

Flashback, A television video recording and commercial viewing reduction device

Marlon De La Cruz, Rene Martinez, Trenton Reed, and Marlon Smith

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

Abstract — This paper presents the design methodology and underlying concepts used to achieve a real - time commercial detection used in the Flashback device by realizing 3 stages. Stage one is Black Screen Detection, stage two is Cut Rate and stage three is Logo Placement Detection. This paper also discusses all the hardware components and integrated circuits used and how they were laid out in our Printed Circuit Board PCB design strategy. It is also discussed how and why major design decisions were made to create a complete system.

Index Terms — Printed Circuit Board (PCB), Digital Signal Processor (DSP), High Definition Media Interface (HDMI), Ethernet Physical Transceiver Chip (PHY), Cut Rate Detection, Black Screen and Silence Detection, Logo Placement Detection, Computer on Module (COM)

I. INTRODUCTION

Flashback (patent pending) is a solution to the television viewer's problem of missing the action in their favorite programs. Flashback is a feature designed to detect the end of a commercial break in real-time. The user will be able to enable our feature on a primary television signal of their choosing. After enabling Flashback the user may freely roam to other secondary television signals without the worry of missing primary programming. While the feature is enabled Flashback will analyze the incoming primary television signal to detect the end of the commercial break on the primary program. When it is determined that the commercial break has ended, the user will be prompted and asked if they would like to remain on the secondary program or if they would like to resume the primary program. If the user wishes to remain on the secondary program, then Flashback will begin recording content of the primary programming, so the user may "Flashback" to the moment in time when the primary channel's commercial break ended. In addition to the recording

feature, Flashback will remain active until the user specifies another primary television program or disables Flashback on the original primary program.

In order to accomplish Flashback, we developed a device that will be a simple additional connection to the user's existing television. An over-the-air programming antenna was used to provide Flashback with television programming. To analyze the television signal and detect commercial breaks, we will use digital signal processors and high-level software. To control the Flashback device, our group will use an infrared remote control that will have a user friendly Liquid Crystal Display (LCD) to confirm user commands.

The goal of Flashback is to enhance the user viewing experience while creating a device that is portable, low-powered, easy-to-use, real-time, and most important of all, accurate. We also plan to incorporate the elimination of commercials within the recording of these programs. This feature will provide users with uninterrupted, recorded playback. This aspect of recording will also save space for recorded programs due to the elimination of the commercial blocks that every television program contains.

II. Hardware Requirements

The initial hardware requirements for the Flashback device are as follows. The device needed to have more than 512 MB of RAM to easily run an operating system, a desktop environment and the Flashback software. The device must also have at least 16 GB of available storage to accommodate an operating system, its kernel, the Flashback software and other system tools. The device's processing clock frequency must be at least 1 GHz to be able to stream video from its dual television tuners while recording and playing back at least one of the streams. Lastly the dual television tuners that the Flashback device uses must be able to operate within the frequency range of 42 – 866 MHz; this frequency range is the range of frequencies available for broadcast television in North America. These major hardware requirements were set in mind of the software requirements.

III. Software Requirements

The main functionality of Flashback and its patent claims reside within the software, which will be expanded upon later in this work. The major software requirements of the Flashback device are as follows. A responsive, non-invasive user interface and efficient background threads are the main ideas behind the Flashback device's software and functions. To adhere to these ideas Flashback functions must be able to seamlessly sync audio and video while processing at least 30 video frames per second. To achieve this, the Flashback device's background threads

must uphold: a commercial detection time less than 0.5 seconds, a channel tuning of less than three seconds, and a live television lag no greater than five seconds.

III. Hardware Components

Flashback implements the expansion connectors for a Gumstix Overo Storm Computer on Module board, HDMI, two USB-A ports, Ethernet, USB-B mini (serial console over USB for the remainder of this document), IR sensor, standard definition video, audio for left and right speakers, segment display, power distribution, fan for temperature control, and the ICs that make those ports possible.

A. Gumstix Overo TidalStorm Computer on Module:

The Gumstix Overo TidalStorm Computer on Module is a PCB that contains key components of Flashback’s hardware. Namely, the Overo TidalStorm, as pictured below in **Figure 1**, features the following:

- ARM Cortex-A8 Architecture
- C64x Fixed Point DSP
- Texas Instruments TPS65950 Power Management Integrated Circuit (IC)
- DM 3730 Application Processor
- Up to 1 GHz Processor Clock Rate
- 1 Gb of RAM
- MicroSD Card Slot

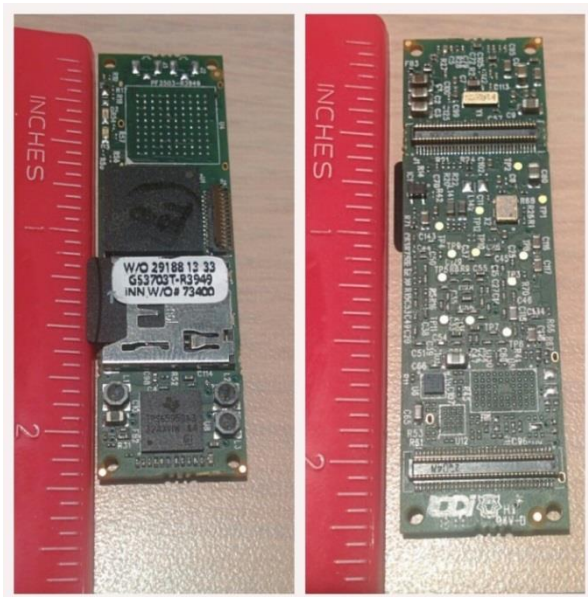


Figure 1: Gumstix Overo TidalStorm COM

The Overo TidalStorm COM was used to remove some of the complexity in the PCB design and layout process for the Flashback device. The Overo TidalStorm is meant to

aid in expediting the prototyping phases of various projects both commercial and personal.

The OveroTidalStorm COM is built to interface with a number of different expansion/base boards. These boards can then be customized to extend the built in I/O capabilities and features of an Overo TidalStorm COM, such as WiFi, DVI-D, Ethernet, USB and Audio. As shown in **Figure 2**, an expansion board is built with a few I/O interfaces that were used for rapid prototyping, testing



Figure 2: Gumstix Summit Expansion Board with OveroTidalStorm COM

and the final product, Flashback Rev.A. As noted in **Figure 2**, this is the Summit expansion board, who’s reference design aided us in designing Flashback’s hardware. It was decided to expand upon the I/O interfaces pictured in **Figure 2** which will later be explained in more detail. As a result of using the Summit expansion board reference design, our final PCB design, which is shown below in **Figure 3**, is very similar to that of the aforementioned expansion board.

B. Mains to DC Signals:

In order to determine the specifications of the power supply, we first determined the requirements of the components being implemented. With voltage rails for: 1) 5 volts needed for the HDMI, USBs, and LCD; 2) 3.3 volts needed for ICs relative to parts, such as the temperature sensor, composite video, and fan; 3) 1.8 volts for the Ethernet port and ICs. The Flashback device’s main power dissipation lies with the two USB-A ports which require 5 V at 0.5 A each, 2.5 W. Therefore, the power supply at the bare minimum must be 5 V and greater than 1 A. The power chips selected, the TPS62111 (3.3V) and TPS62112 (5V), will accept the input power supply and provide a 5V rail and a 3.3V power plane. The 3.3V power plane will be

used as an input to MIC5247 3.3V to 1.8V for the last voltage rail.

C. Television Tuners:

Flashback makes use of two Hauppauge WinTV-HVR-950Q USB television tuners. These devices accept a coaxial input which sends the television signal through an Xceive XC5000 tuner, an Auvitek AU8522 A/V Digital Demodulator, an Auvitek AU0828 USB Bridge, and output via USB 2.0. The tuners are able to output in high-quality MPEG-2 files for streaming and recording a digital video channel due to the fact that television broadcasters are no longer using analog transmission. Broadcasters in North America are using the Advanced Television System Committee's (ATSC) standard. This standard uses an eight-level vestigial sideband (8VSB) for over-the-air (OTA) television broadcasting [1].

D. Ethernet

For the spectrum of this project, Ethernet was not essential for completion, but it did aid in debugging

purposes, later mentioned in this paper. Flashback is envisioned to have constant updating, and since file transfers over serial consoles over USB are inadequate and inefficient, a 10/100 Ethernet port was recommended and applied. The Ethernet Physical Transceiver chip (PHY) provides an analog signal physical access to the link. It coordinates with independent interface that should take care of higher layer functions.

Ethernet would allow internet access, which in turn would also allow updates to Flashback's operating system and kernel, along with various other system tools.

E. Event Controller

The audience can control Flashback in a variety of ways, ranging from an infrared remote control, using RC-5 protocol, to an Android app on their smartphone, confirmed by the 7 segment LCD. This is still a possibility, since an IR receiver circuit was implemented, but since the focus and priority of the project was concept and patent claim, Flashback will be enabled or disabled by user interaction with the GUI, using either the keyboard or mouse. Future revisions of Flashback will have more

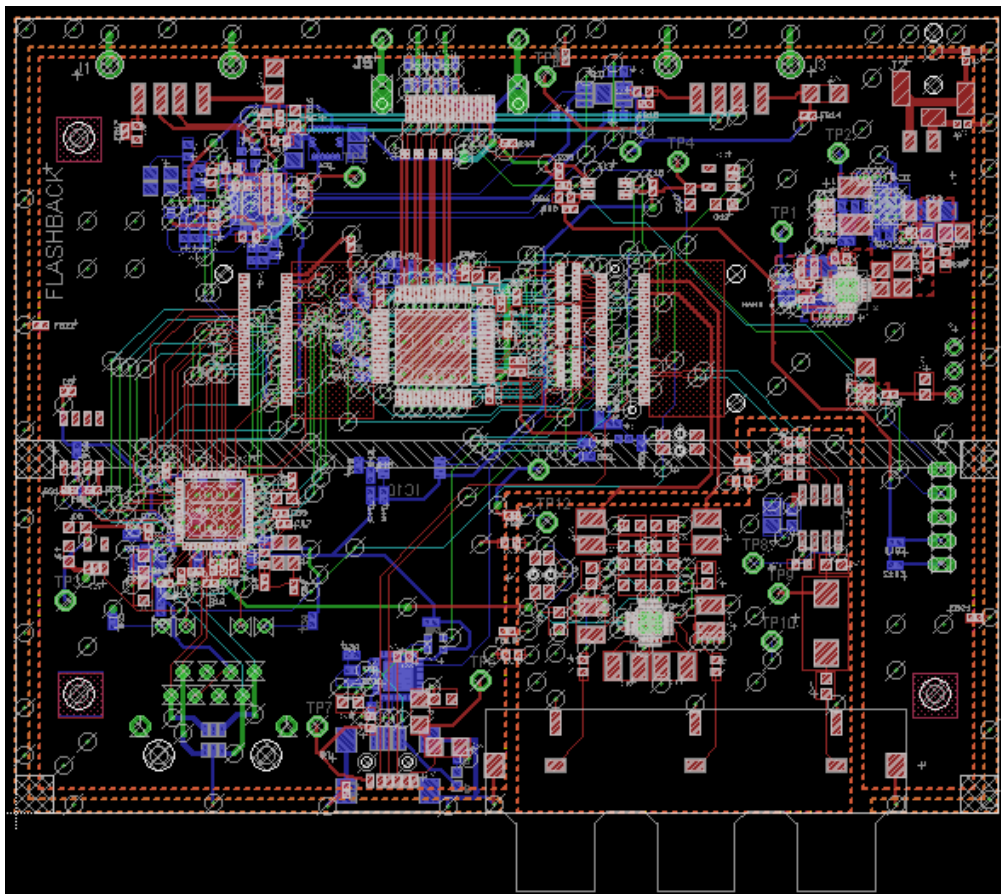


Figure 3: Final Board layout in Eagle CAD

desirable and universalized user control.

F. Standard Definition

Standard definition video is implemented in the device, using the chrominance and luminance signals coming from the Overo Tidal Storm Computer on Module board. These signals pass through a standard video filter which behaves similar to a 5th order Butterworth Low Pass Filter with cutoff frequency at 8.5 MHz. The typical bandwidth associated with standard definition video is approximately 6 MHz. The cutoff frequency falls within Flashback’s specification that the video signal will be able to pass through without being clipped by the video filter. In addition, the video filter provides a 14.3 dB gain to counteract any signal strength loss after passing through the tv tuners, USB ports, USB hub, Overo connectors, USB PHY on the Tidal Storm, DM3730, then back over the Overo connectors before it reaches the video filter. To combine the chrominance and luminance signals after the video filter, Flashback uses a capacitor to essentially shift the bandwidth of the chrominance signal onto the luminance signal. The combination of chrominance and luminance provides a single composite video signal trace.

Flashback will also provide audio outputs for a left and right speaker. Flashback utilizes a Direct Path Amplifier.

The amplifier focuses on providing a high signal to noise ratio of 105 dB. A nice feature of the amplifier our group chose is that it has pop reduction circuitry so the user won’t hear clicks coming from our speakers.

G. High Definition Media Interface (HDMI)

Flashback will be implementing the feature of high definition television through the standard High Definition Multimedia Interface. The device used to control Flashbacks HDMI uses an I2C interface. In addition, our device has enhanced PLL noise immunity, no HSYNC jitter anomaly, negligible data-dependent jitter, and displays using 24-bit true color pixel format.

HDMI requires careful routing to avoid signal integrity losses. Texas Instruments provided guidelines for routing HDMI signals without compromise to the data lines. Some key points taken from TI’s article was to route all data lines output to connectors on the top layer with a solid ground plane to establish controlled impedance and route the slower speed control signals on the bottom layer since they can tolerate the use of vias.

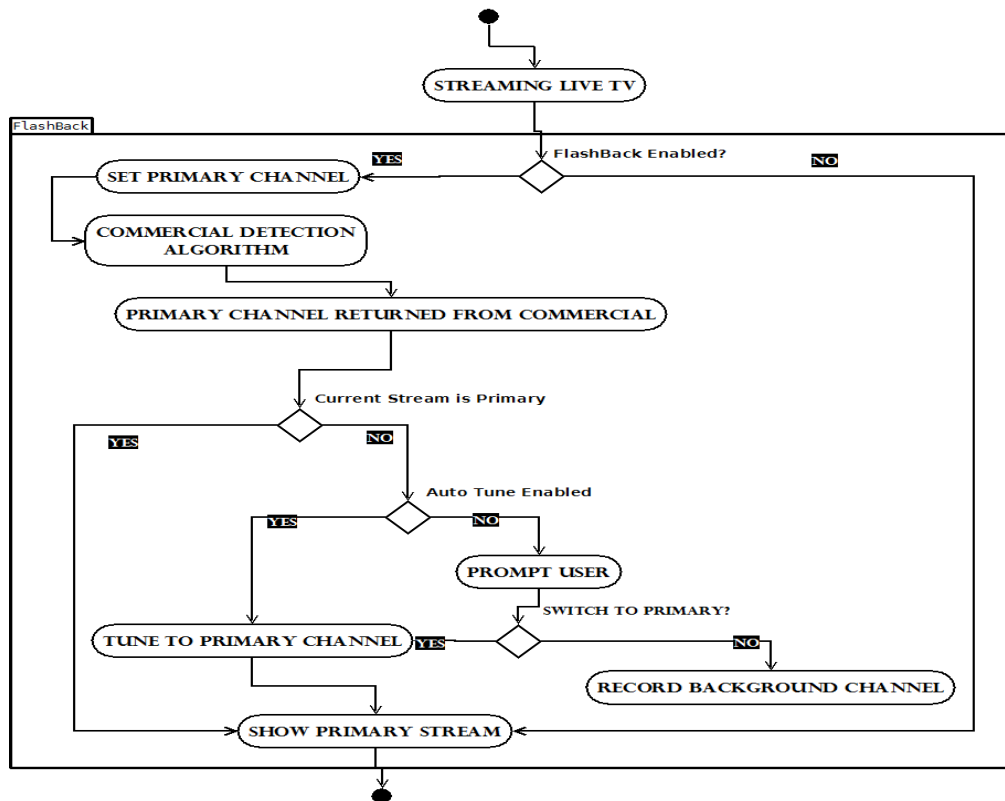


Figure 4: Flashback’s Activity Diagram

III. Hardware Design Integration

Flashback utilizes a 6 layer board, as shown in **Figure 3**, previously shown. In order to route the components in reasonable pcb real estate, using Eagle, ports were ranked on priority of critical timing constraints. Therefore, HDMI, ethernet, and usb data lines are top priority while keeping the chassis in mind. After critical data signals, came the segregation of power supply and analog components. Power signal traces could cause noise to data lines and analog signals could contribute unwanted high frequency components during logic state changes if not properly separated.

Since Flashback incorporates signal traces of this magnitude, discontinuities must be avoided in several situations when routing: 1) where the solder pads meet signal traces; 2) where signal traces meet vias; 3) right angle signal traces; 4) where signal pairs split to route an object. Poor routing execution and any discontinuities present compromise the signal's integrity, thereby compromising the design.

To aid in the routing, Flashback needed to follow a few guidelines. The instances where it was necessary to route around an object, it was done so in parallel, while maintaining the line to line spacing. This project also needed to deeply consider via clearance. There are dozens of them in the design and it was ideal to make sure that the clearance section did not interrupt the path of return current on the ground plane. Lastly, when placing passive components in the signal path, such as coupling capacitors, they needed to be placed next to each other. Although that creates wider trace spacing issues, it helps avoid lumping discontinuities

IV. Software Design Integration

Flashback's intricate software involves the establishment of the proper environment and Graphical User Interface (GUI) in order to operate at an optimal level. Also, the commercial detection algorithm mentioned must all be verified before any logic can be determined, in a sort of "checks and balances" type of manner to increase accuracy

A. System Architecture

In our initial requirements we decided that Flashback, as a system needed to be small in size, both in software and hardware. The ARM architecture provides a small footprint and impressive processing power. To make efficient use of the ARM architecture's features, a customized open source Linux Kernel and operating system have been deployed. The Linux Kernel is based on the Linux mainline branch release, version 3.2 and is

named Gumstix-Overo-Kernel. This kernel is configured for a Texas Instruments OMAP 3 Series processor, which is built on ARM Architecture. A desktop environment that supported Qt GUI applications was necessary along with an operating system that supports an ARM processor. The best fit for this system was the Ubuntu operating system. Specifically Flashback makes use of a port of the Ubuntu operating system that was developed by Linaro.

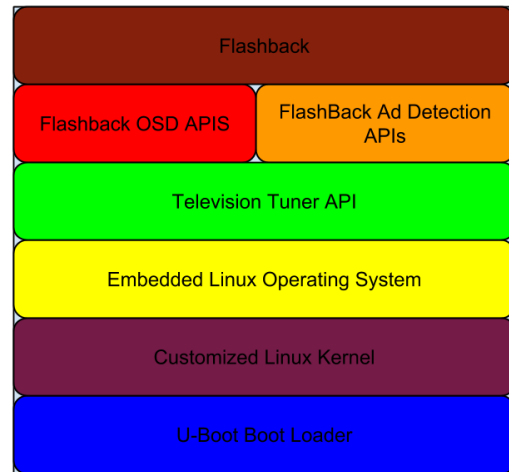


Figure 5: Flashback System Architecture Layers

To summarize Flashback as a system, it has been depicted in **Figure 5**. Starting at the bottom layer is the bootloader which is provided by Gumstix, Inc. This will load the custom configured linux kernel at shown as the next layer above the boot loader in **Figure 5**. The operating system at the third layer in **Figure 5** will then load all the drivers and kernel modules at startup to allow the television tuner API of layer four to be valid. The television tuner API provides an abstraction between the lower level television tuner commands of the linux kernel and the higher level Flashback software. This layer allows for easier access to the necessary data and commands to build more complicated features into Flashback. The layer above the television tuner API contains the business logic of Flashback, that is, it contains the main commercial detection logic and threading that is the core of Flashback's software.

B. Qt GUI

The Qt GUI framework was an easy choice to make given how compatible it is with different operating system platforms and architectures. Qt also has a rich API in regards to GUI development. The GUI is the view port for users to use when they are watching television programming and is the only interface between the user and all of the functions Flashback has to offer. Of the main goals our GUI development has, the most important one is

to provide users with an easy to use interface, one that has a familiar look and feel, similar to that, of a standard DVR. The menu's and notifications are designed to be minimally invasive so users can maintain focus on the program they are viewing while changing settings or being notified about the status of another channel that was being monitored by the Flashback algorithm.

C. Logo Detection Algorithm

Of the different triggers chosen, logo detection requires the most static references, because a database of images associated with the channels they [images] represent is required for this algorithm to work. In this algorithm it is better to use a large sized image, due to fact that there are more pixels available for comparison and a separation of pixel differences is less sporadic. The logos that are displayed on a channel's stream during normal programming will exhibit a change in pixel densities, which can prove troublesome during image comparison. To solve this issue, functions that were available in openCV, were used to Gaussian blur the image then normalize the image into the gray scale. The image of the logo from the database must then be subtracted from the instance of the logo from the scheduled programming and the new image is temporarily stored.

With this process we're able to effectively compare the stored logo for the channel with the logo that is currently visible to the user. Once a temporary mat object is created with the pixel data of the difference in the images we summed up all of the pixel densities and compared it to a threshold for which that specific logo is associated with. If the sum is larger than this threshold, it can be deduced that the logo is absent.

When the logo is absent the thread containing the analytics for this detection will emit a signal claiming that one of the triggers returned true for a commercial break.

D. Black Screen and Silence Detection Algorithm

Before a commercial break occurs, a few seconds of black frames and silence occur. Taking advantage of this known truth, we have our second trigger for commercial detecting. To perform analytics on our frames, we resize the frames to a smaller size for faster calculations on the sum of pixels within a given frame. The logo for the channel will also be removed from these frames to remove any uncertainty when obtaining the sum of the pixel values per frame. Ideally, the sum of pixel values for which black frames occur, is zero, however, after testing we've seen that the sum of pixel values becomes considerably small compared to the sum of pixels we obtained for frames that

have color. After a considerable amount of testing, it was concluded that any frame with a sum less than 500 can be determined to be a black frame. For the silence detection portion of this algorithm, statistics were introduced and used. The data used in these statistics were the standard deviations of the audio frames for the given samples. The reason statistics needed to be introduced, instead of simply checking for a value of zero from an audio frame, was due to the noise that silence produces; taking the standard deviation of each frame in a given sample we can determine where the audio value is close to zero or an extremely large value. Once this thread [Black screen and silence detection] determines that a black video frame along with silence has occurred, another signal will be sent claiming that black screen and silence has occurred at a given point in time.

E. High Cut Rate Detection Algorithm

High cut rates occur during commercials, versus standard programming. A cut occurs when a change in pixel densities is rapid in relation to the standard deviation in the frames in a sample. This algorithm is constantly finding the standard deviation from the current sample and comparing it to the standard deviation of recent samples, if that standard deviation is significantly larger or smaller than the recent standard deviation we can deduce that there has been a cut.

Cuts can occur during any part of the scheduled programming, but what matters is counting these cuts during a given time frame. Currently, the algorithm counts cuts for every thirty seconds of video and when the cuts are above ten the cut rate detection thread sends a signal to the main thread letting the GUI know that a commercial break has been detected.

The cut rate is the most flexible of the three detections, because the sample rate can be change to better fit the intended live programming. It's a matter of statistics for individual cases; since not all programming is universal, there is no one, hard – coded method that will apply to all over the air transmitted streams.

F. Commercial Detection Check

As shown in **Figure 6**, all three of these detection methods must pass or Flashback will not deem the frame sample as a commercial break. It is highly improbable that regular programming will be deemed a commercial due to this process and logic.

It is taken into account that some programming may, for example, have a high cut rate for a noticeable period of time, but since there probably wouldn't be a black screen

or silence detection, or the logo removed, it would still pass as regular programming, justifying the algorithm.

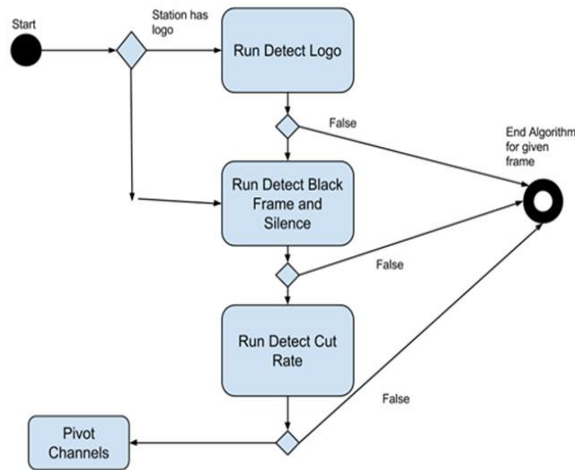


Figure 6: Commercial Detection Algorithms Activity Diagram

G. Debugging

Software debugging is the most time consuming portion of the software development process. If it can be shortened with efficiency then more time and effort can be dedicated to implementing more features within the software. The idea of ensuring an efficient debugging process led us to make major design decisions that were outlined earlier.

One such decision was the addition of Ethernet, Wi-Fi over USB and the inclusion of a serial console over USB. We made heavy use of the Secure Shell Command (SSH) within the Linux command line terminal while a network was available. Using the SSH protocol allowed for easy file transfer, software package updating, and a convenient remote interface with the Flashback device. The biggest benefit of all is being able to make minor software changes and rebuilding the code base locally on the Flashback device.

While the SSH protocol is extremely helpful for real-time debugging, a serial console over USB was just as helpful for pre-boot debugging. The serial console over USB was used for setting environment variables of the boot loader. This allowed for quick testing and post-build configuration of the many newly configured kernels that were deployed to the Flashback system. Using the SSH protocol over Ethernet and Wi-Fi over USB, along with serial console over USB were and still are extremely helpful.

V. Conclusion

This project has been pursued over the course of the 2013-2014 academic year with the hopes of being able to understand and implement commercial detection and the machinery necessary to achieve it. With the utilization of the commercial detection described, Flashback demonstrates a potential revolution for the common television audience.

With cooperation from major television broadcasters, the possibility of expanding Flashback's range to cable television could have been implemented, but for proof of concept and claim, with respect to our patent application, design simplifications were made, with the intent of future endeavors.

The current design of Flashback has not met our initial ambitious standards and has not realized its full potential. We are also aware of the limitations of the design and have been notified of ways the overall efficiency can be improved. It is a unanimous intent to revisit the design and implement what we've learned post graduation and fulfill our initial ambitions.

Acknowledgements

The authors would like to thank Dr. Samuel Richie for his helpful project supervision and helpful advice that led to making important project design decisions.

We would like to thank Edward McNicholas of General Dynamics C4 Systems, for his assistance with the layout and design of our PCB. Furthermore, we would like to thank him for his advice with respect to smaller software issues of the project. We would also like to thank our review board for providing us with their valued time and feedback. Lastly we would like to thank friends, family and colleagues that have supported us.

References

- [1] (Ong, C. Y., Song, J., Pan, C., & Li, Y.(2010, May). Technology and Standards of Digital Television Terrestrial Multimedia Broadcasting [Topics in Wireless Communications], Communications Magazine, IEEE , 48(5),119-127)
- [2] (Texas Instruments (2007 ,June). HDMI Design Guide. Texas Instruments Guides, 2(15) – 3, 5)

Authors



Rene Martinez is a double major student, pursuing an Electrical and Computer Engineering degrees at the University Of Central Florida. He has worked for Lee County Electric Cooperative and Geo Semiconductor Incorporated. After graduating in May, Rene will be working as a Software Quality Insurance Engineer for Hewlett Packard in Colorado



Trenton Reed, age 22, is currently attending the University of Central Florida, pursuing his B.S. in Electrical Engineering. He has worked in UCF's College Work Experience Program for the past two and a half years. After graduation in May 2014, he will be working with Texas Instruments as an Applications Engineer.



Marlon De La Cruz, age 22, is currently attending the University of Central Florida, pursuing his B.S. in Electrical Engineering with a minor in Mathematics. He plans to continue his education by obtaining his Master's soon after gaining adequate experience in the field. His interests include playing chess, basketball, and classical piano.



Marlon Smith is a Computer Engineering student at the University Of Central Florida. He is currently working at General Dynamics C4 Systems as a Software/Systems Engineering Intern. He plans to continue working with General Dynamics as a full-time Software Engineer after graduating in May 2014.

