

# 嵌入式大雜燴週記| 第1 期

原創 ZhengN 嵌入式大雜燴 2022-02-07 21:30

收錄於話題

#嵌入式大雜燴週記 1 #嵌入式 156

大家好，我是ZhengN。

嵌入式大雜燴週記主要是一些實用項目學習分享，每週一篇，每篇一個主題。

內容主要來源於我們之前收集的資料：

<https://gitee.com/zhengnianli/EmbedSummary>

## 本期主角：gear-lib

gear-lib是一組通用的C基礎庫。

- 全部用POSIX C實現，目標是為了跨平台兼容linux, windows, android, ios。
- 適用於物聯網，嵌入式，以及網絡服務開發等場景。

gear-lib地址：

<https://github.com/gozfree/gear-lib>或者

<https://gitee.com/gozfree/gear-lib>

license：MIT。

關於開源軟件協議相關文章：[常用的開源協議有哪些？](#)

## gear-lib的使用

gear-lib整理涵蓋如下內容：



下載得到：



下面基於Ubuntu環境，整體編譯、安裝：

```
./build.sh  
sudo ./build.sh install
```

下面我們選取幾個庫實踐感受一下（庫的文件夾下面有對應的使用示例，可參考這些示例）。

## 1、libconfig

libconfig支持ini、json、lua三种格式配置文件的解析，并且统一操作接口，即可以调用同一套接口对ini、json、lua三种格式配置文件进行解析。libconfig会自动识别配置文件后缀调用对应格式文件的操作接口。

### (1) 解析ini文件：

```
static int ini_test(void)
{
    struct config *conf = conf_load("ini/ini_test.ini");
    printf("=====ini_test=====\n");
    printf("修改之前ini_test.ini的内容： \n");
    printf("name = %s\n", conf_get_string(conf, "test:name"));
    printf("num = %d\n", conf_get_int(conf, "test:num"));
    conf_set_string(conf, "test:name", "LinuxZn");
    conf_set_string(conf, "test:num", "888");
    conf_dump(conf);
    printf("修改之后ini_test.ini的内容： \n");
    printf("name = %s\n", conf_get_string(conf, "test:name"));
    printf("num = %d\n", conf_get_int(conf, "test:num"));
    conf_save(conf);
    conf_unload(conf);

    return 0;
}
```

编译、运行：

ini文件解析的往期文章：[嵌入式开发小记，实用小经验分享](#)

( 2 ) 解析json文件：

```
static int json_test(void)
{
    struct config *conf = conf_load("json/json_test.json");
    printf("=====json_test=====\n");
    printf("修改之前json_test.json的内容： \n");
    printf("name = %s\n", conf_get_string(conf, "test", "abc", 1, "name"));
    printf("num= %s\n", conf_get_string(conf, "test", "abc", 1, "num"));

    conf_set_string(conf, "test", "abc", 1, "num", "888");
}
```

```
printf("修改之后json_test.json的内容： \n");

printf("name = %s\n", conf_get_string(conf, "test", "abc", 1, "name"));

printf("num= %s\n", conf_get_string(conf, "test", "abc", 1, "num"));

conf_save(conf);
conf_unload(conf);

return 0;
}
```

编译、运行：

json文件解析的往期文章：[例说嵌入式实用知识之JSON数据](#)

## 2、liblog

```
static void test_no_init(void)
{
    int i;
    char tmp[32] = "abcd";
    for (i = 0; i < 10; i++) {
        loge("test rsyslog\n");
        logw("debug msg %d, %s\n", i, tmp);
        logd("debug msg %d, %s\n", i, tmp);
        logi("debug msg %d, %s\n", i, tmp);
        logv("debug msg %d, %s\n", i, tmp);
    }
    log_deinit();
}
```

编译、运行：

log相关往期文章：

[bug解决不了？使用日志法](#)

[嵌入式软件打log的一些心得！](#)

[没有串口，你会如何输出log？](#)

### 3、libdarray

```
static void darray_test(void)
{
    DARRAY(int) darray;

    /* 初始化一个动态数组 */
    da_init(darray);

    /* 给数组赋值 */
    for (int i = 0; i < 10; i++)
    {
        da_push_back(darray, &i);
    }
}
```

```
/* 遍历数组元素并打印 */  
  
for (int i = 0; i < 10; i++)  
{  
    printf("%d ", darray.array[i]);  
}  
printf("\n");  
  
/* 数组清理 */  
da_free(darray);  
}
```

编译、运行：

动态数组相关往期文章：[什么是不完全类型？](#)

## 4、libtime

```
static void time_test(void)  
{  
    printf("time_now_sec_str:    %s", time_now_sec_str());  
    printf("time_now_sec:      %" PRIu64 "\n", time_now_sec());  
    printf("time_now_msec:       %" PRIu64 "\n", time_now_msec());  
    printf("time_now_usec:       %" PRIu64 "\n", time_now_usec(NULL));  
    printf("time_now_nsec:       %" PRIu64 "\n", time_now_nsec());  
}
```

编译、运行：



## 5、libhal

```
static void hal_test(void)
{
    struct cpu_info cpu_info;

    cpu_get_info(&cpu_info);
    printf("cpu name : %s\n", cpu_info.name);
    printf("cpu cores : %d\n", cpu_info.cores);
    printf("cpu cores_available : %d\n", cpu_info.cores_available);
}
```

编译、运行：

## 6、libsort

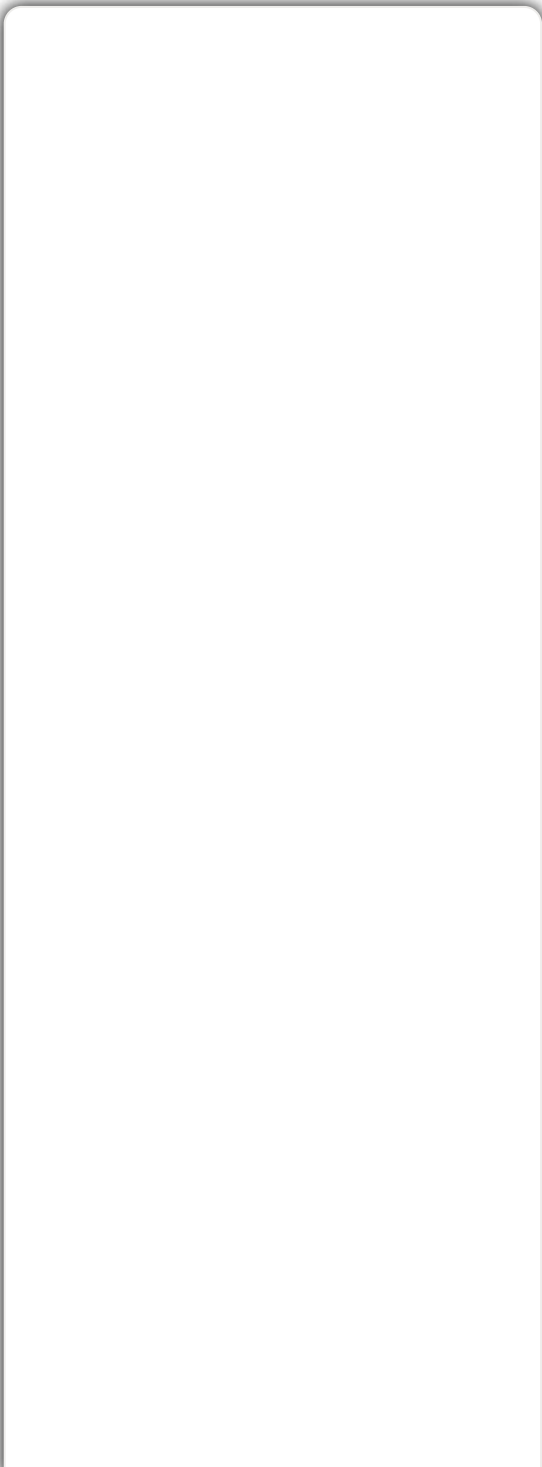
```
void test_bubblesort()
{
    int a[]={4,1,2,5,3};
    print_array(int, "%d\t", a);
    bubble_sort(a, sizeof(a)/sizeof(a[0]), sizeof(int), 0);
    print_array(int, "%d\t", a);
}
```

编译、运行：



以上就是本期分享的内容。gear-lib 包含的库很丰富，覆盖很多方面，大家可以去简单了解一下，看看是否有一些库是自己学习、工作中用得上的，避免重复造轮子。

另外，gear-lib 是码云上 GVP（最有价值开源项目），值得我们去学习。gear-lib 虽然总的代码量也不少，但是各个库都相对独立（尽管有些库之间有依赖）：





这也很方便我们选取一些库来进行代码研读。一起读代码，一起进步~

这就是本期的内容，咱们下期见~

## 往期推荐：

[干货 | 项目乏力？nanopb助你一臂之力](#)

[长文 | 花了两天时间整理了STM32中的一些C语言知识点](#)

[分享一个有趣的库，让你学习C语言不会觉得那么枯燥](#)

[嵌入式 C | 结构体完全笔记，收藏！](#)

[实用 | 一个高性能通信库的简单使用分享](#)

[实用工具 | LVGL GUI-Guider的使用分享](#)

[基于vs2019的lvgl模拟器使用](#)

[lvgl最新版本在STM32上的移植使用](#)

[实用 | 10分钟教你搭建一个嵌入式web服务器](#)

[嵌入式开发小记，实用小知识分享](#)

[分享一款嵌入式人必备绘图工具！](#)

[干货 | .protobuf-c之嵌入式平台使用](#)

[干货 | 嵌入式必备技能之Git的使用](#)

[例说嵌入式实用知识之JSON数据](#)

[C语言、嵌入式中几个非常实用的宏技巧](#)

[一个小巧、开源的信号发生器，酷！](#)





Hello Linux  
11篇原创内容

公众号



嵌入式大杂烩  
本公众号专注于嵌入式技术，包括但不限于C/C++、嵌入式、物联网、Linux等编程学...  
267篇原创内容

公众号

在公众号聊天界面回复**1024**，可获取嵌入式资源；回复 **m**，可查看文章汇总。

点击**阅读原文**，查看更多分享。

收录于话题 #嵌入式 156

上一篇 · 分享一个小巧的嵌入式日志模块（附代码）

阅读原文

喜欢此内容的人还喜欢

八大嵌入式 C 语言的难点揭秘  
硬件攻城狮

嵌入式编程，基于cpost的上下文快速切换，文末附代码链接  
STM32嵌入式开发

嵌入式C语言编程——.h文件与.c文件  
STM32嵌入式开发