

Re: PIC vs AVR vs ARM

Source: <http://coding.derkeiler.com/Archive/General/comp.arch.embedded/2006-10/msg00251.html>

- *From:* "rickman" <gnuarm@xxxxxxxx>
 - *Date:* 5 Oct 2006 05:38:25 -0700
-

Ulf Samuelsson wrote:

"rickman" <gnuarm@xxxxxxxx> skrev i meddelandet
news:1159903397.463883.11040@xx

We have used AVR MCUs in many of our products and were very happy with them. On a new project I decided to take a look at the ARM MCUs to see if we could branch out from some of the limitations of the AVR. We did a very exhaustive comparison between the various ARM processors and the ATmega128 and found that the ARM chips were generally lower power, lower cost and fit in a smaller footprint on the board. We also were able to use a much smaller crystal.

When power is an issue, you typically have to spend as much time as possible in sleep mode, and the Picopower AVR will be at least an order of magnitude better than the AT91SAM7 here.

And whether this is significant depends on the application. The AVR could be ten orders of magnitude better than the ARM in sleep mode, but if the active mode power is 90% of your power budget, the sleep mode will not have much of an impact on total power.

As I said, depending on your application this may be important. If you are designing a data logger where it is asleep for two weeks and then is triggered to record something for a few seconds, sleep mode power may be important. But this is a very small percentage of applications.

Also, when running from an R/C oscillator you can turn on/off almost instantly, while the AT91SAM7 probably have to start the PLL which will take ~16 ms. One drawback of Picopower is that the startup time from sleep is increased from a few clock cycles to about 70 us. This is the time it takes to activate the brownout detector which is disabled in deep sleep.

Re: PIC vs AVR vs ARM

(Don't worry, the part is protected from Brown-Out by the Power On Reset in deep sleep)

And the R/C oscillator is only useful in a small percentage of applications where you don't need any more timing precision than what is required to run a UART, and just barely that!

I think the PicoPower AVR is therefore hard to beat when you really need low power.

Yep, in the small percentage of apps where you need the unique features of an 8 bit MCU tailored to low power when it is not running, then it can do a good job. But most MCU apps run the MCU a fair percentage of the time if not 100%. In those apps the power is dominated by the active current and the ARM can beat the AVR there.

.