

## **Repetitive Flash Circular Buffer For the Recycler Ring BPM Front-end**

The Recycler BPM front-end provides a history data buffer containing Flash position, intensity and status data collected around the “sudden beam loss” Tclk \$F9 trigger event. The history buffer works by storing Repetitive Flash measurements, in circular buffer fashion and at the Repetitive Flash sample rate, until a given number of samples after the trigger event occurs. The history buffer holds 4096 samples, so with a maximum sample rate of 100 Hz it can contain over 40 seconds of data.

The Tclk trigger event is specified with ACNet device R:BPxBLT (for Beam Position house x Beam Loss buffer Trigger) which can take on the value of any valid Tclk event. The default value is \$F9, but the value will be restored to its most recent setting value upon reboot.

The Number of data samples retained from before the trigger event is specified with ACNet device R:BPxBLC (for Beam Position house x Beam Loss buffer Control) which can take any value from zero through 4095. This value will also be restored to its most recent setting value upon reboot. The index of the buffer element (1..4096) containing the first post trigger data is available in the reading property of the R:BPxBLC device. When the buffer is spinning this value will read as zero. The buffer can be restarted at any time (whether it is spinning or not) by sending a reset to the basic control property of the R:BPxBLC device.

The buffer data can be read in temporal order from ACNet device R:BPxBLD (for Beam Position house x Beam Loss buffer Data) using the ReadoutSpecification based data readout protocol.<sup>1</sup> The ReadoutSpecification's \_beginTurn element is used to specify the data buffer sample number of interest in the range one through 4096 with entry one being the oldest. With this scheme setting the buffer control device R:BPxBLC to  $n$  means that the first post trigger data will be in the  $n^{\text{th}}+1$  element of the buffer. The structure of each buffer element is identical to that of the BPM's Repetitive Flash data. It is an error to try to read the buffer when it is spinning.

---

<sup>1</sup> The Recycler ReadoutSpecification based data readout protocol is described Appendix A of “Event Driven Data Acquisition for the Recycler Ring BPM Front-end.”

The ACNet device R:BPxBL1 (for Beam Position house x Beam Loss buffer 1 channel data) provides a channel oriented view of the buffer data. This device uses the same ReadoutSpecification ACNet device but with a slightly different definition. Here the ReadoutSpecification's `_channel` element is used to specify the channel number of interest in the range zero through 47. The structure of the R:BPxBL1 data is as follows:

```
// BeamLossChannel circular buffer data structure
class BeamLossChannel : private DataHeader {

    public:

        class ChannelData {          // BeamLossChannel channel data structure
        public:
            long int      _milliseconds;    // time relative to trigger
            float         _position;       // position - mm
            float         _intensity;      // intensity - sum
            unsigned long int _bpmStatus;   // intensity status
        } __attribute ((packed));

    protected:

        unsigned long int  _elements;      // always kNumCircBufIndexs
        class ChannelData  _data[ kNumCircBufIndexs ];

    } __attribute ((packed));
```

The data header is identical to that used for the BPM's Repetitive Flash data and is taken from the first buffer sample after the trigger event. It is an error to try to read the buffer when it is spinning.

The ReadoutSpecification associated with both buffer data devices is ACNet device R:BPxBLS (for Beam Position house x Beam Loss buffer readout Specification) and has the following definition:

```
// structure sent by ACNet for R:BPxBLD & R:BPxBL1 data reads
class ReadoutSpecification {
    eEventIndex      _eventIndex;
    eDataType        _dataType;
    eTurnNumber      _beginTurn;
    eTurnCount       _numTurns;
    eDataChannel     _channel;
};
```

For R:BPxBLD data the ReadoutSpecification fields have the following definition:

`_eventIndex = kEventRepetitive (1)`  
`_dataType = one of the data types listed below (0..10)`  
`_beginTurn = the desired sample number (1..4096 with 1 the oldest)`  
`_numTurns = 0`  
`_channel = 0`

For R:BPxBL1 data the ReadoutSpecification fields have the following definition:

`_eventIndex = kEventRepetitive (1)`  
`_dataType = one of the data types listed below (0..10)`  
`_beginTurn = 0`  
`_numTurns = 0`  
`_channel = the desired channel number (0..47)`

```
// legal range of _dataType member
typedef enum {
    kDataTypeMin = 0,
    kBunchedData = kDataTypeMin,
    kBunch1Data,
    kBunch2Data,
    kBunch3Data,
    kBunch4Data,
    kHotData,
    kHotHeadData,
    kHotTailData,
    kColdData,
    kColdHeadData,
    kColdTailData,
    kDataTypeMax = kColdTailData,
    kNumDataTypes,
    kDataTypeDefault = kBunchedData
} eDataType;
```

End.