Km002c接口说明

Km002c提供三种接口, 用户可以根据自己的需求采用哪种接口

1. **USER**, WINDOWS平台叫WINUSB,其它平台叫USB USER,传输速率200KB/s

2. **CDC** 虚拟串口,波特率可以无视,任意值都可以.比较容易上手,任何串口调试工具都能调试, 传输速率200KB/s ,

3. **HID** 任何平台都不需要安装驱动,速率一般,数据长度限制为64byte 传输速率60KB/s

命令是通用的USER , CDC, HID都可以使用相同的命令.

命令封装成一个结构体,占用4个字节.

typedef union

{

uint32\_t object;

uint16\_t word[2];

uint8\_t byte[4];

struct

{

uint32\_t type : 7; /\*!< 大于63为数据 \*/

uint32\_t extend : 1; /\*!< 用于传输超大数据包 \*/

uint32\_t id : 8; /\*!< id 默认 0 \*/

uint32\_t : 1; /\*!< 数据为加密格式 \*/

uint32\_t att : 15; /\*!< 属性代码 \*/

}ctrl;

struct

{

uint32\_t type : 7;

uint32\_t extend : 1;

uint32\_t id : 8;

uint32\_t : 6;

uint32\_t obj : 10; /\*!< 4字节一个对象 \*/

}data;

struct

{

uint32\_t att : 15;

uint32\_t next : 1;

uint32\_t chunk : 6;

uint32\_t size : 10; /\*!< 不能超过1024-8个字节 chunk \* size 不能超过4080 HID接口不能超过60 \*/

}header;

}MsgHeader\_TypeDef;

/\* Message type \*/

enum cmd\_ctrl\_msg\_type

{

/\* 0 Reserved \*/

CMD\_SYNC = 1,

CMD\_CONNECT,

CMD\_DISCONNECT,

CMD\_RESET,

CMD\_ACCEPT,

CMD\_REJECT,

CMD\_FINISHED,

CMD\_JUMP\_APROM,

CMD\_JUMP\_DFU,

CMD\_GET\_STATUS,

CMD\_ERROR,

/\*app\*/

CMD\_GET\_DATA,

CMD\_GET\_FILE

};

/\* Data Message type \*/

enum cmd\_data\_msg\_type

{

/\* 0 - 63 Reserved \*/

CMD\_HEAD = 64,

CMD\_PUT\_DATA,//

};

/\* attribute type \*/

enum attribute\_data\_type

{

/\* 0 - 63 Reserved \*/

ATT\_ADC = 0x001,

ATT\_ADC\_QUEUE = 0x002,

ATT\_ADC\_QUEUE\_10K = 0x004, //10K数据

ATT\_SETTINGS = 0x008,

ATT\_PD\_PACKET = 0x010,

ATT\_PD\_STATUS = 0x020,

ATT\_QC\_PACKET = 0x040

};

举例

获取ADC数据,

MsgHeader\_TypeDef head;

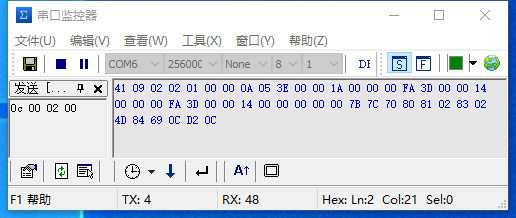
head.object = 0;

head.ctrl.type = CMD\_GET\_DATA; /\*cmd 0x0C\*/

head.ctrl.id = 0/\*\*/;

head.ctrl.att = ATT\_ADC /\*attribute\*/;

转换HEX后得到0c 00 02 00



收到一串数据由数据头部和ADC结构体

typedef struct

{

int32\_t Vbus; //单位 1uV

int32\_t Ibus; //单位 1uA

int32\_t Vbus\_avg; //平滑滤波后的平均电压 单位 1uV

int32\_t Ibus\_avg; //平滑滤波后的平均电流 单位 1uA

int32\_t Vbus\_ori\_avg; //平滑滤波后未校准的平均电压

int32\_t Ibus\_ori\_avg; //平滑滤波后未校准的平均电流

int16\_t Temp; //内部温度

uint16\_t Vcc1; //分辨率 0.1mV

uint16\_t Vcc2;

uint16\_t Vdp;

uint16\_t Vdm;

uint16\_t Vdd; //内部VDD电压

uint8\_t Rate:2;

uint8\_t n[3];

}AdcData\_TypeDef;

把 41 09 02 02 转换为MsgHeader\_TypeDef head;

Head.type == 41, 41表示002c回复的数据包类型

Head.id == 09 随机数

Head.att == ATT\_ADC /\*attribute\*/;

把01 00 00 0A 转换为MsgHeader\_TypeDef head\_ext;

把 01 00 00 0A 05 3E 00 00 1A 00 00 00 FA 3D 00 00 14

00 00 00 FA 3D 00 00 14 00 00 00 00 00 7B 7C 70 80 81 02 83 02

4D 84 69 0C D2 0C 转换为 AdcData\_TypeDef adc;