PROBLEM STATEMENT :

In a rolling display program of news display on a smart TV or Computer Display the input strings are supplied by another computer connected through wireless networks.Develop necessary app using Scala/Python/ Java/ C++.

OBJECTIVE :

- To develop problem solving abilities for smart devices.
- To develop problem solving abilities for gamifications.
- To develop problem solving abilities of pervasiveness,embedded security and NLP.
- To study algorithmic examples in distributed, concurrent and parallel environments

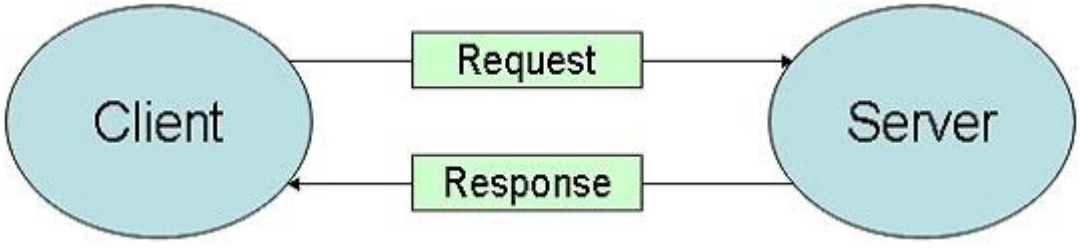
THEORY :

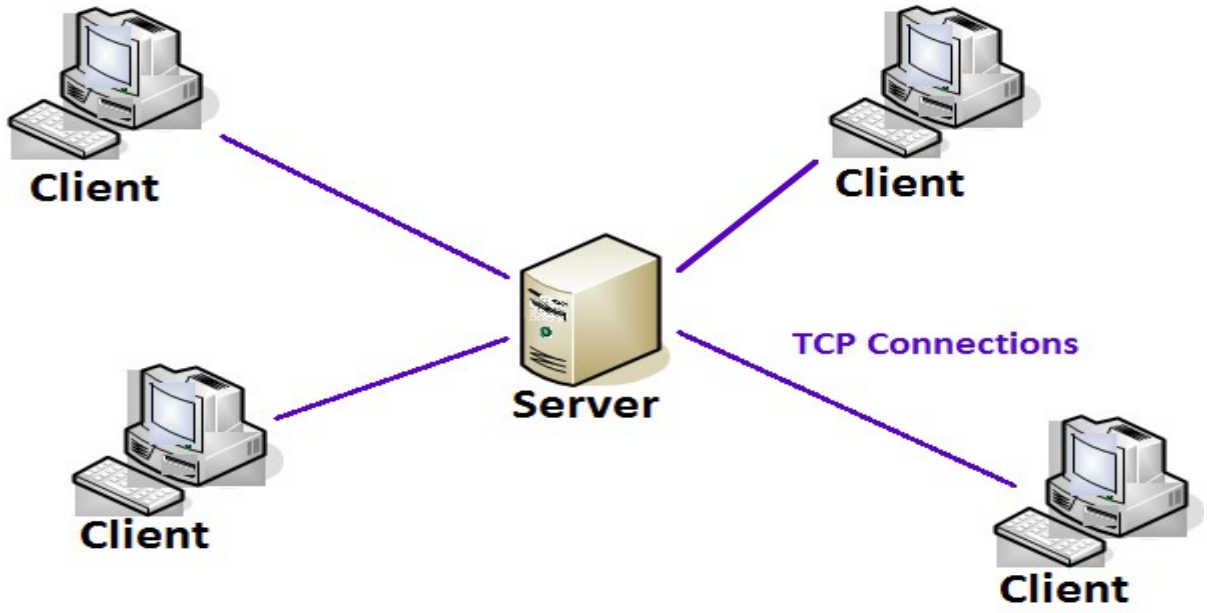
Client – Server Architecture :

Client/server architecture is a computing model in which the server hosts, delivers and manages most of the resources and services to be consumed by the client. This type of architecture has one or more client computers connected to a central server over a network or Internet connection. Client/server architecture may also be referred to as a networking computing model because all the requests and services are delivered over a network.

Client/server architecture is a producer-consumer computing architecture where the server acts as the producer and the client as a consumer. The server houses and provides high-end, computing-intensive services to the client on demand. These services can include applications access, storage, file sharing, printer access and/or direct access to the server's raw computing power.

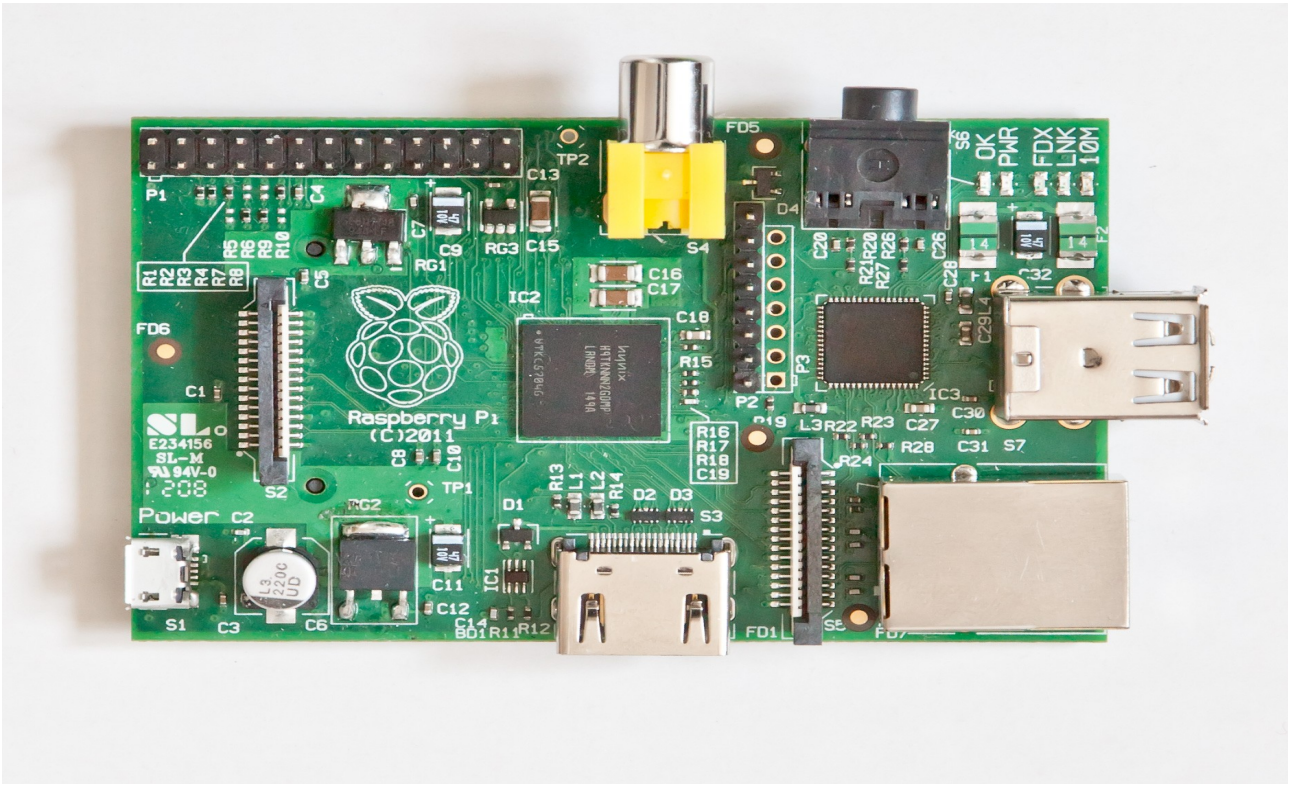
This architecture works when the client computer sends a resource or process request to the server over the network connection, which is then processed and delivered to the client. A server computer can manage several clients simultaneously, whereas one client can be connected to several servers at a time, each providing a different set of services. In its simplest form, the Internet is also based on client/server architecture where the Web server serves many simultaneous users with Web page and or website data.





RASPBERRY PI :

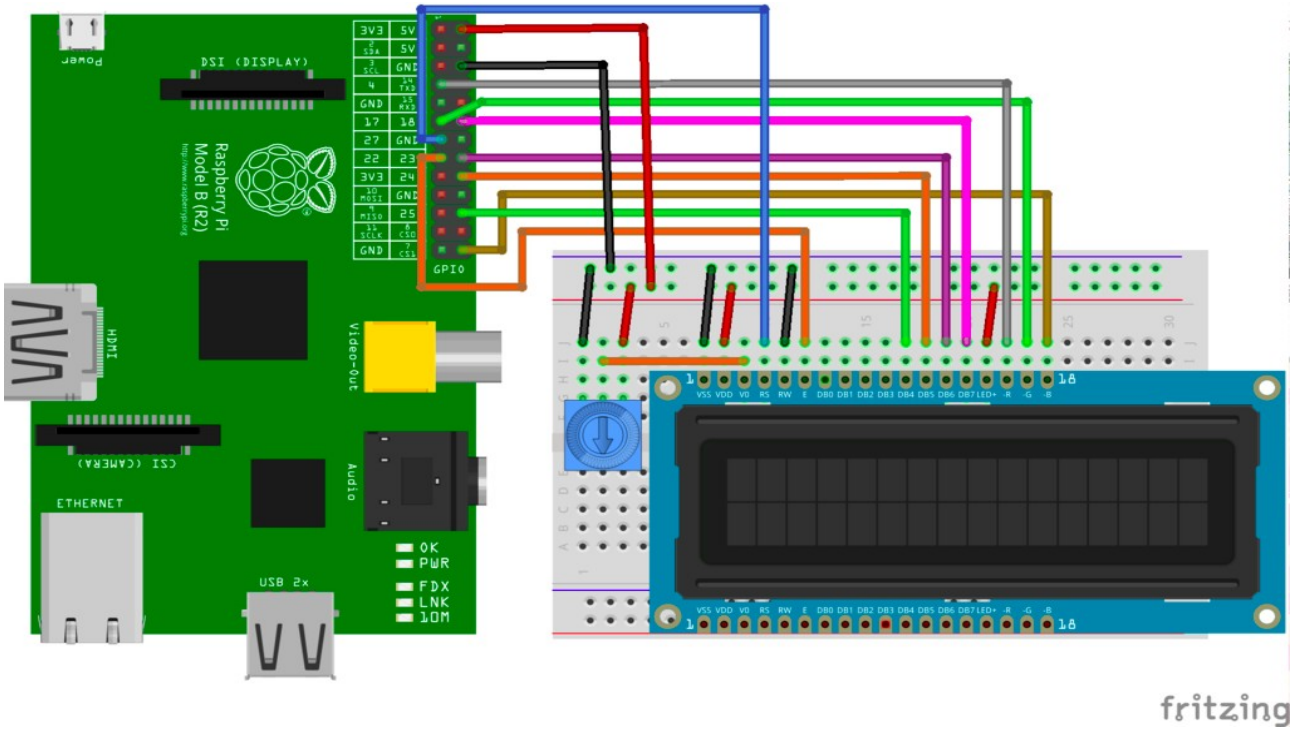
The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games**.** What's more, the Raspberry Pi has the ability to interact with the outside world, and has been used in a wide array of digital maker projects, from music machines and parent detectors to weather stations and tweeting birdhouses with infra-red cameras.



Raspberry PI Hardware General Specifications :

Feature	Model A	Model B	Model A+	Model B+
BRCM2835 SoC	Yes	Yes	Yes	Yes
Standard SoC Speed	700Mhz	700Mhz	700Mhz	700Mhz
RAM	256MB	512MB*	256MB	512MB
Storage	Full SD	Full SD	Micro SD	Micro SD
Ethernet 10/100	No	Yes	No	Yes
HDMI output port	Yes	Yes	Yes	Yes
Composite video output	Yes	Yes	On 3.5mm jack	On 3.5mm jack
Number of USB2.0 ports	1	2	1	4
Expansion header	26	26	40	40

INTERFACING DIAGRAM :



ALGORITHM :

1. Create a Server on Raspberry Pi .
2. Create a Client on another PC . The PC and Raspberry Pi are connected to each other wirelessly
3. Enter an Input string from the PC which sends it to the server .
4. The server receives the string and gives it to the LCD .
5. The input string is displayed in a rolling fashion on the LCD .

INPUT :

Enter input string from a PC .

EXPECTED OUTPUT :

Rolling display of the input string on the LCD .

MATHEMATICAL MODEL :

Let P be the solution perspective .

P ={ S , E , I , O, F }

S = { Initial state of the system consisting of the server listening for connections }

I = Input of the system → { I1 }

where I1 = { Input String }

O = Output of the system → { O1 }
where O1 = { Rolling display on the LCD }

F = Functions used → { f1 , f2 , f3 ,f4 , f5 }
where f1 = { listen() : Listen for incoming connections }
f2 = { connect() : Connect to the server }
f3 = { bind() : Bind a host to a particular port }
f4 = { send() : Send a mesage }
f5 = { receive() : Receive a mesage }

E = End state of the system showing rolling dislay of the input string on LCD on a wirelessly connected Raspberry Pi.

CONCLUSION :

Hence we have successfully implemented a rolling display program by sending the string from a wirelessly connected PC to another display device .

OUTCOMES ACHIEVED :

COURSE OUTCOME	**ACHIEVED(√)**
Problem solving abilities for smart devices.	√
Problem solving abilities for gamifications.	
Problem solving abilities of pervasiveness,embedded security and NLP.	√
To solve problems for multicore or distributed,concurrent/Parallel environments	