

## 1、 程序框架介绍

我司提供了串口屏驱动代码和范例程序，目前支持的单片机平台有51、STM32，用户只需修改几个函数即可以移植到其它平台。

基于范例程序，用户只需要添加自己的功能代码（例如温湿采集、开关控制等），指令的解析和处理流程完全由驱动代码提供。范例程序结构如下图所示。

main - 主程序，硬件初始化，用户代码

cmd\_proces: 指令处理流程，串口屏通知响应函数

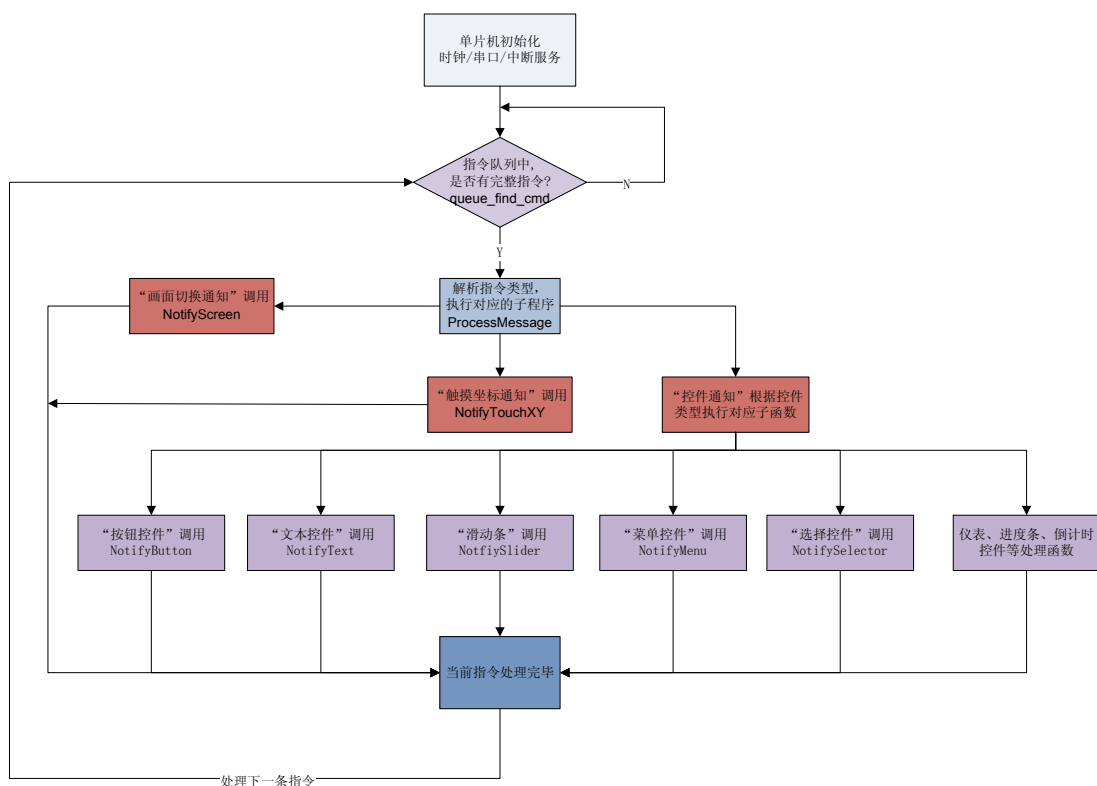
cmd\_queue: 串口屏指令队列，可从中获取指令

hmi\_driver: 串口命令驱动函数，例如设置控件的值

hmi\_user\_uart: 串口初始化、数据收发处理

## 2、 程序工作流程

程序工作流程图如下所示，无论用户当前使用什么MCU平台，只需替换串口底层的收发函数即可，程序处理流程不需要任何变化。



### 3、 程序配置

hmi\_dirver.h中包含几个宏定义，用户可根据需要调整

#define CRC16\_ENABLE 0 // CRC16校验功能，VisualTFT中需要相应设置，如下所示



#define CMD\_MAX\_SIZE 20 //单条指令大小，根据需要调整，尽量设置大一些

#define QUEUE\_MAX\_SIZE 80 //指令接收缓冲区大小，根据需要调整，尽量设置大一些

### 4、 如何移植到其它MCU平台

如果客户需要把工程移植到其他平台，建议复制所有源文件，然后根据需要修改下面几个地方：

a) hmi\_user\_uart.c -串口初始化及数据发送

```
void UartInit()    //初始化通信串口，设置波特率等
void SendChar(char ch) //通过串口发送一个字节
```

b) 串口数据接收中断处理

//8051平台下串口数据接收处理如下所示，其它平台可能需要修改

//接收的串口数据通过queue\_push添加的指令队列末尾

```
void serial() interrupt 4
{
    if(RI)//接收到窗口数据
    {
        RI= 0;
        queue_push(SBUF);//压入到指令缓冲区
    }
}
```