论文作者：何克劲

院 系：电子信息学院 \_

专 业：计算机科学与技术

研究方向：智能通风柜手势控制界面设计与实现

指导教师：曾祥绪老师

学士学位论文

智能通风柜手势控制界面设计与实现

Design and Implementation of Hand Gesture Control Interface for Intelligent Fume Cabinet

2019年5月24日

学校代码： 11458

学 号： 151002200336

分类号： TP39 密 级：\_\_\_\_\_\_\_

U D C：\_\_\_\_\_\_\_

摘 要

随着传感器的不断发展，简单的信号已经不能满足人们日益增长的对控制的需求。同时，为了方便，传感器向着简化生活的方向不断发展。对于普通的通风柜来说，它单一的功能不能满足简化生活的目的，因此出现了智能通风柜。智能通风柜在单片机的作用下，把输入设备与输出设备有机的结合起来，不仅可以简化操作流程，同时其智能的操作，不仅能够出色的完成当前的任务，同时节省时间，加快工作效率。

关键词：传感器，手势识别， 控制界面，单片机

**ABSTRACT**

With the continuous development of sensors, simple signals can no longer meet the growing demand for control. At the same time, for convenience, sensors are developing in the direction of simplifying life. For ordinary ventilation cabinet, its single function can not meet the purpose of simplifying life, so there is intelligent ventilation cabinet. Intelligent ventilation cabinet, under the function of single-chip computer, combines input equipment and output equipment organically, not only can simplify the operation process, but also its intelligent operation can not only accomplish the current task, but also save time and speed up work efficiency.

**Key words:** sensor，Gesture Recognition，Control Interface，Single Chip Microcomputer

目 录

1绪论---------------------------------------------------------------------------------------------------------------------1

1.1课题研究背景和意义-------------------------------------------------------------------------------------1

1.2国内外研究现状和发展----------------------------------------------------------------------------------1

1.3研究内容-------------------------------------------------------------------------------------------------------2

1.3.1单片机的通讯--------------------------------------------------------------------------------------2

1.3.2单片机的数据处理-------------------------------------------------------------------------------2

1.3.3操作界面的设计与数据显示-----------------------------------------------------------------2

1.4研究方案-------------------------------------------------------------------------------------------------------3

2单片机概述---------------------------------------------------------------------------------------------------------4

2.1单片机介绍与选择-----------------------------------------------------------------------------------------4

2.2Arduino单片机控制原理--------------------------------------------------------------------------------4

2.2.1Arduino Uno概述---------------------------------------------------------------------------------4

2.2.2输入输出--------------------------------------------------------------------------------------------4

2.2.3通讯---------------------------------------------------------------------------------------------------5

2.2.4模拟信号--------------------------------------------------------------------------------------------6

2.2.5通讯方式--------------------------------------------------------------------------------------------7

3手势控制技术概述---------------------------------------------------------------------------------------------8

3.1手势模块介绍------------------------------------------------------------------------------------------------8

3.2 PAJ7620型模块---------------------------------------------------------------------------------------------8

3.3 MARS型模块--------------------------------------------------------------------------------------------------9

3.4比较分析与最终选择------------------------------------------------------------------------------------10

4控制界面概述---------------------------------------------------------------------------------------------------11

4.1图形界面设计----------------------------------------------------------------------------------------------11

4.2信号的反馈与处理----------------------------------------------------------------------------------------11

5 硬件电路的设计与实现----------------------------------------------------------------------------------13

5.1硬件总体结构----------------------------------------------------------------------------------------------13

5.2主要硬件模块的设计------------------------------------------------------------------------------------13

5.2.1电机驱动模块的设计--------------------------------------------------------------------------13

5.2.2电机模块设计------------------------------------------------------------------------------------15

5.2.3触摸屏模块设计---------------------------------------------------------------------------------15

5.2.4手势识别模块设计-----------------------------------------------------------------------------15

5.2.5限位模块设计------------------------------------------------------------------------------------15

5.3电机的选择和电机驱动模块的设计---------------------------------------------------------------15

6总系统的设计与验证---------------------------------------------------------------------------------------17

6.1限位检测系统设计---------------------------------------------------------------------------------------17

6.2执行系统设计----------------------------------------------------------------------------------------------18

6.3显示界面设计----------------------------------------------------------------------------------------------20

6.3.1矢量图的制作------------------------------------------------------------------------------------20

6.3.2背景图片的编辑---------------------------------------------------------------------------------21

6.3.3数据的传递---------------------------------------------------------------------------------------22

6.4手势识别系统设计---------------------------------------------------------------------------------------22

6.5总系统设计与布局---------------------------------------------------------------------------------------23

7总结与展望-------------------------------------------------------------------------------------------------------24

7.1本文的工作总结-------------------------------------------------------------------------------------------24

7.2进一步展望--------------------------------------------------------------------------------------------------24

参考文献--------------------------------------------------------------------------------------------------------------25

致谢----------------------------------------------------------------------------------------------------------------------26

附页----------------------------------------------------------------------------------------------------------------------27

**1 绪论**

1.1课题研究背景和意义

人工智能是目前越来越火的词，同时也是未来发展的趋势。让机器懂得人类的语言，懂得人类的手势，让其更加人性化。在过去的几十年，计算机技术在飞速地发展，各行各业中，计算机技术给人们带来非常多的便捷，人们己经与计算机技术密不可分。其中，人与计算机的信息交互是必不可缺的一步[1]。手势是一种人类的基本特征，也是一种交流的语言。手势识别技术的发展，为人与机器的交互提供了无尽的可能性。同时，手势识别也能让机器听懂人类的一种语言。手势控制对于简化操作有着非常重要的意义。它不仅可以让机器听懂人的意思，还可以节省时间，方便使用它的人，让生活更便利。

随着科技的不断发展，对手势识别技术的研究已经成为非常热门的课题，基于手势识别技术的应用也开始渗透到人们生活的方方面面。同时，手势识别技术无论是在手势样本的采集，还是手势设计上都有自己的优势，手势识别的深入研究有着很重要的意义[2]。手势识别技术主要有下面几个应用领域：

（1）人机交互

最常见的应用在计算机端游戏中，通过设备采集当前人物的动作或手势，间接取代传统的敲击键盘或点击鼠标进行游戏控制。近年来，一些虚拟现实的设备中已经应用了手势识别作为输入信号。

（2）安全驾驶

将手势识别应用到辅助驾驶系统中，可以一定程度的提高行车驾驶的安全性，减少交通事故的发生。

（3）手语认知

手势也是一种语言。手语是聋哑人的主要表达方式。但对于未经过专业训练的人来说，很难理解手势的具体语言，因此需要一些辅助设备能够帮助识别手势所代表的意义。

（4）机器控制

手势控制在智能家居中也是很常见的一种控制方式。随着人工智能的普及，越来越多的家庭的家居开始变得更加智能化。使用手势控制取代传统的按键控制，将会使更多用户感到舒适自然与智能。

1.2国内外研究现状和发展

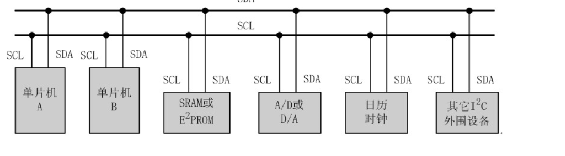
根据应用目的的不同，手势可分为四大类：控制手势、对话手势、通信手势和操作手势。其中最重要的对话手势和通信手势即为“手语”，由于其结构性很强，很容易被用于计算机视觉算法的实验平台的设计［3］。国外手势识别的研究起步较早，发展比较快。William [4]等人使用静态手势来控制电视机。L.H.Howe [5]等人提出一种在使用肤色概率模型和帧相减的技术实现双手的检测与分割。M.Elmezain[6]等人在 YcrCb 空间使用混合高斯模型对肤色进行建模，使用阈值的方法进行手势分割，使用隐马科夫模型进行手势的判定，跟踪手势的运动轨迹，实现 0－9这十个阿拉伯数字的动态手势识别。

我国的手势识别研究起步相对于国外较晚，但是发展较快，取得了一定成果。张秋余、姚开博[7]使用矩形特征描述手势来满足精度和速度的要求，同时使用改进的Adaboost算法解决分类器过度训练问题，在手势识别和跟踪上取得了较好的效果。中科院的单彩峰[8]提出了带均值漂移的粒子滤波，提出时序模板轨迹的概念，即将时空轨迹运用到时序模板中，从而将跟踪得到的手势运动轨迹在单幅图像中表示，实现对动态手势识别。

1.3研究内容

1.3.1单片机的通讯

单片机通讯可以使用TX/RX的串口通讯，同时也可以使用I2C协议进行通讯。虽然它们之间的协议不尽相同，但是都可以实现设备间的通讯，完成数据间的交流与处理。TX/RX的串口通讯协议只需设备间的TX/RX相互交错连接，即可实现通讯，且可同时实现通讯。I2C通讯中总线只有两根双向信号线，一根是数据线SDA，另一根时时钟线SCL。原理如图所示：



**图1-1 I2C通讯原理**

1.3.2单片机的数据处理

当单片机读取到串口中的信号时，反馈给一个引脚高/低电平的数字信号，从而实现数据间的处理。

1.3.3操作界面的设计与数据显示

其操作界面可以通过自己制作贴图的方式，同时给予触摸位置按钮开关的命令，实现在满足界面的条件下，对外部设备的控制与显示的作用。

在考虑有没有下注的情况下，进行不同的反馈。同时给予一定的动画效果，让其看上去更加真实。具体效果如图所示：

本课题主要研究的内容有单片机与显示屏的通信，单片机的数据处理何显示屏的界面设计。论文分为七章：

第一章：论述了国内外对于手势识别的一些发展，并简要介绍论文的主要研究内容与研究方案。

第二章：介绍了当前的单片机的种类和未来的发展趋势，简单介绍Arduino单片机的控制原理。

第三章：阐述了手势控制的一些原理和方法，介绍选用的两种模块在对手势控制的原理，比较选用的两种手势控制传感器的不同与优缺点，和最终选用的手势识别传感器。

第四章：介绍了采用怎样的图形界面显示屏，和对其界面的设计，将单片机的信号处理并反馈在显示屏中和显示屏的信号传入单片机时单片机的处理和执行。

第五章：介绍了一些在实现这些功能的同时所需要的一些其他的电器元件，及其整个系统的电气线路图的布局。

第六章：对整个系统的整合与设计。

第七章：展望与总结。

1.4研究方案

本文将采用两个手势识别传感器对其进行可行性的探究，并发现他们在各方面的优点，并选择其中一种手势识别传感器完成接下来的工作。

使用两种以上的触碰传感器，比较优劣，并根据具体工况下选择合理的触碰传感器。选择适合的触摸屏，并且能够设计多种方案，完成触摸屏的界面设计。

**2单片机概述**

2.1单片机介绍与选择

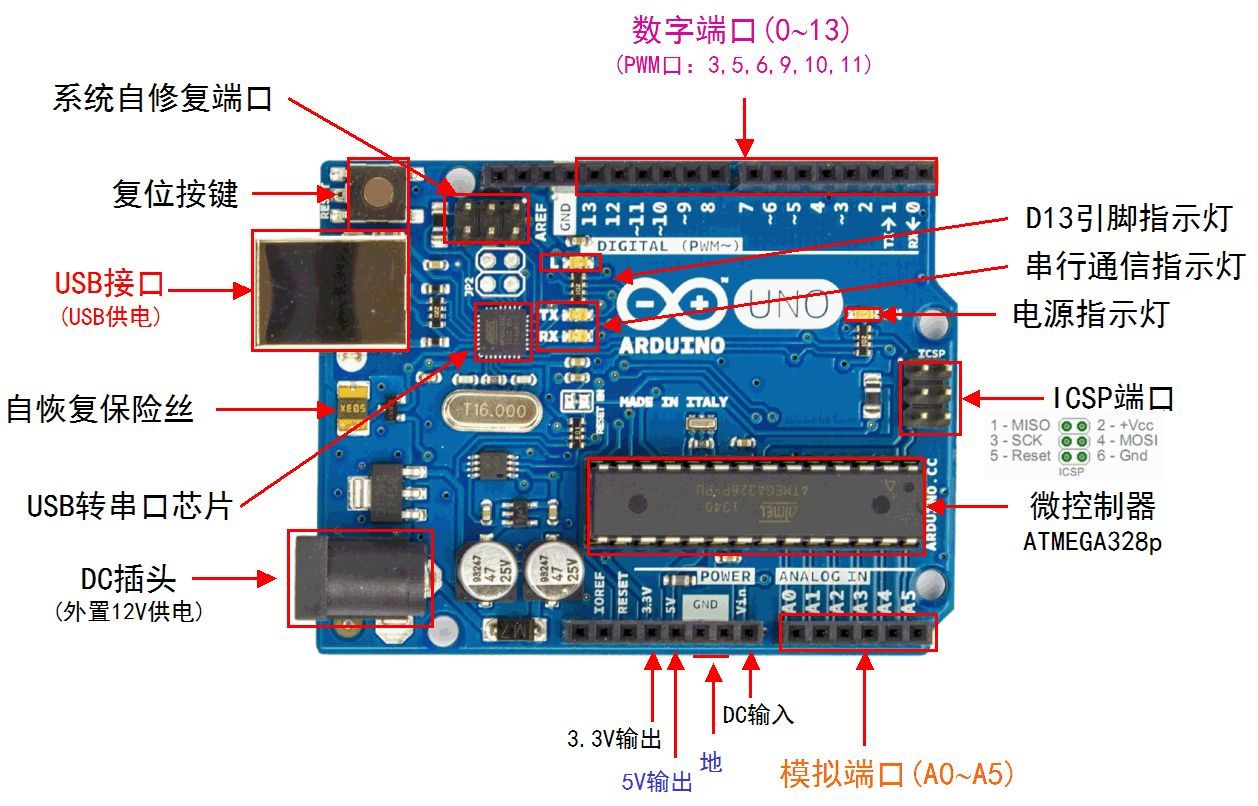
单片机称为单片微控制器,它不是完成某一个逻辑功能的芯片,而是把一个计算机系统集成到一个芯片上。相当于一个微型的计算机，和计算机相比，单片机只缺少了I/O设备。它的体积小、质量轻、价格便宜、为学习、应用和开发提供了便利条件。同时，学习使用单片机是了解计算机原理与结构的最佳选择。

单片机的使用领域已十分广泛，如智能仪表、实时工控、通讯设备、导航系统、家用电器等。各种产品一旦用上了单片机，就能起到使产品升级换代的功效，常在产品名称前冠以形容词——“智能型”，如智能型洗衣机等。

2.2 Arduino单片机控制原理

2.2.1Arduino Uno概述

Arduino UNO是基于ATmega328P的Arduino开发板。它有14个数字输入/输出引脚（其中6个可用于PWM输出）、6个模拟输入引脚，一个16 MHz的晶体振荡器，一个USB接口，一个DC接口，一个ICSP接口，一个复位按钮。它包含了微控制器所需的一切，你只用简单地把它连接到计算机的USB接口，或者使用AC-DC适配器，再或者用电池，就可以驱动它。



**图2-1 Arduino单片机**

2.2.2输入输出

Arduino Uno有14个数字输入输出引脚，可使用 pinMode()、digitalWrite() 和 digitalRead() 控制。其中一些带有特殊功能，这些引脚如下：

(1)Serial：

0（RX）、1（TX），被用于接收和发送串口数据。这两个引脚通过连接到ATmega16u2来与计算机进行串口通信。

(2)外部中断：

2、3，可以输入外部中断信号。中断有四种触发模式：低电平触发、电平改变触发、上升沿触发、下降沿触发。

(3)PWM输出：

3、5、6、9、10、11，可用于输出8-bit PWM波。对应函数 analogWrite() 。

(4)SPI：

10（SS）、11（MOSI）、12（MISO）、13（SCK），可用于SPI通信。可以使用官方提供的SPI库操纵。

(5)L-LED：

13号引脚连接了一个LED，当引脚输出高电平时打开LED，当引脚输出低电平时关闭LED。

(6)TWI：

A4（SDA）、A5（SCL）和TWI接口，可用于TWI通信，兼容I²C通信。可以使用官方提供的Wire库操纵。

Arduino Uno 6个模拟输入引脚，可使用analogRead()读取模拟值。每个模拟输入都有10位分辨率（即1024个不同的值）。默认情况下，模拟输入电压范围为0～5V，可使用 AREF引脚和analogReference()函数设置其他参考电压。

(7)AREF：

模拟输入参考电压输入引脚。

(8)Reset：

复位端口。接低电平会使Arduino复位，复位按键按下时，会使该端口接到低电平，从而让Arduino复位。

2.2.3通讯

Arduino UNO具备多种通信接口，可以和计算机、其他Arduino或者其他控制器通信。ATmega328 提供了UART TTL (5V)串口通信，其位于0 (RX) 和1 (TX)两个引脚上。Uno上的ATmega16U2会在计算机上模拟出一个USB串口，使得ATmega328 能和计算机通信。Arduino IDE提供了串口监视器，使用它可以收发简单文本数据。Uno上的RX\TX两个LED可以指示当前Uno的通信状态。SoftwareSerial库可以将Uno的任意数字引脚模拟成串口，从而进行串口通信。ATmega328也支持I2C (TWI)和SPI通信。Arduino IDE自带的Wire库，可用于驱动I2C总线，自带的SPI库，可用于SPI通信。

单片机在通讯加持下可以完成数据传递任务，在传输消息同时，提高数据传递及接收效率，并可以实现循环往复通讯目标。单片机作为可编程逻辑控制器会率先接收数据及相关指令，随后向上位机传递数据指令，待单片机接收信息后读取、识别、执行数据指令，完成要求的部分功能，继而达到单片机通讯目的[9]。

2.2.4模拟信号

(1)数字信号

数字信号是简单的逻辑信号，只有0、1，为离散信号，且不连续。这种信号的自变量用整数表示，因变量用有限数字中的一个数字来表示。在计算机中，数字信号的大小常用有限位的二进制数表示。计算机领域中数字信号应用最广。通过组合，可以形成00，01，10，11等信号，扩展了信号信息。由于它是简单的逻辑信息，因此在抗干扰能力方面较强，不易丢失细节信息。对于生活中，数字信号体现在电灯的开关，门的开与关、阴阳等。在单片机方面，可以实现简单的逻辑检测与单一信号的赋予。

由于数字信号是用两种物理状态来表示0和1的，故其抵抗材料本身干扰和环境干扰的能力都比模拟信号强很多；在现代技术的信号处理中，数字信号发挥的作用越来越大，几乎复杂的信号处理都离不开数字信号；或者说，只要能把解决问题的方法用数学公式表示，就能用计算机来处理代表物理量的数字信号[10]。

（2）模拟信号

模拟信号是数字信号的延续，体现在它的连续性。它显著的有点是能保存较为准确的具体数据，但是在传输中，又容易损失细节，抗干扰能力较弱。对于失真的信号将会失去其作为数据的意义。在生活中体现在温度、压力、电流电压的变化等。实际生产生活中的各种物理量，如摄相机摄下的图像、录音机录下的声音、车间控制室所记录的压力、流速、转速、湿度等等都是模拟信号。

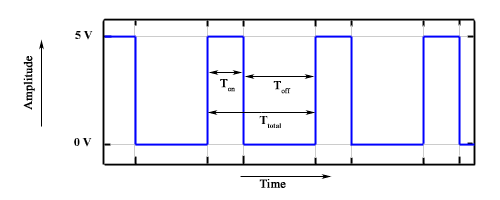
（3）脉冲宽度调制

脉冲宽度调制是利用微处理器的数字输出来对模拟电路进行控制的一种非常有效的技术，广泛应用在从测量、通信到功率控制与变换的许多领域中。

脉冲宽度调制是一种模拟控制方式，其根据相应载荷的变化来调制晶体管基极或MOS管栅极的偏置，来实现晶体管或MOS管导通时间的改变，从而实现开关稳压电源输出的改变。这种方式能使电源的输出电压在工作条件变化时保持恒定，是利用微处理器的数字信号对模拟电路进行控制的一种非常有效的技术[11]。

简单的说，比如你有5V电源，要控制一台灯的亮度，有一个传统办法，就是串联一个可调电阻，改变电阻，灯的亮度就会改变。还有一个办法，就是PWM调节。不用串联电阻，而是串联一个开关。假设在1秒内，有0.5秒的时间开关是打开的，0.5秒关闭，那么灯就亮0.5秒，灭0.5秒。这样持续下去，灯就会闪烁。如果把频率调高一点，比如是1毫秒，0.5毫秒开，0.5毫秒灭，那么灯的闪烁频率就很高。我们知道，闪烁频率超过一定值，人眼就会感觉不到。所以，这时你看不到灯的闪烁，只看到灯的亮度只有原来的一半。同理，如果1毫秒内，0.1毫秒开，0.9毫秒灭，那么，灯的亮度就只有原来的十分之一。这就是PWM的基本原理。

PWM控制技术以其控制简单，灵活和动态响应好的优点而成为电力电子技术最广泛应用的控制方式，也是人们研究的热点。由于当今科学技术的发展已经没有了学科之间的界限，结合现代控制理论思想或实现无谐振波开关技术将会成为PWM控制技术发展的主要方向之一。



**图2-2 PWM控制原理**

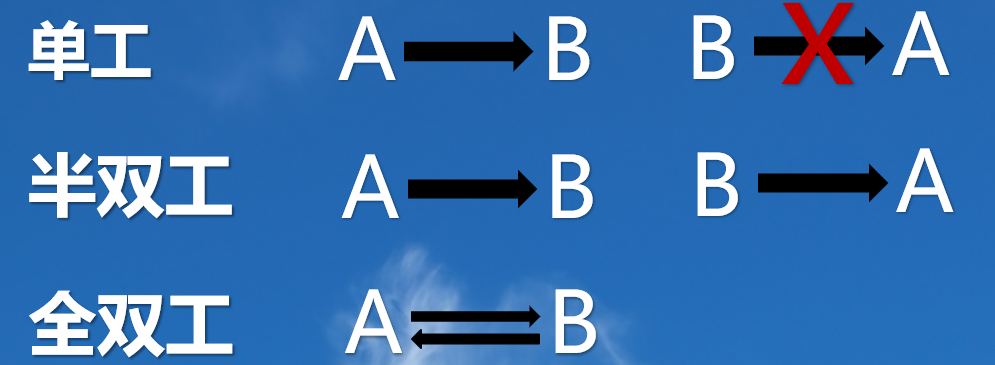
2.2.5通讯方式

双工通信犹如打电话一般，可以在同一时刻完成数据双向传输任务，将接收机与发射机置于两个频率上并同时工作，双工囊括双工车载机、双工手持机、双供基地/中转台等，通常在 UHF、VHF 频段上进行跨段工作，UHF、VHF 均可交叉接收或发射信号，其中双工/中站台及双工车载机可以同时进行跨段双工工作，通过无线接驳器还可将有线网络与无线网络联接到一起，实现对讲机与有线电话通信目标，同时，对讲机也可通过拨号与有线电话进行通信，完成数据传输工作[12]。

全双工是指在发送数据的同时也能够接收数据，两者同步进行，这好像我们平时打电话一样，说话的同时也能够听到对方的声音。目前的网卡一般都支持全双工。

所谓半双工是指一段时间内，只有一种动作发生，例如一条窄路，而只有一辆车可以通行，当前有两辆车相对时，在这种情况下，只有一辆车先开，等到另一辆车的头部再开，这个例子生动地说明了原始半双工通讯。早期对讲机、早期集线器等设备都是基于半双工产品的。随着技术的不断进步，半双工会逐渐退出历史舞台。

单工通信是指通信线路上的数据按单一方向传送。



**图2-23 通讯方式**

**3 手势控制技术概述**

3.1手势识别模块介绍

手势识别作为人机交互的重要组成部分，其研究发展影响着人机交互的自然性和灵活性。目前大多数研究者均将注意力集中在手势的最终识别方面，通常会将手势背景简化，并在单一背景下利用所研究的算法将手势进行分割，然后采用常用的识别方法将手势表达的含义通过系统分析出来 但在现实应用中，手势通常处于复杂的环境下，例如: 光线过亮或过暗有较多手势存在手势距采集设备距离不同等各种复杂背景因素。这些方面的难题目前尚未得到解决，且将来也难以解决 因此需要研究人员就目前所预想到的难题在特定环境下加以解决，进而通过多种方法的结合来实现适于不同复杂环境下的手势识别，由此对手势识别研究及未来人性化的人机交互做出贡献。手势识别传感器可以用于非接触式控制场景，如鼠标、智能家居、汽车设备控制、机器人交互等。

对于在当下应用较为广泛的人机交互设备，它们由于操作模式的原因，达不到让操作者自如使用的目标，不具备良好的交互性。伴随着信息技术的飞速发展，与鼠标等传统人机交互模式相比，更加自然直观的触控类、语音类、视觉类等人机交互模式相继出现且不断完善[13]。 肢体语言是人类传递信息的一种重要手段，而在其中的手势更是一种自然、直观、且符合人们生活习惯的信息传递方式。

对于一些生活中比较简单常见的手势，其含义通常被大众所接受，但为了让手势识别应用能够推广，需要更多的手势来对应用进行控制[14]。由于地域以及个人习惯的不同，既可能出现使用者用一个手势表达多种意思的情况，也可能出现不同的使用者使用不同的手势表达同一意思的情况，这就给基于手势识别的而应用带来了极大的困难[15]。

3.2 PAJ7620型模块

PAJ7620手势识别传感器是将手势识别功能与通用I2C接口集成到单个芯片中。它可以识别九种手势，包括向上移动，向下移动，向左移动、向右移动、向前移动、向后移动、顺时针方向、逆时针方向、摇摆运动。手势信息通过I2C总线访问，使用对应arduino库函数即可实现变成控制，手势识别模块返回的信息可以用作机器接受的控制信号，从而实现对机器的控制。内置的识别算法相对智能，能够把双手从生硬的按键中解放出来。

PAJ7620手势识别传感器具有如下特点：

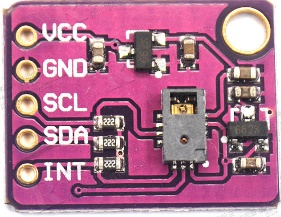
(1)可以识别九种手势；

(2)手势速度可以达到60°/s至600°/s

(3)环境光免疫力<100klux；

(4)内置了接近检测

(5)I2C接口高达400kbit/s。



**图3-1 PAJ7620**

3.3 MARS型模块

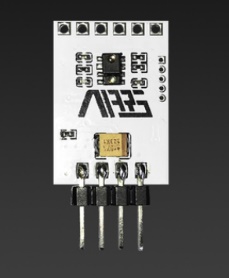
MARS型模块是火星科技自主研发的手势识别模块，是代替传统家居机械按键或触摸按键的完美选择。其隔空操作的模式简单易学、快捷、方便、舒适，只需一个手势动作,即可实现功能切换。能支持上、下、左、右、接近、远离共6种手势动作识别。识别最大距离约0~10cm，反应速度快、识别灵敏、识别率高；该模块提供了贴片和排针方式，既可以通过贴片方式作为客户设计的电路板中的一部分，也可以通过排针方式单独做为一个普通模块使用；极小的体积，十分方便嵌入到各种产品中，性价比很高；模块自带一颗MCU，用于处理底层驱动和生成帧通信格式，用户无须为复杂的底层驱动、传感器感应计算、校准等工作而烦恼，识别结果直接通过IO口映射输出 和 串口数据帧格式输出，降低了客户的开发难度，极大的缩短产品开发周期。只要是具有串口功能的MCU，如STM32、STC15、Arduino、MSP430等单片机，都可以通过解析简单的数据帧，获取识别结果。IO口映射输出的方式无须单片机参与，模块板载的MCU即可在指定IO口输出编码好的高低电平，通过简单的搭建外围电路，实现直接控制负载，如LED、可控硅、继电器等。它的应用非常广，可以手势识别、颜色识别、环境光照检测、智能手机、替代传统机械开关。

该传感器相较于传统的机械或触摸按键，具有如下优点：

(1)避免指纹触摸屏幕，可穿戴设备屏幕较小，采用手势控制更为方便。

(2)避免用湿/脏手触摸设备。

(3)使可穿戴产品炫酷。



**图3-2 MARS模块**

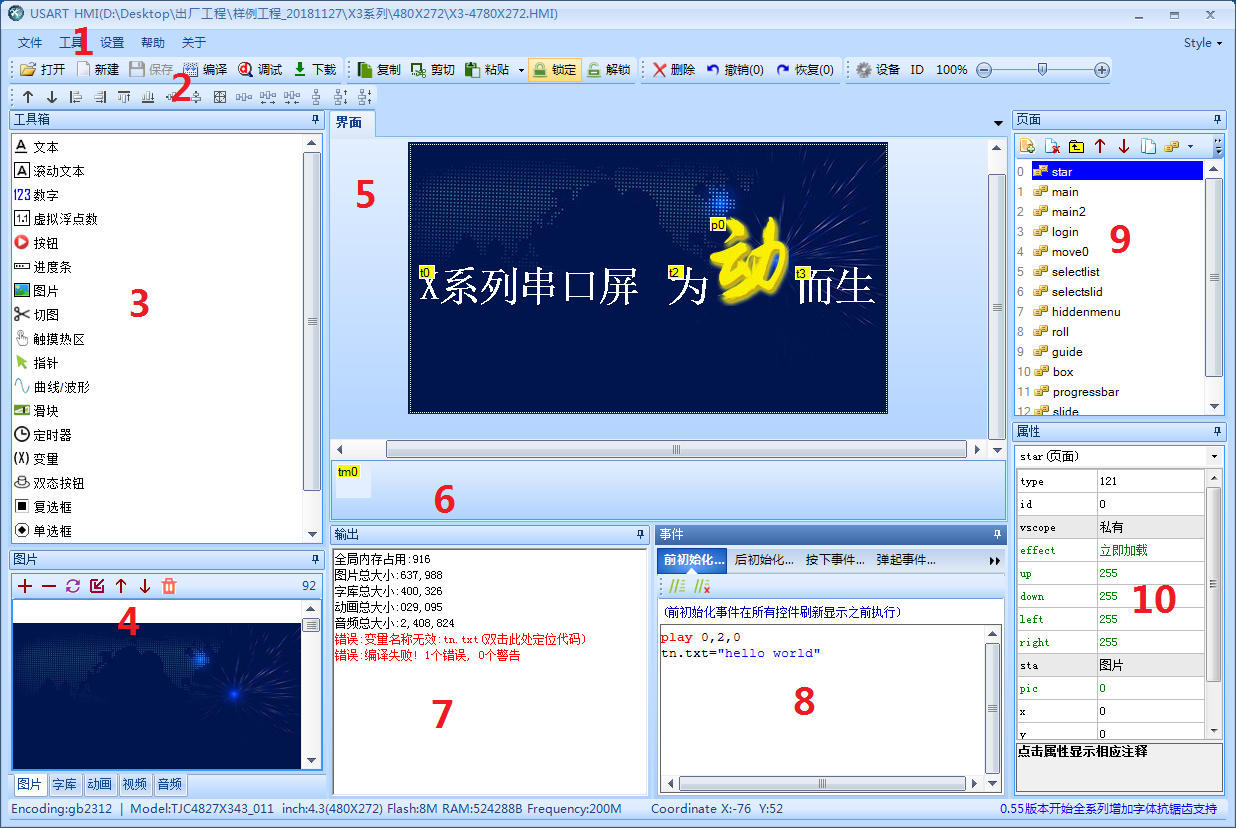
3.4比较分析与最终选择

两者在使用时都能很好的识别出正确的手势，并且都能够达到预期的目的，但相比起来，PAJ7620能够识别的距离更远，且在较远的距离识别的准确率相对于MARS模块来说更高，PAJ可以识别9中，但MARS只能识别六种，且MARS的价格比PAJ的价格高，但是MARS做工更加精美，且具有完善MCU芯片，能够更好的处理数据。综上所述，选择使用PAJ7620作为课题的使用模块。

**4 控制界面概述**

4.1图形界面

图形界面使用USART HMI智能串口液晶屏，能够基本满足本论文的工作量。USART HMI显示屏能够满足显示的情况下，通过电容触摸的方式采集当前界面所对应的反馈信号，实现触摸反馈。触摸屏能够通过开源软件进行编程，实现对应的信号处理，其上位机如图所示：

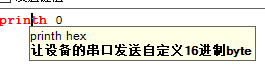


**图4-1 HMI图形界面**

3为可以使用的工具箱，通过拖动的方法，实现图形化编程，里面可以显示文字、数字、图片等，同时可以拖动按钮等，实现触发信号，触发信号可以通过串口的方式将信号发送给单片机，让单片机实现具体的触发动作；4为库，需要把需要的图片、字体、动画、音频等文件导入其中才可使用具体素材；5为当前人物页面状态，可以在此进行编辑；7为状态栏，可以查看是否编辑成功；8为具体编写程序位置，可以在对应的操作中添加具体的代码，实现控制功能；9为页面，可以实现相互的跳转；10为变量定义，可以进行名称的定义与参数的赋予。

4.2信号的反馈与处理

使用printh让设备的串口发送自定义16进制的数据，如图所示：



**图4-2 16进制数据**

因此可以在按下事件中使用pirith函数实现当被按下时，发送自定义的16进制的数据，当单片机收到这个数据时，控制执行器执行相应的操作。硬件如图所示：

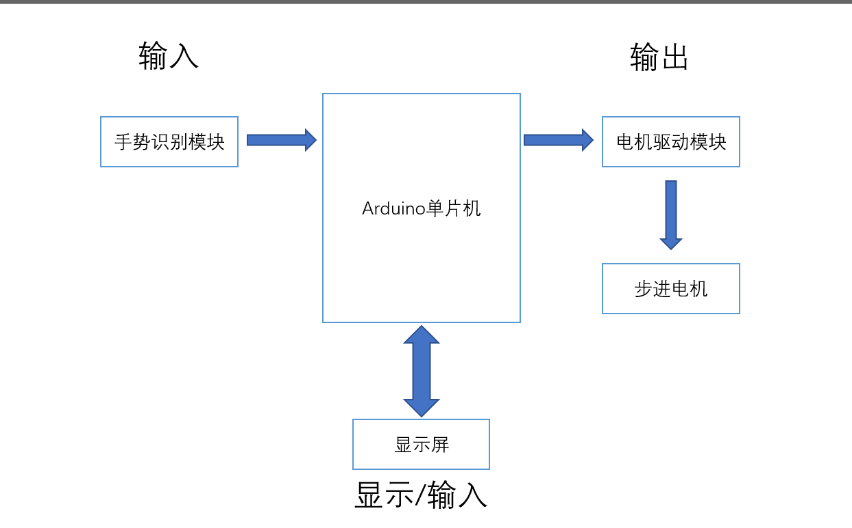


**图4-3 HMI硬件**

**5 硬件电路的设计与实现**

5.1硬件总体结构

本次的智能通风柜的硬件系统的实质是围绕怎样用主控芯片去控制外设，所以整个系统由Arduino单片机、电机驱动模块、步进电机、触摸屏、手势识别模块等组成。Arduino单片机主要负责处理来自触摸屏中传递的信号，接收和处理来自手势识别模块的信号和输出控制信号给电机驱动模块控制步进电机。硬件系统结构图如图所示：

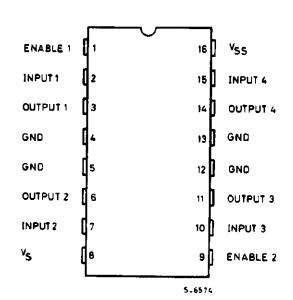


**图5-1 硬件系统结构图**

5.2主要硬件模块的设计

5.2.1电机驱动模块的设计

电机驱动模块采用L293D芯片实现其电机的正转与反转。在控制时可以直接采用L293D控制板，但是其在设计中费用相对直接采用芯片驱动来说，价格更贵，且在写程序的时候需要额外增加库，需要了解它的定义，因此最终采用的是L293D芯片去直接驱动。L293D芯片引脚定义如下：

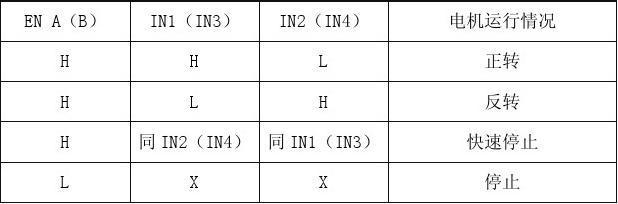


**图5-2 L293D芯片引脚**

因为只需要控制一个电机，因此我们只需要使用一半的引脚即可实现控制电机的正反转。具体控制原理如下：

ENABLE主要作用是实现控制的快满，通过脉冲宽度调制的方法，可以实现其变速，但是目前的功能不需要，因此可以直接给它供弱电即可实现功能；

INPUT1、INPUT2作为输入信号，通过简单的逻辑实现电机的真反转与停止，它的真值表如图所示：



**图5-3 L293D芯片逻辑关系**

INPUT与单片机的数字引脚相连，可以发送高低电平给芯片，当IN1与IN2收到的信号互为相反时，可以实现电机的正反转；当他们都为高电平或低电平时时，实现电机的停止。

OUTPUT1、OUTPUT2与不能直接与电机两端连接，需要通过继电器实现弱电控制强电，如果直接连接，可能造成的后果是，若用弱电驱动，则带不动电机的旋转，驱动效果很差；若用强点驱动，则过高的电压会直接击穿单片机，造成单片机的损坏，因此需要使用到继电器来间接去控制电机的旋转。

GND为接地线，VSS为弱电供电处，可以给其供5v弱电。具体接线可如图所示：

5.2.2电机模块设计

电机模块可以直接采用带有减速器的电机，因为电机驱动都需要12V或着24V的电压才可以实现正常的运动，因此，需要使用外部供电的方法，通过弱电控制强电的方法，实现单片机对其的控制。

5.2.3触摸屏模块设计

触摸屏使用USART HMI触摸屏，能够将触摸屏的信息通过编程将信号发送给单片机进行处理，实现其功能。

5.2.4手势识别模块设计

手势识别使用的是PAJ7620模块，他能够实现9种手势的识别。

5.2.5限位模块设计

限位开关使用的触碰开关，使用触碰开关的作用是，当升降的门窗快到达极限位置时，能够给予触发信号，让其停止。触碰开关为数字信号，当被触碰到时，给予0信号，当单片机收到这个信号时，控制电机停止，实现限位的作用。



**图5-4 限位开关**

5.3电机的选择和电机驱动模块的设计

使用直径为20mm的套筒时，若要实现每秒上升5cm的速度时，需要计算额定转速。每转一圈直线行程为：

转换单位为：

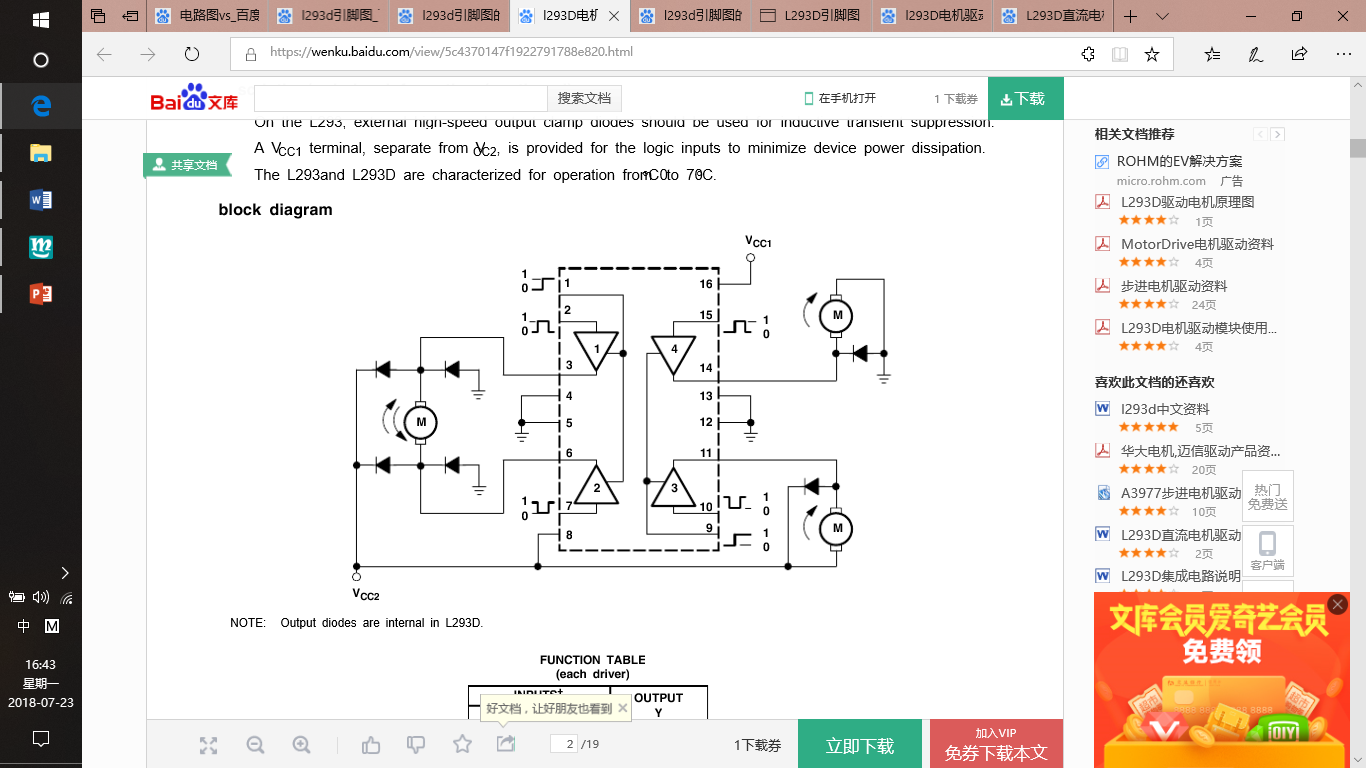
若要实现每秒5cm的速度时，则需要电机的转速为：

通过查阅下表，可以采用DV24V2435型号，额定转速为5rpm的电机，可以实现每秒上升5cm的功能。



**图5-5 电机选型表**

电路设计如图所示，通过L293D与继电器和电机通过连接，可以实现对电机的控制，电机供电为24V，单片机和L293D芯片使用5V供电，通过继电器，实现弱电控制强电。



**图5-6 L293D芯片连线原理**

**6 总系统的设计与验证**

6.1限位检测系统设计

限位检测系统主要的作用是，限制通风柜在上升到极限高度时能够停下，防止一直的运动导致整体系统的损坏。限位检测系统运用非常广泛，很多机构都采用限位检测来实现整体功能的限位与误操作。限位系统包括2个部分，一部分为上限位，一部分为下限位，通过限位开关实现其限位作用。传输信号为数字信号，其实现原理为：当触碰到限位开关时，限位开关发送信号0，当单片机收到连接该传感器的引脚的信号时，去控制执行机构停止运动，具体实现代码如下所示：

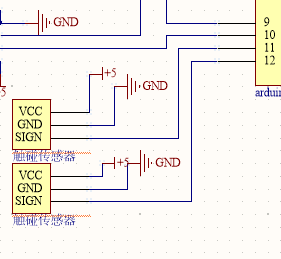
|  |
| --- |
| if (digitalRead(11) == 0)//如果限位开关被触发  {  digitalWrite(9,HIGH);//电机停止  digitalWrite(10,HIGH);//电机停止  }  if (digitalRead(12) == 0)//如果限位开关被触发  {  digitalWrite(9,HIGH);//电机停止  digitalWrite(10,HIGH);//电机停止  } |

其中，Pin11连着上触碰开关，Pin12连着下触碰开关，Pin9、Pin10分别连着控制继电器，继电器分别控制电机的两个引脚，实现正反转。使用if语句进行判断，当数字Pin11被触碰到，发出一个信号0时，单片机检测到发送的信号时，给予的反馈是将Pin9、Pin10置为高电平，及信号1，对应L293D芯片的快速停止指令；同理，当数字Pin12被触碰到，发出一个信号0时，单片机检测到发送的信号时，给予的反馈是将Pin9、Pin10置为高电平，及信号1，对应L293D芯片的快速停止指令。

digitalWrite（）函数是Arduino单片机的数字信号的函数，第一个参数为需要改变的引脚号，第二个参数为高低电平，参数为HIGH和LOW，对应数字信号的0与1，在使用这个函数的同时，对于单片机来说，传感器是将信号传入到单片机里，因此触碰开关的引脚是作为输入的，将使用的引脚定义为输入端口，需要在初始化中对其进行定义，定义代码如图所示：

|  |
| --- |
| pinMode(11, INPUT);//定义一个限位开关为输入设备  pinMode(12, INPUT);//定义另一个限位开关为输入设备 |

其中pinMode（）函数有两个参数，功能为定义引脚是输入还是输出端。第一个参数为需要定义的引脚号，第二个参数为需要定义为输入还是输出端，参数为INPUT和OUTPUT。其电路连线如图所示:



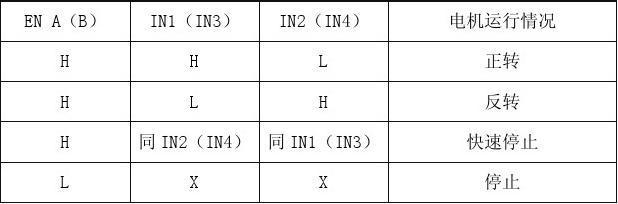
**图6-1 限位开关原理图**

6.2执行系统设计

执行机构是整个系统的重要部分，包括了L293D控制芯片，2路继电器，执行电机，12V变压器。执行系统是最常用的系统。控制原理如下：因为电机的驱动电压为12V，单片机的供电电压为5V，两者的电压不同必然导致不能使用同样的电压进行供电，因此驱动电机的电压需要外接。而弱电控制强电最主要的电器材料就是继电器，通过单片机给予的数字信号实现继电器的闭合，从而控制电机的正反转。电机的没有正负极之分，控制电机的正反转主要是电机的哪一级为正极，另一极为负极，即可实现转向，当反过来是，则实现电机的反转。因此，通过L293D芯片的逻辑，去控制2路继电器，当一端继电器常开，一端继电器常闭，实现电机的正转，反之则可实现电机的反转。实现电机的转动的代码如下所示：

|  |
| --- |
| digitalWrite(9,HIGH/LOW);  digitalWrite(10,HIGH/LOW); |

使用L293D芯片的真值表如下所示：



**图6-2 L293D芯片真值表**

单片机Pin9、Pin10分别与L293D芯片的INPUT1与INPUT2连接，对于单片机来说，控制执行是作为输出设备，同样的，我们可以使用digitalWrite()函数实现将引脚置为需要的状态，通过真值表，我们可以实现电机的正转、反转与快速停止状态，实现电机的正转可以有如下代码：

|  |
| --- |
| digitalWrite(9,HIGH);  digitalWrite(10,LOW); |

同理，实现电机的反转可以由如下代码实现：

|  |
| --- |
| digitalWrite(9,LOW);  digitalWrite(10,HIGH); |

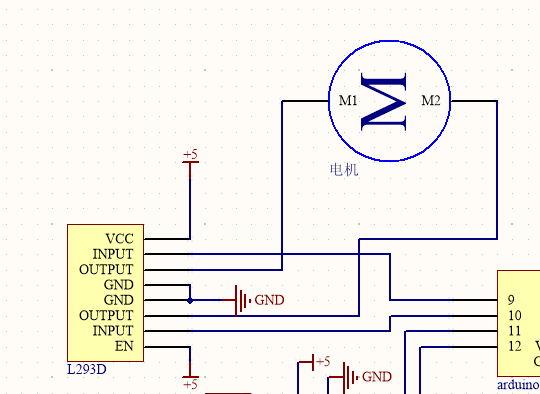
电机的停止代码如下所示：

|  |
| --- |
| digitalWrite(9,HIGH);  digitalWrite(10,HIGH); |

同样的我们需要对Pin9、Pin10进行初始化，而作为输出设备，需要初始化为输出，输出话如下所示：

|  |
| --- |
| pinMode(9, OUTPUT);//定义电机一引脚为输出端  pinMode(10, OUTPUT);//定义另一电机引脚为输出端 |

设计完程序，则需要设计硬件的线路连接与布局，布局如图所示：

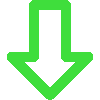
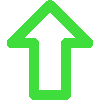


**图6-3 L293D芯片原理图**

6.3显示界面设计

6.3.1矢量图的制作

使用Photoshop软件制作如图所示的矢量图，制作方法为使用画笔工具绘制，并将颜色设置为较为有对比的颜色，将图片背景设置为透明，保存格式为\*.png格式。绘制如图所示图片：



**图6-4 矢量图片**

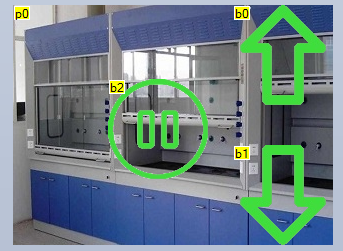
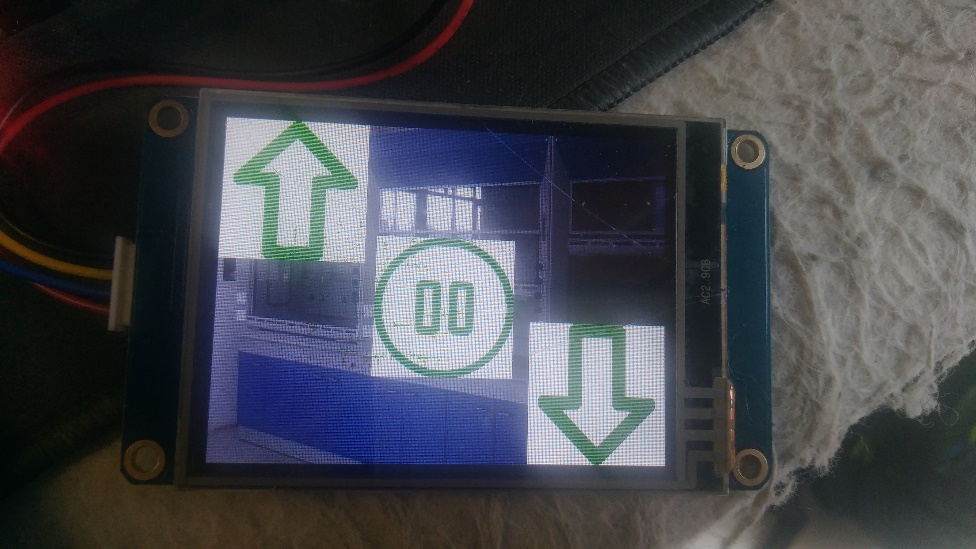
6.3.2背景图片的编辑

因为使用的触摸屏的分辨率为320x240的，如果图片分辨率过大不能显示完全，太小又会导致图片太糊，因此需要设置背景图片。设置背景图片如图所示：



**图6-5 背景图片**

具体设计界面如图所示：

**图6-6 界面布局及实景图片**

6.3.3数据的传递

数据的传递通过printh函数发送16进制的数据，当单片机收到信号时，能够对其执行器控制并反馈信息。当触摸屏触摸到停止按钮时，发送数据0；当触摸到上时，发送数据1；当触摸下时，发送数据2。单片机中如果收到对应的数据时，触发对应的操作。单片机中代码如下：

|  |
| --- |
| while (Serial.available() > 0) {//判断串口是否有信号输入，有则执行switch选择  switch (item) {//使用switch来判断收到什么信号执行怎样的操作  case '0'://信号0  //停止  digitalWrite(9,HIGH);  digitalWrite(10,HIGH);  break;  //电机左转  case '1':  digitalWrite(9,HIGH);  digitalWrite(10,LOW);  break;  //电机右转  case '2':  digitalWrite(9,LOW);  digitalWrite(10,HIGH);  break;  //收到干扰信号时，保证电机停止状态  default:  digitalWrite(9,HIGH);  digitalWrite(10,HIGH);  break;  } |

使用Serial.available()来判断串口是否收到信号，当大于0时，则表示有数据流进入，则开始进入while循环中，如果没有，则不会进入到while循环中。使用switch语句来判断收到了怎样的信号，通过不同的信号来执行不同的功能，使用item来存储收到的信号，使用Serial.read（）函数来读取串口数据，并将读取到的数据赋值给item，通过item的具体数据来执行不同功能。

|  |
| --- |
| Serial.begin(9600);//定义串口波特率为9600  volatile char item;//定义item存储读取信息  item = Serial.read(); //读取串口信息 |

调试结果如下所示：



**图6-7 反馈信号**

能够正确反馈具体的定义数据。

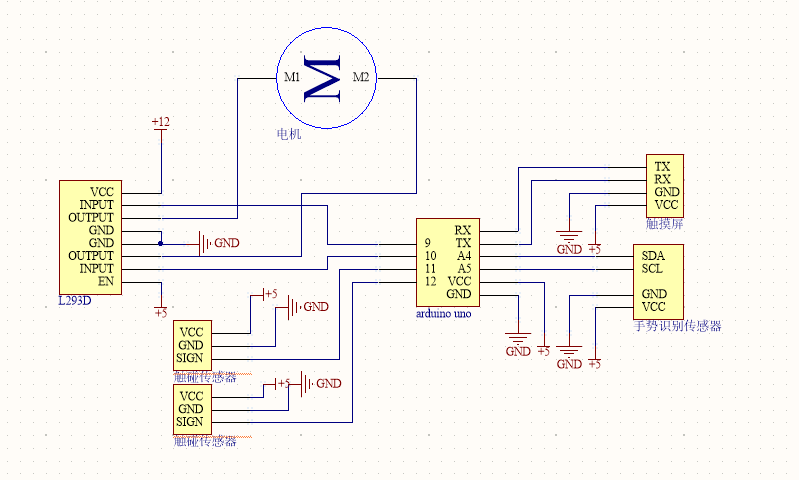
6.4手势识别识别系统设计

手势识别使用PAJ7620的库文件在PAJ7620库中可以看到其定义的读取规则如下

|  |
| --- |
| #define GES\_RIGHT\_FLAG PAJ7620\_VAL(1,0)//右  #define GES\_LEFT\_FLAG PAJ7620\_VAL(1,1)//左  #define GES\_UP\_FLAG PAJ7620\_VAL(1,2)//上  #define GES\_DOWN\_FLAG PAJ7620\_VAL(1,3)//下  #define GES\_FORWARD\_FLAG PAJ7620\_VAL(1,4)//前  #define GES\_BACKWARD\_FLAG PAJ7620\_VAL(1,5)//后  #define GES\_CLOCKWISE\_FLAG PAJ7620\_VAL(1,6)//顺时针  #define GES\_COUNT\_CLOCKWISE\_FLAG PAJ7620\_VAL(1,7) //逆时针  #define GES\_WAVE\_FLAG PAJ7620\_VAL(1,0)//摇摆 |

6.5总系统设计与布局

综上所示，设计如图所示电器电路原理图：



**图6-8 总系统原理图**

供电系统分为强电驱动和弱电驱动系统。强电带动电机驱动，且不会干扰到单片机的正常运行，单片机通过L293D芯片赋予工作逻辑，并通过继电器将信号以强电的形式控制电机；触碰传感器作为外设反馈将极限位置信息反馈至单片机中，并通过单片机收到的信号对电机做出正确的控制，触摸屏作为输入设备，可以起到按钮的作用，且优化了其界面，相对于单纯的按钮来说更加优美且方便；手势识别作为其系统重要部件，起到用手势去控制通风柜的运动，是整个系统的重中之重和亮点。

**7 总结与展望**

7.1本文的工作总结

本文通过使用Arduino作为控制器，电机作为执行器，触碰开关、手势识别作为传感器、触摸屏作为显示器，完成了手势识别去控制智能通风柜的操作。

本文的主要工作主要包括以下内容：

（1）对人机交互、手势识别的定义做了简单介绍，并阐述了手势识别的发展历史和当前研究情况。

（2）介绍了数字信号、模拟信号、单片机通讯、通讯方式等基本知识。

（3）分析了PAJ7620和MARS手势识别模块的优劣，并确定了在之后使用的手势识别模块。

（4）学习了使用Arduino单片机，并且能够将得到的信号作为触发信号对执行器进行控制。

（5）了解了单片机的通信协议与通信接口，能够将触摸屏、手势识别传感器、单片机、电机、触碰传感器有机地结合起来，实现特定的功能。

（6）能够自主的使用Altiun Designer软件设计系统的电路原理图，对之后的创新提供了思路。

7.2展望

本文对基于Arduino单片机的手势识别技术进行了一定的设计与研究，对一些错误的触发信号进行了一些避免，能够正确处理动态信号，但是本文还是存在一定的缺陷，需要在之后的研究中进一步解决。

（1）反馈时间较长，不能够及时相应操作。

（2）对于较远距离，不能够正确识别对应手势。

（3）线路搭建不是很完善，有很大的改进空间。

作为一种便捷、自然、人性化的交互方式，手势在人机交互中有着非常广泛的应用前景，也因此成为国内外广大学者深入研究的对象。自然手势交互作为一种新的技术给人机交互方面的工作带来了极大的便利，而且这种技术也逐渐被应用到生活中去，如应用于手机、PDA、遥控器等各种嵌入式移动设备中。而深度摄像技术的发展和普及化将有力提高手势识别的能力，从而促进自然手势交互的进一步发展。

我希望我能够朝着这个方面进行研究，不仅是因为我的兴趣使然，同时也是我希望我能在这个方面能够发挥自己的余热，让更多的人能够了解这个方面，同时为以后的研究起抛砖引玉的作用。

**参考文献**

[1]. 蒋有绪,郭泉水,马娟,等．远程教育概论[M]．北京：中央广播电视大学出版社,2001.49-53

[1]付潇聪.基于视觉的手势识别及其应用研究[D].南京理工大学,2017

[2]刘奇琴.基于视觉的手势识别技术研究[D].大连理工大学,2014.

[3]冯志全,蒋彥.手势识别研究综述[J].济南大学学报：自然科学版2013(4):336-341.

[4] William T.F.,Craig D.W.. Television Control by Hand Gesture[J]. International Workshop on Automation Face and Gesture Recognition of IEEE,1995,(1):1-7

[5] Howe L.W.,Farah W.,Ali C.. Comparison of Hand Segmentation Methodologies for Hand Gesture Recognition[J]. In Proceedings of International Symposium on Information Technology,2008,2(1):1-7

[6] Elmezain M.,Hamadi A.A.,Appenrodt J.,Michaelis B.. A Hidden Markov Model-Based Isolated and Meaningful Hand Gesture Recognition[J]. International Journal of Electrical,2009,(1):156-163.

[7]张秋余,姚开博,基于矩形特征和改进 Adaboost 算法的手势检测方法[J]. 计算机工程,2008,1(14):176-178.

[8]单彩峰. 用于人机交互的视觉手势识别[D]. 北京:中国科学技术大学,2001

[9]王兆安,刘进军.电力电子技术（第五版）：机械工业出版社,2011

[10]杨毅明,数字信号处理.北京：机械工业出版社,2013：1-30,59

[11]刘茜.基于PLC与单片机通讯的概述[J].科技与创新,2018(23):66-67.

[12]周皓瑗,雷维嘉.信息和能量同传全双工中继信道的物理层安全方案[J].重庆邮电大学学报(自然科学版),2019,31(02):221-230.

[13]陈静.基于Kinect的手势识别技术及其在教学中的应用[D].上海交通大学, 2013.

[14]邬大鹏.基于视觉的手势识别及人机交互研究[D].南京航空航天大学, 2010.

[15]郭子雷.基于计算机视觉的手势识别系统的设计与实现[D].华中科技大学, 2016.

**致 谢**

时间过的飞快，转眼间我就到了快毕业的时间了。回想起我刚进大学时的情景，校园变得更美了，处处都充满着便利。曾经的自己和现在比起来，我变得更加成熟，做事也学会了冷静分析，独立思考，再也不是那个莽莽撞撞、鹤立独行的野蛮小孩了。在这四年里，我学习了很多有关机械的东西，很多东西对于我之后的生活具有积极的指导作用，我也在这四年里，学会了为人处世，喜欢上了演讲，对生活更加的自信，也喜欢将自己所学所想所思与他人分享，鼓励别人做自己喜欢的事。

首先，我最想感谢的是我的父母及我的亲人，他们在我最无助最孤单的时候给予我温暖，让我能够在这世间感受爱的滋味，他们教会了我如何爱别人，也教会了我如何被别人爱，如何付出又如何收获，人间世道，处世之理。感谢父母尊重我的选择，让我能够来到上海这座城市，感受魔幻又现代的发展氛围，开阔了自己的视野，也让自己有了奋斗的目标。

其次，我要感谢我自己，不枉四年的学习，虽然在四年里我也有过迷茫，有时甚至不知道人生的意义是什么。但是感谢自己，没有放弃希望，没有放弃生命，感谢自己找到了自己的兴趣爱好，能够为自己的喜爱的事情奋斗，找到了一份与兴趣爱好相投合的工作，至少工作时不是痛苦的，感谢自己逐渐地变好，变得更加成熟与稳健，能够有男人地担当。

同时，我要感谢所有教过我的老师以及我的指导老师曾祥绪老师，没有他们课上的悉心教导，没有他们的循循善诱，知识也不能萦绕着我，让我感受到科技的魅力，感谢我的指导老师曾祥绪老师，每次为我遇到的问题提供解决方案，提供我解决问题的思路，提供实验场所，为我能够顺利完成毕业论文打下了坚实的基础。

也感谢我的同学，我所认识的人，是你们帮助我成长，陪伴我，给予我鼓励。感谢我的同学陈沛宇，带我参与了很多科创项目，参加了很多的比赛，让我将所学知识投入到具体问题上，让我有了解决具体问题的能力，也感谢启发我的书籍作者和论坛作者，谢谢他们默默的付出，授之以渔。

最后，感谢所有人，未来一起加油！

**附页**

|  |
| --- |
| #include <Wire.h>//使用wire.h的库  #include "paj7620.h"//使用额外拓展的Paj7620手势识别的库  #define GES\_REACTION\_TIME 800//设置反应延时800ms  #define GES\_QUIT\_TIME 1000//设置退出延时1000ms  volatile char item;//定义item存储读取信息  //初始化  void setup()  {  Serial.begin(9600);//定义串口波特率为9600  pinMode(11, INPUT);//定义一个限位开关为输入设备  pinMode(12, INPUT);//定义另一个限位开关为输入设备  pinMode(9, OUTPUT);//定义电机一引脚为输出端  pinMode(10, OUTPUT);//定义另一电机引脚为输出端  item = Serial.read();//读取串口信息  }  //循环体  void loop()  {  uint8\_t data = 0, data1 = 0, error; //定义数据data，data1，和error  error = paj7620ReadReg(0x43, 1, &data);//读取0x43位置的手势结果  if (!error) //如果没有错误，执行  {  if(data == GES\_UP\_FLAG)//信号为上挥动  {  delay(GES\_REACTION\_TIME);//延时800ms反应时间  paj7620ReadReg(0x43, 1, &data);//赋值  //执行上升操作  Serial.println("Up");//打印up  digitalWrite(9,HIGH);//电机正转  digitalWrite(10,LOW);//电机正转  }  else if(data == GES\_DOWN\_FLAG)//信号为下挥动  {  delay(GES\_REACTION\_TIME);//延时800ms反应时间  paj7620ReadReg(0x43, 1, &data);//赋值  //执行下降操作  Serial.println("Down");//打印down  digitalWrite(9,LOW);//电机反转  digitalWrite(10,HIGH);//电机反转  }  else  {  Serial.println("else sign");//打印down  digitalWrite(9,HIGH);//电机反转  digitalWrite(10,HIGH);//电机反转  }  delay(GES\_QUIT\_TIME);//延时1000ms  }  if (digitalRead(11) == 0)//如果限位开关被触发  {  digitalWrite(9,HIGH);//电机停止  digitalWrite(10,HIGH);//电机停止  }  if (digitalRead(12) == 0)//如果限位开关被触发  {  digitalWrite(9,HIGH);//电机停止  digitalWrite(10,HIGH);//电机停止  }  while (Serial.available() > 0) {//判断串口是否有信号输入，有则执行switch选择  switch (item) {//使用switch来判断收到什么信号执行怎样的操作  case '0'://信号0  //停止  digitalWrite(9,HIGH);//电机停止  digitalWrite(10,HIGH);//电机停止  break;  //电机左转  case '1':  digitalWrite(9,HIGH);//电机左转  digitalWrite(10,LOW);//电机左转  break;  //电机右转  case '2':  digitalWrite(9,LOW);//电机右转  digitalWrite(10,HIGH);//电机右转  break;  //收到干扰信号时，保证电机停止状态  default:  digitalWrite(9,HIGH);//电机停止  digitalWrite(10,HIGH);//电机停止  break;  }  }  } |