

## Re: uC selection

---

*Source:* <http://sci.tech-archive.net/Archive/sci.electronics.design/2008-03/msg02983.html>

---

- *From:* [nico@xxxxxxxxxxx](mailto:nico@xxxxxxxxxxx) (Nico Coesel)
  - *Date:* Mon, 17 Mar 2008 22:08:02 GMT
- 

"David L. Jones" <[altzone@xxxxxxxxxxx](mailto:altzone@xxxxxxxxxxx)> wrote:

On Mar 15, 3:01 pm, "Jon Slaughter" <[Jon\\_Slaugh...@xxxxxxxxxxx](mailto:Jon_Slaugh...@xxxxxxxxxxx)> wrote:

<[a7yvm109gf...@xxxxxxxxxxx](mailto:a7yvm109gf...@xxxxxxxxxxx)> wrote in message

[news:86a4146e-ddcc-40aa-8316-5ded547c1fc9@xx](mailto:news:86a4146e-ddcc-40aa-8316-5ded547c1fc9@xx)

On Mar 14, 7:14 pm, "Jon Slaughter"  
<[Jon\\_Slaugh...@xxxxxxxxxxx](mailto:Jon_Slaugh...@xxxxxxxxxxx)> wrote:

Currently I use microchip pics but I'm looking possibly to switch, but what?  
Is Atmel worth it? What about TI? I'm looking for something similar to microchip but more of a commercial aspect. I have never seen any commercial device that uses a pic and I assume there are reasons for this? It seems that pic's are only for hobbyists so using them in a commercial product is a no-no?

It's spelled "hobbyist". Just like "lobbyist".  
PICs are used in cheap, high volume applications where you won't even see the part number or logo.  
What language do you program in? What kind of applications? Simple button-LCD-I2C or more complex signal processing?

## Re: uC selection

I have programmed in a large number of languages and that's not really the issue (python, php, C/C++/C#, java, assembly, pascal, etc..). The main thing is the functionality and scalability.

PIC are hard to beat for scalability.

Check out the real details instead of just guessing.

The development toolsets are consistent across 8-bit, 16-bit, DSP, and 32bit lines.

And crappy... The PIC32 bit toolchain uses an ancient GCC compiler. Free and student versions are crippled on purpose (not all optimisation options available). So if you want the most from the PIC32, you really don't want to use the toolset from Microchip.

Besides, why mess around with obfuscated 8 bit devices which requires proprietary tools while a proper ARM based device with open tools can be bought for the same amount of money? Microchip has noticed this too hence the introduction of a microcontroller with a real CPU. I bet MIPS made them a very good offer to choose a MIPS core over an ARM core.

When it comes to a consistent toolset I'm quite sure Eclipse beats most microcontroller development environments hands down. And Eclipse can work with many more targets than just PIC. If you want a top-notch consistent development toolset which works for many devices, get Eclipse and GNU tools.

Look at the high end 32bit PICs:

[http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&nodeId=2591](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2591)

And there is still a free student version C compiler for it.

If you need anything bigger than those then you are no longer in the realm of microcontrollers any more, but microprocessors like the ARM that can access massive amounts of memory and other resources.

Like I stated above, PIC32 is MIPS32 based. They shouldn't call it PIC. It just doesn't do the MIPS core right. MIPS and ARM are almost the same. Both are 32bit CPUs with a 32 bit instruction set and a 16 bit compact instruction subset.

—

Programmeren in Almere?

E-mail naar [nico@nctdevpunt.nl](mailto:nico@nctdevpunt.nl) (punt=.)

.