

*Applied Technology Project*

SmartCam



***Motion Capture Video Recorder Using Raspberry Pi***

May 20, 2020

Mansoor Ali

2016459

# Table of Contents

[Table of Contents 1](#_Toc40906113)

[List of Figures: 3](#_Toc40906114)

[Introduction: 4](#_Toc40906115)

[Chapter 1: Literature Review 5](#_Toc40906116)

[Problem Area: 5](#_Toc40906117)

[Solution: 6](#_Toc40906118)

[Target Market: 6](#_Toc40906119)

[Hardware: 7](#_Toc40906120)

[Raspberry Pi 3 B 7](#_Toc40906121)

[Motion Detector 8](#_Toc40906122)

[Camera Module 9](#_Toc40906123)

[Software 9](#_Toc40906124)

[Raspbian OS 9](#_Toc40906125)

[Programming Languages 10](#_Toc40906126)

[Python 10](#_Toc40906127)

[Libraries: 10](#_Toc40906128)

[GPIO Zero 10](#_Toc40906129)

[Time 10](#_Toc40906130)

[DateTime 10](#_Toc40906131)

[PiCamera 10](#_Toc40906132)

[Cloud Computing 10](#_Toc40906133)

[OwnCloud 11](#_Toc40906134)

[Chapter 2 - Methodology System Design 12](#_Toc40906135)

[Use Case Diagram 14](#_Toc40906136)

[Chapter 3 - Design & Implementation 15](#_Toc40906137)

[Experiment 1 - Testing motion sensor using python script 15](#_Toc40906138)

[Connecting motion sensor with raspberry pi 15](#_Toc40906139)

[First python script 15](#_Toc40906140)

[Observations 16](#_Toc40906141)

[Experiment 2 - Take picture/record video 17](#_Toc40906142)

[Connect Camera module with raspberry pi 17](#_Toc40906143)

[Python script to capture picture 18](#_Toc40906144)

[Python script to shoot video 18](#_Toc40906145)

[Observation 19](#_Toc40906146)

[Experiment 3 – Install MotionEyeOS in raspberry pi 19](#_Toc40906147)

[Install MotionEyeOS 19](#_Toc40906148)

[Back to Raspbian: 19](#_Toc40906149)

[Configure Camera Module 20](#_Toc40906150)

[Observations 20](#_Toc40906151)

[Chapter 4 - Results 21](#_Toc40906152)

[Final Prototype – Combine altogether 21](#_Toc40906153)

[Combine experiment 1 & 3 21](#_Toc40906154)

[Conclusion 25](#_Toc40906155)

[References: 26](#_Toc40906156)

[Appendix 27](#_Toc40906157)

# List of Figures:

[Figure 1‑1: Project Architecture 6](file:///C:\Users\Manso\OneDrive\project\ProjectProposal.docx#_Toc40902120)

[Figure 1‑2: Raspberry Pi Board 7](file:///C:\Users\Manso\OneDrive\project\ProjectProposal.docx#_Toc40902121)

[Figure 1‑3: PIR motion sensor 8](file:///C:\Users\Manso\OneDrive\project\ProjectProposal.docx#_Toc40902122)

[Figure 1‑4: Camera moduleV2 for Raspberry Pi 9](file:///C:\Users\Manso\OneDrive\project\ProjectProposal.docx#_Toc40902123)

[Figure 2‑1: High Level Diagram of Proposed Solution 12](#_Toc40902124)

[Figure 2‑2: Raspberry Pi 3 GPIO pins layout 13](#_Toc40902125)

[Figure 2‑3 UseCase Diagram - User Functionalities 14](#_Toc40902126)

[Figure 3‑1: Motion Sensor Connection 15](#_Toc40902127)

[Figure 3‑2: First experiment script 16](#_Toc40902128)

[Figure 3‑3; Camera module connection with Raspberry pi 17](#_Toc40902129)

[Figure 3‑4: Program taking picture 18](#_Toc40902130)

[Figure 3‑5: Code to capture video 19](#_Toc40902131)

[Figure 3‑6: Picture Taken with Camera Module 20](#_Toc40902132)

[Figure 4‑1: Both modules connected with Raspberry Pi 21](#_Toc40902133)

[Figure 4‑2: Video generated after motion occur 22](#_Toc40902134)

[Figure 4‑3: TimeStamp naming 23](#_Toc40902135)

[Figure 4‑4: Program running & waiting for motion 24](#_Toc40902136)

# Introduction:

The idea of controlling objects remotely can be very appealing. Understanding how computers work in an affordable low cost and highly efficient way is the main idea behind the Raspberry Pi and the reason this project has been built using this technology.

SmartCam is a prototype, which presents camera and motion sensor connecting to raspberry pi and it allows to monitor remotely, while saving disk memory and electricity consumption. It works with motion sensor, it means it record video only when it detect some motion. The system is based on python and its highly useful libraries, such as: GPIO Zero, picamera, Time, DateTime. The system is able to detect motion continuously and when it record video it saves video with the current time as name of the video, by doing this, it has become easy to distinguish, what happens when and vice versa.

The first chapter of this document comprises the theoretical literature review and provide information about several technologies used in this project. It presents information about communication technologies, software, and different hardware components that are used in order to build this project.

The second and third chapter of this document provide the detail about the designing of system and then its implementation. It provide different diagrams of the system which makes it easy to understand the whole system and how its work. It also provides information about different experiments conducted in order to reach final prototype. It also covers problems faced during the completion of this project.

The last chapter covers the building of final prototype and discuss the final results and provide a better understanding of how the whole system works.

The motivation behind this project was not only the curiosity to control things remotely and securely, but also to understand how different technologies works and motivation was also to apply a practical opportunity to the knowledge gain during my academic journey.

# Chapter 1: Literature Review

In this chapter, I have highlighted the problem that my project aim is to solve. I have also describe about the target Market for the prototype, I have worked upon. I have also discuss about the Hardware, Software and languages, I am going to use in order to build my project final prototype. This chapter includes the whole literature stuff and provide detail information about components, software, languages and other services used in order to build this project.

## Problem Area:

CCTV cameras are working fine from numbers of decades and use of these cameras is increasing day by day. But, these cameras have some disadvantages too i.e. security, data storage, notification. There is no need to record a video if it cannot stop burglary on the spot. Attackers can damage the cameras or put some sort of spray on camera lens to disturb its functioning and user will not get any alert. Cyber attackers can easily hack the entire network of an organization by injecting Distributed denial of service DDoS attacks to CCTV systems as they are not very secure or some does not support SSL/TLS encryption. Any insecure device connected to the internet is the main target for attackers to enter into target network and this can put a lot of important data at huge risk. Notification is also an issue, when it comes to CCTV cameras. Unlike other intrusion detection systems such as smoke alarms, fire alarms etc. CCTV cameras are not programmed to notify the administrator, if any incidents occur. CCTV cameras just record the specified area in there range and the administrator will have to check all the recording to catch the culprit, in case any disturbance occurred in their absence. For storing videos CCTV systems need external hard drives or some sort of cloud storage platform and that need some extra cost, which is also an issue for small businesses or home owners who are looking for cheap security system in order to get some discounts on insurance. A research done by cloudview a company who transform visual that in a way to make it accessible and manageable state different vulnerabilities of CCTV DVR systems and those vulnerabilities are:

No Oversight, firmware updates, and port forwarding etc. These are the issues that need to be consider and need some attention or IT professional to look after these systems and that cost a lot to hire someone to look after CCTV systems and check if there any security breach in those systems.

## Solution:

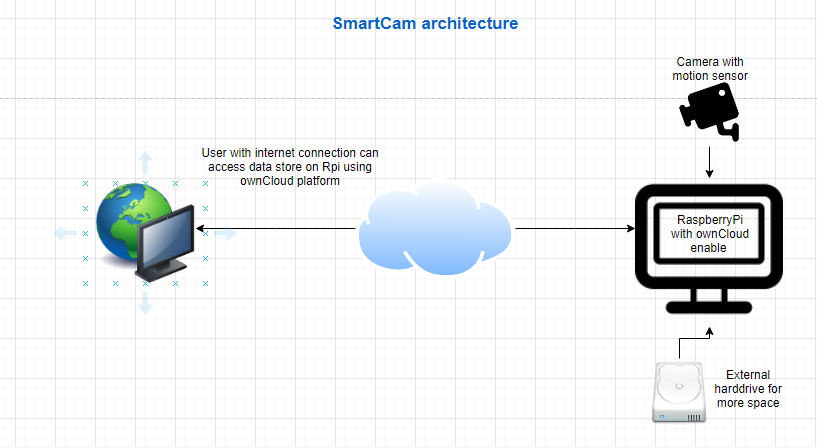


Figure 1‑: Project Architecture

I came up with the solution of SmartCam in order to overcome issue arise by CCTV cameras. The SmartCam will be built using Raspberry Pi, PIR motion sensor, and a camera Module. Upon successful completion of this system, it will be able to capture pictures, record videos only if any motion occurs and also, the Raspberry Pi the base of this system will be programmed to notify the administrator of this System. In order to deal with data storage problem, I have solution which is OwnCloud. It is a service that can be enabled on Raspberry Pi freely and the Raspberry Pi will act as Server and the individuals (admins) can access data (pictures/videos) stored on it remotely. . If the individual wanted to keep all of its content and require big storage media, so OwnCloud also allow adding external hard drive to act as cloud service Admins will also be able to delete not usable content in order to free device space. Raspberry Pi runs on Operating system Raspbian, which is Linux based and very secure and also its an open source project and a lot of work is going on to make this device able to give us full advantage of it. There are a lot of IOT based projects that are built using Raspberry Pi and are doing incredible jobs.

## Target Market:

Once I have the final prototype, that is “A camera using Raspberry Pi who capture pictures videos only when some motion/movement occur” it will be very useful for Home owners who need some security cameras in their homes and don’t want to spent a lot on electricity and storage i.e. CCTV cameras runs 24/7 in order to record video and use a lot of electricity. So, the home owners can have a secure home using this prototype and in that way they can get some discounts from insurance company if they want to have insurance for their houses. My final prototype will also be very useful for small businesses, who need some security camera to monitor their office remotely and don’t want to spent a lot on management of these IT equipment’s. Big companies who need a lot of security cameras in order to monitor the company’s building will not be able to get benefit from this prototype as this will be based on one camera, but later upon successful completion of this project, the concept can be extended to add more cameras and motion sensors to make it useful for big companies. Especially, for security companies who have different sites to look after, will be able to use this extended prototype and can reduce cost spent on employee’s they hire.

## Hardware:

This section is about hardware components that are used to build the SmartCam.

### Raspberry Pi 3 B

****

Figure 1‑: Raspberry Pi Board

I have use Raspberry Pi 3 B to build my project. This small credit card size device work exactly as desktop computer, and will act as the main device on which I will be working on in order to accomplish my project goal. In order to interact with it we only need to add mouse, keyboard and monitor. It use micro SD card, that work as hard drive of this small device and the operating system will be loaded into this hard drive. The operating system “Raspbian” can be downloaded freely from Raspberry Pi official website. This is not the latest version of Raspberry Pi, as there is a fourth edition of Raspberry Pi as well but, that is quite similar to it except the processor speed, support different size of RAMs and support smaller HDMI ports for display.

According to Raspberry Pi foundation the specification of model 3 B is specified below:-

* + 1GB RAM
  + Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
  + BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
  + 100 Base Ethernet
  + 40-pin extended GPIO
  + 4 USB 2 ports
  + 4 Pole stereo output and composite video port
  + Full size HDMI
  + CSI camera port for connecting a Raspberry Pi camera
  + DSI display port for connecting a Raspberry Pi touchscreen display
  + Micro SD port for loading your operating system and storing data
  + Upgraded switched Micro USB power source up to 2.5A

### Motion Detector

Figure 1‑: PIR motion sensor

In this project, I am going to use PIR passive infrared motion sensor. These are mostly useful in burglar alarm systems. It emits infrared radiations, when the temperature of object/organism is above absolute zero (-273.15 C). Infrared radiations wavelength cannot be seen by human eye, but if any disturbance occur, it can be detected by motion sensors. This have nothing to do with movement of things, it works by adjusting itself to infrared signature (temperature disturbance) of room it is in and then start watching for changes. Any movement occur will disturb the infrared signature (temperature) and then PIR sensor will detect this disturbance and we can program it to do something with this disturbance.

These can be used with lights, security cameras, and smoke detectors etc. For my project I will be using this module with raspberry pi, in order to capture motion and camera will also be added to Rpi to start taking pictures/videos.

### Camera Module

Figure 1‑: Camera moduleV2 for Raspberry Pi

The Rpi Camera Module V2 is the second version of module introduced in April 2016. It has 8 megapixel Sony camera that aid in taking high quality pictures or record videos. CSI port on raspberry pi can be used to connect this camera via a 15cm long ribbon cable. This camera module work on all versions of Rpi 1, 2, 3, and 4 except Rpi 0. According to Raspberry Pi official website this camera is very popular in making home security applications and wildlife traps. I will be using this module with Rpi 3B in order to achieve my project goals. There are numerous libraries that are built for it including Picamera python library, which I will be using to get full benefits of camera module.

## Software

### Raspbian OS

Raspbian is the Operating system supported on Raspberry Pi. It is like a Linux OS and easy to use in order to interact with Rpi. It has plenty of pre-installed software for education, programming and general use. It also supports programming languages like java, python, scratch and Sonic Pi etc. It can be downloaded from Raspberry Pi website and can be installed on your Raspberry Pi so that you can interact with Raspberry Pi. I have already installed this in my Rpi in order to interact with the device.

## Programming Languages

### Python

Python is an object-oriented, interpreter, high-level programming language. Python is very simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. I am going to use this language in my project to write scripts that can work with camera module and motion sensor. I will be using the PiCamera python library. I will write codes/commands in python to access Camera module in order to record videos when some motion occur. Python provides features of dynamic typing and dynamic binding, which make it very attractive for Rapid application development, as well as for use as a scripting. Most of the programmers fall in love with Python because of the increased productivity it provides and the edit-test-debug cycle is incredibly fast. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

## Libraries:

Different python libraries will be used in this project. I have listed their name and details below.

### 

### GPIO Zero

The GPIO Zero library includes interfaces to several different components, as well as some complex functions such as sensors, analogue to digital, LEDs, and others etc. I will be using this library to use MotionSensor class for motion detection.

### 

### Time

Time is another library, I will be using in order to set time video would be capture after some motion occur.

### 

### DateTime

DateTime library will be used to put TimeStamp as a name for a media storage, as giving name statically is not possible So, I will be using this library to name files as current time and by doing this I will be able to distinguish different videos/pictures captured.

### 

### PiCamera

PiCamera is another library can be used to import picamera class in order to deal with camera module and start making videos/pictures for project prototype.

## Cloud Computing

Cloud Computing has made a significant presence in the field of technology and has become one of the most important and growing area in IT sector. The process of storing data and providing different services through internet is an on-demand facility and it benefit both individual users and big companies. According to Connor, M. (2018) users can benefit from cloud computing in several matters:

* Enhanced security and compliance (security of data is enhanced)
* Mobility (employees can work from home with the help of internet)
* Increased Collaboration (easy access to edit files and share documents anytime, anywhere)
* Cloud Storage (Access data anywhere using internet)
* Cost Reduction (reduce cost of hardware and provide pay as you go services)

Considering all these amazing benefits of cloud computing, a cloud based platform will be used in order to store data and access it remotely.

### OwnCloud

As an alternative to Google Drive, iCloud, One Drive, OwnCloud offers security and privacy of your data and allows you to access it remotely from anywhere using internet. It can provide you extra layer of security as users don’t need to relay on different cloud vendors and allow users to make their own cloud platform free of cost. With free desktop client and ownCloud app users can access their data at anytime from anywhere. OwnCloud is one of the best program for running a cloud storage system on Raspberry Pi. Once configure the OwnCloud service on Raspberry Pi, we can access data stored on the drive of raspberry pi remotely. An external hard drive can be added to Raspberry Pi to provide a fully functional and secure cloud platform in free of cost. OwnCloud platform will be used in my project in order to access data (videos/pictures captured when motion occur) stored on Raspberry pi from any device connected to internet.

# Chapter 2 - Methodology System Design

In this chapter, the methodology and deigns decisions followed to develop a SmartCam are described.

The proposed solution will be able to capture videos only when some motion occur in the area of focus, the captured video is then stored to OwnCloud, a personal cloud service which give access to data stored in cloud server to any device using OwnCloud Client app. The raspberry pi here, will act as cloud server and user can access stored data in the raspberry pi remotely and will also be able to delete unnecessary data, if not needed anymore.

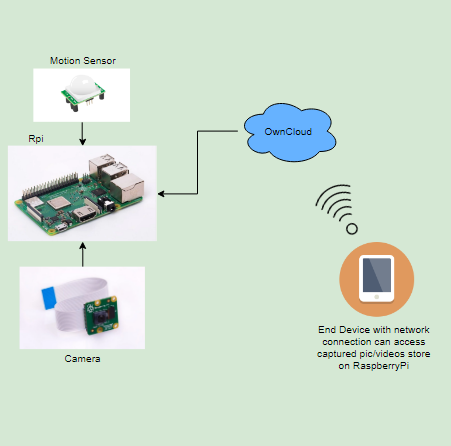


Figure 2‑: High Level Diagram of Proposed Solution

The goal of this project was to develop a prototype that can be manage through internet and on the other hand, save device storage as it only record when some movements occur.

The motion sensor will be connected to the raspberry pi using GPIO pins on device board. We can control the functionality of motion sensor using GPIO Zero library. GPIO Zero is a zero-boilerplate library for python and allow physical computing more accessible and efficient. Below is the image of raspberry pi 3 GPIO pins layout provided by raspberry pi official website:

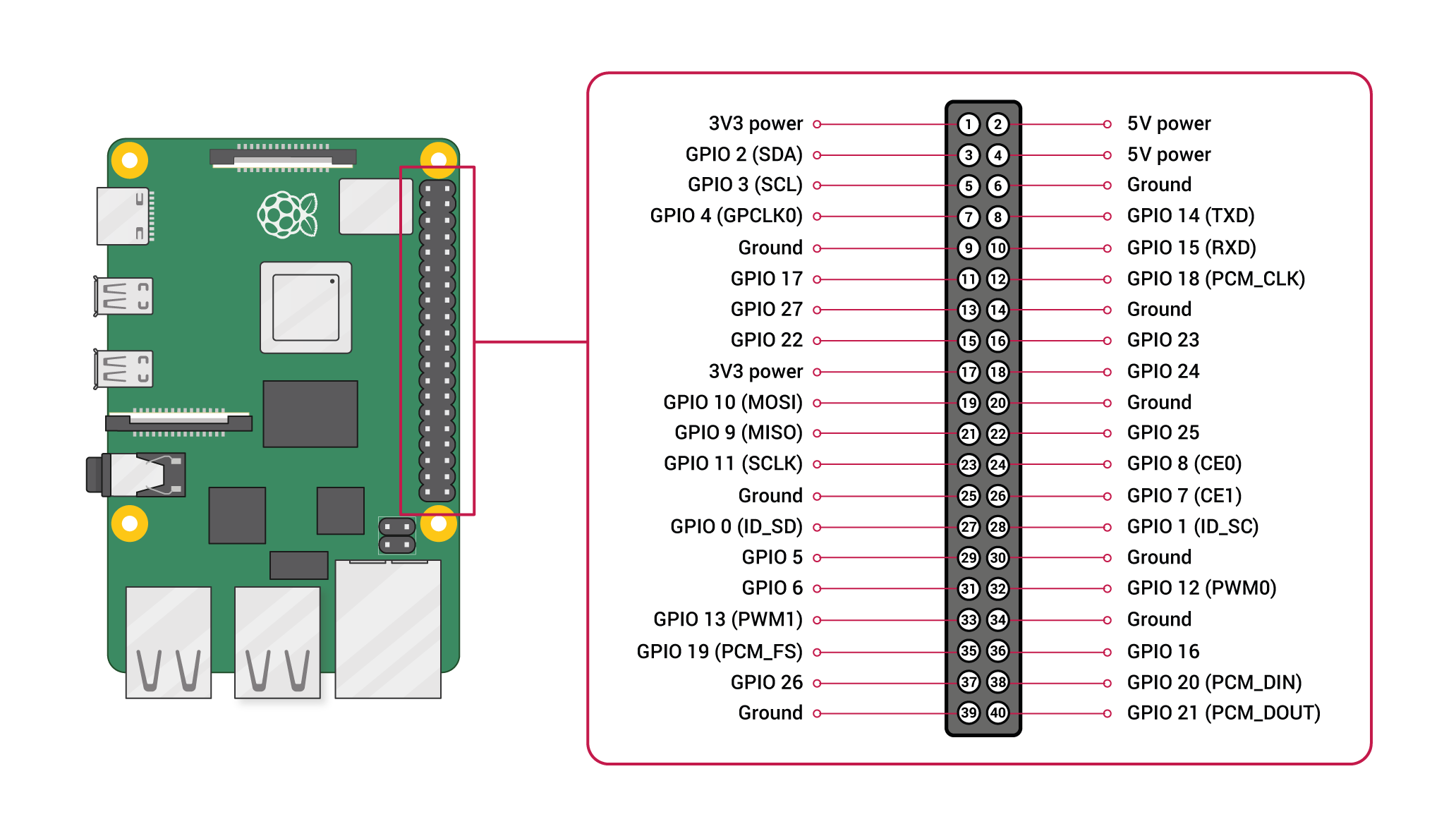


Figure 2‑: Raspberry Pi 3 GPIO pins layout

### 

## Use Case Diagram

Below is another diagram, which shows the SmartCam system that offer some functionalities represented as UML (Unified Modelling Language) Use Case Diagram.

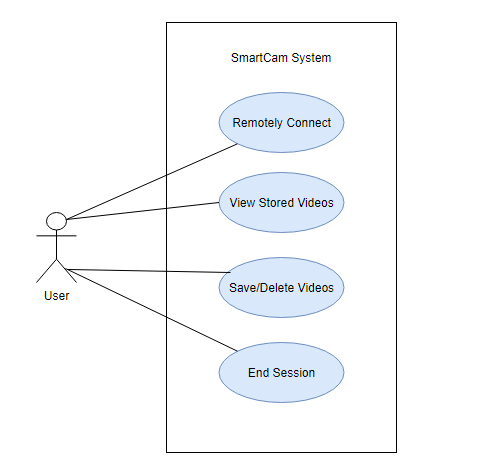


Figure 2‑3 UseCase Diagram - User Functionalities

# Chapter 3 - Design & Implementation

This chapter describes the different little experiments involved in making the SmartCam. Every little task leads me to the actual results that is presented in next chapter.

## Experiment 1 - Testing motion sensor using python script

First of all, I have install Raspbian operating system in device following raspberrypi.org guide of installation. Installing OS was not very difficult as I have done that couple of times earlier. And then I connect motion sensor to raspberry pi.

### 

### Connecting motion sensor with raspberry pi

In this first experiment, I have connected raspberry pi to motion sensor using GPIO pins. There are three pins on the motion sensor, named as VCC, GND and OUT. All three pins need to be connected to raspberry pi pins. I have connected VCC to 5V, GND to Ground and OUT to GPIO 4 on raspberry pi. I have follow this connection setting as recommended by raspberrypi.org in projects section. Image of motion sensor connected to raspberry pi is shown below:

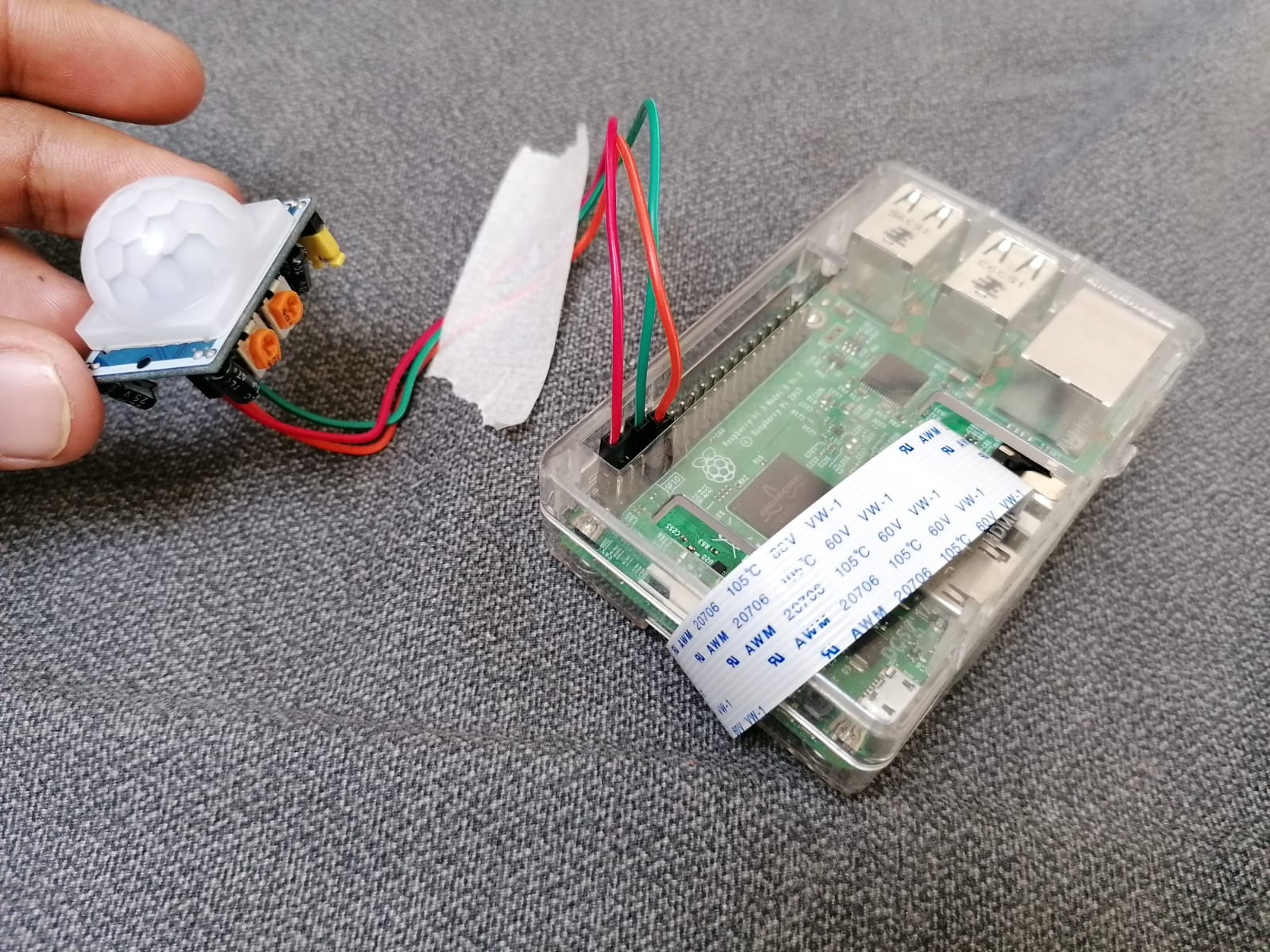


Figure 3‑: Motion Sensor Connection

After connecting motion sensor to the raspberry pi we need to turn the raspberry pi so that we can write scripts to print message when some motion occur.

### First python script

### 

In order to print message to screen that motion is detected or not, a programming language is needed. For SmartCam project I have used python, because of simplicity and amount of documentation and libraries available of it, especially regarding the use of GPIO pins. After deciding the language I have decided to write python script that print message if it caught some motion. To write program in python we can use mu an integrated development environment. It is pre-installed in raspberry pi and allow to do programming in python easily. The code I have written for this experiment is shown below:

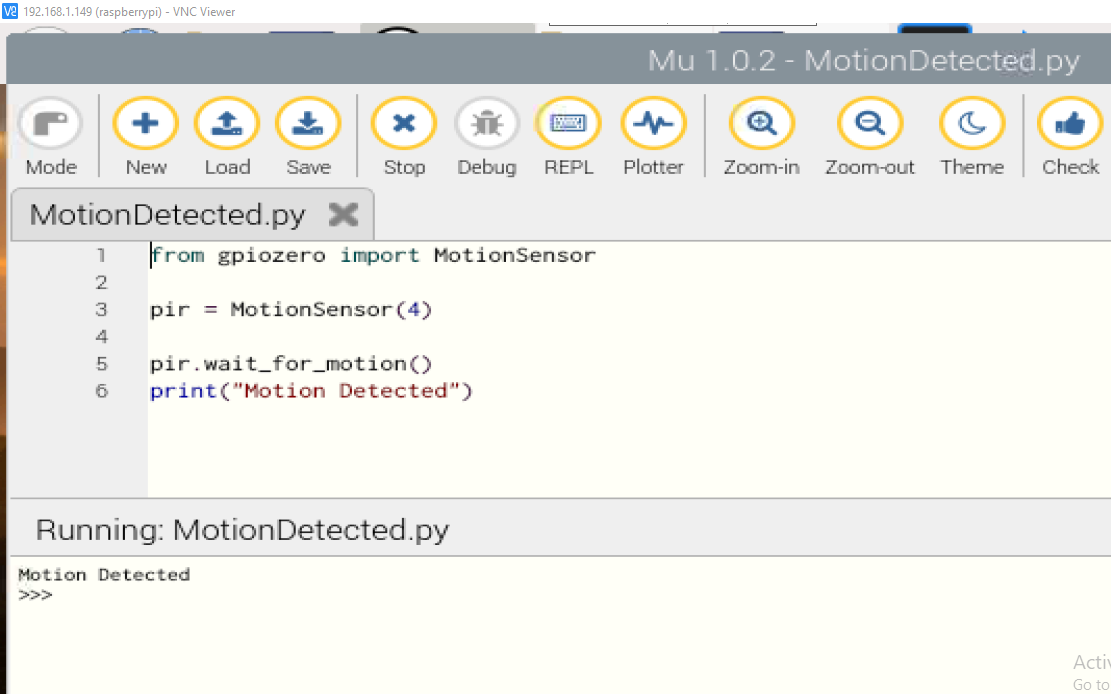


Figure 3‑: First experiment script

In the picture above, there are two main sections. One is the code at the top and below is the output of it. Code is very simple. The first line is used to import MotionSensor class from gpiozero library. And then pir is the object created and pointed towards MotionSensor, where 4 in the brackets tells about the port number on which the OUT pin of Motion Sensor is connected. In the previous section, I have already shown the connection of sensor to raspberry pi. The next line is telling the object pir to wait for motion and if motion detected, next line will print message that motion is detected. When I run program I become freeze for a while and nothing happen and then when I move little bit, it printed the expected message to the screen.

### Observations

Some of key points highlighted in this experiment are:

* Motion sensor works fine and this script can be used in order to progress in project.
* The only problem is that, after detecting motion it stops working and we can overcome this by using loop in our program and make it run again and again as long as I do not stop it manually.
* After putting our program in a loop, we can also use DateTime library and put current TimeStamp, So that every time motion detects it print current time with message.

After becoming successful in experiment 1 and observing its working, I like to progress further, in order to accomplish my project core task.

## Experiment 2 - Take picture/record video

This experiment is about taking picture using camera module with raspberry pi.

### Connect Camera module with raspberry pi

First we have to connect camera module to raspberry pi and then we can write a small program for it to work. We can connect camera module to raspberry pi by using its ribbon cable and insert this cable in camera module port on raspberry pi. In the picture below I have shown how camera module can be connected to raspberry pi.



Figure 3‑; Camera module connection with Raspberry pi

After camera module is connected to raspberry pi, we can turn on the device and configure it to start taking picture or capture videos using it. In order to use camera module and start taking pictures with it we first need to enable camera feature in the raspberry pi and that can be done by going to raspberry pi configurations or entering command shown below in terminal:

$: sudo raspi-config

After entering configurations, we need to go to interfaces and then we can set the camera feature to enable from disable, which is by default. This will ask to reboot system, press enter and after system restart we can start writing program to take picture.

### Python script to capture picture

This program is written after importing picamera class PiCamera library. This will allow to create camera object and then we can say camera object to turn on and capture picture. Once the picture is taken we can say camera object to turn off and we can print message that picture is taken. Below is the result of the program written with the help of guy Harper, A. (2015), who shows how to take picture with python on YouTube.

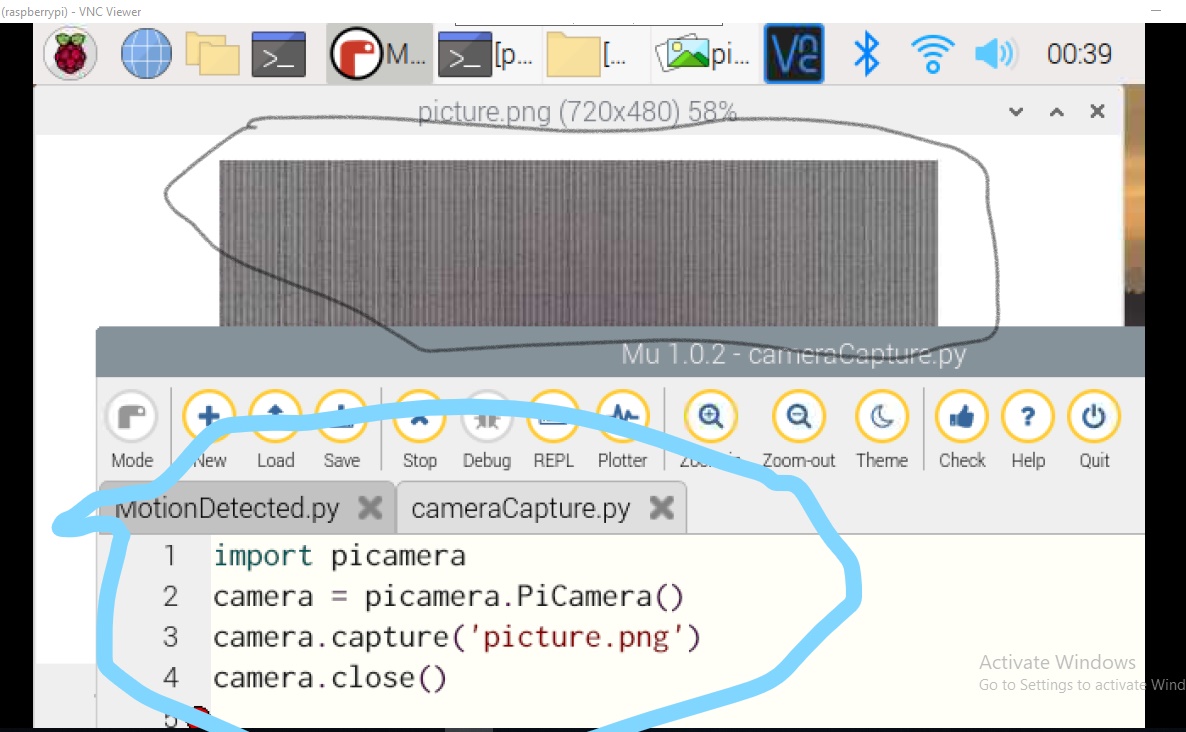


Figure 3‑: Program taking picture

In the picture above the program and its result after running is shown. The section marked with black mark shows the picture taken by code shown in section marked as blue. We can also see that it take picture but the result is really bizarre and out of expectations.

### Python script to shoot video

After getting those out of expectations results, I thought to move on and decided to shoot video by making changes to the code above. I took help from video of the same guy Harper, A. (2015) and follow him and he got the results but I still could not manage to get some positive results.

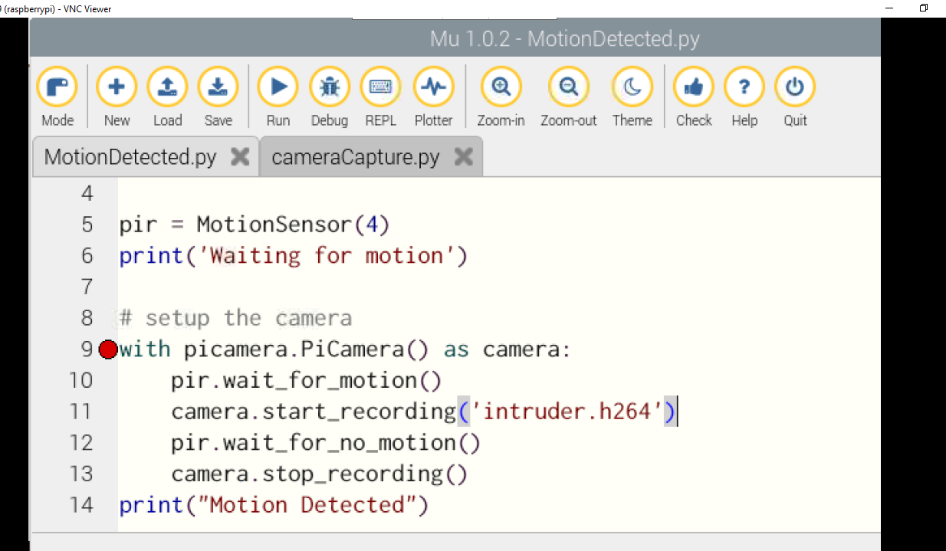


Figure 3‑: Code to capture video

Above is the code that works and capture video. But still present me with the similar screen as shown above in the figure 3-4. Video starts and run for some time, but I did not see anything in it but a pink colour screen.

### Observation

* After doing this experiment I observe that the code I am using for taking picture or capturing video works fine, but it does not provide me with the results I expect or something near to that.
* After seeing the guy I follow got the results, but I am not getting that I observe that there is a problem with camera module or may be the current OS in my device.
* I did some research and find same lines of code for configuring camera module, but mine is not working.

## Experiment 3 – Install MotionEyeOS in raspberry pi

After failing in experiment 2 and realising there is a problem with my camera module, I mail the company who provided me with camera module, but I got no response from then even days passed. I also check the camera module instruction book and find link for [www.okdo.com](http://www.okdo.com) website. Okdo website recommend MotionEyeOS for projects related to camera module. So I decided to install MotionEyeOS in my raspberry pi.

### Install MotionEyeOS

I backup Raspbian OS in my laptop and install MotionEyeOS in pi. Installation was easy and quick. After installing I realised that this OS is only for camera functions and when I tested my module, it present the same blank screen. I decided to switch back to Raspbian and try to find some other way of fixing the problem.

### Back to Raspbian:

After failing with MotionEyeOS I move to previous OS, but this OS was corrupted as well. I was disappointed with it, but decided not to quit and install Raspbian from scratch. After a bit of struggle, I manage to install Raspbian again. And start doing research on how to enable OwnCloud service on pi. I find a suitable article from guy Gus. (2015) to enable OwnCloud. And this guy ask to update raspberry pi before starting to enable OwnCloud. I did updates bu using command shown below.

$: sudo apt-get update

### 

### Configure Camera Module

After updating Raspberry pi, I decided to once again work with camera module while hoping of positive results. I had two camera modules and both of which I tested earlier and got no result that time. This time I use one module and run same code that I make to take picture and this time it was showing error and not even taking picture. I switch to second module and run the same code again and finally got positive results. The result of picture taken is shown below.

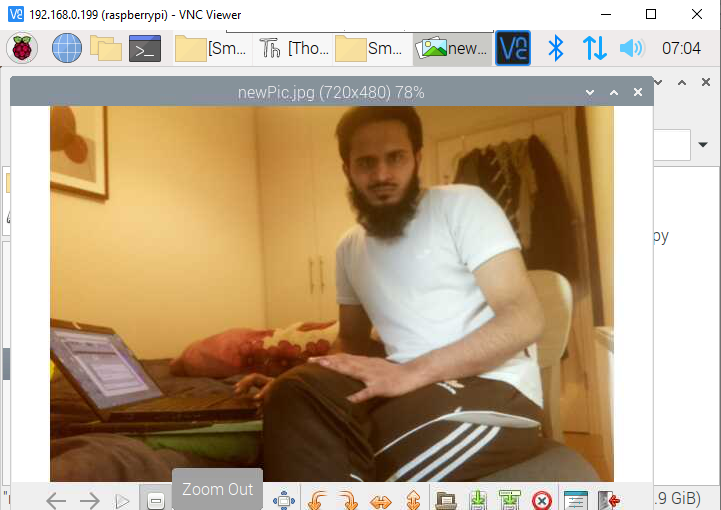


Figure 3‑: Picture Taken with Camera Module

As it can be shown in image above that the camera module works fine now. And I thought to start making video with it as, video is necessary for my project. I did changes in code and capture a video that works fine as well.

### Observations

* After struggling with the problem first, I make different changes and got no results.
* Getting disappointed by negative results made me leave this issue and progress on other tasks as I was getting out of time.
* I decided to install OwnCloud cloud computing feature and for that I did update my raspberry pi by following the research recommendations.
* After doing update, I once again thought to test and camera module and this time I become successful in it and observe that I was having problem because of no updates.

# Chapter 4 - Results

This chapter is about the final prototype that my project SmartCam has to present with. The after doing different experiments shown in previous chapter, I decided to progress with experiment 1 and experiment 3 as experiment 2 was a failure. I also decided not to enable cloud service due to shortage of time and decided to go on with what I have tested and present with final prototype.

## Final Prototype – Combine altogether

Final prototype is created after analysing results of previous experiments and modifying them, as those were just baby steps.

### Combine experiment 1 & 3

I connect both motion sensor and camera module to raspberry pi and decided to test these both together by combining python scripts, which I created earlier. Both modules are connected together and can be seen in picture shown below.

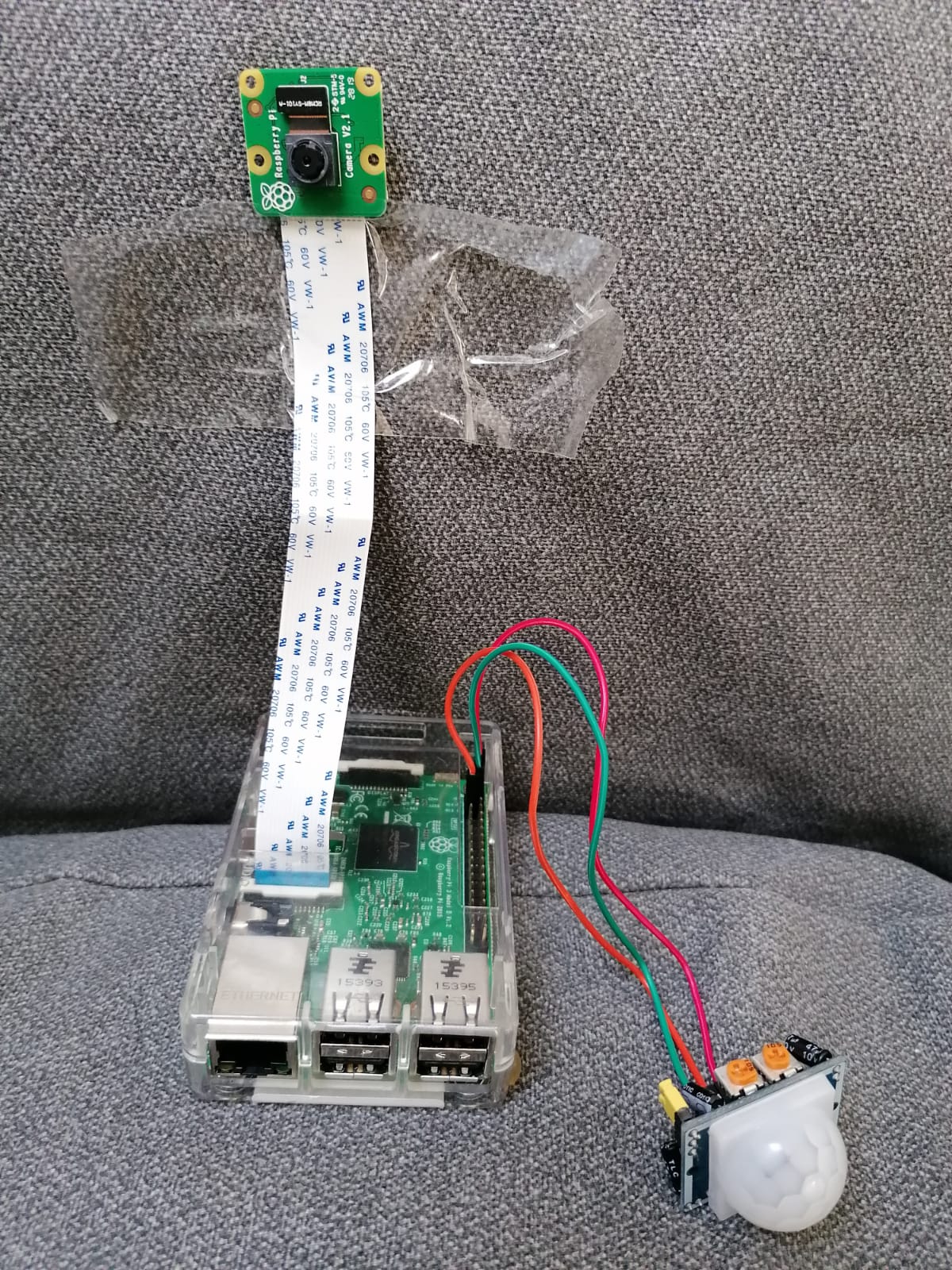


Figure 4‑: Both modules connected with Raspberry Pi

After connecting these I mergre python code so that It do capture only when some motion occur. And I manage to get that done easily and results can be seen in picture shown below.

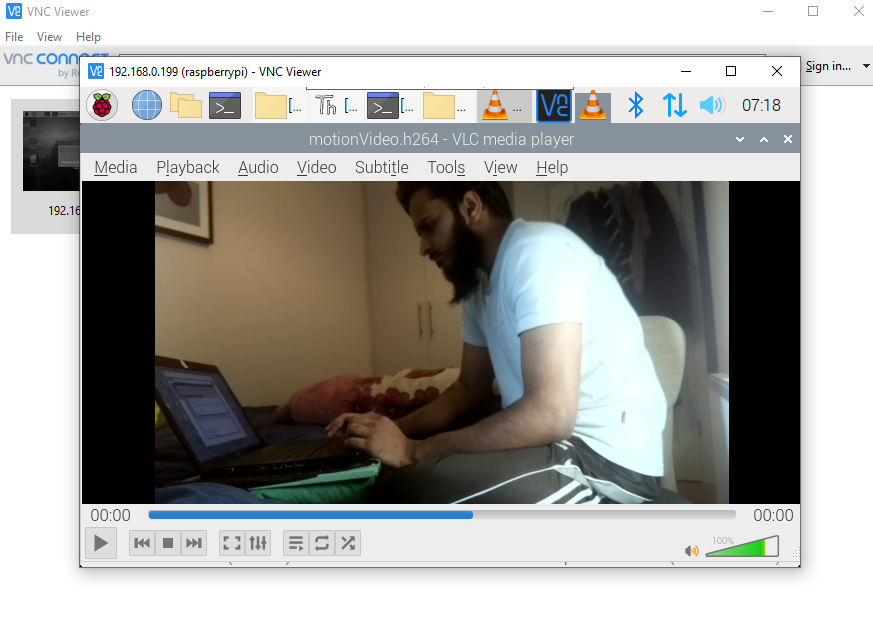


Figure 4‑: Video generated after motion occur

Above is the screen capture of video running, and this video was taken after running the code that start recording after some motion occur. The code for this program is provided in appendix. As I become successful in this, I still need to configure program to start working and wait for motion and if motion occur capture the motion and stop recording and again start waiting for motion. Above shown picture is from the code result that only capture for once In order to meet my project goal, I need to make a program that run in a loop and keep recording videos if some movements occurs. While thinking to put code in a loop, I also thought for a way to name different videos and how will I distinguish them. I did some research on it and find another interesting video on YouTube from Harper, A. (2015) regarding naming picture as current Time Stamp. His tutorial was about naming picture and I was looking for a way to name videos dynamically. As from Harper, A. (2015) tutorial, I found a way but it require some changes and that was not a difficult task as changes required were very simple and understandable. I did those changes and got the results shown below.

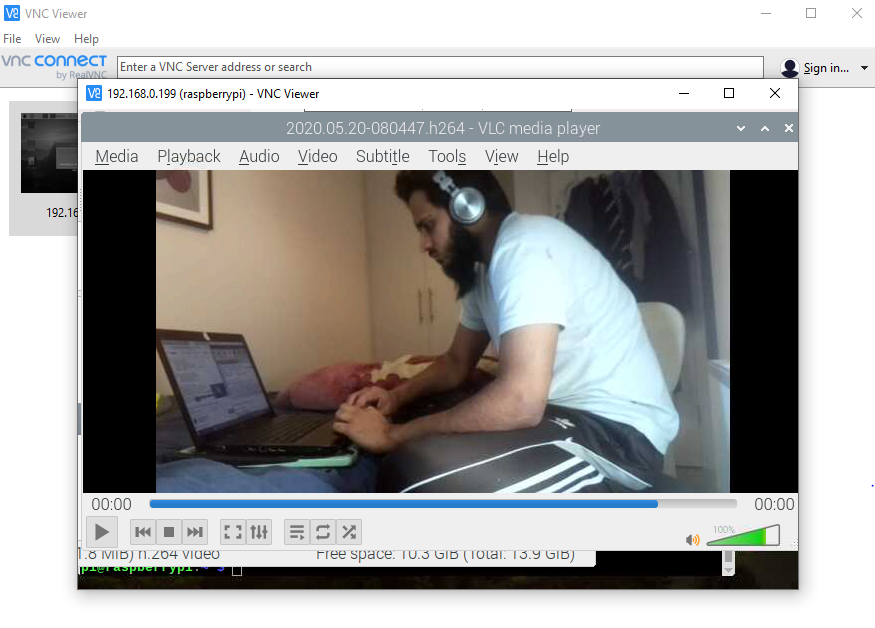


Figure 4‑: TimeStamp naming

It can be seen from the image above that this video is named as the time that video was taken. The naming difference can be seen from figure 4-3 and 4-2 presented above. The figure 4-2 was taken when the python code manually tells program to name video and it was named as motionVideo.h264, while figure 4-3 is taken after making python script to dynamically name video and is named as 2020.05.20.080447.h264. In the name it includes current date with year and month follow by current time including hours, minutes and second and at last ‘h264’ is the extension of video file.

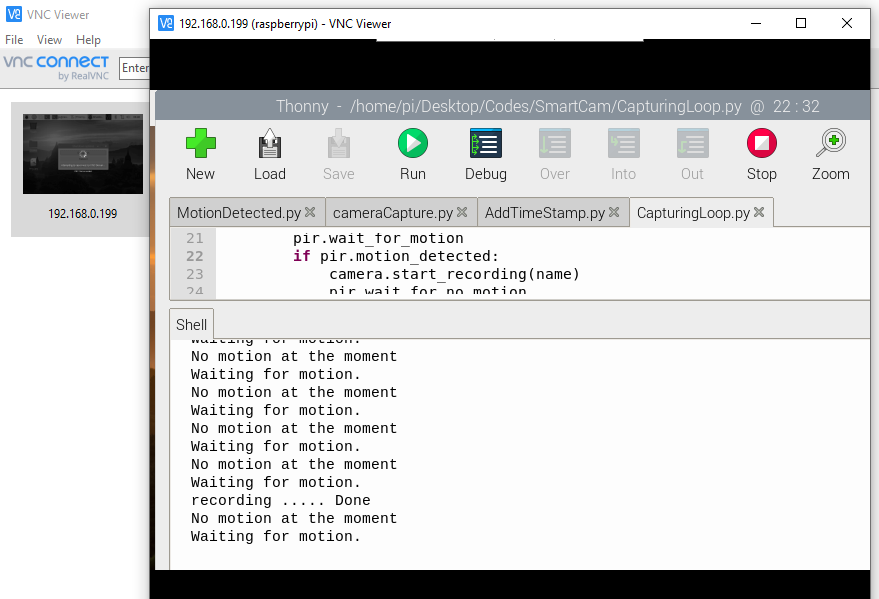


Figure 4‑: Program running & waiting for motion

Figure 4-4 is another image that shows the output when the program was running. This output is of the code that include loop for continuous motion detection and also TimeStamp added to it for naming file. In the figure 4-4, if we look at output section, we can say that the loop is working fine and first it was waiting for motion again and again and when it capture motion it start recording and when finish recording it again start waiting for motion and vice versa.

All the codes that I have written for different experiments and outputs I got from them will be published to GitHub repository and link of it is placed in appendix section at the end of this document and link is also uploaded to moodle as asked by college.

# Conclusion

The initial aim of this project was to explore uncountable projects that can be done using raspberry pi. Also aim of this project was to implement multiple technologies, studied during last four years in Bachelor of Science Honours in Information Technology and benefit from them.

For building a full-working system, it was necessary to deal with different hardware and operating systems, exploring python programming language and one of its amazing GPIO Zero library has provided me with several different tools that can be useful for me in future professional experiences.

The SmartCam project also provide me with the opportunity to deal with raspberry pi again and introduce me to two new hardware, motion sensor and camera module that can be useful in several different projects. It also gave me better understanding of python programming language, a language which I studied briefly, but know with this project I got opportunity to explore it and make use of it.

While recognising that there is a plenty of features that need improvement, especially enabling cloud service in my project, I believe that I have largely achieved the main objective of the problem by doing different experiments and a final prototype. I have learned a lot in order to accomplish this and this will be of huge help for me in future.

# References:

1. homeadvisor.com. (2019). [online] Available at: <https://www.homeadvisor.com/cost/safety-and-security/install-a-surveillance-camera/> [Accessed 7 Dec. 2019].
2. Is your CCTV system secure from Cyber attack. (2019). [ebook] CloudView (UK) limited. Available at: <http://www.cloudview.co/cyberattack> [Accessed 10 Dec. 2019].
3. Raspberrypi.org. (2020). [online] Available at: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/> [Accessed 13 Dec. 2019].
4. Raspberrypi.org. (2020). [online] Available at: <https://www.raspberrypi.org/products/camera-module-v2/> [Accessed 6 Jan. 2020].
5. Raspberry Pi. (2020). *Download Raspbian for Raspberry Pi*. [online] Available at: <https://www.raspberrypi.org/downloads/raspbian/> [Accessed 6 Jan. 2020].
6. Python.org. (2020). *Welcome to Python.org*. [online] Available at: <https://www.python.org/> [Accessed 6 Jan. 2020].
7. O'Connor, M. (2018). *10 Benefits of Cloud Computing | Typetec*. [online] Typetec.ie. Available at: <https://www.typetec.ie/blog/10-benefits-of-cloud-computing/?gclid=EAIaIQobChMI3_H05pnu5gIVibHtCh3FgAx4EAAYASAAEgKM5vD_BwE> [Accessed 6 Jan. 2020].
8. ownCloud. (2020). *✔ ownCloud - The leading OpenSource Cloud Collaboration Platform.*. [online] Available at: <https://owncloud.org/> [Accessed 6 Jan. 2020].
9. youtube.com. 2017. *How To Copy Files From Raspberry Pi To Windows Over Network*. [online] Available at: <https://www.youtube.com/watch?v=WIOpNuQc068> [Accessed 2 May 2020].
10. Harper, A., 2015. *Raspberry Pi Tutorial 12 - Take A Picture With Python*. [online] youtube.com. Available at: <https://www.youtube.com/watch?v=t5laKVNJd8U> [Accessed 14 April 2020].
11. Pi, R., 2020. *How To Setup Raspberry Pi Owncloud*. [online] Pi My Life Up. Available at: <https://pimylifeup.com/raspberry-pi-owncloud/> [Accessed 18 May 2020].
12. Nuttall, B., 2019. *GPIO Zero V1.5 Is Here! - Raspberry Pi*. [online] Raspberry Pi. Available at: <https://www.raspberrypi.org/blog/gpio-zero-v1-5/> [Accessed 19 May 2020].
13. Programiz.com. 2020. *Python If, If...Else, If...Elif...Else And Nested If Statement*. [online] Available at: <https://www.programiz.com/python-programming/if-elif-else> [Accessed 9 May 2020].
14. Projects.raspberrypi.org. 2019. [online] Available at: <https://projects.raspberrypi.org/en/projects/parent-detector/2> [Accessed 14 April 2020].
15. Harper, A., 2015. [online] Available at: <https://www.youtube.com/watch?v=XUFWE2RbeAc> [Accessed 20 April 2020].
16. Lough, B., 2018. [online] Available at: <https://www.youtube.com/watch?v=gOLnIrqmPQc> [Accessed 19 May 2020].
17. Circuit Basics. 2020. *How To Setup A Raspberry Pi Without A Monitor Or Keyboard*. [online] Available at: <https://www.circuitbasics.com/raspberry-pi-basics-setup-without-monitor-keyboard-headless-mode/> [Accessed 18 May 2020].

# Appendix

This is the link to GitHub repository where all the source code is. In the repository there, all the screen capture images taken has also been published.