Chapter 7

Enhancement of ARM Architecture by Porting Operating System on ARM Enabled Raspberry Pi Board

7.1 Overview
7.2 Popular ARM boards
7.3 Introduction to RaspberryPi Model B
7.4 Introduction to Linux
7.5 Available Linux based OS for Porting
7.6 Writing Raspbian OS on SD card
7.7 Formatting SD card with FAT 32
7.8 RaspberryPi as Single Board Computer
7.9 Introduction to Arduino UNO
7.10 Communication between Arduino & RaspberryPi
7.11 Real time monitoring and Data logging system in RaspberryPi with Arduino
7.12 Summary & discussion
7.1 Overview:

As discussed in Chapter 4 that ARM architecture is capable of utilizing power of operating system, of-course ARM 7 TDMI architecture has its own limitation. In this chapter I am using ARM enabled board and utilizing the capabilities of the ARM architecture for porting Operating System on it.

As we know that ARM consists of RISC architecture and it supports a native full-fledged 32-bit instruction set. ARM also specifies two other instruction sets: a 16-bit compressed RISC set called Thumb, and an 8-bit instruction set for Java byte-codes called Jazelle[2]. The ARM architecture has 37 register. These registers are general purpose registers including a program counter and 6 are status registers. Some of these registers are banked and are hidden except when executing in specific processor modes. These registers such as the stack pointer are automatically switched when entering a different processor mode. This design allows fast processing of interrupts as the handler code does not need to manually switch to a new stack. The ARM processors support seven modes[2][3] of operation. All modes except ARM USR have privileges to perform operations that control the MMU and interrupts.

ARM board provides high speed, better accuracy, good flexibility and low cost solution for development of embedded system. It also supports memory management, process management and peripheral management when ported with OS. When ported with OS it can be used to develop a complex embedded machine. Linux open source distributions provide a good solution for developers to download, customize, compile and burn Linux distribution OS to their core for free. I have selected Raspberry Pi ARM board and ported Debian (Raspebian) OS into Raspberry Pi.

7.2 Popular ARM Boards:

There are many varieties of ARM based boards available; here I have listed few popular ARM boards[44] which are available in the market.
MK802:
It is a PC-on-a-stick produced by Rikomagic, a Chinese company using AllWinner A1X SoC, based on an ARM architecture, composed of an ARM V7 based Cortex-A8 1 GHz processor, a Mali-400 MP GPU, Wi-Fi 802.11 b/g/n, and a VPU CedarX capable of displaying 1080p video. The thumb sized MK802, which was first brought into market in May 2012, can turn a display with a HDMI or DVI-D port into an Android computer.

Beagle Board:
It is a low-power open-source hardware single-board computer produced by Texas Instruments in association with Digi-Key. The Beagle Board was also designed with open source software development in mind, and as a way of demonstrating the Texas Instrument's OMAP3530 system-on-a-chip. The board was developed by a small team of engineers as an educational board that could be used in colleges around the world to teach open source hardware and open source software capabilities.

Panda Board:
It is a low-power, low-cost single-board computer development platform based on the Texas Instruments OMAP4430 system on a chip (SoC). The Panda Board ES is a newer version based on the OMAP4460 SoC, with the CPU and GPU running at higher clock rates.

ODROID:
It is the Ultra-Compact and Ultra-low-cost platform with 1.4 GHz Quad-core CPU and 1Gbyte RAM. This board is the world’s smallest and cost effective
US$69 priced Quad-core ARM PC. Board dimension is only 48x52mm (1.9x2.0inch approx.). The new ODROID boards are powered by the Samsung Exynos-4412 and Exynos-4412-Prime chip. **Exynos 4412 Prime** has 4 cores of 1.7 GHz ARM Cortex-A9 processor and 2Gbyte of Low-Power DDR RAM. New ODROID boards can run the Android and various Linux operating systems.

**Raspberry-Pi:**
It is a credit-card-sized single-board computer developed in the UK by the RaspberryPi Foundation with the intention of promoting the teaching of basic computer science in schools. The RaspberryPi has a Broadcom BCM2835 system on a chip (SoC), which includes an ARM1176JZF-S 700 MHZ processor (The firmware includes a number of “Turbo” modes so that the user can attempt over clocking, up-to 1 GHz, without affecting the warranty), VideoCore IV GPU, and originally shipped with 256 megabytes of RAM, later upgraded to 512MB. It does not include a built-in hard disk or solid-state drive, but uses an SD card for booting and long-term storage. The Foundation's goal is to offer two versions, priced at US$25 and US$35.

### 7.3 Introduction To Raspberry Pi Model B:

Raspberry Pi model B is a credit-card-sized single-board computer developed in the UK by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science in schools. The Raspberry Pi has a Broadcom BCM2835 system on a chip (SoC), which includes an ARM1176JZF-S 700 MHz processor (The firmware includes a number of “Turbo” modes so that the user can attempt over clocking, up-to 1 GHz, without affecting the warranty), VideoCore IV GPU, and originally shipped with 256 megabytes of RAM, later upgraded to 512MB. It does not include a built-in hard disk or solid-state drive, but uses an SD card for booting and long-term storage. Actual(physical) look of the board[39] is shown in Figure 7.1 and block diagram of the board is shown in Figure 7.2.
Figure 7.1 Physical Look Of The RaspberryPi Model B

Figure 7.2 Block Diagram Of RaspberryPi Model B
The Specification[39] and detailed information regarding RaspberryPi model B is given in Table 7.1

<table>
<thead>
<tr>
<th>Product details</th>
<th>Raspberry –Pi model B</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOC Type</td>
<td>Broadcom BCM2835</td>
</tr>
<tr>
<td>Core Type</td>
<td>ARM1176JZF-S</td>
</tr>
<tr>
<td>No. Of Cores</td>
<td>1</td>
</tr>
<tr>
<td>GPU</td>
<td>VideoCore IV</td>
</tr>
<tr>
<td>CPU Clock</td>
<td>700 MHz</td>
</tr>
<tr>
<td>RAM</td>
<td>512 MB</td>
</tr>
<tr>
<td>Flash</td>
<td>0 MB</td>
</tr>
<tr>
<td>USB Ports</td>
<td>Yes 2</td>
</tr>
<tr>
<td>Ethernet</td>
<td>Yes 10/100M</td>
</tr>
<tr>
<td>HDMI</td>
<td>Yes</td>
</tr>
<tr>
<td>Analog Video Out</td>
<td>Yes Composite</td>
</tr>
<tr>
<td>Analog Audio Out</td>
<td>Yes</td>
</tr>
<tr>
<td>Analog Audio In</td>
<td>Yes</td>
</tr>
<tr>
<td>SPI</td>
<td>Yes</td>
</tr>
<tr>
<td>I2C</td>
<td>Yes</td>
</tr>
<tr>
<td>GPIO</td>
<td>Yes</td>
</tr>
<tr>
<td>LCD Panel</td>
<td>Yes DSI</td>
</tr>
<tr>
<td>Camera</td>
<td>Yes DSI</td>
</tr>
<tr>
<td>SD/MMC</td>
<td>Yes SD Cage</td>
</tr>
<tr>
<td>Wi-Fi</td>
<td>No</td>
</tr>
<tr>
<td>Bluetooth®</td>
<td>No</td>
</tr>
<tr>
<td>Height</td>
<td>3.37 in (85.6 mm)</td>
</tr>
<tr>
<td>Width</td>
<td>2.12 in (53.98 mm)</td>
</tr>
<tr>
<td>Weight</td>
<td>1.58 oz. (45 g)</td>
</tr>
</tbody>
</table>

Table 7.1 Specifications Of RaspberryPi Model B

7.4 Introduction To Linux:

Linux[41] (also known as GNU/Linux) is a computer operating system, like Microsoft Windows or Apple Mac OS. Unlike those two, however, Linux is built
with a collaborative development model. The operating system and most of its software are created by volunteers and employees of companies, governments and organizations from all over the world. The operating system is free to use and everyone has the freedom to contribute to its development. This co-operative development model means that everyone can benefit. Because of this, it is called as Free Software, or Socially Responsible Software. Closely related is the concept of Open Source Software. Together, Free and Open Source Software is collectively abbreviated as FOSS. This contrasts with the proprietary (or closed source) development model used by some software companies today. Many of the principles behind FOSS are derived from the axiom of standing on the shoulders of giants, most famously used by Isaac Newton, which has guided scientific and industrial development for hundreds of years. Transparency of the code and development process means that it can be participate in and audited at all levels. Software is just another form of information, and people have the right to have full control over that information.

Linux has many other benefits, including speed, security and stability. It is renowned for its ability to run well on more modest hardware. Linux comes from the venerable UNIX family of operating systems, and so has been built from the ground-up with Internet-style networking & security in mind. Hence, viruses, worms, spyware and adware are basically a non-issue on Linux.

7.5 Available Linux Based OS For Porting:

Linux provide open source distributions for developers to download, customize, compile and burn OS to their core for free. Various Linux distributions[36] are available for porting[35] on ARM board in market such as

- The RaspberryPi Fedora Remix
- Debian
- Arch Linux ARM
- Moebius
- OpenELEC
- RISC OS
The Raspberry Pi Fedora Remix:
It is a Linux software distribution for the Raspberry Pi computer. It contains software packages from the Fedora Project (specifically, the Fedora ARM secondary architecture project), packages which have been specifically written for or modified for the Raspberry Pi, and proprietary software provided by the Raspberry Pi Foundation.

Debian:
It was the default distribution on the Alpha boards. Boot time depends on width & speed of SD-card. Alpha board boot into Debian prompt (no GUI) was timed taking about 34 seconds. The Debian distribution for Raspberry Pi is the Cambridge reference file system, which is a fully functional Debian Squeeze installation containing LXDE (desktop) and Midori (browser); development tools; and sample code for accessing the multimedia functionality on the device.

Arch Linux ARM:
It is based on Arch Linux, which aims for simplicity and full control to the end user. It provides a lightweight base structure that allows us to shape the system as per our needs. For this reason, the Arch Linux ARM image for the Raspberry Pi does not come with a graphical user interface, though one can easily install for oneself. Arch Linux ARM is on a rolling-release cycle that can be updated daily through small packages instead of huge updates every few months.

Moebius:
A very compact ARM HF debian based distribution, it fits in a 1 GB SD card, has auto resizing features to better adapt to your SD card size and uses Raspbian huge repositories for installing everything you need. A wise configuration and a small memory footprint are ideal for a headless machine or for interacting with real world I/O devices.
OpenELEC:
OpenELEC is an embedded operating system built specifically to run XBMC, the open source entertainment media hub. The idea behind OpenELEC is to allow people to use their Home Theatre PC (HTPC) like any other device you might have attached to your TV, like a DVD player or Sky box. Instead of having to manage a full operating system, configure it and install the packages required to turn it into a hybrid media center, OpenELEC is designed to be simple to install, manage and use, making it more like running a set-top box than a full-blown computer.

RISC OS:
RISC OS is a fast and lightweight computer operating system designed in Cambridge, England by Acorn. First released in 1987, its origins can be traced back to the original team that developed the ARM microprocessor. RISC OS includes BBC BASIC which was primarily conceived to teach programming skills as part of the BBC computer literacy project. RISC OS Open (ROOL) has released the sources. Community members have ported the OS to the Beagle Board and similar hardware.

PiBangLinux:
PiBangLinux is a Raspbian based distribution. PiBang is insPired by CrunchbangLinux, an i686 and x86_68 Debian based distribution. It comes preconfigured with many helpful scripts and Pipe menus as well as a fork of RasPi-config with increases functions such as support for changing the user and hostname. PiBang is also one of the heavier RaspberryPi distributions boasting a complete package set with favorites such as Abiword, OMXPlayer, GIMP, and VLC all pre-installed.

Plan 9:
Plan 9 is a distributed operating system originally designed and implemented by Ken Thompson, Rob Pike, Dave Presotto, and Phil Winter bottom @ Bell Labs. It is a lean operating system that has been ported to super computers such as IBM's Blue Gene down to tiny boards such the RaspberryPi.
Raspbian:

RaspberryPi + Debian = Raspbian[39]. A project to create a hard float port of Debian Wheezy (7.x) armhf for the RaspberryPi. The intent of Raspbian is to bring to the RaspberryPi user 10,000s of pre-built Debian packages specifically tuned for optimal performance on the RaspberryPi hardware.

7.6 Writing Raspbian OS on SD Card:

One can write the Raspbian OS on SD card in various operating system environments. I have used Windows operating system for writing the Raspbian OS on SD card, steps[39] are mentioned with appropriate sequence. For other operating system like Ubuntu, Mac, Command line(DOS) steps are mentioned in appendix.

Steps for Windows:

2. Download Disk Imager from https://launchpad.net/win32-image-writer/+download. As shown in Figure 7.3

![Win32 Disk Imager](image)

**Figure 7.3 Win32 Disk Imager**

3. Insert your flash media.
4. Note the drive letter assigned to your flash media.
5. Start Disk Imager.
6. Select the downloaded file and target device, and click "Write".
7. Remove your flash media when the operation is complete.
7.7 Formatting SD Card With FAT32:

Here I have used “HP USB Disk Storage Format Tool” for formatting our SD card after porting OS into it as shown in Figure 7.4. Once you port the image of OS the original size of SD card will reduced to some MBs because the OS has blocked some part of memory and so that part cannot be accessed by user before formatting memory card. This utility will format any USB flash drive, with your choice of FAT, FAT32, or NTFS partition types. Optionally one can also make the disk BOOTABLE by specifying a file location.

FIXES:

- Allows creation of a FAT32 volume larger than 32 GB.
- Fixes installation issue where installation process stopped after the earlier version of software was uninstalled and the new software was not automatically installed. The installation process now restarts automatically to install the new software after uninstalling the older version.

7.8 RaspberryPi As Single Board Computer (SBC):

The purpose of using ARM enabled board is to utilize the power of ARM architecture, where by a single board can be used as good as computer. Here I have described various steps which can make single board (RaspberryPi model-B) to work as computer.
7.8.1 Setup:
There are many ways to setup the Raspberry Pi, depending on your needs and the peripherals you wish to use. The set up process[39] can be completed in five basic steps as shown in Figure 7.5.

![Figure 7.5 Steps For RaspberryPi Setup](image)

7.8.2 Basic Equipment Required For General Setup:
1. Micro USB Charger (rated at 5V 700mA minimum), plus micro USB cable if needed.
2. SD Card (2Gb up to SDHC 32Gb) with Raspbian OS which was created as per the section 7.6.
3. HDMI Cable/RCA cable for display
4. Compatible Mouse/Keyboard
5. Powered USB Hub (required for high powered USB devices)
6. Network Cable (Ethernet)
7. Smart Phone (for USB tethering)

7.8.3 Booting For The First Time:
1. On RaspberryPi login screen type “pi” and you will be asked for password which is "raspberry". Figure 7.6 shows the screen shot of the booting and login in the RaspberryPi.
2. When you see the prompt: pi@raspberry ~ $ type "startx" for a graphical environment. Figure 7.7 shows screen shot of the graphical environment after startx in RaspberryPi.
7.8.4 Getting Network Access:
Raspbian consists of dedicated browser in Raspberry Pi named “Midori”. So, one can access internet through it using any of two methods listed below.

- Ethernet
- USB tethering from your smart phone

Using Ethernet:
Connect the Ethernet Cable and open the browser and set the IP Address and proxy and get connected. For excess internet through terminal, type the following for setting IP

```bash
ifconfig eth0 10.10.17.23
```

Here I have used 10.10.17.23 IP. Now issue Ping command to check the internet connection. Figure 7.8 shows the screenshot of the access of the internet through Ethernet using Midori browser in RaspberryPi.

![Figure 7.8 Internet Access Through Ethernet In RaspberryPi](image)

Using Smart Phone via USB Tethering:
Connect smart Phone and RaspberryPi via USB cable and enable USB tethering in your device. Then open terminal and follow the steps given below for configuring RaspberryPi.
1) Get root access by typing “**sudo bash**”

2) For confirming that smart phone (used for tethering) is connected via USB with RaspberryPI issue command **“lsusb”**. I have connected the Samsung smart phone via USB which is shown in Figure 7.9.

![Figure 7.9 Screen shot Of lsusb Command](image)

3) Go to the directory etc/network/

4) Open and file interfaces with vi command and add following lines in file
   auto eth0
   iface eth0 inet dhcp
   iface usb0 inet dhcp

5) Enable the usb tethering in Mobile device that is to be used for the internet connectivity.

6) Save the file and type the following command in terminal for configure the changes, **“ifup usb0”**

7) Now for checking whether Pi have been configured or not, type **“ifconfig”** command which allow the user to view information about the configured network interfaces. Figure 7.10 shows screenshot of ifconfig.
After connecting the smart phone and verifying above steps I have browse the internet using medori browser via smart phone, the screen shot of the internet access shown in Figure 7.11.
So, by using the ARM enabled board of Raspberry Pi, one can feel the power of single board computer. After enhancing the board as good as computer, architecture allows us to develop custom design applications. I have developed an application for real time monitoring and data logging system with ARM architecture of Raspberry Pi and Arduino UNO.

7.9 Introduction Of Arduino UNO:

Arduino[42] is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It is intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software running on a computer (e.g. Flash, Processing, MaxMSP). The physical look of the board is shown in Figure 7.12

![Figure 7.12 Physical Look Of Arduino UNO Board](image)

**7.9.1 Features Of The Arduino Uno:**

- An open source design. The advantage of it being open source is that it has a large community of people using and troubleshooting it. This makes it easy to find someone to help you debug your projects.
An easy USB interface. The chip on the board plugs straight into your USB port and registers on your computer as a virtual serial port. This allows you to interface with it as through it were a serial device. The benefit of this setup is that serial communication is an extremely easy (and time-tested) protocol, and USB makes connecting it to modern computers really convenient.

- Very convenient power management and built-in voltage regulation. You can connect an external power source of up to 12v and it will regulate it to both 5v and 3.3v. It also can be powered directly off of a USB port without any external power.
- An easy-to-find, and dirt cheap, microcontroller "brain." It has countless number of nice hardware features like timers, PWM Pins, external and internal interrupts, and multiple sleep modes.
- A 16 MHz clock. This makes it not the speediest microcontroller around, but fast enough for most applications.
- 32 KB of flash memory for storing your code.
- 13 digital Pins and 6 analog Pins. These Pins allow you to connect external hardware to your Arduino. These Pins are key for extending the computing capability of the Arduino into the real world. Simply plug your devices and sensors into the sockets that correspond to each of these Pins and you are good to go.
- An ICSP connector for bypassing the USB port and interfacing the Arduino directly as a serial device. This port is necessary to re-boot load your chip if it corrupts and can no longer talk to your computer.
- An on-board LED attached to digital Pin 13 for fast an easy debugging of code.
- And last, but not least, a button to reset the program on the chip.

7.9.2 Arduino UNO IDE:
Arduino has dedicated IDE[42] (integrated Development Environment) which uses some low level language which is easy to understand as well as write and IDE converts this language into C/C++ for execution of code. Each program written into Arduino IDE is called “sketch”. Here I have shown the sample sketch[42] in the Figure 7.13 for the LED blinking code.
Verify:
This button is used for compile the written sketch.

Upload:
This button is used for uploading the code into connected board. Before uploading the code into board the IDE first compiles it and checks for and error. The sketch has two main bodies.

- setup
- loop

Setup:
In this portion one can assign the pins to various names and define that either that pin should be input of output. In short setup for the pins is essential.

Loop:
In this portion one can write that part of code that should be repeated continuously. So this part does the work of “while 1”.

Figure 7.13 Sample Arduino UNO Sketch

```cpp
// Example 81: Blinking LED

const int LED = 13; // LED connected to digital pin 13

void setup()
{
  pinMode(LED, OUTPUT); // set the digital pin as output
}

void loop()
{
  digitalWrite(LED, HIGH); // turn the LED on
  delay(1000); // wait for a second
  digitalWrite(LED, LOW); // turn the LED off
  delay(1000); // wait for a second
}
```
7.10 Communication Between Arduino And RaspberryPi:

Once we connect Arduino UNO with RaspberryPi it becomes essential that they must communicate properly. To facilitate the same RaspberryPi should be loaded with Python library for serial communications called ‘pySerial’ which allows RaspberryPi to serially communicate with Arduino using USB. Below are steps which are required to complete for serial communication between RaspberryPi and Arduino.

1. Load the Sketch for serial transmission on Arduino and connect it with RaspberryPi using USB.
2. Power up RaspberryPi and login.
3. Browse and download pyserial-2.5.tar.gz (106.3 kB) and save it somewhere convenient.
4. This is a gun-zipped tar file. This needs unzipping and untaring. To unzip it open a Terminal, this can be found from the 'start menu' under accessories. Now paste the following commands into it.

```
cd /home/Pi/Desktop/other
gunzip pyserial-2.5.tar.gz
tar -xvf pyserial-2.5.tar
```

5. Install pySerial, by typing these lines in terminal window:

```
cd pyserial-2.5
sudo python setup.py install
```

6. Run Python 2. (Obtain from menu under Programming - Use Python 2.)

7. Write following Python code in python 2 to access the Serial port:

```python
import serial
ser=serial.Serial('/dev/ttyACM0',9600)
while 1:
    ser.readline()
```

This will make Pi ready to read the data transmitted serially by Arduino and the data will be shown on Screen as soon as you type last command. Please make sure that baudrate of both the device is same. For checking the baudrate of RaspberryPi go to file serial.py and compare the baudrate mentioned there with your baudrate if it is same it is good if not then change it with 9600 standard baudrate otherwise your device will not talk.

7.11 Real Time Monitoring And Data Logging System In RaspberryPi With Arduino:

For acquiring the information about temperature, I have connected temperature sensor with analog input of Arduino and Arduino is connected to RaspberryPi[34] using USB (as described in earlier). Figure 7.15 showing RaspberryPi, Arduino Uno and breadboard for connecting Temperature sensor, LED and Resistor.
To operate the system of temperature monitoring and data logging the overall task divided into 3 phases:

1. First upload the Python Arduino code for temperature sensing using Arduino IDE into the Arduino UNO board. This will sense the temperature and send it to serial port(USB).

```cpp
const int ledPin = 13;
float tempC;
int reading;
tempPin = 0;
void setup()
{
PinMode(ledPin, OUTPUT);
Serial.begin(9600);
analogReference(INTERNAL);
}
void loop()
```
```

{  
  reading = analogRead(tempPin);  
tempC = reading / 9.31;  
Serial.println(tempC); 
  if (Serial.available()) 
  {  
    flash(Serial.read() - '0');  
  
  }  
  delay(1000);  
} 

void flash(int n)  
  {  
    for (int i = 0; i < n; i++)  
    {  
      digitalWrite(ledPin, HIGH);  
      delay(100);  
      digitalWrite(ledPin, LOW);  
      delay(100);  
    }  
  }


2. Make a python file named “dataLogger.py”. This program reads data coming from the serial port and saves that data to a text file. It expects data in the format "temperature_reading". It assumes that the Arduino shows up in /dev/ttyACM0 on the Raspberry Pi.

```
#!/usr/bin/python
import serial
ser = serial.Serial('/dev/ttyACM0',9600)
try:
  while 1:
    line=ser.readline().rstrip()
    temp2=line
    print("%s"%(temp2))
    f=open('tempLog.dat','a')
    print>>f,"(%s"%(temp2))
```

```
3. Now for plotting the graph using gnuplot first install gnuplot using terminal by following command:
   Sudo apt-get install gnuplot

   After installing gnuplot again make one new file named “plotData.plt” and add following lines into it
   
   reset
   set title "temp sensor"
   set term png
   set xtics rotate by -45
   set xlabel "time (s)"
   set ylabel "Temperature (C)"
   set ytics nomirror
   set grid
   set output "data.png"
   set key left
   plot "tempLog.dat" title ""

   After doing all this do following steps:
   ➢ Go to that particular file location where you have saved dataLogger.py and plotData.plt files
   ➢ Now type following command
     Sudo python dataLogger.py

   This code will continuously shows the temperature in terminal and at the same time that data will be stored into the file named “tempLog.dat”. The temperature is stored with the rate of one entry per second and listed all the sensed temperature into the “tempLog” until we give keyboard interrupt. Figure 7.16 shows the Screen shot of TempLog.dat file.
Now type following command

```
Sudo gnuplot plotData.plt
```

This code will generate the plot of collected temperature data which are stored into the “tempLog.dat” file and generates the PNG file named “data.png” and in this file the image of graph plotted is shown in Figure 7.17.
7.12 Summary And Discussion:

ARM enabled RaspberryPi board which comprises of ARM 11 architecture can be faithfully operated as single board computer. This Chapter explains the design and development of the Data acquisition application using debian operating system. The system with ported operating system also has flexibility to connect with the internet. Unlike the conventional desktop computing, the design system with ARM enabled board allows the user to design custom application. So, performance of ARM architecture is enhanced by porting operating system on it, and utilizing the power of Open Source Access. One can easily extend the data acquisition part by including more parameter for monitoring. To accomplish the task I had invested time on learning Linux and Python programming.