Whitepaper

Making DVR Robust to Power-Loss

Challenges & Solutions

By Darshan Datt, Manager, Media Server Technologies – Systems, Ittiam

March-2018

Abstract

For most embedded systems, dealing with unexpected loss can be a challenge. Though such situations may arise due to power grid failures, it is paramount that the system recovers from such occurrences. While reliance on storage media for embedded systems has increased, storage media still remains most susceptible to power losses of all the components in an embedded system. This white paper analyses the impact of abrupt power losses on embedded systems and outlines storage media considerations for building power loss resilience in such systems.

Introduction

For most embedded systems, abrupt power loss is a pertinent scenario regardless of deployment environment. Power grid failures and manual interventions are most common reasons leading to abrupt power losses. Robustness to abrupt power loss is hence a tacit requirement for most embedded systems. Fulfilling this requirement needs all constituting components as well as their interaction to be robust to abrupt power losses

Of all components in embedded system, storage media remains most susceptible to power losses, despite all advances in technology. Additionally, with features such as field firmware upgrade, etc. becoming increasingly common, embedded systems' reliance on storage media has also increased.

This white paper analyses the impact of abrupt power losses on embedded systems and outlines storage media considerations for building power loss resilience in such systems.

Challenges

Abrupt power loss for storage media results in wide range of problems. Loss of uncommitted (unwritten) data, Corruption of data, Reduced disk space, Degradation of performance in terms of reduce read and write throughputs, complete loss of storage media, etc. are most common results of power loss. If not planned for, some of these problems render systems completely useless; others can lead to performance degradation.

Building a power backup mechanism, capable of holding power, allowing a clean shutdown of the system is not always viable, as such mechanisms would need inclusion of batteries or super capacitors that may not be feasible.

Solution

Storage media used in embedded systems can be classified into read only and programmable (read-write) storage.

Read only storage media are programmed during system manufacture. Any corruption or loss of data in this storage media would need system to be recalled from field to be repaired. Read only storage media are usually highly robust to power losses. Choosing right hardware parts ensures robustness of the embedded system, with respect to read-only media.

Contents of programmable media can change over the course of deployment of the system. Read-write storage media susceptibility to power-loss depends on ongoing operation at the time of power loss. Also, content stored in storage media has varying significance with-reference-to working of embedded system; some content would be more critical to system operation than others.

Recommendations below leverage content criticality and programmable storage media behavior to improve robustness of the embedded system to abrupt power losses.

Details

All content stored in programmable storage media in any embedded system can be classified into one of the following categories

- **Firmware**
  These files are most critical to operation of the system. Any loss or corruption of these files renders the system in-operative. These files do not change during regular operation of the system.
  e.g: uBoot, Device tree, Kernel, root file-system housing all binaries needed for execution, debugging, software update.

- **Firmware Settings**
  These files are necessary for operation of the system. These files may change during regular system operation based on user operation.
  e.g: System boot up configuration files, Password files, etc.

- **Data**
  Data represents input/output of the system. System continues to operate with loss/corruption of these files. However minimizing data loss due to abrupt power loss is highly desirable.

Following characterizes behavior of programmable storage media with-reference-to power loss.

- Probability of storage media failure is negligible, if the media is not being accessed at the time of power failure
- Probability of storage media failure is high, if power loss occurs during a write cycle as compared to a read cycle or idle cycle.
- Any uncommitted (unwritten) data stored in system's temporary memory is always lost due to sudden power loss.

Considering above, here are a few recommendations to increase power loss robustness in embedded system(s):

For Firmware

- Implementation of independent storage media for storage of firmware & firmware settings (e.g: eMMC, NAND Flash) and data (e.g: SD card, HDD, SSD). Usage of logical partitions within the same storage media is an alternate.
- Accessing firmware storage during initial part of power up for copying firmware to temporary memory. Processor can execute firmware instructions from this temporary memory.
- Firmware updates to be done in a contained environment to minimize probability of power interruption. Update of firmware should be verified through CRC check or equivalent measures.
- Redundancy be built with-reference-to firmware storage by maintaining multiple independent instances of firmware on the system at any instance of time.

For Firmware Configuration

- Firmware configuration update to be contained. This can be done by controlling all triggers in the system which result in firmware configuration update.
- Any firmware configuration update to be verified through CRC check or equivalent measures.
- Backup copy of firmware configuration be embedded with the firmware such that system can rely on this back up, in case firmware configuration gets corrupted.

For Data

- Applications on the system should minimize amount of uncommitted data at any instance of time. Committing (or writing) data to storage media on a regular basis, using smaller files instead of larger files are examples of achieving this.
- All considerations discussed with-reference-to firmware storage can be applicable to data storage too.

Availability at Ittiam

Ittiam offers IPs for managing embedded firmware and firmware configuration for embedded systems. Considerations in these offerings are consistent with above recommendation and enable designs to minimize power failure data losses. These are provided as add-ons to Ittiam's SDKs

Ittiam's adroitSDKs (recorder/playback) support split-files and playlists. Split-files allow for independently playable audio video files to be recorded to storage at regular intervals of time. By controlling the Interval, application can control and reduce uncommitted data at any instance and hence minimize loss of data.

Playlists allow multiple split files which are part of one use-case to be viewed and handled as one entity. This eliminates the overhead of application having to manage multiple files.

Why Ittiam?

Time-to-market advantage with off-the-shelf products

Tested and field-proven across a range of processors and operating systems

Patented, patent pending technologies enable high performance

Conclusion

Ittiam offerings enable customers build embedded systems with resilience to power losses.

For more information

Please check Ittiam's field proven Media SDK adroitSDK for embedded systems: https://www.ittiam.com/product/adroitsdk/

About Ittiam

Ittiam Systems is a global technology company with deep R&D driven solutions for media creation, management and consumption, and visual analytics. We provide advanced media codecs, software development platforms, systems and workflows for embedded and online applications to major mobility, cloud and media technology companies around the world. Our solutions are at the heart of over a hundred million lifestyle products that drive mobility, content access, networking and sharing.

About the Author

Darshan Datt K S is Member Technical Staff at Ittiam and manages Media SDK team in the Media Server Technologies – Systems business unit. He has been a part of the company for over nine years, and has extensive experience in development of media systems as well as SDKs (for streaming, recording, playback, transcoding) for embedded platforms. He has one patent to his name, and a B.E degree from Sri Jayachamarajendra Collage of Engineering, Mysore.

Disclaimer

This white paper is for informational purposes only. Ittiam makes no warranties, express, implied or statutory, as to the information in this document. The information contained in this document represents the current view of Ittiam Systems on the issues discussed as of the date of publication. It should not be interpreted to be a commitment on the part of Ittiam, and Ittiam cannot guarantee the accuracy of any information presented after the date of publication.