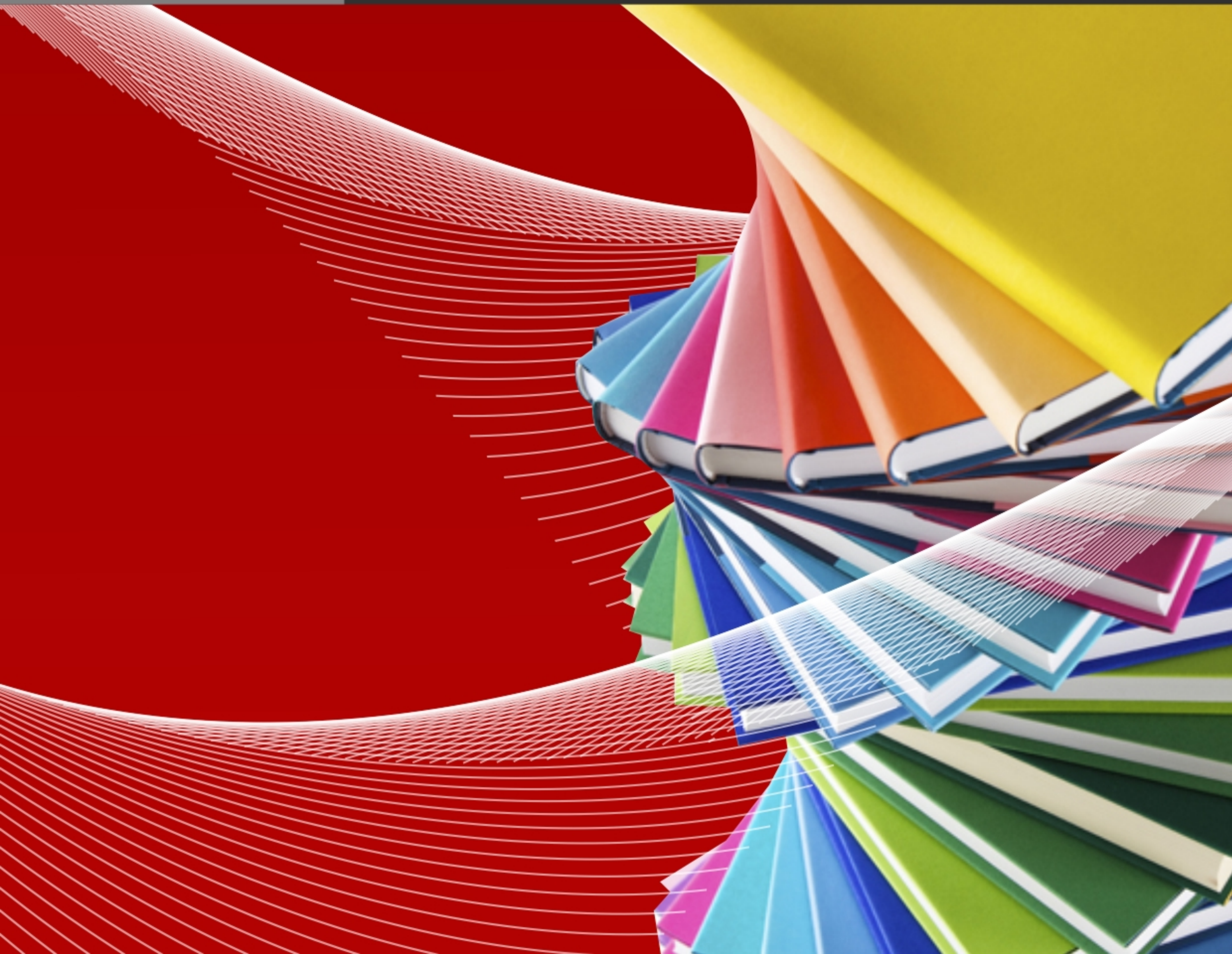
澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

Outstanding Academic Papers by Students
學生優秀作品



Development of an Intelligent Rotational Wayfinding Sign for Smart City

By

KUAN MENG WAI (D-B3-2623-1)

LEI CHON IN (D-B3-2648-5)

Bachelor of Science in Electromechanical Engineering

2016/2017

Faculty of Science and Technology

University of Macau

Development of an Intelligent Rotational Wayfinding Sign for Smart City

By

KUAN MENG WAI (D-B3-2623-1)

LEI CHON IN (D-B3-2648-5)

Final Year Project Report submitted in partial fulfillment of the requirements for the degree of

BSc. in Electromechanical Engineering

Faculty of Science and Technology

University of Macau

2016/2017

University of Macau

Abstract

Development of an Intelligent Rotational Wayfinding Sign for Smart City

By

KUAN MENG WAI	(D-B3-2623-1)
LEI CHON IN	(D-B3-2648-5)

Project Supervisor

Prof. WONG PAK KIN

For travelers who choose walking as traveling method, finding a destination becomes one of the problems that needs to be addressed in the tour. In this industrial project, the design prototype utilizes the raspberry pi and its extension boards and computer program to a smart road sign. Specifically, the program instructions in the electronic display can send the pulse signal to the control board to turn the road sign in a correct direction and to display the destination and distance by the LED matrix to resolve the problem of the orientation and navigation.

ACKNOWLEDGEMENTS

The authors would like to thank to our supervisor, Professor P.K. Wong, for his professional guidance and variable comments. Furthermore, we would like to thank the automotive laboratory technician, Dr. Wong Hang Cheong to give the recommendation and instruments.

Specially the authors would like to thank Brandon Leong and Jump Creative Media, providing the cooperation with University of Macau and the Department of Electromechanical Engineering to study and develop the wayfinding sign system and support us the project materials.

This project has applied a FDCT grant. The grant is currently reviewing by FDCT.

TABLE OF CONTENT

LIST OF FIGURES

NUMBER *PAGE*

LIST OF TABLES

NUMBER *PAGE*

LIST OF ABBREVIATIONS

RPi 3	Raspberry Pi 3
GPIO	General Purpose Input / Output
PWM	Pulse Width Modulation
IDC	Insulation Displacement Connector
VCC	Volt Current Condenser
GND	Ground

CHAPTER 1: INTRODUCTION AND OBJECTIVES

1.1 BACKGROUND

Macau as an international tourist city, based on institutional innovation, led to the rise of various related areas of innovation. The development of innovative tourism products and the construction of an intelligent city have drawn the attention of the public. Among the tourism innovation areas, cultural tourism and community tourism are regarded as the focus of the deep development of tourism.

The Macau Government Tourism Office launched the "Step Out, Experience Macau's Communities"; walking the route in 2013. The main purpose of the route run through various attractions, restaurants, shopping sites, souvenir gift shops, bus routes and arts and ceremonies. The government sets up the tourism billboards (show in Fig. 1-1) and classical direction signs (show in Fig. 1-2) in the urban areas. Although the billboards can guide the orientation and the path, but fails to effectively guiding the navigation.

Figure 1-1 Tourism Billboard

Figure 1-2 Classical direction sign

For those tourists who travel independently, lacking with the tour guide, the orientation and the navigation become more difficult. In the city planning, the tourism signage system is used to solve this problem. The signage system is a navigation system consisting of signs, maps, images, symbols, and various languages that can assist visitors in doing the activities in different places.

Nowadays, the tourists who get the Individual visit and use the "Step out, experience Macau’s Communities – Walking tour routes" (i.e. App) [1] to travel around Macau. This system is limited to regional direction and location. There are three major drawbacks of the App, "Step out, experience Macau’s Communities – Walking tour routes". Firstly, a smart phone with internet connection is a must. Macau Government has increased the number of free Wi-Fi hotspot from 34 in 2010 to 164 in 2015, but there are only about 1.5 million successful connection during the four years. The successful connection rate is very low as compared with the number of tourist that is about 30 million every year. Secondly, the App must be downloaded to the smart phone in order to utilize the digital map and destination information. However, the time for downloading the App and the time for adapting the operation of the App is relatively long so the interest of using the App is not high to the tourists. Lastly, the digital map in the App does provide no orientation or navigation information. It is difficult for the tourists to build up the imagery map for navigation due to lacking of location-based information.

The design of the rotational wayfinding sign which combines the user panel, the Kiosk (Fig. 1-3) and the information system. The rotational wayfinding sign shows the direction, location, distance and destination information to travelers in a timely manner with a user-friendly control panel.

The essential part of this project is wayfinding, the principle of wayfinding is to provide enough information to increase navigablility. Navigability means that the navigator can successfully move from the navigator's present location to a destination [2], so that wayfinding sign is an effective to give the information to the user.

Wayfinding is an important infrastructure for a tourism city to ensure the pleasant journey to every tourist. It enables the tourists to know where to go and how to get there. Besides, it guides the tourists to travel along the intended route and direction without any accident or delay [3]. However, the traditional wayfinding sign has a limitation of the number of destination on the sign is limited, so it can just only provide limited information to the users. Moreover, the destination information updates from time to time which is unlikely to be provided by the traditional wayfinding sign. A smart rotational wayfinding sign is therefore proposed to overcome the aforesaid drawback. Smart rotational wayfinding sign can update the information shown on the sign in response to the user requirement and provides the latest destination information to the user.

The design of the rotational wayfinding sign which combines the user panel, the Kiosk (Fig. 1-3) and the information system. The rotational wayfinding sign shows the direction, location, distance and destination information to travelers in a timely manner with a user-friendly control panel.

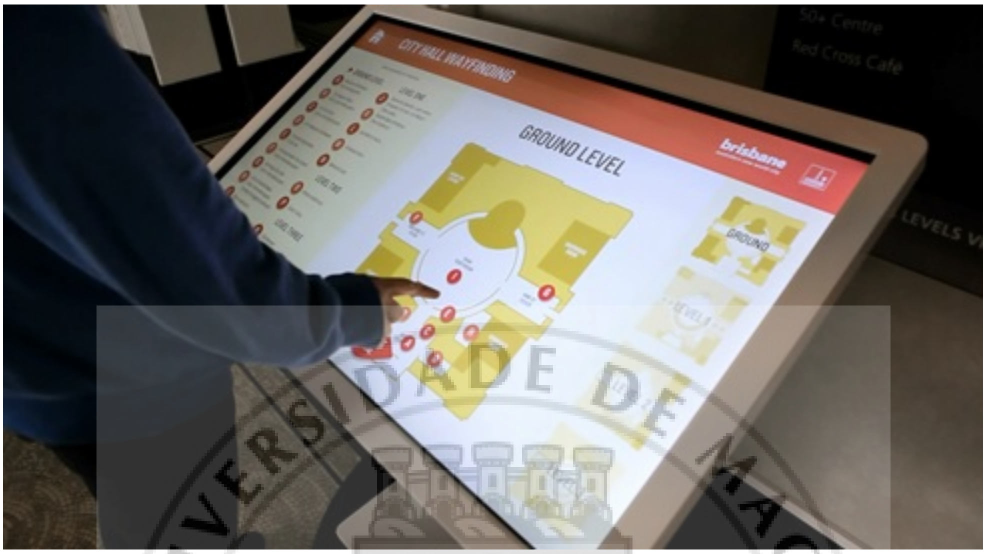


Figure 1-3 Kiosk

The classical direction sign and the smart rotational wayfinding are compared in the table below:

Table 1-1 Comparison between the classical direction sign and the smart rotational wayfinding

	Classical direction sign	Fingerpost 2.0
Information characterisitc	Statics	Dynamic
Information provided	Nearby attractions	Nearby locations, restaurants, shopping points, and more extension locations
Coverage area	Around 50 to 100 meters	Peripheral, regional and city levels
Navigation formation	Destination information shown on the direction	Interactive orientation to establish staged imagery map
Identification network	Fixed, regional changes involve multiple sign modification	Backstage software network
Signage structure	Fixed direction	Mechanical signage with modular design
Display	Flat printing	Electronic display
Management	Manual inspection handling and installation	Remote monitoring and management with natural language processing assistance
Application	Urban destination orientation	Urban destination orientation, traffic network information orientation, large-scale activities flow orientation, integrated community and building orientation

This project is a joint industrial project cooperating with Jump Creative Media which creates a new idea of solving the aforesaid problem. This project aims to design an innovative smart rotational wayfinding sign, which consists of an interactive user panel and rotational pointing arm and a software control system which could transfers the real-time event data via 3G and Wi-Fi from the connected cloud system. Furthermore, this project involves in mechatronics system design.

1.2 OBJECTIVES

The objectives of this study are summarized below:

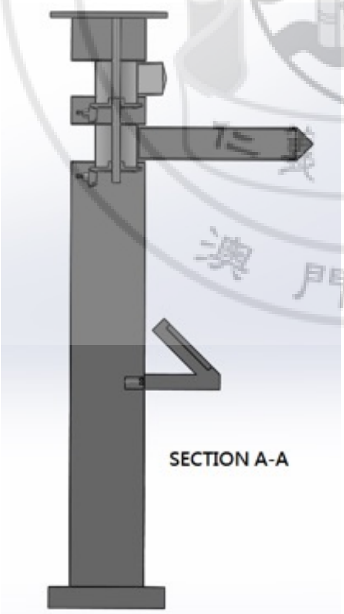
i. Develop the hardware design of a prototype Rotational Wayfinding Sign system.

ii. Develop the software of a prototype for the Rotational Wayfinding Sign system.

iii. Implement and test of the prototype Intelligent Rotational Wayfinding Sign.

CHAPTER 2: SYSTEM DESIGN

Figure 2-1 3D drawing of the prototype



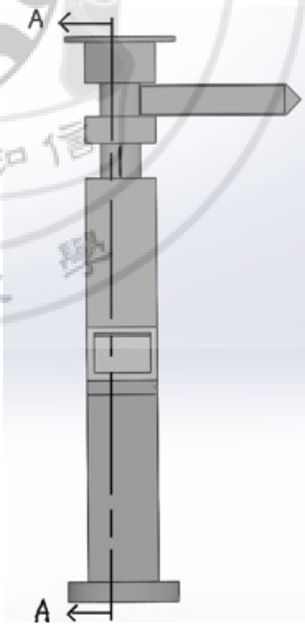


Figure 2-2 Side view of the prototype

Figure 2-3 Front view of the prototype

Figures 2-1, 2-2, & 2-3 show that the drawing of proposed wayfinding sign.

The wayfinding rotational sign is an automation system. The sign has a rotating structure, and electronic display to show the destination and dynamic information. Moreover, this is a controller and user interface in the system.

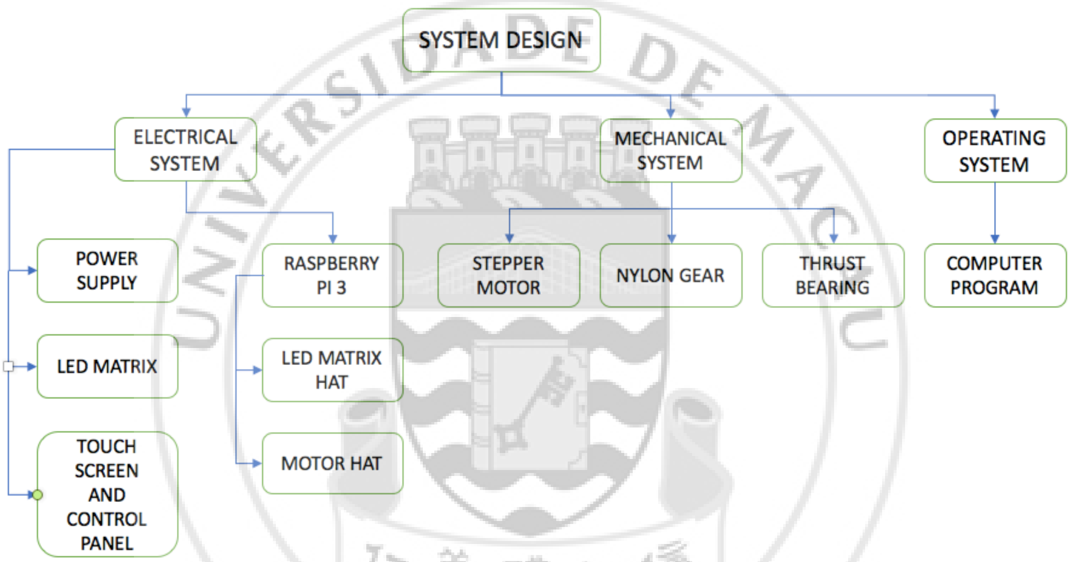


Figure 2-4 System hierarchy

Fig 2-4 illustrates the system hierarchy of the prototype, which mainly consists of the electrical, mechanical systems and software. The electrical system has three basic parts and mechanical system has the stepper motor, thrust bearing and nylon gears.

2.1 ELECTRICAL AND CONTROL SYSTEM

The electrical and control system consists of:

1. Raspberry Pi 3 Model B
2. Adafruit RGB Matrix + Real Time Clock Hat
3. Adafruit DC and Stepper Motor Hat
4. LED matrix display
5. Touch screen panel

2.1.1 Raspberry Pi 3 Model B

The control board of this prototype uses Raspberry Pi 3 Model B (Fig. 2-5), which is an open source software project platform, and includes a simple I/O function of the circuit board and the operating system is Linux system so that numerous Linux software can be run on Raspberry Pi 3 Model B. RPi 3 Model B can read numerous switches and sensor signals, also it can control the lights, LED matrix, motors, actuators and other various physical devices. Furthermore, RPi 3 Model B is similar to the PC device which can be connected to a screen and keyboard to become a microcomputer to implement many software functions such as text graphics, signal processing, game, program compilation and other functions.

Figure 2-5 Raspberry Pi 3 Model B

Raspberry Pi 3 Model B has several major functions and its general functions of RPi 3 are shown in Fig. 2-6.

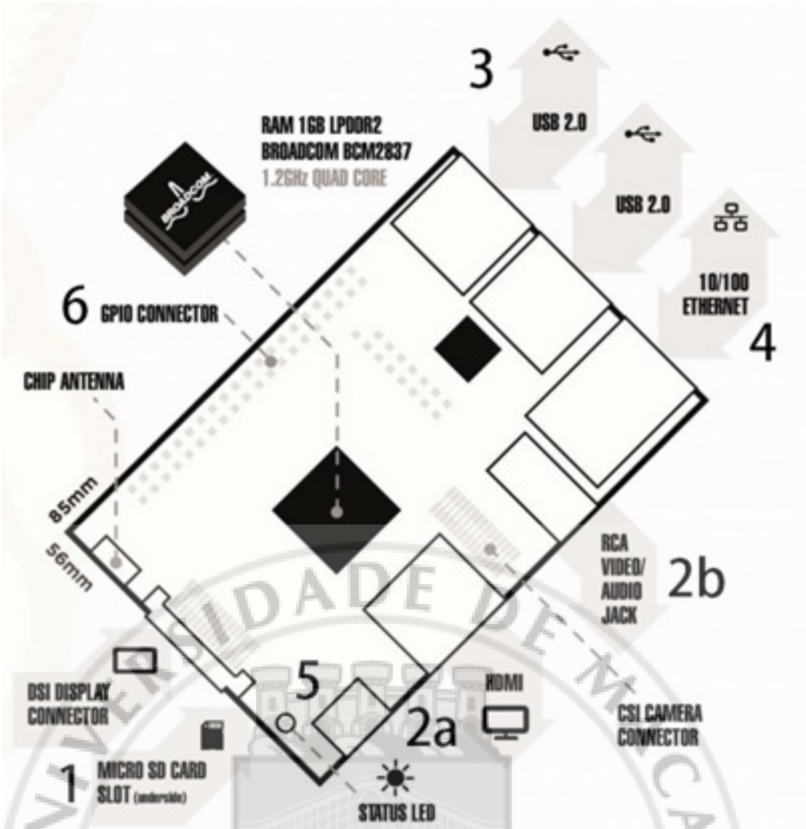


Figure 2-6 General functions of Raspberry Pi 3 Model B

Fig. 2-6 shows that the first part (1) of RPi 3 Model B is an SD Card slot which is used as a built-in hard disk. In general, the capacity of SD card requires at least 4GB. The card reader is located at the back of the motherboard, and the user needs to install the operating system on the memory card. The second parts (2a and 2b) are the display output of HDMI, RCA video and the audio jack sound output which are used to connect the monitor and the speaker. The third part (3) is four connections of USB input which can connect the mouse and keyboard, and the fourth part (4) is the Ethernet input. The fifth part (5) is the micro USB power supply, and the minimum supply voltage and current are 5V voltage and 700mA respectively. The sixth part (6) is the GPIO connector. GPIO in the most basic application of the microcontroller, all the external interfaces are controlled through the GPIO. The full name of GPIO is called General Purpose Input / Output. GPIO goes through the digital I/O pin to perform the signal input or output.

2.1.2 Adafruit RGB Matrix + Real Time Clock HAT

Adafruit RGB Matrix Plus Real Time Clock Hat is an extension part for Raspberry Pi 3 Model B. This apparatus needs to plug into the GPIO connector on the RPi 3 Model B. The main function of this apparatus is to display information in a RGB matrix LED display by using the data cable connected to the 2x8 header on the HAT in order to show the colorful display on the matrix. On the HAT, there is a terminal block that can power the matrix through the HAT.

Figure 2-7 Adafruit RGB Matrix + Real Time Clock Hat

On the HAT (Fig. 2-7), there is a time model called DS1307 Real Time Clock. To use this time model, a 3V lithium battery is required. This model can trace back the time for the RPi 3 Model B even there powered off. The supply voltage is with a supply current between 2A and 4A.

In this project, two 16x32 RGB Matrixes (Fig. 2-8) are required, each matrix panel require 5V power. For one 16x32 matrix, it has 64 pixels. If the display of the matrix shows on full white, each pixel can draw up to 0.06 A. So, the total maximum current per matrix is $64 \times 0.06 = 3.84$ A. There are two matrixes on the prototype, the total maximum operating is 7.68 A.

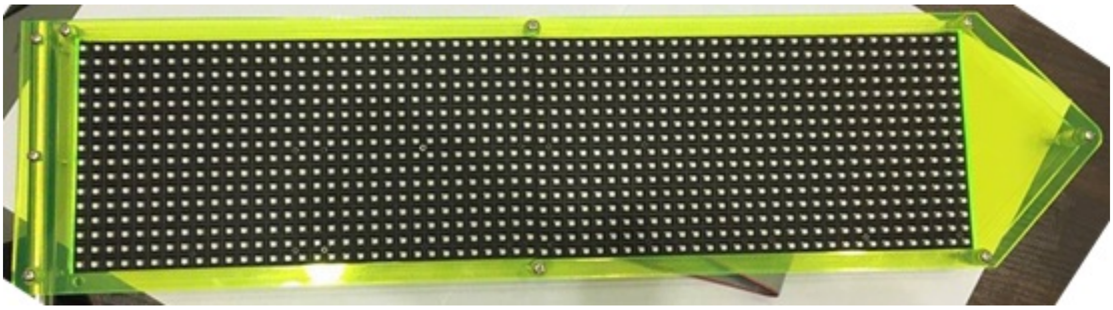


Figure 2-8 Two 16x32 RGB Matrixes

2.1.3 Adafruit DC and Stepper Motor HAT

Adafruit DC and Stepper Motor HAT is another extension part for Raspberry Pi 3. This add-on hardware is suitable for any motion project as it can drive up 4 DC motors or 2 Stepper Motors with the pulse width modulation speed control.

Pulse Width Modulation (PWM) is a digital way to implement the analog output without noise, PWM signal contains two main characters including the duty cycle and the frequency. The duty cycle describes the ratio of the logic time in the signal to the duration of a signal cycle. The frequency is that the frequency of the cycle. By a certain rate of digital signal with a specific period of work, the output power will be equivalent to a specific analog voltage signal.

Figure 2-9 Adafruit DC and Stepper Motor HAT

Fig. 2-9 shows that there are three 2-pin terminal blocks and two 3-pin terminal blocks, one of the 2-pin terminal blocks is used to connect the electric power, the other 2-pin terminal block and the two 3-pin terminal blocks are used to connect the DC or stepper motor and the ground, and the function of the HAT is to drive the stepper motor to rotate the sign.

This HAT cannot connect the DC power supply directly as it does not have the DC jack. So, a female DC power adapter (Fig. 2-10) is employed.

Figure 2-10 Female DC Power adapter –with 2.1mm jack to terminal block

There are some properties on the motor HAT. Firstly, the voltage requirement of the motor controllers on the HAT is designed between 5V and 12V. Secondly, the motor driver chips that are designed to supply a current of 1.2A per motor on the motor Hat, and the peak current is 3A. If the current is up to 2A, the motor driver need to put a heat-sink, otherwise, the chip will have the thermal damage.

2.1.4 Power supply design

Each control board needs to connect with different power supply which is shown in table below, i.e., For RPi 3, due to the network function of Model B, which consumes a larger current supply. Then the power supply needs 5V DC and 1.2 to 2.5A. For Matrix HAT, the power supply needs 5V DC and 2 to 8A. For Motor HAT, the power supply needs 12V DC and 2 to 4A.

2.1.5 Touch screen panel

The intelligent wayfinding rotational sign needs a touch screen panel to give a signal to the RPi 3 in order to give the impulse to the matrix and motor HAT. This touch screen panel (Figure 2-11) connects to the RPi 3. The touch screen panel which uses the 5 inch HDMI LCD, the pixel of it is 800x480.

Figure 2-11 Touch screen monitor

2.2 MECHANICAL SYSTEM

The mechanical parts of the prototype include the following components:

1. Stepper Motor
2. Nylon Straight Bevel Gears
3. Thrust Bearing

2.2.1 Stepper Motor

Stepper motors are great for the (semi-)precise control, it is suitable for many robots and CNC projects. The Motor HAT supports up to 2 stepper motors. The project uses 42BYGH60 stepper motor (Fig.2-12), and the specification of the stepper motor is shown in Table 2-1:

Table 2-1 Specification of 42BYGH34 stepper motor

High	34mm
Voltage	12V
Load Current	0.4A
Torque	0.3Nm
Resistance	24 Ω
Inductance	15 mH
Lead wires	4 PINS
Weight	0.22 kg

When a pulse signal is entered to the stepper motor, it rotates one step angle of 1.8^{o}.

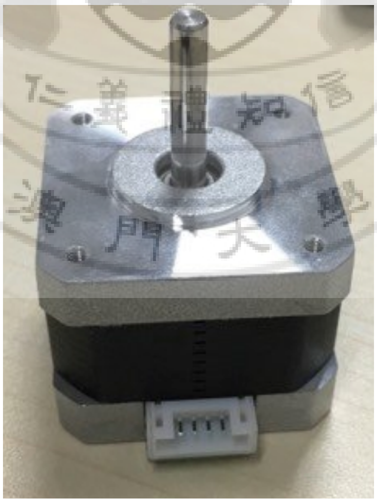


Figure 2-12 Stepper Motor

2.2.2 Nylon Straight Bevel Gears

Gears are common mechanical parts and are widely used in mechanical transmissions. The straight bevel gears are used in this prototype (Fig.2-13). The bevel gears are the main parts that transmit intersecting shaft drive.

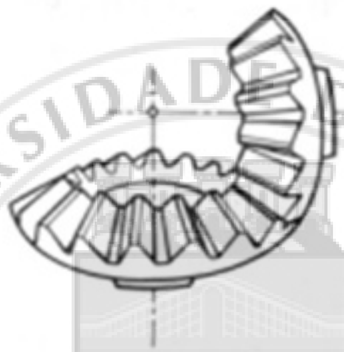


Figure 2-13 Structure of the straight bevel gear

The material for the straight bevel gear is the nylon. Comparing the nylon and the metal, nylon has a good overall performance, including abrasion resistance, heat resistance, chemical resistance and self-lubricating, and the friction coefficient is low. The metal has good thermal conductivity and conductivity, and it is hard, but the disadvantage of metal which is easily rust. Furthermore, the weight of nylon gear is lighter than the metal gear, so the nylon bevel gear (Fig.2-14) is selected for the mechanical part of the wayfinding rotational sign.

Figure 2-14 Nylon bevel gears

Table 2-2 Specification of nylon gear

Parameter	Gear
Module (m)	1.89
Diametral Pitch (P)	0.53
Teeth (N)	26
Pitch Diameter (D)	49mm
Face width (F)	12.5mm

2.2.3 Thrust Bearing

The characteristic of the thrust bearing is the shaft ring and the seat rings are separate components, the shaft ring is compatible with the shaft axis, the seat ring fits the bearing hole.

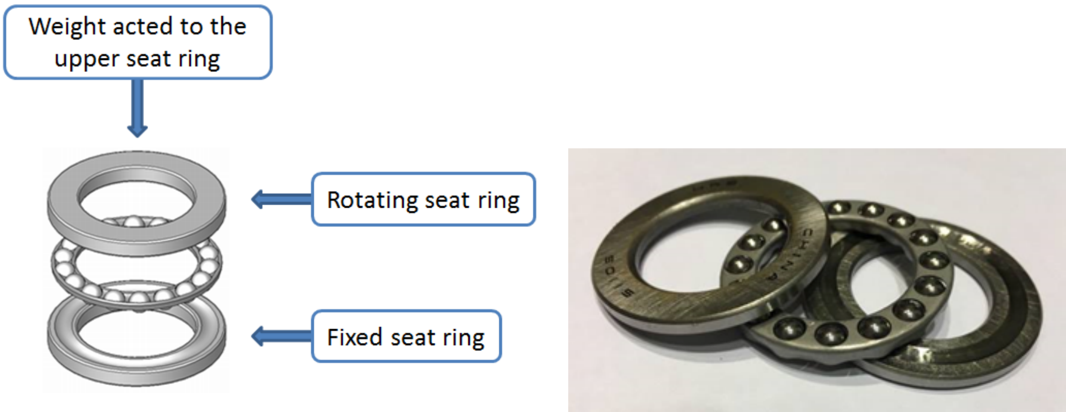


Figure 2-15 Thrust bearing

Figure 2-15 shows that the thrust bearing is to separate the rotating part and the fixed part. One side of the seat ring is attached to the fixed part, and the other side of the seat ring carries the rotating part. In this prototype, the sign is the rotating part and the fixed part is the pillar.

2.2.4 Body Design

To provide a good appearance, the body material for the prototype is transparent acrylic sheet. The body pattern is shown in APPENDIX A.

2.3 SOFTWARE

2.3.1 Operating System

The internal system of RPi 3 model B is the Linux system (Fig.2-16) and it is an UNIX-based operating system, also it is free to download and use.

Figure 2-16 Interface of Linux System in RPi 3 model B

Linux can run on a wide range of hardware, including PCs, Macs, mainframes, supercomputers, some mobile phones and industrial robots [4].

2.3.2 Selection of programming language

Python is selected as the main programming language for this project, because the syntax of Python is similar to C language and compliable to the Linux system. Beside the characteristics of the python programming system include:

1. Literal translation program

   Literal translation program will execute the program step by step, which mean translator is unnecessary to use.

2. Compatible with different kinds of platforms

   Python programming system is compatible with Microsoft, Linux, Mac system etc. This is convenient to the user.

3. Large amount of module to download from the Internet

   Python is an open source programming software, so everyone can design some useful modules and upload it to the Internet for sharing.

2.3.3 Program Design

This part is the core part of the project, since most of the things are controlled by the program and the program is shown in APPENDIX B.

The program can be divided into three parts

1. Interface
2. LED matrix display program
3. Stepper Motor program

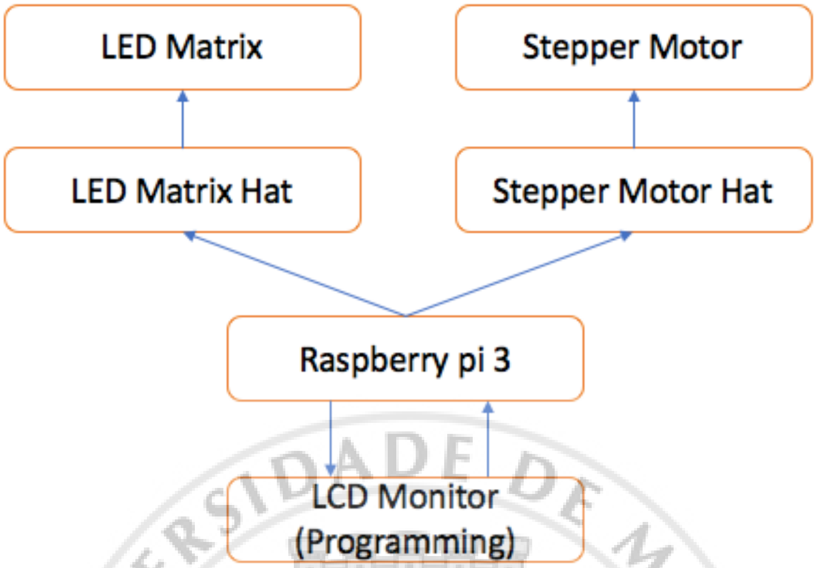


Figure 2-17 Software and Hardware

2.3.3.1 User Interface

Figure 2-18 User interface of the sign

In Fig. 2-18, The picture of Península de Macau is chosen to be the background of the interface. At the top of the interface, it showa the real time, and several buttons are designed to separate different information. In each button, there is a label on it, so the user can easy to select their destination. Besides, to let the user to select the destination, each destination provides a map with route in it. The user just need to click the button next to the destination button, then the map will display.

In Fig. 2-18, the 'Nearest Bus stop' button includes the information of the route of the bus near to this sign, and it briefly shows which route of bus will reach to the specific location. For example: Routes for the A-MA Temple are 9, 18. It provide a suggestion to the user of which bus they should take to reach the destination.

2.3.3.2 LED Matrix Display Program

In the programming part of LED matrix display, it has two functions. It displays the name of the destination which the user chooses in both Chinese and English, as shown in Fig. 2-19 and Fig. 2-20. The Chinese will be displayed first in a marquee way for 3 seconds, then it will display in English. The second function is to show the distance between the destination and the sign as shown in Fig. 2-21 for 3 seconds. The information of the destination is chosen from the database. Also, the font color, display time and the Marquee moving speed could be controlled by the program written in python.

Figure 2-19　Destination name in Chinese

Figure 2-20　Destination name in English

Figure 2-21　Distance between destination and the sign

2.3.3.3 Stepper Motor Program

In this part, when the program receives the signal from RPi 3 to Stepper Motor Hat, it will calculate to the specific angle and drive the stepper motor to turn to the direction of the destination. Through the program, it can control the rotating speed and moving duration of the motor. After the display of the LED Matrix is finished, the program will drive back the motor to the origin. After that, the user can start to choose the other location again.

CHAPTER 3: DESIGN IMPLEMENTATION EVALUATION

The prototype was successfully fabricated and tested. Fig. 3-1 shows the assembly of the prototype. After testing the prototype, some evaluation can be done.

Figure 3-1 Semblance of the prototype

In the assembly, the thrust bearing is placed on the top of the model of the pillar. Then the output bevel gear is placed on the bearing and the input bevel gear is vertically connected with the stepper motor. The steel rod with the LED matrix are placed on the output bevel gear.

After connecting the power supply for the control boards, the user can press the button of the touch screen, then the LED matrix displays the correct information and the stepper motor rotates accurately.

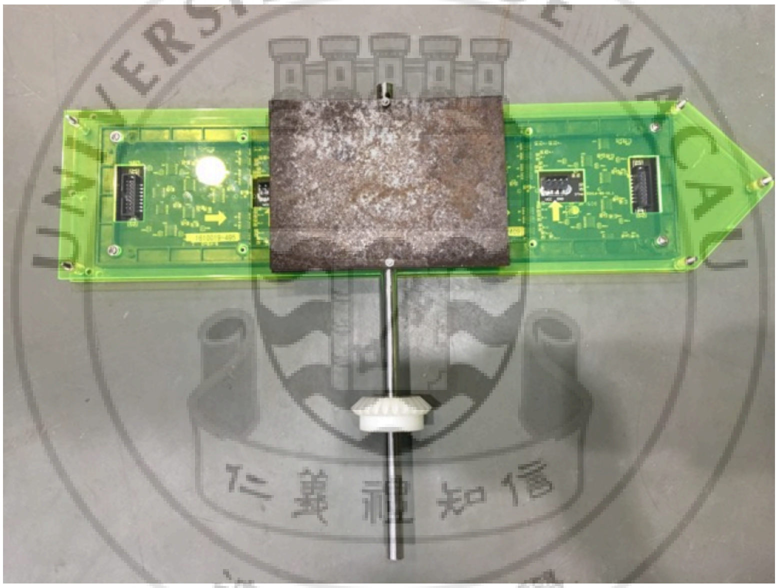


Figure 3-2 Assembly of two matrixes and mechanical parts

Figure 3-2 shows that the assembly of the two LED matrixes, which were fixed on the acrylic sign body with steel rod connecting to the rotational sign.

3.1 CONSTRUCTION COST EVALUATION

The approximate cost of the project is shown below:

Table 3-1 Evaluate cost of the project

Component	Quantity	Cost (MOP)
Raspberry Pi 3 Model B	1	350
LED Matrix Hat	1	250
Stepper Motor Hat	1	250
LCD Touch Monitor	1	300
LED Matrix Board	2	600
Stepper Motor	1	60
Microstep Driver	1	120
Nylon Gear	2	120
Stainless Steel Rod	1	30
Thrust Bearing	1	10
Female DC Power adapter	1	30
2x20 Stacking Header	2	60
Power Supply 5V 8A	1	50
Power Supply 12V 2A	1	50
Acrylic Sheet	5	250
Screw and Nut	~50 of each	10
2x8 data cable	2	10
	Total	2550

From Table 3-1, the cost of the prototype is very low.

3.2 ENERGY CONSUMPTIONS ANALYSIS

The energy consumption is mainly divided into three parts: RPi 3, LED Matrix and the stepper motor. Firstly, the power of these three parts should be determined. The power consumption is shown below:

Power consumption equation: $P = V \times I$ (3-1)

With using the Volt and Ammeter tester, the power consumption of RPi 3 under different modes are:

On sleeping mode (without monitor operation): $P = (5.2V)(0.35A) = 1.82\text{W}$

On normal mode (with monitor operation): $P = (5.2V)(0.4A) = 2.08\text{W}$

Under program processing (when user uses the sign): $P = (5.2V)(0.5A) = 2.6\text{W}$

The power consumption of LED Matrix and the stepper motor are:

Under program processing: $P = Power\ of\ LED\ Matrix + Stepper\ Motor$

$$= (5.2V)(2.5A) + (12V)(0.4A)$$

$$= 17.8\text{W}$$

Since the LCD Monitor with RPi 3 is kept operating most of the time (assume in 24 hours), so the energy consumption per day is

Energy consumption equation: $E = W \times operating\ hours$ (3-2)

Assume the sign is operated for 6 hours and 6 hours on normal mode, then the energy consumption is:

For RPi 3: $E = (2.6W + 2.08W)(6hrs) + (1.82W)(12hrs)$

$= 49.92\text{Wh per day}$

For LED and Motor: $E = (17.8W)(6hrs) = 106.8Wh$ per day

Table 3-2 shows the total energy consumed per day when the LED Matrix and the stepper motor are operated for 0, 6 and 24 hours.

Table 3-2 Evaluation of power consumption of Wayfinding Sign

Power consumed per day	Minimum (0hr)	Ideal (6hrs)	Maximum(24hrs)
Raspberry pi 3	43.68Wh	49.92Wh	62.4Wh
LED Matrix	0Wh	78Wh	312Wh
Stepper Motor	0Wh	28.8Wh	115.2Wh
Total	43.68Wh	156.72Wh	489.6Wh

Since there are some PC-based information systems in the market, such ads Kiosk. The power consumption of the wayfinding sign should be compared with the power consumption of the PC-based information systems.

The energy consumption of the wayfinding sign is lower than desktop computer and laptop computer used in the PC-based information system. For 24 hours' operation, the maximum power consumed in this prototype is 489.6W, while the desktop computer consumes around 6kW and laptop computer consumes around 1.8kW under the same operating condition [4].

3.3 CENTER OF MASS OF THE SIGN

The following calculation aims to find the center of mass of the sign to confirm that the center of mass is always located at a stable position, Fig. 3-3 shows the configuration of the arrow sign.

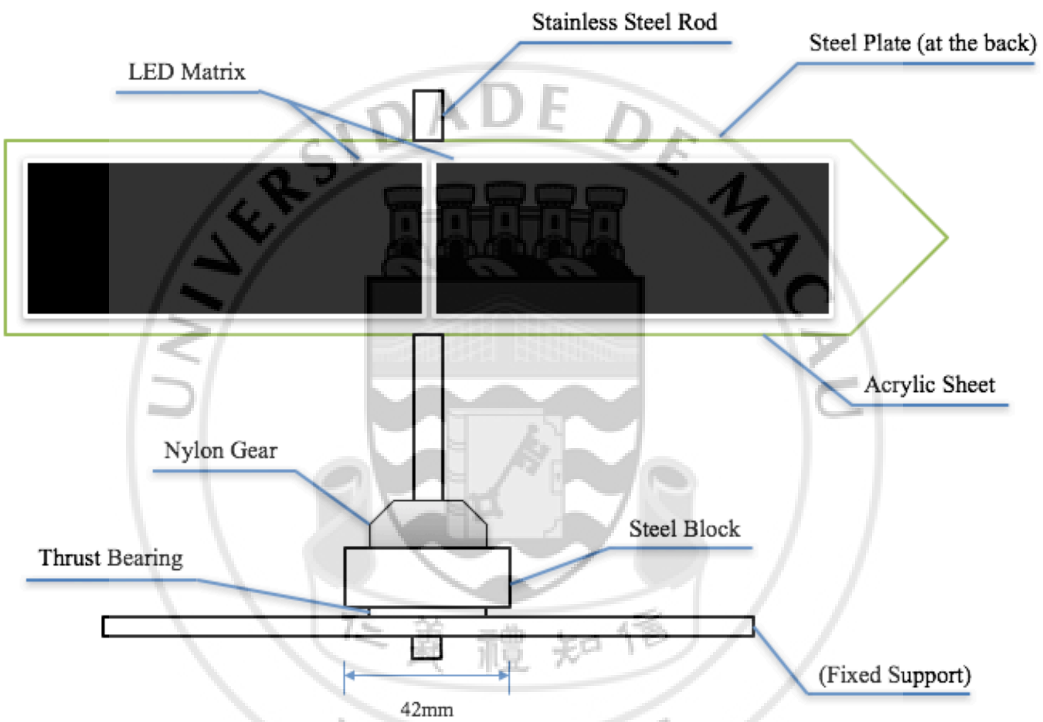


Figure 3-3 Configuration of the arrow sign (Front view)

In Fig. 3-3, the thrust bearing is the support of the sign, so the center of mass must be located inside the diameter of the bearing, the center of mass of the sign by separating into x, y and z-axis.

Center of mass in x-axis:

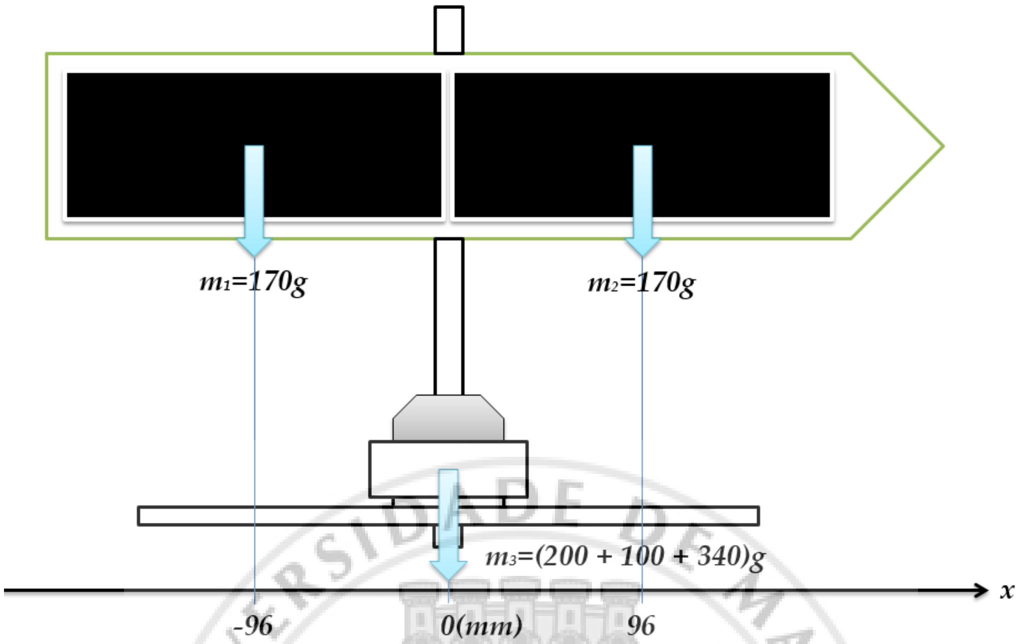


Figure 3-4 Mass distributions in x-axis

Table 3-3 Mass and x-axis of the components

Components		Mass (g)	x-coordinates (mm)
LED Matrix (Left)	m_1	170	+96
LED Matrix (Right)	m_2	170	-96
Stainless Steel Rod	m_3	100	0
Acrylic Sheet		200	~0
Steel Block		340	0
Steel Plate		600	0

$$x = \frac{\sum_{i=3}^{N} m_i x_i}{M} = \frac{m_1x_1 + m_2x_2 + m_3x_3}{m_1 + m_2 + m_3} \tag{3-3}$$

$$= \frac{(170)(96)+(170)(-96)+(200+100+340+600)(0)}{170+170+100+200+340+600}$$

$$= 0$$

Center of mass in y-axis:

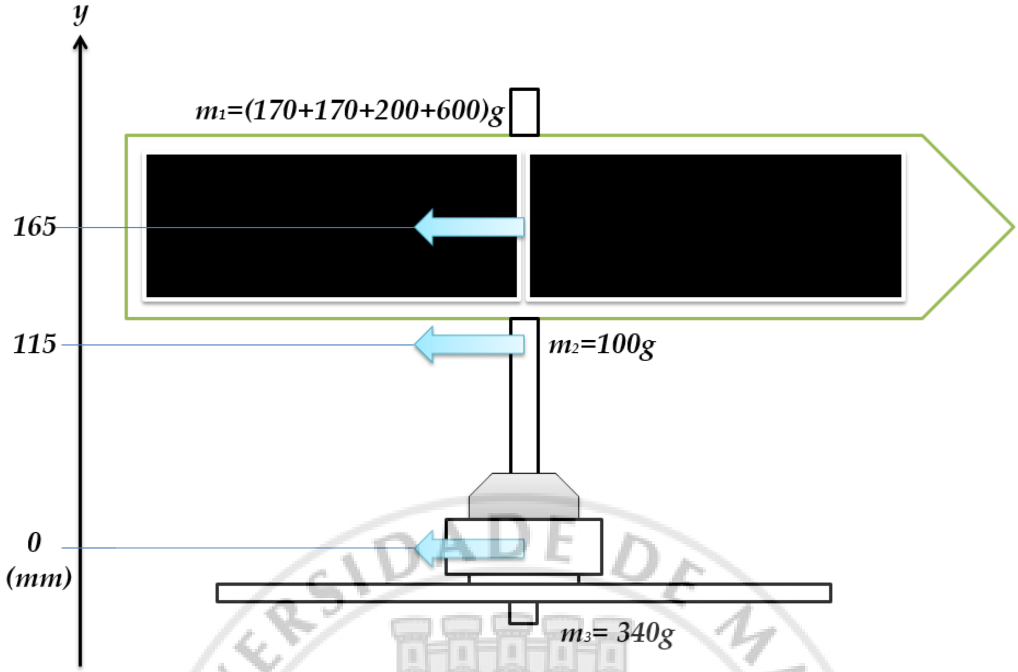


Figure 3-5 Weight distributions in y-axis

Table 3-4 Mass and y-axis of the components

Components		Mass (g)	y-coordinates (mm)
LED Matrix (Left)	m_1	170	+165
LED Matrix (Right)	m_2	170	+165
Stainless Steel Rod	m_3	100	+115
Acrylic Sheet		200	+165
Steel Block		340	0
Steel Plate		600	+165

$$y = \frac{\sum_{i=3}^{N} m_i y_i}{M} = \frac{m_1 x_1 + m_2 x_2 + m_3 x_3}{m_1 + m_2 + m_3} \quad (3\text{-}4)$$

$$= \frac{(170+170+200+600)(165)+(100)(115)+(340)(0)}{170+170+100+200+340+600}$$

$$= 126.33$$

Center of mass in z-axis:

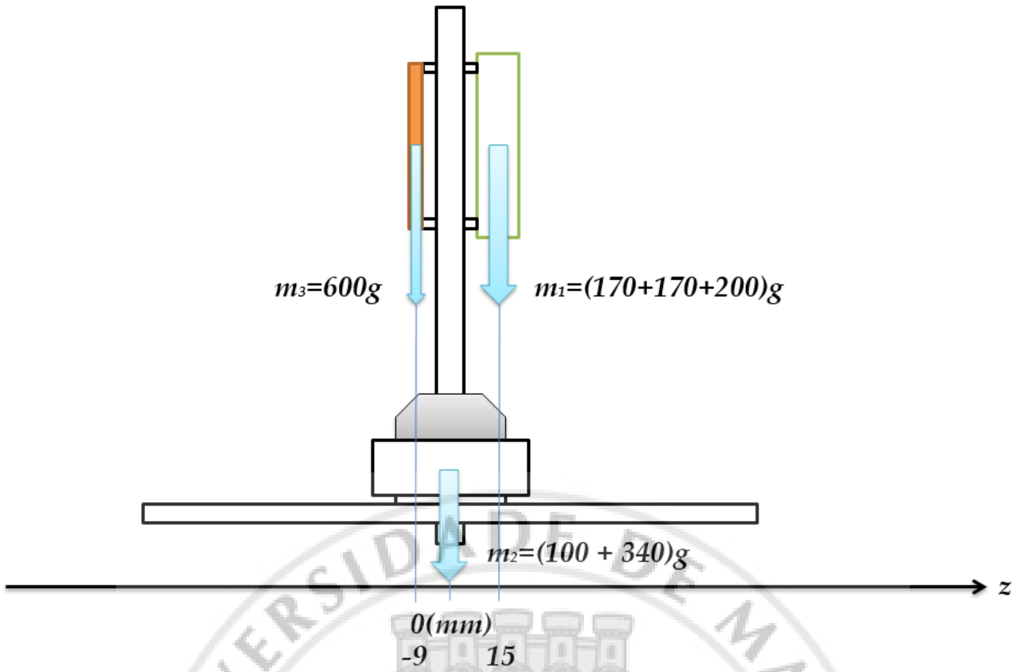


Figure 3-6 Weight distributions in z-axis

Table 3-5 Mass and z-axis of the components

Components		Mass (g)	x-coordinates (mm)
LED Matrix (Left)	m_1	170	+15
LED Matrix (Right)	m_2	170	+15
Stainless Steel Rod		100	0
Acrylic Sheet	m_3	200	+15
Steel Block		340	0
Steel Plate		600	-9

$$z = \frac{\sum_{i=1}^{N} m_i z_i}{M} = \frac{m_1 x_1 + m_2 x_2 + m_3 x_3}{m_1 + m_2 + m_3} \tag{3-5}$$

$$= \frac{(170+170+200)(15)+(100+340)(0)+(600)(-9)}{170+170+100+200+340+600}$$

$$= 1.71$$

By calculation, the center of mass of the sign is (0, 126.33, 1.71). In x and z-axis, the center of mass is located inside the diameter (d = 41mm) of the bearing, so the sign is stable when it is moving.

3.4 STEPPER MOTOR EVALUATION

The minimum τ required of the stepper motor is:

$$\tau = W\mu \times r \times safety\ factor \tag{3-6}$$

τ = Minimum torque required of the stepper motor

W = Loading weight, in N

μ = Coefficient of friction

r = Radius of the bearing, in meters

Assume the safety factor is 2.

$$\tau = (1.58)(9.81)(0.2) \times 0.021 \times 2 = 0.130Nm$$

The minimum power required of the stepper motor is:

$$P_{min} = \tau \times \omega \times safety\ factor \tag{3-7}$$

τ = Minimum torque required of the stepper motor

ω = Angular velocity, in rad/s

Assume the safety factor is 2 and the rpm of stepper motor is 2.

$$P_{min} = (0.130)\times(2)(\frac{2\pi}{60})\times 2 = 0.054W$$

From Table 2-1, the maximum power of the stepper motor is

$$P_{max} = V \times I_{max} \tag{3-8}$$

$$P_{max} = 12 \times 0.4 = 4.8W$$

From Table 2-1, the maximum torque of the stepper motor is 0.3Nm which is larger than 0.130Nm, and the maximum power is 4.8W which is also larger than0.054W, so it is suitable to use the motor given in Table 2-1 for this prototype.

3.5 GEAR MODULE EVALUATION

The minimum module required for the gear in this prototype is:

$$\sigma = \frac{K_v W_t}{FmY} \tag{3-9}$$

$\sigma =$ Bending stress of nylon gear, in MPa

$K_v =$ Velocity factor

$W_t =$ Tangential component of load, in N

$F =$ Face width, in mm (From Table 2-2)

$m =$ Modules, in mm

$Y =$ Lewis Form Factor [5]

First, the velocity factor is:

$$K_v = \frac{3.05+V}{3.05} = \frac{3.05+\frac{\pi dn}{60}}{3.05} \tag{3-10}$$

$V =$ Pitch-line velocity

$d =$ Diameter of the bevel gear, in m (From Table 2-2)

$n =$ Revolution per minute of the gear

Assume the rpm of gear is 2.

$$K_v = \frac{3.05 + \frac{\pi(0.049)(2)}{60}}{3.05} = 1.0014$$

Then, the tangential component of load is:

$$W_t = \frac{60000H}{\pi dn} \tag{3-11}$$

$H =$ Minimum power of the stepper motor

$$W_t = \frac{60000(\frac{0.054}{1000})}{\pi(49)(2)} = 10.5N$$

So, the minimum module required for the gear in this prototype is:

$$\sigma = \frac{K_v W_t}{FmY}$$

$$100 = \frac{(1.0014)(10.5)}{(12.5)(0.346)m}$$

$$m = 0.0253mm$$

Since the module of the gear is 1.89mm > 0.0253mm, it is suitable to use the gear given in Table 2-2 for the prototype.

3.6 DEMONSTRATION

This section demonstrates the operation of the wayfinding sign:

Figure 3-7 User interface of the sign

Fig. 3-7 shows the homepage of the sign; the user can choose the specific location by clicking the buttons on the homepage. In this prototype, there are two main types of locations: Cultural Heritage and Casino/Hotel.

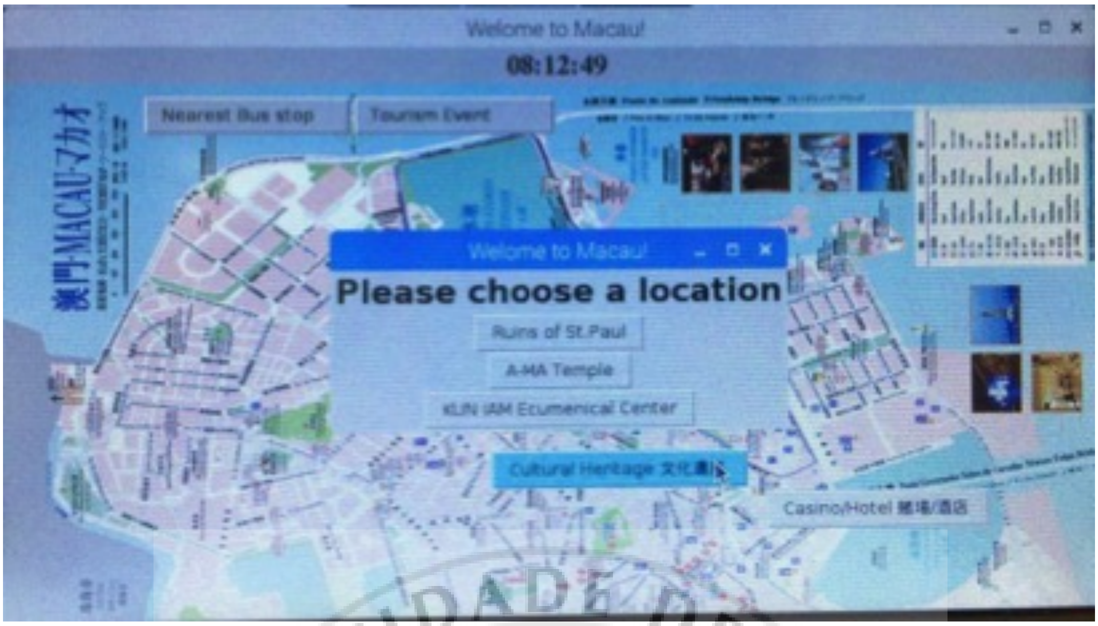


Figure 3-8　　Interface of the locations

When the user clicks the button, then the other interface will pop up in the center, as shown in Fig. 3-8. This interface provides several locations for the user to choose, the user needs to click one of the buttons that they want to go.

Figure 3-9　　Destination name in Chinese

Figure 3-10 Destination name in English

After that, the stepper motor receives the signal and drives the gear to a specific angle, then the LED Matrix shows the name of the destination in both Chinese and English, as shown in Fig 3-9 and 3-10.

Figure 3-11 Distance between the destination and the sign

After that, the sign shows the approximate distance between the destination and the sign (Fig. 3-11). Finally, the Stepper Motor moves back to the origin and the user can choose the other locations.

CHAPTER 4: CONCLUSION AND RECOMMENDATION

4.1 CONCLUSION

This project has successfully designed, fabricated and tested a novel intelligent rotational wayfinding sign which can provide much tourist information of Macau, destination direction and distance. The prototype wayfinding sign consists of mechanical, electrical and software systems. As compared with APPS provided by Macau government, the proposed system can provide travelers information conveniently, particularly for the traveler without a smart phone and the areas where the free Wi-Fi networks are poor.

Testing and evaluation results show that the proposed system is helpful and stable in terms of hardware and software. Moreover, the power consumption is much lower than the PC-based information system. The user also comments the proposed touch-screen panel and user-interface are satisfactory, but there is room for improvement. Therefore, the proposed wayfinding sign is promising.

4.2 PROSPECT OF ROTATIONAL WAYFINDING SIGN IN MACAU

The final goal of the wayfinding sign is expected to locate everywhere and replace the traditional sign to provide different kinds of information to the user such as the location of landmarks, shopping mall, casino, and special events in Macau. By cooperating with the Macau Government, more useful traveling tips and activities in the wayfinding sign, such as "Step out, experience Macau's Communities" can be displayed, so the user can get the latest information directly from the sign.

4.3 RECOMMENDATION OF FUTURE WORK

According to the design of the prototype, there are several points are recommended for the future work.

1. Material

Since the final goal of this project is designed a prototype rotational wayfinding sign, the usage of the material may not be suitable in a real situation. To strengthen the rotational wayfinding sign, aluminum or stainless steel are recommended because the strength, toughness, and durability of metal is normally higher than acrylic sheet.

2. Backend System

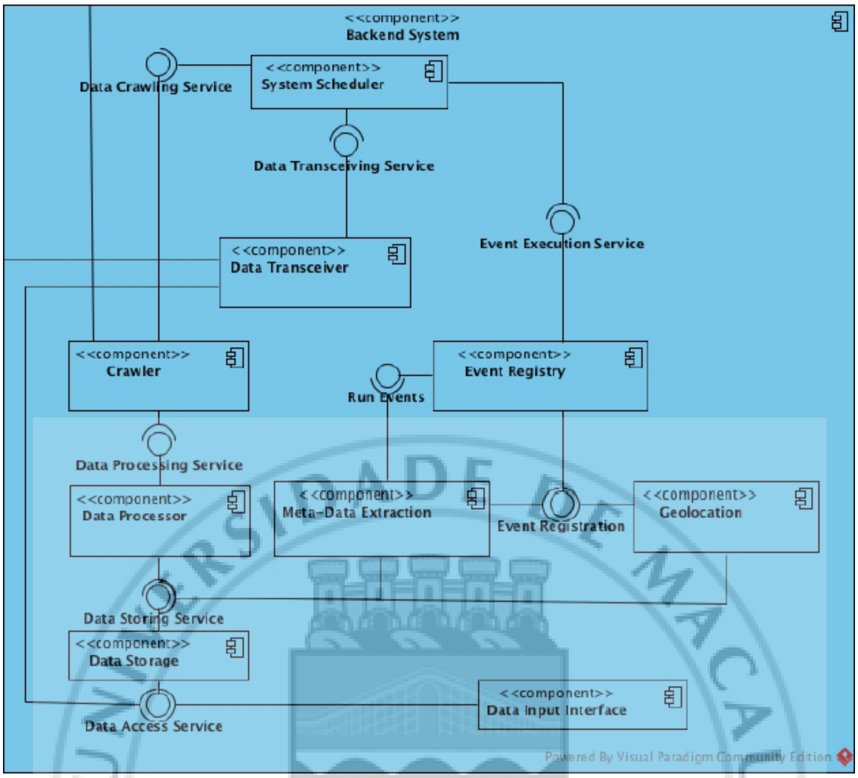


Figure 4-1 Idea of Backend System

In the future, a backend system (Fig.4-1) is considered to create a smart wayfinding sign network in Macau, which means that numerous wayfinding signs need to be constructed, and each sign should provide different kinds of information to the user, therefore a backend system is required. The function of the backend system transfers the information to the wayfinding sign via sensor a 3G/4G or Wi-Fi network. Also, the administration in the backend system can update and control the information to be displayed.

3. More layers of sign / Both sides of LED Matrix

As the main function of the wayfinding sign is to guide the user to reach a specific location, so the display on the sign is important. When the user is interacted with the sign, they are just standing in front of the sign, if the screen on the arrow is rotated to the other side, then the user may find difficult to receive the information form the screen. Therefore, by rotating the same LED Matrix on both sides, then the problem can be solved.

Furthermore, one layer of LED sign on the model can only provide limited information to the user. So, in the future, more layers of LED sign can be added to the wayfinding sign. By programming in Python, then the sign can provide more information at the same time.

4. Weatherproof design

Since most of the wayfinding signs are located at outdoors, so weather proof design should be considered. One of the methods is to use high corrosion resistance materials, such as aluminum alloy or stainless steel, for the body. Besides, there are many electrical components inside the wayfinding sign, therefore, rubber seals and drainage are required.

5. Use of renewable energy

The energy consumption of the sign is relatively low; it may be suitable for storing the renewable energy, such as solar energy, to provide the electricity to the sign.

REFERENCE

[1] *Step out, experience Macau's Communities* from http://en.macaotourism.gov.mo/plan/our_suggested_tours.php

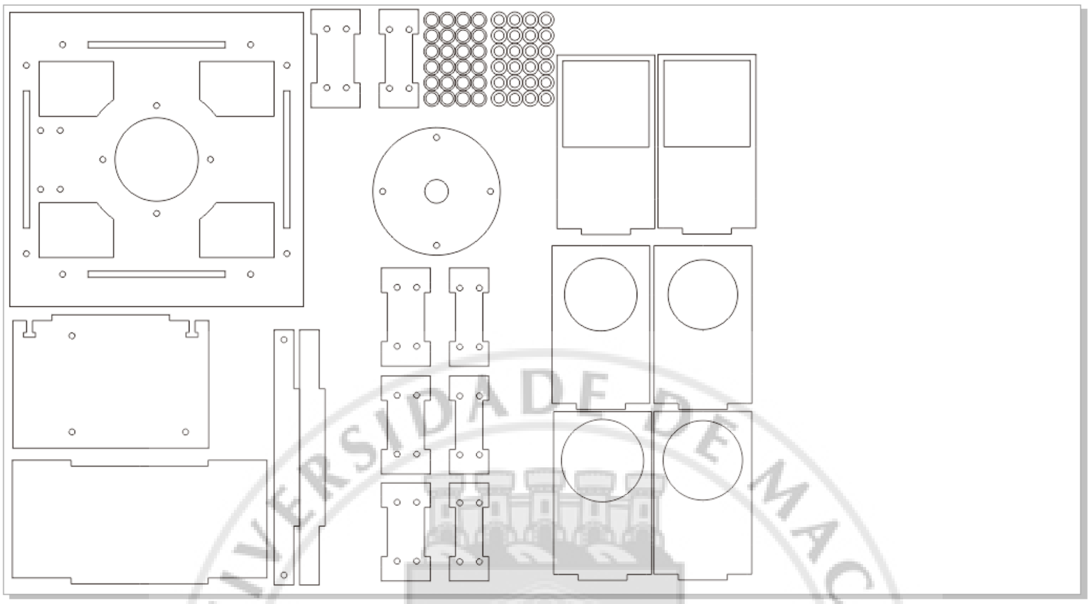
[2] Mark, A. F. (1998, May 20). *Designing Navigable Information Spaces*. Retrieved from http://www.ai.mit.edu/projects/infoarch/publications/mfoltz-thesis/thesis.html

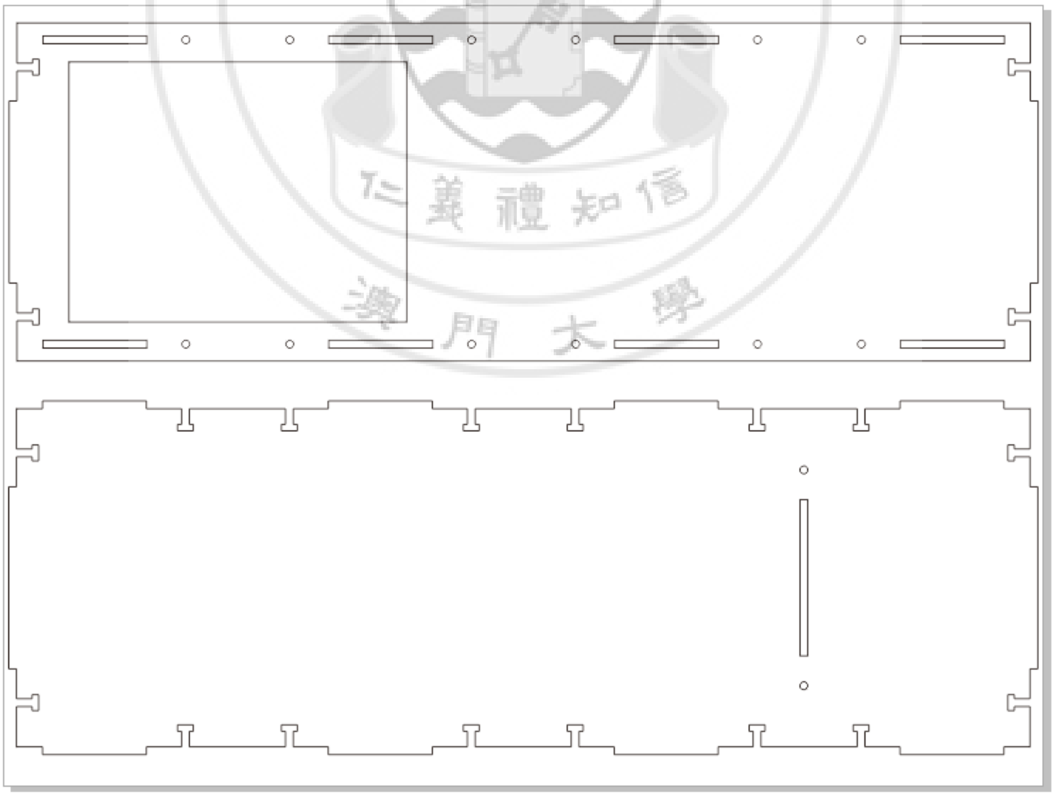
[3] Daniel, R. M., & Corina, S. (2006). *Human Factors of Wayfinding in Navigation*. Retrieved from http://eprints.lancs.ac.uk/42335/1/MontelloSas.pdf

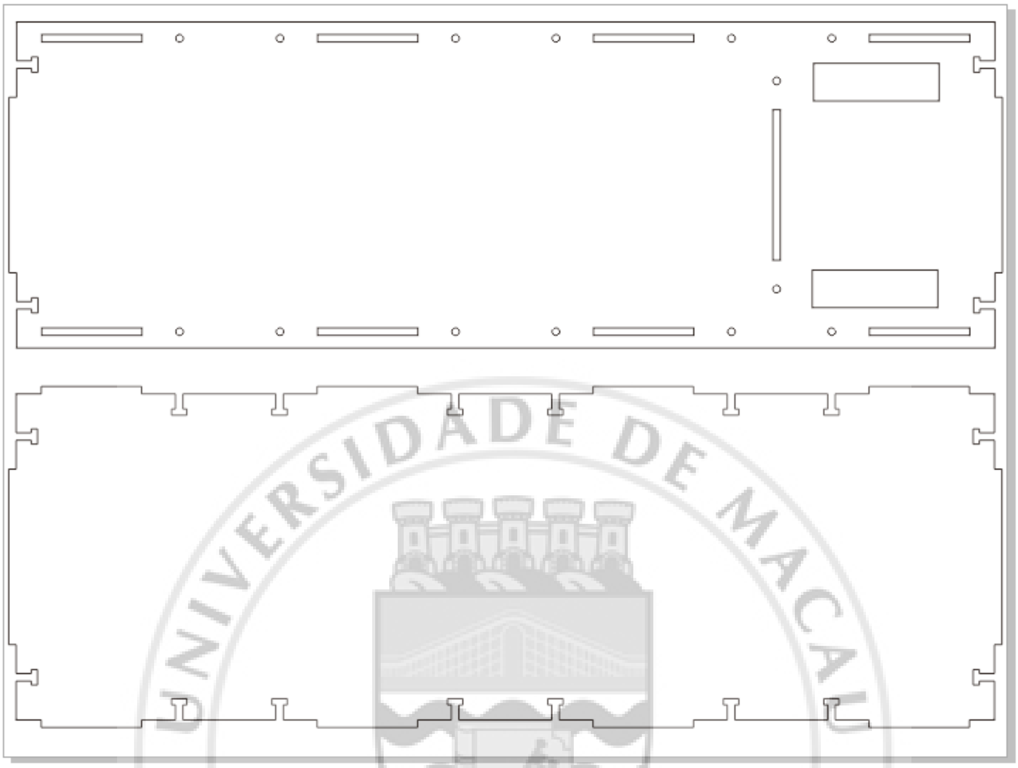
[4] Michael, B. (2012, March). Saving Electricity. Retrieved from http://michaelbluejay.com/electricity/computers.html

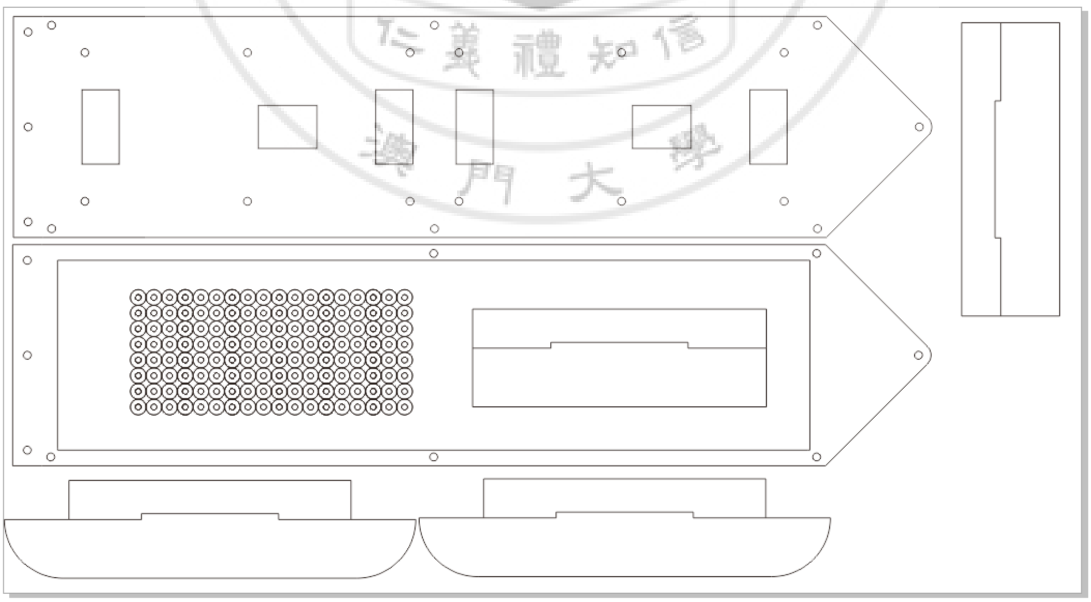
[5] Budynas, R. G., Nisbett, J. K., & Shigley, J. E. (2011). Shigley's mechanical engineering design. New York: McGraw-Hill.

APPENDIX A: BODY DESIGN OF THE PROTOTYPE PILLAR









APPENDIX B: PYTHON CODE

```
# encoding: utf-8

from Tkinter import*
import Image
import ImageDraw
import time
import atexit
import base64
import urllib
import webbrowser

from rgbmatrix import Adafruit_RGBmatrix
from Adafruit_MotorHAT import Adafruit_MotorHAT, Adafruit_DCMotor,
Adafruit_StepperMotor

mh = Adafruit_MotorHAT()

myStepper = mh.getStepper(200, 1)    # 200 steps/rev, motor port #1
myStepper.setSpeed(2)

matrix = Adafruit_RGBmatrix(16, 2)
image = Image.new("1", (32, 32)) # Can be larger than matrix if wanted!!

root= Tk()
root.title("Welome to Macau!")

class Clock:
        def __init__(self):
                self.time1 = ' '
                self.time2 = time.strftime('%H:%M:%S')
                self.mFrame = Frame()
                self.mFrame.pack(side=TOP, expand=YES, fill=X)
```

```
            self.watch = Label(self.mFrame, text=self.time2, font=('times',15,'bold'))
            self.watch.pack()

            self.changeLabel()

        def changeLabel(self):
            self.time2 = time.strftime('%H:%M:%S')
            self.watch.configure(text=self.time2)
            self.mFrame.after(200, self.changeLabel)


def cult1():
        count=0
        count=count + 1

        while count == 1:

            myStepper.step(51, Adafruit_MotorHAT.FORWARD,
Adafruit_MotorHAT.MICROSTEP)

            image = Image.open("cult1.jpg")
            image.load()
            matrix.SetImage(image.im.id, 0, 0)
            time.sleep(3)

            image = Image.open("cult1.jpg")
            image.load()
            for n in range(0, -image.size[0], -1):
                matrix.SetImage(image.im.id, n, 0)
                time.sleep(0.05)

            image = Image.open("cult1_1.jpg")
            image.load()
            matrix.SetImage(image.im.id, 0, 0)
```

```
            time.sleep(3)

            matrix.Clear()

            myStepper.step(51, Adafruit_MotorHAT.BACKWARD,
Adafruit_MotorHAT.MICROSTEP)

            count -= 1


def cult2():
      count=0
      count=count + 1

      while count == 1:

            myStepper.step(76, Adafruit_MotorHAT.FORWARD,
Adafruit_MotorHAT.MICROSTEP)

            image = Image.open("cult2.jpg")
            image.load()
            matrix.SetImage(image.im.id, 5, 0)
            time.sleep(3)

            image = Image.open("cult2.jpg")
            image.load()
            for n in range(5, -image.size[0], -1):
                  matrix.SetImage(image.im.id, n, 0)
                  time.sleep(0.05)

            image = Image.open("cult2_1.jpg")
            image.load()
            matrix.SetImage(image.im.id, 0, 0)
            time.sleep(3)
```

```
            matrix.Clear()

            myStepper.step(76, Adafruit_MotorHAT.BACKWARD,
Adafruit_MotorHAT.MICROSTEP)

            count -= 1


def cult3():
    count=0
    count=count + 1

    while count == 1:

            myStepper.step(100, Adafruit_MotorHAT.BACKWARD,
Adafruit_MotorHAT.MICROSTEP)

            image = Image.open("cult3.jpg")
            image.load()
            matrix.SetImage(image.im.id, 0, 0)
            time.sleep(3)

            image = Image.open("cult3.jpg")
            image.load()
            for n in range(0, -image.size[0], -1):
                matrix.SetImage(image.im.id, n, 0)
                time.sleep(0.05)

            image = Image.open("cult3_1.jpg")
            image.load()
            matrix.SetImage(image.im.id, 0, 0)
            time.sleep(3)

            matrix.Clear()
```

```
            myStepper.step(100, Adafruit_MotorHAT.FORWARD,
Adafruit_MotorHAT.MICROSTEP)

            count -= 1


def hotel1():
        count=0
        count=count + 1

        while count == 1:

            myStepper.step(35, Adafruit_MotorHAT.BACKWARD,
Adafruit_MotorHAT.MICROSTEP)

            image = Image.open("hotel1.jpg")
            image.load()
            matrix.SetImage(image.im.id, -5, 0)
            time.sleep(3)

            image = Image.open("hotel1.jpg")
            image.load()
            for n in range(-5, -image.size[0], -1):
                matrix.SetImage(image.im.id, n, 0)
                time.sleep(0.05)

            image = Image.open("hotel1_1.jpg")
            image.load()
            matrix.SetImage(image.im.id, 0, 0)
            time.sleep(3)

            matrix.Clear()

            myStepper.step(35, Adafruit_MotorHAT.FORWARD,
Adafruit_MotorHAT.MICROSTEP)
```

```
            count -= 1


def hotel2():
        count=0
        count=count + 1

        while count == 1:

            myStepper.step(35, Adafruit_MotorHAT.FORWARD,
Adafruit_MotorHAT.MICROSTEP)

            image = Image.open("hotel2.jpg")
            image.load()
            matrix.SetImage(image.im.id, 10, 0)
            time.sleep(3)

            image = Image.open("hotel2.jpg")
            image.load()
            for n in range(10, -image.size[0], -1):
                matrix.SetImage(image.im.id, n, 0)
                time.sleep(0.05)

            image = Image.open("hotel2_1.jpg")
            image.load()
            matrix.SetImage(image.im.id, 0, 0)
            time.sleep(3)

            matrix.Clear()

            myStepper.step(35, Adafruit_MotorHAT.BACKWARD,
Adafruit_MotorHAT.MICROSTEP)

            count -= 1
```

```
def hotel3():
        count=0
        count=count + 1

        while count == 1:

                myStepper.step(50, Adafruit_MotorHAT.FORWARD,
Adafruit_MotorHAT.MICROSTEP)

                image = Image.open("hotel3.jpg")
                image.load()
                matrix.SetImage(image.im.id, -8, 0)
                time.sleep(3)

                image = Image.open("hotel3.jpg")
                image.load()
                for n in range(-8, -image.size[0], -1):
                        matrix.SetImage(image.im.id, n, 0)
                        time.sleep(0.05)

                image = Image.open("hotel3_1.jpg")
                image.load()
                matrix.SetImage(image.im.id, 0, 0)
                time.sleep(3)

                matrix.Clear()

                myStepper.step(50, Adafruit_MotorHAT.BACKWARD,
Adafruit_MotorHAT.MICROSTEP)

                count -= 1

def bus1():
```

```
    root= Tk()
    root.withdraw()
    root.update_idletasks()
    x = 180
    y = 200
    label1=Label(root, text="Hotel Lisboa  葡京酒店: Route 3,3A,3X,25X,26A,33", font=('Arial',12,'bold'))
    label2=Label(root, text="Galaxy  銀河: Route 25,25X,26A,35,MT1,MT4", font=('Arial',12,'bold'))
    label3=Label(root, text="Wynn Macau  永利澳門: Route 3,3A,3X,25X,26A,33", font=('Arial',12,'bold'))
    root.geometry("+%d+%d" % (x,y))
    root.title("Welome to Macau!")
    label1.pack()
    label2.pack()
    label3.pack()
    root.deiconify()

def bus2():
    root= Tk()
    root.withdraw()
    root.update_idletasks()
    x = 130
    y = 200
    label1=Label(root, text="Ruins of St.Paul  大三巴牌坊: Route 3,3A,3X,25X,26A,33", font=('Arial',12,'bold'))
    label2=Label(root, text="A-MA Temple  媽閣廟: Route 25,25X,26A,35,MT1,MT4", font=('Arial',12,'bold'))
    label3=Label(root, text="KUN IAM Ecumenical Center  觀音像: Route 3,3A,3X,25X,26A,33", font=('Arial',12,'bold'))
    root.geometry("+%d+%d" % (x,y))
    root.title("Welome to Macau!")
    label1.pack()
    label2.pack()
    label3.pack()
```

```
    root.deiconify()

def bus3():
    root= Tk()
    root.withdraw()
    root.update_idletasks()
    x = 100
    y = 200
    label1=Label(root, text="Macau International Airport 澳門國際機場: Route 
26,36,AP1,MT1,MT2", font=('Arial',12,'bold'))
    label2=Label(root, text="Macau Maritime Ferry Terminal 港澳碼頭: Route 
3,3A,10,10A,28A,AP1", font=('Arial',12,'bold'))
    label3=Label(root, text="Boarder Gate 關閘: Route 1,3,5,10,25,AP1", 
font=('Arial',12,'bold'))
    root.geometry("+%d+%d" % (x,y))
    root.title("Welome to Macau!")
    label1.pack()
    label2.pack()
    label3.pack()
    root.deiconify()


def cult():
    root= Tk()
    root.withdraw()
    root.update_idletasks()
    x = 235
    y = 200
    root.geometry("+%d+%d" % (x,y))
    root.title("Welome to Macau!")
    label1=Label(root, text="Please choose a location", font=('Arial',18,'bold'))
    button5=Button(root, text="Ruins of St.Paul 大三巴牌坊", command=cult1)
    button6=Button(root, text="A-MA Temple 媽閣廟", command=cult2)
    button7=Button(root, text="KUN IAM Ecumenical Center 觀音像", 
command=cult3)
```

```
        label1.pack()
        button5.pack()
        button6.pack()
        button7.pack()
        root.deiconify()

def hotel():
        root= Tk()
        root.withdraw()
        root.update_idletasks()
        x = 240
        y = 200
        root.geometry("+%d+%d" % (x,y))
        root.title("Welome to Macau!")
        label1=Label(root, text="Please choose a location", font=('Arial',18,'bold'))
        button5=Button(root, text="Hotel Lisboa  葡京酒店", command=hotel1)
        button6=Button(root, text="Galaxy  銀河", command=hotel2)
        button7=Button(root, text="Wynn Macau  永利澳門", command=hotel3)
        label1.pack()
        button5.pack()
        button6.pack()
        button7.pack()
        root.deiconify()

def bus():
        root= Tk()
        root.withdraw()
        root.update_idletasks()
        x = 250
        y = 200
        root.geometry("+%d+%d" % (x,y))
        root.title("Welome to Macau!")
        button5=Button(root, text="Casino/Hotel  賭場/酒店", command=bus1)
        button6=Button(root, text="Cultural Heritage  文化遺產", command=bus2)
```

```
    button7=Button(root, text="Airport/Terminal/Boarder gate  機場/碼頭/關閘",
command=bus3)
    button5.pack()
    button6.pack()
    button7.pack()
    root.deiconify()

def tourism():
    global count
    count+=1

    while count == 1:

webbrowser.open("http://en.macaotourism.gov.mo/plan/our_suggested_tours.php
",new=0);
        count -= 1

URL =
"https://apionline.sodapdf.com/Public/widgets/convertmyimage/download/map2.gi
f"
link = urllib.urlopen(URL)
raw_data = link.read()
link.close()
next = base64.encodestring(raw_data)
image1 = PhotoImage(data=next)
canvas = Canvas(root, width = 850, height = 410)
canvas.create_image(0,0, anchor = NW, image=image1)

button=Button(root, text="Cultural Heritage  文化遺產", command=cult,
              anchor='w', width = 20, activebackground = "#33B5E5")
button_window = canvas.create_window(350,270, anchor = NW, window=button)

button2=Button(root, text="Casino/Hotel  賭場/酒店", command=hotel,
              anchor='w', width = 17, activebackground = "#33B5E5")
button2_window = canvas.create_window(550,300, anchor = NW, window=button2)
```

```
button3=Button(root, text="Route of Bus Info", command=bus,
                    anchor='w', width = 15, activebackground = "#33B5E5")
button3_window = canvas.create_window(100,10, anchor = NW, window=button3)

button4=Button(root, text="Tourism Event", command=tourism,
                    anchor='w', width = 15, activebackground = "#33B5E5")
button4_window = canvas.create_window(250,10, anchor = NW, window=button4)

obj1 = Clock()

canvas.pack()
root.mainloop()
```

WORKLOAD BREAKDOWN