Introducing the Linux Vendor Firmware Service

Abstract

Updating firmware on devices is very hard for users on Linux systems. Not knowing exact hardware details, where to look for updates and how to run the Windows-specific flashing tools makes it almost impossible to update firmware on devices. As a result, "*broken*" hardware is being RMA'd to the vendor and customer systems are left in an insecure state even when updates have been released that fix the specific issues. Linux as the OS is now mainstream and vendors need to support these customers.

The LVFS is a secure web service that can be used by OEM's to upload prepared firmware files and can also be used by users to securely download metadata about available updates and optionally, the updates themselves. The LVFS is growing fast, delivering metadata to tens of millions of clients every month.

Hundreds of thousands of customer devices are being updated every month thanks to the LVFS.

Revisions

Revision 1.03	25th May 2016
Revision 1.05	29th May 2016
Revision 1.06	3rd June 2016
Revision 1.07	22nd August 2017
Revision 1.10	2nd February 2018

Introduction

Linux users have traditionally had problems with keeping hardware up to date with firmware updates. The problem is threefold:

- They do not know what exact hardware they have installed, the current firmware version, or even if the devices support being upgraded at all.
- They do not know where to look for updates; searching the various vendor websites is an exercise in frustration and as a result most users do not bother.
- The Windows-specific flashing tools do not work on Linux; a significant number of Linux users keep a Windows virtual machine for essential business-critical software that is not available on Linux. This will not work for firmware update utilities that require low level hardware access.

The fwupd project can query supported hardware for the current firmware versions and also deploy new firmware versions to devices, but requires metadata from the LVFS to know the details about available updates. It also requires vendors to prepare the firmware with the required metadata and to use a standardized deployment framework e.g. DFU or UEFI UpdateCapsule.

Using the information from fwupd higher level software centers (like GNOME Software[5]) can show the user the update description in their own language and offer the update to be installed using just three clicks of the mouse. Security updates are handled in the same way as other OS updates meaning it is just one mechanism for the user to understand.

The LVFS supplies the data in a secure format, allowing the fwupd project to install the update safely. Existing approaches have been OEM specific which meant that a large amount of engineering effort was required, making this approach only financially viable for enterprise use-cases which provided either a higher profit per unit or a subscription for ongoing support.

There are a significant number of legal problems with the redistribution of firmware, and we have been working with vendors finding acceptable methods of redistribution whilst ensuring confidentially throughout the process. Being backed by a large Linux vendor with heterogeneous support for many vendors and platforms puts the LVFS in exactly the right place to build this kind of shared infrastructure.

System Architecture

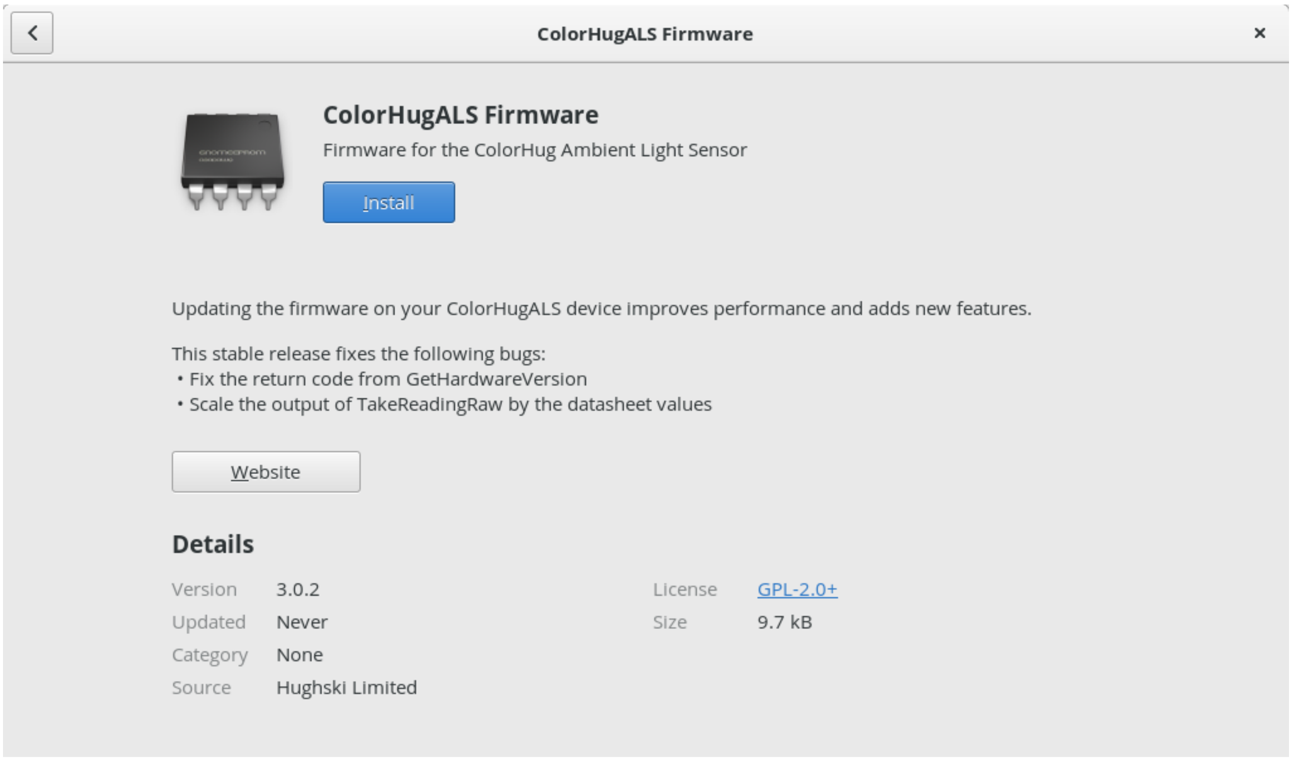
The architecture is built into three layers: a presentation layer, a mechanism and a data-provider and each can be replaced as required as they all use standard protocols.

GNOME Software

GNOME Software is an application store designed to make installing, removing and updating both easy and beautiful. It is available for Linux and used by millions of people on the following distributions:

- RHEL 7
- Fedora 22 or newer
- Ubuntu 16.04 (Xenial) or newer
- Debian 9 (Stretch) or newer

At session start-up (and on every start thereafter) the metadata XML and detached signatures are checked for a specified age, and if required newer files are automatically downloaded from the LVFS and pushed into `fwupd` over D-Bus. When the update list is required we query the `fwupd` daemon over D-Bus for any pending updates. If there are updates that need applying then they are downloaded and the user is notified and the update details are shown in the specified language. The user has to explicitly agree to the firmware update action before the update is performed.



ColorHugALS Firmware

Firmware for the ColorHug Ambient Light Sensor

Updating the firmware on your ColorHugALS device improves performance and adds new features.

This stable release fixes the following bugs:
- Fix the return code from GetHardwareVersion
- Scale the output of TakeReadingRaw by the datasheet values

Details

Version	3.0.2	License	GPL-2.0+
Updated	Never	Size	9.7 kB
Category	None		
Source	Hughski Limited		

fwupd

This project provides a system-activated daemon (`fwupd`) with a D-Bus interface[3] that can be used by unprivileged clients. Clients can perform system wide upgrades and downgrades according to a security policy, which uses PolicyKit to negotiate for authorization if required. The command line tool (`fwupdmgr`) can be used to administer headless clients on the command line over SSH or using a management framework like Red Hat Satellite or Dell CCM[7].

The daemon processes metadata from the LVFS in AppStream[6] format along with a detached GPG or PKCS#7 signature. The `.cab` archives which must contain at least a `.metainfo.xml` file and a detached GPG/PKCS#7 signature of the firmware payload. Other files are permitted in the archive which allows the same deliverable to be used for the Windows Update system.

Internally fwupd creates a device with a unique ID, and then a number of GUIDs are assigned to the device by the provider. It is these GUIDs specified in the update metadata file that are used to match a firmware file to a device. Although it is usually the responsibility of the system vendor to generate a new GUID if the hardware requires a different firmware file, we can match an update that only applies to specific versions of hardware.

The fwupd project ships a number of providers:

- `UEFI`, which reads the ESRT[4] table and adds devices found on the system and then schedules an offline update to run a custom EFI binary.
- `DFU`, which reads and writes data from USB devices supporting the DFU interface.
- `ColorHug`, which supports a custom HID protocol.
- `Logitech`, which supports the Unifying protocol.
- `Thunderbolt`, which supports Thunderbolt controllers and devices.

Adding more providers is of course possible, but where possible vendors should use the existing code and for instance add an ESRT data table when building the system firmware.

When the user agrees to a UEFI firmware update the firmware is unpacked into the EFI System Partition, several UEFI keys are set and the system reboots. On reboot the `fwupdate.efi` binary is run before the boot-loader is started and the `UpdateCapsule` action is performed. For USB devices the device is normally updated live without requiring a reboot.

LVFS

The LVFS provides a OEM-facing website that requires a username and password to access the secure console. There is no charge to vendors for the hosting or distribution of content.

This service should only be used to distribute firmware that is flashed onto non-volatile memory. It is not designed for firmware that has to be uploaded to devices every time the device is used.

When `.cab` firmware files are submitted the following actions are performed:

1. The update metadata in the archive is checked.
2. The firmware capsule is signed with our GPG key or PKCS#7 certificate.
    - Clients **do not** verify the signatures in the catalog file as this is for WU only
3. The new cab file is repacked.
    - Only required files are included in the cabinet file, typically making the download size much smaller
4. The metadata is added to our database.

Many ODMs are distinct and decoupled from the OEM, and in most cases the ODM is allowed to upload new firmware but not make it available for users. For this use case, three classes of user exist:

1. The admin user that can do anything
2. Unprivileged users that can upload files to the testing target
3. QA users that can upload files to testing or stable target, and can move files from testing to stable

Performance

The LVFS runs on a dedicated virtual private server and uses the Amazon S3 CDN for the public metadata. Every month there are over 50 million files being downloaded from around 10 million unique clients. About 5% of the firmware downloads are performed using the `fwupdmgr` command line client manually and 95% are done from the GNOME Software GUI program.

There are currently 161 devices supported on the LVFS and 374 different firmware versions and the system runs consistently with a 99.95% uptime at 3% loaded.

Conclusions

The LVFS has grown to be an essential part of the Linux ecosystem and there are currently 24 vendors using the service, 9 of which are multi-billion dollar companies. 4,000 **new** clients use the service *every day*, with over 3,000,000 returning users. At this point, using anything other than the LVFS to distribute firmware for Linux users would not make sense.

Future Work

Various vendors are working on custom providers for fwupd as they either cannot retrofit older hardware with the ESRT data table, or because they want more control over the low level flashing protocol. We certainly would encourage any new vendors wanting to use the LVFS and fwupd to use a well-known standard like DFU or UEFI UpdateCapsule with ESRT as it means there is no application code to write.

From a system administrators point of view, it will also soon be possible to get notified of updates and perform upgrades using the Cockpit framework as well as the usual client tools.

Looking into the future we might see the LVFS moving to a standards body like the Linux Foundation that is completely independent of any OS vendor. We think it would be the logical place for this kind of distribution and vendor neutral service. If this was to happen any uploaded firmware would be transferred automatically.

Related Work

The Dell Repository Manager[1] allows you to update the firmware on various models of Dell enterprise hardware. There are several software (e.g. the SSU and SBUU) and hardware elements specific to Dell (e.g., the LCC or USC) and most of the stack is proprietary.

Microsoft provides a service called Windows Update[2] which takes driver updates from vendors, optionally performs some quality control on the update, signs the firmware and then hosts the firmware on a CDN. The entire stack is proprietary and for Windows only.

Acknowledgements

Much gratitude has to go to Red Hat for funding my work on this. Many thanks to Amazon for sponsoring the CDN costs. Dell took an leap of faith and has been an early adopter involved from the very start. In particular Mario Limonciello has been immensely supportive with time, code and hardware. Peter Jones from Red Hat has also written a lot of the kernel-side and low-level code in `libfwupdate` which actually allows `fwupd` to schedule UEFI updates.

Glossary

- DFU: Device Firmware Update
- CDN: Content Delivery Network
- ESRT: EFI System Resource Table
- GPG: GNU Privacy Guard
- GUID: Globally Unique Identifier
- HID: Human Interface Device, e.g. a mouse or keyboard
- LVFS: Linux Vendor Firmware Service
- ODM: Original Device Manufacturer
- OEM: Original Equipment Manufacturer
- OS: Operating System
- QA: Quality Assurance
- RMA: Return to Manufacturer Authorisation
- UEFI: Unified Extensible Firmware Interface
- USB: Universal Serial Bus
- WU: Windows Update
- XML: Extensible Markup Language

Citations

1. Dell Driver and Firmware Update Strategies: 24th May 2016:

   http://www.dell.com/support/article/us/en/04/SLN293301?c=us&l=en&s=bsd&cs=04

2. Microsoft Windows Update FAQ: 24th May 2016:

   http://windows.microsoft.com/en-gb/windows/windows-update

3. Vendor page for fwupd: 24th May 2016:

   http://www.fwupd.org/vendors

4. ESRT Table Definition: 24th May 2016

   https://msdn.microsoft.com/en-us/windows/hardware/drivers/bringup/esrt-table-definition

5. GNOME Software Homepage: 25th May 2016

   https://wiki.gnome.org/Apps/Software

6. AppStream v0.9 Documentation: 25th May 2016

   https://www.freedesktop.org/software/appstream/docs/

7. Wyse Cloud Client Manager: 29th May 2016:

   https://www.cloudclientmanager.com/