电子科技大学
University of Electronic Science and Technology of China

2026 年“英才杯”电子设计竞赛试题

参赛注意事项

（1）参赛队员须认真填写参赛报名表，报名信息必须准确无误。
（2）本题每支参赛队伍人数不多于一人。开赛后不得中途更换队员、参赛赛道和作品题目。
（3）参赛队员可以借助互联网、开源软件及人工智能工具辅助设计与开发，但不得与其他参赛队伍进行方案讨论、代码交换或成果共享。

NAS 智能应用创新

随着容器技术和人工智能技术的发展，NAS 正逐步由传统的数据存储设备转变为集数据存储、应用运行、智能处理和个人服务于一体的综合平台。

如果NAS不再只是一个“存放文件的盒子”，它还能为学习、科研、工程实践和日常生活解决哪些实际问题？本赛题鼓励参赛者从真实需求出发，在 NAS 上设计并开发具有创新性、实用性和长期使用价值的软件应用，让 NAS 成为真正能够持续提供服务的个人智能平台。



一. 题目说明

1. 命题方式

本题采用半开放式命题形式，不设置统一的固定题目。

参赛队伍应结合自身兴趣、技术能力和实际需求，自主确定具体作品题目，并从规定的三个选题方向中任选一个方向完成作品。

不同方向之间不进行简单的功能数量比较。参赛队伍应重点说明作品所解决的问题、目标用户、应用场景、技术方案以及实际使用价值。

2. 选题方向说明及要求

本次比赛设置科协日常管理、嵌入式开发指南和自主创新三个选题方向。参赛队伍可结合实际需求选择其中一个方向，自主确定作品题目，并采用 MCP 工具服务、Agent Skill、Web 应用或其他合理的软件形式实现。所有作品均须以 NAS 为运行平台，将所需运行组件通过 Docker Compose 统一部署，并满足数据持久化、可维护及主办方赛后长期使用的要求。

（1）方向一：科协日常管理

参赛者应围绕科协日常运营、设备使用、场地管理及成员协作中的实际需求，开发能够提升管理效率和使用体验的 NAS 应用。

参考内容包括：3D 打印机预约、打印任务登记、耗材统计及设备维护记录；门禁使用申请、人员进出登记及授权记录管理；仪器设备借还、元器件库存管理；成员信息、活动报名、值班安排及项目资料归档等。

作品可选择其中一个或多个场景进行设计，应明确使用对象和管理流程，提供相应的信息登记、查询、状态跟踪及记录保存功能，并合理设置不同用户的访问和操作权限。涉及 3D 打印机、门禁等设备的作品，可通过管理记录、文件导入或模拟数据完成核心功能，不将专用硬件接入作为运行前提。

本方向重点关注作品与科协实际需求的结合程度、管理流程的合理性、使用便利性、数据准确性及持续运行能力。

（2）方向二：嵌入式开发指南

参赛者应围绕嵌入式技术入门、项目实践和开发调试中的常见问题，开发运

行在 NAS 上的学习指导与开发辅助应用，帮助使用者查找资料、理解开发步骤、开展实践并排查问题。

参考内容包括：开发板与开发环境入门指南、外设接口和通信协议学习、示例代码与项目教程管理、芯片及元器件数据手册检索、编译错误和串口日志分析、常见故障排查、开发经验整理等。

作品应围绕明确的学习或开发任务，提供结构清晰、可检索、可操作的指导内容。可结合 MCP 工具服务或 Agent Skill，实现资料查询、示例检索、日志分析和分步骤指导等功能。涉及技术参数、接线说明和故障判断时，应标明资料来源或验证依据；涉及硬件实践时，应提供示例文件、日志或模拟数据，便于主办方独立测试和使用。

本方向重点关注内容的准确性、对初学者的友好程度、指导过程的完整性，以及解决实际开发问题的能力。

（3）方向三：自主创新设计

参赛者可从学习、科研、工程、校园或日常生活中的其他真实需求出发，自主设计并开发具有创新性和实用价值的 NAS 应用。本方向不限定具体应用领域和技术路线。

作品应明确目标用户、拟解决的问题及核心功能，说明 NAS 在数据存储、持续运行、局域网访问或本地处理等方面发挥的作用，并形成从数据或任务输入到结果输出的完整使用流程。

本方向重点关注需求价值、核心功能的原创性、技术实现、使用体验，以及作品在 NAS 上的长期使用和维护价值。

二. 设计任务

围绕 NAS 应用场景，自主确定作品题目，设计并实现一个能够部署、运行在 NAS 上的软件应用。

参赛者应通过作品的设计与实现，掌握 Docker 及 Docker Compose 的基本使用方法，了解 NAS 应用的部署、存储和运行方式，学习 MCP 工具服务或 Agent Skill 的基本开发方法，并提升需求分析、应用设计、软件开发、系统测试和工程实现能力。

作品应关注数据安全、隐私保护、系统可维护性和持续使用价值，能够应用于学习、科研、工程实践、校园服务或日常生活等真实场景。

三. 设计要求

1. 基础要求

（1）**自主选题。** 参赛队伍应结合实际需求自主确定作品题目，明确作品拟解决的问题、目标用户、应用场景和主要功能。

（2）**独立运行。** 作品应能够在 NAS 上独立运行，不得依赖参赛者个人电脑长期提供计算、存储或网络服务。

（3）**容器化部署。**作品的运行组件原则上均须完成 Docker 化，并通过 Docker Compose 实现一键部署。作品所需的服务、端口、目录、环境变量和依赖项应在配置文件或说明文档中明确标注。

（4）**数据持久化。** 作品应具备必要的数据持久化能力。容器重启、更新或重新部署后，重要数据不应无故丢失。

（5）**运行状态与日志。** 作品应提供基本的运行状态信息和日志记录，便于使用者了解系统状态并定位运行过程中出现的问题。

（6）**安装及使用说明。**作品应提供清晰、完整的安装部署说明和使用说明，使评测人员能够根据文档完成部署和基本操作。

（7）**测试与验证。** 作品应提供必要的示例数据、测试方法或演示流程，用于验证主要功能是否能够正常运行。

（8）**可维护与可迁移。** 作品应具备基本的可维护、可迁移和可重复部署能力，不得将关键运行环境、绝对路径或个人设备信息写死在程序中。

（9）**第三方组件说明。** 作品中使用的第三方软件、开源项目、模型、数据集和其他外部资源，应明确说明其来源、用途及开源许可证。

（10）**安全与隐私。** 作品涉及文件访问、用户账号、个人信息或其他敏感数据时，应设置必要的权限控制和安全措施，不得擅自收集、上传或泄露用户数据。

2. 发挥部分

在完成基础要求的基础上，参赛队伍可根据作品特点，选择以下一个或多个

方向进一步完善作品：

（1）针对文件、账号和敏感数据设计更加完善的身份认证、权限管理、数据备份和隐私保护机制；

（2）将作品投入真实场景进行持续使用和测试，并根据实际使用反馈改进功能和交互方式；

（3）提供便于非专业用户使用的图形化管理界面、初始化配置流程、数据管理功能或自动升级机制；

（4）采用模块化或插件化设计并提供标准接口，使用户能够根据自身需求添加数据源、工具或功能模块；

（5）形成具有长期维护价值的开源项目，提供完善的开发文档、版本记录和社区使用说明；

（6）其他能够显著提升作品创新性、实用性、稳定性或使用体验的功能。

四. 挑战内容

挑战内容面向已较好完成基础要求和发挥部分的参赛队伍。参赛队伍可结合自身作品选择以下任意一项或多项进行挑战。

挑战内容应形成完整、可运行的系统能力。仅增加容器、接口或工具数量，但各模块之间缺少实际协同的，不作为挑战指标完成情况认定。

1. 多容器协同系统

设计并实现一个由多个功能独立的容器共同组成的 NAS 应用系统。

系统应至少包含应用服务、数据服务以及任务处理、消息通信、智能计算等类型中的一种扩展服务。反向代理、管理面板等通用辅助组件不单独作为核心业务容器计算。

各容器之间应通过明确的接口、内部网络、消息队列或任务调度机制进行协同，并实现以下主要能力：

① 服务发现和容器间通信；

② 数据持久化和共享数据管理；

③ 容器启动顺序和依赖关系管理；

④ 服务健康检查和异常状态识别；

⑤ 单个服务异常时的故障隔离或自动恢复；

⑥ 统一的配置、日志或运行状态管理。

现场评测时，参赛队伍应完整展示一次由多个容器共同完成的业务流程。评测人员可随机停止其中一个核心服务，检查系统的异常处理和恢复能力。

2. 可扩展的 NAS 工具或应用平台

设计并实现一个能够由其他开发者继续扩展的 NAS 工具或应用平台。

平台应制定明确的插件、工具、数据源或服务接入规范，使新增功能可以通过独立模块完成，而不需要修改平台的主要程序代码。

平台应具备以下主要能力：

① 插件或扩展模块的发现、加载和管理；

② 统一的输入、输出或接口定义；

③ 扩展模块的权限和资源限制；

④ 扩展模块运行状态和错误信息管理；

⑤ 面向开发者的接入文档和示例；

⑥ 至少完成两个功能不同的扩展模块。

现场评测时，参赛队伍应展示扩展模块的安装、启用、调用和停用过程，并说明其他开发者如何按照规范为平台增加新的功能。

3. 跨设备数据采集与智能处理系统

设计并实现一个由 NAS 与嵌入式设备、传感器、移动终端或其他网络设备协同工作的完整系统。

外部设备负责数据采集或任务触发，NAS 负责数据接收、存储、处理、分析和结果展示。

系统应具备以下主要能力：

① 外部设备的识别、接入和状态管理；

② 数据的持续采集和可靠传输；

③ NAS 端的数据存储、处理和查询；

④ 设备离线、网络中断或数据异常时的处理；

⑤ 数据分析结果的展示、通知或反馈控制；

⑥ 完整的数据流向和权限说明。

现场评测时，参赛队伍应展示真实设备与 NAS 之间的数据传输和处理过程。仅使用预先制作的静态数据模拟设备运行的，不作为本项挑战指标完成情况认定。

对于较好地完成以上任意要求（包括小要求）且满足以下招新要求的小组，可以酌情考虑直接纳入英才科协

1. 自觉拥护中国共产党的领导，在生活作风，思想道德层面无污点，在比赛过程中无作弊，高年级学生代打等不端行为，一经发现，取消一切成绩，鼓励相互监督;

2. 能够出色地完成全部（或者关键的）的基础指标与发展指标;

3. 能够在现场（不借助浏览器与 AI）准确且流利地回答测评人员提出的各项技术问题;

4. 组内全部人员均为大一新生;

5. 作品在性能，创新点，完成度等方面具有卓越的表现;

五. 说明（务必认真阅读）

1. 本赛题以大一新生为主要参赛对象，鼓励不同专业学生跨专业组队，通过合理分工完成作品的需求分析、设计、开发、测试和展示。

2. 本题为开放式命题。参赛队伍应根据自身能力选择合理的作品范围，优先保证核心功能完整、运行稳定和实际可用，不以简单堆叠功能数量作为主要目标。

3. 作品的运行组件原则上均须完成 Docker 化，并通过 Docker Compose 实现一键部署。如因特殊技术原因无法使用容器部署，应在设计报告中说明原因及替代部署方案。

4. 作品不得依赖参赛者个人电脑或个人账号长期提供服务。比赛结束后，主办方应能够根据参赛队伍提交的材料，在 NAS 上独立部署、运行和维护优秀作品。

5. 参赛队伍应提前准备完整的现场演示流程，并提供必要的测试账号、示例数据和操作说明。涉及外部网络服务的功能，应同时准备网络异常情况下的演示或替代验证方式。

6. 涉及文件、账号、个人信息或其他敏感数据的作品，应设置合理的身份

认证、访问权限和数据保护机制。不得擅自收集、上传或泄露用户数据。

7．作品应保留必要的运行状态、日志和测试接口。因缺少配置说明、测试条件或访问权限，导致评测人员无法部署或验证的功能，评测时将无法作为有效成果认定。

8．参赛队伍须提交作品源代码、Dockerfile、Docker Compose 配置文件、部署说明、使用说明以及必要的示例数据或测试方法。

9．设计报告应包括作品背景与需求分析、总体方案、系统架构、主要功能、关键技术、数据存储方式、部署方法、安全与权限设计、测试方案及测试结果等内容。

10．作品中由参赛队伍自行编写的程序代码应随设计报告一并提交。自动生成的代码可不在报告正文中完整列出，但应在源代码材料中提供。

11．使用开源项目、第三方代码、人工智能模型、数据集或外部服务时，必须如实注明来源、许可证、具体用途和使用范围，并明确说明由参赛队伍自主完成的部分。

12．抄袭、冒用他人成果或故意隐瞒第三方项目使用情况的，将取消比赛成绩。

13．参赛队伍应遵守有关网络安全、数据安全、个人信息保护和知识产权的规定，不得使用来源不明、未经授权或包含违法违规内容的数据。

14．挑战内容为可选内容，不要求参赛队伍全部完成，也不以完成数量作为唯一评价依据。评审时将综合考虑挑战内容的实际难度、完成质量和作品整体效果。

15．挑战内容必须提供相应的验证依据。性能类挑战应说明测试环境、数据规模和测试方法；运行类挑战应提供现场演示或运行记录。

16．仅在设计报告中提出设想，或者仅通过界面、图片、视频模拟相关效果，但未完成实际功能的，不作为挑战指标完成情况认定。

17．作品的原创程度、技术设计合理性、工程实现质量以及标准化、可维护性等内容，属于整体评分和答辩考察范围，不单独设置为挑战项目。

六．评分标准

本题采用开放式命题形式，不设置与具体功能点一一对应的详细评分表。评委将结合参赛作品的现场演示、设计报告、源代码、部署测试和答辩情况进行综合评价。

评审主要考虑以下方面：

（1）**需求价值与创新性：** 作品是否面向真实问题，需求分析是否合理，解决方案是否具有创新性和实际使用价值。

（2）**功能完整性与实际效果：** 核心功能是否完整，运行是否稳定，作品能否达到预期目标并完成实际任务。

（3）**技术设计与实现质量：** 系统架构、数据处理、接口设计、异常处理及代码实现是否合理，是否体现相应的技术深度和自主开发工作。

（4）**部署与工程化能力：** Docker 及 Docker Compose 配置是否规范，作品是否易于部署、迁移、维护和重复运行。

（5）**安全性与规范性：** 权限控制、数据安全、隐私保护、第三方组件使用和开源许可证说明是否完整、规范。

（6）**使用体验与长期价值：** 作品是否易于理解和使用，是否具有持续运行、后续维护和实际推广价值。

（7）**设计报告与现场答辩：** 设计报告是否完整、清晰、规范，参赛队员能否准确说明作品方案、关键技术、测试结果及各自承担的工作。

发挥部分和挑战内容均作为综合评价的参考。评审将重点关注相关功能的实际完成情况、技术难度和验证结果，不以功能数量作为唯一评价依据。