
综合论文
智能风扇

姓名: 吕光冉

班级: 自 73

学号: 2017011574

日期: 2019 年 6 月 16 日

目录

0.1 摘要	3
1 引言	3
2 设计目标	3
3 实现方式	3
4 基本电路模块电路原理	5
4.1 模拟电路模块	5
4.2 数字电路模块	9
5 模拟电路仿真结果	10
5.1 5V 稳压电路	10
5.2 正负电源电路	12
5.3 温度测量电路	14
5.4 热释电测量电路	16
6 电路搭建与实测结果	20
6.1 模拟电路模块	20
6.2 数字电路模块	24
6.3 综合调试	28
7 实验中电路出现的问题以及原因分析	29
8 项目的改进方向	31
8.1 模拟电路部分	31
8.2 数字电路部分	32

0.1 摘要

本文介绍了一个智能风扇系统的设计与实现。本设计以 Arduino Mega 为主控板, 利用基于 PT1000 的灵敏温度测量电路感知环境温度, 利用基于热释电传感探头 RE200B 的人体感知电路感知人是否在检测范围内出现, 利用这两个方面的环境信息来自动调整风扇转速。同时, 为了实现风扇的可携带性, 本设计在电源管理模块中, 使用了 5V 电压稳压可调节模块, 正负电源模块和 DC/DC 升压斩波电路等模块来配合 12V 锂电池来给电路提供电源。

关键词: 智能风扇; 小型电子系统设计; 模拟电路; 数字电路

1 引言

我所选择的第二个综合论文题目为智能风扇系统的设计与制作。

在炎炎夏日, 一个好的可携带风扇对人们就变得十分重要。在我自己使用风扇的过程中, 我发现市面上所售卖的风扇主要存在以下两个问题: 一方面, 如果忘记关闭风扇电源, 即使是在人已经离开房间以后, 它也不能自动关闭以节省电池电量, 另一方面, 大多数风扇都不具备测温功能, 只能通过用户手动根据体感温度调节风扇转速。

所以说, 我想要自己设计并实现这样一款智能风扇: 一, 它首先要体积较小, 使用电池供电, 具有可携带性; 二, 它要能够感知人体, 在人离开风扇的检测距离以后, 自动关闭风扇; 三, 它应该能够根据环境温度, 智能调节风扇转速; 四, 它最好能够实现比较好的用户互动, 有显示屏和蓝牙 APP 等方式来接收用户指令。

2 设计目标

1. 使用电池供电, 具有便携性
2. 能进行人体检测, 在人离开检测范围后, 关闭风扇
3. 能根据环境温度, 智能调节风扇转速
4. 有显示屏显示当前工作状态
5. 能通过蓝牙进行连接, 与用户手机 APP 进行连接后, 由用户调节其在不同温度下的转速

3 实现方式

整个电子系统的实现框图如图1所示:

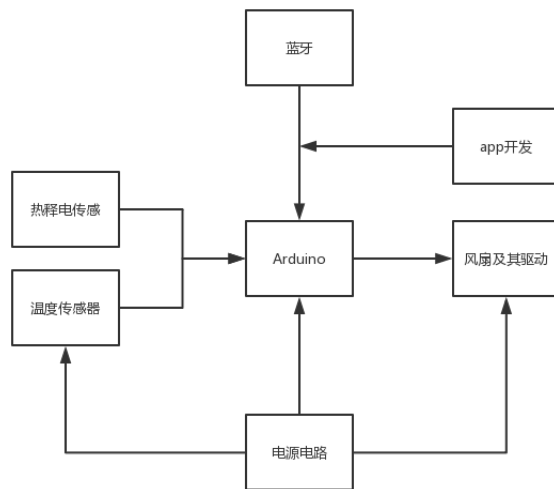


图 1: 整体电路实现框图

在这里，我将整个电子系统分为下列模块：主控板模块，温度传感器模块，热释电传感模块，风扇及其驱动模块，蓝牙模块与对应的 APP，电源电路模块。

其中，热释电传感器是用来实现人体检测的功能，温度传感器负责检测环境温度，风扇及其驱动电路负责驱动风扇运转，蓝牙及 APP 模块负责实现与用户的蓝牙连接，主控板 Arduino Mega 负责实现数字控制逻辑，即获取热释电传感和温度传感器输出的环境信息，根据用户蓝牙设置的预定义功能，自主决定风扇转速。并且，整个电路的供电问题，都由电源管理模块负责。

其中，根据传感器，主控板，风扇等模块所需要的电源类型，电源管理模块包括有：



图 2: 电源管理模块设计

而整个项目我的基本调试流程如下：

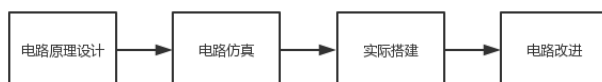


图 3: 基本调试流程

在下文中，我也将根据这一思路，阐述我的电路的设计过程，仿真结果以及实测结果。

4 基本电路模块电路原理

4.1 模拟电路模块

1. 5V 稳压电路

5V 稳压电路电路原理图：

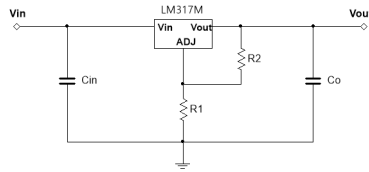


图 4: 5V 稳压电路原理图

我为了实现 5V 电压可以进行微调，使用了一个电压可调电路。其中，三端稳压器使用了 LM317M。在输入电压足够大，公共端 ADJ 处电流可以近似忽略的条件下，电路输出电压满足关系：

$$V_{out} = 1.25(1 + \frac{R_2}{R_1})V \quad (1)$$

由于使用到了三端稳压器，其脉动系数 S 应该不会太大，可以满足后级电路需求。

2. 正负电源电路

我设计的正负电源电路原理图如图5所示：

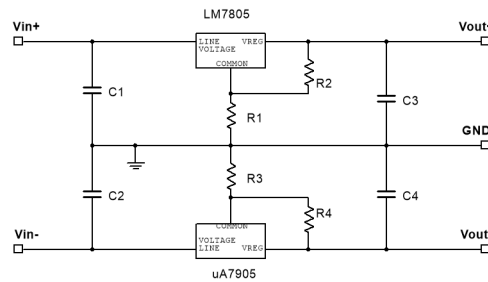


图 5: 正负电源电路

在输入电压足够大，并且 COMMON 端电流可以被近似忽略时，电路输出近似满足下列关系：

$$\begin{cases} V_{out+} = 5(1 + \frac{R_1}{R_2})V \\ V_{out-} = -5(1 + \frac{R_3}{R_4})V \end{cases} \quad (2)$$

3. 温度测量电路

温度测量电路原理图如图6所示：

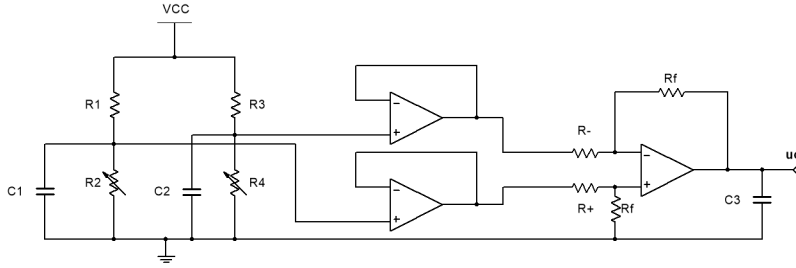


图 6: 温度测量电路原理图

温度测量电路的设计是基于铂电阻（PT1000）的温度特性。铂电阻在不同温度下的阻值不同且数值比较固定，在将铂电阻阻值变化转变为电信号后，经过一定的运算电路就可以得到比较精确的温度测量值。

在电路图中， R_2 为 PT1000 铂电阻，其阻值随温度变化， R_4 为调零电阻，可以调整电路使其在 0°C 时输出为 0。 R_1, R_2, R_3, R_4 构成电桥电路。电桥电路的两个输出值在经过由精密运放构成的跟随器之后，进入最后一级电路做差并被放大。原理图中 C_1, C_2 负责稳定输入电压，消除自激振荡， C_3 负责稳定输出电压，滤去高频噪声。

根据 PT1000 铂电阻的温度特性，在 0 摄氏度时，PT1000 电阻阻值为 $1000\ \Omega$ ，其后，每上升 1 摄氏度，其阻值大致上升 $4\ \Omega$ 。则根据 R_1, R_2 分压关系，可知在 k 摄氏度时，如果假设 R_1 足够大，则电桥左路的输出电压为：

$$V_+ = \frac{R_2}{R_2 + R_1} V_{CC} = \frac{1000 + 4k}{R_1 + 1000 + 4k} V_{CC} \approx \frac{1000 + 4k}{R_1 + 1000} V_{CC} \quad (3)$$

此时，若调节电路在 0 摄氏度时输出为 0，则电桥右路输出电压为：

$$V_- = \frac{R_4}{R_3 + R_4} V_{CC} = \frac{1000}{1000 + R_1} V_{CC} \quad (4)$$

因此，在最后一级，当正极输入电阻和输出电阻阻值相等时，有：

$$V_{out} = \frac{R_f}{R_+}(V_+ - V_-) \approx \frac{R_f}{R_+} \cdot \frac{4k}{R_1 + 1000} V_{CC} \quad (5)$$

通过最后一级的比例调整，将电压近似的转化为主控板可以直接读取的模拟电压值。

4. 热释电测量电路

热释电传感电路用于检测指定距离内是否有人存在。其基本测量原理是热释电基于热释电传感探头（三极管）的三端特性。热释电传感探头可以将吸收到的红外辐射转化为微弱的电信号，而在热释电探头上加上菲涅尔透镜则可以有效削弱环境干扰信号，保留人体热电信号，所以该传感探头可以用于测量是否有人在其探测范围内。

但是经过测量，我发现，我们使用的被动式热释电传感探头实际上并不完全可靠。当有人进入探头的探测范围（由于项目需要，我设定为 20cm）后，热释电传感探头输出电平是一个直流分量和近似为 RC 衰减分量的叠加。其中，直流分量数值不确定，会随着环境光强的变化而变化；RC 衰减分量表现为短时间内电平下降 0.2V，然后逐渐恢复至 0。

同理，在人离开探头的探测范围后，热释电传感探头输出端会有一个大约为 0.1V 左右的压升，然后电平仍然会基本类似于 RC 衰减，缓慢回到原来的电压水平。

根据热释电传感探头的工作原理，我一共设计了两版热传感电路，其工作原理都是基于检测传感器是否有电平变化来判断是否有人进出传感探头检测范围，并且最终的输出量都为一个二值数字量。

我设计的第一版热传感电路原理图如图7所示。

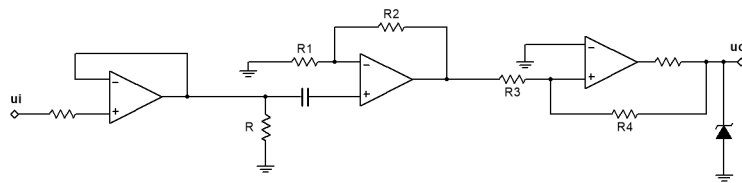


图 7: 热释电测量电路第一版

如图所示，输入信号首先经过一个跟随器以实现阻抗变换，增大输入内阻，然后输入信号通过一个一阶高通电路隔离直流分量，并适当放大变化电压幅值，最后通过滞回比较器来输出一个符合数字电压要求的电平。其中，如果输出由高电平变至低电平，则代表有人进入检测范围，如果由低电平变为高电平，则代表有人离开检测范围。

但是这一版工作电路在我实际测量时发现主要存在以下几个问题：

- 所需要的电源种类过多。原因在于电路中的运放在第一版设计中，需要正负 12V 供电。但是由于我的电路最初为 12V 直流电池供电，所以并不能直接提供这两种电平。这导致电路中需要具备相对比较复杂的正负电源电路才能解决问题。
- 电路出现误检测现象。有人缓慢离开检测范围时，热释电传感探头输出电压会出现幅值过小的问题。这时，如果仅仅是增大电压放大倍数，则会因为有效信号和环境噪声幅值相近，输出电压会多次翻转，最终电平的高低与人是否离开检测范围没有确定关系。

为了解决上述问题，我设计的第二版热释电传感电路如图8所示。

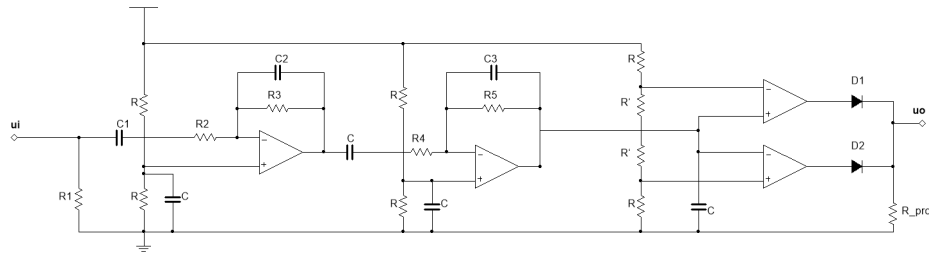


图 8: 热释电传感电路第二版

其中，R1 和 C1 构成第一级高通滤波器，负责过滤直流电平，然后输入信号经过第一级高通滤波电路进一步滤波，并通过 R2, R3 实现电压放大。之后输入信号通过 C 进一步隔离直流分量，并通过第二级电路进行输出值幅值调整并进一步滤波。随后，输入信号通过一个双限电压比较器进行电压比较，在有人进出电路能输出一个高电平。

针对第一版电路中存在的问题，我做了如下几点改进：

- 针对第一个问题，第二版电路中的运放都用 5V 和 0V 供电。并且将电路中各个基准电位用 R 和 C 控制为 2.5v。这样电路中就仅仅需要 0V 和 5V 的电源供电。
- 针对第二个问题，我将最后一级滞回比较器改为一个窗口比较器。这样做可以让电路在无人进出时输出一个低电平，只有在有人进出检测范围时输出一个高电平。虽然此时，仍然存在人缓慢离开而导致的电路误触现象，但是此时可以通过在后一级 MCU 内增加一个最小缓冲时间（即认为在输出电压变化后一定时间内，电压的再次变化为环境噪声所致，可以忽略）来抑制环境噪声。

这样做的缺点在于，电路输出值电平高低与是否有人在检测范围内并没有明确的逻辑对应关系，而仅仅能判断是否有人进出检测范围。但这样的改进结果已经优于第一版电路中，输出电平的高低和是否检测到人与人之间逻辑对应关系可能因环境噪声而错误的结果。

窗口比较器的输入电压为：

$$u_{in} = \frac{R_5}{R_4} \frac{R_3}{R_2} \Delta u_i \quad (6)$$

其中, $\frac{R_5}{R_4}$ 负责进行交流信号的放大部分, 而 $\frac{R_3}{R_2}$ 负责调整 (放大或者减小) 输入窗口比较器的输入电压以满足窗口比较器的电压比较要求。

窗口比较器的两个阈值输入电压为:

$$\begin{cases} u_l = \frac{R}{2(R+R')} \\ u_h = \frac{R+2R'}{2(R+R')} \end{cases} \quad (7)$$

5. 升压斩波电路

在我们学习过的电路中, 并没有一种模拟电路能实现 DC/DC 升压的效果。在经过调查后, 我发现, 通常使用的升压电路为升压斩波电路 (BOOST)。由于身边没有合适的电感, 所以对于这一部分电路我选择直接购买现有的模块, 而不是自己从头搭建。所以, 尽管其作为模拟电路, 其详细原理在此并不详细叙述, 仅仅假设其不仅能够有效升压, 还拥有足够的功率输出能力。

4.2 数字电路模块

1. 18650 12V 直流电源

电路的电源来源于 18650 12V 锂电池。其电池容量为 2600mAh, 瞬时最大输出功率为 60W, 经过大致估算, 从输出功率和电池容量上来说, 应该是满足我的项目需要的。

2. 12V DC 风扇

我购买的为 AFB0812M 12V 直流风扇。当供电电平为 12V 时, 其最大所需电流 (根据规格说明) 大致在 0.18A 左右。由于风扇存在机械转动, 所以风扇应该为电路功率消耗的最主要的来源。经过估算, 在使用 18650 锂电池的情况下, 风扇大致可转动 14 小时, 所以从功率消耗上是满足要求的。

3. L298N 电机驱动模块

为了实现风扇转速可调, 我使用了一个 L298N 电机驱动模块来实现 pwm 波控制风扇转速的功能。其原理是通过 H 桥电路, 将主控板输出的幅值为 5V 的 pwm 波调制为幅值为 12V 的 pwm 波并输出给风扇, 对于直流风扇而言, 在输入的 pwm 波占空比不同时, 相当于在风扇两端加不同电平的直流电压, 从而实现风扇转速可调的功能。

4. Arduino Mega 控制板

我使用的主控板为 Arduino Mega2560, 其外设包括:

- 数字量输入输出，其电平变化范围为 0-5V
- 模拟量输入，电平检测范围为 0-5V
- pwm 输出，可以输出幅值为 0-5V，占空比不同的 pwm 波
- 中断输入，可以用于检测热释电传感器输出值变化
- 5 个串行输入输出，可以用于实现蓝牙调试和连接
- IIC 总线，可以用于与 LCD 显示屏之间的数据通信

Arduino Mega 上的外设是满足我的项目需求的。

5. LCD 显示屏

我使用的显示屏为 LCD1602 显示屏。其使用方式比较简单。它通过 IIC 总线和主控板之间进行数据传输。主控板通过 IIC 总线来写入 LCD1602 上一块 RAM 内存，然后由显示屏根据其上 RAM 的内容来显示对应信息。

在本项目中，我使用 LCD 显示屏来实现来显示实时温度测量结果和是否有人进出热释电传感器检测范围的判断结果。

6. 蓝牙模块

蓝牙模块原理十分简单，其基本原理基于串口通信，在进行波特率设置和通信协议设置后，可以将其直接视为普通串口通信来使用。

而对应的 APP 可以使用上个暑季小学期所掌握的 QT5 开发框架来实现移动端的程序编写。

5 模拟电路仿真结果

5.1 5V 稳压电路

仿真电路如图9所示：

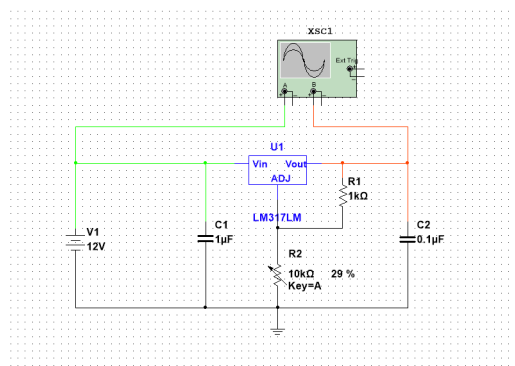


图 9: 5V 稳压仿真电路

这里我使用了一个 $10k\Omega$ 的滑动变阻器来调节电压关系。最终，在 $R_2 = 2.9k\Omega$ 时，电路的输出波形如图10所示：

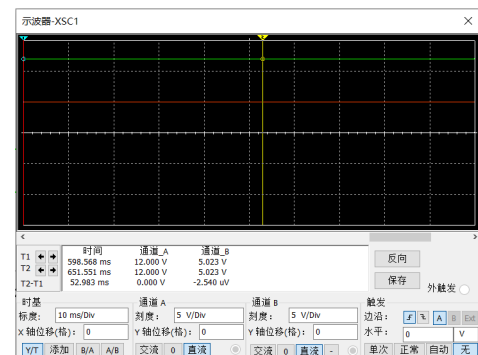


图 10: 5V 稳压电路输出波形

可以看出，在输入电压基本不变时，输出电压比较稳定。

该模块的传输特性曲线如图11所示：

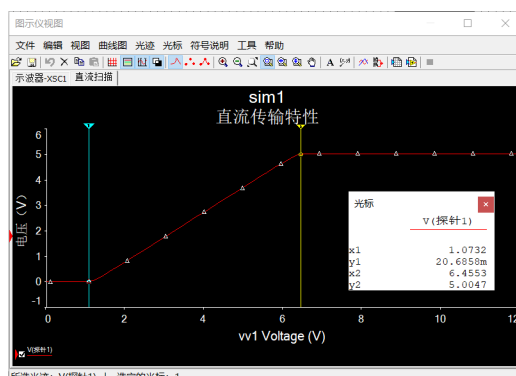


图 11: 5V 稳压仿真电路传输特性曲线

从图中可见，只有当输入电压大于 $6.45V$ 时，输出电压才能稳定在 $5V$ 不变。

在输入电压为 $12V$ 时，电路的输出电压随后一级电路输入电阻的变化曲线如图12所示。

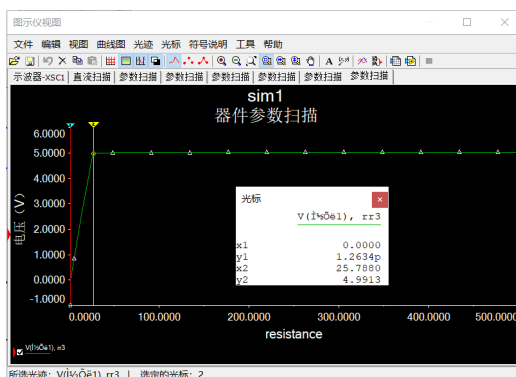


图 12: 5V 输出电压受后一级输入电阻的影响

可以看出，只有当后一级电路的等效输入电阻大于 25Ω 时，输出电压才能稳定在 $5V$ 。电路带载能力较强。

5.2 正负电源电路

正负电源仿真电路如图13所示：

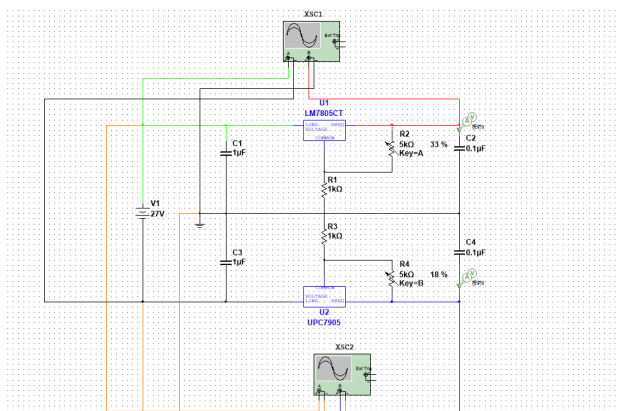


图 13: 正负电源电路

当 $R_2 = 1.65k\Omega$, $R_4 = 0.9k\Omega$ 时，电路的正负电源输出如图14,15所示：

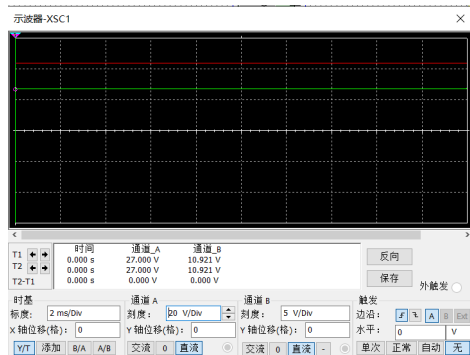


图 14: 正电源输出波形

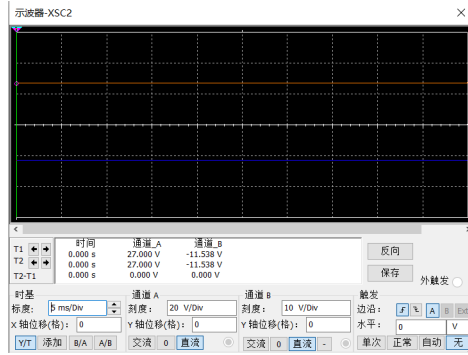


图 15: 负电源输出波形

此时，正电源 $V_+ \approx 10.9V$ ，负电源 $V_- \approx -11.6V$ 。已经满足了后续运放的电压放大需求。
此时，电路的传输特性曲线如图16所示：

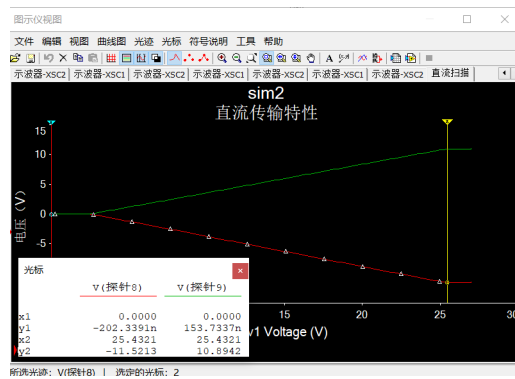


图 16: 电路传输特性曲线

从传输特性曲线中可以看出，要想让电路正常工作，需要保证前一级电路，即升压斩波电路输出电压大于 25V。

而后一级电路的等效输入电阻对于输出电压的影响曲线如图17,18所示：

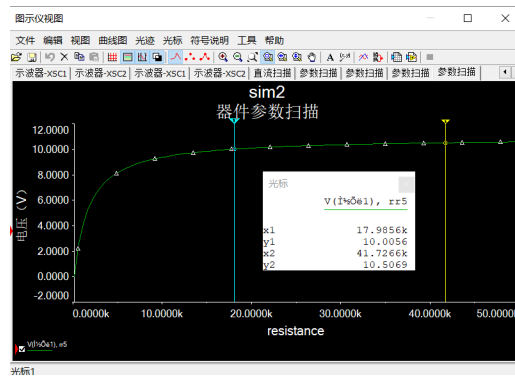


图 17: 后一级电路输入电阻对于正电源影响

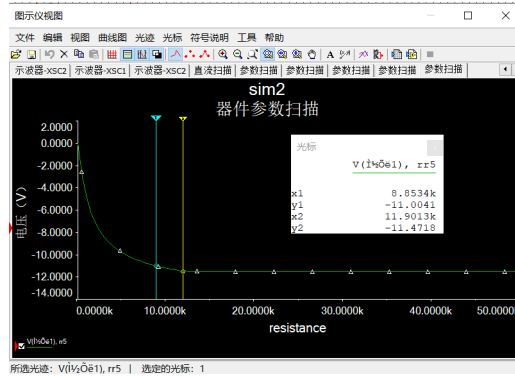


图 18: 后一级电路输入电阻对于负电源影响

可以看出，为获得期望的输出电压，正电源需要保证后一级电路输入电阻大于 $41k\Omega$ ，负电源需要保证输入电阻大于 $11.9k\Omega$ 。这说明电路带载能力较差。在实际搭建过程中，可以考虑用由功放构成的跟随器作为缓冲级电路以增强电路带载能力。

5.3 温度测量电路

温度测量电路的仿真电路如图19所示：

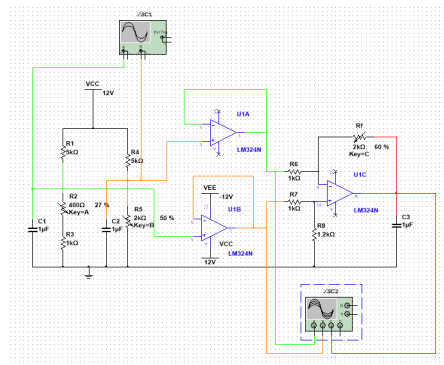


图 19: 温度测量仿真电路

在这里，我使用了一个 $1k$ 的电阻和一个可滑动的，阻值为 400Ω 的电阻串联模拟铂电阻 PT1000。每当该滑动变阻器阻值上升百分之一，就可以近似的看成环境温度上升 1 摄氏度。这意味着该滑动变阻器的百分比可以直接被近似看做环境温度。

当 R_2 滑动变阻器的百分比为 27%（模拟环境温度 27 摄氏度），电桥两端的输出电压如图20所示：

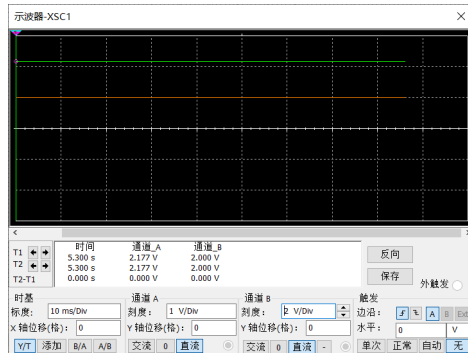


图 20: 电桥输出电压

测量结果与近似计算式 (3) 的计算结果基本一致。

最终输出电压如图21所示。

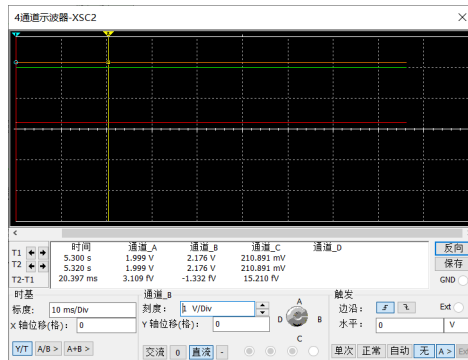


图 21: 电路输出电压

此时，在实际搭建电路，调整电路参数时，可以通过 $\frac{R_L}{R_+}$ 来调节输出电压的变化范围，以满足主控板 Arduino Mega 上 DAC 芯片的电平转换范围。

电路输出电压随温度（ R_2 阻值）的变化特性如图22所示：

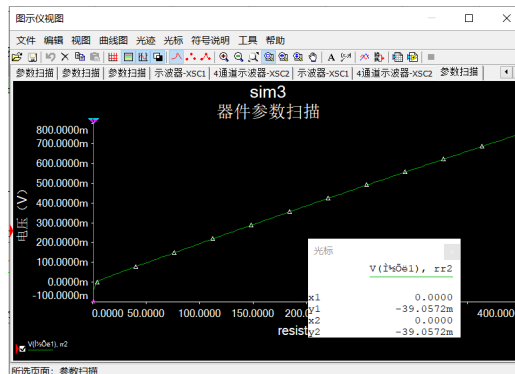


图 22: 输出电压随温度变化的特性曲线

可以看出，此时电路有足够好的线性性质，可以用于简单的温度测量。

5.4 热释电测量电路

第二版热释电测量电路的仿真电路如图23所示:

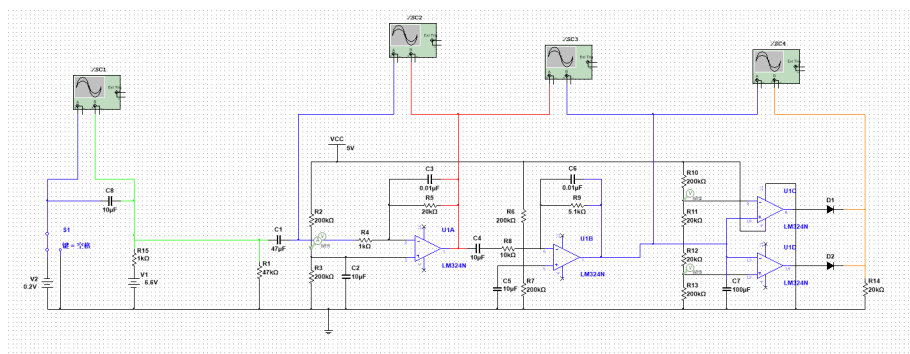


图 23: 热释电测量仿真电路

(模拟) 传感器输入波形如图24,25所示:

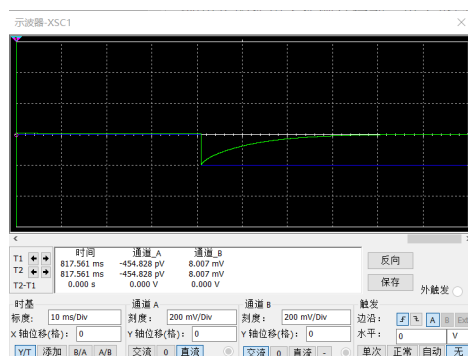


图 24: 传感器输入波形-有人进入

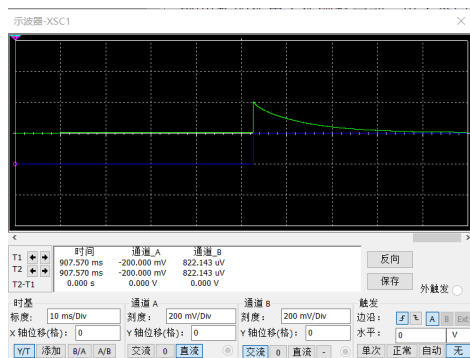


图 25: 传感器输入波形-有人离开

在两种不同的情况下，第一级电路的输出波形分别如图26,27所示：

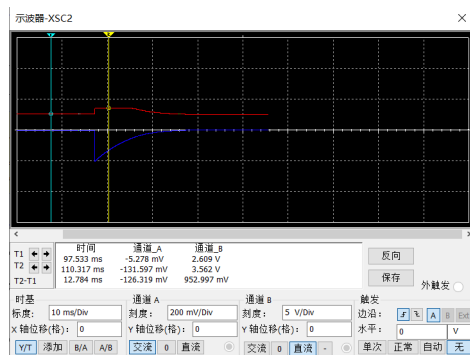


图 26: 第一级输出波形-有人进入

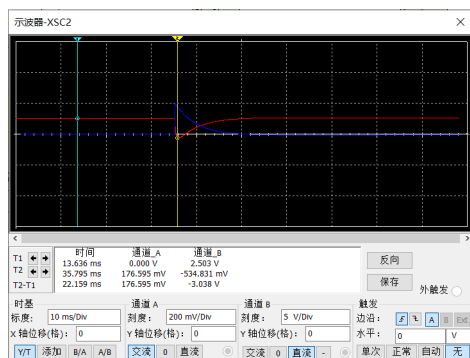


图 27: 第一级输出波形-有人离开

可以看出，此时第一级电路输出的最高电平为 3.5V，输出的最低电平为-0.5V。可以被第二级电路有效识别。

此时第二级电路输出波形如图28,29所示：

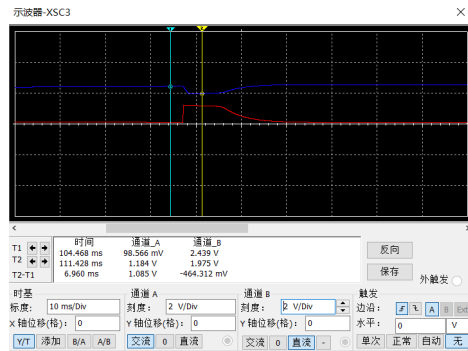


图 28: 第二级输出波形-有人进入

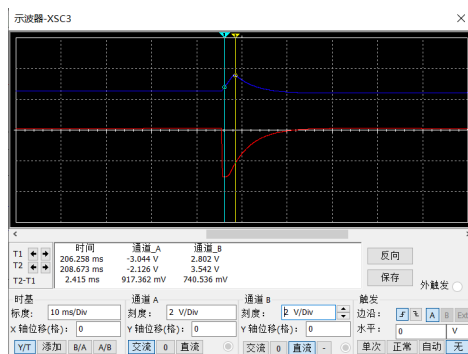


图 29: 第二级输出波形-有人离开

此时，第二级电路经过一定比例的电平缩小以后，其输出的最高电平为 3.5V，输出的最低电平为 1.98V。可以被后一级窗口比较器识别。

最后一级电路的输出波形如图30,31所示：

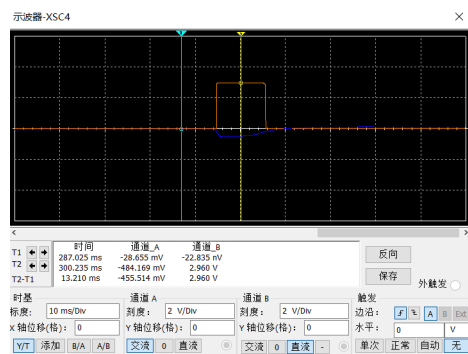


图 30: 最终输出波形-有人进入

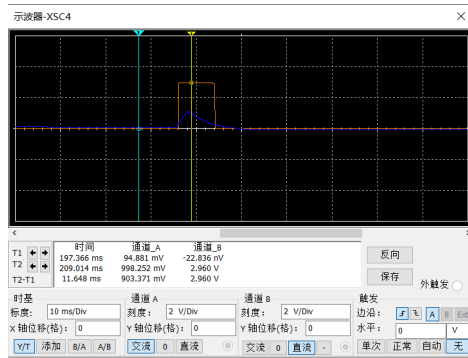


图 31: 最终输出波形-有人离开

可以看出，在有人进出检测范围时，最后一级电路都可以正常输出一个矩形波。
对于电路的响应恢复速度，测量结果如图32,33所示：

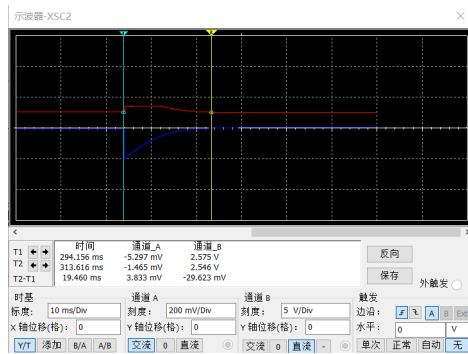


图 32: 电路响应时间-下降

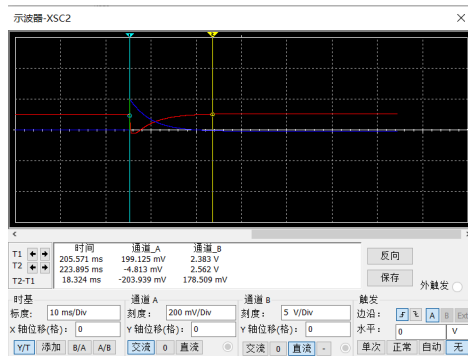


图 33: 电路响应时间-上升

可见，不论是人进入检测范围还是人离开检测范围，系统的响应恢复时间都在 20mS 以内。这说明在现实世界中，电路是可以及时为下一次人的动作变化做好准备的。

至此，所有由自己搭建的模拟电路模块仿真完成。

6 电路搭建与实测结果

6.1 模拟电路模块

1. 5V 电源模块：

我最终搭建的 5V 电源模块如图34所示：

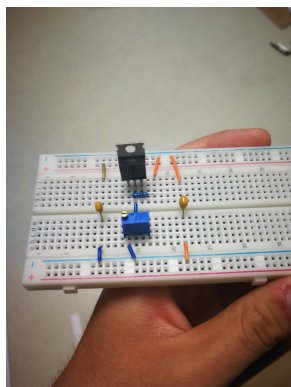


图 34: 5V 电源模块

经过调整，其最终的输出电平波形如图所示：

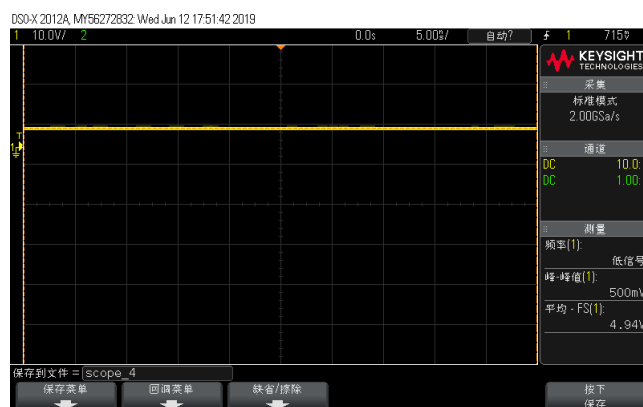


图 35: 5V 电源模块输出波形

最终输出电压为 4.94V，近似为 5V，已经满足下一级电路的电平需求，同时电压上下波动幅度大约只有 0.1-0.2V，电路稳压性能较好。

2. 正负电源模块

我最终搭建的正负电源模块如图36所示：

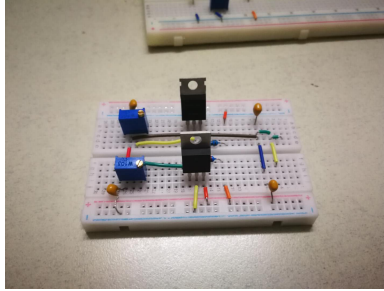


图 36: 正负电源模块

在输入电压大于 26V 时，输出电压波形如图所示：



图 37: 正电压输出波形

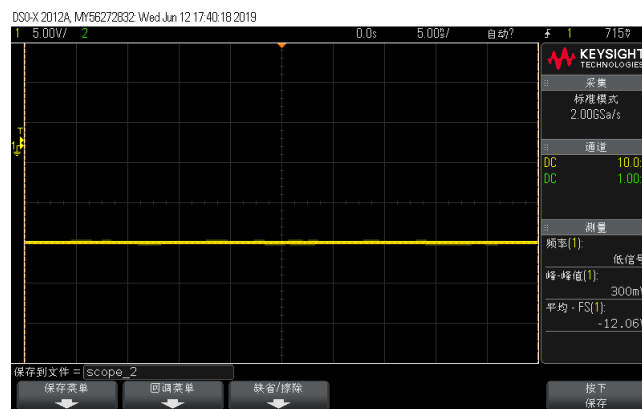


图 38: 负电压输出波形

最终输出的正电压为 12.02V，负电压为-12.06V，同时，电压稳定性能较好。但是如果直接在后一级加入电阻，则发现输出电平会降低，这说明电路的带载能力较弱。在引入功放构成的跟随器做缓冲级后，问题得到了解决。

3. 温度测量模块

我搭建的温度测量模块如图39所示：

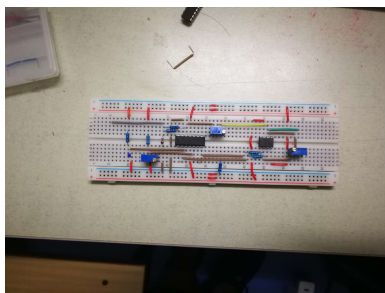


图 39: 温度测量模块

其中，电压跟随器我仍然使用了 LM324，但是运算电路部分使用了一个 OPA277 精密运放来提升电路运算精度。同时，我引入了多个可调变阻器来快速调节电路参数。

最终，测得环境温度（用手持红外成像仪测量）和输出电压之间的关系如下表所示：

温度 $T/^{\circ}C$	输出电压 V_{out}/V
26	0.261
29	0.279
31	0.299

表 1: 温度和输出电压之间的关系

由于测试条件所限，只能测得这 3 组比较有代表性的输入。根据验算可知，这三点基本在一条直线上，由于数据量较少，所以在此并不做数据拟合。

对于 Arduino Mega 主控板的 ADC 芯片，通过调整前一级电压放大倍数，环境温度可以通过公式8近似计算：

$$temp = (read_{in})/2 \quad (8)$$

其中，read_in 为 ADC 芯片所读取值（变化范围为 0-1023）。虽然这样一来温度测量精度只有 0.5 摄氏度，但是对于相对简单的风扇调节应用来说已经满足其基本需求。

4. 热释电传感模块

我搭建的热释电传感模块如图40,41所示：

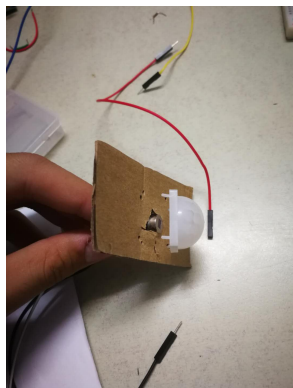


图 40: 自制热释电传感探头

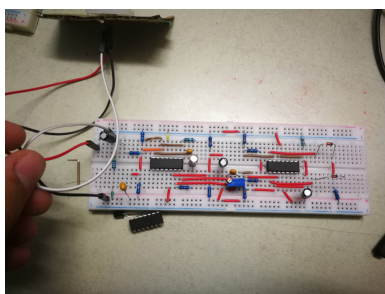


图 41: 热释电传感实验电路

其中，图40为我自己用硬纸板和绝缘胶带制作的带有菲涅尔滤镜的热释电探头，图41为我搭建的热释电实验电路。

最终，在有人进出检测范围（实际测量大约为 20cm）时，电路输出波形如图42所示：

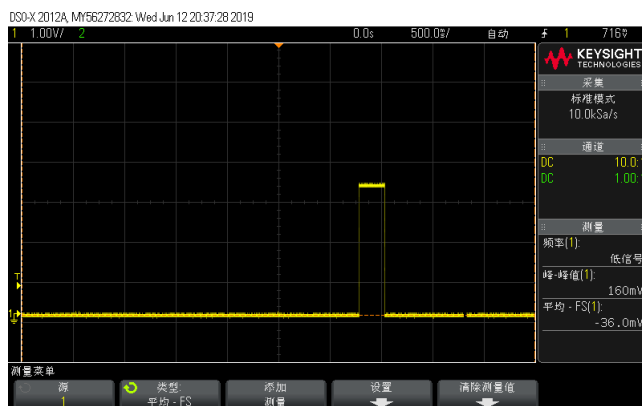


图 42: 最终输出波形

此时电路的输出电压能满足下一级数字电路的检测要求，但是电路的恢复时间大约为 200ms，长于仿真时测量的时间。这是因为现实世界中传感器工作状态和仿真时不同而导致的。实际传感器

输出电压恢复时间比仿真时时间要长，所以最终电路输出波形的实际恢复时间也要长于仿真时间。

5. 升压斩波电路

我使用的升压斩波电路模块如图43所示：

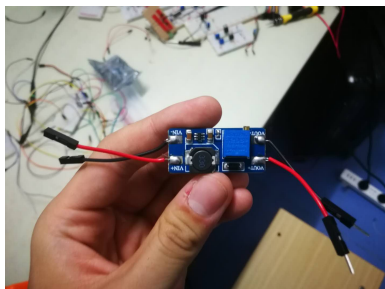


图 43: BOOST 电路模块

其输出电平波形如图44所示：

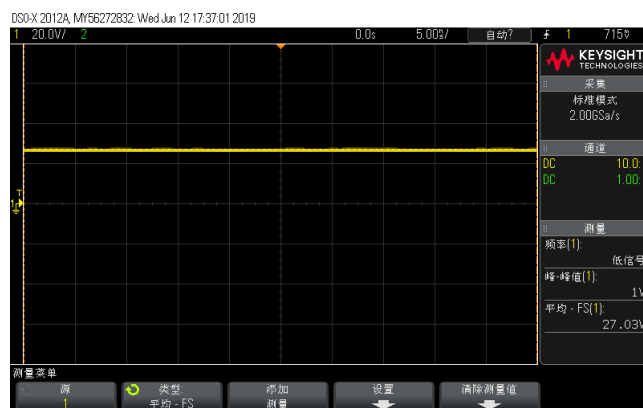


图 44: BOOST 电路模块输出波形

其输出电平为 27.03V，满足后级电路要求，且经过测试发现其带载能力较强。

6.2 数字电路模块

1. Arduino Mega 开发板

我使用的 Mega 板如图45所示：

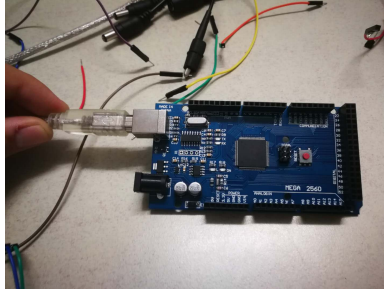


图 45: Arduino Mega 开发板

其主控代码和连线如下：

```
1      /* main.ino */
2      #include <stdlib.h>
3      #include <Wire.h>
4      #include <LiquidCrystal_I2C.h>
5
6      // set the LCD address to 0x20 for a 16 chars and 2 line display
7      LiquidCrystal_I2C lcd(0x20,16,2);
8
9      //pin
10     int sensor = A5 ;
11     int peopel_pin = 2;
12     int fan_pin = 3;
13
14     float temprature;
15     float tmp;
16     bool has_people = false ;
17     unsigned long last_time = 0 ;
18
19     void setup()
20     {
21         attachInterrupt(0,toggle_people,RISING);
22         pinMode(fan_pin,OUTPUT);
23
24         lcd.init();                // initialize the lcd
25         lcd.backlight();
26
27         lcd.home();
28         lcd.clear();
29
30         lcd.print("init complete");
31         delay(2000);
32         lcd.clear();
33     }
```

```

34
35 void loop()
36 {
37     tmp = analogRead(sensor) ;    //read temprature
38     temprature = tmp/2;
39
40     lcd.clear();                  //begin show
41     lcd.setCursor(0,0);
42     lcd.print("temp:");
43     lcd.print(temprature);
44     lcd.setCursor(0,1);
45
46     if(has_people)
47     {
48         lcd.print("has people");
49         if(temprature > 30)
50         {
51             analogWrite(fan_pin,255); //control fan to max speed
52         }
53         else
54         {
55             analogWrite(fan_pin,180); //control fan to lower speed
56         }
57     }
58     else
59     {
60         lcd.print("no people");
61
62         analogWrite(fan_pin,0);    //shutdown fan
63     }
64
65     delay(1000);
66 }
67
68 void toggle_people()              //if detect people in or out
69 {
70     if(millis()-last_time> 500)   //here set a minimum time to avoid
71     {                               //mistake , set at 500 ms
72         has_people = !has_people ;
73     }
74
75     last_time = millis();
76 }

```

需要注意的是，我在这里设置了一个 500ms 的防误触时间，也就是说，如果在 500ms 内有多次人进出检测范围的情况，则主控板只会检测一次。如果为了提升电路检测的实时性，可以适当减少此处的防误触时间。

2. LCD 显示屏

我使用的 LCD 显示屏如图46所示：

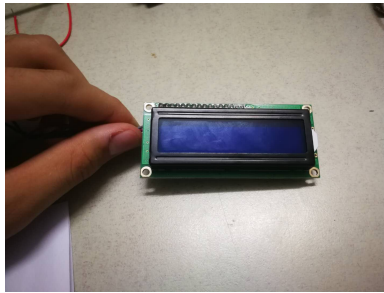


图 46: LCD 显示屏

这是一个只有 2 行，每行 16 个字符的显示屏，仅能初步满足显示需求。

3. L298N 电机驱动模块

我使用的电机驱动模块如图47所示：

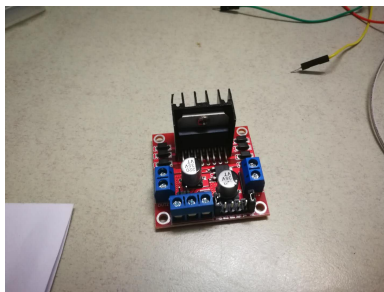


图 47: L298N 电机驱动模块

4. 12V 直流风扇

我使用的 12V 直流风扇如图48所示：



图 48: 12V 直流风扇

5. 18650 锂电池

我使用的锂电池如图49所示：



图 49: 18650 锂电池

后来由于时间问题，我并没有如预期一样开发一个蓝牙 APP，所以在此就并不展示我使用的蓝牙模块了。

以上模块经过测试都没有问题。

6.3 综合调试

整体调试电路如图50所示：

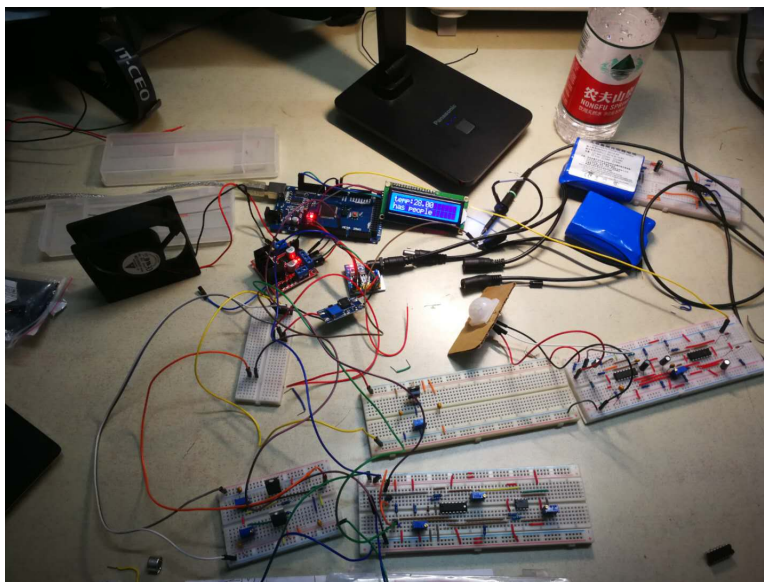


图 50: 整体调试电路

最终测试结果良好，所得展示视频一共有两个：人进出感应范围的视频“people_in_out.mp4”，以及人在范围内不同温度下风扇的转速不同的视频“people_in_temp_rise.mp4”，两个视频现都已经放在了“./2-experiment”文件夹下。

(提示, 温度和有无人的检测结果可以根据 LCD 显示屏上的结果来进行判断, 而对应的风扇转速变化如果不好肉眼观察, 可以通过听风扇转动声响来区分。当风扇转速实在过慢时, 风叶转动会有十分明显的噪声, 与正常情况下风扇的无噪声运动十分不同)

7 实验中电路出现的问题以及原因分析

这个项目的复杂度, 是我在设计之前从未想到的。我在这个过程中, 遇到的主要问题主要有如下几点:

1. 前一级输出电流不足的问题

前一级输出电流不足的问题, 在电源模块中体现的尤为明显。在正负电源模块中, 当我加入后一级电路后, 我发现前一级电路往往因为输出功率不够 (以及后一级输入电阻对前一级电路的影响) 的原因, 原先调整好的电压又会发生偏移, 需要重新对前一级电路的输出电压调整。情况严重时, 甚至无法通过调整的方式使得前一级输出电压恢复到期望电压。

对于这一问题, 我通过在前后级电路中间, 引入由功放构成的电压跟随器解决了问题。这样一来, 一方面功放基本可以满足后一级模拟电路的电流需求, 另一方面, 对于前一级电路而言, 后一级输入电阻相当于无穷大, 降低了对其带载能力的要求。所以带载能力的问题成功得到了解决。

2. 热释电传感器检测范围不足

对于我设计出来的第一版电路, 我最终测试所得的比较稳定的检测范围仅仅为 5-8cm, 这远远低于我项目的使用要求。又由于其还存在其他问题 (见 “4 基本电路模块电路原理/4.1 模拟电路模块/4. 热释电测量电路”), 所以我设计了第二版实验电路, 但即使是这样, 最终稳定的检测距离也只有 20cm 左右, 超出这一范围则会有漏检测和误检测的现象。这样的检测距离, 与市面上的被动式热释电传感器 7m 左右的检测范围相差过大。

经过我调查得知, 现在热释电传感器的主流电路实现方案为使用 BIS0001 运算芯片, 基本电路如图51所示。

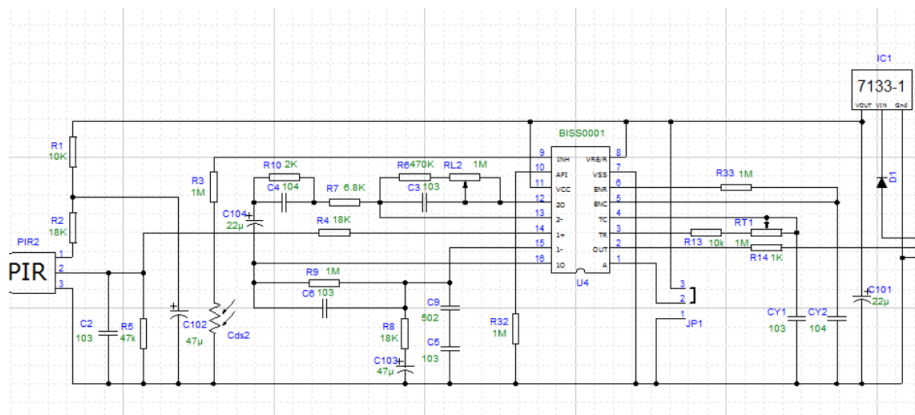


图 51: 经典热释电传感电路

BISS0001 运算芯片内部由多个运放以及一定的数字逻辑电路构成，其电路复杂度比个人的实验电路要复杂得多，所以能实现更为精确的检测精度和更加广阔的检测范围也就并不奇怪了。

3. PT1000 体积过小，给升温 and 温度测量带来了很大的麻烦。

在测量 PT1000 上温度和输出电压之间的关系，以验证电路设计没有问题的过程中，我发现，实验室的 PT1000 电阻为裸电阻，其体积不足半个指甲盖大，这给其上的升温，降温，以及温度检测带来了很大的不便。为了测量其上温度，并且尽量减少对电路的干扰，我从社团处借了一个手持热成像仪，通过主动式红外传感来获取 PT1000 上的近似温度；而为了对它进行升温 and 降温，由于缺乏实验条件，我只能用手捂和吹气的方式来改变 PT1000 表面温度，这也给实验带来了很大的不便。最终我仅仅获得几组比较有代表性的数据，没能够获得电路输出电平随环境温度的近似变化曲线。

4. 电源种类过多

这是我本次电路设计中的硬伤。在我最初设计的电路中，由于要使用到正负 12V 来给运放供电，而整体电路的电源仅为一个 12V 的锂电池，所以需要使用一个正负电源模块。而为了保证正负电源模块正常工作，又需要保证其输入电压大于 26V，这就意味着电路中又需要引入一个 DC/DC 升压模块。同时，由于部分电路和传感器（比如主控板 Arduino Mega 和 L298N 电机驱动模块）需要工作在 5V 电压下，所以电源管理电路中还需要一个 12V 转 5V 的电压模块。仅仅在电源管理电路中，电路就需要用到 3 个模拟电路模块：升压斩波电路，正负电源电路以及 5V 稳压电路模块。

这样的设计一方面会像上文所述，带来输出功率不足的问题；另一方面，功放和三端稳压器的引入会明显提升电路功耗，而这并不是一款便携式电子产品设计所希望看到的。

所以说，在第二版热释电传感电路的设计中，我就充分考虑到了这个问题。一方面，我将所需要的电源总数减少为 1 个，也就是说仅仅需要一个 5V 的电压；同时，为了配合这个 5V 的电压，使

得电路能够正常工作，我将所有运放的正极参考电压全部设置为 2.5V，通过这样的方式让电路近似的以 2.5V 作为中间值，在 2.5V 上下进行电压波动。不论是仿真还是实测都证明了这样的电路设计是可行的。

但是比较遗憾的是，由于时间问题，我没能够将温度传感电路也同样改为只需要 5V 电压的单电源电路。而从原理上来说，由于温度传感电路中，常用的电压值都在 0-5V 之间，所以温度传感电路是存在被改为单电源电路的可能的。

这个问题同时也让我理解到并行工程这一思想的重要性，不仅在设计电源时要考虑到各个电路模块的需求，也需要在设计电路模块时，减少所需要电源种类并降低对于电源质量的要求。这样一来，通过电路模块之间的相互配合来降低电路复杂度，提升电路工作性能。

同时，由于时间关系，整个项目比较遗憾的包括：

1. 没有减少电源种类，使得现在的电源管理模块数量过多
2. 超出课程所学范围的升压斩波电路并没有自己搭建，并研究一下其工作原理和实际工作特性。
3. 由于 Arduino Mega ADC 芯片的精度，最终只能实现精度为 0.5°C 的温度测量
4. Multisim 没能够使用支电路的方式，进行一次整体电路仿真
5. 没能够在调试完整体电路以后，用 Altium Designer 画一块双层 PCB 板来简化电路连线
6. 没有足够的时间来实现自己原本想要编写的手机端 APP
7. 所使用的显示屏过于简单，缺少进一步的人机互动

8 项目的改进方向

项目下一步的改进方向包括有：

8.1 模拟电路部分

1. 对于热释电传感器，可以尝试使用 BIS0001 芯片进行更高层次的模拟电路设计，以此来进一步提升热释电传感的检测精度
2. 对于温度检测电路，在电桥电路处使用恒流源电路来将 PT1000 阻值变化直接转变为电信号，从而省去近似估算的过程，提高电路检测的精确度。

同时，考虑使用在淘宝上有售卖的更大一些的 PT1000 探头而不是 PT1000 裸电阻。这样一来，一方面是方便测试其 PT1000 表面温度，另一方面也方便将其和电路隔离开来，单独对其进行加热

降温等操作，从而获得更大的温度变化范围，能够更加准确的验证输出电压和 PT1000 表面温度之间的关系。

除此以外，将双电源改为单电源供电也应该是下一步改进的方向之一。

最后，可以考虑使用差分放大芯片代替精密运放以实现电路的差分放大。但是我并不确定这样做能够提升检测精度，只能说可以将其仅仅作为一个改进的尝试方向。

3. 对于电源管理电路，一方面，应该尽可能的减少所使用的电源种类，这样一来既可以减少功放管的使用，又可以降低电路功耗。

8.2 数字电路部分

1. 对于主控板，由于在实验过程中，Arduino 上的 ADC 芯片精度不够，导致最终温度的可识别精度也只有 0.5 摄氏度。为了进一步提升温度检测的准确度除了在模拟电路部分进一步做电路改进以外，也应该考虑在主控板上外接一个精度更高的 AD 采集模块（比如 16 位 AD 采集模块），或者更换 Arduino 控制板为 STM32 开发板（前者使用的是一块 10 位 AD 模块，后者使用的是一块 12 位逐次逼近型 AD 转换模块）。
2. 更换 LCD 显示屏。可以更换其为更大的 LCD 显示屏甚至是触摸屏，增强与用户之间的互动性。
3. 开发蓝牙移动端 APP，让用户可以通过手机直接操控风扇