基于AVR单片机的简易风洞PID稳定系统

张振尧 国机1901 2018110071

# 一、系统的总体架构及功能描述

**1.1 系统总体方案设计**

系统以ATMEGA328P-PU单片机为核心模块，通过电机驱动模块带动扇叶，向圆管之内鼓风，将容器吹起。通过激光测距模块将容器在圆管之内的位置检测出来之后返回位置数据给单片机，通过单片机的判断，控制风力的大小，将容器控制在所要求的位置，系统面包板设计及初步连线如图1所示（使用fritzing绘制）。

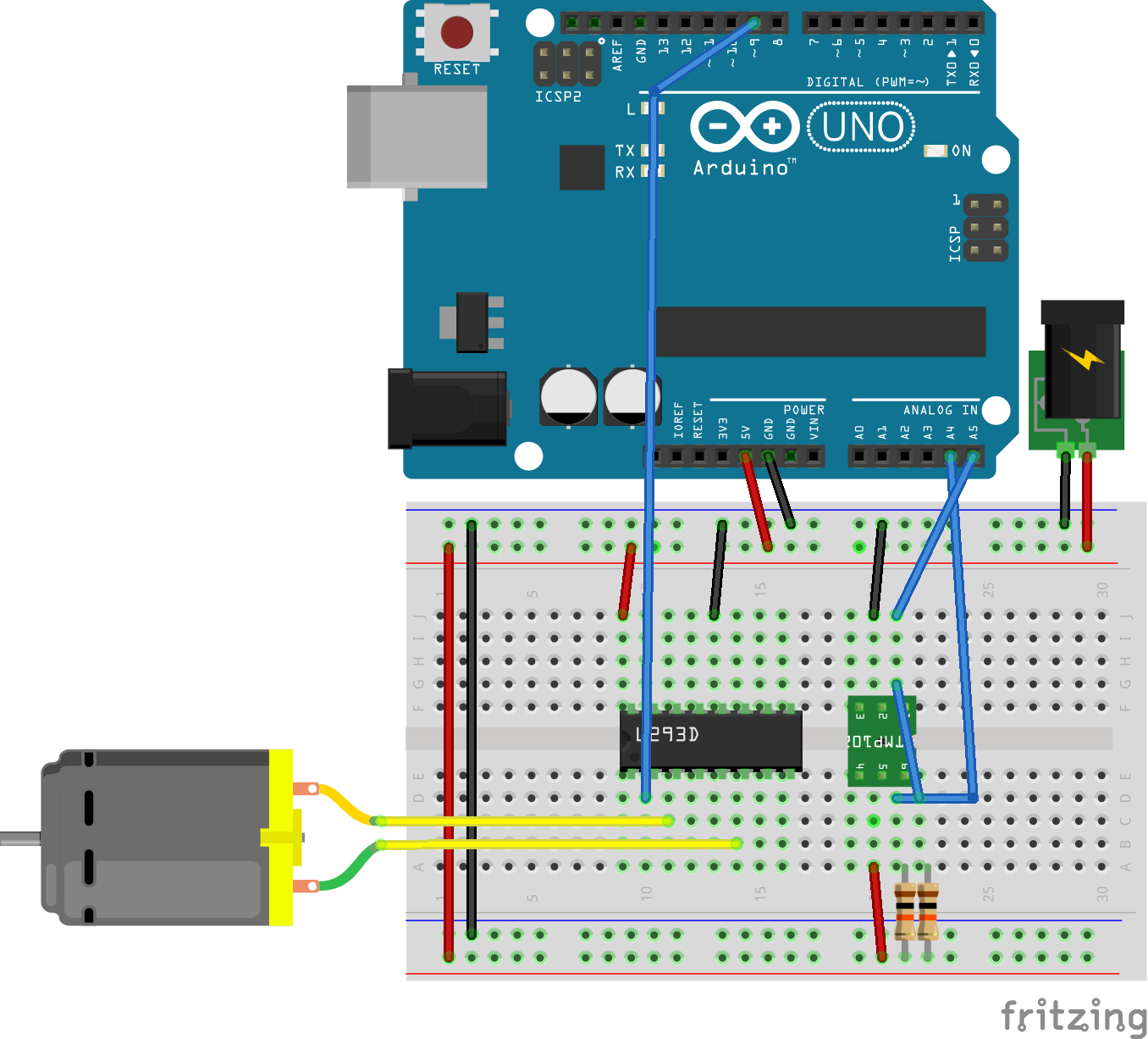


图1 系统面包板设计

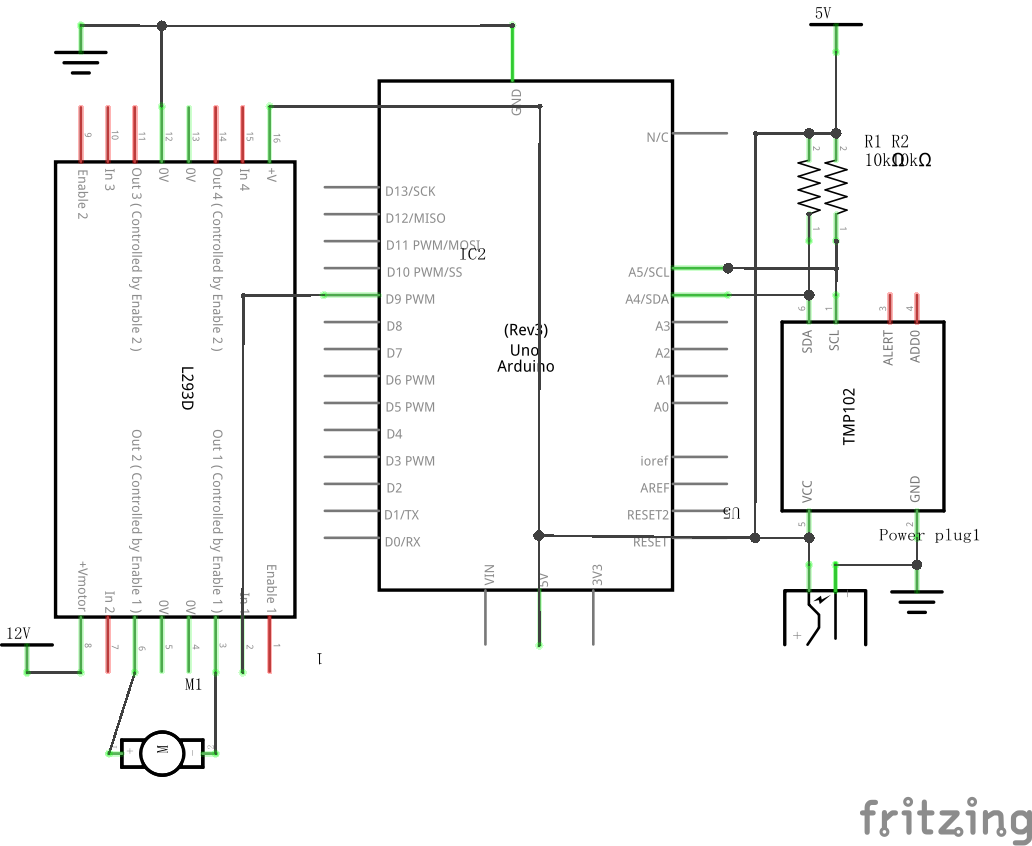


图2 系统初步连线

**1.2 设计方案论证**

1.2.1 电机驱动模块

系统采用无刷电机ESC电子调速器驱动模块。ESC具有三相输出，可以驱动一个直流电机。且该模块的驱动电流最大可以达到10 A。可以满足电机的驱动要求，还可以在输入端加上PWM方波控制转速以及设定刹车，可以对电机转速进行精准控制。

1.2.2 定位模块

VL53L0X激光测距模块的检测距离长，可达到10m，而且该模块的反应速度也较快。可以根据自己所要求的时间来具体调节发射和接收时间，操作较为方便。另一方面，激光检测方式具有模块的操作简单，不足在于在垂直距离较长情况下，需要进行中心对准才能精准测距。

# 二、硬件设计及接口电路图

**2.1 涵道风机的确定**

在系统中，需要将容器在圆管之内上下吹动，这就会对风机的要求很高。普通的小风扇无法满足要求。所以选择了高速无刷电机，采用大PVC管和竖直圆筒，并结合鼓风机结构，将风吸入风道之后，经过一圈回流之后送入圆管。这样无刷涵道风机结构简单，稳定度高，较为可靠。图3为涵道风机示意图。

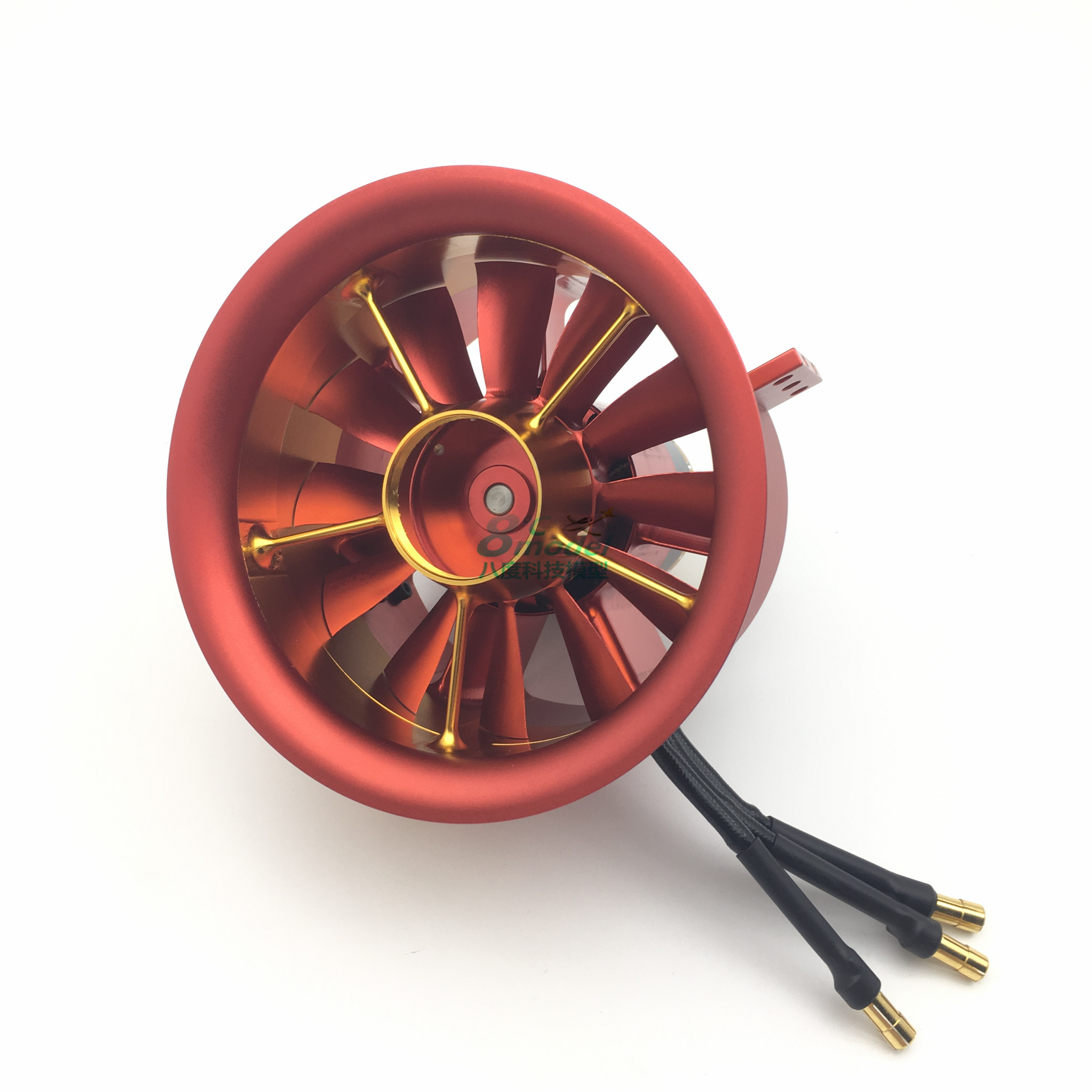


图3 涵道风机示意图

**2.2 电机驱动电路设计**

在输入信号之后，ESC电子调速器将输入的PWM波转换为电流电压相对较大的PWM信号。其信号保持一致或者取反。输出之后可以达到利用小电压小电流驱动大电压大电流的作用。驱动选择了常用的高电压大电流的ESC电子调速器模块，其动态响应能力较好，还可以实现频繁的无极快速启动，制动和反转等优点，电机驱动电路及单片机系统原理图如图4所示（使用立创EDA在线绘制），在绘制过程中，参考了现有的UNO单片机以及无刷电机库文件。

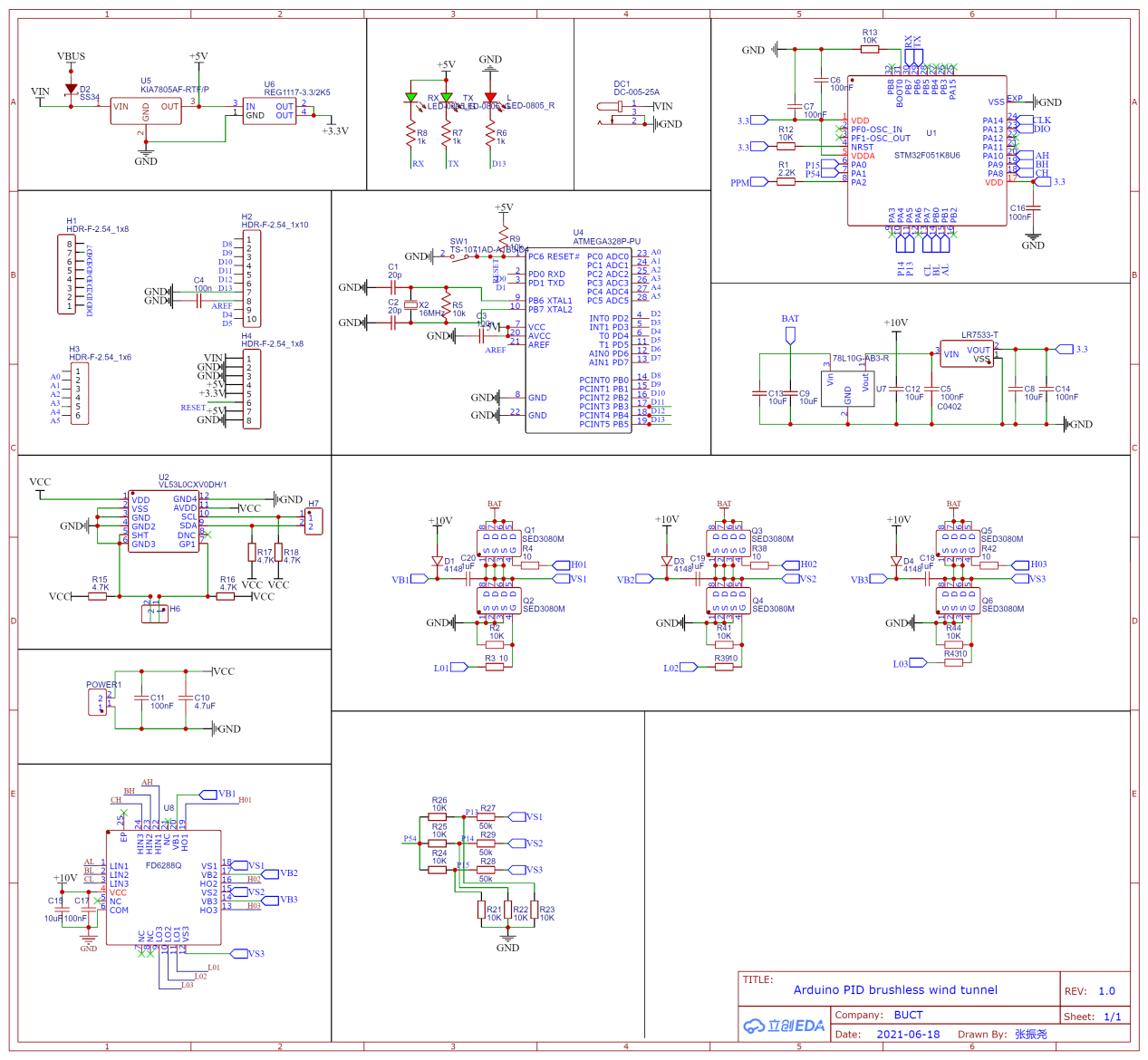


图4 电机驱动及单片机系统整体原理图

# 三、软件设计思路及部分代码

**3.1 激光测距检测距离计算**

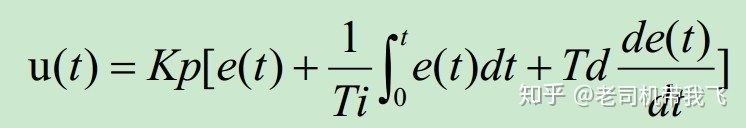
根据激光测距原理，激光测距在空气之中传播遇到障碍物的时候就会马上返回。假设激光测距的传播速度为V，发射和接收的时间差为t，就可以计算出发射点距离障碍物的距离S，如公式(1)所示。

S=V\*△t/2 (1)

根据系统之中的激光测距模块的特点，通过计算可以得出在10 cm到4.5 m之内的距离测量值。

**3.2 电机控制算法的确定**

对于无刷电机转速和容器的运动过程有着严格的要求。所以在控制小球的时候需要进行精确控制，通过选择适当的PID算法和有效的参数正定可以达到高精度控制目的。PID公式如图)所示。



其中：  
Kp -> 控制器的比例系数  
Ti -> 控制器的积分时间，也称积分系数  
Td -> 控制器的微分时间，也称微分系数

根据上述公式，在最开始确定比例系数之后就开始确定微分器和积分器的数值。最终将三个系数写入程序之中，达到稳定。

在实际程序中，体现PID功能的代码如下：

error = distance - desired; //误差计算

pid\_p = kp \* error; //P值计算

if (-8 < error && error < 8) //如果不在8mm误差距离范围内，不启用I控制

pid\_i = pid\_i + (ki \* error);//I值计算

pid\_d = kd \* ((error - previous\_error) / elapsedTime); //D值计算

previous\_error = error; //保存当前误差为上一个误差

PID = pid\_p + pid\_i + pid\_d; //计算PID总值

**3.3 程序开发平台配置**

本程序开发使用PlatformIO综合开发平台，配置文件及具体包含库文件如下

[platformio]

default\_envs = uno //单片机开发环境选择uno

[env:uno]

platform = atmelavr

board = uno //单片机厂商及型号确定

framework = arduino //单片机固件确定

upload\_port = COM[3] //单片机端口号设定

lib\_deps =

//包含了所用到的库文件

    arduino-libraries/Servo@^1.1.8

    adafruit/Adafruit\_VL53L0X@^1.1.1

    adafruit/Adafruit GFX Library@^1.10.10

    adafruit/Adafruit VEML6070 Library@^1.0.6

    adafruit/Adafruit BusIO@^1.7.5

    marlinfirmware/U8glib-HAL@^0.4.5

[smougenot/Adafruit\_VEML6070@0.0.0-alpha+sha.f56ccf3f85](mailto:smougenot/Adafruit_VEML6070@0.0.0-alpha+sha.f56ccf3f85)

**3.3 具体代码及注释**

#include <Arduino.h>

#include <SPI.h> //SPI通信库，用于和激光测距模块通讯

#include <Wire.h>

#include <Servo.h> //SERVO舵机控制程序库

#include "Adafruit\_VL53L0X.h" //激光测距模块库

float desired = 200; //设定预期容器高度

int initalspeed = 1300; //设定风机初始转速，通过调整这个参数，可以使风机一开始就运行在较稳定的状态

double kp = 3.55;  //3.55

double ki = 0.005; //0.003

double kd = 2.55;  //2.05 //设定PID初始控制值

//设定一些变量及初始值

float elapsedTime, time, timePrev, PID, error;

float previous\_error = 0;

float pid\_p = 0;

float pid\_i = 0;

float pid\_d = 0;

int distance;

//设定low和high的值，在后面的判断程序中使PID输出值不超出1000-2000范围

int low = initalspeed - 1000;

int high = 2000 - initalspeed - 200;

设定PWM信号输出的引脚号为9（UNO可用的pwm输出引脚并不多，需要单独进行设定）

#define MOTOR\_PIN 9

Servo motor; //利用SERVO库定义一个motor电机变量

//利用VL53L0X库定义一个lox和measure结构体来存储距离数据包

Adafruit\_VL53L0X lox = Adafruit\_VL53L0X();

VL53L0X\_RangingMeasurementData\_t measure;

void setup()

{

//将用于调试的串口通讯的波特率设定为250000

Serial.begin(250000);

//将PWM输出脚附加到motor变量上，设定输出范围为1000, 2000

motor.attach(9, 1000, 2000);

//获取当前时间，存储到time变量中

time = millis();

//写油门最低值

motor.writeMicroseconds(1000);

    if (!lox.begin())

        while (1)

{

//如果激光测距模块没有正常启动，在串口中输出报错

            Serial.println(F("Failed to boot VL53L0X sensor"));

delay(3000); //防抖功能

}

void loop() //循环执行函数

{

    timePrev = time; //定义上一个时间值

    time = millis(); //获取当前时间值

    elapsedTime = (time - timePrev) / 1000; //计算变换时间值

    lox.rangingTest(&measure, false); //初始化测距模块

    if (measure.RangeStatus != 4) //如果正确获得到距离数据包

        distance = measure.RangeMilliMeter; //存储距离变量到distance中

    else

        while (1) //如果没有正常初始化测距模块，就在串口中报错

            Serial.println(F("Failed to measure"));

Serial.print(measure.RangeMilliMeter); //输出测量到的距离值

//PID程序

    error = distance - desired; //误差计算

    pid\_p = kp \* error; //P值计算

    if (-8 < error && error < 8) //如果不在8mm误差距离范围内，不启用I控制

        pid\_i = pid\_i + (ki \* error);//I值计算

    pid\_d = kd \* ((error - previous\_error) / elapsedTime); //D值计算

    previous\_error = error; //保存当前误差为上一个误差

PID = pid\_p + pid\_i + pid\_d; //计算PID总值

    Serial.print(" ");

Serial.println(PID);

//下面这段程序，用于控制PID输出值low和high范围内，使最终输出值在1000-2000范围内

    if (PID < -low)

        PID = -low;

    if (PID > high)

        PID = high;

//加上初始设定速度就是最终输出速度

PID = PID + initalspeed;

//利用SERVO库写PID电机转速

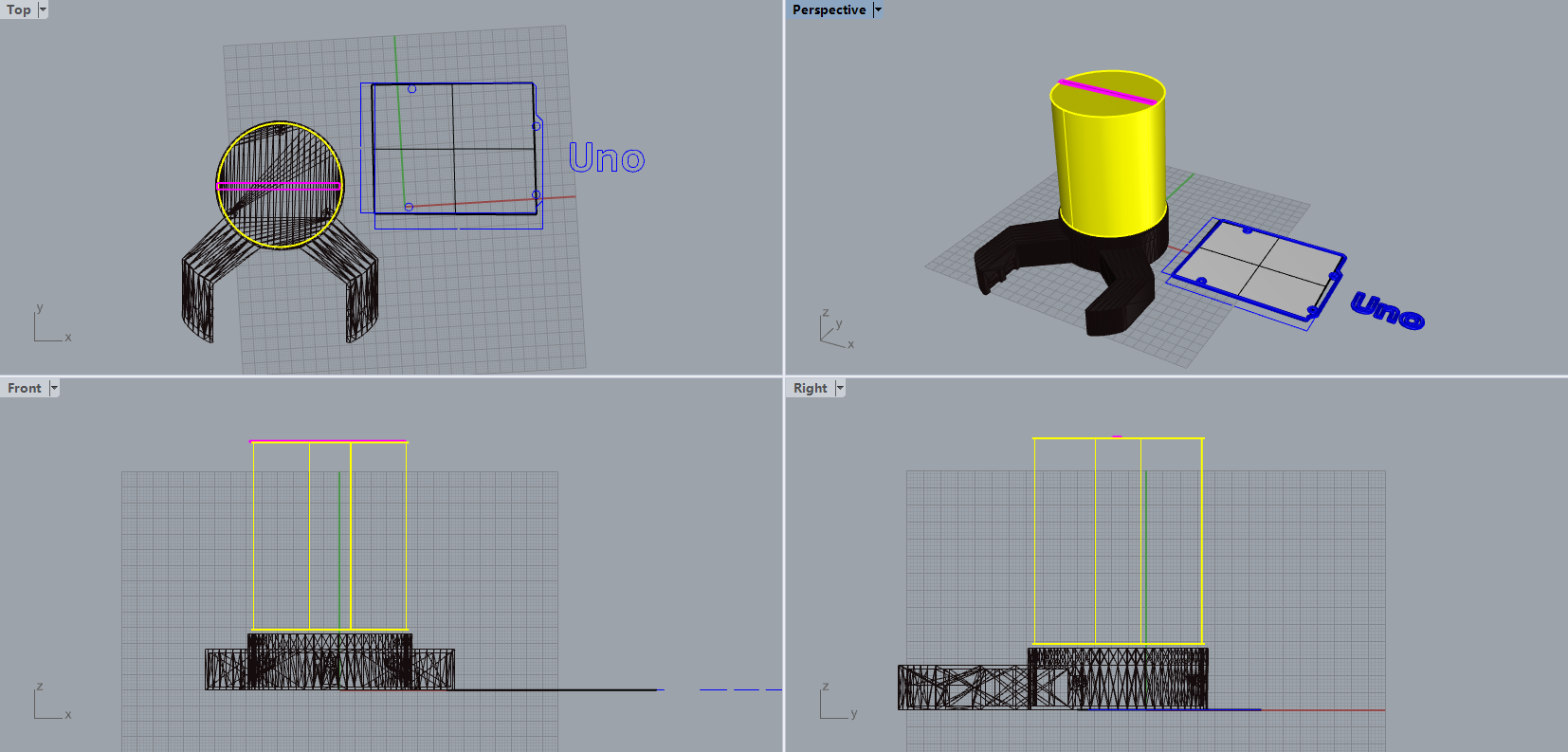
    motor.writeMicroseconds(PID);

}

# 四、测试方案与测试结果

**4.1 系统搭建方法及测试数据**

按图5所示风洞结构图（使用犀牛软件进行建模渲染），进行系统搭建，然后在PVC管内加入容器测试。



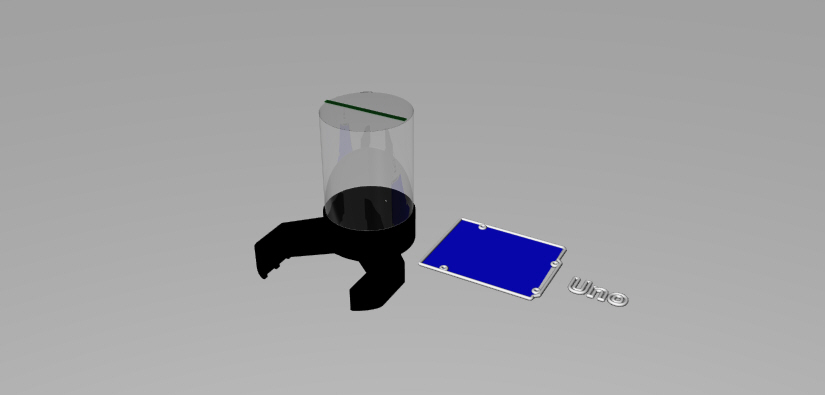


图5 风洞系统建模结构图

**4.2 测试结果**

经过一系列测试表明整个系统能够完全达到要求。实际运行过程中，因为设定了涵道风机的转速区间，所以对容器内配重进行了调整。整个系统经过测试，工作稳定，精度高。系统实物如图6所示。



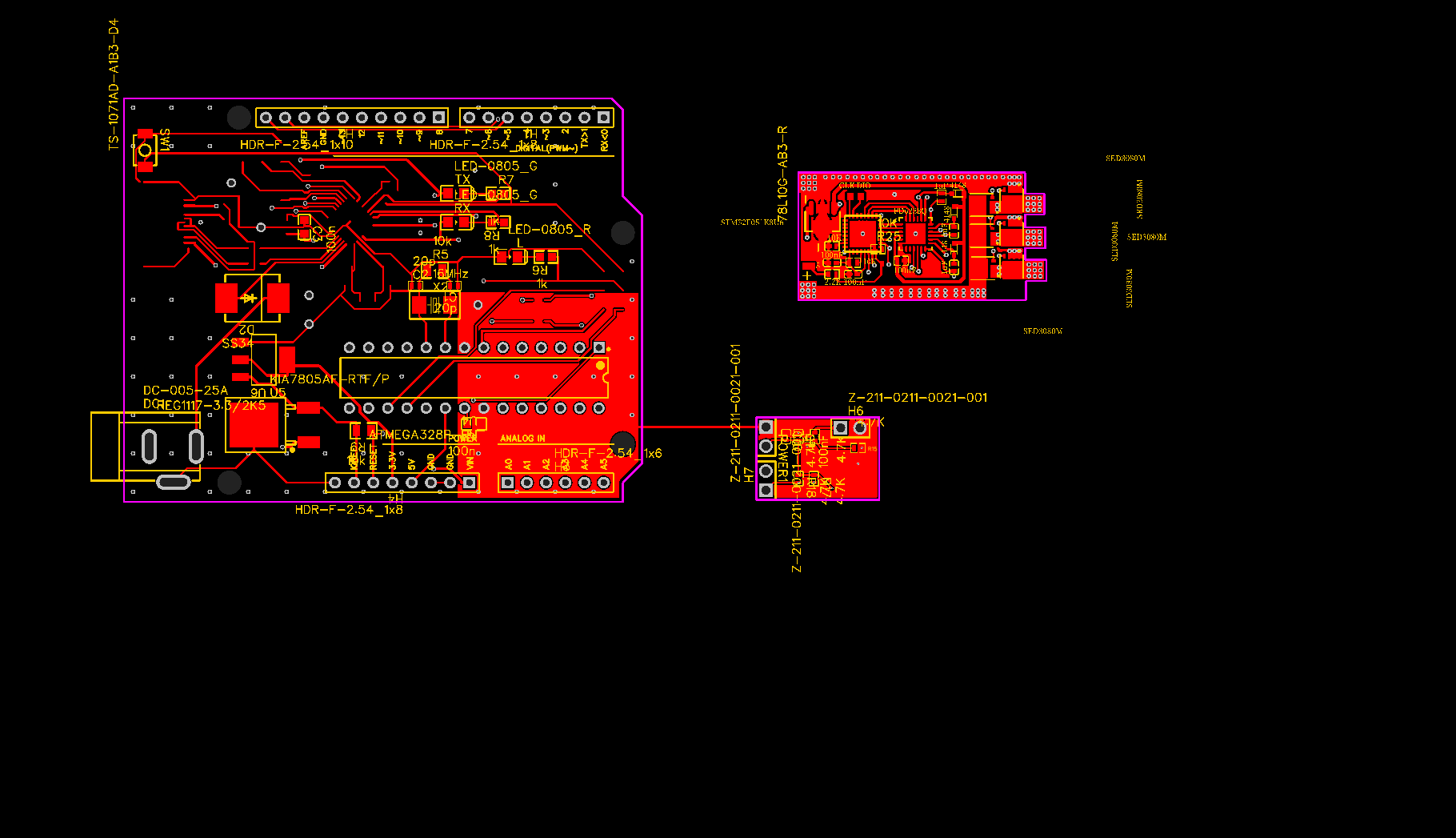
图6 系统实物图

# 五、课程小结

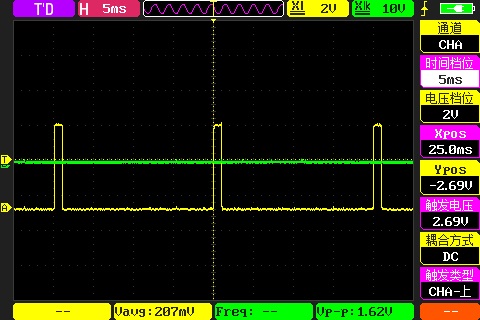
通过单片机入门课程的学习，我了解和掌握了TIVA和51系列单片机的基本功能和使用，对于自动控制原理的理解和运用有极大意义。我通过单片机实验的课程学习到了先进行系统构思，再进行原理图设计，然后在计算机上进行仿真，最后再实物测试这样的一个；流程，其中电脑仿真的过程调试速度快，反应精准，大大加快了开发的进度。我认为，只要精通了一种单片机，掌握了这样一套设计流程，再学习其他单片机是非常快的。

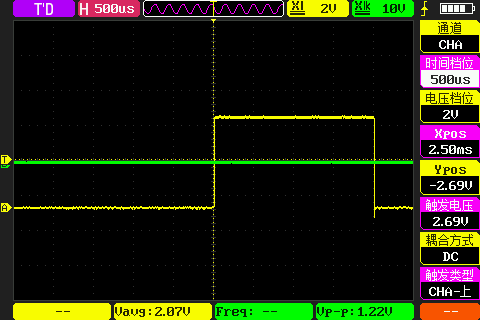
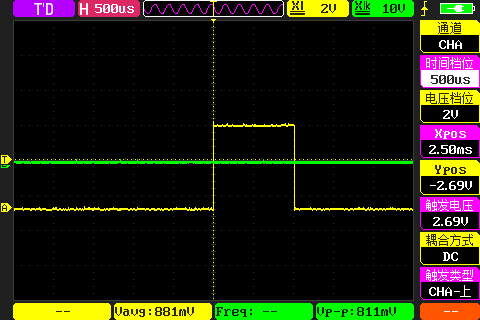
**附录**

**附录1 电路PCB布局**

****

**附录2 利用示波器测量输出PWM脉冲，最小脉冲及最大脉冲输出值**

**、**

****

**附录3 电路芯片BOM表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Name | Designator | Footprint | Quantity | Manufacturer Part | Manufacturer | Price |
| 1 | 20p | C1,C2 | C0402 | 2 |  |  |  |
| 2 | 100n | C3,C4 | C0603 | 2 |  |  |  |
| 3 | 100nF | C5 | C0402 | 1 | CT41G-0603-2X1-50V-0.01uF-K(N) | TORCH | 0.0382 |
| 4 | 100nF | C6,C7,C14 | C0402 | 3 | 0603F104M500 | WTC | 0.0375 |
| 5 | 10uF | C8,C9,C12,C13,C15 | C1206 | 5 | CL31A106KBHNNNE | SAMSUNG | 0.2715 |
| 6 | 4.7uF | C10 | C0402 | 1 | JMK105BBJ475MV-F | TAIYO YUDEN | 0.0346 |
| 7 | 100nF | C11 | C0402 | 1 | CL05B104KO5NNNC | SAMSUNG | 0.006 |
| 8 | 100nF | C16,C17 | C0402 | 2 | 0402B104K500 | WTC | 0.0214 |
| 9 | 1uF | C18,C19,C20 | C0402 | 3 | 0402B104K500 | WTC | 0.0214 |
| 10 | 4148 | D1,D3,D4 | SOD-523\_L1.2-W0.8-LS1.6-RD | 3 | MMBD4148TS\_R1\_00001 | PANJIT International | 0.1225 |
| 11 | SS34 | D2 | SMC\_L7.1-W6.2-LS8.1-RD | 1 | SS34 | MDD | 0.6221 |
| 12 | DC-005-25A | DC1 | DC-IN-TH\_DC005 | 1 | DC-005-25A | HRO | 0.5277 |
| 13 | HDR-F-2.54\_1x8 | H1,H4 | HDR-F-2.54\_1X8 | 2 |  |  | 0.9446 |
| 14 | HDR-F-2.54\_1x10 | H2 | HDR-F-2.54\_1X10 | 1 |  |  | 0.3593 |
| 15 | HDR-F-2.54\_1x6 | H3 | HDR-F-2.54\_1X6 | 1 |  |  | 0.3791 |
| 16 | Z-211-0211-0021-001 | H6,H7,POWER1 | HDR-TH\_2P-P2.54-V | 3 | Z-211-0211-0021-001 | ZHENGLING | 0.121 |
| 17 | LED-0805\_R | L | LED0805\_RED | 1 | 17-21SURC/S530-A3/TR8 | EVERLIGHT(台湾亿光) | 0.085 |
| 18 | SED3080M | Q1,Q2,Q3,Q4,Q5,Q6 | DFN-8\_L3.0-W3.0-P0.65-BL | 6 | SED3080M | SINO-IC | 1.2092 |
| 19 | 2.2K | R1 | R0402 | 1 | RTT0210R7FTH | RALEC | 0.0238 |
| 20 | 10K | R2,R41,R44 | R0402 | 3 | AC0402JR-0710KL | YAGEO | 0.0062 |
| 21 | 10 | R3,R4,R38,R39,R42,R43 | R0402 | 6 | AC0402JR-0710KL | YAGEO | 0.0062 |
| 22 | 10k | R5,R9 | R0805 | 2 |  |  |  |
| 23 | 1k | R6,R7,R8 | R0603 | 3 |  |  |  |
| 24 | 10K | R12,R13 | R0402 | 2 | RTT0210R7FTH | RALEC | 0.0238 |
| 25 | 4.7K | R15,R16,R17,R18 | R0402 | 4 | 0402WGF4701TCE | UniOhm | 0.004 |
| 26 | 10K | R21,R22,R23,R24,R25,R26 | R0402 | 6 | RS-05K470JT | FH | 0.0131 |
| 27 | 50k | R27,R28,R29 | R0402 | 3 | AC0402JR-0710KL | YAGEO | 0.0062 |
| 28 | LED-0805\_G | RX,TX | LED0805\_GREEN | 2 | NCD0805G1 | 国星光电 | 0.13 |
| 29 | TS-1071AD-A1B3-D4 | SW1 | SW-SMD\_L4.0-W3.0-LS5.0 | 1 | TS-1071AD-A1B3-D4 | Yuandi | 0.4124 |
| 30 | STM32F051K8U6 | U1 | UFQFPN-32\_L5.0-W5.0-P0.50-TL | 1 | STM32F051K8U6 | STMicroelectronics | 41.84 |
| 31 | VL53L0CXV0DH/1 | U2 | OPTO-SMD\_12P\_VL53L0CXV0DH/1 | 1 | VL53L0CXV0DH/1 | STMicroelectronics | 20.48 |
| 32 | LR7533-T | U3 | SOT-23-3\_L2.9-W1.6-P1.90-LS2.8-BR | 1 | LR7533-T | LR | 0.2725 |
| 33 | ATMEGA328P-PU | U4 | DIP-28\_L34.6-W7.3-P2.54-LS10.2-BL | 1 | ATMEGA328P-PU | MICROCHIP | 16.13 |
| 34 | KIA7805AF-RTF/P | U5 | TO-252-2\_L6.6-W6.1-P4.57-LS9.9-BR-CW | 1 | KIA7805AF-RTF/P | KEC | 1.725 |
| 35 | REG1117-3.3/2K5 | U6 | SOT-223\_L6.5-W3.5-P2.30-LS7.0-BR | 1 | REG1117-3.3/2K5 | TI | 5.4 |
| 36 | 78L10G-AB3-R | U7 | SOT-89-3\_L4.5-W2.5-P1.50-LS4.2-BR | 1 | 78L10G-AB3-R | UTC | 0.5633 |
| 37 | FD6288Q | U8 | QFN-24\_L4.0-W4.0-P0.50-BL-EP2.6 | 1 | FD6288Q | Fortior Tech | 4.82 |
| 38 | 16MHz | X2 | OSC-SMD\_L5.0-W3.2 | 1 | X503216MSB2GI | YXC | 1.4707 |

**附录4 项目文件地址**

本项目文件开发过程中的建模，代码，原理图设计均已开源上传至github

地址：<https://github.com/00make/wind-tunnel>