Kamigami Pi 0 W Setup

Jay Monga

September 16, 2021
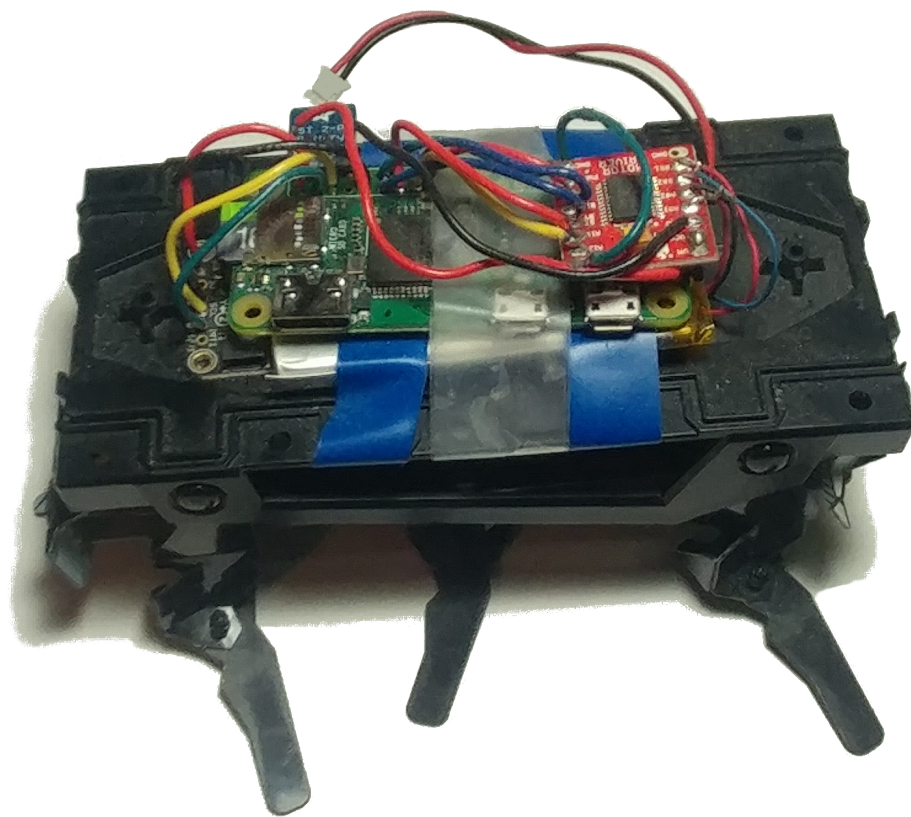
Figure 1: Assembeled Kamigami with Raspberry Pi 0 and attached peripherals

Contents

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 is 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 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 spendign 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

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.

3.1 Installing Raspbian

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.

3.2 Getting ROS

SSH into the Pi 0 and run the following commands to install ROS. Credit to this repo.

```
sudo apt update
sudo apt install -y build-essential gdebi
mkdir -p ~/tmp && pushd ~/tmp
wget https://git.io/JzfTV -O ros4raspbian.zip
unzip ros4raspbian.zip
sudo gdebi ros4raspbian-melodic-armhf-1.0.0.deb
popd
rosdep update
```

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 reccomend 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 chasis 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 approriate pages in the components section.

As we can see, every component is powered by a single LiPo battery. A 3.7V 1300 mAh battery is show in the image, but my implementation uses a 3.7V 1200 mAh battery. The Raspberry Pi and motor drivers recieve 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 chasis using mounting tape so that there is some damping. The battery is taped onto the chasis, and then the Raspberry Pi securely taped to the top of that. The motor driver is held in place by the stiffness of the wires.
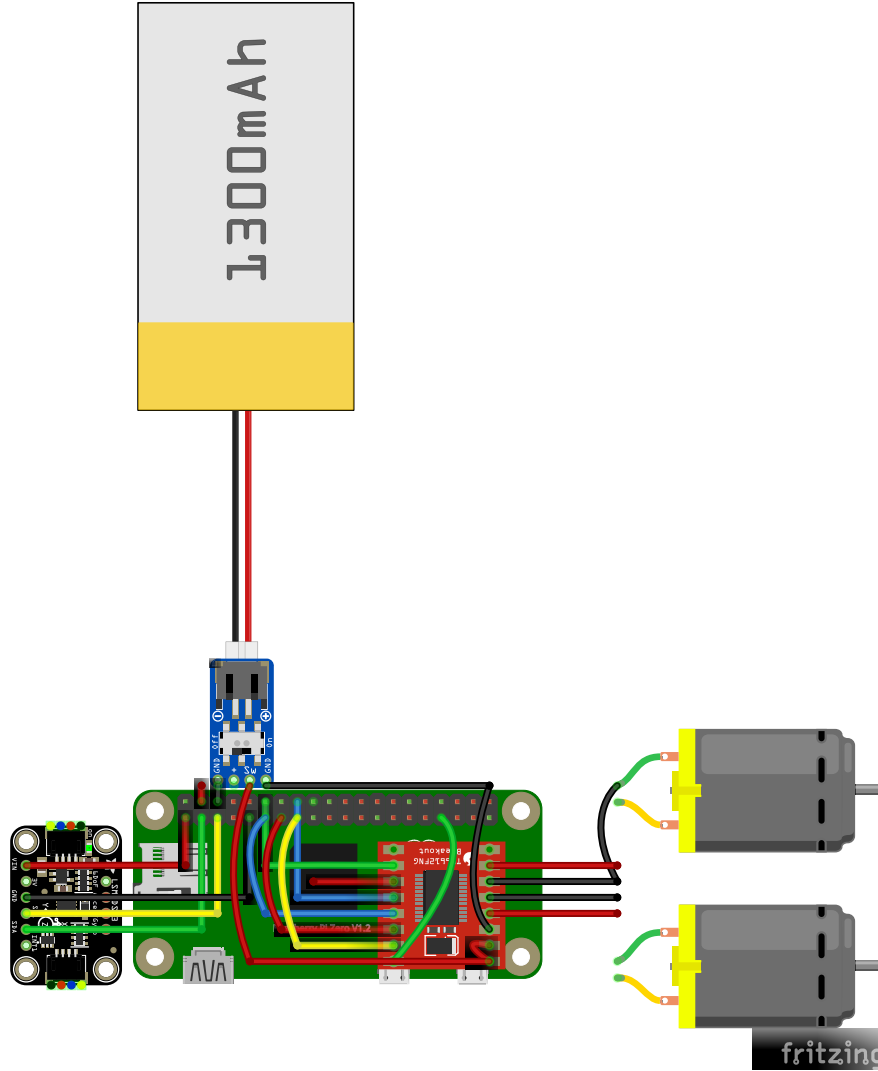
Figure 2: Wiring of Raspberry Pi 0

5   Network Setup

This system relies on a central computer able to be 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 insturctions 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 was installed and plug it into 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.
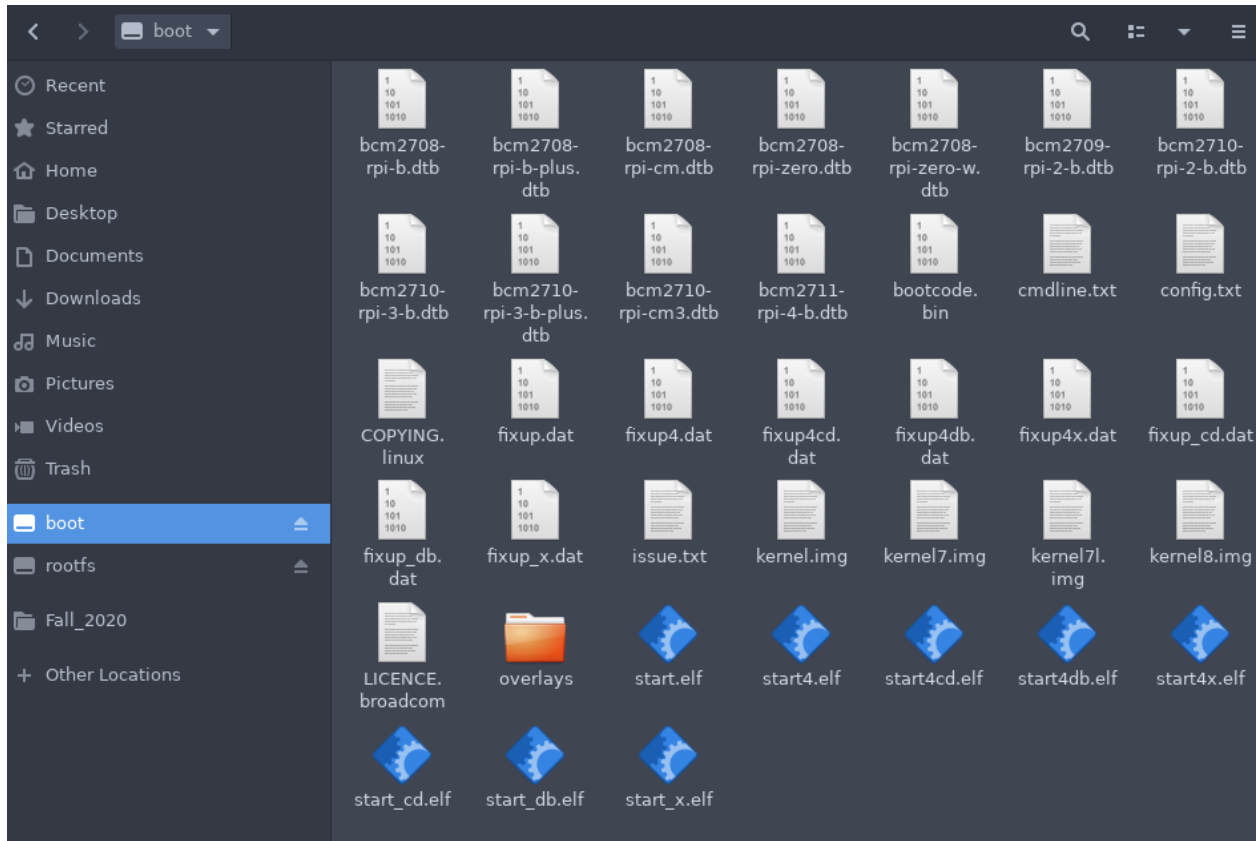
Figure 3: Boot Folder

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 newtork. For `psk`, enter in your WiFi network's password. If your network does not use 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 looks 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.