```c
/* CopyRight(C) LabArtwork.com 2016
 * LabArtwork版权所有
 */
#include <stdint.h>

/* CAN message object structure                                        */
typedef struct  {
  uint32_t id;                /* 29 bit identifier                                */
  uint8_t  data[8];           /* Data field                                       */
  uint8_t  len;               /* Length of data field in bytes                    */
  uint8_t  ch;                /* Object channel                                   */
  uint8_t  format;            /* CAN_FORMAT,0 - STANDARD, 1- EXTENDED IDENTIFIER
  */
  uint8_t  type;              /* CAN_FRAME,0 - DATA FRAME, 1 - REMOTE FRAME               */
}  CAN_msg ;

#define MSG_CH_CAN 0
#define MSG_CH_UART 0x35

/* Symbolic names for formats of CAN message                           */
typedef enum {STANDARD_FORMAT = 0, EXTENDED_FORMAT} CAN_FORMAT;

/* Symbolic names for type of CAN message                              */
typedef enum {DATA_FRAME = 0, REMOTE_FRAME}         CAN_FRAME;

/* 把can数据包发送到uart
 * @bufsize:<=8
 */
int uart_send_canbuf(uint32_t id, void *buf, uint8_t bufsize)
{
    int i,idx=2;
    uint8_t can_buf[32],temp,crc=0,*pbuf=(uint8_t*)buf;

    /* head */
    can_buf[0] = 0xaa;
    can_buf[1] = 0xaa;

    /* ext id */
    id &= 0x3FFFFFFF;

    for(i=0;i<4;i++){
        temp = (id >> (i * 8)) & 0xFF;
        if (temp == 0x55 || temp == 0xAA){
            can_buf[idx++] = 0xa5;
        }
        can_buf[idx++] = temp;
        crc += temp;
    }

    /* crc in data[8] */
    for(i=0;i<bufsize;i++){
        if(pbuf[i]==0x55 || pbuf[i]==0xAA){
            can_buf[idx++] = 0xa5;
        }

        can_buf[idx++] = pbuf[i];
        crc += pbuf[i];
    }
    for(;i<8;i++){
        can_buf[idx++] = 0;
    }

    /* len/formatpe */
    can_buf[idx++] = bufsize;
    can_buf[idx++] = MSG_CH_UART; /* ch=0 */
    can_buf[idx++] = EXTENDED_FORMAT; /*EXTENDED_FORMAT*/
    can_buf[idx++] = DATA_FRAME;
```

```
    /* crc more */
    for (i = 0; i<4; i++){
        crc += can_buf[idx-4+i];
    }
    can_buf[idx++] = crc;

    /* tail */
    can_buf[idx++] = 0x55;
    can_buf[idx++] = 0x55;

    /* to uart */
    i = 0;
    while(i < idx){
        while(UART1_GetFlagStatus(UART1_FLAG_TXE) == RESET);
        UART1_SendData8(can_buf[i++]);
    }

    return 0;
}

int uart_send_canmsg(CAN_msg *pmsg)
{
    return uart_send_canbuf(pmsg->id,pmsg->data,pmsg->len);
}

#define CAN_PKT_SIZE 21
#define MAX_CAN_PARSE_BUF_LEN 32
static uint8_t g_can_frame[MAX_CAN_PARSE_BUF_LEN];

enum frame_sync_e{
    FS_FIND_AA1,
    FS_FIND_AA2,
    FS_FIND_DATA,
    FS_FIND_CRC, /* 未使用 */
    FS_FIND_551,
    FS_FIND_552
};

/* 从串口接收数据后调用本函数，并组包成can_msg */
void uart_rec_canmsg(uint8_t data)
{
    CAN_msg *pmsg;
    uint8_t cur_byte;
    static uint8_t last_byte=0,frame_idx=0;
    static enum frame_sync_e sync_state=FS_FIND_AA1;

    /* 超时复位状态机操作，可根据实际情况删除 */
    if(GET_DIFF_TICK(pudat->last_tick) > 250){
        sync_state = FS_FIND_AA1;
        frame_idx = 0;
    }

    cur_byte = data;
    if(sync_state == FS_FIND_AA1){
        if(cur_byte == 0xAA){
            sync_state = FS_FIND_AA2;
            frame_idx = 0;
            g_can_frame[frame_idx++] = cur_byte;
        }
    }

    else if(sync_state == FS_FIND_AA2){
        if(cur_byte == 0xAA){
            sync_state = FS_FIND_DATA;
            g_can_frame[frame_idx++] = cur_byte;
        }else{
```

```
            sync_state = FS_FIND_AA1;
        }
    }

    /* 删除插入的A5数据 */
    else if(sync_state == FS_FIND_DATA){
        if( (cur_byte==0x55 || cur_byte==0xAA) && frame_idx!=(CAN_PKT_SIZE-3))
        {
            if(last_byte == 0xa5){
                g_can_frame[frame_idx-1] = cur_byte;
            }
            else{
                sync_state = FS_FIND_AA1;
                frame_idx = 0;
            }
        }

        else{
            g_can_frame[frame_idx++] = cur_byte;

            if(frame_idx == (CAN_PKT_SIZE-3)){
                sync_state = FS_FIND_CRC;
            }
        }
    }

    /* CRC不加A5 */
    else if(sync_state == FS_FIND_CRC){
        g_can_frame[frame_idx++] = cur_byte;

        /* check crc */
        sync_state = FS_FIND_551;
    }

    /* 尾巴 */
    else if(sync_state == FS_FIND_551){
        if(cur_byte == 0x55){
            g_can_frame[frame_idx++] = cur_byte;
            sync_state = FS_FIND_552;
        }else
            sync_state = FS_FIND_AA1;
    }

    else if(sync_state == FS_FIND_552){
        sync_state = FS_FIND_AA1;
        if(cur_byte == 0x55){
            g_can_frame[frame_idx] = cur_byte;

            pmsg = (CAN_msg*)&g_can_frame[2];
            pmsg->ch = MSG_CH_UART; /* 视情况自己定义 */

            /* 存入缓冲区待使用 */
            memcpy(&g_uart_dat.msg_rcv,pmsg,sizeof(CAN_msg));
        }
    }

    last_byte = cur_byte;
    pudat->last_tick = get_timer_tick();
}

/* CopyRight(C) LabArtwork.com 2016
 * LabArtwork版权所有
 */
```