

Re: Microcontroller Project

Source: <http://sci.tech-archive.net/Archive/sci.electronics.design/2006-03/msg03907.html>

- *From:* Keith <krw@xxxxxxxxxxx>
 - *Date:* Mon, 27 Mar 2006 16:40:02 -0500
-

In article <1143494228.117311.162780@xx>, altzone@xxxxxxxx says...

Keith wrote:

In article
<1143447041.276029.172680@xx>, altzone@xxxxxxxx says...

Keith wrote:

On Sun, 26 Mar 2006 17:28:03 -0800,
David L. Jones wrote:

akshaychander@xxxxxxxx
wrote:

Sorry for
the late
reply.

Yes, I have
to do the
project in
8051. I have
studied the
theory of
8051, but
have very
little
practical
experience.

As far as
languages
are
concerned,
there is no

Re: Microcontroller Project

restriction. I
am
planning on
using C.

In that case a good C
compiler will take care of
most of the low level
8051 stuff for you.

I disagree! A C compiler will just mask the
nonsense underneath. It
will come back to bite you (think "stack").

Ok, I'm not an 8051 guy, so I'll take your word there is a
potential
issue here.

I looked into various compilers when I did a major (~25kloc) 8051
project a decade ago. The only one that didn't mask the
limitations (and even advantages) of the 8051 was PL/M51.

I'd then suggest you take a look at some more modern compilers perhaps.
Countless people use C on the 8051, I'm sure it's not that bad.

The point is that one is *necessarily* doing bit-banging on the
hardware (the whole point of a uC). C doesn't give much advantage
and a lot of headaches.

Is assembler any different? (especially for a beginner?)

Yes, because of the need to be keenly aware of the processor
hardware limitations. The 8051 isn't an ordinary processor. It
only has 128 bytes of RAM (the variant I used had 256) and the
stack's gotta fit in there too. When the stack overwrites the
registers (memory mapped) all sorts of fun things happen.

If the stack overflow thing is your only gripe then that's not much of
a problem.
You'll get dynamic stack allocation errors on most micros, it's not
that hard to avoid unless your program is doing something very strange.

Re: Microcontroller Project

When the stack overflows you don't see any allocation errors. You simply get hosed special purpose memory, or perhaps registers get overwritten. This sort of thing is **not** easy to track down and for an amateur with no tools it might be a show-stopper. Clearly you don't understand the 8051 memory model(s). It is very strongly Harvard with very little directly addressable memory, which is mapped into several different resources.

There are great many applications where the stack usage is very predictive and guaranteed. Typical rule of thumb is to ensure your stack space is twice as big as the largest you'll anticipate using and find during testing. If you use static allocation for everything (and you should be on a micro really) then you'll usually know exactly how much RAM you have left.

You do realize that the 8051 only has 128 BYTES of memory and the stack has to fit in about half of that. Thumbs are harder to hit when you can throw mega-giga bytes at the problem. The 8051 ain't an x86!

You can get stack overflow errors on the PIC too, and I've written plenty of C programs on that in less RAM than 128bytes.

Now, think about someone who's never used a uC with no tools debugging this mess. Assembler is far easier, IMO; the less abstraction the better.

<snip>

Again, I disagree. Normally one would assume this, but the 8051 is a "special" (as in; designed by a "short-bus" architect) case.

So what are you saying? That C compilers for the 8051 are no good?

I find this very hard to believe.

I wouldn't advise a novice to use C on an 8051, no. If you're **really** good at C it can be done but given the strange architecture I don't see what it'll gain over assembler. I don't see any real code being portable so why bother?

Re: Microcontroller Project

There are a lot of 8051 C compilers around and I'm sure a lot of people use them without issue.

So you've been bitten on the stack overflow issue, but I would not use that as an excuse to stay clear of C.

I see no advantage in using C. Portability isn't an issue since hardware is the name of the game with uCs. Any abstraction will lead to resource conflicts. No, actually, I haven't been bitten by stack issues because I've used assembler. I don't let the compiler assign *anything*; not even what byte the bits go in.

Many people have said that C on a low end PIC (another so-called "crippled" architecture) is useless, but excellent ANSI C compilers are available for them, and the low end ones too.

All sorts of things exist, many for no good reason. No, I don't see any advantage to C on an 8051. I can't speak to the PIC, but my guess is it's much the same story.

Well I can speak for the PIC, it has a notoriously horrible architecture that is very "C unfriendly" yet the modern C compilers like the HiTech C do a fantastic job. I am sure that thousands happily use C on the 8051 without problem. There is no real reason to avoid C on any micro when good compilers exists.

I disagree. C has no advantage over a half-decent macro assembler and a MAKE utility. Abstraction has a *lot* of disadvantages when you're severely hardware constrained. Sure, people use C on 8051s, but I suspect for many C is the only tool in their shed.

You could write the C program on a PC first, and then port to your 8051. Instead of writing to the screen you would turn on a LED on a port of the 8051, so only a small part of your program should change between a PC version and an 8051 version.

Re: Microcontroller Project

...and watch the stack explode. Good idea!

Once again, what are you saying exactly? Are you saying that no C compiler for the 8051 can anticipate a stack overflow and give you a warning?

No, because stack overflows are run-time events. How is a compiler going to anticipate what the program is going to do at run-time.

It's called static allocation and knowing what your program does. In most programs it's not hard to know the maximum number of stack calls you can expect, and good C compilers can detect this and warn you. For instance, if you have 10 nested function calls and your stack can only support say 8 (after the static variable allocation), the compiler will warn you.

You never use conditionals? Your code must be really boring. ;-)

If you are using dynamic allocation for stuff then you are asking for trouble on any micro.

I guess PCs don't work. The point is that you *CANNOT* do dynamic allocation on an 8051. Once you've blown the stack you've eaten data with no one to tell you that your foot is gone.

The advantage of using a high level language like C is that you shouldn't have to care what micro platform it is used on.

A naive statement. "shouldn't?" There's a comforting statement. Very few people can write truly platform independent C. ...I'm certainly not one, and my bet is that given your statements here that you aren't either.

Ok, I should have said "shouldn't have to care too much"

Re: Microcontroller Project

especially for
the OP's case of a simple hangman game.
No C program is ever truly platform independent, but you
can go a long
way to making it so.

They can be made so, or at least C programmers on the comp. groups
think they do it. ;-) ...but it's usually not worth the bother.
Again, the 8051 is a strange beast. Much of its utility comes from
this strangeness (memory mapped registers, bit addressable memory,
strange and wonderful hardware, etc.). Hiding it isn't a good idea
and may invite disaster (e.g. stacks creaming registers).

Check out the PIC some day, some said it was impossible to write an
ANSI C compiler for it, yet many have done it now.

I would like to work on a PIC, but I've gone up the food chain
quite a number of steps. ;-)

<snip>

If you need help with the C
code then the 8051 is the
least of your
problems.

Granted, but working C isn't likely to help
him either.

Maybe not, he'd probably have oodles of problems with the
programmer
settings alone :->

There's that too. I wonder what he's going to use as a development
system. ...edit->compile/link->program eprom->plug->repeat is
going to be a bitch.

He'll be playing with that for weeks!

I don't envy him, unless he has an ICE of some sort, or at *least*
a card with a decent monitor (perhaps on-board with an external
EPROM, which again limits the 8051).

Re: Microcontroller Project

—
Keith
.