



Python在嵌入式项目中的辅助开发

彭树林

摘要: 嵌入式系统设计开发过程中常会遇到诸如算法分析、原型验证、自动化测试、辅助工具设计等工作,其开发效率和质量直接影响到项目的进程。对于目前国内众多开发人员,迫切需要这样一种软件开发工具:易学、高效、功能强。Python作为这样一种开源脚本语言,配合众多实用的第三方函数库,可极大地提高嵌入式系统开发效率和质量。作者结合自己在几个项目开发中的实践经验,对此进行了入门介绍。

关键词: Python; 嵌入式系统

引言

嵌入式设计开发通常是比较系统的工程,对开发人员有较全面的要求,硬件、软件、算法、测试等各个方面都得照顾到,所有环节有效的配合才能保证整个项目的顺利进展,对时间质量要求高的项目更是如此。

实际开发过程中常会遇到问题有算法分析、原型验证、自动化测试、辅助工具设计等,选择好恰当的工具,对提高工作效率和质量至关重要。本文要介绍的Python脚本语言和众多第三方函数库就是这样的利器:易学、高效、功能强,值得推广。

1 Python简介

Python是一种流行的动态脚本语言,经历了十多年的发展现已相当成熟稳定。其语法简洁清晰、面向对象、跨平台、扩展性强,适合完成各种高层任务,既可快速开发脚本程序,也可用来开发大规模软件。

由于遵循开源GPL协议,用户众多,反过来也推动了第三方函数库的稳步发展,现数量众多功能强大的函数库(其中大部分也是开源的)能适应几乎所有的应用:字符处理、系统控制、图像处理、图形界面、数据库、科学计算、网络应用、单元测试等。

本文不介绍Python语言的基础知识,感兴趣的读者请自行查阅相关内容。推荐的学习材料有:官方网站、安装自带的文档、第三方库的文档、网络社区等。

2 函数库介绍

函数库分两类:标准库和第三方库。前者是安装时自带的,能完成基础的字符、文件、系统、网络等控制,后者是在前者基础上扩充开发的,也是Python最成功之处,现就常用于嵌入式开发的几个库分类介绍,以下库均遵循GPL协议。

2.1 科学计算

◆ NumPy、SciPy

NumPy数值计算包提供了矩阵、线性代数、傅里叶变换等的解决方法。

SciPy提供更强的数学计算分析功能:线性计算、信号处理、图像处理、最优化求解等。大多数函数接口也与Matlab类似,容易上手。

◆ Matplotlib

二维图形绘制模块,包括线图、直方图、饼图、散点图以及误差线图,输出效果相当专业,许多函数接口也与Matlab类似。

◆ SymPy

符号运算,可进行算法仿真、模拟。

2.2 图像处理

◆ PIL

提供图像处理操作:裁剪、旋转、滤波、增强等。嵌入式开发中经常需要操纵图形资源,用于点阵液晶屏幕显示,如图形转化成C语言数组、屏幕缓冲数据转换成图片,都可借用此模块实现。

◆ MayaVi

用于绘制三维数据图形。

2.3 图形界面

◆ wxPython

高效的GUI工具箱,提供图形接口,可生成标准风格的Windows程序窗口,新版本中包含的许多控件非常有特色。

◆ wxGlade

图形程序的开发工具,用wxGlade绘制好窗体后生成Python代码,加入函数相应代码即可。

2.4 数据库

◆ pySqlite

轻量级的数据库,现已成为标准库。

2.5 硬件控制

◆ pyParallel、pySerial

用于控制PC机的串口和串口。

◆ pyVISA

控制VISA接口设备。

◆ pyAudio

声音控制,若需用操纵声音资源,集成至嵌入式系统固件中,可用此模块。

2.6 其它

◆ Socket、FTP、Telnet

网络接口,可用于控制测量仪器和设备,TCP/UDP方式调试网络设备。

◆ 动态链接库接口

某些必须由其它语言实现的功能可先编译成动态库,再由Python调用。

3 项目案例

以下介绍作者在实际项目中使用Python提高开发效率的几个例子。

3.1 位图资源的C代码转换

某便携式仪表支持16级灰度点阵液晶,需要显示若干图标、图片。素材先由图形处理软件修饰完成,存为灰度位图格式文件,再调用Python编的处理脚本,转换图片成C语言格式数组,集成到系统项目中,编译至固件,最终显示在仪表屏幕上。

图1为待转换的图标IconHelp.Bmp,转换后的结果IconHelp.c如下:

```
// Converted with PYTHON
```



图1 待转换的图标

```
// 32 x 32 pixels
const unsigned short IconHelp[] = {
    0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0
    xFFFF,
    0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0
    xFFFF,
```

```
0xFFFF,0xFFFF,0xFFDF,0xF7BE,0xDEDB,0xAD75,0x8C71,0x8410,
```

```
0x7BEF,0xE71C,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
```

```
0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,0xFFFF,
```

...

与上述正向转换对应的还有逆向转换：仪表调试过程中下断点，使用调试软件脚本命令按一定ASCII格式取出屏幕缓存，编写相应Python脚本将之转换成位图图片，实现了仪表屏幕的“截图”，可用作产品使用说明书的素材。

3.2 Agilent信号发生器的控制

硬件开发中常用的示波器、信号源、高精度万用表等仪器大都有编程控制功能，通过串口、以太网、GPIB等接口和相关协议控制。以Agilent 3320A信号源为例，通过以太网与PC组网连接后，PC机在5024端口用Telnet协议与信号源建立连接，用Python脚本通过SCPI协议控制。利用Python对象化语言的优势将信号源封装成类，设置功能（如开关、波形类型、幅度、频率、扫频、调制等）均被抽象为事件，使用时只要调用该类对象的函数即可。

如一次硬件测试中需要依次输出频率为10、20、50、100、...10K、20K、50K、100KHz，幅度1V的正弦波，每段的间隔时间0.5秒，只需调用封装好的类，实现如下：

```
d = AGILENT33220A() # 建立实例
d.Connect() # 网络连接
d.Reset() # 仪器复位
for i in [10,20,50,100,200,500,1000,2000,5000,10000,20000,50000,100000]:
```

```
    d.SetParams('SIN', i, 1, 0) # 输出正弦波
```

```
    time.sleep(0.5) # 延时半秒
```

```
d.Output(False) # 关闭输出
```

3.3 开关机电路测试

某款手持式仪表在开机时偶尔出现白屏无法启动的现象，测试人员不得不手动地做开关机测试，每次循环周期约15秒，但故障概率极低，甚至几小时测下来也不能重现，整个测试过程费时费力。

为提高效率，开发了一套自动测试装置：使用Python脚本控制并口，用一输出端口控制连接在仪表电源按键上的继电器，模拟用户的按键操作，再用几个输入端口检测仪表工作电源是否上电、开机后的LED是否点亮等，主程序负责协调整个流程，若发现开机异常，发出蜂鸣器报警。

经过数天的自动测试，累积数千次的开关机测试，都未发

现异常，从而排除了电路方面的因素，将故障被定位在其它方面，如结构、器件质量等。

3.4 辅助工具：三次样条曲线系数计算器

开发过程中常会将一些算法的参数预先计算好，固化到项目程序中去，如FFT的旋转因子、FIR滤波的系数等，利用Scipy库可方便地计算。

这里介绍的系数计算器是一带图形界面的辅助工具，用来生成样条曲线拟合时用的中间参数，同时还能计算拟合值并绘制曲线图形。用wxPython制作窗体界面，matplotlib绘制曲线图形，核心的参数计算代码因为要求与嵌入式CPU通用，故使用了C实现并封装成DLL被调用的。

在计算锂电池电压对应电量关系时的界面如图2，数据源是电压和电量的控制点，计算出的参数在终端以文本形式输出，拟合出的曲线用matplotlib自带的绘图控件显示，可移动、缩放、保存。

3.5 通讯可靠性测试

某仪表内部包含自行开发文件系统，通过串口与PC机通讯，通讯驱动接口被封装成动态库，为测试文件系统和通讯的可靠性，制定了如下方案：

编写Python脚本，动态生成指定大小的测试用二进制文件，内容随机产生，调用驱动与仪表建立连接，将测试文件下载至仪器，再回收并与源文件逐字节作对比，全部正确则测试通过，如此反复测试。

经连续数天测试，发现了仪表底层通讯程序中的一处漏洞，信号量未加锁导致数据丢失，修改程序后再测试问题不再重现，漏洞解决。图3为测试时界面。



图3 通讯可靠性测试

结语

由于上述软件都遵循GPL开源协议，加上能和商业软件媲美的性能，使之有很强的生命力。每个工具都有其擅长的一面，将它们有效组合熟练运用，可以很大程度地提高开发效率和质量。愿本文的案例能抛砖引玉，给同行们的设计工作其点借鉴作用。

H

参考文献

- [1] 周伟, 宗杰. Python开发技术详解[M]. 北京: 机械工业出版社, 2009.
- [2] 周伟, 宗杰. Python开发技术详解[M]. 北京: 机械工业出版社, 2009.
- [3] 陈儒. Python源码剖析——深度探索动态语言核心技术[M]. 北京: 电子工业出版社, 2008.

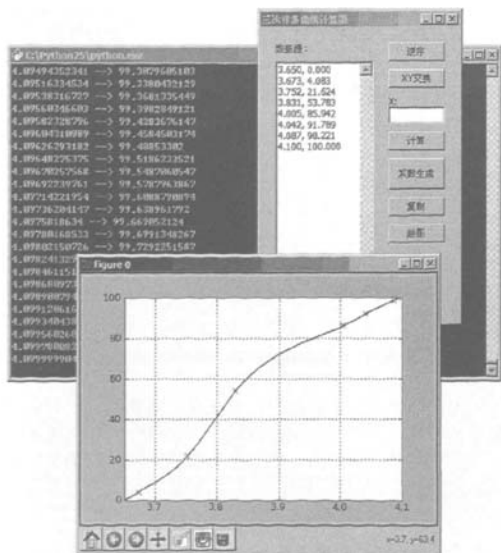


图2 三次样条曲线系数计算器

作者: [彭树林](#)
作者单位:
刊名: [信息系统工程](#)
英文刊名: [CHINA CIO NEWS](#)
年, 卷(期): 2010, "" (7)
被引用次数: 0次

参考文献(3条)

1. 周伟, 宗杰 [Python开发技术详解](#) 2009
2. 周伟, 宗杰 [Python开发技术详解](#) 2009
3. 陈儒 [Python源码剖析—深度探索动态语言核心技术](#) 2008

相似文献(5条)

1. 期刊论文 [张晔, 柴志雷, 涂时亮, ZHANG Mi, CHAI Zhi-lei, TU Shi-liang 动态系统描述语言的特性、设计和实现](#)

-小型微型计算机系统2010, 31(5)

介绍一种新型的动态系统描述语言, PDSDL. 它比SystemC更加高效和灵活. (1) PDSDL使用了Python面向对象动态语言, (2) PDSDL创新性地提出了动态系统的概念. 系统的建模、验证和综合全都依赖于运行时动态系统特性, 而非传统的存储在非易失记忆体(例如文件)的静态描述. 因此设计者能使用程序和高级人工智能方式来进行更高效的结构探索. 本文的重点在于探讨动态系统描述语言的特性和实现机理. 文末给出简要试验结果.

2. 学位论文 [蒋维 基于无线网络的高扩展性嵌入式环境监控系统](#) 2006

环境污染源监控是我国积极贯彻可持续发展战略, 实施环境综合整治, 实现经济发展和环境、生态保护和和谐统一的重要手段. 本文立足于网络时代环境污染源监控的新需求、新挑战, 设计了基于嵌入式Arm-Linux的监控节点的解决方案, 并针对其无线网络应用进行了深入的研究, 最后通过在监控节点嵌入式系统上的Python组件的移植成功, 实现了高扩展性的监控节点架构, 为解决网络时代环境监控的弹性需求提供了全新的技术路线.

纵观全文, 本文的主要工作有: (1) 对环境监测的现状和应用需求进行了系统性的综合分析, 提出了基于嵌入式Linux系统的监控节点技术路线.

(2) 采用Arm嵌入式系统作为环境监控系统的硬件支撑平台. 重点分析了系统设计中所涉及的关键环节及其软件支撑平台. 在软硬件支撑的基础之上, 以面向对象的编程方法设计并实现了环境监控系统框架.

(3) 为克服无线网络传输中存在的断线率高、网络流量受限、传输延迟性大等技术难题, 在监控系统中设计了应用层的交接协议, 并对无线网络中时间同步和网络传输代价等问题进行了深入研究, 提出了一种通过定量优化方法确定元数据包大小的新方法.

(4) 通过在监控节点嵌入式系统上成功移植Python组件, 实现了高扩展性的监控节点架构. 采用计算能力扩展、动作服务扩展、指令集扩展、定时器等多种模式, 能够满足新一代环境监控系统复杂多变的应用需求.

3. 学位论文 [王伟波 嵌入式系统软件IDE设计与实现](#) 2005

本文主要介绍了在探索嵌入式系统软件IDE开发技术的过程中, 实现名为EideP的一个集成开发环境的过程. 本论文的重点是JTAG在线(In-Circuit)调试技术的实现. 在嵌入式开发领域, 最重要、功能最强大的调试方式就是在线调试. 通过目前大多数Processor都带有的JTAG接口, 利用Processor内部提供的特有的调试机制, 可以实现成本最低且功能强大的JTAG在线调试功能. 本文以PXA255处理器上实现JTAG在线调试的过程, 将JTAG在线调试的内部机制以及实现要点进行了比较深入的探讨. 最终, 通过EideP的pygdbserver模块, 将JTAG调试同gdb调试结合在了一起, 形成一个功能强大的嵌入式环境在线调试架构. 本文通过分析EideP中的代码编辑模块、软件调试模块、JTAG在线调试模块, 介绍了嵌入式软件IDE的主要功能与实现, 也提出了理想中的IDE所应具有的特性, 具有一定参考价值.

4. 期刊论文 [赵曦, 杨福兴 一种新型智能机器人控制系统设计](#) -科技信息2009, "" (1)

为异构式多智能体机器人协作研究, 设计并实现了一种基于32位嵌入式系统的分布式控制系统. 针对传统封闭式控制系统的弊端, 采用了开放式模块化的硬件结构, 在ARM核心模块基础上通过IIC总线扩展功能模块. 功能模块以A讯单片机为核心, 独立完成模块功能. 软件系统采用了混合式体系结构, 应用模块在嵌入式Linux平台上使用自顶向下分层设计, 以多任务和基于Tcp协议的可靠多机器人通讯机制管理各功能模块, 并引入Python脚本解决任务策略变换的灵活性问题. 采用了黑板和点对点相结合的多机器人协作通信模式. 实验证明, 该系统具有良好的开放性, 为异构设计、功能扩展、人机混合控制和多机器人协作提供了可靠的平台.

5. 学位论文 [蔡文豪 基于SNMP的太阳能远程管理系统及其SOPC实现](#) 2008

本文针对远程管理太阳能设备的需求, 归纳比较了目前流行的网络管理协议及其发展现状, 最终提出并实现了一种基于简单网络管理协议(SNMP)的太阳能设备能源管理系统. 系统由SNMP Agent端和SNMPManager端两部分组成.

SNMP Agent设计硬件部分采用基于NIOS-II的SOPC系统实现, 软件部分基于uClinux嵌入式操作系统, 采用NET-SNMP 开源软件包, 通过编写前端数据采集的设备驱动程序以及管理信息库(MIB)的agent扩展, 使得对太阳能设备信息的监控通过SNMP协议远程实现. 同时, 为SOPC系统定制U-BOOT, 从而完善SNMP Agent嵌入式系统的引导与固件更新.

SNMP Manager设计基于SNMP++开源软件包, 实现MIB Browser的设计. 同时, 为进一步实现Manager端自动化管理的任务, 采用Python实现自动化Manager端的设计. 测试结果表明, 该设计可以有效地实现远程管理的功能, 对有安全要求的监控领域有应用和参考价值.

本文根据作者实际开发经验, 以理论结合实例的形式归纳总结了作者对于SNMP的理解, 以及在SNMP开发中MIB文件编写的主要规则, 对基于SNMP的网络管理应用有很强的实用价值.

本文链接: http://d.g.wanfangdata.com.cn/Periodical_xxtgc201007007.aspx

授权使用: 北京邮电大学(byab), 授权号: 732a8a77-282a-4fb1-a746-9e4300ec8aaa

下载时间: 2010年12月5日