

# 竺可桢学院 工程教育高级班



- 2025 实践面试题 -

2025. 03

# 目 录

序 • 灵剑弈云	1
重要说明	2
<b>Pre Assignment: Git Usage</b>	4
导言	4
提交说明	4
<b>Part 1 First Repository: First Commit</b>	6
Task 1: Initialize Repository	6
<b>Part 2 Branching: Merge, rebase...</b>	7
Task 2: Merging Branches and Handling Conflicts	7
Task 3: Rebasing Branches and Handling Conflicts	8
<b>Part 3 Remote: Working with others</b>	9
Task 4: Remote Repository Operations	10
<b>Bonus .gitignore: Ignoring Unnecessary Files</b>	10
Task 5: Be Careful when Pushing	11
<b>Assignment A: Hex 棋</b>	12
Part 1	13
Question 1.1	13
Question 1.2	13
Part 2: Hex 博弈程序实现	13
Bonus 1	17
1. 结束情况判断	17
2. 特殊棋形策略库	18
3. 开局库	18

4. 其他优化 . . . . .	18
<b>Part 3: Hex 可视化棋盘 . . . . .</b>	<b>19</b>
<b>Bonus 2 . . . . .</b>	<b>20</b>
1. 棋盘样式控制 . . . . .	20
2. 撤回列表 . . . . .	20
3. 保存棋谱 . . . . .	20
4. 接入智能模型 . . . . .	21
5. 其他优化 . . . . .	21
提交文件夹说明 . . . . .	21
<b>Assignment B: 嵌入式开发实践 . . . . .</b>	<b>22</b>
<b>Part 0: 准备工作 . . . . .</b>	<b>22</b>
<b>Part 1: 原理图与 PCB 识读 . . . . .</b>	<b>24</b>
Question 1.1 . . . . .	25
Question 1.2 . . . . .	25
Question 1.3 . . . . .	25
<b>Part 2: 原理图与 PCB 设计 . . . . .</b>	<b>25</b>
Question 2.1 . . . . .	25
Question 2.2: 支持快充协议 . . . . .	26
Question 2.3: 充电协议嗅探 . . . . .	26
<b>Part 3: 电路焊接 . . . . .</b>	<b>27</b>
<b>Part 4: 嵌入式软件开发 . . . . .</b>	<b>27</b>
Question 4.1: Hello World . . . . .	28
Question 4.2: 串口实现 . . . . .	28
Question 4.3: UI 设计 . . . . .	29
<b>Bonus: 初探 NFC . . . . .</b>	<b>29</b>
Question 5.0: 背景知识 . . . . .	30
Question 5.1: 读取、写入 CUID 卡 . . . . .	31

<b>Question 5.2:</b> 解密 CUID 卡、校园卡 . . . . .	32
<b>Question 5.3:</b> 卡模拟 . . . . .	34
温馨提示: 视频提交说明 . . . . .	34



## 序 · 灵剑弈云

致决云问剑的你：

剑法秘籍薪火相传三十载春秋，其间并未镌刻孤锋独舞的寒芒，却藏有诸多侠客共弈的玄机。昔年祖师以剑为笔，以棋为墨，将黑白纵横之理融入无道剑术之间，唤众人执子列阵，齐心协力，方得出其不意，破局连环，终称霸一方豪杰，写成了剑坛奇迹。后众弟子勤于习剑，亦熟诵先人之教诲：此间江湖，剑道非独行之路，算法亦非孤算之谋。唯勤于练习剑招，布局定一而谋全身，众心合一，方得赴云巅问道，剑指凌云。

先人曰：孤棋易陷困局，单锋亦难胜敌，唯有众刃列阵，循于棋理而推演天机，方能势破连环。深夜鏖战，是剑招与棋路交织的金石铮鸣；方案更迭，似残谱重排后的涅槃新生。败式未必失先，暗藏弈里乾坤；奇径未必无功，自有偏锋之妙。须知博弈之智，非独算尽乾坤，更在剑阵同频、弈心共振。

莫怕云深路险，绝巅总在履痕尽处方见棋剑通明。此番试剑，非求一子定江山，但求以实践为磨石，千锤百炼，而后登峰造极。嵌入式暗匣的千百次啮合，代码经脉的查错推敲，需倾注耐心，方令剑气贯通无碍；电路星火明灭间焊点如棋局，棋法错综的难以捉摸，需众手共点，方在策略上更胜一筹。终有一日，剑阵流转的凛冽气韵，将劈开混沌，星河倾覆时万象皆暗。

灵剑既出，当以三十载众志成城，换弈云一瞬破云峥嵘。此卷为凭，愿诸君执剑为子，列阵成局，共赴云巅问道。剑锋所指，棋路所及，皆是江湖。

二〇二三级工高班

# 重要说明

## 一、题目说明

1. 本次实践任务共分为 **Pre Assignment**、**Assignment A**、**Assignment B** 共三个部分，在这次实践面中，你们将以小组为单位，来解决这些问题。从现在开始你和你的团队伙伴有三天（3.20-3.22）的时间阅读题目、查阅资料、解决问题。你们需要以 **PDF** 的形式，提交完整的解题报告，并展示你们的成果。
2. **实践面试结束后所有参加实践面试的同学将直接进入第一轮面试**，但实践面试的表现将会作为总成绩的组成部分之一，是筛选的重要指标。团队可以使用你们发现的任何“无生命”资源，例如网页、书籍、研究报告等。但是团队不得寻求帮助，不得从除团队成员之外的任何人处获得答案、想法或信息。一旦发现作弊，视情况扣分，情节严重者将取消专家面试资格。
3. 为了规范你们的协作及文件管理，请在开始完成任务之前，认真完成 **Pre Assignment**，并在本次实践面中尝试使用 **Git** 协作。另外，请务必注意：在 **ZJU Git** 或 **Github** 上托管你们的代码和成果时，需要将项目的 **Visibility** 调整至 **Private**，防止违反诚信守则的规定。
4. 良好的团队合作、规范的学术报告、完整的解答过程、严密的逻辑思维将会是本次实践面试的重点考察内容。请各组在解答问题时，注意规范性、严谨性、逻辑性。

## 二、网站提交说明

1. 本次实践面采取 ACEE 试题提交系统的进行作答。试题提交地址为 <https://exam.castle.top>（可用校外 ipv6）、<http://10.70.48.113>（仅限校内 ipv4），登录的帐号为学号，密码为你的姓名拼音首字母小写 + 教务网登记的手机号后六位。（例如：龚皋同学的手机号后六位为 123456，则他的网站帐号为 3240101994，密码为 gg123456。）请使用最新版 Chromium 内核浏览器浏览网站，并按照网站上的相关提示提交你们组的作答。
2. 若网站无法正常提交，请联系联络员反馈。
3. 我们将对过时提交文件的团队进行扣分或者取消资格处理。

## 三、结题面试说明

1. 团队展示（即结题面试）将在 2025 年 3 月 23 日周日 16:00-18:00 进行，各组的具体时间地点待后续通知。
2. 每组展示时长为 15 分钟。

3. 展示材料 (如 PPT 等) 请于 2025 年 3 月 23 日周日 12:00 前提交至网站对应位置。

## 四、安全相关说明

在开始本次实践面之前，请于开题面试前务必仔细阅读 **Assignment B Part 0** 中的焊接安全相关提示部分，并按要求观看相应视频。在开题面试时需签署安全承诺书。





## Pre Assignment: Git Usage

### 导言

Have you ever heard about GitHub? It's a sharing platform for numerous open source projects.

Have you ever wondered why the platform is called "GitHub"? It's mainly because its base software "Git".

Git is a version control software, which is used to take a snapshot of your repository, which you could roll back at any time. Here is a brief introduction tutorial: [About Git](#)

You can also read other tutorial you prefer, after learning basic git operations, you need to accomplish following tasks.

If the task procedure has prepend with symbol "&", please attach **screenshots or record** of each **success** operation in your report.

**Note:** Throughout this assignment, we will use main as the default branch name. Historically, Git used master as the default branch name, but many projects now use main instead to adopt more inclusive terminology. Depending on your Git version and configuration, you might see either name used by default.

### 提交说明

1. 本题不需要提交任何代码，但是需要一篇使用英文撰写的解题报告
2. 报告需要包括：
  - (a) 所有问题的题干和解答
  - (b) 题目规定的截图或命令行输出
3. 在报告的最后，你们需要体现解题过程中所遇到的问题，以及你们的解决方案
4. 本题意在培养或强化你们通过 Git 进行代码管理和团队协作的能力，希望你们每个人都可以尝试做这道题目，并使用 Git 来提升后续 Task 的协作效率。

**特别提醒：**截图或命令行输出的示例：

如果选择提交屏幕截图

- 请确保截图清晰，能分辨出每个字符
- 请确保不遗漏重要信息
- 请确保隐去你们的个人信息



```
(base) /example$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /example/.git/
```

Figure 1: 屏幕截图示例

如果选择提交命令行输出

- 请将其放在代码块中，或者使用 Polacode 等插件创建代码截图
- 请隐去你的个人信息，并将除了 git 存储库所在的文件夹名称用 /path 代替

Listing 1: example\_output

```
1 /path/example$ git init
2 hint: Using 'master' as the name for the initial branch.
   This default branch name
3 hint: is subject to change. To configure the initial
   branch name to use in all
4 hint: of your new repositories, which will suppress this
   warning, call:
5 hint:
6 hint:   git config --global init.defaultBranch <name>
7 hint:
8 hint: Names commonly chosen instead of 'master' are '
   main', 'trunk' and
9 hint: 'development'. The just-created branch can be
   renamed via this command:
10 hint:
11 hint:   git branch -m <name>
12 Initialized empty Git repository in /path/example/.git/
```

```
/path/example$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /path/example/.git/
```

Figure 2: 控制台输出示例

## Part 1 First Repository: First Commit

In this part, you need to create your first repository.

### Task 1: Initialize Repository

1. Create a new folder
2. & Initialize your repository use following command

```
1 git init
```

This would initialize a git repository in your current folder.

3. & Add some files and track them

After you add some file into your repository, run

```
1 git status
```

This would status your current status of your repository. Then tracking your file with

```
1 git add <file_name>
```

**Question:** What changes after tracking your file? Write down your comprehension of status of each file. (Untracked, Modified, Staged, Committed)

#### 4. & Commit your files

```
1 git commit -m "First Commit"
```

You can change message to whatever you like.

**Question:** What happens if you didn't track the file? Which status of file will be committed?

#### 5. & Inspect difference between two commits

- First, modify some of these files and commit again, run

```
1 git log
```

This will show your recent commits.

**Question:** What is the **HEAD** pointer?

- Then compare the difference of your recent two commits with its id.

```
1 git diff <commit1> <commit2>
```

## Part 2 Branching: Merge, rebase...

In this part, you'll explore branching, merging, rebasing, and learn to handle conflicts that arise during these operations.

### Task 2: Merging Branches and Handling Conflicts

Now, let's dive into Git branches, which allow you to work on different versions of your project simultaneously.

#### 1. Make a new branch

- & First, create a new branch called feature

```
1 git branch feature
```

- & Switch to the feature branch

```
1 git checkout feature
```

- Modify a file, then stage and commit it

**Question:** What's the difference between committing to main branch and feature branch?

2. Make conflicting changes on main

- Switch back to main.
- Modify the same file.
- Create a commit.

3. Merge these branches

- & Merge feature into main:

```
1 git merge feature
```

- Git will report a conflict. Check the status:

```
1 git status
```

- Open conflict file. You'll see conflict markers:

```
1 <<<<<< HEAD
2 <Main change>
3 =====
4 <Feature change>
5 >>>>>> feature
```

- & Resolve the conflict, then stage and complete the merge:

```
1 git add <conflict_file>
2 git commit
```

Congratulations! Now you have solved the conflict. Then, think about

**Question:** What causes conflicts? Why git didn't merge these conflict files automatically?

4. Inspect the Merge History

- & View the commit history:

```
1 git log --oneline --graph
```

### Task 3: Rebasing Branches and Handling Conflicts

1. Create a branch for rebasing

a. Create a new branch:

```
1 git checkout -b feature-rebase
```

b. Modify some files, then commit again:

2. Make changes on main branch

3. & Rebase feature-rebase onto main, run

```
1 git rebase main
```

Git will pause due to a conflict. Check the status:

```
1 git status
```

Resolve the conflict, stage it, and continue:

```
1 git add <conflict_files>
2 git rebase --continue
```

4. & Inspect the rebase History

```
1 git log --oneline
```

**Question:** How does rebasing work? What's the difference between merge and rebase?

5. Merge the rebased branch

• Switch to main:

```
1 git checkout main
```

• & Fast-forward merge:

```
1 git merge feature-rebase
```

• & Inspect the history:

```
1 git log --oneline
```

**Question:** In which condition could "Fast Forward"? What's the advantage of rebasing compared to merge directly?

## Part 3 Remote: Working with others

### Task 4: Remote Repository Operations

Now, let's connect your local repository to a remote one (e.g., GitHub).

1. Create a remote repository:

- Go to GitHub (or another Git hosting service) and create a new repository (e.g., my-first-repo).
- Copy the repository URL (e.g., `https://github.com/username/my-first-repo.git`).

2. Link your local repository to the remote:

```
1 git remote add origin <repository_URL>
```

3. & Push your local commits to the remote:

```
1 git push -u origin <branch>
```

**Question:** What does push do? Why might you need the `-u` flag?

4. & Fork Repository

Use another account to fork the repository. Then clone it to local machine.

```
1 git clone <forked_repository_URL>
```

Make some changes and commit and push.

**Question:** What's the relationship between forked repository and origin repository. Will changes in forked repository affect origin repository?

5. & Create a Pull Request:

- On GitHub, create a pull request (PR) in origin repository use another account.
- Accept PR using origin account.

**Question:** What's the purpose of a pull request?

6. & Pull changes from the remote:

After merging the PR on GitHub, update your local repository:

```
1 git pull origin <branch>
```

**Question:** What does pull do? Explain how pull request contributes in team cooperation.

## Bonus .gitignore: Ignoring Unnecessary Files

### Task 5: Be Careful when Pushing

Well, you may find git useful, and maybe you can't wait to push all your folders to your repositories. But wait, **BE CAREFUL BEFORE PUSHING!**

You may notice that, generally speaking, code is usually the scale of kb or mb, while data is often the scale of gb. Therefore, if you put your data and code in the same folder and push everything to GitHub at once, both the uploading and downloading processes will be quite painful.

You could certainly separate your data and code, but here we will adapt another approach: **adding a .gitignore file.**

#### 1. understand .gitignore

- Check the attachments provided for this task
- Create an appropriate .gitignore file for the repository
- & Commit a clean version of the folder that :
  - Keep all necessary JavaScript files (\*.js)
  - Remove all folder start with "."
  - Keep Package.json and README.md
- Ensure the functionality of the repository is not affected. (main.js and its related files)

**Question:** What's the contents of .gitignore? Why do we need .gitignore? What's the difference between \*.tmp and /tmp/\*.tmp in a .gitignore file? How can you archive a clean repository?

**Note:** You should not delete any files directly from the attachments. The .gitignore file should handle excluding unwanted files.

#### 2. Archiving your repository

- & Create an archive of your repository:

```
1 git archive --format=zip HEAD -o repository_archive.zip
```

**Question:** What's the difference between archiving a repository and simply making a copy of the directory?

**Note:** Please commit your final clean archive.



## 导言

——匿名

Hex 棋, 又称 Nash 棋、六贯棋, 最初在丹麦数学家海恩于 1942 年 12 月 26 日在丹麦报纸 Politiken 发表的一篇文章里出现, 当时称为 Polygon。1948 年, 约翰·福布斯·纳什重新独立发明了它。Hex 棋的棋盘由六边形格子组成, 形成一个菱形。两位玩家分别使用不同颜色的棋子进行对弈。目标是用自己的棋子连接棋盘的两个相对边, 形成一条连续的路径。具体来说, 一位玩家需要将棋盘的上下两边连接起来, 另一位玩家则需要将左右两边连接起来。谁先完成这一目标, 谁就获胜。

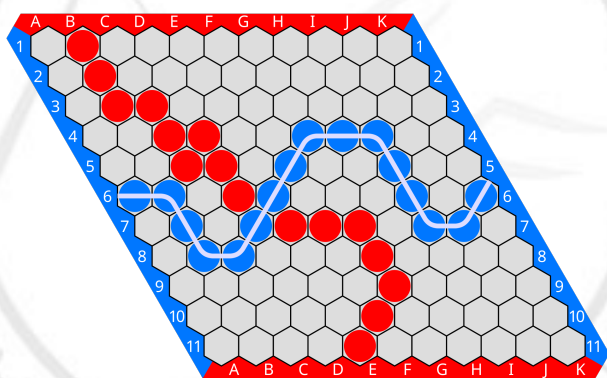


Figure 3: Hex 的棋盘

虽然规则简单，但 Hex 棋的策略非常复杂。不难证明，在棋盘填满后，必然有一方能够连通自己的目标边，所以 Hex 棋无平局。又根据博弈论的策梅洛定理可知：先手或后手当中必有一方有必胜(必不败)的策略。而即便后手有必胜策略，先手也可以将自己转化为等效后手，从而获得必胜策略。因而必存在先手必胜策略。即便如此，已经有研究证明在任意图上的广义 Hex 游戏中判断某个位置是否为必胜位置属于一个 **PSPACE-complete** 问题。换句话说，如果存在该问题的多项式时间解法，则可证明  $P=NP$ ，而这至今仍是一个未解之谜。以  $11 \times 11$  的 Hex 棋为例，其状态空间大小就约为  $2.4 \times 10^{56}$ 。

Hex 棋的复杂性和策略性使其具有较高的竞技性和研究价值，近年来也成为人工智能研究的焦点之一。

## 规则界定

在本题中，我们进行如下规定：

- 棋盘的大小通常为  $n \times n$ ，在本题中我们规定棋盘大小为  $11 \times 11$ 。
- 对弈双方连续相互走棋，已经下在棋盘上的棋子不能移动或消失。
- 红色方（即后文棋盘坐标系中的字母边）为先手
- 由于 Hex 存在显著的先手优势，所以引入**首步交换原则**，即第一步棋落子后，对方可选择将这颗棋子换成自己的棋子，并将其放到与原棋子位置关于棋盘对角线对称的位置上。先后手随机决定。
- 使用同色棋子将棋盘对边连接起来的一方获胜。

如无特别说明，本系列题目涉及的 Hex 棋规则均以上述版本为准。

## Part 1

### Question 1.1

本题导言的第一段话来自某位匿名网友在互联网上的评论。这一段发言隐含了关于如何设计计算机博弈程序的一些思路。请就这段话，谈谈你们的想法。

### 提示

1. 本题旨在加深大家对计算机博弈的理解，你们可以畅所欲言，我们对字数没有要求，表达清楚意思就行。
2. 本题带来的思考也许可以给你们在接下来的题目中起到一点帮助。

### Question 1.2

龚学长开始着手学习计算机博弈。在互联网搜索的时候，他发现了实现 Hex 智能的很多方法。他打算着手去了解这些方法。其中出现最多的两种方法是 **alpha-beta 剪枝结合评估函数**和**蒙特卡洛树搜索结合 UCT 算法**。现在请你们自行搜索相关的一些文档，了解这两种算法，并在本题比较这两种方法的异同，写在解题报告中。

## Part 2: Hex 博弈程序实现

龚学长最近在学习计算机博弈，他对 Hex 棋的博弈程序产生了兴趣，想要自己开发一个关于 hex 棋的智能模型。但是想要实现这样一个模型的方法非常多，怎样能够实现一个能力强大，可以下过大部分棋手的模型呢？

在本题，请查阅相关资料，编写一个能够进行 Hex 棋对弈的模型，并将你们的模型原理，关键参数进行解释，写入解题报告中。

在这里，你们需要让你们的模型尽量作出合理的解，在最后的测评过程中，我们会将你们的模型两两进行对弈，得到结果，进行积分制排名，作为分数评价的部分依据。也就是说，你们可以使用任何你们想得到的方法（不包括调用外部程序接口）来优化你们的程序，使其棋力提升。并且你们要尽可能使模型的响应时间更短，这意味着你们需要做很多的剪枝优化。平均响应时间是我们给模型打分的部分依据。

### 参考资料

1. [Hex \(board game\)](#) —— Wiki 百科上对于 Hex 棋的介绍
2. [Hex Wiki](#) —— 属于 Hex 棋的 Wiki
3. [playhex](#) —— 你们可以在此与机器人对弈
4. [minimax 算法和  \$\alpha - \beta\$  剪枝](#)
5. [Hex 棋评估函数设计示例](#)
6. [蒙特卡洛树搜索的介绍](#)
7. [蒙特卡洛方法->蒙特卡洛树搜索->UCT 介绍](#)

### 提示

1. 比起从查阅大量的资料或规则开始，或许从与同学或机器人的对弈开始能让你们更好地理解 Hex 棋的规则，发现致胜的秘诀。
2. 我们并没有对模型的实现做出任何限定，只要符合接口的规范，你们可以用任何你们能想到的方法或算法来实现你们的 Hex 棋智能。
3. 我们鼓励独立的、创造性的工作，在报告中向我们展示你们对本题独到的见解和剖析，而非简单地复刻现有的算法，we will appreciate it.
4. 你们可以大胆地参考任何你们能找到的资料甚至开源代码，但请不要抄袭已有的工作。我们会对你们提交的所有源代码进行查重。并且再次重申前一点中我们提到的，我们鼓励独立的，创造性的工作。

5. 尽情使用 AI 工具来提高你们的生产力。如果你们尚未拥有能够稳定与 AI 对话的途径，至少你们还可以试试[ZCHAT](#)——对浙大学子来说，应该能免费访问很多模型。如果想更进一步使用 AI 工具辅助开发，你们还可以试试[trae](#)，在它刚发布的这段时间里，你们可以借助它免费使用 Claude-3.5-Sonnet 和 GPT-4o。但请不要直接使用 AI 给出的代码，这可能会使你们的代码无法通过查重。
6. 我们提供了一些推荐使用的方法，可以供你们参考：
  - alpha-beta 剪枝结合评估函数；
  - 蒙特卡洛树搜索与 UCT 算法；
  - 蒙特卡洛树搜索与神经网络；
7. 尽管神经网络不失为一种可行的方法，但通过这种方式能否在短时间内得到一个可靠的 Hex 棋智能具有高度的不确定性。因此，除非你们对自己的实力有足够的自信，不然我们更加推荐通过其他更加成熟稳定的方式来实现。
8. 本题是一道开放性题目，Hex 棋的智能有很多不同的具体实现方法，我们提供的参考仅仅用于在你们没有思路的时候提供一点点灵感，请不要拘泥于我们所提供的资料。

## 具体要求

你们的模型应当能够接收对手的落子，理解棋局，并能够在不犯规的前提下朝着胜利正确地下出自己的每一步棋。

同时为了让测评过程能够顺利进行，你们实现的模型应当服从统一的接口标准——即输入输出的模式应当符合我们定义的规则。阅读以下规则或许也可以帮助你们明确需要做的任务。

### 1. 输入输出说明：

- (a) 输入：每次输入为一条信息 + 一个换行符。相当于在终端键入一行字符然后按下回车。
- (b) 输出：每次输出应为一条信息 + 一个换行符。相当于在终端输出一行字符然后换行到下一行。

需要特别注意的是：你们的模型应当在输出结果时清空标准输出缓冲区，以免结果卡在缓冲区中导致长时间未能被读取而被误判超时。以 C 语言为例，只需要在每个 `printf()` 后跟上一句 `fflush(stdout)`。当然，实际上你们可以使用各种方式来达到这个目的。

### 2. 游戏开始：

- (a) 先手：从裁判处接收字符串 `"first\n"`，则表明本轮游戏中你们的程序为先手，接下来你们的程序应当输出第一步的落子坐标并换行。简单来说，这相当于你们运行了你们实现的算法程序，然后就像往常一样在终端键入 `"first"` 这五个字符，然后按下 `enter` 键（对应 `"\n"`）向程序发送了这条信息，然后你们的程序在下一行返回了如 `i2` 这样的坐标，表示自己的落子位置。



- (b) 后手: 若程序启动后从裁判处接收到的不是"first", 而是一个落子的坐标, 那么表示你们的程序为后手。

### 3. 交换规则:

- (a) 当你们的程序为后手时, 你们的程序有权利在得到对手的第一步落子信息之后 (得到的第一个输入是坐标, 表明自己是后手), 选择将对手的落子变为自己的颜色, 然后将其置于原落点关于棋盘对角线对称处。具体而言, 对称落点的计算依照如下规则, 将原落点的“横纵坐标”分别转化为序号, 然后交换他们的位置, 再转化为字母 + 数字的表示方式。如原落点为"i2", i 字母序为 9, 交换 92 顺序得 29, 则字母表中序号为 2 的是 b, 所以得到坐标"b9"。
- (b) 如何交互: 在接受到对手的坐标输入后, 你们的程序可以选择直接输出一个坐标, 或者输出字符串"change", 此时"change"也将被发送给先手程序, 先手程序要依此判断后手选择了交换。如果输出坐标, 则认为你们的程序选择落子, 如果输出"change", 则认为你们的程序选择使用交换规则。
- 无论你们如何选择, 下一步的控制权都将转交给你们对手, 你们的程序将等待下一个坐标输入。

### 4. 坐标表示:

- (a) 落子均以坐标的形式表示, 其中坐标的定义以图3为准, 即蓝边每行为数字 1, 2, ..., 11, 红边每列为字母 a, b, ..., k。
- (b) 你们的程序应该在每轮输出形如 "i2\n" 这样的字符串, 表示模型给出的落子。第一个字符为"a"-"k"之间的小写字母, 第二个部分为 1-11 之间的阿拉伯数字。
- 注意: 第一个字母应为小写而非大写, 第二个部分可能为 11, 非单字符。

### 5. 游戏进行:

- (a) 在每一回合中, 当前执棋方需要根据当前的棋局给出自己的落子, 以坐标的形式输出。而另一方则通过输入得到对手的落子, 更新自己的棋盘, 给出自己的落子。然后不断重复直至胜负已定。
- (b) 对于模型而言, 在游戏开始部分结束, 即交换规则结束之后, 只需要不断重复“接受坐标输入”, “输出落子坐标”的过程即可。并不硬性要求实现胜负判断的功能, 裁判会判断游戏的局面, 并在胜负已定或一方犯规时终止比赛。

### 6. 游戏结束:

- (a) 游戏结束分多种情况, 一方连通两边取得胜利, 也可能是另一方“花式犯规”导致失败。关于可能出现的犯规在后文会有简单提示。
- (b) 当游戏结束时, 裁判会向程序发送字符串"finish\n", 表示游戏终止。当程序接收到该输入, 而非坐标输入或者"first"或者"change"时, 应当结束程序。

## 7. 犯规:

- (a) 每一步棋限时 10 秒 (从接收到对手落子开始计时直至给出下一步落子), 超出这个时间依旧没有响应 (裁判程序会提供零点几秒的余裕) 将会被视为模型卡死, 将会判负。
- (b) 给出不符合规范的输出, 如在非后手第一轮的时候输出 "change", 或者落子输出格式不符合 `f"{小写字母}"{数字}"` 的格式, 例如数字超出 11, 均会判负。
- (c) 给出的落子坐标处已有棋子, 无法落子, 则会判负。

## 提交要求

1. 提交一份报告, 在报告中阐明你们的设计思路, 以及模型使用的算法及其原理。将报告渲染为 pdf 格式, 命名为 `report.pdf`, 存放于 **附件/report/**。注: 比起模型实际的表现性能, 我们或许会更在意你们原创性的设计思路, 即便模型表现不佳, 报告也将会作为重要评分依据。
2. 将模型相关工程文件打包放于 **附件/model/**, 并提供 `readme` 文件, 其中包含在 **附件/**目录下能够直接运行模型的命令以及模型所依赖的环境配置, 若使用了需要安装的库或依赖均需详细说明安装方式。你们需要确保你们的程序能够在 windows 平台上运行。

## 中期提交 (可选)

本题中, 我们不允许组与组之间相互交流, 考虑到同学们对于模型能力的把控较难, 我们设置了中期提交环节。大家可以选择在 **3 月 21 日 23:00** 之前先提交一个半成品的模型 (模型输入输出具体要求同最终版本), 我们会根据大家给出的模型进行相互对弈, 并在 3 月 22 日把最终输出的棋谱返还给大家, 这样大家就能够了解到自己与其他组之间的差异, 并进一步进行优化, 提升模型能力。

注意, 这部分内容为选择提交, 如果你们小组在提交截止前还未完成相关的工作, 或是并不打算提交, 可以选择不进行提交。本次提交只是为了你们能够对自身小组模型能力有一个较好的理解。

提交要求为**最终提交要求的第二点** (不需要提交报告, 但是需要 `readme` 文件解释环境配置等信息)。

## Bonus 1

这里提供了该部分供选择的 Bonus Task, 若你们实现了相关功能, 请写到解题报告中。

## 1. 结束情况判断

在我们的测评程序中, 我们提供了判断当前棋局是否结束的功能并且会自动停止当前对弈。而现在我们希望你们的程序自己能够判断这一点。请设计一个算法, 来判断当前的 Hex 棋局是否已

经胜利或失败，并接入到你们的程序中。

请在你们的报告中阐明你们结束情况判断的设计思路。

## 2. 特殊棋形策略库

在 Hex 棋对弈的进行过程中，会有一些特殊棋形产生。针对这些棋形的特点，我们可以设计一些特定的策略库，以便于模型快速高效地运行。

这些特殊棋形包括但不限于：

- 无效点位：对于对弈双方都无用的点位

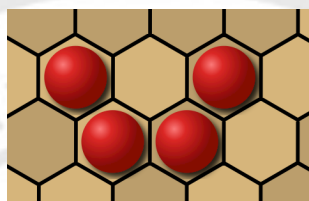


Figure 4: 无效点位示例，四颗红棋子围成的中间点为无效点位

- 边界点位：与边界构成了稳定结构

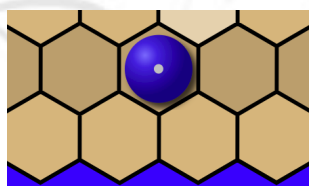


Figure 5: 边界点位示例，蓝棋子与边界点位构成稳定结构，红方无法将其断开

对于这些特殊棋形，模型可以不用再考虑某些位置是否需要走棋。同样的，对于某些特殊棋形，模型可以准确地考虑某些棋形需要在哪里走棋。这增加了模型的精确度，也减少了响应时间。更多的特殊棋形可以参考：[Hex Book](#)。也可以自己多与同学或电脑进行对弈，可以发现一些特殊棋形。

我们会根据特殊棋形的详细程度以及针对每个棋形的策略的优劣进行打分。请注意：只有实现在模型中的部分会被记入得分。

## 3. 开局库

一般来说，由于搜索点位过多，开局点位的选取会耗费一个模型较多的时间。而在我们固定棋盘的情况下，也许可以对最开始的几步对弈建立一个策略库。这样不仅可以加快模型的反应速度，也可以确保在某些情况下获得最佳开局。你们的开局库可以通过自己的博弈程序输出结果而来，也可以是根据你们手工编写。不管使用哪种方法，请在你们的报告中阐明你们开局库的设计思路。

我们会根据开局库的详细程度与策略的优劣进行打分。请注意：只有实现在模型中的部分会被记入得分。



## 4. 其他优化

你们可以在这一部分的解题报告中写上你们对自己的 Hex 博弈程序做的其他优化，我们会根据内容酌情加分。

## Part 3: Hex 可视化棋盘

龚学长在研究 Hex 棋的解法与策略，但是苦于没有一个合适的电子棋盘。他希望使用前端知识制作一个网页棋盘，为此你们可能需要学习关于 HTML、CSS、JavaScript 的相关知识。

HTML (HyperText Markup Language, 超文本标记语言) 是用于构建网页内容结构的基础语言。它通过一系列的标记符号 (标签) 来定义网页中的元素。这些标签告诉浏览器如何展示页面的内容，从而构建出网页的基本框架。HTML 是网页开发的基石，为网页提供了可读和可交互的内容结构。

CSS (Cascading Style Sheets, 层叠样式表) 是一种用于控制网页外观和样式的语言。它通过定义样式规则来设置 HTML 元素的外观，包括字体、颜色、间距、布局等。CSS 可以让网页的视觉效果更加美观和统一。CSS 的层叠特性允许开发者定义通用样式，同时也可以针对特定元素进行覆盖和细化，从而实现复杂而灵活的页面设计。

JavaScript 是一种运行在浏览器中的编程语言，用于为网页添加交互性和动态功能。它可以操作 HTML 和 CSS，实现诸如响应用户点击、单表验证、动态内容更新等功能。它为网页带来了丰富的交互体验，使网页不仅仅是静态的展示内容，而是可以与用户进行实时互动的动态应用。

在这里，你们需要完成一个网页，以实现一个可以交互的 Hex 棋盘。

### 参考资料

1. [Web 开发基础知识](#)
2. [Vue 官方文档](#)
3. [React 官方文档](#)

### 提示

1. 建议使用[Live Server](#)作为网页构建工具，若使用 VS Code，可以直接搜索其扩展。
2. 若你们想要增加棋盘的功能，可以学习使用 Vue、React 等 JavaScript 框架。
3. 在附件 [code/hexworld/](#) 子文件夹中，提供了一个棋盘的示例程序，可以通过使用[Live Server](#)进行运行，或直接使用浏览器打开 index.html 进行查看。

## 具体要求

1. 学习并使用 HTML, CSS, JavaScript 完成网页的构建。若需使用诸如 Vue, React 等 JavaScript 框架, 请使用 CDN 将其集成到 HTML 中, 确保在未配置这些框架的情况下仍可以运行。
2. 棋盘的大小为  $11 \times 11$ , 默认棋子颜色为红蓝, 其中红方先落子, 蓝方后落子。
3. 棋盘通过点击的形式交替落子, 即下了一步棋后, 下一颗棋子要换颜色。注意需要满足首步交换原则, 即红方下完第一步后, 蓝方可以通过点击红方下的那颗棋子 (其他形式也可) 来完成转换。网页需要具备清空棋盘等基础功能。

## 提交要求

1. 录制一个 demo (视频不得过长, 大小最好不超过 200MB), 以展示你们的棋盘的功能。你们可以自行选择是否进行讲解。将其命名为 checkerboard\_demp.mp4 (或是其他常见的视频格式), 存放于 **checkerboard/** 子文件夹中。
2. 将整个网页的工程文件打包压缩为 hex\_checkerboard.zip。其中需要包括棋盘主页面的 checkerboard.html 文件和所有其需要调用的 CSS 与 JavaScript 文件。要求压缩包解压后可以直接使用 [Live Server](#) 等网页构建工具直接打开。
3. 在解题报告中介绍网页的所有功能, 关键步骤要求重点阐释。

## Bonus 2

这里提供了该部分供选择的 Bonus Task, 若你们实现了相关功能, 请写到解题报告中。这会使你们的工作获得更高的分数。

### 1. 棋盘样式控制

不同人对于棋盘的需求不同。在你们的网页中设计一个可以控制棋盘样式的控件, 包括但不限于棋盘的大小、棋盘放置的朝向、棋子的颜色。我们会根据操作的完善程度以及程序的鲁棒性进行打分。

### 2. 撤回列表

有时我们需要返回到之前的某一步重新进行走棋, 这时需要将原来的走棋进行撤回, 请设计一个回溯列表, 能够通过点击将棋盘回溯到之前的某一个状态。我们会根据操作的完善程度以及程序的鲁棒性进行打分。

### 3. 保存棋谱

我们希望当前的棋谱能够作为文件被保存下来，请设计一个保存文件的方式，以便于你们的网页在之后可以重新读取棋谱内容。我们会根据操作的完善程度以及程序的鲁棒性进行打分。

### 4. 接入智能模型

我们希望能够通过你们的棋盘与你们设计的智能模型对弈，请将你们的智能模型接入到网页中，并设置相应的控件，使用户可以选择与你们的智能模型进行对弈。我们会根据操作的完善程度以及程序的鲁棒性进行打分。

### 5. 其他优化

你们可以在这一部分的解题报告中写上你们对自己的 Hex 棋盘做的其他优化，我们会根据内容酌情加分。

### 提交文件夹说明

本题中，你们需要在 A 题提交的对应位置上传名为组号\_A.zip 的压缩包。文件夹结构如下：  
组号\_A/

- T1
  - report
    - \* report.pdf
  - model
    - \* README
    - \* 模型所需的代码
- T2
  - video
    - \* checkerboard\_demo.mp4 （或是其他常见的视频格式）
  - checkerboard
    - \* checkerboard.zip

## Assignment B: 嵌入式开发实践

在某个秋高气爽的下午，嵌入式的狂热爱好者龚学长看着小龟逐渐下降的充电功率突发奇想，想要收集手机充电功率随时间、当前电量的变化曲线。但由于最近忙着工高招生，他只设计了 USB 充电功率测量设备的硬件部分……

### Part 0 准备工作

本小题不评分，仅为开始答题前的准备工作。

#### 背景介绍

本题的主要目标是实现以 **STM8** 为主控的 **USB 充电功率测量仪**。此设备的主要功能如下：实时读取充电器与充电线之间的充电电压、电流，进而计算总功率。在调试阶段，测量仪可将读取信息通过串口回传给上位机，由上位机进行数据处理，但这种方式折损了便捷性。为提高灵活性，你们制作的测量仪必须带有**不依赖上位机的独立用户界面**，即能够通过其上的屏幕、按钮等部件独立工作。

**Bonus** 的主要目标是实现 **STM8** 控制的 **NFC 终端**。你们需要将 STM8S105K4T6 最小系统板连接至 PN5180 射频前端模块，实现对市面上最常见的 M1 卡（Mifare Classic 1K）的读取、写入、破解、卡模拟功能。

请注意：在本题中，各小题并不按照严格的实现顺序排列。请做好对任务的整体把控，合理分配组内任务，尽可能实现全组多线程工作，并及时交换信息，避免出现重大纰漏。

#### 开发环境搭建

本次阅卷采用 **IAR Embedded Workbench for STM8** 作为开发环境。因此，请确保最终提交的工程文件能够被 IAR Embedded Workbench for STM8 **v3.10** 直接打开并成功编译、烧录到芯片中。否则，根据情况的严重程度，你们将受到不同程度的扣分。

请统一使用嘉立创 **EDA 专业版** 进行原理图绘制与 PCB 设计，并确保最终提交的 .epro 工程文件能够被嘉立创 **EDA 专业版** 直接打开查看。否则，根据情况的严重程度，你们将受到不同等级的扣分。嘉立创 EDA 专业版可直接在浏览器中使用。

#### AI、第三方库使用规则

本题涉及较大体量的信息检索与阅读。我们鼓励使用 AI 快速寻找有效信息，但请严格遵守以下四点要求：

1. 最终题目报告中不得含有 **AI 生成内容**。题目报告中需展示你们**真实**的阅读结果、实践体悟、分析流程、核心思路、代码解析。

2. 所有引用信息必须溯源至原始技术文档/论文/标准等可信来源，并在报告中标注明出处。
3. 禁止使用 AI 直接生成整个文件的代码，你们必须最终独立地实现本题的核心内容。
4. 不得直接调用第三方库函数完成代码的核心、关键部分，允许调用第三方库完成次要的、枝节的代码逻辑。

若违反上述要求，根据情况的严重程度，你们将受到不同程度的扣分。

请注意：由于库函数的编写逻辑和面向功能的工程编写逻辑存在明显差异，套用现有库函数通过阅读代码即可发现。

## 焊接安全提示

请务必在进入实验室前仔细阅读本部分内容，并在开题面前完成安全视频的学习，于开题面时仔细阅读并签署《安全承诺书》。

请注意！焊接工作需要使用带有一定危险的电烙铁，其工作温度可能达到 300 度甚至更高，若使用不当，可能导致严重烧伤，甚至引起火灾。

以下为必要的焊接安全提示：

- 烙铁头高温危险：始终假设烙铁头处于高温，禁止触碰
- 烙铁不离架：不用时务必放回烙铁架，避免触碰其他物品，禁止平放在桌面上。
- 断电检查：使用后立即拔掉电源，确认烙铁完全冷却再离开。
- 镊子操作：务必用镊子夹取工件辅助焊接；焊接后，焊点处的元件以及周围的 PCB 都处于高温，用镊子夹取或等待冷却。
- 远离易燃物：工作台保持整洁，远离纸张、布料、酒精等易燃物。
- 通风环境：在空气流通处操作，或使用排气扇/口罩减少吸入焊锡烟雾。
- 专注操作：焊接时禁止分心（如玩手机、转头聊天）。

谨记：手不离柄，眼不离焊，用完立即断电。

## 焊接教程

1. [BV1oV41127Mz](#)
2. [BV1Se4y1S78K](#)
3. [安全教育视频](#)，请务必在开题面前完成观看。



## 物料清点

请于开题面后前往月牙楼 301 领取所有物资，并根据下表仔细核对，其中包含了本题可能使用的所有元器件、材料与工具。物料申领后需当场清点，如有缺漏需当面提出并补齐。请注意：任何当场申领结束后的补领行为，一律视作焊接中损坏，请在 **Part 3** 中完整记录。

实验室的开放时间段为**3 月 21 日（周五）8:00-22:00，3 月 22 日（周六）8:00-12:00**。请合理规划时间，在规定时间内完成焊接。

编号	名称	标号	封装	数量	制造商标识
1	USB-231-ARY	USB1	USB-A-TH_USB-231-ARY	1	USB-231-ARY
2	USB-212-BCW	USB2	USB-A-SMD_USB-212-BCW	1	USB-212-BCW
3	CC6920BSO-5A	U2	SOP-8_L4.9-W3.9-P1.27-LS6.0-BL	1	CC6920BSO-5A
4	STM8S003F3P6	U4	TSSOP-20_L6.5-W4.4-P0.65-LS6.4-BL	1	STM8S003F3P6
5	AMS1117-3.3	U5	SOT-223-4_L6.5-W3.5-P2.30-LS7.0-BR	1	AMS1117-3.3
6	PZ254V-11-04P	H1	HDR-TH_4P-P2.54-V-M	1	PZ254V-11-04P
7	NCD0603G1	LED1	LED0603-RD	1	NCD0603G1
8	NCD0603R1	LED3	LED0603-RD_RED	1	NCD0603R1
9	4.7k $\Omega$ 0603WAF4701T5E	R15,R16	R0603	2	0603WAF4701T5E
10	10 $\mu$ F CL10A106KP8NNNC	C9,C10	C0603	2	CL10A106KP8NNNC
11	100nF CL10B104KB8NNNC	C11,C12,C16,C17,C18,C19	C0603	6	CL10B104KB8NNNC
12	750 $\Omega$ 0603WAF7500T5E	R17,R18	R0603	2	0603WAF7500T5E
13	10k $\Omega$ 0603WAF1002T5E	R22,R23,R24,R25,R26,R19,R33	R0603	7	0603WAF1002T5E
14	1 $\mu$ F CL10A105KB8NNNC	C15	C0603	1	CL10A105KB8NNNC
15	OLED-0.96IN	U12	OLED-0.96IN	1	
16	LM321	U13	SOT-23-5_L3.0-W1.7-P0.95-LS2.8-BR	1	LM321
17	KH-2435-CAJ KH-2435-CAJ	SW3,SW2,SW1	SW-SMD_KH-2435-CAJ	3	KH-2435-CAJ
18	33k $\Omega$ 0603WAF3302T5E	R29,R30,R31,R32	R0603	4	0603WAF3302T5E
19	电烙铁（公用）			1	
20	焊锡丝（公用）			1	
21	镊子（公用）			1	
22	USB 充电测量仪 PCB			1	
23	ST-LinkV2			1	
24	杜邦线			若干	
25	PN5180 射频前端模块			1	
26	STM8S105K4T6 最小系统板			1	
27	CH341A USB 转并口模块			1	
28	带数据 CUID 卡			1	

Table 1: 材料清单

## Part 1: 原理图与 PCB 识读

请查看附件中的 ProPrj\_EDAProj.eprj，并将其导入进嘉立创 EDA 专业版中，即可查看工程的两版原理图与 PCB 设计。如遇到困难，可以参考附件中的主控芯片的数据手册、设备的原理图。请查阅相关资料，回答以下问题，并将详细解答写入解题报告。

参考：[嘉立创 EDA 专业版用户指南](#)

### Question 1.1

我们定义第二版原理图中的一个方框为一个电路单元。

1. 请画出各个单元之间关系的思维导图，理清设备整体的工作原理；
2. 请分条解释各个单元的功能、工作原理以及设计理由；
3. 请指出主控可以复用功能的引脚以什么工作模式进行工作。

### Question 1.2

PCB 的设计往往需要反复尝试与改进，本题中的两版工程也展现出了改进过程。

1. 请找出第一版工程与第二版工程在电路设计上的不同；
2. 请指出第一版原理图存在的问题，并解释第二版原理图做出相应改动的原因。

### Question 1.3

请在完成后续题目后，指出第二版工程在电路设计或 PCB 设计上还存在哪些可以改进的地方？

请注意：请使用自己的语言撰写本题的 report。若检测到大量 LLM 生成的文字，你们将会受到一定程度的扣分。

## Part 2 原理图与 PCB 设计

龚学长希望你们能协助他完善硬件设计。目前，该设备的核心功能是实现电压和电流的采集。若要对硬件进行改进，一方面要完善核心功能，另一方面要增加新的功能。在本部分中，请根据以下提示，提交一个名为 2 Project.eprj 的嘉立创 EDA 工程文件，以帮助他完善设计细节。



## Question 2.1

要完善核心功能，一个最重要的方向就是在保持**成本**在合理区间的同时，提高产品**性能**，减小电压和电流噪声的峰峰值。为了实现以上目的，你们需要进行如下改进：

1. 自主选型（任意选择市场在售的合适的芯片、元件），重新设计原电路的供电电路、电压电流采集电路；在电压和电流的采集电路的合适位置添加二阶有源低通滤波电路，截止频率低于 100Hz；
2. 详细解释你们的选型思路，即在横向比较中，为何选择该厂商、该类型的芯片；
3. 根据上网搜集到的芯片、元件的理论参数，证明在输入电压为 5V、输入电源的纹波等性能保持一致的工作条件下，电压、电流噪声峰峰值结果优于原方案；
4. 给出，100 左右/批的产量生产时，每件设备所需的成本（不计算人工成本）。

你们应当在原电路的工程内，新建一块电路板，在原理图上标注所有主要改进点，并且添加设计注释。请你们依照前两块电路板的层级模式进行设计，并将电路板、原理图、PCB 分别命名为 Gen3、第三版原理图、第三版 PCB。

PCB 上的改进及其注释只作为评分的次要参考依据。

## Question 2.2：支持快充协议

由于当前电路只能支持 5V 的充电电压，龚学长希望能够支持快充协议，以支持更高电压、更高电流的充电及数据采集。任何充电头和手机开始充电时，默认都是 5V 充电，之后，手机将会向充电头发送数据包、达成“握手”，随后才会转为快充。本题充电协议为 PD3.0，最大电压 20V、最大电流 5A。也就是说：

1. 首先，大多数快充协议使用 Type-C 作为接口，因此，你们需要**改变接口类型**；
2. 其次，你们需要让手机与充电头能够**顺畅地互相发送数据包**；
3. 随后，你们需要考虑**升高的电压、电流**可能会对电路承载能力造成的考验；
4. 你们必须让所有芯片和元件都处于其**预期工作电压**之下；
5. 你们必须让主控芯片能够检测到当前其是否工作在默认充电模式下，并保证用户界面显示效果；
6. 最后，你们必须考虑升高的最大电压和电流对各种工作条件（也就是不同充电协议）下采样精度的影响。

### Question 2.3: 充电协议嗅探

本题要求将设备模拟为一部手机，通过向充电器发送伪造的数据包以嗅探充电器支持的快充协议类型。具体需要实现的功能如下：

1. 功能 1：设计主控芯片实现数据包发送与握手，并嗅探所有快充协议；解决主控芯片与手机同时发送数据包的干扰问题；
2. 功能 2：使用常见 STM32 型号作为主控，支持快充协议并提供用户界面；设计独立用户交互方式以支持嗅探功能。

#### 提交要求：

1. 在原有电路工程中新增一块电路板，并按照前两块电路板的层级模式，将该电路板命名为 Gen4；
2. 将原理图命名为“第四版原理图”，PCB 命名为“第四版 PCB”；
3. 在原理图中标注所有主要改进点，并附加设计注释；
4. 对于 PCB 的改进及其相关注释，仅作为次要参考依据；
5. 实现功能 1 为必选项，若进一步拓展实现功能 2，将获得额外加分。

#### 评价标准：

- **电路可行性**：在理想工作条件下，电路是否能够按照设计要求实现预期功能；
- **电路可靠性**：在实际运行环境中，电路在规定时间内持续稳定执行预期功能的概率（仅考虑电路设计层面的可靠性问题，不涉及实际生产工艺等外部因素）。

## Part 3 电路焊接

龚学长希望你们根据之前的设计进行**电路焊接**。请根据相关通知**领取物料和工具**，并在**指定时间、指定地点**完成焊接操作。请**根据原理图和 PCB** 完成焊接，注意**先焊接其他元件，再焊接屏幕**。在焊接屏幕之前，请将半成品拿到学长处**拍照留档**。如因操作不当导致元器件损坏等，需要向学长申请补领材料。如果无法完成焊接，可以向现场的学长请求帮助，在对正反面拍照记录完成情况后，他们将会帮助你们完成焊接，但是将酌情扣分。

你们的报告中须包含**带有手写组号和姓名标识的 PCB 正反面照片**，照片中需保证**细节清晰**。如果在焊接过程中总结出经验或观察到特殊现象，也可以在报告中**进行描述**。

## Part 4 嵌入式软件开发

在完成本模块时，请参考附件中的文件，包括 STM8S003 的数据手册、STM8 的官方库和例程、OLED 模块的原理图、主控数据手册、例程、STM8S103（与 STM8S003 只有非常细微的不同）的相关例程和 I2C 协议手册。

### Question 4.1: Hello World

在本题中，你们需要完成调试电路、串口通信、LED 控制、按钮响应以及复位电路等功能，具体实现要求如下：

1. 调试电路：通过 ST-Link 烧录程序，程序正常运行即验证通过。
2. 串口通信：配置波特率 115200，上电发送 Program Start.，每秒发送 Hello, World!。
3. 上位机接收：用 Python 编写程序，实时接收并显示 STM8 发送的数据。
4. LED 控制：上电时红绿 LED 亮；使用按钮 1 控制绿色 LED 状态翻转，使用按钮 2 控制绿色 LED 状态翻转；“按下-释放”计为一次操作。
5. 复位功能：按复位键后 LED 恢复亮起，串口重新发送 Program Start.。

提交要求：

- 命名为 4-1 Project 的完整工程文件夹，包含可编译的源代码；
- 命名为 4-1 Master.py 的 Python 上位机程序，用于接收并显示串口数据；
- 命名为 4-1 Demo.mp4 的测试视频，证明你们完成了本小题。

### Question 4.2: 串口实现

在本题中，你们需要自行设计代码，完成 STM8 与上位机间的实时数据传输与可视化系统，包括以下两个部分：

1. STM8 实时采集与传输：
  - 以 10Hz 的采样频率实时采集 USB 的电压与电流数据；
  - 通过串口以波特率 115200 发送数据，格式为：“电压值电流值”，例如 5.000 1.000；
  - 单位分别为：电压 (V)、电流 (A)，数据格式为整数部分 1 位，小数部分 3 位，数据之间以空格分隔，末尾附加换行符。

## 2. 上位机数据处理与可视化:

- 使用 Python 实现串口数据接收, 并通过 Matplotlib 实时绘制折线图;
- 横轴表示时间 (最近 10 秒), 纵轴显示三条曲线: 电压、电流及功率;
- 纵轴范围自动调整, 支持低通滤波功能, 并绘制滤波后的功率曲线。

### 提交要求:

- 名为 4-2 Project 的完整 STM8 工程文件夹;
- 名为 4-2 Master.py 的 Python 上位机程序;
- 名为 4-2 Demo.png 的截图, 包含终端输出的数据和绘制的折线图。

## Question 4.3: UI 设计

在本题中, 你们需要设计一个多页面 OLED 用户界面, 实现数据可视化和实时显示功能。采样频率不作硬性要求, 但应保证显示流畅。具体要求如下:

1. 静态文字欢迎页面: 设备上电时显示静态欢迎界面, 包含设备名称和基本信息, 页面布局应清晰美观, 充分利用屏幕空间。
2. 页面导航功能: 通过按钮 1 切换至上一页面, 按钮 2 切换至下一页面。
3. 实时数据显示页面: 显示电压、电流、功率三项数据, 格式为 1 位整数和 3 位小数, 单位分别为 V、A、W。
4. 历史功率页面: 展示功率随时间变化的折线图, 横轴为时间 (跨度约 10 秒), 纵轴为功率 (单位 W), 并附加描述文字和纵轴坐标。

### 提交要求:

- 名为 4-3 Project 的完整工程文件夹, 包含全部源代码和配置文件;
- 名为 4-3 Demo.mp4 的演示视频。

## Bonus: 初探 NFC

NFC 技术已在日常生活中得到广泛应用, 校园卡就是一种 NFC 的典型应用。龚皋学长对射频电路和 NFC 技术充满兴趣, 他希望将深入了解 NFC 的工作原理, 探索其经典的应用场景。

在开始前, 你们需要明确以下内容:



### 1. NFC 终端设计要求：

- 设计一个 NFC 终端，通信范围为 1 至 4 厘米。
- 终端需与符合 Mifare Classic 1K 标准的 NFC 卡通信，通信需遵循 ISO14443-A 协议。

2. **Mifare Classic 1K 标准简介：** Mifare Classic 1K 卡（简称 M1 卡）是最常见的 NFC 卡，符合 NXP 的 S50 标准，校园卡也基于这一标准。

### 3. 硬件与通信限制：

- 不得使用除提供的 NFC 硬件以外的 NFC 设备。
- STM8 无法承受攻击算法所需的计算资源。
- 对于 5.2，与 PN5180 射频前端模块的通信可由上位机完成；对于其他小题，通信只能由 STM8 完成，不能由上位机直接发送。

4. **任务目标：** 组合使用 STM8S105K4T6 与 PN5180 射频前端模块完成所有题目要求。

## Question 5.0：背景知识

在开始前，你们需要先了解一些基础知识：

1. **ISO14443-A 协议：** Mifare Classic 1K 卡与 NFC 读卡器之间的通信遵循 ISO14443-A 标准，载波工作频率为 13.56MHz。该协议定义了卡片与读卡器之间的通信方式，是 NFC 技术的基础。
2. **Mifare Classic 1K 标准：** 校园卡及市场上流行的许多 IC 卡均采用 Mifare Classic 1K 标准，内部代号为 S50。该标准由 NXP（恩智浦）制定，广泛应用于门禁、公交卡等场景。
3. **卡片存储结构：** M1 卡的存储结构分为扇区和区块，其中 0 扇区 0 区块（从 0 开始编号）储存卡号及其他元数据。通常情况下，该区块被锁定，无法写入。但国内存在可擦写的“魔术卡”，如 UID 卡和 CUID 卡。
4. **ISO14443 协议握手：** M1 卡与读卡器通过 ISO14443 协议完成握手，进入 Active 状态后，进行基于 Crypto1 算法的认证以及数据读写操作。
5. **加密算法 Crypto1：** M1 卡使用 Crypto1 作为加密算法。Crypto1 是 NXP 的私有加密算法，尽管其具体实现未公开，但通过逆向工程已揭示了部分细节。
6. **半加密卡与全加密卡：** 根据加密情况，M1 卡可分为半加密卡和全加密卡。半加密卡指部分扇区使用默认或已知密钥的卡，破解难度较低；全加密卡则所有扇区均加密，破解难度较高。
7. **常用攻击算法：** 针对 Crypto1 的常见攻击算法包括 Nested 攻击和 Hard-Nested 攻击等，这些算法利用 Crypto1 的已知漏洞实现破解。

8. **PN5180 NFC 读卡器**：PN5180 是一款常用的 NFC 读卡器主控芯片，内置射频前端和 Cortex-M0 内核。它与天线和外围电路配合，用于发送和接收射频信号。

9. **通信协议**：PN5180 通过 SPI 接口与上位机（如计算机或单片机）通信，通信内容需遵循特定的帧格式以确保数据传输的正确性。

更详细的内容推荐阅读附件中的 ISO14443 协议文件、SPI 协议手册、PN5180 芯片数据手册、MF1S50YYX\_V1 手册（EV1 版本与经典 M1 卡差异极小）及 Crypto1 攻击与逆向工程论文等文件。Bonus 所使用的 STM8S105K4T6 可参考 STM8S105 数据手册。也推荐阅读《中国金融集成电路（IC）卡规范第 8 部分：与应用无关的非接触式规范》，其内容等同于 ISO14443-3 中文版。

### Question 5.1：读取、写入 CUID 卡

本题要求编写一个适用于 STM8S105K4T6 最小系统板的 IAR 工程，通过 SPI 与 PN5180 模块通信，使用给定的密钥与数据 CUID 卡进行交互，实现以下功能：

1. 功能 1：使用给定的密钥读取 1 扇区 0、1、2 区块的数据。具体要求如下：

- 0 区块储存一个长度为 16 字节的二进制唯一标识码（UID1）。
- 为确保数据正确性，1 区块和 2 区块应当作为备份存储 UID1 的数据：
  - 1 区块为 UID1 的按位取反备份。
  - 2 区块为 UID1 的不取反备份。

2. 功能 2：使用给定的密钥在 2 扇区 0、1、2 区块写入数据。具体要求如下：

- 每个区块写入两位同学的中文姓名（最多 4 个字的姓名）。如果某位同学姓名长度超过 4 个字，可以使用符合要求的 4 个字符代号，并在报告中注明。
- 每个区块前 8 个字节储存第一位同学的姓名，后 8 个字节储存第二位同学的姓名。姓名应左对齐，多余的字节用 0 填充。

#### 提交要求：

1. 自主设计工程，提交名为 5-1 Project 的工程文件夹。

2. 在解题报告相应部分插入以下内容：

- 使用 16 进制字符串表示的 UID1，以证明你们已经完成了功能 1。
- 使用 16 进制字符串表示的最终写入 2 扇区 0、1、2 数据，以便进行核验功能 2。

**注意事项:**

1. 每组的 CUID 卡标有组号，内容独立生成，不可互用。卡片 1、2 扇区默认设置：A、B 密钥为 FFFFFFFF，控制位为 FF078069，数据块内容为 0（均为 16 进制表示）。
2. 完成后，请将标有组号的 CUID 卡提交实验室管理同学。未完成者可在 3 月 22 日（周六）12:00 前统一提交，具体时间另行通知。

**Question 5.2: 破解 CUID 卡的加密算法**

Crypto1 是 NXP 为 Mifare Classic 1K 卡设计的私有加密算法，尽管未公开，但通过逆向工程已被还原并发现多个漏洞。常见攻击方法包括 Nested、Hard-Nested、Dark Side、Static Nested 和 Back Door 攻击。由于市场占有率高、成本低廉，Crypto1 的生命周期被延长。

与公开算法如 RSA 相比，Crypto1 未经过验证，安全性依赖于算法保密性，而非密钥保密性，违背了 Kerckhoffs 原则。RSA 等公开算法因广泛验证而更可靠。

不同型号 NFC 卡的破解难度不同。NXP 原厂 Mifare Classic 1K 卡对 Nested、Hard-Nested 等攻击效果较好。国产替代卡如复旦微电子 FM11RF08 系列中，Hard 卡易破解，而无漏洞卡需 Static Nested 或 Back Door 攻击。FM11RF08S（第三代无漏洞卡）目前仅能通过 Back Door 攻击破解。

卡片加密情况影响破解难度。全加密卡需更复杂攻击，而半加密卡可通过 Nested 或 Hard-Nested 攻击破解。提供的 CUID 卡为半加密卡，便于实验。大部分校园卡可通过 Nested 或 Hard-Nested 攻击破解，具体型号需实验验证。

请参阅附件中的以下文献，以便深入了解相关攻击方法：

1. *Wirelessly Pickpocketing a Mifare Classic Card* (Nested 攻击)
2. *Ciphertext-only Cryptanalysis on Hardened Mifare Classic Cards* (Hard-Nested 攻击)
3. *Reverse-Engineering a Cryptographic RFID Tag* (Crypto1 逆向工程)
4. *Dismantling MIFARE Classic* (Crypto1 逆向工程)
5. *The Dark Side of Security By Obscurity* (Dark Side 攻击)
6. *MIFARE Classic: exposing the static encrypted nonce variant* (Back Door 攻击)

本题要求编写一个可在 Windows 11 24H2 上运行的 C++、Python 或 Node.JS 工程（推荐使用 Python）实现以下功能：

1. 功能 1：通过 CH341A USB 转 SPI 模块与 PN5180 模块进行通信。
2. 功能 2：使用 Nested 攻击破解 CUID 卡 3 扇区内的数据。扇区加密和数据格式服从如下规则：



- 3 扇区的两个密钥根据组别独立随机生成，不同组别之间不可通用。除 3 扇区外，其余扇区均未加密。
  - 数据内容为一个长度为 16 字节（即一个区块）的二进制唯一标识码，记作 UID2，存储在 0 区块内。
  - 为确保数据正确性，1 区块和 2 区块中的数据是 0 区块的备份，其中：
    - 1 区块存储 UID2 的按位取反备份。
    - 2 区块存储 UID2 的非取反备份。
3. 可选功能：尝试实现并使用 Nested、Hard-Nested、Back Door 或 Dark Side 攻击破解任意校园卡的所有扇区数据。

#### 提交要求：

1. 提交一个名为 5-2 Project 的工程文件夹，包含完整的代码及相关文件。
2. 在报告中详细描述破解加密算法的步骤，确保我们能够复现你们的工作。
3. 对于 C++ 工程：
  - 确保包含所有使用的第三方库及编译生成的二进制可执行文件。
  - 在报告中明确列出所使用的编译器、编译器版本、编译命令及选项。
4. 对于 Python 或 Node.JS 工程：
  - 在报告中列出所有使用的第三方库及其精确版本。
  - 指明主脚本文件及运行命令（包括任何运行选项）。
5. 完成功能 2 后：
  - 在报告中提供 UID2 的 16 进制字符串表示，以证明成功破解 CUID 卡的加密。
6. 完成可选功能后：
  - 在报告中提供校园卡 0 扇区内所有区块的数据，每个区块一行，使用 16 进制字符串表示。
  - 注意：校园卡数据属于隐私，仅需收集 0 扇区数据以验证破解完成。根据实际测试，校园卡 0 扇区的某个区块可能包含持卡人的姓名，可用此验证本功能的完成情况。
7. 建议在完成破解任务后，将校园卡的所有扇区数据保存为 .dump 文件（不需要提交），以备后续验证或参考。

**注意事项：**

- 浙大校园卡的余额是通过联网存储的，因此复制或模拟校园卡不会增加账户余额。所有消费记录仍会记入原始校园卡的账户中。
- 请提供校园卡的同学务必谨慎，避免滥用或盗刷。如需使提取的校园卡数据失效，或发现校园卡被滥用，请立即挂失并申请新卡。

**Question 5.3：卡模拟**

卡模拟是指通过 NFC 终端模仿 NFC 标签的行为，欺骗其他 NFC 读卡器，使其误认为是一个合法的 NFC 卡。许多手机和智能手环都具备类似功能。卡模拟功能可通过 PN5180 模块实现，具体内容请见 PN5180 官方文档中的 **11.8.8 Autocoll (Card Emulation)** 章节。

本题要求编写适用于 STM8S105K4T6 最小系统板的 IAR 工程，通过 SPI 与 PN5180 模块通信，实现卡模拟的任务。要求如下：

1. 使用 PN5180 模块模拟 Mifare Classic 1K 卡，并将 5.2 中破解的校园卡全扇区数据作为响应数据。配置模拟卡的型号为 Type-A Mifare Classic 1K 卡，确保 SAK (Select Acknowledge) 值及通信特性与原卡一致。
2. 使用移动电源（如充电宝）为 STM8 和 PN5180 模块供电。
3. 利用模拟的校园卡尝试刷开任意浙大门禁，或进行消费操作。
4. 本题允许使用官方库及各种第三方库。

**提交要求：**

1. 请提交名为 **5-3 Project** 的工程文件夹，其中包括完整的源代码及相关项目文件。
2. 在题目报告的附件中，你们需要提交一个测试视频（命名为 **5-3 Demo.mp4**），展示使用 PN5180 模块刷开任意浙大门禁或进行消费操作。

**注意事项：**

- 浙大校园卡的余额是通过联网存储的，因此复制或模拟校园卡不会增加账户余额。所有消费记录仍会记入原始校园卡的账户中。
- 请提供校园卡的同学务必谨慎，避免滥用或盗刷。如需使提取的校园卡数据失效，或发现校园卡被滥用，请立即挂失并申请新卡。

## 温馨提示：视频提交说明

1. 视频需要保持拍摄清晰、不允许进行裁剪。
2. 需要持续出现之前在 PCB 版上标记的组号，以保证完成的独立性。
3. 视频的码率应该小于 5Mbps。
4. 请在视频中展示你们的设计总结，包括但不限于：设计的优点、不足、改进方向等。

