

หุ่นยนต์ต้อนรับสำหรับนำเสนอข้อมูลแก่ผู้เยี่ยมชม

PR LABORATORY ROBOTIC

พัศพงษ์ เจริญพันธ์

PATSAPONG CHAROENPHAN

เลิศชัย ขันติยานุวัฒน์

LERDCHAI KHANTIYANUWAT

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2557

หุ่นยนต์ต้อนรับสำหรับนำเสนอข้อมูลแก่ผู้เยี่ยมชม

## PR LABORATORY ROBOTIC

พัศพงษ์ เจริญพันธ์

PATSAPONG CHAROENPHAN

เลิศชัย ขันติยานุวัฒน์

LERDCHAI KHANTIYANUWAT

อาจารย์ที่ปรึกษา

ผศ.ดร. ภัทรพงษ์ ผาสุกกิจ

ปฏิญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2557

ปริญญาานิพนธ์ปีการศึกษา 2557

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์ต้อนรับสำหรับนำเสนอข้อมูลแก่ผู้เยี่ยมชม

PR Laboratory Robotic

ผู้จัดทำ นายพัศพงษ์ เจริญพันธ์ 54010906

นายเลิศชัย ชันติญาณวัฒน์ 54011101

ปริญญาานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

๐  
๙๙-๙๙



(ผศ.ดร.ภัทรพงษ์ ผาสุขกิจ)

อาจารย์ที่ปรึกษา

หัวข้อโครงการ	หุ่นยนต์ต้อนรับสำหรับนำเสนอข้อมูลแก่ผู้เยี่ยมชม
นักศึกษา	นายพัศพงษ์ เจริญพันธ์ 54010906
	นายเลิศชัย ชันติญาณวัฒน์ 54011101
สาขา	วิศวกรรมอิเล็กทรอนิกส์
พ.ศ.	2558
อาจารย์ที่ปรึกษา	ผศ.ดร. ภัทรพงษ์ ผาสุขกิจ

#### บทคัดย่อ

โครงการนี้จัดทำขึ้นเพื่ออำนวยความสะดวกให้กับบุคคลที่มาเยี่ยมชมห้อง BURN LAB โดยในการทำหุ่นยนต์จะมีส่วนประกอบหลักอยู่ 2 ส่วน คือส่วนโครงสร้างหุ่นยนต์ และส่วนการประมวลผลและควบคุม ในส่วนแรกโครงสร้างของหุ่นยนต์จะมีส่วนของการเคลื่อนที่ของหุ่นยนต์โดยใช้ motor ในส่วนของหน้าตาหุ่นยนต์ จะใช้ iPad เป็นตัวแสดงผลและติดต่อกับผู้ใช้ ในส่วนที่สองคือกาประมวลผลและควบคุมจะใช้บอร์ด RaspberryPi และ Arduino โดยบอร์ด RaspberryPi จะเปรียบเสมือนกับสมองของหุ่นยนต์ที่คิดวิเคราะห์แล้วส่งข้อมูลไปให้ Arduino ควบคุมการเคลื่อนไหวของหุ่นยนต์ หลักการทำงานของหุ่นยนต์คือรับคำสั่งคิดวิเคราะห์พูดโต้ตอบเคลื่อนไหว และสามารถแสดงหน้าตาอารมณ์

Project Title	PR Laboratory Robotic	
Student	Mr. Patsapong Jaronphan	54010906
	Mr. Lerdchai Khantiyanuwatt	54011101
Program	Electronics Engineering	
Year	2014	
Project Advisor	Asst.Prof. Dr. Pattarapong Phasukkit	

### **Abstract**

In this project, We research about Robot for BURN lab room. We have 2 main path to make robot. First path is mechanic of robot. Second path is analysis and control robot. In first path we use motor for robot action. In robot emotions we use iPad to show robot face and user interface. Path 2 analysis and control robot we use RaspberryPi and Arduino. RaspberryPi look like the brains of robot to analysis command. then send data to Arduino for control action of robot. Principle of robot are receive the command to analysis then the robot is talking, action and emotion.

## กิตติกรรมประกาศ

โครงการฉบับนี้สำเร็จเป็นรูปเล่มได้โดยสมบูรณ์ ผู้วิจัยขอขอบคุณ ผศ.ดร.ภัทรพงษ์ ผาสุขกิจ ผู้ให้คำปรึกษาแนวทางตลอดจนชี้แนะให้โครงการเรื่องนี้บรรลุเป้าหมายตามที่คาดไว้

ขอขอบคุณพี่ๆ เพื่อนๆ และน้องๆ ร่วมห้องโปรเจกต์ที่สละเวลาอันมีค่าในการให้คำปรึกษา ช่วยทดสอบระบบมากมาย และสุดท้ายนี้ขอกราบพระคุณ แต่ ครอบครัวของผู้จัดทำผู้ให้การสนับสนุนและเป็นกำลังใจทางด้านการเรียนมาโดยตลอด

นายพัศพงศ์ เจริญพันธ์  
นายเลิศชัย ชันติญาณวัฒน์

## สารบัญ

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูป.....	VIII
สารบัญตาราง.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการทำโครงการ.....	1
1.3 สมมติฐานของการศึกษา.....	1
1.4 ทฤษฎีหรือแนวคิดในการวิจัย.....	1
1.5 ขอบเขตของงานวิจัย.....	2
บทที่ 2 ความรู้พื้นฐานเกี่ยวกับโครงการ.....	3
2.1 Raspberry PI.....	3
2.2 Arduino.....	5
2.3 API (Application Programming Interface).....	6
2.4 การวิเคราะห์เสียง(Speech recognition).....	7

## สารบัญ (ต่อ)

บทที่ 3 การออกแบบและกระบวนการต่างๆ.....	8
3.1 หน่วยประมวลผล.....	8
3.1.1 ติดตั้งระบบปฏิบัติการ Raspbian.....	8
3.1.2 การติดตั้งโปรแกรมสร้างคำสั่ง wit.ai .....	9
3.1.3 เขียนโปรแกรมเปลี่ยนจากเสียงเป็นตัวหนังสือผ่าน api ของ wit.ai.....	11
3.1.4 เขียนโปรแกรมเปลี่ยนจากตัวหนังสือเป็นเสียงพูดผ่าน api ของ google..	12
3.1.5 เขียนโปรแกรมแกนหลักการคิด.....	13
3.1.6 เขียนโปรแกรมส่วนเสริมต่างๆ.....	14
3.1.7 ติดตั้งโปรแกรมที่ใช้ในการเชื่อมต่อข้อมูลระหว่าง RaspberryPi และ Device อื่นๆ.....	17
3.2 ส่วนของการขับเคลื่อนของหุ่นยนต์.....	21
3.2.1 ออกแบบโครงหุ่นยนต์.....	21
3.2.2 สร้างบอร์ดขับเคลื่อนมอเตอร์.....	22
3.2.3 สร้างบอร์ดหลักของการขับเคลื่อนหุ่นยนต์.....	24
3.2.4 เขียนโปรแกรมการขับเคลื่อนของหุ่นยนต์.....	26
3.3 ส่วนของการแสดงผลบน iPad.....	27
3.3.1 ส่วนของการแสดงหน้าตา.....	27
3.3.2 เขียนโปรแกรม.....	29

## สารบัญ (ต่อ)

3.4 ทำวิดีโอพรีเซนห้อง lab.....	30
3.4.1 ถ่ายบรรยากาศ / สัมภาษณ์งานเพื่อนๆ.....	30
3.4.2 ตัดต่อ.....	32
บทที่ 4 การทดลอง.....	33
4.1 ทดสอบการสนทนาถามตอบกับหุ่นยนต์.....	33
4.1.1 ให้แนะนำตัว.....	33
4.1.2 ให้แนะนำห้อง lab.....	34
4.1.3 ถามเพื่อหาคำตอบด้วย google / youtube.....	35
4.1.4 โหมดเลขา .....	36
4.2 ทดสอบการเคลื่อนที่ของหุ่นยนต์.....	37
4.2.1 เรียกในทิศต่างๆ.....	37
4.2.2 เรียกให้เข้ามาหา.....	38
4.2.3 ทดสอบการชนของหุ่นยนต์.....	38
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ.....	39
5.1 วิจัยและสรุปผล.....	39
5.2 แนวทางการพัฒนา.....	39
เอกสารอ้างอิง.....	40
ภาคผนวก การเขียนโปรแกรม.....	41

## สารบัญรูป

### บทที่ 2 ความรู้พื้นฐานเกี่ยวกับโครงการ

2.1 Raspberry Pi.....	4
2.2 ตำแหน่งขา GPIO ของ Raspberry Pi.....	4
2.3 บอร์ด Arduino Mega2560.....	5
2.4 ตัวอย่างการทำงานของ API.....	6

### บทที่ 3 การออกแบบและกระบวนการต่างๆ

3.1.1 เว็บไซต์ดาวน์โหลดระบบปฏิบัติการ Raspbian.....	8
3.1.2 ติดตั้งระบบปฏิบัติการลงแผ่น SDcard.....	8
3.1.3 ย้ายไฟล์คอมไพล์ wit-arm ไปไว้ใน RaspberryPi.....	9
3.1.4 ติดตั้งไลบรารีโปรแกรม.....	9
3.1.5 หน้าต่างการจัดการ wit.ai.....	10
3.1.6 หน้าต่างคำพูดต่างๆที่สอน.....	10
3.1.7 หลักการทำงานเปลี่ยนจากเสียงเป็นตัวหนังสือเป็นเสียง.....	11
3.1.8 ออกแบบโปรแกรมแกนหลักความคิด.....	13
3.1.9 ออกแบบการค้นหาคำตอบด้วย wolfram and google.....	14
3.1.10 ออกแบบระบบผู้ช่วย.....	15
3.1.11 ออกแบบระบบเปิด youtube.....	16
3.1.12 ออกแบบระบบแนะนำห้องแลป.....	16

## สารบัญญรูป (ต่อ)

3.1.13 อัปเดตโปรแกรม.....	17
3.1.14 ติดตั้ง firmware.....	17
3.1.15 ตั้งค่า LAN.....	18
3.1.16 การเชื่อมต่อกับ Ipad.....	18
3.1.17 ติดตั้ง webserver.....	19
3.1.18 ผลลัพธ์.....	19
3.1.19 ติดตั้งไลบรารีที่มิชเชื่อมต่อกับ Arduino.....	20
3.2.1 ออกแบบหุ่นยนต์.....	21
3.2.2 schematic ของวงจรขับเคลื่อน.....	22
3.2.3 PCB ของวงจรขับเคลื่อน.....	22
3.2.4 วงจรขับเคลื่อน.....	23
3.2.5 บอร์ดหลักที่เชื่อมต่อ Arduino กับ sensor ต่างๆ.....	24
3.2.6 รูปผลลัพธ์หลังจากกักบอร์ด.....	25
3.2.7 ออกแบบระบบแนะนำห้องแลป.....	26
3.3.1 รูปหน้าตาหุ่นยนต์ตอนกระพริบตา.....	27
3.3.2 รูปหน้าตาหุ่นยนต์ตอนยิ้ม.....	28
3.3.3 หน้าต่างเขียนโปรแกรม xcode.....	29
3.3.4 การถ่ายทำบรรยากาศในห้อง.....	30

## สารบัญรูป (ต่อ)

3.3.5 การถ่ายทำบรรยากาศในห้อง 2.....	30
3.3.6 การถ่ายทำบรรยากาศในห้อง 3.....	31
3.3.7 ตัดต่อผ่านโปรแกรม iMovie.....	32
บทที่ 4 การทดลอง	
4.1.1 พูดคำสั่งให้แนะนำตัว.....	33
4.1.2 พูดคำสั่งให้แนะนำห้อง burn lab.....	34
4.1.3 คำสั่งใช้งาน google.....	35
4.1.4 แสดงอีเมลที่เข้ามาหลังจากใช้คำสั่งเลข.....	36
4.2.1 เรียกหุ่นยนต์ให้หันมาทิศทางที่เรียก.....	37
4.2.2 เรียกให้หุ่นยนต์เข้ามาหา.....	38
4.2.3 หยุดเมื่อใกล้ชน.....	38

## สารบัญตาราง

### ตารางที่

4.1.1	จับเวลาการสั่งงาน คำสั่ง what is your name.....	33
4.1.2	เวลาการสั่งงาน คำสั่ง who are you.....	33
4.1.3	จับเวลาการสั่งงาน คำสั่ง please introduce lab.....	34
4.1.4	จับเวลาการสั่งงาน คำสั่ง what is burnlab.....	34
4.1.5	จับเวลาการสั่งงาน คำสั่ง how many people in world.....	35
4.1.6	จับเวลาการสั่งงาน คำสั่ง how many people in thailand.....	35
4.1.7	จับเวลาการสั่งงาน คำสั่ง where is my adviser.....	36
4.1.8	จับเวลาการสั่งงาน คำสั่ง i want to see adviser.....	36

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

เทคโนโลยีเรื่องของหุ่นยนต์ เป็นเทคโนโลยีที่นับวันจะยิ่งมีการพัฒนามากยิ่งขึ้นและเป็นเทคโนโลยีที่ถูกจับตามองในหลายๆภาคส่วน อาทิเช่น ด้านอุตสาหกรรม ด้านการศึกษา ด้านการแพทย์ ด้านเกษตรกรรม และอื่นๆอีกมาก เทคโนโลยีหุ่นยนต์มีบทบาทมากขึ้นเรื่อยๆในสังคมสมัยใหม่ เราใช้เทคโนโลยีหุ่นยนต์มาแทนที่การทำงานของมนุษย์มากยิ่งขึ้น ส่วนหนึ่งเป็นเพราะหุ่นยนต์ไม่ต้องการพักผ่อนสามารถทำงานได้ตลอดเวลาแล้วถ้าเป็นงานที่ต้องการทำอะไรซ้ำๆไปมาหุ่นยนต์จะทำงานได้รวดเร็วกว่ามาก

จากเหตุผลดังกล่าวทางกลุ่มของเราจึงสนใจศึกษาเกี่ยวกับการทำหุ่นยนต์โดยมีจุดประสงค์หลักคือ เพื่อใช้ในการอำนวยความสะดวกต่างๆภายในห้องแลปและสามารถอำนวยความสะดวกแก่ผู้รับชม ทั้งยังสามารถบอกเล่า อธิบายโปรเจคในห้องแลป

### 1.2 ความมุ่งหมายและวัตถุประสงค์ของโครงการ

1. เพื่อศึกษาการออกแบบหุ่นยนต์
2. เพื่อศึกษาการใช้งานเบื้องต้นของบอร์ด RaspberryPi
3. เพื่อศึกษาการเขียนโปรแกรมของบอร์ด Arduino เพื่ออ่าน sensor ต่างๆ และควบคุมมอเตอร์
4. เพื่อศึกษาการเขียนโปรแกรมการติดต่อสื่อสารระหว่าง RaspberryPi และ Arduino
5. เพื่อศึกษาการเขียนโปรแกรมแอปพลิเคชันบน iPad
6. เพื่อศึกษาการเขียนโปรแกรมการติดต่อสื่อสารระหว่าง RaspberryPi และ iPad

### 1.3 สมมติฐานของการศึกษา

หุ่นยนต์ต้อนรับสำหรับนำเสนอข้อมูลแก่ผู้เยี่ยมชมจะต้องรับคำสั่งเสียงแล้วปฏิบัติตามได้อย่างถูกต้อง สามารถเคลื่อนที่ไปมาเวลาสั่งการด้วยเสียงได้อย่างแม่นยำและแสดงหน้าความรู้สึกร่างกายออกมาได้

### 1.4 ทฤษฎีหรือแนวคิดการวิจัย

แนวคิดมาจากการต้องการพัฒนาต่อยอดทางความคิดเรื่องหุ่นยนต์อย่างจริงจัง โดยการศึกษาค้นคว้าในเรื่องต่างๆ ที่จะทำให้หุ่นยนต์มีความคิด ความสามารถในการคิดวิเคราะห์ สามารถตอบสนองต่อคำสั่งเสียง และสามารถสื่อสารอารมณ์ผ่านหน้าตาแอปพลิเคชัน เป็นการพลิกแพลงรวบรวมศาสตร์ในหลายๆเรื่องมาพัฒนาต่อยอดทำเป็นหุ่นยนต์ต้อนรับสำหรับนำเสนอข้อมูลแก่ผู้เยี่ยมชม

### 1.5 ขอบเขตของการวิจัย

การพัฒนาหุ่นยนต์ต้อนรับสำหรับนำเสนอข้อมูลแก่ผู้เยี่ยมชมต้องการศึกษาความรู้อื่นๆ รวบรวมมาเป็นหุ่นยนต์ ซึ่งในบางส่วนเป็นนำความรู้มาต่อยอดเช่น การวิเคราะห์เสียงโดยใช้ API ของ wit.ai จะยังไม่มีความแม่นยำเท่าที่ควร และในการตอบคำถาม หรือสั่งงานบางคำสั่งจำเป็นที่จะต้องใช้ Internet ในการค้นหาคำตอบต่างๆ

## บทที่ 2

### ความรู้พื้นฐาน

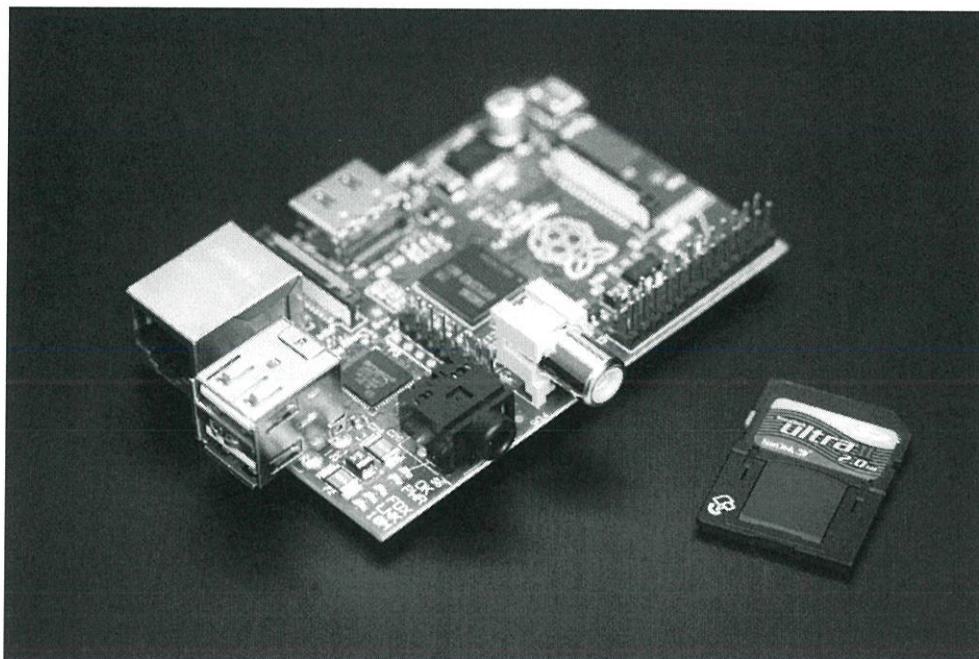
#### 2.1 Raspberry Pi

บอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับ TV และอุปกรณ์ เม้าส์ คีย์บอร์ดได้ ซึ่งเราสามารถนำมาประยุกต์ใช้ในการทำโครงการทางด้านอิเล็กทรอนิกส์, การเขียนโปรแกรม หรือเป็นเครื่องคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก ไม่ว่าจะทำเป็น Speedsheet ,Word processing, ท่องอินเทอร์เน็ต,ส่งอีเมล หรือเล่นเกม เป็นต้น อีกทั้งยังสามารถเล่นไฟล์วิดีโอความละเอียดสูง (High-Definnition) ได้อีกด้วย

บอร์ด Raspberry Pi นี้สามารถรองรับระบบปฏิบัติการ(Operating System) ได้หลายระบบ เช่น Debian ,Fedora และ Arch เป็นต้น ซึ่งจะถูกรวบรวมลงบน SD card และบอร์ด Raspberry Pi นี้ถูกออกแบบให้มี CPU และ GPU และ RAM อยู่ในชิพตัวเดียวกัน และมีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ ได้อีกด้วย

#### คุณสมบัติทางเทคนิคของบอร์ด Raspberry Pi

SoC (system on a chip)	: Broadcom BCM2835
CPU	: 700 MHz ARM1176JZF-S core (ARM11 family, ARMv6 instruction set)
GPU	: Broadcom VideoCore IV @ 250 MHz, OpenGL Es 2.0, MPEG-2 1080p 30 h.264/MPEG-4AVC high-profile decoder and encoder
Memory (SDRAM)	: 512 MB (Shared with GPU)
USB 2.0 Ports	: 2
Video Input	: Camera Module
Video Output	: Composition RCA ,HDMI
Audio Output	: 3.5 mm jack, HDMI
Onboard storage	: SD/MMC/SDIO card slot
Onboard network	: 10/100 Ethernet usb adapter on the third port of the USB hub
Low-level peripherals	: 8 x GPIO , UART , I2C Bus, SPI Bus, +3.3V , +5V,Ground
Power rating	: 700 mA (3.5 W)
Power source	: 5V
Size	: 85.6 mm x 53 Mm
Weight	: 45 g



รูปที่ 2.1 Raspberry Pi

## ตำแหน่งของขา GPIO

wiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	wiringPi Pin
–	–	3.3v	1 2	5v	–	–
8	R1:0/R2:2	SDA0	3 4	DNC	–	–
9	R1:1/R2:3	SCL0	5 6	0v	–	–
7	4	GPIO7	7 8	GPIO14	14	15
–	–	DNC	9 10	GPIO15	15	16
0	17	GPIO0	11 12	GPIO18	18	1
2	R1:21/R2:27	GPIO2	13 14	DNC	–	–
3	22	GPIO3	15 16	GPIO23	23	4
–	–	DNC	17 18	GPIO24	24	5
12	10	MOSI	19 20	DNC	–	–
13	9	MISO	21 22	GPIO25	25	6
14	11	SCLK	23 24	CE0	8	10
–	–	DNC	25 26	CE1	7	11
wiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	wiringPi Pin

รูปที่ 2.2 ตำแหน่งขา GPIO ของ Raspberry Pi

## 2.2 Arduino

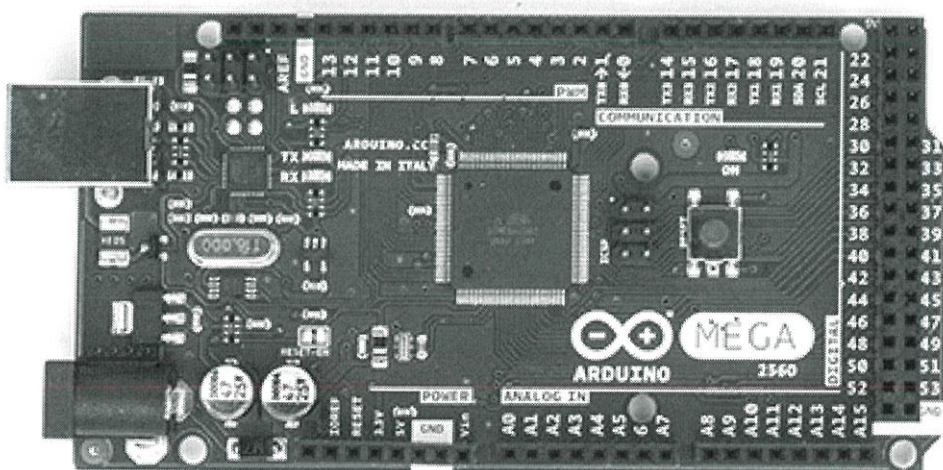
คือ เครื่องมือแพลตฟอร์มเปิดที่ใช้ในการควบคุมสิ่งต่างๆ โดยใช้ไมโครคอนโทรลเลอร์แบบง่ายๆ ในการเขียนโปรแกรมควบคุมโดยการเขียนโปรแกรมผ่านแพลตฟอร์ม Arduino เข้าสู่บอร์ดควบคุม

Arduino สามารถพัฒนาให้สื่อสารกับวัตถุอื่นๆ สามารถรับข้อมูลจากสวิทช์หรือว่าจะเป็นข้อมูลจากเซนเซอร์ หรือว่าใช้ในการควบคุมแสงไฟ มอเตอร์ และอุปกรณ์อิเล็กทรอนิกส์อื่นๆ อีกมาก โปรแกรมที่พัฒนาโดย Arduino สามารถทำงานแยกโดดเดี่ยวออกมาได้หรือว่าจะสื่อสารหากันก็ทำได้

ภาษาที่ใช้ใน Arduino จะใช้ภาษา c++ ในการเขียนโปรแกรมควบคุม ซึ่งเหมาะสมกับผู้เริ่มต้นเขียนโปรแกรมเพราะเป็นภาษาเริ่มต้นที่ง่ายในการศึกษา

### คุณสมบัติทางเทคนิคของบอร์ด Arduino Mega2560

Microcontroller	: ATmega2560
Operating Voltage	: 5V
Input Voltage (recommended)	: 7-12V
Input Voltage (limits)	: 6-20V
Digital I/O Pins	: 54 (of which 15 provide PWM output)
Analog Input Pins	: 16
DC Current per I/O Pin	: 40 mA
DC Current for 3.3V Pin	: 50 mA
Flash Memory	: 256 KB of which 8 KB used by bootloader
SRAM	: 8 KB
EEPROM	: 4 KB
Clock Speed	: 16 MHz



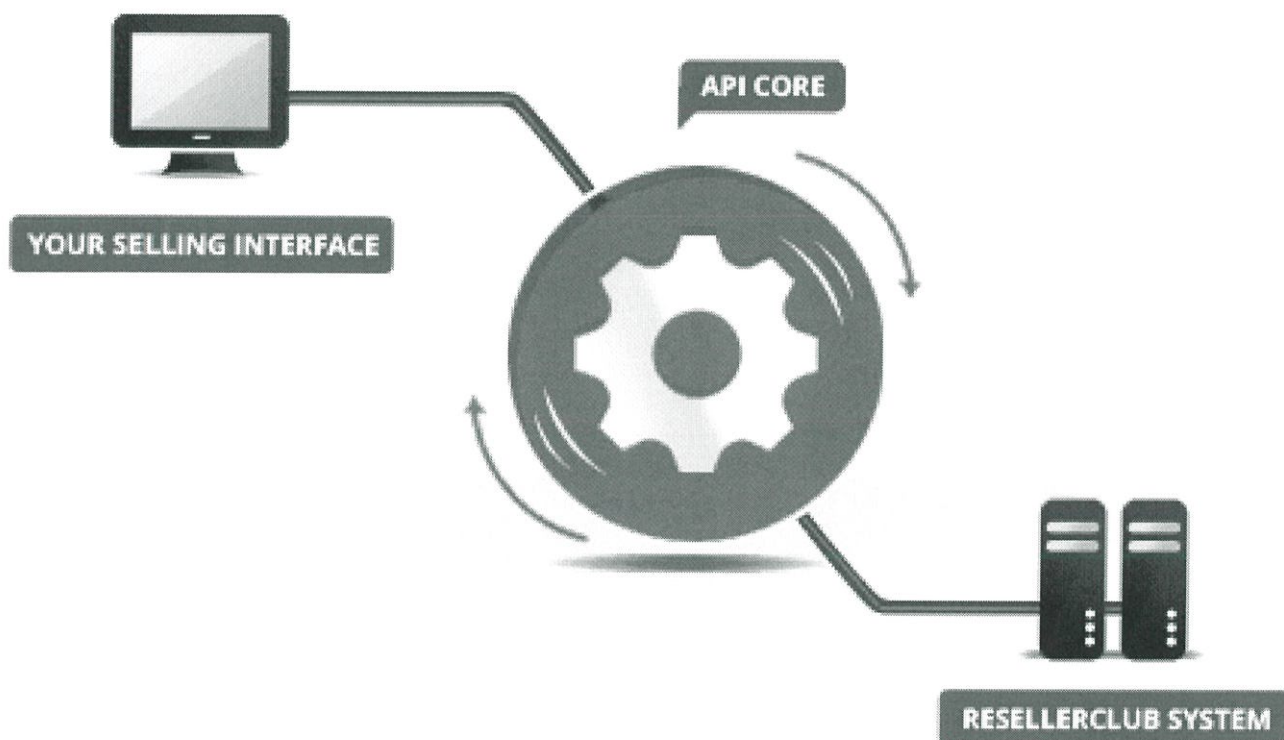
รูปที่ 2.3 บอร์ด Arduino Mega2560

### 2.3 API (Application Programming Interface)

คือ ช่องทางการเชื่อมต่อช่องทางหนึ่งที่จะเชื่อมต่อกับเว็บไซต์ผู้ให้บริการ API จากที่อื่นเป็นตัวกลางที่ทำให้โปรแกรมประยุกต์เชื่อมต่อกับโปรแกรมประยุกต์อื่นๆ หรือเชื่อมต่อกับระบบการ หรือเป็นกลุ่มฟังก์ชัน ขั้นตอน หรือ คลาส ที่ระบบปฏิบัติการ หรือผู้ให้บริการสร้างขึ้นมาเพื่อรองรับการเรียกข้อมูลจากโปรแกรมอื่นๆ ทั้งนี้ API สามารถใช้งานได้กับภาษาที่เขียนโปรแกรมรองรับเท่านั้นซึ่งมักจะถูกทำให้อยู่ในรูปแบบ syntax หรือ element ที่สามารถนำไปใช้ได้อย่างสะดวกสบาย

ประโยชน์ของ API

- สามารถเขียนโปรแกรมรับส่งข้อมูลข้าม Server ได้
- ไม่จำเป็นต้องเข้าหน้าเว็บหลัก ก็มีข้อมูลของเว็บหลัก จากเว็บที่ตั้ง API



รูปที่ 2.4 ตัวอย่างการทำงานของ API

## 2.4 การวิเคราะห์เสียง(Speech recognition)

คือ การเปลี่ยนจากเสียงพูด คำพูด ต่างๆ ออกมาเป็นตัวหนังสือ ในบางระบบของการวิเคราะห์เสียงพูดจะใช้วิธีการสอนให้รู้จักคำต่างๆโดยการให้ผู้ทดลองพูดตัวหนังสือและให้ระบบฟัง แล้ววิเคราะห์คำต่างๆ ซึ่งในแต่ละคนจะมีโทนเสียงเป็นของตัวเอง ปัจจุบันมีโปรแกรมวิเคราะห์เสียงให้ใช้กันอย่างแพร่หลาย เช่น siri ของบริษัทแอปเปิล หรือว่าจะเป็น cortana ของบริษัทไมโครซอฟ

### 1.) CMUSphinx

เป็นโปรแกรม automatic speech recognition ที่เป็น open source พัฒนาโดยทีมนักวิจัยจาก Carnegie Mellon University สามารถรู้จำเสียงได้หลายภาษา แต่ถ้าหากจะให้รู้จำภาษาอื่นๆ ที่ไม่มีในฐานข้อมูลของโปรแกรมเราก็สามารถให้คอมพิวเตอร์จำเสียงของเราเพิ่มเติมได้ผ่านการอ่านข้อความให้ฟัง เทคนิคนี้จะเรียกกันว่า การฝึกภาษา (training) ปัจจุบัน CMUSphinx มีให้ใช้ด้วยกัน 2 แบบ

1. Sphinx4 เป็น speech recognition ตัวล่าสุด มีความยืดหยุ่นสูง ปรับแต่งได้ง่ายเขียนด้วยภาษา Java

2. PocketSphinx เป็น speech recognition ที่พัฒนาให้มีการประมวลผลเร็วและใช้หน่วยความจำน้อยเหมาะสำหรับใช้ใน embedded device ถูกพัฒนาด้วยภาษา C

### 2.) กระบวนการเรียนรู้จดจำเสียง

กระบวนการรู้จำเสียงของ CMUSphinx สามารถอธิบายได้คร่าวๆ ดังนี้ หลังจากที่ได้รับข้อมูลคลื่นเสียงมาแล้ว Sphinx จะทำการแยกออกเป็น utterances โดยใช้เสียงเงียบ (silence) เป็นตัวแบ่งและจะพยายามแปลคำว่าเสียงที่เปล่งออกมานั้นคือคำว่าอะไร (เรียกว่าการรู้จำหรือ recognition) การที่จะรู้จำได้นั้น โปรแกรมจะต้องเช็คคำทุกคำที่เป็นไปได้แล้วเอาแต่ละคำมาเทียบเสียงกับข้อมูลเสียงที่รับเข้ามาว่าตรงกันหรือไม่ ซึ่งการที่จะให้โปรแกรมเทียบเสียงได้นั้นต้องมีข้อมูลทางด้านโครงสร้างทางภาษา หรือข้อมูลสำหรับเสียงที่โปรแกรมใช้ช่วยในการตัดสินใจในการรู้จำข้อมูล เราเรียกกันว่า “โมเดล”

### 3.) โมเดลที่ใช้ในกระบวนการรู้จำเสียงของ CMUSphinx

วิธีการหนึ่งที่จะทำให้โปรแกรมสามารถรู้จำเสียงได้นั้นจะต้องอาศัยโมเดลในการช่วยประมวลผลและตัดสินใจว่าเสียงที่เราพูดออกมาควรจะเป็นคำว่าอะไร โมเดลหลักๆ ที่จะต้องใช้ใน CMUSphinx มีอยู่ด้วยกัน 3 ตัว

1. Acoustic Model เป็นโมเดลที่เก็บคุณสมบัติต่างๆ ของเสียง

2. Phonetic Dictionary เป็นดิกชันนารีเสียงของคอมพิวเตอร์ คอมพิวเตอร์จะใช้ดิกชันนารีอันนี้เพื่อเช็กว่าคำนั้นควรจะออกเสียงอย่างไร

3. Language Model เป็นโมเดลที่บอกว่าหากผู้ใช้พูดคำนี้แล้วคำต่อไปน่าจะเป็นอะไรได้บ้าง ตัวอย่างเช่น ผมพูดคำว่า “หิว” คำต่อไปก็ควรจะเป็น “ข้าว” หรือ “น้ำ” มากกว่า “โต๊ะ” เป็นต้น โมเดลนี้จะช่วยลดจำนวนของคำที่เป็นไปได้ให้คอมพิวเตอร์ ทำให้คอมพิวเตอร์ใช้เวลาค้นหาเพื่อเปรียบเทียบเสียงน้อยลง

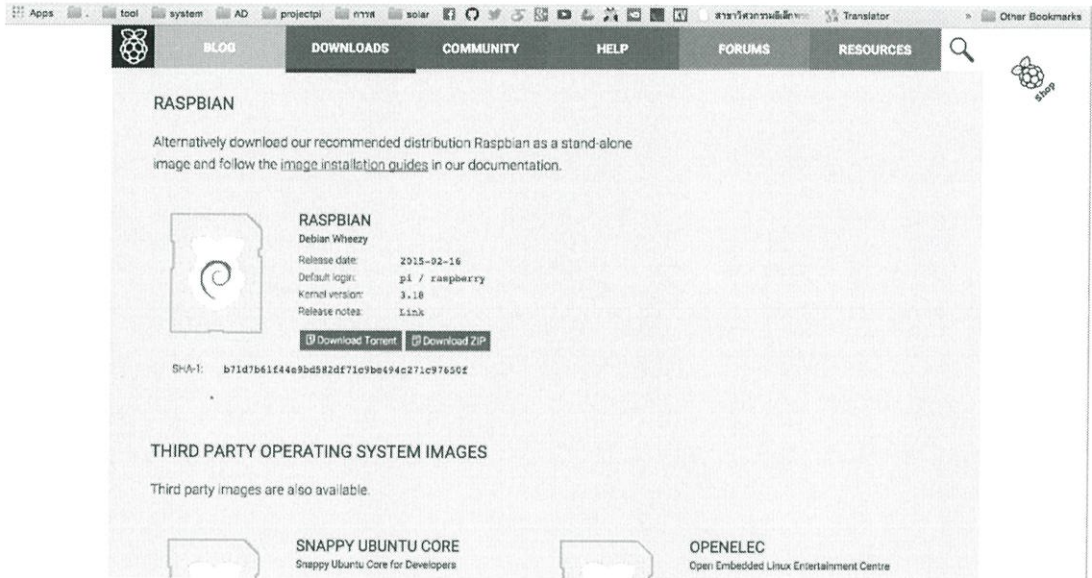
## บทที่ 3

### การออกแบบและกระบวนการต่างๆ

#### 3.1 หน่วยประมวลผล (Raspberrypi)

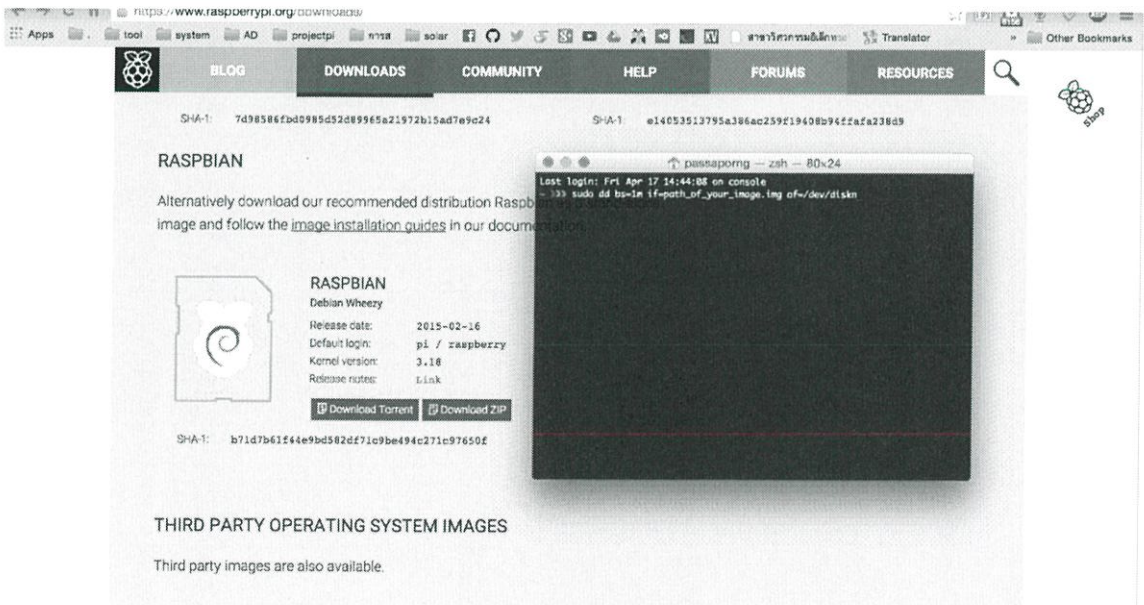
##### 3.1.1 ติดตั้งระบบปฏิบัติการ Raspbian

ดาวน์โหลดไฟล์ระบบปฏิบัติการ Raspbian ได้ที่เว็บ <https://www.raspberrypi.org/downloads/>



รูปที่ 3.1.1 เว็บไซต์ดาวน์โหลดระบบปฏิบัติการ Raspbian

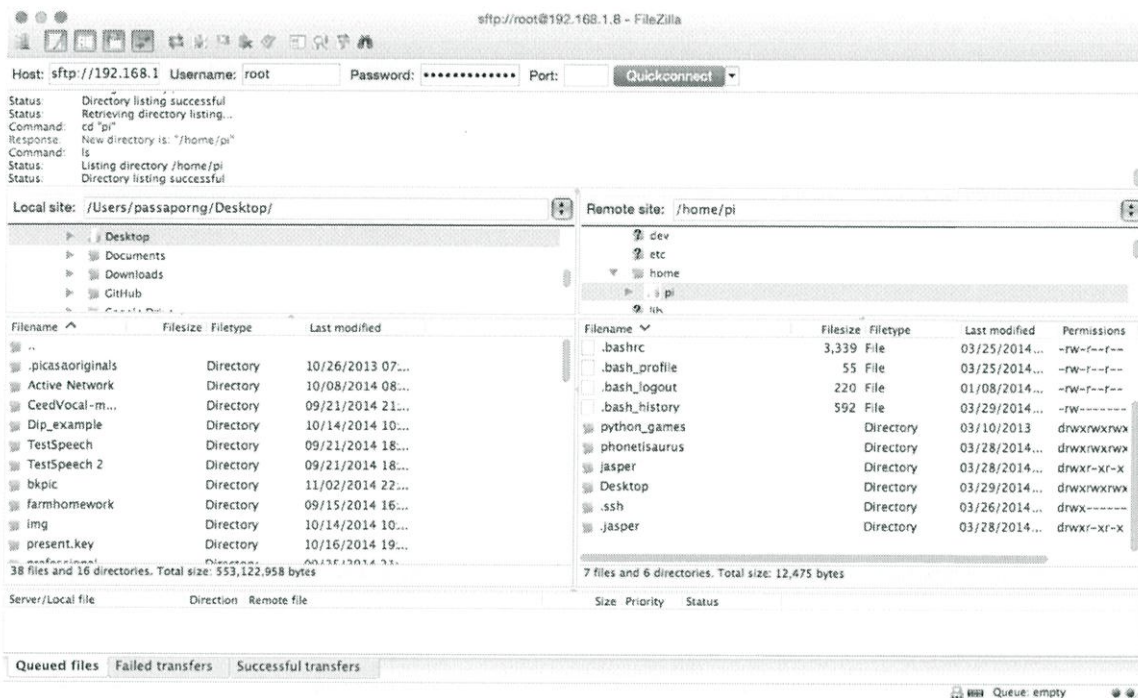
ใส่ SD card เข้าเครื่องคอมพิวเตอร์ ใช้ คำสั่ง dd เพื่อลงไฟล์ ใส่ SD card



รูปที่ 3.1.2 ติดตั้งระบบปฏิบัติการลงแผ่น SDcard

### 3.1.2 ติดตั้งโปรแกรมและสร้างคำสั่ง wit.ai

ดาวน์โหลดโปรแกรมจาก <https://github.com/wit-ai/witd/blob/master/witd-arm?raw=true> นำมาใส่ในโฟลเดอร์ไพ โดยใช้โปรแกรม Filezilla



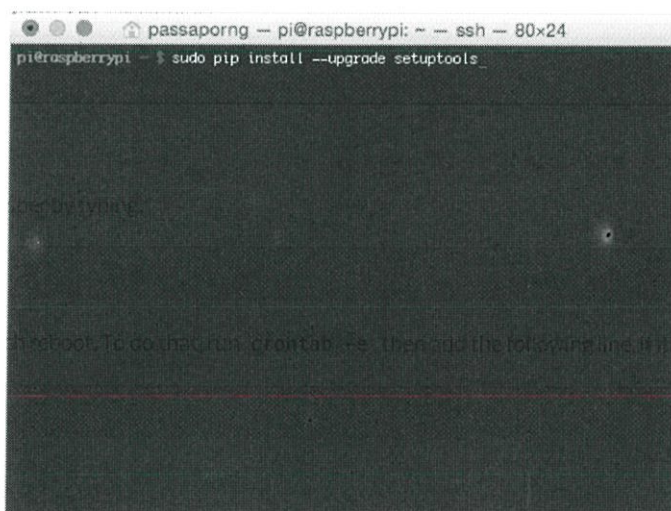
รูปที่ 3.1.3 ย้ายไฟล์คอมไพล์ wit-arm ไปไว้ใน RaspberryPi

ติดตั้งไลบรารีโปรแกรม ด้วยคำสั่ง

```
sudo apt-get install libcurl3
```

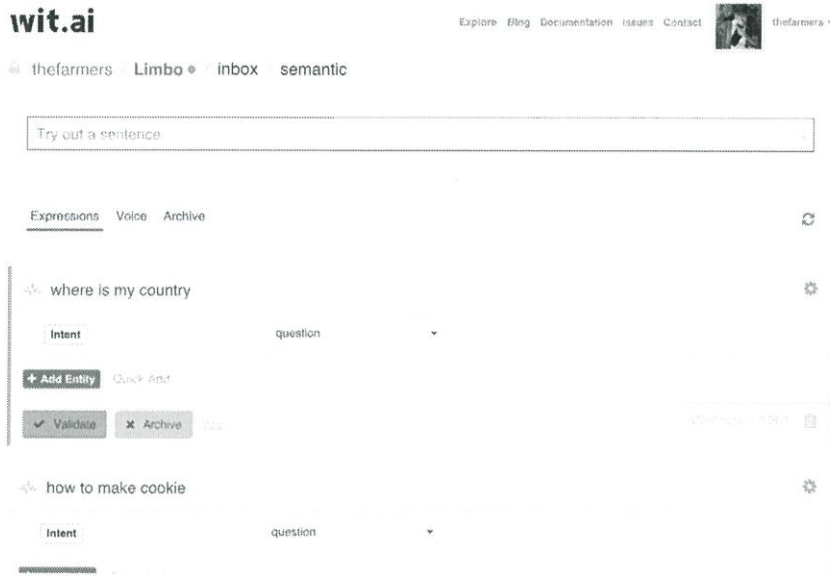
```
sudo apt-get install libsox2
```

```
sudo apt-get install alsa-utils
```



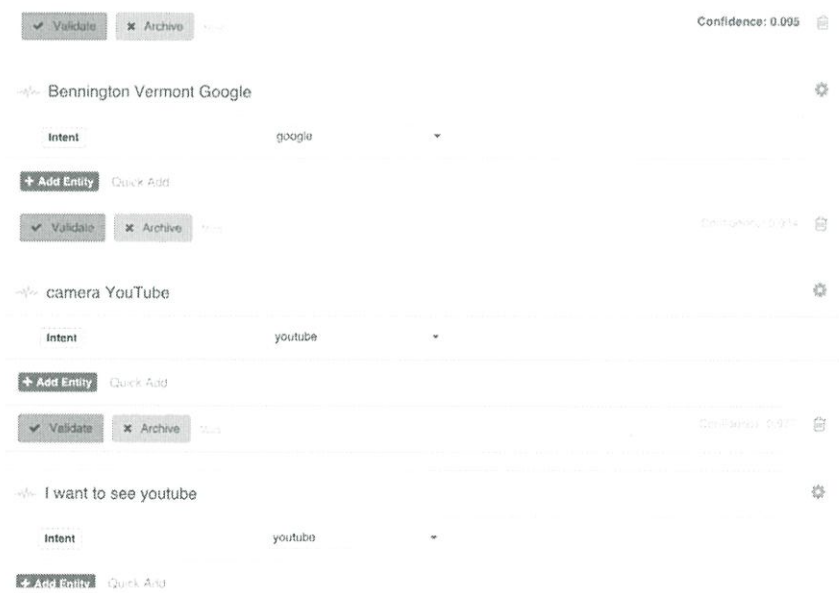
รูปที่ 3.1.4 ติดตั้งไลบรารีโปรแกรม

สมัครสมาชิก wit.ai เพื่อเราจะได้ API key มาใช้ และเราก็จะเริ่มสอนคำพูดต่างๆ



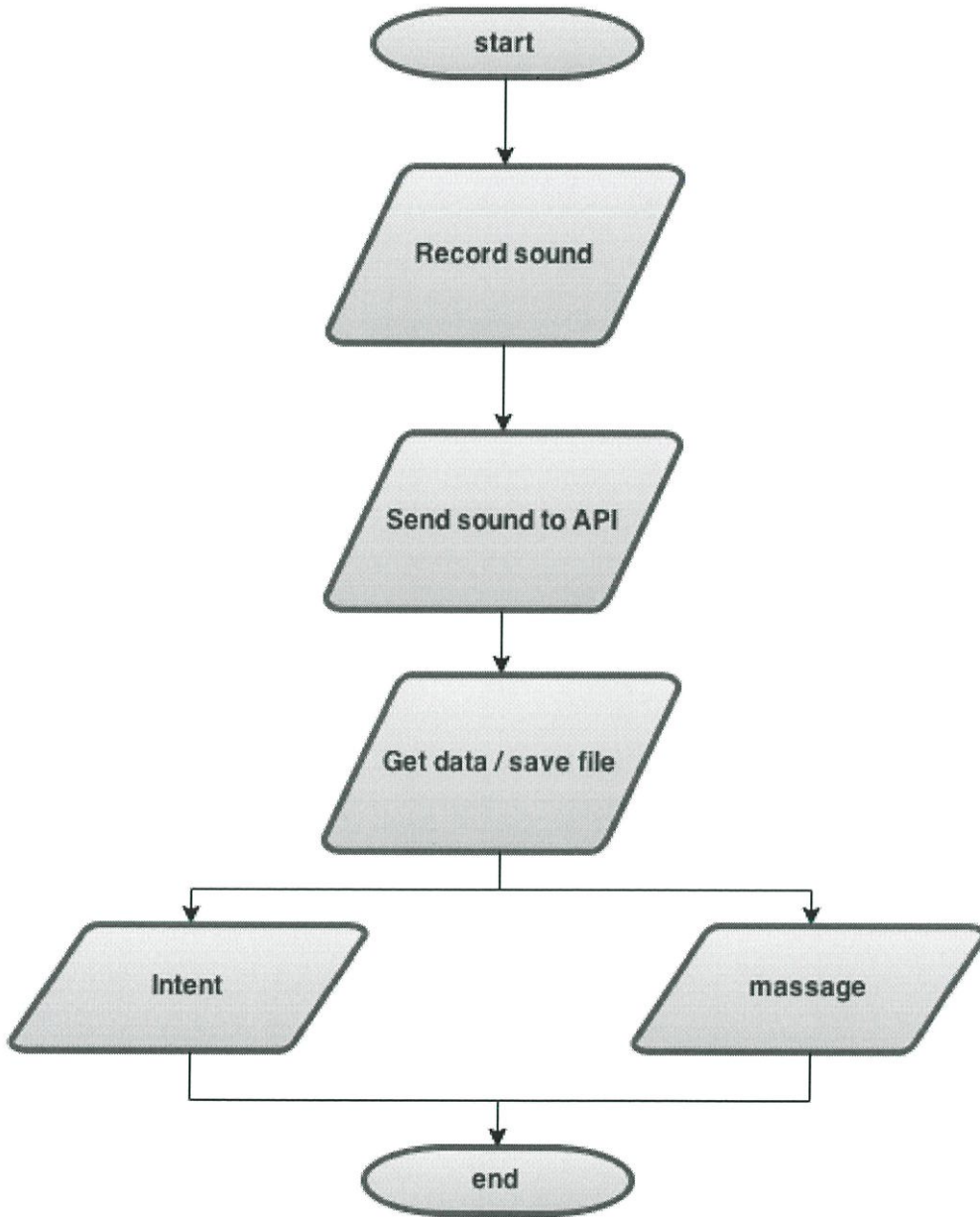
รูปที่ 3.1.5 หน้าต่างการจัดการ wit.ai

สอนการคิดให้กับหุ่นยนต์



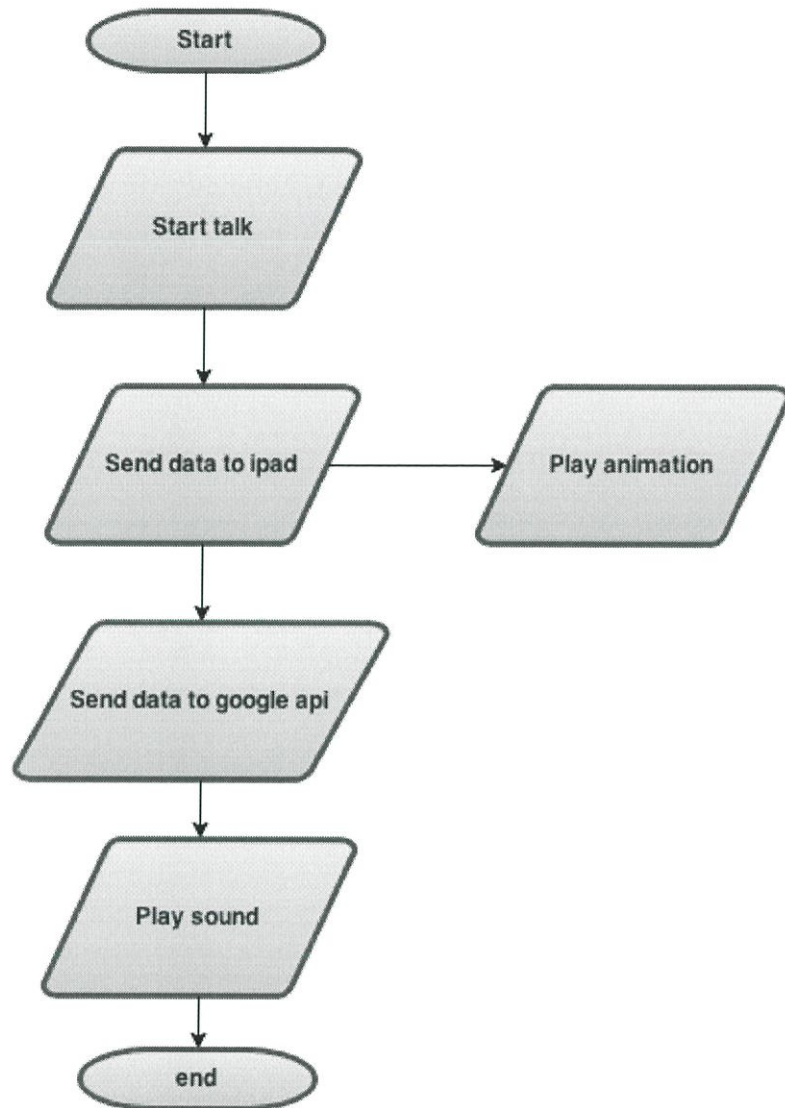
รูปที่ 3.1.6 หน้าต่างคำพูดต่างๆที่สอน

## 3.1.3 ออกแบบโปรแกรมเปลี่ยนจากเสียงเป็นตัวหนังสือผ่าน api ของ wit.ai



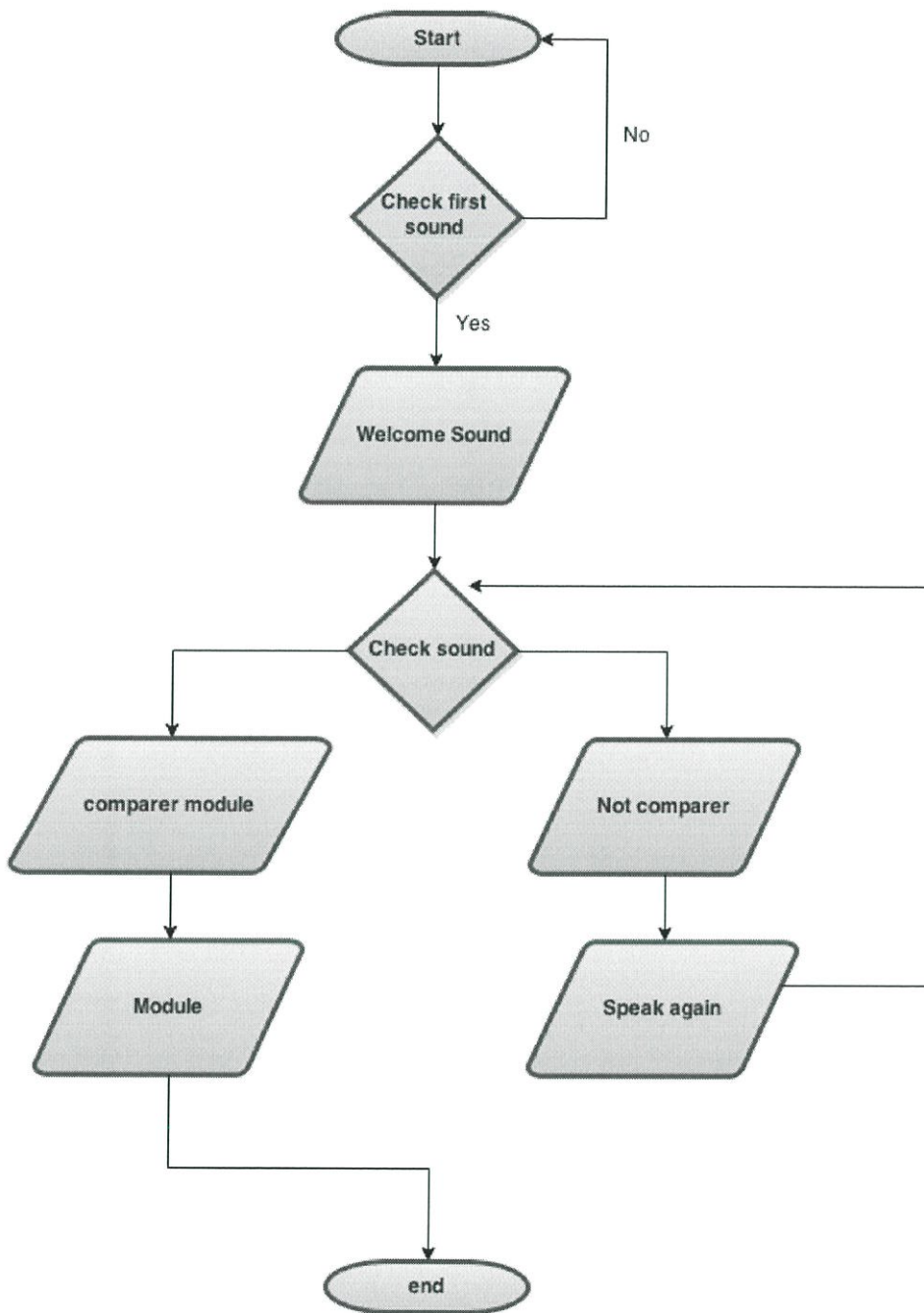
รูปที่ 3.1.7 หลักการทำงานเปลี่ยนจากเสียงเป็นตัวหนังสือ

## 3.1.4 ออกแบบโปรแกรมเปลี่ยนจากตัวหนังสือเป็นเสียงพูดด้วย api ของ google



รูปที่ 3.1.7 หลักการทำงานเปลี่ยนจากตัวหนังสือเป็นเสียง

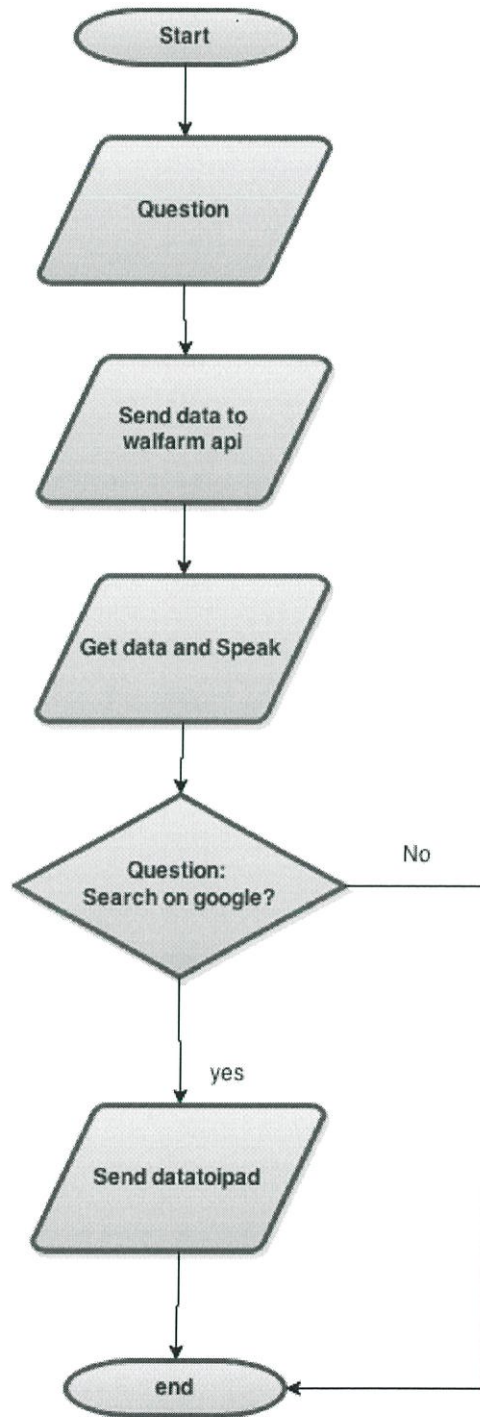
## 3.1.5 ออกแบบโปรแกรมแกนหลักของการคิด



รูปที่ 3.1.8 ออกแบบโปรแกรมแกนหลักความคิด

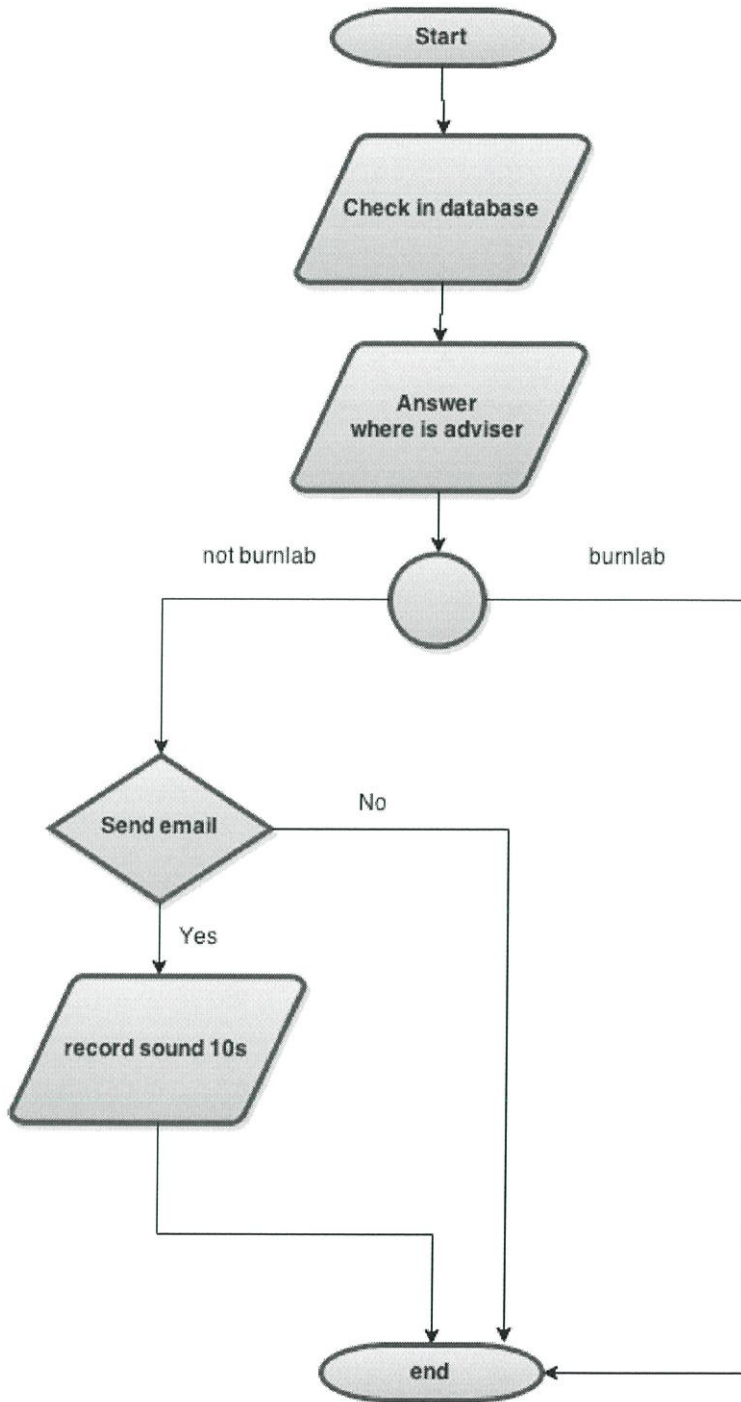
### 3.1.6 เขียนโปรแกรมส่วนเสริมต่างๆ

1) โมดูลการค้นหาคำตอบด้วย wolfram และ google



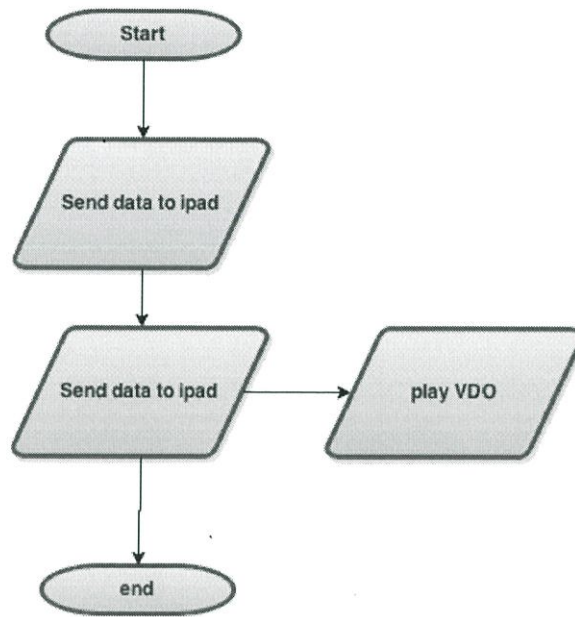
รูปที่ 3.1.9 ออกแบบการค้นหาคำตอบด้วย wolfram and google

## 2) โมดูลผู้ช่วย



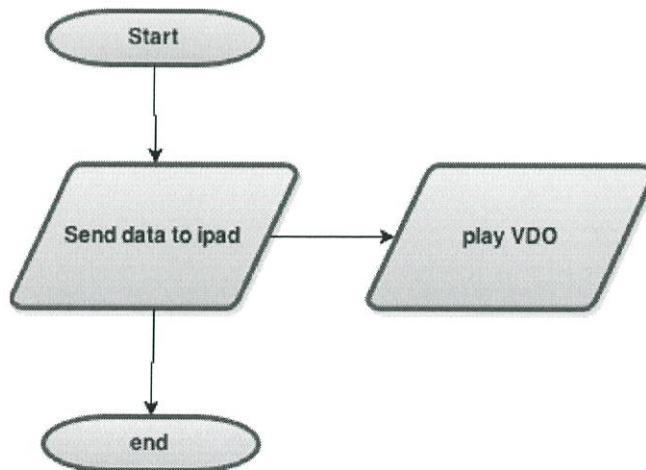
รูปที่ 3.1.10 ออกแบบระบบผู้ช่วย

## 3.1.5.3 โมดูลเปิด youtube



รูปที่ 3.1.11 ออกแบบระบบเปิด youtube

## 3.1.5.4 โมดูลการแนะนำห้อง lab



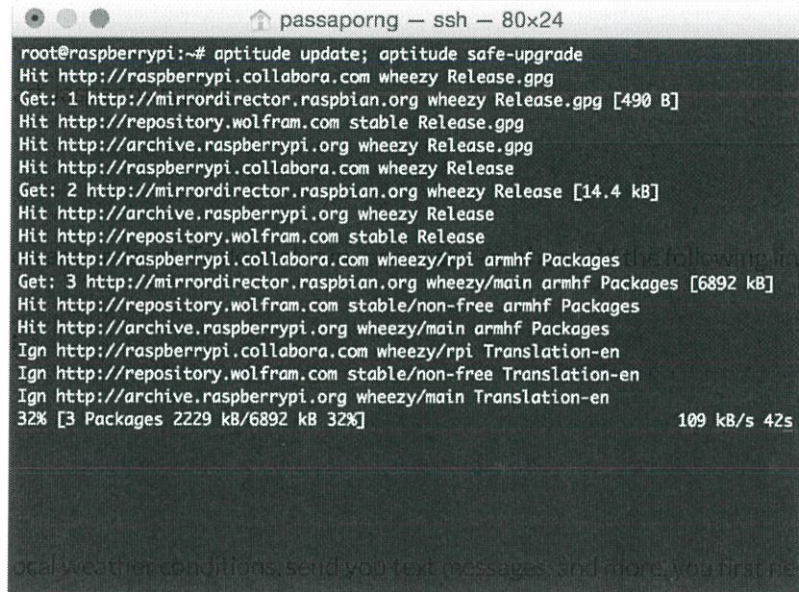
รูปที่ 3.1.12 ออกแบบระบบแนะนำห้อง lab

### 3.1.7 ติดตั้งโปรแกรมที่ใช้ในการเชื่อมต่อข้อมูลระหว่าง RaspberryPi และ Device อื่นๆ

#### 3.1.6.1 ติดต่อกับ iPad

ในที่นี้เราจะติดต่อกับ iPad ผ่าน Wifi ส่งข้อมูลโดยผ่านโปรโตคอล Http

ติดตั้งโปรแกรมเพื่อใช้ในการสร้าง Wifi Accesspoint ออฟเกรดโปรแกรม aptitude ใช้คำสั่ง  
 aptitude update; aptitude safe-upgrade



```

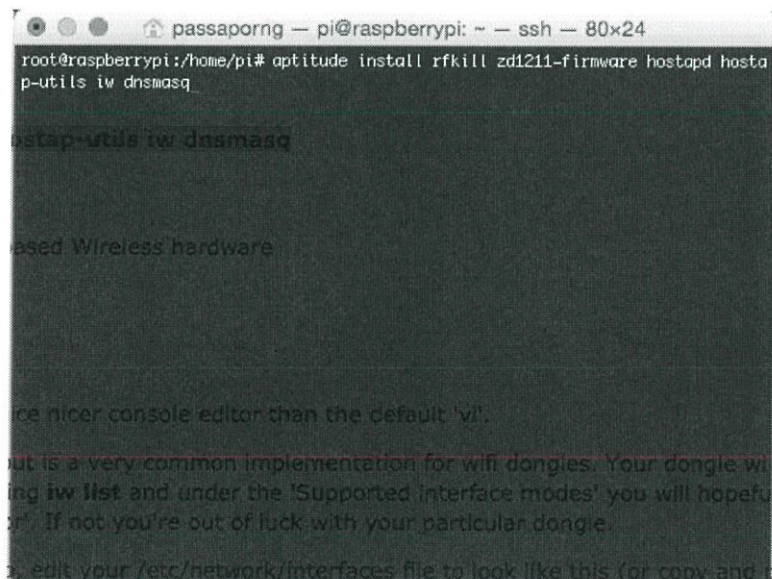
passaporng — ssh — 80x24
root@raspberrypi:~# aptitude update; aptitude safe-upgrade
Hit http://raspberrypi.collabora.com wheezy Release.gpg
Get: 1 http://mirrordirector.raspbian.org wheezy Release.gpg [490 B]
Hit http://repository.wolfram.com stable Release.gpg
Hit http://archive.raspberrypi.org wheezy Release.gpg
Hit http://raspberrypi.collabora.com wheezy Release
Get: 2 http://mirrordirector.raspbian.org wheezy Release [14.4 kB]
Hit http://archive.raspberrypi.org wheezy Release
Hit http://repository.wolfram.com stable Release
Hit http://raspberrypi.collabora.com wheezy/rpi armhf Packages
Get: 3 http://mirrordirector.raspbian.org wheezy/main armhf Packages [6892 kB]
Hit http://repository.wolfram.com stable/non-free armhf Packages
Hit http://archive.raspberrypi.org wheezy/main armhf Packages
Ign http://raspberrypi.collabora.com wheezy/rpi Translation-en
Ign http://repository.wolfram.com stable/non-free Translation-en
Ign http://archive.raspberrypi.org wheezy/main Translation-en
32% [3 Packages 2229 kB/6892 kB 32%] 109 kB/s 42s

```

รูปที่ 3.1.13 ออฟเกรดโปรแกรม

ติดตั้ง firmware ของ usb wifi dongle ใช้คำสั่ง

aptitude install rkill zd1211-firmware hostapd hostap-utils iw dnsmasq



```

passaporng — pi@raspberrypi: ~ — ssh — 80x24
root@raspberrypi:/home/pi# aptitude install rkill zd1211-firmware hostapd hostap-
p-utils iw dnsmasq
...
based Wireless hardware
...
ice nicer console editor than the default 'vi'.
...
ut is a very common implementation for wifi dongles. Your dongle will
ing iw list and under the 'Supported interface modes' you will hopefully
...
If not you're out of luck with your particular dongle.
...
edit your /etc/network/interfaces file to look like this (or copy and p

```

รูปที่ 3.1.14 ติดตั้ง firmwear

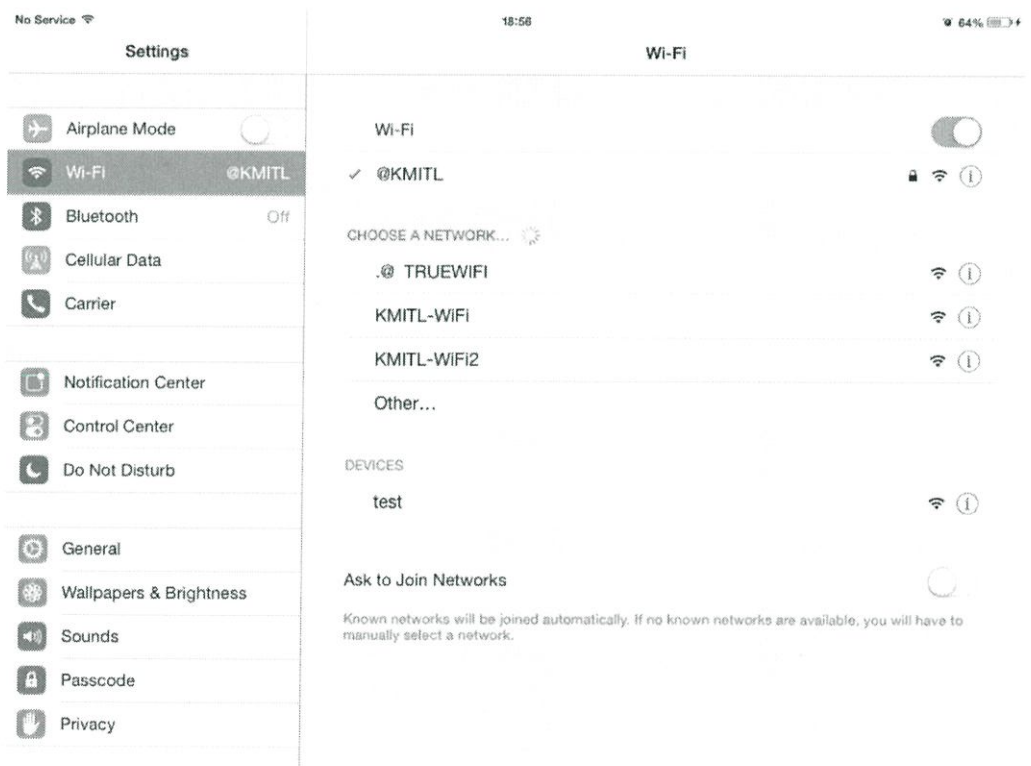
กำหนดค่าต่างๆให้อยู่ในวง Burn Lab network โดยไปแก้ไขไฟล์  
/etc/network/interfaces

```

GNU nano 2.2.6 File: /etc/network/interfaces
auto lo
iface lo inet loopback
iface eth0 inet dhcp
iface wlan0 inet static
address 192.168.2.1
netmask 255.255.255.0
wireless-mode ad-hoc
wireless-channel 1
wireless-ssid test
auto wlan0
  
```

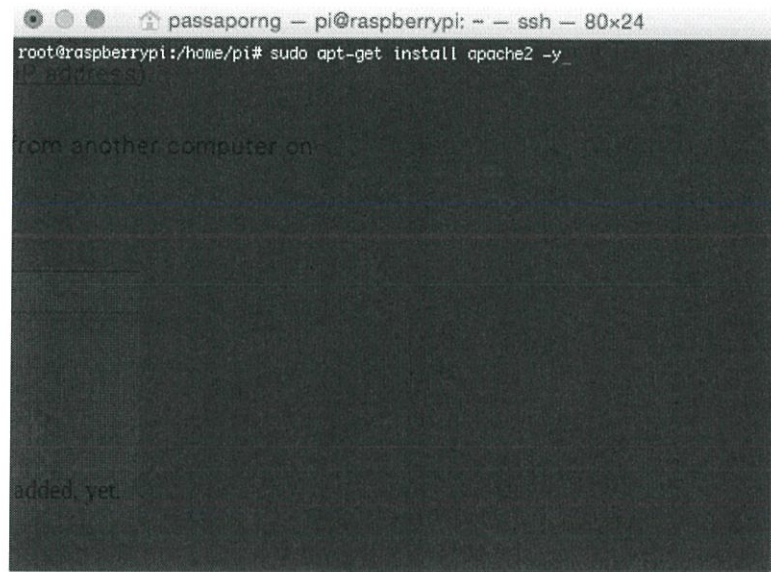
รูปที่ 3.1.15 ตั้งค่า LAN

ผลลัพธ์ มีให้กดเชื่อมต่อกับ iPad



รูปที่ 3.1.16 การเชื่อมต่อกับ Ipad

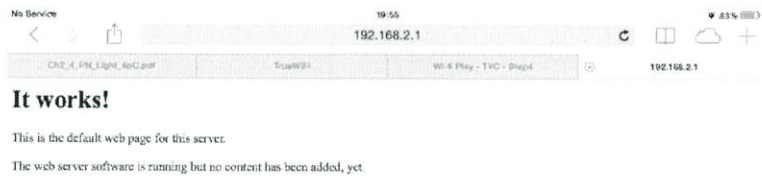
ลง webserver เพื่อส่งข้อมูลผ่านโปรโตคอล http โดยใช้คำสั่ง  
sudo apt-get install apache2 -y



```
passaporng — pi@raspberrypi: ~ — ssh — 80x24
root@raspberrypi:/home/pi# sudo apt-get install apache2 -y
[...]
```

รูปที่ 3.1.17 ติดตั้ง webserver

ผลลัพธ์บนบราวเซอร์

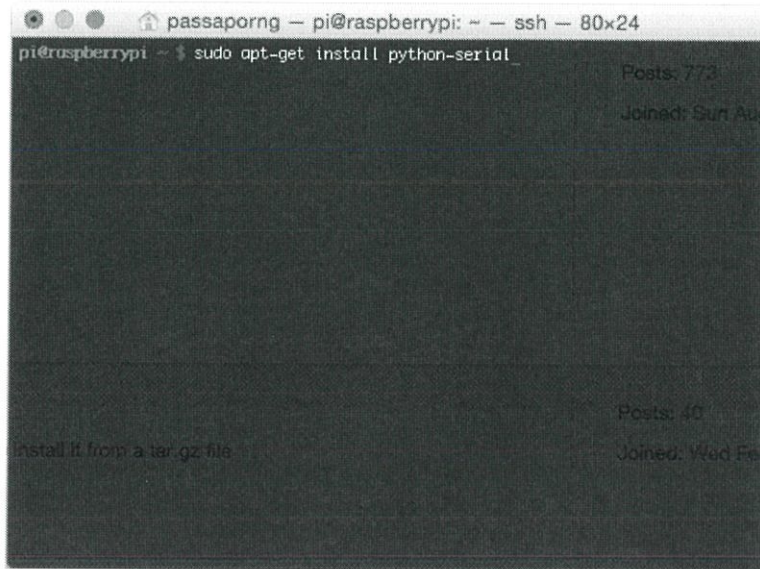


รูปที่ 3.1.18 ผลลัพธ์

### 3.1.6.2 ติดต่อกับ Arduino

ติดตั้งไลบรารีเพื่อใช้ในการติดต่อกับอูยโน่ ด้วยคำสั่ง

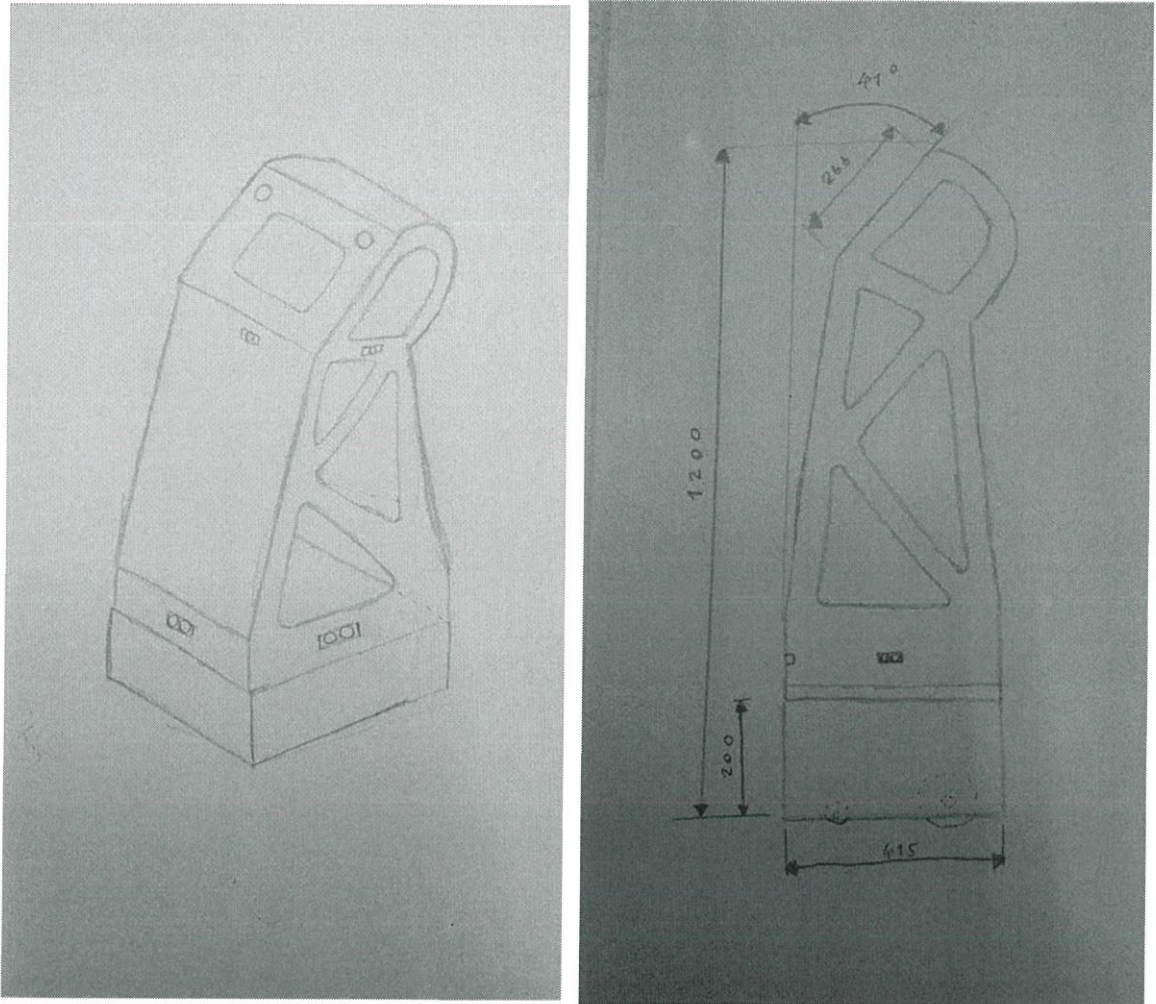
```
sudo apt-get install python-serial
```

A terminal window screenshot showing the command 'sudo apt-get install python-serial' being executed on a Raspberry Pi. The terminal output is mostly black, indicating the command is running. The window title bar shows 'passaporng - pi@raspberrypi: ~ - ssh - 80x24'. There are some faint, illegible text elements on the right side of the terminal window, possibly from a background application or a watermark.

รูปที่ 3.1.19 ติดตั้งไลบรารีที่ใช้เชื่อมต่อกับ Arduino

### 3.2 ส่วนของการขับเคลื่อนหุ่นยนต์

#### 3.2.1 ออกแบบโครงหุ่นยนต์

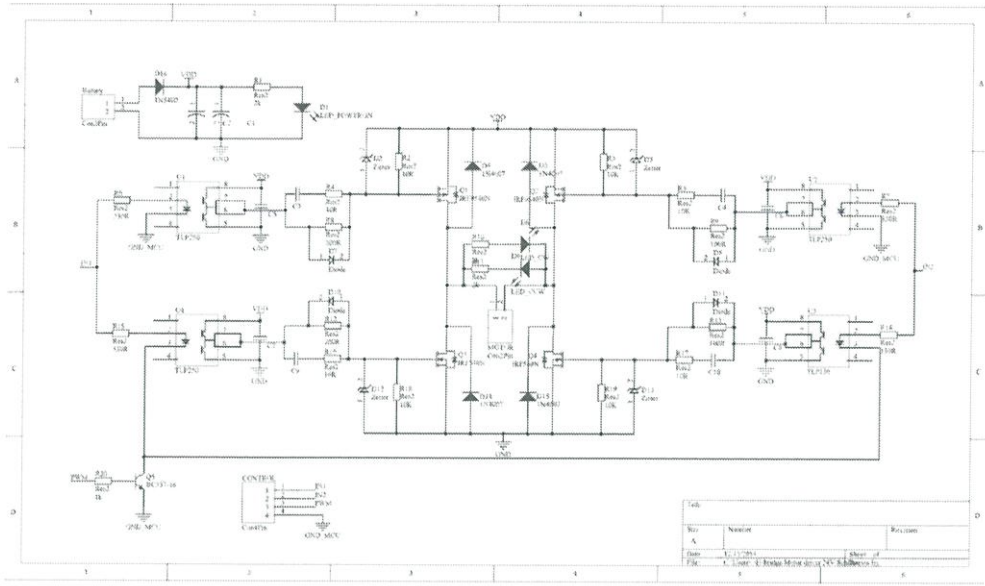


รูปที่ 3.2.1 ออกแบบหุ่นยนต์

จากภาพที่ 3.2.1 เป็นการออกแบบโครงสร้างของตัวหุ่นยนต์โดยการสเก็ตด้วยมือก่อนที่จะนำไปวาดลงในโปรแกรมสำหรับการออกแบบเพื่อนำไปขึ้นรูปโครงสร้างจริง

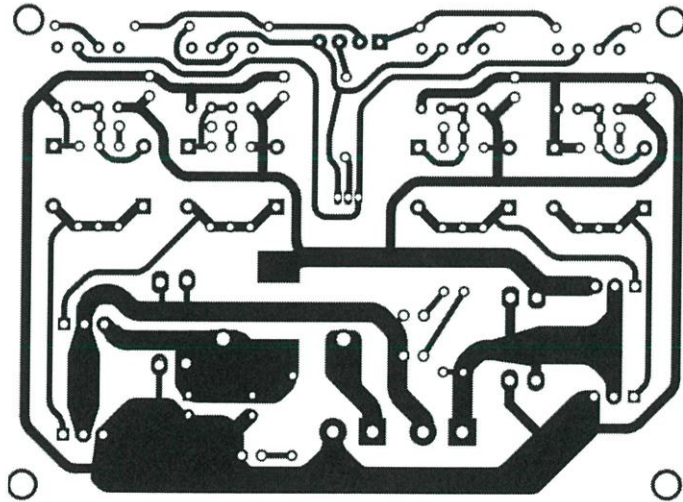
3.2.2 สร้างบอร์ดขับมอเตอร์

1) schematic



รูปที่ 3.2.2 schematic ของวงจรขับมอเตอร์

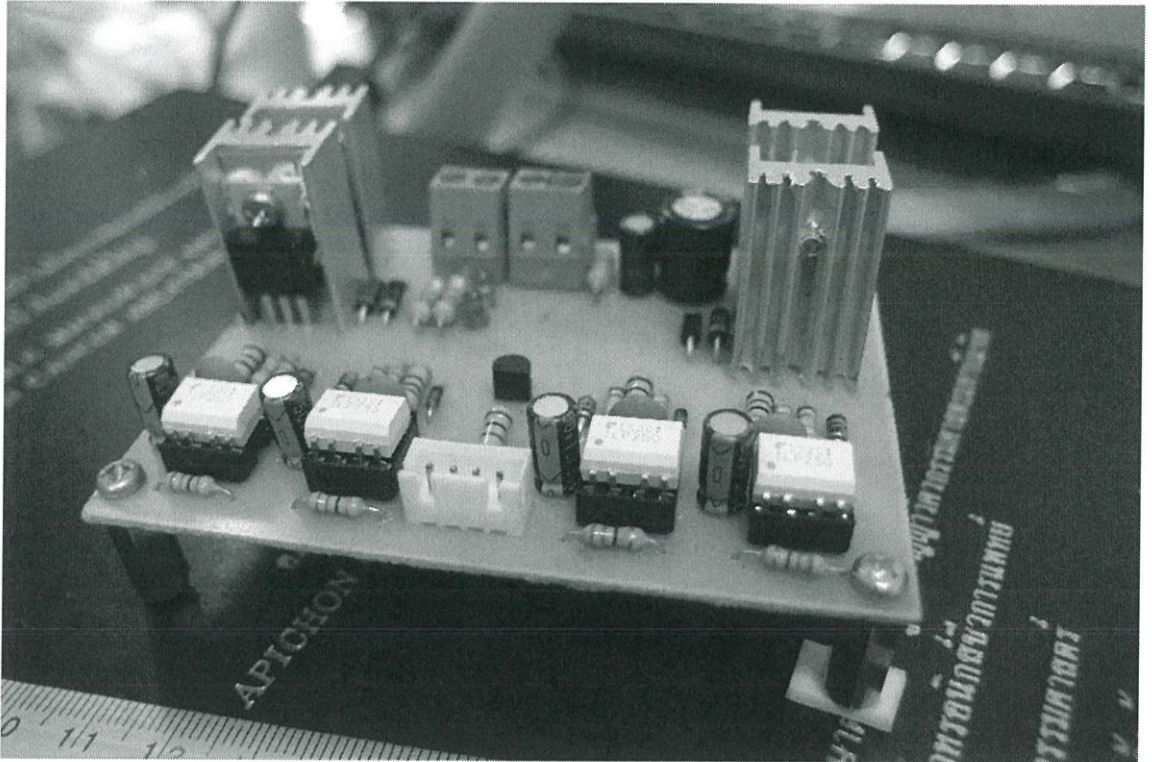
2) PCB



รูปที่ 3.2.3 PCB ของวงจรขับมอเตอร์

ในการออกแบบวงจรเราจะใช้ในแบบของวงจร H-Bridge โดยในวงจรเราจะควบคุมการทำงานโดยการใช้ MOSFET เนื่องจากตัว MOSFET มีการทนกระแสได้สูง แรงดันตกคร่อมน้อยกว่า และความเร็วในการสวิตช์ เร็วกว่า ทรานซิสเตอร์ จึงเลือกใช้ MOSFET โดยการ drive จะใช้ MOSFET ชนิด pnp กับ npn ในการสั่ง high side เราจะใช้ ชนิด pnp แต่ในการสั่ง Low side เราจะใช้ชนิด npn

## 3) ลงอุปกรณ์ต่างๆ



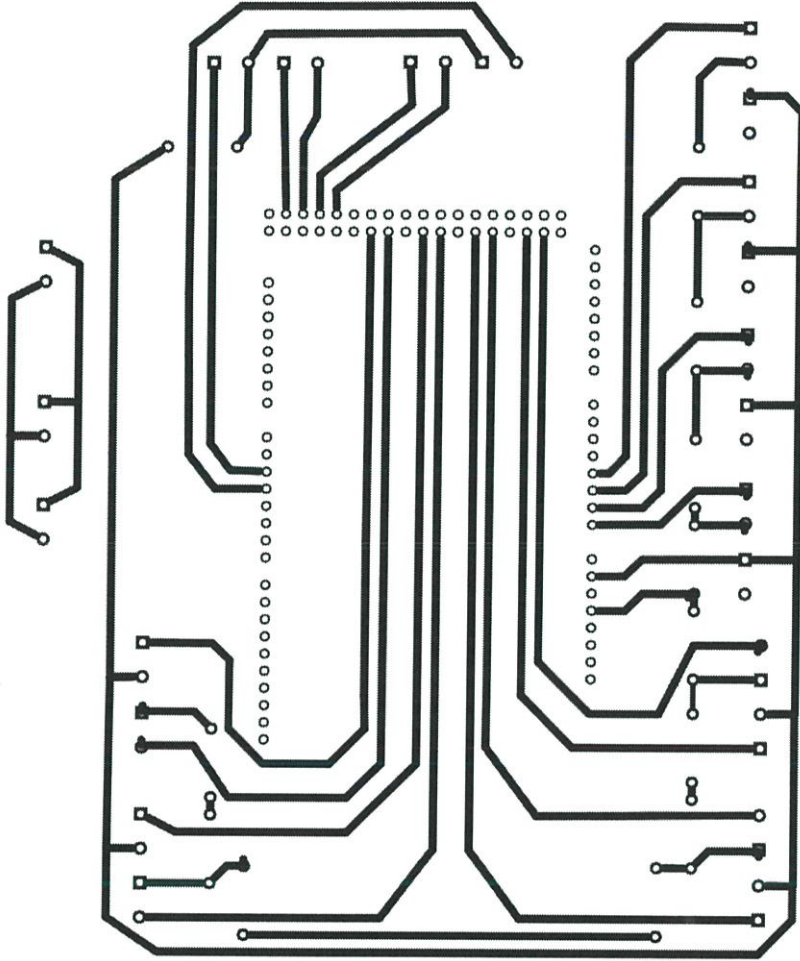
รูปที่ 3.2.4 วงจรขับมอเตอร์

จากในภาพที่ 3.2.2 จะพบว่าในวงจรที่เห็นจะมีการใช้ MOSFET แบบ p-n-p และ n-p-n โดยหากกำหนดให้ MOSFET แบบ p-n-p มีค่า HIGH แล้วให้ MOSFET แบบ n-p-n มีค่าเป็น LOW ตัว MOSFET ที่เป็นแบบ p-n-p จะทำงานเป็นเหมือนกับการสับ สวิตช์เป็นการ short circuit เพื่อให้วงจรทำงานและในทางกลับกัน MOSFET ที่เป็นแบบ n-p-n ก็จะทำงานเป็นการ open circuit แทน โดยการที่มอเตอร์จะหมุนไปทางใดนั้นต้องพิจารณาจากวงจรแล้วนำไปเขียนโค้ดเพื่อขับมอเตอร์

จากในภาพที่ 3.2.2 เป็นการแสดงวงจรที่ทำสำเร็จแล้วพร้อมใช้งานสำหรับการขับมอเตอร์ให้หมุนไปด้านหน้าหรือหมุนกลับในทิศทางตรงกันข้าม

### 3.2.3 สร้างบอร์ดหลักของการขับเคลื่อนหุ่นยนต์

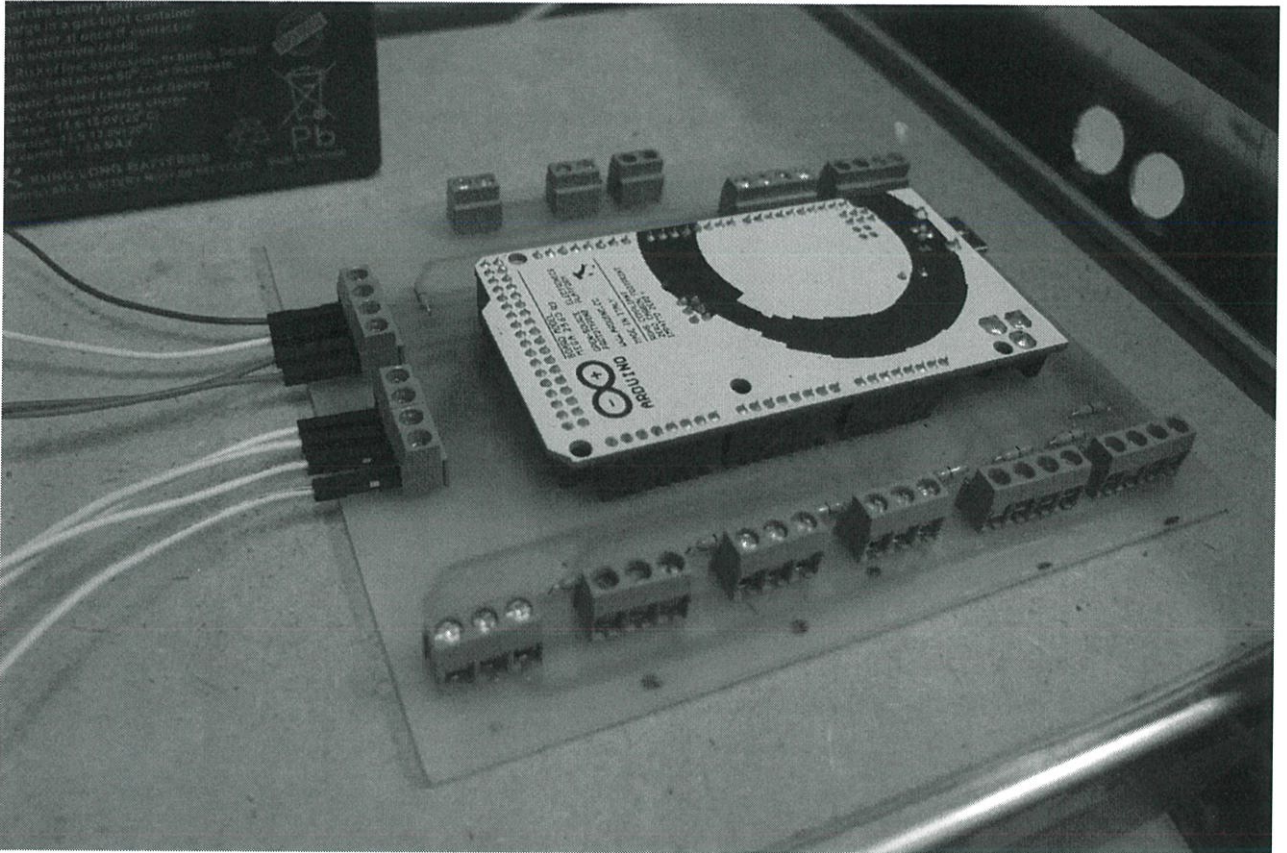
#### 1) ออกแบบ PCB โดยใช้โปรแกรม Fritzing



รูปที่ 3.2.5 บอร์ดหลักที่เชื่อมต่อ Arduino กับ sensor ต่างๆ

จากภาพที่ 3.2.5 เป็นการออกแบบวงจรเพื่อนำอุปกรณ์ต่างๆมาต่อเข้ากับบอร์ด Arduino เป็นเสมือนการออกแบบให้ตัววงจรเป็นศูนย์รวมของอุปกรณ์ต่อพ่วงประเภท Hardware ด้วย

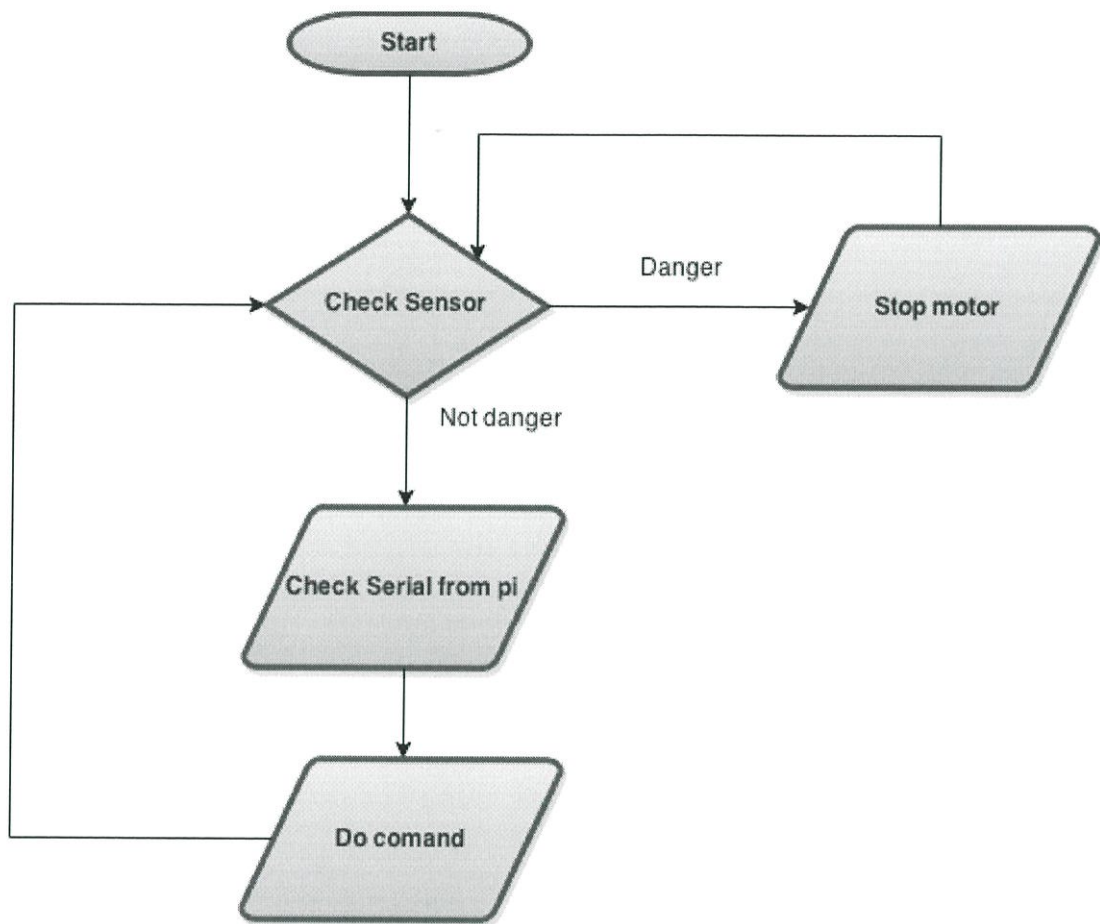
## 2) ลงอุปกรณ์



รูปที่ 3.2.6 รูปผลลัพธ์หลังจากกัตบอร์ด

ในภาพที่ 3.2.6 เป็นตัวบอร์ดที่นำมาใช้งานจริงโดยมีช่องทางสำหรับการเชื่อมต่อกับอุปกรณ์ต่อพ่วงอื่นๆ อยู่หลายช่องทางสำหรับนำอุปกรณ์มาเชื่อมต่อใช้งาน

## 3.2.4 เขียนโปรแกรมการขับเคลื่อนหุ่นยนต์

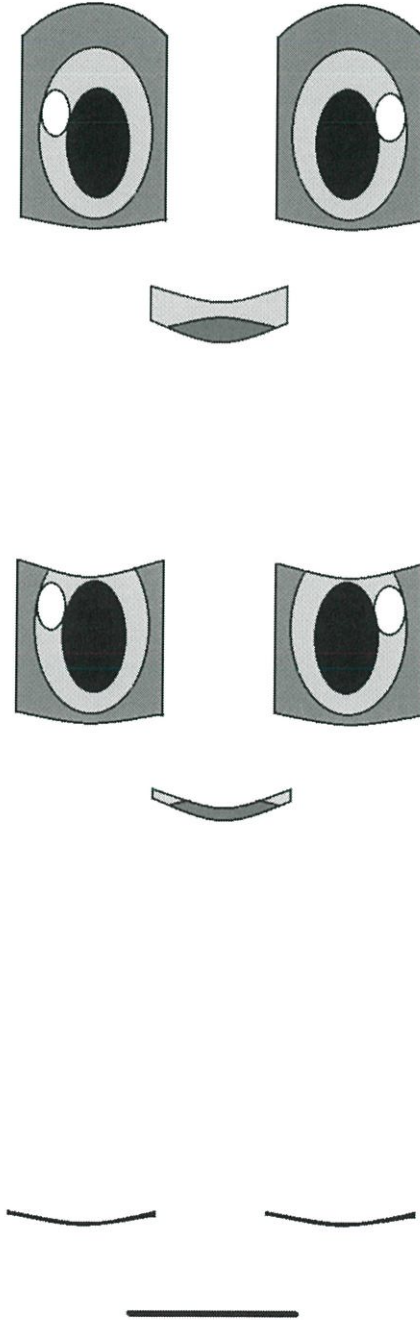


รูปที่ 3.2.7 ออกแบบระบบแนะนำห้อง lab

### 3.3 ส่วนของการแสดงผลบน iPad

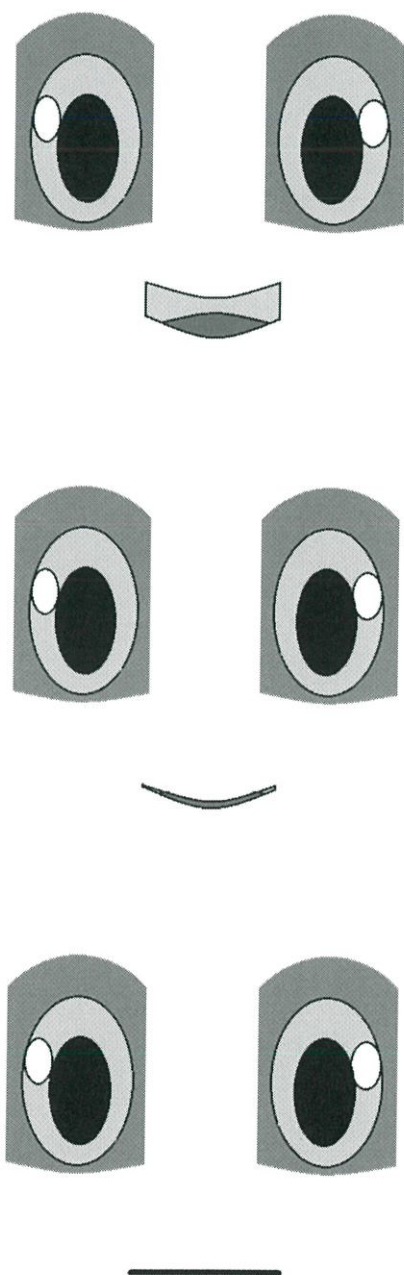
#### 3.3.1 ออกแบบหน้าต่างในกริยาต่างๆ

##### 1) หน้าปกดี (กระพริบตา)



รูปที่ 3.3.1 รูปหน้าต่างหุ่นยนต์ตอนกระพริบตา

## 2) ตอนพูด (ขยับปาก)



รูปที่ 3.3.2 รูปหน้าตาหุ่นยนต์ตอนอิม

จากภาพที่ 3.3.1 และ 3.3.2 เป็นการแสดงให้เห็นถึงท่าทางการแสดงออกของหุ่นยนต์ผ่านหน้าจอ Ipad

### 3.3.2 เขียนโปรแกรม

#### 1) ติดตั้งไลบรารี AFNetworking

##### 1.1) Download CocoaPods

```
$ sudo gem install cocoapods
```

```
$ pod setup
```

##### 1.2) Create a Podfile

```
$ touch Podfile
```

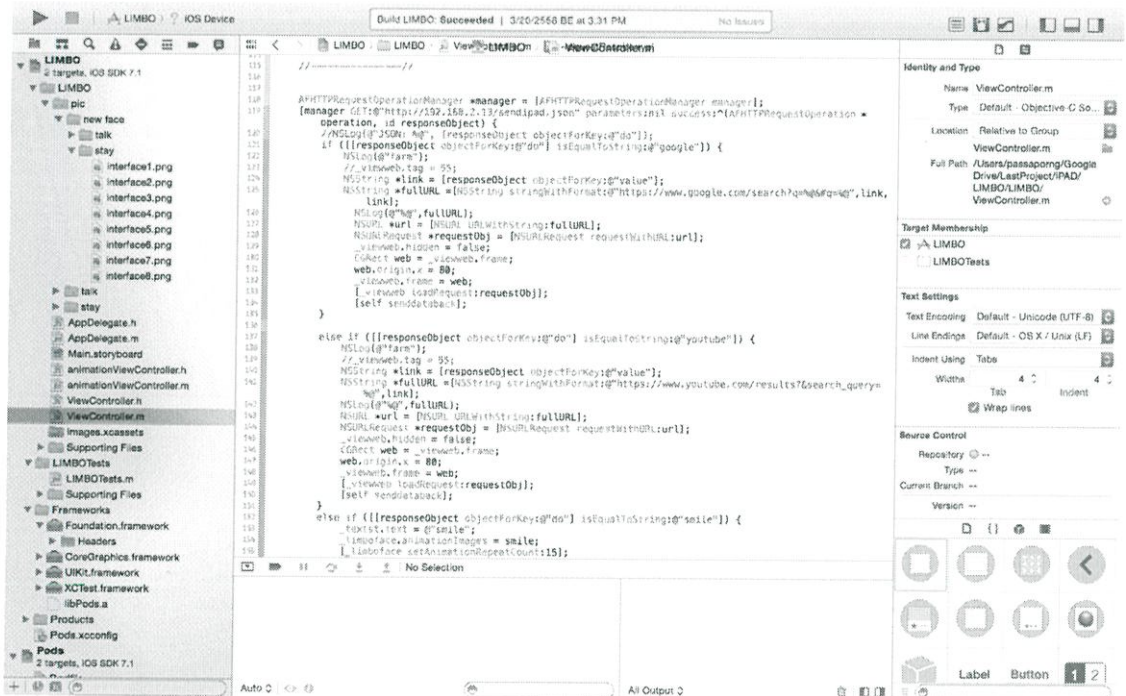
```
$ open -a Xcode Podfile
```

##### 1.3) Install Dependencies

```
$ pod install
```

```
$ open <YourProjectName>.xcworkspace
```

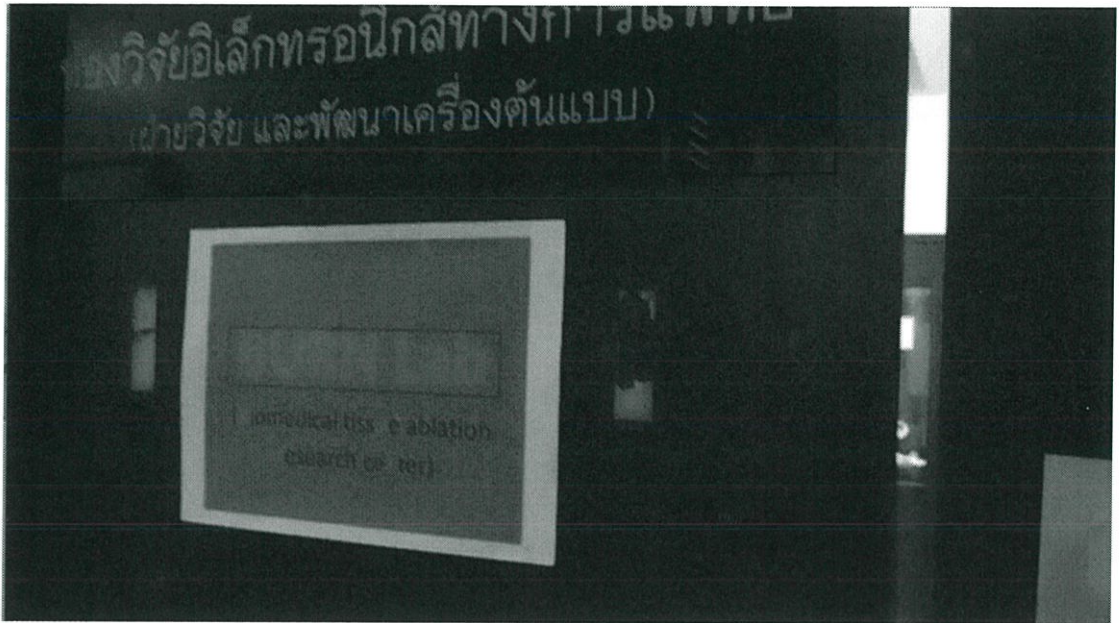
#### 2) เขียนโปรแกรมผ่าน xcode



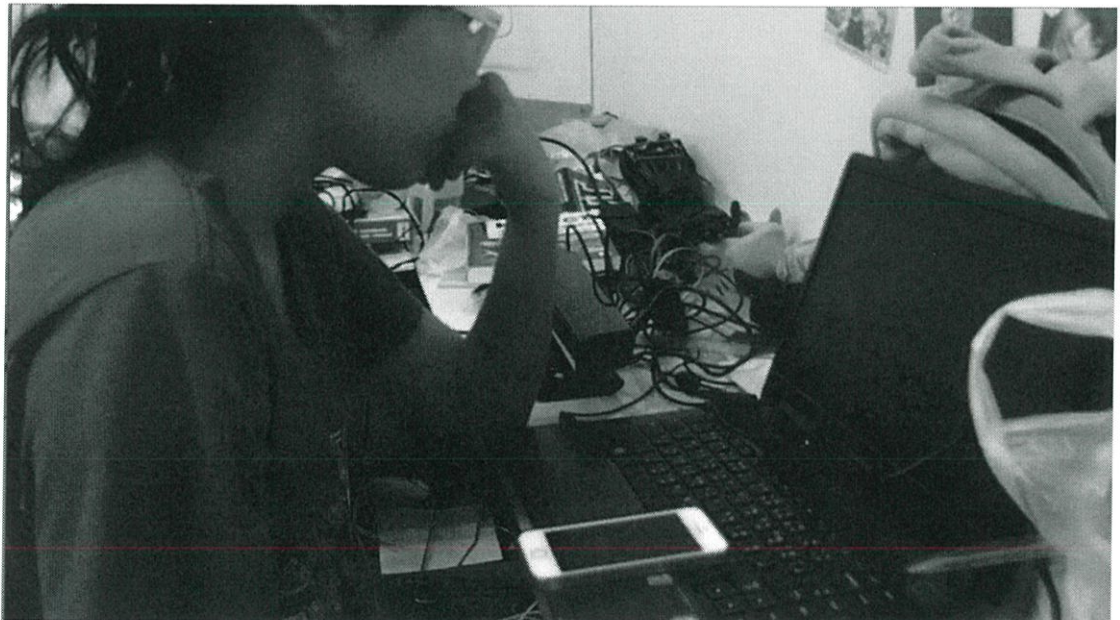
### 3.4 ทำวีดิโอฟรีเซ็นห้อง lab

3.4.1 ถ่ายวีดิโอบรรยากาศ / สัมภาษณ์งานของเพื่อนๆ

1) บรรยากาศของห้อง Lab



รูปที่ 3.3.4 การถ่ายทำบรรยากาศในห้อง



รูปที่ 3.3.5 การถ่ายทำบรรยากาศในห้อง 2



รูปที่ 3.3.6 การถ่ายทำบรรยากาศในห้อง 3

จากภาพที่ 3.3.4 , 3.3.5 และ 3.3.6 เป็นบรรยากาศของห้องแลปตั้งแต่หน้าห้องโดยห้องนี้มีชื่อว่า Burn Lab ภายในห้องจะมีบรรยากาศของการทำงานที่หลากหลายเพราะภายในห้องมีนักศึกษาตั้งแต่ชั้นปีที่ 3 ไปจนถึงนักศึกษาระดับปริญญาโท ที่ช่วยกันสร้างสรรค์ผลงานในรูปแบบต่างๆออกมาเพื่อพัฒนาเทคโนโลยีสมัยใหม่ให้ก้าวไปไกลมากกว่าที่มีอยู่ในปัจจุบัน

## 3.4.2 ตัดต่อ

ใช้โปรแกรม iMovie ในการตัดต่อวิดีโอ



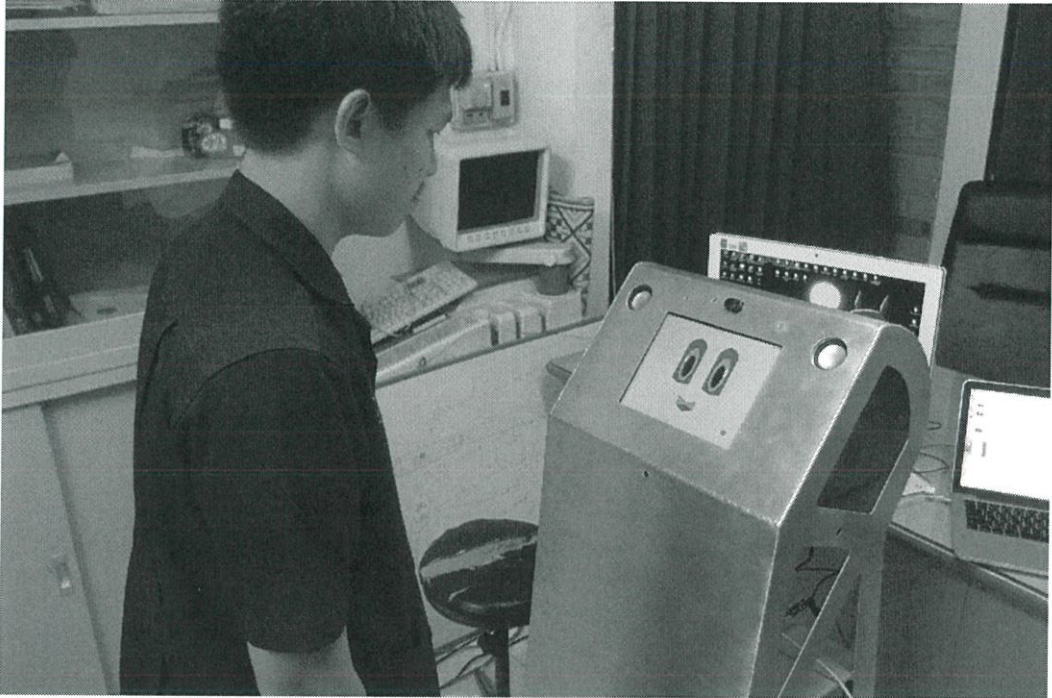
รูปที่ 3.3.7 ตัดต่อผ่านโปรแกรม iMovie

## บทที่ 4

### ผลการทดสอบระบบหุ่นยนต์

#### 4.1 ทดสอบการสนทนาถามตอบกับหุ่นยนต์

##### 4.1.1 ให้แนะนำตัว



รูปที่ 4.1.1 พูดคำสั่งให้แนะนำตัว

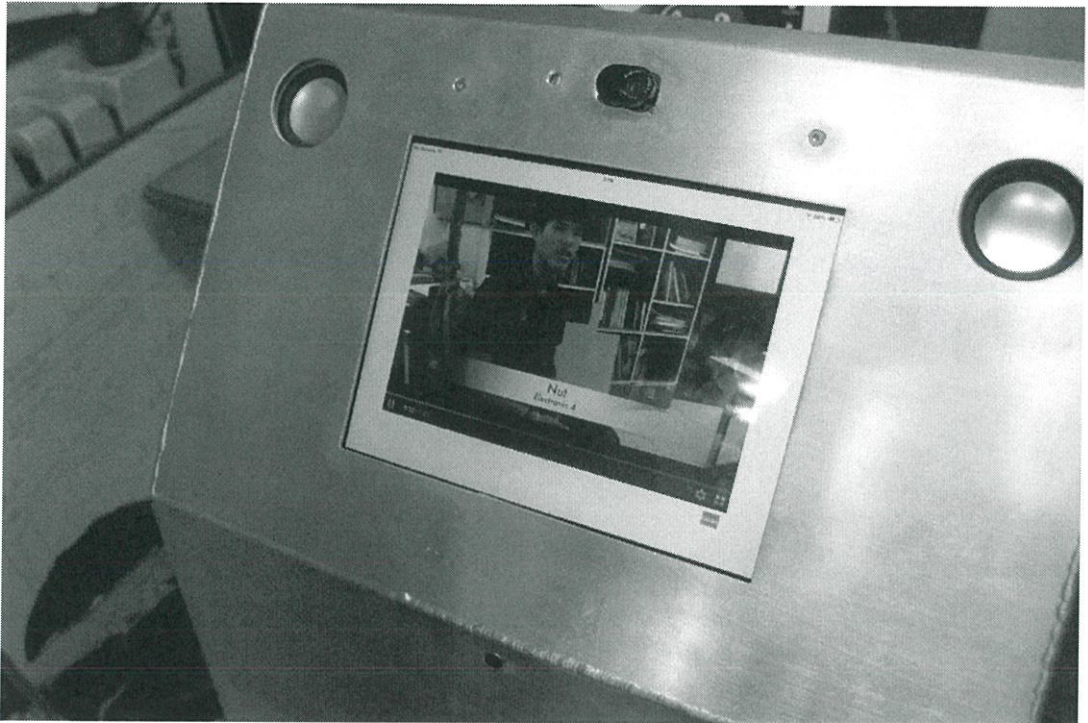
ตารางที่ 4.1.1 จับเวลาการสั่งงาน คำสั่ง what is your name?

บุคคลที่ทำการทดลอง	เวลา (s)				เฉลี่ย
	1	2	3	4	
ฟาร์ม	5.31	5.53	6.21	5.47	5.63
มาร์ค	6.75	6.51	6.43	5.72	6.35
บอม	6.10	6.21	5.57	6.13	6.00

ตารางที่ 4.1.2 เวลาการสั่งงาน คำสั่ง who are you?

บุคคลที่ทำการทดลอง	เวลา (s)				เฉลี่ย
	1	2	3	4	
ฟาร์ม	4.86	5.14	4.93	5.27	5.05
มาร์ค	ผิดพลาด	5.31	5.17	4.83	3.82
บอม	5.54	5.35	5.28	4.95	5.28

## 4.1.2 ให้แนะนำห้องแลป



รูปที่ 4.1.2 พูดคำสั่งให้แนะนำห้อง burn lab

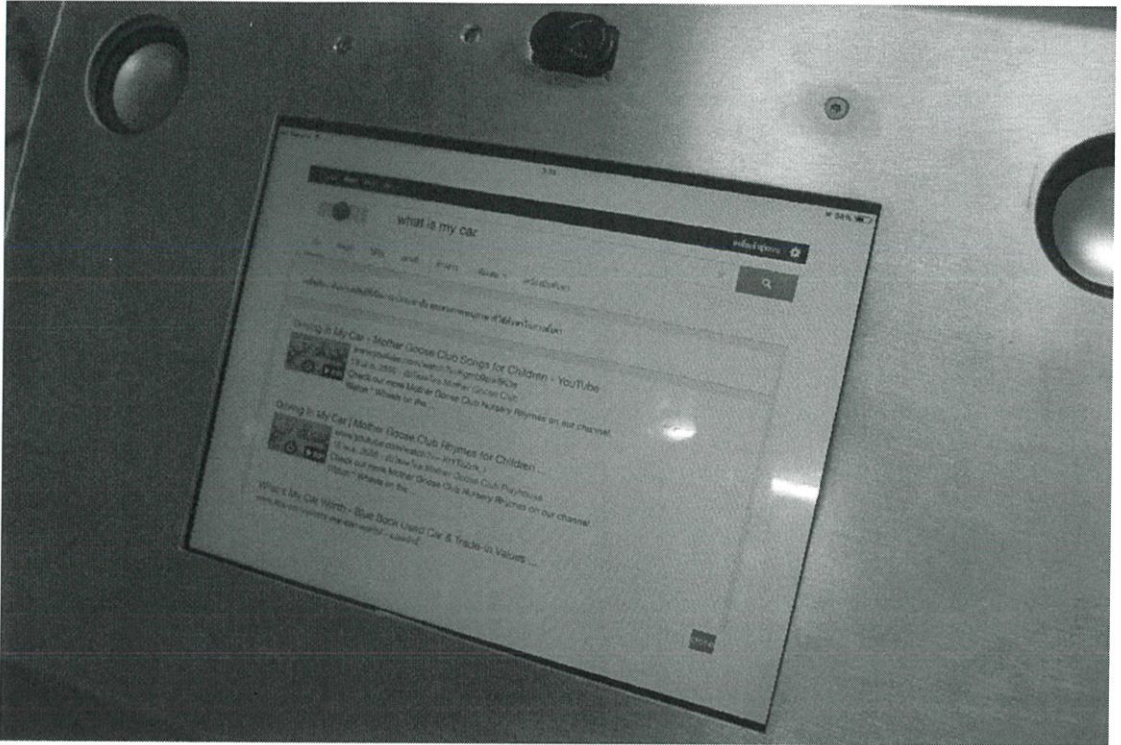
## ตารางที่ 4.1.3 จับเวลาการสั่งงาน คำสั่ง please introduce my lab

บุคคลที่ทำการทดลอง	เวลา (s)				เฉลี่ย
	1	2	3	4	
ฟาร์ม	5.24	5.53	5.67	5.57	5.50
มาร์ค	5.47	5.57	5.77	5.89	5.67
บอม	5.36	5.33	5.21	5.46	5.34

## ตารางที่ 4.1.4 จับเวลาการสั่งงาน คำสั่ง what is burnlab

บุคคลที่ทำการทดลอง	เวลา (s)				เฉลี่ย
	1	2	3	4	
ฟาร์ม	5.78	5.27	5.41	5.66	5.53
มาร์ค	5.43	5.76	5.22	5.13	5.38
บอม	5.51	5.74	5.81	5.51	5.64

## 4.1.3 ถามเพื่อหาคำตอบด้วย google / เล่นเพลง youtube



รูปที่ 4.1.4 คำสั่งใช้งาน google

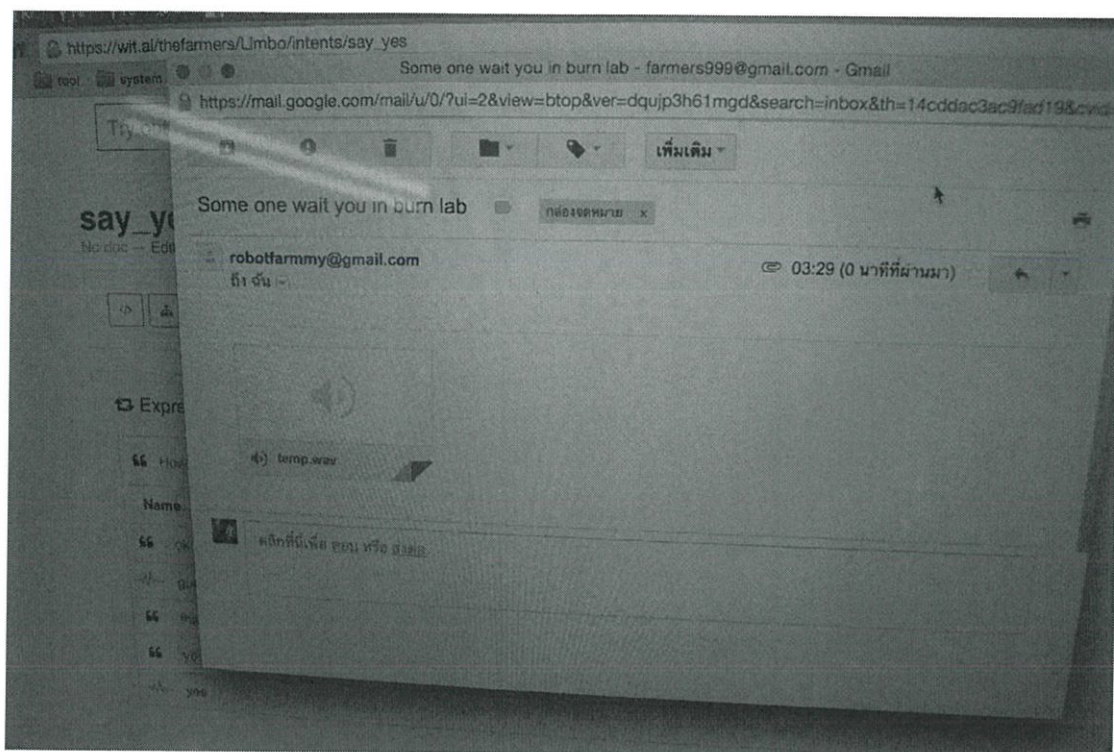
## ตารางที่ 4.1.5 จับเวลาการสั่งงาน คำสั่ง how many people in the world

บุคคลที่ทำการทดลอง	เวลา (s)				เฉลี่ย
	1	2	3	4	
ฟาร์ม	9.12	9.47	9.52	9.67	9.44
มาร์ค	9.84	9.58	9.19	9.71	9.58
บอม	ผิดพลาด	9.31	9.37	9.84	7.13

## ตารางที่ 4.1.6 จับเวลาการสั่งงาน คำสั่ง how many people in thailand

บุคคลที่ทำการทดลอง	เวลา (s)				เฉลี่ย
	1	2	3	4	
ฟาร์ม	12.53	12.58	12.47	13.63	12.80
มาร์ค	14.99	15.12	15.51	15.11	15.18
บอม	15.08	14.84	15.04	15.31	15.06

## 4.1.4 โหมดเลขา



รูปที่ 4.1.5 แสดงอีเมลที่เข้ามาหลังจากใช้คำสั่งเลขา

ตารางที่ 4.1.7 จับเวลาการสั่งงาน คำสั่ง where is my adviser

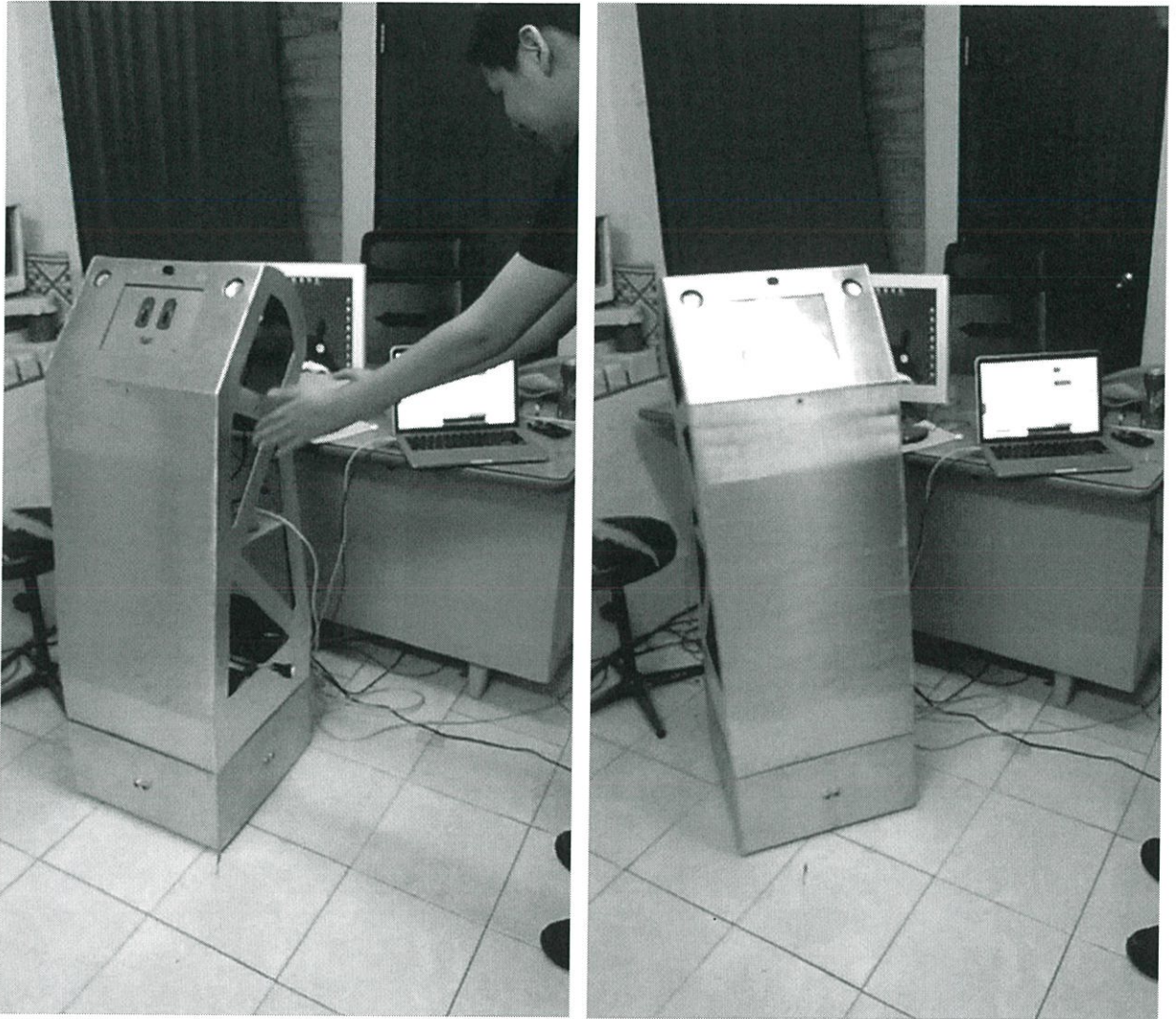
บุคคลที่ทำการทดลอง	เวลา (s)				เฉลี่ย
	1	2	3	4	
ฟาร์ม	ผิดพลาด	7.58	7.73	7.55	5.71
มาร์ค	8.23	7.89	7.74	7.84	7.92
บอม	6.98	7.05	7.22	6.87	7.03

ตารางที่ 4.1.8 จับเวลาการสั่งงาน คำสั่ง i want to see adviser

บุคคลที่ทำการทดลอง	เวลา (s)				เฉลี่ย
	1	2	3	4	
ฟาร์ม	7.55	7.24	7.11	6.89	7.19
มาร์ค	6.94	6.83	7.23	7.21	7.05
บอม	6.00	6.17	6.20	6.74	6.27

## 4.2 ทดสอบการเคลื่อนที่ของหุ่นยนต์

### 4.2.1 เรียกในทิศทางต่างๆ



รูปที่ 4.2.1 เรียกหุ่นยนต์ให้หันมาทิศทางที่เรียก

จากภาพที่ 4.2.1 เป็นการใช้เสียงเรียกเพื่อให้ตัวหุ่นยนต์หันมาตามทิศทางของเสียงที่ใช้เรียก แต่จากในภาพจะเห็นว่าเมื่อทำการใช้เสียงเรียกเพื่อให้ตัวหุ่นยนต์นั้นหันมาตามทิศทางของเสียงจะมีข้อผิดพลาดอยู่เกิดการหันมาไม่ตรงตามทิศทางที่ได้ใช้เสียงเรียกแต่จะมีความคลาดเคลื่อนเนื่องจากตัวเซ็นเซอร์ที่ใช้มีความแม่นยำไม่เพียงพอเท่าที่ควร

จากภาพที่ 4.2.2 และ จากภาพที่ 4.2.3 เป็นการใช้เสียงเรียกตัวหุ่นยนต์เข้ามาหาในตำแหน่งที่ยืนอยู่รวมไปถึงการเช็คระยะของตัวหุ่นยนต์ที่กำหนดไว้ในโปรแกรมเพื่อป้องกันไม่ให้เกิดการชนกับวัตถุรอบข้าง

#### 4.2.2 เรียกให้เข้ามาหา



รูปที่ 4.2.2 เรียกให้หุ่นยนต์เข้ามาหา

#### 4.2.3 ทดสอบการชนของหุ่นยนต์



รูปที่ 4.2.3 หยุดเมื่อใกล้ชน

## บทที่ 5 สรุปผล

### 5.1 สรุปผล

จากงานวิจัยชิ้นนี้ที่มุ่งเน้นเรื่องการศึกษาวิจัยเกี่ยวกับการสร้างหุ่นยนต์เพื่อใช้ในการประชาสัมพันธ์ข้อมูลในห้อง lab โดยสามารถแสดงข้อมูลให้ผู้มาเยี่ยมชมห้อง ทั้งนี้ยังสามารถพูดคุยตอบโต้กับคนที่มาเข้าชม และมีความสามารถในการเคลื่อนที่เข้าไปหาผู้พูดคุย เพื่อให้หุ่นยนต์มีปฏิสัมพันธ์กับผู้ชมมากขึ้น มีการแสดงอารมณ์ตอบโต้กับผู้ติดต่อทำให้ดูน่าสนใจยิ่งขึ้น

ผู้วิจัยได้ศึกษาค้นคว้าในเรื่องต่างๆ ที่สนใจนำความรู้ในแต่ละแขนงมาต่อยอดความรู้ มาประยุกต์ความรู้รวบรวมจนเกิดเป็นองค์ความรู้ต่างๆ เพื่อนำไปประยุกต์ใช้ได้แก่

1. การศึกษาระบบสมองกลฝังตัว Raspberrypi
2. การศึกษาโปรแกรมวิเคราะห์เสียงพูด / ตอบโต้ / การประยุกต์ใช้ api
3. การศึกษาการส่งข้อมูลแบบ serial ผ่านสายส่งเข้าสู่ Arduino ทั้งรับข้อมูลและส่งข้อมูล
4. การศึกษาการเขียนโปรแกรมผ่าน xcode เพื่อสร้างอนิเมชันแบบง่ายๆ
5. การศึกษาการเขียนโปรแกรมผ่าน xcode เพื่อติดต่อแบบไร้สายกับ Raspberrypi
6. การศึกษาการออกแบบตัวหุ่นยนต์

และจากการทดลองการตอบสนองของหุ่นยนต์ ในการถามตอบ จะพบว่าการสนทนาทั่วไปใช้เวลาในการคิดเฉลี่ย 7 - 9 วินาที และการถามเพื่อหาคำตอบใช้เวลาในการคิดเฉลี่ย 12 วินาทีซึ่งมากกว่าเพราะต้องใช้ API ติดต่อกับฐานข้อมูล wolfram และความเร็วในการตอบสนองขึ้นอยู่กับความเร็วอินเทอร์เน็ตด้วย

### 5.2 แนวทางการพัฒนา

หุ่นยนต์ไม่ควรที่จะรับคำสั่งได้ไม่กี่คำสั่งหุ่นยนต์ควรมีความคิด ความฉลาดในการรับคำสั่งที่มากขึ้นไม่เพียงแค่คำสั่งถ้อยได้ เดินได้เท่านั้น แต่ต้องมีการวิเคราะห์ต่างๆให้มากขึ้น เช่น มีเสียงให้ตอบรับ ให้ตกใจ ให้มีอารมณ์ ไปในทิศทางเดียวกัน พฤติกรรมต่างๆต้องปรับปรุงให้เสมือนมันฉลาดมากที่สุด และพูดคุยได้สนุกสนานเหมือนเพื่อนคนหนึ่งให้ได้

หุ่นยนต์ควรมีหน้าตาที่เป็นมิตร สามารถสื่อสารอารมณ์ต่างๆได้อย่างสอดคล้อง เวลาตกใจ ตกใจ ควรมีสีหน้าที่ชัดเจนผ่านใบหน้าของหุ่นยนต์ ไม่ใช่แค่เพียงใบหน้าหน้าตาอารมณ์ที่แสดงอยู่บนหน้าจอ เราสามารถตอบโต้ พิมพ์คำสั่งบนหน้าจอได้ด้วย เปรียบเสมือนเป็นหน้าตาที่เป็นตัวเชื่อมโยงเรากับหุ่นยนต์นอกจากเสียงพูดคุย

หุ่นยนต์ควรมีการออกแบบที่พร้อมจะรับน้ำหนักชิ้นส่วนต่างๆ ได้ ควรออกแบบให้ดูเป็นมิตร และใช้วัสดุที่มีราคาถูกเพื่อเป็นการประหยัด

## เอกสารอ้างอิง

- [1] กอบเกียรติ สระอุบล. การเขียนโปรแกรม obj.c เบื้องต้น. สำนักพิมพ์ อินเทอร์เน็ตเเดีย. 2014
- [2] เอกชัย กัลยาภิตติกุล. เก่ง Solid Works ให้ครบสูตร. สำนักพิมพ์ วิตตี้กรุ๊ป. 2011
- [3] Michael Margolis. Arduino cook book Second edition. Reilly Media Inc. 2011
- [4] Matt Richardson and Shawn Wallace. Getting Started with Raspberry Pi. Reilly Media Inc. 2012
- [5] David Beazley and Brian K. Jones. Python cookbook. Reilly Media Inc. 2013

ภาคผนวก โปรแกรม

### ออกแบบโปรแกรมเปลี่ยนจากเสียงเป็นตัวหนังสือผ่าน api ของ wit.ai

```

import time
import json
import os

output = subprocess.check_output(["curl 'http://localhost:9877/start?
autoend=true&access_token=3S4OYM5GGBYPQEEXWUZEW7W6F7XFLEDZ
'", shell=True])
output_json = json.loads(output)
intent    = output_json['outcomes'][0]['intent']
msg       = output_json['outcomes'][0]['_text']
text_file = open("T_voicetointent.txt", "w")
text_file.write(intent)
text_file.close()
text_file = open("T_voicetomsg.txt", "w")
text_file.write(msg)
text_file.close()
print("msg = ", msg)
print("intent = ", intent)

```

### ออกแบบโปรแกรมเปลี่ยนจากตัวหนังสือเป็นเสียงพูดด้วย api ของ google

```

#!/bin/bash
say() { local IFS=+;/usr/bin/mplayer -ao alsa:device=hw=1.0 -really-quiet -
noconsolecontrols "http://translate.google.com/translate_tts?tl=en&q=$*"; }
say $*

```

3.4.3 สร้างไฟล์ texttosound.sh

3.4.4 เขียนโปรแกรมตัวหลักในการพูด

```

#!/bin/bash
python /home/pi/starttalkipad.py
python /home/pi/M_sendtexttocom.py
./speakraw.sh $(cat T_say.txt)
python /home/pi/stayipad.py

```

ออกแบบโปรแกรมแกนหลักของการคิด

```

import subprocess
import time
import json
while True:
    #checksound from arduino
    #subprocess.call(['python chkSoundArduino.py'],shell = True)

    #listen
    subprocess.call(['python voice2txt.py'],shell = True)

    #check intent
    chk = open('T_voicetointent.txt','r+')
    voiceText1 = chk.read().rstrip().lower()
    chk.close

    #check first sound
    if voiceText1 == 'say_hi':
        text_file = open("T_say.txt", "w")
        text_file.write("Hi,My name is Limbo , i am robot from burn lab what
do your want")
        text_file.close()
        subprocess.call(['./texttosound.sh'],shell = True)

    elif voiceText1 == 'power_off':
        text_file = open("T_say.txt", "w")
        text_file.write("i am shutting down")
        text_file.close()
        subprocess.call(['./texttosound.sh'],shell = True)
        break

    while True:
        subprocess.call(['python voice2txt.py'],shell = True)
        chk_r = open('T_voicetointent.txt','r+')
        voiceText = chk_r.read().rstrip().lower()
        chk_r.close

        if voiceText == 'google':
            subprocess.call(['python M_google.py'],shell = True)

```

```

elif voiceText == 'youtube':
    subprocess.call(['python M_youtube.py'],shell = True)

elif voiceText == 'get_name':
    text_file = open("T_say.txt", "w")
    text_file.write("my name is limbo , i think i love you")
    text_file.close()
    subprocess.call(['./texttosound.sh'],shell = True)

elif voiceText == 'get_fullburnlab':
    subprocess.call(['python M_fullburnlab.py'],shell = True)

elif voiceText == 'mode_assistants':
    subprocess.call(['python M_assistants.py'],shell = True)

elif voiceText == 'mode_chackinout':
    subprocess.call(['python M_chackinout.py'],shell = True)

elif voiceText == 'say_thank':
    text_file = open("T_say.txt", "w")
    text_file.write("your welcome")
    text_file.close()
    subprocess.call(['./texttosound.sh'],shell = True)

elif voiceText == 'bye':
    text_file = open("T_say.txt", "w")
    text_file.write("bye bye ser")                #bye bye
    text_file.close()
    subprocess.call(['./texttosound.sh'],shell = True) #bye bye
    break

else:
    text_file = open("T_say.txt", "w")
    text_file.write("what else")                # text bye bye
    text_file.close()
    subprocess.call(['./texttosound.sh'],shell = True) # say bye bye

```

โมดูลการค้นหาคำตอบด้วย wolfram และ google

```

import subprocess
import time
import json

text_file = open("T_say.txt", "w")
text_file.write("what do you want to know")           # text quetion
text_file.close()
subprocess.call(['./texttosound.sh'],shell = True)     # say quetion
subprocess.call(['python voice2txt.py'],shell = True)  # voice to txt
subprocess.call(['cp T_voicetomsg.txt googlemsg.txt'],shell = True)
subprocess.call(['python starttalkipad.py'],shell = True)
subprocess.call(['./M_google.sh'],shell = True)       # say anser
subprocess.call(['python stayipad.py'],shell = True)

time.sleep(2) # delays for 5 seconds
text_file = open("T_say.txt", "w")
text_file.write("do you want to see any details")     # text quetion
text_file.close()
subprocess.call(['./texttosound.sh'],shell = True)    # say quetion

#listen
subprocess.call(['python voice2txt.py'],shell = True)

#check intent
chk = open('T_voicetointent.txt','r+')
voiceText1 = chk.read().rstrip().lower()
chk.close
if voiceText1 == 'say_yes':
    text_file = open("T_say.txt", "w")
    text_file.write("plase see on ipad")               # text quetion
    text_file.close()
    subprocess.call(['./texttosound.sh'],shell = True) # say quetion

    chk = open('googlemsg.txt','r+')
    voiceText2 = chk.read().rstrip().lower().replace(" ", "+")
    print voiceText2
    chk.close

    text_file2 = open("/var/www/sendipad.json", "w")

```

```
text_file2.write({'do':"google","value":"%s"}%voiceText2)
text_file2.close()
```

```
elif voiceText1 == 'say_no':
    text_file = open("T_say.txt", "w")
    text_file.write("ok") # text quetion
    text_file.close()
    subprocess.call(['./texttosound.sh'],shell = True) # say quetion
subprocess.call(['rm googlemsg.txt'],shell = True)
```

โมดูลผู้ช่วย

```

import subprocess
import time
import json
import smtplib

from email.mime.text import MIMEText
from email.mime.audio import MIMEAudio

text_file = open("T_say.txt","w")
text_file.write("im chack in data base plese wait a minutes")
text_file.close()
subprocess.call(['./texttosound.sh'],shell = True)           #say where and q

i = 0
x = 0

with open("assistanrs.json") as json_file:
    data = json.load(json_file)

if data["member"][0]["where"] != "lab":
    where = data["member"][0]["where"]
    text_file = open("T_say.txt","w")
    txt1 = "OH is not living hare, He live in %s do you want leave a message to
oh" % (where)
    text_file.write(txt1)
    text_file.close()
    subprocess.call(['./texttosound.sh'],shell = True)           #say where and q
    subprocess.call(['python voice2txt.py'],shell = True) # voice to txt
    subprocess.call(['cp T_voicetointent.txt assistanrsmgs.txt'],shell = True)
    chk = open('assistanrsmgs.txt','r+')
    chkassistanrs = chk.read().rstrip().lower()
    chk.close()
    if chkassistanrs == "say_yes":
        text_file = open("T_say.txt","w")
        text_file.write("ok , plese say in ten secound")
        text_file.close()
        subprocess.call(['./texttosound.sh'],shell = True)           #say where
        subprocess.call(['arecord -d 10 temp.wav'],shell = True) #say where
        subprocess.call(['python testsendgmail.py'],shell = True) #say where
        text_file = open("T_say.txt","w")

```

```
text_file.write("send compleat")
text_file.close()
subprocess.call(['./texttosound.sh'],shell = True)    #say where

elif chkassistanrs == "no":
    text_file = open("T_say.txt","w")
    text_file.write("roger that")
    text_file.close()
    subprocess.call(['./texttosound.sh'],shell = True)    #say where
else:
    text_file = open("T_say.txt","w")
    text_file.write("roger that")
    text_file.close()
    subprocess.call(['./texttosound.sh'],shell = True)    #say where
else:
    text_file = open("T_say.txt","w")
    text_file.write("OH is here")
    text_file.close()
    subprocess.call(['./texttosound.sh'],shell = True)    #say where
    print("ok")
```

โมดูลเปิด youtube

```
import subprocess
```

```
import time
```

```
import json
```

```
text_file = open("T_say.txt", "w")
```

```
text_file.write("which video do you want to watch") # text quetion
```

```
text_file.close()
```

```
subprocess.call(['./texttosound.sh'],shell = True) # say quetion
```

```
subprocess.call(['python voice2txt.py'],shell = True) # voice to txt
```

```
chk = open('T_voicetomsg.txt','r+')
```

```
voiceText1 = chk.read().rstrip().lower().replace(' ', '+')
```

```
chk.close
```

```
text_file2 = open("T_say.txt", "w")
```

```
text_file2.write("ok Let's see") # text quetion
```

```
text_file2.close()
```

```
subprocess.call(['./texttosound.sh'],shell = True) # say
```

```
text_file1 = open("/var/www/sendipad.json", "w")
```

```
text_file1.write({'do':"youtube","value":"%s"}%voiceText1)
```

```
text_file1.close()
```

โมดูลการแนะนำห้อง lab

```

import subprocess
import time
import json

text_file = open("T_say.txt", "w")
text_file.write("i will introduce you about burn lab")
text_file.close()
subprocess.call(['./texttosound.sh'],shell = True)
text_file1 = open("/var/www/sendipad.json", "w")
text_file1.write({"do":"youtube2","value":"watch?v=_tvN4mLNsew&autoplay=1"})
text_file1.close()

time.sleep(200) # delays for 200 seconds
text_file = open("T_say.txt", "w")
text_file.write("do you want to know about project")           # text quetion
text_file.close()
subprocess.call(['./texttosound.sh'],shell = True)             # say quetion

#listen
subprocess.call(['python voice2txt.py'],shell = True)

#check intent
chk = open('T_voicetointent.txt','r+')
voiceText1 = chk.read().rstrip().lower()
chk.close()
if voiceText1 == 'say_yes':
    text_file = open("T_say.txt", "w")
    text_file.write("ok let's see project")                       # text quetion
    text_file.close()
    subprocess.call(['./texttosound.sh'],shell = True)          # say quetion
    text_file1 = open("/var/www/sendipad.json", "w")
    text_file1.write({"do":"youtube2","value":"watch?v=_tvN4mLNsew"})
    text_file1.close()
elif voiceText1 == 'say_no':
    text_file = open("T_say.txt", "w")
    text_file.write("ok")                                         # text quetion
    text_file.close()
    subprocess.call(['./texttosound.sh'],shell = True)          # say quetion

```

เขียนโปรแกรม Arduino

```
// LastProjectRobot.ino
```

```
// ----- MOTOR ---- //
```

```
int Lin1 = 29;
```

```
int Lin2 = 27;
```

```
int LPWM = 3;
```

```
int Rin1 = 25;
```

```
int Rin2 = 23;
```

```
int RPWM = 5;
```

```
// ----- SENSOR ---- //
```

```
//Distance
```

```
unsigned long Time_Echo_us = 0;
```

```
unsigned int F_EchoPin = 32;           // connect Pin 2(Arduino digital io) to Echo at  
US-020
```

```
unsigned int F_TriggerPin = 34;       // connect Pin 3(Arduino digital io) to Trig at  
US-020
```

```
unsigned int B_EchoPin = 44;          // connect Pin 2(Arduino digital io) to Echo at  
US-020
```

```
unsigned int B_TriggerPin = 46;       // connect Pin 3(Arduino digital io) to Trig at  
US-020
```

```
unsigned int L_EchoPin = 50;          // connect Pin 2(Arduino digital io) to Echo at  
US-020
```

```
unsigned int L_TriggerPin = 52;       // connect Pin 3(Arduino digital io) to Trig at  
US-020
```

```
unsigned int R_EchoPin = 38;          // connect Pin 2(Arduino digital io) to Echo at  
US-020
```

```
unsigned int R_TriggerPin = 40;       // connect Pin 3(Arduino digital io) to Trig at  
US-020
```

```
//Sound
```

```
// ----- VALUE ---- //
```

```
int FrontDistance;
int BackDistance;
int LeftDistance;
int RightDistance;
int speed = 0;
int attack = 600;
int inputSerial;
int randomWalker;
int addspeed;

int FrontSound;
int BackSound;
int LeftSound;
int RightSound;

int P_FrontSound = A3;
int P_BackSound = A2;
int P_LeftSound = A1;
int P_RightSound = A0;

// ---- mode ---- //
void MotorStop();
void MotorGo(int speed);
void MotorBack(int speed);
void MotorLeft(int speed);
void MotorRight(int speed);
void CheckSensor();

void setup() {
  Serial.begin(115200);

  //set control motor
  pinMode(Lin1, OUTPUT);
  pinMode(Lin2, OUTPUT);
  pinMode(LPWM, OUTPUT);
  pinMode(Rin1, OUTPUT);
  pinMode(Rin2, OUTPUT);
  pinMode(RPWM, OUTPUT);

  //set Distanc
```

```

    pinMode(F_EchoPin, INPUT);           //Set EchoPin as input, to receive
measure result from US-020
    pinMode(F_TrigPin, OUTPUT);         //Set TrigPin as output, used to send
high pusle to trig measurement (>10us)

    pinMode(B_EchoPin, INPUT);         //Set EchoPin as input, to receive
measure result from US-020
    pinMode(B_TrigPin, OUTPUT);        //Set TrigPin as output, used to send
high pusle to trig measurement (>10us)

    pinMode(L_EchoPin, INPUT);         //Set EchoPin as input, to receive
measure result from US-020
    pinMode(L_TrigPin, OUTPUT);        //Set TrigPin as output, used to send
high pusle to trig measurement (>10us)

    pinMode(R_EchoPin, INPUT);         //Set EchoPin as input, to receive
measure result from US-020
    pinMode(R_TrigPin, OUTPUT);        //Set TrigPin as output, used to send
high pusle to trig measurement (>10us)

    //set Distance
    pinMode(P_FrontSound, INPUT);
    pinMode(P_BackSound, INPUT);
    pinMode(P_LeftSound, INPUT);
    pinMode(P_RightSound, INPUT);
}

void loop(){
    CheckSensor();
    //MotorGo(30);
    if (Serial.available()) {
        inputSerial = Serial.read()- '0';

        //come on
        if(inputSerial == 1){
            Serial.println("come on");
            MotorGo(40);
            CheckSensor();
        }
    }
}

```

```
else if(inputSerial == 9){
MotorStop();
  while(1){
    FrontSound = analogRead(P_FrontSound);
    BackSound = analogRead(P_BackSound);
    LeftSound = analogRead(P_LeftSound);
    RightSound = analogRead(P_RightSound);
Serial.println("sound");
    Serial.println(FrontSound);
    Serial.println(BackSound);
    Serial.println(LeftSound);
    Serial.println(RightSound);
    if(FrontSound < 1000){
      Serial.println("face front");
      MotorStop();
      break;
    }
    else if(BackSound < 1000){
      Serial.println("face back");
      MotorStop();
      delay(1000);
      MotorRight(50);
      delay(2500);
      MotorStop();
      break;
    }
    else if(LeftSound < 1000){
      Serial.println("face left");
      MotorStop();
      delay(1000);
      MotorLeft(50);
      delay(1500);
      MotorStop();
      break;
    }
    else if(RightSound < 1000){
      Serial.println("face right");
      MotorStop();
      delay(1000);
      MotorRight(50);
```

```
        delay(1500);
        MotorStop();
        break;
    }
}
delay(5000);

}
//stop
else if(inputSerial == 2){
    MotorStop();
    CheckSensor();
}
else if(inputSerial == 3){
    MotorRight(35);
    delay(2200);
    CheckSensor();
}
else if(inputSerial == 4){
    MotorLeft(35);
    delay(2200);
    MotorStop();
    CheckSensor();
}
//random walk
else if(inputSerial == 5){
    randomWalker = random(1, 4);
    if(randomWalker == 1){
        MotorGo(100);
        CheckSensor();
    }
    else if(randomWalker == 2){
        MotorBack(100);
        CheckSensor();
    }
    else if(randomWalker == 3){
        MotorLeft(100);
        CheckSensor();
    }
    else if(randomWalker == 4){
```

```
        MotorRight(100);
        CheckSensor();
    }
    CheckSensor();
}
else{
    MotorRight(35);
    delay(2200);
    MotorStop();
}
}
}
```

//motor control mode

```
void MotorStop(){
    digitalWrite(Lin1, LOW);
    digitalWrite(Lin2, LOW);
    analogWrite(LPWM, 0);

    digitalWrite(Rin1, LOW);
    digitalWrite(Rin2, LOW);
    analogWrite(RPWM, 0);
}
```

```
void MotorGo(int speed){
    digitalWrite(Lin1, HIGH);
    digitalWrite(Lin2, LOW);
    analogWrite(LPWM, speed);

    digitalWrite(Rin1, HIGH);
    digitalWrite(Rin2, LOW);
    analogWrite(RPWM, speed);
}
```

```
void MotorBack(int speed){
    digitalWrite(Lin1, LOW);
    digitalWrite(Lin2, HIGH);
    analogWrite(LPWM, speed);

    digitalWrite(Rin1, LOW);
```

```

digitalWrite(Rin2, HIGH);
analogWrite(RPWM, speed);
}

```

```

void MotorLeft(int speed){
  digitalWrite(Lin1, LOW);
  digitalWrite(Lin2, HIGH);
  analogWrite(LPWM, speed);

```

```

  digitalWrite(Rin1, HIGH);
  digitalWrite(Rin2, LOW);
  analogWrite(RPWM, speed);
}

```

```

void MotorRight(int speed){
  digitalWrite(Lin1, HIGH);
  digitalWrite(Lin2, LOW);
  analogWrite(LPWM, speed);

```

```

  digitalWrite(Rin1, LOW);
  digitalWrite(Rin2, HIGH);
  analogWrite(RPWM, speed);
}

```

```

void CheckSensor(){
  //calculate
  //front
  digitalWrite(F_TrigPin, HIGH);          //begin to send a high pulse, then US-020
begin to measure the distance
  delayMicroseconds(50);                 //set this high pulse width as 50us (>10us)
  digitalWrite(F_TrigPin, LOW);          //end this high pulse
  Time_Echo_us = pulseIn(F_EchoPin, HIGH); //calculate the pulse width
at EchoPin,
  if((Time_Echo_us < 60000) && (Time_Echo_us > 1)) //a valid pulse width should
be between (1, 60000).
  {
    FrontDistance = (Time_Echo_us*34/100)/2; //calculate the distance by pulse
width, Len_mm = (Time_Echo_us * 0.34mm/us) / 2 (mm)
  }
  //back
}

```

```

    digitalWrite(B_TriggerPin, HIGH);           //begin to send a high pulse, then US-020
begin to measure the distance
    delayMicroseconds(50);                     //set this high pulse width as 50us (>10us)
    digitalWrite(B_TriggerPin, LOW);          //end this high pulse
    Time_Echo_us = pulseIn(B_EchoPin, HIGH);   //calculate the pulse width
at EchoPin,
    if((Time_Echo_us < 60000) && (Time_Echo_us > 1)) //a valid pulse width should
be between (1, 60000).
    {
        BackDistance = (Time_Echo_us*34/100)/2; //calculate the distance by pulse
width, Len_mm = (Time_Echo_us * 0.34mm/us) / 2 (mm)
    }

//left
    digitalWrite(L_TriggerPin, HIGH);         //begin to send a high pulse, then US-020
begin to measure the distance
    delayMicroseconds(50);                     //set this high pulse width as 50us (>10us)
    digitalWrite(L_TriggerPin, LOW);          //end this high pulse
    Time_Echo_us = pulseIn(L_EchoPin, HIGH);   //calculate the pulse width
at EchoPin,
    if((Time_Echo_us < 60000) && (Time_Echo_us > 1)) //a valid pulse width should
be between (1, 60000).
    {
        LeftDistance = (Time_Echo_us*34/100)/2; //calculate the distance by pulse
width, Len_mm = (Time_Echo_us * 0.34mm/us) / 2 (mm)
    }

//right
    digitalWrite(R_TriggerPin, HIGH);         //begin to send a high pulse, then US-020
begin to measure the distance
    delayMicroseconds(50);                     //set this high pulse width as 50us (>10us)
    digitalWrite(R_TriggerPin, LOW);          //end this high pulse
    Time_Echo_us = pulseIn(R_EchoPin, HIGH);   //calculate the pulse width
at EchoPin,
    if((Time_Echo_us < 60000) && (Time_Echo_us > 1)) //a valid pulse width should
be between (1, 60000).
    {
        RightDistance = (Time_Echo_us*34/100)/2; //calculate the distance by pulse
width, Len_mm = (Time_Echo_us * 0.34mm/us) / 2 (mm)
    }

```

```

//Distance
//if(FrontDistance < attack || BackDistance < attack || LeftDistance < attack ||
RightDistance < attack){
while(FrontDistance < attack || LeftDistance < attack){
  Serial.println("stop");
  MotorStop();
  delay(1000);

  //front
  digitalWrite(F_TrigPin, HIGH);          //begin to send a high pulse, then US-020
begin to measure the distance
  delayMicroseconds(50);                  //set this high pulse width as 50us (>10us)
  digitalWrite(F_TrigPin, LOW);           //end this high pulse
  Time_Echo_us = pulseIn(F_EchoPin, HIGH); //calculate the pulse width
at EchoPin,
  if((Time_Echo_us < 60000) && (Time_Echo_us > 1)) //a valid pulse width should
be between (1, 60000).
  {
    FrontDistance = (Time_Echo_us*34/100)/2; //calculate the distance by pulse
width, Len_mm = (Time_Echo_us * 0.34mm/us) / 2 (mm)
  }

  //left
  digitalWrite(L_TrigPin, HIGH);          //begin to send a high pulse, then US-020
begin to measure the distance
  delayMicroseconds(50);                  //set this high pulse width as 50us (>10us)
  digitalWrite(L_TrigPin, LOW);           //end this high pulse
  Time_Echo_us = pulseIn(L_EchoPin, HIGH); //calculate the pulse width
at EchoPin,
  if((Time_Echo_us < 60000) && (Time_Echo_us > 1)) //a valid pulse width should
be between (1, 60000).
  {
    LeftDistance = (Time_Echo_us*34/100)/2; //calculate the distance by pulse
width, Len_mm = (Time_Echo_us * 0.34mm/us) / 2 (mm)
  }
}
//face
}

```