



(12) 发明专利申请

(10) 申请公布号 CN 104858541 A

(43) 申请公布日 2015. 08. 26

(21) 申请号 201510338565. 5

(22) 申请日 2015. 06. 17

(71) 申请人 天津七所高科技有限公司

地址 300409 天津市北辰区北辰科技园高新大道 64 号

(72) 发明人 高忠林 路向琨 成海晖

(74) 专利代理机构 天津滨海科纬知识产权代理有限公司 12211

代理人 陈雅洁

(51) Int. Cl.

B23K 11/36(2006. 01)

B23K 11/24(2006. 01)

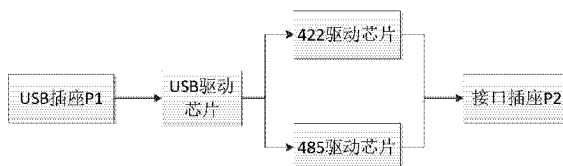
权利要求书3页 说明书7页 附图2页

(54) 发明名称

基于 Android 系统的焊机控制器的控制系统和方法

(57) 摘要

本发明创造提供基于 Android 系统的焊机控制器的控制系统和方法,利用手持设备对焊机控制器进行编程和监测,经过一系列电路设计,使得通讯更加稳定可靠,各种状态指示灯齐全,为以后的硬件故障定位提供了很大方便;通讯板可智能判断是否与手持设备连接,进行睡眠与正常工作的模式切换,实现了低耗能要求;可以根据用户的不同需要在 2 种总线中进行选择;整体硬件结构简单,连接方便,可靠性高。软件界面更加友好,操作简单,人机交互性能有很大改善,使得用户体验性有了很大程度的提高。由于采用上述技术方案,利用手持设备对焊机控制器进行编程,使得系统硬件连接简单、结构简洁,手机 APP 软件界面更加友好,使用方便,系统的稳定性高等优点。



1. 基于 Android 系统的控制焊机控制器的系统,其特征在于:包括手持设备、通讯板、焊机控制器,所述手持设备与通讯板相连,所述通讯板与焊机控制器相连,所述通讯板包括 USB 插座 P1、USB 驱动芯片单元、串口驱动芯片单元和接口插座 P2,所述 USB 插座 P1 与所述 USB 驱动芯片单元相连,所述串口驱动芯片单元包括 422 驱动芯片电路和 485 驱动芯片电路,所述 USB 驱动芯片单元分别与所述 422 驱动芯片电路和所述 485 驱动芯片电路相连,所述 422 驱动芯片电路和所述 485 驱动芯片电路均与所述接口插座 P2 相连。

2. 根据权利要求 1 所述的基于 Android 系统的控制焊机控制器的系统,其特征在于:所述 USB 驱动芯片单元包括驱动芯片 U1 和拨码开关 S1,所述 USB 驱动芯片 U1 的第 9 引脚与电阻 R2 一端相连,所述电阻 R2 另一端与所述 USB 插座 P1 的第 2 引脚相连,同时所述电阻 R2 另一端与电容 C1 的一端相连,所述电容 C1 另一端接地;所述 USB 驱动芯片 U1 的第 8 引脚与电阻 R4 的一端相连,所述电阻 R4 的另一端与所述 USB 插座 P1 的第 3 引脚相连,同时所述电阻 R4 的另一端与电容 C2 的一端相连,所述电容 C2 另一端接地;所述 USB 驱动芯片 U1 的第 15 引脚与电阻 R11 和电阻 R7 的一端相连,所述电阻 R11 另一端接地,所述电阻 R7 另一端接所述 USB 插座 P1 的第 1 引脚;所述 USB 驱动芯片 U1 的第 1、第 4、第 2、第 6 引脚分别与电阻 R1、电阻 R3、电阻 R5、电阻 R6 一端相连,所述电阻 R1、电阻 R3、电阻 R5、电阻 R6 的另一端均接到电源 +5V;所述 USB 驱动芯片 U1 的第 14、第 7、第 16 引脚分别与 LED 灯 D1、LED 灯 D2、LED 灯 D3 的负极相连,所述 LED 灯 D1、LED 灯 D2、LED 灯 D3 的正极分别与电阻 R8、电阻 R9、电阻 R10 的一端相连,所述电阻 R8、电阻 R9、电阻 R10 的另一端与所述 USB 驱动芯片 U1 的第 11、第 10、第 3 引脚共同连接到电容 C3 的一端,所述电容 C3 的另一端与所述 USB 驱动芯片 U1 的第 13、第 5 引脚相连;在所述电阻 R1 与所述 USB 驱动芯片 U1 的第 1 引脚之间引出输出信号 TXD 信号,所述 TXD 信号与所述拨码开关 S1 的第 1、第 2 引脚相连;在所述电阻 R3 与所述 USB 驱动芯片 U1 的第 4 引脚之间引出输入信号 RXD 信号,所述 RXD 信号与所述拨码开关 S1 的第 3、第 4 引脚相连,所述拨码开关 S1 的第 8、第 7、第 6、第 5 引脚分别输出 TXD 422、TXD 485、RXD 422、RXD 485 信号。

3. 根据权利要求 2 所述的基于 Android 系统的控制焊机控制器的系统,其特征在于:所述 USB 驱动芯片 U1 的型号为 FTDI230X。

4. 根据权利要求 1 所述的基于 Android 系统的控制焊机控制器的系统,其特征在于:所述 422 驱动芯片电路包括串口驱动芯片 U2,所述串口驱动芯片 U2 的第 7 引脚分别接电阻 R14 与电阻 R15 的一端,所述电阻 R15 另一端接地;所述串口驱动芯片 U2 的第 8 引脚与所述电阻 R14 另一端相连,同时所述串口驱动芯片 U2 的第 8 引脚与电阻 R12 一端相连,所述电阻 R12 另一端接 +5V;所述串口驱动芯片 U2 的第 1 引脚接 +5V;所述串口驱动芯片 U2 的第 4 引脚接 GND;所述串口驱动芯片 U2 的第 5、第 6 引脚之间并联电阻 R13;所述串口驱动芯片 U2 的第 2 引脚与所述拨码开关 S1 的第 6 引脚相连;所述串口驱动芯片 U2 的第 3 引脚与所述拨码开关 S1 的第 8 引脚相连。

5. 根据权利要求 1 所述的基于 Android 系统的控制焊机控制器的系统,其特征在于:所述 485 驱动芯片电路包括串口驱动芯片 U3,所述串口驱动芯片 U3 的第 1 引脚与所述拨码开关 S1 的第 5 引脚相连;所述串口驱动芯片 U3 的第 4 引脚与所述拨码开关 S1 的第 7 引脚相连;所述串口驱动芯片 U3 的第 2 引脚与第 3 引脚同时连接到 P-MOS 管 Q1 的第 3 引脚,同时所述串口驱动芯片 U3 的第 2 引脚与第 3 引脚与电阻 R17 一端相连,所述电阻 R17 另一

端接地,所述 P-MOS 管的第 2 引脚接 +5V,所述 P-MOS 管的第 1 引脚接电阻 R16 一端,所述电阻 R16 另一端分别与所述拨码开关 S1 的第 7 引脚和所述串口驱动芯片 U3 的第 4 引脚相连;所述串口驱动芯片 U3 的第 8 引脚接 +5V;所述串口驱动芯片 U3 的第 5 引脚接地;所述串口驱动芯片 U3 的第 6、第 7 引脚之间并联电阻 R18。

6. 根据权利要求 4 所述的基于 Android 系统的控制焊机控制器的系统,其特征在于:所述串口驱动芯片 U2 的第 5、第 6、第 7、第 8 引脚分别引出 422Y+2、422Z-2、422B-2、422A+2 信号线,所述串口驱动芯片 U3 的第 6、第 7 引脚分别引出 485A+2、485B-2 信号线,所述 422Y+2、422Z-2、422B-2、422A+2、485A+2、485B-2 信号线均连接到所述接口插座 P2 上。

7. 根据权利要求 6 所述的基于 Android 系统的控制焊机控制器的系统,其特征在于:所述串口驱动芯片 U2 的型号为 ISL8488。

8. 根据权利要求 5 所述的基于 Android 系统的控制焊机控制器的系统,其特征在于:所述串口驱动芯片 U3 的型号为 MAX485。

9. 根据权利要求 8 所述的基于 Android 系统的控制焊机控制器的系统,其特征在于:所述 P-MOS 管 Q1 的型号为 BSS84。

10. 基于 Android 系统的控制焊机控制器的方法,其特征在于:包括以下步骤:

一、程序开始运行后首先判断是否创建了对应的数据库文件和故障记录 text 文件,如果已建立两个所述文件,则直接跳到步骤二,否则创建两个所述文件;

二、装载 USB 驱动程序;

三、开辟后台线程监测是否发生故障,如果发生故障,则进入故障处理模块,否则后台监测是否进入状态检测模块、编程器参数编程模块、主机参数编程模块、参数下载模块、参数上传模块,

所述故障处理模块首先进行故障数据解析判断故障类型,如果是通讯故障则弹出对话框提示通讯故障的类型;否则,读取数据缓冲区数据,通过对十六进制数据进行解析后判断故障类型,如果该故障类型与前一个故障类型不同则将该故障报警的发生时间和类型记录到故障报警文件中,否则就判断本次故障发生时间与上次故障发生时间之间的间隔,时间间隔大于设定值则将故障报警信息保存到故障报警文件中,否则认为是主机连续发送了多个相同的故障信息,故舍弃本次的故障报警信息,

所述状态监测模块不断循环读取接收数据缓冲区中未读取的字节数,判断当字节数 >0 时,读取相应的字节;判断数据格式是否正确,如果正确则在主线程页面中显示读到的各参数值,否则进入故障报警模块,

所述编程器参数编程模块首先加载程序页面选择的规范号和模式,然后链接对应的数据中的相应表单,读取表单数据后在主线程页面中进行全部参数显示,并可以在页面中进行修改保存操作,

所述主机参数编程模块首先加载程序页面选择的规范号和模式,然后发送相应的命令调取主机的相应参数;返回数据格式正确后在主线程页面中进行全部参数显示,并可以进行修改保存操作,

所述参数下载模块首先加载所有已选定的规范和模式,随后发送第一组规范的数据,发送成功后判断是否已发送完成全部已选择的规范,如果还有未发送完成的,则继续发送下一组规范,否则主线程页面在 Toast 对话框中提示全部规范发送成功;此时用户可在主

机参数编程模块查看各规范参数的具体内容,一旦有规范发送失败则进入故障处理模块,

所述参数上传模块首先加载所有已选定的规范和模式,随后索要第一组规范数据,返回数据成功且格式正确后,将数据保存到对应的数据库表单中;判断是否已索要了全部已选择的规范,如果还有未索要的规范,则继续索要一下组规范参数,否则主线程页面在 Toast 对话框中提示全部规范以上传成功;此时用户可以在编程器参数编程模块中查看所有上传的参数是否合适、是否需要修改,一旦有规范索要失败,则进入故障处理模块。

## 基于 Android 系统的焊机控制器的控制系统和方法

### 技术领域

[0001] 本发明创造属于焊机控制器技术领域,尤其是涉及一种基于 Android 系统的焊机控制器的控制系统和方法。

### 背景技术

[0002] 电阻点焊具有生产效率高、焊接变形小、劳动条件好、无需另加焊接材料、操作简单等优点,广泛应用于汽车、家电制造等行业。编程器是焊机控制器的重要部件,用于对焊接参数进行设定以及监测焊接过程中的参数变化,对焊接质量的保证提供重要支撑。现有的编程器多采用单片机 MCU 作为主芯片,通过液晶屏和物理按键与用户交互,进行相应状态切换,通过单片机 UART 口经相应的驱动 IC 进行电平转换后和控制器进行串口通信,完成相应参数的设定和状态监测。但是,目前的这种方案用户体验效果差,编程器重量大,液晶屏幕窗口较小使得界面菜单级联较多,物理按键寿命短,而且单片机性能有限,操作费时,难以全面达到个别用户的定制需求。此外,硬件相对复杂,增加了不可靠因素,成品较高。

### 发明内容

[0003] 本发明创造要解决以上技术问题,提供一种基于 Android 系统的焊机控制器的控制系统和方法。

[0004] 为解决上述技术问题,本发明创造采用的技术方案是:基于 Android 系统的控制焊机控制器的系统,包括手持设备、通讯板、焊机控制器,所述手持设备与通讯板相连,所述通讯板与焊机控制器相连,所述通讯板包括 USB 插座 P1、USB 驱动芯片单元、串口驱动芯片单元和接口插座 P2,所述 USB 插座 P1 与所述 USB 驱动芯片单元相连,所述串口驱动芯片单元包括 422 驱动芯片电路和 485 驱动芯片电路,所述 USB 驱动芯片单元分别与所述 422 驱动芯片电路和所述 485 驱动芯片电路相连,所述 422 驱动芯片电路和所述 485 驱动芯片电路均与所述接口插座 P2 相连。

[0005] 进一步,所述 USB 驱动芯片单元包括驱动芯片 U1 和拨码开关 S1,所述 USB 驱动芯片 U1 的第 9 引脚与电阻 R2 一端相连,所述电阻 R2 另一端与所述 USB 插座 P1 的第 2 引脚相连,同时所述电阻 R2 另一端与电容 C1 的一端相连,所述电容 C1 另一端接地;所述 USB 驱动芯片 U1 的第 8 引脚与电阻 R4 的一端相连,所述电阻 R4 的另一端与所述 USB 插座 P1 的第 3 引脚相连,同时所述电阻 R4 的另一端与电容 C2 的一端相连,所述电容 C2 另一端接地;所述 USB 驱动芯片 U1 的第 15 引脚与电阻 R11 和电阻 R7 的一端相连,所述电阻 R11 另一端接地,所述电阻 R7 另一端接所述 USB 插座 P1 的第 1 引脚;所述 USB 驱动芯片 U1 的第 1、第 4、第 2、第 6 引脚分别与电阻 R1、电阻 R3、电阻 R5、电阻 R6 一端相连,所述电阻 R1、电阻 R3、电阻 R5、电阻 R6 的另一端均接到电源 +5V;所述 USB 驱动芯片 U1 的第 14、第 7、第 16 引脚分别与 LED 灯 D1、LED 灯 D2、LED 灯 D3 的负极相连,所述 LED 灯 D1、LED 灯 D2、LED 灯 D3 的正极分别与电阻 R8、电阻 R9、电阻 R10 的一端相连,所述电阻 R8、电阻 R9、电阻 R10 的另一端与所述 USB 驱动芯片 U1 的第 11、第 10、第 3 引脚共同连接到电容 C3 的一端,所述电容 C3

的另一端与所述 USB 驱动芯片 U1 的第 13、第 5 引脚相连；在所述电阻 R1 与所述 USB 驱动芯片 U1 的第 1 引脚之间引出输出信号 TXD 信号，所述 TXD 信号与所述拨码开关 S1 的第 1、第 2 引脚相连；在所述电阻 R3 与所述 USB 驱动芯片 U1 的第 4 引脚之间引出输入信号 RXD 信号，所述 RXD 信号与所述拨码开关 S1 的第 3、第 4 引脚相连，所述拨码开关 S1 的第 8、第 7、第 6、第 5 引脚分别输出 TXD 422、TXD 485、RXD 422、RXD 485 信号。

[0006] 进一步，所述 USB 驱动芯片 U1 的型号为 FTDI230X。

[0007] 进一步，所述 422 驱动芯片电路包括串口驱动芯片 U2，所述串口驱动芯片 U2 的第 7 引脚分别接电阻 R14 与电阻 R15 的一端，所述电阻 R15 另一端接地；所述串口驱动芯片 U2 的第 8 引脚与所述电阻 R14 另一端相连，同时所述串口驱动芯片 U2 的第 8 引脚与电阻 R12 一端相连，所述电阻 R12 另一端接 +5V；所述串口驱动芯片 U2 的第 1 引脚接 +5V；所述串口驱动芯片 U2 的第 4 引脚接 GND；所述串口驱动芯片 U2 的第 5、第 6 引脚之间并联电阻 R13；所述串口驱动芯片 U2 的第 2 引脚与所述拨码开关 S1 的第 6 引脚相连；所述串口驱动芯片 U2 的第 3 引脚与所述拨码开关 S1 的第 8 引脚相连。

[0008] 进一步，所述 485 驱动芯片电路包括串口驱动芯片 U3，所述串口驱动芯片 U3 的第 1 引脚与所述拨码开关 S1 的第 5 引脚相连；所述串口驱动芯片 U3 的第 4 引脚与所述拨码开关 S1 的第 7 引脚相连；所述串口驱动芯片 U3 的第 2 引脚与第 3 引脚同时连接到 P-MOS 管 Q1 的第 3 引脚，同时所述串口驱动芯片 U3 的第 2 引脚与第 3 引脚与电阻 R17 一端相连，所述电阻 R17 另一端接地，所述 P-MOS 管的第 2 引脚接 +5V，所述 P-MOS 管的第 1 引脚接电阻 R16 一端，所述电阻 R16 另一端分别与所述拨码开关 S1 的第 7 引脚和所述串口驱动芯片 U3 的第 4 引脚相连；所述串口驱动芯片 U3 的第 8 引脚接 +5V；所述串口驱动芯片 U3 的第 5 引脚接地；所述串口驱动芯片 U3 的第 6、第 7 引脚之间并联电阻 R18。

[0009] 进一步，所述串口驱动芯片 U2 的第 5、第 6、第 7、第 8 引脚分别引出 422Y+2、422Z-2、422B-2、422A+2 信号线，所述串口驱动芯片 U3 的第 6、第 7 引脚分别引出 485A+2、485B-2 信号线，所述 422Y+2、422Z-2、422B-2、422A+2、485A+2、485B-2 信号线均连接到所述接口插座 P2 上。

[0010] 进一步的，所述串口驱动芯片 U2 的型号为 ISL8488。

[0011] 进一步的，所述串口驱动芯片 U3 的型号为 MAX485。

[0012] 进一步的，所述 P-MOS 管 Q1 的型号为 BSS84。

[0013] 基于 Android 系统的控制焊机控制器的方法，包括以下步骤：

[0014] 一、程序开始运行后首先判断是否创建了对应的数据库文件和故障记录 text 文件，如果已建立两个所述文件，则直接跳到步骤二，否则创建两个所述文件；

[0015] 二、装载 USB 驱动程序；

[0016] 三、开辟后台线程监测是否发生故障，如果发生故障，则进入故障处理模块，否则后台监测是否进入状态检测模块、编程器参数编程模块、主机参数编程模块、参数下载模块、参数上传模块，

[0017] 所述故障处理模块首先进行故障数据解析判断故障类型，如果是通讯故障则弹出对话框提示通讯故障的类型；否则，读取数据缓冲区数据，通过对十六进制数据进行解析后判断故障类型，如果该故障类型与前一个故障类型相同则将该故障报警的发生时间和类型记录到故障报警文件中，否则就判断本次故障发生时间与上次故障发生时间之间的间隔，

时间间隔大于设定值则将故障报警信息保存到故障报警文件中,否则认为是主机连续发送了多个相同的故障信息,故舍弃本次的故障报警信息,

[0018] 所述状态监测模块不断循环读取接收数据缓冲区中未读取的字节数,判断当字节数 > 0 时,读取相应的字节;判断数据格式是否正确,如果正确则在主线程页面中显示读到的各参数值,否则进入故障报警模块,

[0019] 所述编程器参数编程模块首先加载程序页面选择的规范号和模式,然后链接对应的数据中的相应表单,读取表单数据后在主线程页面中进行全部参数显示,并可以在页面中进行修改保存操作,

[0020] 所述主机参数编程模块首先加载程序页面选择的规范号和模式,然后发送相应的命令调取主机的相应参数;返回数据格式正确后在主线程页面中进行全部参数显示,并可以进行修改保存操作,

[0021] 所述参数下载模块首先加载所有已选定的规范和模式,随后发送第一组规范的数据,发送成功后判断是否已选择的全部规范发送完成,如果还有未发送完成的,则继续发送下一组规范,否则主线程页面在 Toast 对话框中提示全部规范发送成功;此时用户可在主机参数编程模块查看各规范参数的具体内容,一旦有规范发送失败则进入故障处理模块,

[0022] 所述参数上传模块首先加载所有已选定的规范和模式,随后索要第一组规范数据,返回数据成功且格式正确后,将数据保存到对应的数据库表单中;判断是否已索要了全部已选择的规范,如果还有未索要的规范,则继续索要一下组规范参数,否则主线程页面在 Toast 对话框中提示全部规范以上传成功;此时用户可以在编程器参数编程模块中查看所有上传的参数是否合适、是否需要修改,一旦有规范索要失败,则进入故障处理模块。

[0023] 本发明创造具有的优点和积极效果是:基于 Android 系统的焊机控制器的控制系统和方法,利用手持设备对焊机控制器进行编程,经过一系列电路设计,通讯更加稳定可靠,各种状态指示灯齐全,为以后的硬件故障定位提供了很大方便;通讯板可智能判断是否与手持设备连接,进行睡眠与正常工作的模式切换,实现了低耗能要求;可以根据用户的不同需要在 2 种总线中进行选择;整体硬件结构简单,连接方便,可靠性高。软件上为适应主机存储大量数据的方便性,在具体参数的数据表示方法中使用由 4 个 16 进制字节表示的单精度浮点数,使得主机在计算存储数据的偏移地址时可以按照固定字长进行计算,方便了主机操作;软件程序中大量耗时的通讯、数据解析与转换、数据库操作等在后台线程中运行,使得整个页面的跳转和响应更加快速灵活;软件将参数数据和故障报警信息分别保存为 SQLite 数据库和文本文件,方便了第三方软件的进一步开发利用;总体上,整个软件界面更加友好,操作简单,人机交互性能有很大改善,使得用户体验性有了很大程度的提高。由于采用上述技术方案,利用手持设备对焊机控制器进行编程,使得系统硬件连接简单、结构简洁,手机 APP 软件界面更加友好,使用方便,系统的稳定性高等优点。

## 附图说明

[0024] 图 1 是本发明创造的系统框图;

[0025] 图 2 是通讯板原理框图;

[0026] 图 3 是 USB 驱动芯片单元原理图;

[0027] 图 4 是 422 驱动芯片电路原理图;

[0028] 图 5 是 485 驱动芯片电路原理图。

### 具体实施方式

[0029] 下面结合附图对本发明创造的具体实施例做详细说明。

[0030] 如图 1-5 所示,基于 Android 系统的控制焊机控制器的系统,包括手持设备、通讯板、焊机控制器,所述手持设备与通讯板相连,所述通讯板与焊机控制器相连,所述通讯板包括 USB 插座 P1、USB 驱动芯片单元、串口驱动芯片单元和接口插座 P2,所述 USB 插座 P1 与所述 USB 驱动芯片单元相连,所述串口驱动芯片单元包括 422 驱动芯片电路和 485 驱动芯片电路,所述 USB 驱动芯片单元分别与所述 422 驱动芯片电路和所述 485 驱动芯片电路相连,所述 422 驱动芯片电路和所述 485 驱动芯片电路均与所述接口插座 P2 相连。

[0031] 所述 USB 驱动芯片单元包括驱动芯片 U1 和拨码开关 S1,所述 USB 驱动芯片 U1 的第 9 引脚与电阻 R2 一端相连,所述电阻 R2 另一端与所述 USB 插座 P1 的第 2 引脚相连,同时所述电阻 R2 另一端与电容 C1 的一端相连,所述电容 C1 另一端接地;所述 USB 驱动芯片 U1 的第 8 引脚与电阻 R4 的一端相连,所述电阻 R4 的另一端与所述 USB 插座 P1 的第 3 引脚相连,同时所述电阻 R4 的另一端与电容 C2 的一端相连,所述电容 C2 另一端接地;所述 USB 驱动芯片 U1 的第 15 引脚与电阻 R11 和电阻 R7 的一端相连,所述电阻 R11 另一端接地,所述电阻 R7 另一端接所述 USB 插座 P1 的第 1 引脚;所述 USB 驱动芯片 U1 的第 1、第 4、第 2、第 6 引脚分别与电阻 R1、电阻 R3、电阻 R5、电阻 R6 一端相连,所述电阻 R1、电阻 R3、电阻 R5、电阻 R6 的另一端均接到电源 +5V;所述 USB 驱动芯片 U1 的第 14、第 7、第 16 引脚分别与 LED 灯 D1、LED 灯 D2、LED 灯 D3 的负极相连,所述 LED 灯 D1、LED 灯 D2、LED 灯 D3 的正极分别与电阻 R8、电阻 R9、电阻 R10 的一端相连,所述电阻 R8、电阻 R9、电阻 R10 的另一端与所述 USB 驱动芯片 U1 的第 11、第 10、第 3 引脚共同连接到电容 C3 的一端,所述电容 C3 的另一端与所述 USB 驱动芯片 U1 的第 13、第 5 引脚相连;在所述电阻 R1 与所述 USB 驱动芯片 U1 的第 1 引脚之间引出输出信号 TXD 信号,所述 TXD 信号与所述拨码开关 S1 的第 1、第 2 引脚相连;在所述电阻 R3 与所述 USB 驱动芯片 U1 的第 4 引脚之间引出输入信号 RXD 信号,所述 RXD 信号与所述拨码开关 S1 的第 3、第 4 引脚相连,所述拨码开关 S1 的第 8、第 7、第 6、第 5 引脚分别输出 TXD 422、TXD 485、RXD 422、RXD 485 信号。

[0032] 所述 USB 驱动芯片 U1 的型号为 FTDI230X。

[0033] 所述 422 驱动芯片电路包括串口驱动芯片 U2,所述串口驱动芯片 U2 的第 7 引脚分别接电阻 R14 与电阻 R15 的一端,所述电阻 R15 另一端接地;所述串口驱动芯片 U2 的第 8 引脚与所述电阻 R14 另一端相连,同时所述串口驱动芯片 U2 的第 8 引脚与电阻 R12 一端相连,所述电阻 R12 另一端接 +5V;所述串口驱动芯片 U2 的第 1 引脚接 +5V;所述串口驱动芯片 U2 的第 4 引脚接 GND;所述串口驱动芯片 U2 的第 5、第 6 引脚之间并联电阻 R13;所述串口驱动芯片 U2 的第 2 引脚与所述拨码开关 S1 的第 6 引脚相连;所述串口驱动芯片 U2 的第 3 引脚与所述拨码开关 S1 的第 8 引脚相连。

[0034] 所述 485 驱动芯片电路包括串口驱动芯片 U3,所述串口驱动芯片 U3 的第 1 引脚与所述拨码开关 S1 的第 5 引脚相连;所述串口驱动芯片 U3 的第 4 引脚与所述拨码开关 S1 的第 7 引脚相连;所述串口驱动芯片 U3 的第 2 引脚与第 3 引脚同时连接到 P-MOS 管 Q1 的第 3 引脚,同时所述串口驱动芯片 U3 的第 2 引脚与第 3 引脚与电阻 R17 一端相连,所述电



阻 R17 另一端接地,所述 P-MOS 管的第 2 引脚接 +5V,所述 P-MOS 管的第 1 引脚接电阻 R16 一端,所述电阻 R16 另一端分别与所述拨码开关 S1 的第 7 引脚和所述串口驱动芯片 U3 的第 4 引脚相连;所述串口驱动芯片 U3 的第 8 引脚接 +5V;所述串口驱动芯片 U3 的第 5 引脚接地;所述串口驱动芯片 U3 的第 6、第 7 引脚之间并联电阻 R18。

[0035] 所述串口驱动芯片 U2 的第 5、第 6、第 7、第 8 引脚分别引出 422Y+2、422Z-2、422B-2、422A+2 信号线,所述串口驱动芯片 U3 的第 6、第 7 引脚分别引出 485A+2、485B-2 信号线,所述 422Y+2、422Z-2、422B-2、422A+2、485A+2、485B-2 信号线均连接到所述接口插座 P2 上。

[0036] 所述串口驱动芯片 U2 的型号为 ISL8488。

[0037] 所述串口驱动芯片 U3 的型号为 MAX485。

[0038] 所述 P-MOS 管 Q1 的型号为 BSS84。

[0039] 基于 Android 系统的控制焊机控制器的方法,包括以下步骤:

[0040] 一、程序开始运行后首先判断是否创建了对应的数据库文件和故障记录 text 文件,如果已建立两个所述文件,则直接跳到步骤二,否则创建两个所述文件;

[0041] 二、装载 USB 驱动程序;

[0042] 三、开辟后台线程监测是否发生故障,如果发生故障,则进入故障处理模块,否则后台监测是否进入状态检测模块、编程器参数编程模块、主机参数编程模块、参数下载模块、参数上传模块,

[0043] 所述故障处理模块首先进行故障数据解析判断故障类型,如果是通讯故障则弹出对话框提示通讯故障的类型;否则,读取数据缓冲区数据,通过对十六进制数据进行解析后判断故障类型,如果该故障类型与前一个故障类型相同则将该故障报警的发生时间和类型记录到故障报警文件中,否则就判断本次故障发生时间与上次故障发生时间之间的间隔,时间间隔大于设定值则将故障报警信息保存到故障报警文件中,否则认为是主机连续发送了多个相同的故障信息,故舍弃本次的故障报警信息,

[0044] 所述状态监测模块不断循环读取接收数据缓冲区中未读取的字节数,判断当字节数 >0 时,读取相应的字节;判断数据格式是否正确,如果正确则在主线程页面中显示读到的各参数值,否则进入故障报警模块,

[0045] 所述编程器参数编程模块首先加载程序页面选择的规范号和模式,然后链接对应的数据中的相应表单,读取表单数据后在主线程页面中进行全部参数显示,并可以在页面中进行修改保存操作,

[0046] 所述主机参数编程模块首先加载程序页面选择的规范号和模式,然后发送相应的命令调取主机的相应参数;返回数据格式正确后在主线程页面中进行全部参数显示,并可以进行修改保存操作,

[0047] 所述参数下载模块首先加载所有已选定的规范和模式,随后发送第一组规范的数据,发送成功后判断是否已选择的全部规范发送完成,如果还有未发送完成的,则继续发送下一组规范,否则主线程页面在 Toast 对话框中提示全部规范发送成功;此时用户可在主机参数编程模块查看各规范参数的具体内容,一旦有规范发送失败则进入故障处理模块,

[0048] 所述参数上传模块首先加载所有已选定的规范和模式,随后索要第一组规范数据,返回数据成功且格式正确后,将数据保存到对应的数据库表单中;判断是否已索要了全

部已选择的规范,如果还有未索要的规范,则继续索要一下组规范参数,否则主线程页面在 Toast 对话框中提示全部规范以上传成功;此时用户可以在编程器参数编程模块中查看所有上传的参数是否合适、是否需要修改,一旦有规范索要失败,则进入故障处理模块。

[0049] 基于 Android 系统的控制焊机控制器的系统的工作过程为:使用时,USB 驱动芯片 U1 的 8 脚、9 脚与相应的电阻连接,通过 2 个连接电阻的另一端与 USB 插座 P1 的 2、3 脚连接,分别接入 USBDP 和 USBDM 信号;串口驱动芯片 U2 的 5、6、7、8 脚分别引出 422Y+2、422Z-2、422B-2、422A+2 共 4 个信号线,所述串口驱动芯片 U3 的 6、7 脚分别引出 485A+2、485B-2 共 2 个信号线,所述 6 个信号线连接到接口插座 P2 上,完成与焊机控制器的通讯。通讯接口可以通过拨码开关 S1 的选择在 422 与 485 之间转换。

[0050] 手持设备通过 USB 数据线与通讯板的 USB 插座 P1 连接,通讯板的通讯接口与控制器的通讯接口通过通讯电缆连接,硬件连接完成,打开电源开关。当连接上手持设备后,USB 驱动芯片 U1 的 15 脚电平为高,USB 驱动芯片 U1 从休眠状态转到正常工作状态。

[0051] 手持设备编程器的信号经 USB 数据线传输到通讯板,通讯板上该信号经 R2、R4 后连接到 USB 驱动芯片 U1 的 8 脚、9 脚,经过 USB 驱动芯片 U1 从 1 脚输出,同时 USB 驱动芯片 U1 的 14 脚电平发生变化,使得 LED 灯 D1 闪烁指示;当选择 422 总线通讯时,USB 驱动芯片 U1 的 1 脚与拨码开关 S1 的 8 脚相连接,并最终连接到串口驱动芯片 U2 的 3 脚,经串口驱动 IC 后输出信号从串口驱动芯片 U2 的 5 脚、6 脚连接到通讯接口,通过通讯电缆该信号传输到主机;当选择 485 总线通讯时,USB 驱动芯片 U1 的 1 脚与拨码开关 S1 的 7 脚相连接,并最终连接到串口驱动芯片 U3 的 4 脚和 R16 连接,信号经 P-MOS 管 Q1 后串口驱动芯片 U3 从默认接收模式转换发送模式,信号从串口驱动芯片 U3 的 4 脚进入,经串口驱动 IC 后输出信号从串口驱动芯片 U3 的 6 脚、7 脚连接到通讯接口,通过通讯电缆该信号传输到主机。

[0052] 主机收到编程器命令后,进行相应的操作,并返回对应的数据。该数据的信号将通讯电缆输入到通讯板;当选择 422 总线模式时,该信号进入串口驱动芯片 U2 的 7 脚、8 脚,并从串口驱动芯片 U2 的 2 脚输出后连接到拨码开关 S1 的 6 脚;当选择 485 总线模式时,该信号进入串口驱动芯片 U3 的 6 脚、7 脚,并从串口驱动芯片 U3 的 1 脚输出后连接到拨码开关 S1 的 5 脚;该信号最终输入到 USB 驱动芯片 U1 的 4 脚,同时 USB 驱动芯片 U1 的 7 脚电平发生变化,使得 LED 灯 D2 闪烁指示;信号经 USB 驱动 IC 将信号转换为 USB 数据信号,通过 USB 驱动芯片 U1 的 8 脚、9 脚,经 USB 数据线传输到编程器中,此时数据被保存在手持设备内存的数据缓冲区中。

[0053] 打开软件,程序首先判断在安装目录下是否有所需要的数据库以及故障报警 text 文本文件,如果没有则马上创建这两个文件;文件创建完成后装载相应的 USB 驱动程序,此时会判断编程器是否与通讯板已连接,若未连接则提示未连接;连接成功后开辟新的线程用于监测是否有故障报警发生;如果发生故障报警则进入故障报警模块,否则等待用户点击主页面的不同按钮,用于进入其他不同的功能模块。

[0054] 进入故障报警模块后,首先进行故障数据解析判断故障类型。如果是通讯故障则弹出对话框提示通讯故障的类型;否则,读取数据缓冲区数据,通过对十六进制数据进行解析后判断故障类型,如果该故障类型与前一个故障类型相同则将该故障报警的发生时间和类型记录到故障报警文件中,否则就判断本次故障发生时间与上次故障发生时间之间的间隔,时间间隔大于设定值则将故障报警信息保存到故障报警文件中,否则认为是主机连续

发送了多个相同的故障信息,故舍弃本次的故障报警信息。

[0055] 点击主页面相应按钮进入状态监测模块后,会在后台线程中不断读取数据缓冲区数据,当缓冲区未读取字节数大于 0 时,就读取该缓冲区数据,判断数据格式正确后,经数据解析与转换将代表每个参数的 4 个字节的 16 进制数转换为相应的浮点数值,并在主线程的页面显示中进行显示,一旦流程发生故障后进入故障处理模块。

[0056] 点击主页面相应按钮进入读取编程器参数模块,后台线程首先加载所选定的规范号和模式,链接 SQLite 数据中的对应表单,读取所需的浮点数数据,并在主线程页面中进行展示,可对各参数进行修改或查看,并保存到手持设备的 SQLite 数据库中,一旦流程发生故障后进入故障处理模块。

[0057] 点击主页面相应按钮进入读取主机参数模块,后台线程首先加载所选定的规范号和模式,发送索要相应数据的命令,该命令信号经通讯板后传输到主机,主机收到该命令后将对应的规范数据通过通讯板发送到编程器,编程器读取返回的数据,格式正确后,经数据解析与转换将代表每个参数的 4 个字节的 16 进制数转换为相应的浮点数值,并在主线程的页面显示中进行展示,可对各参数进行修改或查看,并保存到手持设备的 SQLite 数据库中,一旦流程发生故障后进入故障处理模块。

[0058] 点击主页面相应按钮进入参数下载模块,后台线程首先加载所选定的规范号和模式,链接 SQLite 数据中的对应表单,读取所需所有规范的浮点数数据,经数据转换将浮点数数据转换为可方便单片机保存的 4 字节的浮点数形式,发送选择的第一条规范的数据内容,数据信号经通讯板传输到主机,主机收到该数据后保存到相应的存储区,并返回保存成功信号,保存成功信号经通讯板后传输到编程器,编程器则认为此规范下载完成;若还有规范未下载则进行下一规范的下下载,否则在主线程页面显示中提示所有规范下载完成,一旦流程发生故障后进入故障处理模块。

[0059] 点击主页面相应按钮进入参数上传模块,后台线程首先加载所选定的规范号和模式,并发送索要所选的第一条规范的命令,命令信号经通讯板后传输到主机,主机收到该命令,返回对应的规范数据,返回的数据信号经通讯板传输到编程器,编程器收到该数据判断格式正确后,经数据解析与转换将代表每个参数的 4 个字节的 16 进制数转换为相应的浮点数值,链接 SQLite 数据库,将数据保存到对应规范的表单中,若还有规范未上传则进行下一条规范的上传,否则认为所选全部规范上传完成,主线程页面显示中提示所有规范上传完成,一旦流程发生故障后进入故障处理模块。

[0060] 以上对本发明创造的一个实施例进行了详细说明,但所述内容仅为本发明创造的较佳实施例,不能被认为用于限定本发明创造的实施例范围。凡依本发明创造申请范围所作的均等变化与改进等,均应仍归属于本发明创造的专利涵盖范围之内。



图 1

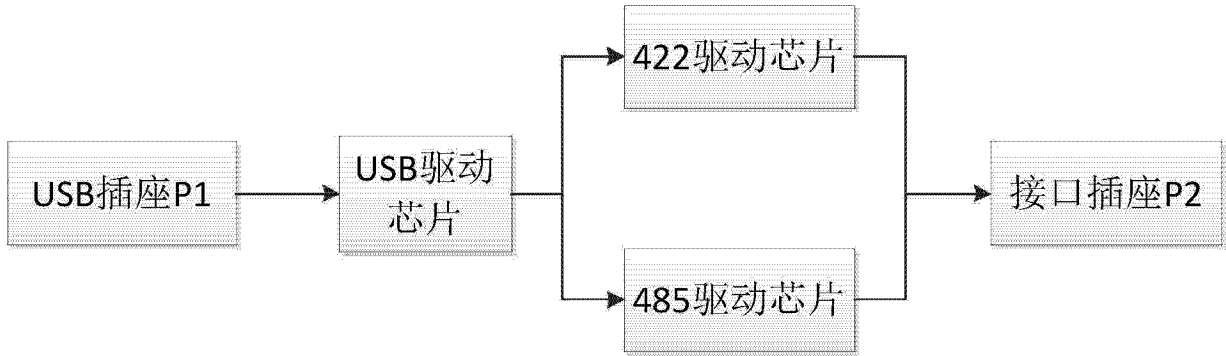


图 2

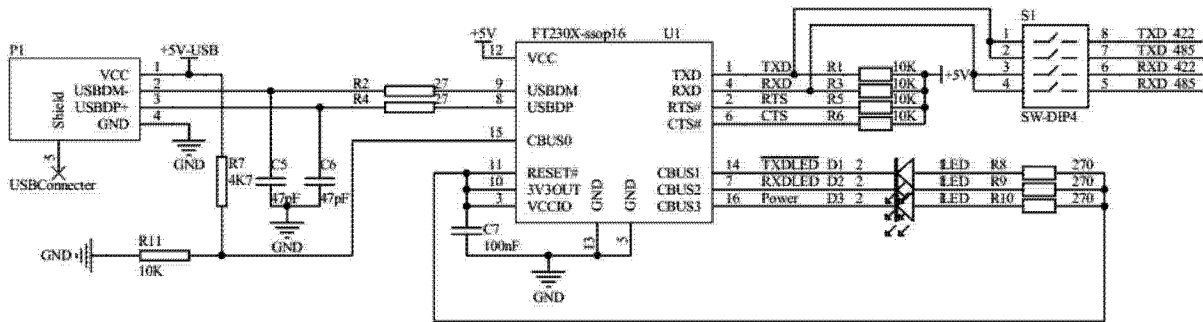


图 3

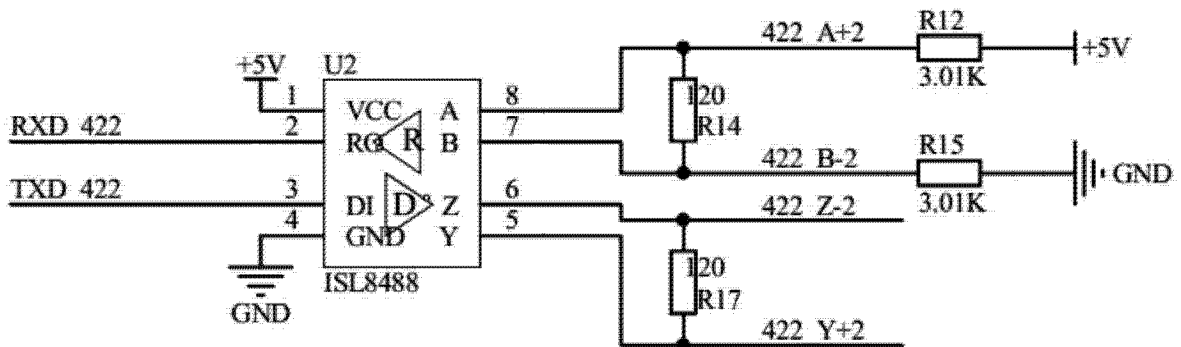


图 4

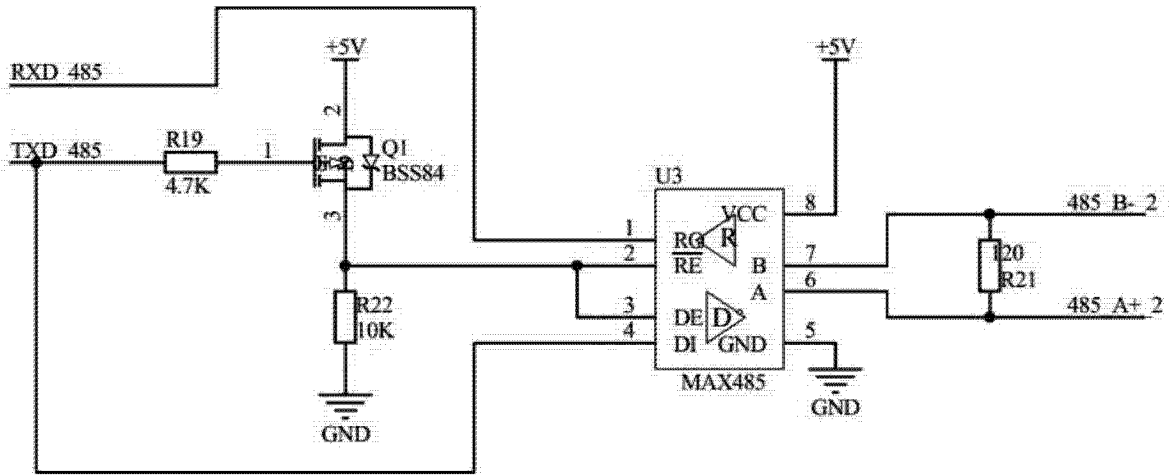


图 5