UART to FLASH Write Protocol

Phyplus Confidential

Phyplus Inc.

2018/8/30

版本控制信息

版本/状态	作者	参与者	起止日期	备注
V0.1	丁明超		5/20/2018	文档初稿
V0.2	ZQ		8/30/2018	Add hexf write protocal (1.1.3) Hexf 文件格式(5.3) Update flash mapping(6.1,6.2)

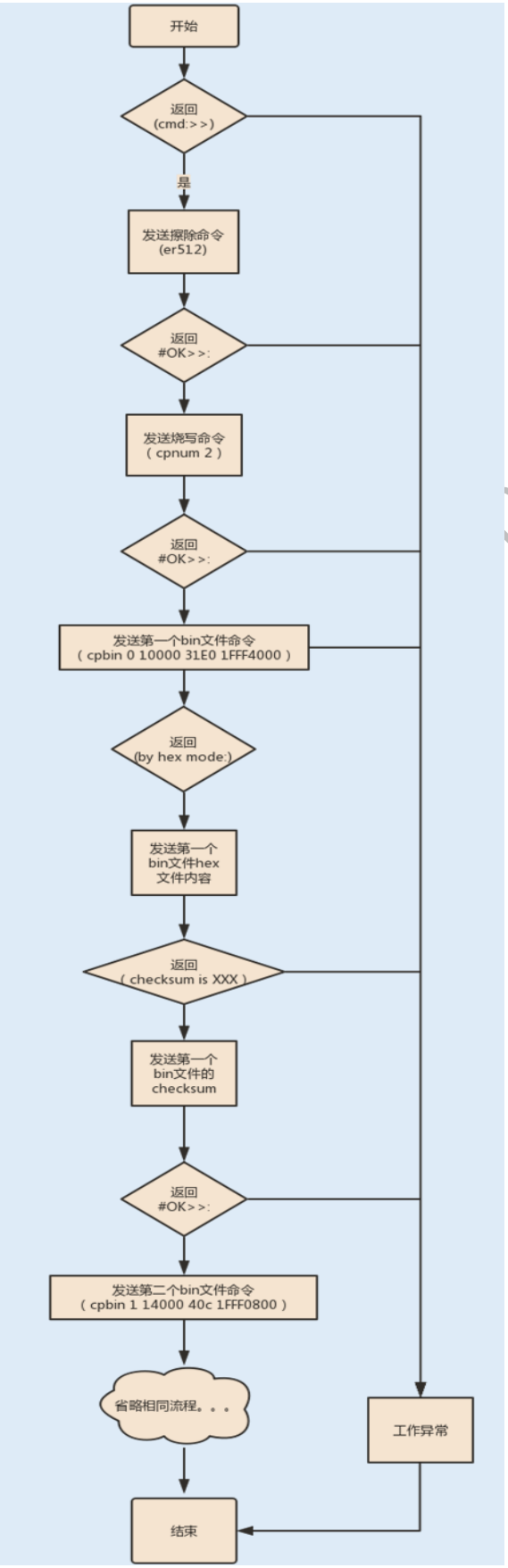
Phyplus Confidential

目录

1. Flash 烧写流程 3
1.1 流程详解 4
1.2 烧写示例 4
1.3 HexF 文件烧写示例 6
2. Flash 单地址写值流程 7
2.1 流程详解 7
2.2 写值示例 7
3. UART 改波特率流程 8
3.1. 流程详解 8
3.2. 变更示例 8
4. 常用交互命令 9
4.1 erase（擦除命令） 9
4.2 cpnum（烧写命令） 9
4.3 cpbin（烧写命令） 10
4.4 write（flash 写命令） 10
4.5 uarts（uart 命令） 11
5. HEX 文件解析 12
5.1 Hex 文件格式 12
5.2 Hex 解析示例 13
5.3 HexF 文件格式 13
6. Flash Mapping 15
6.1 NO OTA Mode Flash Mapping 15
6.2 Support OTA Mode Flash Mapping 15

Phyplus Confidential

1. FLASH 烧写流程


开始
返回
(cmd:>>)
是
发送擦除命令
(er512)
返回
#OK>>:
发送烧写命令
(cpnum 2)
返回
#OK>>:
发送第一个bin文件命令
(cpbin 0 10000 31E0 1FFF4000)
返回
(by hex mode:)
发送第一个
bin文件hex
文件内容
返回
(checksum is XXX)
发送第一个
bin文件的
checksum
返回
#OK>>:
发送第二个bin文件命令
(cpbin 1 14000 40c 1FFF0800)
省略相同流程。。。
工作异常
结束

1.1 流程详解

1. PHY620X 上电后，将 TM 拉高并 Reset 开发板，进入烧写模式。PHY620X 返回 ***cmd>>:***
2. 擦除 Flash，发送擦除命令详见（ERASE 命令），成功执行将返回 ***#OK>>:***
3. 烧写 Flash，发送 cpnum 命令详见（CPNUM 命令），成功执行将返回 ***#OK>>:***
   根据 cpnum 中的 bin 文件总数循环执行下列命令：
   发送 cpbin 命令详见（CPBIN 命令），成功执行将返回 ***by hex mode:***
   发送 bin 文件内容（按字节发送），成功发送将返回 ***checksum is: 0xXXX***
   发送 bin 文件 checksum，成功执行将返回 ***#OK>>:***
4. 烧写 Flash 流程结束

注：

1. 发送 cpnum 命令后，开始循环发送 bin 文件时，必须首先发送 Run Address 为 1FFF4000 的 bin 文件程序；
2. 对应 bin 文件的 Flash Address 需向 4k 个字节对齐，可用 Flash 区域见 Flash Mapping 的 App Bank；
3. 发送 bin 文件数据内容时，按字节传输，从低地址字节数据开始发送；
4. 对 bin 文件 checksum 的计算方式为，求 bin 文件数据的字节累加和，累加和的数据类型为无符号整型四字节，不计进位；
5. 对 Flash 的烧写，可以使用 HEX 文件，文件内容解析详见（HEX 文件解析）；

1.2 烧写示例

下面演示 HEX 文件的烧写过程：

1. 对 PHY620X 上电后，将 TM 拉高 Reset 开发板，进入烧写模式，uart 串口返回 cmd>>:

```
15:17:53.305 [Rx] - cmd>>:
```

2. 发送擦除 Flash 的命令 er512，等待返回#OK>>:

```
15:18:02.285 [Tx] - (size5) er512
15:18:03.431 [Rx] - #OK>>:
```

3. 打开 Hex 文件，Hex 文件内容如下：

```
:020000041FFFDC
:10080000000000003167FF1FCC6FFF1FF06FFF1F5C
:10081000DC78FF1F000000000000000000000000000066
……
:020000041FFFDC
:10400000207EFF1FD540FF1FDD40FF1FDF40FF1F49
:104010000000000000000000000000000000000000A0
……
:020000042000DA
:10000000003B40148019001BD01AF000003B40148F1
:10001000019001BD71B0000003B40148019001BD21
……
```

bin 1

bin 2

bin 3

PHYplus Confidential

:040000051FFF40C1D8
:00000001FF → EOF

注：一共三个扩展线性地址的记录，表明 hex 文件含有三个要烧写的 bin 文件。按照要求统计各个 bin 文件的大小、内容、checksum，已经运行的起始地址。

调整 bin 文件顺序，使 Run Address 为 1FFF4000 优先发送。

（第一个 bin 文件，size: 0x31E0，Run Address: 1FFF4000）

（第二个 bin 文件，size: 0x040C，Run Address: 1FFF0800）

（第三个 bin 文件，size: 0x0984，Run Address: 20000000）

假设烧写的起始 Flash Address 为 10000，后续 bin 文件烧写地址需向 4k 对齐。

（第一个 bin 文件，size: 0x31E0，Run Address: 1FFF4000，Flash Address: 10000）

（第二个 bin 文件，size: 0x040C，Run Address: 1FFF0800，Flash Address: 14000）

（第三个 bin 文件，size: 0x0984，Run Address: 20000000，Flash Address: 15000）

4. 发送 cpnum 命令，等待返回#OK>>:

```
15:18:07.623 [Tx] - (size8) cpnum 3
15:18:07.648 [Rx] - #OK>>:
```

5. 发送 cpbin 命令，等待返回 by hex mode:

```
15:18:07.654 [Tx] - (size27) cpbin 0 10000 31e0 1FFF4000
15:18:07.680 [Rx] - by hex mode:
```

6. 发送 Run Address: 1FFF4000 的 bin 内容，等待返回 check sum is:

```
15:18:07.693 [Tx] - (size12768) 207E FF1F D540 FF1F DD40 FF1F DF40
FF1F 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 E140 FF1F 0000 0000 0000 0000 E340 FF1F E540 FF1F 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0348 8546 00F0 26F8 0048 0047 8966 FF1F 207E
FF1F 0448 8047 0448 0047 FEE7 FEE7 FEE7 FEE7 FEE7 FEE7 5157 FF1F C140
FF1F 0346 0B43 9B07 03D0 09E0 08C9 121F 08C0 042A FAD2 03E0 0B78 0370
401C 491C 521E F9D2 7047 064C 0125 064E 05E0 E368 07CC 2B43 0C3C 9847
```

Bin 内容为：

:10400000207EFF1FD540FF1FDD40FF1FDF40FF1F49

:104010000000000000000000000000000000000A0

数据域为：20 7E FF 1F D5 40 FF 1F DD 40 FF 1F DF 40 FF 1F

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

即发送 bin 文件数据内容时，按字节传输，从低地址字节数据开始发送。

发送完成后，返回 checksum is 0x0012776c

```
6C5A 5105 1250 2D20 0302 0AB0 141A FF4C 4818 0215 FDA5 0693 A4E2 4FB1
AFCF C6EB 0764 7825 2774 6BED C53A 8E20 424C 4520 3E31 7C40 787D 898D
102D 230F 9611 220F 990E 917A 209F
15:18:08.847 [Rx] - checksum is: 0x0012776c
```

Confidential

7. 发送 bin 文件的 checksum，等待返回#OK>>:

checksum 为 bin 文件数据的字节累加和

```
15:18:08.898 [Tx] - (size6) 12776c
15:18:09.137 [Rx] - #OK>>:
```

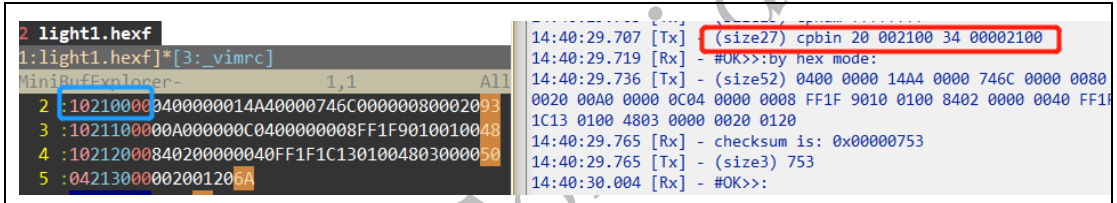
后续过程重复 4-7

1.3 HEXF 文件烧写示例

烧写 hexf 文件的过程和 hex 比较相似，有几个地方需要注意。

1. 并不需要烧写 cpnum，只需要使用 cpbin 一条命令，[cpbin *index flash_addr size run_addr*]
2. *index* 可以从 0x20 开始递增，每烧录一块 bin 文件，*index* ++
3. *flash_addr* 由 hexf 里面的每一段的 bin 文件的启示地址决定。
4. *size* 为每一段 bin 文件的大小
5. *run_addr* 不需要填写可以填入 flash address

表格 1 hexf **烧录示例**



1. 先发送 cpbin 20 002100 34 00002100 （20 为 index）
2. 收到[#OK>>:by hex mode:]
3. 即发送 bin 文件数据内容时，按字节传输，从低地址字节数据开始发送。
4. 发送完成后，返回 checksum is 0x0000753
5. 发送 bin 文件的 checksum，等待返回#OK>>:checksum 为 bin 文件数据的字节累加和
6. 收到[#OK>>:]

如上表所示，烧录 hexf 文件中的每一段 bin 文件都流程都一样。只需要修改每一段 bin 的 index。

Phyplus Confidential

2. FLASH 单地址写值流程

2.1 流程详解

1. 将开发板置于烧写模式下
2. 发送 Flash 写命令详见（WRITE 命令），成功执行将返回 ***#OK>>:***

2.2 写值示例

```
15:18:10.104 [Tx] - (size23) write11004008 FFFFFF32
15:18:10.127 [Rx] - #OK>>:
```

Phyplus Confidential

3. UART 改波特率流程

3.1. 流程详解

1. 将开发板置于烧写模式下
2. 发送 UARTS 命令详见（UARTS 命令），成功执行将返回 ***#OK>>:***

3.2. 变更示例

```
16:06:25.070 [Tx] - (size12) uarts115200
16:06:25.097 [Rx] - #OK>>:
```

Phyplus Confidential

4. 常用交互命令

4.1 ERASE（擦除命令）

该命令说明要擦除的 Flash 大小和地址

Parameter:

Name	Parameters Remarks	Sample
er	固定命令参数	er
size	要擦出的 Flash 大小（choices） 512(全部擦除)，256，32k，64k	512
Address	要擦出的 Flash 地址（choices） （512，256）不填，（32k，64k）必填	/

Requested sample:（命令参数分割如下）

er512

该命令表明要擦除整个 Flash 内容

er32k 10000

该命令表明从 10000 地址开始擦除 32k 内容

Parameter:

Name	Parameters Remarks	Sample
era	固定命令参数	era
size	要擦出的 Flash 大小 4k	4k
Address	要擦出的 Flash 地址 必填	10000

era4k 10000

该命令表明从 10000 地址开始擦除 4k 内容

4.2 CPNUM（烧写命令）

该命令说明要烧写的 bin 文件的总数。

Parameter:

Name	Parameters Remarks	Sample
cpnum	固定命令参数	cpnum
number	烧写的 bin 文件总数（variant）	3

Requested sample:（命令参数用空格分割）

cpnum 3

该命令表明要烧写的 bin 文件数量为 3 个。

Phyplus Confidential

4.3 CPBIN（烧写命令）

该命令说明要烧写的单个 bin 文件的具体参数（Index, Flash Address, Size, Run Address）

Parameter:

Name	Parameters Remarks	Sample
cpbin	固定命令参数	cpbin
index	要烧写 bin 文件的发送序号（从 0 开始）（variant）	0
Flash Address	要烧写 bin 文件的 Flash 地址（variant）	10000
Size	要烧写 bin 文件的文件大小（variant）	31E0
Run Address	要烧写 bin 文件的运行地址（variant）	1FFF4000

Requested sample: （命令参数用空格分割）

cpbin 0 10000 31E0 1FFF4000

该命令表明烧写的 bin 文件为第一个，要烧写的 Flash 地址为 10000，bin 文件大小为 31E0，运行时地址为 1FFF4000.

注：1. 必须先发送主程序所在的 bin 文件（即 Run Address 为 1FFF4000 的 bin 文件）；
2. Flash Address 的可用区域示例详见 Flash Mapping 的 App Bank；

4.4 WRITE（FLASH 写命令）

该命令说明要手动写入 Flash 地址的值。

Parameter:

Name	Parameters Remarks	Sample
write	固定命令参数	write
Flash Address	要写入的 Flash 地址（variant）	4008
Value	要写入上述地址的值（variant）	FFFFFF32

Requested sample:（命令参数分割如下）

write4008 FFFFFF32

Phyplus Confidential

4.5 UARTS（UART 命令）

该命令说明要变更的波特率数值。

Parameter:

Name	Parameters Remarks	Sample
uarts	固定命令参数	uarts
speed	要变更的波特率数值（十进制数）(variant)	250000（dec）

Requested sample:（命令参数不分割）

uarts 250000

该命令表明将 PHY620X 的波特率变更为 250000.

Phyplus Confidential

5. HEX 文件解析

HEX 文件中可以解析到 bin 文件的个数、每个 bin 文件的具体内容、大小以及运行时地址。

Hex 文件是可以烧录到 MCU 中，被 MCU 执行的一种文件格式。如果用记事本打开可发现，整个文件以行为单位，每行以冒号开头，内容全部为 16 进制码（以 ASCII 码形式显示）。

5.1 HEX 文件格式

打开 Hex 文件，看到类似如下文本：

:1000D000E70100088D050008E7010008E7010008B6

（1）起始标识（:）

（2）第 1 字节（10），表示本行数据的长度（0x10）

（3）第 2、3 字节（00D0），表示本行数据的起始地址（0x00D0）

（4）第 4 字节（00）表示数据类型，包括：0x00、0x01、0x02、0x03、0x04、0x05

1. "00" Data Record：记录数据，HEX 文件的大部分记录都是数据记录
2. "01" End of File Record: 标识文件结束
3. "02" Extended Segment Address Record: 用来标识扩展段地址的记录
4. "03" Start Segment Address Record: 开始段地址记录
5. "04" Extended Linear Address Record: 用来标识扩展线性地址的记录
6. "05" Start Linear Address Record: 开始线性地址记录

（5）数据（E70100088D050008E7010008E7010008），共 0x10 个字节

（6）最后一个数据（B6），表示校验和。

注： 计算 0xB6 前所有 16 进制数的累加和，不计进位（即：低位单字节），检验和 = 0x100 - 累加和。

Phyplus Confidential

5.2 HEX 解析示例

在烧写开发板时，我们常用到的 Hex 数据类型为 04，00，01，其中 01 表示文件的结束。我们重点关注 04 和 00，下面为待烧写 Hex 文件的开头 6 行数据截取内容。

:020000041FFFDC

:10080000000000003167FF1FCC6FFF1FF06FFF1F5C

:10081000DC78FF1F00000000000000000000000000 66

……

……

:020000041FFFDC

:10400000207EFF1FD540FF1FDD40FF1FDF40FF1F49

:104010000000000000000000000000000000000000A0

……

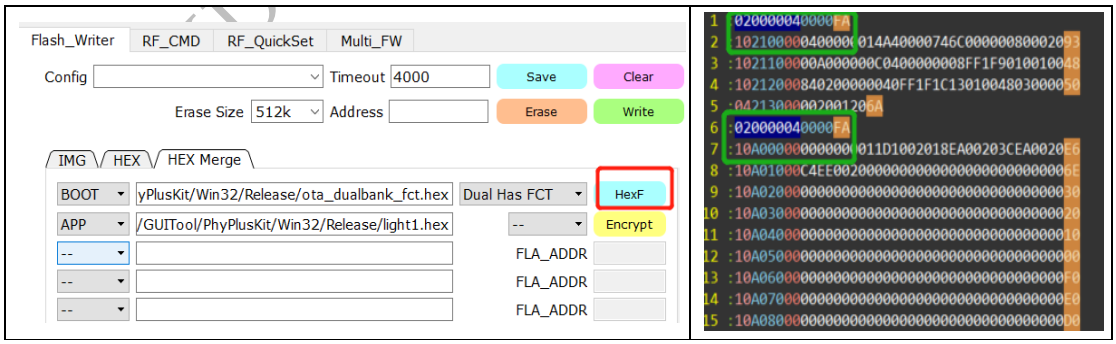
……

:00000001FF

1. 第一行的第四字节为 0x04，表示 Hex 文件的数据地址采用扩展线性地址的方式，数据域 1FFF 表示随后数据的基地址。（注：数据类型为 0x04 的数据起始地址始终为 0x0000，）
2. 后面几行数据的数据类型均为 0x00，表示这几行是 bin 文件的具体数据内容，且数据的起始地址分别为 0x0800, 0x0810…，根据基地址 1FFF，我们可以得知第一个 bin 文件的运行地址为 0x1FFF0800.
3. 当一个 bin 文件的数据内容结束时，会紧跟着一条新的扩展线性地址记录，以及 bin 文件的数据内容。使用同上方法获得后续的 bin 文件运行地址和具体内容。

5.3 HEXF 文件格式

可用通过 PhyPlusKit 的 HEX Merge Page 中的[HexF] 来产生等多个 hex 文件的合并文件，该合并文件的后缀为*.hexf，和章节 5.1 中的 Hex 文件格式一样，都符合 intel Hex 格式。



Hexf 文件包含多个地址段。具体可以参考章节 6，Flash Mapping。

Phyplus Confidential

6. FLASH MAPPING

6.1 NO OTA MODE FLASH MAPPING

Flash Mapping NO OTA			
Function	**Address**	**Size**	**Description**
Reserved	00000~01fff	8k	Reserved For PhyPlus Only
1st Boot info	02000~03fff	8k	Boot Configuration
FCDS	04000~04fff	4k	Factory Configuration Data Storage
UCDS	05000~08fff	16k	User Configuration Data Storage
App Bank	**09000~4ffff**	**284k**	**Application Code**
FCT App	50000~6ffff	128k	Option
NVM	70000~7ffff	64k	Option

6.2 SUPPORT OTA MODE FLASH MAPPING

Support OTA								
	512K版本 (Dual bank) (Has FCT)		512K版本 (Single bank) (Has FCT)		512K版本 (Dual bank) (No FCT)		512K版本 (Single bank) (No FCT)	
Reserved By PhyPlus	00000~01fff	8k	00000~01fff	8k	00000~01fff	8k	00000~01fff	8k
1st Boot info	02000~03fff	8k	02000~03fff	8k	02000~03fff	8k	02000~03fff	8k
FCDS	04000~04fff	4k	04000~04fff	4k	04000~04fff	4k	04000~04fff	4k
UCDS	05000~08fff	16k	05000~08fff	16k	05000~08fff	16k	05000~08fff	16k
2nd Boot info	09000~09fff	4k	09000~09fff	4k	09000~09fff	4k	09000~09fff	4k
OTA bootloader	0a000~11fff	32k	0a000~11fff	32k	0a000~11fff	32k	0a000~11fff	32k
FCT App	12000~2ffff	120k	12000~2ffff	120k		0k		0k
App Bank0	30000~4ffff	128k	30000~4ffff	128k	12000~31fff	128k	12000~31fff	128k
App Bank1	50000~6ffff	128k		0k	32000~51fff	128k		0k
NVM	70000~7ffff	64k	50000~7ffff	192k	52000~7ffff	184k	32000~7ffff	312k

Phyplus confidential