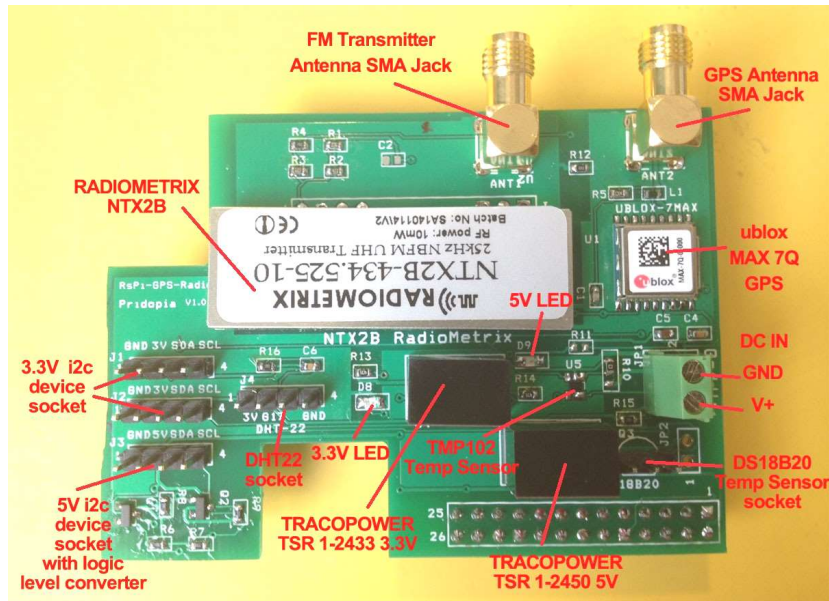


P-HAB (High-Altitude Balloon) 5V User Manual



DC input use DC 9V ~12V

1. Enable serial Port

Raspberry Pi Serial Port Usage

The serial port on the Raspberry Pi is configured as default for console input/output. This allows you to login and interact with the Raspberry Pi via the serial port but you cannot use the serial port with your programs. To use the serial port with other programs and hardware such as modems, arduino boards etc you need to disable the console login.

To Disable Serial Port Login

You need to edit two files in order to use the serial port with your own programs.

When the Raspberry Pi boots, the bootup information is sent to the serial port. You can disable this by editing the `/boot/cmdline.txt` file

The contents of the file look like this

```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200
kgdboc=ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4
elevator=deadline rootwait
Remove all references to ttyAMA0 so the file looks like this:
dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4
elevator=deadline rootwait
Save the file to save your changes.
The second file to edit is /etc/inittab
Edit using: sudo nano /etc/inittab
The /etc/inittab file has the command which enables the login prompt which
needs to be disabled.
Near the end of the file will be a line similar to this:
respawn:/sbin/getty -L ttyAMA0 115200 vt100
Disable this line by adding a # character to the beginning.
#respawn:/sbin/getty -L ttyAMA0 115200 vt100
Save the file.
You should then reboot your raspberry pi with the following command
```

sudo shutdown -r now

Serial port setting detail in our web site

<http://www.pridopia.co.uk/rs-pi-set-serial.html>

1. Make sure your I2C driver are enable

To enable it all you need to do is comment out a line by putting # in front

```
sudo nano /etc/modprobe.d/raspi-blacklist.conf
```

2. Add i2c-dev in /etc/modules by use
 sudo nano /etc/modules

```
192.168.0.13 - PuTTY
GNU nano 2.2.6      File: /etc/modules      Modified
## /etc/modules: kernel modules to load at boot time.
##
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

snd-bcm2835
spi-bcm2708
i2c-bcm2708
i2c-dev
rtc-ds1307
```

3. Next install the python-smbus python module:

sudo apt-get install python-smbus

sudo apt-get install i2c-tools

Now you are ready to use the i2c with python.

If you already install I2c driver ,

i2cdetect -y 1 - for Raspberry Pi V2 Board

```
COM21 - PuTTY
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Type 'startx' to launch a graphical session

i2vroot@raspberrypi:~# i2cdetect -y 1
 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- 21 22 -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
root@raspberrypi:~#
```

i2cdetect -y 0 - for Raspberry Pi V1 Board

```
COM12 - PuTTY
raspberrypi login: root
Password:
Last login: Fri Sep 21 17:20:25 GMT 2012 on tty1
Linux raspberrypi 3.1.9adafruit+ #10 PREEMPT Thu Aug 30 20:07:

The programs included with the Debian GNU/Linux system are free
the exact distribution terms for each program are described in
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Type 'startx' to launch a graphical session

i2root@raspberrypi:~# i2cdetect -y 0
 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: 20 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- 48 49 -- -- -- -- -- --
50: 50 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: 60 -- -- -- -- -- -- -- -- 68 -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
root@raspberrypi:~#
root@raspberrypi:~#
```

Download HAB software

wget <http://pridopia.co.uk/pi-pgm/Installer-hab.tar>

tar xf Installer-hab.tar

python Install.py ---- install necessary software for HAB

python Install-Edu.py -- install eduhab auto-login-autorun

The eduhab.py send GPS location, tmp102, DS18B20, BMP085, DHT22 information through FM UHF Transmitter, the eduhab.py have 3 commands

eduhab.py -s short data stream mode send GPS location and BMP085 Temp only ,

but all logs save to SD card

eduhab.py -p send data also take photos every 30sec

eduhab.py -c prid change the data stream "twick" to "prid"

- 18B20 1-wire Temp sensor have problem when working together with camera, if you don't use camera you can use 18b20 Temp sensor.
- DHT22 use GPIO 17 , we change to i2c base
HTU21D Digital Temp & Humidity Sensor

If you don't have "BMP085", & "HTU21D" sensors setting to False in eduhab.py

```

eduhab.py - WordPad
File Edit View Insert Format Help

##-----##
TESTING = False ## If you want to test the Cut Off, Enable this (True).
##-----##

GPIO.setmode(11)
GPIO.setwarnings(False)
GPIO.setup(CutOff, GPIO.OUT)

Devices = {
    "BMP085": True,
    "TMP102": True,
    "HTU21D": True,
    "DS18B20": False
}

```

Test & Monitor program

Win xp [sdr-inatll.zip](#) [dl-fldigi-dl3.1](#) [VBCABLEDriver_Pack42b](#)

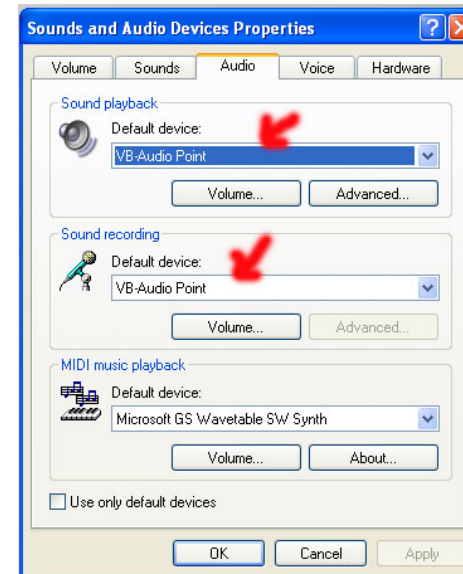
Install VBCable Driver



Install VBCable Audio Driver

Software will auto detect your OS -- WinXP

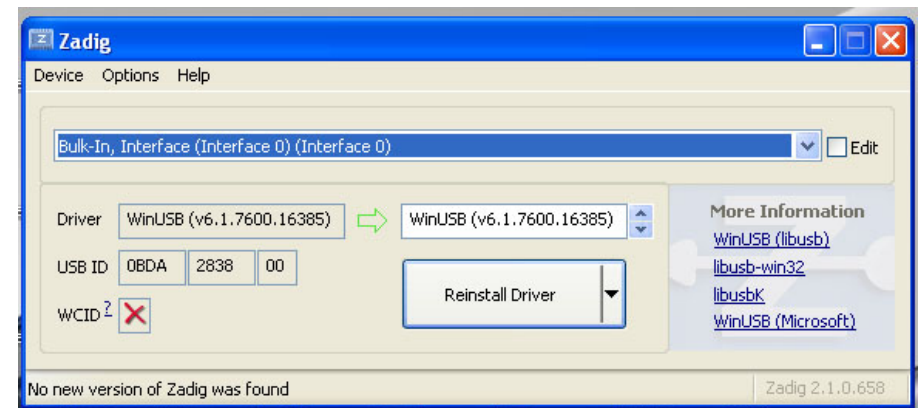
Click "Install Driver"



In Setting-> control panel -> Sounds and Audio Devices

Audio -> Sound playback -> VB-Audio Point

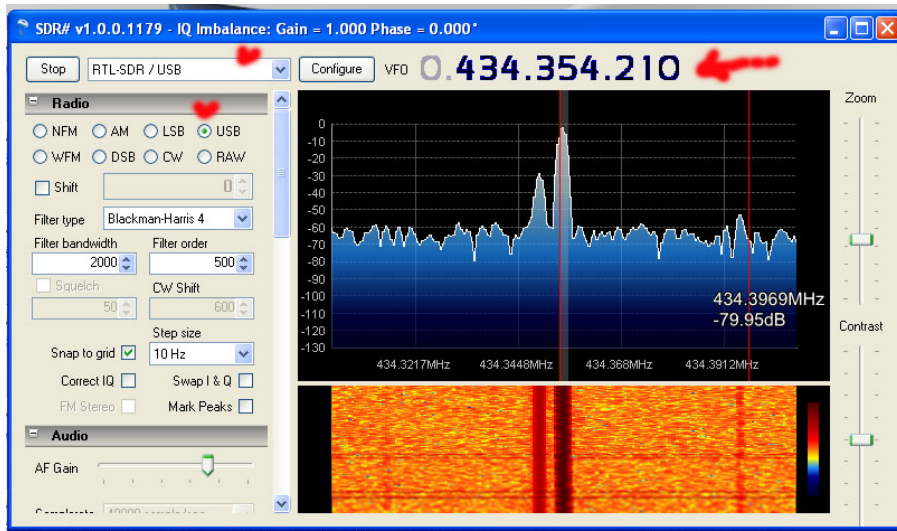
Audio -> Sound recording -> VB-Audio Point



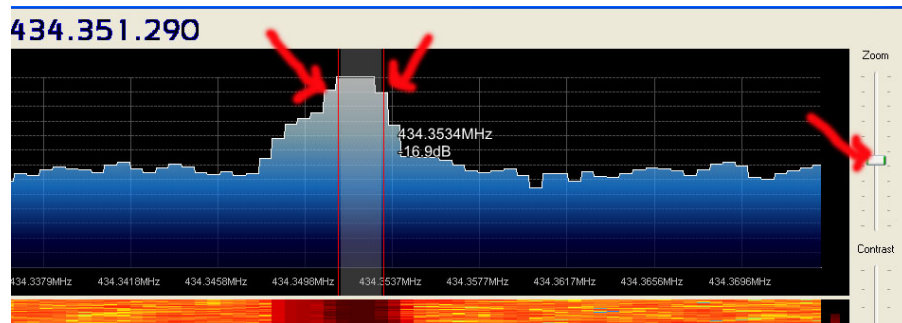
In options choose List all Devices ->

Choose Bulk-In, Interface (Interface0) (Interface0)

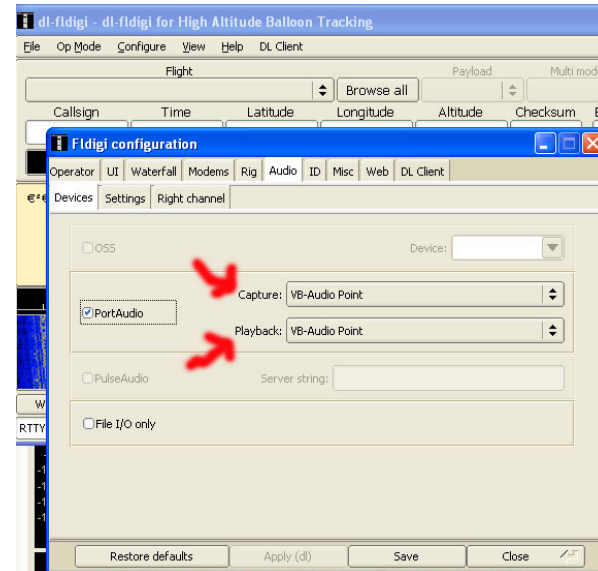
Click "Install Driver"



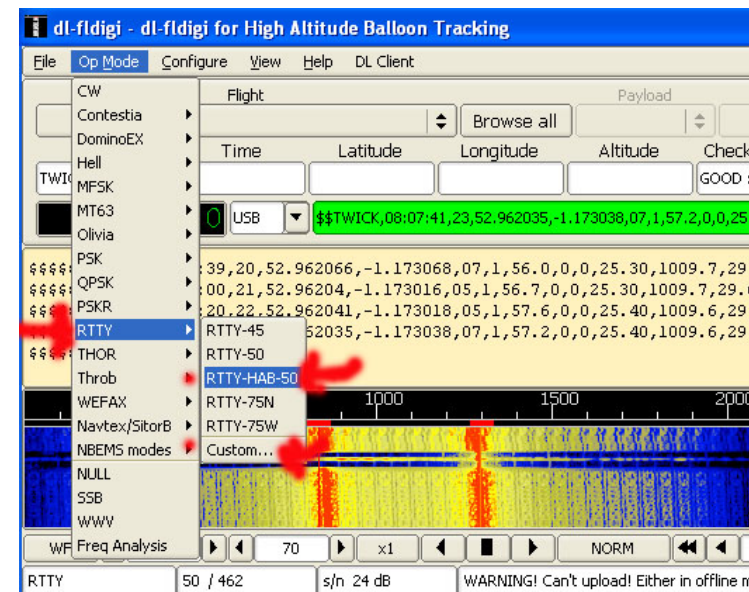
Choose RTL-SDR /USB → radio choose USB -> Click “Play”



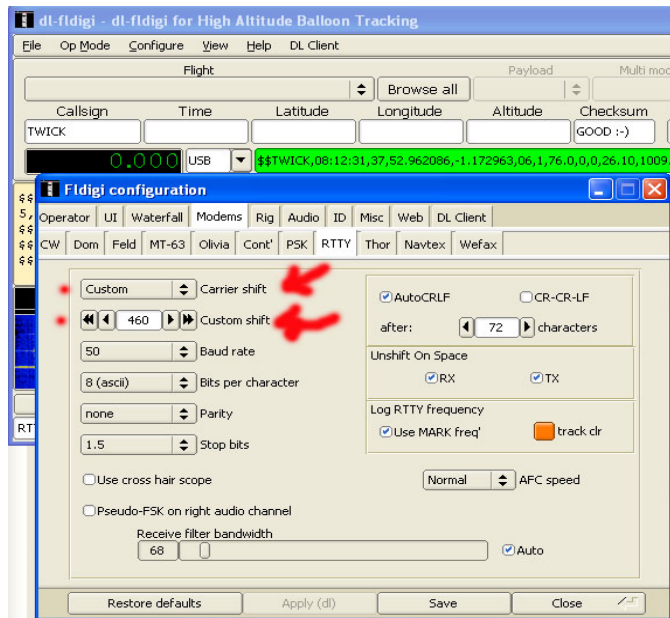
Adjust the wide and Contrast



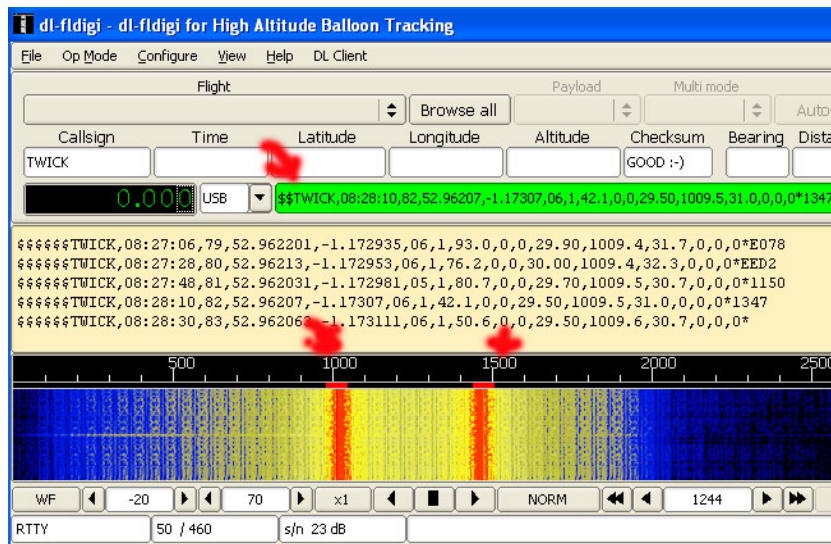
In configure -> Sound card → PortAudio
Capture : choose VB-Audio Point
Playback: choose VB-audio Point



Choose RTTY -> RTTY HAB-50-> & Custom

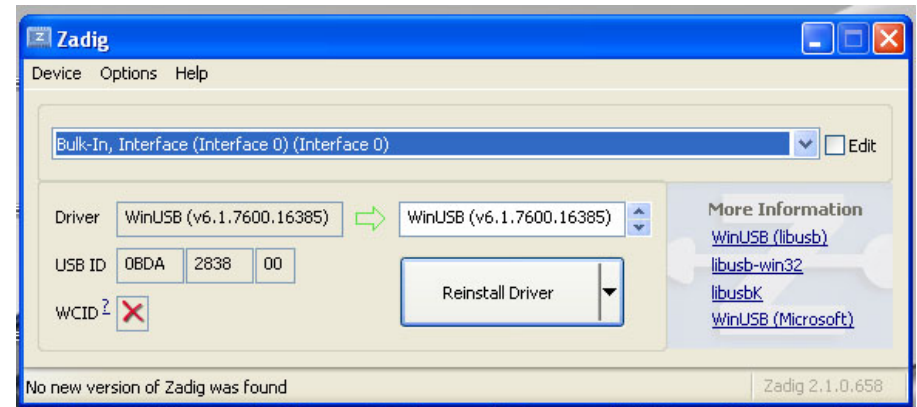


Change shift to about 455 ~ 460



Win7 Setting

Install Zadig



In options choose List all Devices ->

Choose Bulk-In, Interface (Interface0) (Interface0)

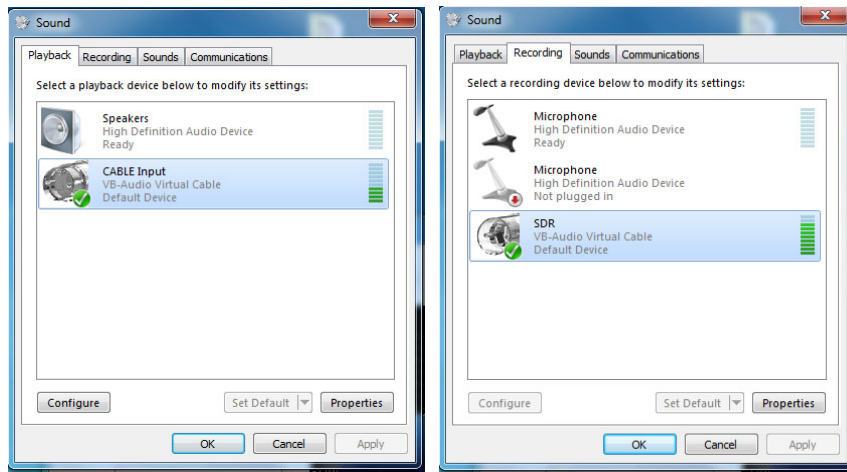
Click "Install Driver"

Install VBCable Audio Driver

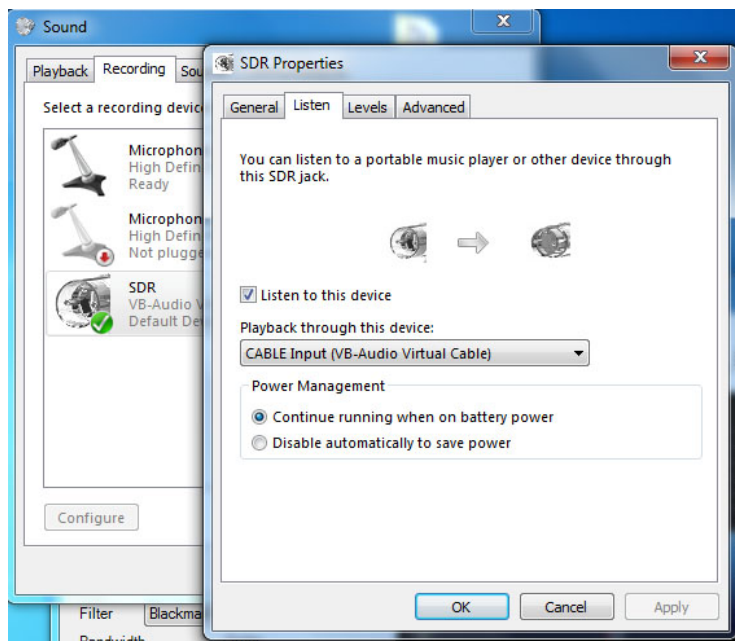
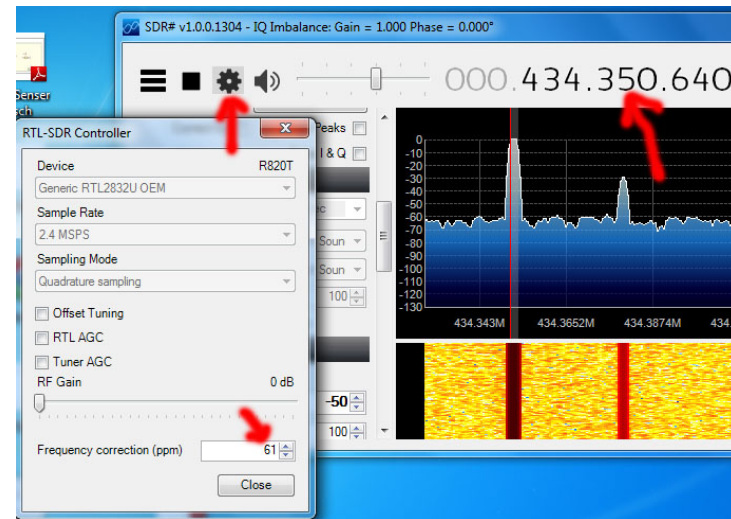
Software will auto detect your OS -- Win7

Click "Install Driver"



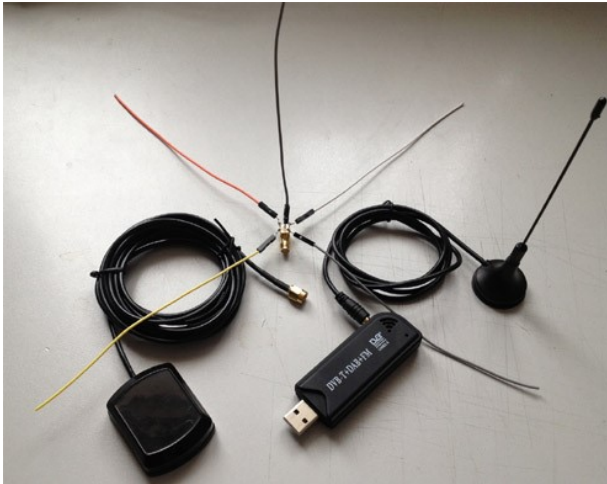


Setting Cable Input as default & SDR as default in “Recording”



In Listen choose playback through this device as “CABLE Input (VB-Audio Virtual Cable)”

Adjust Frequency correction(ppm) as the label in NTX2B-FA 434.xxx



GPS antenna & FM antenna &

USB SDR Encoder (RTL2832U-based DVB-T devices (RTL-SDR)

the eduhab.py send GPS location, tmp102, DS18B20, BMP085, DHT22 information through FM UHF Transmitter, the eduhab.py have 3 commands

eduhap.py -s short data stream mode send GPS location and BMP085 Temp only , but all logs save to SD card

eduhab.py -p send data also take photos every 30sec

eduhab.py -c prid change the data stream "twick" to "prid"

```

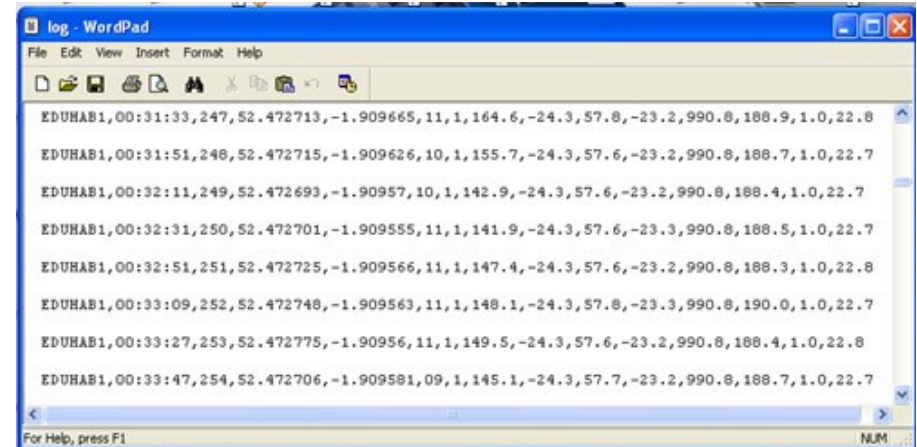
192.168.0.28 - PuTTY
root@raspberrypi:~# cd ..
root@raspberrypi:~# cd /home/pi
root@raspberrypi:/home/pi# cd edu-hab
root@raspberrypi:/home/pi/edu-hab# sudo python eduhab.py
Acquired this data string from serial: $GPGGA,091711.00,5257.72105,N,00110.38025
,W,2,11,0.86,66.8,M,47.2,M,,0000*7D

181
98
5
1
2
0
6
36
50
91
$GPGGA,091711.00,5257.72105,N,00110.38025,W,2,11,0.86,66.8,M,47.2,M,,0000*7D

Thu Mar 27 09:17:00 GMT 2014
now sending the following: $$TWICK,09:17:11,0,52.962016,-1.173003,11,1,66.8,21.7
,29.9,21.2,1008.0,44.1,24.0,21.9*D6F2

Acquired this data string from serial: $GPGGA,091723.00,5257.72169,N,00110.38014
,W,2,11,0.86,66.3,M,47.2,M,,0000*7F

```



data format

EDUHAB1,00:33:27,253,52.472775,-1.90956,11,1,149.5,-24.3,57.6,-
23.2,990.8,188.4,1.0,22.8

EDUHAB1 callsign

00:33:27 TIME

253 LOG COUNTER

52.472775,-1.90956 latitude, longitude GPS location

11 satellites --- ublox Max 7Q GPS module

1 flightmode --- ublox Max 7Q GPS module

149.5 altitude --- ublox Max 7Q GPS module

-24.3 -24.3 -24.3 -24.3 -24.3 temp DHT22 -

Temp External GPIO17

57.6 humidity DHT22 - Humidity GPIO 17

-23.2 temp2 BMP085 - Temp External i2c address 77dress 77

990.8 pressures2 BMP085 - Pressure i2c address 77

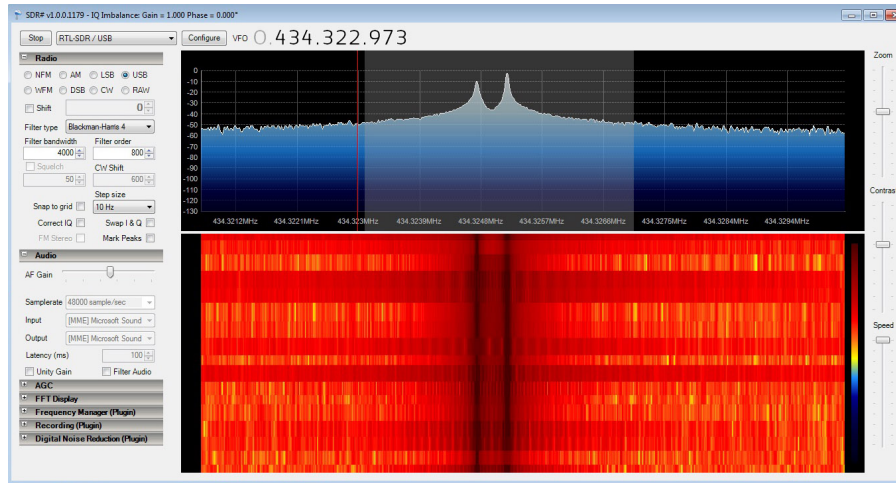
188.4 alt2 BMP085 - altitude i2c address 77

1.0 temp3 TMP102 - on board Temp i2c address 49

22.8 temp4 DS18B20 - Temp External 1-Wire GPIO 4

Tracking using a SDR software-defined radio (SDR)

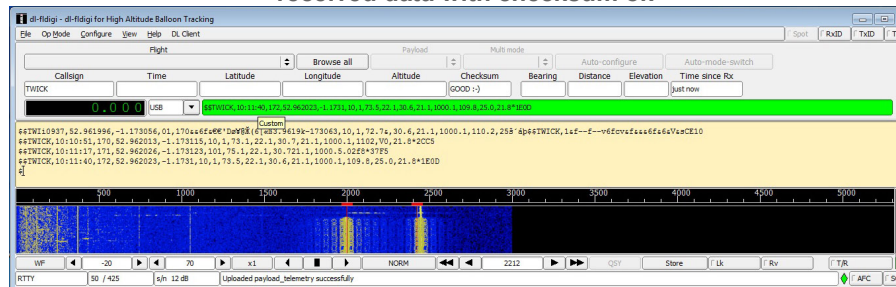
Run sdrsharp.exe and you will see the following window



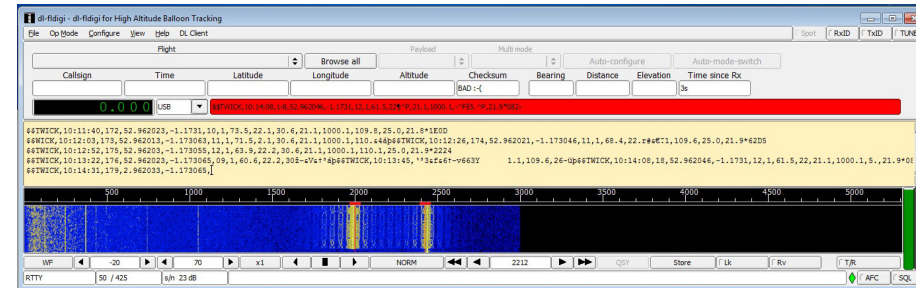
open DL-Fldigi, click Configure → Sound Card Click Capture → SDR(Virtual Audio Cable)

You should now be able to click on the payload as normal and decode :

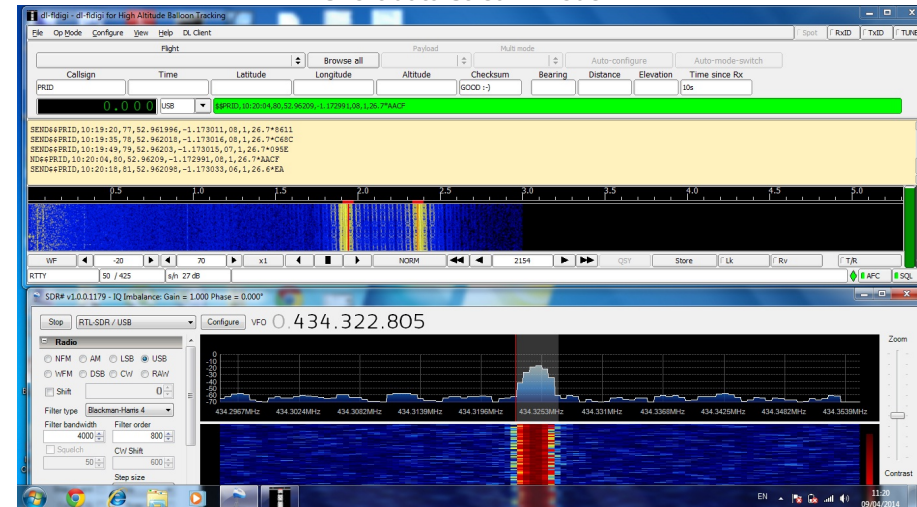
received data with checksum ok



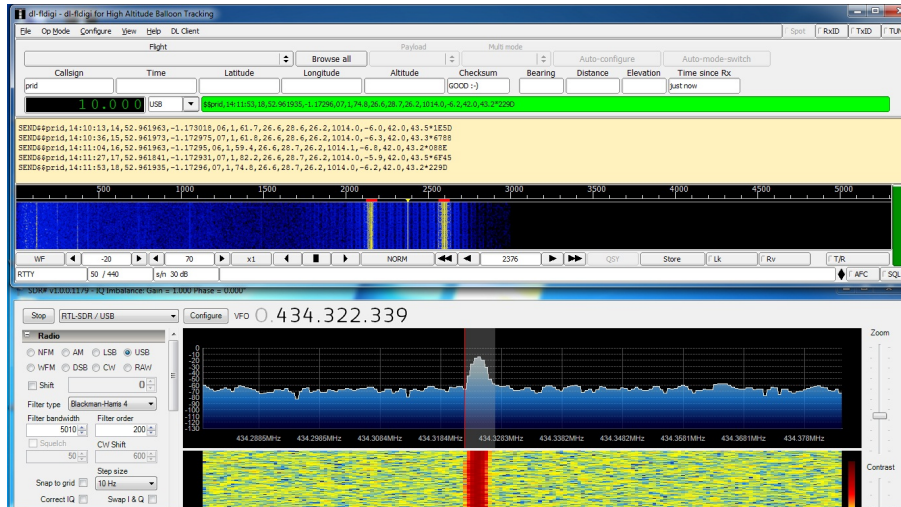
received data with checksum error



short data stream mode



full data stream mode



Track balloon online

