IJCTA, 9(33), 2016, pp. 121-129

The Design of Protocol Converter

A. Suvarnamma* N.veni* and Hannah Pauline*

Abstract : Traditional telecommunications systems are nearing the end of their product life cycles, and worldwide, operators are strategizing whether to repair / replace these systems or he add own a completely new and potentially more vi able path —migration towards an Internet Protocol (IP) based telecommunications systems. Currently, the status of telecom networks in India is a combination of the circuit switched (TDM infra structure) and packet switched (IP based switches) networks. Every device has its own communication protocol as pre-defined by its manufacturer. Hence, there is a need for the conversion of these protocols (working on different as specified bythe manufacturer) so as to establish an INTERCONNECTION between the two devices, by the help of which a communication medium between the devices is established. This asks for a need for a Protocol Converter, which would enable the data to be accessed and sent through / by different protocols.

Keywords : Converter, Internet, Protocol, Packet Switched.

1. INTRODUCTION

Modern digital technology allows different sectors / services viz. telecom, data, radio and television, to be merged together. This phenomenon, commonly known as convergence, is taking place on a global scale and is drastically changing the way in which both people and devices communicate. The back bones for making such convergence possible are IP-based networks. The opportunities presented by IP based networks are immense and will help the telecom service provider (TSPs) to converge their network infrastructures, provide huge bandwidth, consolidate terminating traffic and reduce long-distance charges. The IP based network integrates all telecommunications traffic of an operator in to one network. Due to the high proportion of fixed costs, the unit cost decreases as the traffic grows. IP-based systems are extremely flexible and can be easily modified as per the requirements and technological developments over time. Transporting telecommunications traffic as IP packets instead of reserving a complete channel for each voice call increases the network efficiency. End users will receive increased number of services with higher overall quality than that offered by traditional networks, while the price that they have to pay for those services reduces.

2. DESIGN STRATEGY / METHODOLOGY

After analyzing the current communication systems, it was seen that our communication system demands a rather more efficient and reliable mechanism so that the data (voice, information etc.) may be sent by a rather simple approach based on the conversion of protocol (As the device protocols are established by the manufacturers). The complete block diagram of the process is described here.

The PIC microcontroller emits serial data, which is given to the raspberry pi. The pi will accept the data and will convert the data into IP. The converted data is transmitted to server in UDP format. The server will accept data and save it in a database so that the data can be used for reference in future. Transmission

* Department of Electronics and Communication Engineering SRM University, Kattankulathur, 603203, India

of data to server is done wirelessly. We use UDP format initially because they are in same area, hence there is negligible loss. Data can be accessed by any controlling stations / client from the server. Data will be transmitted in TCP format to the clients. We use TCP because acknowledgements are necessary for transmission of data to keep minimal losses.

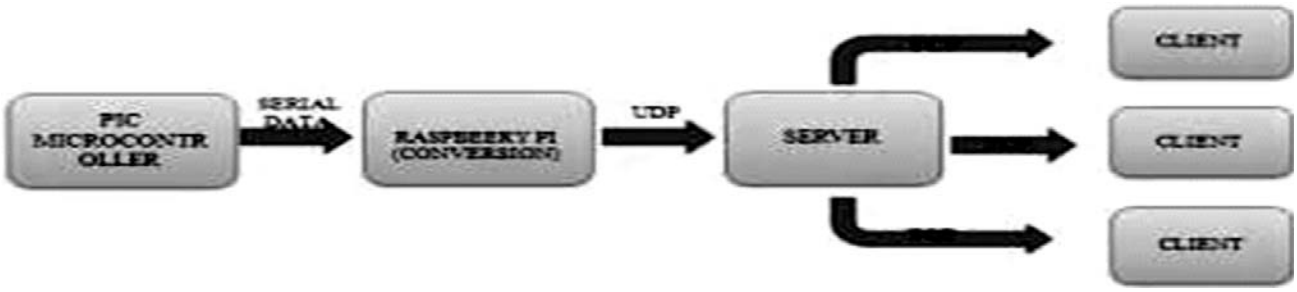

Figure 1

A. PIC Microcontroller

Peripheral Interface Controllers (PIC) is one of the advanced microcontrollers developed by microchip technologies. These microcontrollers are widely used i n modern electronics applications. APIC controller integrates all type of advanced interfacing ports and memory modules. These controllers are more advanced than normal microcontroller like INTEL 8051. The first PIC chip was announced in 1975 (PIC1650). As like normal microcontroller, the PIC chip also combines a microprocessor unit called CPU and is integrated with various types of memory modules (RAM, ROM, EEPROM, etc), I/O ports, timers / counters, communication ports, etc.

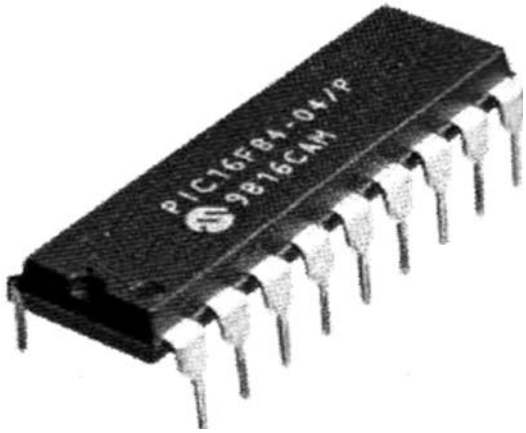

Figure 2

All PIC microcontroller family uses Harvard architecture. This architecture has the program and data accessed from separate memories so the device has a program memory bus and a data memory bus (more than 8 lines in a normal bus). This improves the bandwidth (data through put) over traditional von Neumann architecture where program and data are fetched from the same memory (accesses over the same bus). Separating program and data memory further allows instructions to be sized differently than the 8-bit wide data word. The function of CPU in PIC is same as a normal microcontroller CPU. APIC CPU consists of several sub units such as instruction decoder, ALU, accumulator, control unit, etc. The CPU in PIC normally supports Reduced Instruction Set Computer (RISC) architecture (Reduced Instruction Set Computer (RISC), a type of microprocessor that focuses on rapid and efficient processing of a relatively small set of instructions. RISC design is based on the premise that most of the instructions a computer decodes and executes are simple. As a result, RISC architecture limits the number of instructions that are built in to the microcontroller but optimizes each so it can be carried out very rapidly (usually within a single clock cycle.). These RISC structure gives the Following advantages.

- The RISC structure only has 35 simple instructions as compared to others
- The execution time is same for most of the instructions (except very few numbers).
- The execution time required is very less (5 million instructions / second (approximately).

The memory in a PIC chip used to store the data and programs temporary or permanently. As like normal microcontrollers, the PIC chip also has certain amount of RAM, ROM, EEPROM, other flash memory, etc.

ROM memory is used for permanent storage. The ROM memory also called as n program memory. API chip has certain amount of ROM memory.

EEPROM memory is another category of ROM memory. The contents in the EEPROM changes during runtime and at that time it acts like a RAM memory. But the difference is after the power goes off, the data remains in this ROM chip. This is the one of the special advantages of EEPROM. In the PIC chip the function of EPROM is to store the values created during the runtime.

RAM memory is the one of the complex memory module in a PIC chip. This memory associated with various type of registers (special function registers and general purpose registers) and memory BANK modules (BANK0, BANK1, etc.). Once the power goes off, the contents in the RAM will be cleared. As like normal microcontrollers, the RAM memory is used to store temporary data and provide immediate results.

Flash memory is a special type of memory where READ, WRITE, and ERASE operations can be done many times. This type of memory was invented by INTEL corporation in1980. APIC Chip normally contains a certain amount of flash memory.

Information is stored in a CPU memory location called a register. Registers can be thought of as the CPU's tiny scratch pad, temporarily storing instructions or data. Registers basically classified into the following.

1. General Purpose Register (GPR)

A general purpose register (or process or register) is a small storage area available on a CPU whose contents can be accessed more quickly than other storage that available on PIC. A general purpose e register can store both data address simultaneously.

2. Special Function registers (SFR)

These are also a part of RAM memory locations. As compared to GPR, their purpose is predetermined during the manufacturing time and cannot be changed by the user. It is only for special dedicated functions.

Interrupt is the temporary delay in a running program. These delays stop the current execution for a particular interval. This interval / delay is usually called as interrupt. When an interrupt request arrives into a current execution program, then it stops its regular execution. Interrupt can be performed by externally (hardware interrupt) or internally (by using software).

BUS is the communication or data transmission / reception path in a microcontroller unit.

In a normal microcontroller chip, two types of buses are normally available.

1. Data bus

Data bus is used for memory addressing. The function of data bus is interfacing all the circuitry components inside the PIC chip.

2. Address bus

Address bus mostly used for memory addressing. The function of address bus is to transmit the address from the CPU to memory locations.

These ports are used for the transmission (TX) and reception (RX) of data. These transmissions possible with help of various digital data transceiver modules like RF, IR, Bluetooth, etc. This is the one of the simplest way to communicate the PIC chip with other devices.

Oscillator unit basically an oscillation / clock generating circuit which is used for providing proper clock pulses to the PIC chip. This clock pulses also helps the timing and counting applications. APIC chip normally uses various types of clock generators. According to the application and the type of PIC used, the oscillators and its frequencies may vary. RC (Resistor-Capacitor), LC (Inductor-Capacitor), RLC (Resistor-Inductor-capacitor), crystal oscillators, etc are the normal oscillators used with APIC chip. The entire PIC chip has an area for storing there turn addresses. This area or unit called Stack is used in some Peripheral interface controllers. The hardware stack is not accessible by software. But for most of the controllers, it can be easily accessible.

Limitations of PIC Architecture

- Peripheral Interface Controller has only one accumulator.
- Small instruction set.
- Register banking switch required to access RAM of other devices.
- Operations and registers are not orthogonal.
- Program memory is not accessible.

Advantages of PIC Controlled System

- **Reliability :** The PIC controlled system often resides machines that are expected to run continuously for many years without any error and in some cases recover by themselves if an error occurs (with help of supporting firmware).
- **Performance :** Many of the PIC based embedded system use a simple pipelined RISC processor for computation and most of them provide on-chip SRAM for data storage to improve the performance.
- **Power consumption :** A PIC controlled system operates with minimal power consumption without sacrificing performance. Power consumption can be reduced by independently and dynamically controlling multiple power platforms.
- **Memory :** Most of the PIC based systems are memory expandable and will help in easily adding more and more memory according to the usage and type of application. In small applications the inbuilt memory can be used.

B. Raspberry Pi

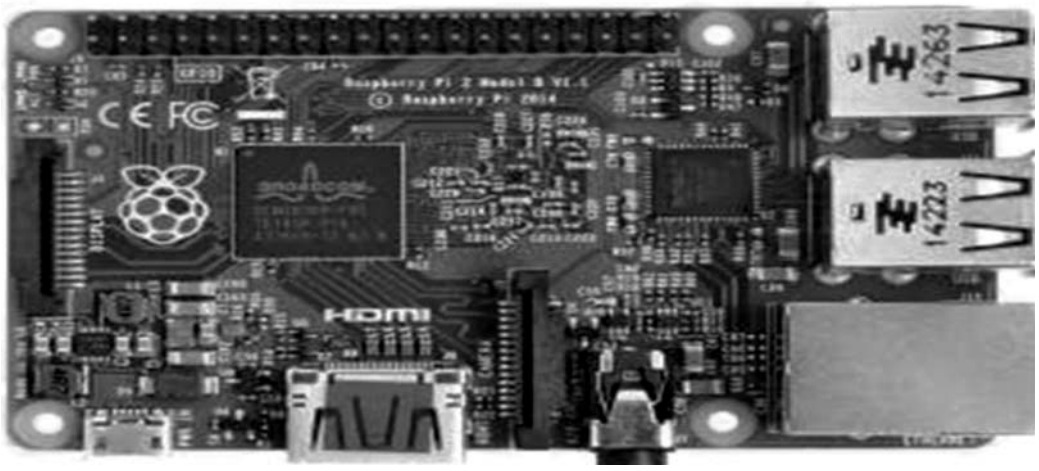

Figure 3

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 cratch 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. 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 bird houses with infra-red cameras.

The Raspberry Pi hardware has evolved through several versions that feature variations in memory capacity and peripheral – device support.

This block diagram depicts models A, B, A+, and B+. Model A and A+ and Zero lack the Ethernet and USB hub components. For our project we are going to use Raspberry Pi 2 model B.

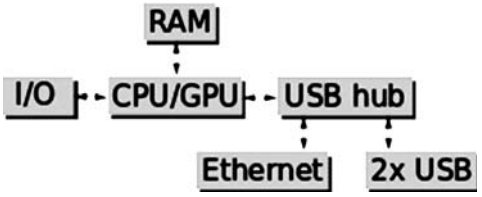

Figure 4

Table 1

Raspberry Pi 2 model B contains following specifications

Specifications	***Raspberry Pi 2 Model B***
System on a chip (SoC)	Broad com BCM 2835
CPU	900 MHz quad-core ARM Cortex - A7
Graphics processing unit (GPU)	BroadcomVideo Core IV @ 250 MHz (BCM 2837 : 3 D part of GPU @ 300 MHz, video part of GPU @ 400 MHz) Open GLES 2.0 (BCM 2835, BCM 2836:24 GFLOPS / BCM 2837:28.8 GFLOPS) MPEG-2 andVC-1 (with license),1080p30 H.264/MPEG-4 AVC high-profile decoder and encoder (BCM 2837:1080p 60)
Memory (SDRAM)	1 GB (shared with GPU)
USB 2.0 Ports	4 (vi a the on-board 5-port USB hub)
Video Input	15- pin MIPI camera interface (CSI) connector, used with the Raspberry Pi camera
Video Output	HDMI composite video (3.5mm TRRS jack)
Audio Output	Analog via 3.5mm phone jack; digital via HDMI
On Board Storage	Micro SDHC slot
On Board Network	10 / 100 M bit / s Ethernet 802. 11 n wireless Bluetooth 4.1
Low –levelPeripherals	17 × GPIO plus the same specific functions, and HATID bus
Power Rating	800mA (4.0W)
Power Source	5V via Micro USB or GPIO header
Size	85.60mm × 56.5mm (3.370 in × 2.224 in)
Weight	45g

C. Software Integration

Python Language

Python is a widely used high-level, general - purpose, interpreted, dynamic programming language. Its design philosophy emphasizes code read ability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java. The language provides constructs intended to enable clear programs on both a small and large scale.

Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

Python provides two levels of access to network services. At a low level, you can access the basic socket support in the under lying operating system, which allows you to implement clients and servers for both connection-oriented and connectionless protocols.

Python also has libraries that provide higher-level access to specific application-level network protocols, such as FTP, HTTP, and soon.

Sockets

Sockets are the end points of a bidirectional communications channel. Sockets may communicate within a process, between processes on the same machine, or between processes on different continents.

Sockets may be implemented over a number of different channel types: Unix domain sockets, TCP, UDP, and soon. The socket library provides specific classes for handling the common transports as well as a generic interface for handling the rest. Sockets have their own vocabulary:

Table 2

Domain	***The family of protocols that Isused as the transport***
Type	The type of communications Between the two end points, typically SOCK_STREAM for connection – oriented protocols and SOCK _ DGRAM for connectionless protocols.
Protocol	Typically zero, this may be used to identify a variant of a protocol within a domain and type.
Hostname	The identifier of a network interface: A string, which can be a hostname, adotted-quad address, oran IPV6 address in colon (and possibly dot) notation A string “<broadcast>”,which specifies an INADDR_BROADCAST address. A zero-length string, which specifies INADDR_ANY, or An Integer, interpreted as a binary address in host byte order.
Port	Each server listens for clients calling on one or more ports. A port may be a Fix num port number, a string containing a port number, or the name of a service.

The socket module

To create a socket, you must use the socket. Socket () function available in socket module, which has the general syntax−

S = socket.socket (socket_family, socket_type, protocol = 0)

Here is the description of the parameters−

Socket_family : This is either AF_UNIX or AF_INET, as explained earlier.

Socket_type : This is either SOCK_STREAM or SOCK_DGRAM.

Protocol : This is usually left out, defaulting to 0.

Once you have socket object, then you can use required functions to create your client or server program.

A. Simple Server

To write Internet servers, we use the socket function available in socket module to create a socket object. A socket object is then used to call other functions to setup a socket server.

Now call bind (hostname, port) function to specify a port for your service on the given host. Next, call the accept method of the returned object. This method waits until a client connects to the port you specified, and then returns a connection object that represents the connection to that client.

The following code is used to create a server :

```
Import socket
importsys
importpsycopg2
HOST=''
PORT=5005
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
print'Socketcreated'
s.bind((HOST,PORT))
print'Socketbindcomplete'
conn=psycopg2.connect(database="l",user="postgres",
password="postgres",host="127.0.0.1",port="5432")
print"Opened data base successfully"
cur=conn.cursor()
cur.execute(\'''CREATETABLEloyola
(IDSERIALPRIMARYKEY,
TIME INT  NOTNULL,
Direction INT NOTNULL,
Distance INT NOTNULL);\''')
#print"Table created successfully"
conn.commit()
conn.close()'''
#id=0
while1:
d=s.recvfrom(1024)
data=d[0]
addr=d[1]
#id+=1
if not data:
break
reply='OK...'+data
s.sendto(reply,addr)
```

```
print'Data['+addr[0]+':'+str(addr[1])+']-'+data.strip()
values=data.split(".")
time=values[0]
Direction=values[1]
Distance=values[2]
conn=psycopg2.connect(database="l",user="postgres",
password="postgres",host="127.0.0.1",port="5432")
#print"Openeddatabasesuccessfully"
cur=conn.cursor()
cur.execute("INSERTINTOloyola(TIME,Direction,Distance)\
VALUES(%s,%s,%s)",(time,Direction,Distance));
conn.commit()
#print"Recordscreatedsuccessfully";conn.close()
s.close()
```

A. Simple Client

Let us write a very simple client program which opens a connection to a given port12345 and given host. This is very simple to create a socket client using Python's socket module function.

The socket.connect (hosname, port) opens a TCP connection to host name on the port. Once you have a socket open, you can read from it like any IO object. When done, remember to close it, as you would close a file.

The following code is used to create client –import socket

```
Import serial ser=serial.Serial('/dev/ttyAMA0',9600)UDP_IP="192.168.0.17"
UDP_PORT=5005
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)while(1):
msg=ser.readline().strip()
s.sendto(msg,(UDP_IP,UDP_PORT))d=s.recvfrom(1024)
reply=d[0]addr=d[1]
print'Serverreply:'+reply
```

3. MY SQL AND WAMP SERVER

My SQL is the most popular Open Source Relational SQL database management system. My SQL is one of the best RDBMS being used for developing web-based software applications.

My SQL Database:

My SQL is a fast, easy-to-use RDBMS being used for many small and big businesses. My SQL is developed, marketed, and supported by My SQL AB, which is a Swedish company. My SQL is becoming so popular because of many good reasons:

My SQL is released under an open-source license. So you have nothing to pay to use it. My SQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.

My SQL uses a standard form of the well-known SQL data language. My SQL works on many operating systems and with many

Languages including PHP, PERL, C, C++, JAVA, etc. My SQL works very quickly and works well even with large datasets.

My SQL is very friendly to PHP, the most appreciated language for web development. My SQL supports large databases, upto 50 million rows or more inatable. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million tera bytes (TB). My SQL is customizable. The open-source GPL license allows programmers to modify the My SQL software to fit their own specific environments. The final outcome of the project is the message received by

The local client, which looks like –

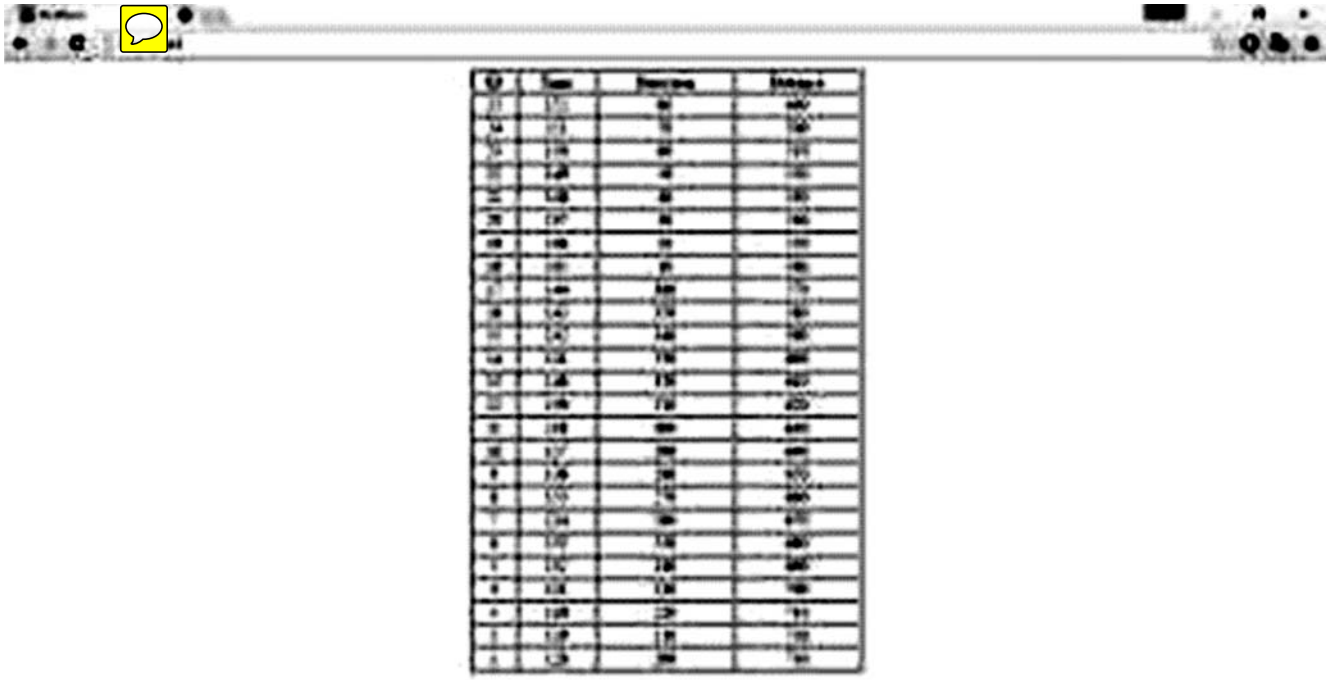

Figure 5

4. REFERENCES

1. The Design of Multi-protocol Converter Based on Functional Analysis :- YanWu, Jing jing Zhou School of information engineering Jiang Xi University of Technology, China.
2. Consultation Paper on Migration to IP based Networks by Telecom Regulatory Authority of India (TRAI) Consultation PaperNo: 8/2014 Publication Date: 30thJune 2014 Publication Place: Mahanagar Door sanchar Bhawan, JawaharLal Nehru Marg, New Delhi – 110002.