

2020 夏季学期电子技术课程设计
体感交互式联机台球游戏操作杆的设计与制作

预习报告

薛传奕 王玺哲

2020 年 8 月 15 日

1 选题背景及课题简介

1.1 选题背景

改革开放以来，台球运动作为“非奥项目”的代表，在我国实现了蓬勃的发展。我国台球竞技体育经历了从兴起、规范化、地区化到职业化、专业化再到爆发式增长、扩张最后到国际化的发展历程，目前我国的台球运动已经形成大众-俱乐部、器材-赛事、学校的良性循环，其中，尤以高校内的流程度最为显著，大学生是从事台球运动的一个重要群体。然而，台球对场地的要求很高，高校内普遍存在台球运动资源相对缺乏的情况，难以满足学生对台球场地的需求。

这种供需不平衡的情况催生了一系列台球对战游戏，但由于台球运动自身较为复杂、操作难度较大，手机游戏或电脑游戏难以模拟真实的玩家视角及击球过程，无法带给用户沉浸式的游戏体验。为克服以上问题，我们在本次课程设计中选择设计一款体感交互式联机台球游戏及与其配套的操作杆，玩家使用操作杆可在有限的空间内模拟真实的击球动作，操作杆上的传感器捕捉玩家的运动数据与 PC 端进行交互，以获得良好的沉浸式游戏体验。

1.2 课题简介

在本次课程设计中，我们成功设计了一款体感交互式联机台球游戏操作杆，并开发出了支持该操作杆的 PC 端游戏。用户可在有限空间内使用该操作杆模拟真实的击球动作，传感器在捕获运动数据后将其发送至 PC 端，实现对 PC 端游戏中台球运动的控制。

台球杆的运动数据捕获通过六轴重力加速度陀螺仪 MPU-6050 实现，球杆的转动和出杆速度分别由 MPU-6050 测得的绕轴角速度与轴向加速度做数据处理后得到；由于 MPU-6050 的轴向加速度具有一定的不稳定性，我们又添加了压力传感器来检测用户的握杆力度，当握杆力度超过某个阈值时同样认为击球成功，提高了游戏的操作流畅度。球杆上附带心率传感器 Pulse Sensor，可以实时测得玩家的心率，并将其展现在游戏界面中。传感器捕捉的数据经数字处理器 Arduino UNO 处理后传入 PC 端，在 PC 端的配套游戏程序中完成交互过程。操作杆与 PC 端之间的信息传输通过蓝牙完成，玩家在击球时拥有更大的操作空间，增强了游戏体验的真实度。在该台球游戏的设计上，我们设计了网络传输模块，实现了玩家之间的远程联机对战，这大大突破了地域上的限制，尤其在疫情期间，玩家可以和来自天南海北的人进行切磋对战。我们还

增加了视频通话的功能，玩家可以切换模式选择与对方进行视频或者对战，使得这款游戏兼具打球与交友的双重功能。为了增强玩家的沉浸式游戏体验感，操作杆上装有 LED 灯与蜂鸣器，LED 灯的亮度会随玩家握杆力度的改变而改变，蜂鸣器会在玩家完成击球时发出声音，充分模拟实际击球时的外部环境。电源管理电路实现了 5V-4V 的 DC-DC 转换，通过 WE-BENCH 平台设计仿真，为心率测量电路供电。

2 电路框图

2.1 前期构想

整体设计的电路框图如图 1 所示。该台球对战游戏的两个参与玩家之间通过 TCP 通信协议进行信息的网络传输，每个用户所在端的内部结构相同，仅以用户 1 为例。Arduino UNO 处理器与 PC 端之间通过 URAT 进行通信，MPU-6050 通过 I2C 通信协议向 Arduino UNO 处理器传递数据，压力传感器向 Arduino UNO 处理器传递模拟量。数据经 Arduino UNO 处理器处理后向 PC 端传递，PC 端进行信息处理后回传游戏状态信息，并通过 Arduino UNO 处理器将游戏状态信息传递至执行机构 LED、蜂鸣器。Arduino UNO 处理器输出的 5V 直流电压经过电源管理电路后变为 3.3V 的稳定直流电压，用于 MPU-6050 和压力传感器的供电。

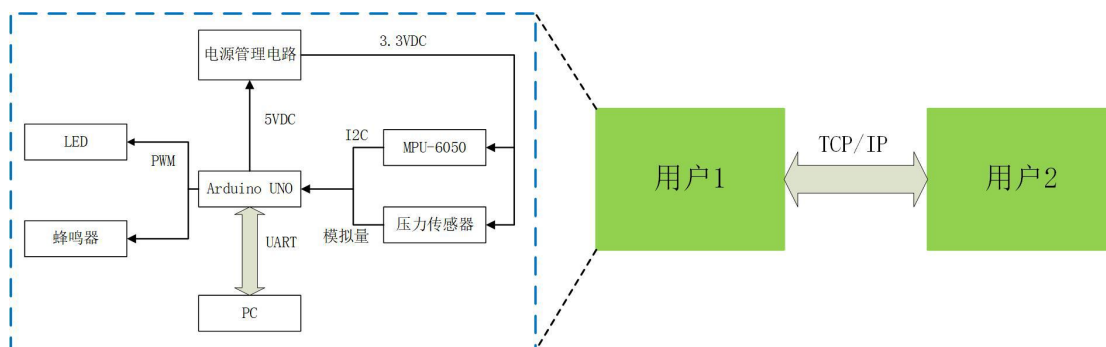


图 1 电路整体框图

3 前期方案比较

3.1 数字处理器选择

基于前期的课程积累以及资料查阅，我们首先确定可以选择的数字处理器有 FPGA、Arduino 单片机以及 STM32 单片机。

首先考虑到台球游戏的需求，将传感器读取的原始数据进行简单的处理并通过串口传输到 PC 机上，这三种数字处理器均可以很好的完成想要的效果。

其次考虑到开发的难度以及其应用的成熟度，可以看到，FPGA 使用较为底层的 Verilog 语言，通过计原课上对 FPGA 一学期的探索，可以明显的感受到无论从逻辑开发上还是从程序 debug 上，FPGA 均需要付出较大的精力；对于 Arduino 以及 STM32 单片机而言，其开发环境封装了很多有用的函数，开发难度较低且易于上手；其次，相较于底层的 Verilog，两种单片机均使用类 C 语言进行开发，这是我们十分熟悉的。由于在大二上学期电设比赛中，我们接触过 Arduino 与 STM32，根据当时的学习体验来看，Arduino 上手程度最为容易，因此我们选用了 Arduino UNO 板作为我们的数字处理器。

3.2 传感器的选择

由于制作一款体感的台球游戏杆，为了完成功能，自然需要跟踪台球杆运动的轨迹以及捕捉出球瞬间的出杆力度，最初的构想是通过角度传感器跟踪台球运动的角度，通过加速度传感器捕捉出杆瞬间的加速度以确定击球力度。

在查阅相关资料并结合京东价格等综合考量之后，我们仅仅确定了一款角度传感器：RPI-1031，其内部具有发光二极管一个，光敏接受三极管两个以及一个圆柱形遮光物所组成，其内部简单的逻辑并不能跟踪实时变化的角度，而仅仅单一输出四个方向，这显然与我们的设计初衷不符，我们被迫放弃了此角度传感器。进一步考虑精度较高的角度传感器时，发现其价格较为昂贵，于是我们放弃了使用了角度传感器的方案。

为了实时跟踪球杆运动的轨迹，我们考虑到使用陀螺仪获得绕三轴的角速度值，可以通过简单的角度积分获得角度值，进而追踪球杆的运动。结合京东的搜索，销量大多集中于 MPU 系列，通过查阅相关资料可知，MPU-6050 传感器是一个三轴加速度计与三轴陀螺仪结合的六轴传感器，这与我们前期构想利用加速度反映出杆力度恰好吻合，因此我们选用了 MPU-6050 模块。

在最初的简单测试中，我们发现 MPU-6050 模块在三轴的加速度测量上与实际的感受有所差距，因此我们决定通过按键的力度来控制出杆的力度以保证出杆力度的精准控制。于是我们购买了薄膜压力传感器（FSR-402）作为力度控制模块，在实际击球中，既可以模拟真实的出杆使用加速度作为出杆力度，又可以为了得到精准的力度控制而使用压力传感器进行控制。

3.3 交互模块的选择

考虑到手指对于压力的感受不太直观，我们选择通过控制 LED 灯的亮度来反映压力传感器接受到的压力大小。

另外，为了提醒接球者球已经击出，我们采用蜂鸣器模拟击球的声音。

3.4 游戏模式的选择

考虑到台球游戏是一款回合对战制的小游戏，只有通过丰富的回合制比赛才能体现台球游戏的魅力。因此，我们在单机游戏与联机游戏中选择了联机游戏。

由于我们距离较远不在一个局域网内且附近没有公网，于是我们利用相关的软件进行端口映射，成功建立了连接，并将传感器的数据通过网络传输，实时的进行台球比赛。

4 基于 WEBENCH 的电源电路仿真

4.1 方案设计

根据实际需求，供电采用 Arduino UNO 板的 5V 电压输出，得到可以驱动传感器工作的 3.3V 稳定直流电压。因此，我们设定输入电压范围为 4-6V，输出电压为 3.3V，最大输出电流值为 0.1A，根据 WEBENCH 平台上 Balanced 的默认排序，得到电路原理图如上图所示。可以看到，该电路选用的稳压芯片为 TPS62237，还包含一个 $2.2\mu\text{F}$ 的电容，一个 $4.7\mu\text{F}$ 的电容以及一个 $1\mu\text{H}$ 的电感。

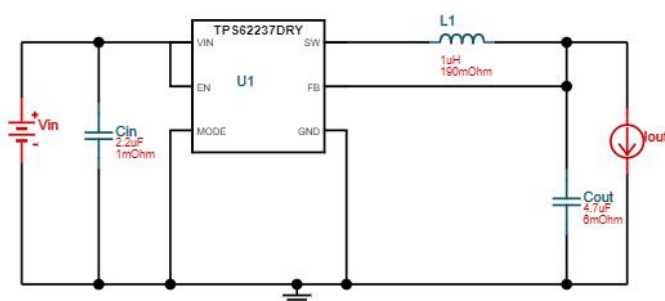


图 2 电源管理模块电路原理图

4.2 电路仿真

利用 WEBENCH 自带的仿真工具对电路进行仿真。

1) 稳压效果

从输入电容纹波电流有效值来看，随着输出电流的增大，纹波电流有效值逐渐增大，且对于电压波动纹波电流变化不大；从输出电容纹波电流有效值来看，随着输出电流的增大，纹波电流有效值逐渐增大，且对于电压波动纹波电流变化很小，可以看到，这与输入电容纹波电流的有效值很类似；从电流纹波的峰值来看，随着输出电流的增大，电流纹波的峰值逐渐减小，尤其当输入电压为 4V 时，电流纹波的峰值减小的幅度越大；从输出电压纹波峰峰值来看，随着输出电流的增大，输出电压纹波峰峰值逐渐变小，尤其当输入电压为 4V 时，其变化的最为快速，实际上，Arduino UNO 板的电压输入波动不大，考虑到 $V_{in}=5V$ ，Output Current=0.06A 时的 V_{out} P-P 为 0.018V 左右，其纹波波动影响很小，可以使得电路稳压效果达到要求。

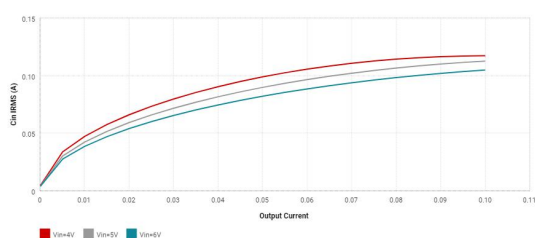


图 3 输入电容纹波电流有效值仿真

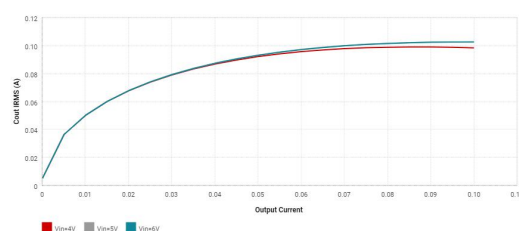


图 4 输出电容纹波电流有效值仿真

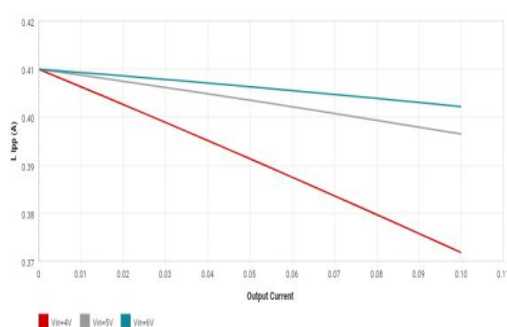


图 5 电流纹波峰值仿真

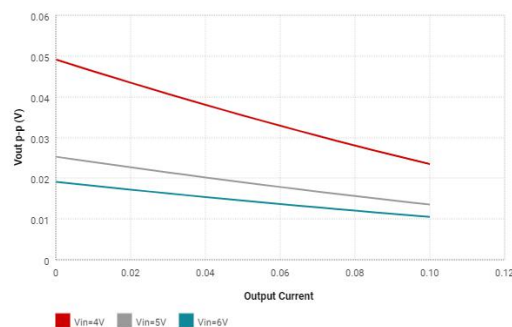


图 6 输出电压纹波峰峰值仿真

2) 功耗

从输入电容的功耗来看，当输入电压为 4V 时，其功耗最大，且随着输出电流的增大，功耗近似线性增大；从输出电容的功耗来看，当输入电压为 6V 时，其功耗最大，且随着输出电

流的增大, 功耗逐渐增大, 但当输出电流大于 0.07A 后, 输出电容的功耗增长缓慢, 近似不变; 从电感功耗来看, 输入电压大小对电感功耗影响不大, 且随着输出电流的增大, 电感功耗近似线性增大; 从芯片功耗看来, 当输入电压为 6V 时, 其功耗最大, 且随着输出电流的增大, 其功耗近似线性增大, 同时可以看到, 芯片功耗几乎比电容电感功耗大一到两个数量级, 因此电路的总功耗取决于芯片的功耗, 当 $V_{in}=5V$, Output Current=0.06A 时, 电路的总功耗近似为 0.03W, 功耗较小, 满足设计要求。

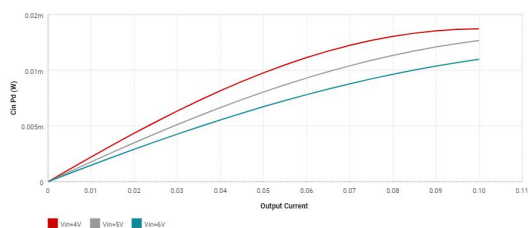


图 7 输入电容功耗仿真

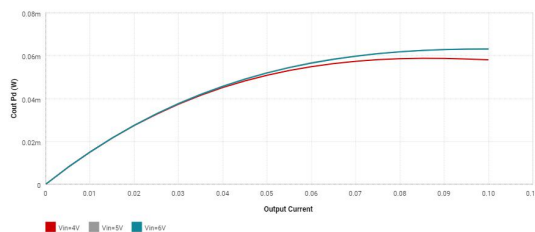


图 8 输出电容功耗仿真

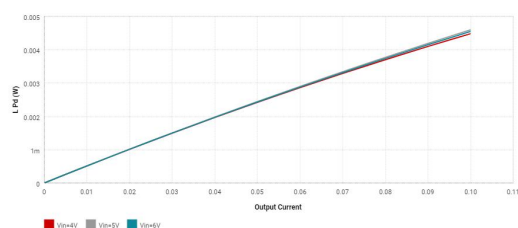


图 9 电感功耗仿真

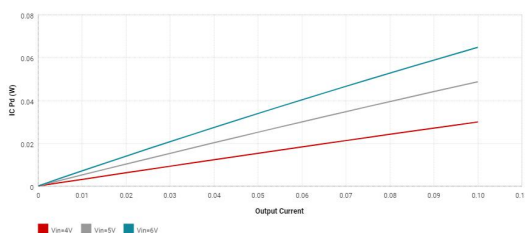


图 10 芯片功耗仿真

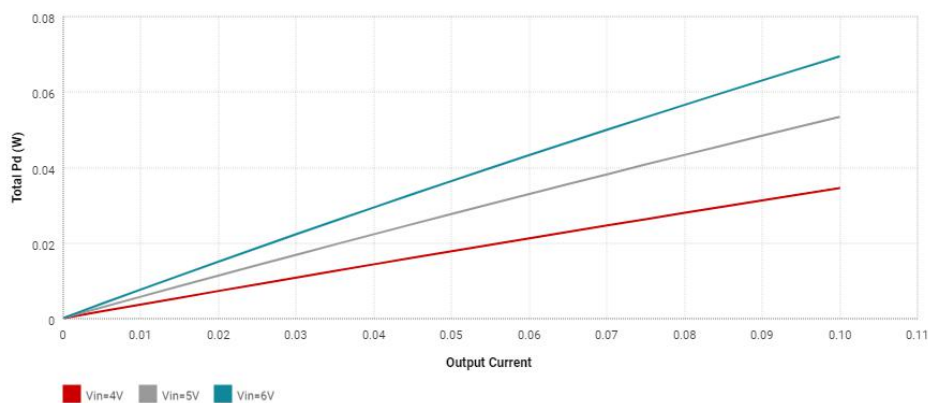


图 11 电路总功耗仿真

3) 效率

从图中可以看到, 当 $V_{in}=5V$ 时, 输出电流大于 0.01A 后, 电路的效率近似为 85%, 可见

此电路的效率较高；进一步将横坐标取对数，可以看到当输出电流从 0.1mA 上升到 1mA 时，电路的效率增加最快，当输出电流大于 1mA 后，电路的效率近似保持稳定，且在一个较高的水平。

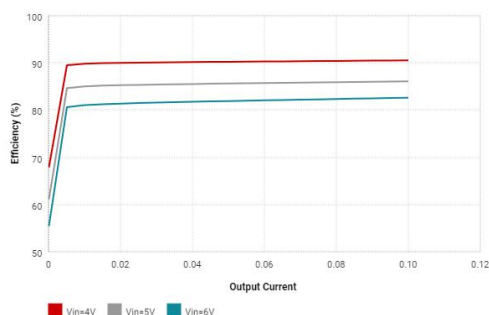


图 12 电路效率与输出电流仿真

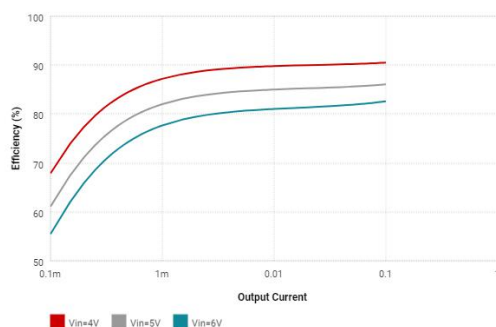


图 13 图 12 横坐标对数作图

4) 动态仿真

从启动仿真来看，输出电压的建立时间大于为 0.15ms，满足电路应用要求；从负载动态响应仿真来看，负载电流大幅度变化时，输出电压变化仅仅不到 20mV，稳压效果显著，可以满足电路设计要求；从输入动态响应仿真来看，当输入电压在 4 到 6V 之间跳变时（实际上 Arduino UNO 板 5V 输出波动达不到这个值），输出电压振荡很快，其变化同时仅仅不到 20mV，稳压效果显著，可以满足传感器输入电压的要求；从稳态仿真来看，输出电压在 3.295V 到 3.310V 之间变化，电压波动很小，可以满足传感器 3.3V 的供电要求。

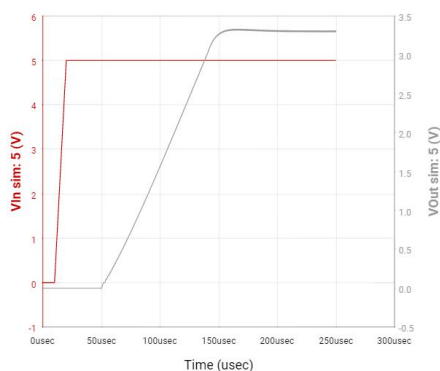


图 14 启动仿真

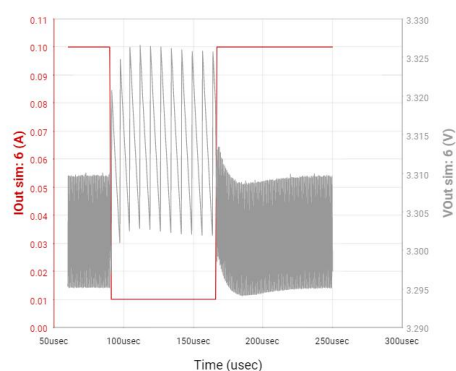


图 15 负载动态响应仿真

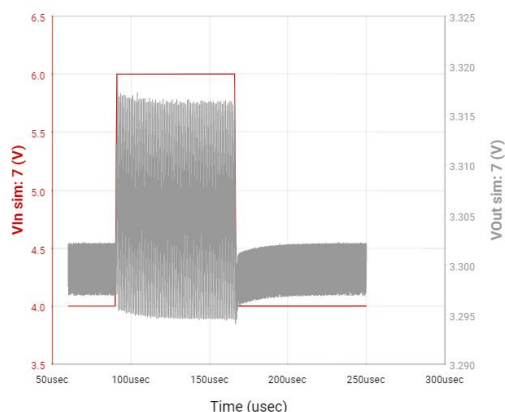


图 16 输入动态响应仿真

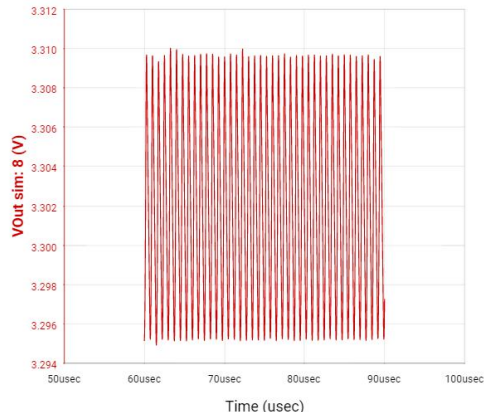


图 17 稳态仿真

5 数字系统流程图

数字系统流程图如图 18 所示。Arduino UNO 处理器的程序流程为：初始化串口、设置引脚和波特率，然后进入主循环，首先接收 PC 端传递的蓝牙信号，并根据该蓝牙信号控制蜂鸣器，读取 MPU-6050、压力传感器、心率传感器的测量值，将数据写入串口向 PC 端发送。Arduino UNO 处理器根据压力传感器数值向 LED 灯传递数据，以控制 LED 灯的亮度，延时 0.1s 后，返回主循环开始。

PC 端程序分为服务端与客户端，其中客户端与服务端程序仅在部分操作步骤上存在顺序差异，在图 2 中有相应体现，下以服务端为例说明 PC 端程序的流程：初始化网络端口，等待客户端连接，然后进入主循环，首先选择对战模式或视频模式，若选择视频模式，则向客户端发送视频模式信号，并进入视频传输循环，该循环跳出条件为在视频窗口上输入字母 q。视频模式跳出或者直接选择对战模式则等待接收客户端发送的游戏状态信息，执行运动函数。当所有球处于静止状态后等待接收客户端发送的模式信号，若为视频模式信号，则进入视频传输循环，当客户端发送结束视频信号时跳出循环。视频模式跳出或者模式信号为对战模式时接收传感器数据，当满足击球条件时向 Arduino UNO 处理器中传递击球信号，向客户端发送游戏状态信息，并执行运动函数。当所有球处于静止状态后返回主循环开始。

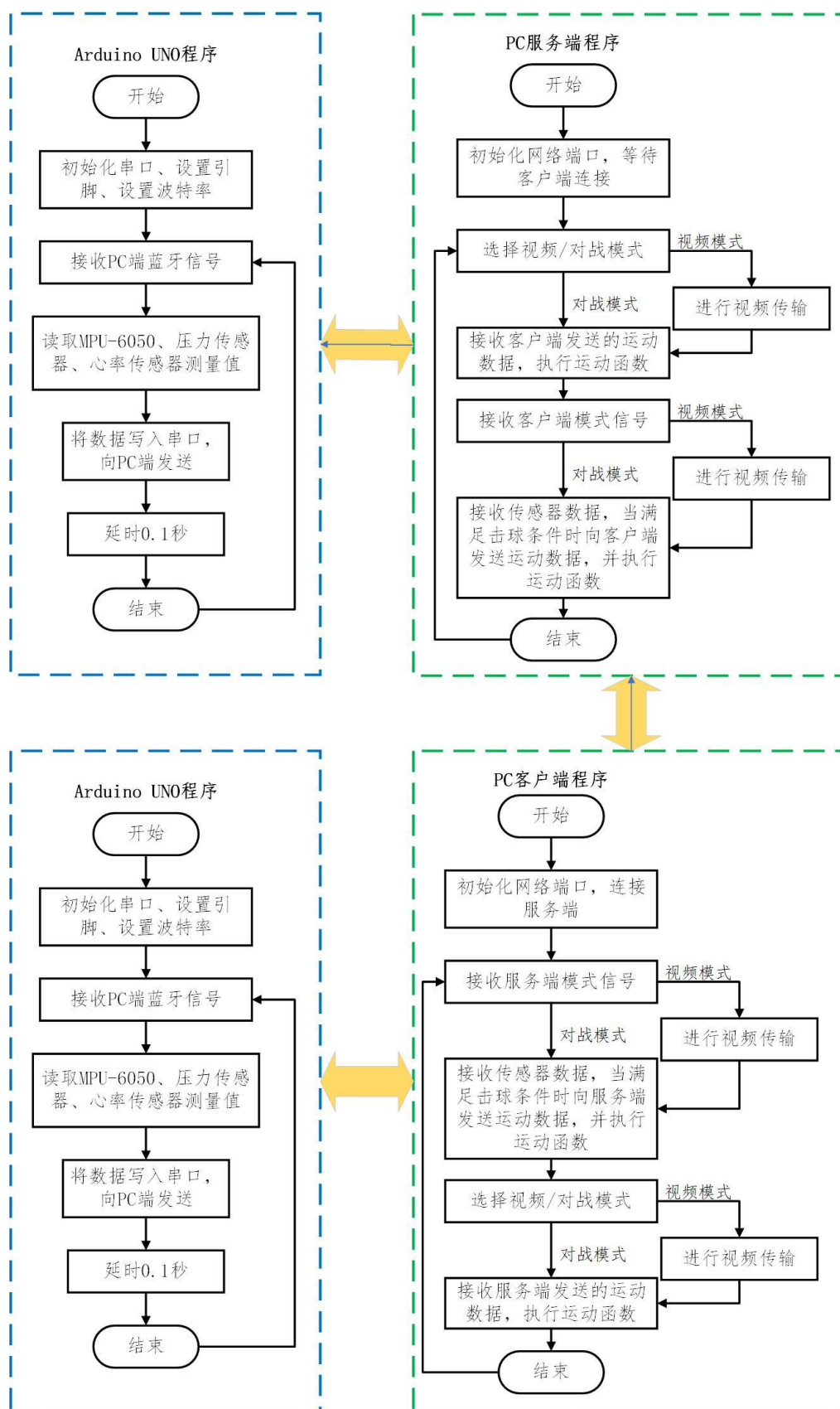


图 18 数字系统流程图