

## 第一章 《振南当年入门 C 语言和单片机的那些事儿》

### 1、注定堕入单片机

#### 1.1 懵懂好奇的我

（小时候好奇的性格经常让我屁股开花。初中开始对计算机产生兴趣，并一发不可收拾。）

#### 1.2 我的 C 语言学习经历

（上大学后自学 C 语言。遇到“能人”加入 ACM 竞赛。感觉 C 语言乐趣多多，程序如人生。）

#### 1.3 C 语言的顶级赛事

（ACM 国际程序设计竞赛在东北被我们发扬光大。ACM 竞赛浙大的一段传奇佳话。振南在关注的 IOCCC 国际混乱 C 代码大赛。网吧包宿学 C 语言惊呆室友。）

#### 1.4 岔路口上选择单片机

（搞纯软件还是搞单片机，这是一个抉择。鬼才杜撰拉我进入单片机快车道。）

#### 1.5 窗户纸破了

（入门阶段的困惑，看破 C 语言与单片机之间的鸿沟。）

### 2、看穿单片机

#### 1.1 CPU 模型

（CISC 与 RISC 指令集。CPU 如何执行指令。汇编不是第一代编程语言，打孔纸带才是。）

#### 1.2 存储器模型

（存储器就是一个指令和数据的容器。）

#### 1.3 总线模型

（地址、数据和控制三大总线。贯穿整个单片机芯片的通路。）

#### 1.4 外设模型

### 3、单片机跑起来

#### 3.1 时钟系统

（时钟是单片机激励和血液。时钟频率不能无限提高。）

#### 3.2 二进制

（为什么单片机采用二进制？振南告诉你如果单片机使用十进制会怎样？）

#### 3.3 中断机制

（中断不是在给 CPU 捣乱。中断对于单片机为什么如何重要？）

## 第二章 《C 语言的一些“骚操作”及其深层理解》

### 一、字符串的实质就是指针

（如何将 35 转为对应的十六进制字符串“0X23”？）

### 二、转义符\

（打入字符串内部的“奸细”。）

### 三、字符串常量的连接

（字符串常量是双面胶，你知道吗？）

### 四、长字符串的拆分技巧

（GPS 数据帧 NMEA、Shell 命令行和 AT 指令的解析，是长串拆分的典型应用。）

### 五、巧取数值的各位数码

(玩多位数码管的必有操作。)

## 六、printf 的实质与使用技巧

(自认为很了解 printf? 那你试过向 3 个 UART 打印吗? 或者打印到液晶屏上?)

## 七、关于浮点数的传输

(浮点只是一种假象, 看清它的本质。)

## 八、关于数据的直接操作

(如何快速计算浮点的相反数, 乘以-1.0? 再想想。)

## 九、浮点的四舍五入与比较

(老师说浮点不能直接判等, 为什么?)

## 十、出神入化的 for 循环

(for 循环很熟悉了吧? OK, 振南出了几道题, 来试试。)

## 十一、隐藏的死循环

(我们在明处, 有时死循环在暗处。)

## 十二、看似多余的空循环

(没用的东西?)

## 十三、独立执行体

(这个概念 C 语言里没学过? 那就对了, 我经常用。)

## 十四、多用()无坏处

(万物皆可加括号。)

## 十五、==的反向测试

(把==错写成=, 能让你调程序调到吐血。)

## 十六、赋值操作的实质

(让数学教授困惑半生的 C 语言赋值操作。)

## 十七、关于补码

(摊牌了, CPU 其实不会作减法。)

## 十八、关于-1

(-1 就是全 F, 全 F 就是-1。)

## 二十、字节快速位逆序

(时间与空间的相互转化--计算机中的相对论)

## 二十一、关于 volatile

(有些东西不可优化。)

## 二十二、关于变量互换

(位操作的奇妙。)

## 二十三、关于 sizeof

(告诉你关于 sizeof 那些少人关注的问题。)

## 二十四、memcpy 的效率

(小小的函数也有大大的背景)

## 二十五、[]的本质

(你以为[]只是数组下标?)

## 二十六、#与##(串化与连接)

(一个不曾出现在 C 语言教材中的知识点)

## 1、烧录方式的更新迭代

### 1.1 古老的烧录方式

（怀旧一下，单片机高压烧录器。）

### 1.2 ISP 与 ICP 烧录方式

（还记得当年我们玩过的 AT89S51？）

### 1.3 更方便的 ISP 烧录方式

#### 1.3.1 串口 ISP

（是 STC 单片机成就了我們，还是我們成就了 STC？）

#### 1.3.2 各种 USB ISP

##### 1) AVR

##### 2) C8051F

##### 3) MSP430

（在当前 STM32 一统天下的时代，上面这些单片机你还在用吗？）

## 2、关于 Bootloader

### 2.1 Bootloader 的基本形态

（Bootloader 先行，APP 在后。）

### 2.2 Bootloader 的两个设计实例

#### 1) 带 Shell 命令行的串口 BL

#### 2) 插 SD 卡即烧录的 BL

（上面这两种 BL 在实际应用中最常见，还讲了一下 Linux 的 Uboot。）

### 2.3 BL 实现的要点

#### 1) 芯片体系架构要支持

#### 2) ROM 要支持 IAP

#### 3) APP 程序的配套修改

#### 4) BL 中的跳转代码

（振南会竭尽所能的去讲解，但 BL 实际上能不能跑通，只能祝你好运。）

## 3、花样百出的 BL

### 3.1 BL(串口传输固件)的实现与延伸

（告诉你一个秘密：STM32F103C8T6 的后 64K ROM 也能用，不信你试。）

### 3.2 10 米之内隔空烧录的实现

（一部安卓手机在手，空中升级调试全有。）

### 3.3 BL 的分散烧录

（你以为 BL 只能给自己烧程序？）

## 4、不走寻常路的 BL

### 4.1 Bootpatcher

（反其道而行之，APP 先行，BL 在后。）

### 4.2 APP 反烧 BL

（你以为只能 BL 烧录 APP？）

## 第四章 《大话文件传输》

### 1、Xmodem 协议族

#### 1.1 Xmodem 的传输过程

#### 1.2 Ymodem 的传输过程

#### 1.3 关于 Zmodem

(1978 年 IBM 的工程师 Ward christensen 的一次文件传输实验，Xmodem 协议族就此诞生，你知道吗？卫星空间文件传输也用它。)

#### 1.4 AVRUBD 的传输过程

(知道网友 shaoziyang 的 AVR 通用 Bootloader 吗？它用的就是 AVRUBD。)

### 2、更多的文件传输协议

#### 2.1 振南的 CAN 文件传输

(用 CAN 总线作文件传输，有点为难自己了，但我实现了。)

#### 2.2 通过 HTTP 下载文件

(天涯海角都不怕，我们从云端进行固件升级。)

#### 2.3 Json 传输文件

(巧妙的 Base64 编解码，你应该了解一下。)

## 第五章 《倾斜传感器并不简单》

### 1、倾斜传感器的那些基础干货

#### 1.1 典型应用场景

(危楼、边坡、古建筑都是对倾斜敏感的。)

#### 1.2 倾斜传感器的原理

##### 1.2.1 滚珠式倾斜开关

##### 1.2.2 加速度式倾斜传感器

##### 1)直接输出倾角

##### 2)加速度计算倾角

##### 3)倾角精度的提高

(如果没看懂，振南教你个好办法：再看一遍。)

### 2、倾斜传感器温漂校准的基础知识

#### 2.1 温漂产生的根源

(万物皆受温度影响。振南给你讲讲“调皮的尺子”。)

#### 2.2 温漂的真实例子

(某项目的奇怪现象，一到中午数据就乱跳。亮一下壮观而精密的自动化校准装置。)

### 3、静态温控的实现与温补装置的迭代

#### 3.1 制冷原理

(振南告诉你如何对传感器温漂校准。温度控制不难，但是不允许有振动，你作得到吗？)

#### 3.2 静态温度控制

##### 3.2.1 TEC 制冷

##### 3.2.2 散热方案

##### 3.2.3 倾角温补校准装置设计方案

##### 3.2.4 多级 TEC 制冷

##### 3.2.5 物理制冷

##### 3.2.6 半导体制热

##### 3.2.7 温控策略

##### 3.2.8 多路温度的同步控制

(如何安静的制冷？TEC 及阵列、水冷、干冰、铝注冷技术、PTC、保温材料、比热容、热阻，还有温控算法，这些你应该了解一下。)

### 4、倾角校准与数据拟合

- 4.1 倾角校准装置的构成
- 4.2 倾角温补校准与数据拟合
- 4.3 分段校准的质疑

（多阶拟合算法，还有开源的 Polyfit 方案。来看看最终效果：温度乱舞，传感器却无波动。）

## 5、其它细节

- 5.1 真值的读取
- 5.2 规避震动干扰
- 5.3 克服地面不平问题
- 5.4 减震设计

（万事的成败在于细节。）

## 第六章 《CPU，你省省心吧！》

### 1、石油测井仪器

#### 1.1 背景知识

（了解一下石油行业。石油到底是怎么找到的？）

#### 1.2 测井数据采集的实现

##### 1.2.1 最直接的初级方案

##### 1.2.2 加入 DMA 的优化方案

（看看一般工程师与高手在技术实现上到底有什么区别。充分利用硬件资源。）

### 2、巧驱摄像头

#### 2.1 摄像头时序分析

#### 2.2 使用 DCMI+DMA

#### 2.3 自搭外部电路

（图像一闪而过，普通单片机你抓得住吗？那我们就给他配个“龙抓手”，硬件 FIFO。）

### 3、M0 单片机巧驱 7 寸液晶大屏

（CPU 就算跑冒烟也刷不过来。让 CPU 省省心，来看看振南的方法是否足够巧妙。）

## 第七章 《制冷设备监测历险记》

（敬佩马斯克及多位实业创业大佬。到任何时候，我们都应该保持初心。）

### 1、需求的产生与硬件方案

#### 1.1 冷食的利润贡献

#### 1.2 冷设监测系统的困难

（制冷设备对于便利店为何如何重要？了解一下你所不知道的便利店和新零售行业。关于电力线载波通信的论战。）

### 2、电路设计

#### 2.1 防护电路

##### 2.1.1 强电防护

##### 2.1.2 弱电防护

（浪涌、脉冲群、静电、过压、雷击，你的电路扛得住吗？加些防护吧。）

#### 2.2 电路复用

(电路设计，仔细思考一下，不要作重复劳动。)

### 3、协议设计

#### 3.1 内外机通信协议

(电力线通信环境是复杂而恶劣的。振南设计的时分复用与冗余编码协议，了解一下。)

#### 3.2 主机与 WIFI Agent 通信协议

(乐鑫 ESP8266 连接 WIFI，数据上私有云。Json 了解一下。)

### 4、自动化生产与测试

#### 4.1 自动化烧录

#### 4.2 自动化测试

(芯片预处理、自动化烧录和测试，半个月生产 9000 套硬件，看看我是如何作到的。)

### 5、工程测试与安装

#### 5.1 工程测试(手机蓝牙远程调试)

#### 5.2 工程安装

(看我们上天入地安装设备。蓝牙调试，几十米外无线烧录，一部手机全搞定。)

### 6、冷设监测数据分析

(开放一些内部数据，看看实际效果。)

### 7、冷设监测故障预判作用评估

#### 6.1 故障预判时效

#### 6.2 对维修保养的验收指导作用

#### 6.3 故障报警受气温的影响

(努力没有白费，省下的是实实在在的真金白银。)

### 8、冷设预警的典型案列

#### 1) 申虹路某店

#### 2) 恒通商务园某店

(这里有 ABC IOT 系统的内部监测数据，一切的努力都归结于这些曲线上。)

向共同奋斗过的同志们致敬，  
尤其是我硬件团队里的宏涛，  
这段经历我永远不会忘记。

## 第八章 《FFT 你知道？那数字相敏检波 DPSD 呢？》

### 1、DPSD 的基础知识

#### 1.1 应用模型

#### 1.2 原理推导

#### 1.3 硬件 PSD

(相敏检波，就是从繁乱复杂的信号中将我们关心的信号检出来，同时对相位敏感。  
数学原理，逃不掉的，硬着头皮看吧。)

### 2、DPSD 的典型应用

#### 2.1 石油测井仪器

(《科拉深孔》和《地心末日》这两部电影看一下。上天难还是入地难？来看看振南参与研发的地下探测仪器，高温高压高噪声，如何将数据采准？)

## 2.2 功率检测

(来看看振南如何检测 220V 交流电压、电流、功率和功率因子。)

## 2.3 电池内阻测量

### 1、直流检测

### 2、交流方法

(电池内阻反映了电池的放电能力和剩余电量，内阻如何测？来看一下。)

## 2.4 风速风向检测

### 1、风杯式（旋转式）

### 2、超声波风速风向

### 3、声共振风速风向（很牛的技术）

(如何测风速风向？用超声 TOF 法。那你 out 了，来看看来自英国山伯利团队的声共振技术，这是一项伟大的发明。振南也作出了原型机。)

## 第九章 《IO 口不够 扩展器来凑》

### 1、基于 74 系列芯片的廉价方案

#### 1.1 并行输出端口扩展

(74HC573 是 8 位并行锁存器，16 位并行锁存器你见过吗？)

#### 1.2 串行输出端口扩展

(74HC595 是 8 位串转并，那 16 位串转并你见过吗？菊花链级联你知道吗？顺便说一下振南趟过的那些芯片选型的坑。在芯片短缺，价格飞涨的大环境下，如何安全的进行芯片选型？)

#### 1.3 并行输入端口扩展

(数据选择器可用于并行输入端口扩展，更巧妙的方法在等你。)

#### 1.4 串行输入端口扩展

(74HC165 是 8 位并转串。165+595+菊花链级联是使用 74 系列芯片实现 IO 端口扩展的终极方案。)

### 2、基于专门 IO 扩展芯片的方案

#### 1、并行 IO 扩展芯片

##### 1) 8255

##### 2) CH351

#### 2、串行 IO 扩展芯片

(振南搜罗了国内外主要芯片厂商的 IO 扩展芯片方案，TI、NXP、沁恒等等。国产芯片在崛起，最安全的芯片选型，就是尽量用国产。中国芯片必胜！)

## 第十章 《硬件开发版本有点乱？赶紧用 Git》

### 1、关于 Git

#### 1.1 Git 的前世今生

(Git 和 Linux 的生父都是 Linus，振南给你讲讲当初关于 Git 的爱恨情愁，其背后其实是开源与闭源两左阵营的明争暗斗。)

#### 1.2 Git 的爆发

(Git 超越时代的分布式思想。振南再给你讲讲旧金山三个年轻人创办 GitHub，打败 Google，逆袭上位的创业故事。据说 GitHub 服务器要放到火星去？)

## 2、用 Git 管理软件代码

### 2.1 Git 的本地化使用

（以实例来讲解代码仓库的创建、提交、分支等基础内容。）

### 2.2 Git 的远端使用

（以实例来讲解仓库的克隆、推送等基础内容。）

### 2.3 代码拯救纪实

（Git 绝不会把代码弄丢。一次有惊无险的代码追回经历，根源是对 Git 机制理解不深。）

## 3、用 Git 管理硬件 PCB

（对于硬件资源你是如何管理的？final\_final\_打死不改\_final\_1.2.zip？还是用 Git 吧。）

### 3.1 Git 的增量式管理

（Git 具体是如何对资源进行管理的？）

### 3.2 AD 中的 Git

（AD 是原生支持 Git 的，让我们把它利用起来。）

### 3.3 PCB 工程的协作开发

（团队协作中的冲突是如何产生的？如何解决冲突？）

向开源者致敬！！

## 第十二章《各大平台串口调试软件大赏》

（串口的重要性不言而喻。为什么很多平台把串口称为 tty，比如 Linux、MacOS 等等，振南告诉你。）

### 1、各平台上的串口调试软件

#### 1.1 Windows

##### 1.1.1 STCISP

（感谢 STC 姚老板设计出 STCISP 这个软件。）

##### 1.1.2 SSCOM

（这应该是使用人数最多的串口助手软件了。）

##### 1.1.3 友善串口调试助手

##### 1.1.4 Xshell

1)多窗口排列

2)按钮定制

##### 1.1.5 SecureCRT

##### 1.1.6 PuTTY

##### 1.1.7 MobaXterm (MobaXVT)

##### 1.1.8 COMTool

（大家可以挑选适合自己的的软件下载下来试试。）

#### 1.2 Linux

##### 1.2.1 CuteCom

（这应该是 Linux 下最常用的串口软件。）

##### 1.2.2 minicom

（一个基于命令行的串口软件，有兴趣来见识一下。）

### 1.3 MacOS

#### 1.3.1 coolTerm

#### 1.3.2 友善串口调试助手

#### 1.3.3 Volt+（伏特加）

（用 MacBook 来搞硬件调试，你真土豪。）

### 1.4 iOS 与安卓

#### 1.4.1 基于 USB 的串口调试软件

##### 1、USB 串口调试助手

##### 2、Serial USB Terminal

#### 1.4.2 基于蓝牙的串口调试软件

##### 1、BlueSPP

##### 2、蓝牙串口

##### 3、Android 蓝牙串口 Pro

（用手机来调试串口，配上这些软件，真是一机在手，一切全有。）

## 2、串口监控的一些方案

### 2.1 硬件方案

（只需一根导线，串口双向数据尽收眼底。）

### 2.2 软件方案

#### 2.2.1 Ser232Mon

#### 2.2.2 Device Monitoring Studio

#### 2.2.3 CommMonitor

#### 2.2.4 AccessPort

#### 2.2.5 ComSpy（串口监视精灵）

#### 2.2.6 jnnevaluator

（串口抓数和协议分析利器，涵盖 Windows 与 Linux。）

## 第十三章 《我的电动车共享充电柜创业项目纪实》

（创业不易，九死一生。在泥沙俱下的洪流之中，你的努力可能仍然微不足道。）

### 1、关于创业思维

#### 1.1 原始思维

（振南较早期的创业行为。我是如何得到自由之身的。）

#### 1.2 升华思维

（那些年，我挣到了人气。我是如何一步步被广大圈内人士知道的。）

#### 1.3 思维的歧途

（要理性还是要任性，一切都在于你自己的选择。它们标志着不同的人生道路。）

#### 1.4 创业之心

（创业不易，但存创业之心，莫问前路坎坷。）

### 2、一切始于那一夜

#### 2.1 入伙创业

#### 2.2 决策成金

（那一夜，一家 KFC，我和合伙人长谈。决策决定人生走向。充电柜，有没有前途？）

### 3、智能充电柜的技术实现

#### 3.1 需求分析

（先定好充电柜要作成什么样子。）

#### 3.2 业务流程设计

##### 3.2.1 充电仓位的开放

##### 3.2.2 服务中与异常处理

##### 3.2.3 充电结束

（我们自认为考虑得很完备，那后来怎么样？你猜。）

#### 3.3 技术总体设计

##### 3.3.1 人员分工

##### 3.3.2 硬件总体设计

###### 1、整体拓扑

###### 2、仓位（柜格）设计

##### 3.3.3 电路原理

###### 1) 主控板

###### 1、主控 MCU

###### 2、稳压电路与电源控制

###### 3、CAN 收发器

###### 4、交流电压电流采集

###### 5、数码管、LED 与按键

###### 6、温度与亮度检测

###### 7、蓝牙无线串口

###### 2) 控制板

###### 1、继电器控制

###### a) 控制端反向电动势防护

###### b) 消弧措施

##### 3.3.4 总体形态

（以上内容基本包含的充电柜所有的硬件设计，包括电路、结构、外观等。）

##### 3.3.5 软件总体设计

###### 1、内部 Flash 的划分

###### 2、前后台与 Shell

###### 3、设备的自我诊断

###### 4、协议设计

###### 5、CAN ID 的自动分配

（不光有嵌入式哦，还包括云端的 PHP 脚本开发。振南给你讲讲什么是 HTTP POST。

所有电路板均可通过 Shell、云端、CAN 总线进行在线升级。还有无线 Shell，现场调试只需要一部手机。）

### 4、智能充电柜的市场投放

#### 4.1 批量装配

（装配工厂就是一间简陋的民房。凌晨 3 点，零下 3 度，四面透风，在这里我们完成 50 台充电柜的装配。）

## 4.2 角色的频繁转换

（我是弱电工、是硬件工程师、是解说员、是业务员、还是……）

## 4.3 现场掠影

（安装现场效果大赏。）

## 4.4 用户的诉求

（前面自认为考虑的足够周面？用户还是提出了很多问题。）

## 4.5 运营分析

（可耻的私心，我通过我的私人服务器拿到了一手运营数据。我需要盈利模式进行亲自分析。）

## 4.6 奇怪的用户行为

（千奇百怪的用户行为，有人用充电柜作储物柜、手机充电柜，更有甚者用它来烧开水。）

## 5、智能充电柜的资金瓶颈

（在钱烧到 250 个 W 的时候，融资无望的我们，无助的我们。这些付出又算什么？）

## 第十四章 《我的无人智能便利店项目纪实》

（在 ABC 遇到 YY，从此走入新零售行业）

### 1、ABC 背景往事

#### 1.1 入局新零售

（被计算机和网络技术武装到脚的智能便利店。）

#### 1.2 新兴冲浪者

（颠覆传统便利店，主推无人便利店。这背后是雄厚的技术实力，无人收银、无人盘点、AI 等等。）

#### 1.3 创业黑历史

（创业面前人人平等，该趟的坑一样都不会少。不同点在于筹码的多少，少则就此没落；多则东山再起。）

### 2、初代无人店

#### 2.1 无人收银台

##### 2.1.1 基于 RFID 的商品识别

##### 2.1.2 基于 RFID 的无人收银台

##### 2.1.3 RFID 的弊端

##### 2.1.4 基于 RFID 的安全门

1、如何判断进门还是出门？

2、如何检测顾客身上有未结算的商品？

（无人收银与防盗损是攻克的两座大山。其根本是与人性斗争，一切科技在人面前都是徒劳的。）

### 3、第二代无人店

#### 3.1 上海无人零售展大赏

##### 3.1.1 总体介绍

1、无人值守零售终端：

2、无人值守零售技术及产品：

##### 3.1.2 掠影与介绍

- 1、智享易店
- 2、缤果盒子 (BingoBox)
- 3、苏猫
- 4、云拿无人店

(这个展览代表了无人便利店的最前沿技术,如基于深度学习的商品识别、基于机器视觉的顾客行为分析等等。有必要看一下,尤其是缤果盒子和云拿无人店。)

### 3.2 我们的新方案

#### 3.2.1 AM (声磁技术)

(振南告诉你什么是声磁。日日用它不识它,不明就里睁眼瞎。)

#### 3.2.2 新方案无人收银台

(来看看振南的新无人收银台,融合了图像识别、称重校验、声磁技术等等。)

#### 3.2.3 商品的训练与录入

##### 1、商品的训练

##### 2、称重录入

(基于深入学习的商品识别,需要数以万计的样本训练,来看看我们是如何在 15 天内完成几万张商品外包装的拍摄和训练的。最终识别率达到 97.5%以上。)

## 第十五章 《研发速度的天花板,我摸到了!》

(回答一个问题:程序员与架构师有什么区别?我提出的硬件技术架构让研发效率开挂。)

### 1、研发低效的症候

(作为大硬件部门主管,我面临最大问题是如何提高硬件研发效率。)

#### 1.1XYZ 公司始末

(独家的技术成就行业应用。)

#### 1.2 难得的机遇

(千万大单推动公司扩张,资本青睐助力价值实现。)

#### 1.3 研发速度瓶颈

(硬件研发难道就注定周期漫长吗?我要改变这一设定,不作拖累公司的猪队友。)

### 2、关于资源复用

#### 2.1 硬件复用

#### 2.2 嵌入式软件复用

##### 2.2.1 函数复用

##### 2.2.2 模块复用

##### 2.2.3 方案复用

##### 2.2.4 二进制复用

(提高研发效率的根本在于提高技术资源复用度。)

### 3、硬件矩阵架构

#### 3.1 产品的维度

(一个不懂产品的研发是作不好研发工作的,更别说提高研发效率。)

#### 3.2 矩阵架构的思想

##### 3.2.1 硬件模块化

##### 3.2.2 嵌入式模块化

(振南提出了硬件矩阵架构,其核心思想是完完全全的彻底模块化。这里所说的模块化,是你未曾见过的。)

## 4、Microbus 嵌入式架构

### 4.1 动态加载的实现

#### 4.1.1 RT-Thread 的 dlmodule

#### 4.1.2 对 dlmodule 的改良

### 4.2 模块的自动注册（消息分发的实现）

（在嵌入式软件上，我提出了 Microbus 微总线架构，它基于消息驱动与动态加载，实现绝对的并行开发，这应该是团队协作的最高水平。）

## 5、Microbus 比你想得更巧妙

### 5.1 消息的同步问题

### 5.2 基于消息的数据传递

### 5.3 自测试

### 5.4 架构总览

（架构师的工作就是构建技术框架、制造工具与实现机制，它决定了工程师的开发模式和效率。在原始而低效的架构下，你所追求的高效就是奢望；而在优越而高效的架构下，你想低效也是一种奢望。）

## 6、硬件矩阵架构应用实例

### 6.1 Q 公司管线监测项目

（20 天，仅 20 天就拿下了这个价值 30 万的项目。这就是硬件矩阵与 Microbus 架构的优越之处。这个项目也很有意思，用水管上的振动来判断泄露点位置。）

### 6.2 某省地质灾害监测项目

（2000 个监测点位、只有自研才有利润空间，但是我们只有半个月，再一次顶住压力，拿出产品。）

### 《振南共勉》

技术无止境，愿君肯登攀。世上无愚人，只是心未尽。  
心志始无竭，终有出头日。定信念，勿相忘，作出骇绩警世人。

==

## 第十六章 《我与郭天祥的那些事儿》

（郭天祥之于我，是合作伙伴、源动力、榜样、假想敌……）

### 1、我与郭天祥的结识

#### 1.1 入圈单片机

（我是软件出身，最终却搞了单片机。来看看杜撰的蛇形机器人，他还上过《小崔说事》。）

#### 1.2 工训中心偶遇郭天祥

（一切说是偶然，但似乎又是必然。机会可能真得只留给有准备的人。大二我获得了自由之身。）

#### 1.3 电子大赛“金三角”

（杜撰、郭天祥和我说称“金三角”，最后却并没有好的结果。）

### 2、我与郭天祥的项目经历

#### 2.1 火炮油量计算器

（振南给你讲讲火炮“驻退机”的原理，还有你信吗？俄罗斯无人机使用了 STC 单片机。）

#### 2.2 电池活化仪项目

（振南讲一讲如何修好老化的蓄电池。记住：金钱是成功的副产品。）

### 2.3 某油田监测项目

（与郭天祥合作的最后一个项目。郭天祥是一个豁达的人，但是继续合作应该不太可能了。）

## 3、我与郭天祥的后期往来

### 3.1 开发板模式的尝试

#### 3.1.1 初涉开发板

（一个深刻的认识：不要给“门内汉”作教程，盈利的根本是抓住根本供需矛盾。）

#### 3.1.2 贴近目标人群

（不要在你的目标人群面前卖弄技巧。）

#### 3.1.3 封山之作

（我的 ZN-X 模块化多元开发板，是我开发板产品的最终形态。感兴趣？进来看看。）

### 3.2 洗清我所知道的那两个负面事件

#### 3.2.1 “XGB 事件”子虚乌有

#### 3.2.2 “SHT 事件”不存在的

（不是所有人都乐于仰望你，在他们眼里，你只不过是一根绳子。）

### 3.3 不可否认的领袖

#### 3.2.1 近些年的关注

#### 3.2.2 疫情下的重新选择

（每个人都有自己的道路，你也不需要像我们一样抛头露面，这不是你的路。）

### 《寄·郭天祥》

双耳不闻诸非议，洁身自有赞声来。矢志不渝踏前路，得取奇迹平众声。

记住，我们本来就很优秀，我们走过的每一步，都是别人所不曾走过的。

## 第十七章 《那些欲哭无泪的奇葩外包项目》

（接项目，需求就像盒子里的巧克力，你永远不知道下一颗是白巧克力，还是老鼠屎。）

### 1、项目承接与需求落定

#### 1.1 项目承接

（信号延时器，看似简单的需求。）

#### 1.2 需求落定

（这么“简单”的需求，花了几天才最终确定下来。）

#### 1.3 刨根问底

（进一步追问之后，需求又变了。）

#### 1.4 变本加厉

（又双叒改需求。）

### 2、项目动工了

#### 2.1 基本功能的实现

#### 2.2 老鼠屎的显露

（竟然不知道什么是开漏输出。）

### 3、讲些技术干货

#### 3.1 良好的设计模式

#### 3.2 模块化编程

（前后台的设计模式。模块化编程能让我们快速构建复杂工程。）

#### 4、项目后期的继续扯淡

##### 4.1 突破认知底线

##### 4.2 忍无可忍

##### 4.3 交付项目

##### 4.4 放开傻子让我来

（佛系的面对一切。初期的白纸黑字起了大作用。）

#### 5、想破解我的程序，没门

##### 5.1 关于芯片破解

##### 5.2 芯片终极破解方法--ROM 染色

##### 5.3 基于唯一 ID 的保密机制

（世界上没有无法破解的加密。）

当我把这些所谓“屎一样的外包项目”说给别人听的时候，很多人的反应是：“我们应该感谢他们，让我们知道了市场险恶。林子大了什么鸟都有，世界大了什么样的傻子都有。我们只能奉劝他们，药别停！”

### 第十八章 《颠覆认知的“工业之眼”--毫米波雷达》

（毫米波雷达，一个十分神奇的东西。）

#### 1、初探毫米波雷达的神奇

##### 1.1 那场突破认知的交流会

###### 1.1.1 桥梁纵向位移测量

###### 1.1.2 钢索拉力测量

###### 1、拉力传感器直接测量

###### 2、振动法测拉力

###### 3、毫米波雷达测拉力

（通过采集振动信号还能计算钢索的拉力？振南来给你讲讲。毫米波雷达，隔空测位移和拉力，有点神奇哦。）

##### 1.2 令人咂舌的测量精度

（精度一出，一片哗然。它轻松超越现有的传感器。）

#### 2、毫米波雷达的背景

##### 2.1 毫米波雷达的发展历史

###### 2.1.1 早期（雷达诞生的故事）

（很少有人会告诉你，雷达最初是如何出现的？都怪那艘无故闯入马可尼实验的渔船。）

###### 2.1.2 商业化迸发的苗头

（时间回到一战二战，振南给你讲讲毫米波雷达是如何发展起来的。）

###### 2.1.3 毫米波雷达的爆发

（1990 年的德国奔驰的率先应用，直到 2012 年自动驾驶的迅速兴起，彻底燃爆了。）

##### 2.2 头部玩家

（全球前四大的毫米波雷达供应商“ABCD”，来看看都是谁。）

###### 2.2.1 隔空科技

###### 2.2.2 矽典微

（毫米波是万金油，国内相关产业正在迅速崛起。）

##### 2.3 最前沿--雷达成像

（振南来讲讲最先进的 MIMO 技术，还有 4D 成像。它们都是高阶自动驾驶的重要基础。）

### 3、毫米波雷达原理真巧妙

#### 3.1 测量距离

##### 3.1.1 测距原理

- 1) FMCW (调频连续波)
- 2) IF (中频信号)
- 3) 实战测距 (看看实际效果)

(这应该是关于毫米波雷达最为通俗的讲解了, 没有之一。)

##### 3.1.2 测距的主要指标

- 1) 测距范围
- 2) 距离分辨率

(振南给你讲讲毫米波雷达最远能测多远, 还两个物体至少间隔多远才能被分辨出来。你还能看到欧洲最先进的 300GHz 毫米波雷达。)

### 4、十步以外便知风吹草动

#### 4.1 速度测量

#### 4.2 微位移检测

(在这里振南将向你揭开毫米波雷达 0.01mm 超高精度的秘密。有实际的演示哦, 我们在 10 米之外复原了微小振动的波形。)

### 5、作出自己的 MMWR 产品

#### 5.1 说一下 SiliconRadar

(它是欧洲一家专作毫米波雷达的公司, 标志着这一领域最先进的技术, 但就是有些牛气。)

#### 5.2 与德国 SiliconRadar 的论战

(被鄙视, 最敲竹杠, 说我们作不出来。就不信邪, 我们就要自研毫米波雷达。)

#### 5.3 自研毫米波雷达

(这里有些资料是保密的, 但我还是拿出来给大家。)

## 第十九章《用单片机实现数码相框/视频播放器/相机录像机》

(开篇振南给出了自己独有的学习方法: 用浅层认知推动学习, 并转化为工程经验。)

### 1、一切源于“烂苹果”

(那些曾经红极一时的论坛, 现在为什么都没落了?)

#### 1.1 什么是“烂苹果”

(看看“烂苹果”都给我们讲了一个什么故事?)

#### 1.2 有屏幕就有“烂苹果”

(在 OLED 上、TFT 大屏上播放“烂苹果”, 甚至在示波器上、MATLAB 上、显微镜下, Oh, My God!)

#### 1.3 实现我们自己的“烂苹果”

##### 1.3.1 从图片显示开始

##### 1.3.2 搞定视频播放

- 1) 读取 SD 卡 (振南的 znFAT 文件系统)
- 2) 关于振南的 ZNV 视频格式
- 3) 关于振南的 AVI-ZNV 转换软件
- 4) 视频播放的实现

(在振南的 ZN-X 开发板上实现了视频播放。)

#### 1.4 单片机实现 AVI 视频播放

##### 1.4.1 先说一下 AVI 视频格式

#### **1.4.2 实现 AVI 视频播放**

#### **1.4.3 AVI 视频音频同步播放**

（AVI 视频其实很简单，我们用 STM32F4 来直接播放。会介绍一些视频处理利器软件。）

### **2、单片机实现数码相机录像机**

#### **2.1 拍照功能的实现**

#### **2.2 数码相机录像机的实现**

（在振南的 ZN-X 上实现直接录像存为 AVI 视频文件，并在电脑上播放。）

**【未完待续】** 11 与 20 章