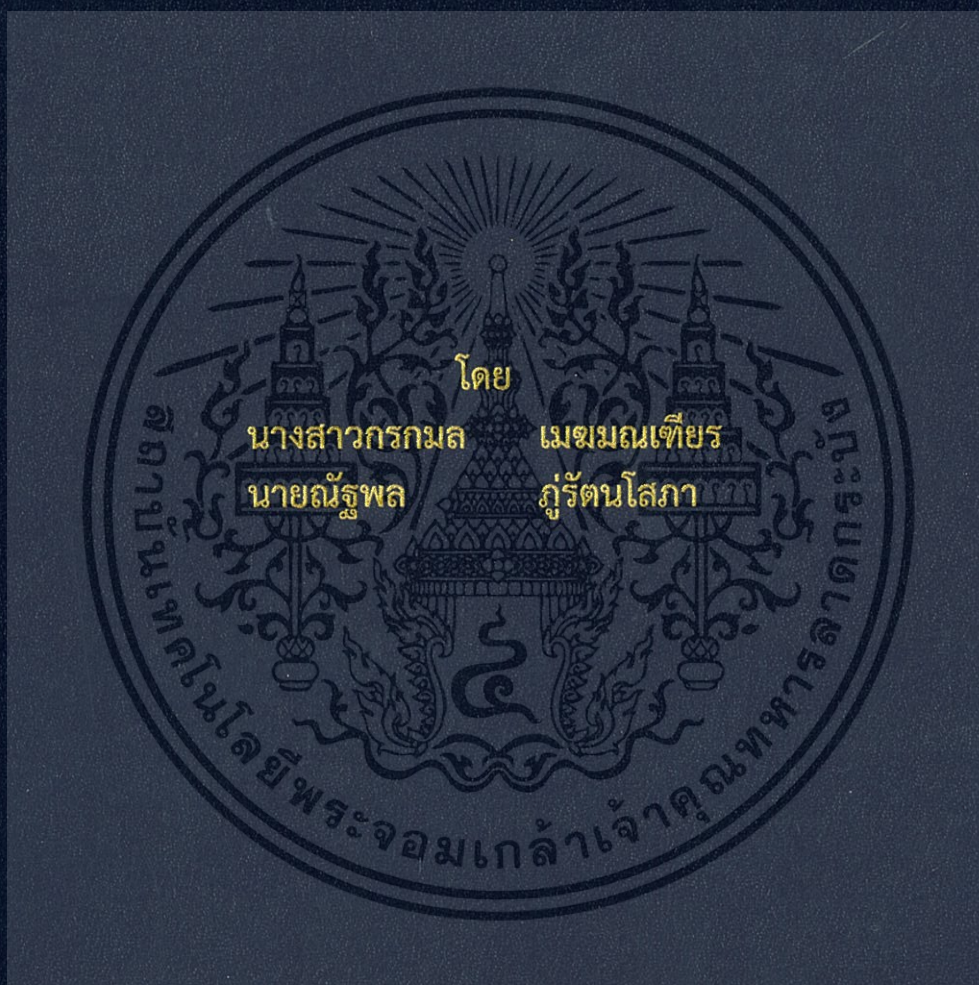


ระบบสั่งอาหารอัตโนมัติ
AUTOMATED FOOD ORDERING SYSTEM



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

ระบบสั่งอาหารอัตโนมัติ
AUTOMATED FOOD ORDERING SYSTEM



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบสั่งอาหารอัตโนมัติ
AUTOMATED FOOD ORDERING SYSTEM

โดย
นางสาวกรกมล เมฆมณเฑียร 56010011
นายณัฐพล ภูรัตนโสภา 56010416

อาจารย์ที่ปรึกษา
ผศ.ดร. กฤษณ์ วรุจิระ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

ผ่านการตรวจรูปเล่มแล้ว

ขอเสนอเป็นเอกสารฉบับร่างสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไป (.....) (.....)

อาจารย์ที่ปรึกษา

ผ่านการตรวจชิ้นงานแล้ว

กรรมการผู้ตรวจชิ้นงาน

19/5/60

ปริญญาานิพนธ์ปีการศึกษา 2559

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบสั่งอาหารอัตโนมัติ

AUTOMATED FOOD ORDERING SYSTEM

ผู้จัดทำ

1. นางสาวกรกมล เมฆมณฑีเยอร์ 56010011
2. นายณัฐพล ภูรัตนโสภา 56010416

..... อาจารย์ที่ปรึกษา
(ผศ.ดร. กฤษณ์ วงศ์จิระ)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความอนุเคราะห์อย่างยิ่งจากอาจารย์ที่ปรึกษา คือ ผศ.ดร. กฤษณ์ วรจิริระ ที่ให้คำแนะนำ คำสั่งสอน ให้ความรู้ ความเข้าใจ ตลอดระยะเวลาในการทำปริญญานิพนธ์นี้ ขอขอบพระคุณท่านในความห่วงใยและความหวังดีที่มีให้แก่ผู้จัดทำเป็นอย่างยิ่ง

ขอขอบคุณอาจารย์ ประจำสาขาวิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้อบรมสั่งสอนและประสิทธิ์ประสาทวิชาความรู้ให้แก่ผู้จัดทำ

ขอขอบคุณผู้มีส่วนเกี่ยวข้องทุกท่านที่คอยช่วยเหลือ ให้กำลังใจแก่ผู้จัดทำเสมอมา จนกระทั่งปริญญานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี หากมีข้อบกพร่องประการใด ผู้จัดทำขอน้อมรับ และยินดีที่จะรับฟังคำแนะนำจากทุกท่านที่ได้เข้ามาศึกษา เพื่อเป็นประโยชน์ในการพัฒนาต่อไป

นางสาวกรกมล เมฆมนเตียร
นายณัฐพล ภูรัตนโสภา

ผู้จัดทำ

ระบบสั่งอาหารอัตโนมัติ
AUTOMATED FOOD ORDERING SYSTEM

โดย นางสาวกรกมล เมฆมนเทียร 56010011
นายณัฐพล ภูรัตนโสภา 56010416

อาจารย์ที่ปรึกษา ผศ.ดร. กฤษณ์ วรจรัส

บทคัดย่อ

ความก้าวหน้าในเทคโนโลยีไร้สายที่ไม่เคยหยุดนิ่ง มีผลต่อการดำเนินชีวิตของเราในทุกๆ ด้าน ปริญญาพนธ์นี้จัดทำขึ้นเพื่อมุ่งมั่นสร้างระบบการสั่งอาหารภายในร้านอาหารให้เป็นแบบอัตโนมัติผ่านระบบเครือข่ายไร้สาย ในปริญญาพนธ์เล่มนี้กล่าวถึงการออกแบบและส่วนประกอบของระบบสั่งอาหารอัตโนมัติสำหรับร้านอาหาร ในส่วนที่ติดต่อและแสดงผลกับผู้ใช้งาน ซึ่งจะมีข้อมูลรายการอาหารทั้งหมดของทางร้าน และเชื่อมต่อกับเครื่องหยอดเหรียญ โดยข้อมูลการสั่งอาหารจากผู้ใช้งานจะถูกส่งไปเก็บไว้ในฐานข้อมูลของระบบ และทำข้อมูลให้เป็นปัจจุบัน จากนั้นข้อมูลการสั่งอาหารจะถูกส่งไปแสดงผลภายในห้องครัวตามลำดับ ผู้ประกอบการร้านอาหารสามารถตรวจสอบข้อมูลและสถิติการสั่งอาหารในแต่ละวันได้อย่างสะดวก โดยระบบจะทำให้ผู้ใช้งานได้รับความสะดวกสบาย อีกทั้งพัฒนาประสิทธิภาพและความถูกต้องแม่นยำให้กับร้านอาหารจากการประหยัดเวลา ลดขั้นตอนการให้บริการ และลดความผิดพลาดที่อาจเกิดขึ้นได้ ในการทำปริญญาพนธ์นี้ จัดทำเพื่อเป็นความรู้ให้กับผู้จัดทำ และหวังว่าจะเป็นประโยชน์ต่อผู้ที่สนใจอยากศึกษา

ABSTRACT

The growth of wireless technology is creating a great impact on our lives. This project work aims to automate the food ordering process in a restaurant via wireless technology. In this paper we discuss about the design and implementation of automated food ordering system for restaurants. The Graphic User Interface (GUI) part, controlled by Raspberry Pi, will show all the menu details and cooperate with the coin acceptor. The order details from customer's interface are wirelessly updated to MySQL database and subsequently sent to kitchen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

respectively. At the end of the day, the restaurant owner could check the statistical data from database easily. The system provided a means of convenience and also improved efficiency and accuracy for restaurant by saving time, reducing human errors. With this project, we could obtain and gain more knowledge. And we hope that it would be useful for people who were interested.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	VIII
บทที่ 1	บทนำ
	1
	1.1 ความเป็นมาและความสำคัญของปัญหา
	1
	1.2 วัตถุประสงค์
	1
	1.3 ขอบเขตของโครงการ
	1
บทที่ 2	ทฤษฎีและหลักการที่เกี่ยวข้อง
	2
	2.1 RASPBERRY PI
	2
	2.2 เครื่องรับเหรียญชนิดหลายเหรียญ
	5
	2.3 ภาษาซีพลัสพลัส (C++)
	7
	2.4 ภาษาพีเอชพี (PHP)
	8
	2.5 ภาษาเอชทีเอ็มแอล (HTML)
	9
	2.6 โปรแกรมฐานข้อมูล MYSQL
	10
	2.7 การสื่อสารข้อมูล
	13
	2.8 โปรแกรมควิที ครีเอเตอร์ (QT CREATOR)
	15
	2.9 หลักการ ทีซีพี/ไอพี ซ็อกเก็ต (TCP/IP SOCKET)
	15
	2.10 ARDUINO IDE
	16
	2.11 WEB SERVER
	17
บทที่ 3	การออกแบบและการจัดทำปริญญานิพนธ์
	22
	3.1 การออกแบบ
	22
	3.2 เครื่องมือที่ใช้ในการทดลอง
	28
	3.3 การจัดเก็บผลการทดลอง
	28

สารบัญ (ต่อ)

	หน้า
บทที่ 4	ผลการทดลอง 30
4.1	การทดสอบการเขียนโปรแกรมควบคุมการทำงานของพอร์ต GPIO ของ RASPBERRY PI 30
4.2	การทดสอบการทำงานของเครื่องรับเหรียญ 32
4.3	การทดสอบการทำงานของโปรแกรมในการควบคุมการทำงานของเครื่องรับเหรียญผ่าน GUI 34
4.4	การทดสอบการทำงานของไมโครคอนโทรลเลอร์ 34
4.5	การทดสอบการทำงานร่วมกันของไมโครคอนโทรลเลอร์ เครื่องรับเหรียญ และ RASPBERRY PI 36
4.6	ทดสอบการสร้างฐานข้อมูลสำหรับเก็บบันทึกข้อมูลการใช้งานระบบสั่งอาหารอัตโนมัติ 37
4.7	ทดสอบการแสดงผลการสรุปผลการสั่งอาหาร 42
4.8	การทดสอบการทำงานของส่วนติดต่อกับผู้ใช้งานหลังจากพัฒนาฟังก์ชัน 43
4.9	การทดสอบการร้องขอ (REQUEST) ข้อมูลจากฐานข้อมูลบนเซิร์ฟเวอร์ 45
4.10	การทดสอบการแจ้งเตือนผ่านหลอดไฟ LED เมื่อมีการสั่งอาหารเข้ามา 47
บทที่ 5	สรุปผลและข้อเสนอแนะ 50
5.1	สรุปผล 50
5.2	ข้อเสนอแนะ 50
บรรณานุกรม	51
ภาคผนวก ก	โค้ดควบคุมการทำงาน 52
ภาคผนวก ข	คุณลักษณะของอุปกรณ์ 74

สารบัญรูป

รูปที่	หน้า
2.1 RASPBERRY PI	2
2.2 ส่วนประกอบของ RASPBERRY PI	3
2.3 ขา GPIO ของ RASPBERRY PI	5
2.4 ข้อมูลขนาด 8 บิต บิตเริ่มต้นและบิตหยุดที่ใช้ในการโอนย้ายข้อมูลแบบ อนุกรม	14
2.5 STATIC WEBSITE	17
2.6 DYNAMIC WEBSITE	18
2.7 การเรียกใช้งานบริการเว็บไซต์ของผู้ใช้งาน	18
2.8 หลักการทำงานของโปรโตคอล HTTP	21
3.1 บล็อกไดอะแกรมของระบบสั่งอาหารอัตโนมัติ	22
3.2 แผนผังขั้นตอนการทำงานของบอร์ด RASPBERRY PI	24
3.3 แผนผังการทำงานของ NODEMCU โมดูล ESP8266	26
4.1 การเชื่อมต่อ LED เข้ากับ RASPBERRY PI	30
4.2 หน้าจอ INTERFACE โปรแกรม	31
4.3 ผลการทดสอบเมื่อทำการกดปุ่ม ON	31
4.4 ผลการทดสอบเมื่อทำการกดปุ่ม OFF	32
4.5 เครื่องรับเหรียญแสดงไฟสถานะเป็นสีเขียว	32
4.6 เครื่องรับเหรียญแสดงไฟสถานะเป็นสีแดง	33
4.7 เมื่อทำการส่งคำสั่ง ENABLE ไปยังเครื่องรับเหรียญ	33
4.8 หน้าจอ INTERFACE เมื่อทำการรับข้อมูลจากเครื่องรับเหรียญ	34
4.9 เมื่อทำการเชื่อมต่อกับไมโครคอนโทรลเลอร์	35
4.10 ไมโครคอนโทรลเลอร์ได้รับข้อมูลจาก RASPBERRY PI	35
4.11 หน้าจอ INTERFACE สำหรับส่งข้อมูล	36
4.12 RASPBERRY PI ได้รับข้อมูลจากเครื่องรับเหรียญจนครบ	36
4.13 ไมโครคอนโทรลเลอร์ได้รับข้อมูลจาก RASPBERRY PI และแสดงผลผ่าน จอแอลซีดี	37
4.14 การเข้าสู่ระบบของ PHPMYADMIN	38
4.15 เลือกรฐานข้อมูล	38
4.16 สร้างตารางสำหรับเก็บข้อมูล	39

สารบัญรูป (ต่อ)

รูปที่	หน้า	
4.17	โครงสร้างของระบบฐานข้อมูลสำหรับเก็บบันทึกข้อมูลการสั่งอาหาร	39
4.18	ทดสอบการเพิ่มข้อมูลเข้าไปในระบบ	39
4.19	ข้อมูลที่แสดงในระบบฐานข้อมูล	40
4.20	ไฟล์ข้อมูลที่ถูกบันทึกหลังจากการสั่งอาหาร	40
4.21	ทำการเรียกใช้โปรแกรมผ่านเว็บเบราว์เซอร์	41
4.22	พื้นที่เก็บไฟล์สำหรับเรียกใช้งานผ่านเว็บเบราว์เซอร์	42
4.23	การยืนยันตัวตนสำหรับเข้าถึงข้อมูล	42
4.24	กราฟแสดงผลสรุปการสั่งอาหารทั้งหมด	43
4.25	ส่วนติดต่อกับผู้ใช้งานหลังจากพัฒนาและปรับปรุงคุณสมบัติการทำงาน	44
4.26	ส่วนติดต่อกับผู้ใช้งานหลังจากทำการชำระราคาอาหารครบ	45
4.27	ฐานข้อมูลบนเซิร์ฟเวอร์ที่เก็บข้อมูลการสั่งอาหารไว้	46
4.28	SERIAL MONITOR แสดงผลการติดต่อระหว่าง NODEMCU กับ เซิร์ฟเวอร์	46
4.29	จอแอลซีดีแสดงผลข้อมูลรายการอาหาร	45
4.30	ส่วนแสดงผลฝั่งห้องครัวแจ้งเตือนว่ามีการสั่งอาหารเข้ามา	48
4.31	การแจ้งเตือนด้วยหลอดไฟ LED เมื่อมีการสั่งอาหารเข้ามา	48
4.32	หลอดไฟ LED ดับลงเมื่อไม่มีการสั่งอาหารเพิ่ม	49

สารบัญตาราง

ตารางที่		หน้า
2.1	ประเภทข้อมูลชนิดจำนวนเต็ม	11
2.2	ประเภทข้อมูลชนิดจำนวนทศนิยม	12
2.3	ประเภทข้อมูลสำหรับวันที่และเวลา	12
2.4	ประเภทข้อมูลสำหรับตัวอักษร	13



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันการติดต่อสื่อสารผ่านเทคโนโลยีไร้สายนั้นกว้างขวางและมีผลอย่างมากในการดำเนินชีวิตของเรา ด้วยความกว้างขวางและเข้าถึงง่ายของเครือข่ายนี้เองจึงสามารถนำไปต่อยอดใช้ประโยชน์ได้ในหลายๆ ด้าน สำหรับในด้านการอำนวยความสะดวก การให้บริการภายในร้านอาหารได้มีการพัฒนาขึ้นมาอย่างต่อเนื่อง เพื่อให้ลูกค้าหรือผู้มาใช้บริการเกิดความประทับใจ และตอบสนองความต้องการได้สะดวกและรวดเร็วที่สุด รวมไปถึงเกิดความผิดพลาดน้อยที่สุด จึงได้นำเอาประโยชน์ของเทคโนโลยีไร้สายมาประยุกต์ใช้กับการให้บริการภายในร้านอาหารเป็นระบบสั่งอาหารอัตโนมัติ ช่วยพัฒนาประสิทธิภาพและความถูกต้องสำหรับร้านอาหาร โดยช่วยประหยัดเวลา และลดความผิดพลาดที่อาจจะเกิดขึ้นได้โดยมนุษย์

1.2 วัตถุประสงค์

- 1) สร้างระบบการสั่งอาหารแบบอัตโนมัติภายในร้านอาหาร
- 2) เพื่อศึกษา การใช้งาน Raspberry Pi และการออกแบบ Graphical user interface (GUI)
- 3) นำเทคโนโลยีไร้สายมาประยุกต์ใช้ในกระบวนการสั่งอาหาร
- 4) ลดขั้นตอนและความผิดพลาดในการสั่งอาหาร
- 5) เพื่อพัฒนาระบบการจัดการร้านอาหาร

1.3 ขอบเขตของปริญญาณิพนธ์

- 1) ออกแบบและสร้างระบบการสั่งอาหารผ่านจอแสดงผล
- 2) รับ-ส่งข้อมูลระหว่างอุปกรณ์ ผ่าน Raspberry pi
- 3) Raspberry pi จะเก็บบันทึกข้อมูลการสั่งอาหารในแต่ละครั้งไว้ และทำการเพิ่มข้อมูลลงในฐานข้อมูล เมื่อผู้ใช้งานชำระเงินผ่านเครื่องรับเหรียญอัตโนมัติ (Coin Acceptor) ตามราคาอาหารครบถ้วนแล้ว
- 4) ข้อมูลการอาหารจะแสดงผลให้พ่อครัวทราบได้ โดยให้นำค่าจากฐานข้อมูลมาแสดงผลบนจอ LCD
- 5) เครื่องรับเหรียญอัตโนมัติไม่สามารถทอนเงินได้

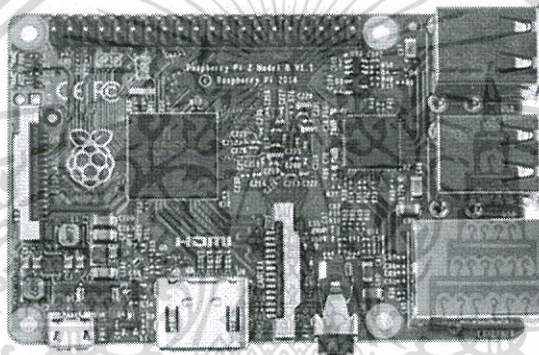
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 Raspberry Pi

Raspberry Pi คือบอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอแสดงผล คีย์บอร์ด และเมาส์ได้ รองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian) Pidora (Fedora) และ Arch Linux เป็นต้น โดยติดตั้งบน SD Card บอร์ด Raspberry Pi ถูกออกแบบมาให้มี CPU GPU และ RAM ภายในชิปเดียวกัน มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ ได้ ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 Raspberry Pi [2]

2.1.1 คุณสมบัติทางเทคนิคของบอร์ด Raspberry Pi

- ระบบควบคุมหลัก Broadcom BCM2835 ซึ่งรวมหน่วยประมวลผลกลาง หน่วยประมวลผลกราฟิก และหน่วยความจำ SDRAM ไว้ภายในตัวเดียวกัน

- หน่วยประมวลผลกลางหรือ CPU คือ Quad-core ARM Cortex-A7 ความเร็ว 900 MHz

- หน่วยประมวลผลกราฟิกหรือ GPU คือ Broadcom VideoCore IV 3D หน่วยความจำ SDRAM 512 MB

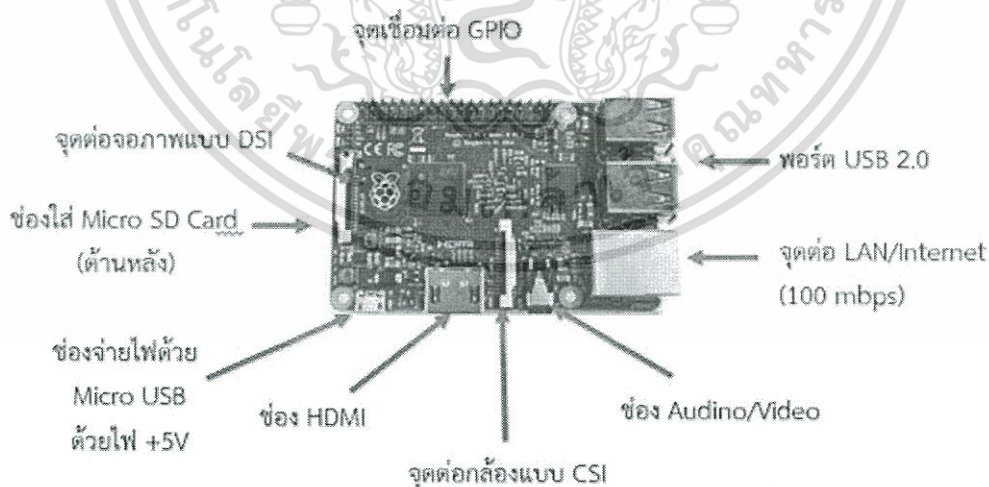
- จุดเชื่อมต่อ : USB 2.0 (4 พอร์ต), HDMI ช่องเอาต์พุตสัญญาณวีดีโอ, ช่องเอาต์พุตเสียงขนาด 3.5 มิลลิเมตร, คอนเน็คเตอร์หรือจุดต่อพอร์ตอินพุตเอาต์พุต (General Purpose Input/Output : GPIO) ที่มีขาต่อบัส SPI (Serial Peripheral Interface Bus) 40 จุด, พอร์ต Ethernet 10/100, ซ็อกเก็ตของ SD Card สำหรับเสียบ Micro SD Card ที่ติดตั้งระบบปฏิบัติการไว้

- ไฟเลี้ยงขนาด +5 V ที่ 700 mA เป็นอย่างน้อย
- ขนาดบอร์ด : 85.60 x 53.98 มิลลิเมตร หรือ 3.370 x 2.125 นิ้ว

2.1.2 ส่วนประกอบของบอร์ด

- HDMI Out (ใช้เชื่อมต่อพอร์ต HDMI เพื่อแสดงผลบนหน้าจอ)
- Ethernet Out (ใช้เชื่อมต่อพอร์ต RJ-45 เพื่อเชื่อมต่อกับอินเทอร์เน็ต)
- USB 2.0 (ใช้เชื่อมต่อสาย USB) 4 พอร์ต
- Audio Out (ใช้เชื่อมต่ออุปกรณ์เสียง)
- CSI Connector Camera (ใช้เชื่อมต่อแบบ CSI Raspberry Pi Camera Module)
- Broadcom quad-core ARM Cortex-A7 900 MHz
- SD Card Slot (ใช้เชื่อมต่อ SD Card)
- DSI Display Connector (ใช้เชื่อมต่อแบบ DSI สำหรับต่อกับจอแสดงผลเพิ่มเติม)
- GPIO Headers (ใช้เชื่อมต่อ GPIO)
- Status LED (สถานะไฟของ LED)

ซึ่งส่วนประกอบของ Raspberry Pi แสดงดังในรูปที่ 2.2



รูปที่ 2.2 ส่วนประกอบของ Raspberry Pi [2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 ส่วนประกอบของบอร์ด

การติดตั้งระบบปฏิบัติการให้ Raspberry Pi จะต้องทำการติดตั้งลงใน SD Card

- 1) ดาวน์โหลดอิมเมจ OS โดยเลือก Raspbian จะเป็นไฟล์ .img
- 2) ดาวน์โหลดโปรแกรม Win32DiskImager
- 3) ต่อ SD Card เข้ากับเครื่องคอมพิวเตอร์ แล้วเปิดโปรแกรม Win32 DiskImager ขึ้นมา โปรแกรมจะค้นหา Driver ของ SD Card ให้อัตโนมัติ
- 4) เลือกไฟล์ .img ของ Raspbian ที่ดาวน์โหลดไว้ เพื่อเขียนข้อมูลลงใน SD Card
- 5) เมื่อลงโปรแกรมเสร็จแล้ว ให้นำ SD Card ไปต่อกับ Raspberry Pi แล้วต่ออุปกรณ์ต่างๆ
- 6) เมื่อเริ่มใช้งาน Raspberry Pi ระบบจะขึ้นหน้าจออิน
Username : pi
Password : raspberry
(หมายเหตุ ใช้คำสั่ง startx เพื่อเข้าสู่หน้าจอ Desktop)

2.1.4 การใช้งานกับระบบเครือข่ายอินเทอร์เน็ต

เมื่อต้องการใช้งานระบบที่เกี่ยวข้องกับการใช้อินเทอร์เน็ต ให้ต่อสาย LAN RJ-45 เข้ากับตัว Raspberry Pi จะสามารถใช้งานได้ทันที โดยไม่ต้องทำการตั้งค่าใดๆ (ต้องมีการเปิดใช้งานจากผู้ให้บริการอินเทอร์เน็ตด้วย)

2.1.5 การจัดการขา GPIO

GPIO ย่อมาจาก General Purpose Input/Output ขา Pin เหล่านี้ มีไว้ใช้งานทั่วไป โดยควบคุมผ่านซอฟต์แวร์ ขา GPIO นั้น จึงสามารถใช้งานได้หลากหลายรูปแบบ ขึ้นอยู่กับว่าผู้ใช้งานต้องการที่จะใช้งานอะไร โดยรูปที่ 2.3 อธิบายการใช้งานขา GPIO

3.3V	○○	5V
2 SDA	○○	5V
3 SCL	○○	GND
4	○○	14 TXD
GND	○○	15 RXD
17	○○	18
27	○○	GND
22	○○	23
3.3V	○○	24
10 MOSI	○○	GND
9 MISO	○○	25
11 SCKL	○○	8
GND	○○	7

รูปที่ 2.3 ขา GPIO ของ Raspberry Pi [1]

2.2 เครื่องรับเหรียญชนิดหลายเหรียญ

เครื่องรับเหรียญ หรือ อุปกรณ์หยอดเหรียญชนิดหลายเหรียญของยี่ห้อ ICT รุ่น UCAE รับเหรียญได้หลายชนิด (Multi-Coin) ดังนี้ 1 บาทเก่า, 1 บาทใหม่, 2 บาทเงิน, 2 บาททอง, 5 บาท และ 10 บาท

หลักการทำงาน คือ เมื่อจ่ายไฟ 12 Vdc เข้าเครื่องรับเหรียญ CPU ของเครื่องรับเหรียญจะเริ่มทำงาน เมื่อมีการหยอดเหรียญ CPU จะเทียบกับค่าของเหรียญตัวอย่าง (Coin Comparator) ถ้าค่าตรงกันก็จะรับเหรียญที่หยอดลงด้านล่าง แล้วส่งสัญญาณออกไปที่ Coin Signal และ Counter Signal การสื่อสารกับเครื่องรับเหรียญจะอยู่ในรูปแบบเลขฐาน 16 มีรหัสอยู่ 2 ประเภทที่ใช้

(Format 1) without DATA : SYNC + LNG + CMD + EXT + CHECKSUM

(Format 2) with DATA : SYNC + LNG + CMD + DATA + EXT + CHECKSUM

ตัวอย่างเช่น ACK

SYNC	LNG	CMD	EXT	CHECKSUM
90h	05h	50h	03h	E8h

การทำงานของการสื่อสาร UCA ถูกแบ่งเป็น 3 ขั้นตอนดังนี้

2.2.1 การตรวจสอบเวอร์ชันของเครื่องรับเหรียญ (UCA)

การตรวจสอบเฟิร์มแวร์ คอมพิวเตอร์ที่ติดต่อสื่อสารด้วย จะต้องส่งรหัส 90h 05h 03h 03h 9Bh เครื่องรับเหรียญจะตอบกลับมาดังนี้

90h 1Bh 43h 41h 45h 23h 53h 39h 30h 30h 39h 49h 30h 30h 31h 32h 54h
48h 42h 23h 9Fh C6h 03h Ach

ซึ่งเป็นเวอร์ชันของเครื่องรับเหรียญ “UCA1#S9001I0012THB##”

2.2.2 การสั่งการทำงานของตัวรับเหรียญ (UCA)

2.2.2.1 การสั่งทำงาน ตัวรับเหรียญทำงาน (ENABLE)

คอมพิวเตอร์ที่ติดต่อกับเครื่องรับเหรียญจะต้องส่งคำสั่ง 90h 05h 01h 03h 99h ไปที่เครื่องรับเหรียญ จากนั้นเครื่องรับเหรียญจะตอบกลับมา ด้วยรหัส 90h 05h 50h 03h E8h แปลว่า เครื่องรับเหรียญทำงานตามคำสั่งแล้ว

2.2.2.2 การสั่งหยุดทำงาน ตัวรับเหรียญหยุดทำงาน (DISABLE)

คอมพิวเตอร์ที่ติดต่อกับเครื่องรับเหรียญจะต้องส่งคำสั่ง 90h 05h 02h 03h 9Ah ไปที่เครื่องรับเหรียญ จากนั้นเครื่องรับเหรียญจะตอบกลับมา ด้วยรหัส 90h 05h 50h 03h E8h แปลว่า เครื่องรับเหรียญทำงานตามคำสั่งแล้ว

2.2.3 การตรวจสอบค่าของเหรียญ

โดยการตรวจสอบค่าของเหรียญ เครื่องรับเหรียญจะมีหน้าที่ในการตรวจสอบค่าของเหรียญ แล้วส่งข้อมูลออกมาเป็นรหัส โดยมีโปรแกรมคอมพิวเตอร์เป็นตัวอ่านค่าจากรหัสที่ส่งมา

1 บาทเก่า	90h 06h 12h 01h 03h Ach
2 บาทเก่า	90h 06h 12h 02h 03h ADh
5 บาท	90h 06h 12h 03h 03h AEh
10 บาท	90h 06h 12h 04h 03h AFh
1 บาทใหม่	90h 06h 12h 06h 03h B1h
2 บาทใหม่	90h 06h 12h 05h 03h B0h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ภาษาซีพลัสพลัส (C++)

ซีพลัสพลัส (C++) เป็นภาษาคอมพิวเตอร์ พัฒนามาจากภาษาซี เพื่อวัตถุประสงค์ทั่วไป ซึ่งสามารถเขียนโปรแกรมได้ทั้งแบบออบเจกต์ (Object-Oriented Program) คือจะมีการรวมคุณสมบัติของโปรแกรมภายใต้ Class และการเขียนแบบปกติทั่วไป และยังมีเครื่องมืออำนวยความสะดวกในการจัดการและเข้าถึงระดับหน่วยความจำ

2.3.1 ข้อดีของภาษาซี (C) และซีพลัสพลัส (C++)

โปรแกรมเมอร์โดยทั่วไปในปัจจุบันนิยมพัฒนาโปรแกรมด้วยภาษาซี (C) และซีพลัสพลัส (C++) ด้วยเหตุผลดังนี้

- 1) โปรแกรมเมอร์สามารถสร้างโปรแกรมที่ควบคุมการทำงานของคอมพิวเตอร์ และการโต้ตอบระหว่างผู้ใช้กับคอมพิวเตอร์ ได้อย่างเต็มประสิทธิภาพ เช่น การเขียนโปรแกรมในลักษณะที่ผู้ใช้ควบคุมโปรแกรมในสภาพแวดล้อมที่เป็น Event-Driven คือ ผู้ใช้สามารถควบคุมเหตุการณ์ต่างๆ ของโปรแกรมในขณะที่ทำงานได้ ไม่ใช่ผู้ใช้ถูกควบคุมโดยโปรแกรม ลักษณะการทำงานแบบ Event-Driven ได้แก่ โปรแกรมที่ทำงานในสภาพแวดล้อมภายใต้ระบบปฏิบัติการวินโดวส์ เป็นต้น
- 2) ภาษาซี และซีพลัสพลัส มีประสิทธิภาพของภาษาอยู่ในระดับที่ใกล้เคียงกับภาษา Assembly มากที่สุด แต่มีความยืดหยุ่นไม่ยึดติดกับฮาร์ดแวร์คอมพิวเตอร์ หรือ Microprocessor รุ่นใดรุ่นหนึ่งทำให้สามารถนำโปรแกรมที่สร้างขึ้นไปทำงานได้กับเครื่องคอมพิวเตอร์ได้ทุกรุ่น
- 3) ภาษาซีพลัสพลัส สนับสนุนการเขียนโปรแกรมในลักษณะเชิงวัตถุ หรือ OOP (Object Oriented Programming) ซึ่งเป็นเทคนิคการเขียนโปรแกรมที่นิยมใช้เขียนโปรแกรมขนาดใหญ่ที่มีจำนวนข้อมูลในโปรแกรมมาก
- 4) โปรแกรมเมอร์ส่วนใหญ่จะนิยมใช้ภาษาซี และซีพลัสพลัส พัฒนาโปรแกรมประยุกต์ในงานด้านต่างๆ เป็นจำนวนมากในปัจจุบัน เพราะประสิทธิภาพของภาษาที่ได้เปรียบกว่าภาษาอื่นๆ

2.4 ภาษาพีเอชพี

พีเอชพี (PHP) ย่อมาจาก PHP Hypertext Preprocessor แต่เดิมย่อมาจาก Personal Home Page Tools พีเอชพีคือภาษาคอมพิวเตอร์จำพวก scripting language ภาษาจำพวกนี้ คำสั่งต่างๆ จะเก็บอยู่ในไฟล์ที่เรียกว่า สคริปต์ (Script) และเวลาใช้งานต้องอาศัยตัวแปรชุดคำสั่ง ลักษณะของภาษาพีเอชพี ที่แตกต่างจากภาษาสคริปต์แบบอื่นๆ คือพีเอชพีได้รับการพัฒนาและออกแบบมา เพื่อใช้งานในการสร้างเอกสารแบบเอชทีเอ็มแอล (HTML) โดยสามารถสอดแทรกหรือแก้ไขเนื้อหาได้โดยอัตโนมัติ ดังนั้นจึงกล่าวว่า พีเอชพีเป็นภาษาที่เรียกว่า server-side หรือ HTML-embedded scripting language คือในทุกๆ ครั้งก่อนที่เครื่องคอมพิวเตอร์ซึ่งให้บริการเป็น Web server จะส่งหน้าเว็บเพจที่เขียนด้วยภาษาพีเอชพีออกมา มันจะทำการประมวลผลตามคำสั่งที่มีอยู่ให้เสร็จก่อน แล้วจึงค่อยส่งผลลัพธ์ที่ได้ออกมา ผลลัพธ์ที่ได้ นั่นคือเว็บเพจ พีเอชพีจึงเป็นเครื่องมือสำคัญชนิดหนึ่งซึ่งช่วยให้สามารถสร้างเว็บเพจที่มีการโต้ตอบกับผู้ใช้ (Dynamic Web pages) ได้อย่างมีประสิทธิภาพและมีลูกเล่นมากขึ้น

พีเอชพี เป็นผลงานจากกลุ่มนักพัฒนาในเชิงเปิดเผยรหัสต้นฉบับหรือ Open Source ดังนั้นพีเอชพีจึงมีการพัฒนาไปอย่างรวดเร็ว และแพร่หลาย โดยเฉพาะอย่างยิ่งเมื่อใช้ร่วมกับ Apache Web server ระบบปฏิบัติการอย่างเช่นลินุกซ์ (Linux) หรือ FreeBSD เป็นต้น ในปัจจุบันพีเอชพีสามารถใช้ร่วมกับ Web Server หลายๆ ตัว บนระบบปฏิบัติการอย่างเช่น Windows 95/98/NT เป็นต้น

2.4.1 คุณสมบัติภาษาพีเอชพี

การแสดงผลของภาษาพีเอชพีจะปรากฏในลักษณะเอชทีเอ็มแอล (HTML) ซึ่งจะไม่แสดงคำสั่งที่ผู้ใช้เขียน ซึ่งเป็นลักษณะเด่นที่พีเอชพีแตกต่างจากภาษาในลักษณะไคลเอนต์-ไซด์ สคริปต์ เช่น ภาษาจาวาสคริปต์ ที่ผู้ชมเว็บไซต์สามารถอ่าน ดู และคัดลอกคำสั่งไปใช้เองได้ นอกจากนี้พีเอชพียังเป็นภาษาที่เรียนรู้และเริ่มต้นได้ไม่ยาก โดยมีเครื่องมือช่วยเหลือและคู่มือที่สามารถอ่านได้บนอินเทอร์เน็ต ความสามารถในการประมวลผลหลักของพีเอชพี ได้แก่ การสร้างเนื้อหาอัตโนมัติ จัดการคำสั่ง การอ่านข้อมูลจากผู้ใช้และประมวลผล การอ่านข้อมูลจากฐานข้อมูล (Database) ความสามารถจัดการกับคุกกี้ (Cookies) ซึ่งทำงานเช่นเดียวกับโปรแกรมลักษณะ CGI คุณสมบัติอื่นเช่น การประมวลผลตามบรรทัดคำสั่ง (command line scripting) ทำให้ผู้เขียนโปรแกรมสร้างสคริปต์พีเอชพี ทำงานผ่าน PHP parser โดยไม่ต้องผ่านเชิร์ฟเวอร์ หรือเบราวเซอร์ ซึ่งมีลักษณะเหมือนกับ Cron (ในยูนิกซ์หรือลินุกซ์) หรือ Task Scheduler (ในวินโดวส์) สคริปต์เหล่านี้สามารถนำไปใช้ในแบบ Simple text processing tasks ได้

การแสดงผลของพีเอชพี ถึงแม้ว่าจุดประสงค์หลักใช้ในการแสดงผลเอชทีเอ็มแอล (HTML) แต่ยังสามารถสร้างเอ็กซ์เอชทีเอ็มแอล (XHTML) หรือ เอ็กซ์เอ็มแอล (XML) ได้ นอกจากนี้สามารถทำงานร่วมกับคำสั่งเสริมต่างๆ ซึ่งสามารถแสดงผลข้อมูลหลัก PDF แฟลช (โดยใช้ libswf และ Ming) พีเอชพีมีความสามารถอย่างมากในการทำงานเป็นตัวประมวลผลข้อความจาก POSIX Extended หรือรูปแบบ Perl ทั่วไป เพื่อแปลงเป็นเอกสาร XML ในการแปลง และเข้าสู่เอกสาร XML

2.4.2 พีเอชพีมายแอตมิน (phpMyAdmin)

เป็นสคริปต์ติดต่อฐานข้อมูลที่สร้างโดยภาษาพีเอชพี ซึ่งใช้จัดการฐานข้อมูลมายเอสคิวแอล ผ่านเว็บเบราว์เซอร์ โดยสามารถที่จะทำการสร้างฐานข้อมูลใหม่หรือทำการสร้างตารางใหม่ และยังมีฟังก์ชันที่ใช้สำหรับการทดสอบการสืบค้นข้อมูลด้วยภาษาเอสคิวแอล และยังสามารถทำการ insert delete update หรือแม้กระทั่งใช้คำสั่งต่างๆ เหมือนกับการใช้ภาษาเอสคิวแอล ในการสร้างตารางข้อมูล

2.5 ภาษาเอชทีเอ็มแอล (HTML)

ภาษา HTML (HyperText Markup Language) เป็นภาษาหลักที่ใช้ในการสร้างเว็บเพจ (Web Page) เป็นภาษาประเภท Markup Language เกิดขึ้นจากการพัฒนาระบบ World Wide Web ในเดือนมีนาคม 1989 โดยนักวิจัยจากสถาบัน CERN (Conseil European Pour La Recherche Nucleaire) ซึ่งเป็นห้องทดลองในเมืองเจนีวา ประเทศสวิตเซอร์แลนด์ ชื่อ ทิมเบอร์เนอร์ – ลี (Tim Berners – Lee) ซึ่งทิมได้นำแนวความคิด ในเรื่อง Hypertext ของ Vannevar Bush และ Ted Nelson มาใช้เพื่อกระจายข้อมูลในองค์กร ต่อมามีการพัฒนาและกำหนดมาตรฐานโดยองค์กรที่ชื่อว่า W3C (World Wide Web Consortium) ภาษา HTML เป็นภาษาที่มีลักษณะของข้อมูลที่เป็นตัวอักษรในมาตรฐานของรหัสแอสกี (ASCII Code) โดยเขียนอยู่ในรูปของเอกสารข้อความ (Text Document) จึงกำหนดรูปแบบและโครงสร้างได้ง่าย ภาษา HTML ได้ถูกพัฒนาขึ้นอย่างต่อเนื่องตั้งแต่ HTML Level1 (รุ่นดั้งเดิม), HTML 2.0, HTML 3.0, HTML 3.2 และ HTML 4.0 ซึ่งเป็นรุ่นที่นิยมเขียนกันในปัจจุบัน (ขณะนี้ W3C ได้พัฒนา HTML 4.01 ออกมาแล้ว เพื่อรองรับมาตรฐานภาษา XML) จึงทำให้ภาษา HTML ในปัจจุบันสามารถแสดงภาพทางกราฟิกและรับเสียงได้ เพื่อตอบสนองการทำงานในปัจจุบัน ภาษา HTML สามารถสร้างขึ้นได้จากโปรแกรมสร้างไฟล์ข้อความ (Text Editor) ทั่วๆ ไป เช่น Notepad หรือ Word Processing ได้ อีกทั้งง่ายต่อการเรียนรู้ เพราะภาษา HTML ไม่มีโครงสร้างความเป็น Programming เลยแม้แต่หน่อย และไฟล์ที่ได้จากการสร้างเอกสาร HTML ยังมีขนาดเล็กอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นามสกุลของไฟล์ HTML จะเป็นไฟล์นามสกุล .htm หรือ .html ซึ่งใช้ในทั้งระบบปฏิบัติการยูนิกซ์ (UNIX) และระบบปฏิบัติการวินโดวส์ และเรียกใช้งานได้จากเว็บเบราว์เซอร์ (Web Browser) เช่น Internet Explorer หรือ NetScape

- Tag

Tag เป็นลักษณะเฉพาะของภาษา HTML ใช้ในการระบุรูปแบบคำสั่ง หรือการลงรหัสคำสั่ง HTML ภายในเครื่องหมาย less-than bracket (<) และ greater-than bracket (>) โดยที่ Tag HTML แบ่งได้ 2 ลักษณะคือ Tag เดี่ยว เป็น Tag ที่ไม่ต้องมีการปิดรหัส เช่น <P>,
, , <HR> เป็นต้น Tag เปิด/ปิด เป็น Tag ที่ประกอบด้วย Tag เปิด และ Tag ปิด โดย Tag ปิด จะมีเครื่องหมาย slash (/) นำหน้าคำสั่งใน Tag นั้นๆ เช่น ..., <BLINK>...</BLINK> เป็นต้น

- Attributes

Attributes เป็นส่วนขยายความสามารถของ Tag จะต้องใส่ภายในเครื่องหมาย <> ในส่วน Tag เปิดเท่านั้น Tag คำสั่ง HTML แต่ละคำสั่งจะมี Attribute แตกต่างกันไป และมีจำนวนไม่เท่ากัน การระบุ Attribute มากกว่า 1 Attribute ให้ใช้ช่องเป็นตัวคั่น

2.6 โปรแกรมฐานข้อมูล MySQL

MySQL (มายเอสคิวแอล) เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) โดยใช้ภาษา SQL แม้ว่า MySQL เป็นซอฟต์แวร์โอเพนซอร์ส แต่แตกต่างจากซอฟต์แวร์โอเพนซอร์สทั่วไป โดยมีการพัฒนาภายใต้บริษัท MySQL AB ในประเทศสวีเดน โดยจัดการ MySQL ทั้งในแบบที่ให้ฟรี และแบบที่ใช้ในเชิงธุรกิจ

2.6.1 คุณสมบัติของโปรแกรมฐานข้อมูล MySQL

MySQL จัดเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS : Relational Database Management System) ซึ่งเป็นที่นิยมใช้กันมากในปัจจุบัน โดยเฉพาะอย่างยิ่งในโลกของอินเทอร์เน็ต เนื่องจาก

- 1.) MySQL เป็นฟรีแวร์ทางด้านฐานข้อมูลที่มีประสิทธิภาพสูง
- 2.) นักพัฒนาฐานข้อมูลที่เคยใช้ MySQL ต่างยอมรับในความรวดเร็ว การรองรับผู้ใช้และขนาดของข้อมูลจำนวนมาก
- 3.) สนับสนุนการใช้งานบนระบบปฏิบัติการมากมาย เช่น UNIX OS/2 MAC OS Windows

4.) สามารถใช้งานร่วมกับ Web Development platform เช่น C, C++, JAVA, Perl, PHP, Python, TCL, หรือ ASP

5.) ได้รับความนิยมอย่างมากในปัจจุบัน และมีแนวโน้มสูงขึ้นเรื่อยๆ

2.6.2 ความสามารถของโปรแกรมฐานข้อมูล MySQL

โปรแกรม MySQL เป็นโปรแกรมระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) หรือเรียกง่ายๆ ก็คือ Database Server โปรแกรม MySQL สามารถการทำงานบนระบบปฏิบัติการ Linux หรือ Windows และที่เด่นคือ โปรแกรม MySQL เปิดให้ใช้งานฟรี เพราะเป็นโปรแกรมประเภท Open Source ซึ่งมีลิขสิทธิ์แบบ GPL

สามารถสรุปความสามารถของโปรแกรมฐานข้อมูล MySQL ในการจัดการต่างๆ เกี่ยวกับฐานข้อมูลได้ ดังนี้

- 1.) ระบบจัดการบัญชีผู้ใช้
- 2.) สิทธิต่างๆ ในการเข้าใช้ฐานข้อมูล
- 3.) ระบบสำรองข้อมูล (Backup)
- 4.) ระบบคืนสภาพข้อมูล (Recovery)
- 5.) ระบบโอนถ่ายข้อมูลไปยังโปรแกรมฐานข้อมูลตัวอื่นๆ
- 6.) จัดเก็บข้อมูลได้หลายชนิดข้อมูล เช่น รูปภาพ ข้อความ ตัวเลข ฯลฯ

2.6.3 ประเภทของข้อมูลใน MySQL

2.6.3.1 ประเภทข้อมูลสำหรับตัวเลข

ไว้สำหรับเก็บข้อมูลตัวเลข ซึ่งอาจใช้ในการคำนวณหรือการจัดเรียงเปรียบเทียบกันในฟิลดนั้นๆ แบ่งออกเป็น จำนวนเต็ม จำนวนทศนิยม และจำนวนจริง

ตารางที่ 2.1 ประเภทข้อมูลชนิดจำนวนเต็ม

ลำดับ	ชื่อประเภทข้อมูล	แบบคิดเครื่องหมาย	แบบไม่คิดเครื่องหมาย	เนื้อที่
1	TINYINT(M)	-128 ถึง 127	0 ถึง 255	1 byte
2	SMALLINT(M)	-32768 ถึง 32767	0 ถึง 65535	2 byte
3	MEDIUMINT(M)	-8388608 ถึง 8388607	0 ถึง 16777215	3 byte
4	INT(M) หรือ INTEGER(M)	-2147483648 ถึง 2147483647	0 ถึง 4294967295.	4 byte
5	BIGINT(M)	-9223372036854775808 ถึง 9223372036854775807	0 ถึง 18446744073709551615	8 byte

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 ประเภทข้อมูลชนิดจำนวนทศนิยม

ลำดับ	ชื่อประเภทข้อมูล	แบบคิดเครื่องหมาย	แบบไม่คิดเครื่องหมาย	เนื้อที่
1	FLOAT(M,D) ค่า M เป็นจำนวนหลักที่ต้องการแสดงผล และค่า D คือจำนวนหลังจุดทศนิยม	-3.402823466E+38 ถึง -1.175494351E-38	0 และ 1.175494351E-38 ถึง .402823466E+38	4 byte
2	DOUBLE (M,D)	-1.7976931348623157E+308 ถึง -2.2250738585072014E-308	0 และ 2.2250738585072014E-308 ถึง 1.7976931348623157E+308	8 byte

2.6.3.2 ประเภทข้อมูลสำหรับวันที่และเวลา

ตารางที่ 2.3 ประเภทข้อมูลสำหรับวันที่และเวลา

ลำดับ	ชื่อประเภทข้อมูล	รายละเอียด	เนื้อที่
1	DATE	ข้อมูลชนิดวันที่ ตั้งแต่วันที่ 1 มกราคม ค.ศ. 1000 ถึง 31 ธันวาคม ค.ศ. 9999 การแสดงผลวันที่อยู่ในรูป 'YYYY-MM-DD'	3 byte
2	DATETIME	ข้อมูลชนิดวันที่และเวลา ตั้งแต่วันที่ 1 มกราคม ค.ศ. 1000 เวลา 00:00:00 ถึง 31 ธันวาคม ค.ศ. 9999 เวลา 23:59:59 การแสดงผลวันที่และเวลาอยู่ในรูปแบบ 'YYYY-MM-DD HH:MM:SS'	8 byte
3	TIME	ข้อมูลประเภทเวลา สามารถเป็นได้ตั้งแต่ '-838:59:59' ถึง '838:59:59' แสดงผลในรูปแบบ HH:MM:SS	3 byte
4	YEAR(2/4)	ข้อมูลประเภทปี ค.ศ. โดยสามารถเลือกว่าจะใช้แบบ 2 หรือ 4 หลัก ถ้าเป็น 2 หลัก จะใช้ได้ตั้งแต่ปี ค.ศ. 1901 ถึง 2155 ถ้าเป็น 4 หลัก จะใช้ได้ตั้งแต่ปี ค.ศ. 1970 ถึง 2069	1 byte

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.3.2 ประเภทข้อมูลสำหรับวันที่และเวลา

ตารางที่ 2.4 ประเภทข้อมูลสำหรับตัวอักษร

ลำดับ	ชื่อประเภทข้อมูล	รายละเอียด	เนื้อที่
1	CHAR(M)	เป็นข้อมูลสตริงที่จำกัดความกว้าง ไม่สามารถปรับขนาดได้ ขนาดความกว้างเป็นได้ตั้งแต่ 1 ถึง 255 ตัวอักษร	ตามจำนวน ตัวอักษรที่ระบุ
2	VARCHAR(M)	คล้ายกับแบบ CHAR(M) แต่สามารถปรับขนาดตามข้อมูลที่เก็บในฟิลด์ได้ ความกว้างเป็นได้ตั้งแต่ 1 ถึง 255 ตัวอักษร	ขนาดข้อมูล จริง + 1 byte
3	TINYTEXT	เป็น text ที่ความกว้างเป็นได้สูงสุด 255 ตัวอักษร	ขนาดข้อมูล จริง + 1 byte
4	TEXT	เป็น text ที่ความกว้างเป็นได้สูงสุด 65,535 ตัวอักษร	ขนาดข้อมูล จริง + 2 byte
5	MEDIUMTEXT	เป็น text ที่ความกว้างเป็นได้สูงสุด 16,777,215 ตัวอักษร	ขนาดข้อมูล จริง + 3 byte
6	LONGTEXT	เป็น text ที่ความกว้างเป็นได้สูงสุด 4,294,967,295 ตัวอักษร	ขนาดข้อมูล จริง + 4 byte
7	ENUM	เป็นข้อมูลประเภทระบุเฉพาะค่าที่ต้องการ หรือถ้าไม่มีจะให้ เป็นค่า NULL สามารถกำหนดค่าได้ถึง 65,535 ค่า	ตามจำนวน ตัวอักษรที่ระบุ
8	SET('value1', 'value2', ...)	เป็นข้อมูลประเภทเซต ประกอบด้วยข้อมูลที่ไม่มีค่าหรือมีค่า ตามสมาชิกที่กำหนด มีจำนวนสมาชิกได้ 64 ตัว	

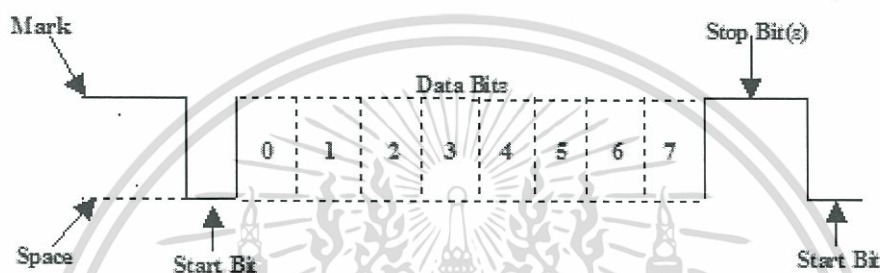
2.7 การสื่อสารข้อมูล

การสื่อสารข้อมูลคือ การส่งข้อมูลที่ทำการเข้ารหัสแล้วระหว่างอุปกรณ์ 2 ชนิด โดยมีองค์ประกอบหลักสำคัญอยู่ 3 ส่วน คือ อุปกรณ์ฝั่งส่ง อุปกรณ์ฝั่งรับ และตัวกลาง ซึ่งตัวกลางจะสามารถแบ่งได้เป็น 2 ประเภท ได้แก่ แบบมีสาย และแบบไร้สาย โดยจะสามารถแบ่งแยก รูปแบบในการสื่อสารข้อมูลได้เป็น 3 แบบ คือ การสื่อสารทางเดียว การสื่อสารทั้งสองทาง และการสื่อสารสองทาง

2.7.1 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมเป็นการสื่อสารที่มีการรับส่งข้อมูลที่ละบิต เรียงต่อกันไปจนสิ้นสุด ซึ่งการสื่อสารแบบนี้จะแตกต่างจากการสื่อสารแบบขนาน เนื่องจาก การสื่อสารแบบขนานนั้น ถ้าจำนวนข้อมูลมากขึ้นก็ต้องเพิ่มจำนวนสายสัญญาณ จึงไม่มีความเหมาะสมในการใช้งานที่มีระยะทางในการรับส่งข้อมูลไกลๆ ซึ่งเป็นการสิ้นเปลืองค่าใช้จ่ายเป็นอย่างมาก แต่การสื่อสารแบบอนุกรมนั้นใช้จำนวนสายสัญญาณเพียง 2 ถึง 3 เส้นเท่านั้น จึงทำให้ใช้เวลาในการรับส่งข้อมูลมากกว่าการสื่อสารแบบขนานแต่ค่าใช้จ่ายลดลง โดยการสื่อสาร

แบบอนุกรมนั้นจะสามารถแบ่งแยกลักษณะของการสื่อสารได้เป็น 2 ประเภท คือ การสื่อสารแบบซิงโครนัส (Synchronous) และการสื่อสารแบบอะซิงโครนัส (Asynchronous) โดยการสื่อสารแบบซิงโครนัสนั้น จำเป็นจะต้องอาศัยสัญญาณนาฬิกามาช่วยกำหนดจังหวะในการสื่อสาร ส่วนการสื่อสารแบบอะซิงโครนัสจะใช้รูปแบบในการส่งข้อมูล (Bit Pattern) เป็นตัวกำหนดว่าข้อมูลส่วนไหนเป็นข้อมูลอะไร ซึ่งรูปแบบโดยทั่วไปจะเป็นดังนี้ คือ เริ่มต้นด้วยบิตเริ่มต้น (Start Bit) ตามด้วยข้อมูล (Data) ตามด้วยบิตพาริตี (Parity Bit) สิ้นสุดด้วยบิตสิ้นสุด (Stop Bit) ดังแสดงในรูปที่ 2.4 โดยการส่งข้อมูลแบบอะซิงโครนัสนั้นจะมีอุปกรณ์ที่เรียกว่า ยูอาร์ท (UART : Universal Asynchronous Receiver/Transmitter) เป็นตัวควบคุมในการรับส่ง



รูปที่ 2.4 ข้อมูลขนาด 8 บิต บิตเริ่มต้นและบิตหยุดที่ใช้ในการโอนย้ายข้อมูลแบบอนุกรม [9]

UART (Universal Asynchronous Receiver Transmitter) หมายถึง อุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัส โดยหน้าที่หลักของ UART คือทำหน้าที่แปลงข้อมูลที่อยู่ในรูปแบบขนานจากคอมพิวเตอร์ ให้อยู่ในรูปแบบอนุกรมแบบอะซิงโครนัส แล้วส่งออกไป และทำหน้าที่แปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนที่จะส่งเข้าคอมพิวเตอร์ ซึ่งนอกจาก UART จะส่งข้อมูลไปยังคอมพิวเตอร์แล้วยังทำการแจ้งข้อมูลอื่นๆ ให้คอมพิวเตอร์ทราบด้วย เช่น อัตราความเร็วในการรับส่งข้อมูล (บอดเรต), รูปแบบการส่งข้อมูล, ความผิดพลาดที่เกิดขึ้นระหว่างการถ่ายทอดข้อมูล (ผิดพลาดจากพาริตี, เพรมข้อมูล, โอเวอร์รัน) เป็นต้น

ในการสื่อสารแบบอนุกรม ข้อมูลที่ใช้ในการสื่อสารจะอยู่ในลักษณะของกลุ่มบิตข้อมูล (Bit Stream) จำเป็นจะต้องสนใจในเรื่องอัตราเร็วในการรับส่งข้อมูล โดยจะใช้การกำหนดอัตราบอดเรต (Baud Rate) ในการกำหนด ซึ่งจะมีค่ามาตรฐาน ได้แก่ 110 150 300 1200 2400 4800 9600 19200 เป็นต้น

2.7.2 มาตรฐานอาร์เอส (RS-232)

ในปัจจุบันมีอุปกรณ์ต่างๆ มากมาย จากผู้ผลิตหลายค่าย จึงจำเป็นต้องมีตัวกำหนดมาตรฐาน เพื่อให้อุปกรณ์ต่างๆ สามารถสื่อสารกันได้ โดยมาตรฐานที่นิยมใช้กันอย่างแพร่หลายนั้นก็คือ มาตรฐานอาร์เอส 232 (RS-232 : Recommended Standard 232) ซึ่งจะมีระดับแรงดันอยู่ในช่วง -15 ถึง 15 โวลต์ โดยลอจิก “0” จะมีแรงดันอยู่ในช่วง 3 ถึง 15

โวลต์ และลอจิก “1” จะมีแรงดันอยู่ในช่วง -15 ถึง -3 โวลต์ ซึ่งอาร์เอส 232 นั้น มีสายสัญญาณ 3 เส้น คือ สายส่งข้อมูล (Tx) สายรับข้อมูล (Rx) และกราวด์ (Gnd)

2.7.3 ทีทีแอล

ทีทีแอล (TTL : Transistor – Transistor Logic) เป็นการกำหนดระดับแรงดันที่เกิดขึ้นในยุคแรกๆ ใช้ในการสื่อสารระหว่างทรานซิสเตอร์ โดยมีช่วงระดับแรงดันอยู่ที่ 0 ถึง 5 โวลต์ และในปัจจุบันมีการพัฒนาระดับแรงดันทีทีแอลแบบทีทีแอลแรงดันต่ำ (LVTTTL : Low-Voltage TTL) ซึ่งมีระดับแรงดันอยู่ที่ 0 ถึง 3.3 โวลต์

2.8 โปรแกรมควิที ครีเอเตอร์ (QT Creator)

ควิทีเป็นเฟรมเวิร์กสำหรับสร้างแอปพลิเคชัน และส่วนติดต่อผู้ใช้งานที่ครอสแพลตฟอร์ม (Cross-Platform) โดยเขียนเพียงครั้งเดียว แต่สามารถนำไปปรับใช้บนระบบปฏิบัติการ และเครื่องมือต่างๆ ได้มากมาย เช่น บนคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการ Windows, Mac, Linux, โทรศัพท์มือถือที่ใช้ Symbian หรือแท็บเล็ต โดยตัวควิทีนั้นเป็นเฟรมเวิร์กสำหรับแอปพลิเคชันที่พัฒนาด้วยภาษาซีพลัสพลัส (C++) เป็นหลัก

2.9 หลักการ ทีซีพี/ไอพี ซ็อกเก็ต (TCP/IP Socket)

TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นชุดของโปรโตคอลที่ใช้ในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต โดยมีวัตถุประสงค์เพื่อให้สามารถใช้สื่อสารจากต้นทางข้ามเครือข่ายไปยังปลายทางได้ และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้เองโดยอัตโนมัติ ถึงแม้ว่าในระหว่างทางอาจจะผ่านเครือข่ายที่มีปัญหา โปรโตคอลก็ยังค้นหาเส้นทางอื่นในการส่งผ่านข้อมูลไปให้ถึงปลายทางได้

ชุดโปรโตคอลนี้ได้รับการพัฒนามาตั้งแต่ปี 1960 ซึ่งถูกใช้เป็นครั้งแรกในเครือข่าย ARPANET ซึ่งต่อมาได้ขยายการเชื่อมต่อไปทั่วโลกเป็นเครือข่ายอินเทอร์เน็ต ทำให้ทีซีพี/ไอพีเป็นที่ยอมรับอย่างกว้างขวางจนถึงปัจจุบัน

ซ็อกเก็ต (Socket) คือการเชื่อมต่อสื่อสารระหว่างจุดสองจุด (เครื่องสองเครื่อง) แบบไปกลับได้ระหว่างโปรแกรมสองโปรแกรม (server กับ client) ภายในเครือข่ายเดียวกัน

2.9.1 โหมดเซิร์ฟเวอร์ (Server mode)

การทำงานในโหมดเซิร์ฟเวอร์ (Server mode) คือ เครื่องคอมพิวเตอร์ หรืออุปกรณ์ที่ทำหน้าที่เป็นเซิร์ฟเวอร์ จะทำการเปิดการเชื่อมต่อ (Connection) รอไว้ที่หมายเลขพอร์ตใดพอร์ตหนึ่ง หรือหลายพอร์ตก็ได้ เพื่อรอการร้องขอและเชื่อมต่อจากคอมพิวเตอร์ หรืออุปกรณ์อื่นๆ ที่ทำหน้าที่เป็นไคลเอนต์ (Client)

2.9.2 โหมดไคลเอนต์ (Client mode)

การทำงานในโหมดไคลเอนต์ (Client Mode) คือ เครื่องคอมพิวเตอร์ หรืออุปกรณ์ที่ทำหน้าที่เป็นไคลเอนต์ จะเป็นผู้ทำการร้องขอ (Request) ไปยังเครื่องคอมพิวเตอร์ หรืออุปกรณ์ที่ทำหน้าที่เป็นเซิร์ฟเวอร์ โดยตัวไคลเอนต์ที่ร้องขอ จะต้องระบุหมายเลขไอพีแอดเดรส (IP Address) และหมายเลขพอร์ต (Port) ของเครื่องเซิร์ฟเวอร์ ที่ทำการเปิดพอร์ตรอ

2.10 Arduino IDE

Arduino เป็นแนวคิดในการสร้างสิ่งที่เรียกว่า แพลตฟอร์ม (Platform) ประกอบด้วย ฮาร์ดแวร์ (Hardware) และ ซอฟต์แวร์ (Software) ใช้สำหรับการเชื่อมต่อคอมพิวเตอร์ กับสิ่งแวดล้อมทางกายภาพและรวมถึงผู้ใช้ด้วย (หรือที่เรียกว่า Physical Computing) Arduino เริ่มต้นเผยแพร่ผลงานในราวปี ค.ศ. 2005 มีวัตถุประสงค์คือ การสร้างฮาร์ดแวร์และซอฟต์แวร์ที่เป็น Open Source เปิดเผยโค้ดต้นฉบับ ผังวงจรสำหรับบอร์ดไมโครคอนโทรลเลอร์ และเอกสารรายละเอียดอื่นๆ เช่น คู่มือการใช้งาน โค้ดตัวอย่าง ภายใต้เงื่อนไขการนำไปเผยแพร่และพัฒนาต่อ เหมาะสำหรับผู้ที่ต้องการสร้างอุปกรณ์อิเล็กทรอนิกส์เป็นต้นแบบ (Electronics Prototyping) และนำไปสู่การนำเสนอผลงาน และแลกเปลี่ยนอย่างเสรีในวงกว้าง

ซอฟต์แวร์ที่ใช้ในการเขียนโค้ดสำหรับ Arduino (เรียกว่า Arduino Sketch) คือ Arduino IDE มีสองเวอร์ชัน 1.0.5 และ 1.5.x (รองรับฮาร์ดแวร์ใหม่ เช่น Due, Yun) มีให้เลือกใช้สำหรับระบบปฏิบัติการ Windows, Linux, Mac OS X การเขียนโค้ด Arduino Sketch จะใช้ภาษาซี/ซีพลัสพลัส ดังนั้นสามารถเขียนโปรแกรมเชิงวัตถุได้ (Object-oriented Programming / OOP) สามารถสร้างและใช้งานคลาส (Class) หรือ ออบเจกต์ (Object) เกี่ยวข้องกับการใช้งานฮาร์ดแวร์ต่างๆ ของไมโครคอนโทรลเลอร์ (Microcontroller) มีชุดคำสั่งสำหรับใช้งานสำหรับ Arduino มีการสร้างฟังก์ชัน (Functions) หรือคำสั่งไว้ให้เรียกใช้งานหลายคำสั่ง และจัดทำให้อยู่ในรูปไลบรารี (C++ Library) เพื่อสะดวกต่อการนำไปใช้งาน

2.11 Web Server

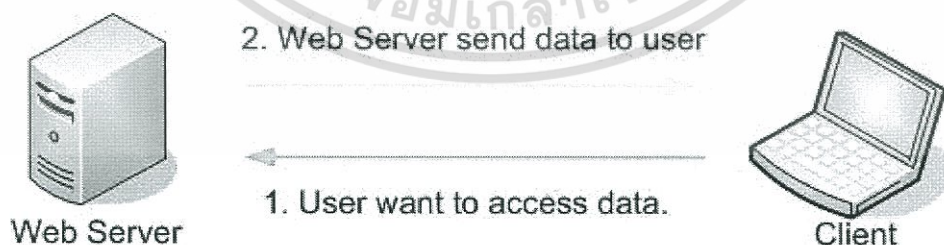
Web Server คือโปรแกรมที่มีอยู่และทำงานบนเครื่องฝั่ง Server ทำหน้าที่ในการรับคำสั่งจากการร้องขอของฝั่ง Client (โดยผ่านทาง Browser) และประมวลการทำงานจากการร้องขอดังกล่าว แล้วส่งข้อมูลกลับไปยังเครื่อง Client ที่ร้องขอ สรุปคือ Web Server คือโปรแกรมที่คอยให้บริการแก่ Client ที่ร้องขอข้อมูลเข้ามาโดยผ่าน Browser เว็บที่เขียนด้วย ASP นั้นจะทำงานได้ จะต้องมีการมี Web Server เป็นตัวประมวลผลอีกทีหนึ่ง ดังนั้นถ้าเราต้องการให้เครื่องของเราสามารถประมวลผล ASP ได้เราจะต้องจำลองเครื่องคอมพิวเตอร์ของเราให้เป็น Server โดยใช้โปรแกรม Web Server ดังที่กล่าวข้างต้น ถ้าเครื่องคอมพิวเตอร์ใช้ Windows 95, Windows 98 หรือ Win Me Web Server ที่ใช้คือ Personal Web Server (PWS) ถ้าเครื่องคอมพิวเตอร์ใช้ Windows NT, Windows 2000 หรือ XP Web Server ที่ใช้คือ Internet Information Server (IIS)

2.11.1 ประเภทการออกแบบรูปแบบของเว็บไซต์ในปัจจุบัน

2.11.1.1 Static Website

Static Website คือ เว็บไซต์ที่มีรูปแบบลักษณะเนื้อหาทั่วไปที่สร้างด้วย HTML โดยที่มีรูปแบบการทำงานคือ เมื่อผู้ใช้มีการเรียกใช้งานเว็บไซต์ ก็จะมีการส่งเนื้อหาในส่วนนั้นไปให้กับผู้ร้องขอ และจะมีการแสดงผลผ่านบราวเซอร์บนเครื่องผู้ใช้

ความเหมาะสมในการออกแบบเว็บไซต์ในรูปแบบของ Static Website นั้นเหมาะสำหรับองค์กรที่มีขนาดเล็ก มีการอัปเดตข้อมูลข่าวสารไม่มาก เพราะการทำงานในรูปแบบนี้จะไม่ได้มีการเชื่อมต่อข้อมูลกับฐานข้อมูล (Database) นั้นหมายความว่า ในกรณีที่ต้องการอัปเดตข้อมูลเว็บไซต์แต่ละครั้ง จะต้องทำการอัปเดตเป็นที่ละส่วน ซึ่งข้อเสียก็คือจะทำให้การทำงานเกิดความยุ่งยากและเป็นไปอย่างล่าช้า เนื่องจากไม่ได้มีฐานข้อมูลที่จัดเก็บข้อมูลเหล่านั้นไว้ ดังรูปที่ 2.5



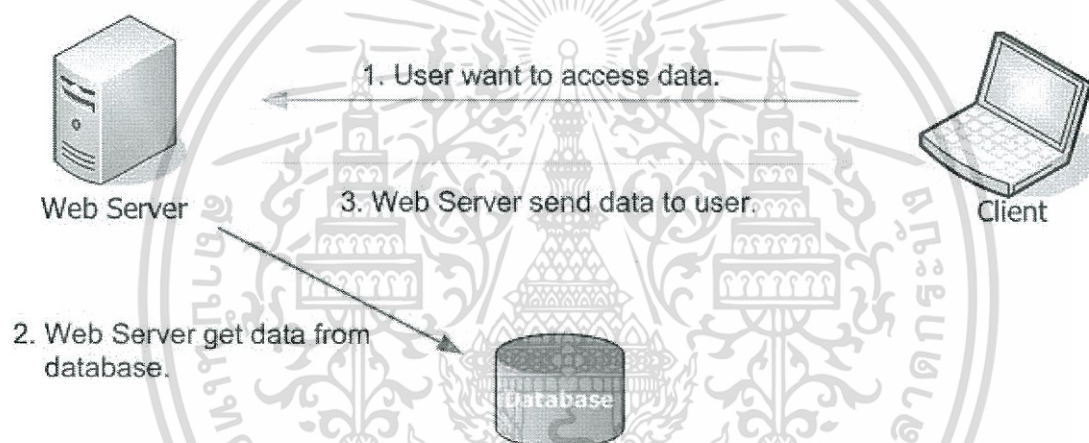
รูปที่ 2.5 Static Website [10]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11.1.2 Dynamic Website

Dynamic Website คือ เว็บไซต์ที่มีรูปแบบและลักษณะที่แตกต่างจาก Static Website เนื่องจากจะมีการจัดเก็บข้อมูลไว้ในรูปแบบของฐานข้อมูล กล่าวคือ เมื่อมีการอัปเดตข้อมูลใดๆ จะสามารถทำได้รวดเร็วและสะดวกกว่ารูปแบบ Static Website

การใช้งานรูปแบบนี้ นิยมใช้กับเว็บไซต์ที่มีการให้บริการเว็บบอร์ดหรือระบบการสืบค้นข้อมูล เป็นต้น ซึ่งทั้งนี้การทำเว็บไซต์ในรูปแบบ Dynamic Website นอกจากจะต้องมีความรู้พื้นฐานในการใช้งาน HTML แล้ว ยังจะต้องมีความรู้ในการเขียนโปรแกรมอย่างน้อย 1 ภาษา เช่น ASP, ASP.NET หรือ PHP เป็นต้น ประกอบการจัดการฐานข้อมูลเพื่อให้ข้อมูลทั้งหมดนั้นสามารถเชื่อมต่อกับฐานข้อมูลและใช้งานได้จริงเมื่อมีการอัปเดตข้อมูลใด ดังรูปที่ 2.6



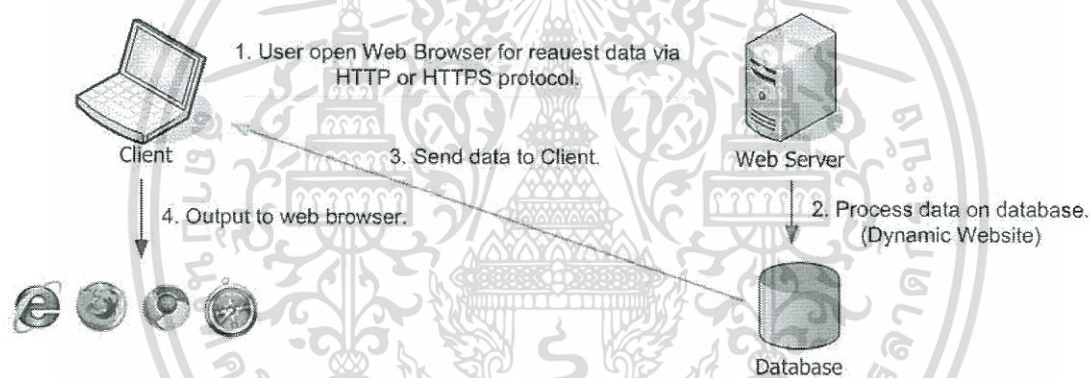
รูปที่ 2.6 Dynamic Website [10]

แต่เนื่องจากปัจจุบันได้มีระบบที่สามารถช่วยให้ผู้ดูแลเว็บไซต์สามารถทำงานได้ง่ายขึ้น หรือที่เรียกว่าระบบ CMS (Content Management System) คือระบบที่จะช่วยผู้ที่ทำเว็บไซต์ สามารถจัดการได้โดยไม่ต้องมีความรู้พื้นฐานการเขียนโปรแกรม กล่าวคือ ทำการศึกษารายละเอียดการใช้ระบบดังกล่าว ก็จะสามารถสร้างเว็บไซต์ได้ทั้งรูปแบบ Static Website และ Dynamic Website ได้ ซึ่งปัจจุบัน มี Web CMS ที่สามารถใช้งานให้ฟรีได้หลายประเภท ยกตัวอย่างเช่น Joomla, Mambo หรือ Wordpress เป็นต้น

2.11.2 หลักการทำงานของ Web Server

หลักการทำงานของ Web Server จะต้องมีซอฟต์แวร์หรือโปรแกรมที่นำมาทำเป็นบริการ (Service) Web Server ซึ่งที่นิยมมากในปัจจุบันจะมีซอฟต์แวร์ที่ชื่อว่า IIS (Internet Information Service) เป็นบริการที่สามารถทำงานได้บนระบบปฏิบัติการ Windows โดยปกติจะมีมาอยู่ในระบบให้โดยอัตโนมัติ และซอฟต์แวร์ที่ชื่อว่า Apache เป็นบริการที่นิยมใช้บนระบบปฏิบัติการ UNIX โดยปกติจะไม่มีมาให้ ซึ่งจะต้องทำการดาวน์โหลดมาติดตั้งถึงจะสามารถใช้งานเป็นเว็บเซิร์ฟเวอร์ได้

ในส่วนของผู้ใช้นั้น การเรียกใช้บริการเว็บไซต์ มีวิธีการคือ จะต้องใส่ที่อยู่ของเว็บไซต์ หรือที่เรียกว่า URL (Uniform Resource Locator) ในโปรแกรมเว็บเบราว์เซอร์ ซึ่งโดยปกติแล้วการเรียกใช้งาน จะมีการเชื่อมต่อโดยโพรโทคอล HTTP (Hypertext Transfer Protocol) หรือ HTTPS (Hypertext Transfer Protocol over Secure Socket Layer) ในการเข้าถึงข้อมูลนั้นๆ ดังภาพตัวอย่างรูปที่ 2.7



รูปที่ 2.7 การเรียกใช้งานบริการเว็บไซต์ของผู้ใช้งาน [10]

2.11.3 ความรู้เบื้องต้นของ HTTP

Hyper Text Transfer Protocol (HTTP) เป็นโพรโทคอลหลักในการใช้งาน World Wide Web โดยมีจุดประสงค์แรกเริ่มในการเป็นช่องทางสำหรับเผยแพร่และแลกเปลี่ยน HTML HTTP เป็นการพัฒนาร่วมกันโดย World Wide Web Consortium ซึ่งเป็นหน่วยงานดูแลมาตรฐานเว็บและคณะทำงานจาก Internet Engineering Task Force โดยออกมาเป็นชุดเอกสาร RFC เอกสารชิ้นที่สำคัญคือ RFC 2616 ซึ่งเป็นมาตรฐาน HTTP 1.1 ที่ใช้กันอย่างกว้างขวางในปัจจุบัน HTTP เป็นโพรโทคอลที่ใช้ร้องขอ/ตอบกลับ ระหว่างเครื่องลูกข่ายที่ใช้เว็บเบราว์เซอร์ กับเครื่องแม่ข่ายที่เรียกว่า เว็บเซิร์ฟเวอร์ โดยทำงานบนโพรโทคอล TCP ตามพอร์ตที่กำหนด (ปกติใช้พอร์ต 80)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เว็บเซิร์ฟเวอร์ (Web server) คือ บริการ HTTP (Hyper Text Transfer Protocol) เพื่อให้ผู้ใช้สามารถอ่านข้อมูลทั้งภาพและเสียงจากเครื่องบริการผ่านเบราว์เซอร์ เช่น บริการ <http://www.thaiall.com> หรือ <http://localhost> เป็นต้น เครื่องบริการที่รับคำสั่งขอจาก Web browser ข้อมูลที่จะส่งไปอาจเป็นเว็บเพจ รูปภาพ หรือเสียง เป็นต้น สำหรับโปรแกรมที่ได้รับความนิยมให้นำมาเปิดบริการ Web คือ Apache web server หรือ Microsoft web server

World Wide Web เรียกสั้นๆ ว่า “Web” หมายถึง บริการทางอินเทอร์เน็ตที่นำเสนอข้อมูลข่าวสาร และสารสนเทศในรูปแบบ multimedia จากเครื่องคอมพิวเตอร์หลายเครื่องหลายบุคคล ที่ต้องเผยแพร่ให้คนทั่วโลกได้รับรู้ผ่านอินเทอร์เน็ต โดยใช้โพรโทคอล HTTP ในการติดต่อสื่อสาร และคอมพิวเตอร์เหล่านี้จะถูกเรียกว่า เว็บเซิร์ฟเวอร์ (Web server) ซึ่งกระจายอยู่ตามประเทศต่างๆ ทั่วโลกและข้อมูลหรือข่าวสารต่างๆ ที่เราสามารถรับชมผ่านโปรแกรมเว็บเซิร์ฟเวอร์ได้นั้น โดยทั่วไปจะเรียกว่า “Web Site”

2.11.4 การติดต่อสื่อสารของ HTTP

หลักการทำงานของโพรโทคอล HTTP จะเป็นในรูปแบบของเครื่องผู้ใช้-เว็บเซิร์ฟเวอร์ ดังแสดงในรูปที่ 2.8 โดยจะมีโปรแกรมเว็บเบราว์เซอร์ในส่วนของเครื่องผู้ใช้ในกรณีที่มีการเรียกดูข้อมูล ส่วนทางด้านเว็บเซิร์ฟเวอร์จะประกอบไปด้วยโปรแกรมที่ชื่อว่า IIS (Internet Information Service) ที่ทำงานอยู่บนระบบปฏิบัติการ Windows หรือโปรแกรมที่ชื่อว่า Apache ที่ทำงานอยู่บนระบบปฏิบัติการ UNIX ซึ่งมีการทำงานอยู่รูปแบบของ HTTP Message โดยแบ่งออกเป็น 2 ประเภท ดังนี้

2.11.4.1 การร้องขอ (Request)

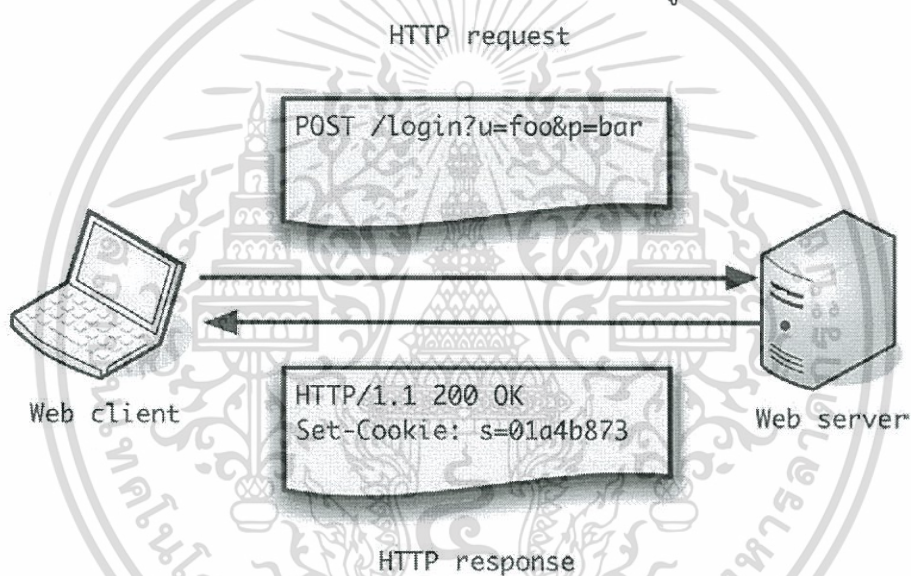
HTTP Request คือวิธีการส่งข้อมูลการร้องขอไปยังเซิร์ฟเวอร์ โดยแบ่งออกเป็น 2 วิธี ดังนี้

- 1.) การส่งข้อมูลโดยการ GET หมายถึง การส่งข้อมูลไปยังเซิร์ฟเวอร์ในรูปแบบของ URL เหมาะสำหรับกรณีที่ข้อมูลเหล่านั้นมีจำนวนไม่มาก ยกตัวอย่างเช่น เว็บไซต์ที่ให้บริการที่เกี่ยวข้องกับการค้นหาข้อมูล เป็นต้น
- 2.) การส่งข้อมูลโดยการ POST หมายถึง การส่งข้อมูลไปยังเซิร์ฟเวอร์ในรูปแบบของ Entity Body ซึ่งวิธีการนี้เหมาะสำหรับการส่งข้อมูลที่มีข้อมูลจำนวนมาก ยกตัวอย่างเช่น เว็บไซต์ที่มีบริการให้สมัครสมาชิกหรือกรอกแบบสอบถาม เป็นต้น

2.11.4.2 การตอบกลับ (Response)

HTTP Response คือการส่งข้อมูลการตอบกลับจากเซิร์ฟเวอร์มายังผู้ใช้ว่าสถานะของเซิร์ฟเวอร์เป็นอย่างไร ยกตัวอย่างเช่น

- 1.) 200 OK หมายถึง การร้องขอข้อมูลบนเซิร์ฟเวอร์ สามารถทำได้สำเร็จ และข้อมูลเหล่านั้นจะถูกส่งกลับไปยังผู้ใช้
- 2.) 400 Bad Request หมายถึง ไม่ทราบ ความหมาย ของ HTTP Request ที่ได้รับเข้ามาจากผู้ใช้
- 3.) 505 HTTP Version Not Supported หมายถึง เซิร์ฟเวอร์ไม่รองรับเวอร์ชันของ HTTP ที่ได้รับขอมาจากผู้ใช้



รูปที่ 2.8 หลักการทำงานของโพรโทคอล HTTP [10]

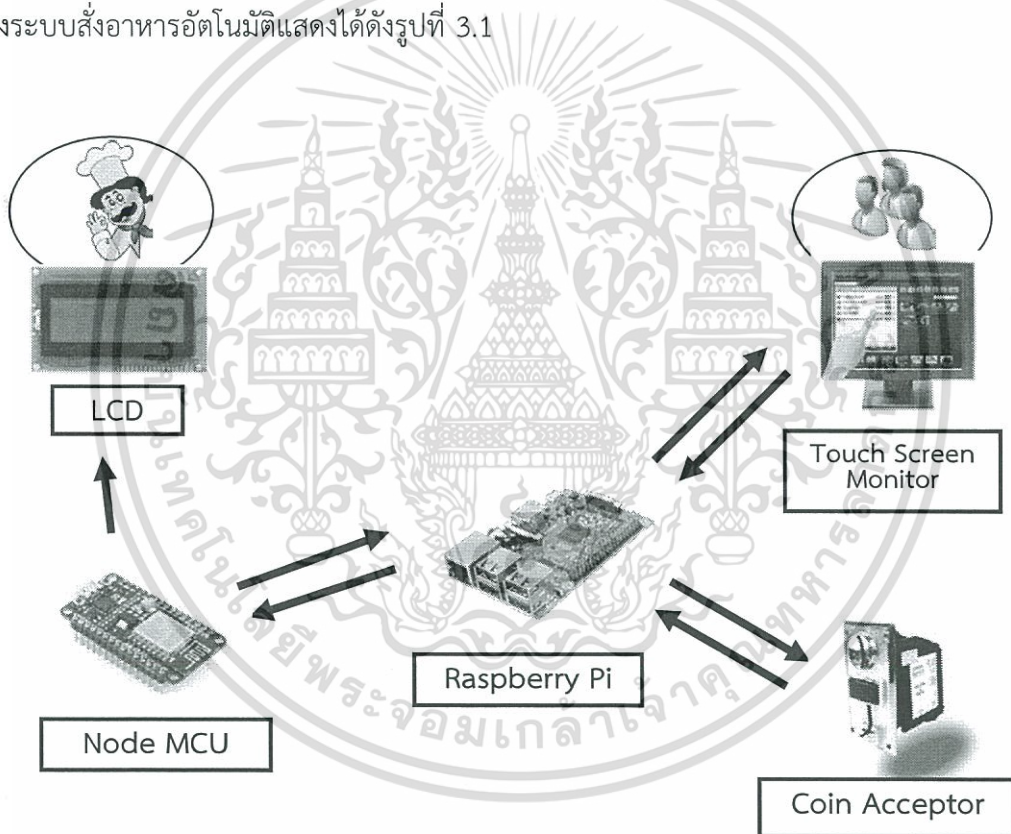
ในการปฏิบัติการของ HTTP จะมีลูกโซ่ของการร้องขอ-ตอบรับ (request-response) รูปแบบที่หนึ่ง เมื่อเครื่องผู้ใช้ทำการเชื่อมต่อโดยตรงสู่เครื่องให้บริการบนพอร์ตหมายเลข 80 (อาจเป็นเบอร์อื่นได้แล้วแต่จะกำหนด) และส่งคำร้องขอ (request) ออกไป เมื่อเครื่องให้บริการซึ่งกำลังรออยู่ได้รับการเชื่อมต่อก็จะสร้างกระบวนการ (process) ใหม่เพื่อให้บริการกับคำร้องขอนั้น เมื่อคำร้องขอได้รับการประมวลแล้ว เครื่องให้บริการจะส่งคำตอบที่ได้กลับไปทางการเชื่อมต่อที่พอร์ตเดิม

บทที่ 3

การออกแบบและการจัดทำปฏิญญานิพนธ์

3.1 การออกแบบ

ระบบสั่งอาหารอัตโนมัติโดยบอร์ด Raspberry Pi ถูกออกแบบมาให้ติดตั้งได้ภายในร้านอาหาร มีจุดประสงค์เพื่ออำนวยความสะดวกให้กับลูกค้าภายในร้าน รวมไปถึงลดขั้นตอนความยุ่งยาก และข้อผิดพลาดในการสั่งอาหาร โดยผู้ใช้งานสามารถสั่งอาหารผ่านหน้าจอแสดงผลที่ควบคุมโดยบอร์ด Raspberry Pi ระบบสั่งอาหารอัตโนมัติจะแบ่งการทำงานออกเป็น 2 ส่วนคือส่วนติดต่อและแสดงผลกับผู้ใช้งาน และส่วนแสดงผลสำหรับห้องครัว บล็อกไดอะแกรมของระบบสั่งอาหารอัตโนมัติแสดงได้ดังรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมของระบบสั่งอาหารอัตโนมัติ

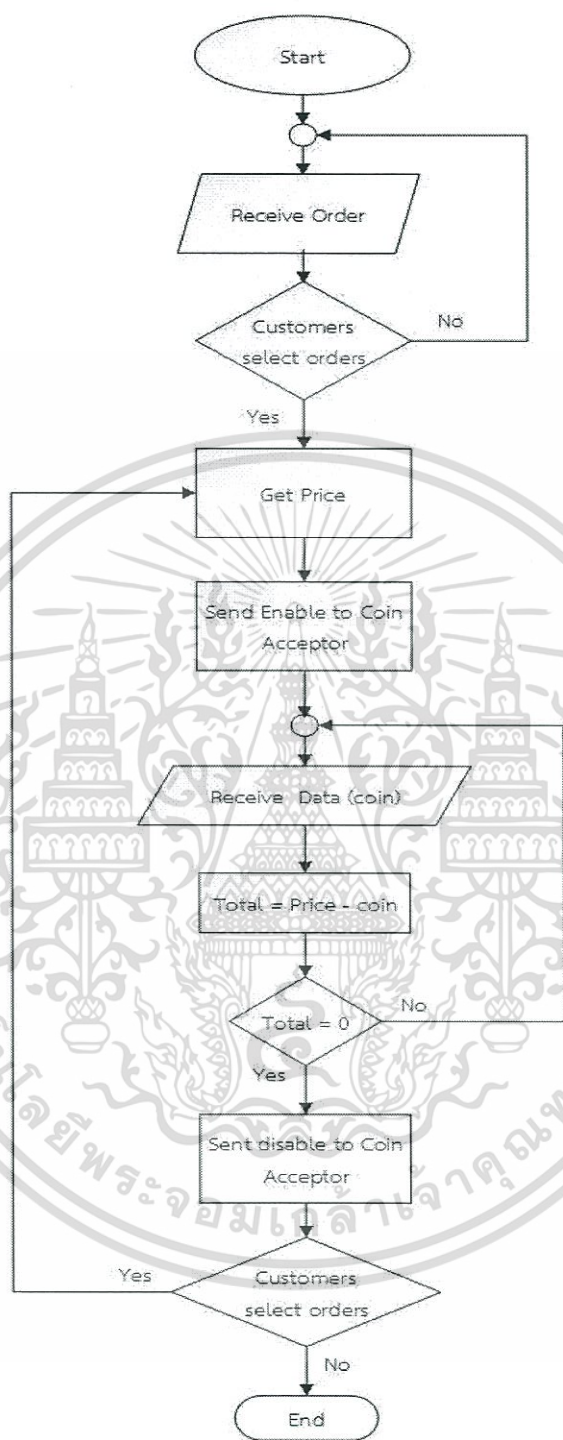
3.1.1 การออกแบบส่วนติดต่อและแสดงผลกับผู้ใช้งาน

ส่วนติดต่อและแสดงผลกับผู้ใช้งาน มีตัวควบคุมหลักคือ บอร์ด Raspberry Pi เชื่อมต่อเข้ากับจอแสดงผลและเครื่องรับเหรียญชนิดหลายเหรียญ โดยทำการออกแบบจอติดต่อและแสดงผล (GUI : Graphical User Interface) ผ่านโปรแกรมควิที ครีเอเตอร์ (QT Creator) ที่ติดตั้งบนระบบปฏิบัติการลินุกซ์ (Linux Operating System) ของบอร์ด Raspberry Pi และติดต่อสื่อสารร่วมกับเครื่องรับเหรียญชนิดหลายเหรียญ

3.1.1.2 การออกแบบซอฟต์แวร์ในบอร์ด Raspberry Pi

ขั้นตอนการทำงานของโปรแกรมควิที ครีเอเตอร์ (QT Creator) ในบอร์ด Raspberry Pi เป็นดังรูปที่ 3.2





รูปที่ 3.2 แผนผังขั้นตอนการทำงานของบอร์ด Raspberry Pi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

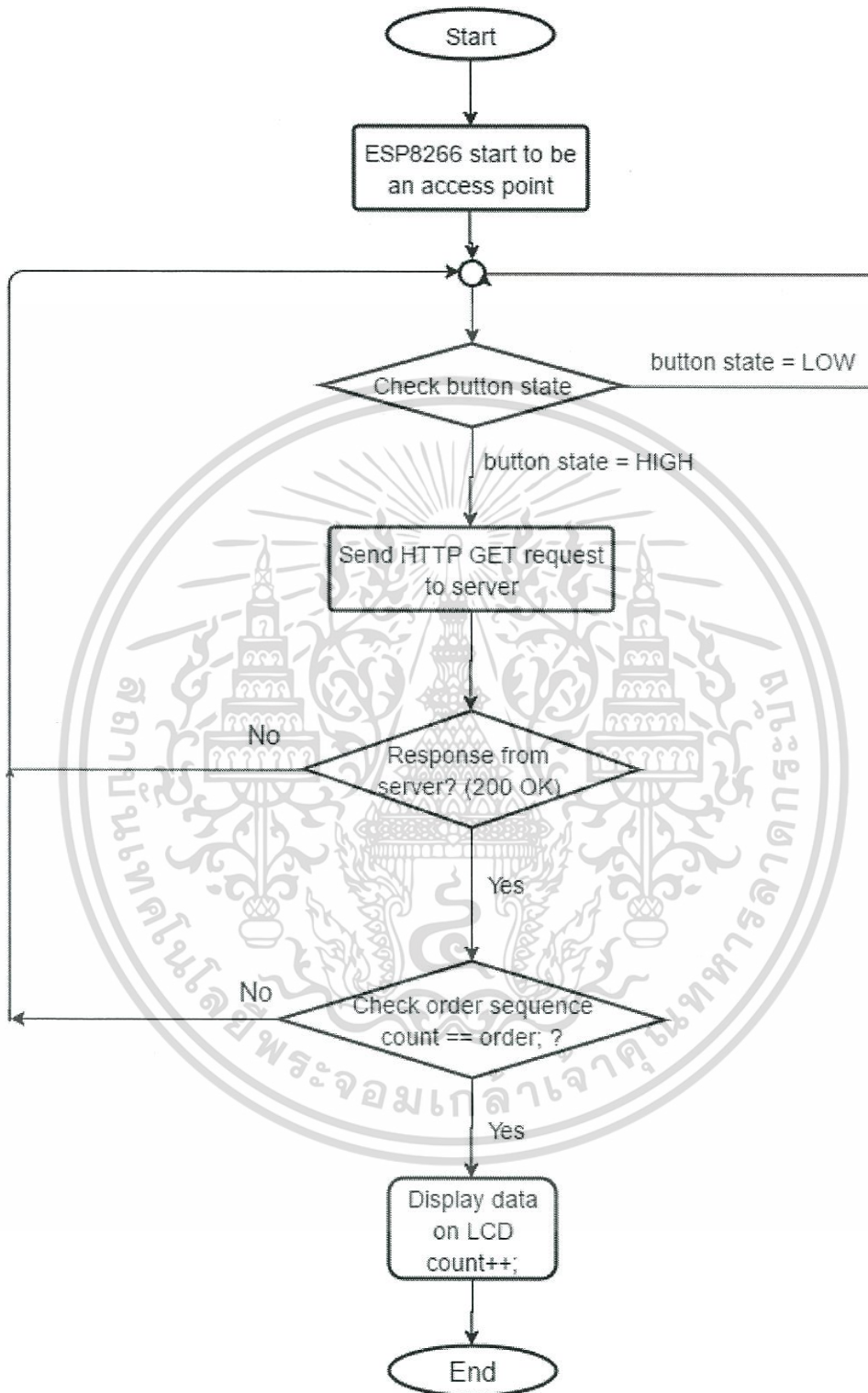
จากรูปที่ 3.2 อธิบายได้ว่า บอร์ด Raspberry Pi เป็นตัวประมวลผลและสั่งการ เมื่อ Raspberry Pi ได้รับข้อมูลการสั่งอาหารจากผู้ใช้งาน จะทำการส่งคำสั่งเริ่มการทำงานไปยังเครื่องรับเหรียญ และรอรับข้อมูลของเหรียญจากเครื่องรับเหรียญ เมื่อมีการส่งข้อมูลเหรียญเข้ามา ข้อมูลจะถูกแปลงเป็นมูลค่าราคาของเหรียญและทำการหักลบจากราคาอาหาร โดยจะทำการรอรับข้อมูลซ้ำจนกว่าข้อมูลที่ได้รับหักลบกับราคาอาหาร เท่ากับ ศูนย์ เมื่อได้รับข้อมูลครบแล้ว Raspberry Pi จะทำการส่งคำสั่งหยุดการทำงานไปยังเครื่องรับเหรียญ และรอรับการสั่งอาหารครั้งต่อไป

3.1.1.2 การติดต่อสื่อสารกับเครื่องรับเหรียญชนิดหลายเหรียญ

เครื่องรับเหรียญที่ใช้จะมีสายสัญญาณจำนวน 4 เส้น ที่ใช้งาน ได้แก่ สายไฟเลี้ยง 12 โวลต์ สายกราวด์ และสายสัญญาณรับ-ส่ง (2 เส้น) โดยในการทดลองจะนำสายสัญญาณส่ง (Tx) ของเครื่องรับเหรียญต่อเข้ากับสายสัญญาณรับ (Rx) จาก Raspberry Pi และต่อสายสัญญาณรับ (Rx) เข้ากับสายสัญญาณส่ง (Tx) จาก Raspberry Pi สายกราวด์ต่อเข้าด้วยกัน จากนั้นต่อไฟเลี้ยง 12 โวลต์ ให้กับเครื่องรับเหรียญ เขียนคำสั่งตรวจสอบสัญญาณที่เข้ามา เพื่อแสดงจำนวนเหรียญที่ผู้ใช้บริการหยอดเหรียญ ผ่านหน้าจอแสดงผลต่อไป

3.1.2 การออกแบบส่วนแสดงผลสำหรับห้องครัว

ส่วนแสดงผลสำหรับห้องครัว มีตัวควบคุมหลักคือ ไมโครคอนโทรลเลอร์ชนิด NodeMCU โมดูล ESP8266 เพื่อที่จะทำการแสดงผลข้อมูลรายการอาหารที่ถูกคำสั่งผ่านจอแอลซีดี NodeMCU จำเป็นจะต้องดึงข้อมูลจากฐานข้อมูลที่บอร์ด Raspberry Pi ส่งไปเก็บไว้ โดย NodeMCU ต้องทำการร้องขอไปยังฐานข้อมูลด้วยวิธี HTTP GET จากนั้นส่งค่าให้แสดงผลบนจอแอลซีดีผ่านพอร์ต DigitalIO (Input/Output) โดยมีขั้นตอนการทำงานของ NodeMCU โมดูล ESP8266 ดังรูปที่ 3.3



รูปที่ 3.3 แผนผังการทำงานของ NODEMCU โมดูล ESP8266

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.3 อธิบายได้ว่า NodeMCU โมดูล ESP8266 เมื่อเริ่มทำงานแล้ว จะถูกตั้งค่าให้เป็น Access Point ทันทีและตลอดการทำงาน โดยเมื่อพ่อครัวต้องการทราบรายการอาหารที่ลูกค้าสั่งไว้ ทำการกดปุ่ม เมื่อสถานะปุ่มกดเป็น HIGH NodeMCU จะส่ง HTTP GET request ไปยังเซิร์ฟเวอร์ที่เป็นเซิร์ฟเวอร์ฐานข้อมูลสำหรับเก็บข้อมูลการสั่งอาหาร หากได้รับการตอบกลับจากเซิร์ฟเวอร์ว่า “200 OK” หมายถึง การร้องขอข้อมูลบนเซิร์ฟเวอร์ สามารถทำได้สำเร็จ และข้อมูลเหล่านั้น จะถูกส่งกลับมา จากนั้น NodeMCU จะดึงข้อมูลการสั่งอาหาร แสดงขึ้นบนจอแอลซีดี ทางพ่อครัวจะสามารถทราบได้ว่ามีการสั่งเมนูอาหารอะไรมาบ้าง และเป็นลำดับที่เท่าไร

3.1.3 การออกแบบฐานข้อมูลสำหรับเก็บข้อมูลการใช้งานระบบสั่งอาหารอัตโนมัติ

ในการใช้งานระบบสั่งอาหารอัตโนมัตินั้น จำเป็นจะต้องมีการเก็บบันทึกข้อมูลสำหรับการใช้งานในแต่ละครั้ง โดยข้อมูลที่ทำกรบันทึกนั้นจะประกอบด้วยข้อมูลจำนวนมาก อาทิ รายการอาหารที่สั่งในแต่ละครั้ง จำนวนราคาอาหารรวมต่อการใช้งานหนึ่งครั้ง หรือจำนวนเหรียญที่ผู้ใช้งานใช้เพื่อชำระตามจำนวนราคาอาหาร ดังนั้นจึงต้องสร้างฐานข้อมูลที่ใช้สำหรับเก็บข้อมูลเหล่านี้ไว้ สำหรับผู้ประกอบการในการตรวจสอบความถูกต้องในการใช้งานระบบสั่งอาหารอัตโนมัติในแต่ละวัน โดยในระบบสั่งอาหารอัตโนมัตินี้จะใช้ฐานข้อมูลชื่อว่า phpMyAdmin โดยจะใช้การจัดการฐานข้อมูล MySQL แทนการคีย์คำสั่ง และสามารถใช้งานผ่านเว็บเบราว์เซอร์ได้

3.1.3.1 การออกแบบการบันทึกข้อมูลการสั่งอาหาร

ในการเก็บข้อมูลเข้าสู่ระบบฐานข้อมูลอัตโนมัติ ด้วยการเขียนโปรแกรมภาษาพีเอชพี (PHP) ผู้จัดทำได้ทำการบันทึกข้อมูลการใช้งานระบบสั่งอาหารอัตโนมัติในแต่ละครั้งไว้เป็นไฟล์ข้อมูลประเภทซีเอชวี (CSV) จากนั้นทำการเรียกใช้งานโปรแกรมภาษาพีเอชพี (PHP) ผ่านเว็บเบราว์เซอร์ โปรแกรมจะทำการอ่านข้อมูลจากไฟล์ซีเอชวีและเพิ่มข้อมูลที่อ่านได้ลงในระบบฐานข้อมูล

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 ไมโครคอนโทรลเลอร์ชนิด NodeMCU โมดูล ESP8266

3.2.2 Raspberry Pi เวอร์ชัน 3

3.2.3 เครื่องรับเหรียญชนิดหลายเหรียญ

3.2.4 จอแอลซีดี (LCD) ขนาด 20x4

3.2.5 สาย USB TTL Converter

3.2.6 บอร์ดและโปรแกรมอาร์ดูโน้

3.3 การจัดเก็บผลการทดลอง

3.3.1 การทดสอบการทำงานของพอร์ต GPIO ของ Raspberry Pi

การทดสอบการทำงานของพอร์ต GPIO ของ Raspberry Pi จะทำการเชื่อมต่อหลอดไฟ LED เข้ากับพอร์ต GPIO บนบอร์ด Raspberry Pi และเขียนโปรแกรมให้พอร์ต GPIO ควบคุมการ เปิด-ปิด ของหลอดไฟ LED โดยจะมีปุ่มกดสำหรับควบคุมการ เปิด-ปิด ของหลอดไฟ LED

3.3.2 การทดสอบให้ NodeMCU โมดูล ESP8266 เป็น Access point

ในส่วนนี้จะทดสอบการทำงานของ NodeMCU โมดูล ESP8266 ให้ทำงานในโหมด Access point โดยจะทำการเขียนโปรแกรมลงบนบอร์ด NodeMCU และใช้โทรศัพท์มือถือในการเชื่อมต่อผ่านเครือข่ายไร้สายเข้ากับ NodeMCU เมื่อเชื่อมต่อสำเร็จ โทรศัพท์มือถือจะได้รับแอดเดรส (IP Address) ในเครือข่ายเดียวกับ NodeMCU

3.3.3 การส่งข้อมูลจาก Raspberry Pi ให้ NodeMCU โมดูล ESP8266

การทดสอบการส่งข้อมูลจาก Raspberry Pi ไปยัง NodeMCU โมดูล ESP8266 จะทำการเชื่อมต่อ Raspberry Pi ที่ใช้คำสั่งส่งข้อมูลผ่านเครือข่ายไร้สาย เข้ากับ NodeMCU ที่ทำงานในโหมด Access point รับข้อมูลที่ส่งจาก Raspberry Pi และแสดงผลออกที่จอ LCD

3.3.4 การรับ-ส่งข้อมูลระหว่างบอร์ด Raspberry Pi กับ เครื่องรับเหรียญ

ในส่วนนี้จะเชื่อมต่อเครื่องรับเหรียญเข้ากับ Raspberry Pi ทำการทดสอบการรับ-ส่ง ข้อมูลระหว่าง Raspberry Pi และเครื่องรับเหรียญ โดยการส่งข้อมูลจะทำการส่งผ่านพอร์ตอนุกรม Raspberry Pi เมื่อเครื่องรับเหรียญได้รับเหรียญจะอ่านค่าของเหรียญ

และส่งข้อมูลมายังบอร์ด Raspberry Pi จากนั้นบอร์ด Raspberry Pi จะแสดงค่าบนโปรแกรมที่ใช้ทดสอบการรับ-ส่งข้อมูล ผ่านพอร์ตอนุกรมในเลขฐานหก

3.3.5 การเก็บบันทึกข้อมูลการสั่งอาหารผ่านส่วนติดต่อกับผู้ใช้งาน

ในการสั่งอาหารผ่านส่วนติดต่อกับผู้ใช้งานนั้น จำเป็นจะต้องมีการบันทึกการสั่งอาหารไว้สำหรับเก็บบันทึกลงในฐานข้อมูล จะทำการทดสอบการเก็บบันทึกโดยเมื่อสั่งอาหารผ่านส่วนติดต่อกับผู้ใช้งาน และผู้ใช้งานหยุดเหรียญครบตามจำนวนราคาอาหารแล้ว จากนั้นจะทำการสร้างไฟล์ข้อมูลสำหรับบันทึกการสั่งอาหารในแต่ละครั้งไว้

3.3.6 การทดสอบการเพิ่มข้อมูลลงในระบบฐานข้อมูล

การทดสอบการเก็บบันทึกข้อมูลลงในระบบฐานข้อมูล จะทดสอบด้วยวิธีการเพิ่มข้อมูลโดยการใช้ฟังก์ชันในระบบฐานข้อมูลโดยตรง และเพิ่มข้อมูลโดยการเขียนโปรแกรมด้วยภาษาพีเอชพี (PHP) ทำการเรียกใช้งานผ่านเว็บเบราว์เซอร์ เพื่อเพิ่มข้อมูลลงในระบบฐานข้อมูลอัตโนมัติ

3.3.7 การทดสอบการสร้างส่วนแสดงผลสำหรับสรุปข้อมูลการสั่งอาหาร

ในส่วนนี้ทางผู้จัดทำได้ออกแบบส่วนแสดงผล ผลสรุปข้อมูลการสั่งอาหารสำหรับผู้ประกอบการหรือเจ้าของระบบสั่งอาหารอัตโนมัติ โดยผู้ที่ต้องการเข้าถึงข้อมูลในส่วนนี้จะต้องยืนยันตัวตนผ่านระบบ เมื่อทำการยืนยันตัวตนเรียบร้อยแล้วโปรแกรมจะทำการแสดงผลสรุปข้อมูลผ่านการเรียกใช้งานบนเว็บเบราว์เซอร์

3.3.8 การทดสอบการร้องขอ (Request) ข้อมูลจากฐานข้อมูลบนเซิร์ฟเวอร์

การทดสอบการร้องขอข้อมูลการสั่งอาหารของลูกค้า จากฐานข้อมูลบนเซิร์ฟเวอร์ นำมาแสดงผลบนจอแอลซีดี เพื่อให้ทางห้องครัวรับรู้ข้อมูลในการทำอาหารต่อไป ในส่วนนี้ทางผู้จัดทำได้ออกแบบ โดยพ่อครัวจะต้องกดปุ่มเพื่อร้องขอข้อมูลการสั่งอาหารของลูกค้า จากนั้นข้อมูลจึงจะแสดงผลบนจอแอลซีดี

3.3.9 การทดสอบการแจ้งเตือนผ่านหลอดไฟ LED เมื่อมีการสั่งอาหารเข้ามา

การทดสอบการแจ้งเตือนผ่านหลอดไฟ LED นี้ จะแจ้งเตือนให้ทางฝั่งห้องครัวทราบ โดยเขียนโปรแกรมให้มีการตรวจสอบข้อมูลกับทางเซิร์ฟเวอร์ตลอดเวลา หากมีการสั่งอาหารเข้ามา หลอดไฟ LED ทางฝั่งห้องครัวจะติด แต่หากทางห้องครัวได้รับข้อมูลการสั่งอาหารล่าสุดแล้ว และไม่มีการสั่งอาหารเพิ่มเข้ามาแล้ว หลอดไฟ LED จะดับลง

บทที่ 4

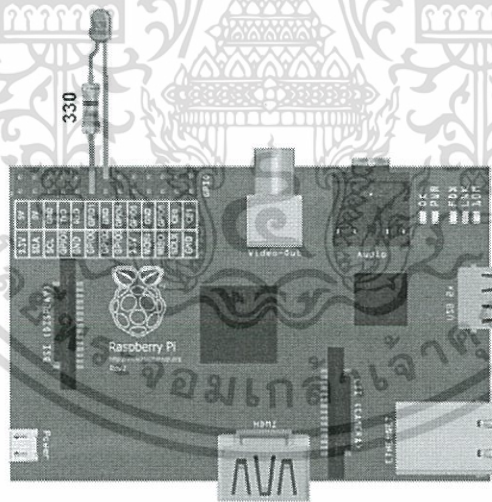
ผลการทดลอง

การทดลองการทำงานของระบบสั่งอาหารอัตโนมัติได้ทำการเก็บผลการทำงานของระบบ โดยแบ่งการทดลองและจัดเก็บผลการทดลองออกเป็นส่วนต่างๆ ดังต่อไปนี้

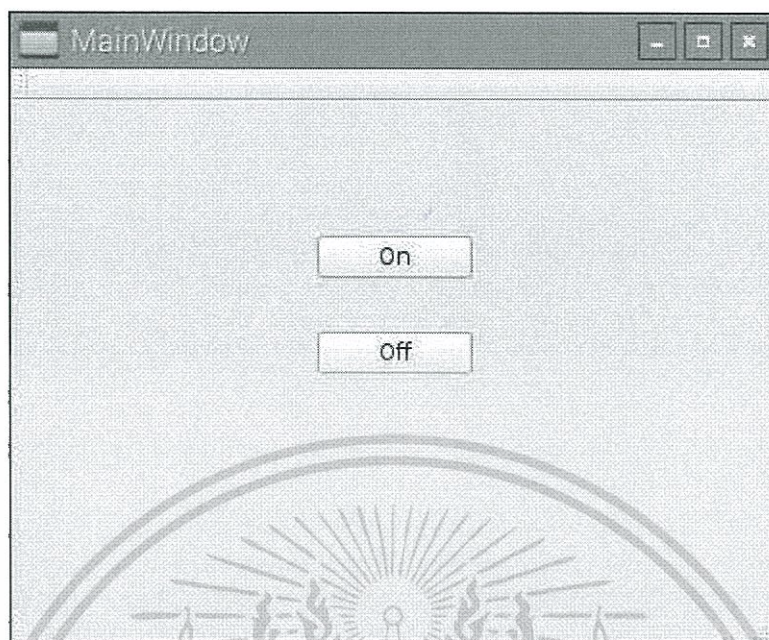
สำหรับการทดลองการทำงานของระบบสั่งอาหารอัตโนมัติ โดยใช้ Raspberry Pi เป็นตัวประมวลผลและสั่งการทำงานเข้ากับเครื่องรับเหรียญ เพื่อให้อุปกรณ์ทั้งหมดทำงานร่วมกันได้ตรงตามวัตถุประสงค์ที่ตั้งไว้ โดยผลการทดสอบเป็นดังนี้

4.1 การทดสอบการเขียนโปรแกรมควบคุมการทำงานของพอร์ต GPIO ของ Raspberry Pi

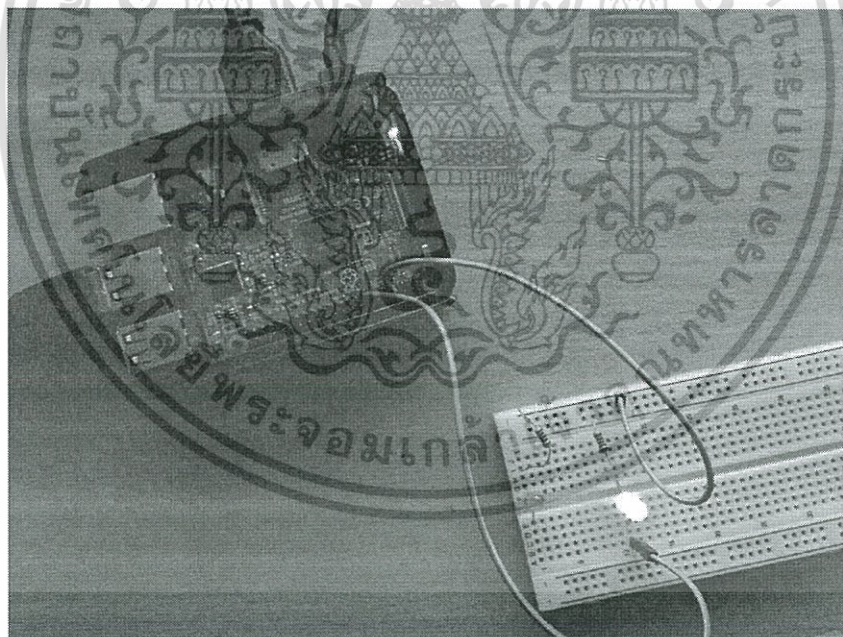
จากรูปที่ 4.1 ผู้จัดทำได้ทำการเชื่อมต่อหลอดไฟ LED เข้ากับ Raspberry Pi ผ่านพอร์ต GPIO เพื่อทดสอบการทำงานของพอร์ต GPIO ด้วยการทดลองเขียนโปรแกรมควบคุมเปิด/ปิด ไฟ LED ผ่าน GUI โดยนำเอาไลบรารี WiringPi เข้ามาใช้งานในการเขียนโปรแกรมควบคุม โดยหน้าจอ Interface จะแสดงดังรูปที่ 4.2 และแสดงผลการทำงานของโปรแกรกดังรูปที่ 4.3 และรูปที่ 4.4



รูปที่ 4.1 การเชื่อมต่อ LED เข้ากับ Raspberry Pi [8]

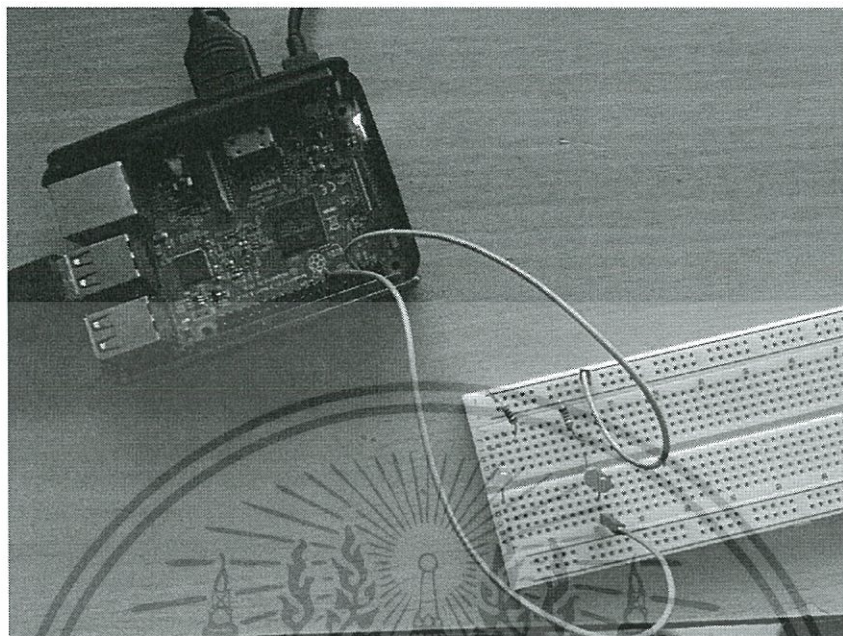


รูปที่ 4.2 หน้าจอ Interface โปรแกรม



รูปที่ 4.3 ผลการทดสอบเมื่อทำการกดปุ่ม On

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 ผลการทดสอบเมื่อทำการกดปุ่ม Off

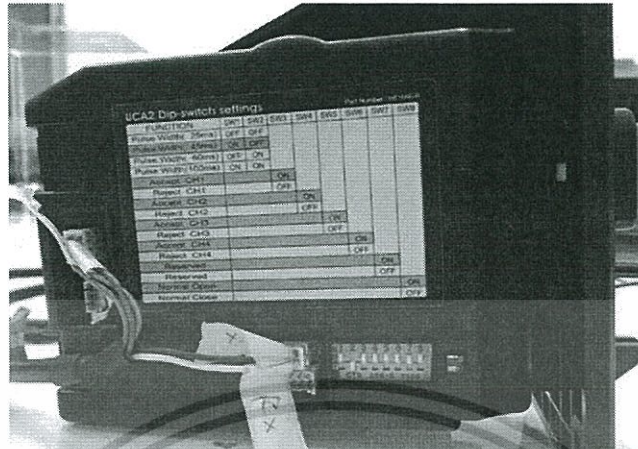
4.2 การทดสอบการทำงานของเครื่องรับเหรียญ

ผู้จัดทำทดสอบการทำงานของเครื่องรับเหรียญ โดยการเชื่อมต่อเครื่องรับเหรียญเข้ากับแหล่งจ่ายไฟ 12V เพื่อทำการทดสอบไฟสถานะของเครื่องรับเหรียญ ซึ่งไฟสีเขียวแสดงถึงสถานะพร้อมใช้งาน สามารถรับเหรียญและส่งข้อมูลของเหรียญได้ ส่วนไฟสีแดงแสดงถึงสถานะไม่พร้อมใช้งาน เครื่องรับเหรียญจะไม่สามารถรับเหรียญและส่งข้อมูลได้



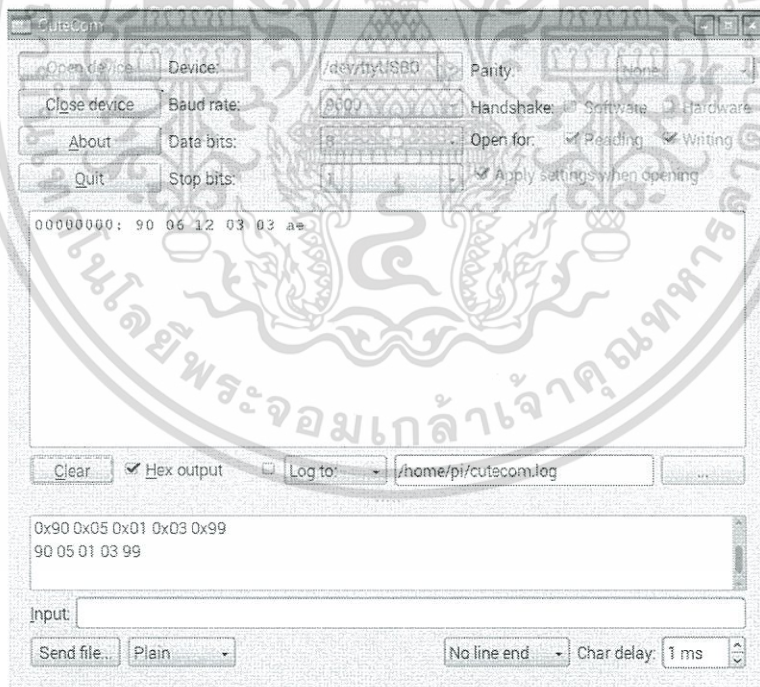
รูปที่ 4.5 เครื่องรับเหรียญแสดงไฟสถานะเป็นสีเขียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 เครื่องรับเหรียญแสดงไฟสถานะเป็นสีแดง

ผู้จัดทำ ทำการเชื่อมต่อ Raspberry Pi เข้ากับเครื่องรับเหรียญ แล้วทดสอบการทำงานของเครื่องรับเหรียญ โดยการใช้โปรแกรม Cutecom ในการรับ-ส่งข้อมูลผ่านพอร์ตอนุกรมของ Raspberry Pi โดยเมื่อทำการส่งคำสั่ง Enable ไปยังเครื่องรับเหรียญ จะมีการตอบ ACK กลับมาในรูปแบบเลขฐาน 6 ดังรูปที่ 4.7

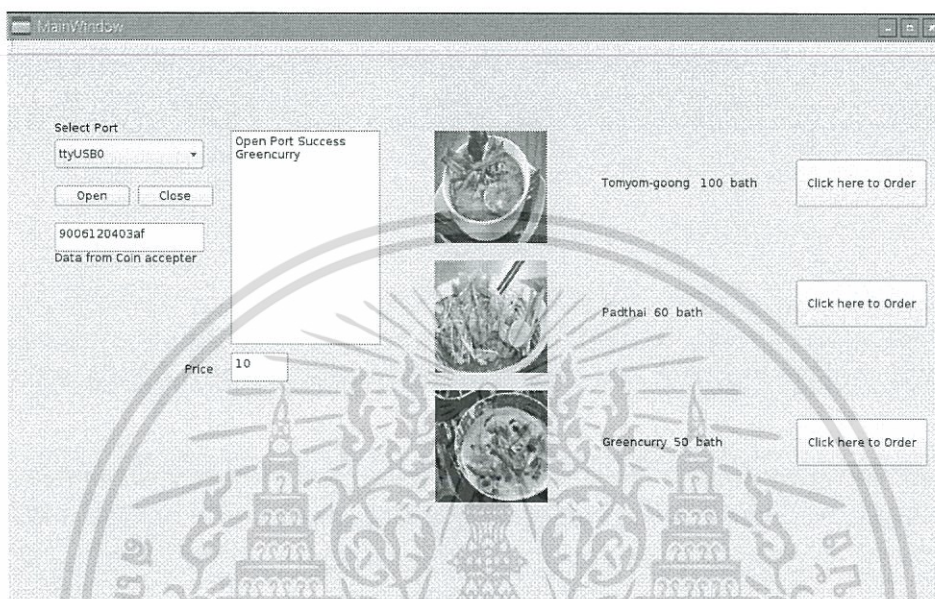


รูปที่ 4.7 เมื่อทำการส่งคำสั่ง Enable ไปยังเครื่องรับเหรียญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดสอบการเขียนโปรแกรมควบคุมการทำงานของเครื่องรับเหรียญผ่าน GUI

ผู้จัดทำ ทำการออกแบบ GUI เพื่อใช้ควบคุมและรับ-ส่ง ข้อมูลกับเครื่องรับเหรียญ ผ่านโปรแกรม QT Creator ซึ่งใช้ในการออกแบบ GUI ดังแสดงในรูปที่ 4.8



รูปที่ 4.8 หน้าจอ Interface เมื่อทำการรับข้อมูลจากเครื่องรับเหรียญ

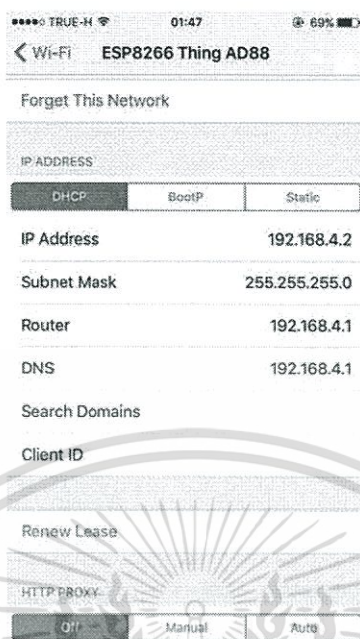
จากรูปที่ 4.8 เมื่อทำการกดสั่งเมนูอาหาร ราคาของอาหารจะแสดงในช่องราคา และ Raspberry Pi จะส่งคำสั่ง Enable ไปยังเครื่องรับเหรียญ เมื่อทำการหยอดเหรียญลงไป ตัวเครื่องรับเหรียญจะส่งข้อมูลกลับมา และแสดงข้อมูลที่ได้ในช่องข้อมูลจากเครื่องรับเหรียญ และในช่องราคา จะถูกหักลบตามจำนวนค่าของเหรียญที่ได้รับมาจากเครื่องรับเหรียญ

4.4 การทดสอบการทำงานของไมโครคอนโทรลเลอร์

ผู้จัดทำ ทำการทดสอบการทำงานของไมโครคอนโทรลเลอร์ เพื่อใช้สำหรับแสดงเมนูอาหารในห้องครัวสำหรับพ่อครัว ซึ่งมีผลการทดสอบดังนี้

4.4.1 ทดสอบการทำงานของไมโครคอนโทรลเลอร์ในโหมด Access Point

การทดสอบการทำงานโดยการเขียนโปรแกรมลงบนไมโครคอนโทรลเลอร์ ให้ทำงานในโหมด Access Point สำหรับการส่งข้อมูลผ่านเครือข่ายไร้สาย ดังรูปที่ 4.9

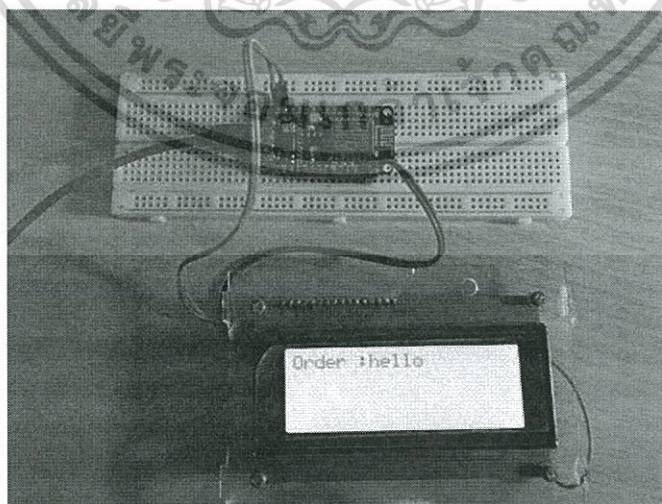


รูปที่ 4.9 เมื่อทำการเชื่อมต่อกับไมโครคอนโทรลเลอร์

จากรูปที่ 4.9 ทำการทดสอบการทำงานของไมโครคอนโทรลเลอร์โดยการใช้โทรศัพท์มือถือ เชื่อมต่อผ่านสัญญาณไวไฟ จะได้รับ IP Address ที่อยู่ภายใต้เครือข่ายเดียวกับไมโครคอนโทรลเลอร์

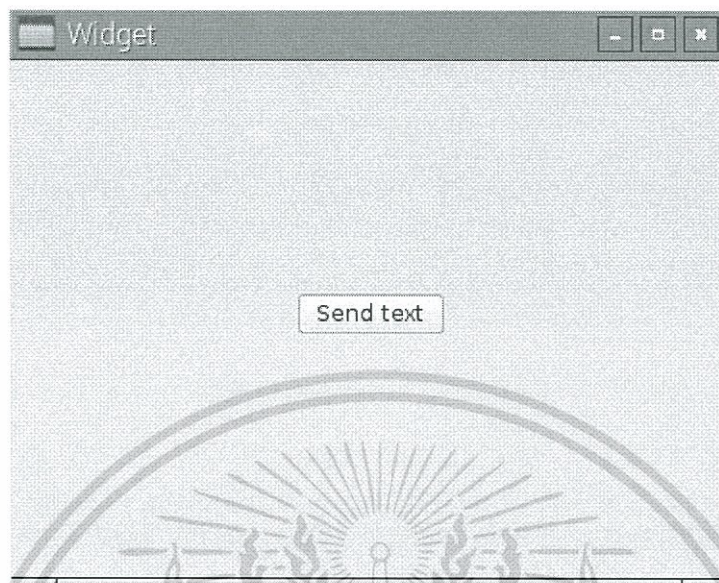
4.4.2 ทดสอบการทำงานร่วมกันของไมโครคอนโทรลเลอร์และ Raspberry Pi

ผู้จัดทำได้ทำการเขียนโปรแกรมลงบน Raspberry Pi ให้ส่งข้อมูลผ่านเครือข่ายไร้สาย เมื่อไมโครคอนโทรลเลอร์ได้รับข้อมูลจะแสดงผลข้อมูลที่รับผ่านทางจอแอลซีดี ดังรูปที่ 4.10



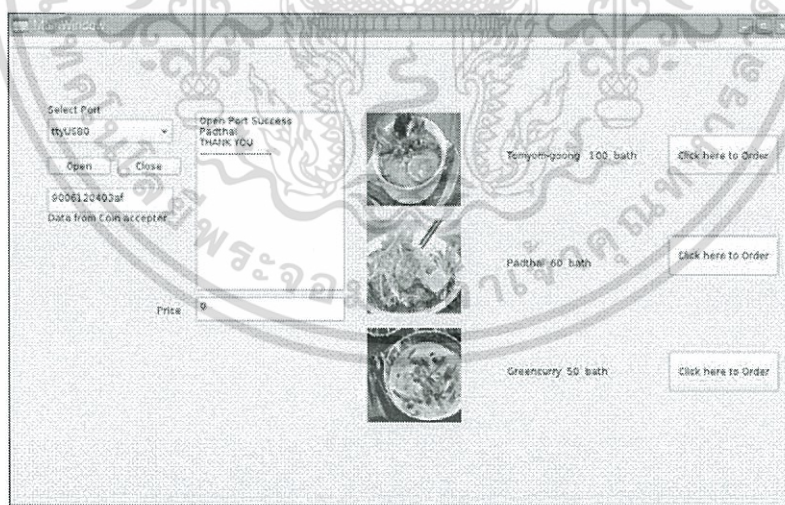
รูปที่ 4.10 ไมโครคอนโทรลเลอร์ได้รับข้อมูลจาก Raspberry Pi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



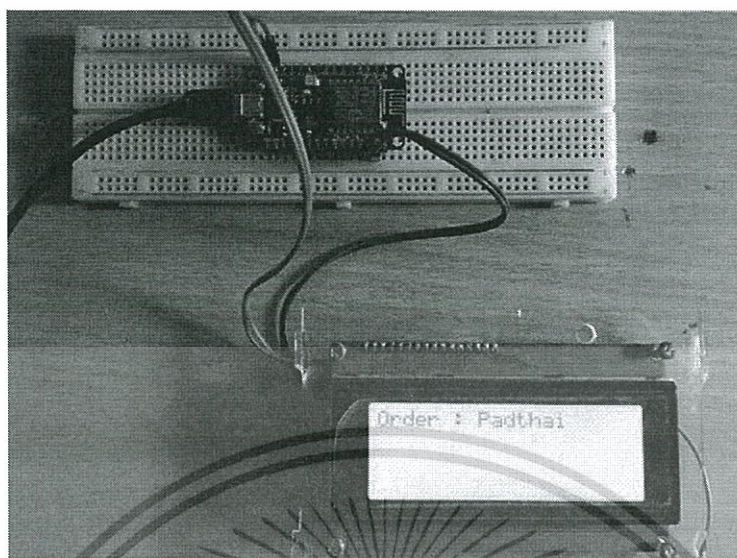
รูปที่ 4.11 หน้าจอ Interface สำหรับส่งข้อมูล

4.5 การทดสอบการทำงานร่วมกันของไมโครคอนโทรลเลอร์ เครื่องรับเหรียญ และ Raspberry Pi



รูปที่ 4.12 Raspberry Pi ได้รับข้อมูลจากเครื่องรับเหรียญจนครบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



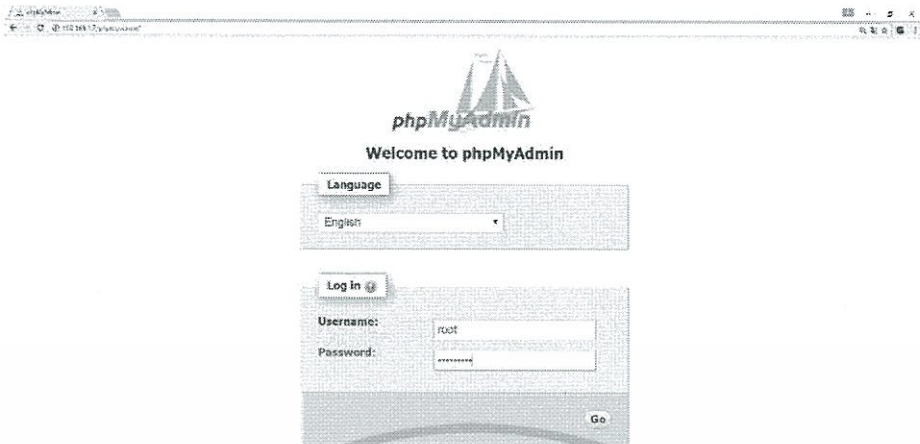
รูปที่ 4.13 ไมโครคอนโทรลเลอร์ได้รับข้อมูลจาก Raspberry pi และแสดงผลผ่านจอแอลซีดี

จากรูปที่ 4.12 และ 4.13 เป็นการทำงานร่วมกันของอุปกรณ์ทั้งหมด โดยเมื่อทำการเลือกเมนู และหยอดเหรียญจนครบตามราคาอาหาร Raspberry Pi จะส่งข้อมูลอาหารไปยังไมโครคอนโทรลเลอร์ และไมโครคอนโทรลเลอร์จะแสดงผลเมนูอาหาร ผ่านทางหน้าจอแอลซีดี

4.6 ทดสอบการสร้างฐานข้อมูลสำหรับเก็บบันทึกข้อมูลการใช้งานระบบสั่งอาหารอัตโนมัติ

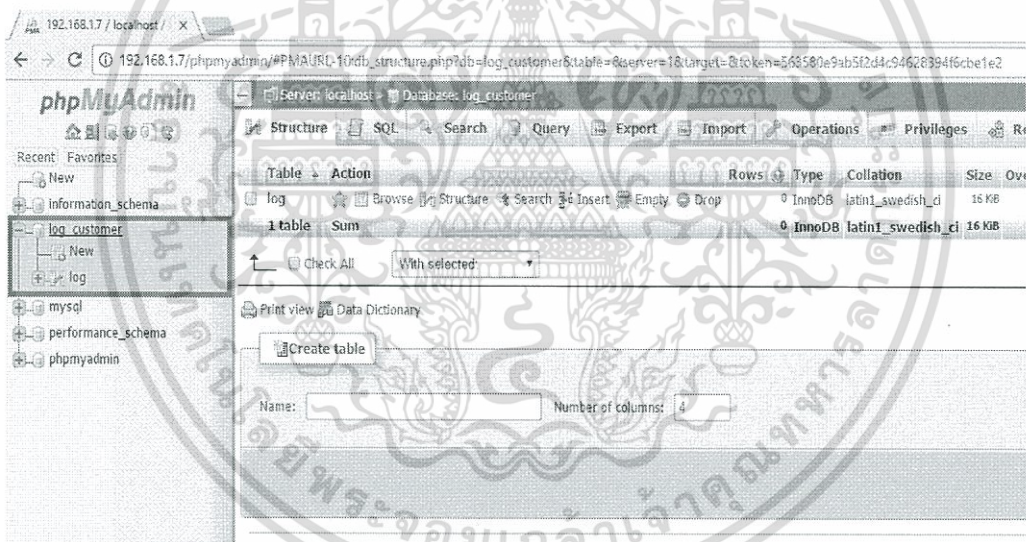
4.6.1 การสร้างตาราง สำหรับเก็บข้อมูลใน phpMyAdmin

การใช้งาน phpMyAdmin จำเป็นจะต้องใส่แอดเดรสของอุปกรณ์ที่ใช้สำหรับเข้าสู่ phpMyAdmin ตามด้วย /phpmyadmin และทำการเข้าสู่ระบบดังรูปที่ 4.14



รูปที่ 4.14 การเข้าสู่ระบบของ phpMyAdmin

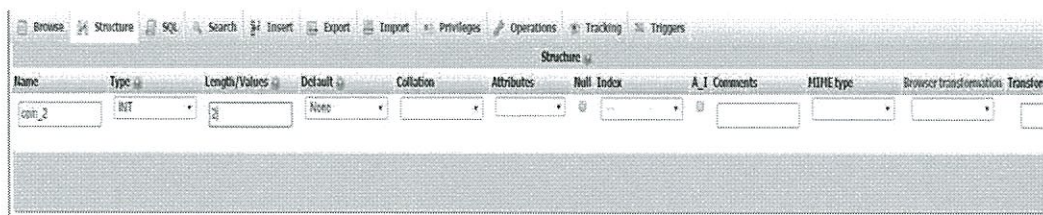
การเลือกฐานข้อมูลที่สร้างไว้ ดังรูปที่ 4.15



รูปที่ 4.15 เลือกฐานข้อมูล

สร้างตาราง (table) สำหรับเก็บข้อมูล ดังรูปที่ 4.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 สร้างตารางสำหรับเก็บข้อมูล

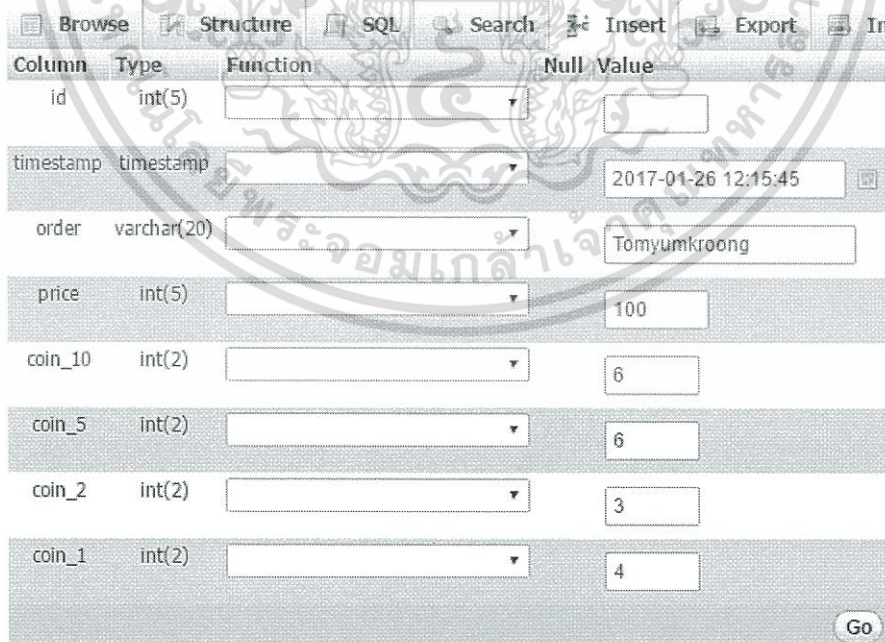
สร้างตารางทั้งหมดสำหรับเก็บข้อมูล ดังรูปที่ 4.17 แสดงโครงสร้างของฐานข้อมูลที่สร้างขึ้น



รูปที่ 4.17 โครงสร้างของระบบฐานข้อมูลสำหรับเก็บบันทึกข้อมูลการสั่งอาหาร

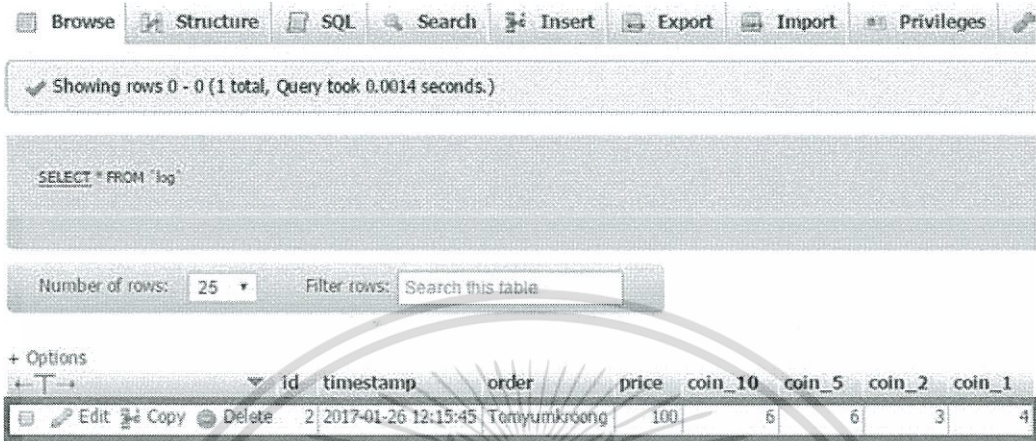
4.6.2 ทดสอบการเพิ่มข้อมูลเข้าสู่ระบบด้วยการใช้ฟังก์ชันภายใน phpMyAdmin

ในการเพิ่มข้อมูลเข้าสู่ระบบสามารถทำได้โดยการเลือก Insert แล้วกรอกข้อมูลลงไปตามคอลัมน์ต่างๆ ที่สร้างไว้ ดังแสดงในรูปที่ 4.18



รูปที่ 4.18 ทดสอบการเพิ่มข้อมูลเข้าไปในระบบ

เมื่อทำการเพิ่มข้อมูลเข้าสู่ระบบแล้ว จะสามารถตรวจสอบได้โดยการเลือกหัวข้อ Browse จะแสดงข้อมูลที่อยู่ในตาราง ที่ทำการเลือกไว้ ดังแสดงในรูปที่ 4.19



รูปที่ 4.19 ข้อมูลที่แสดงในระบบฐานข้อมูล

4.6.3 ทดสอบการเพิ่มข้อมูลเข้าสู่ระบบด้วยการเขียนโปรแกรมด้วยภาษาพีเอชพี

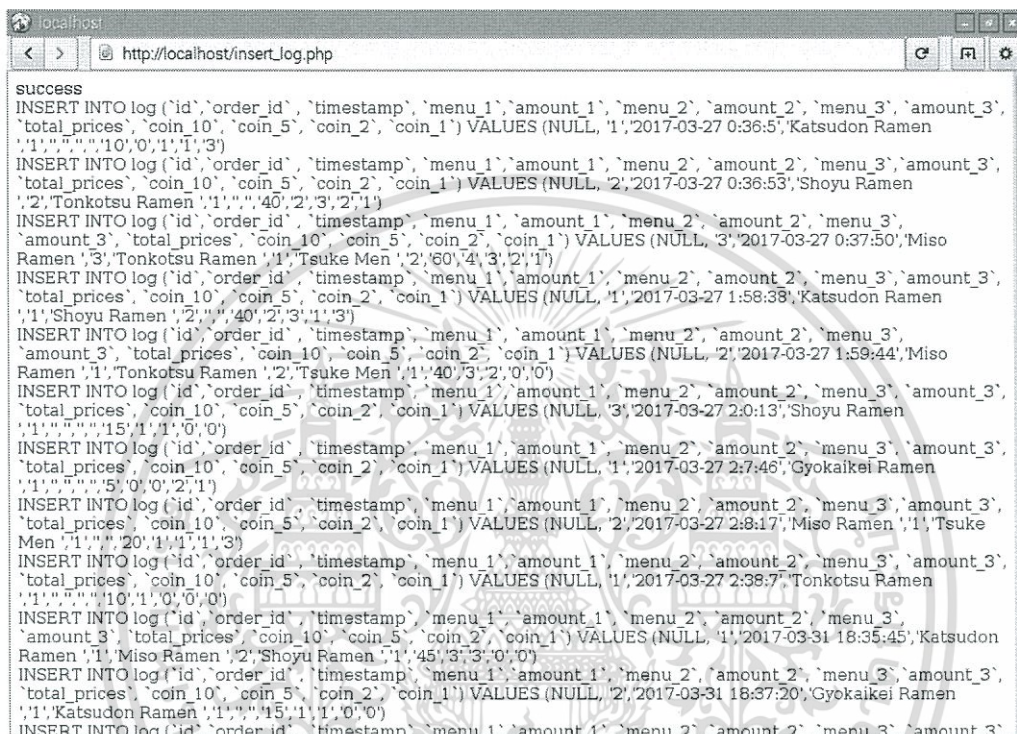
ในการเพิ่มข้อมูลเข้าสู่ระบบฐานข้อมูลด้วยการเขียนโปรแกรมภาษาพีเอชพี สามารถทำได้เช่นกัน โดยจำเป็นจะต้องทำการติดตั้งโปรแกรมพีเอชพี ลงบนบอร์ด Raspberry Pi ก่อน จากนั้นนำไฟล์ข้อมูลสำหรับการเรียกใช้โปรแกรมไปวางบนพื้นที่สำหรับเรียกใช้ผ่านเบราว์เซอร์ จากนั้นทำการเรียกใช้ไฟล์โปรแกรมผ่านเบราว์เซอร์ ข้อมูลการใช้งานระบบสั่งอาหารอัตโนมัติ จะถูกเพิ่มลงในระบบฐานข้อมูลอัตโนมัติ

id	timestamp	order	price	coin_10	coin_5	coin_2	coin_1
1	2017-03-27 0:36:5	Katsudon Ramen	1	10	0	1	1,3
2	2017-03-27 0:36:53	Shoyu Ramen	2	Tonkotsu Ramen	1	40	2,3,2,1
3	2017-03-27 0:37:50	Miso Ramen	3	Tonkotsu Ramen	1	Tsuke Men	2,60,4,3,2,1
1	2017-03-27 1:58:38	Katsudon Ramen	1	Shoyu Ramen	2	40	2,3,1,3
2	2017-03-27 1:59:44	Miso Ramen	1	Tonkotsu Ramen	2	Tsuke Men	1,40,3,2,0,0
3	2017-03-27 2:0:13	Shoyu Ramen	1	15	1	1	0,0
1	2017-03-27 2:7:46	Gyokaikei Ramen	1	5	0	0	2,1
2	2017-03-27 2:8:17	Miso Ramen	1	Tsuke Men	1	20	1,1,1,3
1	2017-03-27 2:38:7	Tonkotsu Ramen	1	10	1	0	0,0
1	2017-03-31 18:35:45	Katsudon Ramen	1	Miso Ramen	2	Shoyu Ramen	1,45,3,3,0,0
2	2017-03-31 18:37:20	Gyokaikei Ramen	1	Katsudon Ramen	1	15	1,1,0,0
3	2017-03-31 18:37:54	Tsuke Men	1	10	1	0	0,0
4	2017-03-31 18:38:58	Shoyu Ramen	3	Tonkotsu Ramen	1	55	3,3,2,6
1	2017-04-01 3:20:23	Katsudon Ramen	2	Tonkotsu Ramen	1	30	2,1,1,3

รูปที่ 4.20 ไฟล์ข้อมูลที่ถูกบันทึกหลังจากการสั่งอาหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.20 เป็นรูปไฟล์ข้อมูลที่ใช้สำหรับบันทึกข้อมูลการสั่งอาหารหลังจากผู้ใช้งานทำการหยอดเหรียญครบตามจำนวนราคาอาหารที่ได้สั่งไว้ โดยข้อมูลที่บันทึกจะมีเวลาที่ทำการสั่งอาหาร เมนูอาหารพร้อมจำนวนอาหารที่สั่งต่อเมนู ราคารวมต่อการสั่งอาหารในแต่ละครั้ง และจำนวนเหรียญทั้งหมดที่ผู้ใช้งานได้ชำระตามราคาอาหาร



```

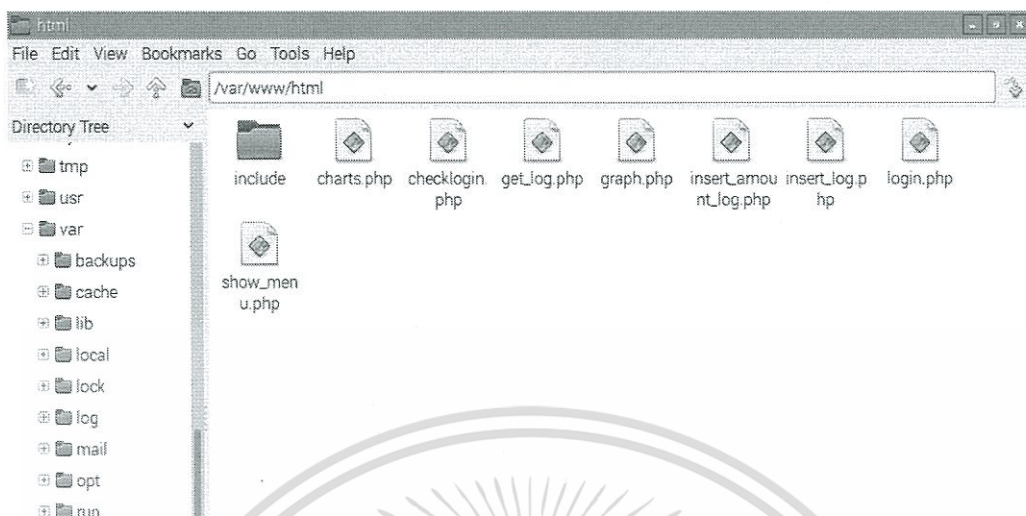
localhost
http://localhost/insert_log.php

success
INSERT INTO log ('id', 'order_id', 'timestamp', 'menu_1', 'amount_1', 'menu_2', 'amount_2', 'menu_3', 'amount_3',
'total_prices', 'coin_10', 'coin_5', 'coin_2', 'coin_1') VALUES (NULL, '1', '2017-03-27 0:36:57', 'Katsudon Ramen
', '1', '0', '0', '1', '1', '3')
INSERT INTO log ('id', 'order_id', 'timestamp', 'menu_1', 'amount_1', 'menu_2', 'amount_2', 'menu_3', 'amount_3',
'total_prices', 'coin_10', 'coin_5', 'coin_2', 'coin_1') VALUES (NULL, '2', '2017-03-27 0:36:53', 'Shoyu Ramen
', '2', 'Tonkotsu Ramen ', '1', '0', '2', '3', '2', '1')
INSERT INTO log ('id', 'order_id', 'timestamp', 'menu_1', 'amount_1', 'menu_2', 'amount_2', 'menu_3',
'amount_3', 'total_prices', 'coin_10', 'coin_5', 'coin_2', 'coin_1') VALUES (NULL, '3', '2017-03-27 0:37:50', 'Miso
Ramen ', '3', 'Tonkotsu Ramen ', '1', 'Tsuke Men ', '2', '60', '4', '3', '2', '1')
INSERT INTO log ('id', 'order_id', 'timestamp', 'menu_1', 'amount_1', 'menu_2', 'amount_2', 'menu_3', 'amount_3',
'total_prices', 'coin_10', 'coin_5', 'coin_2', 'coin_1') VALUES (NULL, '1', '2017-03-27 1:58:38', 'Katsudon Ramen
', '1', 'Shoyu Ramen ', '2', '0', '40', '2', '3', '1', '3')
INSERT INTO log ('id', 'order_id', 'timestamp', 'menu_1', 'amount_1', 'menu_2', 'amount_2', 'menu_3',
'amount_3', 'total_prices', 'coin_10', 'coin_5', 'coin_2', 'coin_1') VALUES (NULL, '2', '2017-03-27 1:59:44', 'Miso
Ramen ', '1', 'Tonkotsu Ramen ', '2', 'Tsuke Men ', '1', '40', '3', '2', '0', '0')
INSERT INTO log ('id', 'order_id', 'timestamp', 'menu_1', 'amount_1', 'menu_2', 'amount_2', 'menu_3', 'amount_3',
'total_prices', 'coin_10', 'coin_5', 'coin_2', 'coin_1') VALUES (NULL, '3', '2017-03-27 2:0:13', 'Shoyu Ramen
', '1', '0', '15', '1', '1', '0', '0')
INSERT INTO log ('id', 'order_id', 'timestamp', 'menu_1', 'amount_1', 'menu_2', 'amount_2', 'menu_3', 'amount_3',
'total_prices', 'coin_10', 'coin_5', 'coin_2', 'coin_1') VALUES (NULL, '1', '2017-03-27 2:7:46', 'Gyokaikei Ramen
', '1', '0', '5', '0', '0', '2', '1')
INSERT INTO log ('id', 'order_id', 'timestamp', 'menu_1', 'amount_1', 'menu_2', 'amount_2', 'menu_3', 'amount_3',
'total_prices', 'coin_10', 'coin_5', 'coin_2', 'coin_1') VALUES (NULL, '2', '2017-03-27 2:8:17', 'Miso Ramen ', '1', 'Tsuke
Men ', '1', '0', '20', '1', '1', '1', '3')
INSERT INTO log ('id', 'order_id', 'timestamp', 'menu_1', 'amount_1', 'menu_2', 'amount_2', 'menu_3', 'amount_3',
'total_prices', 'coin_10', 'coin_5', 'coin_2', 'coin_1') VALUES (NULL, '1', '2017-03-27 2:38:7', 'Tonkotsu Ramen
', '1', '0', '10', '1', '0', '0', '0')
INSERT INTO log ('id', 'order_id', 'timestamp', 'menu_1', 'amount_1', 'menu_2', 'amount_2', 'menu_3', 'amount_3',
'amount_3', 'total_prices', 'coin_10', 'coin_5', 'coin_2', 'coin_1') VALUES (NULL, '1', '2017-03-31 18:35:45', 'Katsudon
Ramen ', '1', 'Miso Ramen ', '2', 'Shoyu Ramen ', '1', '45', '3', '3', '0', '0')
INSERT INTO log ('id', 'order_id', 'timestamp', 'menu_1', 'amount_1', 'menu_2', 'amount_2', 'menu_3', 'amount_3',
'total_prices', 'coin_10', 'coin_5', 'coin_2', 'coin_1') VALUES (NULL, '2', '2017-03-31 18:37:20', 'Gyokaikei Ramen
', '1', 'Katsudon Ramen ', '1', '0', '15', '1', '1', '0', '0')
INSERT INTO log ('id', 'order_id', 'timestamp', 'menu_1', 'amount_1', 'menu_2', 'amount_2', 'menu_3', 'amount_3'

```

รูปที่ 4.21 ทำการเรียกใช้โปรแกรมผ่านเว็บเบราว์เซอร์

จากรูปที่ 4.21 เมื่อเรียกใช้งานโปรแกรมผ่านเว็บเบราว์เซอร์ โปรแกรมจะทำการเชื่อมต่อไปยังระบบฐานข้อมูลที่ตั้งค่าไว้ จากนั้นจะทำการเปิดอ่านข้อมูลจากไฟล์ที่ทำการบันทึกข้อมูลการใช้งานระบบสั่งอาหารอัตโนมัติและทำการเพิ่มข้อมูลทั้งหมดที่อ่านได้จากไฟล์ข้อมูลขึ้นไปยังระบบฐานข้อมูล ในรูปที่ 4.21 จะมีการแสดงสถานะในการเชื่อมต่อไปยังฐานข้อมูลในบรรทัดที่ 1 (success) และในบรรทัดถัดไปจะเป็นการแสดงผลข้อมูลทั้งหมดที่ถูกเพิ่มเข้าไปยังฐานข้อมูล



รูปที่ 4.22 พื้นที่เก็บไฟล์สำหรับเรียกใช้งานผ่านเว็บเบราว์เซอร์

ในการเรียกใช้งานโปรแกรมภาษาพีเอชพี (PHP) ผ่านเว็บเบราว์เซอร์นั้น ต้องนำไฟล์ข้อมูลมาเก็บลงในที่อยู่ตามรูปที่ 4.22 จึงจะสามารถเรียกใช้งานผ่านเว็บเบราว์เซอร์ได้

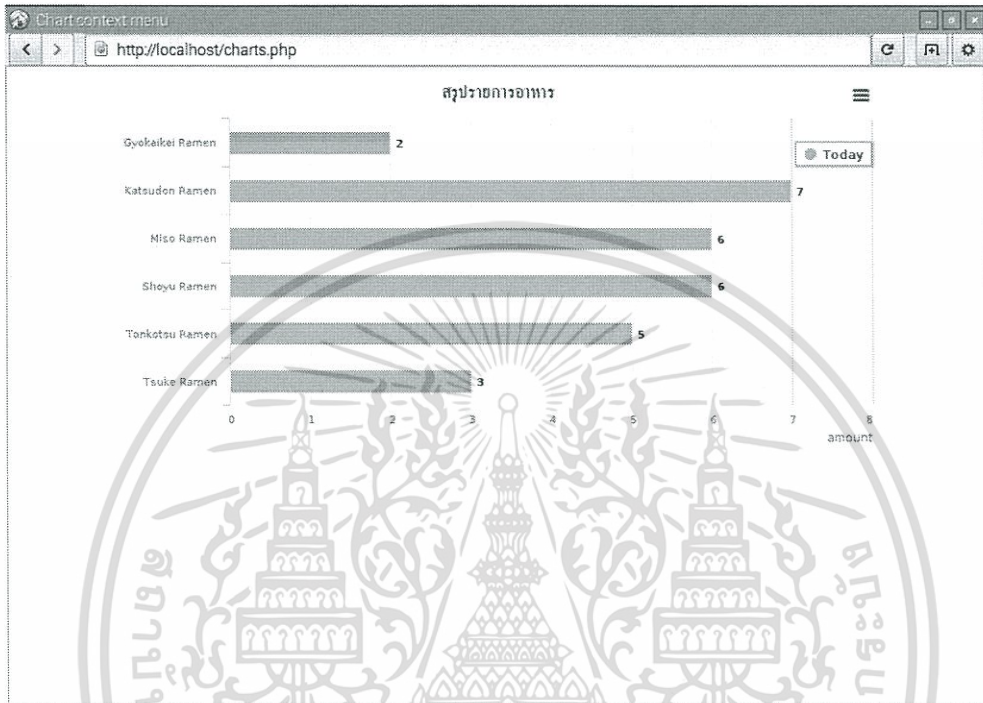
4.7 ทดสอบการแสดงผลการสรุปผลการสั่งอาหาร

ผู้จัดทำได้ออกแบบกราฟสำหรับแสดงผลสรุปของการใช้งานระบบสั่งอาหารอัตโนมัติ ในการสั่งอาหารในแต่ละวัน และทำการแสดงผลออกทางเว็บเบราว์เซอร์

รูปที่ 4.23 การยืนยันตัวตนสำหรับเข้าถึงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการเข้าถึงข้อมูลในส่วนผลสรุปการใช้งานระบบสั่งอาหารอัตโนมัติในแต่ละวัน จำเป็นจะต้องมีการยืนยันตัวตนสำหรับการเข้าถึงข้อมูล เพื่อความปลอดภัยของผู้ประกอบการ ดังนั้นผู้จัดทำจึงทำการออกแบบระบบการยืนยันตัวตนสำหรับการเข้าถึงข้อมูล ดังรูปที่ 4.23



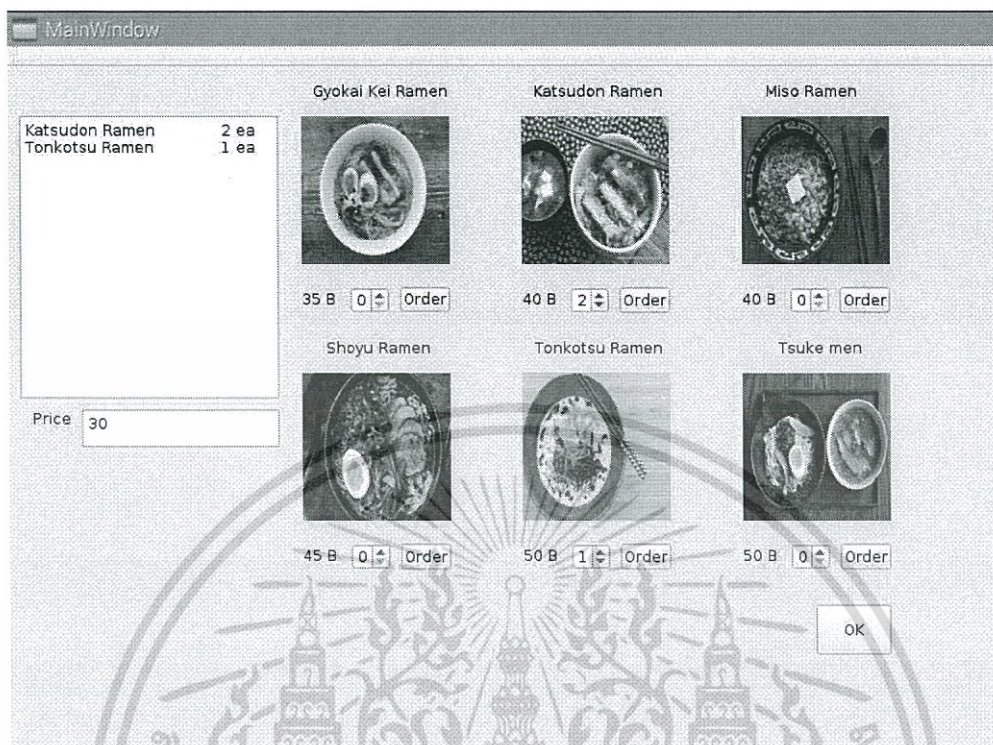
รูปที่ 4.24 กราฟแสดงผลสรุปการสั่งอาหารทั้งหมด

เมื่อทำการยืนยันตัวตนสำเร็จระบบจะทำการแสดงผลสรุปการสั่งอาหารทั้งหมด โดยข้อมูลที่แสดงจะประกอบไปด้วยชื่อเมนูอาหาร และจำนวนที่ถูกสั่งภายในวันนั้น ดังรูปที่ 4.24

4.8 การทดสอบการทำงานของส่วนติดต่อกับผู้ใช้งานหลังจากพัฒนาฟังก์ชัน

ผู้จัดทำได้พัฒนาฟังก์ชันในการสั่งอาหารให้กับระบบ โดยระบบสามารถสั่งอาหารได้พร้อมกันหลายเมนู และเลือกจำนวนอาหารของแต่ละเมนูที่ต้องการต่อการสั่งหนึ่งครั้ง จากเดิมที่สามารถสั่งได้ครั้งละหนึ่งเมนูและไม่สามารถเลือกจำนวนได้ ทำให้ระบบมีความสามารถที่หลากหลายและสะดวกยิ่งขึ้น

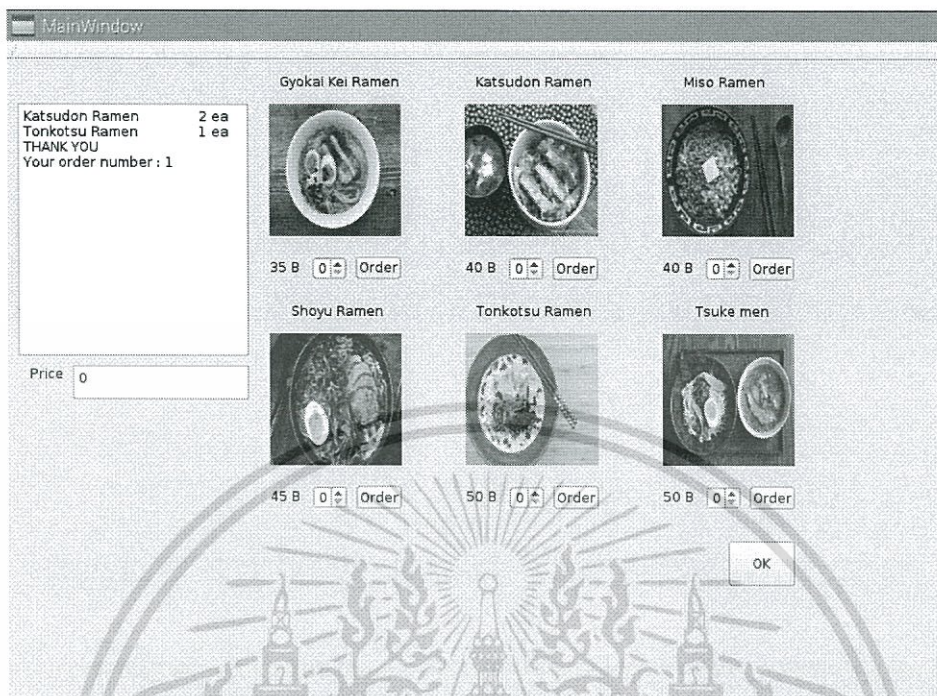
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 ส่วนติดต่อกับผู้ใช้งานหลังจากพัฒนาและปรับปรุงคุณสมบัติการทำงาน

จากรูปที่ 4.25 เป็นรูปที่แสดงหน้าจอสำหรับผู้ใช้งานหลังจากที่ผู้จัดทำได้ทำการพัฒนาและปรับปรุงฟังก์ชันในการสั่งอาหารแล้ว โดยมีขั้นตอนในการใช้งาน โดยเริ่มจากเลือกจำนวนเมนูอาหารที่ต้องการด้วยการกดปุ่มลูกศรขึ้นบริเวณด้านล่างของเมนูที่ต้องการ เมื่อทำการเลือกจำนวนอาหารแล้ว จากนั้นทำการกดปุ่ม "Order" เมนูที่ต้องการ หลังจากเลือกเมนูอาหารที่ต้องการทั้งหมด ให้ทำการกดปุ่ม "OK" เพื่อยืนยันการสั่งอาหาร ต่อมาทำการหยุดเหรียญลงในเครื่องรับเหรียญตามจำนวนราคาอาหารที่สั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.26 ส่วนติดต่อกับผู้ใช้งานหลังจากทำการชำระราคาอาหารครบ

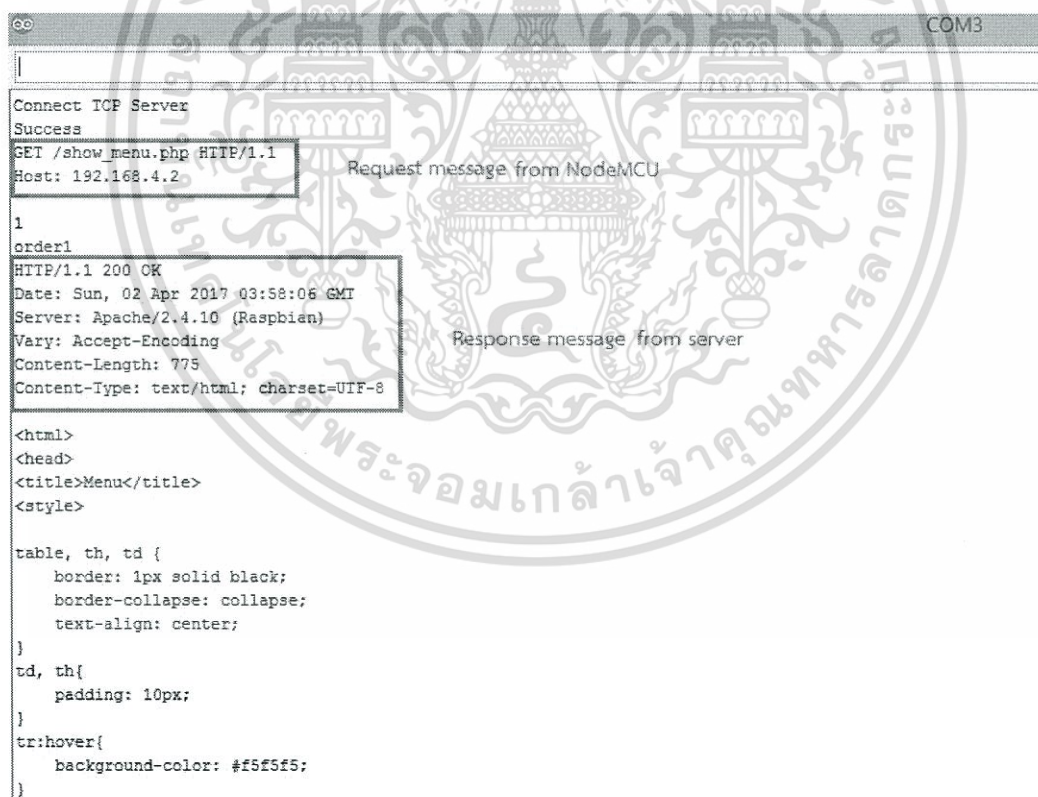
เมื่อทำการหยุดเหรียญครบตามจำนวนราคาอาหารที่สั่งแล้ว ระบบจะแสดงผลเลขลำดับคิวสำหรับการรอรับอาหาร ซึ่งเลขที่แสดงนั้นจะไม่ซ้ำกับลูกค้าท่านอื่น ดังรูปที่ 4.26

4.9 การทดสอบการร้องขอ (Request) ข้อมูลจากฐานข้อมูลบนเซิร์ฟเวอร์

ผู้จัดทำได้ออกแบบ โดยทางห้องครัวจะต้องกดปุ่มเพื่อร้องขอข้อมูลการสั่งอาหารของลูกค้า จากนั้นข้อมูลจึงจะแสดงผลบนจอแอลซีดี โดยเมื่อทำการกดปุ่ม ทางด้าน NodeMCU จะส่ง HTTP GET request ไปให้ทางเซิร์ฟเวอร์ที่เก็บข้อมูลการสั่งอาหารของลูกค้าไว้ ดังแสดงในรูปที่ 4.27 จากนั้นหากได้รับการตอบกลับจากทางเซิร์ฟเวอร์ NodeMCU จะทำการดึงข้อมูลจากฐานข้อมูลมาแสดงผลบนจอแอลซีดี ดังรูปที่ 4.28 และรูปที่ 4.29



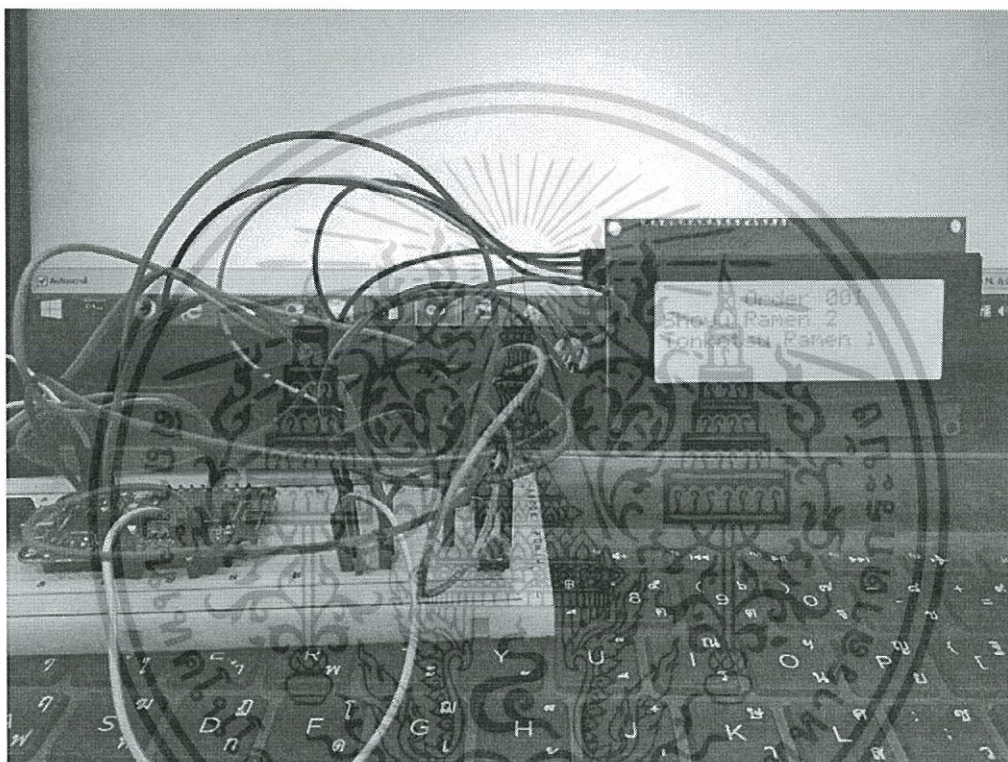
รูปที่ 4.27 ฐานข้อมูลบนเซิร์ฟเวอร์ที่เก็บข้อมูลการสั่งอาหารไว้



รูปที่ 4.28 Serial monitor แสดงผลการติดต่อระหว่าง NodeMCU กับเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

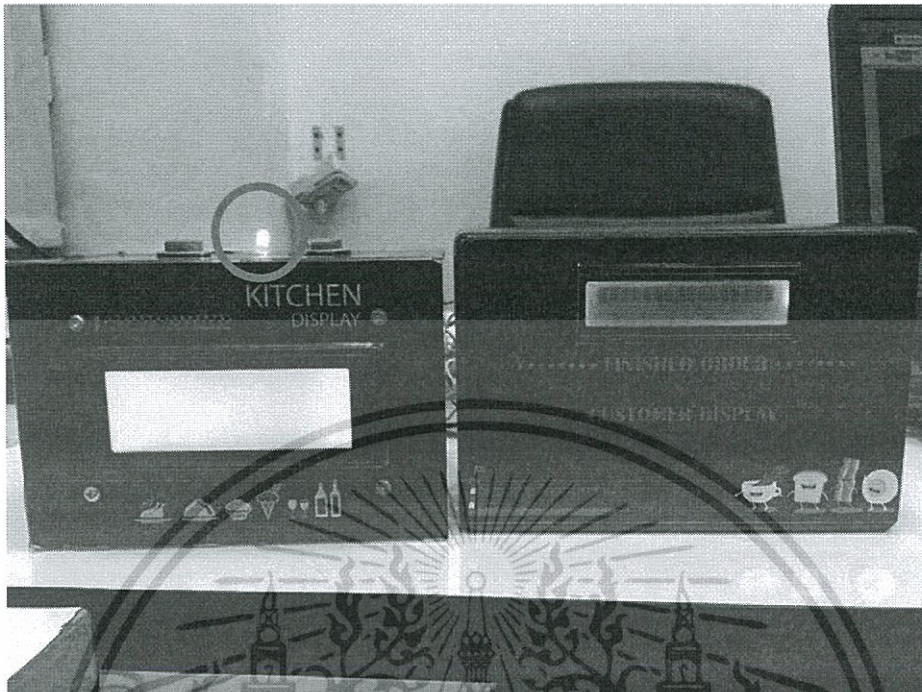
จากรูปที่ 4.28 จะสังเกตว่าข้อความร้องขอจากทาง NodeMCU ได้ส่งไปที่เซิร์ฟเวอร์ หมายเลขไอพีคือ 192.168.4.2 แล้วร้องขอไฟล์ข้อมูล show_menu.php จากนั้นทางเซิร์ฟเวอร์ ได้ส่งข้อความตอบรับกลับมาคือ “HTTP/1.1 200 OK” ซึ่งหมายถึงการร้องขอข้อมูลบนเซิร์ฟเวอร์ สามารถทำได้สำเร็จ และข้อมูลเหล่านั้น จะถูกส่งกลับไปยัง NodeMCU โดยข้อมูลรายการอาหาร จะถูกนำมาแสดงบนจอแอลซีดีตามลำดับ ดังรูปที่ 4.29 ซึ่งจะเห็นว่าข้อมูลบนจอแอลซีดีตรงตาม ข้อมูลจากทางเซิร์ฟเวอร์



รูปที่ 4.29 จอแอลซีดีแสดงผลข้อมูลรายการอาหาร

4.10 การทดสอบการแจ้งเตือนผ่านหลอดไฟ LED เมื่อมีการสั่งอาหารเข้ามา

ผู้จัดทำได้ออกแบบ โดยให้ทางห้องครัวมีฟังก์ชันการตรวจสอบการสั่งอาหาร จากทางลูกค้า โดยตรวจสอบว่ามีการสั่งอาหารเข้ามาใหม่หรือไม่ หากลูกค้าได้สั่งอาหารเข้ามาใหม่ แล้วทางห้องครัวยังไม่ได้กรับข้อมูล หลอดไฟ LED จะติดสว่างขึ้น เพื่อแจ้งเตือนให้ทางฝั่งห้องครัว ทราบ ดังแสดงในรูปที่ 4.30 และรูปที่ 4.31



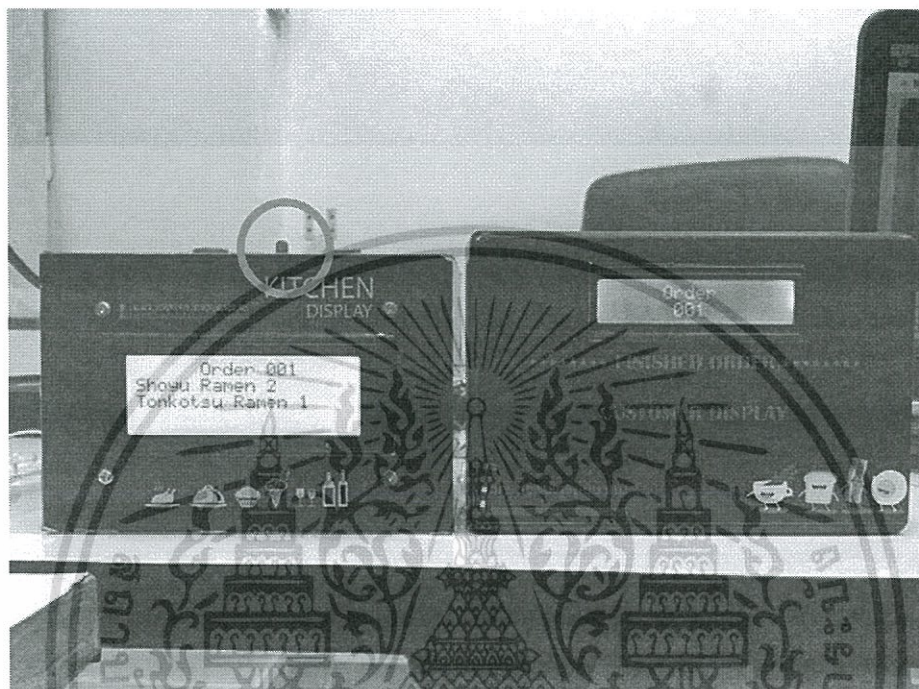
รูปที่ 4.30 ส่วนแสดงผลฝั่งห้องครัวแจ้งเตือนว่ามีการสั่งอาหารเข้ามา



รูปที่ 4.31 การแจ้งเตือนด้วยหลอดไฟ LED เมื่อมีการสั่งอาหารเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยเขียนโปรแกรมให้มีการตรวจสอบข้อมูลกับทางเซิร์ฟเวอร์ตลอดเวลา แล้วเมื่อทางห้องครัวกดรับข้อมูลการสั่งอาหารล่าสุดแล้ว และไม่มีการสั่งอาหารเพิ่มเข้ามาแล้ว หลอดไฟ LED จะดับลง ดังแสดงในรูปที่ 4.32



รูปที่ 4.32 หลอดไฟ LED ดับลงเมื่อไม่มีการสั่งอาหารเพิ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปริญญานิพนธ์เรื่อง “ระบบสั่งอาหารอัตโนมัติ” ที่ได้จัดทำขึ้น ในการจัดทำระบบสั่งอาหารอัตโนมัติ ใช้บอร์ด Raspberry Pi เป็นตัวประมวลผล สั่งการ และใช้สำหรับสร้างส่วนการรับข้อมูลจากผู้ใช้งาน โดยจะแสดงข้อมูลราคาอาหารบนหน้าจอ และรอรับเหรียญตามราคาอาหารจากข้อมูลเมนูอาหารที่ได้รับ เมื่อได้รับเหรียญจนครบ Raspberry Pi จะทำการส่งข้อมูลเมนูอาหารที่ได้สั่งไว้ไปเก็บไว้ในฐานข้อมูลบนเซิร์ฟเวอร์ โดยมีเครื่องรับเหรียญใช้สำหรับรับเหรียญจากผู้ใช้งานระบบ ในส่วนของไมโครคอนโทรลเลอร์ชนิด NodeMCU โมดูล ESP8266 ใช้สำหรับแสดงข้อมูลอาหารในส่วนของห้องครัว NodeMCU จะตรวจสอบข้อมูลกับทางเซิร์ฟเวอร์ตลอดเวลา หากมีการสั่งอาหารเข้ามาจะแจ้งเตือนให้ทางห้องครัวทราบผ่านหลอดไฟ LED โดยทาง NodeMCU จะต้องส่งการร้องขอไปที่เซิร์ฟเวอร์ที่เก็บข้อมูล ถ้าได้รับการตอบกลับจากเซิร์ฟเวอร์ NodeMCU จึงสามารถนำข้อมูลมาแสดงบนจอแอลซีดีได้

5.2 ข้อเสนอแนะ

เครื่องรับเหรียญที่ใช้งานนั้นไม่สามารถรับธนบัตรและไม่สามารถทอนเหรียญให้ผู้ใช้งานได้ ซึ่งผู้ใช้งานจะต้องมีเหรียญครบตามจำนวนราคาอาหาร ทำให้ในบางโอกาสการสั่งอาหารด้วยระบบสั่งอาหารอัตโนมัติมีคุณภาพด้อยกว่าการสั่งอาหารแบบปกติ และในส่วนของแสดงผลสำหรับห้องครัว ในด้านการใช้งานในครัวที่มีขนาดใหญ่ จอแอลซีดีที่ใช้มีขนาดเล็กเกินไป ทำให้ไม่เพียงพอต่อการแสดงข้อมูลเมนูอาหารจากผู้ใช้งานจำนวนมากได้

บรรณานุกรม

- [1] Aaron Shaw. “Raspberry Leaf – Time Saving GPIO Reference Guide.”
<https://www.pi-supply.com/raspberry-leaf-time-saving-gpio-reference-guide/>.
- [2] Home of Maker. “ทำความเข้าใจกับบอร์ด Raspberry Pi.”
<http://www.homeofmaker.com/?p=891>.
- [3] “QDebug Class.” <http://doc.qt.io/qt-5/qdebug.html>.
- [4] “QFile Class.” <http://doc.qt.io/qt-5/qfile.html>.
- [5] “QTextStream Class.” <http://doc.qt.io/qt-4.8/qtextstream.html>.
- [6] Terry King, Mary Alice Osborne, Jun Yuan Peng, Barry King and other contributors. “LCD Displays (Blue and Yellow) with I2C/TWI Interface.”
[Online]. Available : <https://arduino-info.wikispaces.com/LCD-Blue-I2C>. 2016.
- [7] Vimal Mayank, Deep Saraf. “Fast Food Automated Ordering System.”
<https://www.isr.umd.edu/~austin/ense621.d/projects04.d/project-food-ordering.html#sec1>.
- [8] Vivek Kartha. “Led Blinking using Raspberry Pi – Python.”
<https://electrosome.com/led-blinking-raspberry-pi/>.
- [9] Willies Computer Software Co. “INTRODUCTION TO RS232 SERIAL COMMUNICATION.” <https://wscsnet.com/tutorials/introduction-to-rs232-serial-communication/>.
- [10] กมลศักดิ์ มลิอ่อนง, ชีระยุทธ แซ่ลิ้ม. “HTTP.” http://staff.cs.psu.ac.th/noi/cs344-481/group11_Http/HTTP.htm.
- [11] อานนท์ ฅ นองคาย. “ESP8266 NodeMCU คืออะไร? และการติดตั้ง ESP8266 NodeMCU บน Arduino IDE.”
<https://embeddedsystem2558.wordpress.com/esp8266-nodemcu-คืออะไร-และการติดตั้ง-e>.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ดควบคุมการทำงานส่วนของไมโครคอนโทรลเลอร์ NodeMCU ESP8266

```
#include <ESP8266WiFi.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#define SERVER_PORT 80

////////////////////////////////////
// WiFi Definitions //
////////////////////////////////////
const char WiFiAPPSK[] = "123456789";

IPAddress serverip(192, 168, 4, 2); //Database server IP

/*Set the LCD address to 0x27 for LCD1 and 0x26 for LCD2.*/
LiquidCrystal_I2C lcd1(0x27, 20, 4);
LiquidCrystal_I2C lcd2(0x26, 16, 2);

////////////////////////////////////
// Pin Definitions //
////////////////////////////////////
const int buttonPin = 14;
const int reqPin = 12;
int buttonState = 0;
int prevState = 0;
int request = 0;
int cntorder = 1;

char data[10];
int hasData = 0;

String str_get1 = "GET /get_menu.php HTTP/1.1\r\n";
String str_get2 = "GET /show_menu2.php?order=";
String str_get3 = " HTTP/1.1\r\n";
String str_host = "Host: 192.168.4.2\r\n\r\n";
char str[10];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unsigned long previousMillis = 0; //กำหนดตัวแปรเก็บค่า เวลาสุดท้ายที่ทำงาน
const long interval = 10000; //กำหนดค่าตัวแปร ให้ทำงานทุกๆ 10 วินาที

WiFiServer server(SERVER_PORT);
WiFiClient client;

void setup() {
  pinMode(D0, OUTPUT);
  pinMode(buttonPin, INPUT);
  pinMode(reqPin, INPUT);

  setupWiFi();
  server.begin();
  Serial.begin(115200);
  lcd1.begin();
  lcd2.begin();
}

void loop() {
  unsigned long currentMillis = millis();
  /*อ่านค่าเวลาที่ ESP เริ่มทำงานจนถึงเวลาปัจจุบัน*/
  if (currentMillis - previousMillis >= interval)
  /*ถ้าหากเวลาปัจจุบันลบกับเวลาก่อนหน้านี้ มีค่ามากกว่าค่า interval ให้คำสั่งภายใน if ทำงาน*/
  {
    previousMillis = currentMillis;
    /*ให้เวลาปัจจุบัน เท่ากับ เวลาก่อนหน้าเพื่อใช้คำนวณเงื่อนไขในรอบถัดไป*/
    Client_RequestA(); //เรียกใช้งานฟังก์ชัน Client_RequestA
  }
  while (client.available()) //ตรวจเช็คว่ามีคำสั่งส่งค่ากลับมาจาก Server หรือไม่
  {
    String check = client.readStringUntil('\n');
    Serial.println(check); //แสดงค่าที่ได้ทาง Serial Monitor
    Serial.println("check status");
    if (check == "hello") {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.println("Led is ON");
digitalWrite(D0, HIGH);
}
else if (check == "goodbye") {
  Serial.println("Led is OFF");
  digitalWrite(D0, LOW);
}
}

```

```

buttonState = digitalRead(buttonPin);
if (buttonState == HIGH)
{
  delay(100);
  if (prevState == 0)
  {
    lcd2.clear();
    lcd2.setCursor(5, 0);
    lcd2.print("Order");
    lcd2.setCursor(6, 1);
    lcd2.print(data);
    prevState = 1;
    Serial.println("sent to customers");
  }
}
else
{
  if (prevState == 1)
  {
    prevState = 0;
  }
}

```

```

request = digitalRead(reqPin);
if (request == HIGH)
{

```

```

  lcd1.clear();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Client_Request(); //เรียกใช้งานฟังก์ชัน Client_Request

delay(100);
while (client.available()) //ตรวจเช็คว่ามีคำสั่งส่งค่ากลับมาจาก Server หรือไม่
{
    String line = client.readStringUntil('\n'); //อ่านค่าที่ Server ตอบกลับมาทีละบรรทัด
    Serial.println(line);
    Serial.println("Read data");

    get_data1 = cut_string(line, str, "menu1"); //ตัด string ข้อมูลส่วน menu1
    if (get_data1 != "NULL")
    {
        sprintf(data, "00%d", cntorder);
        Serial.print("Order ");
        Serial.println(data);
        Serial.print("menu1");
        Serial.println(get_data1);
        lcd1.setCursor(6, 0);
        lcd1.print("Order ");
        lcd1.print(data);
        lcd1.setCursor(0, 1);
        lcd1.print(get_data1);

        get_data2 = cut_string(line, str, "menu2"); //ตัด string ข้อมูลส่วน menu2
        if (get_data2 != "NULL")
        {
            Serial.print("menu2");
            Serial.println(get_data2);
            lcd1.setCursor(0, 2);
            lcd1.print(get_data2);

            get_data3 = cut_string(line, str, "menu3"); //ตัด string ข้อมูลส่วน menu3
            if (get_data3 != "NULL")
            {
                Serial.print("menu3");
                Serial.println(get_data3);
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        lcd1.setCursor(0, 3);
        lcd1.print(get_data3);
    }
}
cntorder++;
Serial.print(cntorder);
}
else {
}
}
}
}

void Client_RequestA()
{
    Serial.println("Connect TCP Server");
    int cnt = 0;
    while (!client.connect(serverip, SERVER_PORT))
/*เชื่อมต่อกับ Server และรอจนกว่าเชื่อมต่อสำเร็จ*/
    {
        Serial.print(".");
        delay(100);
        cnt++;
        if (cnt > 50)
            //ถ้าหากใช้เวลาเชื่อมต่อเกิน 5 วินาที ให้ออกจากฟังก์ชัน
            return;
    }
    Serial.println("Check data from server");
    delay(500);
    client.print(str_get1 + str_host);    //ส่งคำสั่ง HTTP GET ไปยัง Server
    Serial.println("----HTTP Get request----");
    Serial.print(str_get1 + str_host);
    delay(100);
}

void Client_Request()
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.println("Connect TCP Server");
int cntt = 0;
while (!client.connect(serverip, SERVER_PORT))
/*เชื่อมต่อกับ Server และรอจนกว่าเชื่อมต่อสำเร็จ*/
{
  Serial.print(".");
  delay(100);
  cntt++;
  if (cntt > 50) //ถ้าหากใช้เวลาเชื่อมต่อเกิน 5 วินาที ให้ออกจากฟังก์ชัน
    return;
}
Serial.println("Send HTTP GET request success");
delay(500);
client.print(str_get2 + cntorder + str_get3 + str_host);
/*ส่งคำสั่ง HTTP GET ไปยัง Server*/
Serial.println("----HTTP Get request----");
Serial.print(str_get2 + cntorder + str_get3 + str_host);
Serial.print("Order request : ");
sprintf(str, "order%d", cntorder);
Serial.println(str);
Serial.println("\r\n----Response from server----");
delay(100);
}

String cut_string(String input, char header[10], String get_string)
{
  if (input.indexOf(header) != -1)
/*ตรวจสอบว่าใน input มีข้อความเหมือนใน header หรือไม่*/
  {
    int num_get = input.indexOf(get_string); //หาคำแหน่งของข้อความ get_string ใน input
    if (num_get != -1) //ตรวจสอบว่าตำแหน่งที่ได้ไม่ใช่ -1 (ไม่มีข้อความ get_string ใน input)
    {
      int start_val = input.indexOf("=", num_get) + 1; //หาคำแหน่งแรกที่ "="
      int stop_val = input.indexOf(",", start_val); //หาคำแหน่งสุดท้ายที่ ","
      if (start_val == stop_val) //แสดงว่าไม่มีข้อมูล
      {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return "NULL";
    }
    else
    {
        return (input.substring(start_val, stop_val));
/*ตัดเอาข้อความระหว่าง "=" และ "," มาใช้*/
    }
}
else
{
    return ("NULL"); //Return ข้อความ NULL เมื่อไม่ตรงเงื่อนไข
}
return ("NULL"); //Return ข้อความ NULL เมื่อไม่ตรงเงื่อนไข
}

void setupWiFi()
{
    WiFi.mode(WIFI_AP);
/*Do a little work to get a unique-ish name. Append the last two bytes of the MAC
(HEX'd) to "Thing-";*/
    uint8_t mac[WL_MAC_ADDR_LENGTH];
    WiFi.softAPmacAddress(mac);
    String macID = String(mac[WL_MAC_ADDR_LENGTH - 2], HEX) +
        String(mac[WL_MAC_ADDR_LENGTH - 1], HEX);
    macID.toUpperCase();
    String AP_NameString = "ESP8266 " + macID;

    char AP_NameChar[AP_NameString.length() + 1];
    memset(AP_NameChar, 0, AP_NameString.length() + 1);

    for (int i = 0; i < AP_NameString.length(); i++)
        AP_NameChar[i] = AP_NameString.charAt(i);

    WiFi.softAP(AP_NameChar, WiFiAPPSK);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ดควบคุมการทำงานส่วนของบอร์ด Raspberry Pi

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <wiringPi.h>
#include <QtSerialPort/QSerialPort>
#include <QtSerialPort/QSerialPortInfo>
#include <QFile>
#include <QTextStream>
#include <QDateTime>
#include <QTimer>
#include <QBasicTimer>
#include <QDebug>

// define variable //
int amount1, amount2, amount3, amount4, amount5, amount6;
int p_1, p_2, p_3, p_4, p_5, p_6;
int total_prices;
int menu0=5;
int menu1=10;
int menu2=15;
int menu3=10;
int q=0,sum=0;
int count_down=0;
int Order,count,count2,coin_10,coin_5,coin_2,coin_1;
int num_coin[4] = {0,0,0,0};
char Enable[5] = {0x90,0x05,0x01,0x03,0x99};
char Disable[5] = {0x90,0x05,0x02,0x03,0x9A};
const int CON_COIN[4] = {1,2,5,10};
QSerialPort serial;
QString total = "0";
QByteArray HexEn(Enable, 5);
QByteArray HexDis(Disable, 5);
QByteArray enter = " ea\r";
QByteArray menu_1, menu_2, menu_3, menu_4, menu_5, menu_6;
QByteArray prices_1, prices_2, prices_3, prices_4, prices_5, prices_6;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// define menu and prices //
Food food[] =
{
    Food((QString)"GyokaiKei Ramen\t",menu0),
    Food((QString)"Katsudon Ramen\t",menu1),
    Food((QString)"Miso Ramen\t\t",menu1),
    Food((QString)"Shoyu Ramen\t",menu2),
    Food((QString)"Tonkotsu Ramen\t",menu3),
    Food((QString)"Tsuke Men\t\t",menu3)
};

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);

    // สร้างรูปภาพสำหรับแสดงรายการอาหาร
    QPixmap pic1("/home/pi/Desktop/Ramen/Gyokai Kei Ramen.jpg");
    ui->label_pic1->setPixmap(pic1);
    QPixmap pic2("/home/pi/Desktop/Ramen/KATSUDON.jpg");
    ui->label_pic2->setPixmap(pic2);
    QPixmap pic3("/home/pi/Desktop/Ramen/Miso Ramen.jpg");
    ui->label_pic3->setPixmap(pic3);
    QPixmap pic4("/home/pi/Desktop/Ramen/Shoyu Ramen.jpg");
    ui->label_pic4->setPixmap(pic4);
    QPixmap pic5("/home/pi/Desktop/Ramen/Tonkotsu Ramen.jpg");
    ui->label_pic5->setPixmap(pic5);
    QPixmap pic6("/home/pi/Desktop/Ramen/Tsuke Men.jpg");
    ui->label_pic6->setPixmap(pic6);

    // ceate connection to coin acceptor //
    serial.setPortName("ttyUSB0"); // กำหนดพอร์ทที่ใช้งานเป็น ttyUSB0
    serial.setBaudRate(serial.Baud9600);
    serial.setDataBits(serial.Data8);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

serial.setParity(serial.NoParity);
serial.setStopBits(serial.OneStop);
serial.setFlowControl(serial.NoFlowControl);
connect(&serial,SIGNAL(readyRead()),this,SLOT(readData()));
if(serial.open(QIODevice::ReadWrite)){
    serial.write(HexDis);
}
initial();
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::initial()
{
    ui->textEdit->clear();
    ui->textEdit_2->clear();
    ui->textEdit->insertPlainText("-----Welcome-----"); // กำหนดให้
แสดงข้อความ Welcome
}

int MainWindow::readData()
{
    // receive data from coin acceptor //
    QByteArray coin = serial.readAll(); // สร้างตัวแปรรับข้อมูลจาก serial port
    while (serial.waitForReadyRead(500)) // รับข้อมูลจนกว่าจะครบ 0.5 วินาที
        coin.append(serial.readAll());

    // สร้างเงื่อนไขในการคำนวณจำนวนหยอดที่ได้รับ
    if(coin.toHex() == "9006120403af"){
        sum+=10;
        num_coin[3]++;
        coin_10++;
    }else if(coin.toHex() == "9006120303ae"){
        sum+=5;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    num_coin[2]++;
    coin_5++;
}else if(coin.toHex() == "9006120503b0" || coin.toHex() == "9006120203ad"){
    sum+=2;
    num_coin[1]++;
    coin_2++;
}else if(coin.toHex() == "9006120603b1" || coin.toHex() == "9006120103ac"){
    sum+=1;
    num_coin[0]++;
    coin_1++;
}
total = QString::number(total_prices-sum); // สร้างตัวแปรสำหรับเก็บราคาอาหาร
if(total <= "0" && count == 1){
    q++;
    serial.write(HexDis); // ส่งคำสั่งยกเลิกการทำงานไปยังเครื่องรับเหรียญ
    ui->textEdit_2->setText("0");
    ui->textEdit->insertPlainText("-----\r");
    ui->textEdit->insertPlainText("THANK YOU\r");
    ui->textEdit->insertPlainText("Your order number : ");
    ui->textEdit->insertPlainText(QString::number(q));
    ui->textEdit->insertPlainText("\r");
    QDateTime now(QDateTime::currentDateTime()); // สร้างวันที่และเวลา
    QString datetime = now.toString("yyyy-MM-dd H:m:s"); // ตัวแปรสำหรับเก็บค่า
วันที่และเวลา
    ui->textEdit->insertPlainText("order time : ");
    ui->textEdit->insertPlainText(datetime);
    ui->textEdit->insertPlainText("\r");
// create and write files //
    QString filename="/home/pi/log/log_customer.csv"; // กำหนดที่อยู่ไฟล์ csv
    QFile file(filename);
    file.open(QIODevice::Append | QIODevice::Text); // เปิดอ่านไฟล์ที่กำหนดไว้
    QTextStream stream(&file);
// ทำการบันทึกวันที่ เวลา และข้อมูลการสั่งอาหารแต่ละครั้งลงในไฟล์ที่กำหนดไว้
    stream << "\n";
    stream << q;
    stream << ";";

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

stream << datetime;
stream << ",";
if(amount1 != 0 && menu_1 != ""){
    const char* order1 = "Gyokaikei Ramen ";
    stream << order1;
    stream << ",";
    stream << amount1;
    stream << ",";
}
if(amount2 != 0 && menu_2 != ""){
    const char* order2 = "Katsudon Ramen ";
    stream << order2;
    stream << ",";
    stream << amount2;
    stream << ",";
}
if(amount3 != 0 && menu_3 != ""){
    const char* order3 = "Miso Ramen ";
    stream << order3;
    stream << ",";
    stream << amount3;
    stream << ",";
}
if(amount4 != 0 && menu_4 != ""){
    const char* order4 = "Shoyu Ramen ";
    stream << order4;
    stream << ",";
    stream << amount4;
    stream << ",";
}

if(amount5 != 0 && menu_5 != ""){
    const char* order5 = "Tonkotsu Ramen ";
    stream << order5;
    stream << ",";

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

stream << amount5;
stream << ",";
}
if(amount6 != 0 && menu_6 != ""){
    const char* order6 = "Tsuke Men ";
    stream << order6;
    stream << ",";
    stream << amount6;
    stream << ",";
}
stream << total_prices;
stream << ",";
stream << coin_10;
stream << ",";
stream << coin_5;
stream << ",";
stream << coin_2;
stream << ",";
stream << coin_1;
file.close();

QString filelog="/home/pi/log/amount_log.csv";
QFile log(filelog);
log.open(QIODevice::Append | QIODevice::Text);
QTextStream stream2(&log);
    stream2 << q;
    stream2 << ",";
    stream2 << datetime;
    stream2 << ",";
    stream2 << amount1;
    stream2 << ",";
    stream2 << amount2;
    stream2 << ",";
    stream2 << amount3;
    stream2 << ",";
    stream2 << amount4;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

stream2 << ",";
stream2 << amount5;
stream2 << ",";
stream2 << amount6;
stream2 << "\n";
log.close();

```

```

QNetworkRequest request(QUrl("http://localhost/insert_log.php"));
insert = insert_log.get(request); // เรียกใช้งานไฟล์ PHP เพื่อเก็บข้อมูลลงใน

```

ฐานข้อมูล

```

QNetworkRequest request2(QUrl("http://localhost/insert_amount_log.php"));
insert2 = insert_log2.get(request2);

// set variable to initial //
coin_10=0;coin_5=0;coin_2=0;coin_1=0;sum=0;count = 0;total_prices=0;
menu_1 = "";menu_2 = "";menu_3 = "";menu_4 = "";menu_5 = "";menu_6 = "";
prices_1 = 0;prices_2 = 0;prices_3 = 0;prices_4 = 0;prices_5 = 0;prices_6 = 0;
ui->spinBox->setValue(0); // กำหนดค่าใน spinBox ให้เท่ากับศูนย์
ui->spinBox_2->setValue(0);
ui->spinBox_3->setValue(0);
ui->spinBox_4->setValue(0);
ui->spinBox_5->setValue(0);
ui->spinBox_6->setValue(0);
QTimer::singleShot(10000, this, SLOT(initial())); // เมื่อครบเวลา 10 วินาทีให้เรียกใช้

```

ฟังก์ชัน initial

```

}else{
    ui->textEdit_2->setText(total);
}

```

```

// return variable //
return total_prices;
return amount1;
return amount2;
return amount3;
return amount4;
return amount5;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return amount6;
return menu_1.toInt();
return menu_2.toInt();
return menu_3.toInt();
return menu_4.toInt();
return menu_5.toInt();
return menu_6.toInt();
return prices_1.toInt();
return prices_2.toInt();
return prices_3.toInt();
return prices_4.toInt();
return prices_5.toInt();
return prices_6.toInt();
}

```

```

QByteArray const MainWindow::on_pushButton_4_clicked() // ปุ่มสำหรับสั่งเมนูที่ 1
{
    // Gyokaikei Ramen //
    Order = 0; // กำหนดให้ Order = 0
    menu_1.fill(0, 80);
    QString menu = food[Order].getName(); // สร้างตัวแปรเก็บชื่อเมนูตามตำแหน่ง Order ที่
กำหนด
    menu_1.insert(0, menu.toLocal8Bit()); // แปลงตัวแปรให้อยู่ในรูปแบบ Array
    menu_1.resize(80);
    prices_1.fill(0, 80);
    QString data = QString::number(food[Order].getPrice()); // สร้างตัวแปรเก็บราคาตาม
ตำแหน่ง Order ที่กำหนด
    prices_1.insert(0, data.toLocal8Bit());
    prices_1.resize(80);
    return menu_1;
    return prices_1;
}

```

```

QByteArray const MainWindow::on_pushButton_5_clicked() // ปุ่มสำหรับสั่งเมนูที่ 2
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Katsudon Ramen //
Order = 1;
menu_2.fill(0, 80);
QString menu = food[Order].getName();
menu_2.insert(0, menu.toLocal8Bit());
menu_2.resize(80);
prices_2.fill(0, 80);
QString data = QString::number(food[Order].getPrice());
prices_2.insert(0, data.toLocal8Bit());
prices_2.resize(80);
return menu_2;
return prices_2;
}

QByteArray const MainWindow::on_pushButton_6_clicked() // ปุ่มสำหรับสั่งเมนูที่ 3
{
    // Miso Ramen //
    Order = 2;
    menu_3.fill(0, 80);
    QString menu = food[Order].getName();
    menu_3.insert(0, menu.toLocal8Bit());
    menu_3.resize(80);
    prices_3.fill(0, 80);
    QString data = QString::number(food[Order].getPrice());
    prices_3.insert(0, data.toLocal8Bit());
    prices_3.resize(80);
    return menu_3;
    return prices_3;
}

QByteArray const MainWindow::on_pushButton_7_clicked() // ปุ่มสำหรับสั่งเมนูที่ 4
{
    // Shoyu Ramen //
    Order = 3;
    menu_4.fill(0, 80);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

QString menu = food[Order].getName();
menu_4.insert(0, menu.toLocal8Bit());
menu_4.resize(80);
prices_4.fill(0, 80);
QString data = QString::number(food[Order].getPrice());
prices_4.insert(0, data.toLocal8Bit());
prices_4.resize(80);
return menu_4;
return prices_4;
}

QByteArray const MainWindow::on_pushButton_8_clicked() // ปุ่มสำหรับสั่งเมนูที่ 5
{
    // Tonkotsu Ramen //
    Order = 4;
    menu_5.fill(0, 80);
    QString menu = food[Order].getName();
    menu_5.insert(0, menu.toLocal8Bit());
    menu_5.resize(80);
    prices_5.fill(0, 80);
    QString data = QString::number(food[Order].getPrice());
    prices_5.insert(0, data.toLocal8Bit());
    prices_5.resize(80);
    return menu_5;
    return prices_5;
}

QByteArray const MainWindow::on_pushButton_9_clicked() // ปุ่มสำหรับสั่งเมนูที่ 6
{
    // Tsuke Men //
    Order = 5;
    menu_6.fill(0, 80);
    QString menu = food[Order].getName();
    menu_6.insert(0, menu.toLocal8Bit());
    menu_6.resize(80);
    prices_6.fill(0, 80);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

QString data = QString::number(food[Order].getPrice());
prices_6.insert(0, data.toLocal8Bit());
prices_6.resize(80);
return menu_6;
return prices_6;
}

```

```

int MainWindow::on_pushButton_3_clicked() // ปุ่มยืนยันสำหรับสั่งอาหาร
{
    QString amount_1 = ui->spinBox->text(); // สร้างตัวแปรสำหรับเก็บค่าใน spinbox
    QString amount_2 = ui->spinBox_2->text();
    QString amount_3 = ui->spinBox_3->text();
    QString amount_4 = ui->spinBox_4->text();
    QString amount_5 = ui->spinBox_5->text();
    QString amount_6 = ui->spinBox_6->text();
    amount1 = amount_1.toInt(); // แปลงตัวแปรที่ได้จาก spinbox ให้อยู่ในรูปแบบตัวเลข
    amount2 = amount_2.toInt();
    amount3 = amount_3.toInt();
    amount4 = amount_4.toInt();
    amount5 = amount_5.toInt();
    amount6 = amount_6.toInt();
    if(amount1 == 0 && amount2 == 0 && amount3 == 0 && amount4 == 0 &&
amount5 == 0 && amount6 == 0) // กำหนดเงื่อนไขหากไม่มีการเลือกเมนูอาหาร
    {
        ui->textEdit->insertPlainText("Please select menu\r");
        return 0;
    }else{
        serial.write(HexEn); // สั่งคำสั่งเริ่มทำงานไปยังเครื่องรับเหรียญ
        p_1 = prices_1.toInt()*amount1;
        p_2 = prices_2.toInt()*amount2;
        p_3 = prices_3.toInt()*amount3;
        p_4 = prices_4.toInt()*amount4;
        p_5 = prices_5.toInt()*amount5;
        p_6 = prices_6.toInt()*amount6;
        total_prices = p_1+p_2+p_3+p_4+p_5+p_6;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ui->textEdit_2->setText(QString::number(total_prices));
if(total_prices != 0){
    count = 1;
    ui->textEdit->clear();
    if(menu_1 != "" && amount1 != 0){
        ui->textEdit->insertPlainText(menu_1);
        ui->textEdit->insertPlainText(QString::number(amount1));
        ui->textEdit->insertPlainText(enter);
    }
    if(menu_2 != "" && amount2 != 0){
        ui->textEdit->insertPlainText(menu_2);
        ui->textEdit->insertPlainText(QString::number(amount2));
        ui->textEdit->insertPlainText(enter);
    }
    if(menu_3 != "" && amount3 != 0){
        ui->textEdit->insertPlainText(menu_3);
        ui->textEdit->insertPlainText(QString::number(amount3));
        ui->textEdit->insertPlainText(enter);
    }
    if(menu_4 != "" && amount4 != 0){
        ui->textEdit->insertPlainText(menu_4);
        ui->textEdit->insertPlainText(QString::number(amount4));
        ui->textEdit->insertPlainText(enter);
    }
    if(menu_5 != "" && amount5 != 0){
        ui->textEdit->insertPlainText(menu_5);
        ui->textEdit->insertPlainText(QString::number(amount5));
        ui->textEdit->insertPlainText(enter);
    }
    if(menu_6 != "" && amount6 != 0){
        ui->textEdit->insertPlainText(menu_6);
        ui->textEdit->insertPlainText(QString::number(amount6));
        ui->textEdit->insertPlainText(enter);
    }
    ui->textEdit->insertPlainText("Please insert coin\r");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
return amount1;
return amount2;
return amount3;
return amount4;
return amount5;
return amount6;
return total_prices;
}

int MainWindow::on_cancel_clicked() // ปุ่ม Cancel สำหรับยกเลิกการสั่งอาหาร
{
    initial(); // เรียกใช้งานฟังก์ชัน initial
    serial.write(HexDis);
    coin_10=0;coin_5=0;coin_2=0;coin_1=0;sum=0;count = 0;total_prices=0;
    menu_1 = "";menu_2 = "";menu_3 = "";menu_4 = "";menu_5 = "";menu_6 = "";
    prices_1 = 0;prices_2 = 0;prices_3 = 0;prices_4 = 0;prices_5 = 0;prices_6 = 0;
    ui->spinBox->setValue(0);
    ui->spinBox_2->setValue(0);
    ui->spinBox_3->setValue(0);
    ui->spinBox_4->setValue(0);
    ui->spinBox_5->setValue(0);
    ui->spinBox_6->setValue(0);
    return total_prices;
    return amount1;
    return amount2;
    return amount3;
    return amount4;
    return amount5;
    return amount6;
    return menu_1.toInt();
    return menu_2.toInt();
    return menu_3.toInt();
    return menu_4.toInt();
    return menu_5.toInt();
    return menu_6.toInt();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
return prices_1.toInt();  
return prices_2.toInt();  
return prices_3.toInt();  
return prices_4.toInt();  
return prices_5.toInt();  
return prices_6.toInt();  
}
```

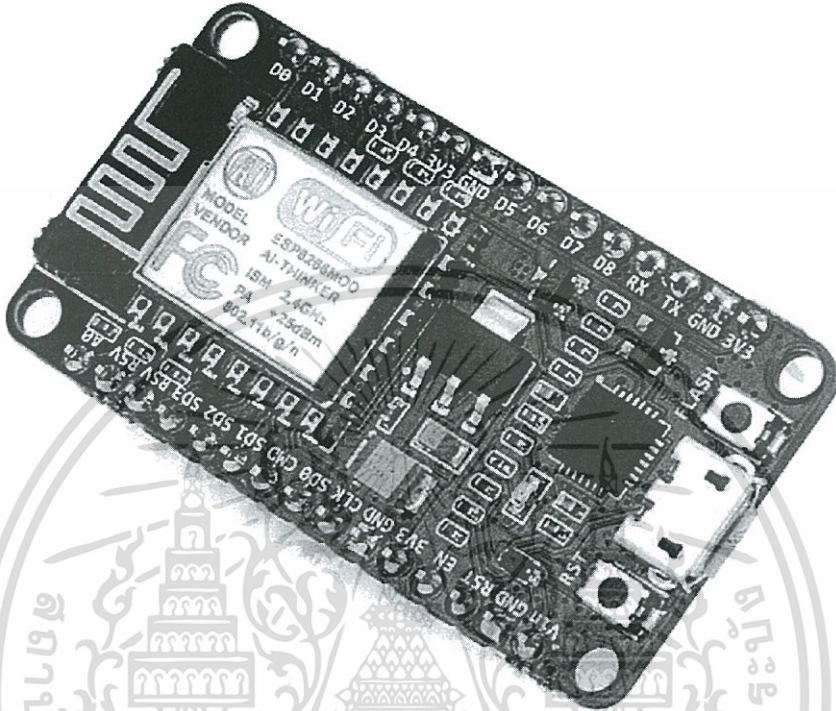


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NodeMCU v2



The NodeMcu is an open-source firmware and development kit that helps you to Prototype your IOT product within a few Lua script lines.

Features:

- Open-source
- Interactive
- Programmable
- Low cost
- Simple
- Smart
- WI-FI enabled

Arduino-like hardware IO

Advanced API for hardware IO, which can dramatically reduce the redundant work for configuring and manipulating hardware. Code like arduino, but interactively in Lua script.

Nodejs style network API

Event-driven API for network applicaitons, which faciliates developers writing code running on a 5mm*5mm sized MCU in Nodejs style.

Greatly speed up your IOT application developing process.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Specification:

The Development Kit based on ESP8266, integrates GPIO, PWM, IIC, 1-Wire and ADC all in one board.

Power your development in the fastest way combining with NodeMCU Firmware!

- USB-TTL included, plug&play
- 10 GPIO, every GPIO can be PWM, I2C, 1-wire
- FCC CERTIFIED WI-FI module (Coming soon)
- PCB antenna

Document

[Schematic&PCB, Source Code, API Documents](#)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Table of Contents

1.	RS-232 PROTOCOL	2
2.	APPENDIX A: REVISION HISTORY	6



RS-232 PROTOCOL**1. Transmission Specification**

Transmission Method Full Duplex Transmission
 Transmission Speed 9600 Baud Rate
 Synchronizing Method Asynchronous Method

Data Format

Start bit 1
 Data bit 8
 Parity bit none
 Stop bit 1

2. General Message Command Format

SYNC + LNG + CMD + DATA + EXT + CHECKSUM

Name	Byte	Contents
SYNC	1	Message transmission start code [90h] fixed
LNG	1	Data Length (Total number of bytes from SYNC to checksum)
CMD	1	Command status
DATA	0~250	Data necessary for command(omitted by CMD)
EXT	1	03h End of Message byte
CHECKSUM	1	check code by SUM result method without carry (from SYNC to EXT)

There are 2 types of messages used

(Format-1) without DATA : SYNC + LNG + CMD + EXT + CHECKSUM

(Format-2) with DATA : SYNC + LNG + CMD + DATA + EXT + CHECKSUM

Ex1. ACK

SYNC	LNG	CMD	EXT	CHECKSUM
90h	05h	50h	03h	E8h

Ex2. ACCEPT Coin of Channel.1

SYNC	LNG	CMD	DATA	EXT	CHECKSUM
90h	06h	12h	01h*	03h	ACh

* number of channel



3. Details of each transmission, reception command.

	CMD	Format	Contents
Controller	0x01	1	Enable UCA
	0x02	1	Disable UCA
	0x03	1	Get Version (*1)
	0x11	1	Request UCA current status
UCA	0x50	1	ACK
	0x4B	1	NAK
	0x03	2	Firmware Version (*1)
	0x11	1	Idling
	0x12	2	Accepting
	0x13	1	Reserve
	0x14	1	Disable
	0x15	1	Reserve
	0x16	2	Sensor Problem
	0x17	1	Fishing
	0x18	1	Program Checksum Error
	0x19	1	Reserve

Note: (*1) New command for protocol v1.1 or higher.



4. Command Message & Responses

4.1 Enable or Disable UCA

COMMAND

90h	05h	01h	03h	99h	Enable UCA
-----	-----	------------	-----	-----	------------

90h	05h	02h	03h	9Ah	Disable UCA
-----	-----	------------	-----	-----	-------------

ACCEPTED RESPONSE

90h	05h	50h	03h	E8h	ACK
-----	-----	------------	-----	-----	-----

REJECTED RESPONSE

90h	05h	4Bh	03h	E3h	NAK
-----	-----	------------	-----	-----	-----

4.2 Request UCA status

COMMAND

90h	05h	11h	03h	A9h	Request status
-----	-----	------------	-----	-----	----------------

ACCEPTED RESPONSE

90h	05h	11h	03h	A9h	Idling
-----	-----	------------	-----	-----	--------

90h	05h	13h	03h	ABh	Ready
-----	-----	------------	-----	-----	-------

90h	05h	14h	03h	ACH	Disable
-----	-----	------------	-----	-----	---------

90h	05h	15h	03h	ADh	Reserved
-----	-----	------------	-----	-----	----------

90h	06h	16h	01h	03h	B0h	Sensor 1 Problem (COIL 1)
-----	-----	------------	-----	-----	-----	---------------------------

90h	06h	16h	02h	03h	B1h	Sensor 2 Problem (COIL 2)
-----	-----	------------	-----	-----	-----	---------------------------

90h	06h	16h	03h	03h	B2h	Sensor 3 Problem (DROP IN 2)
-----	-----	------------	-----	-----	-----	------------------------------

90h	06h	16h	04h	03h	B3h	Sensor 4 Problem (DROP IN 1)
-----	-----	------------	-----	-----	-----	------------------------------

90h	06h	16h	05h	03h	B4h	Sensor 5 Problem (COIN RETURN)
-----	-----	------------	-----	-----	-----	--------------------------------



INTERNATIONAL CURRENCY TECHNOLOGIES CORPORATION

90h	06h	16h	06h	03h	B5h
-----	-----	-----	-----	-----	-----

Sensor 6 Problem (IR sensor)

90h	05h	17h	03h	AFh
-----	-----	-----	-----	-----

Fishing

90h	05h	18h	03h	B0h
-----	-----	-----	-----	-----

Program Checksum Error

90h	05h	19h	03h	EFh
-----	-----	-----	-----	-----

Reserved

REJECTED RESPONSE

90h	05h	4Bh	03h	E3h
-----	-----	-----	-----	-----

NAK

4.3 Accepting (Ex. Channel-1= \$1, Channel-2= \$5, Channel-3= \$10)

90h	06h	12h	01h	03h	ACh
-----	-----	-----	-----	-----	-----

Accepting \$1

90h	06h	12h	02h	03h	ADh
-----	-----	-----	-----	-----	-----

Accepting \$5

90h	06h	12h	03h	03h	AEnh
-----	-----	-----	-----	-----	------

Accepting \$10

Note: The UCA will automatic transmit the accepting message to controller.

4.4 Get UCA Version (Ex. UCA1#S100CI0010WD### and Checksum=3F68h)

COMMAND

90h	05h	03h	03h	9Bh
-----	-----	-----	-----	-----

ACCEPTED RESPONSE

90h	1Bh	03h	55h	43h	41h	31h	23h	53h	31h	30h	30h
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

'U' 'C' 'A' '1' '#' 'S' '1' '0' '0'

43h	49h	30h	30h	31h	30h	57h	44h	23h	23h	23h	3Fh
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

'C' 'I' '0' '0' '1' '0' 'W' 'D' '#' '#' '#' HI

68h	03h	9Ch
-----	-----	-----

LOW

REJECTED RESPONSE

90h	05h	4Bh	03h	E3h
-----	-----	-----	-----	-----



APPENDIX A: REVISION HISTORY

Date	Revision	Changes
26-FEB-2008	1.0	Initial release.
04-MAR-2009	1.1	New command for read the firmware version.
25-MAR-2009	1.2	Modify wrong description of sensor problem response and reserve some response status.

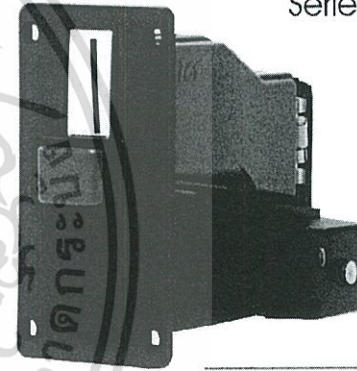


ict

Coin Acceptor

UCA

Series



Installation Guide

Contents

Use of Materials Limitations

International Currency Technologies Corporation (ICT) all rights reserved.

All materials contained are the copyrighted property of ICT.

All trademarks, service marks, and trade names are proprietary to ICT.

ICT reserves the right at all times to disclose or to modify any information as ICT deems necessary to satisfy any applicable law, regulation, legal process or governmental request, or to edit, refuse to post or to remove any information or materials, in whole or in part, in ICT's sole discretion.

1. Introduction	
1-1. Overview.....	2
1-2. Features.....	2
2. Specifications	2
3. Packing List	3
4. Dimension	4
5. Installation	
5-1. How to Install?.....	5
5-2. Harness Application.....	7
5-3. I/O Circuits.....	12
5-4. DIP Switch Setting.....	14
5-5. Software Download and Upgrade.....	14
6. Maintenance	15
7. Troubleshooting	17

1. Introduction

1-1. Overview

UCA series is designed as reliable coin acceptors which are able to work steadily in high temperature for high-security with acceptance rate up to 96% or greater.

Moreover, UCA Series includes UCA1, UCA2, UCA3, UCAE and UCAES for different accepted coins and applications to fulfill your needs.

1-2. Features

- Low power consumption.
- Eight coin channels available at once.
- Mechanical and electrical anti-string functions.
- Accepting speed up to 3 coins per second.
- Coin sorter option available for UCA1, UCA2 and UCAES.

2. Specifications

General

Acceptance Rate	96 % or greater
Accepting Speed	Approx. 3 coin/sec
Interface	Pulse, RS232 (TTL level)
Coin Parameters	Diameter : 20mm~32mm Thickness: 1.2mm~3.2mm

Electrical

Power Source 12V ±10% (10.8V ~ 13.2V DC)

Power Consumption
Standby : 0.05A, 0.6W
Operation: 0.2 A, 2.4W
Maximum: 0.5 A, 6 W

Operation Environment
Operation Temperature: -5°C~60°C
Storage Temperature : -20°C~75°C
Humidity: 30%~85% RH
(no condensation)

Mechanical

Outline Dimension Refer to page. 4

Net Weight Approx. 0.35 kg

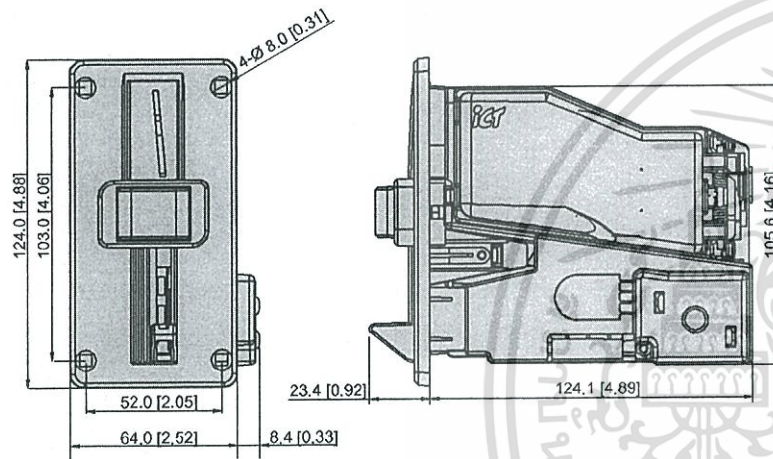
 **Installation: Indoor use only!!**

3. Packing List

Main	UCA Series Coin Acceptor	
Accessory	Harnesses: Refer to 5-2 UCA Series Installation Guide UCA Series Switches Setting Guide	

Mode Type	Harness	Others
Pulse	Refer to 5-2	Screw pack
RS232	Refer to 5-2	Screw pack

4. Dimension



Unit : mm [inch]

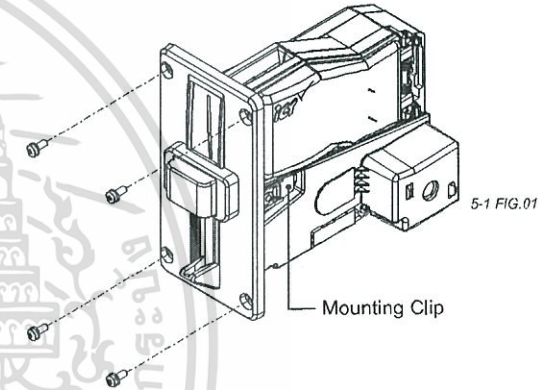
4 FIG.01

5. Installation

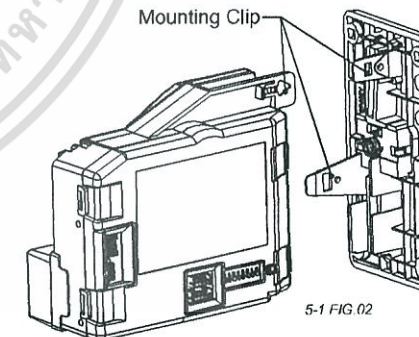
5-1. How to Install?

To install UCA Series coin acceptor on your machine, please follow the steps as below:

1. Use four screws to fix bezel on the machine.



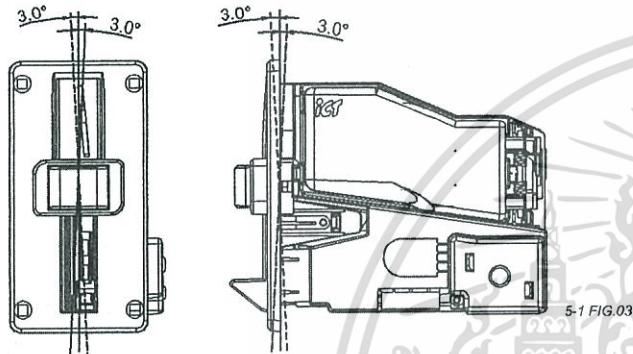
2. Fix main base on bezel by mounting clips.



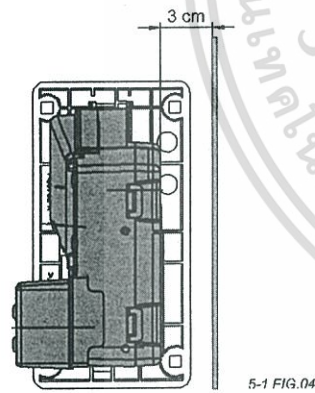
5-2. Harness Application

Connector:

▪ UCA1/ UCA2/ UCAES

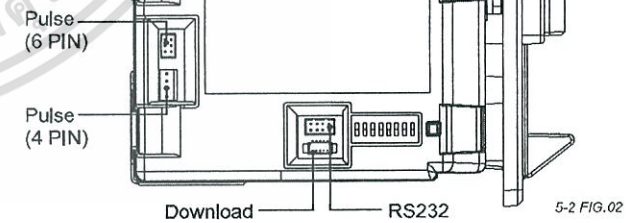
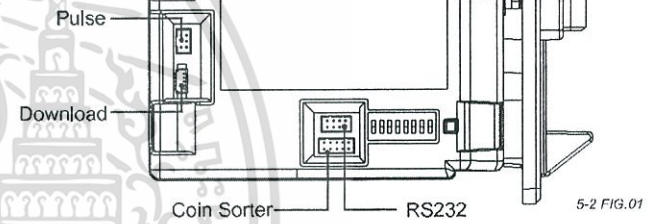


! UCA Series is not supposed to be aslope over 3°.



! To make sure UCA Series work smoothly, install machine farther than 3 cm from metal items is recommended.

▪ UCA3/ UCAE

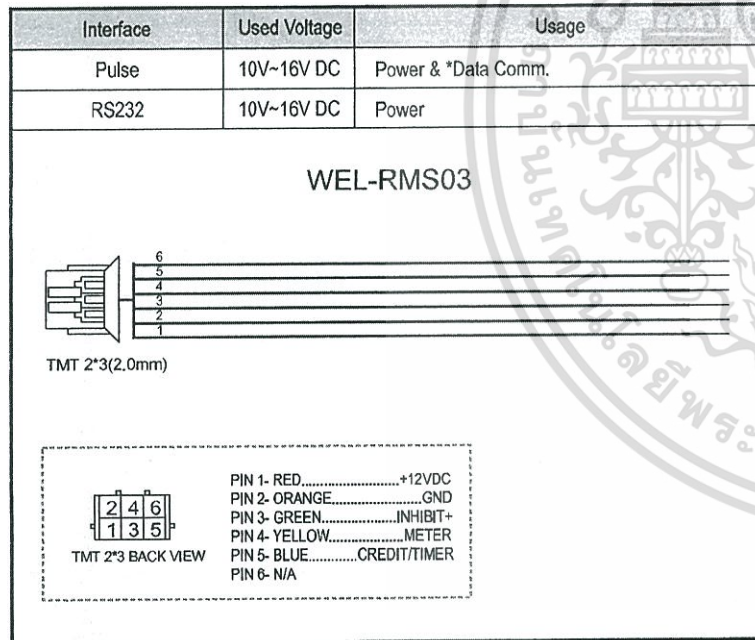


5-2 TABLE 01

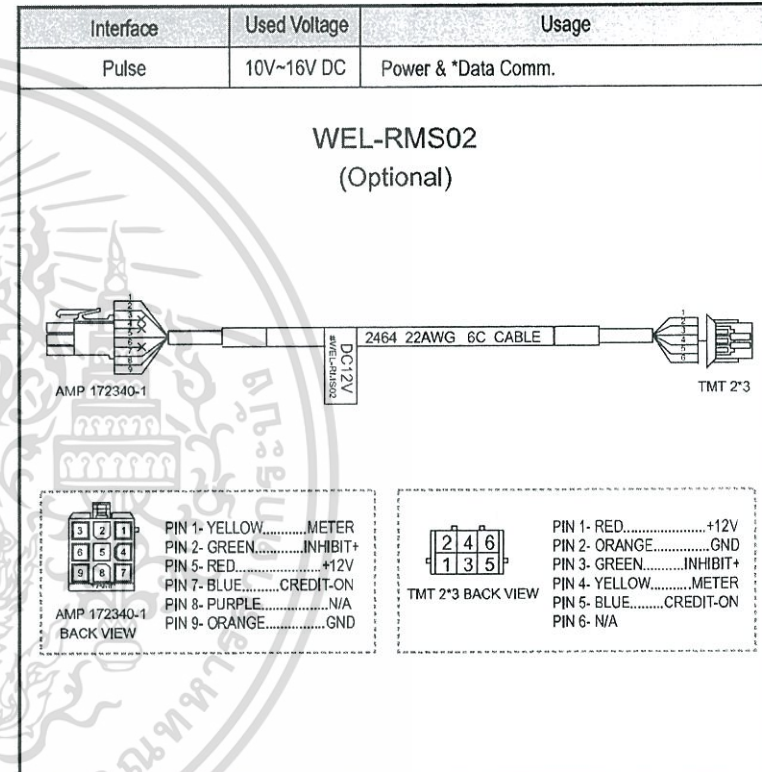
Interface	Used Voltage	Usage	Harness	Page
Pulse	10V~16V DC	Power & *Data Comm.	WEL-RMS03	9
		Power & *Data Comm.	WEL-RMS02(Optional)	10
		Extension Wire	CU-R961-1(Optional)	11
RS232	10V~16V DC	Power	WEL-RMS03	9
	+5V TTL	*Data Comm.	WEL-R7U06-2	12

*Data Comm.: Data Communication.

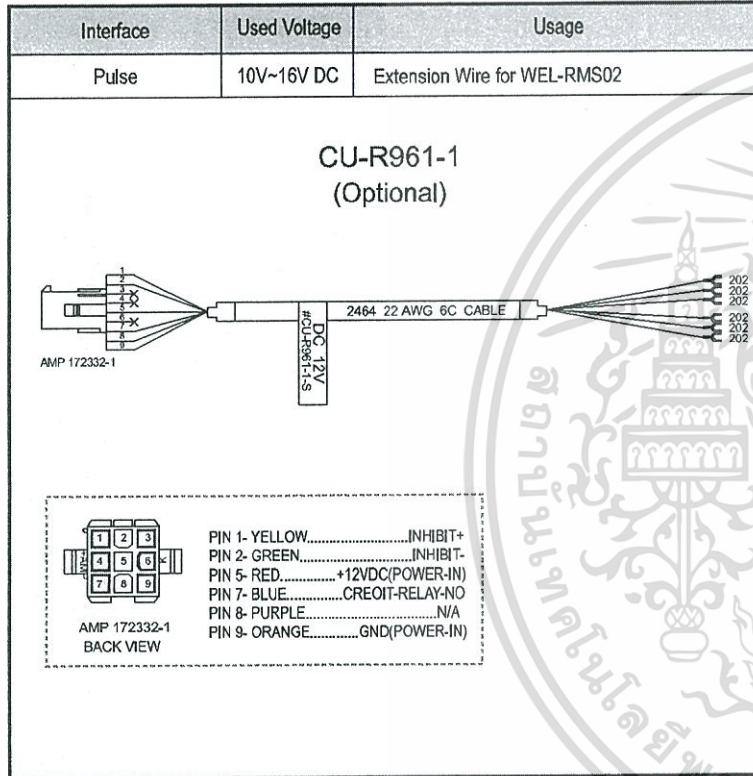
5-2 FIG.03



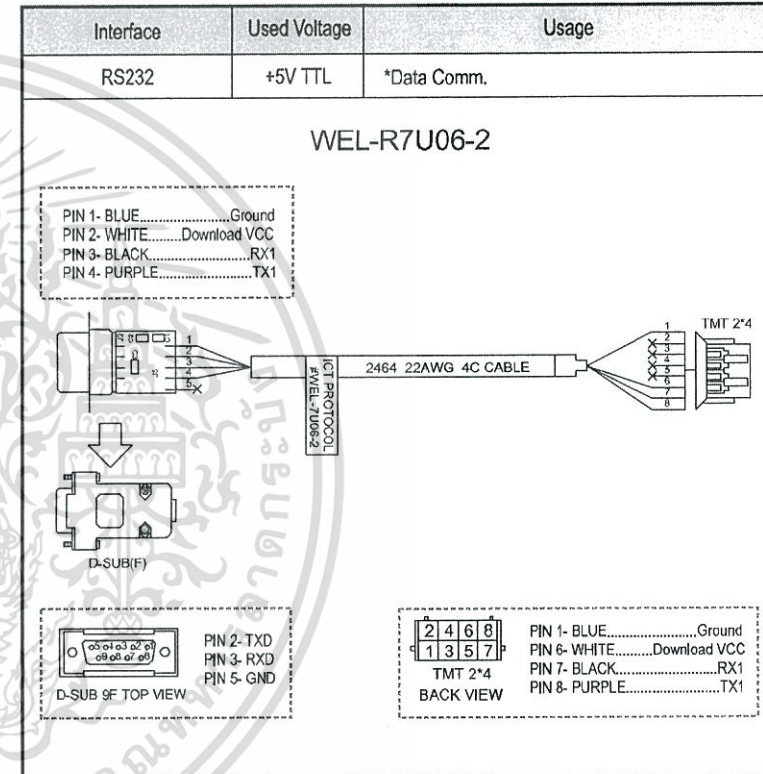
5-2 FIG.04



5-2 FIG.05



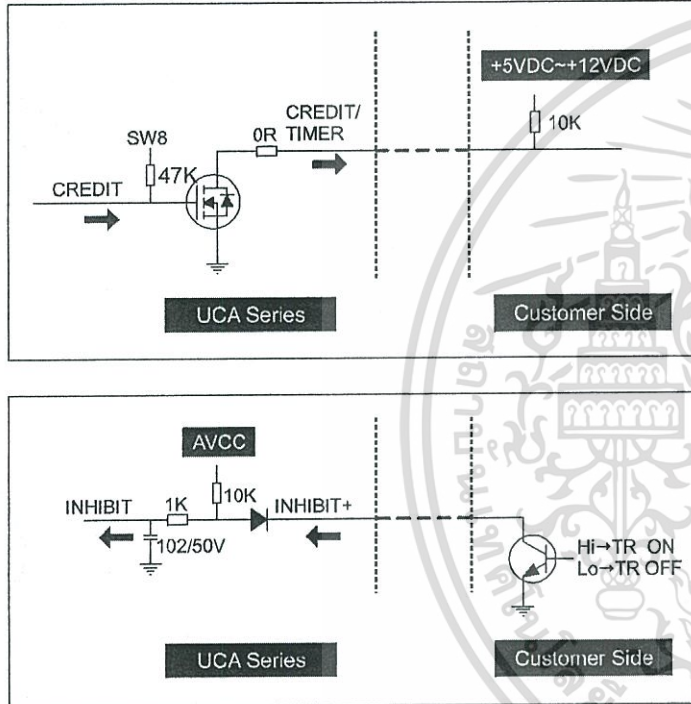
5-2 FIG.06



5-3. I/O Circuits

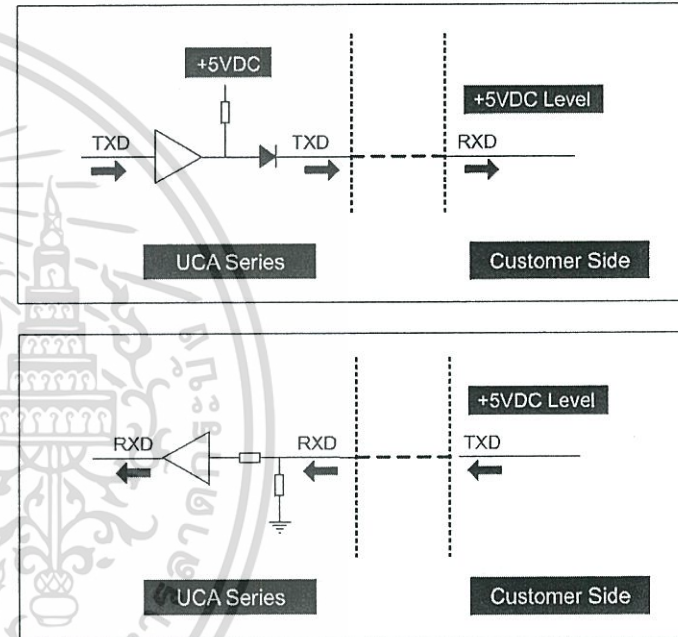
Pulse Interface.

5-3 FIG.01



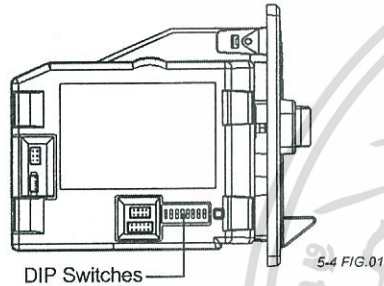
RS232 Interface.

5-3 FIG.02



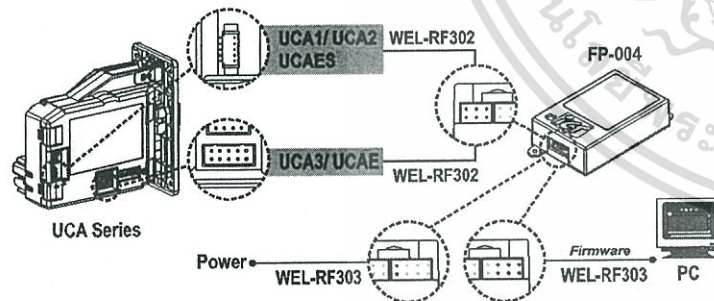
5-4. DIP Switch Setting

The DIP switches are located on side of UCA Series. DIP switch setting varies according to different functions which are used by users. For DIP switch setting which fits your needs, please refer to "UCA Series Switch Setting" guide in the package.



5-5. Software Download and Upgrade

To download and upgrade the software to UCA Series, the programmer (FP-004) is needed. Please contact ICT to purchase FP-004 and refer to FP-004 user guide for software download and upgrade information.



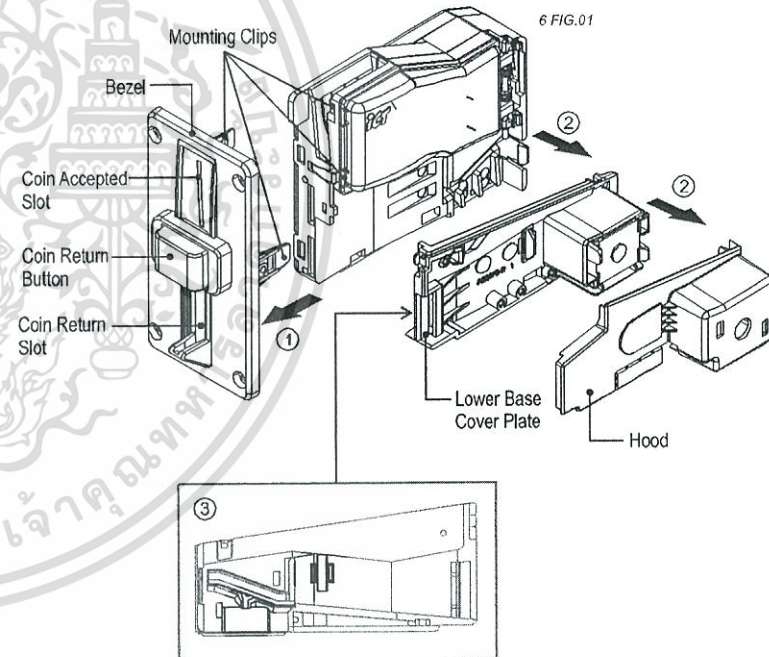
Power must be applied to UCA Series after connecting.

6. Maintenance

To make sure UCA Series coin acceptor always works smoothly, please clean the internal sensors regularly.

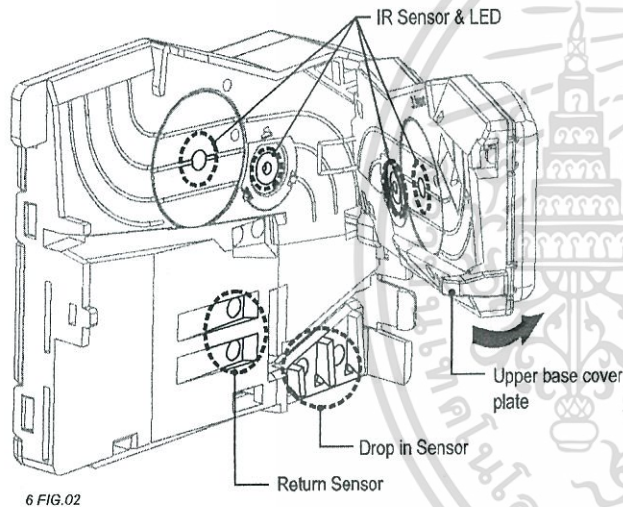
To clean the internal sensors:

1. Remove bezel by releasing mounting clips and pulling it out.
2. Release the clip and remove lower base cover plate and hood.
3. Use a soft, dry cloth, or towel to clean lens.

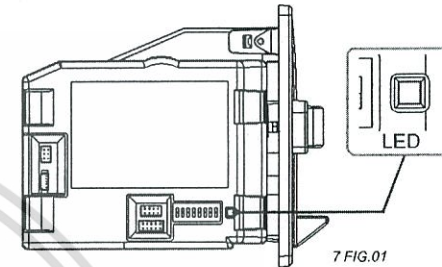


7. Troubleshooting

4. Open upper base cover plate and then use a soft, dry cloth, or towel to clean sensors.



6 FIG.02



7 FIG.01

7 TABLE 01-1

LED Flashes			Status	Corrective Actions
Green	Orange	Red		
N/A	N/A	1	First coil set error	Call ICT or agents for technical support.
N/A	N/A	2	Second coil set error	
N/A	N/A	3	Fish sensor error	Inspect for foreign objects in coin path and clean.
N/A	N/A	4	Drop sensor error	
N/A	N/A	5	Return sensor error	
N/A	N/A	6	IR sensor error	
N/A	N/A	7	Program error	Call ICT or agents for technical support.
Fast 1	N/A	Fast 1	A Stringing attempt has been detected.	Inspect for foreign objects in coin path and clean.

7 TABLE 01-2

LED Flashes			Status
Green	Orange	Red	
ON	N/A	N/A	Power ON
N/A	ON	N/A	INHIBIT
N/A	Flashes	N/A	I/O Test Mode

Maintenance Notice
(Any improper maintenance will result invalid warranty.)

Recommended	Mild, non-abrasive, soap water.
DO NOT USE	Organic solvent, Alcohol, Volatility liquid.



If the error can not be solved after corrective actions or happen again, please contact ICT or agents for technical support.



ict Taiwan

International Currency Technologies Corporation

No.28, Ln. 15, Sec. 6, Minquan E. Rd., Neihu Dist., Taipei City 114, Taiwan (R.O.C.)

sales@ictgroup.com.tw (For Sales)

rma@ictgroup.com.tw (For Customer Service)

Website: www.ictgroup.com.tw