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

Source: http://sci.tech−archive.net/Archive/sci.electronics.design/2007−02/msg03399.html

- *From*: "linnix" <me@xxxxxxxxxxxxxxxxx>
- *Date*: 18 Feb 2007 09:07:05 −0800

On Feb 18, 9:01 am, Joerg <notthisjoerg...@xxxxxxxxxxxxxxxxxxxxx>
wrote:

> linnix wrote:
>
> > On Feb 17, 5:01 pm, Joerg <notthisjoerg...@xxxxxxxxxxxxxxxxxxxxx>
> > wrote:
> >
> > > linnix wrote:
> >
> > > > On Feb 17, 1:05 pm, Joerg
> > > > <notthisjoerg...@xxxxxxxxxxxxxxxxxxxxx>
> > > > wrote:
> >
> > > > > Marte Schwarz wrote:
> >
> > > > > > Hi Jörg,
> >
> > > > > > Coding is easy, I meant the
> > > > > > download procedure (for
> > > > > > people other than us
> > > > > > EE types).
> >
> > > > > Well, we are getting close to a simple
> > > > > downloading solution for ARM.
> > > > > We have a gdb server running on the target
> > > > > and jtag flash the chip via

usb.

We can remotely gdb the chip, or have a friendlier program (WinGdb?) to download it. The remaining work is to figure out how to interpret the AXF file. The goal is to:

Compile in Keil (perhaps WinArm eventually), strip AXF into binary image. TCP it to the target (or programming box), flash it with usb jtag.

By simple I mean something that can be clicked from a PC and then automatically download into the uC. The user interface should consist of an *.exe file that gets double−clicked

from
Windows
Explorer,
no
more.

Exactly as ordered. First function is to read
and write the flash
remotely. This will work with any FTDI
2232 based usb cables. We
currently have a target running Linux 2.6.16
and an usb Arm. If your
target is on static IP, you can remotely
debug/download it across the
country.

That's what is needed. Now if this would be available for a
smaller uC
such as the MSP430

If they use an open standard interface like Jtag, then it would be possible. We tried with AVR, but the Jtag interface is not totally open.

AFAICT they don't, and actually migrated a bit away from JTAG with the new F2xxx family. The old ones can usually be bootloaded quite well.

I believe it would certainly increase its market
penetration into areas that have gone without any uC so far.
IOW not
chasing the competition but truly new markets.

IIRC from the Yahoo MSP430 forum there is a guy working
on it but last
time he was still looking for someone who would take on the
PC side
software.

> > It's hard to do that. We need to work on both sides at the same time. For example, we are now trying to figure out the data transfer protocol, and we need to program both Window and Linux.
> >
> > The gdb protocol for reading is:
> >
> > $ m <addr> , <count> # <checksum>
> > $ m <addr> , <data> # <checksum>
> > with + (ACK) and − (NAK)
> >
> > We would need protocol extensions to write to buffer, then a write flash command.
> >
> > > I don't want to spoil the broth here but, sorry, Linux isn't really an option. If you take me as an example none of my clients uses
> > > Linux. All Windows :−(.... ok, one MAC.
> >
> > But they all use web servers, and many of them are Linux. As long as they don't have to touch them, it would be fine. Think of it as a Jtag programming server.
> >
> > We don't want the client to mess with the Linux box as all. All the user interfaces will be done on the Window side. The Linux box will be running on read−only 64M Flash Drive. There is no user interface to it, except for an RJ−45 for network and a USB cable for programming the target.
>
> If that box is really small it would work.

My target is a lunch box (in size), but there is no reason why you can't use a PDA or cell phone (with usb). There are cell phones running on Linux and internet connected.

> I thought about a uC here after the discussion with Marte but will
> forego it again this time. This board needs good radio silence most of
> the time. The only way to achieve that within a few usec is to have an
> external clock that is gated and then it'll all become quite esoteric.

No problem, you will need a uC eventually, for another project.

Yep, and one is coming. That one must have a uC because it'll have to transmit data via wireless.

−−
Regards, Joerg

http://www.analogconsultants.com

.