Attacking Serial Flash Chip: Case Study of a Black Box

Emma Benoit, Guillaume Heilles, and Philippe Teuwen
{ebenoit,gheilles,pteuwen}@quarkslab.com

Quarkslab

1 Context

The original context that lead to the experiments of the techniques described in this paper was a black box study on an embedded device to be conducted in a very short time. But, rather than describing further this specific case, let's generalise the context to various situations where a physical attack on a serial flash is valuable. This can be a security evaluation of a product or a forensics investigation, whenever the device was the target or the tool of an attacker or maybe just a *witness* having stored some information related to a crime. These various cases often share the same constraints: there is no documentation or firmware image provided, physical tampering is allowed but shall be non destructive, only a few copies of the same product are available (at best) and time is a scarce resource.

In this paper, *embedded device* is used as a general term which encompasses various devices like network devices, industrial control systems (ICS) or Internet of Things (IoT) devices. Most of them rely on their low cost for mass-market adoption. Therefore, they often differ from a traditional system in terms of architecture, real-time OS, low resource usage, etc. and are seldom protected at hardware level, which makes physical approaches particularly effective. Hardware attacks have the reputation — especially among software security researchers — of being difficult, time-consuming, requiring expensive tools, material or skills.

But in recent years, the availability of low-cost hardware tools has drastically lowered the threshold of these attacks. Those are no longer reserved to entities with important resources, they are now affordable even for hobbyists. For security analysts, low-cost hardware attacks are just another tool at their disposal, which should become more and more common.

2 Flash Memory

Nowadays, flash memory chips are found in nearly all embedded devices and are commonly used as a non volatile storage medium to store data which seldom changes, like firmware and configuration data. Flash memories come in two main categories, depending on their memory interfaces. The *serial flash* has a serial bus interface, while the *parallel flash* has a parallel one. The choice between them depends on constraints like data transfer speed and available board space. Serial flash is preferred to parallel flash in embedded devices for its lower cost, smaller package and easier integration as it requires less pins from the microcontroller.

There is no standard way to retrieve data from any flash memory, each method will be specific to a type of chip packaging or a range of devices. While in some circumstances, specific flasher tools may exist for specific devices, we will focus on reading content from the flash memory chip directly, independently of the device itself.

Two options are possible. The *in-circuit* method leaves the chip untouched and attaches probes on the pins of the chip. Using a logic analyser, one can observe the data being read by the device and can reconstruct an image of the memory in use. The *chip-off* method consists in desoldering the chip physically from the printed circuit board (PCB) and reading its content using an EEPROM programmer. While the in-circuit method might suffice in some forensics investigations if the chip has accessible pins, e.g. a small outline package (SOP), it is not possible on complex packages, like ball-grid array (BGA), which have no visible pins and hide the underneath PCB layout. The chip-off technique obviously allows a better observation of the PCB layout, but it also eases more advanced attacks such as tampering with the content of the memory, swapping memory chips (e.g. to validate hypotheses on OTP bits usage), or even conducting man-in-the-middle attacks.

Our contribution is to show that the chip-off technique can be made easily accessible, with off-the-self components and tools, to provide valuable results to security analysts in a matter of hours.

3 Details of the Chip-Off Technique

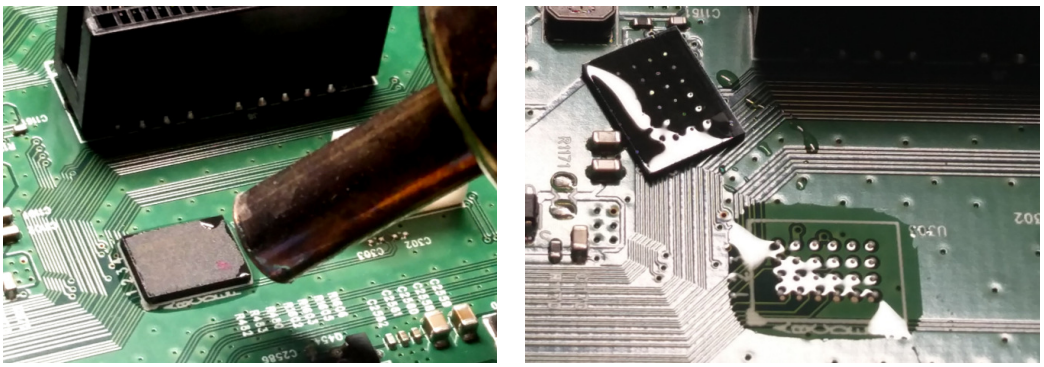
3.1 Identification

The first step while facing an unknown embedded device is to identify its main components, communication ports and debug interfaces, if any. Let's assume this has been done and some promising serial flash chip has

been identified. To illustrate the technique in the following sections, we'll focus on the chips found in our embedded device: two integrated circuits (ICs) in BGA packages, labelled `MX25L3254EXDI` and `MX25L3255EXCI`. BGA packages are not standardised and vary greatly in grid disposition, pitch and balls size. The markings on the chips helped pinning the exact model: `MX` stands for "Macronix International", the manufacturer while `MX25L3254EXDI` and `MX25L3255EXCI` are product denominations. From the datasheet, the ICs are common flash memory, often found in embedded devices: NOR gates are used as the underlying memory technology and SPI as memory interface. They both have a size of 32 Mbit.

3.2 Desoldering

To desolder a flash, a thermal method relying on the usage of a heat gun and a preheater was used, as illustrated by Figure 1. The principle is to apply some flux and to heat the flash memory until the underneath solder balls are melted. This method is simple and fast, and the chip can be removed from the board within a few minutes.



(a) Heating flash (b) Desoldered flash

Fig. 1. Flash desoldering with a heat gun.

Adjacent components will also be affected by the heat and some care must be taken to avoid moving them. The flash chip must not be exposed to heat for too long as this might damage it.

3.3 Designing Adapter Boards

An extract from the datasheet of the flash chips is shown in Figure 2 and describes the pin layout (colours are ours).

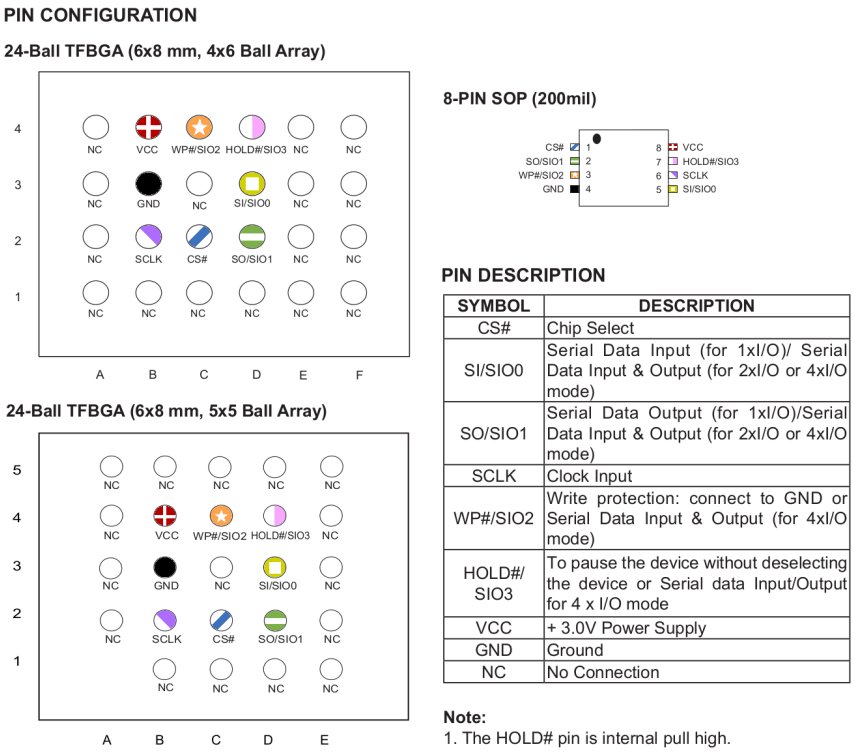


Fig. 2. Pin configuration of the flash chip.

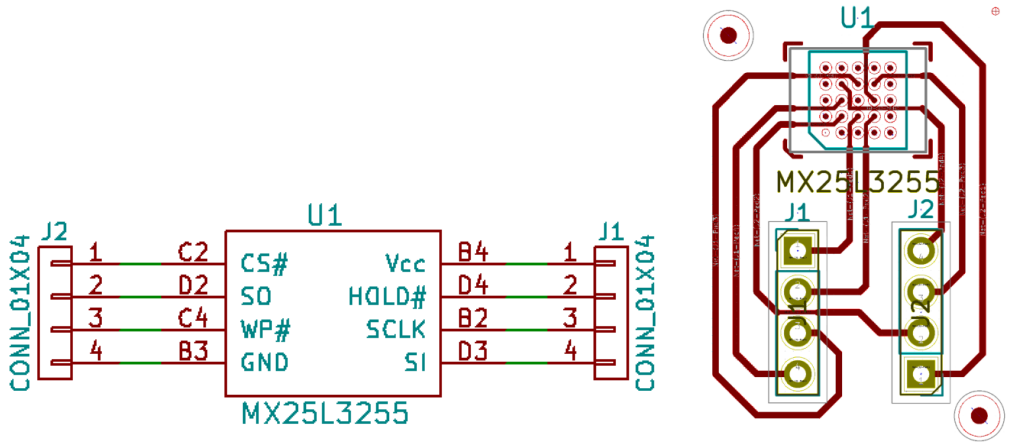
The same IC is available in three different packages: an 8-pin SOP, a 4×6 BGA and a 5×5 BGA. Our MX25L3254EXDI follows the 5×5 disposition and the MX25L3255EXCI the 4×6 one. Of the 24 balls of the BGA, only eight are actually useful, the other pins are marked "NC" which stands for *no connection*.

To communicate with the chips, adapter boards are required to expose the useful pins. If no datasheet is available or a chip can't be identified, some probing and reverse engineering of the device's PCB might be needed to identify the type of bus and recover the function of each pin.

The design of the PCBs was realised using KiCad[1], a popular open source electronics design automation (EDA) suite. First an electronic schematic is created in Eeschema, representing the theoretical electrical circuit. The flash chips are specific components which are not available in the standard KiCad library. Therefore, customised electronic schematics and footprints need to be designed, using the pinout diagrams from the datasheet. The adapter boards are simply composed of two 1×4 headers for the 8 useful pins and of the BGA grid where the flash IC will be soldered. Once the BGA footprint is created, footprints are added for

[1] `http://kicad-pcb.org/`

each component in Pcbnew and tracks are routed to connect them. The electronic schematic and the final PCB design for the adapter board of the `MX25L3254EXDI` can be seen in Figure 3.



(a) Electronic schematic (b) PCB design

Fig. 3. Adapter board for the MX25L3254EXDI.

The two 1×4 connectors are arranged to mimic the SOP8 layout on a dual in-line package (DIP). This arrangement will be useful later when interfacing with an EEPROM programmer.

3.4 Making PCBs

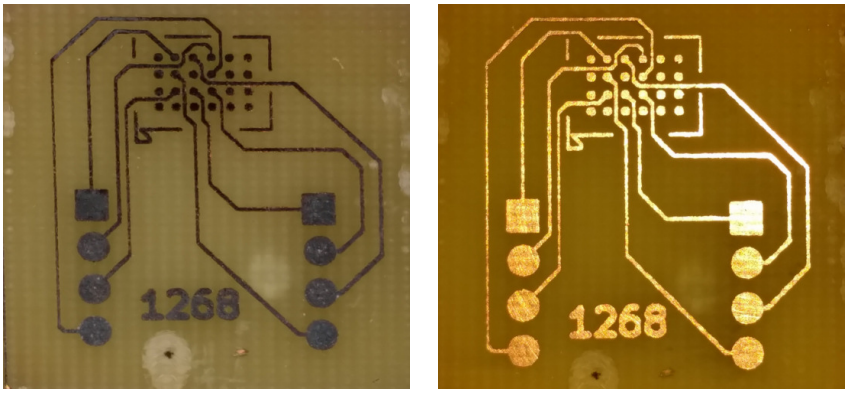
The KiCad design file can be sent to a PCB manufacturer to obtain an actual PCB. However, the manufacturing and shipping delays do not always fit the time constraints of security analysis missions or forensics investigations, especially if results are expected within a few hours.

Several in-house techniques were investigated, which we will describe:

- A chemical technique, using *etching*;
- A mechanical technique, using computer numerical control (CNC) *milling*;
- A mixed technique, using a *laser* on a CNC and chemical etching.

Chemical Technique: Etching refers to the process of using a chemical component to “bite” into the unprotected surface of a metal. Ink is used as a means to delimit the copper routes. To reproduce the design of the adapter on the copper, a toner transfer method is used: the design is

printed on paper and transferred to the copper using heat and pressure. Usually, this is performed with an iron, but we found out that using a pouch laminator in place of the iron gives better results as the heat and pressure are applied more uniformly.



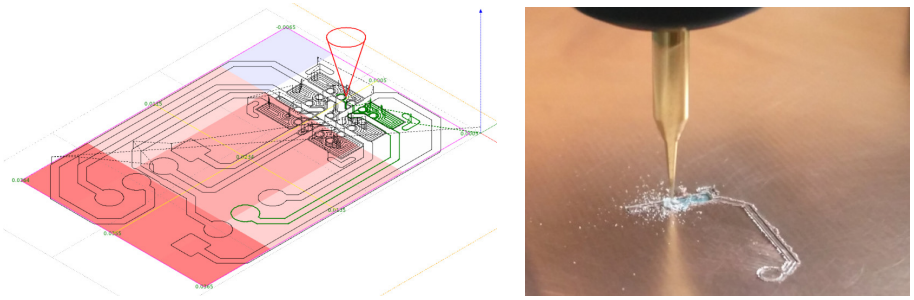
(a) After etching (b) After cleaning

Fig. 4. PCB manufacturing by chemical etching.

The PCB is then immersed into an etching solution of sodium persulfate. As illustrated in Figure 4, copper which is not covered by ink is removed, then the transferred ink is removed using acetone.

Mechanical Technique: To trace routes in the copper layer, a CNC milling machine carves out only the outline of these routes, so the excess copper is left in place. KiCad cannot directly produce a file compatible with a CNC machine. Therefore, the design is exported from KiCad to a Gerber file and imported into FlatCAM[2], a PCB Computer-Aided Manufacturing (CAM) software, to generate the routes outline. The result is then exported to an STL file and imported into bCNC[3], which controls the CNC by sending commands to it. bCNC automatically ensures the levelling: it measures the actual height of the board in several points as the board is never perfectly flat. The result is a "heat map" dynamically used to adjust the tool height depending on the position. Figure 5 shows the FlatCAM outline imported in bCNC, the heat map and the milling process.

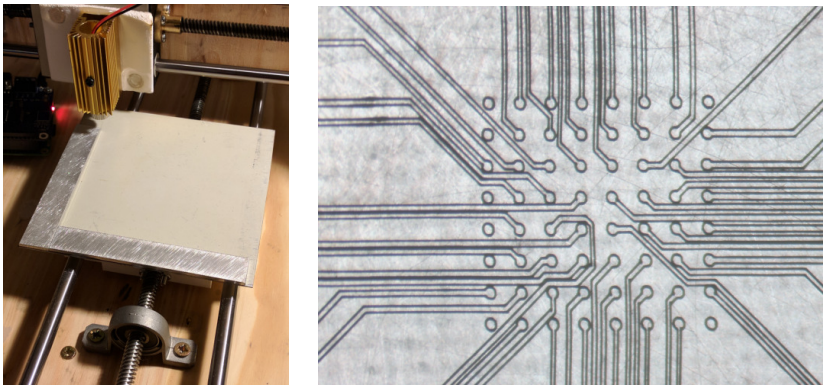
[2] `http://flatcam.org/`
[3] `https://github.com/vlachoudis/bCNC`



(a) Control of the CNC (b) Milling in progress

Fig. 5. PCB manufacturing by mechanical etching.

Laser Technique: Some BGA chips are so dense that the space between two pads is typically less than 0.5 mm and with homemade PCBs, we often have to route two tracks between two pads. Taking into account clearances, this leads to e.g. 0.15 mm tracks and 0.05 mm clearance, which is not feasible with the techniques detailed in the two previous chapters. This technique uses a blue laser to remove some black acrylic paint sprayed on the PCB. Then the PCB is cleaned with an ultrasonic cleaner and etched chemically. Eventually, the paint is removed with acetone. As for the mechanical etching, the laser only removes the outline of the tracks. A high-precision XY table has been developed from scratch with lead screws and anti-backlash nuts to help minimising problems of backlash and repeatability encountered in earlier tests. A cheap 1500 mW laser module is mounted with an anti-reflective lens.



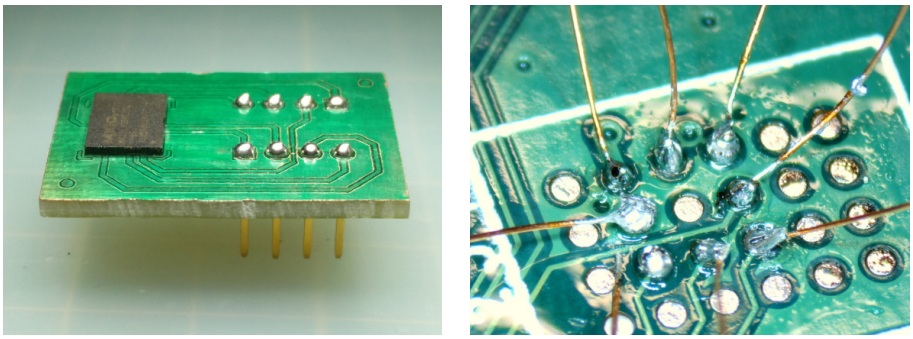
(a) XY laser table (b) PCB after etching and cleaning

Fig. 6. PCB manufacturing by laser and chemical etching.

In Figure 6, the XY table and the resulting PCB can be seen, where each BGA pad is 0.35 mm wide, and each track is 0.15 mm wide.

3.5 Finishing Adapters and Restoring Device Functionality

To finish the PCB adapters, a layer of solder mask is applied and cured with UV light to protect the copper from oxidation and the pads are tinned with solder. Then chips are soldered back on their respective adapter with the heat gun, which requires first to *reball* them manually under a microscope, i.e. to put new solder balls under the BGAs. A finished adapter board is shown in Figure 7a and can be directly used in a universal EEPROM programmer, allowing the flash memory to be read and written.



(a) Finished adapter board

(b) Microsoldering wires on the original board pads

Fig. 7. Last steps...

To be able to easily plug back a chip in place and to unplug it multiple times, DIP8 headers were added on each instance of the device under test and their pins wired to the BGAs pads of the original board, as illustrated by Figure 7b. The adapters can therefore be used as simple DIP8 chips.

4 Conclusion

We hope this short article convinced the readers that, when investigating an embedded device, security analysts can benefit from such low-cost hardware techniques. A rough estimate is about 1,300 € for the soldering tools, microscope, EEPROM programmer, CNC and consumables. Hardware attacks should no longer be considered as expensive, difficult to setup and, as such, reserved to an elite class of attackers.

Still, there are limitations: in-house PCBs are not practical for very large BGA chips requiring multi-layer PCBs with vias.