

华中农业大学

学生实验报告

专业：计算机科学与技术

姓名：高星杰

学号：2021307220712

日期：2024 年 3 月 11 日

成绩：

实验课程	嵌入式系统 A 实验	实验名称	嵌入式编程基础实验	实验类型	验证 综合	设计 基础
实验目的： <ol style="list-style-type: none">了解开发板 Tiny4412 的功能、资源、开发环境及其基本使用；掌握 Ubuntu 上 arm-linux-gcc 交叉编译环境的配置和使用；掌握通过 minicom 实现主机和开发板间串口通信；将 examples 编译，下载到开发板并运行。 实验要求： <ol style="list-style-type: none">C 语言编程基础。掌握在 Linux 下常用编辑器 gedit,vi 的使用。掌握 Makefile 的编写和使用。掌握 Linux 下的程序编译与交叉编译过程掌握 minicom 使用熟悉 Tiny4412 开发环境						
实验环境 （含主要设计设备，器材，软件等） 硬件： ubuntu 主机,Tiny4412 开发板 软件： linux 系统、ubuntu， PC 操作系统 Redhat Linux 9.0 + minicom + armlinux 开发环境						
实验原理 <ol style="list-style-type: none">交叉编译环境<p>原本没有了解过交叉编译和交叉编译环境，但是通过本次实验和资料搜索明白了。交叉编译环境是指在一种计算机环境中运行的编译程序，能编译出在另外一种环境下运行的代码。这个编译过程就叫交叉编译。</p>本次为什么需要交叉编译环境？<p>嵌入式系统通常使用与开发主机不同的架构。因此，需要使用交叉编译器来编译出能够在目标平台上运行的代码。当需要将软件移植到其他平台时，可以使用交叉编译器来避免重新开发。嵌入式系统通常使用与开发主机不同的架构。Tiny4412 开发板采用三星 Exynos4412 四核处理器，该处理器采用 Cortex-A9 架构，Cortex-A9 架构是一款由 ARM 公司开发的 32 位精简指令集处理器架构，因此，需要使用交叉编译器来编译出能够在目标平台上运行的代码。</p>为什么要使用 minicom<p>我们编译完成需要将编译好能够直接运行的文件交付给嵌入式设备，所以我们需要与嵌入式设备通信，而 minicom 就可以负责与嵌入式系统通信，不仅如此我们还可以使用 minicom 进行串口调试，给嵌入式设备发送指令，让嵌入式设备运行我们指定的可以运行的文件。</p>Makfile 有什么作用<p>Makefile 是一个用于自动化编译和链接的文本文件。它可以定义一组规则，告诉计算机如何编译和链接源代码文件，生成目标文件或可执行文件。Makefile 可以将代码的编译规则与代码本身分离，使代码更加易于维护。Makefile 本身只是一个文本文件，它定义了编译和链接</p>						

的规则，但并不会执行具体的编译和链接操作。实际的编译和链接操作是由编译器和链接器完成的。所以我们本次实验编译一些文件时需要用一些 `makefile` 的规则和指令来控制文件的编译和链接。

5. tar 指令的使用方法

`tar` 指令是 Linux 和 macOS 操作系统中用于创建、管理和提取压缩包的命令行工具。它可以将多个文件或目录打包成一个压缩文件，以便于存储、传输和管理。本次实验需要用到的解压缩的操作所以要学习 `tar` 指令的基本用法。

tar 指令的基本语法：

```
tar [选项...] [文件或目录...]
```

常用选项：

- `-c`：创建压缩包
- `-x`：提取压缩包
- `-t`：列出压缩包中的内容
- `-v`：显示详细信息
- `-z`：使用 `gzip` 压缩算法压缩文件
- `-j`：使用 `bzip2` 压缩算法压缩文件
- `-f`：指定压缩包文件名

示例：

- 创建一个 `gzip` 压缩的 `tar` 归档文件

```
tar -czvf archive.tar.gz file1 file2 dir1
```
- 从归档文件中解压缩

```
tar -xvf archive.tar
tar -xzvf archive.tar.gz
```

实验内容：

一、配置交叉编译环境

1. 解压编译器文件

首先进入 `root` 环境 `sudo su` ，输入密码为 123456。

```
oseasy@PC05:~/examples-tiny4412-20131010/examples/adc-test$ sudo su
[sudo] oseasy 的密码：
root@PC05:/home/oseasy/examples-tiny4412-20131010/examples/adc-test#
```

然后使用 `tar` 指令解压编译器文件到指定目录（根目录）下，由于压缩包里的路径已经被老师提前操作好了，所以解压在根目录下，就会直接放到对应的目录下（`/opt /FriendlyARM /toolschain/4.5.1/`）

2. 配置环境变量

(1) 使用 `gedit` 编辑环境变量的文件

```
:/home/oseasy# gedit ~/.bashrc
```

(2) 打开后是这样的

```
打开(O)  .bashrc 保存(S)
64 *)
65 ;;
66 esac
67
68 # enable color support of ls and also add handy aliases
69 if [ -x /usr/bin/dircolors ]; then
70     test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors -b)"
71     alias ls='ls --color=auto'
72     #alias dir='dir --color=auto'
73     #alias vdir='vdir --color=auto'
74
75     alias grep='grep --color=auto'
76     alias fgrep='fgrep --color=auto'
77     alias egrep='egrep --color=auto'
78 fi
79
80 # some more ls aliases
81 alias ll='ls -alF'
82 alias la='ls -A'
83 alias l='ls -CF'
84
85 # Alias definitions.
86 # You may want to put all your additions into a separate file like
87 # ~/.bash_aliases, instead of adding them here directly.
88 # See /usr/share/doc/bash-doc/examples in the bash-doc package.
89
90 if [ -f ~/.bash_aliases ]; then
91     . ~/.bash_aliases
92 fi
93
94 # enable programmable completion features (you don't need to enable
95 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
96 # sources /etc/bash.bashrc).
97 #if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
98 #    . /etc/bash_completion
99 #fi
100 export PATH=$PATH:/home/oseasy/bin
```

(3) 在最后一行加入 `export PATH=$PATH:/opt/FriendlyARM/toolschain/4.5.1/bin`

```
打开(O)  .bashrc 保存(S)
65 ;;
66 esac
67
68 # enable color support of ls and also add handy aliases
69 if [ -x /usr/bin/dircolors ]; then
70     test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors -b)"
71     alias ls='ls --color=auto'
72     #alias dir='dir --color=auto'
73     #alias vdir='vdir --color=auto'
74
75     alias grep='grep --color=auto'
76     alias fgrep='fgrep --color=auto'
77     alias egrep='egrep --color=auto'
78 fi
79
80 # some more ls aliases
81 alias ll='ls -alF'
82 alias la='ls -A'
83 alias l='ls -CF'
84
85 # Alias definitions.
86 # You may want to put all your additions into a separate file like
87 # ~/.bash_aliases, instead of adding them here directly.
88 # See /usr/share/doc/bash-doc/examples in the bash-doc package.
89
90 if [ -f ~/.bash_aliases ]; then
91     . ~/.bash_aliases
92 fi
93
94 # enable programmable completion features (you don't need to enable
95 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
96 # sources /etc/bash.bashrc).
97 #if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
98 #    . /etc/bash_completion
99 #fi
100
101 export PATH=$PATH:/opt/TuxamitoSoftToolchains/arm-arm1176jzfssf-linux-gnueabi/gcc-4.6.4/bin
```

(4) 使用 `source ~/.bashrc` 更新环境变量，`echo $PATH` 显示编译器逻辑则正确；

```
:/home/oseasy# gedit ~/.bashrc
```

```
(gedit:10434): Tepl-WARNING **: 20:07:37.525: GVfs metadata is not supported. Fallback to TeplMetadataManager. Either GVfs is not correctly installed or GVfs metadata are not supported on this platform. In the latter case, you should configure Tepl with --disable-gvfs-metadata.
export PATH=$PATH:/opt/TuxamitoSoftToolchains/arm-arm1176jzfssf-linux-gnueabi/gcc-4.6.4/bin
^Z
[1]+ 已停止      gedit ~/.bashrc
```

(5) 使用 `arm-linux-gcc -v` 显示交叉编译器版本则说明配置正确。

```
gcc-4.6.4/libexec/gcc/arm-arm1176jzfssf-linux-gnueabi/4.6.4/lto-wrapper
Target: arm-arm1176jzfssf-linux-gnueabi
Configured with: /work/build/ldir/src/gcc-4.6.4/configure --build=x86_64-build_unk
nown-linux-gnu --host=x86_64-build_unknown-linux-gnu --target=arm-arm1176jzfssf-
linux-gnueabi --prefix=/opt/TuxamitoSoftToolchains/arm-arm1176jzfssf-linux-gnuea
bi/gcc-4.6.4 --with-sysroot=/opt/TuxamitoSoftToolchains/arm-arm1176jzfssf-linux-
gnueabi/gcc-4.6.4/arm-arm1176jzfssf-linux-gnueabi/sysroot --enable-languages=c,c
++ --with-arch=armv6zk --with-cpu=arm1176jzf-s --with-tune=arm1176jzf-s --with-f
pu=vfp --with-float=softfp --with-pkgversion='crosstool-NG hg+unknown-20130521.1
54019 - tc0002' --disable-sjlj-exceptions --enable-_cxa_atexit --disable-libmud
flap --disable-libgomp --disable-libssp --disable-libquadmath --disable-libquadm
ath-support --with-gmp=/work/build/ldir/arm-arm1176jzfssf-linux-gnueabi/buildtools
--with-mpfr=/work/build/ldir/arm-arm1176jzfssf-linux-gnueabi/buildtools --with-mp
c=/work/build/ldir/arm-arm1176jzfssf-linux-gnueabi/buildtools --with-ppl=/work/bui
lddir/arm-arm1176jzfssf-linux-gnueabi/buildtools --with-cloog=/work/build/ldir/arm
-arm1176jzfssf-linux-gnueabi/buildtools --with-libelf=/work/build/ldir/arm-arm1176
jzfssf-linux-gnueabi/buildtools --with-host-libstdcxx='-static-libgcc -Wl,-Bstat
ic,-lstdc++,-Bdynamic -lm' --enable-threads=posix --enable-target-optspace --wit
hout-long-double-128 --disable-nls --disable-multilib --with-local-prefix=/opt/T
uxamitoSoftToolchains/arm-arm1176jzfssf-linux-gnueabi/gcc-4.6.4/arm-arm1176jzfss
f-linux-gnueabi/sysroot --enable-c99 --enable-long-long
Thread model: posix
gcc version 4.6.4 (crosstool-NG hg+unknown-20130521.154019 - tc0002)
```

至此交叉编译环境配置完毕了。

二、编译代码形成可执行文件

1. 使用 `arm-linux-gcc hello.c -o hello` 编译 `hello` 文件，生成目标文件

```
root@201-42:~/桌面/嵌入式实验# tar -zxvf example*
examples/
examples/vfp-test/
examples/vfp-test/Makefile.orig
examples/vfp-test/Makefile
examples/vfp-test/float-test.soft
examples/vfp-test/float-test.c
examples/vfp-test/float-test.vfp
examples/adc-test/
examples/adc-test/adc-test.c
examples/adc-test/Makefile
examples/led-player/
examples/led-player/Makefile
examples/led-player/led-player.c
examples/buttons/
examples/buttons/buttons_test.c
examples/buttons/Makefile
examples/leds/
examples/leds/led.c
examples/leds/Makefile
examples/pwm/
examples/pwm/Makefile
examples/pwm/pwm_test.c
examples/i2c/
examples/i2c/eeprog.c
examples/i2c/Makefile
examples/i2c/24cXX.h
examples/i2c/24cXX.c
root@201-42:~/桌面/嵌入式实验# ls
about minicom.docx
arm-linux-gcc-4.5.1-v6-vfp-20120301.tgz
examples
examples-tiny4412-20131010.tgz
Makefile
Tiny4412用户手册.pdf
第一次实验说明.docx
第一次实验准备材料.zip
开发环境和调试技术.ppt
嵌入式系统实验1报告_计科190x_姓名_学号.docx
```

2. 依次类推逐条编译 adc-test、buttons、leds、led-player、pwm、vfp-test, 和 ic2。

(1) 编译 adc-test

```
oseasy@PC05:~/examples-tiny4412-20131010/examples/adc-test$ sudo su
[sudo] oseasy 的密码:
root@PC05:/home/oseasy/examples-tiny4412-20131010/examples/adc-test# arm-linux-gcc adc-test.c -o adc-test
```

(2) 编译 buttons

```
root@PC05:/home/oseasy/examples-tiny4412-20131010/examples/buttons# ls
buttons_test.c Makefile
root@PC05:/home/oseasy/examples-tiny4412-20131010/examples/buttons# arm-linux-gcc buttons_test.c -o button_test
```

(3) 使用 make 指令编译 led-player

```
root@PC05:/home/oseasy/examples-tiny4412-20131010/examples# cd led-player/
root@PC05:/home/oseasy/examples-tiny4412-20131010/examples/led-player# make
arm-linux-gcc -Wall -O2 led-player.c -o led-player
```

(4) 使用 make 指令编译 leds

```
root@PC05:/home/oseasy/examples-tiny4412-20131010/examples# cd leds
root@PC05:/home/oseasy/examples-tiny4412-20131010/examples/leds# make
```

(1) 使用 make 指令编译 pwm

```
arm-linux-gcc -Wall -O2 pwm_test.c -o pwm_test
```

(5) 使用 make 指令编译 vfp-test

```
root@PC05:/home/oseasy/examples-tiny4412-20131010/examples/vfp-test# make
make: 对“all”无需做任何事。
root@PC05:/home/oseasy/examples-tiny4412-20131010/examples/vfp-test# make
make: 对“all”无需做任何事。
root@PC05:/home/oseasy/examples-tiny4412-20131010/examples/vfp-test# ls
float-test.c float-test.soft float-test.vfp Makefile Makefile.orig
root@PC05:/home/oseasy/examples-tiny4412-20131010/examples/vfp-test# make
make: 对“all”无需做任何事。
```

(6) 使用 make 指令编译 ic2

```
root@PC05:/home/oseasy/examples-tiny4412-20131010/examples/i2c# ls
24cXX.c 24cXX.h eeprog.c Makefile
root@PC05:/home/oseasy/examples-tiny4412-20131010/examples/i2c# make
arm-linux-gcc -Wall -O2 -c -o eeprog.o eeprog.c
arm-linux-gcc -Wall -O2 -c -o 24cXX.o 24cXX.c
arm-linux-gcc -Wall -O2 eeprog.o 24cXX.o -o i2c
```

三、Minicom 传输可执行文件

1. 修改 minicom 配置

(2) 输入 minicom -s 进入 minicom 配置

```

Welcome to minicom 2.7.1

OPTIONS: I18n
Compiled on Dec 23 2019, 02:06:26.
Port /dev/ttyS0, 03:06:47

Press CTRL-A Z for help on special keys

+-----[configuration]-----+
| Filenames and paths          |
| File transfer protocols     |
| Serial port setup           |
| Modem and dialing           |
| Screen and keyboard         |
| Save setup as dfl           |
| Save setup as..             |
| Exit                         |
+-----+

```

(3) 配置同步文件夹

```

Welcome to minicom 2.7.1

OPTIONS: I18n
Comp+-----+
Port| A - Download directory : /home/oseasy
    | B - Upload directory  : /home/oseasy
Pres| C - Script directory  :
    | D - Script program    : /bin/bash
    | E - Kermit program    :
    | F - Logging options
    |
    | Change which setting? 
+-----+
    | Screen and keyboard
    | Save setup as dfl
    | Save setup as..
    | Exit
+-----+

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | ttyS0

```

(4) 配置传输协议 修改 serial device 和波特率，关闭硬件控制流；

```

Welcome to minicom 2.7.1

OPTI+-----+
Comp| A - Serial Device      : /dev/ttyS0
Port| B - Lockfile Location  : /var/lock
    | C - Callin Program    :
Pres| D - Callout Program   :
    | E - Bps/Par/Bits     : 115200 8N1
    | F - Hardware Flow Control : No
    | G - Software Flow Control : No
    |
    | Change which setting? 
+-----+
    | Screen and keyboard
    | Save setup as dfl
    | Save setup as..
    | Exit
+-----+

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | ttyS0

```

2. 重启开发板，更新配置

```
[ 8.345000] FriendlyARM http://www.arm9.net
[ 8.470000] asoc: wm8960-hifi <-> samsung-i2s.0 mapping ok
[ 8.470000] preallocate_idma_buffer: VA-f0dc0000 PA-2058000 131072bytes
[ 8.470000] asoc: wm8960-hifi <-> samsung-i2s.4 mapping ok

[ 8.540000] rtw driver version=v3.4.4_4749.20121105
[ 8.540000] Build at: Aug 14 2013 17:16:01
[ 8.545000] usbcore: registered new interface driver rtl8192cu
[ 8.555000] Compat-drivers backport release: compat-drivers-v3.8.3-2-snpu
[ 8.555000] Backport based on linux-stable.git v3.8.3
[ 8.555000] compat.git: linux-stable.git
[ 8.580000] cfg80211: Calling CRDA to update world regulatory domain
[ 8.635000] usbcore: registered new interface driver ath9k_htc
[ 8.655000] usbcore: registered new interface driver rt73usb
[ 8.670000] usbcore: registered new interface driver rt2800usb
[ 8.685000] usbcore: registered new interface driver zd1211rw
[ 8.700000] libertas_sdio: Libertas SDIO driver
[ 8.700000] libertas_sdio: Copyright Pierre Ossman
Try to bring eth0 interface up.....[ 8.725000] dm9620 1-2.2:1.0: eth0: linkn
[ 8.730000] link_reset() speed: 10 duplex: 0
[ 8.730000] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
Done

Please press Enter to activate this console. [ 8.770000] jpeg_m2m_open: canne
```

3. 使用 minicom 将目标文件传至开发板

(1) 找到编译文件所在目录，使用 zmodem 将编译生成的目标文件传输至开发板。

```
Welcome to minicom 2.7.1

OPTIONS: I18n
Compiled on Dec 23 2019, 02:06:26.
Port /dev/ttyS0, 03:06:47

+-[Upload]--+
Press CTRL-A Z for help on sp| zmodem |
| ymodem |
| xmodem |
| kermit |
| ascii |
+-----+

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | ttyS0
```

(2) 按空格选择文件，并发送

```
root@PC05: /home/oseasy/桌面
hello,FriendlyARM!
[r+-----[Select one or more files for upload]-----+
24|Directory: /home/oseasy/upload
Ma| [..]
_n| 24cXX.o
a.| adc-test
ad| button_test
ar| eeprog.o
ar| hello
ar| i2c
bi| led
bu| led-player
bu| pwm_test
bu|
da|
de|
ee|
et|
fa| ( Escape to exit, Space to tag )
fl+-----+
float-test.soft      pwm_test      www
fl      [Goto] [Prev] [Show] [Tag] [Untag] [Okay]
fwq      sbin      x??hello?7816
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | ttyS0
```

四、Minicom 串口调试可执行文件

1. 开发板上运行代码，查看结果

(1) 查看上传是否成功 进入嵌入式设备终端 查看文件

```
[root@FriendlyARM /]# ls
Makefile          hhh          server.asm
_new.             home        service.asm
a.out            led         share
adc-test         led-player  sys
arm-udp          led1        term
arm-udptalk      lib         test
arm-udptalkZK   linuxrc    test.sh
bin              lost+found  test.sh~
button           mnt        tiny4412_buttons.ko
buttons_test     opt        tiny4412_leds.ko
data             p          tiny4412_pwm.ko
dev              proc       tmp
etc              pthread    udisk
fa-network-service pthread.o  udptalk.asm
float-test       pwm       usr
float-test.soft  pwm_test  var
float-test.vfp   root     www
fwq              sbin     x?
globalmem.ko    sdcard   x??hello?7816
hello           serve.asm
```

(2) 运行可执行文件 hello

```
[root@FriendlyARM /]# ./hello
hello,FriendlyARM!
```

(3) 运行可执行文件 adc-test


```
AIN0 2088
AIN1 1950
AIN4 2009
AIN5 1992
AIN6 1900
AIN7 2032
```

```
□
```

(4) 运行可执行文件 buttons

```
[root@FriendlyARM /]# ./button_test
key 3 is down
key 3 is up
key 3 is down
key 3 is up
key 2 is down
key 2 is up
key 1 is down
key 1 is up
key 4 is down
key 4 is up
key 3 is down
key 3 is up
key 2 is down
key 2 is up
key 1 is down
key 1 is up
```

```
□
```

(5) 运行可执行文件 led-player

```
[root@FriendlyARM /]# ./led-player
[ 464.225000] Left
[ 468.510000] Down
[ 468.845000] Tap
[ 469.500000] Left
```

```
□
```



(6) 运行可执行文件 led

```
AIN0 2088
AIN1 1950
AIN4 2009
AIN5 1992
AIN6 1900
AIN7 2032
□
```

(7) 运行可执行文件 pwm

```
[root@FriendlyARM /]# ./pwm_test
BUZZER TEST ( PWM Control )
Press +/- to increase/reduce the frequency of the BUZZER
Press 'ESC' key to Exit this program

      Freq = 1000

Superboot-4412 V2.6.3-QT(20190722) by FriendlyARM
Booting from eMMC
```

(8) 运行可执行文件 vfp-test

```
[root@FriendlyARM /]# ./float-test.soft
A:      84.019      39.438      78.310      79.844      91.165      19.77
B:       9.548      12.577       7.296      10.268      19.045      18.39
R:  1686365.875  1679191.500  1685225.875  1685731.000  1688354.625  1676391.05
S:  1451627765.012017
E:  1451627765.125954
E:           0.113937
[root@FriendlyARM /]# ./float-test.vfp
A:      84.019      39.438      78.310      79.844      91.165      19.77
B:       9.548      12.577       7.296      10.268      19.045      18.39
R:  1686365.875  1679191.500  1685225.875  1685731.000  1688354.625  1676391.05
S:  1451627771.784625
E:  1451627771.800818
E:           0.016193
```

(9) 运行可执行文件 i2c

```
[root@FriendlyARM /]# ./i2c
I2C-24C08(256 bytes) Read/Write Program, ONLY FOR TEST!
Base on 'eeprog' by Stefano Barbato (http://codesink.org/eeprog.html)
FriendlyARM Computer Tech. 2009

[line 94]
```

实验结果与总结：

一、遇到的问题

1. 忘记使用 source 指令

更改了环境变量后，但是一直不能调用 arm-gcc 的编译器，后来发现是没有运行 source 重新加载环境变量。

为什么要用 source 指令呢？

在 Linux 中，更改环境变量后需要运行 source ~/.bashrc 指令，这是因为环境变量是在 shell 启动时加载的，如果在 shell 运行过程中更改了环境变量，那么新的环境变量不会生效，需要重新加载环境变量才能使更改生效。

具体来说，当登录 Linux 系统时，系统会启动一个 shell，并加载 /etc/profile 和 ~/.bashrc 等配置文件。这些配置文件中包含了环境变量的设置。

当在 shell 中运行命令时，shell 会使用这些环境变量。

如果在 shell 运行过程中更改了环境变量，那么新的环境变量不会生效，这是因为：shell 已经启动，并加载了环境变量，更改环境变量只会影响当前 shell 实例，不会影响其他 shell 实例。要使更改的环境变量生效，需要重新加载环境变量，可以通过退出并重新登录系统来重新加载环境变量。

也可以运行 source ~/.bashrc 指令来重新加载环境变量。

source ~/.bashrc 指令的作用是重新加载 ~/.bashrc 配置文件，~/.bashrc 配置文件中包含了环境变量的设置，因此重新加载 ~/.bashrc 配置文件可以使更改的环境变量生效。

2. 编译时报错

```
root@PC05:/home/oseasy/examples-tiny4412-20131010/examples/i2c# arm-linux-gcc eeprog.c
-o eeprog
/tmp/ccJpIMRj.o: In function `read_from_eeprom':
eeprog.c:(.text+0x140): undefined reference to `eeprom_read_byte'
/tmp/ccJpIMRj.o: In function `write_to_eeprom':
eeprog.c:(.text+0x2d4): undefined reference to `eeprom_write_byte'
/tmp/ccJpIMRj.o: In function `main':
eeprog.c:(.text+0x410): undefined reference to `eeprom_open'
eeprog.c:(.text+0x4c4): undefined reference to `eeprom_close'
collect2: ld returned 1 exit status
root@PC05:/home/oseasy/examples-tiny4412-20131010/examples/i2c# arm-linux-gcc 24cXX.c -o
24cXX
/opt/TuxamitoSoftToolchains/arm-arm1176jzfssf-linux-gnueabi/gcc-4.6.4/arm-arm1176jzfssf-
linux-gnueabi/sysroot/usr/lib/crt1.o: In function `_start':
:(.text+0x34): undefined reference to `main'
collect2: ld returned 1 exit status
```

编译文件的就是老师给出的命令编译,但是报错,为什么呢? 因为这里的文件是一个项目,文件之间是有相互依赖的,要有特殊的编译规则来编译的,所以用 `makefile` 的指令来编译整个项目。

```
root@PC05:/home/oseasy/examples-tiny4412-20131010/examples/i2c# ls
24cXX.c 24cXX.h eeprog.c Makefile
root@PC05:/home/oseasy/examples-tiny4412-20131010/examples/i2c# make
arm-linux-gcc -Wall -O2 -c -o eeprog.o eeprog.c
arm-linux-gcc -Wall -O2 -c -o 24cXX.o 24cXX.c
arm-linux-gcc -Wall -O2 eeprog.o 24cXX.o -o i2c
```

3. 上传文件失败

重启开发板,连接紧线就好了,同时也有可能是线的问题需要,检查线是不是可以用的,也有可能是开发板的问题。

二、实验结果与总结

实验结果与总结:

本次实验成功完成了嵌入式开发板 Tiny4412 的基本使用,包括配置交叉编译环境、使用交叉编译器编译目标代码、通过 `minicom` 传输并在开发板上运行可执行文件等关键步骤。

通过实践操作,深化了对 Linux 系统命令、开发工具使用的理解,如 `tar` 解压缩、`gedit` 编辑器、`make` 工具、`arm-linux-gcc` 交叉编译器、`minicom` 终端通信等。

亲身体会了嵌入式开发的流程,加深了对嵌入式系统与一般计算机系统差异的认识,如需要交叉编译环境、资源受限等。

实验过程中遇到了一些问题,如环境变量配置、编译错误、文件传输失败等,通过查阅资料、思考分析和尝试解决,锻炼了动手能力和独立解决问题的能力。

总的来说,本次实验是一次很好的嵌入式开发入门实践,既学习了相关理论知识,又上手动手操作,对嵌入式开发有了初步认识和兴趣,但是这次没有参与很多代码的编写,对于嵌入式的代码编写还是有很大的学习空间的,以后要不断加强。

总结感悟:

从这次实验我深刻体会到,嵌入式开发与传统软件开发有着很大区别,需要对硬件系统有更多了解,开发环境配置、交叉编译等都是独有的环节。嵌入式开发的许多细节都需要亲自操作实践才能领会,单纯理论学习是不够的。

另一方面,无论是传统软件还是嵌入式软件开发,遇到问题能够独立查阅资料、分析思考、尝试解决是非常重要的能力。实践操作过程中遇到的各种问题,都锻炼了我的这方面能力。

最后,通过本次实验我对嵌入式系统开发产生了浓厚的兴趣。嵌入式系统无处不在,在物联网、智能硬件等热门领域发挥着重要作用。我期待在后续的嵌入式课程中能继续深入学习,为将来从事相关工作打下坚实基础。