

# Nepathya IOT AND Robotics Union

**Thing**  
of networks that can share information.

# Thing

of networks that can share information.

# Thing

of networks that can share information.

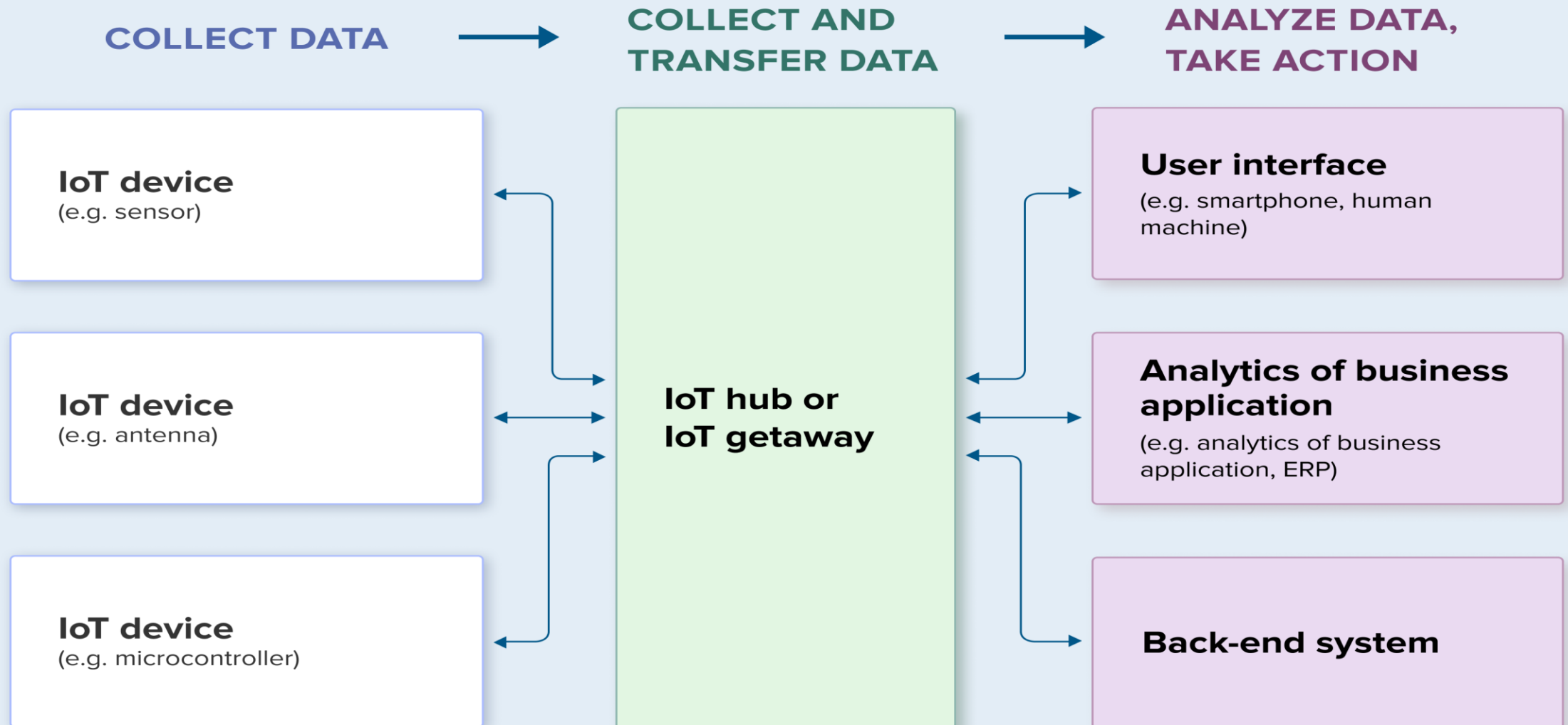
# Thing

of networks that can share information.

# Thing

of networks that can share information.

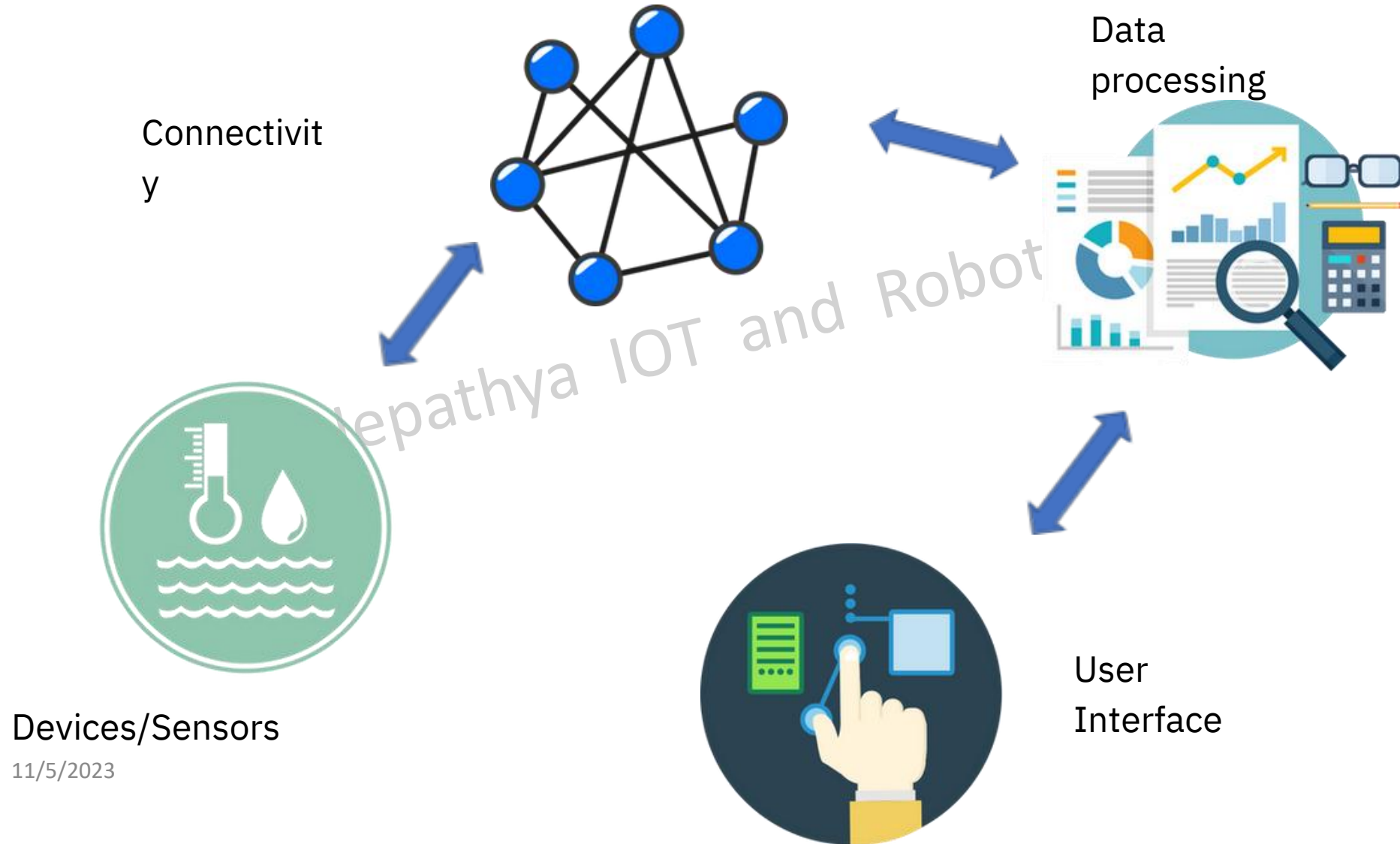
# Example of an IoT system



# Simple Example of IoT- Automatic Water Plant



# Components of IoT

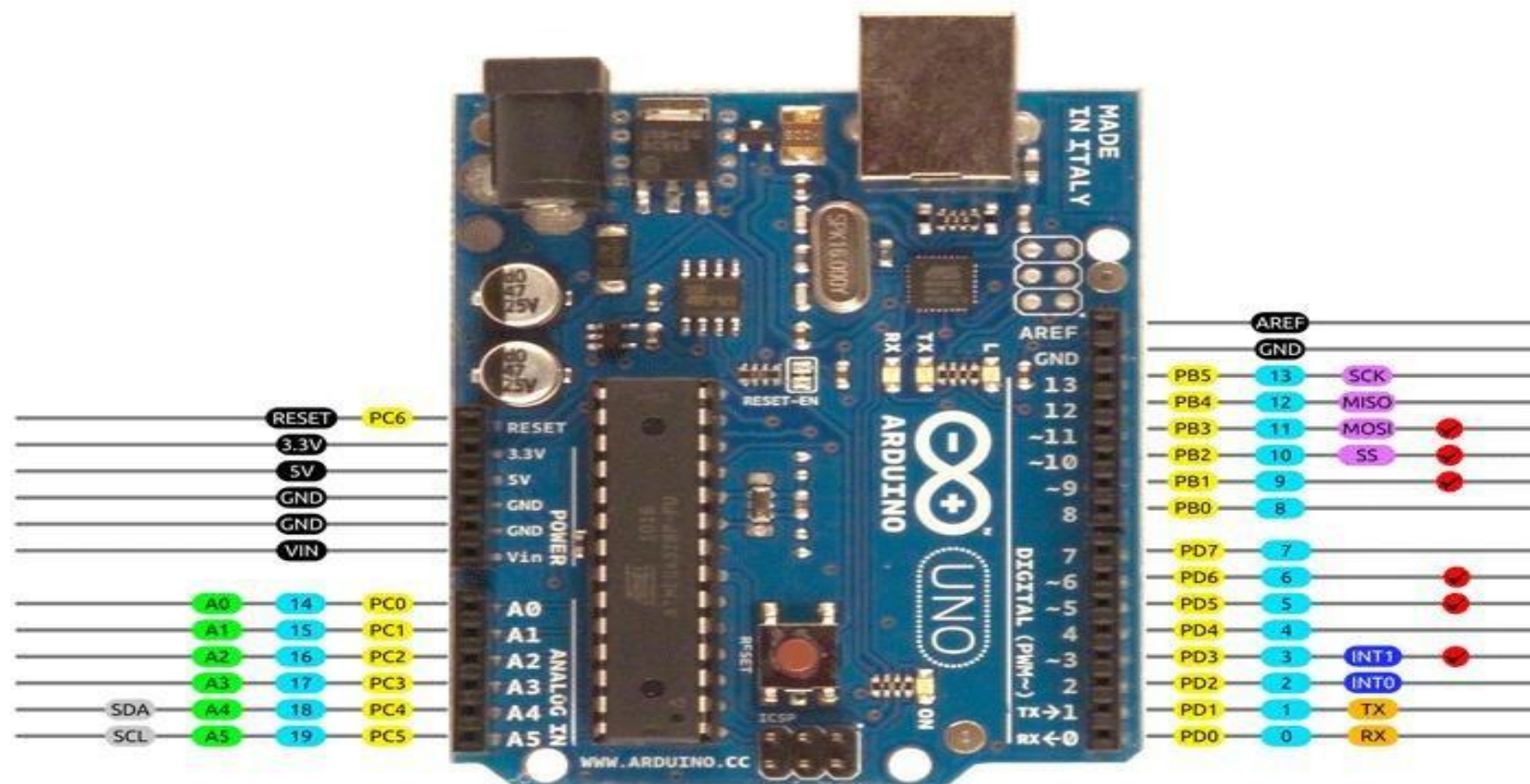


Devices/Sensors

11/5/2023



# Arduino Uno Pin Diagram



AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT



2014 by Bouni  
Photo by Arduino.cc

# Arduino Uno Specifications

Feature	Specification
<b>Microcontroller</b>	ATmega328P
<b>Operating Voltage</b>	5V
<b>Input Voltage (recommended)</b>	7-12V
<b>Input Voltage (limit)</b>	6-20V
<b>Digital I/O Pins</b>	14 (6 PWM pins)
<b>Analog Input Pins</b>	6
<b>PWM Output Pins</b>	6 (Pins: 3, 5, 6, 9, 10, 11)
<b>Clock Speed</b>	16 MHz
<b>Flash Memory</b>	32 KB (2 KB used by bootloader)
<b>SRAM</b>	2 KB
<b>EEPROM</b>	1 KB
<b>USB Interface</b>	USB-B Connector
<b>Communication Interfaces</b>	UART, I2C, SPI
<b>Current per I/O pin</b>	20 mA (max 40 mA)
<b>Power Pins</b>	3.3V, 5V, GND, Vin

# Components of Arduino Uno

## (a) ATmega328P Microcontroller

- The brain of the board, responsible for executing programs.
- Runs at **16 MHz** clock speed.

## (b) Digital I/O Pins (0-13)

- Used to read digital inputs (buttons, sensors) and control digital outputs (LEDs, motors, relays).
- 6 PWM pins (3, 5, 6, 9, 10, 11)** allow simulation of analog signals.

## (c) Analog Input Pins (A0-A5)

- Reads **analog signals** from sensors like temperature, LDR, and potentiometers.
- Each pin provides **10-bit resolution** (values from 0-1023).

## (d) Power Pins

- Vin**: External power supply (7V-12V).
- 5V**: Provides regulated 5V for sensors and modules.
- 3.3V**: Provides 3.3V for low-voltage components.
- GND**: Common ground.



### **(e) USB Port (Type B)**

- Used for programming and serial communication.
- Can power the board via USB (5V).

### **(f) ICSP Header**

- Used for programming the ATmega328P directly with an external programmer.

### **(g) Reset Button**

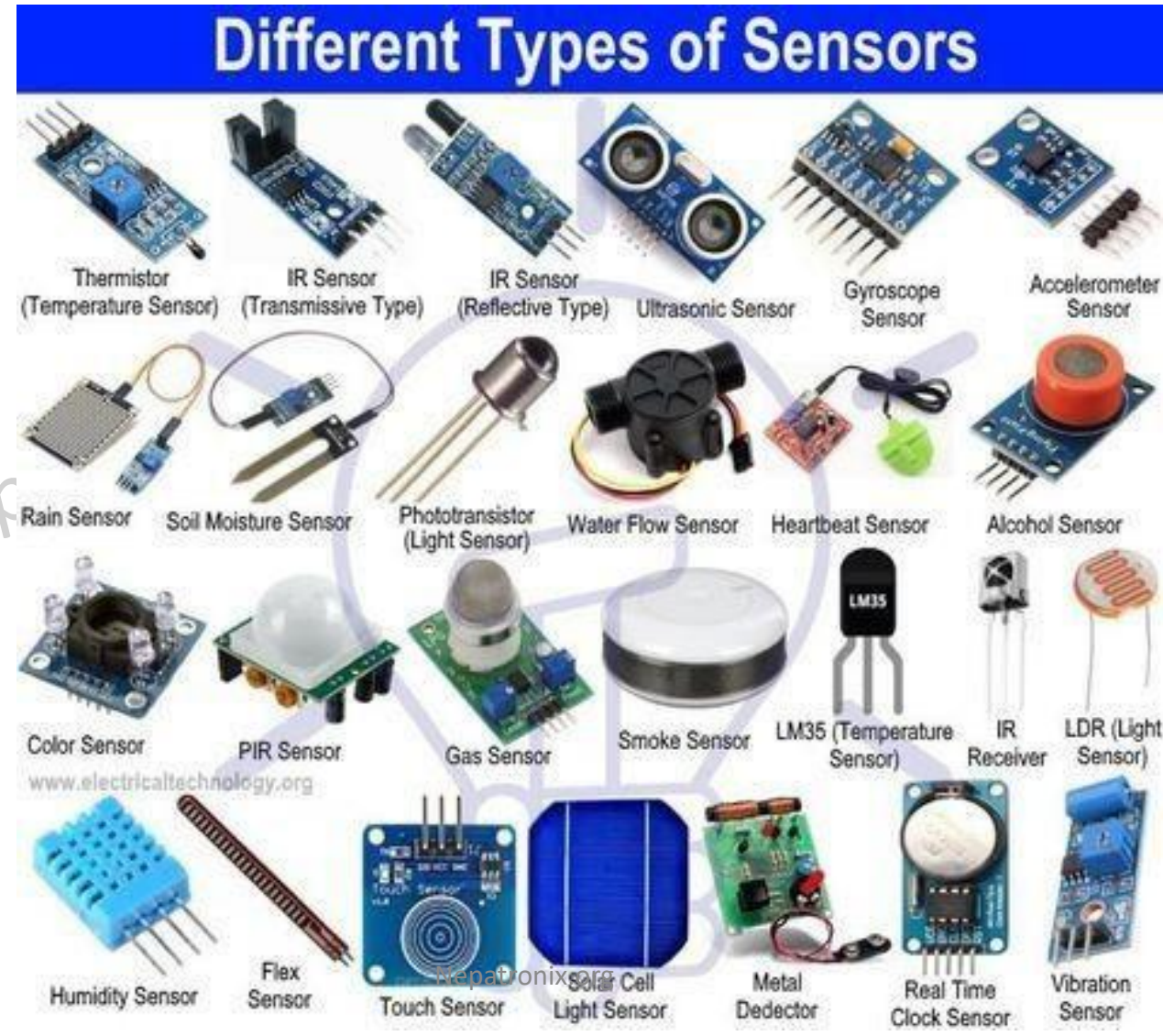
- Restarts the program execution.

### **(h) LED Indicators**

- Power LED (ON)** – Lights up when powered.
- TX/RX LEDs** – Blink during serial communication.

# Devices/Sensors

- Accelerometers
- Magnetometers
- temperature sensors
- proximity sensors
- light sensors
- gas RFID sensors
- humidity sensors
- pressure sensors
- acoustic sensors



# User Interface

The information processed is made available to the end user in some way, like giving Alert, Notification, Monitoring continuous feed or controlling the system remotely.



**Alerts**



**Notification**

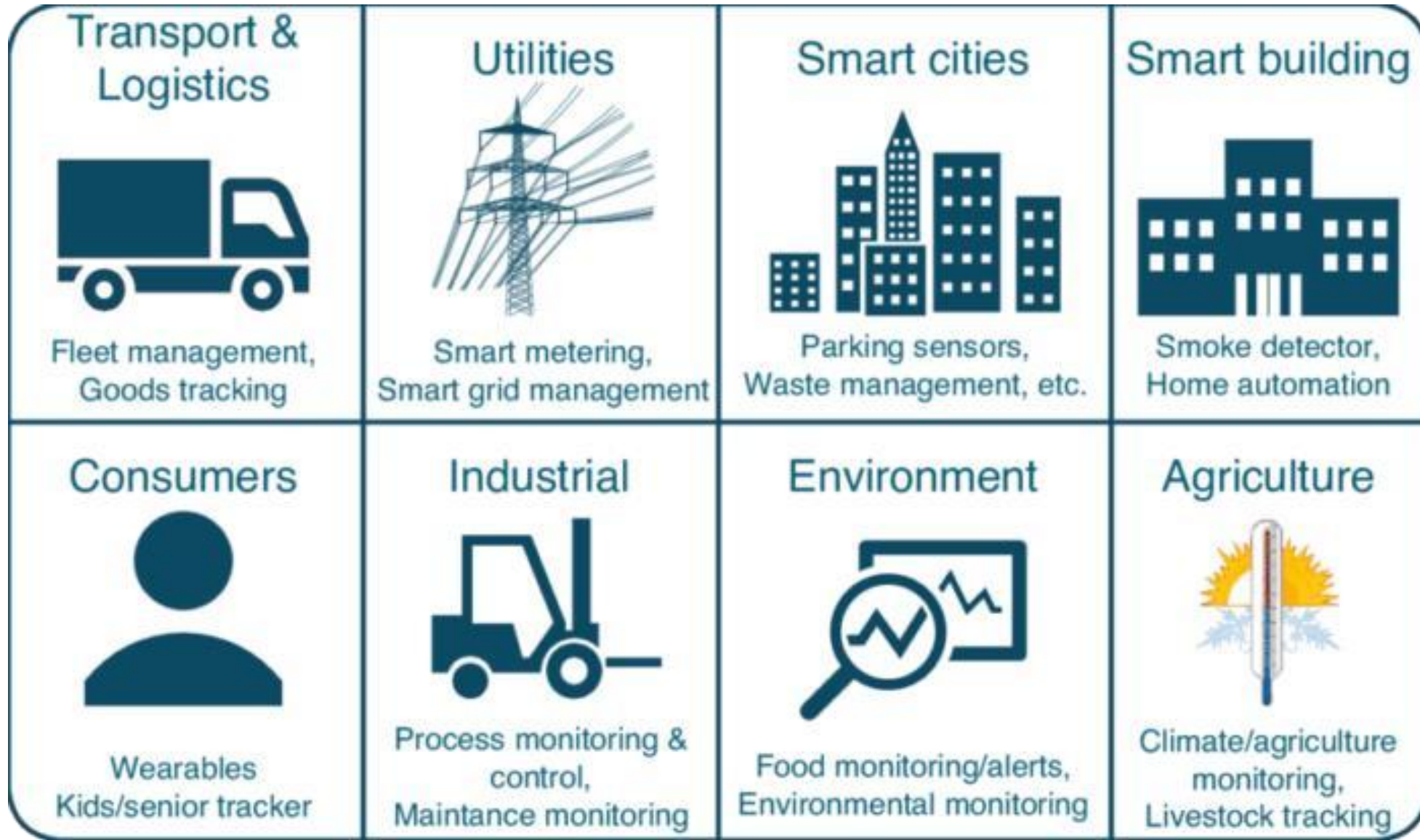


**Live Trends of data**

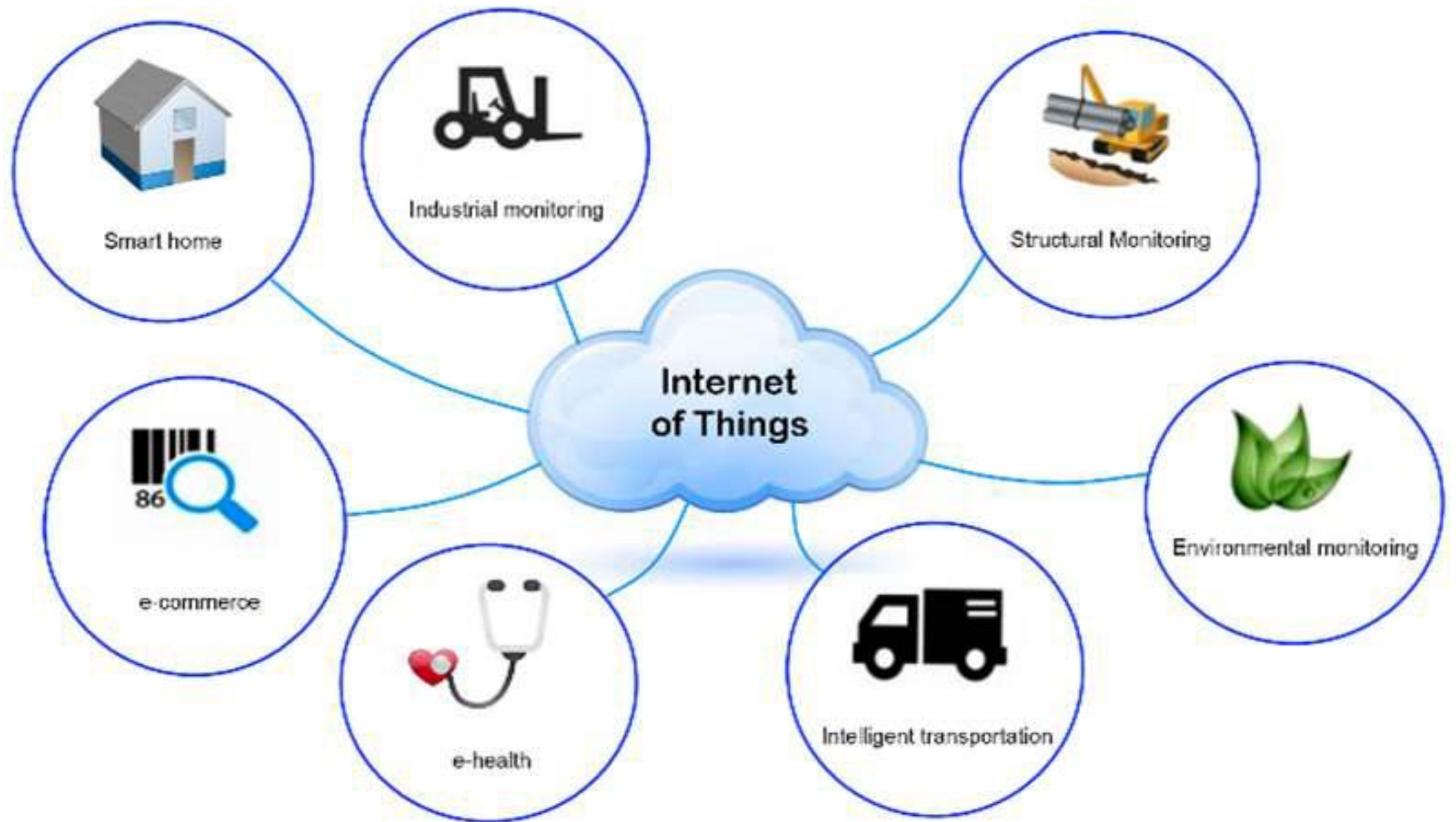


**Remote Control**

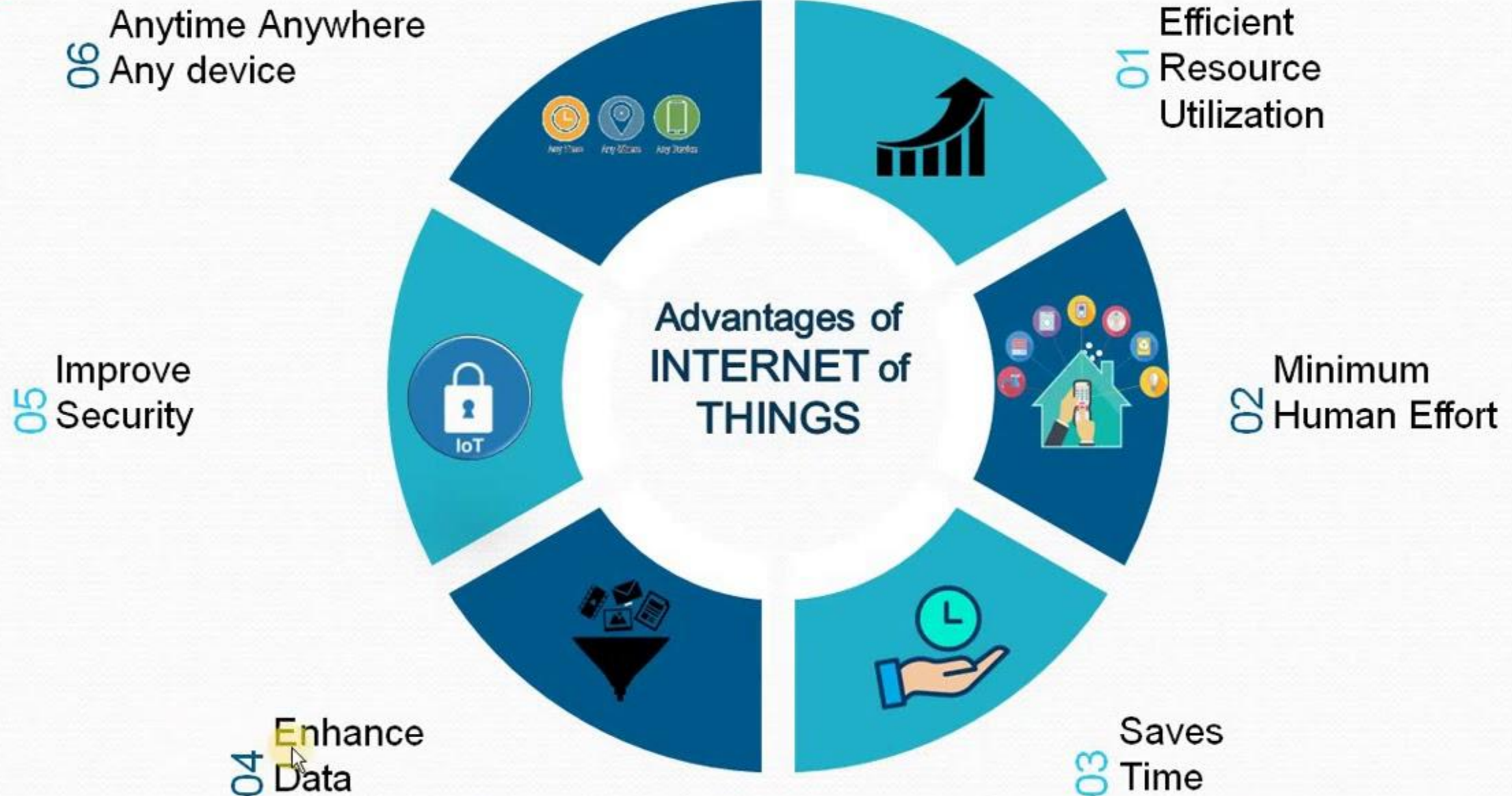
# IoT Applications







# Advantages of INTERNET of THINGS

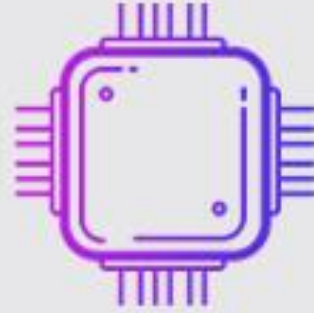




**Data Breach**



**Dependence**



**Complexity**



**Privacy  
Issues**

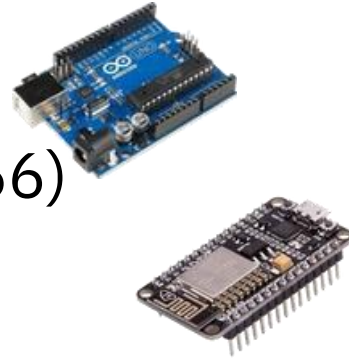
## **Disadvantages Internet of Things**



# IoT Hardwares requires

## Microcontrollers

1. Arduino Uno
2. NodeMCU( ESP8266)
3. ESP32 Dev Kit



## Software

1. Arduino IDE



## Analytic Platforms

1. ThingSpeak
2. bylnk



## Sensors

✓ *Based on your projects*

# IoT Projects:

- 1.Smart Agriculture System.
- 2.Smart Water level Monitoring
- 3.Smart Home
- 4.Smart Health Monitoring System
- 5.Weather data reporting
- 6.Smart Air Pollution Monitoring
- 7.Smart dustbin
- 8.RFID lock
- 9.Smart Street Light
- 10.Fire Alarm
- 11.Gas leakage Detection System
- 12.Laser light security

Nepathya IOT and Robotics Union

# Arduino IDE sketch



```
BareMinimum.ino
1 void setup() {
2   // put your setup code here, to run once:
3 }
4
5 void loop() {
6   // put your main code here, to run repeatedly:
7 }
8
```

Ln 8, Col 1 UTF-8 Arduino Uno on /dev/cu.usbmodem13401

# Data Types

## ➤ **int (Integer):**

**int** stands for "integer," and it is used to store whole numbers. It is typically a 16-bit data type on most Arduino boards, which means it can store integer values from -32,768 to 32,767.

Example: **int age = 25;**

## ➤ **float (Floating-Point):**

**float** is used to store decimal numbers or floating-point numbers. • It is represented as a 32-bit data type, allowing for a wide range of real numbers with fractional parts. • Example: **float**

**temperature = 25.5;**

# Data Types

## ➤ char (Character):

- **char** is used to store a single character.
- It's a 8-bit data type that can hold a single letter, number, or special character.
- Example: **char grade = 'A';**

## ➤ String: •String is a data type used to store text or a sequence of characters.

- It's a part of the Arduino String library, which provides various methods for working with strings.
- Example: **String name = "John";**

# Variable

A variable is a fundamental concept in programming that represents a storage location in memory where you can store and manipulate data. Variables allow you to assign a name to a specific memory location, making it easier to work with data in your programs.

`int age = 25;` // Declaring an integer variable named 'age' and assigning it the value 25.

`float temperature = 25.5;` // Declaring a floating-point variable named 'temperature' with the value 25.5.

`char grade = 'A';` // Declaring a character variable named 'grade' with the value 'A'.

`String name = "John";` // Declaring a String variable named 'name' with the value "John".

# Basic Commands in Arduino

## 1. `pinMode(pin, mode)`:

- Used to configure a specific digital pin as an INPUT or OUTPUT.
- Example: `pinMode(13, OUTPUT)`; sets pin 13 as an output.

## 2. `digitalWrite(pin, value)`:

- Sets the state of a digital pin to HIGH or LOW (1 or 0).
- Example: `digitalWrite(13, HIGH)`; //turns on the LED connected to pin 13.

## 3. `digitalRead(pin)`:

- Reads the digital state of a pin and returns HIGH or LOW.
- Example: `int buttonState = digitalRead(2)`;

//reads the state of pin 2 and stores it in buttonState.



# Basic Commands in Arduino

## 4. `analogWrite(pin,value)`:

- Used to output a PWM (Pulse Width Modulation) signal on a digital pin, allowing for analog-like control of devices like servos and LEDs.
- Example: `analogWrite(9, 128);` //sets the PWM value on pin 9 to 50% duty cycle.

## 5. `analogRead(pin)`:

- Reads an analog voltage on a pin and returns a value between 0 and 1023 (for a 10-bit analog-to-digital converter).
- Example:`int sensorValue = analogRead(A0);` // reads the analog voltage on pin A0 and stores it in `sensorValue`

## 6. `delay(ms)`:

- Pauses the program for the specified number of milliseconds (ms).
- Example: `delay(1000);` //pauses the program for 1 second.



Thank You!!

thya IOT and Robotics Union