

## Re: A simple USB question

---

*Source:* <http://coding.derkeiler.com/Archive/General/comp.arch.embedded/2007-06/msg00302.html>

---

- *From:* "Boudewijn Dijkstra" <boudewijn@xxxxxxxx>
  - *Date:* Fri, 08 Jun 2007 10:55:10 +0200
- 

Op Fri, 08 Jun 2007 10:31:06 +0200 schreef Mouarf <toto@xxxxxx>:

Le Wed, 06 Jun 2007 16:43:29 +0200, Boudewijn Dijkstra <boudewijn@xxxxxxxx> a écrit:

Op Wed, 06 Jun 2007 16:12:23 +0200 schreef Mouarf <toto@xxxxxx>:

Le Wed, 06 Jun 2007 16:00:37 +0200, Boudewijn Dijkstra <boudewijn@xxxxxxxx> a écrit:

Op Wed, 06 Jun 2007 13:56:00 +0200  
schreef vasu <vasu.sharma@xxxxxxxx>:

Hi! I have an  
AT91SAM7S-EK kit with  
an AT91SAM7S256  
microcontroller.  
This microcontroller  
supports 4 Endpoints – EP0,  
EP1, EP2 and EP3. I  
am using the USB to  
communicate with a legacy  
application running on  
the PC. This application can  
only communicate on a  
COM port. So, I use  
the USB to provide a virtual  
COM port on the PC with  
which this  
application can  
communicate. My question  
is – Can I create more than  
1  
virtual COM port on the PC  
with 1 USB connection?

If your legacy driver can only recognize a

Re: A simple USB question

single COM port per USB device, you could create a 'compound device', which acts to the outside as a USB-hub.

what about a composite device with 2 COM ports? Is it possible?

compound == composite

I've read in "USB Complete" that compound=1 device + hub and composite=1 device with multiple functions I may have badly understood

My copy of the USB 2.0 standard says:

"However, a physical package may implement multiple functions and an embedded hub with a single USB cable. This is known as a compound device."

and

"From the host's perspective, a compound device is the same as a separate hub with multiple functions attached."

So my "==" signifies an equality in behaviour as observed by the host. Which is the only thing that matters here, IMHO.

--

Gemaakt met Opera's revolutionaire e-mailprogramma: <http://www.opera.com/mail/>

.