

题目： 三子棋游戏装置（E 题）

摘 要

本小组设计了一款自动下棋机器人，利用 STM32F103C8T6 作为下位机主控，树莓派作为上位机计算平台。上位机通过 OpenCV 进行棋盘识别和图像处理，并通过 UART 接口与下位机通信。下位机负责控制舵机和电磁铁，以实现精准定位以及棋子的抓取。采用运动解算，确保机械臂的平稳运动和路径规划。上位机系统经过测试，能够在多种环境下准确识别棋子位置，控制误差小于 2mm，响应速度快，满足设计初衷和任务要求。

关键词：树莓派；OpenCV；STM32F103C8T6；机械臂；电磁铁；

1. 系统方案

1.1 方案分析与比较

(1) 控制器方案的选择

方案一：采用 TC264 作为主控模块

TC264 是一款高性能的微控制器，适用于多种嵌入式应用。它具有强大的处理能力和丰富的外设接口，能够支持复杂的控制和计算任务。然而，TC264 的成本较高，且编程难度较大，需要较长的开发时间和更高的开发成本。

方案二：采用 STM32F103C8T6 作为主控模块

STM32F103C8T6 是一款性价比较高的微控制器，基于 ARM Cortex-M3 内核。它具有较大的存储器和多个定时器、通信接口，能够满足多种嵌入式应用的需求。该微控制器工作频率较高，运行速度快，且开发资料丰富，易于开发和调试。

选择理由：综上所述，由于本设计需要满足快速响应和高效控制的要求，且考虑到开发成本和难度，STM32F103C8T6 能够很好地满足设计需求。因此，本设计最终选择方案二，采用 STM32F103C8T6 作为主控模块

(2) 上位机方案选择：

方案一：OpenMV

OpenMV 是一款小型、开源的视觉模块，旨在提供易于使用的计算机视觉功能。它由微控制器和摄像头模块组成，可以执行各种图像处理和计算机视觉任务，如颜色跟踪、形状检测、条形码识别等。

缺点：价格昂贵，性价比较低，在需要更高要求的图像处理速度方面无法达到要求。

方案二：树莓派

是一款基于 ARM 架构的单板计算机，内置 Wi-Fi 和蓝牙，支持多种操作系统，如 Raspbian、Ubuntu 等，方便远程进行操作，可以通过在系统内直接调用 OpenCV 等库函数，实现高帧率的图像处理。

选择理由：综上所述，考虑到开发的便捷性和成本，以及对图像高帧率的追求，选择树莓派为最佳方案。

1.2 系统总体方案设计

本系统分为主控模块、棋盘识别模块、机械臂控制模块、交互模块以及电源模块五部分组成。主控模块由 STM32F103C8T6 微控制器构成，负责整个系统的核心控制。该模块接收上位机传输的指令，协调各个功能模块的操作，并通过串口接口与上位机进行数据通信。

棋盘识别模块由树莓派及其摄像头组成，负责棋盘和棋子的识别。树莓派通过摄像头采集图像，使用 OpenCV 库进行图像处理和棋子位置识别，然后将识别结果传输至主控模块。

机械臂控制模块由多个舵机和电磁铁组成，负责实现机械臂的运动和棋子的拾取放置。STM32F103C8T6 通过 PWM 信号控制舵机运动，实现机械臂的精准定位，电磁铁通过 GPIO 口控制其吸附和释放功能。

交互模块包括一个显示屏，用于显示系统状态和操作信息。主控模块将处理后的数据和操作结果显示在串口屏上，以使用户实时查看和操作。

电源模块为系统各部分提供稳定的电源，确保系统正常运行。根据系统功耗需求选择合适的电源模块，为 STM32F103C8T6、树莓派、舵机及其他外围设备供电。

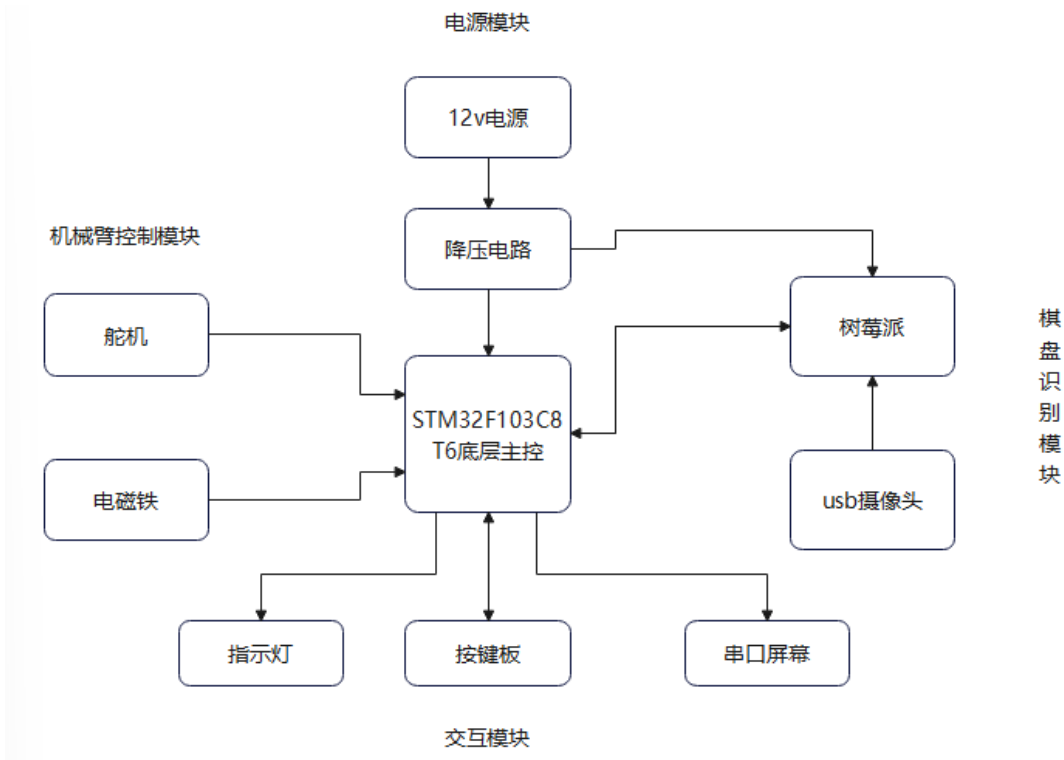


图 1 装置整体框架图

2. 理论分析与计算

2.1 棋盘、棋子识别方法

下棋机器人的上位机基于树莓派 3B 的 Linux 系统开发，主要负责对摄像头回传的视频流进行图像处理以及下棋过程中棋局的算法解析部分。

3.1.1 图像处理

我们选择了以黄色作为棋盘的底色，因为黄色在视觉上具有较高的识别度，便于在后续的图像处理环节进行有效的颜色分割。同时为了提取棋盘的元素，利用了棋盘具有黑色外延轮廓明显的特征，以此来通过图像的轮廓识别技术来准确地找到棋盘的边界，并将其从图像中分离出来。

对于棋盘上的黑色和白色棋子，采用了 HSV 颜色空间的颜色提取技术，能够更有效地分离不同颜色的对象，再根据分离的对象通过矩阵运算可以精确地提取出黑色和白色棋子的坐标

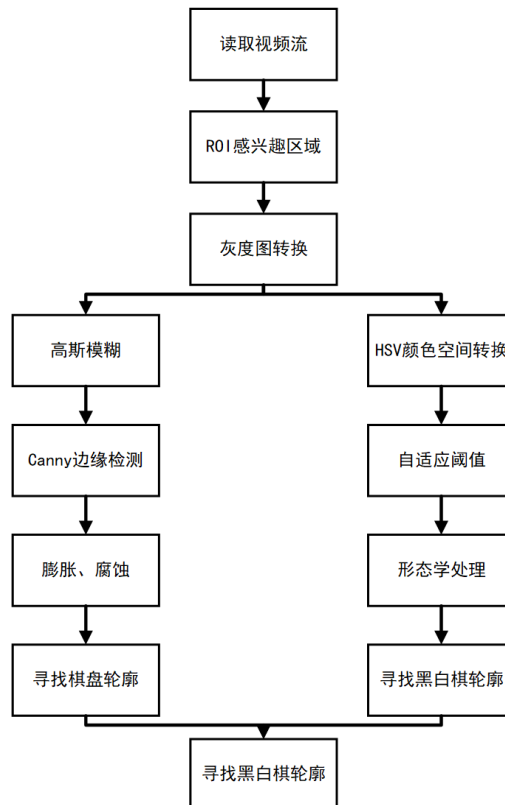


图 2 图像处理流程图

2.2 对弈算法

本次棋局的对弈算法主要针对任务要求的 4、5、6 问，根据分析，三个问题的共同点就是要求装置和人对弈平局或胜，而思路则是在人或者装置棋子能够在横、竖、斜的三个方向上连成三颗棋子，因此在对弈的过程中，摄像头需要不断计算棋局内黑白棋的情况，根据棋局更新黑白棋的情况进行逻辑分析来决定装置下一步应该如何下。其中的算法则采用最简单的棋局遍历法，就是将棋局的情况在上位机拟合出一个 3*3 的数组，0 表示没有棋子，1 表示黑棋，2 表示白棋，每当轮到白棋下的时候，都会通过更新棋局内的最新情况到本地的数组中，然后通过遍历法遍历数组计算棋局的情况。

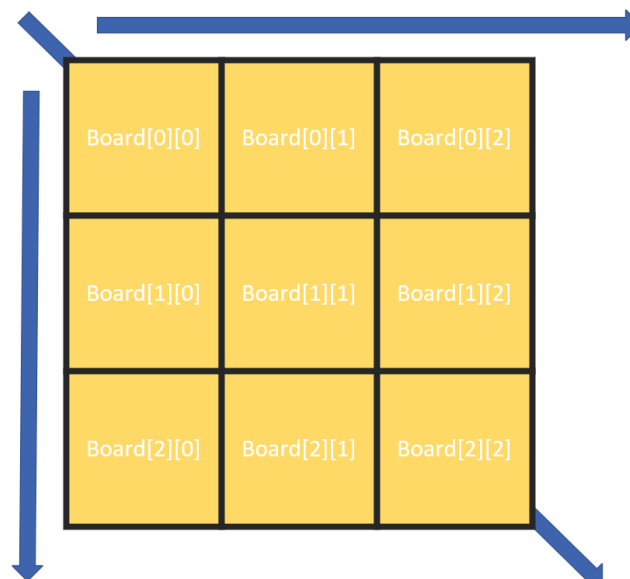


图 3 棋局遍历法流程图

但是由于在三子棋游戏中，黑棋先行若下在中间位置时，白棋下一步位置没有放在对角的位置，这时会出现必输情况，因此棋局对弈算法部分还要区别黑棋第一步是否下在中间，然后再采用不同的算法。

在第四问中，当黑棋第一步下在中间，与此同时白棋未放在对角格时，白棋必输，此时我们设定了装置下两步会遍历整个棋局对角点是否为空，并下在为空的位置。

在第五问中，当人执黑棋先行放于中间时，为了确保装置不输，装置第一步的白棋应下在棋局的对角处，此后装置在下棋部分只需进行正常棋局遍历法即可。

3.1.3 棋局情况检测

从上文我们可以得知，我们的算法对棋局的实时情况要求很高，需要实时对对手的棋子在棋局的位置进行检测，而这其中就涉及到对于如何准确更新的情况，提高准确性。

在程序运行的最开始，程序会在最开始的 20 帧进行对棋局轮廓的获取，将获取到的棋盘九宫格通过矩阵运算得到每个轮廓的 x 、 y 坐标值，同时由于我们也能实时获取到棋子的坐标值，因此当对手下棋时，如果程序检测到某个黑白棋在棋盘上的位置有变动，并且在白棋更新的时间间隔——10 秒，以防止人手或者机械臂导致的误判和重复更新，能够保持坐标位置的稳定，然后通过遍历，计算棋子和九个格子的坐标值的距离差，当小于设定的一个阈值时，会判定为棋子在对应的棋格内，以达到对于棋局情况检测的方式。

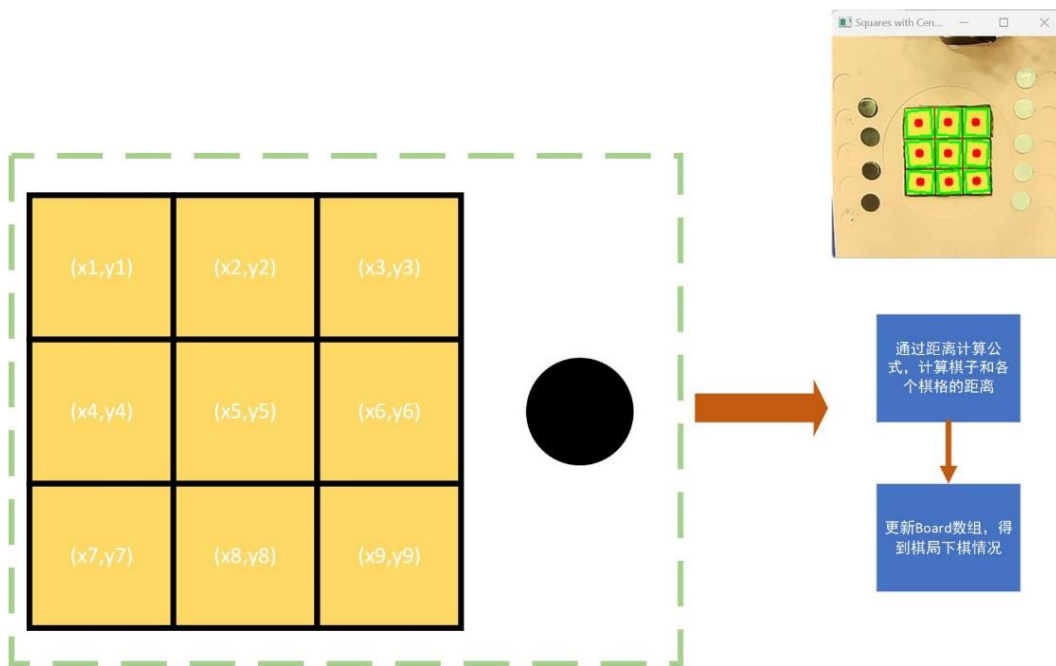


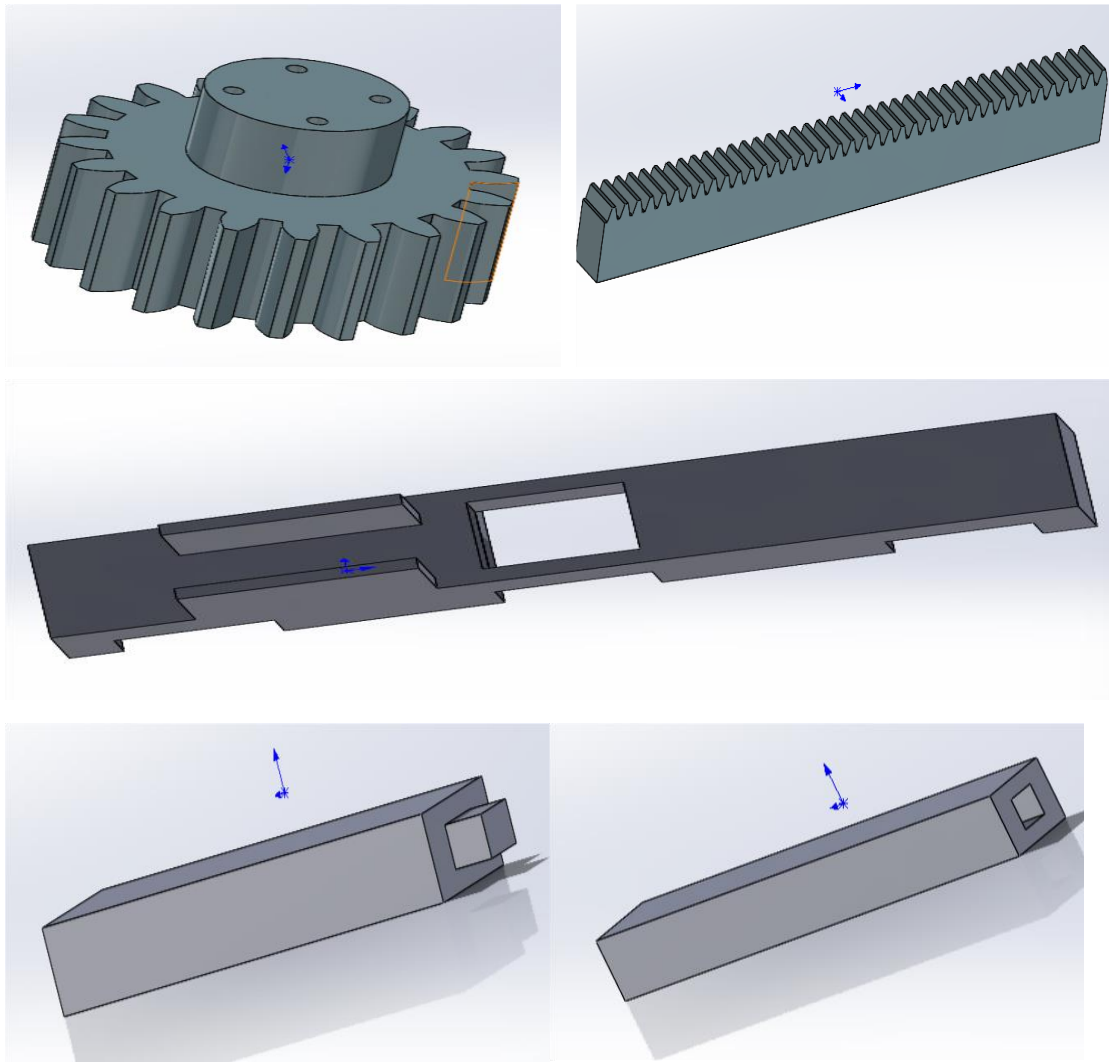
图 4 棋局遍历法流程图

3. 结构，电路与程序设计

3.1 结构设计

一：仿 3d 打印机结构下棋机器人

因为 3d 打印机可以在一定平面内实现 x 、 y 、 z 轴的移动，对于三子棋来说，如果利用 3d 打印机结构来实现棋子的定位与抓取，配合步进电机可以实现较为准确的拾取棋子。以下是我们利用 3d 打印制作的结构。



图组 1 装置 3D 模型图

利用齿轮与齿条的啮合，实现 x , y 轴的平移，在平台上面搭建机械臂与电路板。再使用一个具有两舵机的机械臂去拾取，完成目标任务。但在搭建过程中，发现 3d 打印机的材质强度不够，且 x 轴与 y 轴的位移误差较大，且由于机械臂在平台上方，再运动过程中，平台容易变形弯曲，且重心变化较大易偏移。

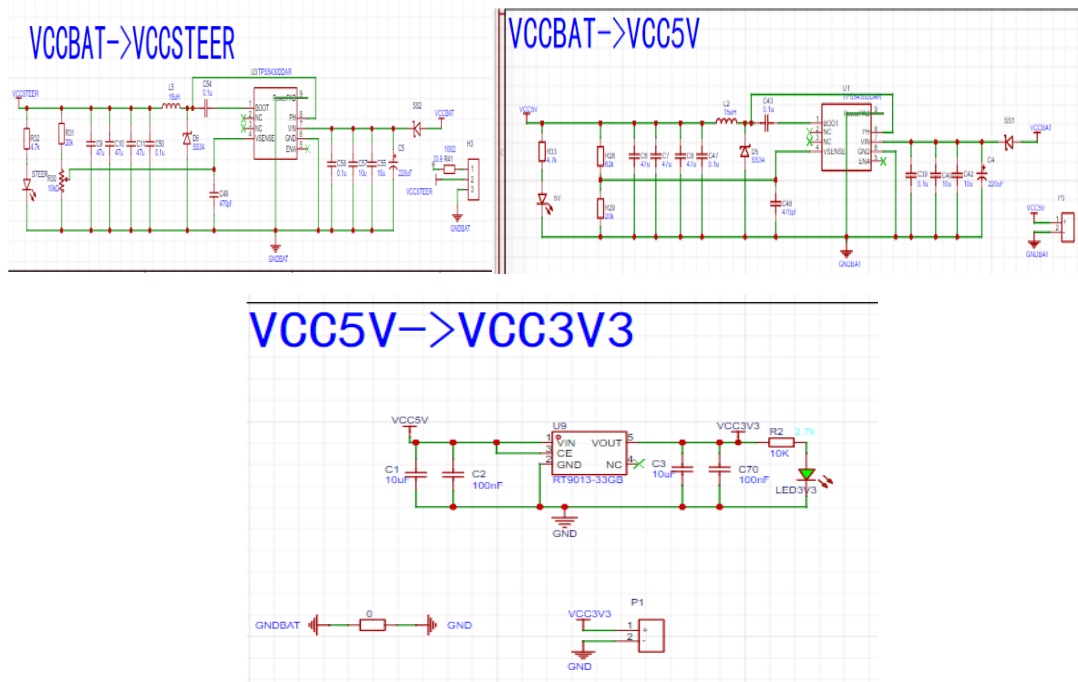
方案二：四舵机自动下棋机器人

由方案一可知，机械臂的运动会导致机器人的重心不稳，故应该设计一种底盘较低，重心较低的简单机构。所以我们设计一个 3d 底盘，中间留空用以增加配重，平衡装置。用四舵机机械臂实现 x , y , z 轴的运动。

方案二较方案一而言，结构更加简单，用到了更少的 3d 打印件，减少了齿轮齿条传动，减少了传动误差。舵机精度相对较高。故选择方案二

3.2 电路设计

本系统采用了基于 stm32 为核心的控制板，内含降压电路、稳压电路、串口电路



图组 2 电源原理图

电源输入端接入 12V 电池，经过降压电路和稳压电路处理后，向 stm32 和机械臂供电。单片机通过串口电路与树莓派等电路双向通信。同时单片机直接控制 LED 灯显示电路和机械臂，完成棋子拾取功能。各部分电路都设计了一定的隔离和保护功能，相对地提高了系统的稳定性。

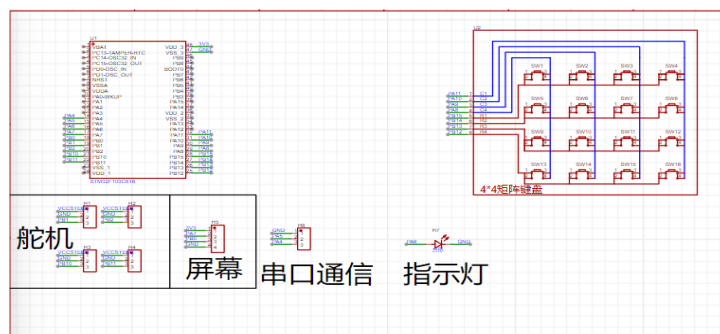


图 5 部分原理图

3.3 程序设计

下棋机器人的上位机基于树莓派 3B 的 Linux 系统开发，主要负责对摄像头回传的视频流进行图像处理以及下棋过程中棋局的算法解析部分。然后，程序将机械臂所需夹取的棋子编号和应放置到的棋格编号发送给下位机。

下位机运行时，可通过矩阵键盘确定工作状态，并通过串口从上位机获得该模式下经上位机图像处理与对弈决策后输出的方格号。下位机根据对应的方格号计算并调整 STM32F103 四个通道的 PWM 值，分别控制四个舵机，同时对 STM32F103 的一个 IO 口进行电平翻转以控制电磁铁，从而完成对棋盘的定位与棋子的抓取，实现题目要求。

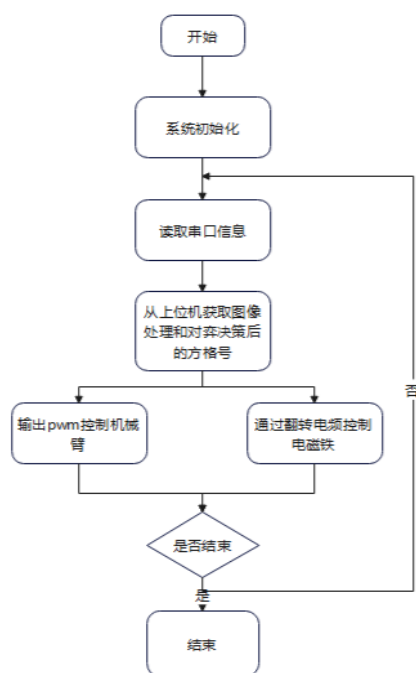


图 6 下位机程序流程图

在上下位机通信过程中，为确保装置的运行过程流畅且避免步骤错漏，我们设定了以下装置执行步骤：当人执棋子下完棋后，需要按下按键板上的特定按钮，发送执行下一步夹取命令的信号，机械臂才能继续执行下一步操作。

4. 测试方案与测试结果

4.1 系统测试条件

编程环境：Keil ARM、Pycharm、Linux 系统

测试工具：

- (1) 精度 0.1cm 电子尺：用于测量机械臂的定位精度
- (2) 测试棋盘和棋子：用于模拟实际操作环境
- (3) 摄像头模块：用于棋盘和棋子图像采集
- (4) 电源模块：确保系统稳定供电

4.2 系统测试方案

4.2.1 棋盘识别测试

通过 HMI 串口屏显示棋盘和棋子的识别结果，验证图像处理算法的准确性和稳定性。经过多次测试，确保在不同光照和环境下，系统能够准确识别棋盘和棋子的位置。测试结果如表 1 所示。

表 1 棋盘识别测试记录表

测量次数	光照条件	棋盘方格号 定位正确次数	棋子识别正 确次数	识别准确率
1	正常光照	20	20	100%

2	弱光照	19	18	95%
3	强光照	18	19	93%
4	正常光照	20	20	100%
5	弱光照	19	19	95%

4.2.2 机械臂定位和操作测试

通过精度 0.1cm 电子尺测量机械臂的定位精度，验证舵机控制和运动解算的效果。测试机械臂在不同距离和角度下的定位精度，确保误差在可接受范围内。测试结果如表 2 所示。

表 2 机械臂定位测试记录表

测量次数	目标位置 (cm)	实际位置 (cm)	误差 (cm)
1	(1, 1)	(1.1, 1.0)	(0.1, 0)
2	(3, 3)	(3.3, 3.05)	(0.3, 0.05)
3	(5, 5)	(4.9, 5.0)	(-0.1, 0)
4	(7, 7)	(7.0, 7.1)	(0, 0.1)
5	(9, 9)	(9.2, 8.9)	(0.2, -0.1)

4.2.3 棋子拾取和放置测试

通过多次测试，验证机械臂能够准确拾取和放置棋子。测试不同位置的棋子拾取和放置的成功率，确保系统在实际操作中的稳定性和可靠性。测试结果如表 3 所示。

表 3 棋子拾取和放置测试记录表

测量次数	棋子位置 (cm)	拾取成功次数	放置成功次数	成功率
1	(1, 1)	20	20	100%
2	(3, 3)	18	19	93%
3	(5, 5)	19	18	93%
4	(7, 7)	20	20	100%
5	(9, 9)	19	19	95%

4.3 总结

通过以上测试方案，验证了自动下棋机器人系统在棋盘识别、机械臂定位与操作、棋子拾取与放置以及系统响应速度等方面的性能和稳定性。系统在多种环境下表现出色，能够满足设计初衷和任务要求。自动下棋机器人具有较高的识别准确率、定位精度和响应速度，适用于各种自动化棋类应用场景。