**龙芯3D打印机设计文档**

V1.0

## 文档控制页

|  |  |  |  |
| --- | --- | --- | --- |
| **文档历史记录** | | | |
| 版本 | 日期 | 姓名 | 变更说明 |
| V1.0 | 2017-01-07 | chinesebear | 建立文档 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[文档控制页 2](#_Toc471755114)

[1 概述 4](#_Toc471755115)

[2 运动结构 4](#_Toc471755116)

[3 主控板 4](#_Toc471755117)

[3.1 龙芯1C 4](#_Toc471755118)

[3.2 操作系统RT-Thread 5](#_Toc471755119)

[3.3 控制程序Marlin 5](#_Toc471755120)

## 1 概述

3D打印机多种多样，大概可以分为生物3D打印机、金属3D打印机、塑料3D打印机等。塑料3D打印机又可分为FDM（堆叠）、SLA（光固化）、SLS（尼龙激光烧结）等。FDM 3D打印机又大概分为delta、xyz等。delta 3D的打印机，俗称三角洲打印机，此类开源了的3D打印机，国内比较出名的是kossel800。这款开源的3D打印机包括运动结构、主控板Arduino、下位机固件Marlin、上位机切片软件Cure等。

龙芯3D打印机也是基于开源技术的FDM delta打印机，主控芯片采用龙芯1C。主控芯片的操作系统采用国产硬实时操作系统RT-Thread，控制程序移植Marlin。

## 2 运动结构

运动结构包括一副机架、四个步进电机、一个电源、一个挤出头。

## 3 主控板

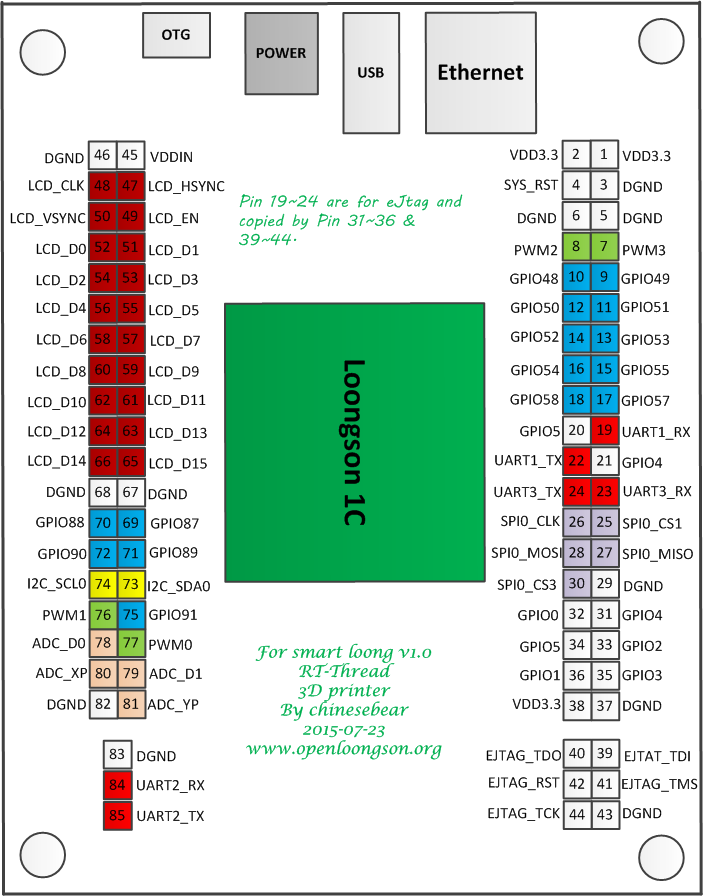


### 3.1 主控芯片龙芯1C

主控基于龙芯1C300A芯片的智龙开发板。龙芯1C300A有2个SPI、4路ADC、4个PWM。

一个三角洲FDM 3D打印机有四个步进电机、两个温度传感器、两个加热器（一个是打印头的，一个是热床的）、两个定位开关需要驱动。一个步进电机至少需要一个PWM，3个GPIO，也就是说四个电机，需要4个PWM和12个GPIO。温度传感器用MAX6675读取，需要一个SPI口，或者需要两组ADC。加热器需要两个GPIO口控制。定位开关需要6个GPIO。

总结一下主控需要驱动4个PWM、18个GPIO、1个SPI。这些龙芯1C均可满足。



### 3.2 操作系统RT-Thread

操作系统采用RT-Thread 2.1。

#### 3.2.1设备驱动

GPIO读写、ADC的存取、中断的处理、串口的读写、SPI的读写、SD卡驱动、定时器timer的控制。

### 3.3 控制程序Marlin

Marlin所需要驱动的外围硬件：



Marlin固件架构

Marlin固件主要由四个部分组成，主程序控制部分（G指令解析与处理）、串口通信部分、PID温度调控部分、和运动控制部分，当然还有其他的，比如：液晶显示，SD卡读取等等。



固件中的运行过程主要有三个部分：主程序里面进行命令解析和处理，定时器1中断进行步进电机的控制和打印头的运动控制，定时器0中断则进行温度的检测和加热控制。这三个部分也都和?固件中的主要部分相关。

Marlin固件分前后台系统。前台系统即主程序，后台系统则是中断程序。在Marlin中用到了两个定时器中断：定时器0和定时器1。

主程序在前台的任务有：

1、与上位机进行通信，获得G指令

2、进行G指令解析，区分指令内容及指令参数，并将参数换算为整数

3、G指令的分类执行

4、温度管理、限位开关和LCD的控制

定时器0主要负责对挤出头的温度进行控制。在主程序中设置好定时器0的定时时间，等待定时中断的产生，并在中断中进行温度检测控制。

定时器1主要负责对步进电机进行控制，是Marlin中运动控制部分的核心。在主程序中先设置好定时器1的初始定时时间，等待中断后，在中断执行block中的运动。首先从block缓冲池中取一个block，分析该block中的运动参数，设置运动方向，然后调控步进电机运动。定时器1的定时时间就是步进电机的运动速度，所以在控制步进电机的速度方面，可以通过时间计算来实现。每个中断执行一个block中的1步或几步，这个参数在系统初始化时进行相关设置。

Marlin的步进电机驱动，是由中断响应函数实现的。然而远没有这句话来的那么简单：我认为整个控制系统的关键在于路径规划器（planner）。如果步进电机一直匀速运动，不需要任何规划器。不过对于3D打印机而言，x,y轴的运动往往速度变化非常频繁，如果采用匀速运动，速度的突然变化会给电路带来很大的冲击，影响机器的稳定性。这个物理层的原理，就决定了算法层应该采用加速减速的运动算法。路径规划器意味着，程序在执行步进电机的动作之前，就已经计算好了整个过程的速度曲线。后面就只是Stepper模块忠实地执行。这样的好处在于减少了中断响应函数中的运算量。由于3D打印机的机械运动相比控制器的16M主频来说要慢很多，采用缓存技术就能够有效的利用控制器速度。这里面蕴藏着“空间换取时间”的思想。路径规划器会在每次调用时，兢兢业业地把动作规划好，压入block\_buffer这个队列。 步进电机的移动速度是严格按照规划器的规划结果，通过设置每次中断的周期寄存器来实现的。

关于调平

不平分为打印出来的物品中间凹凸、朝一边斜两种情况

调平和精度是两回事，调平了，不一定精度就高

先确保不会往一边斜，即四周水平；然后再确保中间不会凹凸，即四周和中间都水平；最后调精度。

通过调整三个电机轴上限位开关的位置来确保不会往一边斜，即四周水平

通过调整打印半径来确保中间不会凹凸

通过调整拉杆长度来确保足够的打印精度

三个拉杆长度本身不可能完全相等，三个电机轴不可能是非常标准的等边三角形，再加上装配误差，所以调试的时候，只要把精度和误差控制在可接受范围内就好。

硬件资源划分

智龙v2.0上有5个led所在引脚可以与电机的使能或者方向引脚共用，这样led就是使能或者电机运转方向的指示灯

为了节省io资源，XYZE共用一个使能引脚

限位开关连接GND

下面以步进电机驱动芯片A4988为例，步进电机驱动芯片可以换为DRV8825等

Vcc ------ 5v A4988可以接3.3v或者5v

X电机

使能 ------ CAMDATA2/GPIO52

方向 ------ CAMDATA3/GPIO53

步进脉冲 ------ PWM0/GPIO06

限位开关 ------ CAMVSYNC/GPIO48

Y电机

使能 ------ EJTAG\_SEL/GPIO00

方向 ------ CAMDATA4/GPIO54

步进脉冲 ------ PWM1/GPIO92

限位开关 ------ CAMHSYNC/GPIO49

Z电机

使能 ------ EJTAG\_TCK/GPIO01

方向 ------ CAMDATA5/GPIO55

步进脉冲 ------ PWM2/CAMPCLKIN/GPIO46 复用pwm2

限位开关 ------ CAMDATA0/GPIO50

E电机

使能 ------ EJTAG\_TMS/GPIO04

方向 ------ CAMDATA6/GPIO56

步进脉冲 ------ PWM2/CAMCLKOUT/GPIO47 复用pwm3

限位开关 ------ CAMDATA1/GPIO51

做板子时，4个电机共用一个gpio作为使能，现在测试为了方便与ramps1.4连接，所以单独分配了一个gpio

TM7705模块

CS -------- SPI0\_CS1/GPIO82

MOSI -------- SPI0\_MOSI/GPIO79

MISO -------- SPI0\_MISO/GPIO80

SCK -------- SPI\_CLK/GPIO78

DRDY -------- I2S\_DI/GPIO87

RESET -------- I2S\_LRCK/GPIO89

VDD -------- 3.3V 因为龙芯1c的spi的SCLK是3.3v的，接5v会加大tm7705出现接口迷失的可能

GND -------- DGND

“安富莱”的TM7705模块测电压的精度还可以，但是接上ntc热敏电阻测温度误差就有点大了，这个误差并不是TM7705芯片引起的误差，而是外围电路。

TM7705芯片外围电路改动如下，

1，去掉模拟输入端串联的两个10k电阻，让ntc热敏电阻直接与TM7705相连。以通道1为例，R3换为0欧电阻，R4直接去掉。通道2类似修改

2，将TM7705的参考电源引出，ntc热敏电阻和4.7k的限流电阻也用TM7705的参考电源供电。这样热敏电阻的Vcc和Vref都是2.5V，即脚本createTemperatureLookup.py中的self.vadc = 2.5并且self.vcc = 2.5

修改后的TM7705芯片的外围电路图放在原理图目录内

挤出机加热

ramps1.4的D10 -------- I2S\_BCLK/GPIO90

挤出机散热

ramps1.4的D9 -------- I2S\_MCLK/GPIO91

暂时不考虑热床

用ramps1.4控制加热头加热和控制风扇时，感觉需要预热一段时间，加热的led过一会才会闪，并且亮度由弱变强，温度也是过一会才会升高；风扇的led会立刻亮，但是也要隔一会风扇才会转动。另外占空比不要太小，否则预热过程可能太长，误以为是哪里出了问题。

注意风扇有正负，接反了不转。当只给风扇提供pwm信号（加热头的占空比为0）时，ramps1.4上风扇对应的led是亮的，但隔了四五分钟后风扇才转起来。如果将加热头的占空比从0升高到0.4，风扇的不变，风扇很快就转起来，不知道是不是和我的开关电源有关，当只有风扇时，耗电不大（风扇为12V 0.1A），开关电源是12V 30A的。

优化

优化的目标：让3d打印机打印得又快又好，在保证打印质量的前提下尽量提高打印速度。

步进电机采用多少细分较好？

最小的喷嘴直径0.2mm，所以切片软件常将“层厚”设为0.2mm。我认为喷嘴的直径也就大致确定了3d打印机精度的级别了，其它因素优化得再好也没用。比如细分为4时，输入一个脉冲，步进电机走0.1mm，那么细分大于4时，输入一个脉冲，步进电机走的距离小于0.1mm，但我认为此时再提高步进电机的精度已经没有多大意义了，提升的步进电机的精度对打印作品的打印质量的改进已经不明显了。

优化的重点是在确保打印质量的前提下提高打印速度。

既然细分并不是越小越好，