Tien Nguyen **IOT ESSENTIALS**

**Report Project 3**

Information Technology

ITMI17SP 25/04/2019

Timo Hynninen

**HOME SECURITY CAMERA**

For the task I would need a RPi and the Raspberry camera module. First, I needed to connect and test the camera module to see how it worked.

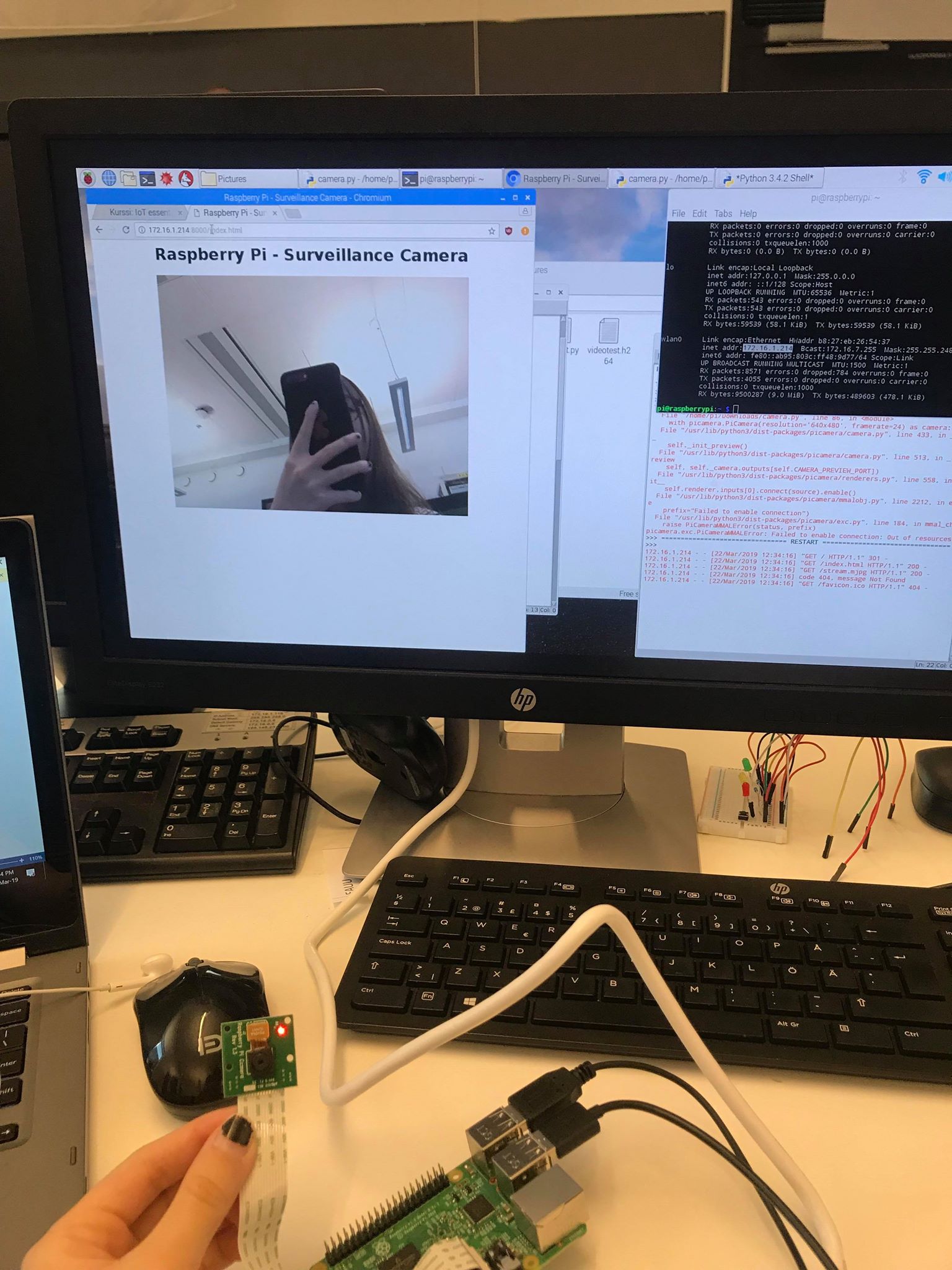
I first connect the camera module to the CSI port tightly with the blue letter facing up. Then I go Main Menu on desktop and select Preferences Configuration, then I chose the enable the Camera on the Interface tab. The setup step was done.

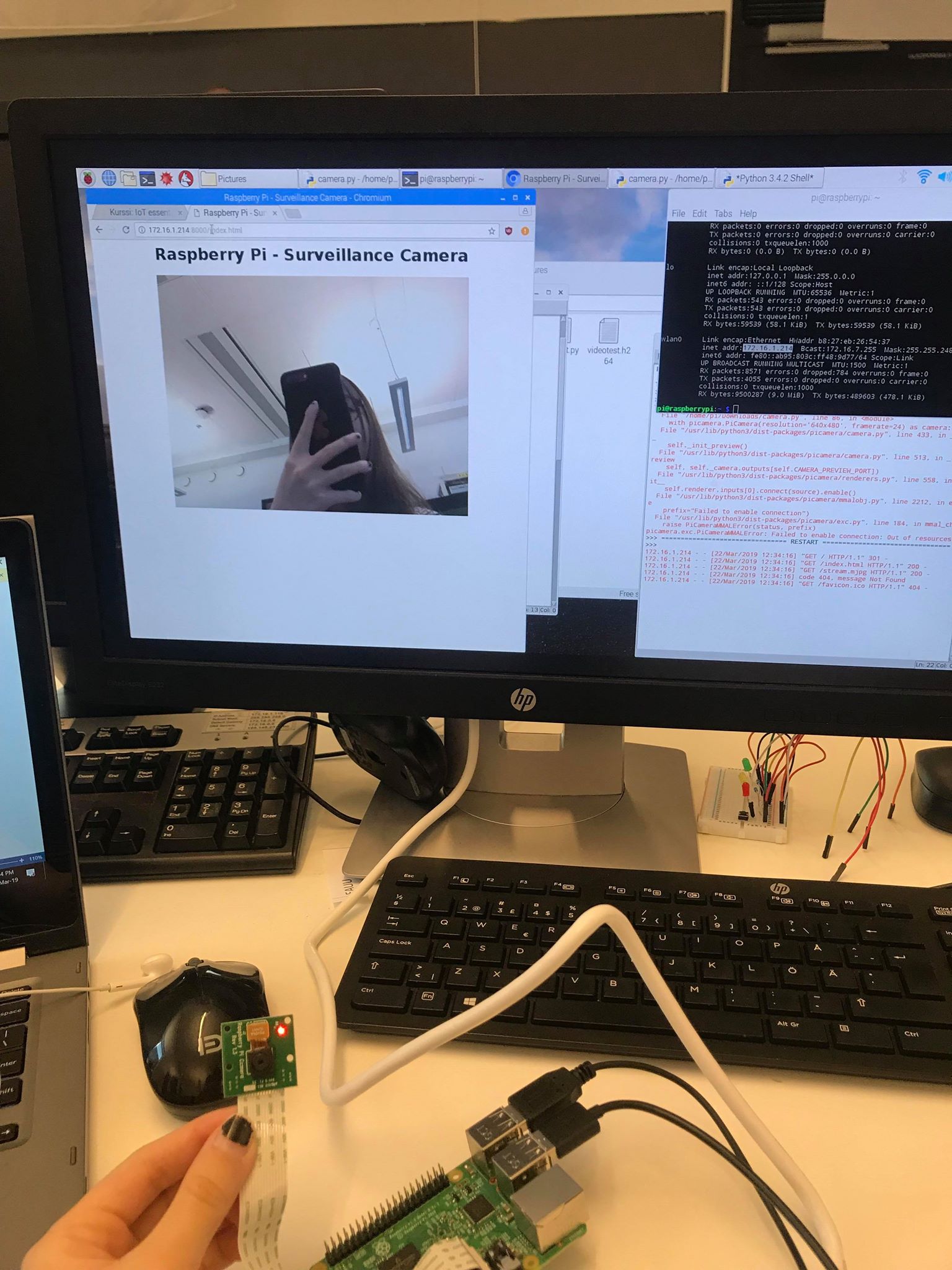
Next, I tested recording a video to see if the camera would work. I created a python script as below:

import picamera  
camera = picamera.PiCamera()  
camera.resolution = (640, 480)  
camera.start\_recording('videotest.h264')  
camera.wait\_recording(60)  
camera.stop\_recording()

First I import the camera module into the codes, and then I assigned the picamera into a variable name ‘camera’ for easie use later. Next, I set the resolution and start recording to the file name “videotest.h264”. The time of recording was for 60 seconds. When in work, the camera red light was turned on (like in the photo). When the time was up, the light turned off and the code terminated. The recorded video was saved in the script’s folder. To watch the video, I went to terminal and type : “omxplayer videotest.h264”.

Next, I was going to build a web page that would be hosted in the RPi. I download the “camera.py” file from Moodle and run the script. The “camera.py” file is also attached with this report.

To access the web page that is using the picamera, I first needed to know the IP address of this RPi. I opened terminal and type in the command “ifconfig”. After knowing the IP, I open a web browser (Chromium) and type in the address “http://172.16.1.214.:8000”

The web browser would then locate to the RPi web server and open on port 8000 as set in the script. On the web page I could see the live stream image from the picamera.

**PROJECT CHALLENGE**

I needed to customize the web page with CSS. The basic elements that I would implement were: change background color, change text, change texts’ fonts and size. The CSS elements were implemented in the PAGE part at the top of the script. The highlighted parts are the CSS parts I added:

PAGE="""\

<html>

<head>

<title>Raspberry Pi - Surveillance Camera</title>

<style>

body {background-color: #000000;}

h1 {font-family:"Times New Roman", Times, serif;

color: #99004d;}

p {color: #ffffff;}

</style>

</head>

<body>

<center><h1 style="font-size:300%";>WELCOME TO TIEN'S STREAM</h1><Center>

<p>Hello guys</p>

<center><h2 style="color:#cc0066;">Raspberry Pi - Streaming Camera</h2></center>

<center><img src="stream.mjpg" width="640" height="480"></center>

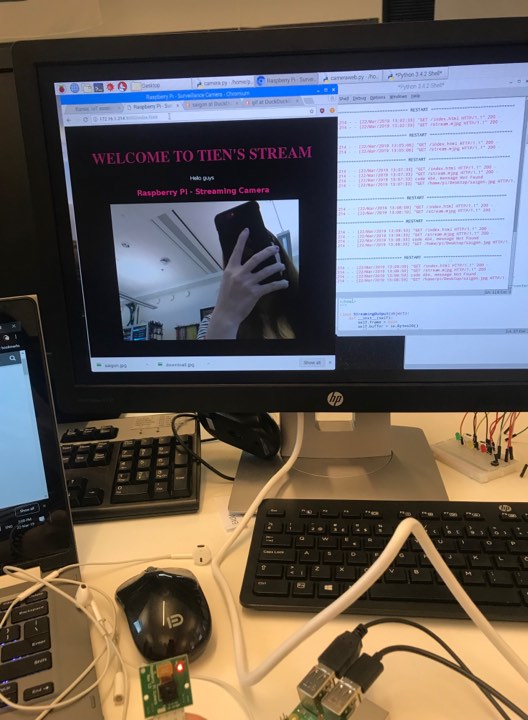
</body>

</html>

"""

I added the CSS elements using 2 different ways: Internal and Inline. The Internal method was done by using a <style> element in the <head> section. The Inline method was done by using the style attribute in HTML elements. The CSS elements I added were simple, basic and pretty self-explained.

The result of the web page after modification looked like this:



The script with the CSS implemented is attached with the name “cameraweb.py”.

**CONCLUSION**

The project challenge is quite simple to do and fun to test. During the project challenge, I actually poked around and try out different CSS elements to make the web page look better. I did try to add a background image but somehow it did not work as desired. Hence, I ended up using just a background color. Nevertheless, this project was very interesting and eye-opening.