OpenMoko

Neo1973 user manual

Federal Communication Commission Interference Statement

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to providereasonable protection against harmful interference in a residential installation. Thisequipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one of the following measures:

- Reorient or relocate the receiving antenna
- Increase the separation between the equipment and receiver
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected
- Consult the dealer or an experienced radio/TV technician for help

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

FCC Caution: Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment.

IMPORTANT NOTE:

FCC Radiation Exposure Statement

This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. End users must follow the specific operating instructions for satisfying RF exposure compliance.

For body worn operation, this phone has been tested and meets the FCC RF exposure guidelines for use with an accessory that contains no metal and the positions the handset a minimum of 1.5 cm from body. Use of other enhancement may not ensure compliance with FCC RF exposure guidelines.

Chapter 1 Free Your Neo1973

OpenMoko is an Open Source project to create the world's first free mobile phone operating system. The OpenMoko project is a community that anyone can join, and help to design their ideal phone. The Neo1973 from FIC is the first of many phones that OpenMoko will run on.

The long term goal of OpenMoko is that phone software will no longer be tied to one phone. You can buy any compatible phone, and install any software over the whole range of phones. If you upgrade your phone, you don't lose the software. Bugs fixed on one phone are fixed on all.

Please join us in collaborating on the OpenMoko project through any of the project resources including the OpenMoko wiki. Please see the wiki editing help page for information on making contributions to this wiki. A core team of developers funded by FIC, Inc. leads the project.

1-1 Neo1973 package list

All the following information could be found at www.openmoko.org Please go to the web site for most updated information.

A Neo1973 standard package including:

- a. 100 to 220 v adaptor plug
- b. USB Charger
- c. USB charger cable
- d. USB A to mini B cable
- e. Stylus
- f. Lanyard

- □ g. Earphone
- □ h. Battery
- □ i. SD-Card
- □ j. Phone pouch

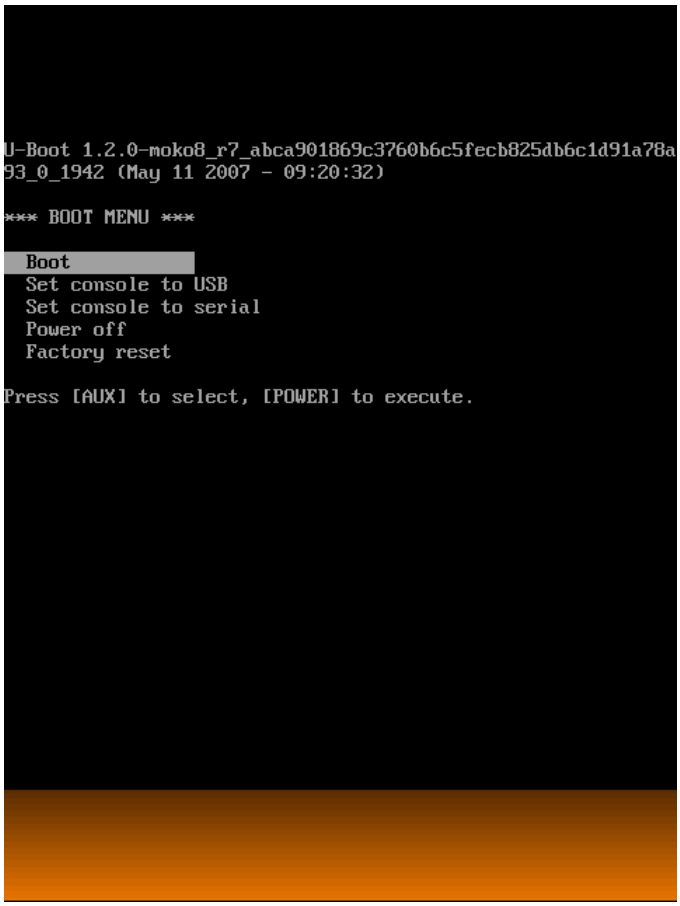
1-2 Neo1973 GTA01 Specification

Name	Neo1973
Size	120.7 x 62 x 18.5 mm
Weight	135 g
Screen	2.8" 480x640 at 285 ppi, maybe multi-touch later
Flash	64 MB (expandable with 2 GB microSD)
CPU	Samsung s3c2410 SoC @ 266 MHz (Source)
SDRAM	128MB
GSM	GSM, GPRS 2.5G (Not EDGE), Bluetooth 2.0 EDR, no WiFi
GPS	Global Locate GPS,
	OpenMoko Linux (Source code available)
	Basic PDA included. Software can be created by normal users.
	No recording. Maybe playback

Sound	Playback and recording, including playback of audio files via GSM and recording/playback of voice calls. 2x1 watt stereo speaker comment on performance, 2.5 mm jack
USB	Standard USB 1.1 (unpowered), with a Mini-B receptor (can be connected via adapter to both host and client devices)
Battery	replaceable 1.2 Ah battery charged via USB

1-3 Enter Neo 1973 U-Boot menu

Please insert the battery into the Neo1973 battery slot, then press Aux key and power key the same time. Then you could enter following u-boot menu.



After enter u-boot menu, you could use [Aux] key to select the item, and [Power] key for confirm. There should be 5 items by default:

1. **Boot**: Direct boot
2. **Set console to USB**: Could use Linux terminal software (like minicom to connect Neo1973)
3. **Set console to Serial**: Re-Direct terminal output to Neo1973 debug port serial interface.
4. **Power off**: Power of Neo1973
5. **Factory reset**: Reset parameter into factory default value.

If your Neo1973 already had kernel image or rootfs installed, once you select the boot, system will boot. If your Neo1973 doesn't contain any kernel image, you will need to use devirginator or DFU-Util to install system and root file system.

If you using the minicon or other terminal software like hyperterminal, please using following settings to connect terminal:

Speed: 115200, N ,8, 1

Hardware flow control: Off

1-4 Install OpenMoko Kernel image by using DFU

1-4-1 Purpose

dfu-util is a program that implements the Host (PC) side of the USB DFU (Universal Serial Bus Device Firmware Upgrade) protocol.

In the OpenMoko project, we use this program to communicate with our specially enhanced u-boot boot loader, which implements the DFU device side.

Using dfu-util and your Neo1973, you can Link* transfer and flash partitions to NAND Flash, such as

- The Linux kernel
- The root file system partition
- The splash screen partition
- The u-boot environment

Last, but not least: The u-boot bootloader itself.

transfer anything into Neo1973 RAM

this can be used for fast development cycles of low-level code such as kernels without flashing them

read out the current NAND partitions

this is an easy and efficient way of doing full backups of your phone

1-4-2 Source Code

dfu-util can be found at http://svn.openmoko.org/trunk/src/host/dfu-util/

Binary packages will be made available as part of the regular OpenMoko builds

1-4-3 help manu

dfu-util - (C) 2007 by OpenMoko Inc.

This program is Free Software and has ABSOLUTELY NO WARRANTY

```
Usage: dfu-util [options] ...
  -h --help                             Print this help message
  -V --version                          Print the version number
  -l --list                             List the currently attached DFU
capable USB devices
  -d --device vendor:product            Specify Vendor/Product ID of DFU
device
  -c --cfg config_nr                    Specify the Configuration of DFU
device
  -i --intf intf_nr                     Specify the DFU Interface number
  -a --alt alt_nr                       Specify the Altseting of the DFU
Interface
  -t --transfer-size                    Specify the number of bytes per USB
Transfer
  -U --upload file                      Read firmware from device into <file>
  -D --download file                    Write firmware from <file> into device
  -R --reset                            Issue USB Reset signalling once
we're finished
```

1-4-4 List function

Using the --list option, you can list the available DFU capable devices, their configuration, interface and altsettings. Below is an example for a current Neo1973 phone in u-boot Runtime Mode

```
# ./dfu-util --list
dfu-util - (C) 2007 by OpenMoko Inc.
This program is Free Software and has ABSOLUTELY NO WARRANTY
```

Found DFU Runtime: [0x1457:0x5119] devnum=0, cfg=0, intf=2, alt=0, name="USB Device Firmware Upgrade"

Below is an example for a current Neo1973 phone in u-boot DFU Mode

```
# ./dfu-util --list
dfu-util - (C) 2007 by OpenMoko Inc.
This program is Free Software and has ABSOLUTELY NO WARRANTY


Found DFU: [0x1457:0x5119] devnum=16, cfg=0, intf=0, alt=0,
    name="RAM 0x32000000"
    Found DFU: [0x1457:0x5119] devnum=16, cfg=0, intf=0, alt=1,
name="u-boot"
```

```
Found DFU: [0x1457:0x5119] devnum=16, cfg=0, intf=0, alt=2,
name="u-boot_env"
Found DFU: [0x1457:0x5119] devnum=16, cfg=0, intf=0, alt=3,
name="kernel"
Found DFU: [0x1457:0x5119] devnum=16, cfg=0, intf=0, alt=4,
name="splash"
Found DFU: [0x1457:0x5119] devnum=16, cfg=0, intf=0, alt=5,
name="rootfs"
```

This shows you six interfaces, all in configuration 0 and interface 0, with altsetting 0...4. The name is currently not yet defined, but will be added to one of the upcoming u-boot releases. The mapping on the Neo1973 is as follows:

```
0: RAM
1: 'u-boot' partition
2: 'u-boot_env' partition
3: 'kernel' partition
4: 'splash' partition
5: 'rootfs' partition
```

1-4-5 device function

You can specify the USB Vendor and Product ID of the device you want to program:

```
dfu-util --device 0x1457:0x5119
```

If you only have one standards-compliant DFU device attached to your PC, this is optional. However, as soon as you have multiple DFU devices, dfu-util will detect this and abort, asking you to specify which device it shall use.

1-4-6 transfer-size

Specifies the size of each individual USB transfer. If you don't use it, the maximum possible size for your combination of host operating system and USB device is chosen (for optimal performance).

1-4-7 download

download the given file into the device.

1-4-8 upload

upload from the DFU device into the given file[name].

NOTE: Upload support is currently broken

Phrasebook

There's no full-fledged manual yet. Instead, some examples:

```
Flashing the kernel
dfu-util -a 3 -R -D /path/to/uImage

Flashing the bootloader
dfu-util -a 1 -R -D /path/to/u-boot.bin
Copying a kernel into RAM
dfu-util -a 0 -R -D /path/to/uImage

Once this has finished, the kernel will be available at the default load address of 0x32000000 in Neo1973 RAM.
```

Chapter 2 Applications

2-1 Overview

This page describes the set of applications being developed by the core set of OpenMoko developers. This is what is intended to be eventually released on the stock firmware of OpenMoko phones (Neo1973 and successors). If the community develops an application of suitable usefulness and quality, it may also be included on stock firmware. Community applications may be found on the Wish List and on projects.openmoko.org

Although OpenMoko is designed for smartphones that use a stylus, it would be foolish to expect people to only operate their handsets with stylus. For this reason, some core phone-related applications will be developed with finger (as opposed to stylus navigation) in mind.

NOTE: Applications listed on this page are listing in order of priority and by phase. 0th phase is for the developers preview release in March, 1st phase is for late March. 2nd phase is September... when we are ready for mass market appeal.

2-1-1 General Principles

All modifications are saved instantenously, there should never be a save command.

Make sure users can't make interface operation errors, or that the effects are easily reversible, instead of just notifying them of the potential consequences of their actions.

When you open a document you should be returned to the place where you were working when you last closed or save it (this is our concept of sessions).

Label buttons with adjectives, which describe the state of the object affected

Designers should seek an efficient monotonous solution to gain benefits, including ease of learning, simplicity of implementation, minimization of documentation, and lowered maintenance costs.

Whenever you find yourself specifying an error message, please stop; then redesign the interface so that the condition that generated the error message doesn't arise.

If the user gets no utility from a process, there is no reason to tell them

that it is happening.

Provide unlimited levels of Undo and Redo whenever possible.

Consider adding a Trash or Deleted Items area that has everything the user has deleted. The user's data is the most important priority. Let's make it hard for them to lose anything and easy to recover it.

NOTE: Innovation usually happens at the bottom. What we need to do is provide consistent ways to develop and deploy appliations.

2-2 OpenMoko Dialer

2-2-1 Overview

This article defines the Dialer Application, an application designed for the OpenMoko platform. The Dialer Application handles all call and SMS related interactions. This is a Native Finger-Based application.

2-2-2 Use Cases

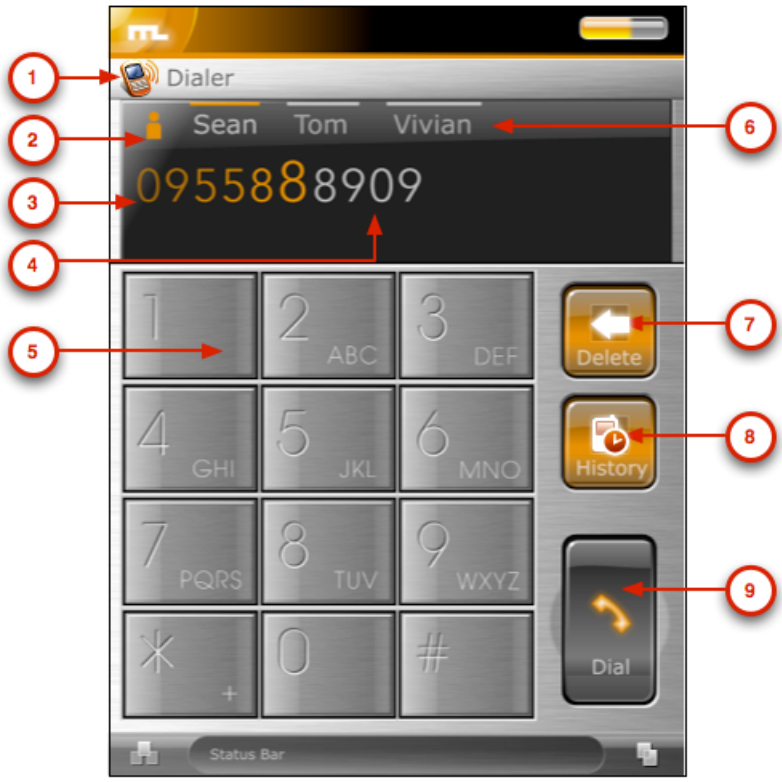
Here are some possible use cases:

I want to be able to dial a number

I want to be able to quickly redial a number

I want to be able to adjust the volume of both the speaker and microphone while in call

2-2-3 Dial Mode



1) Application Title -- Title of this application.

2) Icon / Photo -- Photo of contact (if available) will be displayed if number can be autocompleted.

3) Phone Number -- Phone number is displayed. Larger font is used for first line. After reaching end of first line the entire font size should decrease to allow for two full lines.

4) Autocompleted Tail -- Automatically complete the tail of phone numbers from existing contacts (when matched). This uses the Auto-complete widget.

5) Keypad -- Standard dialing keypad. The '*' key, when pressed quickly, cycles through this list: {*, +, p, w}.

6) Possible Autocompletions -- A list of contacts that can be autcompleted. This list is only as long as the visible screen area permits. (Usually 4-5 contacts). These are buttons that, when pressed, will autocomplete the

current number string.

7) Delete -- Deletes the last entered digit.

8) History -- Goes to the History mode.

9) Dial -- Dials the entered number string.

NOTE: If a PIN is required to access the SIM card, the "Phone number" (3) display will change to enter a PIN. "History" (8) will gray-out and "Dial" (9) will change to "OK."

2-2-3 Incoming Call Mode

1) Call Icon -- Icon representing call type.

2) Icon / Photo -- Photo of contact (if available) will be displayed.

3) Answer -- Answer the incoming call.

4) Title -- Title of this window.

5) Reject -- Rejects call. Returns to last application.

6) Caller ID Title -- If contact is stored in the handset, the name will be displayed (Steve in this case). Otherwise display Unknown. Below displays the phone number of the other party.

2-2-3 In Call Mode

1) Call Icon -- Icon representing call type.

2) Icon / Photo -- Photo of contact (if available) will be displayed.

3) Volume control -- Adjust the speaker volume level. While adjusting 5) will display percentage (%) level information

4) Control Button -- This mode only has volume control. Tapping will do nothing.

5) Speaker Phone -- Turns on the speaker phone mode

6) Application Title -- Title of this application.

7) DTMF -- Changes to Dial Mode for DTMF support.

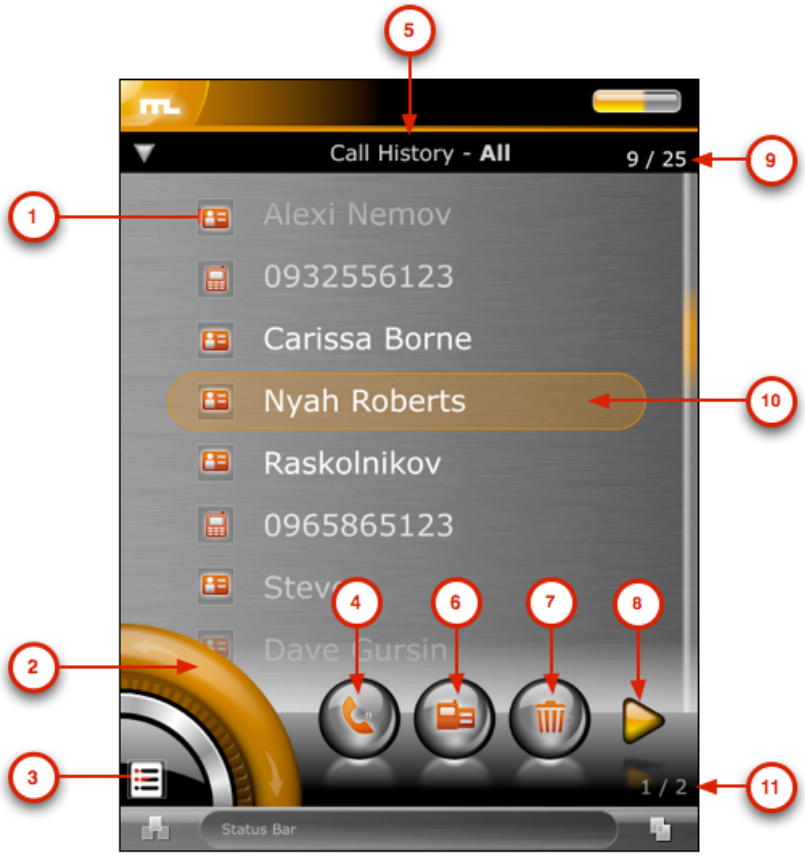
8) Hang-up -- Disconnects the current phone call. Returns to Dial Mode.

9) EMPTY -- Nothing for now. (Can be toggle of dictation)

10) Caller ID Title -- If contact is stored in the handset, the name will be displayed (Steve in this case). Otherwise display Unknown. Below displays the phone number of the other party.

11) Talk Time -- Displays call talk time.

2-2-4 Outgoing Call Mode



1) Call Icon -- Icon representing call type.

2) Icon / Photo -- Photo of contact (if available) will be displayed.

3) Speaker -- Turn on speaker phone mode

4) Title -- Title of this window.

5) Cancel -- Cancels call. Returns to last application.

6) Caller ID Title -- If contact is stored in the handset, the name will be displayed (Steve in this case). Otherwise display Unknown. Below displays the phone number of the other party.

2-2-5 Connection Error Mode

1) Call Icon -- Icon representing call type.

2) Icon / Photo -- Photo of contact (if available) will be displayed.

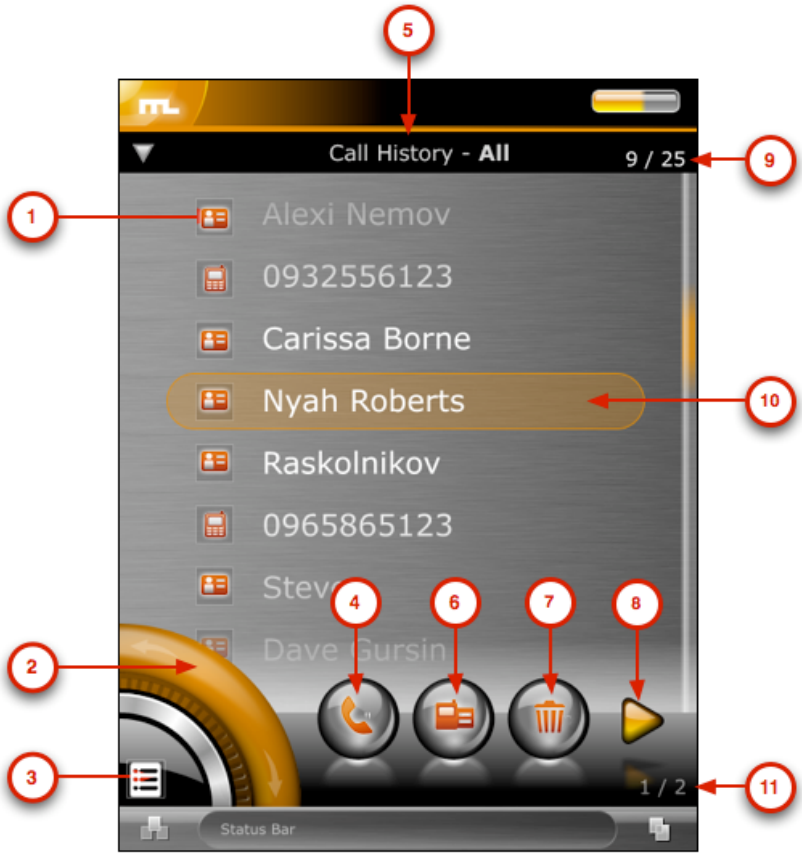
3) Redial -- Redials the last number.

4) Error Title -- Title of this error.

5) Cancel -- Cancels call. Returns to last application..

6) Caller ID Title -- If contact is stored in the handset, the name will be displayed (Steve in this case). Otherwise display Unknown. Below displays the phone number of the other party.

2-2-6 Call History Mode



1) Caller ID Title -- If contact is stored in the handset, the name will be displayed. Otherwise display number. An icon will represent the type (incoming, outgoing, missed) of call.

2) List Scroll -- Scrolls through the call history list. While scrolling, 3) will display the current / total item information (for example 2/8).

3) Control Button -- Tapping will change between the following modes: {Call History Mode, Close and returns to Dialer}.

4) Call -- Call the current selected number. Grayed-out if number is unknown.

5) Application Title -- Title of this application.

6) SMS -- Send a message to this number. This will open the Messages application with a new (SMS) message.

7) Delete -- Removes this number from the call history list.

8) More Items -- Tap to reveal the next list of items. The second list has the following items: { Save, All Calls, Missed Called, Dialled Calls, Received Calls, Back}

9) List counter -- Displays the current list entry / total list entries.

10) Current List Entry -- The current list entry. The position of the highlight is fixed. Only the text moves up/down with scrolling. User can tap a list item to have this move to center. Also, scrolling effect should ease in.

11) More Items Counter -- Displays the current set / total sets of More Items.

NOTE: The call list slides behind the scrolling widget. The scrolling widget and horizontal bar are 80% transparent.

2-3 OpenMoko Main menu

2-3-1 Overview

This article defines the Main Menu Application, an application designed for the OpenMoko platform. The Main Menu application provides both a finger-friendly and a stylus-driven interface to lauch all installed applications.

Here are some possible use cases:

I want to be able to launch an application

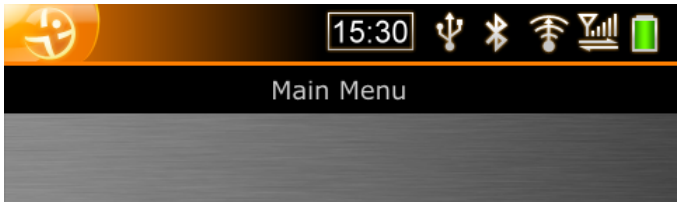
I want to rearrange applications in the menu

Activated by pressing the auxilary key. This is the base layout:

1) Section Name -- The name of the current section.

2) List scrolling -- Will scroll through the list of icons. While scrolling, 3) will show how many items exist in this list.

3) Control Button -- Tapping cycles through the following controls {Scroll List Control, Back}. Currently at Scroll List Control. The List is broken up into sections of applications.



4) History Applications -- Are the last 3 applications launched.

5) Item / Total -- Displays the current selected item number along side the number of items.

6) Sections or Applications -- Tapping on a section will lead to subgroups of applications. Tapping on an application will launch that application.