

# plan

整个过程：硬件集成方案、软件系统开发 和 最终物理封装 (\*)。

---

## 第一步：硬件集成方案 (解决杜邦线问题)

需要设计一块专属的“传感器与通信Pmod扩展板”来替代杜邦线。

### 1. 设计目标

这块扩展板（PCB）的目标是：

- 一端通过标准的Pmod连接器，牢固地插在ZCU102的Pmod接口上。
- 另一端为Lepton 3.5的转接板和SIM800L模块提供稳定、可靠的接口和供电。

### 2. PCB设计要点

- **Pmod连接器:** 在PCB上放置一个或两个2x6针的Pmod公头连接器，用于连接ZCU102。您需根据Lepton的SPI/I2C需求和SIM800L的UART需求，规划好引脚分配。
- **Lepton接口:** Lepton本身需要一个转接板（Breakout Board）。您有两个选择：
  - **方案A (推荐):** 在定制PCB上，设计一个母座，让Lepton的转接板可以直接插在您的PCB上。
  - **方案B:** 直接将Lepton转接板焊接到您的定制PCB上。
- **SIM800L接口:** 在PCB上为SIM800L模块预留插针或母座，用于连接其UART（TX, RX, GND）及电源引脚。
- **独立供电电路 (关键):**
  - SIM800L在发射信号时瞬时电流很大（可达2A）。**绝对不能**从Pmod的3.3V供电。
  - 需要在定制PCB上设计一个独立的电源电路。可以从一个外部5V电源输入，然后通过一个大电流的LDO或DC-DC降压芯片，为SIM800L提供稳定的4.0V左右电压。
  - Lepton也需要稳定的3.3V供电，同样建议由这个独立的电源电路提供。

### 3. 实现方式

- **打样生产:** 嘉立创(JLCEDA)
  - **焊接:** 自己焊接元器件，或者利用平台的SMT贴片和焊接服务。
-

## 第二步：软件系统开发

有了可靠的硬件连接，接下来就是在ZCU102上开发软件系统。

### 1. 操作系统与环境

- **搭建PetaLinux系统:** 为您的ZCU102定制一个Linux操作系统。这是标准的嵌入式开发流程，可以精确控制包含哪些驱动和库。
- **使能接口驱动:** 在PetaLinux的设备树（Device Tree）中，正确配置您使用的Pmod接口，将其声明为SPI、I2C或UART设备。这样，在Linux系统中就会出现对应的设备节点（如 `/dev/spidev1.0` , `/dev/i2c-1` , `/dev/ttyPS1` ）。

### 2. Lepton数据采集

- **驱动开发:** 需要编写一个用户态程序，通过Linux的SPI和I2C设备节点，来与Lepton进行通信。这需要遵循Lepton的数据手册，实现其VoSPI（Video over SPI）协议来抓取每一帧热成像数据。
- **开源库利用:** 可以在GitHub上寻找现有的Lepton驱动代码或库，将其交叉编译后移植到PetaLinux系统中。

### 3. SIM800L通信控制

- **串口编程:** 编写一个函数或一个独立的线程，通过操作对应的UART设备节点（`/dev/ttyPS1`），向SIM800L发送AT指令（例如 `AT+CMGS` 用于发送短信），并读取返回结果。

### 4. 跌倒检测算法实现

- **算法开发:**
- **性能优化:** 可以利用ZCU102的ARM A53处理器运行OpenCV等库进行图像处理，或者将计算密集的部分用Vitis HLS工具转换成硬件逻辑，放到FPGA（PL端）中进行加速。

---

## 第三步：最终物理封装

目标是做出一个整洁、可靠的原型机？

### 1. 设计外壳 (Enclosure)

- **3D建模:** 使用SolidWorks, Fusion 360等软件，根据ZCU102、您的定制扩展板、天线、电源接口等部件的尺寸，设计一个精确的外壳。
- **3D打印:** 将设计好的模型通过3D打印机制作出来。

## 2. 组装与固定

- **主板固定:** 在外壳内部设计螺丝孔位，将ZCU102主板和您的定制扩展板牢固地固定在外壳上。
- **接口引出:** 在外壳上为电源、USB、网口以及SIM800L的天线预留开口。
- **散热考虑:** ZCU102功耗不低，需要在外壳上设计散热孔，甚至可以加装一个小风扇。