Developing IoTs:
Introduction to different IoT tools,
developing applications through IoT tools,
developing sensor based application through embedded system platform,
Implementing IoT concepts with python.
Edge computing, difference between fog computing,
edge computing & IoT,
prototyping the embedded devices for IOT.

Unit V

Developing IoTs

Introduction to IoT Tools

IoT development means combining hardware parts and software programs in such a way that the final product can monitor specific values, collect and transfer data, analyze given data and cause the physical device to act correspondingly. Creating such systems is a true challenge.

Moreover, the Internet of Things has already been transformed into an industry in its own right, so the need for reliable and comprehensive developer toolkits has also increased. IoT development tools needed to create complex applications are represented by IoT hardware devices (boards, SoM, SoC, sensors, gateways, trackers, and more), IoT app development platforms, IoT operating systems (e.g., Embedded Linux) and programming languages.

Top IoT Tools and Devices

- Arduino
- Flutter
- Kinoma
- M2MLabs Mainspring
- Raspberry Pi
- Eclipse IoT

Flutter

Another hardware product for IoT solutions is Flutter — a programmable processor core. The board is based on Arduino, has a powerful ARM processor, built-in battery charging and a security chip. A long-range wireless transmitter makes this board the perfect fit for wireless networks of sensors.

Flutter offers:

- the Basic control module
- the Pro control module
- complete kits (Quick Start Kit, Vehicle Control Kits)
- accessory boards (the Bluetooth adapter, the Breakout, the Remote Control, the Explorer)
- a solar panel, a cylindrical battery and other accessories
- 3D-printed parts for your device.
-

Kinoma

A group of software engineers from Marvell Technology, Inc., a leading manufacturer of memory devices, microcontrollers, telecom equipment and semiconductor devices, has developed a line of open-source **Kinoma software and hardware** products for the Internet of Things and embedded solutions.

Kinoma Create — a hardware platform for prototyping IoT devices. It's powered by JavaScript and has an integrated SD card, speaker, microphone, Bluetooth and Wi-Fi. It enables the interaction of devices with cloud platforms via the WebSocket internet protocol.

Kinoma Studio — an integrated development environment (IDE) that provides the hardware simulator and sensor library, allowing developers to create robust applications.

Kinoma Connect — an application for Android and iOS supporting IoT devices.

M2MLabs Mainspring

The development of machine-to-machine applications is easy with the M2MLabs Mainspring framework. This open-source Java-based framework is widely used for building fleet management apps and remote monitoring projects. It enables flexible device configuration and supports the reliable connection between machines. App prototyping is very quick with M2MLabs Mainspring. Moreover, it ensures long-term data storage and retrieval thanks to a scalable Apache Cassandra database.

Raspberry Pi: “Pi is a single-board computer”. Pi is a small scale computer in the size little bigger than a credit card, it packs enough power to run games, word processor like open office, image editor like Gimp and any program of similar magnitude.

Pi was introduced as an educational gadget to be used for protyping by hobbyists and for those who want to learn more about programming. It certainly cannot be a substitute for our day to day Linux, Mac or Windows PC.

Pi is based on a Broadcom SoC (System of Chip) with an ARM processor [~700 MHz], a GPU and 256 to 512 MB RAM. The boot media is an SD card [which is not included], and the SD card can also be used for persist data. Now that you know that the RAM and processing power are not nearly close to the power house machines you might have at home, these Pi's can be used as a Cheap computer for some basic functions, especially for experiments and education. The Pi comes in three Configurations and we will discuss the specifications of those in the coming sections. The cost of a Pi is around $35 for a B Model and is available through many online and physical stores.

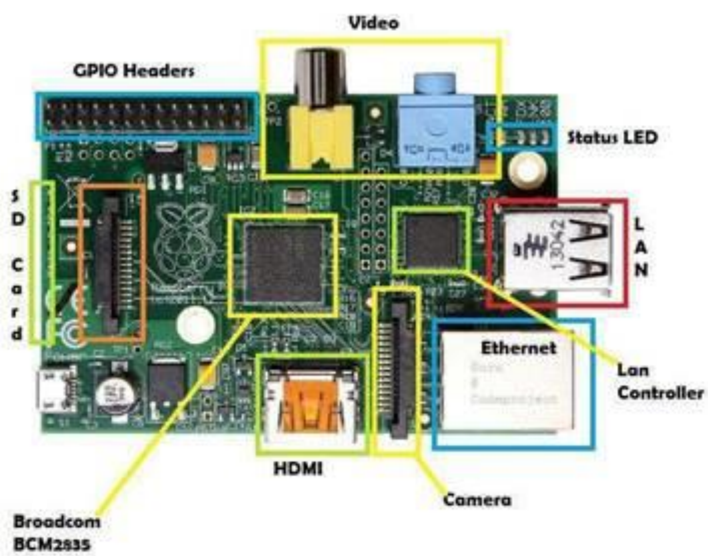


Fig Raspberry Pi Board

2. System Specifications

Computer	A Raspberry Pi

Storage	SD Card and a SD card reader to image the OS [These days laptops have inbuilt card readers]
Power supply	5 volt micro USB adapter, mostly your android phone charger would work

Display	An TV/Monitor with DVI or HDMI port
Display connector	HDMI cable or HDMI to DVI converter cable
Input	USB Mouse
Input	USB Keyboard
Network	Ethernet cable
Case	If you really need one, you can get them online based on the model you have

Raspberry Pi uses

It can be used in following areas/applications:

1. Retro Gaming
2. Raspberry Pi Tablet
3. Low-Cost Desktop PC
4. Raspberry Pi Cluster

5. Raspberry Pi Cloud Server

6. Raspberry Pi Media Center

7. Web Server

8. Home Automation System

9. VPN

10. Robotics

Arduino devices

1. Introduction

Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself (DIY) kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler tool chains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

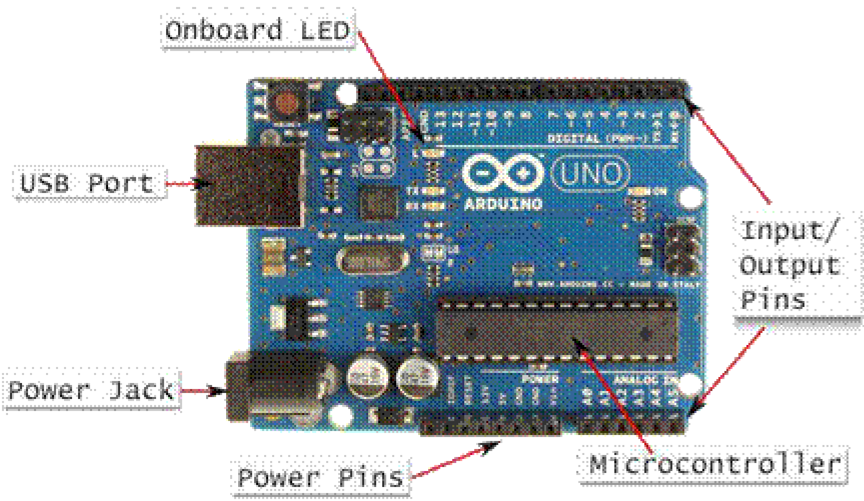


Fig Arduino

System Specifications

Component	Specification
Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6

DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Arduino uses

It can be used in following areas/applications:

1. Tiny Weather display
2. MIDI Controller
3. Fingerprint Scanner to Your Garage Door Opener
4. Auto-Trigger Spray Gun
5. Make Your Own Arduino
6. Add Motion-Triggered Night Lights Under the Bed
7. Mute Any Phrase You Want on Your TV

8. Add an Ambilight Sensor to Your LCD Display

9. Build a Robot Arm

10. Create a Fully Functional Computer Control Panel

Smart City Street lights control & monitoring.

Introduction-Automate street lights are necessary while we are trying to survive in the era of smart world. Automation provides perfection and efficiency. The goal is to automate street lighting, as current system is facing many problems. A user has to deal with numerous problems like maintenance problem, timer problem, connectivity problem, display problem. The solution to these problems is IoT Based Street Lights.

street lights are one of the main city's assets which provide safe roads, inviting public areas, and enhanced security in homes, businesses, and city centers. As they use in average 40% of a city's electricity spending which leads to power consumption.

Following are the issues of existing electric system. Connectivity issue-In existing system, connections of street light are done manually. As each connection requires different contractors and if any one of them is not available then it will leads to functionality problem of street lights.

1. Timer Problem-Contractors needs to manage timer settings manually. As timer requires twelve hour of continuous electricity supply, and if in case it is not available, it will delay further timer settings.

2. Maintenance problem-If any of the streets light gets failed or any problem occurs, it's not resolved immediately.

3. Incorrect Readings-Sometimes exact readings are not shown on to the display. So we cannot conclude how much energy is being consumed which give rise in high billing.

Streetlights are among a city's strategic assets providing safe roads, inviting public areas, and enhanced security in homes, businesses, and city centers. However, they are usually very costly to operate, and they use in average 40% of a city's electricity spending.

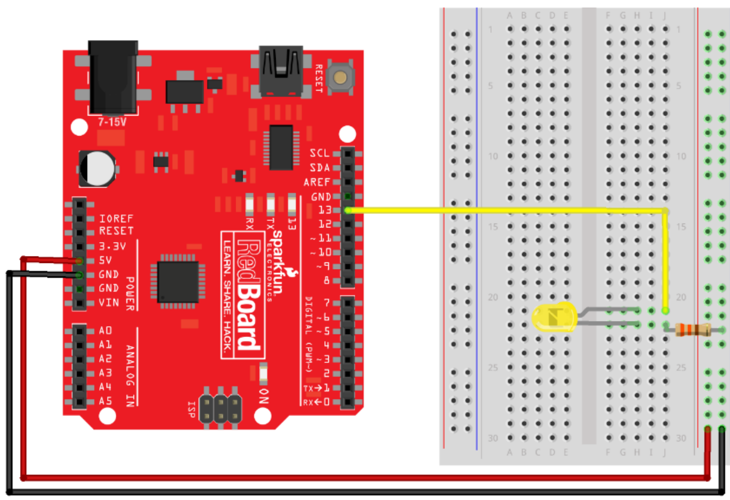
Controlling an LED with Python

In this section, you will learn how to control an LED connected to an external piece of hardware (an Arduino) using Python. To accomplish this task, the following hardware is required:

- A computer running Python
- An Arduino
- An LED
- Wires, a resistor and a breadboard to connect the LED to the Arduino
- A USB cable to connect the Arduino to the computer

Wire the LED to the Arduino

Connect the LED to the Arduino using a resistor, wires and a breadboard. Note the short leg of the LED is connected to ground, and the long leg of the resistor is connected through a resistor to PIN 13. A resistor is needed to prevent too much current from flowing through the LED. This type of resistor is called a pull up resistor.



■ Box 7.1: Switching LED on/off from Raspberry Pi console

```
$echo 18 > /sys/class/gpio/export
$cd /sys/class/gpio/gpio18

#Set pin 18 direction to out
$echo out > direction

#Turn LED on
$echo 1 > value
#Turn LED off
$echo 0 > value
```

■ Box 7.2: Python program for blinking LED

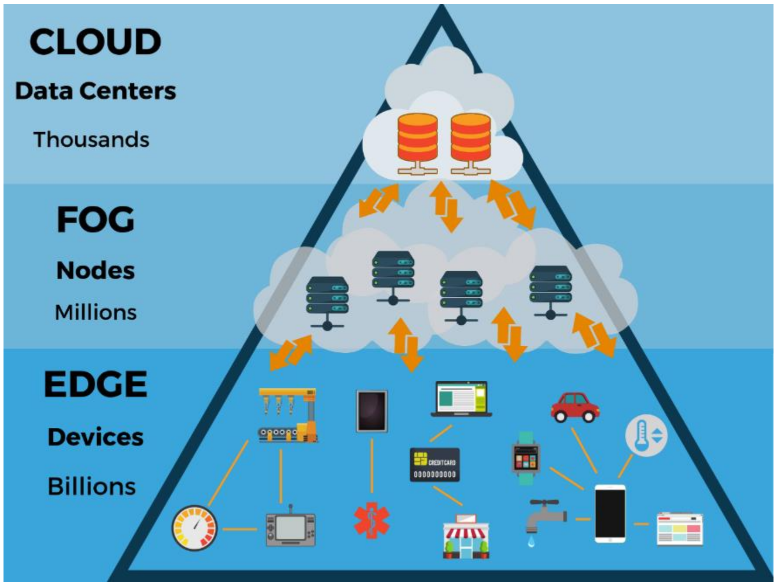
```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(18, GPIO.OUT)

while True:
   GPIO.output(18, True)
   time.sleep(1)
   GPIO.output(18, False)
   time.sleep(1)
```

Edge computing

Computation takes place at the edge of a device's network, which is known as edge computing. That means a computer is connected with the network of the device, which processes the data and sends the data to the cloud in real-time. That computer is known as "edge computer" or "edge node". Edge computing refers to the enabling technologies allowing computation to be performed at the edge of the network, on downstream data on behalf of cloud services and upstream data on behalf of IoT services. Here we define "edge" as any computing and network resources along the path between data sources and cloud data centers. For example, a smart phone is the edge between body things and cloud, a gateway in a smart home is the edge between home things and cloud, a micro data center and a cloudlet is the edge between a mobile device and cloud. The rationale of edge computing is that computing should happen at the proximity of data sources. From our point of view, edge computing is interchangeable with fog computing, but edge computing focus more toward the things side, while fog computing focus more on the infrastructure side. We envision that edge computing could have as big an impact on our society as has the cloud computing. Fig. illustrates the two-way computing streams in edge computing. In the edge computing paradigm, the things not only are data consumers, but also play as data producers. At the edge, the things can not only request service and content from the cloud but also perform the computing tasks from the cloud. Edge can perform computing offloading, data storage, caching and processing, as well as distribute request and delivery service from cloud to user. With those jobs in the network, the edge itself needs to be well designed to meet the requirement efficiently in service such as reliability, security, and privacy protection. Edge Computing Benefits In edge computing we want to put the computing at the proximity of data sources. This have several benefits compared to traditional cloud-based computing paradigm. Here we use several early results from the community to demonstrate the potential benefits. Researchers built a proof-of-concept platform to run face recognition application in, and the response time is reduced from 900 to 169 ms by moving computation from cloud to the edge. clouds to offload computing tasks for wearable cognitive assistance, and the result shows that the improvement of response time is between 80 and 200ms. Moreover, the energy consumption could also be reduced by 30%–40% by cloudlet offloading.



Edge computing is decentralized in nature and demands high levels of monitoring and control. Edge offers an effective solution to emerging network problems associated with transmitting large volumes of data that IoT and IIoT corporations produce. However, edge computing does not have the compute and storage that is necessary to perform advanced analytics. While cloud computing can perform advanced analytics, it is too far away from the source to process and respond in a real-time manner.

Difference Between Edge Computing and Fog Computing

S.NO.	EDGE COMPUTING	FOG COMPUTING
01.	Less scalable than fog computing.	Highly scalable when compared to edge computing.
02.	Billions of nodes are present.	Millions of nodes are present.
03.	Nodes are installed far away from the cloud.	Nodes in this computing are installed closer to the cloud(remote database where data is stored).
04.	Edge computing is a subdivision of fog computing.	Fog computing is a subdivision of cloud computing.
05.	The bandwidth requirement is very low. Because data comes from the edge nodes themselves.	The bandwidth requirement is high. Data originating from edge nodes is transferred to the cloud.
06.	Operational cost is higher.	Operational cost is comparatively lower.
07.	High privacy. Attacks on data are very low.	The probability of data attacks is higher.
08.	Edge devices are the inclusion of the IoT devices or client’s network.	Fog is an extended layer of cloud.

S.NO.	EDGE COMPUTING	FOG COMPUTING
09.	The power consumption of nodes is low.	The power consumption of nodes filter important information from the massive amount of data collected from the device and saves it in the filter high.

IOT Prototyping

It is the process of building IoT hardware and devices enhanced with smart sensors and embedded systems using many off-the-shelf components like sensors, circuit boards, and microcontrollers. A lot of these off-the-shelf solutions are readily available to end consumers. Also, a prototype is by no means a market-ready product. It is just a trial version of your connected solution and acts as proof that your innovative idea will work the way you envision it.

Smart cities

IoT has all it takes to improve the quality of urban life and the experience of its city dwellers. Increasingly, smart cities across the world use IoT to resolve issues with traffic and transportation, energy and waste management, etc.

Platforms help smart cities become more energy-efficient. The solution also enables them to control surveillance cameras, wi-fi coverage, electronic billboards, and other mission-critical devices like environmental sensors and charging stations. Smart cities also use IoT for infrastructure management: controlling the state of water supply and sewer systems, street lighting, waste reduction, garbage collection, etc.

However, among the most advantageous use cases for urban IoT solutions is smart parking. Each year, the number of vehicles grows exponentially, and modern technology aims to curtail traffic congestion, manage city parking wisely, and even cut emissions.

Agriculture

Smart farming is often overlooked when it comes to the business cases for IoT solutions. However, there are many innovative products on the market geared toward forward-thinking farmers.

Some of them use a distributed network of smart sensors to monitor various natural conditions, such as humidity, air temperature, and soil quality. Others are used to automate irrigation systems. One such example of IoT devices, Blossom, offers both. This smart watering system uses real-time weather data and forecasts to create an optimal watering schedule for your yard. Consisting of a smart Bluetooth-powered controller and a mobile app, the system is easy to install, setup, and manage. While the product is initially designed for use at home, similar solutions can also be applied to larger scales.

Internet of Things benefits in agriculture:

- crop, climate, and soil condition monitoring;
- livestock monitoring;
- precision farming;
- watering and fertilization automation;
- automating detection and eradication of pests;
- greenhouse automation; and
- higher crop quality and better yields.

Note: Study some practical performed in IOT lab