Blindsight

Summary

1. Introduction
2. What are the inspirations for this project?
3. Architecture of the system
4. What are the technologies?
5. Description of the Software
6. The Next step
7. Conclusion
8. Source and annexes

# Introduction:

For this final year, we need to create a project, that will “concentrate” all the different aspect of our formation, that’s mean software engineering.

For my project I choose to find a new and innovative way of helping people so I research, how I can help with different handicap at my level, I choose to try to accommodate the life of people with impaired vision.

For doing so I first try to found different way of simulate vision but the user interface was an obstacle for it, so I designed this project based on how nature have done it, indeed the dauphin and bat use the principle of echolocation to help them navigate in the 3D world.

Inspired by this I try to understand the limit of my system and build around them.

So, I will in this report describe how I managed to create this project.

# What are the inspirations for this project?

This project was designed to help people with impaired vision be able to detect some

Higher than ground obstacle. Because even if walking stick and dogs are quite efficient it can happen that they didn’t recognize taller obstacle.

So, for helping them I research how can I help them to accommodate with this issue.

By doing so I read about biomimicry and was inspired by it.

The principle of Biomimicry is how the human design a system based on the imitation of the already existing system present in nature.

In the case of this project I was inspired by the ability that dauphin and bats use to detect their preys.

Indeed, using ultrasound they can know the distance that is between them and their prey. And if we apply this to my project it means that I can know Une image contenant jeu

Description générée automatiquementthe distance between them and an obstacle and inform the user of it.

# Architecture of the System

This project is composed of different parts:

* A vest to hold the different element together
* An Arduino Board for the detection program
* A Raspberry pi 4 for the high-level functions
* 3 Ultrasonic sensors use for detection
* 1 IR distance sensor
* 4 Vibration motor use as an interface
* 1 Microphone for the voice command

Each of this part work together and communicate between each over, to create the main system, for example:

If a Ultrasonic sensor detect a distance below a certain threshold (80 cm) , the Arduino mega will pick it up and then send it to the Raspberry pi then the raspberry will store it, during this time the Arduino will activate the vibration motor corresponding to the sensor.

And on another program the raspberry will be listening to certain voice command and interact with the user if the commands are correct.

The different parts are designed to work in a certain way, for example the Ultrasonic sensor have been chose because of the area of detection, contrary to IR sensor that just have a very precise but it’s far too narrow for the use of the project.

I have chosen to divide how the project work in 3 different programs:

* The first one is the Arduino one, it checks the data of the sensor each loop and send it to another program on the raspberry pi
* The second Program host on the Raspberry pi will receive the number of the sensor that have been activated and stored it in a file.
* The third program will work on the “smart” part of this project using Speech to Text and Speech to Text it will activate different functions when asked for it.

The 3 programs will then work together to create the whole project.

# Arduino & Raspberry

## Arduino

The Arduino system is a software made by the company of the same name, they design for the most part single board computer for the use of embedded, this software is open source that mean that everyone can use and modify it.

In this project I use the Arduino Mega, a board that is different from the “general” use of the Uno Rev3.

Let’s compare both with the use we need for them:

We have 3x HR SO4 ultrasonic sensors, with each of them having 4 pin we need at least 12 digital pins plus the 3 for the IR distance sensor and the 4 vibrator motors we will need a total of 12 + 3 + (4\*2) =23 pins. 22 of them as digital ones and 1 as analogical one.

23 Pins

Arduino Uno Rev3:

* Digital I/O Pins: 14
* Analog Pins: 6
* Total Pins: 20

Arduino Mega Rev3:

* Digital I/O Pins: 54
* Analog Pins: 16
* Total Pins: 16

So, it’s better to use the Arduino Mega because of the number of I/O we will need in this project.

Raspberry Pi

The raspberry pi is a single board Computer designed by the raspberry foundation, designed to the use of teaching and learning computer science, it had developed far beyond expectation and the first model became very popular.

Nowadays they existed a lot more than 1 or 2 models especially with the release in 2019 of the raspberry Pi B (the model we used in this project), this model can have until 4 GB of ram, it’s power supply by USB-C and have 4 different USB port (2 USB port 2 and 2 USB 3).

In this project the Raspberry will help us create the high-level function like Geolocation, send SMS, …

The model we use will be the Raspberry Pi 4 B, 4GB.

# The different code for Arduino and Raspberry

## Arduino

The code for the Arduino is quite simple, we need something that will send a signal if a sensor detects something under a threshold for example:

Object

Threshold

Sensor

In this case we will not send a signal because the object is away from the threshold.

Object

Sensor

The object is now under the distance threshold we can send a signal to the board.

For this we will use a “if” statement as follow:

if (measured distance is inferior to threshold) then

do something ….

This will become in code:

if(distance<30){

string\_to\_send += ‘L’;

Motor(1, ”on”);

}

else {

string\_to\_send += ‘B’;

Motor(1, ”off”);

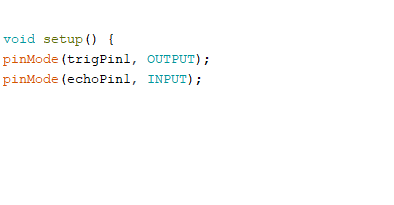
}

The previous code means that if the distance if less than 30 cm we will send add to a string ‘L’ to lesser or ‘B’ to bigger and depending on the condition we will make the Motor 1 vibrate or not.

The program will work like this:

* We define our variable, Input and Output
* We create an empty string
* We enter a loop
* We ask the sensor if something happen and add the answer to the string
* We make the motor vibrate if needed
* We finally send the string to the raspberry Pi

For declaring our Sensor as an Input or as an Output, the variable will be declared as follow:

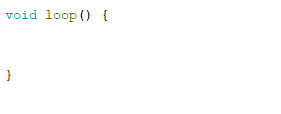


In the setup part we use the pinMode() function and we put in 2 arguments, the first one will be the number of the pin use, the second is INPUT or OUTPUT or INPUT\_PULLUP.

For creating a String it’s: String variable\_name = “…”;

We can also choose to not iniate it: String variable\_name;

The loop will be automatically build by the Arduino IDE and look as follow:

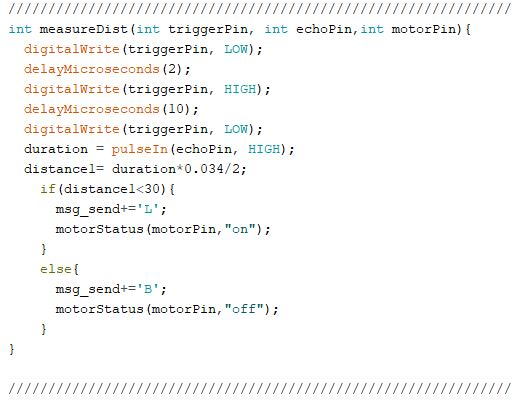


It’s the main part of our program where everything will be.

In this part, first the “Virtual Ground” and “Virtual VCC” will be declared, it’s use is to allow us to have more GND and VCC pins on the board.



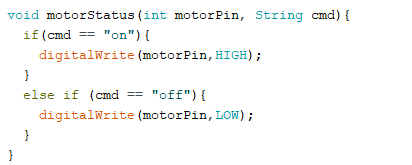
After everything is declared like above, we will go into the main Part of the program, it’s divided in 3 different functions both having different purposes. The first one is made to measure the distance on the Ultrasonic sensor (measureDist) while the second one is for the vibration motor to be activated (motorStatus) the last function is for measuring the distance on the IR sensor (irDist).



This function work as in two parts, the first one measure the distance while the second compare the measured value to the threshold here 30 cm.

If the value is bigger than we add a ‘B’ to the string that we are going to send later, and we make sure the motor is off.

If the value is lower than the threshold than we add ‘L’ to the string, and we put the motor on.



The motorStatus function will take 2 arguments one is the pin Number the other one is the status of the motor ‘on’ or ‘off’ and based on that we activate or not the motor.

The function for the IR sensor will be roughly the same except that we will work in Analog and not digital.

After all of that we will go back into main to send to the Raspberry the string containing the data (a string of ‘L’ or ‘B’).

## Raspberry Pi

For this part we will have 2 different python code, 1 to save the data from the Arduino, the other to manipulate the “High-level” function.

## The first Program

For this program we will use a library called “serial”, this library will help us connect the raspberry to the Arduino and let them talk to each other, it will work as follow:

* Import all the library we need (here only Serial)
* Create some Variables
* Give a path to the file
* Create the link between the 2 boards using serial.Serial()
* Read the value given
* Put them into the file with a timestamp if a ‘L’ appears else discard the data
* Add a number to the value pass in
* Check if the number is bigger than 200
* If it is bigger than delete the file and recreate it
* Else continue the previous work

This part is designed to save some of the value read while if the file become too big and impact the performance of the system dispose of it.