Pds Encoder

Pds::Encoder Namespace Reference

Class ConfigV1

Public Member Functions:

```
 ConfigV1() {}
 ConfigV1( uint32_t chan_num,
           uint32_t count_mode,
           uint32_t quadrature_mode,
           uint32_t input_num,
           uint32_t input_rising,
           uint32_t ticks_per_sec );
 ~ConfigV1() {}

void dump() const;
```

Static Public Member Functions:

```
 static Pds::TypeId typeId()
    { return TypeId( TypeId::Id_EncoderConfig, Version ); }
```

Public Attributes:

```
uint32_t _chan_num;
uint32_t _count_mode;
uint32_t _quadrature_mode;
uint32_t _input_num;
uint32_t _input_rising;
uint32_t _ticks_per_sec;
```

Class DataV1

Public Member Functions:

```
DataV1()
DataV1( uint32_t timestamp, uint32_t count )
int value() const
```

Static Public Member Functions:

```
static Pds::TypeId typeId()
```

Public Attributes:

```
uint32_t _33mhz_timestamp;
uint32_t _encoder_count;
```

Classes:

```
struct count_mode;        // enum: WRAP_FULL,LIMIT,HALT,WRAP_PRESET,END
struct quad_mode;         // enum: CLOCK_DIR,X1,X2,X4,END
```

Class DataV2

Public Member Functions:

```
DataV2()
DataV2( uint32_t timestamp, uint32_t count0, uint32_t count1, uint32_t count2)
DataV2( uint32_t timestamp, uint32_t count0)

int value() const
```

Static Public Member Functions:

```
static Pds::TypeId typeId()
```

Public Attributes:

```
uint32_t _33mhz_timestamp;
uint32_t _encoder_count[3];
```

Classes:

```
struct count_mode;        // enum: WRAP_FULL,LIMIT,HALT,WRAP_PRESET,END
struct quad_mode;         // enum: CLOCK_DIR,X1,X2,X4,END
```