# USMART V2.4 使用说明

USMART 是由 ALIENTEK 开发的一个灵巧的串口调试互交组件,通过它你可以通过串口 助手调用程序里面的任何函数,并执行。因此,你可以随意更改函数的输入参数(支持数字 (10/16 进制)、字符串、函数入口地址等作为参数), 单个函数最多支持 10 个输入参数, 并 支持函数返回值显示。V2.1 新增 hex 和 dec 两个指令。他们可以用于设置函数参数的显示 格式。也可以用于数据的进制转换。

USMART 的特点如下:

- 1, 可以调用绝大部分用户直接编写的函数。
- 2, 资源占用极少(最少情况: FLASH:2.5K; SRAM:72B)。
- 3, 支持参数类型多(数字(包含10/16进制)、字符串、函数指针等)。
- 4, 支持函数返回值显示。
- 5, 支持参数及返回值格式设置。
- 6, 使用方便。

有了 USMART,你可以轻易的修改函数参数、查看函数运行结果,从而快速解决问题。 比如你调试一个摄像头模块,需要修改其中的几个参数来得到最佳的效果,普通的做法:写 函数->修改参数->下载->看结果->不满意->修改参数->下载->看结果->不满意....不停的 循环,直到满意为止。这样做很麻烦不说,单片机也是有寿命的啊,老这样不停的刷,很折 寿的。而利用 USMART,则只需要在串口调试助手里面输入函数及参数,然后直接串口发送 给单片机,就执行了一次参数调整,不满意的话,你在串口调试助手修改参数在发送就可以 了,直到你满意为止。这样,修改参数十分方便,不需要编译、不需要下载、不会让单片机 折寿。

USMART 支持的参数类型基本满足任何调试了,支持的类型有: 10 或者 16 进制数字、 字符串指针(如果该参数是用作参数返回的话,可能会有问题!)、函数指针等。因此绝大部 分函数,可以直接被 USMART 调用,对于不能直接调用的,你只需要重写一个函数,把影 响调用的参数去掉即可,这个重写后的函数,即可以被 USMART 调用了。

经过以上简单介绍,我们对 USMART 有了个大概了解,接下来我们来看看 USMART 的移植和使用。

関索 🛛 | | | | | | | | | | | | | | ▼

# 一、USMART 移植

USMART 组件总共包含 6 文件如下图所示:

Settin	gs\liuj\⊈	皂面\ALIENTEK MINISTM32	实验10	TFTLCD显示实验+USMART	USMART	🖌 🔁 转到
×	**** 	readme.txt 文本文档 1 KB	C	<b>usmart.c</b> C Source File 8 KB	нÌ	<b>usmart.h</b> C/C++ Header File 3 KB
*	C	<b>usmart_config.c</b> C Source File 3 KB	C	<b>usmart_str.c</b> C Source File 8 KB	нÌ	<b>usmart_str.h</b> C/C++ Header File 2 KB
			图 1	1 USMART 组件		

其中 redeme.txt 是一个说明文件,不参与编译。其他五个文件, usmart.c 负责与外部 互交等。usmat str 主要负责命令和参数解析。usmart config.c 主要由用户添加需要由 usmart 管理的函数。

usmart.h 和 usmart\_str.h 是两个头文件,其中 usmart.h 里面含有几个用户配置宏定义, 可以用来配置 usmart 的功能及总参数长度(直接和 SRAM 占用挂钩)。

USMART 的移植,只需要实现两个函数。两个函数都在 usmart.c 里面,第一个是 void usmart init(void)函数,该函数主要实现串口初始化,如果用中断执行 usmart 的扫描,则可 以把中断的初始化代码,也放到这个函数里面。

在 ALIENTEK STM32 开发板上该函数的实现代码如下:

#### //初始化串口控制器

#### void usmart\_init(void)

{

//必须使能串口中断接收

//串口1初始化 uart\_init(72,9600);

#if USMART\_ENTIM2\_SCAN==1

Timer2\_Init(1000,7199); //7200 分频,时钟为 10K ,100ms 中断一次

```
#endif
```

usmart\_dev.sptype=1; //十六进制显示参数

}

上述代码,我们初始化串口波特率为9600,根据USMART\_ENTIM2\_SCAN的值设定是 否开启定时器 2,如果开启,将会每隔 100ms 执行一次 usmart 的扫描。

这里 uart init 设置还包括开启串口中断接收,并实现利用回车键判别接收是否完成, 具体的说明,请参考《STM32 不完全手册 V2.0》2.7.3 节或《例说 STM32》5.3 节。不过我 们这里提供的源码 usart.c 文件已经被更新,为 1.4 版本了。主要修改了 USART\_RX\_STA 这 个自定义寄存器,将其由 u8 变为 u16,这样我们的串口一次可以接收的字节数,最大可以 是2的14次方,主要为了增加对长函数名及函数参数的支持。

Timer2 的中断函数如下:

//定时器2中断服务程序 void TIM2\_IRQHandler(void)

{

```
if(TIM2->SR&0X0001)//溢出中断
```

```
{
usmart_dev.scan();//执行 usmart 扫描
}
TIM2->SR&=~(1<<0);//清除中断标志位
```

}

这里主要就是调用了 usmart\_dev.scan();这个函数,用于及时处理从串口接收到的数据,即上文所说的 usmart 扫描。这里采用的是中断扫描方式(推荐),你也可以使用死循环里面 调用的方式(USMART\_ENTIM2\_SCAN 为 0),只要保证 usmart\_dev.scan();函数每隔一定时间(建议不超过 200ms)被调用一次即可。

第二个要实现的函数就是 void usmart\_scan(void)。该函数用于执行 usmart 扫描,该函数需要得到两个参量,第一个是从串口接收到的数组(USART\_RX\_BUF),第二个是串口接收状态(USART\_RX\_STA)。接收状态包括接收到的数组大小,以及接收是否完成。

该函数的执行过程: 先判断串口接收是否完成(USART\_RX\_STA的最高位是否为1),如果完成,则取得串口接收到的数据长度(USART\_RX\_STA的低14位),并在末尾增加结束符,再执行解析,解析完之后清空接收标记(USART\_RX\_STA 置零)。如果没执行完成,则直接跳过,不进行任何处理。

在 ALIENTEK STM32 开发板上该函数实现代码如下:

//usmart 扫描函数

//通过调用该函数,实现 usmart 的各个控制.该函数需要每隔一定时间被调用一次 //以及时执行从串口发过来的各个函数.

//本函数可以在中断里面调用,从而实现自动管理.

```
//如果非 ALIENTEK 用户,则 USART_RX_STA 和 USART_RX_BUF[]需要用户自己实现 void usmart_scan(void)
```

{

```
u8 sta,len;
```

if(USART\_RX\_STA&0x8000)//串口接收完成?

{

```
len=USART_RX_STA&0x3fff;//得到此次接收到的数据长度
USART_RX_BUF[len]='\0'; //在末尾加入结束符.
sta=usmart_dev.cmd_rec(USART_RX_BUF);//得到函数各个信息
if(sta==0)usmart_dev.exe();//执行函数
else
{
    len=usmart_sys_cmd_exe(USART_RX_BUF);
    if(len!=USMART_FUNCERR)sta=len;
    if(sta)
    {
        switch(sta)
        {
            case USMART_FUNCERR:
            printf("函数错误!\r\n");
            break;
```

```
case USMART_PARMERR:
                     printf("参数错误!\r\n");
                     break;
                 case USMART_PARMOVER:
                     printf("参数太多!\r\n");
                     break;
                 case USMART_NOFUNCFIND:
                     printf("未找到匹配的函数!\r\n");
                     break;
             }
          }
       }
       USART_RX_STA=0;//状态寄存器清空
   }
}
   完成这两个函数的移植,你就可以使用 USMART 了。
```

# 二、USMART 使用

USMART 的使用很简单,下面结合 ALIENTEK MINISTM32 实验 10 TFTLCD 显示实验为 例介绍一下 USMART 的使用(移植好的例程为: ALIENTEK MINISTM32 扩展实验 9 USMART 应用)。首先打开实验 10 的工程,然后将 usart.h 的 EN\_USART1\_RX 和设置为 1,使能串口 中断接收。如下图所示(注意,这里我们用的是 V1.4 版本的串口驱动代码,如果你的还是 旧版本的,请将 usmart 实验源码的 usart 文件夹覆盖你的 usart 文件夹):



图 2.1 修改 EN\_USART1\_RX 为 1

然后,我们复制 USMART 文件夹到工程文件夹下面,如下图所示:

🗁 C:\Documents and Settin	gs\liuj\桌面\ALIENTE	X MINISTM32 实验10 TFTLC	D显示实验	_ 🗆 🔀
文件(E) 编辑(E) 查看(V) 收藏	(A) 工具(T) 帮助(H)			
🔇 后退 🝷 🕥 🔹 🏂 🔎 搜索 🚺	ゔ 文件夹 🛛 🔛 🗸			
地址(D) 🗀 C:\Documents and Settin	ags/liuj/桌面/ALIENTEK MINI	STM32 实验10 TFTLCD显示实验		🖌 芛 转到
文件和文件夹任务 ×	HARDWARE	SYSTEM	USER USER	
其它位置 *	README. TXT			
<ul> <li>② 桌面</li> <li>● 我的文档</li> <li>● 我的电脑</li> <li>● 网上邻居</li> </ul>		USMART		
详细信息				
<b>USMART</b> 文件夹 修改日期: 2011年6月14日, 15:28				
选定 1 个对象			🛛 😼 我的电脑	.::

#### 图 2.2 复制 USMART 文件夹到工程文件夹下

接着,我们在工程里面添加 USMART 组件代码,并把 USMART 文件夹添加到头文件包含路径,在主函数里面加入 include "usmart.h"如下图所示:



图 2.3 添加 USMART 组件代码

由于 USMART 提供了 STM32 的 TIM2 初始化设置代码,我们只需要在 usmart.h 里面设置 USMART\_ENTIM2\_SCAN 为 1,即可完成 TIM2 的设置。此部分代码如下:

```
#if USMART_ENTIM2_SCAN==1
//下面这两个函数,非 USMART 函数,放到这里,仅仅方便移植.
//定时器 2 中断服务程序
void TIM2_IRQHandler(void)
{
   if(TIM2->SR&0X0001)//溢出中断
   {
      usmart_dev.scan();//执行 usmart 扫描
   }
   TIM2->SR&=~(1<<0);//清除中断标志位
}
//使能定时器 2,使能中断.
void Timer2_Init(u16 arr,u16 psc)
{
   RCC->APB1ENR|=1<<0;//TIM2 时钟使能
   TIM2->ARR=arr; //设定计数器自动重装值
   TIM2->PSC=psc; //预分频器 7200,得到 10Khz 的计数时钟
   //这两个东东要同时设置才可以使用中断
   TIM2->DIER|=1<<0; //允许更新中断
   TIM2->DIER|=1<<6; //允许触发中断
   TIM2->CR1|=0x01; //使能定时器 2
```

```
MY_NVIC_Init(3,3,TIM2_IRQChannel,2);//抢占 3,子优先级 3,组 2(组 2 中优先级最低
的)
```

}

### #endif

此时,我们就可以使用 USMART 了,不过在主程序里面还得执行 usmart 的初始化,另 外还需要针对你自己想要被 USMART 调用的函数在 usmart\_config.c 里面进行添加。下面先 介绍如何添加自己想要被 USMART 调用的函数,打开 usmart\_config.c,如下图所示:



#### 图 2.4 添加需要被 USMART 调用的函数

这里的添加函数很简单,只要把函数所在头文件添加进来,并把函数名按上图所示的方式增加即可,默认我们添加了两个函数: delay\_ms 和 delay\_us。这里我们根据自己的需要按上图的格式添加其他函数,添加完之后如下图所示:



#### 图 2.5 添加需要被 USMART 调用的函数

上图中,我们添加了 lcd.h,并添加了很多 LCD 函数,最后我们还添加了 led\_set 和 test\_fun 两个函数,这两个函数在 test.c 里面实现,代码如下: //LED 状态设置函数

void led\_set(u8 sta)

```
{
```

}

LED1=sta;

ledset(sta);

//函数参数调用测试函数

void test\_fun(void(\*ledset)(u8),u8 sta)

```
{
```

}

led\_set 函数,用于设置 LED1 的状态,而第二个函数 test\_fun 则是测试 USMART 对函数参数的支持的,test\_fun 的第一个参数是函数,在 USMART 里面也是可以被调用的。

在添加完函数之后,我们修改主函数,如下:

//Mini STM32 开发板扩展实验

//USMART 测试 实验

//正点原子@ALIENTEK

//技术论坛:www.openedv.com

int main(void)

{

```
Stm32_Clock_Init(9);//系统时钟设置
delay_init(72);
                    //延时初始化
uart_init(72,9600); //串口1初始化
LED_Init();
LCD_Init();
usmart_dev.init(); //初始化 USMART
POINT_COLOR=RED;
LCD_ShowString(30,50,"Mini STM32 ^_^");
LCD_ShowString(30,70,"USMART TEST");
LCD_ShowString(30,90,"ATOM@ALIENTEK");
LCD_ShowString(30,110,"2011/6/18");
while(1)
{
    LED0=!LED0;
    delay_ms(500);
}
```

}

编译之后,我们下下载代码到 ALIENTEK MiniSTM32 开发板上,就可以通过串口来调用 我们在图 2.5 中所添加的函数了。下面简单介绍一下,下载完代码,我们可以看到 DS0 不停 闪烁,然后屏幕上显示了一些字符(就是主函数里面要显示的字符)。

我们打开串口调试助手(由于我写的助手在 WIN7 上有兼容问题,这里使用丁丁的助手),选择正确的串口号,并选择发送新行(即发送回车键)选项。如下图所示(点击扩展->隐藏):

12 SSCOM 3.3	
·	多条字符串定义 串口资料
	l list 1
	1 <u>1d</u> 2
	3
	help 4
	hex 5
	dec 6
	7
	8
	9
	10
	<u> </u>
	12
	Г <u>13</u>
	L 14
	15
	16
	□ 自动循环发送, 间隔: 1000 ms
打开文件 文件名 发送文件 停止发送 串口号 COM10 ▼ ④ 关闭串口 帮助 保存窗口 清除窗口	
波特率 9600 ▼ 「定时发送 10 ms//↓ ▽ 发送新行 数据位 8 ▼ 字符串输入框: 发送 → 地区发送 大 虾 停止位 1 ▼ ?	电子网
www.daxia.com S:0 R:0 COM10已打开 9600bps 8 1	CTS=1DSR=1RLSD=1

图 2.6 启动串口助手

此时我们在右边的栏里面输入各种指令或者函数,就可以得到执行结果如下图:

12 SSCOM 3.3				
□ USMART是由ALIENTEK开发的一个灵巧的串口调试互交组件,通过 它,你可以通过串口助手调用程序里面的任何函数,并执行.因此,你可 以随意更改函数的输入参数(支持数字(10/16进制)、字符串,函数入 口地址等代为参数)、单个函数最多支持10个输入参数,并支持函数返 回信显示.新增参数显示进制设置功能,新增进制转换功能. 技术支持:**** openedv.com USMART有6个系统命令: ?: 获取帮助信息 help:获取帮助信息 help:获取帮助信息 list:可用的函数列表 id: 可用函数的ID列表 hex: 参数16进制显示,后跟空格+数字即执行进制转换 dec: 参数10进制显示,后跟空格+数字即执行进制转换 请按照程序编写格式输入函数名及参数并以回车键结束. ————————————————————————————————————	多条字符串定义     串口资料       HEX     字符串     发送       list     1       id     2       ?     3       help     4       hex     5       dec     6       7     8       9     10       11     12       13     14       15     15			
打开文件     文件名     发送文件     停止发送       串ロ号     COM10     ●     关闭串口     帮助     保存窗口     諸除窗口       波特率     9600     ▼     定时发送     10     ms/次      欠送新行     大虾       数据位     8     ▼     字符串输入框:     发送     上四X发送     大虾	□ 13 16 16 16 16 16 16 16 ms 18 ms 1000 ms 1000 ms 1000 ms 1000 ms 1000 ms 1000 ms 1000 ms 1000 ms 1000 ms			
停止位 1 ▼ ? www.daxia.com S:3 R:716 COM10已打开 9600bps 8 1	CTS=1 DSR=1 RLSD=1			

图 2.7 输入"? /help" 指令

L SSCOM 3.3		×
list: 可用的函数列表	多条字符串定义 串口资料	
id: 可用函数的ID列表	HEX 字符串	发送
hex · 参数16进制显示 后跟空格+数字即执行进制转换	☐ list	$\square$
dec: 参数10进制显示,后跟空格+数字即执行进制转换	[ id	2
请按照程序编写格式输入函数名及参数并以回车键结束.		3
ALIENTEK	her	- *
函数清单	L dec	6
void delay_ms(u16 nms)		7
void delay_us(u32 nus) void LCD Clear(u16 Color)	Г	8
void LCD_Fill (u16 xsta, u16 ysta, u16 xend, u16 yend, u16 color)		9
void LCD_DrawLine(u16 x1, u16 y1, u16 x2, u16 y2)	Г	10
void LLD_Drawnectangie(ulo x1, ulo y1, ulo x2, ulo y2) void Draw Circle(ulo x0.ulo v0.u8 r)		11
void LCD_ShowNum (u16 x, u16 y, u32 num, u8 len, u8 size)		12
void LCD_ShowString(u16 x, u16 y, const u8 *p)		13
void ied_set(uo sta) void test_fun(void(*ledset)(u8),u8 sta)	Г	14
void LCD_Init(void)	Г	15
ul6 LCU_BGK2KGB(ul6 c) ul6 LCD ReadPoint(ul6 x ul6 x)		16
	□ 自动循环发送, 间隔: 1000	ms
打开文件 文件名 发送文件 停止发送	隐藏 IT RTS	
串ロ号 COM10 ▼ ● 	│	
波特率 9600 ▼ 「定时发送 10 ms/次 区 发送新行 大 吓	电子网	
数据位 8 ▼ 字符串输入框: 发送 「 HEX发送		
停止位 1 ▼ list		
www.daxia.com S:12 R:1305 COM10已打开 9600bps 8 1	CTS=1DSR=1RLSD=1	3

图 2.8 输入"list"指令

通过"1	ist"	指令,	我们可以获得当前	USMART	所管理的全部函数。
------	------	-----	----------	--------	-----------

SSCOM 3.3	
请按照程序编写格式输入函数名及参数并以回车键结束.	多条字符串定义 串口资料
ALIENTEK	HEX 字符串 发送
	list 1
void delay_ms(u16 nms)	
void delay_us(u32 nus) void LCD Clear(u16 Color)	3
void LCD_Fill (u16 xsta, u16 ysta, u16 xend, u16 yend, u16 color)	help 4
void LCD_DrawLine(u16 x1, u16 y1, u16 x2, u16 y2) void LCD_DrawBectangle(u16 x1, u16 x1, u16 x2, u16 y2)	
void Draw_Circle (u16 x0, u16 y0, u8 r)	del av. ms (1000)
void LCD_ShowNum (u16 x, u16 y, u32 num, u8 len, u8 size) void LCD_ShowString(u16 x, u16 x, const u8 %n)	8
void led_set (u8 sta)	9
void test_fun(void(*ledset)(u8),u8 sta) void LCD Toit(void)	10
u16 LCD_BGR2RGB (u16 c)	
u16 LCD_ReadPoint (u16 x, u16 y)	12
	13
delay_ms(OX3E8)=OX3E8	
10进制参数显示!	
delay_ms(1000)=1000;	□ 目动循环发送,间隔: 1000 ms
打开文件 文件名 发送文件 停止发送	Ž 隐藏 FTS
串ロ号 COM10 ▼ 🛞 美闭串ロ 🛛 帮助 🔤 保存窗口 清除窗口	」 □ HEX显示 □ DTR
波特率 9600 🔻 🗆 定时发送 10 🛛 ms/次 🔽 发送新行 🚽 🖬	由子図
数据位 8 ▼ 字符串输入框: 发送   □ HEX发送 🔨 🏹	
停止位 1 🔽 delay_ms (1000)	
www.daxia.com S:49 R:1373 COM10已打开 9600bps 8 1	CTS=1DSR=1RLSD=1

通过输入 delay\_ms(1000)函数,观察板子上的 DS0,可以看到会停1秒钟之后再闪烁。

图 2.9 执行: delay\_ms(1000)函数

II SSCOM 3.3					_ 0	X
请按照程序编写格式输入函数名及参数并以回车键结束.	^	多	条字符串定义		L)	_
		HE	( 字	符串		发送
			list			1
void delay_us(u32 nus)			id			2
void LCD_Clear(u16 Color) void LCD Fill(u16 vote u16 vote u16 vend u16 vend u16 color)		닅	?			3
void LCD_DrawLine(u16 x1, u16 y1, u16 x2, u16 y2)		닅	help			4
void LCD_DrawRectangle(u16 x1, u16 y1, u16 x2, u16 y2)		H	hex			
void LCD_ShowNum (u16 x, u16 y, u32 num, u8 len, u8 size)		H	dec del en en (100)	יי		7
void LCD_ShowString(u16 x, u16 y, const u8 *p)		E	delay_ms(1000	5) 67740)		[ <u></u>
void led_set(uö sta) void test fun(void(*ledset)(u8),u8 sta)		Ē	deray_ds (0x1)	01140)		9
void LCD_Init (void)		Ē				10
u16 LCD_BGR2RGB(u16 c) u16 LCD_BeedPoint(u16 x u16 x)		Ē				11
dio Lob_neam orne (dio x, dio y)	≣	Г				12
J.J (0¥0¥0)-0¥0¥0		Г				13
10进制参数显示!		Γ				14
						15
delay_ms(1000)=1000·						16
del av us (1800000)	~	Γ	自动循环发送	, 间隔:	1000	ms
	支送		隐藏   [ F	RTS		
串口号 COM10 ▼ ④ 关闭串口 帮助 保存窗口 清除窗	ГП	Г	нах显示口	DTR		
	<b>t</b> ך	Ħ	已子网			
www.daxia.com S:69 R:1392 COM10已打开 9600bps 8 :	1	CTS	=1 DSR =1 RLSD	=1		

该函数没有返回值,所以输出的返回值 1000 对我们来说没有意义,忽略之,下同。

图 2.10 执行: delay\_us(0x1B7740)函数

这里,我们修改输入参数为16进制,调用 delay\_us 函数延时(注意 delay\_us 同样有延 时范围哦!) 0x1b7740=1800000us=1800ms=1.8s,可以看到 DS0 的闪烁会停的更久一些。 这里输入参数 0x1b7740, 其大小写是自动转换的, 你可以输入 0X1b7740 也可以输入 0x1B7740 都是一样的。

1 SSCOM 3.3	
woid delay_ms(u16 nms) void delay_us(u32 nus) void LCD_Clear (u16 Color) void LCD_Fill (u16 xsta, u16 ysta, u16 xend, u16 yend, u16 color) void LCD_DrawLine (u16 x1, u16 y1, u16 x2, u16 y2) void LCD_DrawLectungle (u16 x1, u16 y1, u16 x2, u16 y2) void LCD_DrawEcterale (u16 x0, u16 y, u32 num, u8 len, u8 size) void LCD_ShowString(u16 x, u16 y, const u8 *p) void LCD_ShowString(u16 x, u16 y, const u8 *p) void led_set(u8 sta) void led_set(u8 sta) void LCD_Init(void) u16 LCD_Rec2RGB(u16 c) u16 LCD_Rec2RGB(u16 c) u16 LCD_ReadFoint(u16 x, u16 y) delay_ms(0X3E8)=0X3E8 10进制参数显示!	多条字符串定义       串口资料         HEX       字符串       发送         1ist       1         id       2         ?       3         help       4         hex       5         dec       6         delay_ms(1000)       7         delay_us(0x1b7740)       8         LCD_Clear(0x7d7c)       10         11       12         13       14         15       14
delay_us(1800000)=1800000; LCD_Clear(32124)=512;	「     16       「     自动循环发送, 间隔: 1000 ms
打开文件     文件名     发送文件     停止发送       串口号     COM10 ▼     ●     关闭串口     帮助     保存窗口     清除窗口       波特率     9600 ▼     「定时发送 10 ms/次 ▼     发送新行     大 虹       数据位     8     ▼     字符串输入框:     发送     □     □     □       停止位     1     □     LCD_Clear (0x7 dTc)     □     □     □	 「 HEX显示 □ DTR 电子网

图 2.11 执行: LCD\_Clear(0X7d7C)函数

可以看到整个 LCD 颜色变为了浅蓝色。

1 SSCOM 3.3			×
void delay_us(u32 nus) void LCD_Clear(u16 Color) void LCD_Fill(u16 xsta,u16 ysta,u16 xend,u16 yend,u16 color)	^	多条字符串定义   串口资料   HEX 字符串	发送
void LUD_DrawLine(ul6 x1, ul6 y1, ul6 x2, ul6 y2) void LCD_DrawRectangle(ul6 x1, ul6 y1, ul6 x2, ul6 y2)		list	1
void Draw_Circle(u16 x0,u16 y0,u8 r) void LCD ShowNum(u16 x,u16 y,u32 num,u8 len,u8 size)			3
void LCD_ShowString(ul6 x, ul6 y, const u8 *p) void lad sat(u8 sta)		help	4
void test_fun(void(*ledset)(u8),u8 sta)		hex	5
ul6 LCD_BGR2RGB (ul6 c)		dec	7
u16 LCD_ReadPoint (u16 x, u16 y)		delay_ms(1000) delay_us(0x1b7740)	8
		LCD_Clear (0x7d7c)	9
delay_ms(UX328)=UX328 10进制参数显示!		Fill (10, 10, 230, 310, 0x07e0)	
			12
delay_ms (1000)=1000;	≣		13
del sy_us (1800000)=1800000;			14
ICD (100r (32124)=512.		<u> </u>	15
			16
LLD_F111(10, 10, 230, 310, 2016)=311;	~	自动循环发送,间隔: 1000	ms
打开文件 文件名 发送文件 停止发	过送	隐藏   F RTS	
串口号 COM10 ▼ 🛞 关闭串口 帮助 保存窗口 清除窗		HEX显示 □ DTR	
波特率 9600 ▼ 「定时发送 10 ms/次 ▼ 发送新行 数据位 8 ▼ 字符串输入框: 发送 「HEX发送 大日	吓	电子网	
停止位 1 【LCD_Fill (10, 10, 230, 310, 0x07e0)			
www.daxia.com S:120 R:1465 COM10已打开 9600bps 8 1	1 0	CTS=1DSR=1RLSD=1	//

图 2.12 执行: LCD\_Fill(10,10,230,310,0x07E0)函数

可以看到 LCD 内出现一个绿色的正方形。

L SSCOM 3.3	💌
void LCD_Fill (u16 xsta, u16 ysta, u16 xend, u16 yend, u16 color) void LCD_DrawLine (u16 x1, u16 y1, u16 x2, u16 y2) void LCD_DrawRectangle (u16 x1, u16 y1, u16 x2, u16 y2) void LCD_Star Circle (u16 x0, u16 y0, u8 r) void LCD_ShowNum (u16 x, u16 y, u32 num, u8 len, u8 size) void LCD_ShowString (u16 x, u16 y, const u8 *p) void LCD_ShowString (u16 x, u16 y, const u8 *p) void led_set (u8 sta) void led_set (u8 sta) void lcD_Init (void) u16 LCD_BGR2RGB (u16 c) u16 LCD_ReadPoint (u16 x, u16 y)	多条字符串定义         串口资料           HEX         字符串         发送           1ist         1           id         2           ?         3           help         4           hex         5           dec         6           delay ms(1000)         7
delay_ms(0X3E8)=0X3E8 10进制参数显示!	delay_us(0x1b7740)         8           LCD_Clear(0x7d7c)         9           _Fill(10, 10, 230, 310, 0x07e0)         10
delay_ms(1000)=1000;	LCD_DrawLine (10, 10, 230, 310)
delay_us(1800000)=1800000;	
LCD_Clear (32124)=512;	L 14
LCD Fill (10, 10, 230, 310, 2016)=311;	15
LCD_DrawLine (10, 10, 230, 310)=301;	「自动循环发送, 间隔: 1000 ms
打开文件     文件名     发送文件     停止发送       串口号     COM10 ▼     ●     关闭串口     帮助     保存窗口     清除窗口       波特率     3600 ▼     「定时发送     10     ms/次 ▽     发送新行     大虾       数据位     8     ▼     字符串输入框:     发送     」 班X发送     大虾	Rīā □ RTS □ □ मयX显示 □ DTR ■ <b>电子网</b>
停止位 1 ▼ LCD_DrawLine (10, 10, 230, 310)	
www.daxia.com S:149 R:1501 COM10已打开 9600bps 8 1	CTS=1DSR=1RLSD=1



可以看到绿色正方形内出现一条红色的对角线。

SSCOM 3.3	
void LCD_DrawRectangle(u16 x1, u16 y1, u16 x2, u16 y2) void Draw_Circle(u16 x0, u16 y0, u8 r) void LCD_ShowNum(u16 x, u16 y, u32 num, u8 len, u8 size) void LCD_ShowString(u16 x, u16 y, const u8 *p) void lcd_set(u8 sta) void led_set(u8 sta)	多条字符串定义     串口资料       HEX     字符串     发送       □     list     1       □     id     2
void LCD_Īnit (void) u16 LCD_BGR2RGB (u16 c) u16 LCD_ReadPoint (u16 x, u16 y)	?         3           help         4           hex         5           dec         6
del ay ms (0X3E8)=0X3E8 10进制参数显示!	delay_ms(1000)         7           delay_us(0x1b7740)         8           LCD_Clear(0x7d7c)         9
delay_ms(1000)=1000; delay_us(1800000)=1800000;	Fill (10, 10, 230, 310, 0x07e0)         10          LCD_DrawLine (10, 10, 230, 310)         11          rawRectangle (10, 10, 230, 310)         11
LCD_Clear(32124)=512:	13 14 15
LCD_DrawLine (10, 10, 230, 310)=301; LCD_DrawRectangle (10, 10, 230, 310)=301;	16 「自动循环发送,间隔:1000 ms
打开文件     文件名     发送文件     停止发送       串口号     COM10      ●     关闭串口     帮助     保存窗口     諸除窗口	院藏 I□ RTS □ HEX显示 □ DTR
波特率 9600 ▼ 定时发送 10 ms/次 ▼ 发送新行 数据位 8 ▼ 字符串输入框: 发送 □ HEX发送 大 虹 停止位 1 ▼ LCD_DrawRectangle(10, 10, 230, 310)	电子网
www.daxia.com S:183 R:1542 COM10已打开 9600bps 8 1	CTS=1DSR=1RLSD=1

图 2.14 执行: LCD\_DrawRectangle(10,10,230,310)函数

可以看到绿色正方形边框变为红色的了。

1 SSCOM 3.3		X
void LCD_DrawRectangle(u16 x1, u16 y1, u16 x2, u16 y2) 🔥 多条字符串5 woid Draw Circle(u16 x0 u16 x0 u8 x)	E义   串口资料	
void LCD_ShowNum (u16 x, u16 y, u32 num, u8 len, u8 size) HEX	字符串	发送
void LUL_ShowString(ul6 x, ul6 y, const u8 *p)		1
void test_fun(void(*ledset)(u8),u8 sta)		2
void LCD_Init(void)		3
u16 LCD_ReadPoint (u16 x, u16 y)		4
Lex Lex		6
10进制参数显示!	(1000.)	7
delay us	(1000) (0x1b7740)	8
del ay_ms (1000)=1000;	(0x7d7c)	9
Fill (10	10,230,310,0x07e0)	10
del ay us (1500000)=1800000; 未找到匹配的函数!	Line (10, 10, 230, 310)	11
☐ rawRecta	ngle(10,10,230,310)	12
LCD_Fill(10, 10, 230, 310, 2016)=311;	cle(120,160,55)	<u>(130</u>
LCD_DrawLine (10, 10, 230, 310)=301;		14
ICD DrewRecterele(10 10 230 310)=301		15
		16
Draw_Circle(120, 160, 55)=512;	发送,间隔:1000	ms
打开文件 文件名 发送文件 停止发送 隐藏	F RTS	
串口号 COM10 ▼ ⑧ 关闭串口 帮助 保存窗口 清除窗口 FHEX显示	DTR	
波特率 9600 ▼ 「定时发送 10 ms/次 ▽ 发送新行 + 市 中 子 6	<b>V</b>	
数据位 8 ▼ 字符串输入框: 发送 「 地区发送 ヘエト・ピ 」 ア	20	
停止位 1 🔽 Draw_Circle(120, 160, 55)		
www.daxia.com S:230 R:2134 COM10己打开 9600bps 8 1 CTS=1 DSR=1	RLSD=1	/

图 2.15 执行: Draw\_Circle(120,160,55)函数

可以看到绿色正方形内出现了一个红色的圆圈。

L SSCOM 3.3		
void LCD_ShowNum (u16 x, u16 y, u32 num, u8 len, u8 size) void LCD_ShowString (u16 x, u16 y, const u8 *p) void led_set (u8 sta)	多条字符串定义 串口资料 HEX 字符串 发送	
void test fun(void(*ledset)(u8), u8 sta) void LCD_Int(void) u16 LCD_REX268(u16 c)	□ list 1 □ id 2	
u16 LCD_ReadPoint (u18 x, u16 y)	? 3 help 4	
10进制参数显示!	hex 5 dec 6	
delay_ms(1000)=1000;	delay_ms(1000)         7           delay_us(0x1b7740)         8	
delay_us(1800000)=1800000; 未找到匹配的函数!	LCD_Clear(0x7d7c) 9 Fill(10,10,230,310,0x07e0) 10	
LCD_Fill (10, 10, 230, 310, 2016)=311;	LCD_DrawLine (10, 10, 230, 310) 11 rawRectangle (10, 10, 230, 310) 12	
LCD_DrawLine (10, 10, 230, 310)=301; LCD_DrawRectangle (10, 10, 230, 310)=301;	Draw_Circle (120, 160, 55)         13           Num (10, 10, 0x45abCdEf, 10, 16)         IA	
Draw_Circle(120, 160, 55)=512;	15	
LCD_ShowNum (10, 10, 1168887279, 10, 16)=10;	「自动循环发送,间隔: 1000 ms	
打开文件     文件名     发送文件     停止发送     隐藏     □ RTS       串口号     COM10     ●     美闭串口     帮助     保存窗口     清除窗口     □ HEX显示     DTR		
波特率     9600     ▼     定时发送     10     ms/次     又送送新行     大虾电子网       数据位     8     ▼     字符串输入框:     发送     □     HEX发送     大虾电子网       停止位     1     ▼     ICD_ShowNum (10, 10, 0x45abCdEf, 10, 16)     16)		
www.daxia.com S:267 R:2177 COM10已打开 9600bps 8 1	CTS=1DSR=1RLSD=1	

图 2.16 执行: LCD\_ShowNum(10,10,0x45abCdEf,10,16)函数

可以看到在 LCD 的绿色区域内显示了 1168887279 (与 0X45ABCDEF 相等)。

🚹 SSCOM 3.3		
void led_set(u8 sta) void test_fun(void(*ledset)(u8),u8 sta) void LCD_Init(void)	多条字符串定义   串口资料   HEX 字符串 发送	
ul6 LCD_BGR2RGB(ul6 c) ul6 LCD_ReadPoint(ul6 x, ul6 y)	list 1 id 2	
10进制参数显示!	help 4 hex 5	
delay_ms(1000)=1000;	dec 6	
del ay_us(1800000)=1800000; 未找到匹配的函数!	del ay_ms (1000) 7 del ay_us (0x1b7740) 8	
LCD_Fill(10, 10, 230, 310, 2016)=311;	LCD_Clear (0x7d7c) 9 Fill (10, 10, 230, 310, 0x07e0) 10	
LCD_DrawLine (10, 10, 230, 310)=301;	LCD_DrawLine (10, 10, 230, 310) 11	
LCD_DrawRectangle(10, 10, 230, 310)=301;	rawRectangle (10, 10, 230, 310)         12           Draw_Circle (120, 160, 55)         13	
Draw_Circle (120, 160, 55)=512;	Num (10, 10, 0x45abCdEf, 10, 16) 14	
LCD_ShowWum (10, 10, 1168887279, 10, 16)=10;	TM32 Dev Kit USMART Test!")	
LCD_ShowString(10,40, "ALIENTEK STM32 Dev Kit USMART Test!")=0; 🗸	「自动循环发送,间隔: 1000 ms	
打开文件 文件名 发送文件 停止发送 隐藏 F RTS		
串口号 COM10 ▼ ⑧ 关闭串口 帮助 保存窗口 清除窗口 厂 HEX显示 I DIR		
19 IIV I LLU_Showstring(ID, 40, ALLENIEK SIM32 Dev Kit USMA	CTC_1 DCD_1 DLCD_1	
www.daxia.com S:328 R:2243 COM10已打开 9600bps 8 1	CTS=1DSR=1RLSD=1	

图 2.17 执行: LCD\_ShowString(10,40,"ALIENTEK STM32 Dev Kit USMART Test!")函数 可以看到在 LCD 的屏幕对应位置显示了 ALIENTEK STM32 Dev Kit USMART Test!字样。

12 SSCOM 3.3			
void LCD_Init (void)	多条字符串定义 串口资料		
ul6 LCD_ReadPoint (ul6 x, ul6 y)	HEX 字符串 发送		
	List 1		
10进制参数显示!	id 2		
	? 3		
delay_ms(1000)=1000;	help 4		
del en us (1800000.)=1800000.	Log 6		
未找到匹配的函数!	del az mg (1000) 7		
ICD #:11 (10 10 000 010 0018)-011.	del av us (0x1b7740) 8		
LLF111 (10, 10, 230, 310, 2010)=311;	LCD Clear (0x7d7c) 9		
LCD_DrawLine (10, 10, 230, 310)=301;	Fill (10, 10, 230, 310, 0x07e0) 10		
LCD DrawRectangle(10, 10, 230, 310)=301;	LCD_DrawLine (10, 10, 230, 310) 11		
	🔽 rawRectangle (10, 10, 230, 310) 12		
Jraw_Circle(120, 160, 55)=512;	Draw_Circle (120, 160, 55) 13		
LCD_ShowNum (10, 10, 1168887279, 10, 16)=10;	Num (10, 10, 0x45abCdEf, 10, 16) 14		
ICD ShowString(10 40 "ATTENTER STW32 Dev Kit USWART Test!")=0.	TM32 Dev Kit USMART Test!") 15		
DES_SHOWS CITER (10, 40, ALLENTER SIMOL DEV RIC COMPANY TESC: )=0,	led_set (0)		
led_set (0)=0;	「自动循环发送,间隔: 1000 ms		
打开文件 文件名 发送文件 停止发送	隐藏 F RTS		
串口号 COM10 ▼ ● 关闭串口 帮助 保存窗口 清除窗口 FHEX显示 DTR			
波特率 9600 ▼ 「定时发送 10 ms/次 ▼ 发送新行 十 市 由 子 W			
数据位 8 ▼ 字符串输入框: 发送   □ HEX发送 - ヘエー	FE J MS		
停止位 1 ▼ led_set (0)			
MMMM davia.com 5:340 ₽:2260 COM10戸北∓ 9600bos 8.1	CTS=1 DSP=1 PI SD=1		

图 2.18 执行: led\_set(0)函数

可以看到 DS1 点亮了。

最后一个函数,其参数为函数指针,需要我们先获得函数的地址,这里通过 id 指令获 得,发送 id 指令,得到所有函数的 id,如下图:

12 SSCOM 3.3		
函数 ID函数 ID	多条字符串定义 串口资料	
delay_ms id is: nxn8nnn31F	HRX 字符串 发送	
delay_us id is:		
0X08000355	L id 2	
OXO8000E11	2 3	
LCD_Fill id is:	help 4	
OXO8001863 ICD Drowling id in:	hex 5	
OXO8001BCD	dec 6	
LCD_DrawRectangle id is:	[ delay ms(1000) 7	
UXU8UU1095 Drew Circle id is:	delay_us(0x1b7740) 8	
0X08001CD5	LCD_Clear (0x7d7c) 9	
LCD_ShowNum id is:	Fill(10, 10, 230, 310, 0x07e0) 10	
LCD ShowString id is:	LCD_DrawLine (10, 10, 230, 310) 11	
0X08001FE1	□ rawRectangle(10, 10, 230, 310) 12	
led_set id is:	□ Draw_Circle(120, 160, 55) 13	
test fun id is:	□ Num (10, 10, 0x45abCdEf, 10, 16) 14	
0X08000223	TM32 Dev Kit USMART Test!") 15	
LCD_Init id is: OVO8000F4B	□ led_set(0) 16	
LCD_BGR2RGB id is:	17	
OXOBOOCF1	□ 自动循环发送, 间隔; 1000 mg	
LLU_Keadfoint id is:		
打开文件 文件名 发送文件 停止发送	隐藏 I I RTS	
串口号 COM10 ▼ @ 关闭串口 帮助 保存窗口 清除窗口	□ HEX显示 □ DTR	
波特率     9600     ▼       数据位     8     ▼       停止位     1     ▼		
www.daxia.com S:344 R:2766 COM10已打开 9600bps 8 1	CTS=1DSR=1RLSD=1	

#### 图 2.19 执行"id"指令

由于 test\_fun 的函数指针我们设计的时候考虑的是 led\_set 函数,所以,我们从上面的 列表中选择 led\_set 的 ID,为 0x0800021d。再作为 test\_fun 的函数参数输入,发送给 ALIENTEK MiniSTM32开发板,得到如下结果:

SSCOM 3.3		
LCD_Clear id is: OXO8000E11 ICD_Fill_id_is:	^	多条字符串定义   串口资料   HRX 字符串 发送
0X08001B63 LCD DrawLine id is:		
LCD_DrawRectangle id is: 0X08001695		
Draw_Circle id is: 0X08801CD5 CCD_ShowNum id is:		hex 5
LCD_ShowString id is:		dec 0 delay_ms(1000) 7
OXO8001FE1 led_set id is: OXO800F21D		delay_us(0x1b7740)         0           LCD_Clear(0x7d7c)         9
test_fun id is: 0X08800223		LCD_DrawLine (10, 10, 230, 310, 0x07e0) 10 LCD_DrawLine (10, 10, 230, 310) 11
LLD_LINIT 1d 1s: 0X08000E4B LCD_B6R2RGB id is:		rawKectangle(10, 10, 230, 310)         12           Draw_Circle(120, 160, 55)         13
OXO8000CF1 LCD_ReadFoint id is: ovo5000000		Mum (10, 10, 0x45abCdEf, 10, 16) 14 TM32 Dev Kit USMART Test!") 15
		1ed_set(0)         16           test_fun(0x0800021d, 1)         III)
test_fun(134218269,1)=1;	~	厂 自动循环发送, 间隔: 1000 ms
打开文件     文件名     送送文件     停止发送     隐藏     IF RTS       串口号     COM10     ▼     ●     关闭串口     帮助     保存窗口     清除窗口     IF HEX显示     DTR		
波特率     9600     ▼     定时发送     10     ms/次     发送新行       数据位     8     ▼       停止位     1     ▼         test_fun (0x0800021 d, 1)		
www.daxia.com S:420 R:3356 COM10 己打开 9600bps 8 1	C	TS=1 DSR=1 RLSD=1

图 2.20 执行: test\_fun(0x0800021d,1)函数

此时可以看到 DS1 灯灭了。说明我们确实通过 test\_fun 调用了 led\_set,并设置了 LED1=1; 这里注意函数参数 ID 一定不能错,如果错了,可能导致程序崩溃!

最后,我们介绍一下 hex 和 dec 这两个系统命令的用法,当使用 hex 和 dec 不带参数 时,用于设置函数的参数及返回值的显示格式。比如发送 hex, 然后再发送 delay\_ms(1000), 我们可以看到如下结果:

12 SSCOM 3.3			
LCD_DrawLine id is: OXO8001BCD	多条字符串定义   串口资料	5)¥	
LCD_DrawRectangle id is:		心达	
Draw Circle id is:	list	1	
0X08001CD5	id	2	
LCD_ShowNum id is:	<b></b> ?	3	
UXUSUULEBF	☐ help	4	
0X08001FE1	hex	5	
led_set id is:	☐ dec	6	
0X0800021D	□ delay_ms(1000) (	1	
test_fun 1d 1s: 0X08000223	☐ delay us(0x1b7740)	8	
LCD Init id is:	LCD Clear (0x7d7c)	9	
OX08000E4B	Fill (10, 10, 230, 310, 0x07e0)	10	
LCD_BGRZRGB id is:	[ ICD DrawLine(10, 10, 230, 310)	11	
LCD ReadPoint id is:	[ rewBectergle(10, 10, 230, 310)	12	
OX08000D0B	Drew Circle (120, 160, 55)	13	
	Num (10, 10, 0x45abCdRf, 10, 16)	14	
tart fun (134218260 1)=1.	TH22 Der Vit 100000 Test ()	15	
cest_1ut(194210200,1)=1;	Timbe bev kit osmaki festi	16	
16进制参数显示!		17	
	test_fun (0x0800021d, 1)	-	
delay_ms (OX3E8)=OX3E8 🛛 🗸 🗸	□ 「 自动循环发送, 间隔: 1000 ,	ns	
打开文件 文件名 发送文件 停止发送	 隐藏 厂 RTS		
串口号 COM10 ▼ ● 关闭串口 帮助 保存窗口 清除窗口 下 HEX显示 DTR			
被特率     9600     ▼       数据位     8     ▼       容符串输入框:     发送     1000       delay_ms(1000)			
www.daxia.com S:487 R:4539 COM10已打开 9600bps 8 1 0	CTS=1DSR=1RLSD=1	1	

### 图 2.21 hex 方式显示参数

可以看到参数及返回值的显示都变为 16 进制显示了,如果再输入 dec,则可以让显示 格式变为10进制,如下图所示:

1 SSCOM 3.3			x
LCD DrawLine id is:	^	多条字符串定义 串口资料	
UXU8UUIBCD LCD DrawRectangle id is:		HEX 字符串 :	发送
0X08001C95		☐ list	1
Draw_Circle id is: nYn8nnich5		T id	2
LCD_ShowNum id is:		?	3
OXOSOO1EBF		help	4
LCD_ShowString id is: nVn8nn1FF1		hex	5
led_set id is:		dec	6
0X0800021D		delay ms(1000)	C10
test_fun 1d 1s:  0X08000223		delay_us(0x1b7740)	8
LCD_Init id is:		LCD_Clear (0x7d7c)	9
OXO8000E4B		Fill (10, 10, 230, 310, 0x07e0)	10
LLD_BGKZKGB 1d 1s: OXO8000CF1		LCD_DrawLine (10, 10, 230, 310)	11
LCD_ReadPoint id is:		<b>rawRectangle</b> (10, 10, 230, 310)	12
ОХО8000ДОВ		Draw_Circle (120, 160, 55)	13
		Num (10, 10, 0x45abCdEf, 10, 16)	14
delay_ms(OX3E8)=OX3E8		TM32 Dev Kit USMART Test!")	15
10进制参数显示!		1ed_set(0)	16
	≡	└── test_fun (0x0800021d, 1)	17
delay_ms(1000)=1000;		□ 自动循环发送, 间隔: 1000	ms
打开文件    文件名 发送文件   停止发	デぼ		_
串口号  COM10 ▼ ) ● _关闭串口帮助保存窗口清除窗口 HEX显示 UTA			
波特率 9600 ▼ □ 定时发送 10 ms/次 ☑ 发送新行 大虾电子网     紫符串输入框: 发送 □ HEX发送 Ⅰ ▼			
停止位 1 🔽 del ay_ms (1000)			
www.daxia.com S:553 R:5112 COM10己打开 9600bps 8 1	L	CTS=1DSR=1RLSD=1	/

2.22 dec 方式显示参数

如果在 dec 和 hex 后加空格再带参数的话, 就可以执行进制转换。比如分别输入: "hex 1234"和"dec 1234"就可以得到 HEX:0X4D2 和 DEC:4660,如下图所示:

SSCOM 3.3		
LCD ShowNum id is: OXO8001EBF	多条字符串定义 串口资料	
LCD_ShowString id is:	HEX 字符串 发送	
0X08001FE1	□ list 1	
1ed_set 1d 1s: 0X08000210	□ id 2	
test fun id is:	<b></b> ? <b>3</b>	
0X08000223	help 4	
LCD_Init id is: OVOSOODE4B	hex 1234 5	
LCD BGR2RGB id is:	dec 0x1234 6	
OXO8000CF1	delay ms (1000) 7	
LCD_ReadPoint id is:	del av us (0x1b7740) 8	
040000000	$\begin{bmatrix} ICD Clear (0x7d7c) & 9 \end{bmatrix}$	
	$\begin{bmatrix} F_{111}(10, 10, 230, 310, 0 \times 07 \bullet 0) \end{bmatrix} \begin{bmatrix} 10 \\ 10 \end{bmatrix}$	
delay_ms(DX3E8)=DX3E8 to进行任务局量。	[ ICD DrewLine (10, 10, 230, 310) 11	
10进制梦致显示!	F memBostonglo (10, 10, 230, 310) 12	
	Draw Circle (120, 160, 55) 13	
del ay_ms (1000)=1000;	Www (10, 10, 0x45abC4W6, 10, 16) 14	
HEY - 02402	TH22 Derr Kit 100 ATT Tort1") 15	
12.4.04482	lad set (0)	
222 2020	Leal Sec (0) 10 17	
DEC:4660	test_run (0x0000021d, 1)1	
×	「 自动循环发送, 间隔: 1000 ms	
打开文件 文件名 发送文件 停止发送	隐藏 「 RTS	
串口号 COM10 ▼ ④ 关闭串口 帮助 保存窗口 清除窗口	「HEX显示「DTR	
波特率     9600     ▼     定时发送     10     ms/次     发送新行       数据位     8     ▼     字符串输入框:     发送     □     HEX发送       停止位     1     ▼     dec 0x1234		
www.daxia.com S:577 R:5141 COM10己打开 9600bps 8 1	CTS=1DSR=1RLSD=1	

2.23 hex 和 dec 的进制转换

通过 dec 和 hex 两个指令,就可以方便快速的进行进制转换,调试起来更加方便。

至此整个 USMART 的使用就介绍完了。通过以上实例,我们可以看出 USMART 的使用 时非常简单的,我相信如果你真的学会了 usmart 的使用,一定会给你的学习/工作带来很大 的方便。

### 三、USMART 注意事项

1, USMART 的函数在函数名与'('之间不要留空格,否则导致无法匹配。比如 delay\_ms(1000),不要弄成 delay\_ms (1000)。

2,在参数为函数指针的时候,其函数 ID 一定不要搞错,否则可能导致程序崩溃。

3, usmart.h 里面, PARM\_LEN 的值最少应该为 4, 他是用来设定保存函数参数的数组 大小的。PARM\_LEN 的值将直接影响到 USMART 组件的 SRAM 占用率。计算公式为: sram=PARM\_LEN+72-4。当 PARM\_LEN 设置为 4 的时候,组件只占用 72 个字节。但此时 的参数长度则限制在4个字节。PARM\_LEN 的值,大家自己根据需要修改。

#### TO BE ADDED

正点原子@ALIENTEK 2011-6-14

