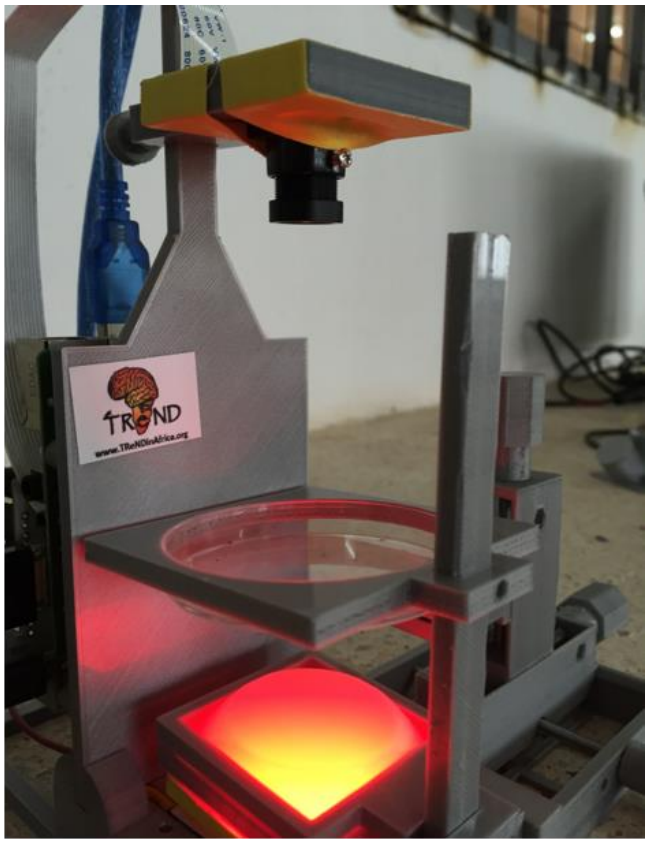
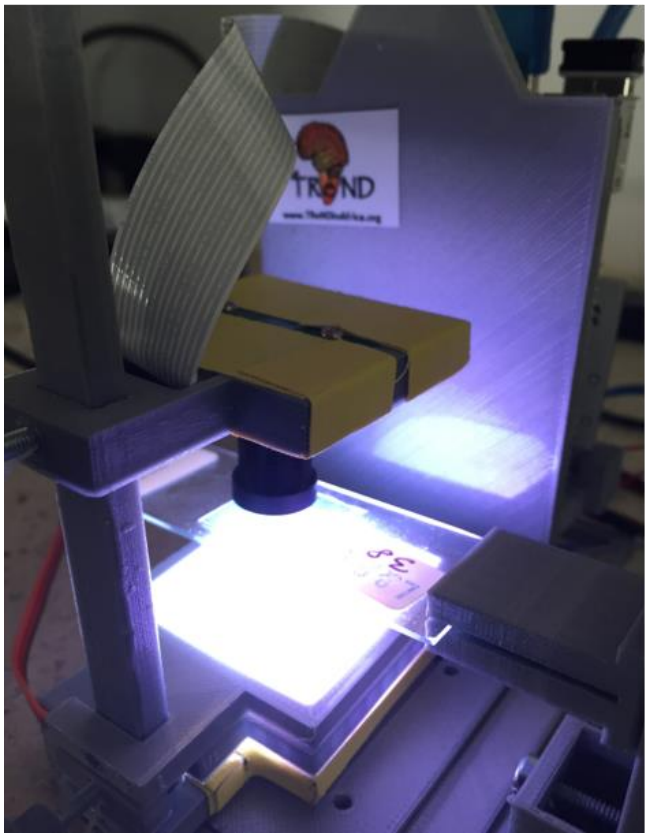
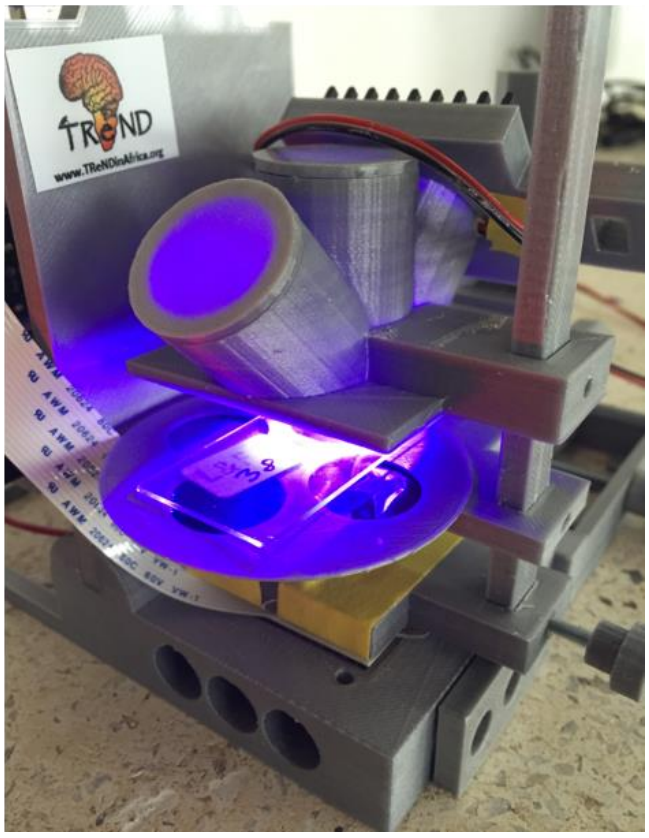
# FlyPi – User and Assembly Manual



# Index

## Software

Here we describe two installation methods.

Method 1 should be used if:

1. If the user wants to know what is “under the hood”, which libraries are necessary, how to update the software in a GNU/Linux distribution, how to install packages and download repositories form GitHub.
2. If the user is located in a place where internet connectivity is always available (but not necessarily reliable as this method requires smaller downloads).
3. The FlyPi is to be used as a teaching tool for building your own lab equipment.

Method 2 should be used if:

1. The user does not mind what is under the hood, and also does not mind using a possibly outdated version of the system.
2. The user is going to use the system in a place where there is poor internet connectivity, and can download a single (large) image file beforehand.

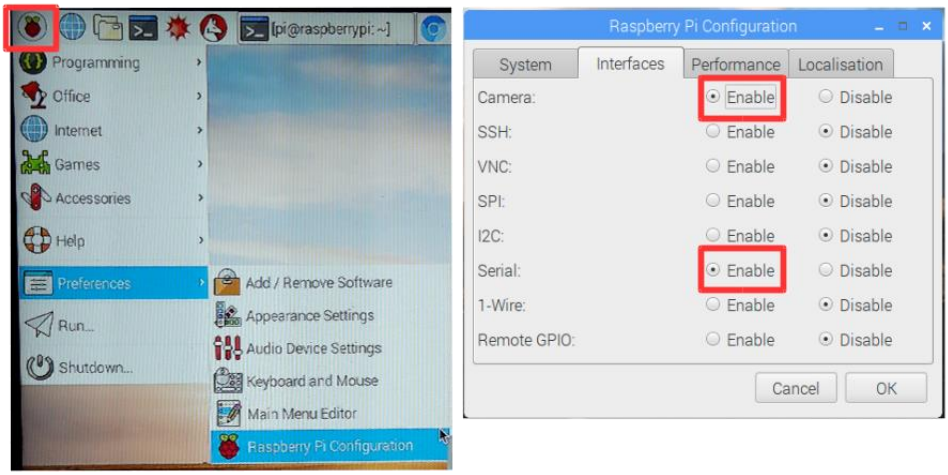
## Method 1 (full)

### Installing the operating system (OS)

The OS used to develop is called Raspian, and can be installed using NOOBS (and https://www.raspberrypi.org/help/videos/#noobs-setup for an excellent video tutorial) or directly via the Raspian image (https://www.raspian.org/RaspianImages).

### Enabling the Camera Module and serial communication

Once the OS has been installed, the camera module and the serial communication need to be enabled. This can be done by going to “Raspberry Pi configuration” and selecting the check-boxes as indicated.



## Running a terminal and installing software on GNU/Linux systems

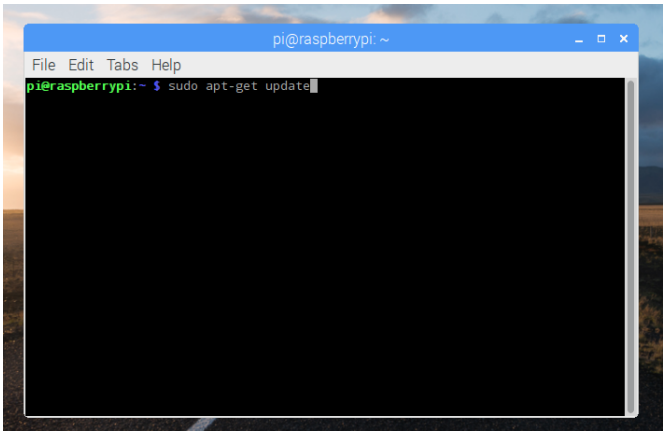
Installing software on GNU/Linux distributions (as the one used on the Raspberry Pi), is easily done using a command terminal. The Rasperry Pi command terminal is called LXTerminal (the Raspberry Pi Foundation has a nice intro to the Terminal, which can be found on: ).

It is recommended to update and install all upgrades for the OS before installing the necessary software. This can be done via Terminal using the commands below:

*sudo apt-get update*

*sudo apt-get upgrade*

(This may take several minutes to complete)


Once the update is done, power off the Raspberry Pi and disconnect it from the power line to connect the camera cable to the camera port as detailed here: https://www.raspberrypi.org/documentation/usage/camera/

The FlyPi runs on two pieces of software, one running on the Arduino board and the second on the Raspberry Pi. They are both described below.

## Arduino

The code used to control the Arduino board, called a “sketch”, can be easily uploaded using the Arduino IDE. The steps to install it are better described at , in the sub-section “Install the Arduino Desktop IDE”. On the same website the steps necessary to upload sketches and check that everything is running properly can be found at the specific section for the Arduino Pro board https://www.arduino.cc/en/Guide/ArduinoPro.

To install the IDE, open a terminal (check the **Running a terminal and installing software on GNU/Linux systems** section) and type the following command:

*sudo apt-get install arduino*

If the LED ring and/or Matrix from Adafruit are going to be used, it is also necessary to install the respective Arduino Libraries, from the Adafruit repository:  and

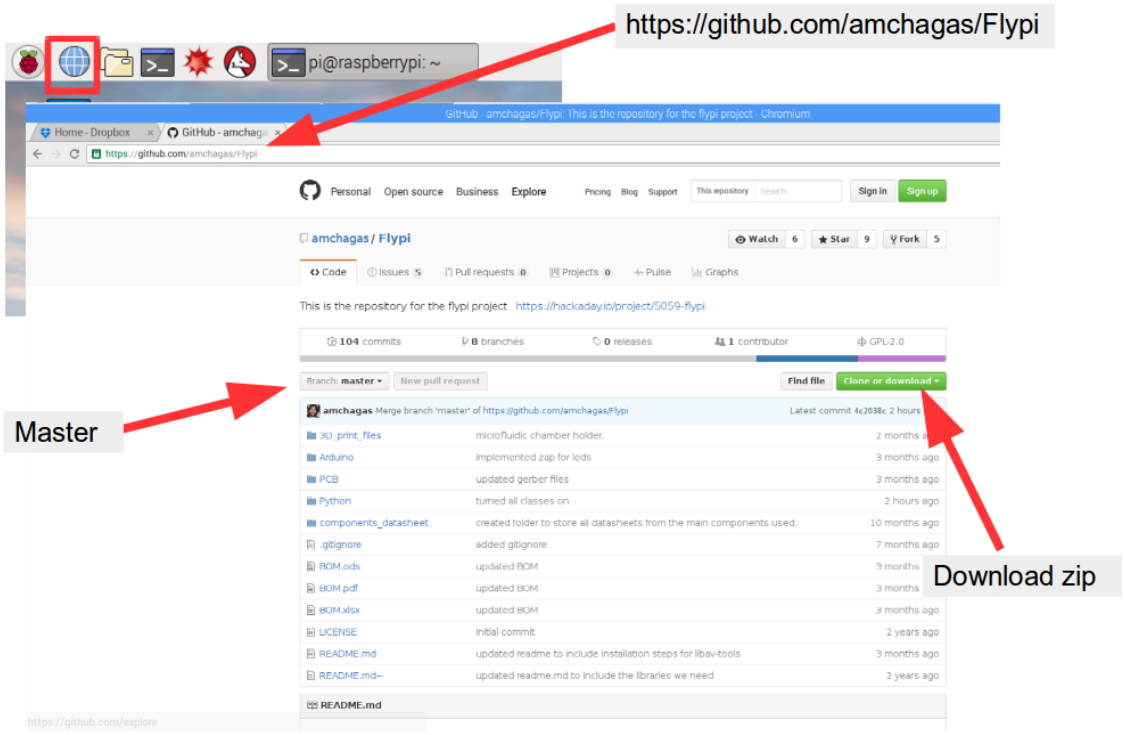
---

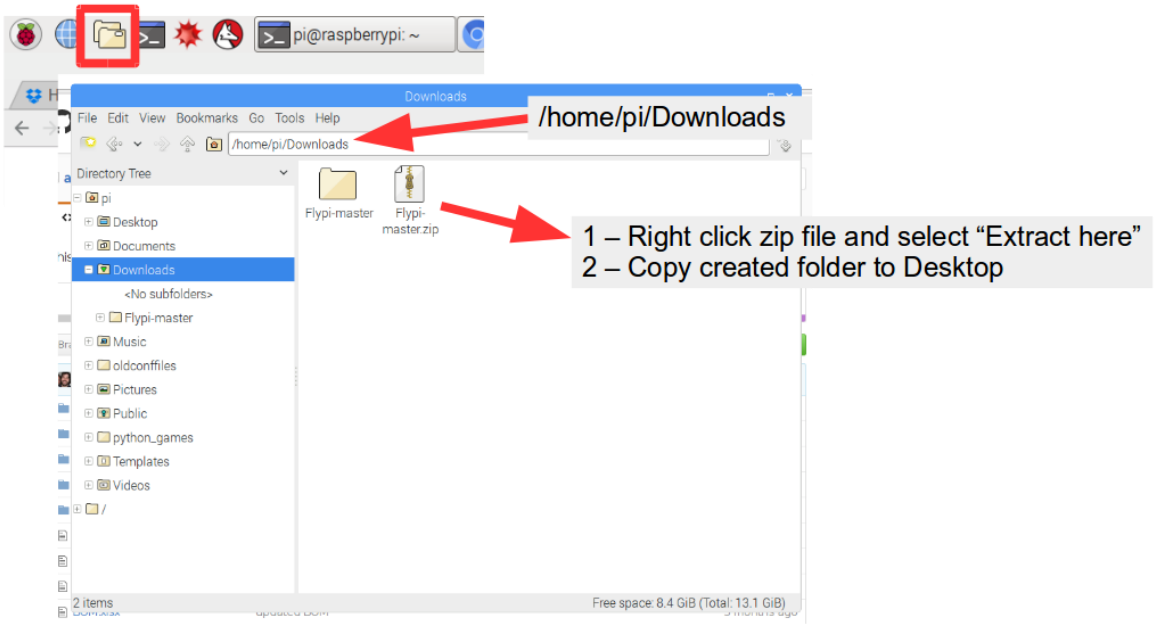
## Setting up the Python Graphical User Interface (GUI):

Open up a terminal (check the **Running a terminal and installing software on GNU/Linux systems** section) and install the library required to convert ".h264" videos (standard output of the PiCamera) to ".avi" videos (that can be loaded by ImageJ and other analysis software). On the terminal type:

*sudo apt-get install libav-tools*

After the necessary libraries are installed, the GUI software can be installed. To do so, download the zip file from the GitHub repository and unzip it on the desktop






Now the system should be ready to run the GUI (see **Starting the GUI** section).

---

## Method 2 (easy)

If the user has a reliable Internet connection with enough band, an “image” file, containing all software pre-installed can be downloaded and cloned to an SD card.

The disadvantage of this method is the necessity to download a 16 GB file and the necessity of having a 16GB (or higher) SD card.

Using a computer connected to the Internet, download the FlyPi image from this link (currently a dropbox folder, maintained by the developers and containing the most up to date version of the system) – and clone it to the SD card to be used with the Raspberry pi.

Installing the image to an SD card is better described here https://www.raspberrypi.org/documentation/installation/installing-images/

After cloning the image, the SD card can be inserted on the Raspberry Pi and the system started.

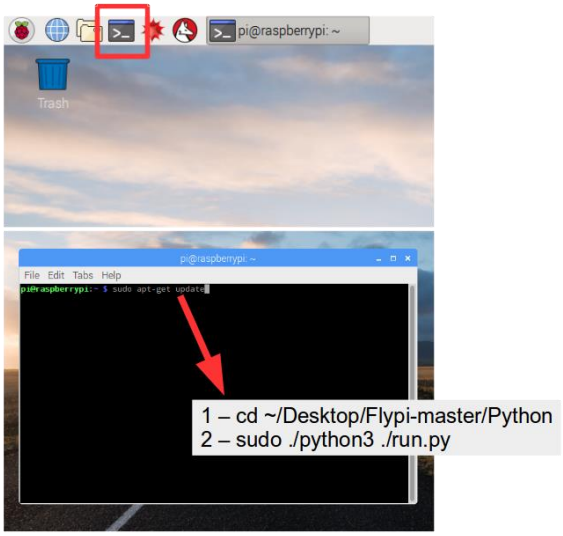
---

## Starting the GUI

Once everything is installed, the GUI can be started by opening up a terminal and typing:

*cd ~/Desktop/Flypi-master/Python/*

*sudo python3 ./run.py*




## Data location and Remote control

All data generated by the FlyPi GUI (videos, snapshots, temperature logs, protocols) are stored on the desktop, in an automatically created folder “flypi_output”. Here, data is further divided into subfolders according to the data type.

To change the place where the data is stored (e.g., to a USB memory stick) locate the variable “basePath” in the file flypiApp.py and change it to the new location:

basePath = '/home/pi/Desktop/flypi_output/'  (default)

basePath = '/dev/sdb/'  (hypothetical location)

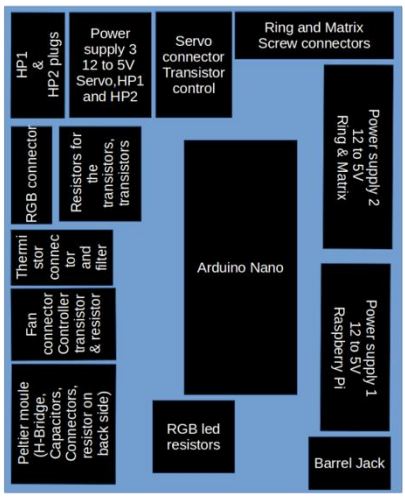
This tutorial provides information on how to locate the path to a USB memory stick on Linux systems.

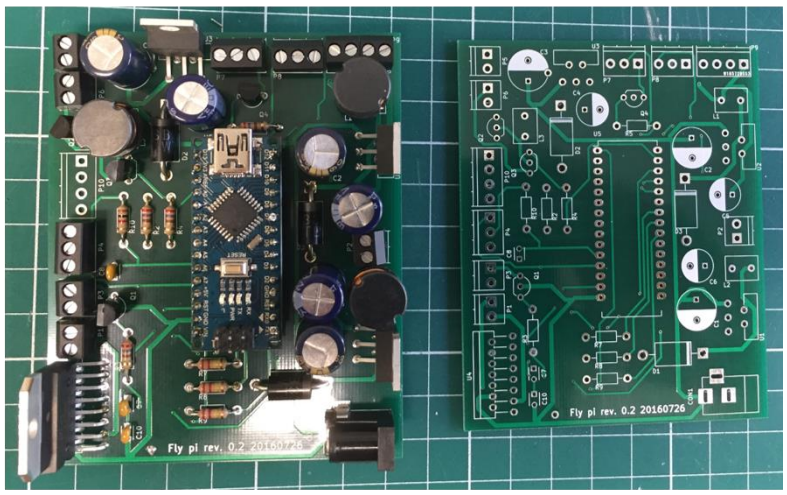
To avoid overwriting files, files created follow this name convention: “Year-Month-Day-Hour-Minute-Seconds”+ extension (e.g., 2017-03-31-10-50-23.h264 for a recorded video).

Once all necessary software is installed, it is possible to control the FlyPi remotely, for example using a Laptop or Desktop computer. There are many useful tutorials online showing how to do this using different protocols (SSH, VNC). One example can be found here. One alternative is the use of software dedicated to this task, such as TeamViewer, which is free (for non-commercial use) and easy to setup, but proprietary.

# Electronics

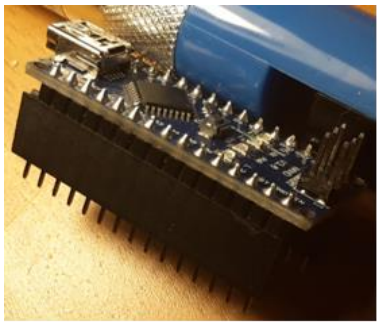
The printed circuit board (PCB) will be populated with “through-hole” components. They are arranged in functional modules and their soldering order does not matter. Our suggestion is to start by deciding which modules will be installed and to start soldering smaller components (e.g. resistors, transistors), independent of which module they belong to, and move on to bigger components (integrated circuits, capacitors) afterwards. Another suggestion is to place all components in place and bend their leads on the bottom side of the PCB if possible. This will allow the user to make sure everything fits and is well placed before soldering.

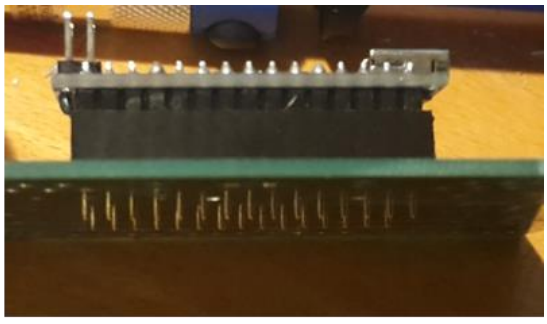


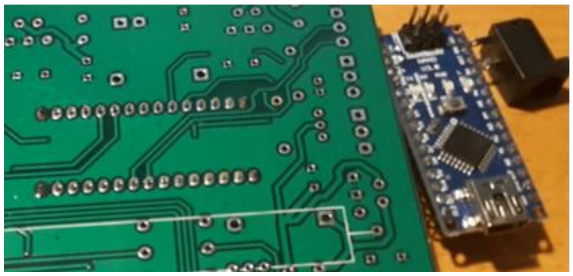




## Arduino connectors

Plug the female headers to the Arduino board. Next, place the headers in the designated area on the PCB and solder only the edge pins (two per header). Disconnect the Arduino from the headers and finish soldering the remaining pins. This method ensures that the Arduino and the header pins will be aligned once soldering is done, and that the microcontroller won’t be damaged by the soldering heat.

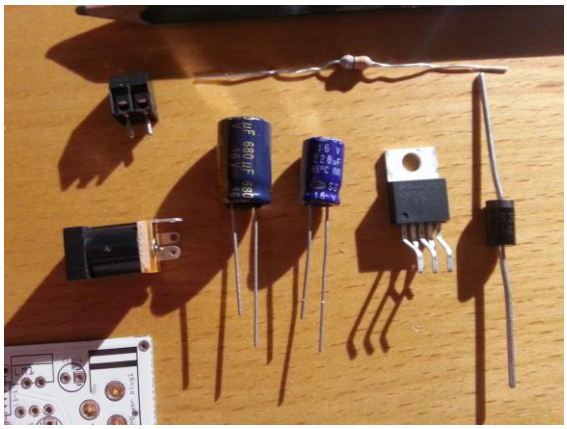


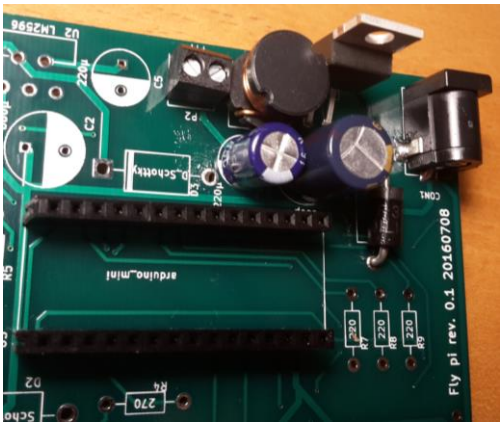




## Raspberry Pi power line

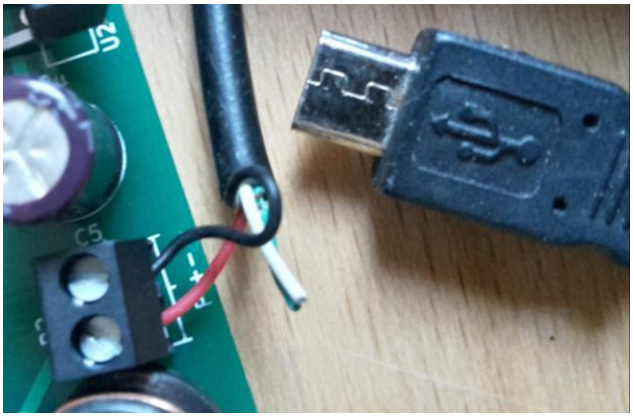
The PCB contains a circuit dedicated to powering the Raspberry Pi (it converts 12V from the main input into 5V). This circuit is close to the barrel jack, and the main concern upon soldering the parts is the orientation of the capacitors and diode. Their position and value are indicated by the print on the PCB. Note: Larger capacitors such as these ones and diodes have a polarity – make sure to solder them the right way around. By convention, the “short” leg is (-) and should be slotted into the white part. Note that usually this is also somehow indicated by the writing/patterning of the component itself.

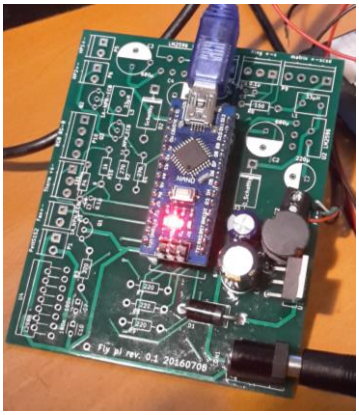





To finish the Pi power line, take the USB cable that powers the Raspberry Pi (A to micro B) and cut it close to the A part (the end that is NOT connected to the Raspberry Pi’s power port). Within the cable, there are four wires, two of which need to be exposed (and the others simply cut short). By convention, the red and black wires are positive (+) and ground (GND) power wires, respectively. Once stripped, they need to be screwed into the corresponding ports of the screw terminal as indicated.

Once the Pi power line and the Arduino connectors are finished, the system should be tested before any other modules are soldered. Plug the Arduino into the connector (USB port side on the opposite side of the Pi power supply), the USB cable from the PCB into the Raspberry Pi. Turn on the system by inserting the power supply plug into the PCB’s barrel jack. Make sure the Raspberry Pi cables (keyboard, mouse and monitor) are connected and the SD card is properly inserted before doing so. The system should boot up.





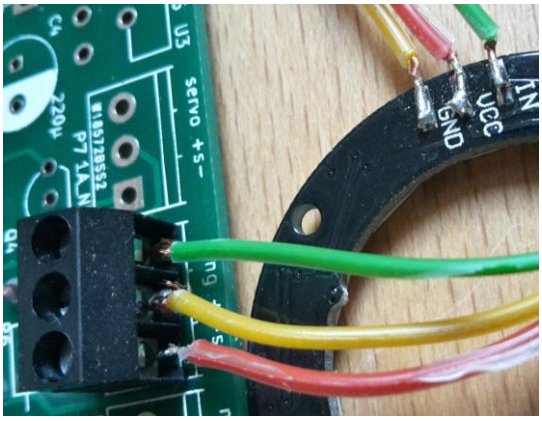
## LED Ring and LED Matrix modules

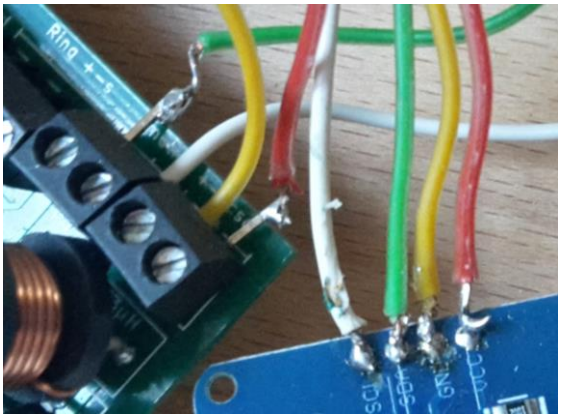
The LED ring and the LED matrix share the same power module (composed of the same components of the Pi power module and located next to it), but can be installed independently from one another. *WARNING: The LEDs are rather bright at full power – directly looking at them for too long could cause damage to your retina!*

The power module and screw terminals should be soldered in the same way as the Pi power module (see above).

Solder 3 cables (~15-20 cm) to the back of the LED ring / diamond as indicated (+, GND, Signal). The 4th connector is left unattached. Care should be taken to not damage the LEDs by overheating the back of the array (it's quite robust, but just in case). If both the ring and the diamond are to be used in tandem, the respective pins can be connected to each other (i.e. + to +, GND to GND, S to S) rather than using 6 long cables. Note, however, that in this mode the individual pixel control will not be straightforward. The entire array can however still be controlled as one without problem which will suffice for most applications. If desired, fix the diamond into the hole of the ring using tape or glue.

Care should be taken that the wires leaving the screw terminals on the PCB connect the correspondent place in ring and/or matrix.



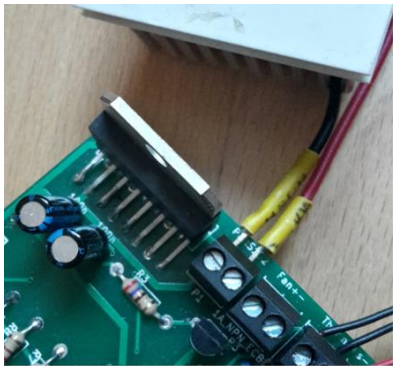

## Peltier Module

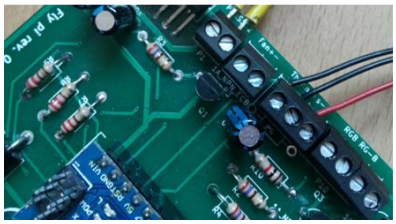
Start with the H-bridge (the tall component with 8 pins), making sure its flat side faces the outside of the PCB. Next solder the capacitors, making sure they are correctly oriented (short leg into white). Place the screw terminals (one for the Peltier itself and one for its cooling fan) and screw in the cables (orientation is also important here. The proper orientation is determined by the marks on the edge of the PCB). Last for the proper functioning of this module the high capacity resistor needs to be soldered on the back side of the PCB. Next solder the resistors for the RGB LED and its screw terminal. Last, place the capacitor and resistor for the temperature sensor and its screw terminal.

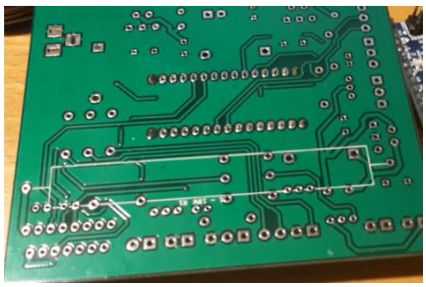
If the RGB and Thermistor are to be used (part of the Peltier module, or in case of the RGB LED also useful as indicator LEDs during "protocols" see GUI) solder 4 cables (~10-15 cm) to the 4 pins of the RGB-LED, taking note which cable connects to which pin. The longest pin is typically the anode, with the single pin next to it being the one for the red LED. The other two, in order, are green, then blue. If in doubt, applying a small voltage (e.g. from a multi-meter in "diode mode") across the LED between the anode pin and one of the others should make the respective LED light up. Insulate the contacts to the LED against each other and slide some

small plastic tubing (or heat-shrink/tape) over the assembly as indicated. The other ends of the RGB led cable should be connected to the assembled PCB as indicated.

The thermistor (optional) is assembled in essentially the same manner (with 3 pins rather than 4, of course) as the RGB led. The pins on the sensor are +, signal and GND, when the curved part of the sensor is facing down and the pins oriented towards the observer (check the sensor datasheet for more details: http://www.analog.com/media/en/technical-documentation/data-sheets/AD22100.pdf)



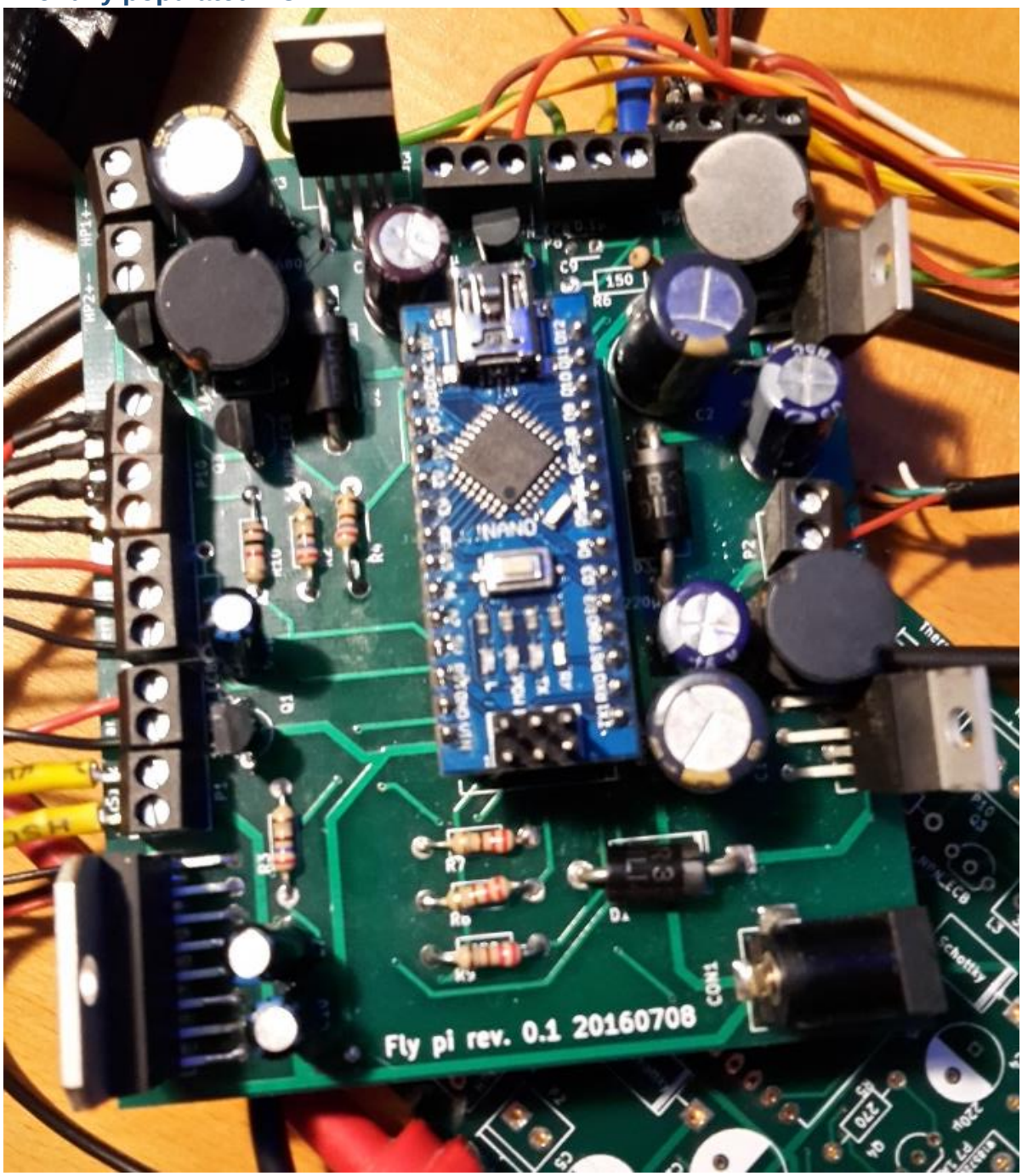





## Servo and high power line module

The servo motor and the high-power lines (e.g. to drive the LEDs of the Fluorescence module) are powered by a third power module, exactly the same as the Pi power module and should be soldered in the same manner. This module has transistors, resistors and screw terminals as the remainder pieces. Start with the resistors and move on to the transistors taking care that they are positioned in the right way, with the rounded side aligned with the drawing in the PCB.

## The fully populated PCB

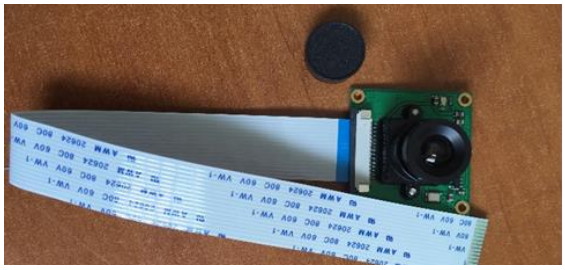


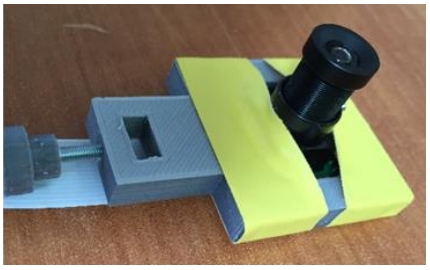
# 3D printed parts

To print all the necessary parts, a printer with minimal print area of 200x200x150 mm should be available, as the longest part has a length of 223.5mm (place it at 45°if required). Alternatively, we suggest to print all parts that would fit the printer's bed and use a print on demand service (there are several available at 3Dhubs.com) to print the remaining ones. If no printer is available at all then this option can also be used to source printed parts – this should come to ~40 € if using a low-cost service.

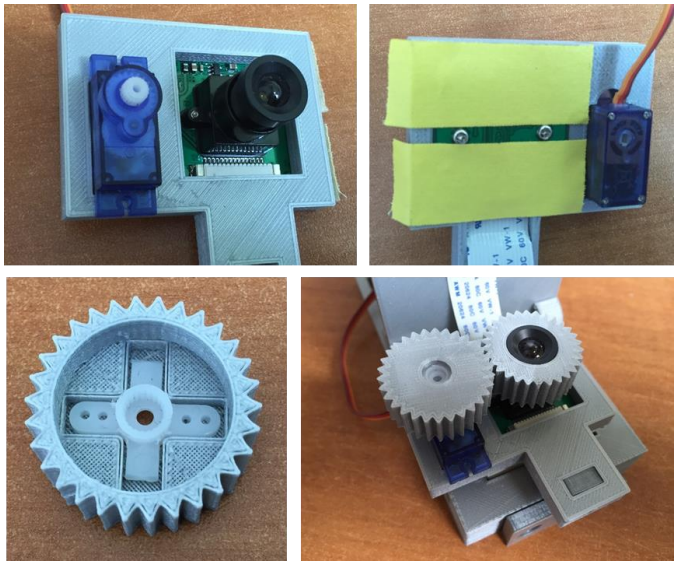
## Mounting the camera

*Option 1 (manual focus).* Place the "Adjustable focus RPi camera" into the 3D printed holder as indicated. The camera PCB should fit easily and finish evenly against the top of the frame (back of camera). If not, file down the 3D printed part until it fits. Fixate camera in place e.g. with tape as indicated, optionally taking care that the integrated LED next to the CCD chip is blocked with tape (this can also be software-disabled if preferred). Next, slot an M3 screw of appropriate length fitted with a 3D printed thumbscrew (e.g. 12-20 mm) into the mounting hole.



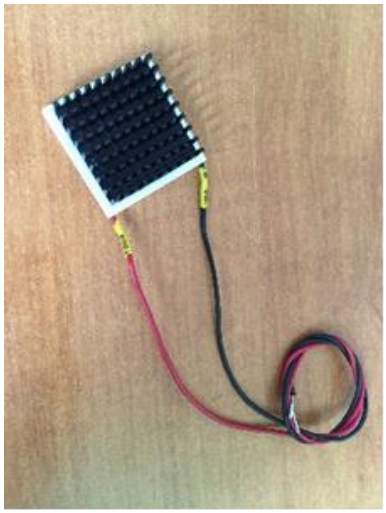
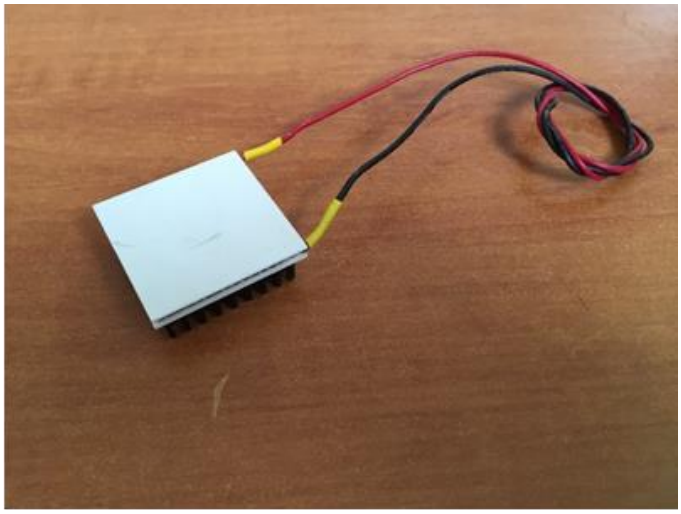
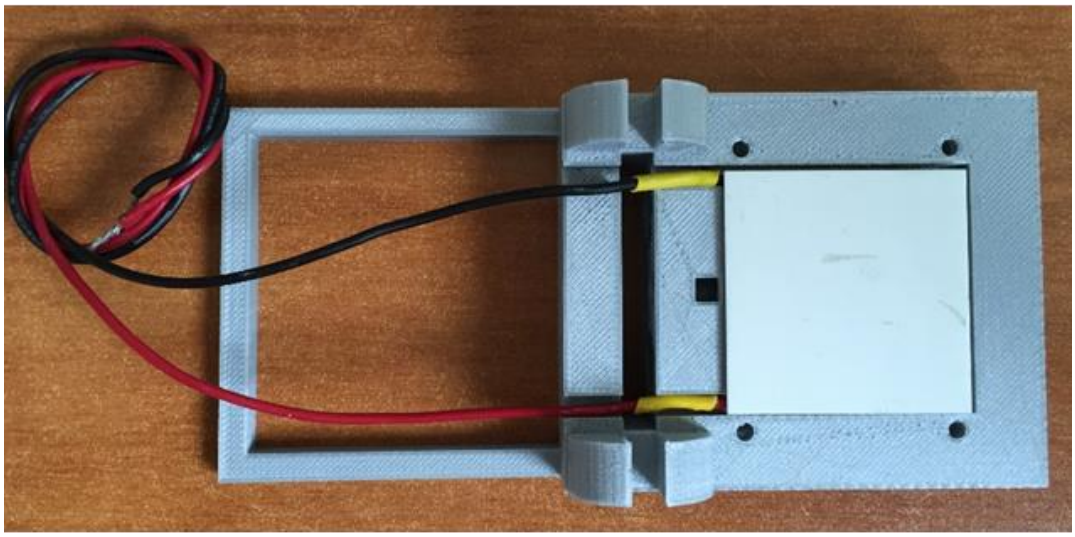
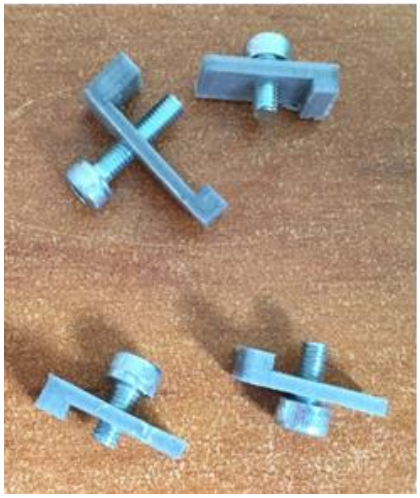
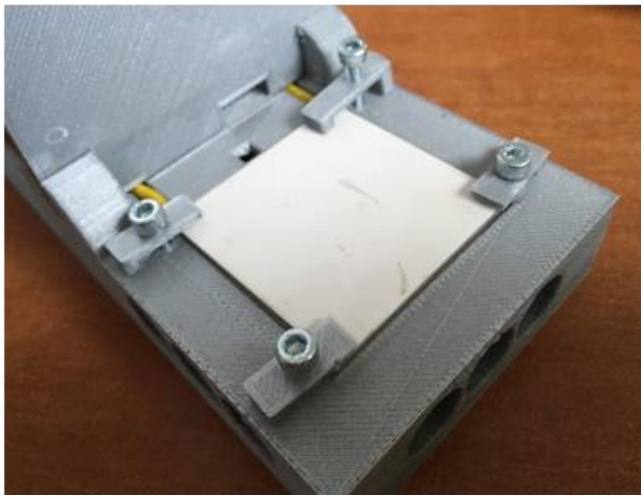


*Option 2 (motorised focus).* As above, but use the extended 3D printed Camera mount and locate the two 3D printed cogwheels. Place the continuous rotation micro servo through the slot next to the main camera mount, carefully noting its orientation as indicated, and fix in place using the screws provided with the motor (you should be able to just screw these straight into the 3D printed frame – predrill small holes with a Dremel or similar if necessary). Alternatively, the motor can be fixed with tape, or may even stay in place by friction alone. Next, fully unscrew the camera objective and fit the cogwheel without the X-shaped inside mounting slots over it as indicated. If it does not stay in place by friction alone, use a small drop of glue, taking care that the objective lens remains clean. Next, locate the X-shaped mounting adapted that comes with the motor and slot it inside the remaining cog-wheel as indicated, again taking careful note of the orientation (the motor will be attached from the same side). Again, use a small drop of glue if the X-mount does not stay in place by itself, carefully ensuring centring. With the camera module mounted as in Option 1, screw the objective back in and complete the assembly by slotting the motor-cogwheel over the motor. The cogs should turn easily when rotated by hand with only the servo resistance blocking motion. If not, carefully check alignment and file down the cogs slightly as required.



## Assemble the Peltier Element with heat dissipation

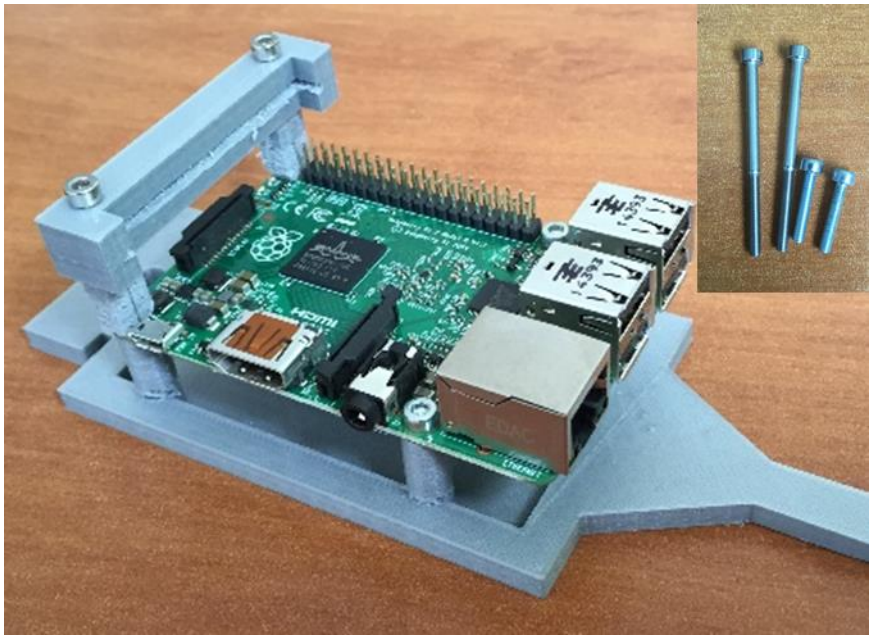
If a Peltier Module will be added, place a generous amount of heat-conducting paste on top of the heat-dissipater (the flat side). In addition, place a small amount of glue to the 4 corners of the same face and glue everything centrally to one side of the Peltier element (either side is fine – just swap the red and black wire connectors on the PCB if it turns out to be “inverted”). Make sure the glue is set before proceeding. Leave the CPU fan aside for now, this will be added at a later stage.

## Mounting the RPi onto the main wall-plate

Lay the 3D printed wall-plate flat against the table with the screw-mounts facing upwards and orient the RPi2 on top. It should fit precisely with all 4 screw-holes aligned. Locate 2 12-16 mm M3 screws and loosely attach the top 2 holes (by the USB ports) as indicated. Next, locate the 3D printed PCB mount and place it on top of the remaining 2 screw-holes such that all holes are aligned. IMPORTANT: Carefully check that the part of the PCB mount that touches the RPi2 will not squish any protruding elements of the RPi2. If there is slight overlap, file down

the concerned part of the PCB mount before proceeding. Then take the two long M3 screws (~30 mm) and mount everything together as indicated. Also tighten the top 2 screws previously inserted.

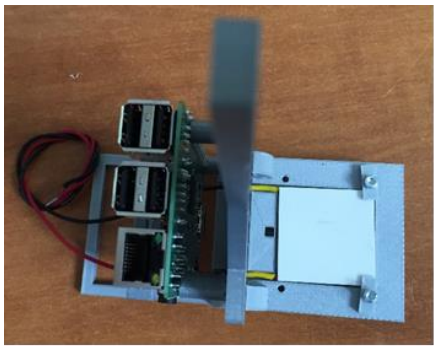


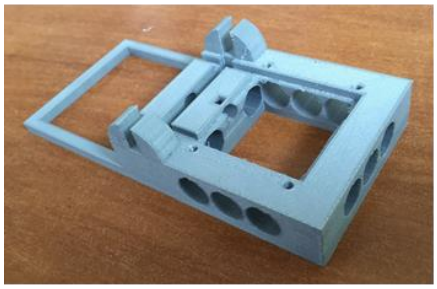
*(note that the picture shows an outdated version of the PCB holder mounted on top of the RPi – the current one is a bit wider)*

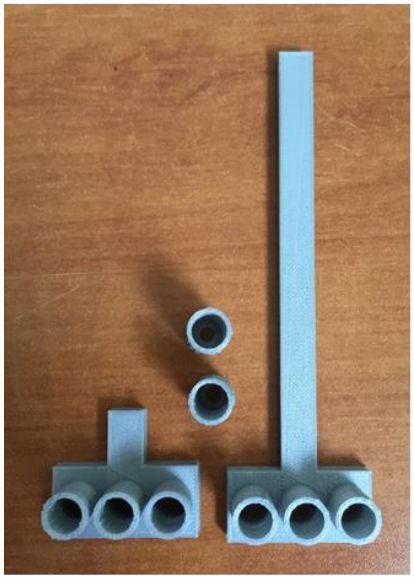
## Mainframe

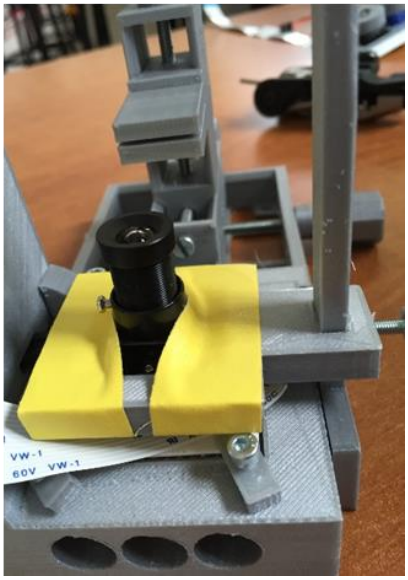
Option 1. With Peltier Element. Place the assembled 4x4 mm Peltier element with heat dissipater below (Part 4) into the 3D printed base as shown and clamp down with 3* x mm screws and the 3 shorter 3D printed Peltier clamps. The 4th, slightly larger clamp will be mounted with a longer M3 screw in the remaining slot as indicated. This will eventually hold the Thermistor in place. Next, slot the assembled wall-plate (above) into the base with the RPi2 on the back as indicated. Make sure the Peltier cables slot easily though the lateral slits provided (you will have to simultaneously slot in the Peltier cables and wall-plate). Next, mount the camera. For this, slot the ribbon cable into its plug on the RPi2 and fix in place using the "clip" that comes pre-attached to the RPi2. Slide the Camera over the wall-plate's turret and fix in place for now using the thumbscrew. Finally, slot the fully-assembled PCB into the mount as indicated and connect the Arduino to the RPi2 using the USB cable provided. By default, the FlyPi scripts are configured to expect the Arduino to be connected in its nearest USB slot (central, upper).

Optional mounting turrets or a 3D printed micromanipulator can now can be slotted into the front or side faces of the base as required.





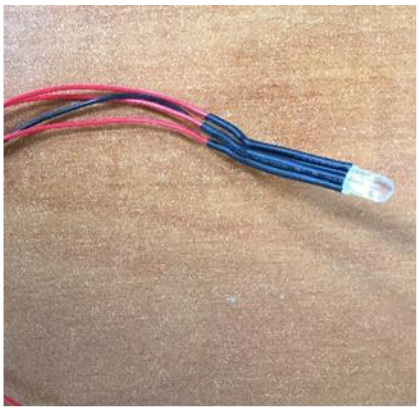
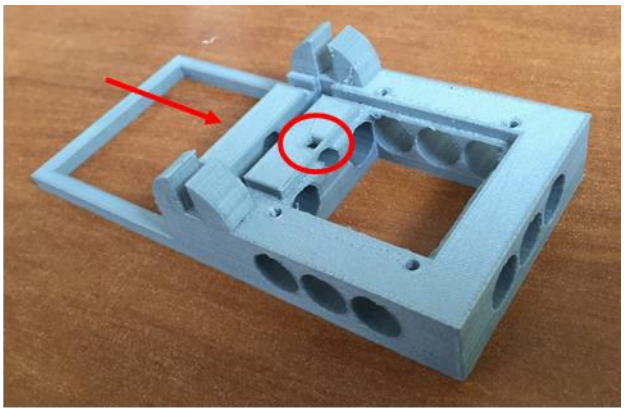
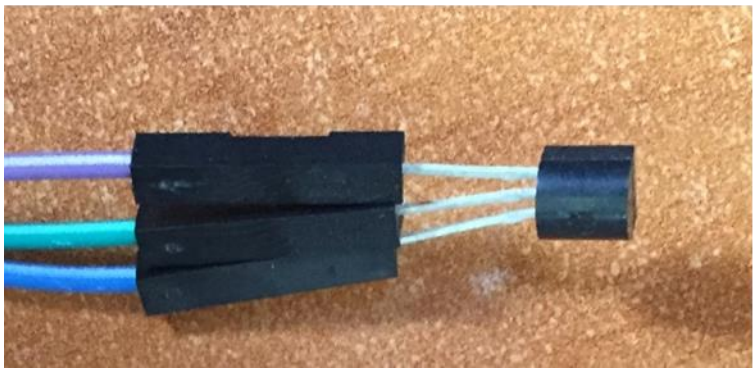
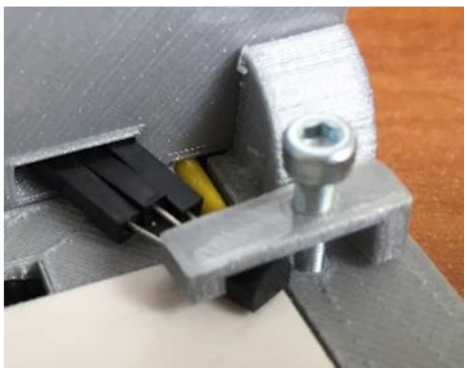




## RGB-LED and Thermistor

The RGB LED can be slotted through the back of the base/wall-plate as indicated until it is clearly visible through the hole from above. (Of course, depending on the final application of the FlyPi, the LED can be positioned in any number of ways.)

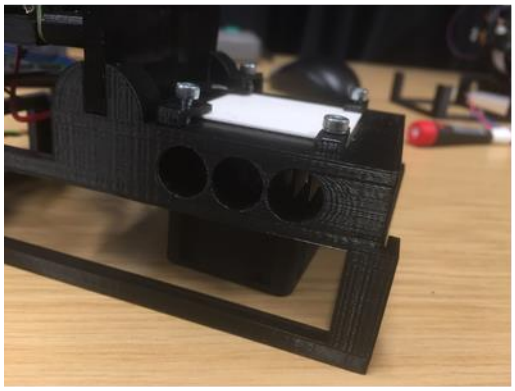
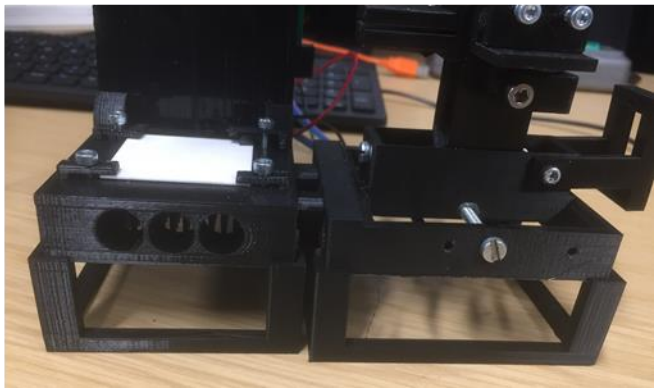
The temperature sensor is slotted through the slightly larger and elongated slot in the wall-plate as indicated.

## Peltier Fan and “Feet”

To better dissipate the heat of the Peltier it can be helpful to add a fan. This is simply placed beneath the Peltier, while the FlyPi is propped up by 3D printed feet. Note that there are also “feet” available for the micromanipulator if required.



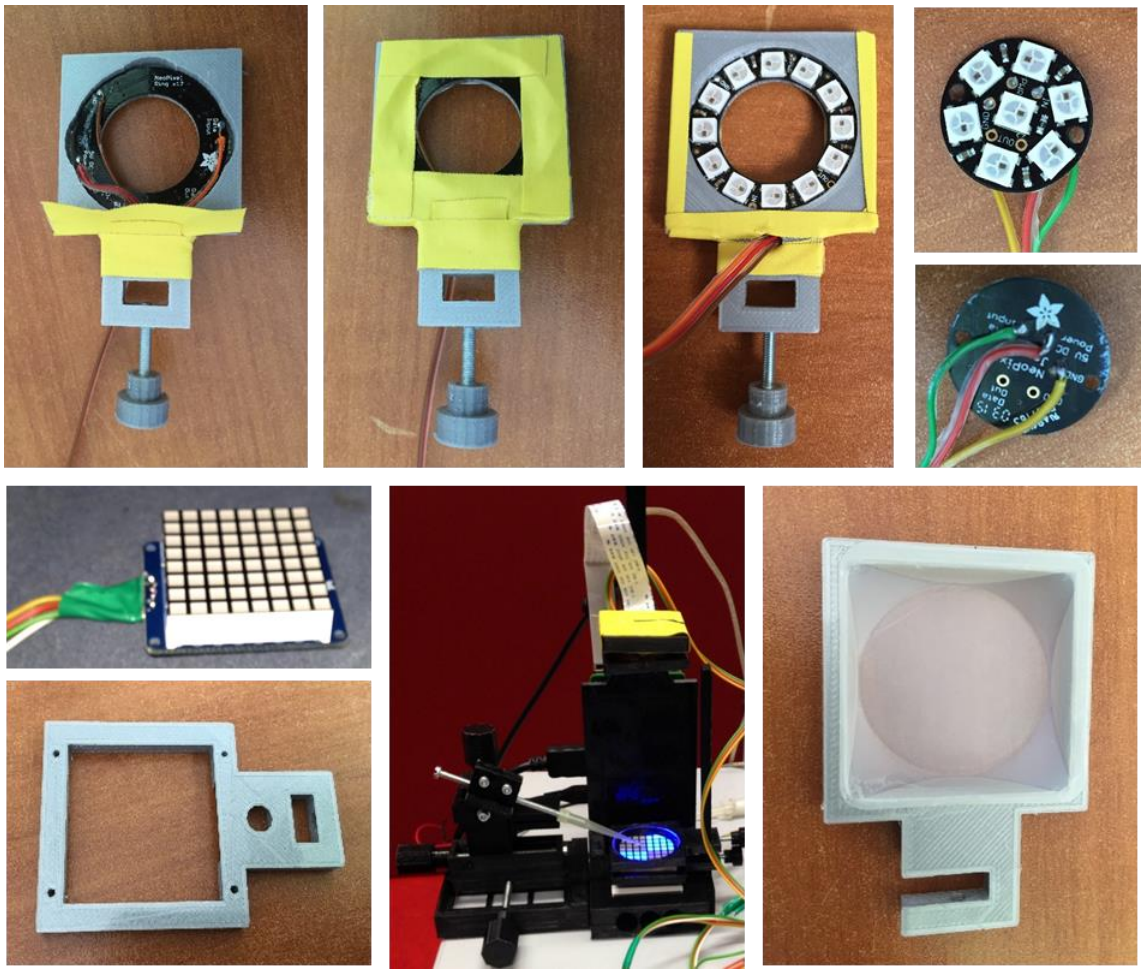
In its most basic format, this completes the FlyPi assembly. In the below, we now describe the assembly of the various optional modules that can be included in the build.

## NeoPixel LED ring / Diamond / Matrix

Place the ring into the 3D printed holder, noting the orientation of the small protrusions relative to the widenings in the mounting adapter. Slot the cables through the hole in the neck. Finally, fix the LED ring in place using tape. The other end of the LED ring cables are connected to the main PCB as indicated.

For the LED Matrix, follow the simple instructions online on Adafruit website (https://learn.adafruit.com/adafruit-led-backpack/), slot it into the 3D printed mount and connect to the PCB as indicated.

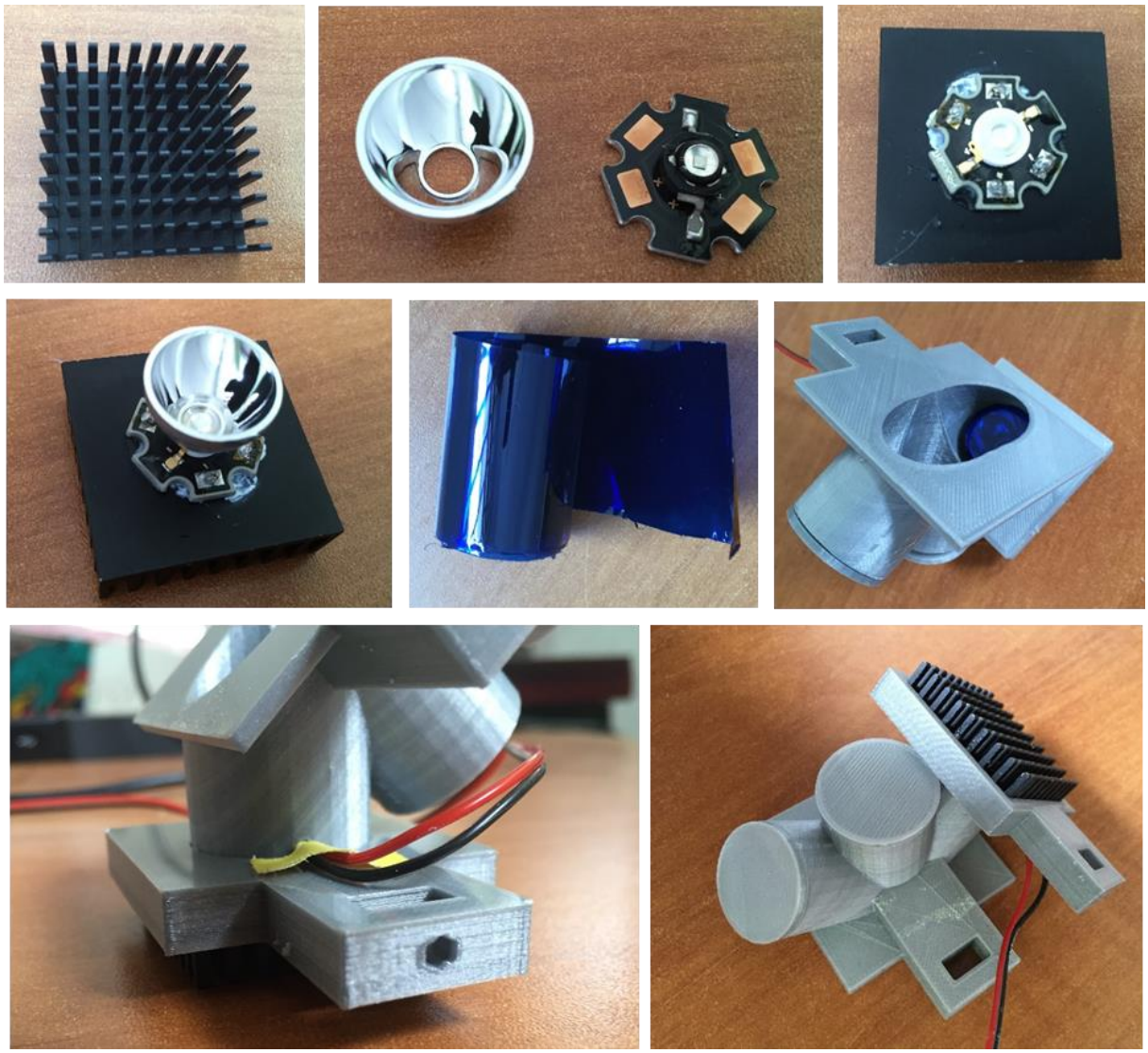
Finally, if required, prepare “diffusors”, e.g. using the milky-white weighing boats glued into place in the 3D printed mounts provided. Alternatively, plain white paper acts as an efficient diffusor as well (but be careful not to burn the paper, the LEDs can get hot), as does any other milky white plastic or a Teflon screen (e.g. plastic milk bottles as used in many countries work great). Two diffusors separated by a few mm are particularly effective (not shown).



## Fluorescence module

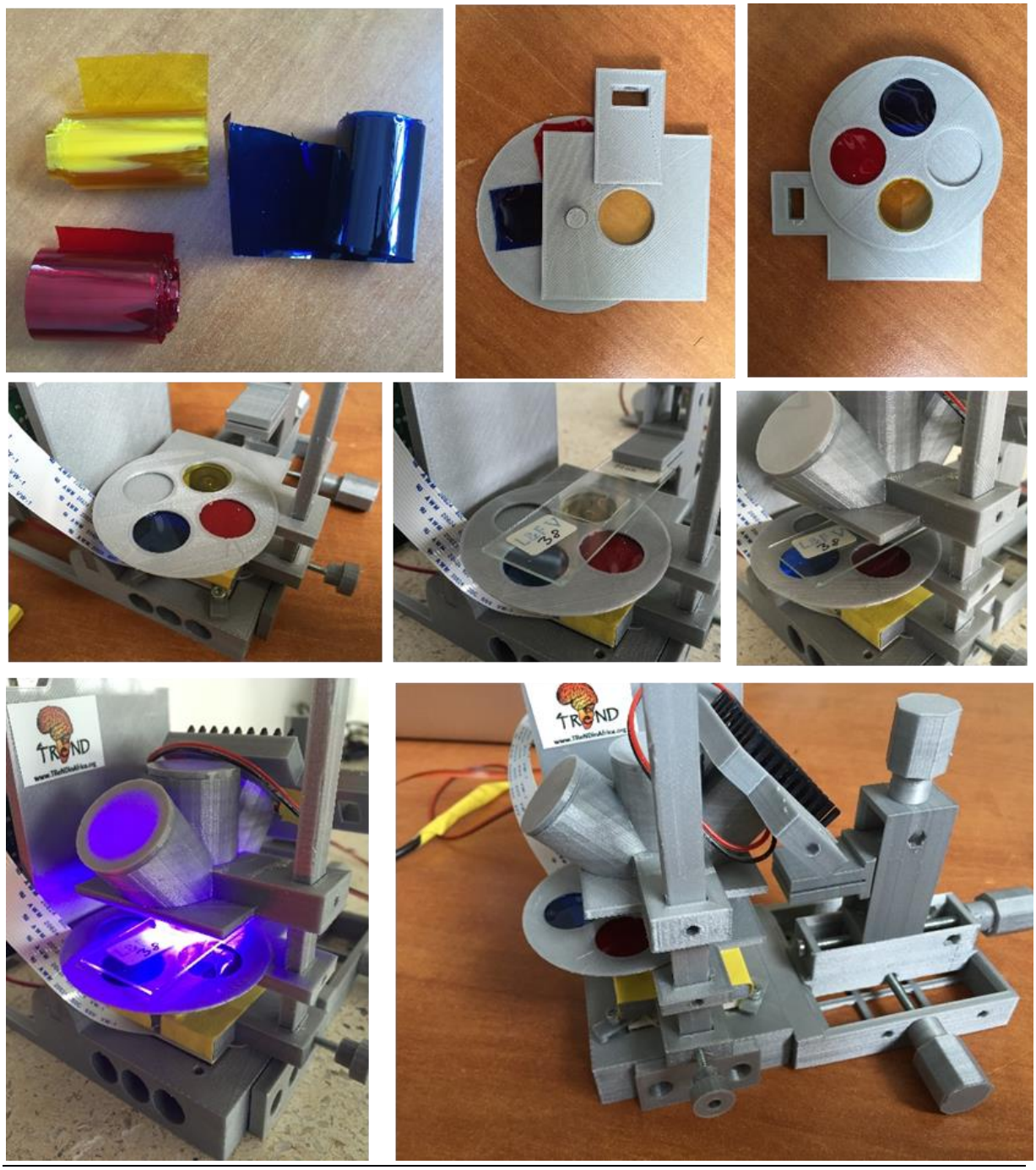
Begin by attaching (+) and GND cable connections (10-15 cm) to the high-power excitation light LED (not shown). Next, attach the reflective collimator on top of the LED and fix with a drop of glue and fix a small piece of the blue Roscolux filter (“Baldassari blue”) on top of the collimator (seen only in subsequent image here). Take care that the glue does not spread over the filter. Now add a drop of heat-conductive paste to the back of the LED and glue everything onto the flat face of a heat-dissipater, assuring that everything is as well centred as possible (note that in the images provided, the cables are not attached yet by accident – this makes it much harder to solder on the contacts as the heat-dissipater will dissipate the heat from the

soldering iron). Next, slot the excitation LED mounted on top of the dissipater into its mount as indicated, and slot everything on top of one of the tubes of the main unit that is angled at 45°. Proceed identically with an optional second excitation LED light source for the other 45° tube (not shown). Any remaining open tube ends should be plugged with the lids provided to block light. Note that it may be handy to leave the vertical tube optionally open/closed as this can be used to transmission illumination using the room lights or the LED ring in tandem with the fluorescence mode - so don't glue the lid in place. Excitation LEDs are attached to the slots provided on the PCB. *WARNING: The UV LED is very bright and poorly visible to the human eye (i.e. it is much brighter than it looks!) – directly looking into it can seriously damage your retina!*



For the filter-wheel, first cut small pieces of Roscolux filters appropriate to just cover a single hole of the wheel, and glue them in place on the underside (the side with the pin) as indicated. Take care not to smudge the filter that covers each hole with glue. Next, plug the filter-wheel's pin into the mounting adaptor as indicated. If the filter wheel comes loose, try slightly heating the pin as it comes through the mounting adaptor's pin hole as to slightly expand its end and thus fixing it in place (e.g. using lighter, or by holding the soldering iron near it without touching). Be careful not to burn the filters which can happen quite easily in this step. If everything is too tight, file down the central pin until it fits well.

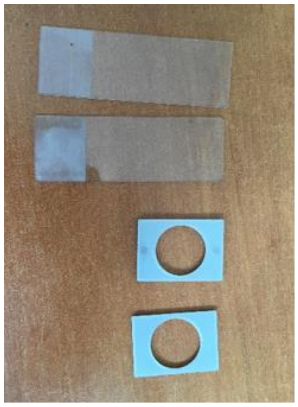
Finally, attach thumbscrews to both excitation mount and filter wheel and put everything together.

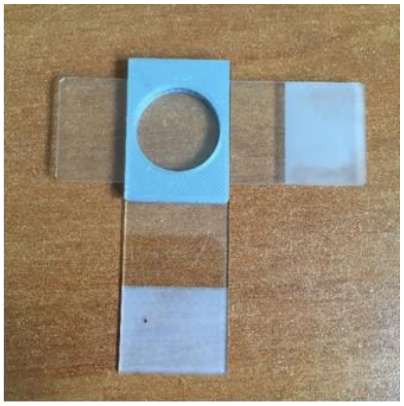


**Petri dish mounts and fly behavioural chamber**

The lid-part of standard small and medium sized petri dishes should easily slide into the 3D printed petri-dish mounts and should stay firmly in place without need for further fixation. Attach to mainframe for the FlyPi using thumbscrews as usual.

The Fly-behavioural chamber consists of 2 standard glass microscope slides and 2 3D printed parts as indicated. Simply glue the back face of one of the 3D printed parts to one of the glass slides, then glue the other 3D printed part on top as indicated. This should create a slot through which the other glass slide can be optionally inserted to form the lid of the chamber. Note that the chamber is made from glass rather than plastic as not to filter blue/UV light which may be useful e.g. for optogenetic stimulation. Note also that the STL file collection also contains a design for equivalent mounting plates for double-size microscope slides.




## Any issues with 3D printed parts?

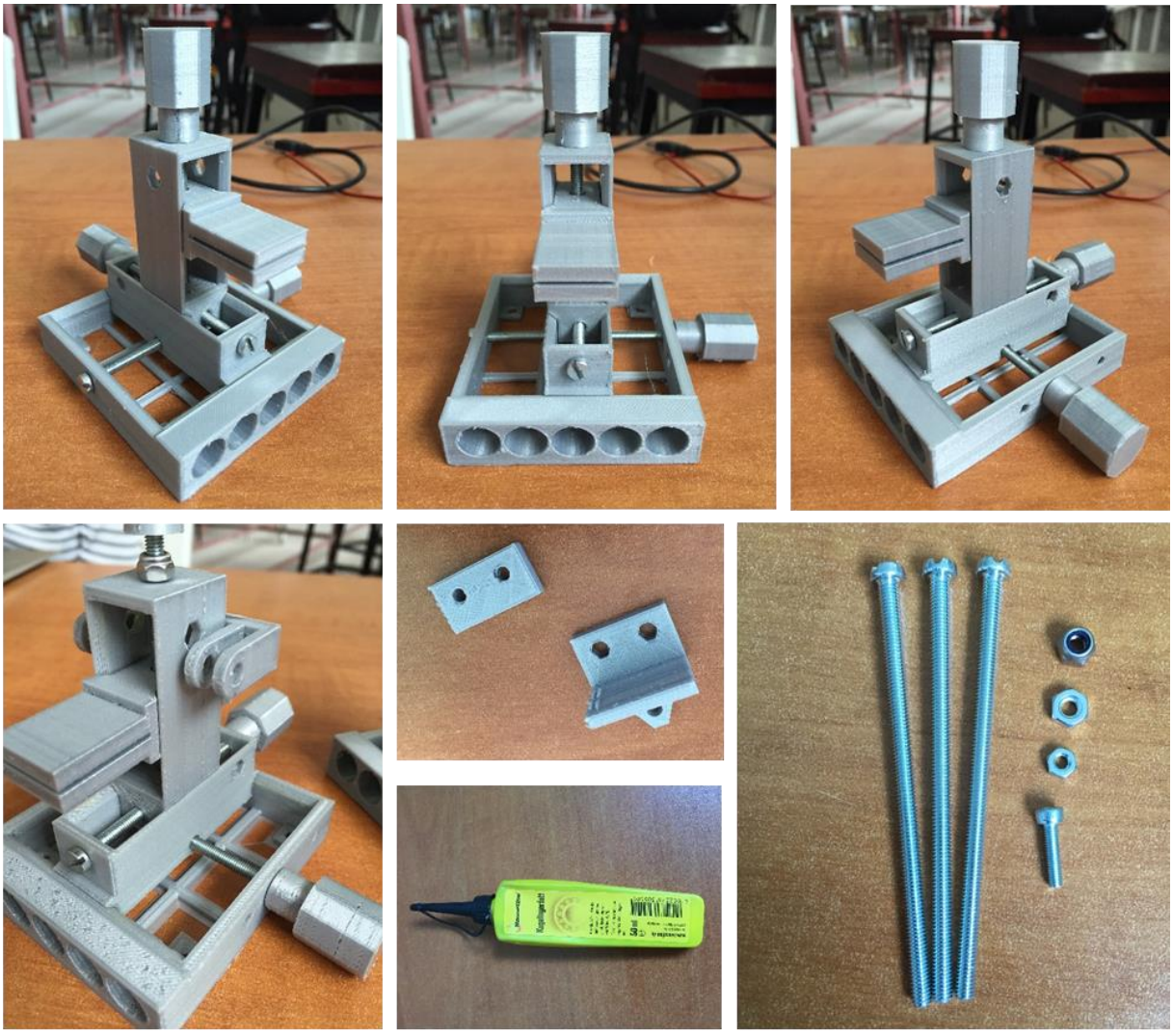
If any 3D printed parts do not seem to fit etc, it can be worth looking at the original OpenSCAD file (Need to download OpenSCAD which is freely available at www.openscad.org ). In the code, each piece is built from a single or multiple “modules” which can be edited. Most key parameters are variables with hopefully self-explanatory names.

One Key parameter is the “Tol” one in line 38. This is a global parameter that is added to all sliding parts, and defines, in mm, the size of every sliding gap. If the pieces are all too tight, try increasing this to e.g. 0.1 (or even 0.2 for a less accurate printer). After any modification, the model needs to be compiled again (F6) and exported as STL for printing. Note that there are several switches in the top that allow switching on (1) and off (0) individual pieces of the model.

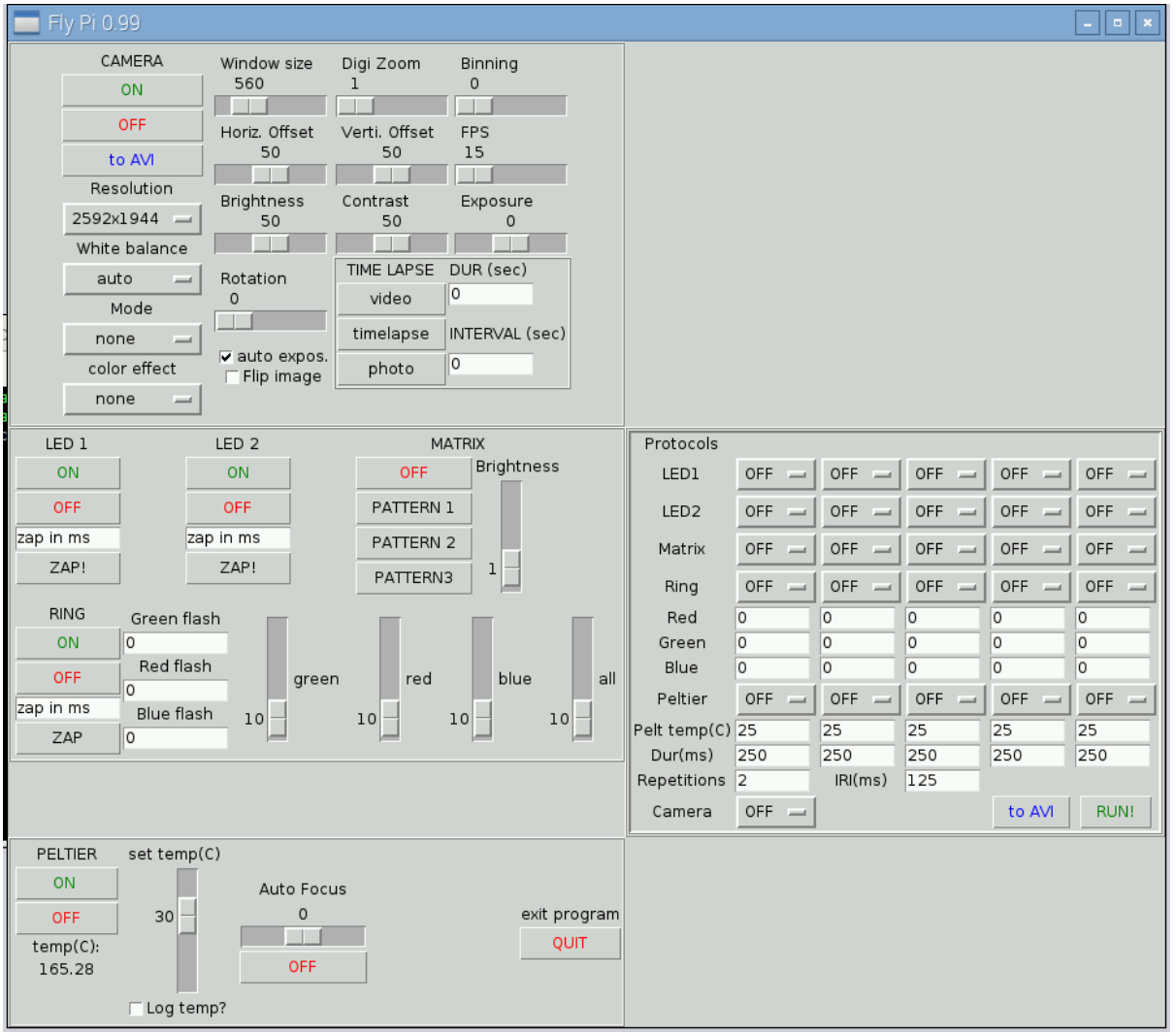
## Micromanipulator assembly and motorisation

This is described in detail elsewhere:

https://openlabwaredotnet.files.wordpress.com/2015/11/manipulator-assembly-instructions.pdf



# Operating the Graphical User Interface (GUI)




## Activating / Deactivating individual parts of the GUI

The GUI is organised in a modular fashion: Camera, LED1, LED2, LED Ring, LED Matrix, Peltier, Auto Focus and Protocols. Each module can be activated and deactivated as required. To do so, edit the file named flypiApp.py and find the variables named xxxxFlag (e.g., cameraFlag). Setting them to zero/one will switch off/on a module, respectively. For users who want to create their own modules, a new file containing a Python class with all necessary methods and variables should be generated, as well as a similar variable in the XXX file, as described above. For more details, please refer to “mock_up_module.py”

## Camera control

The GUI provides for simple access to most built-in camera control options. Some of them, such as Resolution, White Balance, Mode and Colour Effects are controlled via “dropdown” buttons which continuous parameters are set via sliders. This module also provides a subsection for recording data from the camera: The button “video” records “h264” video keeping all settings changed by the user, except resolution, as there are limitations on the PiCamera library, when using maximal resolution (2592X1944). The duration of the recording

is set in the box “DUR (sec)”. The “timelapse” button takes still photographs every “INTERVAL (sec)” seconds for a total of “DUR (sec)” seconds. The “photo” button takes one snapshot. In both cases (timelapse and photo) images are recorded in “jpeg” format

---

## LED control

There are 4 different LED related panels.

The first two control the high-power LED slots as used e.g. for the fluorescence module. An “ON” button turns the respective LED on until the “OFF” button is pressed. For automated timing a “Zap” control is present, where the user can define the LED activation time (in milliseconds) that is counted down upon pressing the “Zap” button. Notably, these ports are general purpose and could be used to control any <=5V element as required (motors, lights etc.). To set a port to the “right” voltage, it may be necessary to switch the resistor positioned next to each port (cf. circuit diagram on GitHub).

The LED-Ring panel has the same interface as the high-power LEDs, with the addition of extra controls to adjust the intensity of each colour channel (Red/Green/Blue, for spectra see Fig. 5B). In the “Zap” controls, this allows for the ring to be on with a specific colour combination, and have a different colour combination “zapped” for a specified amount of time. To control the colour combination for the steady ON state, four different sliders are provided, one for each of the RGB channels and one that controls all channels together. To control the colour of the light “zap”, the user should input numbers in between 0 and 255 into the text boxes (“red flash”, “green flash” and “blue flash”). Notably, Adafruit produces many different shapes and sizes of these so called “NeoPixel” LED arrays, most of which are directly compatible for FlyPi without reprogramming (e.g. all rings and the “diamond”).

The Matrix panel has a different configuration. Here an “OFF” button is present, together with 3 “ON” buttons for different pre-programmed patterns, and a slider that controls the brightness of all LEDs present in the Matrix. Each “pattern” button is pre-configured to generate a specific light pattern, which can be static or dynamic. To change the pre-programmed patterns, update the annotated pattern-related code segment near the top of the Arduino sketch and re-upload it to the Arduino. For more information on how to programme Matrix patterns, the user is referred to the excellent tutorial section on the Adafruit website ().

---

## Peltier control

The Peltier control consists of “ON” and “OFF” buttons, a slider to set the desired temperature and a checkbox for enabling/disabling temperature log (into an asci file). Once the “ON” button is pressed, the temperature command on the slider is transmitted to the Arduino. The temperature slider can be changed at any time, and the final temperature on the Peltier surface is held by a closed loop system between the Peltier current control (an H-Bridge in this case) and the temperature sensor installed on top of the Peltier. If the “OFF” button is pressed, current to the Peltier is interrupted, but temperature changes still occur as the device temperature equalises with room temperature. If the checkbox “Log temp?” was marked during the Peltier use, a file will be created and the readout of temperature sensor.

*Note1: if the Peltier polarity is switched (i.e. if it heats rather than cools and vice versa) the red and black wires as plugged into the PCB need to be swapped.*

*Note 2: As one side of a Peltier cools by a certain degree, the other side will inevitably heat up to a greater degree, leading to an overall increase in temperature. It is therefore possible*

*that at some point the Peltier will fail to cool to the desired temperature. If this happens, make sure a fan is positioned under the Peltier as shown in Fig. 6 – this will help to dissipate the heat. If this is still not sufficient, consider the ambient room temperature – putting the entire system in a cooler spot can dramatically increase overall performance.*

## Auto Focus

Auto Focus controls a continuous rotation servo motor attached to a gear system connected to the lens above the Pi Camera. Moving the slider to one direction will make the motor turn clockwise, and in the opposite direction counter-clockwise. The more the slider is displaced, the faster the motor movement. Since the camera does not require the motor for focusing and operating, users can adapt this module to their own specific needs (e.g., opening and closing physical barriers or to drive an optokinetic drum).

## Protocols

This module is designed for automating stimulation and recording procedures. Each column is executed sequentially and for the time set in “Dur(ms)”. Once all five columns have been executed, the program waits for the time specified in “IRI(ms)” and thereafter repeats from the first column for the number of loops defined in the “Repetition” box. If “camera” is set to “ON” a video will be taken throughout the execution (to avoid system crashes, the user is advised to make sure that the SD card has enough space to record all data). If “camera” is set to “OFF” the protocol is executed without recording video, which is useful for testing or recordings using external devices. In case the Peltier is being used and the “log temperature” checkbox in the Peltier module is checked, a text file containing a timestamp and the read temperature will also be created.

*Note: As the protocol execution is controlled by the Arduino microcontroller (rather than the RPi), time precision achieved at the general purpose LED ports is the order of microseconds. However, since the LED/Matrix ring relies on a multiplexed signal from the Arduino which is executed through the Adafruit library only at 100s of Hz, its precision is slightly lower (~5 ms).*