

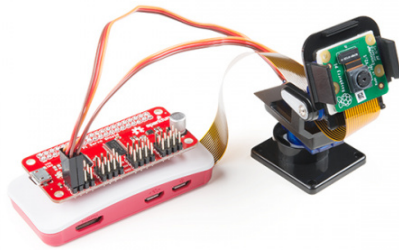


---

# Setting Up the Pi Zero Wireless Pan-Tilt Camera

## Introduction

This tutorial will show you how to assemble, program, and access the Raspberry Pi Zero Wireless Pan-Tilt Camera.



## SparkFun Raspberry Pi Zero W Camera Kit

© KIT-14329

## Required Materials

You'll need a microSD card, a sufficient power supply, and a micro-B USB Cable. We do not recommend the pre-installed NOOBS card, as it may not include a recent enough version of the Pi operating system to support the Pi Zero W.



microSD Card with Adapter -  
16GB (Class 10)

© COM-13833

USB micro-B Cable - 6 Foot

© CAB-10215



**Wall Adapter Power Supply -  
5.1V DC 2.5A (USB Micro-B)**

© TOL-13831

## Required Tools

You'll need a few tools to assemble this kit: just a standard soldering iron, solder, a small screwdriver, and a pair of side cutters.



**Soldering Iron - 30W (US,  
110V)**

© TOL-09507



**Flush Cutters - Hakko**

© TOL-11952



**Solder Lead Free - 15-gram  
Tube**

© TOL-09163

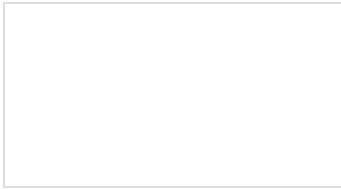


**Tool Kit - Screwdriver and  
Bit Set**

© TOL-10865

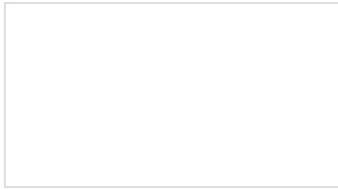
## Suggested Reading

Very little of this kit requires anything but following our step-by-step guide; however, you might want to check out our through hole soldering tutorial as you'll need to do some soldering to put the pins in place on the Pi Zero W and the Pi Servo Hat. Other, tutorials you might be interested in reading include:



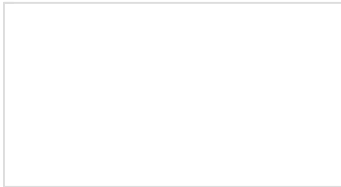
### How to Solder: Through-Hole Soldering

This tutorial covers everything you need to know about through-hole soldering.



### Hobby Servo Tutorial

Servos are motors that allow you to accurately control the rotation of the output shaft, opening up all kinds of possibilities for robotics and other projects.



### Getting Started with the Raspberry Pi Zero Wireless

Learn how to setup, configure and use the smallest Raspberry Pi yet, the Raspberry Pi Zero - Wireless.

## Prepare the MicroSD Card

To prepare the microSD card, we need to:

- Download the latest Raspberry Pi Raspbian Jessie image.
- Create an microSD card that contains this boot image.
- Edit the "config.txt" file on the microSD card to enable the serial debug console.
- Create a "wpa\_supplicant.conf" file on the SD card to enable WiFi.

Let's walk through these steps in more detail!

### Download the Latest Raspbian Jessie Image

You can find the image on the Raspberry Pi Foundation Website. It's a fairly large download so go get a snack or play a game or something.

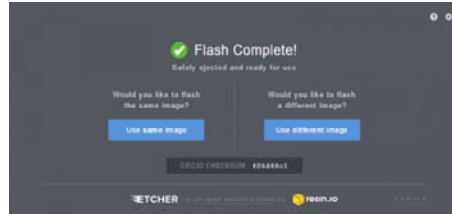
**Note:** Make sure that you don't download "RASPBIAN XYZ LITE", as the Lite installation is lacking several utilities and programs we'll need to get things moving. Note that XYZ is a placeholder for the release name of the distribution; at the time of this writing, that name is "Stretch".

### Burn the Image to an MicroSD Card

To write the image to the card, there exists an excellent utility for Mac/Linux/Windows called Etcher. Simply download and install it to your computer. Then select the image you downloaded earlier, the drive you want to install to (Etcher won't let you install to anything but a removable drive, and if only one drive is available, it will select it automatically!), and click "Flash!"



This is another long-ish process, requiring several minutes to complete. Once complete, the window will display a “Flash Complete!” message.



By default, Etcher “ejects” the card after it has created the image, so you’ll need to pull it out and reinsert it to get your computer to reload the disk.

## Edit the “config.txt” File on the MicroSD Card

To continue, we’re going to need to edit a file on the microSD card. This file will be visible from any operating system, although on a Linux system it’s in a slightly different location.

On a Windows or Macintosh, the files will be loaded in the root directory of the microSD card, so when you open a window of the drive, you’ll see them directly. In Linux, the card will be mounted, and you’ll have to navigate to the “Boot” directory on the card to find the file we’re looking for.

Here’s a list of all the files in the directory of interest.



*Click the image for a closer look.*

Open the file “config.txt” in your favorite text editor (i.e. Notepad++, TextWrangler, ConTEXT Editor, etc.) and add this line to the very end:

```
enable_uart=1
```

**Note:** Despite the presence of a hash (#) in front of the other lines in this file, you **must not** put one in front of this line. That would comment it out, causing it to be ignored. Adding this line will enable the UART on the GPIO pins so you can complete the rest of the process without having to hook your Pi Zero W up to a monitor and keyboard. Pretty nifty!

## Create a “wpa\_supplicant.conf” File on the MicroSD Card to Enable WiFi

The last thing we need to do to prepare the microSD card is to create a “wpa\_supplicant.conf” file on the card. This file contains the information needed for your local wireless network setup.

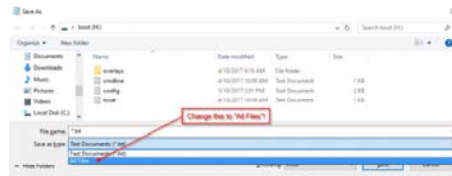
Create a new file using your favorite text editor. On Windows we recommend Notepad, as it provides WYSIWYG content and allows you to save a file with an arbitrary file extension. On MacOS, TextWrangler seems to be the easiest. For Linux, your default system text editor should be fine.

The contents of the file can be quite simple. Most likely, you can get away with something that looks like this:

```
network={
  ssid="YOUR_SSID"
  psk="YOUR_PASSWORD"
  key_mgmt=WPA-PSK
}
```

Once added, modify the network ID and password for your WiFi network. Then save the file as "wpa\_supplicant.conf" to the microSD card.

**Important Note:** On the on the "Save As..." screen for Windows users, you'll need to change the file extension type under the field "Save as type:" from "\*.txt file" to "All files". Then you will need to explicitly name the file "**wpa\_supplicant.conf**", or Notepad will automatically append ".txt" to the file name, breaking this functionality. See the image below.



*Click the image for a closer look.*

Some variation of this "Save as..." and removing the appended ".txt" suffix is necessary in Mac OS and Linux as well, but should be equally simple.

## Pull the MicroSD Card and Insert It into the Pi Zero W

This is all we need to do in preparation of our first boot. You can now remove the card from your computer and put it into the Pi Zero W. Don't power it up yet, though!

## Hardware Assembly

To assemble the hardware, we need to:

- Solder headers to the Raspberry Pi Zero W.
- Solder headers to the Pi Servo Hat.
- Install the Pi Servo Hat on the Pi Zero W.
- Assemble the pan-tilt hardware and connect the servos to the hat.

Let's walk through these steps in more detail!

### Solder Headers to the Raspberry Pi Zero W and the Pi Servo Hat

We recommend soldering the male header to the Pi Zero W and the female to the Pi Servo Hat. If you have any issues with soldering, please check out our learn to solder tutorial.

### Assemble the Pan-Tilt Mechanism

Assembly of the pan-tilt mechanism is fairly straightforward. The trickiest part is making sure the servo motors are centered during assembly.

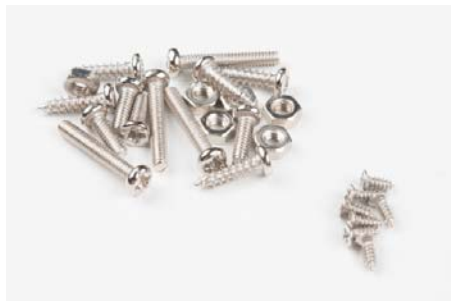
Here's the family portrait of the stuff that comes in the kit. You won't be needing the servo horns that come packaged with the servo motors, just the ones that come packaged separately.



Start by identifying the servo horn with two long arms and two short arms. You'll need to clip off the long arms, as shown below.



Identify the smallest screws in the baggie of screws that came with the kit. These will be used to affix this horn to the base of the pan-tilt mechanism.



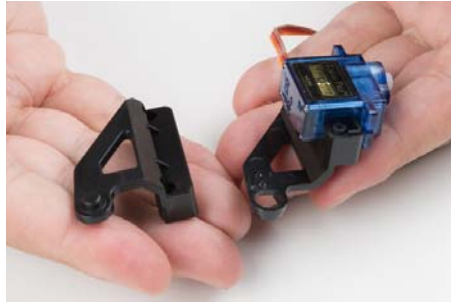
Place the horn in the base as shown, then screw it down by inserting the screws from the bottom and threading them into the horn. Note that there will be extra screws, even beyond the ones to be used later. This is generally true of all the screws in this set.



Next, identify the larger self-tapping screws. These will be used for assembling the next part of the mechanism.



Find the two pieces pictured below that will sandwich the first servo. Note the orientation of the servo in these pieces.



Here's the sandwiched servo. Again, note the orientation for proper assembly.



Here, you can see where the screws identified a couple of steps ago fit into our servo sandwich. Tighten them down, but not too much.



Now fit the shaft of the servo into the fitting on the horn that you previously installed into the base. This is the point where you need to make sure that the shaft is roughly centered in order for the entire assembly to work properly. I do this by turning the shaft all the way to one extreme, then turning it 90 degrees back in the other direction. Then, I remove the base and line it up with the body of the servo motor.



Find the two longer screws that came with the horn kit in the set. There will likely be only two of these, and you need both, so don't lose one!



You'll now need the single arm servo horn, as shown in the image below on the left.



Install the single arm servo horn as shown below. You'll need two of the small self-tapping screws from the first step to affix it to the mechanism.

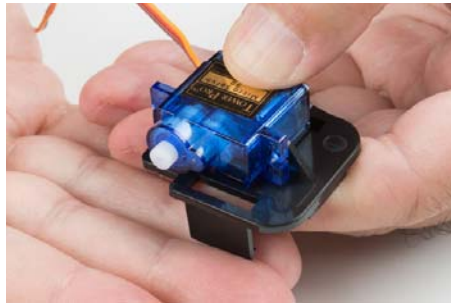


You'll now need the second servo motor and the last piece of the mechanism. The image below shows the relative orientation of these two pieces.





Here's a picture of the two pieces assembled to one another.



Now, find the longest machine screws in the baggie, as shown below. Once again, you'll find that there are more of these than you need.



Thread these screws through the stand off wings on the servo motor and into the last piece of the mechanism. You can use nuts for these if you like, but I've found it to be unnecessary as they thread into the mechanism quite snugly without the nut.



Now connect the two major pieces of the assembly together. The image below shows the orientation of these two parts.



You may need to assemble and disassemble these two parts a couple of times to find the right rotational position of the servo motor so that the tilting portion has its full range of motion. Here's an image of the two bits put together.



Take the final screw that you identified above as a horn attaching screw and use it to secure the horn to the servo motor.

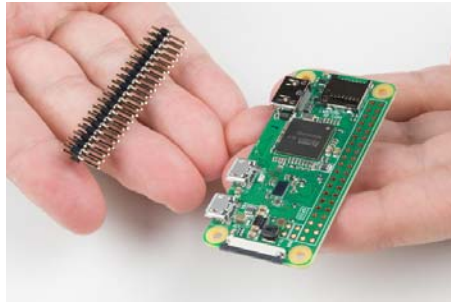


Congratulations, you've finished assembly of the pan-tilt mechanism!



## Solder the Headers onto the Pi Zero W and Pi Servo Hat

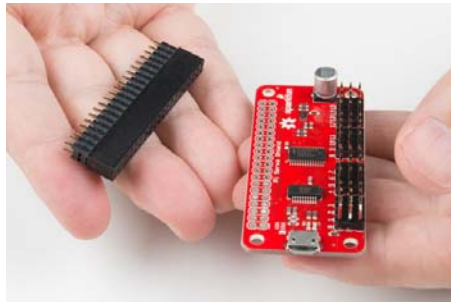
We suggest soldering the male headers onto the Pi Zero W.



My favorite trick for this type of situation is to solder down one pin, then melt the solder on that pin with the iron held in my right hand and use my left hand to adjust the header until it sits flat as shown below. Make sure that you are soldering with the header's shorter side and the longer pins are on the component side. After tacking down one pin, finish soldering all the pins down to the Pi Zero W.



Repeat the steps with the female header and the Pi Servo Hat.

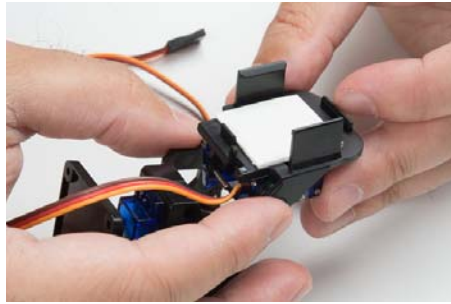


Make sure to insert the short pins from the bottom of the board and add solder to the component side so that the Pi Servo Hat stacks on top of the Pi Zero W's male header pins. You will also need to make sure that the header is sitting level before soldering down all the pins.



### **Affix the Camera Module to the Pan-Tilt Mechanism**

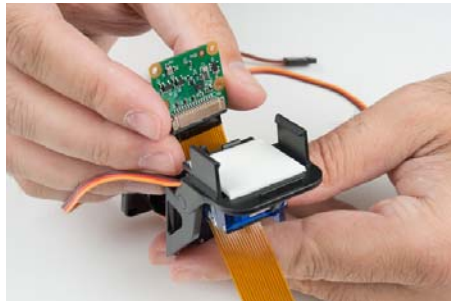
Start by affixing the double stick foam tape to the pan-tilt mechanism as shown in the image below.



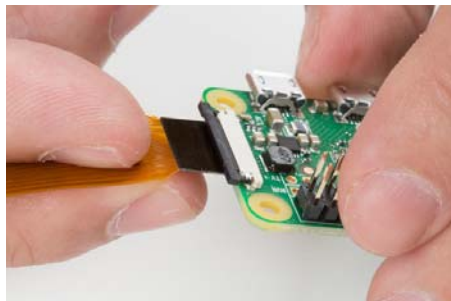
Next, attach the flex cable to the camera module. Take note of the copper "fingers" on the end of the cable. The side with the fingers on it should be **facing** the camera module circuit board. See the image below.



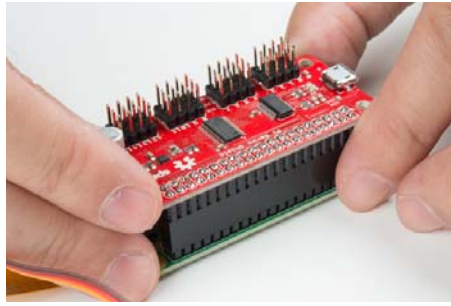
Thread the flex cable through the pan-tilt mechanism as shown below, then press the back of the camera board against the double stick tape to adhere it to the pan-tilt mechanism.



Insert the flex cable from the camera into the Pi Zero W. As was the case with the other end, the side of the cable with the copper fingers should be facing the circuit board. Don't forget to close the clamping mechanism!



You may now attach the Pi Servo Hat to the header on the Pi Zero W.

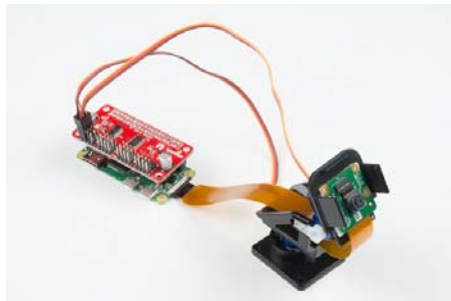


## Connect the Servo Motors to the Pi Servo Hat

The servo motors need to be connected to channels 0 and 1 on the Pi Servo Hat. Connect the “pan” servo (the one on the bottom) to channel 0 and the “tilt” servo to channel 1.

Pi Servo Hat (CH 0)	Pan Servo (Bottom of Pan-Tilt Mechanism)	Pi Servo Hat (CH 1)	Tilt Servo
SIG	Control Signal (Orange)	SIG	Control Signal (Orange)
POW	Vcc (Red)	POW	Vcc (Red)
GND	GND (Brown)	GND	GND (Brown)

Once connected, your setup should look like the image below.



This completes the necessary steps of the hardware assembly!

## Configure the Raspberry Pi

Now that we’ve hooked up our Pi Servo Hat and assembled the hardware, we’re ready to boot the Raspberry Pi Zero W for the first time! To configure the Raspberry Pi, we need to:

- Power the Pi Zero W.
- Connect the Pi Servo Hat’s Serial-to-USB Converter.
- Connect to the serial debugging console on the Pi Zero W.
- Enable camera interface, I2C, and SSH on the Pi Zero W.
- Update the software on the Pi Zero W.
- Download and install the camera interface software from GitHub.
- Make some changes to enable the pan-tilt control functionality.
- Download the Pi Servo Hat interface software from GitHub.
- Make some changes to start the Pi Servo Hat interface software on boot.

Let’s walk through these steps in more detail!

### Power the Pi Zero W

Using a sufficient 5V wall adapter, we can power the Pi Zero W. Plug the wall adapter into a wall outlet for power. Then connecting the micro-b from the power supply to the Pi Zero W's micro-b connector labeled as "PWR IN."

## Connect the Pi Servo Hat's Serial-to-USB Converter

With the Pi Servo Hat stacked on the Pi Zero W, take a micro-B USB cable and connect it to the Pi Servo Hat's micro-B connector. Insert the other end to a computer's standard USB port. This will allow you to connect the Pi through a serial port connection.

## Open a Serial Terminal

Start by checking out our serial terminal basics tutorial. This will get you up and running with a serial terminal. Open a serial terminal program (i.e. PuTTY) to connect.

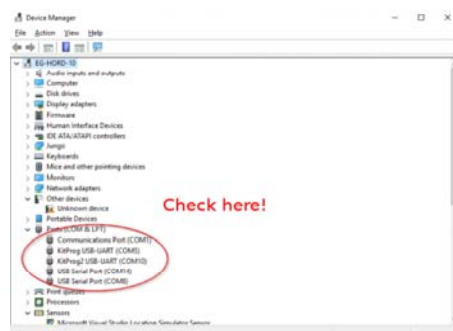
When you make your connection to the Raspberry Pi, you'll need to connect the serial terminal to the COM port that it enumerated on and connect at a baudrate of 115200. Failure to use the proper baud rate setting will result in weird characters and your connection failing to work.

To figure out which port to connect to, I recommend using the Arduino IDE. Under the "Tools" menu, there is a sub-menu for "Port". Since we had connected the USB cable to a computer's COM port already, make note of the items on the listed COM Ports. Then unplug a micro-B USB cable from your computer. Give it a few seconds, then re-open that sub-menu to see what item has disappeared. By process of elimination, we can determine the COM port that the Raspberry Pi enumerated to. Reconnect the cable to the COM port to verify. The COM port should reappear as the same COM number in the sub-menu.

**Installing FTDI Drivers:** It's possible that you'll need to install FTDI drivers to get the COM port to appear. If the following instructions don't result in a port appearing in the list, visit our FTDI driver installation tutorial.

## Windows

If you don't have the Arduino IDE installed and don't want to install it, you can find the same information using built in tools. Under Windows, open up your device manager (if you don't know how to do this, do a search online for OS specific information on how to do it since it's slightly different under various versions of Windows). Take note of the devices on the list, then unplug the Pi and see which port on the list disappears. The port which disappeared from the list is the one you want.



## Mac

Using a Mac OS, you'll need to open a terminal window. To figure out which port the Pi has connected to, type this command:

```
ls /dev/cu.usbserial-*
```

This will return a list of USB-Serial converter ports on the system. Take note of the devices on the list, then unplug the Pi and see which port on the list disappears. The port which disappeared from the list is the one you want. You can then connect to the port in question by typing

```
screen /dev/cu.usbserial-XXXXXXXX 115200
```

where the XXXXXXXXXX is replaced by information gleaned from the first command.

## Linux

Under Linux, the process is similar to Mac OS, only use this command to identify the serial port:

```
ls /dev/ttyUSB*
```

You may use “screen” to connect to the Pi:

```
screen /dev/ttyUSBX 115200
```

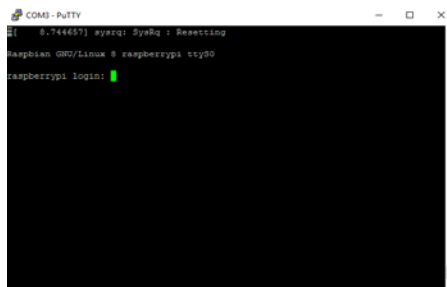
Again, the “X” should be replaced with information gleaned from the `ls` command above. If you receive an error about screen not being installed, you can install `screen` by typing this command:

```
sudo apt-get install screen
```

Then re-enter the above command to connect via `screen`.

## Log In to the Pi

When the Pi finishes booting (about 30 seconds) you should see a prompt on your serial terminal screen that looks like the image below. If you don't, try hitting the enter (or return) key. This will bring up the login information if it printed before you had your serial terminal ready.



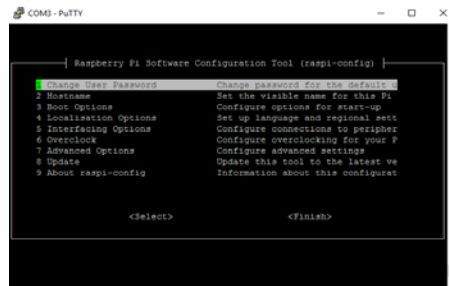
Log in to the system using the username “pi” and the password “raspberry”. You’re now logged in to the Raspberry Pi, and everything else we do we’ll do from this command prompt.

## Use the raspi-config Utility to Enable Camera, I2C, and SSH

There’s a nifty text-only utility called “raspi-config” that will allow you to enable the camera interface, I2C, and remote access via SSH. To use it simply type this command:

```
sudo raspi-config
```

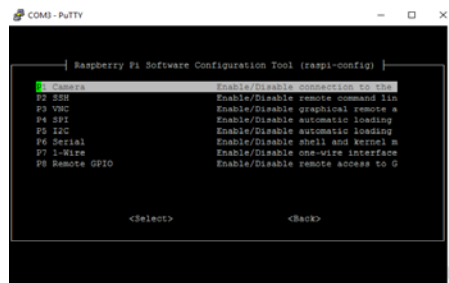
If you're curious, the "sudo" portion of the command tells the OS to run the command as a superuser – basically, a user with full system permissions. This is the menu that you'll then be presented with:



From here you can make several changes. The first change you should make is to change the password for the default user. Highlight option 1 and hit enter. You'll get a warning that you will be asked to change the password, then a prompt will appear at the bottom of the screen asking you to enter a new password. You'll be asked to repeat it, just to make sure that the password is as you intended, then you'll receive a message that the password has been changed.

**Please change your user password!** Leaving the default user password with SSH enabled is a security risk that could allow a malicious user to take over your device remotely.

Now, use the arrow keys to move down to choice 5, "Interfacing Options". You'll be presented with this menu:



Highlight option "P1 Camera", and hit enter. That will bring up this screen:



Highlight "Yes" and hit enter. You'll receive a confirmation message:





Hit enter to leave this screen. You'll be back at the main menu. Return to the "Interfacing Options" menu again and repeat the process two more times, once for the "I2C" option and once for the "SSH" option.

This concludes the work you need to do with the raspi-config utility. From the main menu, press tab twice to highlight "Finish", then hit enter. If you're asked whether to reboot now, select "Yes" and hit enter. Again, it will take as much as 30 seconds for the login prompt to reappear.

## Configure the /etc/network/interfaces File

You'll now need to tell the Raspberry Pi to use the "wpa\_supplicant.conf" file that you created earlier. Type the following command in the serial terminal:

```
sudo nano /etc/network/interfaces
```

Add the following lines to the end of the file:

```
auto wlan0
allow-hotplug wlan0
iface wlan0 inet dhcp
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Hit CTRL-O and enter to save the file, then CTRL-X to quit the nano text editor.

Now reboot the Raspberry Pi by typing:

```
sudo reboot
```

## Update the Pi

You must now update your Pi's software to the latest version. To do this, enter these two commands at the command prompt:

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

This will ensure that your Pi software is up-to-date. The first command takes a minute or two to run, but the second will take much longer. Be patient!

## Install the Remote Camera Web Interface Software

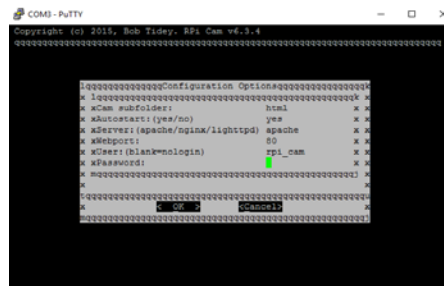
You'll now want to download and install the camera interface software. To do so, execute the following four commands in order:

```
git clone https://github.com/silvanmelchior/RPi_Cam_Web_Interface.git
cd RPi_Cam_Web_Interface
chmod u+x *.sh
./install.sh
```

The first command fetches the software from GitHub. You don't need a GitHub account for this to work. However, if you enter the command or URL wrong, `git` may ask for a username. The second enters the directory that was just fetched from GitHub. The third changes the user permission of the shell scripts in that directory so a user can execute them. The fourth runs the install script for the software.

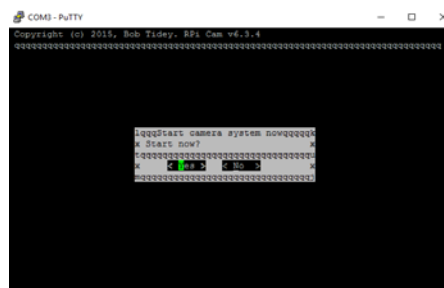
Running the fourth command will launch an interactive script allowing you to change various settings about how the software is set up. We'll walk you through it so your setup mirrors ours for the sake of the rest of the tutorial. It takes a minute or two for the interactive section to come up. Again, patience!

Here's the interactive interface for the installation script. **Use the arrow keys, not TAB, to move through the fields!** Note that we've left everything the same. The only exception is that we added a username and password (but that's not shown). Once you've finished entering a username and password, you may hit the TAB key to go to the "OK" button and hit "Enter" to select it.



```
CGMI - PuTTY
Copyright (c) 2015, Bob Eidey. RPi Cam v6.3.4
=====
Configuration Options
=====
# Install dependencies
# cam subfolder:      h264
# Autostart (yes/no)  yes
# xServer (apache/nginx/lighttpd)  apache
# xServer:            NG
# xUser: (blank=nologin)  rpi_cam
# xPassword:
=====
[ OK ] [ Cancel ]
```

The installation script will download and install all of the dependencies for the project. This can take some time (up to several minutes) so be patient. When everything is finished, you'll be asked if you want to start the service now. Select "Yes".



```
CGMI - PuTTY
Copyright (c) 2015, Bob Eidey. RPi Cam v6.3.4
=====
=====[Start camera system now]====
# Start now?
=====
[ Yes ] [ No ]
```

## Test the Camera Web Interface

Now let's make sure the camera is working before adding in the pan-tilt functionality. To do so, we first must determine the IP address of the Pi Zero W. Type this command:

```
ifconfig
```

Doing so should bring up the following information:

```

COM3 - PuTTY
pi@raspberrypi:~/RPi_Cam_Web_Interface$ ifconfig
lo        Link encap:Ethernet  HWaddr 00:00:00:00:00:00
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:65536 Metric:1
          RX packets:200 errors:0 dropped:0 overruns:0 frame:0
          TX packets:200 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:14604 (14.3 KiB)  TX bytes:14656 (14.2 KiB)

wlan0    Link encap:Ethernet  HWaddr b8:27:eb:f1:aa:1b6
          inet addr:10.8.252.217  Netmask:10.8.252.255  Mask:255.255.252.0
          inet6 addr: fe80::61a1:78bf:9c5:1ad3/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:4218 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4339 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:24046303 (24.0 MiB)  TX bytes:696519 (680.1 KiB)

pi@raspberrypi:~/RPi_Cam_Web_Interface$

```

There's lots of information here but we're only concerned with one small chunk: the "inet addr" entry in the "wlan0" section. In the example image above, that value is 10.8.252.217. This is the **IP address** of your camera.

Open up a web browser and type in that number plus "/html", for the web address. For example, I'd type in "10.8.252.217/html" in the address bar. If everything works as expected, you should be presented with a login request. **This is the username and password that you entered when setting up the camera web interface, not the login information for the Pi itself!** You'll be rewarded with an interface screen similar to the one below.



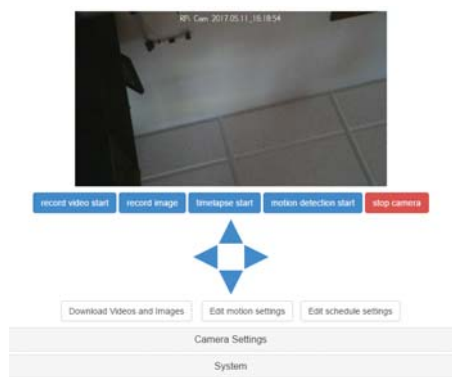
## Adding Pan-Tilt Support

The RPi Cam Interface software supports pan-tilt functionality, but it needs to be enabled. To do this, there are several steps required.

First, we must tell the software that we wish to enable pan-tilt functionality. This is done by renaming a file hidden deep within the directory structure using this command:

```
sudo mv /var/www/html/pipan_off /var/www/html/pipan_on
```

After running this command, go back to your web browser and refreshing the camera interface. Upon reloading, the page should now have four directional arrows directly beneath the image, as seen below.



Next, we need to create a system pipe in order to send commands to the interface software that controls the pan-tilt functionality. A pipe is like a file, only it has no static contents. Only contents defined at run time by the program will control the motors.

```
sudo mknod /var/www/html/FIFO_pipan p
sudo chmod 666 /var/www/html/FIFO_pipan
```

Now that we've created the pipe from the interface software, we need to fetch the control software that runs the servo motors. We've created a simple python program that does just that. Again, we're going to fetch that file from GitHub, using the following commands:

```
cd
git clone https://github.com/mhord/RPi_PanTilt_Cam.git
RPi_PanTilt_Cam/pantilt.py &
```

The first command ensures that you are in the pi user home directory; later instructions will depend upon that being the case. The second command fetches the program from Github, and the third executes it. As soon as you enter the third command, you should see and hear the motors twitch a little as the software centers them.

Now go back to the interface webpage and click on the various arrows. As you click on them, you should see the pan-tilt mechanism respond.

Finally, we need to set up our system to launch the pan-tilt software with the startup of the computer. To do this, we need to edit the "rc.local" file, which handles various on-boot functionality. To open the "rc.local" file, enter this command:

```
sudo nano /etc/rc.local
```

This will launch the nano text editor with "rc.local" loaded into it. Use the down arrow key to scroll to the bottom of the file and add the following text **just above the line that reads "exit 0"**:

```
python /home/pi/RPi_PanTilt_Cam/pantilt.py &
```

It's important that you place that line above the "exit 0" line or it will never be executed. Now, hit CTRL-o to save the file and CTRL-x to quit nano text editor.

## Congratulations!

You've completed the initial setup of your Raspberry Pi Zero W Pan-Tilt Camera! This is outside the scope of the tutorial but there are a few more things you might want to do:

- **Set up motion activation** - The page documenting the interface software has useful information about setting up motion activated image capture.
- **Set a static IP address** - By default, the Pi Zero W may be assigned different IP addresses by the DHCP server on your network each time it boots. This can make interacting with the camera difficult, as you need to check the camera to figure out its IP address after each reboot. Setting up a static IP address is a complex topic and dependent upon the settings of your network, and is left as an exercise for the reader.
- **Communicate with the camera over SSH** - Now that the camera is up and connected to your WiFi network, you no longer need to connect to it via the serial cable. Connecting via SSH is relatively simple, and we won't go into the details here.

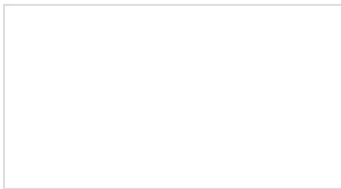
## Resources and Going Further

Now that you've successfully got your Raspberry Pi Zero Wireless Pan-Tilt Camera up and running, it's time to incorporate it into your own project!

For more information, check out the resources below:

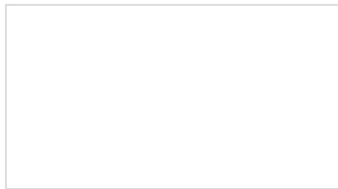
- SparkFun Pi Servo Hat Hookup Guide
- Raspberry Pi Foundation: Raspbian Jesse Image
- Etcher
- Official Raspberry Pi Camera Web Interface Documentation - We've gotten you through installation and setup, but to get the most out of the camera, you'll need to check that page out.
- Raspberry Pi Camera Web Interface GitHub Repository - Official repository for the web based interface for controlling the Raspberry Pi Camera, includes motion detection, time lapse, and image and video recording.
- SparkFun Pan/Tilt Example Code GitHub Repository - Demo code used in the *Setting Up the Pi Zero Wireless Pan-Tilt Camera* tutorial.
- SparkFun Product Showcase: Pi Zero W Pan-Tilt Camera Kit Demo
- Shawn's Pan/Tilt Time Lapse Demo Code - Python demo code that was used for SparkFun's product showcase

Need some inspiration for your next project? Check out some of these related tutorials:



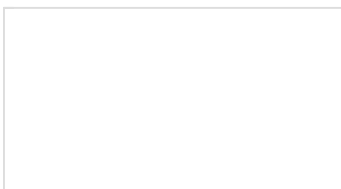
### Pushing Data to Data.SparkFun.com

A grab bag of examples to show off the variety of routes your data can take on its way to a Data.SparkFun.com stream.



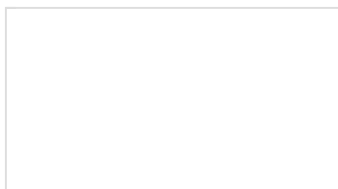
### Hackers in Residence: The Sound Visualizer

A fun project that uses a Raspberry Pi and a custom Java app to create your own sound visualizer using a RGB LED matrix.



### Single Board Computer Benchmarks

How to set up different benchmarking programs on single board computers or computing modules and run them. The results for various generations are shown on the subsequent pages.



### Raspberry Pi SPI and I2C Tutorial

How to use the serial buses on your Raspberry Pi.