

OpenRTM-aist (C++) - バグ #706

OpenRTM-aist(C++)-v1.0.0-RC1 PublisherPeriodic.cppの不具合

2009/06/15 13:39 - 匿名ユーザー

ステータス:	終了	開始日:	2009/06/05
優先度:	通常	期日:	
担当者:		進捗率:	100%
カテゴリ:		予定工数:	0.00時間
対象バージョン:			
説明			
PublisherPeriodic.cppのpushFifo()関数・pushSkip()関数で、BUFFER_FULLの際、出力データが抜ける不具合あり。			

履歴

#1 - 2009/06/15 13:44 - 匿名ユーザー

- ファイル PublisherPeriodic.cpp.diff を追加
- ステータスを新規から解決に変更
- 進捗率を0から100に変更

テスト条件

exsampleのSeqIOよりSeqIn/Outを使用し、以下の手順で操作した。

1. SeqOutから連番1~出力するようコンパイル。
2. SeqOut/InをEclipseに配置、ポート接続、All Activate。
(type:periodic、rate:1~10、m_skipn:9)
3. SeqInのDeactivate -> Activate を繰り返す。
この時、SeqIn出力の連番表示を確認。

pushFifo()関数の不具合現象

- ・SeqInのDeactivate -> Activate時に、BUFFER_FULLとなったデータが表示から抜けている。

原因

- ・put()エラー時に、m_buffer->advanceRptr()を実行している為、読み出しポインタがずれていた。

対処

- ・PublisherPeriodic.cpp pushFifo()関数で、put()エラー時にreturnするよう修正。

修正前 {

```
RTC_TRACE("pushFifo()");
const cdrMemoryStream& cdr(m_buffer->get());
ReturnCode ret(m_consumer->put(cdr));
m_buffer->advanceRptr();

return ret;
}
```

修正後 (不具合調査当初からの修正分全て) {

```
RTC_TRACE("pushFifo()");
//追加部位 --->
if (m_buffer->empty() && !m_readback) {
    RTC_DEBUG("buffer empty");
    return BUFFER_EMPTY;
}
//追加部位 <---
const cdrMemoryStream& cdr(m_buffer->get());
ReturnCode ret(m_consumer->put(cdr));
//追加部位 --->
if (ret != PORT_OK) {
    return ret;
}
//追加部位 <---
m_buffer->advanceRptr();
```

```
return ret;
}
```

動作確認

- ・テスト条件で操作し、データ表示抜けがない事を確認した。

pushSkip()関数の不具合現象

- ・SeqInのDeactivate -> Activate時に、BUFFER_FULLとなったデータが表示から抜けている。

原因

- ・BUFFER_FULLから復帰してput()が正常に行われた際、読み出しポインタがずれていた。

対処

- ・PublisherPeriodic.cpp pushSkip()関数の、ロジックを見直した。

修正前 {

```
static int leftskip; // 残りのスキップ数
```

```
RTC_TRACE(("pushSkip()");
```

```
    int preskip(m_buffer->readable() - leftskip);
```

```
if (preskip < 0)
```

```
{
```

```
    m_buffer->advanceRptr(m_buffer->readable());
```

```
    leftskip = -preskip;
```

```
    // this causes empty() == true
```

```
}
```

```
else
```

```
{
```

```
    m_buffer->advanceRptr(preskip);
```

```
}
```

```
if (m_buffer->empty())
```

```
{
```

```
    RTC_DEBUG(("buffer empty");
```

```
    return BUFFER_EMPTY;
```

```
}
```

```
const cdrMemoryStream& cdr(m_buffer->get());
```

```
    ReturnCode ret(m_consumer->put(cdr));
```

```
int postskip(m_skipn - m_buffer->readable());
```

```
if (postskip < 0)
```

```
{
```

```
    postskip; =
```

```
    >advanceRptr(m_buffer->readable());
```

```
}
```

```
else
```

```
{
```

```
    m_buffer->advanceRptr(m_skipn);
```

```
}
```

```
return ret;
```

```
}
```

修正後（不具合調査当初からの修正分全て） {

```
static int leftskip; // 残りのスキップ数
```

```

RTC_TRACE(("pushSkip()"));
//修正部位 --->
    if (m_buffer->empty() && !m_readback)
    {
        RTC_DEBUG(("buffer empty"));
        return BUFFER_EMPTY;
    }

ReturnCode ret(PORT_OK);
    int preskip(m_buffer->readable() + leftskip);
    int loopcnt(preskip/(m_skipn +1));
    int postskip(m_skipn - leftskip);
    for (int i(0); i < loopcnt; ++i)
    {
        m_buffer->advanceRptr(postskip);
        const cdrMemoryStream& cdr(m_buffer->get());
        ret = m_consumer->put(cdr);
        if (ret != PORT_OK)
        {
            m_buffer->advanceRptr(-postskip); // 読み出しポインタを戻す
            return ret;
        }
        postskip = m_skipn +1;
    }

m_buffer->advanceRptr(m_buffer->readable());
    leftskip = preskip % (m_skipn +1);
//修正部位 <---

return ret;
}

```

動作確認

- ・テスト条件で操作し、データ表示抜けがない事を確認した。
- ・スキップ数単位でデータ（10の倍数）が表示されることを確認した。

以上

#2 - 2009/06/15 13:46 - 匿名ユーザー

- ステータスを 解決 から 終了 に変更

ファイル

PublisherPeriodic.cpp.diff	1.95 KB	2009/06/15	匿名ユーザー
----------------------------	---------	------------	--------