Forensic analysis of mobile phone internal memory

Svein Y. Willassen
Norwegian University of Science and Technology

Abstract

Mobile phones have become a very important tool for personal communication. It is therefore of great importance that forensic investigators have possibilities to extract evidence items from mobile phones. Modern mobile phones store evidence items on SIM-cards as well as internal memories. With the advent of modern functionality, such as camera and multimedia messaging, more and more of these items are stored in internal memory. Proper forensic examination of such memories, including recovery of deleted items, has not been possible until now.

This paper presents two different methods of physical imaging of mobile phone memory units. The methods are applied to several popular modern mobile phones, and it is shown that the methods can be utilized in practice to recover important evidence such as deleted text messages. The discovery of mobile phone internal memory management functionality challenges the current mobile phone analysis paradigm.

1.0 Introduction

It is clear that mobile phones contain information that may have value as evidence in investigations. The mobile phone has become the modern person's primary tool for personal communication, and therefore frequently contains information about a person's activities. Obtaining information on such activites is often a primary goal in an investigation. Analyzing the content of a mobile phone is therefore an invaluable tool for the forensic investigator.

This paper first examines what evidence exists on mobile phone internal memory. Two different methods for imaging phone internal memory are then discussed. Further, memory content is examined for evidence items.

2.0 Evidence in Mobile Phones

Mobile phones are digital media. In principle, this means that mobile phones have the same evidentiary possibilities as other digital media, such as hard drives. For example it is, as will be explored in this paper, possible to extract deleted information from a mobile phone, in the same way it is possible on a hard drive. However, mobile phones also suffer from the same evidentiary problems as other digital media. As with a computer, the content of a mobile phone is fragile and can easily be deleted and overwritten. Mobile phones should therefore be handled with great care and insight, just as any other digital media.

2.1 Evidence Items

A long range of evidence items can be found in modern mobile phones. A few is listed in [1]. In addition to the evidence related to the phone system itself, modern phones have other evidence items that should be mentioned:

- Images
- Sounds
- Multimedia messages
- WAP/web browser history
- Email
- Calendar items
- Contacts

Last but not least, the importance of SMS text messages should not be underestimated. The Short Message Service is very widely used, and messages are stored on the sending and receiving mobile phone. The ability to recover deleted text messages would have great value in many investigations. The investigation of possibilities to recover deleted text messages has been the main motivation for this research.

2.2 Storage Media

With the advent of digital mobile telephone systems, such as GSM, a need for local digital storage emerged. In the GSM system, the single most popular digital mobile phone system, the SIM (Subscriber Identity Module) was specified as storage medium and implemented as a smart card that fits inside the phone. The SIM contains subscriber information and secret encryption keys necessary for the communication. It also implements storage space for contacts and text messages. As described in [1] and [2], the SIM can be soundly forensically analyzed. It is also possible to recover some deleted data from the SIM (deleted text messages on from some phone models). The SIM architecture is also used in 3G systems (USIM).

However, since the late 1990s, mobile phone manufacturers began to use mobile phone internal memory in addition to the SIM for storage of information items. The SIM has a rigorous specification allowing only for certain types of information to be stored. It does not provide a single continous memory that can be utilized for any purpose. As the manufacturers wanted to implement new functions where storage on SIM was not available, mobile phones were gradually equipped with internal memory for storage of items such as missed and received calls, calendar events, text messages, contacts and other items. The first models used a serial EEPROM chip for this purpose. With the growth of memory demand, it gradually became more common to implement internal memory on flash memory, either as a flash chip dedicated for information item storage, or as an area on the flash memory chip storing the phone system software. More recently, with the advent of telephones with cameras and MP3 players, it has also become common to add possibilities for external flash memory in mobile telephones. External memory can be added by inserting a memory card such as SD, MMC, CF or similar. Sound forensic analysis of external memory cards can be accomplished by using existing computer based forensic tools such as [16], [20] and [21] and will not be further examined in this paper.

2.3 Usage of External Memory, Internal Memory and SIM

In order to understand the value of analyzing mobile phone internal memory, one must understand where the different information items in mobile phones are stored. In order to understand this properly, a range of mobile phones with SIM, internal storage, and to a certain extent external flash storage was analyzed to determine what information items are stored on the different media types. The phones were examined by sending and receiving text messages, taking pictures, store contacts and calendar events, exchanging SIM cards and external memory cards, and observe the behaviour. The results indicate that each manufacturer is consistent in the way data is handled, but the variation between manufacturers is significant.

The results are summarized in the following, grouped on each manufacturer.

2.3.1 Nokia

The models 3200, 3410, 3510i, 5110, 6110, 6150, 6210, 6230, 6310i and 6610 were analyzed.

The following behaviour was observed on the analyzed Nokia phones: Text messages are stored on the SIM. When the SIM is full (max 20-30 messages), the phone uses internal memory (up to 150 messages common on most models). Older models store only incoming messages, but newer models store both outgoing and incoming, but only incoming is stored on SIM. With Nokia phones, deleted messages on SIM can be recovered. Contacts can be stored on SIM or internal memory, and the user can select which memory to use. Older Nokia phones cannot store contacts in internal memory. Of the analyzed phones, this was the case only with the 5110.

The analyzed Nokia phones use internal memory for all other data such as calendar events, caller logs, pictures etc. The call log file on the SIM is not used by Nokia. If the SIM is changed, the phone will delete the caller logs, but all other data will remain.

Only one of the analyzed Nokia phones (the 6230) had external flash memory. This could be used as additional storage for pictures and sound files. Text messages cannot be stored on the external card. Multimedia messages will implicitly be stored in internal memory, but may be moved to external memory by the user.

2.3.2 Sony Ericsson

The models A2618s, GH688, R380s, S868, T68, T68i, T610 and T630 were analyzed.

For the analyzed Sony Ericsson phones, it was observed that text messages are stored on internal memory until it is full, and only then will the phone start to use the SIM. As a consequence, SIM cards will in most cases contain nothing when they have been used in Sony Ericsson phones. It was also discovered that the phone deletes all messages in internal memory if the user switches the SIM card. Other items, such as pictures and calendar events are all stored in internal memory. For phones with external memory cards, it is similar to Nokia only possible to copy pictures and sounds to these, and only at the user’s explicit request.

2.3.3 Siemens

The models A60, C25, C60, C62, M55 and M65 were analyzed.

The analyzed Siemens phones use SIM as primary storage for text messages and log of outgoing calls. When the SIM memory is full, internal memory is used. For contacts, the user can choose whether to use internal or SIM memory, but internal memory is the default. None of the Siemens phones deleted items when switching the SIM card. No Siemens phone with external memory was analyzed.

2.4 Discussion

From the above it should be very clear that many, in fact most, information items on modern mobile phones are stored in internal flash memory as opposed to the SIM or external flash memory. It is therefore of outmost importance to find methods to perform sound forensic analysis of mobile phone internal memory.

3.0 Mobile Phone System Architecture

In order to fully understand how forensic analysis of mobile phone systems can be conducted, it is important to understand how mobile phones are built. The system architecture of a mobile can generally be viewed as the architecture on figure 1.

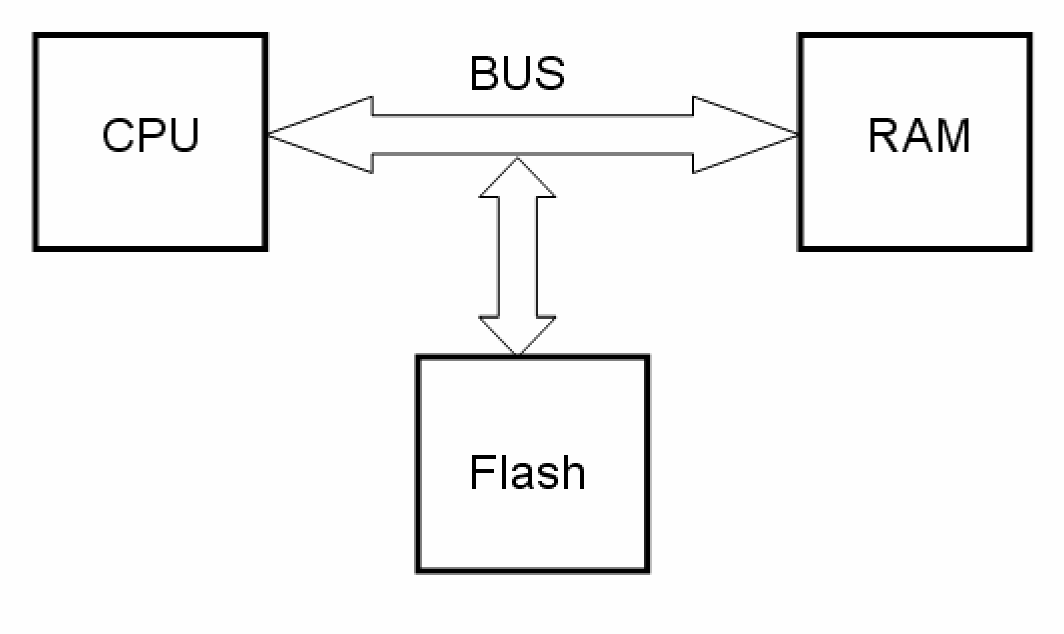


Figure 1: General mobile phone architecture

The central unit of the phone is the CPU. The CPU controls the communication circuits of the phone, in addition to control the communication with the user. For intermediary storage, the CPU uses a RAM. RAM is used for all intermediary storage during communication and user interaction. The RAM can be implemented as a separate intergrated circuit or it can be integrated with the CPU in a single integrated circuit.

The phone also needs a secondary non-volatile storage (shown as secondary storage on the figure). This is needed for storage of all data pertaining to user and communcation that needs to persist during a power failure. Secondary storage can be implemented in various ways. The most common implementation today on mobile phones is a separate flash memory integrated circuit on the system board. In addition to these elements, the CPU has communication with the SIM, and optionally other external storage media. It is also common to have a special unit to control the usage of power in a mobile phone.

One should note the similarity of the architecture of a mobile phone with that of a computer. Where computers normally use a hard drive for secondary storage, a mobile phone uses flash memory. Otherwise the mobile phone architecture is like a computer. In this essence, it is not unfair to say that the mobile phone is a computer. The difference in secondary storage is however very imporant for forensic investigators.

4.0 Current Analysis methods for internal memory

4.1 Analysis process

There is currently no standard method for analyzing mobile phone internal memory. The methods that are currently used, focus on extracting information from the phone by utilizing a cable, infrared or bluetooth connection to the phone, and then extract information by using the AT-command set that has been specified for communication with serial modems. GSM specifcations [18] specify a standard command set for the extraction of certain evidence items. Other items are specific to the different handsets and must be extracted with AT commands that are specific from manufacturer to manufacturer or even from model to model. To aid investigators with such information extraction, several software packages exist to perform this process. Cell-Seizure, TULP and Oxygen Phone Manager are examples of such software packages. [12] [13] [14]

4.2 Weakness of current methods

Extracting information with AT-commands implies obtaining information through the operating system of the phone. Under this paradigm, the investigator is only able to enumerate the content that mobile phone operating system itself will recognize. Thus; deleted information cannot be extracted with this method. This very important weakness should warrant a search for new analysis methods.

As mentioned, important evidence items will on some models become deleted if the phone is analyzed with a new SIM. However, if turned off, one will subsequently need the PIN-code for the SIM in order to analyze the phone with the original SIM. Some therefore suggest that the phone, if turned on when seized, should be kept on until analyzed. To avoid new activity on the phone, it is suggested to utilize a jammer that jams the GSM signal so that the phone cannot communicate. One could also use a Faraday cage or “Faraday bag” to prevent the phone from communicating. Allegedly, this kind of protection will prevent new information from contaminating the phone memory. As will be shown later in this paper, this is not necessarily the case. Also, it should be clear that being barred from signals is not a normal situation for a mobile phone. It will search for network channels, and consume more energy than normal, thereby potentially exhausting the battery. It has also been suggested to analyze the phone with a “cloned” SIM-card, where the PIN-code has been removed. Although more appealing, this method still involves turning the phone on and reading memory through the operating system.

Indeed, as long as the phone is turned on, changes can potentially occur that will overwrite any deleted information resident on the phone. This is the case with mobile phones as with any other digital evidence. In the search for possible new methods, one should therefore look for methods that allow forensic analysis of a “dead” mobile phone. With such a paradigm, mobile phones can be shut off as soon as they have been seized, and investigators can be sure that they do not introduce any changes to the system themselves.

To explore possibilities for imaging and analysis of internal memory, research has been conducted for two possible methods; the use of desoldering techniques to desolder and image a memory integrated circuit, and the use of built-in test methodology to image memory contents. In the following chapters, these two different methods are explored, together with a review of analysis of the resulting image. One could also imagine a third method; loading software into the system RAM that allows reading of flash memory through the system interface. This method would require software specific to each phone and has not been further explored.

5.0 Forensic Desoldering

The first proposed method for imaging of mobile internal memory is by desoldering the memory circuit.

Since content stored inside the mobile phone resides on an on-board flash memory chip, the seemingly obvious method to access the content is to attach to the chip itself and read off the contents. Although seemingly a simple and elegant solution, the approach has some challenges that need to be addressed.

5.1 Ball Grid Array Technology

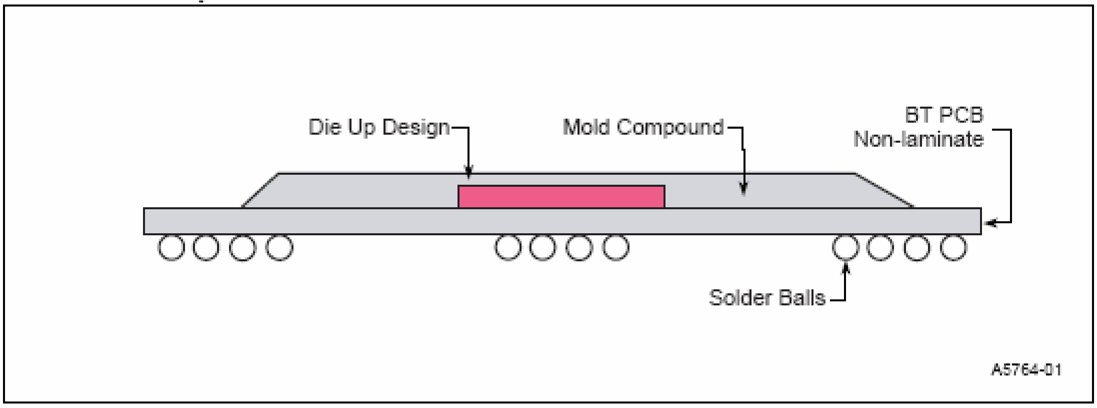


Figure 2: BGA Technology

Ball Grid Array Technology (BGA) is a mounting technology for the mounting of SMT (Surface Mount Technology) components on a PCB. Instead of the conventional usage of small leads leading from the side of the package soldered to pads on the PCB, the chip is manufactured with an array of pads on the bottom. Small balls of solder are placed on these pads. When mounting to the PCB, the chip is placed on top of matching pads on the PCB and soldered with Reflow soldering. Reflow soldering means heating the unit to the melting point (reflow temperature) of the solder, so that the melted balls bond to the pads on the PCB. From the electronics manufacturer’s perspective, the use of BGA technology has several advantages. The foremost advantage is that BGA devices use less space on the PCB. The typical space constraint on a conventional integrated circuit is not the limited area of the chip itself, but the limited area for pins on the sides. With BGA, the full area of the bottom of the chip can be utilized to get the signal out.

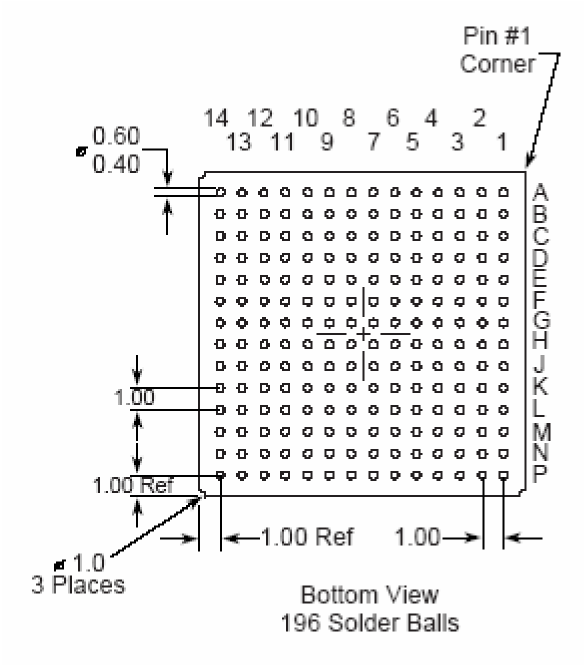


Figure 3: Example BGA ball configuration

From the perspective of the forensic investigator however, the use of BGA technology presents some difficulties. With conventional surface mounting, it would be possible to attach probes to the pins and read and analyze the chip without desoldering it from the PCB. With such analysis it would be possible to analyze the content of the memory without destroying the unit. This is not possible with BGA circuits. Since the chip is bonded directly to the PCB through molten solder balls, it is not possible to attach individual probes to each connection. The chip must therefore be soldered off before it can be read.

Desoldering a chip presents new challenges to the forensic investigator. Desoldering must be performed with great care to avoid damaging the memory circuit. After desoldering, the solders balls will be melted and most likely short circuit the pads of the chip. Solder residue must therefore be removed, and the chip must either be restored to its original state by using a re-balling process, or the contents must be read off by connecting directly to the pads.

5.2 BGA Reballing

Many device readers require that the chip package is intact with balls in place. It may therefore be required to reball the circuit before the contents can be read. Reballing is the process of restoring the solder balls correctly placed on the pads on the lower side of the BGA package. The most common method to achieve this is to use a stencil that has been specificly manufactured to reball a certain package. The stencil has small holes that match the placement and size of the pads on the package. When reballing, the stencil is placed on top of the lower side of the package and aligned, so that each hole matches a pad. Solder paste is then applied on the stencil and evenly distributed using a squeegee until all holes are properly filled. Reflow is then applied, and the solder paste in the holes will form balls that are fixed to the pads on the chip.

BGA reballing using this method requires stencils that match the chip in question. In particular, ball size, inter-ball distance (“pitch”) and ball configuration of the package must be matched by the stencil. Since there exist a long range of different packages with various sizes and configurations, obtaining the correct stencil may be a practical difficulty.

5.3 Reading memory in a device programmer

For the purpose of reading memory circuits, such as EPROM, EEPROM, flash, volatile memories and microcontrollers with embedded memory, one can obtain a device programmer. A device programmer is an electronic device specially made for the reading and programming of memory circuits and microcontrollers with embedded memory. The device attaches to a computer and (most often) allows reading and programming a wide range of memory circuits through the use of adapters. An adapter can be general (such as an adapter for all DIP or SOIC packages with up to 40 leads) or specific (such as an adapter for the Intel 28F640). In addition to the physical adapter, the correct software for reading a specific chip is needed, since the pin configuration can vary greatly between packages with the same physical layout. Manufacturers of device programmers usually supply software with their programmers to read and program of a wide range of devices. Many devices have built-in checksum capabilities that allow for the detection of inconsistencies when reading a device. Checksum capability must be supported in the programming software.

In the application of forensic analysis of mobile phone memory, the device programmer must be used for reading BGA circuits. In this case it is necessary with very specific adapters for each circuit type. The adapter must have pins that match the ball size, intra-ball distance (“pitch”) and ball layout (for example 4x10 or 8x8). In addition, since mounting a chip in the adapter requires mounting the chip in a tight slot to ensure alignment, the adapter must match with the physical size of the package.

Adapters for BGA circuits can be made in several different ways. Two common technologies for the adapter-chip connection are Y-shaped springs or spring-loaded pogo-pins. A Y-shaped pin needs to have a ball in place in order to establish an electric connection. The pogo-pin however, can be used directly on the pads on the chip without balls. Therefore, from a forensic point of view, an adapter with spring-loaded pogo-pins would be a better choice, since device reballing can be avoided. Availability of the two types for different devices seems to vary.

5.4 Desoldering – background and practical possibilites

There exist a long range of different equipment types for performing soldering and desoldering. These equipments are usually based on heating the device and solder through heat conduction, convection or radiation. Equipments range from the simple soldering iron heating a device by conducting heat through its tip to massive reflow ovens for use in production equipment.

The most important for the forensic investigator is to consider the possibilites to damage the unit. Any integrated circuit can not endure more than a certain number of heatings to reflow temperature. Intel states that BGA circuits manufactured by them are guaranteed to endure reflow three times. [4] Since a PCB with SMT components on both sides will have to be assembled with two reflows, there is only one left for the forensic investigator. This is not enough if both desoldering and reballing is needed. Experience however shows that most components will handle many more reflows if handled correctly.

The most important factors to avoid damage are correct maximum temperature and correct temperature gradient. Since it in contrast to conventional mounting technology is not possible to heat only the lead, the whole chip has to be heated. The maximum temperature should be only nominally above the reflow temperature of the solder. The gradient should be so that the temperature rises slow enough to avoid damage. According to Lee, the temperature should have a “tent” profile, with peak temperature around 210 C, heating to this temperature in 3-4 minutes. [5] It is also recommended to pre-bake the unit to remove moisture that has been soaked in the unit. Heating to reflow temperature could otherwise cause a “popcorn-effect”, whereby water vapor cause cracks in the circuit substrate and permanently damage the circuit. This could be extra important for circuits in mobile phones, since these units must be assumed to have lived in conditions of high moisture. Figures for moisture sensivity and conditioning requirement are given in [7].

These requirements rule out any use of hand soldering tools. Forensic desoldering should only be performed by using an automatic soldering station with possibilities for temperature gradient programming and automatic component removal. The most common equipment for professional desoldering is a soldering station with a combination of convection heating through hot air and radiation heating through an infrared light source. The station has a temperature controlled hot air nozzle, and can be programmed to follow a specific heating curve, allowing the operator to program the unit to correctly specify the gradient and the maximum temperature.

5.5 Dismantling and identification

To test the methodology and identify these challenges, several different phone models were dismantled, desoldered and read. The results will be summarized in the following.

Before desoldering, the units were dismantled. Most mobile phones can easily be dismantled when proper tools and care is used. A set of torx screwdrivers is needed in most cases. All dismantling should be done in an electrostatically safe environment since the exposed components are sensitive. In some cases, it is also necessary with special tools from the manufacturer, in which case such tools should be obtained, or a repair business consulted. It is also possible to obtain dismantling instructions for various models by contacting the manufacturer.

After dismantling the unit, one or several printed circuit boards (PCB) are exposed. It was found that most modern phones for reasons of space constraints implement the design on a single circuit board with surface mounted components on both sides. The circuit board is layered with several inner layers that are not exposed. It is therefore not possible to follow the leads on the design by examining the board directly. Among the surface mounted devices there are a number of integrated circuits. The boards that were examined had 5-10 surface mounted integrated circuits, of which most utilize Ball Grid Array Technology (see table 1). These can be identified by examining identification text printed on each circuit. By carefully noting the written text and comparing with datasheets available from databooks and IC manufacturer websites, integrated circuits corresponding to each element in the system architecture were identified.

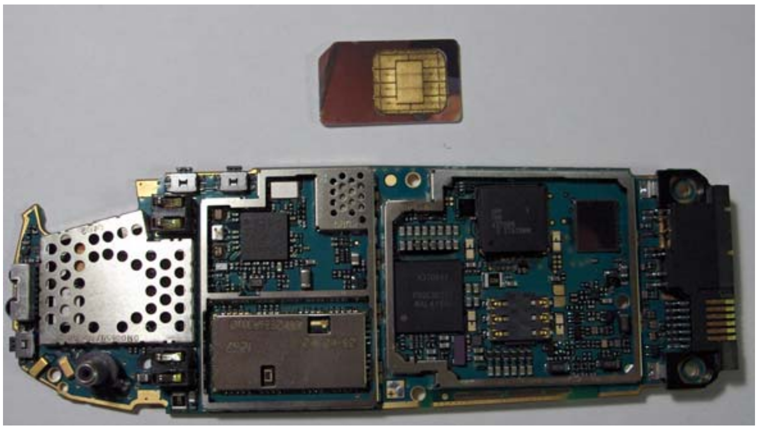


Figure 4: Nokia 6310i main board

For each model, a Central Processing Unit (CPU), Main memory (RAM) and storage memory (flash) was identified. Table shows a summary of the models that were examined in the experiment and identification of the on board flash memory integrated circuit.

Model	***Chip***	***Chip technology***
Nokia 5110	*Sharp L28F800BE-TL85*	*TSOP*
Nokia 6310i	*Intel 28F320B3TA*	*BGA*
Nokia 6610	*Intel 128W18T*	*BGA*
Sony Ericsson T68i	*Intel 28F640W18T*	*BGA*

Table 1: Memory Technologies used

5.6 The desoldering experiments

Having identified the flash memory circuits, the circuits could now be desoldered. The units were pre-baked for 24 hours at 80 C to dry out moisture. The chips were thereafter desoldered using a hot air soldering station with IR preheating. A temperature profile was constructed with a maximum temperature of 220 C, with a gradient such that the temperature was reached in 5 minutes. The chips were removed from the PCB at the maximum temperature with a robot arm mounted on the soldering station using vacuum to remove the hot circuit from the board.

It could be possible to remove the chip with the solder balls intact, by using a vacuum sucker at a temperature where the solder ball is just marginally above the melting point at 183 C. This would in most cases result in the balls being intact due to the surface tension of the molten solder. This practice would however be risky since it is difficult to know the exact temperature under the circuit. Trying to remove the circuit at too low a temperature could possibly rip off the pads and damage the circuit beyond repair. Due to this risk it was decided to use the temperature of 220 C which should be sufficiently above the melting point.

After the removal of the circuits from the board, the undersides of the circuits were exposed. The pads of the chips were exposed, but partially covered with solder residue, from the molten solder balls. The residue covered several pads and short circuited them. Several methods could be utilized to remove the residue. In the experiments, the residue was removed by using a temperature controlled hot air soldering station and soldering wick, a method that proved to be efficient. It is important to understand that also the process of removing solder residue implies heating the unit, and should be performed with the same care as the removal of the circuit itself. A reballing process was now applied for those of the packages where stencils were available. (See table 2)

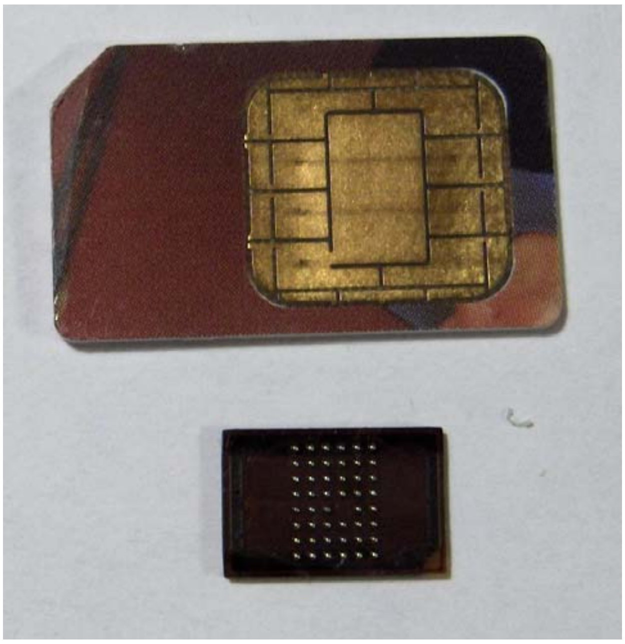


Figure 5: Successfully reballed 8x6 7x11 mm BGA package with 0.75 mm pitch (smart card shown for size comparison)

5.7 Reading in a device programmer

The devices from the phones were mounted in a commercially available device programmer, and content were read using the supplied software. For some of the packages, an adapter was not commercially available at the time. For others, check of the built in checksum resulted in an error. For the examined devices, it was not possible to determine where in the memory dump the error occurred. Thus, it was not possible to determine if the error occured due to damage caused by the desoldering-reballing process or other cause.

Model	**Chip**	**Adapter**	**Read Successful**
Nokia 5110	Sharp L28F800BE-TL85	Yes	Yes
Nokia 6310i	Intel 28F320B3TA	Yes*	Yes
Nokia 6610	Intel 128W18T	No	N/A
Sony Ericsson T68i	Intel 28F640W18T	Yes	Yes

* Modification of the 28F320B3TA adapter was necessary to read the chip

Table 2: Reading desoldered memory packages

The read operation resulted in a computer file of the same size as the chip, containing a dump of all the memory. These files could be used for forensic analysis of the memory contents.

6.0 Using embedded test technology for memory extraction

Most embedded systems are built with embedded test technology. The reason is that manufacturing of electronic devices is a complex process where many things can go wrong. Unless the manufacturer has an automated way of testing the functionality of the device, elements and interconnections, the quality of the resulting product can not be guaranteed.

The current trend in test technology is called “boundary-scan”. Using this technique, each integrated component has a shift register which can be used to probe and set the pins of the components. The shift register can be loaded serially from a test access port accessible as test points on the device. If the CPU of a device has support for boundary-scan, the bus can be controlled from the test access port, and consequently it is possible to read or program the memory attached to the bus. This is called technique is called “in-system programming” and is frequently used in the manufacturing process.

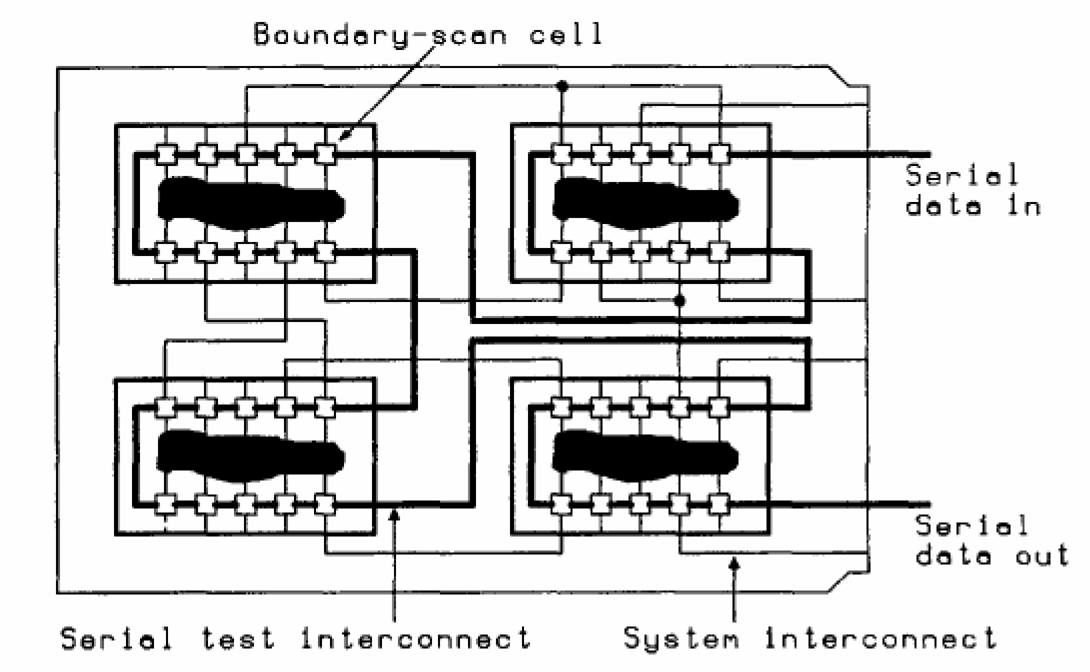


Figure 6: JTAG system overview

The Joint Test Action Group (JTAG) has standardized a procedure for implementing boundary-scan in order to facilitate interoperability between components from different manufacturers. The standard is published as IEEE 1149.1 [8], and is frequently called the JTAG-standard.

6.1 The IEEE 1149.1 standard

The JTAG-standard defines a Test Access Port (TAP) controlling implementation of test functionality in integrated circuits. 7 new signals are defined (see table 3) that has to be provided as external pins on compliant circuits.

Signal	Description

TCK	Test Clock Input
TMS	Test Mode Select Input
TDI	Test Data Input
TDO	Test Data Output
TRST	Test Reset Input (not compulsory)

Table 3: JTAG signals

The test system works by letting the operator clock in data into two shift registers; the instruction- and data register of the TAP controller. Instructions can now be given in order to read or set the values of the boundary scan register. Since each bit in the boundary scan register is coupled to external pins of the circuit, this can be used to set and read values to/from the system bus.

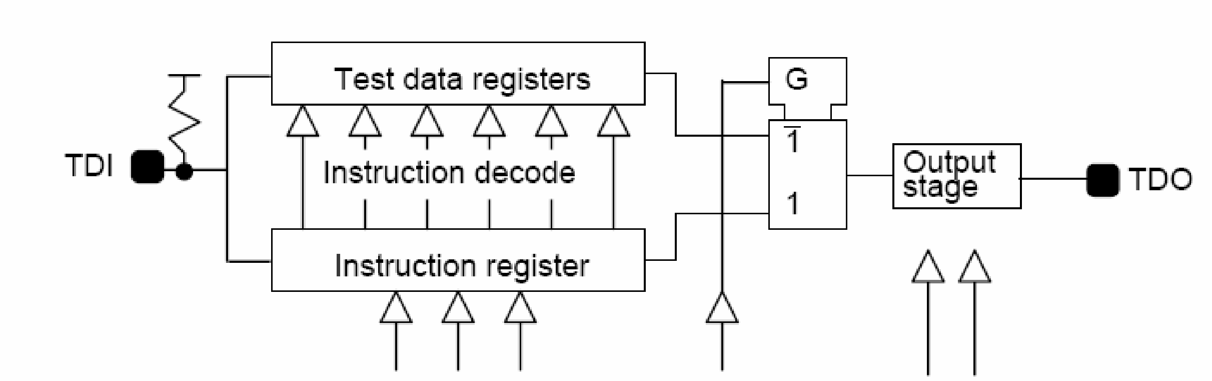


Figure 7: JTAG Test Access Port

On a printed circuit board, the layout can be made specifically to allow for using JTAG as test standard. The manufacturer can lay out test points that directly access the JTAG pins of each integrated circuit. More conveniently, the JTAG pins of each integrated circuit can be interconnected, so that only one set of test points is required. This can be done in various configurations, one of which is serial as shown in figure 8.

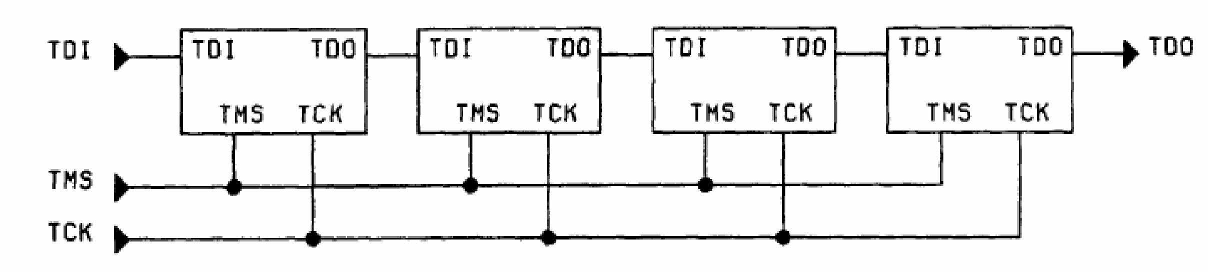


Figure 8: JTAG serial system board interconnection

6.2 In-System Programming with JTAG

In system programming (or reading) can be accomplished with JTAG. If the memory device supports JTAG itself, the TAP of the memory device itself can be addressed through the JTAG interface. For each memory address, the in-system programmers sets the correct address value on the memory device’s address bus by shifting in the proper bits through the boundary scan register. The data bus will now contain the content of that memory address and can be read by shifting the value of the boundary scan register back through the JTAG chain. By probing all memory addresses in this fashion, the entire contents of the memory can be read.

If the memory device itself does not support JTAG, in-system programming can be performed by using the TAP of another device connected to the memory system bus. This device may for instance be the system processor, which almost always will have a direct connection to the memory via the system bus. Reading the memory can be done in a similar fashion as with the memory device TAP by manipulating the boundary scan register of the CPU, setting an address on the bus, and reading back the resulting data through the CPU TAP.

6.3 Challenges with forensic memory reading through JTAG

It is important to understand that JTAG is not a bus standard, but rather specifications standardizing a Test Access Port on each component, allowing interconnection between components on a board. It is entirely up to the manufacturer of each integrated circuit to decide on the configuration and working of the pins on the chip, and the configuration of the boundary scan register. Moreover, it is up to the printed circuit board designer to decide if JTAG ports shall be interconnected, or at all accessible from test points on the board. The designer can decide to not use JTAG at all, leaving the test pins on integrated circuits unconnected. With BGA circuits in such designs, attaching a probe to the test pins is impossible and memory reading through JTAG is not an option. It is however common to implement JTAG in designs with BGA circuits, since the manufacturer otherwise would have no way to test the design.

Before trying to use JTAG to read memory, the following must be considered:

- One must know which system processor and memory circuits are used and how they are connected on a system bus. This is required since it otherwise would be impossible to find the right bits in the boundary scan register.
- One must find test points for the JTAG on the printed circuit board and determine which test point is which signal.
- One must know the protocol for memory reading/writing.
- One must determine the correct voltage. Using too high voltage may damage the circuits.

The voltage can in most cases be determined by measuring on a live board. The memory protocol is in most cases available by downloading information about a memory circuit from the manufacturer’s website. The two first considerations however may be a major challenge, since it in practice is very difficult and tedious to perform this task without complete system documentation including schematics. When it comes to mobile phones, such documentation is in most cases not available. In any case, the implementation of a system for memory reading via JTAG will be different from phone to phone. Even within one specific model, small configuration changes (such as using a different memory circuit) may require another implementation of JTAG memory reading.

6.4 JTAG memory reading experiment

In order to explore the possibility to read memory with JTAG, experiments were conducted. Nokia 5110 was chosen as a test model for the experiment. This model was chosen from the fact that the service manual for this model including schematics was available. Since a large number of this rather old model exist in the market, it was easy to obtain a number of test models without consuming too much resources. Nokia 5110 only stores a few evidence items on the phone itself. It is however believed that a successful read of 5110 memory can be extended to other models.

Starting the experiment, the service manual [10] was examined for signs of JTAG implementation. Indeed, the CPU, listed as “MAD2”, has pinouts for “JTRst”, “JTClk”, “JTDI”, “JTMS” and “JTDO”. These pins are coupled in a line “JTAGEMU” onto a connector which in the schematics is listed as “not assembled”. A 5110 was disabled however, and a set of test points corresponding to this connector on the system board was found. The connections between the test points and the CPU were in part visible (see figure 9) allowing easy identification of the test points.

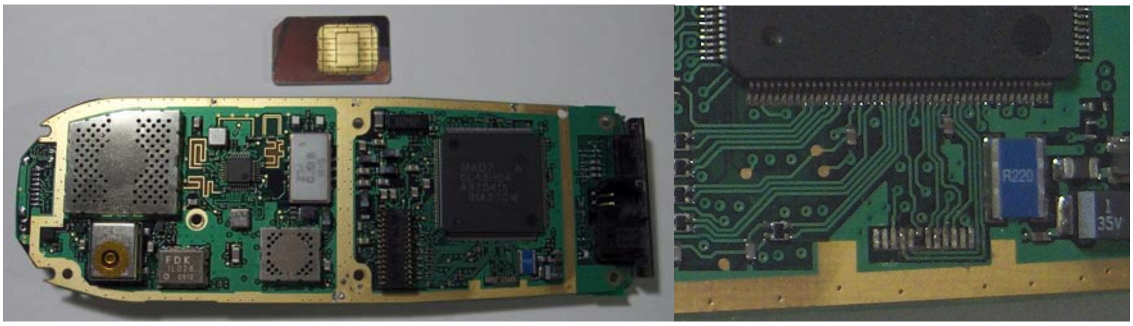


Figure 9: 5110 system board with JTAG test points.

Measurements of the voltage on these test points indicated that they were in fact connected to the JTAG interface of the CPU. Test wires were carefully soldered to the connectors using very thin wires extracted from an 80 lead ATA cable and soldering paste applied from an applicator gun. Since the test points are very small, the soldering was found to be fairly difficult. The key to success is to use soldering paste and sufficiently thin wires. Unlike desoldering of BGA chips, the risk of damaging the evidence medium itself is not very high in this soldering process. It can however be tedious to accomplish proper connections without any short circuits.

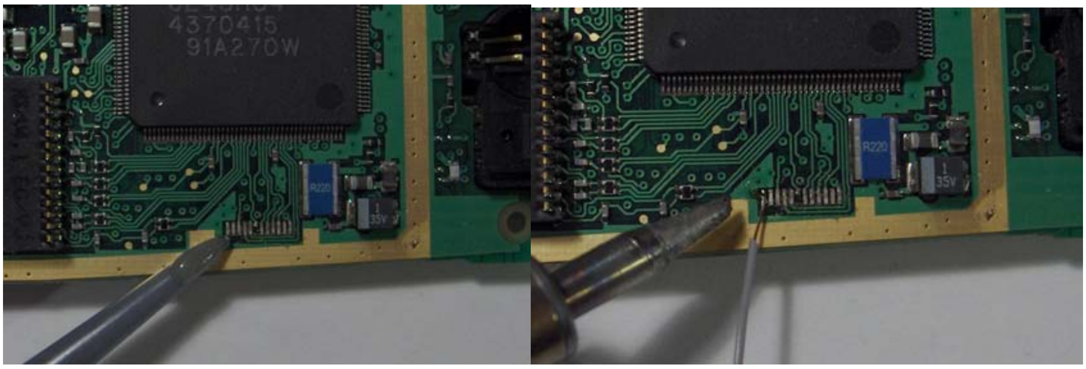


Figure 10: Applying soldering paste and soldering thin wires to the JTAG test points.

By connecting the wires to a breadboard, the 5110 could now be connected to a computer through the JTAG interface. JTAG interfaces can easily be built, but commercial solutions are also available. For this experiment, the “Chameleon POD” programmable JTAG interface [15] was used. When connecting the phone to the interface, the voltage level of the phone system board and the interface should be considered. For the 5110, the system board uses 2,7V technology, being within the handling range of the Chameleon POD. Newer phones use 1.8V technology. At this voltage level, a level shifter would be required. Connecting a 1.8V system board directly to a JTAG adapter of higher voltage could damage the processor.

Now, the JTAG interface was connected to a computer running Linux with the open source package JTAG-Tools installed. This package allows for connection with a JTAG device through a number of different adapters, including several that the Chameleon can be programmed for. The package supports various processors and configurations and with a modular architecture, it should be easy to extend to allow for extensions to support more processors and memories.

After a few changes, a connection with the 5110 processor TAP was established. Although it is known from the service manual that the processor design is based on ARM7, and documentation on the JTAG interface for this processor design is publicly available, it was decided to proceed treating the TAP as a “black box”, as this will be the case in most situations. JTAG-tools allows for black-box analysis through the use of the command “discovery”. This function will cycle a “1” through the JTAG-chain and detect the number of available TAPs and the length of their instruction register (IR). For the 5110, the IR length was determined to be 12 bits, with only one JTAG TAP connected (the processor). The software then continues the analysis by probing all possible values of the IR with cycling of a 1 through the JTAG chain and thereby detecting the data register length for each instruction. Thus, it is possible to determine the values for different instructions. This process proved to be rather time consuming, as 4096 different possibilites were tested, but eventually resulted in the discovery of commands that can be used to set and probe the boundary scan register. The command probing itself will not change the content of the boundary-scan register and can therefore be executed without risk for altering memory content.

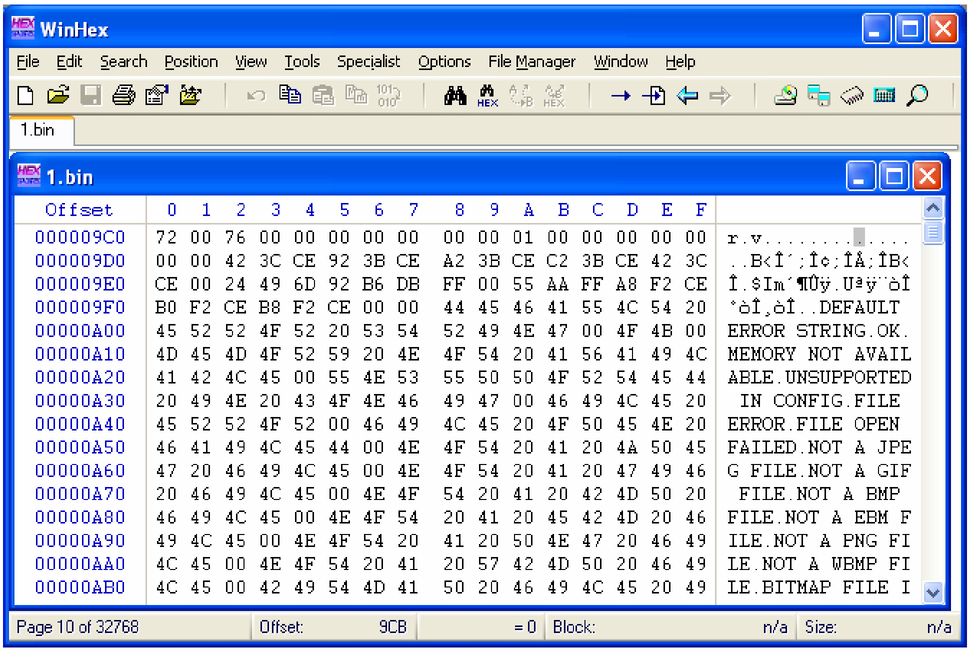
Through detailed examination of the schematics, it was found which bits in the boundary scan register to set/probe for reading the on board Sharp L28F800BE-TL85 flash memory. These details were implemented in definition files for JTAG-Tools, and the memory could be read with the command “readmem”.

7.0 Memory Content analysis

After reading the memory using desoldering or JTAG analysis, the contents of the flash memory is available as a binary file, and can be analyzed. A special tool for this purpose is for the time being not available. The binary file may however be analyzed with a standard Hex-editor, such as Win-Hex. [16]

An initial analysis of the Sony-Ericsson T68i memory dump was conducted. The dump is a raw dump of the 8 Mb of data stored in the Intel 28F640W18T. At first glance, the memory dump seems to contain useless information. Further analysis reveals that useful readable information is available, such as error messages probably belonging to the operating system. (See figure 11) This confirms that the chip has been read successfully and that the content is readable in clear.

Further examination shows that the first 4-5 Mb of the dump seems to contain binary data of high entropy, corresponding to what one would expect system software to look like. The remaining dump contains large areas filled with byte-value 255 (hex FF) and some areas filled with what seems to be data. These could be user-data.

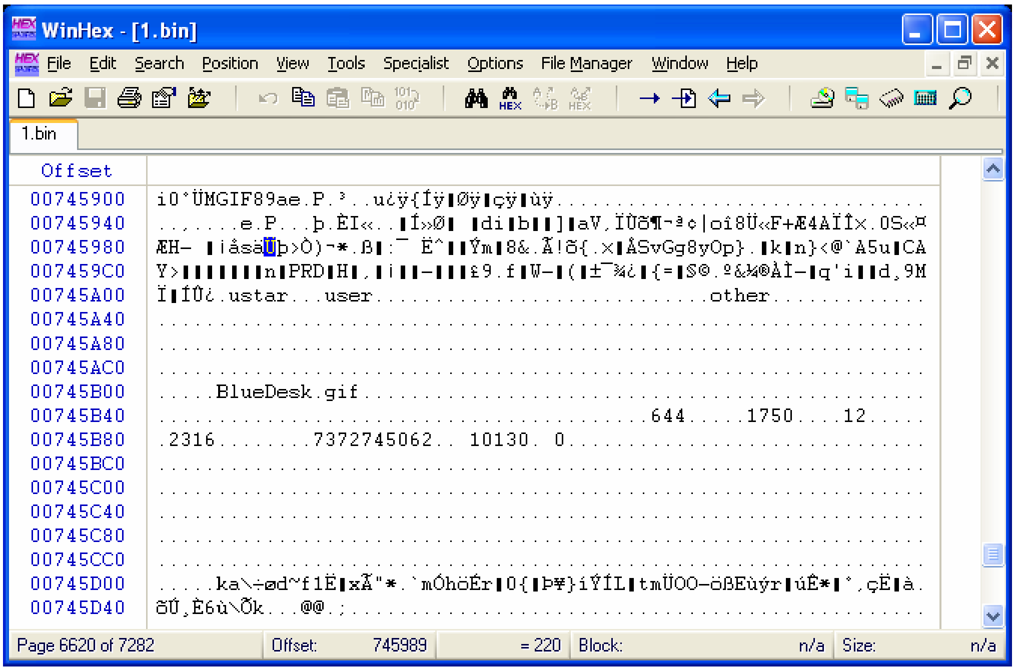


```
  Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
000009C0    72 00 76 00 00 00 00 00  00 00 01 00 00 00 00 00   r.v.............
000009D0    00 00 42 3C CE 92 3B CE  A2 3B CE C2 3B CE 42 3C   ..B<Î´;Î¢;ÎÂ;ÎB<
000009E0    CE 00 24 49 6D 92 B6 DB  FF 00 55 AA FF A8 F2 CE   Î.$Im´¶Ûÿ.Uªÿ¨òÎ
000009F0    B0 F2 CE B8 F2 CE 00 00  44 45 46 41 55 4C 54 20   °òÎ¸òÎ..DEFAULT 
00000A00    45 52 52 4F 52 20 53 54  52 49 4E 47 00 4F 4B 00   ERROR STRING.OK.
00000A10    4D 45 4D 4F 52 59 20 4E  4F 54 20 41 56 41 49 4C   MEMORY NOT AVAIL
00000A20    41 42 4C 45 00 55 4E 53  55 50 50 4F 52 54 45 44   ABLE.UNSUPPORTED
00000A30    20 49 4E 20 43 4F 4E 46  49 47 00 46 49 4C 45 20    IN CONFIG.FILE 
00000A40    45 52 52 4F 52 00 46 49  4C 45 20 4F 50 45 4E 20   ERROR.FILE OPEN 
00000A50    46 41 49 4C 45 44 00 4E  4F 54 20 41 20 4A 50 45   FAILED.NOT A JPE
00000A60    47 20 46 49 4C 45 00 4E  4F 54 20 41 20 47 49 46   G FILE.NOT A GIF
00000A70    20 46 49 4C 45 00 4E 4F  54 20 41 20 42 4D 50 20    FILE.NOT A BMP 
00000A80    46 49 4C 45 00 4E 4F 54  20 41 20 45 42 4D 20 46   FILE.NOT A EBM F
00000A90    49 4C 45 00 4E 4F 54 20  41 20 50 4E 47 20 46 49   ILE.NOT A PNG FI
00000AA0    4C 45 00 4E 4F 54 20 41  20 57 42 4D 50 20 46 49   LE.NOT A WBMP FI
00000AB0    4C 45 00 42 49 54 4D 41  50 20 46 49 4C 45 20 49   LE.BITMAP FILE I

Page 10 of 32768      Offset:      9CB      = 0   Block:      n/a   Size:      n/a
```

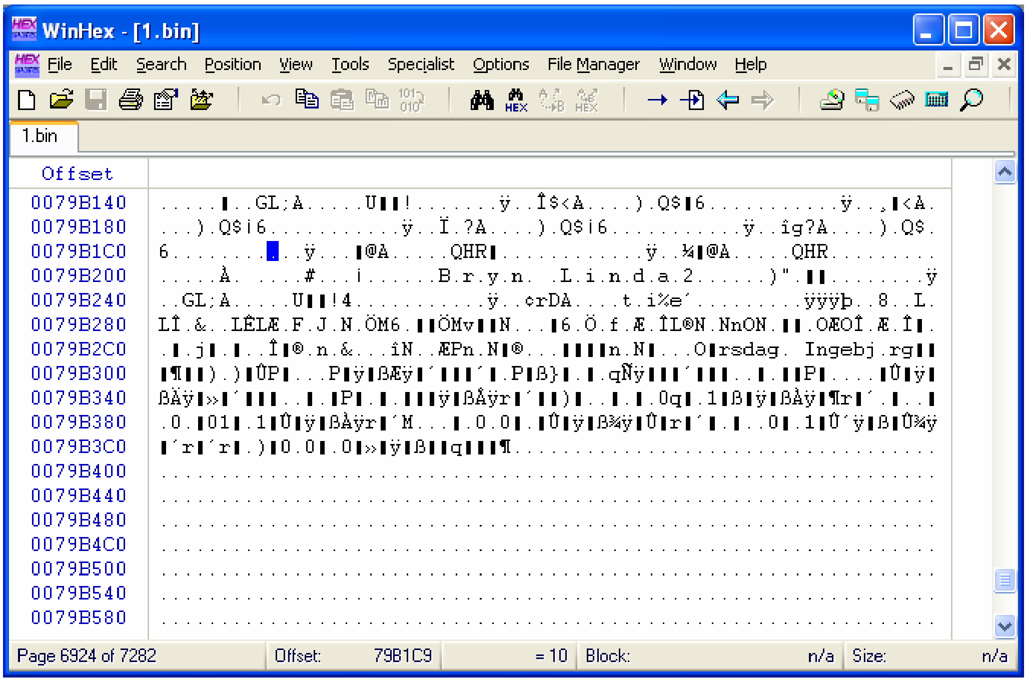
Figure 11: Hex-print from T68i memory dump

The mentioned data areas were thoroughly examined. Through this analysis, interesting data such as gif-images, jpg-images, phone numbers, calendar-items and text messages in the TPDU format [17] were found. Some of the content was recognized as content present and available in the phone operating system. Figure 12 shows the existence of a GIF-picture in the memory dump. It was confirmed that this and many other pictures could be extracted and viewed on a computer. Figure 13 shows items that were recognized as items that were known to be stored on the phone. A stored contact item “Bryn Linda”, and a stored calendar item “Torsdag Ingebjørg” was identified. It is believed that the other information assosiated with these items (phone number, date/time et.c.) can be extracted by interpreting the binary data in the dump. The exact format of the items needs to be understood in order to do this.


WinHex - [1.bin]
File Edit Search Position View Tools Specialist Options File Manager Window Help
HEX
1.bin

```
Offset
00745900  i0˚ÜMGIF89ae.P.³..u¿ÿ{Îÿ▮Øÿ▮çÿ▮ùÿ.............................
00745940  .,.....e.P...þ.ÈI«..▮Î»Ø▮ ▮di▮b▮▮]▮aV,ÏÜõ¶¬ª¢|oí8Ü«F+Æ4AÏÎx.0S«¤
00745980  ÆH- ▮iåsäÜþ>Ò)¬*.ß▮:¯ Ë^▮▮Ým▮8&.Ã!õ{.×▮ÅSvGg8yOp}.▮k▮n}<@`A5u▮CA
007459C0  Y>▮▮▮▮▮▮▮n▮PRD▮H▮,▮¡▮▮–▮▮▮£9.f▮W–▮(▮±¯¾¿▮{=▮S©.º&¼®ÀÎ–▮q'i▮▮d¸9M
00745A00  Ï▮ÎÜ¿.ustar...user..............................other............
00745A40  ................................................................
00745A80  ................................................................
00745AC0  ................................................................
00745B00  .....BlueDesk.gif...............................................
00745B40  .........................................644....1750....12.....
00745B80  .2316.......7372745062.. 10130. 0...............................
00745BC0  ................................................................
00745C00  ................................................................
00745C40  ................................................................
00745C80  ................................................................
00745CC0  ................................................................
00745D00  .....ka\÷ød~f1Ë▮xÃ"*.`mÓhöÉr▮0{▮Þ¥}iÝÎL▮tmÜOO–ößEùýr▮úÊ*▮˚,çË▮à.
00745D40  õÛ¸È6ù\Õk...@@.;...............................................
Page 6620 of 7282   Offset: 745989   = 220   Block: n/a   Size: n/a
```

Figure 12: Content identified as a GIF-picture


WinHex - [1.bin]
File Edit Search Position View Tools Specialist Options File Manager Window Help
HEX
1.bin

```
Offset
0079B140  .....▮..GL;A.....U▮▮!.......ÿ..Î$<A....).Q$▮6...........ÿ..,▮<A.
0079B180  ...).Q$¡6............ÿ..Ï.?A....).Q$¡6............ÿ..ìg?A....).Q$.
0079B1C0  6.........ÿ...▮@A.....QHR▮............ÿ..¼▮@A.....QHR.........
0079B200  .....À.    ....#...¡......B.r.y.n.  .L.i.n.d.a.2......)".▮▮.........ÿ
0079B240  ..GL;A.....U▮▮!4............ÿ..¢rDA....t.i%e´.........ÿÿÿþ..8..L.
0079B280  LÎ.&..LÊLÆ.F.J.N.ÖM6.▮▮ÖMv▮▮N...▮6.Ö.f.Æ.ÎL®N.NnON.▮▮.OÆOÎ.Æ.Î▮.
0079B2C0  .▮.j▮.▮..Î▮®.n.&...íN..ÆPn.N▮®...▮▮▮▮n.N▮...O▮rsdag. Ingebj.rg▮▮
0079B300  ▮¶▮▮).)▮ÛP▮...P▮ÿ▮ßÆÿ▮´▮▮▮´▮.P▮ß}▮.▮.qÑÿ▮▮▮´▮▮▮..▮.▮▮P▮....▮Û▮ÿ▮
0079B340  ßÀÿ▮»▮´▮▮▮..▮.▮P▮.▮.▮▮▮ÿ▮ßÅÿr▮´▮▮)▮..▮.▮.0q▮.1▮ß▮ÿ▮ßÀÿ▮¶r▮´.▮..▮
0079B380  .0.▮01▮.1▮Û▮ÿ▮ßÀÿr▮´M...▮.0.0▮.▮Û▮ÿ▮ß¾ÿ▮Û▮r▮´▮.▮..0▮.1▮Û´ÿ▮ß▮Û¾ÿ
0079B3C0  ▮´r▮´r▮.)▮0.0▮.0▮»▮ÿ▮ß▮▮q▮▮▮¶...............................
0079B400  ................................................................
0079B440  ................................................................
0079B480  ................................................................
0079B4C0  ................................................................
0079B500  ................................................................
0079B540  ................................................................
0079B580  ................................................................
Page 6924 of 7282   Offset: 79B1C9   = 10   Block: n/a   Size: n/a
```

Figure 13: Content identified as a contact and a calendar item

Analysis of the same type was conducted with all the other memory dumps available, with similar result. It is however clear that further work should be done in order to identify evidence items in memory dumps and present them clear and without doubt. Since memory content will vary from phone model to phone model, this is a research that will have to be on going as new models emerge on the market.

7.1 Memory Content Experiments

The focus of this research has been to identify if deleted items can be recovered by analyzing memory dumps, and in particular if deleted text messages can be recovered. Several experiments were done to find out if deleted text messages can be recovered. The experiments were conducted by successively reading out internal memory after commiting changes to the system using the phone operating system. The limitations of the memory reading methods made it difficult to do such experiments on more than a few tests. Another difficulty was the lack of a software tool to do identification and interpretation of TPDU messages inside memory dumps. The implementation of such a tool was beyond the scope of this project.

Several interesting observations were done:

- Text messages were still in flash memory after they had been deleted in the operating system.
- It was found that images, MMS, calendar items and contacts were also in memory after they had been deleted. Contrary from what is possible on SIM-cards it is therefore possible to recover deleted items of this type from internal memory.
- Copies of items from previously used SIM-cards were found in some occasions.

During the analysis of successice memory dumps from the same device, another very important property was discovered. During the search for deleted text messages it was found that memory areas found in one memory dump was moved to quite another area on the following dump. This is most likely due to the existence of a Memory Manager that dynamically reallocates memory during phone usage in order to ensure optimal memory organization and utilization at all times during phone usage. Although not particularly surprising, the existence of such memory managers do have ramifications for the forensic handling of mobile phones.

7.2 Implications for the current analysis paradigm

As previously mentioned, current handling of mobile phones often involves keeping the phone on for a period of time after it has been seized. Phone content is then analyzed by connecting the phone to a computer using a cable and read off the content through the operating system. The phone must be turned on during the analysis, and some therefore recommend to keep the phone on at all times after seizure.

In the light of the discovery of possibilities to recover deleted items, and existence of memory managers for memory remapping, it is clear that the current practice for analysis of mobile phone internal memory is not ideal. By keeping the operating system operational and using the operating system itself for obtaining the memory content, the investigator keeps the memory manager functional, allowing for reorganization of memory potentially overwriting important evidence at any time. The overwriting of evidence items will make the recovery of deleted items that may have existed in memory at the time of seizure with the methods described in this paper impossible. Most likely, it will also rule out any future analysis with new and better analysis methods. It has not been shown that it is possible in practice to recover overwritten data from flash memory, nor is it likely that this will be a possibility any time soon.

The situation is in many ways analogous to the development that has taken place in the development of practice for forensic analysis of computers. In the beginning, forensic analysis of computers was conducted by booting the operating system and looking for evidence.

Today, it is widely recognized that such an approach can be destructive to the evidence, and the method of imaging the hard drive before analysis is widely accepted as standard practice. This development will also have to take place for mobile phones. This is especially the case considering the small size of mobile phone flash memories when compared to computer hard drives, increasing the risk for the overwriting of important evidence by keeping the phone memory manager functional.

8.0 Results

It has been shown that a logical analysis of a memory dump can be conducted in order to enumerate evidence items present on the phone, including items that have been deleted. The existence of embedded memory managers in mobile phones was discovered. Memory manager's reorganization of memory can result in the overwriting of deleted evidence items. Therefore, the current practice of analyzing mobile phone internal memory through phone operating system should be reconsidered.

Based on these results, it must be clear that a new method for forensic sound analysis of mobile phone internal memory must be based on a system for imaging and direct analysis of the flash memory. Under such a paradigm, mobile phone analysis will follow in the tracks of forensic analysis of computers, where the current practice of imaging and analysis of hard drives has been widely accepted by courts. A requirement to adopt this paradigm for mobile phones is to find a practical method to perform such imaging and analysis of flash memory. This paper has presented two different approches in finding a technique for the physical extraction of flash memory. Both methods are challenging, but it has been shown that they are possible in practice.

For the purpose of further research and ultimately as a new standard method, a new paradigm for mobile phone forensic analysis is hereby proposed:

When mobile phones are seized, they must immediately be shut off. Any SIM card or external flash memory is removed and analyzed separately with a forensic analysis solution. The internal memory of the phone is imaged and analyzed using a method that has yet to be defined. Further research should be conducted in order to determine more practical methods for reading mobile phone internal memory, and enumerate evidence items found therein.

As a final remark, it should be noted that the methods and discoveries developed through this research may be valid not only for mobile phones, but for all forensic analysis of electronics with embedded memory. For example, the author finds it more likely than not that memory managers similar to those on mobile phones exist on units such as PDAs, GPS, vehicle navigation systems and other devices with embedded memory. The successful conservation and extraction of deleted items on such items may rely on the usage of physical memory extraction rather than the current practice of operating system access. The memory extraction methods presented in this paper may be utilized also for such units. This is an imporant area of further research.

About the Author

Svein Y. Willassen has a MSc in Telematics from the Norwegian University of Science and Technology. He has worked as a special investigator at the Norwegian National Computer Crime Center and as Computer Forensic Investigation Manager at Ibas AS. Willassen is currently working on a PhD within Digital Forensics in a research project at the Norwegian University of Science and Technology. He can be reached at svein@willassen.no

References

[1] S. Willassen, *Forensics and the GSM mobile telephone system*, International Journal on Digital Evidence 2003:2:1

[2] R. Knijff, *Embedded Systems Analysis*, Handbook of Computer Crime Investigation, Academic Press, 2002

[3] Intel Corporation, *Ball Grid Array Packaging, 2000 Packaging Handbook*

[4] Intel Corporation, *Intel Wireless Communications and Computing Package User's Guide version 1.2*, May 2004.

[5] N. Lee, *Reflow Soldering Processes and Troubleshooting: SMT, BGA,CSP and Flip Chip Technologies*, Newnes, 2001

[6] H. Manko, *Solders and Soldering*, 4th ed. McGraw-Hill, 2001

[7] B. Vaccaro, R. Shook, D. Gerlach, *The Impact of Lead-free Reflow temperatures on the Moisture Sensivity Performance of Plastic Surface Mount Packages*, SMTA International, Chicago, 2000

[8] IEEE 1149.1, *IEEE Standard Test Access Port and Boundary-Scan Architecture*, IEEE-ANSI, 2001

[9] Intel Corporation, *Designing for On-Board Programming Using the IEEE 1149.1 (JTAG) Access Port*, Application Note, November 1996

[10] Nokia Corporation, *Service Manual NSE-1 Series Cellular Phones*, March 1998.

[11] *JTAG-Tools*, Open Source Project, Available: http://openwince.sourceforge.net/jtag/

[12] *Paraben Cell Seizure*, Software Package, Commercial. Available: http://www.paraben.com/

[13] *Oxygen Phone Manager*, Software Package, Commercial. Available: http://www.oxygensoftware.com/

[14] *TULP2G*, Open Source Project, Available: http://sourceforge.net/projects/tulp2g/

[15] *Chameleon POD*, Reconfigurable JTAG-adapter, Available: http://www.amontec.com/

[16] *Win-Hex*, Software Package, Commercial. Available: http://www.winhex.com/

[17] 3G Partnership Project, *ETSI TS 123.040 - Technical Realization of Short Message Service (SMS)*, version 5.6.1, Sept 2003.

[18] 3G Partnership Project, *ETSI TS 300.642 – AT command set for GSM mobile equipment*, version 5.6.1, Oct 1998.

[19] G. Le Bodic, *Mobile Messaging Technologies and Services*, Wiley, 2003

[20] *EnCase*, Software Package, Commercial. Available: http://www.encase.com/

[21] *The Sleuthkit*, Software Package, Open Source. Available: http://www.sleuthkit.org/