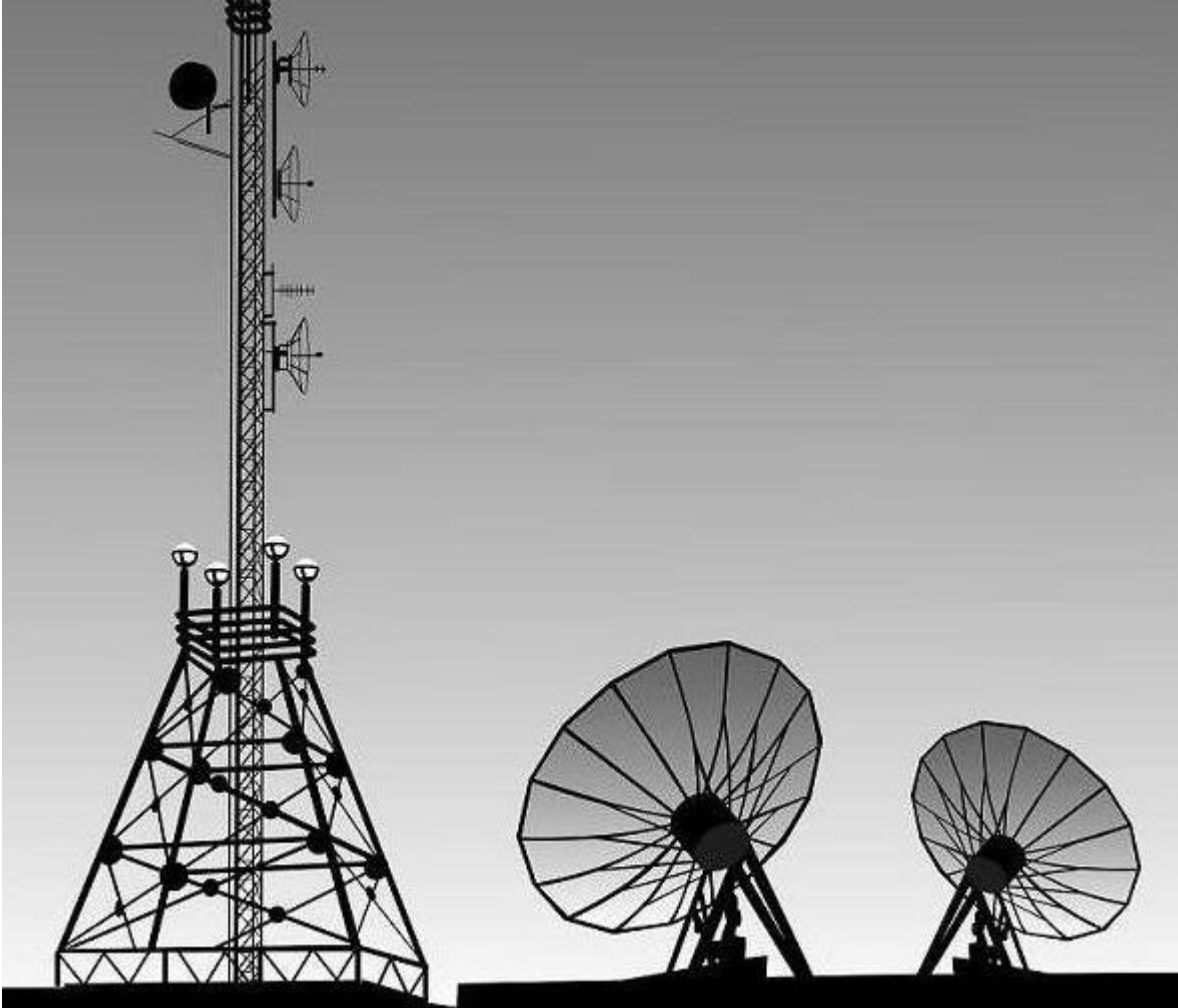


# Posting Content via Mobile Phone



Qtopia Greenphone Grant  
Posting Content to Drupal CMS from a Greenphone via the XML-RPC protocol

by: *Anthony Oliver*

# Table of Contents

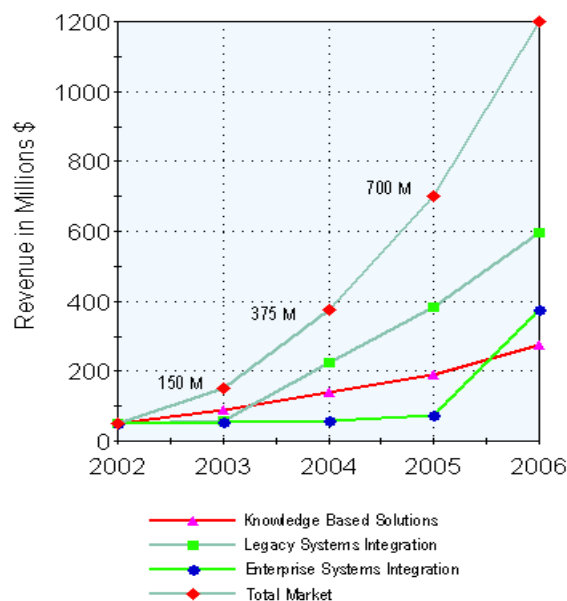
<b><u>Section</u></b>	<b><u>Page</u></b>
Why is posting content from a mobile device necessary? .....	3
Why would I even post to the Internet from my mobile device? .....	3
What type of content would I post with my mobile device? .....	4
What is the scope of this project? .....	4
Feasibility of this project .....	4
What is the Greenphone? .....	5
What is Drupal? .....	6
Project Plan Details .....	7
Actual Project Details .....	8
Summary .....	12
<b><u>Appendix</u></b>	
Greenphone Specifications .....	13
Schedule .....	14
Resources .....	15
Getting Internet to work on the greenphone .....	16

## Why is posting content from a mobile device necessary?

As of 2006, 80% of the world's population has mobile phone coverage. And is expected to reach 90% by 2010. With this type of growth rate and ubiquitousness of the Internet, something like this is necessary. Many mobile devices currently being released offer much more than just a telephone service as well. Take for instance the release of the Apple iPhone. It has many features like the ability to listen to music, play videos, browse the Internet, view photos, and more. The adoption of mobile devices has exploded in mainstream culture over the past 10-15 years, especially with the advent of the Internet. So taking into consideration the rate of growth and adoption, it's a necessity because this is something that not only functions as a device for entertainment but as a tool for business in today's fast paced society.

## Why would I even post to the Internet from my mobile device?

If you look at the image below you can see the rate of the growth of data transmission. As you can see it's growing exponentially. From the chart you can predict that by 2020 almost all data will be transmitted digitally. Currently a lot of data is transmitted over wires, but the rate at which wireless communication is growing exponentially as well. So you can see that many people are posting all types of content (data) over the Internet. So you might ask, "well why would I post anything over a mobile device?" For one, digital data is much easier to process and manipulate. If you look at how businesses operate today, you will find it hard to find any that don't use a computer to handle most of it's business related transactions. With mobile devices this makes it easier to stay in contact, update project status, etc. all from within the palm of your hand. So business is ready to go with you wherever you travel. Many people like to post many other things too and stay in contact. With the explosion of social networks, such as myspace and facebook, many people are posting content (photos, personal information, work information, etc.) to the Internet everyday. So even if you, yourself don't currently post this type of stuff onto the Internet, many people in fact do.



## **What type of content would I post with my mobile device?**

That depends on the person. Some people like to post photos, some people would probably post project updates, others maybe post reviews of a restaurant they've eaten at. One example where this might be useful is news related information. Say someone is at a restaurant and something like a movie star crashes their car into the building next to them. They could snap a few photos and upload that to the Internet. There is even a business model for this in where people could get paid to write and post information on certain events. Typically in the business of News, the first person to get the news scoop gets paid first. So while another person might be taking pictures and writing information about it, then having to drive back to the office to update that information, someone with this type of technology could post information instantly that could be accessed by anyone in the world.

## **What is the scope of this project?**

This project is intended to provide some of the functionality mentioned in the above sections. I submitted a proposal for the Qtopia Greenphone Grant. The grant is given is phone is provided free of charge as well as an Software Development Kit (SDK) license in order that the proposal suggested will be accomplished. My proposal was to use the greenphone to post content to the Drupal Content Management System over the XML-RPC (Extensible Markup Language – Remote Procedure Call) protocol.

## **Feasibility of this project**

One may contend what is the actual feasibility of this project. As was mentioned earlier, mobile device use is growing exponentially and expected to keep growing. Google has stated that the cost of bandwidth is approaching zero, as well as the cost of manufacturing these type of devices. As the cost of the two keep going down, then you can expect the rate that you see them to expand. As for the feasibility of this project, I believe it is very useful. Take for instance the rumors are true that Google is working on a mobile phone currently (they are currently bidding on the 700Mhz spectrum from the FCC). Something like this in a corporate network in conjunction with a Virtual Private Network (VPN) may someday be the way we communicate and everything will be done via the grid (a.k.a. Network).

## What is the Greenphone?

Qtopia Greenphone is a Linux mobile development device open for unlimited software innovation. Offered as part of the Greenphone SDK, it makes Linux-based applications easier to build and faster to bring to market. This powerful GSM/GPRS device provides the perfect platform for creation, testing and demonstration of new mobile technology services and ships loaded with Qtopia Phone Edition. Below is a picture of the greenphone.



## What is Drupal?

Drupal is a free and open source modular Content Management System written in PHP. The main purpose of Drupal is to make posting content to the Internet a very simple process. Drupal offers the following features:

- Post, revise, and categorize content
- Conduct searches
- Post comments
- Take part in forums
- Vote in polls
- Work on collaborative writing projects
- Post and view personal profiles
- Communicate among themselves or with the managers of a site
- Change the look of a site with off-the-shelf or custom-made themes
- Build multi-level menus
- Provide users with an interface in their local language
- Provide RSS feeds
- Gather content from the RSS feeds of other sites
- Register and manage user accounts
- Assign fine-grained user roles, granting users permission to use selected features of a site
- Use access rules to deny site access to specified usernames, e-mail addresses, and IP addresses
- Provide statistics and reports for administrators
- Manage caching and throttling to improve how a site performs in heavy traffic
- Construct and specify various input filters and content types
- Generate user-friendly, easy-to-remember URLs (for example, "www.mysite.com/products" rather than "www.mysite.com/?q=node/432")
- E-commerce systems
- Workflow features
- Photo galleries
- Organic groups
- Google sitemaps
- Mailing list management

With all these features it makes it very convenient to post any type of data you want. Drupal provides the backend support for putting data onto the Internet. Drupal already has XML-RPC support to post content built into it's core. It was chosen for this project because it is very actively developed, and is open source. Finding support is very easy and will allow XML-RPC posting out of the box.

## Project Plan Details

The scope of this project is to focus on the mobile device. Content (photos, text, etc). will be entered in on the greenphone and be posted via XML-RPC calls over the Internet to Drupal. Since Drupal supports XML-RPC by default it will handle everything on the backend side. Drupal is also setup to handle large amounts of data request, so the main focus will be on the client side of this model. You can look at the mock-up picture below to get an idea of how content will look before it is posted. Development will be done using the QT development kit. Due to the fact the phone runs Qtopia phone edition, it requires the software to be written in QT. QT is a software library that is a wrapper for the C++ programming language. Qtopia runs on top of Linux. So the phone itself runs embedded linux, although access to hardware system calls is restricted due to the fact that Qtopia itself is not open source, but that is irrelevant to the scope of this project due to the fact XML-RPC is a lot higher in the software stack and doesn't require any calls to hardware to post content. Although photos may not be within this projects scope due to the fact drupal doesn't support posting of images within the core system, it only supports text data post via XML-RPC calls.

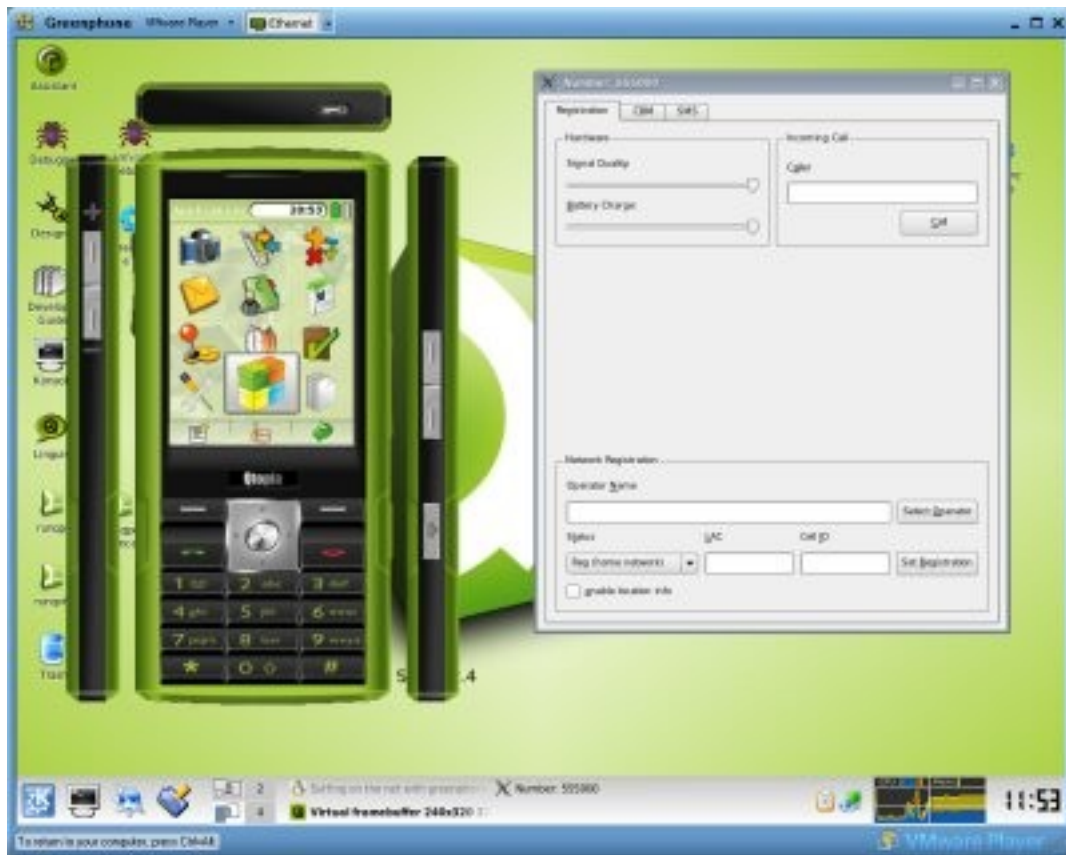
All the development is done via a virtual machine provided as part of the SDK. You build the software in development kit, as well as do testing of the software. Once the software is ready you can create a package file and copy that file to the phone for installation. The phone supports USB or Mini-SD (flash) memory to transfer files to it. As listed in the specifications of the greenphone it supports blue tooth as well as GSM based mobile services. It does not however support any wireless devices, and nothing like a flash based wireless card has been ported to it do to the fact that hardware system calls are not accessible via the software. So posting will be done over GSM using Wireless Application Protocol (WAP) or locally via bluetooth.

Much of this project will be spent learning the various development kits and the various libraries. I believe there probably won't be a lot of code to the actually software, but much of the work will be testing and getting the device to communicate with the server effectively. There is a schedule for this project as well, see the appendix.

## Actual Project Details

The “Project Plan Details” were written at the beginning of the project to give an outline, and a schedule and project scope, along with deliverables. As you can imagine, things didn't go according to plan. In the initial stages of the project, there were many technical difficulties. The first was, the phone can be charged via power adapter or via USB. Well it was plugged in USB, and then locked up one time during the boot process and couldn't seem to get it to turn back on. There was much discussion back and forth between the tech support. Although tech support wasn't very responsive, as I can understand, because this phone was given to me as a grant, whereas they have to support their customers first. They gave me many suggestions such as letting the battery completely drain, trying to re-flash the phone, holding # while the phone boots to skip the boot loader, etc. This dialog took a few weeks with no success. After that I was starting to get frustrated with the issue not being solved, I have a good background in electronics and was certain the battery was dead, after doing some test with a multi-meter across the leads of the battery I determined that the battery had no voltage potential and hence no charge. I finally convinced them this was the problem, and they decided to send me a new battery. Well after about 2 weeks the battery still hadn't arrived, I contacted them again and asked them when they sent it. Well something happened and they forgot to send it, so they sent it priority mail and I received it a few days later. I put the battery in the phone, and immediately the phone booted up. I was pretty happy because I thought, finally, I'm done with the problems and can move onto development.

It turns out the battery wasn't the only issue. The Software Development Kit (SDK) for the greenphone is meant to run inside a virtual machine. At first I thought this was a really good idea, because they can distribute one package (VM image) and all the development tools you will need will be right inside the virtual machine instead of having to worry about getting a development environment setup on your own machine. But this wasn't the case. I came across many issues with the phone due to it transferring the data over USB. For some reason the development kit transfers data over USB using encryption. I am not quite sure why this is the case, it could be due to Qtopia running on the phone which is proprietary desktop environment for the phone and not open source. So I had a lot of issues with the phone not syncing up with the SDK, which was kind of necessary for testing. One thing that was nice is the SDK did have an emulator (see figure below) that allowed you run the software in the virtual machine and test it without having to send it to the phone, but it was limited in functionality compared to the greenphone itself. I kept having issues with it, and so I posted some forum post on the greenphone forums. The support documents were really poor for the phone, as well the the forums were not really active at all. I found that there was an Internet relay chat (IRC) channel available that discussed qtopia development, so I went there and was able to talk to some people. Most of the developers there were running their development environment in linux or on a macintosh, no one there was using windows, which I was as my primary operating system. Someone suggested that I upgraded the firmware on the greenphone so I did. The phone came with Qtopia 4.1.7. I ended up upgrading it to 4.2.4. What was strange is that it was done over USB. The firmware upgrade went fine, but I still was having issues with it transferring the package to install the software I was working on to the greenphone after the firmware upgrade. The phone also kept locking up. So I went back to the IRC channel to see if I could find any help. They said I could possibly upgrade to Qtopia 4.3.0. when it was released, but I told them that I needed to get it done before then. I finally discovered the reason the phone was locking up was due to me not setting the time after I had replaced the battery. I left the date as 1900 when the new battery was inserted, I figured it wouldn't make a difference, but somehow it did and there was a bug, maybe in one of the services running on the phone.

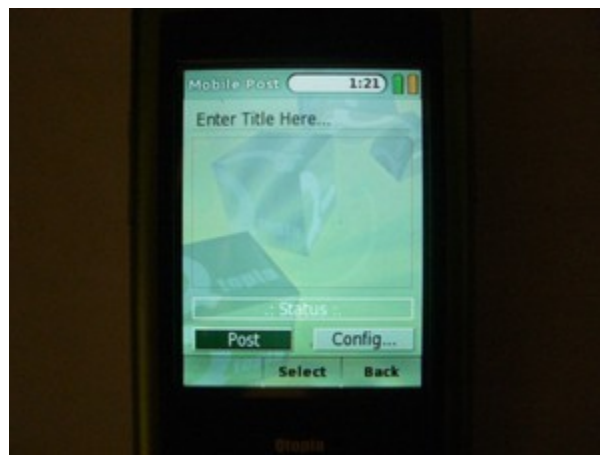


*Screen shot of the virtual phone emulator and SDK*

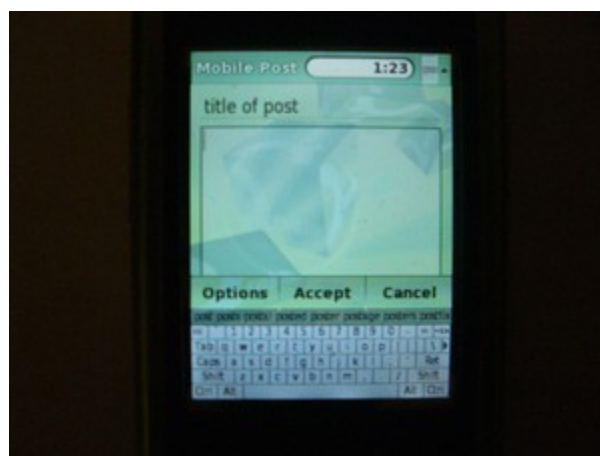
After getting the phone to quit locking up, I had to figure out how to get the data to transfer to the phone. I was still having issues with it, I thought it may be due to the encryption being used, because the phone has to share a public key of a digital signature algorithm (DSA) with the SDK in order to send things over it. It was displaying that keys were being exchanged correctly, so I couldn't figure it out. Well after some trial and error I figured out that it was the virtual machine software itself. Initially I was using VMware workstation to run the virtual machine, I then switched to VMware's player, and all the problems were fixed. I was able to copy my software over and everything ran smoothly. So I was happy that things finally started working out.

So after that I the interface designed and I figured out how everything seemed to be working, I needed to do some testing. Well once again I ran into a snag. This time it was the phone not having an Internet connection. I followed some instructions on the greenphone wiki that explained how to get the Internet working on it, but they didn't work. So I went back to the IRC channel to see if I could figure out the issue. I worked with a few people on it, but couldn't come to a solution. Well finally with the help of someone we figured it out. I needed to be running the network address translation (NAT) service for my virtual machine, as well as configuring the firewall on my windows machine to allow data, configuring the firewall inside the VM to allow data, then configuring the greenphone. I put instructions of how to accomplish this in the appendix of this document.

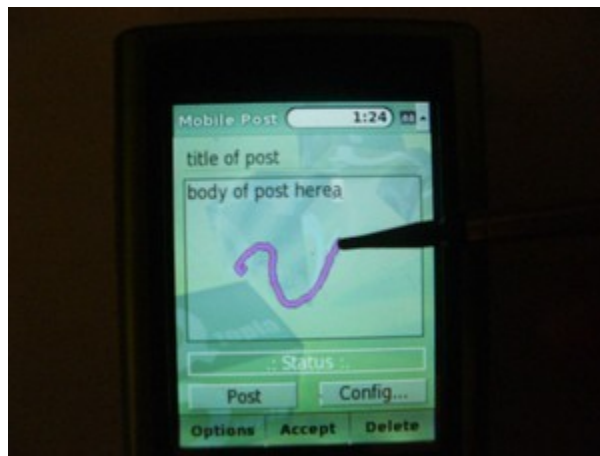
So after much time spent trying to work out the various bugs and problems I was having, I finally got things going smoothly. Below are some pictures of the actually phone as well as a screen shot of the website after data has been posted. The interface was meant to be really simple, you just enter in a title for your post and a body. The phone allowed data to be entered in with a keypad, with a virtual keypad, and writing with the stylus which can be seen in the pictures below. There is also a configuration section which is meant to include the website address of the Drupal CMS you want to post to. Notice the RPC2 at the end of the URL, this is a requirement of Drupal so it knows how to handle the XML-RPC data. The user name and password are also required. There is also a small status section, this changes as data is posted, if there are any errors, or if data is saved. Initially it was designed as a progress bar but in the end was changed to a status box to allow for more information to be displayed to the user.



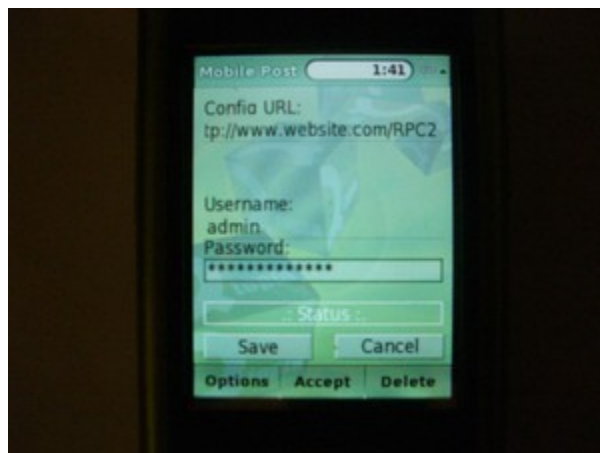
*A new Post, as the screen looks when the program is initially launched*



*Title has been set and body is being entered you can see the virtual keypad at the bottom of the screen for text input*



*This is after the text has been entered, you can see the stylus is capable of entering data as well, character recognition is supported.*



*This is the configuration section of the program, allowing you to set the URL for posting, passwords are also masked to retain security if the phone was lost.*

## Summary

In retrospect much of this project was spent trying to figure out problems with the phone. It would have been nice to spend more time on development. Maybe incorporating such things as saving post to allow for off-line data entry, such that data could be entered, saved, and then posted when an Internet connection was available. Other things too like the ability to post pictures and such would also be a nice feature to have. I do believe this was a good project in either case, there were many obstacles to overcome, but I think after seeing the growth of mobile devices I think something like this is really useful. It is more powerful than SMS, and allows the software to be extended and developed due to it being open source.

During the course of this project, many things in the mobile device have changed. When this project began there was rumors that Google was working on a mobile phone, since then the rumors have been clarified, and they have actually released a mobile operating system based on Linux called, "Android". This operating system is open source and has many nice features that allow people to develop software for it. Currently they have only released the SDK for it, but they are in talks with many mobile device providers in putting operating system on the phone. So something like this project could be ported as a piece of software for Google Android. Also the openmoko phone is about to be released. It is a completely open phone (hardware & software). Whereas only pieces of the greenphone are open. The openmoko phone's operating system is still under development, but you can put Qtopia 4.3.0 on it currently, so my project could actually run on it. At the time of writing this summary the Qtopia greenphone is also sold out, which is worth noting that more people are becoming interested in this sort of thing.

I think this will become a bigger trend as the next few years progress, you will see more active development in open source software as well Linux running on a myriad of mobile devices due to it's low cost and potential for scalability. I think given more time and a bigger budget (hours, etc.) this project could be developed extensively. Such things as posting to other mobile content management systems, and things like posting to facebook using their applications platform are more than feasible. I also believe devices and software like this give the freedom of speech back to the people and take it away from some of the few giants that control media now. They allow anyone to discussion whatever they like, whenever they like, cheaply and effectively, as well as build social communities to help solve problems. I think as time progresses, mobile devices will become a staple of society for media and information, and I will look back on this project and be glad I worked on it at the time when ideas like this were relatively new.

# Appendix

## **Greenphone Specifications**

### *Hardware*

- Touch-screen and keypad UI
- QVGA® LCD color screen
- Marvell® PXA270 312 MHz application processor
- 64MB RAM & 128MB Flash
- Mini-SD™ card slot
- Broadcom® BCM2121 GSM/GPRS baseband processor
- Tri-band (GSM 900/1800/1900) support
- Bluetooth® equipped
- Mini-USB port
- 1.3 megapixel camera

### *Software*

- Qtopia Phone Edition 4.1.7
- Linux kernel 2.4.19

# Schedule

---

Week 1 – 4:

Get in contact with professor, setup design document and schedule

Week 5 – 6:

QT Refresher

Week 6 – 7:

Learn SDK

Week 8 – 10:

Build Basic Program/Demo

Week 11:

Testing

Week 12:

Error Checking Fix Bugs

Week 13:

Final Documents and Program

Week 14:

Release Software Package under GPL.

# Resources

---

Greenphone Website:

<http://trolltech.com/products/qtopia/greenphone>

Greenphone Forum:

[http://www.qtopia.net/modules/newbb\\_plus/viewforum.php?forum=13](http://www.qtopia.net/modules/newbb_plus/viewforum.php?forum=13)

Greenphone Wiki:

[http://greenphone-wiki.org/index.php/Main\\_Page](http://greenphone-wiki.org/index.php/Main_Page)

QT XML-RPC library:

<http://qtnode.net/wiki?title=QxtXmlRpc>

Openmoko Phone Website:

<http://www.openmoko.com/>

Google Android Phone Platform:

<http://code.google.com/android/>

## Getting Internet to work on the greenphone:

---

Steps:

1. Boot up SDK
2. Boot up Greenphone
3. Plug Greenphone in USB.
4. In the SDK open up a terminal and type (this will make you the root user):

```
sudo -s
```

5. In the terminal now type:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

6. In the terminal now type:

```
iptables -t nat -A POSTROUTING -s 10.10.10.0/24 -o eth0 -j MASQUERADE
```

7. Now in the terminal type:

```
telnet 10.10.10.20
```

Your shell should now read something like: [root@greenphone /]

8. You are now connected to the green phone, if it didn't connect you need to make sure you USB is setup correctly. Now you want to type the following:

```
route add default gw 10.10.10.21
```

9. Now we need to get the DNS server info from the windows machine. So open up a terminal in windows (run->cmd). Then type:

```
ipconfig -all
```

This will give all the information (in the DNS server section). On my machine it is connected to a router, so it's IP address is: 192.168.0.1

10. Now back on the greenphone terminal we want to type:

```
vi /etc/resolv.conf
```

11. That will open the vi editor then we want to type:

```
nameserver IP_ADDRESS_OF_WINDOWS_MACHINE_HERE
```

We type in the IP address of the windows machine. In my case it was 192.168.0.1

12. Now press:

```
esc
```

13. Now hold shift and press ; to get a colon. Then type:

```
x
```

and hit enter. It should save it.

Now you can type:

```
ping google.com
```

You should be able to get a some type of feedback, else something went wrong and you need to trouble shoot.

Things to note:

- Make sure that NAT service is running on your windows machine. I had skype running and for some reason was preventing it from starting.
- Your VMplayer needs to be running in NAT mode, not bridged or Host-only.