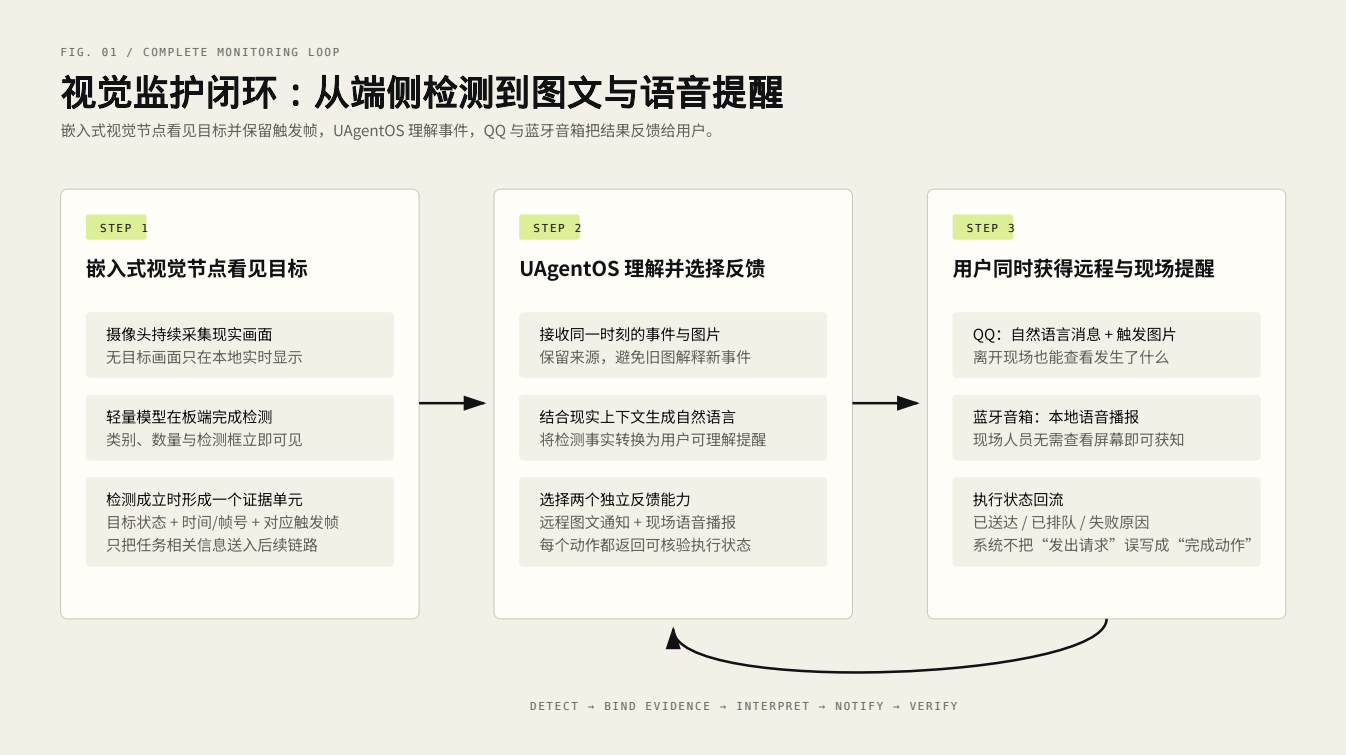
UAgentOS：基于 OpenVela 与 Gemini-S1 的真实设备家居 Agent 技术报告

队伍名称：UAgentOS
参赛方向：基于 OpenVela 智能体的智能家居控制系统

1. 系统总体成果

本项目在 Gemini-S1 开发板上部署 OpenVela Agent，并把真实设备能力注册为 Agent 可调用的工具。系统能够读取板载温湿度、光照和接近数据，查询开发板运行状态，接收外部视觉开发板的目标检测事件，并通过 QQ 与蓝牙音箱向用户反馈。

演示按能力复杂度分为三级：首先进行普通聊天，验证 LLM 的基础语言理解与生成；随后询问当前温度，由 Gemini-S1 板载传感器读取真实数据并以文字或语音反馈；最后由嵌入式视觉节点检测有人并保留触发图片，OpenVela Agent 结合现实上下文生成提醒，通过 QQ 发送图文，同时请求蓝牙音箱现场播报。



这不是只在桌面模拟器中运行的工具调用示例，而是一套部署在真实嵌入式开发板、读取真实传感器并连接真实反馈设备的 OpenVela Agent 原型。

2. 初赛要求与完成内容

初赛要求	评分关注点	本项目完成内容	可核验材料
自定义工具开发	至少 2 个智能家居工具可被 Agent 正确调用（25%）	实现 6 个结构化工具，覆盖环境读取、设备状态、通知、播报、跨节点事件和状态记录	工具注册代码、工具 schema、板端运行记录
自然语言控制闭环	意图理解、工具调用、结果反馈（15%）	完成普通聊天、温度读取/播报、综合视觉监控三类演示	对话日志、传感器读数、QQ 图文、音频状态
可运行与异常处理	文档可复现、错误可解释（10%）	Gemini-S1 实机部署；参数校验；传感器双路径读取；区分送达、排队与失败	运行说明、烟雾测试、错误结果
工程化与开源合规	一键构建、复现信息、许可证（30%）	提供构建/部署脚本、README、依赖说明和凭据隔离	代码仓库、脚本、LICENSE/NOTICE
文档与演示	README、架构与运行演示（20%）	提供技术报告、网页 PPT、演示脚本和运行日志	本提交包

3. 系统结构：Agent 如何使用真实设备

3.1 Agent 主体

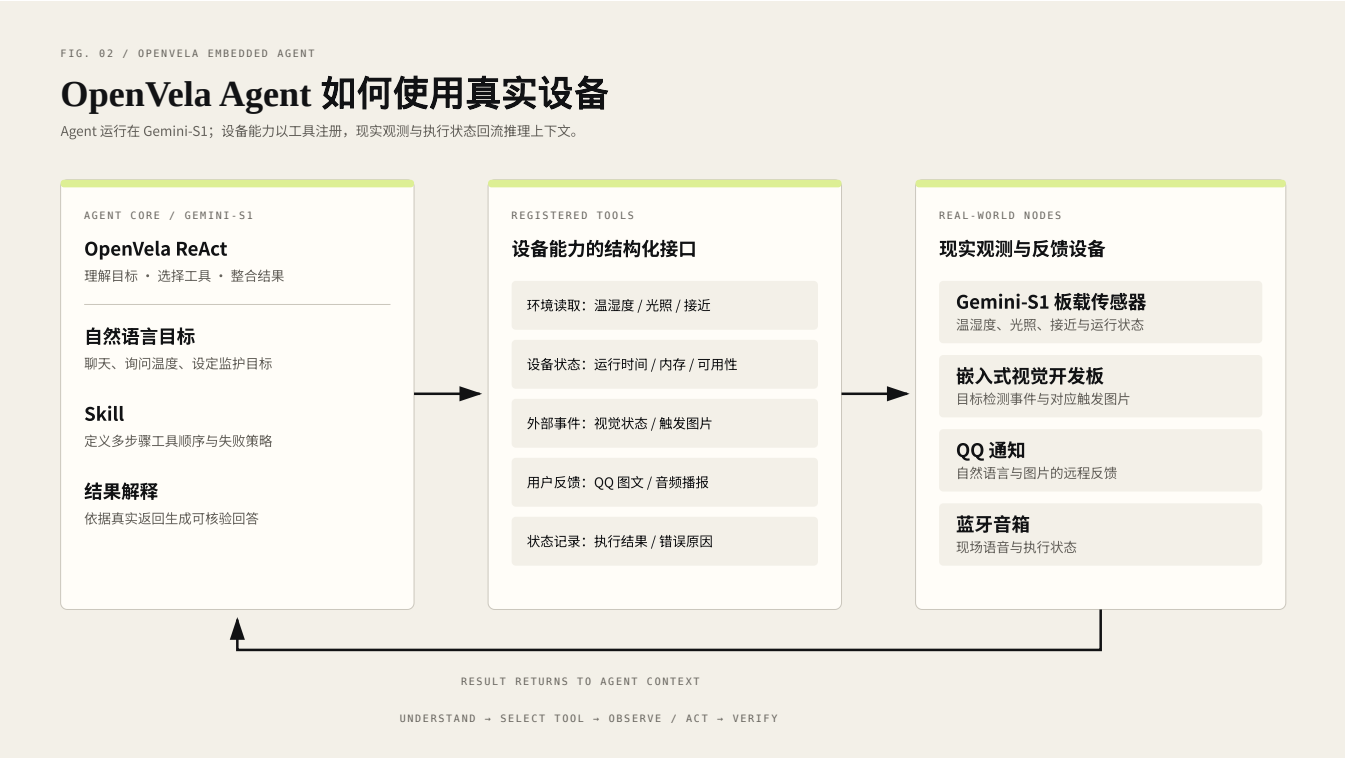
Gemini-S1 运行 OpenVela `ai_agent`。ReAct 负责理解用户目标、选择工具、读取工具结果，并根据现实状态继续推理。Agent 核心运行在嵌入式开发板上，不依赖 PC 端 Python 才能完成基础闭环。

3.2 设备与设备能力

一台真实设备可以提供多项能力。例如 Gemini-S1 同时提供环境读取、设备状态和消息请求；外部视觉开发板提供目标检测与触发图片；QQ 和蓝牙音箱提供面向用户的反馈。项目把每项能力包装成名称、参数、返回值和错误状态明确的工具，Agent 因此不需要记忆具体硬件命令。

3.3 观测与执行结果

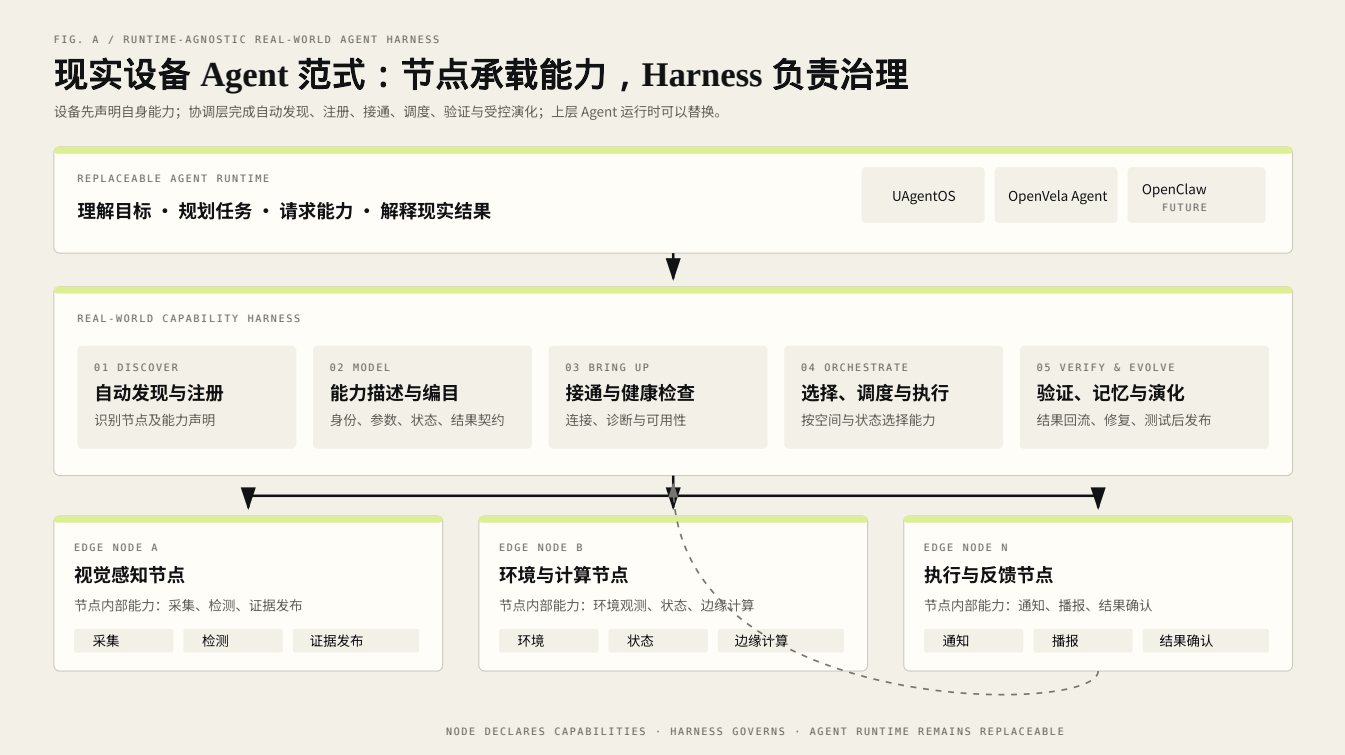
传感器读数、目标检测状态和触发图片属于现实观测；QQ 是否送达、音频是否播放或排队属于执行结果。两类数据都保留来源与状态，Agent 可以据此解释事实，而不是根据一次函数调用猜测物理世界已经改变。



4. 从比赛原型到现实设备 Agent 范式

本项目的工程实现对应一个更一般的研究命题：如何让任意 Agent 友好地调度现实世界，而不被某一块开发板或某一种运行时绑定。

- **端侧设备被表示为节点。** 节点保存身份、连接、位置、当前状态和一组内部能力。
- **硬件功能被表示为节点能力。** 温度读取、视觉检测、通知和播报分别属于提供它们的节点，并以统一参数与结果契约对外声明。
- **节点能力自动发现与注册。** 节点接入时发布能力描述；Harness 负责注册、编目、健康检查和可用性维护，使 Agent 根据语义选择能力。
- **执行结果重新进入推理。** Agent 不把“工具已调用”当作“现实动作已完成”，而是根据观测、送达、排队或失败状态继续判断。
- **Agent 运行时可替换。** 本项目以 OpenVela Agent 作为比赛实现；同一设备语义也可装配到自研 UAgentOS。面向 OpenClaw 等运行时的适配是后续研究方向。



因此，UAgentOS 不只指某个单一 Agent 程序，也代表比赛中实现的一套现实设备协调范式：节点声明能力，Harness 治理能力生命周期，Agent 负责理解目标和解释结果。

5. 自定义工具与现实职责

工具	现实职责	返回结果	在完整故事中的作用
`gemini_environment_read`	读取温度、湿度、光照、接近	每项数值、来源与 ok 状态	为环境问答和视觉事件补充现实上下文
`gemini_device_status`	查询板端运行时间、内存与平台状态	uptime、meminfo、平台信息	在执行前检查 Agent 所在设备是否可用
`device.time.sync`	校准 Gemini-S1 系统时间	校准结果、板端时间与 NTP 状态	让事件、日志和设备状态共享可信时间线
`gemini_qq_notify`	向用户发送自然语言通知与图片	配置、鉴权、请求和提供方结果	把视觉提醒远程发送给用户
`gemini_audio_speak`	请求语音播报	`delivered`、`queued` 或失败原因	在现场播放环境或安全提醒
`gemini_capability_emit`	发布跨设备能力事件	目标、能力和路由状态	把外部视觉节点及未来设备接入 Agent
`gemini_home_state_set`	记录设备或场景状态变化	板端 JSONL 证据	保存可回溯的状态与动作记录

这些工具共同进入 OpenVela Agent 的能力面，不是彼此孤立的演示，而是共同支撑“感知现实、理解目标、执行反馈、核验结果”的完整系统。

6. 三类演示场景

6.1 场景一：普通聊天

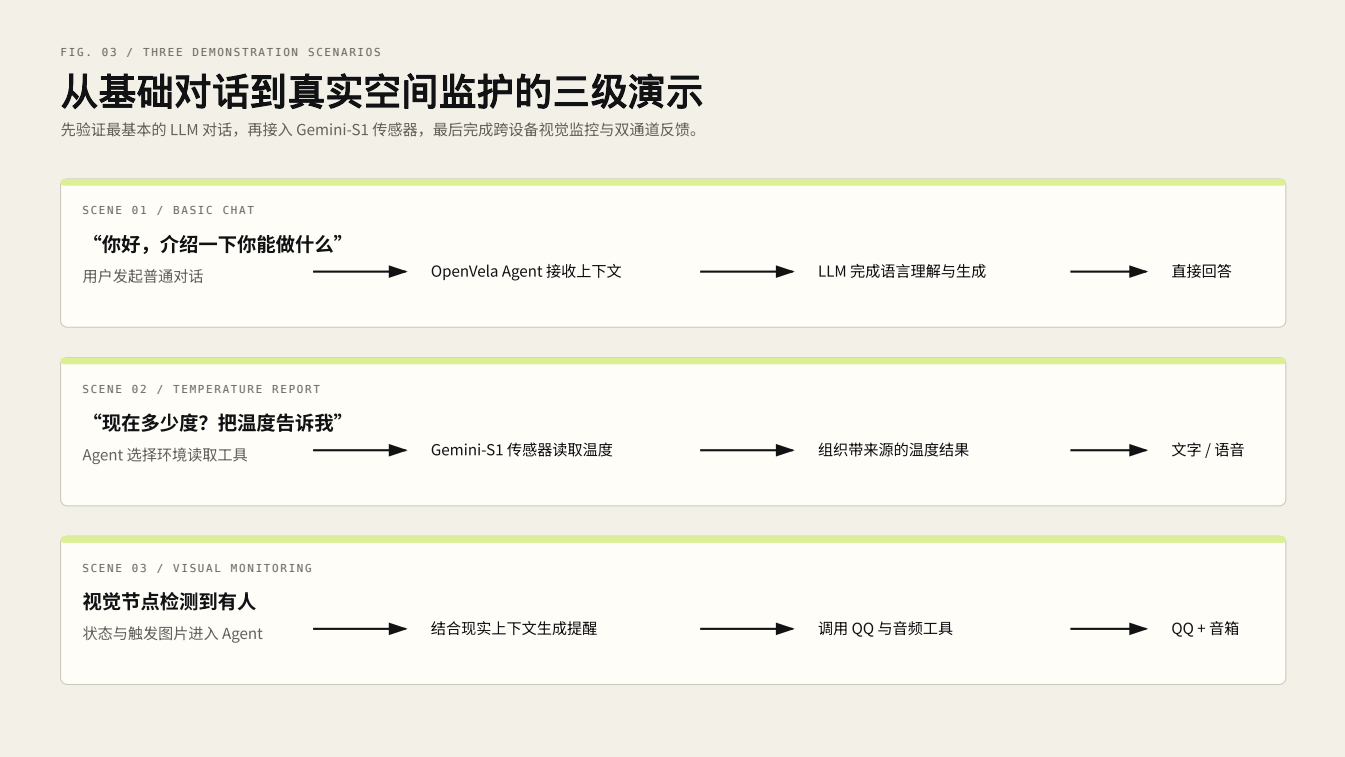
用户进行普通问候或询问“你能做什么”。OpenVela Agent 将对话上下文交给 LLM，完成最基本的语言理解与自然语言生成，不强行调用硬件工具。该场景验证 Agent 的基础交互入口正常。

6.2 场景二：温度读取与播报

用户询问“现在多少度”或要求“把温度告诉我”。Agent 选择 `gemini_environment_read`，由 Gemini-S1 板载传感器读取真实温度，随后生成带来源的文字回答；需要现场播报时继续调用 `gemini_audio_speak`。音频工具明确区分已经播放、已经排队和路由失败。

6.3 场景三：综合视觉监控

独立嵌入式视觉开发板检测到目标后提供状态与触发图片。Gemini-S1 Agent 接收事件，生成自然语言提醒，并调用 `gemini_qq_notify` 发送图文、调用 `gemini_audio_speak` 请求现场播报。该场景展示了 OpenVela Agent 如何把外部感知、板载环境和两种用户反馈组织成同一闭环。



7. Skill：不修改核心代码扩展多步骤行为

项目提供 Markdown 格式的智能家居 Skill，描述工具选择、参数约束、失败处理和多步骤执行顺序。例如视觉监护场景规定：先确认视觉事件及图片来源，再生成通知内容，随后分别调用 QQ 和音频工具，最后根据两个工具的返回状态形成最终回答。

Skill 将“场景策略”与“C 工具实现”分离。新增多步骤家居流程时，可以先修改文本策略和工具组合，而无需改写 Agent 推理核心。

8. 真实开发板适配

项目以 Gemini-S1 实机为运行目标。工具扩展接入官方 `packages_ai_agent` 构建与注册路径；环境读取优先订阅 OpenVela uORB 主题，在主题不可用时读取 `/dev/sensor/*` 设备节点。项目同时提供构建脚本、部署脚本、板端目录说明和运行命令。

这一适配证明 OpenVela Agent 不只能够在 QEMU 中调用模拟工具，也能够在资源受限硬件上读取真实环境并连接其他现实设备。

9. 异常处理与安全边界

- 工具入口校验 JSON、必填参数、字符串长度和枚举值；
- 传感器字段分别返回有效状态，避免部分失败污染整次观测；
- QQ 工具区分本地配置、鉴权、网络请求和提供方拒绝；
- 音频工具区分已播放、已排队和失败，不把 outbox 写入等同于物理播报；
- 未注册能力不会被静默执行；
- 用户密钥、网络凭据和个人标识通过本地配置注入，不提交公开仓库；
- 外设适配器只负责协议转换，工具选择与结果解释仍由 Gemini-S1 上的 Agent 完成。

10. 实验结果与当前边界

已形成的工程证据包括：Gemini-S1 上运行的 `ai_agent`；六个工具的注册与执行代码；环境传感器读取路径；设备状态记录；外部视觉事件与对应图片；QQ 图文反馈；音频能力事件及其状态返回；构建、部署和复现文档。

当前仍在完善 Gemini-S1 到特定蓝牙音箱的原生 A2DP 兼容性。原生通道不可用时可通过适配器传输，但材料不会把“请求已排队”描述为“音箱已经播放”。长期无人值守稳定性和多 LLM 对比属于后续工作。

11. 评分映射与加分点

评审项	本项目证据
自定义工具 25%	6 个进入官方 Registry 的结构化工具，至少两项直接读取 Gemini-S1 真实状态
自然语言闭环 15%	普通聊天验证交互入口；温度读取/播报和视觉监控完成工具调用闭环
可复现与异常 10%	构建部署脚本、双路径传感器读取、结构化错误与执行状态
工程化与合规 30%	官方包集成、实机适配、README/运行说明、许可证与凭据隔离
文档与演示 20%	技术报告、网页 PPT、视频脚本、工具矩阵和日志证据
后续扩展基础	Skill、多设备能力事件、真实开发板和安全边界

12. 复现顺序与结论

复现顺序为：构建并部署 Gemini-S1 固件与 Agent 扩展；确认 `ai_agent` 与工具注册；进行普通聊天；询问并播报真实温度；接入视觉事件；核对 QQ 图文、音频状态和板端记录。详细命令见《运行说明》。

结论： 本项目完成了一个运行在 Gemini-S1 上的 OpenVela 真实设备 Agent。它能够理解自然语言、读取板载环境、接收外部视觉事件、调用图文与语音工具并核验执行状态，完整回答了初赛关于自定义工具、自然语言闭环、实机运行和可复现性的要求。