RETDAT Support at 10 Hz

Refined reply period logic

Wed, Nov 26, 2003

The logic for RETDAT protocol support has always assumed that systems operate at 15 Hz. But there are a few IRMs that operate at rates slower than that, especially those node in A0 that run at 10 Hz. This note describes changes made to produce replies at the proper rate for nodes that operate at rates other than 15 Hz.

Introduction

The RETDAT protocol used for data acquisition by Acnet clients specifies a reply period in units of 60 Hz. A request for 15 Hz replies specifies a value of 4; a request for 1 Hz replies specifies a value of 60. The original logic in RETDAT support assumed that the number of cycles between replies is merely the number of 60 Hz cycles divided by 4, which is exactly right if a cycle is 1/15 second. But if a cycle were 10 Hz, the 1 Hz replies would occur every 1.5 seconds. this was not a problem before now, since the Acnet clients (Vaxes) were not gauging whether replies occurred at the specified rate. But the new Java-based software watches for this, and it was noticed that some nodes seemed to be tardy with their replies, especially those operating at 10 Hz.

Changes made

The RETDAT support is handled by the ACREq module. There are four parts of that code where the assumption is made about 15 Hz operation. A new (small) function was written to assist with these changes, called FTDCYCL. Its single argument is a count of 60 Hz cycles, and its result is a count of local cycles of whatever rate. It makes use of the global variable CYCLNGTH, which is a byte whose values is the length of the last operating cycle in units of half milliseconds. Since this value is obtained by differencing two samples of a 2000 Hz counter, jitter is usually seen in the value. Empirically, it was seen that a good approximation to converting from 60 hz units to local cycle units is:

```
cycles = sixty*34/CYCLNGTH
```

where sixty is the count of 60 Hz cycles, and cycles is the result count in local cycles. But for suitably long periods, which mean large values of the 60 Hz count, the result may be different as the value of CYCLNGTH fluctuates. If the data is being collected through a server node, this opens up the possibility that the result could vary slightly as CYCLNGTH varies. In order to make sure this logic does not cause problems for the present 15 Hz nodes, a special check is made for 15 Hz operation, which is defined as having values of CYCLNGTH in the range 132–135. (The usual values seen are, of course, 133–134.) This special range check has also been used in IntsFP to detect 15 Hz to update the BCD format time-of-day, which is always in 15 Hz units. If 15 Hz operation is so indicated, the 60 Hz count is merely divided by 4, as always. With any other value, the formula is used.

For 10 Hz operation, we might see CYCLNGTH values of 200 and 201, say. For a comparatively slow 10 second reply period, the result of using the formula will be 102 and 101. If a data server node is used, the server node might therefore compute a different reply period than will the target node. And this will result in errors being reported every 10 seconds, in this case, which could be quite annoying. It is probably best not to use a server node for nodes not operating at 15 Hz. But the special check will avoid this problem for any nodes running at 15 Hz. And for more typical rates such as 1 Hz, there should not be such a problem in any case.