

宏晶官方读取 STC15F104W 单片机唯一 ID 号并用模拟串口发送显示 C 例程

原文: [http://www.stcmcu.com/datasheet/stc/STC-NEW-SOURCE-CODE/读取STC15F104W程序区ID号并用软件模拟串口输出显示\(频率为18.430MHz,波特率为38400\).RAR](http://www.stcmcu.com/datasheet/stc/STC-NEW-SOURCE-CODE/读取STC15F104W程序区ID号并用软件模拟串口输出显示(频率为18.430MHz,波特率为38400).RAR)

[\(相关阅读: 单片机唯一 ID 有什么用途? STC 单片机使用唯一 ID 保护的方法和思路\)](#)

```
/*-----*/
/* --- STC MCU Limited -----*/
/*STC15F104W 系列 读取程序区 ID 号并用软件模拟串口输出显示举例-----*/
/* --- Mobile: (86)13922805190 -----*/
/* --- Fax: 86-755-82905966 -----*/
/* --- Tel: 86-755-82948412 -----*/
/* --- Web: www.STCMCU.com -----*/
/*如果要在程序中使用此代码,请在程序中注明使用了宏晶科技的资料及程序*/
/*如果要在文章中应用此代码,请在文章中注明使用了宏晶科技的资料及程序*/
/*-----*/
```

```
//本示例在 Keil 开发环境下请选择 Intel 的 8058 芯片型号进行编译
//假定测试芯片的工作频率为 18.432MHz
```

```
#include "reg51.h"
```

```
//-----
//define baudrate const
//BAUD=256 - FOSC/3/BAUDRATE/M (1T:M=1; 12T:M=12)
//NOTE: (FOSC/3/BAUDRATE) must be greater then 98, (RECOMMEND GREATER THEN 110)
```

```
//#define BAUD 0xF400 // 1200bps @ 11.0592MHz
//#define BAUD 0xFA00 // 2400bps @ 11.0592MHz
//#define BAUD 0xFD00 // 4800bps @ 11.0592MHz
//#define BAUD 0xFE80 // 9600bps @ 11.0592MHz
//#define BAUD 0xFF40 //19200bps @ 11.0592MHz
//#define BAUD 0xFFA0 //38400bps @ 11.0592MHz
```

```
//#define BAUD 0xEC00 // 1200bps @ 18.432MHz
//#define BAUD 0xF600 // 2400bps @ 18.432MHz
//#define BAUD 0xFB00 // 4800bps @ 18.432MHz
```

```

//#define BAUD 0xFD80 // 9600bps @ 18.432MHz
//#define BAUD 0xFEC0 //19200bps @ 18.432MHz
#define BAUD 0xFF60 //38400bps @ 18.432MHz

//#define BAUD 0xE800 // 1200bps @ 22.1184MHz
//#define BAUD 0xF400 // 2400bps @ 22.1184MHz
//#define BAUD 0xFA00 // 4800bps @ 22.1184MHz
//#define BAUD 0xFD00 // 9600bps @ 22.1184MHz
//#define BAUD 0xFE80 //19200bps @ 22.1184MHz
//#define BAUD 0xFF40 //38400bps @ 22.1184MHz
//#define BAUD 0xFF80 //57600bps @ 22.1184MHz

#define ID_ADDR_RAM 0x71 //STC104W 系列ID号的存放在RAM
区的地址
#define ID_ADDR_ROM 0x0ff9 //STC104W 系列ID号的存放在ROM
区的地址

sfr AUXR=0x8E;
sbit RXB=P3^0; //定义串口 TX/RX 引脚
sbit TXB=P3^1;

typedef bit BOOL;
typedef unsigned char BYTE;
typedef unsigned int WORD;

BYTE TBUF, RBUF;
BYTE TDAT, RDAT;
BYTE TCNT, RCNT;
BYTE TBIT, RBIT;
BOOL TING, RING;
BOOL TEND, REND;

void UART_INIT();
void UART_SEND(BYTE dat);

BYTE t, r;
BYTE buf[16];

void main()
{
    BYTE idata *iptr;
    BYTE code *cptr;
    BYTE i;

```

```

TMOD=0x00;           //定时器 0 在 16 位自动重装模式
AUXR=0x80;          //定时器 0 在 1T 模式
TLO=BAUD;
TH0=BAUD>>8;       //初始化定时器 0 并设定重装值
TR0=1;              //定时器 0 开始运行
ET0=1;              //使能定时器 0 中断
PT0=1;              //improve timer0 interrupt prio
rity
EA=1;                //打开中断开关

UART_INIT();

iptr=ID_ADDR_RAM;   //从 RAM 区读取 ID 号
for (i=0; i<7; i++) //读 7 个字节
{
    UART_SEND(*iptr++); //发送 ID 到串口
}

cptr=ID_ADDR_ROM;   //从程序区读取 ID 号
for (i=0; i<7; i++) //读 7 个字节
{
    UART_SEND(*cptr++); //发送 ID 到串口
}

while (1);          //程序终止
}

//-----
//Timer interrupt routine for UART

void tm0() interrupt 1 using 1
{
    if (RING)
    {
        if (--RCNT == 0)
        {
            RCNT=3; //reset send baudrate counter
            if (--RBIT == 0)
            {
                RBUF=RDAT; //保存数据到 RBUF
                RING=0; //停止接收
                REND=1; //设置接收完成标志
            }
        }
    }
}

```

```

        else
        {
            RDAT >>= 1;
            if (RXB) RDAT |= 0x80; //移位 RX 数据到 RX 缓冲
        }
    }
}
else if (!RXB)
{
    RING=1; //设置开始接收标志
    RCNT=4; //initial receive baudrate coun
    ter
    RBIT=9; //初始化接收位数(8 数据位+1 停止
    位)
}

if (--TCNT == 0)
{
    TCNT=3; //reset send baudrate counter
    if (TING) //judge whether sending
    {
        if (TBIT == 0)
        {
            TXB=0; //发送开始位
            TDAT=TBUF; //从 TBUF 加载数据到 TDAT
            TBIT=9; //初始化发送位数(8 数据位+1 停止
            位)
        }
        else
        {
            TDAT >>= 1; //移位数据到 CY
            if (--TBIT == 0)
            {
                TXB=1;
                TING=0; //停止发送
                TEND=1; //设置发送完成标志
            }
            else
            {
                TXB=CX; //写 CY 到 TX 脚
            }
        }
    }
}
}

```

```
}

//-----
//initial UART module variable
void UART_INIT()
{
    TING=0;
    RING=0;
    TEND=1;
    REND=0;
    TCNT=0;
    RCNT=0;
}

//-----
//发送串口数据
void UART_SEND(BYTE dat)
{
    while (!TEND);
    TEND=0;
    TBUF=dat;
    TING=1;
}



---


```