Project Guide for “Coordinated control of autonomous aerial robots”

The goal of this Project Guide is to open-source the details of “Coordinated control of autonomous aerial robots” project accomplished at Tufts University by Pratik Chatrath under the supervision of Prof. Usman Khan during summer 2014. Developers can take help from this guide and build on it.

Table of Contents

Introduction

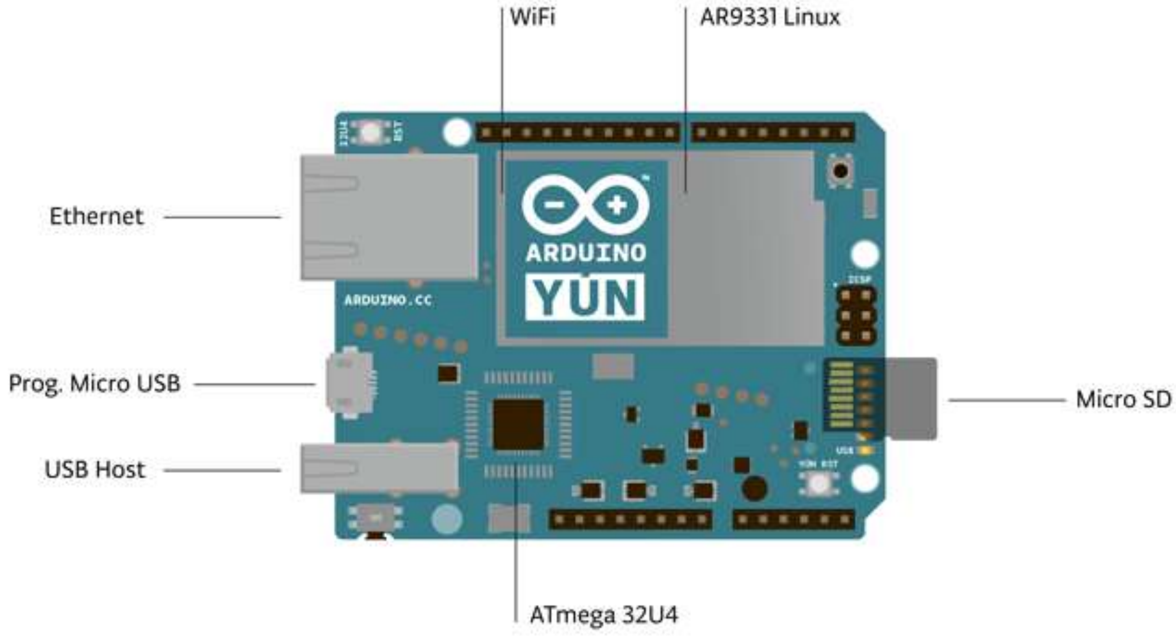
This project guide is step by step instruction guide to implement “Coordinated control of autonomous aerial robots”. It shows how to control multiple AR Drone 2.0 UAVs using Arduino Yun.

Chapter 1 explains details about Arduino Yun configuration from how to power the device to installing all the required packages. Chapter 2 describes the settings of AR Drone 2.0 for establishing Ad-hoc network.

Chapter 1: Setting up Arduino Yun

a) Initial Setup:

Arduino Yun requires a USB cable which has USB micro connector to power it. It can be powered through Laptop or by directly plugging in to a power source.



Steps to configure on-board wifi can be found at:
http://arduino.cc/en/Guide/ArduinoYun#toc13

b) Installing Latest Firmware

The following image is used as a refrence in this guide:

http://download.linino.org/linino_distro/master/latest/openwrt-ar71xx-generic-linino-yun-16M-250k-squashfs-sysupgrade.bin

Guide to install the new firmware can be found at:

http://arduino.cc/en/Tutorial/YunSysupgrade

c) Preparing SD Card and package manager

Arduino Yun’s Linux Microprocessor has 16MB Flash memory. To store large data/packages we need an SD Card. Steps to prepare sd card for installing packages are as follows:

Format SD Card in 16/32 VFAT. SD Formatter software is a good tool to do that. Link. Not all SD Cards are supported. We have tested our code on San Disk 16 GB, 4GB SD Cards.

All instructions here are with respect to Linux operating system. Open terminal window and ssh into Arduino Yun using instruction:

```
sshroot@linino.local
```

You can also write the ip address instead of linino.local by finding out the ip address assigned to Arduino Yun. To find IP address use command: *ping linino.local* in terminal window.

To check whether the sd card is detected by Arduino Yun or not use command:*mount*

You should find name of your SD Card in the list that comes up. Next you need to mount your sd card to /mnt/sda1. Here our sd card is at /dev/sda1. If you get a different location use that.

```
mount –t vfat /dev/sda1 /mnt/sda1
```

If you don’t find your SD Card, you need to check the format of SD card. You may need to manually mount the sd card. If even after proper formatting of the sd card, you dont find the sd card then your Arduino Yun has issues of handling external device in its kernel. So you will have buy a new Yun or get the older one replaced.

After you mount SD card, you are all set for installation procedure.

Yun has an OpenWrt- Yun Linux system. It uses Opkg package manager tool to install, update or remove additional applications and program. Opkg’s default location to install program is 16 MB Flash Memory which isn’t sufficient in our case. Hence, you have to change the destination location to external SD card.

To link external SD Card to opt directory type instruction:

```
ln –s /mnt/sda1 /opt
```

The package manager needs an updated database to display packages available for your system. Running the *opkg update* command updates the list of available packages. Due to the small flash memory available on the Yún, the database with the list of the package is only saved inside RAM. This means that you need to run the opkg update command every time you want to install a program after freeing the RAM or after a reboot. [1]

Next you need to edit opkg.config file in /etc. (press *esc* and *shit+ I* to get into edit mode) and type:

```
vi /etc/opkg.config
```

```
adddestsd /opt
```

in that file. Then press ESC and :wq for save and exit

Next edit /etc/profile file as follows (edit PATH and add LD_LIBRARY_PATH)

```
PATH= <current default path>:/opt/bin:/opt/sbin:/opt/usr/bin:/opt/usr/sbin
```

```
export LD_LIBRARY_PATH = $ LD_LIBRARY_PATH:/opt/lib:/opt/usr/lib
```

Then press ESC and :wq for save and exit

Next on command prompt give command – *source* to apply those changes

d) Installing node.js&Ideino

To install node.js and Ideino give the following commands: [2]

```
opkg update

opkg install node –d sd

cd /opt

curl -LOk https://dl.dropboxusercontent.com/u/12693688/ideino_0.0.5-1_ar71xx.ipk

opkg install ideino_0.0.5-1_ar71xx.ipk -d sd
```

Next you need to burn the proper fuses, to program Arduino side of Arduino Yun wirelessly.

So give the following commands:

```
cd /opt/ideino-linino/node_modules/ideino-linino-lib/ext/fermata/fermata_spi_pullenabled

run-avrdudefermata_spi_pullenabled.hex
```

e) **Installing node-ar-drone module**

Now you are all set to use the Ideino IDE. PresentlyIdeino is available in0.1 beta version.

Open browser window and type the urllininl.local you will get an option of “Play with Ideino” once you login. On clicking “Play with Ideino” option you will be redirected to a new page. Go to processes and start ideino andideino-autorun.

On doing so, Arduino Yun will start communication on port 2424. After a few seconds linino.local:2424 (or the Play with Ideino) will show you the welcome screen. The default password is doghunter

Create a new project called “drone” where you will prepare the code to fly UAVs. Next install the node-ar-drone dependency. Right click the package.json file and edit the default file. Changes to be done can be found at http://www.linino.org/tutorial-01-control-a-parrot-ar-drone-with-linino/

Ideino will run npm command and install the dependencies.

Chapter 2: Setting up UAV in Ad-Hoc mode

For the system of multiple UAVs and Arduino Yun to be able to function as one system you need UAVs to make a network of their own. In our system UAVs will establish an ad-hoc network and Arduino Yun will connect to that network.

For establishing the ad-hoc network, first power up drone and connect your laptop to drone's wifi. Go to terminal and type:

```
telnet 192.168.1.1
```

You need to edit /data/config.ini file. So give command:

```
vi /data/config.ini
```

Change the ssid_singeplayer name to *AR_Drone_Adhoc*

And change mode to 1 (i.e ad-hoc)

Next change /bin/wifi_setup.sh file

```
vi /bin/wifi_setup.sh
```

Change probe to a unique number (say 200). Do similar changes in all the drones and assign each of them a unique ip by giving them unique probe (like 205, 210, 215 etc)

Next give instruction:*reboot*. Now when drone boot's up it will be in ad-hoc mode.

For setting Arduino Yun in Ad-hoc mode:

Power up Yun and go to linino.local on your browser. Go to wifi option. Click *find all available wifi*. You will find *AR_Drone_Adhoc* broadcasted by drones. Connect to it. Type mesh in mode and select ad-hoc mode in type. Next reboot the Yun by giving instruction reboot. Now when Yun boots up it will be connected to AR_Drone_Adhoc

Code – final flight

After Yun is configured in adhoc mode, connect your laptop to AR_Drone_Adhocwifi. Go to linino.local web-address on browser. If linino IDE doesn't come up, find the ip address of Yun by *ping linino.local* instruction and type that as web-address.

Go to linino.local:2424, open drone project and make a new file named coordinated-fligh.js

Code:

```
var arDrone = require('ar-drone');
var drone =  [
                  arDrone.createClient({ip: '192.168.1.200'}),
                  arDrone.createClient({ip: '192.168.1.205'}),
                  arDrone.createClient({ip: '192.168.1.210'})
];

drones.forEach(function(drone){
   drone.takeoff();

  drone
   .after(3000, function() {
      this.stop();
   })
    .after(2000, function() {
     this.up(0.3);
   })
    .after(3000, function() {
      this.stop();
   })
    .after(2000, function() {
      this.animateLeds('snakeGreenRed', 5,2);
   })
    .after(3000, function() {
      this.stop();
    })
   .after(3000, function() {
     this.left(0.2);
   })
   .after(3000, function() {
      this.stop();
   })
    .after(3000, function() {
     this.animate('thetaDance', 5000);
   })
   .after(3000, function() {
      this.stop();
   })
   .after(3000, function() {
     this.right(0.2);
   })
   .after(3000, function() {
      this.stop();
   })
```

```
    .after(3000, function() {
      this.animate('phiDance', 5000);
    })
     .after(3000, function() {
        this.stop();
    })
     .after(2000, function() {
      this.up(0.3);
    })
    .after(9000, function() {
        this.stop();
    })
    .after(2000, function() {
        this.animate('flipLeft', 2000);
    })
     .after(3000, function() {
        this.stop();
    })
     .after(2000, function() {
      this.up(0.3);
    })
    .after(3000, function() {
        this.stop();
        this.land();
       });
});
```

References:

http://www.linino.org/tutorial-01-control-a-parrot-ar-drone-with-linino/

[1] http://arduino.cc/en/Tutorial/YunPackageManager

[2] http://www.linino.org/forums/topic/ideino-board-connection-failure/