# 三子棋游戏装置 (E 题)

摘要：三子棋游戏装置包含对棋子的抓取、三子棋对弈算法的实现、单片机的统筹控制。同时装置可实现三子棋沿着自定义路径在棋盘上运动，实现与人对弈。通过移动平台来抓取和移动棋子，利用传感器检测棋子的位置，根据对弈算法决定下一步的最佳落子位置。对弈算法采用Alpha-Beta剪枝算法，以确保计算速度和策略的优化。装置人机交互界面友好，让用户可以轻松设置游戏参数、查看当前棋盘状态和操作机械手臂。实验结果表明，本装置不仅能够与人类对弈，还能在复杂的环境下保持稳定性和高效性。

**关键字：**STM32F407、步进电机、PID、树莓派、机器视觉

# **一、引言**

依据题目要求，需要设计一份三子棋游戏装置，能够控制机械臂或其他机构放置棋子，实现人机对弈。棋盘为九宫格，人与机器分别从棋子放置处拾取棋子并放置到方格中，先将己方的 3 个棋子连成一线（横连、竖连、斜连皆可）即获胜。

为了实现上述要求，我们设计了不同的模块。其中视觉模块利用树莓派实现黑白棋子识别与位置计算，确保系统准确感知黑白棋子。主控制器STM32F407在系统中起到核心作用，它接收视觉模块传来的数据，经过算法处理后控制步进电机的旋转。这种架构使得系统能够快速响应，有效地调节棋子的位置与方向。步进电机作为执行机构，能够准确地调整棋子的位置，实现各种下棋方案。

利用电磁铁吸取铁质的棋子，断电时放下棋子。

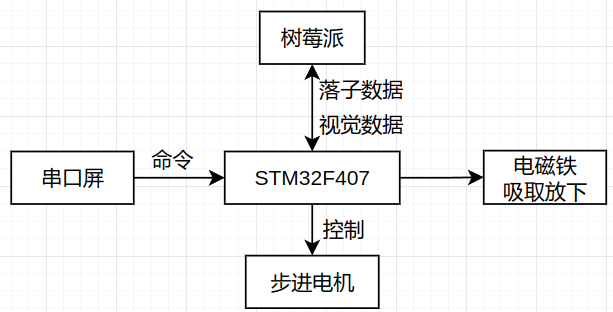
# **二、设计方案**

结合已有知识与对题目的分析，对以下几方面进行了分析和验证。

## **2.1 整体设计方案选择**

使用树莓派进行棋盘和棋子位置的识别，使用STM32F407开发板作为主控制器，使用步进电机驱动的移动平台进行棋子位置的调节，用电磁铁吸取棋子，制作设计本系统，系统框图如图1.

## **2.2 器件选型**

图 1: 系统框图

### **2.2.1 主控制器件的论证与选择**

可选的主控制器有STC89C52单片机、STM32F407单片机、Tiva C Series开发板，下面对这几种方案进行论证。

方案一：STC89C52单片机：

优点：51 单片机是应用最广泛的单片机，在互联网上有大量学习资料和配套资源；同时，51 单片机的价格很便宜，而且易于上手。

缺点：其 ROM、RAM、和外设资源比较有限，无法制作复杂的产品；而且运行速度较慢，I/O 引脚输出能力比较差，不适合本题中使用。

方案二：STM32F407单片机：

优点：STM32F407是一款基于 ARM CortexM3 内核的单片机，该型号单片机应用广泛，有丰富的外设，包括多个定时器、串口、I2C、ADC、API 等，可以满足诸多场景的需求。同时功耗较低且性能可观。且团队成员有比较丰富的开发经验，使用比较熟练。

方案三：Tiva C Series 开发板：

优点：Tiva 芯片采用了 ARM Cortex-M4 内核，主频更高，比 STM32 速度更快，并有丰富的外设，满足多种应用场景。同时支持多种开发工具。

缺点：资料相较于 STM32 较少，团队成员开发经验也不多，所以不适合本题使用。

综上所述，在本题中选用 STM32F407 单片机作为主控制器。

### 2.2.2 视觉模块论证与选择

可选的视觉模块有OpenMV、K210、Raspberry Pi 4B，下面对这几种方案进行论证。

方案一：OpenMV：

优点：Open MV 是一种常用的视觉模块，它是基于 ARM Cortex-M 系列的开源控制器，提供了许多图形处理功能，使用 Micro Python 编程，简单易用。

缺点：性能不够强，运行计算量较大的任务时速度很慢，也不能使用 C++ 进行开发来提高运行速度。

方案二：K210：

优点：K210 是基于 RISC-V 架构的处理器，较常用于机器学习和深度学习等应用，其算力远超 Open MV，同样的操作速度更快。K210芯片搭载了双核64位RISC-V处理器和专门的AI加速器，具有高性能的计算能力。在本题中，不需要复杂的 AI 运算，因此K210的优势无法完全发挥。

缺点：K210 运行时发热严重，而且 flash 空间较小，刷入较大的固件会卡顿。同时在 IDE 中的图像清晰度损失较大，不便于调试。

方案三：Raspberry Pi 4B

优点：树莓派（Raspberry Pi）是一种操作系统和硬件相结合的微型计算机，可以安装完整的Linux操作系统，而且识别功能较强大，可以使用 OpenCV 进行图像处理，可以使用 C++ 来加快处理速度；功能丰富，可以运行许多算法；同时支持 HDMI图形输出，便于现场调试；而且团队成员使用经验丰富。

缺点：GPU 性能弱，进行 AI 运算较慢，但本题不需复杂的 AI 运算。

综上所述，在本题中我们选用 Raspberry Pi 4B 作为视觉追踪和计算模块。

### 2.2.3 移动平台设计方案

可选的移动平台有舵机、直流电机、步进电机，下面对这几种方案进行论证。

方案一：舵机

优点：可以购买到直接可用的支架，结构简单，使用 PWM 操作比较简单，而且可以旋转较大的范围。

缺点：但是舵机旋转的精度不够高，无法满足题目的部分要求，而且电流稳定性较差，在负载变化较大时，可能会出现转矩不稳定的问题。因此不适合在此使用。

方案二：直流电机

优点：直流电机的结构相对简单，价格较便宜。通常具有较高的可靠性。通过控制电压或使用调速器实现较广泛的速度范围调节。

缺点：效率较低，能量损耗较大，耗电较多。不便于获取电机行进的位置和距离。

方案三：步进电机

优点：步进电机可以精确地调节旋转的速度和步数，转速较低。在具有外部驱动的情境下，可以很方便地调节步进电机的旋转，适合对棋子的位置精确地调节，更方便在本题中的调节。

综上所述，在本题中选用步进电机来驱动。

### 2.2.4 传动装置方案

可选的传动装置有机械臂、丝杆、皮带，下面对这几种方案进行论证。

方案一：机械臂：

**优点**：机械臂可以精确定位和重复操作，适合高精度任务，保证一致性并减少人为误差。其编程或学习能力使其适应不同任务和环境。

**缺点**：编程复杂，灵活性较低，处理复杂任务时可能受限，并且需要较大空间。

方案二：丝杆：

**优点**：丝杆提供精确的线性运动，适用于高精度定位，效率高，能量损失小，承受较大轴向和径向力，适合大范围移动。结构简单，维护容易。

**缺点**：行程长度可能有限，需要高精度安装，否则影响精度。长期使用可能磨损，高速运动时可能产生振动和噪音。

方案三：皮带：

**优点**：皮带传动安装简单，维护更换容易，效率可达98%以上。适用于远距离传动，可调节速度，运行平稳安静。

**缺点**：传动精度较低，承载能力有限，敏感于温度变化，易磨损，高速时性能受限，存在传动间隙和无法反向传动的问题。

综上所述，在本题中，选用皮带作为传动装置。

### 2.2.5 棋子移动方案

可选的棋子移动方案有机械爪抓取、磁铁吸附、气阀吸附，下面对这几种方案进行论证。

方案一：机械爪抓取：

**优点**：机械爪具备灵活性和多功能性，能够精确抓取、搬运和放置多种物体。其可编程控制和模块化设计使其适应不同任务。

**缺点**：对复杂形状和易损物品的抓取有限，可能需要定制解决方案，抓取力度控制可能影响适用性。

方案二：磁铁吸附

**优点**：磁铁吸附通过磁力实现无接触操作，安装简便，响应快速，适用于高精度要求的场合，且维护成本低。

**缺点**：仅适用于铁磁性材料，可能对物体造成磁化影响，且在有干扰物时吸附力可能减弱。

方案三：气阀吸附

**优点**：气阀吸附操作简便，响应速度快，适用范围广，可吸附多种材料，维护成本低。

**缺点**：需压缩空气系统，不适合重载应用，气压波动可能影响稳定性，极端条件下性能受限，精确控制略显不足。

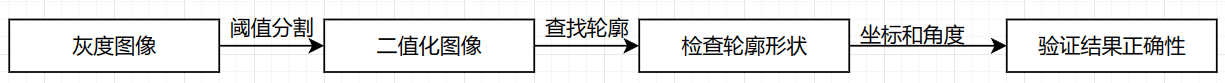
综上所述，在本题中，选用电磁吸附作为吸附方案。

# 三、理论分析与计算

## 3.1 棋盘、棋子识别方案

棋盘格和棋子识别流程如图2：

将相机拍到的灰度图像根据亮度区分出棋盘和背景，通过形状滤除掉环境的干扰，获得棋盘格的轮廓，根据排列角度和位置进行排序，得到坐标，最后验证结果的正确性。通过调整二值化阈值，可以适应不同的光照条件。

图 2: 棋盘棋子识别流程

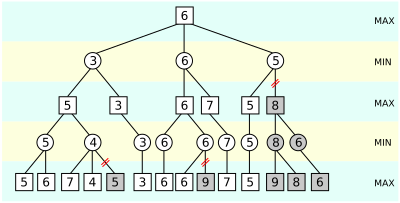
通过该流程可以获得正确顺序的九宫格坐标，发送到 STM32 单片机。

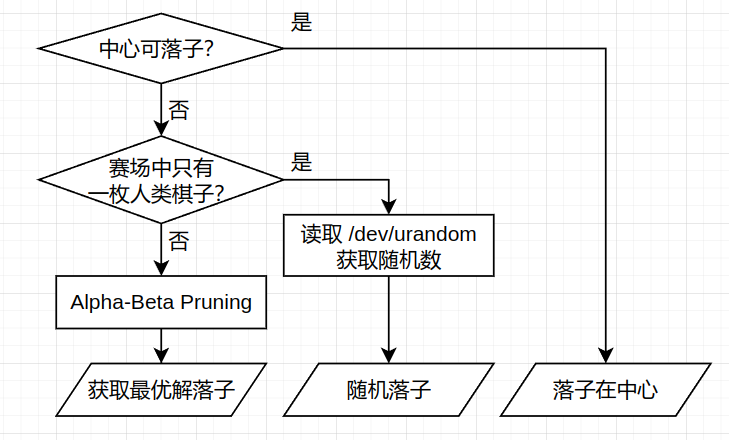
## 3.2 对弈算法

常用的算法有 Minimax (最小化极大算法)和 Alpha-Beta Pruning (Alpha-Beta剪枝，如图)，其中 Alpha-Beta Pruning 算法的计算量更少，可以在嵌入式平台更快的计算，在本题中，我们使用了优化的 Alpha-Beta Pruning 算法来完成题目要求。

Alpha-Beta Pruning 是一种搜索算法，减少Minimax算法搜索树的节点数，是一种对抗算法。当算法评估出某策略的后续走法比之前策略的还差时，就会停止计算该策略的后续发展。该算法和极小化极大算法所得结论相同，但剪去了不影响最终决定的分枝。见图3.

设计要求在任何情况机器不输，且人类玩家出错时能够胜利，因此可对算法进行优化。为了保证不输，考虑到玩家可能出现的错误，每次执行算法前判断是否可以将棋子下在中央，以保证后续计算量较小，且更容易获胜；另外当机器下第二个棋子时，任何位置其实是等效的，因此使用了随机函数来决定落子位置。如图4.

图 3: Alpha-Beta Pruning 算法

图 4: 对弈算法框图

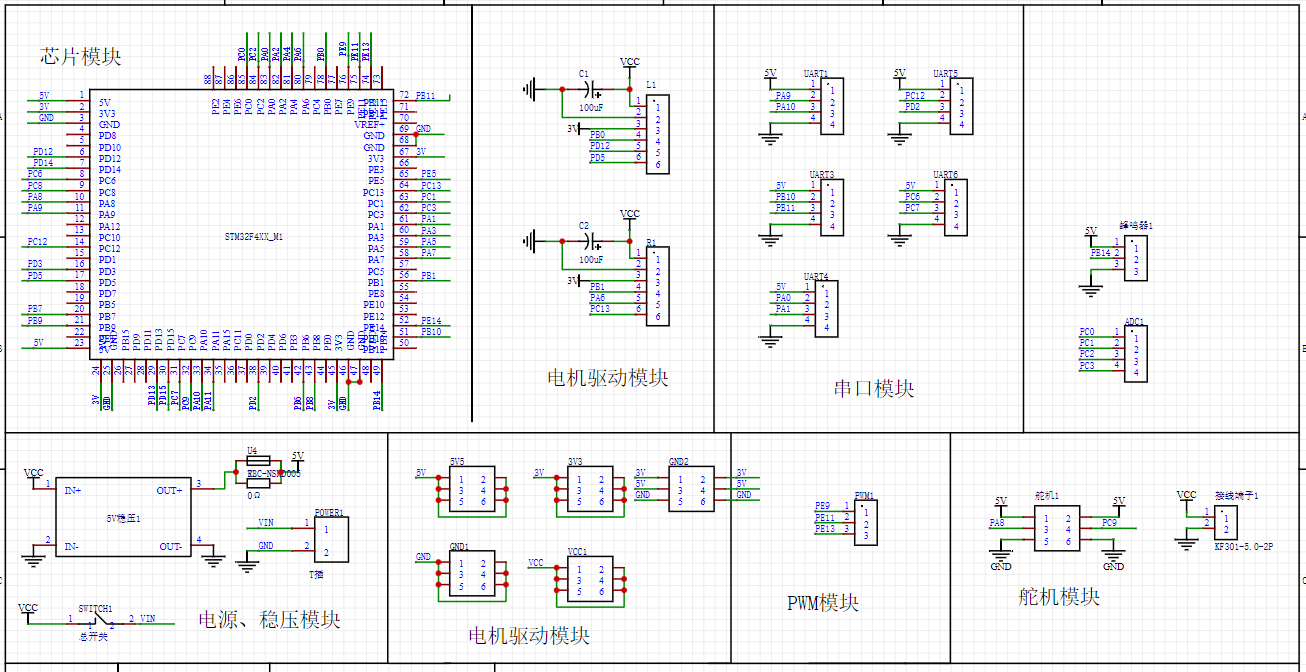
# 四、电路与程序设计

## 4.1 硬件电路设计

绘制PCB板和3D打印件，使用12V电池为电机供电，通过稳压模块输出为5V为单片机和树莓派供电。

单片机控制线与电机和电磁铁连接，以控制电机和电磁铁拿起放下棋子。

## 4.2 软件程序设计

图 5: 硬件原理图

### 4.2.1 程序功能描述与设计思路

功能描述：设计一份三子棋游戏装置，使用STM32单片机控制机械臂或其他机构放置棋子，实现人机对弈。

程序设计思路：

控制部分：记录树莓派发送来的九宫格坐标和棋子坐标，控制电机移动到棋子上方，启动电磁铁，之后移动到棋盘格上方，关闭电磁铁，放下棋子。如果树莓派发送数据表明人类玩家有移动棋子，则控制电机移动到新的位置吸取棋子，移动到原来的位置放下棋子，继续等待玩家下棋。

视觉部分：使用OpenCV识别棋盘格和棋子，发送坐标到STM32；人机对弈时，检查棋盘，将推荐落子的位置发送给STM32。

### 4.2.2通讯协议设计

STM32F407 单片机与 Raspberry Pi 4B 通讯协议设计如下

表 1: STM32与树莓派通信协议

|  |  |  |  |
| --- | --- | --- | --- |
| 帧头 | 消息类型 | 消息内容 | 帧尾 |
| 0xFF | 0xXX | …… | 0xFE |

表 2: 通信分类及其内容

| 消息类型 | 消息内容 |
| --- | --- |
| 0x01：棋盘格 | 5字节：棋盘格序号+4字节横纵坐标 |
| 0x02：棋子位置 | 5字节：棋子序号+4字节横纵坐标 |
| 0x03：落子位置 | 1字节：落子位置序号 |
| 0x04：检查游戏结束 | 1字节：判断游戏是否结束和游戏结果 |
| 0x05：检查移动位置 | 2字节：原始位置和新位置 |

### 4.2.3 执行机构控制算法：PID算法

当被控对象模拟 PID 控制系统组成, 如图6所示

PID 调节器是一种线性调节器, 它将给定值 r(t) 与实际输出值 c(t) 的偏差的比例 (P)、积分 (I)、微分 (D) 通过线性组合构成控制量, 对控制对象进行控制。

图示

描述已自动生成图 6: PID 算法

|  | (1) |
| --- | --- |

在本题中，针对整块棋盘上电机的位置和棋子的位置的相对距离，我们可以调试两组PID参数，根据树莓派返回的数据，进行电机旋转角度的调节。

# 五、测试分析与结论

|  |  |  |
| --- | --- | --- |
| 测试编号 | 测试内容 | 测试结果 |
| 1 | 装置能将任意1颗黑棋子放置到5号方格中 | 通过 |
| 2 | 装置能将任意2颗黑棋子和2颗白棋子依次  放置到指定方格中 | 通过 |
| 3 | 将棋盘绕中心±45°旋转后，装置能将任意2颗  黑棋子和2颗白棋子依次放置到指定方格中 | 通过 |
| 4 | 装置执黑棋先行与人对弈，能获胜 | 通过 |
| 5 | 人执黑棋先行，装置能正确放置白棋子  以保持不输棋 | 通过 |
| 6 | 对弈过程中，变动装置下过的1颗棋子，  装置能自动发现并纠正 | 通过 |
| 7 | 在不同时间位置多次测试，无特殊光照 | 通过 |

经过测试，装置在各个功能模块上表现良好，能够完成所有测试项目，可以准确识别和放置棋子，还在对弈过程中做出良好的决策；能够纠正被移动的棋子。总而言之，本系统完成了题目的所有要求。