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

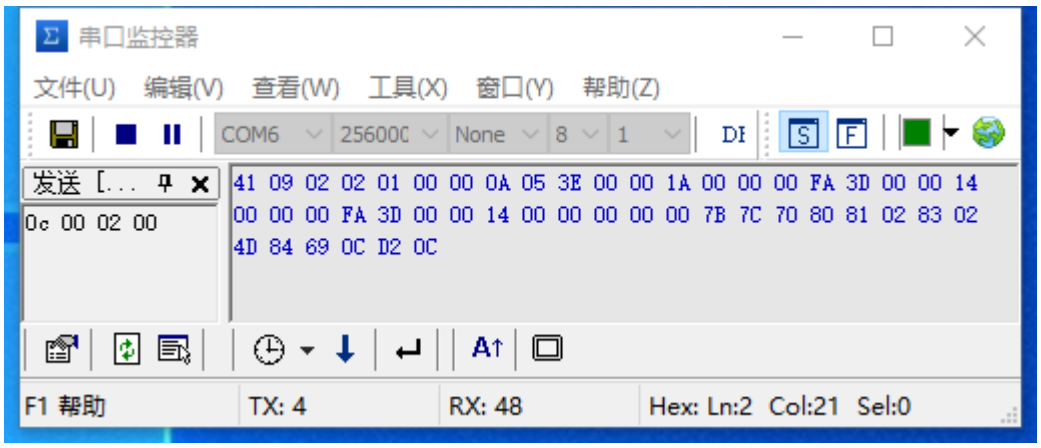
命令是通用的 USER , CDC, HID 都可以使用相同的命令.
命令封装成一个结构体,占用 4 个字节.

```
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;