WiPhone User Manul

HackEDA

Created on : January, 2020

version : 0.1alpha

Documents:

FCC/IC Compliance 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 provide reasonable protection against harmful interference in a residential installation. This equipment 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 or more 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.

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

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.

RF Exposure Information: FCC RF Exposure requirements: The highest SAR value reported under this standard during product certification for use next to the head with the minimum separation distance of 10mm is 1.49W/Kg. This transmitter must not be collocated or operating in conjunction with any other antenna or transmitter. This product is compliance to FCC RF Exposure requirements and refers to FCC website https://apps.fcc.gov/oetcf/eas/reports/GenericSearch.cfm search for FCC ID: 2AHUL-9474663.

EU – Declaration of Conformity:

Shenzhen MZJ Technology Co., Limited declares that WiPhone Model # N0C311 and N0C311P complies with the essential requirements and other relevant provisions of Directive 1999/5/EC. A copy of the Declaration of conformity is available on request. Shenzhen MZJ Technology Co., Limited. Room 803 Chevalier House 45-51 Chatham Road South Tsim Sha Tsui Kowloon, Hong Kong

Caution:

This device contains licence-exempt transmitter(s)/receiver(s) that comply with Innovation, Science and Economic Development Canada's licence-exempt RSS(s). Operation is subject to the following two conditions:

1. This device may not cause interference.
2. This device must accept any interference, including interference that may cause undesired operation of the device.

L'émetteur/récepteur exempt de licence contenu dans le présent appareil est conforme aux CNR d'Innovation, Sciences et Développement économique Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes :

1. L' appareil ne doit pas produire de brouillage;
2. L' appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d' en compromettre le fonctionnement.

This EUT is compliance with SAR for general population/uncontrolled exposure limits in IC RSS-102 and had been tested in accordance with the measurement methods and procedures specified in IEEE 1528 and IEC 62209. This equipment should be installed and operated with minimum distance 10mm between the radiator and your body. This device and its antenna(s)must not be co-located or operating in conjunction with any other antenna or transmitter.

Cet appareil est conforme aux limites d'exposition DAS incontrôlée pour la population générale de la norme CNR-102 d'Industrie Canada et a été testé en conformité avec les méthodes de mesure et procédures spécifiées dans IEEE 1528 et IEC 62209. Cet appareil doit être installé et utilisé avec une distance minimale de 10mm entre i'émetteur et votre corps. Cet appareil et sa ou ses antennes ne doivent pas être co-localisés ou fonctionner en conjonction avec tout autre antenne ou transmetteur.

1 Overview

WiPhone is a unique, minimal phone. It's designed to enable hackers by making it easy to extend and modify the electronics and software. Something typical phones are not good for. WiPhone is also a VoIP mobile phone. It uses WIFI to make HD voice calls, for free. This means that there is no required service contract - and it's yours for life.

WiPhone is different beast from most smartphones these days. WiPhone uses the existing WiFi around you to make HD Voice calls. For free. Buy it once and it's yours.

Works on most broadband WiFi networks (including most home WiFi connections). No service contract required, and you can even upgrade the firmware or expand the hardware to do things it wasn't originally intended for.

2 Get Started

2.1 Charge the battery

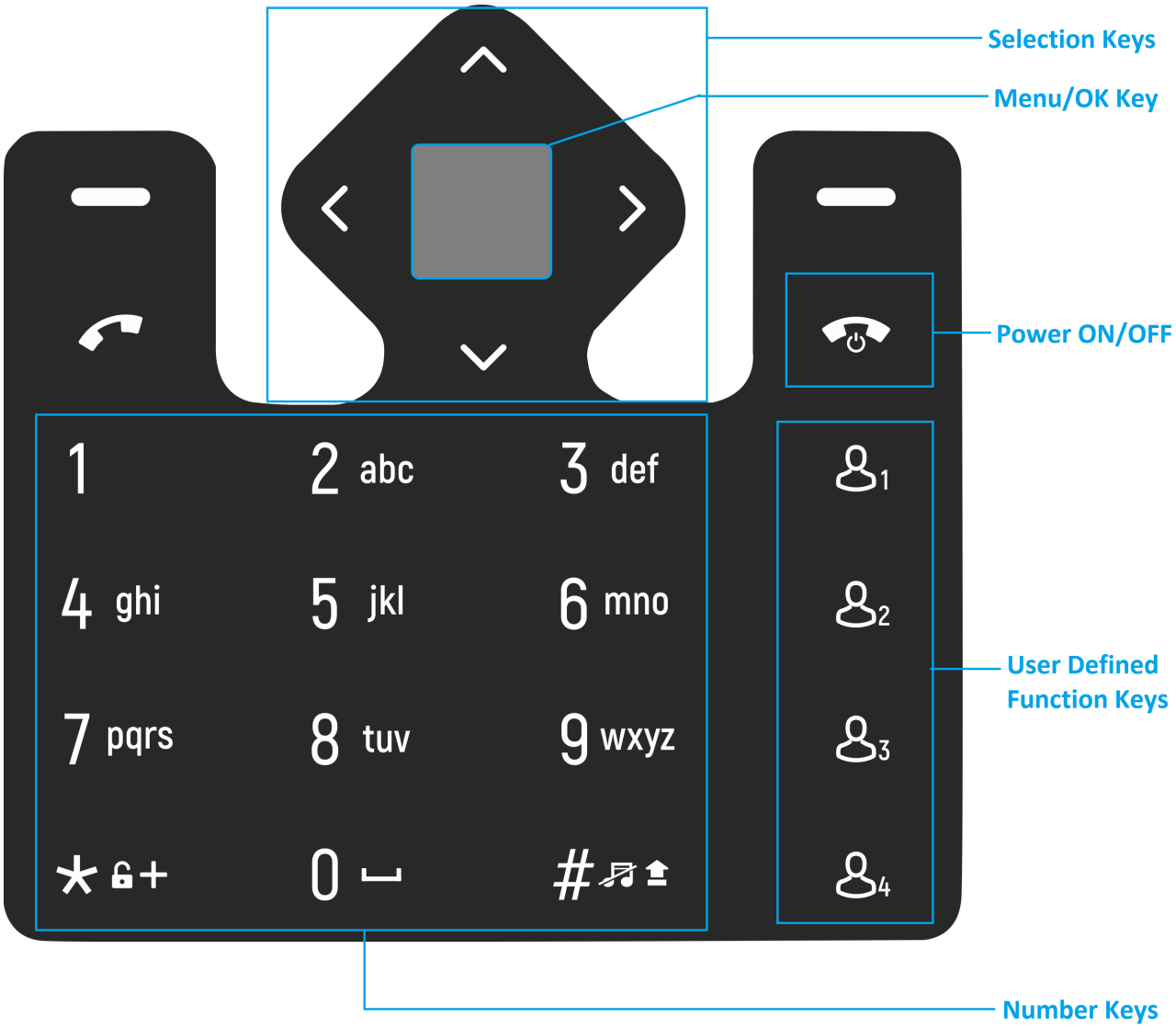
Your battery has been partially charged at the factory, but you may need to recharge it before you can use your phone.

1. Plug the charger into a wall outlet.
2. Connect the charger to the phone. When done, unplug the charger from the phone.
3. You can use USB charging when a wall outlet is not available.

Note: If the battery is completely discharged, it may take several minutes for WiPhone to attain stable power and function properly.

2.2 Keys and Writing Text

- Press a key repeatedly until the letter is shown.
- To type in a space press 0 .
- To type in a special character or punctuation mark, press *.
- To switch between character cases, press # repeatedly.



2.3 Turning WiPhone ON

Press the key to Turn ON the WiPhone. After normal bootup you will see Home screen.

Note: When phone bootup and not connected to WiFi you will see this text on Home Screen *Time: network no NTP* which means you need to setup WiFi connection to aquire time from NTP server.

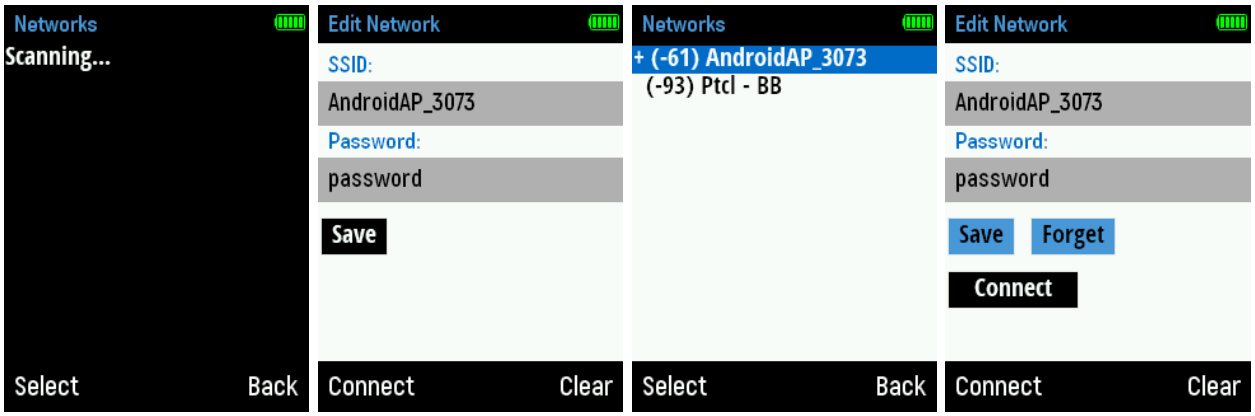
2.4 Turning WiPhone OFF

WiPhone support Software Shutdown and Force Shutdown.

- Press  button for ~4s and the screen of WiPhone will turn blank while in background WiPhone will Save and Stop any running tasks and then shutdown the WiPhone.
- In some cases when software shutdown is not working, you can hold  button fro ~10s to force WiPhone to shutdown.

2.5 Setting Up WiFi

1. Select **Menu** > **Settings** > **Scan WiFi networks**.
2. Select the desired network and Press **OK** button.
3. Enter Password.
4. Press Save Button.
5. Now the password for you desired network is saved and you will see + sign with that network.
6. Select the WiFi network from last step and press connect to WiFi network.



7. After Successful connection you will able to scree Wifi symbol at top status bar and WiPhone will update the date and time from NTP server.

Fig. 1: Main Screen.

2.6 Screen Settings

You can change your screen setting by navigating to : * **Menu** > **Settings** > **Screen config**.

The few options which you can change with screen settings are:

1. You can change the screen brightness. Select low brightness level to save Power.

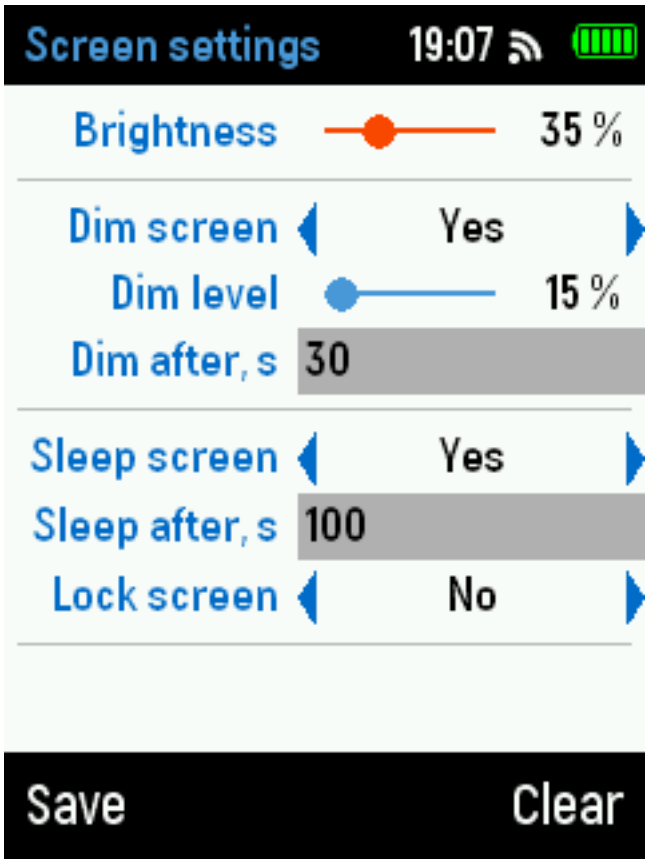


Fig. 2: Screen Settings.

2. Screen Dim option. If you select this option the screen will be automatically dimmed to specified brightness level.
   - Screen Brightness in Dim mode.
   - Time in sec after which screen will enter Dim mode.
3. Sleep screen: If you select this option, The screen will enter sleep mode and you will need to press any key to turn the display ON.
   - Time in sec after which screen will enter sleep mode.
4. Lock Screen: To avoid accidentally pressing the keys, use keypad lock option.

UNLOCK THE KEYS:

Press **OK**, and then * button to Unlock keys.

2.7 Audio Settings

To change the volume of Ear Speaker, Loud Speaker and Headphones

- Select **Menu** > **Settings** > **Audio settings**.

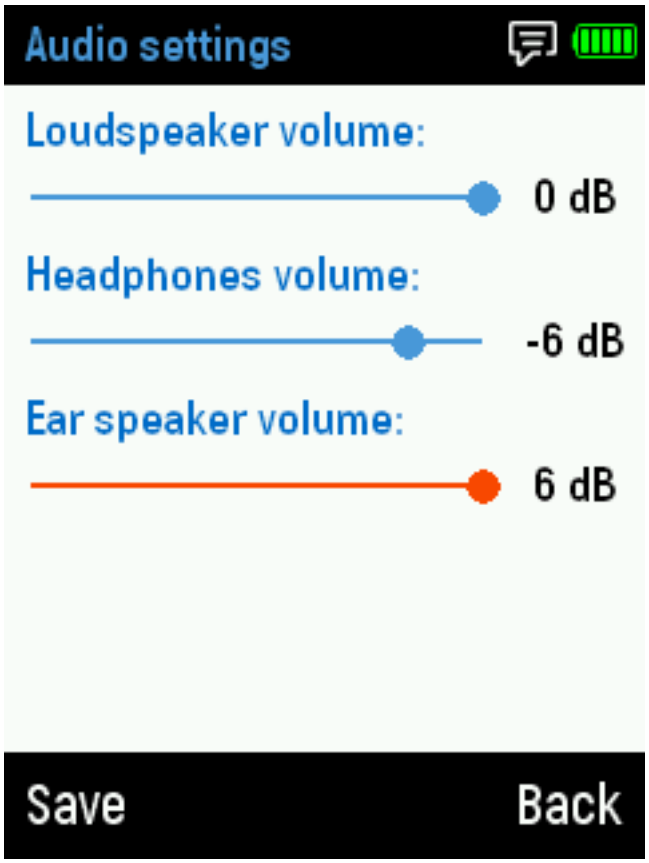


Fig. 3: Audio Settings.

Note: You can change volume of your audio channel during voice calls and while listening to music by **UP** and **DOWN** arrow keys.

2.8 Time Setting

By default Wiphone shows **Universal Time (UTC)**. You can search for time offset for your region and can add to time offset.

- Select **Menu** > **Settings** > **Time setting**.

Note: Switch the device off and disconnect the charger and any other device before removing any covers. Avoid touching electronic components while changing any covers. Always store and use the device with any covers attached.

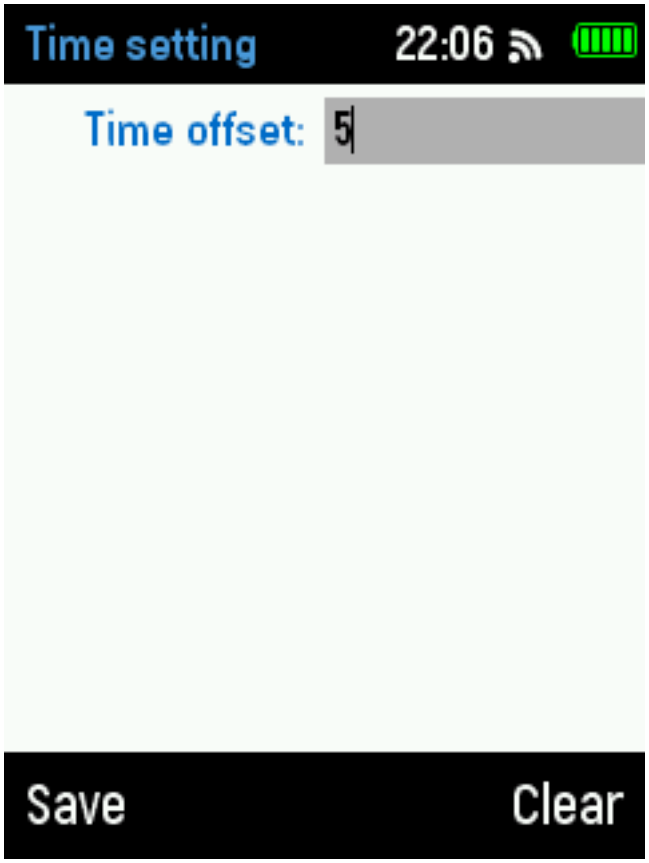
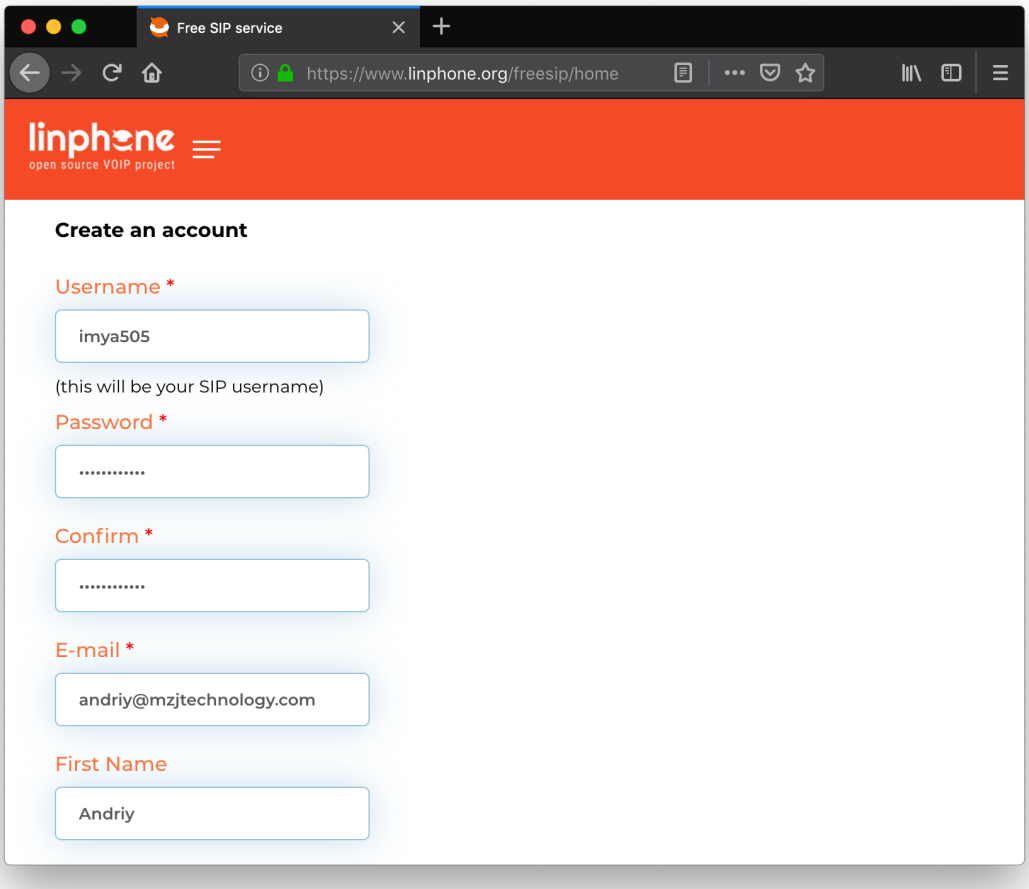


Fig. 4: Time Offset.

3 Setting up SIP account

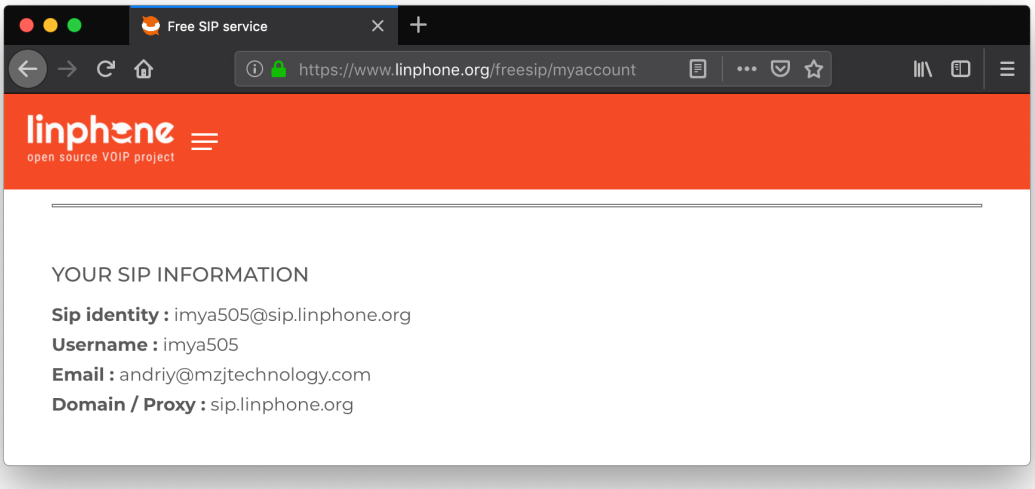
3.1 Registering at Linphone.org

1. Go to https://www.linphone.org/freesip/home in your web browser.
2. Choose username, pick a password, enter your email address and name:



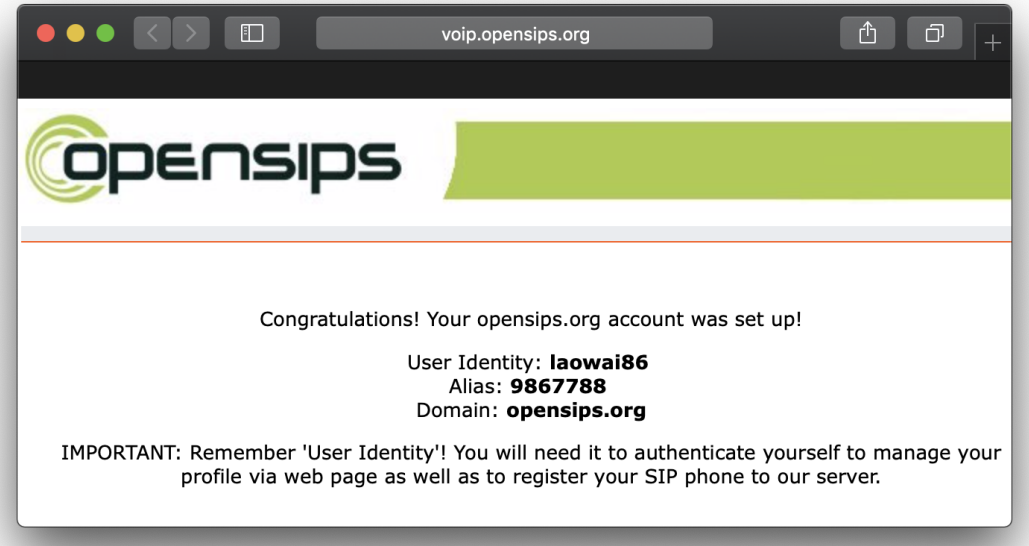
3. Confirm your email address by following a link in the received email.

4. Go to https://www.linphone.org/freesip/myaccount to see your SIP credentials.

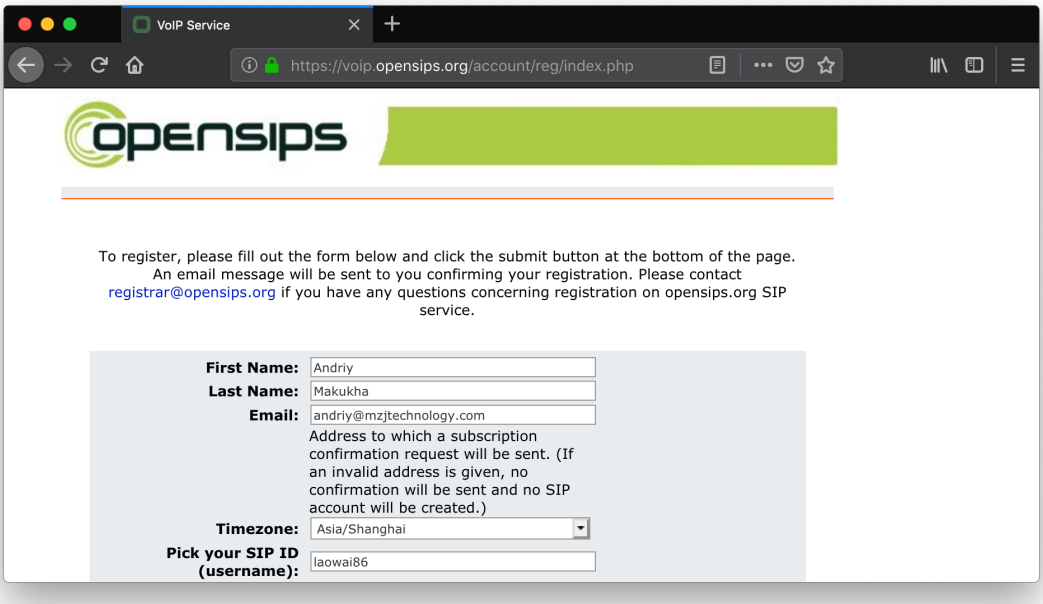


3.2 Registering at OpenSIPS.org

1. Go to http://voip.opensips.org/account/reg/index.php in your web browser.
2. Enter your name and email address, pick a username, SIP alias and password:



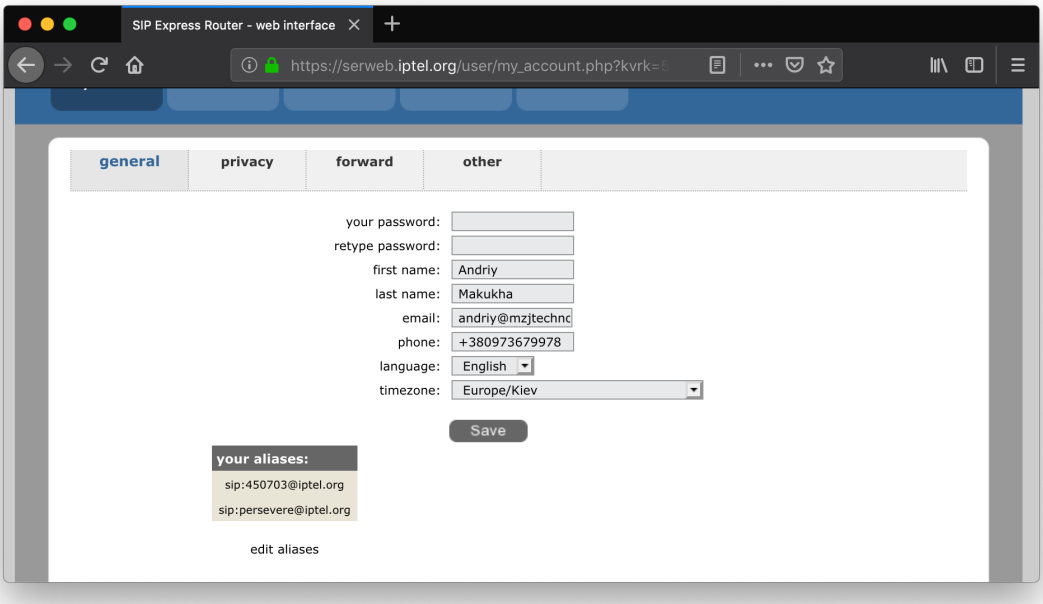
3. Confirm your email address by following a link in the received email.
4. Remember your SIP credentials from the opened window:

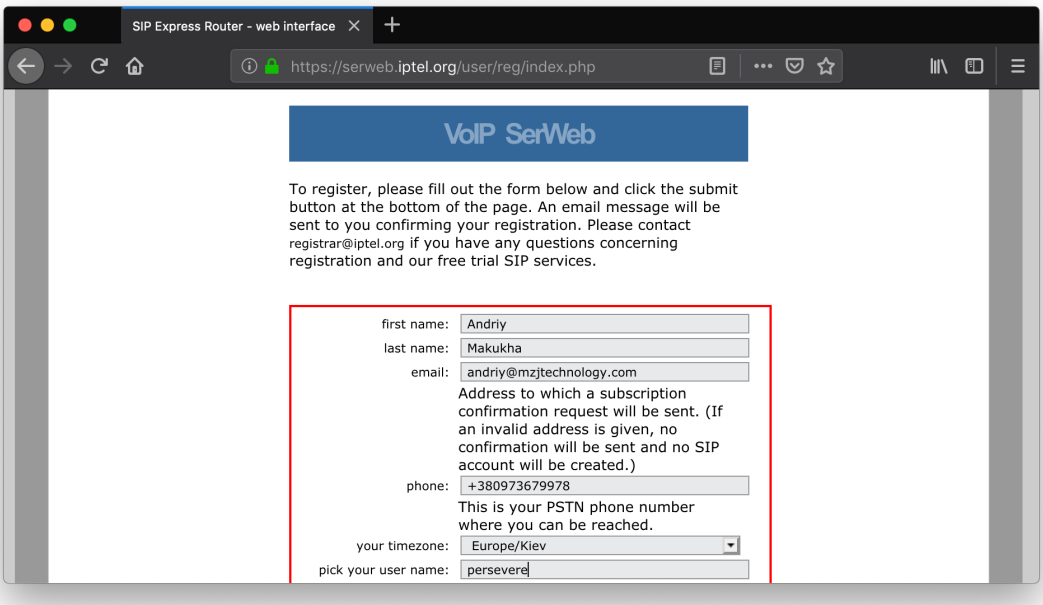


5. Go to http://voip.opensips.org/account/ if you wish to administer your account.

3.3 Registering at IPTel.org

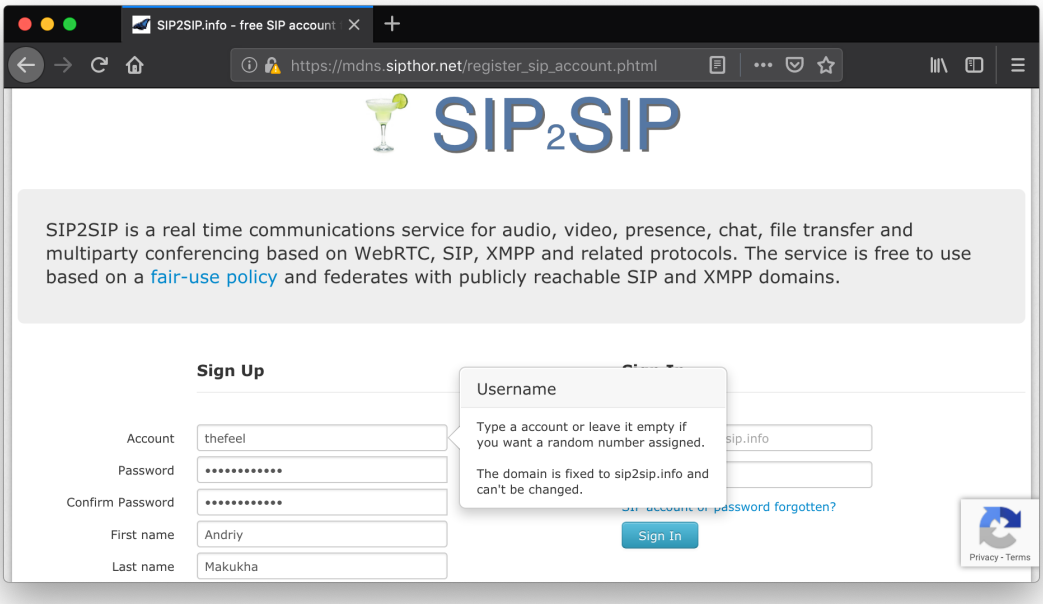
1. Go to https://serweb.iptel.org/user/reg/index.php in your web browser.
2. Enter your name, email, traditional phone number, pick your username and password:
3. Wait until your application is approved.
4. Confirm your email address by following a link in the received email.
5. Go to https://serweb.iptel.org/user/index.php and enter your username and password.



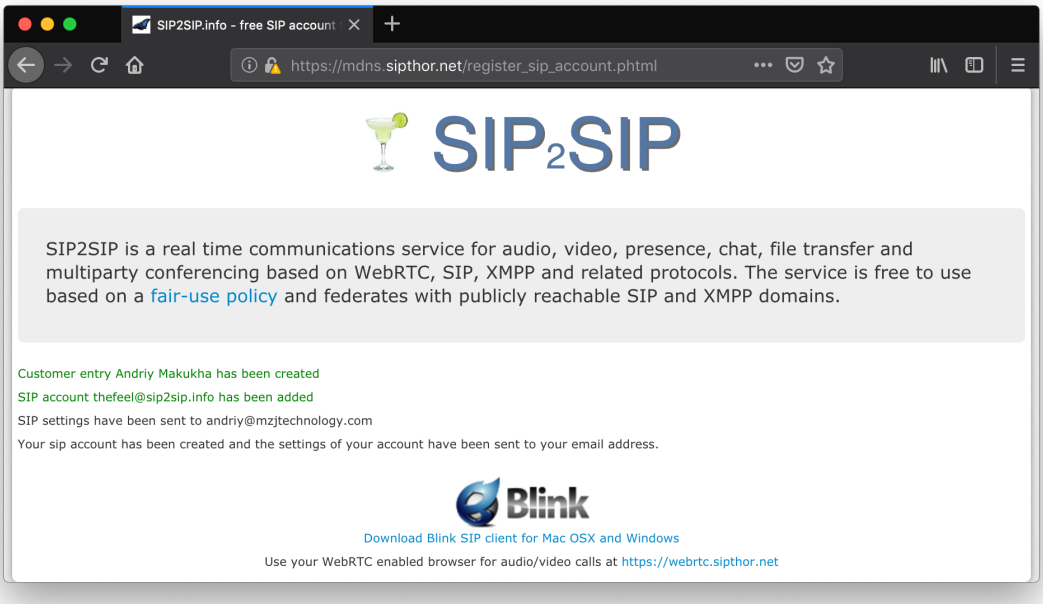


3.4 Registering at Sip2sip.info

1. Go to https://mdns.sipthor.net/register_sip_account.phtml in your web browser.



2. Pick a username and password, enter your name and email address:
3. Wait for an email with SIP settings.



3.5 Set your smartphone up to use your new SIP account

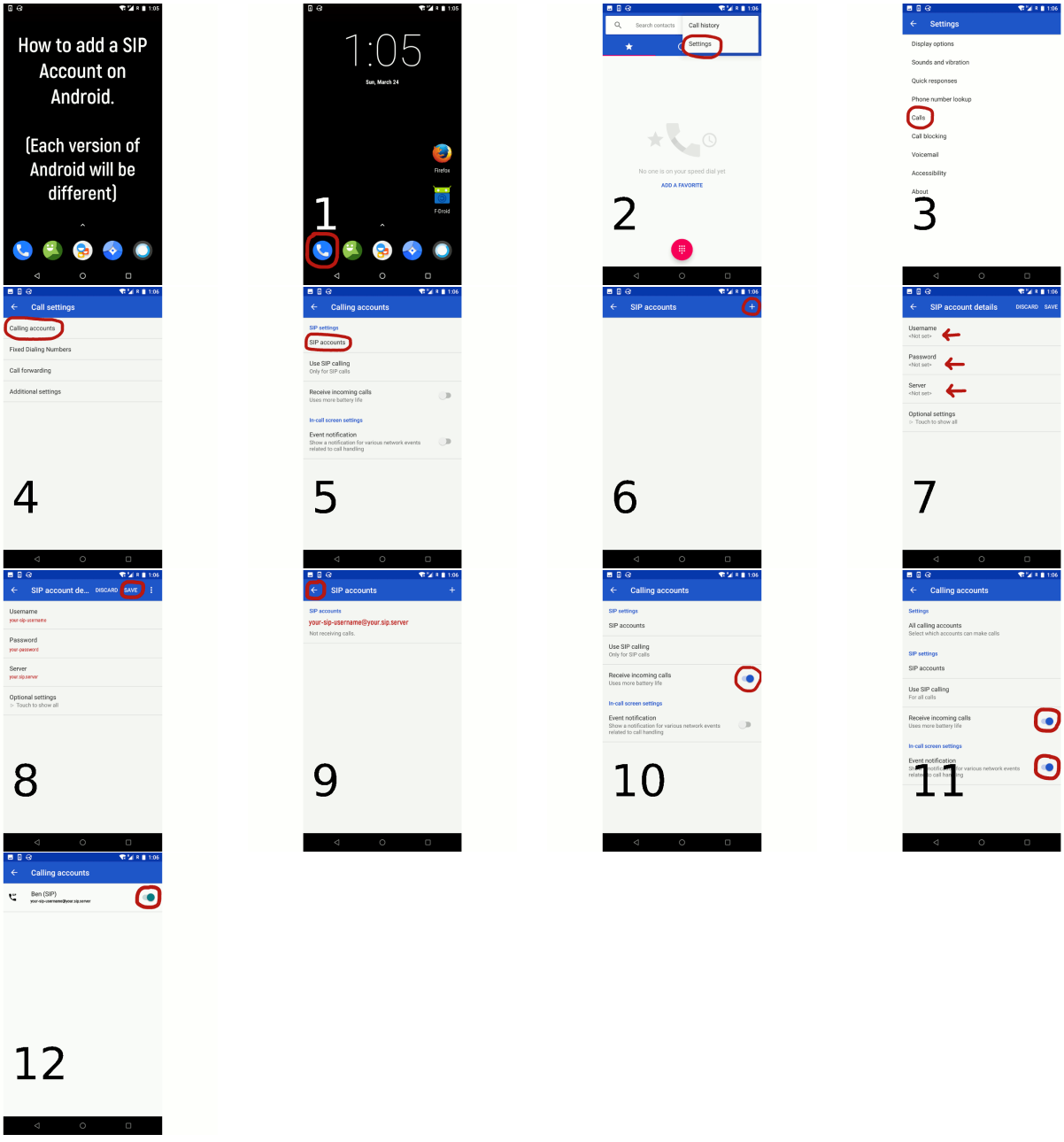
There are quite a few VoIP apps available. Typically you\'ll need your account name/user name, password, the server to connect to, and sometimes other info like which port to use.

LinPhone also has a nice one:

Google Play: https://play.google.com/store/apps/details?id=org.linphone

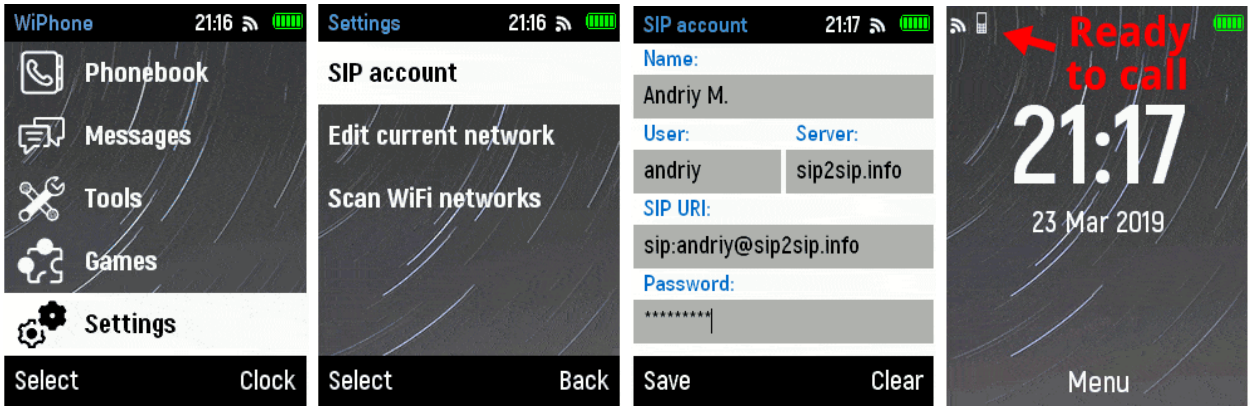
Apple App Store: https://itunes.apple.com/us/app/linphone/id360065638?mt=8

Or you can use the native Android client.



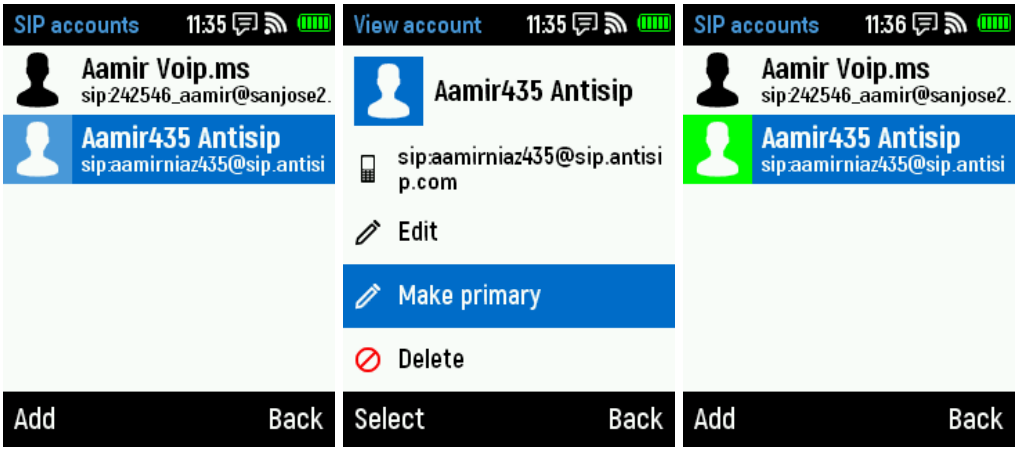
3.6 Log in on WiPhone with new SIP account

- Select Menu > Settings > SIP accounts > Add.



If you have multiple sip accounts you can choose primary account you want to connect to :

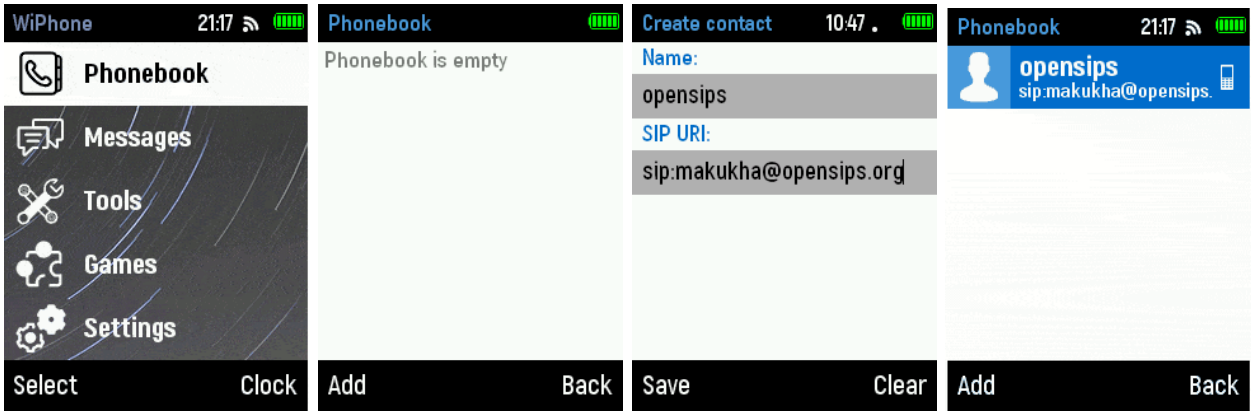
- Select **Menu** > **Settings** > **SIP accounts**.
- Press **OK** button on account you want to make primary and then select option **Make primary**.
- After you select your primary account and it connected. The color of Primary sip account will change to green.



4 Calls and Messages

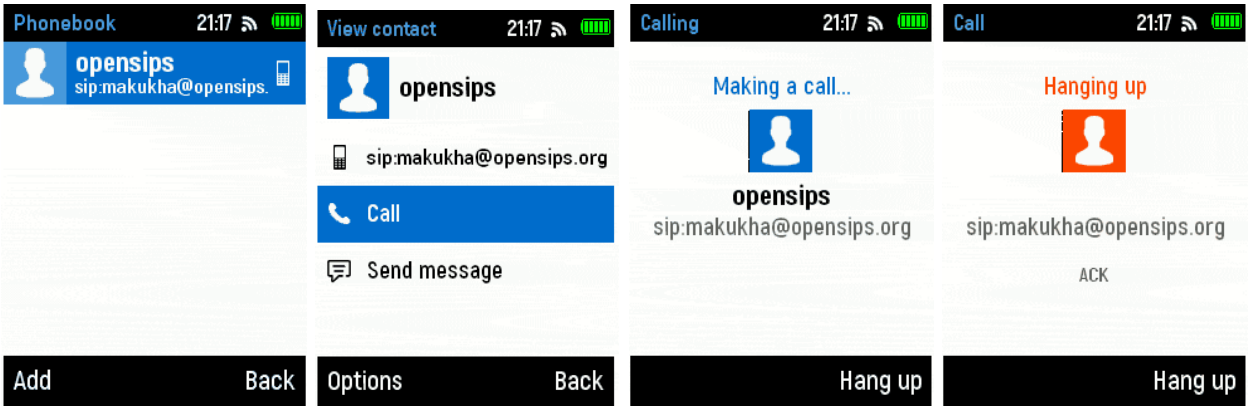
To make a call you need to save recipient sip information in contact. Saving contact information is simple :

- Select **Menu** > **Phonebook** > **Add**.
- Write name of the contact in **Name:** field.
- Enter SIP address to **SIP URI:**.
- Press **Save** key to save the contact.



4.1 Make a Call

- Select **Menu** > **Phonebook** > **Select Contact**.
- Press button to start dialing.Or you can Press **OK** and select **Call**.
- Press button to hang up call.

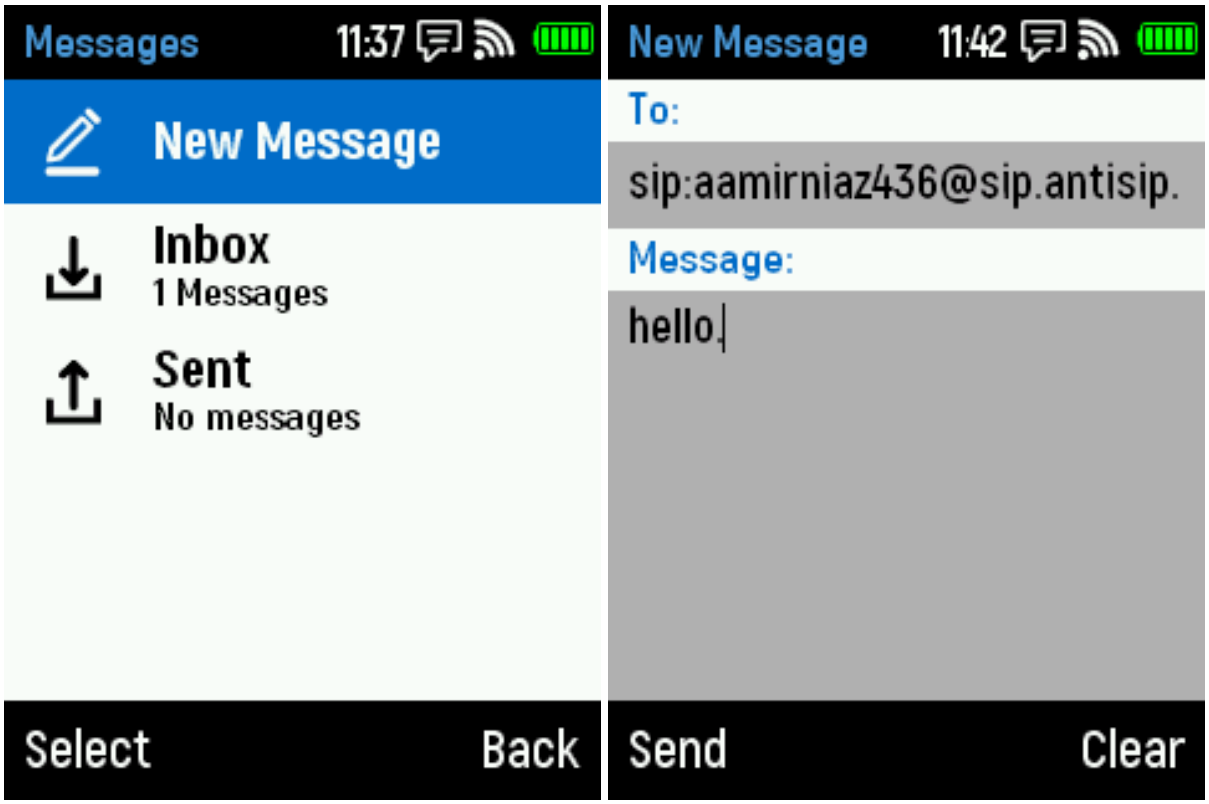


4.2 Receive a Call

- Press  button to Accept Call.
- Press  button to Reject Call.

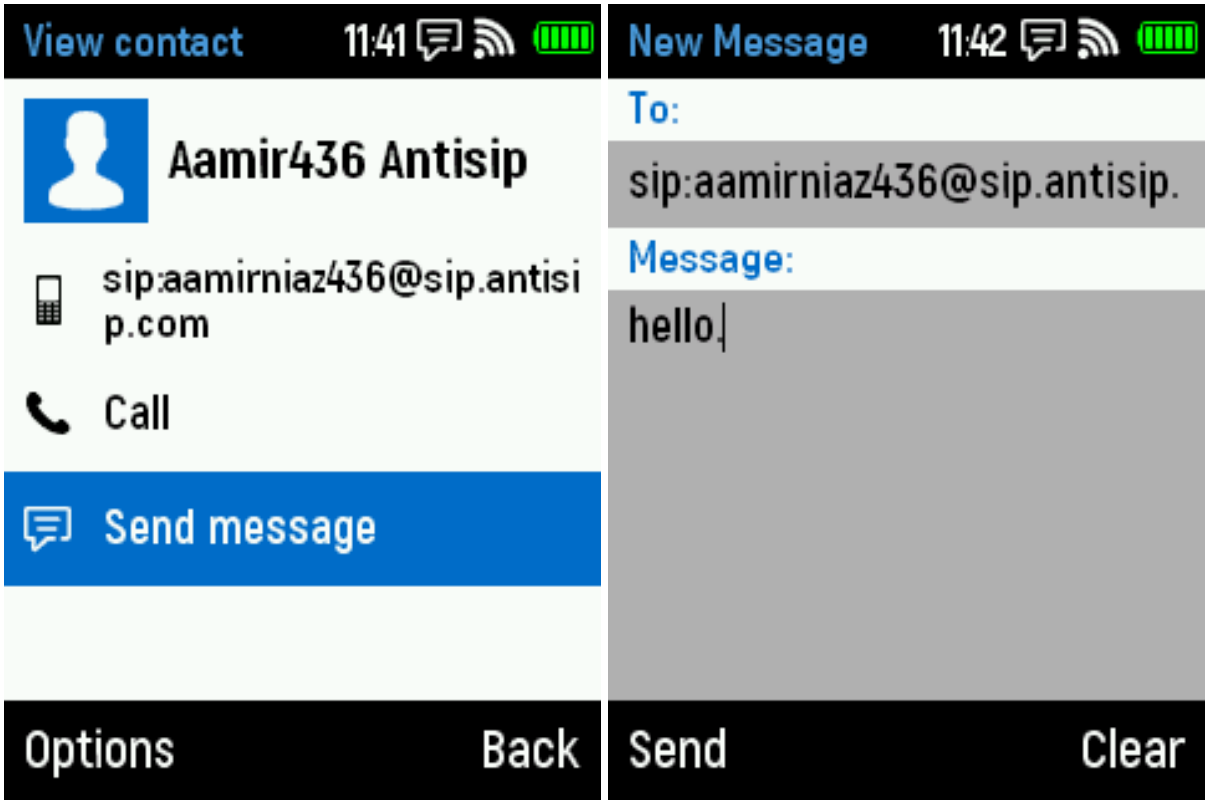
4.3 Messages

- Select **Menu** >  **Message** >  **New Message**.
- Enter sip addressof message recipient in **To:** field and Enter text in **Message:** field.
- Press **Send** button to send message.



OR

- Select **Menu** > **Phonebook** > **Select Contact**.
- Select **Send Message** and write text.
- Press **Send** button to send message.



5 WiPhone Programming Instructions

5.1 Prerequisites

5.1.1 Hardware

For compiling and editing WiPhone firmware, you will need a computer capable of running Arduino Desktop IDE (with either Linux, MacOS or Windows operating systems) and a microUSB cabel.

5.1.2 Software

The following software packages are required to compile WiPhone firmware:

1. Arduino Desktop IDE https://www.arduino.cc
2. Arduino core for WiPhone (WiPhone Arduino) https://wiphone.io/static/releases/arduino_platforms/WiPhone0.1.0.zip
3. Arduino plugin for uploading files to ESP32 file system https://github.com/me-no-dev/arduino-esp32fs-plugin

If you want to contribute to the firmware, you should also have the Git source-control management tool: https://git-scm.com/

5.2 Installing Arduino Desktop IDE

Download an installer for your platform from the official Arduino website: https://www.arduino.cc

Download page: https://www.arduino.cc/en/Main/Software

Follow the installation procedure for each operating system:

- For Linux, download the archive, extract it, run file ”install.sh”.See official instructions for more details: https://www.arduino.cc/en/Guide/Linux

- For MacOS, copy the file from downloaded archive into the Applications folder.See official instructions for more details: https://www.arduino.cc/en/Guide/MacOSX
- For Windows, download and run the installer.See official instructions for more details: https://www.arduino.cc/en/Guide/Windows

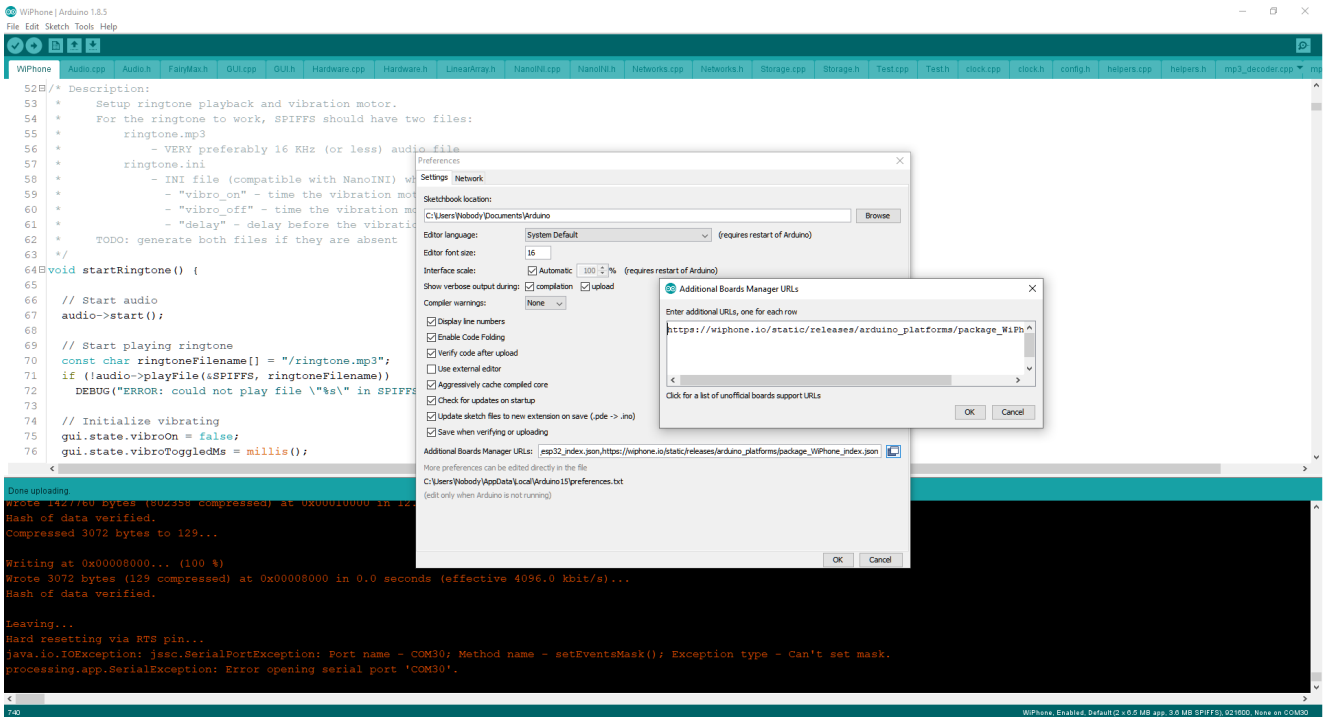
5.3 Installing WiPhone-Arduino using Arduino IDE Boards Manager

If your Arduino IDE is recent enough (ver. 1.8+), you can install Arduino-ESP32 with Arduino IDE's Boards Manager:

1. Start Arduino Desktop IDE.
2. Open Preferences window.

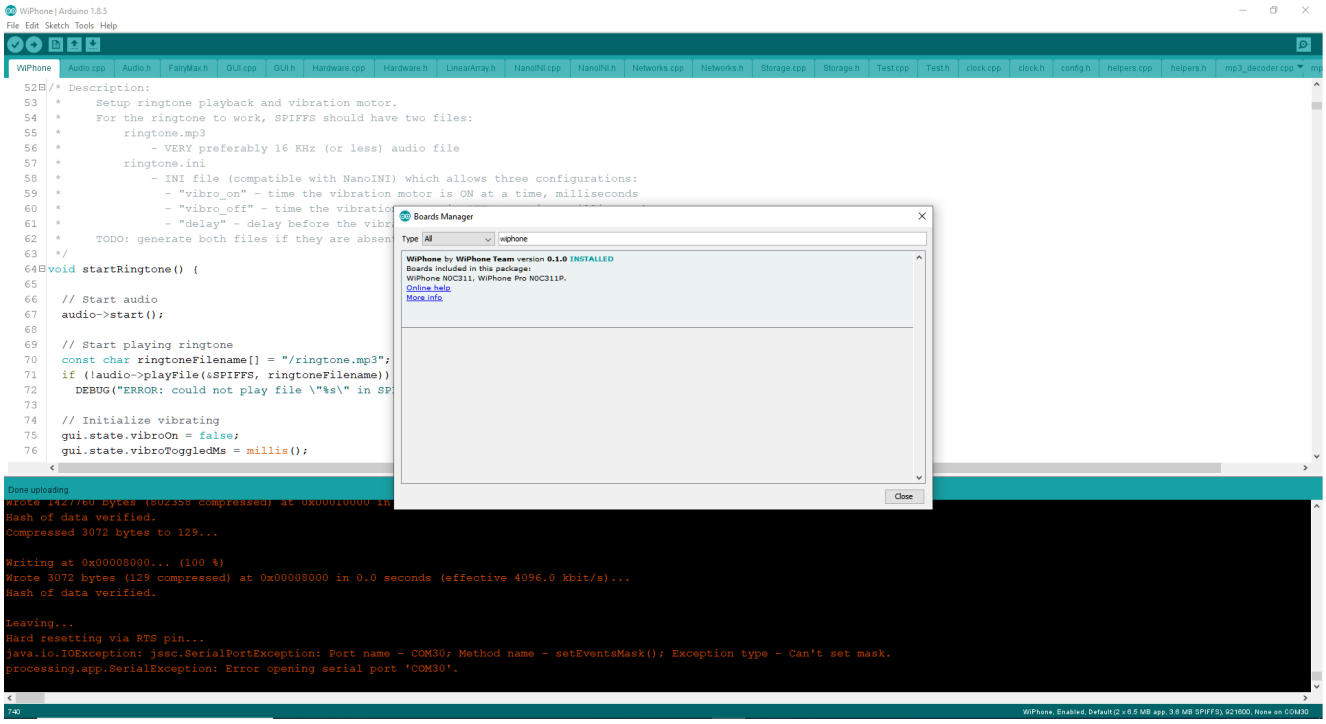
3. Enter "https://wiphone.io/static/releases/arduino_platforms/package_WiPhone_index.json"URL (without quotes) into the *Additional Board Manager URLs* field.

Note: You can add multiple URLs, separating them with commas.



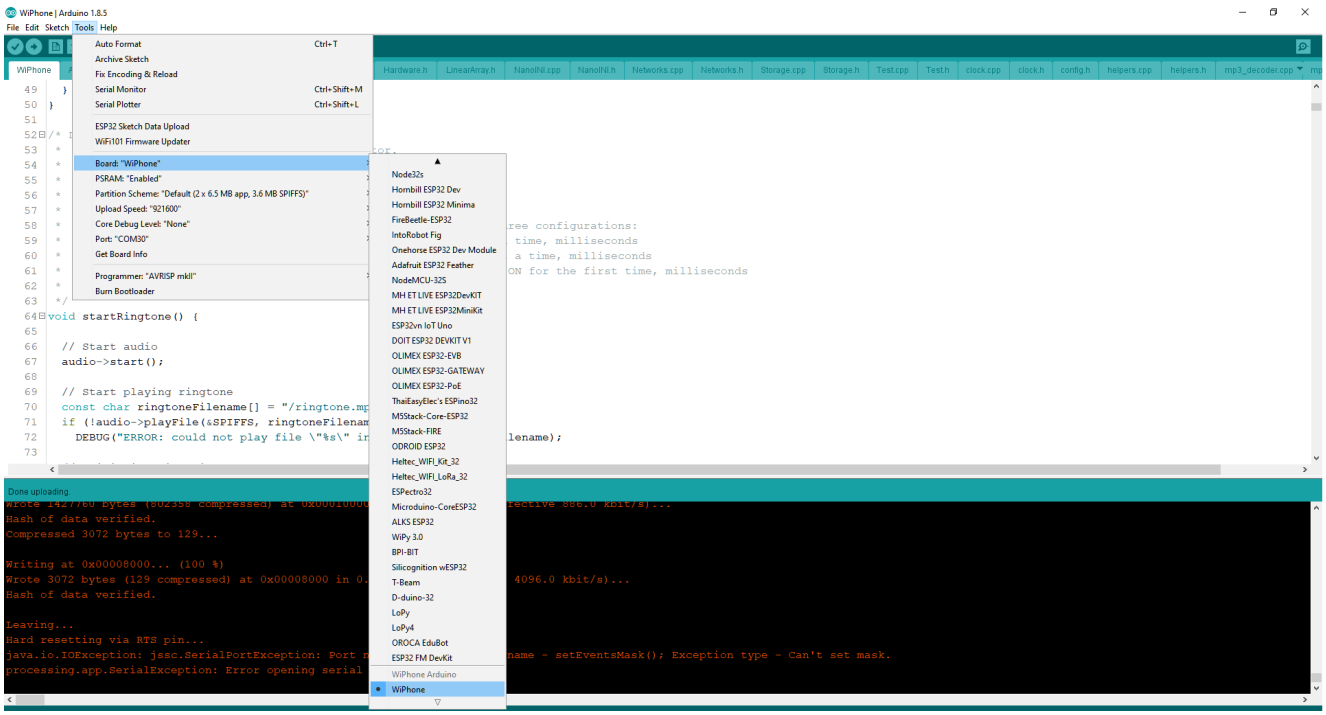
4.Open the *Boards Manager...* window by navigating to *Tools > Board: >Boards Manager...*

5.Search for "wiphone"and install **WiPhone** package by **WiPhone Team**.



5.3.1 Choosing a board type

After installing WiPhone-Arduino, select WiPhone board by navigating to *Tools* > *Board: ...* menu.



WARNING: Changing partition scheme might overwrite the internal flash file system (SPIFFS) causing irreversible data loss of WiPhone data files (like phonebook and SIP accounts). Stick to a single partition scheme early in the development to avoid data loss.

5.4 Installing Arduino plugin for uploading files

This plugin is required to upload data files to WiPhone's internal flash (SPIFFS). It is needed, for example, to load the ringtone file and configuration. It also formats the SPIFFS partition of ESP32 to allow storing WiPhone data (like phonebook and SIP accounts) into the internal flash.

WARNING: Loading files with this plugin into WiPhone will overwrite any existing files in the SPIFFS partition, causing irreversible data loss. Use this only if you know what you are doing!

Installation procedure:

1. Download the plugin archive: https://github.com/me-no-dev/arduino-esp32fs-plugin/releases/download/1.0/ESP32FS-1.0.zip

2. Extract the archive and copy the extracted directory ESP32FS into the tools subdirectory of Arduino sketchbook directory:

- On Linux, create directory ~/Arduino/tools/ (you can run "mkdir -p ~/Arduino/tools/"in terminal) and copy the ESP32FS direcory into it.
- On MacOS, create directory ~/Documents/Arduino/tools/ (you can run "mkdir -p ~/Documents/Arduino/tools/"in terminal) and copy the ESP32FS direcory into it.

Consult the official installation instructions as well: https://github.com/me-no-dev/arduino-esp32fs-plugin#Installation

5.5 Obtaining a copy of firmware

In future, we will publish the firmware on GitHub.

For now, please, request a copy of the firmware from support@wiphone.io

5.6 Compiling and uploading firmware

After obtaining firmware, make sure that all of the files and subdirectories are stored in a directory called 'WiPhone". Then, to compile and upload the firmware to WiPhone:

1. Connect WiPhone to your computer with a microUSB cable (the cable should have microUSB plug on one end to connect to WiPhone, and any other connector that is compatible with your comnputer's sockets on another end, like USB-C or USB).

2. Open the file WiPhone.ini from the project directory in Arduino IDE.

3. Press button "Upload"in the top left corner of the Arduino IDE window (the button looks like a right arrow).

5.7 Uploading data files

WiPhone allows data files to be stored permanently in internal Flash file system (SPIFFS). Particularly, this partition is used to store ringtone music file and configuration.

To upload data files from subdirectory ”data”to SPIFFS, do the following:

1. Make sure that the plugin for uploading files to ESP32 is installed(otherwise, see the dedicated section above).
2. Navigate to ”Tools”menu in Arduino IDE and press ”ESP32 Sketch Data Upload”.

5.8 What's in the project directory

Here are some files and subdirectories from the source directory:

- **WiPhone.ino**

  Arduino project file: a C++ file with the main loop and the main phone logic

- **GUI.h**
- **GUI.cpp**

  main GUI logic files (widgets and apps are defined here)

- **tinySIP.h**
- **tinySIP.cpp**

  our implementation of the SIP protocol

- **src/**

  directory for code that is not in active development (these files are crucial for WiPhone's operation, but will not be opened by Arduino IDE by default, unlike files in the main directory)

  - **src/TFT_eSPI/**

    a library for ESP32 to drive the screen at fast speed; includes sprites and font rendering capabilities

  - **src/VoIP/**

    audio codecs and other definitions for Voice-over-IP

  - **src/assets/**

    static data components (such as fonts, icons, background image, etc.)

  - **src/drivers**

    code that operates different integrated circuits (ICs) of the WiPhone

- **data/**

  files to be loaded into WiPhone □ s internal flash file system (SPIFFS)

- **tools/**

  custom Python scripts to generate ”assets”, namely 3-bit antialiased fonts and icons

- other WiPhone source files in the main directory

5.9 Developing custom ”apps”

Developing custom apps in WiPhone's Arduino firmware is fairly easy and straightforward. Even a beginner programmer with knowledge of C++ should be able to figure it out after learning how existing apps work. Having said that, there are some non-transparent parts that will be covered and discussed in this chapter. As well as some principal design disadvantages that will limit the way apps can be developed (see the disclaimer below).

5.9.1 Disclaimer: single thread

As of September 2019, WiPhone's firmware is not organized as a full-blown operating system with threads and scheduling. We are not using FreeRTOS threads due to instability concerns that couldn't be overcome in our early trials. It was, thus, decided to stick to cooperative multitasking approach as a workaround. Therefore any custom apps that are developed for WiPhone must follow this approach and must never lock or take more than 10-100 ms of CPU time per call in order to make the interface responsive. (Yes, we know that it's not cool. You are welcome to propose a solution.) Also, the biggest drawback of this single-threadedness is that entire phone will appear to be frozen if non-asynchronous connection is attempted, but cannot be established rapidly or at all... This is one of the biggest bugs of the firmware.

5.9.2 Steps for adding a custom app

1. **Declare a unique ID for the new app.** For this purpose, add a new element to the ActionID enumeration:

   Find ”typedef enum ActionID”in file ”GUI.h”. Add a unique name below the line ”GUI_BASE_APP = 0x4000”. For example, ”GUI_APP_NEW_CUSTOM”.

2. **Declare the app class.** The easiest way to do this is to find an existing app similar to the one you want to build, copy its code, change class name and then change that code to do the things you want.

   The app class declaration goes into ”GUI.h”file, definitions of its methods go into ”GUI.cpp”.

   The app class must derive from ”WiPhoneApp” or of its derivatives (like ”WindowedApp”or ”FocusableApp”, or both).

   Make sure the method ”getId”returns the ID that you created in the first step.

3. **Define the app methods.** Steps 2 and 3 are the actual app implementation. This is usually the biggest and longest step.

   This is done in file ”GUI.cpp”.

   Each app defines two main methods: ”processEvent()”and ”redrawScreen()”.

   ”processEvent(EventType event)”is the method that is called by GUI to allow your app process events and update its internal state accordingly. The events are, for example, button presses, timers (as requested by your app) or scheduled events (as scheduled by your app).

   ”redrawScreen(bool redrawAll)”is the method that is called by GUI to allow your app to redraw the screen partially and/or telling your app that it should redraw the screen entirely (like after the screen was redrawn by some other app, like a call or a screen lock).

4. **Add your app ID into the main menu.**

   For that purpose, find ”GUIMenuItem menu” in file ”GUI.h”. Increase size of the array by 1. Add a line of the form ”{ XX, YY, ”My custom app\”, \”\”, \”\”, GUI_APP_NEW_CUSTOM },”into the definition of the array ”menu”; XX– should be a unique ID of the menu item (for simplicity, it should be the current size of the array), YY – is parent ID, or ID of the submenu, in which you want your app to appear.

5. **Instantiate your app.** This is, basically, just letting the GUI know how to call your app s constructor.

   Find definition of method ”GUI::enterApp()”in file ”GUI.cpp”and add code to create a new object out of your app class. (Do it similarly to other apps.)

6 Firmware Download Tool Instructions

To Download Binaries to WiPhone you will need **Flash Download Tool** by Espressif. Link to Download Flash Tool is : https://www.espressif.com/en/support/download/other-tools

After launching the setup file, select ESP32 Download tool.

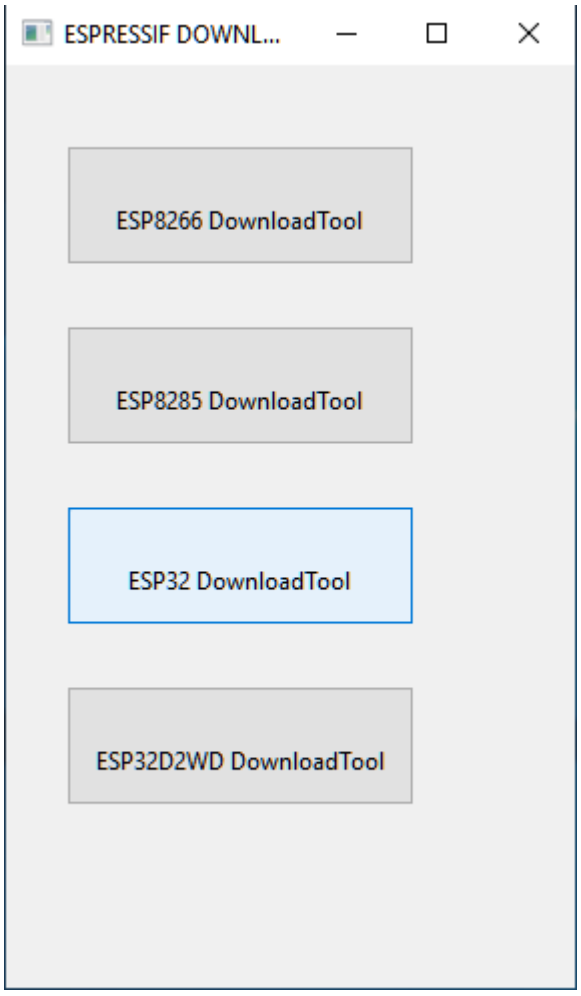


Fig. 1: ESP32 Download Tool selection.

After selecting The ESP32 Download tool a new window will appear with SPI download window.

You will need to select the binaries you want to upload to WiPhone. WiPhone Firmware consists of 5 binaries files.

.bin	Address
boot_app0.bin	0xe000
bootloader_dio_80m.bin	0x1000
WiPhone.ino.bin	0x10000
WiPhone.ino.partitions.bin	0x8000
WiPhone.spiffs.bin	0x00c90000

Latest Binaries can be found at : https://wiphone.io/downloads

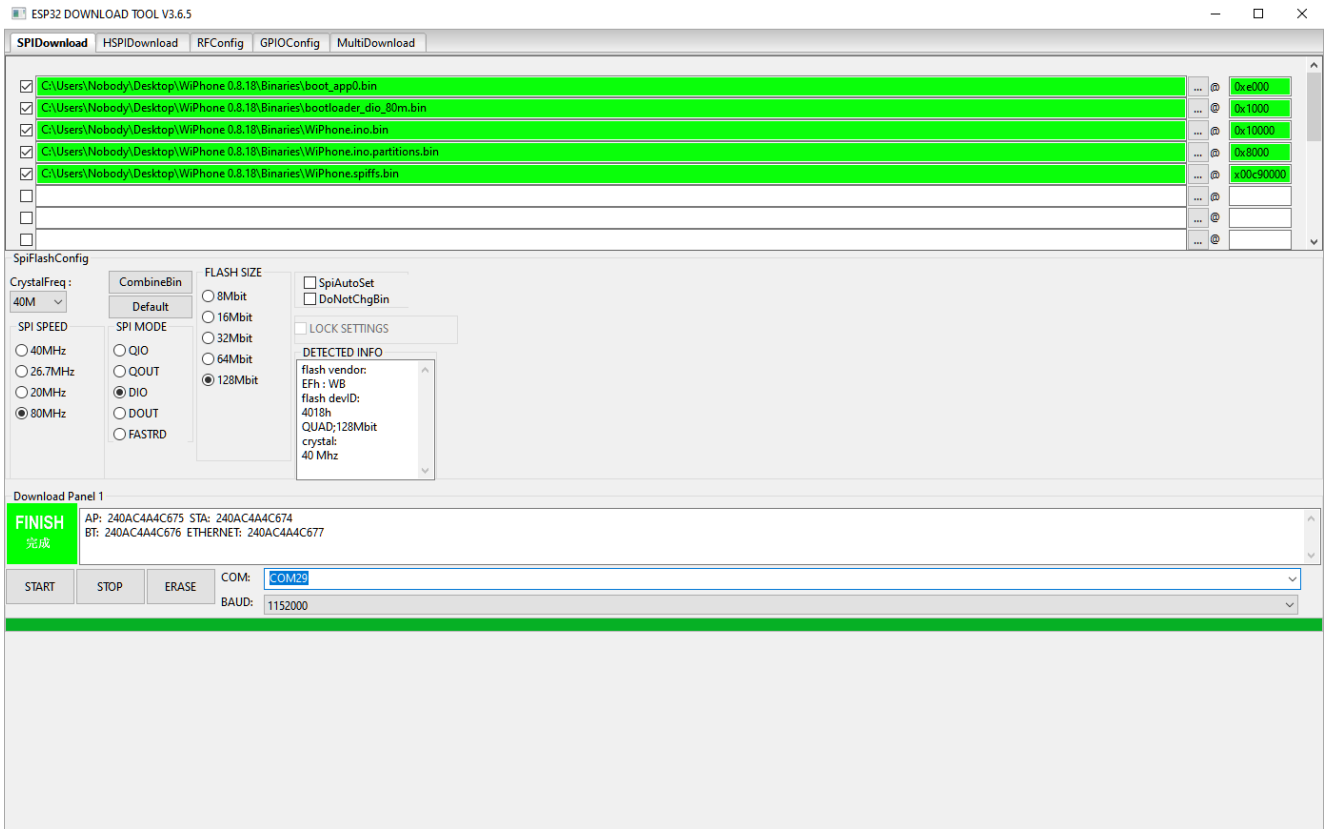


Fig. 2: SPI Download Config.

Select the path to binaries on your PC and select following options :

- CrystalFrq : 80MHz.
- SPI Mode : DIO.
- SPI Speed: 40MHz.
- Flash size: 128Mbit.
- COM : Choose serial port.
- BAUD : Choose the download speed. (Select 1152000 to speed up the download process)

Press the ”START” button and wait until the process complete and you see ”FINISH” sign on Download panel. After successful download, unplug the WiPhone from the computer and press the  for 10s to force shutdown the WiPhone. Press  button again and you WiPhone will bootup with updated firmware.

7 Daughter Board Design Guide

To make it easier for people to design their own daughter boards, We have created the footprint for daughter board. Using which you can easily start making your own daughter boards without considering many design constrains and running into problems.

Most of our daugther boards are designed in Eagle PCB software. (Right now we provide footprint for Eagle PCB software and shortly we will provide footprint for Altium PCB software).

We will walk through the step of designing daughter board and information about different layer in Daughter Board Footprint.

7.1 Schematic Footprint

The Schematics footprint of daughter board is simple and it includes landing pads for POGO pin connectors, A4 size design sheet and for the convience of programmers we have added table to top left side which basically maps the software contrable pins of daughterboard to the arduino pinouts.

There is an excel sheet **DaughterBoard Pinout v2.3.xlsx** with detailed description and functionality of each pin of daughter board. You can download at https://wiphone.io/downloads.html

7.2 PCB Footprint

To design a daughter board PCB which looks great, works without any problem and easily fit on the back of WiPhone, There are lot of things need to happen correctly. But Thanks to PCB footprint for daughter board which already take those things into account and you can just copy PCB footprint into your software and can immediately start your components placement and rounting without running into many problems.

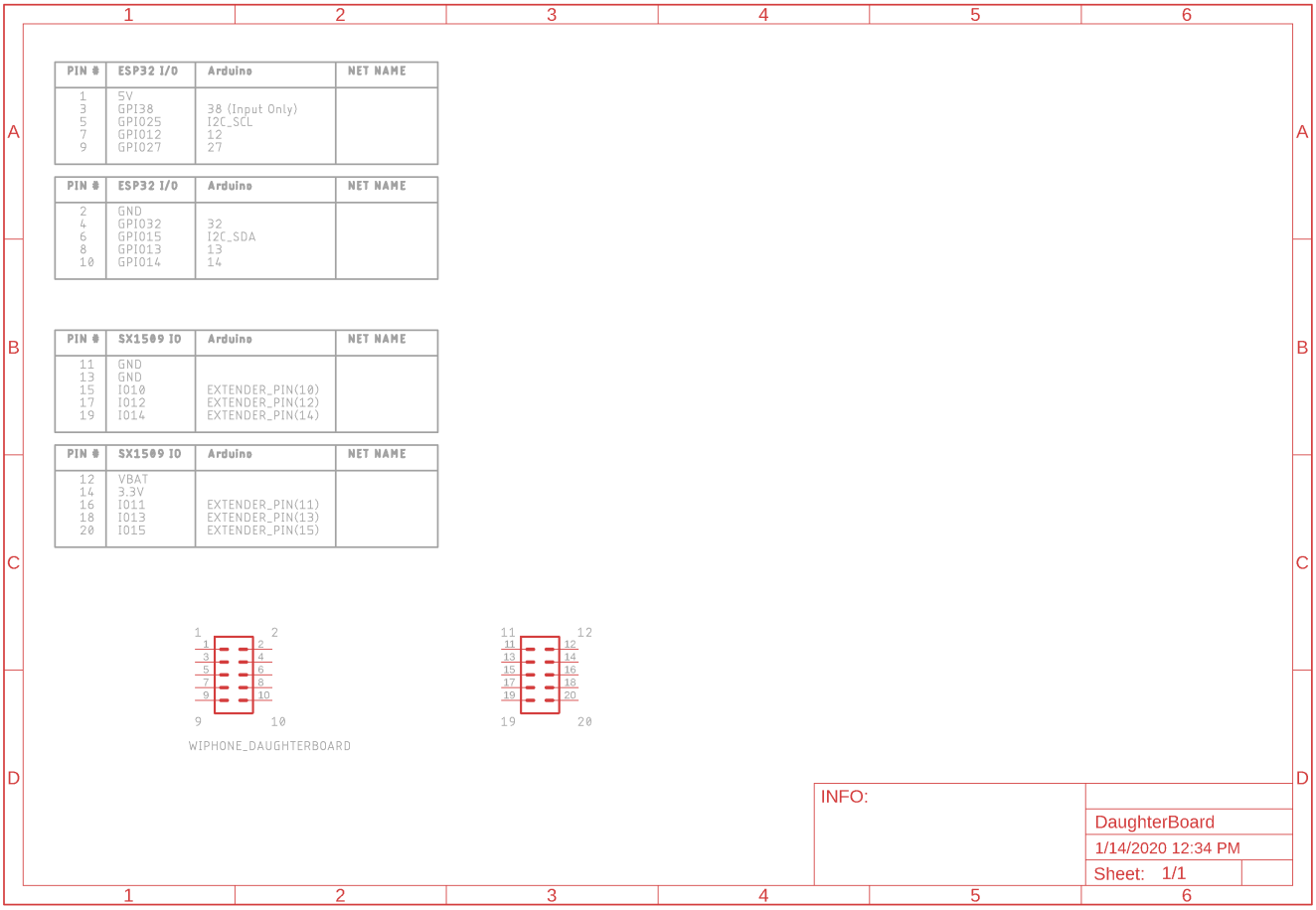


Fig. 1: Daughter Board Schematics Symbol.

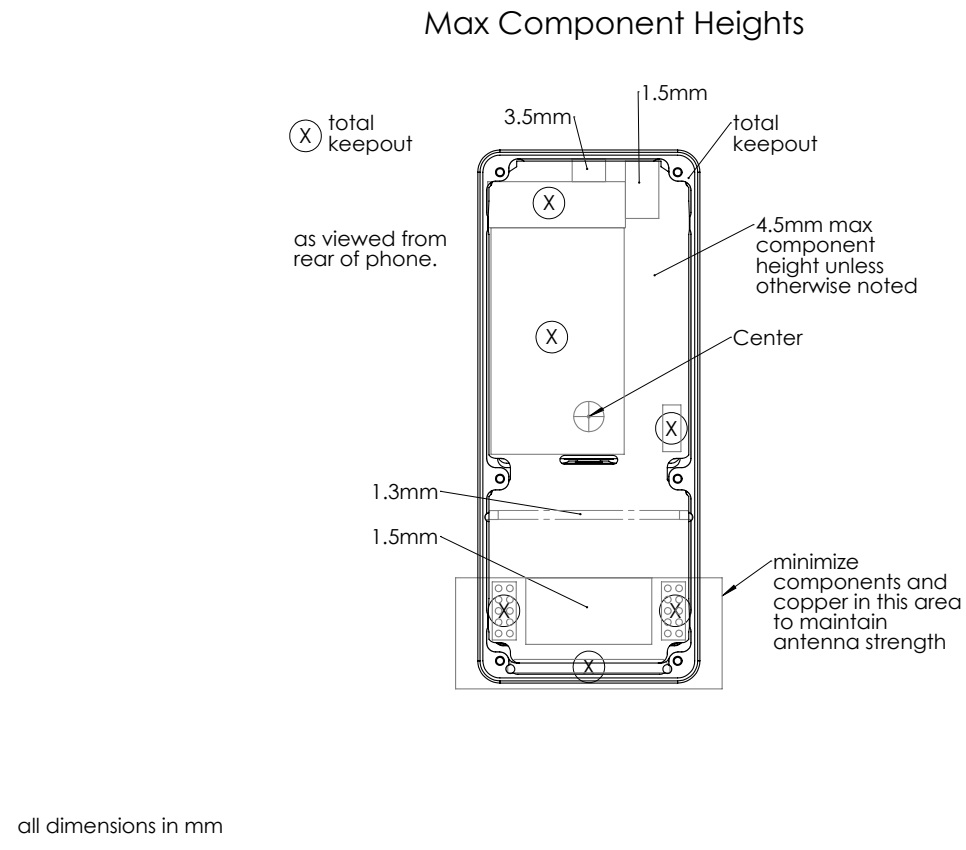


Fig. 2: Daughter Board Keepouts.

7.2.1 Daughter Board Outline

Daughter board dimensions are **117.5*47.5*1.6mm** with six holes for mounting screws. Ideally we want to put all the components on the inside layer of daughter board so that we clear nicer looking back of WiPhone and without having any fragile components on the backside.

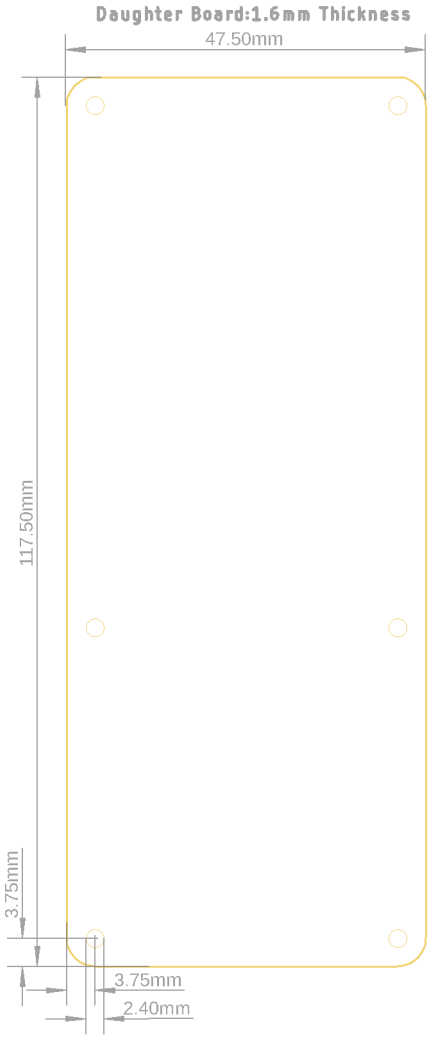


Fig. 3: Daughter Board Dimensions.

7.2.2 Bottom Components Keepout Area

To place components to the inner side of daughter board we need to take care that components on the daughter baord don't hit with components of WiPhone mainboard.

7.2.3 Bottom Copper Restrict Area

For components placement on inner layer we made layer specail layer into PCB footprint with name **BottomKeepouts** which will tell you max height components at different areas. As you can see from the above picture, We defined total keepout area for battery,speaker and vibration

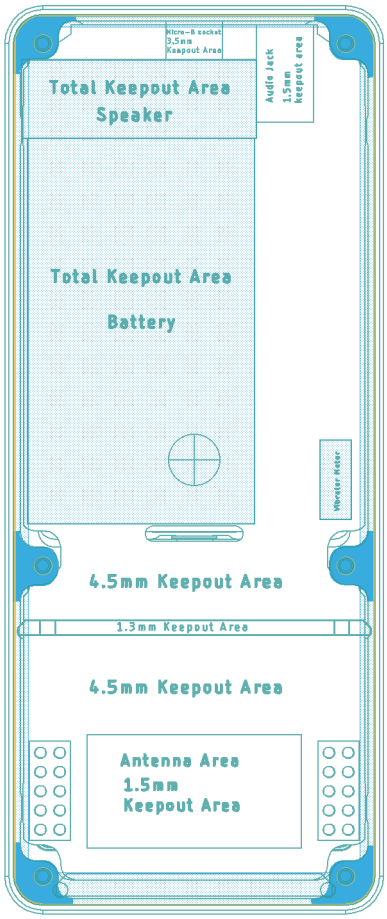


Fig. 4: Daughter Board Bottom Keepouts.

motor. As these are the places where components on bottom side of daughter board can bump into speaker on mainboard, poke into battery of mainboard and give nice smoke feature to your WiPhone :) or vinration motor. So as you can see from above image we also placed bottom keepout to those places so that if you accediently put components to those places your DRC will complain about component placement to those areas. As the daughter board will rest on the frame of WiPhone, so there is small keepout areas near PCB edges and near mounting holes. Other locations like right over Antenna backer there is keepout area of 1.5mm i.e. components with height more than 1.5mm can't be placed in that area. Other areas are 4.5mm keepout areas unless specified.

As already discussed that the daughter board will be placed over the frame of WiPhone and to avoid any expose copper and conenction between daughter board and frame of WiPhone, we have defined bottom restrict layer which will keep safe distance between copper pour or copper traces of daughter board and WiPhone frame.

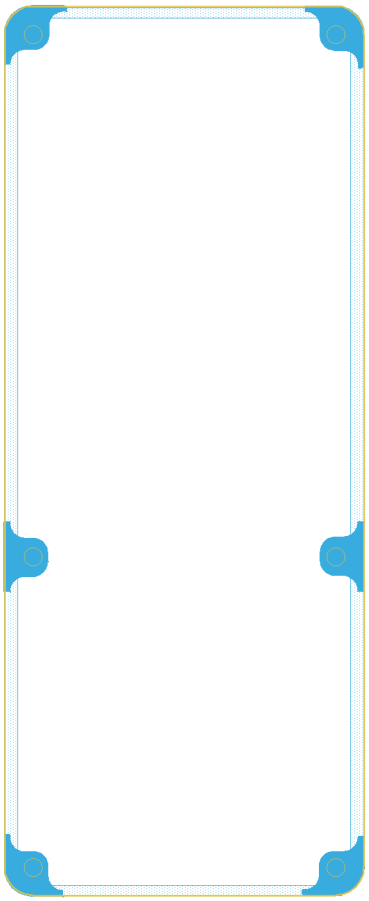


Fig. 5: Daughter Board Bottom Restrict.

7.2.4 Bottom Document Information

Bottom Document *bDocu* layer contain the small table with daughter board pin description so that you don't need to switch between PCB and Schematics to know the pin functionality of daughter board connectors.

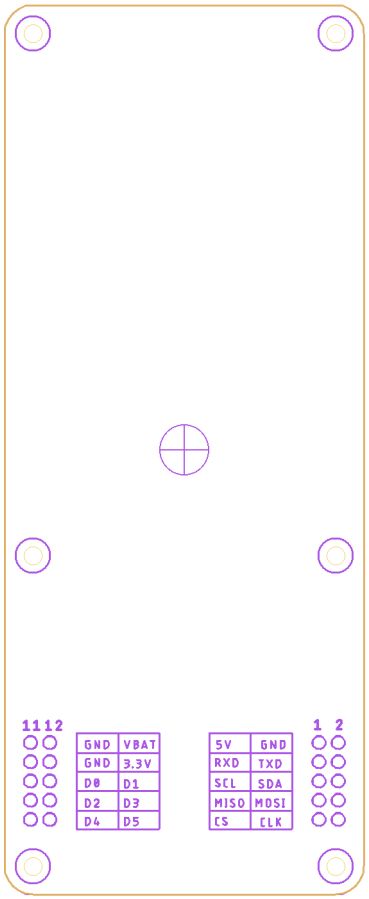


Fig. 6: Daughter Board Bottom Document.

7.2.5 Bottom and Top Copper Keepout Area for Antenna

This is something you need to make sure while designing daughter board that there should be not any unnecessary copper pour and traces in this area so that it will not affect the performance of WiPhone antenna.

7.2.6 Top Components Keepout Area

This may not apply to all daughter board because this keepout will be only required if you are going to put the hard shell case to the back of WiPhone. There are some daughter boards like

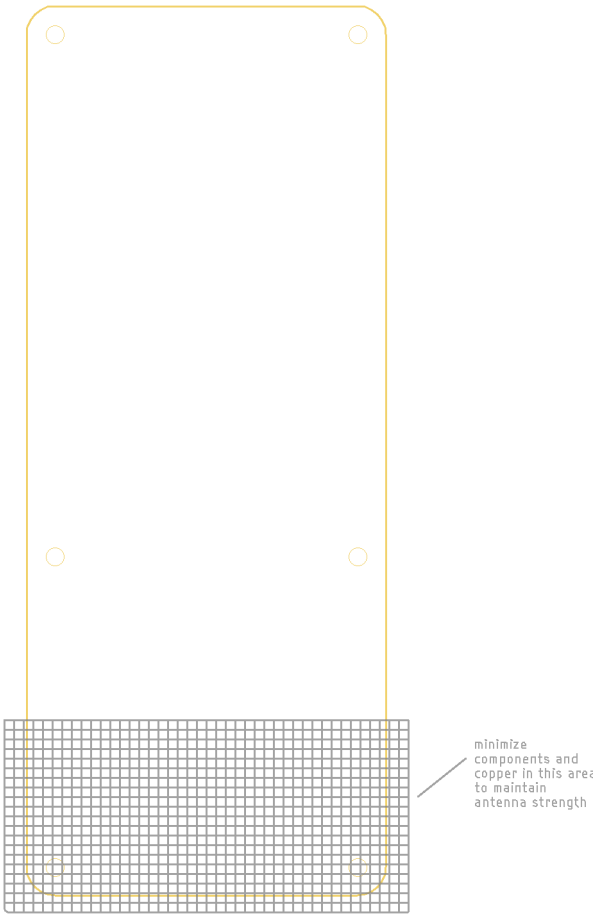


Fig. 7: Daughter Board Copper Keepout Area for Antenna.

4G LTE, 2G GSM/GPRS, LoRa and Mega Battery Pack with use casing on top of daughter board.Use can see examples on our website. So if you are going to use casing with your daughter board make sure there are not any components is this area.

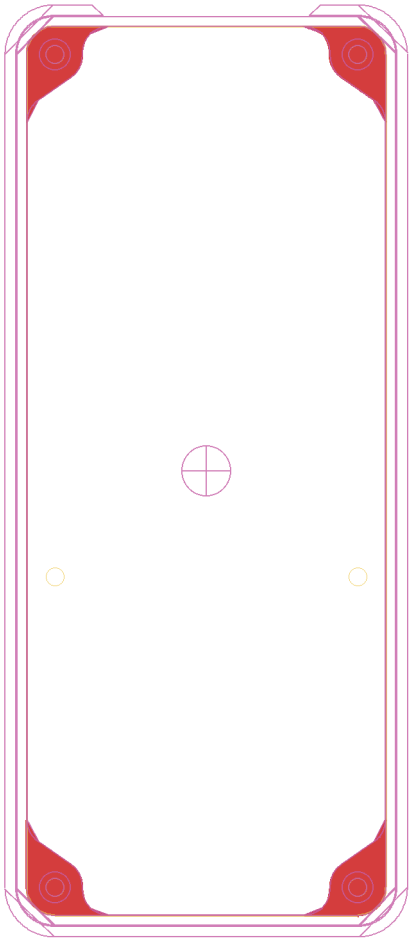


Fig. 8: Daughter Board Top Keepouts.

7.2.7 Top Copper Restrict Area

To give nicer look to WiPhone, we suggest to use counter sink holes for daughter board. For that you need to make sure that while making counter sink holes the copper area will be not expose and may cause problems later.So we need to keep safe distance from the board edges and holes.

7.2.8 Top Document Information

To standardize the look of daughter boards on tDocu layer we defined the box outline to put the daughter board name. While producing many daughter boards with vector and proportional fonts we feel that they don't look nicer and we opt to import bitmap image for daughter board names. Eagle is not really good with importing bitmap images so we have provided you **Board**

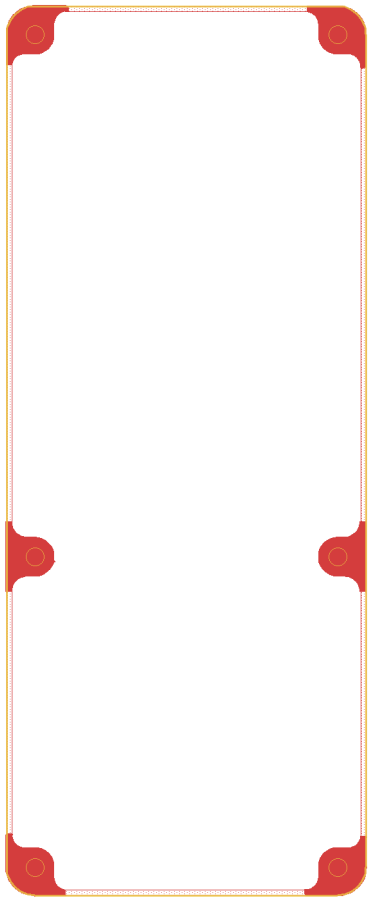


Fig. 9: Daughter Board Top Restrict.

Name.svg for you which you can use to name your daughter board and then import output as monochrome bitmap image. You can import bitmap to the Eagle and place text aligned to rectangular box defined for daughter board names.

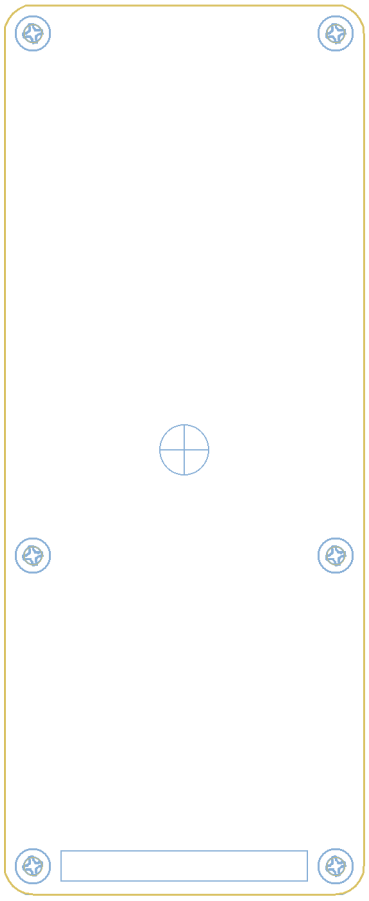


Fig. 10: Daughter Board Top Document.