文章编号：1004-4353(2022)03-0261-06

基于 Unity3D 和 Proteus 平台的智能家居控制系统的设计与仿真

吴云轩
（黎明职业大学 文化传播学院，福建 泉州 362000）

摘要：为提高智能家居控制系统的体验功能，基于 Proteus 和 Unity3D 开发平台构建了一个包含家居环境数据采集、通信控制和家居场景搭建的虚拟仿真系统. 对该系统进行仿真应用表明，该系统不仅可以在 Unity3D 场景中显示 Proteus 平台采集的温度数据和门禁状态，而且也可以通过 Unity3D 交互界面控制 Proteus 平台中的硬件，即实现了双向控制；因此，该系统可以为智能家居控制系统的设计及功能验证提供良好参考.

关键词：智能家居；虚拟仿真；Unity3D 平台；Proteus 平台；人机交互界面
中图分类号：TP37 **文献标识码**：A

Design and simulation of smart home control system based on Unity3D and Proteus platform

WU Yunxuan
(*College of Culture and Communication*, *Liming Vocational University*, *Quanzhou* 362000, *China*)

Abstract: In order to improve the experience function of the smart home control system, a virtual simulation system including home environment data collection, communication control and home scene construction is built based on Proteus and Unity3D development platform. The simulation application of the system shows that the system can not only display temperature data and access status collected by Proteus platform in Unity3D scene, but also control hardware in Proteus platform through Unity3D interactive interface, that is, the system achieve two-way control. Therefore, this system can provide a good reference for the design and functional verification of smart home control system.

Keywords: smart home; virtual simulation; Unity3D platform; Proteus platform; human computer interface

0 引言

随着 5G 和人工智能技术的发展，基于嵌入式系统、无线传感器网络和物联网技术开发的智能家居控制系统给人们带来了更多人性化的功能设计，也大大提高了人们生活的便利性. 在一些需要采集和展示家居环境状态数据的智能家居控制系统中，开发者通常都会设立一个实体的展厅，以满足用户的体验需求，但这种方式目前只适合线下体验，而且费用较高，建设周期较长. 近年来，虚拟现实技术因具有沉浸感、交互性和构想性等优点被应用于智能家居控制系统的仿真设计中[1]. 例如：徐佳[2]利用虚拟现实技术设计了一种可以模拟展示室内部分家居功能的系统，但由于该系统的环境参数是固定的，因此用户的体验感欠佳；甘晨[3]利用 Arduino 单片机(搭建硬件平台)和

收稿日期：2021-11-19
基金项目：福建省教育厅中青年教师教育科研项目(JAS19679)
作者简介：吴云轩(1980—)，男，硕士，副教授，研究方向为数字媒体技术.

Unity3D设计了一种可以感受场景变化的虚拟智能家居系统.唐梦菲等[4]基于Unity3D和单片机应用系统设计了一种手势感应自动开关门衣柜.但在文献[3]和文献[4]中的系统设计中,需要先完成单片机系统硬件电路的设计与调试后才能与Unity3D平台进行交互,因此设计开发时间较长.基于上述研究,本文利用Unity3D和Proteus平台设计了一种具有三维场景展示、漫游交互设计和硬件控制仿真功能的智能家居控制虚拟仿真系统,并通过仿真应用验证了该系统的有效性.

1 系统设计平台及方案

1.1 Unity3D和Proteus软件介绍

1)Unity3D平台.Unity3D是一个跨平台的专业游戏开发引擎,它集成了NVIDIA PhysX物理引擎、光照、动画系统和粒子效果等功能组件,可轻松创建交互式内容,因此被广泛应用于三维虚拟现实场景的设计中.另外,Unity3D还可以跨IOS、Android和Windows等平台发布本地或移动端的应用,因此利用Unity3D还可以实现在不同终端设备上仿真智能家居系统的控制功能,满足不同用户的体验需求.

2)Proteus软件.Proteus是英国Lab Center Electronics公司的EDA工具软件,其器件库封装了多种模拟器件和数字芯片器件[5],如传感器、按键、显示器等外设模型,以及不同类型的单片机芯片模型.利用Proteus可以实现单片机应用系统软、硬件的设计与仿真,以及实现系统功能的开发和验证.

1.2 系统设计方案

本文设计的智能家居控制系统主要包括硬件电路的仿真、虚拟家居环境的构建和人机交互界面的设计,系统结构框图如图1.

1)硬件电路的仿真.本文使用Proteus对硬件电路进行仿真,并使用虚拟串口与Unity3D进行数据通信,以此仿真单片机应用系统与虚拟场景的实时交互控制.

2)虚拟家居环境的构建.根据用户的家居环境和实际需求,本文使用3dsMax创建家居环境数字模型.模型创建完成后,针对不同应用场景设置不同效果的材质贴图,并将其导出为.FBX文件格式加载到Unity3D中[6],以此搭建对应的虚拟家居环境.

3)人机交互界面的设计.本文使用Unity3D设计人机交互界面,其目的是将环境状态数据通过可视化的图形界面进行展示,以此实现不同的交互功能.

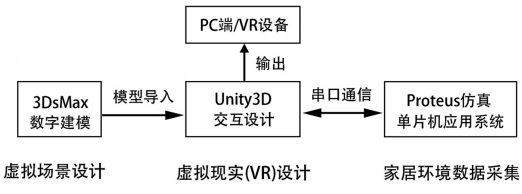

图1 系统结构框图

2 设计案例

2.1 家居环境模型的建立

首先,在3ds Max中制作好房屋的模型,并将该模型的输出文件设置为.FBX格式后导入至Unity3D平台中.然后,使用Unity3D官方标准资源包(Standard Assets)里的预制体和地形编辑器构建室内外场景,如添加草地的材质贴图和树木植被,放置室内外灯具并调整灯光的亮度和照射范围,以及设置摄像机位置等.

2.2 控制系统仿真环境搭建

本文使用Proteus平台仿真单片机控制系统,该系统的电路如图2所示.在系统的输入部分中,选用可直接输出数据的DS18B20温度传感器采集环境温度,其测量范围为$-55\sim+125$ ℃;采用光敏电阻LDR1采集室外光照亮度,亮度数值由单片机集成的A/D器件转换;设置按键K1控制窗帘的开关,设置按键K2模拟门禁传感器.在系统的输出部分中,D1和D2分别为室内和室外灯光的开关状态,液晶显示器用于显示环境状态的数据.系统的控制器选用ATMEGA16单片机.

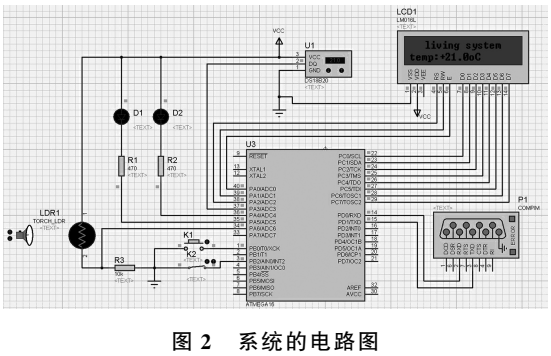

图2 系统的电路图

由于 Proteus 与 Unity3D 的数据通信需要运行一个虚拟串口，因此本文使用 Virtual Serial Port Driver 软件创建了 2 个虚拟串口 COM1（Proteus 端使用）和 COM2（Unity3D 程序端使用）.

2.3　控制系统的软件设计

1）主程序的设计. 采用 C51 编程语言开发单片机控制系统的主程序，主程序流程图如图 3 所示. 流程中，系统初始化主要是对串口、中断系统、AD 转换器和液晶显示器进行初始化. 完成初始化设置后，系统会首先依次执行非法进入检测模块和按键查询模块. 若有响应则系统执行对应的处理子程序，若无响应则顺序执行 A/D 转换、温度数据采集、串口数据处理和液晶显示等子程序，并反复循环上述操作.

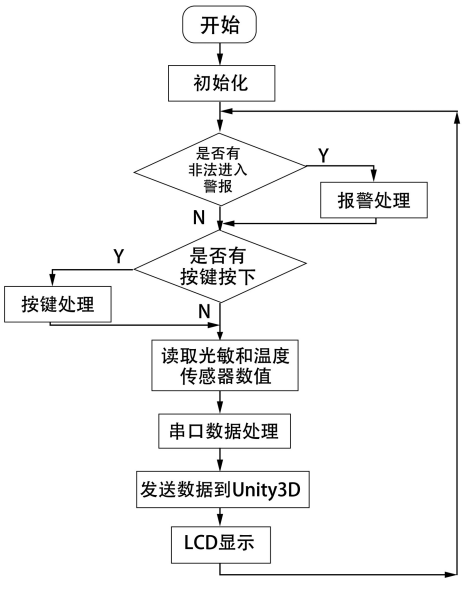

图 3　主程序流程图

2）通信数据格式. Proteus 端与 Untiy3D 端的通信数据主要包括报警状态数值、按键状态数值、光敏数值、温度数值等，但由于所采集的数据类型不同，所以在发送前需将数据转换成统一的格式. 转换方法是：将报警状态和按键状态的布尔类型数值用整数 0 和 1 表示，将光敏和温度的浮点型数值放大至 10 倍并转换成整型数值，这样所有的状态数据就均能够以整型数据进行发送.

为确保通信的可靠性，需对数据的完整性进行验证，其方法为：在通信协议中定义起始符号“%”和停止符号“#”，不同状态值中间用分割符号“|”间隔，有效数据必须含有起始符和停止符[7]. 数据格式为“%报警状态|按键键值|光敏数值|温度数值#”，数据长度固定为 9 个字节.

2.4　Unity3D 串口通信

1）串口通信的调用. 使用 C# 编程语言的 SerialPort 类实现串口编程，在编写程序前需先将 Configuration 中的 Api Compatibility Level 设置为. NET2.0（避免在引用 C# 中的 SerialPort 类时发生编译错误）. 由于 Unity3D 读取串口数据时需要一定的时间（端口设备的反应时间），此时若将串口通信的调用程序直接放在 Update() 主循环中就会过多消耗系统的资源，影响程序运行的流畅性. 为此，本文创建了一个线程用于接收数据，即在 Update() 中对串口接收的数据进行解析. 串口调用的设置步骤如下：

首先在程序开头添加串口和线程命名空间的引用：

```
using System.IO.Ports;//引用串口
using System.Threading;//引用线程
```

然后在程序中定义 1 个串口变量 port、1 个线程变量 dataRcvT 和 4 个接收串口数据的整型变量：

```
private SerialPort port;//定义 1 个串口
private Thread dataRcvT;//定义 1 个线程变量
private int alert=0;//无报警
private int button=0;//无按键按下
private int ldr=0;//光敏数值
private int temp=0;//温度值
```

2）串口初始化设置. 串口初始化设置在 Star()函数中完成. 首先选择串口的端口和设定波特率、通信协议，然后创建一个接收串口数据的线程，最后启动线程执行串口数据，并接收处理的子程序. 相关代码如下：

```
port=new System.IO.Ports.SerialPort("COM2",9600,System.IO.Ports.Parity.None,8,System.IO.Ports.StopBits.One);//初始化串口
port.ReadTimeout=10000;//设置超时时间
  if (! port.IsOpen)
    port.Open();//打开串口
dataRcvT = new Thread(new ThreadStart(DataRcvF)); //创建接收数据线程，DataRcvF 为串口数据接收处理的子程序
dataRcvT.Start();//启动线程
```

3)串口数据的接收与处理.串口数据接收与处理的子程序流程图如图4所示.首先定义一个与接收缓冲区大小一致的动态数组,用以读取接收缓冲区的数据;然后再定义一个byte字节类型的List〈〉数据接口,用以接收去掉起始符和结束符的有效数据.相关代码如下:

```
byte[] readBuffer = new byte[port.ReadBufferSize+1];//创建接收数组
List〈byte〉 listRcvData = new List〈byte〉();//创建接收有效数据的数据接口
```

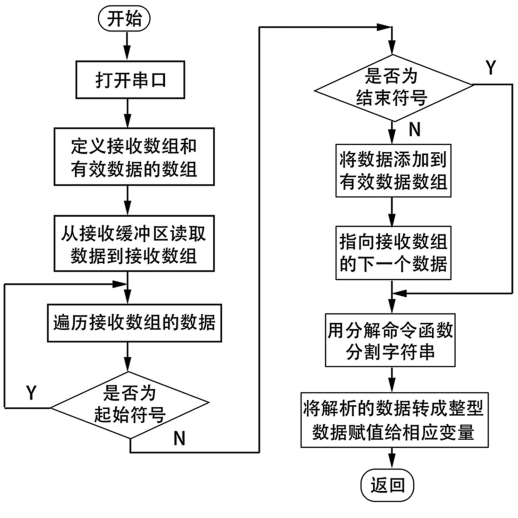

图4 串口数据接收与处理的子程序流程图

串口数据接收与处理的步骤为:①将串口接收缓冲区的数据读取到接收数组中,然后遍历接收数组.检测到起始符号%时,使用listRcvData.Add(readBuffer[i])方法将起始符号后面的数据添加到listRcvData数据中,直至检测到结束符号#.②使用Encoding.ASCII.GetString方法将接收到的有效数据转换成字符串.③用Split方法解析分割符号外的环境变量并赋值.相关代码如下:

```
string S = Encoding.ASCII.GetString(listRcvData.ToArray(),0,listRcvData.Count-1);//数据格式转换为字符串
//调用分解命令函数分割字符串,符合"|"为分割符号
string[] sArray=S.Split('|');
//将解析的数据转成整型数据并赋值其相应的变量
alert=int.Parse(sArray[0]);
button=int.Parse(sArray[1]);
ldr=int.Parse(sArray[2]);
temp=int.Parse(sArray[3]);
```

4)串口数据的发送.Unity3D通过串口向Proteus发送灯光控制命令,每次发送的数据固定为2个字节,其中第1个字节为灯光的编号(0表示室内,1表示室外),第2个字节为状态编号(0表示关灯,1表示开灯).Proteus上的单片机应用系统使用串口中断方式接收来自Unity3D的控制命令,并通过解析数据对相应的灯光进行开关控制.数据发送的子程序如下:

```
public void SendData(byte[] data)
{
  if (port.IsOpen)
  {
    port.Write(data,0,2);
  }
}
```

2.5 人机交互设计

人机交互设计的主要内容包括窗帘的开关动画、灯光的亮度控制、图形界面的设计等.

1)动画的设置.制作动画时,使用Unity3D中的预制体制作窗帘的模型,并在模型上添加Animator组件和Animator Controller;动画的开窗帘和关窗帘片段通过设置关键帧完成,动画的播放使用动画状态机控制.动画状态机的设置页面如图5所示.系统软件运行后,动画状态机首先从Entry进入到与之连接的空状态New State(等待控制信号),当收到控制信号后开启开窗帘动画open或关窗帘动画close的功能.开关窗帘的动画切换由状态参数CTR进行控制,CTR的数值由光敏传感器数值和窗帘开关键(button)的键值共同决定.当button为0时,窗帘关闭;当button为1时,CTR的数值由光敏传感器数值决定.若光敏数值大于80,$CTR=1$,播放开窗帘动画;若光敏数值小于等于80,$CTR=0$,播放关窗帘的动画.相关代码如下:

```
//定义1个动画组件变量,通过查找对象和组件的方法赋值
Animator ani = GameObject.Find("chuanglian").GetComponent〈Animator〉();
```

ani.SetInteger（"CTR"，button&（ldr－80））;//对动画组件的整型变量 CTR 赋值；

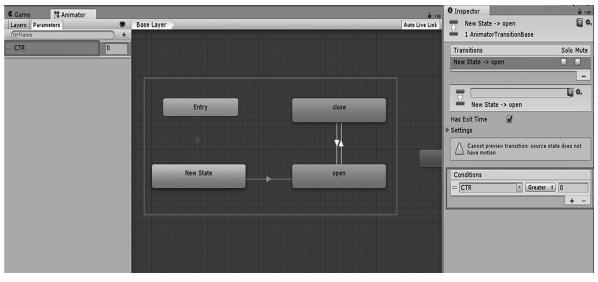

图 5　动画状态机的设置界面

2）灯光的控制. 控制灯光时首先获取灯光的组件属性，然后将串口接收的环境变量（ldr 值）赋值给灯光组件中的 range（以此改变 range 的属性值），由此场景中的灯光即可根据光敏传感器的数值调整灯光的亮度. 相关代码如下：

```
GameObject light1=GameObject.Find("Spotlight");
light1.GetComponent<Light>().range=ldr;
```

3）图形界面的设计. 图形界面的设计使用 Unity3D 的图形界面系统 UGUI，使用前先添加 UI 的命名空间（using UnityEngine.UI）和 UI 系统的文本控件，并定义相应的文本属性变量；然后通过查找游戏对象和组件的方法获取文本控件的文本属性，并将环境变量的数值赋值给文本属性. 相关代码如下：

```
//分别给显示光敏数值和温度数值的文本控件的文本属性赋值
GameObject.Find("Lightness").GetComponent<Text>().text="光敏数值："+ldr;
GameObject.Find("Temp").GetComponent<Text>().text="温度："+temp/10.0f+"℃";
//根据 alert 变量的值显示对应的状态
Text txt1=GameObject.Find("Text 1").GetComponent<Text>();
    if (alert == 0)
      txt1.text="门禁正常！";
    else
      txt1.text="非法闯入！";
```

为了更形象地显示温度值的变化，除了采用文本显示温度外还设置了温度进度条. 实现温度进度条显示的方法是：将进度条图片控件的类型（Image Type）设置为 Filled，并将填充方式（Fill Method）设置为 Horizontal，由此通过图片控件 fillAmount 的属性值（取值范围为[0，1]）即可控制进度条图片的显示比例. 将场景温度的显示范围设定为 －55.0～55.0 ℃，再将计算（实际的温度值加上 55 后除以 110）得到的数值赋给 fillAmount 即可实现对温度进度条的显示控制. 由于单片机系统发送的温度值是将原温度值扩大 10 倍的数值，因此在显示时需将该数值缩小至原温度值后再除以 110. 相关代码如下：

```
Image img=GameObject.Find("tempI").GetComponent<Image>();//获取图片控件
img.fillAmount=(temp+55)/1100.0f+0.5f;//给图片控件组建的 fillAmount 赋值
```

2.6　仿真应用测试

为了验证系统的双向控制功能，分别对环境变量对 Unity3D 的控制和 Unity3D 对 Proteus 平台的控制进行了仿真. 系统仿真的交互界面如图 6 所示.

图 6　系统仿真的交互界面

1）环境变量对 Unity3D 的控制仿真. 在 Proteus 平台上修改温度传感器的数值后，Unity3D 场景中的温度显示数值和进度条随之发生变化. 点击 Proteus 平台上的 K1 键（窗帘控制键）模拟室外太阳光的不同照射亮度时：当光敏数值小于等于 80 时，Unity3D 触发关窗帘动画；当光敏数值大于 80 时，Unity3D 触发开窗帘动画. 再次点击 K1 键时，Unity3D 触发关闭窗帘动画. 上述操作结果表明，环境变量的变化可以在 Unity3D 上产生相应的视觉变化，即本文方案可实现环境变量对 Unity3D 的控制仿真.

2）Unity3D 对 Proteus 平台硬件的控制仿

真.当点击Unity3D交互界面的室内灯光开关按钮时,Proteus平台上所对应的指示灯D1出现亮灭变化,由此表明Unity3D能够对硬件进行实时控制.

由以上仿真结果可知,本文系统能够实现对Unity3D与Proteus的双向控制,可以满足智能家居产品功能仿真的需求.

3 结论

研究表明,本文提出的基于Unity3D和Proteus的智能家居控制系统可实现人机交互界面三维展示的功能.该方法不仅能节约硬件电路制作和实体展厅建设的成本,还能作为智能家居控制系统前期开发的虚拟实践仿真平台,因此具有较好的应用价值.在本文研究中,由于Unity3D与单片机应用系统的通信使用的是基于串口的通信,因此存在无法存储和查看相关记录的不足.在今后的研究中,我们将在系统的功能仿真中增加网络连接功能,实现Unity3D与数据库的连接和交互,以此进一步提高本文方法的适用性.

参考文献:

[1] 赵沁平.虚拟现实综述[J].中国科学(信息科学),2009,39(1):2-46.

[2] 徐佳.智能家居虚拟仿真系统的设计与实现[D].南昌:南昌航空大学,2015:1-2.

[3] 甘晨.基于Unity的虚拟智能家居系统设计与实现[D].大连:大连理工大学,2015:1-45.

[4] 唐梦菲,陈瑶,高建民.基于Unity3D的手势感应自动开关门衣柜设计及功能模拟[J].林业工程学报,2020,29(5):172-176.

[5] 肖亮,李兰英,刘书赫,等.基于Proteus和Arduino的嵌入式系统虚拟实验仿真平台设计[J].科技创新与应用,2021,11(24):57-59.

[6] 褚原峰.基于用户体验的五金产品虚拟展示系统研究[D].广州:华南理工大学,2015:37-39.

[7] 何伟.Unity虚拟现实开发圣典[M].北京:中国铁道出版社,2016:375-376.

(上接第249页)

[2] CHEN L X, BAI W T, YAO Z Q. A secure and privacy-preserving watermark based medical image sharing method[J]. Chinese Journal of Electronics, 2020,29(5):819-825.

[3] 孙俞超,李德.基于图像归一化和NSST的鲁棒零水印算法[J].延边大学学报(自然科学版),2017,43(1):43-50.

[4] ZHANG S, GAO T G, GAO L. A novel encryption frame for medical image with watermark based on hyperchaotic system[J]. Mathematical Problems in Engineering, 2014(1):1-11.

[5] AHERRAHROU N, TAIRI H. PDE based scheme for multi-modal medical image watermarking[J]. Biomedical Engineering Online, 2015(14):1-19.

[6] 陈青,宗偲琦.用于医学图像认证的Contourlet-SVD双水印算法[J].小型微型计算机系统,2019,40(1):205-209.

[7] DO M N, VETTERLI M. The contourlet transform: an efficient directional multiresolution image representation[J]. IEEE Transactions on Image Processing, 2006,14(12):2091-2106.

[8] DO M N, VETTERLI M. Framing pyramids[J]. IEEE Transactions on Signal Processing, 2003,51(9):2329-2342.

[9] 楼偶俊.基于Contourlet域特征点的抗几何攻击水印方法[J].计算机研究与发展,2010,47(1):113-120.

[10] LOWE D G. Distinctive image features from scale-invariant keypoints[J]. International Journal of Computer Vision, 2004,2(60):91-110.

[11] 沈炎斌.基于SIFT特征点的三维医学图像零水印算法[J].贵州大学学报(自然科学版),2020,37(6):92-96.