Getting started with jPhone.

1. Environment

At a minimum, jPhone requires a Java installation together with libraries to interface with the serial port. The environmental set up described here is not the only possible configuration and is used here for illustration purposes.

1.1 Eclipse

The Eclipse IDE is available here: http://www.eclipse.org/

The configuration for Java developers 4.2 is sufficient to run jPhone and provides a number of other convenient plugins in one download.

Optionally, download and install the Eclipse CDT plugins to facilitate browsing the Osmocom reference code included in the git repository.

The Eclipse platform is equipped with Git functionality and can be used to check out the project into the Eclipse workspace.

1.2 Serial Communication

A library to communicate with the phone via the serial port is required as the basic JVM runtime does not provide such functionality. This can be downloaded here:
http://rxtx.qbang.org/wiki/index.php/Main_Page

There is already a version of this in the lib folder of the project and if the project is checked out via Eclipse, it should be preconfigured to use this library.

1.3 Osmocom

Osmocom can be downloaded from: http://bb.osmocom.org/trac/

Once downloaded and compiled, a binary for your phone is produced. jPhone must be told the location of this binary so it can upload it to the phone.

Optionally, a precompiled binary for the C123 phone is located in the lib folder of the project and jPhone is preconfigured to use this. Should you require the use of a different binary please update the code. You must also ensure that the handshake mechanism between the phone and ram loader is the same i.e. the same sequence of messages and the same messages being sent and received for the loader to successfully work. In the future, support for a broader range of phones will be provided.

2. Runtime

Once the environment is configured and the source code is downloaded, you run jPhone like any other Java application in Eclipse. Right click on the Main.java file in the jPhone class and select “Run as Java Application”.

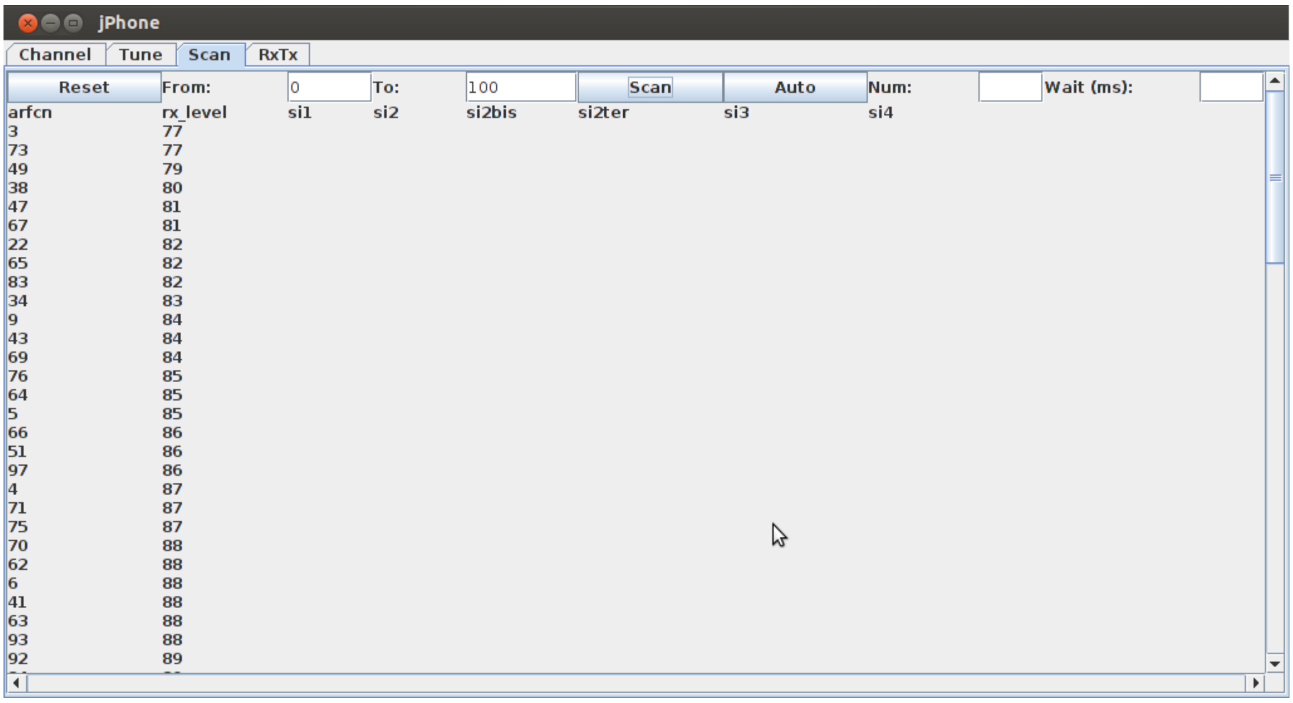
In some instances you may be required to change the access privileges on the serial port in order for the port to be visible. This is done using chmod in Linux or whatever alternative is provided by your operating system.

Once the GUI appears, there will be a progress bar displayed. At this point, briefly press the power button on the phone to begin the upload of the binary. Some hardware has a finite number of writes available so with this in mind, jPhone will test to see if the binary is already uploaded before commencing an upload. Once the progress bar disappears and the phone lights up the upload is complete and the GUI can be used.

Note that in some cases, the code will upload to the phone but it will fail to start. This happens especially if the phone has not been used in a while. Simply remove the battery from the phone to reset it and restart the GUI until the phone starts. This can take a number of times before it works.

2.1 Scan

The scan feature provides the user with the ability to scan all frequencies for towers in their locality. The ARFCN number from and to should be entered in their appropriate boxes as shown below and then the button Scan should be pressed.



From: 0 To: 100 Num: Wait (ms):

arfcn	rx_level	si1	si2	si2bis	si2ter	si3	si4
3	77						
73	77						
49	79						
38	80						
47	81						
67	81						
22	82						
65	82						
83	82						
34	83						
9	84						
43	84						
69	84						
76	85						
64	85						
5	85						
66	86						
51	86						
97	86						
4	87						
71	87						
75	87						
70	88						
62	88						
6	88						
41	88						
63	88						
93	88						
92	89						

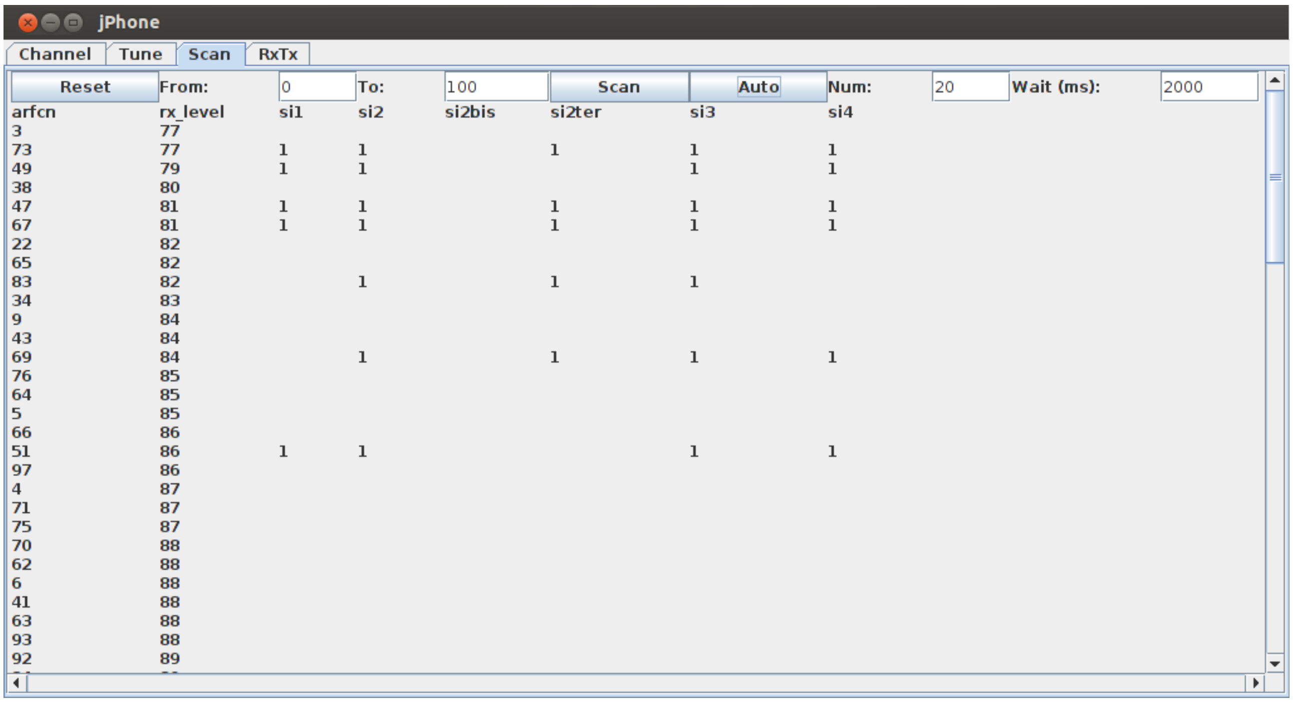
The code can scan all frequencies but for demo purposes ARFCN 0-100 have been used. It is simply a matter of changing the number in the box. This also provides the signal strength indicators for each tower. Towers are sorted based on the signal strength.

2.2 Auto-tune

Some towers that show up in the ARFCN scan do not in fact provide the required System Information in order to sync with them. The auto tune feature provides an automatic way to see which towers are providing the required information.

The parameters available are the number of towers to include in the auto tune and the number is from the top down i.e. the tower with highest signal strength is selected first. The second parameter is the amount of time that jPhone will listen to the tower sync info before moving on to the next tower. The smaller this value, the quicker the scan will complete, however, some System Information from the tower may be missed.

In the example below, tower on AFRCN 73 has the highest signal strength that provides the required information. Note that tower on ARFCN 3 does not provide the information and so it can be concluded that this tower at present is not valid.

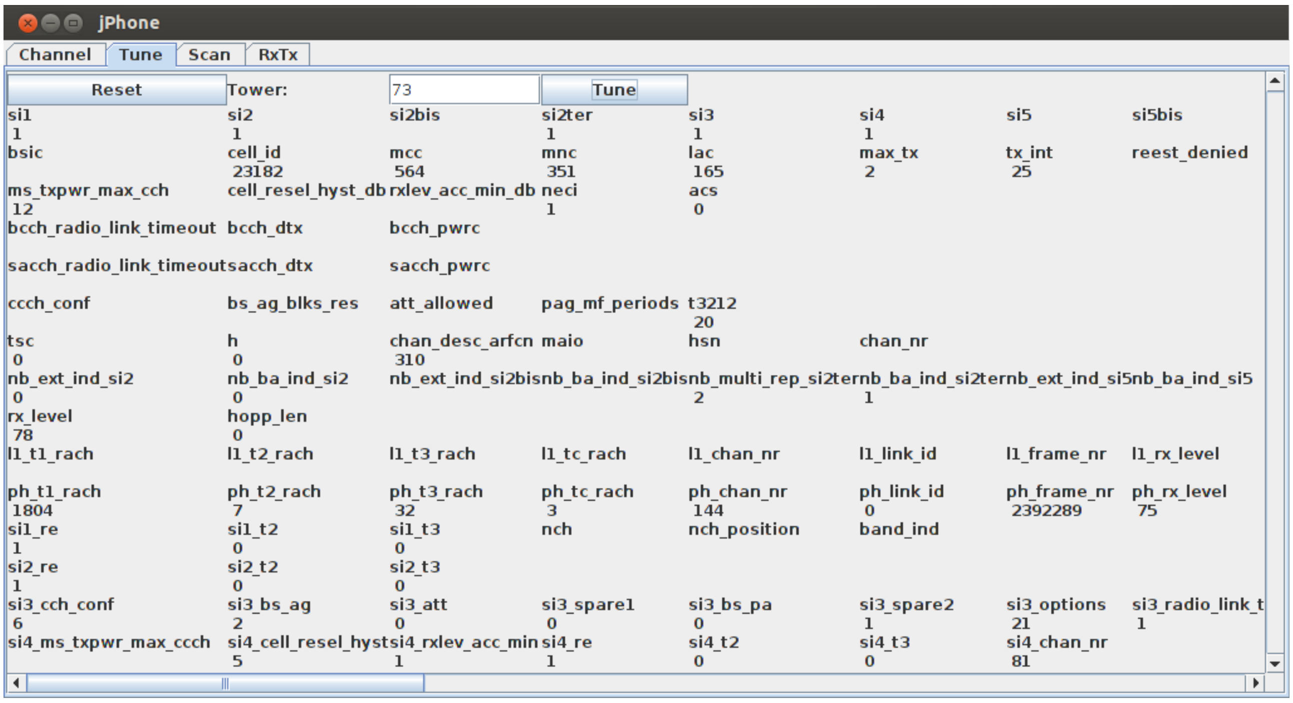


From: 0 To: 100 Num: 20 Wait (ms): 2000

arfcn	rx_level	si1	si2	si2bis	si2ter	si3	si4
3	77						
73	77	1	1		1	1	1
49	79	1	1			1	1
38	80						
47	81	1	1		1	1	1
67	81	1	1		1	1	1
22	82						
65	82						
83	82		1		1	1	
34	83						
9	84						
43	84						
69	84		1		1	1	1
76	85						
64	85						
5	85						
66	86						
51	86	1	1			1	1
97	86						
4	87						
71	87						
75	87						
70	88						
62	88						
6	88						
41	88						
63	88						
93	88						
92	89						

2.3 Tune

The Tune tab allows tuning to the frequency of the tower and deciphering the messages received. Once we have all the parameters decoded, as shown in the diagram, we can then use them to connect to the tower to send and receive data. In the diagram, the only variable that can really be changed is the tower based on the table populated by the previous tab.

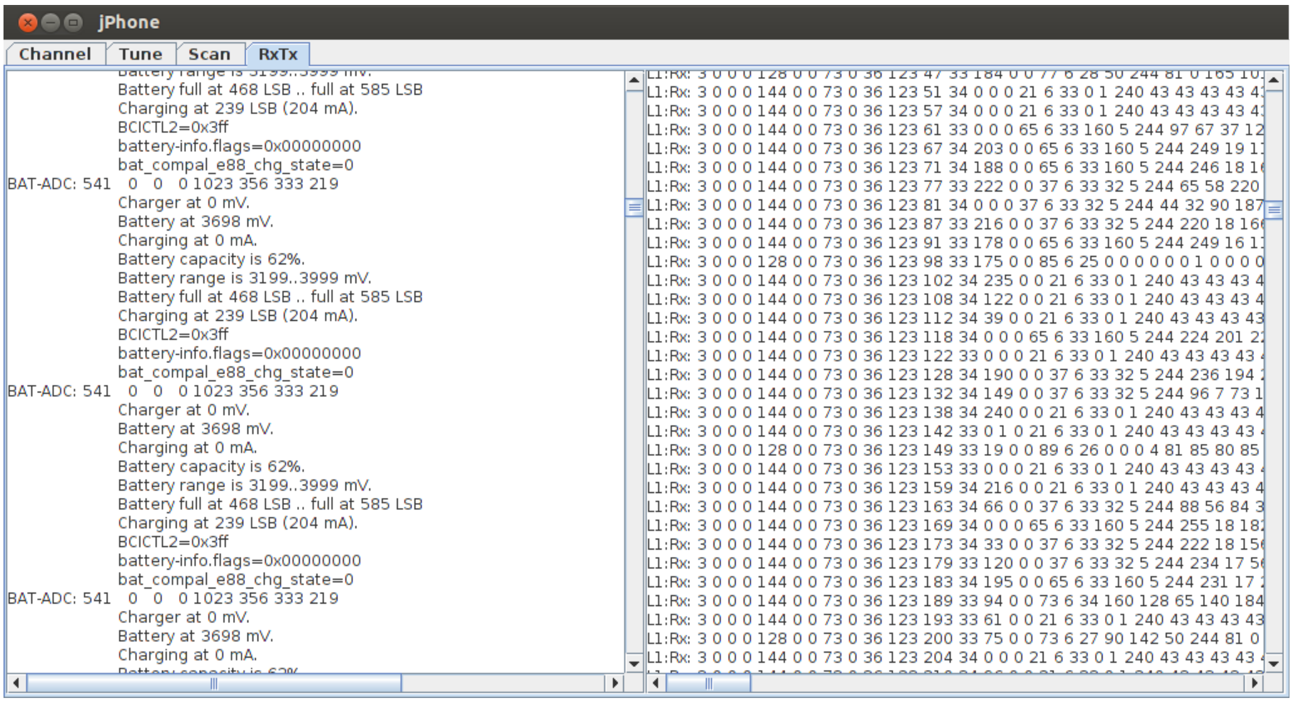


Tower: 73

si1	si2	si2bis	si2ter	si3	si4	si5	si5bis
1	1		1	1	1		
bsic	cell_id	mcc	mnc	lac	max_tx	tx_int	reest_denied
	23182	564	351	165	2	25	
ms_txpwr_max_cch	cell_resel_hyst_db	rxlev_acc_min_db	neci	acs			
12			1	0			
bcch_radio_link_timeout	bcch_dtx	bcch_pwrc					
sacch_radio_link_timeout	sacch_dtx	sacch_pwrc					
ccch_conf	bs_ag_blks_res	att_allowed	pag_mf_periods	t3212			
				20			
tsc	h	chan_desc_arfcn	maio	hsn	chan_nr		
0	0	310					
nb_ext_ind_si2	nb_ba_ind_si2	nb_ext_ind_si2bis	nb_ba_ind_si2bis	nb_multi_rep_si2ter	nb_ba_ind_si2ter	nb_ext_ind_si5	nb_ba_ind_si5
0	0			2	1		
rx_level	hopp_len						
78	0						
l1_t1_rach	l1_t2_rach	l1_t3_rach	l1_tc_rach	l1_chan_nr	l1_link_id	l1_frame_nr	l1_rx_level
ph_t1_rach	ph_t2_rach	ph_t3_rach	ph_tc_rach	ph_chan_nr	ph_link_id	ph_frame_nr	ph_rx_level
1804	7	32	3	144	0	2392289	75
si1_re	si1_t2	si1_t3	nch	nch_position	band_ind		
1	0	0					
si2_re	si2_t2	si2_t3					
1	0	0					
si3_cch_conf	si3_bs_ag	si3_att	si3_spare1	si3_bs_pa	si3_spare2	si3_options	si3_radio_link_t
6	2	0	0	0	1	21	1
si4_ms_txpwr_max_ccch	si4_cell_resel_hyst	si4_rxlev_acc_min	si4_re	si4_t2	si4_t3	si4_chan_nr	
	5	1	1	0	0	81	

2.4 RxTx

The final tab in the GUI is the RxTx tab that just monitors all the messages to and from the phone. It can also logs those messages to the appropriate files.



```
              Battery range is 3199..3999 mV.
              Battery full at 468 LSB .. full at 585 LSB
              Charging at 239 LSB (204 mA).
              BCICTL2=0x3ff
              battery-info.flags=0x00000000
              bat_compal_e88_chg_state=0
BAT-ADC: 541   0   0   0 1023 356 333 219
              Charger at 0 mV.
              Battery at 3698 mV.
              Charging at 0 mA.
              Battery capacity is 62%.
              Battery range is 3199..3999 mV.
              Battery full at 468 LSB .. full at 585 LSB
              Charging at 239 LSB (204 mA).
              BCICTL2=0x3ff
              battery-info.flags=0x00000000
              bat_compal_e88_chg_state=0
BAT-ADC: 541   0   0   0 1023 356 333 219
              Charger at 0 mV.
              Battery at 3698 mV.
              Charging at 0 mA.
              Battery capacity is 62%.
              Battery range is 3199..3999 mV.
              Battery full at 468 LSB .. full at 585 LSB
              Charging at 239 LSB (204 mA).
              BCICTL2=0x3ff
              battery-info.flags=0x00000000
              bat_compal_e88_chg_state=0
BAT-ADC: 541   0   0   0 1023 356 333 219
              Charger at 0 mV.
              Battery at 3698 mV.
              Charging at 0 mA.
              Battery capacity is 62%.
```

```
L1:Rx: 3 0 0 0 128 0 0 73 0 36 123 47 33 184 0 0 77 6 28 50 244 81 0 165 10
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 51 34 0 0 0 21 6 33 0 1 240 43 43 43 43 4
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 57 34 0 0 0 21 6 33 0 1 240 43 43 43 43 4
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 61 33 0 0 0 65 6 33 160 5 244 97 67 37 12
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 67 34 203 0 0 65 6 33 160 5 244 249 19 1
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 71 34 188 0 0 65 6 33 160 5 244 246 18 1
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 77 33 222 0 0 37 6 33 32 5 244 65 58 220
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 81 34 0 0 0 37 6 33 32 5 244 44 32 90 187
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 87 33 216 0 0 37 6 33 32 5 244 220 18 16
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 91 33 178 0 0 65 6 33 160 5 244 249 16 1
L1:Rx: 3 0 0 0 128 0 0 73 0 36 123 98 33 175 0 0 85 6 25 0 0 0 0 0 0 1 0 0 0 0
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 102 34 235 0 0 21 6 33 0 1 240 43 43 43 4
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 108 34 122 0 0 21 6 33 0 1 240 43 43 43 4
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 112 34 39 0 0 21 6 33 0 1 240 43 43 43 43
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 118 34 0 0 0 65 6 33 160 5 244 224 201 2
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 122 33 0 0 0 21 6 33 0 1 240 43 43 43 43
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 128 34 190 0 0 37 6 33 32 5 244 236 194
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 132 34 149 0 0 37 6 33 32 5 244 96 7 73 1
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 138 34 240 0 0 21 6 33 0 1 240 43 43 43 4
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 142 33 0 1 0 21 6 33 0 1 240 43 43 43 43
L1:Rx: 3 0 0 0 128 0 0 73 0 36 123 149 33 19 0 0 89 6 26 0 0 0 4 81 85 80 85
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 153 33 0 0 0 21 6 33 0 1 240 43 43 43 43
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 159 34 216 0 0 21 6 33 0 1 240 43 43 43 4
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 163 34 66 0 0 37 6 33 32 5 244 88 56 84 3
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 169 34 0 0 0 65 6 33 160 5 244 255 18 18
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 173 34 33 0 0 37 6 33 32 5 244 222 18 15
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 179 33 120 0 0 37 6 33 32 5 244 234 17 5
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 183 34 195 0 0 65 6 33 160 5 244 231 17
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 189 33 94 0 0 73 6 34 160 128 65 140 184
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 193 33 61 0 0 21 6 33 0 1 240 43 43 43 43
L1:Rx: 3 0 0 0 128 0 0 73 0 36 123 200 33 75 0 0 73 6 27 90 142 50 244 81 0
L1:Rx: 3 0 0 0 144 0 0 73 0 36 123 204 34 0 0 0 21 6 33 0 1 240 43 43 43 43
```

3. Future Work

– Support for more Osmocom compatible phone.
– Get a channel from the tower
– Send data to the tower
– Respond to messages from the tower e.g paging
– Receive data from the tower
– Mobility Management
– Call Control
– SMS capability