

# GoPiGo Getting Started

## Contents

GoPiGo Getting Started .....	1
GoPiGo with Raspberry Pi OS on MicroSD Card.....	1
Boot the GoPiGo Setup Raspberry Pi OS .....	3
Connect to the GoPiGo with UltraVNC .....	4
Setup Raspberry Pi OS .....	4
Setup GoPiG3 Software .....	5
Email IP Address on Startup .....	5
Run startup_mailer.py Script on Startup.....	7
Power the GoPiGo .....	8
Update Raspbian .....	8
Update GoPiGo Software .....	8
Multiple Wireless SSID's.....	8
Wi-Fi Signal Strength.....	9
Signal Strength .....	9
Link Quality .....	10
Frequency .....	10
Disable Onboard Wi-Fi .....	10
Set Interfaces.....	10
Set 12-Hour Clock.....	11
Backup GoPiGo to a File Image .....	11

**NOTE:** All Python code is compatible with Python 3.5. This is the current version of Python on GoPiGo Raspbian for Robots. If you build a Raspberry Pi OS (Currently Buster) and install the GoPiGo3 software, you will be using Python 3.7.

## GoPiGo with Raspberry Pi OS on MicroSD Card

We want to download and install the latest Raspberry Pi OS.

**NOTE:** The default username and password for a Raspberry Pi:

Username: **pi** Password: **raspberrypi**

**Raspbian is Linux:** All commands in **Raspbian** are **case sensitive**

**NOTE:** Remove all USB drives from computer except the MicroSD adapter.

1. Insert the Transcend adapter with minimum 8 GB MicroSD card into a USB port on your computer.
2. Download and install **Raspberry Pi Imager**  
<https://www.raspberrypi.org/blog/raspberry-pi-imager-imaging-utility>
  - a. **Operating System** → **Choose OS** → **Raspberry Pi OS (32-bit)**
  - b. **Storage:** Choose **TS-RDFS SD Transcend**
  - c. **Write:** Write the image
  - d. When prompted: Remove the Transcend adapter. Put it back in your computer USB Drive.
3. On your computer, use Notepad to create a file named **wpa\_supplicant.conf** (Make sure it does not have a .txt extension.)

The following is an example of adding your home network and WNCC-Internet to your wireless network.

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

network={
    ssid="network_one_here"
    psk="wpa_password"
    id_str="home"
}

network={
    ssid="WNCC-Internet"
    key_mgmt=NONE
    id_str="work"
}
```

4. Copy this file to the boot drive of the MicroSD card.
5. Create a blank file named **ssh** (No file extension) in the boot drive of the MicroSD card.

## Boot the GoPiGo Setup Raspberry Pi OS

1. Make sure the GoPiGo is powered off before inserting the MicroSD card.
2. Insert the MicroSD card in the Raspberry Pi.
3. Power up the GoPiGo. This will take a little longer the first time you boot the robot.

NOTE: There are two ways to configure the GoPiGo on the network.

- Option 1: Connect a keyboard, mouse and monitor to the GoPiGo. Connect to or point to the wireless connection. You want to find the IP address.
4. Easiest option: Use ZenMap with Quick Scan settings to scan your network for the Raspberry Pi and note the IP address.
  5. Go to <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
  6. Download the **putty** client.
  7. Start the putty client. Type in the IP address of the GiPiGo. Click **Open**.
  8. Accept the **Putty Security Alert**.
  9. Login as: **pi**
  10. Password: **raspberrypi**
  11. Type: **sudo raspi-config**
  12. Select **Interface Options → VNC → Select Yes**.
  13. Exit the **raspi-config** interface.

**Note:** To be able to use a non-compatible VNC client like UltraVNC, you have to downgrade the security of the VNC server to use VNC password authentication.

14. At the ssh terminal prompt:

```
sudo nano /root/.vnc/config.d/vncserver-x11
```

15. Add the following lines at the end of the file

```
Authentication=VncAuth  
Encryption=AlwaysOff
```

16. Use the following command to restart the VNC service

```
sudo systemctl start vncserver-x11-serviced.service
```

17. To set the VNC password, use the following command. Password: robots1234

```
vncpasswd -service
```

18. Type **reboot** to restart the pi.

## Connect to the GoPiGo with UltraVNC

With UltraVNC you can copy and paste from your desktop to your robot.

1. Go to [www.uvnc.com](http://www.uvnc.com) Be careful, there are ads all over the place.
2. At the top of the page go to **Downloads → UltraVNC**
3. Toward the bottom you will find **UltraVNC 1.3.4**. (This is the current version as of 9-11-21)
4. Click on the name to download it. This will take you to a page with Installers.
5. You want the one for-64 bit operating systems. Click Download to the right.
6. The download will start in a few seconds.
7. The downloaded file will called **UltraVNC\_1\_3\_4\_X64\_Setup** The version number may be different.
8. Double Click the file to start the installation.
9. When you get to select components → only choose **UltraVNC Viewer**.
10. Continue to install the program.
11. Run the program.
12. Type in the IP address of your robot. Click Connect
13. You will be asked for a password: **robots1234**

## Setup Raspberry Pi OS

1. Click **Next** on the **Welcome to Raspberry Pi** welcome screen.
2. Set Country: **United States** Timezone: **Denver**. Check **Use English Language** and **Use US Keyboard**. Click **Next**.
3. **Change Password: robots1234** Click **Next**.
4. **Set Up Screen**. Click **Next**.

5. **Setup Wifi Network.** Click **Skip**.
6. **Update Software:** Click **Next**. This will take a while.
7. To change the display resolution, open a terminal: **sudo raspi-config**
8. **Display Options → Resolution →** Select the resolution of the GoPiGo screen that works for you.
9. Press the **Tab** key to select **Ok**. Tab to Finish. **Yes** to reboot.

## Setup GoPiG3 Software

1. At the terminal, type the following commands. This will take some time.

```
curl -kL dexterindustries.com/update_gopigo3 | bash  
curl -kL dexterindustries.com/update_sensors | bash
```

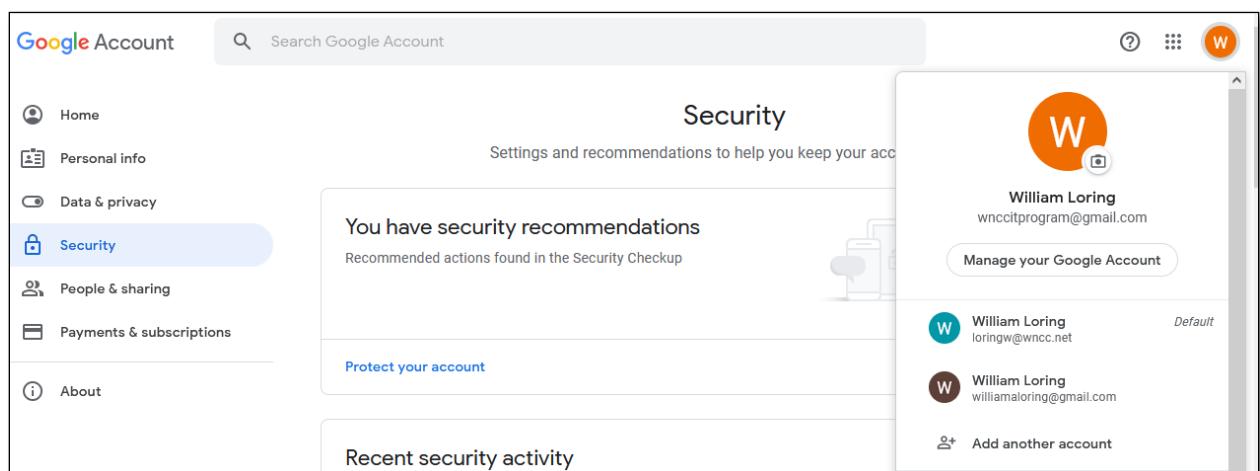
2. Reboot the Raspberry Pi to make the settings take effect: **sudo reboot**

## Email IP Address on Startup

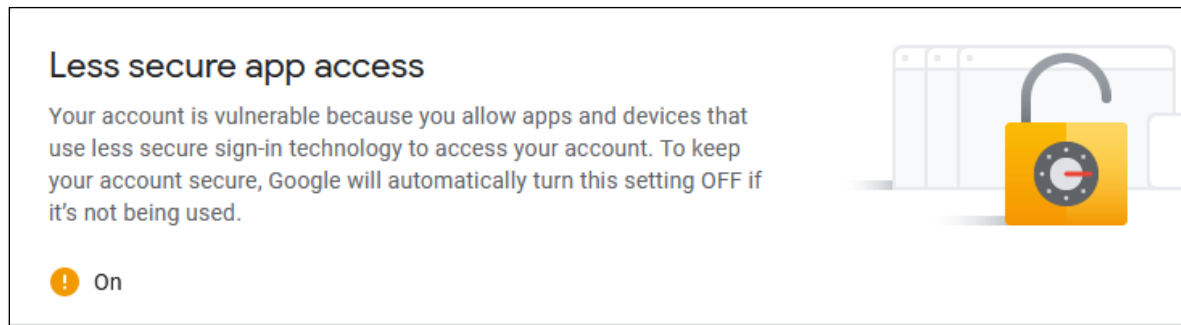
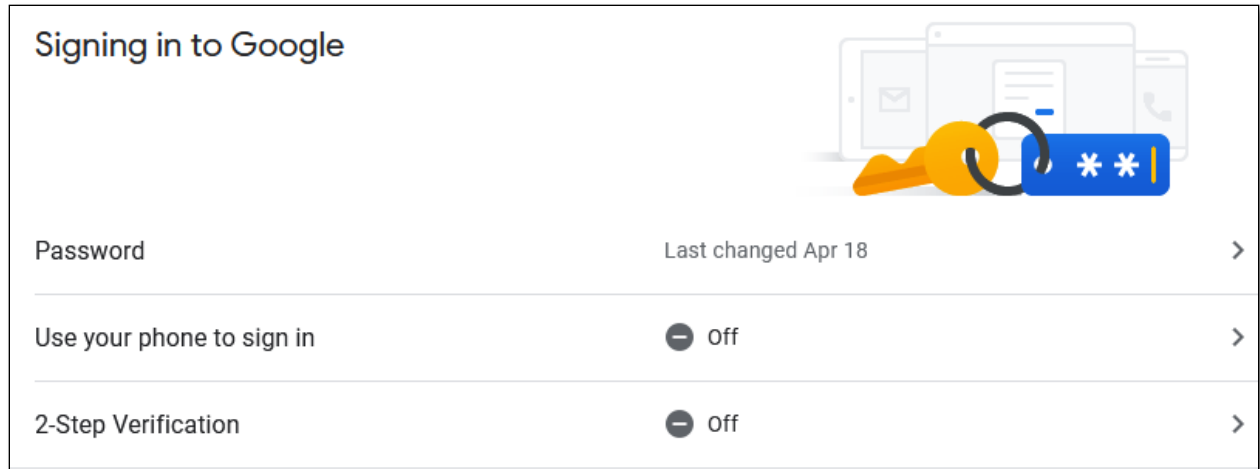
We want our GoPiGo to email us the IP address whenever it starts up.

You will need to use or create your own Gmail account.

1. Logon to your Gmail account on a web browser.
2. Go to **Manage your Google Account → Security**.



3. Scroll down and turn these settings off as shown.



Complete the following steps on the GoPiGo desktop.

1. Use the Web Browser on the upper right side to go to <https://github.com/it instructor/WNCCNASA>
2. Logon with your GitHub account.
3. Go to **Code** → **Download ZIP**.
4. The file will download quickly. On the lower left side of the browser → Right Click on the file → **Open in Folder**.
5. Right Click the Zip file → **Extract Here**.
6. Right Click **startup\_mailer.py** → **Copy**.
7. Create a folder named **Code** → Paste the file into that folder.
8. Right Click on **startup\_mailer.py** → **Geany**
9. Change the **EMAIL\_DESTINATION** email address to your own email address.

10. Change **EMAIL\_SOURCE** and **EMAIL\_PASSWORD** to your gmail account information.
11. Save the file.
12. Open a terminal.
13. Type in the following to make the script executable.

```
sudo chmod 777 /home/pi/Code/startup_mailer.py
```

14. There should not be any errors if the command was successful.
15. Test the script with the following command.

```
python3 /home/pi/Code/startup_mailer.py
```

16. In a few moments, you should receive an email with your GoPiGo IP address.

---

## Run startup\_mailer.py Script on Startup

1. At the terminal, type in the following command to access the Raspbian scheduler.

```
crontab -e
```

2. Press Enter to edit the file with nano
3. Cursor to the bottom of the file. (The mouse will not work.)
4. Type in the following information. (Sleep 10 waits 10 seconds after startup to run the script.)

```
@reboot sleep 15 && python3 /home/pi/Code/startup_mailer.py
```

5. Type **CTRL+O** to Write Out the file.
6. Press **Enter** to Write the file.
7. Press **CTRL+X** to Exit nano.
8. Type **poweroff**
9. Wait until the GoPiGo has a chance to shut down.
10. Turn on your robot and you should receive an email with your IP address.

## Power the GoPiGo

\*Notice the power switch on the battery. The battery will need to be *on* before starting the robot. **However**, the robot should be turned off before turning off the battery.

## Update Raspbian

Once in a while, you will want to update the Raspberry Pi OS. This may take some time.

```
sudo apt-get update
sudo apt-get dist-upgrade
```

## Update GoPiGo Software

Once in a while, you will want to update the GoPiGo software.

```
sudo bash /home/pi/Dexter/GoPiGo3/Install/update_gopigo3.sh
sudo bash /home/pi/Dexter/DI_Sensors/Install/update_sensors.sh
```

## Multiple Wireless SSID's

If you are using the GoPiGo on multiple networks, edit the following file as shown. This command uses nano, a simple text editor built into the operating system.

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

The following is an example of adding the WNCC-Internet to your wireless networks.

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

network={
    ssid="network_one_here"
    psk="wpa_password"
    id_str="home"
}

network={
    ssid="WNCC-Internet"
    key_mgmt=NONE
    id_str="work"
}
```

1. **CTRL+O** (Writes the file)



2. **Enter** to finish saving the file.
3. **CTRL+X** (Exit nano)

The pi will automatically connect to whichever wireless network is closer and has better signal. You can add as many wireless networks to this file as you wish.

## Wi-Fi Signal Strength

**iwconfig** will give you a snapshot of Wi-Fi quality.

**wavemon** will monitor Wi-Fi signal strength in real time.

```
# Install wavemon
sudo apt install wavemon -y
# Run wavemon
wavemon
# Quit wavemon
q
```

### Signal Strength

The higher the signal strength, the more reliable the connection and higher speeds are possible. The signal strength is specified as -dBm (decibels related to one milliwatt).

Values between 0 and -100 are possible, with more being better. -51 dBm is a better signal strength than -60 dBm.

The value 0 is not realistic. Even -30 dBm is hard to reach, and you have to stand almost directly next to the access point.

Some guidance on how to read the results:

- 50 dBm is considered an excellent signal strength.
- 67 dBm is said to be the minimum signal strength for reliable and relatively fast packet delivery.
- 70 dBm is the minimum signal strength for reliable packet delivery.
- The minimum value for a basic connection is -80 dBm. However, packet delivery is no longer necessarily reliable.
- 90 dBm is already very close to the basic noise. Here a connection probably does not work anymore.

## Link Quality

A network can have very good signal strength without good link quality.

This is how much of the data you send and receive will make it to the destination in good condition.

The quality indicator includes data like Bit Error Rate (BER), i.e., the number of bit errors in received bits that have been altered due to noise, interference, distortion, or bit synchronization errors. Others are Signal-to-Noise and Distortion Ratio (SINAD).

It is measured in percentage or on a scale of up to 70. So you will see a value like "60/70".

Unlike signal strength, it is somewhat harder to say which values are still considered to be ok.

If the value is low and your signal strength is high, you may have interference from, e.g., kitchen appliances or other electronic devices. Moving them further away may improve the link quality.

## Frequency

Another interesting indicator is the Wi-Fi frequency.

This shows if your Raspberry Pi connects to the slower and longer range 2.4 GHz network, or the faster but shorter range 5 GHz version, provided, of course, that your router offers both networks.

## Disable Onboard Wi-Fi

Any external Wi-Fi antenna will have better signal strength and range. To use an external Wi-Fi antenna, you must disable the internal Wi-Fi.

```
# Edit this file with nano
sudo nano /boot/config.txt
# Add this line to the file and save it
dtoverlay=disable-wifi
```

Once you have disabled the on board Wi-Fi, you must always plug a Wi-Fi adapter into the USB port.

## Set Interfaces

1. Go to the Raspberry icon on the left side of the toolbar.
2. **Preferences → Configuration → Raspberry Pi Configuration → Localisation.**

3. Click **Interfaces**
  - a. Enable Camera, SSH, SPI, I2C
4. Click OK.

## Set 12-Hour Clock

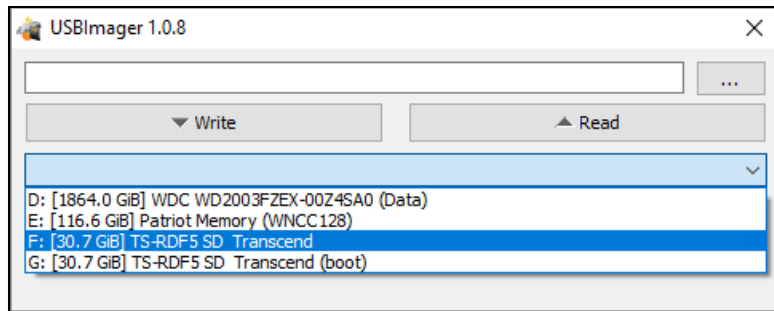
To change the clock from military time to 12 hour time:

1. Right Click on the clock on the right hand side of the toolbar → **Digital Clock Settings.**
2. Change Clock Format to: **%I:%M %p %x**
  - a. **%I:%M** = Hours Minutes
  - b. **%p** = AM PM
  - c. **%x** = current date
3. Click OK.

## Backup GoPiGo to a File Image

Things go wrong. It is a good idea to back up your MicroSD card to a file image at this point and any point prior to making big changes.

1. Go to <https://gitlab.com/bztsrc/usbimager>
2. Download the windows GDI version.
3. This is a portable program, there is not an installation, the program runs from wherever you put it.
4. Click the downward pointing triangle as shown to select your MicroSD card. The card will show 2 partitions, it doesn't matter which one you choose, the entire card will be backup up to a file image.



5. Click **Compress**.
6. Click **Read**. The program will automatically create a compressed backup file of approximately 3GB on your Desktop.

