There is a Python library to access all capabilities of an **official WhatsApp on Raspberry pi** - **[Yowsup](https://github.com/tgalal/yowsup" \t "_blank).** It enables you to use your WhatsApp account to exchange messages without the original app with your contacts. This library allows the user to create a full-fledged custom **WhatsApp client on Raspberry Pi**.

Here we will explain the complete process of **installing and configuring WhatsApp on Raspberry Pi.**

For activating the WhatsApp on pi,

* you will need a new Phone no.
* as it won’t work with currently running WhatsApp no.

So unfortunately for configuring Yowsup we still need a phone and a SIM card, but after the first step you can leave it alone.

So, let’s get started.

### **Requirements**

1. Raspberry Pi 2 or newer (with Raspbian Jessie or newer installed in it).
2. Working Internet Connection

Here, we will use SSH to access Raspberry Pi on the laptop. You can use VNC or Remote Desktop connection on the laptop, or can connect your Raspberry pi with a monitor. Learn more about [setting up Raspberry Pi headlessly here without a monitor.](https://circuitdigest.com/microcontroller-projects/raspberry-pi-headless-setup)

Here, we will use SSH to access Raspberry Pi on the laptop. You can use VNC or Remote Desktop connection on the laptop, or can connect your Raspberry pi with a monitor. Learn more about [setting up Raspberry Pi headlessly here without a monitor.](https://circuitdigest.com/microcontroller-projects/raspberry-pi-headless-setup)

Installing Yowsup (WhatsApp) Library in Raspberry Pi

1. First thing that we have to do is update and upgrade our Pi.

**sudo apt-get update && sudo apt-get upgrade**

2. Install following dependencies as required by *Yowsup* library.

**sudo apt-get install python-dev libncurses5-dev**

**sudo apt-get install build-essential libssl-dev libffi-dev python-dev**

Now, install cryptography packages (required for encrypting the data used in WhatsApp) which is already installed in our Raspberry Pi but the version is not compatible as required by *Yowsup* library.

**pip install cryptography**

3. Download the *Yowsup* library from GitHub using below command

**git clone git://github.com/tgalal/yowsup.git**

4. Go to *yowsup* directory using **cd yowsup**and install the downloaded library using following command

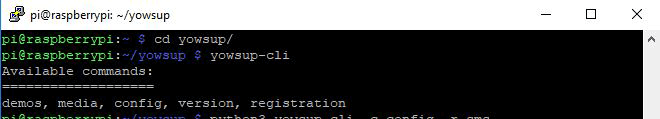
**cd yowsup**

**sudo python setup.py install**

5. Now, we are ready to setup the WhatsApp account. First check whether the library is installed correctly or not. For this, run the given command

**yowsup-cli**

If you are getting the below output then library is installed correctly.



Setup WhatsApp Account on Raspberry Pi

After installing WhatsApp on Raspberry Pi, its time to activate and configure the WhatsApp on pi.

1. Save the country code “cc” and phone number in a file so that Yowsup client can fetch this information for authorization. Open nano editor using **sudo nano config** and save the info using ctrl+x as shown below.

**sudo nano config**

**cc=503**

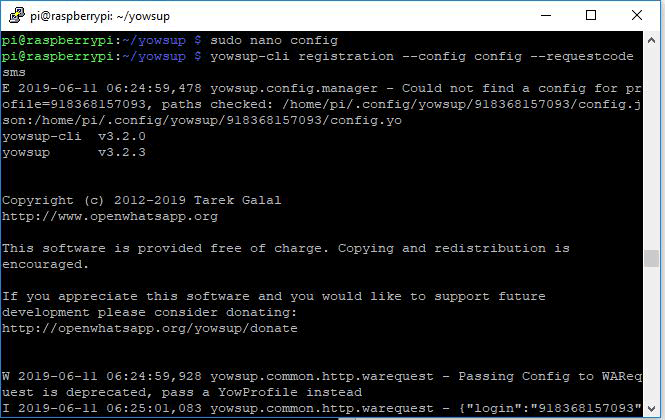
**phone=70020015**

Make sure the phone number entered **is not used by another WhatsApp account** **and the SIM card should be in phone** because you will receive registration code in next step.

2. Now, register your WhatsApp account using below command.

**yowsup-cli registration --config config** **--requestcode sms**

After running the above command you will receive the six digit code through an SMS.

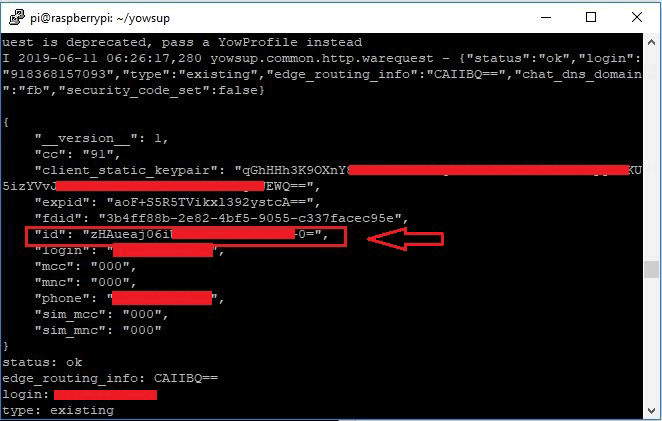


3. Use that code to register the phone number using the command

**yowsup-cli registration --config config --register xxx-xxx**

Replace xxx-xxx with the code you received on your phone.

After a second or two, you'll receive a response from WhatsApp on the Pi terminal as shown below

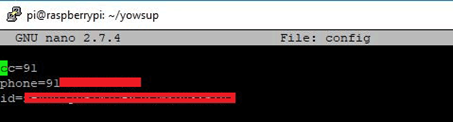


The response contains the password for your WhatsApp. The ID, shown by the arrow in above image, is your password without double quotes. Save this password in the same config file.

So, open the config file using nano editor and paste the id as

**id=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Now, config file will look like this. Save this file.

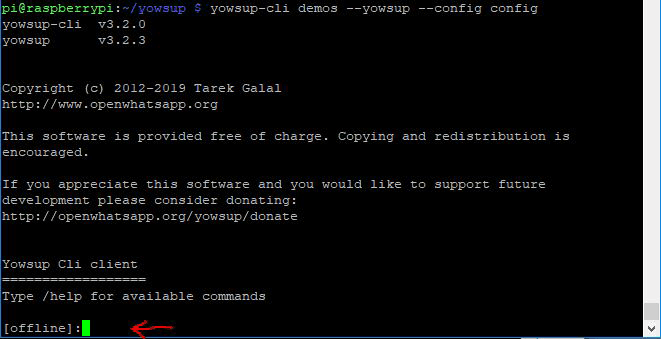


Running the WhatsApp on Raspberry Pi

1. Run the below command to start the WhatsApp.

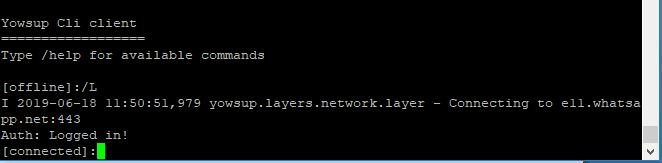
**yowsup-cli demos --yowsup --config config**

You will see following response from WhatsApp. We have to login now. Enter “/L” and hit enter. For exploring more commands you can type /help.



If you are failed to login, restart your Raspberry Pi.

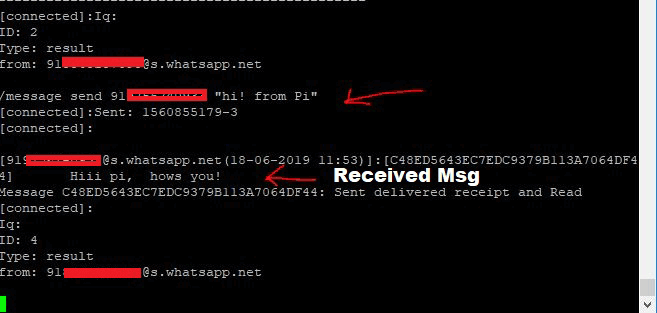
2. After login you are all set to send and receive messages.

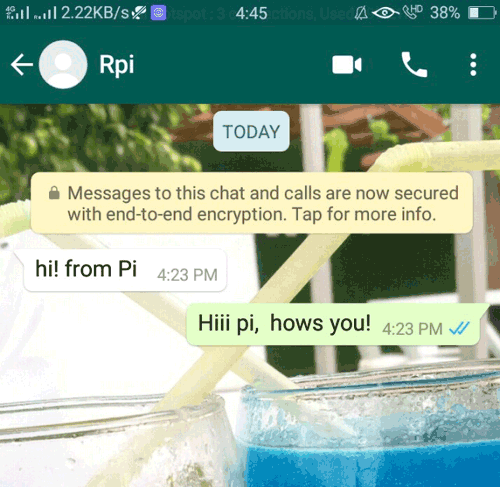


For sending message to other WhatsApp number type the following command.

**/message send 91\*\*\*\*\*\*\*\*\*\* "Your message"**

Replace \*\*\*\*\*\*\*\*\*\* with the receiver WhatsApp number. Sent and received messages will be shown on the same terminal as shown below.





So this is how the **Raspberry Pi can be converted into a WhatsApp client** to send and receive messages from another WhatsApp number.

We can also implement the [Home Automation](https://circuitdigest.com/home-automation-projects) using WhatsApp messages but the *Yowsup* library is not working properly for that purpose and the improvement is going on to use this functionality .

#### COMMANDS

Available commands are registration, demos and version.

’registration' is used to register a mobile number to WhatsApp network.

’demos' is the proper command to regular use as a WhatsApp client.

’version' shows the **yowsup-cli** version.

#### REGISTRATIONOPTIONS

Optional arguments

**-h,** **--help**

Show this help message and exit.

**-v,** **--**version

Print version info and exit.

**-d,** **--debug**

Show debug messages.

**--help-config**

Print a config file sample.

Configuration options

**-c** CONFIG**,** **--config** CONFIG

Path to config file. If this is not set then you must set at least --phone

and **--cc** arguments. The config file is optional. Other configuration

arguments have higher priority if given, and will override those specified

in the config file.

**-m** MCC**,** **--mcc** MCC

MCC (Mobile Country Code). Check your MCC at a specific list (see below).

**-n** MNC**,** **--mnc** MNC

MNC (Mobile Network Code). Check your MNC at a specific list (see below).

**-p** PHONE**,** **--**phone PHONE

The full phone number including the country code defined in 'cc', without

preceding '+' or '00'.

**-C** CC**,** **--cc** CC

CC (Country Code).

Modes to register

**-r** **(sms|voice),** **--requestcode** **(sms|voice)**

Request the digit registration code from WhatsApp.

**-R** code**,** **--register** code

Register account on WhatsApp using a code previously received.

Specific lists for MCC and MNC

· [http://mcclist.com](http://mcclist.com/)

· [http://www.mcc-mnc.com](http://www.mcc-mnc.com/)

· <http://en.wikipedia.org/wiki/Mobile_country_code>

Specific lists for country codes

· https://countrycode.org

· <http://mcclist.com/country-calling-codes.asp>

· <http://www.ipipi.com/help/telephone-country-codes.htm>

#### DEMOSOPTIONS

Optional arguments

**-h,** **--help**

Show this help message and exit.

**-v,** **--**version

Print version info and exit.

**-d,** **--debug**

Show debug messages.

**--help-config**

Print a config file sample.

Configuration options

**-l** phone:b64password**,** **--login** phone:b64password

WhatsApp login credentials, in the format phonenumber:password, where

password is base64 encoded.

**-c** CONFIG**,** **--config** CONFIG

Path to config file containing authentication info. For more info about the

config format, use **--help-config**.

**-M,** **--unmoxie**

Disable support for WhatsApp E2E encryption.

Text interface demo

**-y,** **--yowsup**

Start the Yowsup text interface client (interactive mode). Note that in this

mode two clients or a group can talk in chat mode, send images, etc.

Echo client demo

**-e,** **--echo**

Start the Yowsup Echo client.

Send client demo

**-s** phone message**,** **--send** phone message

Send a message to specified phone number, wait for server receipt and exit.

Sync contacts

**-S** contacts**,** **--sync** contacts

Sync (check valid) WhatsApp contacts.

#### CONFIGURATIONFILE

A configuration file, known as login credentials, can be used to gather some basic

information that useful to register or login. This configuration file will be called by

'**-c**' option.

WhatsApp protocol is basically a modified version of XMPP. It uses JID and password for

login. The JID is the phone number registered with, which is a combination of the country

code and the phone number. To login will be needed a password. These info can be saved in

a text file, in the following format:

cc=55

phone=556101234567

password=NDkxNTIyNTI1NjAyMkBzLndoYXRzYXBwLm5ldA==

The line 'cc=' must be filled with the country code.

The 'phone=' field will be composed by 'country code' + 'area code' + 'phone number'. In the current example was used 55 (country = Brazil) + 61 (area code = Brasilia) + 01234567

(phone number). The country code must be provided without '+' or '00' leading.

Finally, the 'password=' field has the login password. This password will be gotten when you register the phone number using **yowsup-cli**. See the REGISTERING A NUMBER section. If you are registering a number, you can leave this field blank.

For security reasons, is recommended set a permission '600' to configuration file and make it hidden. More details about the configuration file can be viewed using this command:

$ yowsup-cli registration --help-config

#### REGISTERINGANUMBER

WhatsApp registration involves 2 steps.

First you need to request a registration code.

After this, you resume the registration with code that you got.

In both cases, you can or not use a configuration file (see the above section in this manual page).

An example not using a configuration file:

yowsup-cli registration --requestcode sms --phone 556101234567 --cc 55 --mcc 123 --mnc 456

yowsup-cli registration --register 123-456 --phone 556101234567 --cc 55

Now, using a configuration file:

yowsup-cli registration --requestcode sms --mcc 123 --mnc 456 -c /home/user/.yowsuprc

yowsup-cli registration --register 123-456 -c /home/user/.yowsuprc

The first command (the '**--requestcode**' action) will show something similar to:

status: sent

retry\_after: 10805

length: 6

method: sms

The code will then be delivered to your mobile phone number via your preferred

method (SMS/voice). Once you have obtained the code, you proceed with the second step, which is to send that code using the '**--register**' argument. This example will consider the received code as being 123-456:

yowsup-cli registration --register 123-456 -c /home/user/.yowsuprc

After the last command, you will see something similar to:

status: ok

kind: free

pw: 1coGO6bwjT8Uukw9qa8v3ql2KXM=

price: US$0.99

price\_expiration: 1407002374

currency: USD

cost: 0.99

expiration: 1435773701

login: 556101234567

type: new

If you successfully register, WhatsApp will generate a password for your which will be displayed in the command output (see the 'pw:' line). If you want to use a config file, you MUST MANUALLY store this password in this config file. See an example:

cc=55

phone=556101234567

password=1coGO6bwjT8Uukw9qa8v3ql2KXM=

#### SENDINGMESSAGES

**Send** **a** **simple** message **to** **+0055** **11** **190102030** **(using** **a** **config** **file):**

$ **yowsup-cli** demos **-c** /path/.config\_file **-s** 5511190102030 "Testing WhatsApp..."

**Send** **a** **simple** message **to** **+0055** **11** **190102030** **(without** **a** **config** **file),** **from** **556101234567:**

$ **yowsup-cli** demos **-l** 556101234567:1coGO6bwjT8Uukw9qa8v3ql2KXM= **-s** 5511190102030

"Testing WhatsApp..."

**Start** **the** **interactive** **mode** **(using** **a** **config** **file)**

$ **yowsup-cli** demos **-c** /path/.config\_file **-y**

Start a chat with 5511190102030, after connected in interactive mode (see the last

command):

/L

/message send 5511190102030 "Testing WhatsApp..."

Send a picture to 5511190102030, after connected in interactive mode:

/L

/image send 5511190102030 /home/user/picture.jpg

To see all commands available in interactive mode:

/help

#### AUTHOR

**yowsup-cli** was developed by Tarek Galal <[tare2.galal@gmail.com](mailto:tare2.galal@gmail.com)>.

This manual page was written by Joao Eriberto Mota Filho <[eriberto@debian.org](mailto:eriberto@debian.org)> for the

Debian project (but may be used by others).