

## Re: Why should I (not) use an internal oscillator for 8-bit micros

**Source:** <http://coding.derkeiler.com/Archive/General/comp.arch.embedded/2004-08/1275.html>

---

**From:** Alan ([me\\_at\\_somewhere.com.au](mailto:me_at_somewhere.com.au))

**Date:** 08/16/04

Date: Mon, 16 Aug 2004 15:49:07 +0800

On Mon, 16 Aug 2004 09:08:13 +0200, "David Brown"  
<david@no.westcontrol.spam.com> wrote:

>  
> "Neil Bradley" <nb\_no\_spam@synthcom.com> wrote in message  
> news:10hvv8scv52i489@corp.supernews.com...  
>> "Doug Dotson" <dougdotson@NOSPAMcablespeed.NOSPAMcom> wrote in message  
>> news:ismdnTKuDkd\_T4LcRVn-hQ@cablespeedmd.com...  
>> > I believe that UART stands for "Universal ASYNCRONOUS Receiver  
>> > Transmitter". You need to go back and study the difference between  
>> > sync and async.  
>>  
>> Nope, I understand the concept perfectly. When using a UART, it's required  
>> that both sides of the serial transmission be synchronized. If you don't  
>> believe me, try using a crystal at a low baud rate with a 20% tolerance.  
>> Devices won't be able to talk to it. When the byte comes is the  
> asynchronous  
>> part, and that wasn't even the topic being discussed.  
>>  
>  
> The phrase "when you are in a hole, stop digging" springs to mind...  
>  
> In the context of serial communications, "synchronous" means there is a  
> clock line (or an "embedded clock" in the data signal) going between the  
> sender and receiver, while "asynchronous" means there is no directly shared  
> clock, so each side uses its own time reference to clock the transfers. A  
> "clock" signal does not have to be regular – for synchronous transfers, it  
> can vary as much as you want. For asynchronous transfers, you normally have  
> a regular clock – while you could theoreticly have a varying one, it would  
> be a challenge to implement reliably. But if the clocks on the two sides  
> are not joined directly to synchronize their clocks, they are not  
> synchronous – it doesn't matter if their speeds match at 0.00001% accuracy.  
>  
> For standard uart asynchronous communication, the limit for communication  
> over a good line (noise-free, and nice sharp edges) is a 5% match between  
> sender and receiver, so that by the 10th bit they are no more than 50% of a

comp.arch.embedded: Re: Why should I (not) use an internal oscillator for 8-bit micros

>bit in difference. Normally, that means ensuring that each side is within  
>2.5% of the nominal baud rate. The absolute baud rate does not matter.

>

>

>

>

>

>> The question was in reference to the baud rate generating clock, not when

>> the data comes in. For the period of the byte transmission, both sides

>must

>> be synchronized. There is no common clock between them. If you have

>separate

>> clock and data lines, the clock can vary wildly with no adverse effect on

>> communication. No synchronization between devices needed. Is this a hard

>> concept to grasp?

>>

>> You do know that the words synchronous and asynchronous can mean different

>> things depending on the context, right?

>>

>> -->Neil

>>

>>

>

It's no wonder that there are so many framing errors on Async comms when some people don't understand the basic concepts of Async and Sync comms.

Sync transmits the clock either directly (on a clock line) or indirectly (embedded in the data) so that the receiver know when to look at the data bits.

Async uses the "start" bit of each byte to tell the receiver to start timing to look at the bits of this one byte only.

The term Asynchronous here means that the sender can send data at anytime without having to worry about whether or not the receiver is in sync.

How many designers try to send or receive at say 5% tolerance instead of aiming for as close to 0% as possible. Makes for interesting interfacing when one unit is 5% high and the other 5% low because they've been designed by "prima donnas" who know better than the rest of the world.

As for crystal versus baud rate tolerances – if the crystal is 1% low then the baud rate is 1% low. Simple as that – always allowing for the fact that you are using the "correct" frequency crystal to start with and allowing for any division errors in the baud rate generator/clock.

Alan

Re: Why should I (not) use an internal oscillator for 8-bit micros

comp.arch.embedded: Re: Why should I (not) use an internal oscillator for 8-bit micros

++++  
Jenal Communications  
Manufacturers and Suppliers of HF Selcall  
P O Box 1108, Morley, WA, 6943  
Tel: +61 8 9370 5533 Fax +61 8 9467 6146  
Web Site: <http://www.jenal.com>  
e-mail: <http://www.jenal.com/?p=1>  
++++