JNIB Java Network Interface Box

JNIB-102

Overview

JNIB-102 is the second generation of Java™ language programmable compact, low cost, Internet enabled automation controller designed for real-time monitoring, data acquisition, and control applications. JNIB-102 is ideally suited to provide network access to low level sensors and actuators that require real-time response. Typical applications of the JNIB-102 include home, school, office, and building automation, utility energy control and management, traffic monitoring, surveillance, and security systems.

The JNIB-102 JME platform is built upon aJile's high performance direct execution SOC, aJ-102, and Sun Microsystems' JME/CLDC and CDC/Foundation Profile libraries. The powerful combination of direct JVM bytecode execution, integrated multithreading support, patented multiple JVM environments, and TCP/IP networking protocol provide an efficient, safe, and reliable Java platform for networking realtime automation applications. As a platform completely programmed in the Java language, it ensures all applications including device drivers can fully leverage Java's proven benefits of lower development and maintenance costs, portability, scalability, and security. Applications can be dynamically downloaded or updated over the Internet to enable life-cycle management remotely and securely.

JNIB-102 system consists of 32B SDRAM, 32MB NAND flash, and a variety of standard interfaces for devices including 10/100 Ethernet port, dual serial ports (RS232), USB 2.0 host, SD slot, and 8-bit GPIO port. The 8-bit GPIO port is fully software programmable to configure discrete signals as either inputs or outputs.

Figure 1: JNIB-102 Top Side

Applications

The ubiquity of broadband and wireless networks require a new generation of Internet edge devices that enable operators and service providers to monitor, control, and manage systems on demand. The JNIB-102 provides an intelligent, easy-to-use Java platform to interface non-networked devices with LAN, WLAN, or cellular networks. The JNIB-102 Java platform can also be combined with third party middlewares that support industry standard APIs (e.g. OSGi) to simplify the porting and maintenance of applications that are traditionally run on larger, more expensive systems. JNIB-102 is an ideal network edge device for applications such as:

- Wireless M2M remote control, sensing, and telemetry for health care, security systems, utility metering, home/building automation, and transportation monitoring
- RFID systems for asset tracking, inventory control, and logistics operations
- Wireless sensor networks (e.g Zigbee) and location systems (e.g GPS)

Features

Processor
- aJile network direct execution SOC "aJ-102"

Memory configuration
- 32 MB SDRAM
- 32 MB NAND
- Flash

Standard I/O interfaces
- 10/100 base-T Ethernet port
- Dual serial ports (RS-232)
- USB 2.0 host port
- Secured digital card slot (SD/SDIO)
- 8-bit GPIO port

Real-time clock with battery backup

UMTS/HSDPA connector
- 50-pin slimstack receptacle
- Cinterion HC25/28 module

Zigbee module connector
- 20-pin header
- XBee/XBee-PRO modules

Real-time clock with a battery backup

Status LEDs
- Serial port
- Ethernet port
- SD slot

- Zigbee port
- UMTS/HSPDA
- Power

Audible buzzer

Power supply
- 100-240V 5V/3A DC external adapter
- Optional rechargeable Li-Ion battery pack @ 3.7V 1800 mAh

JTAG header
- JTAG debug interface
- Srerial debug channel (UART3)

5VDC/2A Power input

Compact size
- 130mm (w) x 135mm (d) x 32mm (h)

Agency listings
- UL 60950-1
- CSA IC22.2 No. 60950-1
- CE EN 60950-1
- FCC part 15 Class B

Environment
- Operating temperature range
  - $0\ ^{o}$C to $50\ ^{o}$ C (32° F to 122° F)
- Storage temperature range
  - $0\ ^{o}$ C to 70° C (32° F to 158° F)

System development support

The JNIB-102 bundled with the aJile RTOS, an optimizing application builder (JEMBuilder), debugging tools and an evaluation board provides a complete silicon-based solution for the JME platform. The key components are:

aJile RTOS

The aJile RTOS is implemented entirely in Java as illustrated in the figure 2. In addition, the aJile Multiple JVM (MJM) enables multiple applications to execute concurrently and independently in a deterministic, timesliced schedule. This allows hard real-time applications to run independently and safely exist with networked applications.
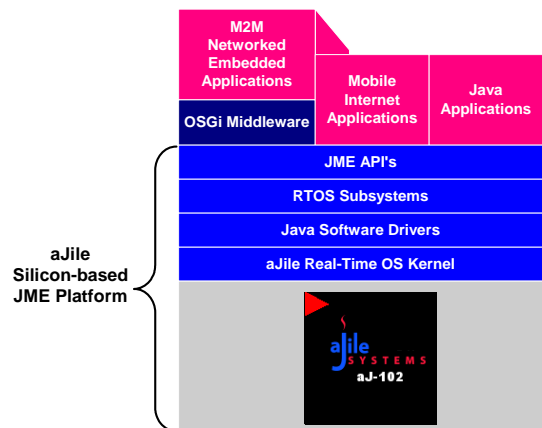
Figure 2.   The Silicon-based JME Platform

The primary components of the aJile real-time operating system are outlined as follows:

- **API**
  - CLDC 1.1
  - CDC 1.1/Foundation Profile (FP)
- **Network Stack, File System, and Security Framework**
  - TCP/IPJME runtime libraries and aJile's Java implementation of JNI, graphics primitives
  - TCP/IP network stack including PPP, DHCP, DNS, SNMP services
  - FAT 32 file system for USB and SD memory cards
  - USB 2.0 host/slave stack
  - Security frame work
  - Bootloader for emote application updates
- **Java Software Drivers**
  - All integrated I/O's
  - USB serial drivers for memory sticks, keyboards, and mice
  - SD memory card and WLAN card
- **aJile Real-time OS Kernel**

  aJile processors include an internal microprogrammed real-time kernel. It performs the traditional operating system functions (scheduling, context switching, interrupt preprocessing, error preprocessing, and object synchronization). Java threads are native threads on the aJile processor, and extended bytecode instructions are used to implement these Java threading primitives (sleep, wait, notify, notifyall, yield, monitor enter, monitor exit, and interrupt) in order to provide extremely fast and atomic (non-interruptible) executive operations. The on-chip real-time thread manager performs priority-based preemptive scheduling with extremely fast context switch times of less than 1µs. In addition, aJile Multiple JVM (MJM) technology enables multiple applications to execute concurrently and independently in a deterministic, timesliced scheduling. Each JVM employs its' own threading and memory policies to enable real-time applications to execute concurrently with networked applications without the threat of garbage collection (G.C) pauses and other interruptions. The MJM capability takes the Java "sandbox" security model to the next level, providing a mechanism to easily isolate applications and allocated resources. aJile RTOS enables hard real-time applications to run independently and safely co-exist with networked applications.

Development Tools

The development environment allows the use of any off-the-shelf IDE that produces Java standard class files such as Eclipse or Netbeans. It consists of the following key components:

- **Optimizing Linker/Application Builder (JEMBuilder)**

- **Application Debugging Tools**

- **Evaluation Kit (aJ-102EK)**
  - JNIB-102
  - JTAG cable
  - JTAG-to-USB converter
  - USB cable
  - AC power adapter
  - Schematics, and gerber file can be downloaded via aJile website: **www.ajile.com**

aJile
SYSTEMS