DKE LABS

grim-trigger-docs Documentation

Release 0.0.1

Matt Oldfield

Sep 17, 2017

1 DDrone Hardware **3**
1.1 Devices Used . . . 3

2 IOIO (YoYo) documentation **5**
2.1 IOIO usage info . . . 5

3 Raspberry Pi Zero W **7**
3.1 Raspberry Pi Zero W . . . 7

4 Power hub devices **9**
4.1 Rav Power Portable Charger . . . 9

5 DDrone Purpose **11**
5.1 Summary . . . 11
5.2 App State . . . 11
5.3 Hardware Diagram . . . 13
5.4 Communication with the Raspberry Pi Zero W . . . 13

6 Permissions **15**

7 Raspberry Pi Zero W **17**
7.1 Bluetooth . . . 17

8 Creating Raspian Lite Image **19**
8.1 Summary . . . 19
8.2 Instructions . . . 19
8.3 Options . . . 20
8.4 Wifi . . . 20
8.5 Bluetooth . . . 20

9 Writing documentation **23**
9.1 Basic Structure . . . 23
9.2 Writing Sphinx and reStructured text documentation . . . 23

10 Main Section **25**
10.1 Secondary Section . . . 25

Info on the general development process of DDrone

- *Hardware*
- *Android App*
- *Raspberry Pi App*
- *Writing These Docs*

CHAPTER

ONE

DDRONE HARDWARE

Devices Used

Table 1.1: Hardware Devices

Device	Description
Android Device	
Raspberry Pi Zero W	
IOIO Board	
Twitter Kit	
RAVPower 20100mAh USB Type-C Power Bank	

CHAPTER

TWO

IOIO (YOYO) DOCUMENTATION

IOIO usage info

- IOIO Wiki

CHAPTER

THREE

RASPBERRY PI ZERO W

Raspberry Pi Zero W

- Specs

CHAPTER

FOUR

POWER HUB DEVICES

Rav Power Portable Charger

- Rav Power Site

CHAPTER

FIVE

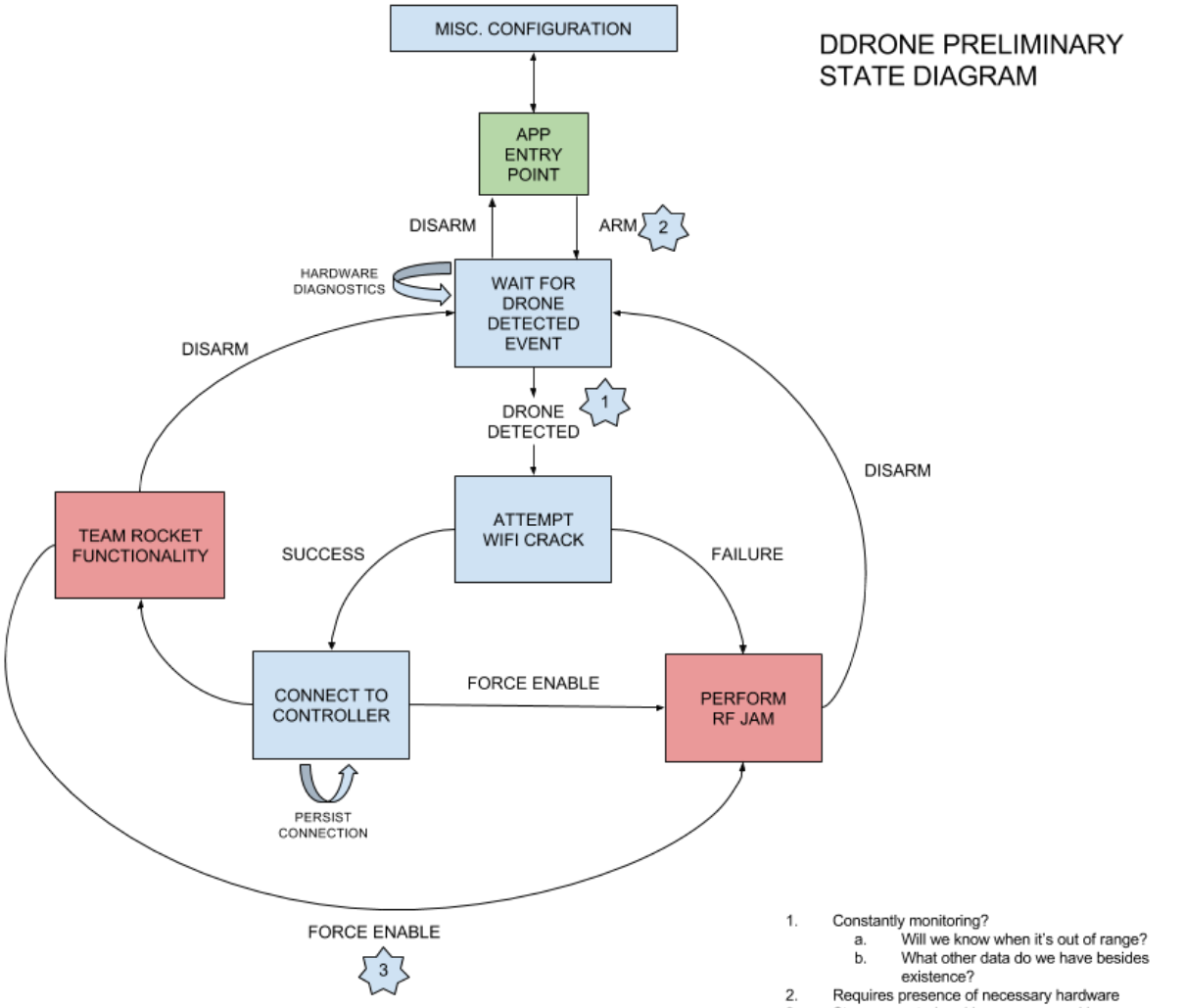
DDRONE PURPOSE

Summary

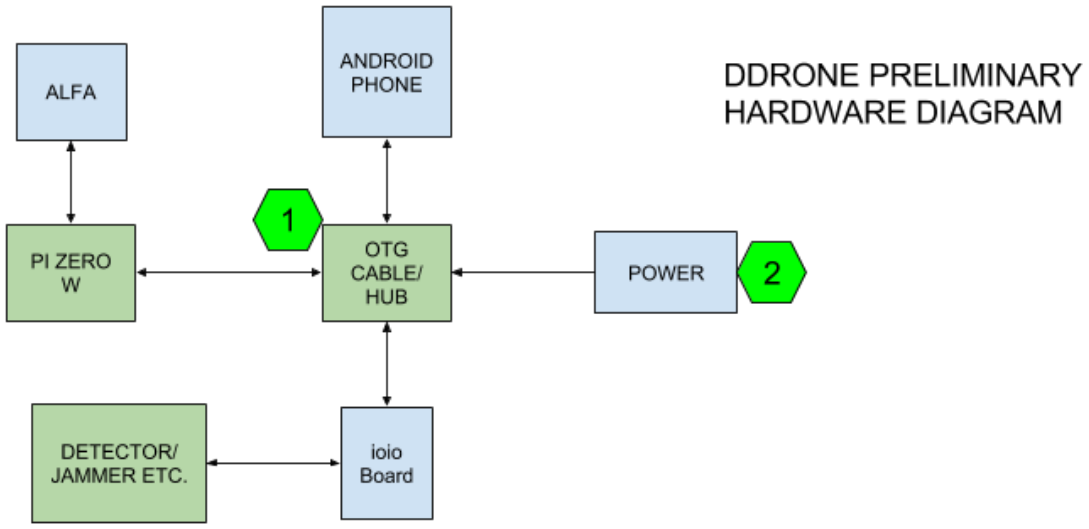
DDrone is an Android application that is used to interfere with a DJI Phantom 3 Standard drone by...

App State

State is managed using the `SharedPreferences` library.


DDRONE PRELIMINARY
STATE DIAGRAM
MISC. CONFIGURATION
APP
ENTRY
POINT
DISARM
ARM
2
HARDWARE
DIAGNOSTICS
WAIT FOR
DRONE
DETECTED
EVENT
DISARM
DRONE
DETECTED
1
DISARM
TEAM ROCKET
FUNCTIONALITY
ATTEMPT
WIFI CRACK
SUCCESS
FAILURE
CONNECT TO
CONTROLLER
FORCE ENABLE
PERFORM
RF JAM
PERSIST
CONNECTION
FORCE ENABLE
3
1. Constantly monitoring?
a. Will we know when it's out of range?
b. What other data do we have besides existence?
2. Requires presence of necessary hardware
3. Stay connected and keep app armed in background, ready to force jam. Let's use open DJI app.

Hardware Diagram



Communication with the Raspberry Pi Zero W

There are several ways to communicate through *OTG* with the Raspberry Pi Zero W including serial and SSH.

CHAPTER

SIX

PERMISSIONS

..Variables .. include:: ../vars.rst

Permissions are declared in `AndroidManifest.xml`.

Permission	Needed For
WRITE_EXTERNAL_STORAGE	Required for the **SharedPermissions** lib
READ_EXTERNAL_STORAGE	Required for the **SharedPermissions** lib
ACCESS_NETWORK_STATE	
CHANGE_WIFI_STATE	Connect with the drone's controller once password has been cracked.
ACCESS_COARSE_LOCATION	
INTERNET	
BLUETOOTH	
BLUETOOTH_ADMIN	

CHAPTER

SEVEN

RASPBERRY PI ZERO W

Bluetooth

Install on pi

```
$ sudo apt-get install bluez python-bluez
```

Append `DisablePlugins = pnat` to `/etc/bluetooth/main.conf` because the `pnat` plugin messes with bluez

Make device discoverable

```
sudo hciconfig hci0 piscan
```

Set the device name

```
sudo hciconfig hci0 name 'Device Name'
```

Wake console from sleep

Useful for testing with the pi commands and there is an hdmi screen plugged in.

```
sudo bash -c 'echo -ne "\033[9;0]" > /dev/tty1'
```

CHAPTER

EIGHT

CREATING RASPIAN LITE IMAGE

Summary

Here are instructions for creating a *"headless"* Raspian Jessie Lite `.img` on a microSD card to use with a Raspberry Pi Zero W. To make the headless image various config files have to be edited and `SSH` has to be turned on so that `SSH` can work over USB OTG along with an Android `SSH` library to send commands.

Instructions

1. Download the Raspian Jessie Lite image
2. Plug in microSD card and use Etcher to install the `.zip` file.
3. Remount that microSD card
4. Open the `config.txt` file in the `boot` partition and add `dtoverlay=dwc2` to it's own line at the bottom of the file.

Note: The `dwc2` is a USB driver implemented with the device tree overly

1. Open the `cmdline.txt` file and just after `rootwait` put a space then add `modules-load=dwc2,g_ether` and follow it with a space if there are more commands after.
2. In the same root directory of the boot partition create a blank file named `ssh` with no extension using `touch ssh`. When the system boots up for the first time it will delete that file and enable ssh.
3. Put the microSD card into the Raspberry Pi Zero W and plug in only the data micro USB cable into both the pi then the computer/android and let it boot up for a minute or two.
4. If you are able to see a second ethernet connection in network monitor with *RNDIS* in the name the it worked.
5. Go into network connections and edit the connection in the previous step. It should be something like *Wired Connection 2*.
6. Click on the *IPv4* settings tab and select *Link-Local Only* and also select use IPv4 connections only (Ubuntu). *(If you skip this step you will make your life harder)*
7. You can now `ssh` into the pi using `$ ssh pi@raspberrypi.local` with the password `raspberry`. In order to use the local hostname you need *Bonjour* on a Mac or Windows and *Avahi* for Ubuntu. There is also *Zeroconf* which *may* be able to be used with Kali.
8. Immediately change the password using the `passwd` command to avoid bots gaining access.

Here is a detailed document listing the various ways to connect and use the Raspberry Pi Zero W as a USB Gadget

Options

remove known hosts public keys

```
$ ssh-keygen -f "/home/asmattic/.ssh/known_hosts" -R raspberrypi.local
```

Change hostname

```
$ sudo raspi-config
```

Change Username

Change password

Boot into command line and require login

Interfacing options

allow shell from serial

Expand the filesystem hostname

dkelabspi

To add a new user with the same permissions as the pi user:

`sudo useradd -m fred -G sudo` This adds a new user called fred, creates a home folder, and adds the user to the sudo group. You now need to set a password for the new user:

`sudo passwd fred` Log out and log back with the new account details. Check your permissions are in place (i.e. you can sudo) by trying the following.

`sudo visudo` The visudo command can only be run by an account with sudo privileges. If it runs successfully, then you can be sure that the new account is in the sudo group.

Wifi

`/etc/wpa_supplicant/wpa_supplicant.conf`

```
# At bottom
network={
        ssid="NETWORK_NAME"
        psk="NETWORK_PASSWORD"
        key_mgmt=WPA-PSK
}
```

Bluetooth

Terminal example through bluetooth

https://hacks.mozilla.org/2017/02/headless-raspberry-pi-configuration-over-bluetooth/

https://learn.adafruit.com/install-bluez-on-the-raspberry-pi/installation

Security

Extra SSH Security measures

Scheduling tasks or on boot

https://www.raspberrypi.org/documentation/linux/usage/cron.md

CHAPTER

NINE

WRITING DOCUMENTATION

Basic Structure

```
Main Section
============

.. Comment (include file below)
.. include:: ../vars.rst

.. Link to secondary section that can be referenced anywhere
.. _secondary-sec:

Secondary Section
-----------------

Some regular text.

Bullet List

        * Bullet 1
        * Bullet 2
        * Bullet 3

Tertiary Section
~~~~~~~~~~~~~~~~

Some tertiary text.

.. Include any file as the literal text
.. literalinclude:: basic-structure.rst
```

Writing Sphinx and reStructured text documentation

- *Curated list of Sphinx docs <awesomeSphinxDocs_>_* - A curated list of awesome extra libraries, software and resources for Sphinx (Python Documentation Generator). Inspired by awesome-sqlalchemy.
- reStructured text for writers - Blog post about reStructured Text
- reStructured text documentation - Official documentation
- reStructured text cheat sheet - Cheat sheet written by matplotlib creators
- Sphinx documentation - Official documentation

- Read The Docs Sphinx - RTD styleguide

CHAPTER

TEN

MAIN SECTION

Secondary Section

Some regular text.

Bullet List

- Bullet 1
- Bullet 2
- Bullet 3

Tertiary Section

Some tertiary text.

```
Main Section
============

.. Comment (include file below)
.. include:: ../vars.rst

.. Link to secondary section that can be referenced anywhere
.. _secondary-sec:

Secondary Section
-----------------

Some regular text.

Bullet List

        * Bullet 1
        * Bullet 2
        * Bullet 3

Tertiary Section
~~~~~~~~~~~~~~~~

Some tertiary text.

.. Include any file as the literal text
.. literalinclude:: basic-structure.rst
```