

This guide assumes you have a Raspberry Pi Zero W and want to set it up to run headless on your wireless network with SSH enabled (for management) and Cloud9 installed to allow for remote, instant Python coding for your projects.

## Getting the Pi Zero operating on your network

1. Download the latest image of Raspbian Stretch Lite:  
<https://www.raspberrypi.org/downloads/raspbian/>
2. Using Etcher (or a similar image writing application), burn the image to your SD card
3. Mount the resulting burned SD image; you should see a boot volume with a number of files inside.
4. Create a new file called `wpa_supplicant.conf` in the root of the boot volume, with the following contents:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=US

network={
    ssid="YOURSSID"
    psk="YOURPASSWORD"
    scan_ssid=1
    key_mgmt=WPA-PSK
}
```

5. Create an empty file called `"ssh"`. This will be picked up by the boot process, and enable ssh support on the image.
6. Edit the `config.txt` file, and add the following lines at the end:

```
# Enable UART
enable_uart=1
```

7. Unmount the image and put the SD in the raspberry pi. Power it up and wait a couple of minutes.
8. If all goes well, the device will register it on your local network using mDNS and appear as `"raspberrypi.local"`. You can ping and ssh into the device now.

If the device doesn't show up, then you need to go into troubleshooting mode. For example, if you're working on a small local network where you don't have a network admin to yell at you for running scans, you can power off the device and run an nmap scan (`"sudo nmap -sn 192.168.0.0/24"`) to identify devices on your local subnet. Boot the Pi Zero and scan again, and chances are that any new device is your Pi.

## Updating the OS and installing Cloud9

1. Log into the Pi and run:

```
sudo apt-get update && sudo apt-get upgrade
```

2. Follow the steps in Chintan Pathak's article here:  
<https://medium.com/@chintanp/using-cloud9-3-0-ide-on-raspberry-pi-954cf2d6ab8e>
3. Follow the Adafruit guide for setting up your Raspberry Pi for i2c here:  
<https://learn.adafruit.com/adafruit-16-channel-servo-driver-with-raspberry-pi/configuring-our-pi-for-i2c>
4. If you're using the Adafruit PWM/Servo breakout board with your Pi (like we are), follow the Adafruit guide for installing the Adafruit library for the breakout board here:  
<https://learn.adafruit.com/adafruit-16-channel-servo-driver-with-raspberry-pi/using-the-adafruit-library>

Once all of those installations are complete, you should be able to use Cloud9 via web browser, and use the Adafruit library and example code to control servos and ESCs from the Pi.

## Optional stuff - extra python libraries, etc

You're likely going to want Python:

```
sudo apt-get update  
sudo apt-get install -y python3 python3-pip python-dev  
sudo pip3 install rpi.gpio
```

Depending on what you're doing, you might also be interested in the following libraries via git or pip:

Adafruit GPIO:

```
sudo pip3 install adafruit-gpio
```

Python driver for Adafruit FXOS8700: [https://github.com/MomsFriendlyRobotCompany/nxp\\_imu](https://github.com/MomsFriendlyRobotCompany/nxp_imu)

```
cd
git clone https://github.com/MomsFriendlyRobotCompany/nxp_imu
sudo pip3 install build_utils
cd nxp_imu
sudo python3 setup.py install
```

SMBus:

```
sudo pip3 install smbus2
```

Adafruit (or compatible) PWM servo driver:

```
Sudo pip3 install adafruit-pca9685
```

*You may also need go into the Adafruit\_Python\_PCA9685 directory and run "sudo python3 setup.py install".*

Python library for the vl53l1x laser distance sensor:

```
sudo pip install vl53l1x
```

Install numpy:

```
sudo pip3 install numpy
```

Jupyter notebook:

```
python3 -m pip install --upgrade pip
python3 -m pip install jupyter
```

From your home directory, run:

```
jupyter notebook --generate-config
```

Edit the file with the following values:

```
c.NotebookApp.ip = '*'  
c.NotebookApp.password = ''  
c.NotebookApp.open_browser = False
```

*Warning: this sets up no encryption or authentication, which is not recommended.*

Consider adding a line to your /etc/rc.local script (or crontab) to start on bootup.

**Reference:**

<https://learn.adafruit.com/adafruit-arduino-lesson-14-servo-motors/servo-motors>