USR-C322 透传 PC1 加密说明

V1.0

1、PC1 加密说明

首先查看模块版本号，如果版本低于 V2.1.10，则需要升级固件才能够支持此功能。

USR-C322 具有透传加密功能，加密方式为 PC1 16 字节加密。

USR-C322 串口接收到透传数据后，模块根据配置的加密字对数据进行 PC1 加密计算，然后将数据传输到服务端。模块收到服务端传送来的加密数据后，对数据进行解密，然后将数据透传到串口。模块工作流程如下：

开启 PC1 加密后注意事项：

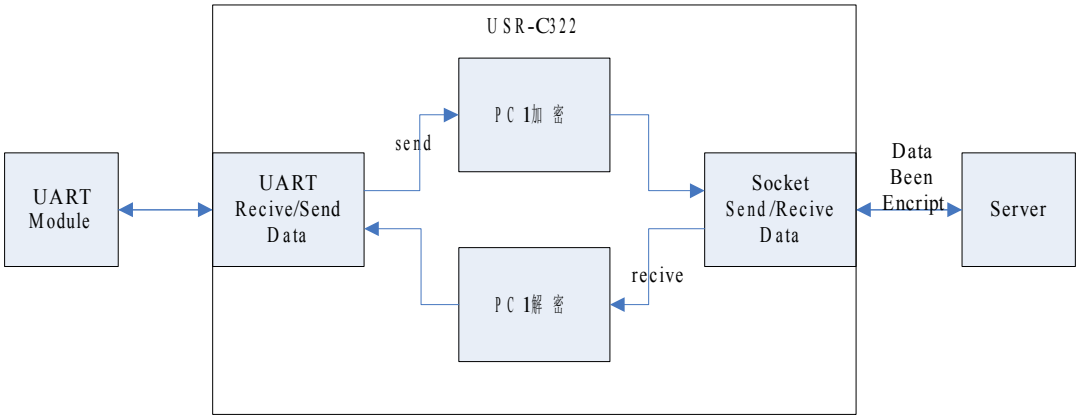
1、 数据 PC1 加密是逐个字节加密，前一个字节数据会对后续数据加密产生影响，所以数据加密必须以包为单位，要求用户在发送数据包时增大发送间隔，避免合包现象。

2、 模块加密字必须与服务端加密字相同，否则加密解密数据会产生错误。

3、 加密字配置：

AT+TRENC=on,1234567890abcdefabcdef1234567890\r

At 指令设置 32 个字符 0-9，a-f，或 A-F，模块将 32 个字符，组合成 16 字节 hex 作为加密字。



2、工具说明

PC1 加密解密测试工具如下图，首先需要设定加密字，加密字必须和模块的对应才能够正确收发数据。

3、PC1 加密及解密代码。

```
unsigned int inter, cfc, cfd;
unsigned int si, x1a2;
unsigned char Key[16]={
                        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                        };//16字节加密字
unsigned char DecryptKey[16] ;
void exchange(unsigned int *a, unsigned int *b)
{
  unsigned int tmp;
  tmp = *a;
  *a = *b;
  *b = tmp;
}
void DecryptInit()
{
    unsigned char tmp;
    si = 0;
    x1a2 = 0;
    for(tmp=0;tmp<16;tmp++)
    {
        DecryptKey[tmp]= Key [tmp];
    }
}

// 加密一个字节
void PC1assemble128()
{
  unsigned char i;
  unsigned int ax, bx, cx, dx;
  unsigned int x1a0[9];
  inter = 0;
  x1a0[0] = 0;
  for(i = 0; i < 8; i++)
  {
    x1a0[i + 1] = x1a0[i] ^ ((DecryptKey[i * 2] * 256) + DecryptKey[i * 2 + 1]);
    dx = x1a2 + i;
    ax = x1a0[i + 1];
    cx = 0x015A;
    bx = 0x4E35;
    exchange(&ax, &si);
```

```
        exchange(&ax, &dx);
        if (ax != 0)
        {
           ax = ax * bx;
        }
        exchange(&ax, &cx);
        if (ax != 0)
        {
           ax = ax * si;
           cx = ax + cx;
        }
        exchange(&ax, &si);
        ax = ax * bx;
        dx = cx + dx;
        ax = ax + 1;
        x1a2 = dx;
        x1a0[i + 1] = ax;
        inter = inter ^ (ax ^ dx);
    }
}

//加密数据块
void EncryptBlock(unsigned char *buf, unsigned int nSize)
{
        unsigned int n;
        unsigned char i;
        DecryptInit();
        for( n = nSize; n > 0; n--)
        {
                PC1assemble128();
                cfc = inter >> 8;
                cfd = inter & 255;
                for(i = 0; i < 16; i++)
                        DecryptKey[i] = DecryptKey[i] ^ (*buf);
                *buf = *buf ^ (cfc ^ cfd);
                buf++;
        }
}
//解密一个字节
unsigned char PC1Dec128Byte(unsigned char c)
{
   unsigned char i;

   PC1assemble128();
```

```
    cfc = inter >> 8;
    cfd = inter & 255;
    c = c ^ (cfc ^ cfd);

    for(i = 0; i < 16; i++)
        DecryptKey[i] = DecryptKey[i] ^ c;

    return c;
}
//解密数据块
void DecryptBlock(unsigned char *buf, unsigned int nSize)
{
    unsigned int n;

    DecryptInit();

    for( n = nSize; n > 0; n--)
    {

        *buf = PC1Dec128Byte(*buf);
        buf++;
    }
}
```

〈结束〉