

上海交通大学

大葱一体化处理 设计与制造2课程项目报告



小组成员: 唐永华 余建涵 张燊杰 汪远鹏

组号: 第三组

指导教师: 庄春刚

课程: 设计与制造二

目录

1. 项目背景.....	3
2. 预期产品及设计目标.....	3
3. 方案设计.....	4
4. 装置设计与选型.....	4
4.1 进料装置与运动方式选择.....	4
4.2 推动装置.....	4
4.3 切割装置.....	5
4.4 去皮装置.....	5
4.5 清洗装置.....	5
4.6 其他部位.....	5
5. 切割部分运动分析.....	6
6. 机械部分.....	6
7. 集成调试结果.....	8
8. 控制部件选取与代码实现.....	10
8.1 控制部件选取.....	10
8.2 代码实现.....	10
9. 外购清单.....	12
10. 总结与改进.....	13
10.1 总结.....	13
10.2 存在问题与改进.....	13
11. 小组成员分工.....	13
12. 致谢.....	14

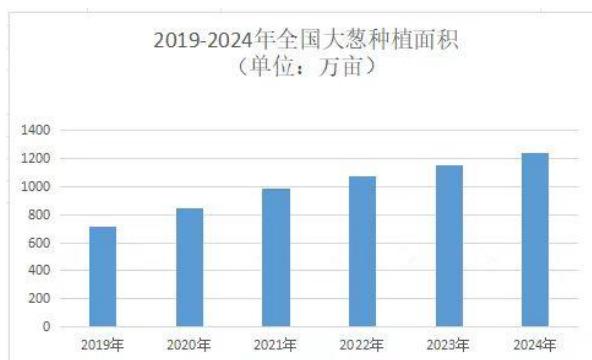
大葱一体化处理

1. 项目背景

大葱是我国北方地区重要的蔬菜作物之一，种植面积广、产量高、消费量大。然而，大葱在采收后到进入市场的流通过程中，普遍存在“净化处理环节落后”的问题。目前，大多数地区仍依靠人工方式进行清洗、去泥、修整和分级包装，劳动强度大、效率低，且清洗质量不稳定，影响大葱的商品外观和保鲜效果。

随着农业机械化和农产品加工装备的快速发展，市场对蔬菜清洗与分选设备的自动化、集成化需求日益增长。传统单一的清洗设备往往无法实现泥土分离、杂质去除、烘干与包装等多环节的连续作业，而一体化、模块化、节能环保的净化处理设备成为农业装备发展的重要方向。

目前我国的蔬菜清洗设备主要集中在叶菜类、根茎类和水果，针对大葱这种细长类蔬菜的专用设备较少。部分厂家（如山东、河南、江苏等地农机企业）开发了简易型“气泡喷淋一体机”或“毛辊去泥清洗机”，但多用于多种蔬菜通用，针对性不强，且存在自动化程度低，需人工干预；结构设计粗糙，已造成大葱损坏；清洗不彻底，有泥土残留等问题。



图片选自中国报告大厅《2025年大葱行业现状分析：全国大葱种植面积将达1300万亩》

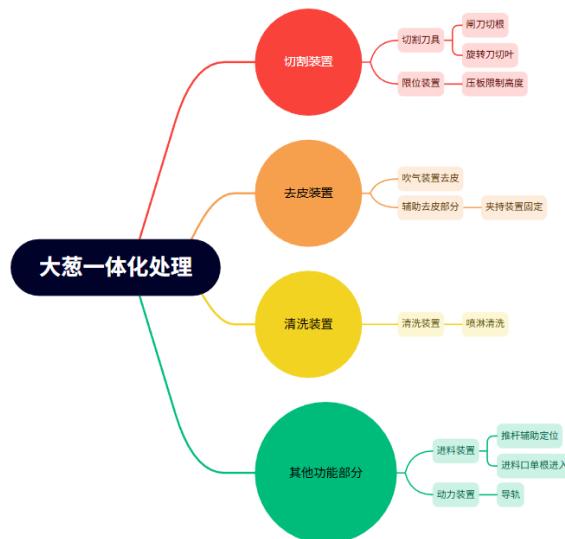
2. 预期产品及设计目标

我们的目标是把刚采收的农产品大葱经过一些处理后转变成在超市买到的商品大葱，而刚采收的大葱需要处理的部分有它身上的泥土、根须、多余的叶子以及根部最外层的薄皮，而因为清洗所涉及到的部分比较麻烦和复杂，所以我们只对根须、叶子和薄皮进行处理。为了实现目标，我们的预期产品为集切根切叶、去老皮以及清洗为一体的大葱处理机器。

3. 方案设计

为了实现大葱一体化处理，我们将装置分为以下部分：切割装置固定并准确切除大葱的叶和根；去皮装置去除大葱根部硬化的老皮；清洗装置清洗经过各类处理的大葱。

考虑到部分老皮连接在根上，不切去根会使老皮不易吹去，因此将切割流程放在去皮流程前；同时清洗后水及时老皮粘连在大葱上，不易吹去，因此将清洗流程放在去皮流程后。



4. 装置设计与选型

4.1 进料装置与运动方式选择

为了更好地处理大葱，我们选择将大葱变为单根与单根分离独立的状态，因此我们选择控制进料口的宽度，通过宽度限制单根大葱落下从而达成分离转态。

对于运动方式，本组成员原计划采取传送带运输来达成连续进行的多根大葱处理流程。后考虑到成本问题以及传送带尺寸问题，将传送带运输改为导轨运输，取单个大葱的处理流程为典型进行过程演示。

4.2 推动装置

推动装置的目的是使进料大葱处于相同的位置，因为对大葱切根和切叶需将大葱限制于合适的位置，这样可对切割位置有较好的选择，产品的质量与外形也有较好地保证，因此采取推动装置对每根进料大葱进行等距推动。

思路改进：原方案对大葱位置的处理为利用重力使其在斜面上滑到相同的底板上，从而

达成目标。但考虑到实际情况下摩擦力大小以及后续模块的装配固定问题，将处理方式改为用凸轮推杆机构进行等距推动。

4.3 切割装置

切割装置要实现对根和叶多余部分的切除，考虑到根和叶的形态和硬度差别，应采取不同的切割方法。根形状稳定，硬度更高，因此采取闸刀形式进行切割；而叶形状不稳定，较为杂乱且会有上下弯曲，同时硬度较低，因此采取旋转刀的形式进行切割。

根切割与叶切割都采取电机驱动，根切割采用凸轮推杆形式，通过推程与高度、受力情况等设计（具体分析放在后文）。叶切割采取电机直连旋转刀片，切割过程为在导轨运动到指定位置前启动叶切割电机，在大葱靠近过程中逐渐切去，避免了大葱移动被拦截或是大葱到达指定位置后旋转刀片受到阻力大的情况。

4.4 去皮装置

去皮装置主体由吹风机和机械爪构成。吹风机提供力来吹去老皮。在调研过程中发现目前已有的大型连续去皮方法使用最广的为吹气去皮，在经过线下实验发现老皮能被吹去，因此采取吹风机吹去等等方法。

机械爪的作用是固定大葱，防止在吹气时大葱被吹走。本组成员曾考虑过用压板的方法固定，但压板会导致老皮同时被压住，风吹入的面积小，因此采取机械爪抓住靠近叶部的位置进行固定。

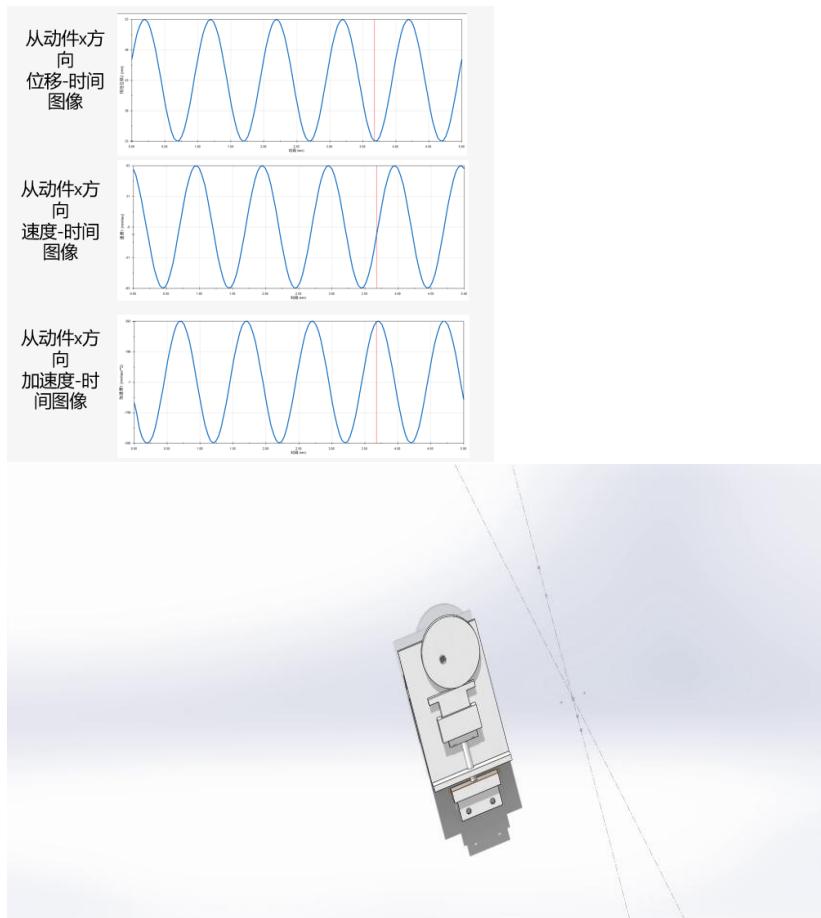
4.5 清洗装置

清洗装置采取水泵抽水喷淋头喷淋的形式进行清洗，在大葱经过上述处理移动到对应位置后启动水泵进行清洗。

4.6 其他部位

- (1) 连接在导轨上的运输板存在两个限位装置，防止大葱在移动过程中掉落
- (2) 外壳上连接有一个挡板，挡板位于靠近根的位置，在切割时压住大葱从而限制其高度，避免大葱切割时弯曲或飞出。

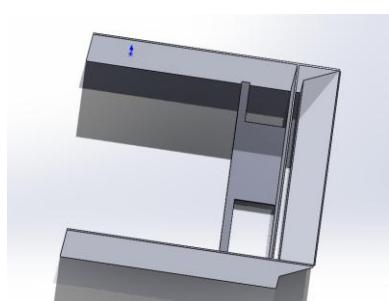
5. 切割部分运动分析



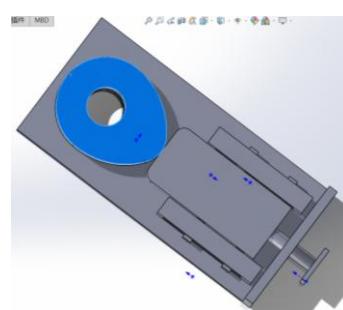
对切割部分进行分析设计。

6. 机械部分

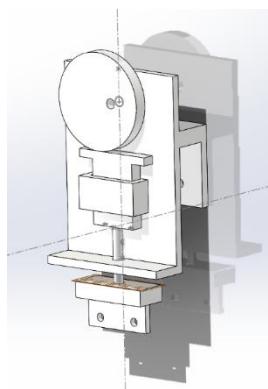
进料口:



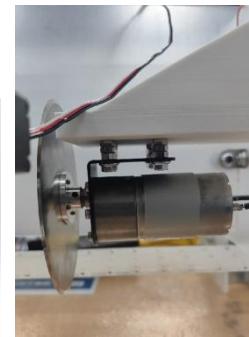
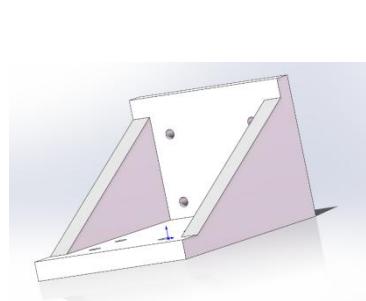
推动装置:



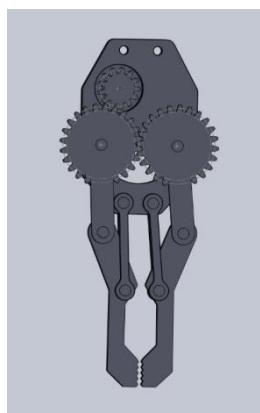
根切割装置:



叶切割装置:



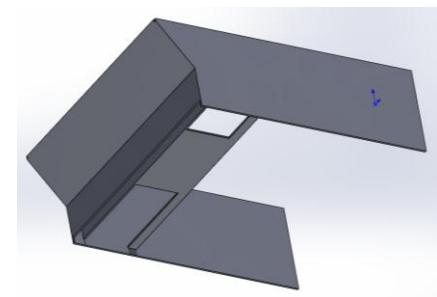
夹持装置:



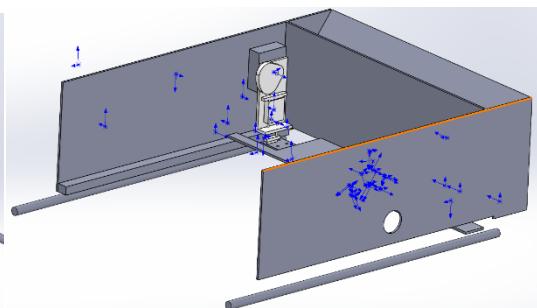
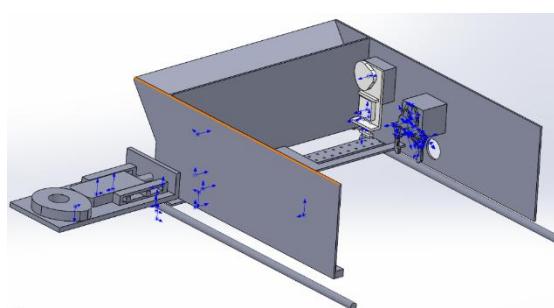
吹风装置:



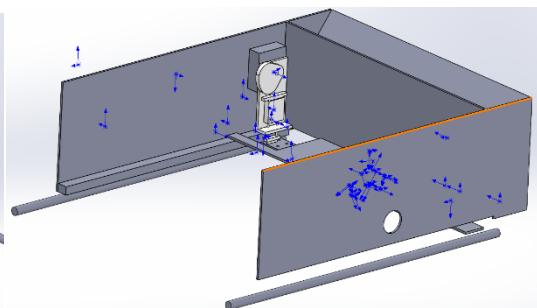
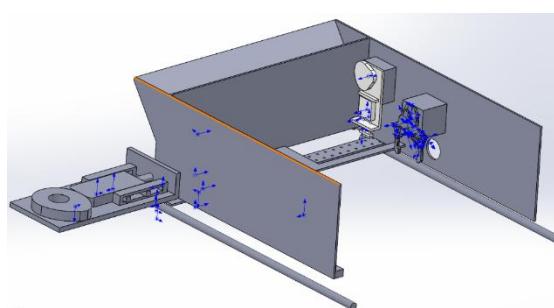
动力装置:



外壳:



总装配体:



7. 集成调试结果

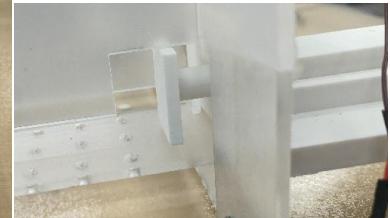
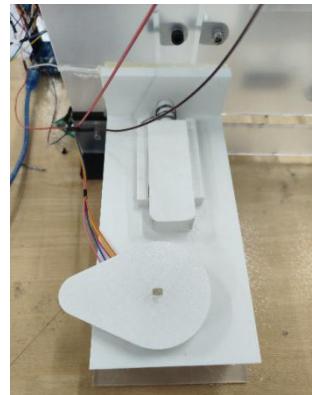
整体实物图:



进料装置:



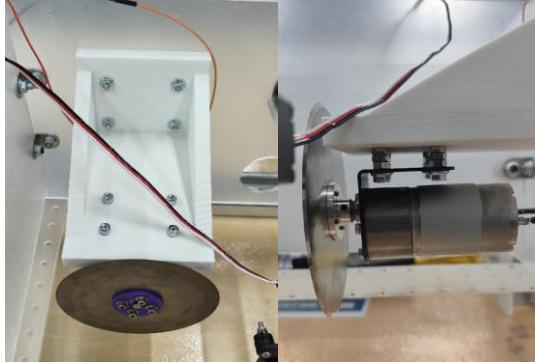
推动装置:



根切割装置:



叶切割装置:

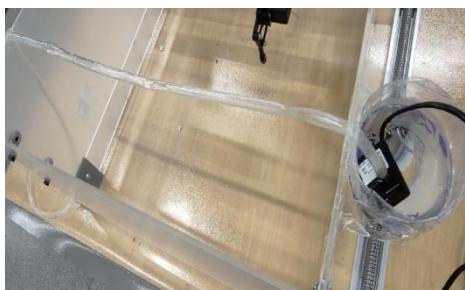


去皮装置:

吹风机与吹风口:



清洗装置:



动力装置:

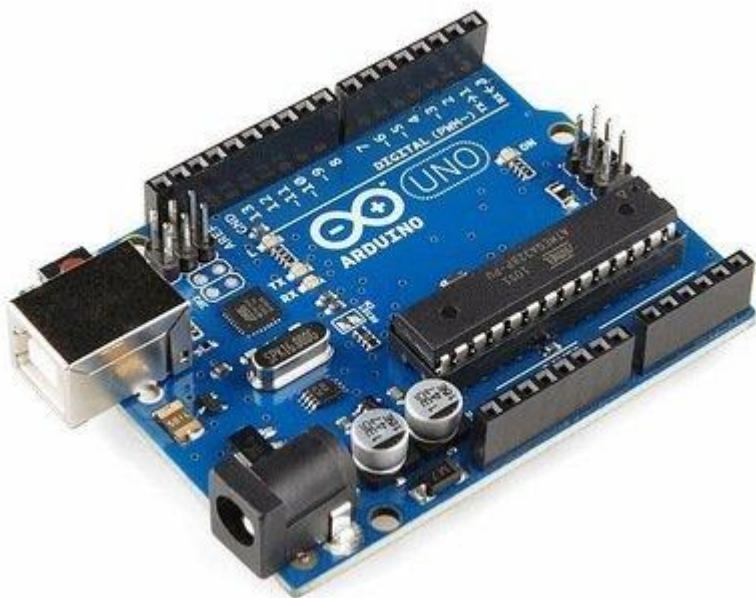


切割压板:



8. 控制部件选取与代码实现

8.1 控制部件选取



采用 arduino R3 板进行整体的控制，利用驱动器控制导轨及电机运动。



8.2 代码实现

```

1 #include <Servo.h>
2
3 // ====== 减速电机1控制 (IN1~6、IN2~7、ENA=12) ======
4 const int DEC1_IN1 = 6; // 减速电机1 IN1接Arduino 6
5 const int DEC1_IN2 = 7; // 减速电机1 IN2接Arduino 7
6 const int DEC1_ENA = 12; // 减速电机1 使能脚
7 bool dec1MotorRunning = false;
8
9 // ====== 减速电机2控制 (IN3~10、IN4~11、ENA=13) ======
10 const int DEC2_IN1 = 10; // 减速电机2 IN1接Arduino 10
11 const int DEC2_IN2 = 11; // 减速电机2 IN2接Arduino 11
12 const int DEC2_ENA = 13; // 减速电机2 使能脚
13 bool dec2MotorRunning = false;
14
15 // ====== PWM舵机 (LDX-335MG) 参数 ======
16 #define SERVO_PIN 5
17 #define SERVO_VOLTAGE "6-8.4V"
18 #define STEP_ANGLE 0.3
19 #define MOVE_DELAY 7
20 #define MIN_PWM 500
21 #define MAX_PWM 2500
22 #define ANGLE_RANGE 180.0
23
24 // ====== 57步进电机参数 ======
25 const int PUL_PIN = 8;
26 const int DIR_PIN = 9;
27 const long PULSES_PER_REV = 6400;
28 const int STEPPER_SPEED = 50;
29
30 // ====== 全局变量 ======
31 Servo myServo;
32 float currentServoAngle = 0.0;
33
34 // ====== 减速电机1独立控制函数 ======
35 void dec1MotorFullForward() {
36     digitalWrite(DEC1_IN1, HIGH);
37     digitalWrite(DEC1_IN2, LOW);
38     digitalWrite(DEC1_ENA, HIGH);
39     dec1MotorRunning = true;
40     Serial.println("【 减速电机1启动】全速正转");
41 }
42 void dec1MotorBrake() {
43     digitalWrite(DEC1_IN1, LOW);
44     digitalWrite(DEC1_IN2, LOW);
45     digitalWrite(DEC1_ENA, LOW);
46     dec1MotorRunning = false;
47     Serial.println("【 减速电机1执行刹车】");
48 }
49
50 // ====== 减速电机2独立控制函数 ======
51 void dec2MotorFullForward() {
52     digitalWrite(DEC2_IN1, HIGH);
53     digitalWrite(DEC2_IN2, LOW);
54
55     digitalWrite(DEC2_ENA, HIGH);
56     dec2MotorRunning = true;
57     Serial.println("【 减速电机2启动】全速正转【IN3=10、IN4=11】");
58 }
59 void dec2MotorBrake() {
60     digitalWrite(DEC2_IN1, LOW);
61     digitalWrite(DEC2_IN2, LOW);
62     digitalWrite(DEC2_ENA, LOW);
63     dec2MotorRunning = false;
64     Serial.println("【 减速电机2执行刹车】");
65 }
66 // ====== 双减速电机同步控制函数 (核心恢复) ======
67 // 同步启动两个减速电机全速正转
68 void startAllDecMotors() {
69     dec1MotorFullForward();
70     dec2MotorFullForward();
71     Serial.println("【 双减速电机同步启动完成】");
72 }
73 // 同步刹车两个减速电机
74 void brakeAllDecMotors() {
75     dec1MotorBrake();
76     dec2MotorBrake();
77     Serial.println("【 双减速电机同步刹车完成】");
78 }
79
80 // ====== 57步进电机控制 (恢复同步触发双减速电机) ======
81 void run57stepper(long revolutions, int actionType, unsigned long triggerTimeMs) {
82     long totalPulses = revolutions * PULSES_PER_REV;
83     long donePulses = 0;
84     unsigned long startRotateTime = millis();
85     bool actionTriggered = false;
86
87     Serial.print("\n[57步进电机启动] 正转");
88     Serial.print(revolutions);
89     Serial.print("圈,开始转动后");
90     Serial.print(triggerTimeMs / 1000);
91     Serial.println("秒触发双减速电机动作");
92     digitalWrite(DIR_PIN, HIGH);
93
94     while (donePulses < totalPulses) {
95         digitalWrite(PUL_PIN, HIGH);
96         delayMicroseconds(STEPPER_SPEED);
97         digitalWrite(PUL_PIN, LOW);
98         delayMicroseconds(STEPPER_SPEED);
99         donePulses++;
100    }
101   // 恢复同步触发逻辑
102   if (actionTriggered && (millis() - startRotateTime >= triggerTimeMs)) {
103       if (actionType == 1) {
104           startAllDecMotors(); // 同步启动双减速电机
105       } else if (actionType == 2) {
106           brakeAllDecMotors(); // 同步刹车双减速电机
107       }
108       actionTriggered = true;
109   }
110   Serial.print("【 57步进电机完成】");
111   Serial.print(revolutions);
112   Serial.println("圈转动");
113 }
114
115 // 57电机单独反转函数 (无减速电机触发)
116 void run57stepperSimple(long revolutions) {
117     long totalPulses = revolutions * PULSES_PER_REV;
118     Serial.print("\n[57步进电机启动] 正转");
119     Serial.print(revolutions);
120     Serial.println("圈");
121     Serial.println("【 (无减速电机触发) */");
122     digitalWrite(DIR_PIN, HIGH);
123
124     for (long i = 0; i < totalPulses; i++) {
125         digitalWrite(PUL_PIN, HIGH);
126         delayMicroseconds(STEPPER_SPEED);
127         digitalWrite(PUL_PIN, LOW);
128         delayMicroseconds(STEPPER_SPEED);
129     }
130     Serial.print("【 57步进电机完成】");
131     Serial.print(revolutions);
132     Serial.println("圈转动");
133 }
134
135 // ====== 指令执行函数 ======
136 void executeCmd1() {
137     Serial.println("【*** 开始执行指令1 ***】");
138
139     // 1. 舵机复位到0°
140     Serial.println("\n[步骤1] 舵机复位到0°");
141     setServoAngle(0.0);
142
143     // 2. 57电机正转22圈 + 1秒后同步启动双减速电机
144     Serial.println("\n[步骤2] 57步进电机正转22圈【1秒后同步启动双减速电机】");
145     run57stepper(22, 1, 1000);
146
147     // 3. 继续等待剩余时间 (总等待10秒)
148     Serial.println("\n[步骤3] 继续等待剩余时间【总等待10秒】");
149     delay(8000);
150
151     // 4. 57电机正转31圈 + 10秒后同步刹车双减速电机
152     Serial.println("\n[步骤4] 57步进电机正转31圈【10秒后同步刹车双减速电机】");
153     run57stepper(31, 2, 10000);
154
155     // 5. 舵机转到95°
156     Serial.println("\n[步骤5] 舵机转到95°");
157     setServoAngle(95.0);
158
159     // 6. 等待10秒
160     delay(10000);
161
162     // 7. 舵机复位到0°
163     Serial.println("\n[步骤7] 舵机复位到0°");
164     setServoAngle(0.0);
165
166     // 8. 新增步骤: 57电机正转47圈
167     Serial.println("\n[步骤8] 57步进电机正转47圈");
168     run57stepperSimple(50);
169
170     Serial.println("\n*** 指令1执行完成 ***");
171 }
172
173 void executeCmd0() {
174     Serial.println("\n*** 开始执行指令0 ***");
175     Serial.println("【步骤1】57步进电机反转53圈");
176     digitalWrite(DIR_PIN, LOW);
177     long totalPulses = 103 * PULSES_PER_REV;
178     for (long i = 0; i < totalPulses; i++) {
179         digitalWrite(PUL_PIN, HIGH);
180         delayMicroseconds(STEPPER_SPEED);
181         digitalWrite(PUL_PIN, LOW);
182         delayMicroseconds(STEPPER_SPEED);
183     }
184     Serial.println("【 57步进电机反转完成】");
185     Serial.println("【*** 指令0执行完成 ***】");
186 }
187
188 // ====== 舵机控制函数 ======
189 int angleToPWM(float angle) {
190     angle = constrain(angle, 0.0, ANGLE_RANGE);
191     return MIN_PWM + (angle * (MAX_PWM - MIN_PWM) / ANGLE_RANGE);
192 }
193 void smoothMove(float targetAngle) {
194     targetAngle = constrain(targetAngle, 0.0, ANGLE_RANGE);
195     if (abs(targetAngle - currentServoAngle) < 0.1) {
196         Serial.print("▲ 舵机已处于");
197         Serial.print(targetAngle);
198         Serial.println("°, 避免堵转! ");
199         return;
200     }
201     float step = (targetAngle > currentServoAngle) ? STEP_ANGLE : -STEP_ANGLE;
202     while (abs(targetAngle - currentServoAngle) > abs(step)/2) {
203         currentServoAngle += step;
204         currentServoAngle = constrain(currentServoAngle, 0.0, ANGLE_RANGE);
205         myServo.writeMicroseconds(angleToPWM(currentServoAngle));
206         delay(MOVE_DELAY);
207     }
208     currentServoAngle = targetAngle;
209     myServo.writeMicroseconds(angleToPWM(currentServoAngle));
210     Serial.print("【 舵机已到: */");
211     Serial.print(currentServoAngle);
212 }

```

```

213     Serial.println("==");
214 }
215 void setServoAngle(float targetAngle) {
216     smoothMove(targetAngle);
217     delay(500);
218 }
219
220 // ===== 初始化函数 =====
221 void setup() {
222     Serial.begin(9600);
223
224     // 步进电机1引脚初始化 (初始刹车)
225     pinMode(DEC1_IN1, OUTPUT);
226     pinMode(DEC1_IN2, OUTPUT);
227     pinMode(DEC1_ENA, OUTPUT);
228     dec1MotorBrake();
229
230     // 步进电机2引脚初始化 (初始刹车)
231     pinMode(DEC2_IN1, OUTPUT);
232     pinMode(DEC2_IN2, OUTPUT);
233     pinMode(DEC2_ENA, OUTPUT);
234     dec2MotorBrake();
235
236     // 舵机初始化
237     myServo.attach(SERVO_PIN);
238     setServoAngle(0.0);
239
240     // 5V电瓶引脚初始化
241     pinMode(PUL_PIN, OUTPUT);
242     pinMode(DIR_PIN, OUTPUT);
243     digitalWrite(PUL_PIN, LOW);
244     digitalWrite(DIR_PIN, LOW);
245
246     // 重新输出提示 (恢复步进电机同步插头)
247     Serial.println("== 多电机协同控制程序 (同步双减速电机) ==");
248     Serial.println("◆ 步进电机1: IN1=6, IN2=7, ENA=12 [初始刹车]");
249     Serial.println("◆ 步进电机2: IN3=10, IN4=11, ENA=13 [初始刹车]");
250     Serial.println("◆ 5V电瓶: PUL=8, DIR=9 [2圈1秒启双减速/3圈10秒刹车双减速, 未尾反转47圈]");
251     Serial.print("▲ 舵机工作电压: ");
252     Serial.print(SERVO_VOLTAGE);
253     Serial.println("输入指令 E [全速运行 | 0~57电机反转53圈]");
254 }
255
256 void loop() {
257     if (Serial.available() > 0) {
258         char cmd = Serial.read();
259         cmd = toupper(cmd);
260         if (cmd == '1') {
261             executeCmd1();
262         } else if (cmd == '0') {
263             executeCmd0();
264         } else if (cmd == '\n' && cmd != '\r') {
265             Serial.println("X 输入错误! 仅支持1或0");

```

9. 外购清单

名称	数 量	价 格	总 价	链接
德力普 (Delipow) 24V 锂电池组	1	106. .7	106.7	#小程序://京东购物 点外卖领国补 /ZMnnVkfARPpaUJE
2/57 步进电机驱动器 TB6600	1	27. 6	27.6	#小程序://京东购物 点外卖领国补 /pHWkIKBxfyWafD
2W 水泵	1	10. 39	10.39	#小程序://京东购物 点外卖领国补 /ID8xcfHmICTqE6z
DZYJarduino uno r3 开发板套件	1	59. 5	59.5	#小程序://京东购物 点外卖领国补 /ZBYCXsFqlunmpig
德立士吹风鼓风机	1	94	94	#小程序://京东购物 点外卖领国补 /j3bBvjMEXEd4SFz
幻尔黑色硬铝合金爪子	1	119	119	#小程序://京东购物 点外卖领国补 /g5ZZVymCkJ9Rt7k
康馨雅单线轨滚珠丝杆导轨	1	382	382	#小程序://京东购物 点外卖领国补 /Xjrz3dVjicbOEiz
L298N 电机驱动板模块 两路	2	8.7 2	17.44	https://e.tb.cn/h.7PxtdOZecrRxGc0?tk=mOvKU3J8jYe

L型 90 度直角不锈钢角码	3	7.7 3	23.19	https://e.tb.cn/h.7PxuPRLGXINDaOd?tk=vNo5U3JQ9Hz
JGY370 直流减速电机 12v	2	19. 21	38.42	https://e.tb.cn/h.7lFSKPQkx1hdsfK?tk=xJVIU3JQzuO
YDI-J32 旋转切割装置	1	99. 5	99.5	https://e.tb.cn/h.7PmaHmGIHRE54ZB?tk=9XuiU3JI00s
不锈钢三孔刀片	10	2.5	25	https://e.tb.cn/h.7PbA4RxLZcMiuJ0?tk=9qGhU3Jlk52
全部			1002. 74	

10. 总结与改进

10.1 总结:

本项目较好地完成了大葱一体化处理的流程，同时也综合运用了典型机构相关知识解决问题。

10.2 存在问题与改进:

1. 受限于经费原因，将传送带连续处理转变为导轨单个处理演示。
2. 进料时大葱叶处挡板应留有过渡空间，便于弯曲葱叶经过。
3. 机械爪可进一步拓展功能，在夹住叶片时可稍微抬起，便于吹去老皮。
4. 吹风时可额外设计一个导风管，减小吹风的损耗。
5. 喷淋装置可进一步优化，可提高抽水装置的功率以及设计更好的喷淋头。

11. 小组成员分工

汪远鹏：推动和清洗部分、报告
 唐永华：去皮和代码部分、采购零件
 余建涵：切割和仿真部分
 张燊杰：外壳部分

12. 致 谢

感谢小组成员在项目中做出的贡献！

感谢课程老师庄春刚以及助教胡鸿波在课程及项目中提供的帮助！

感谢项目指导老师袁志远为本项目提供的建议！