Smart Glasses, a new way to interact within the world

Alejandro Riquelme Cencerrado

Resum— Aquests últims anys hem experimentat grans avenços tecnològics. Alguns d'ells es podrien haver considerat com impossibles d'haver-los pensats fa només uns pocs anys enrere. El nostre projecte està influenciat per l'exitós concepte de l'anomenat *"Smartphone"*, que de fet es pot considerar com una extensió de les actuals funcionalitats dels éssers humans. Partint d'aquest concepte, van nèixer dispositius similars com ara *Smartwatches*, *Smartbands*, etc. Tot apunta al tan habitual terme *"smart" o intel·ligent,* que consisteix en un dispositiu electrònic connectat a una xarxa i dotat amb un cert nivel d'interactivitat i autonomia. La idea d'aquest projecte és doncs, crear unes *Smart Glasses* funcionals, utilitzant les funcionalitats que ens aporta la plataforma de Google Cloud, amb el motiu d'aportar als humans la capacitat d'interactuar amb el món. La interacció principal amb el nostre dispositiu intel·ligent serà mitjançant comandes de veu, que accionaran les següents funcions: una aplicació de mapes GPS, reconeixement de senyals, recordatoris, informació del temps, control de dispositius intel·ligents, detecció de text mitjançant OCR + traducció, etc.

Paraules clau—Smart glasses, Google Cloud Platform, Dialogflow, Raspberry Pi Zero, Pi Camera.

Abstract— This recent years we have experienced great technological advances. Some of them, could have been considered as impossible when thinking on them a few years ago. Our project is influenced by the successful concept of the well-known *"Smartphone"*, which can actually be considered as an extension of the current functionalities of the humans. From this concept, similar potential devices were born such as *Smartwatches*, *Smartbands*, etc. Everything is pointing to the *"smart"* term, which consists in an electronic device connected to a network and with a certain level of interactivity and autonomy. The idea of this project is to create a cheap and functional *Smart Glasses* device by using the Google Cloud Platform capabilities, with the objective to give the humans the capacity to interact with the world. The mainly interaction with our smart device will be done by means of a voice assistant that will trigger several functions like a Maps GPS application, Traffic signals recognition, Reminders, Weather information, Smart devices control, text detection through OCR + translation, etc.

Index Terms— Smart glasses, Google Cloud Platform, Dialogflow, Raspberry Pi Zero, Pi Camera.

1 Introduction

THERE is no doubt that we are living through one of the most exciting stages in the history of humanity since the arrival of the information age. We are witnessing how technology transforms the world around us, how we live and work.

The human's current trend is to connect everything to the cloud in order to enjoy the high availability that offers us as well as having the possibility of making interactions between our devices.

This fact is bringing us the capability of interact with the world, giving us therefore the capacity to extract interesting and precise data that on the past we didn't take into account.

At this point, this is where the Smart Glasses would play an important role in our society by approaching us the ability to experience an extra dimension in our lives.

2 State of the Art

After Google presented the Google Glasses in 2013, the market of the Smart Glasses grew up with a lot of competitors such as Sony with its SmartEyeGlass, ODG R7 AR or Vuzix Blade offering a great number of functionalities but with a deorbited price. Another type of Smart Glasses are the Snap Spectacles 2.0, which offers limited functions like the possibility of taking photos or videos along with a modern design.

The future of the Smart Glasses is not stated yet, there are now a wide variety of available options, but no one offers something with both, low price and several functionalities. The reason of this project is therefore to give to the users a cheap and equilibrated device.

- *E-mail de contacte: alejandro.riquelme@e-campus.uab.cat*
- *Menció realitzada: Enginyeria de Computació.*
- *Treball tutoritzat per: Felipe Lumbreras*
- *Curs 2018/19*

3 REQUIREMENTS

The requirements for this project will be divided in two main parts, for one part the hardware and on the other hand the software, both equally important.

3.1 Hardware

In order to accomplish all the functions we want to implement in this project, we will need the following components.

Main electronic devices

- ***Raspberry Pi Zero W [4]:*** the Raspberry Pi will be the performer of all the actions. We are going to use the *Zero W* version due to its little size. It also implements WIFI and Bluetooth modules so that we can connect it to the mobile phone and the headphones. This device has enough power action to manage all the needed actions with the minimum delay.

- ***Pi Zero Camera 2.5MP OV5647 Sensor [5]:*** the *Pi Zero Camera* offers a proper quality of image so that we can apply some computer vision algorithms and extract useful information.

- ***Screen WaveShare RGB OLED 0.95" [6]:*** this screen will display useful information for the user.

- ***Sonoff [7]:*** it's a smart switch that offers the possibility to power off or on some devices via WiFi. This will be triggered with IFTTT [16] in our application.

- ***Broadlink RM [8]:*** it's a smart infrared trigger that offers the possibility to do some actions like power on the television, change the temperature of the air conditioner, etc. all of this via WiFi. This device will be triggered through IFTTT in our application.

Accessories

- ***Headphones with Bluetooth [9]:*** these headphones will be used to provide additional information to the display.

- **Case printed with a 3D printer:** the Smart Glasses will be mounted into a printed case that will fit the exact measures to not to be extra-large or extra-small.

- ***PowerBank:*** *the Raspberry Pi Zero W* need to be powered by an external power supply. In this case, we will use a PowerBank of 15,000 mAh to have more than 2 days of usability.

Connecting elements

- ***Micro USB Cable:*** the micro USB cable will be used to connect the Rapsberry Pi Zero W to the PowerBank.

- ***GPIO Cables:*** the GPIO cables are used to connect the display to the Raspberry Pi Zero W.

3.2 Software

In order to accomplish all the functions we want to implement in this project, we will need the following components.

The project will be developed using some Google technologies, most of them are included in *Google Cloud Platform [10]*. It will be the brain of this project, due to its enormous power and its compatibility with a huge amount of technologies. Also it offers us a huge variety of tools which are going to be very useful for our project:

One of the tools we are going to use is *BigQuery Machine Learning [11],* which is a RESTful web service that allows massive interactive analysis of large data sets in conjunction with *Google Cloud Storage [15]*. In our project, *BigQuery* will be used as a complement of the Machine Learning API to create models with an enormous amount of data that can be processed within seconds.

Furthermore, in order to implement traffic signals recognition, we will use *Google Vision API [12]* since that Google Vision API offers already trained models. Those artificial intelligence functions will use *OpenCV [13]* opensource library, developed by Intel. It also has been used for multiple projects of autonomous vehicles, security systems, and so on. We will use *Google Vision API* along with *Google Maps API [14]* so that we can ask for directions, translate places from string to coordinates and also to customize maps.

Several extra functions will be implemented such as the *Wolfram Alpha API* [28] will be implemented in order to ask for mathematics calculations and general knowledge-based questions. Finally, the *Weather API [29]* will help us to obtain some weather information according to our location or any other desired city.

All these functions will be interpreted and triggered by a Voice assistant or by *IFTTT* (If This Then That), for instance in our project, the action will be a HTTP request, then will activate some actions, this will be used to control smart devices. In order to perform this function *Google Assistant [17]* will trigger *DialogFlow [18],* which will create chatbots using natural language conversations through the implementation of machine learning.

Therefore, the functions of the Smart Glasses will be triggered by the human voice and will call the functions we will implement on Google Cloud Platform with the Raspberry Pi.

Python [19] will be the main programming language due to the high availability of artificial intelligence libraries. Moreover, the functions that will be triggered by the Google Assistant, will be executed by means of *Cloud Functions [20]*. *Cloud Functions* are integrated as a Webhooks of *DialogFlow*, this integration allow us to do programming in node.js and do some actions in *Google Cloud Platform* when a determined action needs to be triggered by the chatbot.

For this purpose, Google *Speech to Text API [21]* as well as *Text to Speech API [22]* will be fundamental in order to allow us to interpret the human voice as well as respectively translate it into text in order to communicate to the customer via audio. Also, *Translation API [23]* will be used in order to help the people to understand other languages. In our application it will be used with the OCR function in order to bring to the user the requested translation of every text.

At the end, all this processes are data flows which need to be processed and analyzed. In this project, *DataStore [24]* will be used to store information about functions such as reminders (shopping lists, interesting logs, etc.).

4 TASKS

In order to develop this project and reach our goal, it will be necessary to go through several steps:

1. *Research several information:*

   This will be widely explained on the section 4.1, however, it can be summarized into gathering information about similar projects, technologies to implement, *Google Cloud Platform*, etc.

2. *Buy the components:*
   The components will be acquired in some of the most well-known and cheap web pages with Chinnese origin, such as Aliexpress, Bangood or GearBest.

3. *Setting up our work station:*

   This process includes installing the Raspbian SO in our *Raspberry Pi Zero W*, interconnecting and testing our different modules (*Pi Zero camera*, *WaveShare screen*, Powerbank, etc.).

4. *Modular development:*

   Our application will be modularly designed:

   4.1. *Google Assistant*, *DialogFlow and IFTTT* implementation and configuration.

   4.2. *Google Maps API* implementation.

   4.3. Traffic signals recognition.

   4.4. Additional features (Reminders, OCR + translation, Smart devices control, Wolfram Alpha API, Weather API, etc.).

5. *Joining all the modules into a unique application:*

   All the described modules will be implemented into a single application. When the user executes a command by voice to the *Google Assistant*, the chatbot is going to call the corresponding function.

6. *Testing and debugging our application:*

   This is a fundamental process that is present in every industry before a concrete product is launched.

4.1 Research

The research in this project will be focused mainly in seven tasks.

1. *Are there similar projects?*

The first step before planning the project was to do some investigation about similar projects. In my case, the most significative one has been the Google Glasses. Another examples are the Sony Glasses or the AsGe Glasses. However, all those options are really expensive, which is far away to what we focus on this project: to develop a cheap and powerful Smart Glasses.

2. *Which technologies should I use?*

The first plan was to use an on-premises server, such as my personal computer to perform all the computation tasks, but this option was discarded since that this project could have some potential, not only as a student project, but as a commercial project in a more advanced and future stage.

Cloud computing is a powerful and new technology, so, I wanted my project to be available globally and I wanted to be charged only for what I am using. *Google Cloud Platform* was the chosen platform to host my project due to all the machine learning possibilities that offers.

2.1. *Learn how to use Google Cloud:* This is maybe the most wide and complicated part of the project. The use of *Google Cloud* as a platform and understand how everything works together is a task that I'm still not dominating at all. There are many options and products to work with, for that reason is amazing and at the same time easy to get lost. However, it's amazing the wide range of possibilities that offers thanks to all the functionalities and pre-trained machine learning that *Google Cloud Platform* has.

2.2. *Do a research of which APIs can I use to develop this project*: Same as the previous point, there are a lot of APIs in Google Cloud and to know what is the best to use in your project is a hard task to accomplish.

3. *Search the components:*

The adequate components to develop this project have to be cheap, small and somehow powerful enough. The research was tough and many options were considered.

The optimal option for this project was a *Raspberry Pi Zero W* due to its high computation power and easy management compared to other devices such as an ATTiny (with Bluetooth and WIFI modules also), which was discarded due to low computation power.

Another problem in the components research was to found a display to show all the information, but without hindering the vision of the user. For that reason, an OLED screen was the chosen screen type, since that it has self-illumined LEDs and can be used as a projector by using a mirror and a glass to display the image.

4.2 Hardware

Several steps will have to be taken in order to get our hardware ready for this project:

1. *Install Raspbian OS in the Raspberry Pi Zero W:* the Raspbian OS is the compatible OS for the Raspberry Pi Zero W and offers all I need to develop this project.

2. *Install the drivers to use the Pi Camera:* the Pi Camera Zero to work with the *Raspberry Pi Zero W* needs some drivers.

3. *Install the drivers to use the display:* the *WaveShare display* to work with the Raspberry Pi Zero W needs some drivers.

4. *Test the components:* in order to test whether the *Pi Camera Zero* and the *WaveShare display* works in the *Raspberry Pi Zero W* after installing the drivers, we need to do some programming, in this case, *Python.*

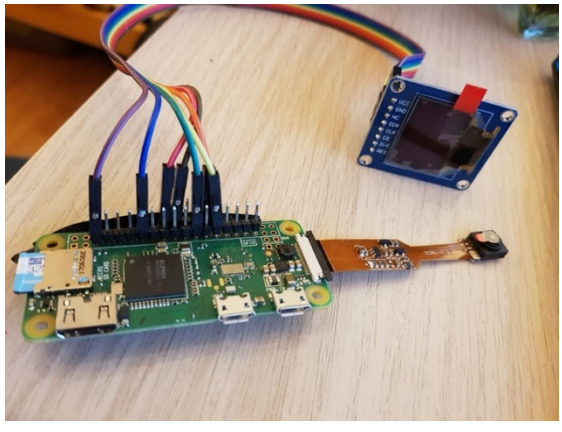


Figure 1: Raspberry Pi Zero W with display and camera

4.3 Functionality

Our project will implement several functions or modules in order to perform interesting utilities:

First of all, it will be needed to establish a link between our Smart Glasses and the Cloud Server so that the Chatbot call and activate some functions in both sides, in our Cloud platform and also in our local *Raspberry Pi Zero W*.

As we stated on the Software section, Google Assistant will trigger *DialogFlow*, so a chatbot will be created in order to interact with our Smart Glasses, we will be able to use our voice in order to trigger some functions of our device. Recognizing the human voice and interpreting it, will be vital for our project, so a *Speech to Text function* will be implemented in order to achieve our goal.

We will be able to ask him questions like “What’s the weather in Barcelona today?” and so on. The objective is to make the user interact with the voice, without touching any screen. After introducing that command by voice, our device will give us an answer reproduced on our headphones.

Several smart functions will be developed:

- ***Create route and show directions through Google Maps API:*** the *Google Maps API* will help us to trace routes and show directions in our Smart Glasses, it will show us in the display when we have to turn the left or the right and some other information like the time we will need to arrive to our destination. This application would prevent people from having to constantly look to their Smartphones. Also, the fact that this application is triggered and controlled by voice, which makes it very friendly with a wider range of users.

- ***Neural network to recognize traffic signals****:* it will allows us to recognize some traffic signals, like Stops, velocity reductions, etc. Those functionalities would be specially interesting to avoid and prevent traffic disasters.

Also, a bunch of additional and interesting features will be created:

- ***Reminders application:*** this feature will save our daily reminders. It has several utilities, for instance, creating shopping lists in order to not forget anything when we do our shopping in the supermarket.

- ***OCR + translation:*** this application will take a picture from a text, like for instance a newspaper or similar, and it will translate it into the desired language. The first step will be to call the *Google Vision API* in order to extract the text from the Smart Glasses Camera streaming and generate an internal text file such as a JSON. This JSON file will be sent to a Text Translation

function that will interpret text through the integrated camera and will translate it into the desired language. The most interesting part is that we don't need to specify from which language we will want to translate, the application will auto-detect it.

- ***Taking and saving pictures:*** this application will work as a camera application and a gallery. Pictures will be saved in G*oogle Cloud Storage* and displayed on our screen.

- ***Controlling smart home devices:*** an important functionality of this Smart Glasses is that it can be used to control some of our smart devices, such as switches, televisions, air conditioners and so on.

- ***Wolfram Alpha API and WeatherMap API:*** the first application will perform simple calculations as well as solve common knowledge questions whilst the second one, will provide us weather information according to our location.

4.4 Documentation

1. Do the "Dossier"

Create this "Dossier" that will explain all the functionalities of this project and how it has been developed.

2. Do the presentation

Create the final presentation to be used in front of the jury in order to summarize my project and do a technical demonstration.

5 METHODOLOGY

This project will be developed following the *Kanban [25]* methodology, with this technology the project will be much easier and organized without bottlenecks and everything will be well managed since the first day. Also, this is a big project, for that reason I chose the Kanban methodology, because it can scale to be use in a big group and it can be combined with *Scrum [26]*, if in the future the external help as a team is needed, it could be scalable.

6 DEVELOPMENT

In this section we will tackle how the project has been developed and which tools and devices described above has been used in this project. At the very beginning, this project was planned at minor scale by using an on premises server with a lower computational capacity. However, due to the scalability and global availability features, I turned to use *Google Cloud Platform*. Also, the main electronic device that was going to be connected to the server was the ATtiny85. This idea was considered due to its reduced size, which was perfect in order to fit inside the Smart Glasses.

Nonetheless, this option presented some problems such as the connectivity between other electronic components such as the camera, the display and even the Bluetooth module. For this reason, I decided to change my core electronic device to a *Raspberry Pi Zero W*, which offers us a higher computation power as well as connectivity problems are solved and still has an adequate size. However, note that if we wanted to bring this project (in a more mature stage) to real consumers, the processing unit would be much more limited in order to further decrease the size as well as adjusting the needed computational power to the minimal needed one.

Going deep into more details, as we stated before, developing this project in a local server wasn't possible. This was due to the lack of power when developing a neural network. For instance, in the case of creating a neural network in order to detect traffic signals, it would imply a time lapse of weeks or even months of computational processes. Instead, when using *Google Cloud Platform* it's a matter of minutes to do so. Also, in case that this project was commercialized as we talked before, the global availability of our services would be a vital feature.

Google Cloud Platform offers us a great amount of functionalities from *Software as a Service (SaaS)*, such as the Google Vision API and also *to Infrastructure as a Service (IaaS)* like for instance Google Compute Engine. It is really useful to have this in order to save time. The brain orchestrator of this project is the *Dialogflow* service, which connects everything. This is actually a Cloud Function with some specific tools to be used as a chatbot application. However, before understanding how *Dialogflow* works we need to first know what a *Cloud Function* is.

A *Cloud Function* is a service provided by Google that allows us to create microservices to run on the Cloud, it is just, as the name says, a piece of code that contains a determined function. In fact, this function is actually a real cloud application which is running on the top of a *Google Managed Virtual Machine (VM)*. This means that we are not able directly access to the internal code of those VM's. However, this doesn't have any impact on our project, in fact it makes our work easier since that we don't need to worry about building from scratch a Virtual Machine, this is one of the interesting features of *Platform as a Service (Paas)*, we will have our microservice easily running in different zones, therefore providing us a high availability.

Furthermore, *Dialogflow* also offers us some built-in machine learning functionalities to understand natural language utterances by extracting and matching structured data. *Dialogflow* is divided into intents or switches, so depending on the command we introduce by voice to our assistant, it will trigger one function or another. If for instance we say *"I want to go to Plaça Catalunya"*, it will interpret that you want to get directions to go from your position to Plaça Catalunya. This is inter-

preted on this way because it gets the phrase "I want to go to", so it will understand that the user wants to trigger the directions function and the place will be send as a parameter. Furthermore, there are pre-configured parameters in *Dialogflow* such as names or dates, nonetheless, in this case it is needed to set it up previously.

Another important feature of *Dialogflow* is that it can learn from training phrases that you provide as well as some language models already built-in. That is to say that the model that *Dialogflow* creates within this application is unique to this project. As described above, this application uses some additional application programming interfaces to develop this project as a complex and useful project.

6.1 Google Maps API and Traffic signals

One of the first modular functionalities developed in this project was the implementation of the *Google Maps API*. This integration performs two mainly functionalities:

- The first one is that it sends to the *Google Maps API* the name of the place where we want to go by using the geocode function and then returning it into a JSON file with the coordinates of the place in latitude and longitude form.

- The second one consists in getting the GPS position (latitude and longitude) of the user, and sending these parameters to the *Google Maps API* again, but this time using the *getDirections* function along with the GPS position of the place where we wanted to go, returning again a JSON file with a detailed description of how to go to that place.

Transforming the JSON directions to a human readable format, will allow us to show the instructions of how to reach the destination place to the user. This information will be showed via the integrated display on the Smart glasses, for example, showing information turn to the right or the left, and finally, the name of the next street.

Figure 2: Waveshare screen showing destination and distance to our destination

Figure 3: Waveshare screen showing the following indications to reach our destination

Regarding the directions function integrated on the Smart Glasses, we will have an interesting feature to increase the road safety. It consists on the implementation of an automated machine learning system to detect traffic signals in order to display those signals in the screen. This feature was made using the *Google Auto Machine Learning* service since that it give us a great amount of power to train our model. Thanks to the help of the Google models, we are able to obtain a high reliability (around a 99%) for all the existent traffic signals over the Spanish territory.

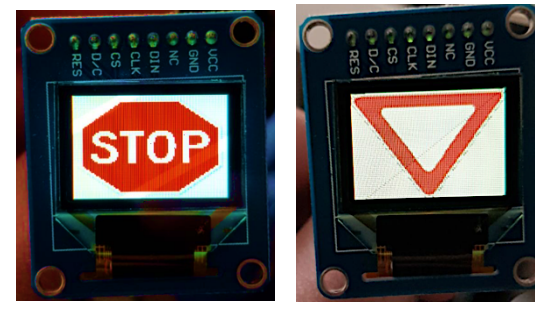


Figure 4: Waveshare screen interpreting and displaying signals

When introducing by voice the command we described before: *"I want to go to Zaragoza"*, it will perform all the steps described above and also, if we choose the driving or bicycle option, it will check every second whether there is a traffic signal. In order to do that, our integrated camera module will take a photo and then, it will be sent to *Google Cloud* to process it.

As stated before, the final result of this processing will be a JSON file with all the information, so when parsing it we will obtain all the possible traffic signals and where they are. An example of that would be to advice the user to reduce the velocity or if it is a stop signal, advise him through both sides.

6.2 Smart devices control

As we live in a connected world, we are always dealing with the "Smart" concept in our daily life. Taking this into account, I decided to implement a function to control our smart home devices with our voice, just in the same way as the Google Assistant does.

However, since that it is a small project (and it is not possible to go directly to the companies that develop this smart things and ask them to integrate our application on their systems), I used the *IFTTT* application (If This Then That), which has a widely variety of integrated smart devices. Getting into more details about the *IFTTT* concept, it is a service that creates chains of simple conditional statements, which means that if one condition happens, it will be triggered and will consequently call another function.

In this case, I created a simple button that works on Android phones so that every time this button is pressed, the following actions will be performed:

1. It will go to the *IFTTT* server and will send a request to the server of my smart device company.

2. This server will send a request to our device and it will be consequently activated.

I inspected the internal code of the button in order to obtain the API call that is used to call to the IFTTT server. The objective of inspecting the code is to find the URL that has access to trigger my smart device.

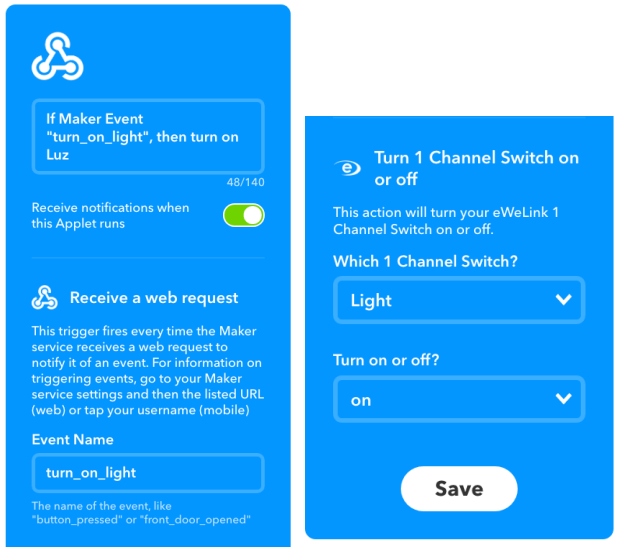


Figure 5. Configuration of an applet in IFTTT

Anyway, this task is not automated, note that it is easy to obtain the API call but it is needed to be extracted by the final user, so that this API can call our system in order to activate the smart device, an example of how an *IFTTT API* call works is:

https://maker.ifttt.com/trigger/turn_on_light/with/key/[KEY]

Once we obtained our URL, we just need to insert it into our Smart Glasses application and set it with the name used in order to be configured in our Smart Glasses assistant. For example, if we set this URL to be called when we say, *"Activate smart turn on the light"* it will search in our *Google Cloud Datastore* database whether exists a *"turn on the light"* entry associated with my user. If that entry exists, it will get the URL showed above and after that, it will make a request to perform the action.

6.3 Reminders

This function will help the user to remind him daily actions or notes that he had previously inserted, such as telling to the assistant that he needs to go to the bakery tomorrow at 3pm.

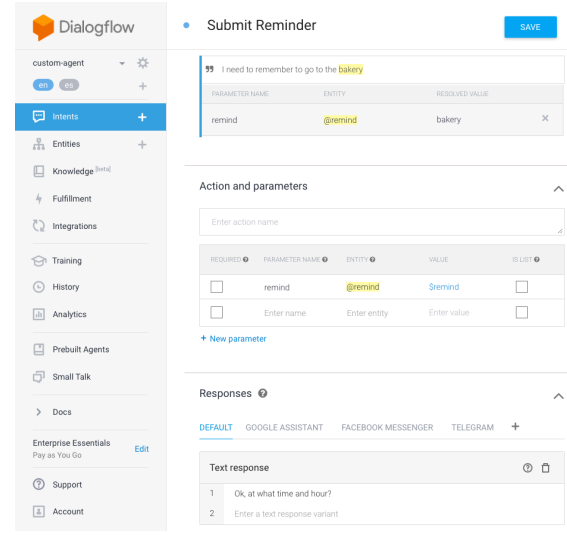


Figure 6. Creation of an intent in Dialogflow

First, we will say to Google Assistant something like "I need to remember to do XXXX" or "Can you remind me to YYYY?". Then, our assistant will ask us: "Ok, at what time and hour?", and we would reply something like "On 21 of December at 12:00h". Finally, this reminder will be inserted to Datastore. In order to extract the inserted reminders we will just have to say something like "Do I have any reminder?".

6.4 Wolfram Alpha API

The Smart Glasses will also use the *Wolfram Alpha API* to search on his database in order to find solutions to different inquiries, so, it will work like a little Wikipedia. For instance, if I ask about "How fast can a cheetah run?", the answer will be:

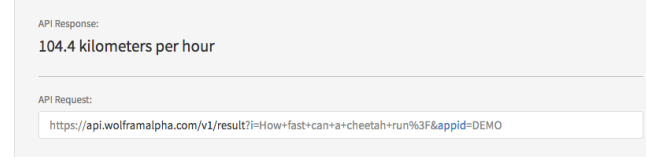


Figure 7. Call to Wolfram Alpha API

This will help the user to find quick and short answers and will improve the knowledge of himself and also, the knowledge of the people around him.

6.5 WeatherMap API

Another implementation made to help the users on their daily basis is the integration of the *WeatherMap API*, that will send us the weather of the place in which we are. This will be performed by using again the GPS position of our mobile phone:

1. The coordinates will be sent to our *Google Cloud Project* and then.

2. *A Google Cloud Function* will call the *WeatherMap API*.

3. *The Google Cloud Function* will return a JSON with all the information about the weather in our current location.

6.6 OCR + translation

This functionality was made using again *Cloud Functions*, but this time we will use some extra features. The first one is *Google Cloud Pub/sub [32]*. This is a fully-managed real-time messaging service that allows to send and receive messages between independent applications. Another one will be *Translation API*, which will be used to translate text from one language to another.

I'll explain this feature with the diagram in the Figure 8. In the first step the raspberry Pi will take a photo using the PiCamera, then, will send the photo using the *Google Cloud Storage API* for Python. This function will upload the photo to a bucket in my Google Cloud Project. After that, the image will be uploaded and automatically the *Cloud Function* will convert the image to text using OCR. This function wrote in Python will call to the *Google Vision API* and will perform the transformation, then it will send this text to a topic hosted in *Pub/sub.*

Once the message is received in *Pub/Sub,* automatically the *Cloud Function* in charge to do the translation will be triggered. As I said in my previous sentence, this *Cloud Function* will translate the text from whatever language to English, so it will automatically detect the original language. To achieve this, we will use the *Translation API* Client library for Python. The following step will be to send this result to another Topic hosted in *Pub/Sub*.

As I stated before, once the message is received in the Topic, the last *Cloud Function* will be triggered. This one will upload the image to another bucket in *Google Cloud Storage*. At the same time, the Raspberry Pi will be continuously requesting information to that bucket in order to show the desired translation throught the Display.

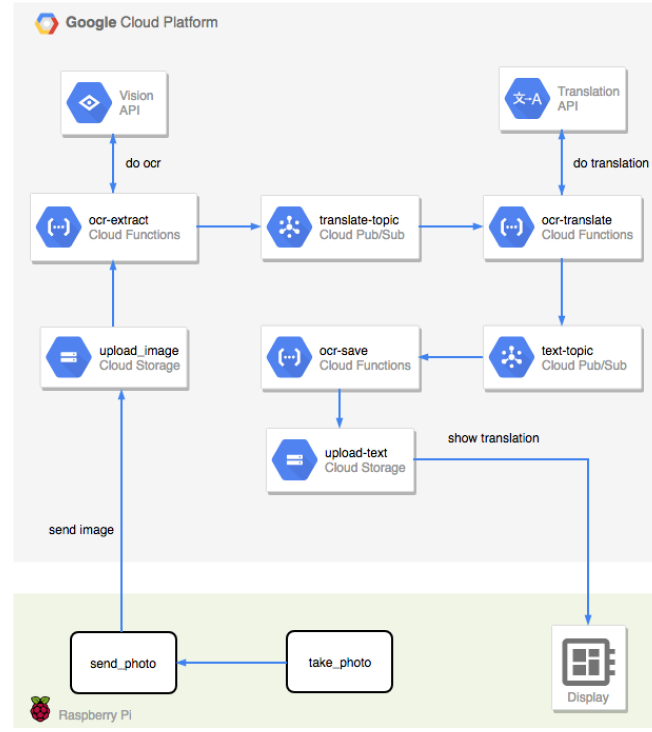


Figure 8. OCR + translation Flowchart

6.7 Taking pictures

I also implemented the feature of taking pictures from our environment and display it on the Smart Glasses screen. First, a picture in JPG format is taken by means of implemented through the *PiCamera* library [30]. Then, using the *PIL [31]* library, it resizes the picture into 96 x 64 pixels and transforms the image from JPG to bitmap so that the screen can correctly interpret and finally display it through a RGB range.

The reason of why I used *PIL* library is because it is possible to resize the image without losing quality and to conserve as best as possible, the original colors.

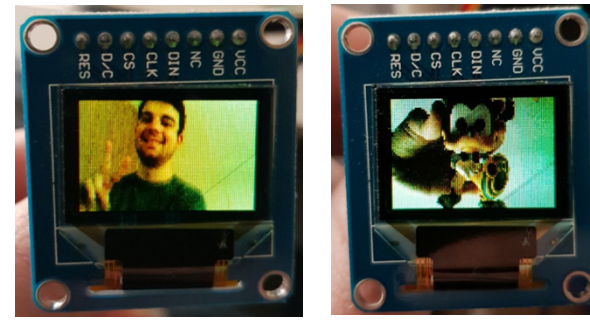


Figure 9. Own taken pictures with PiCamera displayed on Waveshare screen

6.8 Smart Glasses case building

This application has been tested with multiples users, situations and with software developed to try every aspect of the app to break it in different ways. Everything works as expected and with the help of Google models, so now is the moment to build the case that will contain our *Raspberry Pi Zero W*, its *Pi Camera* and the *Waveshare Screen*, which all together will form the Smart Glasses.

For the body of the *Raspberry Pi Zero W* I chose a plastic card box because of its small size and resistance. I attached a metal diadem that also holds a carton box with the *Waveshare Screen* inside, with a mirror and a piece of plastic on the outside, where the image is displayed. The connection cables between the *Raspberry Pi Zero W* and the *Waveshare Screen* are located behind the neck so that they can be more accessible.

Figure 10. Smart Glasses

7 Conclusion

Throughout this project, I've been able to develop a basic Smart Glasses prototype to fullfil a dream to create a device that could help the world in general and the needed people in particular if this concept develops further at a more mature phase.

At the beginning, this project was supposed to be implemented from scratch, without using any cloud computing technology. But I felt overwhelmed by the wingspan of the project, I wanted to implement a lot of things and the vast majority of them wouldn't be possible due to the lack of time. After I started to work as a Technical Solutions Representitaive of Google Cloud Platform, I learnt how to manage Google Cloud and how it could help me to develop this project on time.

As this project has too many implementations, and I used multiple languages and technologies, I found a great number of restrictions and issues, starting with the way of connect an ARM device such as the Raspberry Pi to Google Cloud Platform, to the use of differents languages that I didn't know before, such as Node.js. If I need to highlight one thing about this project, is that I learnt how is to develop an application from scratch being methodical and troubleshooting tons of issues that I faced without decay and always finding a proper solution to every error.

As I stated before, without Google Cloud Platform, this project wouldn't have been possible, developing for example the traffic signal detector it could have taken me almost a month, and it wouldn't be as accurate as this detector. Instead, Google Cloud offered me a lot of computational power to work with without spending so much money and in a short period of time. I truly think that cloud computing is not the future, is actually the present, the natural flow of the computer science and we will need to learn how to squeeze every aspect of it.

To conclude, what I accomplished is just the beginning of this application, this is just a small project compared with what could be in the future. As the base is now set, it's much easier to include more functionalities and applications. I would like to expand this project to truly help other people, not just to be a fancy toy, but to help people with vision issues and other pathologies that will be described on "Future lines" section.

8 Future lines

The concept that is surrounding this project, it's just the beginning of a potential wide range of similar devices with incredible applications. Computer vision is reaching unsuspected limits that can be practically endless when combining them with the Cloud features such as high availability, Platform-as-a-service, Infrastructure-as-a-service, access to huge databases of neural networks, high scalability, application security, etc.

The use of the Cloud infrastructure along with the Computer vision, could make user's life easier and especially, I think it could help people with special difficulties such as: blind people, deaf people, people with mental deficiencies, etc. This Smart glasses concept could help them to be like any other person of the world. For instance, address and maps GPS along with signals report could make up them for the lack of eyesight. They could even read a book with the Smart glasses by implementing OCR and export the result to the speakers.

Also, for deaf people it would be helpful that the glasses could display on the screen every sentence that a person is talking with him. Furthermore, it would be interesting to integrate a translation service that could simultaneously translate a conversation in real time and display or tell by voice the result.

This concept could be also helpful for people with mental anomalies such Alzheimer. For instance, the Smart Glasses could implement facial recognition and tell to the person with that mental disease who is the person in front of him, if there is something pending to tell (reminders), etc. The future of the concept of the Smart Glasses can be very extensive and apply for a diverse group of persons, with lots of different potential applications that could be developed for those devices.

GRATEFULNESS

This Final Degree project is dedicated to:

- My supervisor Felipe Lumbreras for granting me the possibility of devolping this fascinating project, which has allowed me to apply some amount of concepts I've learned on this degree to the newly discovered world of the Cloud.

- All the UAB professors who have been involved in conveying me a spirit of self-surpassing and passion for what I have been doing during these years. It has been a pleasure to learn from them.

- My family: my parents Pilar and Roberto, and my brother Diego for everything they have done for me to make this possible and because they have always supported me.

- Mario, because he has always been by my side when I needed animical support and comprehension to carry out this project.

- The whole Google Cloud team, which currently are my co-workers and allow me to continuously learn about the platform. I really love my job.

ACKNOWLEDGEMENT

Calibri, 9.5 Normal. Acknowledge Universitat Autònoma de Barcelona.

REFERENCES

Conference Papers:

[1] Babak Taraghi and Mahdi Babaei, "Object detection using Google Glass" in 2016 IEEE Conference on Open Systems (ICOS), Bandar Melaka, Malaysia, 2015.

https://ieeexplore.ieee.org/document/7377285

Thesis:

[2] Vahab Pout Roudsari Farnaz, "Smart Glasses Deign, Exploring user perception of wearable computing" Master Thesis, University of Lapland, 2016.

Books:

[3] Valliappa Lakshamanan, Data Science on the Google Cloud Platform, O'reilly, 2016.

Web:

[4] Raspberry Pi Zero W – Product overview.
https://www.raspberrypi.org/products/raspberry-pi-zero-w

[5] Pi Zero Camera 2.5MP Module Support 1080p.
https://www.banggood.com/5MP-Camera-Module-Support-1080p-For-Raspberry-Pi-Zero-V1_3Zero-W-p-1290619.html

[6] Screen WaveShare RGB OLED 0.95".
https://www.waveshare.com/0.95inch-rgb-oled-b.htm

[7] Sonoff – Official website.
https://sonoff.itead.cc/en/

[8] Broadlink RM Pro
https://www.broadlink.com.es/broadlink-rm-pro-domotica-mando-distancia-universal.html

[9] Airpods - Bluetooth Headphones.
https://www.apple.com/shop/product/MMEF2AM/A/airpods

[10] Google Cloud Platform Official Website.
https://cloud.google.com/

[11] Introduction to BigQuery ML | Google Cloud
https://cloud.google.com/bigquery/docs/bigqueryml-intro

[12] Google Vision API – Image Content Analysis
https://cloud.google.com/vision/

[13] OpenCV library
https://opencv.org

[14] Geo-location APIs | Google Cloud
https://cloud.google.com/maps-platform/

[15] Google Cloud Storage
https://cloud.google.com/storage/?hl=es

[16] IFFTTT Official Website
https://ifttt.com/

[17] Google Assistant Official Website
https://assistant.google.com/intl/es_es/

[18] DialogFlow | Google Cloud
https://cloud.google.com/dialogflow-enterprise/

[19] Python Official Website
https://www.python.org/

[20] Cloud Functions | Google Cloud
https://cloud.google.com/functions/docs/

[21] Cloud Speech-to-Text | Google Cloud
https://cloud.google.com/text-to-speech

[22] Cloud Text-to-Speech | Google Cloud
https://cloud.google.com/speech-to-test

[23] Cloud Translation API | Google Cloud
https://cloud.google.com/translation/

[24] Cloud DataStore | Google Cloud
https://cloud.google.com/datastore/docs/concepts/overview

[25] Kanban | Development
https://en.wikipedia.org/wiki/Kanban_(development)

[26] Scrum | Development
https://en.wikipedia.org/wiki/Scrum_(software_development)

[27] Tasks list
https://drive.google.com/open?id=15Use7sDzdSYNndBAgrvf3_MiH7hqE2e_

[28] Wolfram Alpha API Reference documentation
https://products.wolframalpha.com/short-answers-api/documentation/

[29] Weather API Reference documentation
https://openweathermap.org/api

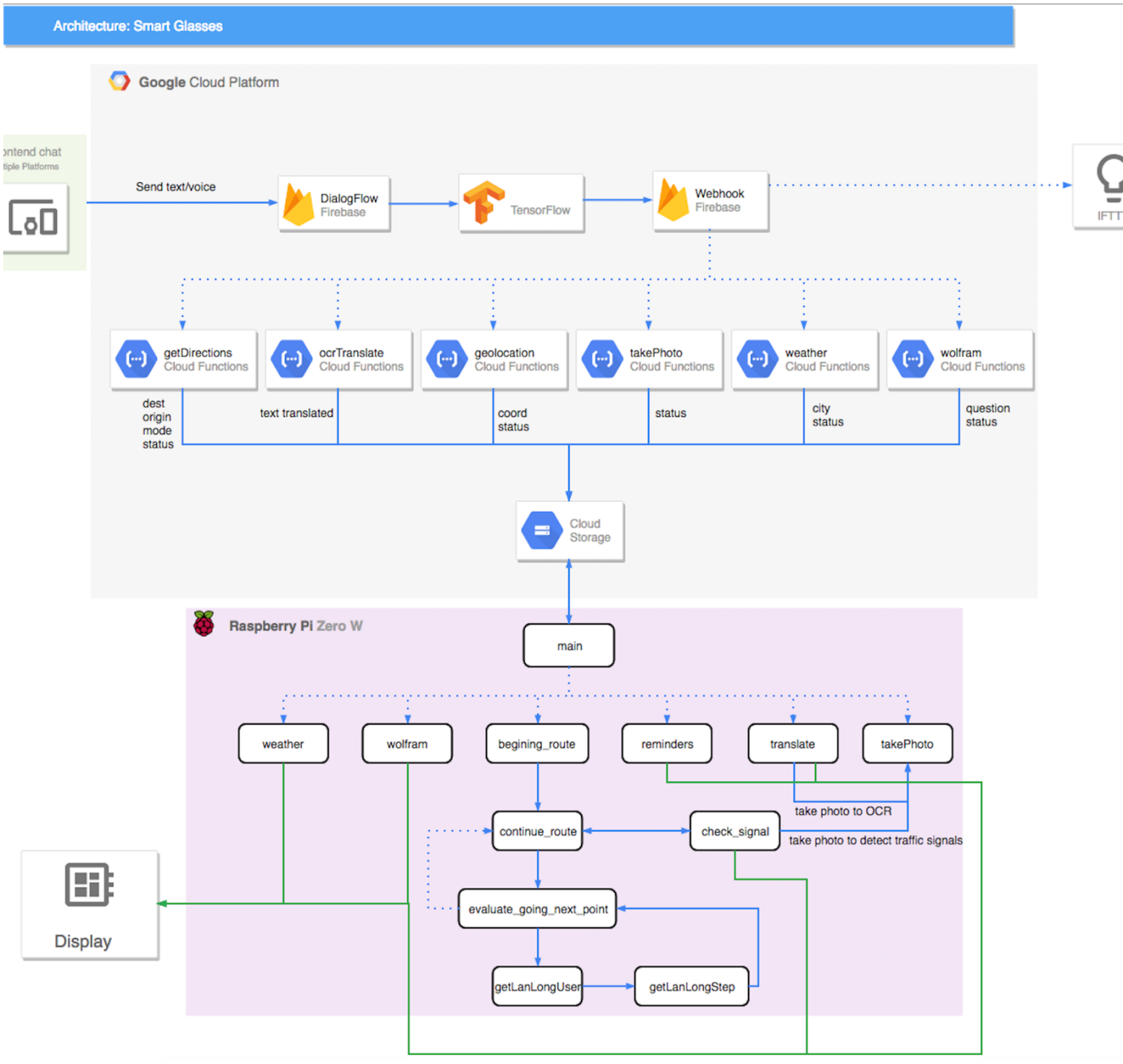
[30] PiCamera Library
https://picamera.readthedocs.io/en/release-1.13/

[31] PIL Library
https://pillow.readthedocs.io/en/stable/

[32] Google Cloud Pub/Sub
https://cloud.google.com/pubsub/docs/overview

APPENDIX

A1. FLOWCHART OF THE WHOLE PROJECT



A2. SMART GLASSES PICTURE GALLERY