

Embedded Device Server White Paper

English ver 1.2

February 5, 2007



Overview

In the field of device networking, device servers have been playing a huge role in providing network connectivity for serial devices. A **device server** connects individual serial device to the network, working as a serial to LAN converter. It is composed of at least one LAN port and one or more serial ports, so that serial devices can be connected to the device server using their serial ports, and eventually can be connected to LAN with a single LAN cable. Devices attached to the device server operate as if they were connected to LAN directly, yet using serial communication internally. With the help of the device server, devices logically and indirectly, rather than physically and directly, are linked to the target machine through LAN connection. Consequently, more effective and dynamic tasks can be performed.

There are many kinds of device servers available in the market, and they are slightly different in features including extended temperature, serial port isolation, power requirement, etc. These could be considered rather minor advancements, compared to the recent major trends in device servers: wireless and embedded. With wireless device servers, now serial devices can transfer data over the wireless network. Mobility can be significantly increased with wireless products. With **embedded device server**, device manufacturers can network-enable their products in a built-in style. In a customized and an easier way, embedded device servers internally serve as a device server, without an additional external device server.

This white paper focuses on the background of embedded device server, key factors of selecting an embedded device server, SystemBase embedded device server Eddy and applications of it.

Embedded Device Server

Nowadays, vast majority of electronic devices require data transfer over the network. They send, receive, and share data with embedded network interface. In order to implement network interface inside these devices, designers had to build the feature from scratch. This often led to delay in time-to-market, and engineers occasionally fell into trouble with hardware and software design issues. Embedded device servers provide device designers / engineers with an easy network-enabling solution, helping them concentrate on other design aspects. With embedded device servers, network interface can be simply added to the target device in a reliable, fast, and cost-effective manner.

Then, what are key factors in choosing the best embedded device server?

Integration – How easy it is to integrate the embedded device server with user’s hardware and software environment? How can the device server be connected? By what kind of pins and connectors? How are the serial and LAN interface outputs provided? Are software / hardware manuals and documents thorough and fully-described?

Customization – How much can the embedded device server be modified to reflect users’ customized environment? SDK (Software Development Kit) / API (Application Programming Interface) can be helpful for programmers.

Size – Since the embedded device server is not a stand-alone device and therefore should be embedded in another hardware case, size can be a major concern. Also, the shape and connection options should not bother users’ hardware design.

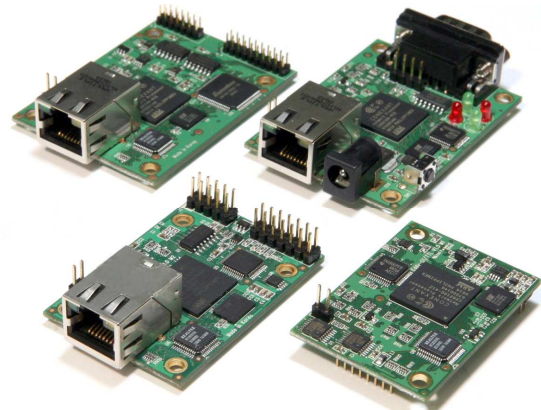
Price – Since major customers for the embedded device server is device server distributors and industrial device manufacturers, not end users, price does matter and volume pricing can be effective.

SystemBase Eddy

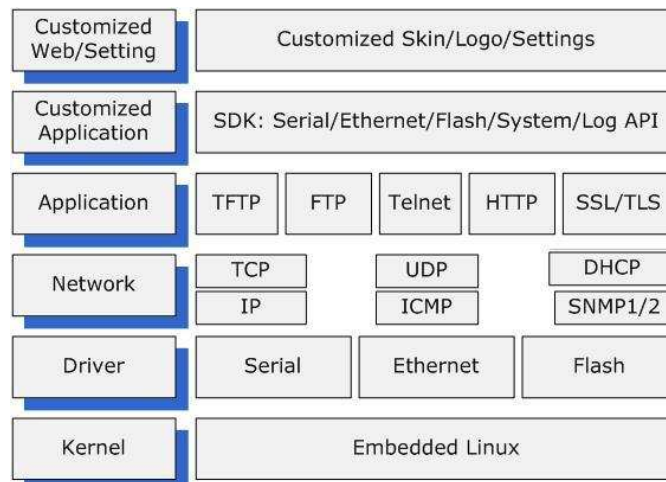
Eddy is a programmable embedded device server designed by SystemBase. It has been designed to maximize users' integration and customization needs.

- **Run Your Own Application on the Device**

Eddy is distinguished with other embedded device servers in that it can upload customized user applications and execute them. With such feature, a user can upload any socket / serial communication application that can run on standard Linux environment with no or little modification. This openness gives



users a chance to apply a wide variety of operations on the device server, with considerably less limitations. To help programmers work on their own application to be run on the device server, Eddy supports **SDK (Software Development Kit)** and cross-compile environment. With SDK and ready-to-run example codes, programmers can easily build up their own applications under the standard Linux environment. Cross compiler running on Linux will help the application run on Eddy with ease.



Eddy Software Architecture

- **High-End Level Specifications**

Tired of constraints with your embedded device server's 8-bit CPU and 256KB memory? Eddy completely solves this problem by adopting a 32-bit CPU with 168MHz clock, 4MB of Flash memory, and 8MB of SDRAM. Your applications can be large in size and will run faster, in a more stable way under the embedded Linux operating system. And all these high-end hardware is embedded on a 55mm * 38mm board!

- **Device Connectivity Management**

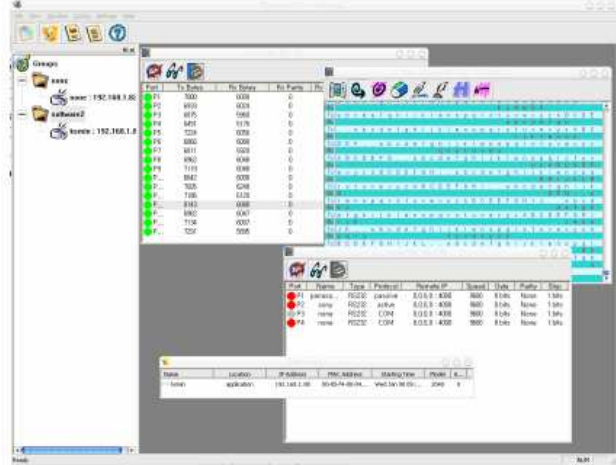
As a qualified device server, Eddy receives RS232/422/485 signals from connected devices and transmits them to the network, and vice versa. All three RS interfaces (RS232/422/485) are supported with Eddy, and can be selected by pins on the board. Serial communication speed can be 921.6Kbps maximum, providing fast as well as stable data transmission. Various network protocols are supported, including TCP, UDP, Telnet, ICMP, DHCP, TFTP, HTTP, and PPP to ensure more flexibility. SNMP (Simple Network Management Protocol) 1 and 2 are also supported, helping you manage devices with complete control with the worldwide management standard.

- **Configuration**

There are many factors that determine the device server's environment, and Eddy offers you easy but in-depth configuration selections. The settings include serial communication, network, security, management, and other advanced settings. You can configure settings through web or telnet. When configuring by web, you can just use your web browser to access your Eddy and easy web interface will guide you through options. On the other hand, when configuring by telnet, you can connect to Eddy by telnet and use preset commands to apply changes.

● **PortView**

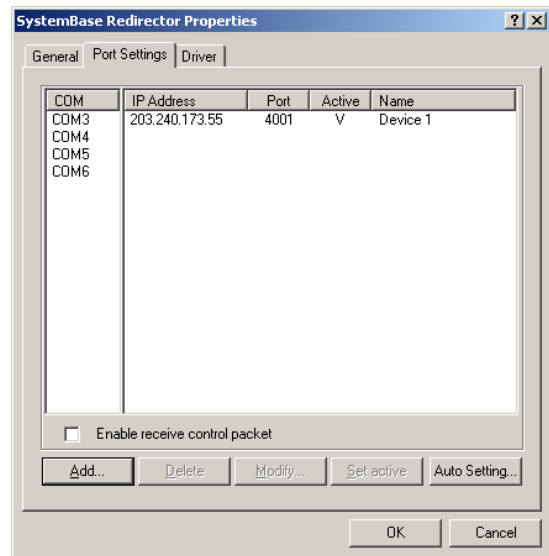
PortView is a real-time communication status-monitoring program for SystemBase device servers. From the PC using Windows operating system, communication status and all input/output data through each serial port can be monitored from the remote site. When the connected devices suffer from any problematic situation,



remote diagnosis and solution can be performed. Serial data can also be saved as a file, providing an effective debugging data for later use.

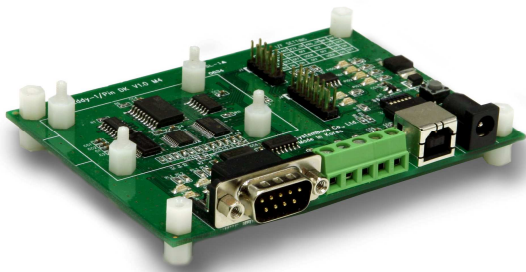
● **COM Port Redirector**

COM Port Redirector is an automatic serial/LAN conversion driver installed on a PC. When installed, a user can use the serial port of Eddy that is connected through network as if it were a COM port on the user PC. This saves a lot of extra work that might have been necessary: all previous serial communication programs can be used without any modification, such as conversion to socket program.

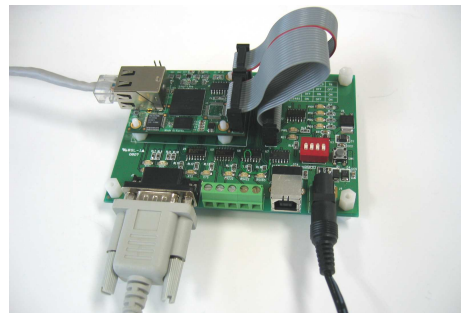


Eddy DK: the Development Kit

Eddy Development Kit helps developers test their own applications and evaluate Eddy modules easily. Evaluation board itself is a guideline for integrated hardware design with Eddy module mounted. With LED displays on Power, Ready, communication interface, GPIO, and serial line status, engineers can easily recognize the operation status of the module. Compile environment including the cross-compiler, sample code, documentation, and tools are included to accelerate integration and customization process.



Eddy-S1/Pin Evaluation Board



Eddy-S1/Pin Module Attached

Applications of Eddy

- **Factory / Industrial Automation**

PLC, Robot arms, Human-Machine Interface, Warehouse rails
 Medical instruments, Inspection equipment controllers
 Alarming units

- **Home Appliances / Electronic Devices**

Power controller, Gaming machines
 Scales, Gas detection units, Water & pollution metering devices
 Data collection and distribution units

- **Financial / Building Automation**

Card readers, Barcode scanners, KIOSKs, Point-Of-Sale related devices
 Serial printers, Cash registers, Credit card authorization terminals

Biometric detection units, Security devices

- **OEM Device Server Distributors**

OEM device server with distributor's own case & brand

Ready-to-go device or customized application / setup mode can be inserted