

Re: Purchase microcontroller dev. kit

Source: <http://sci.tech-archive.net/Archive/sci.electronics.design/2006-09/msg05850.html>

- *From:* nico@xxxxxxxxxxx (Nico Coesel)
 - *Date:* Tue, 26 Sep 2006 06:39:52 GMT
-

"David L. Jones" <altzone@xxxxxxxxxx> wrote:

Nico Coesel wrote:

Donald <donald@xxxxxxxxxxxxxxxxxxx> wrote:

Nico Coesel wrote:

mrdarrett@xxxxxxxxxx wrote:

Nico Coesel wrote:

ydoubleuz@xxxxxxxxxx
wrote:

Hi
all,

I
am
new
to
this
and
i
hope
to
purchase
a
development
kit

Re: Purchase microcontroller dev. kit

for
dev.
microcontrollers.
Due
to
the
numerous
varieties
available
in
the
market,
i
am
lost
as
where
i
should
start
and
what
stuffs
to
look
out
for
when
purchasing
these
kits.

Whatever
you buy,
make sure
the
microcontroller
has one
addressing
space (no
8051, no
AVR, no
PIC) if you
want to
keep your
code
portable.
You
wouldn't be
the first

Re: Purchase microcontroller dev. kit

developer
who has
found the
platform
that looked
so
promising
in the past
turns out to
be a
deadlock.

No 8051, no AVR, no PIC?
What *would* you
recommend, then? ;-)

Hitachi/Renesas H8 / H8S, Texas
Instruments MSP430, Analog Devices
Blackfin DSP.

Oh crap, now you did IT !!!

The religious war about my CPU is better than your CPU is
going to start.

No not at all. Look at the big picture here. Its not the CPU that
matters, its where you want to go in the future that matters!

A choice for a CPU should be driven by the question: "What if I want
to move to a different platform". With some platforms the answer to
this question is: "throw away everything you wrote and start over". So
a choice for a platform should be made with great care.

There is NO best processor, this is too small or too large, but
you will
learn this for yourself.

That's exactly why I listed a general purpose microcontroller series,
a micropower series and a full blown 300+ MHz 32 bit DSP with MMU
capable of running a genuine OS like Linux. However, generic C code
written for one, can be moved to the other.

Re: Purchase microcontroller dev. kit

Gee that's stange, how on earth have I moved C code almost seamlessly from a PIC to an AVR to a Rabbit then?

try to compile the following on an AVR:

```
const char a[]={ "hello world" }  
char b[30];  
char *p;
```

```
p=a;  
strcpy(b,p);  
strcat(b, "?");  
printf("%s %s", a, b);
```

It won't work because the different memory areas will screw-up the pointers. Some very smart and expensive compilers may be able to solve these problems at the expense of speed.

You can't be serious suggesting that a beginner think about the "big picture" and choose some oddball processor based on some perceived future requirement, that is crazy.

Oddball?? I wouldn't call the MSP 430 series oddball (lots of people are using these). You don't even need a programmer to program it! An RS232 to TTL converter (or a USB to serial converter chip) is enough to get going.

A beginner needs something that is common beginner platform so that they can get tons of support, sample code, books and other beginner level stuff etc That basically means PIC or AVR these days, that's where the action is.

So we should all be like lemmings? Even if the first one that jumps into the water drowns, the rest should follow?

Besides, you make my point exactly: if you choose PIC, then you're stuck with PIC examples. If you choose a processor for which any generic C code can be compiled, you can use any piece of C code you can get your hands on.

—

Reply to nico@nctdevpunt.nl (punt=.)
Bedrijven en winkels vindt U op www.adresboekje.nl

.