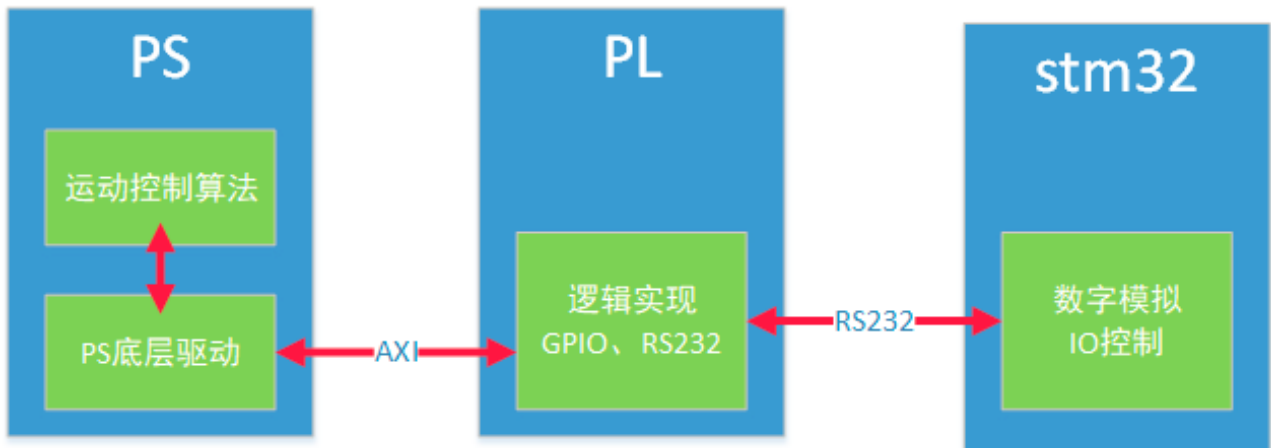


软件架构



zynq芯片的PS部分运行Linux操作系统，上层为机器人运动控制算法，这部分已经完成。PS部分的底层为与PL之间的通信。PL负责信号与逻辑处理，包括GPIO口的操作，串口通信等。PS与PL之间通过AXI总线进行通讯，主要实现PS对外部状态的读取和控制信号的发送。stm32作为扩展处理器，主要用于处理数字IO和模拟IO，由PL部分对其实现主动状态控制和读取，通过RS232实现（该处通信方式不限，可根据具体情况选择）。

该板卡主要完成的是对机器人轨迹插补后的脉冲分发，在轨迹插补完成后，PS会向PL按照预设的周期发送每个运动轴需要向驱动器发送的脉冲数，PL部分先将接收到的数据存在缓冲区内，然后逐条提取数据，解析，完成脉冲发送。脉冲的频率=脉冲数量 / 周期。

在发送脉冲过程中如果发现伺服状态关闭、驱动出错、急停触发、超出限位等情况，则立马停止发送脉冲，并向PS发送出错信息。

其他功能则基本都是GPIO的一般操作，表现为输出状态切换和输入读取。

PS与PL通信

- 设置周期
- 发送脉冲控制
- 获取发送的脉冲数
- 设置发送的脉冲数
- 发送伺服状态
- 获取伺服状态
- 获取输入信号

- 获取输出信号
- 发送输出信号
- 获取错误信息
- 获取外部指令（运行、停止、回归初始位置）
- 设置回零传感器使能
- 设置限位传感器使能

PL与stm32通信

- 获取数字输入状态
- 获取数字输出状态
- 设置数字输出状态
- 获取模拟输入状态
- 获取模拟输出状态
- 设置模拟输出状态