

Re: Simple multi-channel serial ADC (8-ch)?

Source: <http://sci.tech-archive.net/Archive/sci.electronics.design/2007-02/msg02714.html>

- *From:* Joerg <notthisjoergsch@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 15 Feb 2007 20:56:11 GMT
-

linnix wrote:

On Feb 15, 12:00 pm, Joerg <notthisjoerg...@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
wrote:

linnix wrote:

On Feb 15, 10:16 am, Joerg
<notthisjoerg...@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
wrote:

linnix wrote:

On Feb 15, 8:52 am, Joerg
<notthisjoerg...@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
wrote:

Rene
Tschaggelar
wrote:

Joerg
wrote:

Hello
Folks,

Re: Simple multi-channel serial ADC (8-ch)?

As
an
analog
guy
I
get
goose
pimples
when
I
see
converters
like
the
TLV1508.
Impressive
performance
but,
boy,
do
they
need
a
complicated
liturgy
to
get
data
out
of
them.
Many
of
them
must
be
told
via
SPI
that
I
want
to
read
this,
that
and
the
other

Re: Simple multi-channel serial ADC (8-ch)?

value.
Then
you
must
patiently
wait
x
clock
cycles,
wasting
away
bus
time.
The
guy
who
designs
the
host
processing
will
throw
his
coffee
mug
at
me
because
there'll
be
a
dozen
of
these.

Here
is
what
I
am
looking
for:
10
bits
or
so,
8
channels,
bone-simple

Re: Simple multi-channel serial ADC (8-ch)?

readout,
can
very
very
slow,
low
kHz
range.
All
I
need
is
to
read
DC
values
and
it
doesn't
matter
when
the
exact
sampling
time
was.
SPI,
I2C,
anything
serial
works.
I
mean,
with
a
FIFO
in
there
they
should
be
able
to
do
continuous
autoscan
and
just
dump
the
last

Re: Simple multi-channel serial ADC (8-ch)?

Re: Simple multi-channel serial ADC (8-ch)?

valid
set
of
values
upon
request.

Any
ideas?
I
am
also
looking
for
10-12
bit
DAC
with
lots
of
channels
but
that
seems
to
be
much
simpler.

Joerg,
I'd
employ
a
controller
such
as
the
AVR
Tiny26
with
8
differential
or
11
single
ended
10

Re: Simple multi-channel serial ADC (8-ch)?

bit
channels.
Since
you
need
a
dozend
of
them
just
hook
them
together
with
an
SPI
or
an
USI.

I wanted to
avoid a uC
on there
because it
wouldn't be
needed for
any
other
purpose.

Micros are sometimes
cheaper than standalone
ADCs.
You don't need to use them
for anything else.

But they need to be flash programmed and
are still more noisy.

They are
too slow for
the analog
stuff I am

Re: Simple multi-channel serial ADC (8-ch)?

doing on the
boards.

How fast do you need?
Is 500KSPS too slow for
you?
How about 1MSPS?

The ADC can be slow, a few kHz at the most. Not critical. What I meant is that I can't use a uC for any of the control gear tasks. That would have justified the hassle of in circuit programming. Unless you buy gang programmers many manufacturers do not offer file dumps. They give you a nice and cozy design environment but, for example, TI does not offer a feature where you connect an EZ430 pod or the like and just download a file right from the OS or from a web browser. You have to do that from the IAR console. Not such a good thing in this case since the engineers at my client are all specialized in optics and process control.

Yes, I understand your concern. Programming environment is one reason I am avoiding the MSP. At least with ARM, JTAG is possible. For our particular project, we need to be able to JTAG one micro from another.

I told them that, and certainly others did. All they'd have to do is maybe hire an intern who can write a simple download routine so you can dump data via SBW into the device without the design suite open. Preferably without even needing to have a clue of what's happening. Just by selecting a file, connecting the cable and hitting "Program Now". It's not rocket science. But so far, nada.

But you are assuming that the bootloader is reliable, I would not rely

Re: Simple multi-channel serial ADC (8-ch)?

Re: Simple multi-channel serial ADC (8-ch)?

on that unless the bootloader is in hardware rom, not flash and not eeprom.

The original one is hard-coded in but I don't remember whether that ain't just a write-protected flash area.

The topper would be a wee sub-program that allows one-pin programming so you could connect a little IR photodiode and then "beam it over". With good encryption to avoid data errors in ambient light. Look, ma, no cable! While it's no big deal to write such code for the MSP430 the PC part would be a challenge, at least for an analog guy like me.

If the spec is open, someone would write it for you.
Problem is that most micro makers are protecting their programming interfaces like trade secrets.

Yep, and that is a rather silly business behavior. For TI it has already cost them sales in my case. I had bugged them numerous times about some more data about their MSP430 devices, then gave up on it. Went analog, actually.

Just like a vendor who refused to disclose the schematic of a laser controller, a schematic I only needed to find out what the plant parameters would be. So, I designed it out. Slam bam, door shut. Even if they came back now and repented it would be two days too late.

--

Regards, Joerg

<http://www.analogconsultants.com>

.