A PROJECT REPORT ON

INTERACTIVE TV SYSTEM

SUBMITTED TO THE UNIVERSITY OF PUNE, PUNE IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE AWARD OF THE DEGREE
OF
BACHELOR OF ENGINEERING
(ELECTRONICS & TELECOMMUNICATION)

SUBMITTED BY

ASHTEKAR SWAPNIL TUKARAM	**B3053008**
BARAWKAR ADITYA AVINASH	**B3053012**
CHETAN ARVIND PATIL	**B3053018**

Guided By

PROF. V. B. VAIJAPURKAR

PUNE INSTITUTE OF COMPUTER TECHNOLOGY
DHANKAWADI, PUNE- 411043

DEPARTMENT OF ELECTRONICS &
TELECOMMUNICATION ENGINEERING
2010-2011

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING
PUNE INSTITUTE OF COMPUTER TECHNOLOGY
DHANKAWADI, PUNE- 411043
2010-2011

Certificate

This is to certify that the project report entitled

INTERACTIVE TV SYSTEM

Submitted by

ASHTEKAR SWAPNIL TUKARAM	**B3053008**
BARAWKAR ADITYA AVINASH	**B3053012**
CHETAN ARVIND PATIL	**B3053018**

Is a bonafide work carried out by them under the supervision of **Prof. V.B. Vaijapurkar** and it is approved for the partial fulfillment of the requirement of University of Pune, Pune for the award of the degree of Bachelor (Electronics & Telecommunication Engineering.)
This project work has not been earlier submitted to any other Institute or University for the award of any degree or diploma.

Prof.V.B.Vaijapurkar
Internal Guide

Prof. Dr.Y. Ravinder
Principal & HOED , PICT

Place: Pune
Date:

ACKNOWLEDGEMENT

We would like to thank and acknowledge the invaluable efforts and guidance of our project guide Prof. V.B Vaijapurkar.

We would also like to thank our HOD Prof. Dr. Y. Ravinder and our project coordinator Mr. R.C. Jaiswal for their continuous support and help. Also we would like to thank all our professors and lecturers. Our thanks also goes to the department of Electronics and Telecommunication of PICT with whose help and co-operation we have been able to carry out our work on the proposed system.

LIST OF FIGURES

INDEX

1. INTRODUCTION:

1.1 ABSTRACT

One of the great strengths of the modern technology is how better today one make use of each pixel of screen to display useful and colorful contents. This strength can be further enhanced through the appropriate use of technology. We propose to develop and deploy a platform/module for Television. Such a platform/module should simplify the headache of using different screen for different purpose and put everything or most of the thing on Television screen which is now days just used for watching channels or at the most channels plus internet. This will promote more informed screen of television and will surely save cost by not investing in different display devices. Overall, it should provide the user with a better in house/office experience and should enhance out-of-class interactions.

We are proposing to develop and deploy a platform/module. This will consist of any portable Television with a large display upon which numerous contents can be displayed and the DSP processor TMS320DM6446 which will be the base of driving all the contents apart from regular channels videos. This will allow user to watch television and also keep watch on other things by connecting home security, chat window internet, mobile, Audio/Video player.

This platform/module should enhance the interactive TV experience. So we have provided various interfaces that are not available in the commercially available Television. As we are not targeting any particular class or set of television, all these features have been provided on a docking station which can be plugged into the device whenever required.

1.2 AIM

Our aim is to develop a platform/module for users, with great emphasis on requirements of multitasking environment on television screen. A good information environment is one that provides better way for user to get regular update without being roaming to places. Our plan is to leverage the major advances of the Television and produce a platform/module that not only is a sophisticated entertainment tool, but is also a major advance in the use of digital signal processor technology for aiding informative and multitasking interaction. We take advantage of wireless communication for communicating with Television or all the other stuff which will come into use.

1.3 PROBLEM STATEMENT

To develop an platform/module which

- Is essentially a portable device that will enhance Television to next level.
- Acts as a substitute to all display screens available for information display.
- Can act as an effective medium to exchange data and information wirelessly between different modules.
- Will have all the features of a fully equipped interactive television.

1.4 BRIEF HISTORY

Television has been essential part of information and entertainment. Since always man, has used various means to keep check on things and going on news around world. The ways have evolved from messenger in form of man, pigeons, post cards, letters etc and then invention of Television brought about a revolution. All the thoughts and ideas could be seen and learned. However every technology has its drawback and on this case was consumption of space and screen display for one thing i.e. channel or video watching. With the advent of computers, digital data moved ahead. With more features than a normal bulky television and power of internet made it very useful tool. This was a fast and extremely convenient way of getting data. However the main problem was that the PCs also were delivering same thing like television i.e. showcase of data one at a time. The problem of Television being a box was solved by introduction of digital services with the help of set up box. One can get hands on games on television and little use of internet on television. But it still could not solve the basic problem of using the screen, the big screen more efficiently. Also size was one of the major issues which limited its portability. Thus the LCD, PLASMA Television with interactive features with the help of set-top box was invented.

An **Interactive television** represents a continuum from low interactivity (TV on/off, volume, changing channels) to moderate interactivity (simple movies on demand without player controls) and high interactivity in which, for example, an audience member affects the program being watched. The most obvious example of this would be any kind of real-time voting on the screen, in which audience votes create decisions that are reflected in how the show continues. A return path to the program provider is not necessary to have an interactive program experience. Once a movie is downloaded for example, controls may all be local. The link was needed to download the program, but texts and software which can be executed locally at the set-top box or IRD (Integrated Receiver Decoder) may occur automatically, once the viewer enters the channel.

December 1977: The world's first commercial interactive TV service opens in Ohio. Qube offers 30 channels divided between broadcast TV, pay-per-view and interactive programming. Despite its popularity, it is not a commercial success.**1979:** Prestel is launched in the UK. Although designed to be used on TV sets using a special adaptor (with a modem), many users access the service via home computers. First commercially available technology to link the TV with the telephone.**1994:** Channel Four programme Games master takes messages from an internet chat room and puts them on to TV via Teletext subtitles. Videotron pilots interactive TV in the south-east. The analogue system allows viewers to choose content and make sports bets. Kellogg's Frosties broadcast an interactive ad.

October 1998: Sky Digital launches its 140-channel service via satellite. The handset gives access to TV guides on screen, though customers wait a year before the interactive shopping service Open is available. **March 1999:** NTL launches a trial interactive TV service. **July 1999:** Cable & Wireless signs 10,000 subscribers. The interactive services allow access to a range of websites. C&W is now part of NTL. Two-Way TV launches its games and quiz service. **August 1999:** Interactive football makes its debut on Sky Digital. Viewers of the Arsenal v Manchester United game can view highlights during the game, access statistics and select different camera angles.

Autumn 2000: Telewest's Active Digital shopping platform finally goes live after being launched nearly a year before. Services to be rolled out in 2001 include video on demand. NTL rolls out its internet TV service. **March 2001:** The Advertising Standards Authority rules that an ON digital ad "misleadingly exaggerated" its claim that its interactive TV service offered "full internet access". **April 2001:** Newham Council in east London announces plans to issue set top boxes to council tenants to report faults

With today's extended semiconductor life, better display resolution, memory, and wireless Internet access, the Television have become more useful and informative. Mobility issues have been addressed by developing televisions in the form of LCD and PLASMA and LED that can be carried around easily.

1.5 RECENT TRENDS AND DEVELOPMENTS

A number of manufacturers produce Television models that differ in size, processing speed, design, etc. to meet anyone's need. Television are often used where normal display are impractical or unwieldy, or do not provide the needed functionality. A popular use of the Television is to enable display and organization of data, and reading-listening of large amounts of digital information. In particular, casual user may see and download electronic versions of their video to watch on their PCs with the help of Set top box, or user may scan in the channels and just bring any entertainment or information they want to have. Commercially Television has been used for advertisement is shopping malls or to show case any music video in a Café.

The technological advancements over the last few years have been massive. While at one time wireless communication seemed a distant dream, now days it’s a part of our day to day lives. We use Wi-Fi and Bluetooth daily in our cell phones and laptops to transfer data. Television also come equipped with such things. Recently DTH (Direct to Home) wireless satellite service has changed the way we use Television screen and way we show get connected to information world. Power consumption has always been a major area of concern. Now a day’s chips, displays like LCD, PLASMA and LED have been developed which support low and extremely low power consumption.

The display technology has undergone a remarkable transformation over the past decade. Driven by the ongoing digital revolution standards, it has taken a giant leap forward. Organic LEDs, flexible displays and APR screens are an example of how much the technology has evolved in recent times. These have not only brought about a phenomenal change in power consumption but also reduced other overheads in flexibility, sizes and mobility. With the latest NXT sound Vu technology we can install our Television speakers behind the screen. With latest developments these technologies are available at affordable prices as well. We are certainly living in an era of electronic revolution.

From the Electromechanical Television to the Color Television or 3D Television, we're getting cooler toys every day. Not only do the latest platforms offer more computing power, better battery life, and greater usability than ever before, the variety of factors is increasing at an incredible pace. The recent introduction of the Google TV is a good example of net-generation mobility tools that continue to push the envelope on portability and power.

1.6 LITERATURE SURVEY

Consider the old-fashioned Television. We’re all familiar with it and need no instruction on its use. We go to room or a meeting and take use of the screen in it. We can check news; see carton, some games with game box and other stuff. Later we can go over our news channel and entertainment stuff, can record with electronic stuff available and then perhaps share them with others.

Television is easy to use and is universal. The only compatibility issue we have to worry about is video and audio quality. Everything can be seen using the one of the available services.

Televisions have their limitations, however. It acts like a box which is playing itself and the user get bored of switching channels after using if for a period of time. Sometimes we need to keep check on other stuff while watching TV. Like if wife is cooking something then she has to attend his favorite show then doing two stuffs becomes irritable. Like this other issues comes in form of only one things being show at a time. For news you have to switch on to different channels, for cartoon on third and for sports on fourth.

Despite amazing progress in computer technology and semiconductor field over the past decades, only now, with the introduction of the set top box, are we seeing the beginnings of a useful things we can do with Television screen. Like Tablet PC, a technology from Microsoft that has been licensed to a number of hardware vendors, has dealt well with a number of issues: it provides a great interface that, among other important features, has the look and feel of a real notebook. Television now days have gone into this touch screen mode. In particular, it lets one use a stylus to interact with television. Television includes wireless transceivers for access to the Internet or they do it with the help of set top box.

This project builds upon earlier work in which a simple Television was added with the feature of OSD (On Screen Display) on a platform and used it to display the channel logo and the message like the News channel do. User have already given thumbs up to this idea receive, view, and read notes, which were displayed while the video content is running. Modern day Television have many features integrated in them ranging from USB ports, touch screens, memory card readers and graphics cards etc.

Our plan is to move our design closer to the modern and next generation interactive Television. We intend to develop a platform/module for user and industry, with great emphasis on requirements of learning environment, information and other important features that have not been touched yet. Our main aim is to improve upon various display aspects like some widgets, video in form OSD form and ultimately produce a portable *premium* platform/module for user incorporating all the features necessary for them.

1.7 TECHNOLOGICAL SURVEY

Currently these things are working and available in the market:-

- Television with CRT display. This one is bulky and takes more space and power consumption is also more. Some CRT TV can handle 50 channels, some 100.

- Television with LCD, PLASMA, LED display. These are less bulky and are flat and consume very less space and power.

- Set Top Box: These boxes can be used with any kind of TV and can give high quality video and audio facility.

- Hybrids TV(or IPTV) , which share the features of the normal TV (can be of any one type said above) and Set Top Box or *Chipset* inbuilt to give more features like Internet access and simultaneously two channel view. But it has not received any further updates in recent times.

We intend on building a platform/module. That can be interfaced with any given Television. Building such platform will require to be updated with all the modern technologies and terminologies of modern consumer electronics and audio-video engineering. There are many things involved in building such platform. The first and most important is development of the communication module between the DSP which we will use and the given television with the help of our development environment.

First step in development of the platform/module was to select the digital signal processor to be used. We studied various processors and their architectures e.g. ARM, MIPS, Power PC etc. The next step after selecting (ARM) the microprocessor was building the PCB (if needed). For this thorough study of PCB designing was mandatory. So we studied PCB design rules and the PCB designing software Protel 2004. Designing of PCB involves preparing the schematics, making their netlists and then finally routing them to get the final PCB layout.

The latest digital signal processors that are available deliver very high processing power and performance and consume very less power was from Texas Instrument TMS320DM6446 EVM which comes with the ARM9 and C64x TI processor.

Apart from the board (TARGET) we had to decide upon other hardware such as communication module and other input devices.

The various DSP which are available from Texas Instrument were:

1) TMS320DM64x DSP with 1 C64x+; DaVinci Video.
2) TMS320DM646x SOC with 1 C64x+; 1 ARM9; DaVinci High Definition Video.
3) TMS320DM644x SOC with 1 C64x+; 1 ARM9; DaVinci Video.
4) TMS320DM643x DSP with 1 C64x+; DaVinci Video.
5) TMS320DM643x DSP with 1 C64x+; DaVinci Video.
6) TMS320DM37xSOC with 1 C64x+ARM Cortex-A8.

After comparing all the datasheets of the above available audio-video digital signal processors from the Texas Instruments and consulting our guide. We came to conclusion that we will go for TMS320DM644x SOC with 1 C64x+; 1 ARM9; DaVinci Video. So we will be going for the Evaluation Module of TMS320DM644x SOC with 1 C64x+; 1 ARM9; DaVinci Video.

Thus our TARGET board became TMS320DM644x SOC with 1 C64x+; 1 ARM9; DaVinci Video. The TMS320DM6446 (also refer to as the DM6446) make full use of the TI DaVinci ™ technology, to meet the network application of encoding and decoding media processing requirements, a new generation of embedded devices. The DM6446 can be the OEM and ODM products to market quickly with the strong support of the operating system, a richer user interface, high performance and long battery life, the greatest degree of flexibility through a fully integrated mixed processor solution program. The ARM Subsystem is designed to give the ARM926EJ-S (ARM9) master control of the device. In general, the ARM is responsible for configuration and control of the device; including the DSP Subsystem, the VPSS Subsystem, and a majority of the peripherals and external memories.

Components of DM6446:

- DM6446 Core Board

- A/V Expansion Board
- Video Camera
- 12V Power Adapter for Camera
- 5V Power Adapter for Core & Expansion Boards
- RS232 Cable
- Ethernet Cable
- A/V Cables
- IR Remote
- U-Boot Boot loader
- H.264 Encode / Decode Demo
- Hardware Drivers Source Code
- Flash Programmer Source Code
- DSP/BIOS for Linux
- Source code of Linux Kernel & Drivers with Target File System.
- Integrated Linux Development Environment VMware Image
- Compilers, Linkers, and Related build tools for TI DSPs.
- GNU Development Tools – ARM Compilers, Linkers and Related Build Tools
- Codec Engine & Framework Components
- Core & Expansion Board Schematics (Full version – DSN files)
- Core & Expansion Board PCB Layouts (Full Version – PCB files)
- BOM (Bills of Materials)
- Manufacturing instructions
- All Hardware Specification Documents
- Software Debug Guide
- User Reference Manual
- Hardware Quick Start Guide
- Hardware Setting Guide
- Software Getting Started Guide.
- Linux System Building Guide

We need a HOST. A HOST is a computer loaded with Windows and Linux environment. We installed the *Windows* and *Linux Fedora* on all the available laptops and Desktop with us (One laptop and one Desktop). This is required to interact with the TARGET (TMS320DM6446). Some application will be designed with the SDK provided with board on *Linux Fedora* and some with the help of Windows on Code Composer Studio.

Next we decided upon the software/application part of the design. There are many platforms available for interacting with the DSP. For the operating system we had many options viz.

- Microsoft based Operating systems
    - Windows XP
    - Windows Vista
    - Windows CE
- Linux based Operating systems
    - Fedora
    - Ubuntu
    - Suse

We chose a Linux based Operating System because the *Software Development Kit (SDK)* is *Red Hat (Fedora)* compatible. Being from Electronics background we had to start from scratch and so we studied various terminologies associated, the installation procedure and the general instructions. We are still discovering more and more things about the Linux Operating system. Even after going for *Linux* at some place we will surely need the help of Windows so we kept that too with our software stack.

We need a HOST. A HOST is a computer loaded with Windows and Linux environment. We installed the *Windows* and *Linux Fedora* on all the available laptops and Desktop with us (One laptop and one Desktop). This is required to interact with the TARGET (TMS320DM6446). Some application will be designed with the SDK

provided with board on *Linux Fedora* and some with the help of Windows on Code Composer Studio.

2.1 THEORETICAL DETAILS OF THE TOPIC:

The Interactive Television platform/module will enhance the interactive TV experience. User will use the Television not only to interact with various channels; it will also do and see other stuff simultaneously. The problem of having different display screen will be solved as we will be enhancing the display properties and using the OSD layer of the TV to drive various other things which we intend to run. With the various peripherals available on the TMS320DM644x SOC we can connect internet and can have chat while watching TV. We can integrate the home security system with the TV screen by displaying who’s at the gate on one corner of the TV on top layer with back layer showing switched channels. User can also chat back with the help of universal remote and can send chat messages or door opening instruction with the help of the platform which will use TMS320DM644x SOC properties. Person working in next room can give message on TV screen and the concerned person can take notice. Parents can watch their siblings playing in backyard with camera while they enjoying TV program.

2.2 APPLICATIONS

2.2.1 GMAIL

The Gmail application which we have developed is a very basic application, where a Gmail login form is displayed to you on the TV screen at the center with the live TV playing at the back, and the login form is translucent, so that the live TV at the back is also visible. Also the form can be placed anywhere on the screen with the help of drag and drop by a mouse pointer.

When a user enter the login information and click on the login button his inbox will be displayed.

2.2.2 BLUETOOTH:

To communicate between TV and other devices we developed a Bluetooth module. The Bluetooth module is basically a hardware that can be controlled by the board via AT commands. The interfacing of the Bluetooth module to the board is via UART port.

The Bluetooth module establishes a serial wireless link between two devices. Now the control commands from the device to the board (TV) and from the board (TV) to the device can be sent over this wireless link. Currently we have interfaced this module with a nokia mobile phone, and we can execute various AT commands of mobile via the board. So we can actually do anything with the mobile phone via the board.

This type of interfacing is the crux of our platform and which no television now in market is offering, also the idea of keeping the platform open source has a very high potential. As by keeping it open source infinite applications can be developed, which can be downloaded and no user will be left without an application developed for his specific purpose.

The benefits of the electronic Interactive Television are in:

- ***Efficiency***. Much of the overhead of just having a bulky television will be removed and user will see that most the space and investment done on TV is useful as the efficiency of any TV has been improved with this platform/module.

- ***Collecting materials***. User will be able to collect more material at one place only. That is at the TV screen. User can watch channels, chat on internet, display stream from security system, other stuff. Thus material or the information at his disposal is more than it used to be.

- ***Organizing materials***. User will be free to arrange things on the TV screen. User can display streaming video on left or right side. User can get message on top or any corner of the screen.

- ***Searching.*** An important aspect of organizing materials is being able to find things.

Though improvement in efficiency is important, real improvement will come only if user can and do take advantage of this efficiency to obtain a useful things that will be at his disposal. Our expectation is that by providing the tools that make it easy to explore additional material and user will actually do this and benefit from it. The Interactive Television would greatly simplify this task, allowing user to process more features.

3.1 BLOCK DIAGRAMS:

3.1.1 DEVICES AND LINKS:

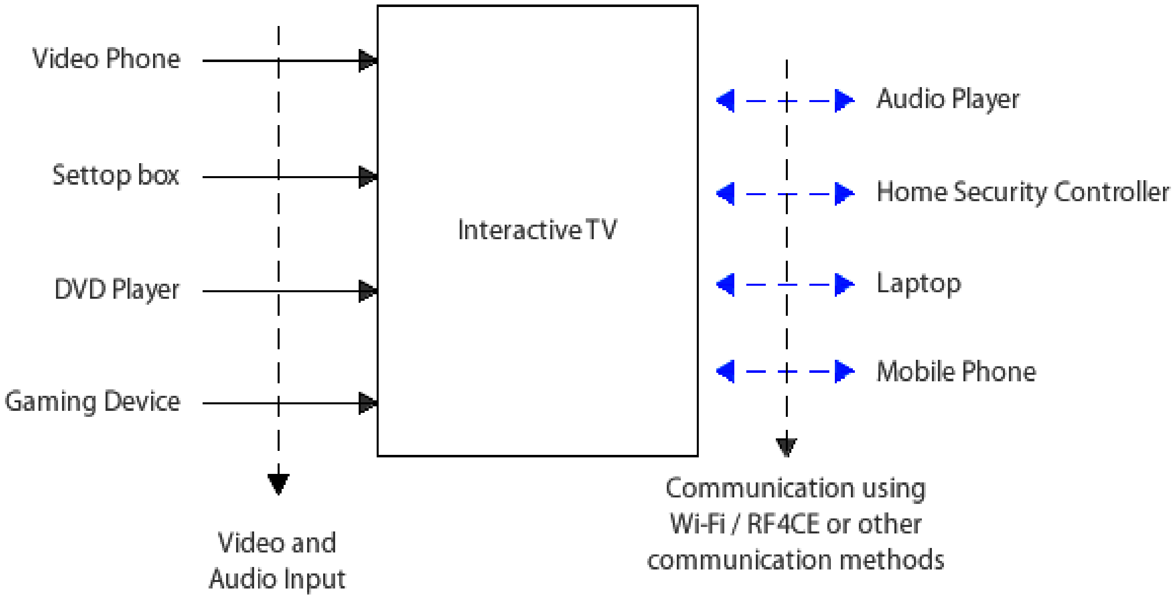


Fig1

3.1.2 FINAL SETUP:

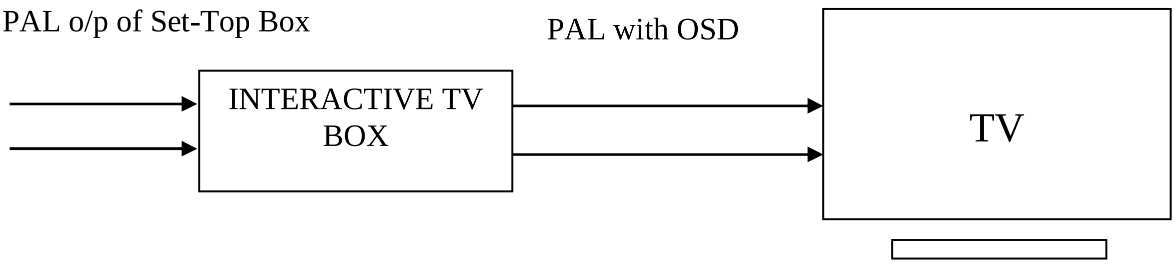


Fig2

3.1.3 BLOCK DIAGRAM

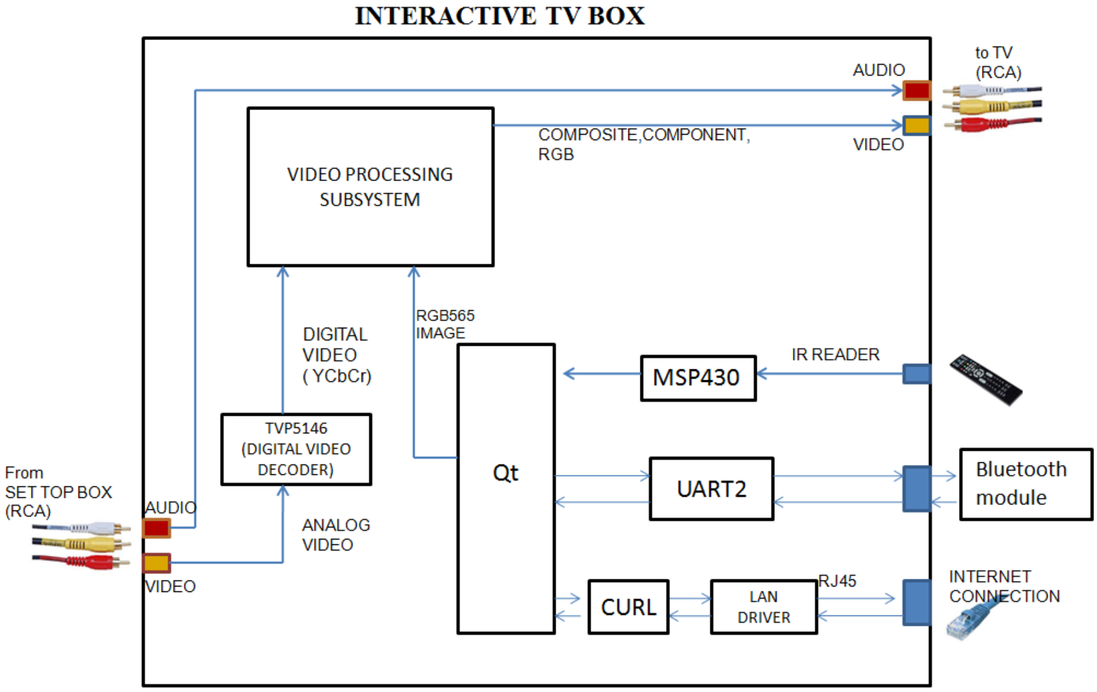


Fig3

3.1.4 PHOTOGRAPH OF BOARD

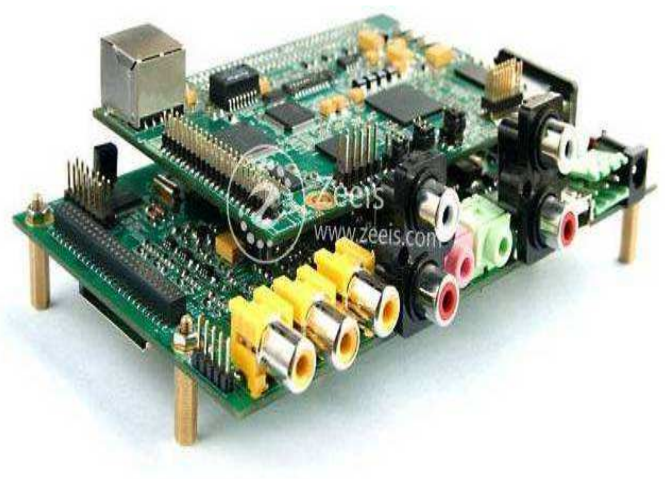


Fig4

3.1.5 BLOCK DIAGRAM OF BOARD

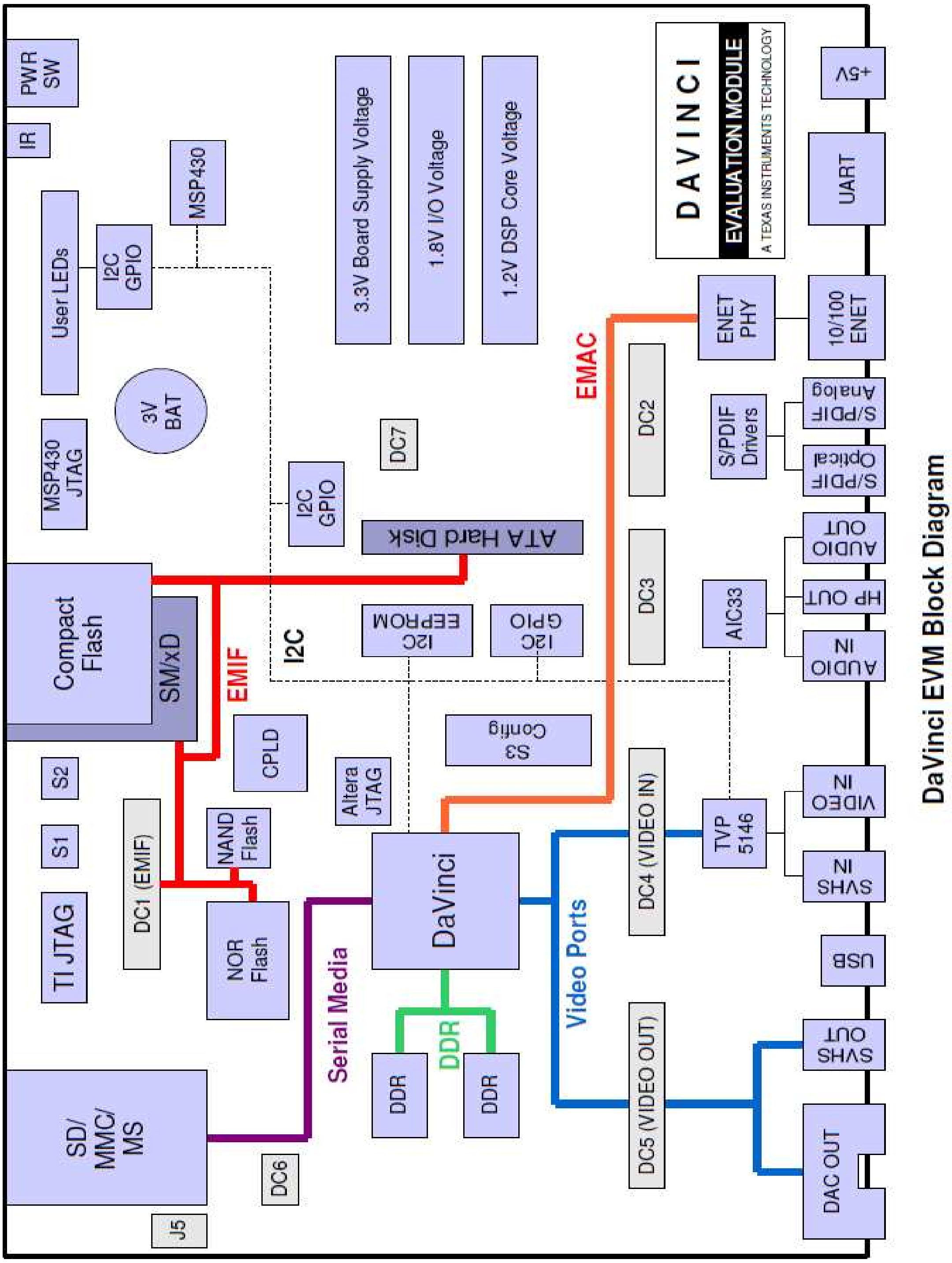


Fig5

3.2.1 SYSTEM DESIGN:

The design of the system will comprise of the target board and communication modules with other devices. The fig1 shows the audio video input devices like home security controller or output from set top box etc. The other side of the figure shows the communication links like RF4CE and wifi.

So, we have designed a system that will take input from the set top box, and also communicate wirelessly (via Bluetooth) to remote device. The audio video design will comprise of capturing the video, adding an OSD layer to it and then displaying it on the TV. For this we need the drivers of the decoder TVP5146, and a custom written program to do encoding and decoding.

The second point is adding the graphic user interface (GUI) for the control purpose. For this we worked upon Qt for a Gmail application. We used Qt creator for developing the application. The next thing was to add this GUI on the OSD layer with the live TV playing at back.

The next thing we developed was a Bluetooth module. It is nothing but a wireless serial link. So the control commands can be transferred to compatible devices.

The system after complete design will look like shown in figure 2. The board connected to the TV and output of set top box connected to the board.

For development purpose we need a host (computer) connected to the board. The target board comprises of an embedded linux operating system, the monta vista linux. So the terminal of the monta vista linux is seen through the serial port connection at 115200bps, 8 data bits, with no parity and no flow control. The file system of the board will be on the host which is mounted via Ethernet cable connected back to back between *target* (board) and the *host* (computer) using the *network file system (NFS).*

So the system in development phase consists of:

- The HOST *Laptop or Desktop* (containing file system and SDK installed).

- The TARGET *TMS320DM6446 & TV*.

THE HOST:

The HOST comprises of the following software modules:

- Linux Fedora
- Windows XP.
- Serial port
- DaVinci SDK on Fedora.
- Terminal Window (minicom).

THE TARGET:

The target comprises of following interfaces:

- TMS320DM6446 EVM.
- Cables and Peripherals.
- Remote.
- Camera o/p or set top box o/p.

Power Supply:

The power supply for TARGET *TV HOST Laptop or Desktop* will be normal AC input.

The TARGET TMS320DM6446 will require 12V Power Adapter for Camera 5V Power Adapter for Core & Expansion Boards.

TV Interface:

We have used RC cables to interface audio video of DM6446 with the TV.

Interfacing DM6446 to other modules:

To interface the DM6446 with other modules we will use following ports:

- External Memory Interfaces (EMIFs)
    - 32-Bit DDR2 SDRAM Memory Controller With 256M-Byte Address Space (1.8-V I/O)
    - Asynchronous16-Bit Wide EMIF (EMIFA) With 128M-Byte Address Reach
        - Flash Memory Interfaces
            - NOR (8-/16-Bit-Wide Data)
            - NAND (8-/16-Bit-Wide Data)

- Flash Card Interfaces
    - Multimedia Card (MMC)/Secure Digital (SD) with Secure Data I/O (SDIO)
    - Compact Flash Controller With True IDE Mode
    - Smart Media

- Three UARTs (One with RTS and CTS Flow Control)
- One Serial Port Interface (SPI) With Two Chip-Selects
- Master/Slave Inter-Integrated Circuit (I2C Bus™)

- 10/100 Mb/s Ethernet MAC (EMAC)
    - IEEE 802.3 Compliant
    - Media Independent Interface (MII)

- USB Port With Integrated 2.0 PHY
    - USB 2.0 High-/Full-Speed (480-Mbps) Client
    - USB 2.0 High-/Full-/Low-Speed Host (Mini-Host, Supporting One External Device)

Refer figure 5 for block diagram of the board.

3.2.2. DESIGN OF GMAIL

For creating the Qt GUI for Gmail application on OSD , we include the following inbuilt libraries of Qt:

- **i.** QtGui
- **ii.** QDesktopServices
- **iii.** QTextStream
- **iv.** QFile
- **v.** QString
- **vi.** QBrush
- **vii.** QWSServer

Compile the code for GUI using “Makefile” with cross-compiler arm_v5t_le-gcc and arm_v5t_le - g++.

3.2.3. ALGORITHM FOR GMAIL APPLICATION

1. Run the binary for Gmail application.
2. Enter the “Username” and “Password” in the GUI displayed on the OSD, the username and password is then passed to the cURL command.
3. When you press Sign In button, the cURL script will be executed, which is responsible to fetch the data from the Internet.
4. Username and Password are checked, if they are incorrect, QWidget with Log In Failed is displayed and go back to step 4.
5. For authenticated user, the unread mails are fetched from Gmail Inbox and displayed with QWidget on OSD.

3.2.4. FLOW CHART FOR GMAIL APPLICATION

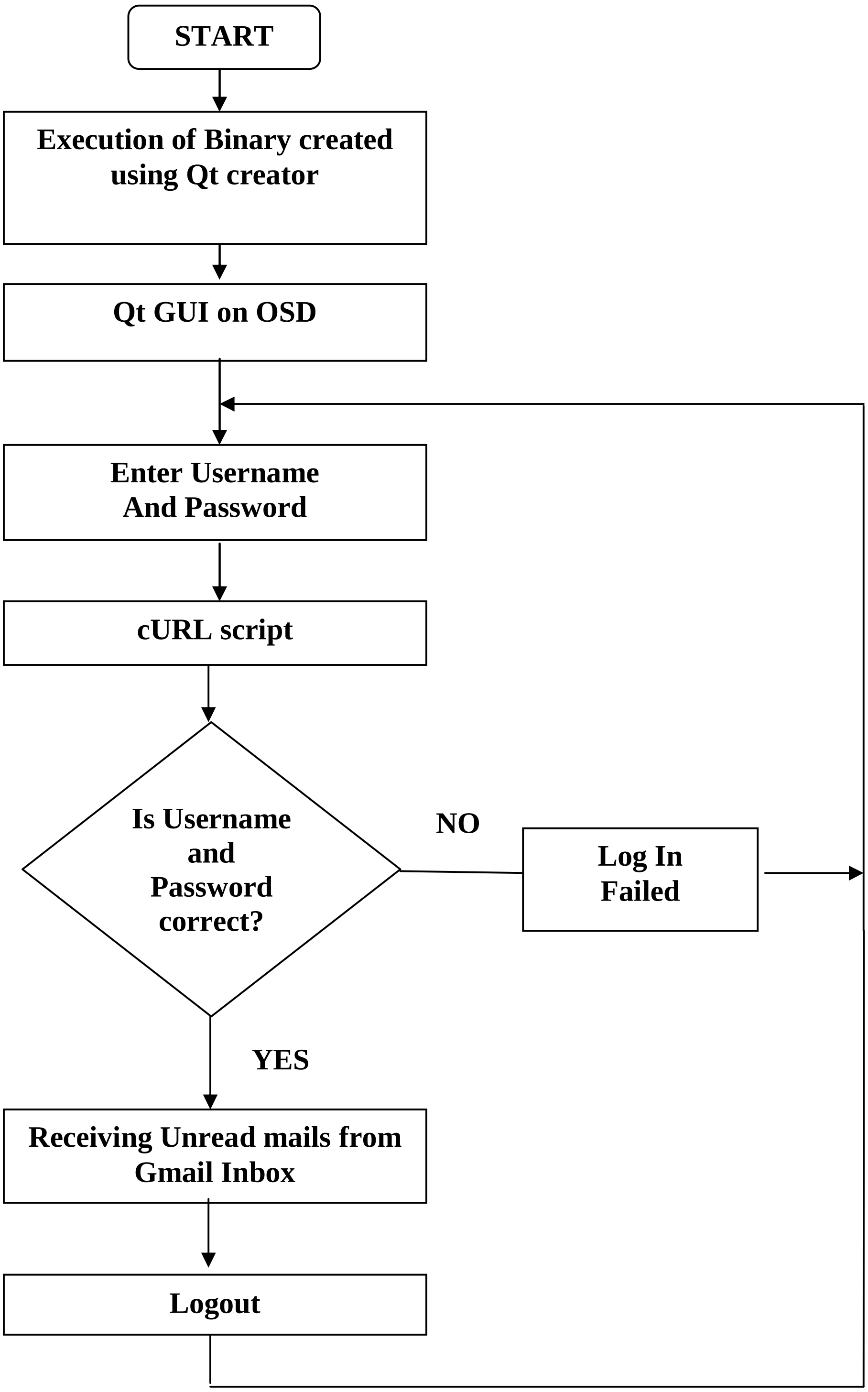

START
Execution of Binary created using Qt creator
Qt GUI on OSD
Enter Username And Password
cURL script
Is Username and Password correct?
NO
Log In Failed
YES
Receiving Unread mails from Gmail Inbox
Logout

Fig 6

3.2.5. BLUETOOTH MODULE

We have selected KC-21; class Bluetooth Data Module for the development of Bluetooth module which is used to establish a wireless serial link. The Bluetooth module works on power supply of 3.3v which is recommended value which we got from the data sheet.

The board has a JTAG port which has a power supply pin of 3.3V. So we need not design a power supply for the Bluetooth module. As we do not use JTAG, we make use of the available pin for power supply.

The KC-21 Bluetooth module works with the UART port. The UART is compatible with the 16450 industry standard. Four signals are provided with the UART interface. The TXD and RXD pins are used for data while the CTS and RTS pins are used for flow control. The UART pins operate at TTL voltage level and must be translated to higher RS-232 voltage levels for communicating with PC hosts. We use IC MAX 3232.

3.2.6 SCHEMATICS

3.2.6.1. LEVEL SHIFTER IC

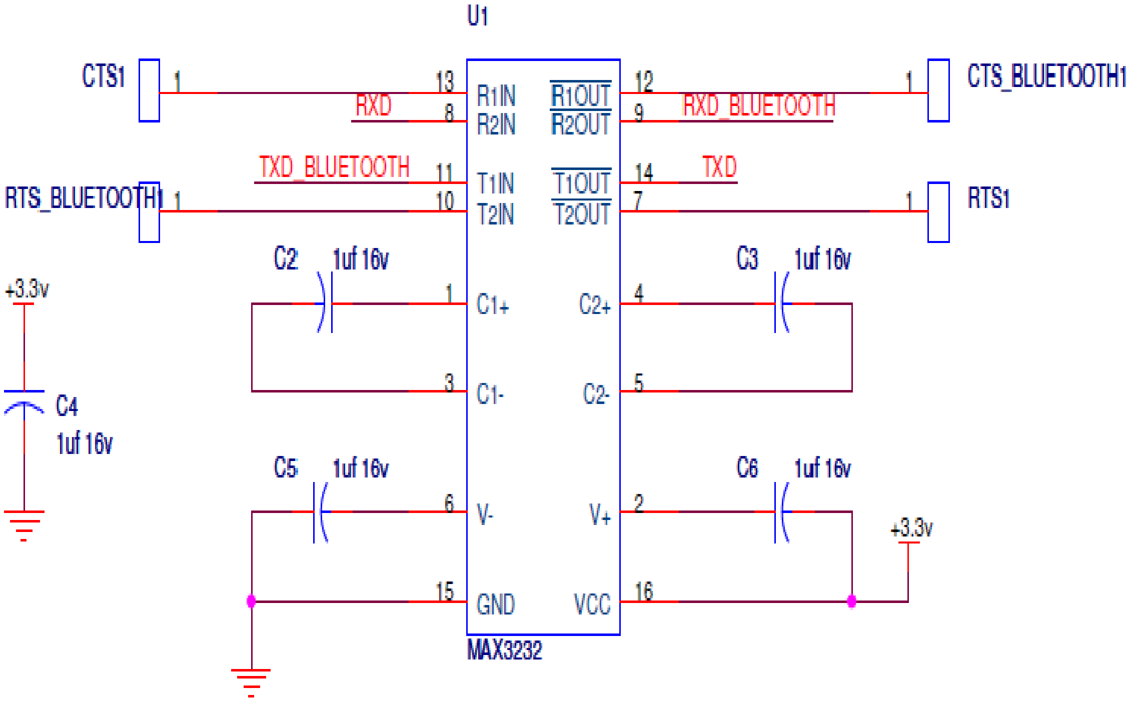


Fig 7

3.2.6.2 KC-21 BLUETOOTH MODULE

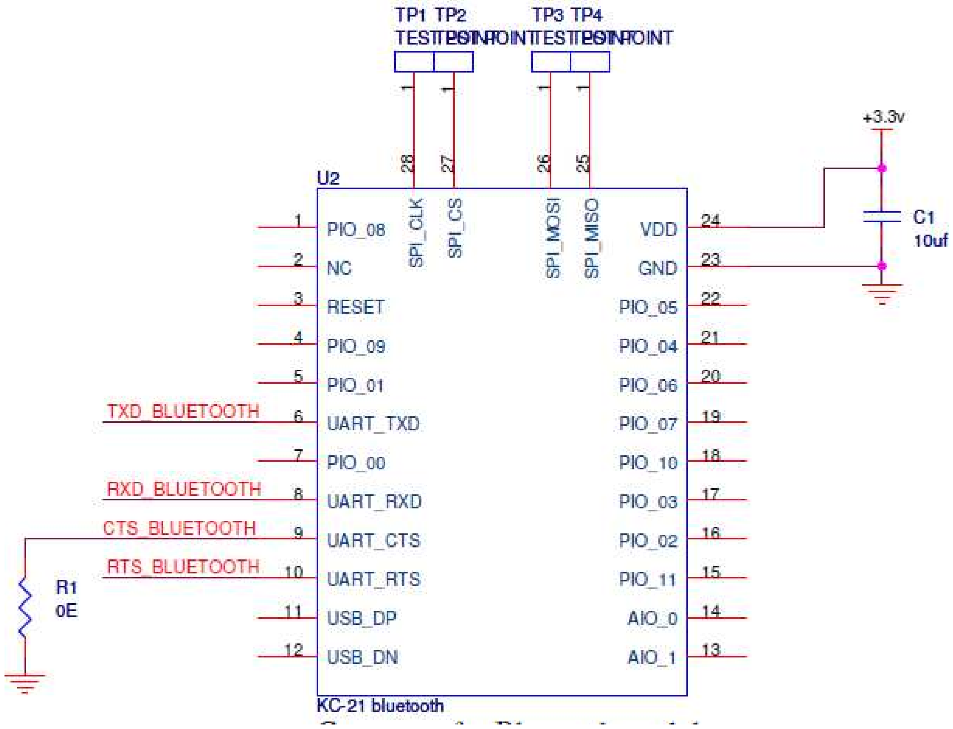


Fig 8

3.2.6.3. CONNECTORS OF BLUETOOTH MODULE

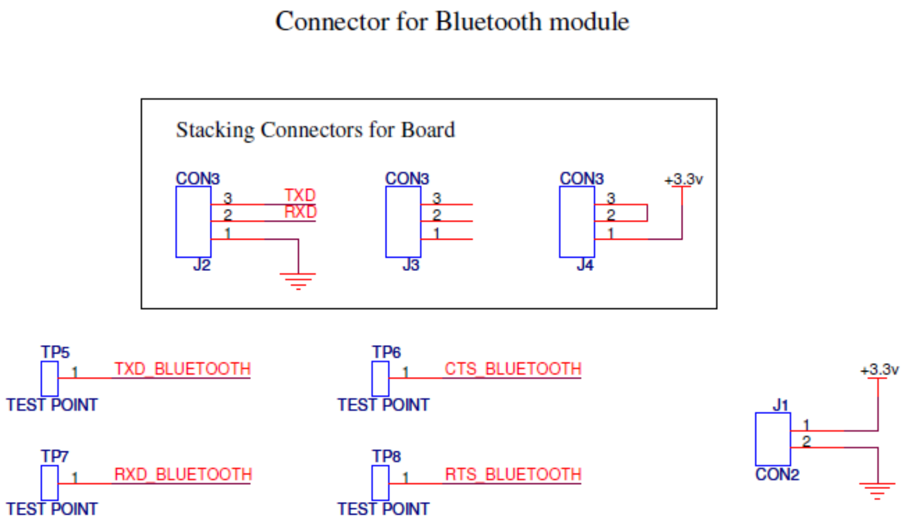


Fig 9

3.2.6.4. PCB LAYOUT OF BLUETOOTH MODULE:

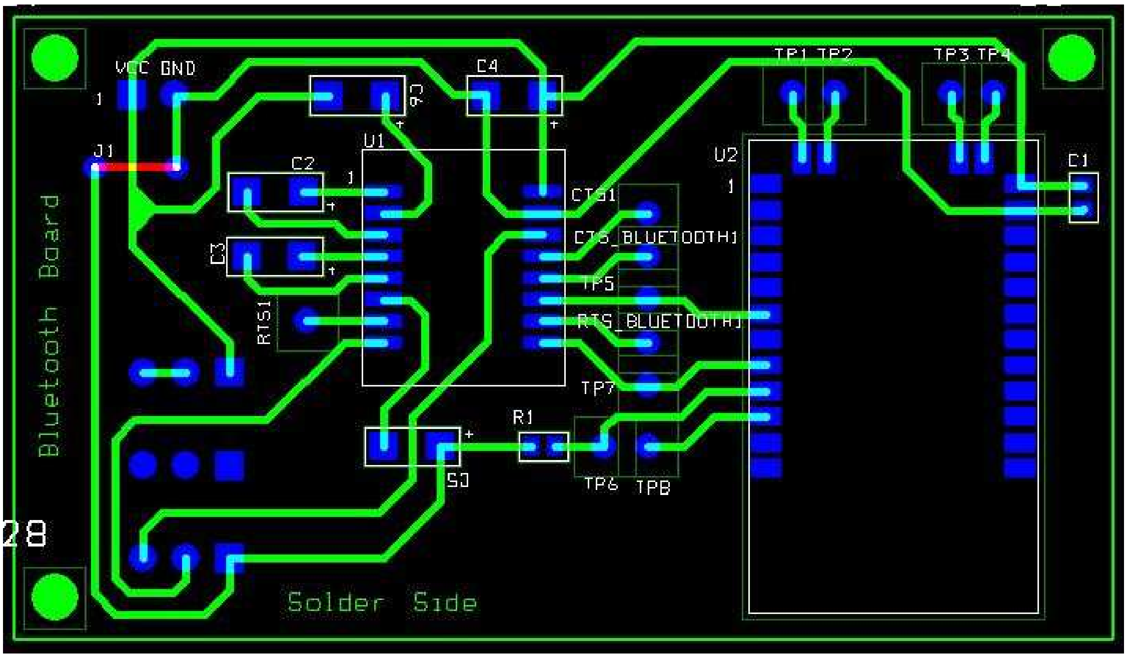


Fig 10

4. WORKING

- Basic operation of the device.
  - You press the "On" button on the Board DM6446 DMEVM.
  - You see the BIOS software performing the **power-on self-test** (POST). On many machines, the BIOS displays text describing such data as the amount of memory installed in your computer and the type of hard disk you have.
  - During this boot sequence, the U-BOOT will search for the booting device. It will boot from the port where it has been configured to..
  - The BIOS displays some details about your system. This typically includes information about the following:
    - Processor
    - Memory
    - BIOS revision and date
    - Display
  - Any special drivers are loaded from the memory and the BIOS displays the information.
  - Then the Board will be given necessary instructions from the HOST and TARGET will execute things accordingly on to the TV.

- Consider that we are taking the input from camera and displaying it on TV:

  Steps involved are:

  - The camera will take capture video and output it in NTSC/PAL format which is input to the board.
  - The TVP5146 decoder is responsible for capturing the video from analog input and to convert it into digital data and store it in the frame buffer.
  - The Video Processing Back End does all the necessary conversation and video quality enhancement stuff.
  - This part is called as the Encoding.

- Once the video is in the frame buffer there is processing on it if necessary. We do not process the video ,we just pass it to the video window, which is nothing but linux device /dev/fb/1
- This part is called as the Decoding.
- The graphic user interface (Gmail) is then added onto the OSD screen which is nothing but the linux device /dev/fb/0.
- Once data is written to the /dev/fb devices it is automatically displayed on the TV screen.
- This part is nothing but the Decoding.

4.1 GMAIL

The binary or the executable of the Gmail application is cross compiled using the the arm_v5t_le- cross compiler. Once the file is executed it displays the graphical user interface on the OSD screen. The environment variable of Qt is set according to tell the application which device or the display window should be used for displaying the graphical user interface. We set the environment variable by typing the following in the console *"QWS_DISPLAY=LinuxFb:/dev/fb/0"*.

As the Qt Gmail application is started it calls another thread of encode decode. The Encode-Decode thread will capture the input to the board and then display it onto the video window */dev/fb/1*. So the video or the live TV is playing at the back and Gmail graphic user interface at the front.

When user clicks on the sign in button using Remote as shown in the fig3 and after filling the login form displayed, Qt calls a curl script which runs at the back, it fetches the unread mails from the internet which is available to the board via either an Ethernet port or through a USB modem. Curl writes the information about the mails in a file; Qt in turn reads the file and displays the information on the OSD.

4.2. BLUETOOTH

Once the Bluetooth module is power on, it sends its hardware address on the UART port to which it is connected. The Bluetooth module is configured at 115200 bps, 8 data bits, no parity, 1 stop bit, no flow control.

After this we have to issue AT command to the Bluetooth module, to make use of the module. Firstly we discover various Bluetooth devices that are available. The command used is *AT+ZV Discovery*. In response we get the device name and hardware address of the various devices that are available.

Secondly we issue "*AT+ZV Bond <hardware address> 1234*" to establish a connection (pairing) with the remote Bluetooth device. When the bond is established we get a response *BondOk* from the KC-21 bluetooth module.

Thirdly when the bond is established, we issue "*AT+ZV DunConnect <hardware address>*" to establish a wireless serial link with that remote device. When the link is established the KC-21 Bluetooth module goes into bypass mode and we get a response "*<BYPASS MODE>*".

BYPASS MODE mode is nothing but a serial wireless link with the remote device. So what all data is sent on the UART port of one of the device is received on the UART port of the other device. So now it is as good as the board is connected to the mobile phone with a UART port. Now we can issue AT commands to the mobile phone. Issuing *"ATD<phone number>;"* will dial the phone number from the mobile phone. Like wise any AT command can be used to communicate with the mobile phone.

5. 1. IMPLEMENTATION:

Television is display devices so we can use the screen to display the stuff we want to. We need to make it interactive with the help of DM6446 EVM. It requires lots of interfacing and other peripheral and communication.

The first and most important step is study of the target board *processor AMR9 and C64x+*:

A *processor* is the central or primary semiconductor of device making up a complex electronic system. The basic purpose of the *processor* is to provide the electrical and logical connections by which the other components of the system communicate. *Processor* is the brain of any device. Other components such as external storage, controllers for video display and sound, and peripheral devices are typically attached to the *processor* via edge connectors and cables.

Now days the concept of SBCs or Single Board Computers has come up. The DM6446 EVM which we are using comes under this. A SBC can be defined as:

Single Board Computer is a printed circuit board that contains a complete computer, including processor, memory, I/ O and clock.

Studying of a SBC involves various key aspects. First of all we need to decide upon the microprocessor instruction set and its addresses to be used. The basic requirements of the processor are high processing power and low power consumption. Different processors that are available in our board , we can call it as Dual Core Processors involving ARM9 and C64x+ processors.

C64x+ Processor:

The TI C64x+ is powerful serious of processor used by many application as it has many features and is prefer while having Dual Core architecture. C64x+ processor brings x86 power and versatility to applications for entertainment, business, education, and embedded markets. It is one of the most advanced TI processor for demanding embedded applications. This level of capability may not be needed for every application, but when premium performance is needed, the C64x+ processor delivers. The C64x+ processor's integrated, innovative architecture is one of the most energy-efficient x86 solutions in the industry and can lead to longer battery life and enable small form-factor designs.

FEATURES:

- Advanced Very-Long-Instruction-Word (VLIW) TMS320C64x+™ DSP Core
  - Eight Highly Independent Functional Units
    - Six ALUs (32-/40-Bit), Each Supports Single 32-Bit, Dual 16-Bit, or Quad 8-Bit Arithmetic per Clock Cycle
    - Two Multipliers Support Four 16 x 16-Bit Multiplies (32-Bit Results) per Clock Cycle or Eight 8 x 8-Bit Multiplies (16-Bit Results) per Clock Cycle
  - Load-Store Architecture With Non-Aligned Support
  - 64 32-Bit General-Purpose Registers
  - Instruction Packing Reduces Code Size
  - All Instructions Conditional
  - Additional C64x+™ Enhancements
    - Protected Mode Operation
    - Exceptions Support for Error Detection and Program Redirection
    - Hardware Support for Modulo Loop Operation

- C64x+ Instruction Set Features
  - Byte-Addressable (8-/16-/32-/64-Bit Data)
  - 8-Bit Overflow Protection
  - Bit-Field Extract, Set, Clear
  - Normalization, Saturation, Bit-Counting
  - Compact 16-Bit Instructions
  - Additional Instructions to Support Complex Multiplies

- C64x+ L1/L2 Memory Architecture

- 32K-Byte L1P Program RAM/Cache (Direct Mapped)
- 80K-Byte L1D Data RAM/Cache (2-Way Set-Associative)
- 64K-Byte L2 Unified Mapped RAM/Cache (Flexible RAM/Cache Allocation)

ARM9 Processor:

FEATURES:

The ARM Subsystem includes the following features:

- ARM926EJ-S RISC processor
- ARMv5TEJ (32/16-bit) instruction set
- Little endian
- Co-Processor 15 (CP15)
- MMU
- 16KB Instruction cache
- 8KB Data cache
- Write Buffer
- 16KB Internal RAM (32-bit wide access)
- 16KB Internal ROM (ARM boot loader for non-EMIFA boot options)
- Embedded Trace Module and Embedded Trace Buffer (ETM/ETB)
- ARM Interrupt controller
- PLL Controller
- Power and Sleep Controller (PSC)
- System Module

- ARM926EJ-S Core

  - Support for 32-Bit and 16-Bit (Thumb® Mode) Instruction Sets
  - DSP Instruction Extensions and Single Cycle MAC
  - ARM® Jazelle® Technology
  - Embedded ICE-RT™ Logic for Real-Time Debug

- ARM9 Memory Architecture

  - 16K-Byte Instruction Cache
  - 8K-Byte Data Cache
  - 16K-Byte RAM
  - 8K-Byte ROM

- Embedded Trace Buffer™ (ETB11™) With 4KB Memory for ARM9 Debug**.**

I/O

- o 1xRS-232 (Serial port)
- o 1 parallel port
- o 4 USB 2.0

Display

- Television: 1 CRT or LCD.

Audio

- Audio Interface : Mic in, speakers

Ethernet

- 10/100 Mb/s Ethernet MAC (EMAC)
    - IEEE 802.3 Compliant
    - Media Independent Interface (MII)

The most striking feature of the DM6446 which makes strikingly suitable for our applications is the 4 window display architecture. This board provides us four windows. We have two video windows and two OSD windows. The video windows can be used to play the videos or the live television in our case. Refer figure below, we use video window 0 or the /dev/fb/1 to display the TV. The GUI which we developed using Qt is displayed on the OSD window 0 or /dev/fb/0.

When ever we write a rgb565 image to these devices, the image is displayed on the TV screen. The OSD window 1 can be set as an attribute window. So that writing a value which ranges from 0 to 7 to OSD window 1, will reflect the transparency of the OSD window 0 on the TV.

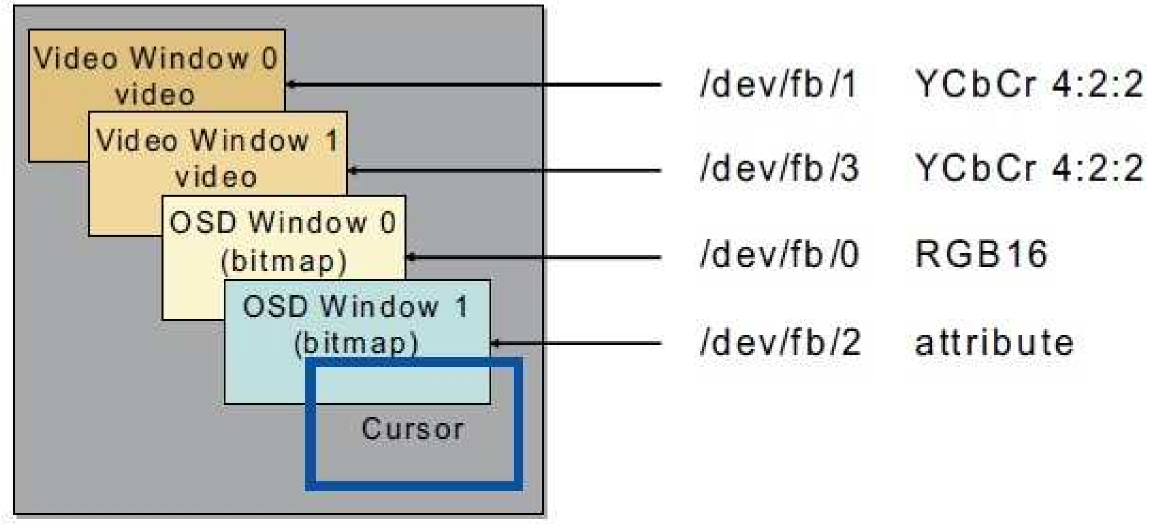


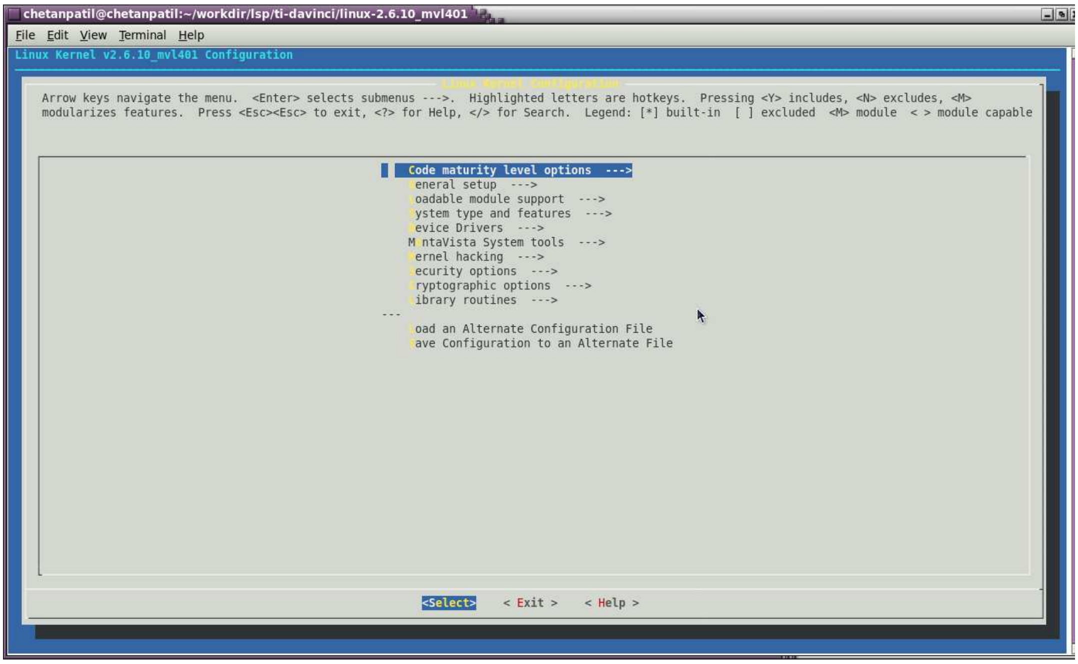
Fig 11(WINDOW ARCHITECTURE)

5.2. SELECTION OF THE *TV:*

In here we don't need to make any changes to the TV. We just need to have the TV with us and all the changes and programming will be loaded on to the DM6446 EVM and the system will be standalone. In terms if selection on the TV part we just need to make sure that it has the necessary ports so that we can connect it to the DM6446EVM and necessary output can be achieved. In the end we just need a TV whether its CRT based or LCD based.

5.3. BUILDING NEW LINUX KERNEL:

Initially when the board was shipped to us, the kernel in the flash was present and was not supporting for DaVinci encoder manager, uart2, and v4l2 drivers etc. Also we were using new SDKs and the kernel on the board was not able to sync with the SDKs and we were facing error of kernel panic while booting up the board. So after searching on mailing lists and forums, we came to know about the issue of kernel version and SDKs compatibility. So we downloaded the older SDK version1.3, then we were able to boot up the board.

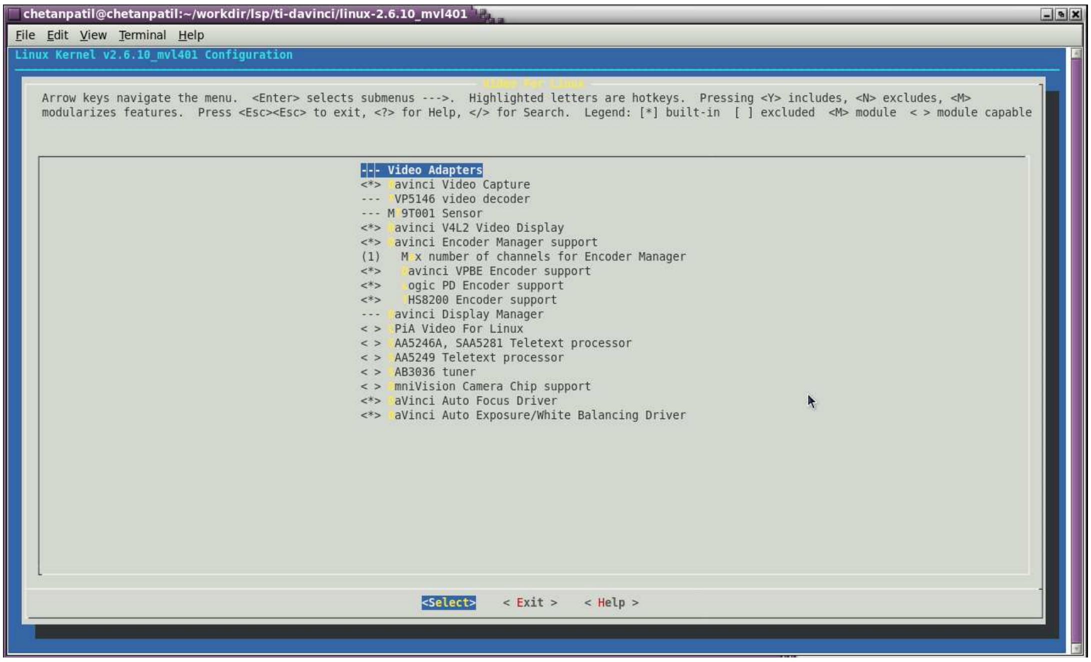


chetanpatil@chetanpatil:~/workdir/lsp/ti-davinci/linux-2.6.10_mvl401
File Edit View Terminal Help

```
Linux Kernel v2.6.10_mvl401 Configuration

                              Linux Kernel Configuration
 Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M>
 modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                         Code maturity level options  --->
                         General setup  --->
                         Loadable module support  --->
                         System type and features  --->
                         Device Drivers  --->
                         MontaVista System tools  --->
                         Kernel hacking  --->
                         Security options  --->
                         Cryptographic options  --->
                         Library routines  --->
                     ---
                         Load an Alternate Configuration File
                         Save Configuration to an Alternate File

                    <Select>    < Exit >    < Help >
```

Fig 12 (CONFIGURATION MENU FOR MONTA VISTA KERNEL 2.6.10)

As the encoder manager support was not present we were unable to run the demos. So after contacting with the board manufacturers, we decided to go for New Linux Kernel of Monta Vista. So after downloading the new kernel sources from Texas Instruments official website, we were able to compile the kernel with the DaVinci encoder manager support and v4l2 drivers in it. Now the video encode, decode and encode decode demos were running fine except the audio.



```
chetanpatil@chetanpatil:~/workdir/lsp/ti-davinci/linux-2.6.10_mvl401
File  Edit  View  Terminal  Help
Linux Kernel v2.6.10_mvl401 Configuration
                                   Video For Linux
 Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M>
 modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                                   --- Video Adapters
                                   <*> Davinci Video Capture
                                   --- TVP5146 video decoder
                                   --- MT9T001 Sensor
                                   <*> Davinci V4L2 Video Display
                                   <*> Davinci Encoder Manager support
                                   (1)   Max number of channels for Encoder Manager
                                   <*>   Davinci VPBE Encoder support
                                   <*>   Logic PD Encoder support
                                   <*>   THS8200 Encoder support
                                   --- Davinci Display Manager
                                   < > CPiA Video For Linux
                                   < > SAA5246A, SAA5281 Teletext processor
                                   < > SAA5249 Teletext processor
                                   < > SAB3036 tuner
                                   < > OmniVision Camera Chip support
                                   <*> DaVinci Auto Focus Driver
                                   <*> DaVinci Auto Exposure/White Balancing Driver

                              <Select>    < Exit >    < Help >
```

Fig 13 (LINUX KERNEL 2.6.10 SHOWING ENCODER MANAGER SUPPORT)

For the audio we were getting the I2C NACK error, after searching on this particular issue for a long time, we came to know that the hardware for sound the AIC33 is not being able to register while the board boots up. So after contacting Zeeis they suggested us to change the hardware register address of the AIC33 in a file in the kernel sources. Now the AIC33 was being registered and the sound devices were also working fine.

5.4. CROSS COMPILING cURL

For the internet application of Gmail. We need cURL script at the back to run. cURL is a tool to transfer data from or to a server , using one of the supported protocols such as http, https, imap, pop3, ftp, file, tftp, smtp etc.

The cURL command set is designed to work without any user interaction. It has support of libcurl for the transfer related features.

We downloaded the sources for cURL , then we had a look into the documentation of cURL , accordingly the files for cURL along with libcurl were cross compiled and ported to the board. Now cURL was ready to be used on the board.

5.5. CROSS COMPILING Qt

For the GUI development purpose, we needed Qt libraries to be cross compiled and ported to the board.Qt is a frame work by nokia. It is platform independent thing. There are documents for installing Qt on a wide range of platform architecture. We downloaded the sources for Qt, there was no documentation for compiling the sources for our particular arm and dm6446 architecture. Looking upon various mailing lists and the internet we managed to cross compile Qt and port all the libraries of it to the board. The example or demos were working. So now we developed a Gmail application on the laptop using Qt Creator.

5.6. USB PORT

Initially the kernel did not have the required driver for the USB port to work. The USB host controller was not initialized while booting up. So looking into the matter we came to know that we enabled the USB Host controller support in the kernel configuration. After compiling the new kernel we now were able to use mass storage device on the board.

USB mouse was also not working even after the USB Host Controller was up. So upon reading into the USB guide for normal linux on a desktop , we came to know that we need to enable the support for Human Interface Device (HID) in the kernel sources. After enabling the support for HIDs and compiling the kernel, the mouse on board was working fine!

5.7. USB MODEM

We need internet access on the board. So for that we are using a USB modem like the idea netsetter or tata photon to access the internet. For the USB modem to work we need various utilities like the modeswitch, linusb , pppd dialer, wvdialer. We are in the process of cross compiling all the utilities needed so that the USB modem is working, and the board does not have to depend on an Ethernet connection for internet access.

5.8. TESTING AND DEBUGGING

- For testing the UART we put it in loop back configuration i.e. Rx and Tx are shorted. We should receive what we send. The UART was tested in this way.
- The solutions for the problems which we faced while making the UART work were found by mailing the support team of the board i.e. Zeeis.
- For testing the Bluetooth module, first of all the PCB of the module was tested using a simple DMM, the continuity at various points was checked for any unwanted short circuits etc, before and after soldering the SMD ICs..
- After that the module was powered the module was checked directly for its functionality and it is working fine.
- For testing the Qt was properly ported to the board, we cross compiled various demo examples and they were working fine. Faced a lot of difficulties while cross compiling Qt, the solution was found on the mailing lists.
- cURL was compiled and ported to the board, then it was tested by running a simple script example of it.
- Gmail application was developed using Qt and tested on laptop first, it was working fine. Then it was ported to the board.
- USB mouse, i.e. HID device was made to work with linux by enabling the drivers for the same while configuring the kernel before compiling the uImage for the kernel.
- The mouse was tested by using the command “*cat /dev/input/mice*” in the console. Now when we move the mouse we see characters in the console.

6. CURRENT STATUS:

- The entire project can be divided into three broad phases.
  - Learning the various technologies involved
  - Developing the hardware (If needed)
  - Developing the software (Platform or the code or the algorithm)
  - Testing and debugging
- Last semester was dedicated to learning about all the technologies involved, deciding upon the various specifications and the hardware to be used.
- We have decided upon the basic TMS320DM6446 and placed order and interfaces that we plan to provide in our product.
- The basic SDK files have been downloaded and installed both on the Fedora and Windows XP.
- We have read about the U-Boot Loader, Image library, and Video Processing Unit, Instruction Sets.
- The order for the material needed was placed,so that it gives us enough gestation time till the time the products are delivered.
- Qt the software platform for graphic user interface has been configured and compiled for the board. Now the libraries of Qt are on the file system of the board.
- cURL is cross compiled and ported to board.
- The development environment is properly set and all the demos are working fine.
- Issue of the I2C error for AIC33 sound device has been resolved by making changes in the kernel for the hardware register address of the AIC33 device. Now the sound device on the board is working fine.
- All the audio video and speech demos working fine.
- Development of the Gmail application has been done, using Qt and cURL and it is executable on the board.

- The circuit diagram, PCB designing and schematics of the Bluetooth module has been done. The hardware is a SMD type. The soldering of various components is done and the module is working.
- The second UART is up and working on the board required for communication with Bluetooth module.
- A voice call can be initiated from a mobile through the Bluetooth module on board.
- The USB port on the board is up and running. A USB mass storage device can be plugged into and can be mounted to access the files in it.
- Now SD can be used like the USB mass storage device, with the new kernel sources.
- Interfacing of mouse with the Monta Vista Linux running on board is done and interfacing the mouse with Qt is remaining.

7. RESULTS:

1. Most expected part was to have an overlay of video with the application in OSD. Currently the overlaying of the gmail application with video at back is done.
2. In order to the Bluetooth communication, the UART2 is up.
3. Also we are able to control the devices (Mobile Phones) using Bluetooth wirelessly via AT commands.
4. USB port on the board is up and running, USB mass storage device can be mounted to file access.
5. Interfacing of USB mouse with the board is done.

CONCLUSION:

Interactive TV system demonstrates the interactive experience by integrating various device displays with the help of TV screen and controls them. So we provided various interfaces with TV which are generally not available now days. Overall, it is providing us better in house/office experience of entertainment/work along with the other tasks. We believe that improvements in our system will further bring more enhanced and optimized applications and leverage the current interactive experience to higher levels.

FUTURE SCOPE

- Scheduling of programs on TV can be done with the help of Image Processing by detection of channel Logo.
- With the integration of Wireless modules (Wi-Fi, ZigBee etc) the platform will communicate and control the desired devices.
- The Industry level interfacing is also possible

  e.g. The interfacing with the Temperature Sensor will provide the temperature related information on TV screen to the supervisor in a factory.

- The domestic level interfacing is also possible
  e.g. The Air Conditioning system, Video Door Phone etc. in house can also be controlled using the TV screen.

- In future, all devices will use only one screen i.e. Television for displaying the information.

8.CODE OF Qt GMAIL APPLICATION

window.h

```
#ifndef WINDOW_H
#define WINDOW_H
#include <QWidget>
#include <QWSServer>

    class QPushButton;
    class QLabel;
    class QLineEdit;
    class QWSServer;

class Window : public QWidget
{
    Q_OBJECT

public:
    Window(QWidget *parent = 0);



public slots:
     void accept();
     void close();

private:

    QWidget *newwindow;

    QString line;

    QPushButton *button1;
    QString name1;
    QString name2;
    QString name3;
    QPushButton *button3;

    QLineEdit *text1;
    QLineEdit *text2;
};

#endif // WINDOW_H
```

window.cpp

```
#include "window.h"
#include <QtGui>
#include <QDesktopServices>
#include <QTextStream>
#include <QFile>
#include <QString>


Window::Window(QWidget *parent) :
    QWidget(parent)
{

QPushButton *button1 = new QPushButton(tr("Sign In"));
QObject::connect(button1,SIGNAL(clicked()),this, SLOT(accept()));
button1->show();

QLabel *label1 = new QLabel(tr("Username"));
label1->show();



QLabel *label2 = new QLabel(tr("Password:"));
label2->show();

text1 = new QLineEdit;
text1->clear();
text1->show();

text2 = new QLineEdit;
text2->setEchoMode(QLineEdit::Password);
text2->clear();
text2->show();


QVBoxLayout *buttonLayout1 = new QVBoxLayout;
buttonLayout1->addWidget(button1);

QGridLayout *mainLayout = new QGridLayout;

mainLayout->addWidget(label1, 0, 0);
mainLayout->addWidget(label2, 1, 0);
mainLayout->addWidget(text1, 0, 1);
mainLayout->addWidget(text2, 1, 1);
mainLayout->addLayout(buttonLayout1, 2, 1);

setLayout(mainLayout);
setWindowTitle(tr("Gmail"));

QWidget::resize(250,150);
QWidget::move(250,220);
}

void Window::accept()
```

```
{
    QString name1 = text1->text();
    QString name2 = text2->text();

    if(1)
     {


        QFile file1 ("test.sh");

          if (file1.open (QIODevice::WriteOnly | QIODevice::Text) == true) {
            QTextStream out (&file1);
            out << "#!/bin/bash\ncurl -u "<<name1<<":"<<name2<<" --silent
\"https://mail.google.com/mail/feed/atom\" | perl -ne 'print \"\t\" if /<name>/; print \"$2\n\" if
/<(title|name)>(.*)<\\/\\1>/;'";
            file1.close();
          }

             QLabel *Inbox = new QLabel;
          this->hide();

        QWidget::move(400,300);
```

main.cpp

```
#include <QtGui>
#include <QBrush>
#include <QWSServer>
#include "window.h"
#include <QTextStream>
#include <QFile>
#include <QString>

int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    QWSServer::setBackground(QBrush(Qt::black));

    QWSServer *abc = new QWSServer;
    abc->closeMouse();
    abc->openMouse();


    Window wiNdow;

    wiNdow.show();

    return app.exec();
}
```

9.REFERENCES:

- TMS320DM6446:
  - http://focus.ti.com/docs/prod/folders/print/tms320dm6446.html
  - http://focus.ti.com/lit/ds/symlink/tms320dm6446.pdf
  - http://dm6446.org/
  - http://www.zeeis.com/products/
- Referred papers:
  - http://groups.csail.mit.edu/tds/papers/Shvartsman/SPIE95.pdf
  - Integrating distributed multimedia systems and interactive television networks by Alex Allister Shvartsman
  - http://www.busim.ee.boun.edu.tr/~sankur/SankurFolder/EUSIPCO2009_Automatic_TV_Logo.pdf
  - Automatic TV logo Detection and classification in broadcast ideas by Nedret OZAY, B ulent SANKUR
- DaVinci Linux Drivers:
  - http://focus.ti.com/lit/an/sprs566a/sprs566a.pdf
- Television:
  - http://widgets.yahoo.com/
  - http://en.wikipedia.org/wiki/Interactive_television
  - http://www.mhp.org/docs/itv-design_v1.pdf
  - http://www.google.com/tv/
- Linux:
  - http://www.mvista.com/
  - http://fedoraproject.org/

- Books referred
    - “Printed Circuit Boards” by R S Khandpur
    - “Linux Device Drivers” by Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman.