问题集

# C语言

## char类型数据和uint8\_t 类型之间赋值出现数据错误

char类型数据存储范围为-127~128之间（8个数据位），在内存中是以补码的形式存储，而uin8\_t类型数据就是整型存储（非补码形式存储），所以两者的存储方式不同，如果需要相互赋值则必须经过强制转换，否则两者在内存中的值是不相同的，直接使用会导致错误。

## uint16\_t的指针指向一块连续的地址后，通过指针获取值错误

# C++

# QT

## QRect赋值不成功：

对QRect对象的赋值需要对其整个对象的(x, y, w, h)同时赋值（setRect或创建对象时一次性传递），单独赋值无效，原因不明

## QTableWidget删除所有行不彻底:

多次删除应该从最后一行开始逐行删除；

## 自定义类继承的某些类功能无法使用

参考qt帮助文档，通常需要在.pro文件里加入QT += \*\*\*

# python

## pycharm创建虚拟环境:

将每个工程的环境分割开互不影响

## 代码对其以及字体等宽设置：

须设置字体font：Courier New

设置table大小

## PySide 编译.ui文件生成对应的.py文件，以及编译资源文件：

pyside-uic MainWindow.ui -o UI\_MainWindow.py

## python生成requirements文件：

pip freeze > requirements.txt ：生成文档

pip install –r requirements.txt ：安装依赖

## Python docx表格宽度设置无效：

table.autofit = **False # 关闭自动宽度**

table.cell(row, cloumn).width = Inches(0.8) #宽度设置为英尺

# Git使用

## git 相关子模块在clone时，工程中未下载：

子模块需要通过执行以下命令进行下载

git submodule init

git submodule update [--depth]

## git 如何添加版本号：

在git终端进入项目目录，输入如下命令

git tag -a v1.0.1 //进入vim输入相关信息

## git 版本号相关版本信息本地与服务器未同步：

git 获取版本号(git describe --tags --dirty)后，需要执行以下命令将tages上传至服务器

git push --tags

## git 上传至gitlab时换行符问题：

在项目中添加如下上传是文档，按约定自动将换行符替换掉

.gitattributes

## gitk 使用gitdiff 文件对比无法打开：

需要在服务器（\\172.16.0.5\云擎科技\100\_常用软件\039\_git比对工具）下载并安装meld 并添加环境变量（例如： E:\Program Files (x86)\Meld）

## git 终端如何远程Ubuntu：

需要安装SSH远程接入工具；

Ubuntu安装openssh-server

git终端输入：ssh username@ip地址.

## Ubuntu gitlab无法clone工程：

gitlab主机地址没有指定：

sudo vi /etc/hosts 添加如下

172.16.0.5 gitlab.local

## No names foud

编译版本号时未打标签，需要先添加版本号（见5.2）。

## git stash使用

将当前修改的文件添加到栈中，git stash

将栈中临时保存的文件出栈，git stash pop

## git init 关联到远程仓库

git init -> git add . ->git commit；

git remote add origin <https://github.com/rukehu/ngr_rps.git>。（添加远程仓库）

git push --set-upstream origin master

git branch --set-upstream-to=origin/master master。（本地仓库与远程仓库关联）

git pull origin master --allow-unrelated-histories。（拉取远程仓库内容到本地）

# Svn使用

# bat脚本

## 问题：

脚本运行时显示echo处于关闭状态

## 原因：

在脚本中变量定义时两端含有空格或table，bat脚本中谨慎使用空格

# 软件设计

问题：

1.考虑不全面；

2.设计的合理性；

3.使用的合理性；

4.接口设计的合理性；