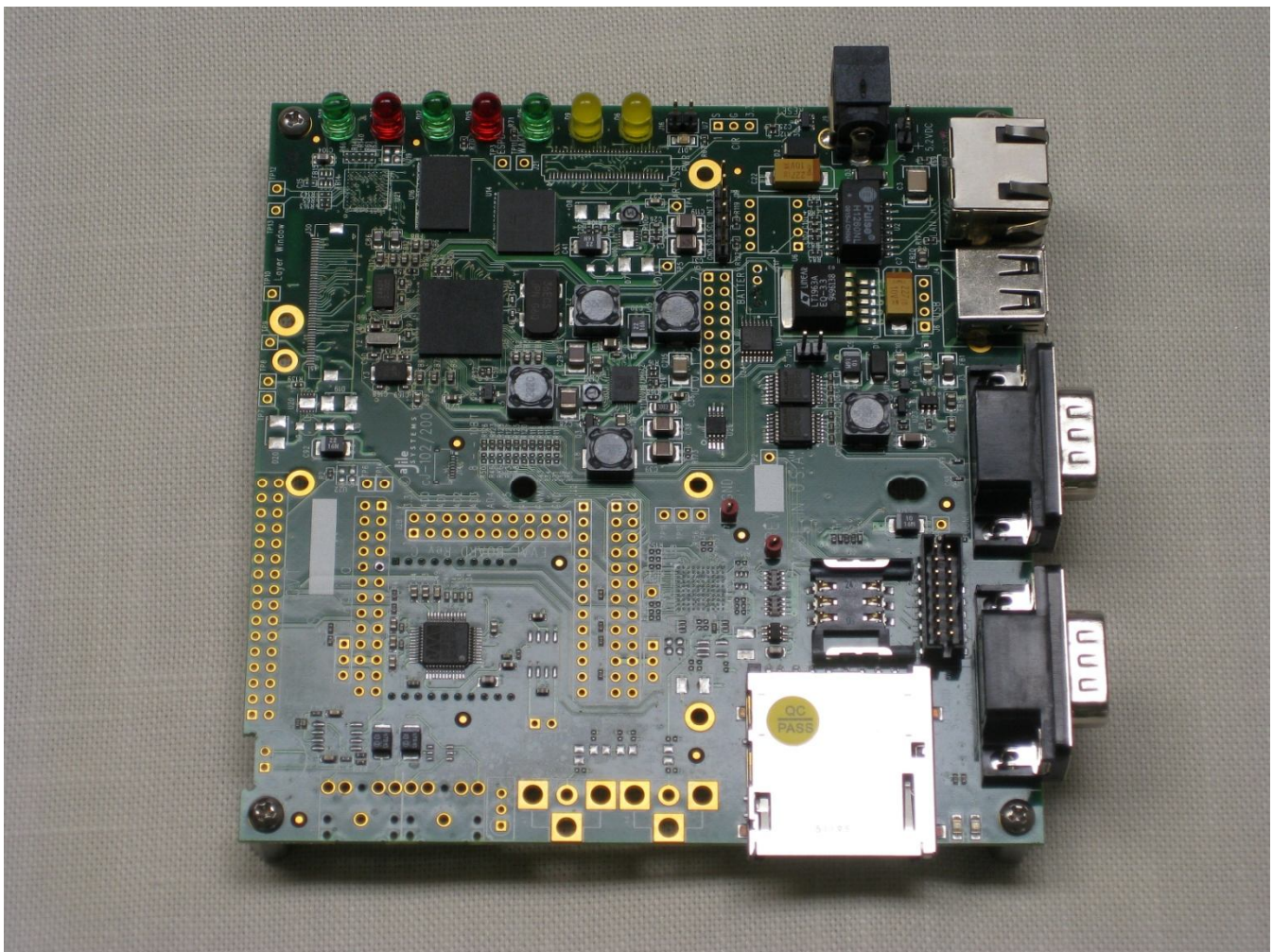


Overview

The aJile aJ-102EK is a compact and versatile evaluation kit for the aJ-102 SOC that directly executes both Java bytecode instructions, and real-time Java threading primitives. The aJ-102EK is designed with a typical I/O configuration required for a M2M network edge device to perform the real-time monitoring, control, and data acquisition of smart sensors over Internet.

The aJ-102EK bundled with the aJile RTOS, an optimizing application builder (JEMBuilder), and a debugging tool provide a complete silicon-based solution for the JME platform. Using commercial Java IDEs, application developers can create applications written entirely in Java language with the performance and memory efficiency of system programmed in C and assembly. The aJ-102EK allows users to evaluate, prototype, and create aJ-102-based M2M network edge devices for home, building automation, utility control and management, traffic monitoring, surveillance, and security systems. The block diagram of the aJ-102EK is in Figure 1. The picture is shown below:



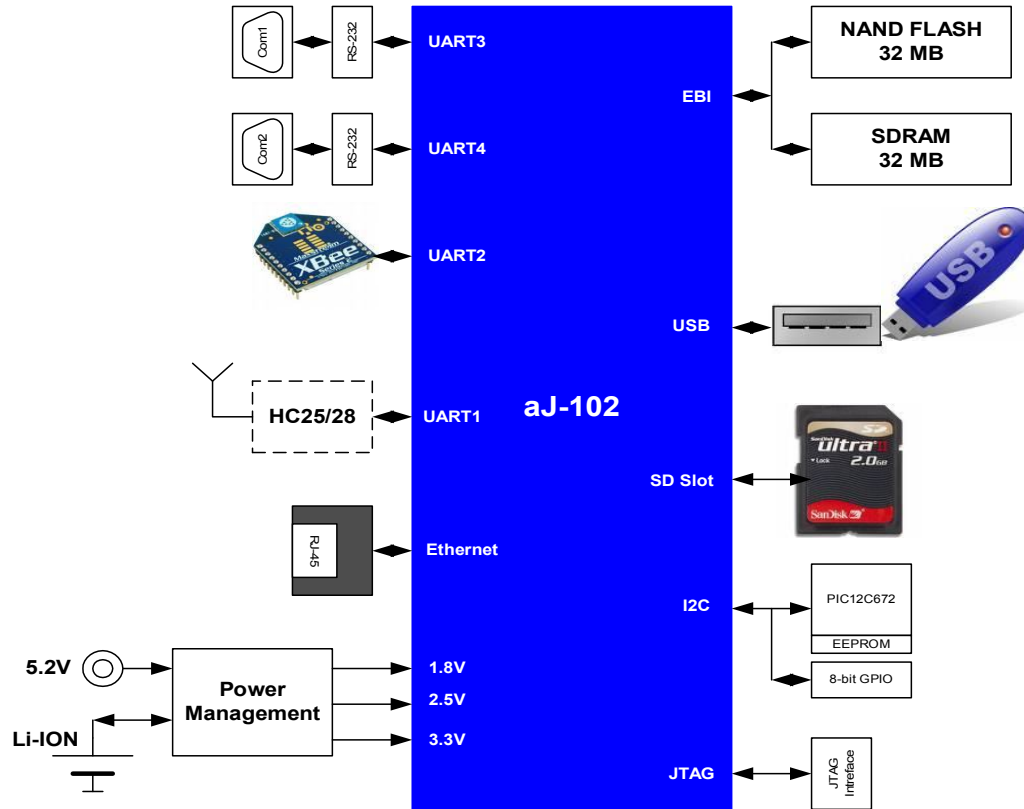


Figure 1. Block Diagram of aJ-102EK

Features

Processor

- aJile network direct execution Java processor, aJ-102

Memory configuration

- 32 MB SDRAM
- 32 MB NAND Flash

10/100 base-T Ethernet port

- RJ-45 connector

Dual serial channels

- RS-232
- DB9M connector

USB master port

- USB type A receptacle

Secure digital card slot

- SD/SDIO
- Memory card
- WLAN 802.11b/g

UMTS/HSDPA connector

- 50-pin slimstack receptacle
- Optional Cinterion HC25/28 module

Zigbee module connector

- 20-pin header
- Optional XBee/XBee-PRO modules

Real-time clock with a battery backup

8-bit GPIO port (J15 header)

Status LEDs

- Ethernet port
- USB port
- SD slot
- Zigbee
- UMTS/HSPDA
- Power on

Power supply

- 100-240V 5V/3A DC external adapter
- Rechargeable Li-Ion battery pack @ 3.7V 1800 mAh

JTAG header

- JTAG debug interface
- Debug serial channel (UART3)

System Development Support

The aJ-102EK 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.

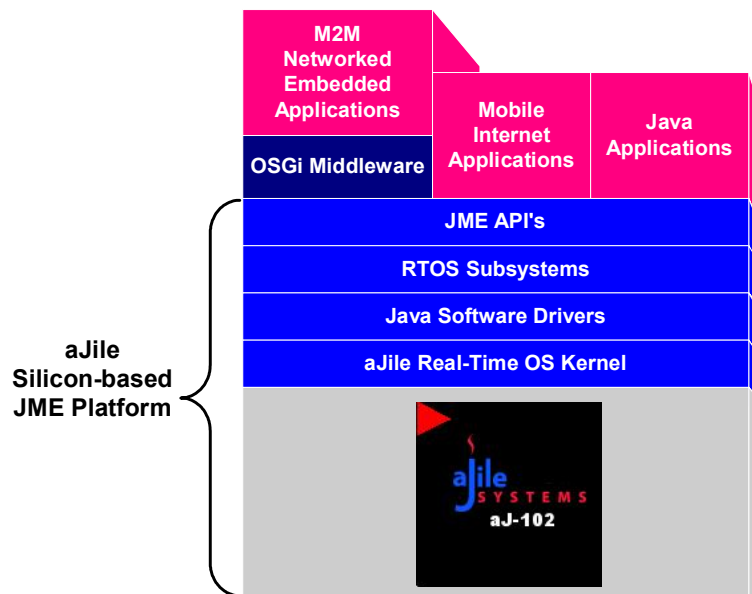


Figure 2. The Silicon-based JME Platform

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

- **JME APIs**
 - CLDC1.1
 - CDC1.1/Foundation Profile (FP)
- **Network Stack, File System, and Security Framework**
 - JME 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 remote application updates
- **Java Software Drivers**
 - All integrated peripherals
 - 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 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 1 μ s. 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)**
 - aJ-102EK board
 - JTAG cable
 - JTAG-to-USB converter
 - USB cable
 - AC power adapter
 - Schematics, and gerber file can be downloaded via aJile website www.ajile.com