**Interim Written Project Report**

****

**Group Members: Jason Lee, Dayoung Kim**

**Date: 3/17/23**

**Prof: Ben Shefler**

**Section: NBBL**

**Table of contents**

[**Executive Summary 4**](#_heading=h.gjdgxs)

[**Introduction 5**](#_heading=h.30j0zll)

[**Functional Features of the Product 6**](#_heading=h.1fob9te)

[**Specifications of the Product 7**](#_heading=h.3znysh7)

[**Operating Instructions 8**](#_heading=h.2et92p0)

[**Product Design, Implementation, and Operation of the System 9**](#_heading=h.tyjcwt)

[a. System block-diagram and software UML diagram 9](#_heading=h.3dy6vkm)

[b. Component images and components description 11](#_heading=h.2s8eyo1)

[c. Captures of the major GUIs that you created in your project (or those of third parties) 16](#_heading=h.1ksv4uv)

[d. Theory of operation of the entire system 18](#_heading=h.z337ya)

[Camera module 18](#_heading=h.1y810tw)

[Monitor 19](#_heading=h.4i7ojhp)

[Tri-LEDs 19](#_heading=h.2xcytpi)

[Pushbutton 20](#_heading=h.1ci93xb)

[**Maintenance Requirements 21**](#_heading=h.3whwml4)

[**Conclusion 22**](#_heading=h.2bn6wsx)

[**Further Developments 23**](#_heading=h.qsh70q)

[**Appendix 24**](#_heading=h.3as4poj)

[1) Electrical Schematics 24](#_heading=h.1pxezwc)

[2) Parts List/Bill of Materials 25](#_heading=h.147n2zr)

[3) List of all usernames in project 25](#_heading=h.23ckvvd)

[4) Quotations/Citations 26](#_heading=h.32hioqz)

[5) Contact Information 27](#_heading=h.1hmsyys)

**Table of Figures**

[Figure 1: System block-diagram 10](#_heading=h.1t3h5sf)

[Figure 2: software UML diagram 11](#_heading=h.4d34og8)

[Figure 3: Component images 12](#_heading=h.17dp8vu)

[Figure 4: Pi camera module 13](#_heading=h.3rdcrjn)

[Figure 5: Push button switch 14](#_heading=h.26in1rg)

[Figure 6: Monitor Display Screen 15](#_heading=h.lnxbz9)

[Figure 7: Tri-LED 16](#_heading=h.35nkun2)

[Figure 8: Source Code 18](#_heading=h.2jxsxqh)

[Figure 9: Electrical Schematic Connection (Display, Pushbutton & LED to Pi) 25](#_heading=h.49x2ik5)

[Figure 10: RPi Camera Module 26](#_heading=h.2p2csry)

**Lists of Tables**

[Table 1: Parts List/Bill of Materials 26](#_heading=h.3o7alnk)

[Table 2: Contact Information 28](#_heading=h.41mghml)

# **Executive Summary**

The “Automatic Smile Camera” provides quality pictures that capture the perfect facial expression of the users. Our goal is to avoid having to go through the hassle of having to take multiple pictures to get a single perfect shot. The device will automatically capture a photo of the users by taking a picture once they have made the appropriate face.The “Automatic Smile Camera” product is created by engineering students of Seneca College. The two students that created this device, Jason Lee and Dayoung Kim, who are the lead designers and creators of the project. Our users will have access to the standard feature of having a picture automatically taken while smiling, but other features such as choice of expression, zoom, manual mode. The target audience of our product could be effective for everyone, since everyone takes pictures of friends or family. We hope to make it easy to use and accessible for all, so no one has a bad picture anymore.

# **Introduction**

Taking a photo for the first time while everyone is smiling is almost impossible, that is why usually people take multiple pictures. On top of not being able to smile on the first capture, people tend to have pictures with their eyes closed. Our team has created a device to solve this problem and get a clean picture taken for the first time. The device's purpose is to capture a picture while everyone has the perfect facial expression they desire automatically. This product should allow the users to get that perfect shot on the first try.

# **Functional Features of the Product**

* Automated Camera the core of the device
* Monitor to display image of camera and facial expression software
* Camera detects facial features of users
* Camera captures picture during those facial features
* Able to change facial expression settings of the detection for the camera
* Able to set automated capture system for camera
* LEDs or Lights to brighten the picture when its being taken
* Pushbutton to take a picture manually

# **Specifications of the Product**

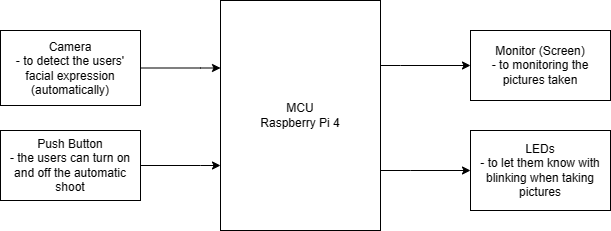
* Power Supply
  + To supply power to raspberry pi 4.
  + 5V/3A USB-C
* PC Application
  + Written in python
* Camera Module
  + To detect faces, recognize smiles, and take a picture automatically/manually with raspberry pi 4.
  + Product Dimensions: 22 mm x 24 mm x 9 mm
  + Product Weight: 3 grams
  + Maximum Photo Resolution: 2592 x 1944 pixel
  + Supported Video Resolution: 1080p30, 720p60 and 640x480p90
  + Interface: CSI MINI connect (15cm Ribbon Cable)
  + Supported OS: Raspbian
* Pushbutton
  + To take a picture manually when users press the pushbutton.
  + Size: 6 mm x 6 mm x 5 mm
  + Withstand Voltage: AC250V
  + Rated load: DC12V 50mA
* LEDs
  + To blink to let users know that the picture has been taken.
  + Type of head: Round
  + Forward Voltage: 2.0V to 2.2V
  + Current: 20mA
* Monitor
  + To allow users to see the process of taking pictures and the results.
  + Resolution
    - LCD I2C: 3.5 inch, 480 x 320 resolution

# **Operating Instructions**

1. Begin by pressing the button to turn on the device/program.
2. A live feed of the camera will appear on the monitor
3. Place the camera in front of the user/users
4. **If the users are smiling:**
   1. **The LED will turn from Red to Green.**
   2. **The monitor display will show a yellow box around their faces or smiles.**
   3. **A photo will then be taken.**
   4. **Will display picture for a few seconds.**
5. **If the users are not smiling:**
   1. **The LED will remain RED, indicating no picture has been taken.**
   2. **The monitor display will just have the same camera live feed**
   3. **A photo will not be taken.**
6. If the user wishes not to take an automatic picture, they can choose to take a manual picture by pressing the manual photo button
   1. The LED will turn from Red to Green.
   2. The photo will be displayed for a few seconds.

# **Product Design, Implementation, and Operation of the System**

## System block-diagram and software UML diagram

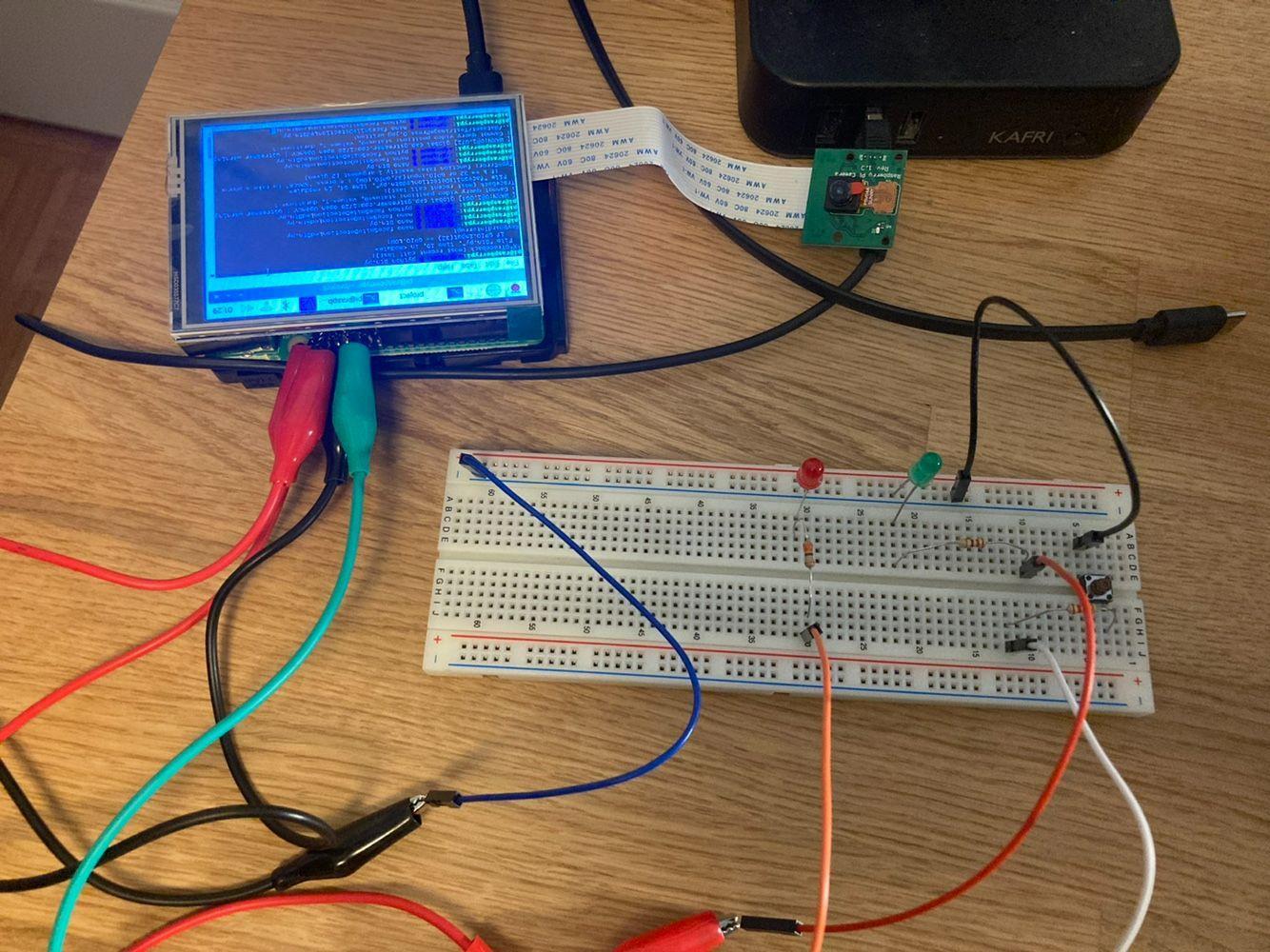


*Figure 1: System block-diagram*



*Figure 2: software UML diagram*

## Component images and components description



*Figure 3: Component images*

*Inputs:*



*Figure 4: Pi camera module*

**Pi Camera**

The raspberry pi camera is used to capture the photo of the users. This camera is allowed to use facial recognition to determine and detect when the users are smiling. Once the program is running the camera should produce a live feed on the monitor to allow the user to see what the camera is identifying. The users should be allowed to either choose to take a manual shot or an automatic photo. The camera pi is attached to a ribbon cable, which connects straight to the raspberry pi.



*Figure 5: Push button switch*

**Pushbutton**

The push buttons are used to start the program and to manually take a picture. To make the device more user friendly, we decided to add a simple push button to start the camera and the program running. We also decided to add a manual capture image button, because we found it would be redundant if we had a camera that only could take automated photos.

*Output:*



*Figure 6: Monitor Display Screen*

**Monitor**

The monitor is used to display the live feed coming from the camera. The live feed will also display exactly which faces its recognizing/detecting in a yellow square box. It is also a touch screen which could allow the user to interact with it. The main purpose of the monitor is to give the users a UI to interact with the device and view the photo that is being taken.



*Figure 7: Tri-LED*

**Tri-LED**

The Tri-LED is a LED with 4 Leads with one common cathode, and different anodes for each color. The common cathode is connected to the ground, while depending on which color desired to each anode. The colors used in this project are Red and Green. These colors are used to indicate towards the user if the photo is taken or not. Each one is connected to a different GPIO to select those colors when it is called in the code. In our case, GPIO 40 is for green and GPIO 38 is for red. The LEDs will be connected with a 330 Ohm resistor to prevent burn out.

## Captures of the major GUIs that you created in your project (or those of third parties)

## 

*Figure 8: Source Code*

## Theory of operation of the entire system

### 

### Camera module

The camera module is a OV5647 Sensor Video Webcam for Raspberry Pi. The camera provides 5MP Webcam Video 1080p and 30 fps for image capture speed. The camera is also capable of 2592 x 1944 pixel static images and connections included ribble cable to the CSI (Camera Serial Interface) port on Raspberry Pi. This camera is used to detect smiling faces and take pictures of the user automatically. The facial expression recognition uses OpenCV and the user can take pictures manually. Since OpenCV is downloaded in the Raspberry Pi OS (terminal), it can be imported by the **import cv2** command. In order to utilize the video function to show the video to the user, the **cv2.VideoCapture(0)** command should be used. A face recognition file must be downloaded to the Raspbian OS terminal to use the face recognition detection. The program uses the **cv2.CascadeClassifier** command to call the face recognition file with the file path. For detecting the smiling face, the **detectMultiScale** method and *for loop* that show the range of recognition use in the python code. Video is shown by calling the **cv2.imshow('video',img)** method on the LCD monitor and laptop screens. When the program detects the smiling face of the user, the **cv2.imwrite(img\_name, img)** method will save the pictures in the file and the name of the file will be saved to the path

**img\_name = "../dataset/"+ name + “/imageSmile\_{}.jpg".format(imgSmile\_counter)**.

### Monitor

The monitor LCD I2C is a 3.5 inch standard Display Touch Screen for Raspberry Pi. The LCD provides a resolution of 480 x 320 dots offering a clear and vivid image. The LCD is also compatible and can be directly plugged into all versions of Raspberry Pi motherboards. It can be used to download the official system for Raspberry Pi. Video is shown on the LCD monitor by calling the **cv2.imshow('video',img)** method. The user can execute the code to operate the automated smile camera and check the result of the photos through the embedded Touch Screen.

### Tri-LEDs

The LEDs were created to inform the user about the current operating state of the camera module by declaring the **GPIO.setup(*pin\_number*,GPIO.OUT,initial=GPIO.LOW)** command. In this code, two colors of LEDs were used - green and red. If the LED turns red, it means the automated smile camera is working and the camera is ready to detect smiling faces and take pictures manually at any time. On the other hand, when the user takes a picture, the LED turns on from red to green for 2 seconds, by the **sleep(2)** command**,** and then returns back to red, by the **GPIO.output(*pin\_number\_green*,GPIO.LOW)** and **GPIO.output(*pin\_number\_red*,GPIO.HIGH)**. The function **"../dataset/"+ name +"/imageSmile\_{}.jpg".format(imgSmile\_counter)** sets the file path for the picture and allows the name of the photo to be saved as *imageSmile\_{},jpg*.

### Pushbutton

The pushbutton was designed to provide the user with a manual photo option by declaring the “**GPIO.setup(*pin\_number*,GPIO.IN,pull\_up\_down=GPIO.PUD\_UP)”** command. The function “**if k == 32 or GPIO.input(32) == GPIO.LOW :”** makes the user take a picture manually if they press the spacebar on the keyboard or pushbutton with the color change of the LED. Similar to automatic capture, the **"../dataset/"+ name +"/imagePress\_{}.jpg".format(imgPress\_counter)** sets the file path for the picture and allows the name of the photo to be saved as *imgPress\_{}.jpg*.

# **Maintenance Requirements**

The maintenance requirements for the device are minimal. The requirements to maintain this device are simple checks or clean ups:

1. The device will be required to charge, such as the main core of it, which would be the raspberry pi 4.
2. The raspberry pi 4 itself will not require any updates, and is recommended to stay on the current patch it's on as the coding is compatible with the specific patch.
3. The device will be required to be cleaned after long periods of being unused to prevent dust build up. Specifically the monitor and the camera, as those are the two main features of the product.
4. To ensure the LEDs are not burning out and possible replace after long term use to prevent dimming or burning out the LED.

# **Conclusion**

Capturing that perfect smile in a picture should not be as difficult as it should be. The “Automated Smile Camera” should accomplish this task with ease. Many people at all ages can benefit from a device created to assist in building strong moments with your friends and families. This project also acts as an assist tool to those who may not be able to use a regular camera. The device should recognize the users faces and take the shot once everyone is making the correct facial expression. It will fulfill the needs for the user without having to rely on the undependable human eye and timing to create the perfect photo.

# **Further Developments**

The “Automated Smile Camera” has potential for more unique features that could be added to make it a more defining product. Our team produces the required aspects of the project, this could be surpassed but due to limiting things such as partner distancing, cost, and time management with other courses, the project is only at its core functions. If my partner and I had more time to dedicate towards the project and extra resources we could add additional features to the camera. In our current project, we feel that the project lacks a lot of unique features but although it works to its fullest, shines in its core aspects but lacks in varying components.

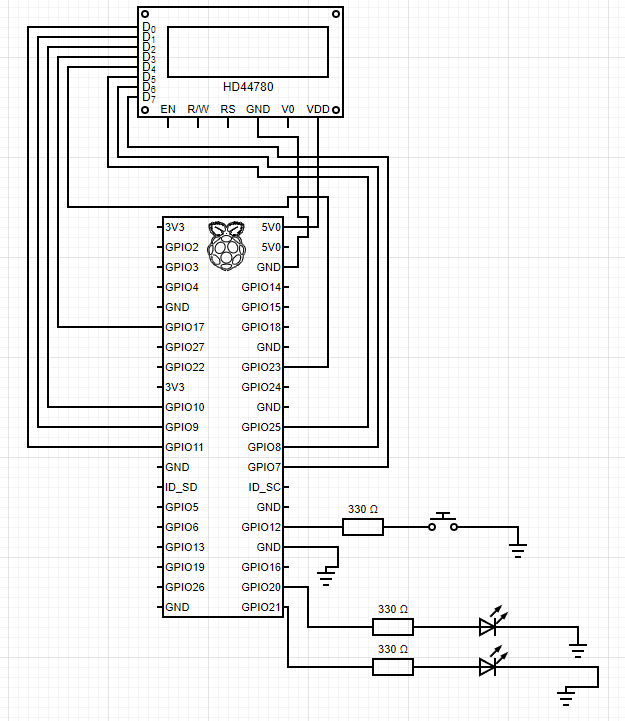
In the eyes of a product stand point, it would be difficult to sell our product, this is due to the competitive market in this kind of specific application. My partner and I agree that making our device smaller would help with being a more modern device, as of now our product is bulky and big. This would just be an adjustment in design choice if we had more time and resources to this specific project.

The project itself was a great learning curve to understand, and from the things our team has completed. The group has learned that the importance of a finished product is to add the minor details to clean it up. Things such as the look of the enclosure, the buttons of the device, labeling of the device, etc.

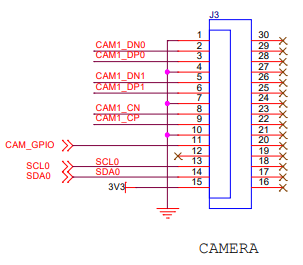
For any future development for this product, will come from not just adding features and refining the overall look for the device, but to market it better. Since the product application itself is in a popular market, it is important to try and stand out in a crowd of similar products. The product should slowly adapt into a more casual product that can be accessible to everyone.

# **Appendix**

## Electrical Schematics



*Figure 9: Electrical Schematic Connection (Display, Pushbutton & LED to Pi)*



*Figure 10: RPi Camera Module*

## Parts List/Bill of Materials

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Quantity** | **Unit Price (CAD)** | **Total Cost of Units** |
| Pi 4 4GB Starter Kit - 32GB | 1 | $119.95 | $119.95 |
| 5MP Webcam Video 1080p Camera | 1 | $14.45 | $14.45 |
| Pushbutton Switch | 1 | $0.08 | $0.08 |
| 5mm Led Lamp | 1 | $0.33 | $0.33 |
| LCD IC2: 3.5 Inch, 480 x 320 resolution | 1 | $32.98 | $32.98 |
| Total |  |  | $167.79 |

*Table 1: Parts List/Bill of Materials*

## List of all usernames in project

N/A

## 

## Quotations/Citations

Aamir, H. (2018, January 31). *Raspberry Pi 3 Complete Tutorial.* Trinkiknow <https://trickiknow.com/raspberry-pi-3-complete-tutorial-2018-lets-get-started/>

Adams. (2019). SCHEMATIC1. AJ, Raspberry Pi HQ. <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-reduced-schematics.pdf>

Camera Module [image], by KEYESTUDIO, (Figure 5), (n.d.)

<https://www.amazon.ca/Keyestudio-Camera-Module-5MP-Raspberry/dp/B073RCXGQS/ref=asc_df_B073RCXGQS/?tag=googleshopc0c-20&linkCode=df0&hvadid=292968375828&hvpos=&hvnetw=g&hvrand=14733550601260397024&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9000786&hvtargid=pla-569182936534&th=1>

Circuit Diagram. (2021). Circuit Diagram Editor. <https://www.circuit-diagram.org/editor/>

Core Electronics (2021, July 5). *Face Recognition With Raspberry Pi + OpenCV + Python* [Video]. YouTube <https://www.youtube.com/watch?v=o-x1PE0LVKM>

Multicolor LED [Image], by RoboticsBD, (Figure 7), (2023). <https://store.roboticsbd.com/led/1591-rgb-2-pin-led-5mm-pack-of-5-robotics-bangladesh.html>

Pushbutton Switch [Image], by DAOKI, (Figure 6), (2016). <https://www.amazon.ca/C-J-SHOP-Miniature-Momentary-Tactile/dp/B01CGMP9GY/ref=sr_1_7?keywords=Push+Button+Switch&qid=1678847487&sr=8-7>

Raspberry Pi. (2018). Raspberry Pi Documentation, Raspberry Pi HQ. <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>

Rose (2019, January 9). *Real-time Face Recognition Camera.* <https://blog.naver.com/PostView.naver?blogId=ljy9378&logNo=221438192568&redirect=Dlog&widgetTypeCall=true&directAccess=false>

Soren (2018, January 11). *Making a LED blink using the Raspberry Pi and Python.* Raspberry Pi HQ. <https://raspberrypihq.com/making-a-led-blink-using-the-raspberry-pi-and-python/>

Soren (2018, February 8). *Using a push button with Raspberry Pi GPIO*. Raspberry Pi HQ. <https://raspberrypihq.com/use-a-push-button-with-raspberry-pi-gpio/>

Tim (2023, February 16) *Face Recognition With Raspberry Pi and OpenCV.* core electronics <https://core-electronics.com.au/guides/face-identify-raspberry-pi/>

## Contact Information

|  |  |  |  |
| --- | --- | --- | --- |
| Names | Email | Number | Others |
| Jason Lee | jlee642@myseneca.ca | 647-640-9310 |  |
| Dayoung Kim | dkim175@mysneca.ca | 416-220-9231 |  |

*Table 2: Contact Information*