

# 浙江大学实验报告

专业： 计算机科学与技术

姓名： 陆 洲

学号： 3120101973

日期： 2015/06/12

地点： 曹西 501

课程名称： 嵌入式系统 指导老师： 蔡铭 成绩：

实验名称： 夏任务 103：做一个网络时钟 实验类型： 基础实验 同组学生姓名： 陈以珊、董希

## 一、实验目的和要求

- (1) 掌握通过 GPIO 控制七段数码管或 LCD 的方法；
- (2) 掌握通过 ntp 客户端获取时间的办法。

## 二、实验内容和原理

做一个网络时钟，通过 ntp 从一个互联网服务器得到时间，实时在 7 段数码管或 LCD 上显示时间。

## 三、主要仪器设备

- (1) pcDuino v2 板一块；
- (2) 5V/1A 电源一个；
- (3) microUSB 线一根；
- (4) 面包板一块；
- (5) 两位 7 段数码管（共阳）一颗；
- (6) 8 段 LED 柱状显示器一颗；
- (7)  $360\Omega$  1/8W 电阻 8 颗；
- (8) 10k 1/8W 电阻 2 颗；
- (9) 按钮两个；
- (10) 面包线若干。

以下为自备（可选）器材：

- (1) PC (Windows/Mac OS/Linux) 一台；
- (2) USB-TTL 串口线一根 (FT232RL 芯片或 PL2303 芯片)；
- (3) 以太网线一根（可能还需要路由器等）；
- (4) 1602 LCD 一块（带配套的 5k 微调电阻）；
- (5) 9g 伺服电机一只；
- (6) 8x8 LED 矩阵一个；
- (7) 8 颗各色 LED (5mm)。

软件：

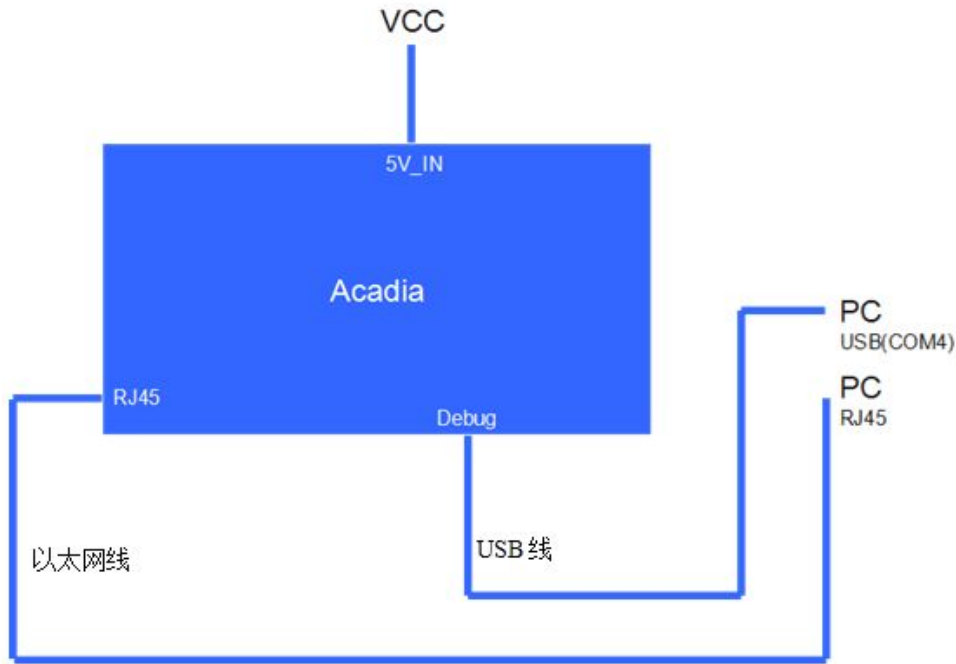
- (1) 编译软件；
- (2) Fritzing。

## 四、操作方法和实验步骤

- (1) 安装 ntp 客户端；
- (2) 编写程序通过 ntp 客户端获取时间并在七段数码管或 LCD 上显示；
- (3) 进行验证。

## 五、实验结论记录和分析

- (1) 画出你所实际实施的连接示意图；



**(2) 给出所用的器材的列表;**

- (1) Acadia 主板一块;
- (2) 5V/1A 电源一个;
- (3) microUSB 线一根;
- (4) 面包板一块;
- (5) 两位 7 段数码管 (共阳) 一颗;
- (6) 8 段 LED 柱状显示器一颗;
- (7)  $360\Omega$  1/8W 电阻 8 颗;
- (8) 10k 1/8W 电阻 2 颗;
- (9) 按钮两个;
- (10) 面包线若干。

以下为自备 (可选) 器材:

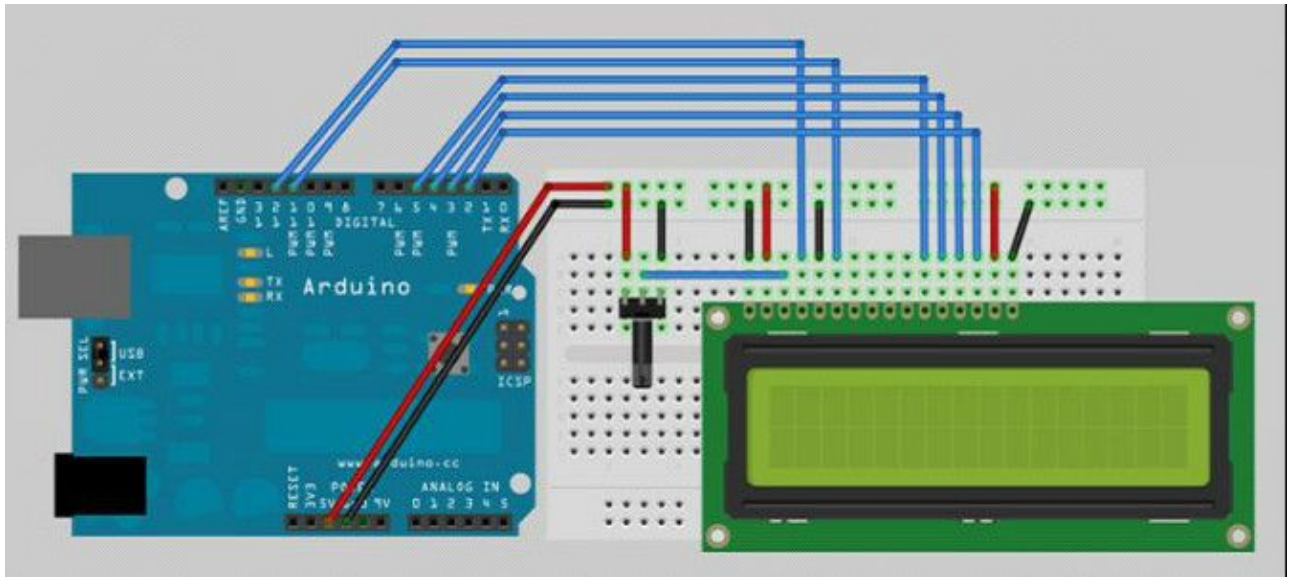
- (1) PC (Windows/Mac OS/Linux) 一台;
- (2) USB-TTL 串口线一根 (FT232RL 芯片或 PL2303 芯片);
- (3) 以太网线一根 (可能还需要路由器等);
- (4) 1602 LCD 一块 (带配套的 5k 微调电阻);
- (5) 9g 伺服电机一只;
- (6) 8x8 LED 矩阵一个;
- (7) 8 颗各色 LED (5mm)。

软件:

- (1) 编译软件;
- (2) Fritzing。

**(3) 用 Fritzing 画出外部设备的连线图, 附实物照片;**

因为没有 pcduno 的板子, 所以用 arduino 来代替, 他们的 gpio 引脚编号是相通的。



(4) 描述所做的实验步骤，给出各步操作的命令和结果；

(1) 安装 ntp 客户端；

1. 首先，在 Acadia 中安装 ntpdate 工具。

```
sudo apt-get install ntpdate
```

```
root@Acadia:~# sudo apt-get install ntpdate
Reading package lists... Done
Building dependency tree
Reading state information... Done
ntpdate is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

2. 使用命令 tzselect 来修改时区，可以选择洲、国、以及国家里的特定时区。

```
Tzselect
```

```
root@Acadia:~# tzselect
Please identify a location so that time zone rules can be set correctly.
Please select a continent or ocean.
 1) Africa
 2) Americas
 3) Antarctica
 4) Arctic Ocean
 5) Asia
 6) Atlantic Ocean
 7) Australia
 8) Europe
 9) Indian Ocean
10) Pacific Ocean
11) none - I want to specify the time zone using the Posix TZ format.
#? 5
```

```
Please select a country.
 1) Afghanistan      18) Israel           35) Palestine
 2) Armenia           19) Japan            36) Philippines
 3) Azerbaijan        20) Jordan           37) Qatar
 4) Bahrain           21) Kazakhstan       38) Russia
 5) Bangladesh        22) Korea (North)    39) Saudi Arabia
 6) Bhutan            23) Korea (South)    40) Singapore
 7) Brunei            24) Kuwait           41) Sri Lanka
 8) Cambodia          25) Kyrgyzstan       42) Syria
 9) China             26) Laos             43) Taiwan
10) Cyprus            27) Lebanon          44) Tajikistan
11) East Timor        28) Macau            45) Thailand
12) Georgia           29) Malaysia         46) Turkmenistan
13) Hong Kong         30) Mongolia         47) United Arab Emirates
14) India             31) Myanmar (Burma)  48) Uzbekistan
15) Indonesia         32) Nepal            49) Vietnam
16) Iran              33) Oman             50) Yemen
17) Iraq              34) Pakistan

#? 9
```

```
Please select one of the following time zone regions.
1) east China - Beijing, Guangdong, Shanghai, etc.
2) Heilongjiang (except Mohe), Jilin
3) central China - Sichuan, Yunnan, Guangxi, Shaanxi, Guizhou, etc.
4) most of Tibet & Xinjiang
5) west Tibet & Xinjiang

#? 1
```

3. 设定时区成功。时区已经更改为亚洲/上海。

```
You can make this change permanent for yourself by appending the line
    TZ='Asia/Shanghai'; export TZ
to the file '.profile' in your home directory; then log out and log in again.

Here is that TZ value again, this time on standard output so that you
can use the /usr/bin/tzselect command in shell scripts:
Asia/Shanghai
```

4. 使用命令来联网获取当前时间。

`ntpdate cn.pool.ntp.org`

```
root@Acadia:~# ntpdate cn.pool.ntp.org
11 Jun 22:32:57 ntpdate[3172]: adjust time server 202.112.31.197 offset -0.03766
5 sec
```

(2) 编写程序通过 ntp 客户端获取时间并在七段数码管或 LCD 上显示;

1. 连接 LCD1602 与 Acadia

LCD1602 液晶屏提供了 16 列 x2 行的 ASCII 字符显示能力，工作电压 5V，提供 4 位数据与 8 位数据两种工作模式，Raspberry Pi 的 GPIO 口数量很有限，所以我们使用 4 位数据模式。

脚位连接说明：

引脚编号	功能
1	Vss 接地 (0V)
2	Vdd 电源 (+5V)
3	Vo 或与 Vee - 对比度调整(0-5V), 可接一颗 1K 的电阻或接一个 5K 的电位器
4	RS Register Select: 参数: {1: D0 - D7 设置为 1 时当作资料解释 0: D0 - D7 设置为 0 时当作指令解释}

5	R/W Read/Write mode: 参数: {1: 从 LCD 读取资料 0: 写资料到 LCD(一般情况下很少从 LCD 读取资料, 把这个脚接地可省 I/O 脚位)}
6	E Enable
7	D0 Bit 0 LSB
8	D1 Bit 1
9	D2 Bit 2
10	D3 Bit 3
11	D4 Bit 4
12	D5 Bit 5
13	D6 Bit 6
14	D7 Bit 7 MSB
15	A+ 背光(串联一个 1K 的电阻接地或串联一个 5K 的电位器调整背光)
16	K- 背光(GND)

连接电路图见上。

2. 安装 python 环境。

```
root@Acadia:~# git clone https://github.com/pcduino/python-pcduino
Cloning into 'python-pcduino'...
remote: Counting objects: 123, done.
remote: Total 123 (delta 0), reused 0 (delta 0), pack-reused 123
Receiving objects: 100% (123/123), 28.45 KiB, done.
Resolving deltas: 100% (37/37), done.
```

3. 驱动 LCD1602, 下载 github 已有的脚本, 并根据 acadia 的引脚连接做一点修改。

`wget https://raw.githubusercontent.com/adafruit/master/Adafruit_CharLCD/Adafruit_CharLCD.py`

4. 然后自己写一个显示时钟的脚本。

`nano clock.py`

```
#!/usr/bin/python
from Adafruit_CharLCD import Adafruit_CharLCD
from subprocess import *
from time import sleep, strftime
from datetime import datetime
lcd = Adafruit_CharLCD()
lcd.begin(16,1)
while 1:
    lcd.clear()
    lcd.message(datetime.now().strftime('%Y %b %d %a\n %H : %M : %S'))
    sleep(1)
```

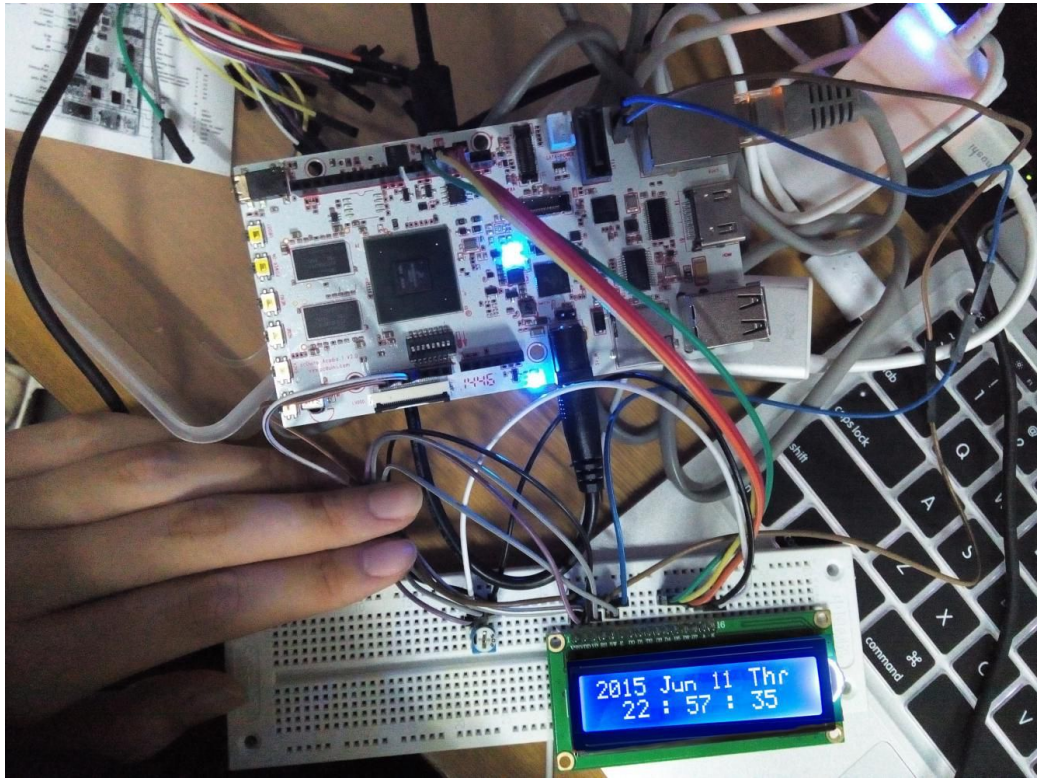
执行脚本

`sudo python clock.py`

```
root@Acadia:~# sudo python clock.py
```

(3) 进行验证。





## 六、讨论、心得

本次实验主要就是下载 ntp 客户端，驱动 LCD1602 等部分，其他的还是比较简单的。第一次写 python 脚本，有点不熟悉，但是基本上来说只要逻辑正确，再修改一下语法就可以了。Python 确实是很方便的语言。