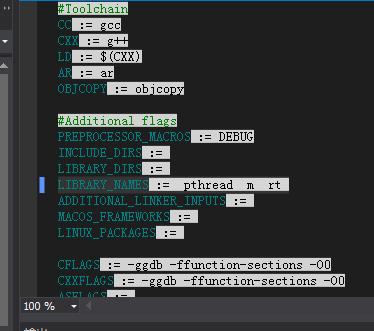
**问题1**： VisualGDB : error : Command-line action failederror VGDB1000: undefined reference to `mq\_unlink' D:\jdagv\jdagv\jdagv\jdagvserver\mq.c 72 1 jdagvC:\Program Files (x86)\MSBuild\Microsoft.Cpp\v4.0\V120\Microsoft.MakeFile.Targets(38,5): error MSB3073: 命令“"C:\Program Files (x86)\Sysprogs\VisualGDB\\VisualGDB.exe" /build "D:\handsome\jdagvserver\LinuxProject1\LinuxProject1

解决1：找不到源文件的头文件，库路径设置问题。放在根目录下，在mak文件中添加一句LIBRARY\_NAMES := pthread m rt 

或者用配置添加文件夹作为索引也行**问题2**：不可执行的二进制代码

解决：可执行文件关机时，权限没有同步上，程序中调用fsync（），命令行时用sync指令

**问题3**：二进制代码执行时没有data，下载的文件时乱码

解决3：程序写数据时，fllush只写到了内核，没有立即写到磁盘中，只要更改文件候，就要调用fsync（）；

**问题4**：实时查看文件的变化

解决4： tail –f filename

**问题5**：git远程仓库版本回退

解决：1.git revert HEAD(时间点) ——> 2. git push origin master(分支)

**问题6**：查看wifi wlan0无线信号强度

解决：

**问题7**：vi vim打开文件出现提示，不能直接打开

解决7: rm ~/.viminfo

**问题8**：查看wifi信号强度

解决8:iwconfig wlan0 或 iwlist wlan0 sacn

**问题9**：查看本地静态路由表

解决9：route –n

[root@localhost ~]# route  
Kernel IP routing table  
Destination Gateway Genmask Flags Metric Ref Use Iface  
192.168.120.0 \* 255.255.255.0 U 0 0 0 eth0  
192.168.0.0 192.168.120.1 255.255.0.0 UG 0 0 0 eth0  
10.0.0.0 192.168.120.1 255.0.0.0 UG 0 0 0 eth0  
default 192.168.120.240 0.0.0.0 UG 0 0 0 eth0

说明：

第一行表示主机所在网络的地址为192.168.120.0，若数据传送目标是在本局域网内通信，则可直接通过eth0转发数据包;

第四行表示数据传送目的是访问Internet，则由接口eth0，将数据包发送到网关192. 168. 120. 240

Flags标志说明：

U Up表示此路由当前为启动状态

H Host，表示此网关为一主机

G Gateway，表示此网关为一路由器

R Reinstate Route，使用动态路由重新初始化的路由

D Dynamically,此路由是动态性地写入

M Modified，此路由是由路由守护程序或导向器动态修改

! 表示此路由当前为关闭状态

**问题10**:重启无线网卡，断开/连接网卡

解决10：sudo nmcli nm wifi off ——>sudo nmcli nm wifi on; command-line tool for con‐

trolling NetworkManager

ifconfig wlan0 down/up

查看网卡已保存的连接：nmcli con

删除网卡已保存的连接：nmcli con delete uuid cecee8d9-4dad-49bb-8f5d-f2d59499b65e

查看硬件ap：nmcli dev wifi

查找所有可用的wifi接入点，记录接入点名称（essid）：iwlist scan|egrep 'ESSID|Link|Signal'

**问题11**：查看文件大小

解决11：du –sh filename

**解决12**：Xshell 无法连接虚拟机中的ubuntu的问题

解决12：1 VM中网络连接设置了Bridge模式。2. apt-get install openssh-server 3. /etc/init.d/ssh start——启动ssh

**问题13**：本地仓库关联到远程仓库

解决13：…or create a new repository on the command line

git init

git add README.md

git commit -m "first commit"

git remote add origin git@github.com:wmh11112345/unix311.git

git push -u origin master

…or push an existing repository from the command line

git remote add origin git@github.com:wmh11112345/unix311.git

git push -u origin master

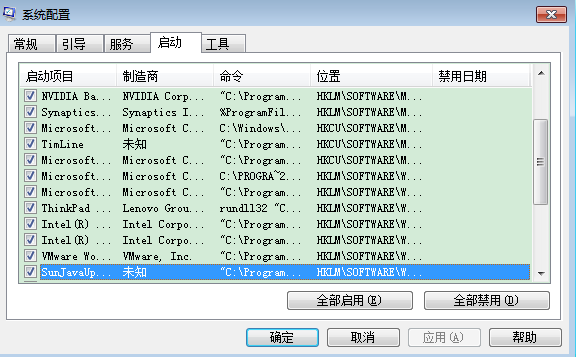
**问题14**：修改远程仓库的commit信息

**解决14：**A：git rebase –i –-root ——> fixup ——>git push –u origin master –f //不会影响代码，只影响log记录

B：回退版本：git revert #sha //#sha版本的改动回退并commit

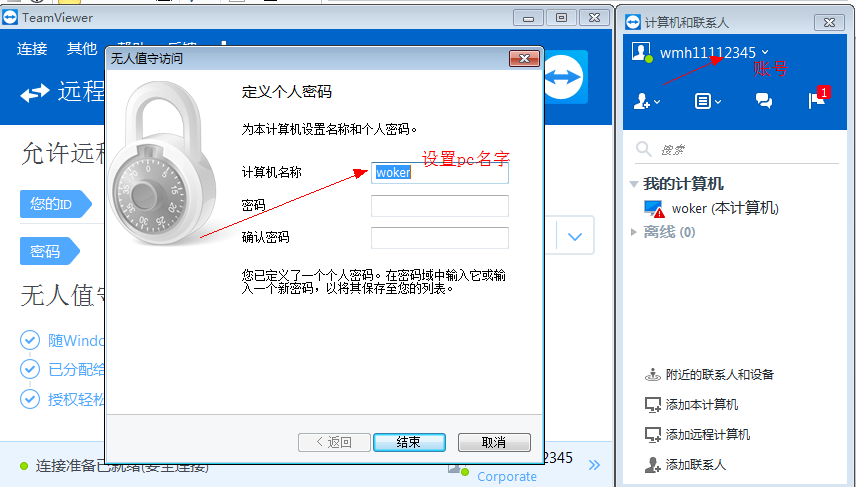
C: git commit –amend,修改上一次的提交记录 ——> git push –f

**问题15：**windows开机自启动设置

**解决15：**1.找到windows 7开始运行--在运行命令框中输入：msconfig 回车确认进入系统配置实用程序； 2. 

**问题16：**远程控制电脑

**解决16：**在pc端安装teamviewer软件，注册账号——>将pc添加到账号中——>并设置为“无人值守访问”，



**问题17：**查看端口收到、发出的数据？

**解决17：**tcpdump udp -i wlan0 -vv -n |grep 3001

所有主机的消息 tcpdump host 192.168.1.99

查看主机的ip端口是否启用：telnet 192.168.1.99 3001

**问题18：**电机与抱闸（刹车闸）的关系？

**解决18：**有些自带抱闸，自带抱闸的上电会解抱闸，断电会自动抱闸；有些不自带抱闸的可以人为给添加上抱闸装置，通过控制也能给电机抱上闸**问题19：**实时监控文件的最后10行，并标记多个关键字

**解决19：**tail -f /var/log/jdagv.log | egrep 'write\_|mc\_param'

**问题20：**输出文件中的所有的关键字的行

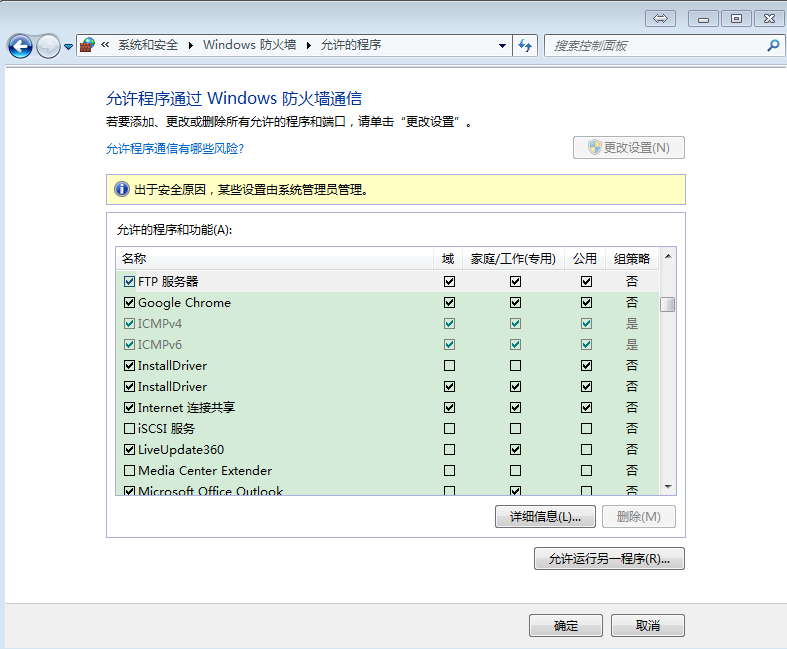
**解决20**：less /var/log/jdagv.log |egrep 'svalue|fAverageYError|eeprom|keeping|finished| BelowZeroPercent'

**问题21**：设置系统时间

**解决21：**date -s "2017-12-05 14:56:54" ——> hwclock -w

**问题22：**本地可以登陆ftp服务器，但虚拟机和实验班登陆ftp服务器超时

**解决22：**关闭防火墙或是允许ftp程序通过Windows防火墙通信



**问题23：**如何使用win7，passive模式连接到ftp服务器

**解决23**：使用xshell，默认使用passive模式

**问题24**：keil如何将hex文件转为bin文件

**解决24**：C:\Keil\ARM\ARMCC\bin\fromelf.exe --bin -o ..\output\turn\_bed.bin ..\output\turn\_bed.axf

**问题25**：两台电脑就如何通过网线传文件

**解决25**：一台电脑自动获取动态IP，组成局域网，再用飞秋传。

**问题26**：如何查看网关

**解决26**：route指令：route

**问题27：**AGV网络不可到达

**解决27：**小车ping网关不同，重启网卡导致

**问题28：**高质量c语言命名风格，函数接口的质量评估，源文件规划

**解决28：**

**原则：易于理解、便于维护**

1. 对数据操作避免裸数据，应使用结构体来定义数据，及操作——优点：改动时，只要改动结构体就行了，改一处即可，不必处处都更改。

2.尽量少使用字符串数组来作为描述——优点：对应关系明显，多处调用时得多处修改，易于维护。

3.枚举值统一用大写。

**原则：避免命名冲突**

1.自定义的数据结构统一为全大写，全局变量增加g\_xxx(xxx为大写源文件名)前缀，静态变量增加s\_前缀，避免命名冲突，结构体成员加前缀，如：

typedef struct os\_flag\_grp { /\* Event Flag Group \*/

INT8U u8OSFlagType; /\* Should be set to OS\_EVENT\_TYPE\_FLAG\*/

void \*pOSFlagWaitList; /\* Pointer to first NODE of task waiting on event flag\*/

OS\_FLAGS OSFlagFlags; /\* 8, 16 or 32 bit flags \*/

#if OS\_FLAG\_NAME\_EN > 0u

INT8U \*pOSFlagName;

#endif

} OS\_FLAG\_GRP;

Int g\_iFTPIp;

Static int s\_iIp;

2.文件命名统一小写 xxx\_xxx.h或xxx.h，Windows系统不区分大小写，但是Linux系统则区分，如：stm32f4xx\_spi.c ，ftp.h

3.函数命名:接口采用：xxx\_主\_胃、xxx\_动词或xxx\_动词\_名词；非接口采用：static主\_胃、static xxx\_动词或static xxx\_动词\_名词;形参为指针时const修饰

**原则：质量优先**

1. 正确性，指程序要实现设计要求的功能

2. 简洁性，指程序易于理解并且易于实现。

3. 可维护性，指程序被修改的能力，包括纠错、改进、新需求或功能规格变化的适应能力

4. 可靠性，指程序在给定时间间隔和环境条件下，按设计要求成功运行程序的概率。

优化：1.结构体储存结构 2.if swtich分支按触发概率来排序书写，且循环时不出现函数调用

3.避免反复从内存加载数据到cache，如：多维数据，回跳跃式访问数组成员

4. 创建资源库，以减少分配对象的开销，如：队列采用数组来实现。

接口设计

**原则：隐藏细节；**

同一级别创建与释放资源；较低层次发现错误，在较高层次处理错误；减少耦合；参数取值校验；参数的属性

**问题 29：**ftp传输小文件时，比如44bytes，发生粘包

**解决 29：**增大小文件的大小。

**问题 30：**Ubuntu系统中没有/var/log/messages文件

**解决 30：**系统日志信息在/etc/rsyslog.d/50-default.conf中规划，用vim打开/etc/rsyslog.d/50-default.conf文件增加一行内容如下：

　　\*.info;mail.none;authpriv.none;cron.none /var/log/messages——>重启系统

**问题31：**如何定位小车/linux系统重启原因

**解决31：**1重启记录的log 2系统电压监控 3 重启的类型、行为及信号

cat /var/log/messages | grep 'reboot'

查看系统重启时间：last reboot

**ubuntu下var-log下各个日志文件的作用说明**

/var/log/alternatives.log-更新替代信息都记录在这个文件中

**/var/log/apport.log -应用程序崩溃记录**

/var/log/apt/ -用apt-get安装卸载软件的信息

/var/log/auth.log -登录认证log

/var/log/boot.log -包含系统启动时的日志。

/var/log/btmp -记录所有失败启动信息

/var/log/Consolekit - 记录控制台信息

/var/log/cpus - 涉及所有打印信息的日志

/var/log/dist-upgrade - dist-upgrade这种更新方式的信息

/var/log/dmesg -包含内核缓冲信息（kernel ringbuffer）。在系统启动时，显示屏幕上的与硬件有关的信息

/var/log/dpkg.log - 包括安装或dpkg命令清除软件包的日志。

/var/log/faillog - 包含用户登录失败信息。此外，错误登录命令也会记录在本文件中。

/var/log/fontconfig.log -与字体配置有关的log。

/var/log/fsck - 文件系统日志

/var/log/faillog -包含用户登录失败信息。此外，错误登录命令也会记录在本文件中。

/var/log/hp/

/var/log/install/

/var/log/jokey.log

/var/log/kern.log –包含内核产生的日志，有助于在定制内核时解决问题。

/var/log/lastlog —记录所有用户的最近信息。这不是一个ASCII文件，因此需要用lastlog命令查看内容。

/var/log/faillog –包含用户登录失败信息。此外，错误登录命令也会记录在本文件中。

/var/log/lightdm/

/var/log/mail/ – 这个子目录包含邮件服务器的额外日志。

/var/log/mail.err -类似于上面的

/var/log/news/

/var/log/pm-powersave.log

/var/log/samba/ –包含由samba存储的信息。

/var/log/syss.log

/var/log/speech-dispacher/

/var/log/udev

/var/log/ufw.log

/var/log/upstart/

/var/log/uattended-upgrades/

/var/log/wtmp —包含登录信息。使用wtmp可以找出谁正在登陆进入系统，谁使用命令显示这个文件或信息等。

/var/log/xorg.\*.log— 来自X的日志信息。

**问题32：**init: failsafe main process (2249) killed by TERM signal是什么意思

**解决32：**failsafe is a process on systems that use upstart that detects if the system failed to boot properly and then takes appropriate action. It always starts during the boot process but is killed by upstart if the boot was successful.

**问题33：**CAN消息重复接收两次

**解决33：**在过滤的时候，添加了两次，导致消息发送了两次

**问题34：**芯片流水线对程序设计的影响，比如ARM芯片

**解决34：**指令周期分为取值，译码，直行等，取指令、存储器读、存储器写等，这每一项工作称为一个基本操作（完成一个基本操作所需要的时间称为机器周期），几级流水线是同时执行几个基本操作的能力，对优化代码的方式待定，例如for循环把有依赖的代码分解成几个可以在流水线执行单元中并行执行的没有依赖的代码链《编写高质量代码，建议21》

**问题35：**log文件中最后几行出现乱码

**解决35：**乱码是由于没有将buffer中的数据刷到磁盘导致，与开关文件没有关系，可以调用时强制刷到磁盘中

**问题36：**#！／bin／sh -e参数的意思

**解决36:** -e的参数的作用是：每条指令之后后，都可以用#？去判断他的返回值，零就是正确执行，非零就是执行有误，加了-e之后，就不用自己写代码去判断返回值，返回非零，脚本就会退出。

**问题37：**如何查看shell脚本的退出值

**解决37：**echo变量含义

$0 当前脚本的文件名

$n 传递给脚本或函数的参数。n 是一个数字，表示第几个参数。例如，第一个参数是$1，第二个参数是$2。

$# 传递给脚本或函数的参数个数。

$\* 传递给脚本或函数的所有参数。

$@ 传递给脚本或函数的所有参数。被双引号(" ")包含时，与 $\* 稍有不同，下面将会讲到。

$? 上个命令的退出状态，或函数的返回值。

$$ 当前Shell进程ID。对于 Shell 脚本，就是这些脚本所在的进程ID。

**问题38：**RTNETLINK answers: Device or resource busy

**解决38：**原因是设备已经被打开，仍然被设置属性，解决办法，先把设备关闭再设置属性

问题39：can接口的属性查看

解决39：设置：ip link set can0 type can bitrate xxx ；关：ifconfig can0 down ;开ip link set can0 up type can;查：ip -details link show can0；

问题40：如何定义变参数函数和宏函数

解决40：

小结：调车收敛是angle< 0.004平均值的波动会比较小。Angle和y反复调整

方案1：之调节y，效果不好

方案2：先调节angle，再调节y，会导致angle比较大，一致都稳定不下来

方案3：循环调节angle和y，并收敛时如果在angle<0.004和y的平均误差<0.45mm，结束自学习。

效果：调好的车95%误差在12mm以内，85%以上在10mm以内，几乎不超过15mm的误差

使用：tail -f /var/log/jdagv.log |egrep 'finished'——有输出自学习结束