

# Cutepie (Ver.1) Desktop Assistant Non-Official Documentation

**School Project for a gateway-education course**

**Poon Ting Kwok**

<b>Introduction.....</b>	<b>2</b>
Background .....	2
<b>Schedule and Planning .....</b>	<b>3</b>
Production schedule .....	3
List of Equipment/Resources .....	4
1.Hardware .....	4
2.Software .....	4
3.Language .....	4
Finalized Set Up Plan.....	5
1.Goals and expectations.....	5
<b>Design Documentation.....</b>	<b>6</b>
Product design Concept Art .....	6
Interactivity .....	6
<b>Prototype Design and Implementation .....</b>	<b>7</b>
Paper Prototype .....	7
Physical Prototype Building Process .....	7
Preparation .....	7
The building of prototype .....	7
Software prototype.....	12
<b>Future development .....</b>	<b>15</b>
<b>Reflection .....</b>	<b>15</b>
<b>Reference .....</b>	<b>16</b>
Arduino and grove hardware and coding .....	16
Python Serial communication .....	17
OpenCV Python.....	17
Machine learning Python .....	17
Physical prototype building .....	17

# Introduction

## Background

Nowadays, we have entered the information age. People all over the world are bonded with the use of computers. Over the past sixty years, artificial intelligence is developing rapidly. Researches all over the world are discovering the ways to control the computers so that they can imitate human behaviours, such as speaking, thinking, logical reasoning, etc. The use of A.I. is spreading like wild fire, and its application can be seen across different disciplines and fields. For example, business sectors use A.I. technology to predict customer behaviours and formulate a business model base on customers' preference to maximize profit. Popular browsers such as Google predict users' behaviours by collecting search records and users' speech so that Google could modify search results and prompt the advertisements that users might be interested. A friend of mine who is a City University of Hong Kong's PhD student in biomedical science use machine learning techniques to analysis lab research results. As a student majoring in computer science, I always fascinated about how A.I. technology works, and how it can be used to benefit people's life.

As a student, sometimes we need to sit in front of our laptops, work on our assignments and revise for our examination on our own unavoidably. This is also true for people working in the offices. In busy days, they need to work around the clock and stay in there sits for a long period of time. If there is a cute physical device that accompanies with people and being a helpful assistant like Siri, I am sure that people will be able to work with joy.

Here comes to my idea: Building a cute and small robot assistant that can help people at work, and I call it 'CutiePie' (V1).

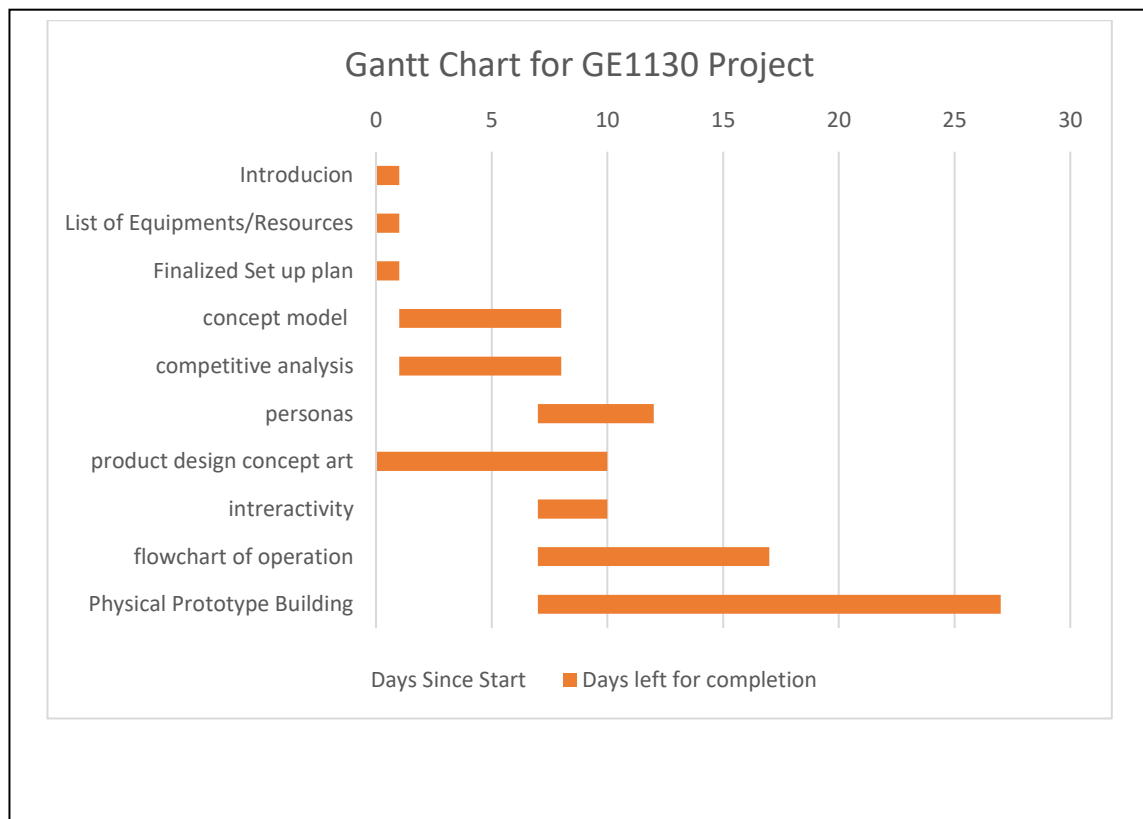
## Schedule and Planning

Having a good planning is crucial so that I do not need to burn to mid-night oil to get things done before the deadline.

### Production schedule

With the use of excel table, I generated the following schedule.

The starting date of this Gantt Chart is 24 March 2019. It is helpful for me to monitor my progress and know what should be done for the project.



## List of Equipment/Resources

The making of Cutiepie requires both hardware and software.

### 1. Hardware

1. Arduino Uno board (qty: 1)
2. Wires (qty: numerous)
3. Web cam (Logitech C270 Desktop/Laptop webcam) (qty: 1)
4. Servo Motors MG995 (qty: 2)
5. Pan and tilt device framework (qty: 1)
6. Converter (Purpose: convert incoming voltage to 5V) (qty: 1)
7. Lithium Battery (qty: 1)
8. A laptop (qty: 1)
9. Resistors (qty: numerous)
10. LED light bulb (qty: 3)
11. Breadboard (qty: 1)
12. USB type-A cable (qty: 1)
13. Cardboard (qty: numerous)
14. Tape & hot glue gun (qty: numerous)
15. Grove expansion shield (qty: 1)
16. Grove Buzzer (qty: 1)
17. Grove LCD RGB backlight (qty: 1)

### 2. Software

1. JetBrains PyCharm Community Edition 2019.1.1
2. Arduino IDE

### 3. Language

1. Python
2. C++

## Finalized Set Up Plan

### 1.Goals and expectations

CutiePie is expected to achieve the following main goals:

- To accompany with people who are working or studying so as to reduce stress and loneliness and give care to them.
- To observe people's feelings and make responses for relaxation.
- To inform people about essential information, such as weather conditions, new messages, etc.

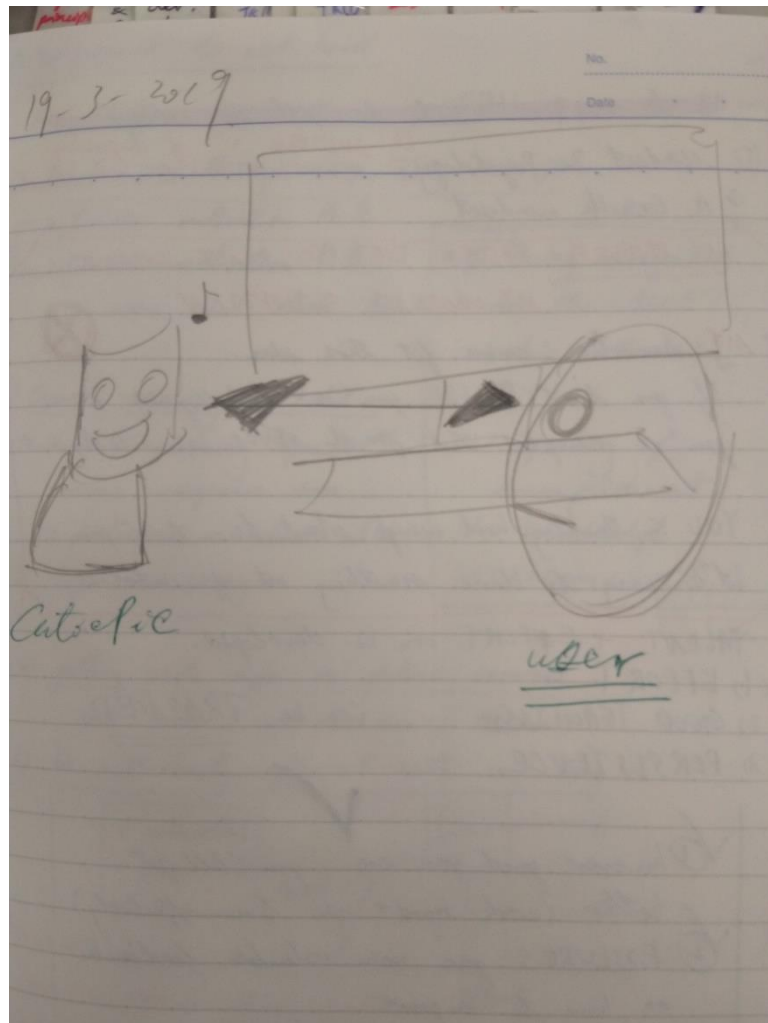
CutiePie is expected to have the following characteristics:

- It is easy to be used by people.
- It is small and able to be placed on the desk.
- It looks cute and friendly

# Design Documentation

## Product design Concept Art

Here is a very brief concept art about my design of the robot.



## Interactivity

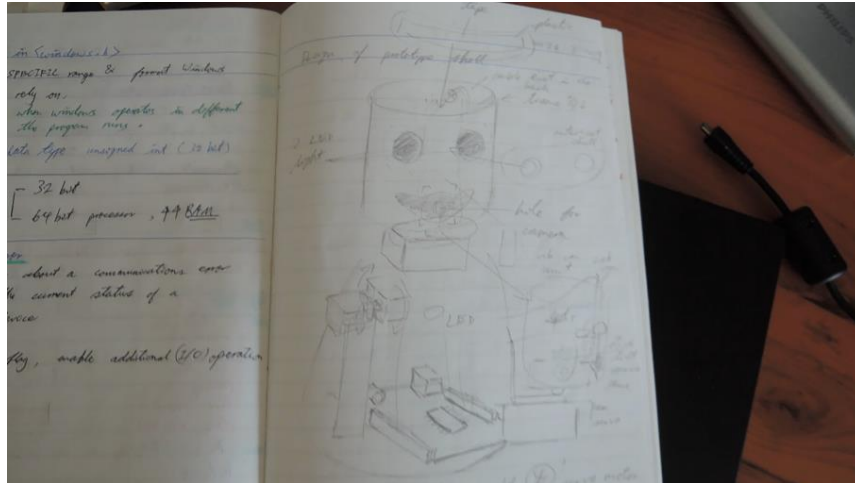
The details of the expected interactivities of CutiePie with users are demonstrated below:

1. When user enters the field of view of the webcam, CutiePie's head will move with accordance to the user's head location.
2. When CutiePie detect that the user is sad, it will play a warning tone and the LCD display will show some texts to cheer user up.
3. When CutiePie detect that the user is happy, it will play happy tone and display happy message.
4. When CutiePie receive incoming message, it will inform the user by play a tone.

# Prototype Design and Implementation

## Paper Prototype

During the first stage of the production process, I have come up with the initial design and sketched it in my notebook.



## Physical Prototype Building Process

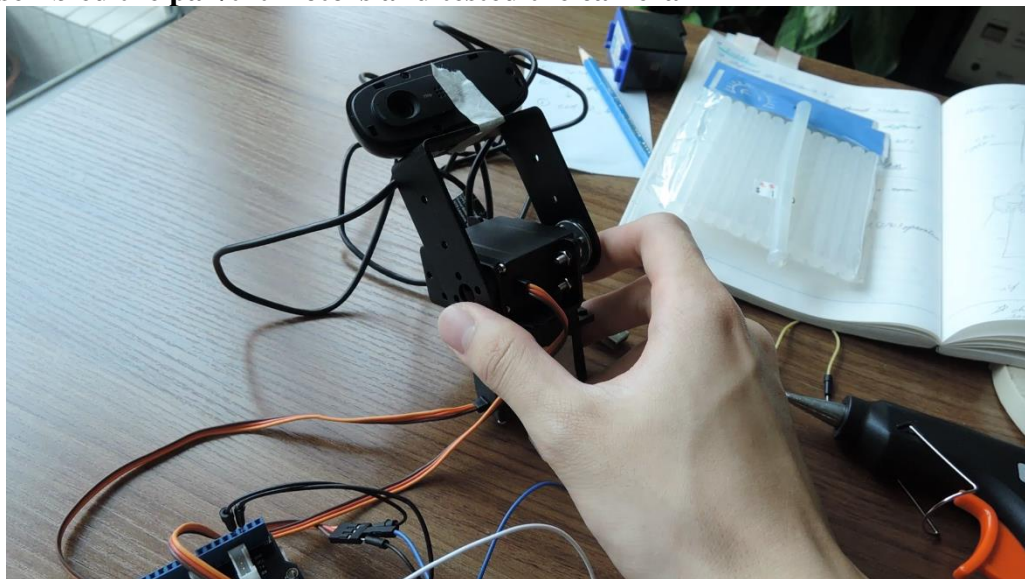
### Preparation

I found a wasted cardboard box which is sturdy enough to build the structure of the robot. I then bought all the necessary electronic components, including wires, resistors, sockets, battery, breadboards, LEDs, etc. Then, I purchased a hot glue gun for the assembly of parts.

### The building of prototype

Here are the photos showing the building process.

### Step 1: assembled the pan/tilt motors and tested the camera





**Step 2: Prepare the working environment and get all the parts.**

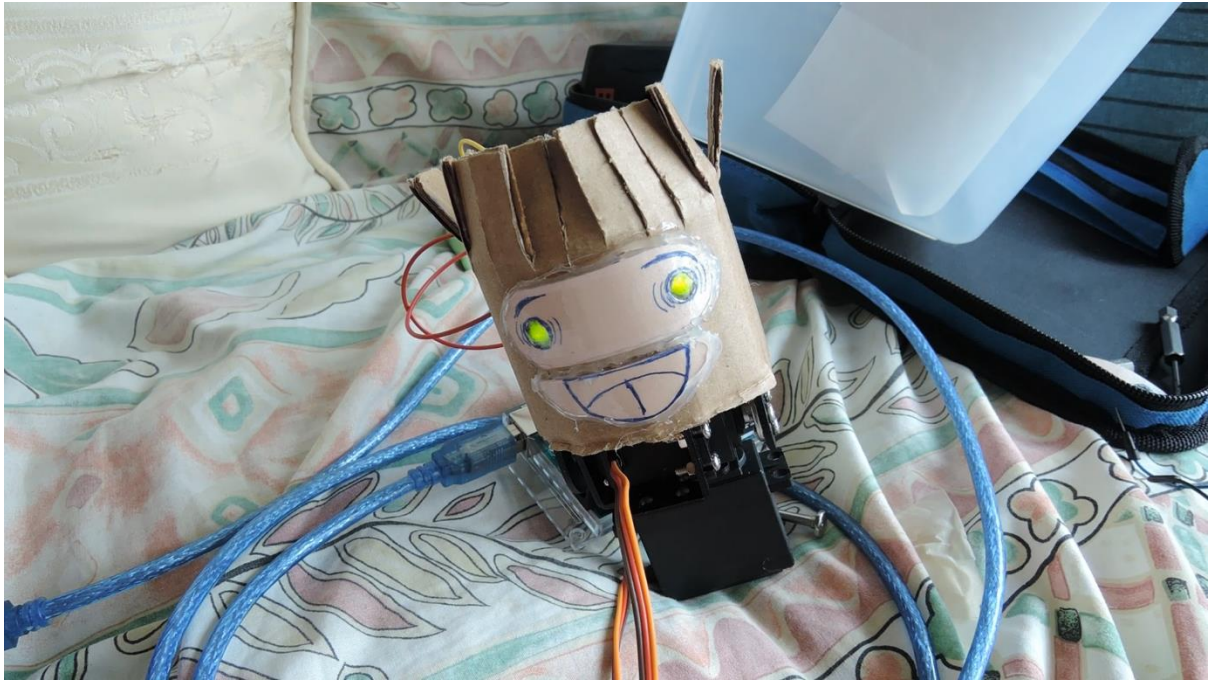


**Step 3: Built the basic structure of the head of Cutiepie.**



**Step 4: First testing of robot in progress**

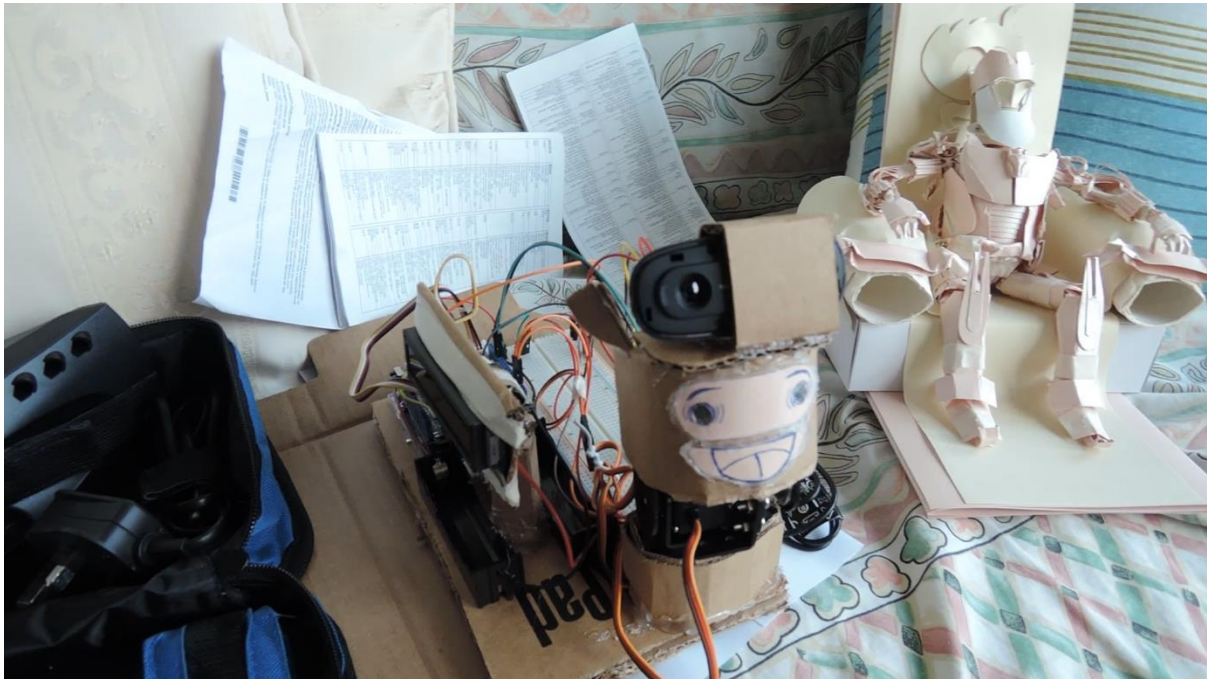




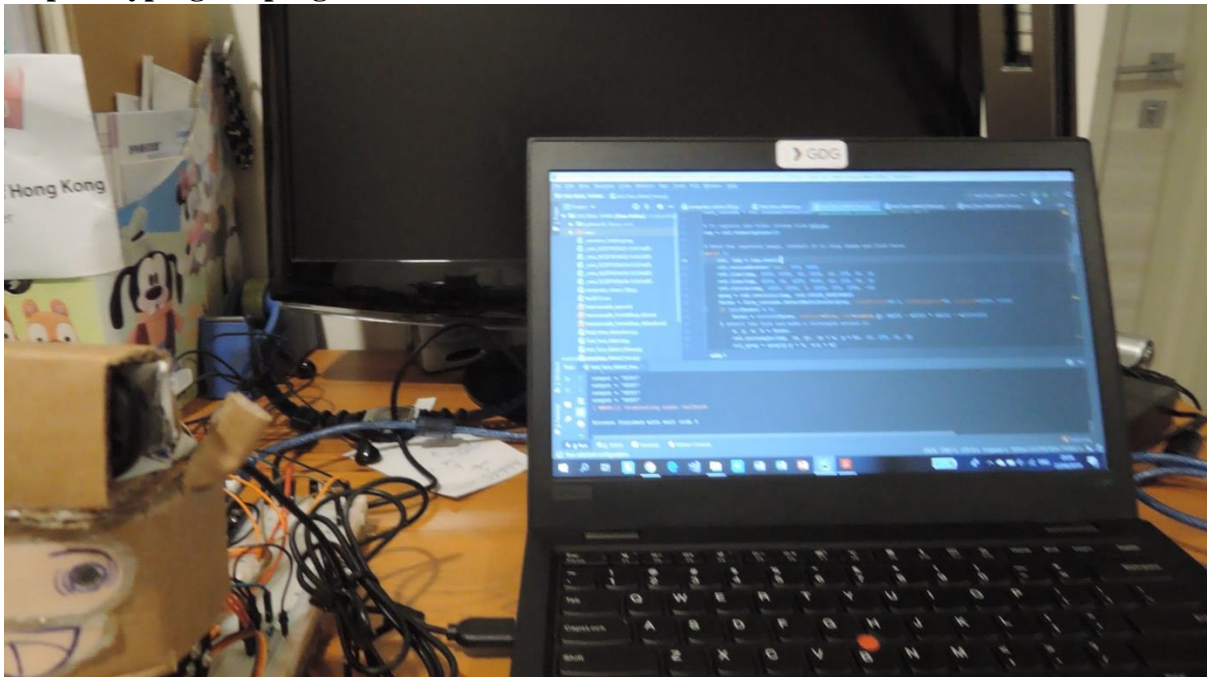
**Step 5: Building in progress. My living room was a mess.**



**Step 6: Completion of the structure of the robot and second testing of robot was conducted.**

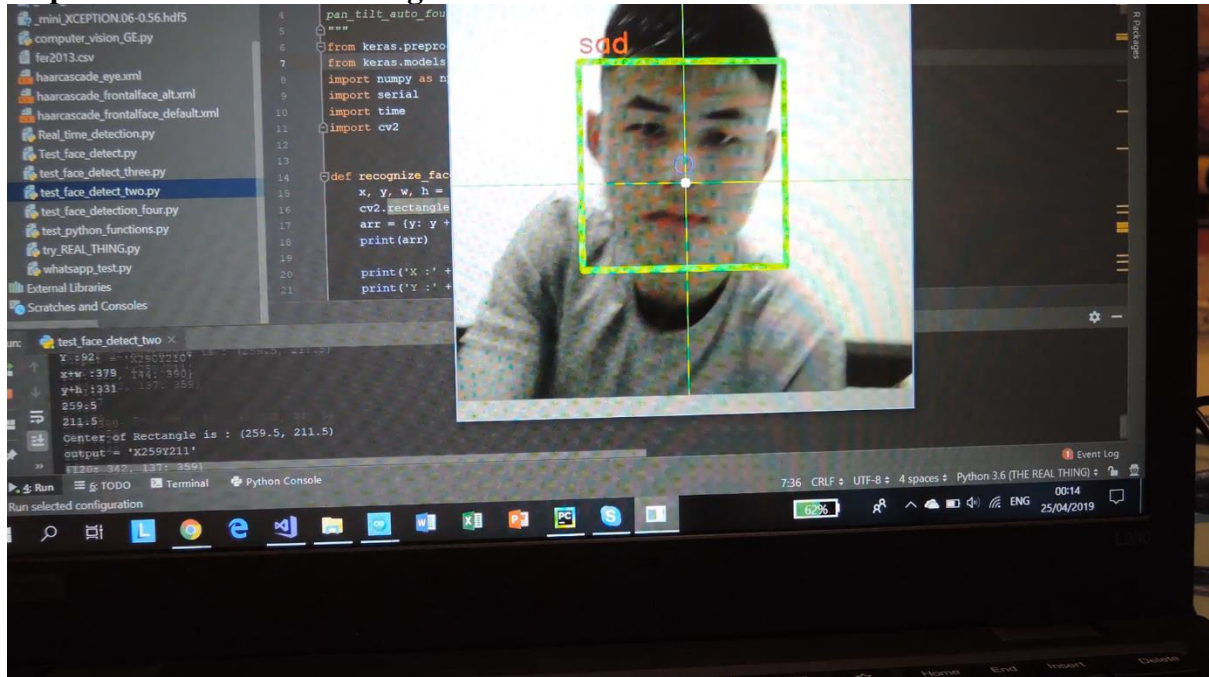


**Step 7: Typing the program**

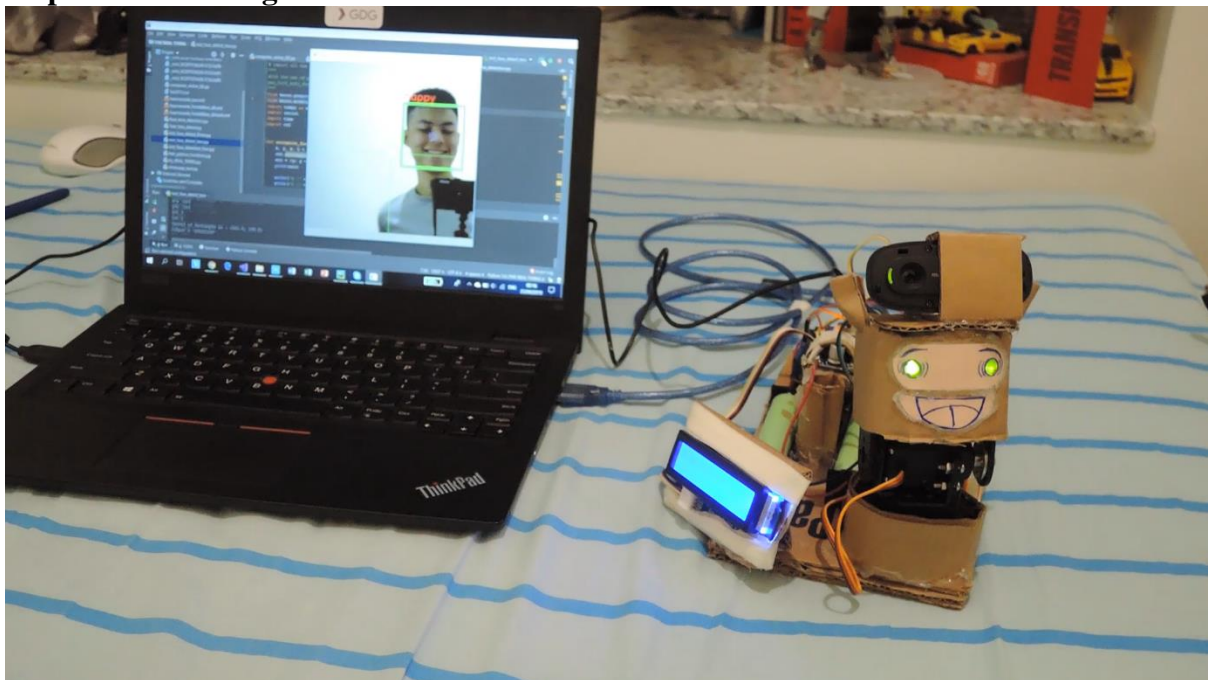




## Step 8: Achieved face tracking and emotion detection



## Step 9: Final testing of robot was conducted

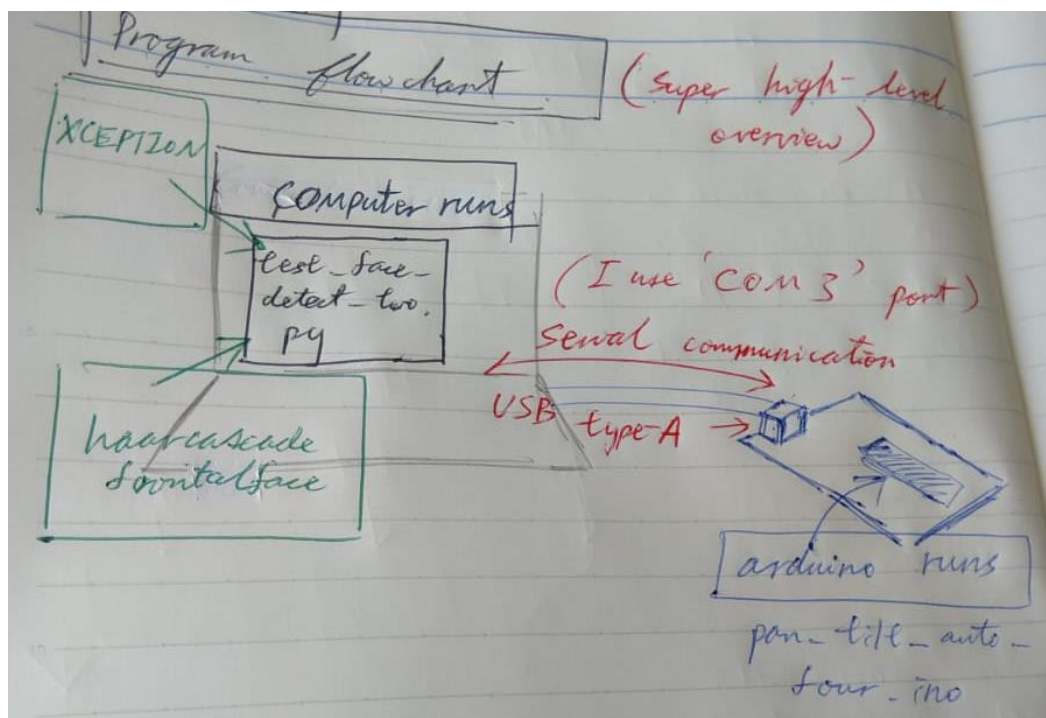


## Software prototype

In this section, I will demonstrate how the software was developed. The whole development process can be divided into several stages.

Stage	Description
Research	I search for several websites and yielded information that could be useful for my project.
Set up reasonable goals that I want my program to achieve	I defined my goals clearly to produce the program which is suitable for my project
Understand and implement serial communication	This is an important stage since the robot cannot work if serial connection between the computer and Arduino UNO fails.
Study Arduino Library tutorials and try sample code	I need to understand how to control the digital output from the Arduino.
Study OpenCV documentation and read sample code in the Internet	OpenCV library is used for computer vision tasks.
Study online tutorials for machine learning for face detection program	Since I am an absolute newbie to machine learning, I need to study sample code in order to understand how to implement the machine learning concepts.
Finalizing the program	The finalized design of software is to make one program for Arduino and one program for the face and emotion detection task

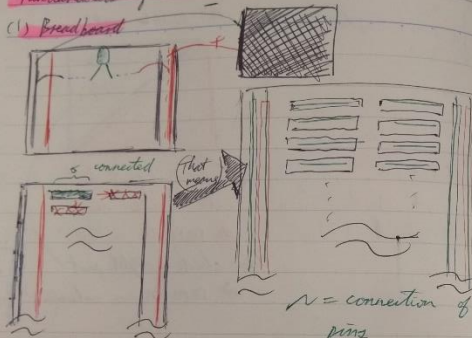
I would like to display some of my notes during the research process.



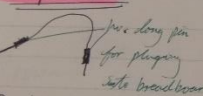
L2

Fundamentals of electronics + opencv + circuit

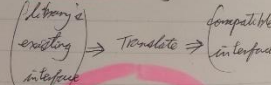
(1) Breadboard



Samper wife.



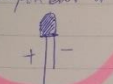
Wrapper Library



Brief design of robot

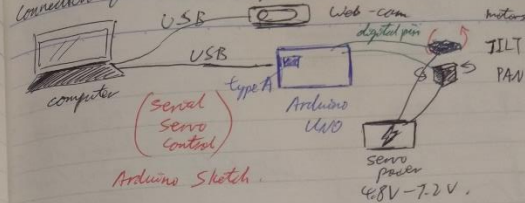


London met #

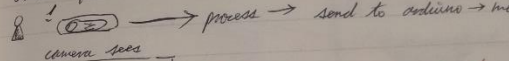


## hard words

connection for the GE project



Overall flock



Arduino code

Purpose - creates a serial  
interface to control  
robot

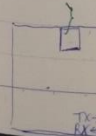
Counter code

purpose:- read webcam  
- detect  
- control robot (v  
interface)

Sensal communication

set up serial port


```
if (Serial.available() > 0) {
    ... (receiving !!)
```



pin 0 = RX } for UART transmission  
pin 1 = TX } universal asynchronous receiver  
transmission

R: receive use pin 1, 0 to connect  
T: Transmitter with other component (Use

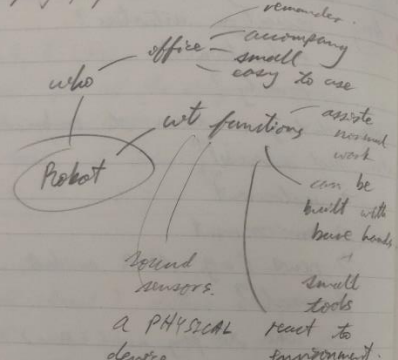
L 3

\* (color band)! —    
 e.g. 68  $\Omega$     blue grey black gold.

LED

BTW, WHY you make this?

In the age of AI & IoT technology,  
It has a lot of potential  
, helping people  
working



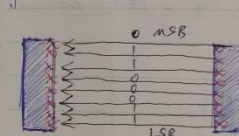
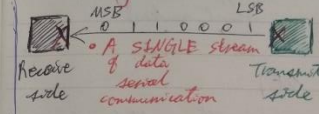
real-life interaction,  
easy to use, helpful  
accompany with people,  
assist people in work

\* Including OpenCV in IDE! (Visual Studio)

- ① C/C++ → General → Additional Include Directory  
... \opencv\build\include
  - ② Linker → Additional Library directory  
... \opencv\build\x64\vc15\lib
  - ③ Linker → Input → Additional Dependencies  
opencv\_world400d.lib
- DONE!

What is sexual communication

- sending data 1 bit at a time sequentially



parallel

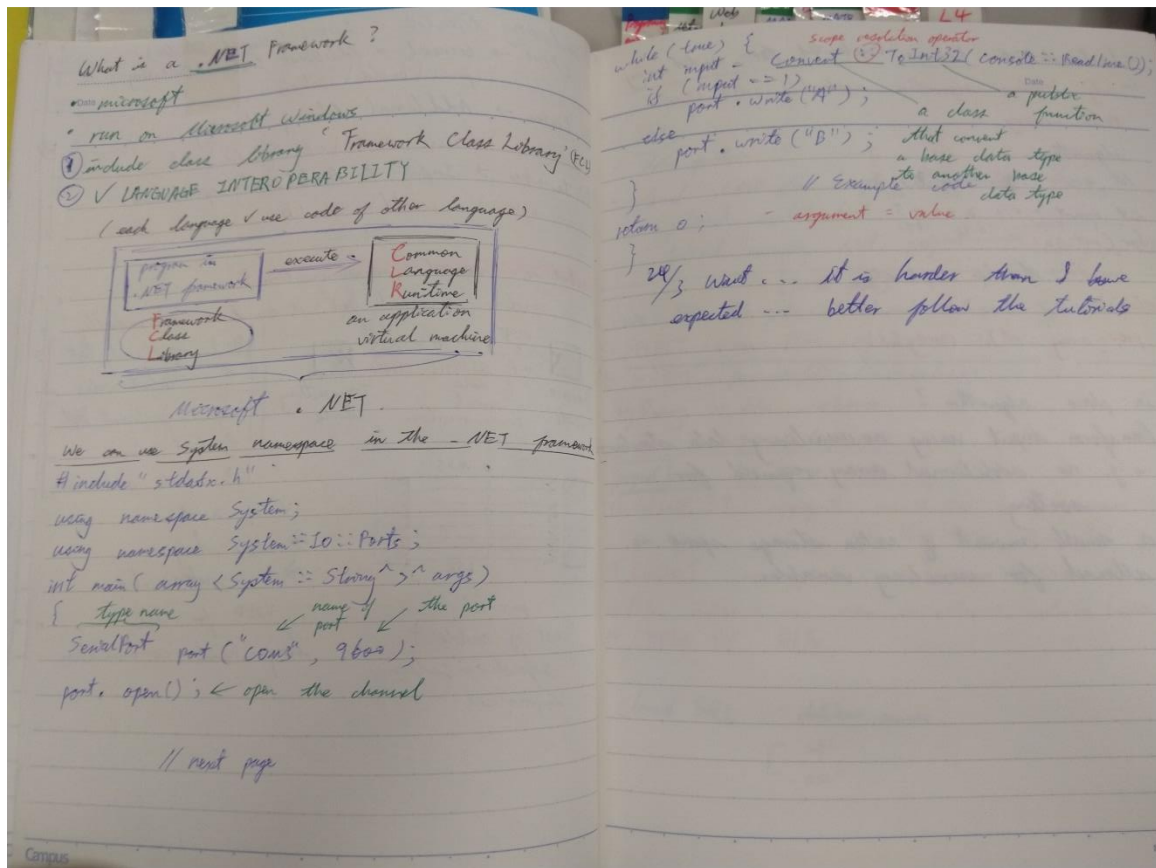
- ① cost of cable  $\uparrow$
- ② synchronization

impractical

- ✓ of serial
- ⇒ Cable cheaper
- ⇒ X synchronization

∴ USED for all long-haul comm





All the code and files are inside the Github repository. The link is:

[https://github.com/MRSlothAlan/CutiePie\\_version\\_1\\_code](https://github.com/MRSlothAlan/CutiePie_version_1_code)

## Future development

Right now, the prototype has achieved the face tracking, face detection and emotion detection function. However, it can be further improved as the third goal of the project, which is 'To inform people about essential information, such as weather conditions, new messages, etc.', is not yet achieved.

1. I would like to include notification function, so that when a new message is received, CutiePie is expected to inform their users.
2. The physical size of CutiePie should be further reduced and the new design should fully encapsulate the whole internal structures, including the microprocessor, motors and wires. A computer with smaller physical size, such as RaspberryPi, should be used.
3. The face and emotion detection program are not highly accurate. Further research can be done in order to learn and implement better methods for face detection.

## Reflection

After making this project, I have a clearer picture about computer vision and machine learning and it provoke my interest in these field of study. It is a fun process to build something new from scratch and this is really an unforgettable experience.



## Reference

I have searched more than 200 websites, and I would like to list out several useful ones in MLA citation format. Every effort has been made to trace copyright, but in the event of any accidental infringement, I shall be pleased to come to a suitable arrangement with the rightful owner.

I am grateful to the following people who like to share good stuff with others.

1. **Adrian at 'PyimageSearch'**: He provides several examples about how to use the OpenCV library for computer vision project, which is inspiring, and I have learnt some applications of computer vision.  
Website: <https://www.pyimagesearch.com/>
2. **Jeff Goergen**: At the very first stage of the design process, I found his YouTube channel and I am fascinated about his project. Although I have not examined his code, he provided a list of hardware used and provide a brief diagram about how the components are connected in his Github repository. I finally bought the MG995 servo motors and a Pan/Tilt assembly, and a webcam online.  
His channel: [https://www.youtube.com/channel/UCg8RE6u8D5YrNh\\_cvSdOajw](https://www.youtube.com/channel/UCg8RE6u8D5YrNh_cvSdOajw)
3. **ABHIJEET KUMAR**: His article named '*Demonstration of Facial Emotion Recognition on Real Time Video Using CNN: Python & Keras*' provides the method to train a model and make use of the model to predict the most likely emotion of the user. His work is awesome, and I really appreciate that he shares his work to all the people across the globe. I am now having a brief understanding about how machine learning is like and I will keep investigating when I have time.  
His website: <https://appliedmachinelearning.blog/author/abhijeetchar/>

The following is a list of reference for my project. I have divided the list into several sections and each section contains the list of references for a research area.

### Arduino and grove hardware and coding

"A Video Overlay Shield for Arduino." Open Electronics, [www.open-electronics.org/a-video-overlay-shield-for-arduino/](http://www.open-electronics.org/a-video-overlay-shield-for-arduino/).

Bill. "Grove - Buzzer." Seeedstudio, [wiki.seeedstudio.com/Grove-Buzzer/](http://wiki.seeedstudio.com/Grove-Buzzer/).

Bill. "Grove - 16x2 LCD Series." Seeedstudio, [wiki.seeedstudio.com/Grove-16x2\\_LCD\\_Series/](http://wiki.seeedstudio.com/Grove-16x2_LCD_Series/).

"Home." Arduino, [www.arduino.cc/](http://www.arduino.cc/).

Instructables. "FACE TRACKING USING ARDUINO !!!" Instructables, Instructables, 20 Apr. 2019, [www.instructables.com/id/FACE-TRACKING-USING-ARDUINO-/](http://www.instructables.com/id/FACE-TRACKING-USING-ARDUINO-/).

Instructables. "How to Easily Play Music With Buzzer on Arduino (The Imperial March - STAR WARS)." Instructables, Instructables, 19 Oct. 2017, [www.instructables.com/id/How-to-easily-play-music-with-buzzer-on-arduino-Th/](http://www.instructables.com/id/How-to-easily-play-music-with-buzzer-on-arduino-Th/).

Kazarinoff, Peter D. "Using Python to Control an Arduino." Python for Undergraduate Engineers, 20 Dec. 2018, [pythonforundergradengineers.com/python-arduino-LED.html](http://pythonforundergradengineers.com/python-arduino-LED.html).

### Python Serial communication

infinf 19.1k96494, and TJDTJD 10.2k11930. “Serial Port: HW Handshaking Turned on Only on Sender Side, Transmission Still Works.” Stack Overflow, [stackoverflow.com/questions/12395997/serial-port-hw-handshaking-turned-on-only-on-sender-side-transmission-still-wo](https://stackoverflow.com/questions/12395997/serial-port-hw-handshaking-turned-on-only-on-sender-side-transmission-still-wo).

WaveShapePlay. “Arduino and Python Serial Communication with PySerial Part 1.” YouTube, YouTube, 15 June 2018, [www.youtube.com/watch?v=iKGYbMD3NT8](https://www.youtube.com/watch?v=iKGYbMD3NT8).

### OpenCV Python

“Face Detection Using Haar Cascades.” OpenCV, [docs.opencv.org/3.4.2/d7/d8b/tutorial\\_py\\_face\\_detection.html](https://docs.opencv.org/3.4.2/d7/d8b/tutorial_py_face_detection.html).

“OpenCV Documentation Index.” OpenCV Documentation Index, [docs.opencv.org/](https://docs.opencv.org/).

“Pan/Tilt Face Tracking with a Raspberry Pi and OpenCV.” PyImageSearch, 22 Apr. 2019, [www.pyimagesearch.com/2019/04/01/pan-tilt-face-tracking-with-a-raspberry-pi-and-opencv/](https://www.pyimagesearch.com/2019/04/01/pan-tilt-face-tracking-with-a-raspberry-pi-and-opencv/).

### Machine learning Python

abhijeet3922. “abhijeet3922/FaceEmotion\_ID.” GitHub, 7 Jan. 2019, [github.com/abhijeet3922/FaceEmotion\\_ID](https://github.com/abhijeet3922/FaceEmotion_ID).

“Demonstration of Facial Emotion Recognition on Real Time Video Using CNN : Python & Keras.” Machine Learning in Action, 28 Nov. 2018, [appliedmachinelearning.blog/2018/11/28/demonstration-of-facial-emotion-recognition-on-real-time-video-using-cnn-python-keras/](https://appliedmachinelearning.blog/2018/11/28/demonstration-of-facial-emotion-recognition-on-real-time-video-using-cnn-python-keras/).

### Physical prototype building

Goergen, Jeff. “Cambot! Servo Controlled Motion Tracking Camera.” YouTube, YouTube, 9 Nov. 2017, [www.youtube.com/watch?v=nbJNLVBo\\_g4](https://www.youtube.com/watch?v=nbJNLVBo_g4).