

# ITX Motherboards vs. Single Board Computers (SBCs) vs. Microcontrollers (MCU)

Tak Auyeung, Ph.D.

May 12, 2005

## 1 What are We Talking about?

This article discusses the relative advantages and disadvantages of Mini ITX form factor PC motherboards versus single board computers (SBCs) versus microcontrollers (MCUs) in embedded system control applications. This article may be of some value if you are thinking about putting together a relatively complex system, particularly when you need a TCP/IP stack combined with web serving abilities and/or some kind of database/file system backend.

**Disclaimer:** Dr. Tak explicitly declares that all outcome as a result of reading this document is solely that of the reader. Dr. Tak is not responsible for any property damage, personal injury or other losses as the result of reading this document.

## 2 What is an ITX form factor motherboard?

ITX motherboards (also known as mini-ITX) are small PC motherboards (17cm by 17cm) that are typically self-contained like their larger ATX counterparts. How self-contained? Try to envision the inclusion of memory sockets, processor, VGA card, IDE bus, floppy port, 10/100 base-T ethernet controller, sound card, firewire (IEEE 1394) and USB ports. Some, but not all, also have Disk-on-Chip (M-system) connectors for solid state storage. Some even have optional RS-422/485 abilities.

Mini-ITX motherboards typically come with a 500MHz to 2GHz low power processor. Some come with processors that do not require fan cooling at room temperature, others require a small (*small*) fan to keep the processor cool enough. Some case manufacturer make aluminum cases with enough heat sinking so that even the faster processors can run without fan cooling (Nehemiah Hush Silent Mini-ITX PCs).

Computers built from Mini-ITX motherboards are typically not as fast as a regular desktop or notebook computer. However, they still pack quite a bit of processing power due to internal FPUs and 32-bit architectures. Most mini-ITX motherboard use regular DIMMs as their full-size ATX counterparts, which makes memory very inexpensive.

## 3 What is an SBC?

An SBC (single board computer) is just what the name implies, a computer that consists of a single printed circuit board. The name SBC was significant back in a time when PCs require many peripheral cards in addition to the motherboard.

But that has been changed.

Almost all PC motherboards are now SBCs. In other words, there is no need to add any additional peripheral cards in addition the the motherboard. Most modern ATX motherboards include everything a PC user needs, so they are, by definition, SBCs.

In a narrower scope, SBCs typically refer to *ruggedized* single board computers that can handle a wide range of temperature and humidity. SBCs also usually claim to require less power and single flexible voltage source (power supply) instead of multiple voltages required by mini-ITX motherboards.

## 4 MCUs

An MCU is basically a computer-on-a-chip. A “typical” MCU is the ATMega128 from Atmel. It is available in a TQFP-64 package, as well as a smaller MELF package that is difficult for hand soldering.

To get an MCU working, you just need VCC and ground. In most cases, you don’t even need regulated VCC, just some voltage within a fairly wide range (3V to 5V, for example). The I/O pins of an MCU are usually not protected, but we’ll talk about that later.

Most modern MCUs store programs in flash and are ISP (in system programmable). This means you only need a 5 to 6 pin connector left open for programming, and have the MCU soldered onto a PCB permanently. Most program flash memory are rated for at least 10,000 erase cycles.

As far as speed is concerned, modern MCUs can have 8 or 16 bit pipelined processors. This often translates to one clock per instruction for most instructions. The clock speed of MCUs ranges from sub-megahertz (for low-power applications) to 20+MHz for performance.

One major drawback of MCUs is the size of internal RAM. Most only come with 8kB or less internal RAM. This is more than sufficient for *many* applications, including Micromouse robots. However, there are some applications that require more RAM than MCUs can offer.

## 5 Let’s Compare Them!

This section lists some of the most important factors for embedded control, and compare mini-ITX motherboards to “traditional” SBCs.

### 5.1 Cost

#### 5.1.1 Mini-ITX

A mini-ITX motherboard ranges from US\$100 (with a 0.5GHz processor) to US\$168 (with a 1GHz processor) without memory. The cost of memory is the same as regular motherboards. As of the writing of this article, 128MB of PC133 memory is about US\$25. That’s not much.

A 32MB disk-on-chip memory can be found at US\$20 at surplus stores. For some applications, that’s more than enough. For more capacity and frequent rewriting, a small capacitor hard disk (IDE, 30GB) is about US\$50. Furthermore, most Mini-ITX boards now come with USB 2.0 interfaces. This means you can use a US\$50 1GB jump drive as your mass storage.

Mini-ITX motherboards can use regular ATX power supplies. However, one can also use smaller power supplies. A kit consisting of a 55W 12VDC DC-DC converter *and* a 110-230 VAC to 12VDC power supply is about US\$60.

By the time you have a fully functional computer, you probably will have spent about US\$220 to US\$300.

#### 5.1.2 SBC

It is difficult to compare. Most SBCs do not have the same features as a mini-ITX. For comparison purposes, I choose a Z-World OEM2500 Coyote SBC. This SBC has a 29.4 8-bit Z80-derived process, 10/100 base-T ethernet controller, 128kB of RAM and 256kB of flash. It also has four serial ports and some digital inputs suitable for switching small loads.

The price for one SBC is about US\$100. However, you must order with a proprietary development software platform. The development kit, including the OEM2500, is US\$300.

#### 5.1.3 MCUs

The cost of a single MCU with 128kB of program space and 4kB of RAM space is about US\$10. This is dirt cheap, even when compared to low-end SBCs. However, a single MCU (as a chip) doesn’t do much. You’ll need some external circuits and a circuit board to use an MCU in an application.

Depending on what you need to interface with, the add-on circuit can be inexpensive, or very expensive. To compare to a “typical” SBC designed for embedded control, let us use the following components:

- Voltage regulator

- Various inexpensive passive components
- 3A MOSFETs (8 times US\$0.50)
- Flyback diodes (8 times US\$0.20)
- LEDs (\$1)
- Ceramic resonator (\$0.5)
- DIP switch (\$1)

The circuit board can be as simple as a breadboard, or a custom PCB. We'll just say it costs US\$10. After adding all the costs, a function circuit board based on an MCU is about US\$20. With volume discount, it is likely to reduce the cost to less than US\$10. This set up is comparable to an SBC that costs easily \$75 or more.

#### 5.1.4 Apples and Oranges?

This is not a direct comparison. If you look at the processing power, the mini-ITX is at least 20 times faster at integer math, and who knows how much (more than 200 times) faster in floating point math. The mini-ITX also wins *big time* with storage and RAM capacity. As with all PCs, the mini-ITX also offers user interfaces such as VGA ports, mouse ports and keyboard ports.

The SBC solution, on the other hand, requires a very modest (read inexpensive) power supply. The SBC also includes drivers suitable for switching small DC loads. In other words, the two items being compared here do not exactly have the same type of features.

The MCU solution, however, is directly comparable to the SBC solution. There is no doubt that the MCU solution is significantly less expensive than the SBC solution.

Read on, we'll analyze how these two items compare in terms of other aspects of embedded system development.

## 5.2 Operating System

### 5.2.1 Mini-ITX

A mini-ITX based system can literally run any operating system. For those addicted to Windows, *yes, you can run Windows*. However, for much better reliability and stability, I recommend either good old DOS or Linux.

Linux in an embedded system? Are you crazy?

Nope, Dr. Tak is not crazy. I have worked on jobs where Linux was used as the OS of an embedded system. But doesn't Linux require lots of memory and disk space?

The answer is yes, and no. A stripped down version of Linux (with minimal stripping and no kernel hacking) can run with 8MB of RAM and 8MB of flash memory. Of course, how much memory is needed depends greatly on what features are needed as well as the nature of the embedded application. With memory prices at 128MB for US\$25 and disk-on-chip at 32MB for US\$20, there is no problem sizing up.

Don't forget, Linux comes with many features. If you can fit a small hard drive into the system, you can *easily* run a database server, a web server, the whole TCP/IP stack and *whatever* you want to run that's available to a desktop Linux system. Yes, that includes X windows and all kinds of applications that come with X windows.

For those who have hard real-time requirements, you can run QNX and most other real-time operating systems with a mini-ITX. If you don't want to pay lots of money, you can also run RTLinux, a real-time variant of Linux.

In other words, with a mini-ITX motherboard, there is no limit as to what operating system you can run.

### 5.2.2 SBC

Depending on which SBC you use, you may have the same flexibility as a mini-ITX motherboard. Or, you may have no choice.

For example, if you choose a i386-based SBC, you probably have some flexibility. But the range of flexibility still depends on the support chipset (or there lack of) as well as RAM/storage capacity. If you choose an SBC with a proprietary processing, you have no choice but to use whatever the manufacturer provides.

For our comparison, the Z-World OEM2500 board uses a proprietary Rabbit processor, which limits your choices to *none*. You must use the kernel that comes with Dynamic C (the development platform from Z-World). Z-World offers a TCP/IP stack and related software (such as a small web server) with their development platform, though.

### 5.2.3 MCU

Depending on which MCU you use, you may find a variety of free real-time kernels. The AVR-family of Atmel has many GPLed real-time kernels to choose from.

## 5.3 Input/Output

### 5.3.1 Mini-ITX

A mini-ITX motherboard has no ability to control machines. In other words, there is no provision to control relays, read back the states of switches and measure voltages. In order to perform these tasks, you need to design and use another PCB connected to the mini-ITX system.

How much this costs depends greatly on your requirements. Utilizing an MCU (microcontroller unit), one can easily design and build small printed circuit boards with a variety of input/output abilities. For example, an ATmega128 (from Atmel) has about 48 input/output points that can be configured for CMOS/TTL I/O, PWM, UART, ADC and external interrupt applications. You still need to add interface circuits such as transistors, diodes and opamps to perform industrial control. It is easy to design such a printed circuit board, even route it by hand, because an MCU does not need memory busses.

The mini-ITX system can communicate with this extra PCB via many protocols. The easiest method is RS-232. Because RS-232 is widely supported by many operating systems, it is easy to debug the interface between the mini-ITX system and the extra PCB. For higher speed and more flexibility, some mini-ITX also has I2C busses. If you use an MCU with built-in USB support, all mini-ITX motherboards have USB support.

### 5.3.2 SBC

Most SBCs come with *some* control interface circuits. For example, the Z-World OEM2500 has two PWM output, 8 digital inputs and 8 200mA capable outputs. If that fits your need, you don't have to do anything.

On the other hand, if your application requires to switch more than 200mA, you have to design your own PCB. If your application requires one or more H-bridges (for motor control), you need to design your own PCB. If your application requires high precision ADC ability, you need to design your own PCB.

SBCs are great when you don't need any more than the provided features. As soon as you need any more than that, however, SBCs do not offer much advantages of mini-ITX motherboards.

### 5.3.3 MCU

An MCU needs external components for real-life input/output interface. However, such components and circuits are inexpensive to design. One also has the flexibility of prototyping with a breadboard first, then design a PCB for mass production.

## 5.4 Software Development

### 5.4.1 Mini-ITX

Because a mini-ITX motherboard is really a PC, you can choose any development tool available for a PC. Dr. Tak has successfully used gcc (GNU C compiler) in a Linux-based environment. If you are a fan of Windows, you can use tools from Borland, Microsoft and etc.

Setting up the development environment can be easy, or it can be a little challenging. The determining factor is how close the target system resembles a regular PC system. If your target system has 128MB of RAM and 30GB, it can be its own development environment! You can easily install gcc, gdb, telnet and other tools in the so called target system, and remotely develop your application using telnet.

On the other hand, if your target system has little resources, it can be difficult to set up the software development platform. For example, Dr. Tak's 8MB RAM and 8MB flash deployment required some fine tweaking to make everything

fit. You can still develop much of the software on a resourceful system, but building an image for deployment requires attention.

With the mini-ITX solution, you also need to develop the application running on the separate printed circuit board. This cross development is a little more difficult than porting code from a resourceful PC to a less resourceful mini-ITX platform.

#### 5.4.2 SBC

Some SBCs, particularly the ones equipped with x86-type processors, can be programmed by older Borland or Microsoft tools (DOS mode applications). Others, particularly ones using proprietary processors, can only use whatever the manufacturer supplies. For example the Z-World OEM2500 can only use Dynamic C as a development platform.

The lack of flexibility sometimes translates to integration. With the Z-World OEM2500 and Dynamic C, one can probably set up the development platform in less than 15 minutes and start to code. This is mostly because everything (hardware, software, documentation and etc.) is designed and manufactured by the same company.

#### 5.4.3 MCU

Some MCUs, such as the AVR-family, are supported by the GCC toolchain. This means the development tools are free. For those who insist to pay for something, commercial MCU compilers start at US\$300 or so. I don't see any reason to purchase commercial ones, however. The GCC toolchain is professional, standard conforming and reliable.

#### 5.4.4 Amortized Cost

A mini-ITX based system has a much more difficult first-time software development platform set up. However, you need to ask yourself: "how often do I need to do this?" Once you know how to set up for one project, setting up for the next project requires very little time.

### 5.5 Ruggedness

#### 5.5.1 Mini-ITX

Mini-ITX motherboards are no more rugged than their full-size ATX counterparts. In other words, they are design to run in a controlled indoor environment at room temperature and comfortable humidity. The additional PCB, however, can be designed to be as rugged as it needs to be. If you can put the mini-ITX system in a controlled environment and only put the custom PCB into the hostile environment, you still have a viable solution.

For applications that do not have any controlled environment, such as engine control for an automobile, you probably don't want to use a mini-ITX based solution.

Note that you *can* provide a controlled environment at your expense. A sealed enclosure with thermal-electrical cooling/heating can provide the environmental shield for the operation of a mini-ITX system. Does it make sense to use this type of enclosure? It depends. In an expensive and complicated system, you may have to do this *anyway* for other sensitive electronics. For cost sensitive or space constrained applications, however, a controlled environment may not be an option.

#### 5.5.2 SBC

How an SBC survives in hostile (hot, cold, humid, dusty and etc.) environments depends on the design and manufacturing. For example, the Z-World OEM2500 has a rated operating temperature range of -40C to 70C and operating relative humidity of 5% to 95%. This is much more than most mini-ITX motherboards can claim.

This is one factor that make SBCs your only choice in certain applications (other than designing your own SBC from scratch).

#### 5.5.3 MCU

MCUs are available in commercial or industrial rating. In other words, MCUs can be just as rugged as SBCs. The other components you connect to an MCU are typically the same type of components you find in an SBC, so an MCU solution can be designed to be just as rugged as the most rugged SBC.

## 6 Summary

What is the verdict? Which option you should consider depends on your requirements. Dr. Tak provides a brief summary here.

### 6.1 Mini-ITX

You should at least consider a mini-ITX solution if you can provide the following for the control system:

- controlled environment (temperature, humidity, dust, vibration)
- a relatively large space
- plenty of power (20W or more continuously, 50W peak)
- ample development platform set up/installation time (for the first project)

You should definitely look into a mini-ITX solution if you provide the previously mentioned prerequisite, and you require the following from the control system:

- lots of processing power
- lots of RAM to run complicated programs
- lots of storage to store data
- many features of a full-blown OS, such as
  - full TCP/IP stack
  - file storage system
  - web server, email server, file server, etc.
  - multi-processing and multi-threading
  - remote administration
  - database backend
- standardized programming languages/envirionments

### 6.2 SBC

You should consider using an SBC if you have the following requirements:

- operation in a hostile environment (temperature and humidity)
- little power consumption
- small space
- quick first-time development platform set up and installation
- more RAM space than MCUs internally support

If an SBC offers the required input/output requirements for your application, it is most likely a less expensive and cost effective solution compared to a mini-ITX system.

### 6.3 MCU

You should consider using an MCU-based solution if you have the following requirements:

- very low per-unit cost
- low power consumption
- very customized interface
- free and professional software development tools

The only drawback of an MCU-based solution is that you need to spend some time to design the interfacing circuit. Most electrical engineers should be able to handle this type of design, though.