**数据存储通用平台**

实现方法说明（V1.0版）

附表：修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 版本 | 修订内容 | 修订日期 |
| 01 | V1.0 | 初次编订 | 2015-11-04 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1 实现背景 3](#_Toc434424329)

[2 实现目的 3](#_Toc434424330)

[3 应用范围 3](#_Toc434424331)

[4 实现概述 4](#_Toc434424332)

[4.1 数据属性 4](#_Toc434424333)

[4.2 属性归纳 4](#_Toc434424334)

[5 实现方案 4](#_Toc434424335)

[5.1 实现过程 4](#_Toc434424336)

[6 实现方法 5](#_Toc434424337)

## 1 实现背景

经过研发能耗在线监测装置，温度面积法热计量分摊装置，计量间仪表管控装置过程中，发现目前我们系统的应用对硬件底层的应用主要为通信（RS232，RS485，M-BUS，GPRS）和存储（SDCARD，NandFlash，FRAM，SPI FLASH）两方面。其中针对不同项目的数据存储内容和数据项大小都不尽相同，每次不同项目研发都需要根据每个需要存储的数据项内容编写每个数据项存储的存储程序，并且需要针对不同存储介质，介质的物理地址，以及存储方法的不同更改驱动程序，占用了研发时间中比较大的部分。

## 2 实现目的

经过几个项目开发过程以及参考《**采集计算器GLF-TA存储规划-V6.1版**》存储分配说明，编写《温度法热计量分摊装置数据存储分配说明》过程中归纳发现每个项目的数据存储内容虽然不同，但是都具备一些共同的特性。

本数据存储通用平台（**Universal Storage Platform**）的目的是总结已开发项目的数据存储内容共同特性和应用需求，开发一个通用的数据存储方法和接口，降低数据存储的实现复杂性，最终目的为减少后续项目研发时间，降低数据存储过程中出错几率。

## 3 应用范围

本实现主要为统一不同存储介质的不同操作方法以及不同存储数据项内容的编程接口，减少研发过程中编程时间和编写修改程序过程中出错几率。

主要应用范围为针对应用层的数据操作接口为二进制方式，包括直接操作连续存储的硬件存储介质（FRAM、EEPROM）接口和以块和页为单位读写操作（NandFlash，SPIFlash,SDCARD）等需要擦写均衡，坏块管理算法支持的上层应用接口。不支持基于文件和目录操作方式的基于文件系统的存储应用。

## 4 实现概述

### 4.1 数据属性

针对每个不同的存储数据项内容和程序接口，归纳总结发现都具备下面表格中的属性：

|  |  |  |
| --- | --- | --- |
| **序号** | **属性** | **注释** |
| 1 | 数据类型 | 系统参数，热量表数据，电表数据，室温采集器数据，压力数据等 |
| 2 | 数据条项 | 初始化数据，即时数据，周期数据，历史数据等条项，条项最后包含双重校验 |
| 3 | 设备类型 | FRAM，EEPROM，SPIFlash，NandFlash，SDCARD等 |
| 4 | 设备地址 | 0,1,2… |
| 5 | 数据起始地址 | 每种“数据类型”起始地址固定，在编程时指定。 |
| 6 | 数据条项大小 | 每个“数据条项”大小固定，在编程时指定 |
| 7 | 数据大小 | 上述“数据类型”每种类型总数据存储空间固定，在编程时指定。 |
| 8 | 设备总大小 | 对应“设备类型”、“设备地址”的总空间固定，根据需求在编程时指定。 |

表4-1 数据条项属性

### 4.2 属性归纳

§4.1 内描述的“数据属性”，表格内每个“数据类型”都含有序号2-8属性，区别只是根据不同“数据类型”序号2-8内的部分属性有区别，并且序号2-8每个属性都是在编程时确定，可以固化在MCU Flash内，所以具备编写通用的数据存储接口条件。

## 5 实现方案

### 5.1 实现过程

整体实现思想为以具体某一个设备为一个整体规划分区（设备类型+设备地址），每个分区按照不同的“数据类型”依次分配地址存储，相同“数据类型”内的所有“数据条项”按照规定顺序存放。

|  |  |  |
| --- | --- | --- |
| **设备** | **数据类型** | **数据条项** |
| 设备类型+设备地址  （FM24CL64） | 系统参数 | 系统参数 |
| 热量表 | 初始化信息 |
| 实时数据 |
| 周期数据 |
| 。。。 | 。。。 |
| 。。。 | 。。。 |

表5-1 设备分区存储示例

1）预先规划该**项目设备**需要存储的所有数据类型（系统参数，热量表，通断表，室温采集器，压力传感器，电能表等）。

2）对每种**数据类型**的具体存储内容进行分类，比如将所有“系统参数”放置在一起，按照一个“数据条项”存储，“热量表”一般分为初始化信息、实时数据、周期数据等

3）计算项目中每个“**数据类型**”所需要存储的具体数量，如需要存储的热量表数据数量，然后对每个“**数据条项**”内容和大小进行规划。

4）最后根据《表4-1 数据条项属性》依次列出每个数据条项内容，根据此内容，用户在读写具体条项时，只需要提供预先定义的**“数据类型”+”数据条项”+”数据索引”**即可按照《表4-1 数据条项属性》内容操作具体存储区域。

5）按照规划内容，分配存储空间，每个数据条项最后3字节需要预留为数据存储校验使用。

## 6 实现方法

按照以上归纳总结，参考之前项目研发的具体存储实现，已在MbusGate网关项目中具体应用此方法，下面为具体实施内容，共分为4个部分：

1）应用层配置文件record\_cfg.h

2）应用层接口文件record.h record.c

3）数据存储分配表record\_tbl.h record\_tbl.c

4）数据存储硬件层record\_hw.h record\_hw.c

/\*\*

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \*               Copyright(C) 2015-2025 GDKY  All Rights Reserved

  \*

  \* **@file**    : record\_cfg.h

  \* **@author**  : ZouZH

  \* **@version** : V1.00

  \* **@date**    : 04-Nov-2015

  \* **@brief**   : 数据存储通用平台相关配置和定义文件.

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \*/

/\* Define to prevent recursive inclusion --------------------------------------\*/

#ifndef \_\_RECORD\_CFG\_H

#define \_\_RECORD\_CFG\_H

#ifdef \_\_cplusplus

 extern "C" **{**

#endif  /\* \_\_cplusplus \*/

/\* INCLUDES ----------------------------------------------------------------- \*/

#include <stdint.h>

/\* TYPEDEFS ----------------------------------------------------------------- \*/

/\*\*

 \* 存储错误代码

 \*/

**typedef** enum

**{**

  REC\_OK            **=** 0**,**  /\* 执行正常 \*/

  REC\_ERR\_PARAM     **=** 1**,**  /\* 参数错误 \*/

  REC\_ERR\_CHECK     **=** 2**,**  /\* 校验错误 \*/

  REC\_ERR\_DRIVE     **=** 3**,**  /\* 底层驱动错误 \*/

  REC\_ERR\_PASSWD    **=** 4**,**  /\* 系统密码错误 \*/

  REC\_ERR\_CFG       **=** 5**,**  /\* 配置参数错误 \*/

  REC\_ERR\_OVER      **=** 6**,**  /\* 地址溢出 \*/

**}**RecordErr\_t**;**

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 以下定义需要与实际使用的 RecordTBL 内存分配表对应

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

/\*\*

 \* 数据类型(DataType)定义

 \*/

**typedef** enum

**{**

  REC\_DT\_NONE        **=** 0**,**  /\* 未指定数据类型 \*/

  REC\_DT\_SysParam    **=** 1**,**  /\* 系统参数  - System Parameter \*/

  REC\_DT\_HeatMeter   **=** 2**,**  /\* 热量表    - HeatMeter \*/

  REC\_DI\_History     **=** 3**,**  /\* 历史数据  - History \*/

  REC\_DT\_MAX**,**              /\* 最大数据类型 \*/

**}**RecordDataType\_t**;**

/\*\*

 \* 数据条项(DataItem)定义

 \*/

**typedef** enum

**{**

  REC\_DI\_NONE        **=** 0**,**  /\* 未指定数据条项 \*/

  REC\_DI\_InitInfo    **=** 1**,**  /\* 初始化信息 - Initial Information \*/

  REC\_DI\_RTimeData   **=** 2**,**  /\* 实时数据   - Real Time Data \*/

  REC\_DI\_CycleData   **=** 3**,**  /\* 周期数据   - Cycle Data \*/

  REC\_DI\_HistoryData **=** 4**,**  /\* 历史数据   - History Data \*/

  REC\_DI\_Event       **=** 5**,**  /\* 事件记录   - Event \*/

  REC\_DI\_Alarm       **=** 6**,**  /\* 报警记录   - Alarm \*/

  REC\_DI\_MAX**,**              /\* 最大数据项 \*/

**}**RecordDataItem\_t**;**

/\*\*

 \* 存储设备(Device)定义

 \*/

**typedef** enum

**{**

  REC\_DEV\_NONE        **=** 0**,**  /\* 未指定设备 \*/

  REC\_DEV\_FM24CL64    **=** 1**,**  /\* FM24CL64B-G，8KByte \*/

**}**RecordDev\_t**;**

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 以下为用户存储的数据条项定义,必须按照以下标准存储结构(即最后3字节必须为校验)

\* \_\_packed typedef struct

\* {

\*   uint8\_t  xxx;

\*   uint16\_t xxx;

\*   uint32\_t xxx;

\*

\*   uint8\_t  bcc;              // BCC校验

\*   uint16\_t crc;              // CRC校验

\*

\* }UserDataItem\_t

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

/\*\*

 \*  系统参数 Bytes

 \*/

\_\_packed **typedef** struct

**{**

  uint32\_t version**;**          /\* 版本号: xx.xx.xx.xx(BCD Code ) \*/

  // 上行串口参数

  uint16\_t upBaudrate**;**

  uint8\_t  upDatabits**;**

  uint8\_t  upParity**;**

  uint8\_t  upStopbits**;**

  // 下行串口参数

  uint16\_t dnBaudrate**;**

  uint8\_t  dnDatabits**;**

  uint8\_t  dnParity**;**

  uint8\_t  dnStopbits**;**

  // 抄表模式

  uint8\_t  collectMode**;**      /\* 01-实时抄表, 02-定时抄表 \*/

  uint8\_t  collectTimeout**;**   /\* 命令响应超时时间: 秒 \*/

  uint8\_t  collectInterval**;**  /\* 定时抄表时间间隔: 01=0.5h,02=1h,03=8h,04=12h,05=24h \*/

  // 通讯协议

  uint8\_t  upComProtocol**;**    /\* 上行协议，0001=Modbus,0002=透明传输 \*/

  uint8\_t  dnComProtocol**;**    /\* 下行协议，0001=CJ/T188,0002=EN1434,0003=透明传输 \*/

  uint8\_t  gateAddr**;**         /\* 网关地址 \*/

  // 热量表信息

  uint8\_t  heatMeterNbr**;**     /\* 热量表数量 \*/

  uint8\_t  heatMeterTbl**[**8**];**  /\* 热量表位置表 \*/

  uint8\_t  bcc**;**              /\* BCC校验 \*/

  uint16\_t crc**;**              /\* CRC校验 \*/

**}**SysParam\_t**;**

/\*\*

 \*  热量表信息 Bytes

 \*/

**typedef** struct

**{**

  uint8\_t  index**;**

  uint8\_t  protocol**;**

  uint16\_t baudrate**;**

  uint8\_t  databits**;**

  uint8\_t  parity**;**

  uint8\_t  stopbits**;**

  uint8\_t  vender**;**

  uint8\_t  addr**[**7**];**

  uint8\_t  bcc**;**              /\* BCC校验 \*/

  uint16\_t crc**;**              /\* CRC校验 \*/

**}**HeatMeterInfo\_t**;**

/\*\*

 \*  热量表数据 Bytes

 \*/

**typedef** struct

**{**

  float    instHeat**;**

  float    curCold**;**

  float    curHeat**;**

  float    instFlow**;**

  float    totalFlow**;**

  uint16\_t supplyTemp**;**

  uint16\_t backTemp**;**

  uint16\_t tempDiff**;**

  uint16\_t status**;**

  uint8\_t  bcc**;**              /\* BCC校验 \*/

  uint16\_t crc**;**              /\* CRC校验 \*/

**}**HeatMeterData\_t**;**

/\* MACROS  ------------------------------------------------------------------ \*/

/\* CONSTANTS  --------------------------------------------------------------- \*/

/\*\*

 \* 用户存储参数配置

 \*/

#define RECORD\_HEATMETER\_MAX\_NBR 64  /\* 热量表最大数目 \*/

/\*\*

 \* 存储运行配置参数项

 \*/

#define RECORD\_HW\_MAX\_NBR         3  /\* 最多设备数, 0-255 \*/

#define RECORD\_ERR\_RETRY\_CNT      3  /\* 存储发生错误重试次数 \*/

/\* GLOBAL VARIABLES --------------------------------------------------------- \*/

/\* GLOBAL FUNCTIONS --------------------------------------------------------- \*/

/\* LOCAL VARIABLES ---------------------------------------------------------- \*/

/\* LOCAL FUNCTIONS ---------------------------------------------------------- \*/

#ifdef \_\_cplusplus

**}**

#endif  /\* \_\_cplusplus \*/

#endif /\* \_\_RECORD\_CFG\_H \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* END OF FILE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \*               Copyright(C) 2015-2025 GDKY  All Rights Reserved

  \*

  \* **@file**    : record.h

  \* **@author**  : ZouZH

  \* **@version** : V1.00

  \* **@date**    : 04-Nov-2015

  \* **@brief**   : 数据存储通用平台应用层接口.

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \*/

/\* Define to prevent recursive inclusion --------------------------------------\*/

#ifndef \_\_RECORD\_H

#define \_\_RECORD\_H

#ifdef \_\_cplusplus

 extern "C" **{**

#endif  /\* \_\_cplusplus \*/

/\* INCLUDES ----------------------------------------------------------------- \*/

#include "record\_cfg.h"

/\* TYPEDEFS ----------------------------------------------------------------- \*/

/\* MACROS  ------------------------------------------------------------------ \*/

/\* CONSTANTS  --------------------------------------------------------------- \*/

/\* GLOBAL VARIABLES --------------------------------------------------------- \*/

/\* GLOBAL FUNCTIONS --------------------------------------------------------- \*/

RecordErr\_t Record\_Init**(**void**);**

RecordErr\_t Record\_Read**(**RecordDataType\_t REC\_DT\_x**,** RecordDataItem\_t REC\_DI\_x**,**

uint16\_t itemIdx**,** void **\***pvBuf**,** uint32\_t len**);**

RecordErr\_t Record\_Write**(**RecordDataType\_t REC\_DT\_x**,** RecordDataItem\_t REC\_DI\_x**,**

uint16\_t itemIdx**,** void **\***pvBuf**,** uint32\_t len**);**

/\* LOCAL VARIABLES ---------------------------------------------------------- \*/

/\* LOCAL FUNCTIONS ---------------------------------------------------------- \*/

#ifdef \_\_cplusplus

**}**

#endif  /\* \_\_cplusplus \*/

#endif /\* \_\_RECORD\_H \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* END OF FILE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \*               Copyright(C) 2015-2025 GDKY  All Rights Reserved

  \* **@file**    : record.c

  \* **@author**  : ZouZH

  \* **@version** : V1.00

  \* **@date**    : 04-Nov-2015

  \* **@brief**   : 数据存储通用平台应用层接口.

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \*/

/\* INCLUDES ----------------------------------------------------------------- \*/

#include "record.h"

#include "includes.h"

#include "record\_hw.h"

/\* TYPEDEFS ----------------------------------------------------------------- \*/

/\* MACROS  ------------------------------------------------------------------ \*/

/\* CONSTANTS  --------------------------------------------------------------- \*/

/\* GLOBAL VARIABLES --------------------------------------------------------- \*/

/\* GLOBAL FUNCTIONS --------------------------------------------------------- \*/

/\* LOCAL VARIABLES ---------------------------------------------------------- \*/

/\* LOCAL FUNCTIONS ---------------------------------------------------------- \*/

/\*\*

 \* **@brief** : 存储初始化

 \*

 \* **@param** : 无

 \*

 \* @retval: 0 成功，其他失败

 \*/

RecordErr\_t Record\_Init**(**void**)**

**{**

  RecordErr\_t recErr **=** REC\_OK**;**

  // 硬件初始化

  recErr **=** Record\_HWInit**(NULL);**

**if** **(**recErr **!=** REC\_OK**)**

**{**

**return** recErr**;**

**}**

  // 读系统参数

**return** recErr**;**

**}**

/\*\*

 \* **@brief** : 读取数据

 \* **@param** : REC\_DT\_x 数据类型(1-REC\_DT\_MAX)

 \* **@param** : REC\_DI\_x 数据条项(0-REC\_DI\_MAX)

 \* **@param** : itemIdx  数据条项索引(0-65535)

 \* **@param** : pvBuf    数据缓存

 \* **@param** : len      数据长度

 \* @retval: 0 成功，其他失败

 \*/

RecordErr\_t Record\_Read**(**RecordDataType\_t REC\_DT\_x**,** RecordDataItem\_t REC\_DI\_x**,**

uint16\_t itemIdx**,** void **\***pvBuf**,** uint32\_t len**)**

**{**

  RecordErr\_t recErr **=** REC\_OK**;**

  uint8\_t  i **=** 0**;**

  uint8\_t  tbl\_idx **=** 0**;**

  uint32\_t addr **=** 0**;**

  ASSERT**(**REC\_DT\_x **&&** **(**REC\_DT\_x **<** REC\_DT\_MAX**)** **&&** **(**REC\_DI\_x **<** REC\_DI\_MAX**)** **&&** pvBuf **&&** len**);**

  // 搜索存储分配表中对应数据项

**while** **(**RecordTBL**[**i**].**dataType **&&** RecordTBL**[**i**].**devType**)**

**{**

**if** **((**REC\_DT\_x **==** RecordTBL**[**i**].**dataType**)** **&&** **(**REC\_DI\_x **==** RecordTBL**[**i**].**dataItem**))**

**break;**

**else**

      i **+=** 1**;**

**}**

  // 未搜索到有效条项

**if** **(!**RecordTBL**[**i**].**dataType **||** **!**RecordTBL**[**i**].**devType**)**

**{**

**return** REC\_ERR\_PARAM**;**

**}**

**else**

**{**

    tbl\_idx **=** i**;**

**}**

  // 计算地址

  i **=** tbl\_idx**;**

  addr **=** RecordTBL**[**tbl\_idx**].**dataStartAddr**;**

**do**

**{**

    addr **+=** itemIdx **\*** RecordTBL**[**i**].**dataItemSize**;**

**if** **(**i **>** 0**)**

**{**

      i **-=** 1**;**

**}**

**}while** **((**i **>** 0**)** **&&** **(**RecordTBL**[**tbl\_idx**].**dataType **==** RecordTBL**[**i**].**dataType**));**

  // 判断地址范围

**if** **(**addr **+** len **>=** RecordTBL**[**tbl\_idx**].**devTotalSize**)**

**{**

**return** REC\_ERR\_OVER**;**

**}**

**for** **(**i **=** 0**;** i **<** RECORD\_ERR\_RETRY\_CNT**;** i**++)**

**{**

**if** **(**recErr **!=** REC\_OK**)**

**{**

      recErr **=** Record\_HWInit**(&**RecordTBL**[**tbl\_idx**]);**

**}**

**if** **(**Record\_HWRead**(&**RecordTBL**[**tbl\_idx**],** addr**,** pvBuf**,** len**)** **!=** REC\_OK**)**

**{**

      recErr **=** REC\_ERR\_DRIVE**;**

**continue;**

**}**

**if** **(**chk\_crc16\_MB**((**const uint8\_t **\*)**pvBuf**,**  RecordTBL**[**tbl\_idx**].**dataItemSize**))**

**{**

      recErr **=** REC\_ERR\_CHECK**;**

**continue;**

**}**

**if** **(**chk\_bcc**((**const uint8\_t **\*)**pvBuf**,** RecordTBL**[**tbl\_idx**].**dataItemSize **-** 2**))**

**{**

      recErr **=** REC\_ERR\_CHECK**;**

**continue;**

**}**

**else**

**{**

      i **=** 0x55**;**

**}**

**}**

**return** recErr**;**

**}**

/\*\*

 \* **@brief** : 写入数据

 \*

 \* **@param** : REC\_DT\_x 数据类型(1-REC\_DT\_MAX)

 \* **@param** : REC\_DI\_x 数据条项(0-REC\_DI\_MAX)

 \* **@param** : itemIdx  数据条项索引(0-65535)

 \* **@param** : pvBuf    数据缓存

 \* **@param** : len      数据长度

 \* @retval: 0 成功，其他失败

 \*/

RecordErr\_t Record\_Write**(**RecordDataType\_t REC\_DT\_x**,** RecordDataItem\_t REC\_DI\_x**,**

uint16\_t itemIdx**,** void **\***pvBuf**,** uint32\_t len**)**

**{**

  RecordErr\_t recErr **=** REC\_OK**;**

  uint8\_t  i **=** 0**;**

  uint8\_t  tbl\_idx **=** 0**;**

  uint32\_t addr **=** 0**;**

  ASSERT**(**REC\_DT\_x **&&** **(**REC\_DT\_x **<** REC\_DT\_MAX**)** **&&** **(**REC\_DI\_x **<** REC\_DI\_MAX**)** **&&** pvBuf **&&** len**);**

  // 搜索存储分配表中对应数据项

**while** **(**RecordTBL**[**i**].**dataType **&&** RecordTBL**[**i**].**devType**)**

**{**

**if** **((**REC\_DT\_x **==** RecordTBL**[**i**].**dataType**)** **&&** **(**REC\_DI\_x **==** RecordTBL**[**i**].**dataItem**))**

**break;**

**else**

      i **+=** 1**;**

**}**

  // 未搜索到有效条项

**if** **(!**RecordTBL**[**i**].**dataType **||** **!**RecordTBL**[**i**].**devType**)**

**{**

**return** REC\_ERR\_PARAM**;**

**}**

**else**

**{**

    tbl\_idx **=** i**;**

**}**

  // 计算地址

  i **=** tbl\_idx**;**

  addr **=** RecordTBL**[**tbl\_idx**].**dataStartAddr**;**

**do**

**{**

    addr **+=** itemIdx **\*** RecordTBL**[**i**].**dataItemSize**;**

**if** **(**i **>** 0**)**

**{**

      i **-=** 1**;**

**}**

**}while** **((**i **>** 0**)** **&&** **(**RecordTBL**[**tbl\_idx**].**dataType **==** RecordTBL**[**i**].**dataType**));**

  // 判断地址范围

**if** **(**addr **+** len **>=** RecordTBL**[**tbl\_idx**].**devTotalSize**)**

**{**

**return** REC\_ERR\_OVER**;**

**}**

**\*(**uint8\_t **\*)(((**uint8\_t **\*)**pvBuf**)** **+** RecordTBL**[**tbl\_idx**].**dataItemSize **-** 3**)** **=**

    chk\_bcc**((**const uint8\_t **\*)**pvBuf**,** RecordTBL**[**tbl\_idx**].**dataItemSize **-** 3**);**

**\*(**uint16\_t **\*)(((**uint8\_t **\*)**pvBuf**)** **+** RecordTBL**[**tbl\_idx**].**dataItemSize **-** 2**)** **=**

    chk\_crc16\_MB**((**const uint8\_t **\*)**pvBuf**,** RecordTBL**[**tbl\_idx**].**dataItemSize **-** 2**);**

**for** **(**i **=** 0**;** i **<** RECORD\_ERR\_RETRY\_CNT**;** i**++)**

**{**

**if** **(**recErr **!=** REC\_OK**)**

**{**

      recErr **=** Record\_HWInit**(&**RecordTBL**[**tbl\_idx**]);**

**}**

**if** **(**Record\_HWWrite**(&**RecordTBL**[**tbl\_idx**],** addr**,** pvBuf**,** len**)** **!=** REC\_OK**)**

**{**

      recErr **=** REC\_ERR\_DRIVE**;**

**continue;**

**}**

**else**

**{**

      i **=** 0x55**;**

**}**

**}**

**return** recErr**;**

**}**

/\*\*

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \*               Copyright(C) 2015-2025 GDKY  All Rights Reserved

  \*

  \* **@file**    : record\_tbl.h

  \* **@author**  : ZouZH

  \* **@version** : V1.00

  \* **@date**    : 04-Nov-2015

  \* **@brief**   : 存储分配表配置文件,包含数据类型，设备类型，地址范围等定义.

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \*/

/\* Define to prevent recursive inclusion --------------------------------------\*/

#ifndef \_\_RECORD\_TBL\_H

#define \_\_RECORD\_TBL\_H

#ifdef \_\_cplusplus

 extern "C" **{**

#endif  /\* \_\_cplusplus \*/

/\* INCLUDES ----------------------------------------------------------------- \*/

#include "record\_cfg.h"

/\* TYPEDEFS ----------------------------------------------------------------- \*/

/\*\*

 \* Record Table

 \*/

**typedef** struct

**{**

  RecordDataType\_t  dataType**;**

  RecordDataItem\_t  dataItem**;**

  RecordDev\_t       devType**;**

  uint8\_t           devAddr**;**

  uint32\_t          dataStartAddr**;**

  uint32\_t          dataItemSize**;**

  uint32\_t          dataTotalSize**;**

  uint32\_t          devTotalSize**;**

**}**RecordTBL\_t**;**

/\* MACROS  ------------------------------------------------------------------ \*/

/\* CONSTANTS  --------------------------------------------------------------- \*/

/\* GLOBAL VARIABLES --------------------------------------------------------- \*/

extern const RecordTBL\_t RecordTBL**[];**

/\* GLOBAL FUNCTIONS --------------------------------------------------------- \*/

/\* LOCAL VARIABLES ---------------------------------------------------------- \*/

/\* LOCAL FUNCTIONS ---------------------------------------------------------- \*/

#ifdef \_\_cplusplus

**}**

#endif  /\* \_\_cplusplus \*/

#endif /\* \_\_RECORD\_TBL\_H \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* END OF FILE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \*               Copyright(C) 2015-2025 GDKY  All Rights Reserved

  \* **@file**    : record\_tbl.c

  \* **@author**  : ZouZH

  \* **@version** : V1.00

  \* **@date**    : 04-Nov-2015

  \* **@brief**   : 存储分配表配置文件,包含数据类型，设备类型，地址范围等定义.

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \*/

/\* INCLUDES ----------------------------------------------------------------- \*/

#include "record\_tbl.h"

#include "fm24clxx.h"

/\* TYPEDEFS ----------------------------------------------------------------- \*/

/\* MACROS  ------------------------------------------------------------------ \*/

/\* CONSTANTS  --------------------------------------------------------------- \*/

/\*\*

 \* 存储分配表定义

 \* 1)相同dataType必须相邻,必须按照dataItem从小到大排序,dataItem为0时,代表只有一种条项;

 \* 2)一个dataType具有多个dataItem时，dataItem序号必须从大于0开始递增

 \* 3)devType和devAddr必须大于0

 \* 4)dataStartAddr为每个dataType的存储起始地址

 \* 5)dataItemSize为数据条项大小

 \* 6)dataTotalSize为每个dataType总存储大小

 \* 7)devTotalSize为devType总空间大小

 \*/

const RecordTBL\_t RecordTBL**[]** **=**

**{**

  // 系统参数

**{** **.**dataType      **=** REC\_DT\_SysParam**,**

**.**dataItem      **=** REC\_DI\_NONE**,**

**.**devType       **=** REC\_DEV\_FM24CL64**,**

**.**devAddr       **=** 0x01**,**

**.**dataStartAddr **=** 0x00000000**,**

**.**dataItemSize  **=** **sizeof(**SysParam\_t**),**

**.**dataTotalSize **=** **sizeof(**SysParam\_t**),**

**.**devTotalSize  **=** FM24CL64\_MAX\_ADDR

**},**

  // 热量表-初始化信息

**{** **.**dataType      **=** REC\_DT\_HeatMeter**,**

**.**dataItem      **=** REC\_DI\_InitInfo**,**

**.**devType       **=** REC\_DEV\_FM24CL64**,**

**.**devAddr       **=** 0x01**,**

**.**dataStartAddr **=** **sizeof(**SysParam\_t**),**

**.**dataItemSize  **=** **sizeof(**HeatMeterInfo\_t**),**

**.**dataTotalSize **=** RECORD\_HEATMETER\_MAX\_NBR **\*** **(sizeof(**HeatMeterInfo\_t**)** **+**

**sizeof(**HeatMeterData\_t**)),**

**.**devTotalSize  **=** FM24CL64\_MAX\_ADDR

**},**

  // 热量表-实时数据

**{** **.**dataType      **=** REC\_DT\_HeatMeter**,**

**.**dataItem      **=** REC\_DI\_RTimeData**,**

**.**devType       **=** REC\_DEV\_FM24CL64**,**

**.**devAddr       **=** 0x01**,**

**.**dataStartAddr **=** **sizeof(**SysParam\_t**),**

**.**dataItemSize  **=** **sizeof(**HeatMeterData\_t**),**

**.**dataTotalSize **=** RECORD\_HEATMETER\_MAX\_NBR **\*** **(sizeof(**HeatMeterInfo\_t**)** **+**

**sizeof(**HeatMeterData\_t**)),**

**.**devTotalSize  **=** FM24CL64\_MAX\_ADDR

**},**

  // 存储分配表-结束标志

**{** **.**dataType      **=** REC\_DT\_NONE**,**

**.**dataItem      **=** REC\_DI\_NONE**,**

**.**devType       **=** REC\_DEV\_NONE**,**

**.**devAddr       **=** 0x00**,**

**.**dataStartAddr **=** 0x00000000**,**

**.**dataItemSize  **=** 0x00000000**,**

**.**dataTotalSize **=** 0x00000000**,**

**.**devTotalSize  **=** 0x00000000

**},**

**};**

/\* GLOBAL VARIABLES --------------------------------------------------------- \*/

/\* GLOBAL FUNCTIONS --------------------------------------------------------- \*/

/\* LOCAL VARIABLES ---------------------------------------------------------- \*/

/\* LOCAL FUNCTIONS ---------------------------------------------------------- \*/

/\*\*

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \*               Copyright(C) 2015-2025 GDKY  All Rights Reserved

  \*

  \* **@file**    : record\_hw.h

  \* **@author**  : ZouZH

  \* **@version** : V1.00

  \* **@date**    : 04-Nov-2015

  \* **@brief**   : 数据存储通用平台底层硬件驱动.

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \*/

/\* Define to prevent recursive inclusion --------------------------------------\*/

#ifndef \_\_RECORD\_HW\_H

#define \_\_RECORD\_HW\_H

#ifdef \_\_cplusplus

 extern "C" **{**

#endif  /\* \_\_cplusplus \*/

/\* INCLUDES ----------------------------------------------------------------- \*/

#include "record\_cfg.h"

#include "record\_tbl.h"

/\* TYPEDEFS ----------------------------------------------------------------- \*/

/\* MACROS  ------------------------------------------------------------------ \*/

/\* CONSTANTS  --------------------------------------------------------------- \*/

/\* GLOBAL VARIABLES --------------------------------------------------------- \*/

/\* GLOBAL FUNCTIONS --------------------------------------------------------- \*/

RecordErr\_t Record\_HWInit**(**const RecordTBL\_t **\***ptbl**);**

RecordErr\_t Record\_HWRead**(**const RecordTBL\_t **\***ptbl**,** uint32\_t addr**,** void **\***pvbuf**,**

uint32\_t len**);**

RecordErr\_t Record\_HWWrite**(**const RecordTBL\_t **\***ptbl**,** uint32\_t addr**,** const void **\***pvbuf**,**

uint32\_t len**);**

/\* LOCAL VARIABLES ---------------------------------------------------------- \*/

/\* LOCAL FUNCTIONS ---------------------------------------------------------- \*/

#ifdef \_\_cplusplus

**}**

#endif  /\* \_\_cplusplus \*/

#endif /\* \_\_RECORD\_HW\_H \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* END OF FILE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \*               Copyright(C) 2015-2025 GDKY  All Rights Reserved

  \*

  \* **@file**    : record\_hw.c

  \* **@author**  : ZouZH

  \* **@version** : V1.00

  \* **@date**    : 04-Nov-2015

  \* **@brief**   : 数据存储通用平台底层硬件驱动.

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \*/

/\* INCLUDES ----------------------------------------------------------------- \*/

#include "record\_hw.h"

#include "includes.h"

#include "fm24clxx.h"

/\* TYPEDEFS ----------------------------------------------------------------- \*/

/\* MACROS  ------------------------------------------------------------------ \*/

/\* CONSTANTS  --------------------------------------------------------------- \*/

/\* GLOBAL VARIABLES --------------------------------------------------------- \*/

/\* GLOBAL FUNCTIONS --------------------------------------------------------- \*/

/\* LOCAL VARIABLES ---------------------------------------------------------- \*/

/\* LOCAL FUNCTIONS ---------------------------------------------------------- \*/

/\*\*

 \* **@brief** : 初始化存储分配表(RecordTBL)内所有硬件

 \*

 \* **@param** : ptbl 指定初始化分配表中设备,为NULL时初始化所有设备

 \*

 \* @retval: 0成功,其他失败

 \*/

RecordErr\_t Record\_HWInit**(**const RecordTBL\_t **\***ptbl**)**

**{**

  uint8\_t i **=** 0**;**

  uint8\_t err **=** 0**;**

**if** **(**ptbl **!=** **NULL)**

**{**

**if** **(**ptbl**->**devType **==** REC\_DEV\_FM24CL64**)**

**{**

      FRAM\_Init**();**

**}**

**}**

**else**

**{**

**while** **(**RecordTBL**[**i**].**devType **&&** **(**i **<** RECORD\_HW\_MAX\_NBR**))**

**{**

**if** **(**RecordTBL**[**i**].**devType **==** REC\_DEV\_FM24CL64**)**

**{**

        FRAM\_Init**();**

**}**

      i **+=** 1**;**

**}**

**}**

**if** **(!**err**)**

**{**

**return** REC\_OK**;**

**}**

**else**

**{**

**return** REC\_ERR\_DRIVE**;**

**}**

**}**

/\*\*

 \* **@brief** : 从底层硬件中读取数据

 \* **@param** : ptbl   存储表项

 \* **@param** : addr   存储地址

 \* **@param** : pvbuf  数据缓存

 \* **@param** : len    数据长度

 \* @retval: 0成功,其他失败

 \*/

RecordErr\_t Record\_HWRead**(**const RecordTBL\_t **\***ptbl**,** uint32\_t addr**,** void **\***pvbuf**,**

uint32\_t len**)**

**{**

  ASSERT**((**ptbl **!=** **NULL)** **&&** **(**pvbuf **!=** **NULL));**

**if** **((**ptbl**->**devType **==** REC\_DEV\_FM24CL64**)** **&&** **(**1 **==** ptbl**->**devAddr**))**

**{**

**if** **(**FRAM\_Read**(**FM24CL64\_1**,** addr**,** pvbuf**,** len**))**

**return** REC\_OK**;**

**else**

**return** REC\_ERR\_DRIVE**;**

**}**

**if** **((**ptbl**->**devType **==** REC\_DEV\_FM24CL64**)** **&&** **(**2 **==** ptbl**->**devAddr**))**

**{**

**if** **(**FRAM\_Read**(**FM24CL64\_2**,** addr**,** pvbuf**,** len**))**

**return** REC\_OK**;**

**else**

**return** REC\_ERR\_DRIVE**;**

**}**

**return** REC\_ERR\_CFG**;**

**}**

/\*\*

 \* **@brief** : 向底层硬件中写入数据

 \* **@param** : ptbl   存储表项

 \* **@param** : addr   存储地址

 \* **@param** : pvbuf  数据缓存

 \* **@param** : len    数据长度

 \* @retval: 0成功,其他失败

 \*/

RecordErr\_t Record\_HWWrite**(**const RecordTBL\_t **\***ptbl**,** uint32\_t addr**,** const void **\***pvbuf**,** uint32\_t len**)**

**{**

  ASSERT**((**ptbl **!=** **NULL)** **&&** **(**pvbuf **!=** **NULL));**

**if** **((**ptbl**->**devType **==** REC\_DEV\_FM24CL64**)** **&&** **(**1 **==** ptbl**->**devAddr**))**

**{**

**if** **(**FRAM\_Write**(**FM24CL64\_1**,** addr**,** pvbuf**,** len**))**

**return** REC\_OK**;**

**else**

**return** REC\_ERR\_DRIVE**;**

**}**

**if** **((**ptbl**->**devType **==** REC\_DEV\_FM24CL64**)** **&&** **(**2 **==** ptbl**->**devAddr**))**

**{**

**if** **(**FRAM\_Write**(**FM24CL64\_2**,** addr**,** pvbuf**,** len**))**

**return** REC\_OK**;**

**else**

**return** REC\_ERR\_DRIVE**;**

**}**

**return** REC\_ERR\_CFG**;**

**}**