Carter Dack, 2024.

Extracting and Examining SoC Firmware from Amazon Echo's embedded Multi-Media Card (eMMC)

Modern embedded devices such as the Amazon Echo are fundamentally similar to a typical computer. They contain a CPU, some sort of input and output (I/O), and a memory interface(s). eMMC is a memory format similar to a solid state drive (SSD) in which data can be loaded to it and persist across power cycles. eMMC is a compact and relatively inexpensive chip, which is why it is commonly used in mobile phones and embedded devices to store the device's firmware. This practice is commonly referred to as System-on-Chip (SoC), in which some or all of a computer's components are integrated into one integrated circuit.

This demonstration will show how eMMC packages can be removed from an embedded device, loaded into an adapter, and its contents extracted. In doing so, the embedded firmware will be examined, ideally to reveal sensitive system information.

Disassembly

When taking apart the "Amazon Echo Dot 2nd Generation", various components are revealed, primarily the "SKhynix H9TQ32A4GTMC BGA221" eMMC Flash Memory Chip, highlighted in red in Figure 1.3.

Figure 1.1 - Amazon Echo Dot Gen II

Figure 1.2 - Amazon Echo Dot Gen II Teardown

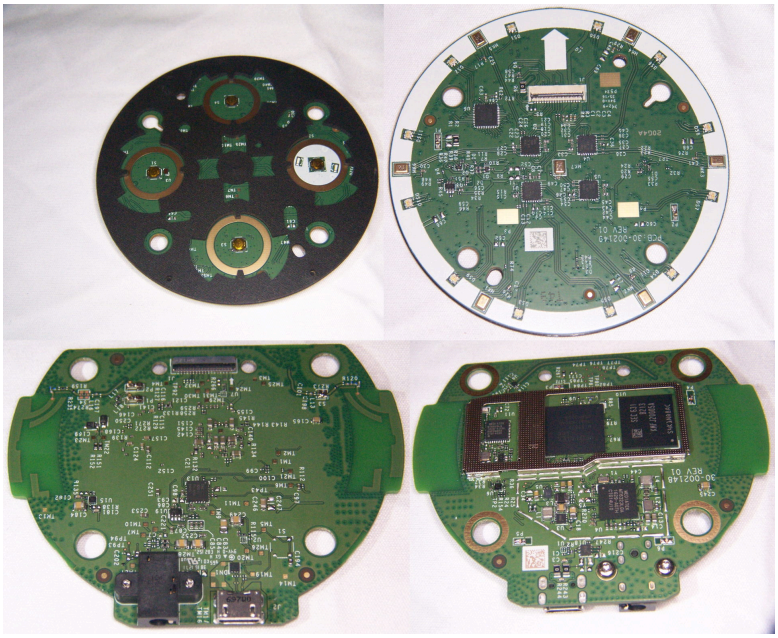
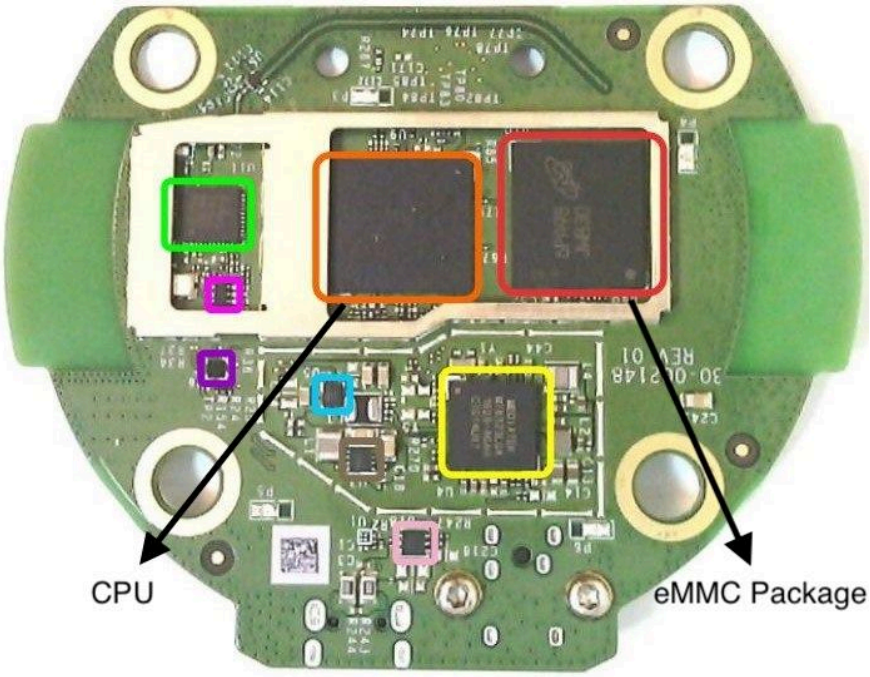


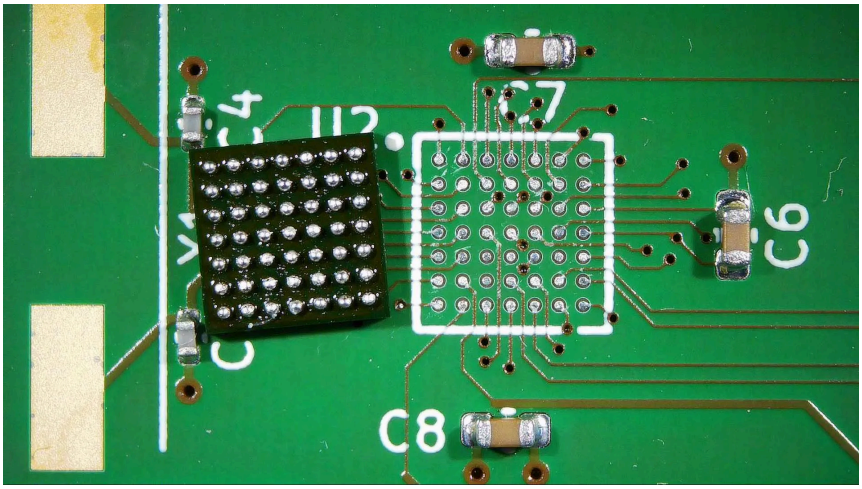
Figure 1.3 - Amazon Echo Dot Gen II eMMC Flash Memory Package



In order to be able to load the eMMC chip into an adapter and read its contents, it must first be removed from the circuit board. These chips often use Ball Grid Array (BGA) packages, in which a grid of solder pads on the underside of the silicon chip are used to connect it to the circuit board.

In order to remove these chips, some flux is added to the top before a heat gun is used to melt the solder on the underside of the chip. Once melted, the chip can be removed with tweezers before the solder cools down and re-solidifies, as demonstrated in Figure 1.4.

Figure 1.4 - Removal of a BGA Chip

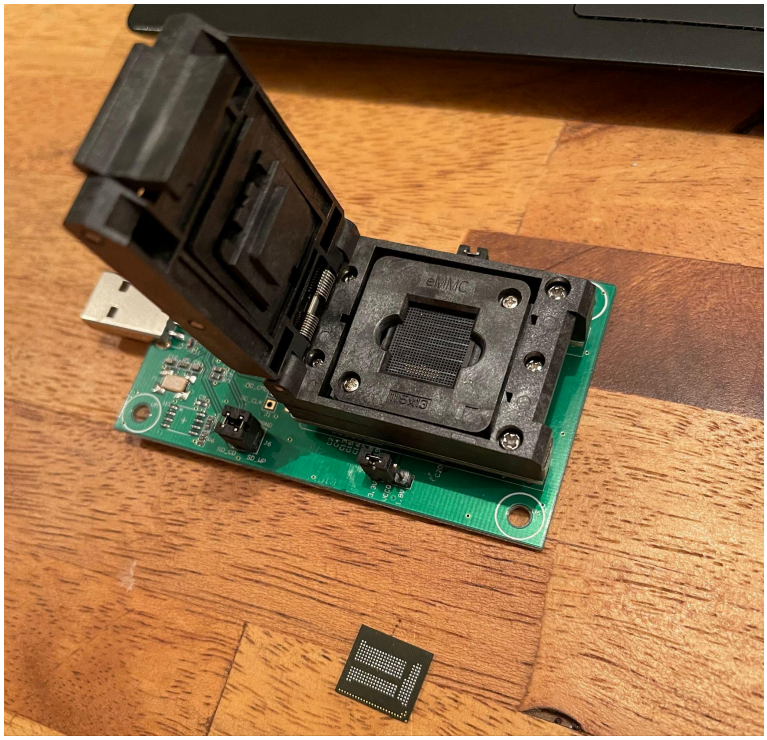


Once the BGA chip has been removed, any remaining solder on the underside of the array can be cleaned off using a solder wick. This step is important as the tolerances inside the adapter socket are quite small. Adapters can come in various sizes and prices, however they are commonly a clamshell style socket the same size as the chip that is being read, and inside the socket are very small pins that align with necessary pinouts on the ball grid array.

When power is applied, the adapter is able to read the chip and its contents (unless otherwise encrypted). The cleaned chip and its adapter can be seen in Figure 1.5.

Figure 1.5 - Cleaned Chip Ready for Adapter

Carter Dack, 2024.



Once the chip is loaded into the adapter (ensuring it is aligned properly with the pins and arrows), the adapter can be plugged into a laptop. Once plugged in, the adapter will be mounted as a block device with the various partitions of the eMMC chip mounted underneath. Using the “dd” command, the user can copy each partition to the local file system for further examination as seen in Figure 1.6.

Figure 1.6 - Mounted Block Devices to dd

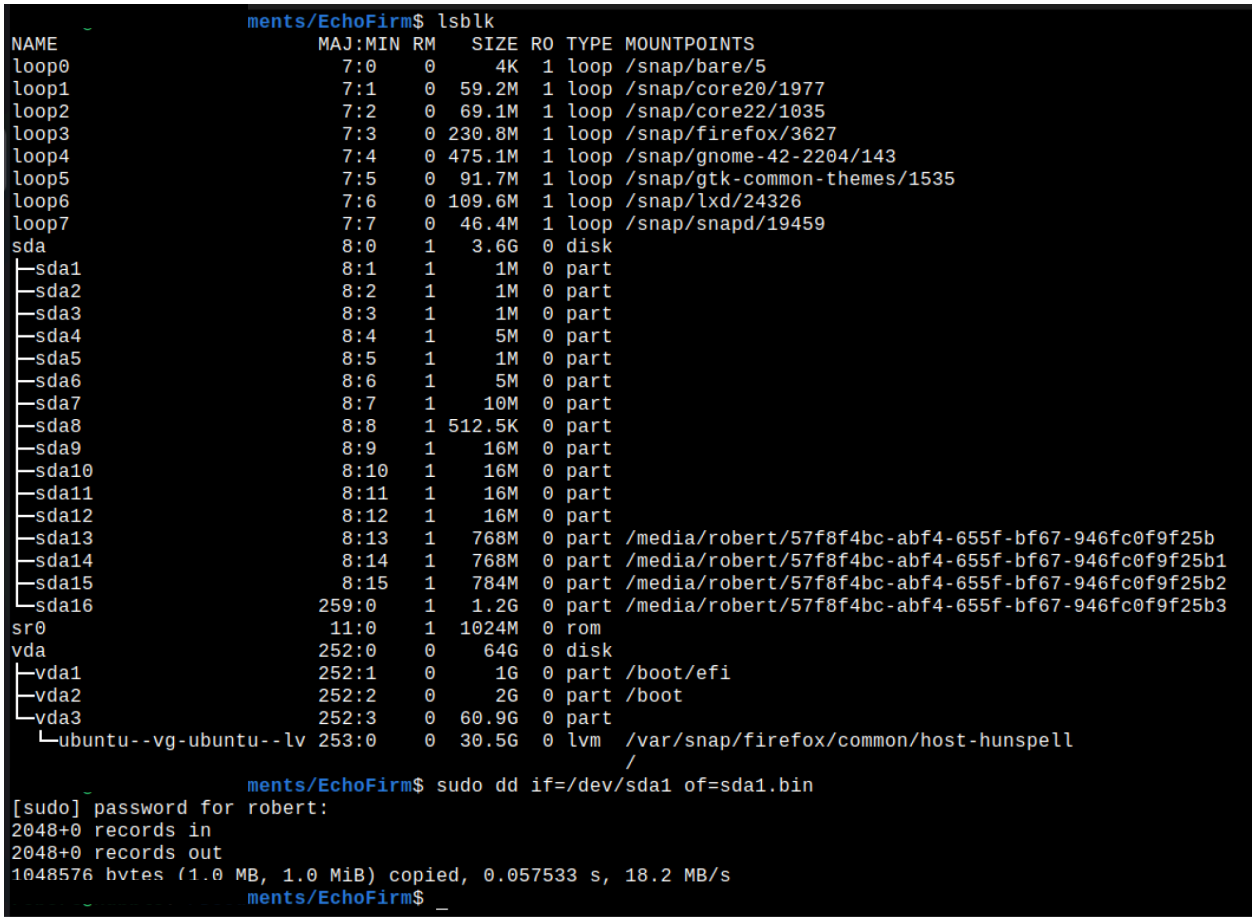


Figure 1.7 - Copied Partitions

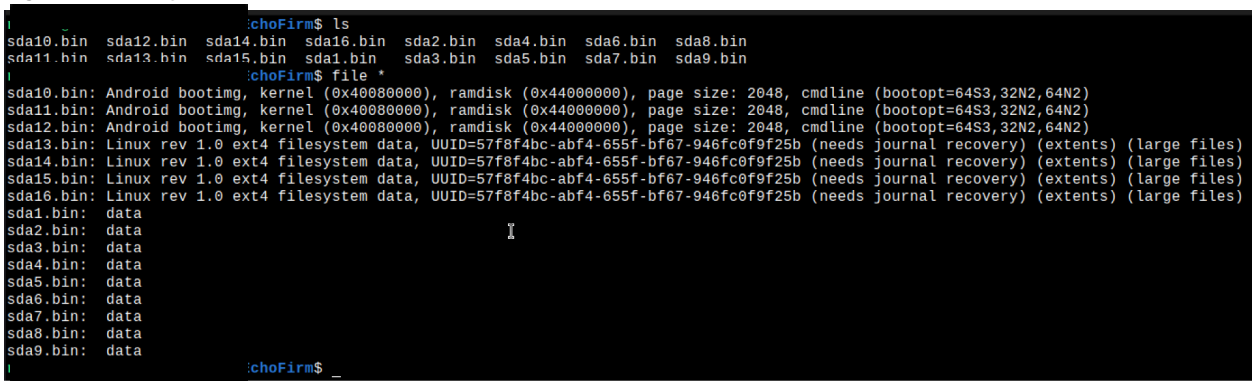
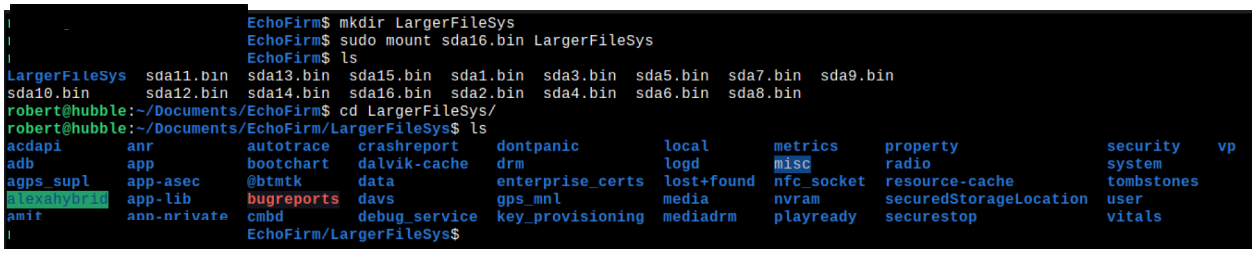


Figure 1.7 shows what types of files each of the copied partitions are. In this case, it would appear that there are three (3) identical Android Boot Images, three (4) identical Linux filesystems, one (1) larger Linux file system, and nine (9) data files. The most likely reason for the various identical partitions could be for ensuring continuity during updates, hot-swapping partitions during update cycles.

Carter Dack, 2024.

Examination will start with the larger Linux filesystem, which can be mounted to the local filesystem using the “mount” command. In doing so, the ext4 filesystem will be expanded, and the user is able to inspect its files and folders directly.

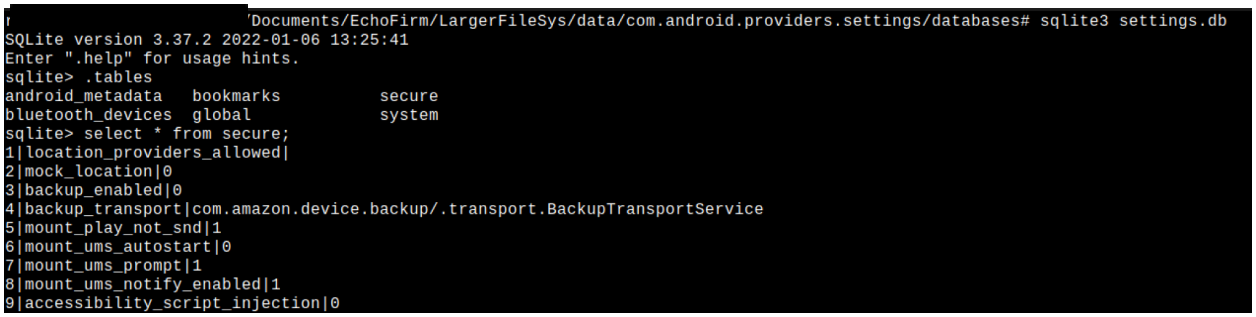
Figure 1.7 - Mounting FS



Within the expanded filesystem, there are many different files. Since this eMMC chip is effectively the flash memory of the device, data can and will be written to the filesystem upon normal use of the device, meaning there may be some sensitive information included in various locations.

There are numerous ways of searching for this data, from grepping for things like “password” to manually inspecting each file recursively. Many of the files that exist are sqlite3 database files, meaning we can use sqlite to comb through the various databases as seen in Figure 1.8.

Figure 1.8 - Looking Through Sqlite Tables



After doing some digging, the below pieces of sensitive information were identified along with their corresponding file paths. Although this information is somewhat sensitive in nature, the true “crown jewels” of the Amazon Echo were not found, which might be things like the word for word commands that were issued to the device throughout its lifetime, i.e. its search history.

It was noted however, that there was configuration that suggested the use of amazon owned APIs, to which the Echo device likely offloads these “crown jewels” for storage, compute, and data sensitivity reasons.

Figure 1.x - Sensitive Data Identified

Type of Data	Filepath
WiFi network name	/data/com.android.providers.settings/databases/settings.db
Amazon account owner first & last name	/system/users/0/accounts.db
Last time the user was active	/system/usagestats/0/yearly/x
Device's "friendly name"	/local/smarthomed/x5cecx15-cca5-52x0-ax11-6xb040eed3xxx/schema.json
Nearby bluetooth device names and addresses	/local/whad/btdevice.db.json
Out-of-box-experience (oobe) web setup private key	/local/oobe-web-setup.cert
Wifi network name + password	/misc/wifi/wpa_supplicant.conf
Connected device name, MAC address, and IP address	/misc/dhcp/dnsmasq.leases

Outlook

This exercise could be likened to a forensic investigation, as the files and folders contained within the firmware of our device were able to be inspected. Though not fully critical in nature, one could begin to imagine how the data extracted from this eMMC package could be leveraged, such as by obtaining the password of a WiFi network for further exploitation.

Additionally, this exercise exemplifies the various ways in which the data that most users never even think about might be stored, captured, and examined. eMMC chips are a unique storage component, however they have become ubiquitous, particularly in low power environments, making it more relevant than ever to understand how to examine these media containers.