

コア - バグ #3509

Peiorid と New のパフォーマンスとロックの調査と見直し

2016/04/11 13:35 - n-ando

ステータス:	新規	開始日:	2016/04/11
優先度:	通常	期日:	
担当者:		進捗率:	0%
カテゴリ:		予定工数:	0.00時間
対象バージョン:			
説明			
https://github.com/fkanehiro/hrpsys-base/issues/552#issuecomment-207982877			
理論的には Newの方がPeriodicよりパフォーマンスは良いはずだが、そうはなっていないようなので、スレッド、ロック回りを再度見直す。 ありがとうございます。			
(1) 両方とも非リアルタイムスレッドです。それ故、periodicの周期の正確さは保証されません。 (2) newだと、onExecute() がバッファに書き込み後、signalでpublisherスレッドを起こすので、publish作業(正確には、publisherスレッドがバッファからデータを取り出す時間)が 4ms-2ms (250Hz-500Hz) 以内に終われば、次のonExecute()の書き込みを妨げない。ゆえに、ロックを取り合うことはほとんどない。 periodic の場合は、仮に onExecute()が500Hzで、publisherが50Hzとすると、publisherが1回送信して待っている間に、onExecute()は10回書き込み、デフォルトバッファ長8のままだとを超えるので、すぐにバッファフル状態になります。(バッファ長が有限長ならいずれフル状態になる。) ロックを取り合う確率がperiodicが高いのは、publisherの実行タイミングとonExecute (のwrite部分) の実行タイミングに関して、 newであれば、onExecuteのwriteのタイミングでpublisherが呼ばれるので、ほぼpublisherとonExecuteは同期がとれてることになる。そのため、概ね猶予は4msくらいはあるので、送信がこれまでに終わって入れれば大丈夫 periodicであれば、publiserの呼ばれるタイミングはonExecuteとは同期がとれてないことになる (publisherが非実時間スレッドなため)。そのため、送信時間がたとえ短くても、onExecuteのタイミングと周期実行時のpublisherの実行タイミングが近いと、ロックを取り合う確率が高くなる。 ということでしょうか。			
少し前にこちらで試してみた現象としては、実時間制御周期250[hz]で 実時間制御RTC < = > 外部のRTC の通信を行ったときに、 subscription_typeがnewで、実時間制御RTCが4msでまわらず、実時間をまもれてない subscription_typeがperiodicで、push_rateを50[hz]とすると実時間制御RTCが4[ms]を守れている のようでした。 もしかしたら送信時間が実時間制御側の周期をこえてしまっているのかもしれない。			
@n-andoさんにお教えいただいている条件ですと、 publisherスレッドの送信にかかる時間を現状の構成になってからこちらで測定してないのでしょうかと思うのですが、 現状の条件は 周期250hz 接続しているデータポートのポート数は29個 上記29個の総計データ量は2412バイトほどで、型での内訳はdoubleが254個、longが 93個、boolが 8個 (TimedXXXなど含め) となっており、データ数・ポート数が多いのかもしれない。			
ちなみに、データポートの通信で、同じデータ量でも 小さいデータ型をたくさんのポートで通信 大きいデータ型を少ないポートで通信 を比較すると、後者の方がよかったりすることもありますでしょうか。 データ自体のオーバーヘッドもありそうですが、今回のケースのように データポート1個につき1つのpublisherスレッドなので、 後者のほうがスレッドのmutexにかかるタイミングもへったりしますでしょうか。			