

NET CAMERA INTERFACE USING RABBIT MICROPROCESSOR
DEVELOPMENT BOARD RCM3365

Parth Shah
B.E, Bhavnagar University, India, 2008

PROJECT

Submitted in partial satisfaction of
the requirements for the degree of

MASTER OF SCIENCE

in

ELECTRICAL AND ELECTRONIC ENGINEERING

at

CALIFORNIA STATE UNIVERSITY, SACRAMENTO

SPRING
2011

NET CAMERA INTERFACE USING RABBIT MICROPROCESSOR
DEVELOPMENT BOARD RCM3365

A Project

by

Parth Shah

Approved by:

_____, Committee Chair
Jing Pang, Ph. D.

_____, Second Reader
Manish Gajjar, M.S., M.B.A.

Date

Student: Parth Shah

I certify that this student has met the requirements for the format contained in the University format manual, and that this project is suitable for shelving in the Library and credits is to be awarded for the Project.

_____, Graduate Coordinator
Preetham Kumar, Ph. D.

Date

Department of Electrical and Electronic Engineering

Abstract
of
NET CAMERA INTERFACE USING RABBIT MICROPROCESSOR
DEVELOPMENT BOARD RCM3365

by
Parth Shah

In this high technology world, as we know in day to day life the importance of different application using internet is becoming widespread. Rapid growth of such different application is due to the introduction of Rabbit Microprocessors. This processor is widely used in applications related to net camera. The net camera application using Rabbit Microprocessor development board RCM3365 is the most recent developing application in this field. In this project, I am using C328 CMOS image sensor camera to capture the image. There is a motion sensor, which will sense any motion, taking place in front of it and will in turn activate the camera to take the picture. The other way of activating the camera to take the picture is to enable the feature of capturing the snapshot every 0-30 seconds. CMOS image sensor is controlled by two servo motors; first one is the pan servo motor and second is the tilt servo motor. The captured image can be operated on, in different ways depending on the different applications. The image can be stored in flash memory available on the development board and can be accessed later on in time. It can also be viewed using a web browser. The entire board including the camera

and the servo motors can be operated through web browser using Ethernet protocols. Also, the captured image can be sent in email by setting up the email settings and adding the email address we want to send the email to. As there is a motion sensor and two servo motors connected to the camera and the development board RCM3365, the prototyping board can also be used to implement an application including security live feed of any small area. This is a very flexible development board. Code can be written to achieve a GUI (Graphics User Interface) to access it from anywhere in the world via internet. This project also includes a software code written in Java language that creates a database of different images read. A webpage with the user interface from where a user can insert, delete and view the images is developed. Also, to search within the database a search algorithm of B-Tree is also implemented.

_____, Committee Chair
Jing Pang, Ph. D.

Date

ACKNOWLEDGMENTS

Here, I would like to take the opportunity to express my sincere thanks to all the people involved in this project without whom it would have ever been possible to achieve the milestone that I have achieved today.

With great respect, I would specially like to thank my project advisor and first reader Dr. Jing Pang for her continuous and valuable guidance directing me in the right direction towards successful completion of my project work. I am very thankful to her as she gave me the opportunity to work on this project and helped me in enhancing my knowledge of hardware interface and implementing different protocols of internet.

Also, I would like to thank and express my gratitude to Manish Gajjar for providing me very crucial suggestions time to time to enhance the project output. He has been supportive to me during the entire project and also during proof reading the project report.

Finally, I would like to thank my family and friends for supporting me through this entire project and also in making this project a success.

Parth Shah

TABLE OF CONTENTS

	Page
Acknowledgements.....	vi
List of Figures.....	ix
Chapter	
1. INTRODUCTION	1
1.1 Overview.....	1
1.2 Objective of the Project	3
2. PROJECT RESOURCES: HARDWARE	4
2.1 Rabbit ^R 3000 Microprocessor	5
2.2 Different Memory Modules and I/O Interface	5
2.2.1 SRAM.....	5
2.2.2 Flash EPROM.....	5
2.2.3 NAND Flash	6
2.3 Different Types of Ports.....	7
2.3.1 Ethernet Port	7
2.3.2 Serial Ports.....	7
2.3.3 Serial Programming Port	7
2.4 Serial Programming Cable	8
3. PROJECT RESOURCES: SOFTWARE.....	9
3.1 Overview.....	9
3.2 Tools and Packages	10
3.2.1 Eclipse Tool	10

3.2.2 Language Package.....	11
3.3 Binary Search Tree Algorithm	12
4. PROJECT DETAILS	18
4.1 Camera C328 CMOS Image Sensor and Servo Motor.....	19
4.2 Rabbit Microprocessor 3000	20
4.2.1 Set Up the Internet Connection and Run event_capture.c	21
4.3 Software Code Implemented in Java	23
4.3.1 Binary Tree Implementation in java.....	23
4.3.2 Sorting the Image.....	25
4.3.3 Pseudo Code for Binary Search B - Tree Algorithm.....	27
4.3.4 Web Interface Using Java	30
4.4 Limitation of the Design.....	32
5. SIMULATION AND RESULTS.....	34
5.1 Inserting the Image	34
5.2 Viewing the Image.....	36
5.3 Deleting the Image.....	37
6. CONCLUSION AND FUTURE EXPANSION	39
6.1 Conclusion	39
6.2 Future Expansion	40
REFERENCES	41

LIST OF FIGURES

	Page
1. Figure 2.1 Prototyping Board RCM 3365.....	4
2. Figure 2.2 Rabbit ^R 3000.....	5
3. Figure 2.3 Insertion and Removal of xD-Picture Card (64MB).....	7
4. Figure 2.4 Use of Rabbit 3000 Ports	8
5. Figure 2.5 Serial Programming Cable.....	9
6. Figure 3.1 Adding Package and Server to the New File.....	11
7. Figure 3.2 Binary Search Tree with Nodes and Subtree.....	13
8. Figure 3.3 Flow Chart Showing Search Algorithm Implemented in binary tree ...	14
9. Figure 3.4 Deleting a Node From binary search tree	15
10. Figure 3.5 B – Tree of Order 2.....	16
11. Figure 3.6 Best Case ($\log_2 N$) v/s Worst Case ($\log_2 N - 1$).....	17
12. Figure 4.1 Block diagram for hardware connection.....	18
13. Figure 4.2 STDIO Window Output for servo_calib.c.....	19
14. Figure 4.3 Micro Servo Motor.....	20
15. Figure 4.4 Network Connection Setting for Windows based PC.....	21
16. Figure 4.5 Network Connection Setting for Windows based PC.....	22
17. Figure 4.6 GUI for event_capture.c.....	23
18. Figure 4.7 Block diagram-connection between front end and back end design...	31
19. Figure 4.8 GUI of Web Application in jsp	32
20. Figure 5.1 GUI for Inserting Image.....	34

21. Figure 5.2 Message Display Stating Image has been Added to the Database	35
22. Figure 5.3 data entry in the Table and the Database Folder	35
23. Figure 5.4 Procedure to Follow for Viewing the Image	36
24. Figure 5.5 Image Opened in mspaint	36
25. Figure 5.6 Procedure to Delete the Image.....	37
26. Figure 5.7 Deleted table entry and Image From the Database.....	37
27. Figure 5.8 Implementation of binary tree	38
28. Figure 5.9 Properties File Showing binary tree Implementation	38

Chapter 1

INTRODUCTION

In 21st Century, with the advent in technology and increasing use of internet almost all the application in today's world are related to internet in one or the other way. Whether it is banking online or filling out the admissions forms, everything can be done online via internet nowadays. If I own a motel and I have the security cameras installed everywhere in my motel, I can look at different activities happening around in my motel just by sitting in my office. However, in today's fast growing and rapid moving world everyone wants to access their stuffs online via internet. They want access to different things anywhere in the world from a place where they are right now. With the development of Rabbit microprocessor's camera interface application kit, RCM3365, accessing the camera module for taking pictures, motion sensing and many more applications can be controlled via internet from a place anywhere in the world with internet connection.

1.1 Overview

In day-to-day life, use of internet is becoming widespread. As we know there are a number of vendors who builds camera interface application kit, but with Rabbit Microprocessor's RCM3365 net camera interface application kit, there is an added advantage of accessing the kit via internet from anywhere in the world. The kit can be configured with a specific IP address, which when tried to reach via internet browser gives us a GUI to operate the kit remotely. RCM3365 uses powerful Rabbit 3000^R

microprocessor @ 44.2 MHz [1]. This microprocessor development board is ideal for network-enabling security and access systems. It is also compatible with most of the industrial standards available in market when it is used with a specific Rabbit web, FAT file system and SSL software modules [1]. Major advantage of this project implemented with RCM3365 is the free source availability of all the software. It has all modules including memory, different protocols which are compatible with the processor core, available readily in the market. This is the latest trend in technology related to internet interface. With the addition of the Java program for image database management, handling of the images captured by the prototyping board become very easy [2]. The user interface created with the help of Java, again via internet, to access and handle lot many images at the same time is very helpful.

Rabbit core module and the prototyping board are low cost, low power and very easy to understand and use type of product. User program for different applications can be downloaded, simulated and debugged using Dynamic C and a programming cable. No external emulator or simulator of any kind is required for this board. Because of its low cost and low power design, this prototyping board is the widely used hardware nowadays in the market for net camera interface applications. It has a specific user interface of LED light, which glows for different kinds of errors and indicating different kinds of operating conditions. These features allow RCM 3365 prototyping board to be used worldwide.

1.2 Objective of the Project

The main objective of this project is to develop an embedded net camera interface using Rabbit RCM 3365 prototyping board. To create a database of the images captured and implement B-Tree search algorithm for search within the database.

The specific objectives of the overall project could be described as:

1. With the help of motion sensor, capture images for different movements
2. Storing the captured image on to the memory of the prototyping board
3. Saving the data from the onboard memory to the hard diskette drive of the machine
4. Creating a database management system with all the captured images and implementing a B-Tree search algorithm
5. Develop a Graphical User Interface, through which operation and controlling of the prototyping board interfaced camera can be done. In addition, the same GUI can be used to access and handle the database of the images captured by the camera.

This project report describes the hardware of the RCM 3365 prototyping board, its interface with the embedded camera. It also includes a graphical user interface to control the motion of the camera and the two servo motors. A description of the software code written in Java for image database management is included in the report. Along with this, a B-Tree search algorithm is also included in this report.

Chapter 2

PROJECT RESOURCES: HARDWARE

This project is performed on a development board RCM 3365. Different components of the Rabbit core development kit includes a fully improved Rabbit[®] 3000 microprocessor, different memory which includes SRAM, flash memory (for program, fixed flash memory and also removable NAND flash), 10/100 Base-T Ethernet port, six serial ports, digital I/O ports, a serial programming cable with 10 pin header and DE9 connections. My project also includes a CMOS image sensor connected with two servo motors.



Figure 2.1: Prototyping Board RCM 3365

2.1 Rabbit^R 3000 Microprocessor

Rabbit Semiconductor designs the microprocessor for many different applications. They even design the prototyping development board for various applications available in the market. The development board has different components in it which include the Rabbit core processor, one each in every development board. In RCM 3365 development board used in this project has Rabbit^R 3000 core processor mounted.



Figure 2.2: Rabbit^R 3000

2.2 Different Memory Modules and I/O Interface

2.2.1 SRAM

Prototyping board RCM 3365 has 512K of SRAM for rapid access to the code for execution, at the port U11 [4]. It acts as cache for storing the code of the program to be executed. It also has another 512K of SRAM, which runs on battery even if it loses power.

2.2.2 Flash EPROM

Prototyping board RCM 3365 has 512K of flash EPROM for writing and reading some static or permanent data. This memory is at port U9 [4]. Functions like WriteUserBlock and ReadUserBlock are used to access this memory frequently.

2.2.3 NAND Flash

Prototyping board RCM 3365 has on board support of 32MB of NAND flash. It also supports an additional xD-Picture Card™ [4]. It can handle up to 128MB of external memory meaning this xD-Picture Card could be up to 128MB wide. Different features namely reliability and integrity of the data is achieved by the use of NAND flash because it supports ECC (Error Correcting Code) [4]. This prototyping board has wide application range, which requires huge data storage. For application like taking picture as it is in my case requires a memory card which can hold up to hundreds of images. NAND flash is mostly used for mass data storage and not for storing program code. In applications with multiple reads and writes to the memory for storing a large amount of data NAND flash is very useful. It also guarantees low rate of error in the data, as low as 1 in 100,000 writes [4].



Figure 2.3: Insertion and Removal of xD-Picture Card (64MB)

2.3 Different Types of Ports

2.3.1 Ethernet Port

There are two types of Ethernet port (RJ-45) cable available in the market; first one is the straight cable and the second is the crossover cable. Straight cable has the same color code at both the ends of the cable while the crossover cable has different color codes at both the ends. Each cable has their own applications of data transfer. There are three LEDs next to a RJ-45 Ethernet port in the prototyping board. First one is for showing Ethernet link, second is for the link activity (ACT), and the third is for the 10/100 Base-T speed. [4]. There are two different modes of speeds available with 10/100 Base-T, full-duplex and half-duplex mode of operation. In this project I have used 10/100 Base-T half-duplex mode of operation.

2.3.2 Serial Ports

There are six serial ports namely A, B, C, D, E and F. They all operate at a baud rate of max. clock divided by 8 in asynchronous mode. Port A is used as programming port, port B is used as asynchronous port, port C, D, E and F can be used as serial ports, where C and D can be operated with serial clocked pins and E and F can be operated with HDLC [4].

2.3.3 Serial Programming Port

There are two ways of communication with Rabbit microprocessor. One is to use the programming cable, description of which follows in the next section. Second is to program it with the RJ-45 Ethernet port. When programmed with programming cable

port A can be used as the programming port. In this project, I have used the programming cable with port A to program the Rabbit microprocessor. There is also a RESET_IN pin to reset the entire board and start the board in run mode.

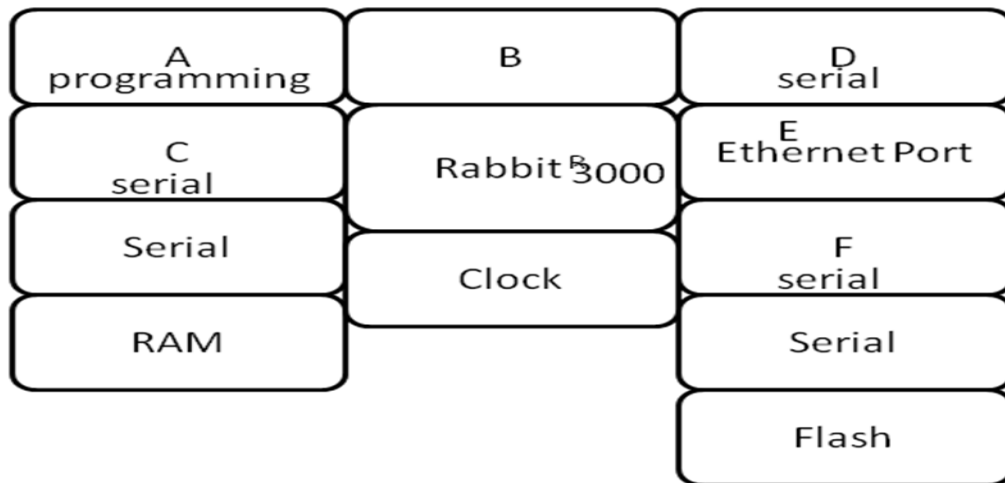


Figure 2.4: Use of Rabbit 3000 Ports

2.4 Serial Programming Cable

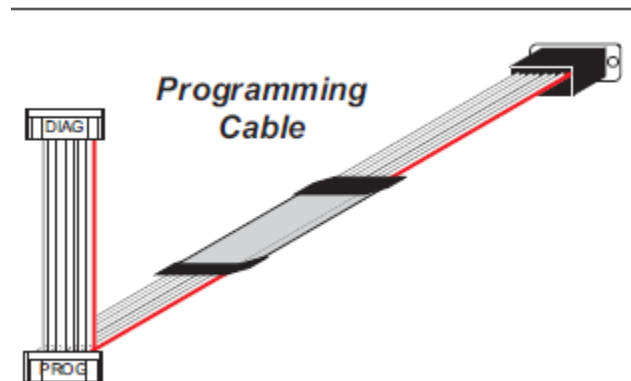


Figure 2.5: Serial Programming Cable [4]

Chapter 3

PROJECT RESOURCES: SOFTWARE

3.1 Overview

As discussed earlier, the entire project is divided into two parts: Hardware and Software. In the Hardware portion of the project, I used Rabbit Microprocessor development board to capture remote images and store it on to the desktop hard drive. Now, in the software portion of the project, I, have to create a graphical user interface through which a user can insert, delete and view the images captured. To create this web application through which a user can interact with the images captured, there were a couple of software resources that I used. Different software resources that I am using in this project includes tools like Eclipse Ganymede for J2EE, the language pack of J2EE and implemented it on to the windows xp platform. I also have to implement the Binary Tree search algorithm in the folder database that I created to store the images. The following sections mention detailed description of the different types of tools and packages used and also about different types of tree structures for database that are implemented in different database tools. It also includes detail description of B-Tree algorithm implemented in my project.

3.2 Tools and Packages

3.2.1 Eclipse Tool

The tool used here in my project to deploy and run the java program created is Eclipse IDE for java EE developers. It is a free tool available online for download. Following is a brief description about installing Eclipse and creating first java program using Eclipse.

Eclipse is used mainly as an Integrated Development Environment for java and java EE. It has its own different applications for different areas, but is mainly used as an IDE for different platform applications. Eclipse is created by an open source community and hence it is free to download online and use. Installation for Eclipse is very standard other than that it requires java runtime environment already installed on the system. Java runtime environment required to run java program in Eclipse is jre 6.0 or later. So, firstly jre 6.0 or later version is required to be installed and then installing Eclipse is a simple standard software program installation. Starting Eclipse is simple double click on the icon. Eclipse is provided with workspace, perspective, views and editor [7]. Workspace is the physical location in the drive on the machine. Perspective, views and editor are used to work around the layout or hierarchy of database and to open the editor. From within this we can make changes to different views and layout, for opening up the editor.

Creating the first java program has simple steps to be followed. I am representing those steps in form of pictures as follows:

The next step in the process is as follows:

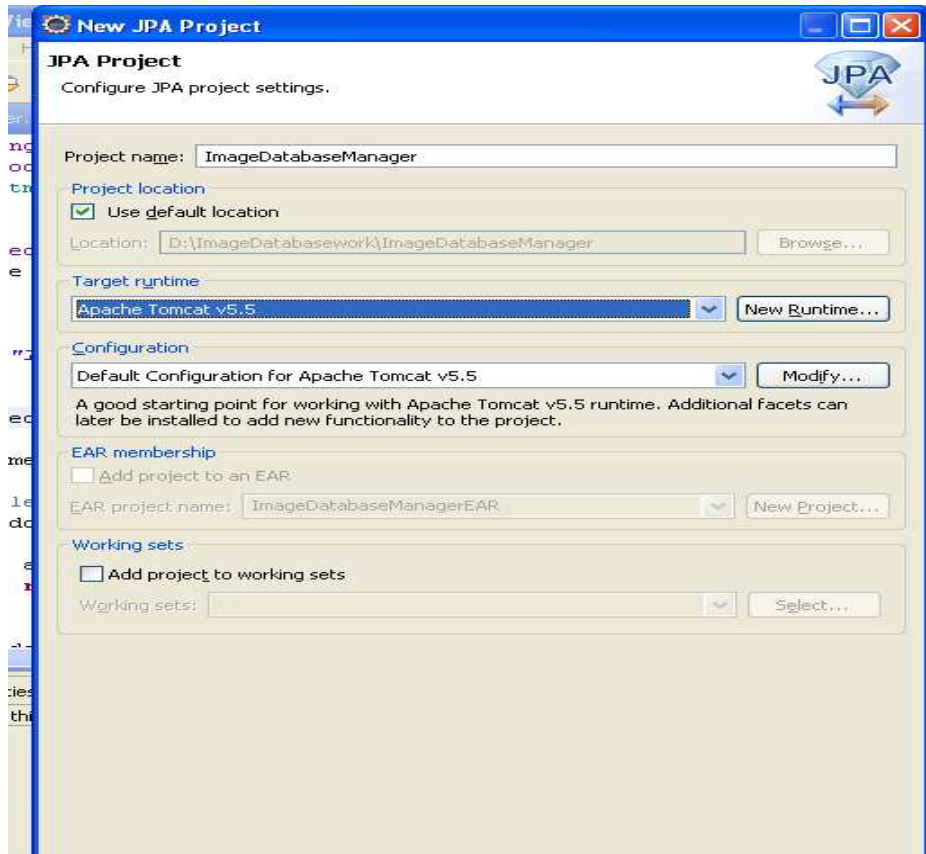


Figure 3.1: Adding Package and Server to the New File

After the package and the program code are added, it is very simple to run the program following the instruction in the following figure.

3.2.2 Language Package

The language used in this project for coding is Java. Java is a very high-level object oriented programming language. There are two different point needed to be pointed out at this point of time. First, Java language for coding and the second is the Java runtime environment through which a Java application runs. I am using Java language for coding the required web application of implementing the B-Tree search

algorithm, which is described in the next section. I am using the Java EE i.e. Java Enterprise Edition platform which is a runtime environment for running the java applications. Java platform consists of two separate and most important things namely Virtual Machine (VM) and Application Programming Interface (API) [8]. Virtual Machine is the program that runs the java applications and Application Programming Interface is the software components through which one can create more software components and applications. This feature supported by Java Platforms allows the java program to run on different platforms with very easy deployment and with full security feature. Java EE, the platform used in this project is built on Java SE (Standard Edition) and it has the API and environment to be used for very large applications. Coding using different object oriented programming concepts makes implementation of the problem very easy. With the help of feature like class and running methods like 'post', it actually becomes a little easy in implementing a huge task. Creating a web application is the beginner mode is easy using java, but using it with database management system asks for quite a good research work. I am not using any database here in my project to achieve the goal of the project, but instead I am using a local database on to the local host to achieve different functionalities required in the project that serves the project requirement very well.

3.3 Binary Search Tree Algorithm

Binary search tree algorithm is a very important aspect of my project. In the hardware portion of the project, I am using the Rabbit Microprocessor development board and the integrated camera to capture the images and store it on to the hard drive. In

the software portion of the project, I am creating a web application for inserting, deleting and viewing the image. In addition, implementing the binary search tree algorithm on the folder used for image database management.

Binary tree is a node based tree data structure. It is also known as sorted binary tree algorithm [9]. A node is placed in the middle of the entire data structure. All the nodes with keys less than the node's value are placed on left of the node. All the nodes with keys greater than the node's value are on the right side of the node. Again, both the sub tree nodes are binary search tree. The hierarchy of the algorithm continues in the similar manner for as many numbers of nodes we want to enter. The binary search tree algorithm can be explained neatly using the following diagram.

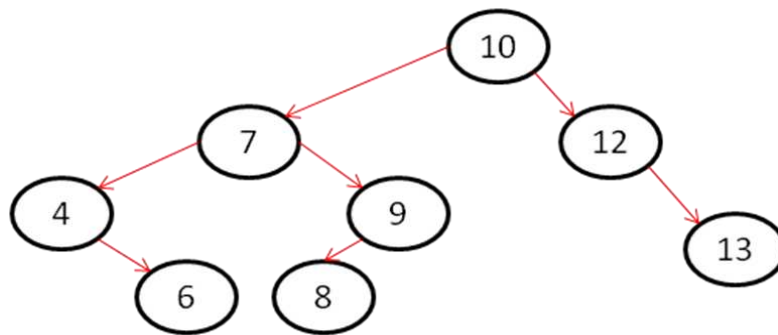


Figure 3.2: Binary Search Tree with Nodes and Subtree

Actually how does a binary tree search works can be explained using the following flow chart. Below is the flow chart which shows the implementation of the binary search algorithm for $N = 8$ number of nodes. In applications related to database, the database management tools like MySQL, Oracle and many others have already implemented the binary search algorithm for metadata indexing. All the data is inserted into the database following the rule of binary tree. If in case any changes needs to be

made to the implementation of binary tree it then becomes easy, as now we only have to implement the changes and not the entire algorithm as binary search tree is all supported in all the major database management tools available in market today for metadata indexing.

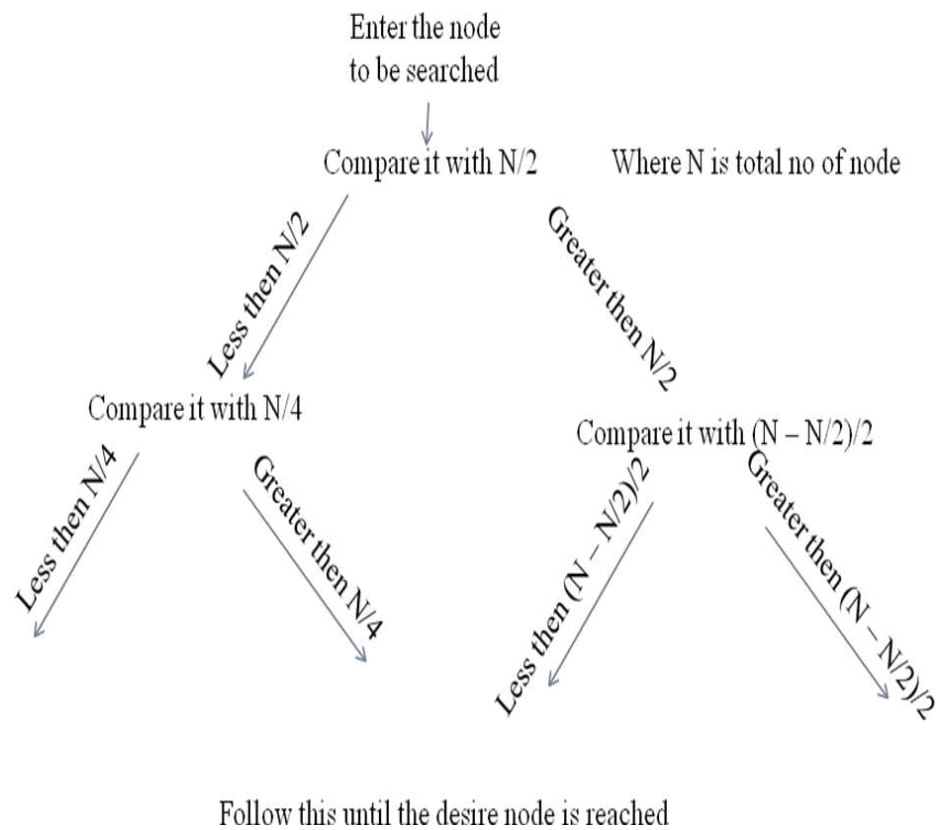


Figure 3.3: Flow Chart Showing Search Algorithm Implemented in binary tree

Three basic actions taken on the binary tree sorted database are insertion, deletion and sorting or reordering the nodes. Usually the nodes contain the record rather than just the data. Whenever a node is inserted in the tree data structure, it is always first checked against the first node that it encounters. If the inserted node is less than the node already present than it traverses on to the left sub tree of the tree data structure. If the node that is

inserted is greater than the node already, present than the inserted node is forwarded to the right sub tree structure. This process of forwarding the inserted node to either left sub tree or right sub tree depending on the value continues until the inserted node reaches a place where it is the final node. Same algorithm applies for deletion of the node. Any random node that we want to delete is checked for its value either less than or greater than the node and finally it is found in the tree data structure and deleted from wherever it was present. After deleting any node from within the tree data structure it rearranges or reorders to maintain the binary tree structure for the entire database.

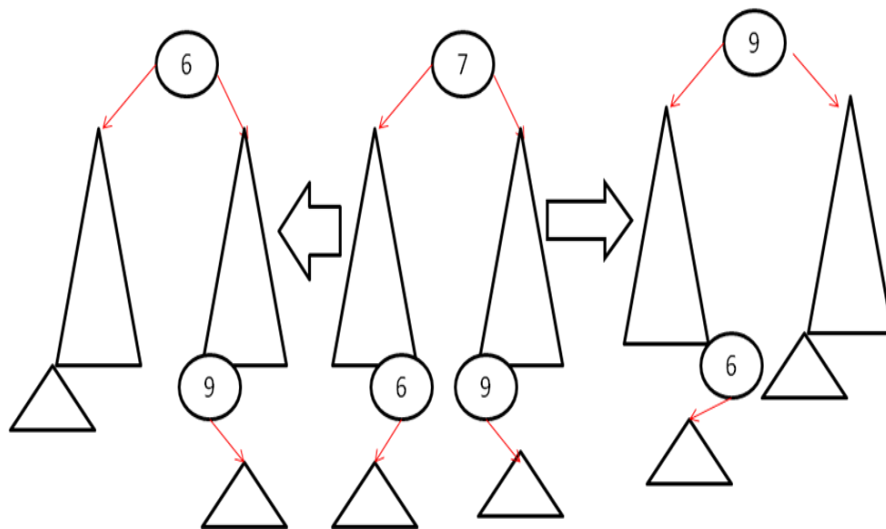


Figure 3.4: Deleting a Node from binary tree structure

There are other binary tree data structures available in the market. They are B – Tree, B + Tree and B * tree. They are all similar to each other with a little change in the implementation. Different binary tree data structures available in the market are found implemented in the database systems currently used. Database management tools like

mySQL, Oracle and many others have binary tree search algorithm already implemented in them.

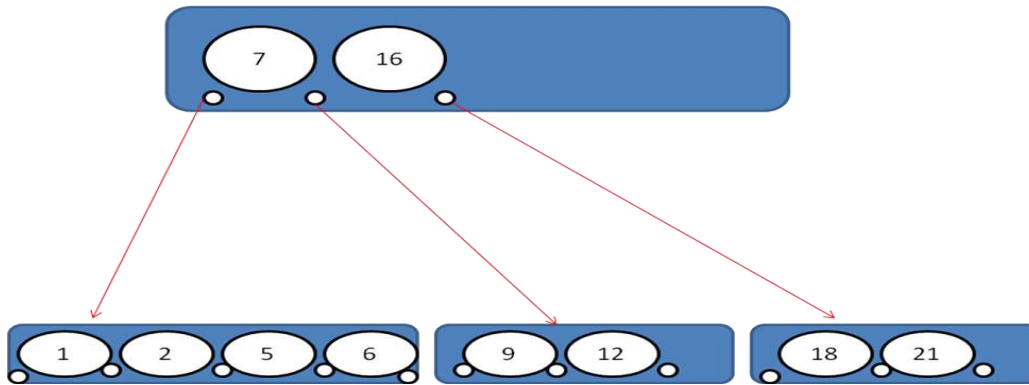


Figure 3.5: B – Tree of Order 2

The third tree in the hierarchy is the, B* -Tree, which follows the B – Tree and B + Tree. A B* -Tree is an ordered binary tree in the entire hierarchy. It requires the non root nodes to be filled at least 2/3 to maintain it. It follows two steps for inserting or removing the data from its nodes namely, clustering and de-clustering [10]. It has the flexibility of sharing the keys with the next node in line, if in case the present node gets full. It tries to share the keys with the next node for some time before splitting up the node. When both of them are full then and then only the two nodes are split into three nodes. It is not as efficient in searching than other binary trees available in the market. B +Tree is more efficient in searching the database but it requires a lot more insertion to be done in the process, as all the nodes are connected in the same manner as in linked list in C.

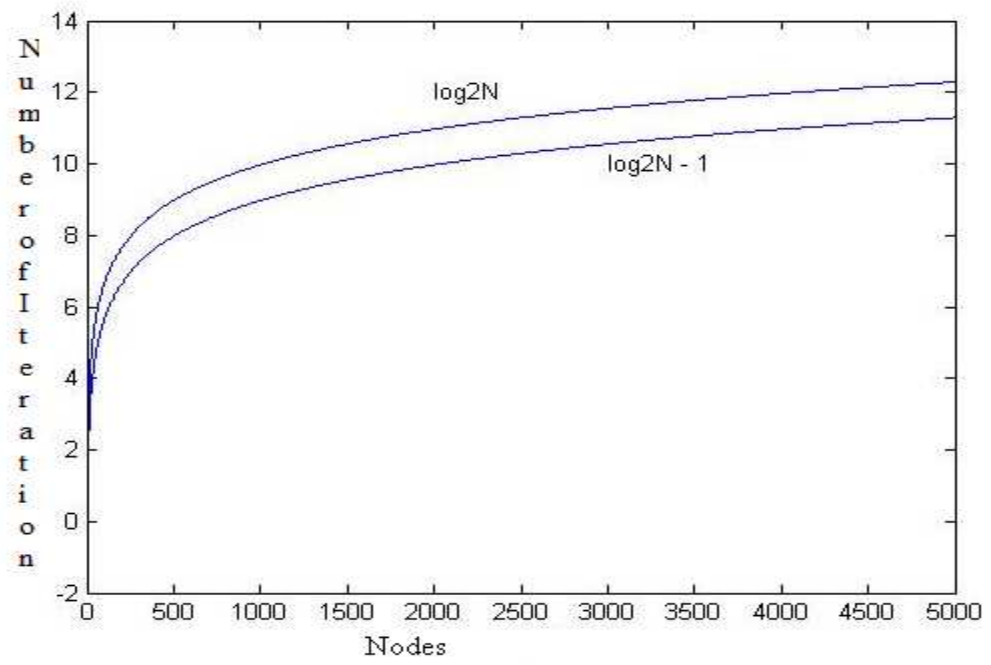


Figure 3.6: Best Case ($\log_2 N$) v/s Worst Case ($\log_2 N - 1$)

Chapter 4

PROJECT DETAILS

This project mainly deals with hardware part in which with the help of Graphical User Interface one can control the hardware. In the Software part one can manage the database. Graphical User Interface created gives user a webpage through which he can control the servo motor movements also capture the image and many other things. The other part of this project is the software part in which java coding is done to create a user interface to handle and manage the database of the images. These images are captured from the camera module on the prototyping board. Therefore, this project is divided into two parts:

1. Windows based Graphical User Interface to control the hardware
2. A user interface to handle and manage the database through software

Following is the flow chart for the entire design.

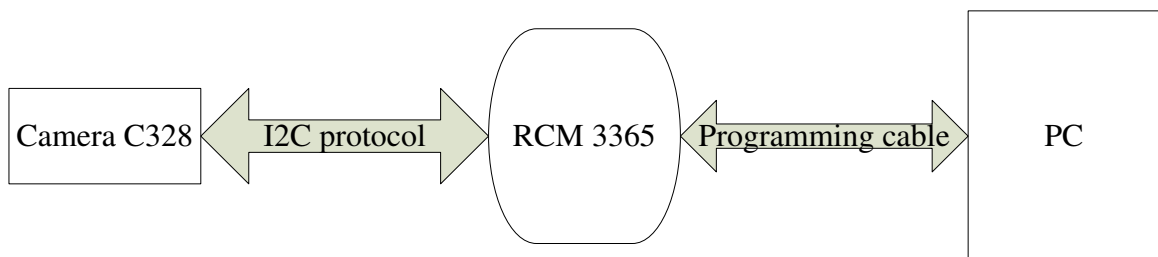
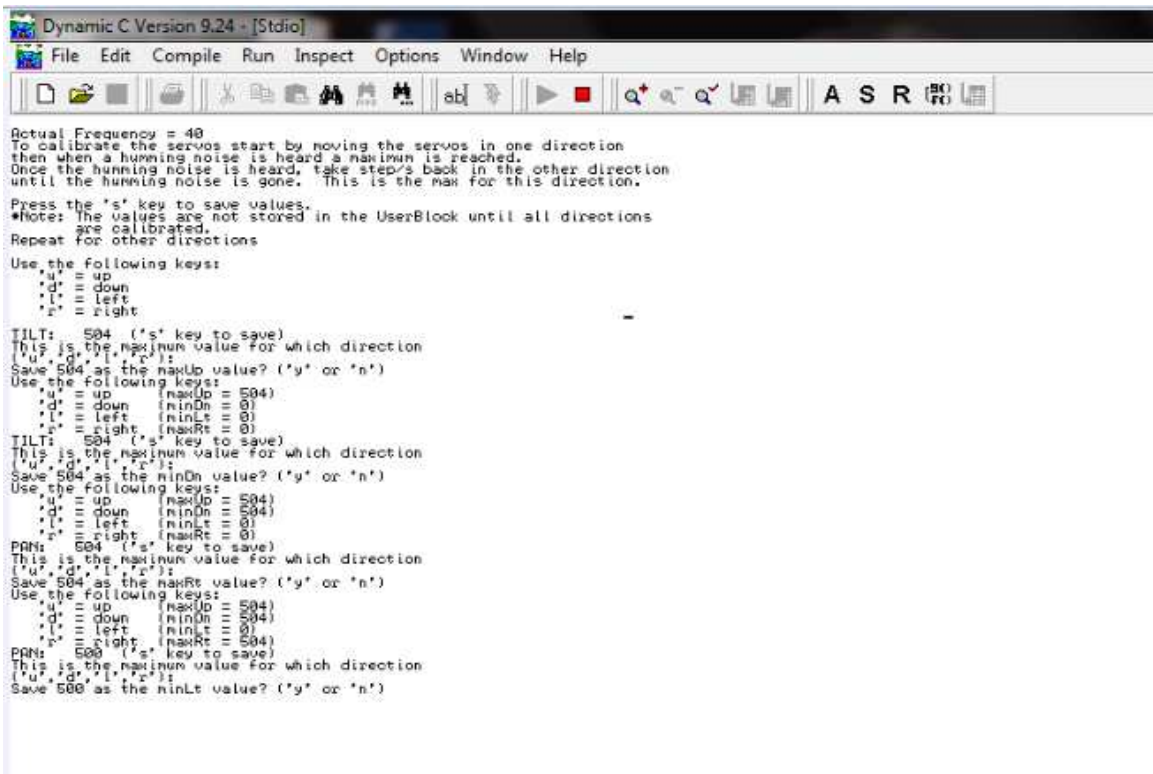


Figure 4.1: Block diagram for hardware connection

4.1 Camera C328 CMOS Image Sensor and Servo Motor

The camera used in this project is C328 VGA camera module. It is JPEG compression type camera module which supports a wide variety of resolutions like 640*680 (VGA)/ 320*240 (QVGA)/160*128/80*64 [5]. It is low power, 3.3V DC operating at 60mA current, low cost camera module, which uses RS-232(TTL) interface at a rate of 115.2kbps with 0.75 frame per seconds. It runs in two different modes, either it gives output data in JPEG compression mode or the data is output RAW. It is connected with two micro-servo motors, which generates a torque of 0.4kg-cm when operated at 4V DC turning clockwise at a speed of 1500-1900 microseconds. These servo motors are also low power, low cost motors available in the market.



```

Dynamic C Version 9.24 [Stdio]
File Edit Compile Run Inspect Options Window Help
[Icons] [ab] [Play] [Stop] [Search] [Print] [A S R]

Actual Frequency = 48
To calibrate the servos start by moving the servos in one direction
then when a humming noise is heard a maximum is reached.
Once the humming noise is heard, take steps back in the other direction
until the humming noise is gone. This is the max for this direction.

Press the 's' key to save values.
*note: The values are not stored in the UserBlock until all directions
are calibrated.
Repeat for other directions

Use the following keys:
'u' = up
'd' = down
'l' = left
'r' = right

TILT: 504 ('s' key to save)
This is the maximum value for which direction
('u', 'd', 'l', 'r'):
Save '504' as the maxUp value? ('y' or 'n')
Use the following keys:
'u' = up (maxUp = 504)
'd' = down (minDn = 0)
'l' = left (minLt = 0)
'r' = right (maxRt = 0)

TILT: 504 ('s' key to save)
This is the maximum value for which direction
('u', 'd', 'l', 'r'):
Save '504' as the minDn value? ('y' or 'n')
Use the following keys:
'u' = up (maxUp = 504)
'd' = down (minDn = 504)
'l' = left (minLt = 0)
'r' = right (maxRt = 0)

PAN: 504 ('s' key to save)
This is the maximum value for which direction
('u', 'd', 'l', 'r'):
Save '504' as the maxRt value? ('y' or 'n')
Use the following keys:
'u' = up (maxUp = 504)
'd' = down (minDn = 504)
'l' = left (minLt = 0)
'r' = right (maxRt = 504)

PAN: 500 ('s' key to save)
This is the maximum value for which direction
('u', 'd', 'l', 'r'):
Save '500' as the minLt value? ('y' or 'n')

```

Figure 4.2: STDIO Window Output for servo_calib.c



Figure 4.3: Micro Servo Motor

4.2 Rabbit Microprocessor 3000

Rabbit semiconductor mainly designs processor for low-level applications. Rabbit microprocessor designer's designs processor for embedded computer application for use in MSI (Medium Scale Integration) and SSI (Small Scale Integration). Rabbit[®] 3000 is an 8-bit processor supporting 8-bit architecture and operating at a speed of up to 55.5MHz. It supports up to 6MB of memory for code and data. The supply voltage range required for normal operating condition is 1.8V to 3.6V. It is smaller size 10*10*1.2 mm and wide range of operating conditions allows it to used worldwide. It has a total of more than 62 ports including digital and serial input output ports. Address and data bus are 20bit and 8bit wide respectively and it has 8 address and 8 data external I/O bus. It is 128 pins 0.8mm BGA (Ball Grid Array) package. It supports 10/100 Base-T half-duplex mode for Ethernet connection. There are two timers available for use with Rabbit[®] 3000 namely, Timer A and Timer B. Timer A has ten 8-bit counters and Timer B has a 10-bit counter, two match registers and two-step registers. It also supports two modes of operation known as system and user modes. It keeps the two modes isolated from each other thereby providing the security and safety for codes from being affected by the processor setup for different applications.

4.2.1 Set Up the Internet Connection and Run event_capture.c

This code, basically deals with the graphical user interface to access the camera module and also handle the servo motors. Capturing the image and storing those images on to the hard drive or email them to an email address is also a part of this graphical user interface. Before running this sample code there were some network connection issues that needed the attention. As this is graphical user interface via internet I needed to do some network connection settings to run this GUI and control the entire module. This code saves the captured image using the FAT file system on to the external drive connected i.e. NAND flash. So in case if the flash is full and we want to format the drive there is also an option to do that. Running this program on the module will give the STDIO window output as shown below.

After this program runs the next step in the process is to configure the PC to be able to use the GUI and access the entire module from the internet. The PC configurations for a windows based system are as follows:

First of all go to the control panel > network connection settings.



Figure 4.4: Network Connection Setting for Windows based PC

Then select the network interface card that is present in the PC and configuring the properties of the network card to work at 10/100 Base-T half-duplex speed [6]. Then selecting the IP configuration tab and entering the numbers in the specific column as mentioned below. In the IP address tab, enter the value 10.10.6.101, then in the subnet mask entering the value 255.255.255.0, and entering the default gateway to be 10.10.6.1 [6].

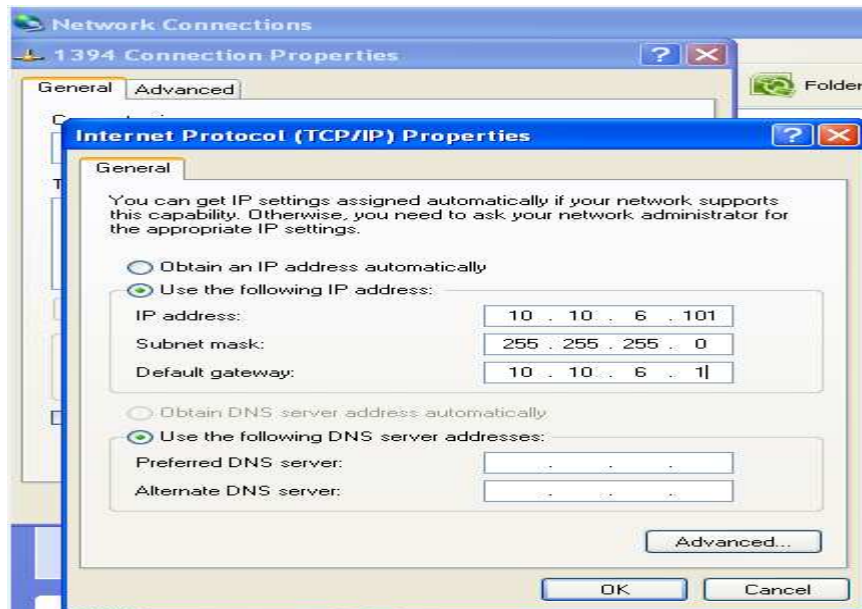


Figure 4.5: Network Connection Setting for Windows based PC

Now, it's the turn of connecting the PC and the development board with the straight cable with RJ-45 connector discussed above. Once this is done, I disconnected the programming cable and pressed reset on the kit. This will make the kit work in run mode. To use the GUI built, open up the browser and enter the address as `http://10.10.6.100`, once we enter this address it asks for a username and password already set default for the kit. The username is 'username@isp.com' and password is 'test' [6]. This allows us to use the GUI built to handle the camera module and servo motors. This GUI looks like the following figure.



Figure 4.6: GUI for event_capture.c

With the radio buttons 'up', 'down', 'left', 'right' I can manipulate and operate the servo motors: pan and tilt both. With the radio button 'capture', I can capture the image as shown above in the figure. Right clicking on the image will give the option of saving the image to the desktop hard drive on that PC. We can also adjust the resolution of the image from there on the GUI selecting different image resolutions from the drop down list next to the radio button 'Resolution'.

4.3 Software Code Implemented in Java

4.3.1 Binary Tree Implementation in java

The following code describes implementation of binary tree search algorithm in java.

```
//This function uses the binary search algorithm
```

```
public static int binarySearch(int[] sorted, int first, int upto, int key) {  
    while (first < upto) {  
        int mid = (first + upto) / 2; // Compute mid point.  
        if (key < sorted[mid]) {  
            upto = mid; // repeat search in bottom half.  
        } else if (key > sorted[mid]) {  
            first = mid + 1; // Repeat search in top half.  
        } else {  
            return mid; // Found it. return position  
        }  
    }  
    return -(first + 1); // Failed to find key  
}
```

Static function which returns integer type is declared. It reads as argument the integer value inserted by the user which is recognized as the first key and the last value entered as the upto key. The loop runs from the first key entered up to the last key entered. Firstly, the variable mid is calculated as the midpoint of the two values. The key value entered is now compared to the value entered for search in the database. Now a decision is made whether the entered key is less than the midpoint value or greater than the midpoint value. Depending on this answer the loop is continued either in the first half, which includes the values from the first key to the midpoint key. If these answer is in the second half than the above loop is continued in the later half of the database.

Whenever there is a node to be searched, it is inserted in the flow chart shown in Figure 3.5. Let us take an example where we have total number of nodes N equal to 8. If I want to search value at node 3, first of all the node 3 is compared to total number of nodes, in this case $N = 8$. Comparing it with $N/2$ i.e. 4, we know that 3 is less than 4 and not greater than 4. So the tree is now followed into left direction. If the node is 6 the tree would be followed in right direction. Now, in our case we know that it is on the left subtree of the binary search tree. Hence, node 3 is now compared with $(N-N/2)/2$ i.e. 2. The output is that 3 is greater than 2, which implies that the node we wanted to search is on right side of 2 and on left subtree of 4. So we come to know that the node is 3 and also the place where it resides inside a binary tree implementation. $\log_2(N)-1$ is the regular expected number of hops it would take to reach a particular node. In the worst-case scenario, it will take $\log_2 N$ hops to reach a particular node.

4.3.2 Sorting the Image

Following is the code which shows how the image is sorted into the database depending on the variable 'Imagename'.

```
//Sorting the Integer Array
Arrays.sort(intArr);
int mid =0;
try
{
```

```
mid=binarySearch(intArr,0,intArr.length,Integer.parseInt(imageName));  
    }  
    catch(Exception e)  
    {  
        mid=0;  
    }  
    System.out.println(mid);  
    if(mid >= 0)  
    {  
        System.out.println(intArr[mid]);  
    }  
    else  
    {  
        return "File does not exist.";  
    }  
}
```

A function 'arrays.sort' is called. This function is the default function available in java library files. A variable which holds the middle value is initialized to zero. After this the binary search algorithm implemented above is passed to the variable which holds the middle value. And the function in turn returns the image names in sorted order.

4.3.3 Pseudo Code for Binary Search B - Tree Algorithm

//Purpose: find Item X in the Tree

Inputs: data object X (object to be found), binary-search-tree node node

// Output: bst-node n containing X, if it exists; NULL otherwise.

Function find(X, node)

if(node = NULL)

return NULL

if(X = node:data)

return node

else if(X < node:data)

return find(X,node:leftChild)

else // X > node:data

return find(X,node:rightChild)

Pseudo Code for Finding the Minimum is as Described Below

//Purpose: return least data object X in the Tree

//Inputs: binary-search-tree node node

// Output: bst-node n containing least data object X, if it exists; NULL otherwise.

Function findMin(node)

if(node = NULL) //empty tree

return NULL

if(node:leftChild = NULL)

```
return node
```

```
return findMin(node:leftChild)
```

Pseudo Code for Insertion Into Binary Tree

```
//Purpose: insert data object X into the Tree
```

```
//Inputs: data object X (to be inserted), binary-search-tree node node
```

```
//Effect: do nothing if tree already contains X;
```

```
// otherwise, update binary search tree by adding a new node containing data
```

object X

```
Function insert(X, node)
```

```
if(node = NULL)
```

```
node = new binaryNode(X,NULL,NULL)
```

```
return
```

```
if(X = node:data)
```

```
return
```

```
else if(X < node:data)
```

```
insert(X, node:leftChild)
```

```
else // X > node:data
```

```
insert(X, node:rightChild)
```

Pseudo Code for Deletion From Binary Tree

```
//Purpose: delete data object X from the Tree
```

```
//Inputs: data object X (to be deleted), binary-search-tree node node
```

```

//Effect: do nothing if tree does not contain X;

// otherwise, update binary search tree by deleting the node containing data object

Function delete(X, node)

if(node = NULL) //nothing to do

return

if(X < node:data)

delete(X, node:leftChild)

else if(X > node:data)

delete(X, node:rightChild)

else f // found the node to be deleted! Take action based on number of node
children

if(node:leftChild = NULL and node:rightChild = NULL)f

delete node

node = NULL

return

else if(node:leftChild = NULL)f

tempNode = node

node = node:rightChild

delete tempNode

else if(node:rightChild = NULL)f

(similar to the case when node:leftChild = NULL)

else f

```

```
//replace node:data with minimum data from right subtree  
tempNode = findMin(node.rightChild)  
node:data = tempNode:data  
delete(node.data,node.rightChild)
```

4.3.4 Web Interface Using Java

To create the web interface, I wrote script file in java. These javascript(.jsp) file has different radio buttons in it to select either of the option to insert, view or delete the image from or within the database. These script file in turn calls the java program which has all the function for inserting the image, viewing the image and also to delete the image. Following is the block diagram of the software code running in java language. The java file ImageDatabaseManager.java calls the ImageDatabaseViewer.jsp through the 'dopost' method. Post method is declared in the jsp file with a link to the java program. This can be visualized by the following code lines.

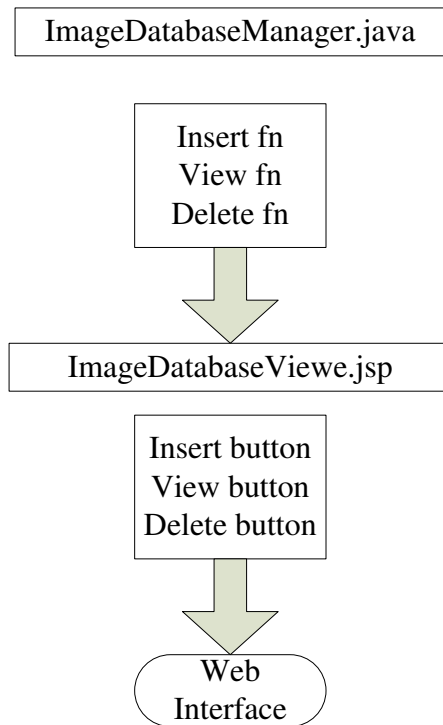


Figure 4.7: Block diagram-connection between front end and back end design

The user interface created is the front-end design, which is open to the user. This is the only part of the design, which is in direct link to the user. Java program and the java script files are running in the background handling and managing the image database.

A web application using jsp programming is created through which a user can interact with the application created in java through java programming. The user interface gives the user ways to handle and manage large amount of images. The graphical representation of the web application looks like the following Figure 4.3.

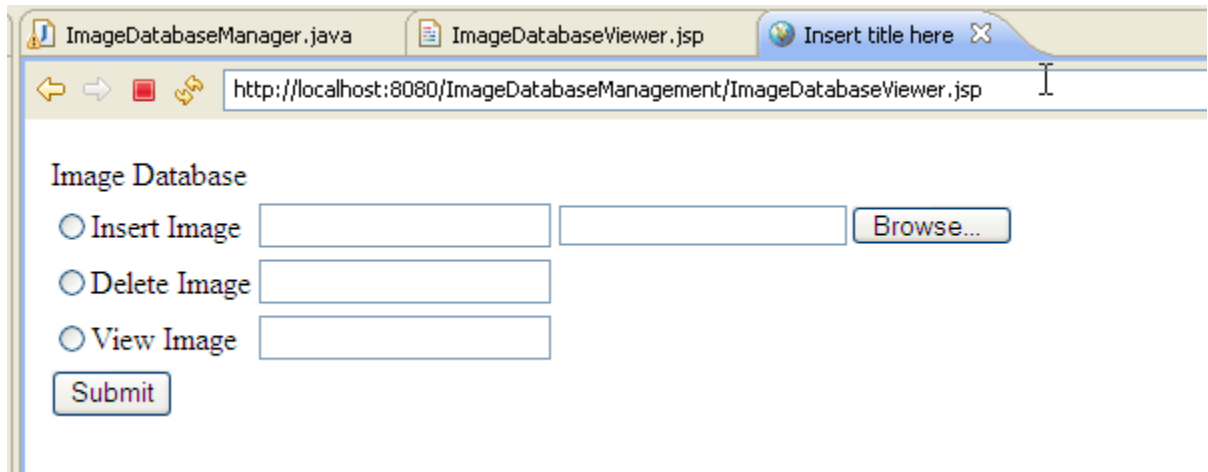


Figure 4.8: GUI of Web Application in jsp

Figure 4.8 shows the GUI of web application created in jsp. This application is running, as discussed in the earlier section, on apache tomcat server version 5.5, and the tool used to run this application is Eclipse jee Helios for windows 32. Figure 4.3 shows the application, which has three radio buttons in it. 'Insert Image', 'Delete Image' and 'View Image' are the three options used to handle the image database. Final selection of the action is done using the 'submit' button. As soon as the 'submit' button is hit the respective action from the radio buttons is performed.

4.4 Limitation of the Design

The design created here has its own limitations in the real world applications. The web interface created using java is running on the local host. I used eclipse tool to run the java program and the java script on the local host. Connecting the design with the database management tools like MySQL, Oracle can be the next phase of the project. As the design runs on the local host it could be configured and coded in such a way that it

can run online on internet. Live web application is the limitation of this project. If an interface could be found that can read the image, which is captured by the Rabbit microprocessor automatically, then this design can be integrated with the online database. Once the interface read the dynamic image captured by the Rabbit microprocessor and transferred it to the web application, this developed project would be able to run on internet with rabbit interface.

After this program runs the next step in the process is to configure the PC to be able to use the GUI and access the entire module from the internet. The PC configurations for a windows based system are as follows:

Chapter 5

SIMULATION AND RESULTS

5.1 Inserting the Image

When the 'Insert Image' radio button is selected the application is supposed to insert the image which is browsed into the second text box to the images folder. Also, here the user can give the image an ID in the first text box to keep a track of the image he inserted into the database.

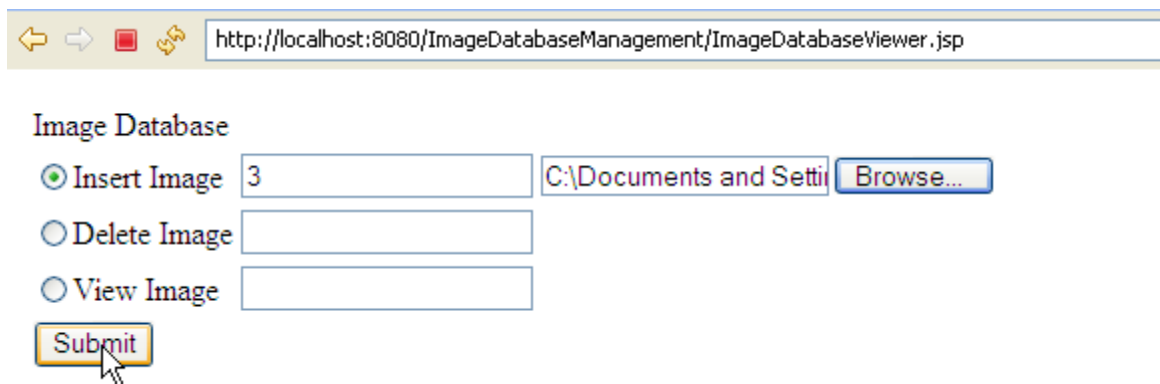


Figure 5.1: GUI for Inserting Image

In the first text box, image ID is entered to keep the track of the image and in the second text box with the browse button the original source path of the image is entered from where we want to insert the image. After entering the information in both the text box, the 'submit' button is hit which display the message shown in the next figure 5.2.

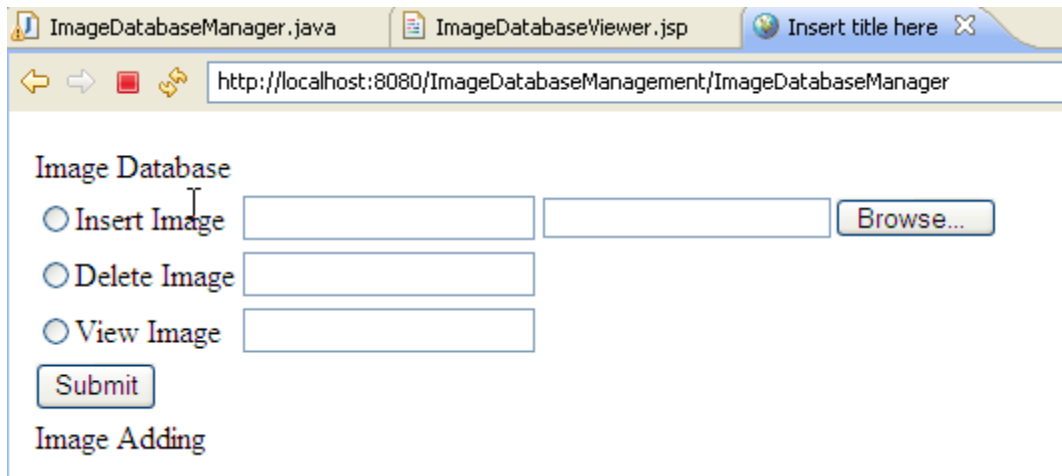


Figure 5.2: Message Display Stating Image has been Added to the Database

In the next figure 5.3, it is shown very clearly that the image mentioned in figure 5.2 has been added to the database entry table and also we can see that the image has been added in the database folder specified by me.

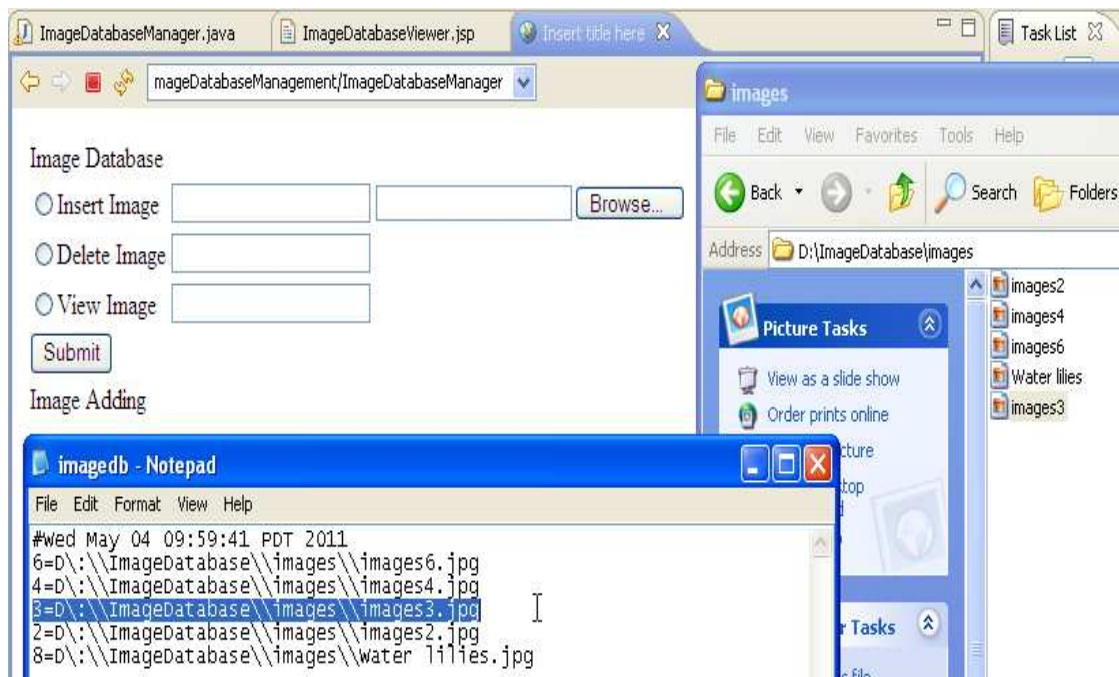


Figure 5.3: data entry in the Dable and the Database Folder

5.2 Viewing the Image

The same application has one more radio button ‘View Image’. If this radio button is selected and the image ID is entered in the first text box next to it, the image related to that particular image ID from the database is opened in mspaint. The procedure of performing this step is shown in the following figure 5.4.

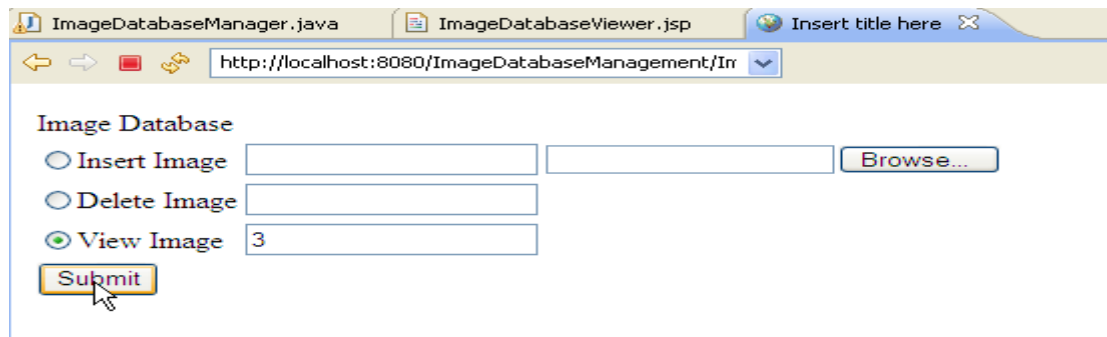


Figure 5.4: Procedure to Follow for Viewing the Image

After this step is followed and the button ‘submit’ is hit, the application goes and looks for the image in the database entry and the respective image is opened in the program mspaint.

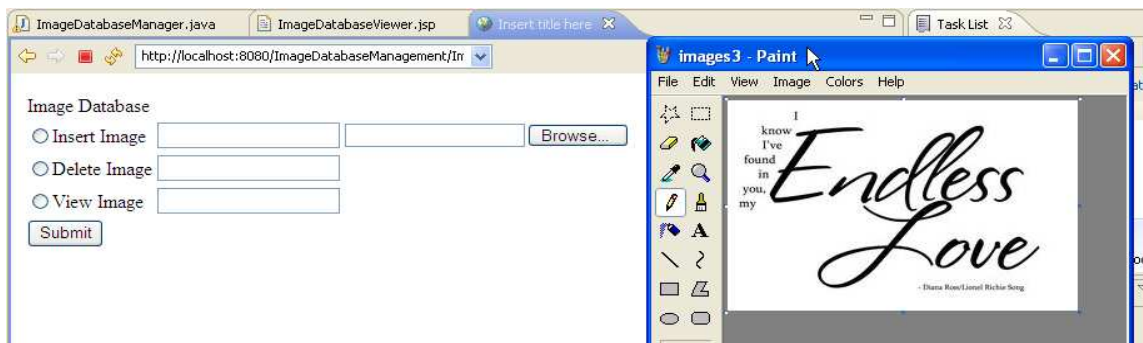


Figure 5.5: Image Opened in mspaint

5.3 Deleting the Image

There is a simple procedure to be followed while deleting the image same as done in viewing the image. The image we want to delete has a particular ID in the entry table, so the image ID is entered in the first text box next to the radio button ‘Delete Image’.



Figure 5.6: Procedure to Delete the Image

After following the procedure of entering the image ID in the text box and then clicking on the ‘submit’ button as shown in figure 5.6 the next image displays the message ‘Image Deletion Successful’. It can be confirmed from the next figure which shows the deleted entry from the table and also the image deleted from the images folder.

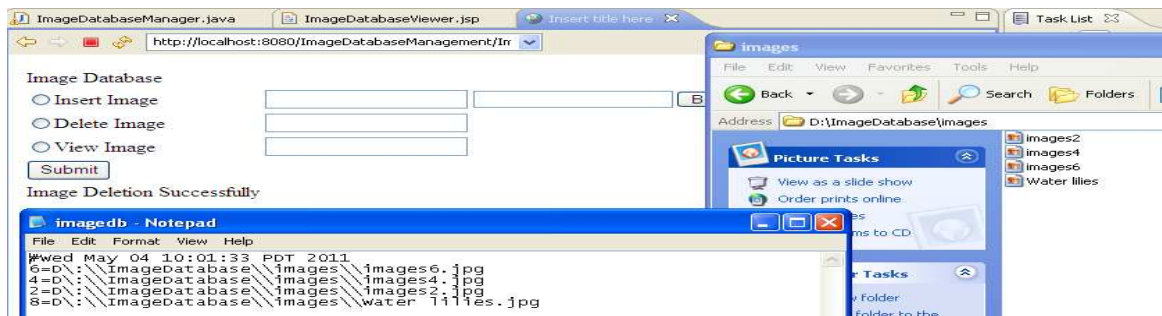


Figure 5.7: Deleted Table Entry and Image From the Database Folder

We can see from the table entry file that all the images are stored in a particular format following the binary tree search algorithm. If we enter the image ID = 2 for inserting the image into the database and then next image ID = 3, 4, 6 and 8 and so on, images are inserted into the database following the binary tree algorithm. Following two figures show the binary tree approach followed in the entry table and also the actual implementation when viewed in two dimensions.

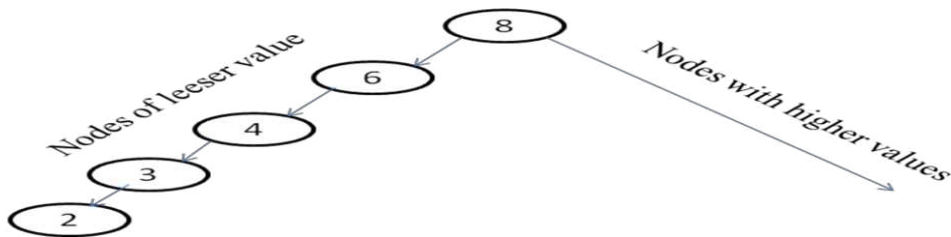


Figure 5.8: Implementation of binary tree

```

#Wed May 04 10:03:11 PDT 2011
6=D:\ImageDatabase\images\images6.jpg
5=D:\ImageDatabase\images\images5.jpg
4=D:\ImageDatabase\images\images4.jpg
3=D:\ImageDatabase\images\images3.jpg
2=D:\ImageDatabase\images\images2.jpg
8=D:\ImageDatabase\images\water lilies.jpg
  
```

Figure 5.9: Properties File Showing binary tree Implementation

Chapter 6

CONCLUSION AND FUTURE EXPANSION

6.1 Conclusion

As stated at the starting of the report and proved in the project, Rabbit microprocessor designers design Rabbit Core Microprocessors keeping in mind the applications closely linked to SSI (Small Scale Integration) and MSI (Medium Scale Integration). It has been proved over years that for small scale applications or in other words, for low-level applications, Rabbit microprocessors and the development board are more efficient and in budget processors. Applications like remotely monitoring the security systems, as implemented in my project, implementing the network security and access systems and many more like those are widely famous over the market. Very easy and user friendly development environment, and also crystal clear graphical Interface to access the entire module and also ease of use has made this processor to be used worldwide for small, remote and security related applications.

This development kit, “camera interface application kit RCM 3365”, is considered ideal for applications related to remote monitoring of different places like offices, houses or some confidential secret places, with greater efficiency [5]. It can also be configured to set up an alarm if something goes wrong or some movement is detected around it. It's easy use and rapid access to the system via internet makes it the best in the business for such an applications.

To summarize, I would like to make a note that Zigbee based Web Enabled power Monitor will soon serve as an efficient and cost effective application for controlling and monitoring the devices.

6.2 Future Expansion

Recent studies are going on the Rabbit Core Microprocessor's to use it for application related to a large scale entity. Research on RCM 3365 to extend its use for security feeds, where simultaneously many number of the same module can be connected with each other, so that they can be used for security of a bigger place and from various angles. Research on development kit, which can support large amount system memory on board is going on right now.

REFERENCES

- [1] Rabbit Semiconductor, (2005), “RCM3365 RabbitCore™”, USA
- [2] E Balagurusamy, (2004), “Programming with JAVA A Primer”, INDIA, pp 263-323
- [3] Camera Interface Application Kit, RCM 3365, [ONLINE], Available
http://ftp1.digi.com/support/documentation/0190150_g.html
- [4] Paulo Cappelletti, Carla Golla and Piero Olivia, (1999), “Flash Memories”, USA pp 243 - 264
- [5] Camera Interface Rabbit-based Photo and Motion Event Capture, [ONLINE], Available: ftp://61.4.189.51/Rabbit/CameraInterfaceKit/Camera_App_Kit_DS.pdf
- [6] Rabbit, Z-World, Embedded Control, [ONLINE], Available:
http://ftp1.digi.com/support/cdocumentation/0200105_a.pdf
- [7] Eclipse Project Release, IBM Corp. and Others, (2008), [ONLINE], Available:
http://www.eclipse.org/eclipse/development/readme_eclipse_3.4.2.html
- [8] Richard Neapolitan and Kumarss Naimipour, (2004), Foundations of Algorithms Using Java, UK, pp 48 – 71
- [9] Hsun-Cheng Lee, Yao-Wen Chang, Jer-Ming Hsu, and Hannah H Yang, (2004), “Multilevel Floorplanning/Placement for Large-Scale Modules Using B* -Trees”, USA