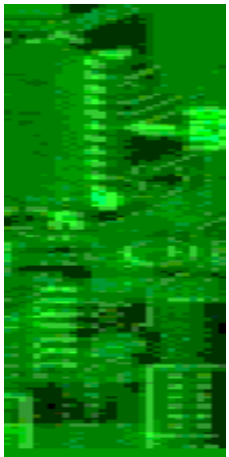
Real-time Support under Linux from Innvo Systems Pte Ltd, Singapore



SYSTEM BUILDER

Innvo's System Builder (SB) provides tools and utilities for the developer of the embedded device to design, develop and deploy his embedded system.

SB is customized to the requirements of the device, giving the developer a jumpstart into embedded application development.

The tools within SB include:

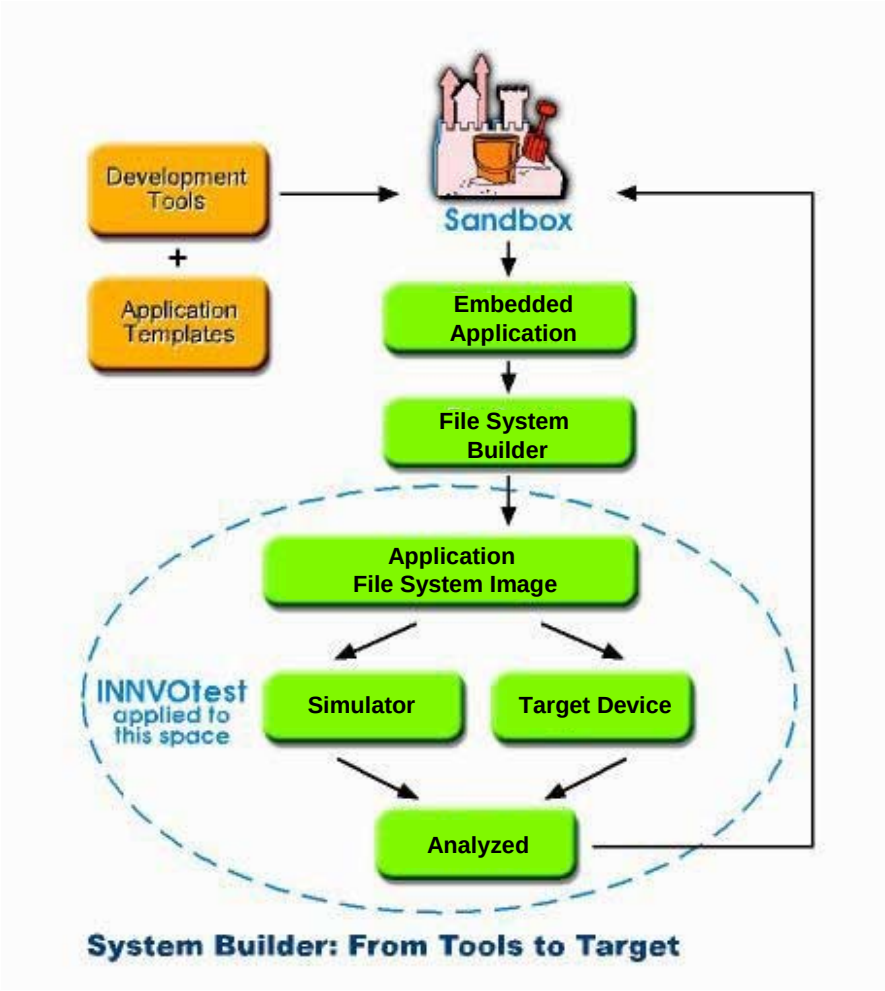
- Development Tools
- Application Templates
- Sandbox
- File System Builder
- Simulator
- INNVOtest
- INNVO Embedded Middleware (IEM)

Features of System Builder

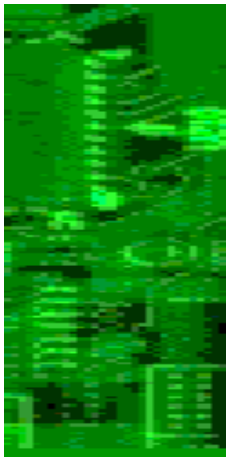
- **Development Tools** – consist of cross compilers, debuggers, library-optimizers, etc. that are well documented and easily extensible.
- **Application Templates** – is customized for the developer to suit his needs in application development efforts.
- **Sandbox** – provides an application testing and debugging environment, which can be done directly on the target or on a Simulator running on a host machine.
- **Simulator** – provide the developer with an application simulation framework of the target device.
- **INNVOtest** – is a leading edge code testing and verification tool for qualifying an application's “code quality”.
- **IEM** – is a patent-pending and first of its kind middleware platform for Embedded Linux.

System Builder: From Tools to Target

With System Builder, device developers will be able to significantly reduce their development time and extend their efforts. They will enjoy reduced systems development requirements:



- Development Tools and a customized set of Application Templates are provided to adhere to the requirements of the developer.
- Within the Application Templates are “feature demos” which can be used to further extend the embedded application – the developer does not have to start from scratch.
- These are provided within Sandbox, which is the work space where the developer design, test, and integrate his application.



- The embedded applications and its other components are integrated by the File System Builder and made into either a Simulator image or target device image.
- This image is then put through rigorous automated early phase testing and analysis by INNVOtest – a leading edge code testing and verification tool for qualifying an application s "code quality".

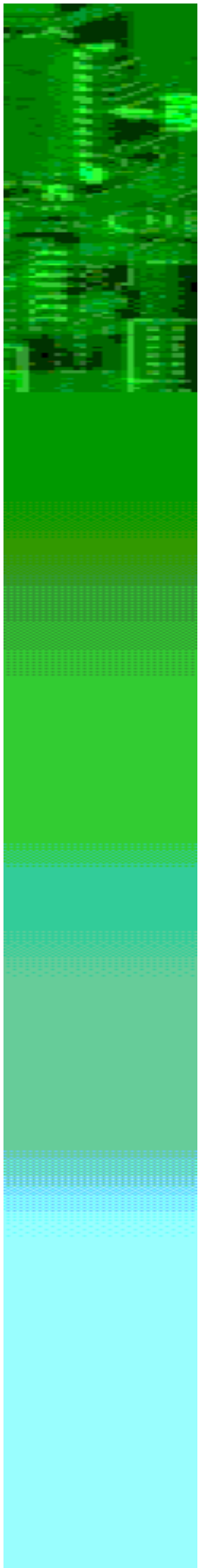
PCM713 BOARD SUPPORT PACKAGE

Innvo provides a complete range of support services and packages for you to confidently design your devices and rollout your products on time.

Whether you are in the process of explorative development or fighting the clock to build your perfect device, Innvo can offer you the kind of support you always hope for. The PCM7130 Board Support Package will comprise the following:

Standard Suite
INNVOBoot Bootloader
Linux Kernel v2.4.18 for PCM7130
MTD/Flash support
Filesystem - JFFS2/CRAMFS enabled
Epson S1D13806 Framebuffer Support
SMC91C96 10 Mbps Ethernet Support
SA1111 PS/2 Mouse/Keyboard Support
GPIO/DIO Support
USB Client, Networking Support
PCMCIA / CF device support
Support for RTLinux modules in kernel
OpenRTLinux v3.0 Suite
Innvo "System Builder"
Extended Suite
SA1111 AC97 Audio Codec Support
DS1670 Watchdog Timer / Buzzer Support
USB Host Controller Support
Extra Value
RTLinux Pro v3.1 Suite
GPIO/DIO Customized Driver

Demo Suite
INNVOBoot Bootloader
Linux Kernel v2.4.18 for PCM7130
MTD/Flash support
Filesystem - JFFS2/CRAMFS enabled
Epson S1D13806 Framebuffer Support
SMC91C96 10 Mbps Ethernet Support
SA1111 PS/2 Mouse/Keyboard Support
GPIO/DIO Support
USB Client, Networking Support
PCMCIA / CF device support
Support for RTLinux modules in kernel
Innvo "System Builder" Demo Suite
OpenRTLinux v3.0 Suite



Standard and Extended Suite

The normal Board Support Package will provide the Standard Suite. The Extended Suite is optional, and will be made available on a need-provide basis. Customers can choose components from the Extended Suite depending on their requirements. The solution is priced accordingly.

Extra Value

RTLinux v3.1 Professional will be made available to those who have Hard Realtime requirements, but cannot license their applications under GPL (as required under OpenRTLinux v3.x). We also offer special customization of drivers for exclusive requirements.

Demo Suite

The Demo Suite is for pure trial purposes and is to let the customer have a feel of the product before making a buy decision

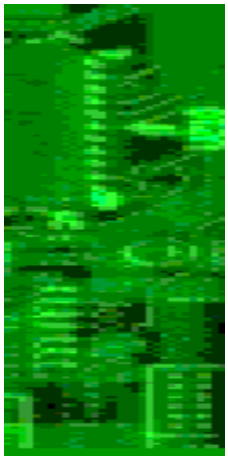
The Standard and Extended Suites of the Board Support Package will be available with the complete source code of all the packages. InnvoBoot source can be optionally made available under special licensing terms only.

OPENRTLINUX V3.0 ON PCM7130

Introduction

The term ‘Realtime’ can be used in several contexts in Embedded Systems. If a task needs to perform operations with a short and consistent response time to an input, it is normally referred to as a Realtime task.

Examples of such tasks are Input device handling, Network stream handling, Multimedia Video support and the like. In all these cases, it is required to process the task within a predetermined time limit. In simple terms each task has to be done within a pre-defined deadline.



Hard Realtime and Soft Realtime

Realtime tasks are broadly classified into Hard Realtime and Soft Realtime tasks.

Soft Realtime

Soft Realtime tasks are a class of Realtime tasks where the deadline can be missed without any serious penalty.

A typical example is while playing Video; some frames can be dropped without any serious loss of quality just to maintain synchronization. Many tasks involving input devices, multimedia fall into this category.

Hard Realtime

Hard Realtime tasks are those in which missing the deadline creates a serious penalty and in most cases total system failure.

A typical example of this is a realtime process that fires the propulsion systems in sequence in a rocket. If one of the rockets is fired slightly after the deadline, the system ends in disaster.

Another example would be the task of inflating the airbag inside a car during an accident. If that task were to miss its deadline, it would result in serious injury to the passengers.

Industrial controllers are required to work with hard realtime guarantees. Hard Realtime tasks cannot miss the deadlines. The deadlines are in terms of milliseconds.

Realtime Support in Linux

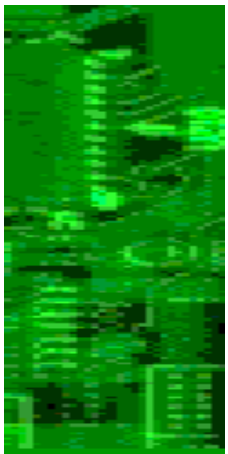
Latency

Definition

In all modern Operating Systems, including Linux, there is a significant and measurable delay between the time a task is scheduled for running and the actual start time of the task. This is normally termed as latency. Since it is associated with task execution, it is more appropriately referred to as “Task Latency”.

Task Release Jitter

Due to overheads in processing by the operating system, this latency is unavoidable. This is dependent on another metric termed as “Task Release Jitter” which is the time taken by the scheduling program to select and run a task in Realtime based on some arbitrary scheduling policy.



Interrupt Latency

Even in the case of Interrupt handling, there is a sufficient delay between the time, the interrupt actually occurred and the start time for processing the interrupt. This is termed as “Interrupt Latency”.

If these latencies are kept minimal, then it would be possible to meet deadlines at that level of resolution.

Linux

Linux is primarily meant to be an Application Operating System. Hence, the system is optimized on running a heavy load of tasks rather than a few tasks within a specified range of latency.

The actual maximal task latencies supported by Linux run into milliseconds. In typical x86 based systems running at 600MHz clock speed, the latencies run into hundreds of milliseconds for tasks. Interrupt latency is close to the millisecond range. There are studies showing actual task latency of up to 1000 milliseconds.

The typical responsiveness of a Linux machine is usually in tens of milliseconds as defined by an internal parameter.

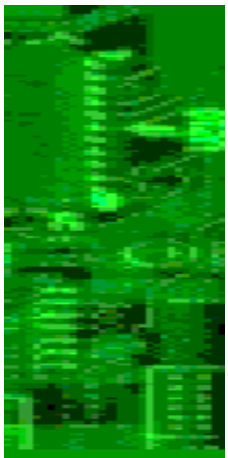
RTLinux

RTLinux is a Realtime Add-on to Linux invented by FSMLabs (www.fsmlabs.com) which provides Realtime capability to Linux based on the US Patent 5,995,745. It is available under the RTLinux Pro licensing scheme and the OpenRTLinux (GPL) Licensing scheme. (The RTLinux Demo platform from Innvo is provided under the OpenRTLinux (GPL) Licensing scheme).

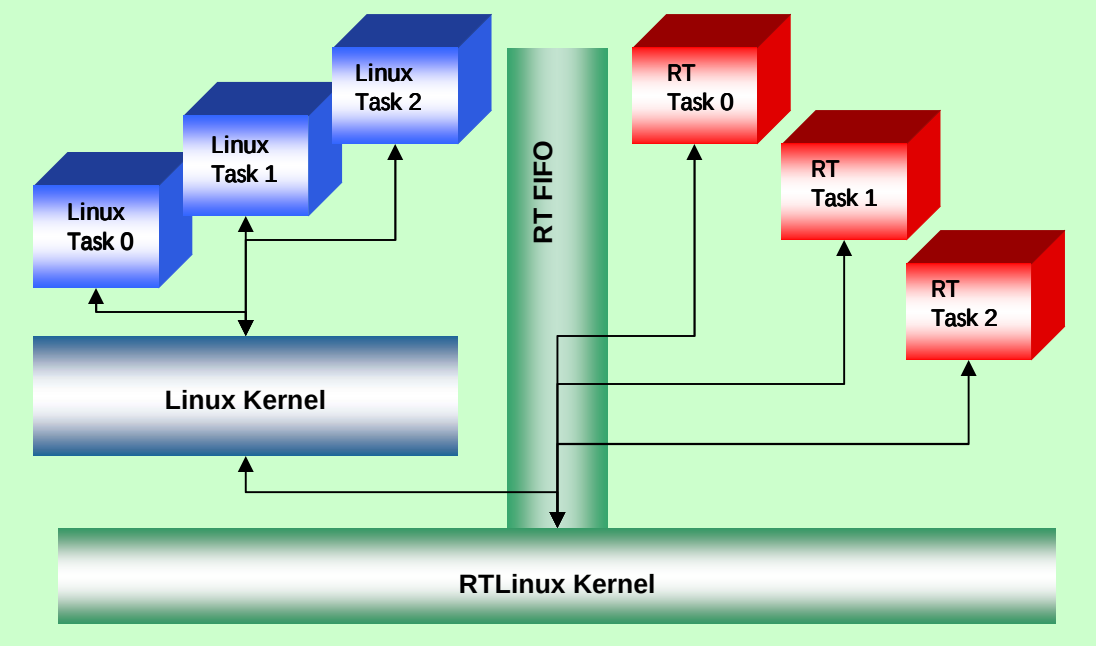
Advantages of RTLinux

- **Realtime Guarantees**
  It is notable that RTLinux provides the best Realtime guarantees (in terms of latency) among all available Linux Realtime variants. It has been observed to provide less than 10 microsecond worst case latencies on 800 MHz Intel Pentium III based systems.

- **Nanosecond Resolution**
  The fundamental principal of RTLinux is to provide a mechanism through the Linux kernel which allows tasks to work with latencies of the order of microseconds. It also allows the tasks to define their period in nanosecond resolution.



- **Make use of Linux Kernel Resources**
  Furthermore it provides a framework to run the tasks with hard Realtime guarantees and also make use of the resources provided by the Linux kernel. RTLinux is implemented by running the entire Linux kernel and its subtasks as a low priority thread under a Realtime microkernel called RTLinux.



Critical and Application Tasks

In the above figure, time critical tasks are colored in Red, while application tasks which are not deadline sensitive are indicated in Blue.

The Linux kernel itself runs as a low priority task of the RT kernel underneath. By doing so, it is possible to provide hard realtime guarantees to a select group of tasks (termed as Realtime tasks).

RTFIFO

Definition

The RTFIFO is an interface used to pass information between Realtime and Non-Realtime tasks. In most cases, the RT Task acts as the provider and the Non-RT tasks act as the receivers. The converse is also possible, though it is used in few cases.

Guarantee

This guarantees worst case latencies of just 15 microseconds on a 600 MHz Pentium III machine. Worst case latencies of up to 35 microseconds are achievable on a StrongARM platform running at 206MHz.

Proving Realtime Performance under OpenRTLinux v3.0

Tests

There are simple tests that can be conducted to observe the Realtime performance on a system. A task can be launched to execute periodically with the help of a timer. We can use a one-shot timer or a periodic timer for this depending on its availability in hardware.

The period of this task can be of the order of milliseconds. Each time the task is run, it stores the expected time and the actual time using a FIFO. This data can be picked up by an application and presented.

The difference in the actual time and expected time is the task execution latency. These timings can be obtained with nanosecond resolution depending on the clock available to the processor.

There are a series of tests labeled regression tests that are available with the standard OpenRTLinux v3.0 package. These tests are designed to test all the features of the RTLinux subsystem and its performance. This includes testing the FIFO interface designed to pass information between Realtime and Non-Realtime tasks.

The number of tasks executing on the processor is usually incremented through the test and several applications are made to execute simultaneously. This increases CPU usage and adds extensive overhead.

The Latencies measured under this condition give us practical values for “worst case latencies” when the system is under heavy stress. The same test is also run with only an average number of tasks running, which gives the “worst case latencies” under average load.

The difference in these figures gives us an understanding of the effect of CPU loading on Realtime tasks.

Innvo RTLinux

The RTLinux Demo platform from Innvo provides two test programs – *loadtest* and *burntest* which test Realtime performance under average load and heavy load respectively.

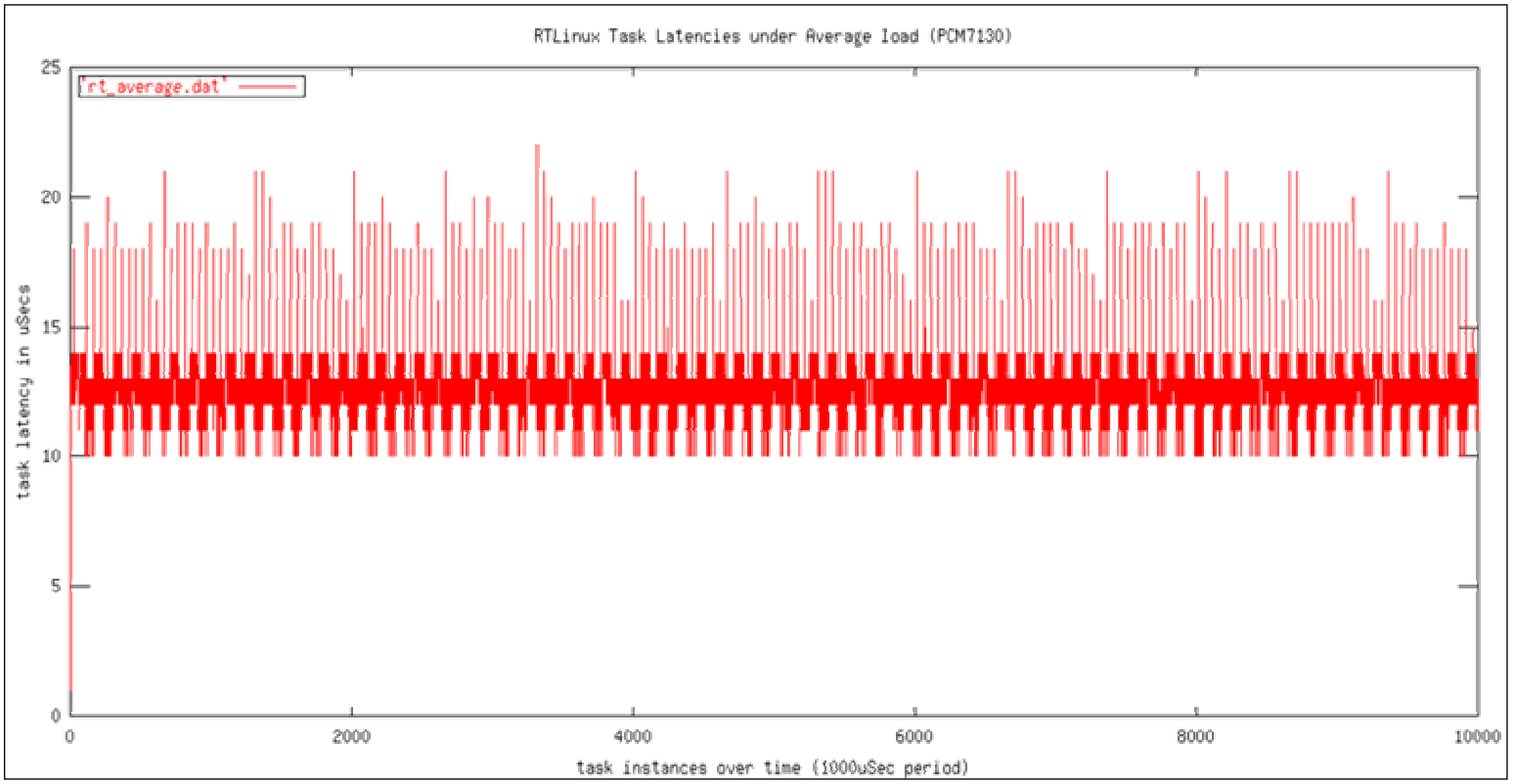
Burntest

The *burntest* program is to be run only for a short duration as it gradually loads the processor with hundreds of tasks, slowly exceeding the maximum process limit under linux (65,535).

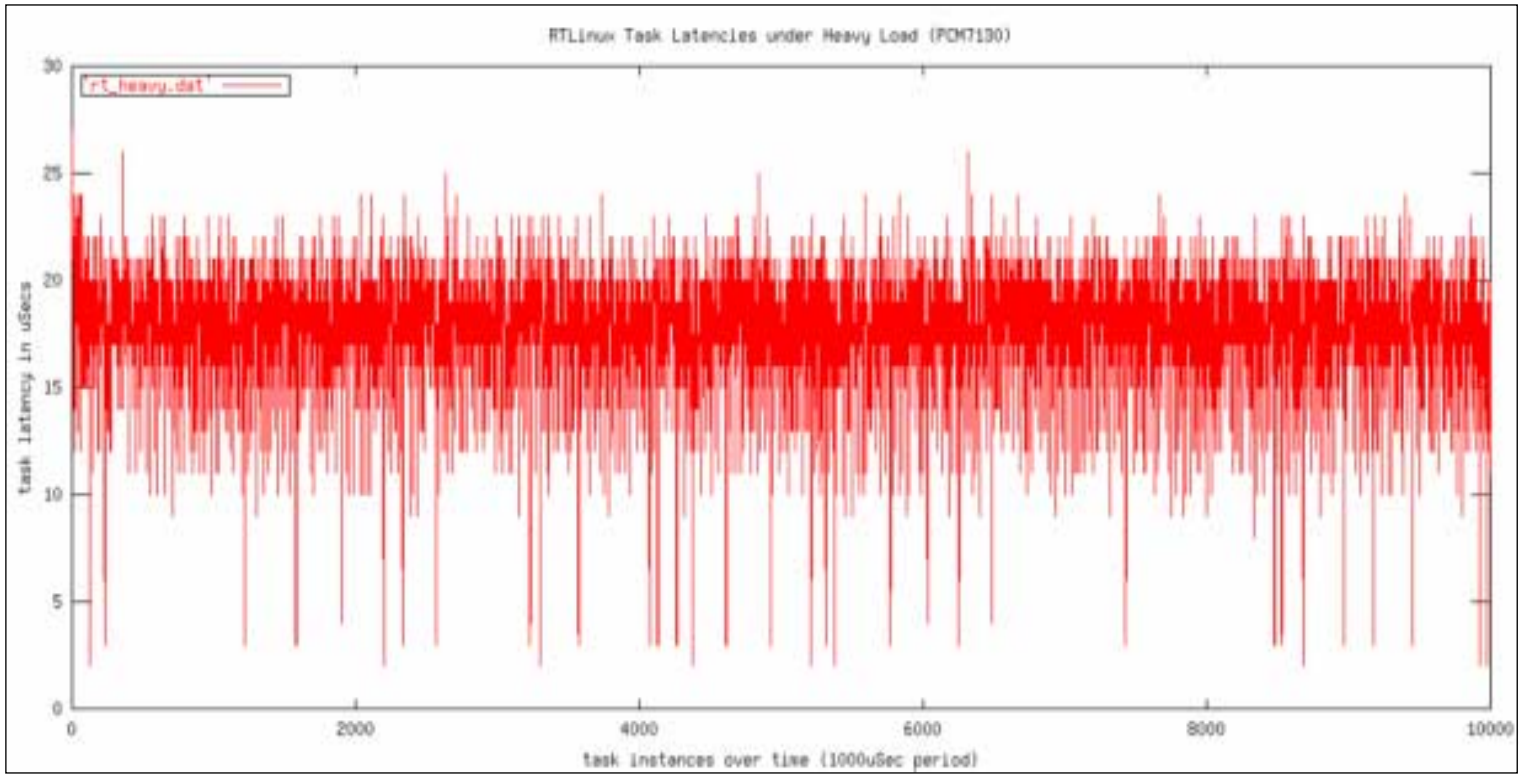
This creates heavy scheduling overhead and hence affects system performance. Realtime tasks under such strenuous conditions are still found to run with maximum "**worst case task latency**" of **31 microseconds**.

Test Results

The graphs in the next two pages shows the task latencies under average load and heavy load respectively.



Task latencies under average load



Task latencies under heavy load

Note: These tests were done on Linux v2.4.18-rmk7 running on Advantech PCM7130 SA-1110 Platform with OpenRTLinux v3.0 using loadtest and burntest scripts demonstrating the one-shot timer test. The test period for average load was 1 hour and the test period for heavy load was 15 minutes.

It can be easily seen that the maximum latencies lie between 10 and 20 microseconds under average load. They shift to between 15 and 25 microseconds on adding heavy load to the processor.

However the maximum latency stays within 30 microseconds in this sample. This demonstrates the guarantees on task latencies provided by RTLinux.

Innvo Systems Pte Ltd

Innvo Systems is a provider of turnkey Embedded Software Solutions to ODMs and OEMs. Our unparalleled expertise in Mobile Java, Wireless technologies and Real-time Operating Systems enable manufacturers to design and market highly integrated, cost effective devices with reduced time-to-market.

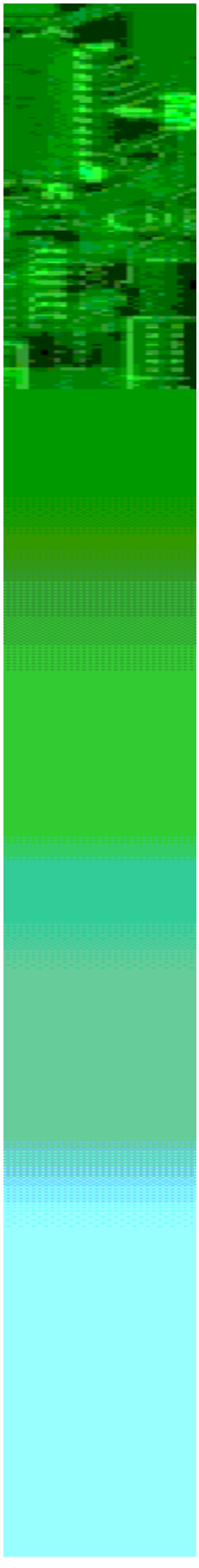
With key enabling technologies, Innvo takes our customers to the forefront of the Next-Gen Mobile Phone, Telematics and Industrial Automation arena.

Markets

Next-Gen Mobile Phones

Next-Gen Mobile Phone makers are looking for cost-effective customer growth, market differentiation and brand loyalty. In order to stay in the running, mobile phone will have integrated video, high quality audio, multimedia messaging and intuitive graphically interface, to name a few.

All these requires a broad range of skills in areas such as Real-time OSs and communication protocol stacks, as well as, the ability to integrate various components, and work with 3rd party hardware partners. Innvo enables our customers with optimized JVM, MMI,



real-time OS porting and support, and expertise in integration of all components.

With reduced time-to-market, high quality multi-services and efficient upgrade service, Innvo will take Next Generation Mobile Phones to higher ground.

Telematics

Telematics is making its way to taking the automotive industry by storm. The enticing benefits for the automobile customer, and immense revenue opportunities for automobile manufacturers, are driving the telematics market onto the fast track.

In-vehicle solutions providers and manufacturers need flexible technology and upgradeable platforms to ensure their highly customizable telematics will meet customers' extreme standards in entertainment, safety and security.

Innvo has developed the know-hows and technology needed to meet such demands while reducing your development time. Our patent-pending middleware technology and upgrade manager is set to create new and continuous revenue streams in harmony with customers profile and pattern.

Industrial Automation

The $150 billon automation market is demanding more innovative and smarter technologies. Innvo provides cost-effective industrial automation and control solutions to system integrators, automation specialists and equipment vendors who demand the utmost in product reliability, integration and ease-of-use.

Innvo's experience in real-time OSs, device river writing, creation of custom board support packages, give Innvo the required expertise to properly evaluate and execute the complex software systems required in industrial automation.

Business

Professional Services

Innvo's Professional Services Team provides RTOS porting service, Device Driver Development and Embedded Software consultancy. With our profound knowledge of real time operating systems such as Nucleus, uITRON, Rex OS, AMX, eCOS and together with expertise in application operating systems such as Linux and Symbian OS, Innvo offers its customers with turnkey product solutions.

Our OS skills coupled with our expertise in Mobile Java expertise (J2ME) we create an unmatched value proposition to our customers who are developing next generation mobile terminals and telematics devices.

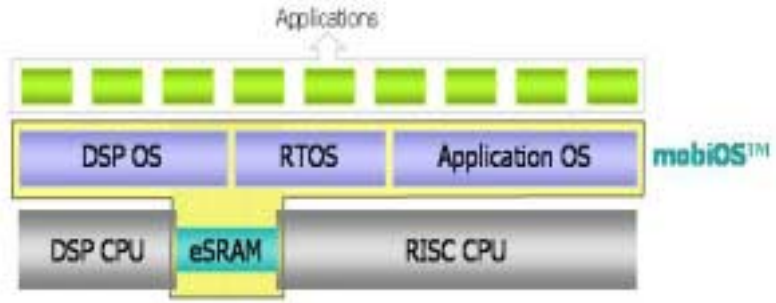
Products

Our core products include - mobiOS™, mobiOS™ MPEG4 Engine, Innvo MMI

mobiOS™

Innvo mobiOS™ is a patent pending OS technology that allows seamless integration of two operating systems – A hard realtime operating system and an application operating system on a single RISC CPU.

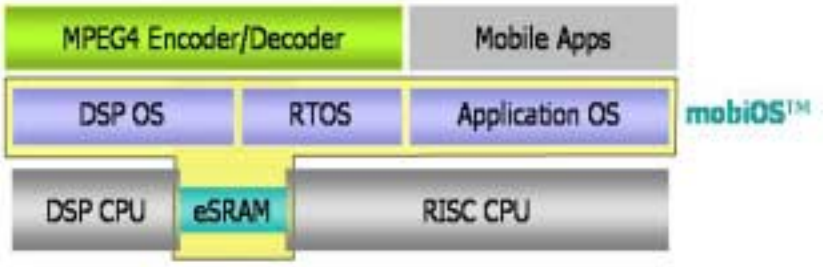
mobiOS™ is designed to maximize the utilization of "embedded SRAM" within an asymmetric dual core CPUs and to provide a seamless application environment for such CPUs. Hence our mobiOS™ technology provides direct cost and power savings to the chip manufacturer and enables our customers to drive high performance multimedia applications using lower cost CPUs.



Innvo today is being targeted at integrated baseband CPUs, where mobiOS™ technology with simultaneous execution of a hard realtime operating system that runs the protocol stack together with an Application OS that executes mobile applications.

mobiOS™ MPEG4 Engine

Built on the proven mobiOS™ technology, Innvo's MPEG4 engine provides a deterministic MPEG4 encoding/decoding performance for the Application Processor in the next generation 3G handset.



Since the entire MPEG4 encoding and decoding is done in real time, using an RTOS and DSP, the performance is be guaranteed. The Dual OS would enable existing Mobile Applications to seamlessly execute on the Application OS such as Symbian.

Innvo MMI

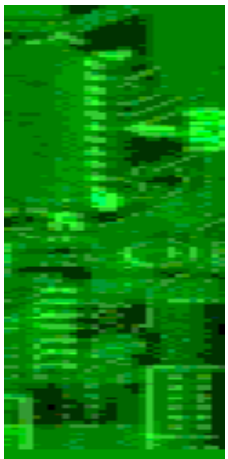
Innvo MMI is a highly customizable MMI engine that provides a "palmtop" type of user interface for low and mid range phones. As these phones get better color displays and Java capabilities, traditional text based MMI lose their relevance. Innvo MMI interfaces with existing phone applications and RTOS environments with minimal effort, and empowers operators, tier 2 & 3 handset manufactures to create stronger brand identity and product differentiation.

Engagement and Support Model

Innvo typically engages its customers very early during their product development cycle. Using our technology expertise and IPs we influence some of the key technology decisions made by our customers.

Innvo support model is highly focused on customer's requirements and is tailored for their individual needs. Since we have a customer base with varied skillsets, we support our customers right from

implementing certain specific technologies to assisting them in developing turnkey products. Our IPs provide core technology for our customers' products and hence we give a very high importance to software testing and reliability for all technologies that we deliver.



Customers pay Innvo an upfront NRE for integration of our technologies and professional services. We are also paid a royalty for our IPs that are shipped together with the customers' products. Support packages and fees are tailored specifically for customers' requirements.

Background

History

Established in year 2000, Innvo is a spin-off from the National University of Singapore. Innvo closed its first round of investment in November 2001 and is presently very actively involved in engaging more customers, developing its core technologies and creating market visibility. Innvo has its headquarters in Singapore and operates with representatives in Japan and Australia. Innvo has business dealings in US, Japan, Taiwan, Korea, China and India.

Founders

Rajiv M Ranganath (COO), Abhaya Shenoy (Product Manager), Srijon Biswas (Product Manager), Prof. A L Ananda (Advisor)

Funders and Investors

Innvo is supported by the following Angels and institutional investors – Mr. Chay Kwong Soon (Director and Chairman of the Board), Mr. Lee Keen Whye (Director), Mr. Guruprakash S K (Director and CEO), TIF Ventures Pte Ltd (A wholly owned subsidiary of the Economic Development Board of Singapore, EDB.)