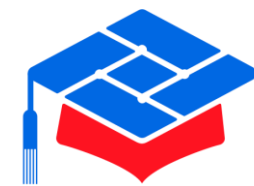


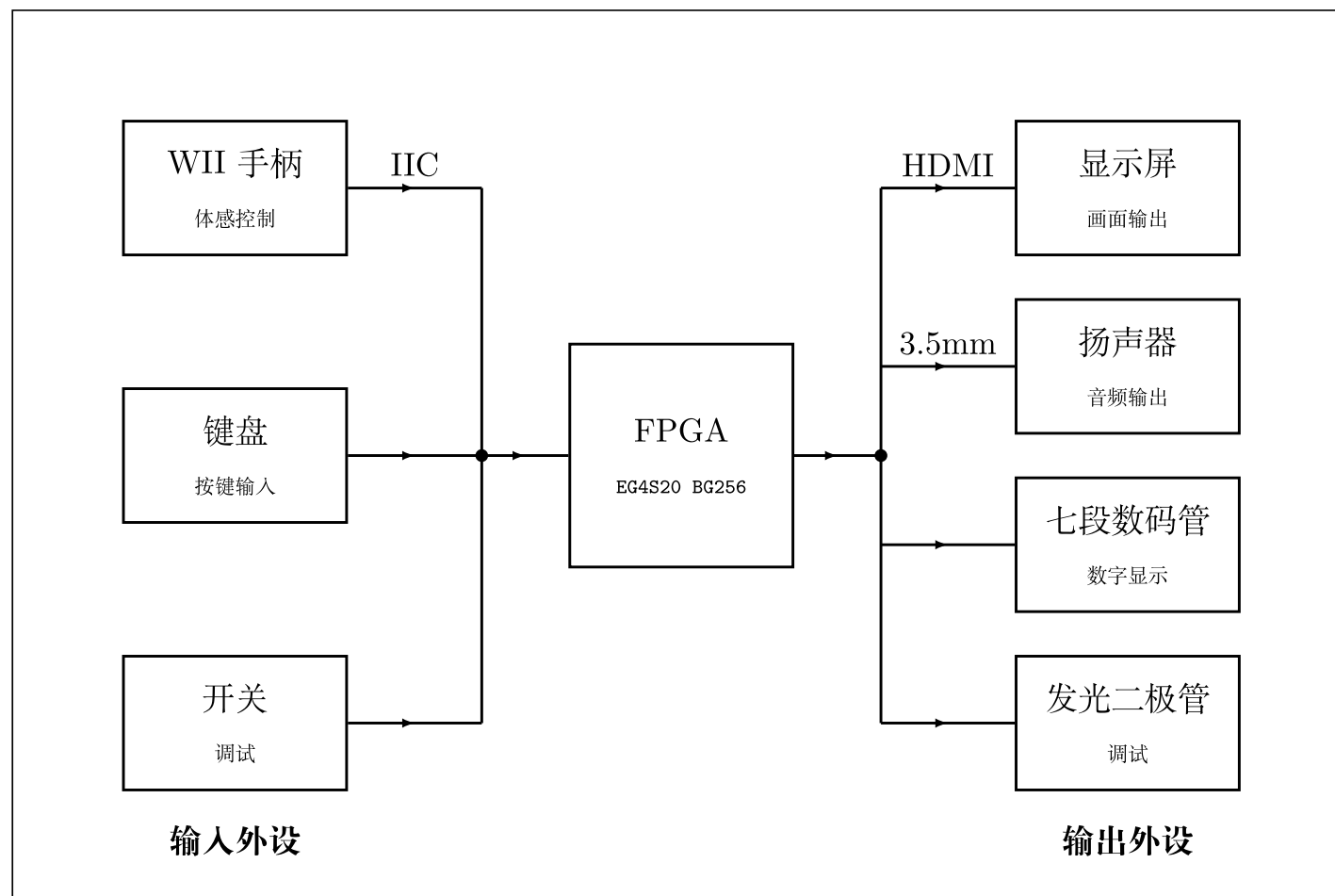
# 基于ARM处理器的智能游戏机

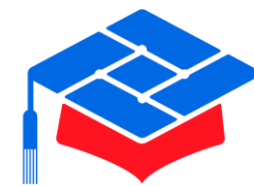
棱镜项目组CICC1164 李宇轩 张岩 李铭杰



# 系统总体框图

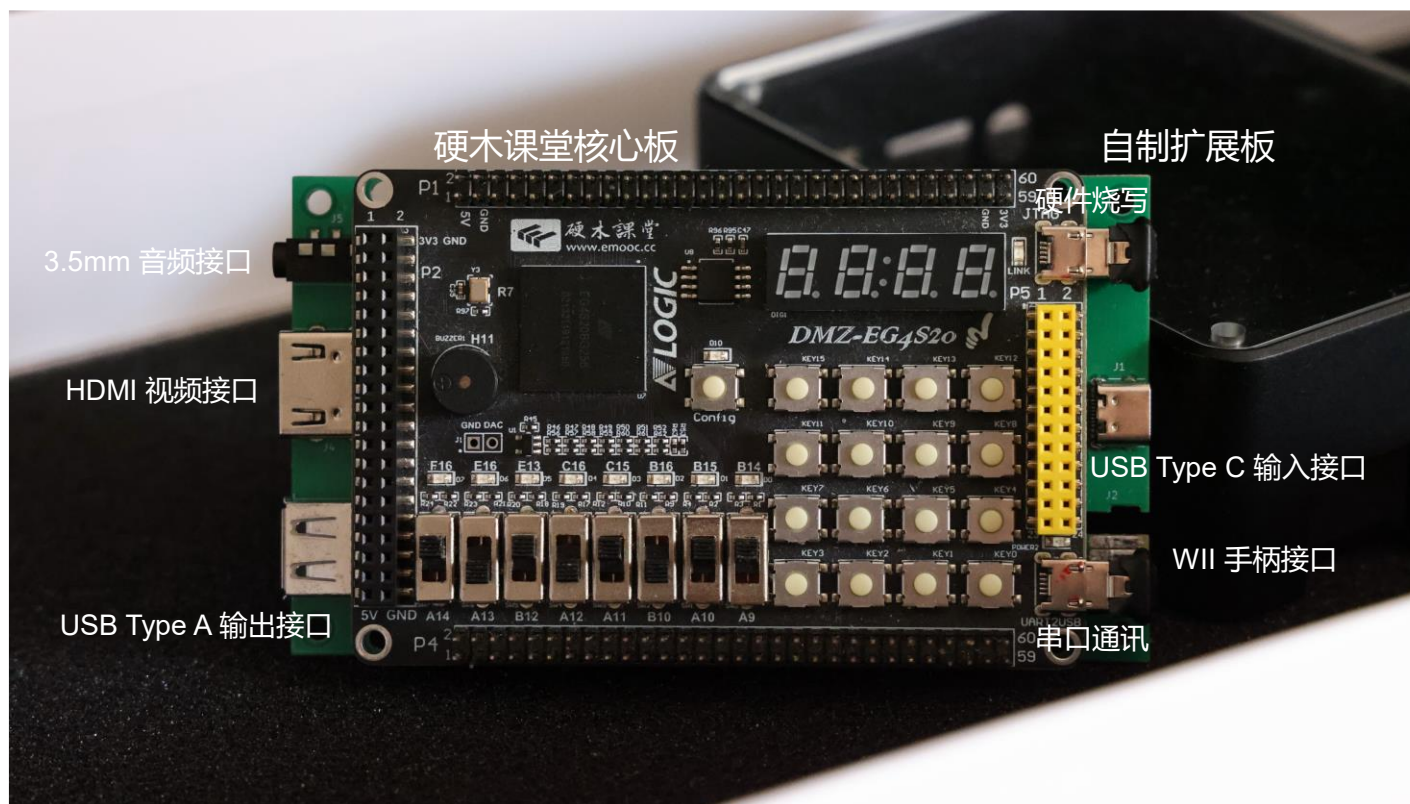
- EG4S20 BG256
- WII 手柄 体感控制
- 显示屏 HDMI 画面输出
- 扬声器 3.5mm 音频输出

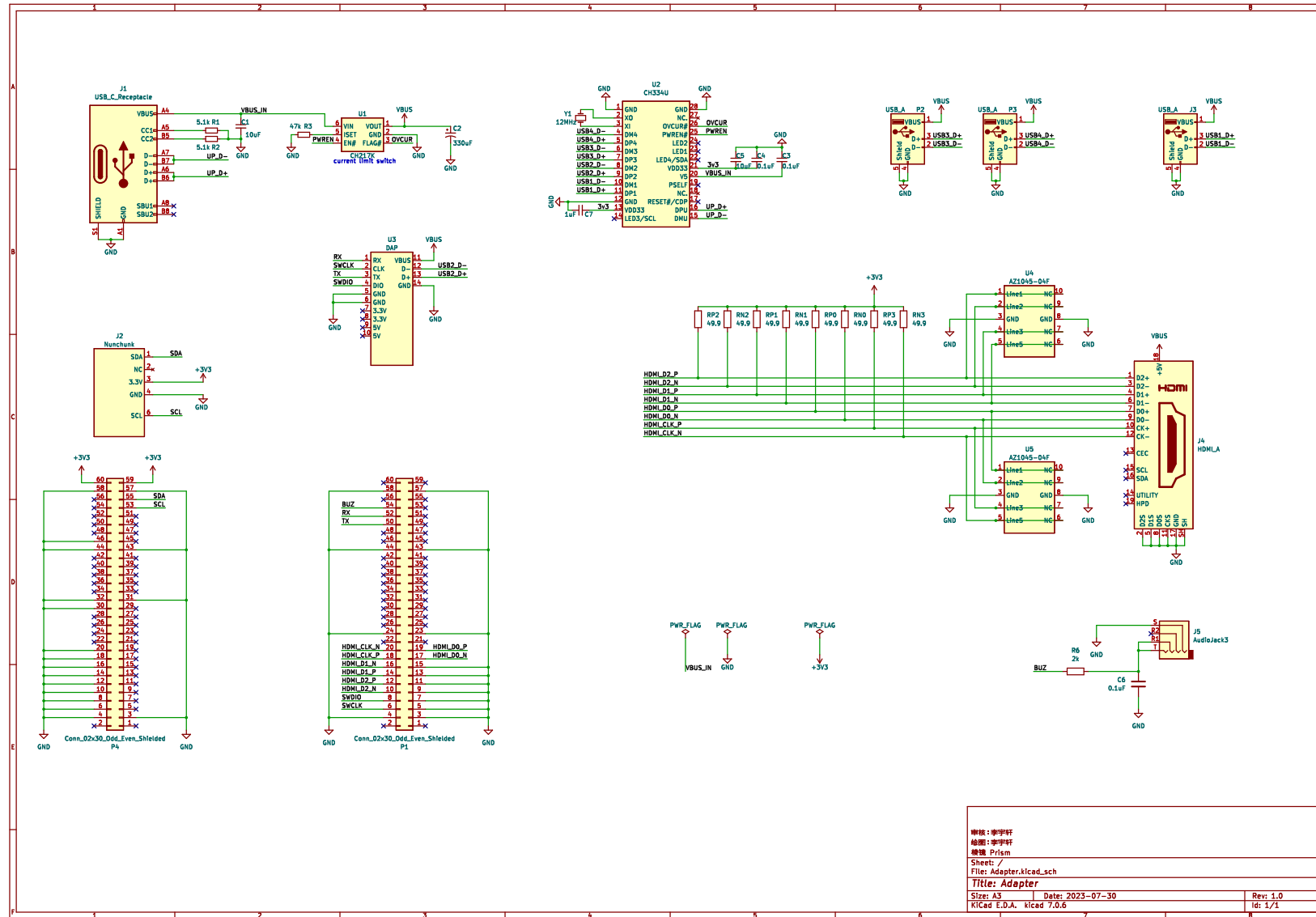




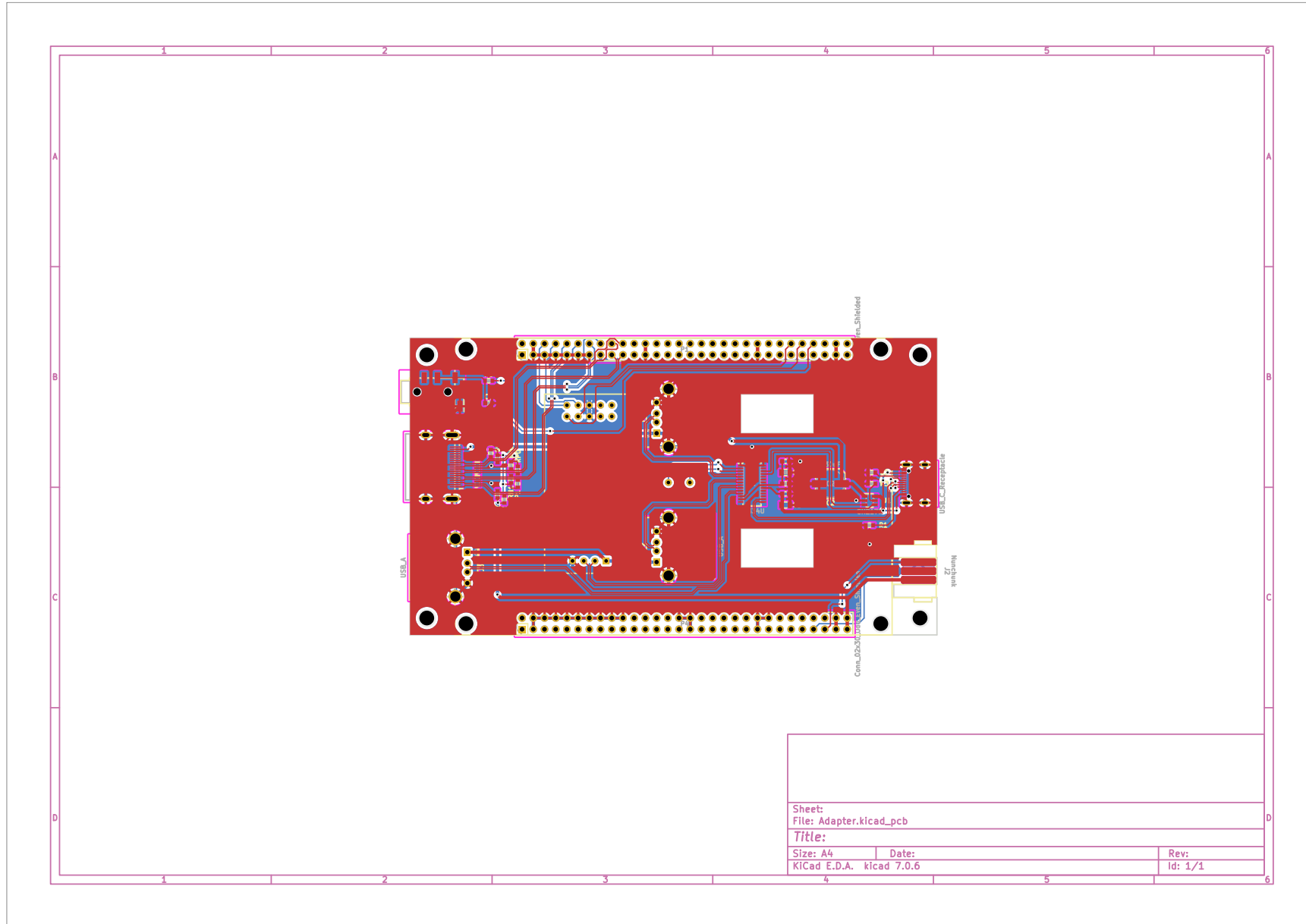
# 系统展示

- 硬木课堂EG4S20核心板
- 自制扩展板
- 自制产品外壳

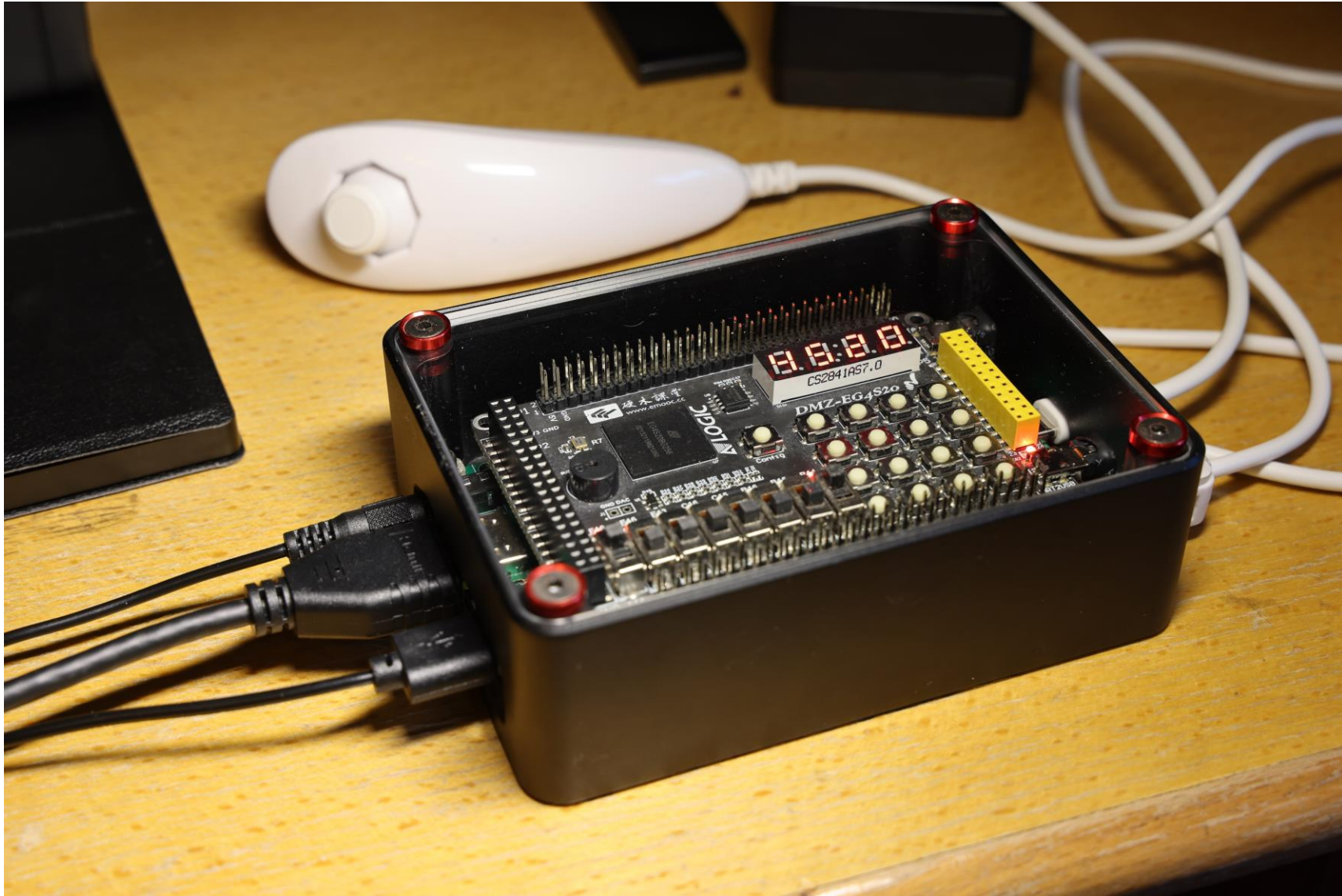




自制板卡的原理图

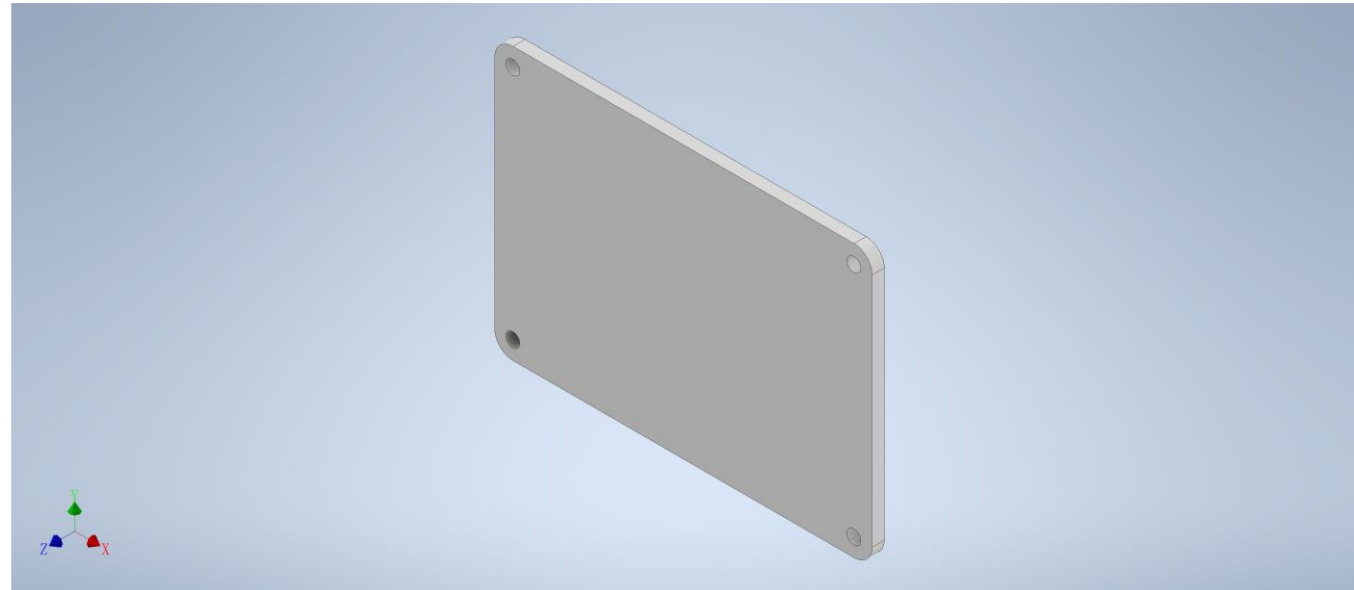
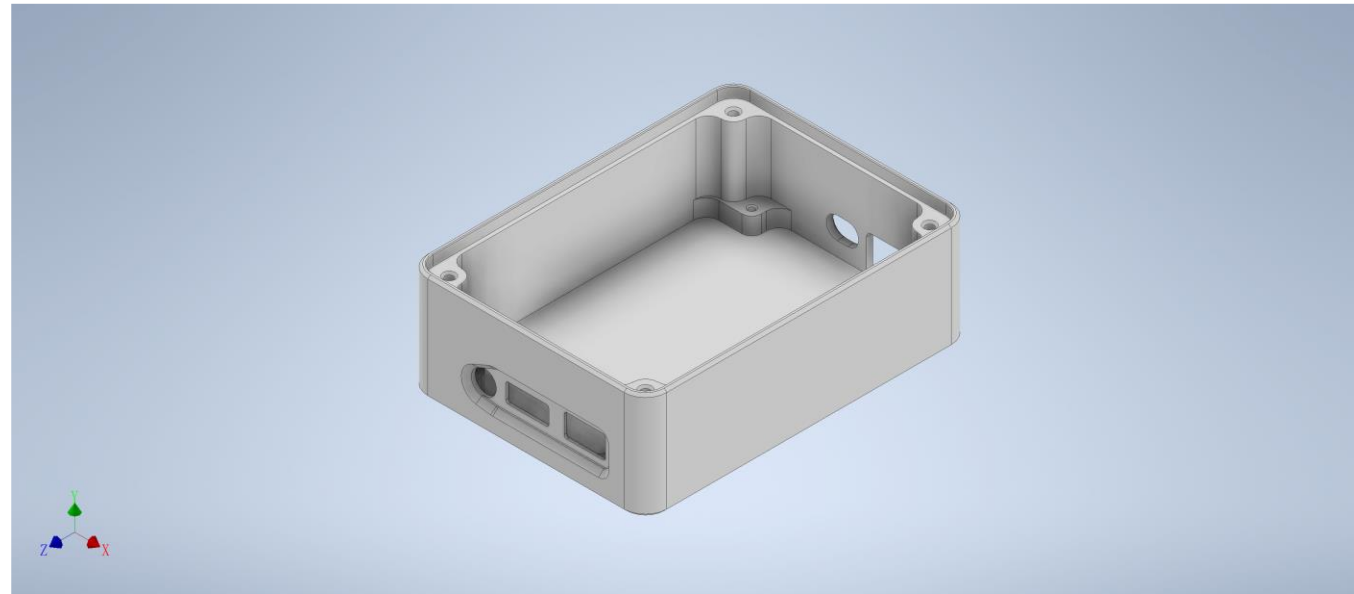


自制板卡的版图

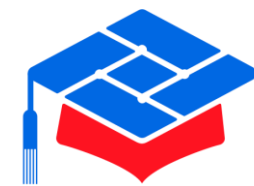


自制产品外壳实拍图



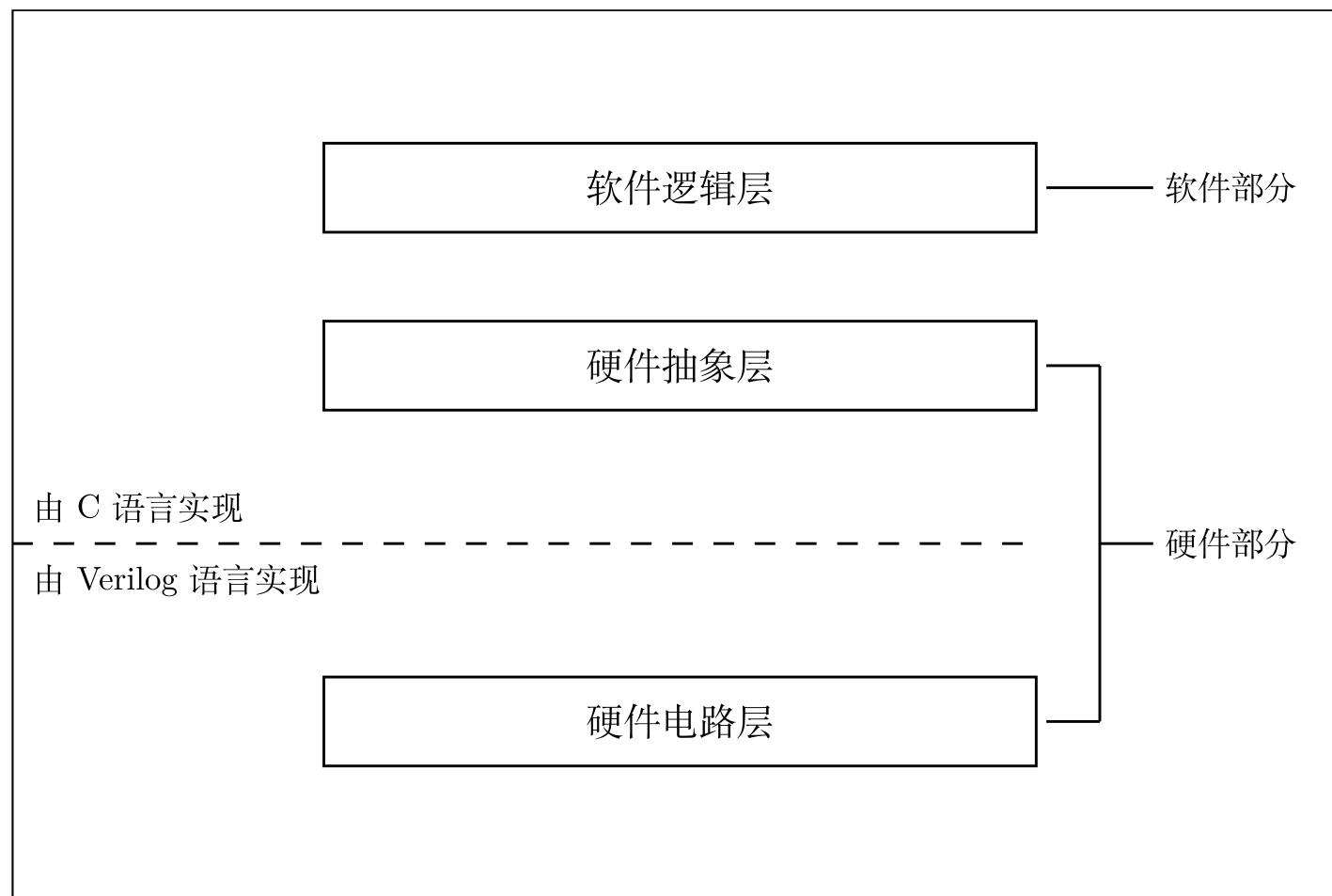


自制产品外壳的3D模型



# 系统架构

- 硬件部分：实现需要复杂时序控制的部分，如显示编码等，将硬件状态控制封装为寄存器。
- 软件部分：实现具体的游戏逻辑和游戏画面。

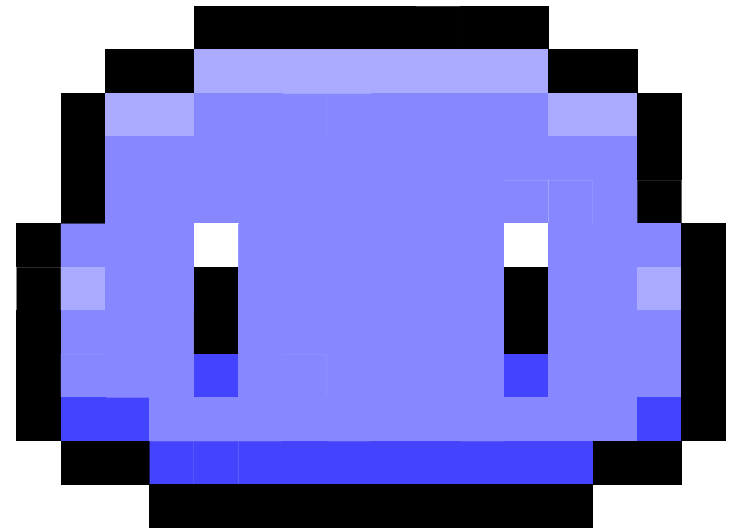
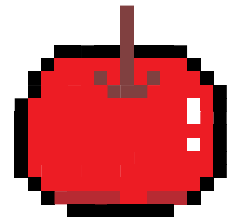
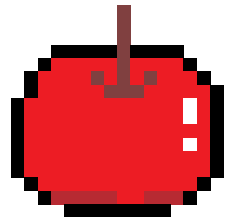
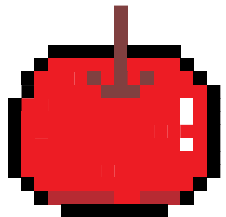






# 设计亮点1: HDMI显示

- 显示分辨率1024\*600。
- 显示模块实现了HDMI编码和VESA时序，适用性好。
- 显示模块应用片上SDRAM资源作为显存。
- 显示由软件控制，灵活性高，游戏画面丰富。





# 显示IP的技术要点

- 显示模块由两部分组成：SDRAM控制器模块、HDMI编码模块。
- 显存的写入——以LCDRectangle函数的工作为例。
- 显存的读取，以及HDMI编码和VESA时序。
- 有关Ping Pong机制。



# 显示IP的技术要点

- 第一个周期中，M0向显存1写入，HDMI从显存2读取。
- 第二个周期中，M0向显存2写入，HDMI从显存1读取。
- 读写在两块空间中交替进行，避免闪烁。



## 设计亮点2：Wii Nunchuk手柄

- Wii Nunchuk手柄基于IIC协议，通过软件实现。
- 2个按钮（Z按钮和C按钮）
- 2轴摇杆（摇杆控制）
- 3轴加速度传感器（体感控制）
- 智能化的交互体验





## 设计亮点2: Wii Nunchuk手柄

- 手柄初始化: START、0x40、0x00、STOP
- 手柄数据读取: START、0x00、STOP、READ 6 byte。



Byte	Bit							
	7	6	5	4	3	2	1	0
1	JX[7:0]							
2	JY[7:0]							
3	AX[9:2]							
4	AY[9:2]							
5	AZ[9:2]							
6	AX[1:0]		AY[1:0]		AZ[1:0]		$\neg$ BC	$\neg$ BZ

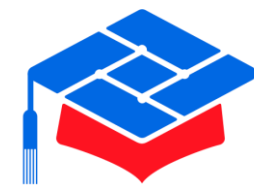
Wii Nunchuk手柄的数据



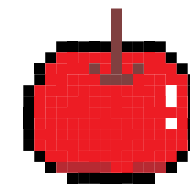
# 项目开源

<https://github.com/liyuxuan3003/PrismGC>

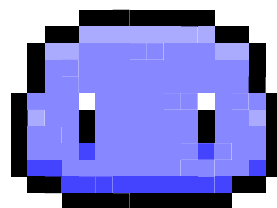




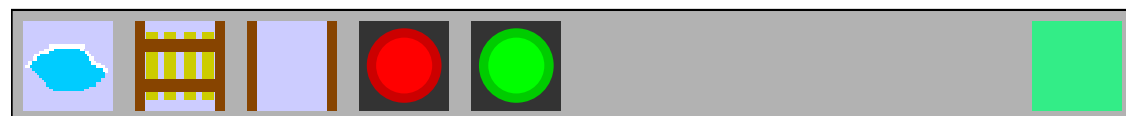
# 游戏设计：无敌史莱姆大冒险



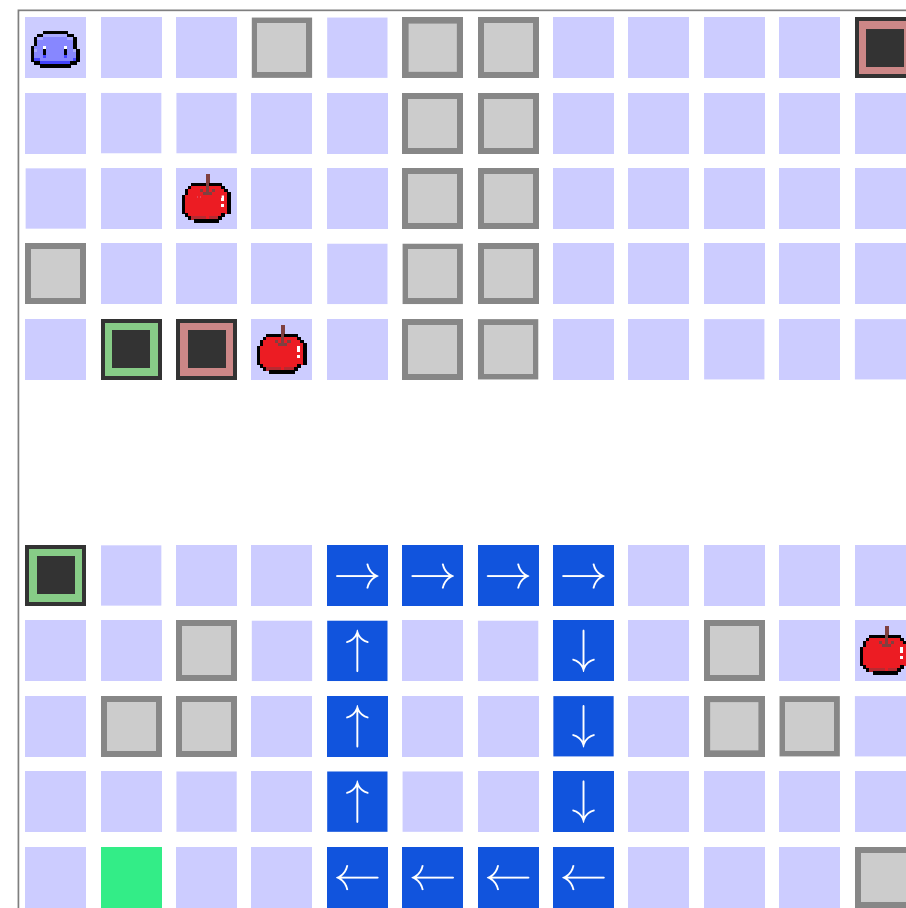
- Invincible Slime's Adventure
- 创意核心玩法
- 精美游戏角色
- 丰富关卡（多达48关！）

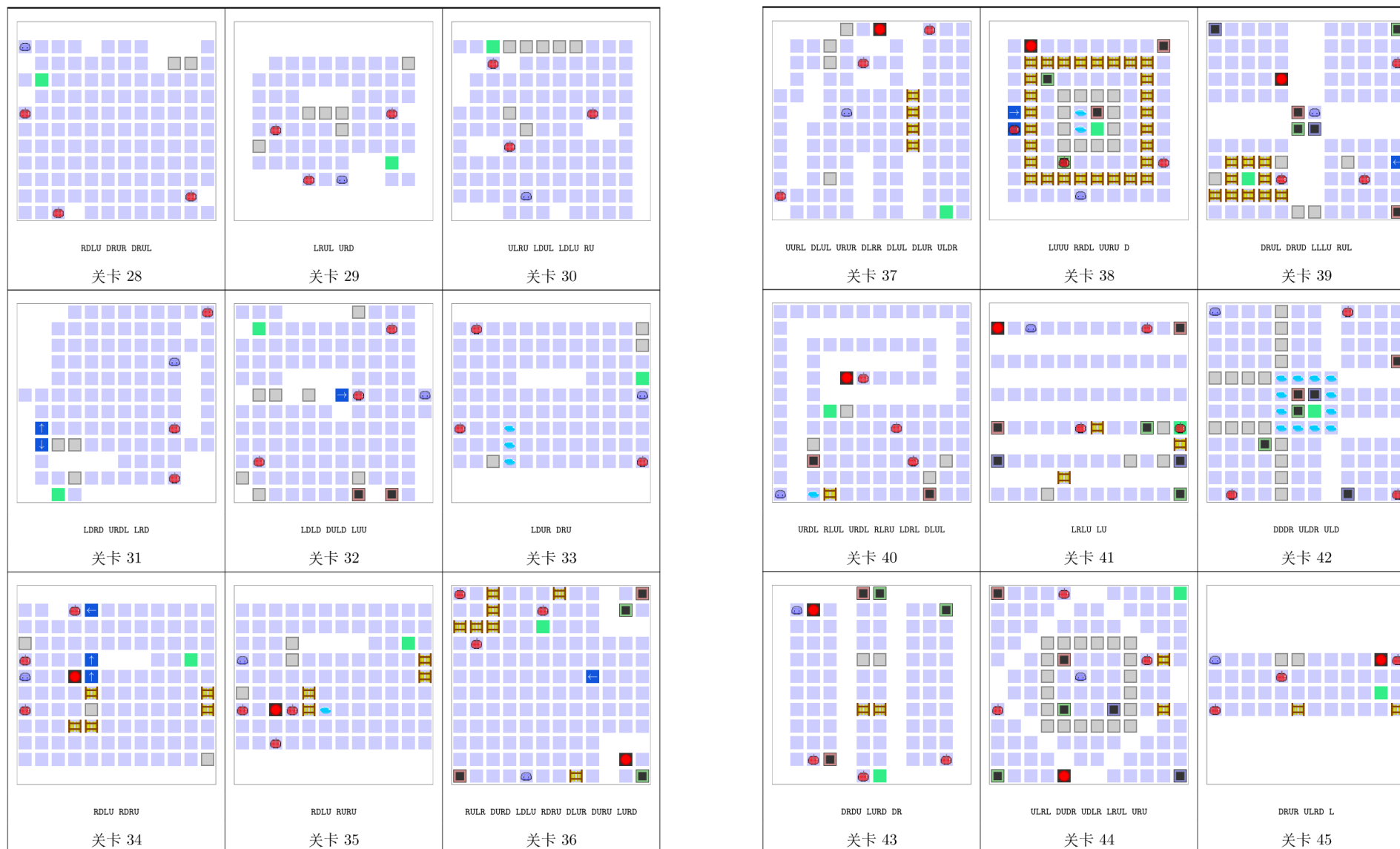
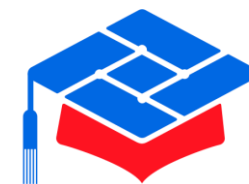


冰面 雪堆 机关-左 机关-右 机关-上 机关-下 碎石 法阵 1 法阵 2 法阵 3



陷阱 闸门-关 闸门-开 按钮-关 按钮-开 庇护所





部分关卡地图展示

Thanks!