**HOW TO SET UP THE PARTICLE DETECTOR PROJECT**

**Giorgia Mazzini**

**Matilde Mazzini**

**University of Hawaii at Manoa, 2017**

**Part List:** Here are some the things you need to build your Particle Detector!

|  |  |  |  |
| --- | --- | --- | --- |
| **Description** | **Unit Cost** | **Vendor** | **Link** |
| **Micro SD** card at least 8 Gb class 10 | $12,99 | Best Buy | <http://www.bestbuy.com/site/sandisk-ultra-8gb-microsdhc-uhs-i-class-10-memory-card-gray-red/5577759.p?skuId=5577759> |
| **Adapter** Micro SD to SD | $6,99 | Amazon | <https://www.amazon.com/Adapter-Standard-Connector-Smartphones-Function/dp/B01BXSKPES/ref=sr_1_4?s=pc&ie=UTF8&qid=1489542940&sr=1-4&keywords=sd+adapter> |
| **Raspberry Pi 3 - Model B** - ARMv8 with 1G RAM AND built-in WiFi | $39,95 | Adafruit | <https://www.adafruit.com/product/3055> |
| **5V 2.4A Switching Power Supply** w/ 6' MicroUSB Cable | $7,50 | Adafruit | <https://www.adafruit.com/product/1995> |
| **USB Logitech c270 Webcam** | $19,99 | Best Buy | <http://www.bestbuy.com/site/logitech-hd-webcam-c270-black/9928354.p?skuId=9928354> |
| Plugable USB 2.0 4-Port High Speed **Charging Hub** (power supply+ cable hub to USB are going to be inside) | $16,95 | Amazon | <https://www.amazon.com/Plugable-Charging-Adapter-Support-Android/dp/B005P2BY5I/ref=cm_cr_arp_d_product_top?ie=UTF8> |
| **Adafruit Parts Pal**  Where you can find:  -Green and Red LED 5mm  -Tactile Button switch (6mm)  -Breadboard  -560 ohm 5% axial resistors (x2)  - Male/Male Jumper Wires 6’’ (x6) | $19,95 | Adafruit | <https://www.adafruit.com/products/2975> |
| **Adafruit Assembled Pi Cobbler Plus with GPIO Ribbon Cable** for Raspberry Pi Model A+/B+/Pi 2/Pi 3 - (40 pins) | $6,95 | Adafruit | <https://www.adafruit.com/products/2029> |
| **GPS Module** for Raspberry Pi 3B with USB Port | $27,71 | Gearbest | <http://www.gearbest.com/raspberry-pi/pp_436481.html?currency=USD&vip=988995&gclid=Cj0KEQjw76jGBRDm1K-X_LnrmuEBEiQA8RXYZ3mkDETFqQ7fMYozk2GGndECKIIpUreg0Eg1IK4xdCwaAt_k8P8HAQ> |

You will also need :

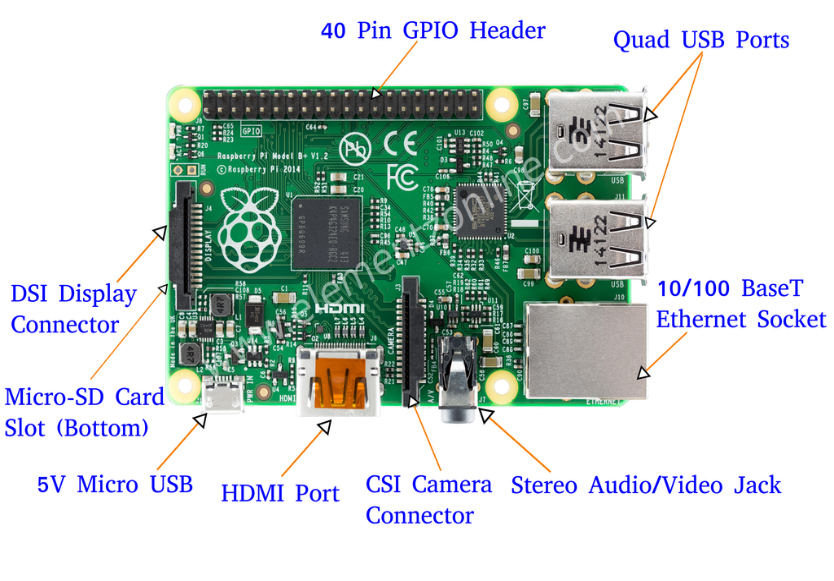
Monitor + cable: Monitor port to HDMI (Raspberry Pi port)

USB Keyboard and mouse

But you can find those wherever you want.

**THE RASPBERRY PI:**

The **Raspberry Pi** is a small single-board computer developed to promote the teaching of basic **computer science** in schools.



http://beliketechies.blogspot.com/

**Installation Guide:** You can follow these steps in order to have a working Raspberry Pi.

If an error occurs, the steps that you need to follow are:

* Double-check the spelling.
* Check that your Wi-Fi is on and you are able to surf the net!
* Try again.

**Setting up SD:** First thing you need to do is to prepare your SD card installing on it the OS(Operating System).

Insert the SD in your computer (it is a Micro SD, maybe your computer needs an adaptor!).

IF YOUR SD IS NEW/EMPTY/ALREADY FORMATTED YOU CAN START FROM THE INSTALLATION OF THE OPERATING SYSTEM. (Just jump the next step!)

ONLY IF YOUR SD IS NOT NEW/EMPTY/ALREADY FORMATTED YOU WILL NEED TO:

Download and Open “SD formatter” from https://www.sdcard.org/downloads/index.html

In Options choose: Full (erase) and Size adjustment on.

In “drive” be sure to choose the right device (you can always investigate before formatting something you don’t want to ;) ) that has to be formatted.

Format SD.

IF YOUR SD IS NEW/EMPTY/ALREADY FORMATTED YOU CAN START FROM HERE.

**Install Operating System:** Now you need to install the Raspberry OS on the SD that you just formatted.

From <https://www.raspberrypi.org/downloads/raspbian/>

Download Raspbian Jessie (NOT lite) 1.5 Gb.

Unzip it. Download and open Win 32 Disk Imager from https://sourceforge.net/projects/win32diskimager/

Load Raspbian Jessie image. Choose the right device where to save the OS (the SD card). Write - Yes.

Now your SD with the Raspberry Pi OS is ready! Insert it in your Raspberry Pi.

(Sometimes the installation of Raspbian will create a strange partition in your SD but we will tell you how to fix this problem few steps later!)

**Setting up the RSB:**

Connect Raspberry to Monitor+Keyboard+Mouse. Connect it to power and wait for it to switch ON.

It can ask for a **Password: raspberry (login: pi)** IMPORTANT: When you type your password you will not see anything!!Just type and press Enter.

…Here is your Raspberry Pi: before starting to configure it you can take a look to all its features like installed programs or games!!

We are going to say to Raspberry what to do using LXTerminal: it is a prompt where you insert commands to the Raspberry Pi.

Remember that it is sensitive to capital letters and spaces, type all commands with care ☺

Open LXTerminal and type: sudo raspi-config

From here you can:

Change your password (do it because ‘raspberry’ is common to every Raspberry!!)IMPORTANT: When you type your password you will not see anything!!Just type and press Enter.

Advanced options. Expand file system. Reboot. (There we are!! You have now fixed the partition problem we told you about, that shows up during the installation of OS. This command will let you use the whole SD card.)

**Configuring Wifi**:

It should work by itself (eventually just insert Wi-Fi password as on a normal computer or choose the right connection from the Wi-Fi icon in the top right).

You can change it if needed: Language/Country/Timezone/Keyboard just navigating here :

Start Panel. Preferences. Raspberry Pi Configuration.

Before moving on we suggest to update and upgrade your Raspberry. (We also suggest you to do it once or twice a week).

LXTerminal: sudo apt-get update

LXTerminal: sudo apt-get upgrade (can take up to 3 hours)

(you need to be connected to Wi-Fi in order to do this 2 steps)

Always reboot Raspberry after upgrading.

**Setting up the Webcam:**

Connect the USB Hub to Power and to Raspberry Pi.

Connect the USB cable of the Webcam to the USB Hub.

Enable Serial from the Start panel (the symbol is a raspberry ;) ) of Raspberry. Preferences. Raspberry Pi Configuration. Interfaces.

To check if it works install SimpleCV and do the following steps.

Install **SimpleCV** Library

LXTerminal: sudo apt-get install ipython python-opencv python-scipy python-numpy python-setuptools

yes (about 10 min)

LXTerminal: sudo pip install https://github.com/sightmachine/SimpleCV/zipball/master

LXTerminal: sudo pip install pyparsing svgwrite

LXTerminal: sudo aptitude install lsof

yes

Test if the Camera and SimpleCV work:

LXTerminal: simplecv //(open a bash of simplecv)

LXTerminal: cam = Camera()

LXTerminal: image\_name =cam.getImage()

LXTerminal: image\_name.show()

LXTerminal: image\_name.save(“image\_name.jpg”)

[NOTE: image\_name is something that you can modify as you want!]

**Hacking Camera**

In order to make it sensitive to ionizing radiation, the webcam should be protected from visible light. You can make the changes in a very simple and fast way. (The camera should be disconnected.)

First, the front cover must be removed (you can use a screwdriver), then remove also the underlying base by removing the three small screws.

**GLUE**

Sometimes in the back of the lenses there is a tiny (but very strong) piece of glue. You need to remove it using something very small like a needle. We did it with a hairpin!!

Gently unscrew counterclockwise the camera lens. Now you can finally see the (super tiny) CMOS sensor!

To block ambient light from reaching the CMOS sensor it is necessary to shield it with an

aluminium sheet (I know doing it is very annoying..) and then add a piece of electrical black tape.

**LED**

We also removed the LED on the left side of the camera in order to avoid any kind of noise.

You can now reconnect your webcam to your Raspberry Pi.

Now your webcam is ready to detect particles!!!

(You can always go back to the original setup if you just need to take nice pictures!)

**Setting up the GPS:** In order to know where our experiments are done we need something to track our position…!

Connect the USB cable to USB Hub. Wait for the GPS led to blink (it means it has fixed on the satellites).

To check if it works insert the following command:

LXTerminal: stty –F/dev/ttyUSB0 9600 cs8 –parenb

(this command will set the baud rate options for the gps to 9600.)

LXTerminal: cat /dev/ttyUSB0

(if everything is ok it will print the list of GPS information –NMEA sentences- until you stop it.)

To stop the stream of information press CTRL+C.

**Install Exiftool:** Our code uses this tool to modify the Exif data of the Images.

LXTerminal: sudo apt-get install libimage-exiftool-perl

Yes

Exiftool is now installed. To test it type:

LXTerminal: exiftool image\_name.jpg

You can see the list of the EXIF tags of the image.

**Setting Up the Breadboard:**

First of all you need to connect the Ribbon cable to Raspberry, then connect the Cobbler to breadboard in the position you prefer (we choose the top), connect cable to cobbler. Be sure to connect it in the right way so that the GPIO pins numbers in Raspberry Board coincide with the GPIO pins numbers in the cobbler!! (usually there is a white stripe in the ribbon that goes to the 3.3V to indicate which is the right way!)

Connect the other components (when Raspberry is off ).

(Remember to connect each component to GND!)

Brown wire: from GND to the Negative row. (Remember that all holes in this row are connected)

Green Led:

-Green wire from Gpio 17 to the positive of green Led.

-Resistor1 from the negative of Led to GND row.

Red Led:

-Red wire from Gpio 5 to the positive of red Led.

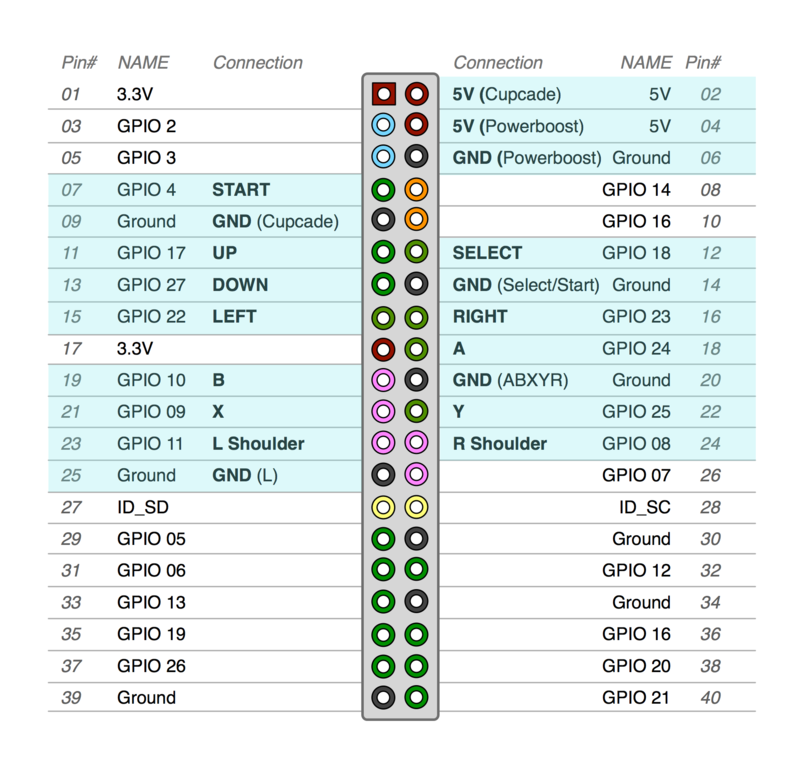
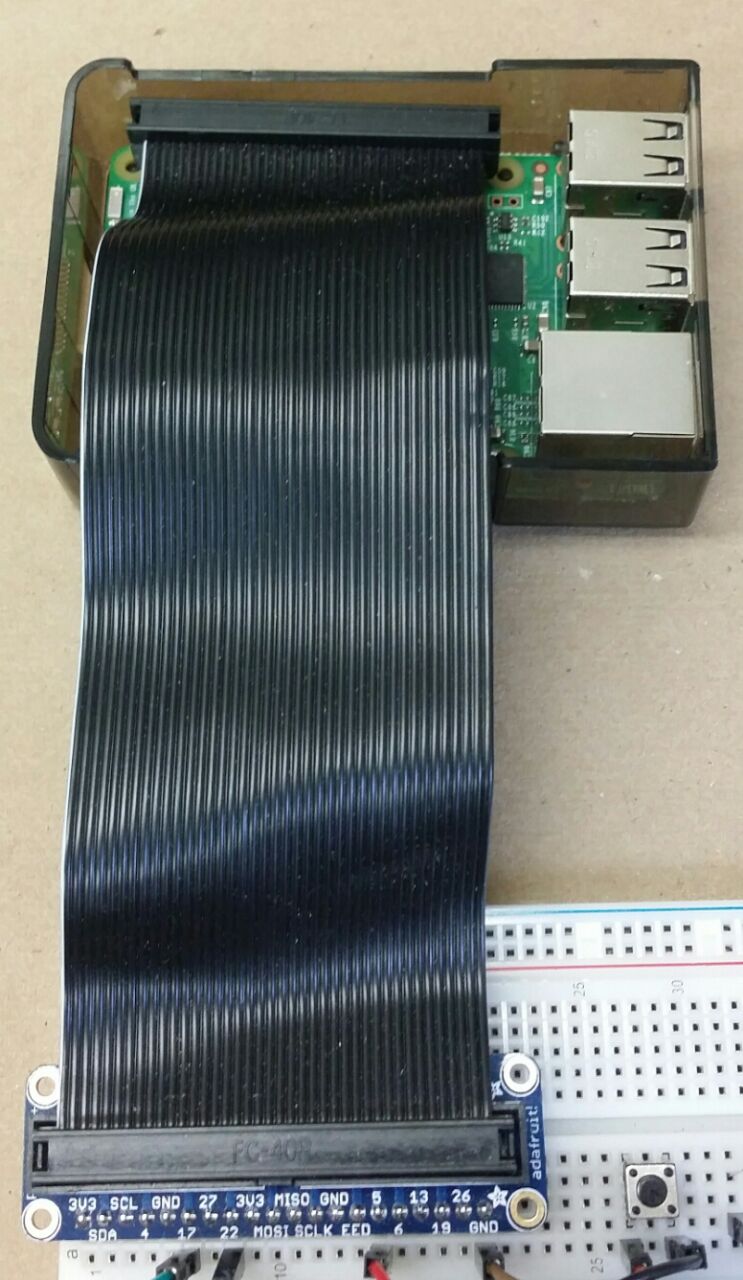
-Resistor2 from the negative of Led to GND row.

Switch Button:

-Blue wire from Gpio 22 to one pin of the button.

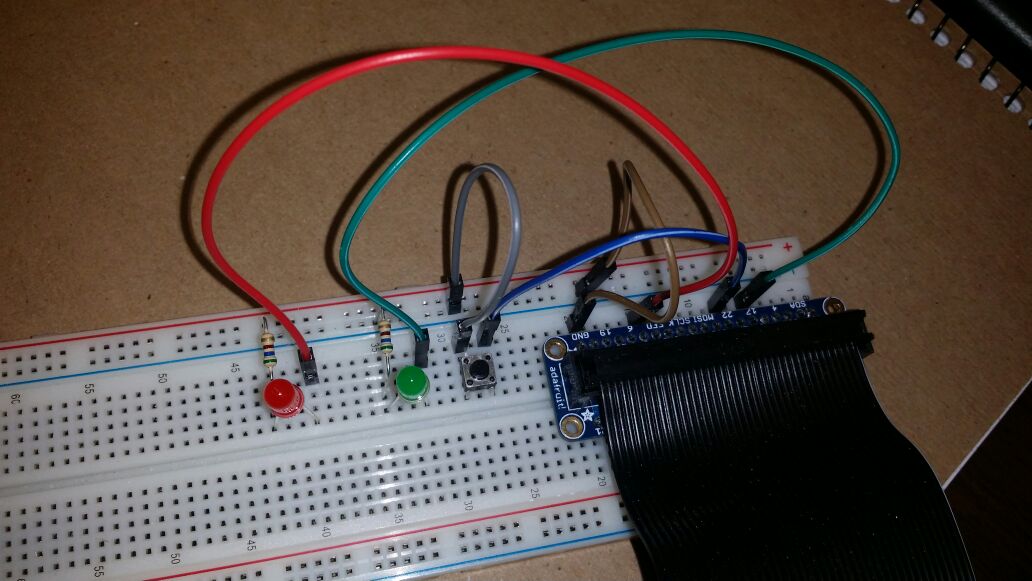
-Grey wire from the second pin of the button to GND row.

(The numbers of GPIO are assigned –for example 22 for the button- but you can always choose to modify whatever you want! If you do, remember to change that port number in the code!!)



**COBBLER**

**RIBBON CABLE**



**GND ROW**

**Using the Code:**

The code we wrote in Python is called **particledetectorGM.py**

Download this code from the website LINK

Put it in a USB pen, insert the pen in Raspberry. Save the code in the main directory /**home/pi/**.

You are going to use the code by typing simple commands in the LXTerminal so before doing it, we suggest you to study some stuff about LXT and feel comfortable with it.

The following commands are our most commonly used commands (like going around the directories of your system), but you can find more in:

https://en.wikibooks.org/wiki/Guide\_to\_Unix/Commands/File\_System\_Utilities

**cd**  to change the **c**urrent **d**irectory of the shell. This current directory will be used by other programs launched from the shell.   
Eg: **cd /home/pi/** to move inside the /pi directory.

**cd ..** to change to parent directory.

**ls**  is a utility for listing the files of the current directory.

**ctrl+c** to forcibly interrupt a program.

**cat filename.txt** to see the content of the file.

**sudo** to give the administrator permission before any command.

**exit** to exit from the terminal.

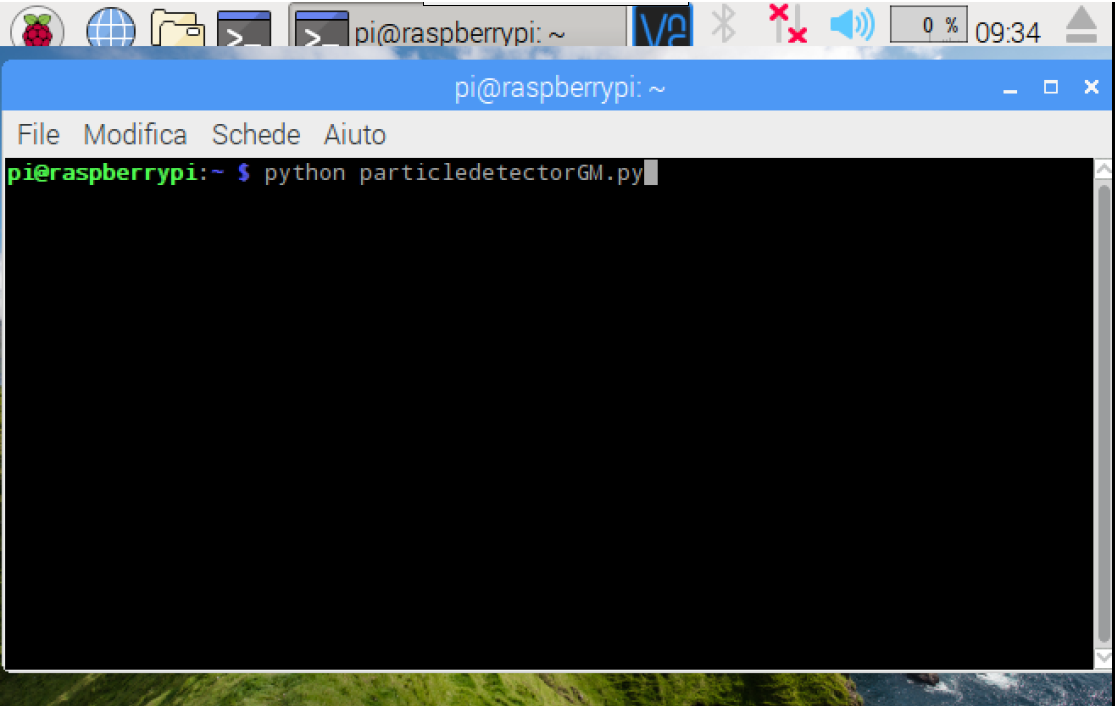
First of all make sure the code **particledetectorGM.py** is in the same directory as the one in which the shell is located. (Otherwise use the cd command to navigate to the proper directory!!)

Usually you have no problems if you leave your python code in pi directory and you can just open the Lxterminal and type:

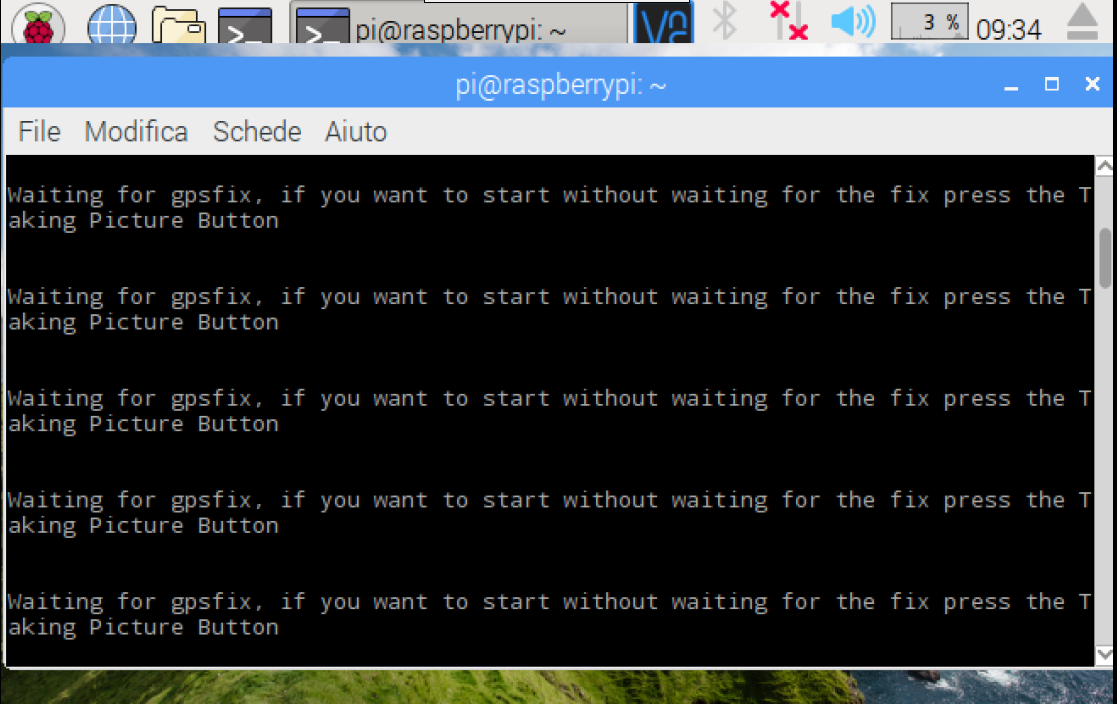
**python particledetectorGM.py**

Press Enter.

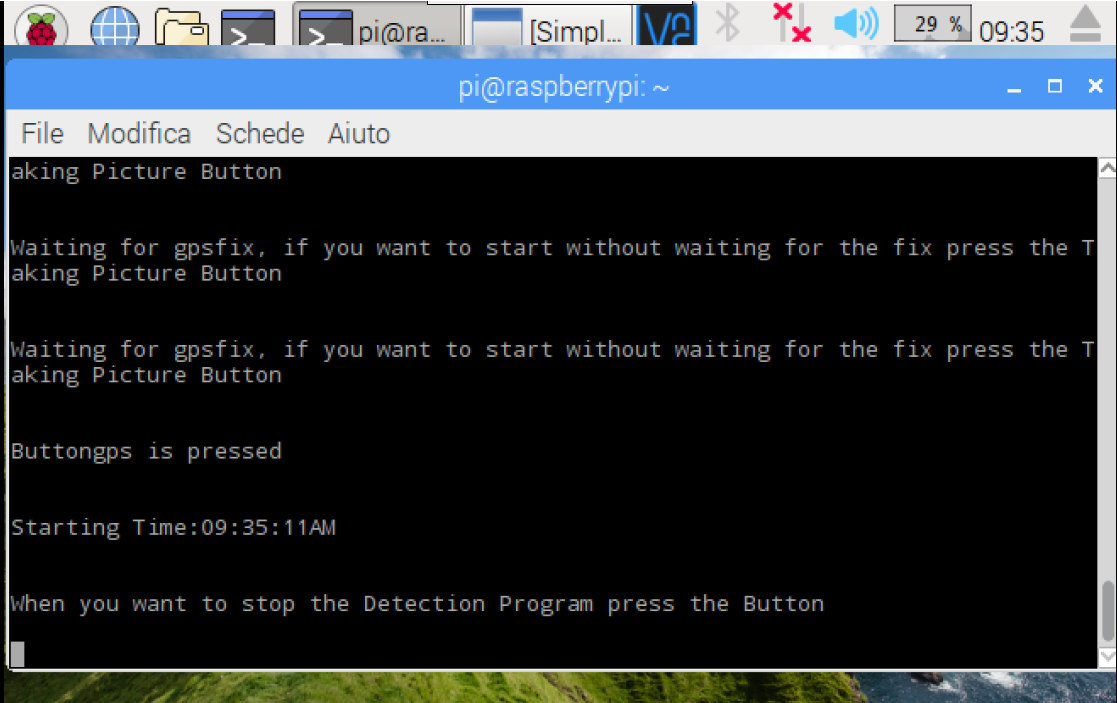
The program will start, the green and the red LEDs will switch on.



You can wait for the GPS to get a fix with the satellites (the GPS led starts to blink) or you can press the button on the breadboard to start the detection without waiting for the GPS to be ready. Depending on what you decide the red led acts differently (if you press the button the led will blinks to tell you that the program received the signal from the button).



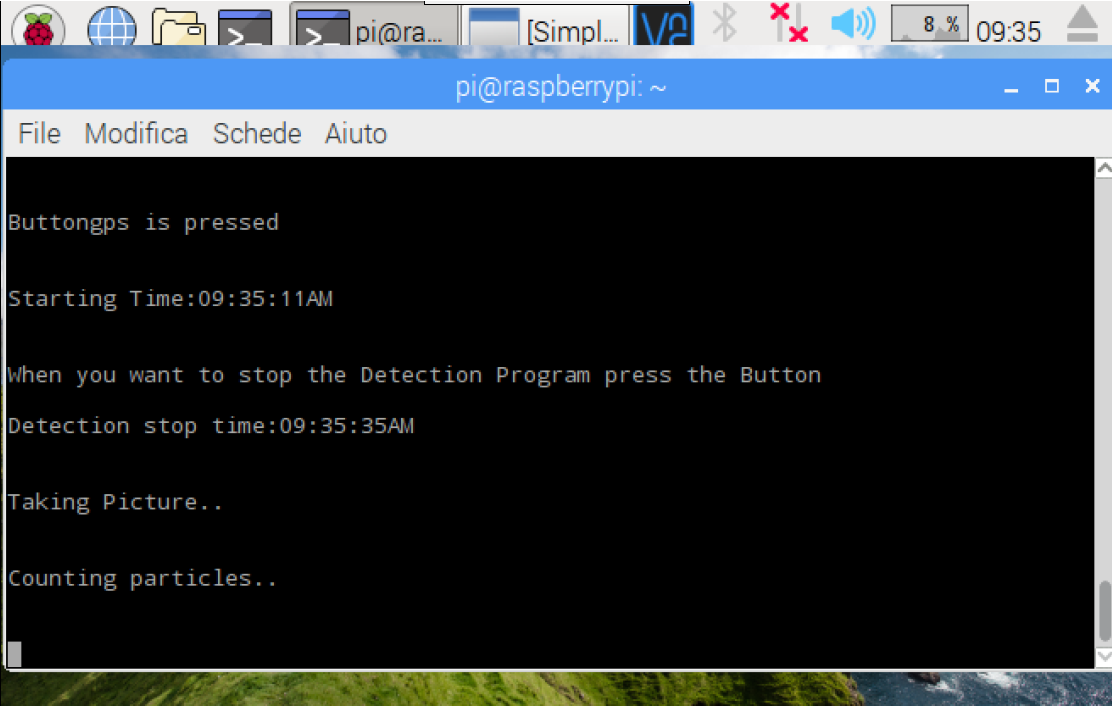
As soon as the button is pressed or the GPS is fixed, the red led switches off and the program outputs the starting time.



You can also see that the SimpleCV window appears, this is where the final frames you take with the camera are shown.

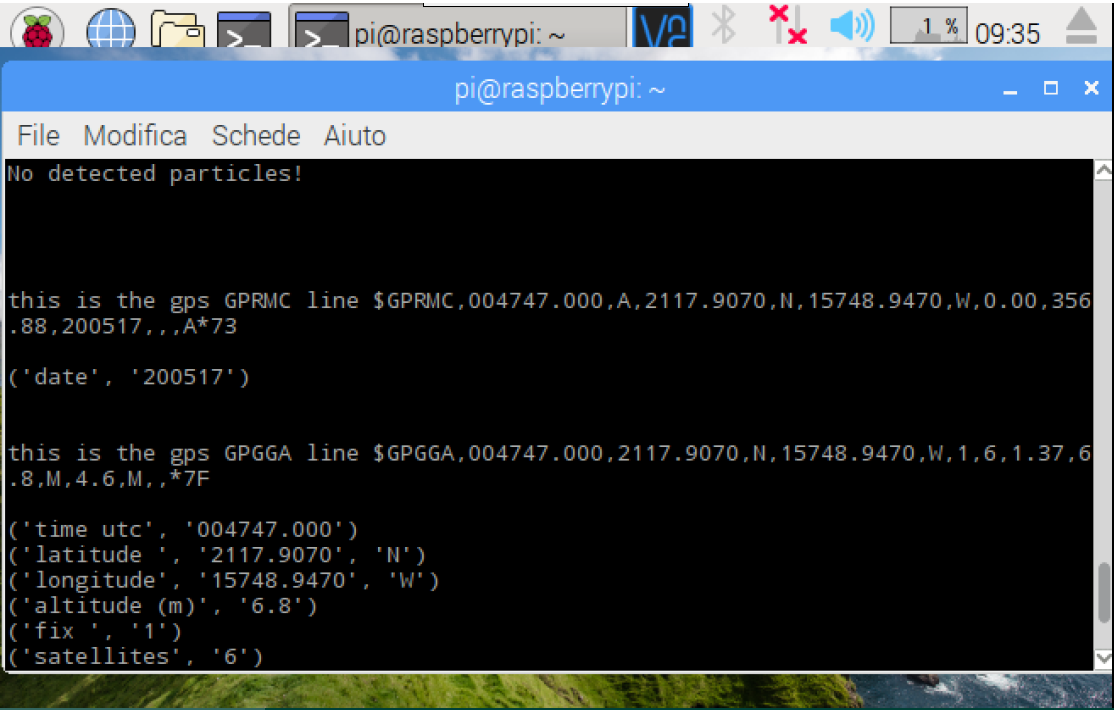


To stop the program just press the button again and make sure that the green led blinks (this tells you that the program received the input from the button). As soon as the button is pressed you will see in the screen the stopping time and the program starts to count the particles of the final frame.

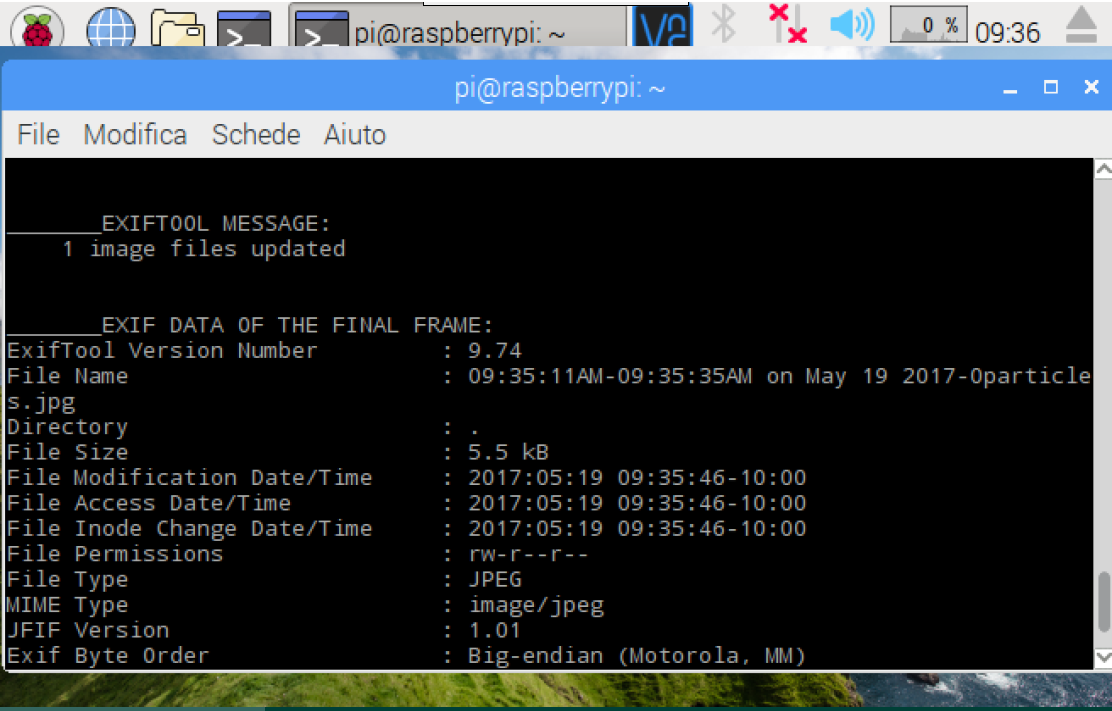


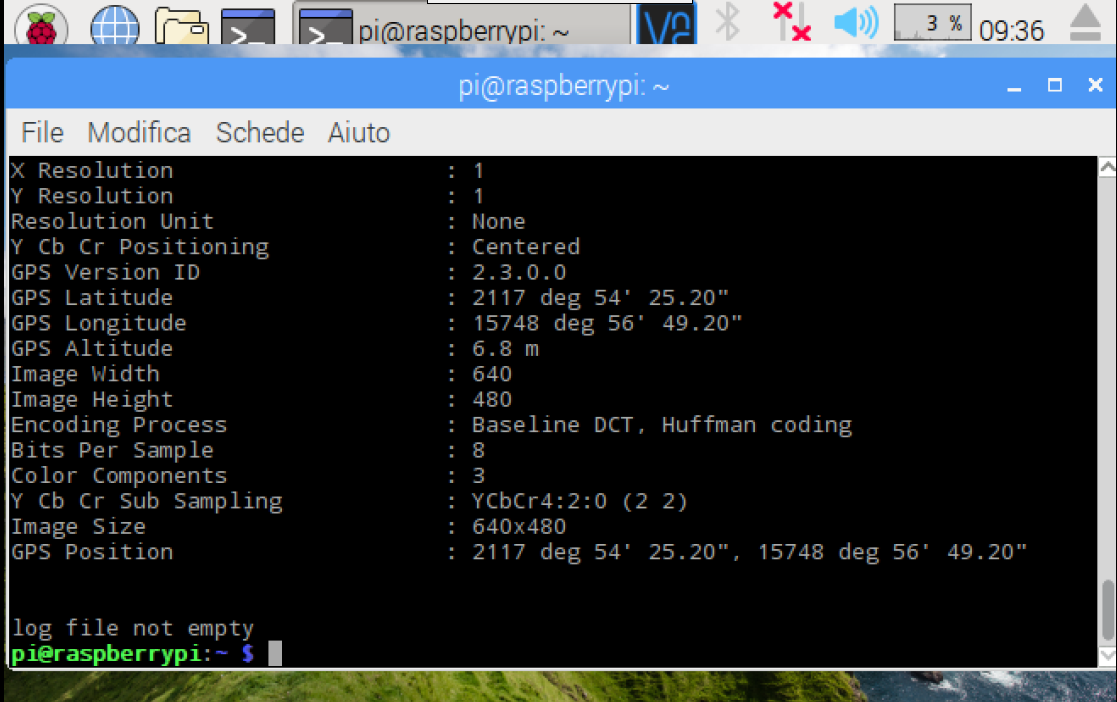
At the end it shows the number of particles detected, the GPS information and the EXIF data attached to the frame.

GPS information’s:



EXIF data:





**Gps data**

The last line tells if the log file was already existing or not.

Now that the program has stopped you can see the final frame with the result of what you detected, and the logfile.csv in which you can read all the information, in the Pi directory.

This is the schema of how the **LEDs are blinking** during the stages of the program:

Start program

GPS Searching for fix Fix Found

Fix not Found. Waiting for Button Press

If Button is pressed

Taking Picture/Searching for Particles

Stop Taking Picture Button is Pressed

Counting Particles

If Founded 0 Particles

If Founded >0 Particles

If GPS data is not saved

END

**RASPBERRY AS AN ACCESS POINT**

If you want to make your Raspberry PORTABLE WHEN YOU DON’T HAVE AN AVAILABLE WIFI CONNECTION you will need to use it as an Access Point.

It means you are going to change the Wi-Fi configuration, in order to make possible to control Raspberry with ANY device that can use Wi-Fi. (This is the Headless mode: when you cannot use your monitor and keyboard, you can use your smartphone! Easy!)

To set it up we just followed the steps indicated in the official website of Raspberry:

<https://www.raspberrypi.org/documentation/configuration/wireless/access-point.md>

Anyway those are…

As always check that what you are writing is correct!! ;)

NOTE THAT AFTER THIS CONFIGURATION RASPBERRY WILL NOT BE ABLE TO CONNECT TO INTERNET WITHOUT ETHERNET CABLE!! (or you have to make a bridge…)

[This means that Raspberry will not be able to update the current date. If you need to change it, do it manually and insert this command. Modify the date as you need!!

LXTerminal: sudo date –s “Mon May 29 12:25:41 UTC 2017”

]

First of all ensure that your Raspberry Pi is updated.

LXTerminal: sudo apt-get update

LXTerminal: sudo apt-get dist-upgrade

Install all the required software with this command:

LXTerminal: sudo apt-get install dnsmasq hostapd

Yes.

Since the configuration files are not ready yet, turn the new software off as follows:

LXTerminal: sudo systemctl stop dnsmasq

LXTerminal: sudo systemctl stop hostapd

**CONFIGURING A STATIC IP**

Raspberry Pi needs to have a static IP address assigned to the wireless port.

This documentation assumes that we are using the standard 192.168.x.x IP addresses for our wireless network, so we will assign the server the IP address 192.168.0.1. It is also assumed that the wireless device being used is wlan0.

First, the standard interface handling for wlan0 needs to be disabled. Normally the dhcpd daemon will search the network for another DHCP server to assign a IP address to wlan0. This is disabled by editing the configuration file:

LXTerminal: sudo nano /etc/dhcpcd.conf

Add

*denyinterfaces wlan0*

to the end of the file (but above any other added interface lines) and save the file and exit.

To configure the static IP address, edit the interfaces configuration file with:

LXTerminal: sudo nano /etc/network/interfaces

Find the wlan0 section and edit it so that it looks like the following:

allow-hotplug wlan0

iface wlan0 inet static

address 192.168.0.1

netmask 255.255.255.0

network 192.168.0.0

Instead of erasing the stuff that we don’t want, just comment it with # at the beginning of every sentence. Save and Exit.

Now you will restart the dhcpcd daemon and set up the new wlan0 configuration:

LXTerminal: sudo service dhcpcd restart

LXTerminal: sudo ifdown wlan0

LXTerminal: sudo ifup wlan0

**CONFIGURING THE DHCP SERVER (DNSMASQ)**

You are now going to rename the dnsmasq configuration file, and edit a new one:

LXTerminal: sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig

LXTerminal: sudo nano /etc/dnsmasq.conf

Type or copy the following information into the dnsmasq configuration file and save it:

interface=wlan0 ( Use the require wireless interface - usually wlan0!!!)

dhcp-range=192.168.0.2,192.168.0.20,255.255.255.0,24h

Save and Exit.

So for wlan0, we are going to provide IP addresses between 192.168.0.2 and 192.168.0.20, with a lease time of 24 hours. If you are providing DHCP services for other network devices (e.g. eth0), you could add more sections with the appropriate interface header, with the range of addresses you intend to provide to that interface.

**CONFIGURING THE ACCESS POINT HOST SOFTWARE (HOSTAPD)**

You need to edit the hostapd configuration file, to add the various parameters for your wireless network. After initial install, this will be a new/empty file.

LXTerminal: sudo nano /etc/hostapd/hostapd.conf

Add the information below to the configuration file. This configuration assumes we are using channel 7, with a network name of Pi-AP, and a password Yourpassword. Note that the name and password should **not** have quotes around them.

interface=wlan0

driver=nl80211

ssid=Pi-AP

hw\_mode=g

channel=7

wmm\_enabled=0

macaddr\_acl=0

auth\_algs=1

ignore\_broadcast\_ssid=0

wpa=2

wpa\_passphrase=Yourpassword

wpa\_key\_mgmt=WPA-PSK

wpa\_pairwise=TKIP

rsn\_pairwise=CCMP

Save and Exit.

We now need to tell the system where to find this configuration file.

LXTerminal: sudo nano /etc/default/hostapd

Find the line with #DAEMON\_CONF, and replace it with this:

DAEMON\_CONF="/etc/hostapd/hostapd.conf"

Save and Exit

**START IT UP**

Now start up the remaining services:

LXTerminal: sudo service hostapd start

LXTerminal: sudo service dnsmasq start

Using a wireless device (your smartphone!), search for Wi-Fi networks. The network SSID should now be present, and it should be accessible with the specified password.

If SSH is enabled on the Raspberry Pi access point, it should be possible to connect to it from another Linux box (or a system with SSH connectivity present) as follows, assuming the pi account is present:

ssh [pi@192.168.0.1](mailto:pi@192.168.0.1)

By this point, the Raspberry Pi is acting as an access point, and other devices can associate with it.

Using it like an **Access Point:**Switch on RSB (SW3). Find VNC Viewer in the Start Panel/Internet

Go to Start Panel/Preferences/Raspberry Pi Configuration / ENABLE VNC.

Switch on the other device (ex Phone): Download VNC viewer from https://www.realvnc.com/download/viewer/.

Connect via Wi-Fi to the Raspberry connection (Pi-AP) (password: Yourpassword).

Open VNC insert Raspberry IP (we configured 192.168.0.1).

VNC will ask for RSB username and password (pi, raspberry-or whichever password you entered before).

Use Phone to control the Raspberry Pi as you would use the keyboard, mouse, and monitor.