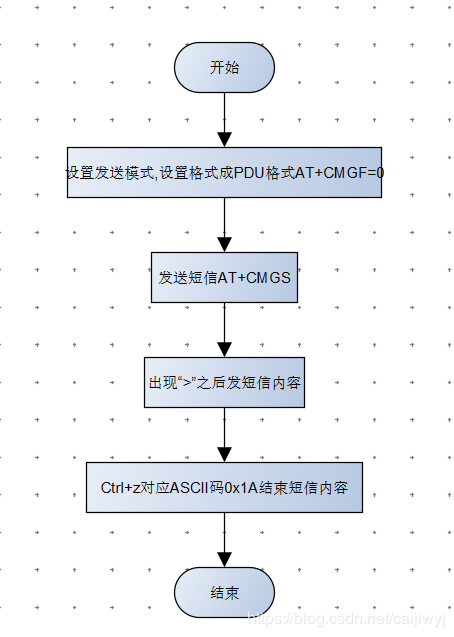
**AT指令发送TEXT格式和PDU格式短信,以及linux c编程实现3G模块发送短信**

原创 [WuYuJun's blog](https://me.csdn.net/caijiwyj) 发布于2019-07-26 18:55:52 阅读数 229 收藏

展开

发送短消息常用Text和PDU(Protocol   Data   Unit，协议数据单元)模式。使用Text模式收发短信代码简单，实现起来十分容易，但最大的缺点是不能收发中文短信；而PDU模式不仅支持中文短信，也能发送英文短信。PDU模式收发短信可以使用3种编码：7-bit、8-bit和UCS2编码。7-bit编码用于发送普通的ASCII字符，8-bit编码通常用于发送数据消息，UCS2编码用于发送Unicode字符。

**发送短息流程：**



发送TEXT格式与PDU格式的短信都是这个流程，不过发送短信AT+CMGS这里会有不同，发送TEXT格式短信AT+CMGS=”${手机号码}”,而发送PDU格式的短信AT+CMGS=${信息长度},后面也大致相同，只是发送的内容PDU格式的数据需要进行比较复杂的转换，下面慢慢说

**发送TEXT格式短信：**

**(1)首先先设置发送模式为TEXT格式**

AT+CMGF=1

OK

查看当前发送模式

AT+CMGF?

+CMGF: 1

OK

**(2)发送短信**

发送TEXT格式短信是AT+CMGS=”${手机号码}”,出现”>”之后输入短信内容

AT+CMGS="86189349355xx"

> Hello World

+CMGS: 116

OK

在最后结束输入是按Ctrl+z结束输入，对应的ASCII码为0x1A

发送英文短信到这里就完成了

如果需要使用TEXT模式发送中文内容，请先使用以下命令确认是否支持设置TEXT模式编码：

1.1 AT+CSCS=? 查询支持哪些编码

AT+CSCS=?

+CSCS: ("GSM","HEX","IRA","PCCP437","UCS2","8859-1")

OK

如上至少支持UCS2编码才能使用，HEX模式也可以，但是建议使用UCS2。具体格式以模块厂家的实际设计

**发送TEXT格式短信代码：**

1. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. \*  描述：发TEXT格式短信
3. \*
4. \*  返回值：
5. \*          成功返回0, 失败返回负数
6. \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
8. int send\_mess(const char \*name,const char \*phone\_num,const char \*sms)
10. {


14. int             rv = 0 ;
16. char            rbuf[1024] ;
18. char            mess[512] ;
20. char            at\_cmgs[128] ;
22. st\_comport      \*comport = NULL ;


26. comport = comport\_init(name, 115200,8,'N',1,'N') ;
28. if(!comport)
30. {
32. printf("comport\_init() failed\n") ;
34. rv = -1 ;
36. goto cleanup;
38. }
40. comport->fd = open\_comport(comport) ;
42. if(comport->fd < 0)
44. {
46. printf("Open\_comport() Failed\n") ;
48. rv = -1 ;
50. goto cleanup ;
52. }


56. if((!phone\_num) || (!sms))
58. {
60. printf("Invail input paremeter in %s\n",\_\_FUNCTION\_\_) ;
62. rv = -2 ;
64. goto cleanup ;
66. }


70. /\*     设置信息格式为TEXT方式  \*/
72. rv = send\_cmd(comport, "AT+CMGF=1\r", rbuf, sizeof(rbuf), 800);
74. if(rv < 0 )
76. {
78. printf("AT+CMGS FAILED\n");
80. rv = -3 ;
82. goto cleanup;
84. }
86. memset(at\_cmgs,0,sizeof(at\_cmgs)) ;
88. snprintf(at\_cmgs, sizeof(at\_cmgs),"AT+CMGS=\"%s\"\r", phone\_num) ;
90. rv = send\_cmd(comport, at\_cmgs, rbuf, sizeof(rbuf), 1200);
92. if(rv < 0)
94. {
96. printf("AT+CMGS FAILED\n");
98. rv = -4 ;
100. goto cleanup;
102. }
104. if(strstr(rbuf,">") == NULL)
106. {
108. printf("Can't receive \">\"\n") ;
110. rv = -5 ;
112. goto cleanup ;
114. }
116. memset(mess, 0, sizeof(mess));
118. snprintf(mess, sizeof(mess), "%s",sms) ;
120. strcat(mess,"\x1a");
122. rv = send\_cmd(comport, mess, rbuf, sizeof(rbuf), 3500);
124. if(rv < 0 )
126. {
128. printf("send message FAILED\n");
130. rv = -5 ;
132. goto cleanup;
134. }
136. if(!strstr(rbuf, "OK"))
138. {
140. printf("Send Message Failed\n") ;
142. rv = -6 ;
144. goto cleanup ;
146. }
148. rv = 0 ;
150. cleanup:
152. comport\_term(comport) ;
154. return rv ;
156. }

**发送PDU格式的中文短信：**

参考：<https://blog.csdn.net/a_tu_/article/details/47808017>

**(1)设置发送模式为PDU格式**

AT+CMGF=0

OK

查看当前发送模式

AT+CMGF?

+CMGF: 0

OK

**(2)发送短信**

这里和发送TEXT格式短信不同，TEXT格式后面跟的手机号码，而这里跟的是长度AT+CMGS=${长度}，在发送这个AT指令之前先不要着急，需要先做准备工作，而最复杂的也就是再这了

首先得先知道短信中心号码

使用AT指令AT+CSCA?查询短信中心号码，如果查不到，可以百度当地的短信中心号码

AT+CSCA?

+CSCA: "+8613010710500",145

这里得到的短信中心号码是+8613010710500我百度了一下，嗯是湖北武汉地区联通的短信中心的号码，都可以百度得到

发送对方的手机号码也得需要知道，假设发送的手机号码为：+8618934935552

发送的内容：测试发中文短信

列出来看得清楚一点

短信中心号码:+8613010710500

发送手机号码:+8618934935552

发送短信内容:测试发中文短信

**第一步——格式化短信中心号码**

**1 - 要去掉“+”号，然后看看长度要是奇数的话就在末尾追加一个“F”。**

    +8613010710500 ——> **8613010710500F**

**2 - 将奇数位和偶数位交换位置。**

    8613010710500F ——> **683110700105F0**

**3 - 在前面加上“91”，是国际化的意思。**

    683110700105F0 ——> **91683110700105F0**

**4 - 最后算一算它有多长了，把长度除于2，再把结果转换为2位的16进制数，加在最前面。**

    91683110700105F0长16,16/2=8,8的16进制“08” 一定要注意是2位

91683110700105F0 ——> **0891683110700105F0**

短信中心号码就处理完了，处理完的短信中心号码：**0891683110700105F0**

**第二步——格式化收信号码**

**1 - 要去掉“+”号，然后看看长度要是奇数的话就在末尾追加一个“F”。**

    +8618934935552 ——> **8618934935552F**

**2 - 将奇数位和偶数位交换位置。**

    8618934935552F ——> **688139945355F2**

**第三步——格式化发送内容**

**1 - 把字符串转换为Unicode格式。**

我这里找了个在线中文转Unicode编码的网站转换的，如果失效了可以自己百度找其他的:

[http://tool.sufeinet.com/Code/ChineseUnicode.aspx?t=1&str=%u6D4B%u8BD5%u53D1%u4E2D%u6587%u77ED%u4FE1](http://tool.sufeinet.com/Code/ChineseUnicode.aspx?t=1&str=%25u6D4B%25u8BD5%25u53D1%25u4E2D%25u6587%25u77ED%25u4FE1)

得到的结果：\u6d4b\u8bd5\u53d1\u4e2d\u6587\u77ed\u4fe1，去掉\u，一个中文编码后长度为为4，找\u可以看得清楚一点 ,4\*7=28,长度为28

测试发中文短信 ——> **6d4b8bd553d14e2d658777ed4fe1**

**2 - 把串Unicode码的长度除于2，再把结果转换为2位的16进制数，加在最前面。**

6d4b8bd553d14e2d658777ed4fe1长28,28/2=14,14的16进制”0E”;

6d4b8bd553d14e2d658777ed4fe1 ——> **0E6d4b8bd553d14e2d658777ed4fe1**

**看看我们都有了什么东西**

处理完的短信中心号码：**0891683110700105F0**

格式化后收信方的手机号码：**688139945355F2**

格式化后的发送内容：**0E6d4b8bd553d14e2d658777ed4fe1**

**第四步——组合处理**  
**1 - 算一下收信号码的长度，不包括“+”号。这个收信号码指的是+8618934935552**  
+8618934935552**——> 8618934935552 长度为“OD”（2位16进制数表示）**  
**2 - 在长度前后分别加“1100”。**  
**OD ——> 1100OD**  
**3 - 如果收信号码是手机的话就在后面追加“91”，如果是小灵通的话就要追加“81”。**  
**+8618934935552是手机号码所以加“91”**  
**11000D ——> 11000D91**  
**4 - 11000D91 + 格式化后的收信号码 + “000800” + 格式化后的发送内容**  
**11000D91 ——> 11000D91688139945355F20008000E6d4b8bd553d14e2d658777ed4fe1**  
**5 - 计算上面那一大字符串的长度除于2，10进制表示。**  
**11000D91688139945355F20008000E6d4b8bd553d14e2d658777ed4fe1长58 58/2=29**

**“1100”和“000800”分别是收信方号码段固定的前缀和后缀**

**到这里该准备的东西终于准备好了**

格式化后的短信中心：**0891683110700105F0**

    实际的发送内容：**11000D91688139945355F20008000E6d4b8bd553d14e2d658777ed4fe1**

发送内容的长度：29

终于可以发短信了！

**查看发送模式，0是PDU格式**

AT+CMGF?

+CMGF: 0

OK

**发送指令AT+CMGS=${信息长度}发送内容的长度刚计算得到的29**

AT+CMGS=29

> **11000D91688139945355F20008000E6d4b8bd553d14e2d658777ed4fe1**(**出现”>”后输入实际发送内容，结束输入Ctrl+z**)

+CMGS: 119

OK

**成功收到短信**



有点尴尬，刚刚忘记加上短信中心号码就发出去了，但也能收到，如果没加短信中心号码应该是会用AT+CSCA?查询到的短信中心号码。

再来一遍：

AT+CMGS=29

> **0891683110700105F011000D91688139945355F20008000E6d4b8bd553d14e2d658777ed4fe1**

+CMGS: 124

OK



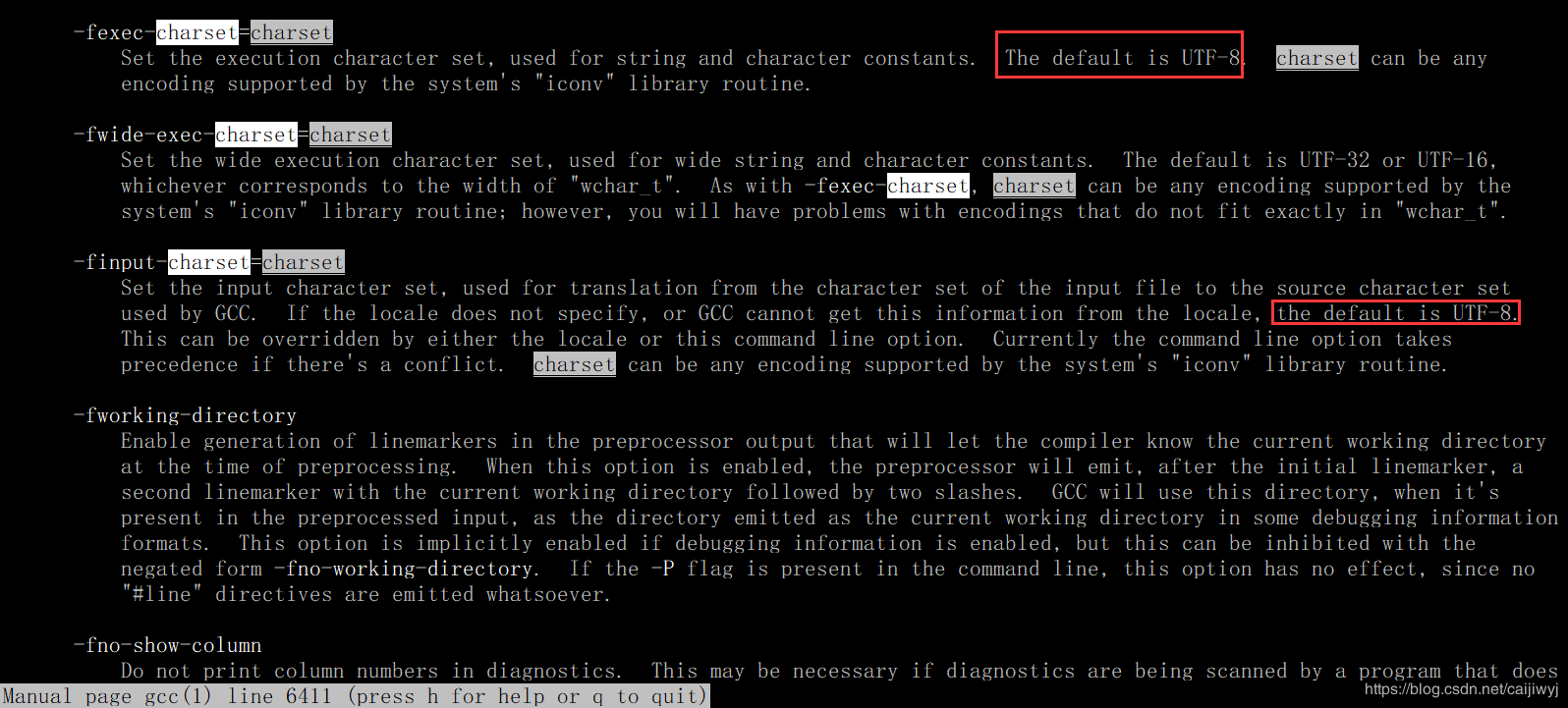
到这用AT指令发PDU格式中文短信就完成了

本来想编码实现,但难点在中文字符串转Unicode,百度找了挺多，我最先看到的是：<https://yq.aliyun.com/articles/10534>

这里的源码转载至上面的链接<https://yq.aliyun.com/articles/10534>

1. #include <stdio.h>
3. #include <string.h>
4. #include <math.h>
5. #include <wchar.h>
7. int main()
8. {
9. wchar\_t \*chinese\_str = L"网络人VS灰鸽子";
10. unsigned int \*p = (wchar\_t \*)chinese\_str;
11. int i;
12. printf("Uniocde: \n");
13. for (i = 0; i < wcslen(chinese\_str); i++)
14. {
15. printf("0x%x ", p[i]);
17. }
19. printf(" \n");
21. return 0 ;
23. }

确实可行，他这是固定的wchar\_t类型的字符串，而我们发短信的内容是会变的而不是固定的，所以我就去搜索了wchar\_t,百度百科是这么说的wchar\_t是C/C++的字符类型，是一种扩展的存储方式。wchar\_t类型主要用在国际化程序的实现中，但它不等同于unicode编码。unicode编码的字符一般以wchar\_t类型存储。而这个程序看上面的链接，应该是利用gcc的特性，gcc的编码格式、解析编码和运行编码默认用UTF-8



Unicode与UTF-8的关系：Unicode只是一个符号集，它只规定了符号的二进制代码，却没有规定这个二进制代码应该如何存储UTF-8是Unicode的实现方式之一，UTF-8在互联网上使用最广的一种unicode的实现方式。

短信内容会变，用上面的代码涉及到宽字符(wchar\_t类型)和多字节字符( 一个字符用一个或者多个字节表示) 转换。

**把多字节字符转换成宽字符：**

size\_t mbstowcs(wchar\_t \*wcstr,const char \*mbstr,size\_t count );

**把宽字符把转换成多字节字符串：**

size\_t wcstombs(char \*mbstr,const wchar\_t \*wcstr,size\_t count);

将上面的代码修改了一下，将输入的第二个参数字符串转换转换成宽字符，然后输出Unicode码：

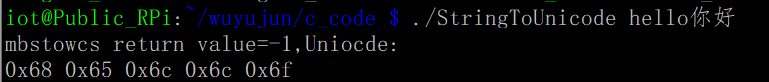
1. #include <stdio.h>
3. #include <stdlib.h>
5. #include <string.h>
7. #include <math.h>
9. #include <wchar.h>
11. #include <locale.h>

14. #define LEN 256
16. int main(int argc, char \*\*argv)
18. {
20. wchar\_t         chinese\_str[LEN];
22. int             i;
24. int             rv = 0 ;


28. if(argc < 2)
30. {
32. printf("Usage %s ${The Chinese string you want to convert}\n}", argv[0]) ;
34. return -1 ;
36. }
38. setlocale(LC\_ALL, "zh\_CN.utf8");
40. wmemset(chinese\_str,0,LEN) ;
42. rv = mbstowcs(chinese\_str,argv[1],LEN) ;
44. /\*  swprintf(chinese\_str, LEN, L"%S",L"你好") ;\*/
46. printf("mbstowcs return value=%d,Uniocde: \n",rv);
48. for (i = 0; i < wcslen(chinese\_str); i++)
50. {
52. printf("0x%x ",(unsigned int)chinese\_str[i]);
54. }
56. printf(" \n");


60. return 0 ;
62. }

一开始我是在树莓派上编译运行，运行结果：



英文部分字符转换成宽字符成功，而中文部分转换失败了，原来的代码运行没问题，而我也只是加了将char类型字符转换成wchar\_t类型字符，所以问题出在mbstowcs()，网上查别人直接用mbstowcs直接成功的，纠结了好久才找到一个帖子，linux下用mbstowcs()将汉字转成unicode失败：<http://bbs.chinaunix.net/thread-1307991-1-1.html>

才明白mbstowcs与locale有关，需要确定源串是什么编码，中文字符串需要是汉字字符串编码为UTF-8，也就是zh\_CN.UTF-8，代表中国人使用的unicode字符集。

而locale又是什么呢，最初装系统的时候见过，也没在意所有根本不知道是什么东西，也是去网上查了一波，下面摘至：<https://www.cnblogs.com/idlo/p/10427409.html>

locale这个单词中文翻译成地区或者地域，其实这个单词包含的意义要宽泛很多。Locale是根据计算机用户所使用的语言，所在国家或者地区，以及当地的文化传统所定义的一个软件运行时的语言环境。

locale把按照所涉及到的文化传统的各个方面分成12个大类，这12个大类分别是：

1、语言符号及其分类(LC\_CTYPE)

2、数字(LC\_NUMERIC)

3、比较和排序习惯(LC\_COLLATE)

4、时间显示格式(LC\_TIME)

5、货币单位(LC\_MONETARY)

6、信息主要是提示信息,错误信息,状态信息,标题,标签,按钮和菜单等(LC\_MESSAGES)

7、姓名书写方式(LC\_NAME)

8、地址书写方式(LC\_ADDRESS)

9、电话号码书写方式(LC\_TELEPHONE)

10、度量衡表达方式 (LC\_MEASUREMENT)

11、默认纸张尺寸大小(LC\_PAPER)

12、对locale自身包含信息的概述(LC\_IDENTIFICATION)。

除了这12个变量可以设定以外，为了简便起见，还有两个变量：LC\_ALL和LANG。它们之间有一个优先级的关系：LC\_ALL > LC\_ >LANG。可以这么说，LC\_ALL是最上级设定或者强制设定，而LANG是默认设定值。

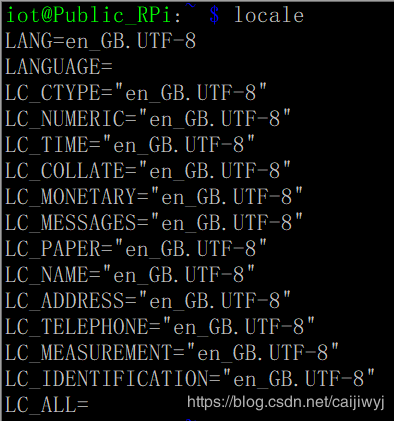
另外LANG和LANGUAGE有什么区别呢？

LANG - Specifies the default locale for all unset locale variables

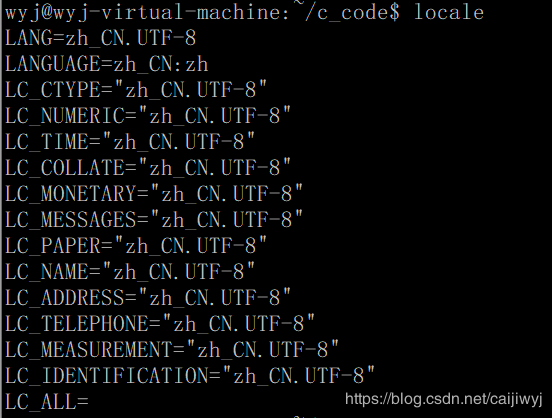
LANGUAGE - Most programs use this for the language of its interface

LANGUAGE是设置应用程序的界面语言。而LANG是优先级很低的一个变量，它指定所有与locale有关的变量的默认值。

树莓派查看locale,好的全是英文

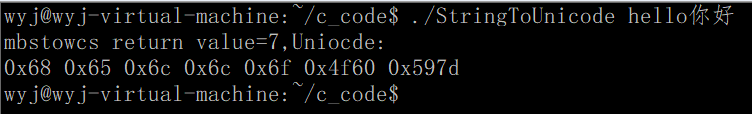


而我虚拟机下的Ubuntu的locale

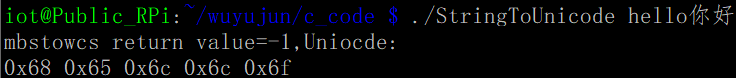


是zh\_CN.UTF-8 的语言环境

在我虚拟机Ubuntu14.04运行的结果：



在树莓派上运行的结果：



在线中文转Unicode格式的结果：



安装和设置中文字符集网上一搜一抓一大把，也就不说了，感觉说偏了很多，主要还是自己不懂的太多，博客就当是自己做的记录吧哈哈哈。

关于编码还不清楚，linux有iconv的库用来做编码的转换，这里先记一下，以后可能会用到

iconv函数族也就三个函数iconv\_open(),iconv(),iconv\_close(),原型如下:

iconv\_t iconv\_open(const char \*tocode, const char \*fromcode);

此函数说明将要进行哪两种编码的转换,tocode是目标编码,fromcode是原编码,该函数返回一个转换句柄,供以下两个函数使用。

size\_t iconv(iconv\_t cd,char \*\*inbuf,size\_t \*inbytesleft,char \*\*outbuf,size\_t \*outbytesleft);

此函数从inbuf中读取字符,转换后输出到outbuf中,inbytesleft用以记录还未转换的字符数,outbytesleft用以记录输出缓冲的剩余空间。

int iconv\_close(iconv\_t cd);

此函数用于关闭转换句柄,释放资源。

有关iconv的博客：<https://www.cnblogs.com/etangyushan/p/3753847.html>

中文转Unicode编码或者iconv有关转换中文的编码都需要语言环境支持中文字符集，所以一般不会交给板子去做，都是PC端转好去给板子直接去用就行了，中文转Unicode搞定之后就是简单的装包了

**首先是交换相邻的奇偶位**

1. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. \*  Describe:swap strings the odd position and even position of the neighbour
3. \*  Input parameter:
4. \*                  const char \*str, the string you want to swap
5. \*                  int str\_len, strlen of the string , it must be even
6. \*                  int out\_str, char \*str\_out size, output string size
7. \*  Output parameter:
8. \*                  char \*str, the string after swap output to char \*str\_out
9. \*  Return:Successful return 0, Fail return negative
10. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
12. int swap\_odd\_even(const char \*str,int str\_len, char \*str\_out, int out\_size)
14. {
16. int         i ;
18. char        temp ;


22. if( !str || str\_len <= 0 || !str\_out|| out\_size < str\_len)
24. {
26. printf("Invail input parameter in %s\n",\_\_FUNCTION\_\_) ;
28. return -1 ;
30. }
32. if( str\_len % 2 )//The string is odd
34. {
36. printf("Please input a size of string is even\n") ;
38. return -2 ;
40. }
42. snprintf(str\_out, out\_size, "%s", str) ;
44. for( i = 0; i < str\_len-1; i = i+2 )
46. {
48. temp = str\_out[i] ;
50. str\_out[i] = str\_out[i+1] ;
52. str\_out[i+1] = temp ;
54. }
56. return 0 ;
58. }

**然后是格式化手机号码，不管是短信中心号码还是收信号码，都是如果开头有“+”去掉“+”，如果长度是奇数，在末尾加个“F”,相邻奇偶位交换**

1. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. \*  Describe:fomatt phone\_number for send pdu sms , Delete '+',
3. \*      if string is odd add 'F', and swap the odd position and even position
4. \*  Input parameter:
5. \*                  char \*phone\_num,phone\_number string you want to format
6. \*  Output parameter:
7. \*                  char \*out\_str, After fomat phone number string output to out\_str
8. \*  Return :
9. \*          Successful retrun 0; Fail return negative
10. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
12. int format\_phone\_num(const char \*phone\_num, char \*out\_str, int out\_size)
14. {
16. const char      \*ptr = NULL ;
18. int             len = 0 ;
20. char            temp\_out[64] ;
22. int             rv ;


26. if(!phone\_num || !out\_str || out\_size<=0 )
28. {
30. printf("Invail parameter in %s\n",\_\_FUNCTION\_\_) ;
32. return -1 ;
34. }
36. ptr = phone\_num ;
38. if(ptr[0] == '+') //if have '+' Delete '+'
40. {
42. ptr = ptr+1 ;
44. if( !(\*ptr) )//if ptr point end of string "/0"
46. {
48. printf("Invail phone number\n") ;
50. return -2 ;
52. }
54. }
56. memset(temp\_out, 0, sizeof(temp\_out)) ;
58. snprintf(temp\_out, sizeof(temp\_out), "%s", ptr) ;
60. len = strlen(ptr) ;
62. if( (len % 2) != 0) //if phone number string is odd, add 'F' in the last
64. {
66. strncat(temp\_out, "F", sizeof(temp\_out)) ;
68. }
70. rv = swap\_odd\_even(temp\_out, strlen(temp\_out), out\_str, out\_size) ;
72. if(rv < 0)
74. {
76. printf("swap phone number strings the odd position and even position of the neighbour failed\n") ;
78. return -3 ;
80. }


84. return 0 ;
86. }

**格式化短信中心号码，需要将转化格式后（也就是上面的去掉“+”加上“F”）,计算长度除2以16进制加在最开头**

1. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. \*  Describe:format center sms phone number for send pdu sms
3. \*  Input parameter:
4. \*              const char \*center\_num is string of center sms phone number
5. \*  Output parameter:
6. \*              char \*out\_str is after format center sms phone number output
7. \*  Return:
8. \*          OK return 0; Fail return negative
9. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
11. int format\_center\_num(const char \*center\_num,char \*out\_str, int out\_size)
13. {
15. char            temp\_str[64] ;
17. int             rv = -1 ;
19. int             length = 0 ;


23. if( !center\_num || !out\_str || out\_size <= 0)
25. {
27. printf("Invail parameter in %s\n", \_\_FUNCTION\_\_) ;
29. return -1 ;
31. }


35. memset(temp\_str, 0, sizeof(temp\_str)) ;
37. rv = format\_phone\_num(center\_num, temp\_str, sizeof(temp\_str)) ;
39. if(rv < 0)
41. {
43. printf("format\_phone\_num %s fail in %s\n",center\_num, \_\_FUNCTION\_\_) ;
45. return -2 ;
47. }


51. length = strlen(temp\_str) ;
53. length = length + 2 ; //Add the number 91 logo of Internationalization in the head
55. length = length/2 ;//Length divided by 2(hex values)
57. if(length < 16)
59. {
61. snprintf(out\_str, out\_size, "0%x91%s", length, temp\_str) ;
63. }
65. else
67. {
69. snprintf(out\_str, out\_size,"%x91%s", length, temp\_str);
71. }


75. return 0 ;
77. }

**格式化收短信的号码，需要先加上91表示是手机号码不是小灵通号码，现在应该没有小灵通了，所有我直接写死了固定的91，计算原始去掉“+”后手机号码长度加在前面，1100作为固定的收信前缀，008000收信号码后缀**

1. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. \*  Describe:format receiver sms phone number for send pdu sms
3. \*  Input parameter:
4. \*              const char \*rec\_num is string of receiver sms phone number
5. \*  Output parameter:
6. \*              char \*out\_str is after format receiver sms phone number output
7. \*  Return:
8. \*          OK return 0; Fail return negative
9. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
11. int format\_receive\_num(const char \*rec\_num,char \*out\_str, int out\_size)
13. {
15. const char      \*ptr = NULL ;
17. char            temp\_str[64] ;
19. int             rv = -1 ;
21. int             length = 0 ;


25. if( !rec\_num || !out\_str || out\_size <= 0)
27. {
29. printf("Invail parameter in %s\n", \_\_FUNCTION\_\_) ;
31. return -1 ;
33. }
35. memset(temp\_str, 0, sizeof(temp\_str)) ;
37. rv = format\_phone\_num(rec\_num, temp\_str, sizeof(temp\_str)) ;
39. if(rv < 0)
41. {
43. printf("format\_phone\_num %s fail in %s\n",rec\_num, \_\_FUNCTION\_\_) ;
45. return -2 ;
47. }


51. ptr = rec\_num ;
53. if(ptr[0] == '+')
55. {
57. ptr = ptr+1 ;
59. if( !(\*ptr) )
61. {
63. printf("invail receive phone number:%s\n",rec\_num) ;
65. return -3 ;
67. }
69. }
71. length = strlen(ptr) ; //Length of the receive phone number
73. /\* "1100" and "000800" are fixed prefixes and suffixes of the receiver phone number
74. \*  if the number of the receiver is mobile phone, it will add "91", if it is PHS add "81"\*/
76. if(length < 16)
78. {
80. snprintf(out\_str, out\_size, "11000%x91%s000800", length, temp\_str) ;
82. }
84. else
86. {
88. snprintf(out\_str, out\_size,"1100%x91%s000800", length, temp\_str);
90. }
92. return 0 ;
94. }

**最后就是组合到一起，这里短信内容格式需要是中文转Unicode格式之后的字符串。**

1. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. \*  Describe:format receiver sms phone number for send pdu sms
3. \*  Input parameter:
4. \*              const char \*center\_num is string of center sms phone number
5. \*              const char \*rec\_num is string of receiver sms phone number
6. \*              const char \*mess is message content by Unicode ,mess content must be Unicode!!!
7. \*  Output parameter:
8. \*              char \*out\_str is after packet pdu sms
9. \*  Return:
10. \*          OK return send length for AT+CMGS=${Send Length}; Fail return negative
11. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
13. int packet\_pdu\_sms(const char \*center\_num, const char \*rec\_num, const char \*mess, char \*out\_str, int out\_size)
15. {
17. int     rv = -1 ;
19. char    center[64] ;
21. char    receiver[64] ;
23. char    sms\_mess[512] ;
25. int     sms\_length = 0;
27. int     cmgs\_len = 0 ;




33. if(!center\_num || !rec\_num || !mess || !out\_str || out\_size <= 0)
35. {
37. printf("Invail paremeter in %s\n", \_\_FUNCTION\_\_) ;
39. return -1 ;
41. }
43. memset(center, 0, sizeof(center)) ;
45. rv = format\_center\_num(center\_num, center, sizeof(center)) ;
47. if(rv < 0)
49. {
51. printf("format center sms phone number error\n") ;
53. return -2 ;
55. }


59. memset(receiver, 0, sizeof(receiver)) ;
61. rv = format\_receive\_num(rec\_num, receiver, sizeof(receiver)) ;
63. if(rv < 0)
65. {
67. printf("format receiver phone number error\n") ;
69. return -3 ;
71. }


75. sms\_length = strlen(mess) ;
77. sms\_length = sms\_length/2 ;
79. memset(sms\_mess, 0, sizeof(sms\_mess)) ;
81. if(sms\_length<16)
83. {
85. snprintf(sms\_mess, sizeof(sms\_mess),"0%x%s",sms\_length,mess) ;
87. }
89. else{
91. snprintf(sms\_mess, sizeof(sms\_mess),"%x%s",sms\_length,mess) ;
93. }
95. snprintf(out\_str, out\_size,"%s%s%s", center,receiver,sms\_mess) ;
97. cmgs\_len = strlen(receiver)+strlen(sms\_mess) ;
99. cmgs\_len = cmgs\_len/2;


103. return cmgs\_len ;
105. }

**用于测试的代码**

1. int main(int argc, char \*\*argv)
3. {
5. int     rv = -1 ;
7. char    sms[512] ;

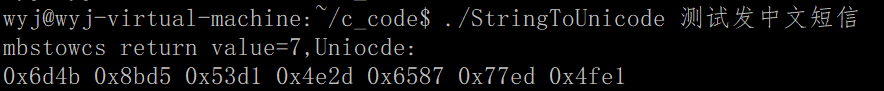

11. if(argc < 4 )
13. {
15. printf("Usage %s ${center\_sms\_number} ${receiver phone number} ${sms content by unicode}\n}", argv[0]) ;
17. return -1 ;
19. }


23. memset(sms, 0, sizeof(sms)) ;


27. rv = packet\_pdu\_sms(argv[1], argv[2], argv[3], sms, sizeof(sms)) ;
29. if(rv < 0)
31. {
33. printf("format error\n") ;
35. return -2 ;
37. }
39. printf("After packet,length = %d,pdu sms:%s\n",rv,sms) ;

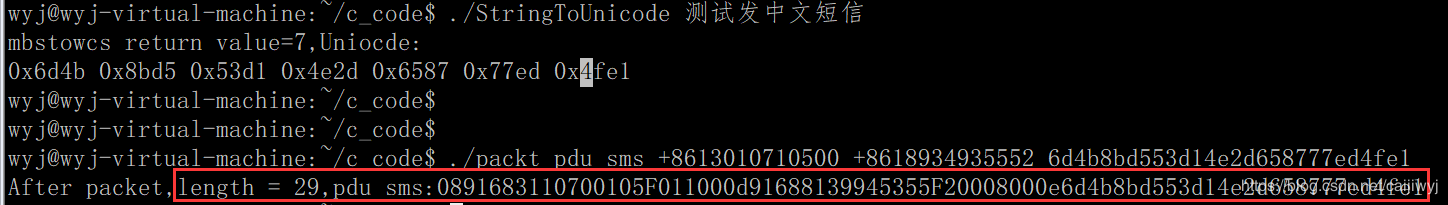

43. return 0 ;
45. }

**先将发送短信的内容用之前的程序转换成Unicode格式：**



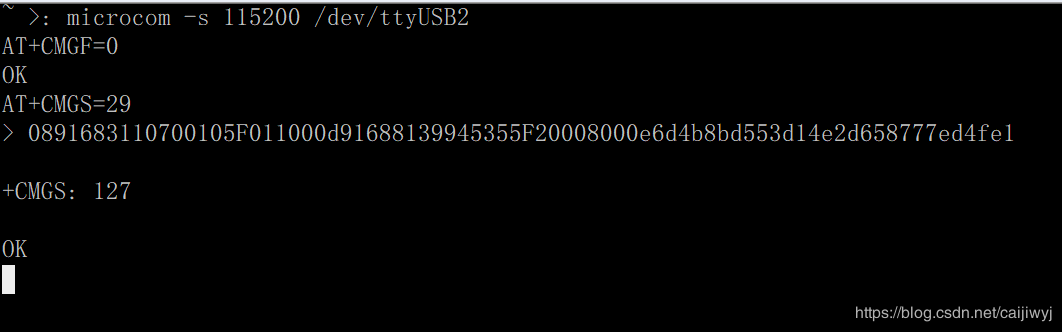
6d4b8bd553d14e2d658777ed4fe1

**用测试程序获取PDU装包后的数据**



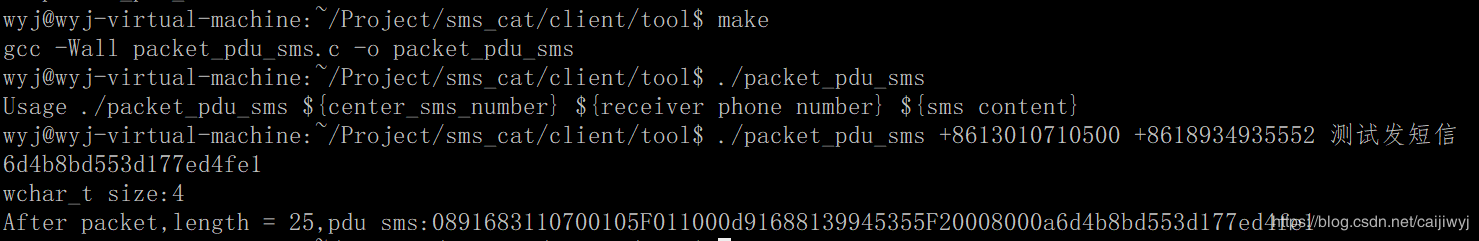
得到的信息：length = 29,pdu sms:0891683110700105F011000d91688139945355F20008000e6d4b8bd553d14e2d658777ed4fe1

将得到的东西，PDU短信用AT命令发送：

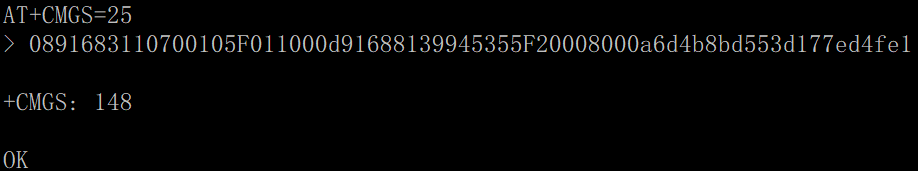


发送成功

更新：2019/8/1 将中文转Unicode和装包写到一个程序里面



用AT指令发送



中英文混合编码的时候，Unicode编码里4E00-9FBB是汉字，如果是英文前面要加上00