

ステータス:	終了	開始日:	2009/09/25
優先度:	高め	期日:	
担当者:	n-ando	進捗率:	100%
カテゴリ:		予定工数:	0.00時間
対象バージョン:			
<b>説明</b> OpenRTM-1.0.0-RC1-C++を使っていますが、データポート型のデータ(例えばTimedDooubleSeqなど)にある「tm.sec」「tm.nsec」について、0.4.2では空だったものが、1.0.0ではCPU時刻が自動で付与されているようです。  仕様でしょうか？  実時間で処理する際、特にロボット実機で処理する際には何ら不都合はないのですが、OpenHRP3のようにシミュレータを使う際に、シミュレーション時刻の取得が出来ないので不都合が生じてます。  ちなみに、自分で自作RTCのデータ送信元にてtm.secなどの値を入れて送信すると、受信側では、CPU時刻に上書きされているようです。  どのタイミングでCPU時刻がつけられているのでしょうか？ 複数台PCIに分散したRTCでNW上でやりとりする場合、もしPCの時刻がNTPなどで一致させていない場合、どうなるのでしょうか？  自動で、どのタイミングでしているか分かりませんが、0.4.2では、よく受け取り側でデータ順が入れ替わって取得することがありました。 その際、自作RTCで付与したtimestampにてソートをして正確性を保ってましたが、データの入れ替わりが1.0.0系でも起きる場合の弊害などが懸念される。  自動付与、手動付与の選択が可能、もしくは、手動付与がされていない場合（tmがnullの時）だけ自動付与などはどうでしょう。  とりあえずの報告です。			

**履歴**

#1 - 2009/09/25 15:39 - 匿名ユーザー

- 優先度を通常から高めに変更

新たな問題が見つかりました。「バグ」です。

テスト用RTCを3つ用意し、RTCa->RTCb->RTCcとデータポートでやりとりし、RTCaでテスト用のデータに内部でtimestampを付与したものを送信するテストを実施。

RTCbでRTCaからのデータを受信すると、内部で付与したtimestampは無視されCPU自国が付与されている。

これをそのまま内部でコピーしたものをRTCcに送信すると、本来変更されてはならないtimestamp値が、RTCcで受信した際には新たなCPU時刻になっている。

つまり、唯一のデータの時刻データが、例えばスルーさせるだけのRTCなどを挟んだ際に、その時の送信時刻で再書きされている。

単純に送信時にCPU時刻を付与するというロジックでは、問題あり。

#2 - 2009/09/28 15:53 - 匿名ユーザー

バグではなく、仕様とのこと。  
(OpenRTM-ML00961参照)

#3 - 2010/01/08 13:59 - n-ando

- ステータスを新規から終了に変更
- 担当者をn-andoにセット

タイムスタンプの自動打刻は行わないように変更し、ユーザが打刻するための関数テンプレートを新たに定義した。  
この関数テンプレートを使用するには、データポートのデータ型がメンバRTC::Time tmを持つ必要がある。

#4 - 2010/01/08 13:59 - n-ando

- 進捗率を0から100に変更