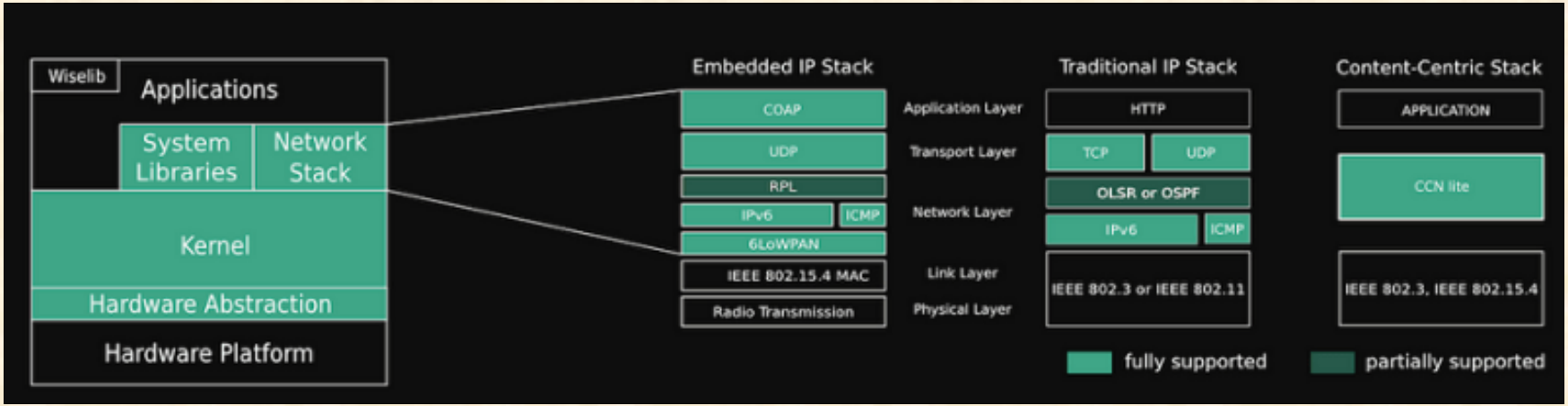
Documentation Support

LiteOS documentation can be found on the LiteOS home page at:
http://www.liteos.net.

RIOTOS

- Microkernel (for robustness)
- Modular structure to deal with varying requirements
- Tickless scheduler
- Deterministic kernel behavior
- Low latency interrupt handling
- POSIX like API
- Native port for testing and debugging

RIOT structure



RIOT: hardware support

- CPUs:
  - ARM7
    - NXP LPC2387
    - Freescale MC1322
  - ARM Cortex
    - STM32f103 (Cortex M3)
    - STM32f407 (Cortex M4)
    - NXP LPC1768
  - MSP430
    - MSP430x16x
    - CC430
- Boards:
  - FUB Hardware
    - MSB-A2
    - PTTU
    - AVSExtrem
    - MSB-430(H)
  - TelosB
  - Redbee Econotag
  - WSN430 (Senslab)
  - TI eZ430-Chronos(Watch)
  - AgileFox (FIT testbed)
  - More to come, e.g. mbed hardware

RIOT: the native port

- Run RIOT as is on your Linux computer
- Emulates a network using virtual network devices
- Allows for enhanced debugging with gdb, valgrind, wireshark etc.

```
--> ./bin/rpl_udp_router.elf tap0
RIOT native interrupts/signals initialized.
RIOT native uart0 initialized.
LED_GREEN_OFF
LED_RED_ON
RIOT native board initialized.
RIOT native hardware initialization complete.

kernel_init(): This is RIOT! (Version: 2013.08-622-ga723-tbilisi)
Scheduler...[OK]
kernel_init(): jumping into first task...
UART0 thread started.
uart0_init() [OK]
RPL router v1.1
> ps
ps
         pid | name                 | state    Q | pri | stack ( used) location   | runtime | switches
           0 | idle                 | pending  Q |  31 |  8192 ( 1100) 0x8068ce0 | 99.927% |        1
           1 | main                 | running  Q |  15 | 16384 ( 3268) 0x8064ce0 |  0.012% |        6
           2 | uart0                | bl rx    _ |  14 |  8448 (  896) 0x806ad20 |  0.005% |       10
             | SUM                  |             |     | 33024
>
```

RIOT: Microkernel

- minimalistic kernel

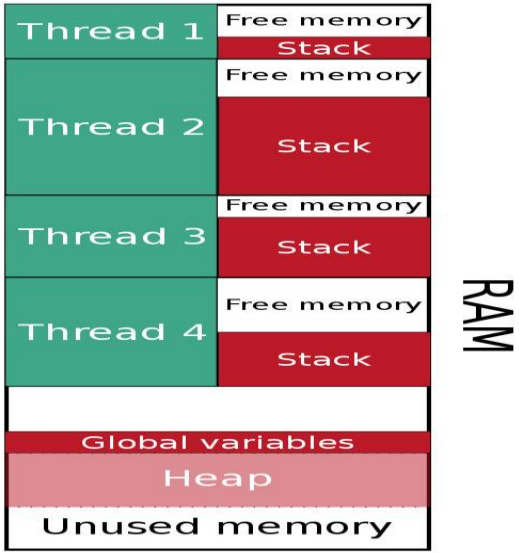
Language	files	blank	comment	code
C	12	350	281	1017
C Header	22	202	719	377
SUM:	34	552	1000	1394

- Kernel functions:
  - Scheduler
  - IPC
  - Mutexes
  - Timer
- Modules and drivers communicate over IPC
- Deterministic runtime of all kernel functions
- Optimized context switching

RIOT: scheduler

- Tickless, i.e. no periodic timer event
  - more complex to implement, but most energy-efficient
- Run-to-complete, i.e. scheduler does not distribute equally to all threads
- Priority based
  - Priorities have to be chosen carefully to fulfill real-time requirements

RIOT: memory management

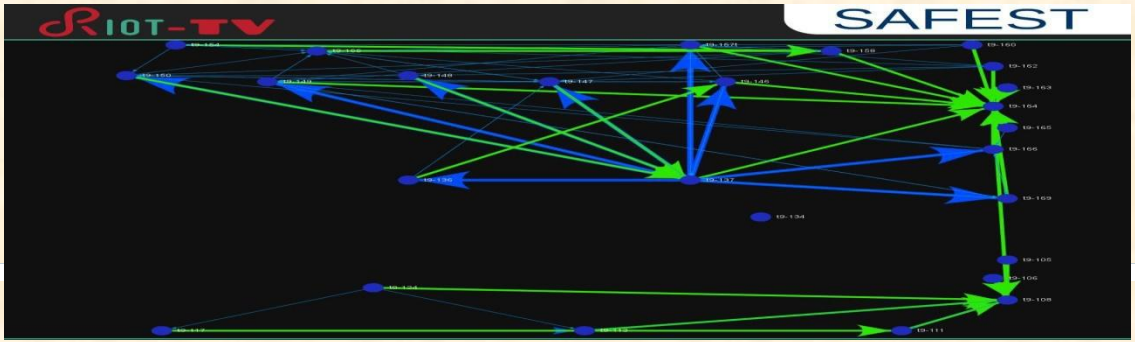


- Every thread has its own stack
- The stack also contains the tcb
- There‘s no memory protection

  => a stack overflow can destroy another stack

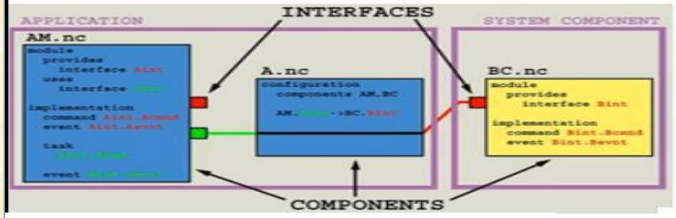
RIOT as a tool for research

- Network protocols like 6LoWPAN, RPL, CCN etc.
- Distributed operating systems
- Various testbeds and virtual networks with desvirt
- Measurement of energy consumption



Tiny OS

An ***operating system*** for motes in WSNs



TinyOS

- It's an Open source Embedded Operating System having BSD(Berkeley Software Distribution
- Specially Designed to be used in Wireless Sensor Network
- Development started by **University of California in cooperation** with **Intel Research and Crossbow Technology in 1999**
- 1st version Ver. 0.6 was released in 2001
- Latest Version is Ver. 2.1.2 released in August 2012

why do we need a new OS?

- ✓Lower Power
- ✓Limited memory
- ✓Slow CPU
- ✓Size (Small)
- ✓Limited hardware parallelisms
- ✓Communication using radio
- ✓Low-bandwidth
- ✓Short range

- ✓Huge !
- ✓Multi-threaded architecture =>large memory
- ✓I/O model
- ✓Kernel and user space separation
- ✓Typically no energy constraints
- ✓Ample available resources

TinyOS Solutions

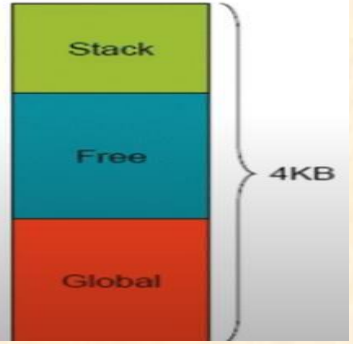
- Supports Concurrency
  - -event-driven architecture
- Software Modularity
  - -application=scheduler + graph component
  - - a component contains commands, event handlers, internal storage, tasks
- Efficiency: get done quickly and then sleep
- Static memory allocation.

Tiny OS Memory Model

- ✓STATIC memory allocation!
  - ✓No heap (malloc)
  - ✓No function pointers
  - ✓No dynamic, run-time allocation

Global variables
Available on a per-frame basis
Conserve memory
Use pointers

Local variables
Saved on the stack
Declared within a method

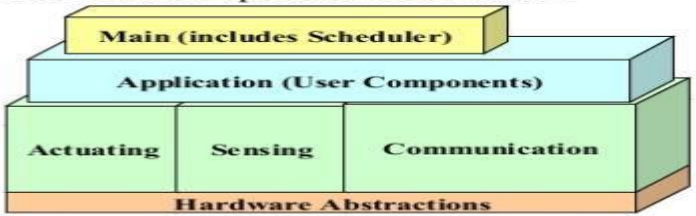


TinyOS Architecture:

- The scheduler has a fixed-length queue, FIFO
- Task run atomically
- Interrupt handlers
- Complex interactions among components
- Event

In most mote applications, execution is driven solely by timer events and the arrival of radio messages

- Application = scheduler + graph of components
  - Compiled into one executable
- Event-driven architecture
- Single shared stack
- No kernel/user space differentiation



In the era of automation, the use of Wireless Sensor Network is increasing day by day

- In India, due to the *Smart City project, it's going to be used in very* wide range
- TinyOS is very useful for WSN as it efficiently manipulates every events in WSN and gives better monitoring

ContikiOS

•Contiki is an open source operating system for the Internet of Things.

•runs on networked embedded systems and wireless sensor nodes.

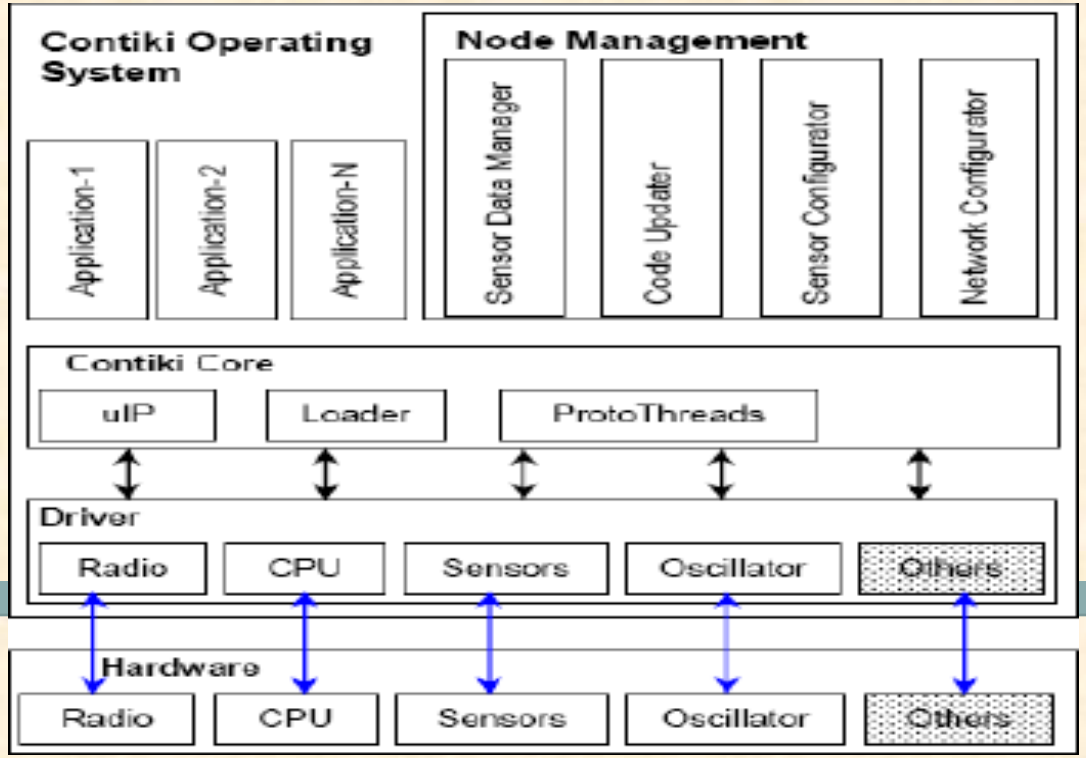
Contiki is a highly portable OS and it is build around an event-driven kernel.

•Contiki provides preemptive multitasking that can be used at the individual process level.

•Polling mechanism is used to avoid race condition.

•Any scheduled event will run to completion, however, event handlers can use internal mechanisms for preemption.

ContikiOS Architecture



ContikiOS

FunctionalAspect:

It'skernel functions as an event-driven kernel; multithreading is supported by an application library. Inthis sense it is a hybrid OS.

Contiki realises the separation of concern of the basic system support form the rest of the dynamically loadable and programmable services (called processes).

The services communicate with each other through the kernel by posting events.

The ContikiOS kernel does not provide any hardware abstraction; but it allows device drivers and application directly communicate with the hardware.

Each Contiki service manages its own state in a private memory space and the kernel keeps a pointer to the process state.

ContikiOS

Programming Model:

•Contiki supports preemptive multithreading. Multi-threading is implemented as a library on top of the event-driven kernel.

•The library can be linked with applications that require multithreading. The Contiki multithreading library is divided in two parts: a platform independent part and a platform specific part

•Contiki uses protothreads.

•Protothreads are designed for severely memory constraint devices because they are stack-less and lightweight.

•The main features of protothreads are: very small memory overhead (only two bytes per protothread), no extra stack for a thread, highly portable (*i.e., they are fully written in C and hence there is no architecture-specific assembly code). Since events run to completion and Contiki does not allow interrupt handlers to post new events, no process synchronization is provided in Contiki.*

ContikiOS

Scheduling

Contiki is an event-driven OS, therefore it does not employ any sophisticated scheduling algorithm. Events are fired to the target application as they arrive. In case of interrupts, interrupt handlers of an application runs w.r.t. their priority.

Memory Management and Protection:

Contiki supports dynamic memory management. Apart from this it also supports dynamic linking of the programs. In order to guard against memory fragmentation Contiki uses a Managed Memory Allocator.

ContikiOS

Communication Protocol Support:

- Contiki supports a rich set of communication protocols. In Contiki, an application can use both versions of IP *i.e., IPv4 and IPv6. Contiki provides an implementation of uIP, a TCP/IP protocol stack for small 8 bit micro-controllers. uIP does not require*
- *its peers to have a complete protocol stack, but it can communicate with peers running a similar ightweight stack.*
- Contiki provides another lightweight layered protocol stack, called Rime, for network-based communication. Rime provides single hop unicast, single hop broadcast, and multi-hop communication support. Rime supports both best effort and reliable transmission.

ContikiOS

Resource Sharing

Since events run to completion and Contiki does not allow interrupt handlers to post new events, Contiki provides serialized access to all resources.

Support for Real-Time Applications

Contiki does not provide any support for real-time applications, hence there is no implementation of any real-time process scheduling algorithm in Contiki.

ContikiOS

Additional features Security Support

Contiki does not provide support for secure communication. A proposal and implementation of a secure communication protocol with the name ContikiSec has been provided.

Simulation Support

Contiki provides sensor network simulations through Cooja

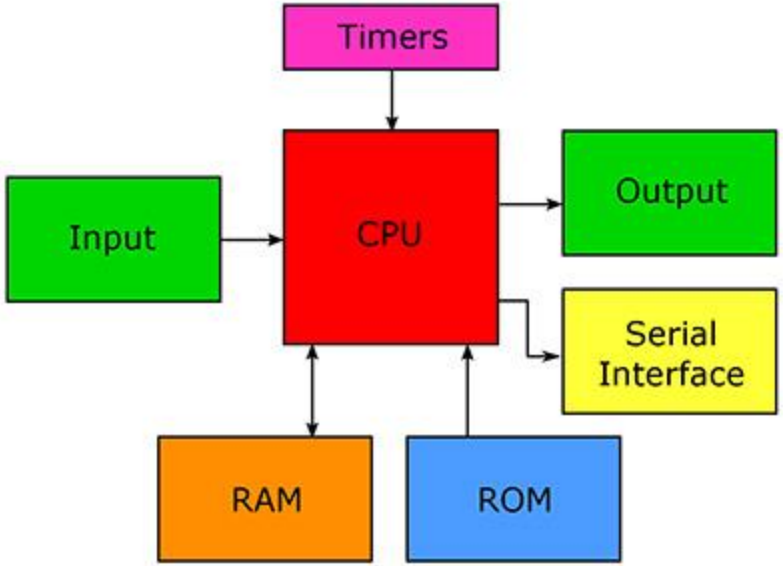
Language Support

Cotiki supports application development in the C language.

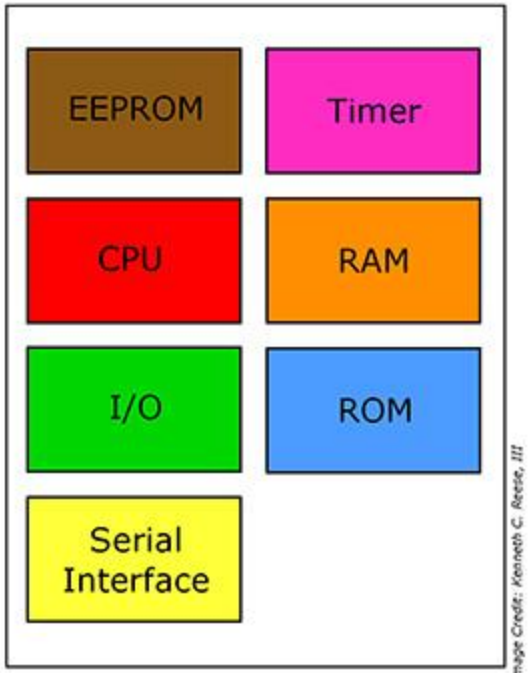
Supported Platforms

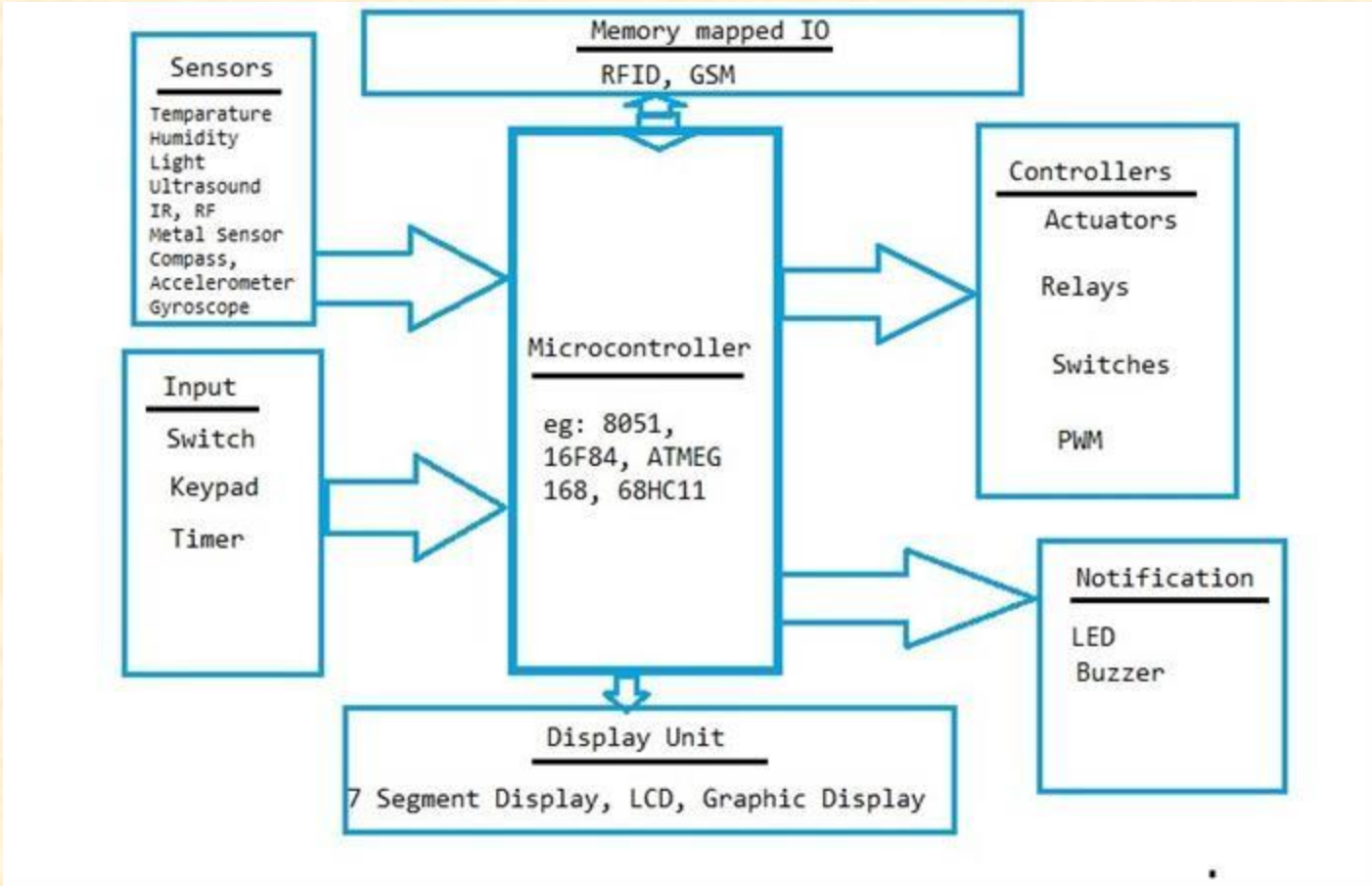
Contiki supports the following sensing platforms: Tmote AVR series MCU/

Microprocesser: CPU
and several supporting chips.



Microcontroller: CPU
on a single chip.




Memory mapped IO
RFID, GSM
Sensors
Temparature
Humidity
Light
Ultrasound
IR, RF
Metal Sensor
Compass,
Accelerometer
Gyroscope
Microcontroller
eg: 8051,
16F84, ATMEG
168, 68HC11
Controllers
Actuators
Relays
Switches
PWM
Input
Switch
Keypad
Timer
Notification
LED
Buzzer
Display Unit
7 Segment Display, LCD, Graphic Display

Thank You