**新**

**版**

**手**

**持**

**机**

**协**

**议**

**说**

**明**

**文**

**档**

目录

[前言 1](#_Toc2761048)

[一、标签信息 1](#_Toc2761049)

[1、51822发送给1754标签信息格式 1](#_Toc2761050)

[二、设置信息 2](#_Toc2761051)

[1、1754设置51822频率 2](#_Toc2761052)

[2、1754设置51822 mac地址 2](#_Toc2761053)

[3、1754设置51822 rf速度 2](#_Toc2761054)

[4、1754设置51822 数据长度 3](#_Toc2761055)

[5、1754重启51822 3](#_Toc2761056)

[三、查询信息(查询命令待添加) 3](#_Toc2761057)

[1、1754查询51822 radio信息 3](#_Toc2761058)

[附录I 4](#_Toc2761059)

注意：这份协议只说明了手持机可以进行哪些操作，发送内容可以根据最下面程序进行计算,假如计算太麻烦，请自行使用vs做一个串口工具即可，由于时间关系说明文档写的不详细，命令仅用于测试使用。如要详细命令请后续补充。当下面枚举和程序有差异时请及时更新。

## 前言

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Head | Len | Softver | Cmdtype | Datatype | Data |
| 7e | Xx | 0x01 | Xx | Xx | Xx |

typedef enum{ //设计成美剧主要是为了后面维护工作，使用宏定义不利于后续维护工作

settype=0x1a, //设置命令类型

seltype, //查询命令类型

respond, //回应设置命令类型

selinfo, //查询标签信息数据类型

setfer, //设置频率数据类型

setmac, //设置地址码数据类型

setspeed=0x20, //设置速度数据类型

setdatalen, //设置数据长度数据类型

seldata, //查询标签信息数据类型

selrf, //查询频率数据类型

setsendch, //设置发送通道数据类型

setsendpow, //设置发送功率数据类型

target, //表示标签信息数据类型

reset, //表示重新启动

}\_STATUS;

## 一、标签信息

### 1、51822发送给1754标签信息格式

**Eg:7E 0C 01 26 01 CC BB 12 34 55 49 52 90**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 命令类新 | 数据类新 | 内容 |
| 发送 | 0x26 | 0x01表示通道号 | CC BB 12 34 55 49 |

## 二、设置信息

### 1、1754设置51822频率

**Eg:7E 07 01 1a 1e 01 10 5b 2e**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 命令类新 | 数据类新 | 内容 |
| 发送 | 0x1a | 0x1e | 0x10 |
| 设置成功回应 | 0x1c | 0x1e | Ch表示通道号 |
| 设置不成功 | Null | Null | Null |

### 2、1754设置51822 mac地址

**Eg:7E 0b 01 1a 1f 01 12 34 56 78 90 16 5e**

**7e 0b 01 1a 1f 03 a5 aa 55 10 01 d2 22 如设置68号段地址码**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 命令类新 | 数据类新 | 内容 |
| 发送 | 0x1a | 0x1f | 0x12,0x34,0x56,0x78,0x90 |
| 设置成功回应 | 0x1c | 0x1f | Ch表示通道号 |
| 设置不成功 | Null | Null | Null |

### 3、1754设置51822 rf速度

**Eg: 7E 07 01 1a 20 01 aa bb 51**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 命令类新 | 数据类新 | 内容 |
| 发送 | 0x1a | 0x20 | 0xaa |
| 设置成功回应 | 0x1c | 0x20 | Ch表示通道号 |
| 设置不成功 | Null | Null | Null |

### 4、1754设置51822 数据长度

**Eg: 7e 07 01 1a 21 03 09 ab 88**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 命令类新 | 数据类新 | 内容 |
| 发送 | 0x1a | 0x21 | 0xaa |
| 设置成功回应 | 0x1c | 0x21 | Ch表示通道号 |
| 设置不成功 | Null | Null | Null |

### 5、1754重启51822

**Eg:7e 06 01 1a 27 03 d7 ca**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 命令类新 | 数据类新 | 内容 |
| 发送 | 0x1a | 0x27 | 0x03表示那个通道 |
| 设置成功回应 | Null | Null | null |
| 设置不成功 | Null | Null | Null |

## 三、查询信息(查询命令待添加)

### 1、1754查询51822 radio信息

**Eg: 7e 06 01 1b 1d 01 15 6b**

**respond ：1D 01 0A 09 D2 EC FA 10 01 01**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 命令类新 | 数据类新 | 内容 |
| 发送 | 0x1b | 0x1d | Ch |
| 设置成功回应 | 0x1c | 0x1b | \*data |
| 设置不成功 | Null | Null | Null |

## 附录I

校验功能实现程序

#include "stdio.h"

#define u8 unsigned char

#define u16 unsigned int

const u16 CRCTable[] =

{

0x0000,0xC0C1,0xC181,0x0140,0xC301,0x03C0,0x0280,0xC241,

0xC601,0x06C0,0x0780,0xC741,0x0500,0xC5C1,0xC481,0x0440,

0xCC01,0x0CC0,0x0D80,0xCD41,0x0F00,0xCFC1,0xCE81,0x0E40,

0x0A00,0xCAC1,0xCB81,0x0B40,0xC901,0x09C0,0x0880,0xC841,

0xD801,0x18C0,0x1980,0xD941,0x1B00,0xDBC1,0xDA81,0x1A40,

0x1E00,0xDEC1,0xDF81,0x1F40,0xDD01,0x1DC0,0x1C80,0xDC41,

0x1400,0xD4C1,0xD581,0x1540,0xD701,0x17C0,0x1680,0xD641,

0xD201,0x12C0,0x1380,0xD341,0x1100,0xD1C1,0xD081,0x1040,

0xF001,0x30C0,0x3180,0xF141,0x3300,0xF3C1,0xF281,0x3240,

0x3600,0xF6C1,0xF781,0x3740,0xF501,0x35C0,0x3480,0xF441,

0x3C00,0xFCC1,0xFD81,0x3D40,0xFF01,0x3FC0,0x3E80,0xFE41,

0xFA01,0x3AC0,0x3B80,0xFB41,0x3900,0xF9C1,0xF881,0x3840,

0x2800,0xE8C1,0xE981,0x2940,0xEB01,0x2BC0,0x2A80,0xEA41,

0xEE01,0x2EC0,0x2F80,0xEF41,0x2D00,0xEDC1,0xEC81,0x2C40,

0xE401,0x24C0,0x2580,0xE541,0x2700,0xE7C1,0xE681,0x2640,

0x2200,0xE2C1,0xE381,0x2340,0xE101,0x21C0,0x2080,0xE041,

0xA001,0x60C0,0x6180,0xA141,0x6300,0xA3C1,0xA281,0x6240,

0x6600,0xA6C1,0xA781,0x6740,0xA501,0x65C0,0x6480,0xA441,

0x6C00,0xACC1,0xAD81,0x6D40,0xAF01,0x6FC0,0x6E80,0xAE41,

0xAA01,0x6AC0,0x6B80,0xAB41,0x6900,0xA9C1,0xA881,0x6840,

0x7800,0xB8C1,0xB981,0x7940,0xBB01,0x7BC0,0x7A80,0xBA41,

0xBE01,0x7EC0,0x7F80,0xBF41,0x7D00,0xBDC1,0xBC81,0x7C40,

0xB401,0x74C0,0x7580,0xB541,0x7700,0xB7C1,0xB681,0x7640,

0x7200,0xB2C1,0xB381,0x7340,0xB101,0x71C0,0x7080,0xB041,

0x5000,0x90C1,0x9181,0x5140,0x9301,0x53C0,0x5280,0x9241,

0x9601,0x56C0,0x5780,0x9741,0x5500,0x95C1,0x9481,0x5440,

0x9C01,0x5CC0,0x5D80,0x9D41,0x5F00,0x9FC1,0x9E81,0x5E40,

0x5A00,0x9AC1,0x9B81,0x5B40,0x9901,0x59C0,0x5880,0x9841,

0x8801,0x48C0,0x4980,0x8941,0x4B00,0x8BC1,0x8A81,0x4A40,

0x4E00,0x8EC1,0x8F81,0x4F40,0x8D01,0x4DC0,0x4C80,0x8C41,

0x4400,0x84C1,0x8581,0x4540,0x8701,0x47C0,0x4680,0x8641,

0x8201,0x42C0,0x4380,0x8341,0x4100,0x81C1,0x8081,0x4040

};

u16 check\_crc(u8 \*buf, u16 len, u16 FirstData)

{

u16 i;

u8 index;

u8 CRCH = 0;

u8 CRCL = 0;

CRCH = (u8)(FirstData>>8);

CRCL = (u8)(FirstData);

for( i = 0 ; i < len ; i++ )

{

index = CRCH ^ buf[i];

CRCH = (u8)CRCL ^ (u8)CRCTable[index];

CRCL = CRCTable[index] >> 8;

}

return( CRCL<<8|CRCH);

}

//u8 pbuff[] = {0x7E,0x07,0x01,0x1a,0x1e,0x01,0x10,0x5b,0x2e};//设置通道1频率为2410

//u8 pbuff[] = {0x7e,0x07,0x01,0x1a,0x1e,0x01,0x20,0x5b,0x3a};//设置通道1频率为2420

//u8 pbuff[] = {0x7e,0x07,0x01,0x1a,0x1e,0x01,0x30,0x5a,0xf6};//设置通道1频率为2430

u8 pbuff[] = {0x7e,0x07,0x01,0x1a,0x1e,0x01,0x40,0x5b,0x12};//设置通道1频率为2440

main()//程序打印出来的命令是可以对51822进行设置的

{

u8 i;

u8 buf[20];

u8 h,l;

//因为lpc1754是小端模式，故将crc\_temp高低字节颠倒一下

u16 crc\_temp;

crc\_temp = check\_crc(&(pbuff[1]),(pbuff[1]-1),0xffff);

h = crc\_temp >> 8;

l = crc\_temp & 0x00ff;

printf("%04x\r\n",crc\_temp);

for(i=0;i<pbuff[1];i++)

{

buf[i] = pbuff[i];

printf("0x%02x,",buf[i]);

}

printf("0x%02x,",l);

printf("0x%02x,",h);

printf("\r\n");

}