

# The Uicom™ IP2022 Multi-Protocol Processor

DELIVERS LOW-COST CONNECTIVITY FOR THE INTERNET EDGE

White Paper

**In the next 10 years** it's expected that billions of devices will connect to the Internet. Within that realm lies the huge frontier of the Internet edge. In contrast to the Internet's core where servers and network routers do their heavy lifting, the Internet edge is where networks meet client devices.

IP 2022



635 Clyde Avenue

Mountain View, CA 94043

tel 650.210.1500

fax 650.210.8896

[www.ubicom.com](http://www.ubicom.com)



## THE UBICOM™ IP2022 MULTI-PROTOCOL PROCESSOR DELIVER LOW-COST CONNECTIVITY FOR THE INTERNET EDGE

In the next 10 years it's expected that billions of devices will connect to the Internet. Within that realm lies the huge frontier of the Internet edge. In contrast to the Internet's core where servers and network routers do their heavy lifting, the Internet edge is where networks meet client devices.

The more visible examples are Net-connected consumer products such as DSL modems, cable modems, Wi-Fi wireless network cards, home-network routers, cell phones, PDAs, and MP3 players. What's not so obvious is the wider scope of the Internet edge comprised of less-visible embedded devices. Security systems, LCD projectors, environmental controls in buildings, automotive telematics, and thermostats in commercial refrigerators are just a few examples where a wired or wireless link to the Internet can add great value.

Exactly how those devices will use Internet connectivity varies from application to application. One common thread, however, is the ability of a vendor or administrator to remotely check the status of the device. If a commercial washing machine malfunctions, an Internet link could enable the vendor to determine if the problem requires hands-on repair work or a remote firmware upgrade. By repairing the device over the Internet, the vendor could save the significant cost of sending out a technician.

Embedding Internet connectivity in everyday products requires very inexpensive technology. It must be inexpensive in the narrow sense of a low-cost bill of materials (BOM), and it must be inexpensive in the broader sense of being readily adaptable to many different kinds of applications without redundancy.

Adaptability is a particular challenge because of the large and growing array of wired and wireless protocols commonly found at the Internet edge. Among those protocols are HomePlug™, Bluetooth™, GPRS, and 802.11b on the wireless side, and USB, Ethernet, GPSI, and PCMCIA on the wired side. It's much too costly to craft a specialized chip for each protocol, and specialized chips can't adapt to protocols that are rapidly evolving. The other alternative – affordable general-purpose processors – often have trouble delivering the performance and low-latency requirements in this environment.

Ubicom has a better solution that meets all those challenges. The IP2022 multi-protocol processor is a low-cost, low-power solution that readily adapts to many different protocols and has plenty of performance for Internet-edge applications. It's an off-the-shelf chip that eliminates the need for a costly, time-consuming ASIC project. It's as programmable as a general-purpose processor, but it has an architecture that's carefully optimized for Internet-edge protocol processing. The IP2022 is a no-compromise solution that fits a wide variety of Internet-enabled applications.

### HOW THE IP2022 CUTS COSTS

There's no secret to reducing the cost of a microprocessor: simply make it smaller. The trick is making it smaller without sacrificing necessary performance and flexibility. Ubicom's IP2022 processor slashes the cost of embedding Internet connectivity in several ways:

- It has a special architecture optimized for Internet-edge applications, discarding the old baggage of general-purpose processors.
- It handles protocol processing in software instead of in hard-wired logic, reducing the complexity of the chip while making it more adaptable to evolving standards.
- It uses efficient memory-to-memory operations to process packets as they arrive, eliminating the need for large, costly on-chip caches and packet buffers.
- It works with a compact, finely tuned real-time operating system (RTOS) that delivers deterministic real-time response without the memory-hungry requirements and licensing fees of other RTOSes.
- It allows designers to use the same chip across a wide variety of Internet-edge products simply by changing the software, thereby reducing nonrecurring engineering (NRE) costs.



Compare Ubicom's IP2022 to other solutions. General-purpose processors carry the burden of old instruction sets that weren't designed with networking applications in mind. They also have large instruction and data caches that waste silicon and are nondeterministic for real-time processing. If pushed hard enough, these processors can usually handle the demands of packet processing, but usually at clock frequencies and power-consumption levels that make them unsuitable for low-cost embedded products.

Another solution is a system-on-chip (SoC) standard product that's targeted at Internet-edge applications. But they are either protocol-specific or support a narrow range of protocols, because they must include all the needed functional blocks on-chip. They're relatively large, costly, and inflexible.

Creating your own SoC – whether you're an SoC vendor or a device maker crafting a custom ASIC for your application – presents another set of problems. First, you have to license a microprocessor core from an IP (intellectual property) vendor, then go to another silicon IP vendor to get a PCI core, then shop somewhere else for an Ethernet MAC core, and so on. Meanwhile, your programmers have to license (or write from scratch) an operating system and a protocol stack – in addition to writing the application software that should be their primary responsibility, because that's what actually differentiates the product. Keep in mind that your competitors are licensing the same IP and there's little room for meaningful differentiation at the level of protocol stacks and device drivers. It's the application-level differentiation that matters.

Finally, you have to integrate all that IP together and hope it works the first time, or else bear the brunt of costly re-spins. Because the hardware IP and software IP you licensed were designed to be general purpose and were designed for portability instead of your specific application, the finished system isn't as well-optimized as it should be. In addition, the complexity and risk associated with designing an ASIC can add a year or more to your project and significantly delay your time to market.

Ubicom takes a different approach. Ubicom provides the whole solution as a fully integrated platform – the RTOS, the protocol stack, and the necessary hardware. Ubicom's IP2022 chip embeds some basic hardware, but it lets you combine it with on-chip software to support any of the most prevalent protocols. On the same device you can support Ethernet, HomePlug, HomePNA™, HomeRF™, Bluetooth wireless technology, IEEE-802.11, and so on.

The key to this approach is Software SoC™ technology, a Ubicom innovation. Software SoC implements Internet-edge protocols in software on a processor with a highly optimized architecture. Because the Ubicom IP2022 chip handles the I/O in software, you can inventory one chip and use it for multiple applications. That's an advantage not only in terms of volume cost savings for the chip, but also for the engineering teams designing products around the chip.

Most teams are building products that talk to multiple protocols. With the IP2022, they can support multiple protocols or make a change in the protocol with a simple firmware upgrade and/or a small board modification. They don't need an engineering team that's trained on Vendor A's architecture and another trained on Vendor B's architecture. Instead, they need only one engineering team and Ubicom's Software SoC technology to create all their products. That saves engineering time and training, and it allows people to easily move back and forth from one project to another.

Because almost everything needed for Internet-edge applications is integrated on the IP2022, projects that would normally take from 1 to 3 years (to spin a custom SoC and write the software) are possible in a mere 12 weeks.

When a protocol changes, designers can make the upgrade in software. This flexibility allows more differentiation, a major concern when competitors are all using the same licensed IP.



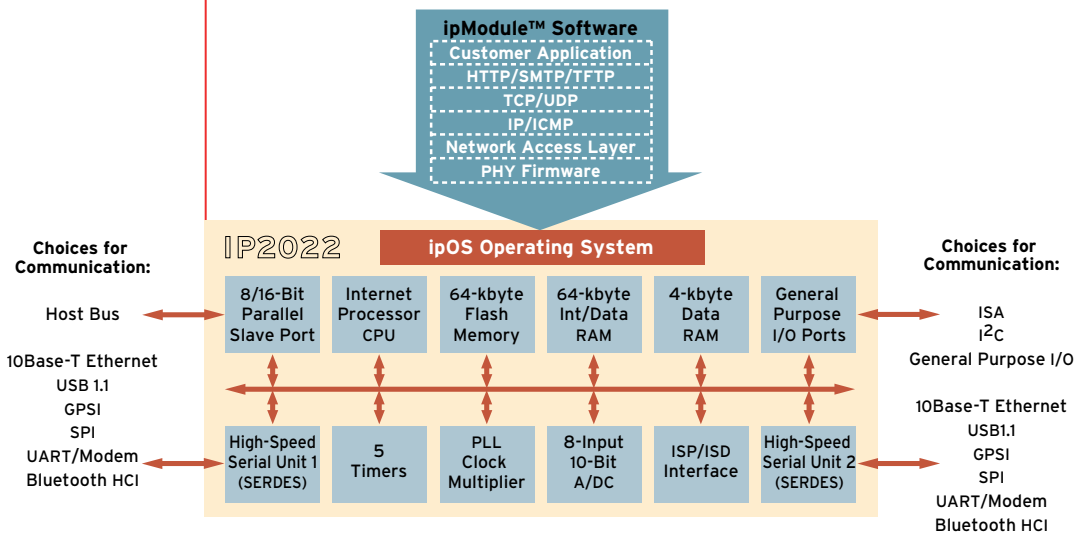
### INSIDE THE IP2022

In basic terms, what's needed to bring the Internet into a client device is a two-port bridge. One port talks to the Internet Protocol (IP) and the other port talks to the device. The Ubicom IP2022 can handle almost any two-port bridging application at the Internet edge. Depending on the application, those two ports could be Ethernet/802.11 wireless, or HomePlug/Ethernet, or USB/Bluetooth, and so forth. Most Internet-edge applications use two ports, but some use more. In the future, Ubicom's next generation multi-protocol processors will have the ability to link 4-, 5- or 6-port devices as well.

Although the IP2022 implements I/O in software, it does have some generic I/O circuitry in hardware: two serializer/deserializers for serial I/O and a set of all-purpose I/Os for parallel interfaces. The serializer/deserializer (SerDes) units include a 10Base-T Ethernet PHY enabling the chip to talk directly to the Ethernet magnetics with no external silicon. The SerDes also supports common serial protocols such as USB 2.0 Full Speed, UART, SPI, and GPSI.

The parallel protocols use the I/Os directly, with no additional hardware on-chip. It's all done through software operating directly on the pins. The IP2022 can implement numerous parallel buses, including PCMCIA, ISA, I2C, and Utopia. Also, by using those general-purpose I/O pins, the IP2022 can link to external flash memory devices of almost any size or configuration.

The IP2022 embeds four linear feedback shift register (LFSR) units that provide hardware support for the compute-intensive inner loops of algorithms commonly used in data communications. The LFSR units are programmable and can be adapted for algorithms used by the Bluetooth, Ethernet, HomePlug, HomePNA, HomeRF, 802.11, and USB communication protocols. Such algorithms include cyclic redundancy checking (CRC), data scrambling, data whitening, encryption/decryption, and hashing.



**FIGURE 1:**  
**IP2022 BLOCK DIAGRAM**



### **EFFICIENT MEMORY ARCHITECTURE**

Memory, both on-chip and off-chip, can have a huge impact on the total BOM cost of any system. With that in mind, Ubicom crafted the IP2022's memory architecture to be right-sized for Internet-edge applications. The IP2022 implements a Harvard architecture, meaning separate data memories and code memories. At 4 Kbytes, the on-chip data memory appears to be very small. It doesn't seem big enough to do the kind of packet-buffering needed for traditional Ethernet bridging applications. But because the IP2022 can process packets as they arrive, there's no need for a large data memory. The IP2022 never buffers packets in its 4 Kbytes of working memory. Buffering is handled by the chip's program memory or external SRAM.

For on-chip code storage, the IP2022 has 64 Kbytes of flash memory, which runs at 40MHz. At boot-up, the system copies the code that needs to run at higher speeds – mainly the interrupt service routines – into the program memory (PRAM). The 16 Kbyte PRAM (which consists of SRAM) runs at the chip's full 120MHz speed and can also buffer packets. Because the IP2022 uses a memory-to-memory architecture – a single instruction can make two memory references – the data memory is dual-ported. The PRAM and on-chip flash memory blocks are single-ported.

### **EXTERNAL MEMORY IS OPTIONAL**

In typical Internet-edge networking applications, where you're simply bringing a network protocol into the device, no external memory is required. The IP2022 does support external flash memory and SRAM for applications that need it. For example, an embedded web server that has complex web pages of 200 Kbytes or more will need that extra flash memory. The IP2022 treats external flash as a file system, so if a program asks for a URL that's actually a local file, the IP2022 simply pulls the web page out of the external flash.

On-line firmware upgrades would also require external flash memory. The ability to upgrade firmware over the Internet is a great way to cut costs because the vendors or customers don't have to recall the product or send out a service technician. For in-field upgrades, the device needs enough memory to buffer a complete firmware image, because it could be disabled if a dropped network connection or power outage interrupts the "brain swap."

For that reason it's better to have an authenticated checksum image of the new code before replacing the old code. That means downloading a complete copy of the new firmware, verifying it with a checksum, and then copying it onto the chip. If the code image uses all or most of the 64 Kbytes of on-chip flash, then external flash is necessary to buffer the new code. As mentioned earlier, because the IP2022 links to external flash memory via software I/O, you can use virtually any size or configuration of external flash.

The IP2022 also supports up to 2 Mbytes of external SRAM. In typical applications that SRAM would be used for packet buffering. Buffering is necessary when bridging between two protocols that have different data rates. When a packet comes in at a high data rate through one interface and goes out another interface at a low data rate, they will bunch up. About 30 to 100 Kbytes of SRAM is a reasonable amount of buffering for this kind of application.

### **FAST PACKET THROUGHPUT**

Because Ubicom's Software SoC technology handles I/O in software, it must deliver performance that's on par with data movement over dedicated hardware pins. The IP2022, when combined with its custom OS, actually does better than that – it delivers true hard real-time operation.

Enabling that level of performance requires very fast context-switching and interrupt-handling. In a conventional hardware SoC with an RTOS, the typical strategy is to trigger one interrupt per packet. As a packet comes into a FIFO buffer on the I/O pins, the processor sends the packet to memory via DMA (direct memory access). When the DMA transfer completes, the processor interrupts the RTOS, which operates on a full packet per interrupt.



Ubicom takes a different approach. The IP2022 triggers an interrupt whenever an I/O pin changes, operating on the data directly off the pin. For serial protocols like USB and Ethernet, the 16-bit FIFO buffer in each SerDes handles the data. The IP2022 takes an interrupt every 16 bits on the serial interface. So while a conventional SoC might take an interrupt every 1500 bytes when an Ethernet packet arrives, the IP2022 takes an interrupt every two bytes. For that reason, the IP2022 doesn't need to waste silicon for a large on-chip FIFO buffer or DMA logic. Moreover, because the IP2022 is more responsive, it can process a packet as it's coming in instead of waiting for the whole packet to arrive. That results in lower latency and higher performance.

#### **INTERRUPTS ARE ALWAYS ON**

The IP2022 can respond to an interrupt in three clock cycles, or 25 nanoseconds at 120MHz – a level of determinism beyond the capability of conventional processors running more complex operating systems such as Linux® or VxWorks®. The IP2022 makes a complete copy of the machine state in this interval, so the interrupt handler starts running three clocks after an event triggers the interrupt, such as an I/O event or a timer event.

Another important difference between the IP2022 and conventional solutions is that the IP2022 never turns off the interrupts. In contrast, an operating system such as Linux typically turns off the interrupts for hundreds of microseconds at a time.

The reason why Ubicom's interrupt handler works so fast is that its job is simply moving the data or doing very simple operations on the data and then returning from the handler. The only time it doesn't take an interrupt is when an event is already in the handler. As a result, the interrupt latency is really determined by the length of time that one event is in the handler.

#### **MORE CONTRASTS WITH LINUX**

Nothing better highlights how the IP2022 is right-sized for Internet-edge applications than a comparison with Linux-based solutions. Although Linux is gaining impressive acceptance in many types of embedded and network appliance applications, it still brings along a lot of server/desktop baggage. The Linux kernel was developed as the core of a server OS. It's a multiuser, multitasking, multiprocessor, full-function operating system used for program development. Its networking stacks are derived from the Unix realm – designed for high-throughput server applications, not for small code size.

To right-size a solution for Internet-edge applications, Ubicom invented a better wheel. Ubicom created the ipOS™ and ipStack™ from scratch, both targeted at these applications. The OS and protocol software require only 37 Kbytes for a typical embedded web-server application.

Unlike Linux, the ipOS is not a protected-mode, multi-process system – it basically does one thing at a time. As a result, it doesn't have to do much more than manage memory and the timer resources. Meanwhile, the ipStack is a run-to-completion protocol stack. When a packet arrives, the stack processes the whole packet through all the layers and then moves on to the next packet.

This simplified stack processing avoids some significant bottlenecks. In contrast, most other stacks do a lot of queuing at each layer. Complex OSes have multiple packets in a process simultaneously, with multiple processes and multiple context switches happening in the stack at the same time. Each of those context switches causes a loss in performance due to OS overhead.

The Ubicom ipStack avoids context switches and avoids queuing the packets, which wastes memory. Instead, it just takes a packet as it comes in and processes it all the way through the whole stack, operating on one packet at a time.



An ability to process packets in parallel is a big deal in a large machine like a Linux server with five network interface cards all vying for attention. Such systems typically use Gigabit Ethernet or several 100-Mbit Fast Ethernet links. In contrast, protocols at the Internet edge are running at 20 Mbits per second (20Mbits/s) or less. With that in mind, the Ubicom IP2022 is optimized for low latency and line-speed performance.

#### IT'S JUST RIGHT FOR THE 10-MBITS/S UNIVERSE

For most devices at the Internet edge, 10 Mbits/s is sufficient. At 120MHz, the IP2022 can achieve 120 native MIPS, which is plenty of performance for any 10-Mbit protocol. The IP2022 can run most Internet-edge protocols at full speed using only a quarter of its capacity. All these protocols run at around 10 Mbits/s – Ethernet, 802.11b, and HomePlug, for example. Other Internet-enabled embedded devices, such as a refrigerator, an LCD projector, or a security camera, have even lower data rates.

The IP2022 can even talk to a 10/100 Ethernet PHY chip and implement a 100-Mbit Ethernet MAC in software. Although it can't reach the full 100-Mbits/s speed, it can keep pace up to around 30 Mbits/s. In a two-port application, only one port is typically 100-Mbits/s anyway, so there's no need to sustain even close to 30 Mbits/s through the device.

Figure 2 demonstrates how the IP2022 can significantly reduce the BOM cost of an Ethernet-enabled embedded product while delivering superior performance. The table compares the performance and features of the IP2022 with some typical competing solutions.

PERFORMANCE*	Ubicom IP2022	NetSilicon™ NET+ARM™	
Max Operating Speed	<b>120 MHz</b>	44 MHz	
Native MIPS	<b>120</b>	44	
TCP Throughput			
10Base-T Ethernet	<b>&gt; 9 Mbits/sec</b>	N/A	
10/100 Ethernet	<b>&gt; 25 Mbits/sec</b>	> 4 Mbits/sec	
<b>EMBEDDED ETHERNET APPLICATION SYSTEM REQUIREMENTS**</b>			
Processor	<b>IP2022</b>	Net+50	Rabbit 3000
Crystal CPU	<b>4.8 MHz</b>	Variable (18 MHz and up)	
Ethernet MAC	<b>Integrated</b>	Integrated	
Ethernet PHY	<b>Integrated</b>	10/100 PHY	
Crystal-Ethernet	<b>Integrated</b>	25 MHz	
Flash	<b>Integrated</b>	2 MB	256 kB
RAM	<b>Integrated</b>	4 MB	128 kB
Estimated Total BOM Cost	<b>\$ 13.50</b>	\$ 36.75	\$ 17.55

**FIGURE 2:  
THE IP2022 STACKS UP WELL  
AGAINST THE COMPETITION.**

\*Based on public information available from NetSilicon and Rabbit Semiconductor.

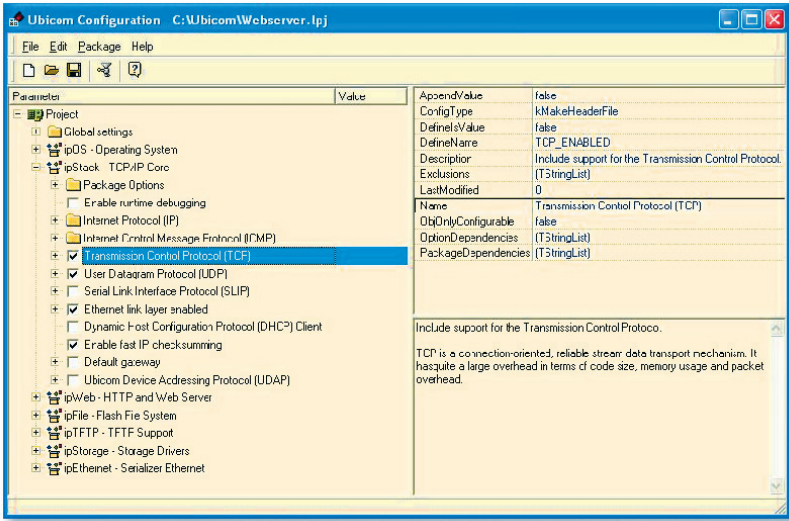
**EASY I/O CONFIGURATION**

Although Software SoC technology is a more sophisticated approach, Ubicom hides the complexity from users. To make it easier for users to create their own I/O interfaces, Ubicom provides the ipModule™ configuration tool. This tool lets developers select specific functions for their solution, set parameters, and then automatically combine the functions into integrated object code. Developers can then download the program into the IP2022 through an ISP/ISD interface and start debugging. Once the program is stable, developers can generate an executable file without debug code and download it into the IP2022's flash memory.

The ipModule configuration tool has a simple graphical user interface. If your application needs one USB port and one Ethernet port at half duplex, you just check those options on the screen. The ipModule tool then builds the appropriate make files and header files to compile the entire application as a binary image. Users can concentrate on writing their high-level application code, which might call web pages from a web server or perform other application-specific tasks, such as turning other devices on or off.

There are lucrative opportunities waiting in the Internet-edge market. But it's becoming clear that shoehorning generic SoCs or general-purpose processors and bulky full-featured OSEs into a small design is a costly solution. What's needed is a right-sized solution like the IP2022. It can slash your BOM costs by taking unnecessary hardware off the silicon and reducing memory requirements.

Because the IP2022 lets every protocol use the same pins in a different way, it also provides the flexible multiprotocol support needed to serve the diverse needs of the Internet edge. By processing the packets as they're received with low-latency interrupt handlers, the IP2022 can significantly boost overall system performance. And by accelerating your development, the IP2022 frees up more engineering resources for meaningful product differentiation while reducing your time to market.



**FIGURE 3:**  
**THE IPMODULE TOOL HAS A POINT-AND-CLICK GRAPHICAL USER INTERFACE FOR CONFIGURING I/O OPTIONS.**



635 Clyde Avenue

Mountain View, CA 94043

tel 650.210.1500

fax 650.210.8896

www.ubicom.com

Ubicom, Software SoC, ipOS, and ipStack are trademarks of Ubicom, Inc. All other trademarks and registered trademarks contained herein are the property of their respective owners.