

# 任军

嵌入式软件工程师

185 7164 0900

[819280802@qq.com](mailto:819280802@qq.com)

Personal Blog: [cedar-renjun.github.io](http://cedar-renjun.github.io)

## 教育背景

郑州大学 本科 电气自动化（弱电方向） 08 级  
CET4 读写优秀，听说良好

## 技术能力

扎实的电子技术基础  
扎实的 C/C++ 基础，熟悉 ARM 汇编语言  
丰富的 51，AVR，ARM Cortex-M 平台编程经验  
熟悉常见的嵌入式 RTOS 并具有应用软件编程经验  
熟悉嵌入式常见通信接口并有丰富驱动编程经验  
掌握嵌入式系统中常用设计模式和数据结构

## 工作经验

2012-07 至今	Premier Farnell 集团---CooCox 嵌入式开发工具业务线	嵌入式研发工程师
<ol style="list-style-type: none"><li>负责编写 ARM Cortex-M 系列芯片 Cox 接口库，优化测试架构以提高测试效率，同时优化和修订 Cox 规范，有效提高 Cox 易用性</li><li>负责编写基于通用接口的驱动，并编写驱动接口规范，极大简化驱动使用难度，同时提高驱动移植性</li><li>辅助 CoIDE 设计，为 CoIDE 增加 C++ 支持和制定文档生成规范</li><li>参与制定开源项目推广策略和 CooCox 产品战略规划</li></ol>		

## 项目经验

2012-01 至 2012-06	完成《基于 STM32 的触摸屏 MP3》毕业设计
2011-07 至 2011-09	中国机器人大赛暨 ROBOCUP 中国公开赛仿人组标准组，担任队长负责核心技术和团队管理，并取得 <b>二等奖</b>
2010-10 至 2011-04	在 <b>大学生创新实验项目</b> 《基于 mega128 的超声波路况检测系统》中作为 <b>技术负责人</b> 负责项目软件架构
2010-09 至 2010-11	在广茂达杯武术擂台赛标准组比赛中，担任队长负责软件算法，并取得 <b>三等奖</b>
2010-06 至 2010-08	在 iCan 传感器大赛中，作为《基于 MEMS（微机电系统）三轴加速度传感器的智能小车》 <b>技术负责人</b> ，编写下位机软件和多机通信

## 个人特点

技术方面：  
爱好分享，爱好开源  
技术完美主义，对一切好玩的东西保持好奇心

做事习惯：  
Do one thing and do it well

团队建设：  
胜则举杯相庆，败则拼死相救

## 【附】

# 工作/项目详细介绍

任军 嵌入式软件工程师

185 7164 0900

[819280802@qq.com](mailto:819280802@qq.com)

Personal Blog: [cedar-renjun.github.io](http://cedar-renjun.github.io)

## 工作经历

---

✧ 2012-07 至今 英蓓特科技有限公司--CooCox 嵌入式开发工具线 嵌入式软件研发

## 在校经历

---

✧ 2012-01 至 2012-06 毕业设计：基于 STM32 的触摸屏 MP3 设计 项目负责人

✧ 2011-07 至 2011-09 中国机器人大赛暨 ROBOCUP 中国公开赛仿人组标准平台 技术负责人

✧ 2010-10 至 2011-04 创新实验项目：基于 mega128 的超声波路况检测系统 组长+技术负责人

✧ 2010-09 至 2010-11 广茂达杯国际灭火机器人比赛 组长+代码编写

✧ 2010-06 至 2010-08 基于 MEMS（微机电系统）三轴加速度传感器的智能小车 组长+技术负责人

## ✧ 工作经历 1: 英蓓特科技有限公司--CooCox 开发工具链

嵌入式软件研发工程师

起止时间: 2012-07 至今

软件环境: windows7, IAR, MDK, CoIDE (Gcc), VS2008, 汇编, C/C++

硬件环境:

1. ST/NXP/Freescale/TI/Nuvoton Cortex-M0~4 系列开发板
2. 各种开源硬件商提供的模块 (Seeed Studio/DFRobot/Arduino/Mbed)

任务职能:

**1. 编写 Cortex-M 系列 Cox 库, 并改进 Cox 接口规范**

各个半导体厂商在推出 Cortex-M 库时, 为了加速应用者开发速度, 都会推出一套对应的芯片支持库, ARM 仅仅对内核部分做了统一化处理 (CMSIS 接口), 但外设 (如 UART/PWM/DMA 等) 接口尚无统一标准, 这使得基于 Cortex-M 芯片开发的软件, 没有良好的移植性。所以, CooCox 在标准厂商库的基础上, 提炼出一套外设 HAL 接口标准, 这就是 Cox 驱动接口层。基于 Cox 接口编写的软件, 可以平滑的在各个 Cortex-M MCU 平台之间切换

我刚进公司时, 负责编写基于 STM32 芯片的 Cox 代码 (部分功能模块); 前期的 Cox 库注释标准和接口不清晰, 我独立整理了 Doxygen 注释规范, 并独自完成一套 NXP 17xx M3 芯片 Cox 库, 同时更新了 Cox 测试框架。这些工作, 有效的提高的 Cox 易读性, 易用性

参考链接:

- |                        |                                                                                                       |
|------------------------|-------------------------------------------------------------------------------------------------------|
| 1. Cox 的介绍资料           | <a href="http://www.coocox.org/cox.html">http://www.coocox.org/cox.html</a>                           |
| 2. Cox 的 github 地址     | <a href="https://github.com/coocox/cox">https://github.com/coocox/cox</a>                             |
| 3. LPC17xx 的 github 地址 | <a href="https://github.com/cedar-renjun/Cox_LPC17xx">https://github.com/cedar-renjun/Cox_LPC17xx</a> |

**2. 整理并编写市面上的各种硬件驱动**

CooCox 拥有完整的嵌入式开发工具链, 包括新一代网络化的 CoIDE, 组件化的驱动平台; 在 CooCox 这个驱动平台上, 任何人都可以上传自己的代码 (基础库, 中间件, 示例性代码), 其它人可以免费使用。CooCox 试图打造一个嵌入式开发者共享交流平台, 我和组件小组其它 3 个人, 负责收集市面上的硬件驱动模块/中间件, 并编写基于 Cox 的驱动代码。开始时, 驱动不好移植, 我着手制定了一个驱动接口规范来解决这个问题; 同时调研 Mbed 接口驱动支持架构, 准备让 CoIDE 支持 Mbed 接口标准 (规划 ing, 待做)

参考资料:

- |                   |                                                                                                                 |
|-------------------|-----------------------------------------------------------------------------------------------------------------|
| 1. CooCox 驱动列表    | <a href="http://www.coocox.com/driver.html">http://www.coocox.com/driver.html</a>                               |
| 2. 个人 github 仓库地址 | <a href="https://github.com/cedar-renjun?tab=repositories">https://github.com/cedar-renjun?tab=repositories</a> |

**3. 配合和协调 CoIDE**

组件这里配合 Java 组, 为 CoIDE 增加新功能和制定规范, 具体包括:

**3.1 为 CoIDE 增加 C++ 支持**

CoIDE 1.7.5 之前的版本不支持 C++, 为此 Coocox 的很多国外用户不断抱怨, 后来我负责调研 gcc 支持基本原理, 编写 GCC 链接脚本, 配合上层, 为 CoIDE 增加 C++ 支持 (1.7.6 版本中已支持 C++)

**3.2 编写 CoIDE 文档规范**

现在 CoIDE 改变进行中, 新的 UI 表现层弃用原有 Java 引擎, 改用新的 Web 内核, 所有界面采用 HTML5 来编写, 同时引入 github 的支持, 增加公有仓库和私有仓库功能; 我负责组件 (指所有模块化

的嵌入式代码) 的文档规范和 CoIDE 交互行为需求, 主要包括: Markdown 和 Doxygen 交互接口规范, 组件文档组织规范, 修订代码编写规范

#### 4. 参与制定市场推广策略和产品战略规划

做的不是具体工作, 主要是战略层的考虑, 制定决策 (涉及公司机密, 细节不便透露), 可以细分为两大部分:

##### 4.1 CooCox 产品战略

分析 CooCox 竞争对手的优缺点, 进行 SWTO 分析;

时刻注意竞争对手的动作, 并考虑可能对我们造成的威胁, 适时制定决策加以应对

##### 4.2 产品推广策略

细分顾客目标群, 并分析 CooCox 产品顾客行为特征, 针对不同用户制定不同的推广策略;  
制定策略吸引开发者来参与开源项目

## ◇ 项目 5: 毕业设计: 基于 STM32 的触摸屏 MP3 设计

项目时间: 2012-01 至 2012-06

软件环境: windows7, IAR

硬件环境: STM32 103VET6, VS1003

开发工具: IAR, JlinkV8

项目描述:

=====设计目标=====

设计一款电阻触摸屏的 MP3 播放器, 该 MP3 可以自动读取 SD 卡上的 MP3, wav 音频文件, 搜索完存储卡后, 自动提供播放列表供使用者查阅, 使用者可以选中某一首歌曲, 然后点击播放按钮开始播放歌曲, 在播放过程中, 可以停止播放歌曲, 可以上一曲, 下一曲播放, 同时该播放器提供音量增减功能

=====技术分析=====

- 1: 本项目的核心芯片为 STM32103VT6, 它采用 ARM 最新设计的 cortex-m3 内核, 该软核指令执行效率很高, 同时内嵌 NVIC (嵌套向量中断控制器), SysTick, 调试组件, 极大的方便了使用者的移植工作。同时 STM32 内部模块丰富, 功能强大, 且 ST 公司提供标准库函数, 方便了移植和用户使用。
- 2: 采用 ucos-ii 操作系统, 操作系统的引入, 多任务并发执行, 提高了 cpu 的利用效率, 简化编程
- 3: 由于项目中采用 TFT 液晶屏, 加上触摸屏, 直接操作 TFT 非常复杂, 所以引入 UCGUI, 便于用户使用, 同时可以创建美观大方的用户界面
- 4: 由于 MP3 的体积比较大, 所以需要外部存储器, 采用 SD 卡比较方便, 需要在 SD 上建立文件系统, 采用嵌入式文件系统 FatFS, 该文件系统结构精简, 便于移植, 非常适合于嵌入式系统
- 5: 采用 VS1003 解码芯片用解码 mp3 文件, 处理器不需要软解码 mp3, 只需要将 mp3 数据通过 SPI 接口发送给 VS1003 模块即可

项目职责:

本作品是毕业设计作品, 本人作为整个项目负责人

## ✧ 项目 4: 中国机器人大赛暨 ROBOCUP 中国公开赛: 仿人组标准平台

**项目时间:** 2011-07 至 2011-09

**软件环境:** windows7, C 语言, 汇编语言

**硬件环境:** 机器人大赛指定开发套件 (高级版)

**开发工具:** IAR

**项目描述:**

=====竞赛要求=====

利用指定的机器人套件, 设计一款人形的机器人, 该机器人具有明显的人体结构, 如头, 手臂, 腰部, 且能自主上台, 并在台上自由行进, 不掉下台子。比赛场地大小为长、宽分别为是 2400 mm, 高 150mm 的正方形矮台, 台上表面即为擂台场地。底色从外侧四角到中心分别为纯黑到纯白渐变的灰度。场地的两个角落设有坡道, 机器人从出发区启动后, 沿着该坡道走上擂台。在指定的擂台上双方机器人和 5 个中国象棋棋子。双方机器人模拟中国传统擂台搏击的规则, 互相击打或者推挤。如果一方机器人整体离开擂台区域或者不能再继续行动或被对方机器人打倒, 则另一方获胜。

=====技术分析=====

### 1: 硬件平台

采用博创机器人高级套件, 核心控制板为 AVR mega128, 机器人形状为人型, 底盘为轮型

### 2: 手臂和腰部

手臂和腰部采用舵机组成, 核心控制板通过 UART 半双工串口控制总线式舵机, 通过预设定的动作来产生击打动作

### 3: 运动控制

采用直流减速电机, 驱动卡为博创的控制卡, 采用半双工总线控制

### 4: 地面场地检测

地面场地是从中心往外灰度递减的, 故采用灰度传感器来检测地面灰度值, 从而确定机器人在场地中的位置

### 5: 场地边缘的识别

采用固定距离值的红外传感器来检测边缘, 该传感器为数字接口, 当检测到边缘时输出高电平, 该接口直接连接至 mcu 的中断线, 从而紧急后退

### 6: 敌人的识别

敌人识别采用红外距离传感器, 其接口采用模拟接口, 障碍物与机器人本体的距离与输出的模拟量成正比, 通过 ADC 模块, 通过软件插值法可确定实际距离值

**项目职责:**

本人作为组长, 负责人员的协调和项目代码的编写工作

### 1: 作为组长, 负责培训新人和协调整个团队

负责培训队里的新人学习 C 语言和相关机器人模块, 负责分配整个队伍的任务安排

### 2: 负责整体代码的编写工作

该项目采用 ucos-ii 实时多任务环境, 负责 ucos-ii 的操作系统的移植和应用代码的编写, 以及底层驱动的部分, 如 R/C 舵机控制, ADC, GPIO 等

负责上层比赛策略的实现

### ◇ 项目 3: 创新实验项目: 基于 mega128 的超声波路况检测系统

**项目时间:** 2010-10 至 2011-04  
**软件环境:** windows7, C 语言, VB 编程语言  
**硬件环境:** mega128 开发板, 超声波模块  
**开发工具:** Winavr, VS2008  
**项目描述:**

=====设计要求=====

设计一个系统,该系统能实时检测道路车况信息(如车流量,车速等)并将采集到的数据通过无线模块发送到数据中心,数据中心将采集到的数据图形化显示出来,以便供高层决策者分析

=====技术分析=====

#### 1:整体构思

本项目为郑州大学创新实验项目,重点在于让学生通过该项目,培养工程意识,提高动手能力.且若做成实际使用的系统,需要在道路上实际安装调试,但此时存在安装调试不便,且不安全,所以经过讨论,整个实验项目采用模拟方式进行,按比例制作出实物模型,通过测试实物来达到预期目的

#### 2:芯片选择

可选芯片有 51 和 AVR, 因 AVR 系列自带定时器具具有输入捕捉功能, 便于测量超声波模块的输入脉冲, 且内部定时器资源丰富, 主频比较高, 相比而言, 若采用 51 系列处理器, 性价比不高, 且编程复杂难调试. 故最后选择 AVR 系列的 mega128

#### 3: 测量模块

本项目要求同时测量道路车速和车辆数目等信息, 故至少需要两个超声波探头, 且为了避免探头相互干扰, 模块的安装间距需要符合要求. 同时为了不影响车辆的正常行驶, 探头需要安装在道路的正上方.

#### 4: 数据中心的模拟

编写个上位机来模拟数据中心, 该上位机采用 VB 编写, 通过 comm 控件来获取串口信息, 同时将采集到的数据用图形化的方式表现在屏幕上

### 项目职责:

本人在项目中负责下位机代码及论文的撰写工作

#### 1:下位机部分

下面简述下位机代码思路

本系统采用成品超声波模块, 该模块接口为 VCC, GND, ECHO, TRIG. 使用时先在 TRIG 端口施加一个高电平触发脉冲 (大于 10us), 然后 ECHO 会输出一个高电平脉冲, 该脉冲宽度正比于障碍物到探头的距离, 使用 mega128 定时器的输入捕捉模式可以测量该脉冲宽度. 设定一个距离值 H\_Lim, 当检测到距离值小于 H\_Lim 时, 可以断定检测到车辆当前面的探头检测到车辆后, 车辆计数器增 1, 同时记录此时的时间值, 当后面的车辆检测到车辆后, 记录此时的时间值, 根据这两次的时间差和探头的间距值, 可以测算车辆的通过速度. 得到车辆速度和车辆数目这些路况信息后, 通过 mega128 的串口发送到模拟数据中心

#### 2:论文撰写

根据本项目的情况, 撰写论文《基于超声波传感器的车流量检测系统》, 并发表在核心期刊《实验室研究与创新》(2011-8, 72-74 页)

## ✧ 项目 2: 广茂达杯国际灭火机器人比赛

项目时间: 2010-09 至 2010-11  
软件环境: windows7, C 语言  
硬件环境: 广茂达专用灭火机器人套件  
开发工具: MF09(灭火机器人专用开发工具)  
项目描述:

=====竞赛要求=====

在模拟房间中,放置一个蜡烛,设计个灭火机器人从出发点开始,自主搜寻每个房间并检测是否起火(用点燃的蜡烛模拟),机器人找到灭火源后进入房间,并驶向火源,走到距火源一定距离时,停止行进,开始灭火,扑灭火源后,自动返回出发点并停止在出发点处

=====技术分析=====

- 1:机器人硬件平台采用比赛方指定套件,核心处理器为 300M 的高速 DSP
- 2:火源的识别采用热红外传感器,该传感器输出量为模拟值,通过 DSP 的 ADC 接口转换为数字量,在软件中设定火源检测阈值,大于阈值则认为发现火源
- 3:采用夏普红外距离传感器检测墙壁和房间入口,该传感器为模拟接口,距离值与输出模拟量近似成线性关系,通过检测模拟量,即可获取机器人与墙壁距离信息
- 4:出发点和房间门口均为白色,其它地面为黑色,故采用灰度传感器来检测黑白地面
- 5:为便于搜寻房间和扑灭火焰,故设计成车型机器人,动力装置采用直流减速电机,无速度反馈装置,驱动方式为 PWM 控制
- 6:因场地限制无法采用灭火器和 水来扑灭火焰,故采用高速电风扇吹风方式扑灭火源

### 项目职责:

本次比赛为组队参加,三人一组,我作为队长负责整体安排和项目代码编写  
具体所做工作如下:

- 1: 作为队长负责整体安排  
负责整体项目进度,分阶段分配队员任务
- 2: 负责机器人代码编写工作  
由于比赛方提供机器人套件底层驱动接口,所以我们只需要编写机器人搜索房间策略和规划路径策略由我们组三个人共同讨论制定,我负责编码实现,并根据测试情况及时修改代码



## ◇ 项目 1：美新杯国际物联网大赛：基于 MEMS（微机电系统）三轴加速度传感器的智能小车

项目时间： 2010-06 至 2010-08

软件环境： windows7，C 语言

硬件环境： AT89S52，无线模块 NRF24L01，液晶模块，直流电机驱动器，超声波模块，三轴加速度传感器

开发工具： Keil

项目描述：

=====设计目标=====

设计一个智能小车，本车具备无人操作和人工控制两种模式

自动控制模式：智能小车自主漫游，主动躲避障碍物

人工控制模式：操作者利用装有三轴加速度传感器的遥控器，通过感知遥控器和地面的倾角来控制小车的运动

同时该车具备液晶显示屏，可以实时的显示一些控制信息：如是否有障碍物，车速，当前控制模式等

=====技术分析=====

- 1：车体和遥控器采用 AT89S52 单片机（因为当时大二，只会这一种单片机）
- 2：考虑到需要远程控制,故采用 nordic 公司 NRF24L01 模块，该芯片采用 SPI 接口和单片机通信
- 3：AT89S52 不具备接口,需 GPIO 模拟接口与三轴加速度传感器(MXC6202xG)自带接口通信
- 4：采用飞思卡尔 mc33886 电机芯片来控制直流电机，采用 PWM 模式控制，频率小于 10KHz
- 5：因车体部分含有液晶，超声波，直流电机驱动卡，无线等模块，单 MCU 控制方案会导致该 MCU 负担过重，所以采用双 MCU 控制方案，一个 MCU 负责和 NRF24L01 交互，负责液晶显示，另外一个负责控制 RC 舵机，直流减速电机，超声波模块

项目职责：

作为队长，负责整体项目进度安排和代码编写工作

具体所做工作如下：

- 1：根据组员的自身特长，合理分配任务，如：
  - 某位组员的口才比较好，让他负责作品的宣传工作
  - 某位组员熟悉底层，就让他参与硬件驱动编写
- 2：编写部分底层的硬件驱动程序
  - 2-1 负责的底层模块有：
    - 三轴加速度传感器，直流减速电机驱动器，NRF24L01 无线模块，R/C 舵机驱动
  - 2-2 负责设计底层通信协议：
    - 车体上两个 MCU 通过 UART 通信，负责制定数据传输协议
- 3：负责上层应用软件的开发
  - 3-1：调用 LCD 驱动函数，在车载 LCD 液晶屏实时显示车辆状态信息
  - 3-2：根据采集到的三轴加速度数据，来判断遥控器相对于地面的倾角信息，然后控制车体运动