

Kamigami Pi 0 W Setup

Jay Monga

May 27, 2021

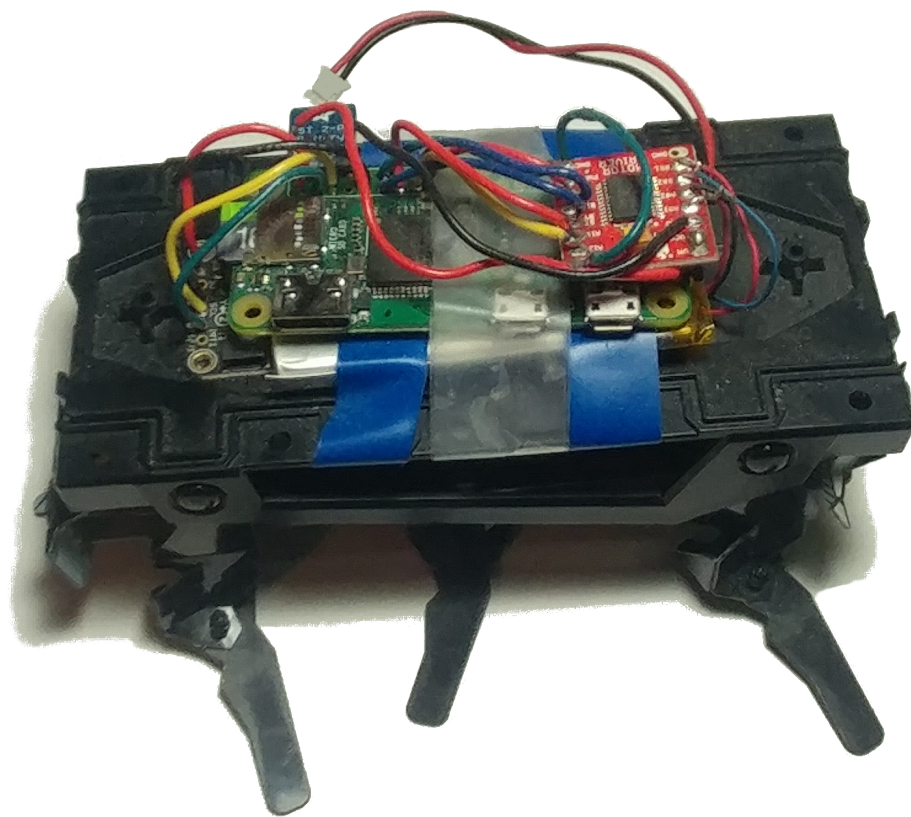


Figure 1: Assembled Kamigami with Raspberry Pi 0 and attached peripherals

Contents

1	Introduction	3
2	Components	3
3	Setting up the Pi 0 W	3
3.1	Installing Raspbian	4
3.2	Installing ROS	5
3.3	Getting ROS Packages	5
4	Adding the Pi 0 W to the Kamigami	5
4.1	Removing Kamigami Electronics	5
4.2	Wiring	5
4.3	Mounting	6
5	Network Setup	6
5.1	Connecting the Raspberry Pi to WiFi	6
5.2	Setup Hostnames and Master	8
5.3	Allow ROS Master Machine to Launch Nodes on the Pi 0 W	8
6	Development	8

1 Introduction

The [Kamigami toy](#) is an inexpensive platform for legged multi-robot research. With a fair bit of work, you can replace its circuit board with a Raspberry Pi 0 W, allowing you to use the robot in a ROS system, which allows for easier multirobot applications.

2 Components

Here are the parts I used in my setup, but you can feel free to add or remove peripherals (like the IMU) as you see fit. I've hyperlinked product links, but you may find them cheaper from other sources.

- [Kamigami toy](#) This is the actual robot you will be using. There are other models with different shells, but I believe the only differences are cosmetic, which will not be apparent for our uses since we will not build the Kamigami with the added shell pieces to reduce weight.
- [Raspberry Pi 0 W](#) This model of the Raspberry Pi gives us an inexpensive, light, and compact way to add a linux interface to the Kamigami. This Pi is equipped with WiFi and Bluetooth out of the box, making networking relatively painless.
- [Motor Driver](#) Sparkfun's motor driver board allows us to control both speed and direction of both motors without much hassle. Although the output current is a bit low, I did not find this impeding the operation of my Kamigami.
- [IMU](#) This cheap board comes with an accelerometer and gyroscope, but it may be worth spending a bit extra to get something with a magnetometer for more accuracy. Still seems to do the job decently well.
- [SD Cards](#) These will be needed for storing the OS, ROS, and perhaps data log files for your robot.
- [LiPo Battery Charger](#) Depending on what battery you use to power the Kamigami, you will end up recharging it somewhat often. I have found this charger works relatively quickly and is pretty straightforward to use.
- [JST-PH 2-Pin Right Angle Connector](#) Soldering this onto the Raspberry Pi makes it really easy to connect the battery for power and disconnect for charging.
- [LiPo Battery](#) This battery was pretty satisfactory in terms of compromise between size, weight, cost, and power storage and can power the Kamigami robot with added peripherals for several hours of use before a recharge is needed.

3 Setting up the Pi 0 W

Note: These instructions are incomplete. While I work on writing out the full instructions, please download and flash the disk image with everything pre-installed instead.

You're welcome to flash an image of my OS with ROS setup, but I would recommend following the instructions below to get the most updated versions of everything, and to catch any mistakes/things I could be doing better. Note: there seem to be [instructions for getting ROS Melodic](#)

[setup on the Pi 0](#), but I have not followed these as I did not see them when I had started my project. In the future, I may attempt to migrate over to Melodic and will update the instructions here. My instructions are based off of [this guide](#).

3.1 Installing Raspbian

1. First, you will want to [get the newest version of Raspbian](#), now called Raspberry Pi OS, flashed to your SD card. As of writing this, the newest version is based off of Debian Buster, so I will tailor my instructions to that. I would recommend the light/minimal version, if it is available.
2. You need to setup ROS repositories for your system so we can install ROS things. Make sure to substitute 'buster' with the appropriate OS in use.

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu buster main" >
↳ /etc/apt/sources.list.d/ros-latest.list'
wget https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -O - |
↳ sudo apt-key add -
```

Make sure to run update & upgrade commands too.

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

3. You will also need bootstrap dependencies.

```
sudo apt-get install python-pip python-setuptools python-yaml
↳ python-distribute python-docutils python-dateutil python-six
sudo pip install rosdep rosinstall_generator wstool rosinstall
```

4. The last thing dependency wise to do is to initialize rosdep.

```
sudo rosdep init
rosdep update
```

5. Now it's time to move on to starting the actual build process. You will want to make some kind of directory to serve as the ROS workspace for your core ROS packages and go there.

```
mkdir ~/ros_ws
cd ~/ros_ws
```

- 6.

3.2 Installing ROS

3.3 Getting ROS Packages

4 Adding the Pi 0 W to the Kamigami

4.1 Removing Kamigami Electronics

4.2 Wiring

Before mounting all additional components to the kamigami, I would recommend soldering on all necessary wires between components. Not only is it easier to wire the components when they aren't fixed to the kamigami body, but you also don't have to worry about burning through the plastic chassis when soldering. A Fritzing diagram of the circuit is shown below; the Fritzing file should be in the same directory as this document. The pinout for the Raspberry Pi can be seen [here](#), and the pinouts for other components can be viewed at the appropriate pages in the [components section](#).

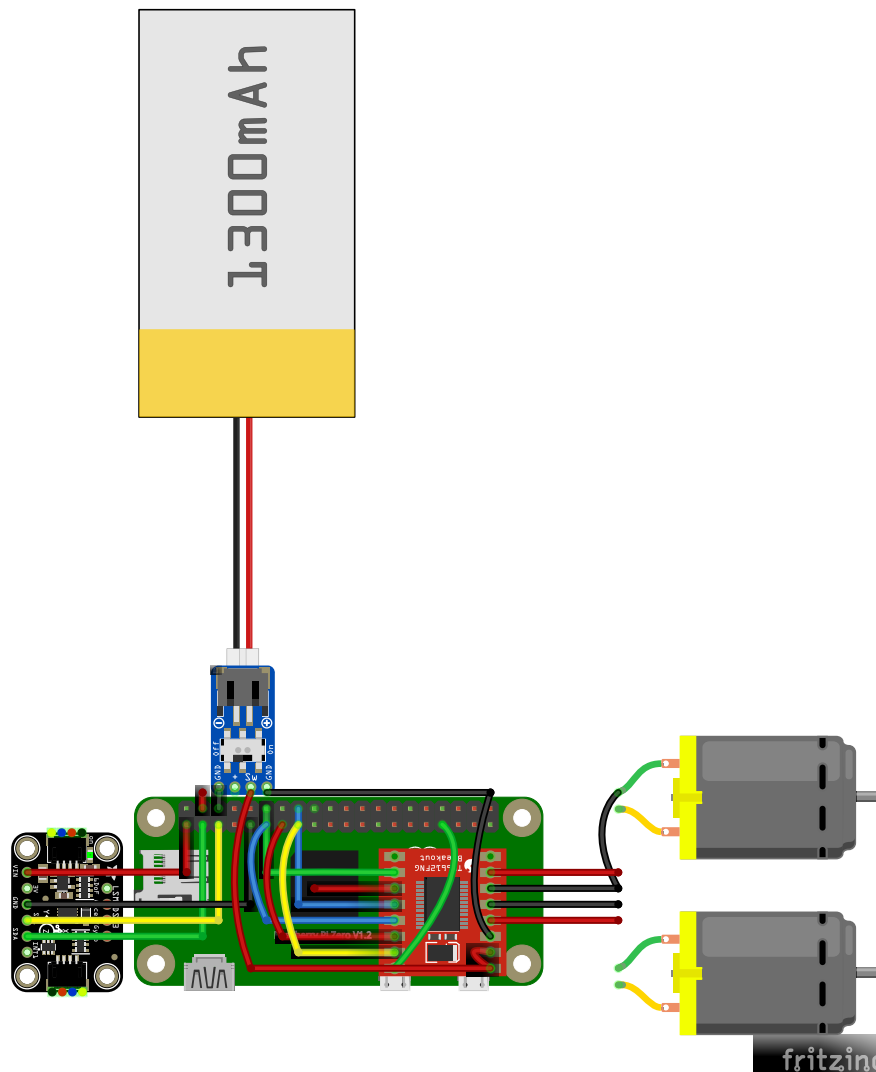


Figure 2: Wiring of Raspberry Pi 0

As we can see, every component is powered by a single LiPo battery. A 3.7V 1300 mAh battery is shown in the image, but my implementation uses a 3.7V 1200 mAh battery. The Raspberry Pi and motor drivers receive power directly from the battery, and the IMU is powered from the Raspberry Pi 0's 3.3V output. Notice that the Raspberry Pi is taking in 3.7V into its 5V power input; this is fine as the Raspberry Pi internally steps down the power to 3.3V for all pins. The DC motors in the image refer to motors on actual Kamigami, which you aren't expected to remove but you should wire into the motor driver.

4.3 Mounting

After completing the wiring, we can affix the components to the Kamigami. You can refer to the image of the [assembled kamigami](#) to see how I've done it. I attached the IMU to the chassis using mounting tape so that there is some damping. The battery is taped onto the chassis, and then the Raspberry Pi is securely taped to the top of that. The motor driver is held in place by the stiffness of the wires.

5 Network Setup

This system relies on a central computer able to communicate with several Kamigami robots easily. Thus, there is some degree of networking setup with ROS in order to get this working right. The current system allows us to run a launch file on the central computer that starts up nodes on both the central computer and the Kamigami.

5.1 Connecting the Raspberry Pi to WiFi

These instructions were adapted from [here](#).

First, we must ensure that the Raspberry Pi's are on the same network as the main computer. Note that unless you are using a special WiFi dongle, the Raspberry Pi 0 W can only connect to 2.4 GHz WiFi networks. Take the microSD card where your Raspberry Pi operating system was installed and plug it into a computer where you can edit files on that microSD. You may need to use an adapter for this. The microSD card should have a `boot` folder that you can access.

In this folder, you will create a file `wpa_supplicant.conf` with information about your WiFi network. The file should be formatted as follows:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=<Insert 2 letter ISO 3166-1 country code here>

network={
    ssid="<Name of your wireless LAN>"
    psk="<Password for your wireless LAN>"
}
```

Fill in the appropriate fields with your network information. The two letter country code for the USA is just US. You can find other country codes [here](#). For `ssid`, put in the SSID or name of your network. For `psk`, enter in your WiFi network's password. If your network does not use

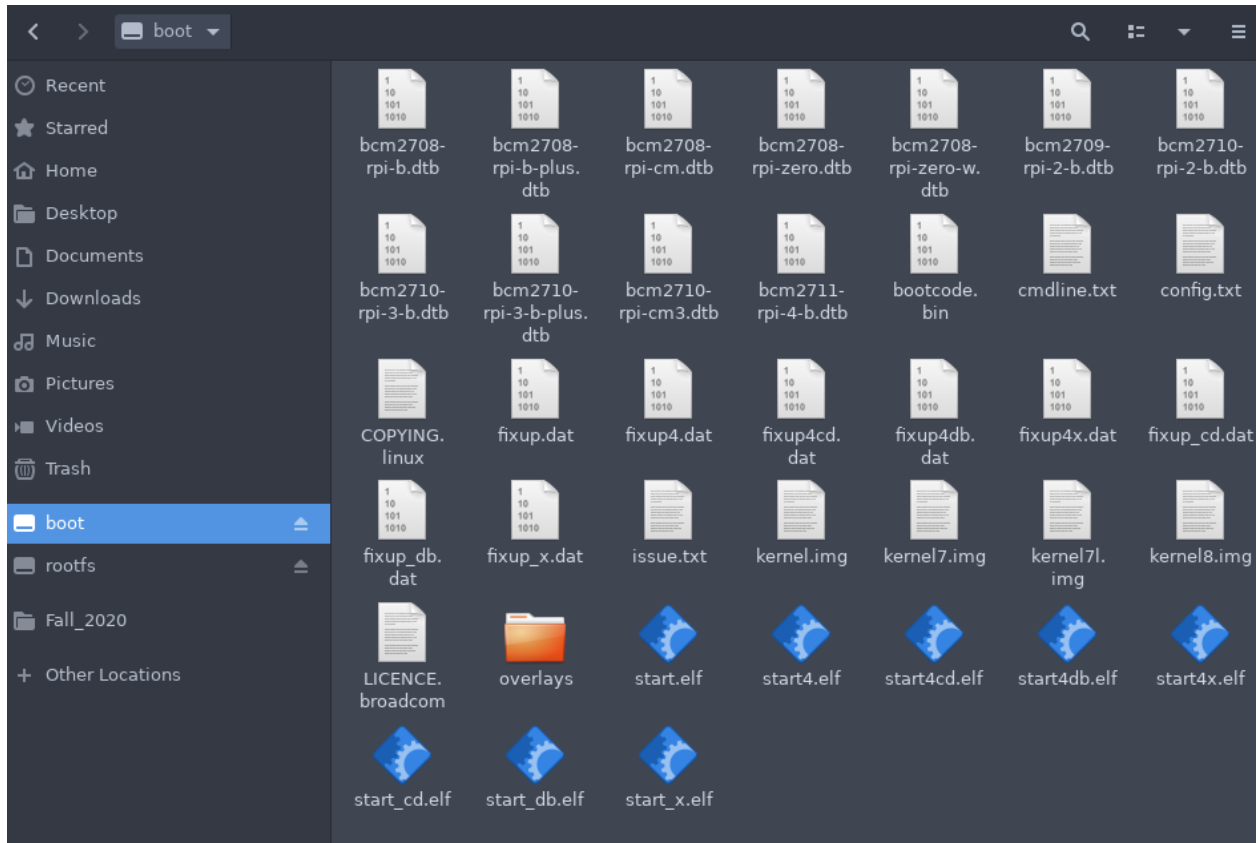


Figure 3: Boot Folder

standard WPA2 security or you just need additional information, then you should refer to [this](#).

If I lived in the USA, my WiFi name is **freewifi**, and my WiFi password is **password**, my `wpa_supplicant.conf` would look like this

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

network={
    ssid="freewifi"
    psk="password"
}
```

After saving the file to `boot/wpa_supplicant.conf`, you may remove the microSD card from your computer, plug it into the Raspberry Pi, and then turn on the Raspberry Pi. The green power light for the Raspberry Pi will probably flicker for some time, and then should eventually stay a constant green. At this point, you should wait for a few minutes and then attempt to reach the Raspberry Pi over a network connection. In your terminal, enter the command

```
ssh pi@raspberrypi.local
```

You should be prompted for a password. The default password for a fresh Raspberry Pi is just `raspberry`. We will change this shortly.

If you have made it here, congrats! You have successfully connected your Raspberry Pi to a WiFi network. We can now set it up to recognize our main computer as the ROS master.

5.2 Setup Hostnames and Master

5.3 Allow ROS Master Machine to Launch Nodes on the Pi 0 W

6 Development

The kamigami ROS package stack can be found at the [kamigami_common github repo](#). Here, every node, topic, message type, launch file is documented.