POLITECNICO DI TORINO

Area dell'Ingegneria
Corso di Laurea Magistrale in Ingegneria Informatica - Computer Engineering
(Percorso Software e Sistemi Digitali)

Tesi di Laurea Magistrale

Performance analysis of embedded systems

Relatore:
prof. Violante Massimo
Tutor:
Ing. Macario Gianpaolo

Candidato:
Francesco Castagnotto

Dicembre 2013

A mamma, che sarebbe stata fiera di me.

A Sara, che mi ha sostenuto
in tutto questo percorso.

*”The Linux philosophy is ’Laugh in the face of danger’.
Oops. Wrong One. ’Do it yourself’. Yes, that’s it.”*

(Linus Torvarlds)

Contents

1 Introduction **6**

1.1 Operating Systems virtualizations 8

1.2 Environment Open-Source 9

1.2.1 GPL license 10

1.2.2 Code development and software evolution 11

1.2.3 Community support 12

1.3 Genivi Alliance 13

1.3.1 Project LXCBENCH 17

1.3.2 Operating system mel6-lxcbench 18

1.4 Phoronix Test Suite 18

1.4.1 Dbench 1.0.0 19

1.4.2 Stream 1.1.0 20

1.4.3 Cachebench 1.0.0 20

1.4.4 Encode MP3 1.4.0 21

1.5 Testing boards 21

1.5.1 Freescale i.MX53 QSB 22

1.5.2 TI PandaBoard ES 24

1.6 Sourcery Code Bench 27

2 Linux embedded **28**

2.1 Building an ARM linux operating system 31

2.1.1 Compiling the Linux kernel 3.8.6 32

2.1.2 Deploy software and firmware on target 33

2.2 Operating system mel6-lxcbench 35

2.3 Cross compilation of Phoronix tests 35

2.3.1 Cross compilation of Dbench 1.0.0 37

2.3.2 Cross compilation of Stream 1.1.0 38

2.3.3 Cross compilation of Cachebench 1.0.0 39

2.3.4 Cross compilation of Encode-mp3 1.4.0 41

3 LXC - LinuX Containers 45
3.1 LXC setup . . . 46
3.1.1 Parameter Control Groups . . . 47
3.1.2 Parameter User Namespace . . . 47
3.1.3 Parameter Memory Resource Controller for Control Groups 48
3.1.4 Parameter Cgroup Namespace . . . 49
3.2 LXC execution . . . 49
3.2.1 Creation of a linux container . . . 50
3.2.2 Container start and stop . . . 51
3.3 LXC containers . . . 51
3.3.1 Debian container . . . 51
3.3.2 mel6-lxcbench container . . . 53

4 Tests 55
4.1 Performance on multiple containers . . . 55
4.1.1 Results of Dbench 1.0.0 . . . 56
4.1.2 Results of Stream 1.1.0 . . . 57
4.1.3 Results of Cachebench 1.0.0 . . . 58
4.1.4 Results of Encode-mp3 1.4.0 . . . 58
4.2 Comparison with multiple tests active . . . 59
4.2.1 Results of Freescale i.MX53 QSB . . . 59
4.2.2 Results of TI PandaBoard ES . . . 60

5 Conclusions 61
5.1 Better exploitation of resources . . . 61
5.2 Future implementations . . . 62

6 Glossary 63

7 References 67

A build_kernel.sh 68

B lxc-checkconfig.in 72

C lxc-create.in 75

D lxc-lxcbench (i.MX53) 82

E .config (kernel 3.8.6) 89

Chapter 1

Introduction

The current incredible and fast evolution of computer science technologies led many industrial realities to study new techniques to make their products as ”smart” as possible. Some portable devices created for a more comfortable living, like smartphone and music players, have enhanced the research and development of increasingly smaller circuit with increasingly bigger performance in terms of computing capability and power saving, until reaching the point in which devices as big as a credit card have a processing power that rivals standard computers.
This evolution begins from the industrial optimized ARM family of processors, created for industrial uses where efficiency, power savings and design, materials and assembler optimizations are the main priorities. Then, the next step regarded re-engineering and shrinking all the other peripherals and components, exploiting the pre-existing production technologies. Products sold in the past 10 years proven that the dream of having smaller computers finally become reality.

This technology revolution led the low-level computer science research branch to find new ways to create embedded systems. Before this revolution, developers typically prepared different custom solutions per each problem they had to face, thus creating an extremely fragmented environment in which it was very hard to produce innovation, because every problem had to be solved by building, by scratch, a new customized solution. Industrial automation systems were few and usually based on Intel systems. For this reason the only devices used as controllers were general purpose systems, such as standard computers.
Thanks to this revolution, industrial production realities discovered the benefits of the high level development: developers had to move to work at an higher level of abstraction with systems enabling them to replace entire applications with very little effort. Thanks to this change the industry of embedded devices grew up very quickly to cope with the fast adoption rate of their general purpose solutions.

Due to its origins, these embedded general purpose devices have been extremely optimized: in the world of industrial manufacturing, in fact, wasting processing power is rarely an option. More often, if a product must be used in industrial environments, the hardware is chosen in order to minimize costs: for this reason, if the software it not able to exploit the hardware at its full potential, it is discarded. For this reason, every device is usually equipped with the minimal amount possible of software, to avoid wasting memory, while at the same time every device is carefully configured to enable software to use every resource at its best. Modern hardware is so powerful that it is very difficult to find applications able to efficiently use all the available resources all the time, thus using each device at its full potential. A very elegant solution to solve this problem is to virtualize several entire operating systems in the same machine: with this technique several applications can coexist on the same hardware without interferring each other, thus potentially using all the available resources without the need to customize each application to the capabilities of the host device. This solution quickly became very popular and it is still widely used, especially in modern server farms. Virtualization allows to share hardware between different operating systems and sometimes different kernels, and each operating system can be dedicated to a special job, indipendently from each other; but not all the virtualization systems are the same, because each one has its own method for sharing resources between virtual systems and to manage and process the related tasks.

Particularly *Linux Containers* (*LXC*) provides operating system virtualization without the need of setting up a virtual machine, like the most virtualization programs, but rather provides a virtual environment that has its own process and network space. It allows to run multiple virtual units simultaneously: those units, similar to chroots, are sufficiently isolated to guarantee the required security, but they efficiently share available resources, as everything runs on the same kernel. This virtualization system can be optimal for embedded systems thanks to its lightweight nature and because it can improve perfomance with respect to normal virtual machines, on which a complex virtualization software manages several virtualized operating system instances, with a consequent huge overhead.

The final goal of this thesis is to practically demonstrate that this type of virtualization can greatly increase embedded systems performance: it greatly increase startup speed, and it allows to switch between systems and to manage them in an extremely simple way. These benefits come from the LXC characteristic of using single tasks to contain entire virtual operating systems. Particularly a specific objective is to demonstrate that in architecture with multiple cores, it can be easily allocate the amount of work of a single virtual system (the linux container) on a dedicated core; if each container uses a reserved core for its own work, a multi-core system can make the most of performance for each core: this would be a big revolution for the computer efficiency.

1.1 Operating Systems virtualizations

One of the main costs for an informatic company is the power consumption of the computers. This has been an unresolved problem for a long period of time and in the meantime the computers grew up in capabilities and power consumption. At the same time, every machine needs to be continuously maintained for what concerns operating systems, software and hardware components, with consequent assistance costs.
To solve these problems a brilliant idea is the ”Virtualization”, a common solution to computer science problems used when the hardware is not enough. In this case the virtualization affects the operating system: if a single hardware can run more operating system instances, it is possible to simulate more than one system in just a single physical machine. This idea is enforced by the famous Moore’s law : ”*the number of transistors of a processor and his performance had doubled every 18 months*”. This means that even if problems remain unaltered during time there are cases in which it is convenient to replace old hardware, just because the more recent devices will be more powerful and less hungry in terms of power consumption. For this reason system engineers continuously have at their disposal new hardware, and each of them is twice as powerful than the previous ones: it would be useful to find a simple way to use this new hardware to replace old machines without the need to re-write the software in order to fully exploit their processing capabilities.

A solution to the described problems consists in virtualizing operating systems: this technique is based on using a different virtual environment, composed on a series of virtual peripherals and resources, to run each operating system. This approach enables to map several virtual environments to the same physical hardware, thus enabling several operating systems to coexist in a single machine without even being aware of this sharing of resources. This idea helps to reduce maintenance and power consumption costs, because it makes easier to move software to fewer modern (and efficient) machines in place of being forced to keep several old, slow and expensive devices.

Virtual operating systems

A virtual operating system is an operating system built using references of virtualed resources and peripherals. It is completely working like a normal operating system, but has the limitation defined by the virtualization software and cannot directly access to physical peripherals: some software virtualize the entire system; some others virtualize all but the kernel, that is shared between the systems. Common type of virtual operating systems used in virtualization are the *Virtual Machines*, completely emulated machines with virtualization from the kernel to each peripheral. This allows to simoultaneously virtualize several operating systems based on different kernel and different virtual hardware (such as the number of processors, amount of system memory and video memory and dimension of hard disk).

These features make a virtual system to be completely independent from the hardware, because it works on the virtual hardware defined by the virtualization software and then it has less hardware problems and compatibility with peripherals. All the environment problems of these systems can be solved with the virtualization software that has full control over virtual systems. As a bright side, the virtualization of operating systems requires less maintenance work, has less costs for hardware and software and it has also less power consumption based on the reduction of physical machines.

Linux containers

The *Linux Container* (LXC) is a particular virtualization method where all virtualized systems share the same kernel, the kernel of the base system on which LXC is running. In this method, every virtual system is enclosed inside a single system task and then the management is very simple: all system can be easily run, terminated and managed like a normal process on the system.

LXC is created for the Linux operating systems and relies on the Linux kernel *cgroups* functionality that was released in kernel version 2.6.24. It also relies on other kinds of namespace-isolation functionality, which were developed and integrated into the mainline Linux kernel. The operating of this method is similar to the functionality of *chroot*, but offers much more isolation and all the functionality of a normal system, almost completely indipendently from the base system with the namespace-isolation functionality and his own network space.

1.2 Environment Open-Source

The open-source phylosophy is a rivoluctionary idea for the development of technology solutions.
It was born in the early 1989, with the first version of kernel Linux and GNU

(GNU’s Not Unix) operating system, with the aim to oppose to the ”buy the idea”‘ phylosophy of the main software companies.

The concept is to share the developed code and pay only the work of the programmer, not the algorithms or the code. This causes both the creation of a more collaborative environment and more competition between companies, because a programmer can use the free shared code, asking for clarifications to the author of the code (even if it is a competitor) and then it can create his own version of program. Depending on software costs and development requirements, a customer can choose the preferred version of created programs.

Another important aspect is the centralization of the author of the code: in the open-source development the first person that upload an implemented code is recognized as the author; this means that the companies pay the quality of the code and not just the originality or the unicity of the idea.
If two or more competitors can share information, at least a little,, they collaborate to speed up the evolution of the developed technology: this provides benefits for both of them, as they can produce new technologies sharing problems and solutions, instead using a special product that is the best (and is profitable) just for the little period in which the product does not have competitors.

1.2.1 GPL license

The *GNU General Public Licence* is the most widely used free software license, which guarantees end users (individuals, organizations, companies) the freedoms to use, study, share (copy), and modify the software. Software that ensures that these rights are retained is called free software. The license was originally written by Richard Stallman of the Free Software Foundation (FSF) for the GNU project.

The GPL was written by Richard Stallman in 1989 inside the GNU project, merging similar licenses used for early versions of GNU Emacs the GNU Debugger and GNU C Compiler, to obtain one license that could be used for any project, instead the previous single program licences that were specific to each program rendering them incompatible: this idea making possible the sharing code between different projects.

The GPL is a copyleft license, which means that derived works can only be distributed under the same license terms: a programmer can do anything he wants with the licensed code, but he cannot apply a different license to his work. This does not prohibit the sale of the developed software, but his source code must be free provided. The consequences of this type of licensing is that a programmer that needs to adapt a program to his purposes, can known how the analyzed GPL program works and then he does not need to create a new program, but he can only examine the algorithm of the program in the code and add some configuration files to simple reach his goal.

However, the principles of sharing and ”free code” doesn’t affect the merits of the author, because the rule is ”the first that upload the code on a repository is the author of that code” and then is always recognized as with the copyright. For this, a very important rule of the ”code sharing” is to mention the author of the code that a programmer use for his own purposes, though it is used only a little part of the source code.

1.2.2 Code development and software evolution

The creation of code open-source is an innovated way to create programs. The main idea is to share the created code with anyone, whether if the program is free or f the code is used to a commercial program for sale. This idea brings a single developer’s code into a community of people that can reads that code and suggest ideas to improve the program.

A consequence of this mechanism is that a user of the program that gets some execution errors and knows how to solve them, can implement the patch to correct the wrong functionality and then share this solution with the community: people with same errors can easily solve them and the developer gets free solutions. Using the various web way of communications, the mechanism can be also inverted: the developer that has problems on his code can notify them with relative error messages on a mailing list, a forum or other social media online, to ask if anyone can help him to find the solution: ”all for one, one for all”.

In some cases, the wrong functionalities are not derived from code errors, but from the wanted design limitation and then the developer may not want to solve them; in this case the open-source code allows to someone, or some people, to implement a ”fork” of the program: from the source code is created a new version of the program with functionalities that solve the problems, maybe with some new features, and with another name of program to differentiate from the original program. So the community of users of this program can then choose from different programs, based on their needs and preferences.
In this context, the developer can obviously be the user of another existing open-source program and he can take some part of existing open-source codes to create his program: this can mean also the creation of a fork, that with the adding of some features and changing some part can become the desired program. Moreover, the developer can use some existing programs, with purposes different from his project, because they can be useful for it adding them (maybe with changes) into his project, to make a better and more efficient product: this means more results with half the effort, because someone has done some work for me (like being in a worldwide teamwork).

It seems that sharing the source code with all the world is not a good idea, because is not a good way to earn money; but it is not true at all, because any programmer can analyze this shared code, use the code for his purposes and say an opinion about the product, making good advertising for the developer and for the product in the community or in the workplace. Similarly, a client that want to buy the program can control the quality of the implementation and the work style of the developer before pay for it. Besides, a client is encouraged to choose an open-source product because there is a community that helps the developer to upgrade and improve the program more quicly and efficiently.

1.2.3 Community support

The support of the community is an important part of the open-source development. The word ”community” means all the people with any skills that take part to discussions and problems solving of open-source programs, on various websites (like forum or mailing list). Generally, ”community” indicates any person of the world that decides to partecipate online to the life of open-source program, in any way and with any skills. So a program that have a ”community support” means that every code writed can be analyzed and correct from anyone that found the solution: not always the better solutions comes from the better programmers, but can come from co-workers, normal developers or unknown open-source fans.
For example, the sharing of the code permits to anyone to copy and modify a my program and create his own version (the process of ”fork”). This not means only

have a better program in a version with more features, but if anyone can create a new program starting from my code and ideas (also a competitive program), it creates a mechanism of continue evolution of code, also between competitors, and a more fast evolution of code compared to the development of similar closed-source code programs.

The use of a open-source program has the great characteristic that it is easy to contact the author of the code for support or for some explanations, because an open-source developer is always waiting the opinions of the community to improve his code. Again, if I use a open-source program and I find an error on my particular implementation, I can contact the author of the code to report the problem: when the developer solves the problems, perhaps with a new version of the program, I get a program more efficient for my purposes.

The concept of ”community” is the keystone to spread efficiently knowledge across a wide amount of people. Just few years ago Steve Ballmer, former Microsoft CEO, blamed Linux and the GPL license because, in his opinion, they behave like a virus, by infecting software intellectual properties. This is inaccurate because, as a direct consequence of GPL licence statements, the author of an open-source application is always recognized for its merits as every use of that code must reference the original developers; thus, intellectual property is preserved and, at the same time, the more a product is used and modified by others the more it becomes popular.

1.3 Genivi Alliance

GENIVI is a non-profit industry alliance committed to driving the broad adoption of an In-Vehicle Infotainment open-source development platform. The alliance aims to align requirements, deliver reference implementations, offer certification programs, and foster a vibrant open-source IVI (In-Vehicle Infotainment) community. In other words, the GENIVI Alliance is a non-profit consortium whose goal is to establish a globally competitive, Linux-based operating system, middleware and platform for the automotive IVI industry.

The GENIVI Alliance was founded on March 2, 2009 by BMW Group, Delphi, GM, Intel, Magneti-Marelli, PSA Peugeot Citroen, Visteon, and Wind River

Systems. Since its founding, the alliance has expanded to more than 160 members who are working together to deliver an open and globally consistent software platform based on Linux for use by the whole automotive industry. Among them members, can be found the following:

Automotive manufacturers

- BMW Group
- GM
- Hyundai
- Jaguar Land Rover
- Nissan
- PSA Peugeot Citroen
- Renault
- SAIC Motor

Semiconductor manufacturers

- Altera
- Freescale
- Intel
- Micrel
- nVidia
- Renesas
- STMicroelectronics
- Texas Instruments
- Xilinx

Device manufacturers

- Aisin AW
- Alpine Electronics
- Bosch
- Clarion
- Continental AG
- Delphi
- Denso
- Harman
- Johnson Controls
- LG Electronics
- Magneti-Marelli
- Mobis
- Mitsubishi
- Opensynergy
- peiker
- Pioneer Electronics
- TomTom
- Valeo
- Visteon
- XSe (previously owned by Harman)

Services & software companies

- Canonical

- Elektrobit
- Access Co.
- AllGo Embedded
- Electric Cloud
- EmSyS, Larsen & Toubro Limited
- Ericpol Telecom
- ESG
- ICT Software Engineering
- Jungo LTD
- Klocwork
- KPIT Cummins
- Luxoft
- Mentor Graphics
- MontaVista
- Neusoft Automotives
- Nomovok Oy
- Parasoft
- Parrot
- Pelagicore AB
- Tata Consultancy Services
- Tata Elxsi
- Tieto
- Symphony Teleca Corporation
- Wind River

- Wipro

GENIVI aims to produce a range of compliance statements, and a compliance program for GENIVI certification. To aid this, GENIVI is producing a reference platform to enable members to develop ideas quickly. Internally, the GENIVI community exchanges ideas through a members-only wiki. A GENIVI-compliant operating system is based on Linux. Major Commercial Vendors such as Canonical, Mentor Graphics, MontaVista and Wind River have registered compliant implementations.

1.3.1 Project LXCBENCH

The Genivi project LXCBENCH is an incubation project which has as purpose to gain a better understanding of the capabilities of Linux Containers through benchmarking. The benchmarking activity foreseen aims to measure the performance of an LXC-equipped system versus a reference system not equipped with virtualization. For this purpose it is selected the Phoronix Test Suite (PTS). The wanted measurement of the project to be executed on a given hardware platform are the following:

- Out-of-the-box Linux system. Run PTS to collect a set of measurements to be used as the baseline.
- Out-of-the-box Linux system enriched with 1 container, running a Linux system. Run PTS in the baseline system only, run PTS in the container, run PTS in both concurrently.

After collecting the performance measurements in the above scenario, overheads will be evaluated by comparing the performance score of the baseline system with those of the container-enriched system. It is relevant to underline that the goal of the project is not to measure raw performance of a given hardware platform, rather to understand the overhead coming from the adoption of virtualization on different hardware configurations. For this purpose, it is intended to repeat the same measurements on different hardware platforms, equipped with both single-core and multi-core, and to quantify the overheads before and after virtualization.

The outcomes of this project are twofold:

- A set of configuration scripts to set-up the test environment and run the tests. This will allow interested individuals to reproduce the same experiments that will performed.

- Overhead figures showing the impact of virtualization. The idea of the project is that providing the raw performance figure is not relevant, as this activity does not intend to measure the pure performance of a certain architecture or to compare the pure performance of different architectures. Conversely, it aims at quantifying the price system architects have to pay when including LXC as the virtualization solution in their architectures.

1.3.2 Operating system mel6-lxcbench

For the work of this project it is necessary to use an embedded operating system to collect results. This operating system needs to be very similar to a industrial final product, to have results the more reliable possible. So for this it is used the *mel6-lxcbench* operating system, created into the project *LXCBENCH*. This system is expecially built with some tools needed for every single purposes.

Particularly, the system *mel6-lxcbench* contains:

- a set of minimal Unix commands, to use and configure the system, like *find*, *vi*, *ping*, *ip*
- *LXC* version 0.9.0, the required version to correctly operate on this thesis work
- the *Phoronix Test Suite*, latest version 4.4.1
- the arm-compiled PTS tests *Dbench 1.0.0*, *Stream 1.1.0*, *Cachebench 1.0.0* and *Encode MP4 1.4.0*

This system is derived from a distribution of Mentor Embedded linux-based and made through Yocto, that has the base requirement for the system and can be fully customized for the target needs. For example, every target boards will have his personal embedded system, depending on its own hardware and peripherals.

1.4 Phoronix Test Suite

Phoronix Test Suite (*PTS*) is a free, open-source benchmark software for Linux and other operating systems, developed by Phoronix Media with cooperation from an undisclosed number of hardware and software vendors.
This suite contains a great number of tests to validate software and hardware of a systems. The tests and the suite are free downloadable and open source; they operate simulating complex situations of work and registering the response of the system in each case.

Phoronix is a technology website that offers product reviews, Linux distribution screenshots, interviews, and news while maintaining a pure Linux orientation. The website was founded on 5 June 2004 with a handful of reviews; in the years, it has continued the work of reviews and information about the compare between different systems, but on 5 June 2008 it has released the version 1.0 of the Phoronix Test Suite, a suite of tests with which test can be runned at home by a normal user. At the beginning the Phoronix Test Suite was made up of 57 test profiles and 23 test suites; currently, numbers are about 160 tests and 45 suites. But the numbers are expected to rise, because newer tests are continuously invented and added to this tool.

For this work I used four Phoronix tests: *Dbench version 1.0.0*, *Stream version 1.1.0*, *Cachebench version 1.0.0* and *Encode-MP3 version 1.4.0* (which include for its functioning the test *Timed Audio Encode version 1.0.0*).

1.4.1 Dbench 1.0.0

Dbench is a tool to generate I/O workloads to either a filesystem or to a networked CIFS or NFS server. It can be used to stress a filesystem or a server to see which workload it becomes saturated and can also be used for preditcion analysis to determine ”How many concurrent clients/applications performing this workload can my server handle before response starts to lag?”. It provides a similar benchmarking and client emulation that is implemented in SMBTORTURE using the BENCH-NBENCH test for CIFS, but Dbench can play these loadfiles onto a local filesystem instead of to a CIFS server. Using a different type of loadfiles Dbench can also generate and measure latency for NFS.

Features include:

- Reading SMBTORTURE BENCH-NBENCH loadfiles and emulating this workload as posix calls to a local filesystem

- NFS style loadfiles which allows Dbench to mimic the i/o pattern of a real application doing real I/O to a real server
- iSCSI support and iSCSI style loadfiles

Loadfiles

The heart of Dbench is the concept of a ”loadfile”. A loadfile is a sequence of operations to be performed a statement at a time. This could be operations such as ”Open file XYZ”, ”Read 5 bytes from offset ABC” or ”Close the file”. By carefully crafting a loadfile it is possible to describe an I/O pattern that almost exactly matches what a particular application performs. While cumbersome to produce, such a loadfile it does allow you to describe exactly how and what an application performs and ”replay” this sequence of operations any time you want. Each line in the Dbench loadfile contain a timestamp for the operation. This is used by Dbench to try to keep the same rate of operations as the original application. This is very useful since this allows to perform accurate scalability predictions based on the exact application we are interested in and not an artificial benchmark which may or may not be relevant to our particular applications workload pattern.

1.4.2 Stream 1.1.0

The Stream Benchmark is the de facto industry standard benchmark for the measurement of computer memory bandwidth. It was created by John McCalpin, while he was a professor at the University of Delaware. The Stream benchmark measures ”real world” bandwidth sustainable from ordinary user programs, not the theoretical ”peak bandwidth” provided by most vendors. The Stream benchmark is applicable to a wide range of computers, from PC’s & Mac’s to vector supercomputers and massively parallel processors. Source code in Fortran and C, and binaries for PC’s and Mac’s are available from the Stream ftp site.

1.4.3 Cachebench 1.0.0

The Cachebench is a program to empirically determine some parameters about an architectures memory subsystem. CacheBench has now been integrated into *LLCbench* (*Low Level Characterization Benchmarks*) and measures some architecture operation like Read, Write, Tuned Read, Tuned Write, memcpy e memset, due on the parameters defined at running time.

1.4.4 Encode MP3 1.4.0

The Encode MP3 test is based on the program *Lame* that performs the conversion of a ”Wave” audio file into an mp3 compressed audio file. It uses the secondary *timed-audio-encode* tests, that is not a real tests, but contains the audio files used in the Encode-mp3 test. The goal of this test is to measure the time of conversion of program *Lame* (*Lame Ain't an Mp3 Encoder*) version 3.98, of a wave audio file into a compressed audio file mp3 with bitrate of 128kbps, and based on this time classify the perfomance of the system.

1.5 Testing boards

Testing software before put them on the market is maybe the most important phase in the production of embedded systems. This phase is important to know if the program is well done, but also to see real problems of software applications in the real world. This is the reason for what many money are spent to have optimal tests, sometimes with trial uses in the real world and real users of the final product, before the sale.

Before the current evolution of minimum size of microprocessors and controllers, the testing boards had to be the best similar on operating to the real systems, and sometimes this objective could be reached only through virtualization software or system simulations. This meant testing systems not very optimal, too much different respect to real systems and high costs for testing boards production or for developing software to run on the emulated systems. Sometimes the testing boards must been hardware built and customized for each case, depending on the needs of the producer and of the project, with consequentely high costs to create a good hardware prototype.

Nowadays, the boards are smaller and more powerful than previous and costs have lowered. The low costs of production of these systems permit to have same low-cost boards either on testing than on production: thus tests in the design phase can really measure the real performance and return results of real systems. Besides, this means that in the design phase are not needed expensive prototypes, with hardware customized each time, because these boards are produced for a set of final applications: this makes these embedded system boards almost general-purpose boards and the producer customization can be done at the software level, less expensive.

For this work, I used two different boards, widely used in automotive application.

These boards represent a similar sets of modern boards, starting with a Freescale i.MX53 with single core and then the others with more CPU cores, memory and peripherals.

1.5.1 Freescale i.MX53 QSB

The i.MX53 Quick Start board is a open-source, multi-purpose embedded development platform. The i.MX53 comes with a power-efficient ARM Cortex-A8 based 1 GHz processor with peripherals and hardware accelerated graphics to support applications like human-machine interface (HMI) and support for HD multimedia functions. Also integrated in this platform is the MC34708 power management integrated circuit (PMIC) solution.

It is produced by Freescale, an American company that produces and designs embedded hardware, with 17 billion semiconductor chips in use around the world. The company focuses on the automotive, consumer, industrial and networking markets with its product portfolio including microprocessors, microcontrollers, digital signal processors, digital signal controllers, sensors, RF power ICs and power management ICs. In addition, the company offers software and development tools to support product development.

freescale™
semiconductor

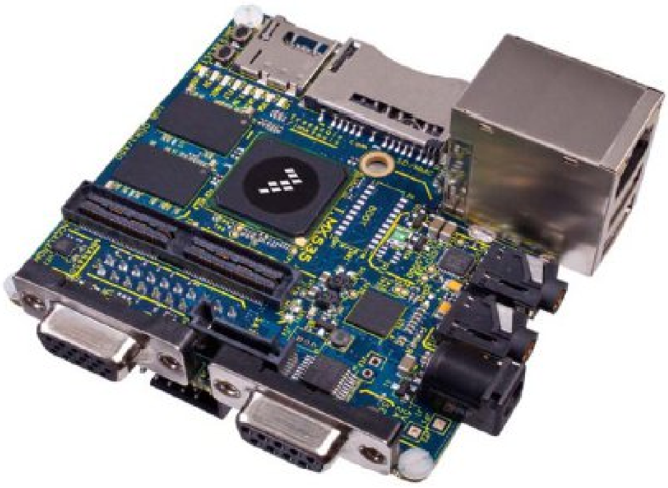


Figure 1.1: Freescale i.MX53 QSB

Its hardware equipment includes:

- processor ARM Cortex-A8 with 1 GHz of frequency
- main memory 1GB DDR3
- power management IC MC34708
- LVDS connector, that support a 10.1” for 1024x768 of pixel resolution (XGA) with capacitive touch (MCIMX-LVDS1)
- VGA connector
- headphone jack
- microphone jack
- analog sensor Freescale 3-axis Accelerometer MMA8450QT
- full–size SD/MMC card slot
- micro SD card slot
- SATA data connector 7–pin
- RJ45 Ethernet port 10/100BT
- 2x high–speed USB port
- micro USB port
- JTAG connector
- DB–9 UART port

This board is optimized for both performance and power to meet the demands of advanced applications. Complete with highly optimized drivers and software, the i.MX53 enables broad-based applications for the embedded consumer, industrial and medical markets.

1.5.2 TI PandaBoard ES

The ”TI PandaBoard ES”, where TI are initials of Texas Instruments, is a low-power and low-cost single-board computer development platform, based on the system on a chip TI OMAP4460; it is the latest version of PandaBoard boards, with processor OMAP4460 clocked at 1.2 GHz. The PandaBoard is a platform designed to provide access to as many of the powerful features of the OMAP4460 Multimedia Processor as possible, while maintaining a low cost. It supports development of additional capabilities/functionality, providing expandability via on-board connectors.

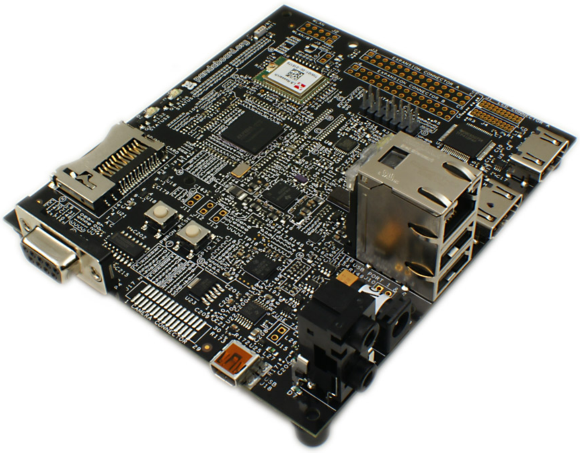


Figure 1.2: TI PandaBoard ES

Its hardware equipment includes:

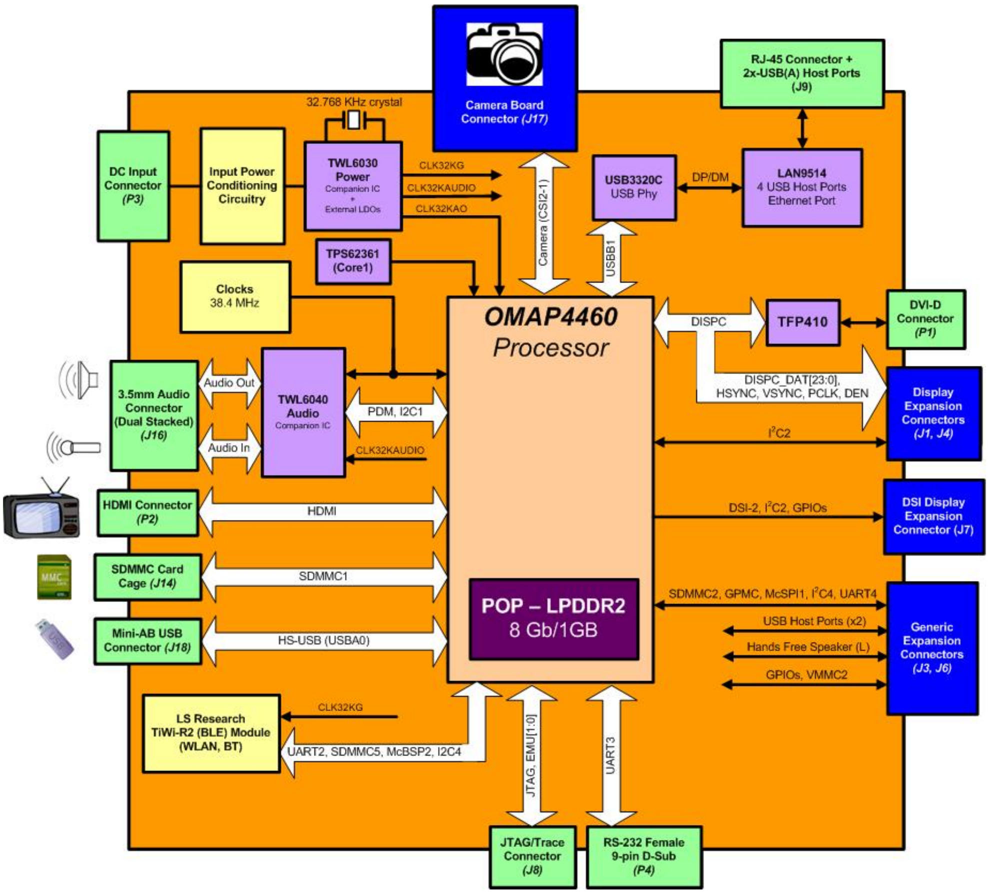
- dual-core ARM Cortex-A9 MPCore with Symmetric Multiprocessing (SMP) at 1.2 GHz each

- full HD 1080p multi-standard video encode/decode

- imagination technologies POWERVR SGX540 graphics core supporting all major API’s including OpenGL ES v2.0, OpenGL ES v1.1, OpenVG v1.1 and EGL v1.3

- HDMI v1.3 Connector (Type A)

- DVI-D Connector (can drive a 2nd display, simultaneous display)
- LCD expansion header
- DSI Support
- 1 GB low power DDR2 RAM
- full size SD/MMC card cage with support for High-Speed & High-Capacity SD cards
- 3.5” Audio in/out



Camera Board Connector (J17)
RJ-45 Connector + 2x-USB(A) Host Ports (J9)
32.768 KHz crystal
DC Input Connector (P3)
Input Power Conditioning Circuitry
TWL6030 Power Companion IC + External LDOs
CLK32KG
CLK32KAUDIO
CLK32KAO
Camera (CSI2-1)
USB3320C USB Phy
DP/DM
LAN9514 4 USB Host Ports Ethernet Port
USBB1
TPS62361 (Core1)
Clocks 38.4 MHz
OMAP4460 Processor
DISPC
TFP410
DVI-D Connector (P1)
DISPC_DAT[23:0], HSYNC, VSYNC, PCLK, DEN
Display Expansion Connectors (J1, J4)
I²C2
3.5mm Audio Connector (Dual Stacked) (J16)
Audio Out
Audio In
TWL6040 Audio Companion IC
PDM, I2C1
CLK32KAUDIO
HDMI Connector (P2)
HDMI
DSI-2, I²C2, GPIOs
DSI Display Expansion Connector (J7)
SDMMC Card Cage (J14)
SDMMC1
MMC
POP – LPDDR2 8 Gb/1GB
SDMMC2, GPMC, McSPI1, I²C4, UART4
USB Host Ports (x2)
Hands Free Speaker (L)
GPIOs, VMMC2
Generic Expansion Connectors (J3, J6)
Mini-AB USB Connector (J18)
HS-USB (USBA0)
USB
CLK32KG
LS Research TiWi-R2 (BLE) Module (WLAN, BT)
UART2, SDMMC5, McBSP2, I2C4
JTAG, EMU[1:0]
UART3
JTAG/Trace Connector (J8)
RS-232 Female 9-pin D-Sub (P4)

Figure 1.3: TI PandaBoard ES diagram

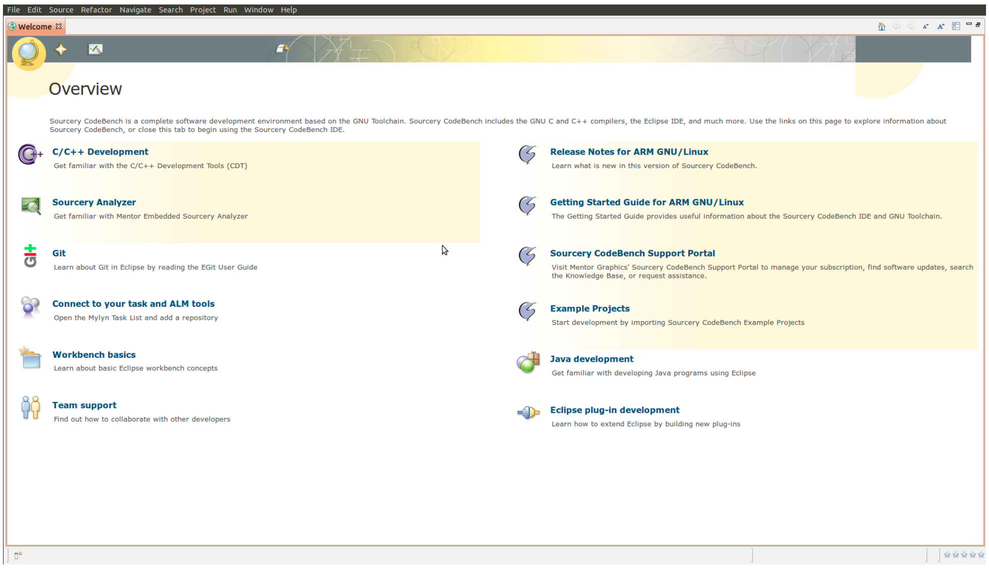
- RJ45 Ethernet connector 10/100
- Bluetooth v2.1

- 1x USB 2.0 High-Speed On-the-go portability
- 2x USB 2.0 High-Speed host ports
- General purpose expansion header (I2C, GPMC, USB, MMC, DSS, ETM)
- Camera expansion headers
- JTAG
- UART/RS-232
- 2 status LEDs (configurable)
- 1 GPIO Button

It is a development platform supported by the pandaboard.org community. It is optimized to run various mobile open source software distributions such as Android, Ubuntu, Firefox OS and more.

1.6 Sourcery Code Bench

Sourcery Code Bench is an integrated development environment (IDE) developed by Mentor Graphics and customized for embedded works. It is a complete development environment for embedded C/C++ development on ARM, Power, Coldfire, and other architectures. Sourcery Code Bench gives many developing tool like compilers xand run-time libraries, source-and assembly-level debugger and more. Being an extension of Eclipse IDE, it allows to use a huge set of useful Eclipse tools, with many new software solutions that make easy the embedded development, like the cross compiling to another target architecture, and the remote debugging on the target hardware, without deploy the program on the board system.



Overview

Sourcery CodeBench is a complete software development environment based on the GNU Toolchain. Sourcery CodeBench includes the GNU C and C++ compilers, the Eclipse IDE, and much more. Use the links on this page to explore information about Sourcery CodeBench, or close this tab to begin using the Sourcery CodeBench IDE.

C/C++ Development
Get familiar with the C/C++ Development Tools (CDT)

Sourcery Analyzer
Get familiar with Mentor Embedded Sourcery Analyzer

Git
Learn about Git in Eclipse by reading the EGit User Guide

Connect to your task and ALM tools
Open the Mylyn Task List and add a repository

Workbench basics
Learn about basic Eclipse workbench concepts

Team support
Find out how to collaborate with other developers

Release Notes for ARM GNU/Linux
Learn what is new in this version of Sourcery CodeBench.

Getting Started Guide for ARM GNU/Linux
The Getting Started Guide provides useful information about the Sourcery CodeBench IDE and GNU Toolchain.

Sourcery CodeBench Support Portal
Visit Mentor Graphics' Sourcery CodeBench Support Portal to manage your subscription, find software updates, search the Knowledge Base, or request assistance.

Example Projects
Start development by importing Sourcery CodeBench Example Projects

Java development
Get familiar with developing Java programs using Eclipse

Eclipse plug-in development
Learn how to extend Eclipse by building new plug-ins

Sourcery Code Bench is produced by Mentor Graphics, a multinational corporation dealing in electronic design automation for electrical engineering and electronics. Founded in 1981, the company is headquartered in Wilsonville, Oregon and employs roughly 4,400 people worldwide. This company produce some software solutions for automotive, aerospace and military, low power, 3D-IC design and test, accelerating ARM-based design, and more. Among its products, in addition to Sourcery Code Bench, can be found the Nucleos OS, an efficient real-time operating system, the Mentor Embedded Linux, a customized embedded operating system linux-based, and the Mentor's Inflexion UI for Android, for implement visually rich user interfaces on Android devices for OEMs and ODMs.

Chapter 2

Linux embedded

The ”embedded operating systems” are operating systems customized for the type and purpose of the target machine hardware. In the past years for each machine was produced a big number of different embedded operating systems, designed on the specific machine, on the specific purposes and required future uses, and made using the code language preferred by the developers. This type of reasoning had created some problems for the maintenance of the systems, where the technician had to known all the possible languages used and all the possible hardware used; some problems was created also from the costs of production: make a single product, like a production of a prototype, is more expensive respect to the serial production. The solution for these problems can be an environment where the hardware of the machine (expecially the computer and micro-controller hardware used to make the machine intelligent) is common between different type of applications, and where the software used to develop is an high-level software, using high-level code languages that permit a reduction of software unicity and of less used code languages (it is less expensive use a new, more performant and more known code language) and more easy way to improve the code and make new code evolutions to resolve problems or expand the goals of the machine. Also, an high-level embedded program permits the compatibility with some common used high-level programs and technologies: the portability of the HTTP protocol on the mobile phones is one of many example of this fact, an incredible thing if compared with older methods to bring internet on phones, like the wap.

For all this reasons, a very good solution is the Linux for embedded systems. Linux is one of the most used kernel for embedded operating systems, because can be completely customized for the specific uses or hardware. It is a simple monolithic operating system, developed in C that is one of the most known programming language, and it was continuously evolved in this years by the support of its worldwide community. Linux is a kernel for general purpose operating sys-

tems, so the customization means to create a system designed for the target but at the same time the system can built to support programs that run on general purpose systems linux-based: this is the interesting feature of derive an embedded system starting from a general purpose system. The characteristic of the monolithic structure is not very optimal for the execution, but has a good influence on the development by virtue of the simplicity of system construction, like the construction of new drivers or functionalities for the target, that can be simple inserted into the operating system tree.
Linux is a kernel developed to operate on a large number of machines, because was invented and developed to satisfy needs of a community of people with different computers. So for this, has a large compatibility of hardware and different processors: this characteristic can be used for a easy development for compatibility on different embedded systems hardware. The existence of a common operating system for the similar -but not the same- hardware is a good benefit for the embedded systems, because if can be created a operating system with kernel for a family of hardware, it eliminates the downside of development of a single system for every single machine, and it reduce the costs of development of a new operating system (it is only needed making some changes to previously version). This also has influence on all the problems of maintenance, because each machine has its kernel but the management commands are similar and the system functionalities are the same. Besides if there is an open-source production line, many companies can easily find an existing optimal software for their machines and can collaborate in the community of developers to have a better product.

Android

A good example of all these benefits is Android. Android is a multi-platform embedded operating system Linux-based. It is developed for multiple hardware, which have multiple different peripherals (like camera, IR, bluetooth or GPS) and different applications based on the type of device (like smartphone, PDA, tablet, car computer or others). This implies that the parameters to choose the better product between different device vendors are the type of hardware and peripherals, the amount of memories, performance of components and version of Android deployed on it: this is a revolution where the old idea of ”better is the most beautiful“ is replaced by ”better is the most performant“, and where the average customer has more computer skills. The ”average customer” becames also more competent in application development for this embedded linux-based operating system, as demonstrate by the big number of developers of Android App from many non-computer environments, and based also on development of new derived version of the operating system, like the various forks of Android easily available on Internet: the strength of the open-source! Moreover, the more knowledge of the average buyer leads to a good indirect advertising to anyone: few people know what exactly is ”Android”, but all know how to use it and say that is the on of the best. Be an ”open-source operating system” has one of the most benefits in the production, because the Google developer team creates the base version of Android and entrusts the final work of system customization for each single target to the various hardware producers and sellers, that among other things can add some feature for their purposes: this means that the embedded software developer provides only a semi-finished product, instead a finished product compatible only with few devices, and the vendor becames the final producer and developer of the software. For this, with this technique the number of vendors can easily grow.

The benefit of the use of a common-base system is to enter in a special community of ”Android users”, where a single user uses Android like others with different devices; so the hardware limitation is relegated only to few difference of peripherals, but everybody can install and uses the same ”Android App”. This is a revolution for the developers, because for first they can do not care about final target in the development, the operating system permits to have an high-level development (with benefits of more tools and potentialities) and they can base their idea on a generic ”Android product”, without too many problems of hardware compatibility: most of these problems are solved with the vendor’s customization on his Android version. Another good result of the high-level is to carry some users of common devices, to the development of Android software, because the high-level is less complex to learn instead of the low-level, and often some development tools for high-level languages permit to create good programs by anyone with almost none knowledge of programming; furthermore, the basic language to develope on Android is derived from Java, a language easy to learn.

Also the future uses of Android are influenced by all these characteristics, because before Android not only each vendor had his personal operating system, but often each device of the same vendor had different operating systems between them, with big differences between similar versions; that implied a big incompatibility between similar devices of the same vendor. Instead, the Android compatibility leads to know that the future operating systems of any ”Android device” will be Android. So a single product can have a future evolution, like a fork of an older version, and the newer apps will be compatible with older versions, installed on older versions of Android: this means a more beautiful future for every app-teams and for every developers!

2.1 Building an ARM linux operating system

The start of my work is the creation of an embedded distro Linux-based on which test the functionalities of an embedded system and the methods of LXC program to create and run the containers.
I used a board *Freescale i.MX53 QSB*, a *linux kernel version 3.8.6* and an operating system *Debian ARM version 6.0.6*. I followed the instructions of compilation of Robert Nelson, the author of the wiki `http://www.eewiki.net`, and I used some tools provided on various GIT repositories.

For first, I needed the right compiler to cross-compile on my machine, some parts to functioning on the target architecture. My targets are boards with ARM architectures and then I chose the free ARM compiler *gcc-arm-linux-gnueabi* easily findable on the repositories of the modern GNU/Linux distributions, because is a part of the compiler system *GNU Compiler Collection* (*GCC*), the standard compiler of most modern Unix-like computer operating systems. The next step was to get a bootloader, to obtain a startup of the board, and *Das U-Boot* was a good solution. *Das U-Boot* universal bootloader is an open source, primary boot loader used in embedded devices. It is available for a number of different computer architectures, including PPC, ARM, MIPS, and x86, and it is arguably the richest, most flexible, and most actively developed open source bootloader available.

To download a version of u-boot I used this command:

```
$ git clone git://git.denx.de/u-boot.git
```

after the download, I added to the bootloader two patches provided by Robert Nelson,

```
$ cd u-boot/
# wget https://raw.github.com/RobertCNelson/Bootloader-Builder/master/
patches/v2013.01/0001-enable-bootz-and-generic-load-features.patch
# wget https://raw.github.com/RobertCNelson/Bootloader-Builder/master/
patches/v2013.01/0002-imx-convert-to-uEnv.txt-n-fixes.patch
# git checkout v2013.01 -b tmp
# patch -p1 < 0001-enable-bootz-and-generic-load-features.patch
# patch -p1 < 0002-imx-convert-to-uEnv.txt-n-fixes.patch
```

and I compiled the bootloader with the arm compiler

```
# make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- distclean
# make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- mx53loco_config
# make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-
```

Then, I had to download the Linux kernel and I used a download method of Robert Nelson. In this download is already present an automatic script *build_kernel.sh* to easily compile the kernel. Following, the download and compilation of the Linux kernel.

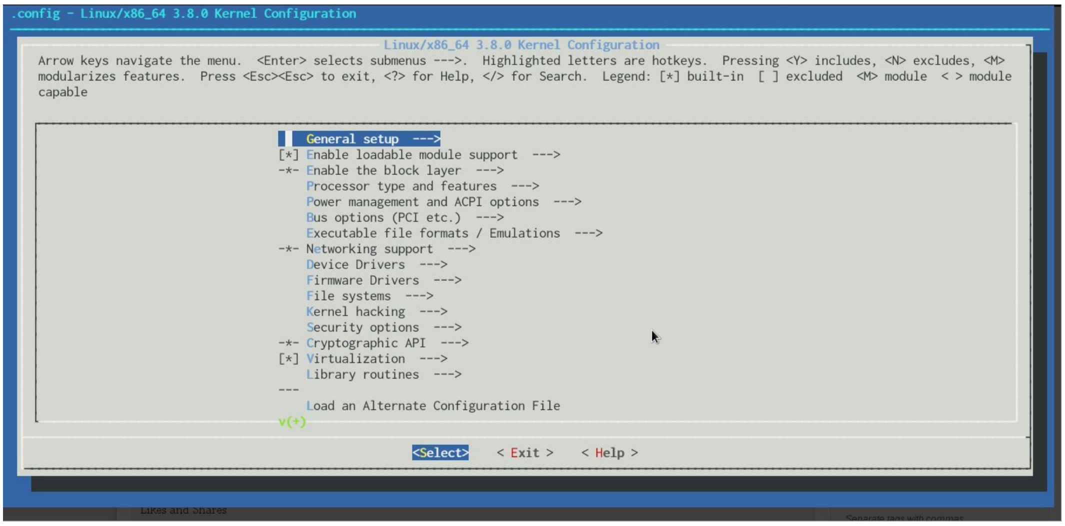
```
$ git clone git://github.com/RobertCNelson/imx-devel.git

$ cd ./imx-devel
# git checkout origin/v3.8.x-imx -b tmp
# ./build_kernel.sh
```

(The code of this script is in the *Appendix A*)

2.1.1 Compiling the Linux kernel 3.8.6

During the execution, the script *build_kernel.sh* opens the *make menuconfig*, an useful tool that permits to configure the Linux kernel source. *make menuconfig* with a menu-driven user-interface, allows to choose the features and other options of the Linux kernel that will be compiled.



```
.config - Linux/x86_64 3.8.0 Kernel Configuration
                              Linux/x86_64 3.8.0 Kernel Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M>
  modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module
  capable

                                   General setup  --->
                               [*] Enable loadable module support  --->
                               -*- Enable the block layer  --->
                                   Processor type and features  --->
                                   Power management and ACPI options  --->
                                   Bus options (PCI etc.)  --->
                                   Executable file formats / Emulations  --->
                               -*- Networking support  --->
                                   Device Drivers  --->
                                   Firmware Drivers  --->
                                   File systems  --->
                                   Kernel hacking  --->
                                   Security options  --->
                               -*- Cryptographic API  --->
                               [*] Virtualization  --->
                                   Library routines  --->
                               ---
                                   Load an Alternate Configuration File
                               v(+)

                                        <Select>    < Exit >    < Help >
```

Figure 2.1: make menuconfig

The compilation options are identified by some parameters that can be setted in this menu on *yes* or *no*, to define if the respective kernel modules are required. Some item depends on others: so it is necessary set to *yes* the dependent options, to be able to enable the desired parameter.
The various versions of kernel can have different name for same parameter, due the evolution or the entire change of the functionality. Some of these changes are based on requirement for reinventing the kernel. Thus sometimes, the building of the kernel require searches on Internet, on mailing lists, on forum or on other web sites, to find the new names of wanted parameters.

In this case to correctly run the LXC *version 0.7.5* on the system, the needed parameters are the following:

```
# --- LXC SETUP ---#
CONFIG_GROUP_SCHED=y
CONFIG_FAIR_GROUP_SCHED=y
CONFIG_RT_GROUP_SCHED=y
CONFIG_CGROUP_SCHED=y
CONFIG_CGROUPS=y
CONFIG_CGROUP_NS=y
CONFIG_CGROUP_FREEZER=y
CONFIG_CGROUP_DEVICE=y
CONFIG_CPUSETS=y
CONFIG_PROC_PID_CPUSET=y
CONFIG_CGROUP_CPUACCT=y
```

```
CONFIG_RESOURCE_COUNTERS=y
CONFIG_CGROUP_MEM_RES_CTLR=y
CONFIG_CGROUP_MEM_RES_CTLR_SWAP=y
CONFIG_MM_OWNER=y
CONFIG_NAMESPACES=y
CONFIG_UTS_NS=y
CONFIG_IPC_NS=y
CONFIG_USER_NS=y
CONFIG_PID_NS=y
CONFIG_NET_NS=y
CONFIG_NET_CLS_CGROUP=y
CONFIG_SECURITY_FILE_CAPABILITIES=y
CONFIG_DEVPTS_MULTIPLE_INSTANCES=y
CONFIG_VETH=y
CONFIG_MACVLAN=y
CONFIG_VLAN_8021Q=y
CONFIG_POSIX_MQUEUE=y
```

To control the kernel configuration on a running system and show the setted parameters, I used the command:

```
$ zcat /proc/config.gz
```

If the required parameters are not setted, it is necessary recompile the kernel. When I downloaded the Linux kernel, I obtained the version 3.8.6. In this version some parameters are different and others are no longer present; for this reason, following items was not in my *make menuconfig*:

```
CONFIG_GROUP_SCHED
CONFIG_CGROUP_NS
CONFIG_CGROUP_MEM_RES_CTLR
CONFIG_CGROUP_MEM_RES_CTLR_SWAP
CONFIG_SECURITY_FILE_CAPABILITIES
```

(The final parameters configuration of my kernel is in the *Appendix E*)

Then, I needed to proceed on my work with the system installation, because it is possible that some missing options do not affect the kernel functionalities or they can be enabled at run time. Once closed the make menuconfig, the build script continued with the compilation and complete building of the kernel.

2.1.2 Deploy software and firmware on target

When the kernel is compiled, it needs an OS to operate with it and make the target operating. For my work I used a Debian linux, version for ARM architectures. So, I easily got the distro and untar on my machine:

```
$ wget -c http://rcn-ee.net/deb/minfs/squeeze/debian-6.0.6-minimal-armel-2013-02-17.tar.xz
$ tar xvJf debian-6.0.6-minimal-armel-2013-02-17.tar.xz
```

This operating system will be simply copied to the target hard disk bit-to-bit. Then, it is only necessary download the firmware for the target board used (in this example is the Freescale i.MX53) and compile it

```
# git clone git://git.pengutronix.de/git/imx/sdma-firmware.git
# cd sdma-firmware && make CC=gcc
```

and then prepare the microSD card used for hard disk of the i.MX53

```
$ export DISK=/dev/mmcblk0
$ sudo dd if=/dev/zero of=${DISK} bs=1024 count=1024 && sudo parted --script ${DISK} mklabel msdos
$ sudo mkfs.ext2 /dev/mmcblk0p1 -V && sudo mkfs.ext4 /dev/mmcblk0p2 -V
```

Finally, it is possible to copy all the sources on the relative partitions on the card.

Deploy the software

The u-boot bootloader must be copied in the first sector of the card

```
# dd if=./u-boot/u-boot.imx of=/dev/mmcblk0 seek=1 bs=1024
```

and it needs a configuration file *uEnv.txt* that defines the boot parameters like this:

```
initrd_high=0xffffffff
fdt_high=0xffffffff
dtb_file=imx53-qsb.dtb
console=ttymxc0,115200
mmcroot=/dev/mmcblk0p2 ro
mmcrootfstype=ext4 rootwait fixrtc
optargs=console=tty0
video=mxcdi1fb:1280x720M@60
mmc_load_image=load mmc 0:1 0x70010000 zImage
mmc_load_initrd=load mmc 0:1 0x72000000 initrd.img; setenv initrd_size ${filesize}
mmc_load_dtb=load mmc 0:1 0x71ff0000 /dtbs/${dtb_file}
mmcargs=setenv bootargs console=${console} ${optargs} root=${mmcroot}
rootfstype=${mmcrootfstype} ${video}
#Just: zImage
xyz_mmcboot=run mmc_load_image; run mmc_load_dtb; echo Booting from mmc ...
loaduimage=run xyz_mmcboot; run mmcargs; bootz 0x70010000 - 0x71ff0000
#zImage and initrd
#xyz_mmcboot=run mmc_load_image; run mmc_load_initrd; run mmc_load_dtb; echo Booting
from mmc ...
#loaduimage=run xyz_mmcboot; run mmcargs; bootz 0x70010000 0x72000000:${initrd_size}
0x71ff0000
```

The other software can be copied on the relative partitions:

```
# cp -v ./imx-devel/deploy/3.8.6-imx5.zImage /media/boot/zImage
# mkdir -p /media/boot/dtbs
# tar xfv ./imx-devel/deploy/3.8.6-imx5-dtbs.tar.gz -C /media/boot/dtbs/
# mkdir -p /media/rootfs/lib/firmware/sdma
# cp -v ./sdma-firmware/sdma*.bin /media/rootfs/lib/firmware/sdma/
# tar xfvp ./debian-6.0.6-minimal-armel-2013-02-17/armel-rootfs-debian-squeeze.tar -C /media/rootfs/
# tar xfv ./imx-devel/deploy/3.8.6-imx5-modules.tar.gz -C /media/rootfs/
```

Configuration of the operating system

The last steps are the configurations of the operating system needed to correctly starts Debian. So it is needed to add following lines to `/media/rootfs/etc/fstab`

```
/dev/mmcblk0p2    /             auto    errors=remount-ro   0    1
/dev/mmcblk0p1    /boot/uboot   auto    defaults            0    2
```

and this line to the file `/media/rootfs/etc/inittab` to indicate the terminal to use as system console (here it is indicated *ttymxc0* to use the serial console of the board i.MX53)

```
T2:23:respawn:/sbin/getty -L ttymxc0 115200 vt102
```

Finally the configuration of ethernet interfaces to get connected to an ethernet network, by adding these lines to `/media/rootfs/etc/network/interfaces`:

```
auto lo
iface lo inet loopback
auto eth1
iface eth1 inet dhcp
```

Now, the microSD can be inserted into the board and the embedded system can start.

2.2 Operating system mel6-lxcbench

The *mel6-lxcbench* is a customized embedded operating system developed into the project LXCBENCH. This Linux-based distribution is built starting from a Yocto distribution of Mentor Graphics with some customization and additions, to have a complete linux distribution for the purposes of the LXCBENCH project.

As bootloader it uses the universal bootloader *Das U-Boot* and the kernel is the Linux version 2.6.35 for the Freescale i.MX53, and the Linux version 3.1.0 for the TI PandaBoard ES; these kernels are compiled with the right parameters for the purpose of the project, for example settings the parameters needed for the functioning of the LXC containers. The operating system is a customization owned by Mentor Graphics with some programs added, to have the minimum tools useful for the project LXCBENCH. For example, there is the latest LXC version 0.9.0, the latest Phoronix Test Suite version 4.4.1 and the cross-compiled tests of Phoronix: Dbench version 1.0.0, Stream version 1.1.0, CacheBench version 1.0.0 and Encode-mp3 version 1.4.0. There are also some common-use commands, like the *ip* set of commands or some unix-base commands (*dd*, *cp*, *adduser*, *cat*) and some useful programs like *vi*, *ps*, *top*, *grep*.

2.3 Cross compilation of Phoronix tests

When LXC try to create a container, it needs a tool to download and unpack the right version of operating system selected in the command. This tool is *debootstrap* that connects to the repository online of the selected distro, it downloads and unpacks the archive containing the system image, and then provides the downloaded system to LXC that can continue the installation of the operating system inside the container. It is a useful tool because it resolve all the problem about the right repository for every distro and the checksum controls for the archive and if necessary it download again the archive, if corrupted. It provide an useful and automatic tool, called directly from LXC without additional request to the user, from the command *lxc-create*.

Whithout this automated tool, tests must be downloaded and compiled for the target architecture, before can run them. Following sections shows the scripts and procedures that I used to download, cross-compile and prepare to run the tests on the target system. A script that is common for every tests is *script-pts-xml.sh*, a script that create the file *pst-install.xml*, required from the Phoronix Test Suite, to know that the test is installed on the system and to get some information about it.

An example of *pts-install.xml* is the following:

```
<?xml version="1.0"?>
<!--Phoronix Test Suite v4.4.1 (Forsand)-->
<?xml-stylesheet type="text/xsl"
 href="file:////home/debian/.phoronix-test-suite/xsl/pts-test-installation-viewer.xsl"?>
<PhoronixTestSuite>
  <TestInstallation>
    <Environment>
      <Identifier>pts/dbench-1.0.0</Identifier>
      <Version>1.0.0</Version>
      <CheckSum>dc383f29025b00a6cad1f825d2b87b99</CheckSum>
      <CompilerData>{"compiler-type":"CC","compiler":"gcc",
      "compiler-options":"-lpopt"}</CompilerData>
      <SystemIdentifier>QVJNdjcgcmV2IDVfX0ZyZWVzY2FsZSBpLk1YNTMgKERldmljZSBUcmVlIFN1
      cHBvcnQpX19EZWJpYW4gNi4wLjdfX0dDQyA0LjQuNQ==</SystemIdentifier>
    </Environment>
    <History>
      <InstallTime>2013-05-03 16:11:35</InstallTime>
      <InstallTimeLength>139</InstallTimeLength>
      <LastRunTime>0000-00-00 00:00:00</LastRunTime>
      <TimesRun>0</TimesRun>
      <AverageRunTime></AverageRunTime>
      <LatestRunTime></LatestRunTime>
    </History>
  </TestInstallation>
</PhoronixTestSuite>
```

The script *script-pts-xml.sh* then must create a file like the above, with the correct values of test name and version. The source of that scripts is the following.

```
#This script generates the pts-install.xml file, needed by the Phoronix Suite to correctly
#run the tests.
#The generated file must be copied inside the folder of the test
#"$HOME/.phoronix-test-suite/installed-tests/pts/<name_test>/"
#It assumes to default that the user on the target system is "debian":
#if not, change it in the "href" field.

#!/bin/bash

if [ -z $1 ];
then
echo "ERROR: missing parameter (name of test)"
else

echo "<?xml version=\"1.0\"?>
<!--Phoronix Test Suite v4.4.1 (Forsand)-->
<?xml-stylesheet type=\"text/xsl\"
href=\"file:////home/debian/.phoronix-test-suite/xsl/pts-test-installation-viewer.xsl\"?>
<PhoronixTestSuite>
  <TestInstallation>
    <Environment>
```

```
      <Identifier>pts/$1</Identifier>
      <Version>1.0.0</Version>
      <CheckSum>dc383f29025b00a6cad1f825d2b87b99</CheckSum>
<CompilerData>{\"compiler-type\":\"CC\",\"compiler\":\"gcc\",\"compiler-options\":\"-lpopt\"}
      </CompilerData>
      <SystemIdentifier>
QVJNdjcgcmV2IDVfX0ZyZWVzY2FsZSBpLk1YNTMgKERldmljZSBUcmVlIFN1
cHBvcnQpX19EZWJpYW4gNi4wLjdfX0dDQyA0LjQuNQ==
      </SystemIdentifier>
    </Environment>
    <History>
      <InstallTime>2013-05-03 16:11:35</InstallTime>
      <InstallTimeLength>139</InstallTimeLength>
      <LastRunTime>0000-00-00 00:00:00</LastRunTime>
      <TimesRun>0</TimesRun>
      <AverageRunTime></AverageRunTime>
      <LatestRunTime></LatestRunTime>
    </History>
  </TestInstallation>
</PhoronixTestSuite>

" > pts-install.xml
chmod 0777 pts-install.xml

fi
```

This script is launched from the automated scripts described in the following section, because every test needs its own *pts-install.xml* file.

2.3.1 Cross compilation of Dbench 1.0.0

For an automated cross compilation of the test Dbench, I created the scripts *script-dbench.sh* and *install-dbench.sh*. Following is described the process of the execution of the cross compilation using these scripts.

First step, I downloaded the test using this command on the target board:

```
$ phoronix-test-suite download-test-files pts/dbench
```

The following *script-dbench.sh* is a simple script that unzip the test archive and downloads the test sources using the url inside the file *download.xml*. After, it launches the script *install-dbench.sh* for the script setup and *script-pts-xml.sh* to create a customized *pts-install.xml* for Dbench.

The script *script-dbench.sh* is:

```
####################################################################################
#   DBENCH-1.0.0      #
####################################################################################

#This script run the download of the source of the Phoronix test Dbench-1.0.0
#and the cross compilation to functioning on a ARM linux architecture.

#It require the ZIP test file "dbench-1.0.0.zip" in the same folder of this script.
#(The file can be downloaded on the target using the Phoronix Suite by the command
#"phoronix-test-suite download-test-files pts/dbench", and then can be found inside
#the folder "$HOME/.phoronix-test-suite/openbenchmarking.org/pts/")
```

```
#!/bin/bash
unzip dbench-1.0.0.zip
wget $(xpath -e "//URL" downloads.xml 2>/dev/null | \
sed -e ’s/<\/*URL>//g’ -e ’s/<URL>//g’ -e ’s/<\/URL>/ /g’)
sudo ./install-dbench.sh
./script-pts-xml.sh dbench-1.0.0
```

This script launches the second part of setup, that is the *install-dbench.sh*, which source is below:

```
##################################################################################
#   INSTALL DBENCH-1.0.0                                                         #
##################################################################################
#This script run the cross-compilation of the Dbench test.

#!/bin/sh

tar -zxvf dbench-4.0.tar.gz
mkdir $HOME/dbench_/
cd dbench-4.0/

./autogen.sh
./configure -build=i686-linux --host=arm-linux-gnueabi --prefix=$HOME/dbench_/

make -j $NUM_CPU_JOBS
echo $? > ˜/install-exit-status
make install
cp client.txt ../
cd ..
rm -rf dbench-4.0/

echo "#!/bin/sh
./dbench_/bin/dbench \$@ -c client.txt > \$LOG_FILE 2>&1
echo \$? > ˜/test-exit-status" > dbench

chmod +x dbench
```

The final step is to put the directory `$HOME/dbench_/`, the file `client.txt`, the file `pts-install.xml` and the script `dbench` under the target system path `$HOME/.phoronix-test-suite/installed-tests/pts/dbench-1.0.0`

2.3.2 Cross compilation of Stream 1.1.0

For an automated cross compilation of the test Stream, I created the scripts *script-stream.sh* and *install-stream.sh*. Following is described the process of the execution of the cross compilation using these scripts.

First step, I downloaded the test using the command on the target board:

```
$ phoronix-test-suite download-test-files pts/stream
```

The following *script-stream.sh* is a simple script that unzip the test archive and downloads the test sources using the url inside the file *download.xml*. After, it launches the script *install-stream.sh* for the script setup and *script-pts-xml.sh* to create a customized *pts-install.xml* for Stream.

The script *script-stream.sh* is:

```
####################################################################################
#   STREAM-1.1.0      #
####################################################################################

#This script run the download of the source of the Phoronix test Stream-1.1.0
#and the cross compilation to functioning on a ARM linux architecture.

#It require the ZIP test file "stream-1.1.0.zip" in the same folder of this script.
#(The file can be downloaded on the target using the Phoronix Suite by the command
#"phoronix-test-suite download-test-files pts/stream", and then can be found inside
#the folder "$HOME/.phoronix-test-suite/openbenchmarking.org/pts/")

#!/bin/bash

unzip stream-1.1.0.zip
wget $(xpath -e "//URL" downloads.xml 2>/dev/null | \
sed -e 's/<\/*URL>//g' -e 's/<URL>//g' -e 's/<\/URL>/ /g')
sudo ./install-stream.sh
./script-pts-xml.sh stream-1.1.0
```

This script launches the second part of setup, that is the *install-stream.sh* which source is below:

```
####################################################################################
#   INSTALL STREAM-1.1.0                                                           #
####################################################################################
#This script run the cross-compilation of the Stream test.

#!/bin/sh

tar -zxvf stream-2009-04-11.tar.gz

arm-linux-gnueabi-gcc stream.c -O2 -fopenmp -o stream-bin
echo \$? > ~/test-exit-status

echo "#!/bin/sh
export OMP_NUM_THREADS=\$NUM_CPU_CORES
./stream-bin > \$LOG_FILE 2>&1
echo \$? > ~/test-exit-status" > stream

chmod +x stream
```

The final step, is to put the script `stream`, the binary file `stream-bin` and the `pts-install.xml` file under the path target system path `$HOME/.phoronix-test-suite/installed-tests/pts/stream-1.1.0`

2.3.3 Cross compilation of Cachebench 1.0.0

For an automated cross compilation of the test Cachebench, I created the scripts *script-cachebench.sh* and *install-cachebench.sh*. Following is described the process of the execution of the cross compilation using these scripts.

First step, I downloaded the test using the command on the target board:

```
$ phoronix-test-suite download-test-files pts/cachebench
```

The following *script-cachebench.sh* is a simple script that unzip the test archive and downloads the test sources using the url inside the file *download.xml*. After, it launches the script *install-cachebench.sh* for the script setup and *script-pts-xml.sh* to create a customized *pts-install.xml* for CacheBench.

The script *script-cachebench.sh* is:

```
######################################################################################
#   CACHEBENCH-1.0.0      #
######################################################################################
#This script run the download of the source of the Phoronix test Cachebench-1.0.0
#and the cross compilation to functioning on a ARM linux architecture.
#It require the test files (that can be downloaded on the target by the Phoronix Suit
#by the command "phoronix-test-suite download-test-files pts/cachebench") in the
#same folder of this script.

#!/bin/bash

unzip cachebench-1.0.0.zip
wget $(xpath -e "//URL" downloads.xml 2>/dev/null | \
sed -e ’s/<\/*URL>//g’ -e ’s/<URL>//g’ -e ’s/<\/URL>/ /g’)
sudo ./install-cachebench.sh
sudo ./script-pts-xml.sh cachebench-1.0.0
```

This script launches the second part of setup, that is the *install-cachebench.sh*, which source is below:

```
######################################################################################
#   INSTALL CACHEBENCH-1.0.0                                                         #
######################################################################################
#This script run the cross-compilation of the Cachebench test.

#!/bin/sh

tar -zxvf llcbench.tar.gz
cd llcbench/

sudo cp Makefile Makefile2
sudo echo "CC = arm-linux-gnueabi-gcc" > Makefile2
#sudo echo "LIBS= -lrt" > Makefile2
#sudo echo "CC= arm-linux-gnueabi-gcc" >> Makefile2
sudo cat Makefile >> Makefile2
sudo mv Makefile2 Makefile

make linux-mpich
make cache-bench

echo $? > ˜/install-exit-status
cd ..

echo "#!/bin/sh
cd llcbench/cachebench/
./cachebench \$@ > \$LOG_FILE" > cachebench

chmod +x cachebench
```

The final step is to put the directory `llcbench`, the file `cachebench`, and the file `pts-install.xml` under the path target system path `$HOME/.phoronix-test-suite/installed-tests/pts/cachebench-1.0.0`

2.3.4 Cross compilation of Encode-mp3 1.4.0

To cross-compile the test Encode-mp3, a simple automated script as the previous is not enough. This test needs to use the system library of Linux *GLIBC* and then the cross compilation with *gcc-arm-linux-gnueabi* can not function correctly if the target (or the cross compiler) has a version of *GLIBC* different from the version present on the target. For example, in the first try I used a pc with Ubuntu 12.04 that uses the last version *GLIBC_2.15*, but the target system uses the older version *GLIBC_2.11*. The solution is to cross compile the test and include the GLIBC used by the cross-compiler into the final program. For this purpose I used the Sourcery Code Bench program that permits this work with an useful tool to remote debug on the target machine, without the need of deploy the program on the target to run.

After the cross compilation using Sourcery Code Bench, I only needed to modify the *lame* script as follows:

```
 #!/bin/sh

./lame-3.99.3/Debug/encode-mp3 -h ../timed-audio-encode-1.0.0/pts-trondheim.wav /dev/null 2>&1

echo $? > ~/test-exit-status
```

and then the test can correctly run with the command

```
$ phoronix-test-suite batch-run encode-mp3
```

Cross compilation using Sourcery Code Bench

To cross-compile the *Encode-mp3* test, I needed to open Sourcery Code Bench and create a new C project, using the file of the test (I worked only the extracted sub-folder *lame-3.99.3*).

Then I set the following ”building options”:

- I enabled the *No shared libraries (-static)* under *C/C++ Build* → *Settings*, tab *Tool Settings*, item *Sourcery CodeBench C Linker* and sub-item *General*
- I added the reference for the math header for the linker, under *C/C++ Build* → *Settings*, tab *Tool Settings*, item *Sourcery CodeBench C Linker* and sub-item *Libraries*, with the *Add* button and typing ”*m*”

Then I *Apply* and press *OK*.

Before run the compilation, I needed to delete some files and folder useless for my purposes; particularly:

- the folder `lame-3.99.3/misc`
- the folder `lame-3.99.3/mac`

- the file `lame-3.99.3/frontend/mp3rtp.c`
- the file `lame-3.99.3/frontend/mp3x.c`
- the file `lame-3.99.3/frontend/gpkplotting.c`
- the file `lame-3.99.3/frontend/gpkplotting.h`
- the file `lame-3.99.3/frontend/gtkanal.c`
- the file `lame-3.99.3/frontend/gtkanal.h`
- the folder `lame-3.99.3/DLL`

and then I run the first building of the project, using the menu *Project* → *Build Project*.

The next step was to correct the build errors in these ways:

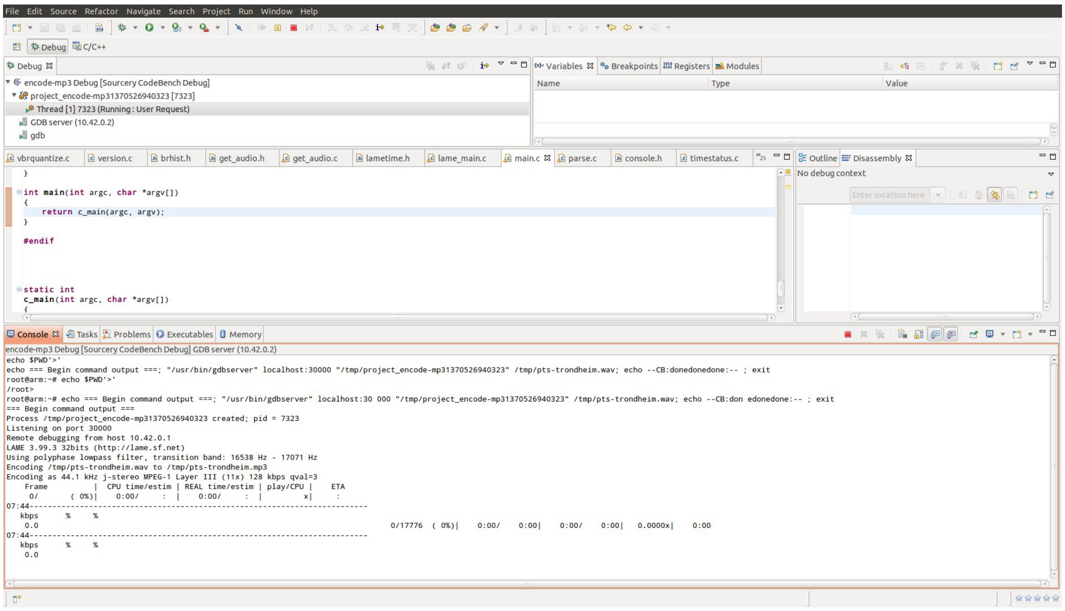
- `"error: lame.h: No such file or directory"` → I changed the include references of the file `lame.h` with `lame-3.99.3/include/lame.h`
- `"error: machine.h: No such file or directory"` (or some other *.h files) → I changed the reference for the files indicated, located under `lame-3.99.3/libmp3lame`
- `"error: unknown type name 'uint16_t'"` (or some other error for standard integer types) → I added `♯include <stdint.h>` at the top of the files with the same error
- `"error: unknown type name 'ieee754_float32_t'"` → I changed this data type into `float` type
- `"error: 'INT_MAX' undeclared"` (or some other similar errors) → I deleted the `♯ifdef` structure before and after the declaration of the header `♯include <limits.h>`
- `"error: gtk/gtk.h: No such file or directory"` → I deleted the declaration `♯include <gtk/gtk.h>` from the top of the files
- `"error: unknown type name 'FILE'"` → I added `♯include <stdio.h>` at the top of the files with this error

Then I could did the final build of the cross compiled test.

Remote debugging using Sourcery Code Bench

To control the correctly functioning of the cross-compiled test on the target board, I can use a comfortable tool of this IDE, that is the *remote debug*. This tool creates an remote connection to the target board and then creates a remote copy of the program, that will be runs on the target system, printing the output messages on the local output window.

To create a new ”Debug Configuration” I used the menu *Run* → *Debug Configurations* then the *Sourcery CodeBench Debug* item on the left, and then the *New* button on the top of the window. Using the ”*Search Project...*” I could select the project to debug by the *Program Selection* window that was opened. By the button *New* on the bottom of the window, I created a remote connection to the target: it was opened the *New Connection* window where I could set the ssh parameters, and other remote parameters in the next windows. Once ended the ssh connection settings, I could modify the others parameters in the main *Debug Configuration* window: under the tab *Debugger* I set other informations for debugging on the target, such as the *Remote GDB* server path (on my target, the *gdbserver* is under `/usr/bin/`). Then I could *Apply* the configuration.



```
encode-mp3 Debug [Sourcery CodeBench Debug]
  project_encode-mp31370526940323 [7323]
    Thread [1] 7323 (Running : User Request)
  GDB server (10.42.0.2)
  gdb
```

```
}

int main(int argc, char *argv[])
{
    return c_main(argc, argv);
}

#endif



static int
c_main(int argc, char *argv[])
{
```

```
encode-mp3 Debug [Sourcery CodeBench Debug] GDB server (10.42.0.2)
echo $PWD'>'
echo === Begin command output ===; "/usr/bin/gdbserver" localhost:30000 "/tmp/project_encode-mp31370526940323" /tmp/pts-trondheim.wav; echo --CB:donedonedone:-- ; exit
root@arm:~# echo $PWD'>'
/root>
root@arm:~# echo === Begin command output ===; "/usr/bin/gdbserver" localhost:30 000 "/tmp/project_encode-mp31370526940323" /tmp/pts-trondheim.wav; echo --CB:don edonedone:-- ; exit
=== Begin command output ===
Process /tmp/project_encode-mp31370526940323 created; pid = 7323
Listening on port 30000
Remote debugging from host 10.42.0.1
LAME 3.99.3 32bits (http://lame.sf.net)
Using polyphase lowpass filter, transition band: 16538 Hz - 17071 Hz
Encoding /tmp/pts-trondheim.wav to /tmp/pts-trondheim.mp3
Encoding as 44.1 kHz j-stereo MPEG-1 Layer III (11x) 128 kbps qval=3
    Frame          |  CPU time/estim | REAL time/estim | play/CPU |    ETA
     0/       ( 0%)|    0:00/     :  |    0:00/     :  |         x|     :
07:44----------------------------------------------------------------------
   kbps      %     %
    0.0                                                                  0/17776  ( 0%)|    0:00/    0:00|    0:00/    0:00|  0.0000x|    0:00
07:44----------------------------------------------------------------------
   kbps      %     %
    0.0
```

Figure 2.2: remote debug of Encode-mp3

Finally, I copied the needed file `timed-audio-encode-1.0.0/pts-trondheim.wav` under the `/tmp` folder on the target, I added the line `/tmp/pts-trondheim.wav`

into the *Arguments* tab of my configuration, and I launched the Debug.
The console window in the image shows the results of the *Lame* process in the remote target system (target related to the image is the Freescale i.MX53 QSB).

Chapter 3

LXC - LinuX Containers

LXC is a linux program that provide an operating system-level virtualization method for running multiple isolated server installs (containers) on a single control host. LXC does not provide a virtual machine, but rather provides a virtual environment that has its own process and network space. It is similar to a chroot, but offers much more isolation.

The virtual systems managed are called *linux containers* and they run using the kernel of the system onto runs the LXC program, sharing the kernel between containers and base system. Moreover, each single container runs onto a single system process, that can be managed by the base system like a normal process. These two characteristics define a very light virtualization, where each container needs few space and resources and it does not affect too much on the system load.

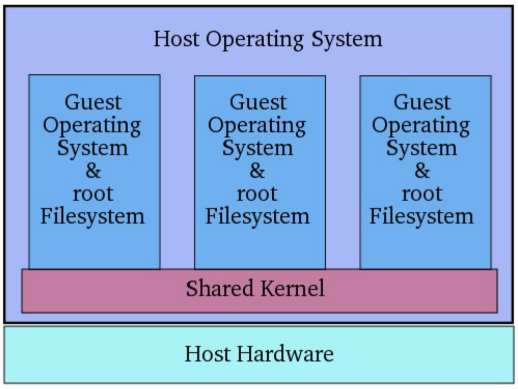

Host Operating System
Guest Operating System & root Filesystem
Guest Operating System & root Filesystem
Guest Operating System & root Filesystem
Shared Kernel
Host Hardware

Figure 3.1: lxc virtualization

The virtualization based on the same kernel, instead that on an hypervisor or a complex software of management, like software for virtual machines where the software virtualizes all the resources and the microprocessor for each machine, is more flexible and light. This permits more evolution in embedded systems where the amount of system load must be the least possible and the more efficient, based on the limited possibility of the hardware present on every single machine: a software for virtual machines requires a system with much memory and microprocessor capabilities, indeed is usually used in server machines where the resources are so many.

The virtualization based on a shared kernel has the defect of being able to create only virtual systems that use the same kernel. However this is not a major lack, because the sharing of the kernel permits to able each operating system virtualized to communicate directly with the real kernel and the real hardware, without passing through a software of resource virtualization. So each virtual system can known the real characteristics of the physical system, without any problems arising from the virtualization of resources that often does not allow the programmer to known where are the problems between virtualization software and virtual machine. Furthermore this software can only implement a mechanism to share the kernel functionalities between virtual systems and few other operations, and then it can be much lighter than the others virtualization software.

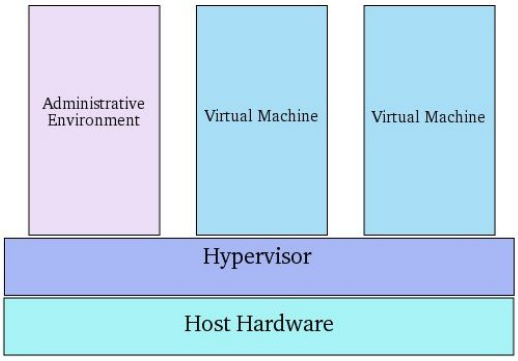


Figure 3.2: virtual machines virtualization

3.1 LXC setup

The LXC program needs a kernel with some specific parameters to function. The required parameters for every version of the program, are checked by the command *lxc-checkconfig*. This command does an examination of the kernel and prints the parameters existing and missing in the kernel. In case of an embedded system where the kernel is compiled from time to time, there is a method to examine the kernel parameters before deploy the system on the target. Particularly, after the compilation described in the above section 2.1.1 (”compiling linux kernel 3.8.6”), it is created a file `.config` into the folder `imx-devel/KERNEL`, that is a configuration file containing all the parameters configuration of the compiled kernel. To check directly this file instead of run and check the complete system, can be done by configuring the system variable *CONFIG*; in my compiled kernel the operation and results are the following:

```
$ CONFIG=.config /usr/bin/lxc-checkconfig

 --- Namespaces ---
Namespaces: enabled
Utsname namespace: enabled
Ipc namespace: enabled
Pid namespace: enabled
User namespace: missing
Network namespace: enabled
Multiple /dev/pts instances: enabled
--- Control groups ---
Cgroup: enabled
Cgroup clone_children flag: enabled
Cgroup device: enabled
Cgroup sched: enabled
Cgroup cpu account: enabled
Cgroup memory controller: missing
Cgroup cpuset: enabled
--- Misc ---
Veth pair device: enabled
Macvlan: enabled
Vlan: enabled
File capabilities: enabled
```

From this results, it is possible to add the missing configurations with a new compilation of the kernel where the relative parameters will be set correctly.

(The code of this script is in the *Appendix B*)

3.1.1 Parameter Control Groups

In the results appears the section *Control groups*, also identified as *Cgroups*. *Cgroups* allows to allocate resources—such as CPU time, system memory, network bandwidth, or combinations of these resources—among user-defined groups of processes running on a system. It is possible monitor the cgroups to configure, deny cgroups access to certain resources, and even reconfigure the cgroups dynamically on a running system. The cgconfig (control group config) service can be configured to start up at boot time and reestablish your predefined cgroups, thus making them persistent across reboots. By using cgroups, system administrators gain fine-grained control over allocating, prioritizing, denying, managing, and monitoring system resources. Hardware resources can be smartly divided up among tasks and users, increasing overall efficiency.

3.1.2 Parameter User Namespace

If the lxc-checkconfig result on the compiled kernel has the row

```
User namespace: missing
```

the problem is the the kernel parameter *USER_NS* that is not setted, relative to the *User Namespace* of the system. The *User namespaces* in Linux has to make it safe for an unprivileged user to unshare namespaces and provide separate limits and accounting for userids in different namespaces. To enable this function into the kernel it is necessary to enable the item *USER_NS* in the *make_menuconfig*, that it can be enable only if some others item are setted for the correctly functioning of the kernel. The needed settings are listed below:

```
NET_9P = n
9P_FS = n
AFS_FS = n
CEPH_FS = n
CIFS = n
CODA_FS = n
GFS2_FS = n
NCP_FS = n
NFSD = n
NFS_FS = n
OCFS2_FS = n
XFS_FS = n
```

After these configurations the *USER_NS* can be enabled and the problem of lxc-checkconfig onto the kernel is solved.

3.1.3 Parameter Memory Resource Controller for Control Groups

If the lxc-checkconfig result on the compiled kernel has the row

```
Cgroup memory controller: missing
```

the problem is the unsetted kernel parameter *CONFIG_CGROUP_MEM_RES_CTLR* but this item is not present in my compiled kernel, because from the kernel linux version 3.6 the parameter has changed into *CONFIG_MEMCG*. Then, I must set this new parameter in the *make_menuconfig* but is not enough because the lxc-checkconfig controls the presence of the older module *CONFIG_CGROUP_MEM_RES_CTLR*. To solve this problem, I must upgrade the lxc program (currently at version 0.7.5) to the lastest version *0.9.0*: that version checks if the option *CONFIG_CGROUP_MEM_RES_CTLR* is enabled (for older versions of kernel), or if it is enabled the newer option *CONFIG_MEMCG*, as described in these rows of the source file *lxc-checkconfig.in*:

```
...
echo -n "Cgroup memory controller: "
if [ $KVER_MAJOR -ge 3 -a $KVER_MINOR -ge 6 ]; then
    is_enabled CONFIG_MEMCG
else
    is_enabled CONFIG_CGROUP_MEM_RES_CTLR
fi
...
```

With the option *CONFIG_MEMCG* enabled and the new version of lxc, the result of *lxc-checkconfig* become the following:

```
$ CONFIG=.config /usr/bin/lxc-checkconfig

--- Namespaces ---
Namespaces: enabled
Utsname namespace: enabled
Ipc namespace: enabled
Pid namespace: enabled
User namespace: enabled
Network namespace: enabled
Multiple /dev/pts instances: enabled
--- Control groups ---
Cgroup: enabled
Cgroup namespace: required
Cgroup device: enabled
Cgroup sched: enabled
Cgroup cpu account: enabled
Cgroup memory controller: enabled
Cgroup cpuset: enabled
--- Misc ---
Veth pair device: enabled
Macvlan: enabled
Vlan: enabled
File capabilities: enabled
```

where the *Cgroup memory controller* option is finally enabled and it appears a new item *Cgroup namespace* that is required for the right functioning of the lxc version 0.9.0.

3.1.4 Parameter Cgroup Namespace

The new version of lxc-checkconfig highlights a particular problem

```
Cgroup namespace: required
```

that define the need of the *Cgroup namespace* into the kernel to correctly run the program. The *ns_cgroup* was replaced by a compatibility flag *clone_children*, where a newly created cgroup will copy the parent cgroup values: the userspace has to manually create a cgroup and add a task to the 'tasks' file. LXC will run without warning even if full cgroup isolation is unavailable, but if *clone_children* were not available (by the kernel) then it would fail to start the containers.

3.2 LXC execution

The LXC program manages the creation of new virtual operating systems (called "containers") from a list of available GNU/Linux distribution, the execution and termination of the containers and it provides the resource management through the control groups, aka process containers, and resource isolation through the namespaces. Using an apposite configuration file for the container parameters, the LXC program creates a container on which run the desired GNU/Linux distribution, and with the appropriate commands allows the start and stop of the container, the batch execution of programs into it, the chroot functionalities and the destroy of the container. The creation of a container is based on a ”template” that describes all the operation to do to build the virtual system and mount the needed parts from the base system. In some embedded contexts, not all the functionalities are available, depending on which tools has the system and on how is made every container.

3.2.1 Creation of a linux container

The creation of a container with LXC is made using the command *lxc-create*. This is a complex script that needs at least three parameters: the name of the container, the configuration file and the name of the template to be used to build the container. The execution of this script creates the folder for the container, where put the filesystem, the configuration file where set the environment variables (like the mount points) respect to base system for the startup of the container, then it reads the template indicated and executes it: the template provide the operations needed to obtain the desired virtual linux distribution.

(The code of this script is in the *Appendix C*)

Container configuration file

Each container needs of some information to startup correctly on the system, because these configurations indicate how the container can interact with the base system and the resources. In a configuration file it is possible to indicate the number of tty of the new system, the network parameters of the container (like mac address and IP address, physical interface to use as physical network device, type of sharing the network device) and the branch of the system to be mounted into the virtual system.

Container templates

The container templates are simple bash scripts that permit to create a new virtual operating system inside a linux container. They are called and executed by the command *lxc-create* and each template refers to a specific GNU/Linux distribution. The first step of the templates is to get the filesystem of the relative linux distribution: in some cases this filesystem is automatic downloaded from the official distribution repository online, in other cases be used the existing sources on the base system. After, the filesystem tree is copied into the folder reserved for the container and are created the various system files under `/dev` folder (like the *null* file, the *tty* character files and the *console* file), and is created the default user to login into the system, with relative group and password settings. Then the last configurations are the startup scripts under `/etc/init.d` and the startup configuration on `/etc/inittab`, the mounting of the required branches of the system into the container and some others configurations and scripts for the correct operating.

3.2.2 Container start and stop

Once created, a container can start with the simple command *lxc-create*, followed by the name of the container:

```
$ lxc-create -n cont01
```

On startup, the program verifies the configuration settings of the referred container, to create the virtual references of the physical resources, and run the startup scripts of the virtual system that run and print the login interface to enter in the system. From this point, the container is fully operating on the machine in parallel at the base system. The container process can be viewed into the system task (like using the *ps* command): the task is shown in the form `lxc-start -n cont01`.

Stopping the container can be done in the canonical mode, using the system command *halt* inside the container (or others deriving from this), or using the command *lxc-kill* into the base system: this command kill the task of the container and close the relative instance. In case of need, the process can be terminated directly killing the process with the system command `kill <PID>` on the base system.

3.3 LXC containers

After all the preparations, some containers can be easily created; and if all the tools are present on the system, all the process can be done also on an embedded operating system with few set of system commands. The following sections are an analysis of containers Debian based and Yocto based, running on the target embedded system *mel6-lxcbench*.

3.3.1 Debian container

An example of container can be done with the Debian distribution, a much famous distribution and easily to configure, with a big community support.

Debian configuration file

For this distribution, the configuration file is the following:

```
lxc.network.type = macvlan
lxc.network.flags = up
lxc.network.link = eth0
lxc.network.name = eth0
lxc.network.hwaddr = 02:00:00:00:00:72

lxc.network.ipv4 = 10.42.0.10/24
lxc.network.ipv4.gateway = 10.42.0.1

lxc.rootfs = /var/lib/lxc/CN22/rootfs
lxc.tty = 4
lxc.pts = 1024
lxc.utsname = CN22
```

where the *macvlan* network type define a configuration where the containers simultaneously active on the same network device can communicate between them without let out packets in the network. The `lxc.network.link` parameter define the network device to use, and the `lxc.network.name` define the name of network device used on the virtual system. Then, `lxc.network.hwaddr`, `lxc.network.ipv4` and `lxc.network.ipv4.gateway` define the ethernet parameter of the virtual network device. Finally, `lxc.rootfs` define the absolute path of the virtual root inside the base system, `lxc.tty` the number of default tty of the system, `lxc.pts` the number of default created remote terminals and `lxc.utsname` the name of the virtual host.

Debian filesystem

The template *lxc-debian* that creates this container, needs to download the Debian filesystem (version for arm target) from the official repository online at the url `http://cdn.debian.net/debian`. This process is usually done by the system tool *Deboostrap*. *Debootstrap* is a tool which will install a GNU/Linux base system into a subdirectory of another, already installed system, just accessing to the repository of the distribution.

In the embedded operating system *mel6-lxcbench* this tool is not present, as often happens in embedded operating systems due to the efficiency and lightness of the system. For this reason create a Debian system onto this system can be difficulty, because needs some changes into the Debian template of lxc, but the fast method to get a Debian container onto this system is to use another embedded system with debootstrap installed: I used the embedded Debian, based on kernel 3.8.6, that I used for the first tests of this work. With this Debian embedded system and using the same version of LXC (version 0.9.0), I could create a lxc Debian container: once created, it can easily compressed into an archive and copied on the final target where it correctly start.

3.3.2 mel6-lxcbench container

After some tries on the Debian container, I could focus my work on the creation of a container based on the Yocto GNU/Linux distribution *mel6-lxcbench*. The result is not optimal, because a standard container get the login on the same terminal where it is launched by the command *lxc-start*; instead, this container does not show the prompt login on the same launch terminal, but needs a login via ssh (that correctly functioning). For the rest, the container has all the characteristics of a standard container. Moreover, it has all the good characteristics of the base embedded system, for example it is sure that all the command of the base system are present on this container, and it has all the features of the base system (and then it is easily to work on it).

mel6-lxcbench configuration file

The configuration file used for this container was the following:

```
lxc.rootfs = /var/lib/lxc/cont01/rootfs
lxc.utsname = cont01
lxc.tty = 4
lxc.pts = 1024

# mounts point
lxc.mount.entry = proc proc proc nodev,noexec,nosuid 0 0
lxc.mount.entry = sysfs sys sysfs defaults  0 0
lxc.mount.entry = /lib lib none ro,bind 0 0
lxc.mount.entry = /usr/lib usr/lib none ro,bind 0 0
```

It does not differ too much from the configuration file of the Debian distribution. Interesting are the mount points, necessary for the right functioning of this container.

mel6-lxcbench template

Then, I needed to create a template for a lxc container and I started from the *lxc-busybox* template, for his simplicity and lightness to easily change it and add some useful features. The filesystem for this container is downloaded from the official repository of the project LXCBENCH. After copying the source and built the folder tree under the folder reserved for the container, the template creates the special files, that are the various *tty* character files, the block file *ram0*, the character files *null* and *urandom*. Then it proceed to the creation of the default root user with its group. Then is writed the file `/etc/init.d/rcS`, that is the startup script of the system. Next are the configuration of hostname and fstab, and is created the `/etc/inittab` file with the configurations that the system reads on startup to known the priority processes, script and terminal for the system. This inittab is:

```
id:3:initdefault:
si::sysinit:/etc/init.d/rcS
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
z6:6:respawn:/sbin/sulogin
c1:12345:respawn:/sbin/getty 38400 tty1 linux
c2:12345:respawn:/sbin/getty 38400 tty2 linux
c3:12345:respawn:/sbin/getty 38400 tty3 linux
c4:12345:respawn:/sbin/getty 38400 tty4 linux
p6::ctrlaltdel:/sbin/init 6
p0::powerfail:/sbin/init 0
```

where are mainly defined the first process initdefault with its priority, the scripts that need to be executed on startup and the console terminal active on the system. Finally, the templates contains the instructions to create some script to set system configurations, like the network scripts under `/usr/share/udhcpc`, password settings and few other system options.

(The source code of the template `lxc-lxcbench` is in the *Appendix D*)

Chapter 4

Tests

The final part of my work is the collecting results from the testing hardware. The final systems have the embedded operating system *mel6-lxcbench* and run some containers based on the same system.

First results are collected to understand if multiple containers can optimize the use of the hardware, reserving a core for each container active (for multi-core hardware). Then the results compare the various responses of the system when are running two containers at the same times, with the same Phoronix test active. For this reason there is a section with tests done for each PTS test used. The wanted result is to obtain that, for example, that the PandaBoard with dual-core microprocessor obtains the same result if it is running one single test on one container, whether if are running two same tests on two simultaneously running containers.

The second part of results collection is about the performance obtained from a Phoronix test on a container when there is the same or a different test running on a parallel container on the same board. For this reason, the results are divided into four section, one for each test used and the tables of results show the corresponding results of the test on the row, when on another container is running the test indicated on the column.

4.1 Performance on multiple containers

For this measures, I run the same test on multiple containers and on the base system, to verify if the LXC container virtualization divides the work of multiple containers onto the various cores available on the architecture. The wanted result is to have a core of the microprocessor reserved for a single container, if the number of cores are enough respect the number of active containers. For the i.MX53 (single core) and the PandaBoard (dual core), the first measures are done on the system with only one container active: the test runs on the system while the container is in idle state, and viceversa it runs on the container while the system in idle state. After these, the measures are done on two containers active simultaneously and running the same Phoronix test.

Following tables have on the first column the results of each test executed on the base operating system, while one container is running in idle state (it is doing nothing); on the second, the results of each test executed on a container while the base operating system is in idle state. The third column is for the circumstance in which two containers are active and running the same test at the same time.

4.1.1 Results of Dbench 1.0.0

The Dbench test is executed using the command:

```
$ phoronix-test-suite batch-run dbench
```

and following are the results of this test:

	host + guest(IDLE) [MB/s]	guest + host(IDLE) [MB/s]	guest+guest [MB/s]	
i.MX53	9.45	8.57	1.94	1.84
	3.39	3.11	0.32	0.34
	0.77	0.70	0.20	0.21
	0.38	0.42	0.19	0.30
	0.49	0.43	0.34	0.83
	0.50	0.25	0.30	0.31
PandaBoard	10.01	11.01	2.95	2.98
	3.91	3.39	1.20	1.20
	2.06	1.87	0.48	0.81
	1.39	1.49	0.64	1.22
	1.52	1.28	0.40	0.72
	0.89	0.94	0.61	0.41

The result on the first column and second are similar but not the same. It probably depends on the long time of execution of the test (about eight hours), that can affect the accuracy of measurements. However they are not too different: that means that execution on the base system or on the container requires the same system workload.

When two containers are running, results are quite lower. This may depend on the operating mode of the test, because it creates some virtual instances to stress the system, like many concurrent clients or applications. Then with multiple containers, there are simultaneously Dbench tests that try to stress at maximum each system: probably the LXC program limits the resources of each container to prevent problems between systems.

However, it is important to highlight that the results between concurrent containers are about the same: that means that a system has the same execution priority than each other.

4.1.2 Results of Stream 1.1.0

The Stream test is executed using the command:

```
$ phoronix-test-suite batch-run stream
```

and following are the results of this test:

	host + guest(IDLE) [MB/s]	guest + host(IDLE) [MB/s]	guest+guest [MB/s]	
i.MX53	460.13	458.09	374.70	228.11
	320.71	339.56	288.18	175.24
	326.51	300.69	296.68	152.07
	348.14	335.71	325.48	170.55
PandaBoard	617.94	615.94	616.21	611.12
	594.68	593.06	591.36	588.04
	628.37	629.50	628.54	630.28
	422.08	422.36	422.06	421.26

For this test the results on the first column and on the second are very similar. Then it can be confirmed that an execution on the base system or on a container for this test has the same working.

The situation with two containers active highlights that in an architecture with multiple core, a core can be reserved for a specific container: for example, the PandaBoard results in two containers mode are the same that in single container mode.

4.1.3 Results of Cachebench 1.0.0

The test Cachebench is executed using the command:

```
$ phoronix-test-suite batch-run cachebench
```

and following are the results of this test:

	host + guest(IDLE) [MB/s]	guest + host(IDLE) [MB/s]	guest+guest [MB/s]	
i.MX53	85.05	85.03	57.30	41.14
	85.10	85.09	67.19	41.01
	85.10	85.09	82.39	41.08
PandaBoard	1691.14	1691.13	1693.89	1687.57
	1691.14	1690.99	1693.90	1687.45
	1692.12	1691.00	1694.00	1687.05

Also this test, in single container mode, confirms the equal working of a container respect to the base system.

With two container running at the same time, the PandaBoard results on a container are slightly higher than the single container condition, meanwhile results on the second container are slightly lower. The difference between results is little and can anyway confirm the idea that each container work on a core reserved for his own work.

4.1.4 Results of Encode-mp3 1.4.0

The test Encode-mp3 is executed using the command:

```
$ phoronix-test-suite batch-run encode-mp3
```

and following are the results of this test:

	host + guest(IDLE) [s]	guest + host(IDLE) [s]	guest+guest [s]	
i.MX53	995.20	986.79	2024.58	2019.68
PandaBoard	145.75	144.59	145.37	145.84

The analysis of this results is more complex than previous tests, because it produces only one measure for every run (the measure of encode a wave audio file into a compressed mp3 file). But it can confirm either the execution equality between base system and container, than the division of work on the various core, as reserving a single core for a single container.

4.2 Comparison with multiple tests active

The following measures aim to know the influence that has a single test running on a container, while another test is running on another container active (the test can also be the same). These results can be important to known if some concurrent containers, working on some jobs, are totally independent from the running of other containers.

4.2.1 Results of Freescale i.MX53 QSB

Following the results on the board Freescale i.MX53 QSB, with two containers running at the same time.

		container2			
i.MX53		dbench	stream	cachebench	encode-mp3
container1	dbench [MB/s]	1.84	9.16	6.41	6.45
		0.34	3.08	2.81	2.53
		0.21	0.68	0.83	0.69
		0.30	0.48	0.36	0.40
		0.83	0.43	0.49	0.43
		0.31	0.32	0.32	0.47
	stream [MB/s]	399.29	228.11	227.74	240.91
		285.62	175.24	171.76	179.88
		256.23	152.07	158.89	167.47
		282.52	170.55	171.69	179.76
	cachebench [MB/s]	51.75	78.21	41.14	40.97
		64.93	82.38	41.01	40.91
		62.91	82.38	41.08	41.04
	encode-mp3 [s]	3420.67	1010.76	1199.53	2019.68

The table shows the great influence of Dbench test executed in the second container, particularly while on the first are executed the test Encode-mp3 and Dbench. The Stream test works without problems from the test that is running on the second container: indeed when on the second container is running the Dbench test, the Stream results are higher. The Cachebench test has the same results with every concurrent tests on the second containers, aside when on second container is running the Stream test that for his lightness put up the CacheBench results: so this test does not affects too much from other containers operations. The encode-mp3 has worst perfomance only when the concurrent test is Dbench.

From these values it can say that only jobs with huge workload (like the Dbench) influence the other containers, that in other cases are not too much affected from the other containers operations.

4.2.2 Results of TI PandaBoard ES

Following the results on the board TI PandaBoard ES, with two containers running at the same time.

		container2			
	PandaBoard	dbench	stream	cachebench	encode-mp3
container1	dbench [MB/s]	2.98	10.06	12.18	9.53
		1.20	3.85	4.76	2.58
		0.81	2.19	2.03	2.05
		1.22	1.52	1.55	1.51
		0.72	1.54	1.62	1.66
		0.41	0.95	0.94	0.96
	stream [MB/s]	616.31	611.12	618.52	594.14
		598.12	588.04	594.03	573.97
		625.49	630.28	629.75	586.79
		421.32	421.26	422.67	408.26
	cachebench [MB/s]	1683.17	1687.11	1687.57	1688.74
		1685.75	1690.74	1687.45	1690.62
		1686.68	1690.78	1687.05	1690.68
	encode-mp3 [s]	181.02	146.16	149.89	145.84

In this dual-core board, the only results that can be highlighted are the Dbench test results and Encode-mp3 test result on the first container, while the Dbench test is running on second container. The influence of Dbench on the two cases described may be for the network operating of the Dbench test, that can affect the physical network device for every virtual systems running.

Other test values are rather indipendent from the test that is running on the second container, due the multi-core architecture and the LXC capability of reserving a core for a single container.

Chapter 5

Conclusions

At the end of this work, after all the measurements, I reached my objectives. The tables of result show that almost in all situations, the LXC virtualization permits a good division of the amount of load between the various cores of the system. This is highlighted when the results of a single test executed in a multi-core architecture reach the same value of many tests, executed on different containers, where the number of containers active (with test running) is the same of the number of cores of the architecture. This is a new point of view for virtualization of operating systems, because in some embedded system implementation these results permits to have multiple virtual systems without too many increasing of hardware and potentiality of components. Moreover, this virtualization don't needs too many resources to run the virtualization management program. For this, it is possible to reserve a single core for a single container, because the virtualization program needs few resources to run, and dedicate the system resources almost completely to the virtual systems active.

5.1 Better exploitation of resources

These results are very important for the current increasing of power and potentiality of electronic components. In these years where the importance of production was focused on the speed, dimension and power comsumption of hardware, and consequentely on the reduction of costs, the main ideas was to change components with newer, rather then optimize the uses of the hardware. And were also the easiest ideas, that did not need too researches and optimization developments. This virtualization permits to optimize the existing hardware, without changes of components. Exploit the hardware architectures is the best way to buy only the needed hardware, reducing costs for new electronic and powerful components, and increasing system activity at best.

5.2 Future implementations

Future implementations of this virtualization can be multiple. Benefits of this virtualization can be derived by companies that need new systems active without increasing hardware, cost of components or work of maintenance. Furthermore, this virtualization is more faster than a virtualization based on virtual machines, due its characteristic of virtualization kernel-based; for this, it is better for embedded systems that usually require the less time of management and to reserve the more activity to run the virtual systems.

An example, for an optimal final implementation of this virtualization may be a car computer that implements the general car controller system (with a container) and also the Graphical User Interface (GUI) that allows car passengers to control the interior comfort system, driver information systems and others services provided by the car (with another container). A use like this explains well the importance of fast startup and switching time of virtual operating systems. It highlights also the importance of load on the hardware of each system, that can be the lower possible and it must influence as little as possible the other systems active.

Chapter 6

Glossary

Chroot

Chroot is the process of changing of the apparent disk root directory (and the current running process and its children) to another root directory. When you change root to another directory you cannot access files and commands outside that directory. This directory is called a *chroot jail*. Changing root is commonly done for system maintenance, such as reinstalling the bootloader or resetting a forgotten password.

Embedded system

An embedded system is a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. Embedded systems control many devices in common use today. Embedded systems contain processing cores that are either microcontrollers, or digital signal processors (DSP). The key characteristic, however, is being dedicated to handle a particular task. Since the embedded system is dedicated to specific tasks, design engineers can optimize it to reduce the size and cost of the product and increase the reliability and performance. Physically, embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, and largely complex systems like hybrid vehicles and avionics. Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure.

GNU's Not Unix

GNU is a Unix-like operating system developed by the GNU Project. It is composed wholly of free software. It is based on the GNU Hurd kernel and is intended to be a "complete Unix-compatible software system". GNU is a recursive acronym for "GNU's Not Unix!", chosen because GNU's design is Unix-like, but differs from Unix by being free software and containing no Unix code. Development of GNU was initiated by Richard Stallman in 1983 and was the original focus of the Free Software Foundation (FSF), but no stable release of GNU yet exists as of May 2013. Non-GNU kernels, most famously the Linux kernel, can also be used with GNU.

In-Vehicle Infotainment

The "In-Vehicle Infotainment" (IVI) is a collection of hardware devices installed into automobiles, or other forms of transportation, to provide audio and/or audio/visual entertainment, as well as automotive navigation systems. This includes playing media such as CDs, DVDs, TV, USB and other optional surround sound, or DSP systems. Also increasingly common in IVI installs are the incorporation of video game consoles into the vehicle. In-Vehicle Infotainment has been become more widely available due to reduced costs of devices such as LCD screen, and the reducing cost to the consumer of the converging media playable technologies. A special type of IVI are the Carputers, that are specially adapted computers, designed to operate in a car environment. They can provide many functions, such as video and audio playback, games and Internet connectivity.

Integrated Development Environment

An "Integrated Development Environment" (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, build automation tools and a debugger. Some IDEs contain a compiler and an interpreter, and many IDEs are multi-language A widely used IDE is "Eclipse", an open-source and multi-language IDE, comprising a base workspace and an extensible plug-in system for customizing the environment. It is written mostly in Java and it can be used to develop applications in Java or, by means of various plug-ins, other programming languages including C, C++, PHP, Python, Ruby and others. It is currently maintained by the "Eclipse Foundation", a nonprofit organization annoverating as involved participants IBM, Google, Oracle, AT&T, Cisco and many others.

LinuX Container

A ”Linux container” is a operating system virtualization, similar to the most famous *virtual machine*. It is a virtualizazion based on the software LXC and it is based on the kernel Linux. A linux container has the characteristic of sharing the kernel with the others container running and so it is necessary have virtual system running on the same kernel , that is also the kernel of the base system, and on the same architecture. Another important characteristic is that a linux container runs in into a single system process: then it can be managed from the base system like a normal task and it requires low space and resources.

System on Chip

A ”system on chip” (SoC) is an integrated circuit that integrates all components of a computer or other electronic system into a single chip. The SoC chip can includes processors, main memory, GPU and some common used peripherals. It may contain digital, analog, mixed-signal, and often radio-frequency functions all on a single chip substrate. This type of integrated circuit is commonly used in embedded applications, due the small size that they achieve with the integration of all components. The newer SoC chips are built with powerful processors, capable of running software such as Linux or Windows which are used with various external peripherals.

TTY

The ”tty” identifies under Unix-like systems the serial terminals, each of them identified by a number (like tty1, tty2, tty3..). Each GNU/Linux system has a predefined number of default serial terminals active at startup through the user can interact with the relative operating system. Each terminal is configured with a respective character file under the `/dev` folder, that has the same name of the terminal, created with with major number 5 and minor number 0, usually of mode `0666` and `owner.group root.tty`. A operating system can be completely configured on the number of tty serial terminal needed, on who own the terminal and print on it, and on which physical peripheral refers the terminal (like serial output RS-232, usb or vga).

Virtual Machine

A ”virtual machine” (VM) is a software based, fictive computer. Virtual machines may be based on specifications of a hypothetical computer or emulate the computer architecture and functions of a real world computer.
A virtual machine was originally defined by Popek and Goldberg as ”*an efficient, isolated duplicate of a real machine*”. Current use includes virtual machines which have no direct correspondence to any real hardware. A operating system virtualized inside a virtual machine is completely independent, because it has its own virtual resources: this means that every system can have different kernel and architecture, because the software emulates the required system resources, peripherals and architecture and demand to the physical system the command to get the correctly results to the virtual system.

Yocto project

The Yocto Project is an open source project whose focus is on improving the software development process for embedded Linux distributions. The Yocto Project provides interoperable tools, metadata, and processes that enable the rapid, repeatable development of Linux-based embedded systems. The project was announced by the Linux Foundation in 2010. The Yocto Project has the aim and objective of attempting to improve the lives of developers of customised Linux systems supporting the ARM, MIPS, PowerPC and x86/x86 64 architectures. It is working towards this by focusing on one set of great tools rather than the several sets of mediocre ones that existed when the project was founded. A key part of this is an open source build system, based around the OpenEmbedded architecture, that enables developers to create their own Linux distribution specific to their environment.

Chapter 7

References

Cgroups
https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Resource_Management_Guide/ch01.html

Cross-compilation scripts
http://git.projects.genivi.org/?p=lxcbench.git;a=search;s=Francesco+Castagnotto;st=author

Freescale i.MX53 Quick Start Board
http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=IMX53QSB

Genivi Alliance
http://www.genivi.org/

Genivi Project LXCBENCH
http://projects.genivi.org/lxcbench/about

LXC - Linux Containers
http://linuxcontainers.org/
http://lxc.sourceforge.net
https://wiki.archlinux.org/index.php/Linux_Containers

Pandaboard
http://www.pandaboard.org/

Phoronix Test Suite
http://www.phoronix-test-suite.com
http://openbenchmarking.org/tests/pts

Appendix A

build_kernel.sh

```
#!/bin/bash -e
#
# Copyright (c) 2009-2013 Robert Nelson <robertcnelson@gmail.com>
#
# Permission is hereby granted, free of charge, to any person obtaining a copy
# of this software and associated documentation files (the "Software"), to deal
# in the Software without restriction, including without limitation the rights
# to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in
# all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
# THE SOFTWARE.

DIR=$PWD

mkdir -p ${DIR}/deploy/

function patch_kernel {
cd ${DIR}/KERNEL

export DIR GIT_OPTS
/bin/bash -e ${DIR}/patch.sh || { git add . ; exit 1 ; }

if [ ! "${RUN_BISECT}" ] ; then
git add .
git commit --allow-empty -a -m "${KERNEL_TAG}-${BUILD} patchset"
fi

#Test Patches:
#exit

if [ "${LOCAL_PATCH_DIR}" ] ; then
for i in ${LOCAL_PATCH_DIR}/*.patch ; do patch  -s -p1 < $i ; done
BUILD+=’+’
```

```
fi

cd ${DIR}/
}

function copy_defconfig {
cd ${DIR}/KERNEL/
make ARCH=arm CROSS_COMPILE=${CC} distclean
make ARCH=arm CROSS_COMPILE=${CC} ${config}
cp -v .config ${DIR}/patches/ref_${config}
cp -v ${DIR}/patches/defconfig .config
cd ${DIR}/
}

function make_menuconfig {
cd ${DIR}/KERNEL/
make ARCH=arm CROSS_COMPILE=${CC} menuconfig
cp -v .config ${DIR}/patches/defconfig
cd ${DIR}/
}

function make_kernel {
cd ${DIR}/KERNEL/
echo "-----------------------------"
echo "make -j${CORES} ARCH=arm LOCALVERSION=-${BUILD} CROSS_COMPILE=\"${CC}\"
 ${CONFIG_DEBUG_SECTION} zImage modules"
echo "-----------------------------"
time make -j${CORES} ARCH=arm LOCALVERSION=-${BUILD} CROSS_COMPILE="${CC}"
${CONFIG_DEBUG_SECTION} zImage modules

unset DTBS
cat ${DIR}/KERNEL/arch/arm/Makefile | grep "dtbs:" &> /dev/null && DTBS=1
if [ "x${DTBS}" != "x" ] ; then
echo "make -j${CORES} ARCH=arm LOCALVERSION=-${BUILD} CROSS_COMPILE=\"${CC}\" ${CONFIG_DEBUG_SECTION} dtbs"
time make -j${CORES} ARCH=arm LOCALVERSION=-${BUILD} CROSS_COMPILE="${CC}" ${CONFIG_DEBUG_SECTION} dtbs
ls arch/arm/boot/* | grep dtb || unset DTBS
fi

KERNEL_UTS=$(cat ${DIR}/KERNEL/include/generated/utsrelease.h | awk ’{print $3}’ | sed ’s/\"//g’ )
if [ -f ./arch/arm/boot/zImage ] ; then
cp arch/arm/boot/zImage ${DIR}/deploy/${KERNEL_UTS}.zImage
cp .config ${DIR}/deploy/${KERNEL_UTS}.config
else
echo "-----------------------------"
echo "Error: make zImage modules failed"
exit
fi
cd ${DIR}/
}

function make_modules_pkg {
cd ${DIR}/KERNEL/

echo "-----------------------------"
echo "Building Module Archive"
echo "-----------------------------"

rm -rf ${DIR}/deploy/mod &> /dev/null || true
mkdir -p ${DIR}/deploy/mod
make ARCH=arm CROSS_COMPILE=${CC} modules_install INSTALL_MOD_PATH=${DIR}/deploy/mod
echo "-----------------------------"
echo "Building ${KERNEL_UTS}-modules.tar.gz"
```

```
cd ${DIR}/deploy/mod
tar czf ../${KERNEL_UTS}-modules.tar.gz *
echo "-----------------------------"
cd ${DIR}/
}

function make_firmware_pkg {
cd ${DIR}/KERNEL/

echo "-----------------------------"
echo "Building Firmware Archive"
echo "-----------------------------"

rm -rf ${DIR}/deploy/fir &> /dev/null || true
mkdir -p ${DIR}/deploy/fir
make ARCH=arm CROSS_COMPILE=${CC} firmware_install INSTALL_FW_PATH=${DIR}/deploy/fir
echo "-----------------------------"
echo "Building ${KERNEL_UTS}-firmware.tar.gz"
cd ${DIR}/deploy/fir
tar czf ../${KERNEL_UTS}-firmware.tar.gz *
echo "-----------------------------"
cd ${DIR}/
}

function make_dtbs_pkg {
cd ${DIR}/KERNEL/

echo "-----------------------------"
echo "Building DTBS Archive"
echo "-----------------------------"

rm -rf ${DIR}/deploy/dtbs &> /dev/null || true
mkdir -p ${DIR}/deploy/dtbs
find ./arch/arm/boot/ -iname "*.dtb" -exec cp -v ’{}’ ${DIR}/deploy/dtbs/ \;
cd ${DIR}/deploy/dtbs
echo "-----------------------------"
echo "Building ${KERNEL_UTS}-dtbs.tar.gz"
tar czf ../${KERNEL_UTS}-dtbs.tar.gz *
echo "-----------------------------"
cd ${DIR}/
}

/bin/bash -e ${DIR}/tools/host_det.sh || { exit 1 ; }

if [ ! -f ${DIR}/system.sh ] ; then
cp ${DIR}/system.sh.sample ${DIR}/system.sh
fi

unset CC
unset DEBUG_SECTION
unset LINUX_GIT
unset LOCAL_PATCH_DIR
source ${DIR}/system.sh
/bin/bash -e "${DIR}/scripts/gcc.sh" || { exit 1 ; }
source ${DIR}/.CC
echo "debug: CC=${CC}"

source ${DIR}/version.sh
export LINUX_GIT

unset CONFIG_DEBUG_SECTION
if [ "${DEBUG_SECTION}" ] ; then
```

```
CONFIG_DEBUG_SECTION="CONFIG_DEBUG_SECTION_MISMATCH=y"
fi

#unset FULL_REBUILD
FULL_REBUILD=1
if [ "${FULL_REBUILD}" ] ; then
/bin/bash -e "${DIR}/scripts/git.sh" || { exit 1 ; }

if [ "${RUN_BISECT}" ] ; then
/bin/bash -e "${DIR}/scripts/bisect.sh" || { exit 1 ; }
fi

patch_kernel
copy_defconfig
fi
if [ ! ${AUTO_BUILD} ] ; then
make_menuconfig
fi
make_kernel
make_modules_pkg
make_firmware_pkg
if [ "x${DTBS}" != "x" ] ; then
make_dtbs_pkg
fi
echo "-----------------------------"
echo "Script Complete"
echo "eewiki.net: [user@localhost:~$ export kernel_version=${KERNEL_UTS}]"
echo "-----------------------------"
```

Appendix B

lxc-checkconfig.in

```
#!/bin/sh

# Allow environment variables to override grep and config
: ${CONFIG:=/proc/config.gz}
: ${GREP:=zgrep}

SETCOLOR_SUCCESS="printf \\033[1;32m"
SETCOLOR_FAILURE="printf \\033[1;31m"
SETCOLOR_WARNING="printf \\033[1;33m"
SETCOLOR_NORMAL="printf \\033[0;39m"

is_set() {
    $GREP -q "$1=[y|m]" $CONFIG
    return $?
}

is_enabled() {
    mandatory=$2

    is_set $1
    RES=$?

    if [ $RES -eq 0 ]; then
        $SETCOLOR_SUCCESS && echo "enabled" && $SETCOLOR_NORMAL
    else
        if [ ! -z "$mandatory" -a "$mandatory" = yes ]; then
            $SETCOLOR_FAILURE && echo "required" && $SETCOLOR_NORMAL
        else
            $SETCOLOR_WARNING && echo "missing" && $SETCOLOR_NORMAL
        fi
    fi
}

if [ ! -f $CONFIG ]; then
    echo "Kernel configuration not found at $CONFIG; searching..."
    KVER="`uname -r`"
    HEADERS_CONFIG="/lib/modules/$KVER/build/.config"
    BOOT_CONFIG="/boot/config-$KVER"
    [ -f "${HEADERS_CONFIG}" ] && CONFIG=${HEADERS_CONFIG}
    [ -f "${BOOT_CONFIG}" ] && CONFIG=${BOOT_CONFIG}
    GREP=grep
    if [ ! -f $CONFIG ]; then
```

```
        echo "$(basename $0): unable to retrieve kernel configuration" >&2
        echo >&2
        echo "Try recompiling with IKCONFIG_PROC, installing the kernel headers," >&2
        echo "or specifying the kernel configuration path with:" >&2
        echo "  CONFIG=<path> $(basename $0)" >&2
        exit 1
    else
        echo "Kernel configuration found at $CONFIG"
    fi
fi

echo "--- Namespaces ---"
echo -n "Namespaces: " && is_enabled CONFIG_NAMESPACES yes
echo -n "Utsname namespace: " && is_enabled CONFIG_UTS_NS
echo -n "Ipc namespace: " && is_enabled CONFIG_IPC_NS yes
echo -n "Pid namespace: " && is_enabled CONFIG_PID_NS yes
echo -n "User namespace: " && is_enabled CONFIG_USER_NS
echo -n "Network namespace: " && is_enabled CONFIG_NET_NS
echo -n "Multiple /dev/pts instances: " && is_enabled DEVPTS_MULTIPLE_INSTANCES
echo
echo "--- Control groups ---"

print_cgroups() {
  # print all mountpoints for cgroup filesystems
  awk ’$1 !˜ /#/ && $3 == mp { print $2; } ; END { exit(0); } ’  "mp=$1" "$2" ;
}

CGROUP_MNT_PATH=‘print_cgroups cgroup /proc/self/mounts | head -1‘
KVER_MAJOR=$($GREP ’ˆ# Linux’ $CONFIG | \
    sed -r ’s/.* ([0-9])\.[0-9]{1,2}\.[0-9]{1,3}.*/\1/’)
if [ "$KVER_MAJOR" = "2" ]; then
KVER_MINOR=$($GREP ’ˆ# Linux’ $CONFIG | \
    sed -r ’s/.* 2.6.([0-9]{2}).*/\1/’)
else
KVER_MINOR=$($GREP ’ˆ# Linux’ $CONFIG | \
    sed -r ’s/.* [0-9]\.([0-9]{1,3})\.[0-9]{1,3}.*/\1/’)
fi

echo -n "Cgroup: " && is_enabled CONFIG_CGROUPS yes

if [ -f $CGROUP_MNT_PATH/cgroup.clone_children ]; then
    echo -n "Cgroup clone_children flag: " &&
    $SETCOLOR_SUCCESS && echo "enabled" && $SETCOLOR_NORMAL
else
    echo -n "Cgroup namespace: " && is_enabled CONFIG_CGROUP_NS yes
fi
echo -n "Cgroup device: " && is_enabled CONFIG_CGROUP_DEVICE
echo -n "Cgroup sched: " && is_enabled CONFIG_CGROUP_SCHED
echo -n "Cgroup cpu account: " && is_enabled CONFIG_CGROUP_CPUACCT
echo -n "Cgroup memory controller: "
if [ $KVER_MAJOR -ge 3 -a $KVER_MINOR -ge 6 ]; then
    is_enabled CONFIG_MEMCG
else
    is_enabled CONFIG_CGROUP_MEM_RES_CTLR
fi
is_set CONFIG_SMP && echo -n "Cgroup cpuset: " && is_enabled CONFIG_CPUSETS
echo
echo "--- Misc ---"
echo -n "Veth pair device: " && is_enabled CONFIG_VETH
echo -n "Macvlan: " && is_enabled CONFIG_MACVLAN
echo -n "Vlan: " && is_enabled CONFIG_VLAN_8021Q
echo -n "File capabilities: " && \
    ( [ "${KVER_MAJOR}" = 2 ] && [ ${KVER_MINOR} -lt 33 ] && \
       is_enabled CONFIG_SECURITY_FILE_CAPABILITIES ) || \
    ( ( [ "${KVER_MAJOR}" = "2" ] && [ ${KVER_MINOR} -gt 32 ] ) || \
         [ ${KVER_MAJOR} -gt 2 ] && $SETCOLOR_SUCCESS && \
         echo "enabled" && $SETCOLOR_NORMAL )

echo
echo "Note : Before booting a new kernel, you can check its configuration"
echo "usage : CONFIG=/path/to/config $0"
echo
```

Appendix C

lxc-create.in

```
#!/bin/sh

#
# lxc: linux Container library

# Authors:
# Daniel Lezcano <daniel.lezcano@free.fr>

# This library is free software; you can redistribute it and/or
# modify it under the terms of the GNU Lesser General Public
# License as published by the Free Software Foundation; either
# version 2.1 of the License, or (at your option) any later version.

# This library is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# Lesser General Public License for more details.

# You should have received a copy of the GNU Lesser General Public
# License along with this library; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

. @DATADIR@/lxc/lxc.functions

usage() {
    echo "usage: $(basename $0) -n NAME [-f CONFIG_FILE] [-t TEMPLATE] [FS_OPTIONS] --" >&2
    echo "         [-P lxcpath] [TEMPLATE_OPTIONS]" >&2
    echo >&2
    echo "where FS_OPTIONS is one of:" >&2
    echo "  -B none" >&2
    echo "  -B dir [--dir rootfs_dir]" >&2
    echo "  -B lvm [--lvname LV_NAME] [--vgname VG_NAME] [--fstype FS_TYPE]" >&2
    echo "    [--fssize FS_SIZE]" >&2
    echo "  -B btrfs" >&2
}

help() {
    usage
    echo >&2
    echo "Create a new container on the system." >&2
    echo >&2
    echo "Options:" >&2
    echo "  -n NAME            specify the name of the container" >&2
    echo "  -f CONFIG_FILE     use an existing configuration file" >&2
    echo "  -t TEMPLATE        use an accessible template script" >&2
    echo "  -B BACKING_STORE   alter the container backing store (default: none)" >&2
    echo "  --lxcpath path     specify an alternate container patch (default: $lxc_path)" >&2
    echo "  --lvname LV_NAME   specify the LVM logical volume name" >&2
    echo "                      (default: container name)" >&2
    echo "  --dir ROOTFS_DIR   specify path for custom rootfs directory location" >&2
    echo "  --vgname VG_NAME   specify the LVM volume group name (default: lxc)" >&2
    echo "  --fstype FS_TYPE   specify the filesystem type (default: ext4)" >&2
    echo "  --fssize FS_SIZE   specify the filesystem size (default: 500M)" >&2
    echo >&2
    if [ -z "$lxc_template" ]; then
        echo "To see template-specific options, specify a template. For example:" >&2
        echo "  $(basename $0) -t ubuntu -h" >&2
        exit 0
    fi
    if [ -x ${templatedir}/lxc-$lxc_template ]; then
        echo >&2
        echo "Template-specific options (TEMPLATE_OPTIONS):" >&2
        ${templatedir}/lxc-$lxc_template -h
    fi
}

usage_err() {
    [ -n "$1" ] && echo "$1" >&2
    usage
    exit 1
}

optarg_check() {
    if [ -z "$2" ]; then
        usage_err "option ’$1’ requires an argument"
    fi
}

backingstore=_unset
fstype=ext4
fssize=500M
vgname=lxc
custom_rootfs=""

while [ $# -gt 0 ]; do
    opt="$1"
    shift
    case "$opt" in
        -h|--help)
            help
            exit 1
            ;;
        -n|--name)
            optarg_check $opt "$1"
            lxc_name=$1
            shift
            ;;
        -f|--config)
            optarg_check $opt "$1"
            lxc_config=$1
            shift
            ;;
        -P|--lxcpath)
            optarg_check $opt "$1"
            lxc_path=$1
            shift
            ;;
        -t|--template)
            optarg_check $opt "$1"
            lxc_template=$1
            shift
            ;;
        -B|--backingstore)
            optarg_check $opt "$1"
            backingstore=$1
            shift
            ;;
        --dir)
            optarg_check $opt "$1"
            custom_rootfs=$1
            shift
            ;;
        --lvname)
            optarg_check $opt "$1"
            lvname=$1
            shift
            ;;
        --vgname)
            optarg_check $opt "$1"
            vgname=$1
            shift
            ;;
        --fstype)
            optarg_check $opt "$1"
            fstype=$1
            shift
            ;;
        --fssize)
            optarg_check $opt "$1"
            fssize=$1
            shift
            ;;
        --)
            break;;
        -?)
            usage_err "unknown option ’$opt’"
            ;;
        -*)
            # split opts -abc into -a -b -c
            set -- $(echo "${opt#-}" | sed ’s/\(.\)/ -\1/g’) "$@"
            ;;
        *)
            usage
            exit 1
            ;;
    esac
done

# If -h or --help was passed into the container, we’ll want to cleanup
# afterward
wantedhelp=0
for var in "$@"; do
    if [ "$var" = "-h" ] || [ "$var" = "--help" ]; then
        help
        exit 1
    fi
done
```

```
if [ -z "$lxc_path" ]; then
    echo "$(basename $0): no configuration path defined" >&2
    exit 1
fi

if [ ! -r $lxc_path ]; then
    echo "$(basename $0): configuration path ’$lxc_path’ not found" >&2
    exit 1
fi

if [ -z "$lxc_name" ]; then
    echo "$(basename $0): no container name specified" >&2
    usage
    exit 1
fi

if [ -z "$lvname" ]; then
    lvname="$lxc_name"
fi

if [ "$(id -u)" != "0" ]; then
   echo "$(basename $0): must be run as root" >&2
   exit 1
fi

if [ -n "$custom_rootfs" ] && [ "$backingstore" != "dir" ]; then
   echo "--dir is only valid with -B dir"
fi

case "$backingstore" in
    dir|lvm|none|btrfs|_unset) :;;
    *)
        echo "$(basename $0): ’$backingstore’ is not known (try ’none’, ’dir’, ’lvm’, ’btrfs’)" >&2
        usage
        exit 1
        ;;
esac

if [ -d "$lxc_path/$lxc_name" ]; then
    echo "$(basename $0): ’$lxc_name’ already exists" >&2
    exit 1
fi

rootfs="$lxc_path/$lxc_name/rootfs"

if [ "$backingstore" = "_unset" ] || [ "$backingstore" = "btrfs" ]; then
# if no backing store was given, then see if btrfs would work
    if which btrfs >/dev/null 2>&1 && \
        btrfs filesystem df "$lxc_path/" >/dev/null 2>&1; then
        backingstore="btrfs"
    else
        if [ "$backingstore" = "btrfs" ]; then
            echo "$(basename $0): missing ’btrfs’ command or $lxc_path is not btrfs" >&2
            exit 1;
        fi
        backingstore="none"
    fi
fi

if [ "$backingstore" = "lvm" ]; then
```

```
    which vgscan > /dev/null 2>&1
    if [ $? -ne 0 ]; then
        echo "$(basename $0): vgscan not found (is lvm2 installed?)" >&2
        exit 1
    fi

    grep -q "\<$fstype\>" /proc/filesystems
    if [ $? -ne 0 ]; then
        echo "$(basename $0): $fstype is not listed in /proc/filesystems" >&2
        exit 1
    fi

    vgscan | grep -q "Found volume group \"$vgname\""
    if [ $? -ne 0 ]; then
        echo "$(basename $0): could not find volume group \"$vgname\"" >&2
        exit 1
    fi

    rootdev=/dev/$vgname/$lvname
    lvdisplay $rootdev > /dev/null 2>&1
    if [ $? -eq 0 ]; then
        echo "$(basename $0): backing store already exists: $rootdev" >&2
        echo "please delete it (using \"lvremove $rootdev\") and try again" >&2
        exit 1
    fi

elif [ "$backingstore" = "btrfs" ]; then
    mkdir "$lxc_path/$lxc_name"
    if ! out=$(btrfs subvolume create "$rootfs" 2>&1); then
        echo "$(basename $0): failed to create subvolume in $rootfs: $out" >&2
        exit 1;
    fi
fi

cleanup() {
    if [ "$backingstore" = "lvm" ]; then
        umount $rootfs
        lvremove -f $rootdev
    elif [ "$backingstore" = "btrfs" ]; then
        btrfs subvolume delete "$rootfs"
    fi

    ${bindir}/lxc-destroy -n $lxc_name
    echo "$(basename $0): aborted" >&2
    exit 1
}

trap cleanup HUP INT TERM

mkdir -p $lxc_path/$lxc_name

if [ -z "$lxc_config" ]; then
    lxc_config="@SYSCONFDIR@/lxc/default.conf"
    echo
    echo "$(basename $0): No config file specified, using the default config $lxc_config"
fi

if [ ! -r "$lxc_config" ]; then
    echo "$(basename $0): '$lxc_config' configuration file not found" >&2
    exit 1
fi
```

```
if [ ! -z "$lxc_template" ]; then
    # Allow for a path to be provided as the template name
    if [ -x "$lxc_template" -a $(echo "$lxc_template" | cut -c 1) = '/' ]; then
        template_path=$lxc_template
    else
        template_path=${templatedir}/lxc-$lxc_template
    fi

    if ! [ -x "$template_path" ]; then
        echo "$(basename $0): unknown template '$lxc_template'" >&2
        cleanup
    fi

    sum=$(sha1sum $template_path | cut -d ' ' -f1)
    echo "# Template used to create this container: $lxc_template" >> $lxc_path/$lxc_name/config
    if [ -n "$*" ]; then
        echo "# Parameters passed to the template: $*" >> $lxc_path/$lxc_name/config
    fi
    echo "# Template script checksum (SHA-1): $sum" >> $lxc_path/$lxc_name/config
    echo "" >> $lxc_path/$lxc_name/config
fi

cat $lxc_config >> $lxc_path/$lxc_name/config
echo "" >> $lxc_path/$lxc_name/config

if [ -n "$custom_rootfs" ]; then
    if grep -q "lxc.rootfs" $lxc_path/$lxc_name/config ; then
        echo "configuration file already specifies a lxc.rootfs"
        exit 1
    fi
    if [ -d "$custom_rootfs" ]; then
        echo "specified rootfs ($custom_rootfs) already exists.  Bailing."
        exit 1
    fi
    echo "lxc.rootfs = $custom_rootfs" >> $lxc_path/$lxc_name/config
fi

# Create the fs as needed
if [ "$backingstore" = "lvm" ]; then
    [ -d "$rootfs" ] || mkdir $rootfs
    lvcreate -L $fssize -n $lvname $vgname || exit 1
    udevadm settle
    mkfs -t $fstype $rootdev || exit 1
    mount -t $fstype $rootdev $rootfs
fi

if [ ! -z "$lxc_template" ]; then
    $template_path --path=$lxc_path/$lxc_name --name=$lxc_name $*
    if [ $? -ne 0 ]; then
        echo "$(basename $0): failed to execute template '$lxc_template'" >&2
        cleanup
    fi

    echo "'$lxc_template' template installed"
fi

if [ "$backingstore" = "lvm" ]; then
    echo "Unmounting LVM"
    umount $rootfs

    # TODO: make the templates set this right from the start?
    sed -i '/lxc.rootfs/d' $lxc_path/$lxc_name/config
    echo "lxc.rootfs = $rootdev" >> $lxc_path/$lxc_name/config
fi

echo "’$lxc_name’ created"
```

Appendix D

lxc-lxcbench (i.MX53)

```
#!/bin/bash

#
# lxc: linux Container library

# Authors:
# Daniel Lezcano <daniel.lezcano@free.fr>
########################################################
# Modifier:
# Francesco Castagnotto <f_castagnotto@yahoo.it>
# # -- this is a modified copy of lxc-busybox template
#########################################################

# This library is free software; you can redistribute it and/or
# modify it under the terms of the GNU Lesser General Public
# License as published by the Free Software Foundation; either
# version 2.1 of the License, or (at your option) any later version.

# This library is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
# Lesser General Public License for more details.

# You should have received a copy of the GNU Lesser General Public
# License along with this library; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

install_busybox()
{
    rootfs=$1
    name=$2
    res=0
    mkdir $rootfs


    cp -v /var/cache/lxc/core-image-lxcbench-imx53qsb.tar.bz2 $rootfs
    echo ">>Archive copied"
    cd $rootfs
    echo "pwd->" && pwd
    bunzip2 core-image-lxcbench-imx53qsb.tar.bz2
    echo ">>Bunzipped"
    tar xpf core-image-lxcbench-imx53qsb.tar
    echo ">>Tar archive extracted"
    ls -lha

        tree="\
$rootfs/selinux \
$rootfs/dev \
$rootfs/etc \
$rootfs/root \
$rootfs/etc/init.d \
$rootfs/bin \
$rootfs/usr/bin \
$rootfs/sbin \
$rootfs/usr/sbin \
$rootfs/proc \
$rootfs/mnt \
$rootfs/tmp \
$rootfs/var/log \
$rootfs/usr/share/udhcpc \
$rootfs/dev/pts \
$rootfs/dev/shm \
$rootfs/lib \
$rootfs/usr/lib \
$rootfs/lib64 \
$rootfs/usr/lib64"

    mkdir -p $tree || return 1
    chmod 0777 $tree || return 1

    echo ">>Make dir /dev"

    mkdir $rootfs/dev
    mkdir $rootfs/dev/shm
    mkdir $rootfs/dev/pts

    echo ">>Mounted 'devpts', 'tmpfs', 'proc' and 'sysfs'"

    pushd $rootfs/dev > /dev/null || return 1

    cd $rootfs/dev
    echo ">>Directory where make device files"
    pwd
    # minimal devices needed for busybox
    rm tty
    mknod tty c 4 0       || res=1


    rm console
    mknod console c 5 1   || res=1
    chmod 666 tty console || res=1

    rm tty0
    rm tty1
    rm tty2
    rm tty3
    rm tty4
    mknod tty0 c 4 0      || res=1
    mknod tty1 c 4 1      || res=1
    mknod tty2 c 4 2      || res=1
    mknod tty3 c 4 3      || res=1
    mknod tty4 c 4 4      || res=1
    chmod 666 tty0 tty1 tty2 tty3 tty4       || res=1

    rm ram0
    mknod ram0 b 1 0      || res=1
    chmod 600 ram0        || res=1

    rm null
    mknod null c 1 3      || res=1
    chmod 666 null        || res=1

    rm urandom
    mknod urandom c 1 9   || res=1
    chmod 666 urandom     || res=1

    echo ">>Device files &co done!"

    popd > /dev/null

    # root user defined
    cat <<EOF >> $rootfs/etc/passwd
    root:x:0:0:root:/home/root:/bin/sh
EOF
    cat <<EOF >> $rootfs/etc/group
root:x:0:
EOF

    cp $rootfs/etc/init.d/rcS file_tmp
    echo "#!/bin/sh" > $rootfs/etc/init.d/rcS
    cat <<EOF >> $rootfs/etc/init.d/rcS
/bin/syslogd
/bin/mount -a
EOF

  cat file_tmp >> $rootfs/etc/init.d/rcS

    rm file_tmp
    # executable
    chmod 744 $rootfs/etc/init.d/rcS || return 1

    echo $name > $rootfs/etc/hostname

    # writable and readable for other
    chmod 644 $rootfs/etc/fstab || return 1

    # launch rcS first then make a console available
    # and propose a shell on the tty, the last one is
    # not needed
    cp $rootfs/etc/inittab $rootfs/etc/inittab.OLD
    cat << EOF > $rootfs/etc/inittab
id:3:initdefault:
si::sysinit:/etc/init.d/rcS
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
# Normally not reached, but fallthrough in case of emergency.
z6:6:respawn:/sbin/sulogin
c1:12345:respawn:/sbin/getty 38400 tty1 linux
c2:12345:respawn:/sbin/getty 38400 tty2 linux
c3:12345:respawn:/sbin/getty 38400 tty3 linux
c4:12345:respawn:/sbin/getty 38400 tty4 linux
p6::ctrlaltdel:/sbin/init 6
```

```
p0::powerfail:/sbin/init 0
EOF
    # writable and readable for other
    chmod 644 $rootfs/etc/inittab || return 1

    cat <<EOF >> $rootfs/usr/share/udhcpc/default.script
#!/bin/sh
case "\$1" in
    deconfig)
        ip addr flush dev \$interface
        ;;

    renew|bound)
        # flush all the routes
        if [ -n "\$router" ]; then
            ip route del default 2> /dev/null
        fi

        # check broadcast
        if [ -n "\$broadcast" ]; then
            broadcast="broadcast \$broadcast"
        fi

        # add a new ip address
        ip addr add \$ip/\$mask \$broadcast dev \$interface

        if [ -n "\$router" ]; then
            ip route add default via \$router dev \$interface
        fi

        [ -n "\$domain" ] && echo search \$domain > /etc/resolv.conf
        for i in \$dns ; do
            echo nameserver \$i >> /etc/resolv.conf
        done
        ;;
esac
exit 0
EOF

    chmod 744 $rootfs/usr/share/udhcpc/default.script

    return $res
}

configure_busybox()
{
    rootfs=$1

    #which busybox >/dev/null 2>&1
    $rootfs/bin/busybox > /dev/null 2>&1

    if [ $? -ne 0 ]; then
        echo "busybox executable is not accessible"
        return 1
    fi

    #file $(which busybox) | grep -q "statically linked"
    file $rootfs/bin/busybox | grep -q "statically linked"
    if [ $? -ne 0 ]; then
        echo "warning : busybox is not statically linked."
        echo "warning : The template script may not correctly"
        echo "warning : setup the container environment."
    fi

    pushd $rootfs/bin > /dev/null || return 1
    $rootfs/bin/busybox --help | grep ’Currently defined functions:’ -A300 | \
      grep -v ’Currently defined functions:’ | tr , ’\n’ | \
      xargs -n1 ln -fs busybox
    popd > /dev/null

    # passwd exec must be setuid
    chmod +s $rootfs/bin/passwd
    touch $rootfs/etc/shadow

    # setting passwd for root
    CHPASSWD_FILE=$rootfs/root/chpasswd.sh

    cat <<EOF >$CHPASSWD_FILE
echo "setting root password to \"root\""

mount --bind /lib $rootfs/lib
if [ \$? -ne 0 ]; then
    echo "Failed bind-mounting /lib at $rootfs/lib"
    exit 1
fi

chroot $rootfs chpasswd <<EOFF 2>/dev/null
root:root
EOFF


if [ \$? -ne 0 ]; then
    echo "Failed to change root password"
    exit 1
fi

umount $rootfs/lib

EOF

    lxc-unshare -s MOUNT -- /bin/sh < $CHPASSWD_FILE
    #rm $CHPASSWD_FILE

    # add ssh functionality if dropbear package available on host
    which dropbear >/dev/null 2>&1
    if [ $? -eq 0 ]; then
        # copy dropbear binary
        cp $(which dropbear) $rootfs/usr/sbin
        if [ $? -ne 0 ]; then
            echo "Failed to copy dropbear in the rootfs"
            return 1
        fi

        # make symlinks to various ssh utilities
        utils="\
            $rootfs/usr/bin/dbclient \
            $rootfs/usr/bin/scp \
            $rootfs/usr/bin/ssh \
            $rootfs/usr/sbin/dropbearkey \
            $rootfs/usr/sbin/dropbearconvert \
        "

        echo "here!!"
        echo $utils | xargs -n1 ln -fs /usr/sbin/dropbear
        # add necessary config files
        mkdir $rootfs/etc/dropbear
        dropbearkey -t rsa -f $rootfs/etc/dropbear/dropbear_rsa_host_key &> /dev/null
        dropbearkey -t dss -f $rootfs/etc/dropbear/dropbear_dss_host_key &> /dev/null

        echo "'dropbear' ssh utility installed"
    fi

    return 0
}

copy_configuration()
{
    path=$1
    rootfs=$2
    name=$3

grep -q "^lxc.rootfs" $path/config 2>/dev/null || echo "lxc.rootfs = $rootfs" >> $path/config
cat <<EOF >> $path/config
lxc.utsname = $name
lxc.tty = 4
lxc.pts = 1024

EOF

if [ -d "$rootfs/lib" ]; then
cat <<EOF >> $path/config

# mounts point
lxc.mount.entry = proc proc proc nodev,noexec,nosuid 0 0
lxc.mount.entry = sysfs sys sysfs defaults  0 0
EOF
fi

    libdirs="\
        lib \
        usr/lib \
        lib64 \
        usr/lib64"

    for dir in $libdirs; do
        if [ -d "/$dir" ] && [ -d "$rootfs/$dir" ]; then
            echo "lxc.mount.entry = /$dir $dir none ro,bind 0 0" >> $path/config
        fi
    done
}

usage()
{
    cat <<EOF
$1 -h|--help -p|--path=<path>
EOF
    return 0
}


options=$(getopt -o hp:n: -l help,path:,name: -- "$@")
if [ $? -ne 0 ]; then
    usage $(basename $0)
    exit 1
fi
```

```
eval set -- "$options"

while true
do
    case "$1" in
        -h|--help)      usage $0 && exit 0;;
        -p|--path)      path=$2; shift 2;;
        -n|--name)      name=$2; shift 2;;
        --)             shift 1; break ;;
        *)              break ;;
    esac
done

if [ "$(id -u)" != "0" ]; then
    echo "This script should be run as ’root’"
    exit 1
fi

if [ -z "$path" ]; then
    echo "’path’ parameter is required"
    exit 1
fi

# detect rootfs
config="$path/config"
if grep -q ’ˆlxc.rootfs’ $config 2>/dev/null ; then
    rootfs=‘grep ’lxc.rootfs =’ $config | awk -F= ’{ print $2 }’‘
else
    rootfs=$path/rootfs
fi


install_busybox $rootfs $name
if [ $? -ne 0 ]; then
    echo "failed to install busybox’s rootfs"
    exit 1
fi

configure_busybox $rootfs
if [ $? -ne 0 ]; then
    echo "failed to configure busybox template"
    exit 1
fi

copy_configuration $path $rootfs $name
if [ $? -ne 0 ]; then
    echo "failed to write configuration file"
    exit 1
fi
```

Appendix E

.config (kernel 3.8.6)

```
#
# Automatically generated file; DO NOT EDIT.
# Linux/arm 3.8.6 Kernel Configuration
#

CONFIG_ARM=y
CONFIG_SYS_SUPPORTS_APM_EMULATION=y
CONFIG_GENERIC_GPIO=y
CONFIG_HAVE_PROC_CPU=y
CONFIG_NO_IOPORT=y
CONFIG_STACKTRACE_SUPPORT=y
CONFIG_LOCKDEP_SUPPORT=y
CONFIG_TRACE_IRQFLAGS_SUPPORT=y
CONFIG_RWSEM_GENERIC_SPINLOCK=y
CONFIG_ARCH_HAS_CPUFREQ=y
CONFIG_GENERIC_HWEIGHT=y
CONFIG_GENERIC_CALIBRATE_DELAY=y
CONFIG_NEED_DMA_MAP_STATE=y
CONFIG_FIQ=y
CONFIG_VECTORS_BASE=0xffff0000
CONFIG_ARM_PATCH_PHYS_VIRT=y
CONFIG_GENERIC_BUG=y
CONFIG_DEFCONFIG_LIST="/lib/modules/$UNAME_RELEASE/.config"
CONFIG_HAVE_IRQ_WORK=y
CONFIG_IRQ_WORK=y
CONFIG_BUILDTIME_EXTABLE_SORT=y

#
# General setup
#
CONFIG_EXPERIMENTAL=y
CONFIG_INIT_ENV_ARG_LIMIT=32
CONFIG_CROSS_COMPILE=""
CONFIG_LOCALVERSION=""
# CONFIG_LOCALVERSION_AUTO is not set
CONFIG_HAVE_KERNEL_GZIP=y
CONFIG_HAVE_KERNEL_LZMA=y
CONFIG_HAVE_KERNEL_XZ=y
CONFIG_HAVE_KERNEL_LZO=y
# CONFIG_KERNEL_GZIP is not set
# CONFIG_KERNEL_LZMA is not set
CONFIG_KERNEL_XZ=y
```

```
# CONFIG_KERNEL_LZO is not set
CONFIG_DEFAULT_HOSTNAME="(none)"
CONFIG_SWAP=y
CONFIG_SYSVIPC=y
CONFIG_SYSVIPC_SYSCTL=y
CONFIG_POSIX_MQUEUE=y
CONFIG_POSIX_MQUEUE_SYSCTL=y
CONFIG_FHANDLE=y
CONFIG_AUDIT=y
CONFIG_AUDITSYSCALL=y
CONFIG_AUDIT_WATCH=y
CONFIG_AUDIT_TREE=y
# CONFIG_AUDIT_LOGINUID_IMMUTABLE is not set
CONFIG_HAVE_GENERIC_HARDIRQS=y

#
# IRQ subsystem
#
CONFIG_GENERIC_HARDIRQS=y
CONFIG_GENERIC_IRQ_PROBE=y
CONFIG_GENERIC_IRQ_SHOW=y
CONFIG_HARDIRQS_SW_RESEND=y
CONFIG_GENERIC_IRQ_CHIP=y
CONFIG_IRQ_DOMAIN=y
CONFIG_IRQ_DOMAIN_DEBUG=y
CONFIG_SPARSE_IRQ=y
CONFIG_KTIME_SCALAR=y
CONFIG_GENERIC_CLOCKEVENTS=y
CONFIG_GENERIC_CLOCKEVENTS_BUILD=y
CONFIG_GENERIC_CLOCKEVENTS_BROADCAST=y

#
# Timers subsystem
#
CONFIG_TICK_ONESHOT=y
CONFIG_NO_HZ=y
CONFIG_HIGH_RES_TIMERS=y

#
# CPU/Task time and stats accounting
#
CONFIG_TICK_CPU_ACCOUNTING=y
CONFIG_BSD_PROCESS_ACCT=y
CONFIG_BSD_PROCESS_ACCT_V3=y
CONFIG_TASKSTATS=y
CONFIG_TASK_DELAY_ACCT=y
CONFIG_TASK_XACCT=y
CONFIG_TASK_IO_ACCOUNTING=y

#
# RCU Subsystem
#
CONFIG_TREE_RCU=y
# CONFIG_PREEMPT_RCU is not set
CONFIG_RCU_FANOUT=32
CONFIG_RCU_FANOUT_LEAF=16
# CONFIG_RCU_FANOUT_EXACT is not set
# CONFIG_RCU_FAST_NO_HZ is not set
# CONFIG_TREE_RCU_TRACE is not set
# CONFIG_RCU_NOCB_CPU is not set
CONFIG_IKCONFIG=y
CONFIG_IKCONFIG_PROC=y
```

```
CONFIG_LOG_BUF_SHIFT=17
CONFIG_CGROUPS=y
# CONFIG_CGROUP_DEBUG is not set
CONFIG_CGROUP_FREEZER=y
CONFIG_CGROUP_DEVICE=y
CONFIG_CPUSETS=y
CONFIG_PROC_PID_CPUSET=y
CONFIG_CGROUP_CPUACCT=y
CONFIG_RESOURCE_COUNTERS=y
CONFIG_MEMCG=y
CONFIG_MEMCG_SWAP=y
CONFIG_MEMCG_SWAP_ENABLED=y
CONFIG_MEMCG_KMEM=y
CONFIG_CGROUP_PERF=y
CONFIG_CGROUP_SCHED=y
CONFIG_FAIR_GROUP_SCHED=y
CONFIG_CFS_BANDWIDTH=y
CONFIG_RT_GROUP_SCHED=y
CONFIG_BLK_CGROUP=y
# CONFIG_DEBUG_BLK_CGROUP is not set
# CONFIG_CHECKPOINT_RESTORE is not set
CONFIG_NAMESPACES=y
CONFIG_UTS_NS=y
CONFIG_IPC_NS=y
CONFIG_USER_NS=y
CONFIG_PID_NS=y
CONFIG_NET_NS=y
CONFIG_UIDGID_CONVERTED=y
CONFIG_UIDGID_STRICT_TYPE_CHECKS=y
CONFIG_SCHED_AUTOGROUP=y
CONFIG_MM_OWNER=y
# CONFIG_SYSFS_DEPRECATED is not set
CONFIG_RELAY=y
CONFIG_BLK_DEV_INITRD=y
CONFIG_INITRAMFS_SOURCE=""
# CONFIG_RD_GZIP is not set
CONFIG_RD_BZIP2=y
CONFIG_RD_LZMA=y
CONFIG_RD_XZ=y
CONFIG_RD_LZO=y
CONFIG_CC_OPTIMIZE_FOR_SIZE=y
CONFIG_SYSCTL=y
CONFIG_ANON_INODES=y
CONFIG_EXPERT=y
CONFIG_HAVE_UID16=y
CONFIG_UID16=y
CONFIG_SYSCTL_SYSCALL=y
CONFIG_KALLSYMS=y
CONFIG_KALLSYMS_ALL=y
CONFIG_HOTPLUG=y
CONFIG_PRINTK=y
CONFIG_BUG=y
CONFIG_ELF_CORE=y
CONFIG_BASE_FULL=y
CONFIG_FUTEX=y
CONFIG_EPOLL=y
CONFIG_SIGNALFD=y
CONFIG_TIMERFD=y
CONFIG_EVENTFD=y
# CONFIG_SHMEM is not set
CONFIG_AIO=y
CONFIG_EMBEDDED=y
```

```
CONFIG_HAVE_PERF_EVENTS=y
CONFIG_PERF_USE_VMALLOC=y

#
# Kernel Performance Events And Counters
#
CONFIG_PERF_EVENTS=y
# CONFIG_DEBUG_PERF_USE_VMALLOC is not set
CONFIG_VM_EVENT_COUNTERS=y
CONFIG_SLUB_DEBUG=y
# CONFIG_COMPAT_BRK is not set
# CONFIG_SLAB is not set
CONFIG_SLUB=y
# CONFIG_SLOB is not set
CONFIG_PROFILING=y
CONFIG_TRACEPOINTS=y
CONFIG_OPROFILE=y
CONFIG_HAVE_OPROFILE=y
CONFIG_KPROBES=y
# CONFIG_JUMP_LABEL is not set
CONFIG_KRETPROBES=y
CONFIG_HAVE_KPROBES=y
CONFIG_HAVE_KRETPROBES=y
CONFIG_HAVE_ARCH_TRACEHOOK=y
CONFIG_HAVE_DMA_ATTRS=y
CONFIG_HAVE_DMA_CONTIGUOUS=y
CONFIG_USE_GENERIC_SMP_HELPERS=y
CONFIG_GENERIC_SMP_IDLE_THREAD=y
CONFIG_HAVE_REGS_AND_STACK_ACCESS_API=y
CONFIG_HAVE_CLK=y
CONFIG_HAVE_DMA_API_DEBUG=y
CONFIG_HAVE_HW_BREAKPOINT=y
CONFIG_HAVE_ARCH_JUMP_LABEL=y
CONFIG_ARCH_WANT_IPC_PARSE_VERSION=y
CONFIG_HAVE_ARCH_SECCOMP_FILTER=y
CONFIG_HAVE_MOD_ARCH_SPECIFIC=y
CONFIG_MODULES_USE_ELF_REL=y
CONFIG_CLONE_BACKWARDS=y

#
# GCOV-based kernel profiling
#
# CONFIG_GCOV_KERNEL is not set
CONFIG_HAVE_GENERIC_DMA_COHERENT=y
CONFIG_SLABINFO=y
CONFIG_RT_MUTEXES=y
CONFIG_BASE_SMALL=0
CONFIG_MODULES=y
CONFIG_MODULE_FORCE_LOAD=y
CONFIG_MODULE_UNLOAD=y
CONFIG_MODULE_FORCE_UNLOAD=y
CONFIG_MODVERSIONS=y
CONFIG_MODULE_SRCVERSION_ALL=y
# CONFIG_MODULE_SIG is not set
CONFIG_STOP_MACHINE=y
CONFIG_BLOCK=y
CONFIG_LBDAF=y
CONFIG_BLK_DEV_BSG=y
CONFIG_BLK_DEV_BSGLIB=y
CONFIG_BLK_DEV_INTEGRITY=y
# CONFIG_BLK_DEV_THROTTLING is not set
```

```
#
# Partition Types
#
CONFIG_PARTITION_ADVANCED=y
CONFIG_ACORN_PARTITION=y
# CONFIG_ACORN_PARTITION_CUMANA is not set
# CONFIG_ACORN_PARTITION_EESOX is not set
CONFIG_ACORN_PARTITION_ICS=y
# CONFIG_ACORN_PARTITION_ADFS is not set
# CONFIG_ACORN_PARTITION_POWERTEC is not set
CONFIG_ACORN_PARTITION_RISCIX=y
CONFIG_OSF_PARTITION=y
CONFIG_AMIGA_PARTITION=y
CONFIG_ATARI_PARTITION=y
CONFIG_MAC_PARTITION=y
CONFIG_MSDOS_PARTITION=y
CONFIG_BSD_DISKLABEL=y
CONFIG_MINIX_SUBPARTITION=y
CONFIG_SOLARIS_X86_PARTITION=y
CONFIG_UNIXWARE_DISKLABEL=y
CONFIG_LDM_PARTITION=y
# CONFIG_LDM_DEBUG is not set
CONFIG_SGI_PARTITION=y
CONFIG_ULTRIX_PARTITION=y
CONFIG_SUN_PARTITION=y
CONFIG_KARMA_PARTITION=y
CONFIG_EFI_PARTITION=y
# CONFIG_SYSV68_PARTITION is not set

#
# IO Schedulers
#
CONFIG_IOSCHED_NOOP=y
CONFIG_IOSCHED_DEADLINE=y
CONFIG_IOSCHED_CFQ=y
# CONFIG_CFQ_GROUP_IOSCHED is not set
# CONFIG_DEFAULT_DEADLINE is not set
CONFIG_DEFAULT_CFQ=y
# CONFIG_DEFAULT_NOOP is not set
CONFIG_DEFAULT_IOSCHED="cfq"
CONFIG_INLINE_SPIN_UNLOCK_IRQ=y
CONFIG_INLINE_READ_UNLOCK=y
CONFIG_INLINE_READ_UNLOCK_IRQ=y
CONFIG_INLINE_WRITE_UNLOCK=y
CONFIG_INLINE_WRITE_UNLOCK_IRQ=y
CONFIG_MUTEX_SPIN_ON_OWNER=y
CONFIG_FREEZER=y

#
# System Type
#
CONFIG_MMU=y
CONFIG_ARCH_MULTIPLATFORM=y
# CONFIG_ARCH_INTEGRATOR is not set
# CONFIG_ARCH_REALVIEW is not set
# CONFIG_ARCH_VERSATILE is not set
# CONFIG_ARCH_AT91 is not set
# CONFIG_ARCH_BCM2835 is not set
# CONFIG_ARCH_CNS3XXX is not set
# CONFIG_ARCH_CLPS711X is not set
# CONFIG_ARCH_GEMINI is not set
# CONFIG_ARCH_SIRF is not set
# CONFIG_ARCH_EBSA110 is not set
# CONFIG_ARCH_EP93XX is not set
# CONFIG_ARCH_FOOTBRIDGE is not set
# CONFIG_ARCH_MXS is not set
# CONFIG_ARCH_NETX is not set
# CONFIG_ARCH_H720X is not set
# CONFIG_ARCH_IOP13XX is not set
# CONFIG_ARCH_IOP32X is not set
# CONFIG_ARCH_IOP33X is not set
# CONFIG_ARCH_IXP4XX is not set
# CONFIG_ARCH_DOVE is not set
# CONFIG_ARCH_KIRKWOOD is not set
# CONFIG_ARCH_MV78XX0 is not set
# CONFIG_ARCH_ORION5X is not set
# CONFIG_ARCH_MMP is not set
# CONFIG_ARCH_KS8695 is not set
# CONFIG_ARCH_W90X900 is not set
# CONFIG_ARCH_LPC32XX is not set
# CONFIG_ARCH_TEGRA is not set
# CONFIG_ARCH_PXA is not set
# CONFIG_ARCH_MSM is not set
# CONFIG_ARCH_SHMOBILE is not set
# CONFIG_ARCH_RPC is not set
# CONFIG_ARCH_SA1100 is not set
# CONFIG_ARCH_S3C24XX is not set
# CONFIG_ARCH_S3C64XX is not set
# CONFIG_ARCH_S5P64X0 is not set
# CONFIG_ARCH_S5PC100 is not set
# CONFIG_ARCH_S5PV210 is not set
# CONFIG_ARCH_EXYNOS is not set
# CONFIG_ARCH_SHARK is not set
# CONFIG_ARCH_U300 is not set
# CONFIG_ARCH_U8500 is not set
# CONFIG_ARCH_NOMADIK is not set
# CONFIG_PLAT_SPEAR is not set
# CONFIG_ARCH_DAVINCI is not set
# CONFIG_ARCH_OMAP is not set
# CONFIG_ARCH_VT8500_SINGLE is not set

#
# Multiple platform selection
#

#
# CPU Core family selection
#
# CONFIG_ARCH_MULTI_V6 is not set
CONFIG_ARCH_MULTI_V7=y
CONFIG_ARCH_MULTI_V6_V7=y
# CONFIG_ARCH_MULTI_CPU_AUTO is not set
# CONFIG_ARCH_MVEBU is not set
# CONFIG_ARCH_BCM is not set
# CONFIG_GPIO_PCA953X is not set
# CONFIG_KEYBOARD_GPIO_POLLED is not set
# CONFIG_ARCH_HIGHBANK is not set
CONFIG_ARCH_MXC=y

#
# Freescale i.MX support
#
# CONFIG_MXC_IRQ_PRIOR is not set
CONFIG_MXC_TZIC=y
```

```
CONFIG_MXC_DEBUG_BOARD=y
CONFIG_HAVE_IMX_GPC=y
CONFIG_HAVE_IMX_MMDC=y
CONFIG_HAVE_IMX_SRC=y
CONFIG_ARCH_MXC_IOMUX_V3=y
CONFIG_ARCH_MX5=y
CONFIG_ARCH_MX51=y
CONFIG_ARCH_MX53=y
CONFIG_SOC_IMX5=y
CONFIG_SOC_IMX51=y

#
# i.MX51 machines:
#
CONFIG_MACH_IMX51_DT=y
# CONFIG_MACH_MX51_BABBAGE is not set
CONFIG_MACH_EUKREA_CPUIMX51SD=y
CONFIG_MACH_EUKREA_MBIMXSD51_BASEBOARD=y

#
# Device tree only
#
CONFIG_SOC_IMX53=y
CONFIG_SOC_IMX6Q=y
CONFIG_IMX_HAVE_PLATFORM_FEC=y
CONFIG_IMX_HAVE_PLATFORM_FSL_USB2_UDC=y
CONFIG_IMX_HAVE_PLATFORM_GPIO_KEYS=y
CONFIG_IMX_HAVE_PLATFORM_IMX2_WDT=y
CONFIG_IMX_HAVE_PLATFORM_IMX_I2C=y
CONFIG_IMX_HAVE_PLATFORM_IMX_SSI=y
CONFIG_IMX_HAVE_PLATFORM_IMX_UART=y
CONFIG_IMX_HAVE_PLATFORM_MXC_EHCI=y
CONFIG_IMX_HAVE_PLATFORM_MXC_NAND=y
CONFIG_IMX_HAVE_PLATFORM_SDHCI_ESDHC_IMX=y
CONFIG_IMX_HAVE_PLATFORM_SPI_IMX=y
CONFIG_IMX_HAVE_PLATFORM_AHCI=y
# CONFIG_ARCH_SOCFPGA is not set
# CONFIG_ARCH_SUNXI is not set
CONFIG_ARCH_VEXPRESS=y

#
# Versatile Express platform type
#
CONFIG_ARCH_VEXPRESS_CORTEX_A5_A9_ERRATA=y
# CONFIG_ARCH_VEXPRESS_CA9X4 is not set
CONFIG_PLAT_VERSATILE_CLCD=y
CONFIG_PLAT_VERSATILE_SCHED_CLOCK=y
# CONFIG_ARCH_VT8500 is not set
# CONFIG_ARCH_ZYNQ is not set
CONFIG_PLAT_VERSATILE=y
CONFIG_ARM_TIMER_SP804=y

#
# Processor Type
#
CONFIG_CPU_V7=y
CONFIG_CPU_32v6K=y
CONFIG_CPU_32v7=y
CONFIG_CPU_ABRT_EV7=y
CONFIG_CPU_PABRT_V7=y
CONFIG_CPU_CACHE_V7=y
CONFIG_CPU_CACHE_VIPT=y
```

```
CONFIG_CPU_COPY_V6=y
CONFIG_CPU_TLB_V7=y
CONFIG_CPU_HAS_ASID=y
CONFIG_CPU_CP15=y
CONFIG_CPU_CP15_MMU=y

#
# Processor Features
#
# CONFIG_ARM_LPAE is not set
# CONFIG_ARCH_PHYS_ADDR_T_64BIT is not set
CONFIG_ARM_THUMB=y
CONFIG_ARM_THUMBEE=y
# CONFIG_ARM_VIRT_EXT is not set
CONFIG_SWP_EMULATE=y
# CONFIG_CPU_ICACHE_DISABLE is not set
# CONFIG_CPU_DCACHE_DISABLE is not set
# CONFIG_CPU_BPREDICT_DISABLE is not set
CONFIG_OUTER_CACHE=y
CONFIG_OUTER_CACHE_SYNC=y
CONFIG_MIGHT_HAVE_CACHE_L2X0=y
CONFIG_CACHE_L2X0=y
CONFIG_CACHE_PL310=y
CONFIG_ARM_L1_CACHE_SHIFT_6=y
CONFIG_ARM_L1_CACHE_SHIFT=6
CONFIG_ARM_DMA_MEM_BUFFERABLE=y
CONFIG_ARM_NR_BANKS=8
CONFIG_MULTI_IRQ_HANDLER=y
# CONFIG_ARM_ERRATA_430973 is not set
CONFIG_PL310_ERRATA_588369=y
CONFIG_ARM_ERRATA_720789=y
CONFIG_PL310_ERRATA_727915=y
CONFIG_PL310_ERRATA_753970=y
CONFIG_ARM_ERRATA_754322=y
# CONFIG_ARM_ERRATA_754327 is not set
CONFIG_ARM_ERRATA_764369=y
CONFIG_PL310_ERRATA_769419=y
CONFIG_ARM_ERRATA_775420=y
CONFIG_ARM_GIC=y
CONFIG_ICST=y

#
# Bus support
#
CONFIG_ARM_AMBA=y
# CONFIG_PCI_SYSCALL is not set
# CONFIG_PCCARD is not set

#
# Kernel Features
#
CONFIG_HAVE_SMP=y
CONFIG_SMP=y
CONFIG_SMP_ON_UP=y
CONFIG_ARM_CPU_TOPOLOGY=y
# CONFIG_SCHED_MC is not set
# CONFIG_SCHED_SMT is not set
CONFIG_HAVE_ARM_SCU=y
CONFIG_ARM_ARCH_TIMER=y
CONFIG_HAVE_ARM_TWD=y
CONFIG_VMSPLIT_3G=y
# CONFIG_VMSPLIT_2G is not set
# CONFIG_VMSPLIT_1G is not set
CONFIG_PAGE_OFFSET=0xC0000000
CONFIG_NR_CPUS=4
CONFIG_HOTPLUG_CPU=y
CONFIG_LOCAL_TIMERS=y
CONFIG_ARCH_NR_GPIO=0
CONFIG_PREEMPT_NONE=y
# CONFIG_PREEMPT_VOLUNTARY is not set
# CONFIG_PREEMPT is not set
CONFIG_HZ=100
# CONFIG_THUMB2_KERNEL is not set
CONFIG_AEABI=y
# CONFIG_OABI_COMPAT is not set
# CONFIG_ARCH_SPARSEMEM_DEFAULT is not set
# CONFIG_ARCH_SELECT_MEMORY_MODEL is not set
CONFIG_HAVE_ARCH_PFN_VALID=y
CONFIG_HIGHMEM=y
CONFIG_HIGHPTE=y
CONFIG_HW_PERF_EVENTS=y
CONFIG_SELECT_MEMORY_MODEL=y
CONFIG_FLATMEM_MANUAL=y
CONFIG_FLATMEM=y
CONFIG_FLAT_NODE_MEM_MAP=y
CONFIG_HAVE_MEMBLOCK=y
CONFIG_MEMORY_ISOLATION=y
CONFIG_PAGEFLAGS_EXTENDED=y
CONFIG_SPLIT_PTLOCK_CPUS=4
CONFIG_COMPACTION=y
CONFIG_MIGRATION=y
# CONFIG_PHYS_ADDR_T_64BIT is not set
CONFIG_ZONE_DMA_FLAG=0
CONFIG_BOUNCE=y
CONFIG_VIRT_TO_BUS=y
# CONFIG_KSM is not set
CONFIG_DEFAULT_MMAP_MIN_ADDR=4096
CONFIG_CROSS_MEMORY_ATTACH=y
# CONFIG_CLEANCACHE is not set
# CONFIG_FRONTSWAP is not set
CONFIG_FORCE_MAX_ZONEORDER=11
CONFIG_ALIGNMENT_TRAP=y
# CONFIG_UACCESS_WITH_MEMCPY is not set
# CONFIG_SECCOMP is not set
# CONFIG_CC_STACKPROTECTOR is not set
# CONFIG_XEN is not set

#
# Boot options
#
CONFIG_USE_OF=y
CONFIG_ATAGS=y
# CONFIG_DEPRECATED_PARAM_STRUCT is not set
CONFIG_ZBOOT_ROM_TEXT=0
CONFIG_ZBOOT_ROM_BSS=0
# CONFIG_ARM_APPENDED_DTB is not set
CONFIG_CMDLINE="noinitrd console=ttymxc0,115200"
CONFIG_CMDLINE_FROM_BOOTLOADER=y
# CONFIG_CMDLINE_EXTEND is not set
# CONFIG_CMDLINE_FORCE is not set
CONFIG_KEXEC=y
CONFIG_ATAGS_PROC=y
# CONFIG_CRASH_DUMP is not set
CONFIG_AUTO_ZRELADDR=y
```

```
#
# CPU Power Management
#

#
# CPU Frequency scaling
#
CONFIG_CPU_FREQ=y
CONFIG_CPU_FREQ_TABLE=y
CONFIG_CPU_FREQ_GOV_COMMON=y
CONFIG_CPU_FREQ_STAT=y
CONFIG_CPU_FREQ_STAT_DETAILS=y
CONFIG_CPU_FREQ_DEFAULT_GOV_PERFORMANCE=y
# CONFIG_CPU_FREQ_DEFAULT_GOV_POWERSAVE is not set
# CONFIG_CPU_FREQ_DEFAULT_GOV_USERSPACE is not set
# CONFIG_CPU_FREQ_DEFAULT_GOV_ONDEMAND is not set
# CONFIG_CPU_FREQ_DEFAULT_GOV_CONSERVATIVE is not set
CONFIG_CPU_FREQ_GOV_PERFORMANCE=y
CONFIG_CPU_FREQ_GOV_POWERSAVE=m
CONFIG_CPU_FREQ_GOV_USERSPACE=m
CONFIG_CPU_FREQ_GOV_ONDEMAND=m
CONFIG_CPU_FREQ_GOV_CONSERVATIVE=m
CONFIG_GENERIC_CPUFREQ_CPU0=y

#
# ARM CPU frequency scaling drivers
#
# CONFIG_ARM_EXYNOS4210_CPUFREQ is not set
# CONFIG_ARM_EXYNOS4X12_CPUFREQ is not set
# CONFIG_ARM_EXYNOS5250_CPUFREQ is not set
CONFIG_ARM_IMX6Q_CPUFREQ=y
CONFIG_CPU_FREQ_IMX=y
CONFIG_CPU_IDLE=y
CONFIG_CPU_IDLE_MULTIPLE_DRIVERS=y
CONFIG_CPU_IDLE_GOV_LADDER=y
CONFIG_CPU_IDLE_GOV_MENU=y
# CONFIG_ARCH_NEEDS_CPU_IDLE_COUPLED is not set

#
# Floating point emulation
#

#
# At least one emulation must be selected
#
CONFIG_VFP=y
CONFIG_VFPv3=y
CONFIG_NEON=y

#
# Userspace binary formats
#
CONFIG_BINFMT_ELF=y
CONFIG_ARCH_BINFMT_ELF_RANDOMIZE_PIE=y
CONFIG_CORE_DUMP_DEFAULT_ELF_HEADERS=y
CONFIG_HAVE_AOUT=y
CONFIG_BINFMT_AOUT=m
CONFIG_BINFMT_MISC=m
CONFIG_COREDUMP=y

#
```

```
# Power management options
#
CONFIG_SUSPEND=y
CONFIG_SUSPEND_FREEZER=y
CONFIG_PM_SLEEP=y
CONFIG_PM_SLEEP_SMP=y
CONFIG_PM_AUTOSLEEP=y
CONFIG_PM_WAKELOCKS=y
CONFIG_PM_WAKELOCKS_LIMIT=100
CONFIG_PM_WAKELOCKS_GC=y
CONFIG_PM_RUNTIME=y
CONFIG_PM=y
CONFIG_PM_DEBUG=y
# CONFIG_PM_ADVANCED_DEBUG is not set
# CONFIG_PM_TEST_SUSPEND is not set
CONFIG_PM_SLEEP_DEBUG=y
# CONFIG_APM_EMULATION is not set
CONFIG_ARCH_HAS_OPP=y
CONFIG_PM_OPP=y
CONFIG_PM_CLK=y
CONFIG_CPU_PM=y
CONFIG_ARCH_SUSPEND_POSSIBLE=y
CONFIG_ARM_CPU_SUSPEND=y
CONFIG_NET=y

#
# Networking options
#
CONFIG_PACKET=y
CONFIG_PACKET_DIAG=m
CONFIG_UNIX=y
CONFIG_UNIX_DIAG=m
CONFIG_XFRM=y
CONFIG_XFRM_ALGO=m
CONFIG_XFRM_USER=m
# CONFIG_XFRM_SUB_POLICY is not set
# CONFIG_XFRM_MIGRATE is not set
# CONFIG_XFRM_STATISTICS is not set
CONFIG_XFRM_IPCOMP=m
CONFIG_NET_KEY=m
# CONFIG_NET_KEY_MIGRATE is not set
CONFIG_INET=y
CONFIG_IP_MULTICAST=y
CONFIG_IP_ADVANCED_ROUTER=y
# CONFIG_IP_FIB_TRIE_STATS is not set
CONFIG_IP_MULTIPLE_TABLES=y
CONFIG_IP_ROUTE_MULTIPATH=y
CONFIG_IP_ROUTE_VERBOSE=y
CONFIG_IP_ROUTE_CLASSID=y
CONFIG_IP_PNP=y
CONFIG_IP_PNP_DHCP=y
CONFIG_IP_PNP_BOOTP=y
# CONFIG_IP_PNP_RARP is not set
CONFIG_NET_IPIP=m
# CONFIG_NET_IPGRE_DEMUX is not set
CONFIG_IP_MROUTE=y
# CONFIG_IP_MROUTE_MULTIPLE_TABLES is not set
CONFIG_IP_PIMSM_V1=y
CONFIG_IP_PIMSM_V2=y
# CONFIG_ARPD is not set
CONFIG_SYN_COOKIES=y
CONFIG_NET_IPVTI=m
```

```
CONFIG_INET_AH=m
CONFIG_INET_ESP=m
CONFIG_INET_IPCOMP=m
CONFIG_INET_XFRM_TUNNEL=m
CONFIG_INET_TUNNEL=m
CONFIG_INET_XFRM_MODE_TRANSPORT=m
CONFIG_INET_XFRM_MODE_TUNNEL=m
CONFIG_INET_XFRM_MODE_BEET=m
CONFIG_INET_LRO=y
CONFIG_INET_DIAG=y
CONFIG_INET_TCP_DIAG=y
CONFIG_INET_UDP_DIAG=y
CONFIG_TCP_CONG_ADVANCED=y
CONFIG_TCP_CONG_BIC=m
CONFIG_TCP_CONG_CUBIC=y
CONFIG_TCP_CONG_WESTWOOD=m
CONFIG_TCP_CONG_HTCP=m
CONFIG_TCP_CONG_HSTCP=m
CONFIG_TCP_CONG_HYBLA=m
CONFIG_TCP_CONG_VEGAS=m
CONFIG_TCP_CONG_SCALABLE=m
CONFIG_TCP_CONG_LP=m
CONFIG_TCP_CONG_VENO=m
CONFIG_TCP_CONG_YEAH=m
CONFIG_TCP_CONG_ILLINOIS=m
CONFIG_DEFAULT_CUBIC=y
# CONFIG_DEFAULT_RENO is not set
CONFIG_DEFAULT_TCP_CONG="cubic"
CONFIG_TCP_MD5SIG=y
CONFIG_IPV6=y
CONFIG_IPV6_PRIVACY=y
# CONFIG_IPV6_ROUTER_PREF is not set
# CONFIG_IPV6_OPTIMISTIC_DAD is not set
CONFIG_INET6_AH=m
CONFIG_INET6_ESP=m
CONFIG_INET6_IPCOMP=m
# CONFIG_IPV6_MIP6 is not set
CONFIG_INET6_XFRM_TUNNEL=m
CONFIG_INET6_TUNNEL=m
CONFIG_INET6_XFRM_MODE_TRANSPORT=m
CONFIG_INET6_XFRM_MODE_TUNNEL=m
CONFIG_INET6_XFRM_MODE_BEET=m
CONFIG_INET6_XFRM_MODE_ROUTEOPTIMIZATION=m
CONFIG_IPV6_SIT=m
CONFIG_IPV6_SIT_6RD=y
CONFIG_IPV6_NDISC_NODETYPE=y
CONFIG_IPV6_TUNNEL=m
CONFIG_IPV6_GRE=m
CONFIG_IPV6_MULTIPLE_TABLES=y
# CONFIG_IPV6_SUBTREES is not set
# CONFIG_IPV6_MROUTE is not set
CONFIG_NETLABEL=y
CONFIG_NETWORK_SECMARK=y
# CONFIG_NETWORK_PHY_TIMESTAMPING is not set
CONFIG_NETFILTER=y
# CONFIG_NETFILTER_DEBUG is not set
CONFIG_NETFILTER_ADVANCED=y
CONFIG_BRIDGE_NETFILTER=y

#
# Core Netfilter Configuration
#
```

```
CONFIG_NETFILTER_NETLINK=m
CONFIG_NETFILTER_NETLINK_ACCT=m
CONFIG_NETFILTER_NETLINK_QUEUE=m
CONFIG_NETFILTER_NETLINK_LOG=m
CONFIG_NF_CONNTRACK=m
CONFIG_NF_CONNTRACK_MARK=y
CONFIG_NF_CONNTRACK_SECMARK=y
CONFIG_NF_CONNTRACK_ZONES=y
CONFIG_NF_CONNTRACK_PROCFS=y
CONFIG_NF_CONNTRACK_EVENTS=y
# CONFIG_NF_CONNTRACK_TIMEOUT is not set
# CONFIG_NF_CONNTRACK_TIMESTAMP is not set
CONFIG_NF_CT_PROTO_DCCP=m
CONFIG_NF_CT_PROTO_GRE=m
CONFIG_NF_CT_PROTO_SCTP=m
CONFIG_NF_CT_PROTO_UDPLITE=m
CONFIG_NF_CONNTRACK_AMANDA=m
CONFIG_NF_CONNTRACK_FTP=m
CONFIG_NF_CONNTRACK_H323=m
CONFIG_NF_CONNTRACK_IRC=m
CONFIG_NF_CONNTRACK_BROADCAST=m
CONFIG_NF_CONNTRACK_NETBIOS_NS=m
# CONFIG_NF_CONNTRACK_SNMP is not set
CONFIG_NF_CONNTRACK_PPTP=m
CONFIG_NF_CONNTRACK_SANE=m
CONFIG_NF_CONNTRACK_SIP=m
CONFIG_NF_CONNTRACK_TFTP=m
CONFIG_NF_CT_NETLINK=m
CONFIG_NF_CT_NETLINK_TIMEOUT=m
# CONFIG_NF_CT_NETLINK_HELPER is not set
CONFIG_NETFILTER_NETLINK_QUEUE_CT=y
CONFIG_NF_NAT=m
CONFIG_NF_NAT_NEEDED=y
CONFIG_NF_NAT_PROTO_DCCP=m
CONFIG_NF_NAT_PROTO_UDPLITE=m
CONFIG_NF_NAT_PROTO_SCTP=m
CONFIG_NF_NAT_AMANDA=m
CONFIG_NF_NAT_FTP=m
CONFIG_NF_NAT_IRC=m
CONFIG_NF_NAT_SIP=m
CONFIG_NF_NAT_TFTP=m
CONFIG_NETFILTER_TPROXY=m
CONFIG_NETFILTER_XTABLES=m

#
# Xtables combined modules
#
CONFIG_NETFILTER_XT_MARK=m
CONFIG_NETFILTER_XT_CONNMARK=m

#
# Xtables targets
#
# CONFIG_NETFILTER_XT_TARGET_AUDIT is not set
CONFIG_NETFILTER_XT_TARGET_CHECKSUM=m
CONFIG_NETFILTER_XT_TARGET_CLASSIFY=m
CONFIG_NETFILTER_XT_TARGET_CONNMARK=m
CONFIG_NETFILTER_XT_TARGET_CONNSECMARK=m
CONFIG_NETFILTER_XT_TARGET_CT=m
CONFIG_NETFILTER_XT_TARGET_DSCP=m
CONFIG_NETFILTER_XT_TARGET_HL=m
CONFIG_NETFILTER_XT_TARGET_HMARK=m
```

```
CONFIG_NETFILTER_XT_TARGET_IDLETIMER=m
CONFIG_NETFILTER_XT_TARGET_LED=m
CONFIG_NETFILTER_XT_TARGET_LOG=m
CONFIG_NETFILTER_XT_TARGET_MARK=m
CONFIG_NETFILTER_XT_TARGET_NETMAP=m
CONFIG_NETFILTER_XT_TARGET_NFLOG=m
CONFIG_NETFILTER_XT_TARGET_NFQUEUE=m
CONFIG_NETFILTER_XT_TARGET_NOTRACK=m
CONFIG_NETFILTER_XT_TARGET_RATEEST=m
CONFIG_NETFILTER_XT_TARGET_REDIRECT=m
CONFIG_NETFILTER_XT_TARGET_TEE=m
CONFIG_NETFILTER_XT_TARGET_TPROXY=m
CONFIG_NETFILTER_XT_TARGET_TRACE=m
CONFIG_NETFILTER_XT_TARGET_SECMARK=m
CONFIG_NETFILTER_XT_TARGET_TCPMSS=m
CONFIG_NETFILTER_XT_TARGET_TCPOPTSTRIP=m

#
# Xtables matches
#
CONFIG_NETFILTER_XT_MATCH_ADDRTYPE=m
CONFIG_NETFILTER_XT_MATCH_CLUSTER=m
CONFIG_NETFILTER_XT_MATCH_COMMENT=m
CONFIG_NETFILTER_XT_MATCH_CONNBYTES=m
CONFIG_NETFILTER_XT_MATCH_CONNLIMIT=m
CONFIG_NETFILTER_XT_MATCH_CONNMARK=m
CONFIG_NETFILTER_XT_MATCH_CONNTRACK=m
CONFIG_NETFILTER_XT_MATCH_CPU=m
CONFIG_NETFILTER_XT_MATCH_DCCP=m
CONFIG_NETFILTER_XT_MATCH_DEVGROUP=m
CONFIG_NETFILTER_XT_MATCH_DSCP=m
CONFIG_NETFILTER_XT_MATCH_ECN=m
CONFIG_NETFILTER_XT_MATCH_ESP=m
CONFIG_NETFILTER_XT_MATCH_HASHLIMIT=m
CONFIG_NETFILTER_XT_MATCH_HELPER=m
CONFIG_NETFILTER_XT_MATCH_HL=m
CONFIG_NETFILTER_XT_MATCH_IPRANGE=m
CONFIG_NETFILTER_XT_MATCH_IPVS=m
CONFIG_NETFILTER_XT_MATCH_LENGTH=m
CONFIG_NETFILTER_XT_MATCH_LIMIT=m
CONFIG_NETFILTER_XT_MATCH_MAC=m
CONFIG_NETFILTER_XT_MATCH_MARK=m
CONFIG_NETFILTER_XT_MATCH_MULTIPORT=m
CONFIG_NETFILTER_XT_MATCH_NFACCT=m
CONFIG_NETFILTER_XT_MATCH_OSF=m
CONFIG_NETFILTER_XT_MATCH_OWNER=m
CONFIG_NETFILTER_XT_MATCH_POLICY=m
CONFIG_NETFILTER_XT_MATCH_PHYSDEV=m
CONFIG_NETFILTER_XT_MATCH_PKTTYPE=m
CONFIG_NETFILTER_XT_MATCH_QUOTA=m
CONFIG_NETFILTER_XT_MATCH_RATEEST=m
CONFIG_NETFILTER_XT_MATCH_REALM=m
CONFIG_NETFILTER_XT_MATCH_RECENT=m
CONFIG_NETFILTER_XT_MATCH_SCTP=m
CONFIG_NETFILTER_XT_MATCH_SOCKET=m
CONFIG_NETFILTER_XT_MATCH_STATE=m
CONFIG_NETFILTER_XT_MATCH_STATISTIC=m
CONFIG_NETFILTER_XT_MATCH_STRING=m
CONFIG_NETFILTER_XT_MATCH_TCPMSS=m
CONFIG_NETFILTER_XT_MATCH_TIME=m
CONFIG_NETFILTER_XT_MATCH_U32=m
# CONFIG_IP_SET is not set
```

```
CONFIG_IP_VS=m
CONFIG_IP_VS_IPV6=y
# CONFIG_IP_VS_DEBUG is not set
CONFIG_IP_VS_TAB_BITS=12

#
# IPVS transport protocol load balancing support
#
CONFIG_IP_VS_PROTO_TCP=y
CONFIG_IP_VS_PROTO_UDP=y
CONFIG_IP_VS_PROTO_AH_ESP=y
CONFIG_IP_VS_PROTO_ESP=y
CONFIG_IP_VS_PROTO_AH=y
CONFIG_IP_VS_PROTO_SCTP=y

#
# IPVS scheduler
#
CONFIG_IP_VS_RR=m
CONFIG_IP_VS_WRR=m
CONFIG_IP_VS_LC=m
CONFIG_IP_VS_WLC=m
CONFIG_IP_VS_LBLC=m
CONFIG_IP_VS_LBLCR=m
CONFIG_IP_VS_DH=m
CONFIG_IP_VS_SH=m
CONFIG_IP_VS_SED=m
CONFIG_IP_VS_NQ=m

#
# IPVS SH scheduler
#
CONFIG_IP_VS_SH_TAB_BITS=8

#
# IPVS application helper
#
CONFIG_IP_VS_FTP=m
CONFIG_IP_VS_NFCT=y
CONFIG_IP_VS_PE_SIP=m

#
# IP: Netfilter Configuration
#
CONFIG_NF_DEFRAG_IPV4=m
CONFIG_NF_CONNTRACK_IPV4=m
CONFIG_NF_CONNTRACK_PROC_COMPAT=y
CONFIG_IP_NF_QUEUE=m
CONFIG_IP_NF_IPTABLES=m
CONFIG_IP_NF_MATCH_AH=m
CONFIG_IP_NF_MATCH_ECN=m
CONFIG_IP_NF_MATCH_RPFILTER=m
CONFIG_IP_NF_MATCH_TTL=m
CONFIG_IP_NF_FILTER=m
CONFIG_IP_NF_TARGET_REJECT=m
CONFIG_IP_NF_TARGET_ULOG=m
CONFIG_NF_NAT_IPV4=m
CONFIG_IP_NF_TARGET_MASQUERADE=m
CONFIG_IP_NF_TARGET_NETMAP=m
CONFIG_IP_NF_TARGET_REDIRECT=m
CONFIG_NF_NAT_PROTO_GRE=m
CONFIG_NF_NAT_PPTP=m
```

```
CONFIG_NF_NAT_H323=m
CONFIG_IP_NF_MANGLE=m
CONFIG_IP_NF_TARGET_CLUSTERIP=m
CONFIG_IP_NF_TARGET_ECN=m
CONFIG_IP_NF_TARGET_TTL=m
CONFIG_IP_NF_RAW=m
CONFIG_IP_NF_SECURITY=m
CONFIG_IP_NF_ARPTABLES=m
CONFIG_IP_NF_ARPFILTER=m
CONFIG_IP_NF_ARP_MANGLE=m

#
# IPv6: Netfilter Configuration
#
CONFIG_NF_DEFRAG_IPV6=m
CONFIG_NF_CONNTRACK_IPV6=m
CONFIG_IP6_NF_IPTABLES=m
CONFIG_IP6_NF_MATCH_AH=m
CONFIG_IP6_NF_MATCH_EUI64=m
CONFIG_IP6_NF_MATCH_FRAG=m
CONFIG_IP6_NF_MATCH_OPTS=m
CONFIG_IP6_NF_MATCH_HL=m
CONFIG_IP6_NF_MATCH_IPV6HEADER=m
CONFIG_IP6_NF_MATCH_MH=m
CONFIG_IP6_NF_MATCH_RPFILTER=m
CONFIG_IP6_NF_MATCH_RT=m
CONFIG_IP6_NF_TARGET_HL=m
CONFIG_IP6_NF_FILTER=m
CONFIG_IP6_NF_TARGET_REJECT=m
CONFIG_IP6_NF_MANGLE=m
CONFIG_IP6_NF_RAW=m
CONFIG_IP6_NF_SECURITY=m
CONFIG_NF_NAT_IPV6=m
CONFIG_IP6_NF_TARGET_MASQUERADE=m
CONFIG_IP6_NF_TARGET_NPT=m

#
# DECnet: Netfilter Configuration
#
CONFIG_DECNET_NF_GRABULATOR=m
CONFIG_BRIDGE_NF_EBTABLES=m
CONFIG_BRIDGE_EBT_BROUTE=m
CONFIG_BRIDGE_EBT_T_FILTER=m
CONFIG_BRIDGE_EBT_T_NAT=m
CONFIG_BRIDGE_EBT_802_3=m
CONFIG_BRIDGE_EBT_AMONG=m
CONFIG_BRIDGE_EBT_ARP=m
CONFIG_BRIDGE_EBT_IP=m
CONFIG_BRIDGE_EBT_IP6=m
CONFIG_BRIDGE_EBT_LIMIT=m
CONFIG_BRIDGE_EBT_MARK=m
CONFIG_BRIDGE_EBT_PKTTYPE=m
CONFIG_BRIDGE_EBT_STP=m
CONFIG_BRIDGE_EBT_VLAN=m
CONFIG_BRIDGE_EBT_ARPREPLY=m
CONFIG_BRIDGE_EBT_DNAT=m
CONFIG_BRIDGE_EBT_MARK_T=m
CONFIG_BRIDGE_EBT_REDIRECT=m
CONFIG_BRIDGE_EBT_SNAT=m
CONFIG_BRIDGE_EBT_LOG=m
CONFIG_BRIDGE_EBT_ULOG=m
CONFIG_BRIDGE_EBT_NFLOG=m
```

```
CONFIG_IP_DCCP=m
CONFIG_INET_DCCP_DIAG=m

#
# DCCP CCIDs Configuration (EXPERIMENTAL)
#
# CONFIG_IP_DCCP_CCID2_DEBUG is not set
CONFIG_IP_DCCP_CCID3=y
# CONFIG_IP_DCCP_CCID3_DEBUG is not set
CONFIG_IP_DCCP_TFRC_LIB=y

#
# DCCP Kernel Hacking
#
# CONFIG_IP_DCCP_DEBUG is not set
# CONFIG_NET_DCCPPROBE is not set
CONFIG_IP_SCTP=m
CONFIG_NET_SCTPPROBE=m
# CONFIG_SCTP_DBG_MSG is not set
# CONFIG_SCTP_DBG_OBJCNT is not set
CONFIG_SCTP_DEFAULT_COOKIE_HMAC_MD5=y
# CONFIG_SCTP_DEFAULT_COOKIE_HMAC_SHA1 is not set
# CONFIG_SCTP_DEFAULT_COOKIE_HMAC_NONE is not set
CONFIG_SCTP_COOKIE_HMAC_MD5=y
CONFIG_SCTP_COOKIE_HMAC_SHA1=y
CONFIG_RDS=m
CONFIG_RDS_TCP=m
# CONFIG_RDS_DEBUG is not set
CONFIG_TIPC=m
CONFIG_TIPC_PORTS=8191
CONFIG_ATM=m
CONFIG_ATM_CLIP=m
# CONFIG_ATM_CLIP_NO_ICMP is not set
CONFIG_ATM_LANE=m
CONFIG_ATM_MPOA=m
CONFIG_ATM_BR2684=m
CONFIG_ATM_BR2684_IPFILTER=y
CONFIG_L2TP=m
CONFIG_L2TP_DEBUGFS=m
CONFIG_L2TP_V3=y
CONFIG_L2TP_IP=m
CONFIG_L2TP_ETH=m
CONFIG_STP=y
CONFIG_GARP=y
CONFIG_BRIDGE=m
CONFIG_BRIDGE_IGMP_SNOOPING=y
CONFIG_HAVE_NET_DSA=y
CONFIG_NET_DSA=y
CONFIG_NET_DSA_TAG_DSA=y
CONFIG_NET_DSA_TAG_EDSA=y
CONFIG_NET_DSA_TAG_TRAILER=y
CONFIG_VLAN_8021Q=y
CONFIG_VLAN_8021Q_GVRP=y
CONFIG_DECNET=m
# CONFIG_DECNET_ROUTER is not set
CONFIG_LLC=y
CONFIG_LLC2=m
# CONFIG_IPX is not set
# CONFIG_ATALK is not set
# CONFIG_X25 is not set
CONFIG_LAPB=m
CONFIG_WAN_ROUTER=m
```

```
CONFIG_PHONET=m
CONFIG_IEEE802154=m
CONFIG_IEEE802154_6LOWPAN=m
CONFIG_MAC802154=m
CONFIG_NET_SCHED=y

#
# Queueing/Scheduling
#
CONFIG_NET_SCH_CBQ=m
CONFIG_NET_SCH_HTB=m
CONFIG_NET_SCH_HFSC=m
CONFIG_NET_SCH_ATM=m
CONFIG_NET_SCH_PRIO=m
CONFIG_NET_SCH_MULTIQ=m
CONFIG_NET_SCH_RED=m
# CONFIG_NET_SCH_SFB is not set
CONFIG_NET_SCH_SFQ=m
CONFIG_NET_SCH_TEQL=m
CONFIG_NET_SCH_TBF=m
CONFIG_NET_SCH_GRED=m
CONFIG_NET_SCH_DSMARK=m
CONFIG_NET_SCH_NETEM=m
CONFIG_NET_SCH_DRR=m
# CONFIG_NET_SCH_MQPRIO is not set
# CONFIG_NET_SCH_CHOKE is not set
# CONFIG_NET_SCH_QFQ is not set
CONFIG_NET_SCH_CODEL=m
CONFIG_NET_SCH_FQ_CODEL=m
# CONFIG_NET_SCH_INGRESS is not set
# CONFIG_NET_SCH_PLUG is not set

#
# Classification
#
CONFIG_NET_CLS=y
CONFIG_NET_CLS_BASIC=m
CONFIG_NET_CLS_TCINDEX=m
CONFIG_NET_CLS_ROUTE4=m
CONFIG_NET_CLS_FW=m
# CONFIG_NET_CLS_U32 is not set
CONFIG_NET_CLS_RSVP=m
CONFIG_NET_CLS_RSVP6=m
# CONFIG_NET_CLS_FLOW is not set
CONFIG_NET_CLS_CGROUP=y
CONFIG_NET_EMATCH=y
CONFIG_NET_EMATCH_STACK=32
CONFIG_NET_EMATCH_CMP=m
CONFIG_NET_EMATCH_NBYTE=m
CONFIG_NET_EMATCH_U32=m
CONFIG_NET_EMATCH_META=m
CONFIG_NET_EMATCH_TEXT=m
CONFIG_NET_EMATCH_CANID=m
CONFIG_NET_CLS_ACT=y
CONFIG_NET_ACT_POLICE=m
CONFIG_NET_ACT_GACT=m
CONFIG_GACT_PROB=y
CONFIG_NET_ACT_MIRRED=m
CONFIG_NET_ACT_IPT=m
CONFIG_NET_ACT_NAT=m
CONFIG_NET_ACT_PEDIT=m
CONFIG_NET_ACT_SIMP=m
```

```
CONFIG_NET_ACT_SKBEDIT=m
CONFIG_NET_ACT_CSUM=m
# CONFIG_NET_CLS_IND is not set
CONFIG_NET_SCH_FIFO=y
CONFIG_DCB=y
CONFIG_DNS_RESOLVER=y
CONFIG_BATMAN_ADV=m
CONFIG_BATMAN_ADV_BLA=y
# CONFIG_BATMAN_ADV_DAT is not set
# CONFIG_BATMAN_ADV_DEBUG is not set
CONFIG_OPENVSWITCH=m
CONFIG_RPS=y
CONFIG_RFS_ACCEL=y
CONFIG_XPS=y
CONFIG_NETPRIO_CGROUP=m
CONFIG_BQL=y
CONFIG_BPF_JIT=y

#
# Network testing
#
CONFIG_NET_PKTGEN=m
# CONFIG_NET_TCPPROBE is not set
# CONFIG_NET_DROP_MONITOR is not set
CONFIG_HAMRADIO=y

#
# Packet Radio protocols
#
CONFIG_AX25=m
CONFIG_AX25_DAMA_SLAVE=y
CONFIG_NETROM=m
CONFIG_ROSE=m

#
# AX.25 network device drivers
#
CONFIG_MKISS=m
CONFIG_6PACK=m
CONFIG_BPQETHER=m
CONFIG_BAYCOM_SER_FDX=m
CONFIG_BAYCOM_SER_HDX=m
CONFIG_YAM=m
CONFIG_CAN=m
CONFIG_CAN_RAW=m
CONFIG_CAN_BCM=m
CONFIG_CAN_GW=m

#
# CAN Device Drivers
#
CONFIG_CAN_VCAN=m
CONFIG_CAN_SLCAN=m
CONFIG_CAN_DEV=m
CONFIG_CAN_CALC_BITTIMING=y
CONFIG_CAN_MCP251X=m
CONFIG_HAVE_CAN_FLEXCAN=y
# CONFIG_CAN_FLEXCAN is not set
CONFIG_CAN_GRCAN=m
# CONFIG_CAN_SJA1000 is not set
CONFIG_CAN_C_CAN=m
CONFIG_CAN_C_CAN_PLATFORM=m
```

```
# CONFIG_CAN_CC770 is not set

#
# CAN USB interfaces
#
CONFIG_CAN_EMS_USB=m
CONFIG_CAN_ESD_USB2=m
CONFIG_CAN_KVASER_USB=m
CONFIG_CAN_PEAK_USB=m
# CONFIG_CAN_SOFTING is not set
# CONFIG_CAN_DEBUG_DEVICES is not set
CONFIG_IRDA=m

#
# IrDA protocols
#
CONFIG_IRLAN=m
CONFIG_IRNET=m
CONFIG_IRCOMM=m
CONFIG_IRDA_ULTRA=y

#
# IrDA options
#
CONFIG_IRDA_CACHE_LAST_LSAP=y
CONFIG_IRDA_FAST_RR=y
CONFIG_IRDA_DEBUG=y

#
# Infrared-port device drivers
#

#
# SIR device drivers
#
CONFIG_IRTTY_SIR=m

#
# Dongle support
#
CONFIG_DONGLE=y
CONFIG_ESI_DONGLE=m
CONFIG_ACTISYS_DONGLE=m
CONFIG_TEKRAM_DONGLE=m
CONFIG_TOIM3232_DONGLE=m
CONFIG_LITELINK_DONGLE=m
CONFIG_MA600_DONGLE=m
CONFIG_GIRBIL_DONGLE=m
CONFIG_MCP2120_DONGLE=m
CONFIG_OLD_BELKIN_DONGLE=m
CONFIG_ACT200L_DONGLE=m
CONFIG_KINGSUN_DONGLE=m
CONFIG_KSDAZZLE_DONGLE=m
CONFIG_KS959_DONGLE=m

#
# FIR device drivers
#
CONFIG_USB_IRDA=m
CONFIG_SIGMATEL_FIR=m
CONFIG_MCS_FIR=m
CONFIG_BT=m
```

```
CONFIG_BT_RFCOMM=m
CONFIG_BT_RFCOMM_TTY=y
CONFIG_BT_BNEP=m
CONFIG_BT_BNEP_MC_FILTER=y
CONFIG_BT_BNEP_PROTO_FILTER=y
CONFIG_BT_HIDP=m

#
# Bluetooth device drivers
#
CONFIG_BT_HCIBTUSB=m
CONFIG_BT_HCIBTSDIO=m
CONFIG_BT_HCIUART=m
CONFIG_BT_HCIUART_H4=y
CONFIG_BT_HCIUART_BCSP=y
CONFIG_BT_HCIUART_ATH3K=y
CONFIG_BT_HCIUART_LL=y
CONFIG_BT_HCIUART_3WIRE=y
CONFIG_BT_HCIBCM203X=m
CONFIG_BT_HCIBPA10X=m
CONFIG_BT_HCIBFUSB=m
CONFIG_BT_HCIVHCI=m
CONFIG_BT_MRVL=m
CONFIG_BT_MRVL_SDIO=m
CONFIG_BT_ATH3K=m
CONFIG_BT_WILINK=m
CONFIG_AF_RXRPC=m
# CONFIG_AF_RXRPC_DEBUG is not set
CONFIG_RXKAD=m
CONFIG_FIB_RULES=y
CONFIG_WIRELESS=y
CONFIG_WIRELESS_EXT=y
CONFIG_WEXT_CORE=y
CONFIG_WEXT_PROC=y
CONFIG_WEXT_SPY=y
CONFIG_WEXT_PRIV=y
CONFIG_CFG80211=m
# CONFIG_NL80211_TESTMODE is not set
# CONFIG_CFG80211_DEVELOPER_WARNINGS is not set
CONFIG_CFG80211_REG_DEBUG=y
# CONFIG_CFG80211_CERTIFICATION_ONUS is not set
CONFIG_CFG80211_DEFAULT_PS=y
CONFIG_CFG80211_DEBUGFS=y
# CONFIG_CFG80211_INTERNAL_REGDB is not set
CONFIG_CFG80211_WEXT=y
CONFIG_LIB80211=m
# CONFIG_LIB80211_DEBUG is not set
CONFIG_MAC80211=m
CONFIG_MAC80211_HAS_RC=y
CONFIG_MAC80211_RC_PID=y
CONFIG_MAC80211_RC_MINSTREL=y
CONFIG_MAC80211_RC_MINSTREL_HT=y
# CONFIG_MAC80211_RC_DEFAULT_PID is not set
CONFIG_MAC80211_RC_DEFAULT_MINSTREL=y
CONFIG_MAC80211_RC_DEFAULT="minstrel_ht"
CONFIG_MAC80211_MESH=y
CONFIG_MAC80211_LEDS=y
CONFIG_MAC80211_DEBUGFS=y
# CONFIG_MAC80211_MESSAGE_TRACING is not set
# CONFIG_MAC80211_DEBUG_MENU is not set
CONFIG_WIMAX=m
CONFIG_WIMAX_DEBUG_LEVEL=8
```

```
CONFIG_RFKILL=m
CONFIG_RFKILL_LEDS=y
CONFIG_RFKILL_INPUT=y
CONFIG_RFKILL_REGULATOR=m
CONFIG_RFKILL_GPIO=m
# CONFIG_NET_9P is not set
CONFIG_CAIF=m
# CONFIG_CAIF_DEBUG is not set
CONFIG_CAIF_NETDEV=m
CONFIG_CAIF_USB=m
CONFIG_CEPH_LIB=m
# CONFIG_CEPH_LIB_PRETTYDEBUG is not set
# CONFIG_CEPH_LIB_USE_DNS_RESOLVER is not set
CONFIG_NFC=m
CONFIG_NFC_NCI=m
CONFIG_NFC_HCI=m
CONFIG_NFC_SHDLC=y
# CONFIG_NFC_LLCP is not set

#
# Near Field Communication (NFC) devices
#
# CONFIG_PN544_HCI_NFC is not set
CONFIG_NFC_PN533=m
# CONFIG_NFC_WILINK is not set
CONFIG_HAVE_BPF_JIT=y

#
# Device Drivers
#

#
# Generic Driver Options
#
CONFIG_UEVENT_HELPER_PATH=""
CONFIG_DEVTMPFS=y
CONFIG_DEVTMPFS_MOUNT=y
CONFIG_STANDALONE=y
CONFIG_PREVENT_FIRMWARE_BUILD=y
CONFIG_FW_LOADER=y
CONFIG_FIRMWARE_IN_KERNEL=y
CONFIG_EXTRA_FIRMWARE=""
# CONFIG_DEBUG_DRIVER is not set
# CONFIG_DEBUG_DEVRES is not set
# CONFIG_SYS_HYPERVISOR is not set
# CONFIG_GENERIC_CPU_DEVICES is not set
CONFIG_REGMAP=y
CONFIG_REGMAP_I2C=y
CONFIG_REGMAP_SPI=y
CONFIG_REGMAP_MMIO=y
CONFIG_DMA_SHARED_BUFFER=y
CONFIG_CMA=y
# CONFIG_CMA_DEBUG is not set

#
# Default contiguous memory area size:
#
CONFIG_CMA_SIZE_MBYTES=16
CONFIG_CMA_SIZE_SEL_MBYTES=y
# CONFIG_CMA_SIZE_SEL_PERCENTAGE is not set
# CONFIG_CMA_SIZE_SEL_MIN is not set
# CONFIG_CMA_SIZE_SEL_MAX is not set
```

```
CONFIG_CMA_ALIGNMENT=8
CONFIG_CMA_AREAS=7

#
# Bus devices
#
CONFIG_CONNECTOR=y
CONFIG_PROC_EVENTS=y
CONFIG_MTD=y
# CONFIG_MTD_TESTS is not set
# CONFIG_MTD_REDBOOT_PARTS is not set
CONFIG_MTD_CMDLINE_PARTS=y
# CONFIG_MTD_AFS_PARTS is not set
CONFIG_MTD_OF_PARTS=y
# CONFIG_MTD_AR7_PARTS is not set

#
# User Modules And Translation Layers
#
CONFIG_MTD_CHAR=y
CONFIG_MTD_BLKDEVS=y
CONFIG_MTD_BLOCK=y
CONFIG_FTL=m
CONFIG_NFTL=m
CONFIG_NFTL_RW=y
CONFIG_INFTL=m
CONFIG_RFD_FTL=m
CONFIG_SSFDC=m
CONFIG_SM_FTL=m
CONFIG_MTD_OOPS=m
# CONFIG_MTD_SWAP is not set

#
# RAM/ROM/Flash chip drivers
#
CONFIG_MTD_CFI=y
CONFIG_MTD_JEDECPROBE=y
CONFIG_MTD_GEN_PROBE=y
# CONFIG_MTD_CFI_ADV_OPTIONS is not set
CONFIG_MTD_MAP_BANK_WIDTH_1=y
CONFIG_MTD_MAP_BANK_WIDTH_2=y
CONFIG_MTD_MAP_BANK_WIDTH_4=y
# CONFIG_MTD_MAP_BANK_WIDTH_8 is not set
# CONFIG_MTD_MAP_BANK_WIDTH_16 is not set
# CONFIG_MTD_MAP_BANK_WIDTH_32 is not set
CONFIG_MTD_CFI_I1=y
CONFIG_MTD_CFI_I2=y
# CONFIG_MTD_CFI_I4 is not set
# CONFIG_MTD_CFI_I8 is not set
CONFIG_MTD_CFI_INTELEXT=y
CONFIG_MTD_CFI_AMDSTD=y
CONFIG_MTD_CFI_STAA=y
CONFIG_MTD_CFI_UTIL=y
# CONFIG_MTD_RAM is not set
# CONFIG_MTD_ROM is not set
# CONFIG_MTD_ABSENT is not set

#
# Mapping drivers for chip access
#
# CONFIG_MTD_COMPLEX_MAPPINGS is not set
# CONFIG_MTD_PHYSMAP is not set
# CONFIG_MTD_PHYSMAP_OF is not set
# CONFIG_MTD_IMPA7 is not set
# CONFIG_MTD_PLATRAM is not set

#
# Self-contained MTD device drivers
#
CONFIG_MTD_DATAFLASH=y
# CONFIG_MTD_DATAFLASH_WRITE_VERIFY is not set
# CONFIG_MTD_DATAFLASH_OTP is not set
CONFIG_MTD_M25P80=y
CONFIG_M25PXX_USE_FAST_READ=y
CONFIG_MTD_SST25L=y
# CONFIG_MTD_SLRAM is not set
# CONFIG_MTD_PHRAM is not set
# CONFIG_MTD_MTDRAM is not set
# CONFIG_MTD_BLOCK2MTD is not set

#
# Disk-On-Chip Device Drivers
#
# CONFIG_MTD_DOC2000 is not set
# CONFIG_MTD_DOC2001 is not set
# CONFIG_MTD_DOC2001PLUS is not set
# CONFIG_MTD_DOCG3 is not set
CONFIG_MTD_NAND_ECC=y
# CONFIG_MTD_NAND_ECC_SMC is not set
CONFIG_MTD_NAND=y
# CONFIG_MTD_NAND_ECC_BCH is not set
# CONFIG_MTD_SM_COMMON is not set
# CONFIG_MTD_NAND_MUSEUM_IDS is not set
# CONFIG_MTD_NAND_DENALI is not set
# CONFIG_MTD_NAND_GPIO is not set
CONFIG_MTD_NAND_IDS=y
# CONFIG_MTD_NAND_DISKONCHIP is not set
# CONFIG_MTD_NAND_DOCG4 is not set
# CONFIG_MTD_NAND_NANDSIM is not set
CONFIG_MTD_NAND_GPMI_NAND=y
# CONFIG_MTD_NAND_PLATFORM is not set
# CONFIG_MTD_ALAUDA is not set
CONFIG_MTD_NAND_MXC=y
# CONFIG_MTD_ONENAND is not set

#
# LPDDR flash memory drivers
#
# CONFIG_MTD_LPDDR is not set
CONFIG_MTD_UBI=y
CONFIG_MTD_UBI_WL_THRESHOLD=4096
CONFIG_MTD_UBI_BEB_LIMIT=20
# CONFIG_MTD_UBI_FASTMAP is not set
# CONFIG_MTD_UBI_GLUEBI is not set
CONFIG_DTC=y
CONFIG_OF=y

#
# Device Tree and Open Firmware support
#
CONFIG_PROC_DEVICETREE=y
# CONFIG_OF_SELFTEST is not set
CONFIG_OF_FLATTREE=y
CONFIG_OF_EARLY_FLATTREE=y
```

```
CONFIG_OF_ADDRESS=y
CONFIG_OF_IRQ=y
CONFIG_OF_DEVICE=y
CONFIG_OF_I2C=y
CONFIG_OF_NET=y
CONFIG_OF_MDIO=y
CONFIG_OF_MTD=y
# CONFIG_PARPORT is not set
CONFIG_BLK_DEV=y
# CONFIG_BLK_DEV_COW_COMMON is not set
CONFIG_BLK_DEV_LOOP=y
CONFIG_BLK_DEV_LOOP_MIN_COUNT=8
CONFIG_BLK_DEV_CRYPTOLOOP=m
CONFIG_BLK_DEV_DRBD=m
# CONFIG_DRBD_FAULT_INJECTION is not set
CONFIG_BLK_DEV_NBD=m
CONFIG_BLK_DEV_RAM=y
CONFIG_BLK_DEV_RAM_COUNT=16
CONFIG_BLK_DEV_RAM_SIZE=65536
# CONFIG_BLK_DEV_XIP is not set
CONFIG_CDROM_PKTCDVD=m
CONFIG_CDROM_PKTCDVD_BUFFERS=8
# CONFIG_CDROM_PKTCDVD_WCACHE is not set
CONFIG_ATA_OVER_ETH=m
CONFIG_MG_DISK=m
CONFIG_MG_DISK_RES=0
# CONFIG_BLK_DEV_RBD is not set

#
# Misc devices
#
# CONFIG_SENSORS_LIS3LV02D is not set
CONFIG_AD525X_DPOT=m
CONFIG_AD525X_DPOT_I2C=m
CONFIG_AD525X_DPOT_SPI=m
# CONFIG_ATMEL_PWM is not set
CONFIG_ICS932S401=m
CONFIG_ENCLOSURE_SERVICES=m
# CONFIG_APDS9802ALS is not set
CONFIG_ISL29003=m
# CONFIG_ISL29020 is not set
CONFIG_SENSORS_TSL2550=m
CONFIG_SENSORS_BH1780=m
# CONFIG_SENSORS_BH1770 is not set
# CONFIG_SENSORS_APDS990X is not set
CONFIG_HMC6352=m
CONFIG_DS1682=m
CONFIG_TI_DAC7512=m
# CONFIG_ARM_CHARLCD is not set
CONFIG_BMP085=y
CONFIG_BMP085_I2C=m
CONFIG_BMP085_SPI=m
# CONFIG_USB_SWITCH_FSA9480 is not set
# CONFIG_C2PORT is not set

#
# EEPROM support
#
CONFIG_EEPROM_AT24=m
CONFIG_EEPROM_AT25=m
CONFIG_EEPROM_LEGACY=m
CONFIG_EEPROM_MAX6875=m
```

```
CONFIG_EEPROM_93CX6=m
CONFIG_EEPROM_93XX46=m

#
# Texas Instruments shared transport line discipline
#
CONFIG_TI_ST=m
# CONFIG_SENSORS_LIS3_SPI is not set
# CONFIG_SENSORS_LIS3_I2C is not set

#
# Altera FPGA firmware download module
#
# CONFIG_ALTERA_STAPL is not set

#
# SCSI device support
#
CONFIG_SCSI_MOD=y
CONFIG_RAID_ATTRS=m
CONFIG_SCSI=y
CONFIG_SCSI_DMA=y
CONFIG_SCSI_TGT=m
CONFIG_SCSI_NETLINK=y
CONFIG_SCSI_PROC_FS=y

#
# SCSI support type (disk, tape, CD-ROM)
#
CONFIG_BLK_DEV_SD=y
CONFIG_CHR_DEV_ST=m
CONFIG_CHR_DEV_OSST=m
CONFIG_BLK_DEV_SR=y
# CONFIG_BLK_DEV_SR_VENDOR is not set
CONFIG_CHR_DEV_SG=y
CONFIG_CHR_DEV_SCH=m
CONFIG_SCSI_ENCLOSURE=m
CONFIG_SCSI_MULTI_LUN=y
CONFIG_SCSI_CONSTANTS=y
CONFIG_SCSI_LOGGING=y
CONFIG_SCSI_SCAN_ASYNC=y

#
# SCSI Transports
#
# CONFIG_SCSI_SPI_ATTRS is not set
CONFIG_SCSI_FC_ATTRS=m
CONFIG_SCSI_FC_TGT_ATTRS=y
CONFIG_SCSI_ISCSI_ATTRS=m
CONFIG_SCSI_SAS_ATTRS=m
CONFIG_SCSI_SAS_LIBSAS=m
CONFIG_SCSI_SAS_ATA=y
CONFIG_SCSI_SAS_HOST_SMP=y
CONFIG_SCSI_SRP_ATTRS=m
CONFIG_SCSI_SRP_TGT_ATTRS=y
CONFIG_SCSI_LOWLEVEL=y
CONFIG_ISCSI_TCP=m
# CONFIG_ISCSI_BOOT_SYSFS is not set
CONFIG_LIBFC=m
CONFIG_LIBFCOE=m
CONFIG_SCSI_DEBUG=m
CONFIG_SCSI_DH=y
```

```
CONFIG_SCSI_DH_RDAC=m
CONFIG_SCSI_DH_HP_SW=m
CONFIG_SCSI_DH_EMC=m
CONFIG_SCSI_DH_ALUA=m
# CONFIG_SCSI_OSD_INITIATOR is not set
CONFIG_HAVE_PATA_PLATFORM=y
CONFIG_ATA=y
# CONFIG_ATA_NONSTANDARD is not set
CONFIG_ATA_VERBOSE_ERROR=y
CONFIG_SATA_PMP=y

#
# Controllers with non-SFF native interface
#
CONFIG_SATA_AHCI_PLATFORM=y
CONFIG_ATA_SFF=y

#
# SFF controllers with custom DMA interface
#
CONFIG_ATA_BMDMA=y

#
# SATA SFF controllers with BMDMA
#
CONFIG_SATA_HIGHBANK=m
CONFIG_SATA_MV=m

#
# PATA SFF controllers with BMDMA
#
# CONFIG_PATA_ARASAN_CF is not set
CONFIG_PATA_IMX=y

#
# PIO-only SFF controllers
#
# CONFIG_PATA_PLATFORM is not set

#
# Generic fallback / legacy drivers
#
CONFIG_MD=y
CONFIG_BLK_DEV_MD=m
CONFIG_MD_LINEAR=m
CONFIG_MD_RAID0=m
CONFIG_MD_RAID1=m
CONFIG_MD_RAID10=m
CONFIG_MD_RAID456=m
# CONFIG_MULTICORE_RAID456 is not set
CONFIG_MD_MULTIPATH=m
CONFIG_MD_FAULTY=m
CONFIG_BLK_DEV_DM=y
CONFIG_DM_DEBUG=y
CONFIG_DM_CRYPT=m
CONFIG_DM_SNAPSHOT=m
# CONFIG_DM_THIN_PROVISIONING is not set
CONFIG_DM_MIRROR=m
CONFIG_DM_RAID=m
# CONFIG_DM_LOG_USERSPACE is not set
CONFIG_DM_ZERO=m
CONFIG_DM_MULTIPATH=m
```

```
CONFIG_DM_MULTIPATH_QL=m
CONFIG_DM_MULTIPATH_ST=m
# CONFIG_DM_DELAY is not set
# CONFIG_DM_UEVENT is not set
# CONFIG_DM_FLAKEY is not set
# CONFIG_DM_VERITY is not set
# CONFIG_TARGET_CORE is not set
CONFIG_NETDEVICES=y
CONFIG_NET_CORE=y
CONFIG_BONDING=m
CONFIG_DUMMY=m
CONFIG_EQUALIZER=m
CONFIG_MII=y
CONFIG_IFB=m
# CONFIG_NET_TEAM is not set
CONFIG_MACVLAN=y
CONFIG_MACVTAP=y
# CONFIG_VXLAN is not set
# CONFIG_NETCONSOLE is not set
# CONFIG_NETPOLL is not set
# CONFIG_NET_POLL_CONTROLLER is not set
CONFIG_TUN=m
CONFIG_VETH=y
CONFIG_ATM_DRIVERS=y
# CONFIG_ATM_DUMMY is not set
# CONFIG_ATM_TCP is not set

#
# CAIF transport drivers
#
# CONFIG_CAIF_TTY is not set
# CONFIG_CAIF_SPI_SLAVE is not set
# CONFIG_CAIF_HSI is not set

#
# Distributed Switch Architecture drivers
#
CONFIG_NET_DSA_MV88E6XXX=y
CONFIG_NET_DSA_MV88E6060=y
CONFIG_NET_DSA_MV88E6XXX_NEED_PPU=y
CONFIG_NET_DSA_MV88E6131=y
CONFIG_NET_DSA_MV88E6123_61_65=y
CONFIG_ETHERNET=y
CONFIG_NET_CADENCE=y
CONFIG_ARM_AT91_ETHER=y
CONFIG_MACB=y
CONFIG_NET_VENDOR_BROADCOM=y
# CONFIG_B44 is not set
# CONFIG_NET_CALXEDA_XGMAC is not set
CONFIG_NET_VENDOR_CIRRUS=y
# CONFIG_CS89x0 is not set
CONFIG_DM9000=m
CONFIG_DM9000_FORCE_SIMPLE_PHY_POLL=y
# CONFIG_DNET is not set
CONFIG_NET_VENDOR_FARADAY=y
# CONFIG_FTMAC100 is not set
# CONFIG_FTGMAC100 is not set
CONFIG_NET_VENDOR_FREESCALE=y
CONFIG_FEC=y
CONFIG_NET_VENDOR_INTEL=y
CONFIG_NET_VENDOR_I825XX=y
CONFIG_NET_VENDOR_MARVELL=y
```

```
# CONFIG_MVMDIO is not set
CONFIG_NET_VENDOR_MICREL=y
CONFIG_KS8842=m
CONFIG_KS8851=m
# CONFIG_KS8851_MLL is not set
CONFIG_NET_VENDOR_MICROCHIP=y
CONFIG_ENC28J60=m
# CONFIG_ENC28J60_WRITEVERIFY is not set
CONFIG_NET_VENDOR_NATSEMI=y
CONFIG_NET_VENDOR_8390=y
# CONFIG_AX88796 is not set
# CONFIG_ETHOC is not set
CONFIG_NET_VENDOR_SEEQ=y
# CONFIG_SEEQ8005 is not set
CONFIG_NET_VENDOR_SMSC=y
CONFIG_SMC91X=y
# CONFIG_SMC911X is not set
CONFIG_SMSC911X=y
# CONFIG_SMSC911X_ARCH_HOOKS is not set
CONFIG_NET_VENDOR_STMICRO=y
# CONFIG_STMMAC_ETH is not set
CONFIG_NET_VENDOR_WIZNET=y
CONFIG_WIZNET_W5100=m
CONFIG_WIZNET_W5300=m
# CONFIG_WIZNET_BUS_DIRECT is not set
# CONFIG_WIZNET_BUS_INDIRECT is not set
CONFIG_WIZNET_BUS_ANY=y
CONFIG_PHYLIB=y

#
# MII PHY device drivers
#
# CONFIG_AT803X_PHY is not set
# CONFIG_AMD_PHY is not set
CONFIG_MARVELL_PHY=y
CONFIG_DAVICOM_PHY=y
CONFIG_QSEMI_PHY=y
CONFIG_LXT_PHY=y
CONFIG_CICADA_PHY=y
CONFIG_VITESSE_PHY=y
CONFIG_SMSC_PHY=y
CONFIG_BROADCOM_PHY=y
# CONFIG_BCM87XX_PHY is not set
CONFIG_ICPLUS_PHY=y
CONFIG_REALTEK_PHY=y
CONFIG_NATIONAL_PHY=y
CONFIG_STE10XP=y
CONFIG_LSI_ET1011C_PHY=y
CONFIG_MICREL_PHY=y
# CONFIG_FIXED_PHY is not set
# CONFIG_MDIO_BITBANG is not set
# CONFIG_MDIO_BUS_MUX_GPIO is not set
# CONFIG_MDIO_BUS_MUX_MMIOREG is not set
# CONFIG_MICREL_KS8995MA is not set
CONFIG_PPP=m
CONFIG_PPP_BSDCOMP=m
CONFIG_PPP_DEFLATE=m
CONFIG_PPP_FILTER=y
CONFIG_PPP_MPPE=m
CONFIG_PPP_MULTILINK=y
CONFIG_PPPOATM=m
CONFIG_PPPOE=m
```

```
CONFIG_PPPOL2TP=m
CONFIG_PPP_ASYNC=m
CONFIG_PPP_SYNC_TTY=m
CONFIG_SLIP=m
CONFIG_SLHC=m
CONFIG_SLIP_COMPRESSED=y
CONFIG_SLIP_SMART=y
CONFIG_SLIP_MODE_SLIP6=y

#
# USB Network Adapters
#
CONFIG_USB_CATC=m
CONFIG_USB_KAWETH=m
CONFIG_USB_PEGASUS=m
CONFIG_USB_RTL8150=m
CONFIG_USB_USBNET=y
CONFIG_USB_NET_AX8817X=m
CONFIG_USB_NET_CDCETHER=y
CONFIG_USB_NET_CDC_EEM=m
CONFIG_USB_NET_CDC_NCM=y
CONFIG_USB_NET_CDC_MBIM=m
CONFIG_USB_NET_DM9601=m
CONFIG_USB_NET_SMSC75XX=m
CONFIG_USB_NET_SMSC95XX=m
CONFIG_USB_NET_GL620A=m
CONFIG_USB_NET_NET1080=m
CONFIG_USB_NET_PLUSB=m
CONFIG_USB_NET_MCS7830=m
CONFIG_USB_NET_RNDIS_HOST=y
CONFIG_USB_NET_CDC_SUBSET=m
CONFIG_USB_ALI_M5632=y
CONFIG_USB_AN2720=y
CONFIG_USB_BELKIN=y
CONFIG_USB_ARMLINUX=y
CONFIG_USB_EPSON2888=y
CONFIG_USB_KC2190=y
CONFIG_USB_NET_ZAURUS=m
CONFIG_USB_NET_CX82310_ETH=m
CONFIG_USB_NET_KALMIA=m
CONFIG_USB_NET_QMI_WWAN=m
CONFIG_USB_HSO=m
CONFIG_USB_NET_INT51X1=m
CONFIG_USB_CDC_PHONET=m
CONFIG_USB_IPHETH=m
CONFIG_USB_SIERRA_NET=m
CONFIG_USB_VL600=m
CONFIG_WLAN=y
CONFIG_LIBERTAS_THINFIRM=m
# CONFIG_LIBERTAS_THINFIRM_DEBUG is not set
CONFIG_LIBERTAS_THINFIRM_USB=m
CONFIG_AT76C50X_USB=m
CONFIG_USB_ZD1201=m
CONFIG_USB_NET_RNDIS_WLAN=m
CONFIG_RTL8187=m
CONFIG_RTL8187_LEDS=y
CONFIG_MAC80211_HWSIM=m
CONFIG_ATH_COMMON=m
CONFIG_ATH_CARDS=m
# CONFIG_ATH_DEBUG is not set
CONFIG_ATH9K_HW=m
CONFIG_ATH9K_COMMON=m
```

```
CONFIG_ATH9K_BTCOEX_SUPPORT=y
CONFIG_ATH9K=m
# CONFIG_ATH9K_AHB is not set
# CONFIG_ATH9K_DEBUGFS is not set
CONFIG_ATH9K_RATE_CONTROL=y
CONFIG_ATH9K_HTC=m
# CONFIG_ATH9K_HTC_DEBUGFS is not set
CONFIG_CARL9170=m
CONFIG_CARL9170_LEDS=y
# CONFIG_CARL9170_DEBUGFS is not set
CONFIG_CARL9170_WPC=y
# CONFIG_CARL9170_HWRNG is not set
CONFIG_ATH6KL=m
# CONFIG_ATH6KL_SDIO is not set
CONFIG_ATH6KL_USB=m
# CONFIG_ATH6KL_DEBUG is not set
CONFIG_AR5523=m
CONFIG_B43=m
CONFIG_B43_SSB=y
# CONFIG_B43_SDIO is not set
CONFIG_B43_PIO=y
CONFIG_B43_PHY_N=y
CONFIG_B43_PHY_LP=y
# CONFIG_B43_PHY_HT is not set
CONFIG_B43_LEDS=y
CONFIG_B43_HWRNG=y
# CONFIG_B43_DEBUG is not set
CONFIG_B43LEGACY=m
CONFIG_B43LEGACY_LEDS=y
CONFIG_B43LEGACY_HWRNG=y
CONFIG_B43LEGACY_DEBUG=y
CONFIG_B43LEGACY_DMA=y
CONFIG_B43LEGACY_PIO=y
CONFIG_B43LEGACY_DMA_AND_PIO_MODE=y
# CONFIG_B43LEGACY_DMA_MODE is not set
# CONFIG_B43LEGACY_PIO_MODE is not set
# CONFIG_BRCMFMAC is not set
# CONFIG_HOSTAP is not set
CONFIG_LIBERTAS=m
CONFIG_LIBERTAS_USB=m
CONFIG_LIBERTAS_SDIO=m
CONFIG_LIBERTAS_SPI=m
CONFIG_LIBERTAS_DEBUG=y
# CONFIG_LIBERTAS_MESH is not set
CONFIG_P54_COMMON=m
CONFIG_P54_USB=m
CONFIG_P54_SPI=m
# CONFIG_P54_SPI_DEFAULT_EEPROM is not set
CONFIG_P54_LEDS=y
CONFIG_RT2X00=m
CONFIG_RT2500USB=m
CONFIG_RT73USB=m
CONFIG_RT2800USB=m
CONFIG_RT2800USB_RT33XX=y
CONFIG_RT2800USB_RT35XX=y
CONFIG_RT2800USB_RT53XX=y
CONFIG_RT2800USB_UNKNOWN=y
CONFIG_RT2800_LIB=m
CONFIG_RT2X00_LIB_USB=m
CONFIG_RT2X00_LIB=m
CONFIG_RT2X00_LIB_FIRMWARE=y
CONFIG_RT2X00_LIB_CRYPTO=y
```

```
CONFIG_RT2X00_LIB_LEDS=y
# CONFIG_RT2X00_LIB_DEBUGFS is not set
# CONFIG_RT2X00_DEBUG is not set
CONFIG_RTL8192CU=m
CONFIG_RTLWIFI=m
CONFIG_RTLWIFI_DEBUG=y
CONFIG_RTL8192C_COMMON=m
CONFIG_WL_TI=y
CONFIG_WL1251=m
CONFIG_WL1251_SPI=m
CONFIG_WL1251_SDIO=m
CONFIG_WL12XX=m
CONFIG_WL18XX=m
CONFIG_WLCORE=m
CONFIG_WLCORE_SPI=m
CONFIG_WLCORE_SDIO=m
CONFIG_WL12XX_PLATFORM_DATA=y
CONFIG_ZD1211RW=m
# CONFIG_ZD1211RW_DEBUG is not set
CONFIG_MWIFIEX=m
CONFIG_MWIFIEX_SDIO=m
CONFIG_MWIFIEX_USB=m

#
# WiMAX Wireless Broadband devices
#
# CONFIG_WIMAX_I2400M_USB is not set
# CONFIG_WAN is not set
CONFIG_IEEE802154_DRIVERS=m
CONFIG_IEEE802154_FAKEHARD=m
CONFIG_IEEE802154_FAKELB=m
CONFIG_IEEE802154_AT86RF230=m
CONFIG_IEEE802154_MRF24J40=m
# CONFIG_ISDN is not set

#
# Input device support
#
CONFIG_INPUT=y
CONFIG_INPUT_FF_MEMLESS=m
CONFIG_INPUT_POLLDEV=y
CONFIG_INPUT_SPARSEKMAP=m
CONFIG_INPUT_MATRIXKMAP=m

#
# Userland interfaces
#
CONFIG_INPUT_MOUSEDEV=y
CONFIG_INPUT_MOUSEDEV_PSAUX=y
CONFIG_INPUT_MOUSEDEV_SCREEN_X=1024
CONFIG_INPUT_MOUSEDEV_SCREEN_Y=768
CONFIG_INPUT_JOYDEV=m
CONFIG_INPUT_EVDEV=y
# CONFIG_INPUT_EVBUG is not set

#
# Input Device Drivers
#
CONFIG_INPUT_KEYBOARD=y
CONFIG_KEYBOARD_ADP5588=m
CONFIG_KEYBOARD_ADP5589=m
CONFIG_KEYBOARD_ATKBD=y
```

```
CONFIG_KEYBOARD_QT1070=m
CONFIG_KEYBOARD_QT2160=m
CONFIG_KEYBOARD_LKKBD=m
CONFIG_KEYBOARD_GPIO=m
CONFIG_KEYBOARD_TCA6416=m
CONFIG_KEYBOARD_TCA8418=m
CONFIG_KEYBOARD_MATRIX=m
CONFIG_KEYBOARD_LM8323=m
CONFIG_KEYBOARD_LM8333=m
CONFIG_KEYBOARD_MAX7359=m
CONFIG_KEYBOARD_MCS=m
CONFIG_KEYBOARD_MPR121=m
CONFIG_KEYBOARD_IMX=m
CONFIG_KEYBOARD_NEWTON=m
CONFIG_KEYBOARD_OPENCORES=m
CONFIG_KEYBOARD_SAMSUNG=m
CONFIG_KEYBOARD_STOWAWAY=m
CONFIG_KEYBOARD_SUNKBD=m
CONFIG_KEYBOARD_XTKBD=m
CONFIG_INPUT_MOUSE=y
CONFIG_MOUSE_PS2=m
CONFIG_MOUSE_PS2_ALPS=y
CONFIG_MOUSE_PS2_LOGIPS2PP=y
CONFIG_MOUSE_PS2_SYNAPTICS=y
CONFIG_MOUSE_PS2_TRACKPOINT=y
CONFIG_MOUSE_PS2_ELANTECH=y
CONFIG_MOUSE_PS2_SENTELIC=y
CONFIG_MOUSE_PS2_TOUCHKIT=y
CONFIG_MOUSE_SERIAL=m
CONFIG_MOUSE_APPLETOUCH=m
CONFIG_MOUSE_BCM5974=m
CONFIG_MOUSE_VSXXXAA=m
CONFIG_MOUSE_GPIO=m
CONFIG_MOUSE_SYNAPTICS_I2C=m
CONFIG_MOUSE_SYNAPTICS_USB=m
CONFIG_INPUT_JOYSTICK=y
CONFIG_JOYSTICK_ANALOG=m
CONFIG_JOYSTICK_A3D=m
CONFIG_JOYSTICK_ADI=m
CONFIG_JOYSTICK_COBRA=m
CONFIG_JOYSTICK_GF2K=m
CONFIG_JOYSTICK_GRIP=m
CONFIG_JOYSTICK_GRIP_MP=m
CONFIG_JOYSTICK_GUILLEMOT=m
CONFIG_JOYSTICK_INTERACT=m
CONFIG_JOYSTICK_SIDEWINDER=m
CONFIG_JOYSTICK_TMDC=m
CONFIG_JOYSTICK_IFORCE=m
CONFIG_JOYSTICK_IFORCE_USB=y
CONFIG_JOYSTICK_IFORCE_232=y
CONFIG_JOYSTICK_WARRIOR=m
CONFIG_JOYSTICK_MAGELLAN=m
CONFIG_JOYSTICK_SPACEORB=m
CONFIG_JOYSTICK_SPACEBALL=m
CONFIG_JOYSTICK_STINGER=m
CONFIG_JOYSTICK_TWIDJOY=m
CONFIG_JOYSTICK_ZHENHUA=m
CONFIG_JOYSTICK_AS5011=m
CONFIG_JOYSTICK_JOYDUMP=m
CONFIG_JOYSTICK_XPAD=m
CONFIG_JOYSTICK_XPAD_FF=y
CONFIG_JOYSTICK_XPAD_LEDS=y
```

```
CONFIG_INPUT_TABLET=y
CONFIG_TABLET_USB_ACECAD=m
CONFIG_TABLET_USB_AIPTEK=m
CONFIG_TABLET_USB_GTCO=m
CONFIG_TABLET_USB_HANWANG=m
CONFIG_TABLET_USB_KBTAB=m
CONFIG_TABLET_USB_WACOM=m
CONFIG_INPUT_TOUCHSCREEN=y
CONFIG_TOUCHSCREEN_ADS7846=m
CONFIG_TOUCHSCREEN_AD7877=m
CONFIG_TOUCHSCREEN_AD7879=m
CONFIG_TOUCHSCREEN_AD7879_I2C=m
CONFIG_TOUCHSCREEN_AD7879_SPI=m
CONFIG_TOUCHSCREEN_ATMEL_MXT=m
CONFIG_TOUCHSCREEN_AUO_PIXCIR=m
CONFIG_TOUCHSCREEN_BU21013=m
CONFIG_TOUCHSCREEN_CY8CTMG110=m
CONFIG_TOUCHSCREEN_CYTTSP_CORE=m
CONFIG_TOUCHSCREEN_CYTTSP_I2C=m
CONFIG_TOUCHSCREEN_CYTTSP_SPI=m
CONFIG_TOUCHSCREEN_DYNAPRO=m
CONFIG_TOUCHSCREEN_HAMPSHIRE=m
CONFIG_TOUCHSCREEN_EETI=m
# CONFIG_TOUCHSCREEN_EGALAX is not set
CONFIG_TOUCHSCREEN_FUJITSU=m
CONFIG_TOUCHSCREEN_ILI210X=m
CONFIG_TOUCHSCREEN_GUNZE=m
CONFIG_TOUCHSCREEN_ELO=m
CONFIG_TOUCHSCREEN_WACOM_W8001=m
CONFIG_TOUCHSCREEN_WACOM_I2C=m
CONFIG_TOUCHSCREEN_MAX11801=m
CONFIG_TOUCHSCREEN_MCS5000=m
CONFIG_TOUCHSCREEN_MMS114=m
CONFIG_TOUCHSCREEN_MTOUCH=m
CONFIG_TOUCHSCREEN_INEXIO=m
CONFIG_TOUCHSCREEN_MK712=m
CONFIG_TOUCHSCREEN_PENMOUNT=m
CONFIG_TOUCHSCREEN_EDT_FT5X06=m
CONFIG_TOUCHSCREEN_TOUCHRIGHT=m
CONFIG_TOUCHSCREEN_TOUCHWIN=m
CONFIG_TOUCHSCREEN_PIXCIR=m
CONFIG_TOUCHSCREEN_WM97XX=m
CONFIG_TOUCHSCREEN_WM9705=y
CONFIG_TOUCHSCREEN_WM9712=y
CONFIG_TOUCHSCREEN_WM9713=y
CONFIG_TOUCHSCREEN_USB_COMPOSITE=m
# CONFIG_TOUCHSCREEN_MC13783 is not set
CONFIG_TOUCHSCREEN_USB_EGALAX=y
CONFIG_TOUCHSCREEN_USB_PANJIT=y
CONFIG_TOUCHSCREEN_USB_3M=y
CONFIG_TOUCHSCREEN_USB_ITM=y
CONFIG_TOUCHSCREEN_USB_ETURBO=y
CONFIG_TOUCHSCREEN_USB_GUNZE=y
CONFIG_TOUCHSCREEN_USB_DMC_TSC10=y
CONFIG_TOUCHSCREEN_USB_IRTOUCH=y
CONFIG_TOUCHSCREEN_USB_IDEALTEK=y
CONFIG_TOUCHSCREEN_USB_GENERAL_TOUCH=y
CONFIG_TOUCHSCREEN_USB_GOTOP=y
CONFIG_TOUCHSCREEN_USB_JASTEC=y
CONFIG_TOUCHSCREEN_USB_ELO=y
CONFIG_TOUCHSCREEN_USB_E2I=y
CONFIG_TOUCHSCREEN_USB_ZYTRONIC=y
```

```
CONFIG_TOUCHSCREEN_USB_ETT_TC45USB=y
CONFIG_TOUCHSCREEN_USB_NEXIO=y
CONFIG_TOUCHSCREEN_USB_EASYTOUCH=y
CONFIG_TOUCHSCREEN_TOUCHIT213=m
CONFIG_TOUCHSCREEN_TSC_SERIO=m
CONFIG_TOUCHSCREEN_TSC2005=m
CONFIG_TOUCHSCREEN_TSC2007=m
CONFIG_TOUCHSCREEN_W90X900=m
CONFIG_TOUCHSCREEN_ST1232=m
CONFIG_TOUCHSCREEN_TPS6507X=m
CONFIG_INPUT_MISC=y
CONFIG_INPUT_AD714X=m
CONFIG_INPUT_AD714X_I2C=m
CONFIG_INPUT_AD714X_SPI=m
CONFIG_INPUT_BMA150=m
# CONFIG_INPUT_MC13783_PWRBUTTON is not set
CONFIG_INPUT_MMA8450=y
CONFIG_INPUT_MPU3050=m
CONFIG_INPUT_GP2A=m
CONFIG_INPUT_GPIO_TILT_POLLED=m
CONFIG_INPUT_ATI_REMOTE2=m
CONFIG_INPUT_KEYSPAN_REMOTE=m
CONFIG_INPUT_KXTJ9=m
# CONFIG_INPUT_KXTJ9_POLLED_MODE is not set
CONFIG_INPUT_POWERMATE=m
CONFIG_INPUT_YEALINK=m
CONFIG_INPUT_CM109=m
CONFIG_INPUT_UINPUT=y
CONFIG_INPUT_PCF8574=m
# CONFIG_INPUT_PWM_BEEPER is not set
CONFIG_INPUT_GPIO_ROTARY_ENCODER=m
CONFIG_INPUT_ADXL34X=m
CONFIG_INPUT_ADXL34X_I2C=m
CONFIG_INPUT_ADXL34X_SPI=m
CONFIG_INPUT_CMA3000=m
CONFIG_INPUT_CMA3000_I2C=m

#
# Hardware I/O ports
#
CONFIG_SERIO=y
CONFIG_SERIO_SERPORT=m
# CONFIG_SERIO_AMBAKMI is not set
CONFIG_SERIO_LIBPS2=y
# CONFIG_SERIO_RAW is not set
# CONFIG_SERIO_ALTERA_PS2 is not set
# CONFIG_SERIO_PS2MULT is not set
# CONFIG_SERIO_ARC_PS2 is not set
CONFIG_GAMEPORT=m
# CONFIG_GAMEPORT_NS558 is not set
# CONFIG_GAMEPORT_L4 is not set

#
# Character devices
#
CONFIG_VT=y
CONFIG_CONSOLE_TRANSLATIONS=y
CONFIG_VT_CONSOLE=y
CONFIG_VT_CONSOLE_SLEEP=y
CONFIG_HW_CONSOLE=y
CONFIG_VT_HW_CONSOLE_BINDING=y
CONFIG_UNIX98_PTYS=y
```

```
CONFIG_DEVPTS_MULTIPLE_INSTANCES=y
# CONFIG_LEGACY_PTYS is not set
# CONFIG_SERIAL_NONSTANDARD is not set
# CONFIG_N_GSM is not set
# CONFIG_TRACE_SINK is not set
CONFIG_DEVKMEM=y

#
# Serial drivers
#
# CONFIG_SERIAL_8250 is not set

#
# Non-8250 serial port support
#
# CONFIG_SERIAL_AMBA_PL010 is not set
# CONFIG_SERIAL_AMBA_PL011 is not set
# CONFIG_SERIAL_KGDB_NMI is not set
# CONFIG_SERIAL_MAX3100 is not set
# CONFIG_SERIAL_MAX310X is not set
CONFIG_SERIAL_IMX=y
CONFIG_SERIAL_IMX_CONSOLE=y
CONFIG_SERIAL_CORE=y
CONFIG_SERIAL_CORE_CONSOLE=y
CONFIG_CONSOLE_POLL=y
# CONFIG_SERIAL_SCCNXP is not set
# CONFIG_SERIAL_TIMBERDALE is not set
# CONFIG_SERIAL_ALTERA_JTAGUART is not set
# CONFIG_SERIAL_ALTERA_UART is not set
# CONFIG_SERIAL_IFX6X60 is not set
# CONFIG_SERIAL_XILINX_PS_UART is not set
# CONFIG_SERIAL_ARC is not set
# CONFIG_TTY_PRINTK is not set
# CONFIG_HVC_DCC is not set
# CONFIG_IPMI_HANDLER is not set
CONFIG_HW_RANDOM=y
# CONFIG_HW_RANDOM_TIMERIOMEM is not set
# CONFIG_HW_RANDOM_ATMEL is not set
# CONFIG_HW_RANDOM_EXYNOS is not set
# CONFIG_NVRAM is not set
# CONFIG_R3964 is not set
# CONFIG_RAW_DRIVER is not set
# CONFIG_TCG_TPM is not set
CONFIG_I2C=y
CONFIG_I2C_BOARDINFO=y
CONFIG_I2C_COMPAT=y
CONFIG_I2C_CHARDEV=y
# CONFIG_I2C_MUX is not set
CONFIG_I2C_HELPER_AUTO=y
CONFIG_I2C_SMBUS=m
CONFIG_I2C_ALGOBIT=y

#
# I2C Hardware Bus support
#

#
# I2C system bus drivers (mostly embedded / system-on-chip)
#
# CONFIG_I2C_CBUS_GPIO is not set
# CONFIG_I2C_DESIGNWARE_PLATFORM is not set
# CONFIG_I2C_GPIO is not set
```

```
CONFIG_I2C_IMX=y
# CONFIG_I2C_NOMADIK is not set
# CONFIG_I2C_OCORES is not set
# CONFIG_I2C_PCA_PLATFORM is not set
# CONFIG_I2C_PXA_PCI is not set
# CONFIG_I2C_SIMTEC is not set
# CONFIG_I2C_VERSATILE is not set
# CONFIG_I2C_XILINX is not set

#
# External I2C/SMBus adapter drivers
#
CONFIG_I2C_DIOLAN_U2C=m
CONFIG_I2C_PARPORT_LIGHT=m
CONFIG_I2C_TAOS_EVM=m
CONFIG_I2C_TINY_USB=m

#
# Other I2C/SMBus bus drivers
#
CONFIG_I2C_STUB=m
# CONFIG_I2C_DEBUG_CORE is not set
# CONFIG_I2C_DEBUG_ALGO is not set
# CONFIG_I2C_DEBUG_BUS is not set
CONFIG_SPI=y
# CONFIG_SPI_DEBUG is not set
CONFIG_SPI_MASTER=y

#
# SPI Master Controller Drivers
#
# CONFIG_SPI_ALTERA is not set
CONFIG_SPI_BITBANG=y
# CONFIG_SPI_GPIO is not set
CONFIG_SPI_IMX=y
# CONFIG_SPI_OC_TINY is not set
# CONFIG_SPI_PL022 is not set
# CONFIG_SPI_PXA2XX_PCI is not set
# CONFIG_SPI_SC18IS602 is not set
CONFIG_SPI_XCOMM=m
# CONFIG_SPI_XILINX is not set
# CONFIG_SPI_DESIGNWARE is not set

#
# SPI Protocol Masters
#
CONFIG_SPI_SPIDEV=y
# CONFIG_SPI_TLE62X0 is not set
# CONFIG_HSI is not set

#
# PPS support
#
# CONFIG_PPS is not set

#
# PPS generators support
#

#
# PTP clock support
#
```

```
# CONFIG_PTP_1588_CLOCK is not set

#
# Enable PHYLIB and NETWORK_PHY_TIMESTAMPING to see the additional clocks.
#
# CONFIG_PTP_1588_CLOCK_PCH is not set
CONFIG_PINCTRL=y

#
# Pin controllers
#
CONFIG_PINMUX=y
CONFIG_PINCONF=y
# CONFIG_DEBUG_PINCTRL is not set
CONFIG_PINCTRL_IMX=y
CONFIG_PINCTRL_IMX51=y
CONFIG_PINCTRL_IMX53=y
CONFIG_PINCTRL_IMX6Q=y
# CONFIG_PINCTRL_SINGLE is not set
# CONFIG_PINCTRL_EXYNOS is not set
# CONFIG_PINCTRL_EXYNOS5440 is not set
CONFIG_ARCH_HAVE_CUSTOM_GPIO_H=y
CONFIG_ARCH_REQUIRE_GPIOLIB=y
CONFIG_GPIOLIB=y
CONFIG_OF_GPIO=y
CONFIG_DEBUG_GPIO=y
CONFIG_GPIO_SYSFS=y
CONFIG_GPIO_GENERIC=y

#
# Memory mapped GPIO drivers:
#
# CONFIG_GPIO_GENERIC_PLATFORM is not set
# CONFIG_GPIO_EM is not set
CONFIG_GPIO_MXC=y
# CONFIG_GPIO_PL061 is not set
# CONFIG_GPIO_TS5500 is not set

#
# I2C GPIO expanders:
#
# CONFIG_GPIO_MAX7300 is not set
# CONFIG_GPIO_MAX732X is not set
# CONFIG_GPIO_PCF857X is not set
# CONFIG_GPIO_SX150X is not set
# CONFIG_GPIO_ADP5588 is not set
# CONFIG_GPIO_ADNP is not set

#
# PCI GPIO expanders:
#

#
# SPI GPIO expanders:
#
# CONFIG_GPIO_MAX7301 is not set
# CONFIG_GPIO_MCP23S08 is not set
# CONFIG_GPIO_MC33880 is not set
# CONFIG_GPIO_74X164 is not set

#
# AC97 GPIO expanders:
```

```
#

#
# MODULbus GPIO expanders:
#

#
# USB GPIO expanders:
#
CONFIG_W1=y
CONFIG_W1_CON=y

#
# 1-wire Bus Masters
#
# CONFIG_W1_MASTER_DS2490 is not set
# CONFIG_W1_MASTER_DS2482 is not set
CONFIG_W1_MASTER_MXC=m
# CONFIG_W1_MASTER_DS1WM is not set
CONFIG_W1_MASTER_GPIO=m

#
# 1-wire Slaves
#
CONFIG_W1_SLAVE_THERM=m
CONFIG_W1_SLAVE_SMEM=m
CONFIG_W1_SLAVE_DS2408=m
CONFIG_W1_SLAVE_DS2423=m
CONFIG_W1_SLAVE_DS2431=m
CONFIG_W1_SLAVE_DS2433=m
CONFIG_W1_SLAVE_DS2433_CRC=y
CONFIG_W1_SLAVE_DS2760=m
CONFIG_W1_SLAVE_DS2780=m
CONFIG_W1_SLAVE_DS2781=m
CONFIG_W1_SLAVE_DS28E04=m
CONFIG_W1_SLAVE_BQ27000=m
CONFIG_POWER_SUPPLY=y
# CONFIG_POWER_SUPPLY_DEBUG is not set
# CONFIG_PDA_POWER is not set
CONFIG_GENERIC_ADC_BATTERY=m
# CONFIG_TEST_POWER is not set
# CONFIG_BATTERY_DS2760 is not set
# CONFIG_BATTERY_DS2780 is not set
# CONFIG_BATTERY_DS2781 is not set
# CONFIG_BATTERY_DS2782 is not set
# CONFIG_BATTERY_SBS is not set
# CONFIG_BATTERY_BQ27x00 is not set
# CONFIG_BATTERY_MAX17040 is not set
# CONFIG_BATTERY_MAX17042 is not set
# CONFIG_CHARGER_ISP1704 is not set
# CONFIG_CHARGER_MAX8903 is not set
# CONFIG_CHARGER_LP8727 is not set
CONFIG_CHARGER_GPIO=m
# CONFIG_CHARGER_MANAGER is not set
# CONFIG_CHARGER_BQ2415X is not set
# CONFIG_CHARGER_SMB347 is not set
# CONFIG_POWER_RESET is not set
CONFIG_POWER_AVS=y
CONFIG_HWMON=y
CONFIG_HWMON_VID=m
# CONFIG_HWMON_DEBUG_CHIP is not set
```

```
#
# Native drivers
#
CONFIG_SENSORS_AD7314=m
CONFIG_SENSORS_AD7414=m
CONFIG_SENSORS_AD7418=m
CONFIG_SENSORS_ADCXX=m
CONFIG_SENSORS_ADM1021=m
CONFIG_SENSORS_ADM1025=m
CONFIG_SENSORS_ADM1026=m
CONFIG_SENSORS_ADM1029=m
CONFIG_SENSORS_ADM1031=m
CONFIG_SENSORS_ADM9240=m
CONFIG_SENSORS_ADT7410=m
CONFIG_SENSORS_ADT7411=m
CONFIG_SENSORS_ADT7462=m
CONFIG_SENSORS_ADT7470=m
CONFIG_SENSORS_ADT7475=m
CONFIG_SENSORS_ASC7621=m
CONFIG_SENSORS_ATXP1=m
CONFIG_SENSORS_DS620=m
CONFIG_SENSORS_DS1621=m
CONFIG_SENSORS_F71805F=m
CONFIG_SENSORS_F71882FG=m
CONFIG_SENSORS_F75375S=m
CONFIG_SENSORS_G760A=m
CONFIG_SENSORS_GL518SM=m
CONFIG_SENSORS_GL520SM=m
CONFIG_SENSORS_GPIO_FAN=m
CONFIG_SENSORS_HIH6130=m
CONFIG_SENSORS_IT87=m
CONFIG_SENSORS_JC42=m
CONFIG_SENSORS_LINEAGE=m
CONFIG_SENSORS_LM63=m
CONFIG_SENSORS_LM70=m
CONFIG_SENSORS_LM73=m
CONFIG_SENSORS_LM75=m
CONFIG_SENSORS_LM77=m
CONFIG_SENSORS_LM78=m
CONFIG_SENSORS_LM80=m
CONFIG_SENSORS_LM83=m
CONFIG_SENSORS_LM85=m
CONFIG_SENSORS_LM87=m
CONFIG_SENSORS_LM90=m
CONFIG_SENSORS_LM92=m
CONFIG_SENSORS_LM93=m
CONFIG_SENSORS_LTC4151=m
CONFIG_SENSORS_LTC4215=m
CONFIG_SENSORS_LTC4245=m
CONFIG_SENSORS_LTC4261=m
CONFIG_SENSORS_LM95241=m
CONFIG_SENSORS_LM95245=m
CONFIG_SENSORS_MAX1111=m
CONFIG_SENSORS_MAX16065=m
CONFIG_SENSORS_MAX1619=m
CONFIG_SENSORS_MAX1668=m
CONFIG_SENSORS_MAX197=m
CONFIG_SENSORS_MAX6639=m
CONFIG_SENSORS_MAX6642=m
CONFIG_SENSORS_MAX6650=m
CONFIG_SENSORS_MCP3021=m
CONFIG_SENSORS_NTC_THERMISTOR=m
```

```
CONFIG_SENSORS_PC87360=m
CONFIG_SENSORS_PC87427=m
CONFIG_SENSORS_PCF8591=m
CONFIG_PMBUS=m
CONFIG_SENSORS_PMBUS=m
CONFIG_SENSORS_ADM1275=m
CONFIG_SENSORS_LM25066=m
CONFIG_SENSORS_LTC2978=m
CONFIG_SENSORS_MAX16064=m
CONFIG_SENSORS_MAX34440=m
CONFIG_SENSORS_MAX8688=m
CONFIG_SENSORS_UCD9000=m
CONFIG_SENSORS_UCD9200=m
CONFIG_SENSORS_ZL6100=m
CONFIG_SENSORS_SHT15=m
CONFIG_SENSORS_SHT21=m
CONFIG_SENSORS_SMM665=m
CONFIG_SENSORS_DME1737=m
CONFIG_SENSORS_EMC1403=m
CONFIG_SENSORS_EMC2103=m
CONFIG_SENSORS_EMC6W201=m
CONFIG_SENSORS_SMSC47M1=m
CONFIG_SENSORS_SMSC47M192=m
CONFIG_SENSORS_SMSC47B397=m
CONFIG_SENSORS_SCH56XX_COMMON=m
CONFIG_SENSORS_SCH5627=m
CONFIG_SENSORS_SCH5636=m
CONFIG_SENSORS_ADS1015=m
CONFIG_SENSORS_ADS7828=m
CONFIG_SENSORS_ADS7871=m
CONFIG_SENSORS_AMC6821=m
CONFIG_SENSORS_INA2XX=m
CONFIG_SENSORS_THMC50=m
CONFIG_SENSORS_TMP102=m
CONFIG_SENSORS_TMP401=m
CONFIG_SENSORS_TMP421=m
# CONFIG_SENSORS_VEXPRESS is not set
CONFIG_SENSORS_VT1211=m
CONFIG_SENSORS_W83781D=m
CONFIG_SENSORS_W83791D=m
CONFIG_SENSORS_W83792D=m
CONFIG_SENSORS_W83793=m
CONFIG_SENSORS_W83795=m
# CONFIG_SENSORS_W83795_FANCTRL is not set
CONFIG_SENSORS_W83L785TS=m
CONFIG_SENSORS_W83L786NG=m
CONFIG_SENSORS_W83627HF=m
CONFIG_SENSORS_W83627EHF=m
# CONFIG_SENSORS_MC13783_ADC is not set
CONFIG_THERMAL=y
CONFIG_THERMAL_HWMON=y
CONFIG_THERMAL_DEFAULT_GOV_STEP_WISE=y
# CONFIG_THERMAL_DEFAULT_GOV_FAIR_SHARE is not set
# CONFIG_THERMAL_DEFAULT_GOV_USER_SPACE is not set
# CONFIG_FAIR_SHARE is not set
CONFIG_STEP_WISE=y
# CONFIG_USER_SPACE is not set
CONFIG_CPU_THERMAL=y
CONFIG_WATCHDOG=y
CONFIG_WATCHDOG_CORE=y
CONFIG_WATCHDOG_NOWAYOUT=y
```

```
#
# Watchdog Device Drivers
#
# CONFIG_SOFT_WATCHDOG is not set
# CONFIG_ARM_SP805_WATCHDOG is not set
# CONFIG_DW_WATCHDOG is not set
# CONFIG_MPCORE_WATCHDOG is not set
# CONFIG_MAX63XX_WATCHDOG is not set
CONFIG_IMX2_WDT=y

#
# USB-based Watchdog Cards
#
CONFIG_USBPCWATCHDOG=m
CONFIG_SSB_POSSIBLE=y

#
# Sonics Silicon Backplane
#
CONFIG_SSB=m
CONFIG_SSB_BLOCKIO=y
CONFIG_SSB_SDIOHOST_POSSIBLE=y
# CONFIG_SSB_SDIOHOST is not set
# CONFIG_SSB_SILENT is not set
# CONFIG_SSB_DEBUG is not set
# CONFIG_SSB_DRIVER_GPIO is not set
CONFIG_BCMA_POSSIBLE=y

#
# Broadcom specific AMBA
#
# CONFIG_BCMA is not set

#
# Multifunction device drivers
#
CONFIG_MFD_CORE=y
# CONFIG_MFD_88PM860X is not set
# CONFIG_MFD_88PM800 is not set
# CONFIG_MFD_88PM805 is not set
# CONFIG_MFD_SM501 is not set
# CONFIG_MFD_ASIC3 is not set
# CONFIG_MFD_TI_AM335X_TSCADC is not set
# CONFIG_HTC_EGPIO is not set
# CONFIG_HTC_PASIC3 is not set
# CONFIG_HTC_I2CPLD is not set
# CONFIG_UCB1400_CORE is not set
# CONFIG_MFD_LM3533 is not set
# CONFIG_TPS6105X is not set
# CONFIG_TPS65010 is not set
# CONFIG_TPS6507X is not set
# CONFIG_MFD_TPS65217 is not set
# CONFIG_MFD_TPS6586X is not set
# CONFIG_MFD_TPS65910 is not set
# CONFIG_MFD_TPS65912_I2C is not set
# CONFIG_MFD_TPS65912_SPI is not set
# CONFIG_MFD_TPS80031 is not set
# CONFIG_TWL4030_CORE is not set
# CONFIG_TWL6040_CORE is not set
# CONFIG_MFD_STMPE is not set
# CONFIG_MFD_TC3589X is not set
# CONFIG_MFD_TMIO is not set
# CONFIG_MFD_T7L66XB is not set
# CONFIG_MFD_SMSC is not set
# CONFIG_MFD_TC6387XB is not set
# CONFIG_MFD_TC6393XB is not set
# CONFIG_PMIC_DA903X is not set
# CONFIG_MFD_DA9052_SPI is not set
# CONFIG_MFD_DA9052_I2C is not set
# CONFIG_MFD_DA9055 is not set
# CONFIG_PMIC_ADP5520 is not set
# CONFIG_MFD_LP8788 is not set
# CONFIG_MFD_MAX77686 is not set
# CONFIG_MFD_MAX77693 is not set
# CONFIG_MFD_MAX8907 is not set
# CONFIG_MFD_MAX8925 is not set
# CONFIG_MFD_MAX8997 is not set
# CONFIG_MFD_MAX8998 is not set
# CONFIG_MFD_SEC_CORE is not set
# CONFIG_MFD_ARIZONA_I2C is not set
# CONFIG_MFD_ARIZONA_SPI is not set
# CONFIG_MFD_WM8400 is not set
# CONFIG_MFD_WM831X_I2C is not set
# CONFIG_MFD_WM831X_SPI is not set
# CONFIG_MFD_WM8350_I2C is not set
# CONFIG_MFD_WM8994 is not set
# CONFIG_MFD_PCF50633 is not set
CONFIG_MFD_MC13783=y
CONFIG_MFD_MC13XXX=y
CONFIG_MFD_MC13XXX_SPI=y
CONFIG_MFD_MC13XXX_I2C=y
# CONFIG_ABX500_CORE is not set
# CONFIG_EZX_PCAP is not set
CONFIG_MFD_WL1273_CORE=m
# CONFIG_MFD_TPS65090 is not set
# CONFIG_MFD_AAT2870_CORE is not set
# CONFIG_MFD_RC5T583 is not set
CONFIG_MFD_SYSCON=y
# CONFIG_MFD_PALMAS is not set
# CONFIG_MFD_VIPERBOARD is not set
# CONFIG_MFD_RETU is not set
# CONFIG_MFD_AS3711 is not set
CONFIG_VEXPRESS_CONFIG=y
CONFIG_REGULATOR=y
# CONFIG_REGULATOR_DEBUG is not set
# CONFIG_REGULATOR_DUMMY is not set
CONFIG_REGULATOR_FIXED_VOLTAGE=y
# CONFIG_REGULATOR_VIRTUAL_CONSUMER is not set
# CONFIG_REGULATOR_USERSPACE_CONSUMER is not set
# CONFIG_REGULATOR_GPIO is not set
# CONFIG_REGULATOR_AD5398 is not set
# CONFIG_REGULATOR_FAN53555 is not set
CONFIG_REGULATOR_ANATOP=y
CONFIG_REGULATOR_MC13XXX_CORE=y
CONFIG_REGULATOR_MC13783=y
CONFIG_REGULATOR_MC13892=y
# CONFIG_REGULATOR_ISL6271A is not set
# CONFIG_REGULATOR_MAX1586 is not set
# CONFIG_REGULATOR_MAX8649 is not set
# CONFIG_REGULATOR_MAX8660 is not set
# CONFIG_REGULATOR_MAX8952 is not set
# CONFIG_REGULATOR_MAX8973 is not set
# CONFIG_REGULATOR_LP3971 is not set
# CONFIG_REGULATOR_LP3972 is not set
# CONFIG_REGULATOR_LP872X is not set
# CONFIG_REGULATOR_TPS51632 is not set
# CONFIG_REGULATOR_TPS62360 is not set
# CONFIG_REGULATOR_TPS65023 is not set
# CONFIG_REGULATOR_TPS6507X is not set
# CONFIG_REGULATOR_TPS6524X is not set
# CONFIG_REGULATOR_VEXPRESS is not set
CONFIG_MEDIA_SUPPORT=y

#
# Multimedia core support
#
CONFIG_MEDIA_CAMERA_SUPPORT=y
CONFIG_MEDIA_ANALOG_TV_SUPPORT=y
CONFIG_MEDIA_DIGITAL_TV_SUPPORT=y
CONFIG_MEDIA_RADIO_SUPPORT=y
CONFIG_MEDIA_RC_SUPPORT=y
CONFIG_MEDIA_CONTROLLER=y
CONFIG_VIDEO_DEV=y
CONFIG_VIDEO_V4L2_SUBDEV_API=y
CONFIG_VIDEO_V4L2=y
# CONFIG_VIDEO_ADV_DEBUG is not set
# CONFIG_VIDEO_FIXED_MINOR_RANGES is not set
CONFIG_VIDEO_TUNER=m
CONFIG_V4L2_MEM2MEM_DEV=m
CONFIG_VIDEOBUF_GEN=m
CONFIG_VIDEOBUF_VMALLOC=m
CONFIG_VIDEOBUF_DMA_CONTIG=m
CONFIG_VIDEOBUF_DVB=m
CONFIG_VIDEOBUF2_CORE=m
CONFIG_VIDEOBUF2_MEMOPS=m
CONFIG_VIDEOBUF2_DMA_CONTIG=m
CONFIG_VIDEOBUF2_VMALLOC=m
CONFIG_DVB_CORE=y
CONFIG_DVB_NET=y
CONFIG_DVB_MAX_ADAPTERS=8
CONFIG_DVB_DYNAMIC_MINORS=y

#
# Media drivers
#
CONFIG_RC_CORE=y
CONFIG_RC_MAP=m
CONFIG_RC_DECODERS=y
CONFIG_LIRC=y
CONFIG_IR_LIRC_CODEC=m
CONFIG_IR_NEC_DECODER=m
CONFIG_IR_RC5_DECODER=m
CONFIG_IR_RC6_DECODER=m
CONFIG_IR_JVC_DECODER=m
CONFIG_IR_SONY_DECODER=m
CONFIG_IR_RC5_SZ_DECODER=m
CONFIG_IR_SANYO_DECODER=m
CONFIG_IR_MCE_KBD_DECODER=m
CONFIG_RC_DEVICES=y
CONFIG_RC_ATI_REMOTE=m
CONFIG_IR_IMON=m
CONFIG_IR_MCEUSB=m
CONFIG_IR_REDRAT3=m
CONFIG_IR_STREAMZAP=m
CONFIG_IR_IGUANA=m
CONFIG_IR_TTUSBIR=m
```

```
CONFIG_RC_LOOPBACK=m
CONFIG_IR_GPIO_CIR=m
CONFIG_MEDIA_USB_SUPPORT=y

#
# Webcam devices
#
CONFIG_USB_VIDEO_CLASS=m
CONFIG_USB_VIDEO_CLASS_INPUT_EVDEV=y
CONFIG_USB_GSPCA=m
CONFIG_USB_M5602=m
CONFIG_USB_STV06XX=m
CONFIG_USB_GL860=m
CONFIG_USB_GSPCA_BENQ=m
CONFIG_USB_GSPCA_CONEX=m
CONFIG_USB_GSPCA_CPIA1=m
CONFIG_USB_GSPCA_ETOMS=m
CONFIG_USB_GSPCA_FINEPIX=m
CONFIG_USB_GSPCA_JEILINJ=m
CONFIG_USB_GSPCA_JL2005BCD=m
CONFIG_USB_GSPCA_KINECT=m
CONFIG_USB_GSPCA_KONICA=m
CONFIG_USB_GSPCA_MARS=m
CONFIG_USB_GSPCA_MR97310A=m
CONFIG_USB_GSPCA_NW80X=m
CONFIG_USB_GSPCA_OV519=m
CONFIG_USB_GSPCA_OV534=m
CONFIG_USB_GSPCA_OV534_9=m
CONFIG_USB_GSPCA_PAC207=m
CONFIG_USB_GSPCA_PAC7302=m
CONFIG_USB_GSPCA_PAC7311=m
CONFIG_USB_GSPCA_SE401=m
CONFIG_USB_GSPCA_SN9C2028=m
CONFIG_USB_GSPCA_SN9C20X=m
CONFIG_USB_GSPCA_SONIXB=m
CONFIG_USB_GSPCA_SONIXJ=m
CONFIG_USB_GSPCA_SPCA500=m
CONFIG_USB_GSPCA_SPCA501=m
CONFIG_USB_GSPCA_SPCA505=m
CONFIG_USB_GSPCA_SPCA506=m
CONFIG_USB_GSPCA_SPCA508=m
CONFIG_USB_GSPCA_SPCA561=m
CONFIG_USB_GSPCA_SPCA1528=m
CONFIG_USB_GSPCA_SQ905=m
CONFIG_USB_GSPCA_SQ905C=m
CONFIG_USB_GSPCA_SQ930X=m
CONFIG_USB_GSPCA_STK014=m
CONFIG_USB_GSPCA_STV0680=m
CONFIG_USB_GSPCA_SUNPLUS=m
CONFIG_USB_GSPCA_T613=m
CONFIG_USB_GSPCA_TOPRO=m
CONFIG_USB_GSPCA_TV8532=m
CONFIG_USB_GSPCA_VC032X=m
CONFIG_USB_GSPCA_VICAM=m
CONFIG_USB_GSPCA_XIRLINK_CIT=m
CONFIG_USB_GSPCA_ZC3XX=m
CONFIG_USB_PWC=m
# CONFIG_USB_PWC_DEBUG is not set
CONFIG_USB_PWC_INPUT_EVDEV=y
CONFIG_VIDEO_CPIA2=m
CONFIG_USB_ZR364XX=m
CONFIG_USB_STKWEBCAM=m
```

```
CONFIG_USB_S2255=m
CONFIG_USB_SN9C102=m

#
# Analog TV USB devices
#
CONFIG_VIDEO_AU0828=m
CONFIG_VIDEO_PVRUSB2=m
CONFIG_VIDEO_PVRUSB2_SYSFS=y
CONFIG_VIDEO_PVRUSB2_DVB=y
# CONFIG_VIDEO_PVRUSB2_DEBUGIFC is not set
CONFIG_VIDEO_HDPVR=m
CONFIG_VIDEO_TLG2300=m
CONFIG_VIDEO_USBVISION=m
CONFIG_VIDEO_STK1160=m
CONFIG_VIDEO_STK1160_AC97=y

#
# Analog/digital TV USB devices
#
CONFIG_VIDEO_CX231XX=m
CONFIG_VIDEO_CX231XX_RC=y
CONFIG_VIDEO_CX231XX_ALSA=m
CONFIG_VIDEO_CX231XX_DVB=m
CONFIG_VIDEO_TM6000=m
CONFIG_VIDEO_TM6000_ALSA=m
CONFIG_VIDEO_TM6000_DVB=m

#
# Digital TV USB devices
#
CONFIG_DVB_USB=m
CONFIG_DVB_USB_DEBUG=y
CONFIG_DVB_USB_A800=m
CONFIG_DVB_USB_DIBUSB_MB=m
CONFIG_DVB_USB_DIBUSB_MB_FAULTY=y
CONFIG_DVB_USB_DIBUSB_MC=m
CONFIG_DVB_USB_DIB0700=m
CONFIG_DVB_USB_UMT_010=m
CONFIG_DVB_USB_CXUSB=m
CONFIG_DVB_USB_M920X=m
CONFIG_DVB_USB_DIGITV=m
CONFIG_DVB_USB_VP7045=m
CONFIG_DVB_USB_VP702X=m
CONFIG_DVB_USB_GP8PSK=m
CONFIG_DVB_USB_NOVA_T_USB2=m
CONFIG_DVB_USB_TTUSB2=m
CONFIG_DVB_USB_DTT200U=m
CONFIG_DVB_USB_OPERA1=m
CONFIG_DVB_USB_AF9005=m
CONFIG_DVB_USB_AF9005_REMOTE=m
# CONFIG_DVB_USB_PCTV452E is not set
CONFIG_DVB_USB_DW2102=m
CONFIG_DVB_USB_CINERGY_T2=m
CONFIG_DVB_USB_DTV5100=m
CONFIG_DVB_USB_FRIIO=m
CONFIG_DVB_USB_AZ6027=m
CONFIG_DVB_USB_TECHNISAT_USB2=m
CONFIG_DVB_USB_V2=m
CONFIG_DVB_USB_CYPRESS_FIRMWARE=m
CONFIG_DVB_USB_AF9015=m
CONFIG_DVB_USB_AF9035=m
```

```
CONFIG_DVB_USB_ANYSEE=m
CONFIG_DVB_USB_AU6610=m
CONFIG_DVB_USB_AZ6007=m
CONFIG_DVB_USB_CE6230=m
CONFIG_DVB_USB_EC168=m
CONFIG_DVB_USB_GL861=m
CONFIG_DVB_USB_IT913X=m
CONFIG_DVB_USB_LME2510=m
CONFIG_DVB_USB_MXL111SF=m
CONFIG_DVB_USB_RTL28XXU=m
CONFIG_SMS_USB_DRV=m
CONFIG_DVB_B2C2_FLEXCOP_USB=m
CONFIG_DVB_B2C2_FLEXCOP_USB_DEBUG=y

#
# Webcam, TV (analog/digital) USB devices
#
CONFIG_VIDEO_EM28XX=m
CONFIG_VIDEO_EM28XX_ALSA=m
CONFIG_VIDEO_EM28XX_DVB=m
CONFIG_VIDEO_EM28XX_RC=m
CONFIG_V4L_PLATFORM_DRIVERS=y
CONFIG_VIDEO_TIMBERDALE=m
CONFIG_SOC_CAMERA=m
CONFIG_SOC_CAMERA_PLATFORM=m
# CONFIG_VIDEO_MX3 is not set
CONFIG_VIDEO_SH_MOBILE_CSI2=m
CONFIG_VIDEO_SH_MOBILE_CEU=m
CONFIG_V4L_MEM2MEM_DRIVERS=y
# CONFIG_VIDEO_CODA is not set
CONFIG_VIDEO_MEM2MEM_DEINTERLACE=m
CONFIG_V4L_TEST_DRIVERS=y
CONFIG_VIDEO_VIVI=m
CONFIG_VIDEO_MEM2MEM_TESTDEV=m

#
# Supported MMC/SDIO adapters
#
CONFIG_SMS_SDIO_DRV=y
CONFIG_RADIO_ADAPTERS=y
CONFIG_RADIO_SI470X=y
CONFIG_USB_SI470X=m
CONFIG_I2C_SI470X=m
CONFIG_USB_MR800=m
CONFIG_USB_DSBR=m
# CONFIG_RADIO_SHARK is not set
CONFIG_RADIO_SHARK2=m
CONFIG_I2C_SI4713=m
CONFIG_RADIO_SI4713=m
CONFIG_USB_KEENE=m
CONFIG_RADIO_TEA5764=m
CONFIG_RADIO_SAA7706H=m
CONFIG_RADIO_TEF6862=m
CONFIG_RADIO_WL1273=m

#
# Texas Instruments WL128x FM driver (ST based)
#
CONFIG_RADIO_WL128X=m
CONFIG_MEDIA_COMMON_OPTIONS=y

#
```

```
# common driver options
#
CONFIG_DVB_B2C2_FLEXCOP=m
CONFIG_DVB_B2C2_FLEXCOP_DEBUG=y
CONFIG_SMS_SIANO_MDTV=y
CONFIG_SMS_SIANO_RC=y
# CONFIG_MEDIA_SUBDRV_AUTOSELECT is not set

#
# Media ancillary drivers (tuners, sensors, i2c, frontends)
#
CONFIG_VIDEO_TVEEPROM=m
CONFIG_VIDEO_IR_I2C=y

#
# Encoders, decoders, sensors and other helper chips
#

#
# Audio decoders, processors and mixers
#
CONFIG_VIDEO_TVAUDIO=m
CONFIG_VIDEO_TDA7432=m
CONFIG_VIDEO_TDA9840=m
CONFIG_VIDEO_TEA6415C=m
CONFIG_VIDEO_TEA6420=m
CONFIG_VIDEO_MSP3400=m
CONFIG_VIDEO_CS5345=m
CONFIG_VIDEO_CS53L32A=m
CONFIG_VIDEO_TLV320AIC23B=m
CONFIG_VIDEO_WM8775=m
CONFIG_VIDEO_WM8739=m
CONFIG_VIDEO_VP27SMPX=m

#
# RDS decoders
#
CONFIG_VIDEO_SAA6588=m

#
# Video decoders
#
CONFIG_VIDEO_ADV7180=m
CONFIG_VIDEO_ADV7183=m
CONFIG_VIDEO_ADV7604=m
CONFIG_VIDEO_BT819=m
CONFIG_VIDEO_BT856=m
CONFIG_VIDEO_BT866=m
CONFIG_VIDEO_KS0127=m
CONFIG_VIDEO_SAA7110=m
CONFIG_VIDEO_SAA711X=m
CONFIG_VIDEO_SAA7191=m
CONFIG_VIDEO_TVP514X=m
CONFIG_VIDEO_TVP5150=m
CONFIG_VIDEO_TVP7002=m
CONFIG_VIDEO_VPX3220=m

#
# Video and audio decoders
#
CONFIG_VIDEO_SAA717X=m
CONFIG_VIDEO_CX25840=m
```

```
#
# MPEG video encoders
#
CONFIG_VIDEO_CX2341X=m

#
# Video encoders
#
CONFIG_VIDEO_SAA7127=m
CONFIG_VIDEO_SAA7185=m
CONFIG_VIDEO_ADV7170=m
CONFIG_VIDEO_ADV7175=m
CONFIG_VIDEO_ADV7343=m
CONFIG_VIDEO_ADV7393=m
CONFIG_VIDEO_AD9389B=m
CONFIG_VIDEO_AK881X=m

#
# Camera sensor devices
#
CONFIG_VIDEO_APTINA_PLL=y
CONFIG_VIDEO_SMIAPP_PLL=m
CONFIG_VIDEO_OV7670=m
CONFIG_VIDEO_VS6624=m
CONFIG_VIDEO_MT9M032=m
CONFIG_VIDEO_MT9P031=y
CONFIG_VIDEO_MT9T001=m
CONFIG_VIDEO_MT9V011=m
CONFIG_VIDEO_MT9V032=m
CONFIG_VIDEO_TCM825X=m
CONFIG_VIDEO_SR030PC30=m
CONFIG_VIDEO_NOON010PC30=m
CONFIG_VIDEO_M5MOLS=m
CONFIG_VIDEO_S5K6AA=m
CONFIG_VIDEO_S5K4ECGX=m
CONFIG_VIDEO_SMIAPP=m

#
# Flash devices
#
CONFIG_VIDEO_ADP1653=m
CONFIG_VIDEO_AS3645A=m

#
# Video improvement chips
#
CONFIG_VIDEO_UPD64031A=m
CONFIG_VIDEO_UPD64083=m

#
# Miscelaneous helper chips
#
CONFIG_VIDEO_THS7303=m
CONFIG_VIDEO_M52790=m

#
# Sensors used on soc_camera driver
#

#
# soc_camera sensor drivers
```

```
#
CONFIG_SOC_CAMERA_IMX074=m
CONFIG_SOC_CAMERA_MT9M001=m
CONFIG_SOC_CAMERA_MT9M111=m
CONFIG_SOC_CAMERA_MT9T031=m
CONFIG_SOC_CAMERA_MT9T112=m
CONFIG_SOC_CAMERA_MT9V022=m
CONFIG_SOC_CAMERA_OV2640=m
CONFIG_SOC_CAMERA_OV5642=m
CONFIG_SOC_CAMERA_OV6650=m
CONFIG_SOC_CAMERA_OV772X=m
CONFIG_SOC_CAMERA_OV9640=m
CONFIG_SOC_CAMERA_OV9740=m
CONFIG_SOC_CAMERA_RJ54N1=m
CONFIG_SOC_CAMERA_TW9910=m
CONFIG_MEDIA_ATTACH=y
CONFIG_MEDIA_TUNER=y

#
# Customize TV tuners
#
CONFIG_MEDIA_TUNER_SIMPLE=m
CONFIG_MEDIA_TUNER_TDA8290=m
CONFIG_MEDIA_TUNER_TDA827X=m
CONFIG_MEDIA_TUNER_TDA18271=m
CONFIG_MEDIA_TUNER_TDA9887=m
CONFIG_MEDIA_TUNER_TEA5761=m
CONFIG_MEDIA_TUNER_TEA5767=m
CONFIG_MEDIA_TUNER_MT20XX=m
CONFIG_MEDIA_TUNER_MT2060=m
CONFIG_MEDIA_TUNER_MT2063=m
CONFIG_MEDIA_TUNER_MT2266=m
CONFIG_MEDIA_TUNER_MT2131=m
CONFIG_MEDIA_TUNER_QT1010=m
CONFIG_MEDIA_TUNER_XC2028=m
CONFIG_MEDIA_TUNER_XC5000=m
CONFIG_MEDIA_TUNER_XC4000=m
CONFIG_MEDIA_TUNER_MXL5005S=m
CONFIG_MEDIA_TUNER_MXL5007T=m
CONFIG_MEDIA_TUNER_MC44S803=m
CONFIG_MEDIA_TUNER_MAX2165=m
CONFIG_MEDIA_TUNER_TDA18218=m
CONFIG_MEDIA_TUNER_FC0011=m
CONFIG_MEDIA_TUNER_FC0012=m
CONFIG_MEDIA_TUNER_FC0013=m
CONFIG_MEDIA_TUNER_TDA18212=m
CONFIG_MEDIA_TUNER_E4000=m
CONFIG_MEDIA_TUNER_FC2580=m
CONFIG_MEDIA_TUNER_TUA9001=m

#
# Customise DVB Frontends
#

#
# Multistandard (satellite) frontends
#
CONFIG_DVB_STB0899=m
CONFIG_DVB_STB6100=m
CONFIG_DVB_STV090x=m
CONFIG_DVB_STV6110x=m
```

```
#
# Multistandard (cable + terrestrial) frontends
#
CONFIG_DVB_DRXK=m
CONFIG_DVB_TDA18271C2DD=m

#
# DVB-S (satellite) frontends
#
CONFIG_DVB_CX24110=m
CONFIG_DVB_CX24123=m
CONFIG_DVB_MT312=m
CONFIG_DVB_ZL10036=m
CONFIG_DVB_ZL10039=m
CONFIG_DVB_S5H1420=m
CONFIG_DVB_STV0288=m
CONFIG_DVB_STB6000=m
CONFIG_DVB_STV0299=m
CONFIG_DVB_STV6110=m
CONFIG_DVB_STV0900=m
CONFIG_DVB_TDA8083=m
CONFIG_DVB_TDA10086=m
CONFIG_DVB_TDA8261=m
CONFIG_DVB_VES1X93=m
CONFIG_DVB_TUNER_ITD1000=m
CONFIG_DVB_TUNER_CX24113=m
CONFIG_DVB_TDA826X=m
CONFIG_DVB_TUA6100=m
CONFIG_DVB_CX24116=m
CONFIG_DVB_SI21XX=m
CONFIG_DVB_DS3000=m
CONFIG_DVB_MB86A16=m
CONFIG_DVB_TDA10071=m

#
# DVB-T (terrestrial) frontends
#
CONFIG_DVB_SP8870=m
CONFIG_DVB_SP887X=m
CONFIG_DVB_CX22700=m
CONFIG_DVB_CX22702=m
CONFIG_DVB_S5H1432=m
CONFIG_DVB_DRXD=m
CONFIG_DVB_L64781=m
CONFIG_DVB_TDA1004X=m
CONFIG_DVB_NXT6000=m
CONFIG_DVB_MT352=m
CONFIG_DVB_ZL10353=m
CONFIG_DVB_DIB3000MB=m
CONFIG_DVB_DIB3000MC=m
CONFIG_DVB_DIB7000M=m
CONFIG_DVB_DIB7000P=m
CONFIG_DVB_DIB9000=m
CONFIG_DVB_TDA10048=m
CONFIG_DVB_AF9013=m
CONFIG_DVB_EC100=m
CONFIG_DVB_HD29L2=m
CONFIG_DVB_STV0367=m
CONFIG_DVB_CXD2820R=m
CONFIG_DVB_RTL2830=m
CONFIG_DVB_RTL2832=m
```

```
#
# DVB-C (cable) frontends
#
CONFIG_DVB_VES1820=m
CONFIG_DVB_TDA10021=m
CONFIG_DVB_TDA10023=m
CONFIG_DVB_STV0297=m

#
# ATSC (North American/Korean Terrestrial/Cable DTV) frontends
#
CONFIG_DVB_NXT200X=m
CONFIG_DVB_OR51211=m
CONFIG_DVB_OR51132=m
CONFIG_DVB_BCM3510=m
CONFIG_DVB_LGDT330X=m
CONFIG_DVB_LGDT3305=m
CONFIG_DVB_LG2160=m
CONFIG_DVB_S5H1409=m
CONFIG_DVB_AU8522=m
CONFIG_DVB_AU8522_DTV=m
CONFIG_DVB_AU8522_V4L=m
CONFIG_DVB_S5H1411=m

#
# ISDB-T (terrestrial) frontends
#
CONFIG_DVB_S921=m
CONFIG_DVB_DIB8000=m
CONFIG_DVB_MB86A20S=m

#
# Digital terrestrial only tuners/PLL
#
CONFIG_DVB_PLL=m
CONFIG_DVB_TUNER_DIB0070=m
CONFIG_DVB_TUNER_DIB0090=m

#
# SEC control devices for DVB-S
#
CONFIG_DVB_LNBP21=m
CONFIG_DVB_LNBP22=m
CONFIG_DVB_ISL6405=m
CONFIG_DVB_ISL6421=m
CONFIG_DVB_ISL6423=m
CONFIG_DVB_A8293=m
CONFIG_DVB_LGS8GL5=m
CONFIG_DVB_LGS8GXX=m
CONFIG_DVB_ATBM8830=m
CONFIG_DVB_TDA665x=m
CONFIG_DVB_IX2505V=m
CONFIG_DVB_IT913X_FE=m
CONFIG_DVB_M88RS2000=m
CONFIG_DVB_AF9033=m

#
# Tools to develop new frontends
#
# CONFIG_DVB_DUMMY_FE is not set

#
```

```
# Graphics support
#
CONFIG_DRM=y
# CONFIG_DRM_EXYNOS is not set
# CONFIG_DRM_UDL is not set
# CONFIG_VGASTATE is not set
# CONFIG_VIDEO_OUTPUT_CONTROL is not set
CONFIG_FB=y
CONFIG_FIRMWARE_EDID=y
# CONFIG_FB_DDC is not set
# CONFIG_FB_BOOT_VESA_SUPPORT is not set
CONFIG_FB_CFB_FILLRECT=y
CONFIG_FB_CFB_COPYAREA=y
CONFIG_FB_CFB_IMAGEBLIT=y
# CONFIG_FB_CFB_REV_PIXELS_IN_BYTE is not set
CONFIG_FB_SYS_FILLRECT=m
CONFIG_FB_SYS_COPYAREA=m
CONFIG_FB_SYS_IMAGEBLIT=m
# CONFIG_FB_FOREIGN_ENDIAN is not set
CONFIG_FB_SYS_FOPS=m
# CONFIG_FB_WMT_GE_ROPS is not set
CONFIG_FB_DEFERRED_IO=y
# CONFIG_FB_SVGALIB is not set
# CONFIG_FB_MACMODES is not set
# CONFIG_FB_BACKLIGHT is not set
# CONFIG_FB_MODE_HELPERS is not set
# CONFIG_FB_TILEBLITTING is not set

#
# Frame buffer hardware drivers
#
# CONFIG_FB_ARMCLCD is not set
# CONFIG_FB_UVESA is not set
# CONFIG_FB_S1D13XXX is not set
# CONFIG_FB_TMIO is not set
# CONFIG_FB_SMSCUFX is not set
# CONFIG_FB_UDL is not set
# CONFIG_FB_VIRTUAL is not set
# CONFIG_FB_METRONOME is not set
CONFIG_FB_MX3=y
# CONFIG_FB_BROADSHEET is not set
# CONFIG_FB_AUO_K190X is not set
# CONFIG_EXYNOS_VIDEO is not set
CONFIG_BACKLIGHT_LCD_SUPPORT=y
CONFIG_LCD_CLASS_DEVICE=y
CONFIG_LCD_L4F00242T03=y
# CONFIG_LCD_LMS283GF05 is not set
# CONFIG_LCD_LTV350QV is not set
# CONFIG_LCD_ILI9320 is not set
# CONFIG_LCD_TDO24M is not set
# CONFIG_LCD_VGG2432A4 is not set
CONFIG_LCD_PLATFORM=y
# CONFIG_LCD_S6E63M0 is not set
# CONFIG_LCD_LD9040 is not set
# CONFIG_LCD_AMS369FG06 is not set
CONFIG_BACKLIGHT_CLASS_DEVICE=y
CONFIG_BACKLIGHT_GENERIC=y
# CONFIG_BACKLIGHT_PWM is not set
# CONFIG_BACKLIGHT_ADP8860 is not set
# CONFIG_BACKLIGHT_ADP8870 is not set
# CONFIG_BACKLIGHT_LM3630 is not set
# CONFIG_BACKLIGHT_LM3639 is not set
# CONFIG_BACKLIGHT_LP855X is not set

#
# Console display driver support
#
CONFIG_DUMMY_CONSOLE=y
CONFIG_FRAMEBUFFER_CONSOLE=y
CONFIG_FRAMEBUFFER_CONSOLE_DETECT_PRIMARY=y
CONFIG_FRAMEBUFFER_CONSOLE_ROTATION=y
CONFIG_FONTS=y
CONFIG_FONT_8x8=y
CONFIG_FONT_8x16=y
# CONFIG_FONT_6x11 is not set
# CONFIG_FONT_7x14 is not set
# CONFIG_FONT_PEARL_8x8 is not set
# CONFIG_FONT_ACORN_8x8 is not set
# CONFIG_FONT_MINI_4x6 is not set
# CONFIG_FONT_SUN8x16 is not set
# CONFIG_FONT_SUN12x22 is not set
# CONFIG_FONT_10x18 is not set
CONFIG_LOGO=y
CONFIG_LOGO_LINUX_MONO=y
CONFIG_LOGO_LINUX_VGA16=y
CONFIG_LOGO_LINUX_CLUT224=y
# CONFIG_FB_SSD1307 is not set
CONFIG_SOUND=y
CONFIG_SOUND_OSS_CORE=y
CONFIG_SOUND_OSS_CORE_PRECLAIM=y
CONFIG_SND=y
CONFIG_SND_TIMER=y
CONFIG_SND_PCM=y
CONFIG_SND_HWDEP=m
CONFIG_SND_RAWMIDI=m
CONFIG_SND_COMPRESS_OFFLOAD=y
CONFIG_SND_JACK=y
# CONFIG_SND_SEQUENCER is not set
CONFIG_SND_OSSEMUL=y
CONFIG_SND_MIXER_OSS=m
CONFIG_SND_PCM_OSS=m
CONFIG_SND_PCM_OSS_PLUGINS=y
# CONFIG_SND_HRTIMER is not set
# CONFIG_SND_DYNAMIC_MINORS is not set
CONFIG_SND_SUPPORT_OLD_API=y
CONFIG_SND_VERBOSE_PROCFS=y
CONFIG_SND_VERBOSE_PRINTK=y
CONFIG_SND_DEBUG=y
# CONFIG_SND_DEBUG_VERBOSE is not set
# CONFIG_SND_PCM_XRUN_DEBUG is not set
CONFIG_SND_VMASTER=y
# CONFIG_SND_RAWMIDI_SEQ is not set
# CONFIG_SND_OPL3_LIB_SEQ is not set
# CONFIG_SND_OPL4_LIB_SEQ is not set
# CONFIG_SND_SBAWE_SEQ is not set
# CONFIG_SND_EMU10K1_SEQ is not set
CONFIG_SND_AC97_CODEC=m
CONFIG_SND_DRIVERS=y
# CONFIG_SND_DUMMY is not set
# CONFIG_SND_ALOOP is not set
# CONFIG_SND_MTPAV is not set
# CONFIG_SND_SERIAL_U16550 is not set
# CONFIG_SND_MPU401 is not set
# CONFIG_SND_AC97_POWER_SAVE is not set
```

```
CONFIG_SND_ARM=y
# CONFIG_SND_ARMAACI is not set
CONFIG_SND_SPI=y
CONFIG_SND_USB=y
CONFIG_SND_USB_AUDIO=m
# CONFIG_SND_USB_UA101 is not set
# CONFIG_SND_USB_CAIAQ is not set
# CONFIG_SND_USB_6FIRE is not set
CONFIG_SND_SOC=y
CONFIG_SND_SOC_DMAENGINE_PCM=y
# CONFIG_SND_DESIGNWARE_I2S is not set
CONFIG_SND_SOC_FSL_SSI=y
CONFIG_SND_SOC_FSL_UTILS=y
CONFIG_SND_IMX_SOC=y
CONFIG_SND_SOC_IMX_SSI=y
CONFIG_SND_SOC_IMX_PCM=y
CONFIG_SND_SOC_IMX_PCM_FIQ=y
CONFIG_SND_SOC_IMX_PCM_DMA=y
CONFIG_SND_SOC_IMX_AUDMUX=y
CONFIG_SND_SOC_EUKREA_TLV320=y
CONFIG_SND_SOC_IMX_SGTL5000=y
CONFIG_SND_SOC_IMX_MC13783=y
CONFIG_SND_SOC_I2C_AND_SPI=y
CONFIG_SND_SOC_ALL_CODECS=y
CONFIG_SND_SOC_WM_HUBS=y
CONFIG_SND_SOC_WM_ADSP=y
CONFIG_SND_SOC_AD1836=y
CONFIG_SND_SOC_AD193X=y
CONFIG_SND_SOC_AD73311=y
CONFIG_SND_SOC_ADAU1373=y
CONFIG_SND_SOC_ADAV80X=y
CONFIG_SND_SOC_ADS117X=y
CONFIG_SND_SOC_AK4104=y
CONFIG_SND_SOC_AK4535=y
CONFIG_SND_SOC_AK4641=y
CONFIG_SND_SOC_AK4642=y
CONFIG_SND_SOC_AK4671=y
CONFIG_SND_SOC_ALC5623=y
CONFIG_SND_SOC_ALC5632=y
CONFIG_SND_SOC_CS42L51=y
CONFIG_SND_SOC_CS42L52=y
CONFIG_SND_SOC_CS42L73=y
CONFIG_SND_SOC_CS4270=y
CONFIG_SND_SOC_CS4271=y
CONFIG_SND_SOC_CX20442=y
CONFIG_SND_SOC_JZ4740_CODEC=y
CONFIG_SND_SOC_L3=y
CONFIG_SND_SOC_DA7210=y
CONFIG_SND_SOC_DA732X=y
CONFIG_SND_SOC_DA9055=y
CONFIG_SND_SOC_DFBMCS320=y
CONFIG_SND_SOC_ISABELLE=y
CONFIG_SND_SOC_LM49453=y
CONFIG_SND_SOC_MAX98088=y
CONFIG_SND_SOC_MAX98090=y
CONFIG_SND_SOC_MAX98095=y
CONFIG_SND_SOC_MAX9850=y
CONFIG_SND_SOC_PCM3008=y
CONFIG_SND_SOC_RT5631=y
CONFIG_SND_SOC_SGTL5000=y
CONFIG_SND_SOC_SPDIF=y
CONFIG_SND_SOC_SSM2602=y
```

```
CONFIG_SND_SOC_STA32X=y
CONFIG_SND_SOC_STA529=y
CONFIG_SND_SOC_TLV320AIC23=y
CONFIG_SND_SOC_TLV320AIC26=y
CONFIG_SND_SOC_TLV320AIC32X4=y
CONFIG_SND_SOC_TLV320AIC3X=y
CONFIG_SND_SOC_TLV320DAC33=y
CONFIG_SND_SOC_UDA134X=y
CONFIG_SND_SOC_UDA1380=y
CONFIG_SND_SOC_WL1273=m
CONFIG_SND_SOC_WM0010=y
CONFIG_SND_SOC_WM1250_EV1=y
CONFIG_SND_SOC_WM2000=y
CONFIG_SND_SOC_WM2200=y
CONFIG_SND_SOC_WM5100=y
CONFIG_SND_SOC_WM8510=y
CONFIG_SND_SOC_WM8523=y
CONFIG_SND_SOC_WM8580=y
CONFIG_SND_SOC_WM8711=y
CONFIG_SND_SOC_WM8727=y
CONFIG_SND_SOC_WM8728=y
CONFIG_SND_SOC_WM8731=y
CONFIG_SND_SOC_WM8737=y
CONFIG_SND_SOC_WM8741=y
CONFIG_SND_SOC_WM8750=y
CONFIG_SND_SOC_WM8753=y
CONFIG_SND_SOC_WM8770=y
CONFIG_SND_SOC_WM8776=y
CONFIG_SND_SOC_WM8782=y
CONFIG_SND_SOC_WM8804=y
CONFIG_SND_SOC_WM8900=y
CONFIG_SND_SOC_WM8903=y
CONFIG_SND_SOC_WM8904=y
CONFIG_SND_SOC_WM8940=y
CONFIG_SND_SOC_WM8955=y
CONFIG_SND_SOC_WM8960=y
CONFIG_SND_SOC_WM8961=y
CONFIG_SND_SOC_WM8962=y
CONFIG_SND_SOC_WM8971=y
CONFIG_SND_SOC_WM8974=y
CONFIG_SND_SOC_WM8978=y
CONFIG_SND_SOC_WM8983=y
CONFIG_SND_SOC_WM8985=y
CONFIG_SND_SOC_WM8988=y
CONFIG_SND_SOC_WM8990=y
CONFIG_SND_SOC_WM8991=y
CONFIG_SND_SOC_WM8993=y
CONFIG_SND_SOC_WM8995=y
CONFIG_SND_SOC_WM8996=y
CONFIG_SND_SOC_WM9081=y
CONFIG_SND_SOC_WM9090=y
CONFIG_SND_SOC_LM4857=y
CONFIG_SND_SOC_MAX9768=y
CONFIG_SND_SOC_MAX9877=y
CONFIG_SND_SOC_MC13783=y
CONFIG_SND_SOC_ML26124=y
CONFIG_SND_SOC_TPA6130A2=y
# CONFIG_SND_SIMPLE_CARD is not set
# CONFIG_SOUND_PRIME is not set
CONFIG_AC97_BUS=m

#
```

```
# HID support
#
CONFIG_HID=y
# CONFIG_HID_BATTERY_STRENGTH is not set
CONFIG_HIDRAW=y
CONFIG_UHID=y
CONFIG_HID_GENERIC=y

#
# Special HID drivers
#
CONFIG_HID_A4TECH=m
CONFIG_HID_ACRUX=m
CONFIG_HID_ACRUX_FF=y
CONFIG_HID_APPLE=m
CONFIG_HID_AUREAL=m
CONFIG_HID_BELKIN=m
CONFIG_HID_CHERRY=m
CONFIG_HID_CHICONY=m
CONFIG_HID_PRODIKEYS=m
CONFIG_HID_CYPRESS=m
CONFIG_HID_DRAGONRISE=m
CONFIG_DRAGONRISE_FF=y
CONFIG_HID_EMS_FF=m
CONFIG_HID_ELECOM=m
CONFIG_HID_EZKEY=m
CONFIG_HID_HOLTEK=m
CONFIG_HOLTEK_FF=y
CONFIG_HID_KEYTOUCH=m
CONFIG_HID_KYE=m
CONFIG_HID_UCLOGIC=m
CONFIG_HID_WALTOP=m
CONFIG_HID_GYRATION=m
# CONFIG_HID_ICADE is not set
CONFIG_HID_TWINHAN=m
CONFIG_HID_KENSINGTON=m
CONFIG_HID_LCPOWER=m
CONFIG_HID_LENOVO_TPKBD=m
CONFIG_HID_LOGITECH=m
CONFIG_HID_LOGITECH_DJ=m
CONFIG_LOGITECH_FF=y
CONFIG_LOGIRUMBLEPAD2_FF=y
CONFIG_LOGIG940_FF=y
CONFIG_LOGIWHEELS_FF=y
CONFIG_HID_MAGICMOUSE=m
CONFIG_HID_MICROSOFT=m
CONFIG_HID_MONTEREY=m
CONFIG_HID_MULTITOUCH=m
CONFIG_HID_NTRIG=m
CONFIG_HID_ORTEK=m
CONFIG_HID_PANTHERLORD=m
CONFIG_PANTHERLORD_FF=y
CONFIG_HID_PETALYNX=m
CONFIG_HID_PICOLCD=m
CONFIG_HID_PICOLCD_FB=y
CONFIG_HID_PICOLCD_BACKLIGHT=y
CONFIG_HID_PICOLCD_LCD=y
CONFIG_HID_PICOLCD_LEDS=y
CONFIG_HID_PICOLCD_CIR=y
CONFIG_HID_PRIMAX=m
CONFIG_HID_PS3REMOTE=m
CONFIG_HID_ROCCAT=m
```

```
CONFIG_HID_SAITEK=m
CONFIG_HID_SAMSUNG=m
CONFIG_HID_SONY=m
CONFIG_HID_SPEEDLINK=m
CONFIG_HID_SUNPLUS=m
CONFIG_HID_GREENASIA=m
CONFIG_GREENASIA_FF=y
CONFIG_HID_SMARTJOYPLUS=m
CONFIG_SMARTJOYPLUS_FF=y
CONFIG_HID_TIVO=m
CONFIG_HID_TOPSEED=m
CONFIG_HID_THRUSTMASTER=m
CONFIG_THRUSTMASTER_FF=y
CONFIG_HID_WACOM=m
CONFIG_HID_WIIMOTE=m
CONFIG_HID_WIIMOTE_EXT=y
CONFIG_HID_ZEROPLUS=m
CONFIG_ZEROPLUS_FF=y
CONFIG_HID_ZYDACRON=m
CONFIG_HID_SENSOR_HUB=m

#
# USB HID support
#
CONFIG_USB_HID=y
CONFIG_HID_PID=y
CONFIG_USB_HIDDEV=y

#
# I2C HID support
#
# CONFIG_I2C_HID is not set
# CONFIG_USB_ARCH_HAS_OHCI is not set
CONFIG_USB_ARCH_HAS_EHCI=y
# CONFIG_USB_ARCH_HAS_XHCI is not set
CONFIG_USB_SUPPORT=y
CONFIG_USB_COMMON=y
CONFIG_USB_ARCH_HAS_HCD=y
CONFIG_USB=y
# CONFIG_USB_DEBUG is not set
CONFIG_USB_ANNOUNCE_NEW_DEVICES=y

#
# Miscellaneous USB options
#
# CONFIG_USB_DYNAMIC_MINORS is not set
# CONFIG_USB_SUSPEND is not set
# CONFIG_USB_OTG_WHITELIST is not set
# CONFIG_USB_OTG_BLACKLIST_HUB is not set
# CONFIG_USB_DWC3 is not set
CONFIG_USB_MON=y
# CONFIG_USB_WUSB_CBAF is not set

#
# USB Host Controller Drivers
#
# CONFIG_USB_C67X00_HCD is not set
CONFIG_USB_EHCI_HCD=y
CONFIG_USB_EHCI_ROOT_HUB_TT=y
CONFIG_USB_EHCI_TT_NEWSCHED=y
CONFIG_USB_EHCI_MXC=y
# CONFIG_USB_OXU210HP_HCD is not set
# CONFIG_USB_ISP116X_HCD is not set
# CONFIG_USB_ISP1760_HCD is not set
# CONFIG_USB_ISP1362_HCD is not set
# CONFIG_USB_EHCI_HCD_PLATFORM is not set
# CONFIG_USB_U132_HCD is not set
# CONFIG_USB_SL811_HCD is not set
# CONFIG_USB_R8A66597_HCD is not set
# CONFIG_USB_IMX21_HCD is not set
# CONFIG_USB_HCD_SSB is not set
# CONFIG_USB_MUSB_HDRC is not set
CONFIG_USB_CHIPIDEA=y
# CONFIG_USB_CHIPIDEA_UDC is not set
CONFIG_USB_CHIPIDEA_HOST=y
# CONFIG_USB_CHIPIDEA_DEBUG is not set
# CONFIG_USB_RENESAS_USBHS is not set

#
# USB Device Class drivers
#
CONFIG_USB_ACM=m
CONFIG_USB_PRINTER=m
CONFIG_USB_WDM=m
CONFIG_USB_TMC=m

#
# NOTE: USB_STORAGE depends on SCSI but BLK_DEV_SD may
#

#
# also be needed; see USB_STORAGE Help for more info
#
CONFIG_USB_STORAGE=y
# CONFIG_USB_STORAGE_DEBUG is not set
CONFIG_USB_STORAGE_REALTEK=m
CONFIG_REALTEK_AUTOPM=y
CONFIG_USB_STORAGE_DATAFAB=m
CONFIG_USB_STORAGE_FREECOM=m
CONFIG_USB_STORAGE_ISD200=m
CONFIG_USB_STORAGE_USBAT=m
CONFIG_USB_STORAGE_SDDR09=m
CONFIG_USB_STORAGE_SDDR55=m
CONFIG_USB_STORAGE_JUMPSHOT=m
CONFIG_USB_STORAGE_ALAUDA=m
CONFIG_USB_STORAGE_ONETOUCH=m
CONFIG_USB_STORAGE_KARMA=m
CONFIG_USB_STORAGE_CYPRESS_ATACB=m
CONFIG_USB_STORAGE_ENE_UB6250=m

#
# USB Imaging devices
#
CONFIG_USB_MDC800=m
CONFIG_USB_MICROTEK=m

#
# USB port drivers
#
CONFIG_USB_SERIAL=m
CONFIG_USB_SERIAL_GENERIC=y
CONFIG_USB_SERIAL_AIRCABLE=m
CONFIG_USB_SERIAL_ARK3116=m
CONFIG_USB_SERIAL_BELKIN=m
```

```
CONFIG_USB_SERIAL_CH341=m
CONFIG_USB_SERIAL_WHITEHEAT=m
CONFIG_USB_SERIAL_DIGI_ACCELEPORT=m
CONFIG_USB_SERIAL_CP210X=m
CONFIG_USB_SERIAL_CYPRESS_M8=m
CONFIG_USB_SERIAL_EMPEG=m
CONFIG_USB_SERIAL_FTDI_SIO=m
CONFIG_USB_SERIAL_FUNSOFT=m
CONFIG_USB_SERIAL_VISOR=m
CONFIG_USB_SERIAL_IPAQ=m
CONFIG_USB_SERIAL_IR=m
CONFIG_USB_SERIAL_EDGEPORT=m
CONFIG_USB_SERIAL_EDGEPORT_TI=m
CONFIG_USB_SERIAL_F81232=m
CONFIG_USB_SERIAL_GARMIN=m
CONFIG_USB_SERIAL_IPW=m
CONFIG_USB_SERIAL_IUU=m
CONFIG_USB_SERIAL_KEYSPAN_PDA=m
CONFIG_USB_SERIAL_KEYSPAN=m
CONFIG_USB_SERIAL_KEYSPAN_MPR=y
CONFIG_USB_SERIAL_KEYSPAN_USA28=y
CONFIG_USB_SERIAL_KEYSPAN_USA28X=y
CONFIG_USB_SERIAL_KEYSPAN_USA28XA=y
CONFIG_USB_SERIAL_KEYSPAN_USA28XB=y
CONFIG_USB_SERIAL_KEYSPAN_USA19=y
CONFIG_USB_SERIAL_KEYSPAN_USA18X=y
CONFIG_USB_SERIAL_KEYSPAN_USA19W=y
CONFIG_USB_SERIAL_KEYSPAN_USA19QW=y
CONFIG_USB_SERIAL_KEYSPAN_USA19QI=y
CONFIG_USB_SERIAL_KEYSPAN_USA49W=y
CONFIG_USB_SERIAL_KEYSPAN_USA49WLC=y
CONFIG_USB_SERIAL_KLSI=m
CONFIG_USB_SERIAL_KOBIL_SCT=m
CONFIG_USB_SERIAL_MCT_U232=m
CONFIG_USB_SERIAL_METRO=m
CONFIG_USB_SERIAL_MOS7720=m
CONFIG_USB_SERIAL_MOS7840=m
CONFIG_USB_SERIAL_MOTOROLA=m
CONFIG_USB_SERIAL_NAVMAN=m
CONFIG_USB_SERIAL_PL2303=m
CONFIG_USB_SERIAL_OTI6858=m
CONFIG_USB_SERIAL_QCAUX=m
CONFIG_USB_SERIAL_QUALCOMM=m
CONFIG_USB_SERIAL_SPCP8X5=m
CONFIG_USB_SERIAL_HP4X=m
CONFIG_USB_SERIAL_SAFE=m
CONFIG_USB_SERIAL_SAFE_PADDED=y
CONFIG_USB_SERIAL_SIEMENS_MPI=m
CONFIG_USB_SERIAL_SIERRAWIRELESS=m
CONFIG_USB_SERIAL_SYMBOL=m
CONFIG_USB_SERIAL_TI=m
CONFIG_USB_SERIAL_CYBERJACK=m
CONFIG_USB_SERIAL_XIRCOM=m
CONFIG_USB_SERIAL_WWAN=m
CONFIG_USB_SERIAL_OPTION=m
CONFIG_USB_SERIAL_OMNINET=m
CONFIG_USB_SERIAL_OPTICON=m
CONFIG_USB_SERIAL_VIVOPAY_SERIAL=m
CONFIG_USB_SERIAL_ZIO=m
CONFIG_USB_SERIAL_ZTE=m
CONFIG_USB_SERIAL_SSU100=m
CONFIG_USB_SERIAL_QT2=m
```

```
CONFIG_USB_SERIAL_DEBUG=m

#
# USB Miscellaneous drivers
#
CONFIG_USB_EMI62=m
CONFIG_USB_EMI26=m
CONFIG_USB_ADUTUX=m
CONFIG_USB_SEVSEG=m
CONFIG_USB_RIO500=m
CONFIG_USB_LEGOTOWER=m
CONFIG_USB_LCD=m
CONFIG_USB_LED=m
CONFIG_USB_CYPRESS_CY7C63=m
CONFIG_USB_CYTHERM=m
CONFIG_USB_IDMOUSE=m
CONFIG_USB_FTDI_ELAN=m
CONFIG_USB_APPLEDISPLAY=m
CONFIG_USB_SISUSBVGA=m
CONFIG_USB_SISUSBVGA_CON=y
CONFIG_USB_LD=m
CONFIG_USB_TRANCEVIBRATOR=m
CONFIG_USB_IOWARRIOR=m
CONFIG_USB_TEST=m
CONFIG_USB_ISIGHTFW=m
CONFIG_USB_YUREX=m
CONFIG_USB_EZUSB_FX2=m

#
# USB Physical Layer drivers
#
CONFIG_USB_ISP1301=m
CONFIG_USB_RCAR_PHY=m
CONFIG_USB_ATM=m
CONFIG_USB_SPEEDTOUCH=m
CONFIG_USB_CXACRU=m
CONFIG_USB_UEAGLEATM=m
CONFIG_USB_XUSBATM=m
CONFIG_USB_GADGET=y
# CONFIG_USB_GADGET_DEBUG is not set
# CONFIG_USB_GADGET_DEBUG_FILES is not set
CONFIG_USB_GADGET_DEBUG_FS=y
CONFIG_USB_GADGET_VBUS_DRAW=2
CONFIG_USB_GADGET_STORAGE_NUM_BUFFERS=2

#
# USB Peripheral Controller
#
# CONFIG_USB_FSL_USB2 is not set
# CONFIG_USB_FUSB300 is not set
# CONFIG_USB_R8A66597 is not set
# CONFIG_USB_IMX is not set
# CONFIG_USB_MV_UDC is not set
# CONFIG_USB_M66592 is not set
# CONFIG_USB_NET2272 is not set
# CONFIG_USB_DUMMY_HCD is not set
CONFIG_USB_LIBCOMPOSITE=y
# CONFIG_USB_ZERO is not set
# CONFIG_USB_AUDIO is not set
CONFIG_USB_ETH=y
CONFIG_USB_ETH_RNDIS=y
CONFIG_USB_ETH_EEM=y
```

```
# CONFIG_USB_G_NCM is not set
# CONFIG_USB_GADGETFS is not set
# CONFIG_USB_FUNCTIONFS is not set
# CONFIG_USB_MASS_STORAGE is not set
# CONFIG_USB_G_SERIAL is not set
# CONFIG_USB_MIDI_GADGET is not set
# CONFIG_USB_G_PRINTER is not set
# CONFIG_USB_CDC_COMPOSITE is not set
# CONFIG_USB_G_NOKIA is not set
# CONFIG_USB_G_ACM_MS is not set
# CONFIG_USB_G_MULTI is not set
# CONFIG_USB_G_HID is not set
# CONFIG_USB_G_DBGP is not set
# CONFIG_USB_G_WEBCAM is not set

#
# OTG and related infrastructure
#
CONFIG_USB_OTG_UTILS=y
# CONFIG_USB_GPIO_VBUS is not set
# CONFIG_USB_ULPI is not set
# CONFIG_NOP_USB_XCEIV is not set
CONFIG_USB_MXS_PHY=y
CONFIG_MMC=y
# CONFIG_MMC_DEBUG is not set
CONFIG_MMC_UNSAFE_RESUME=y
# CONFIG_MMC_CLKGATE is not set

#
# MMC/SD/SDIO Card Drivers
#
CONFIG_MMC_BLOCK=y
CONFIG_MMC_BLOCK_MINORS=8
CONFIG_MMC_BLOCK_BOUNCE=y
CONFIG_SDIO_UART=m
# CONFIG_MMC_TEST is not set

#
# MMC/SD/SDIO Host Controller Drivers
#
# CONFIG_MMC_ARMMMCI is not set
CONFIG_MMC_SDHCI=y
CONFIG_MMC_SDHCI_IO_ACCESSORS=y
CONFIG_MMC_SDHCI_PLTFM=y
CONFIG_MMC_SDHCI_ESDHC_IMX=y
# CONFIG_MMC_SDHCI_PXAV3 is not set
# CONFIG_MMC_SDHCI_PXAV2 is not set
# CONFIG_MMC_MXC is not set
# CONFIG_MMC_DW is not set
# CONFIG_MMC_VUB300 is not set
# CONFIG_MMC_USHC is not set
# CONFIG_MEMSTICK is not set
CONFIG_LEDS_GPIO_REGISTER=y
CONFIG_NEW_LEDS=y
CONFIG_LEDS_CLASS=y

#
# LED drivers
#
CONFIG_LEDS_LM3530=m
CONFIG_LEDS_LM3642=m
CONFIG_LEDS_PCA9532=m
```

```
# CONFIG_LEDS_PCA9532_GPIO is not set
CONFIG_LEDS_GPIO=y
CONFIG_LEDS_LP3944=m
CONFIG_LEDS_LP5521=m
CONFIG_LEDS_LP5523=m
CONFIG_LEDS_PCA955X=m
CONFIG_LEDS_PCA9633=m
CONFIG_LEDS_DAC124S085=m
CONFIG_LEDS_REGULATOR=m
CONFIG_LEDS_BD2802=m
CONFIG_LEDS_LT3593=m
# CONFIG_LEDS_MC13783 is not set
# CONFIG_LEDS_RENESAS_TPU is not set
CONFIG_LEDS_TCA6507=m
CONFIG_LEDS_LM355x=m
CONFIG_LEDS_OT200=m
CONFIG_LEDS_BLINKM=m
CONFIG_LEDS_TRIGGERS=y

#
# LED Triggers
#
CONFIG_LEDS_TRIGGER_TIMER=y
CONFIG_LEDS_TRIGGER_ONESHOT=y
CONFIG_LEDS_TRIGGER_HEARTBEAT=y
CONFIG_LEDS_TRIGGER_BACKLIGHT=y
CONFIG_LEDS_TRIGGER_CPU=y
CONFIG_LEDS_TRIGGER_GPIO=y
CONFIG_LEDS_TRIGGER_DEFAULT_ON=y

#
# iptables trigger is under Netfilter config (LED target)
#
CONFIG_LEDS_TRIGGER_TRANSIENT=y
# CONFIG_ACCESSIBILITY is not set
# CONFIG_EDAC is not set
CONFIG_RTC_LIB=y
CONFIG_RTC_CLASS=y
CONFIG_RTC_HCTOSYS=y
CONFIG_RTC_HCTOSYS_DEVICE="rtc0"
# CONFIG_RTC_DEBUG is not set

#
# RTC interfaces
#
CONFIG_RTC_INTF_SYSFS=y
CONFIG_RTC_INTF_PROC=y
CONFIG_RTC_INTF_DEV=y
CONFIG_RTC_INTF_DEV_UIE_EMUL=y
# CONFIG_RTC_DRV_TEST is not set

#
# I2C RTC drivers
#
CONFIG_RTC_DRV_DS1307=m
CONFIG_RTC_DRV_DS1374=m
CONFIG_RTC_DRV_DS1672=m
CONFIG_RTC_DRV_DS3232=m
CONFIG_RTC_DRV_MAX6900=m
CONFIG_RTC_DRV_RS5C372=m
CONFIG_RTC_DRV_ISL1208=m
CONFIG_RTC_DRV_ISL12022=m
```

```
CONFIG_RTC_DRV_X1205=m
CONFIG_RTC_DRV_PCF8523=m
CONFIG_RTC_DRV_PCF8563=m
CONFIG_RTC_DRV_PCF8583=m
CONFIG_RTC_DRV_M41T80=m
CONFIG_RTC_DRV_M41T80_WDT=y
CONFIG_RTC_DRV_BQ32K=m
CONFIG_RTC_DRV_S35390A=m
CONFIG_RTC_DRV_FM3130=m
CONFIG_RTC_DRV_RX8581=m
CONFIG_RTC_DRV_RX8025=m
CONFIG_RTC_DRV_EM3027=m
CONFIG_RTC_DRV_RV3029C2=m

#
# SPI RTC drivers
#
CONFIG_RTC_DRV_M41T93=m
CONFIG_RTC_DRV_M41T94=m
CONFIG_RTC_DRV_DS1305=m
CONFIG_RTC_DRV_DS1390=m
CONFIG_RTC_DRV_MAX6902=m
CONFIG_RTC_DRV_R9701=m
CONFIG_RTC_DRV_RS5C348=m
CONFIG_RTC_DRV_DS3234=m
CONFIG_RTC_DRV_PCF2123=m

#
# Platform RTC drivers
#
CONFIG_RTC_DRV_CMOS=m
CONFIG_RTC_DRV_DS1286=m
CONFIG_RTC_DRV_DS1511=m
CONFIG_RTC_DRV_DS1553=m
CONFIG_RTC_DRV_DS1742=m
CONFIG_RTC_DRV_STK17TA8=m
CONFIG_RTC_DRV_M48T86=m
CONFIG_RTC_DRV_M48T35=m
CONFIG_RTC_DRV_M48T59=m
CONFIG_RTC_DRV_MSM6242=m
CONFIG_RTC_DRV_BQ4802=m
CONFIG_RTC_DRV_RP5C01=m
CONFIG_RTC_DRV_V3020=m
CONFIG_RTC_DRV_DS2404=m

#
# on-CPU RTC drivers
#
# CONFIG_RTC_DRV_IMXDI is not set
# CONFIG_RTC_DRV_PL030 is not set
# CONFIG_RTC_DRV_PL031 is not set
CONFIG_RTC_DRV_MC13XXX=y
CONFIG_RTC_DRV_MXC=y
# CONFIG_RTC_DRV_SNVS is not set
CONFIG_DMADEVICES=y
# CONFIG_DMADEVICES_DEBUG is not set

#
# DMA Devices
#
# CONFIG_AMBA_PL08X is not set
# CONFIG_DW_DMAC is not set
```

```
CONFIG_MX3_IPU=y
CONFIG_MX3_IPU_IRQS=4
CONFIG_TIMB_DMA=m
# CONFIG_PL330_DMA is not set
CONFIG_IMX_SDMA=y
# CONFIG_IMX_DMA is not set
CONFIG_MXS_DMA=y
CONFIG_DMA_ENGINE=y

#
# DMA Clients
#
# CONFIG_NET_DMA is not set
# CONFIG_ASYNC_TX_DMA is not set
# CONFIG_DMATEST is not set
# CONFIG_AUXDISPLAY is not set
CONFIG_UIO=m
CONFIG_UIO_PDRV=m
CONFIG_UIO_PDRV_GENIRQ=m
# CONFIG_UIO_DMEM_GENIRQ is not set

#
# Virtio drivers
#
# CONFIG_VIRTIO_MMIO is not set

#
# Microsoft Hyper-V guest support
#
CONFIG_STAGING=y
# CONFIG_USBIP_CORE is not set
# CONFIG_W35UND is not set
CONFIG_PRISM2_USB=m
# CONFIG_ECHO is not set
# CONFIG_ASUS_OLED is not set
# CONFIG_RTLLIB is not set
CONFIG_R8712U=m
# CONFIG_RTS5139 is not set
# CONFIG_TRANZPORT is not set
# CONFIG_LINE6_USB is not set
# CONFIG_USB_SERIAL_QUATECH2 is not set
# CONFIG_VT6656 is not set

#
# IIO staging drivers
#
CONFIG_IIO_ST_HWMON=m
CONFIG_IIO_SW_RING=m

#
# Accelerometers
#
CONFIG_ADIS16201=m
CONFIG_ADIS16203=m
CONFIG_ADIS16204=m
CONFIG_ADIS16209=m
CONFIG_ADIS16220=m
CONFIG_ADIS16240=m
CONFIG_KXSD9=m
# CONFIG_LIS3L02DQ is not set
CONFIG_SCA3000=m
```

```
#
# Analog to digital converters
#
CONFIG_AD7291=m
CONFIG_AD7606=m
CONFIG_AD7606_IFACE_PARALLEL=m
CONFIG_AD7606_IFACE_SPI=m
CONFIG_AD799X=m
CONFIG_AD799X_RING_BUFFER=y
CONFIG_AD7780=m
CONFIG_AD7816=m
CONFIG_AD7192=m
CONFIG_ADT7410=m
CONFIG_AD7280=m

#
# Analog digital bi-direction converters
#
CONFIG_ADT7316=m
CONFIG_ADT7316_SPI=m
CONFIG_ADT7316_I2C=m

#
# Capacitance to digital converters
#
CONFIG_AD7150=m
CONFIG_AD7152=m
CONFIG_AD7746=m

#
# Direct Digital Synthesis
#
CONFIG_AD5930=m
CONFIG_AD9832=m
CONFIG_AD9834=m
CONFIG_AD9850=m
CONFIG_AD9852=m
CONFIG_AD9910=m
CONFIG_AD9951=m

#
# Digital gyroscope sensors
#
CONFIG_ADIS16060=m
CONFIG_ADIS16080=m
CONFIG_ADIS16130=m
CONFIG_ADIS16260=m
CONFIG_ADXRS450=m

#
# Network Analyzer, Impedance Converters
#
CONFIG_AD5933=m

#
# Inertial measurement units
#
CONFIG_ADIS16400=m

#
# Light sensors
#
```

```
CONFIG_SENSORS_ISL29018=m
CONFIG_SENSORS_ISL29028=m
CONFIG_SENSORS_TSL2563=m
CONFIG_TSL2583=m
CONFIG_TSL2x7x=m

#
# Magnetometer sensors
#
CONFIG_SENSORS_AK8975=m
CONFIG_SENSORS_HMC5843=m

#
# Active energy metering IC
#
CONFIG_ADE7753=m
CONFIG_ADE7754=m
CONFIG_ADE7758=m
CONFIG_ADE7759=m
CONFIG_ADE7854=m
CONFIG_ADE7854_I2C=m
CONFIG_ADE7854_SPI=m

#
# Resolver to digital converters
#
CONFIG_AD2S90=m
CONFIG_AD2S1200=m
CONFIG_AD2S1210=m

#
# Triggers - standalone
#
CONFIG_IIO_PERIODIC_RTC_TRIGGER=m
CONFIG_IIO_GPIO_TRIGGER=m
CONFIG_IIO_SYSFS_TRIGGER=m
# CONFIG_IIO_SIMPLE_DUMMY is not set
# CONFIG_ZSMALLOC is not set
# CONFIG_USB_ENESTORAGE is not set
# CONFIG_BCM_WIMAX is not set
# CONFIG_FT1000 is not set

#
# Speakup console speech
#
# CONFIG_SPEAKUP is not set
# CONFIG_TOUCHSCREEN_CLEARPAD_TM1217 is not set
# CONFIG_TOUCHSCREEN_SYNAPTICS_I2C_RMI4 is not set
CONFIG_STAGING_MEDIA=y
CONFIG_DVB_AS102=m
CONFIG_LIRC_STAGING=y
CONFIG_LIRC_IGORPLUGUSB=m
CONFIG_LIRC_IMON=m
CONFIG_LIRC_SASEM=m
CONFIG_LIRC_SERIAL=m
CONFIG_LIRC_SERIAL_TRANSMITTER=y
CONFIG_LIRC_SIR=m
CONFIG_LIRC_ZILOG=m

#
# Android
#
```

```
CONFIG_ANDROID=y
CONFIG_ANDROID_BINDER_IPC=y
CONFIG_ANDROID_LOGGER=m
CONFIG_ANDROID_TIMED_OUTPUT=y
CONFIG_ANDROID_TIMED_GPIO=m
# CONFIG_ANDROID_LOW_MEMORY_KILLER is not set
CONFIG_ANDROID_INTF_ALARM_DEV=y
# CONFIG_USB_WPAN_HCD is not set
# CONFIG_WIMAX_GDM72XX is not set
CONFIG_CSR_WIFI=m
# CONFIG_CED1401 is not set
# CONFIG_DRM_IMX is not set
# CONFIG_DGRP is not set
CONFIG_CLKDEV_LOOKUP=y
CONFIG_HAVE_CLK_PREPARE=y
CONFIG_COMMON_CLK=y

#
# Common Clock Framework
#
# CONFIG_COMMON_CLK_DEBUG is not set
CONFIG_COMMON_CLK_VERSATILE=y

#
# Hardware Spinlock drivers
#
CONFIG_CLKSRC_MMIO=y
CONFIG_IOMMU_SUPPORT=y
CONFIG_OF_IOMMU=y

#
# Remoteproc drivers (EXPERIMENTAL)
#
# CONFIG_STE_MODEM_RPROC is not set

#
# Rpmsg drivers (EXPERIMENTAL)
#
# CONFIG_VIRT_DRIVERS is not set
CONFIG_PM_DEVFREQ=y

#
# DEVFREQ Governors
#
CONFIG_DEVFREQ_GOV_SIMPLE_ONDEMAND=y
CONFIG_DEVFREQ_GOV_PERFORMANCE=y
CONFIG_DEVFREQ_GOV_POWERSAVE=y
CONFIG_DEVFREQ_GOV_USERSPACE=y

#
# DEVFREQ Drivers
#
# CONFIG_EXTCON is not set
# CONFIG_MEMORY is not set
CONFIG_IIO=y
CONFIG_IIO_BUFFER=y
CONFIG_IIO_BUFFER_CB=y
CONFIG_IIO_KFIFO_BUF=y
CONFIG_IIO_TRIGGERED_BUFFER=y
CONFIG_IIO_TRIGGER=y
CONFIG_IIO_CONSUMERS_PER_TRIGGER=2
```

```
#
# Accelerometers
#
CONFIG_HID_SENSOR_ACCEL_3D=m

#
# Analog to digital converters
#
CONFIG_AD_SIGMA_DELTA=m
CONFIG_AD7266=m
CONFIG_AD7298=m
CONFIG_AD7791=m
CONFIG_AD7793=m
CONFIG_AD7476=m
CONFIG_AD7887=m
CONFIG_MAX1363=m
CONFIG_TI_ADC081C=m

#
# Amplifiers
#
CONFIG_AD8366=m

#
# Hid Sensor IIO Common
#
CONFIG_HID_SENSOR_IIO_COMMON=m
CONFIG_HID_SENSOR_IIO_TRIGGER=m
# CONFIG_HID_SENSOR_ENUM_BASE_QUIRKS is not set

#
# Digital to analog converters
#
CONFIG_AD5064=m
CONFIG_AD5360=m
CONFIG_AD5380=m
CONFIG_AD5421=m
CONFIG_AD5624R_SPI=m
CONFIG_AD5446=m
CONFIG_AD5449=m
CONFIG_AD5504=m
CONFIG_AD5755=m
CONFIG_AD5764=m
CONFIG_AD5791=m
CONFIG_AD5686=m
CONFIG_MAX517=m
CONFIG_MCP4725=m

#
# Frequency Synthesizers DDS/PLL
#

#
# Clock Generator/Distribution
#
CONFIG_AD9523=m

#
# Phase-Locked Loop (PLL) frequency synthesizers
#
CONFIG_ADF4350=m
```

```
#
# Digital gyroscope sensors
#
CONFIG_ADIS16136=m
CONFIG_HID_SENSOR_GYRO_3D=m

#
# Inertial measurement units
#
CONFIG_ADIS16480=m
CONFIG_IIO_ADIS_LIB=m
CONFIG_IIO_ADIS_LIB_BUFFER=y

#
# Light sensors
#
CONFIG_ADJD_S311=m
CONFIG_VCNL4000=m
CONFIG_HID_SENSOR_ALS=m

#
# Magnetometer sensors
#
CONFIG_HID_SENSOR_MAGNETOMETER_3D=m
CONFIG_PWM=y
CONFIG_PWM_IMX=y
# CONFIG_IPACK_BUS is not set

#
# File systems
#
CONFIG_DCACHE_WORD_ACCESS=y
CONFIG_EXT2_FS=y
CONFIG_EXT2_FS_XATTR=y
CONFIG_EXT2_FS_POSIX_ACL=y
CONFIG_EXT2_FS_SECURITY=y
# CONFIG_EXT2_FS_XIP is not set
CONFIG_EXT3_FS=y
CONFIG_EXT3_DEFAULTS_TO_ORDERED=y
CONFIG_EXT3_FS_XATTR=y
CONFIG_EXT3_FS_POSIX_ACL=y
CONFIG_EXT3_FS_SECURITY=y
CONFIG_EXT4_FS=y
CONFIG_EXT4_FS_POSIX_ACL=y
CONFIG_EXT4_FS_SECURITY=y
# CONFIG_EXT4_DEBUG is not set
CONFIG_JBD=y
# CONFIG_JBD_DEBUG is not set
CONFIG_JBD2=y
# CONFIG_JBD2_DEBUG is not set
CONFIG_FS_MBCACHE=y
CONFIG_REISERFS_FS=m
# CONFIG_REISERFS_CHECK is not set
# CONFIG_REISERFS_PROC_INFO is not set
CONFIG_REISERFS_FS_XATTR=y
CONFIG_REISERFS_FS_POSIX_ACL=y
CONFIG_REISERFS_FS_SECURITY=y
CONFIG_JFS_FS=m
CONFIG_JFS_POSIX_ACL=y
CONFIG_JFS_SECURITY=y
# CONFIG_JFS_DEBUG is not set
CONFIG_JFS_STATISTICS=y
```

```
# CONFIG_XFS_FS is not set
# CONFIG_GFS2_FS is not set
# CONFIG_OCFS2_FS is not set
CONFIG_BTRFS_FS=y
CONFIG_BTRFS_FS_POSIX_ACL=y
# CONFIG_BTRFS_FS_CHECK_INTEGRITY is not set
CONFIG_NILFS2_FS=m
CONFIG_FS_POSIX_ACL=y
CONFIG_EXPORTFS=y
CONFIG_FILE_LOCKING=y
CONFIG_FSNOTIFY=y
CONFIG_DNOTIFY=y
CONFIG_INOTIFY_USER=y
CONFIG_FANOTIFY=y
CONFIG_FANOTIFY_ACCESS_PERMISSIONS=y
CONFIG_QUOTA=y
CONFIG_QUOTA_NETLINK_INTERFACE=y
CONFIG_PRINT_QUOTA_WARNING=y
# CONFIG_QUOTA_DEBUG is not set
CONFIG_QUOTA_TREE=m
CONFIG_QFMT_V1=m
CONFIG_QFMT_V2=m
CONFIG_QUOTACTL=y
CONFIG_AUTOFS4_FS=m
CONFIG_FUSE_FS=y
CONFIG_CUSE=m

#
# Caches
#
CONFIG_FSCACHE=m
CONFIG_FSCACHE_STATS=y
CONFIG_FSCACHE_HISTOGRAM=y
# CONFIG_FSCACHE_DEBUG is not set
# CONFIG_FSCACHE_OBJECT_LIST is not set
CONFIG_CACHEFILES=m
# CONFIG_CACHEFILES_DEBUG is not set
# CONFIG_CACHEFILES_HISTOGRAM is not set

#
# CD-ROM/DVD Filesystems
#
CONFIG_ISO9660_FS=m
CONFIG_JOLIET=y
CONFIG_ZISOFS=y
CONFIG_UDF_FS=m
CONFIG_UDF_NLS=y

#
# DOS/FAT/NT Filesystems
#
CONFIG_FAT_FS=y
CONFIG_MSDOS_FS=y
CONFIG_VFAT_FS=y
CONFIG_FAT_DEFAULT_CODEPAGE=437
CONFIG_FAT_DEFAULT_IOCHARSET="iso8859-1"
CONFIG_NTFS_FS=m
# CONFIG_NTFS_DEBUG is not set
CONFIG_NTFS_RW=y

#
# Pseudo filesystems
```

```
#
CONFIG_PROC_FS=y
CONFIG_PROC_SYSCTL=y
CONFIG_PROC_PAGE_MONITOR=y
CONFIG_SYSFS=y
# CONFIG_HUGETLB_PAGE is not set
CONFIG_CONFIGFS_FS=m
CONFIG_MISC_FILESYSTEMS=y
CONFIG_ADFS_FS=m
# CONFIG_ADFS_FS_RW is not set
CONFIG_AFFS_FS=m
CONFIG_ECRYPT_FS=m
CONFIG_HFS_FS=m
CONFIG_HFSPLUS_FS=m
CONFIG_BEFS_FS=m
# CONFIG_BEFS_DEBUG is not set
CONFIG_BFS_FS=m
CONFIG_EFS_FS=m
CONFIG_JFFS2_FS=m
CONFIG_JFFS2_FS_DEBUG=0
CONFIG_JFFS2_FS_WRITEBUFFER=y
# CONFIG_JFFS2_FS_WBUF_VERIFY is not set
# CONFIG_JFFS2_SUMMARY is not set
# CONFIG_JFFS2_FS_XATTR is not set
CONFIG_JFFS2_COMPRESSION_OPTIONS=y
CONFIG_JFFS2_ZLIB=y
CONFIG_JFFS2_LZO=y
CONFIG_JFFS2_RTIME=y
# CONFIG_JFFS2_RUBIN is not set
# CONFIG_JFFS2_CMODE_NONE is not set
CONFIG_JFFS2_CMODE_PRIORITY=y
# CONFIG_JFFS2_CMODE_SIZE is not set
# CONFIG_JFFS2_CMODE_FAVOURLZO is not set
CONFIG_UBIFS_FS=m
# CONFIG_UBIFS_FS_ADVANCED_COMPR is not set
CONFIG_UBIFS_FS_LZO=y
CONFIG_UBIFS_FS_ZLIB=y
CONFIG_LOGFS=m
CONFIG_CRAMFS=m
CONFIG_SQUASHFS=m
# CONFIG_SQUASHFS_XATTR is not set
CONFIG_SQUASHFS_ZLIB=y
CONFIG_SQUASHFS_LZO=y
CONFIG_SQUASHFS_XZ=y
# CONFIG_SQUASHFS_4K_DEVBLK_SIZE is not set
# CONFIG_SQUASHFS_EMBEDDED is not set
CONFIG_SQUASHFS_FRAGMENT_CACHE_SIZE=3
CONFIG_VXFS_FS=m
CONFIG_MINIX_FS=m
CONFIG_OMFS_FS=m
# CONFIG_HPFS_FS is not set
CONFIG_QNX4FS_FS=m
CONFIG_QNX6FS_FS=m
# CONFIG_QNX6FS_DEBUG is not set
CONFIG_ROMFS_FS=m
CONFIG_ROMFS_BACKED_BY_BLOCK=y
# CONFIG_ROMFS_BACKED_BY_MTD is not set
# CONFIG_ROMFS_BACKED_BY_BOTH is not set
CONFIG_ROMFS_ON_BLOCK=y
# CONFIG_PSTORE is not set
CONFIG_SYSV_FS=m
# CONFIG_UFS_FS is not set
```

```
CONFIG_F2FS_FS=y
CONFIG_F2FS_STAT_FS=y
CONFIG_F2FS_FS_XATTR=y
CONFIG_F2FS_FS_POSIX_ACL=y
CONFIG_NETWORK_FILESYSTEMS=y
# CONFIG_NFS_FS is not set
# CONFIG_NFSD is not set
# CONFIG_CEPH_FS is not set
# CONFIG_CIFS is not set
# CONFIG_NCP_FS is not set
# CONFIG_CODA_FS is not set
# CONFIG_AFS_FS is not set
CONFIG_NLS=y
CONFIG_NLS_DEFAULT="iso8859-1"
CONFIG_NLS_CODEPAGE_437=y
CONFIG_NLS_CODEPAGE_737=m
CONFIG_NLS_CODEPAGE_775=m
CONFIG_NLS_CODEPAGE_850=m
CONFIG_NLS_CODEPAGE_852=m
CONFIG_NLS_CODEPAGE_855=m
CONFIG_NLS_CODEPAGE_857=m
CONFIG_NLS_CODEPAGE_860=m
CONFIG_NLS_CODEPAGE_861=m
CONFIG_NLS_CODEPAGE_862=m
CONFIG_NLS_CODEPAGE_863=m
CONFIG_NLS_CODEPAGE_864=m
CONFIG_NLS_CODEPAGE_865=m
CONFIG_NLS_CODEPAGE_866=m
CONFIG_NLS_CODEPAGE_869=m
CONFIG_NLS_CODEPAGE_936=m
CONFIG_NLS_CODEPAGE_950=m
CONFIG_NLS_CODEPAGE_932=m
CONFIG_NLS_CODEPAGE_949=m
CONFIG_NLS_CODEPAGE_874=m
CONFIG_NLS_ISO8859_8=m
CONFIG_NLS_CODEPAGE_1250=m
CONFIG_NLS_CODEPAGE_1251=m
CONFIG_NLS_ASCII=m
CONFIG_NLS_ISO8859_1=y
CONFIG_NLS_ISO8859_2=m
CONFIG_NLS_ISO8859_3=m
CONFIG_NLS_ISO8859_4=m
CONFIG_NLS_ISO8859_5=m
CONFIG_NLS_ISO8859_6=m
CONFIG_NLS_ISO8859_7=m
CONFIG_NLS_ISO8859_9=m
CONFIG_NLS_ISO8859_13=m
CONFIG_NLS_ISO8859_14=m
CONFIG_NLS_ISO8859_15=m
CONFIG_NLS_KOI8_R=m
CONFIG_NLS_KOI8_U=m
CONFIG_NLS_MAC_ROMAN=m
CONFIG_NLS_MAC_CELTIC=m
CONFIG_NLS_MAC_CENTEURO=m
CONFIG_NLS_MAC_CROATIAN=m
CONFIG_NLS_MAC_CYRILLIC=m
CONFIG_NLS_MAC_GAELIC=m
CONFIG_NLS_MAC_GREEK=m
CONFIG_NLS_MAC_ICELAND=m
CONFIG_NLS_MAC_INUIT=m
CONFIG_NLS_MAC_ROMANIAN=m
CONFIG_NLS_MAC_TURKISH=m
```

```
CONFIG_NLS_UTF8=m
CONFIG_DLM=m
# CONFIG_DLM_DEBUG is not set

#
# Kernel hacking
#
CONFIG_PRINTK_TIME=y
CONFIG_DEFAULT_MESSAGE_LOGLEVEL=4
CONFIG_ENABLE_WARN_DEPRECATED=y
CONFIG_ENABLE_MUST_CHECK=y
CONFIG_FRAME_WARN=1024
CONFIG_MAGIC_SYSRQ=y
# CONFIG_STRIP_ASM_SYMS is not set
# CONFIG_READABLE_ASM is not set
CONFIG_UNUSED_SYMBOLS=y
CONFIG_DEBUG_FS=y
# CONFIG_HEADERS_CHECK is not set
# CONFIG_DEBUG_SECTION_MISMATCH is not set
CONFIG_DEBUG_KERNEL=y
# CONFIG_DEBUG_SHIRQ is not set
# CONFIG_LOCKUP_DETECTOR is not set
# CONFIG_PANIC_ON_OOPS is not set
CONFIG_PANIC_ON_OOPS_VALUE=0
CONFIG_DETECT_HUNG_TASK=y
CONFIG_DEFAULT_HUNG_TASK_TIMEOUT=120
# CONFIG_BOOTPARAM_HUNG_TASK_PANIC is not set
CONFIG_BOOTPARAM_HUNG_TASK_PANIC_VALUE=0
CONFIG_SCHED_DEBUG=y
CONFIG_SCHEDSTATS=y
CONFIG_TIMER_STATS=y
# CONFIG_DEBUG_OBJECTS is not set
# CONFIG_SLUB_DEBUG_ON is not set
# CONFIG_SLUB_STATS is not set
CONFIG_HAVE_DEBUG_KMEMLEAK=y
# CONFIG_DEBUG_KMEMLEAK is not set
# CONFIG_DEBUG_RT_MUTEXES is not set
# CONFIG_RT_MUTEX_TESTER is not set
# CONFIG_DEBUG_SPINLOCK is not set
# CONFIG_DEBUG_MUTEXES is not set
# CONFIG_DEBUG_LOCK_ALLOC is not set
# CONFIG_PROVE_LOCKING is not set
# CONFIG_SPARSE_RCU_POINTER is not set
# CONFIG_LOCK_STAT is not set
# CONFIG_DEBUG_ATOMIC_SLEEP is not set
# CONFIG_DEBUG_LOCKING_API_SELFTESTS is not set
CONFIG_STACKTRACE=y
# CONFIG_DEBUG_STACK_USAGE is not set
# CONFIG_DEBUG_KOBJECT is not set
# CONFIG_DEBUG_HIGHMEM is not set
CONFIG_DEBUG_BUGVERBOSE=y
# CONFIG_DEBUG_INFO is not set
CONFIG_DEBUG_VM=y
# CONFIG_DEBUG_VM_RB is not set
# CONFIG_DEBUG_WRITECOUNT is not set
# CONFIG_DEBUG_MEMORY_INIT is not set
# CONFIG_DEBUG_LIST is not set
# CONFIG_TEST_LIST_SORT is not set
# CONFIG_DEBUG_SG is not set
# CONFIG_DEBUG_NOTIFIERS is not set
# CONFIG_DEBUG_CREDENTIALS is not set
# CONFIG_BOOT_PRINTK_DELAY is not set
# CONFIG_RCU_TORTURE_TEST is not set
CONFIG_RCU_CPU_STALL_TIMEOUT=60
# CONFIG_RCU_CPU_STALL_INFO is not set
# CONFIG_RCU_TRACE is not set
# CONFIG_KPROBES_SANITY_TEST is not set
# CONFIG_BACKTRACE_SELF_TEST is not set
# CONFIG_DEBUG_BLOCK_EXT_DEVT is not set
# CONFIG_DEBUG_FORCE_WEAK_PER_CPU is not set
# CONFIG_DEBUG_PER_CPU_MAPS is not set
# CONFIG_LKDTM is not set
# CONFIG_NOTIFIER_ERROR_INJECTION is not set
# CONFIG_FAULT_INJECTION is not set
# CONFIG_DEBUG_PAGEALLOC is not set
CONFIG_NOP_TRACER=y
CONFIG_HAVE_FUNCTION_TRACER=y
CONFIG_HAVE_FUNCTION_GRAPH_TRACER=y
CONFIG_HAVE_DYNAMIC_FTRACE=y
CONFIG_HAVE_FTRACE_MCOUNT_RECORD=y
CONFIG_HAVE_SYSCALL_TRACEPOINTS=y
CONFIG_HAVE_C_RECORDMCOUNT=y
CONFIG_TRACE_CLOCK=y
CONFIG_RING_BUFFER=y
CONFIG_EVENT_TRACING=y
CONFIG_EVENT_POWER_TRACING_DEPRECATED=y
CONFIG_CONTEXT_SWITCH_TRACER=y
CONFIG_RING_BUFFER_ALLOW_SWAP=y
CONFIG_TRACING=y
CONFIG_TRACING_SUPPORT=y
CONFIG_FTRACE=y
# CONFIG_FUNCTION_TRACER is not set
# CONFIG_IRQSOFF_TRACER is not set
# CONFIG_SCHED_TRACER is not set
# CONFIG_ENABLE_DEFAULT_TRACERS is not set
# CONFIG_FTRACE_SYSCALLS is not set
CONFIG_BRANCH_PROFILE_NONE=y
# CONFIG_PROFILE_ANNOTATED_BRANCHES is not set
# CONFIG_PROFILE_ALL_BRANCHES is not set
# CONFIG_STACK_TRACER is not set
# CONFIG_BLK_DEV_IO_TRACE is not set
CONFIG_KPROBE_EVENT=y
CONFIG_PROBE_EVENTS=y
# CONFIG_RING_BUFFER_BENCHMARK is not set
# CONFIG_RBTREE_TEST is not set
# CONFIG_INTERVAL_TREE_TEST is not set
# CONFIG_DYNAMIC_DEBUG is not set
# CONFIG_DMA_API_DEBUG is not set
# CONFIG_ATOMIC64_SELFTEST is not set
# CONFIG_ASYNC_RAID6_TEST is not set
# CONFIG_SAMPLES is not set
CONFIG_HAVE_ARCH_KGDB=y
CONFIG_KGDB=y
CONFIG_KGDB_SERIAL_CONSOLE=y
# CONFIG_KGDB_TESTS is not set
CONFIG_KGDB_KDB=y
CONFIG_KDB_KEYBOARD=y
# CONFIG_TEST_KSTRTOX is not set
CONFIG_STRICT_DEVMEM=y
CONFIG_ARM_UNWIND=y
CONFIG_DEBUG_USER=y
# CONFIG_DEBUG_LL is not set
CONFIG_DEBUG_IMX6Q_UART_PORT=1
CONFIG_DEBUG_LL_INCLUDE="mach/debug-macro.S"
```

```
# CONFIG_OC_ETM is not set
# CONFIG_ARM_KPROBES_TEST is not set
# CONFIG_PID_IN_CONTEXTIDR is not set

#
# Security options
#
CONFIG_KEYS=y
# CONFIG_ENCRYPTED_KEYS is not set
CONFIG_KEYS_DEBUG_PROC_KEYS=y
# CONFIG_SECURITY_DMESG_RESTRICT is not set
CONFIG_SECURITY=y
CONFIG_SECURITYFS=y
CONFIG_SECURITY_NETWORK=y
# CONFIG_SECURITY_NETWORK_XFRM is not set
CONFIG_SECURITY_PATH=y
CONFIG_LSM_MMAP_MIN_ADDR=0
CONFIG_SECURITY_SELINUX=y
CONFIG_SECURITY_SELINUX_BOOTPARAM=y
CONFIG_SECURITY_SELINUX_BOOTPARAM_VALUE=0
CONFIG_SECURITY_SELINUX_DISABLE=y
CONFIG_SECURITY_SELINUX_DEVELOP=y
CONFIG_SECURITY_SELINUX_AVC_STATS=y
CONFIG_SECURITY_SELINUX_CHECKREQPROT_VALUE=1
# CONFIG_SECURITY_SELINUX_POLICYDB_VERSION_MAX is not set
CONFIG_SECURITY_SMACK=y
CONFIG_SECURITY_TOMOYO=y
CONFIG_SECURITY_TOMOYO_MAX_ACCEPT_ENTRY=2048
CONFIG_SECURITY_TOMOYO_MAX_AUDIT_LOG=1024
# CONFIG_SECURITY_TOMOYO_OMIT_USERSPACE_LOADER is not set
CONFIG_SECURITY_TOMOYO_POLICY_LOADER="/sbin/tomoyo-init"
CONFIG_SECURITY_TOMOYO_ACTIVATION_TRIGGER="/sbin/init"
CONFIG_SECURITY_APPARMOR=y
CONFIG_SECURITY_APPARMOR_BOOTPARAM_VALUE=1
# CONFIG_SECURITY_YAMA is not set
# CONFIG_IMA is not set
# CONFIG_EVM is not set
# CONFIG_DEFAULT_SECURITY_SELINUX is not set
# CONFIG_DEFAULT_SECURITY_SMACK is not set
# CONFIG_DEFAULT_SECURITY_TOMOYO is not set
CONFIG_DEFAULT_SECURITY_APPARMOR=y
# CONFIG_DEFAULT_SECURITY_DAC is not set
CONFIG_DEFAULT_SECURITY="apparmor"
CONFIG_XOR_BLOCKS=m
CONFIG_ASYNC_CORE=m
CONFIG_ASYNC_MEMCPY=m
CONFIG_ASYNC_XOR=m
CONFIG_ASYNC_PQ=m
CONFIG_ASYNC_RAID6_RECOV=m
CONFIG_CRYPTO=y

#
# Crypto core or helper
#
CONFIG_CRYPTO_ALGAPI=y
CONFIG_CRYPTO_ALGAPI2=y
CONFIG_CRYPTO_AEAD=m
CONFIG_CRYPTO_AEAD2=y
CONFIG_CRYPTO_BLKCIPHER=y
CONFIG_CRYPTO_BLKCIPHER2=y
CONFIG_CRYPTO_HASH=y
CONFIG_CRYPTO_HASH2=y
```

```
CONFIG_CRYPTO_RNG=m
CONFIG_CRYPTO_RNG2=y
CONFIG_CRYPTO_PCOMP=m
CONFIG_CRYPTO_PCOMP2=y
CONFIG_CRYPTO_MANAGER=y
CONFIG_CRYPTO_MANAGER2=y
# CONFIG_CRYPTO_USER is not set
CONFIG_CRYPTO_MANAGER_DISABLE_TESTS=y
CONFIG_CRYPTO_GF128MUL=m
CONFIG_CRYPTO_NULL=m
# CONFIG_CRYPTO_PCRYPT is not set
CONFIG_CRYPTO_WORKQUEUE=y
CONFIG_CRYPTO_CRYPTD=m
CONFIG_CRYPTO_AUTHENC=m
CONFIG_CRYPTO_TEST=m

#
# Authenticated Encryption with Associated Data
#
CONFIG_CRYPTO_CCM=m
CONFIG_CRYPTO_GCM=m
CONFIG_CRYPTO_SEQIV=m

#
# Block modes
#
CONFIG_CRYPTO_CBC=y
CONFIG_CRYPTO_CTR=m
CONFIG_CRYPTO_CTS=m
CONFIG_CRYPTO_ECB=y
CONFIG_CRYPTO_LRW=m
CONFIG_CRYPTO_PCBC=m
CONFIG_CRYPTO_XTS=m

#
# Hash modes
#
CONFIG_CRYPTO_HMAC=m
CONFIG_CRYPTO_XCBC=m
CONFIG_CRYPTO_VMAC=m

#
# Digest
#
CONFIG_CRYPTO_CRC32C=y
CONFIG_CRYPTO_GHASH=m
CONFIG_CRYPTO_MD4=m
CONFIG_CRYPTO_MD5=y
CONFIG_CRYPTO_MICHAEL_MIC=m
CONFIG_CRYPTO_RMD128=m
CONFIG_CRYPTO_RMD160=m
CONFIG_CRYPTO_RMD256=m
CONFIG_CRYPTO_RMD320=m
CONFIG_CRYPTO_SHA1=y
CONFIG_CRYPTO_SHA1_ARM=y
CONFIG_CRYPTO_SHA256=m
CONFIG_CRYPTO_SHA512=m
CONFIG_CRYPTO_TGR192=m
CONFIG_CRYPTO_WP512=m

#
# Ciphers
```

```
#
CONFIG_CRYPTO_AES=y
CONFIG_CRYPTO_AES_ARM=y
CONFIG_CRYPTO_ANUBIS=m
CONFIG_CRYPTO_ARC4=m
CONFIG_CRYPTO_BLOWFISH=m
CONFIG_CRYPTO_BLOWFISH_COMMON=m
CONFIG_CRYPTO_CAMELLIA=m
CONFIG_CRYPTO_CAST_COMMON=m
CONFIG_CRYPTO_CAST5=m
CONFIG_CRYPTO_CAST6=m
CONFIG_CRYPTO_DES=y
CONFIG_CRYPTO_FCRYPT=m
CONFIG_CRYPTO_KHAZAD=m
CONFIG_CRYPTO_SALSA20=m
CONFIG_CRYPTO_SEED=m
CONFIG_CRYPTO_SERPENT=m
CONFIG_CRYPTO_TEA=m
CONFIG_CRYPTO_TWOFISH=m
CONFIG_CRYPTO_TWOFISH_COMMON=m

#
# Compression
#
CONFIG_CRYPTO_DEFLATE=m
CONFIG_CRYPTO_ZLIB=m
CONFIG_CRYPTO_LZO=m

#
# Random Number Generation
#
CONFIG_CRYPTO_ANSI_CPRNG=m
CONFIG_CRYPTO_USER_API=m
CONFIG_CRYPTO_USER_API_HASH=m
CONFIG_CRYPTO_USER_API_SKCIPHER=m
CONFIG_CRYPTO_HW=y
CONFIG_ASYMMETRIC_KEY_TYPE=m
# CONFIG_ASYMMETRIC_PUBLIC_KEY_SUBTYPE is not set
CONFIG_BINARY_PRINTF=y

#
# Library routines
#
CONFIG_RAID6_PQ=m
CONFIG_BITREVERSE=y
CONFIG_RATIONAL=y
CONFIG_GENERIC_STRNCPY_FROM_USER=y
CONFIG_GENERIC_STRNLEN_USER=y
CONFIG_GENERIC_PCI_IOMAP=y
CONFIG_GENERIC_IO=y
CONFIG_STMP_DEVICE=y
CONFIG_PERCPU_RWSEM=y
CONFIG_CRC_CCITT=m
CONFIG_CRC16=y
CONFIG_CRC_T10DIF=y
CONFIG_CRC_ITU_T=m
CONFIG_CRC32=y
# CONFIG_CRC32_SELFTEST is not set
CONFIG_CRC32_SLICEBY8=y
# CONFIG_CRC32_SLICEBY4 is not set
# CONFIG_CRC32_SARWATE is not set
# CONFIG_CRC32_BIT is not set
```

```
CONFIG_CRC7=m
CONFIG_LIBCRC32C=y
CONFIG_CRC8=y
CONFIG_AUDIT_GENERIC=y
CONFIG_ZLIB_INFLATE=y
CONFIG_ZLIB_DEFLATE=y
CONFIG_LZO_COMPRESS=y
CONFIG_LZO_DECOMPRESS=y
CONFIG_XZ_DEC=y
CONFIG_XZ_DEC_X86=y
CONFIG_XZ_DEC_POWERPC=y
CONFIG_XZ_DEC_IA64=y
CONFIG_XZ_DEC_ARM=y
CONFIG_XZ_DEC_ARMTHUMB=y
CONFIG_XZ_DEC_SPARC=y
CONFIG_XZ_DEC_BCJ=y
# CONFIG_XZ_DEC_TEST is not set
CONFIG_DECOMPRESS_BZIP2=y
CONFIG_DECOMPRESS_LZMA=y
CONFIG_DECOMPRESS_XZ=y
CONFIG_DECOMPRESS_LZO=y
CONFIG_TEXTSEARCH=y
CONFIG_TEXTSEARCH_KMP=m
CONFIG_TEXTSEARCH_BM=m
CONFIG_TEXTSEARCH_FSM=m
CONFIG_BTREE=y
CONFIG_HAS_IOMEM=y
CONFIG_HAS_DMA=y
CONFIG_CPU_RMAP=y
CONFIG_DQL=y
CONFIG_NLATTR=y
CONFIG_ARCH_HAS_ATOMIC64_DEC_IF_POSITIVE=y
CONFIG_LRU_CACHE=m
CONFIG_AVERAGE=y
# CONFIG_CORDIC is not set
# CONFIG_DDR is not set
```