



西北工业大学

# 本科毕业设计论文

题 目      基于 GNU Radio 的 DVB-T 传输系统

---

专业名称      电磁场与无线技术

---

学生姓名      姜阳

---

指导教师      李彬

---

完成时间      2017.05.20

---

## 目 录

1 Linux 安装 .....	1
1.1 x86 平台 .....	1
1.1.1 虚拟机 .....	1
1.1.2 双系统 .....	1
1.1.3 安装 .....	1
1.1.4 软件源 .....	1
1.2 ARM 平台—树莓派 .....	1
1.2.1 简介 .....	1
1.2.2 烧录系统镜像 .....	3
1.2.3 软件源 .....	3
1.2.4 无线网络配置 .....	3
1.2.5 镜像备份 .....	3
2 环境配置 .....	4
2.1 CMake .....	4
2.1.1 简介 .....	4
2.1.2 安装 .....	4
2.2 Boost C++ Libraries .....	5
2.2.1 简介 .....	5
2.3 Doxygen .....	5
2.3.1 简介 .....	5
2.4 Python .....	6
2.4.1 简介 .....	6
2.5 SWIG .....	6
2.5.1 简介 .....	6
2.6 USRP .....	7
2.6.1 简介 .....	7
2.7 HackRF .....	7
2.7.1 简介 .....	7
2.7.2 osmosdr .....	8
2.8 gqrx .....	8
2.8.1 简介 .....	8
2.9 git .....	8
2.9.1 简介 .....	8
2.9.2 常用命令 .....	9
2.9.3 版本控制 .....	9
2.9.4 github .....	9

2.10 Docker .....	10
2.10.1 简介.....	10
2.10.2 构建容器.....	11
2.11 GNU Radio .....	11
2.11.1 简介.....	11
2.11.2 版本选择.....	11
3 DVB-T 发射端 .....	13
3.1 能量扩散.....	13
3.2 RS 编码.....	13
3.3 卷积交织.....	13
3.4 卷积编码).....	13
3.5 比特交织.....	13
3.6 符号交织.....	13
3.7 星座映射.....	13
3.8 参考信号.....	13
3.9 IFFT.....	13
3.10 OFDM 循环前缀.....	13
3.11 常数 .....	13
3.12 重采样 .....	13
3.13 USRP 发射 .....	13
参考文献 .....	14

## 第一章 Linux 安装

### 1.1 x86 平台

#### 1.1.1 虚拟机

VMware

Hyper-V

#### 1.1.2 双系统

空间分配

- /
- /swap
- /boot
- /home

LVM

快照

#### 1.1.3 安装

#### 1.1.4 软件源

- 新立得包管理器
- 阿里云

### 1.2 ARM 平台——树莓派

#### 1.2.1 简介

树莓派（英语：Raspberry Pi），是一款基于 Linux 的单板机电脑。它由英国的树莓派基金会所开发，目的是以低价硬件及自由软件促进学校的基本计算机科学教育。

树莓派的生产是通过有生产许可的两家公司：Element 14/Premier Farnell 和 RS Components。这两家公司都在网上出售树莓派。

树莓派配备一枚博通 (Broadcom) 出产的 ARM 架构 700MHz BCM2835 处理器, 256MB 内存 (B 型已升级到 512MB 内存), 使用 SD 卡当作存储媒体, 且拥有一个 Ethernet、两个 USB 接口、以及 HDMI (支持声音输出) 和 RCA 端子输出支持。树莓派只有一张信用卡大小, 体积大概是一个火柴盒大小, 可以运行像《雷神之锤 III 竞技场》的游戏和进行 1080p 视频的播放。操作系统采用开源的 Linux 系统如 Debian、ArchLinux, 自带的 Iceweasel、KOffice 等软件, 能够满足基本的网络浏览、文字处理以及电脑学习的需要。分 A、B 两种型号, 售价分别是 A 型 25 美元、B 型 35 美元。树莓派基金会从 2012 年 2 月 29 日开始接受 B 型的订货。

树莓派基金会提供了基于 ARM 架构的 Debian、Arch Linux 和 Fedora 等的发行版供大众下载, 还计划提供支持 Python 作为主要编程语言, 支持 BBC BASIC、C 语言和 Perl 等编程语言。树莓派基金会于 2016 年 2 月发布了树莓派 3, 较前一代树莓派 2, 树莓派 3 的处理器升级为了 64 位的博通 BCM2837, 并首次加入了 Wi-Fi 无线网络及蓝牙功能, 而售价仍然是 35 美元。[1]

## ● 树莓派 3B+ 参数

SoC	Broadcom BCM2837 (CPU, GPU DSP 和 SDRAM、USB)
CPU	ARM Cortex-A53 64 位 (ARMv8 系列) 1.2GHz (四核心)
GPU	Broadcom VideoCore IV[43], OpenGL ES 2.0, 1080p 30 h.264/MPEG-4 AVC 高清解码器
RAM	1024 MB (LPDDR2)
外设	14 个 GPIO 及 HAT 规格铺设
额定功率	4.0 瓦 (5V/800mA)
电源输入	5V 电压 (通过 MicroUSB 或经 GPIO 输入)
总体尺寸	85.60 × 53.98 毫米 (3.370 × 2.125 英寸)
重量	45 g (1.6 oz)

1.2.2 烧录系统镜像

1.2.3 软件源

1.2.4 无线网络配置

1.2.5 镜像备份

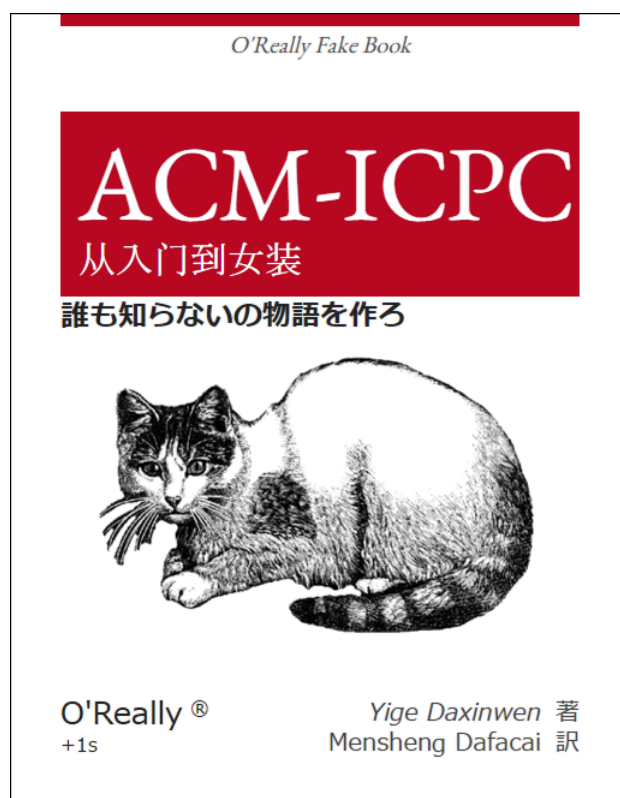


图 1.1 这里是个普通的标题

## 第二章 环境配置

### 2.1 CMake

#### 2.1.1 简介

CMake 是个开源的跨平台自动化建构系统，它用配置文件控制建构过程 (build process) 的方式和 Unix 的 Make 相似，只是 CMake 的配置文件取名为 CMakeLists.txt。Cmake 并不直接建构出最终的软件，而是产生标准的建构档 (如 Unix 的 Makefile 或 Windows Visual C++ 的 projects/workspaces)，然后再依一般的建构方式使用。这使得熟悉某个集成开发环境 (IDE) 的开发者可以用标准的方式建构他的软件，这种可以使用各平台的原生建构系统的能力是 CMake 和 SCons 等其他类似系统的区别之处。CMake 可以编译源代码、制做程序库、产生适配器 (wrapper)、还可以用任意的顺序建构可执行文件。CMake 支持 in-place 建构 (二进档和源代码在同一个目录树中) 和 out-of-place 建构 (二进档在别的目录里)，因此可以很容易从同一个源代码目录树中建构出多个二进档。CMake 也支持静态与动态程序库的建构。[2]

#### 2.1.2 安装

CMake 要求每个目录下均有一个 CMakeLists.txt 文件，一个通用的目录结构如下：

```
.
├── app
│   ├── CMakeLists.txt
│   ├── main.cc
│   └── main.h
├── include
│   ├── CMakeLists.txt
│   └── test.h
├── lib
│   ├── test_impl.cc
│   └── test_impl.h
├── CMakeLists.txt
└── README.md
```

其中：

app目录下放置main函数所在文件，可以同时包含多个将要生成的程序，最终将在./build/app/目录下生成与该文件同名的可执行文件，在./build/app/目录下执行./main即可运行。

include目录下放置头文件。

lib目录下为需要该项目生成的库文件。

README.md为用 Markdown 书写的说明文档。

## 2.2 Boost C++ Libraries

### 2.2.1 简介

Boost C++ 库 (Libraries) 是一组扩充 C++ 功能的经过同行评审 (Peer-reviewed) 且开放源代码程序库。大多数的函数为了能够以开放源代码、封闭项目的方式运作，而授权于 Boost 软件许可协议 (Boost Software License) 之下。许多 Boost 的开发人员是来自 C++ 标准委员会，而部分的 Boost 库成为 C++ 的 TR1 标准之一。

为了确保库的效率与弹性，Boost 广泛的使用模板 (template) 功能。而它是针对各式领域的 C++ 用户与应用领域 (Application Domain) 上，包含的库类别从像 smart\_ptr 库这种类通用库，到像是文件系统的操作系统抽象层，甚至能够利用 Boost 来开发额外的库或是给高级的 C++ 用户利用，像是 MPL。  
[3]

## 2.3 Doxygen

### 2.3.1 简介

Doxygen 是一个 C++、C、Java、Objective-C、Python、IDL (CORBA 和 Microsoft flavors)、Fortran、VHDL、PHP、C# 和 D 语言的文档生成器。可以在大多数类 Unix 的系统上运行，以及 Mac OS X 操作系统和 Microsoft Windows。初始版本的 Doxygen 使用了一些旧版本 DOC++ 的源代码；随后，Doxygen 源代码由 Dimitri van Heesch 重写。

Doxygen 是一个编写软件参考文档的工具。该文档是直接写在源代码中，因此比较容易保持更新。Doxygen 可以交叉引用文档和源代码，使文件的读者可以很容易地引用实际的源代码。



KDE 使用 Doxygen 作为其部分文档且 KDevelop 具有内置的支持。Doxygen 的发布遵守 GNU 通用公共许可证，并且是自由软件。[4]

## 2.4 Python

### 2.4.1 简介

Python (英国发音: / pa ən/ 美国发音: / pa n/), 是一种面向对象、解释型的计算机程序语言。它包含了一组功能完备的标准库, 能够轻松完成很多常见的任务。它的语法简单, 与其它大多数程序设计语言使用大括号不一样, 它使用缩进来定义语句块。

与 Scheme、Ruby、Perl、Tcl 等动态语言一样, Python 具备垃圾回收功能, 能够自动管理内存使用。它经常被当作脚本语言用于处理系统管理任务和网络程序编写, 然而它也非常适合完成各种高级任务。Python 虚拟机本身几乎可以在所有的作业系统中运行。使用一些诸如 py2exe、PyPy、PyInstaller 之类的工具可以将 Python 源代码转换成可以脱离 Python 解释器运行的程序。

Python 的官方解释器是 CPython, 该解释器用 C 语言编写, 是一个由社区驱动的自由软件, 目前由 Python 软件基金会管理。

Python 支持命令式程序设计、面向对象程序设计、函数式编程、面向侧面的程序设计、泛型编程多种编程范式。[5]

## 2.5 SWIG

### 2.5.1 简介

SWIG 是个帮助使用 C 或者 C++ 编写的软件能与其它各种高级编程语言进行嵌入联接的开发工具。SWIG 能应用于各种不同类型的语言包括常用脚本编译语言例如 Perl, PHP, Python, Tcl, Ruby and PHP。支持语言列表中也包括非脚本编译语言, 例如 C#, Common Lisp (CLISP, Allegro CL, CFFI, UFFI), Java, Modula-3, OCAML 以及 R, 甚至是编译器或者汇编的计划应用 (Guile, MzScheme, Chicken)。SWIG 普遍应用于创建高级语言解析或汇编程序环境, 用户接口, 作为一种用来测试 C/C++ 或进行原型设计的工具。SWIG 还能够导出 XML 或 Lisp s-expressions 格式的解析树。SWIG 可以被自由使用, 发布, 修改用于商业或非商业中。[6]

## 2.6 USRP

### 2.6.1 简介

Universal Software Radio Peripheral (USRP) is a range of software-defined radios designed and sold by Ettus Research and its parent company, National Instruments. Developed by a team led by Matt Ettus, the USRP product family is intended to be a comparatively inexpensive hardware platform for software radio, and is commonly used by research labs, universities, and hobbyists.

Most USRPs connect to a host computer through a high-speed link, which the host-based software uses to control the USRP hardware and transmit/receive data. Some USRP models also integrate the general functionality of a host computer with an embedded processor that allows the USRP device to operate in a stand-alone fashion.

The USRP family was designed for accessibility, and many of the products are open source hardware. The board schematics for select USRP models are freely available for download; all USRP products are controlled with the open source UHD driver, which is free and open source software. USRPs are commonly used with the GNU Radio software suite to create complex software-defined radio systems. [7]

## 2.7 HackRF

### 2.7.1 简介

HackRF 是一款由 Michael Ossmann 发起的开源软件无线电外设，旨在从 30MHz 到 6GHz，于 2012 年从 DARPA 处拿了一笔经费，制作了 500 块测试版本 Jawbreaker，并向社会分发测试。在经过用户对 Jawbreaker 的反馈后，作者对硬件板卡做了重新布线，改善了射频性能，这一点我们将会在后文详细讨论。随后于 2013 年 7 月 31 日至 9 月 4 日共计 35 天的时间，在著名的社会化融资平台 Kickstarter 上，迅速地获得多达 1991 人的预订，共预订出价值为 \$602,960 的 HackRF One。

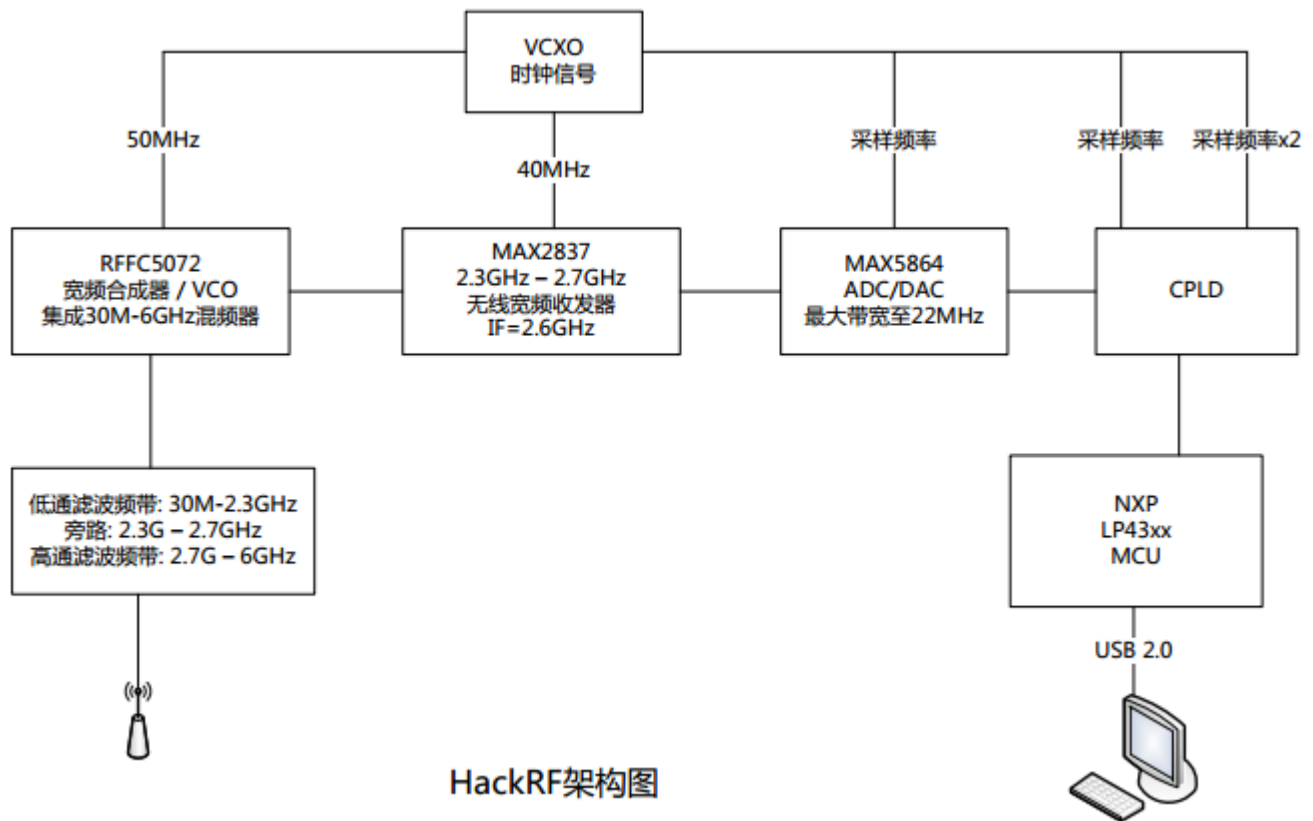


图 2.1 HackRF\_ 架构图

## 2.7.2 osmosdr

## 2.8 gqrx

### 2.8.1 简介

## 2.9 git

### 2.9.1 简介

git 是用于 Linux 内核开发的版本控制工具。与 CVS、Subversion 一类的集中式版本控制工具不同，它采用了分布式版本库的作法，不需要服务器端软件，就可以运作版本控制，使得源代码的发布和交流极其方便。git 的速度很快，这对于诸如 Linux 内核这样的大项目来说自然很重要。git 最为出色的是它的合并追踪（merge tracing）能力。

实际上内核开发团队决定开始开发和使用 git 来作为内核开发的版本控制

系统的时候，世界上开源社区的反对声音不少，最大的理由是 git 太艰涩难懂，从 git 的内部工作机制来说，的确是这样。但是随着开发的深入，git 的正常使用都由一些友善的命令稿来执行，使 git 变得非常好用。现在，越来越多的著名项目采用 git 来管理项目开发，例如：wine、U-boot 等。

作为开源自由原教旨主义项目，git 没有对版本库的浏览和修改做任何的权限限制，通过其他工具也可以达到有限的权限控制，比如：gitosis、CodeBeamer MR。原本 git 的使用范围只适用于 Linux/Unix 平台，但在 Windows 平台下的使用也日渐成熟，这主要归功于 Cygwin、msysgit 环境，以及 TortoiseGit 这样易用的 GUI 工具。git 的源代码中也已经加入了对 Cygwin 与 MinGW 编译环境的支持且逐渐完善，为 Windows 用户带来福音。

### 2.9.2 常用命令

在众多教程中，廖雪峰的官方网站 (<http://www.liaoxuefeng.com/>) 很适合初学者，深入浅出的讲解了使用 git 过程中常用的许多命令。

### 2.9.3 版本控制

### 2.9.4 github

GitHub 是一个通过 Git 进行版本控制的软件源代码托管服务，由 GitHub 公司（曾称 Logical Awesome）的开发者 Chris Wanstrath、PJ Hyett 和 Tom Preston-Werner 使用 Ruby on Rails 编写而成。

GitHub 同时提供付费账户和免费账户。这两种账户都可以创建公开的代码仓库，但是付费账户还可以创建私有的代码仓库。根据在 2009 年的 Git 用户调查，GitHub 是最流行的 Git 访问站点。除了允许个人和组织创建和访问保管中的代码以外，它也提供了一些方便社会化共同软件开发的功能，即一般人口中的社区功能，包括允许用户追踪其他用户、组织、软件库的动态，对软件代码的改动和 bug 提出评论等。GitHub 也提供了图表功能，用于概观显示开发者们怎样在代码库上工作以及软件的开发活跃程度。

截止到 2015 年，GitHub 已经有超过九百万注册用户和 2110 万代码库。事实上已经成为了世界上最大的代码存放网站和开源社区。[8]

本设计使用的众多开源库、软件均已托管于 GitHub，网址如下：

- gnuradio/gnuradio  
<https://github.com/gnuradio/gnuradio>
- gnuradio/pybombs  
<https://github.com/gnuradio/pybombs>
- EttusResearch/uhd  
<https://github.com/EttusResearch/uhd>

## 2.10 Docker

### 2.10.1 简介

Docker 是一个开放源代码软件项目，让应用程序布署在软件容器下的工作可以自动化进行，借此在 Linux 操作系统上，提供一个额外的软件抽象层，以及操作系统层虚拟化的自动管理机制。Docker 利用 Linux 核心中的资源分脱机制，例如 cgroups，以及 Linux 核心名字空间 (name space)，来创建独立的软件容器 (containers)。这可以在单一 Linux 实体下运作，避免引导一个虚拟机造成的额外负担。Linux 核心对名字空间的支持完全隔离了工作环境中应用程序的视野，包括进程树、网络、用户 ID 与挂载文件系统，而核心的 cgroup 提供资源隔离，包括 CPU、存储器、block I/O 与网络。从 0.9 版本起，Docker 在使用抽象虚拟是经由 libvirt 的 LXC 与 systemd - nspawn 提供界面的基础上，开始包括 libcontainer 库做为以自己的方式开始直接使用由 Linux 核心提供的虚拟化的设施，

依据行业分析公司“451 研究”：“Docker 是有能力打包应用程序及其虚拟容器，可以在任何 Linux 服务器上运行的依赖性工具，这有助于实现灵活性和便携性，应用程序在任何地方都可以运行，无论是公有云、私有云、单机等。” [9]

在构建应用时，如果在实机上进行运行测试，很容易因为依赖的问题导致应用无法正常运行，严重时会造成系统的崩溃。Docker 提供了一个虚拟的环境来构建自己的应用，同时也能够与 git 进行版本控制，与回滚，在 Docker 中构建应用也能更快的部署应用。需要大规模部署应用时只需要从仓库中拉取自己需要的环境以及版本即可，进行少量的配置即可运行。

## 2.10.2 构建容器

## 2.11 GNU Radio

### 2.11.1 简介

- 源码编译

- apt-get

- 树莓派

jessie 源中提供的 GNU Radio 版本为 3.7.5。

使用 jessie 源时需要手动编译 gr-dvbt 并链接动态库。

stretch 源中提供的 GNU Radio 版本为 3.7.10。

如果需要使用 stretch 源则需要修改 source.list 文件。

- Ubuntu

- pybombs 安装

### 2.11.2 版本选择

- GNU Radio 3.7.9 及以下版本 gr-dvbt 编译

- x86 平台编译

- arm 平台编译

由于 gr-dvbt 中 viterbi 解码模块使用了汇编，调用了 x86 平台上的汇编库来加速运算，所以在 arm 平台无法编译 viterbi 模块，也就导致了仅能在 arm 平台上编译发射端的部分。

需要注释以下文件中的以下语句：

- GNU Radio 3.7.10

安装完成之后使用 python 测试，gnuradio 需要 Python 2，在 Python3 环境下无法运行。

```
Python 2.7.13 (default, Jan 19 2017, 14:48:08)
```

```
[GCC 6.3.0 20170118] on linux2
```

Type `"help"`, `"copyright"`, `"credits"` or `"license"` for more information.

```
>>> import gnuradio
>>>
```

如果没有出现no module named gnuradio则表示安装成功。

### 第三章 DVB-T 发射端

#### 3.1 能量扩散

#### 3.2 RS 编码

#### 3.3 卷积交织

#### 3.4 卷积编码)

#### 3.5 比特交织

#### 3.6 符号交织

#### 3.7 星座映射

#### 3.8 参考信号

#### 3.9 IFFT

#### 3.10 OFDM 循环前缀

#### 3.11 常数

#### 3.12 重采样

信号的采样率，用于满足另一个系统的要求

#### 3.13 USRP 发射



## 第四章 DVB-T 接收端

### 4.1 USRP 接收、信号源

### 4.2 重采样

### 4.3 乘常数

### 4.4 OFDM 符号查询

### 4.5 FFT

### 4.6 解参考信号

### 4.7 星座解映射

### 4.8 符号解交织

### 4.9 比特内解交织

### 4.10 矢量串流

### 4.11 维特比解码

汇编加速

### 4.12 卷积解交织

### 4.13 RS 解码

### 4.14 能量解扩散

### 4.15 文件输出

## 参考文献

- [1] 维基百科. 树莓派 — 维基百科, 自由的百科全书 2017.
- [2] 维基百科. Cmake — 维基百科, 自由的百科全书 2017.
- [3] 维基百科. Boost c++ libraries — 维基百科, 自由的百科全书 2017.
- [4] 维基百科. Doxygen — 维基百科, 自由的百科全书 2017.
- [5] 维基百科. Python — 维基百科, 自由的百科全书 2017.
- [6] Wikipedia. Swig — wikipedia, the free encyclopedia2017.
- [7] Wikipedia. Universal software radio peripheral — wikipedia, the free encyclopedia2017.
- [8] 维基百科. Github — 维基百科, 自由的百科全书 2017.
- [9] 维基百科. Docker (軟體) — 维基百科, 自由的百科全书 2017.