

## rtl-sdr - Bug #3589

### Zerocopy Buffers Unusable Outside Callback Without memcpy()

09/25/2018 12:58 PM - xloem

<b>Status:</b> New	<b>Start date:</b> 09/25/2018
<b>Priority:</b> Normal	<b>Due date:</b>
<b>Assignee:</b>	<b>% Done:</b> 0%
<b>Category:</b>	
<b>Target version:</b>	
<b>Description</b>	
<p>1. In May 2018, rtl-sdr was improved to work directly with DMA sample buffers <a href="http://git.osmocom.org/rtl-sdr/commit/?id=a854ae8b48d42e8dad514c75d3a4c6cfb62707da">http://git.osmocom.org/rtl-sdr/commit/?id=a854ae8b48d42e8dad514c75d3a4c6cfb62707da</a> There is now no need for a memcpy() call if all processing is done in the callback function.</p> <p>2. Unfortunately, major uses of librtlsdr process the data in another thread rather than the callback function. This is how gr-osmosdr and soapyrtl function.</p> <p>3. Additionally, tracking buffer backlog requires returning from the callback rapidly, so that buffers can be counted as they are received. This is how gr-osmosdr behaves, outputting 'O' when processing is too slow: <a href="http://git.osmocom.org/gr-osmosdr/tree/lib/rtl/rtl_source.c.c#n307">http://git.osmocom.org/gr-osmosdr/tree/lib/rtl/rtl_source.c.c#n307</a> This lets the user of a high-level library know when latency is corrupting their stream, and currently requires the use of an extra memcpy() call.</p> <p>4. These high-level libraries could take advantage of the DMA buffer improvement if there were a way to get buffers to stay alive after return from the callback.</p> <p>After managing to listen to some concerns on the chat, I've posted an updated patch to the mailing list for this: <a href="http://lists.osmocom.org/pipermail/osmocom-sdr/2018-September/001830.html">http://lists.osmocom.org/pipermail/osmocom-sdr/2018-September/001830.html</a></p>	