

adinstbpm - Task #17724

Milestone # 17719 (New): Booster operational test of one crate alongside existing BPM system

Task # 17720 (Work in progress): Booster BPM ACNET Interface

ACNET device BPM Turn-by-Turn Scaled Positon

09/15/2017 11:20 AM - John Diamond

Status:	Work in progress	Start date:	09/15/2017
Priority:	Normal	Due date:	
Assignee:	John Diamond	% Done:	90%
Category:		Estimated time:	4.00 hours
Target version:		Spent time:	5.50 hours
Description			
Implement Booster BPM ACNET device 0x0015			

History

#1 - 09/19/2017 04:12 PM - John Diamond

- Status changed from Assigned to Work in progress

#2 - 09/19/2017 10:05 PM - John Diamond

- % Done changed from 0 to 70

- Estimated time set to 4.00 h

Developed a device for returning positions from the most recent turn-by-turn measurement according to the specification given for the existing Booster BPM system. A couple of things need to be resolved still:

- The 'data' and 'time' fields are specified as 'Int4' fields but it's not clear what this really means. Will need clarification from Bill Marsh. For now these fields return '0' and '1' respectively.
- Not clear to me what the 'cycle_type' field is meant to contain. For now it always returns '4'.
- The maximum number of positions we can return (as 4-byte floats) is 8187 (TBT_POSITION_ARRAY_SIZE). Will need clarification from Bill Marsh how the existing system returns the full 20000 turns. Note that the n_samples field contains the actual number of samples contained in the measurement and can be > 8187.
- This device was specified with no reading PDB and an element size of 1, which means that the console application is probably expecting big-endian data but we are returning little-endian data because the BBPMTS processor is an x86.

Created a test device on BBPMTS:

- Z:VTS900

Control and Basic Status properties still need to be implemented.

#3 - 10/18/2017 09:26 AM - John Diamond

- % Done changed from 70 to 60

Excerpt of an e-mail from Bill that mentions undocumented fields in the TBT header:

```
For the turn by turn data Sharon seems to not have documented the inclusion of the GPS timestamp The data structure is as follows
```

```
Int4 micro date
Int4 micro timestamp
Int4 GPS sec (UTC time)
Int4 GPS nsec (nanosec from GPS sec)
Int4 number of turns
Int4 cycle type
Flt4 turn position data
```

```
Again the positions data iieee floats are word swapped but not byte swapped while the header data is used as is.
```

The full e-mail, which describes endianness and timestamp encoding is documented in [#17720](#).

#4 - 10/18/2017 08:10 PM - John Diamond

- % Done changed from 60 to 80

Filled in the missing fields per Bill's e-mail. Also added the gps time stamp fields to the device. Did not test yet.

#5 - 10/19/2017 02:40 PM - John Diamond

GPS time fields look correct. Implemented basic status and control properties but all these do is toggle an internal variable for now. They will need to be attached to the Booster BPM state machine once it is implemented.

#6 - 10/19/2017 02:40 PM - John Diamond

- % Done changed from 80 to 90