Tinkering with Iris-3000 aka CU776

PBX in a Flash forum
prepared by Iris3000

July 5, 2012

Contents

1 Special thanks

1.1 A man who made my day

This package and document would not be possible without Joshoa who made first Firmware Update package. He was a man who provided sufficient information to get me started.

1.2 A paitient wife

My very special thanks goes to my wife who has enough patience to be around when I tinker with hardware and software. And her support in form of beatiful smile and honest laud laugh what makes me want to kiss her.

2 About the phone

2.1 Who is manufacturer

The true name of the phone is *CU776* which is produced by *UMEC* – "Universal Microelectronics co." with headquarters in Taiwan. The phone listed on their website UMEC VoIP CU-776 phone.

The phone CU776 is listed on website where we can see that *UMEC* produces their own VoIP software or at least GUI is different.

2.2 CU776 Phone's specification

The phone has next specification

```
Screen: 7 inch (16:9)
Mainboard: MX27
Memory: 126 MB
Processor: ARM926Ej-S rev 4 (v51)
Hardware: Freescale i.MX27ADS
Revision: 27021
Flash Chip: 125 MB
```

2.3 Phone with MTD

```
mx27# cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00040000 00010000 ''RedBoot''
mtd1: 001b0000 00010000 ''kernel''
mtd2: 0000f000 00010000 ''FIS directory''
mtd3: 00001000 00010000 ''RedBoot config''
mtd4: 08000000 00020000 ''rootfs''

mx27# cat /proc/cmdline
noinitrd console=ttymxc0,115200 root=/dev/mtdblock4 rootfstype=jffs2 \
init=linuxrc ip=none fec_mac=00:04:9f:00:82:4c video=mxcfb:TV-NTSC

mx27# cat /proc/version
Linux version 0.1.12.10 2.6.22.6 (root@localhost.localdomain) \
(gcc version 4.1.2) #453 PREEMPT Fri Oct 3 15:58:45 CST 2008

mx27# cat /proc/filesystems
nodev   sysfs
nodev   rootfs
nodev   bdev
nodev   proc
nodev   sockfs
nodev   usbfs
nodev   pipefs
nodev   anon_inodefs
nodev   futexfs
nodev   tmpfs
nodev   inotifyfs
nodev   devpts
        ext3
nodev   ramfs
        msdos
        vfat
nodev   jffs2

mx27# cat /proc/cpuinfo
Processor       : ARM926EJ-S rev 4 (v5l)
BogoMIPS        : 199.47
Features        : swp half thumb fastmult edsp java
CPU implementer : 0x41
CPU architecture: 5TEJ
CPU variant     : 0x0
```

```
CPU part        : 0x926
CPU revision    : 4
Cache type      : write-back
Cache clean     : cp15 c7 ops
Cache lockdown  : format C
Cache format    : Harvard
I size          : 16384
I assoc         : 4
I line length   : 32
I sets          : 128
D size          : 16384
D assoc         : 4
D line length   : 32
D sets          : 128

Hardware        : Freescale i.MX27ADS
Revision        : 27021
Serial          : 0000000000000000

mx27# cat /proc/meminfo
MemTotal:       126736 kB
MemFree:         28668 kB
Buffers:             0 kB
Cached:          41312 kB
SwapCached:          0 kB
Active:          58384 kB
Inactive:        30764 kB
SwapTotal:           0 kB
SwapFree:            0 kB
Dirty:               0 kB
Writeback:           0 kB
AnonPages:       47840 kB
Mapped:          20356 kB
Slab:             4552 kB
SReclaimable:      912 kB
SUnreclaim:       3640 kB
PageTables:        468 kB
NFS_Unstable:        0 kB
Bounce:              0 kB
CommitLimit:     63368 kB
Committed_AS:   250492 kB
VmallocTotal:   122880 kB
VmallocUsed:     33160 kB
```

2.4 Phones with SDA

```
mx27# cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00040000 00010000 ''RedBoot''
mtd1: 001b0000 00010000 ''kernel''
mtd2: 0000f000 00010000 ''FIS directory''
mtd3: 00001000 00010000 ''RedBoot config''

mx27# cat /proc/cmdline
noinitrd console=ttymxc0,115200 root=/dev/sda1 rootfstype=ext3 \
init=linuxrc ip=none fec_mac=00:04:9f:00:82:4c video=mxcfb:TV-NTSC

mx27# cat /proc/version
Linux version 1.1.16.10 2.6.22.6 (root@localhost.localdomain) \
(gcc version 4.1.2) #502 PREEMPT Fri Mar 6 23:08:01 CST 2009

mx27# cat /proc/filesystems
nodev   sysfs
nodev   rootfs
nodev   bdev
nodev   proc
nodev   sockfs
nodev   usbfs
nodev   pipefs
nodev   anon_inodefs
nodev   futexfs
nodev   tmpfs
nodev   inotifyfs
nodev   devpts
        ext3
nodev   ramfs
        msdos
        vfat
nodev   nfs
nodev   jffs2
nodev   rpc_pipefs

mx27# cat /proc/cpuinfo
Processor       : ARM926EJ-S rev 4 (v5l)
BogoMIPS        : 199.47
Features        : swp half thumb fastmult edsp java
CPU implementer : 0x41
CPU architecture: 5TEJ
```

```
CPU variant     : 0x0
CPU part        : 0x926
CPU revision    : 4
Cache type      : write-back
Cache clean     : cp15 c7 ops
Cache lockdown  : format C
Cache format    : Harvard
I size          : 16384
I assoc         : 4
I line length   : 32
I sets          : 128
D size          : 16384
D assoc         : 4
D line length   : 32
D sets          : 128

Hardware        : Freescale i.MX27ADS
Revision        : 27021
Serial          : 0000000000000000

mx27# cat /proc/meminfo
MemTotal:       126536 kB
MemFree:         25520 kB
Buffers:          5496 kB
Cached:          40664 kB
SwapCached:          0 kB
Active:          62320 kB
Inactive:        30672 kB
SwapTotal:           0 kB
SwapFree:            0 kB
Dirty:              28 kB
Writeback:           0 kB
AnonPages:       46836 kB
Mapped:          19904 kB
Slab:             3632 kB
SReclaimable:     1268 kB
SUnreclaim:       2364 kB
PageTables:        476 kB
NFS_Unstable:        0 kB
Bounce:              0 kB
CommitLimit:     63268 kB
Committed_AS:   249968 kB
VmallocTotal:   122880 kB
VmallocUsed:     33148 kB
```

```
VmallocChunk:    81916 kB
```

3 Tinkering with Iris-3000

3.1 ACN and it's service

One day I was approached by ACN representative with invitation to check presentation on new technologiy and particulary VoIP. It got my attention and I accepted the offer.

The presentation was not too "sparky" but as I was looking for not that expencive way to communicate with my mother who lived in other country I decided to give it a try. My package included Iris-3000 with a plan, internet access through third provider and Samsung Chocolate cell phone with a good plan from same provider as Internet.

The problem started very shortly after signing the contract. And the problem was with internet access. ACN was saying that problem is on provider side and provider was saying that as you signed the contract with ACN then I should resove the issue with them.

After some "fiting" with ACN I came to conclution that if you lucky and everything went throgh smooth rigth away then it might be Ok. But if you got in situation like me then it will be a nightmare until the contract expires. In short I dropped their VoIP and Internet service as soon as it become clear that any problem in future will take too much my time to resolve.

In the end I had Iris-3000 phone which I paid for with no use.

3.2 Iris-3000 without any use

It passed about two years and "Iris-3000" VoIP phone still was sitting in the box under my table for a couple years. One day I decided to "Google" if anything could be done with the phone to make it serve it's purpose. Shortly after I came accross "PBX in a Flash" forum where information stated that it can be done.

3.3 PBX in a Flash - new hope

Through some reading in the forum I came to a conclution that Joshoa would be a right person to address my question. I attempted to contact him and shortly after I got a responce from him. It gave me some head start to work on the phone. Very shortly the phone was updated with other firmware and I got access through network into the Iris-3000 system. I found that the phone could register with my "Asterisk" server but could not place a phone call. New E-Mail was directed to Joshoa and very soon with his help my Iris-3000 was capable to place a call.

At this moment I did not had a chance to test Video calls as I did not have a second video phone. I've attempted to use free software X-Lite which didn't work with the phone. I contacted Joshoa once more to get a confirmation that X-Lite does not work with Iris-3000 phone.

3.4 Second Iris-3000 with a twist

A year latter I dropped into "Value Village" second hand store and discovered that somebody "donated" three Iris-3000 phones. I bought one for $12 dollars to play with it.

Unfortunately same evening I found that something was wrong with the phone. It did not boot properly and on the screen of the phone image of "World Map" with "ACN" logo was staying forever.

I went back to forum in search of information what can be done to solve the issue. This time I came across a post from Joshoa that he figured out how to resolve this issue. I was incouraged to try his new approach.

I went through recommended "Red Boot Fix" but it did not work. I contacted Joshoa once more in search of his "wisdom" and "help". Joshoa has given recomendation to try some variation of commands but it did not help.

At this time I returned into forum. This time I was looking for standard recovery procedure. I followed the procedure by pressing "1+#" and holding simulteniously and turned the phone on. In about 11 seconds I got "World Map" with "ACN" logo and was expecting to see a red progress bar which would indicate that firmware upgrade is in progress. But it did not happen as I would expect.

I played with my router and tried directly connect the phone to cable modem but nothing did work. Then I went back to forum to grab Joshoa's file which he was able to save from the company. I've copied software on Linux computer, configure IP alias for network card and followed standard recovery procedure once more. This time to my surprise the procedure went forward and I got the phone to state where I could access it.

I went through procedure a couple times to play around and found that as Joshoa described during the procedure I can telnet into the phone. This discovery has opened a new oportunities to play with the system. I was able to inspect /etc/rc.d/rc.local file to see how the procedure works. At this time I figured out that I can interrupt the procedure and use my own image to reflash the chip.

I was very close to the point where my phone can be repaired. I decided to look at the chip which presented by Iris-3000 system as /dev/sda disk. I issued a command "fdisk -l" to look at the partions and what I've found gave me an explanation why I could not revive this phone with all my effort before. Somehow partions /dev/sda1 has disappeared and of course the software could not be copied to none existant partition. I've recreated the partions through "fdisk" and went through recover procedure once more. This time it worked without a hitch.

4 Iris-3000 Firmware Update

4.1 Two types of hardware

There are two types of Iris-3000 phone. This phone has Linux installed as OS. The flash chip in Linux is represented as or /dev/sda1 or /dev/mtd/4 devices/partitions.

4.2 How to identify type of hardware

Joshoa identifies the phones with storage device /dev/sda1 as *"HW1'* and the phones with storage device /dev/mtd/4 as *"HW2"*. The phones with /dev/sda1 partition use EXT3 file system and phones which have /dev/mtd/4 partition have JIFFS2 file system.

Due this reason Joshoa provides two different images for two types of hardware.

4.3 Firmware Update

I've inspected Joshoa's script and wrote my own which adds some debugging information in form of messages on Iris-3000 LCD screen what makes much easier to know when reflash process is complete – no more guessing.

The procedure of Iris-3000 Firmware Update is quite simple. You need to download a file, then extract it into root directory of FAT32 formated SD card, insert SD card into SD slot of Iris-3000 VOIP phone and power on the phone.

Firmware Update procedure will start shortly and should take about 10–11 minutes. Once the procedure completed you will see a message on the screen of Iris-3000 that it completed with suggestion to power cycle the phone. You need to "turn off' the phone, eject SD card and "power on" the phone again.

Firmware Update procedure generates logfile.txt which I recommend to inspect to find how long it has taken to flash your phone.

4.4 Access over network – SSH

Once firmware has been updated and your phone requested IP address from DHCP server you can ssh into the phone over network. You can ssh into your phone on port 7022 with root account and password *"1234"*.

In linux you can ssh with next command

```
ssh root@ipofyourphone -p 7022
```

4.5 Factory Reset

You can do factory reset procedure if the need will rise. *"Factory Reset"* can be done by selecting in menu *"Settings-System Settins-Factory Reset"*. You will be asked for a password which is *"7517517"*.

4.6 Root pasword after Factory Reset

*Factory Reset*procedure changes *"root"* password to *"root"*. Please remember than once you go through *"Factory Reset"* procedure your old *"root"* password will be not valid any more.

4.7 VeriCall Web Configuration

The Firmware *20.9.37* does not allow to configure your Iris-3000 through GUI in phone itself. You provided with "VeriCall Web Configuration" interface. The webpage is accessible through URL
`http://voip_phone_ip:8080/resorce/AppWeb/login.esp`.

4.8 Disabling/Enabling VeriCall password

The default password for *VeriCall Web Configuration* interface is unknown. Joshoa has figured out a way to surcumvent the system to allow you login by editing some file. To make this procedure easier for people not familiar with Linux and terminal editors he wrote a coulple scripts (web-off.sh and web-on.sh). I have borrowed the idea and integrated both scripts into one with additional features like help, usage, on/off/status. Just run the script without any parameter to get usage information. Once you will see this piece of information rest can be easily figured out.

Type next command shows usage of the script

```
/web_login.sh
```

Type next command gives help on script's options

```
/web_login.sh help
```

Type next command to disable *"VeriCall"* password

```
/web_login.sh password off
```

5 How to unbrick Iris-3000

5.1 SD Card method

SD Card reflashing procedure is probably most simple but it might not work in some instances. In my particular case I had a phone without /dev/sda1 partition and this method did not work.

The process is as simple as turning your phone *off*, inserting FAT32 formated SD Card into slot of VoIP phone then turning the phone *on*. Waiting some period of time and if you use script prepared by me then once you see a message on the screen of the phone that *reflash* is finished you need to power *off* the phone, eject SD Card and power the phone *on*.

The phone should boot properly. If you use images from Joshoa then root password is *root* and if you use my image then password is *root*.

At this time you can do *Factory Reset* with password *7517517* if you desire to do so. After *Factory Reset* root password is *root*.

Now you should be able to ssh into the phone with next command

```
ssh root@voip_phone_ip -p 7022
```

and disable password for *"VeriCall"* configuration web page which is accessible through url

```
http://voip_phone_ip:8080/resource/AppWeb/login.esp
```

To disable *VeriCall* password in case of Joshoa image issue next command

```
/web-off.sh
```

and in case of my image issue next command

```
/web_login password off
```

Now login into *VeriCall* VoIP configuration web page and make required configuration.

5.2 NFS method #1

NFS method a little bit more trickier as you need NFS server. I must admit that I didn't try it with *Linux Live CD/DVD* although I do not exclude that it could work, in my case I used my own Linux machine.

For this method you will need to create a directory on your *NFS* server */var/umec/code* and add it into */etc/exports* configuration file like next

```
/var/umec/code  192.168.0.0/24(rw,sync,no_subtree_check)
```

NOTE: IP address 192.168.0.0/24 indicates private network I use at my home. Please *adjust* it to you settings.

Once you add record above into */etc/exports* you need instruct your *Linux* system to reexport it again what can be atchived with next command

```
exportfs -ra
```

Then grub file prepared by Joshoa from internet at next address and extract it's content into */var/umec/code* directory. Software can be downloaded by following next link UMEC Software.

Iris-3000 phone preconfigured to communicate with some external server on internet and it would not work in our settings unless we fool the system to believe that our server is *external server* with IP address 8.5.244.17. In *Linux* this can be easily achived with next command

```
/sbin/ipconfig eth0.1 8.5.244.17 netmask 255.255.255.0
route add -net 8.5.244.0 device eth0.1
```

Well we are ready to test our setup. Turn *off* your Iris-3000 then simulteniously press and hold *1+#* and turn your phone. Wait for about 11 second and you should see on LCD screen "Wolrd Map" with "ACN" logo. At this moment you can release *1+#*. Shortly after you will see some red progress bar on LCD screen. At some point the phone will restart and boot into new software.

After reflashing the password is *acnum3c* and you can attempt to *telnet* into the phone with next command

```
telnet voip_phone_ip 23
```

Now the phone can be *upgraded* to more recent version of software *20.9.37* through *SD Upgrade* method.

5.3 NFS method #2

This method is some variation of method #1. In this case you start the update process through *1+#* but once network connection is up, we interrupt recovery process by killing two processes and restore our own image into flash memory.

Once you get *World Map with ACN logo* on the screen in *Linux* terminal as root we issue next command

```
server# nmap voip_phone_ip

Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap/ )
Warning:  You are not root -- using TCP pingscan rather than ICMP
```

```
All 1554 scanned ports on  (192.168.0.77) are: closed

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second

server# nmap voip_phone_ip
..........
server# nmap 192.168.0.77

Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap/ )
Warning:  You are not root -- using TCP pingscan rather than ICMP
Interesting ports on  (192.168.0.77):
(The 1548 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
23/tcp     open        telnet
79/tcp     open        finger
113/tcp    open        auth
513/tcp    open        login
514/tcp    open        shell


Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
server#
```

to see what ports are open. Once we get information that phone has some network ports open we can *telnet* into phone system which in this expample obtained IP address *192.168.0.77* with a command

```
root# telnet 192.168.0.77
Trying 192.168.0.77...
Connected to 192.168.0.77.
Escape character is '^]'.

        Welcome to Freescale Semiconductor Embedded Linux Environment

freescale login:root
Password:
```

where root password is *root*. Once we logged in we issue next command

```
mx27# ps -ef
......
......
 1684 root        448 S   /sbin/syslogd
```

```
 1686 root        432 S   /sbin/klogd
 1692 root        624 S   /usr/sbin/inetd
 1693 root        980 S   /bin/sh /etc/rc.d/rc.local
 1696 root        356 S   /testfb
 1705 root        544 R   cp /rootfs.tgz /dev/shm
 1706 root        544 S   telnetd
 1707 root       1180 S   -sh
 1708 root        580 R   ps -ef
```

to find processes *rc.local* and *cp /rootfs..tgz /dev/shm*.

The example above gives us process numbers *1693* and *1705*. These two processes must be killed with a command *kill* as in exapmple bellow

```
mx27# kill -9 1693
mx27# kill -9 1705
```

Now we are free to explore and do whatever we want with the VoIP phone OS. Ok, lets restore our image from server with IP address *192.168.0.72* on a phone with */dev/sda1* storage chip. On the server firmware stored in directory */fm/iris-3000/sda20937.img*. I will use *ssh* to run a command on server and pump data to Iris-3000 phone where image will get copied into partition */dev/sda1*.

```
mx27# ssh root@192.168.0.72 'dd if=/fw/iris-3000/sda20937.img' \
bs=128k | dd of=/dev/sda1 bs=128k

Host '192.168.0.72' is not in the trusted hosts file.
(fingerprint md5 37:b4:05:50:fc:45:98:06:00:35:b4:75:86:a2:d6:4d)
Do you want to continue connecting? (y/n)
y
Password:
mx27#
```

In this example password is for user *root* on computer where firmware image is stored.

Once the command finishes you will be provided with a prompt *mx27#*. At this moment you can *power cycle* the phone and the phone will boot into newly installed OS.

5.4 Red Boot method

Red Boot is a software stored in the phone which activated for very brief time at boot.

You can read documentaion about *Red Boot* which avalable by following next link Red Boot.

During boot process *Red Boot* activates and waits for user input. If this input does not happend during assigned time frame *Red Boot* activates normal boot process by passing control to kernel with parameters.

To access *Red Boot* you need special setup for your computer. The best option is a computer running *Linux*, small hub or switch and Iris-3000 VoIP phone. You need hub or switch to maintain connection with the computer otherwise network software *"forgets"* assigned IP address if you turn off the phone.

Once *Red Boot* activates it assigns predefined IP address to network interface. During this short period of time you can *telnet* into VoIP phone and interrupt normal boot process.

I've run some tests and in most cases *Red Boot* was accessible in 3 seconds after powering *on* the phone. But it is not guaranted and you may have to try a few times.

To make access to *Red Boot* easier you can start *ping* command in separate terminal and observe the moment when *Red Boot* assigns IP address.

I found that it worked quite well if you follow next sequence: in terminal type command and be ready to press *Enter*. Now press *Enter* and immideately switch VoIP phone *on*.

You should see next message in terminal windows where you *telnet* from

```
Escape character is '^]'.
== Executing boot script in 0.900 seconds - enter ^C to abort
```

Red Boot assigns IP address *10.73.61.201* to network interface of VoIP phone. To be able to establish *telnet* session your computer interface should be on same subnet. In *Linux* this can be achived with next command

```
/sbin/ifconfig eth0 10.713.61.100 netmask 255.255.255.0
```

In dedicated terminal window type next command

```
telnet 10.73.61.201
```

hit *Enter* and immideately power *on* your Iris-3000. If everything goes according the plan you should get a message in terminal where it states that you need press *Ctrl+C* in assigned time frame which vary from 0.25 to 1 second. If you successfull in pressing *Ctrl+C* then boot process will be interrupted.

Now you will need to figure out what hardware you have in your VoIP phone. At *Red Boot* prompt type next command

```
RedBoot> fis list
Name              FLASH addr  Mem addr    Length      Entry point
RedBoot           0x00000000  0x00000000  0x00020000  0x00000000
RedBoot config    0x0007F000  0x0007F000  0x00001000  0x00000000
FIS directory     0x00070000  0x00070000  0x0000F000  0x00000000
RootFS            0x00100000  0x00100000  0x001a0000  0x00200000
```

if you see a line with *rootfs* then you have *MTD* version otherwise it must be *SDA* version. And to confirm this findings you can issue other comand

```
fconfig -l
```

check output for *root=...* which should confirm what type flash memory your phone equipped with.

```
RedBoot> fconfig -l
Run script at boot: true
Boot script:
.. fis load kernel
.. exec -w 1 -b 0x100000 -l 0x200000 -c ''noinitrd \
console=ttymxc0,115200 root=/dev/sda1 rootfstype=ext3 \
init=linuxrc ip=none fec_mac=00:04:9f:00:82:4c video=mxcfb:TV-NTSC''

Boot script timeout (1000ms resolution): 1
Use BOOTP for network configuration: false
Gateway IP address: 10.0.0.81
Local IP address: 10.73.61.201
Local IP address mask: 255.0.0.0
Default server IP address: 10.73.61.56
Board specifics: 0
Console baud rate: 115200
Set eth0 network hardware address [MAC]: false
Set FEC network hardware address [MAC]: false
GDB connection port: 9000
Force console for special debug messages: false
Network debug at boot time: false
Default network device: mxc_fec
RedBoot>
```

Now according Joshoa we need to ussue next two commands which should grant us access into VoIP phone system

```
RedBoot> fis load kernel
.....
RedBoot> exec -w 1 -b 0x100000 -l 0x200000 -c ''noinitrd \
```

```
console=ttymxc0,115200 root=/dev/sda1 rootfstype=ext3 \
init=linuxrc ip=none fec_mac=00:04:9f:00:82:4c video=mxcfb:TV-NTSC''
....
```

The second command can be found in previous block of output. This method did not work in my case as partition table of the phone was incorrect.

6 Software links

6.1 Test SD Card image – Iris3000

This is my initial SD Card image which includes some improvements on UI side Version 0.1: 20.9.37. I posted it in the blog for Joshoa's inspection to give his "Good to Go".

6.2 NFS method image – Joshoa

This software is marked by Joshua as UMEC Codetar which used in NFS method of software update and in my method #2 to unbrick Iris-3000 phone.

6.3 Latest SD Card image – Joshoa

Latest version of SD Card provided by Joshoa is available for download from next link Version 2: 20.9.37. Root password is *1234*, /web-on.sh and /web-off.sh included.

6.4 Previous SD Card image – Joshoa

Previous version of SD Card provided by Joshoa is available for download from next link Version 1: 20.9.37

6.5 Test SD Card image – Joshoa

Test version of SD Card provided by Joshoa is available for download from next link Version test: 2.9.37.

7 Passwords you could try on your system

7.1 Default passwords used on ACN Iris-3000 phone

```
Factory Reset Password: 7517517
GUI Admin Password: 5157919 (Version 0.2.55-3.1-US)
Default Root Password: acnum3c (Version 0.2.55-3.1-US)
Default Root Password: A1exV37
Default Root Password: 1234  (Joshoa)
Default Root Password: root  (iris3000)
```