

自动识别系统中中文编码规则

1 概述

ITU-R M.1371-1 《在 VHF 海上移动频段采用时分多址 (TDMA) 技术的通用船载自动识别系统 (AIS) 的技术特性》规定, 消息中收发的字符应采用其《6 位 ASCII 字符码和 8 位标准 ASCII 字符码的对照表》(以下简称对照表) 中的 6 位 ASCII 码。本方案明确了在消息中使用汉字交换信息的方法, 以使规定使用 6 位 ASCII 码收发信息的船载自动识别系统, 能用汉字代码收发信息。

本方案适用于消息 6 和消息 8 内用中文表示的信息内容, 其它有收发汉字需要的消息可参照使用。

2 代码定义

(1) 收发中文信息的应用标示码

消息 6 和消息 8 中二进制数据域内, 应用标示码中 DAC 给定为 413, FI 给定为 01, 让收发双方知道以下收发的是含汉字代码的收发代码。

(2) 机内码

机内码是一般嵌入式计算机拥有的含汉字代码的编码, 即单字节 8 位二进制数, 第 8 位为“0”, 表示标准 ASCII 字符代码; 双字节 16 位二进制数, 两个字节第 8 位都为“1”, 表示 GB2312 基本集汉字代码。机内码用于船载自动识别系统计算机的字符和汉字的存储、处理。

(3) 收发代码

收发代码是一种含 13 位 GB2312 基本集汉字代码的编码, 其具体

组成和与机内码的互相转换见第 4 节；收发代码是 AIS 在收发信息时使用的 ASCII 字符代码和汉字代码。

3 收发双方约定

(1) 船载自动识别系统收发消息时，其初始字符编码默认采用对照表的 6 位 ASCII 码；

(2) 当且仅当消息 6 和消息 8 收发的应用标示码中 DAC 为 413 且 FI 为 01 后，船载自动识别系统收发的字符编码采用含汉字代码的收发代码；消息收发结束，系统恢复默认采用对照表的 6 位 ASCII 码。

4 收发代码及其同机内码的互相转换

收发代码用 1 个 7 位码表示 ASCII 字符代码，即第 7 位（最高位）为“0”，余下 6 位为对照表的 6 位 ASCII 码；收发代码用 2 个 7 位码共 14 位表示汉字代码，即第 14 位（最高位）为“1”，余下 13 位为表示 GB2312 基本集的汉字 13 位代码。

收发代码和机内码的转换方法见附件。

附件

收发代码和机内码的转换方法

1 收发代码和机内码互相转换的方法

(1) 机内码转换为收发代码的方法

取一个字节 8 位的机内码记为 x_1 ;

如果 x_1 的第 8 位为“0”，则为标准 ASCII 码，按下列转换算法转换为 7 位表示 ASCII 字符的收发代码 y_1 :

$$y_1 = \begin{cases} x_1, & x_1 < 0x40; \\ x_1 - 0x40, & x_1 \geq 0x40; \end{cases}$$

式中:

y_1 —— 7 位有效;

$0x$ —— 表示其后是 16 进制数字, 下同。

如果 x_1 的第 8 位为“1”，则为汉字代码，取第二个字节 8 位机内码 x_2 ，将 x_1 、 x_2 的第 8 位清成“0”， x_1 、 x_2 即为机内码表示汉字的 14 位代码，第一字节 x_1 记为 A ，低 7 位有效，第二字节 x_2 记为 B ，低 7 位有效；转换成收发代码表示汉字的 13 位代码，第一字节记为 a ，低 6 位有效，第二字节记为 b ，低 7 位有效；转换算法如下:

$$a = \begin{cases} (A - 0x30) * 4 + |B / 0x20| \text{取商}, & A < 0x40; \\ A - 0x40, & A \geq 0x40; \end{cases} \quad (\text{算式一})$$

$$b = \begin{cases} B \% 0x20 & A < 0x40; \\ B, & A \geq 0x40; \end{cases} \quad (\text{算式二})$$

式中:

* —— 乘法;

/ —— 除法;

$|x/y|$ 取商 —— x 除以 y ，结果仅用整数商;

$x \% y$ —— x 除以 y ，结果仅用余数;

将 13 位汉字代码 a 、 b ，其字节 a 的第 7 位置“1”，并将 a 、 b 记为 y_1 、 y_2 ，即构成 14 位表示汉字的收发代码。

(2) 收发代码转换为机内码的方法

取一个 7 位的收发代码记为 y_1 ;

如果 y_1 的第 7 位为 “0”，则表示 6 位 ASCII 码，按下列转换算法转换为 8 位表示 ASCII 字符的机内码 x_1 ：

$$x_1 = \begin{cases} y_1 + 0x40, & y_1 < 0x20; \\ y_1, & y_1 \geq 0x20; \end{cases}$$

如果 y_1 的第 7 位为 “1”，则为汉字代码，取第二个 7 位的收发代码 y_2 ，将第 1 字节 y_1 的第 7 位清 “0”， y_1 、 y_2 即为收发代码表示汉字的 13 位代码，第一字节 y_1 记为 a ，低 6 位有效，第二字节 y_2 记为 b ，低 7 位有效；转换成机内码表示汉字的 14 位代码第一字节记为 A ，低 7 位有效，第二字节记为 B ，低 7 位有效；转换算法如下：

$$A = \begin{cases} 0x30 + |a/4| \text{ 取商}, & b < 0x20; \\ a + 0x40, & b \geq 0x20; \end{cases} \quad (\text{算式三})$$

$$B = \begin{cases} b + (a \& 3) * 0x20, & b < 0x20; \\ b, & b \geq 0x20; \end{cases} \quad (\text{算式四})$$

式中：

* ——乘法；

/ ——除法；

$|x/y|$ 取商 —— x 除以 y ，结果仅用整数商；

$x \& y$ —— x 按位与 y 。

将 14 位汉字代码 A 、 B ，其两个字节第 8 位置 “1”，记为 x_1 、 x_2 ，即构成 16 位表示汉字的机内码。

2 实施举例

使用计算机编程语言 C 编写程序，使消息 6 和消息 8，应用标示码中 DAC 为 413 且 FI 为 01 时，用含中文信息的收发代码收发信息。本实施例显示和描述了本方案的原理和特征，实际使用不受此例的限制。

函数 `xToY_send` 和 `yToX_receive`、`AB14_to_ab13()` 和 `ab13_to_AB14()` 实施了本方案的算法；其它函数与本方案关系不大，故程序略。

```
unsigned char  a, b, AA, BB;
unsigned char  sendData[162];    //要发送的机内码。
unsigned char  receiveData[162]; //收到的已转换的机内码。
void AB14_to_ab13(); //将 14 位汉字码 AA, BB 转换为汉字 13 位码 a, b。
```

```

void ab13_to_AB14(); //将汉字 13 位码 a, b 转换为 14 位汉字码 AA、BB。

void send_8bit(unsigned char y); //将 y 的 8 位送入发送数据区，程序略。

void send_7bit(unsigned char y); //将 y 的低 7 位送入发送数据区，程序略。

unsigned char receive_7bit(); //从接收数据区取 7 位值作为返回值，程序略。

/*-----

** 函数名: xToY_send

** 功能: 在消息 6 或消息 8 发含中文信息的收发代码时调用。发出 DAC 为 413 且 FI 为 01;
将数组 sendData 中的机内码转换为收发代码，送入发送数据区。机内码的汉字代码转换为汉字 13
位码时，调用函数 AB14_to_ab13()。

** 输入: sn, 要转换的数组元素个数;
        数组 sendData, 元素是机内码。

** 输出: 函数 send_8bit(c), 把 8 位码送入发送数据区;
        函数 send_7bit(c), 把 7 位收发代码送入发送数据区。

-----*/

void xToY_send(sn)
unsigned char sn;
{
    unsigned char i, x1, x2, y1, y2;
    i=0;
    x1=0x67; x2=0x41; // DAC 为 413 且 FI 为 01 即 0x6741 送入发送数据区。
    send_8bit(x1); //把 8 位码 x1 送入发送数据区。
    send_8bit(x2); //把 8 位码 x2 送入发送数据区。
    while(i<sn) //循环直至 (sn) 个。
    {
        x1=sendData[i]; //x1 取机内码。
        i++;
        if(x1<0x80)
        { //x1 最高位为 0, x1 转换为 7 位收发代码(ASCII 码) y1。
            if(x1>=0x40) y1=x1-0x40;
            else y1=x1;

```

```

        send_7bit(y1); //把 7 位收发代码 y1 送入发送数据区。
    }
    else
    {
        //x1 最高位为 1, x1、x2 转换为收发代码(汉字码) y1、y2。
        x2=sendData[i]; //x2 取机内码。

        i++;

        AA=x1&0x7f; //x1、x2 最高位清 0, 成 14 位汉字码 AA, BB。
        BB=x2&0x7f;

        AB14_to_ab13(); //将 14 位汉字码 AA, BB 转换为汉字 13 位码 a, b。
        y1=a|0x40; //汉字 13 位码添上最高位 1, 成 14 位收发代码 y1、y2。
        y2=b;

        send_7bit(y1); //把 y1、y2 两个 7 位收发代码送入发送数据区。
        send_7bit(y2);
    }
} //循环直至 (sn) 个。
}

```

/*-----*/

** 函数名: yToX_receive

** 功能: 消息 6 或消息 8 当 DAC 为 413 且 FI 为 01 时, 接收含中文信息的收发代码时调用。
 将接收的收发代码转换为机内码, 送入数组 receiveData。汉字 13 位码转换为机内码的汉字代码时,
 调用函数 ab13_to_AB14()。

** 输入: rn, 接收数据区 7 位收发代码总数;

函数 receive_7bit(), 从接收数据区取 7 位收发代码。

** 输出: 数组 receiveData。

-----*/

```
void yToX_receive(rn)
```

```
unsigned char rn;
```

```
{
```

```
    unsigned char i, x1, x2, y1, y2;
```

```
    i=0;
```

```

while(i<rn)    //循环直至(rn)个。
{
    //7位收发代码 y1 或 y1、y2 转换为机内码。
    y1=receive_7bit()&0x7f;    //y1 从接收数据区取 7 位收发代码。
    if(y1<0x40)
    {
        //y1 第 7 位为 0, y1 转换为机内码(ASCII 码) x1。
        if(y1<0x20) x1=y1+0x40;
        else x1=y1;
        receiveData[i]=x1;    //机内码 x1 送入数组 receiveData。
        i++;
    }
    else
    {
        //y1 第 7 位为 1, y1、y2 转换为机内码(汉字码) x1、x2。
        y2=receive_7bit()&0x7f;    //y2 从接收数据区取 7 位收发代码。
        a=y1&0x3f;    //y1 第 7 位清 0, y1、y2 成汉字 13 位码 a,b。
        b=y2;
        ab13_to_AB14(); //将汉字 13 位码 a,b 转换为 14 位汉字码 AA、BB。
        x1=AA|0x80; //AA、BB 最高位置 1, 成机内码(汉字码) x1、x2。
        x2=BB|0x80;
        receiveData[i]=x1;    //机内码 x1、x2 送入数组 receiveData。
        i++;
        receiveData[i]=x2;
        i++;
    }
} //循环直至(rn)个。
}

/*-----
** 函数名: AB14_to_ab13
** 功能: 将 14 位汉字码 AA, BB 转换为 13 位汉字码 a, b。
** 输入: AA, 14 位汉字码第一字节, 低 7 位有效;
        BB, 14 位汉字码第二字节, 低 7 位有效。
*/

```

```

** 输出: a, 13 位汉字码第一字节, 低 6 位有效;
        b, 13 位汉字码第二字节, 低 7 位有效;

-----*/

void AB14_to_ab13()
{
    if (AA<0x40)
    {
        b=BB/0x20;           // (BB/0x20) 的商
        a=(AA-0x30)*4+b;
        b=BB%0x20;         // (BB/0x20) 的余数
    }
    else
    {
        a=AA-0x40;
        b=BB;
    }
}

/*-----

** 函数名: ab13_to-AB14
** 功能: 将 13 位汉字码 a, b 转换为 14 位汉字码 AA, BB。
** 输入: a, 13 位汉字码第一字节, 低 6 位有效;
        b, 13 位汉字码第二字节, 低 7 位有效;
** 输出: AA, 14 位汉字码第一字节, 低 7 位有效;
        BB, 14 位汉字码第二字节, 低 7 位有效。

-----*/

void ab13_to-AB14()
{
    if (b<0x20)
    {

```

```
        AA=0x30+a/4;
        BB=b+(a&3)*0x20;
    }
else
    {
        AA=a+0x40;
        BB=b;
    }
}
```