



```

    该结构体包括了将定时器数据转换为SENT的半字节值时，数据处理过程中的数据和状态
    SENT_TIM_DataProcessType
    SENT_TIM_DataProcess_FastData
    typedef struct
    {
      uint8_t NibbleVal[SENT_TIM_MAX_NIBBLE_DATA + 2];
      uint8_t NibbleIdx;
      uint8_t NibbleStream;
      uint32_t TimerVal;
      uint32_t CalibrationPulseCcr;
      SENT_TIM_StatusFastProcessType FastState;
      SENT_TIM_StatusType MsgState;
      SENT_TIM_ErrorStatusType MsgErrState;
      SENT_TIM_DataProcessType;
    }
  
```

```

    该结构体包括了处理后符合SENT协议的报文数据
    SENT_TIM_FastMsgType
    SENT_TIM_ReceiveFastMsg
    typedef struct
    {
      uint32_t Tick;
      uint32_t Length;
      uint8_t DataNibble[SENT_TIM_MAX_NIBBLE_DATA];
      uint8_t StatusCommunication;
      uint8_t FastCrc;
      bool MsgJudge;
    }
  
```

**STATUS\_SENT\_TIM\_FAST\_SYNC\_CALIB部分功能**

1. 校验接收的Nibble长度是否符合  $\leq 27$  or  $\geq 12$
2. 获取完整报文长度，单位为Tick
3. 软件CRC校验，校验获取报文的正确性
4. 将成功接收的报文数据保存到指定结构体

**STATUS\_SENT\_TIM\_FAST\_IDLE部分功能**

1. 同步SYNC，在未同步到SYNC前，所有采集数据会被忽略
2. 获取实际SYNC长度，误差 $\pm 25\%$
3. 检测连续SYNC脉冲误差 $> 1.5625\%$ ，并标记报文false/true
4. 计算实时SENT Tick值

**STATUS\_SENT\_TIM\_FAST\_PAUSE\_PULSE部分功能**

1. 获取带PAUSE长度的完整报文长度

