

Re: I2C Questions -- Not In the FAQ

Source: <http://sci.tech-archive.net/Archive/sci.electronics.design/2004-09/2613.html>

From: Roland Zitzke (FODVRUTMSTJC_at_spammotel.com)

Date: 09/13/04

Date: Mon, 13 Sep 2004 12:23:29 +0200

Hi,

"Anthony Presley" <anthony@zoraptera.com> schrieb im Newsbeitrag
news:a55ad738.0409120803.2fa5b6a0@posting.google.com...

> *offboard sensors using I2c. However, we really would prefer to have*
> *multiple masters where possible, as some of the sensors may be "alarm"*
> *related. As far as I know, we have a couple of options.*
>

Please be aware that implementing all aspects of a multymaster bus can be a
tricky and difficult task especially if you bit-bang your signals instead of
using a hardware device.

> *1. Add a Phillips I2c hardware controller onto the existing design --*
> *as well as on each of the sensors [which will have their own MSP430's,*

So the controllers would run in slave mode while the main device is the
(multy)-master?

> *probably the cheapest of the same family]. This would allow us a I2c*
> *multi-master configuration, based on the hardware design.*
> *- In this case, how is unique addressing done? IE, we may have six*
> *of the same sensors on the bus, but want to send data to/from only*
> *one. I've read conflicting information about unique addressing, and*

When operating, there is only one master device – the device that sends the
clock.

As mentioned by other people before there are usually several address
choices available by setting / clearing certain pins in hardware.

> *2. Switch to an MSP430 which has I2c within the hardware, something*
> *like the F169. However, this is probably not desired. Similar*
> *consequences as [1] above.*
>

This is correct.

- > 3. *Continue with the bit-banging in the software. This is probably*
- > *the easiest option for me [I come from a software background].*
- > *However, I'm a little confused how this would work in a multi-master*
- > *configuration.*

If you run out of luck (i.e. hardware capabilities) it won't work at all.

There are several important tasks to consider in a multymaster environment:

- a) You need to keep track of the arbitration i.e. during the addressing phase the device writing more zeros (or is slower) wins the arbitration. This is to assure that no two masters address the bus at a time. For this you need to be able to read back the state of the i2c pins.
- b) You need to observe the bus to avoid talking during ongoing traffic. (not only during address phase)
- c) You need to define in your software what needs to be done if arbitration is lost or if the bus is busy.

If I were you I would spend a chip select line for each sensor which is controlled via a simple i2c io expander like the PCF8574.

Each sensor can implement its slave device which is only activated if its corresponding chip select is enabled.

Advantages:

- * You get by with a single-master software
- * No multiple address assignment issues.
- * easy to debug.
- * The master operates the io expander and the sensors through the same i2c bus – no additional pins are needed.
- * easy to extend to more sensors by adding a second io expander.

/Roland