

文章编号:1007-2780(2015)06-1057-06

红外触摸屏响应分析及延时优化

郭 瑞*,朱沛立

(太原航空仪表有限公司,山西 太原 030006)

摘要:对机载多功能显示器红外触摸屏响应时间进行分析和计算,寻找缩短响应时间以便满足用户要求的方法。通过 VC++ 对响应时间进行了计算,根据触摸过程和计算结果,通过提高 RS-232 的波特率、中断实时接收处理触点数据和合并任务的方法对响应时间进行优化。优化后触摸响应的最大延时时间可缩短约 37.5 ms,优化后的延时约为 89.5 ms。保证了飞行员可以及时得到需要的画面,满足用户需求。

关 键 词:多功能显示器;红外触摸屏;响应时间;延时优化

中图分类号:TP391 **文献标识码:**A **doi:**10.3788/JYXS20153006.1057

Analysis and optimization of response time in infrared touch display

GUO Rui*, ZHU Pei-li

(Taiyuan Aero-Instruments Co., Ltd., Taiyuan 030006, China)

Abstract: The response time of infrared touch airborne display is analyzed and tested, and the optimization method of response time is introduced to meet the needs of users. The response time is counted in VC++. According to the touch process and the count result of response time, the response time can be reduced through increasing RS232 signal Baud rate, interrupt method and merging tasks. The optimized response time is 89.5 ms, which reduces 37.5 ms than before. The optimized response time ensures the aviator can see general picture in time.

Key words: multifunctional display; infrared touch display; response time; time-delay optimization

1 引 言

机载多功能显示器是显示与控制系统重要的人机接口设备,用于显示机上各种视频信息。随着航空工业的发展,机载多功能显示器人机交互不仅仅满足于传统的按键操作,触摸屏因其具有操作简单,使用灵活的特点在航空多功能显示器

中得到了越来越广泛的应用,红外触摸屏是其中的一种,它是在屏幕四边放置红外发射管和红外接收管,微处理器控制驱动电路依次接通红外发射管并检测相应的红外接收管,形成横竖交叉的红外光阵列,它无需薄膜,光透过率为 100%,而且不受电流、电压和静电干扰,适宜恶劣的环境条件,因此很适宜在航空机载设备中使用^[1-2]。在红外触摸屏的各项参数指标中,响应时间是其中的

主要指标之一,触摸画面能否及时响应,保证用户体验具有很重要的意义,该时间越短飞行员越可以及时得到需要的画面,因此如何进行响应时间分析、计算及优化是触摸屏需要研究的课题。

2 红外触摸响应分析

2.1 触摸响应流程分析

当红外触摸屏用于机载多功能显示器时,根据产品需求画面调用过程对其触摸响应流程进行分析,见图 1,从人员发出触屏操作到观测显示图形,多功能显示器需经过 10 个处理过程,分别为 P1 红外行列扫描、P2 触点处理并发送、P3 触点到主控模块的总线传输、P4 主控模块触点接收处理、P5 主控模块计算和设置画面参数、P6 触点到图形处理模块的总线传输、P7 图形处理模块对数据的接收处理、P8 图形处理、P9 视频信号传输和 P10 图形显示。其中,红外触摸屏负责完成过程 P1 和 P2,连续不断的周期执行,P1 和 P2 之间没有时间间隔;P3 为数据在总线上传输过程(本文用 RS-232 信号进行总线传输);P4 和 P5 为主控模块上运行的两个周期任务,任务周期均为 20 ms;P6 为数据从总线发送到图形处理模块的过程(本文通过 CPCI 总线进行数据传输);P7 和 P8 在图形处理模块上处理,两个环节循环执行,无时间间隔;P9 为视频传输过程(本文为 LVDS 信号);P10 为液晶显示屏刷新图像的过程。

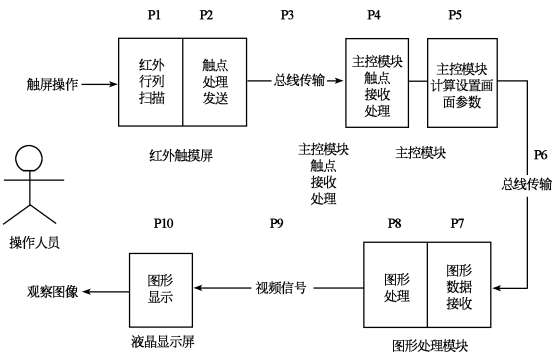


图 1 触摸响应流程

Fig. 1 Process diagram of touch display

2.2 各过程执行时间分析

P1 过程:红外触摸屏的 CPU 进行红外扫描,即将某个红外发射灯管打开,发射红外信号,检测对应的红外接收三极管状态,关闭该发射灯管,打

开下一个灯管,继续扫描。分别进行垂直方向、水平方向、斜向扫描。根据传统经验,P1 执行总时间约为 20 ms^[3],本文采用的触摸屏根据扫描时间和灯管数量的最优化处理,该部分执行时间为 10.5 ms。

P2 过程:红外触摸屏的 CPU 将 P1 过程的扫描结果进行计算分析,得到触点信息,并将该信息写入 RS-232 发送单元。该过程的执行时间很短,约为 1.5 ms。

P3 过程:根据红外触摸屏与主控模块之间的通讯协议规定,RS-232 的波特率为 38 400,1 个起始位,1 个停止位,8 个数据位,1 个奇偶校验位,有触点时数据包中共计 14 字节。因此总线传输时间为 $(14 \times 11 / 38\ 400) = 4\text{ ms}$ 。

P4 过程:主控模块查询接收当前 RS-232 通道的数据,并对数据包完整性进行判断,得到触点坐标。该过程执行时间通过计算得到。

P5 过程:主控模块通过触点坐标解码、设置相应画面的参数和数据。该过程执行时间通过计算得到。

P6 过程:将相应的画面参数进行格式封装,并通过 CPCI 总线数据传输,该过程执行时间通过计算得到。

P7 过程:图形处理模块查询接收控制发送的触点数据。该过程执行时间很短,小于 1 ms。

P8 过程:图形处理模块生成显示画面,并生成 LVDS 信号。根据图形处理模块硬件资源,该过程的执行时间约为 11 ms。

P9 过程:LVDS 信号传输过程。时间为 $(1 / 60)$ 约为 16.6 ms。

P10 过程:液晶显示屏显示 LVDS 信号。时间很短,可忽略。

从上述分析可见,红外触摸用于机载多功能显示器时响应时间的计算主要为主控模块对需求画面的解码、设置、封装过程的计算。

2.3 延时分析

红外触摸屏和图形处理模块的处理方式为循环处理,任务执行时间基本与周期相同,该过程执行延时分析见图 2。

如图 2 所示,任务 P1 和 P2 循环执行,执行周期(T)和执行时间(t)相同,需处理的事件在 t_1 时刻或 t_2 时刻发生时,均只有在下一周期才能得到处理,即在 T_2 时刻完成处理,因此某事件发生

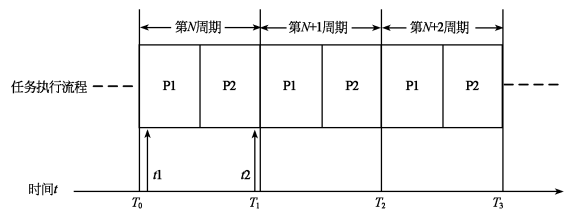


图 2 循环任务延时分析

Fig. 2 Analysis diagram of cycle task

到必定处理完成所需的最大时间 $T_{\max} = T + t = T \times 2$ 。

主控模块的两个过程 P4 和 P5 的处理方式为周期任务查询处理,任务执行时间小于执行周期,处理延时见图 3。

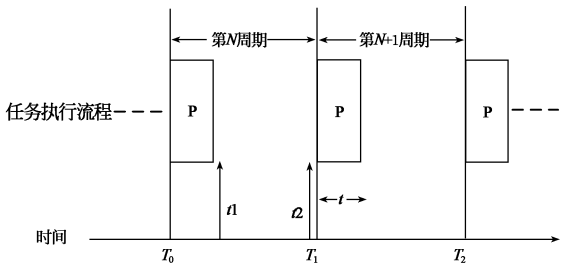


图 3 周期任务延时分析

Fig. 3 Analysis diagram of periodic task

如图 3 所示,任务执行的周期为 T ,任务执行时间为 $t, t < T$,事件在 t_1 时刻或 t_2 时刻发生时,均能在下一周期得到处理,即 $(T_1 + t)$ 时刻完成,因此事件发生到处理完成的最大时间 $T_{\max} = T + t$ 。

2.4 代码实现

根据上述的各过程时间分析和延时分析,采用 VC++ 对触摸响应过程的最大时间进行计算,如下:

```
int main(int argc, char * argv[])
{
    CreateThread ( NULL, 0x2000,
(LPTHREAD_START_ROUTINE) Task _
TouchScreen, 0, 0, NULL);
    CreateThread ( NULL, 0x2000,
(LPTHREAD_START_ROUTINE)task_main,
0, 0, NULL);
    CreateThread ( NULL, 0x2000,
(LPTHREAD_START_ROUTINE)task_GPM,
0, 0, NULL);
    return 0;
```

```
}
//周期任务
void Task_TouchScreen( void )
{
    while(Task_Serial232_Flag)
    {
        Receive_Touch_Msg();
        Sleep(20);
    }
}
//接收红外触摸屏传送的数据,得到触点坐标
void Receive_Touch_Msg(void)
{
    if((20 == RS232_ch2_Data.len) &&
(RS232_ch2_Data.buf[5] == 0x05))
    {
        str_LCD.dot_num = RS232_ch2_
Data.buf[6]; //触点个数
        str_LCD.dot_x.data.high = RS232_
ch2_Data.buf[9];
        str_LCD.dot_x.data.low = RS232_
ch2_Data.buf[10]; //触点 X 坐标
        str_LCD.dot_y.data.high = RS232_
ch2_Data.buf[11];
        str_LCD.dot_y.data.low = RS232_
ch2_Data.buf[12]; //触点 Y 坐标
        x1 = (unsigned int)(str_LCD.dot_
x.all * 0.1875);
        y1 = (unsigned int)(str_LCD.dot_
y.all * 0.1875);
        y1 = 768-y1;
        Infrared_Info.data.dot1_status =
RS232_ch2_Data.buf[8];
        Infrared_Info.data.dot1_x = x1;
        Infrared_Info.data.dot1_y = y1;
        str_LCD.dot_status = RS232_ch2_
Data.buf[8]; //触摸点状态
        if (touchstartflag == 0)
        {
            acTimeBaseGet(&m_touchstar-
ttime.time32.upper32, &m_touchstarttime.
time32.lower32);
            touchstartflag = 1;
```

```
        }
    }
}

//主控模块通过触点坐标解码、设置相应画面的
参数和数据
ACoreOs_task task_main( void )
{
    while(Task_MAIN_Flag)
    {
        Deal_Key_Touch();
    }
}

//将相应的画面参数进行格式封装,并进行数据
更新
ACoreOs_task task_GPM( void )
{
    Task_GPM_Flag = TRUE;
    while(Task_GPM_Flag)
    {
        semTake(Sem_GPM, WAIT_FOR-
EVER);
        if(KeyNum_transt)
        {
            key_GPM = KeyNum_transt;
            KeyNum_transt = 0;
        }
        Pic_Logi_c_Deal (key _ GPM, Pic-
Num);
        if(key_GPM)
        {
            picnum_temp = PicNum;
            PicNum = get_pic(key_GPM,Pic-
Num,1);
        }
        data_updata(key_GPM);
        key_GPM = 0;
    }
}

//画面数据更新
int data_updata(INT32 key)
{
    DCM_Command. data. Cur_PIC_ID = Pic-
```

```
Num/100;
    m_cooperate_pic_name  = DCM_Com-
mand,data,Cur_PIC_ID;
    m_cooperate_pic_shape = V_PIC_SHAPE_
NORMAL;
    switch (DCM_Command, data, Cur_PIC_
ID)
    {
        case PIC_EFIS_NOR_PAGE:
        {
            efis_data_set_normal();
            send_efis_data_to_GPM( Temp_ad-
dress,Temp_length);
            if (touchstartflag == 1)
            {
                acTimeBaseGet (&m_touchendtime,
time32, upper32, &m _ touchendtime, time32,
lower32);
                touchstartflag = 0;
                m _ PxTime = ( m _ touchendtime,
time32,lower32- m_touchstarttime,time32,low-
er32)/25000;
                m_TotalTime = m_P1Time + m_
P2Time + m_P3Time + m_PxTime + m_Py-
Time + m_P9Time;
            }
            break;
            default:
            break;
        }
    }
    return 0;
}
#endif
```

2.5 计算结果

通过计算,得到响应时间最大值为 127 ms。见图 4。

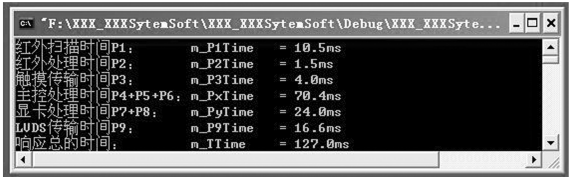


图 4 响应时间结果

Fig. 4 Count result of response time

3 优化设计方案

通过对目前多功能显示器触摸响应的计算结果进行分析,该结果不能满足指标要求的不大于 100 ms 的要求,可选择的优化方法分析见下。

3.1 减小传输过程延时

传输过程延时主要产生在 P3、P6、P9 过程, P6 过程延时时间很短,可忽略不计; P9 过程时间由液晶屏视频刷新率确定,不可变更;因此可通过提高波特率减小 P3 过程执行时间, RS-232 波特率最高为 115 200 bps,因此 P3 执行时间可减小为约 1.4 ms。

3.2 缩短周期任务执行时间

周期任务执行时间主要产生在 P4、P5 过程,缩短周期任务执行的时间可采用如下 3 种方法:

(a) 缩短任务执行周期

将主控模块 RS-232 查询接收和更新控制数据的任务周期缩短。该方法可以减少 P4、P5 的过程延时,但会影响主控模块上其他周期任务的执行,因此该方法不予采用。

(b) 合并任务,减少过程

将更新触点数据的功能移植到查询接收 RS-232 的任务中,即将 P5 过程与 P4 过程合并。该方法能够减小 P4、P5 过程的延时,缩短触摸响应时间 20 ms 且不会影响软件的功能和性能。

(c) 中断处理

采用中断方式处理 RS-232 通讯,在中断服务函数中处理 RS-232 数据同时将触点数据发送到图形处理模块。该方法可以缩短触摸响应时间 20 ms 且不会影响软件的功能和性能。

3.3 缩短循环任务执行时间

循环任务执行时间主要来源为红外触摸屏和

图形处理模块的 P1、P2、P7、P8 过程,通过减少红外扫描每个灯管的工作时间、减少灯管个数可以缩短红外触摸屏处理时间^[4-5],在满足红外触摸屏稳定可靠的基础上,该部分时间已经得到了优化。

若想缩短图形处理模块的处理时间则需更换硬件方案采用更高速度的图形处理模块和处理器,这会增加图形处理模块的功耗,这必然对产品的热设计带来压力,使产品高温工作性能和稳定性得不到保证,因此该方法也不予采用。

3.4 优化结果

综上所述,通过采用提高 RS232 波特率、中断实时接收处理触点数据和合并任务的方法可以缩短响应时间,优化后的计算结果见图 5。

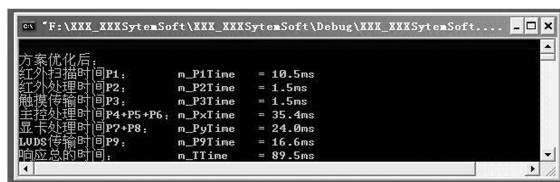


图 5 优化后响应时间结果

Fig. 5 Count result of optimized response time

4 结 论

机载多功能显示器红外触摸屏作为机上重要的人机接口设备,要求快速响应飞行员的触摸信息,文中对响应过程进行了分析,计算出了响应时间,为了提高响应速度,对任务处理进行了优化,通过提高 RS-232 的波特率、中断实时接收处理触点数据和合并任务的方法,触摸响应的最大延时时间可缩短约 37.5 ms,优化后的延时约为 89.5 ms,满足指标不大于 100 ms 的要求。

参 考 文 献:

- [1] 李钧. 一种新型红外多点触摸识别算法[J]. 计算机与现代化, 2012(9): 178-180, 189.
Li J. A new infrared multi-touch recognition algorithm [J]. *Computer and Modernization*, 2012(9): 178-180, 189. (in Chinese)
- [2] 朱博,董义鹏,金锋. 载人航天器仪表系统红外触摸屏硬件电路设计[J]. 现代电子技术, 2013, 36(18): 39-43.
Zhu B, Dong Y P, Jin F. Hardware circuit design of infrared touch screen on instrument system of manned spacecraft[J]. *Modern Electronics Technique*, 2013, 36(18): 39-43. (in Chinese)
- [3] 杨顺,李玉翠. 红外触摸屏非正交扫描算法研究[J]. 激光杂志, 2014, 35(6): 15-17.
Yang S, Li Y C. Research on infrared touch screen for non-orthogonal scanning algorithm[J]. *Laser Journal*, 2014,

35(6): 15-17. (in Chinese)

[4] 李方园. 触摸屏工程应用[M]. 北京:电子工业出版社,2008.
Li F Y. *Touch Screen Engineering Application* [M]. Beijing: *Electronics Industry Press* , 2008.

[5] 吕焱,邓春健,李文生. 高分辨率多点触控红外触摸屏设计[J]. 液晶与显示,2015,30(1):77-82.
Lv Y, Deng C J, Li W S. Design of high resolution infrared multi-touch screen[J]. *Chinese Journal of Liquid Crystals and Displays* , 2015, 30(1): 77-82. (in Chinese)

作者简介:郭瑞(1980—),女,山西临汾人,硕士,中级工程师,主要从事机载多功能显示器设计方面的研究。E-mail: 15343409087@163.com