# ระบบควบคุมเครื่องซักผ้าหลายเครื่องในธุรกิจซักผ้าแบบบริการตัวเอง

# SYSTEM FOR CONTROLLING MULTIPLE WASHING MACHINES FOR SELF SERVICE LAUNDRY BUSINESS


โดย

นางสาวนนทพร ทองอยู่

นางสาวบัณฑิตา ใจใส

นายภาณุพงศ์ น้ำแก้ว


ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

# ระบบควบคุมเครื่องซักผ้าหลายเครื่องในธุรกิจซักผ้าแบบบริการตัวเอง

# SYSTEM FOR CONTROLLING MULTIPLE WASHING MACHINES FOR SELF SERVICE LAUNDRY BUSINESS


โดย

นางสาวนนทพร ทองอยู่

นางสาวบัณฑิตา ใจใส

นายภาณุพงศ์ น้ำแก้ว


ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

# ระบบควบคุมเครื่องซักผ้าหลายเครื่องในธุรกิจซักผ้าแบบบริการตัวเอง
# SYSTEM FOR CONTROLLING MULTIPLE WASHING MACHINES FOR SELF SERVICE LAUNDRY BUSINESS


โดย

| | | |
|---|---|---|
| นางสาวนนทพร | ทองอยู่ | 53010782 |
| นางสาวบัณฑิตา | ใจใส | 53010879 |
| นายภาณุพงศ์ | น้ำแก้ว | 53011234 |

อาจารย์ที่ปรึกษา
รศ.ดร. จีรสุดา โกษียาภรณ์
รศ.ดร. ปราโมทย์ วาดเขียน


ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

ผ่านการตรวจรูปเล่มแล้ว
(.....)
อาจารย์ที่ปรึกษา
11 / มี.ค. / 57
KMITL วิศวกรรมโทรคมนาคม Telecommunications Engineering

ผ่านการตรวจชิ้นงานแล้ว
(.....)
กรรมการผู้ตรวจชิ้นงาน
17 / 3 / 57
KMITL วิศวกรรมโทรคมนาคม Telecommunications Engineering

ปริญญานิพนธ์ปีการศึกษา 2556
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง ระบบควบคุมเครื่องซักผ้าหลายเครื่องในธุรกิจซักผ้าแบบบริการตัวเอง
SYSTEM FOR CONTROLLING MULTIPLE WASHING MACHINES
FOR SELF SERVICE LAUNDRY BUSINESS

ผู้จัดทำ

| | | | |
|---|---|---|---|
| 1. | นางสาวนนทพร | ทองอยู่ | 53010782 |
| 2. | นางสาวบัณฑิตา | ใจใส | 53010879 |
| 3. | นายภาณุพงศ์ | น้ำแก้ว | 53011234 |

.....  อาจารย์ที่ปรึกษา
รศ.ดร. จีรสุดา โกษียาภรณ์

.....  อาจารย์ที่ปรึกษาร่วม
รศ.ดร. ปราโมทย์ วาดเขียน

# กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความอนุเคราะห์อย่างดียิ่งจากอาจารย์ที่ปรึกษา คือ รศ.ดร.จีรสุดา โกษียาภรณ์ และ รศ.ดร.ปราโมทย์ วาดเขียน ที่ให้คำแนะนำ คำสั่งสอน ให้ความรู้ ความเข้าใจตลอดระยะเวลาในการทำปริญญานิพนธ์นี้ ขอขอบพระคุณท่านในความห่วงใยและความหวังดีที่ให้แก่ผู้จัดทำเป็นอย่างยิ่ง

ขอบพระคุณคณาจารย์ ประจำสาขาวิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกท่าน ที่ได้อบรมสั่งสอนและประสิทธิ์ประสาทวิชาความรู้ให้แก่ผู้จัดทำ

ขอขอบคุณคุณผู้มีส่วนเกี่ยวข้องทุกท่านที่คอยช่วยเหลือ ให้กำลังใจแก่ผู้จัดทำเสมอมา จนกระทั่งปริญญานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี ความดีอันเกิดจากการทำปริญญานิพนธ์นี้ ผู้จัดทำขอมอบแด่บิดา มารดา ครู อาจารย์ เพื่อนนักศึกษาและผู้มีพระคุณทุกท่าน

นางสาวนนทพร ทองอยู่
นางสาวบัณฑิตา ใจใส
นายภาณุพงศ์ น้ำแก้ว

ผู้จัดทำ

**ระบบควบคุมเครื่องซักผ้าหลายเครื่องในธุรกิจซักผ้าแบบบริการตัวเอง**

SYSTEM FOR CONTROLLING MULTIPLE WASHING MACHINES FOR SELF SERVICE LAUNDRY BUSINESS

**โดย** นางสาวนนทพร ทองอยู่ 53010782
นางสาวบัณฑิตา ใจใส 53010879
นายภาณุพงศ์ น้ำแก้ว 53011234

**อาจารย์ที่ปรึกษา** รศ.ดร. จีรสุดา โกษียาภรณ์
รศ.ดร. ปราโมทย์ วาดเขียน

**บทคัดย่อ**

ปริญญานิพนธ์นี้เป็นการออกแบบและสร้างระบบให้บริการเครื่องซักผ้าผ่านระบบเครือข่ายอินเตอร์เน็ต โดยใช้เครื่องหยอดเหรียญเพียงหนึ่งเครื่อง ที่มีความสามารถในการรองรับเครื่องซักผ้าได้ 4 เครื่อง เพื่อเป็นการประหยัดต้นทุนของเครื่องหยอดเหรียญที่ติดตั้งกับเครื่องซักผ้าแต่ละตัว โดยระบบจะประกอบไปด้วยเซิร์ฟเวอร์ทำหน้าที่เป็นระบบฐานข้อมูลและเว็บให้บริการการจองและไมโครคอนโทรลเลอร์ทำหน้าที่ในการควบคุมเครื่องหยอดเหรียญ แสดงผล ควบคุมสวิตช์เพื่อทำการเปิดปิดเครื่องซักผ้า พร้อมทั้งรับส่งข้อมูลการใช้งานเครื่องซักผ้ากับฐานข้อมูลผ่านเครือข่ายอินเตอร์เน็ต โดยระบบสามารถคำนวณรายได้และบันทึกการเข้าใช้งานของผู้ใช้บริการที่ทำการจองด้วย

ABSTRACT

This thesis is to design and build a system for controlling washing machine via web service on Internet by using only one vending machine. The system has ability to manage 4 washing machines in order to reduce cost in installing a vending machine for each washing machine. The system consists of a server which serves as a database server and a web service for machine booking and a microcontroller used to control the vending machine and on/off switching and display command. The proposed system can record the data of machine booking and usage which the income obtained from the service can be achieved.

# สารบัญ


หน้า

กิตติกรรมประกาศ I
บทคัดย่อ II
สารบัญ III
สารบัญรูป VI
สารบัญตาราง X

**บทที่ 1 บทนำ**

1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ 2
1.3 ขอบเขตของปริญญานิพนธ์ 2
1.4 ผลลัพธ์สุดท้ายที่จะได้รับ 2

**บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง**

2.1 เครื่องซักผ้า 3
2.2 เครื่องหยอดเหรียญชนิดเหรียญเดียวอัตโนมัติ 6
2.3 ไมโครโปรเซสเซอร์ และไมโครคอนโทรลเลอร์ 8
2.3.1 บอร์ด CP-JR ARM7 LPC2368 8
2.3.2 การ Download Hex file ให้กับ MCU ของบอร์ด 9
2.3.3 ไมโครคอนโทรลเลอร์ตระกูลเอวีอาร์ 10
2.3.4 บอร์ดอาร์ดูอิโน่ 13
2.4 การสื่อสารข้อมูล 13
2.4.1 การสื่อสารแบบอนุกรม 13
2.4.2 มาตรฐานอาร์เอส 232 (RS-232) 14
2.4.3 ทีทีแอล 14
2.4.4 การแปลงระดับแรงดัน 15
2.5 ไอซี ISD1790P 16
2.6 วงจรสวิตช์ (Keypad) 18
2.7 ตัวต้านทานไวแสง (Light Dependent Resistor: LDR) 19
2.8 อินเตอร์เน็ต (Internet) 20
2.9 แอปเซิร์ฟ (Appserv) 22
2.10 การใช้งานโปรแกรมคิวที ครีเอเตอร์ (QT Creator) 25
2.11 การใช้งานโปรแกรม KEIL uVision 25

# สารบัญ(ต่อ)

หน้า

**บทที่ 3 การออกแบบและการจัดทำปริญญานิพนธ์**

3.1 การออกแบบ 26
3.1.1 การออกแบบอุปกรณ์ฮาร์ดแวร์ 26
3.1.1.1 การเชื่อมต่อระบบ 26
3.1.1.2 เครื่องหยอดเหรียญอัตโนมัติ 27
3.1.1.3 วงจรบันทึกและเล่นเสียง 27
3.1.1.4 ไมโครคอนโทรลเลอร์ 29
3.1.1.5 ภาคแสดงผล 30
3.1.1.6 วงจรสวิตช์ (Keypad) สำหรับกรอกรหัสผ่าน 31
3.1.1.7 การตรวจสอบสถานะการทำงานของเครื่องซักผ้า 32
3.1.1.8 การควบคุมและการตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า 38
3.1.2 การออกแบบซอฟต์แวร์ 39
3.1.2.1 ไมโครคอนโทรลเลอร์หลัก 39
3.1.2.2 เซิร์ฟเวอร์และฐานข้อมูล 51
3.2 เครื่องมือที่ใช้ในการทดลอง 67
3.3 การจัดเก็บผลการทดลอง 67
3.3.1 การทำงานของเครื่องหยอดเหรียญ 67
3.3.2 การทำงานของวงจรบันทึกและเล่นเสียง 68
3.3.3 การทำงานของจอแสดงผลแอลซีดี 69
3.3.4 วงจรตรวจสอบสถานะโดยใช้แอลดีอาร์รับแสง 69
3.3.5 วงจรตรวจสอบสถานะโดยใช้ออปโต้คัปเปลอร์ 69
3.3.6 การทำงานของส่วนการรับส่งข้อมูล 70
3.3.7 ระบบการยืนยันตนด้วยรหัสผ่าน 70
3.3.8 การทำงานของฐานข้อมูล 70

**บทที่ 4 ผลการทดลอง**

4.1 การทำงานของเครื่องหยอดเหรียญ 71
4.2 การการทำงานของวงจรบันทึกและเล่นเสียง 71
4.3 การทำงานของจอแสดงผลแอลซีดี 73
4.4 การรับค่ารหัสผ่านโดยใช้คีย์แพด 74
4.5 การทำงานของวงจรตรวจสอบสถานะของเครื่องซักผ้าวงจร 75
4.5.1 วงจรตรวจสอบสถานะโดยใช้แอลดีอาร์รับแสง 75
4.5.2 วงจรตรวจสอบสถานะโดยใช้ออปโต้คัปเปลอร์ 76

# สารบัญ(ต่อ)

หน้า


**บทที่ 4 ผลการทดลอง(ต่อ)**

4.6 การทำงานของส่วนการรับส่งข้อมูล 77
4.7 ระบบการยืนยันตนด้วยรหัสผ่าน 78
4.8 การส่งอีเมล์แจ้งเตือนผู้ใช้งาน 80
4.9 การทำงานของฐานข้อมูล 81
4.10 การคำนวณรายได้ 82
4.11 กล่องรับเหรียญและกล่องควบคุม 83
4.12 การทดสอบการใช้งานระบบ 84
4.12.1 สำหรับผู้ใช้บริการที่เดินเข้ามาใช้งานหน้ากล่องควบคุมโดยตรง (ไม่ผ่านระบบการจองใดๆ) 84
4.12.2 สำหรับผู้ใช้บริการที่ต้องการทำการจอง กรณีของผู้ที่ไม่ได้เป็นสมาชิก 85
4.12.3 สำหรับผู้ใช้บริการที่ต้องการทำการจอง กรณีผู้ที่เป็นสมาชิก 88

**บทที่ 5 สรุปผลและข้อเสนอแนะ**

5.1 สรุปผลการทดลอง 94
5.2 ข้อเสนอแนะ 94

**บรรณานุกรม**

**ภาคผนวก**

# สารบัญรูป


| รูปที่ | | หน้า |
|---|---|---|
| 2.1 | ส่วนต่างๆ ของเครื่องซักผ้า | 4 |
| 2.2 | แผงควบคุมของเครื่องซักผ้า | 5 |
| 2.3 | ส่วนประกอบต่างๆ ของเครื่องหยอดเหรียญ | 7 |
| 2.4 | ตัวอย่างแสดงการทำงานของโปรแกรม Flash magic | 9 |
| 2.5 | การจัดวางขาของเอทีเมกา168 | 12 |
| 2.6 | ตัวอย่างฮาร์ดแวร์ของเอวีอาร์ | 12 |
| 2.7 | รูปแบบในการส่งข้อมูล | 14 |
| 2.8 | การเรียงขาของไอซีแม็กซ์ 232 | 15 |
| 2.9 | ไอซี ISD1790P | 16 |
| 2.10 | ขาต่างๆ ของไอซี ISD1790P | 16 |
| 2.11 | สวิตซ์กดขนาด 3 แถว 3 บรรทัด (3x3) | 18 |
| 2.12 | ตัวต้านทานไวแสง | 19 |
| 2.13 | กราฟแสดงความสัมพันธ์ระหว่างความเข้มแสง (Lux) และค่าความต้านทาน (Ω) | 20 |
| 3.1 | การเชื่อมต่อของระบบ | 26 |
| 3.2 | การเชื่อมต่อเครื่องหยอดเหรียญกับไมโครคอนโทรลเลอร์ | 27 |
| 3.3 | วงจรบันทึกเสียง | 28 |
| 3.4 | วงจรบันทึกเสียงที่ใช้งานจริง | 28 |
| 3.5 | ไมโครคอนโทรลเลอร์หลัก ARM7 LPC-2368 | 29 |
| 3.6 | ไมโครคอนโทรลเลอร์รอง ATMEGA168 | 30 |
| 3.7 | การเชื่อมต่อสายสัญญาณจากไมโครคอนโทรลเลอร์หลักไปยังจอแสดงผลแอลซีดี | 31 |
| 3.8 | โครงสร้างภายในของระบบควบคุมเครื่องซักผ้า ยี่ห้อซัมซุง รุ่น WA85G5 | 32 |
| 3.9 | วงจรรับแสงโดยใช้แอลดีอาร์ | 32 |
| 3.10 | ด้านบนของเครื่องซักผ้า ยี่ห้อซัมซุง รุ่น WA85G5 | 33 |
| 3.11 | เปิดฝาบนของเครื่องซักผ้าออก | 33 |
| 3.12 | เปิดแผงควบคุมเครื่องซักผ้า | 34 |
| 3.13 | เปิดแผงควบคุมเครื่องซักผ้า | 34 |
| 3.14 | การติดตั้งวงจรตรวจสอบสถานะ | 35 |
| 3.15 | เดินสายไฟออกมาจากแผงควบคุม | 35 |
| 3.16 | สวิตซ์ Power และ Start | 36 |
| 3.17 | สวิตซ์ Power และ Switch บนแผงควบคุม | 36 |
| 3.18 | การติดตั้งสายสัญญาณ | 37 |
| 3.19 | ปิดฝาแผงวงจร | 37 |
| 3.20 | วงจรตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า | 38 |

# สารบัญรูป(ต่อ)


| รูปที่ | | หน้า |
|---|---|---|
| 3.21 | วงจรควบคุมและตรวจสอบสถานะการทางานของอุปกรณ์ไฟฟ้า | 38 |
| 3.22 | โฟลว์ชาร์ตแสดงขั้นตอนการทำงานของไมโครคอนโทรลเลอร์หลัก | 39 |
| 3.23 | การเชื่อมต่อบอร์ดทดลองกับไมโครคอนโทรลเลอร์ | 40 |
| 3.24 | ตัวอย่างโปรแกรม Easyweb | 41 |
| 3.25 | ไฟล์ของ Easyweb | 41 |
| 3.26 | โค้ดตัวอย่างที่ใช้ในการออกแบบ | 42 |
| 3.27 | กำหนดไอพีแอดเดรสต้นทางและปลายทาง | 43 |
| 3.28 | โฟลว์ชาร์ตและโค้ดโปรแกรมการแสดงผลผ่านจอแอลซีดี | 44 |
| 3.29 | โฟลว์ชาร์ตและโค้ดโปรแกรมการทางานการตรวจสอบสัญญาณเครื่องหยอดเหรียญ | 44 |
| 3.30 | ขาใช้งานของคีย์แพด | 45 |
| 3.31 | โฟลว์ชาร์ตและโค้ดโปรแกรมการตรวจสอบรหัสผ่าน | 46 |
| 3.32 | โฟลว์ชาร์ตและโค้ดโปรแกรมการส่งค่าสถานะเครื่องซักผ้าไปยังเซิร์ฟเวอร์ | 47 |
| 3.33 | โฟลว์ชาร์ตและโค้ดโปรแกรมการรับค่ารหัสผ่านจากเซิร์ฟเวอร์ | 48 |
| 3.34 | การรับค่าและแสดงค่ารหัสผ่าน | 49 |
| 3.35 | การตรวจสอบรหัสผ่านจากผู้ใช้และเซิร์ฟเวอร์ | 50 |
| 3.36 | การตรวจสอบจำนวนเหรียญและการสั่งเครื่องซักผ้าทำงาน | 50 |
| 3.37 | ช่องและพอร์ตเชื่อมต่อสัญญาณ RS232 | 51 |
| 3.38 | โฟลว์ชาร์ตและโค้ดโปรแกรมการทำงานของเซิร์ฟเวอร์ผ่านโปรแกรม QT Creator | 52 |
| 3.39 | การทำงานของเว็บเซิร์ฟเวอร์ร่วมกับเครื่องลูกข่าย | 53 |
| 3.40 | หน้าต่างของพีเอชพีมายแอดมิน | 54 |
| 3.41 | ฐานข้อมูลที่ต้องการสร้างชื่อ “customer” | 54 |
| 3.42 | การสร้างตารางใหม่ | 55 |
| 3.43 | การกำหนดค่าต่างๆให้กับรายละเอียดต่างๆของตาราง | 56 |
| 3.44 | การสร้างตารางใหม่ที่เสร็จเรียบร้อย | 57 |
| 3.45 | ข้อมูลที่อยู่ในฐานข้อมูลตามรายละเอียดของตารางที่สร้างขึ้นมา | 57 |
| 3.46 | หน้าจอสำหรับเพิ่มข้อมูล | 58 |
| 3.47 | หน้าต่างการลบข้อมูลภายในตาราง | 58 |
| 3.48 | หน้าต่างการแก้ไขข้อมูลในตาราง | 59 |
| 3.49 | หน้าต่างการลบตารางออกจากฐานข้อมูล | 59 |
| 3.50 | การเพิ่มฟิลด์ใหม่ลงไปในตารางที่มีอยู่ | 60 |
| 3.51 | หน้าต่างการสร้างชื่อสำหรับฟิลด์ใหม่ | 60 |
| 3.52 | หน้าต่างการลบฐานข้อมูล | 61 |

# สารบัญรูป(ต่อ)


| รูปที่ | | หน้า |
|---|---|---|
| 3.53 | หน้าเว็บเพจ | 61 |
| 3.54 | โฟลว์ชาร์ตที่แสดงสถานะของเครื่องซักผ้า | 62 |
| 3.55 | โฟลว์ชาร์ตการแสดงรหัสผ่าน | 63 |
| 3.56 | โฟลว์ชาร์ตการตั้งเวลาการจอง | 64 |
| 3.57 | หน้าเว็บแสดงฐานข้อมูล | 65 |
| 3.58 | หน้าเว็บการคำนวณรายได้ | 65 |
| 3.59 | โฟลว์ชาร์ตแสดงการทำงานการส่งอีเมล์แจ้งเตือนผู้ใช้งาน | 66 |
| 3.60 | การเก็บผลสัญญาณจากเครื่องหยอดเหรียญ | 68 |
| 3.61 | การเก็บผลการทดลองของวงจรบันทึกและเล่นเสียง | 68 |
| 3.62 | การเก็บผลแรงดันที่ขาคอลเลคเตอร์เมื่อเครื่องซักผ้าทำงาน | 69 |
| 3.63 | การทำงานของส่วนการรับส่งข้อมูล | 70 |
| 3.64 | การสร้างฐานข้อมูลโดยรวม | 70 |
| 4.1 | สัญญาณจากเครื่องหยอดเหรียญในโหมด N.O. (Normally Open) | 71 |
| 4.2 | สัญญาณไซน์เปรียบเทียบระหว่างอินพุตและเอาต์พุต | 72 |
| 4.3 | สัญญาณแสดงสถานะของไอซีบันทึกและเล่นเสียง | 72 |
| 4.4 | สถานะทั่วไป (ยังไม่มีการใช้งาน) | 73 |
| 4.5 | แสดงรหัสผ่านที่ผู้ใช้กดจำนวน 4 หลัก | 73 |
| 4.6 | ข้อความเมื่อผู้ใช้บริการกดรหัสผ่านไม่ถูกต้อง | 73 |
| 4.7 | แสดงจำนวนเหรียญที่ผู้ใช้บริการหยอด | 73 |
| 4.8 | แสดงสถานะเมื่อไม่มีเครื่องซักผ้าเครื่องใดว่าง | 74 |
| 4.9 | การกดตัวเลขผ่านคีย์แพดผ่าน Serial monitor | 74 |
| 4.10 | แรงดันวัดที่ขาคอลเลคเตอร์เมื่อเครื่องซักผ้าทำงาน (ไฟติด) หรือ -0.4 โวลต์ | 75 |
| 4.11 | แรงดันวัดที่ขาคอลเลคเตอร์เมื่อเครื่องซักผ้าสิ้นสุดการทำงาน (ไฟดับ) หรือ 3 โวลต์ | 75 |
| 4.12 | ระดับแรงดันไฟฟ้าจากออปโต้คัปเปลอร์ 0 โวลต์ (สถานะไม่ว่าง) | 76 |
| 4.13 | ระดับแรงดันไฟฟ้าจากออปโต้คัปเปลอร์ 5 โวลต์ (สถานะว่าง) | 76 |
| 4.14 | การทำงานการรับส่งข้อมูลระหว่างไมโครคอนโทรลเลอร์กับเซิร์ฟเวอร์ | 77 |
| 4.15 | การตรวจจับการรับส่งข้อมูลด้วยโปรแกรมวายชาร์ก | 78 |
| 4.16 | หน้าเว็บแสดงรหัสผ่าน เมื่อมีผู้ใช้บริการทำการจอง | 78 |
| 4.17 | จอแอลซีดีที่เชื่อมต่อกับไมโครคอนโทรลเลอร์ เพื่อยืนยันการใช้งานจากผู้ใช้บริการ | 79 |
| 4.18 | โปรแกรมวายชาร์กตรวจจับการรับส่งค่ารหัสผ่านระหว่างหน้าเว็บและไมโครคอนโทรลเลอร์ | 79 |

# สารบัญรูป(ต่อ)


| รูปที่ | | หน้า |
|---|---|---|
| 4.19 | การส่งอีเมล์แจ้งเตือนผู้ใช้งาน | 80 |
| 4.20 | ฐานข้อมูล revenue พร้อมฟิลด์ต่างๆ | 81 |
| 4.21 | ข้อมูลที่เข้ามาในฐานข้อมูล revenue | 81 |
| 4.22 | ฐานข้อมูล revenue แสดงเป็นตารางในหน้าเว็บ | 82 |
| 4.23 | การคำนวณรายได้ | 82 |
| 4.24 | อุปกรณ์ภายในกล่องควบคุม | 83 |
| 4.25 | กล่องควบคุม | 84 |
| 4.26 | บล็อกไดอะแกรมแสดงการทำงานของระบบสำหรับผู้ใช้บริการที่เดินเข้ามาใช้งานที่กล่องควบคุมโดยตรง | 85 |
| 4.27 | บล็อกไดอะแกรมแสดงการทำงานของระบบสำหรับผู้ใช้บริการที่ต้องการทำการจอง กรณีของผู้ที่ไม่ได้เป็นสมาชิก | 86 |
| 4.28 | หน้าเว็บเพจ | 86 |
| 4.29 | หน้าเว็บเพจการจอง แสดงการเลือกวันเวลาในการจอง | 87 |
| 4.30 | เมื่อผู้ใช้บริการกดจองและสามารถจองได้ หน้าเว็บเพจจะแสดงรหัสผ่าน 2342 | 87 |
| 4.31 | การที่ผู้ใช้บริการกดจอง แต่ไม่สามารถจองได้ จะแสดงข้อความ “จองไม่ได้” | 88 |
| 4.32 | บล็อกไดอะแกรมแสดงการทำงานของระบบสำหรับผู้ใช้บริการที่ต้องการทำการจอง กรณีผู้ที่เป็นสมาชิก | 89 |
| 4.33 | หน้าเว็บสำหรับสมาชิก | 89 |
| 4.34 | การล็อกอินเข้าใช้งาน | 90 |
| 4.35 | เมื่อล็อกอินสำเร็จ หน้าเว็บเพจจะแสดงข้อมูลของผู้ใช้บริการ | 90 |
| 4.36 | หน้าเว็บเข้าสู่การจอง และการเลือกเวลาจอง | 91 |
| 4.37 | ผู้ใช้บริการวสามารถจองได้ หน้าเว็บเพจจะแสดงรหัสผ่าน 3896 | 91 |
| 4.38 | ผู้ใช้บริการไม่สามารถจองได้ หน้าเว็บเพจจะแสดงข้อความ “จองไม่ได้” | 92 |
| 4.39 | จำนวนเงินคงเหลือหลังจากการหักเงินเมื่อเข้าสู่การจอง | 92 |
| 4.40 | ข้อมูลการใช้งานของผู้ใช้บริการ | 93 |
| 4.41 | ผู้ใช้บริการไม่สามารถได้ เนื่องจากเงินในระบบไม่เพียงพอ | 93 |

# สารบัญตาราง


| ตารางที่ | | หน้า |
|---|---|---|
| 2.1 | อ้างอิงรายละเอียดของหมายเลขฟังก์ชันต่างๆ | 6 |
| 2.2 | ส่วนประกอบต่างๆ ที่สำคัญของไมโครคอนโทรลเลอร์ | 11 |
| 2.3 | รายละเอียดของขาต่างๆ ที่เกี่ยวข้องกับการสื่อสารข้อมูล | 15 |
| 2.4 | รายละเอียดของขาต่างๆ ของไอซี ISD1790P | 17 |
| 3.1 | ตำแหน่งขาใช้งานจอแสดงผลแอลซีดี | 30 |
| 3.2 | การเชื่อมต่อขาต่างๆระหว่างไมโครคอนโทรลเลอร์และคีย์แพด | 45 |
| 3.3 | เลขฐานสิบเทียบกับรหัสแอสกี้ | 49 |

# บทที่ 1

# บทนำ

## 1.1 ความเป็นมาและความสำคัญของปัญหา

ในอดีตถึงปัจจุบันมนุษย์มีการพัฒนาเครื่องมืออำนวยความสะดวกสบายขึ้นมากมาย เช่น รถยนต์, ตู้เย็น, โทรทัศน์, โทรศัพท์, หม้อหุงข้าว ฯลฯ สำหรับการใช้ชีวิตประจำวัน ซึ่งการซักผ้าก็มีการพัฒนาขึ้นมาเป็นลำดับ การพัฒนาเครื่องซักผ้าขึ้นโดยในยุคแรกๆ เป็นเครื่องซักผ้าแบบกึ่งอัตโนมัติซึ่งยังคงต้องใช้คนช่วยในการซักผ้าแต่ละครั้ง จนกระทั่งมีการพัฒนาเครื่องซักผ้าอัตโนมัติขึ้นในปัจจุบันที่สามารถเลือกโปรแกรมการซักผ้าที่เหมาะสมในการซักผ้าแต่ละครั้งทำให้การซักผ้าเป็นเรื่องง่ายและมีเวลาเหลือที่จะปฏิบัติภารกิจอื่นๆ ได้ แต่เครื่องซักผ้าอัตโนมัติยังคงมีราคาสูง เราจึงไม่สามารถที่จะจัดหามาใช้กันได้ทุกครัวเรือน หรือในกรณีพักอาศัยอยู่ตามอพาร์ตเมนต์ หรือตามหอพักต่างๆ ซึ่งภายในห้องจะมีพื้นที่ที่จำกัดมาก จึงไม่สะดวกที่จะมีเครื่องซักผ้าอัตโนมัติไว้ภายในห้อง จากเหตุผลดังกล่าวจึงได้เกิดธุรกิจเครื่องซักผ้าหยอดเหรียญขึ้นเพื่ออำนวยความสะดวกให้กับบุคคลที่ต้องการใช้งาน

ดังนั้น ด้วยปัญหาดังกล่าวผู้จัดทำปริญญานิพนธ์จึงมีความคิดออกแบบและสร้างวงจรควบคุมเครื่องหยอดเหรียญควบคุมเครื่องซักผ้าอัตโนมัติขึ้นใหม่ โดยวงจรควบคุมเครื่องหยอดเหรียญควบคุมเครื่องซักผ้าอัตโนมัติที่สร้างขึ้นใหม่ 1 ชุด จะมีความสามารถควบคุมเครื่องซักผ้าได้ 4 เครื่อง โดยผู้ใช้งานสามารถเลือกและจองเครื่องซักผ้าผ่านระบบอินเตอร์เน็ตได้ ในเชิงธุรกิจจะเป็นการลดต้นทุนของการที่จะซื้อเครื่องหยอดเหรียญ และเพิ่มความสะดวกกับผู้ใช้งานที่จะสามารถเห็นสถานะของเครื่องซักผ้าว่ามีการใช้งานอยู่หรือไม่ โดยทำการตรวจสอบผ่านระบบอินเตอร์เน็ตได้ โดยระบบจะประกอบไปด้วยไมโครคอนโทรลเลอร์ ทำหน้าที่ในการควบคุมเครื่องหยอดเหรียญและสั่งการไปควบคุมสวิตช์เปิดปิด รับค่าและส่งค่าผ่านระบบอินเตอร์เน็ต และเริ่มการทำงานสำหรับเครื่องซักผ้าที่ผู้ใช้งานได้จองเอาไว้ รวมถึงมีการบันทึกข้อมูลการใช้งานเครื่องซักผ้าแต่ละเครื่องในฐานข้อมูลและสามารถคำนวณรายได้ที่เกิดจากการให้บริการด้วย

## 1.2 วัตถุประสงค์

1. เพื่อออกแบบและสร้างระบบควบคุมเครื่องซักผ้าหลายเครื่องโดยใช้เครื่องหยอดเหรียญเพียงเครื่องเดียว
2. ใช้ระบบอินเตอร์เน็ตเพื่อเพิ่มความสะดวกในการใช้งานเครื่องซักผ้าแก่ผู้ให้บริการและผู้ใช้บริการ
3. เพื่อพัฒนาเป็นระบบควบคุมเครื่องซักผ้าอัตโนมัติในเชิงธุรกิจต่อไป

## 1.3 ขอบเขตของปริญญานิพนธ์

สมรรถนะของระบบควบคุมเครื่องซักผ้าอัตโนมัติที่นำเสนอจะมีดังนี้

1. ระบบการใช้บริการการจองเครื่องซักผ้าผ่านระบบอินเตอร์เน็ต
2. สามารถรองรับการให้บริการแบบเดิมได้ (สำหรับกรณีผู้ใช้บริการเข้ามาใช้บริการโดยไม่ผ่านระบบจองทางอินเตอร์เน็ต)
3. สามารถเก็บข้อมูลการใช้งานเครื่องซักผ้าในฐานข้อมูลและสามารถนำข้อมูลมาคำนวณหารายได้ของการให้บริการได้
4. รองรับการทำงานของเครื่องซักผ้าจำนวน 4 เครื่อง

## 1.4 ผลลัพธ์สุดท้ายที่จะได้รับ

1. ได้ระบบสำหรับควบคุมเครื่องซักผ้าอัตโนมัติที่รองรับการทำงานของเครื่องซักผ้า 4 เครื่อง
2. ผู้ใช้งานสามารถเลือกจองเครื่องซักผ้า และตรวจสอบสถานะการทำงานของเครื่องซักผ้าผ่านอินเตอร์เน็ตได้
3. ได้ต้นแบบในการพัฒนาระบบสำหรับควบคุมเครื่องซักผ้าหยอดเหรียญในเชิงธุรกิจต่อไป

# บทที่ 2

# ทฤษฎีและหลักการที่เกี่ยวข้อง

## 2.1 เครื่องซักผ้า

เครื่องซักผ้า (Washing Machine) คือเครื่องจักรที่ใช้สำหรับทำความสะอาดหรือขจัด แยกสิ่งสกปรกออกจากเนื้อผ้า โดยใช้องค์ประกอบร่วมในการทำงาน เช่น น้ำ ความร้อน เคมี เครื่องซักผ้าในปัจจุบันมีหลากหลายรูปแบบ ซึ่งสามารถแบ่งออกเป็นกลุ่มๆ ได้หลายลักษณะ เช่น แยกตามขนาดของเครื่อง ลักษณะของเครื่อง หรือลักษณะการใช้งาน เครื่องซักผ้าประเภทใช้ตามบ้านเรือน สามารถแบ่งได้ 2 ประเภท ดังนี้

1. เครื่องซักผ้า 2 ถังกึ่งอัตโนมัติ

เครื่องฝาบน 2 ถัง กลุ่มนี้เป็นเครื่องประเภทประหยัด ราคาถูก ส่วนใหญ่เป็นเครื่องที่ผลิตในประเทศแถบเอเชีย เช่น ญี่ปุ่น เกาหลี จีน ข้อเสียของเครื่องแบบนี้คือผู้ใช้ต้องคอยเฝ้าดูขณะเครื่องซักผ้าทำงาน และเครื่องประเภทนี้ใช้น้ำและเคมีในแต่ละครั้งปริมาณมาก

2. เครื่องซักผ้าอัตโนมัติ

เครื่องซักผ้ากลุ่มนี้มีราคาสูงกว่ากลุ่มแรก ซึ่งสามารถแบ่งได้เป็น 3 ประเภท คือ เครื่องเปิดฝาหน้า เครื่องเปิดฝาด้านบน และเครื่องเปิดฝาด้านข้าง

1) เครื่องเปิดฝาหน้า

เป็นเครื่องซักผ้าที่ส่วนใหญ่ผลิตมาจากประเทศแถบยุโรป เช่น Electrolux Sanusi แต่ในปัจจุบันเครื่องประเภทนี้ผลิตในเอเชียหลายยี่ห้อ เช่น เครื่องฝาหน้า Samsung LG เป็นต้น ข้อดีของเครื่องประเภทนี้คือ ทนทาน ประหยัดน้ำ เคมี และส่วนใหญ่สามารถทำความร้อนในตัวได้ แต่ข้อเสียคือราคาค่อนข้างสูง

2) เครื่องเปิดฝาด้านบนแบ่งได้เป็น 2 กลุ่ม

- ฝาบนใบพัดแกนสูง ซึ่งส่วนใหญ่เป็นเครื่องที่ผลิตในอเมริกาเช่น SpeedQueen Worldpool เครื่องซักผ้ากลุ่มนี้มีราคาค่อนข้างสูง น้ำหนักมาก ปริมาณการซักขนาดตั้งแต่ 7.5-12 กิโลกรัม

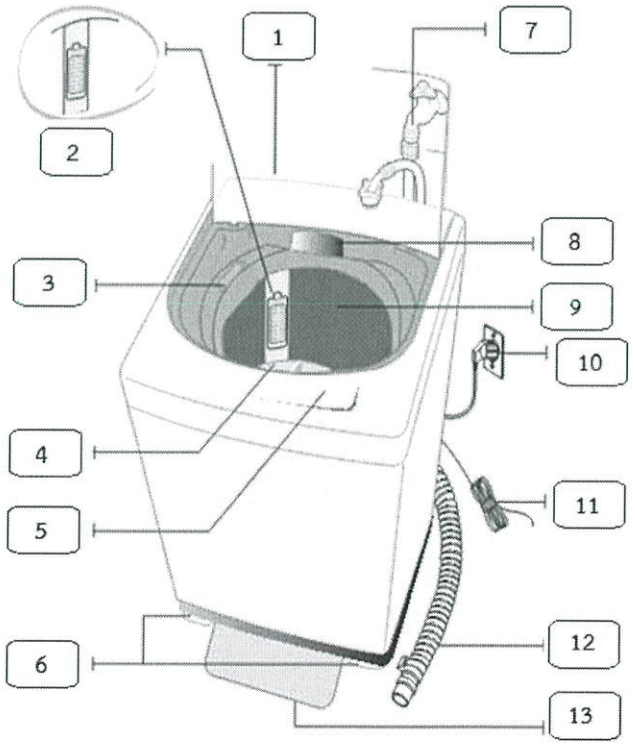
- ฝาบนใบพัดแกนต่ำ เป็นที่นิยมมากในบ้านเรามีให้เลือกหลากหลายยี่ห้อ ส่วนใหญ่เป็นเครื่องที่ผลิตจากประเทศในแถบเอเชีย เช่น ญี่ปุ่น เกาหลี จีน ข้อดีของเครื่องประเภทนี้คือราคาค่อนข้างต่ำ มีให้เลือกหลากหลายใช้งานง่ายและยืดหยุ่นมีขนาดถังซักตั้งแต่ 5 กิโลกรัมจนถึง 15 กิโลกรัมข้อเสียของกลุ่มเครื่องประเภทฝาบนคือใช้น้ำและเคมีในการซักแต่ละครั้งมาก ส่วนใหญ่ไม่มีระบบทำน้ำร้อนในตัว

3) เครื่องอัตโนมัติเปิดฝาข้าง

ในปัจจุบันแทบไม่มีให้เห็นในตลาดบ้านเรา ส่วนใหญ่เป็นเครื่องที่ผลิตจากยุโรป ลักษณะของเครื่องประเภทนี้ คล้ายกับเครื่องเปิดฝาด้านบน ถังซักตั้งวางในแนวนอน เปิดฝาด้านข้างสำหรับเอาผ้าเข้าออก ข้อดีของเครื่องประเภทนี้คือประหยัดน้ำเคมี และมีรอบสลัดที่สูง การทำงานค่อนข้างนิ่งไม่สั่น

### 2.1.2 คำอธิบายส่วนต่างๆ ของเครื่องซักผ้า

ส่วนต่างๆ ของเครื่องซักผ้ามีคำอธิบาย ดังรูปที่ 2.1



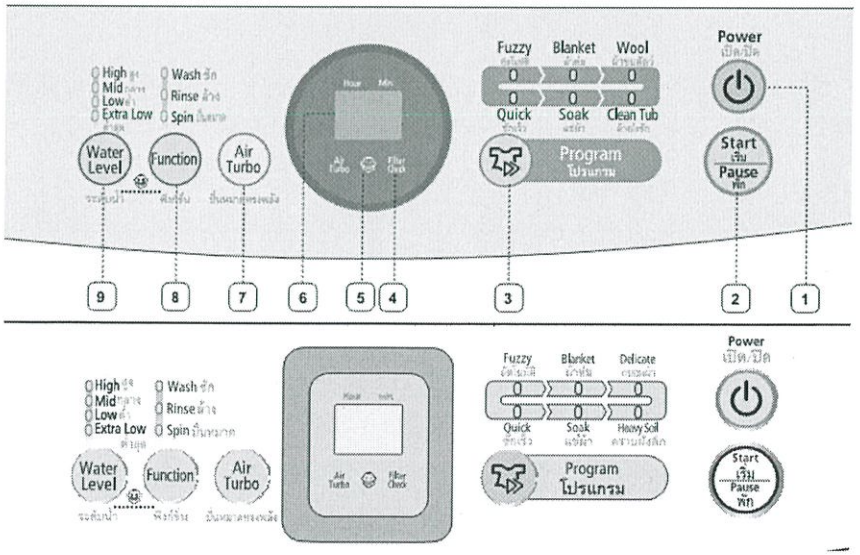

รูปที่ 2.1 ส่วนต่างๆ ของเครื่องซักผ้า

คำอธิบายแต่ละหมายเลขมีดังนี้

1. ฝาปิด โปรดปิดฝาตลอดเวลาที่ซักและปั่นผ้า
2. ตัวกรอง เครื่องจะดักจับเศษด้วยระหว่างการซัก
3. ช่องใส่น้ำยาปรับผ้านุ่ม
4. ใบพัด
5. แผงควบคุมด้านหน้า
6. ขาตั้งปรับได้ ปรับความยาวของขา เมื่อติดตั้งเครื่องซักผ้า
7. ท่อจ่ายน้ำ(เย็น)
8. ช่องใส่ผงซักผ้า
9. ถังปั่นหมาด
10. สายไฟ
11. สายลงกราวด์ ถ้าท่อน้ำเป็นท่อโลหะ โปรดครอบปลายของสายกราวด์
12. ท่อน้ำทิ้ง
13. แผงกันหนู

### 2.1.3 แผงควบคุมเครื่องซักผ้า

แผงควบคุมเครื่องซักผ้า มีคำอธิบายดังรูปที่ 2.2 และตารางที่ 2.1




รูปที่ 2.2 แผงควบคุมของเครื่องซักผ้า

ตารางที่ 2.1 อ้างอิงรายละเอียดของหมายเลขฟังก์ชันต่างๆ

| หมายเลขฟังก์ชัน | รายละเอียด | หมายเลขฟังก์ชัน | รายละเอียด |
|---|---|---|---|
| 1 | เปิด/ปิด | 6 | สัญญาณแสดงเวลาคงเหลือ การตั้งเวลา ข้อผิดพลาด |
| 2 | เริ่ม/พัก | | |
| 3 | เลือกโปรแกรม | 7 | ปั่นหมาดทรงพลัง |
| 4 | ฟิลเตอร์เช็ค (Filter check) | 8 | ฟังก์ชันการซัก |
| 5 | ล็อคป้องกันเด็ก | 9 | ระดับน้ำ |

## 2.2 เครื่องหยอดเหรียญชนิดเหรียญเดียวอัตโนมัติ

### 2.2.1 คุณสมบัติ

1. ตัวหยอดเหรียญ หรือ อุปกรณ์หยอดเหรียญชนิดเหรียญเดียว เป็นชนิดเหรียญเดียวสามารถเปลี่ยนเหรียญตัวอย่างได้

2. สามารถต่อเคาน์เตอร์ได้

3. นิยมใช้งานสำหรับเครื่องซักผ้าหยอดเหรียญ, ตู้คาราโอเกะหยอดเหรียญ, ตู้น้ำหยอดเหรียญ, คอมพิวเตอร์หยอดเหรียญ เป็นต้น

### 2.2.2 หลักการทำงานของเครื่องหยอดเหรียญ

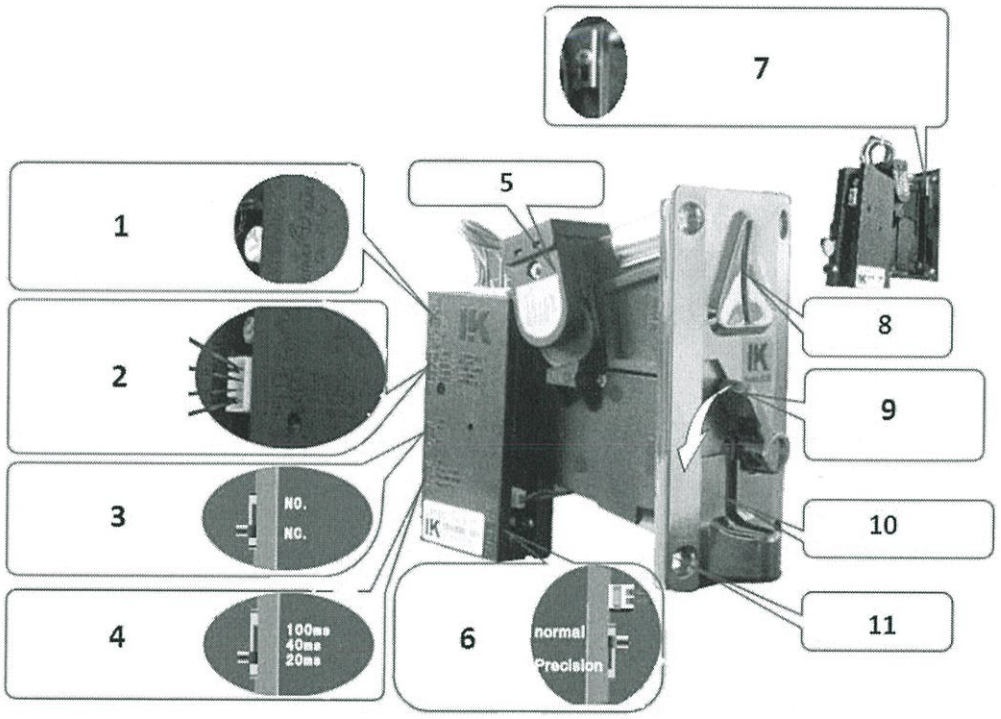
เริ่มต้นใช้งานเครื่องหยอดเหรียญ ทำการใส่เหรียญตัวอย่าง ลงในช่องสำหรับใส่เหรียญตัวอย่างก่อนทำการจ่ายไฟ 12 โวลต์ เข้าที่ตัวเครื่องหยอดเหรียญ (สายไฟสีแดงต่อไฟบวก 12 โวลต์ และสายไฟสีดำต่อกราวน์ (Ground) CPU ของตัวหยอดเหรียญจะเริ่มทำงาน เมื่อมีการหยอดเหรียญ ซีพียูในตัวเครื่องจะทำการเทียบกับค่าของเหรียญตัวอย่างโดย

1. ถ้าเหรียญที่ใส่เข้าไปตรงกับเหรียญตัวอย่าง เครื่องจะรับเหรียญที่หยอดลงด้านล่าง แล้วส่งสัญญาณออกไปที่สัญญาณเหรียญ (สายไฟสีขาว) โดยเหรียญ 1 เหรียญ จะได้รูปสัญญาณ 1 พัลส์

2. ถ้าเหรียญที่ใส่เข้าไปไม่ตรงกับเหรียญตัวอย่าง เครื่องจะคืนเหรียญที่หยอดออกมาด้านหน้าของอุปกรณ์หยอดเหรียญ โดยไม่มีการส่งสัญญาณออกไปที่ สัญญาณเหรียญ (สายไฟสีขาว)

### 2.2.3 ส่วนประกอบของเครื่องหยอดเหรียญ

สำหรับส่วนประกอบต่างๆของเครื่องหยอดเหรียญ แสดงได้ดังรูปที่ 2.3




รูปที่ 2.3 ส่วนประกอบต่างๆ ของเครื่องหยอดเหรียญ

คำอธิบายแต่ละหมายเลข มีดังนี้

1. ปรับความไวแสง ค่าที่ถูกตั้งมาจากโรงงาน คือ ค่าที่ดีที่สุด
2. สายเคาน์เตอร์ (สีเทา) 2 เส้น
   สายไฟเลี้ยง 12 โวลต์ (สีแดง) 1 เส้น
   สายกราวด์ (สีดำ) 1 เส้น
   สายสัญญาณเอาต์พุต (สีแดง) 1 เส้น
3. N.O. (Normally Open) เมื่อไม่มีการหยอดเหรียญจะมีสถานะเป็น High (Vcc) เมื่อมีการหยอดเหรียญ (และรับเหรียญ) จะมีสถานะเป็น Low (Ground) แล้วจึงเปลี่ยนสถานะกลับเป็น High ดังเดิม N.C. (Normally Close) เมื่อไม่มีการหยอดเหรียญจะมีสถานะเป็น Low (Ground) เมื่อมีการหยอดเหรียญ (และรับเหรียญ) จะมีสถานะเป็น High (Vcc) แล้วจึงเปลี่ยนสถานะกลับเป็น Low (Ground) ดังเดิม

4. ปรับคาบของสัญญาณ 20 มิลลิวินาที, 40 มิลลิวินาที หรือ 100 มิลลิวินาที

5. ที่ใส่เหรียญต้นแบบ

6. เลือกความไวในการจับสัญญาณ (ไว หรือ ปกติ)

7. ที่ปรับขนาดของช่อง สำหรับรับเหรียญเข้ามา

8. ช่องหยอดเหรียญ

9. คันโยก กดเมื่อต้องการเอาเหรียญที่หยอดค้างที่เครื่อง เนื่องจากเหรียญไม่ตรงกับเหรียญต้นแบบ

10.เมื่อมีเหรียญแปลกปลอม (เหรียญที่หยอดไม่ตรงกับเหรียญต้นแบบ) เครื่องจะคืนเหรียญออกมาตามช่องนี้

11.ช่องสำหรับนำเครื่องหยอดเหรียญไปติดตั้ง

## 2.3 ไมโครโปรเซสเซอร์ และไมโครคอนโทรลเลอร์

ไมโครโปรเซสเซอร์ โดยทั่วไปหมายถึงซีพียูเท่านั้น การใช้งานจำเป็นต้องต่อหน่วยความจำและอินพุต เอาต์พุตพอร์ตภายนอกเพิ่มเข้าไป แต่ถ้านำมารวมเป็นชิพเดี่ยวจะเรียกว่าไมโครคอนโทรลเลอร์แบบชิพเดี่ยว (Single Chip Microcontroller) คือ แผงวงจรรวมหรือไอซี (IC: Integrated Circuit) ที่ถูกออกแบบขึ้นมาให้มีขนาดเล็กเพียงชิพเดียว เหมาะสำหรับใช้ควบคุมอุปกรณ์ไฟฟ้าอื่นๆ โดยสามารถทำงานตามชุดคำสั่งที่กำหนดให้ทำได้ตามที่ผู้ใช้ต้องการ

### 2.3.1 บอร์ด CP-JR ARM7 LPC2368

เป็นบอร์ดไมโครคอนโทรเลอร์ในตระกูล ARM7TDMI-S Core ซึ่งเลือกใช้ไมโครคอนโทรเลอร์ 16/32-Bit ขนาด 100 Pin (LQFP) แบบใช้พลังงานต่ำเป็น MCU ประจำบอร์ด ซึ่งบอร์ดนี้เลือกใช้ MCU เบอร์ LPC2368 ของ Philips (NXP) โดยการออกแบบโครงสร้างของบอร์ดนั้นจะเน้นเรื่องของการจัดวางอุปกรณ์พื้นฐานที่จำเป็นต่อการนำไปประยุกต์ใช้งาน และ ศึกษาทดลอง ขั้นพื้นฐานรวมไว้อย่างครบถ้วน เช่น LED แสดงสถานะของ Output Logic และ Push Button Switch สำหรับสร้างสัญญาณ Logic เพื่อทดสอบการทำงานของ Input หรือ Volume ปรับค่าแรงดัน เพื่อใช้ทดสอบการทำงานของ A/D รวมถึงวงจรขับเสียงโดยใช้ Mini-Speaker สำหรับสร้างเสียง Beep ต่างๆ เป็นต้น นอกจากวงจรขั้นพื้นฐานดังกล่าวข้างต้นแล้ว บอร์ด CP-JR ARM7 LPC2368 ยังได้ออกแบบวงจรสำหรับประยุกต์ใช้งานขั้นสูงจัดเตรียมไว้ให้ใช้งานด้วย เช่น

- วงจรเชื่อมต่อกับ USB ซึ่งรองรับการเชื่อมต่อกับ USB 2.0 ได้ ซึ่งผู้ใช้สามารถนำบอร์ดไปพัฒนาเป็น USB Device แบบต่างๆได้โดยสะดวก

- วงจรเชื่อมต่อกับการ์ดหน่วยความจำ ซึ่งสามารถใช้งานได้กับการ์ดหน่วยความจำแบบ SD Card และการ์ดหน่วยความจำแบบ MMC Card

- วงจรเชื่อมต่อ Ethernet LAN แบบ 10/100Mb สามารถเชื่อมต่อกับเครือข่าย LAN แบบ Ethernet ได้ทั้งระบบ 10Mb และ 100Mb

- วงจรเชื่อมต่อกับ Dot-Matrix LCD แบบ Character พร้อมวงจรปรับความสว่าง

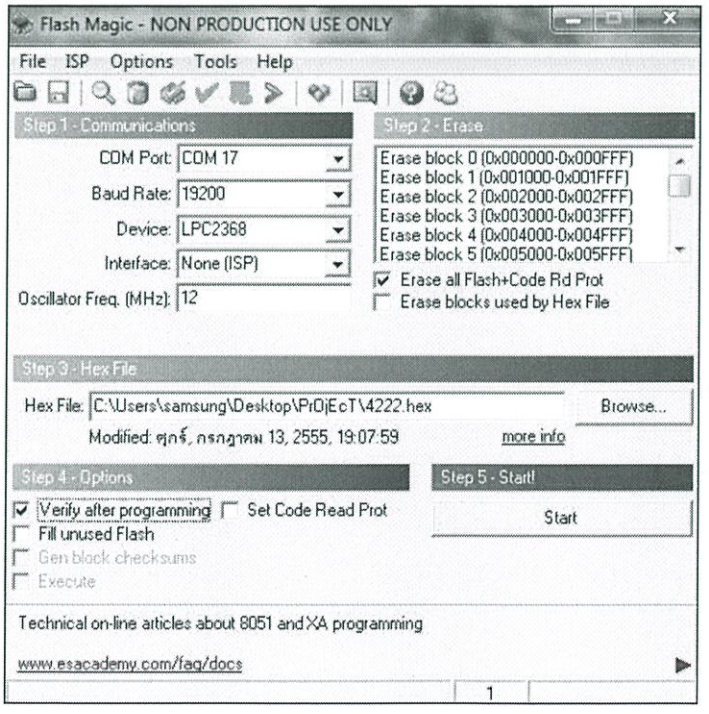
- วงจรสื่อสารข้อมูลแบบ RS232 พร้อม Line Driver จำนวน 2 ช่อง

- วงจรสื่อสารข้อมูลแบบ RS422/485 ทั้งแบบ Half-Duplex และ Full-Duplex

ซึ่งจะเห็นได้ว่า โครงสร้างโดยรวมของบอร์ด CP-JR ARM7 LPC2368 จะมีความหลากหลายและครบถ้วนพอสมควรเหมาะที่จะใช้เป็นบอร์ดทดลองเรียนรู้ และนำไปประยุกต์ดัดแปลงสร้างเป็น Application ใช้งานในด้านต่างๆได้มากมาย

### 2.3.2 การ Download Hex file ให้กับ MCU ของบอร์ด

การ Download Hex File ให้กับหน่วยความจำ Flash ของ MCU ในบอร์ดนั้น จะใช้โปรแกรมชื่อ Flash Magic ของ "Embedded System Academy,Inc" ซึ่งจะติดต่อกับ MCU ผ่าน Serial Port ของคอมพิวเตอร์ PC โดยโปรแกรมดังกล่าวสามารถดาวน์โหลดฟรีได้ที่ www.esacademy.com ตัวอย่างการทำงานของโปรแกรมแสดงดังรูปที่ 2.4




รูปที่ 2.4 ตัวอย่างแสดงการทำงานของโปรแกรม Flash magic

**ขั้นตอนการ Download HEX File ให้กับ MCU**

1. ต่อสายสัญญาณ RS232 ระหว่างพอร์ตสื่อสารอนุกรม RS232 ของ PC และบอร์ด UART-0

2. จ่ายไฟเลี้ยงวงจรให้กับบอร์ด ซึ่งจะสังเกตเห็น LED PWR ติดสว่าง

3. สั่ง Run โปรแกรม Flash Magic ซึ่งถ้าเป็น Version 4.02.260 จะได้ผลดังรูปที่ 2.4

4. เริ่มต้นกำหนดค่าตัวเลือกต่างๆให้กับโปรแกรมตามต้องการ ซึ่งในกรณีนี้ใช้กับ LPC2368 ของบอร์ดARM7 LPC2368 ของ อีทีที ให้เลือกกำหนดค่าต่างๆให้โปรแกรมดังนี้

1) เลือก COM Port ให้ตรงกับหมายเลข COM Port ที่ใช้งานจริง (ในตัวอย่างใช้ COM17)

2) ตั้งค่า Baud Rate อยู่ที่ระหว่าง 2400 - 115200 ซึ่งถ้าเลือกใช้ค่า Baud rate สูงๆ แล้วเกิดข้อผิดพลาด ให้ลดค่า Baud Rate ให้ต่ำลง (จากในรูปที่ 2.4 ใช้ค่า 19200)

3) กำหนด Device เป็น LPC2368

4) กำหนด Interface เป็น None ISP

5) กำหนดค่าคริสตอลออสซิลเลเตอร์ ให้ตรงกับที่ใช้ในจริงภายในบอร์ด โดยกำหนดให้มีหน่วยเป็น MHz ในที่นี้ใช้ค่า 12.000MHz ซึ่งต้องกำหนดเป็น 12

6) ให้กดสวิตช์ ISP LOAD และ RESET ที่บอร์ด "CP-JR ARM7 LPC2368" เพื่อทำการ Reset ให้ MCU ทำงานใน Boot Loader ตามขั้นตอนดังต่อไปนี้

- กดสวิตช์ ISP LOAD ค้างไว้
- กดสวิตช์ RESET โดยที่สวิตช์ ISP LOAD ยังกดค้างอยู่
- ปล่อยสวิตช์ RESET โดยที่สวิตช์ ISP LOAD ยังกดค้างอยู่
- ปล่อยสวิตช์ ISP LOAD เป็นลำดับสุดท้าย

5. เลือกรูปแบบการลบข้อมูลเป็น "Erase all Flash + Code Rd Prot"

6. เลือกกำหนด Option เป็น "Verify after programming"

7. ให้คลิกเลือกที่ "Browse" เพื่อทำการเลือกกำหนด HEX File ที่จะทำการสั่งดาวน์โหลด

8. ให้ทำการคลิกเลือกที่ "Start" ซึ่งโปรแกรม Flash Magic จะเริ่มต้นทำการดาวน์โหลดข้อมูลให้กับ MCU ทันที โดยสังเกตการทำงานที่ Status bar โดยในขั้นตอนนี้ให้รอจนกว่าการทำงานของโปรแกรมจะเสร็จสิ้น

9. เมื่อทำงานของโปรแกรมเสร็จเรียบร้อยแล้ว ให้กดสวิตช์ Reset ที่บอร์ด ซึ่ง MCU จะเริ่มต้นทำงานตามโปรแกรมที่ดาวน์โหลดเข้าไปทันที

### 2.3.3 ไมโครคอนโทรลเลอร์ตระกูลเอวีอาร์ (AVR Microcontroller)

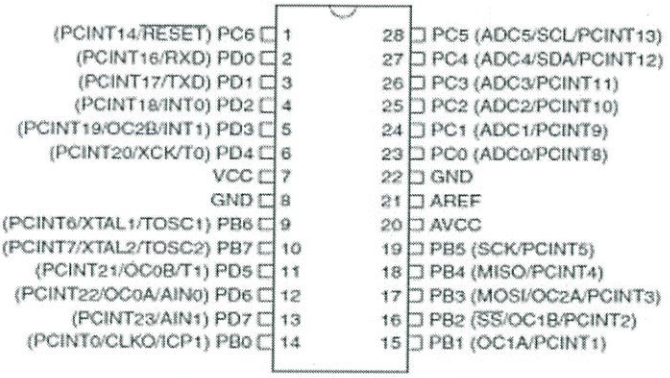
โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์ (Microcontroller) จะมีส่วนประกอบที่สำคัญตามตารางที่ 2.2

ตารางที่ 2.2 ส่วนประกอบต่างๆ ที่สำคัญของไมโครคอนโทรลเลอร์

| ส่วนประกอบ | ความหมาย |
|---|---|
| Central Processing Unit (CPU) | ซีพียู เป็นหน่วยประมวลผลกลาง ประกอบไปด้วย 2 ส่วนใหญ่ๆ คือ หน่วยควบคุม และหน่วยคำนวณ |
| Memory | หน่วยความจำเป็นที่เก็บข้อมูล และโปรแกรมสั่งงาน หน่วยความจำแบ่งเป็น 2 ประเภท คือ แรม และรอม |
| Port | พอร์ต เป็นส่วนที่ใช้ติดต่อกับภายนอก มี 2 ประเภท คือ อินพุตพอร์ต (Input Port) และ เอาท์พุตพอร์ท (Output Port) |
| Bus | บัส เป็นส่วนที่ใช้ในการแลกเปลี่ยนข้อมูลระหว่าง ซีพียู, หน่วยความจำ และ พอร์ต แบ่งเป็น 3 ประเภท คือ บัสที่อยู่ (Address Bus), บัสข้อมูล (Data Bus) และบัสควบคุม (Control Bus)<br>บัสที่อยู่: เป็นบัสทิศทางเดียวที่ส่งค่าแอดเดรสไปยังหน่วยความจำ<br>บัสข้อมูล: เป็นบัส 2 ทิศทางที่เป็นทางผ่านของข้อมูล<br>บัสควบคุม: เป็นบัสที่ส่งหรือรับสัญญาณควบคุมระหว่าง ซีพียู กับอุปกรณ์ภายนอก |

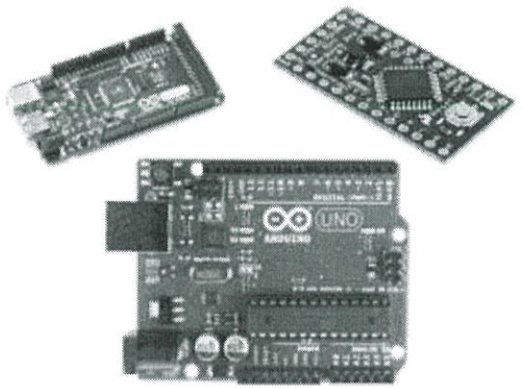
สถาปัตยกรรมของเอวีอาร์ประกอบด้วยหลากหลายขนาดแต่ในปริญญานิพนธ์นี้จะใช้ไมโครคอนโทรลเลอร์เอทีเมกา 168 (ATMega168) ซึ่งเป็นสถาปัตยกรรมแบบ 8 บิต (8 Bits Microcontroller) ที่มีการจัดเรียงขาต่างๆ ดังรูปที่ 2.5 และเอทีเมกา 168 นั้นจะมีข้อมูลจำเพาะต่างๆ ดังนี้

- หน่วยความจำโปรแกรมแบบแฟลช (Flash Memory) ขนาด 16 กิโลไบต์
- หน่วยความจำข้อมูลแบบเอสแรม (SRAM) ขนาด 2 กิโลไบต์
- หน่วยความจำข้อมูลแบบอีอีพรอม (EEPROM) ขนาด 1 กิโลไบต์
- สนับสนุนการรับส่งข้อมูลแบบไอแสควร์ซี (IC: Inter Integrated Bus)
- อินพุตพอร์ต (Input Port) และเอาท์พุตพอร์ท (Output Port) จำนวน 20 พอร์ท
- ตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล (Analog to Digital Converter) ขนาด 10 บิต จำนวน 6 ช่อง
- ความถี่ใช้งานสูงสุด 20 เมกกะเฮิรตซ์
- ทำงานตั้งแต่แรงดัน 1.8-5.5 โวลต์




รูปที่ 2.5 การจัดวางขาของเอทีเมกา168

โดยที่เอวีอาร์จะสามารถใช้เอวีอาร์สตูดิโอ (AVRStudio) หรืออาร์ดูอิโน่ (Arduino) ในการพัฒนาตัวไมโครคอนโทรลเลอร์ได้ โดยในปริญญานิพนธ์นี้จะเลือกใช้ซอฟต์แวร์ของอาร์ดูอิโน่ในการพัฒนาตัวโปรแกรมของไมโครคอนโทรลเลอร์เนื่องจากเป็นซอฟต์แวร์ที่ใช้งานง่ายและมีหมวดหมู่ของโปรแกรม (Program Library) เป็นจำนวนมาก เพราะว่าอาร์ดูอิโน่เป็นโครงการพัฒนาไมโครคอนโทรลเลอร์ของเอวีอาร์ซึ่งเป็นซอฟต์แวร์แบบที่สามารถให้ผู้ใช้พัฒนาได้ (Open Source) ที่ประกอบไปด้วยส่วนของฮาร์ดแวร์ที่ใช้ไมโครคอนโทรลเลอร์เอทีเมกา168 (Development Board) ตัวอย่างเช่นในรูปที่ 2.6 และส่วนของการพัฒนาโปรแกรมที่ใช้สำหรับติดต่อกับไมโครคอนโทรลเลอร์โดยจะทำการติดต่อผ่านทางอาร์เอส 232 (RS232)




รูปที่ 2.6 ตัวอย่างฮาร์ดแวร์ของเอวีอาร์

### 2.3.4 บอร์ดอาร์ดูอิโน่

การที่จะให้ไมโครคอนโทรลเลอร์ทำงานนั้นผู้ใช้จำเป็นจะต้องเขียนโปรแกรมคำสั่งเข้าไปสั่งการให้ไมโครคอนโทรลเลอร์ทำงานต่างๆ เช่น อ่านค่าจากตัวเซนเซอร์ แสดงผลค่าผ่านจอแสดงผล เป็นต้น ในปริญญานิพนธ์นี้จะใช้ตัวโปรแกรมที่มีชื่อว่าอาร์ดูอิโน่ในการพัฒนาซึ่งตัวโปรแกรมนี้จะมีลักษณะคล้ายๆ กับภาษาซี (C Language) โดยในการเขียนคำสั่งนั้นจะต้องมีฟังก์ชันอย่างน้อย 2 ฟังก์ชัน คือ ฟังก์ชันสำหรับการตั้งค่า (Set up Function) และฟังก์ชันสำหรับการลูปของโปรแกรม (Loop Function) โดยรูปแบบการเขียนฟังก์ชันทั้ง 2 เป็นไปดังนี้

- void setup (): ฟังก์ชันนี้เป็นฟังก์ชันการตั้งค่า ซึ่งมีไว้สำหรับกำหนดการทำงานของระบบหรือคุณสมบัติของระบบซึ่งคำสั่งที่อยู่ภายในฟังก์ชันนี้นั้นจะเปรียบเสมือนโปรแกรมย่อย
- void loop (): ฟังก์ชันนี้เป็นฟังก์ชันสำหรับคำสั่งที่ต้องการให้มีการวนลูป (ทางซ้าย) ซึ่งคำสั่งต่างๆ ที่อยู่ภายในจะเปรียบเสมือนโปรแกรมหลักโดยที่การเขียนคำสั่งต่างๆ นั้นจะคล้ายคลึงกับการเขียนภาษาซีและ สามารถหาหมวดหมู่ของโปรแกรมต่างๆ เพิ่มเติมได้

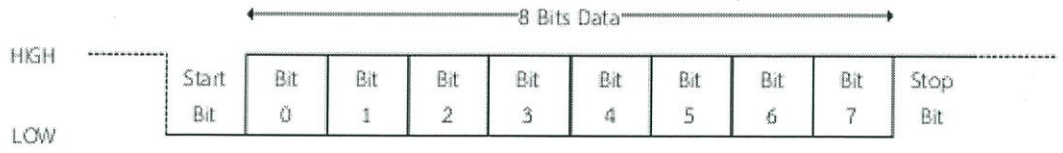
## 2.4 การสื่อสารข้อมูล

การสื่อสารข้อมูลคือการส่งข้อมูลที่ทำการเข้ารหัสแล้วระหว่างอุปกรณ์ 2 ชนิด โดยมีองค์ประกอบหลักสำคัญอยู่ 3 ส่วน คือ อุปกรณ์ฝั่งส่ง อุปกรณ์ฝั่งรับ และตัวกลาง ซึ่งตัวกลางจะสามารถแบ่งได้เป็น 2 ประเภทได้แก่ แบบมีสาย และแบบไร้สาย โดยจะสามารถแบ่งแยกรูปแบบในการสื่อสารข้อมูลได้เป็น 3 แบบคือ การสื่อสารทางเดียว การสื่อสารกึ่งสองทาง และการสื่อสารสองทาง

### 2.4.1 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมเป็นการสื่อสารที่มีการรับส่งข้อมูลทีละบิตเรียงต่อกันไปจนสิ้นสุดซึ่งการสื่อสารแบบนี้จะแตกต่างจากการสื่อสารแบบขนานเนื่องจากการสื่อสารแบบขนานนั้น ถ้าจำนวนข้อมูลมากขึ้นก็ต้องเพิ่มจำนวนสายสัญญาณจึงไม่มีความเหมาะสมในการใช้งานที่มีระยะทางในการรับส่งข้อมูลไกลๆ ซึ่งเป็นการสิ้นเปลืองค่าใช้จ่ายเป็นอย่างมาก แต่การสื่อสารแบบอนุกรมนั้นใช้จำนวนสายสัญญาณเพียง 2 ถึง 3 เส้นเท่านั้น จึงทำให้ใช้เวลาในการรับส่งข้อมูลมากกว่าการสื่อสารแบบขนานแต่ค่าใช้จ่ายจะลดลง โดยการสื่อสารแบบอนุกรมนั้นจะสามารถแบ่งแยกลักษณะของการสื่อสารได้เป็น 2 ประเภท คือ การสื่อสารแบบซิงโครนัส และการสื่อสารแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัส (Synchronous) การสื่อสารแบบซิงโครนัสนั้นจำเป็นจะต้องอาศัยสัญญาณนาฬิกามา

ช่วยกำหนดจังหวะในการสื่อสาร การสื่อสารแบบอะซิงโครนัส (Asynchronous) การสื่อสารแบบอะซิงโครนัสนั้นจะใช้รูปแบบในการส่งข้อมูล (Bit Pattern) เป็นตัวกำหนดว่าข้อมูลส่วนไหนเป็นข้อมูลอะไรซึ่งรูปแบบโดยทั่วไปจะเป็นดังนี้คือ เริ่มต้นด้วยบิตเริ่มต้น (Start Bit) ตามด้วยข้อมูล (Data) ตามด้วยบิตพาริตี้ (Parity Bit) สิ้นสุดด้วยบิตสิ้นสุด (Stop Bit) โดยการส่งข้อมูลแบบอะซิงโครนัสนั้นจะมีอุปกรณ์ที่เรียกว่า ยูอาร์ท (UART : Universal Asynchronous Receiver/Transmitter) เป็นตัวควบคุมในการรับส่งสามารถแสดงรูปแบบในการส่งข้อมูลได้ดังรูปที่ 2.7




รูปที่ 2.7 รูปแบบในการส่งข้อมูล

ในการสื่อสารแบบอนุกรมข้อมูลที่ใช้ในการสื่อสารจะอยู่ในลักษณะของกลุ่มบิตข้อมูล (Bit Stream) จำเป็นจะต้องสนใจในเรื่องอัตราเร็วในการรับส่งข้อมูลโดยจะใช้การกำหนดอัตราบอร์ด (Baud Rate) ในการกำหนดซึ่งจะมีค่ามาตรฐานได้แก่ 110 150 300 1200 2400 4800 9600 19200 เป็นต้น

### 2.4.2 มาตรฐานอาร์เอส 232 (RS-232)

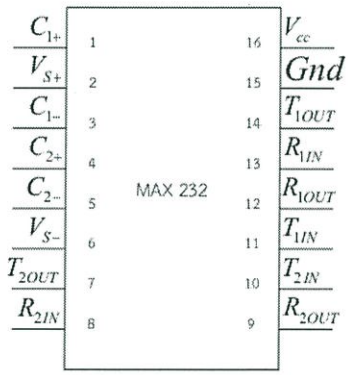
ในปัจจุบันมีอุปกรณ์ต่างๆ มากมายจากผู้ผลิตหลายๆ ค่ายจึงจำเป็นต้องมีตัวกำหนดมาตรฐานเพื่อให้อุปกรณ์ต่างๆ สามารถสื่อสารกันได้ โดยมาตรฐานที่นิยมใช้กันอย่างแพร่หลายนั้นก็คือมาตรฐานอาร์เอส 232 (RS 232 : Recommended Standard 232) ซึ่งจะมีระดับแรงดันอยู่ในช่วง -15 ถึง 15 โวลต์ โดยลอจิก “0” จะมีแรงดันอยู่ในช่วง 3 ถึง 15 โวลต์ และลอจิก “1” จะมีแรงดันในช่วง -15 ถึง -3 โวลต์ ซึ่งอาร์เอส 232 นั้นมีสายสัญญาณ 3 เส้นก็คือ สายส่งข้อมูล (Tx) สายรับข้อมูล (Rx) และกราวด์ (Gnd)

### 2.4.3 ทีทีแอล

ทีทีแอล (TTL : Transistor – Transistor Logic) เป็นการกำหนดระดับแรงดันที่เกิดขึ้นในยุคแรกๆ ใช้ในการสื่อสารระหว่างทรานซิสเตอร์โดยมีช่วงของระดับแรงดันอยู่ที่ 0 ถึง 5 โวลต์ และในปัจจุบันมีการพัฒนาระดับแรงดันทีทีแอลแบบทีทีแอลแรงดันต่ำ (LVTTL : Low-Voltage TTL) ซึ่งมีระดับแรงดันอยู่ที่ 0 ถึง 3.3 โวลต์

### 2.4.4 การแปลงระดับแรงดัน

การที่จะทำการสื่อสารระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์นั้นจำเป็นจะต้องทำการแปลงระดับแรงดัน เนื่องจากข้อมูลที่ออกมาจากพอร์ตอนุกรมของคอมพิวเตอร์มีระดับแรงดันตามมาตรฐานอาร์เอส 232 แต่ข้อมูลที่สามารถสื่อสารกับไมโครคอนโทรลเลอร์ได้นั้นต้องมีระดับแรงดันแบบทีทีแอล จะเห็นว่ามีระดับแรงดันที่ต่างกันจึงจำเป็นจะต้องทำการแปลงระดับแรงดันโดยที่การแปลงระดับแรงดันนั้นจะอาศัยไอซีเบอร์แม็กซ์ 232 (IC MAX232) ในการแปลงระดับแรงดันซึ่งแม็กซ์ 232 นั้นมีการจัดเรียงขาแสดงดังรูปที่ 2.8




รูปที่ 2.8 การเรียงขาของไอซีแม็กซ์ 232

จากรูปที่ 2.15 จะเห็นได้ว่าขาของไอซีแม็กซ์ 232 ที่เกี่ยวข้องกับการสื่อสารข้อมูลจะได้แก่ขา 7 8 9 10 11 12 13 และ 14 โดยที่รายละเอียดต่างๆ ของขาดังกล่าวแสดงได้ตามตารางที่ 2.3

ตารางที่ 2.3 รายละเอียดของขาต่างๆ ที่เกี่ยวข้องกับการสื่อสารข้อมูล

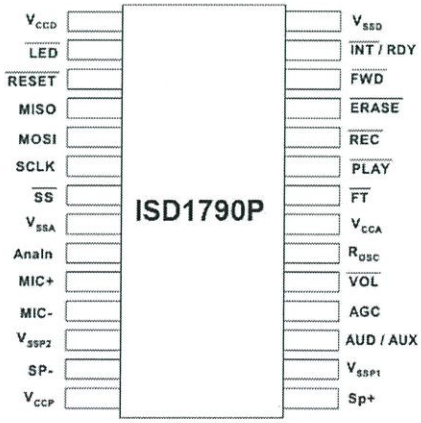
| หมายเลขของขา | ชื่อขา | รายละเอียดของขา | ระดับแรงดัน (โวลต์) |
|---|---|---|---|
| 7 | $T_{2OUT}$ | ส่งออกสัญญาณอาร์เอส 232 | -15 ถึง 15 |
| 8 | $R_{2IN}$ | รับเข้าสัญญาณอาร์เอส 232 | -15 ถึง 15 |
| 9 | $R_{2OUT}$ | ส่งออกสัญญาณทีทีแอล | 0 ถึง 3.3 หรือ 0 ถึง 5 |
| 10 | $T_{2IN}$ | รับเข้าสัญญาณทีทีแอล | 0 ถึง 3.3 หรือ 0 ถึง 5 |
| 11 | $T_{1IN}$ | รับเข้าสัญญาณทีทีแอล | 0 ถึง 3.3 หรือ 0 ถึง 5 |
| 12 | $R_{1OUT}$ | ส่งออกสัญญาณทีทีแอล | 0 ถึง 3.3 หรือ 0 ถึง 5 |
| 13 | $R_{1IN}$ | รับเข้าสัญญาณอาร์เอส 232 | -15 ถึง 15 |
| 14 | $T_{1OUT}$ | ส่งออกสัญญาณอาร์เอส 232 | -15 ถึง 15 |

จากตารางที่ 2.3 จะเห็นว่าไอซีแม็กซ์ 232 สามารถแปลงระดับแรงดันได้ 2 ชันแนล และการที่จะแปลงสัญญาณนั้นจำเป็นจะต้องเลือกใช้ขาของไอซีแม็กซ์ 232 ให้ตรงกัน คือ ถ้าเลือกใช้ขา 8 ในการรับเข้าสัญญาณอาร์เอส 232 สัญญาณทีทีแอลที่จะออกจะออกที่ขา 9 เป็นต้น

## 2.5 ไอซี ISD1790P

ไอซี ISD1790P ดังแสดงรูปที่ 2.9 เป็นไอซีที่อยู่ในกลุ่ม ISD1700 Series เป็นอุปกรณ์ที่ใช้สำหรับอัดเสียงข้อความ และเล่นเสียงที่อัดไว้ออกมาได้ ซึ่งไอซีตระกูลนี้จะมีหลายเบอร์ให้เลือกใช้ตามความต้องการของผู้ใช้งาน มีระยะเวลาในการอัดเสียงระหว่าง 26-120 วินาที และสามารถออกแบบให้มีความถี่สุ่มสัญญาณระหว่าง 4 กิโลเฮิรตซ์ – 12 กิโลเฮิรตซ์ ซึ่งระยะเวลาในการอัดเสียงและความถี่สุ่มสัญญาณนี้จะขึ้นอยู่กับค่าความต้านทานภายนอก ($R_{OC}$) ที่นำมาต่อเข้าในวงจร มีรายละเอียดของขาต่างๆ ดังรูปที่ 2.10 และตารางที่ 2.4

รูปที่ 2.9 ไอซี ISD1790P

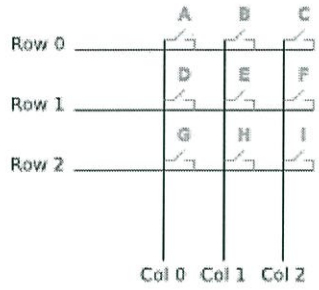



รูปที่ 2.10 ขาต่างๆ ของไอซี ISD1790P

ตารางที่ 2.4 รายละเอียดของขาต่างๆ ของไอซี ISD1790P

| ขา | รายละเอียด | ขา | รายละเอียด |
|---|---|---|---|
| $V_{CCD}$ | แหล่งจ่ายไฟเลี้ยง | SP+ | ขั้วบวกของลำโพง |
| LED | ไฟแสดงสถานะเอาต์พุต | $V_{SSP1}$ | กราวด์ |
| RESET | กดเพื่อกลับไปยังค่าเริ่มต้น | AUD/AUX | Auxiliary Output |
| MISO | เลื่อนข้อมูลไป 1 ครั้ง | AGC | ปรับอัตราขยาย |
| MOSI | ซีเรียลคล็อก (Serial Clock) : การคล็อกของเอสทีไออินเตอร์เฟส | VOL | ปรับเสียงได้ 8 ระดับ |
| SCLI | สำหรับเชื่อมต่ออุปกรณ์ภายนอก | $R_{OSC}$ | ตัวต้านทานสำหรับกำหนดความถี่ |
| SS | เลือกอุปกรณ์ภายนอกที่จะทำการติดต่อ | $V_{CCA}$ | ไฟเลี้ยงวงจร |
| $V_{SSA}$ | กราวด์ | FT | Feed-through |
| AnaIn | อินพุตอนาล็อก กดเพื่ออัดเสียง | PLAY | เล่นเสียง |
| MIC+ | วงจรขยายสัญญาณเสียงแบบไม่กลับเฟสของคอนเดนเซอร์ไมค์ | REC | อัดเสียง |
| MIC- | วงจรขยายสัญญาณเสียงแบบกลับเฟสของคอนเดนเซอร์ไมค์ | ERASE | ลบเสียง |
| $V_{SSP2}$ | กราวด์ | FWD | เลื่อนข้อมูลไปข้างหน้า |
| SP- | ขั้วลบของลำโพง | RDY/INT | หยุดการทำงานของข้อมูล |
| $V_{CCP}$ | ไฟเลี้ยงสำหรับลำโพง | $V_{SSD}$ | กราวด์ |

## 2.6 วงจรสวิตช์ (Keypad)




รูปที่ 2.11 สวิตช์กดขนาด 3 แถว 3 บรรทัด (3x3)

ในการทดลอง ได้มีการใช้วงจรสวิตช์ในการรับค่าตัวเลขซึ่งเป็นรหัสผ่านสำหรับผู้ใช้งาน ดังรูปที่ 2.11 ซึ่งการออกแบบวงจรสามารถแบ่งออกได้เป็น 2 ประเภทตามลักษณะของงาน ได้แก่

1. การต่อวงจรแบบสวิตช์ขาร่วม โดยในวิธีนี้จะเหมาะสำหรับงานที่มีความต้องการใช้สวิตช์จำนวนไม่มากนัก หรือในงานที่ต้องการแยกสัญญาณอินพุตของแต่ละจุดออกจากกันโดยสิ้นเชิง ซึ่งในวิธีการแบบนี้จะต้องใช้จำนวนบิตของอินพุต มากเท่ากับจำนวนสวิตช์ที่ต้องการจะใช้ในระบบ แต่อย่างไรก็ตามวิธีนี้มีความสะดวกและง่ายในการออกแบบวงจรและเขียนโปรแกรมเพื่อตรวจสอบหาตำแหน่งของสวิตช์ที่ถูกกดได้ง่ายกว่าวิธีการอื่นๆ เพราะการทำงานของสวิตช์แต่ละตัวจะแยกออกจากกันอย่างอิสระ แต่วิธีการนี้จะมีข้อเสียคือความสิ้นเปลืองจำนวนพอร์ตมากขึ้นเมื่อต้องใช้กับระบบที่มีความจำเป็นต้องใช้อินพุตมากๆ ซึ่งวิธีการนี้จะต่อวงจรโดยนำหน้าสัมผัสของสวิตช์ทุกตัวมารวมเข้าด้วยกันเป็นขาร่วม (common) ไว้ สำหรับหน้าสัมผัสอีกด้านหนึ่งของสวิตช์แต่ละตัวก็จะต่อเข้ากับพอร์ตอินพุตของระบบโดยตรง ซึ่งส่วนมากแล้วพอร์ตอินพุตนี้จะทำการคงสถานะของสัญญาณที่แน่นอนไว้ก่อนเสมอ โดยการต่อตัวต้านทานเข้ากับทุกบิตของพอร์ตอินพุตเข้ากับแหล่งจ่ายไว้ (Pull-Up Resistor) เพื่อให้ค่าสถานะของลอจิกของพอร์ตอินพุต มีค่าเป็น “1” อยู่ตลอดเวลาในขณะที่ยังไม่มีการกดสวิตช์ตัวใด สำหรับขาร่วมของสวิตช์ก็จะต่อลงกราวด์ไว้ ดังนั้นเมื่อมีการกดสวิตช์ตัวใดก็จะทำให้ตำแหน่งของบิตอินพุตนั้นมีค่าเป็น “0”

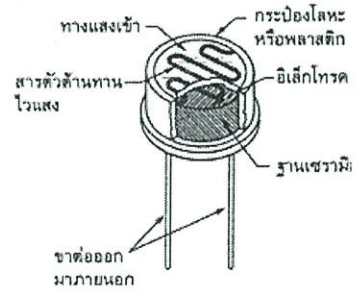
2. การต่อวงจรสวิตช์แบบเมทริกซ์ โดยวิธีการนี้เหมาะกับระบบที่มีความจำเป็นต้องใช้งานสวิตช์มากๆ เช่น วงจรแป้นพิมพ์ (keyboard) ที่ใช้สำหรับป้อนค่าตัวเลข ตัวอักษร และข้อความต่างๆ ซึ่งจะเกิดความไม่สะดวกสำหรับผู้ใช้งานเป็นอย่างมาก ถ้าออกแบบให้มีจำนวนสวิตช์น้อยๆ เพราะจะยากแก่การค้นหาตำแหน่งของตัวเลข ตัวอักษรที่อยู่ซ้อนกันอยู่ในคีย์เดียวหลายๆชั้น โดย

วิธีการนี้จะต้องใช้พอร์ต 2 ส่วน คือ พอร์ตสำหรับอ่านค่าสถานะของสวิตช์จากทางแถว (Row) และพอร์ตสำหรับทำหน้าที่ส่งค่าออกไปสแกนคีย์ในแต่ละหลัก (Column) ของวงจร โดยจำนวนของสวิตช์จะขึ้นอยู่กับขนาดของแถวและหลักที่ใช้ เช่น ถ้าเป็นขนาด 4x4 ก็จะได้ทั้งหมด 16 ตำแหน่ง เป็นต้น

วิธีการสแกนแบบเมทริกซ์นี้จะทำทีละหลัก โดยเริ่มจากหลักแรกไปหาหลักสุดท้ายตามลำดับ สำหรับลักษณะของการต่อวงจรโดยทั่วไปของวิธีการนี้จะนิยมคงสถานะของสัญญาณด้านที่เป็นอินพุตให้มีค่าเป็น "1" รอไว้ก่อนเสมอโดยการต่อตัวต้านทาน Pull-Up เข้ากับพอร์ตอินพุตรอไว้ก่อน โดยในการสแกนจะทำทางด้านหลัก โดยส่งค่าออกไปทางด้านพอร์ตอินพุตให้มีค่าเป็น "0" ครั้งละ 1 บิต แล้วก็อ่านค่าจากพอร์ตอินพุตเข้ามาตรวจสอบว่าทุกบิตยังคงเป็น "1" อยู่หรือไม่ ซึ่งถ้าพบว่ามีบิตใดเป็น "0" (Column Active = "0") ก็สามารถทราบได้ทันทีว่ามีการกดคีย์ขึ้นที่ตำแหน่งแถวและหลักนั้นๆ แต่ถ้าทุกบิตยังคงมีค่าเป็น "1" ก็ให้เปลี่ยนการสแกนไปยังหลัก ถัดไปอีกโดยทำเหมือนกันกับหลักแรกจนครบทุกหลัก

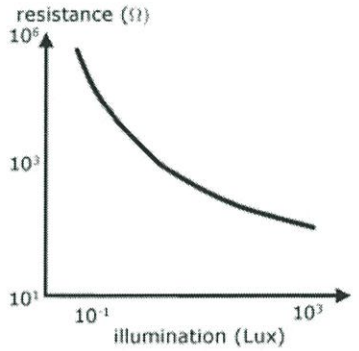
## 2.7 ตัวต้านทานไวแสง (Light Dependent Resistor: LDR)

ตัวต้านทางไวแสง คือ ความต้านทานชนิดที่ไวต่อแสง กล่าวคือ ตัวความต้านทานนี้สามารถเปลี่ยนสภาพทางความนำไฟฟ้า ได้เมื่อมีแสงมาตกกระทบ บางครั้งเรียกว่าโฟโตรีซีสเตอร์ (Photo Resistor) หรือโฟโตคอนดัคเตอร์ (Photo Conductor) เป็นตัวต้านทานที่ทำมาจากสารกึ่งตัวนำ (Semiconductor) ประเภทแคดเมี่ยมซัลไฟด์ (Cds : Cadmium Sulfide) หรือแคดเมี่ยมซิลินายส์ (CdSe : Cadmium Selenide) ซึ่งทั้งสองตัวนี้ก็เป็นสารประเภทกึ่งตัวนำที่เอามาฉาบลงบนแผ่นเซรามิกที่ใช้เป็นฐานรองแล้วต่อขาจากสารที่ฉาบไว้ออกมา ดังรูปที่ 2.12




รูปที่ 2.12 ตัวต้านทานไวแสง

เมื่อมีแสงมาตกกระทบที่แอลดีอาร์ จะทำให้ค่าความต้านทานภายในตัวแอลดีอาร์นั้นลดลง จะลดลงมากหรือน้อยขึ้นอยู่กับแสงที่ตกกระทบ ในกรณีที่ไม่มีแสงหรืออยู่ในตำแหน่งที่มืด ค่าความต้านทานภายในตัวแอลดีอาร์จะมีค่าเพิ่มมากขึ้น ดังรูปที่ 2.13 การทดสอบแอลดีอาร์อย่างง่ายๆ คือ ต่อสายมิเตอร์เข้ากับแอลดีอาร์ตั้งย่านวัดที่โอห์ม หาอุปกรณ์ให้แสงสว่าง เช่น ไฟฉายหรือหลอดไฟ โดยให้แสงตกกระทบที่ตัวแอลดีอาร์ตรงด้านหน้า แล้วสังเกตค่าความต้านทานจากมิเตอร์จะมีค่าลดลง ถ้ามีอุปกรณ์ไปบังแสงทำให้มืด ค่าความต้านทานจะเพิ่มขึ้น ตัวอย่างการนำเอาแอลดีอาร์ไปประยุกต์ใช้ เช่น วงจรเปิดปิดไฟในเวลากลางคืน, วงจรส่งสัญญาณเตือนเมื่อมีสิ่งกีดขวาง, วงจรหุ่นยนต์เดินตามเส้น เป็นต้น




รูปที่ 2.13 กราฟแสดงความสัมพันธ์ระหว่างความเข้มแสง (Lux) และค่าความต้านทาน (Ω)

## 2.8 อินเตอร์เน็ต (Internet)

อินเตอร์เน็ต คือเครือข่ายคอมพิวเตอร์ที่ระบบต่างๆเชื่อมโยงกัน มาจากคำว่า Inter Connection Network อินเตอร์เน็ตเป็นระบบเครือข่ายคอมพิวเตอร์ที่มีขนาดใหญ่ เครื่องคอมพิวเตอร์ทุกเครื่องทั่วโลกสามารถติดต่อสื่อสารถึงกันได้โดยใช้มาตรฐานในการรับส่งข้อมูลที่เป็นหนึ่งเดียว หรือที่เรียกว่าโปรโตคอล (Protocol) ซึ่งโปรโตคอลที่ใช้ในระบบเครือข่ายอินเตอร์เน็ตที่มีชื่อว่าทีซีพี/ไอพี (TCP/IP : Transmission Control/Internet Protocol) ลักษณะของระบบอินเตอร์เน็ตเป็นเสมือนใยแมงมุมที่ครอบคลุมทั่วโลกในแต่ละจุดที่เชื่อมต่ออินเตอร์เน็ตนั้นสามารถสื่อสารกันได้หลายเส้นทางตามความต้องการโดยไม่กำหนดตายตัว และไม่จำเป็นต้องไปตามเส้นทางโดยตรง อาจจะผ่านจุดอื่นๆหรือเลือกเส้นทางอื่นได้หลายๆเส้นทาง การติดต่อสื่อสารผ่านระบบเครือข่ายอินเตอร์เน็ตนั้นอาจเรียกว่า การติดต่อแบบไร้มิติ หรือไซเบอร์สเปซ (Cyberspace)

### 2.8.1 โปรโตคอลทีซีพี/ไอพี (TCP/IP)

โปรโตรคอล ทีซีพี/ไอพี เป็นชุดของโปรโตคอลที่ถูกใช้ในการสื่อสารผ่านเครือข่ายอินเตอร์เน็ต โดยมีวัตถุประสงค์เพื่อให้สามารถใช้สื่อสารจากต้นทางข้ามเครือข่ายไปยังปลายทางได้ และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้เองโดยอัตโนมัติ ถึงแม้ว่าในระหว่างทางอาจจะผ่านเครือข่ายที่มีปัญหาโปรโตคอลก็ยังคงหาเส้นทางอื่นในการส่งผ่านข้อมูลไปให้ถึงปลายทางได้

ชุดโปรโตคอลนี้ได้รับการพัฒนามาตั้งแต่ปี 1960 ซึ่งถูกใช้เป็นครั้งแรกในเครือข่าย ARPANET ซึ่งต่อมาได้ขยายการเชื่อมต่อไปทั่วโลกเป็นเครือข่ายอินเตอร์เน็ต ทำให้ทีซีพี/ไอพี เป็นที่ยอมรับอย่างกว้างขวางจนถึงปัจจุบัน

ทีซีพี/ไอพี มีจุดประสงค์ของการสื่อสารตามมาตรฐาน 3 ประการคือ

1. เพื่อใช้ติดต่อสื่อสารระหว่างระบบที่มีความแตกต่างกัน

2. ความสามารถในการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย เช่นในกรณีที่ผู้ส่งและผู้รับยังคงมีการติดต่อกันอยู่ แต่โหนดกลางที่ใช้เป็นผู้ช่วยรับ-ส่งเกิดเสียหายใช้การไม่ได้ หรือสายสื่อสารบางช่วงถูกตัดขาด กฎการสื่อสารนี้จะต้องสามารถจัดหาทางเลือกอื่นเพื่อทำให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ

3. มีความคล่องตัวต่อการสื่อสารข้อมูลได้หลายชนิดทั้งแบบที่ไม่มีความเร่งด่วน เช่น การจัดส่งแฟ้มข้อมูล และแบบที่ต้องการรับประกันความเร่งด่วนของข้อมูล เช่น การสื่อสารแบบ real-time และทั้งการสื่อสารแบบเสียง (Voice) และข้อมูล (data)

โปรโตคอลที่สำคัญในโปรโตคอลทีซีพี/ไอพี มีดังนี้

**1. ทีซีพี (TCP)**

ทีซีพีเป็นหนึ่งในโพรโทคอลหลักในเครือข่ายอินเตอร์เน็ต หน้าที่หลักของทีซีพี คือ ควบคุมการรับส่งข้อมูลระหว่างโฮส ถึง โฮส ในเครือข่าย เพื่อใช้แลกเปลี่ยนข้อมูลระหว่างกัน โดยตัวโปรโตคอลจะรับประกันความถูกต้อง และลำดับของข้อมูลที่ส่งผ่านระบบเครือข่าย นอกจากนั้นทีซีพียังช่วยจำแนกข้อมูลให้ส่งผ่านไปยังแอปพลิเคชันที่ทำงานอยู่บนโฮสเดียวกันให้ถูกต้องด้วย

งานหลักที่สำคัญของทีซีพีอีกงานหนึ่งคือ เป็นโปรโทคอลที่ขั้นกลางระหว่างแอปพลิเคชันและเครือข่ายไอพี ทำให้แอปพลิเคชันจากโฮสหนึ่ง สามารถส่งข้อมูลออกยังอีกโฮสหนึ่งผ่านเครือข่ายเปรียบเสมือนมีท่อส่งข้อมูลระหว่างกัน

ทีซีพี เป็นโปรโทคอลที่ได้รับความนิยมที่สุดในโลกของอินเตอร์เน็ต มีแอปพลิเคชันจำนวนมากที่ใช้โปรโทคอลทีซีพีเป็นสื่อกลางในการเชื่อมต่อ เช่น เวิลด์ไวด์เว็บ เป็นต้น

**2. ไอพีแอดเดรส (IP Address)**

ไอพีแอดเดรสคือหมายเลขประจำเครื่องคอมพิวเตอร์ ซึ่งประกอบด้วยตัวเลข 4 ชุด มีเครื่องหมายจุดขั้นระหว่างชุด ไอพีแอดเดรส ที่ใช้กันอยู่นี้เป็น ตัวเลขไบนารีขนาด 32 บิตหรือ 4 ไบต์ แต่เมื่อต้องการเรียกไอพีแอดเดรสจะใช้การแปลงเลขไบนารีหรือเลขฐานสองแต่ละไบต์ (8 บิต) ให้เป็นตัวเลขฐานสิบโดยมีจุดคั่น เช่น 192.168.1.3

การสื่อสารแถะรับส่งข้อมูลในระบบอินเตอร์เน็ตนั้นสิ่งสำคัญคือที่อยู่ของคอมพิวเตอร์แต่ละเครื่อง ดังนั้นเพื่อให้เกิดความถูกต้องแม่นยำ จึงได้มีการกำหนดหมายเลขประจำเครื่องที่เราเรียกว่า ไอพี แอดเดรส และเพื่อไม่ให้เกิดความสับสนและซ้ำกัน จึงได้มีการก่อตั้งองค์กรเพื่อแจกจ่าย ไอพีแอดเดรสโดยเฉพาะ ชื่อองค์กรว่า InterNIC (International Network Information Center) อยู่ที่ประเทศสหรัฐอเมริกา การแจกจ่ายนั้นทาง InterNIC จะแจกจ่ายเฉพาะ เน็ตเวิร์คแอดเดรส ให้แต่ละเครือข่าย ส่วนลูกข่ายของเครื่อง ทางเครือข่ายนั้นจะเป็น ผู้แจกจ่ายอีกทอดหนึ่ง ไอพีแอดเดรส แต่ละกลุ่มที่ได้รับการจัดสรร จะได้รับการควบคุมการกำหนดเส้นทางโดยอุปกรณ์จำพวกเราท์เตอร์ และสวิตชิ่ง ทำนองเดียวกัน หน่วยงานย่อยที่ได้รับไอพีแอดเดรส ไปเป็นกลุ่มก็สามารถนำไอพีแอดเดรส ที่ได้รับไปจัดสรรแบ่งกลุ่มด้วยอุปกรณ์เราเตอร์หรือสวิตชิ่งได้ การกำหนดไอพีแอดเดรส จะต้องอยู่ภายในกลุ่มของตนเท่านั้น มิฉะนั้น อุปกรณ์เราเตอร์จะไม่สามารถทำงานรับส่งข้อมูลได้

## 2.9 แอปเซิร์ฟ (Appserv)

แอปเซิร์ฟเป็นโปรแกรมที่ติดตั้งเพื่อใช้ทดสอบการใช้งานภาษาต่างๆในการพัฒนาโปรแกรมระบบหรือเว็บไซต์ โดยที่โปรแกรมแอปเซิร์ฟนี้รวบรวมเอาโอเพ่นซอร์ส ซอฟต์แวร์หลายๆ อย่างมารวมกัน เมื่อติดตั้งโปรแกรมแอปเซิร์ฟก็สามารถใช้โปรแกรมที่ติดมาทั้งหมดได้ โดยโปรแกรมที่ติดมามีดังนี้

**1. อาปาเช่ เว็บเซิร์ฟเวอร์ (Apache HTTP Server)**

เป็นซอฟต์แวร์สำหรับเปิดให้บริการเซิร์ฟเวอร์บนโพรโทคอลเอชทีทีพี (HTTP) โดยสามารถทำงานได้บนหลายระบบปฏิบัติการและเป็นโปรแกรมจำลองเว็บเซิร์ฟเวอร์ มีหน้าที่จัดเก็บโฮมเพจและส่งโฮมเพจไปยังเว็บเซิร์ฟเวอร์ที่เก็บโฮมเพจนั้นอยู่ อาปาเช่อยู่ในลักษณะของโอเพ่นซอร์สที่เปิดให้บุคคลทั่วไปสามารถเข้ามาร่วมพัฒนาส่วนต่างๆ ของอาปาเช่ได้ ซึ่งทำให้เกิดเป็นโมดูลที่เกิดประโยชน์

มากมาย ซึ่งเป็นโมดูลที่ทำให้อาปาเช่สามารถใช้ประโยชน์ และทำงานร่วมกับภาษาอื่นได้ แทนที่จะเป็นเพียงเซิร์ฟเวอร์ที่ให้บริการเพียงแค่เอชทีเอ็มแอลอย่างเดียว นอกจากนี้อาปาเช่ยังมีความสามารถอื่นๆ ด้วย เช่น การยืนยันตัวบุคคลหรือเพิ่มความปลอดภัยในการสื่อสารผ่าน โปรโทคอลเอชทีทีพี

**2. มายเอสคิวแอล (MySQL)**

โปรแกรมมายเอสคิวแอลเป็นที่นิยมใช้กันมากสำหรับฐานข้อมูลสำหรับเว็บไซต์ เช่น มีเดียวิกิ และ พีเอชพีบีบีและนิยมใช้งานร่วมกับภาษาโปรแกรม ซึ่งมักจะได้ชื่อว่าเป็นคู่ จะเห็นได้จากคู่มือคอมพิวเตอร์ต่างๆ ที่จะสอนการใช้งานมายเอสคิวแอล และพีเอชพีควบคู่กันไป นอกจากนี้ หลายภาษาโปรแกรมที่สามารถทำงานร่วมกับฐานข้อมูลมายเอสคิวแอล ซึ่งรวมถึง ภาษาซี ซีพลัสพลัส ปาสคาล ซีชาร์ป ภาษาจาวา ภาษาเพิร์ล พีเอชพี ไพทอน รูบี และภาษาอื่น โปรแกรมนี้ช่วยในการจัดการฐานข้อมูลและทำงานกับฐานข้อมูลในการจัดการฐานข้อมูล

**ฟังก์ชันในการติดต่อฐานข้อมูล**

พีเอชพีสามารถเชื่อมต่อข้อมูลได้หลายแบบ โดยที่การเชื่อมต่อกับฐานข้อมูลแบบต่างๆ ก็จะใช้ฟังก์ชั่นที่แตกต่างกันด้วย สำหรับฟังก์ชั่นในการเชื่อมต่อมายเอสคิวแอล ชื่อฟังก์ชั่นจะขึ้นต้นด้วยคำว่า mysql เป็นส่วนใหญ่ทำให้นำไปใช้ได้ง่าย โดยฟังก์ชั่นมีดังนี้

- ฟังก์ชัน mysql_connect

เป็นฟังก์ชันที่ใช้ในการการเชื่อมต่อกับฐานข้อมูลมายเอสคิวแอล เป็นฟังก์ชันแรกที่ต้องใช้เสมอในการติดต่อกับมายคิวเอสแอลและหากฟังก์ชันนี้ทำงานไม่สำเร็จ ก็ไม่สามารถทำงานอย่างอื่นต่อไปได้ ดังนั้นจึงควรตรวจสอบผลลัพธ์ของฟังก์ชันนี้ทุกครั้งก่อนจะดำเนินการใดๆ ต่อไป หากฟังก์ชันนี้ทั้งหมดทำงานสำเร็จหรือสามารถเชื่อมต่อกับมายเอสคิวแอลจะได้ค่าทรู (true) ถ้าการเชื่อมต่อไม่สำเร็จจะคืนค่ากลับมาเป็นฟอลท์ (false) มีรูปแบบดังนี้

int mysql_connect( string hostname, string username, string password);

- hostname คือชื่อ โฮสต์ (host) ที่ติดตั้งมายคิวเอสแอล
- username คือชื่อของผู้ใช้ที่มีสิทธ์เข้าใช้มายคิวเอสแอล
- password คือรหัสผ่านของผู้ใช้

- ฟังก์ชัน mysql_close

เป็นฟังก์ชันการปิดการเชื่อมต่อกับมายเอสคิวแอลหลังใช้งานเสร็จ

รูปแบบ : mysql_colse (connection_name) โดย connection_name คือตัวแปรที่เกิดจากการใช้ฟังก์ชัน mysql_connect()

- ฟังก์ชัน คำสั่ง mysql_select_db ( )
เป็นฟังก์ชันในการกำหนดชื่อฐานข้อมูลที่จะใช้งาน
รูปแบบ : mysql_select_db (db_name) โดยdb_name คือ ชื่อของฐานข้อมูล

**3. พีเอชพี (PHP)**

พีเอชพีเป็นภาษาคอมพิวเตอร์ในลักษณะเซิร์ฟเวอร์-ไซด์ สคริปต์ ภาษาพีเอชพีใช้สำหรับจัดทำเว็บไซต์ และแสดงผลออกมาในรูปแบบเอชทีเอ็มแอล โดยมีรากฐานโครงสร้างคำสั่งมาจากภาษาซี ภาษาจาวา และ ภาษาเพิร์ล ซึ่ง ภาษาพีเอชพี นั้นง่ายต่อการเรียนรู้ ซึ่งเป้าหมายหลักของภาษานี้ คือให้นักพัฒนาเว็บไซต์สามารถเขียนเว็บเพจที่มีความตอบโต้ได้อย่างรวดเร็ว ย่อมาจากคำว่า PHP Hypertext Preprocessor หรือชื่อเดิม Personal Home Page

ถึงแม้ว่าจุดประสงค์หลักของพีเอชพีใช้ในการแสดงผลเอชทีเอ็มแอลแต่ก็ยังสามารถสร้างเอ็กซ์ทีเอ็มแอล (XHTML) หรือ เอ็กซ์เอ็มแอล (XML) ได้ นอกจากนี้สามารถทำงานร่วมกับคำสั่งเสริมต่างๆ ซึ่งสามารถแสดงผลข้อมูลหลักพีดีเอฟ (PDF) พีเอชพีสามารถทำงานร่วมกับฐานข้อมูลได้หลายชนิด ซึ่งฐานข้อมูลส่วนหนึ่งที่รองรับได้แก่ ออราเคิล ดีเบส(dBase) มายเอสคิวแอล และอีกมากมาย

**ลักษณะเด่นของ PHP มีดังนี้**

- ใช้ได้ฟรี
- พีเอชพีเป็นโปรแกรมทำงานบนเซิร์ฟเวอร์ ดังนั้นจึงมีขีดความสามารถไม่จำกัด
- พีเอชพีสามารถทำงานได้บนระบบปฏิบัติการ ยูนิกซ์ (UNIX), ลีนุกซ์ (Linux), วินโดวส์ (Windows)
- เรียนรู้ง่าย เนื่องจากพีเอชพีฝังเข้าในเอชทีเอ็มแอล และใช้โครงสร้างและไวยากรณ์ภาษาง่ายๆ
- เร็วและมีประสิทธิภาพ
- ใช้ร่วมกับเอ็กซ์เอ็มแอลได้ทันที
- ใช้กับระบบแฟ้มข้อมูลได้
- ใช้กับข้อมูลตัวอักษรได้อย่างมีประสิทธิภาพ
- ใช้กับโครงสร้างข้อมูลได้หลายๆแบบ
- ใช้กับการประมวลผลภาพได้

**4. พีเอชพีมายแอดมิน (phpMyAdmin)**

เป็นสคริปต์ติดต่อฐานข้อมูลที่สร้างโดยภาษาพีเอชพี ซึ่งใช้จัดการฐานข้อมูลมายเอสคิวแอล ผ่านเว็บเบราว์เซอร์ โดยสามารถที่จะทำการสร้างฐานข้อมูลใหม่ หรือทำการสร้างตารางใหม่ และยังมีฟังก์ชันที่ใช้สำหรับการทดสอบการสืบค้นข้อมูลด้วยภาษาเอสคิวแอลพร้อมกันนั้น ยังสามารถทำการ insert delete update หรือแม้กระทั่งใช้ คำสั่งต่างๆ เหมือนกับกันการใช้ภาษาเอสคิวแอล ในการสร้างตารางข้อมูล

## 2.10 การใช้งานโปรแกรมคิวที ครีเอเตอร์ (QT Creator)

คิวทีเป็นเฟรมเวิร์กสำหรับสร้างแอปพลิเคชัน และส่วนติดต่อผู้ใช้ที่ครอสแพลตฟอร์ม (Cross-Platform) โดยเขียนเพียงครั้งเดียว แต่สามารถนำไปปรับใช้บนระบบปฏิบัติการ และเครื่องมือต่างๆ ได้มากมาย เช่น บนคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการ Windows, Mac, Linux, โทรศัพท์มือถือที่ใช้ Symbian หรือแท็บเล็ต โดยตัวคิวทีนั้นเป็นเฟรมเวิร์กสำหรับแอปพลิเคชันที่พัฒนาด้วยภาษาซีพลัสพลัส (C++) เป็นหลัก

## 2.11 การใช้งานโปรแกรม KEIL uVision

ในการพัฒนาโปรแกรมภาษาซีสำหรับไมโครคอนโทรลเลอร์ตระกูล ARM7 นั้น การใช้งานจะประกอบด้วย 2 ส่วนด้วยกันคือตัวเครื่องตัววงจรที่เรียกว่า ฮาร์ดแวร์ (Hardware) และอีกส่วนหนึ่งคือคำสั่งที่สั่งให้ไมโครคอนโทรลเลอร์ทำงานที่เรียกว่า ซอฟต์แวร์ (Software) คำสั่งที่ไมโครคอนโทรลเลอร์ต้องการคือชุดคำสั่งที่เป็นภาษาเครื่อง ซึ่งเป็นภาษาที่มนุษย์ไม่สามารถเข้าใจได้ ดังนั้นจึงเกิดภาษาอื่นที่มนุษย์เข้าใจได้แล้วใช้เครื่องมือแปลงภาษาไปเป็นภาษาเครื่องอีกครั้ง ภาษาที่นิยมใช้งานคือ ภาษาซี เครื่องมือที่ใช้ในการแปลงภาษาซีให้เป็นภาษาเครื่องที่นิยมมากตัวหนึ่งคือ KEIL uVision สามารถดาวน์โหลดได้ที่ http://www.keil.com ซึ่งเป็นแบบทดลองใช้ โปรแกรม KEIL uVision จะช่วยทำให้การเขียนโปรแกรมภาษาซีได้รวดเร็วยิ่งขึ้น และสามารถแปลงภาษาซีเป็นโค้ด HEX ได้เลย

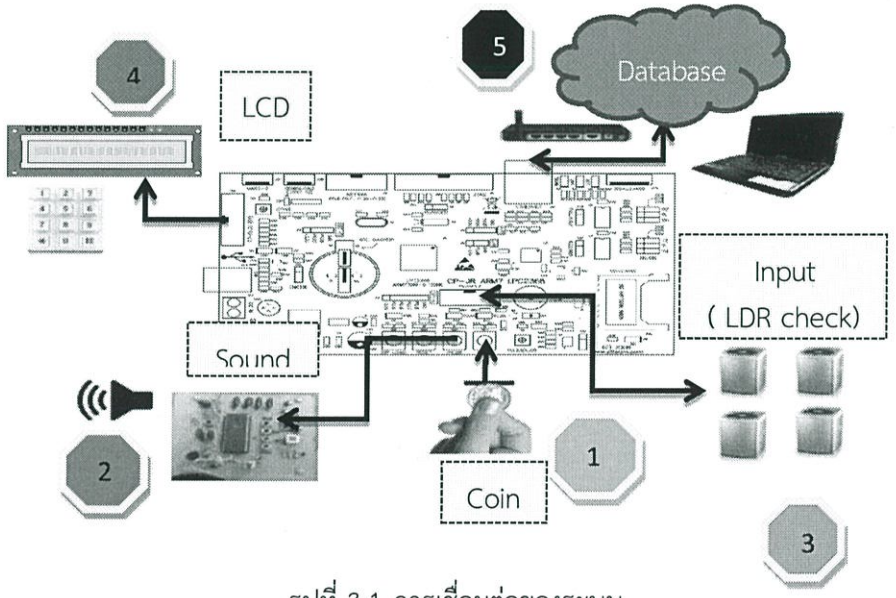
# บทที่ 3

# การออกแบบและการจัดทำปริญญานิพนธ์

## 3.1 การออกแบบ

### 3.1.1 การออกแบบอุปกรณ์ฮาร์ดแวร์

#### 3.1.1.1 การเชื่อมต่อระบบ




รูปที่ 3.1 การเชื่อมต่อของระบบ

การเชื่อมต่อของระบบนั้นได้ถูกออกแบบไว้ดังรูปที่ 3.1 โดยจะแบ่งออกเป็น 5 ส่วน ได้แก่

หมายเลข 1 คือ เครื่องหยอดเหรียญ จะต่อเข้ากับพอร์ตที่ 4.29

หมายเลข 2 คือ วงจรเสียง จะต่อเข้ากับพอร์ตที่ 4.28

หมายเลข 3 คือ วงจรตรวจสอบสถานะเครื่องซักผ้าและเอาต์พุตเพื่อส่งสัญญาณไปทำการเปิดสวิตซ์เครื่องซักผ้า จะต่อเข้ากับพอร์ตที่ 2.0 ถึง 2.7

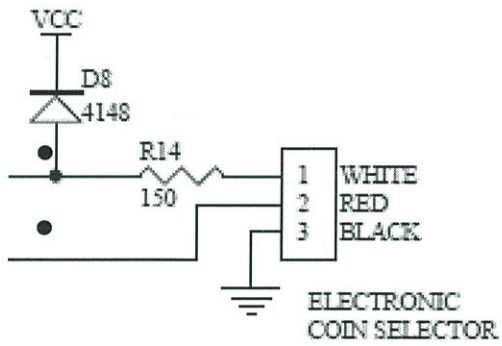
หมายเลข 4 คือ จอแอลซีดี จะต่อกับพอร์ตที่ 1.24 ถึง 1.31 และคีย์แพด จะต่อกับพอร์ต UART-2

หมายเลข 5 คือ อินเตอร์เน็ต จะถูกต่อกับพอร์ตอินเตอร์เน็ต

ซึ่งจะอธิบายรายละเอียดในส่วนย่อยไว้ในหัวข้อต่อไป

### 3.1.1.2 เครื่องหยอดเหรียญอัตโนมัติ

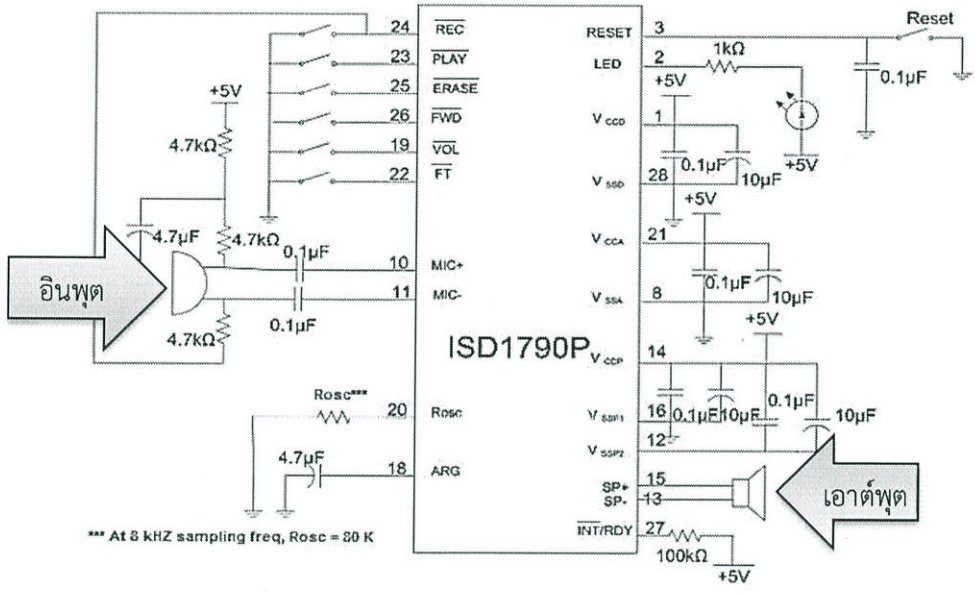
เครื่องหยอดเหรียญที่ใช้จะมีสายสัญญาณจำนวน 5 เส้น ได้แก่ สายไฟเลี้ยง 12 โวลต์ สายกราวด์ สายเคาน์เตอร์ (2 เส้น) และสายสัญญาณ โดยในการทดลองจะนำสายสัญญาณ (สีขาว) และกราวด์ (สีดำ) เชื่อมต่อกับไมโครคอนโทรลเลอร์ที่พอร์ต 4.29 โดยผ่านตัวต้านทานขนาด 150 โอห์มที่ต่อกับ Vcc ขนาด 3.3 โวลต์ เพื่อยกระดับสัญญาณให้ไมโครคอนโทรลเลอร์สามารถอ่านค่าได้ว่ามีสัญญาณเข้ามาหรือไม่ จากนั้นเขียนคำสั่งตรวจสอบสัญญาณที่เข้ามา เพื่อแสดงจำนวนเหรียญที่ผู้ใช้บริการหยอดผ่านหน้าจอแอลซีดีต่อไป ดังรูปที่ 3.2



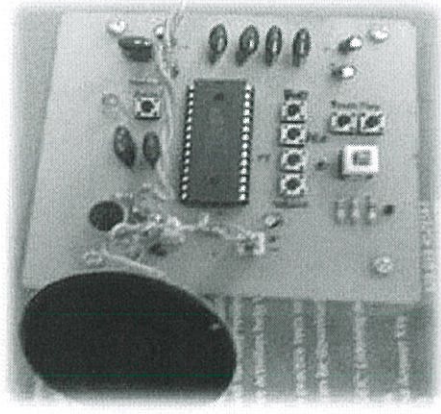

รูปที่ 3.2 การเชื่อมต่อเครื่องหยอดเหรียญกับไมโครคอนโทรลเลอร์

### 3.1.1.3 วงจรบันทึกและเล่นเสียง

วงจรบันทึกและเล่นเสียง (Sound Recorder) จะเลือกใช้ไอซีในกลุ่ม ISD1700 Series คือ ISD1790P เป็นอุปกรณ์ที่ใช้สำหรับอัดเสียงข้อความ และเล่นเสียงที่อัดไว้ออกมาได้ ซึ่งไอซีตระกูลนี้จะมีหลายเบอร์ให้เลือกใช้ตามความต้องการของผู้ใช้งาน มีระยะเวลาในการอัดเสียงระหว่าง 26-120 วินาที และสามารถออกแบบให้มีความถี่สุ่มสัญญาณระหว่าง 4 กิโลเฮิรตซ์ – 12 กิโลเฮิรตซ์ ซึ่งระยะเวลาในการอัดเสียงและความถี่สุ่มสัญญาณนี้จะขึ้นอยู่กับค่าความต้านทานภายนอก ($R_{OC}$) ที่นำมาต่อเข้าในวงจร ไอซีตัวนี้เป็นไอซีสำเร็จรูป สามารถต่อตัวต้านทานภายนอกได้ซึ่งใช้ตัวต้านทานขนาด 80 กิโลโอห์ม และสามารถควบคุมการทำงานได้จากการกดสวิตช์หรือจากการเขียนคำสั่งจากไมโครคอนโทรลเลอร์ โดยมีอินพุต คือ คอนเดนเซอร์ไมค์ และเอาต์พุต คือ ลำโพง ดังรูปที่ 3.3 และวงจรที่ใช้งานได้จริง ดังรูปที่ 3.4




รูปที่ 3.3 วงจรบันทึกเสียง

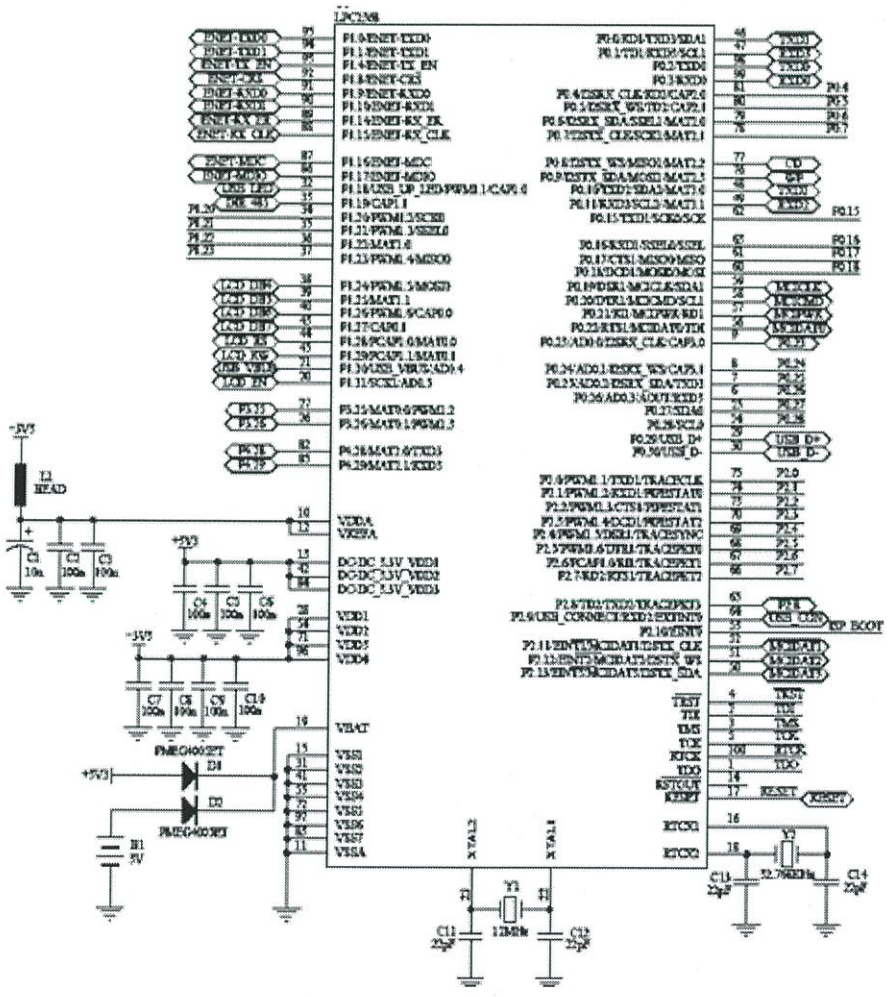


รูปที่ 3.4 วงจรบันทึกเสียงที่ใช้งานจริง

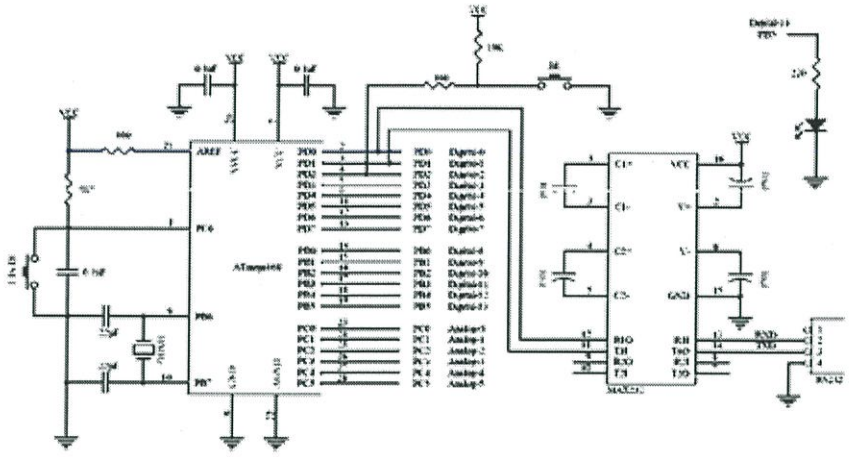
การเชื่อมต่อไมโครคอนโทรลเลอร์กับวงจรบันทึกและเล่นเสียงนั้น ผู้จัดทำปริญญานิพนธ์นำวงจรบันทึกและเล่นเสียงมาใช้งานเพื่อแจ้งให้ผู้ใช้บริการทราบว่าได้ใช้งานเครื่องที่เท่าไหร่ เมื่อผู้ใช้บริการกดรหัสผ่านและหยอดเหรียญถูกต้อง จะมีสัญญาณเสียงบอกหมายเลขเครื่อง 1, 2, 3 หรือ 4 โดยการเชื่อมต่อกับไมโครคอนโทรลเลอร์ ARM7 เพื่อทำการควบคุมการเล่นเสียงนั้น จะต่อไมโครคอนโทรลเลอร์เข้ากับขา PLAY เพื่อสั่งให้เล่นเสียง FWD เพื่อสั่งให้เล่นเสียงถัดไป และ RESET เพื่อเริ่มต้นการทำงานใหม่ (ย้อนกลับไปที่เสียงแรก)

### 3.1.1.4 ไมโครคอนโทรลเลอร์

ในการทำปริญญานิพนธ์นี้ใช้ไมโครคอนโทรลเลอร์จำนวน 2 บอร์ด โดยบอร์ดหลักจะใช้บอร์ดสำเร็จรูป ARM7 LPC-2368 จำนวน 1 บอร์ด ซึ่งจะเป็นส่วนกลางในการรับค่าอินพุต และส่งค่าเอาต์พุต ได้แก่ รับสัญญาณจากเครื่องหยอดเหรียญ, แสดงผลผ่านจอแอลซีดี, รับค่าอินพุตจากวงจรตรวจสอบสถานะเครื่องซักผ้า (วงจรแอลดีอาร์และวงจรไดโอด) ส่งเอาต์พุตเพื่อเปิดปิดเครื่องซักผ้า และเชื่อมต่อกับฐานข้อมูลด้วยพอร์ตอินเตอร์เน็ต แสดงพอร์ตต่างๆของไมโครคอนโทรลเลอร์ ดังรูปที่ 3.5 และบอร์ดรองใช้ ATMEGA168 จำนวน 1 บอร์ด ดังรูปที่ 3.6 เพื่อรับค่ารหัสผ่านจากผู้ใช้บริการ และส่งรหัสผ่านไปยังพอร์ต UART-2 ของไมโครคอนโทรลเลอร์หลักเพื่อทำการตรวจสอบต่อไป



รูปที่ 3.5 ไมโครคอนโทรลเลอร์หลัก ARM7 LPC-2368



รูปที่ 3.6 ไมโครคอนโทรลเลอร์รอง ATMEGA168

### 3.1.1.5 ภาคแสดงผล

สำหรับการเชื่อมต่อนั้นจะสามารถใช้ได้กับแอลซีดีแบบ Character Dot-Matrix เท่านั้น โดยเชื่อมต่อแบบ 4 บิต Data โดยสัญญาณที่ใช้เชื่อมต่อกับแอลซีดี จะเป็นสัญญาณจาก P1[24...29] และ P1[31] จำนวน 7 บิต โดยในการเชื่อมต่อสายสัญญาณจากขั้วต่อของพอร์ตแอลซีดีไปยังจอแสดงผลแอลซีดีนั้น ให้ยึดชื่อสัญญาณเป็นจุดอ้างอิง โดยให้ต่อสัญญาณที่มีชื่อตรงกันเข้าด้วยกันให้ครบทั้ง 14 เส้น (ในปริญญานิพนธ์นี้ไม่ได้ใช้ขาที่ 15 และ 16) ดังตารางที่ 3.1

ตารางที่ 3.1 ตำแหน่งขาใช้งานจอแสดงผลแอลซีดี

| ขาที่ | สัญญาณ | รายละเอียด |
|---|---|---|
| 1 | GND | ขั้วต่อแรงดัน 0 โวลต์ |
| 2 | VCC | ขั้วต่อแรงดัน 5 โวลต์ |
| 3 | VO | ขั้วต่อแรงดันสำหรับปรับค่าความสว่างของจอแสดงผล ใช้แรงดันตั้งแต่ 0 ถึง 5 โวลต์ (ถ้าขั้วนี้ต่อลงกราวด์จะมีความสว่างสูงสุด) |
| 4 | RS | RS (Register Select) สัญญาณเลือกการทำงานระหว่างรีจิสเตอร์ควบคุมกับหน่วยความจำของจอแสดงผลแอลซีดี |
| 5 | R/W | R/W (Register read or write) สัญญาณควบคุมการอ่านหรือเขียนข้อมูล |
| 6 | EN | EN (Enable) สัญญาณควบคุมการเริ่มทำงานของจอแสดงผล |
| 7 | D0 | บิตข้อมูล Data 0 |
| 8 | D1 | บิตข้อมูล Data 1 |
| 9 | D2 | บิตข้อมูล Data 2 |
| 10 | D3 | บิตข้อมูล Data 3 |

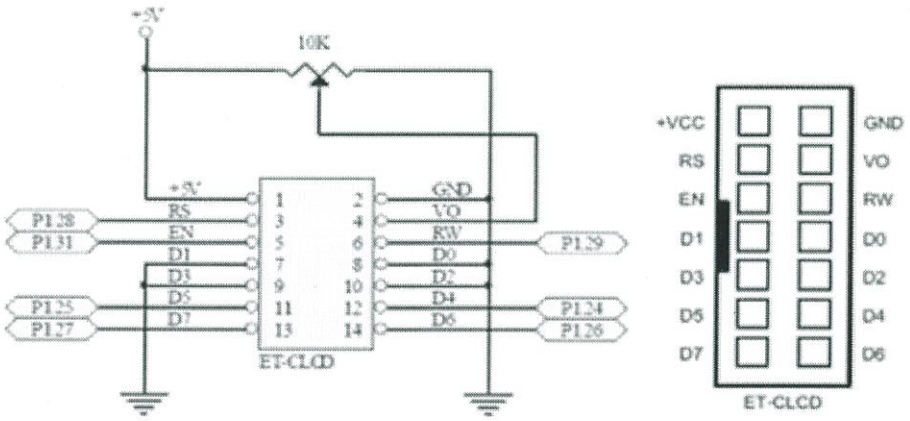
ตารางที่ 3.1 ตำแหน่งขาใช้งานจอแสดงผลแอลซีดี(ต่อ)

| ขาที่ | สัญญาณ | รายละเอียด |
|---|---|---|
| 11 | D4 | บิตข้อมูล Data 4 |
| 12 | D5 | บิตข้อมูล Data 5 |
| 13 | D6 | บิตข้อมูล Data 6 |
| 14 | D7 | บิตข้อมูล Data 7 |
| 15 | A | ขั้วต่อแรงดัน 5 โวลต์ ของหลอดไฟส่องสว่างด้านหลังจอแสดงผล |
| 16 | K | ขั้วต่อแรงดัน 0 โวลต์ ของหลอดไฟส่องสว่างด้านหลังจอแสดงผล |

สัญญาณการเชื่อมต่อกับแอลซีดีให้ต่อกับพอร์ตต่างๆ ดังรูปที่ 3.7

DB4 → P1 [24] RW → P1 [29] DB6 → P1 [26] EN → P1 [31]

DB5 → P1 [25] RW → P1 [29] DB7 → P1 [28]




รูปที่ 3.7 การเชื่อมต่อสายสัญญาณจากไมโครคอนโทรลเลอร์หลักไปยังจอแสดงผลแอลซีดี

### 3.1.1.6 วงจรสวิตช์ (Keypad) สำหรับกรอกรหัสผ่าน

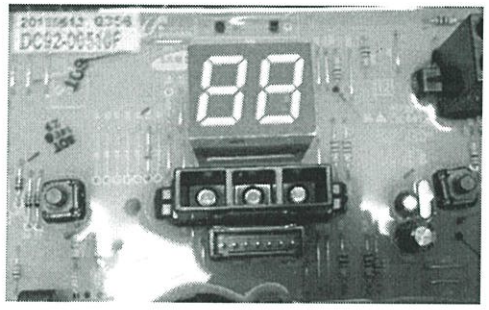
กรณีผู้ใช้งานทำการจองเครื่องผ่านอินเตอร์เน็ต และในขณะเดียวกันก็มีผู้ใช้ท่านอื่นที่ไม่ได้ทำการจองมาใช้เครื่องซักผ้า ผลคือ ผู้ใช้ท่านอื่นที่ไม่ได้ทำการจองจะสามารถใช้เครื่องซักผ้าได้ ซึ่งจะเป็นการลัดคิวของผู้ใช้ที่ทำการจองเอาไว้ ผู้จัดทำจึงได้ออกแบบระบบให้มีการกรอกรหัสผ่านสำหรับการใช้งาน โดยที่เมื่อเข้าสู่ระบบการจองและกรอกข้อมูลเรียบร้อย ผู้จองจะได้รับรหัสผ่าน 4 ตัว เพื่อนำไปกรอกที่หน้าเครื่องซักผ้า ผู้จัดทำได้ใช้วิธีการต่อแบบขาร่วม คือ จะต่อวงจรโดยนำหน้าสัมผัสของสวิตช์ทุกตัวมารวมเข้าด้วยกันเป็นขาร่วม สำหรับหน้าสัมผัสอีกด้านหนึ่งของสวิตช์แต่ละตัวก็จะต่อเข้ากับพอร์ตอินพุตของระบบโดยตรง

### 3.1.1.7 การตรวจสอบสถานะการทำงานของเครื่องซักผ้า

ผู้จัดทำใช้การตรวจสอบสถานะของเครื่องซักผ้า 2 แบบ ได้แก่ ใช้วงจรแอลดีอาร์รับแสง และวงจรไดโอดร่วมกับออปโต้คัปเปลอร์

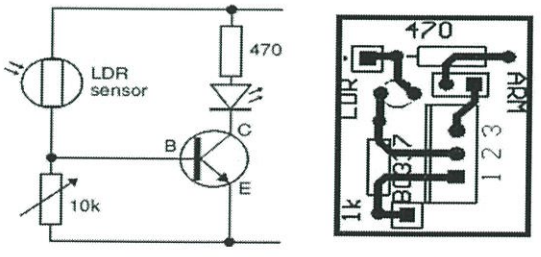
1. วงจรแอลดีอาร์รับแสง

จากการศึกษาโครงสร้างภายในของเครื่องซักผ้า ยี่ห้อซัมซุง 6.5 กิโลกรัม รุ่น WA85G5 พบว่า ภายในระบบควบคุมจะมีไฟแสดงสถานะ จำนวน 3 หลอด ได้แก่ สีเขียวจำนวน 2 หลอด และสีแดงจำนวน 1 หลอด ดังรูปที่ 3.8



รูปที่ 3.8 โครงสร้างภายในของระบบควบคุมเครื่องซักผ้า ยี่ห้อซัมซุง รุ่น WA85G5

จากการทดสอบเมื่อเริ่มให้เครื่องซักผ้าทำงานตามปกติจนสิ้นสุดกระบวนการซัก พบว่า ไฟสีเขียวด้านซ้าย จะติดเมื่อเครื่องซักผ้าทำงาน, ไฟสีแดง จะติดเมื่อเครื่องซักผ้าพร้อมทำงาน และไฟสีเขียวด้านขวา จะติดเมื่อเครื่องซักผ้าสิ้นสุดการทำงานและจะดับลง ผู้ทดลองจึงได้ทำการใช้วงจรแอลดีอาร์เพื่อเป็นตัวรับแสงไฟสีเขียวทางด้านขวาดังรูปที่ 3.9 (ก) และลายวงจรที่ออกแบบ ดังรูปที่ 3.9 (ข)




(ก) (ข)

รูปที่ 3.9 วงจรรับแสงโดยใช้แอลดีอาร์

(ก) วงจรรับแสง

(ข) ลายวงจรที่ออกแบบ

มีขั้นตอนการติดตั้งและออกแบบดังนี้

1) ด้านบนของเครื่องซักผ้า ยี่ห้อซัมซุง 6.5 กิโลกรัม รุ่น WA85G5 ดังรูปที่ 3.10

รูปที่ 3.10 ด้านบนของเครื่องซักผ้า ยี่ห้อซัมซุง รุ่น WA85G5

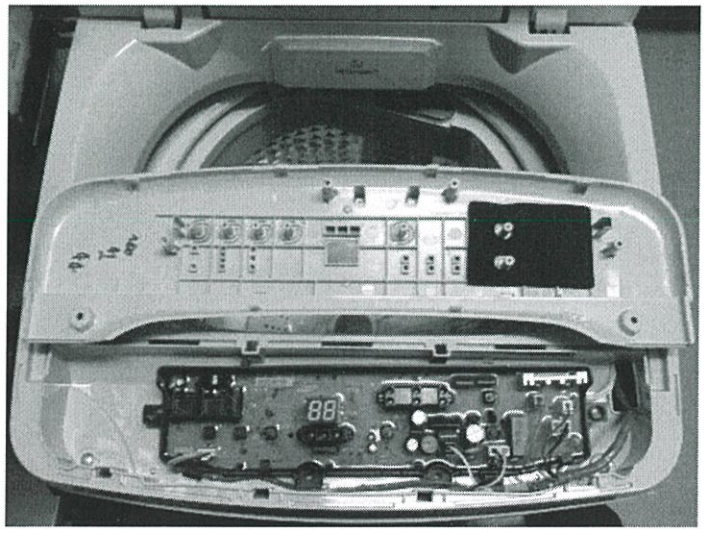
2) เปิดฝาบนของเครื่องซักผ้าออก ดังรูปที่ 3.11

รูปที่ 3.11 เปิดฝาบนของเครื่องซักผ้าออก

3) เปิดแผงควบคุมของเครื่องซักผ้า โดยใช้ไขควงไขน็อตทั้งสองฝั่งออก ดังรูปที่ 3.12

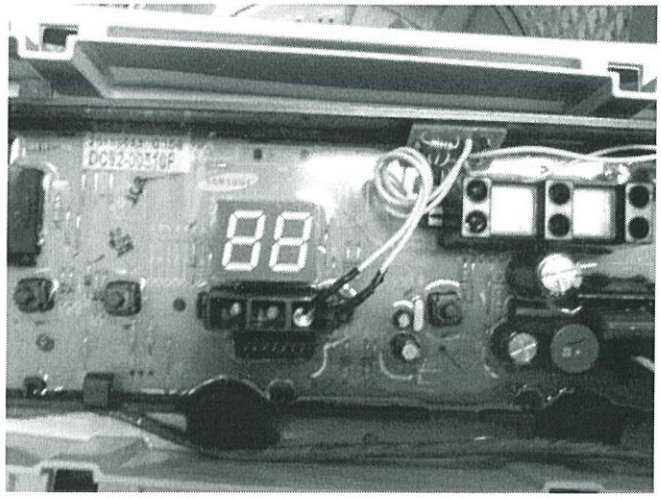
รูปที่ 3.12 เปิดแผงควบคุมเครื่องซักผ้า

4) หลังจากทำการไขน็อตทั้งสองตัวออกมาแล้ว เปิดฝาของแผงควบคุมออก จะเห็นแผงควบคุม ดังรูปที่ 3.13



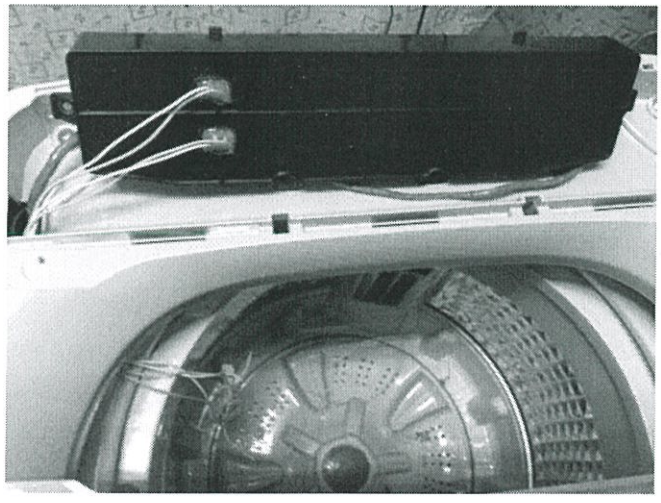
รูปที่ 3.13 เปิดแผงควบคุมเครื่องซักผ้า

5) นำวงจรตรวจสอบสถานะไปติดตั้งบริเวณหลอดไฟสีเขียวทางด้านขวา ดังรูปที่ 3.14



รูปที่ 3.14 การติดตั้งวงจรตรวจสอบสถานะ

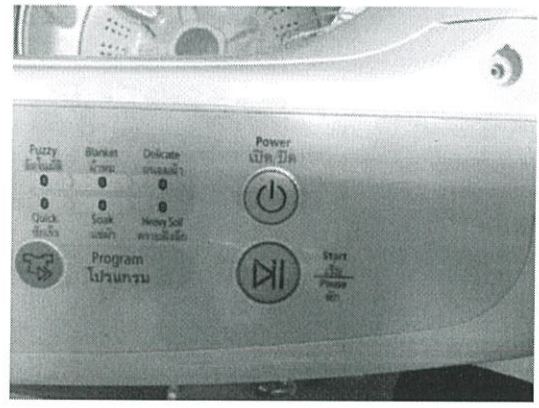
6) เดินสายไฟออกมาจากวงจร ระวังไม่ให้ชิ้นส่วนใดขวางฝาปิดแผงวงจรควบคุม จากรูปที่ 3.14 เดินสายสัญญาณ โดยทำการเจาะรูออกมาจากแผงควบคุม จากนั้นทำการติดตั้งกลับเข้าไป ไขน็อตให้แน่น และปิดฝาแผงควบคุม ดังรูปที่ 3.15



รูปที่ 3.15 เดินสายไฟออกมาจากแผงควบคุม

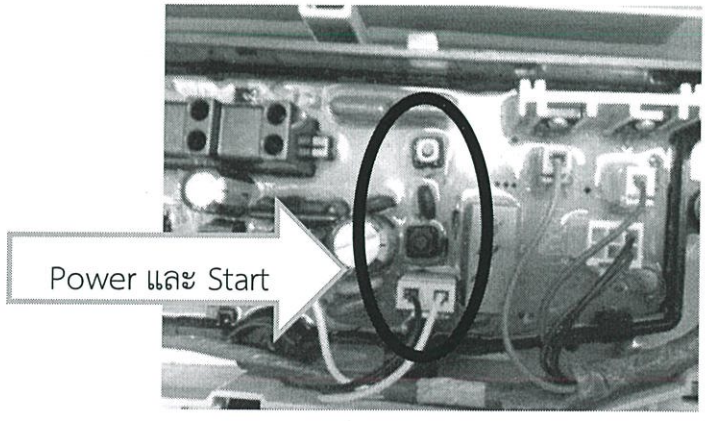
เนื่องจากระบบควบคุมที่ออกแบบ ต้องมีการสั่งเปิดปิดโดยใช้ไมโครคอนโทรลเลอร์ ดังนั้น ผู้ใช้งานจึงไม่สามารถกดปุ่มเลือกที่ด้านหน้าของเครื่องซักผ้าได้โดยตรง ผู้ออกแบบจึงได้ทำการนำสายสัญญาณไปติดตั้ง โดยมีขั้นตอนเช่นเดียวกับการติดตั้งวงจรตรวจสอบสถานะ แต่ให้ทำการเพิ่มสายสัญญาณไปที่สวิตช์ Power และ Start มีขั้นตอนดังนี้

ขั้นตอนที่ 1 สังเกตสวิตช์ Power และ Start โดยสวิตช์ Power จะเป็นสวิตช์กดสำหรับเปิด/ปิดเครื่องซักผ้า กดหนึ่งครั้งเพื่อทำการเปิด และกดอีกหนึ่งครั้งเพื่อทำการปิด และสวิตช์ Start กดหนึ่งครั้งเพื่อให้เครื่องซักผ้าเริ่มทำงาน ดังรูปที่ 3.16



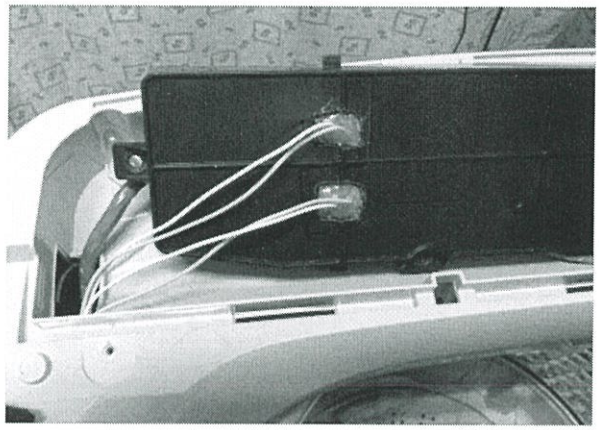

รูปที่ 3.16 สวิตช์ Power และ Start

ขั้นตอนที่ 2 เปิดแผงวงจรควบคุมออก จะเห็นสวิตช์กดสองอัน โดยสวิตช์ Power จะอยู่ด้านบน และสวิตช์ Start จะอยู่ด้านล่าง ดังรูปที่ 3.17



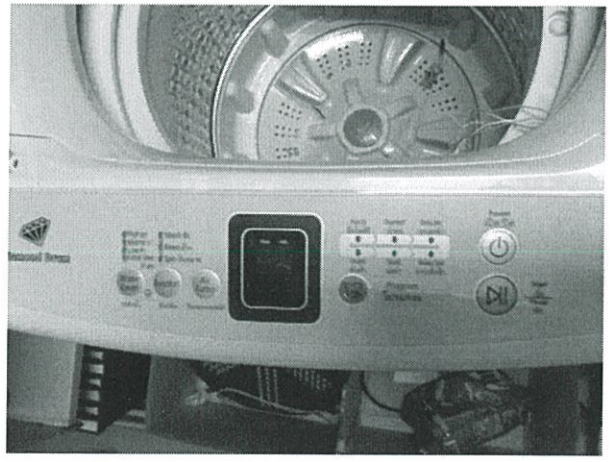

รูปที่ 3.17 สวิตช์ Power และ Switch บนแผงควบคุม

ขั้นตอนที่ 3 ทำการติดตั้งสายสัญญาณโดยเจาะพลาสติกที่ด้านล่างของแผงควบคุม ระมัดระวังการเจาะให้ตรงกับสวิตช์ทั้งสองตัว จะเห็นแผงวงจร นำสายสัญญาณบัดกรีเข้ากับแผงวงจร โดยให้สายสีขาวเป็นกราวด์ และสายสีเหลืองเป็นสายสัญญาณ ทำเช่นเดียวกันทั้งสองสวิตช์ ดังรูปที่ 3.18



รูปที่ 3.18 การติดตั้งสายสัญญาณ

ขั้นตอนที่ 4 ติดตั้งอุปกรณ์ทั้งหมดกลับเข้าไป ไขน็อตให้แน่น ปิดฝาแผงควบคุม ดังรูปที่ 3.19

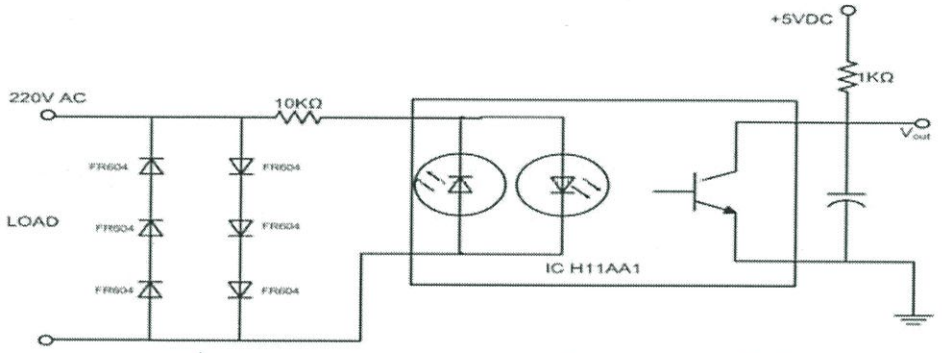


รูปที่ 3.19 ปิดฝาแผงวงจร

ขั้นตอนที่ 5 ทำการตรวจสอบการทำงานของเครื่องซักผ้า โดยนำสายสัญญาณที่ติดตั้งไว้สองเส้นสัมผัสกัน (แทนการกดสวิตช์) ทำเช่นเดียวกันทั้งสองเส้น ถ้าเครื่องซักผ้าทำงานจะมีเสียงดัง “ติ๊ด” สามครั้ง ถ้ายังไม่มีเสียง ให้ทำการตรวจสอบว่าบัดกรีสายสัญญาณติดแน่นหรือไม่

2. วงจรไดโอดร่วมกับออปโต้คัปเปลอร์

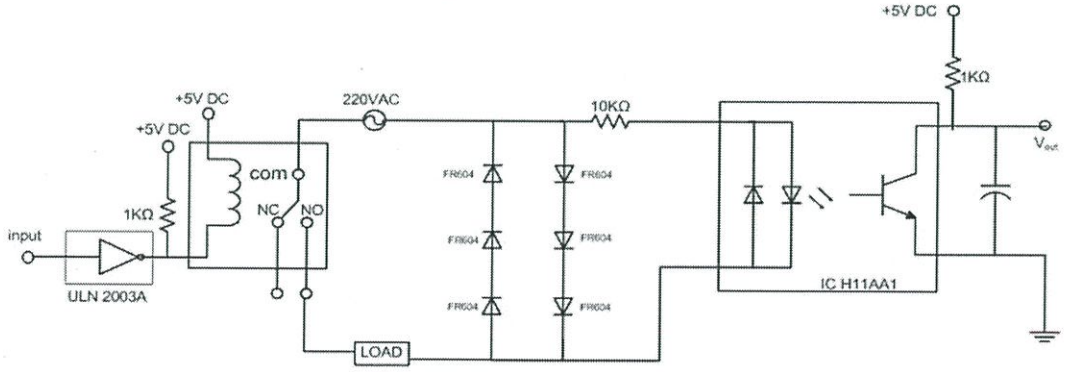
การตรวจสอบสถานะเครื่องซักผ้าแบบนี้ จะใช้ระดับแรงดันเอาต์พุตจากออปโต้คัปเปลอร์เป็นสัญญาณอินพุตป้อนให้กับไมโครคอนโทรลเลอร์หลัก เมื่ออุปกรณ์ไฟฟ้าทำงาน แรงดันจะตกคร่อมไดโอดทั้ง 6 ตัว ออปโต้คัปเปลอร์จะทำงานเมื่อมีแรงดันไฟฟ้าผ่านไดโอด ดังรูปที่ 3.20




รูปที่ 3.20 วงจรตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า

### 3.1.1.8 การควบคุมและการตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า

วงจรควบคุมและตรวจสอบสถานะของอุปกรณ์ไฟฟ้า ดังรูปที่ 3.21 จุด A เป็นอินพุตจากไมโครคอนโทรลเลอร์หลัก เพื่อเป็นสัญญาณในการควบคุมการทำงานของอุปกรณ์ไฟฟ้า แต่เนื่องจากสัญญาณจากบอร์ดไม่เพียงพอที่จะขับรีเลย์ได้จึงต้องทำการเชื่อมต่ออุปกรณ์ ULN2003 เพื่อทำงานเพิ่มกระแสให้กับอินพุตก่อนนำไปขับรีเลย์ต่อไป เมื่อรีเลย์ทำงานจะมีแรงดันไฟฟ้า 220 โวลต์ผ่านไดโอด ขณะที่ไม่มีแรงดันไฟฟ้าผ่านไดโอด ออปโต้คัปเปลอร์จะไม่ทำงาน ทำให้มีแรงดันเอาต์พุต 5 โวลต์ ที่จุด C เมื่อมีแรงดันไฟฟ้าผ่านไดโอด ออปโต้คัปเปลอร์จะทำงาน ทำให้มีแรงดันเอาต์พุต 0 โวลต์ ที่จุด C โดยแรงดันเอาต์พุตที่ได้จะถูกนำไปใช้ในการตรวจสอบสถานะของเครื่องซักผ้าต่อไป

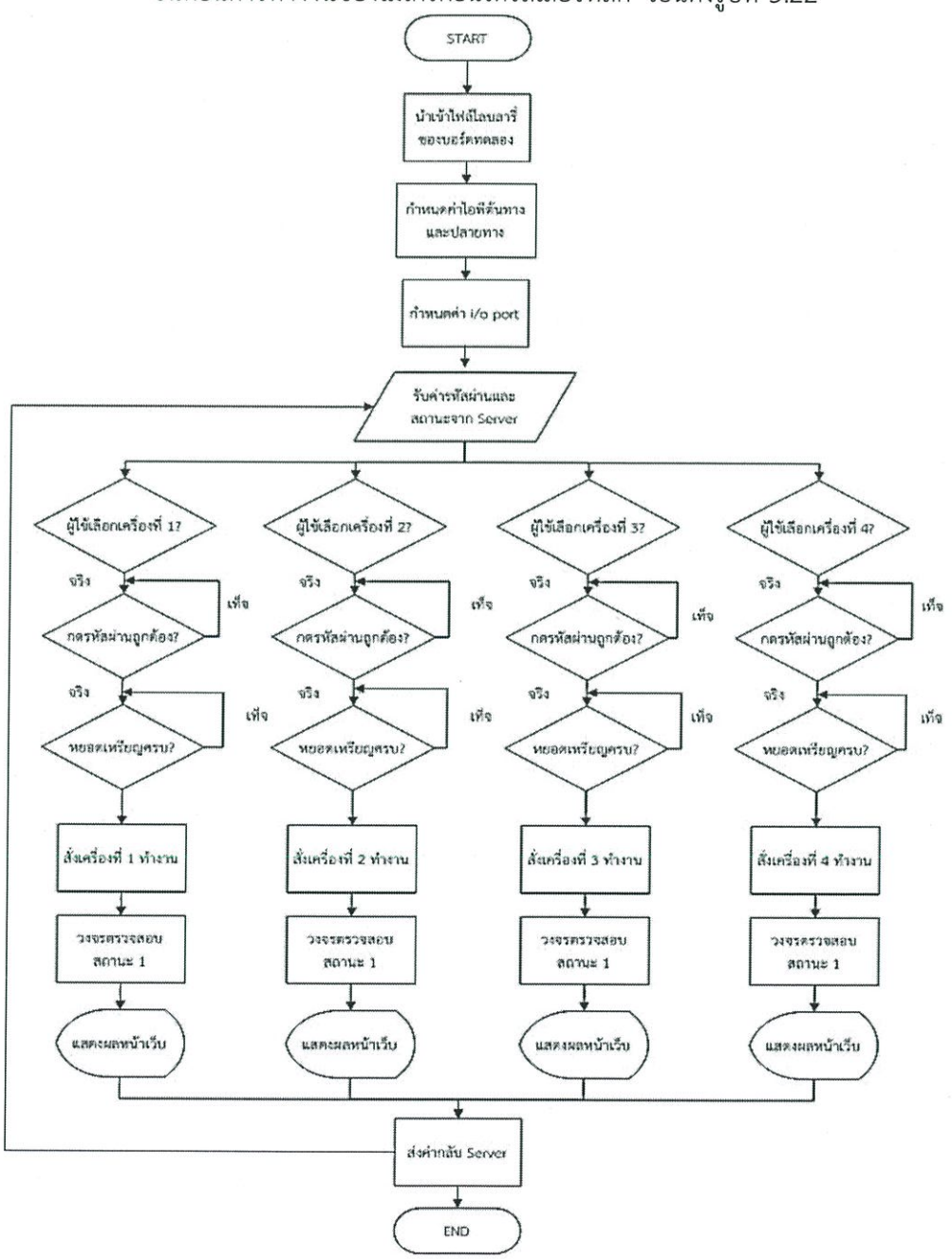



รูปที่ 3.21 วงจรควบคุมและตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า

### 3.1.2 การออกแบบซอฟต์แวร์

#### 3.1.2.1 ไมโครคอนโทรลเลอร์หลัก

ขั้นตอนการทำงานของไมโครคอนโทรลเลอร์หลัก เป็นดังรูปที่ 3.22

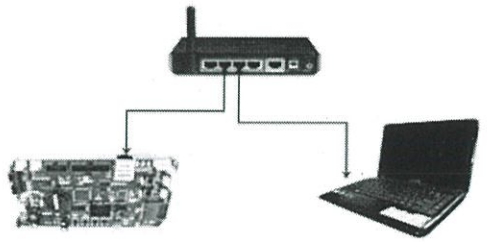



รูปที่ 3.22 โฟลว์ชาร์ตแสดงขั้นตอนการทำงานของไมโครคอนโทรลเลอร์หลัก

จากโฟลว์ชาร์ตรูปที่ 3.22 อธิบายได้ว่า หน้าที่ของไมโครคอนโทรลเลอร์หลัก (ARM7) จะมีหน้าที่การติดต่อกับฐานข้อมูล โดยจะแบ่งการทำงานออกเป็น ขั้นตอนแรก จะทำการขอการเชื่อมต่อ จากนั้นจะขออัพเดตสถานะการทำงาน กรณีทุกเครื่องไม่ว่าง จะมีการขออัพเดตสถานะการทำงานอีกครั้ง เมื่อเครื่องว่าง จึงทำการเลือก จากนั้นจะเข้าสู่ขั้นตอนการยืนยันตนเอง โดยผู้ใช้งานจะได้รับรหัสผ่านจากหน้าเว็บเพื่อมานำมากดที่หน้าเครื่อง เมื่อกดรหัสผ่านถูกต้อง ผู้ใช้ต้องทำการหยอดเหรียญจนครบตามจำนวน และขั้นตอนสุดท้าย คือการเปิดเครื่องซักผ้าหยอดเหรียญ ซึ่งจะได้อธิบายรายละเอียดในหัวข้อต่อไป

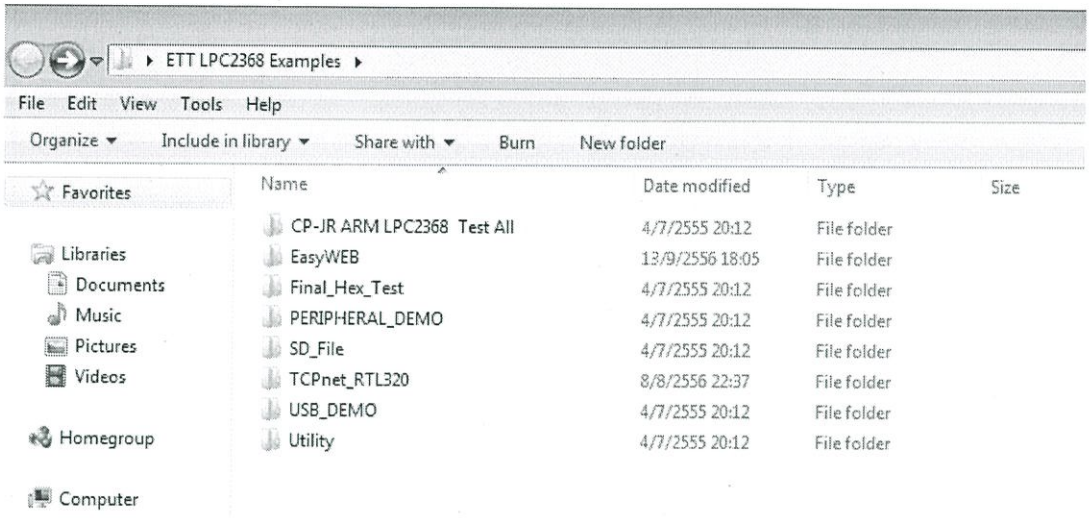
1) โปรแกรมคอมไพลเลอร์

ผู้จัดทำได้ออกแบบการเชื่อมต่อกับเครือข่าย ระหว่างบอร์ด CP-JR ARM7 LPC2368 นั้น จะใช้ขั้วต่อมาตรฐาน Ethernet แบบ RJ45 โดยวงจรส่วนนี้จะใช้ขาสัญญาณ P1[0,1,4,8,9,10,14..17] ดังรูปที่ 3.23 ซึ่งแสดงการเชื่อมต่อพอร์ตอินเตอร์เน็ตของไมโครคอนโทรลเลอร์และคอมพิวเตอร์กับเราท์เตอร์



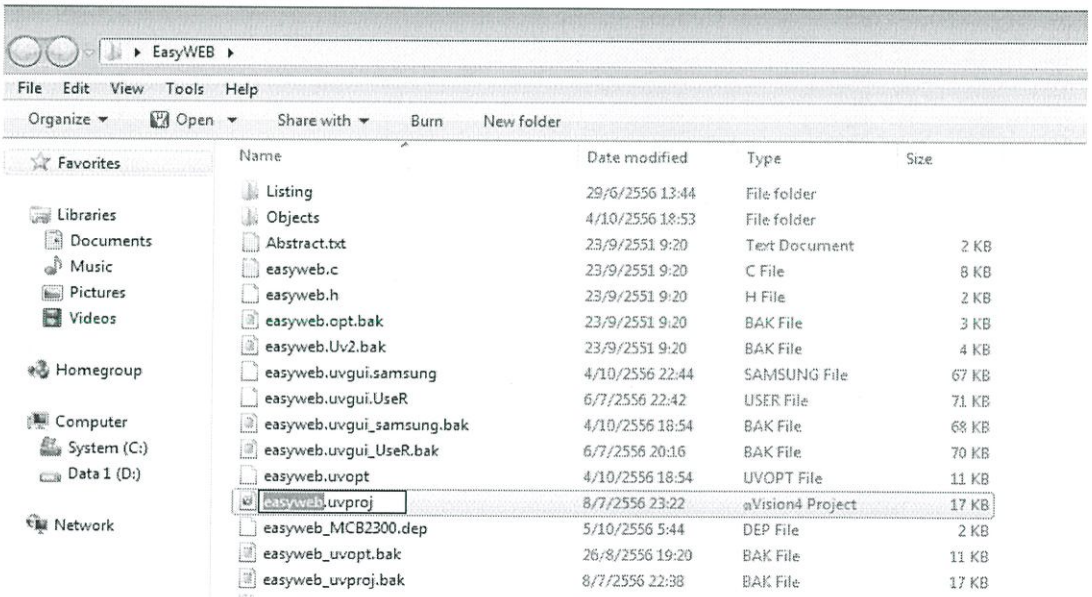
รูปที่ 3.23 การเชื่อมต่อบอร์ดทดลองกับไมโครคอนโทรลเลอร์

การออกแบบหน้าเว็บเพจ และการควบคุมการเชื่อมต่อระหว่างคอมพิวเตอร์และไมโครคอนโทรลเลอร์ ผู้จัดทำได้ออกแบบตามโครงสร้างพื้นฐานของ EasyWEB example ของบริษัท Embedded Internet issue of the German magazine Design&Electronic ดังรูปที่ 3.24



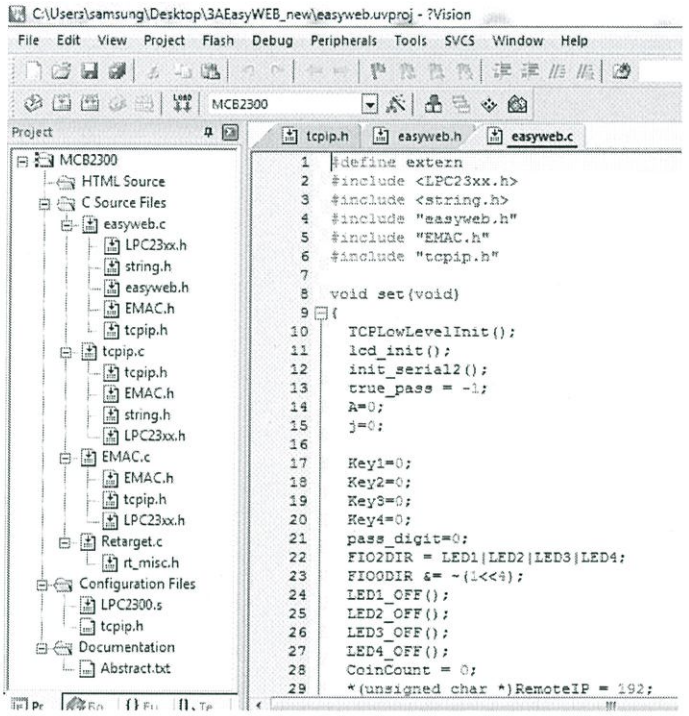

รูปที่ 3.24 ตัวอย่างโปรแกรม EasyWeb

เข้าไปในโฟลเดอร์ ดับเบิ้ลคลิกที่ไฟล์ easyweb.uvprog ดังรูปที่ 3.25




รูปที่ 3.25 ไฟล์ของ Easyweb

เมื่อเข้ามาในหน้าของคอมไพลเลอร์แล้วจะพบซอร์สโค้ดที่ใช้ในการออกแบบ ดังรูปที่ 3.26

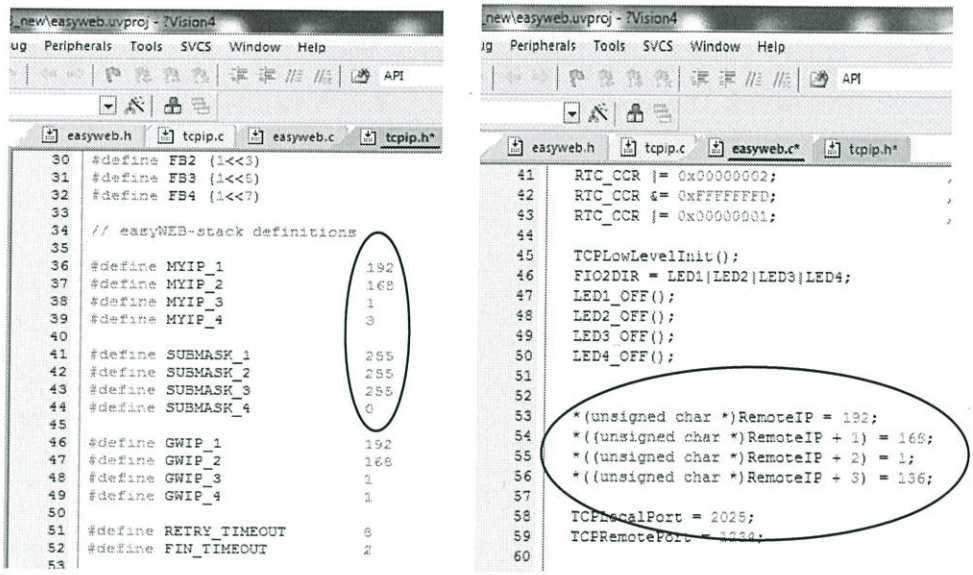



รูปที่ 3.26 โค้ดตัวอย่างที่ใช้ในการออกแบบ

จากโค้ดตัวอย่างดังรูป 3.26 มีโครงสร้างพื้นฐานที่จำเป็นต้องใช้ในการออกแบบ 3 ไฟล์ ได้แก่ tcpip.h ใช้สำหรับกำหนดพารามิเตอร์ในการติดต่อสื่อสาร, easyweb.h ใช้สำหรับเก็บตัวแปร Globol และ easyweb.c ใช้สำหรับเขียนคำสั่งควบคุม

2) การกำหนดพารามิเตอร์ในการรับส่งข้อมูลสำหรับไมโครคอนโทรลเลอร์

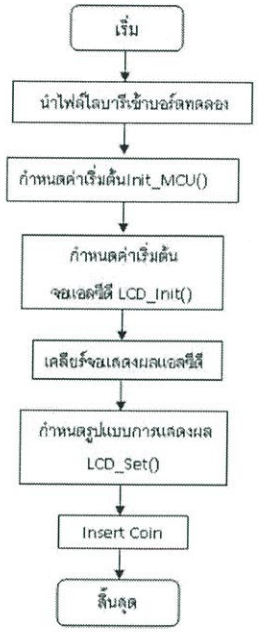
กำหนดไอพีแอดเดรสต้นทาง (ไมโครคอนโทรลเลอร์) เป็น 192.168.1.3 และไอพีแอดเดรสปลายทาง (เซิร์ฟเวอร์) เป็น 192.168.1.136 และตั้งค่าหมายเลขพอร์ตเป็น 1234 ในไฟล์ tcpip.h ดังรูปที่ 3.27




รูปที่ 3.27 กำหนดไอพีแอดเดรสต้นทางและปลายทาง

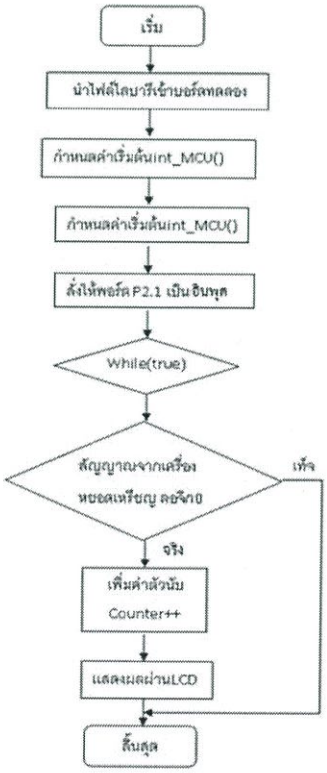
3) โปรแกรมย่อยการแสดงผลจำนวนเหรียญผ่านแอลซีดี

ในการทดลอง ได้ออกแบบให้มีหน้าจอแอลซีดีแสดงจำนวนเหรียญที่ผู้ใช้บริการหยอด และเมื่อทำการหยอดจนครบตามจำนวนจึงจะสามารถใช้งานเครื่องซักผ้าได้ การต่อวงจรควบคุมการแสดงผลด้วยจอแอลซีดี จะต่อเข้ากับพอร์ต P1.24 ถึง P.30 ของบอร์ดไมโครคอนโทรลเลอร์หลัก จากนั้นทำการเขียนโปรแกรมสั่งงานให้แสดงผลเป็นตัวอักษร และตัวเลข โดยจะรับค่าสัญญาณจากเครื่องหยอดเหรียญที่พอร์ต 4.29 ทุกๆครั้งที่ทำการหยอดเหรียญ จะมีการเปลี่ยนแปลงของระดับสัญญาณจากสายเอาต์พุต (สีขาว) จากเครื่องหยอดเหรียญ ตามที่ได้แสดงสายสัญญาณไว้ จากนั้นจะทำการเพิ่มค่าตัวแปรทีละ 10 และแสดงผลผ่านจอแอลซีดี มีโฟลว์ชาร์ตและโค้ดโปรแกรมการทดลองดังรูปที่ 3.28 และมีโฟลว์ชาร์ตการทำงานการตรวจสอบสัญญาณเครื่องหยอดเหรียญ ดังรูปที่ 3.29




```
Text1*
1  #include <stdio.h>
2  #include "LPC23xx.H"
3  #include "Smart_Function.h"
4
5  int main(void)
6  {
7      init_MCU();
8
9      LCD_Init();
10     LCD_ClrScr();
11     LCD_Set(SCR_ON, CUR_NOBLINK, CUR_HIDE);
12
13     LCD_WriteStr(2,1 "INSERT COIN");
14     LCD_WriteStr(4,2 "BATH");
15
16     while(TRUE);
17 }
```

รูปที่ 3.28 โฟลว์ชาร์ตและโค้ดโปรแกรมการแสดงผลผ่านจอแอลซีดี




```
Text1*
1  #include <stdio.h>
2  #include "LPC23xx.H"
3  int main(void)
4  {
5      int A,
6      unsigned char str_temp[20];
7      A=0,
8      {
9       lcd_init();
10      while(1)
11         {
12         if((FIO2PIN & (2 << 1)) == 0)
13         {
14         A+=10
15         sprintf(str_temp,"%d",A);
16         goto_cursor(0x44);
17         lcd_print (str_temp);
18         goto_cursor(0x02);
19
20         lcd_print("INSERT COIN");
21         goto_cursor(0x48);
22         lcd_print("BATH");
23
24         delay(8000000);
25         }
26         }
27     }
28 }
```

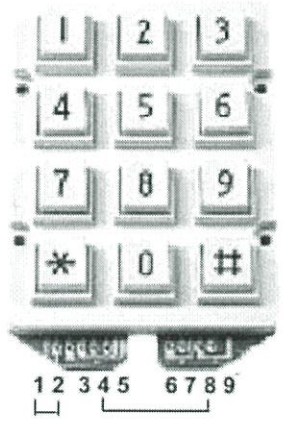
รูปที่ 3.29 โฟลว์ชาร์ตและโค้ดโปรแกรมการทำงานการตรวจสอบสัญญาณเครื่องหยอดเหรียญ

4) โปรแกรมย่อยการส่งและรับค่ารหัสผ่านด้วยคีย์แพด

ผู้จัดทำได้ออกแบบสวิตช์คีย์แพดขนาด 4x3 โดยต้องเชื่อมต่อขาที่ 1 และ 2, 4 และ 8 เข้าด้วยกันก่อน ดังรูปที่ 3.30 จากนั้นต่อเข้ากับไมโครคอนโทรลเลอร์ ATMEGA168 โดยจะเชื่อมต่อ Digital Pin7 ถึง Digital Pin13 โดยแสดงได้ดังตารางที่ 3.2

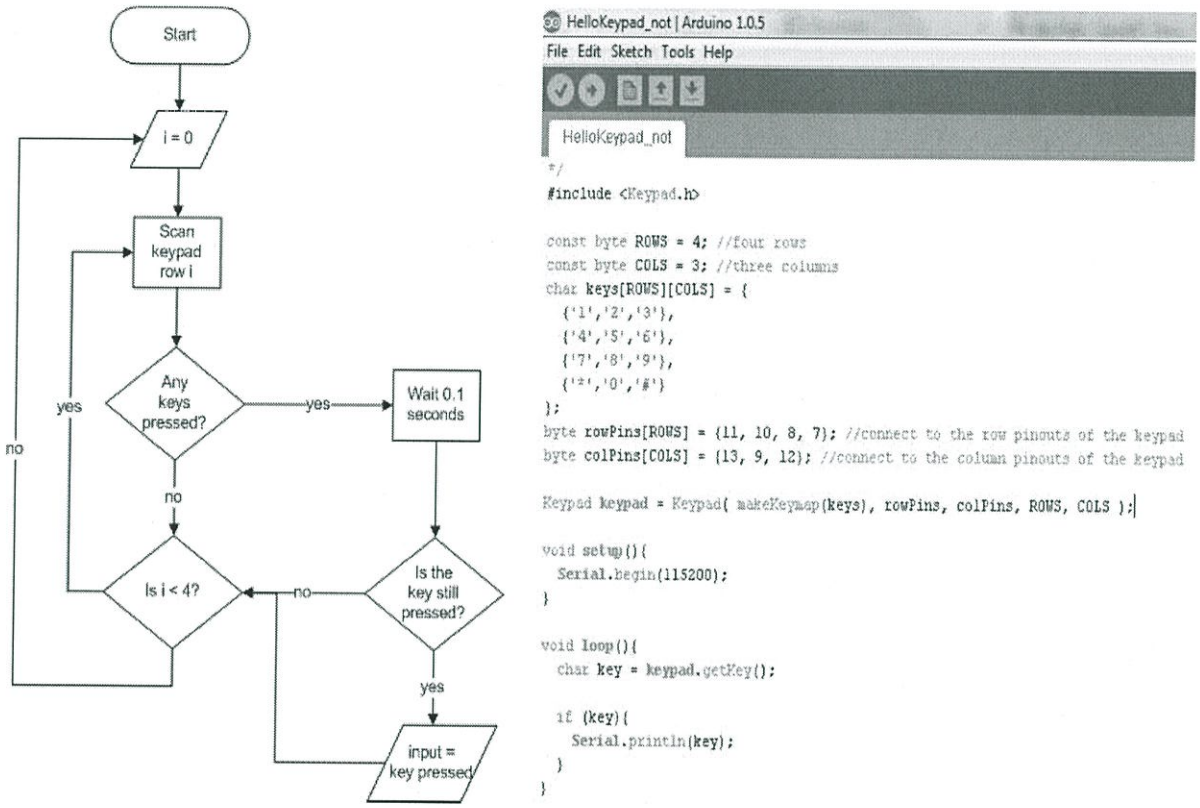
ตารางที่ 3.2 การเชื่อมต่อขาต่างๆระหว่างไมโครคอนโทรลเลอร์และคีย์แพด

| Digital Pin | ขาของไมโครคอนโทรลเลอร์ | ขาของคีย์แพด |
|---|---|---|
| 7 | 13 | 1,2 |
| 8 | 14 | 3 |
| 9 | 15 | 4,8 |
| 10 | 16 | 5 |
| 11 | 17 | 6 |
| 12 | 18 | 7 |
| 13 | 19 | 9 |



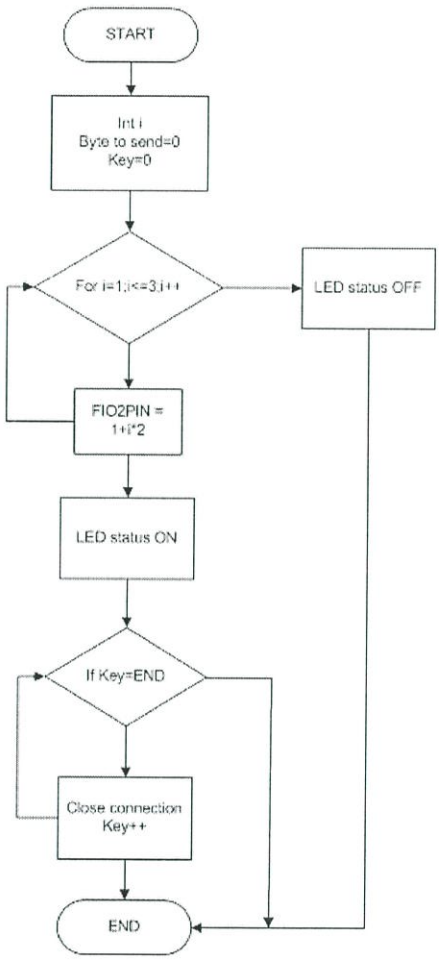

รูปที่ 3.30 ขาใช้งานของคีย์แพด

จากนั้นเขียนชุดคำสั่งรับและส่งค่าระหว่างไมโครคอนโทรลเลอร์หลักและไมโครคอนโทรลเลอร์รอง โดยมีโฟลว์ชาร์ตและโค้ดโปรแกรม ดังรูปที่ 3.31




รูปที่ 3.31 โฟลว์ชาร์ตและโค้ดโปรแกรมการตรวจสอบรหัสผ่าน

5) โปรแกรมย่อยการส่งค่าสถานะเครื่องซักผ้าไปยังเซิร์ฟเวอร์ ดังรูปที่ 3.32




easyweb.h easyweb.c

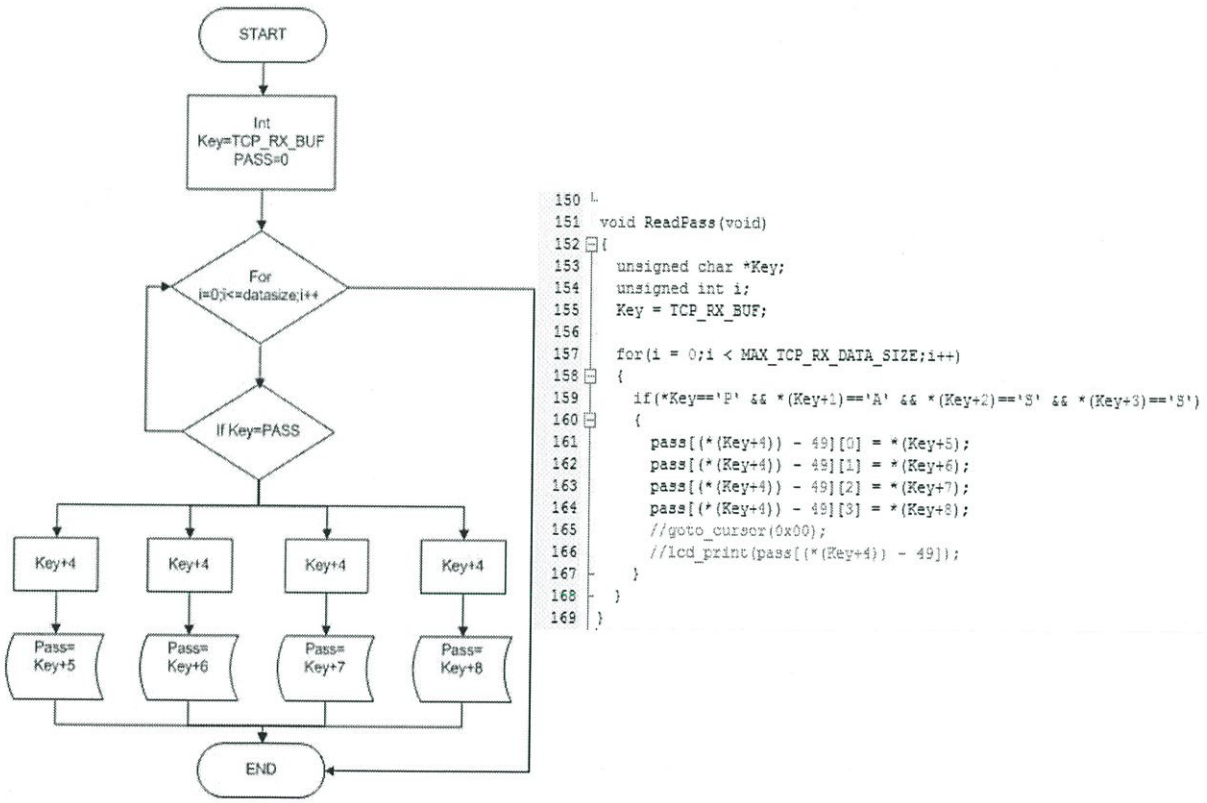
```
106 }
107
108  void Send(void)
109 {
110   unsigned int ByteToSend=0;
111   int i;
112   TCPReleaseRxBuffer();
113
114   for(i = 0;i <= 3;i++)
115   {
116     if(FIO2PIN & (1<<(1 + i * 2)))
117     {
118       memcpy(TCP_TX_BUF+ByteToSend, LEDstatus_off[i] , 10);
119     }
120     else
121     {
122       memcpy(TCP_TX_BUF+ByteToSend, LEDstatus_on[i] , 10);
123     }
124     ByteToSend+=10;
125   }
126
127   TCPTxDataCount = ByteToSend;
128   TCPTransmitTxBuffer();
129   CheckReciveFinished = ReciveNotFinished;
130
131   }
132
133  void CheckEndReceive(void)
134 {
135   unsigned char *Key;
136   unsigned int i;
137   Key = TCP_RX_BUF;
138   for(i = 0;i < MAX_TCP_RX_DATA_SIZE;i++)
139   {
140     if(*Key == 'E' && *(Key+1) == 'N' && *(Key+2) == 'D')
141     {
142       TCPClose();
143       ReadPass();
144       CheckReciveFinished = ReciveFinished;
145       return;
146     }
147   Key++;
148   }
149 }
```

รูปที่ 3.32 โฟลว์ชาร์ตและโค้ดโปรแกรมการส่งค่าสถานะเครื่องซักผ้าไปยังเซิร์ฟเวอร์

จากรูปที่ 3.32 อธิบายได้ว่า สถานะของเครื่องซักผ้าจะเก็บไว้ในตัวแปร ByteToSend (เริ่มต้นจะเป็นค่า 0 หมายถึงยังไม่มีค่าเก็บไว้) จากนั้นจะใช้ลูป for ในการวนอ่านค่าแต่ละพอร์ตด้วยตัวแปร i โดยทำการเลื่อนบิตข้อมูลไปทีละ 1 + i * 2 หมายความว่า ถ้า i = 0 จะต้องอ่านพอร์ตที่ 1 + 0 * 2 = 1 , ถ้า i = 1 จะต้องอ่านพอร์ตที่ 1 + 1 * 2 = 3 ,ถ้า i = 2 จะต้องอ่านพอร์ตที่ 1 + 2 * 2 = 5 และถ้า i = 3 จะต้องอ่านพอร์ตที่ 1 + 3 * 2 = 7 นั่นคือพอร์ตที่ 1,3,5 และ 7 ซึ่งตั้งค่าไว้ให้เป็นพอร์ตตรวจสอบสถานะเท่านั้น จากนั้นทำการส่งค่าออกไปโดยใช้คำสั่ง TCPTransmiTxBuffer ในการตรวจสอบว่าข้อมูลมีการส่งไปจริงหรือไม่ จะให้เซิร์ฟเวอร์ส่งคำว่า END กลับมาเมื่อได้รับข้อมูลแล้ว จากนั้นเขียนคำสั่งให้ไมโครคอนโทรลเลอร์ตรวจสอบเพื่อดำเนินการต่อไป

6) โปรแกรมย่อยการรับค่ารหัสผ่านจากเซิร์ฟเวอร์

เมื่อเซิร์ฟเวอร์ทราบสถานะของเครื่องซักผ้าแต่ละเครื่องแล้ว จะนำไปแสดงผลให้ผู้ใช้บริการทราบว่าเครื่องซักผ้าเครื่องใดทำงานอยู่บ้าง เมื่อผู้ใช้บริการจองเครื่องซักผ้า เซิร์ฟเวอร์จะทำการแจ้งรหัสการใช้เครื่องซักผ้าเพื่อให้ผู้ใช้บริการนำมายืนยันตัวตน ซึ่งรหัสดังกล่าวจะต้องมีการส่งมายังไมโครคอนโทรลเลอร์ด้วยเพื่อให้สามารถตรวจสอบได้ว่ารหัสผ่านที่ผู้ใช้บริการกดตรงกับรหัสที่รับมาหรือไม่ ถ้าตรงกันจึงจะสามารถใช้งานเครื่องซักผ้าได้ มีโค้ดโปรแกรมรับค่ารหัสผ่านและโฟลว์ชาร์ต ดังรูปที่ 3.33




รูปที่ 3.33 โฟลว์ชาร์ตและโค้ดโปรแกรมการรับค่ารหัสผ่านจากเซิร์ฟเวอร์

จากโค้ดโปรแกรมอธิบายได้ว่า รหัสผ่านที่ได้รับมาจะเก็บไว้ในตัวแปร pass และใช้ลูป for ในการวนรับค่ารหัสผ่านจนครบทั้ง 4 ตัว โดยเซิร์ฟเวอร์จะทำการส่งคำว่า PASS นำหน้ารหัสผ่านมาเพื่อให้การตรวจสอบง่ายขึ้น (ถ้าตรวจพบตัวอักษร PASS ครบทุกตัวหมายความว่านั่นคือรหัสผ่าน) จากนั้นทำการเก็บค่ารหัสผ่านหลังตัวแปร PASS คืออักขระตัวที่ 5, 6, 7 และ 8 (Key+5, Key+6, Key+7 และ Key+8) เพื่อนำไปตรวจสอบในขั้นตอนต่อไป

7) การตรวจสอบรหัสผ่านจากผู้ใช้และเซิร์ฟเวอร์

เมื่อผู้ใช้บริการกดรหัสผ่านจากคีย์แพดดังได้กล่าวไว้แล้วในหัวข้อที่ 4) ไมโครคอนโทรลเลอร์หลักจะรับค่ารหัสผ่านที่ผู้ใช้กดเข้าทางพอร์ต UART-2 ว่าเป็นตัวเลขใดเก็บไว้ใน uart_data โดยใช้เงื่อนไข if ในการตรวจสอบค่า จากโปรแกรมดังรูปที่ 3.34 จะใช้การตรวจสอบโดยตัวเลขที่เป็นรหัสแอสกี้ ซึ่งเทียบตัวเลข 0-9 ได้ดังตารางที่ 3.3

ตารางที่ 3.3 เลขฐานสิบเทียบกับรหัสแอสกี้

| Decimal | Char | Decimal | Char |
|---|---|---|---|
| 48 | 0 | 53 | 5 |
| 49 | 1 | 54 | 6 |
| 50 | 2 | 55 | 7 |
| 51 | 3 | 56 | 8 |
| 52 | 4 | 57 | 9 |

ถ้าเงื่อนไขเป็นจริง จะเก็บรหัสผ่านที่ได้รับมาไว้ใน pass_buf และแสดงผลผ่านจอแอลซีดี

```
179   while(1)
180   {
181     if(U2LSR & 0x01)
182     {
183       uart_data = getchar2();
184       if ((uart_data)>=48 && (uart_data)<=57)
185       {
186         pass_buf[pass_digit] = uart_data;
187         pass_digit++;
188         goto_cursor(11);
189         lcd_print(pass_buf);
190       }
191     }
```

รูปที่ 3.34 การรับค่าและแสดงค่ารหัสผ่าน

ในขั้นตอนต่อไป รหัสที่ได้รับมาจะถูกตรวจสอบกับรหัสผ่านที่ส่งมาจากเซิร์ฟเวอร์ โดยเมื่อรหัสที่ผู้ใช้บริการกดเข้ามาครบ 4 ตัว จะทำการตรวจสอบตัวแปร pass (เก็บรหัสผ่านจากเซิร์ฟเวอร์) กับ pass_buf (เก็บรหัสผ่านจากผู้ใช้บริการ) ถ้ามีค่าตรงกันทั้งหมด หมายความว่ารหัสถูกต้อง และถ้าไม่ถูกต้อง ผู้ใช้บริการต้องกดรหัสผ่านใหม่ มีโค้ดโปรแกรม ดังรูปที่ 3.35

```
193     if(pass_digit == 4)
194     {
195       pass_digit = 0;
196       goto_cursor(11);
197       lcd_print("       ");
198
199       for(loop = 0;loop <= 3;loop++)
200       {
201         if((pass[loop][0] == pass_buf[0]) && (pass[loop][1] == pass_buf[1])
202           && (pass[loop][2] == pass_buf[2]) && (pass[loop][3] == pass_buf[3]))
203         {
204           true_pass = loop;
205         }
206       }
207
208       if(true_pass > 3 || true_pass < 0) //If pass wrong
209       {
210         true_pass = -1;//Enter new pass
211         goto_cursor(0x00);
212         lcd_print("Wrong password");
213         delay(10000000);
214         goto_cursor(0x00);
215         lcd_print("Enter pass:   ");
216       }
217       else
218       {
219         goto_cursor(0x00);
220         lcd_print("Password accept");
221       }
```

รูปที่ 3.35 การตรวจสอบรหัสผ่านจากผู้ใช้และเซิร์ฟเวอร์

8) การตรวจสอบจำนวนเหรียญและการสั่งเครื่องซักผ้าทำงาน

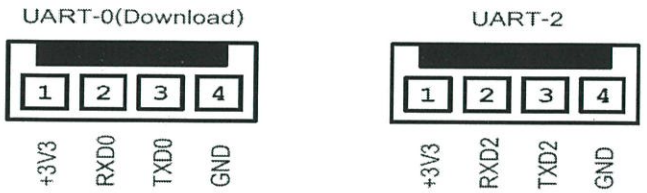
เมื่อผู้ใช้บริการกดรหัสผ่านถูกต้องแล้ว จะเข้าสู่ขั้นตอนการหยอดเหรียญ เมื่อหยอดเหรียญจนครบ (ในการทดลองตั้งไว้ที่ 30 บาท) จะสั่งให้เครื่องซักผ้าทำงานด้วยคำสั่ง FIOSET = LED เป็นการสิ้นสุดขั้นตอน มีโค้ดโปรแกรม ดังรูปที่ 3.36

```
229     getCoin();
230     if(true_pass <= 3 && true_pass >= 0) //If pass ok
231     {
232       if(CoinCount == 3)
233       {
234         pass[true_pass][0] = 'X';
235         pass[true_pass][1] = 'X';
236         pass[true_pass][2] = 'X';
237         pass[true_pass][3] = 'X';
238         switch(true_pass)
239         {
240           case 0:
241             FIO2SET = LED1;
242             true_pass = -1;//Ready to enter next pass
243             CoinCount = 0;
244             break;
245           case 1:
246             FIO2SET = LED2;
247             true_pass = -1;//Ready to enter next pass
248             CoinCount = 0;
249             break;
250           case 2:
251             FIO2SET = LED3;
252             true_pass = -1;//Ready to enter next pass
253             CoinCount = 0;
254             break;
255           case 3:
256             FIO2SET = LED4;
257             true_pass = -1;//Ready to enter next pass
258             CoinCount = 0;
259             break;
260         }
```

รูปที่ 3.36 การตรวจสอบจำนวนเหรียญและการสั่งเครื่องซักผ้าทำงาน

9) การใช้งาน RS232 เพื่อทำการโหลดโปรแกรมเข้าบอร์ด

หลังจากที่ทำการคอมไพล์โปรแกรมสำเร็จ จะทำการอัพโหลดลงไมโครคอนโทรลเลอร์ โดยผ่านพอร์ต RS232 เป็นสัญญาณ RS232 ซึ่งผ่านวงจรแปลงระดับสัญญาณ MAX3232 เรียบร้อยแล้วโดยมีจำนวน 2 ช่อง ด้วยกันคือ UART-0 และ UART-2 ดังรูปที่ 3.37 โดยทั้ง 2 ช่องสามารถใช้เชื่อมต่อกับสัญญาณ RS232 เพื่อรับส่งข้อมูลได้ นอกจากนี้แล้ว UART-0 ยังสามารถใช้งานเป็น ISP Download สำหรับทำการดาวน์โหลด Hex File ให้กับ MCU ได้ด้วย โดยในปริญญานิพนธ์นี้ใช้งานร่วมกับ SW1 (ISP LOAD) และ SW2 (RESET) เพื่อรีเซ็ตให้ ซีพียูเริ่มต้นทำงานใน Boot-Loader Mode เพื่อทำการ Download Hex File ให้กับซีพียู



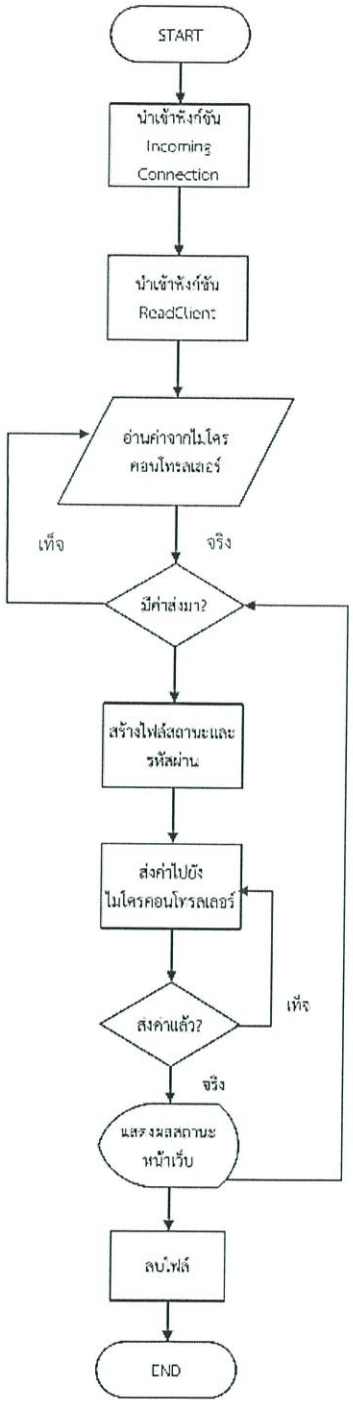

รูปที่ 3.37 ช่องและพอร์ตเชื่อมต่อสัญญาณ RS232

จากรูปที่ 3.37 ช่องและพอร์ตเชื่อมต่อสัญญาณ RS232 ใช้สำหรับเชื่อมต่อบอร์ดทดลอง ARM7 LPC2368 เข้ากับคอมพิวเตอร์ที่ไม่มีพอร์ตอนุกรม จะไม่สามารถใช้สายเคเบิล RS232 เชื่อมต่อเข้ากับคอมพิวเตอร์ได้โดยตรง ดังนั้นจึงต้องใช้พอร์ตอนุกรมเสมือนแบบยูเอสบี เป็นตัวกลางในการสื่อสารข้อมูลระหว่างคอมพิวเตอร์และบอร์ดทดลอง

### 3.1.2.2 เซิร์ฟเวอร์และฐานข้อมูล

1) การทำงานของเซิร์ฟเวอร์

การทำงานของเซิร์ฟเวอร์ต้องผ่านโปรแกรมกลางซึ่งคือ โปรแกรม QT Creator เพื่อเป็นตัวกลางในการติดต่อระหว่างไมโครคอนโทรลเลอร์กับเซิร์ฟเวอร์ เหตุผลที่ต้องมีโปรแกรมกลางคือเพื่อให้ลดภาระและปัญหาที่จะตามมาต่อการเขียนโปรแกรมของไมโครคอนโทรลเลอร์ โดยมีโฟลว์ชาร์ตและโค้ดโปรแกรมการทำงาน ดังรูปที่ 3.38

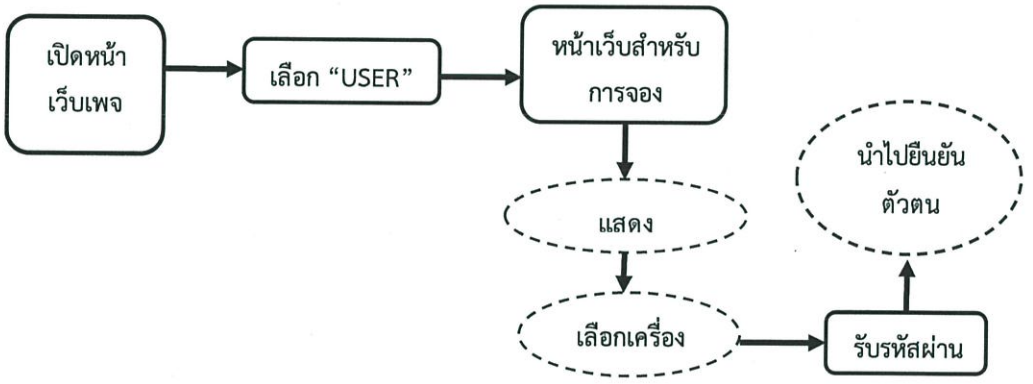



```
#include "server.h"

server::server(quint16 port, QObject *parent) :
    QTcpServer(parent)
{
    listen(QHostAddress::Any, port);
}
void server::incomingConnection(qintptr socketDescriptor)
{
    QTcpSocket *Socket = new QTcpSocket(this);
    connect(Socket,SIGNAL(readyRead()),this,SLOT(readClient()));
    if(!Socket->setSocketDescriptor(socketDescriptor))
    {
        delete Socket;
        return;
    }

}
void server::readClient(void)
{
    QTcpSocket *Socket =(QTcpSocket*)sender();
    if(Socket->canReadLine())
    {   QFile file ("C:\\AppServ\\www\\Customer\\file.txt");
        file.open(QIODevice::WriteOnly);
        QTextStream in(&file);
        QString request = Socket->readAll();
        in<<request;
        file.close ();
        qDebug() << Socket->peerAddress().toString();

        QFile file2 ("C:\\AppServ\\www\\Customer\\file2.txt");
        file2.open(QIODevice::ReadOnly);
        QTextStream out(&file2);
        QString common;
        out>>common;
        file2.close();
        QFile::remove("C:\\AppServ\\www\\Customer\\file2.txt");



        QTextStream os(Socket);
        os << common<<"END";
    }
    Socket->close();
    if(Socket->state() == QTcpSocket::UnconnectedState)
    {
        delete Socket;
    }
}
```

รูปที่ 3.38 โฟลว์ชาร์ตและโค้ดโปรแกรมการทำงานของเซิร์ฟเวอร์ผ่านโปรแกรม QT Creator

2) การทำงานของเว็บเซิร์ฟเวอร์ร่วมกับเครื่องลูกข่าย

การทำงานของเว็บเซิร์ฟเวอร์สามารถแสดงได้ดังรูปที่ 3.39 อธิบายได้ว่าเครื่องลูกข่ายสามารถใช้งานของระบบนี้ได้เมื่อทำการเปิดหน้าเว็บเพจ แล้วกดเลือก USER ถ้าต้องการจองเซิร์ฟเวอร์จะทำการส่งหน้าเว็บเพื่อให้เลือกเครื่องที่ต้องการจองและแสดงสถานะของแต่ละเครื่อง เมื่อเลือกเครื่องที่ต้องการแล้ว จะขึ้นรหัสผ่านเพื่อนำไปยืนยันตัวตนและขึ้นเวลาคงเหลือของเครื่องซักผ้าบวกกับเวลาจำกัดการใช้งานผู้ใช้ ที่เวลา 15 นาที ถ้าผู้ใช้ที่จองไม่มายืนยันตัวตนภายในเวลา 15 นาที หลังจากเครื่องซักผ้าเครื่องนั้นหมดเวลา ระบบจะตัดให้เครื่องนั้นว่างทันที



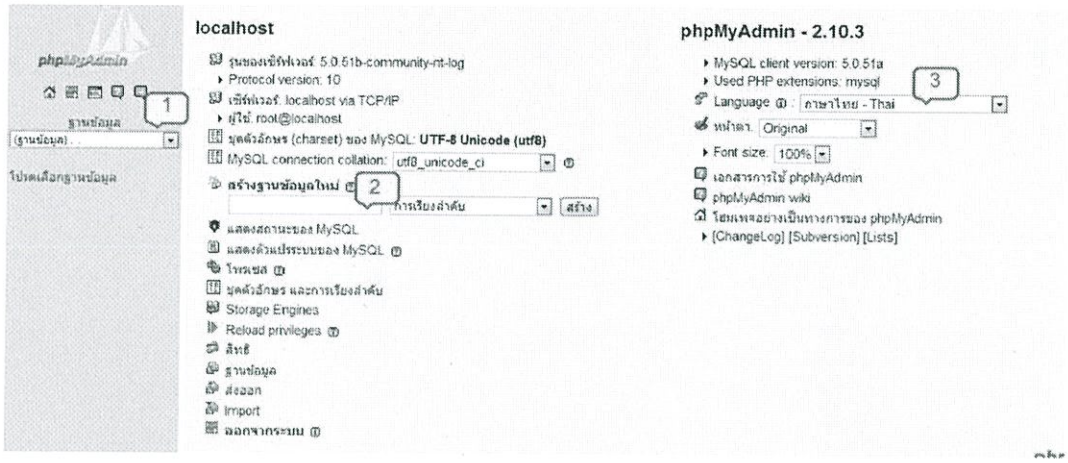

รูปที่ 3.39 การทำงานของเว็บเซิร์ฟเวอร์ร่วมกับเครื่องลูกข่าย

3) ฐานข้อมูล

การออกแบบระบบฐานข้อมูล ได้ออกแบบให้สอดคล้องกับข้อมูลที่รับเข้ามา ซึ่งใช้โปรแกรมAppServ (แอปเซิร์ฟ) ในส่วนของ phpMyAdmin (พีเอชพีมายแอดมิน) ซึ่งใช้จัดการฐานข้อมูล MySQL (มายเอสคิวแอล) ที่ต้องการ สำหรับการเริ่มต้นจะติดตั้งโปรแกรมแอปเซิร์ฟเพื่อให้เครื่องเป็นเครื่องฐานข้อมูล โดยการเข้าถึงการใช้งานของพีเอชพีมายแอดมินและฐานข้อมูลมายเอสคิวแอลมีวิธีทำดังขั้นตอนต่อไปนี้

1. การเข้าใช้งาน

การเข้าใช้งานโปรแกรมพีเอชพีมายแอดมินจะใช้งานผ่าน Web Browser (เว็บบราวเซอร์) และพิมพ์URL (ยูอาร์แอล) ไปยังไดเรกทอรี่ของพีเอชพีมายแอดมิน เช่น http://localhost/ phpMyAdmin เป็นต้น (ชื่อไดเรกทอรี่อาจจะเป็นชื่ออื่น) จะได้หน้าต่างของพีเอชพีแอดมิน ดังรูปที่ 3.40




รูปที่ 3.40 หน้าต่างของพีเอชพีมายแอดมิน

จากรูปที่ 3.40 จะเห็นได้ว่าหน้าต่างของพีเอชพีมายแอดมินแบ่งออกเป็น 3 ส่วน อธิบายในแต่ละส่วนได้ดังนี้

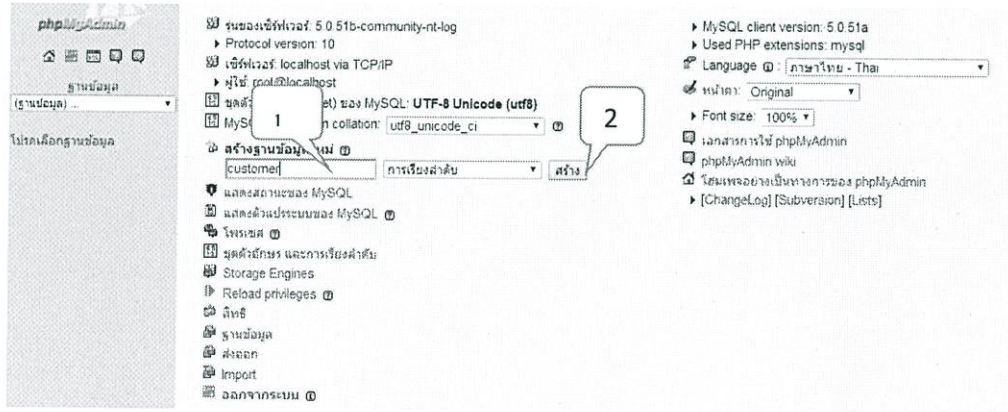
หมายเลข: 1 ใช้สำหรับเลือกเพื่อสร้างหรือเปลี่ยนแปลงฐานข้อมูล

หมายเลข: 2 ใช้สำหรับใส่ชื่อฐานข้อมูลที่ต้องการสร้าง

หมายเลข: 3 ใช้สำหรับเลือกภาษาที่ต้องการใช้ในโปรแกรมพีเอชพีมายแอดมิน

2. การสร้างฐานข้อมูล

ก่อนจะเก็บข้อมูลลงในมายเอสคิวแอล เราจำเป็นต้องสร้างชื่อของฐานข้อมูลนั้นๆก่อน จากนั้นจะทำการสร้างตาราง ฟิลด์ เพื่อบันทึกข้อมูล ต่อไปเป็นตัวอย่างการสร้างฐานข้อมูลใหม่ ซึ่งมีฐานข้อมูลชื่อว่า “customer” ดังรูปที่ 3.41




รูปที่ 3.41 ฐานข้อมูลที่ต้องการสร้างชื่อ “customer”

จากรูปที่ 3.41 อธิบายการสร้างฐานข้อมูลใหม่เป็นส่วนต่างๆ ได้ดังนี้

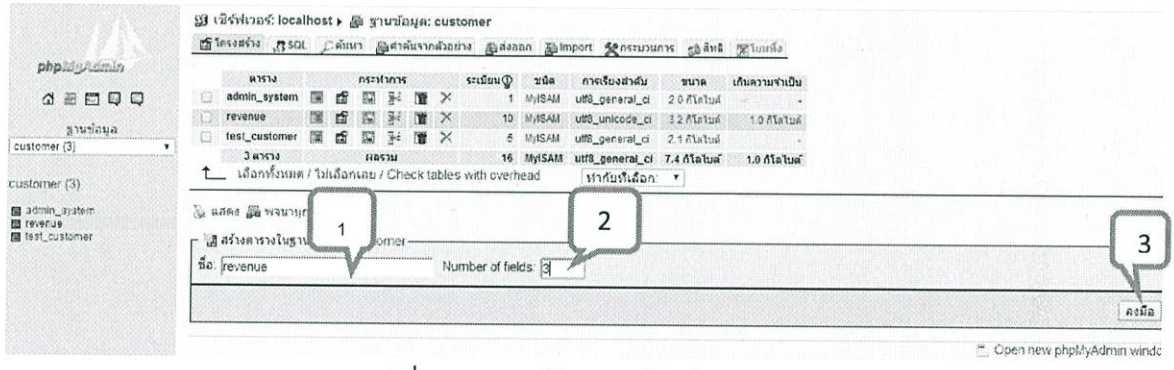
หมายเลข 1: ใส่ชื่อของฐานข้อมูลที่ต้องการสร้าง ในที่นี้ชื่อ “customer”

หมายเลข 2: คลิกที่ปุ่ม “สร้าง” เพื่อสร้างฐานข้อมูล

หลังจากสร้างฐานข้อมูลที่ต้องการเรียบร้อยแล้ว พีเอชพีมายแอดมินจะแจ้งให้ทราบว่าฐานข้อมูลยังไม่มีตารางใดๆ อยู่เลย ฐานข้อมูลที่สร้างขึ้นมาต้องการที่จะเก็บข้อมูลของผู้ใช้ โดยรายละเอียดของตารางจะมี ดังนี้

- ID คือ การเรียงลำดับ
- DATE คือ เวลา
- AMOUNT คือ จำนวนเงิน

และการสร้างตารางใหม่ แสดงดังรูปที่ 3.42




รูปที่ 3.42 การสร้างตารางใหม่

จากรูปที่ 3.42 แสดงการสร้างตารางใหม่ โดยอธิบายได้ดังนี้

หมายเลข 1: ใส่ชื่อตารางที่ต้องการสร้าง

หมายเลข 2: ใส่จำนวนฟิลด์ของตาราง สำหรับตาราง “customer” จะมีจำนวน 3 ฟิลด์

หมายเลข 3: คลิกปุ่ม “ลงมือ” เพื่อดำเนินการขั้นต่อไป

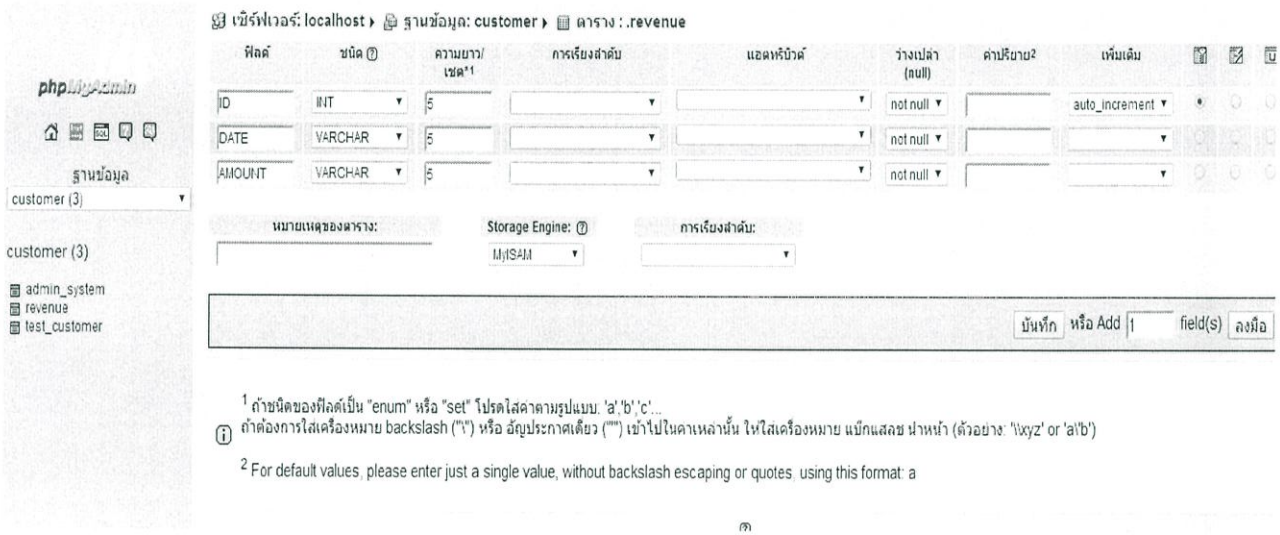
ขั้นตอนต่อไป ภายหลังจากคลิกปุ่ม “ลงมือ” ในขั้นตอนก่อนหน้า จะเป็นการกำหนด ชื่อฟิลด์ ชนิดของข้อมูลที่เก็บ และรายละเอียดอื่นๆ ซึ่งจะอธิบายในส่วนต่างๆ ได้ดังนี้

- ฟิลด์ สำหรับใส่ชื่อฟิลด์
- ชนิด สำหรับเลือกชนิดของข้อมูลที่ต้องการเก็บในฟิลด์นั้น
- ความยาว/เซต สำหรับกำหนดขนาดของข้อมูล

- แอตทริบิวต์ สำหรับเลือกลักษณะเฉพาะของข้อมูลที่จะเก็บ เช่น ตัวเลขแบบคิดเครื่องหมาย บวกหรือลบ เป็นต้น

- ค่าว่าเปล่า (null) สำหรับเลือกว่า ฟิลด์นั้นสามารถใส่ค่าว่างได้หรือไม่

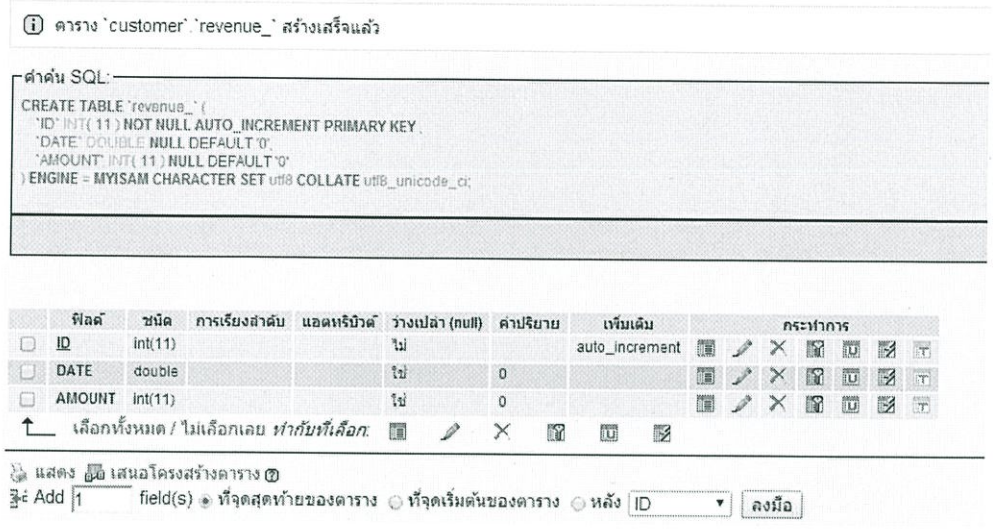
- ค่าปริยาย สำหรับกำหนดค่าเริ่มต้นของฟิลด์ (ค่า Default) โดยขั้นตอนนี้แสดงได้ดังรูปที่ 3.43




รูปที่ 3.43 การกำหนดค่าต่างๆให้กับรายละเอียดต่างๆ ของตาราง

จากรูปที่ 3.43 เป็นการกำหนดชื่อฟิลด์ ชนิดของข้อมูลที่เก็บ และรายละเอียดอื่นๆ ซึ่งจะอธิบายในส่วนต่างๆ โดยในแต่ละช่องนั้น ขึ้นอยู่กับรายละเอียดของตารางที่ได้ถูกกำหนดไว้ตั้งแต่แรก

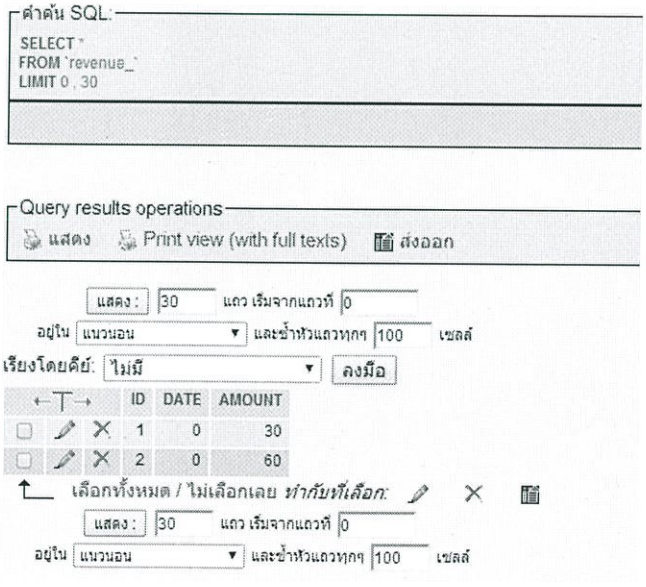
ขั้นตอนต่อไป หลังจากคลิกที่ปุ่ม “บันทึก” โปรแกรมจะแจ้งให้ทราบว่าได้สร้างตาราง revenue เรียบร้อยแล้ว และจะเข้าสู่หน้าจอสำหรับการจัดการตาราง ซึ่งจะมีเมนูต่างๆ หลายเมนูเพื่อใช้ในการจัดการตาราง และมีการแจ้งให้ผู้ใช้ทราบรายละเอียดอื่นๆที่เกี่ยวข้องกับตารางนั้นๆ เช่น ดัชนี เนื้อที่ที่ใช้งาน ค่าสถิติต่างๆ เป็นต้น ดังรูปที่ 3.44




รูปที่ 3.44 การสร้างตารางใหม่ที่เสร็จเรียบร้อย

3. การเปิดข้อมูลดูภายในตาราง

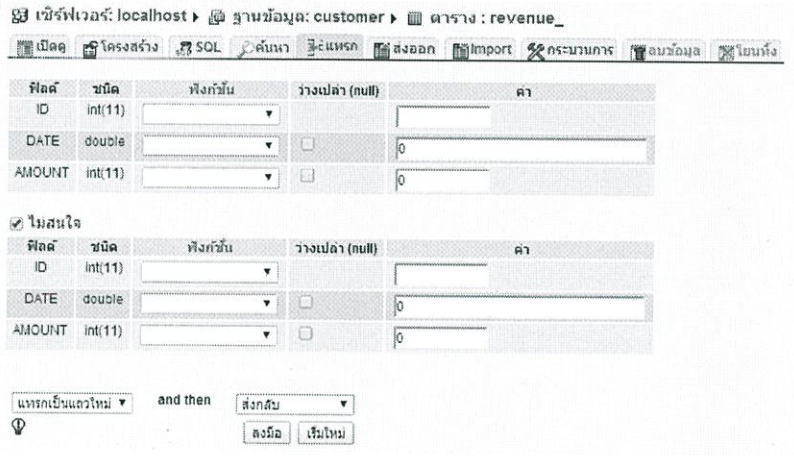
การเปิดดูข้อมูลภายในตาราง ทำได้โดยคลิกที่เมนู เปิดดู ซึ่งจะแสดงข้อมูลได้ก็ต่อเมื่อมีข้อมูลเข้ามาภายในตารางแล้วเท่านั้น สำหรับเมนู “เปิดดู” แสดงดังรูปที่ 3.45




รูปที่ 3.45 ข้อมูลที่อยู่ในฐานข้อมูลตามรายละเอียดของตารางที่สร้างขึ้นมา

4. การเพิ่มข้อมูลลงยังตาราง

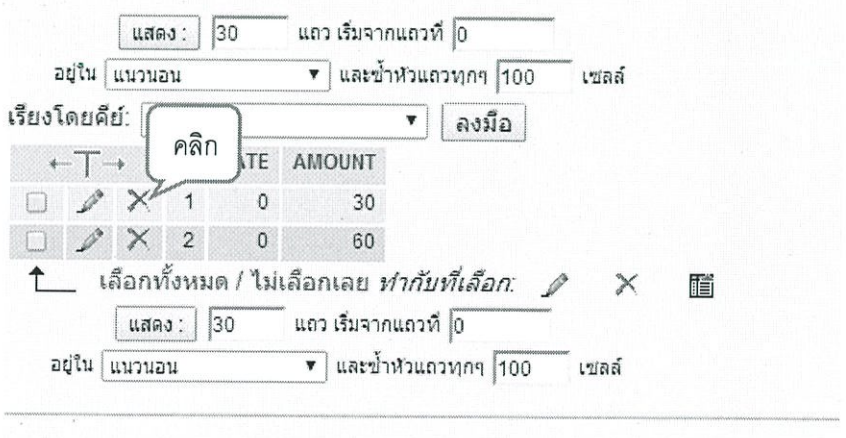
การเพิ่มข้อมูลลงยังตาราง สามารถทำได้โดยคลิกที่เมนู “แทรก” โปรแกรมจะแสดงหน้าจอสำหรับเพิ่มข้อมูลดังรูปที่ 3.46 จากนั้นให้กรอกข้อมูลที่ต้องการลงไปในคอลัมน์ ค่า (Value) เมื่อกรอกข้อมูลเรียบร้อยแล้ว หากต้องการกลับมาเพิ่มข้อมูลในแถวต่อไปเลย ให้คลิกที่ “แทรกระเบียนใหม่” จากนั้นคลิกที่ “ลงมือ”




รูปที่ 3.46 หน้าจอสำหรับเพิ่มข้อมูล

5. การลบข้อมูลในตาราง

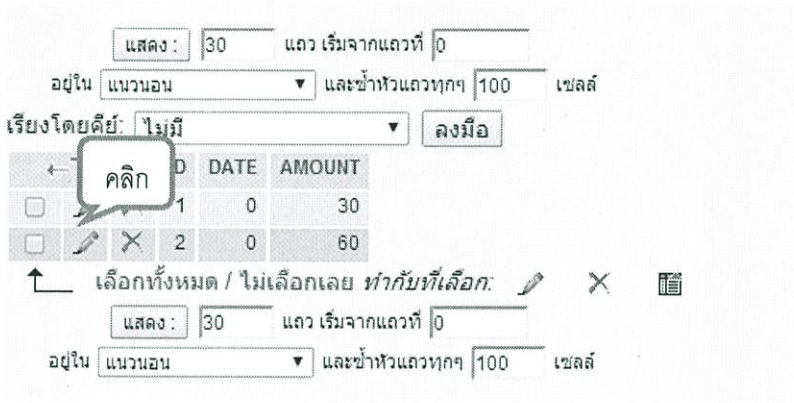
การลบข้อมูลภายในตาราง คลิกที่เมนู “เปิดดู” โปรแกรมจะแสดงรายการข้อมูล จากนั้นคลิก เพื่อลบข้อมูล ดังรูปที่ 3.47 (สามารถให้คำสั่ง SQL ในการลบข้อมูลได้อีกวิธีหนึ่ง)




รูปที่ 3.47 หน้าต่างการลบข้อมูลภายในตาราง

6. การแก้ไขข้อมูลในตาราง

การแก้ไขข้อมูลในตาราง คลิกที่เมนู “เปิดดู” โปรแกรมจะแสดงรายการข้อมูลจากนั้นคลิก เพื่อแก้ไขข้อมูล ดังรูปที่ 3.48 (สามารถให้คำสั่ง SQL ในการลบข้อมูลได้อีกวิธีหนึ่ง) เมื่อคลิกที่รูปแก้ไขแล้ว จะปรากฏหน้าจอสำหรับแก้ไขข้อมูลดังรูปที่ 3.48 เป็นช่อง สำหรับแก้ไข โดยจะแสดงทุกๆ ฟิลด์ภายในตาราง และหน้าจอแก้ไขนี้สามารถบันทึกหรือแทรกข้อมูลเป็นแถวใหม่ได้ด้วย




รูปที่ 3.48 หน้าต่างการแก้ไขข้อมูลในตาราง

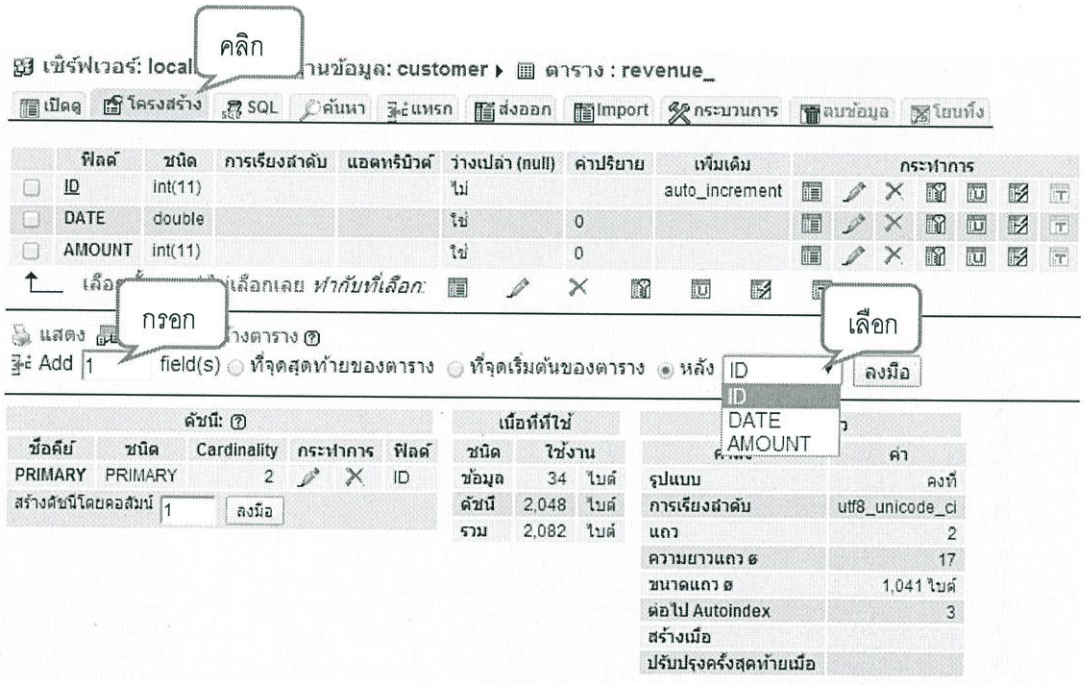
7. การลบตาราง

การลบตารางออกจากฐานข้อมูล ทำได้โดยคลิกที่ชื่อฐานข้อมูลทางด้านซ้ายของโปรแกรม เพื่อแสดงตารางทั้งหมดขึ้นมา จากนั้นเลือกคลิกเพื่อลบตารางที่ต้องการ แสดงการลบตาราง ดังรูปที่ 3.49

รูปที่ 3.49 หน้าต่างการลบตารางออกจากฐานข้อมูล

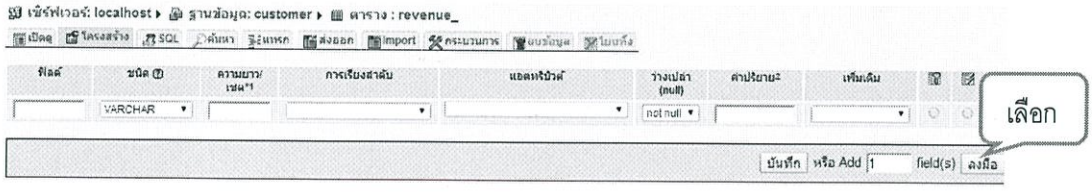
8. การเพิ่มฟิลด์ใหม่ลงไปในตารางที่มีอยู่แล้ว

การเพิ่มฟิลด์ใหม่ลงในตารางที่มีอยู่แล้ว จะดำเนินการผ่านเมนูโครงสร้าง ขั้นแรกให้คลิกที่ขั้นแรกให้คลิกที่เมนูโครงสร้าง จะพบส่วนของการเพิ่มฟิลด์ใหม่ ซึ่งต้องระบุจำนวนฟิลด์ที่ต้องการเพิ่มและระบุว่าจะเพิ่มฟิลด์ใหม่ไว้หน้าหรือหลังฟิลด์ใดในตาราง การเพิ่มฟิลด์ใหม่แสดงดังรูปที่ 3.50




รูปที่ 3.50 การเพิ่มฟิลด์ใหม่ลงไปในตารางที่มีอยู่

ขั้นตอนต่อไปเป็นการสร้างชื่อฟิลด์ใหม่ คลิกที่ “ลงมือ” จะได้ฟิลด์ใหม่ตามที่ต้องการ ดังรูปที่ 3.51




รูปที่ 3.51 หน้าต่างการสร้างชื่อสำหรับฟิลด์ใหม่

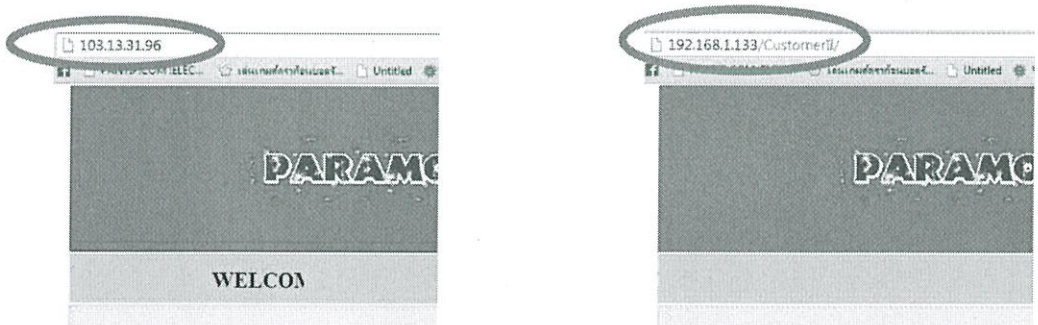
9. การลบฐานข้อมูล (Drop Database)

การลบฐานข้อมูลทำได้โดยคลิกเลือกที่ชื่อฐานข้อมูลจากเมนูทางด้านซ้ายของหน้าจอก่อน จากนั้นคลิกเลือกที่ "โยนทิ้ง" โปรแกรมจะแสดงข้อความยืนยันก่อนลบฐานข้อมูล แสดงดังรูปที่ 3.52

รูปที่ 3.52 หน้าต่างการลบฐานข้อมูล

4) เว็บเพจ

ผู้จัดทำได้ออกแบบหน้าเว็บให้สามารถใช้งานได้กับไอพีแอดเดรสภายนอกและภายในวงแลนเดียวกันได้ ซึ่งผู้ใช้งานสามารถใช้งานหน้าเว็บเพจได้จากภายนอกวงแลนโดยผ่านไอพีแอดเดรส 103.13.31.96 และใช้งานจากไอพีของเครื่องเซิร์ฟเวอร์ภายในวงแลน 192.168.1.133 ได้ดังรูปที่ 3.53



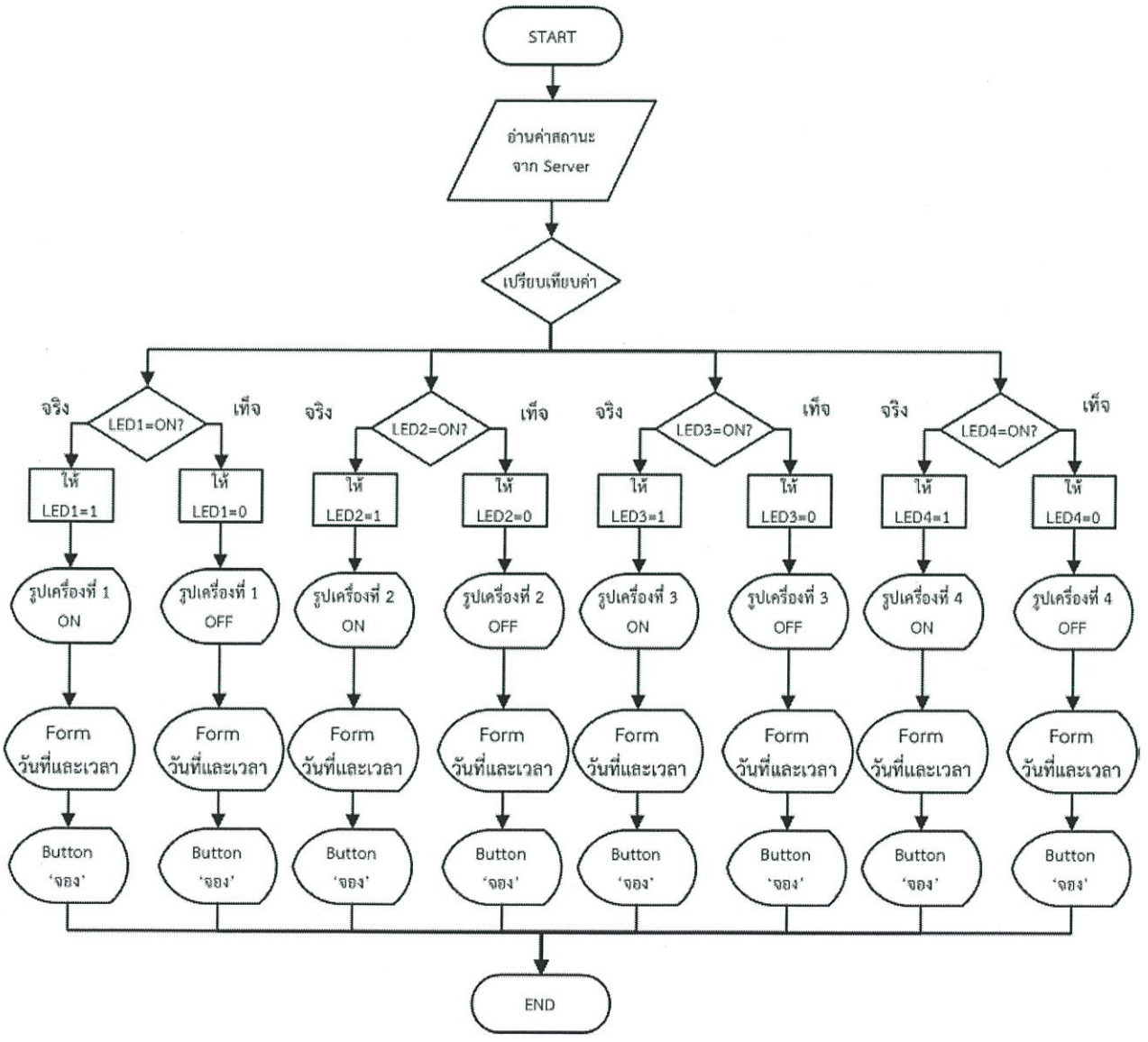

(ก) (ข)

รูปที่ 3.53 หน้าเว็บเพจ

(ก) หน้าเว็บเพจเมื่อแสดงผ่านไอพีภายนอกวงแลน 103.13.31.96

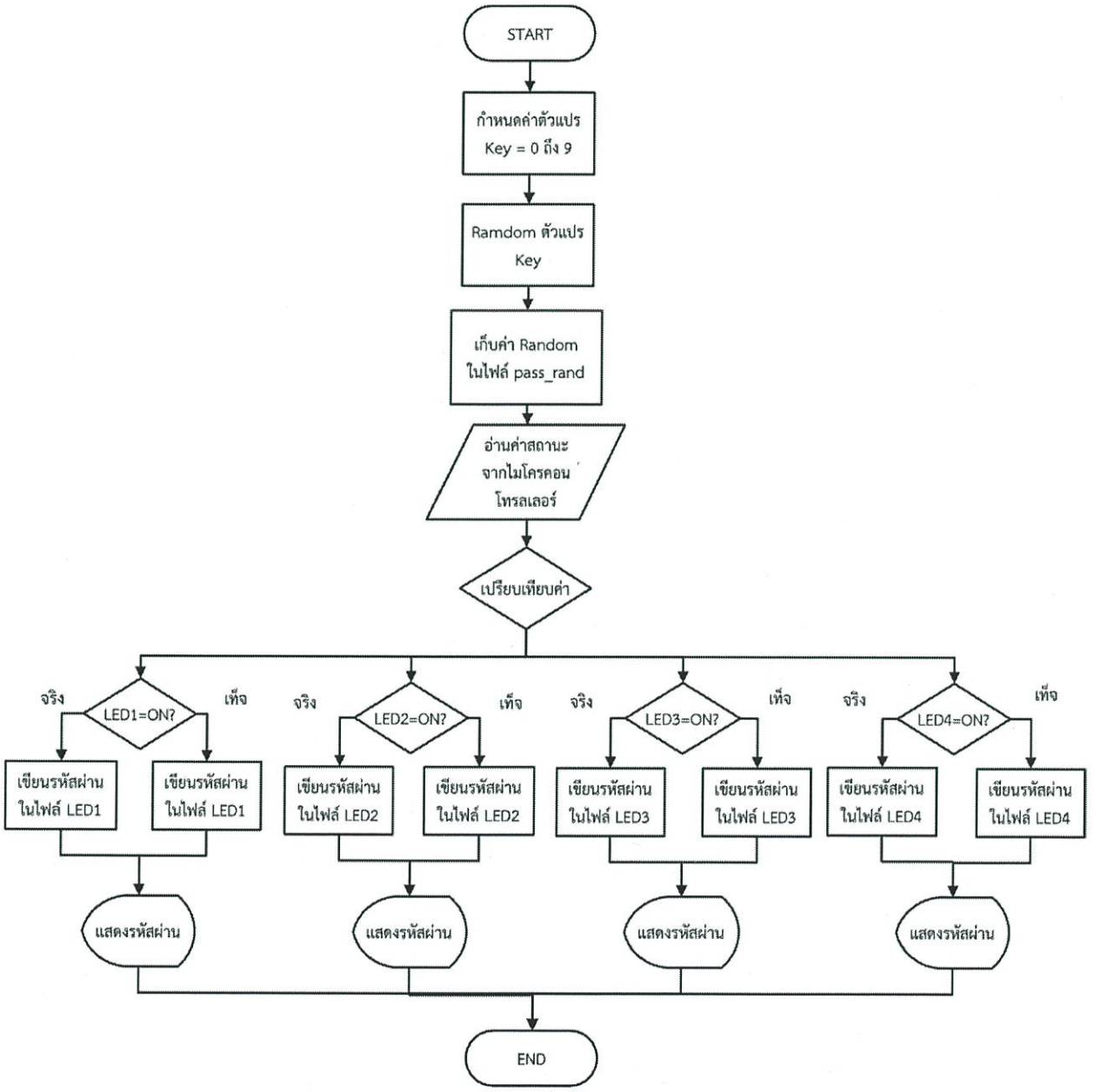
(ข) หน้าเว็บเพจเมื่อแสดงผ่านไอพีภายในวงแลน 192.168.1.133

เมื่อผู้ใช้เลือก USER จะขึ้นหน้าเว็บสำหรับการจอง โดยการสร้างหน้าเว็บนี้มีโฟลว์ชาร์ตแสดงสถานะของเครื่องซักผ้า ดังรูปที่ 3.54 ซึ่งอธิบายได้ว่าการแสดงสถานะของเครื่องซักผ้า จะมีการรับค่าสถานะมาจากเซิร์ฟเวอร์ โดยค่านี้รับมาจากไมโครคอนโทรลเลอร์อีกที และนำค่ามาเปรียบเทียบค่า LED ของแต่ละเครื่องว่า ON หรือ OFF ถ้า ON ซึ่งมีค่าคือ1จะแสดงรูปที่กำหนดไว้ รูปแบบวันที่เวลา และปุ่มกด “จอง” แต่ถ้า OFF ซึ่งมีค่าคือ 0 จะแสดงรูปภาพที่กำหนดไว้ รูปแบบวันที่เวลา และปุ่มกด “จอง”และจบการทำงาน



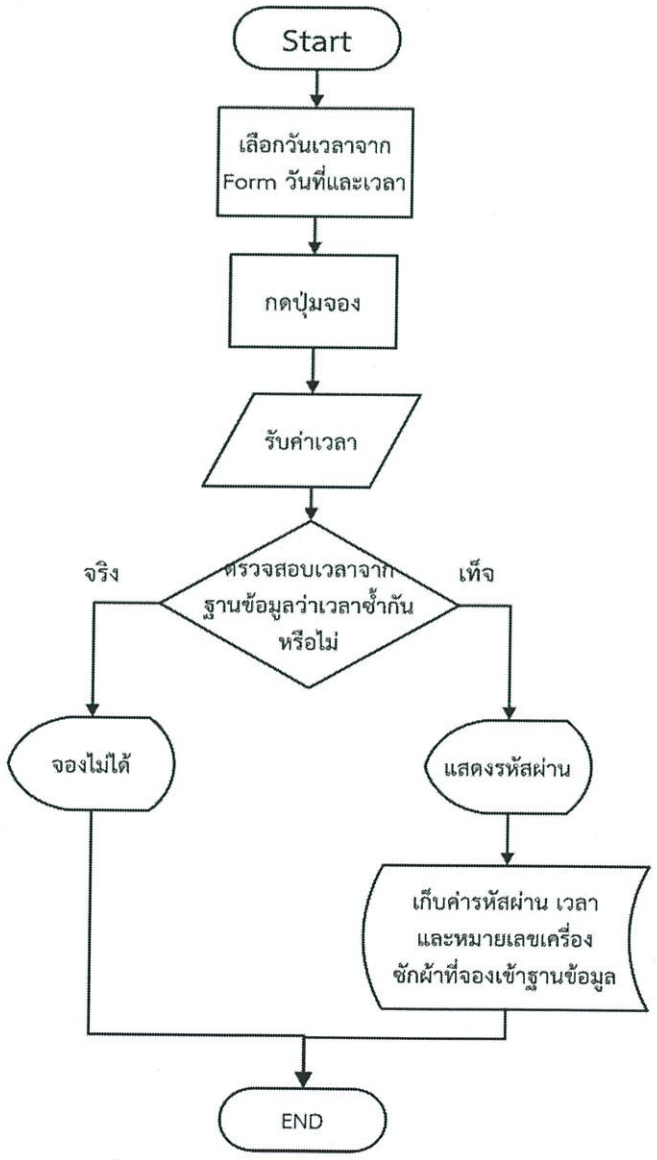

รูปที่ 3.54 โฟลว์ชาร์ตที่แสดงสถานะของเครื่องซักผ้า

ส่วนประกอบอีกส่วนหนึ่งของหน้าเว็บคือการแสดงรหัสผ่าน มีโฟลว์ชาร์ตการทำงาน ดังรูปที่ 3.55 อธิบายการทำงานได้ว่ากำหนดตัวแปร key ให้มีค่า 0-9 หลังจากนั้นทำการสุ่มเลขขึ้นมา 4 หลัก และให้เก็บค่าไว้ในตัวแปร pass_rand โดยรหัสผ่านจะแสดงก็ต่อเมื่อมีการกดปุ่มจอง



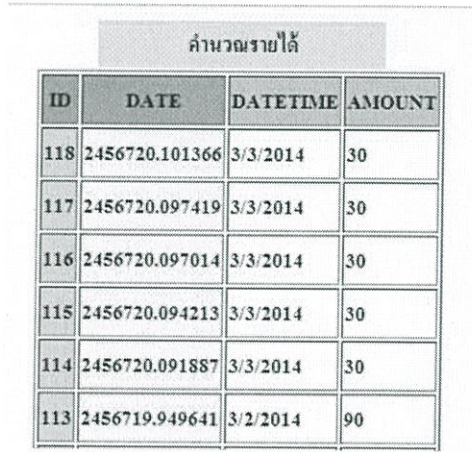

รูปที่ 3.55 โฟลว์ชาร์ตการแสดงรหัสผ่าน

ส่วนประกอบสุดท้ายของหน้าเว็บนี้คือการเลือกเวลาจอง มีโฟลว์ชาร์ตการทำงาน ดังรูปที่ 3.56 อธิบายการทำงานได้ว่า การเลือกเวลาจองนี้จะเป็นการจำลองเพียงหนึ่งเครื่อง อีกสามเครื่องที่เหลือทำงานเหมือนกัน ถ้าต้องการจองจะสามารถเลือกวันและเวลาที่ต้องการได้โดยมีรูปแบบวันและเวลาให้เลือก เมื่อกดปุ่มจองจะรับค่าเวลาที่เลือก แล้วนำไปตรวจสอบกับเวลาในฐานข้อมูล “booking” ที่มีการบันทึกเวลาจองอื่นๆ ไว้ ถ้าเวลาตรงกันก็จะไม่ให้จอง จะแสดงข้อความว่า “จองไม่ได้” แต่ถ้าไม่ตรงกันก็จะให้จองได้ แสดงรหัสผ่าน และเก็บค่ารหัสผ่าน เวลาที่ต้องการจอง หมายเลขเครื่องที่จอง ไว้ในฐานข้อมูล “booking” ต่อไป




รูปที่ 3.56 โฟลว์ชาร์ตการเลือกเวลาการจอง

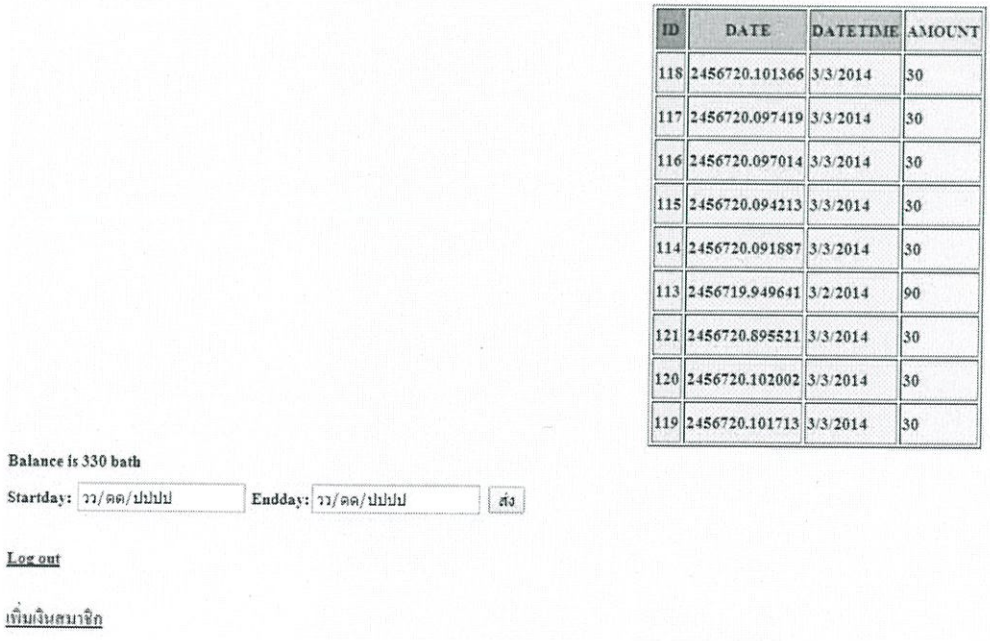
หน้าเว็บแสดงฐานข้อมูลประกอบไปด้วยตารางฐานข้อมูลที่เก็บค่า ดังรูปที่ 3.57 และการคำนวณรายได้ ดังรูปที่ 3.58 ซึ่งผู้จัดทำได้ออกแบบให้มีการเลือกวันเริ่มต้นและวันสิ้นสุดของการคำนวณรายได้ได้



คำนวณรายได้

| ID | DATE | DATETIME | AMOUNT |
|---|---|---|---|
| 118 | 2456720.101366 | 3/3/2014 | 30 |
| 117 | 2456720.097419 | 3/3/2014 | 30 |
| 116 | 2456720.097014 | 3/3/2014 | 30 |
| 115 | 2456720.094213 | 3/3/2014 | 30 |
| 114 | 2456720.091887 | 3/3/2014 | 30 |
| 113 | 2456719.949641 | 3/2/2014 | 90 |

รูปที่ 3.57 หน้าเว็บแสดงฐานข้อมูล



| ID | DATE | DATETIME | AMOUNT |
|---|---|---|---|
| 118 | 2456720.101366 | 3/3/2014 | 30 |
| 117 | 2456720.097419 | 3/3/2014 | 30 |
| 116 | 2456720.097014 | 3/3/2014 | 30 |
| 115 | 2456720.094213 | 3/3/2014 | 30 |
| 114 | 2456720.091887 | 3/3/2014 | 30 |
| 113 | 2456719.949641 | 3/2/2014 | 90 |
| 121 | 2456720.895521 | 3/3/2014 | 30 |
| 120 | 2456720.102002 | 3/3/2014 | 30 |
| 119 | 2456720.101713 | 3/3/2014 | 30 |

Balance is 330 bath

Startday: วว/ดด/ปปปป Endday: วว/ดด/ปปปป ส่ง

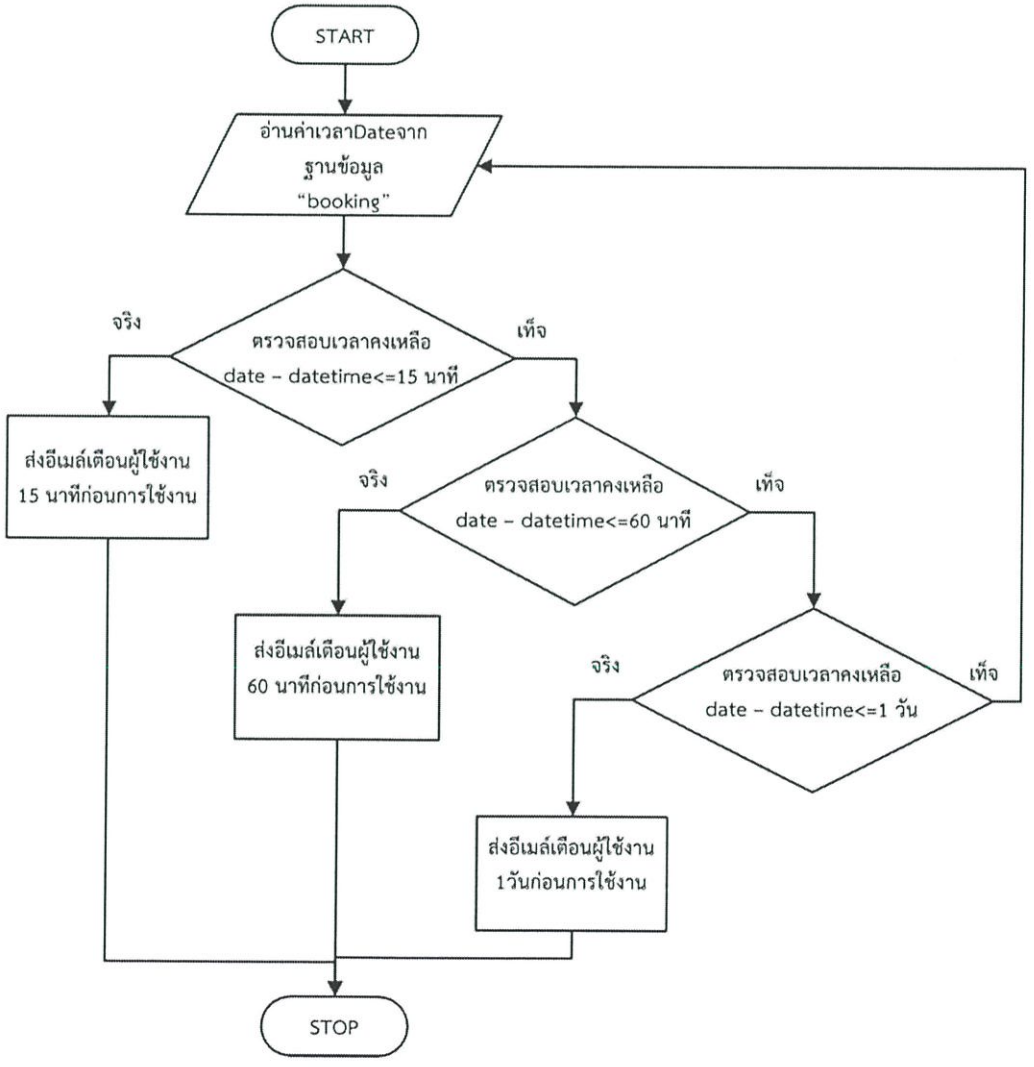
Log out

เพิ่มเงินสมาชิก

รูปที่ 3.58 หน้าเว็บการคำนวณรายได้

5) การส่งอีเมล์แจ้งเตือนผู้ใช้งาน

การส่งอีเมล์แจ้งเตือนผู้ใช้งาน มีโฟลว์ชาร์ตการทำงานดังรูปที่ 3.59 อธิบายการทำงานได้ว่า ระบบจะอ่านค่าเวลา “Date” จากฐานข้อมูล “booking” ทำการตรวจสอบเวลาคงเหลือโดย นำ “date-datetime” ซึ่ง date คือเวลาในฐานข้อมูล datetime คือเวลาปัจจุบัน ถ้าผลต่างมีค่าน้อยกว่าหรือเท่ากับ 15 นาที ระบบจะส่งอีเมล์แจ้งเตือนผู้ใช้งาน แต่ถ้าเกินเวลา 15 นาที ให้ทำเงื่อนไขต่อไป ซึ่งเงื่อนไขต่อไปก็คือถ้าผลต่างมีค่าน้อยกว่าหรือเท่ากับ 60 นาที ให้ส่งอีเมล์แจ้งเตือนผู้ใช้งาน แต่ถ้าไม่ก็ทำเงื่อนไขสุดท้าย ซึ่งเงื่อนไขสุดท้ายก็คือถ้าผลต่างมีค่าน้อยกว่า 1 วัน ระบบจะส่งอีเมล์แจ้งเตือนผู้ใช้งาน แต่ถ้าไม่ก็กลับไปอ่านค่าเวลาใหม่เรื่อยๆ




รูปที่ 3.59 โฟลว์ชาร์ตแสดงการทำงานการส่งอีเมล์แจ้งเตือนผู้ใช้งาน

## 3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 เครื่องซักผ้า ยี่ห้อซัมซุง 6.5 กิโลกรัม รุ่น WA85G5

3.2.2 เครื่องหยอดเหรียญ

3.2.3 ไมโครคอนโทรลเลอร์ CP-JR ARM7 LPC2368

3.2.4 บอร์ดและโปรแกรมอาร์ดูอิโน่

3.2.5 วงจรตรวจสอบสถานะโดยใช้แอลดีอาร์รับแสง

3.2.6 วงจรตรวจสอบสถานะโดยใช้ออปโต้คัปเปลอร์

3.2.7 วงจรบันทึกและเล่นเสียง

3.2.8 ฐานข้อมูล

3.2.9 เครื่องออสซิลโลสโคป (Oscilloscope)

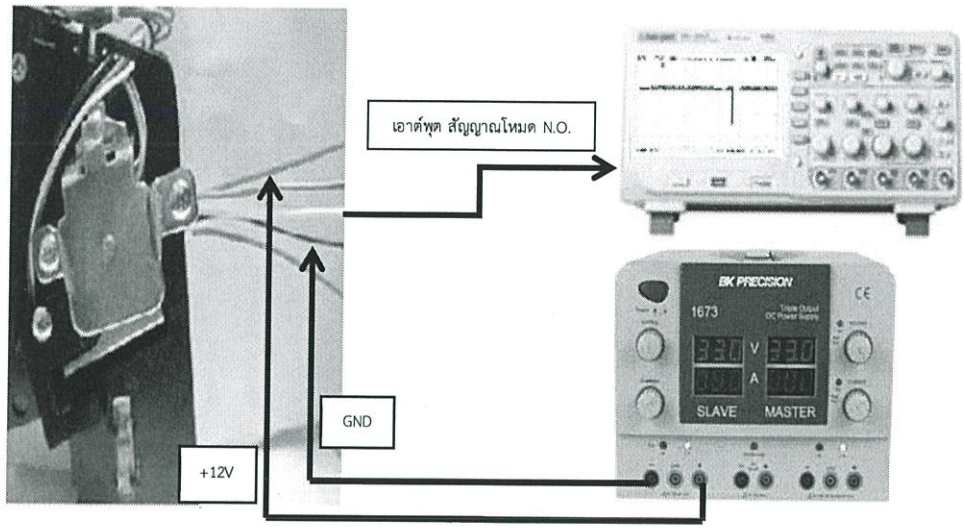
3.2.10 เครื่องกำเนิดสัญญาณ (Function Generato)

3.2.11 พาวเวอร์ซัพพลาย (Power Supply)

## 3.3 การจัดเก็บผลการทดลอง

### 3.3.1 การทำงานของเครื่องหยอดเหรียญ

สัญญาณที่ออกจากเครื่องหยอดเหรียญปกติพบว่ามีขนาดของสัญญาณต่ำเกินกว่าที่จะนำไปเข้าพอร์ตอินพุตของไมโครคอนโทรลเลอร์ ซึ่งในปริญญานิพนธ์นี้จะเลือกใช้โหมดการทำงานของเครื่องหยอดเหรียญเป็นโหมด N.O. (Normally Open) การแปลงสัญญาณนั้น สัญญาณเอาต์พุตที่ได้ต้องมีขนาดแรงดัน 3.3 โวลต์ (High) เพื่อนำไปใช้ต่อกับไมโครคอนโทรลเลอร์ ARM7 โดยการใช้ R Pull up คือ การใช้ตัวต้านทานต่อเข้ากับ $V_{cc}$ (+3.3 โวลต์) เพื่อรักษาระดับของแรงดันให้คงที่ ทำให้อยู่ในสถานะลอจิก "1" ตลอดเวลา ทำการเก็บผลการทดลอง ดังรูปที่ 3.60 และผลการทดลอง ดังรูปที่ 4.1

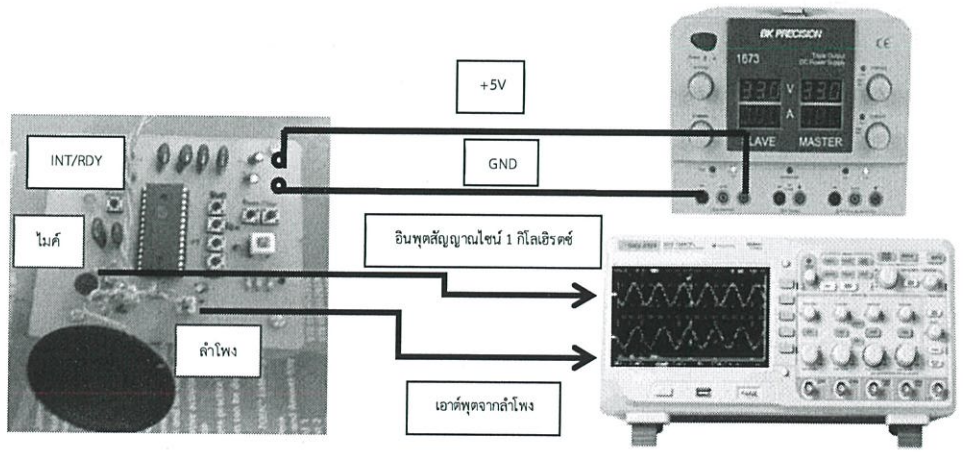



รูปที่ 3.60 การเก็บผลสัญญาณจากเครื่องหยอดเหรียญ

### 3.3.2 การทำงานของวงจรบันทึกและเล่นเสียง

1) การป้อนสัญญาณไซน์ที่อินพุต เพื่อเปรียบเทียบกับสัญญาณที่เอาต์พุต และ

ผู้จัดทำปริญญานิพนธ์ได้ทำการบันทึกเสียงลงในไอซีบันทึกและเล่นเสียง แล้วจึงกดสวิตช์เพื่อทำการเล่นเสียงที่ถูกบันทึกไว้ออกทางลำโพง โดยเสียงที่บันทึกเป็นเสียงสัญญาณไซน์ความถี่ 1 กิโลเฮิรตซ์ ทำการเก็บผลการทดลอง ดังรูปที่ 3.61 และทำการบันทึกผลการทดลองระหว่างสัญญาณอินพุตเปรียบเทียบกับสัญญาณเอาต์พุต ได้ผลการทดลองดังรูปที่ 4.2




รูปที่ 3.61 การเก็บผลการทดลองของวงจรบันทึกและเล่นเสียง

2) การตรวจสอบสถานะของไอซีบันทึกเสียงและเล่นเสียง

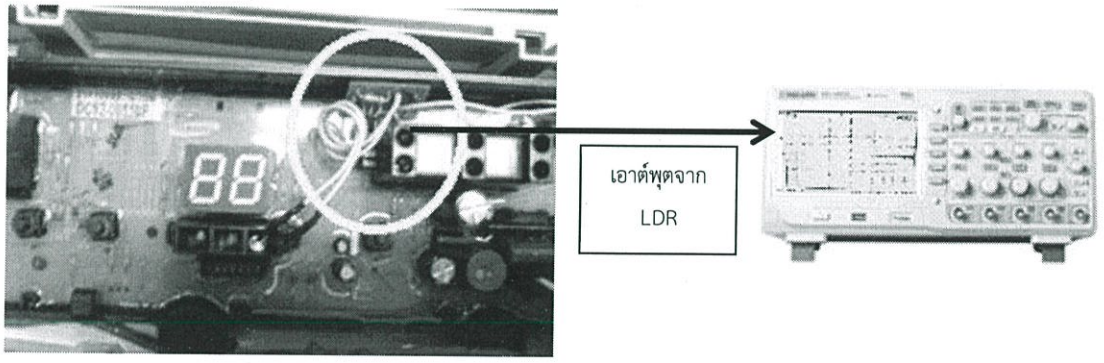
ผู้จัดทำปริญญานิพนธ์ได้ทำการวัดสัญญาณที่ขา INT/RDY ของไอซีบันทึกเสียงและเล่นเสียง ตรวจสอบสถานะของไอซีบันทึกและเล่นเสียงว่ามีการเล่นเสียงอยู่หรือไม่ และเพื่อนำมาใช้ในการเขียนคำสั่งให้กับไมโครคอนโทรเลอร์ต่อไป ผลการทดลองแสดงดังรูปที่ 4.3

### 3.3.3 การทำงานของจอแสดงผลแอลซีดี

จอแสดงผลแอลซีดีในปริญญานิพนธ์นี้เป็นแอลซีดีชนิด 16x2 เพื่อใช้สำหรับผู้ใช้บริการทราบข้อมูลขณะทำการกรอกรหัสผ่าน และจำนวนเหรียญที่หยอด ซึ่งได้ทำการเชื่อมต่อจอแสดงผลแอลซีดีใช้พอร์ต LCD ของไมโครคอนโทรลเลอร์ ARM7 โดยขั้นตอนการแสดงผลจะแบ่งออกเป็น 5 ขั้นตอน ได้แก่ 1. แสดงสถานะทั่วไป 2. เมื่อผู้ใช้บริการทำการกดรหัสผ่าน 3. เมื่อผู้ใช้บริการกดรหัสผ่านไม่ถูกต้อง 4. เมื่อผู้ใช้บริการกดรหัสผ่านถูกต้อง 5. เมื่อเครื่องซักผ้าเต็ม มีผลการทดลองดังรูปที่ 4.4-4.9

### 3.3.4 วงจรตรวจสอบสถานะโดยใช้แอลดีอาร์รับแสง

นำวงจรตรวจสอบสถานะโดยใช้แอลดีอาร์รับแสงไปใช้กับแสงจากแอลดีอาร์สถานะของเครื่องซักผ้า ทำการเก็บผลทดลอง ดังรูปที่ 3.62 และมีผลการทดลองดังรูปที่ 4.10 และรูปที่ 4.11



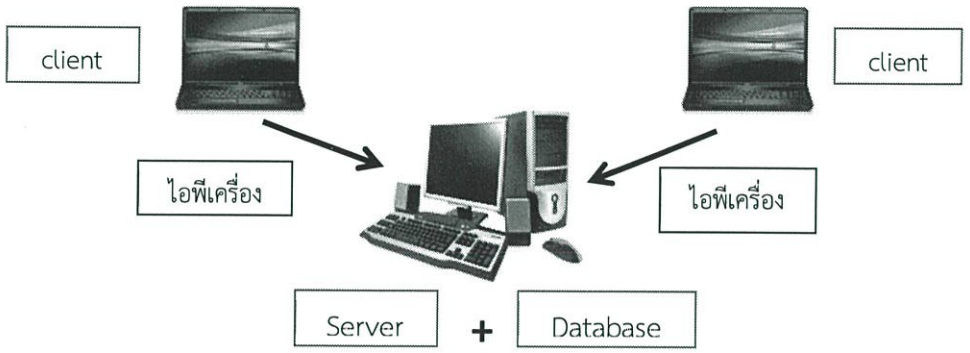

รูปที่ 3.62 การเก็บผลแรงดันที่ขาคอลเลคเตอร์เมื่อเครื่องซักผ้าทำงาน

### 3.3.5 วงจรตรวจสอบสถานะโดยใช้ออปโต้คัปเปลอร์

วงจรตรวจสอบสถานะของการทำงานของอุปกรณ์ไฟฟ้าเป็นการตรวจสอบสถานะของการทำงานของอุปกรณ์ โดยใช้ระดับแรงเอาต์พุตจากออปโต้คัปเปลอร์ไปเป็นสัญญาณอินพุตเพื่อต่อเข้ากับบอร์ดไมโครคอนโทรลเลอร์ ARM7 ในการทำวงจรเปิด/ปิดและวงจรตรวจสอบสถานะของการทำงานของอุปกรณ์ไฟฟ้า มีผลการทดลองดังรูปที่ 4.12 และรูปที่ 4.13

### 3.3.6 การทำงานของส่วนการรับส่งข้อมูล

ส่วนการรับส่งข้อมูลมีโปรแกรมกลางเพิ่มเข้ามา ซึ่งคือโปรแกรม QT Creator ทำหน้าที่เป็นตัวกลางการเชื่อมต่อระหว่างไมโครคอนโทรเลอร์กับเซิร์ฟเวอร์ ซึ่งสามารถรับส่งข้อมูลผ่านไอพีของเครื่องเซิร์ฟเวอร์หรือส่งผ่านไอพีภายนอกได้ด้วยเช่นกัน จะแสดงการทำงานของส่วนรับส่งข้อมูลดังรูปที่ 3.63 มีผลการทดลองดังรูปที่ 4.14



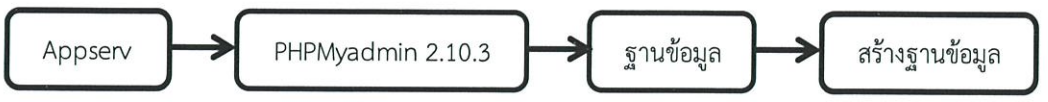

รูปที่ 3.63 การทำงานของส่วนการรับส่งข้อมูล

### 3.3.7 ระบบการยืนยันตนด้วยรหัสผ่าน

ขั้นตอนการยืนยันรหัสผ่านสำหรับผู้ใช้งานนั้น จะใช้วิธีการรับส่งค่ารหัสผ่านระหว่างหน้าเว็บและไมโครคอนโทรลเลอร์โดยมีเซิร์ฟเวอร์เป็นตัวกลาง เมื่อผู้ใช้ได้รับรหัสผ่านจะมีเวลาคงเหลือของการใช้เครื่องซักผ้าแสดงที่หน้าเว็บ สำหรับรหัสผ่านผู้ใช้บริการที่ทำการจองต้องนำรหัสผ่านที่ได้จากหน้าเว็บ ไปยืนยันที่หน้าตู้ควบคุมเพื่อยืนยันตัวตนการใช้งาน และทำการหยอดเหรียญตาม เพื่อเริ่มการใช้งานต่อไป ผลการทดลอง ดังรูปที่ 4.16 - 4.18 ตามลำดับ สำหรับผู้ใช้บริการที่เป็นสมาชิก แล้วทำการจองจะได้รับรหัสผ่าน และอีเมล์แจ้งเตือนการใช้งานเครื่องซักผ้าในระยะเวลา 15, 60 นาที และ 1 วัน ก่อนการใช้งาน ขึ้นอยู่กับการเลือกเวลาการจองของผู้ใช้บริการ ดังรูปที่ 4.19

### 3.3.8 การทำงานของฐานข้อมูล

การสร้างฐานข้อมูล(Database) โดยรวมเป็นดังรูปที่ 3.64 ค่าที่รับมานั้นจะถูกส่งไปอยู่ในฐานข้อมูลของเซิร์ฟเวอร์ มีผลการสร้างฐานข้อมูลดังรูปที่ 4.20-4.22



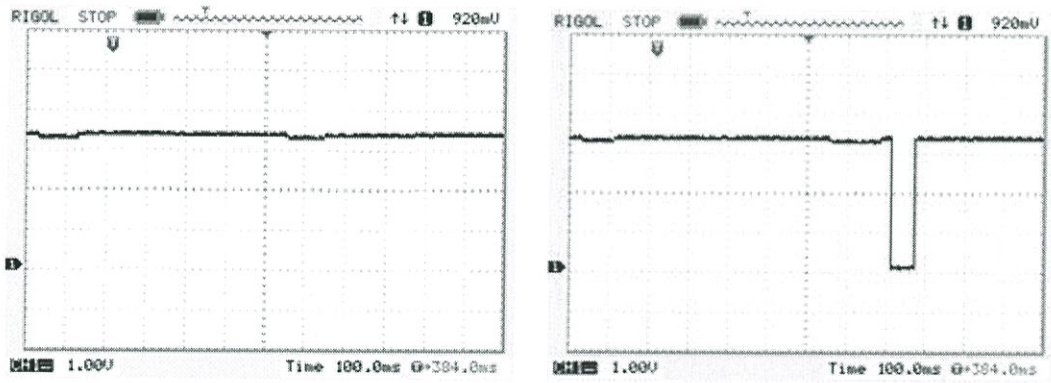

รูปที่ 3.64 การสร้างฐานข้อมูลโดยรวม

# บทที่ 4

# ผลการทดลอง

## 4.1 สัญญาณจากเครื่องหยอดเหรียญ

ปริญญานิพนธ์นี้ได้เลือกใช้โหมดการทำงานของเครื่องหยอดเหรียญเป็นโหมด N.O. (Normally Open) เมื่อไม่มีการหยอดเหรียญจะมีสถานะเป็น 3.3 โวลต์ (High) เมื่อมีการหยอดเหรียญ (และรับเหรียญ) จะมีสถานะเป็น 0 โวลต์ (Low) แล้วจึงเปลี่ยนสถานะกลับเป็น 3.3 โวลต์ (High) ดังเดิม ทำการเก็บผลการทดลอง ดังรูปที่ 3.60 และผลการทดลอง ดังรูปที่ 4.2



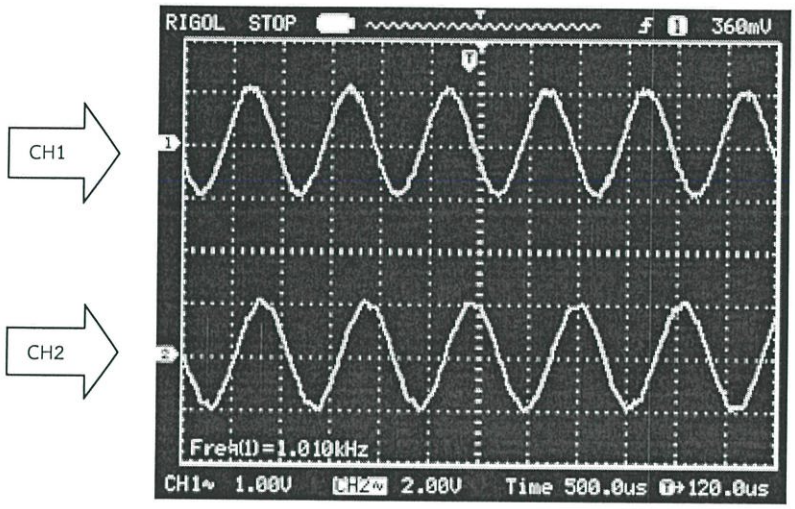

(ก) (ข)

รูปที่ 4.1 สัญญาณจากเครื่องหยอดเหรียญในโหมด N.O. (Normally Open)
(ก) เมื่อไม่มีการหยอดเหรียญ
(ข) เมื่อมีการหยอดเหรียญ

## 4.2 การทำงานของวงจรบันทึกและเล่นเสียง

### 4.2.1 การป้อนสัญญาณไซน์ที่อินพุต เพื่อเปรียบเทียบกับสัญญาณที่เอาต์พุต

ผู้จัดทำปริญญานิพนธ์ได้ทำการบันทึกเสียงลงในไอซีบันทึกและเล่นเสียง โดยเสียงที่บันทึกเป็นเสียงสัญญาณไซน์ ความถี่ 1 กิโลเฮิรตซ์ ทำการเก็บผลการทดลอง ดังรูปที่ 3.61 และทำการบันทึกผลการทดลองระหว่างสัญญาณอินพุตเปรียบเทียบกับสัญญาณเอาต์พุต ได้ผลการทดลองดังรูปที่ 4.2

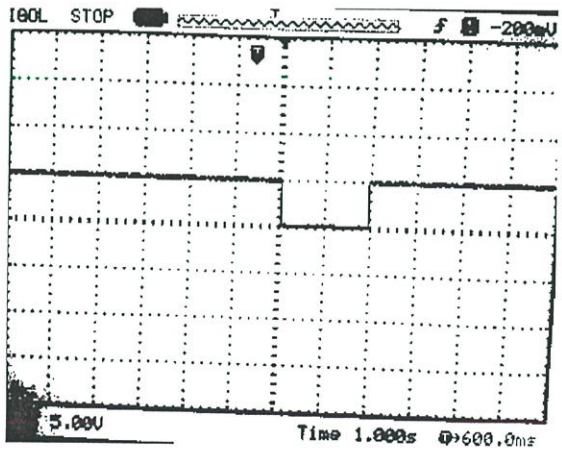



รูปที่ 4.2 สัญญาณไซน์เปรียบเทียบระหว่างอินพุตและเอาต์พุต

CH1: สัญญาณอินพุต ป้อนเข้าที่คอนเดนเซอร์ไมค์ของไอซีบันทึกและเล่นเสียง 1 กิโลเฮิรตซ์
CH2: สัญญาณเอาต์พุต เป็นสัญญาณที่ออกจากลำโพง และผ่านวงจรกรองความถี่ต่ำผ่าน

### 4.2.2 การตรวจสอบสถานะของไอซีบันทึกเสียงและเล่นเสียง

ผู้จัดทำปริญญานิพนธ์ได้ทำการบันทึกเสียงลงในไอซีบันทึกและเล่นเสียง และกดสวิตซ์เพื่อทำการเล่นเสียงที่ถูกบันทึกไว้ออกทางลำโพง โดยช่วงที่มีการเล่นเสียงเมื่อวัดสัญญาณที่ขา INT/RDY สัญญาณจะเป็นลอจิก “0” คือ ช่วงที่มีการบันทึกเสียง และลอจิก “1” คือ ช่วงที่ไม่มีการบันทึกเสียง ผลการทดลองแสดงดังรูปที่ 4.3

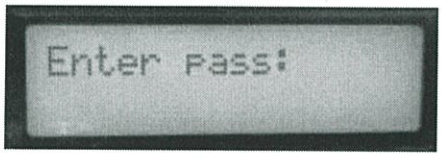



รูปที่ 4.3 สัญญาณแสดงสถานะของไอซีบันทึกและเล่นเสียง

## 4.3 การทำงานของจอแสดงผลแอลซีดี

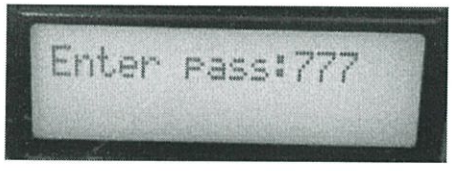
เมื่อทำการเชื่อมต่อจอแสดงผลแอลซีดี ใช้พอร์ต LCD ของไมโครคอนโทรลเลอร์ ARM7 โดยขั้นตอนการแสดงผลจะแบ่งออกเป็น

ขั้นตอนที่ 1 แสดงสถานะทั่วไป (ยังไม่มีการใช้งาน) จะแสดงข้อความ “Enter pass:” ดังรูปที่ 4.4




รูปที่ 4.4 สถานะทั่วไป (ยังไม่มีการใช้งาน)

ขั้นตอนที่ 2 เมื่อผู้ใช้บริการทำการกดรหัสผ่าน จะแสดงรหัสที่ผู้ใช้กด 4 หลัก ดังรูปที่ 4.5




รูปที่ 4.5 แสดงรหัสผ่านที่ผู้ใช้กดจำนวน 4 หลัก

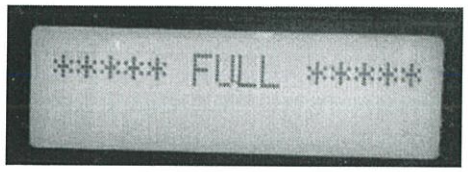
ขั้นตอนที่ 3 เมื่อผู้ใช้บริการกดรหัสผ่านไม่ถูกต้อง จะขึ้นข้อความ “Wrong password” ดังรูปที่ 4.6

รูปที่ 4.6 ข้อความเมื่อผู้ใช้บริการกดรหัสผ่านไม่ถูกต้อง

ขั้นตอนที่ 4 เมื่อผู้ใช้บริการกดรหัสผ่านถูกต้อง จะเข้าสู่ขั้นตอนการหยอดเหรียญ โดยหน้าจอจะแสดงค่าเหรียญที่หยอด ดังรูปที่ 4.7

รูปที่ 4.7 แสดงจำนวนเหรียญที่ผู้ใช้บริการหยอด

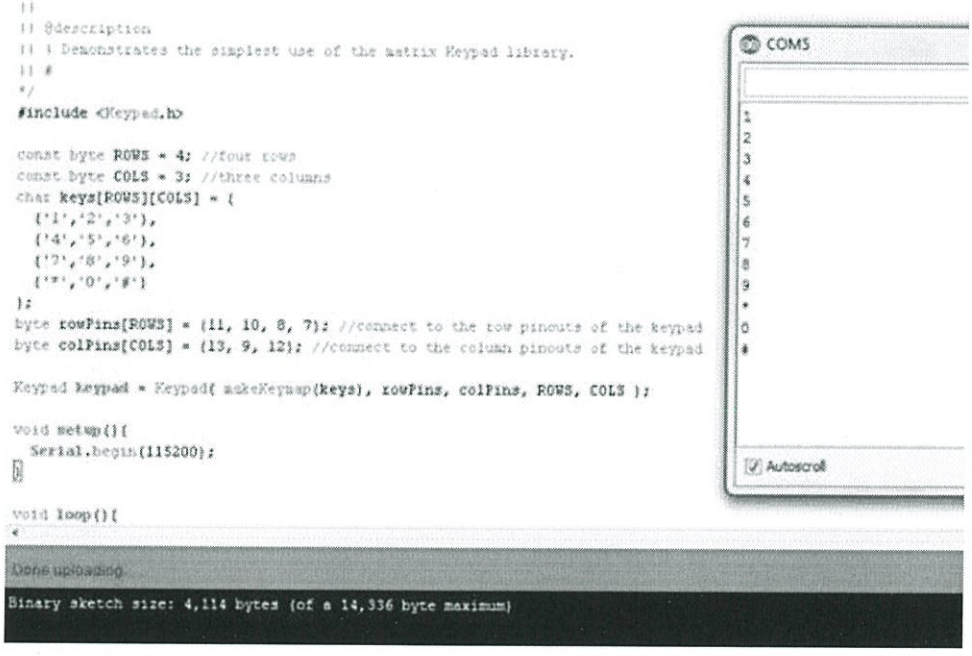
ขั้นตอนที่ 5 แสดงสถานะเมื่อไม่มีเครื่องซักผ้าเครื่องใดว่าง จะแสดงข้อความ “FULL” ดังรูปที่ 4.8




รูปที่ 4.8 แสดงสถานะเมื่อไม่มีเครื่องซักผ้าเครื่องใดว่าง

## 4.4 การรับค่ารหัสผ่านโดยใช้คีย์แพด

ทดลองรับค่ารหัสผ่านโดยคีย์แพด และแสดงตัวเลขผ่านทางพอร์ต UART2 ของไมโครคอนโทรลเลอร์ ARM7 และแสดงผลผ่าน Serial monitor พบว่ารหัสผ่านที่กดมีค่าตรงกับค่าที่แสดงผล ดังรูปที่ 4.9

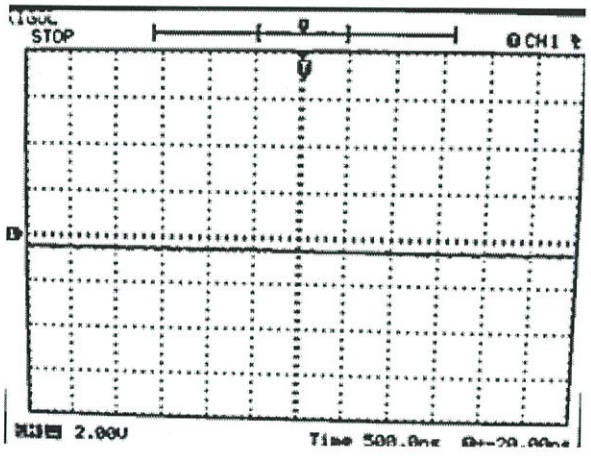



รูปที่ 4.9 การกดตัวเลขผ่านคีย์แพดผ่าน Serial monitor

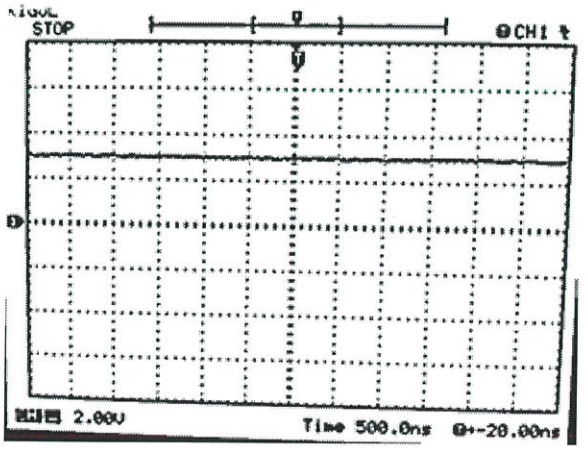
## 4.5 การทำงานของวงจรตรวจสอบสถานะของเครื่องซักผ้า

### 4.5.1 วงจรตรวจสอบสถานะโดยใช้แอลดีอาร์รับแสง

นำวงจรตรวจสอบสถานะโดยใช้แอลดีอาร์รับแสงไปใช้กับแสงจากแอลดีอาร์สถานะของเครื่องซักผ้า เมื่อเครื่องซักผ้าทำงาน วัดแรงดันที่ขาคอลเลคเตอร์ได้ -0.4 โวลต์ และเมื่อเครื่องซักผ้าสิ้นสุดกระบวนการซัก วัดแรงดันที่ขาคอลเลคเตอร์ได้ 3 โวลต์ ซึ่งจากการทดลอง จะนำแรงดันที่ได้นี้ไปต่อเข้ากับไมโครคอนโทรลเลอร์หลัก โดยแบ่งเป็นลอจิกได้ 2 ระดับ คือ -0.4 โวลต์ แทนลอจิก "0" และ 3 โวลต์ แทนลอจิก "1" ทำการเก็บผลทดลอง ดังรูปที่ 3.62 และมีผลการทดลองดังรูปที่ 4.10 และรูปที่ 4.11



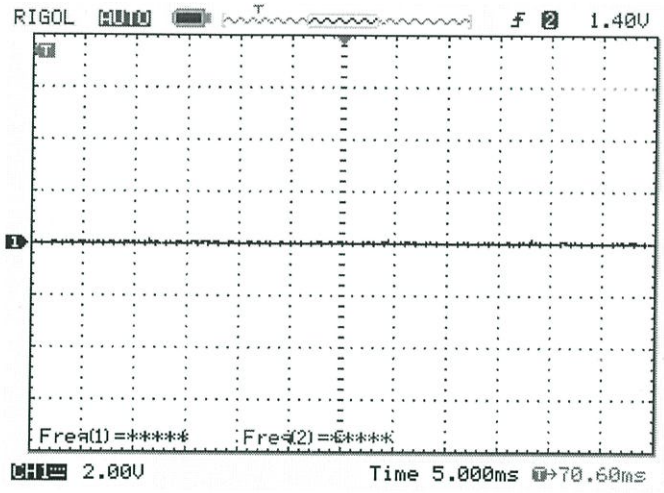

รูปที่ 4.10 แรงดันวัดที่ขาคอลเลคเตอร์เมื่อเครื่องซักผ้าทำงาน (ไฟติด) หรือ -0.4 โวลต์



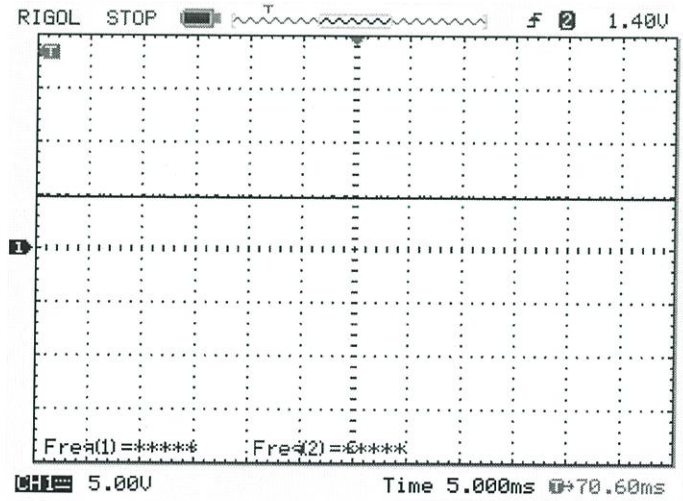

รูปที่ 4.11 แรงดันวัดที่ขาคอลเลคเตอร์เมื่อเครื่องซักผ้าสิ้นสุดการทำงาน (ไฟดับ) หรือ 3 โวลต์

### 4.5.2 วงจรตรวจสอบสถานะโดยใช้ออปโต้คัปเปลอร์

การทดลองวัดแรงดันเอาต์พุตก่อนและหลังการจ่ายกระแสวงจรตรวจสอบสถานะโดยใช้ออปโต้คัปเปลอร์ ได้ผลการทดลอง ดังรูปที่ 4.12 และรูปที่ 4.13




รูปที่ 4.12 ระดับแรงดันไฟฟ้าจากออปโต้คัปเปลอร์ 0 โวลต์ (สถานะไม่ว่าง)




รูปที่ 4.13 ระดับแรงดันไฟฟ้าจากออปโต้คัปเปลอร์ 5 โวลต์ (สถานะว่าง)

## 4.6 การทำงานของส่วนการรับส่งข้อมูล

การรับส่งข้อมูลเป็นรูปแบบข้อมูลแบบตัวอักษรทั่วไป กำหนดให้ไมโครคอนโทรลเลอร์ส่งข้อความ “T101 Washing Test” ทดสอบรับข้อมูลที่เซิร์ฟเวอร์ ดังรูปที่ 4.14 (ก) และ (ข) และตรวจจับแพคเกจข้อมูลโดยใช้โปรแกรมวายชาร์ก กำหนดต้นทางเป็นไอพีแอดเดรสของไมโครคอนโทรลเลอร์ คือ 192.168.1.3 และปลายทางเป็นไอพีแอดเดรสของเซิร์ฟเวอร์ คือ 192.168.1.134 ดังรูปที่ 4.15 ซึ่งจะเห็นว่าข้อมูลมีการส่งจากไมโครคอนโทรลเลอร์ไปยังเซิร์ฟเวอร์ได้

```
34
35  void InitOsc(void);
36  void InitPorts(void);
37  void Send(void);
38  //void InsertDynamicValues(void);
39  //void get_req(void);
40  //void InsertLEDStatus(void);
41 /*unsigned int GetAD7Val(void);
42  unsigned int GetTempVal(void);*/
43  const unsigned char GetResponse[] =
44 {
45    "T101 Washing Test\r\n"
46  };
47  //unsigned char HTTPStatus;
48  //#define HTTP_SEND_PAGE              0x01
49
```

(ก)

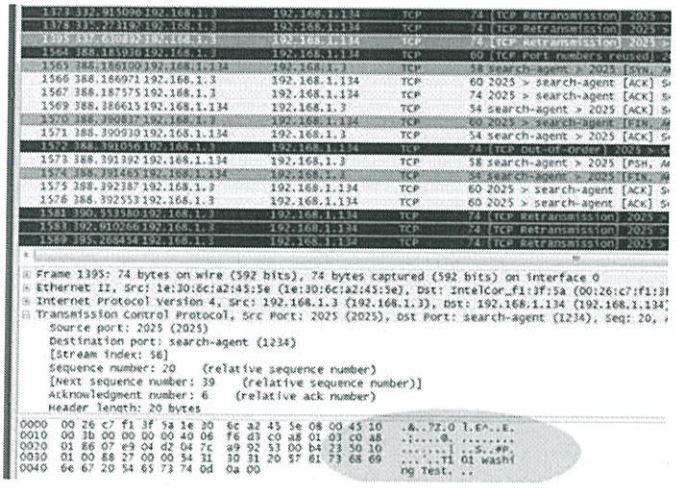
```
"
"192.168.1.3"
"Test
"
"192.168.1.3"
"T101 Washing Test
"
"192.168.1.3"
"T101 Washing Test
"
"192.168.1.3"
```

(ข)

รูปที่ 4.14 การทำงานการรับส่งข้อมูลระหว่างไมโครคอนโทรลเลอร์กับเซิร์ฟเวอร์

(ก) ข้อมูลที่ไมโครคอนโทรลเลอร์ต้องการส่ง “T101 Washing Test”

(ข) ข้อมูลที่เซิร์ฟเวอร์รับได้แสดงผลผ่านโปรแกรม QT Creator

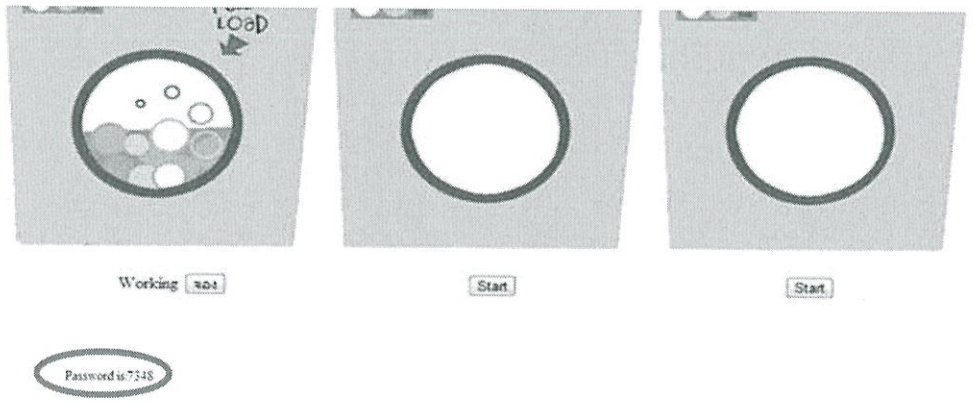


รูปที่ 4.15 การตรวจจับการรับส่งข้อมูลด้วยโปรแกรมวายชาร์ก

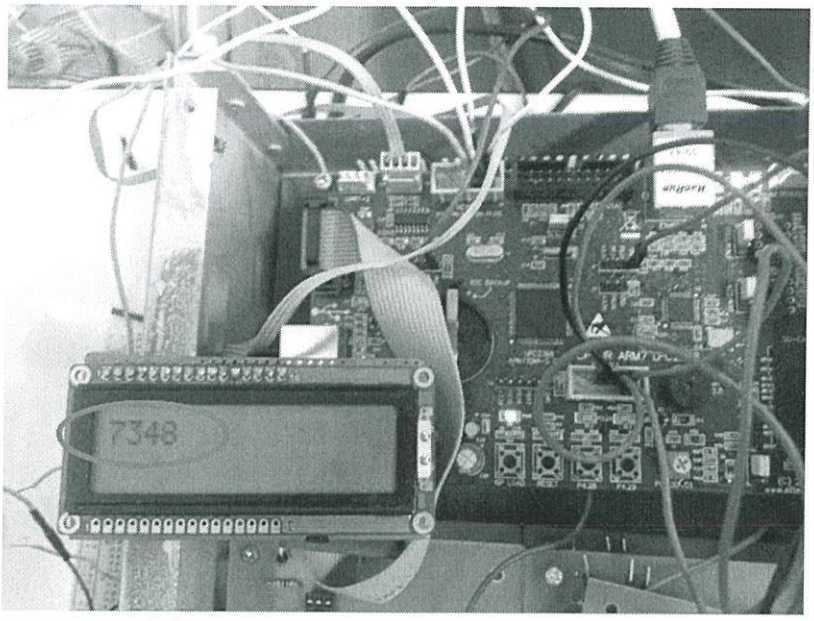
จากรูปที่ 4.15 อธิบายได้ว่า มีการรับส่งข้อมูล “T101 Washing Test” ระหว่างไมโครคอนโทรลเลอร์และเซิร์ฟเวอร์ได้ โดยใช้โปรแกรมวายชาร์กในการตรวจจับ

## 4.7 ระบบการยืนยันตนด้วยรหัสผ่าน

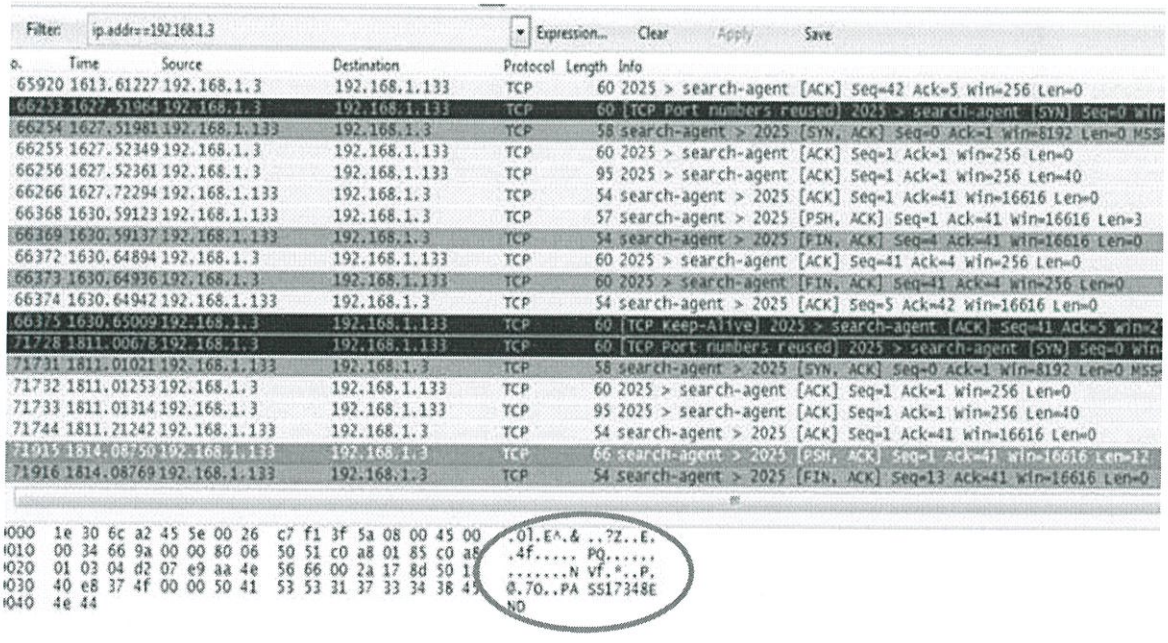
เมื่อผู้ใช้บริการทำการจองเรียบร้อยแล้ว ผู้ใช้บริการที่ทำการจองต้องนำรหัสผ่านที่ได้จากหน้าเว็บ ดังรูปที่ 4.16 ไปยืนยันที่หน้ากล่องควบคุมเพื่อยืนยันตัวตนการใช้งาน ดังรูปที่ 4.17 และทำการหยอดเหรียญตาม เพื่อเริ่มการใช้งานต่อไป รูปที่ 4.18 เป็นการตรวจจับการส่งรหัสผ่านด้วยโปรแกรมวายชาร์กแสดงการรับส่งค่ารหัสผ่านระหว่างหน้าเว็บ และไมโครคอนโทรลเลอร์




รูปที่ 4.16 หน้าเว็บแสดงรหัสผ่าน เมื่อมีผู้ใช้บริการทำการจอง




รูปที่ 4.17 จอแอลซีดีที่เชื่อมต่อกับไมโครคอนโทรลเลอร์ เพื่อยืนยันการใช้งานจากผู้ใช้บริการ



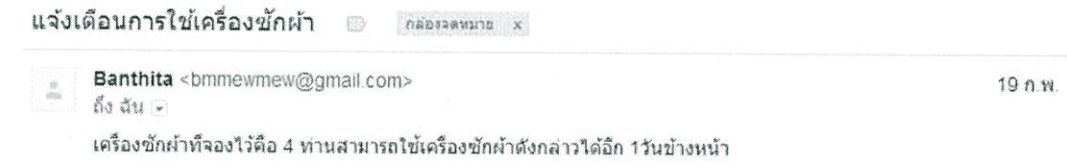

รูปที่ 4.18 โปรแกรมวายชาร์กตรวจจับการรับส่งค่ารหัสผ่าน
ระหว่างหน้าเว็บและไมโครคอนโทรลเลอร์

## 4.8 การส่งอีเมล์แจ้งเตือนผู้ใช้งาน

การส่งอีเมล์แจ้งเตือนผู้ใช้งาน ระบบออกแบบให้ส่วนของเซิร์ฟเวอร์เป็นตัวส่งอัตโนมัติ โดยให้ส่งที่เวลาต่างๆ ดังนี้ ก่อนใช้งานเครื่องซักผ้า 15 นาที, ก่อนใช้งานเครื่องซักผ้า 60 นาที และก่อนใช้งานเครื่องซักผ้า 1 วัน มีผลการทดลองดังรูปที่ 4.19 (ก), (ข) และ (ค)

(ก)

(ข)

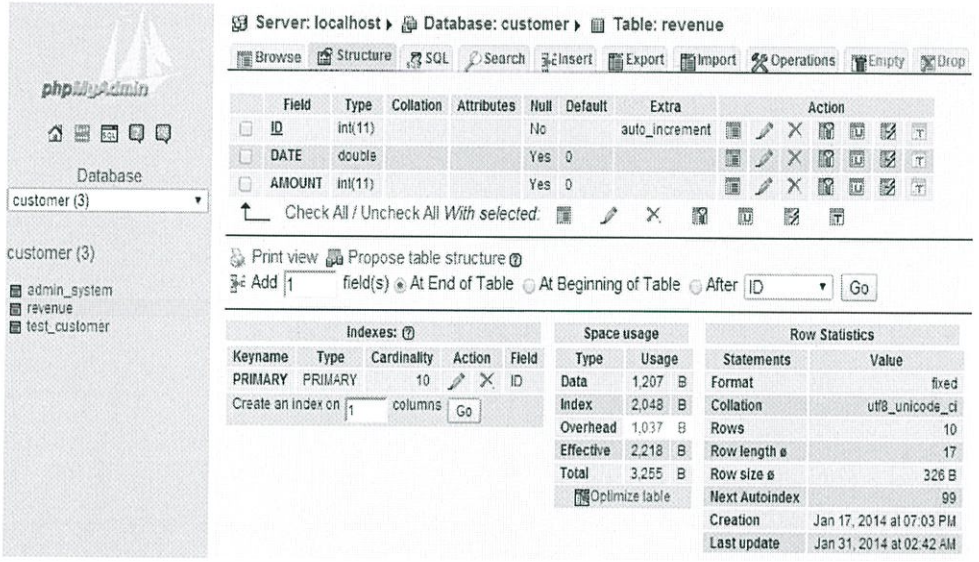



(ค)

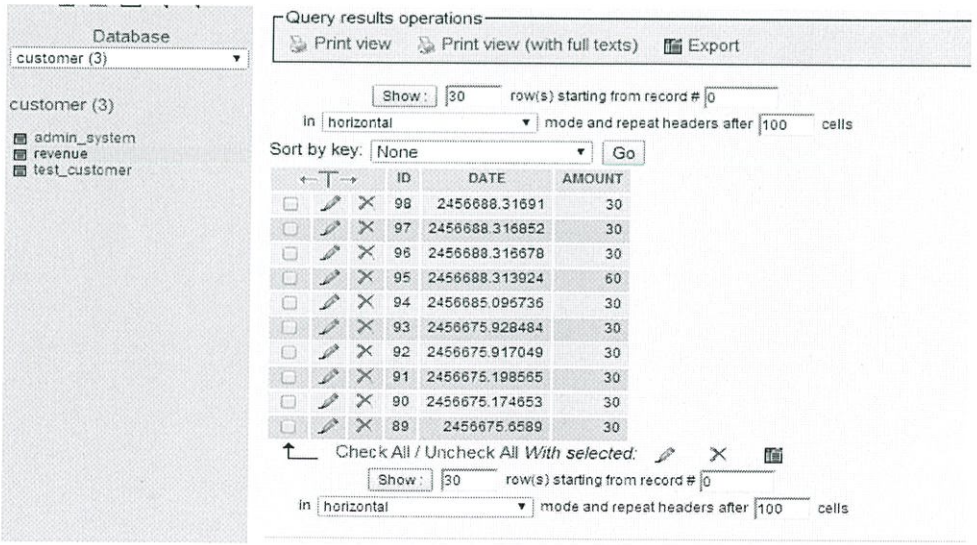
รูปที่ 4.19 การส่งอีเมล์แจ้งเตือนผู้ใช้งาน
(ก) การส่งอีเมล์แจ้งเตือนผู้ใช้งานก่อนการใช้งาน 15 นาที
(ข) การส่งอีเมล์แจ้งเตือนผู้ใช้งานก่อนการใช้งาน 60 นาที
(ค) การส่งอีเมล์แจ้งเตือนผู้ใช้งานก่อนการใช้งาน 1 วัน

## 4.9 การทำงานของฐานข้อมูล

ฐานข้อมูลที่ถูกสร้างชื่อว่า revenue ประกอบด้วย 3 ฟิลด์ ได้แก่ ID, DATE และ AMOUNT ดังรูปที่ 4.20 โดยข้อมูล DATE และ AMOUNT เป็นข้อมูลที่รับเข้ามาจากไมโครคอนโทรลเลอร์ ดังรูปที่ 4.21 และฐานข้อมูล revenue แสดงเป็นตารางในหน้าเว็บ ดังรูปที่ 4.22




รูปที่ 4.20 ฐานข้อมูล revenue พร้อมฟิลด์ต่างๆ




รูปที่ 4.21 ข้อมูลที่เข้ามาในฐานข้อมูล revenue

คำนวณรายได้

| ID | DATE | DATETIME | AMOUNT |
|---|---|---|---|
| 118 | 2456720.101366 | 3/3/2014 | 30 |
| 117 | 2456720.097419 | 3/3/2014 | 30 |
| 116 | 2456720.097014 | 3/3/2014 | 30 |
| 115 | 2456720.094213 | 3/3/2014 | 30 |
| 114 | 2456720.091887 | 3/3/2014 | 30 |
| 113 | 2456719.949641 | 3/2/2014 | 90 |

รูปที่ 4.22 ฐานข้อมูล revenue แสดงเป็นตารางในหน้าเว็บ

## 4.10 การคำนวณรายได้

การคำนวณรายได้สามารถเลือกวันที่ต้องการได้ โดยมีวันเริ่มที่เริ่มต้นและวันที่สิ้นสุดให้เลือก วันที่เลือกจะเป็นวันที่ปกติ แต่การคำนวณนั้นจะต้องเปลี่ยนให้เป็น Julian Day ให้สอดคล้องกับค่าที่รับมาในฐานข้อมูล หน้าเว็บเพจแสดงการคำนวณรายได้ของผู้ให้บริการ ดังรูปที่ 4.23

Balance is 330 bath

Startday: วว/ดด/ปปปป Endday: วว/ดด/ปปปป ส่ง

Log out

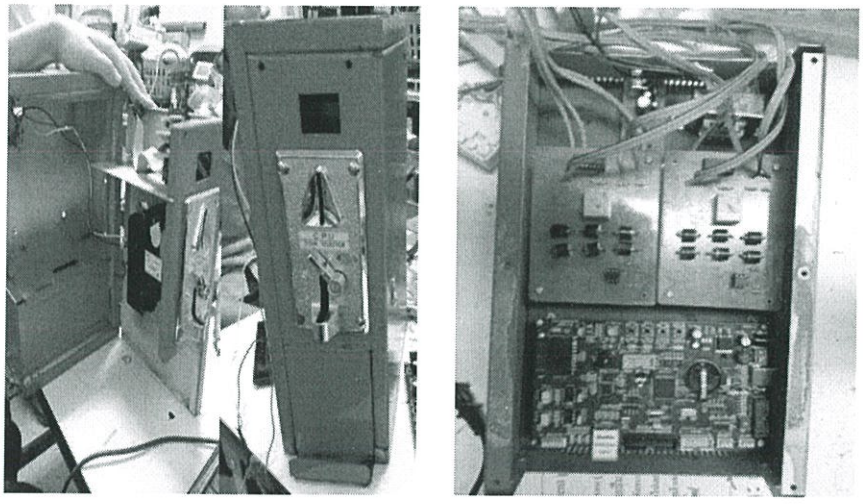
เพิ่มเงินสมาชิก

| ID | DATE | DATETIME | AMOUNT |
|---|---|---|---|
| 118 | 2456720.101366 | 3/3/2014 | 30 |
| 117 | 2456720.097419 | 3/3/2014 | 30 |
| 116 | 2456720.097014 | 3/3/2014 | 30 |
| 115 | 2456720.094213 | 3/3/2014 | 30 |
| 114 | 2456720.091887 | 3/3/2014 | 30 |
| 113 | 2456719.949641 | 3/2/2014 | 90 |
| 121 | 2456720.895521 | 3/3/2014 | 30 |
| 120 | 2456720.102002 | 3/3/2014 | 30 |
| 119 | 2456720.101713 | 3/3/2014 | 30 |

รูปที่ 4.23 การคำนวณรายได้

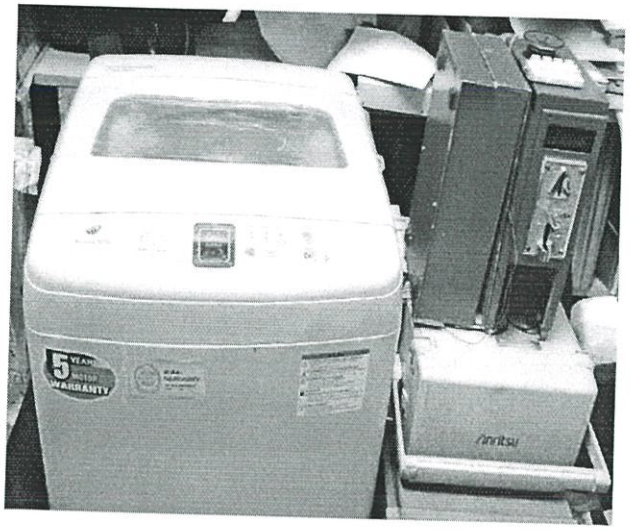
## 4.11 กล่องรับเหรียญและกล่องควบคุม

ผู้จัดทำได้ออกแบบกล่องควบคุม ดังรูปที่ 4.25 โดยแบ่งเป็นกล่องสำหรับรับเหรียญ 1 กล่อง และกล่องควบคุมสำหรับใส่วงจรอีก 1 กล่องแยกออกจากกัน กล่องรับเหรียญจะประกอบด้วย ตัวเครื่องหยอดเหรียญ, สวิตช์คีย์แพดขนาด 4x3 และจอแอลซีดีขนาด 16x2 สำหรับแสดงผล ข้อความและจำนวนเหรียญที่ผู้ใช้บริการหยอด ดังรูปที่ 4.24 (ก) และกล่องสำหรับใส่วงจรควบคุม ซึ่ง ภายในบรรจุบอร์ดไมโครคอนโทรลเลอร์, วงจรเปิดปิดและตรวจสอบสถานะและหม้อแปลง 220/12 โวลต์ ดังรูปที่ 4.24 (ข)



(ก) (ข)

รูปที่ 4.24 อุปกรณ์ภายในกล่องควบคุม
(ก) กล่องสำหรับรับเหรียญ
(ข) กล่องสำหรับใส่วงจรควบคุม




รูปที่ 4.25 กล่องควบคุม

## 4.12 การทดสอบการใช้งานระบบ

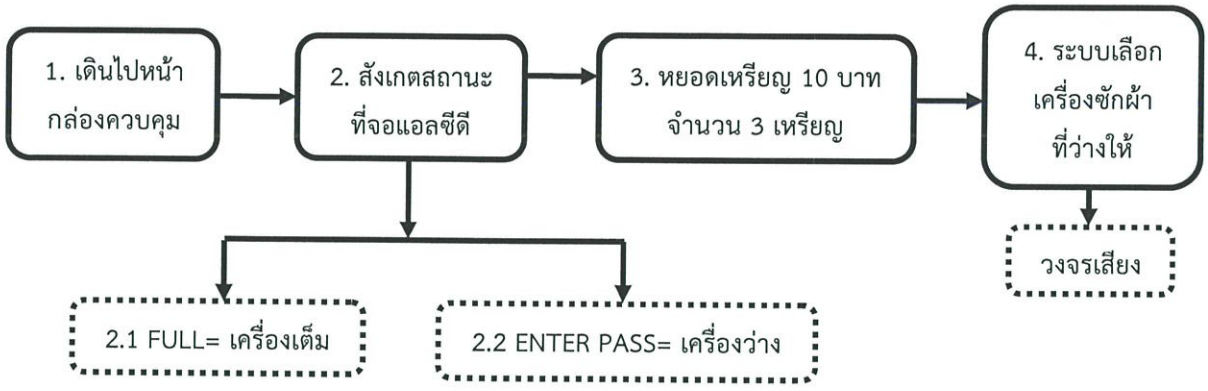
ทดสอบการใช้งานระบบ เป็นการทดสอบใช้งานระบบควบคุมเครื่องซักผ้าหลายเครื่องในธุรกิจซักผ้าแบบบริการตัวเองแบบเต็มรูปแบบ คือทดสอบการใช้งานในกรณีต่างๆที่เป็นไปได้ทั้งหมด โดยแบ่งการใช้งานเป็น 3 กรณี ดังนี้

1. สำหรับผู้ใช้บริการที่เดินเข้ามาใช้งานกล่องควบคุม
2. สำหรับผู้ใช้บริการที่ต้องการทำการจอง กรณีของผู้ที่ไม่ได้เป็นสมาชิก
3. สำหรับผู้ใช้บริการที่ต้องการทำการจอง กรณีผู้ที่เป็นสมาชิก

สามารถอธิบายในแต่ละกรณีได้ดังนี้

### 4.12.1 สำหรับผู้ใช้บริการที่เดินเข้ามาใช้งานหน้ากล่องควบคุมโดยตรง (ไม่ผ่านระบบการจองใดๆ)

ผู้ใช้บริการสามารถเดินไปหน้ากล่องควบคุมและสามารถหยอดเหรียญได้เลย ระบบจะเลือกเครื่องซักผ้าที่ว่างให้อัตโนมัติ มีบล็อกไดอะแกรมแสดงการทำงานดังรูปที่ 4.26



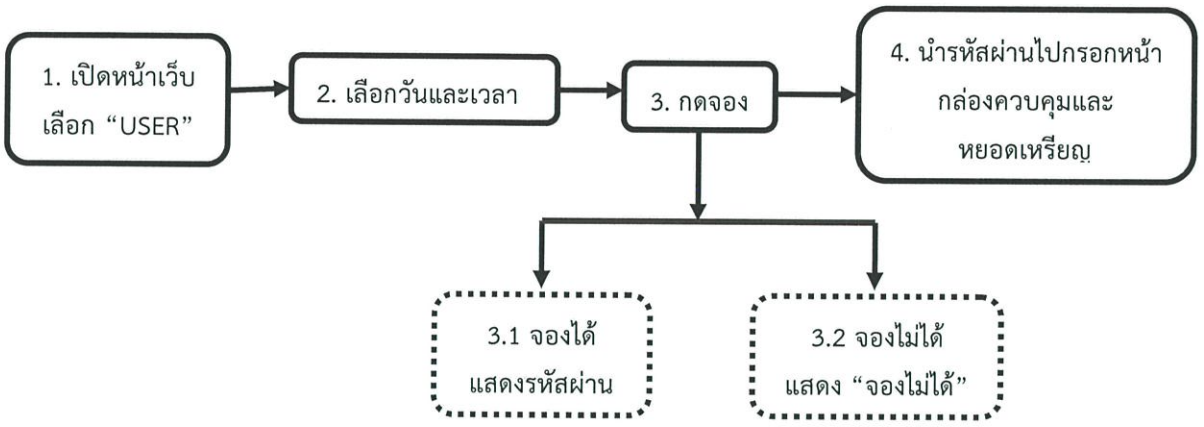

รูปที่ 4.26 บล็อกไดอะแกรมแสดงการทำงานของระบบสำหรับผู้ใช้บริการ
ที่เดินเข้ามาใช้งานที่กล่องควบคุมโดยตรง

จากรูปที่ 4.26 อธิบายได้ว่า

- หมายเลข 1 คือ การเดินไปหน้ากล่องควบคุมโดยไม่ได้ทำการจองใดๆ ทั้งสิ้น แสดงได้ดังรูปที่ 4.26
- หมายเลข 2 คือ การสังเกตสถานะที่จอแอลซีดี ดังหมายเลข 2.1 และ 2.2
- หมายเลข 2.1 คือ ถ้าแสดงคำว่า FULL คือเครื่องซักผ้าเต็มทั้งหมดหมด เต็มในที่นี้ คือ เต็มที่รวมการจองในช่วงเวลานั้นๆ ด้วย แสดงได้ดังรูปที่ 4.8
- หมายเลข 2.2 คือ ถ้าแสดงข้อความ ENTER PASS คือมีเครื่องซักผ้าว่าง พร้อมที่ถูกใช้งาน รอการหยอดเหรียญหรือการกดรหัสผ่าน แสดงได้ดังรูปที่ 4.4
- หมายเลข 3 คือ การหยอดเหรียญ 10 บาท จำนวน 3 เหรียญ แสดงได้ดังรูปที่ 4.7

### 4.12.2 สำหรับผู้ใช้บริการที่ต้องการทำการจอง กรณีของผู้ที่ไม่ได้เป็นสมาชิก

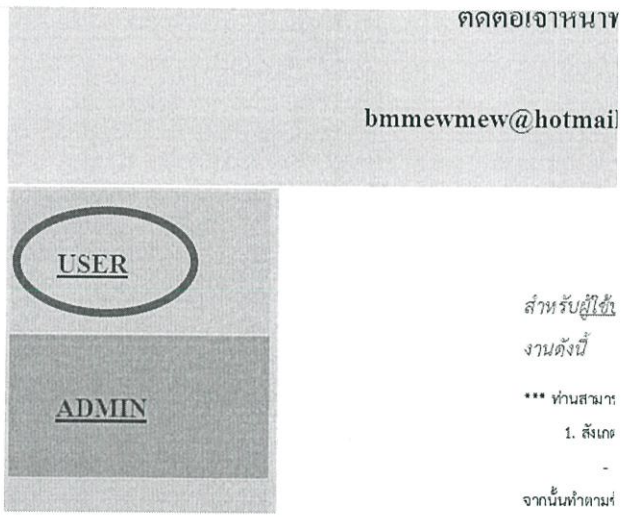
กรณีผู้ใช้บริการทำการจองผ่านหน้าเว็บเพจ แต่ไม่ได้เป็นสมาชิก เมื่อต้องการใช้งานเครื่องซักผ้า ให้นำรหัสผ่านที่ได้รับจากการจองไปยืนยันตนหน้ากล่องควบคุมและหยอดเหรียญ 10 บาท จำนวน 3 เหรียญ มีบล็อกไดอะแกรมแสดงขั้นตอนดังรูปที่ 4.27




รูปที่ 4.27 บล็อกไดอะแกรมแสดงการทำงานของระบบสำหรับผู้ใช้บริการที่ต้องการทำการจอง กรณีของผู้ที่ไม่ได้เป็นสมาชิก

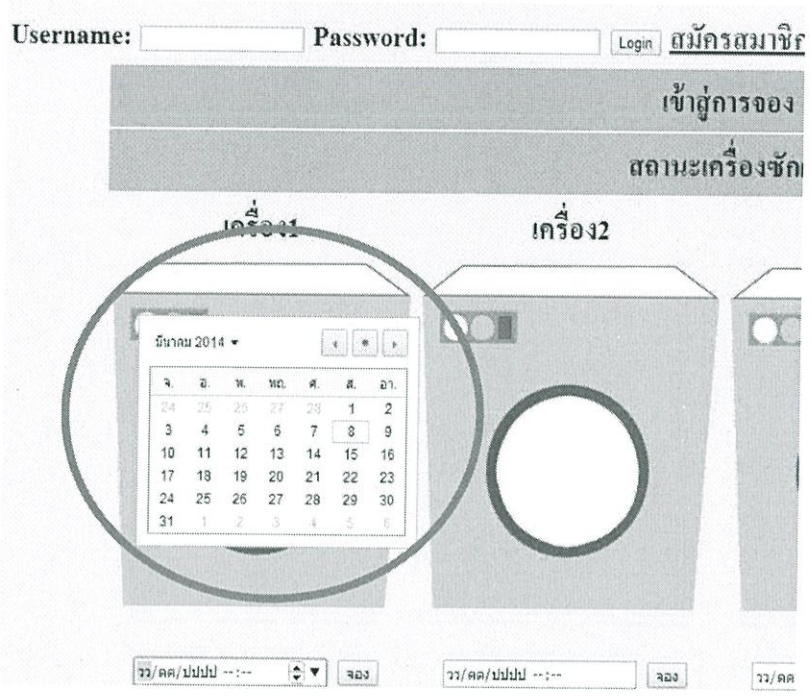
จากรูปที่ 4.27 อธิบายได้ว่า

- หมายเลข 1 คือการเปิดหน้าเว็บเพจ จะมี USER ให้ผู้ใช้บริการเลือก ดังรูปที่ 4.28



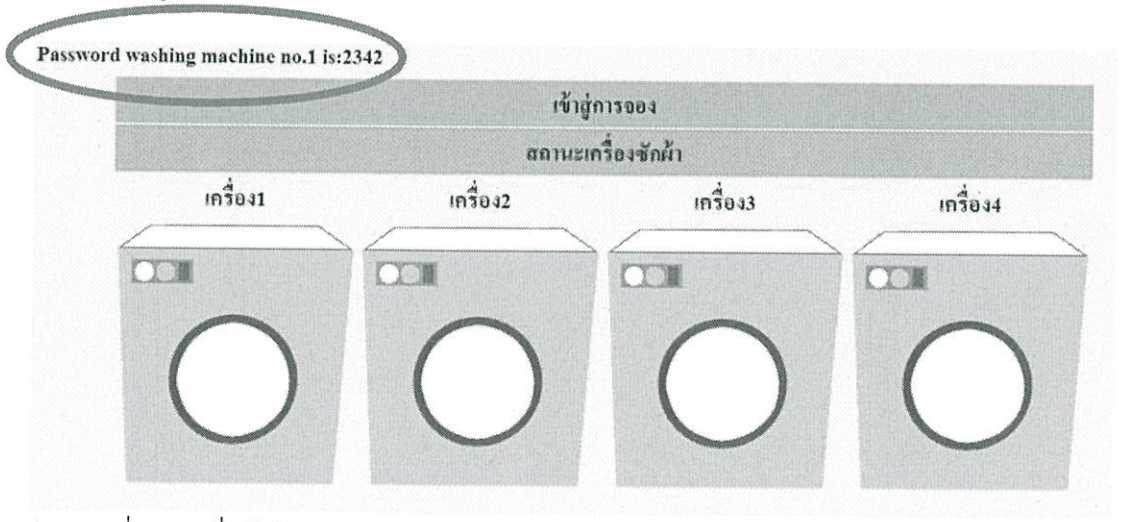

รูปที่ 4.28 หน้าเว็บเพจ

- หมายเลข 2 คือ การเลือกวันและเวลา โดยมีรูปแบบวันและเวลาให้เลือก ดังรูปที่ 4.29



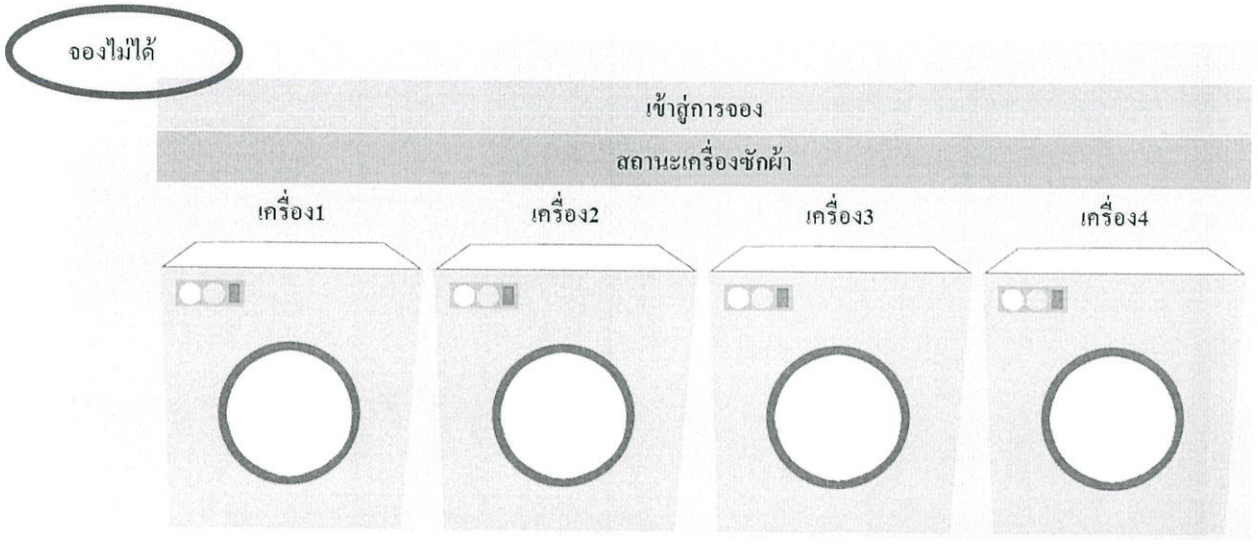

รูปที่ 4.29 หน้าเว็บเพจการจอง แสดงการเลือกวันเวลาในการจอง

- หมายเลข 3.1 คือ เมื่อผู้ใช้บริการกดจองแล้วสามารถจองได้ หน้าเว็บเพจจะมีการแสดงรหัสผ่าน ดังรูปที่ 4.30




รูปที่ 4.30 เมื่อผู้ใช้บริการกดจองและสามารถจองได้ หน้าเว็บเพจจะแสดงรหัสผ่าน 2342

- หมายเลข 3.2 คือ การที่ผู้ใช้บริการกดจอง แต่ไม่สามารถจองได้ จะแสดงข้อความ “จองไม่ได้” เนื่องจากเวลาที่เลือกนั้นซ้ำกับคาบเวลาที่เครื่องนั้นถูกจองอยู่ ดังรูปที่ 4.31

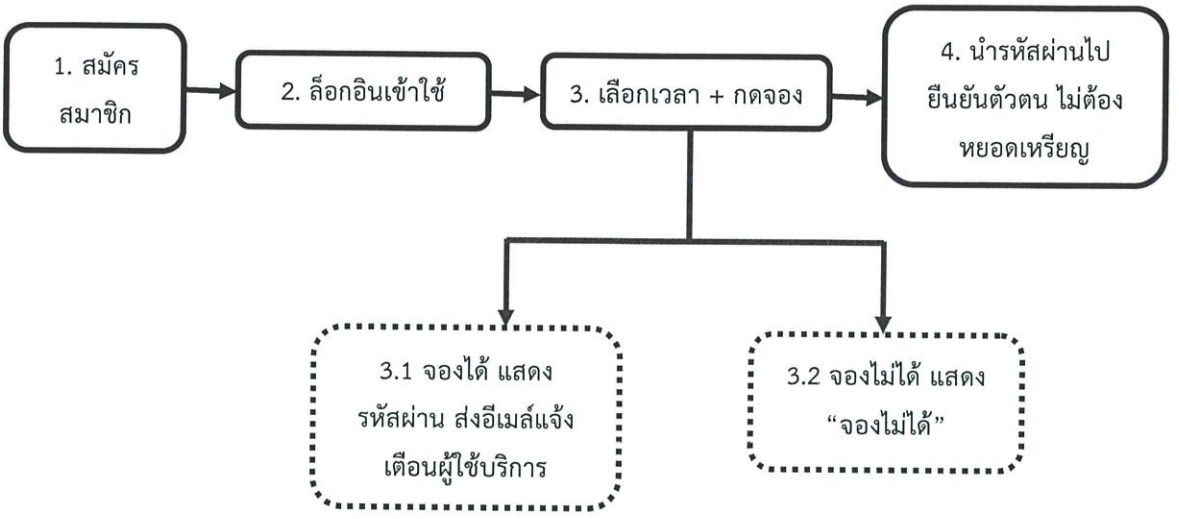



รูปที่ 4.31 การที่ผู้ใช้บริการกดจอง แต่ไม่สามารถจองได้ จะแสดงข้อความ “จองไม่ได้”

- หมายเลข 4 คือ การที่ผู้ใช้บริการนำรหัสผ่านไปยืนยันตัวตนที่หน้ากล่องควบคุม โดยนำรหัสผ่านที่ได้รับไปยืนยัน หลังจากยืนยันแล้ว ต้องทำการหยอดเหรียญ 10 บาท จำนวน 3 เหรียญ ถึงจะทำให้เครื่องซักผ้าทำงานได้ ดังรูปที่ 4.5 และรูปที่ 4.7

### 4.12.3 สำหรับผู้ใช้บริการที่ต้องการทำการจอง กรณีผู้ที่เป็นสมาชิก

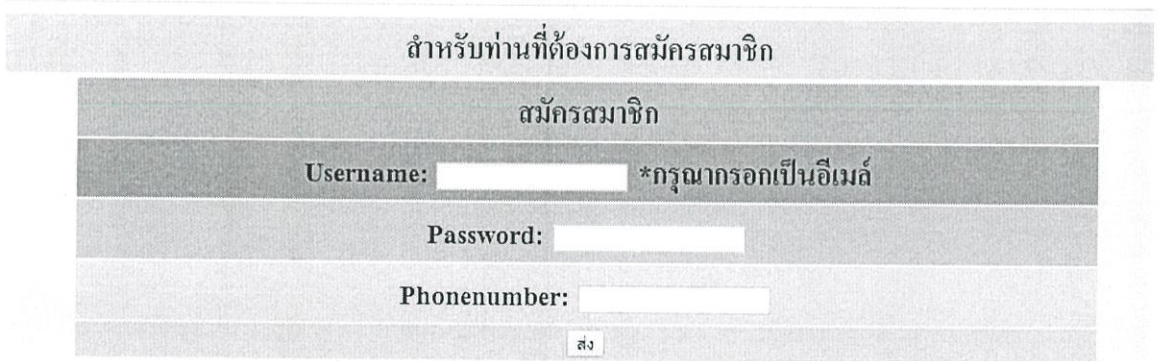
กรณีนี้ผู้ใช้บริการต้องทำการสมัครสมาชิกและเติมเงินก่อน ถึงจะจองได้ กรณีนี้จะมีการส่งอีเมล์แจ้งเตือนการใช้งานเครื่องซักผ้า และไม่ต้องไปหยอดเหรียญ แต่ใช่วิธีการหักเงินเอาจากเงินที่ผู้ใช้บริการเติมเข้าบัญชีตนเอง มีบล็อกไดอะแกรม ดังรูปที่ 4.32




รูปที่ 4.32 บล็อกไดอะแกรมแสดงการทำงานของระบบสำหรับผู้ใช้บริการที่ต้องการทำการจอง กรณีผู้ที่เป็นสมาชิก

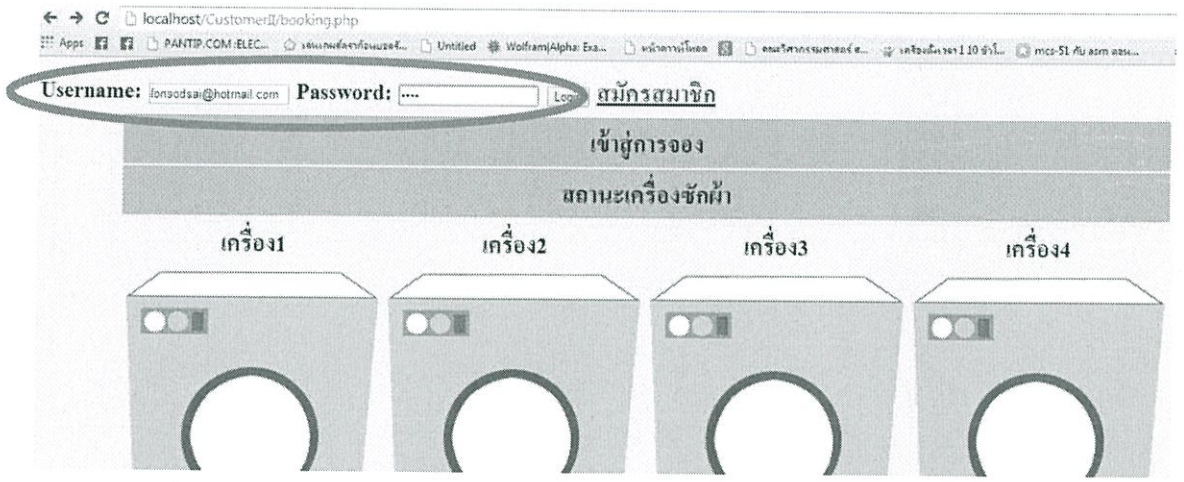
จากรูปที่ 4.32 อธิบายได้ว่า

- หมายเลข 1 คือการสมัครสมาชิก สำหรับผู้ที่ต้องการเป็นสมาชิก ต้องทำการสมัครสมาชิกก่อน ดังรูปที่ 4.33

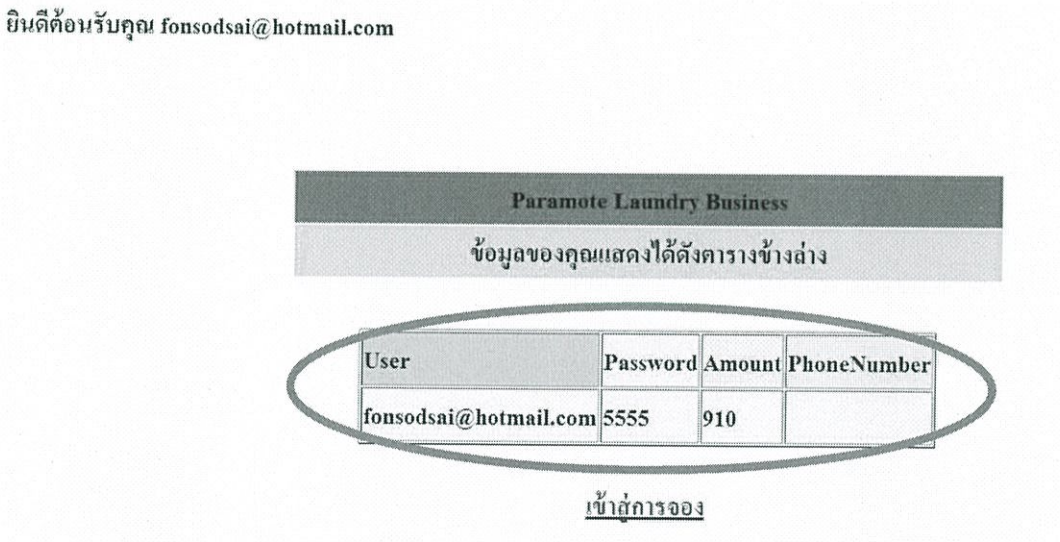



รูปที่ 4.33 หน้าเว็บสำหรับสมาชิก

- หมายเลข 2 คือ การล็อกอินเข้าใช้งาน ดังรูปที่ 4.34 เมื่อล็อกอินเข้าใช้งานสำเร็จจะแสดงข้อมูลของผู้ใช้บริการนั้นๆ ดังรูปที่ 4.35

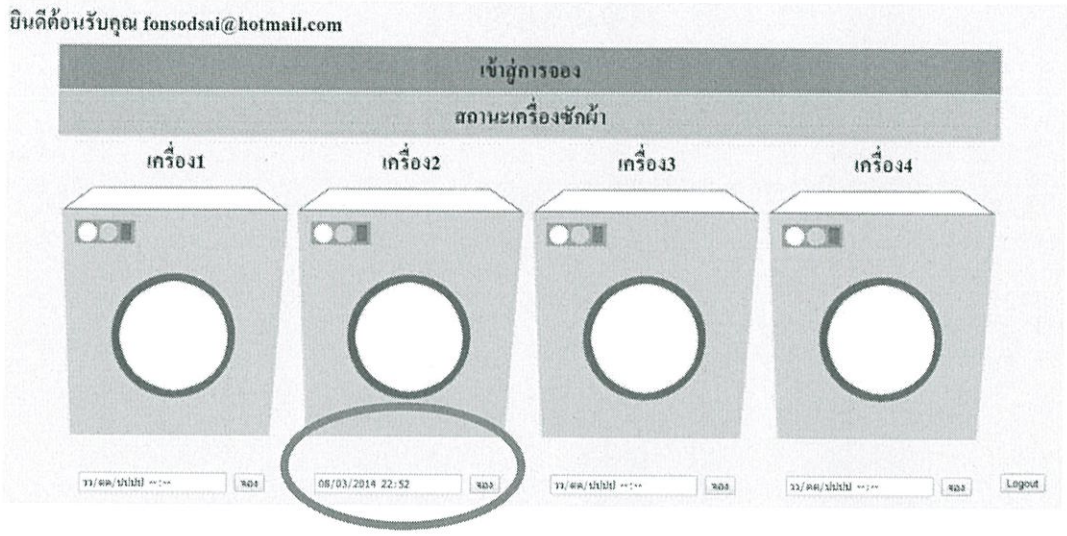



รูปที่ 4.34 การล็อกอินเข้าใช้งาน



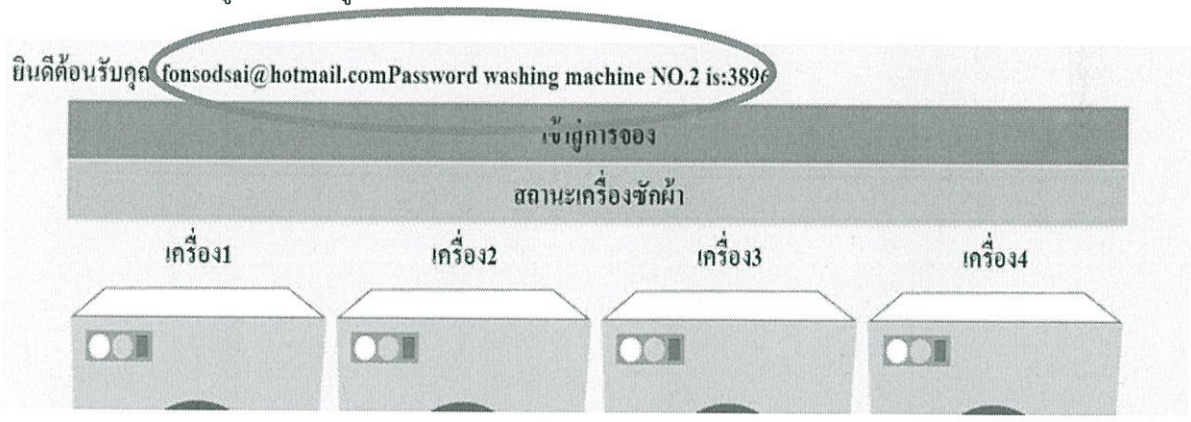

รูปที่ 4.35 เมื่อล็อกอินสำเร็จ หน้าเว็บเพจจะแสดงข้อมูลของผู้ใช้บริการ

- หมายเลข 3 คือการเลือกเวลาและการจอง เลือกเวลาที่ต้องการและกดจอง ดังรูปที่ 4.36




รูปที่ 4.36 หน้าเว็บเข้าสู่การจอง และการเลือกเวลาจอง

- หมายเลข 3.1 คือ กดจอง แล้วจองสำเร็จ จะแสดงรหัสผ่าน ดังรูปที่ 4.37 และมีการส่งอีเมล์แจ้งเตือนผู้ใช้งาน ดังรูปที่ 4.19

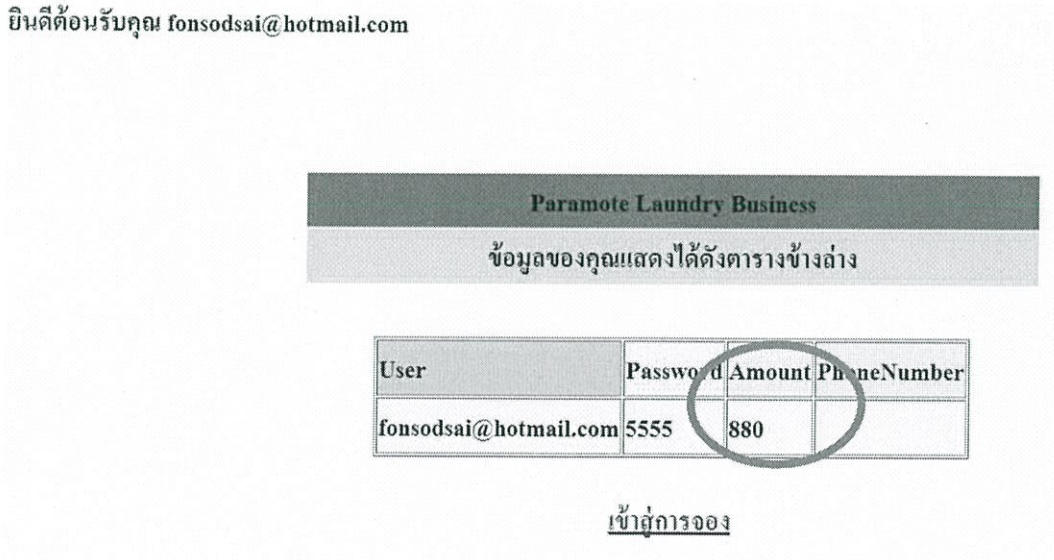



รูปที่ 4.37 ผู้ใช้บริการสามารถจองได้ หน้าเว็บเพจจะแสดงรหัสผ่าน 3896

- หมายเลข 3.2 คือ กดจอง แต่ไม่สามารถจองได้ เนื่องจากเวลาซ้ำกัน แสดงข้อความ “จองไม่ได้” ดังรูป 4.38

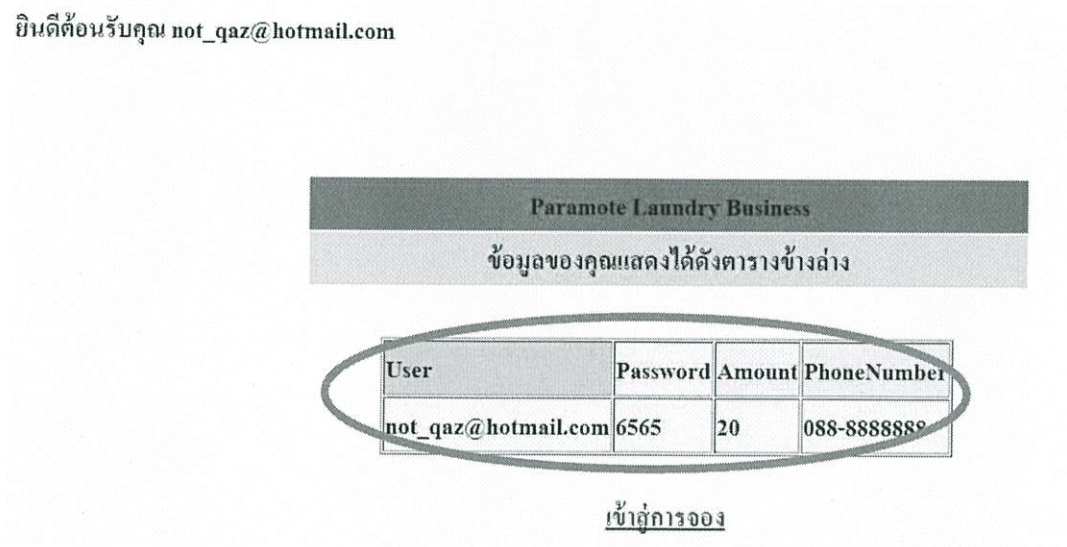
รูปที่ 4.38 ผู้ใช้บริการไม่สามารถจองได้ หน้าเว็บเพจจะแสดงข้อความ “จองไม่ได้”

- หมายเลข 4 คือการยืนยันตัวตน นำรหัสผ่านที่ได้ไปกดยืนยันที่กล่องควบคุมภายในเวลาที่ตนเองจองไว้ เครื่องซักผ้าจะสามารถทำงานได้เลย ไม่ต้องหยอดเหรียญ แต่ใช้วิธีการหักเงิน 30 บาท จากเงินที่ผู้ใช้บริการเติมเข้าบัญชีตนเอง ดังรูปที่ 4.39




รูปที่ 4.39 จำนวนเงินคงเหลือหลังจากการหักเงินเมื่อเข้าสู่การจอง

สำหรับสมาชิกที่มีเงินในระบบไม่ถึง 30 บาท ก็ไม่สามารถจองได้เช่นกัน ดังรูปที่ 4.40 และรูปที่ 4.41




รูปที่ 4.40 ข้อมูลการใช้งานของผู้ใช้บริการ

รูปที่ 4.41 ผู้ใช้บริการไม่สามารถได้ เนื่องจากเงินในระบบไม่พอ

# บทที่ 5

# สรุปผลและข้อเสนอแนะ

## 5.1 สรุปผลการทดลอง

1. ระบบควบคุมสามารถควบคุมการเปิดปิดเครื่องซักผ้า และมีการสั่งให้เครื่องซักผ้าสามารถทำงานได้

2. ระบบควบคุมสามารถแสดงจำนวนเหรียญที่หยอดและแสดงการกรอกรหัสผ่านผ่านจอแอลซีดี

3. ระบบควบคุมมีการกดรหัสผ่าน ผ่านคีย์แพดเพื่อยืนยันการใช้งานเครื่องซักผ้าที่ได้ทำการจองไว้

4. ระบบสามารถสั่งวงจรบันทึกเสียงและเล่นเสียงให้เล่นเสียงเมื่อมีบุคคลเข้ามาใช้บริการเครื่องซักผ้าที่ตนเองจองหรือเครื่องซักผ้าที่ตนเองต้องการใช้ได้

5. ระบบมีฐานข้อมูลเพื่อเก็บข้อมูลรายได้จากการให้บริการและมีระบบเว็บเซิร์ฟเวอร์ รวมทั้งมีหน้าเว็บเพจเพิ่มความสะดวกสบายแก่ผู้ใช้

## 5.2 ข้อเสนอแนะ

1. สามารถพัฒนาให้มีระบบการแจ้งเตือนในกรณีที่เครื่องซักผ้าที่ต้องการใช้ว่างหรือสิ้นสุดกระบวนการซักเพื่อให้ผู้ใช้บริการทราบผ่าน SMS ฯลฯ

2. สามารถพัฒนาระบบให้สามารถใช้บริการภายในอพาร์ตเมนท์ ห้องพัก หรือโรงแรม โดยผู้ใช้บริการสามารถใช้คีย์การ์ด (Keycard) ในการใช้บริการระบบได้

# บรรณานุกรม


[1] โชติกา พัฒนธีรเดช, ไชยยศ โกศัลลกูฏ และพฤกษสิทธิ์ อ่อนจิตร, ระบบควบคุมเครื่องใช้ไฟฟ้าภายในบ้านผ่านอินเตอร์เน็ต, ปริญญานิพนธ์รหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต, สาขาวิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2552.

[2] ผศ.ดร.พูลศักดิ์ โกษียาภรณ์, ณัฐพล จะสูงเนิน, เรียนรู้โครคอนโทรเลอร์ ARM7 ด้วยภาษา C, กรุงเทพฯ : สมาร์ทเลิร์นนิ่ง, 2553.

[3] ผศ.ธีรวัฒน์ ประกอบผล, การโปรแกรมภาษาซีสำหรับงานวิทยาศาสตร์, พิมพ์ครั้งที่1, กรุงเทพฯ : Imprint, 2545.

[4] นิรุธ อำนวย, คู่มือเขียนโปรแกรมด้วยภาษาซีด้วยตนเอง, กรุงเทพฯ : โปรวิชัน, ปี 2546.

[5] "คู่มือการใช้การควบคุมเครื่องซักผ้า", ttp://www.multihitech.net/manual/pmt258w.pdf [สืบค้นวันที่ 10 สิงหาคม 2556]

# ภาคผนวก

Source Code

โค้ดโปรแกรม ไฟล์ easyweb.c

```
#define extern
#include <LPC23xx.h>
#include <string.h>
#include "easyweb.h"
#include "EMAC.h"
#include "tcpip.h"
void set(void)
{
        TCPLowLevelInit();
        lcd_init();
        init_serial2();
        true_pass = -1;
        A=0;
        j=0;
        Key1=0;
        Key2=0;
        Key3=0;
        Key4=0;
        pass_digit=0;
        FIO2DIR = LED1|LED2|LED3|LED4;
        FIO0DIR &= ~(1<<4);
        LED1_OFF();
        LED2_OFF();
        LED3_OFF();
        LED4_OFF();
        CoinCount = 0;
        *(unsigned char *)RemoteIP = 192;
        *((unsigned char *)RemoteIP + 1) = 168;
        *((unsigned char *)RemoteIP + 2) = 1;
        *((unsigned char *)RemoteIP + 3) = 134;
        TCPLocalPort = 2025;
        TCPRemotePort = 1234;
        t=RTC_CTIME0 & MASKSEC;
        CheckReciveFinished=0;
        RTC_CCR  = 0;
```

```
        RTC_CCR |= 0x00000010;
        RTC_CCR |= 0x00000002;
        RTC_CCR &= 0xFFFFFFFD;
        RTC_CCR |= 0x00000001;
        goto_cursor(0x00);
        lcd_print("Enter pass:  ");
        pass_buf[4] = '\0';
        pass[0][0] = '\0';
        pass[1][0] = '\0';
        pass[2][0] = '\0';
        pass[3][0] = '\0';
        for(j = 0;j <=3 ; j++)
        {
                pass[j][0] = 'X';
                pass[j][1] = 'X';
                pass[j][2] = 'X';
                pass[j][3] = 'X';
        }
}
        int getCoin(void)
{
        int j=0;
        if(button_down)
        {
                if((FIO4PIN & (1 << 28)))
                {
                        button_down = 0;
                }
        }
        else
        {
                if(!(FIO4PIN & (1 << 28)))
                {
                        for(j = 0;j <= 3;j++)
                        {
                                if((FIO2PIN & (1<<(1 + j * 2))) && (pass[j][0] == 'X')
                                && ((true_pass < 0 || true_pass > 3)))
```

```
                                {
                                        true_pass = j;
                                }
                        }
                        CoinCount++;
                        button_down = 1;
                }
        }
        if((true_pass) >=0 && (true_pass)<=3)
        {
                A = CoinCount * 10;
                goto_cursor(0x44);
                sprintf(str_temp1,"%d BAHT ",A);
                lcd_print(str_temp1);
        }
        if(CoinCount == 3)
        {
                goto_cursor(0x00);
                lcd_print("Enter pass:   ");
                goto_cursor(0x44);
                lcd_print("        ");
                return 1;
        }
        else
        {
                return 0;
        }
}
void Send(void)
{
        unsigned int ByteToSend=0;
        int i;
        TCPReleaseRxBuffer();
        for(i = 0;i <= 3;i++)
        {
                if(FIO2PIN & (1<<(1 + i * 2)))
                {
```

```
                memcpy(TCP_TX_BUF+ByteToSend, LEDstatus_off[i] , 10);
            }
            else
            {
                memcpy(TCP_TX_BUF+ByteToSend, LEDstatus_on[i] , 10);
            }
            ByteToSend+=10;
        }
        TCPTxDataCount = ByteToSend;
        TCPTransmitTxBuffer();
        CheckReciveFinished = ReciveNotFinished;
        }
void CheckEndReceive(void)
{
        unsigned char *Key;
        unsigned int i;
        Key = TCP_RX_BUF;
        for(i = 0;i < MAX_TCP_RX_DATA_SIZE;i++)
        {
            if(*Key == 'E' && *(Key+1) == 'N' && *(Key+2) == 'D')
            {
                TCPClose();
                ReadPass();
                CheckReciveFinished = ReciveFinished;
                return;
            }
        Key++;
        }
}
void ReadPass(void)
{
        unsigned char *Key;
        unsigned int i;
        Key = TCP_RX_BUF;
        for(i = 0;i < MAX_TCP_RX_DATA_SIZE;i++)
        {
            if(*Key=='P' && *(Key+1)=='A' && *(Key+2)=='S' && *(Key+3)=='S')
```

```
                {
                        pass[(*(Key+4)) - 49][0] = *(Key+5);
                        pass[(*(Key+4)) - 49][1] = *(Key+6);
                        pass[(*(Key+4)) - 49][2] = *(Key+7);
                        pass[(*(Key+4)) - 49][3] = *(Key+8);
                        //goto_cursor(0x00);
                        //lcd_print(pass[(*(Key+4)) - 49]);
                }
        }
}
int main(void)
{
        unsigned int z,loop;
        unsigned char Keypass[4];
        int uart_data;
        set();
        while(1)
        {
                if(U2LSR & 0x01)
                {
                        uart_data = getchar2();
                        if ((uart_data)>=48 && (uart_data)<=57)
                        {
                                pass_buf[pass_digit] = uart_data;
                                pass_digit++;
                                goto_cursor(11);
                                lcd_print(pass_buf);
                        }
                }
                if(pass_digit == 4)
                {
                        pass_digit = 0;
                        goto_cursor(11);
                        lcd_print("      ");
                        for(loop = 0;loop <= 3;loop++)
                        {
                        if((pass[loop][0] == pass_buf[0]) && (pass[loop][1] ==
```

```
pass_buf[1]) && (pass[loop][2] == pass_buf[2]) && (pass[loop][3] == pass_buf[3]))
                        {
                                true_pass = loop;
                        }
                }
                if(true_pass > 3 || true_pass < 0) //If pass wrong
                {
                        true_pass = -1;//Enter new pass
                        goto_cursor(0x00);
                        lcd_print("Wrong  password");
                        delay(10000000);
                        goto_cursor(0x00);
                        lcd_print("Enter  pass:   ");
                }
                else
                {
                        goto_cursor(0x00);
                        lcd_print("Password  accept");
                }
                for(z=0;z<=3;z++)               //clear pass buffer
                {
                        pass_buf[z]= ' ';
                }
        }
        getCoin();
        if(true_pass <= 3 && true_pass >= 0) //If pass ok
        {
                if(CoinCount == 3)
                {
                        pass[true_pass][0] = 'X';
                        pass[true_pass][1] = 'X';
                        pass[true_pass][2] = 'X';
                        pass[true_pass][3] = 'X';
                        switch(true_pass)
                        {
                                case 0:
                                FIO2SET = LED1;
```

```
                        delay(5000000);
                        FIO2CLR = LED1;
                        true_pass = -1;//Ready to enter next pass
                        CoinCount = 0;
                        break;
                        case 1:
                        FIO2SET = LED2;
                        delay(5000000);
                        FIO2CLR = LED2;
                        true_pass = -1;//Ready to enter next pass
                        CoinCount = 0;
                        break;
                        case 2:
                        FIO2SET = LED3;
                        delay(5000000);
                        FIO2CLR = LED3;
                        true_pass = -1;//Ready to enter next pass
                        CoinCount = 0;
                        break;
                        case 3:
                        FIO2SET = LED4;
                        delay(5000000);
                        FIO2CLR = LED4;
                        true_pass = -1;//Ready to enter next pass
                        CoinCount = 0;
                        break;
        }
        }
}
DoNetworkStuff();
if ((RTC_CTIME0 & MASKSEC)-t>=5)
{
        TCPClose();
        t=RTC_CTIME0 & MASKSEC;
        CheckReciveFinished = ReciveFinished;
        TCPActiveOpen();
}
```

```
                DoNetworkStuff();
                if ((SocketStatus & SOCK_CONNECTED))
                {
                        if(CheckReciveFinished == ReciveFinished)
                        {
                                Send();
                        }
                        CheckEndReceive();
                }
        }
}
void init_serial2 (void)
{
        PINSEL0 &= 0xFF0FFFFF;                  // Reset P0.10,P0.11 Pin Config
        PINSEL0 |= 0x00100000;                  // Select P0.10 = TxD(UART2)
        PINSEL0 |= 0x00400000;                  // Select P0.11 = RxD(UART2)
        PCONP   |= 0x01000000;                  // UART2 Power-ON
        U2LCR = 0x83;                           // 8 bits, no Parity, 1 Stop bit
        U2DLL = 3;                      // Baud 115200BPS for 12MHz PCLK Clock
        U2FDR = 0x67;                           // Fractional Divider
        U2LCR = 0x03;                           // DLAB = 0
}
int getchar2(void)
{
        while (!(U2LSR & 0x01));                // Wait RXD Receive Data Ready
        return (U2RBR);                         // Get Receice Data & Return
}
```

## โค้ดโปรแกรมส่วนของโปรแกรมQT Creator

**ส่วนของHeader**

```
#ifndef SERVER_H
#define SERVER_H
#include <QObject>
#include <QtNetwork>
#include <QtSql>
class server : public QTcpServer
{
    Q_OBJECT
public:
    explicit server(quint16 port, QObject *parent = 0);
protected:
    void incomingConnection(qintptr  socketDescriptor);
    QString info[4];
private slots:
    void readClient(void);
    void discardClient(void);
public slots:
};
#endif // SERVER_H
```

**ส่วนของ Main**

```
#include <QCoreApplication>
#include "server.h"
server *oserver;
int main(int argc, char *argv[])
{
    QCoreApplication  a(argc, argv);
    oserver= new server(1234,&a);
    return a.exec();
}
```

**ส่วนของServer**

```
#include "server.h"
#include <stdio.h>
server::server(quint16 port, QObject *parent) :
    QTcpServer(parent)
{
    listen(QHostAddress::Any, port);
}
void server::incomingConnection(qintptr socketDescriptor)
{
    QTcpSocket *Socket = new QTcpSocket(this);
    connect(Socket,SIGNAL(readyRead()),this,SLOT(readClient()));
    if(!Socket->setSocketDescriptor(socketDescriptor))
    {
        delete Socket;
        return;
    }
}
void server::readClient(void)
{
    QTcpSocket *Socket =(QTcpSocket*)sender();
    if(Socket->canReadLine())
    {
        QTextStream os(Socket);
        QString common;
        QString input[4];
        QFile file ("C:\\AppServ\\www\\CustomerII\\file.txt");
        file.open(QIODevice::WriteOnly);
        QTextStream in(&file);
        QString request = Socket->readAll();
        in<<request;
        qDebug() << "request   " << request;
        file.close ();
        int startposition=request.indexOf("O",1);
        input[0]= request.mid(startposition,3);
        startposition+=3 ;
```

```
        startposition= request.indexOf("O",startposition);
        input[1]= request.mid(startposition,3);
        startposition+=3 ;
        startposition= request.indexOf("O",startposition);
        input[2]= request.mid(startposition,3);
        startposition+=3 ;
        startposition= request.indexOf("O",startposition);
        input[3]= request.mid(startposition,3);
        {
        QSqlDatabase db =QSqlDatabase::addDatabase("QMYSQL","customer");
        db.setHostName("localhost");
        db.setDatabaseName("customer");
        db.setUserName("root");
        db.setPassword("548199");
        db.open();
        QString data= "INSERT INTO revenue (id,date,amount) VALUES (NULL,";
        double datetime=0;
        QDate now =QDate::currentDate();
        datetime=now.toJulianDay();
        qDebug() << "Julian date       " << now.toJulianDay();
        datetime+= (double)(QTime::currentTime().hour()-19)/24;
        datetime+= (double)(QTime::currentTime().minute())/(24*60);
        datetime+= (double)(QTime::currentTime().second())/(24*60*60);
        data.append(QString::number(datetime,'f',6));
        data.append(",");
        int money = 0;
        int i ;
        for(i=0;i<4;i++)
        {
            qDebug() << "Info =======" << i << info[i];
            qDebug() << "Input =======" << i << input[i];
            if (info[i].compare("OFF")==0&&input[i].compare("ON_")==0)
            {
                money+=30;
                QFile timefile
(tr("C:\\AppServ\\www\\CustomerII\\timewash").append(QString::number(i+1)).append(
".txt"));
```

```
            timefile.open(QIODevice::WriteOnly);
            QTextStream in(&timefile);
            QString  timewash =
QString::number(QDateTime::currentMSecsSinceEpoch()/1000/60+50);
            in<<timewash;
            qDebug() << "timewash   " << timewash;
            timefile.close ();
        }
        info[i]=input[i];
    }
    data.append(QString::number(money));
    data.append(");");
    qDebug() << "DataTime =======" << datetime;
    qDebug() << data;
    if(money != 0)
        db.exec(data);
    }
    QSqlDatabase::removeDatabase("customer");
    qDebug() << Socket->peerAddress().toString();
    QFile file2 ("C:\\AppServ\\www\\CustomerII\\file2.txt");
    if(QFile::exists("C:\\AppServ\\www\\CustomerII\\file2.txt"))
    {
        file2.open(QIODevice::ReadOnly);
        QTextStream out(&file2);
        out>>common;
        file2.close();
        qDebug() << "common    " <<common;
        QFile::remove("C:\\AppServ\\www\\CustomerII\\file2.txt");
    }
    QFile file3 ("C:\\AppServ\\www\\CustomerII\\led1.txt");
    QFile nfile3 ("C:\\AppServ\\www\\CustomerII\\Booking1.txt");
    if(QFile::exists("C:\\AppServ\\www\\CustomerII\\led1.txt"))
    {
        file3.open(QIODevice::ReadOnly);
        QTextStream out1(&file3);
        QString normal ;
        out1>>normal;
```

```
    file3.close();
    if(request.indexOf("D1_OFF"))
    {
        qDebug() << "led1    " << normal;
        os<<normal;
        QFile::remove("C:\\AppServ\\www\\CustomerII\\led1.txt");
    }
}
QFile file4 ("C:\\AppServ\\www\\CustomerII\\led2.txt");
QFile nfile4 ("C:\\AppServ\\www\\CustomerII\\Booking2.txt");
if(QFile::exists("C:\\AppServ\\www\\CustomerII\\led2.txt"))
{
    file4.open(QIODevice::ReadOnly);
    QTextStream out2(&file4);
    QString normal1 ;
    out2>>normal1;
    fie4.close();
    if(request.indexOf("LED2_OFF"))
    {
        qDebug() << "led2    " << normal1;
        os<<normal1;
        QFile::remove("C:\\AppServ\\www\\CustomerII\\led2.txt");

}
QFile file5 ("C:\\AppServ\\www\\CustomerII\\led3.txt");
QFile nfile5 ("C:\\AppServ\\www\\CustomerII\\Booking3.txt");
if(QFile::exists("C:\\AppServ\\www\\CustomerII\\led3.txt"))
{
    file5.open(QIODevice::ReadOnly);
    QTextStream out3(&file5);
    QString normal2 ;
    out3>>normal2;
    file5.close();
    if(request.indexOf("LED3_OFF"))
    {
        qDebug() << "led3    " << normal2;
        os<<normal2;
```

```
            QFile::remove("C:\\AppServ\\www\\CustomerII\\led3.txt");
        }
    }
    QFile file6 ("C:\\AppServ\\www\\CustomerII\\led4.txt");
    QFile nfile6 ("C:\\AppServ\\www\\CustomerII\\Booking4.txt");
    if(QFile::exists("C:\\AppServ\\www\\CustomerII\\led4.txt"))
    {
        file6.open(QIODevice::ReadOnly);
        QTextStream out4(&file6);
        QString normal3 ;
        out4>>normal3;
        file6.close();
        if(request.indexOf("LED4_OFF"))
        {
            qDebug() << "led4   " << normal3;
            os<<normal3;
            QFile::remove("C:\\AppServ\\www\\CustomerII\\led4.txt");
        }
     }
    os << common << "END";
  }
  Socket->close();
  if(Socket->state() == QTcpSocket::UnconnectedState)
  {
     delete Socket;
  }
}
void server::discardClient(void)
{
  QTcpSocket *Socket = (QTcpSocket*)sender();
  Socket->deleteLater();
  qDebug() << "Discard client" ;
}
```