MICROCHIP

AN2772

Linux® Basics and Solutions for Microprocessors

Scope

This application note provides all the information required to get a quick start on Microchip microprocessors using a Linux operating system. It presents www.linux4sam.org, as well as essential information and general principles regarding open source. In addition, it offers links to key resources on the web to help developers of embedded Linux solutions to run, build and use this feature-rich ecosystem.

Table of Contents

1. Glossary

Term	Definition
Bootloader	Software component required to start an operating system (OS). It configures the minimum infrastructure required for the next level bootloader or operating system kernel to start. In a Linux context, bootloaders configure the external memory, set up some of the clocks and provide the information needed for the Linux kernel to boot. For a Microchip MPU system, the bootloader provides the device tree location to the Linux kernel, for example. AT91Bootstrap and U-Boot are bootloaders used in a Linux-based system on Microchip SoCs.
(Linux) Distribution	A collection of software components around the Linux kernel that, assembled together, form a coherent operating system with a set of applications.
GIT	Source code management system used by a software project to store, organize and trace the evolution of the source code.
Kernel	Central software piece of an operating system. The kernel provides a consistent and simplified access to the hardware. It ties together all other software components by providing services and organizing their co-operation. Due to its proximity to the hardware, the kernel usually has extended access rights to the hardware.
Mainline	Main source of a software component, and place where the project is being developed first. All derivatives of the software components are based on this unique ancestor. In a Linux kernel context, the Mainline is the place where all components are assembled regularly by its creator and head contributor Linus Torvalds, accessible on the popular website www.kernel.org. The same concept exists with most open source projects such as U-Boot or Buildroot.
Maintainer	The person that is responsible for a software project's source code, or for an identified piece of this project. In open source projects, the maintainer determines if an enhancement can be applied or not to the software, advises contributors and organizes development by showing the direction.
MPU	Microprocessor (stands for MicroProcessor Unit)
Patch	In a software development context for one project, a set of modifications collected together that covers a single topic. This enhancement contains changes to the source code of one or several files and can be described with a summary and an introduction text for better understanding. A patch usually takes the form of a text file with a special format which can be read, commented and easily exchanged through digital communication channels like e-mails.

Term	Definition
Root Filesystem (rootfs)	Files and directories, stored in a filesystem, that form the base of a Linux-based operating system. These components are a collection of applications, configuration files and libraries that form the operating system user space.
SoC	System on Chip: processor core and associated peripherals, connected through buses and memories that make a complete electronic system on a single piece of silicon. Microchip MPUs are SoCs.
Upstream	When a software component is copied and derived from an original component, forming a hierarchical organization, the original source of all derivatives is said to be “upstream”. See *Mainline*.
User Space	Part of the operating system and applications that has restricted access to hardware. Communicates with the hardware through the kernel (see *Kernel*).

2. Microchip Linux4SAM

2.1 Making a Long Story Short

Linux and Microchip's microprocessors have gone hand in hand since 2002 when our first MPU, the AT91RM9200, was introduced. Mainlining the AT91RM9200 drivers became a centerpiece of our Linux strategy and was achieved in 2004. The AT91RM9200 peripheral drivers are still the basis for all Microchip's MPUs, which facilitates software portability between them.

With our commitment to the Linux open source community, we provide full coverage of SoC peripherals in the Linux kernel as well as bootloaders such as AT91Bootstrap and U-Boot. Cross-build environments such as Buildroot, OpenWrt and distributions from the Yocto Project enable generation of Linux user space applications and customized root filesystems.

Microchip offers the Linux4SAM website (www.linux4sam.org) to provide access to new developments reviewed by our Linux mainliners and fully tested on our hardware platforms while waiting to be mainlined. Long-Term Support (LTS) Linux kernel updates are merged regularly with our kernel branch, to provide access to maintenance periods of 2 to 6 years.

Linux beginners can learn to build distributions from binary demos and step-by-step instructions featured on the Linux4SAM website. Experienced Linux users can visit Linux4SAM for source code exposed in well-structured GIT trees and matching open-source project quality standards, including detailed enhancement information, changelogs, plus the ability to contribute source code.

2.2 Introducing Linux4SAM

www.linux4sam.org is the reference website for information related to Linux and open source for Microchip microprocessors. The website provides comprehensive solutions to help elaborate Linux-based systems, from prebuilt reference distributions to associated instructions to build the systems from source code.

Figure 2-1. Linux4SAM Website Landing Page

2.3 Supported SoCs and Boards

The boards supported by our Linux4SAM offer are listed below. The links directly lead to prebuilt Linux distribution images hosted on the www.linux4sam.org website. The website also provides all the instructions needed to program and use the boards:

SAMA5D27 SOM1 EK

SAMA5D2 PTC EK

SAMA5D2 Xplained

SAMA5D3 Xplained

SAMA5D4 Xplained

AT91SAM9x5-EK

2.4 Components of a Linux-based Solution

A Linux-based operating system is comprised of bootloaders, a Linux kernel and a root filesystem. To start the operating system, bootloaders are executed sequentially, then they hand over operation to the kernel. From this point, the kernel and components of the root filesystem run together to execute the operating system services.

2.4.1 Bootloaders: AT91Bootstrap and U-boot

AT91Bootstrap is the second-level bootloader for Microchip's microprocessors. It provides a set of algorithms to manage the hardware initialization, such as clock speed configuration, PIO settings and DRAM initialization.

U-Boot (Universal-Boot Loader) is a third-level bootloader. It configures the main interfaces such as USB and Ethernet, and launches the Linux operating system. U-Boot facilitates the initial board bring-up and software development phases.

Note: The first-level bootloader is the embedded ROM code included in the MPU. Functionality and operation are described in the datasheet of the MPU.

2.4.2 Linux Kernel

The kernel, the central piece of software of the operating system, was created in 1991 by Linus Torvalds. It has been developed since then by a community of software enthusiasts, companies, universities and computer science professionals. By extension, the term can refer to the whole operating system and not only its kernel.

The Linux4SAM Linux Kernel pages provide information about the Linux kernel for Microchip microprocessors.

2.4.3 Yocto Project

The Yocto Project is an open source collaborative project that provides templates, tools, methods and documentation to help create Linux-based systems for embedded products. The free tools (emulation environments, debuggers, Application Toolkit Generator, etc.) are easy to use to get started and powerful, and they allow projects to be carried forward over time.

On Linux4SAM, the Yocto Project helped us build a distribution for Microchip microprocessors.

2.4.4 Buildroot

Buildroot is a simple, efficient and easy-to-use tool to generate embedded Linux systems through cross-compilation. It is an alternative to the Yocto Project and also an open source project, with many developers contributing to it.

On Linux4SAM, the Buildroot page provides instructions to build a Buildroot distribution for Microchip microprocessors.

2.4.5 OpenWrt

OpenWrt is a network-oriented Linux-based operating system. It allows the user to tailor an embedded distribution and is known to have a small footprint. Its state-of-the-art and comprehensive toolset and handy Web-based user interface are valuable components of the OpenWrt offer.

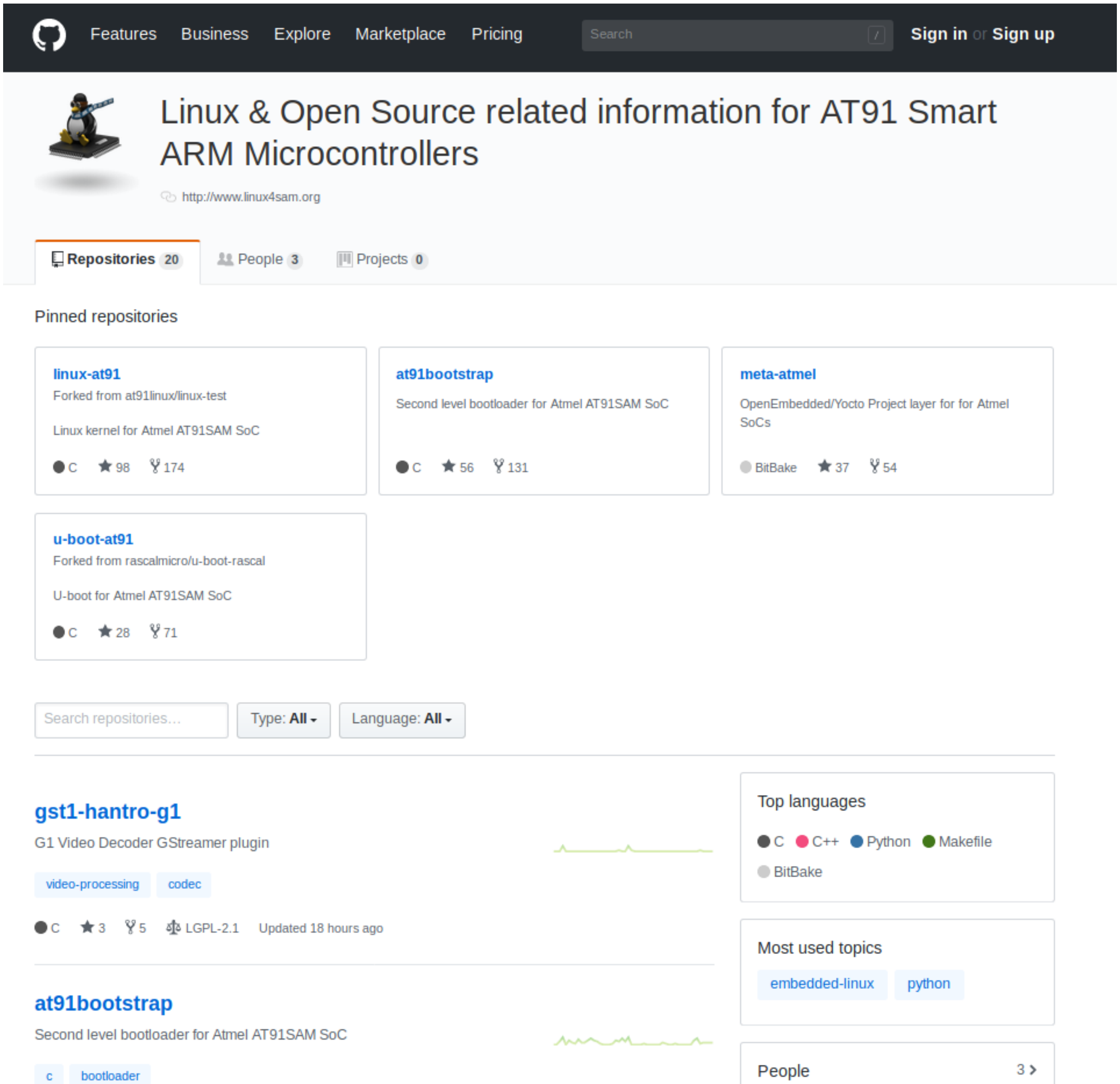
On Linux4SAM, the OpenWrt page provides instructions to build an OpenWrt distribution for Microchip microprocessors.

2.5 Linux4SAM Source Code

All source code for all software components is hosted on Linux4SAM GitHub:

https://www.github.com/linux4sam

Figure 2-2. Linux4SAM GitHub Landing Page Giving Access to Source Code



These repositories are organized by component, using the standard source code management tool GIT. GitHub provides an additional layer of presentation of the GIT trees. As the command line interface to GIT, it can be used to browse the source code and history of changes thoroughly.

The landing page, shown above, provides direct access to the components listed in the previous chapter.

In addition, Linux4SAM GitHub hosts source code for reference applications and libraries used by our prebuilt software solutions like:

- Gstreamer plug-in for our hardware video decoder: gst1-hantro-g1
- Hardware video decoder library: g1_decoder
- DRM LCD planes library: libplanes
- Microchip Peripheral I/O Python® Package: mpio
- SAMA5D2 Peripheral Touch Controller firmware, configuration files and examples: ptc_examples

- QT5 graphical application examples: application-launcher, home-automation

2.6 Linux4SAM Links to Communities

This section aggregates the contact information from open source communities that are represented in our ecosystem. The links to the main original projects (i.e., component mainlines) are the following:

- AT91Bootstrap
  Contributing_to_AT91Bootstrap
- U-Boot
  http://www.denx.de/wiki/U-Boot
- Linux Kernel
  http://lkml.org or http://lore.kernel.org

  Arm® architecture dedicated mailing lists:

  http://www.arm.linux.org.uk/mailinglists/

  http://lists.infradead.org/mailman/listinfo/linux-arm-kernel
- Yocto Project
  https://www.yoctoproject.org/tools-resources/community
- Buildroot
  https://buildroot.org/contribute.html
- OpenWrt
  https://openwrt.org/submitting-patches

A forum is open on www.at91.com to enable discussions about the overall offer and binary demos, and to collect ideas and contributions to Microchip components.

3. Microchip Linux4SAM Open Source Policy

3.1 Mainline Benefits

All of Microchip's Linux microprocessor development code is mainlined in the Linux and open source communities. With our experience, we fully understand and promote the benefits of publishing code in the Mainline. These benefits are discussed in the sections that follow.

3.1.1 A Community

By nature, open source is built by contributions from people willing to participate in a software project. These contributions are combined following predefined rules of quality and conformance, with the goal of the project assessed by the maintainers. The whole process takes place in public so that everyone can contribute, discuss technical arguments, contradict the author or even verify the good health of the project.

This method of developing software provides access to the world's largest community of qualified software engineers. These engineers give advice, review modifications, apply a demanding publication process and finally accept your code for inclusion! This is an invaluable support and definitively gives solid grounds to your project.

3.1.2 Covering a Wide Range of Needs

Customers or users of microprocessors must address a wide range of needs. As the coordinator of the embedded system, the application processor running a full-featured operating system is often at the crossroads of peripherals, interfaces and communication links. With its comprehensive set of device drivers or user space applications, the Linux ecosystem is well positioned to cover those needs.

However, it is important to have access to this diversity although some drivers or stacks are only available in the latest revisions of the open source project. Some special enhancements to the Linux kernel, for example the popular real-time patch PREMPT_RT or the Long-Term Support kernel, are only available on some revisions of the kernel.

With the use of Mainline kernel, difficult alignments between project needs, hardware support availability, support for a special flavor of Linux and a version of the kernel are no longer challenges: choose the Mainline and benefit from all those advantages.

3.1.3 Ease of Use

As described above, being in control and choosing whichever version of Linux suits your needs makes the developer's or integrator's life easier. Mainline becomes the only reference and www.kernel.org the only website to monitor for updates. Project software designers face fewer constraints since they do not have to collect drivers through different sources, nor to deal with heterogeneous websites or ways to distribute the source code.

One aspect is to have the source code, another is to be able to build a usable application from it. For embedded systems, the building procedure usually involves a complicated cross-platform build infrastructure known as cross-compilation. Using the kernel Mainline eases this process, as the procedure to build is standard and well-known, and documented in detail on internet. Furthermore, as the kernel Mainline is already integrated in all embedded build systems such as the Yocto Project, Buildroot or OpenWrt, no modification of the standard, well-proven process is required to gain access to a freshly built kernel.

3.1.4 Peace of Mind

The rigorous Linux mainlining review and acceptance process, combined with the worldwide user base, creates a solid and proven code base. The presence of device driver code in the Mainline is a proof of commitment from the peripheral's vendor that reassures the user. Presence in the Mainline is a common criterion for choosing hardware.

The source code contributed to the Mainline following the publication process is kept throughout project revisions and is not supposed to be removed, unless it is unmaintained for a long time or completely broken. This source code durability and continuity are factors of stability and lower the risk that support for specific hardware disappears.

Open source is independent from any software vendor and, in large projects, ownership is distributed. This also lowers the risk of losing track of a software project or seeing it completely changing its purpose or initial policy.

Trust and continuity are values that make users choose open source and bring them peace of mind.

3.2 Targeting Mainline First

Now that the benefits of open source have been described, it is time to explain how it works for Microchip microprocessors.

3.2.1 Being Part of It

Based on years of experience, our team tailors its work according to the open source principles and plays the community game. In other words, we are part of the community — using its tools, participating in events and in decision-making processes and acting as official maintainers. However, what our experience has taught us is that there are many ways to fail while trying to contribute to open source projects.

3.2.2 Human Factor

It is important to remember that the Mainline is a moving target, constantly enhanced by human developers. So, although it is based on highly technical topics, the contribution to an open source project leaves a great deal of freedom to human interaction. Fail to grasp this aspect and you lose the opportunity to participate in this movement.

The Mainline is important to us, as we consider that both contributors of open source and upstream maintainers serve the benefit of the whole project and general interest. We submit the code early during the development process to get feedback, and involve the whole community during discussions on the source code, because we value reviews from contributors and trust that discussing different options makes the final output better. Discussion with seasoned developers of one subsystem helps to figure out the bigger picture, beyond the special use case currently addressed. The solution is then much more generic and solid because it applies to a wider user base, is more tried and tested and is consequently more likely to be free of bugs.

Writing new code, submitting it publicly for discussion, paying attention to reviews and implementing suggested enhancements is the routine that we apply on source code that we create in addition to the project's Mainline.

3.2.3 A Fast Moving Target

To give an idea of how fast open source projects can move, here are the statistics collected from the latest Linux kernel source tree at the time of this writing. The release is Linux 4.17, and is standard in terms of statistics:

- The total amount of code is about **17 million lines of source code over 47,900 files** (mostly C) as counted by cloc, which computes effective lines of code without comments or blank lines.
- The release of the 4.17 Linux kernel took **2 months** (from 1st April to 3rd June 2018). Each new version was released after a development period which lasted from 2 to 2.5 months.
- During these 2 months, the total of changed lines was 1,560,440, i.e., an average of more than **24,300 changed lines a day**.
- During these 2 months, the total number of change sets or “patches” was 13,541, which represents an average of **8.8 changes an hour**!
- This amount of work for this kernel revision was handled by **1,714 developers** working together on the same project from all around the world.

Sources of statistics:

- git log of Linus Torvalds' kernel tree

- gdm (git data miner): git://git.lwn.net/gitdm.git

- cloc (count lines of code): https://github.com/AlDanial/cloc

These numbers make it clear that not following the pace of developments in such a project can make a contribution obsolete after a short period if it is developed on an older revision of the kernel, outside the Mainline. Therefore, trying to catch up with Linux development while staying outside, or starting with an already obsolete version, is a waste of time and energy.

The right way to address this "moving target phenomenon", as we have been doing for years, is to:

1. choose the Mainline as a starting point at each stage of the development — a process that we call “Mainline first”,
2. prepare a Linux port to a new SoC, a new driver or a new feature as early as the prototyping phase so that a “Request For Comments” can be posted early,
3. if code developed has to be collected, packaged and delivered before the mainlining process is completed, create a vendor’s tree that follows the Mainline closely. This vendor’s tree for Microchip microprocessors is called “Linux4SAM”.

These steps should be followed in this order to avoid porting the code endlessly from one kernel revision to the other, or refactoring the code completely after an unfavorable in-depth review.

4. Linux Mainline and Linux4SAM

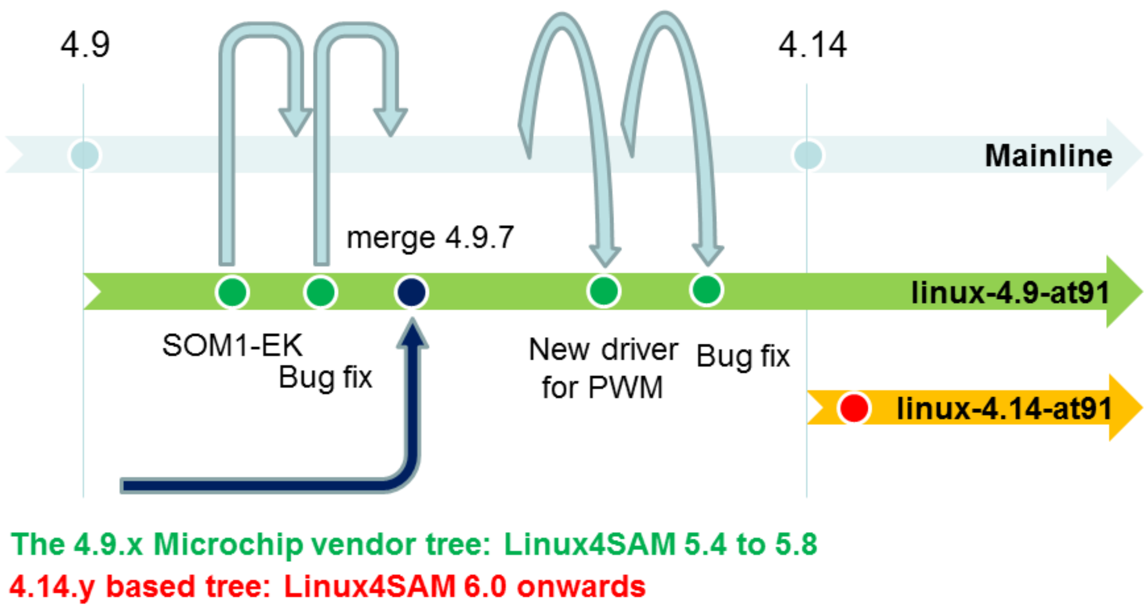
To overcome the difficulties of the Mainline process, Microchip has created Linux4SAM distributions. These distributions make it easier to move from a development perspective to a ready-to-production delivery.

Component versions are chosen according to the latest stable releases, as is the case for the Yocto Project or LTS versions for Linux. Furthermore, some backported developments are added that have not yet made it to the Mainline. Some other parts are enhancements that cannot be merged into the Mainline, such as a hardware video decoder or 9-bit serial line communication support.

For a given version of Linux, our Linux4SAM release includes the latest stable patches from the official Linux stable tree.

We have been repeating this process for years, evolving from one component revision to the next, following closely the development of the open source projects used. For example, we have provided Linux4SAM kernels from version 2.6.39 to the latest LTS 4.14.

Figure 4-1. Relationships between Linux Mainline and Linux4SAM Kernel Branches



Each Linux4SAM release includes:

- A new release of our second-level bootloader AT91Bootstrap (with the form v3.x.y)
- A unique version or tag on each software component (with the form linux4sam_x.y)

Tags can be browsed in all GIT trees and on GitHub, and are noted in the table presenting binary archives. This process offers a complete traceability from the source code to the prebuilt reference distributions.

Figure 4-2. Example of Demo Archives Table for SAMA5D27 SOM1 EK

Demo archives

MEDIA TYPE	BOARD	SCREEN	BINARY	DESCRIPTION
YOCTO PROJECT / POKY BASED DEMO				
SD Card image	SAMA5D27 SOM1 EK	-	linux4sam-poky-sama5d27_som1_ek-5.8.img.bz2 (~ 48 MB) md5: f31108e066f80bf55886fee021e8ca46	Linux4SAM Yocto Project / Poky based demo compiled from tag **linux4sam_5.8** Follow procedure: #Create_a_SD_card_with_the_demo
		PDA 4.3"	linux4sam-poky-sama5d27_som1_ek_pda4-5.8.img.bz2 (~ 139 MB) md5: 5edc9da4f7763e0d4a16b68dafeb0881	
		PDA 7" TM7000	linux4sam-poky-sama5d27_som1_ek_pda7-5.8.img.bz2 (~ 139 MB) md5: ac10167653e6e368d015f8309de84cd3	
		PDA 7" TM7000B	linux4sam-poky-sama5d27_som1_ek_pda7b-5.8.img.bz2 (~ 139 MB) md5: ccec62299cd99cb6cdc6dba3f5cfd7d8	
BUILDROOT BASED DEMO				
SD Card image	SAMA5D27 SOM1 EK	-	linux4sam-buildroot-sama5d27_som1_ek-5.8.img.bz2 (~ 28 MB) md5: 28451375df22638f8c6a2de0b4fc1f07	Linux4SAM BuildRoot based demo compiled from tag **linux4sam_5.8** Follow procedure: #Create_a_SD_card_with_the_demo
		PDA 7" TM7000	linux4sam-buildroot-sama5d27_som1_ek_pda7-5.8.img.bz2 (~ 154 MB) md5: 3efb700c5eda5ed08e47dbca4ab48252	
OPENWRT BASED DEMO				
SD Card image	SAMA5D27 SOM1 EK	-	linux4sam-openwrt-sama5d27_som1_ek-5.8.img.gz (~ 7 MB) md5: 13e059c833eb305217ec22319b826b2a	Linux4SAM OpenWrt based demo compiled from tag **linux4sam_5.8** Follow procedure: #Create_a_SD_card_with_the_demo

Choosing a Linux4SAM release provides the following benefits:

- Simplified evaluation: an archive is proposed for each board, each screen arrangement and each root filesystem “flavor” (see table in the above figure)
- Flashing methods provided: all archives have SAM-BA scripts(see Note) for flashing your demo on your board. SD Card images are ready to be programmed.
- Out-of-the-box experience: all components are tested independently and then integrated. Integration tests are performed on these deliveries.
- For prebuilt distributions, we provide prebuilt SDK for user-space application development.

Note: Microchip's SAM Boot Assistant (SAM-BA) software provides an open set of tools for programming Microchip Arm-based microprocessors. Customers can use SAM-BA as a tool to reprogram the MPU without connecting a debugger.

5. Conclusion

www.linux4sam.org is a Microchip resource that collects information and source code and provides prebuilt reference distributions to ease the evaluation of our microprocessors. The website is easy to access and is the main starting point for users looking for a Linux-based Microchip solution. Microchip is committed to participating in the lively open source communities that characterize Linux and to contributing to the Mainline. Our policy regarding source code development ensures that our customers always have access to the latest resources in order to get started quickly.

6. Revision History

6.1 Rev. A - 08/2018

This is the initial released version of this application note.

The Microchip Web Site

Microchip provides online support via our web site at http://www.microchip.com/. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Customer Change Notification Service

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at http://www.microchip.com/. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.

- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Trademarks

ISBN: 978-1-5224-3383-5

Quality Management System Certified by DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4450-2828 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-67-3636 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-7289-7561 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820