Raspberry Pi + Ar.Drone 2

With some external controls

The components

Raspberry Pi 3 Model B

Wireless Keyboard (from kano kit, https://kano.me)

Raspberry Pi 7in Capacitive Touch Screen

DesignSpark Raspberry Pi LCD Touch Screen Case

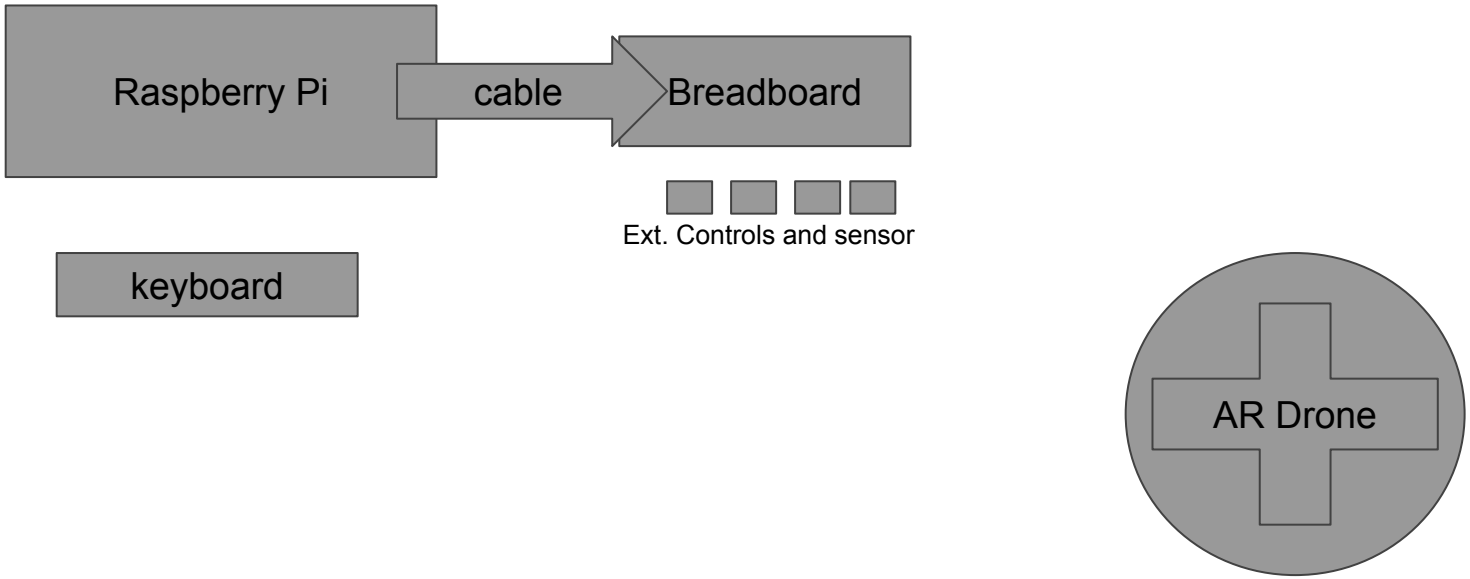
Breadboard, 40 pin GPIO cable and T-Cobbler

Shed load of jumper wires and 3/5-pin anti reverse cables

LCD 1602 Module, Ultrasonic sensor, Joystick (PS2) controller + AD/DA converter

Parrot AR Drone 2.0

Component layout !

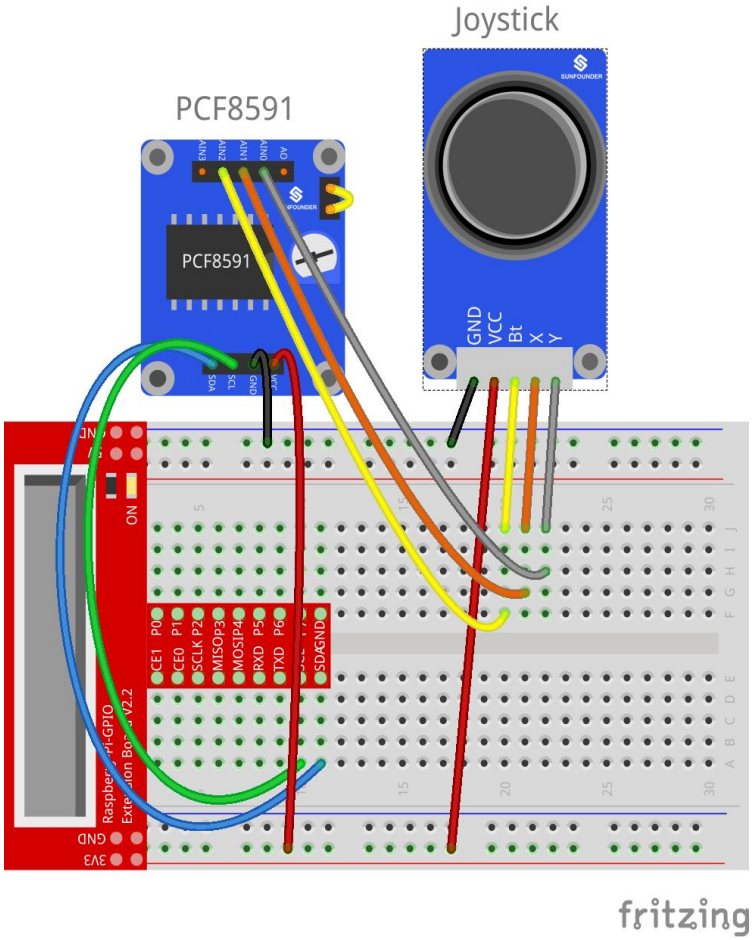


The Controls

---

1. Joystick PS2 Controller

Move up once to take off once drone is taken off **up** will move drone up, pushing down drone goes down etc. Press in and drone Will land

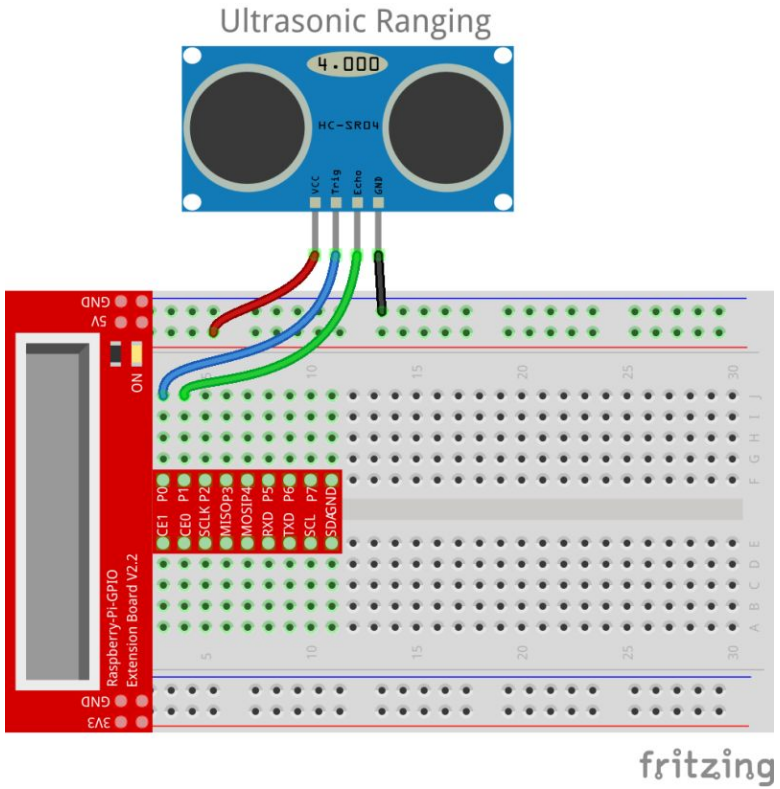


The Controls cont...

---

2. Ultrasonic Sensor

This sensor can read distance between it and an object (your hand). Moving your hand up over it will move the drone up and down will bring the drone down (hopefully)

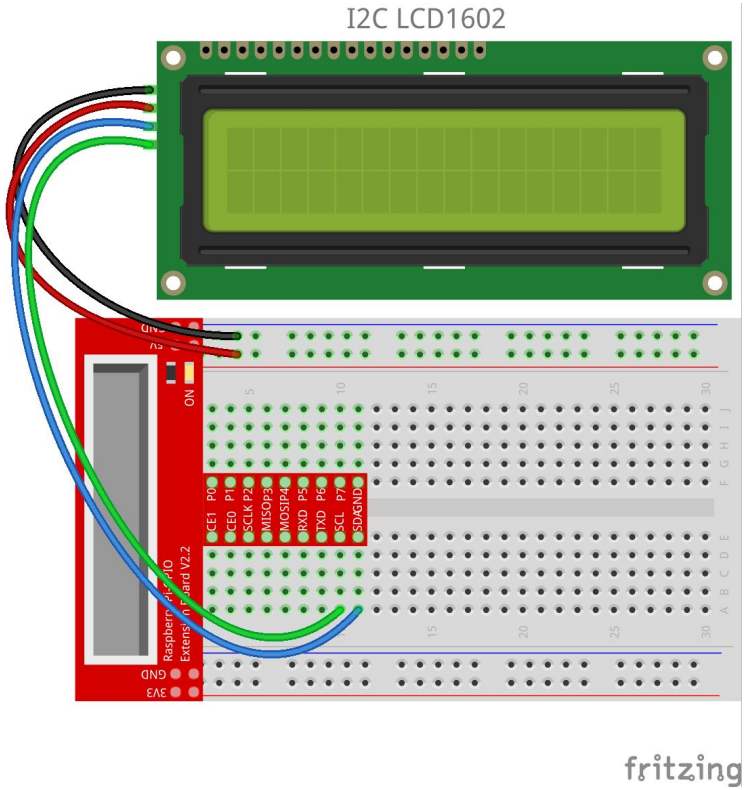


The Controls end...

---

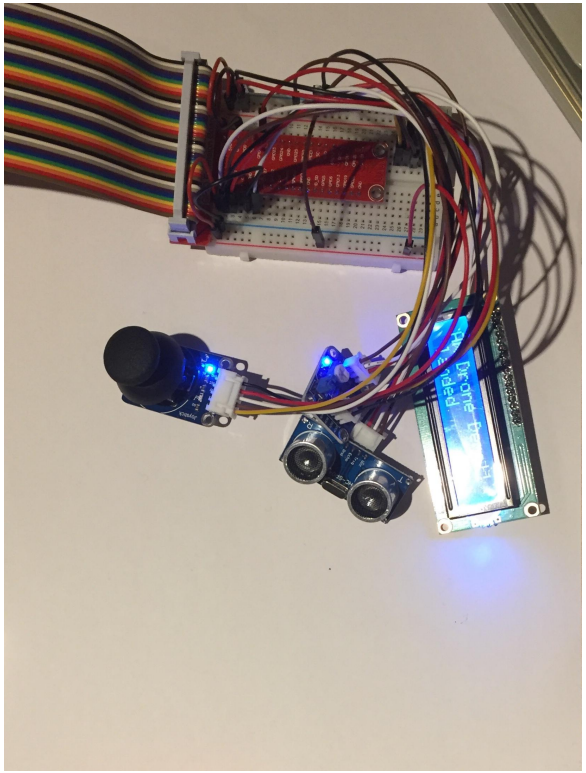
3. LCD1602

This is used to show what is going on, direction of drone, height, battery levels etc etc



All together now...

The demo merges each of the above controls into one system to control the drone. Code was added to output drone direction to the LCD, cripple the drones height and auto land when it goes too high. The sun founder sensor kit comes with a lot more sensors, so hopefully this gives you a flavour of what can be hacked together.



The Code

The demo code is written in Python, I used some of the sample code from the Sun Founder examples and merged with my own to control the drone using Python-ardrone library.

```
while running:
  try:

    d = direction()
    if flying:
      dis = distance()

    if d == 'home':
      if flying:
        drone.hover()
      status = d
    elif d == 'pressed':
      flying = False
      running = False
    elif d == 'up':
      if flying == False:
        flying = True
        drone.takeoff()

      if flying:
        drone.move_up()
      dis = 0
    elif d == 'down':
      drone.move_down()
      dis = 0
    elif d == 'left':
      drone.move_left()
    elif d == 'right':
      drone.move_right()
    else:
      print drone.navdata.get(0, dict()).get("battery", 0)

    if flying and dis > 0 and dis <= MAX_HAND_DISTANCE and current_dist > 0 and current_dist <= MAX_HAND_DISTANCE:
      if dis <= MAX_HAND_DISTANCE:
        if dis > current_dist:
          alt = drone.navdata.get(0, dict()).get("altitude", 0)
          print 'drone up', dis, current_dist, alt
          drone.move_up()
          showDirect('up %d %d' %(dis, alt))
        elif dis < current_dist:
          alt = drone.navdata.get(0, dict()).get("altitude", 0)
          print 'drone down', dis, current_dist, alt
          drone.move_down()
          showDirect('down %d %d' %(dis, alt))

    if status != '' and status != d:
      showDirect(d)
    status = d
    current_dist = dis

    if flying:
      alt = drone.navdata.get(0, dict()).get("altitude", 0)
      vx = drone.navdata.get(0, dict()).get("vx", 0)
      vy = drone.navdata.get(0, dict()).get("vy", 0)

    if alt > MAX_HEIGHT:
      drone.land()
      flying = False
      running = False

    if drone.navdata.get(0, dict()).get("battery", 0) < 20:
      drone.land()
      print "Battery too low landin, Please change %d" % (battery_level)
      running = False
    time.sleep(0.2)
  except:
    pass

print "Shutting down...",
print "Battery ", drone.navdata.get(0, dict()).get("battery", 0)
drone.land()
drone.halt()
GPIO.cleanup()
showDirect('landed')
print "Ok."
```

If we just had a bit more time :)

We got a lot done, more than I had expected in fact, but there are some nice to haves that could have been added.

Controlling the drone with your voice, face recognition and more using the drones camera, predefined flight path, WiFi demo of hacking into the drone, cripple the space the drone can operate in, connect more sensors (breadboard is pretty packed however) etc etc etc….

Resources

Sunfounder sensor kit https://www.sunfounder.com/starterkit/arduino/sensor-kit-v2-0.html

Raspberry Pi http://ie.rs-online.com/web/p/processor-microcontroller-development-kits/1259525/?sra=pmpn

ArDrone https://www.parrot.com/fr/drones/parrot-ardrone-20-elite-edition#parrot-ardrone-20-elite-edition

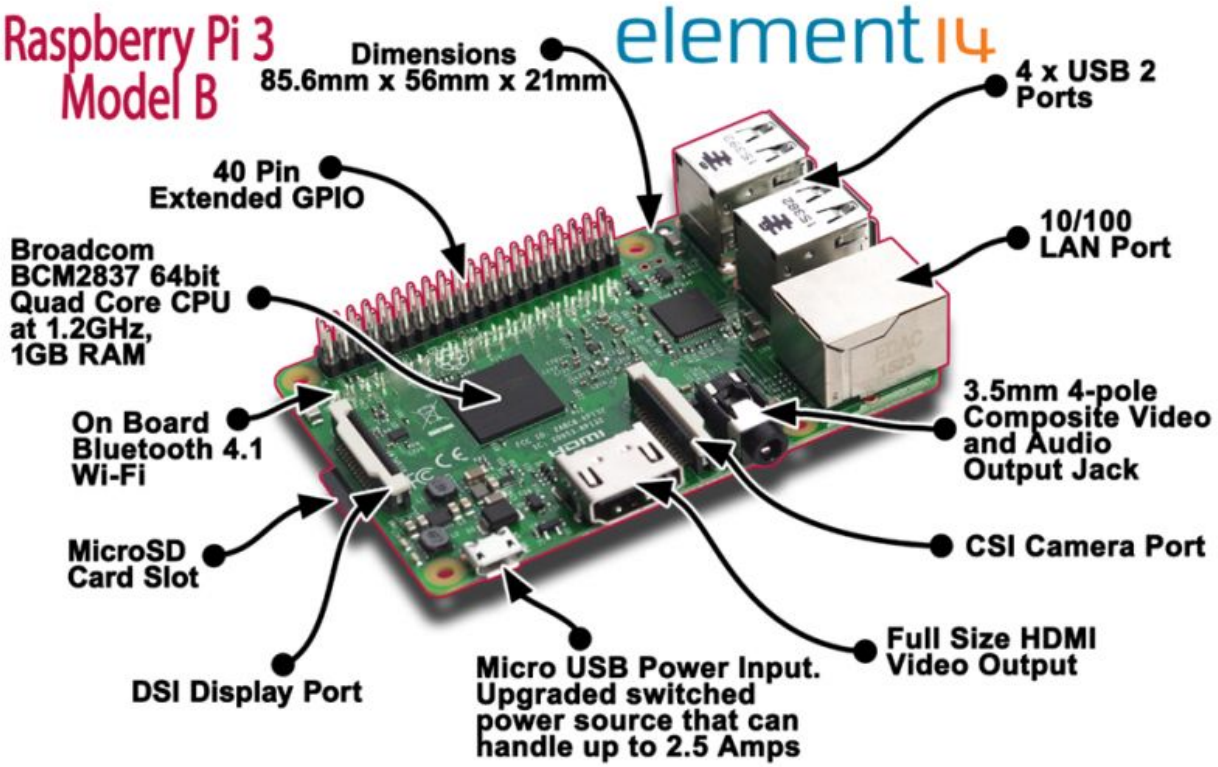
Ar Drone lib. https://github.com/venthur/python-ardrone

I will put the full demo up on Github if anyone wants to try it for themselves or come chat with us during the day for more info. Check the coderdojo website in the coming days/ weeks :)

Sunfounder Sensor Kit

1 Dual-Color LED
2 RGB LED Module
3 7-Color Auto-flash LED
4 Relay Module
5 Laser Emitter Module
6 Button Module
7 Tilt-Switch Module
9 IR Receiver Module
10 Buzzer Module
11 Reed Switch
12 Photo-interrupter
13 PCF8591
14 Rain Detection Module
15 Joystick PS2
16 Potentiometer Module
17 Hall Sensor
18 Thermistor Module
19 Sound Sensor
20 Photoresistor Module
21 Flame Sensor
22 Gas Sensor
23 IR Remote control
24 Touch Switch
25 Ultrasonic Ranging Module
26 DS18B20 Temperature Sensor
27 Rotary Encoder Module
28 Humiture Sensor
29 IR Obstacle Avoidance Module
30 I2C LCD1602
31 Barometer
32 MPU6050 Gyro Acceleration Sensor
33 RTC DS1302
34 Tracking Sensor
35 Intelligent Temperature Measurement System

Raspberry Pi 3 Model B



Thank You