

OzturkLab Report

Peker Milas

March 9, 2021

Contents

1	Self-recovering Wired Ethernet Connection	2
1.1	Preparation and Software Installations	2
1.1.1	Chrome Installation	2
1.1.2	Anaconda Installation	2
1.1.3	Selenium Library Installation	3
1.1.4	Selenium WebDriver Installation	5
1.2	Implementation-Coding	6
1.3	Implementation-Automatization	7
2	Photoluminescence (PL) Setup	12
2.1	Reflecting Light Microscopes Overview	12
2.2	Confocal Systems Overview	14
2.3	Confocal Reflecting Light Microscope for PL Measurements . . .	16
2.3.1	Excitation, Zone 1	18
2.3.2	Emission, Zone 2	19
2.3.3	Brightfield, Zone 3	20
2.3.4	Objective, Zone 4	21
3	Mask Design Walkthrough for ODMR Antenna	21
3.1	Preparation and Software Installations	21
3.1.1	Chrome Installation	21
3.1.2	Anaconda Installation	21
3.1.3	Selenium Library Installation	22
3.1.4	Selenium WebDriver Installation	24
3.2	Implementation-Coding	25
3.3	Implementation-Automatization	26

1 Self-recovering Wired Ethernet Connection

This is a software based workaround for our lab's computers to keep the in the wired network. To implement the solution, following software is required to be installed;

- **Chrome** web browser
- **Anaconda** or another Python distribution.
- **Selenium** libraries for Anaconda to be able to interface with Selenium.
- **Selenium** Web Driver

1.1 Preparation and Software Installations

1.1.1 Chrome Installation

To install Chrome, go to following web page and download the suitable installer version for your operating system (OS).¹

1.1.2 Anaconda Installation

To install Anaconda, go to web page and download Anaconda installer suitable for your OS.² During the installation, at some point installer will give you an option to add Anaconda to system path (\$PATH) as shown in Figure21.

¹<https://www.google.com/chrome/>

²<https://www.anaconda.com/products/individual>

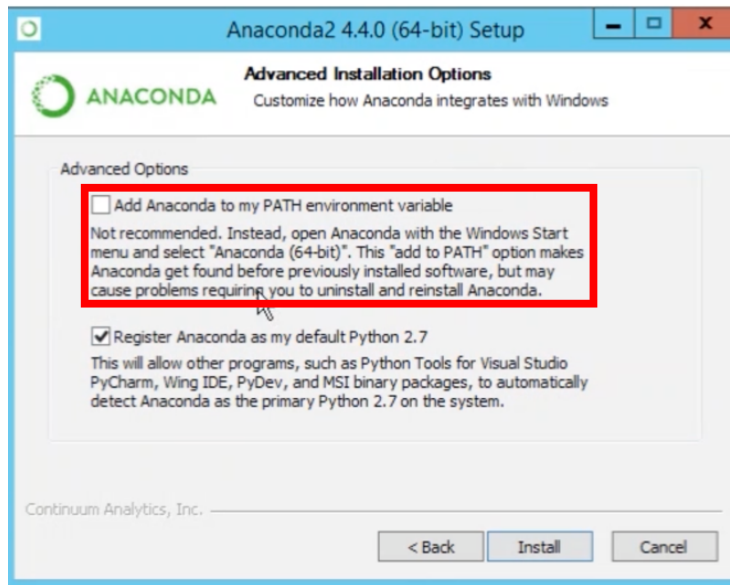


Figure 1: Installation step for adding Anaconda to system path.

Here, go ahead and click the box on the left to add Anaconda to system path. This will make Anaconda your system-wide Python library. Its issues are not going to be discussed here for simplicity.

1.1.3 Selenium Library Installation

To install Selenium libraries for allowing Anaconda to interface with Selenium web driver you can use Anaconda Prompt as shown in Figure22.

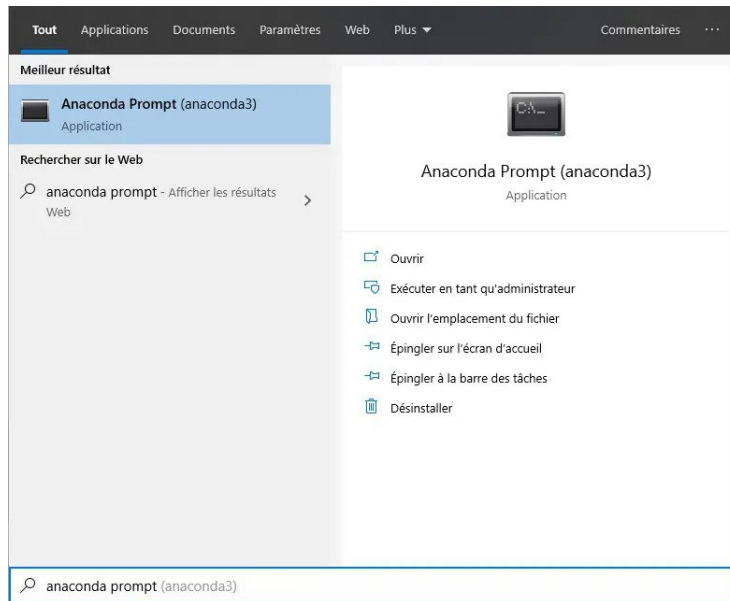


Figure 2: Opening up Anaconda Prompt.

In Anaconda Prompt, you can simply type **conda install -c conda-forge selenium** and click **Enter**, as shown in Figure23.

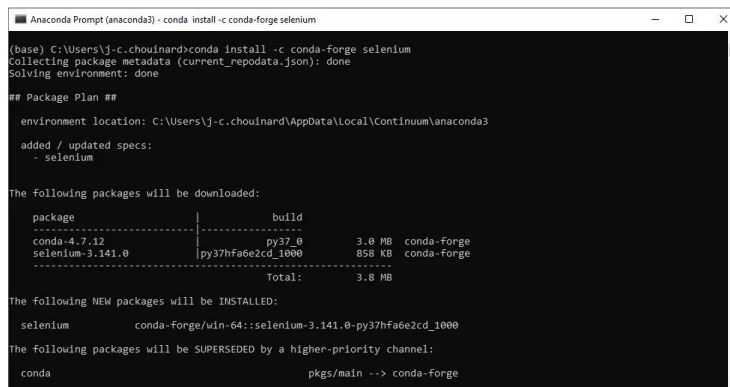


Figure 3: Selenium library installation for Anaconda.

Once Anaconda figures out where Selenium library is and possible additional required libraries, it will ask your confirmation as shown in Figure 24. Here you can type **y** and click **Enter**.

```
Proceed ([y]/n)? y

Downloading and Extracting Packages
conda-4.7.12 | 3.0 MB | ##### | 100%
selenium-3.141.0 | 858 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(base) C:\Users\j-c.chouinard>
```

Figure 4: Selenium library installation confirmation screen.

1.1.4 Selenium WebDriver Installation

To install Selenium WebDriver for allowing Selenium-Anaconda library to control the web browser (**here we use Chrome!!!**), go to the following web page, <https://sites.google.com/a/chromium.org/chromedriver/> as shown in Figure 25.

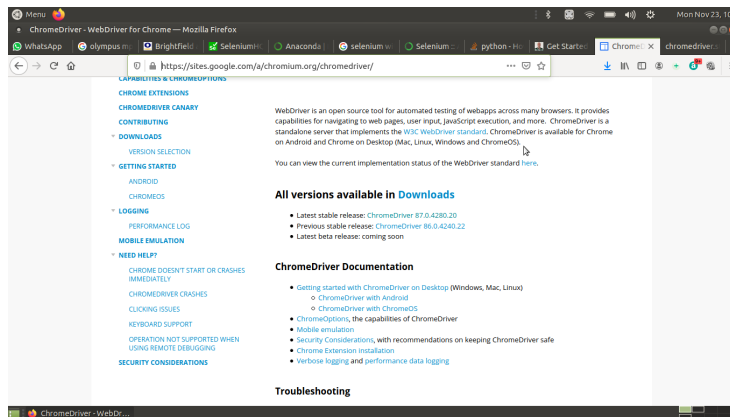


Figure 5: Selenium library installation confirmation screen.

Within other options, click on the **Latest Stable Release**. This will direct you another web page with multiple files you can download as shown in Figure 26. **Here, you have to download 32-bit driver!!!**

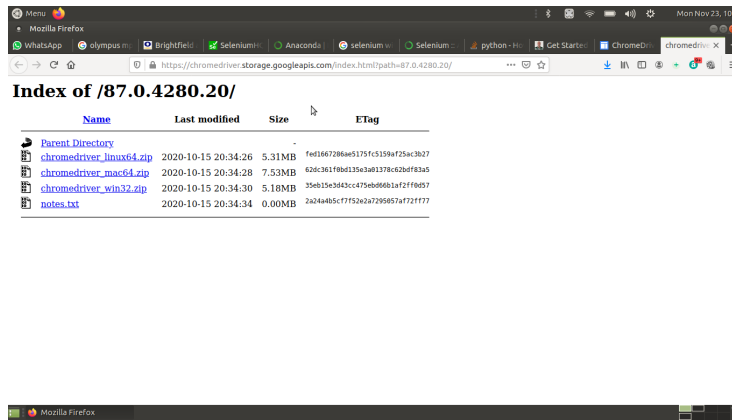


Figure 6: Selenium library installation confirmation screen.

Once download finishes, you can unzip the the webdriver file with any software you desire.

1.2 Implementation-Coding

To invoke Selenium I wrote Python Script. An overview of the code is as following;

1. **Lines 1-4:** Imports essential libraries.
2. **Lines 6-8:** Pings Google's DNS server to check whether the PC is connected to wired network.
3. **Lines 10-11:** If output of lines 6-8 are both indicating connection proceed to the end of the code.
4. **Lines 12-27:** If PC is not connected; **Line 13:** Run Chrome driver
Line 14: Send Chrome to university's Log-In page. Here "https://secure-access.morgan.edu/guest/ngnauth.php?" is a fixed string. "&mac=f8:b1:56:b3:7a:0a&_browser=1" is the MAC address of our machine and browser number which is the default browser.
5. **Lines 17-20:** Fill out the form on university's Log-In page using your username and password.
6. **Lines 21-22:** Click the box to accept the terms.
7. **Lines 23-24:** Click on the Accept button to Log-In.
8. **Lines 25-27:** Wait for the webpage to load to university's homepage then close the browser.
9. **Line 28:** Kill python.exe to stop the process cleanly.

```

1  from selenium import webdriver
2  from selenium.webdriver.common.keys import Keys
3  import time
4  import os
5
6  temp = os.popen("ping 8.8.8.8").read()
7  test_connection1 = temp.find('PING: transmit failed. General
    ↳ failure.')
8  test_connection2 = temp.find('Request timed out.')
9
10 if (test_connection1 < 0 and test_connection2 < 0):
11     print("Already connected. Exiting!!!")
12 else:
13     driver = webdriver.Chrome()
14
15     ↳ driver.get("https://secure-access.morgan.edu/guest/ngnauth.php?
        &mac=f8:b1:56:b3:7a:0a&_browser=1")
16
17     elem1 =
18     ↳ driver.find_element_by_id('ID_formcd662c6a_weblogin_user')
19     elem1.send_keys("Peker.Milas")
20     elem2 =
21     ↳ driver.find_element_by_id('ID_formcd662c6a_weblogin_password')
22     elem2.send_keys("Onder1981")
23     elem3 =
24     ↳ driver.find_element_by_id('ID_formcd662c6a_weblogin_visitor_accept_terms')
25     elem3.click()
26     elem4 =
27     ↳ driver.find_element_by_id('ID_formcd662c6a_weblogin_submit')
28     elem4.click()
29     time.sleep(10)
30
31     driver.close()
32 os.system("taskkill /F /im python.exe")

```

1.3 Implementation-Automatization

As I have indicated above, the provided Python script is meant to execute once. Therefore, to first sense a loss in wired connection and invoke the Selenium for automatic Log-In, we use Windows' **Task Scheduler**. This is done as following;

1. Click on the Windows search bar and type Task Scheduler. This will pop a widow above Windows logo to let you run Task Scheduler which is shown in Figure27

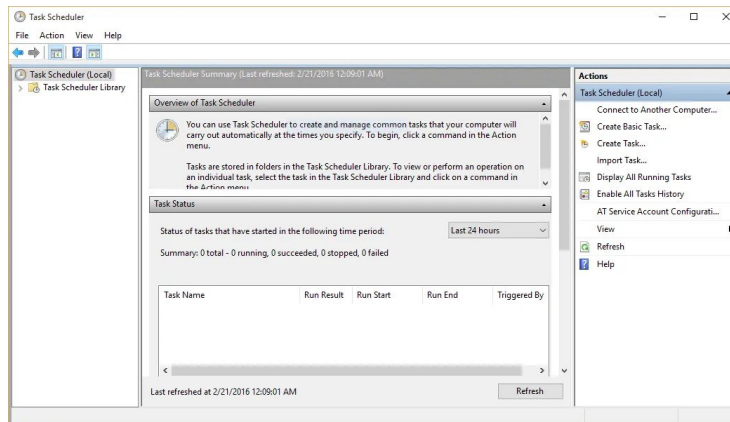


Figure 7: Windows10 Task Scheduler.

2. Click on the "Action→Create Task", as shown in Figure 28

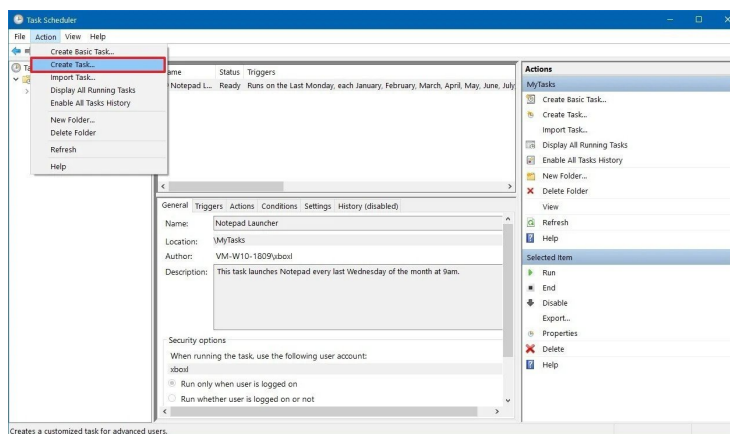


Figure 8: Create a task in Task Scheduler.

3. In popped-up window, on General tab, as shown in Figure29, give a unique name to your task and describe it for future reference.

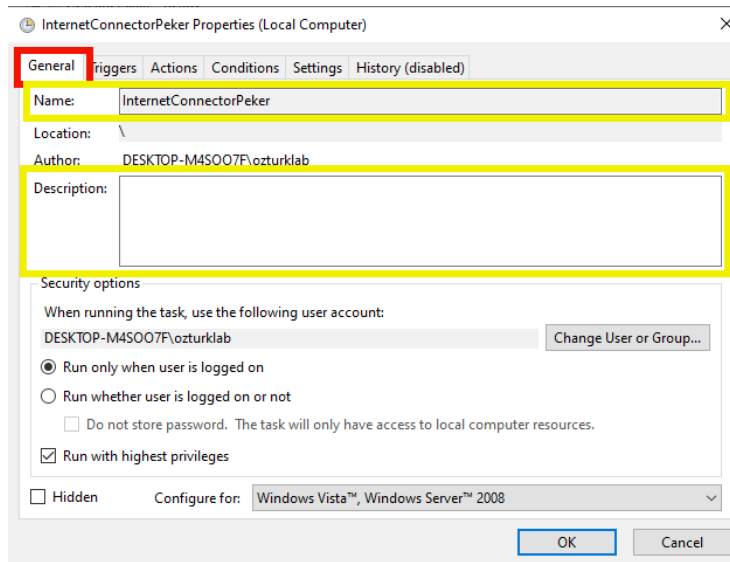


Figure 9: Name and Description fields under General tab.

4. On the Triggers tab, as shown in Figure30, click on New. In popped-up window, set the Start time and date. Also make your task runs for One time. Let the task repeat at specified times (here every 30 minutes) and keep it run indefinitely.

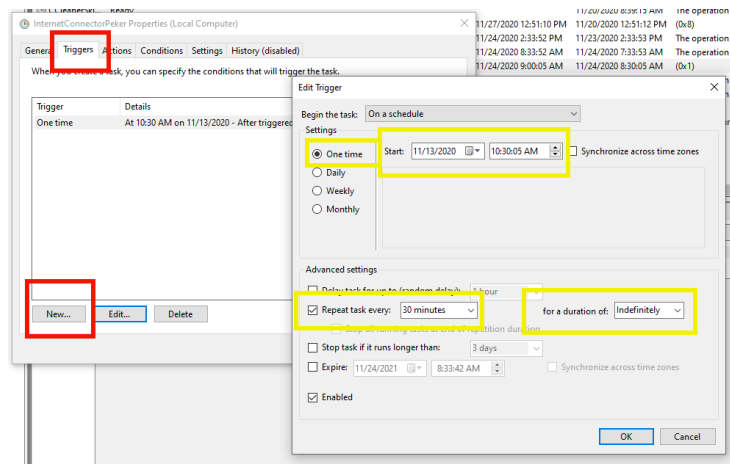


Figure 10: Timer and repetition related fields under Triggers tab.

5. On the Actions tab, as shown in Figure31, click on New. In popped-up window, make Action Start a program. In Programs/script field, add the

location of your python executable (python.exe). In Arguments field, add your python script's name. In Start in (Optional) field, add the location of your python script.

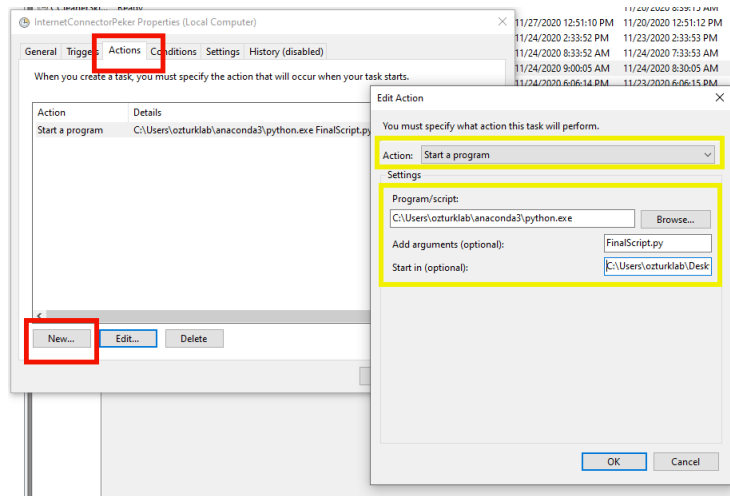


Figure 11: Script and arguments related fields under Triggers tab.

6. On the Conditions tab, as shown in Figure32, most of the fields are related with hardware such as whether you have a laptop which can run on battery or not. Therefore you can simply keep it as shown in Figure 32

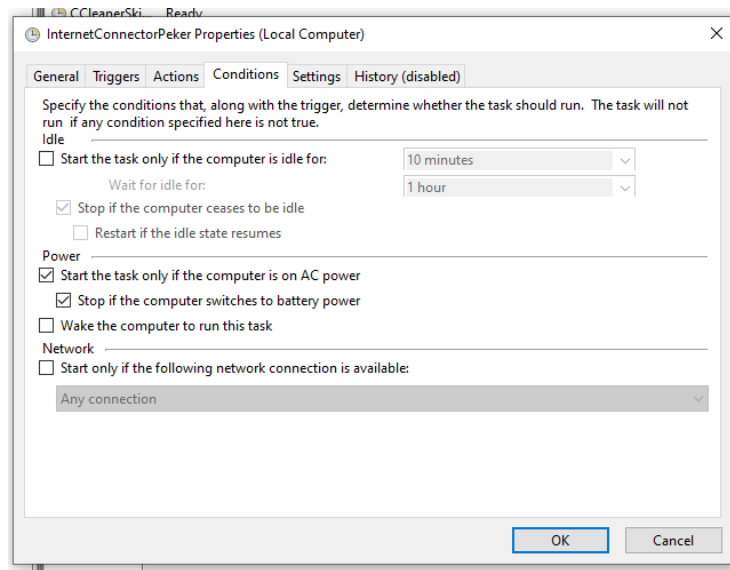


Figure 12: Already prepared Conditions tab.

7. On the Setting tab, as shown in Figure33, you can again set relevant fields as shown in Figure33.

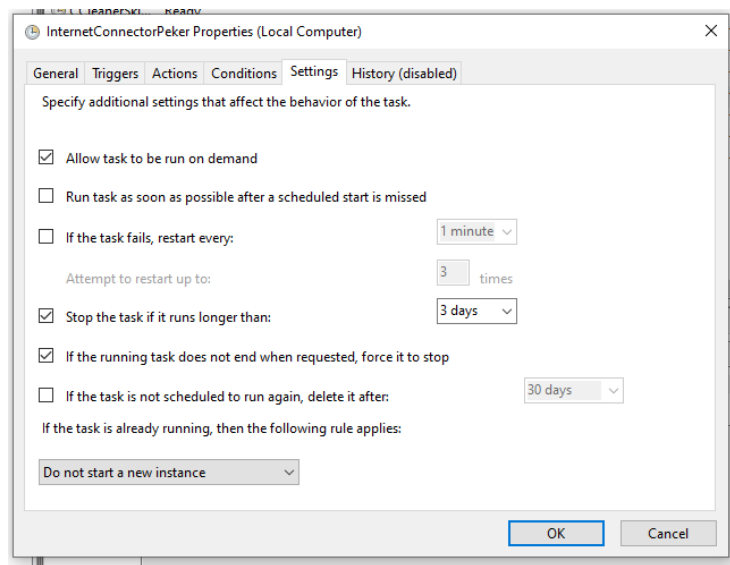


Figure 13: Already prepared Settings tab.

8. Once you are done with setting up your scheduled task, save it and close

its window. This will add your new task to default windows tasks as shown in Figure34. Right-click on your task and run it for the first time. After that point, it can run by itself.

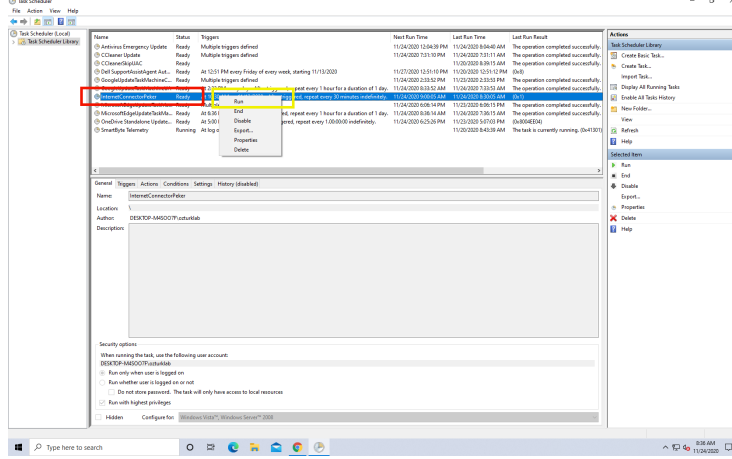


Figure 14: First run of the new task.

2 Photoluminescence (PL) Setup

Photoluminescence is the emission of light by the excited molecules and/or molecule clusters. Here we are intended to observe photon emission from the defects within crystalline structures such as diamond and cubic boron nitride. Given that these materials known to be opaque in visible light range, we are building a **reflecting light microscope**[refhere]. Although wide-field excitation (similar to epifluorescence [refhere]) based emission measurements from an ensemble of defects have their own importance in characterization, we pursue investigating properties of individual defects/ emitters. This requires our reflecting light microscope to be a **confocal system**.

2.1 Reflecting Light Microscopes Overview

On typical an upright or inverted microscope, illumination light comes from a direction relative to sample surface, while the collection or imaging is done in the opposite direction. Therefore in terms of optical components, illumination and imaging sides differ in those systems. For example in illumination side a special lens called **Condenser** focuses the light on to the sample while in the imaging side another special lens called **Objective** lens collects the light coming from the sample. These special lenses and the typical upright and inverted microscope configurations are shown in Figure15.

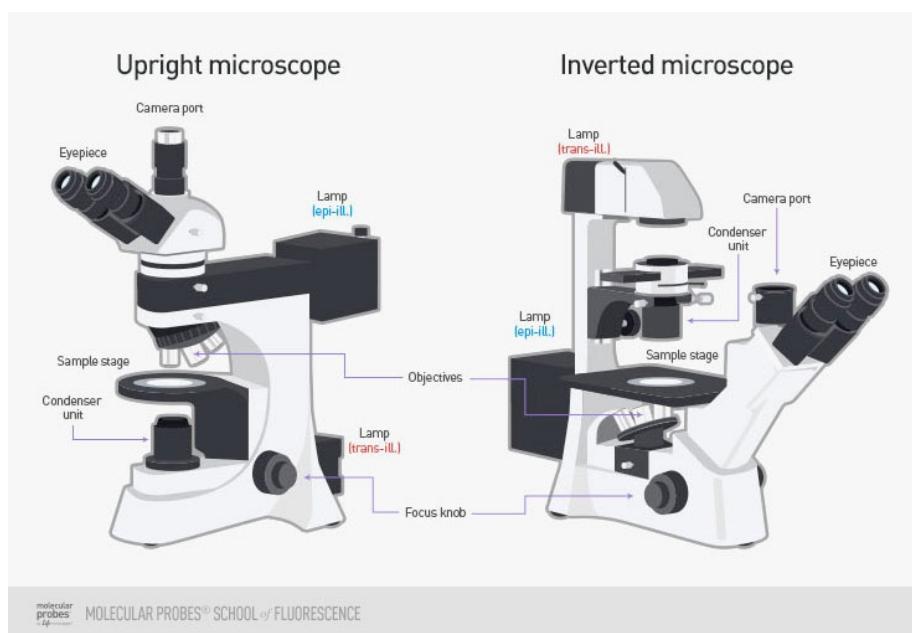


Figure 15: A typical upright (**left**) and an inverted (**right**) microscope structures indicating the positions of special lenses (**Condenser and objective**).

On the other hand, in a reflecting microscope illumination and imaging are done through the same optical path or by same optics. As it is shown in Figure 16, shared optical path allows one (1) additional port for imaging and/or illuminating the sample on a typical reflecting microscope. This port is very frequently assigned to white-light (**bright-field**) imaging through a camera, so it is called the camera port. As a result any additional illumination and/or imaging on top of the default system, require a custom design optical system with the ability to finely couple with the existing microscope.

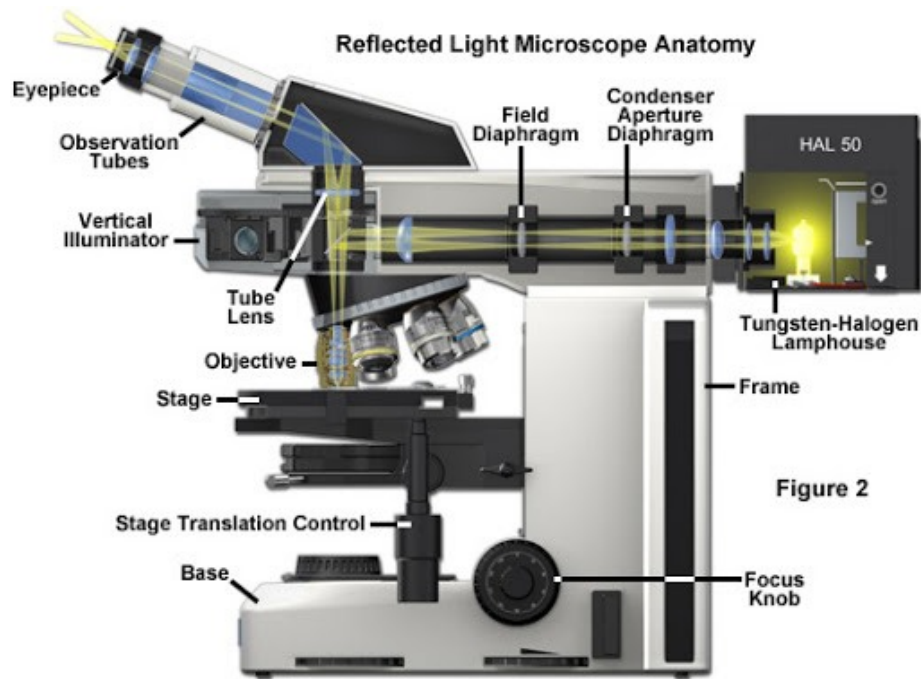


Figure 16: Schematics of a typical reflecting microscope with light path and internal optical components.

2.2 Confocal Systems Overview

A **confocal system** is an imaging system that utilizes a spatial filter (or simply a pinhole) before the imaging component for strictly matching the illumination volume with the observation volume as it is shown in Figure17. Because of very likely chromatic aberrations ³ through the imaging components, usually illumination is done by a single color light source such as a laser. The pinhole suppresses the out-of-excitation volume light dramatically, so it increases the resolution of imaging system, Figure18. REF HERE!!!!

³<https://www.microscopyu.com/tutorials/chromatic>

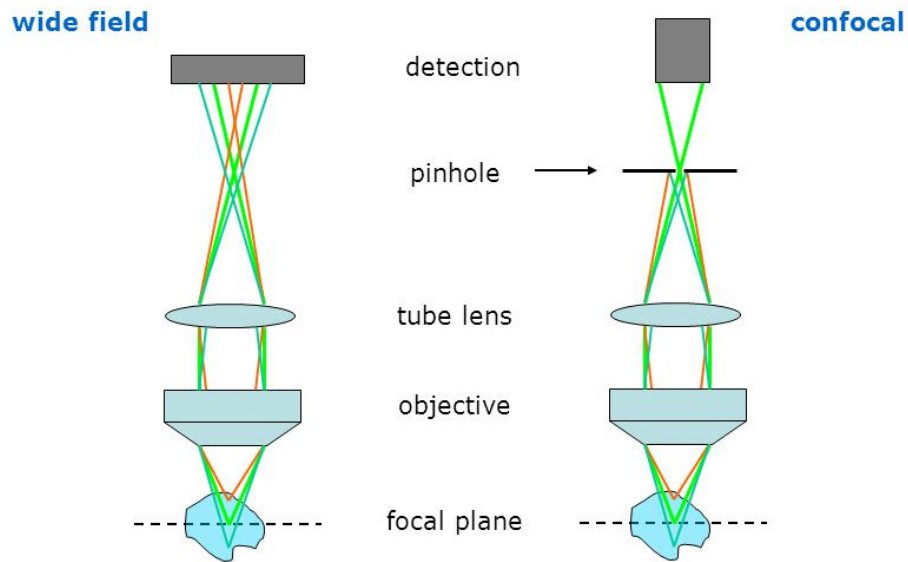


Figure 17: Simplified structure of a typical widefield (**left**) and a confocal (**right**) imaging system with spatial filter. ADD REF-FOOTNOTE HERE!!!

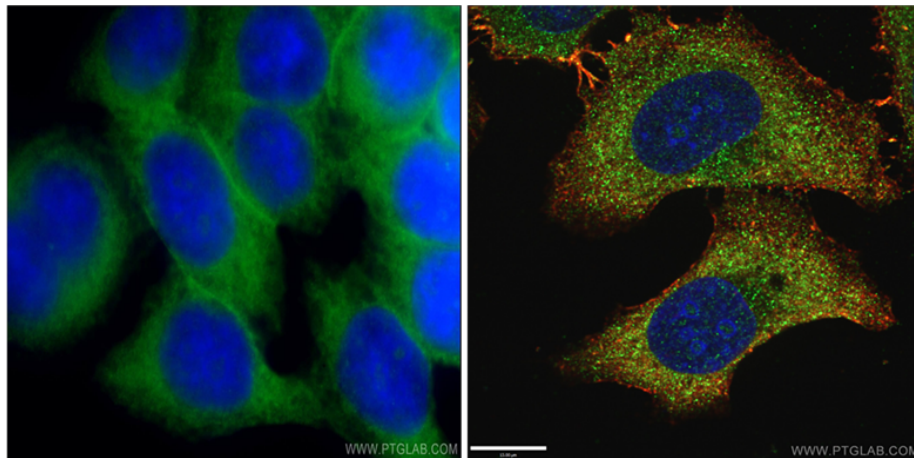


Figure 18: Imaging of HeLa cells with a typical widefield (**left**) and with a confocal (**right**) imaging system. ADD REF-FOOTNOTE HERE!!!

Microscope manufacturers very frequently incorporate additional openings (called **microscope ports**) on sides, front and even at the bottom of microscope bodies which allows the operator to couple additional imaging and illumination

systems into the microscope. Although this is a very common practice in the case of transmission microscopes (upright and inverted as well), reflecting microscopes usually don't have these additional ports, Figure16. Because of missing ports, turning a reflecting microscope is considerably harder into a confocal system compared to transmission microscopes.

2.3 Confocal Reflecting Light Microscope for PL Measurements

Our goal is to observe emission from excited lattice-defects in solids like boron nitride(BN) and diamond[REF HERE!!!]. These materials are not particularly transparent to visible light⁴. Moreover defect formation (particularly in the case of BN) is induced by ion bombardment, so neither the density nor spatial distribution of defects are known prior to a PL measurement. The lack of information on defect positions introduces the requirement of active search for defects within the sample volume, so a confocal imaging system with the capability of scanning. To overcome the refractive index issue and satisfying the confocality in imaging, we built a confocal reflecting light microscope as shown in Figure19.

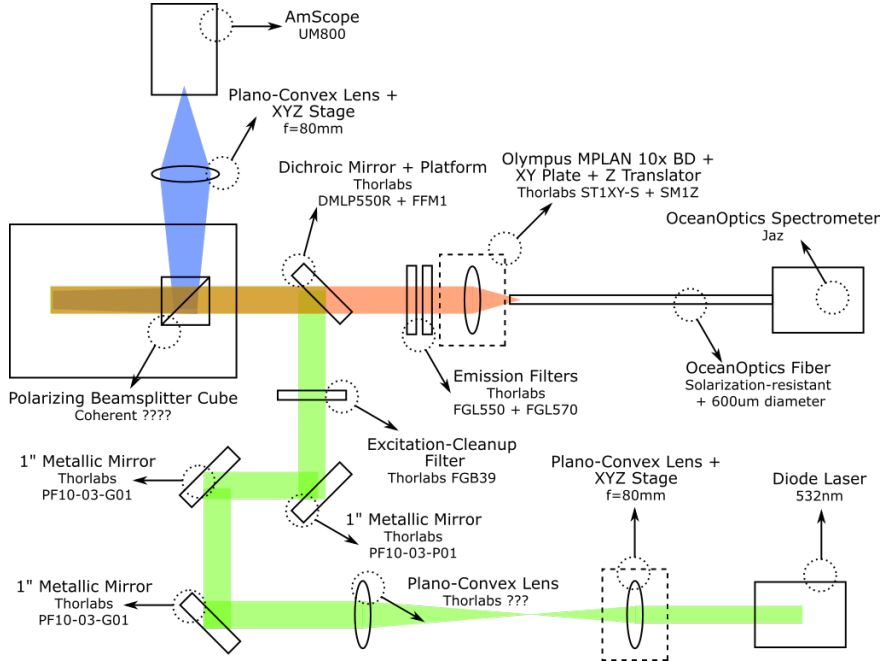


Figure 19: Our homemade confocal reflecting microscope for PL measurements.

Although the purpose of individual parts can not be explained accurately by

⁴https://en.wikipedia.org/wiki/Boron_nitride

ignoring their interactions with the remaining system components, it is a common practice to divide the whole system into 4 essential zones for educational purposes as in Figure20.

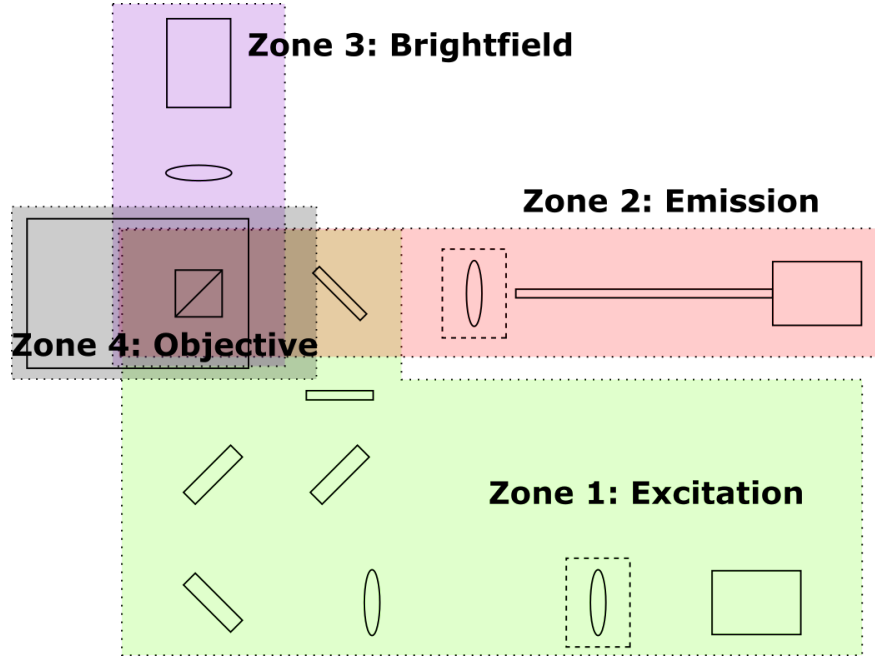


Figure 20: Zones in PL microscope.

To briefly describe these zones;

1. **Excitation, Zone 1 :** This is the zone where the excitation light is directed into the microscope. The excitation source is frequently a laser, but there are systems which prefer broadband light sources such as UV lamps. In this zone, beam characteristics of excitation light (i.e. size, divergence, polarization, etc.) are adjusted to match the requirements imposed by other zones.
2. **Emission, Zone 2 :** This is the zone where the emission or reflection from the sample is collected and analyzed by a detector. The detector can be any photon sensitive device (i.e. PMT, APD, CCD camera, etc.). Based on the preferred coupling mechanism in the sensory device, characteristics of the emission from the sample are adjusted similar to the case the adjustment of excitation light.
3. **Brightfield, Zone 3 :** This is the zone responsible to visual inspection of the sample surface and excitation/emission light profiles on the sample. Its mere purpose is to create an image of the sample surface and everything in near proximity on a camera and allow guided tracking.

4. **Objective, Zone 4 :** This is the zone where the primary imaging lens called objective lens resides. As a result, it is the core of the whole optical system and couples with all other zones. In other words, optical components in other zones are chosen/built essentially for matching the objective lens to utilize the whole measurement system at its maximum efficiency.

As in any other optical system, efficiency of imaging in this system is strongly correlated with match ratio of all these zones. Although it appears to be the Objective (Zone) 4 is the central piece that all other zones required to be matched, there are matching requirements for among other zones too. As a result, proper description of the microscope require deeper understanding of all zones.

2.3.1 Excitation, Zone 1

In terms of the structure, this zone directly couples into Objective (Zone 4) and indirectly couples into Emission (Zone 2). In terms of the alignment, it directly couples into Brightfield (Zone 3). These couplings and the required matching conditions are as following;

2.3.1.1 Direct structural coupling between Excitation and Objective Zones

This coupling is dictated by the three essential **optical design rules**;

1. Excitation light going into the Objective has to be collimated, so it should not be converging or diverging in free space for a long enough distance. On the other hand, the level of collimation is very frequently not perfect, because of the real/non-ideal objective lenses. To address this we placed a telescope in the path of excitation light as shown in the Figure 19. In addition, the first lens in the telescope is positioned on a XY-stage to finely tune the convergence/divergence of the excitation light.
2. Propagation direction of the incoming light to the objective lens has to be parallel with the surface normal of the objective lens. A well aligned excitation beam removes undesired distortions in the structure of focused light after the objective lens. This is addressed by introducing two 1" mirrors to the light path right before the excitation- cleanup filter. As it is shown in the Figure19, the mirror closest to the filter is responsible for angular change while the one before that mirror is responsible for translation of the excitation beam, both relative to the objective lens' surface normal.
3. The ratio of the diameter of excitation light and the diameter of back-aperture of the objective lens (**aperture filling ratio**) has to be known and fixed to a value depending on the experimental needs (i.e. tight focus, scanning beam, etc.). previously mentioned telescope not only sets the angular dispersion of the laser but it also adjusted the beam diameter. By using a 200mm focal lens in the telescope, we set the beam diameter slightly larger than the back-aperture of the objective lens.

2.3.1.2 Indirect structural coupling between Excitation and Emission Zones

This coupling is dictated by the confocality of the system. To be more precise, while excitation zone is describing the input excitation beam, it also defines the distance and size of the focused beam on the front-focal plane of the objective lens. As it was indicated in Section 2.2, spatial filter (pinhole) position and its size has to be adjusted according to the focus position and size.

2.3.1.3 Direct alignment coupling between Excitation and Bright-field Zones

In the case of a conventional microscope, optics in the brightfield zone need to be aligned prior to sending additional input light. This is called Köhler illumination⁵ which is not fully implemented in our microscope. Instead we illuminate the sample surface with a ring of LEDs around the objective lens housing and collect the reflecting light either with a camera or with an additional long focal length lens. Purpose of the long focal length lens is for the adjustment of field of view (FOV⁶). Therefore before adjusting the focal position of the laser in the front focal plane of the objective lens, we try to see the image of the sample surface. Once a clear image is observed, we adjust the convergence/divergence of the laser beam to place the laser focus on to the image plane.

2.3.2 Emission, Zone 2

In terms of the structure, this zone directly couples into Objective (Zone 4) and indirectly couples into Excitation (Zone 1). In terms of the alignment, it directly couples into Brightfield (Zone 3). These couplings and the required matching conditions are as following;

2.3.2.1 Direct structural coupling between Emission and Objective Zones

This coupling is dictated by the objective lens being non-ideal or real. [ADD A PICTURE!!!]. Even though microscope objectives are called infinity-corrected⁷, like all real lenses they have two focal planes namely back (BFP) and front (FFL) focal planes⁸. A confocal system uses BFL of the objective lens or its image to spatially filter the emission. In our system the fiber optics cable is used to spatially filtering the emission at a proper position where there is the first image of BFL is formed.

2.3.2.2 Indirect structural coupling between Emission and Eexcitation Zones

⁵<https://www.olympus-lifescience.com/en/microscope-resource/primer/anatomy/kohler/>

⁶<https://www.edmundoptics.com/knowledge-center/application-notes/imaging/understanding-focal-length-and-field-of-view/>

⁷<https://www.olympus-ims.com/en/microscope/terms/feature15/>

⁸[https://en.wikipedia.org/wiki/Cardinal_point_\(optics\)](https://en.wikipedia.org/wiki/Cardinal_point_(optics))

Although this coupling is explained in Excitation section, it is worth mentioned that apart from the coupling through objective lens' focal plane positions Excitation Zone sets the operable size range for spatial filter by defining the excitation beam profile. The coupling considered to be crucial for the resolution of the whole microscope, so a very detailed work is provided in literature.[REF HERE!!!]

2.3.2.3 Indirect alignment coupling between Emission and Bright-field Zones

Although this coupling is explained in Excitation section, it is worth mentioned that apart from the coupling through objective lens' focal plane positions Excitation Zone sets the operable size range for spatial filter by defining the excitation beam profile. The coupling considered to be crucial for the resolution of the whole microscope, so a very detailed work is provided in literature.[REF HERE!!!]

2.3.3 Brightfield, Zone 3

Given that this zone's couplings with Excitation and Emission zones are already presented in sections 2.3.1 and 2.3.2, its remaining coupling is with the Objective (Zone 4).

2.3.3.1 Direct structural coupling between Brightfield and Objective Zones

A proper brightfield image of the sample plane means that an image formed on the CCD camera chip with minimal optical aberrations⁹. Among which chromatic aberration, field curvature, distortion, and defocus are the ones frequently observed in homemade systems.

- There is not a perfect solution for chromatic aberration because it is related with the objective lens' response to excitation and emission wavelengths. Objective lenses are quite a bit expensive instruments and it is highly unlikely that a research group can effort purchasing unique objective lenses for individual experiments. As a result, a partial solution for chromatic aberration is to image the excitation light focus and the sample surface within the range of operable wavelengths.
- Defocus and distortion are the ones related with the relative axial positions of the objective lens, camera lens, and the camera. These aberrations are required to be corrected by altering lens and camera positions in an ad-hoc fashion on the light propagation axis. It is also useful to have an imaging lens of similar focal length to tube lens of objective lens was designed for.
- Field curvature occurs by the mismatch between objective lens and the remaining lens(es) in the imaging pathway. It is highly likely that the

⁹<https://www.edmundoptics.com/knowledge-center/application-notes/optics/comparison-of-optical-aberrations/>

image forming lens for camera is not exactly same as actual tube lens in a commercial microscope. As a result it is very likely that outer edges of the image will be distorted in a homemade system. As long as the desired portion of the image around center is good enough, there is no need to put extra effort to find an exact copy of tube lens.

2.3.4 Objective, Zone 4

Coupling of Objective (Zone 4) with other zones are covered in sections 2.3.1, 2.3.2, and 2.3.3.

3 Mask Design Walkthrough for ODMR Antenna

This is a software based workaround for our lab's computers to keep the in the wired network. To implement the solution, following software is required to be installed;

- **Chrome** web browser
- **Anaconda** or another Python distribution.
- **Selenium** libraries for Anaconda to be able to interface with Selenium.
- **Selenium** Web Driver

3.1 Preparation and Software Installations

3.1.1 Chrome Installation

To install Chrome, go to following web page and download the suitable installer version for your operating system (OS).¹⁰

3.1.2 Anaconda Installation

To install Anaconda, go to web page and download Anaconda installer suitable for your OS.¹¹. During the installation, at some point installer will give you an option to add Anaconda to system path (\$PATH) as shown in Figure21.

¹⁰<https://www.google.com/chrome/>

¹¹<https://www.anaconda.com/products/individual>

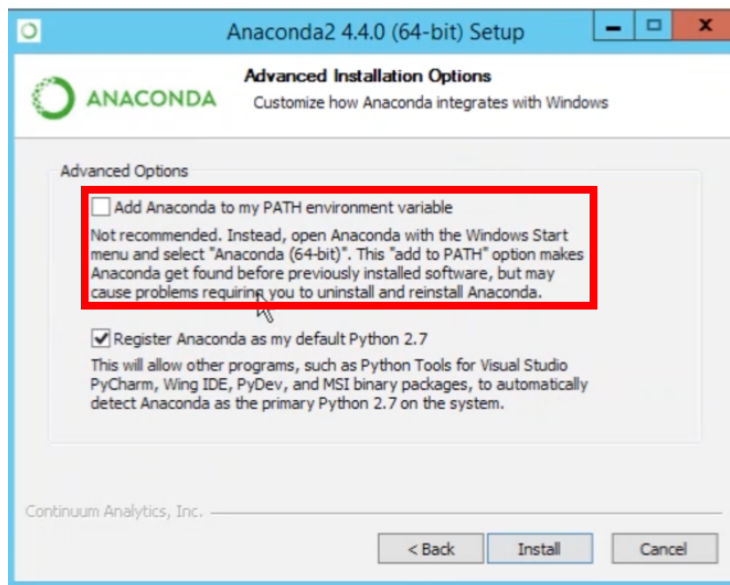


Figure 21: Installation step for adding Anaconda to system path.

Here, go ahead and click the box on the left to add Anaconda to system path. This will make Anaconda your system-wide Python library. Its issues are not going to be discussed here for simplicity.

3.1.3 Selenium Library Installation

To install Selenium libraries for allowing Anaconda to interface with Selenium web driver you can use Anaconda Prompt as shown in Figure22.

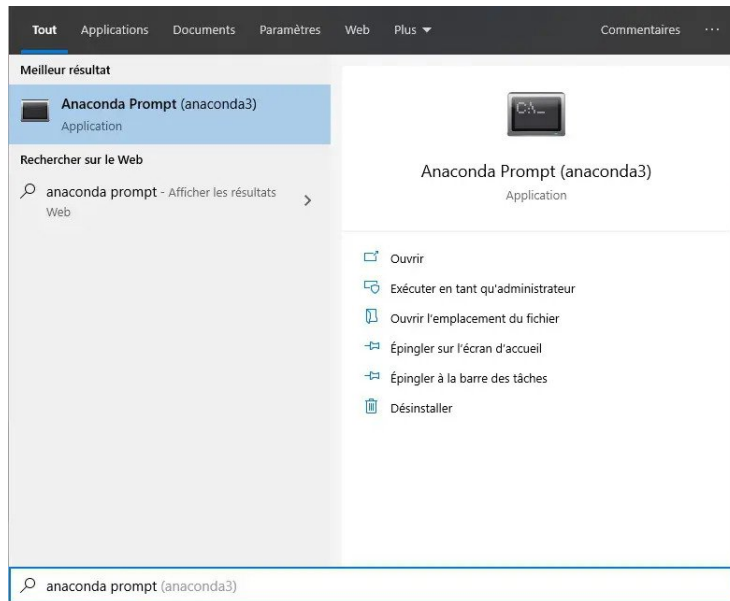


Figure 22: Opening up Anaconda Prompt.

In Anaconda Prompt, you can simply type **conda install -c conda-forge selenium** and click **Enter**, as shown in Figure23.

```

Anaconda Prompt (anaconda3) - conda install -c conda-forge selenium
(base) C:\Users\j-c.chouinard>conda install -c conda-forge selenium
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##
  environment location: C:\Users\j-c.chouinard\AppData\Local\Continuum\anaconda3
  added / updated specs:
    - selenium

The following packages will be downloaded:

```

package	build	size	channel
conda-4.7.12	py37_0	3.0 MB	conda-forge
selenium-3.141.0	py37hfa6e2cd_1000	858 KB	conda-forge
Total:		3.8 MB	

```

The following NEW packages will be INSTALLED:
  selenium                conda-forge/win-64::selenium-3.141.0-py37hfa6e2cd_1000

The following packages will be SUPERSEDED by a higher-priority channel:
  conda                   pkgs/main --> conda-forge

```

Figure 23: Selenium library installation for Anaconda.

Once Anaconda figures out where Selenium library is and possible additional required libraries, it will ask your confirmation as shown in Figure 24. Here you can type **y** and click **Enter**.


```
Proceed ([y]/n)? y

Downloading and Extracting Packages
conda-4.7.12 | 3.0 MB | ##### | 100%
selenium-3.141.0 | 858 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(base) C:\Users\j-c.chouinard>
```

Figure 24: Selenium library installation confirmation screen.

3.1.4 Selenium WebDriver Installation

To install Selenium WebDriver for allowing Selenium-Anaconda library to control the web browser (**here we use Chrome!!!**), go to the following web page, <https://sites.google.com/a/chromium.org/chromedriver/> as shown in Figure 25.

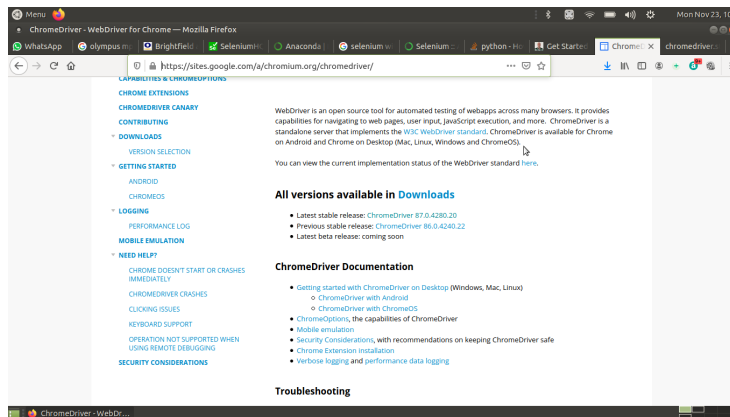


Figure 25: Selenium library installation confirmation screen.

Within other options, click on the **Latest Stable Release**. This will direct you another web page with multiple files you can download as shown in Figure 26. **Here, you have to download 32-bit driver!!!**

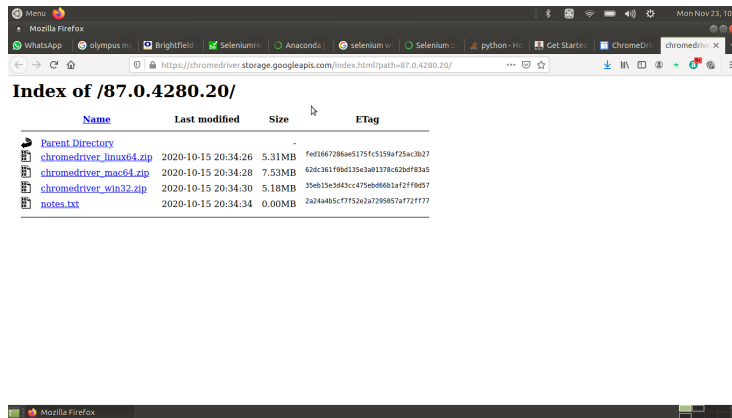


Figure 26: Selenium library installation confirmation screen.

Once download finishes, you can unzip the the webdriver file with any software you desire.

3.2 Implementation-Coding

To invoke Selenium I wrote Python Script. An overview of the code is as following;

1. **Lines 1-4:** Imports essential libraries.
2. **Lines 6-8:** Pings Google's DNS server to check whether the PC is connected to wired network.
3. **Lines 10-11:** If output of lines 6-8 are both indicating connection proceed to the end of the code.
4. **Lines 12-27:** If PC is not connected; **Line 13:** Run Chrome driver
Line 14: Send Chrome to university's Log-In page. Here "https://secure-access.morgan.edu/guest/ngnauth.php?" is a fixed string. "&mac=f8:b1:56:b3:7a:0a&_browser=1" is the MAC address of our machine and browser number which is the default browser.
5. **Lines 17-20:** Fill out the form on university's Log-In page using your username and password.
6. **Lines 21-22:** Click the box to accept the terms.
7. **Lines 23-24:** Click on the Accept button to Log-In.
8. **Lines 25-27:** Wait for the webpage to load to university's homepage then close the browser.
9. **Line 28:** Kill python.exe to stop the process cleanly.

```

1  from selenium import webdriver
2  from selenium.webdriver.common.keys import Keys
3  import time
4  import os
5
6  temp = os.popen("ping 8.8.8.8").read()
7  test_connection1 = temp.find('PING: transmit failed. General
    ↳ failure.')
8  test_connection2 = temp.find('Request timed out.')
9
10 if (test_connection1 < 0 and test_connection2 < 0):
11     print("Already connected. Exiting!!!")
12 else:
13     driver = webdriver.Chrome()
14
15     ↳ driver.get("https://secure-access.morgan.edu/guest/ngnauth.php?
        &mac=f8:b1:56:b3:7a:0a&_browser=1")
16
17     elem1 =
18     ↳ driver.find_element_by_id('ID_formcd662c6a_weblogin_user')
19     elem1.send_keys("Peker.Milas")
20     elem2 =
21     ↳ driver.find_element_by_id('ID_formcd662c6a_weblogin_password')
22     elem2.send_keys("Onder1981")
23     elem3 =
24     ↳ driver.find_element_by_id('ID_formcd662c6a_weblogin_visitor_accept_terms')
25     elem3.click()
26     elem4 =
27     ↳ driver.find_element_by_id('ID_formcd662c6a_weblogin_submit')
28     elem4.click()
29     time.sleep(10)
30
31     driver.close()
32 os.system("taskkill /F /im python.exe")

```

3.3 Implementation-Automatization

As I have indicated above, the provided Python script is meant to execute once. Therefore, to first sense a loss in wired connection and invoke the Selenium for automatic Log-In, we use Windows' **Task Scheduler**. This is done as following;

1. Click on the Windows search bar and type Task Scheduler. This will pop a widow above Windows logo to let you run Task Scheduler which is shown in Figure27

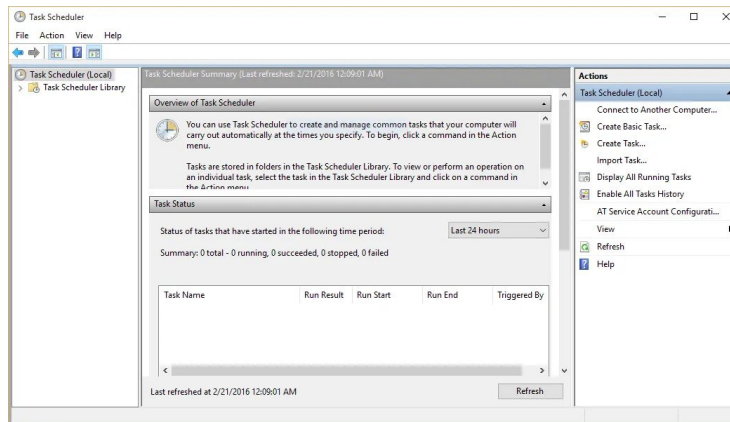


Figure 27: Windows10 Task Scheduler.

2. Click on the "Action→Create Task", as shown in Figure 28

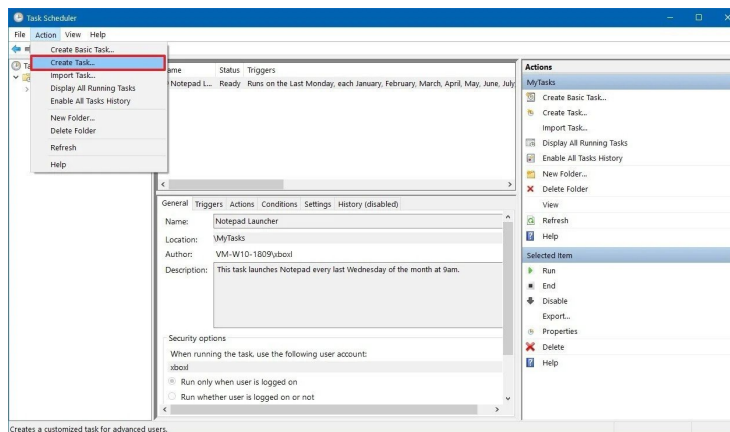


Figure 28: Create a task in Task Scheduler.

3. In popped-up window, on General tab, as shown in Figure29, give a unique name to your task and describe it for future reference.

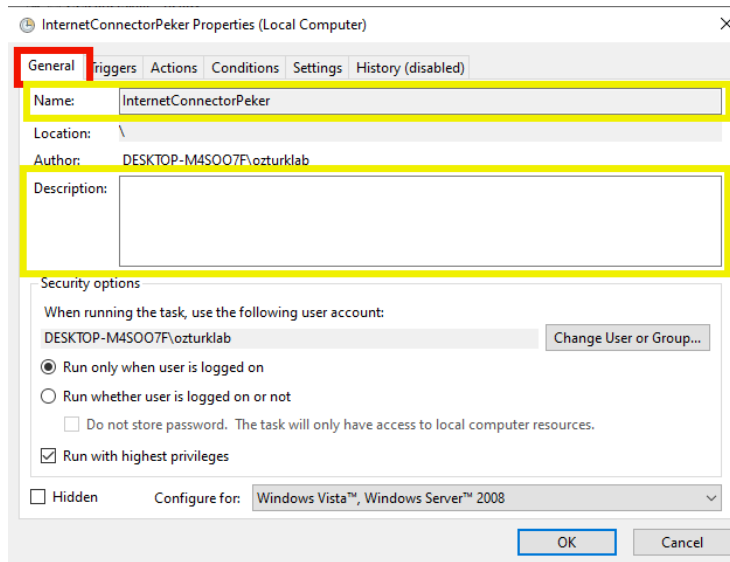


Figure 29: Name and Description fields under General tab.

4. On the Triggers tab, as shown in Figure30, click on New. In popped-up window, set the Start time and date. Also make your task runs for One time. Let the task repeat at specified times (here every 30 minutes) and keep it run indefinitely.

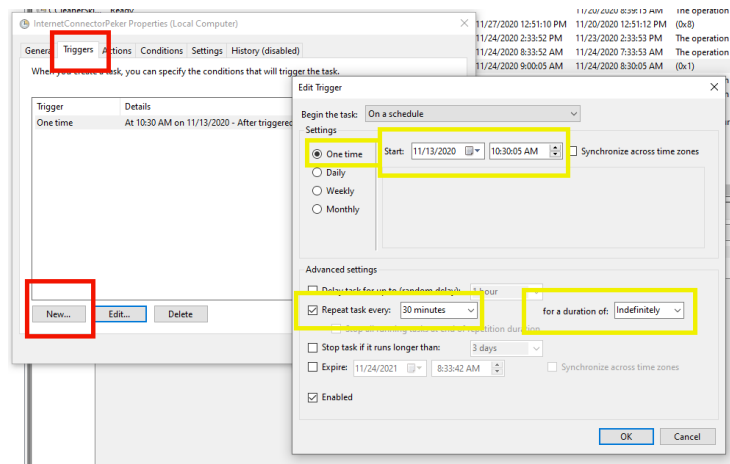


Figure 30: Timer and repetition related fields under Triggers tab.

5. On the Actions tab, as shown in Figure31, click on New. In popped-up window, make Action Start a program. In Programs/script field, add the

location of your python executable (python.exe). In Arguments field, add your python script's name. In Start in (Optional) field, add the location of your python script.

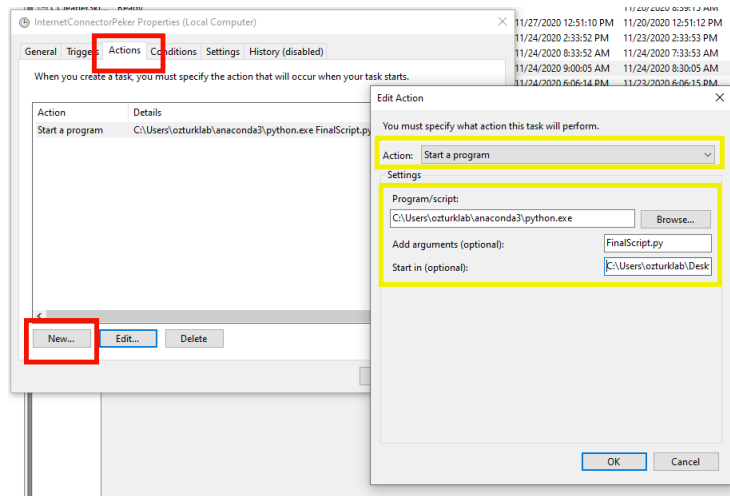


Figure 31: Script and arguments related fields under Triggers tab.

6. On the Conditions tab, as shown in Figure32, most of the fields are related with hardware such as whether you have a laptop which can run on battery or not. Therefore you can simply keep it as shown in Figure 32

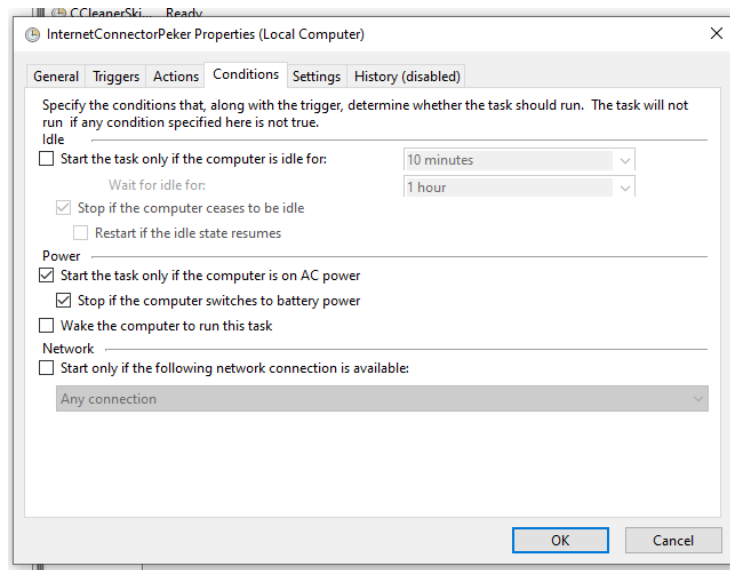


Figure 32: Already prepared Conditions tab.

7. On the Setting tab, as shown in Figure33, you can again set relevant fields as shown in Figure33.

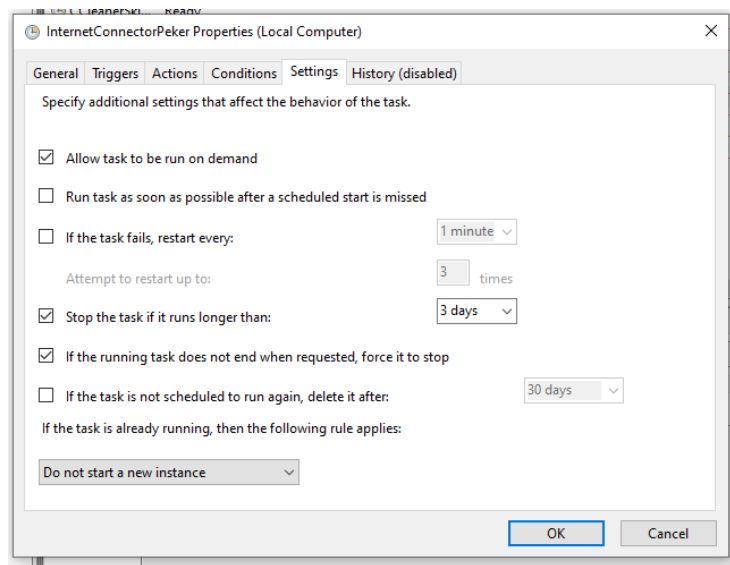


Figure 33: Already prepared Settings tab.

8. Once you are done with setting up your scheduled task, save it and close

its window. This will add your new task to default windows tasks as shown in Figure34. Right-click on your task and run it for the first time. After that point, it can run by itself.

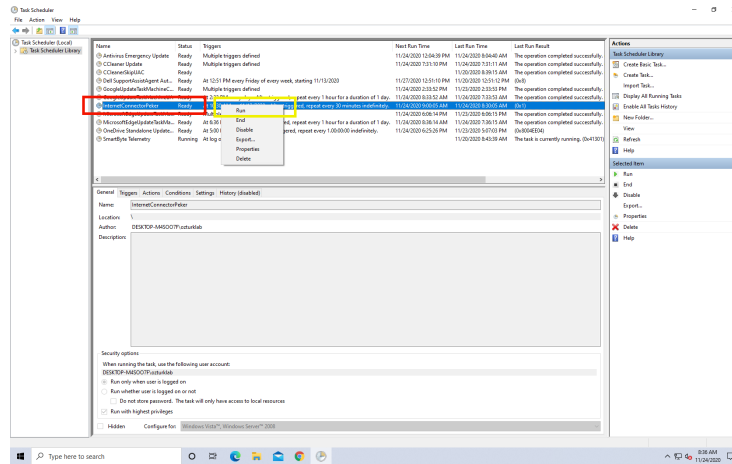


Figure 34: First run of the new task.