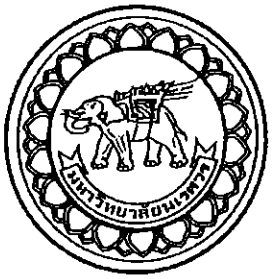


การเปิดและปิดปั๊มน้ำผ่านส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ต ร่วมกับระบบแลนไร้สาย

WATER PUMP ON-OFF CONTROL VIA WEB USER INTERFACE WITH WIRELESS FIDELITY

นายจิรายุส เชื้อเมืองพาน รหัส 54361008

นางสาวณัฐวดี สุริยาส่องไพร รหัส 54363743

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 20 ก.ค. 2558
เลขทะเบียน 10862822
เลขเรียกหนังสือ ปร
มหาวิทยาลัยนเรศวร ว 538 ส 2557

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2557

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน การเปิดและปิดปั๊มน้ำผ่านส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ตร่วมกับระบบแลนไร้สาย

ผู้ดำเนินโครงงาน นายจิรายุส เชื้อเมืองพาน รหัส 54361008

นางสาวณัฐวดี สุริยาส่องไพร รหัส 54363743

ที่ปรึกษาโครงงาน ผู้ช่วยศาสตราจารย์ ดร. นิพัทธ์ จันทรมินทร์

สาขาวิชา วิศวกรรมไฟฟ้า

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2557

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....ที่ปรึกษาโครงงาน

(ผู้ช่วยศาสตราจารย์ ดร. นิพัทธ์ จันทรมินทร์)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. ศุภวรรณ พลพิทักษ์ชัย)

.....กรรมการ

(ดร. สราวุฒิ วัฒนวงค์พิทักษ์)

ชื่อหัวข้อโครงงาน	การเปิดและปิดปั๊มน้ำผ่านส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ตร่วมกับระบบแลนไร้สาย
ผู้ดำเนินโครงงาน	นายจิรายุส เชื้อเมืองพาน รหัส 54361008
	นางสาวณัฐวดี สุริยาส่องไพร รหัส 54363743
ที่ปรึกษาโครงงาน	ผู้ช่วยศาสตราจารย์ ดร. นิพัทธ์ จันทรมินทร์
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2557

.....

บทคัดย่อ

ปริญญานิพนธ์นี้นำเสนอการพัฒนารูปแบบการควบคุมการเปิดและปิดปั๊มน้ำผ่านระบบแลนไร้สายโดยใช้ราสเบอร์รี่พายควบคุมการรับและส่งสัญญาณเพื่อควบคุมการเปิดและปิดปั๊มน้ำด้วยรีเลย์ซึ่งขึ้นอยู่กับระดับน้ำในถังเก็บน้ำที่ถูกวัดค่าด้วยตัวรับรู้ความดันส่วนต่าง ทั้งนี้ผู้ใช้สามารถกำหนดขีดจำกัดบนและขีดจำกัดล่างของระดับน้ำที่ต้องการในถังได้ รวมทั้งสามารถเลือกแบบวิธีการควบคุมซึ่งมีทั้งแบบวิธีการควบคุมด้วยมือและแบบวิธีการควบคุมอัตโนมัติ ในโครงงานนี้มีการเขียนชุดคำสั่งควบคุมการรับและส่งข้อมูลรวมถึงการสร้างส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ตด้วยโปรแกรมอะโดบี ดรีมวีฟเวอร์ และได้สร้างแบบจำลองเพื่อทดสอบการทำงานของระบบควบคุม

Project title	Water Pump On-Off Control via Web User Interface with Wireless Fidelity		
Name	Mr. Jirayus	Chuamuangphan	ID. 54361008
	Ms. Nattawadee	Suriyasongprai	ID. 54363743
Project advisor	Asst. Prof. Niphat Jantharamin, Ph.D.		
Major	Electrical Engineering		
Department	Electrical and Computer Engineering		
Academic year	2014		

Abstract

This thesis presents the development of an on-off control scheme for water pump through wireless local area network (LAN) by using a Raspberry Pi, which processes signals for the pump control via relay switching. The pump operating status depends upon a water level in a tank, which is measured by a differential pressure sensor. The user can determine not only water upper and lower limits but also an operating mode of the control scheme, either manual or automatic one, in real time. Instruction codes are programed to dictate reception and transmission of the control signals. In addition, a web user interface (WUI) is developed by using Adobe Dreamweaver and a model of water pump control is built to show feasibility of the developed control scheme.

กิตติกรรมประกาศ

กลุ่มผู้ดำเนินโครงงานขอขอบคุณ ผู้ช่วยศาสตราจารย์ ดร. นิพัทธ์ จันทรมินทร์ ที่ปรึกษาโครงงานที่ให้คำแนะนำในการพัฒนาชิ้นงาน การทดสอบและปรับปรุงชิ้นงาน ตลอดจนสอนหลักการเขียนปริญญานิพนธ์ด้วยความใส่ใจในรายละเอียด ทำให้การดำเนินโครงงานและการเขียนปริญญานิพนธ์สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณนายจรีเมศวร์ คำปันโย วิศวกรระดับ 5 การไฟฟ้าส่วนภูมิภาคเขต 2 ภาคเหนือ จังหวัดพิษณุโลก ที่ให้คำปรึกษาเกี่ยวกับการใช้งานราสเบอร์รี่พายเพื่อรับและข้อมูลแบบไร้สาย รวมถึงการเชื่อมต่อรีเลย์เพื่อเปิดและปิดมอเตอร์ปั๊มน้ำ

ขอขอบคุณคณาจารย์ทุกท่านที่ให้อบรมสั่งสอนตลอดการศึกษาเล่าเรียนในระดับปริญญาตรี ทำให้สามารถนำความรู้และทักษะในหลายๆด้านมาประยุกต์ใช้กับการดำเนินโครงงานนี้ รวมทั้งขอขอบคุณภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ที่ให้ยืมเครื่องมือวัดจนกระทั่งดำเนินโครงงานสำเร็จ

ขอขอบคุณการไฟฟ้าส่วนภูมิภาคเขต 2 ภาคเหนือ จังหวัดพิษณุโลก ที่ให้โอกาสได้ไปเรียนรู้และพัฒนาทักษะหลายด้านในช่วงฝึกงานจนได้แนวคิดในการพัฒนาชื่อหัวข้อโครงงาน

เหนือสิ่งอื่นใด ผู้ดำเนินโครงงานขอกราบขอบพระคุณบิดามารดาที่มอบความรักความเข้าใจ และคอยเป็นกำลังใจให้อยู่เสมอจนทำให้ประสบกับความสำเร็จอย่างทุกวันนี้

นายจิรายุส เชื้อเมืองพาน
นางสาวณัฐวดี สุริยาส่องไพร

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์ ..... ก
บทคัดย่อภาษาไทย ..... ข
บทคัดย่อภาษาอังกฤษ ..... ค
กิตติกรรมประกาศ ..... ง
สารบัญ ..... จ
สารบัญตาราง ..... ซ
สารบัญรูป ..... ฌ

บทที่ 1 บทนำ 1
1.1 ที่มาและความสำคัญของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบเขตของโครงงาน 2
1.4 ขั้นตอนและแผนการดำเนินงาน 2
1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน 3
1.6 งบประมาณ 3

บทที่ 2 หลักการทำงานของราสเบอร์รี่พายร่วมกับระบบเครือข่ายไร้สาย 4
2.1 ราสเบอร์รี่พาย 4
2.1.1 คุณลักษณะของราสเบอร์รี่พาย 4
2.1.2 คุณสมบัติทางเทคนิคของราสเบอร์รี่พาย 5
2.1.3 ส่วนประกอบของราสเบอร์รี่พาย 6
2.2 ตัวรับส่งแบบยูเอสบีของระบบแลนไร้สาย 9
2.3 ระบบเครือข่ายไร้สาย 10
2.4 เทคโนโลยีระบบแลนไร้สาย 13
2.5 การเขียนโปรแกรมเพื่อใช้งานราสเบอร์รี่พาย 14
2.5.1 ภาษาพีเอชพี 14
2.5.2 ภาษาไพธอน 16
2.5.3 ภาษาจาวาสคริปต์ 20

สารบัญ (ต่อ)

หน้า

2.6 ตัวรับรู้ความดันส่วนต่าง 21
2.7 การสร้างส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ต 23

บทที่ 3 การออกแบบและสร้างระบบควบคุมปั๊มน้ำร่วมกับระบบแลนไร้สาย 25
3.1 การออกแบบขั้นตอนการทำงานของระบบ 25
3.2 การออกแบบระบบควบคุมการทำงาน 31
3.2.1 การติดตั้งระบบปฏิบัติของราสเบอร์รี่พายลงในเอสดีการ์ด 31
3.2.2 การตั้งค่าราสเบอร์รี่พายให้ทำงานร่วมกับระบบแลนไร้สาย 32
3.2.3 การเขียนโปรแกรมติดต่ออินพุตหรือเอาท์พุตโดยใช้ไลบรารีไวริ่งพาย 34
3.2.4 การจัดการไฟล์บนราสเบอร์รี่พาย 35
3.2.5 การตั้งค่าราสเบอร์รี่พายร่วมกับส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ต 36
3.2.6 การประมวลผลและควบคุมรีเลย์ด้วยราสเบอร์รี่พาย 40
3.3 แบบจำลองของระบบมอเตอร์ปั๊มน้ำ 42

บทที่ 4 ผลการทดสอบและการวิเคราะห์ผล 45
4.1 การทดสอบวัดแรงดันไฟฟ้าจากตัวรับรู้ความดันส่วนต่าง 45
4.2 การทดสอบหาค่าความคลาดเคลื่อนของค่าระดับน้ำที่แสดงผล 46
4.3 การทดสอบการทำงานในแบบวิธีการควบคุมอัตโนมัติ 48
4.4 การทดสอบการทำงานในแบบวิธีการควบคุมด้วยมือ 48
4.5 การทดสอบการทำงานที่มีการตั้งเวลา 49

บทที่ 5 สรุปผลและข้อเสนอแนะ 50
5.1 สรุปผลการดำเนินงาน 50
5.2 ปัญหาและแนวทางการแก้ไข 50
5.3 แนวทางในการพัฒนาต่อไป 51

เอกสารอ้างอิง 52

สารบัญ (ต่อ)

หน้า

ภาคผนวก ก รหัสต้นฉบับเพื่อสร้างชุดคำสั่งสำหรับควบคุมการทำงานของระบบ 53
ภาคผนวก ข รายละเอียดข้อมูลของโมดูลตัวแปลงแอนะล็อกเป็นดิจิตอล รุ่น PCF8591 73
ภาคผนวก ค รายละเอียดข้อมูลของตัวรับรู้ความดันส่วนต่าง รุ่น MPX5010DP..... 87

ประวัติผู้ดำเนินโครงงาน 94

สารบัญตาราง

ตารางที่ หน้า

2.1 การเปรียบเทียบคุณสมบัติของเทคโนโลยีการเข้าถึงแบบไร้สาย 12
4.1 ค่าแรงดันไฟฟ้าจากตัวรับรู้ความดันส่วนต่าง 45
4.2 ค่าระดับน้ำที่แสดงบนส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ต 47

มหาวิทยาลัยนเรศวร

สารบัญรูป

รูปที่ หน้า

2.1 ส่วนประกอบของราสเบอร์รี่พาย 6
2.2 ตำแหน่งขาช่องสำหรับเชื่อมต่ออินพุตหรือเอาท์พุต 7
2.3 ตัวรับส่งแบบยูเอสบีของระบบแลนไร้สาย 9
2.4 การทำงานของคอมไพเลอร์ 17
2.5 การทำงานของตัวแปลคำสั่ง 18
2.6 กราฟคุณลักษณะของตัวรับรู้ความดันส่วนต่าง รุ่น MPX5010DP 22
3.1 ขั้นตอนการทำงานระบบเปิดและปิดปั๊มน้ำ โดยการควบคุมผ่านระบบแลนไร้สาย 25
3.2 ลำดับการทำงานของส่วนควบคุมที่กำหนดด้วยส่วนต่อประสานกับผู้ใช้ 27
3.3 ลำดับการทำงานของส่วนควบคุมแบบวิธีการควบคุมอัตโนมัติ 28
3.4 ลำดับการทำงานของส่วนควบคุมแบบวิธีการควบคุมด้วยมือ 30
3.5 ไฟล์อิมเมจของระบบปฏิบัติการ 31
3.6 โปรแกรมวิน 32 ดิสก์อิมเมจเจอร์ 32
3.7 โปรแกรมแอดวานซ์ไอพีสแกนเนอร์ 33
3.8 โปรแกรมพุตตี้ 33
3.9 ตารางสำหรับเลือกใช้งานอินพุตหรือเอาท์พุตของราสเบอร์รี่พาย 34
3.10 โปรแกรมวินเอสซีพี 35
3.11 หน้าต่างสำหรับการจัดการไฟล์ 36
3.12 หน้าต่างส่วนต่อประสานกับผู้ใช้สำหรับเข้าสู่ระบบ 37
3.13 หน้าต่างส่วนต่อประสานกับผู้ใช้สำหรับการควบคุม 38
3.14 การเชื่อมต่ออุปกรณ์สำหรับประมวลผลและควบคุมด้วยราสเบอร์รี่พาย 40
3.15 แผนภาพแสดงวงจรขับโลจิกโดยใช้ตัวเชื่อมต่อด้วยแสงบแรงดันไฟสูง 41
3.16 โมดูลรีเลย์แบบใช้ออปโต 41
3.17 กล่องอุปกรณ์สำหรับควบคุมและสั่งการปั๊มน้ำ 42
3.18 ตัวรับรู้ความดันส่วนต่าง รุ่น MPX5010DP 43
3.19 มอเตอร์ปั๊มน้ำ รุ่น AP2500 พิกัด 2,500 W 43
3.20 ส่วนประกอบและการเชื่อมต่ออุปกรณ์ในแบบจำลองของระบบควบคุมปั๊มน้ำ 44
4.1 ความสัมพันธ์ระหว่างแรงดันไฟฟ้าจากตัวรับรู้ความดันส่วนต่างกับระดับน้ำ 46
4.2 ข้อความแจ้งเตือนถึงสภาวะความผิดปกติที่เกิดขึ้นกับระบบ 49

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

การควบคุมอุปกรณ์ในระยะทางไกลจะมีสิ่งกีดขวางจากอุปกรณ์อื่นๆซึ่งเป็นอุปสรรค ทำให้การเชื่อมต่อสายควบคุมนั้นเป็นไปได้ยุ่งยาก และการใช้สายควบคุมที่ยาวจะมีค่าใช้จ่ายในการติดตั้งสูง จึงทำให้มีการนำเทคโนโลยีมาพัฒนาระบบควบคุมทางไกลในโรงงานอุตสาหกรรมที่ห้องควบคุมอยู่ไกลจากเครื่องจักรหรืออุปกรณ์ไฟฟ้าที่ต้องการควบคุมซึ่งมีการพัฒนาระบบควบคุมโดยผ่านทางเทคโนโลยีสื่อสารไร้สายแบบต่างๆ เช่น บลูทูธ (Bluetooth) ระบบแลนไร้สาย (Wireless fidelity, Wi-Fi) เป็นต้น เพื่อให้สามารถควบคุมได้จากระยะทางไกลแทนการใช้สายควบคุม และการควบคุมผ่านทางเทคโนโลยีสื่อสารไร้สายยังมีความคล่องตัวสูง สั่งการสะดวกสบายโดยที่ไม่ต้องอยู่กับที่ รวมไปถึงระบบความปลอดภัยที่ให้เฉพาะผู้ได้รับอนุญาตเท่านั้นที่สามารถเข้าถึงข้อมูล ดังนั้นในการนำเทคโนโลยีด้านการสื่อสารไร้สายมาใช้จึงเป็นการช่วยลดความยุ่งยากในการติดตั้งระบบเครือข่าย ช่วยลดปัญหาในการวางสายระบบเครือข่าย และเกิดความเป็นระเบียบ เรียบร้อยให้แก่ระบบ

ในโครงงานนี้จึงนำการควบคุมทางไกลแบบไร้สายมาพัฒนาควบคุมระบบเปิดและปิดปั๊มน้ำผ่านระบบแลนไร้สาย โดยใช้ราสเบอร์รี่พาย (Raspberry Pi) เป็นอุปกรณ์ในการประมวลผลและควบคุมการทำงานของรีเลย์ซึ่งใช้เป็นอุปกรณ์สำหรับตัดและต่อวงจรไฟฟ้าให้กับปั๊มน้ำ โดยสามารถควบคุมการเปิดและปิดปั๊มน้ำและแสดงสถานะการทำงานของปั๊มน้ำที่ถูกเชื่อมต่อเข้ากับระบบการควบคุมทางไกลแบบไร้สาย อีกทั้งยังมีการออกแบบส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ต (Web user interface, WUI) เพื่อให้ความสะดวกต่อผู้ใช้งาน

1.2 วัตถุประสงค์ของโครงงาน

เพื่อพัฒนาระบบควบคุมและแสดงสถานะการทำงานของปั๊มน้ำ โดยสั่งการผ่านอุปกรณ์สื่อสารไร้สายที่สามารถเชื่อมต่อกับอินเทอร์เน็ตได้ โดยมีการเชื่อมต่อเข้ากับใช้ราสเบอร์รี่พายเพื่อใช้สำหรับประมวลผลและควบคุมการทำงาน

1.3 ขอบเขตของโครงงาน

1) เขียนชุดคำสั่งเพื่อกำหนดค่าเริ่มต้นการทำงานของราสเบอร์รี่พายให้มีการสื่อสารผ่านโปรแกรมเทอร์มินอลบนอินเทอร์เน็ต

2) เขียนชุดคำสั่งสำหรับใช้สั่งการอินพุตหรือเอาท์พุต (General purpose input/output, GPIO) ของราสเบอร์รี่พาย

3) สร้างส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ตเพื่อใช้ราสเบอร์รี่พายควบคุมและแสดงสถานะการทำงานของปั๊มน้ำ

1.4 ขั้นตอนและแผนการดำเนินงาน

รายละเอียด	พ.ศ. 2557					พ.ศ. 2558			
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
1) ศึกษาหลักการทำงานและการตั้งค่าของราสเบอร์รี่พาย									
2) ศึกษาวิธีการติดต่อสื่อสารราสเบอร์รี่พายผ่านโปรแกรมเทอร์มินอลบนอินเทอร์เน็ต									
3) เขียนชุดคำสั่งสำหรับใช้สั่งการอินพุตหรือเอาท์พุตของราสเบอร์รี่พาย									
4) สร้างส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ต									
5) สร้างแบบจำลองเพื่อทดสอบการทำงานของปั๊มน้ำ									
6) สรุปผลและจัดทำรูปเล่มปริญญานิพนธ์									

1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน

ระบบควบคุมการทำงานทางไกลแบบไร้สายที่พัฒนาขึ้นในโครงงานนี้ สามารถนำไปประยุกต์ใช้กับการควบคุมปั๊มน้ำที่อยู่ไกลจากห้องควบคุม โดยผ่านระบบแลนไร้สาย ซึ่งมีการติดตั้งใช้งานอย่างแพร่หลาย และยังเป็นระบบที่มีเสถียรภาพในการสื่อสารมากกว่าการสื่อสารไร้สายชนิดอื่น อีกทั้งยังสามารถสั่งการผ่านระบบบนอินเทอร์เน็ตได้โดยไม่มีข้อจำกัดในเรื่องของระยะทาง เพื่อเป็นการอำนวยความสะดวกแก่ผู้ที่นำไปใช้ในงาน และการควบคุมการทำงานทางไกลแบบไร้สายนี้ช่วยยืดอายุการใช้งานของอุปกรณ์ให้ยาวนาน และเป็นการลดต้นทุนจากระบบเดิมที่ใช้สายไฟในการเดินสายไปยังห้องควบคุม

1.6 งบประมาณ

1) ราสเบอร์รี่พาย รุ่น B แรม 512 MB	1,600 บาท
2) ตัวรับส่งแบบยูเอสบีของระบบแลนไร้สาย	300 บาท
3) โมดูลตัวแปลงแอนะล็อกเป็นดิจิตอล รุ่น PCF8591	150 บาท
4) โมดูลรีเลย์ 2 ช่องทาง	150 บาท
5) ตัวรับรู้ความดันส่วนต่าง รุ่น MPX5010DP	500 บาท
6) ค่าถ่ายเอกสารและเข้าเล่มปริญญานิพนธ์	1,000 บาท
รวมเป็นเงินทั้งสิ้น (สามพันเจ็ดร้อยบาทถ้วน)	3,700 บาท

หมายเหตุ: ถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการทำงานของราสเบอร์รี่พายร่วมกับระบบเครือข่ายไร้สาย

เนื่องจากในโครงงานนี้ได้มีการนำราสเบอร์รี่พาย รุ่น B แรม 512 MB มาใช้ควบคุมการเปิดและปิดปั๊มน้ำผ่านระบบแลนไร้สาย ในระยะทางไกล โดยการเขียนชุดคำสั่งให้ราสเบอร์รี่พายประมวลผลข้อมูลที่ได้รับ เพื่อควบคุมการเปิดและปิดปั๊มน้ำ จึงควรมีความเข้าใจคุณลักษณะและหลักการทำงานเบื้องต้นของราสเบอร์รี่พาย คุณสมบัติของตัวรับส่งแบบยูเอสบีของระบบแลนไร้สาย (USB Wi-Fi) รวมถึงการสร้างส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ตที่นำมาใช้ในโครงงานนี้เพื่อควบคุมการทำงานของปั๊มน้ำ รวมทั้งความรู้เบื้องต้นเกี่ยวกับระบบเครือข่ายไร้สาย (Wireless local area network, WLAN)

2.1 ราสเบอร์รี่พาย

ราสเบอร์รี่พาย เป็นแผงวงจรที่รองรับระบบปฏิบัติการลินุกซ์ (Linux) ที่พัฒนาโดยพื้นฐานของราสเบอร์รี่พายที่ใช้ไมโครคอนโทรลเลอร์ (Microcontroller) ตระกูลเออาร์เอ็ม (ARM) โดยคำว่าพาย (Pi) นั้นมาจากไพธอน (Python) ซึ่งเป็นภาษาหลักที่ใช้ในการพัฒนา โดยสามารถติดตั้งระบบปฏิบัติการลงไปได้ และระบบปฏิบัติการที่ถูกติดตั้งลงไปนั้นถูกออกแบบพัฒนาเพื่อราสเบอร์รี่พายโดยเฉพาะ ก็คือราสเบียน (Raspbian) ซึ่งระบบปฏิบัติการดังกล่าวเกิดจากการนำเดเบียน (Debian) ซึ่งเป็นระบบปฏิบัติการในตระกูลลินุกซ์มาตัดแต่งให้เข้ากับราสเบอร์รี่พาย จึงได้เรียกว่าราสเบียน

2.1.1 คุณลักษณะของราสเบอร์รี่พาย

ราสเบอร์รี่พายมีความเล็กกะทัดรัด ทำให้มีความสะดวกเวลาเคลื่อนย้าย ประหยัดพลังงานเพราะใช้แรงดันไฟไม่ถึง 5 V มีการใช้ซอฟต์แวร์แบบเปิดเผยรหัส (Open source) ทั้งหมด โดยสามารถทำตัวเองเป็นผู้ให้บริการ (Server) เพื่อรองรับการร้องขอ (Request) จากผู้รับบริการ (Client) ได้มีช่องสำหรับเชื่อมต่ออินพุตหรือเอาท์พุต สามารถเชื่อมต่อกับตัวรับรู้ (Sensor) แบบต่างๆหรือแผงวงจรควบคุมอื่นๆได้อย่างอิสระ อีกทั้งยังมีราคาประหยัดและสามารถหาอุปกรณ์เสริมอื่นๆ เพื่อใช้งานร่วมกับราสเบอร์รี่พายได้อย่างหลากหลาย แต่อย่างไรก็ตามราสเบอร์รี่พายยังมีข้อจำกัดในเรื่องของความเร็ว ที่ค่อนข้างต่ำ แต่เมื่อใช้งานในแบบบรรทัดคำสั่ง (Command line) จะทำให้มีการทำงานที่เร็วขึ้น ซึ่งการตั้งค่าบางอย่างบนราสเบอร์รี่พายจำเป็นต้อง

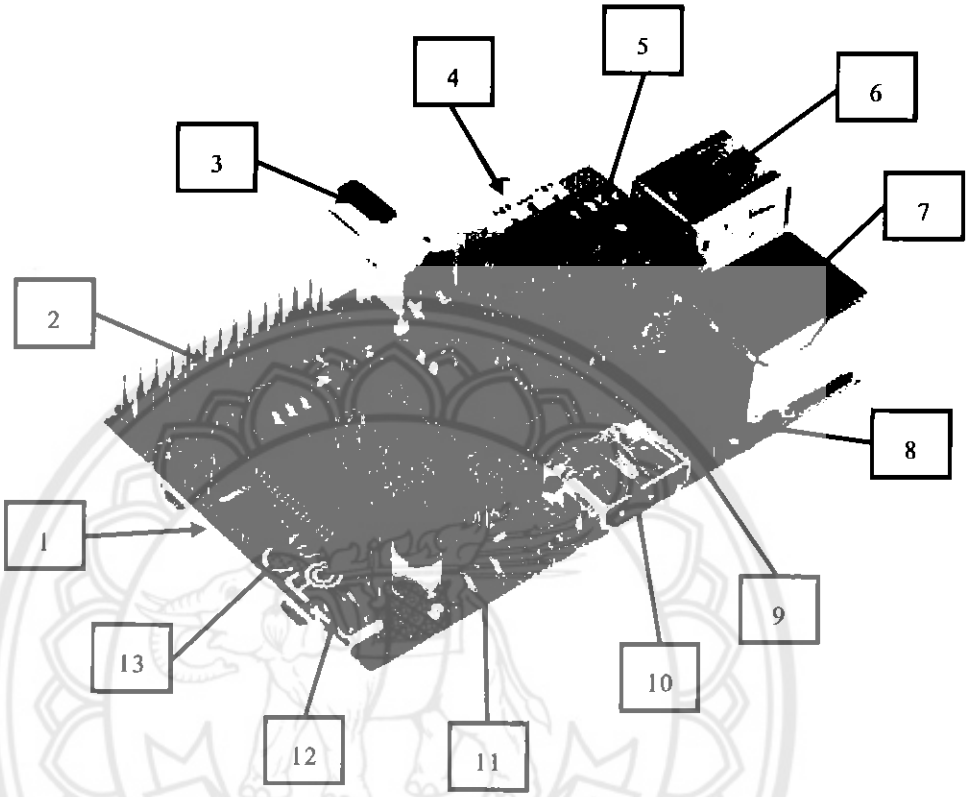
ใช้การสั่งการแบบบรรทัดคำสั่ง ซึ่งผู้ใช้งานอาจไม่ถนัดเมื่อมีการนำไปใช้งาน และด้วยรากฐานระบบปฏิบัติการที่เป็นลินุกซ์ ซึ่งอาจจะทำให้เกินความยุ่งยากแก่ผู้ที่ไม่คุ้นเคยใช้งานมาก่อน [1]

2.1.2 คุณสมบัติทางเทคนิคของราสเบอร์รี่พาย

1) หน่วยประมวลผลหลัก Broadcom BCM2835 ซึ่งรวมหน่วยประมวลผลกลาง (Central processing unit, CPU) หน่วยประมวลผลกราฟิก (Graphics processing unit, GPU) และหน่วยความจำเอสดี (SD RAM) ไว้ภายในตัวเดียวกัน
2) หน่วยประมวลผลกลาง ARM11 Core ARM1176JZF-S ความเร็ว 700 MHz
3) หน่วยประมวลผลกราฟิก Broadcom VideoCore IV, 1080p
4) หน่วยความจำเอสดีแรม 512 MB
5) ช่องต่อยูเอสบี (USB) 2.0 จำนวน 2 ช่อง
6) ช่องต่อสัญญาณภาพแบบอาร์ซีเอ (RCA) หรือแบบเอชดีเอ็มไอ (High definition multimedia interface, HDMI)
7) ช่องต่อสัญญาณเสียง โดยใช้ช่องต่อลำโพง ขนาด 3.5 mm.
8) ช่องสำหรับเชื่อมต่ออินพุตหรือเอาท์พุต
9) ระบบปฏิบัติการของแผงวงจรจะทำงานผ่านเอสดีการ์ด (SD card)
10) ช่องต่อแลน (LAN) 10/100 Mbps
11) ไฟเลี้ยงแผงวงจร 5 Vdc กระแสอย่างน้อย 700 mA
12) ขนาดของแผงวงจร 85 mm x 56 mm [2]

2.1.3 ส่วนประกอบของราสเบอร์รี่พาย

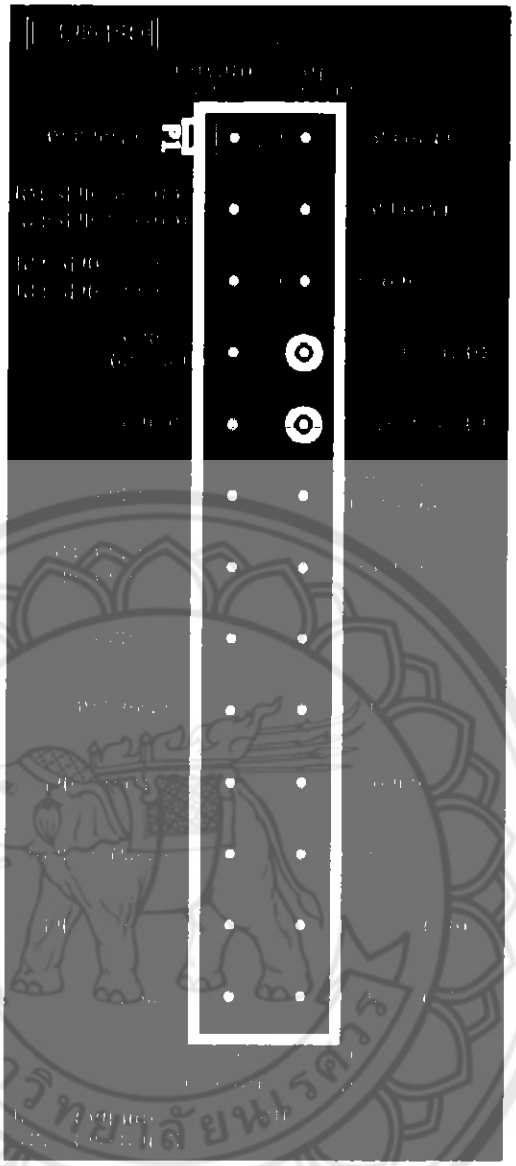
ในโครงงานนี้เลือกใช้ ราสเบอร์รี่พาย รุ่น B ดังรูปที่ 2.1



รูปที่ 2.1 ส่วนประกอบของราสเบอร์รี่พาย [1]

หมายเลข 1 ช่องต่อเอสดีการ์ด (ติดตั้งอยู่ใต้แผงวงจร) ซึ่งเป็นหน่วยความจำที่เก็บระบบปฏิบัติการและซอฟต์แวร์ทั้งหมดที่ถูกติดตั้งลงไปหรือเปรียบเทียบกับคอมพิวเตอร์ก็คือหน่วยความจำสำรองหรือฮาร์ดดิสก์

หมายเลข 2 ช่องสำหรับเชื่อมต่ออินพุตหรือเอาท์พุต เป็นช่องเชื่อมต่อที่สามารถกำหนดให้เป็นอินพุตหรือเอาท์พุตได้โดยการเขียนโปรแกรมขึ้นหรือใช้ซอฟต์แวร์ที่มีมาให้ดาวน์โหลดใช้ ทั้งนี้ยังสามารถทำงานได้หลายอย่าง เช่น สามารถที่จะเชื่อมต่อตัวรับรู้ต่างๆ โดยการป้อนค่าให้กับโปรแกรมตามที่ต้องการได้ เป็นต้น ขึ้นอยู่กับการเขียนโปรแกรม โดยตำแหน่งขาต่างๆ ดังรูปที่ 2.2



รูปที่ 2.2 ตำแหน่งขาช่องสำหรับเชื่อมต่ออินพุตหรือเอาท์พุต

ที่มา: http://www.elinux.org

หมายเลข 3 ช่องต่อสัญญาณภาพแบบอาร์ซีเอ เป็นช่องต่อสัญญาณภาพชนิดขั้วแบบอาร์ซีเอ โดยสามารถต่อเข้ากับโทรทัศน์ทั่วไปได้ มีประโยชน์ในการส่งสัญญาณภาพออกมาเพื่อตั้งค่าการเริ่มต้นใช้งานผ่านจอโทรทัศน์ได้

หมายเลข 4 ช่องต่อสัญญาณเสียง ขนาด 3.5 mm เป็นช่องต่อลำโพงสำหรับเปิดไฟล์สื่อที่มีเสียงได้ในซอฟแวร์ของราสเบอร์รี่พาย

หมายเลข 5 ไฟแอลอีดี (LED) เป็นสถานะไฟ เพื่อแสดงการทำงานของราสเบอร์รี่พาย

หมายเลข 6 ช่องต่อยูเอสบี 2.0 เป็นช่องสื่อสารไว้สำหรับเชื่อมต่ออุปกรณ์ต่างๆที่เป็นแบบยูเอสบีได้หลากหลายชนิดอุปกรณ์ เช่น กล้องเว็บแคม เมาส์ คีย์บอร์ด รวมถึงตัวรับส่งแบบยูเอสบีของระบบแลนไร้สาย เป็นต้น

หมายเลข 7 ช่องต่อแลน 10/100 Mbps เป็นช่องอีเทอร์เน็ต มีไว้สำหรับเชื่อมต่อราสเบอร์รี่พายเข้ากับระบบเครือข่ายคอมพิวเตอร์ โดยผ่านสายแลน (RJ 45) โดยการเชื่อมต่อนี้จะเป็นการนำราสเบอร์รี่พายเข้าสู่ระบบเครือข่ายท้องถิ่น และสามารถต่ออินเทอร์เน็ตได้ ทำให้สามารถอัพเดตซอฟต์แวร์ได้อิสระตามต้องการมากขึ้น ซึ่งเป็นปัจจัยหลักในการติดตั้งซอฟต์แวร์ต่างๆ

หมายเลข 8 ช่องต่อซีเอสไอ (CSI) เป็นช่องสื่อสารสำหรับเชื่อมต่อกับโมดูลกล้องที่ออกแบบมาเพื่อราสเบอร์รี่พาย โดยลักษณะการต่อจะเป็นสายแพ (Ribbon cable) ที่มากับตัวกล้องต่อเข้าไปกับช่องสื่อสารนี้และสามารถตั้งค่าการใช้งานกล้องได้อย่างสมบูรณ์แบบกว่าการนำกล้องแบบอื่นมาต่อเพราะกล้องที่นำมาต่อกับช่องนี้ถูกออกแบบมาเพื่อราสเบอร์รี่พายโดยเฉพาะ

หมายเลข 9 ตัวควบคุมแลนไร้สาย (LAN controller) เป็นหน่วยประมวลผลควบคุมการทำงานของช่องแลน

หมายเลข 10 ช่องต่อเอชดีเอ็มไอ คือช่องเสียบสายเอชดีเอ็มไอจากราสเบอร์รี่พาย เพื่อเข้าสู่จอแสดงผลที่รองรับการเชื่อมต่อภาพด้วยเอชดีเอ็มไอ เช่น จอคอมพิวเตอร์รุ่นใหม่ หรือสมาร์ตทีวี (Smart TV) ที่รองรับการเชื่อมต่อ

หมายเลข 11 หน่วยประมวลผล Broadcom BCM2835 เป็นหน่วยประมวลผลหลัก ซึ่งรวมหน่วยประมวลผลกลาง หน่วยประมวลผลกราฟิก และหน่วยความจำเอสดีไว้ภายในตัวเดียวกัน

หมายเลข 12 ช่องต่อไมโครยูเอสบี (Micro USB) เป็นช่องสำหรับต่อไฟเลี้ยงหรือช่องต่อตัวปรับต่อ (Adapter) เพื่อจ่ายไฟให้กับราสเบอร์รี่พาย โดยแรงดันไฟต้องไม่เกิน 5 V

หมายเลข 13 ช่องต่อดีเอสไอ (DSI) เป็นช่องสื่อสารที่มีไว้ต่อกับจอชนิดที่เป็นสายแพสำหรับเชื่อมต่อกับจอภาพ [1]

2.2 ตัวรับส่งแบบยูเอสบีของระบบแลนไร้สาย

ตัวรับส่งแบบยูเอสบีของระบบแลนไร้สาย รุ่น N 150 Mbps RALINK RA5370 ดังรูปที่ 2.3 มีการออกแบบระบบแลนไร้สายให้มีความเร็วสูงถึง 150 Mbps เหมาะสำหรับการใช้งานบนอินเทอร์เน็ต สอดคล้องกับมาตรฐาน IEEE 802.11

รูปที่ 2.3 ตัวรับส่งแบบยูเอสบีของระบบแลนไร้สาย [3]

การทำงานมี 2 รูปแบบคือ โครงสร้างพื้นฐาน (Infrastructure) และเอดีเอชโอซี (Ad-Hoc) โดยใช้เทคโนโลยีมัลติเพิลอินพุตมัลติเพิลเอาท์พุต (Multiple input multiple output, MIMO) ทำให้อัตราการส่งไร้สายได้ถึง 150 Mbps และเทคโนโลยีซีซีเอ (CCA) ช่วยเพิ่มเสถียรภาพของสัญญาณ

การตั้งค่าการรักษาความปลอดภัยสอดคล้องกับดับเบิลยูพีเอส (Wi-Fi protected setup, WPS) สามารถเข้ารหัสความปลอดภัยแบบไร้สายได้ง่ายโดยกดปุ่มคิวเอสเอส (QSS) มีการรองรับที่ 64/128 บิต ดับเบิลยูอีพี (Wired equivalent privacy, WEP) เช่นเดียวกับการเข้ารหัสลับข้อมูล WPA/WPA2 และ WPA-PSK/WPA2-PSK สามารถรองรับวินโดวส์ 2000 (Win2K) วินโดวส์เอกซ์พี (WinXP) วินโดวส์ 7 (Win7) แมค (MAC) ระบบปฏิบัติการลินุกซ์ และสนับสนุนเทคโนโลยีไร้สายสำหรับการเชื่อมต่อไร้สายที่มีประสิทธิภาพสูง [3]

2.3 ระบบเครือข่ายไร้สาย

ระบบเครือข่ายไร้สายเป็นระบบการสื่อสารข้อมูลที่มีความคล่องตัวอย่างมาก ซึ่งอาจจะนำมาใช้ทดแทนหรือเพิ่มเพื่อต่อกับระบบเครือข่ายแลนใช้สายแบบดั้งเดิมโดยใช้การส่งคลื่นความถี่ในย่านวิทยุ (Radio frequency, RF) และคลื่นอินฟราเรด (Infrared) ในการรับและส่งข้อมูลระหว่างคอมพิวเตอร์แต่ละเครื่อง ผ่านอากาศ ทะลุกำแพง เพดานหรือสิ่งก่อสร้างอื่นๆโดยปราศจากเดินสาย นอกจากนั้นระบบเครือข่ายไร้สายยังมีคุณสมบัติครอบคลุมทุกอย่างเหมือนกับระบบแบบใช้สาย แต่มีความแตกต่างที่สำคัญคือไม่ต้องใช้สายตัวนำ จึงทำให้การเคลื่อนย้ายในการใช้งานสามารถทำได้สะดวก ในขณะที่ระบบแบบใช้สายต้องใช้เวลาและการลงทุนในการปรับเปลี่ยนตำแหน่งการใช้งานเครื่องคอมพิวเตอร์

เทคโนโลยีการเข้าถึงข้อมูลเฉพาะในส่วนที่เป็นการเข้าถึงไร้สาย (Broadband wireless access, BWA) นั้นสามารถแบ่งออกตามลักษณะของการเข้าถึงได้ดังต่อไปนี้

1) ระบบการติดต่อไร้สายส่วนบุคคล (Personal area network, PAN) คือเทคโนโลยีการเข้าถึงไร้สายในพื้นที่เฉพาะส่วนบุคคลบริเวณไม่กว้างมากนัก ระยะทางไม่เกิน 10 m และมีอัตราการรับและส่งข้อมูลความเร็วสูงไม่เกิน 1 Mbps เทคโนโลยีเหล่านี้ติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์และอุปกรณ์ต่อพ่วง (Peripherals) ให้สามารถรับและส่งข้อมูลถึงกันได้
2) ระบบเครือข่ายท้องถิ่น (Local area network, LAN) คือเทคโนโลยีการเข้าถึงไร้สายในพื้นที่เฉพาะมีระยะทางที่ส่งไม่เกิน 100 m มีอัตราการรับและส่งข้อมูลความเร็วในช่วง 11 - 54 Mbps และติดตั้งสถานีเรียกว่าการต่อจุดทั้งหมดเพื่อทำหน้าที่เชื่อมต่อสัญญาณระหว่างอุปกรณ์ปลายทาง (Terminal equipment) ที่เป็นเซลล์ขนาดเล็กมาก (Pico cells) ข้อจำกัดการใช้งานเทคโนโลยีนี้คือจำนวนของผู้ใช้งานในขณะใดขณะหนึ่งพร้อมกัน ระยะห่างระหว่างจุดทั้งหมดกับอุปกรณ์ปลายทาง และความพอเพียงของคลื่นความถี่ เนื่องจากส่วนใหญ่เป็นการใช้งานในลักษณะที่ได้รับการยกเว้นใบอนุญาต (Unlicensed) จึงต้องใช้คลื่นความถี่ร่วมกันกับผู้ประกอบการ
3) ระบบเครือข่ายระดับเมือง (Metropolitan area network, MAN) คือเทคโนโลยีการเข้าถึงไร้สายในพื้นที่เขตเมืองหรือพื้นที่ขนาดใหญ่ ซึ่งมีระยะการส่งอยู่ในช่วงประมาณ 10 - 50 km ทั้งนี้ขึ้นอยู่กับคลื่นความถี่ที่ใช้งาน และมีอัตราการรับและส่งข้อมูลที่ความเร็วสูงในช่วง 11 - 100 Mbps ขึ้นไปโดยขึ้นอยู่กับการใช้งานว่าเป็นเอ็นแอลโอเอส (Non-line-of-sight, NLOS) หรือแอลโอเอส (Line-of-sight, LOS) แต่เดิมนั้นเทคโนโลยีการเข้าถึงดังกล่าวมุ่งเน้นที่การใช้งานแบบประจำที่ (Fixed) ซึ่งอุปกรณ์ของผู้ใช้บริการมักติดตั้งอยู่กับที่ภายนอกอาคาร (Outdoor) ก่อนมีการพัฒนาไปเป็น

การใช้งานภายในอาคาร (Indoor) แล้วจึงพัฒนาออกแบบให้สามารถใช้งานแบบเคลื่อนที่ (Mobile) ได้

4) ระบบเครือข่ายระหว่างประเทศ (Wide area network, WAN) คือเทคโนโลยีการเข้าถึงไร้สายบริเวณกว้างมากโดยครอบคลุมพื้นที่ในเขตภูมิภาคหรือทั่วประเทศมีอัตราการรับและส่งข้อมูลที่มีความเร็วไม่เกิน 1.5 Mbps เพราะเน้นการใช้งานในแบบเคลื่อนที่

ปัจจุบันระบบเครือข่ายแบบไร้สายถูกสร้างและพัฒนาเทคโนโลยีต่างๆขึ้นมาเพื่อให้เกิดความสะดวกสบายในการใช้งานหลายด้านมากขึ้น แต่ละเทคโนโลยีของระบบเครือข่ายแบบไร้สายนี้มีคุณสมบัติแตกต่างกันขึ้นอยู่กับการนำไปใช้งาน ดังนั้นก่อนเลือกใช้เทคโนโลยีต่างๆจึงจำเป็นต้องรู้คุณสมบัติของแต่ละเทคโนโลยีว่ามีข้อดีหรือข้อจำกัดเพื่อเลือกให้เหมาะสมกับงานที่นำไปใช้ เช่น เทคโนโลยีการเข้าถึงข้อมูลเฉพาะแบบไร้สาย ระยะทางการรับส่งสัญญาณ ความถี่ เป็นต้น ดังแสดงในตารางที่ 2.1 [4]

ซึ่งในโครงงานนี้ได้เลือกใช้เทคโนโลยีแบบระบบแลนไร้สาย เนื่องจากเป็นเทคโนโลยีที่มีการติดตั้ง ใช้งานอย่างแพร่หลาย และยังเป็นระบบที่มีความมั่นคงในการสื่อสารมากกว่าระบบสื่อสารไร้สายชนิดอื่น เช่น บลูทูธ ระบบสื่อสารไร้สาย (Zigbee) เป็นต้น อีกทั้งยังสามารถสั่งการผ่านระบบบนอินเทอร์เน็ตได้โดยไม่มีข้อจำกัดในเรื่องของระยะทางอีกด้วย

ตารางที่ 2.1 การเปรียบเทียบคุณสมบัติของเทคโนโลยีการเข้าถึงแบบไร้สาย [4]

เทคโนโลยี	เทคโนโลยีแบบไร้สาย	ระยะการทำงาน (m)	ความถี่ (GHz)
ยูดับเบิลยูบี (UWB)	ระบบการติดต่อไร้สายส่วนบุคคล	10	เปลี่ยนได้
บลูทูธ (Bluetooth)	ระบบการติดต่อไร้สายส่วนบุคคล	10	2.4
ระบบสื่อสารไร้สาย (Zigbee)	ระบบการติดต่อไร้สายส่วนบุคคล	10	868 MHz, 915 MHz, 2.4 GHz
ระบบแลนไร้สาย หรือ วายฟาย (Wi-Fi)	ระบบเครือข่ายท้องถิ่น	100	5
	ระบบเครือข่ายท้องถิ่น	100	2.4
	ระบบเครือข่ายท้องถิ่น	100	2.4
วายแม็กซ์ (WiMAX)	ระบบเครือข่ายระดับเมือง	6,400 - 9,600	11
	ระบบเครือข่ายระดับเมืองแบบเคลื่อนที่	1,600 - 4,800	2 - 6
วายด์แบนด์โคดดิวิชันมัลติเพิลแอ็คเซส	ระบบเครือข่ายระหว่างประเทศ	1,600 – 8,000	1.8, 1.9, 2.1
โคดดิวิชันมัลติเพิลแอ็คเซส (Cdma 2000)	ระบบเครือข่ายระหว่างประเทศ	1,600 – 8,000	0.4, 0.8, 0.9, 1.7, 1.8, 1.9, 2.1
เอ็มบีดับเบิลยูเอ (MBWA)	ระบบเครือข่ายระหว่างประเทศแบบเคลื่อนที่	4,000 - 12,000	3.5

จากตารางที่ 2.1 ยูดับเบิลยูบี เป็นเทคโนโลยีการสื่อสารไร้สายรูปแบบใหม่ใช้การส่งผ่านข้อมูลแบบพัลส์ (Pulse) สั้นๆผ่านคลื่นวิทยุความถี่กว้างต่างจากการส่งผ่านข้อมูลผ่านคลื่นความถี่วิทยุแบบระบบวิทยุแบบแถบความถี่แคบ และการส่งผ่านข้อมูลผ่านคลื่นความถี่วิทยุแบบกว้าง ในขณะที่วายแม็กซ์ คือเครือข่ายบริการอินเทอร์เน็ตไร้สายความเร็วสูง ที่มีพื้นที่ครอบคลุมบริเวณกว้างมาก ส่วนวายด์แบนด์ โคด ดิวิชัน มัลติเพิล แอ็คเซส (Wideband code division multiple access, WCDMA) คือระบบเครือข่ายมาตรฐานใหม่ที่พัฒนามาจากสามจี (3G) [4]

2.4 เทคโนโลยีระบบแลนไร้สาย

ในปัจจุบันอินเทอร์เน็ตมีการใช้งานกันอย่างมาก และนับวันอัตราการใช้อินเทอร์เน็ตก็ได้เพิ่มขึ้นเรื่อยๆอย่างต่อเนื่อง ส่งผลทำให้ระบบแลนไร้สายได้ถูกพัฒนาขึ้นมา ซึ่งเป็นเทคโนโลยีเครือข่ายไร้สายภายใต้เทคโนโลยีการสื่อสารมาตรฐาน IEEE 802.11 โดยเป็นมาตรฐานที่ถูกอนุมัติให้ใช้จาก IEEE (The Institute of Electrical and Electronics Engineers) เพื่อให้อุปกรณ์คอมพิวเตอร์สามารถสื่อสารกันได้บนมาตรฐานการทำงานแบบเดียวกัน สำหรับเทคโนโลยีเครือข่ายไร้สายนี้จะใช้คลื่นความถี่วิทยุและคลื่นความถี่อินฟราเรดในการรับและส่งข้อมูล คลื่นความถี่วิทยุของเครือข่ายไร้สายจึงสามารถทะลุทะลวงกำแพงหรือสิ่งกีดขวางได้ ทำให้การใช้งานบนเครือข่ายไร้สายมีความคล่องตัวและสะดวกสบายมากขึ้น โดยสามารถเชื่อมต่อเข้าสู่เครือข่ายได้ทุกที่ที่มีคลื่นสัญญาณ ข้อมูลจะถูกรับและส่งผ่านคลื่นวิทยุความถี่ 2.4 GHz ด้วยความเร็ว 11 Mbps ระยะห่างประมาณ 300 ft

การนำระบบแลนไร้สายมาใช้ประโยชน์ มีดังนี้

1) ช่วยลดค่าใช้จ่ายในการวางระบบเครือข่าย คือไม่ต้องมีการวางระบบเครือข่ายให้ยุ่งยาก ไม่ต้องมีการเดินสายแลน แค่มีอุปกรณ์จุดเข้าถึง (Access point) ที่กระจายสัญญาณได้ สามารถทำการขยายระบบได้ง่าย และทำให้ปรับองค์กรได้อย่างเหมาะสมตรงกับความต้องการ
2) ทำให้การสื่อสารง่ายขึ้น สะดวกต่อการเคลื่อนย้ายและเปลี่ยนแปลง ทำให้เกิดการทำงานที่สะดวกมากยิ่งขึ้น และในปัจจุบันอุปกรณ์คอมพิวเตอร์มีแนวโน้มที่จะพัฒนาให้มีขนาดเล็ก ประสิทธิภาพสูง สามารถพกติดตัวได้สะดวก พร้อมทั้งการทำงานที่เปลี่ยนแปลงไปตลอด จึงมีความสะดวกต่อการใช้งานโดยไม่ต้องอยู่กับที่
3) ช่วยอำนวยความสะดวกในการติดต่อทางธุรกิจ การซื้อขายบนระบบออนไลน์ และการทำธุรกรรมทางธนาคาร อีกทั้งยังเป็นการช่วยเสริมสร้างรายได้ เป็นแรงจูงใจดึงดูดลูกค้า และตอบสนองความต้องการของลูกค้า เช่น การให้บริการระบบแลนไร้สายตามร้านอาหาร ร้านกาแฟ โรงแรม ซึ่งทำให้ลูกค้าอยากเข้ามาใช้บริการมากขึ้น

ในอนาคตอุปกรณ์ต่างๆภายในบ้านเรือนจะมีระบบแลนไร้สายติดตั้งไว้เพื่อช่วยอำนวยความสะดวก จะทำให้สามารถควบคุมการทำงานผ่านคอมพิวเตอร์ ผ่านบริการระบบแลนไร้สายตามจุดต่างๆ และในอนาคตจะมีผู้ให้บริการโทรศัพท์ที่มีเทคโนโลยีการสื่อสารผ่านเครือข่ายอินเทอร์เน็ตความเร็วสูง (Broadband) โดยทำการรวมวอยซ์โอเวอร์ไอพี (Voice over IP, VoIP) และระบบแลนไร้สายไว้ในโทรศัพท์เคลื่อนที่ จุดเด่นของวอยซ์โอเวอร์ไอพีก็คือ เป็นการสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต จะไม่เสียเงินในการโทรโดยไม่มีการจำกัดเวลาในการใช้ติดต่อ เพราะว่าวอยซ์โอเวอร์ไอพีอาศัยการส่งข้อมูลผ่านทางอินเทอร์เน็ต และยังใช้โทรทางไกลไปยัง

ต่างประเทศได้อีกด้วย ซึ่งทำให้เกิดความสะดวกสบายยิ่งขึ้น และลดค่าใช้จ่ายลงได้ [5] ระบบแลนไร้สาย จึงเป็นระบบที่จะทำให้ทุกคนเข้าถึงเครือข่ายได้ในทุกพื้นที่และง่ายต่อการใช้งานและมีการนำระบบการสื่อสารไร้สายไปใช้งานมากยิ่งขึ้น โดยนำไปประยุกต์ประกอบกับเทคโนโลยีสื่อสารไร้สายอื่นๆ เพื่อช่วยในการอำนวยความสะดวกในการดำรงชีวิตของมนุษย์ในอนาคต

2.5 การเขียนโปรแกรมเพื่อใช้งานราสเบอร์รี่พาย

ในการใช้งานราสเบอร์รี่พาย จะมีการตั้งค่าและเขียนคำสั่งต่างๆเพื่อให้ราสเบอร์รี่พายสามารถทำงานได้ ดังนั้นจำเป็นต้องรู้จักกับภาษาที่ใช้ในการเขียนโปรแกรม มีดังต่อไปนี้

2.5.1 ภาษาพีเอชพี

พีเอชพี (PHP) เป็นภาษาคอมพิวเตอร์ในลักษณะสคริปต์ด้านเซิร์ฟเวอร์ (Server-side scripting) โดยลิขสิทธิ์อยู่ในลักษณะเปิดเผยรหัสข้อมูล ภาษาพีเอชพีใช้สำหรับจัดทำเว็บไซต์ และแสดงผลออกมาในรูปแบบเอชทีเอ็มแอล (HTML) โดยมีรากฐานโครงสร้างคำสั่งมาจากภาษาซี (C) ภาษาจาวา (Java) และภาษาเพิร์ล (Perl) ซึ่งภาษาพีเอชพีนั้นง่ายต่อการเรียนรู้ ซึ่งเป้าหมายหลักของภาษานี้ คือให้นักพัฒนาเว็บไซต์สามารถเขียนหน้าเว็บที่มีการตอบโต้ได้อย่างรวดเร็ว ซึ่งสรุปความสามารถของภาษาพีเอชพีได้ดังนี้

1) พีเอชพีเป็นภาษาที่มีลักษณะเป็นแบบเปิดเผยรหัส ผู้ใช้สามารถดาวน์โหลด และนำรหัสต้นฉบับ (Source code) ของพีเอชพีไปใช้ได้โดยไม่เสียค่าใช้จ่าย
2) พีเอชพีเป็นสคริปต์ด้านเซิร์ฟเวอร์ ดังนั้นจึงทำงานบนเว็บเซิร์ฟเวอร์ไม่ส่งผลกับการทำงานของเครื่องผู้รับบริการ โดยพีเอชพีจะอ่านรหัส และทำงานที่เซิร์ฟเวอร์ จากนั้นจึงส่งผลลัพธ์ที่ได้จากการประมวลผลมาที่เครื่องของผู้ใช้ในรูปแบบของเอชทีเอ็มแอล ซึ่งรหัสของพีเอชพีนี้ผู้ใช้จะไม่สามารถมองเห็นได้
3) พีเอชพีสามารถทำงานได้ในระบบปฏิบัติการที่ต่างชนิดกัน เช่น ยูนิกซ์ (Unix) วินโดวส์ (Windows) แมคโอเอส (Mac OS) หรือ ริสก์โอเอส (Risc OS) อย่างมีประสิทธิภาพ เนื่องจากพีเอชพีเป็นสคริปต์ที่ต้องทำงานบนเซิร์ฟเวอร์ ดังนั้นคอมพิวเตอร์สำหรับเรียกใช้คำสั่งพีเอชพี จึงจำเป็นต้องติดตั้งโปรแกรมเว็บเซิร์ฟเวอร์ไว้ด้วย เพื่อให้สามารถประมวลผลพีเอสพีได้
4) พีเอชพีสามารถทำงานได้ในเว็บเซิร์ฟเวอร์หลายชนิด เช่น พีดับเบิลยูเอส (Personal web server, PWS) อะแพชี (Apache) ออมนิเอชทีทีพีดี (OmniHttpd) และ ไอไอเอส (Internet information service, IIS) เป็นต้น

5) ภาษาพีเอชพีสนับสนุนการเขียนโปรแกรมเชิงวัตถุ (Object oriented programming)
6) พีเอชพีมีความสามารถในการทำงานร่วมกับระบบจัดการฐานข้อมูลที่หลากหลาย ซึ่งระบบจัดการฐานข้อมูลที่สนับสนุนการทำงานของพีเอชพี เช่น ออราเคิล (Oracle) มายเอสคิวแอล (MySQL) ไฟล์โปร (FilePro) ซอลลิด (Solid) ฟรอนท์เบส (FrontBase) เอ็มเอสคิวแอล (mSQL) และเอ็มเอสเอสคิวแอล (MS SQL) เป็นต้น
7) พีเอชพีอนุญาตให้ผู้ใช้สร้างเว็บไซต์ซึ่งทำงานผ่านโพรโทคอล (Protocol) ชนิดต่างๆ ได้ เช่น แอลดีเอพี (Lightweight directory access protocol, LDAP) ไอแมป (Internet message access protocol, IMAP) เอสเอ็นเอ็มพี (Simple network management protocol, SNMP) พ็อปทรี (Post office protocol 3, POP3) และเอชทีทีพี (Hypertext transfer protocol, HTTP) เป็นต้น
8) รหัสพีเอชพีสามารถเขียน และอ่านในรูปแบบของเอกซ์เอ็มแอล (Extensible markup language, XML) ได้

หลักการทำงานของภาษาพีเอชพี มีดังนี้

1) จากไคลเอนต์จะเรียกไฟล์พีเอชพีสคริปต์ (PHP script) ผ่านทางโปรแกรมค้นดูหรือบราวเซอร์ (Browser)
2) บราวเซอร์จะส่งคำร้องไปยังเว็บเซิร์ฟเวอร์ผ่านทางเครือข่ายอินเทอร์เน็ต
3) เมื่อเว็บเซิร์ฟเวอร์รับคำร้องขอจากบราวเซอร์แล้วจะนำสคริปต์พีเอชพีที่เก็บอยู่ในเซิร์ฟเวอร์มาประมวลผลด้วยโปรแกรมแปลภาษาพีเอชพีที่เป็นตัวแปลคำสั่ง (Interpreter)
4) กรณีที่พีเอชพีสคริปต์มีการเรียกใช้ข้อมูลจะติดต่อกับฐานข้อมูลต่างๆผ่านทางโอดีบีซี (Open database connectivity, ODBC) ถ้าเป็นฐานข้อมูลที่มีอยู่ในพีเอชพีไลบรารี (PHP library) ในการเชื่อมต่อฐานข้อมูลเพื่อดึงข้อมูลออกมาหลังจากแปลสคริปต์พีเอชพีเสร็จแล้วจะได้รับไฟล์เอชทีเอ็มแอลใหม่ที่มีแต่ข้อความเอชทีเอ็มแอลไปยังเว็บเซิร์ฟเวอร์
5) เว็บเซิร์ฟเวอร์ส่งไฟล์เอชทีเอ็มแอลที่ได้ผ่านการแปลแล้วกลับไปยังบราวเซอร์ที่ร้องขอผ่านทางเครือข่ายอินเทอร์เน็ต
6) บราวเซอร์รับไฟล์เอชทีเอ็มแอลที่เว็บเซิร์ฟเวอร์ส่งมาให้แปลเอชทีเอ็มแอลแสดงผลออกมาทางจอภาพเป็นหน้าเว็บโดยใช้ตัวแปลภาษาเอชทีเอ็มแอลที่อยู่ในบราวเซอร์ซึ่งเป็นตัวแปลคำสั่งเช่นเดียวกัน

พีเอชพีเป็นภาษาเขียนโปรแกรมที่นิยมใช้เพราะนอกจากเรื่องของซอร์ฟแวร์ที่เป็นเสรีแล้ว พีเอชพียังเป็นภาษาโปรแกรมที่สามารถเขียนงานที่มีความซับซ้อน ประโยชน์ของงานเขียนใน

พีเอชพี ก็คือการที่สามารถเข้าถึงได้อย่างรวดเร็วและหากเขียนอย่างถูกต้องจะมีความปลอดภัยด้วย นอกจากนี้ในการทำงานสามารถนำไปปรับแต่งให้เหมาะสมกับความชอบหรือการนำไปใช้ประโยชน์ตามความต้องการของผู้ใช้

การพัฒนาเว็บมีการเติบโตอย่างรวดเร็ว ตั้งแต่มาการใช้งานอินเทอร์เน็ตอย่างแพร่หลาย จนกลายเป็นอุตสาหกรรมที่มีการแข่งขันสูง เนื่องจ เกมีผู้สนใจในการเขียนโปรแกรมมากมาย และมีความเชี่ยวชาญ ดังนั้นการเรียนรู้พีเอชพี จึงเป็นการพัฒนาที่จะเป็นประโยชน์ต่อผู้สนใจในการเขียนโปรแกรมบนอินเทอร์เน็ต ในการสร้างเว็บไซต์การที่ต้องรู้ภาษาพีเอชพีเป็นหนึ่งช่องทางที่ดีและง่ายต่อการใช้โปรแกรมภาษา ที่สามารถเรียกใช้บนระบบปฏิบัติการใดๆสำหรับการจัดการการเชื่อมต่อฐานข้อมูลสามารถขยายได้อย่างง่าย เหมาะสำหรับการทำงานที่ต้องการเพิ่มหรือลดข้อมูลในเว็บไซต์ เพื่อทำให้ข้อมูลมีความทันสมัย

พีเอชพีจะขึ้นอยู่กับภาษาซีพลัสพลัส (C++) โปรแกรมและไวยากรณ์ที่ใช้ในพีเอชพีคล้ายภาษาซี ซีพลัสพลัส และพีเอชพีได้รับการสนับสนุนจากบริษัทหลากหลาย สามารถเป็นทางเลือกเพื่อเรียกใช้โปรแกรมประยุกต์บนแพลตฟอร์มด้านภาษาสคริปต์ เพื่อสร้างเว็บไซต์แบบไดนามิก และคำสั่งพีเอชพี สามารถช่วยให้ดึงข้อมูลจากเว็บไซต์อื่นๆมาใช้งานได้อย่างสะดวก [6]

2.5.2 ภาษาไพธอน

ไพธอน เป็นภาษาระดับสูงภาษาหนึ่ง ที่มีความสามารถสูงถูกสร้างขึ้นในปี ค.ศ. 1989 โดย Guido Van Rossum ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือสามารถรันภาษาไพธอนได้ทั้งบนระบบยูนุกซ์ ลินุกซ์ วินโดวส์ หรือแม้แต่ระบบฟรีบีเอสดี (FreeBSD) อีกอย่างหนึ่งภาษาตัวนี้เป็นภาษาลักษณะเปิดเผยรหัสเหมือนอย่างพีเอชพี

ในปัจจุบันภาษาที่ใช้ในการพัฒนาเว็บแอพพลิเคชั่น (Web application) มีมากมายหลายภาษา เช่น ภาษาเพิร์ล พีเอชพี จาวา เอเอสพี ทีซีแอล และไพธอน เป็นต้น สำหรับภาษาไพธอนนับว่ายังใหม่ในวงการพัฒนาโปรแกรมบนเว็บ แต่ด้วยข้อดีหลายประการของภาษาไพธอน ทำให้มีผู้นิยมใช้มากขึ้นเรื่อยๆ ซึ่งสามารถสรุปความสามารถของภาษาไพธอน ได้ดังนี้

1) ง่ายต่อการเรียนรู้ โดยภาษาไพธอนมีโครงสร้างของภาษาไม่ซับซ้อนเข้าใจง่าย ซึ่งโครงสร้างภาษาไพธอนจะคล้ายกับภาษาซีมาก เพราะภาษาไพธอนสร้างขึ้นมาโดยใช้ภาษาซี ทำให้ผู้ที่คุ้นเคยภาษาซีอยู่แล้วใช้งานภาษาไพธอนได้ไม่ยาก นอกจากนี้โดยตัวภาษาเองมีความยืดหยุ่นสูงทำให้การจัดการกับงานด้านข้อความ และแฟ้มข้อความ (Text file) ได้เป็นอย่างดี

2) ไม่ต้องเสียค่าใช้จ่ายใดๆทั้งสิ้น เพราะตัวแปรภาษาไพธอนอยู่ภายใต้ลิขสิทธิ์จีเอ็นยู (GNU)

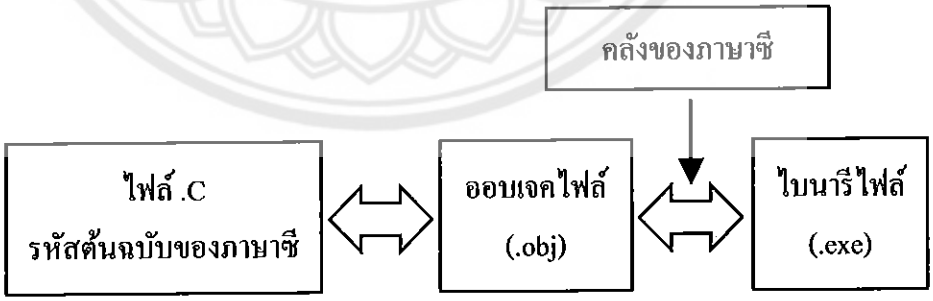
3) ใช้ได้หลายแพลตฟอร์ม ในช่วงแรกภาษาไพธอนถูกออกแบบใช้งานกับระบบยูนุกซ์ แต่ในปัจจุบันได้มีการพัฒนาตัวแปลภาษาไพธอนให้สามารถใช้กับระบบปฏิบัติการอื่นๆ เช่น ลินุกซ์ วินโดวส์ และโอเอส เป็นต้น

4) ภาษาไพธอนถูกสร้างขึ้นโดยได้มีการรวบรวมเอาส่วนดีของภาษาต่างๆเข้ามาไว้ด้วยกัน เช่น ภาษาซี ซีพลัสพลัส จาวา และเพิร์ล เป็นต้น

5) ภาษาไพธอนเป็นภาษาประเภทสคริปต์ด้านเซิร์ฟเวอร์ คือการทำงานของภาษาไพธอนจะทำงานด้านผู้ให้บริการแล้วส่งผลลัพธ์กลับมายังผู้รับบริการ ทำให้มีความปลอดภัยสูง

เมื่อมีเขียนรหัสขึ้นมาตามโครงสร้างของโปรแกรมภาษา เพื่อให้รหัสคำสั่งสามารถทำงานได้ต้องมีตัวแปรภาษามาจัดการแปลรหัสคำสั่ง โดยลักษณะของตัวแปลภาษานั้นแบ่งได้ 2 ประเภทใหญ่ๆ คือ

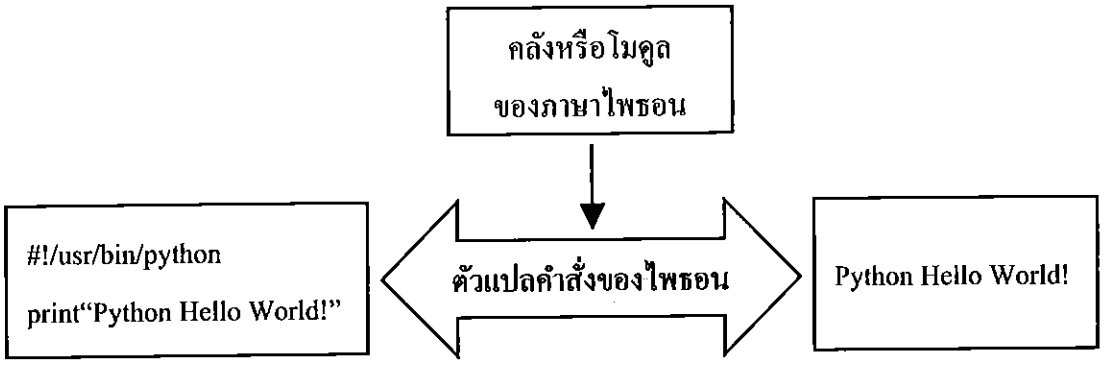
1) คอมไพเลอร์ (Compiler) เป็นตัวแปลภาษาสำหรับภาษาซี ซีพลัสพลัส และปาสกาล (Pascal) โดยมีหน้าที่ตรวจสอบความผิดพลาดของรหัสคำสั่งตั้งแต่ต้นจนจบก่อน หรือเรียกว่าการแปลให้เป็นภาษาคอมพิวเตอร์ ถ้าไม่มีข้อผิดพลาดจะทำการแปลรหัสคำสั่งให้เป็นไฟล์นามสกุลออบเจคไฟล์ (Object file) .obj จากนั้นก็ทำการแปลไฟล์ .obj ให้เป็นไบนารีไฟล์ (Binary file) .exe เพื่อทำงานต่อไป ดังตัวอย่างการทำงานของคอมไพเลอร์ภาษาซี ดังรูปที่ 2.4



รูปที่ 2.4 การทำงานของคอมไพเลอร์

ที่มา: http://python.cmsthailand.com/

2) ตัวแปลคำสั่ง (Interpreter) จะทำงานเป็นบรรทัดต่อบรรทัด คือ อ่านรหัสคำสั่งมาบรรทัดหนึ่งแล้วก็ทำงานให้ผลออกมา ดังรูปที่ 2.5



รูปที่ 2.5 การทำงานของตัวแปลคำสั่ง

ที่มา: http://python.cmsthailand.com/

จากรูปที่ 2.5 ในกรณีที่มีการเรียกใช้ฟังก์ชันจากคลัง (Library) หรือโมดูล (Module) ของภาษาไพธอน ตัวแปลคำสั่งของภาษาไพธอน จะไปทำการเรียกฟังก์ชันเหล่านั้นให้ทำงานแล้วจึงแสดงผลการทำงานออกมา

ในส่วนของประสิทธิภาพการทำงานนั้นตัวแปลภาษาแบบคอมไพเลอร์จะทำงานได้เร็วกว่าตัวแปลภาษาและตัวแปลคำสั่ง เพราะรหัสคำสั่งถูกแปลและลิเชื่อมโยงโดยตัวแปลภาษาแบบคอมไพเลอร์ผ่านแล้วได้เป็นไฟล์ .exe ออกมา จากนั้นก็เป็นขั้นตอนการทำงานอย่างเดียว

คุณลักษณะเด่นของภาษาไพธอน มีดังนี้

1) ภาษาไพธอนสนับสนุนแนวคิดแบบการเขียนโปรแกรมเชิงวัตถุ (Object oriented programming, OOP)
2) ไพธอนเป็นภาษาคอมพิวเตอร์ที่ไม่คิดมูลค่าการใช้งานและมีความยืดหยุ่นสูงมาก
3) รหัสที่เขียนด้วยไพธอนสามารถนำไปรันบนระบบปฏิบัติการอื่นๆได้ เช่น ลินุกซ์ ไมโครซอฟต์วินโดวส์ (Microsoft windows) เป็นต้น
4) ไพธอนสนับสนุนเทคโนโลยีของไมโครซอฟต์วินโดวส์
5) ไพธอนสามารถใช้ได้กับระบบเอกซ์วินโดวส์ (X windows) ไมโครซอฟต์ วินโดวส์ และแมคอินทอช (Macintosh) การใช้คำสั่งเอพีไอ อินเตอร์เฟสทีเค (Tkinter API) ช่วยให้ผู้พัฒนาโปรแกรมไม่ต้องแก้ไขรหัสเมื่อนำไปดำเนินงานบนระบบปฏิบัติการอื่นๆ
6) ไพธอนเป็นชนิดไดนามิก คือสามารถเปลี่ยนชนิดข้อมูลได้ง่ายและสะดวก
7) ไพธอนมีโครงสร้างของข้อมูลที่สามารถใช้ได้ในไพธอน ประกอบด้วย ลิสต์ ดิกชันนารี สตริง ที่ง่ายต่อการใช้งานและมีประสิทธิภาพสูง

8) ไพธอนมีเครื่องมือต่างๆมากมาย เช่น การประมวลผลแฟ้มข้อความ การเรียงข้อมูล การเชื่อมต่อสตริง การตรวจสอบเงื่อนไขของข้อความ การแทนคำ เป็นต้น
9) ไพธอนมีมอดูลสำหรับจัดการการส่งข้อมูล
10) ไพธอนจัดการหน่วยความจำอย่างอัตโนมัติ สามารถจักการพื้นที่หน่วยความจำที่ไม่ต่อเนื่องให้ทำงานได้อย่างมีประสิทธิภาพ
11) ไพธอนอนุญาตให้ฝังชุดคำสั่งของไพธอนเอาไว้ภายในรหัสภาษาซี/ซีพลัสพลัสได้
12) ไพธอนอนุญาตให้ผู้พัฒนาโปรแกรมสร้างไดนามิกลิงก์ไลบรารี (Dynamic link libray, DLL) เพื่อใช้ร่วมกับไพธอน
13) ไพธอนมีมอดูลสนับสนุนเกี่ยวกับระบบเครือข่าย (Network) กระบวนการ (Process) สายโยงใย (Thread) เอกซ์เอ็มแอล ส่วนต่อประสานกราฟิกกับผู้ใช้ (Graphical user interface, GUI) และอื่นๆ
14) ไพธอนประกอบด้วยมอดูลสำหรับสร้างอินเทอร์เน็ตสคริปต์ (Internet script) และติดต่อกับอินเทอร์เน็ตผ่านเต้ารับ (Sockets) และทำหน้าที่เป็นซีจีไอสคริปต์ (CGI Script) ตลอดจนใช้งานคำสั่งเอฟทีพี เอกซ์เอ็มแอล และอื่นๆ
15) ไพธอนสามารถประมวลผลทางด้านวิทยาศาสตร์ และวิศวกรรมศาสตร์ได้อย่างมีประสิทธิภาพ
16) ไพธอนมีฟังก์ชันสนับสนุนฐานข้อมูล เช่น มายเอสคิวเอล ไซเบส (Sybase) ออราเคิล (Oracle) อินฟอร์มิก (Informix) การเชื่อมต่อฐานข้อมูลแบบเปิด (ODBC) และอื่นๆ
17) ไพธอนมีไลบรารีสนับสนุนด้านการสร้างภาพกราฟิก เช่น ทำภาพเบลอ หรือภาพชัด หรือเขียนข้อความบนภาพ ตลอดจนบันถึกไฟล์ในรูปแบบต่างๆได้อย่างสะดวกและมีประสิทธิภาพ
18) ไพธอนมีไลบรารีสำหรับสร้างเอกสารพีดีเอฟ (PDF) โดยไม่ต้องติดตั้งแอโครแบตไรเตอร์ (Acrobat writer)
19) ไพธอนมีไลบรารีสำหรับสร้างช็อกเวฟแฟลช (Shockwaves flash, SWF) โดยไม่ต้องติดตั้งแมโครมีเดียแฟลช (Macromedia flash) [7]

2.5.3 ภาษาจาวาสคริปต์

จาวาสคริปต์เป็นภาษาสำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต โดยสามารถเขียนโปรแกรมจาวาสคริปต์เพิ่มเข้าไปในหน้าเว็บเพื่อใช้ประโยชน์สำหรับงานด้านต่างๆทั้งการคำนวณ การแสดงผล การรับและส่งข้อมูล และที่สำคัญคือสามารถโต้ตอบกับผู้ใช้งานได้อย่างทันที นอกจากนี้ยังมีความสามารถด้านอื่นๆอีกหลายประการที่ช่วยสร้างความน่าสนใจให้กับหน้าเว็บได้อย่างมาก

จาวาสคริปต์เป็นภาษาสคริปต์เชิงวัตถุที่มีเป้าหมายในการออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สามารถทำงานข้ามแพลตฟอร์มได้ ทำงานร่วมกับภาษาเอชทีเอ็มแอล และภาษาจาวาได้ทั้งทางผู้รับบริการและทางผู้ให้บริการ

คุณลักษณะเด่นของภาษาจาวาสคริปต์ มีดังนี้

1) ความง่าย (Simple) ภาษาจาวาเป็นภาษาที่ง่ายต่อการศึกษาและการพัฒนาโปรแกรม ทั้งนี้เพราะภาษาจาวาพัฒนาโดยตัดข้อด้อยของภาษาซีพลัสพลัสออกไป เช่น เรื่องของการใช้รูปแบบการเปลี่ยนรูปปุ่มชี้เป็นรูปอื่นเมื่อมีการเชื่อมโยงไปยังข้อความอื่น
2) ภาษาเชิงวัตถุ (Object oriented) ภาษาจาวาเป็นภาษาคอมพิวเตอร์เชิงวัตถุที่สมบูรณ์ โดยมีคุณลักษณะเด่นคือ การสืบทอด การห่อหุ้ม และการมีได้หลายรูปแบบ
3) การกระจาย (Distributed) ภาษาจาวามีชุดคำสั่งที่เป็นโปรแกรมสำเร็จ (Package) ในการจัดการกับโปรโตคอล TCP/IP ทำให้สามารถพัฒนาโปรแกรมผ่านระบบเครือข่ายอินเตอร์เน็ตได้ง่าย
4) การป้องกันข้อผิดพลาด (Robust) ภาษาจาวาออกแบบมาเพื่อเป็นโปรแกรมที่น่าเชื่อถือ โดยมีการตรวจสอบการผิดพลาดที่อาจเกิดขึ้นในขั้นตอนต่างๆ
5) ความปลอดภัย (Secure) ภาษาจาวาออกแบบมาเพื่อพัฒนาโปรแกรมบนระบบเครือข่าย ดังนั้นจึงต้องสร้างระบบป้องกันไวรัส (Virus) และการแก้ไขโปรแกรมจากภายนอก
6) เคลื่อนย้ายง่าย (Portable) ข้อกำหนดของภาษาจาวาจะไม่ขึ้นอยู่กับระบบปฏิบัติการใดโดยเฉพาะ ดังนั้นโปรแกรมที่เขียนด้วยภาษาจาวาจึงสามารถประมวลผลได้กับระบบคอมพิวเตอร์ทุกประเภท [8]

2.6 ตัวรับรู้ความดันส่วนต่าง

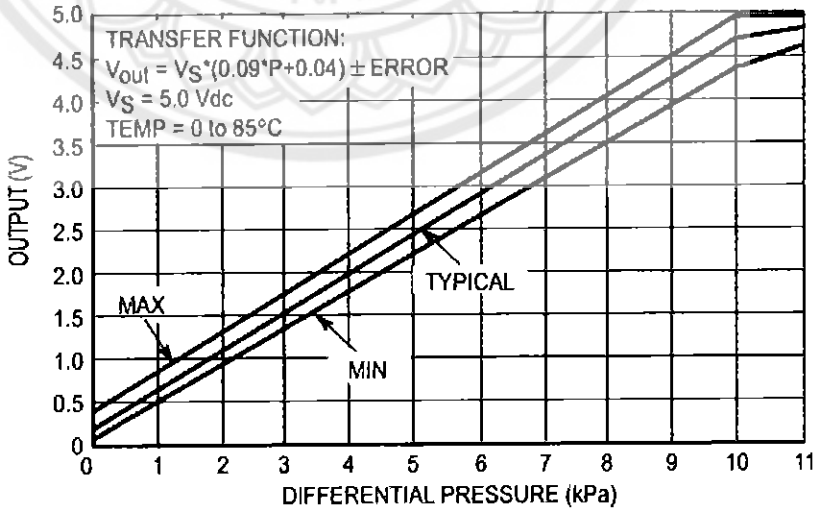
โครงงานนี้ได้นำตัวรับรู้จำแนกตามข้อมูลหรือวัตถุประสงค์ในการวัดเข้ามาใช้งาน เนื่องจากเป็นการวัดระดับน้ำในระดับต่างๆเท่านั้น โดยการแบ่งชนิดของตัวรับรู้อาศัยหลักเกณฑ์ต่างๆ ดังต่อไปนี้

1) การจำแนกตามความต้องการพลังงาน
   ก) แบบแอกทีฟ (Active sensors) เป็นทรานสดิวเซอร์ที่สามารถปล่อยพลังงานเองได้ เช่น เทอร์โมคัปเปิ้ล (Thermocouple) เพียซโซ (Piezo) และเซลล์แสงอาทิตย์ (Solar cell) เป็นต้น อุปกรณ์เหล่านี้ไม่ต้องมีแหล่งจ่ายกำลังจากภายนอกที่สอดคล้องกับปริมาณที่วัดโดยให้ก็สามารถให้สัญญาณแรงดันหรือกระแส
   ข) แบบพาสซีฟ (Passive sensors) ต้องใช้แหล่งจ่ายจากภายนอกเพื่อให้ตรวจรู้ได้ เช่น ตัวรับรู้ที่ใช้หลักการเปลี่ยนค่าความต้านทาน ค่าความจุ ค่าความเหนี่ยวนำ เป็นต้น
2) การจำแนกตามลักษณะกลไกในการทำงาน
   ก) การเปลี่ยนแปลงค่าความจุ (Variable capacitance transducer)
   ข) การเปลี่ยนแปลงค่าความเหนี่ยวนำ (Variable inductance transducer)
   ค) การเปลี่ยนแปลงค่าความต้านทาน (Variable resistance transducer)
3) การจำแนกตามชนิดของการเปลี่ยนแปลงพลังงาน
   ก) เปลี่ยนพลังงานกลเป็นพลังงานไฟฟ้า
   ข) เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล
   ค) เปลี่ยนพลังงานแสงเป็นพลังงานไฟฟ้า
   ง) เปลี่ยนพลังงานความร้อนเป็นพลังงานไฟฟ้า
4) การจำแนกตามชนิดของสัญญาณที่ใช้
   ก) แบบแอนะล็อก ซึ่งให้สัญญาณเป็นแบบต่อเนื่อง
   ข) แบบไบนารี ซึ่งให้สัญญาณแบบเปิด-ปิด (ON-OFF)
   ค) แบบดิจิตอล ซึ่งให้สัญญาณเป็นแบบดิจิตอล
5) การจำแนกตามตำแหน่งที่ใช้ในระบบ
   ก) ทรานสดิวเซอร์ด้านเข้า (Input transducer) อยู่ทางด้านเข้าของระบบเครื่องมือ เช่น ไมโครโฟน เป็นต้น
   ข) ทรานสดิวเซอร์ด้านออก (Output transducer) เช่น ลำโพงของระบบเครื่องขยายเสียง
6) การจำแนกตามข้อมูลหรือวัตถุประสงค์ในการวัด ได้แก่ ทรานสดิวเซอร์ที่ใช้วัดการเคลื่อนที่ อุณหภูมิ ความดัน อัตราการไหล และตำแหน่ง เป็นต้น

ในโครงงานนี้ได้เลือกใช้ตัวรับรู้ความดันส่วนต่าง รุ่น MPX5010DP มาใช้วัดระดับน้ำ เนื่องจากตัวรับรู้ความดันส่วนต่างมีเสถียรภาพและความแม่นยำสูงในการวัดระดับน้ำ โดยใช้หลักการที่ความลึกของระดับน้ำเดียวกันมีค่าความดันเท่ากันทุกทิศทางและตั้งฉากกับผิวภาชนะที่บรรจุอยู่ รูปร่างของภาชนะที่บรรจุน้ำจึงไม่ส่งผลต่อค่าความดันของน้ำ ทำให้สามารถวัดความดันน้ำได้ทั้งในน้ำนิ่งหรือน้ำไหล อย่างไรก็ตามความพลิ้วบริเวณผิวน้ำอาจส่งผลต่อความคลาดเคลื่อนของค่าความดันที่วัดโดยเฉพาะในขณะสูบน้ำเข้าในถังน้ำ คุณสมบัติหลักของ MPX5010DP มีดังนี้

1) ใช้หลักการวัดความดันส่วนต่าง (Differential pressure)
2) จำนวนขาทั้งหมด 6 ขา
3) ใช้งานในช่วงความดัน 0 - 10 kPa
4) ช่วงแรงดันไฟฟ้า 4.75 - 5.25 V

ตัวรับรู้ความดันส่วนต่าง รุ่น MPX5010DP เปรียบเทียบค่าความดันของน้ำกับความดันบรรยากาศภายในตัวรับรู้แล้วให้ค่าออกมาเป็นแรงดันไฟฟ้าตามกราฟคุณลักษณะดังรูปที่ 2.6 โดยแกนตั้งเป็นแรงดันเอาท์พุตของตัวรับรู้ ส่วนแกนนอนเป็นผลต่างระหว่างค่าความดันของน้ำกับความดันบรรยากาศ โดยผลต่างมีค่าอยู่ในช่วง 10 kPA ซึ่งสัมพันธ์กับความสูงของระดับน้ำ 1 m ค่าต่ำสุดของกราฟอยู่ในช่วง 0 - 0.2 V อันเกิดจากความคลาดเคลื่อนในการทดสอบ ในขณะที่ค่าสูงสุดอยู่ในช่วง 4.75 - 5 V อย่างไรก็ตามเนื่องจากตัวรับรู้ความดันส่วนต่างนี้ได้รับแรงดันไฟเลี้ยง +5 V ดังนั้นจากกราฟคุณลักษณะจะเห็นว่า MPX5010DP สามารถให้แรงดันเอาท์พุตสูงสุดไม่เกินค่าไฟเลี้ยงนั่นเอง



รูปที่ 2.6 กราฟคุณลักษณะของตัวรับรู้ความดันส่วนต่าง รุ่น MPX5010DP [9]

2.7 การสร้างส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ต

ในโครงงานนี้จะมีการสร้างส่วนที่ใช้โต้ตอบกับผู้ใช้งาน ซึ่งก็คือส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ต จึงมีการเลือกใช้โปรแกรมอะโดบี ดรีมวีฟเวอร์ (Adobe Dreamweaver) มาใช้ในการเขียนหน้าต่างเพื่อโต้ตอบกับผู้ใช้ ให้มีความเหมาะสมและอำนวยความสะดวกต่อผู้ใช้งาน

โปรแกรมอะโดบี ดรีมวีฟเวอร์ เป็นโปรแกรมประเภทออกแบบเว็บ ซึ่งมีคุณสมบัติในการใช้งานประเภทออกแบบโปรแกรมมาอย่างไร ก็เห็นงานเป็นแบบนั้น ในการใช้งานโปรแกรมจะอำนวยความสะดวกให้กับการออกแบบหน้าเว็บเป็นอย่างมาก เนื่องจากไม่จำเป็นต้องเรียนรู้รหัสของเอชทีเอ็มแอลก็สามารถสร้างเว็บไซต์ได้

ในการเขียนหน้าเว็บ จะมีลักษณะคล้ายกับการพิมพ์งานในโปรแกรมแก้ไขข้อความ (Text Editor) ทั่วไป คือจะเรียงชิดซ้ายบนตลอดเวลา ไม่สามารถย้าย หรือนำไปวางตำแหน่งที่ต้องการได้ทันทีเหมือนโปรแกรมกราฟิก เพราะฉะนั้นหากต้องการจัดวางรูปแบบตามที่ต้องการ ก็ใช้ตาราง (Table) เข้ามาช่วยจัดตำแหน่ง ซึ่งเมื่อมีการจัดวางรูปแบบที่ซับซ้อนมากขึ้น การเขียนภาษาเอชทีเอ็มแอลก็ซับซ้อนยิ่งขึ้นเช่นกัน โปรแกรมอะโดบี ดรีมวีฟเวอร์อาจจะไม่สามารถเขียนเว็บได้ตามที่ต้องการทั้งหมด วิธีการแก้ไขปัญหาที่ดีที่สุดคือ ควรจะเรียนรู้หลักการของภาษาเอชทีเอ็มแอลไปด้วย และคุณลักษณะของโปรแกรมอะโดบี ดรีมวีฟเวอร์สรุปได้ดังนี้

1) สนับสนุนการทำงานตามแบบที่ต้องการ ซึ่งก็คือเว็บที่เขียนบนหน้าจอโปรแกรมอะโดบี ดรีมวีฟเวอร์ จะแสดงแบบเดียวกันกับหน้าเว็บจริง ช่วยให้สามารถเขียนหน้าเว็บง่ายขึ้น ไม่ต้องเขียนรหัสเอชทีเอ็มแอล
2) มีเครื่องมือในการช่วยสร้างหน้าเว็บที่มีความยืดหยุ่นสูง
3) สนับสนุนภาษาสคริปต์ต่างๆทั้งด้านผู้รับบริการและผู้ให้บริการ เช่น จาวา เอเอสพี (Active server page, ASP) พีเอชพี โปรโตคอลเครือข่ายคอมพิวเตอร์ (Common gateway interface, CGI) และวิชวลเบสิกสคริปต์ (Visual basic script, VBScript)
4) มีเครื่องมือในการอัพโหลดหน้าเว็บไปที่เครื่องเซิร์ฟเวอร์ เพื่อทำการเผยแพร่งานที่เราสร้างในอินเทอร์เน็ต โดยการส่งผ่านเอฟทีพี (File transfer protocol, FTP) หรือการใช้โปรแกรมเอฟทีพีภายนอกช่วย
5) รองรับมัลติมีเดีย เช่น การใส่เสียง การแทรกไฟล์วิดีโอและการใช้งานร่วมกับโปรแกรมอะโดบี แฟลช (Adobe Flash)
6) โปรแกรมอะโดบี ดรีมวีฟเวอร์ยังเป็นโปรแกรมที่ช่วยจัดการกับเว็บไซต์ได้ดี โดยมีเครื่องมือมากมาย เช่น
   6.1) เอฟทีพี สามารถแก้ไขหน้าเว็บและส่งไปแสดงผลที่เซิร์ฟเวอร์ได้ทันที เพราะโปรแกรมอะโดบี ดรีมวีฟเวอร์จะติดต่อกับเซิร์ฟเวอร์ให้เราและแสดงไฟล์ที่อยู่

ในเซิร์ฟเวอร์ให้เห็นและแก้ไขได้ทันที่ที่ต้องการถือเป็นเครื่องมือที่สะดวกมาก นอกจากนั้นยังช่วยให้มีข้อมูลของเว็บไซต์สำรองไว้ในเครื่องตลอดด้วย

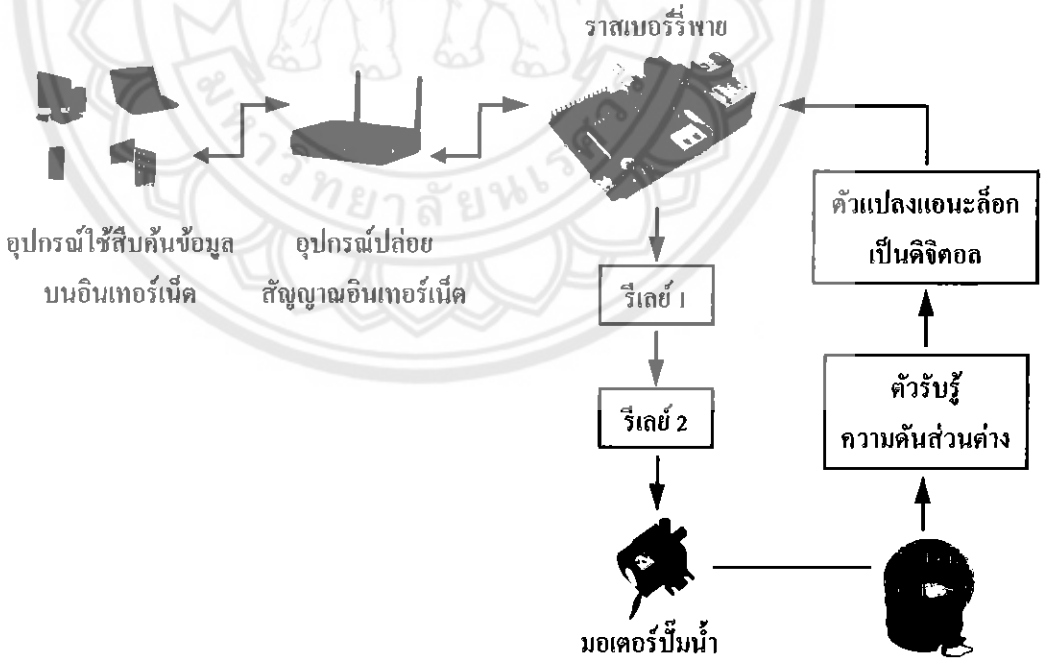
6.2) แผนที่เว็บไซต์ (Site map) เป็นเครื่องมือที่ช่วยแสดงผลเว็บไซต์ของเราให้เป็นรูปร่างขึ้นมา โดยจะแสดงให้เห็นว่าหน้าใดเชื่อมโยง (Link) ไปยังหน้าใดบ้าง โดยสามารถย้าย หรือเปลี่ยนแปลงการเชื่อมโยงได้ โดยที่โปรแกรมอะโดบีดรีมวีฟเวอร์จะทำการเขียนรหัสใหม่ทันที ถือเป็นเครื่องมือที่ดีมาก เพราะความจริงแล้วในการสร้างหน้าเว็บต้องมีการแก้ไขตลอดเพื่ออัพเดตข้อมูล [10]

บทที่ 3
การออกแบบและสร้างระบบควบคุมปั๊มน้ำร่วมกับระบบแลนไร้สาย

หลังจากศึกษาหลักการทำงานของราสเบอร์รี่พายแล้วจึงเริ่มการกำหนดขั้นตอนการดำเนินงานด้วยการเขียนผังงานเพื่อออกแบบการทำงานของระบบทั้งหมด หลังจากนั้นทำการเชื่อมต่อราสเบอร์รี่พายเข้ากับระบบเครือข่ายผ่านทางตัวรับส่งแบบยูเอสบีของระบบแลนไร้สาย แล้วติดตั้งการทำงานให้กับราสเบอร์รี่พาย และใช้โปรแกรมอะโดบี ดรีมวีฟเวอร์ ช่วยในการสร้างส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ตสำหรับควบคุมโต้ตอบการทำงานร่วมกับราสเบอร์รี่พาย รวมถึงการออกแบบจำลองของระบบเปิดและปิดปั๊มน้ำ

3.1 การออกแบบขั้นตอนการทำงานของระบบ

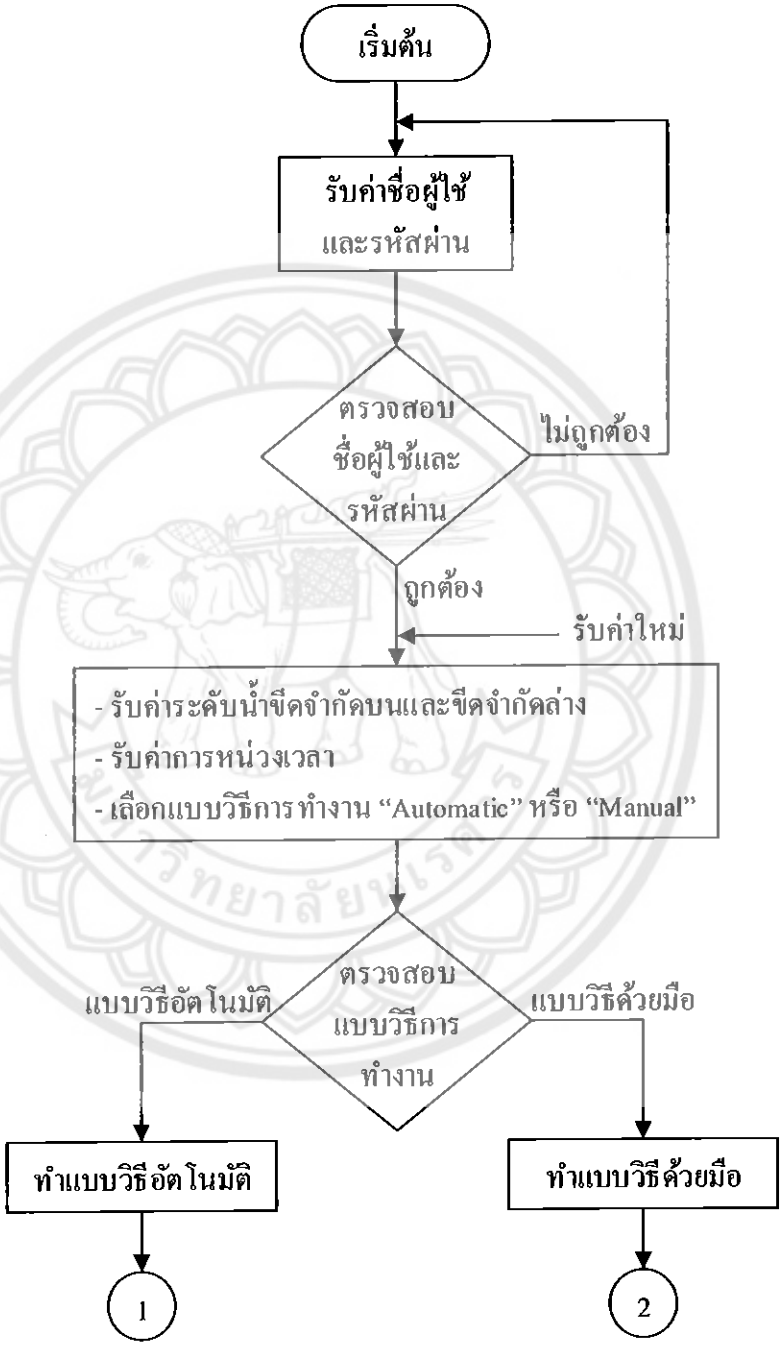
ขั้นตอนการทำงานของระบบควบคุมการเปิดและปิดปั๊มน้ำด้วยวิธีการควบคุมทางไกลผ่านระบบแลนไร้สายแสดงด้วยแผนภาพดังรูปที่ 3.1



รูปที่ 3.1 ขั้นตอนการทำงานระบบเปิดและปิดปั๊มน้ำ โดยการควบคุมผ่านระบบแลนไร้สาย

ผู้ใช้สามารถเชื่อมต่อให้ระบบเริ่มทำงานด้วยหน้าต่างส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ตผ่านทางเบราเซอร์บนอุปกรณ์ที่สามารถเชื่อมต่อและสืบค้นข้อมูลทางอินเทอร์เน็ตได้ ซึ่งเป็นการเชื่อมต่อกับระบบแลนไร้สายตัวเดียวกับที่ติดต่อกับราสเบอร์รี่พาย โดยราสเบอร์รี่พายจะถูกเชื่อมต่อกับปั๊มน้ำ เมื่อมีการส่งข้อมูลให้กับราสเบอร์รี่พายจะประมวลผล เมื่อเข้าเงื่อนไขการเปิดปั๊มน้ำราสเบอร์รี่พายจะสั่งให้รีเลย์ทำงาน โดยต่อหน้าสัมผัสเพื่อเริ่มเดินเครื่องปั๊มน้ำ น้ำจะถูกสูบเข้าไปยังถังเก็บน้ำซึ่งมีตัวรับรู้ความดันส่วนต่างคอยตรวจวัดระดับน้ำ โดยเปรียบเทียบความดันที่ได้จากภายในสายยางกับความดันบรรยากาศแล้วแปลงค่าออกมาเป็นค่าแรงดันไฟฟ้า เพื่อส่งให้ราสเบอร์รี่พายประมวลผลและส่งข้อมูลกลับไปแสดงผลยังหน้าต่างส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ตโดยผ่านทางระบบแลนไร้สาย เพื่อแสดงสถานะการทำงานของปั๊มน้ำผ่านหน้าจอว่ามีการทำงาน หลังจากที่ระดับน้ำเพิ่มสูงจนถึงขีดจำกัดบน (Upper limit) ที่ผู้ใช้กำหนด ราสเบอร์รี่พายจะสั่งให้รีเลย์เปิดวงจรเพื่อให้ปั๊มน้ำหยุดทำงาน รวมทั้งส่งข้อมูลไปแสดงสถานะการทำงานของปั๊มน้ำ ว่าหยุดการทำงานบนหน้าต่างส่วนต่อประสานกับผู้ใช้ และจะยังคงมีการตรวจสอบระดับน้ำอยู่ โดยเป็นการตรวจสอบจากค่าระดับน้ำลดต่ำกว่าขีดจำกัดล่าง (Lower limit) ที่ผู้ใช้กำหนด โดยราสเบอร์รี่พายจะสั่งให้รีเลย์ปิดวงจรเพื่อให้ปั๊มน้ำเริ่มการทำงาน รวมทั้งส่งข้อมูลไปแสดงสถานะการทำงานของปั๊มน้ำ ว่ามีการทำงานอีกครั้ง โดยจะเป็นการตรวจสอบเช่นนี้ไปเรื่อยๆและสามารถหยุดการตรวจสอบแบบนี้ทำได้โดยผู้ใช้งาน จากการกดที่ปุ่มหยุดการทำงาน (STOP) ระบบจะหยุดการทำงานและแสดงสถานะการทำงานของปั๊มน้ำ ว่าหยุดการทำงาน

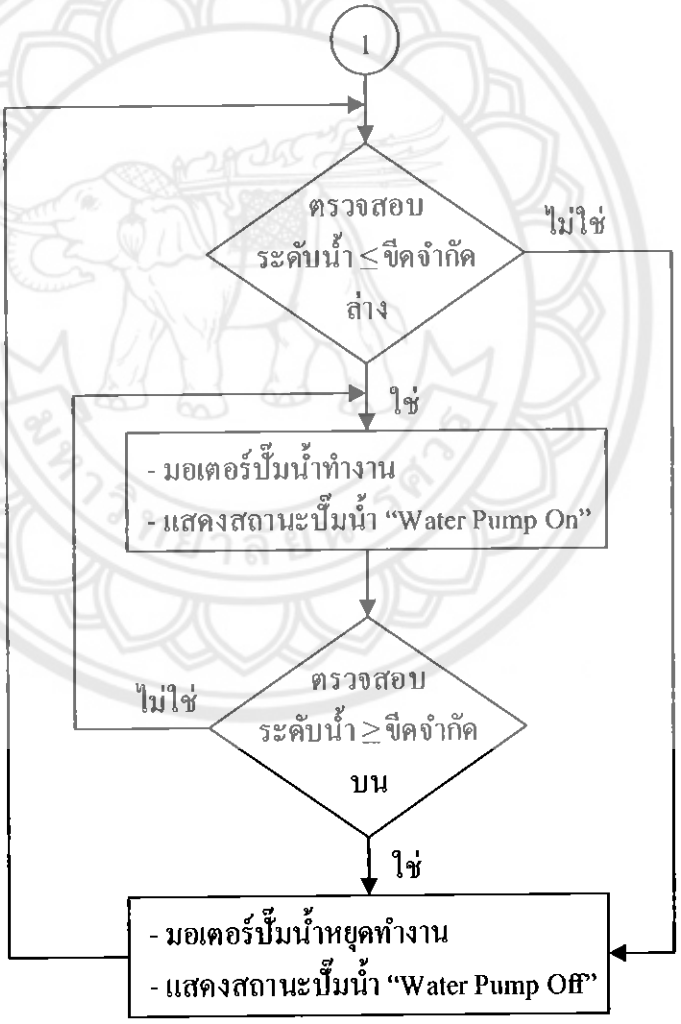
ลำดับการทำงานของส่วนควบคุมที่กำหนดด้วยส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ต แสดงดังรูปที่ 3.2



รูปที่ 3.2 ลำดับการทำงานของส่วนควบคุมที่กำหนดด้วยส่วนต่อประสานกับผู้ใช้

จากแผนผังลำดับการทำงานของส่วนควบคุมในรูปที่ 3.2 เริ่มต้นให้ผู้ใช้งานทำการเข้าสู่ระบบโดยกรอกชื่อผู้ใช้และรหัสผ่านที่ได้ตั้งไว้ เมื่อเข้าสู่ระบบสำเร็จ จะเข้าสู่ส่วนควบคุม ผู้ใช้ต้องกำหนดค่าระดับน้ำขีดจำกัดบนและค่าระดับน้ำขีดจำกัดล่าง การตั้งระยะการหน่วงเวลา รวมไปถึงการเลือกแบบวิธีการทำงาน ซึ่งมีให้เลือก 2 แบบวิธี คือ แบบวิธีอัตโนมัติและแบบวิธีด้วยมือให้กับระบบ จากนั้นข้อมูลถูกส่งออกไปยังราสเบอร์รี่พาย เพื่อประมวลผล ในขณะเดียวกันหลังจากที่มีการรับข้อมูลแล้ว ผู้ใช้สามารถปรับเปลี่ยนการตั้งค่าระดับน้ำ การหน่วงเวลา และแบบวิธีการทำงานได้ใหม่ได้ทุกเมื่อที่ต้องการให้มีการปรับเปลี่ยน

ลำดับการทำงานของส่วนควบคุมที่กำหนดด้วยส่วนต่อประสานกับผู้ใช้แบบวิธีอัตโนมัติแสดงดังรูปที่ 3.3



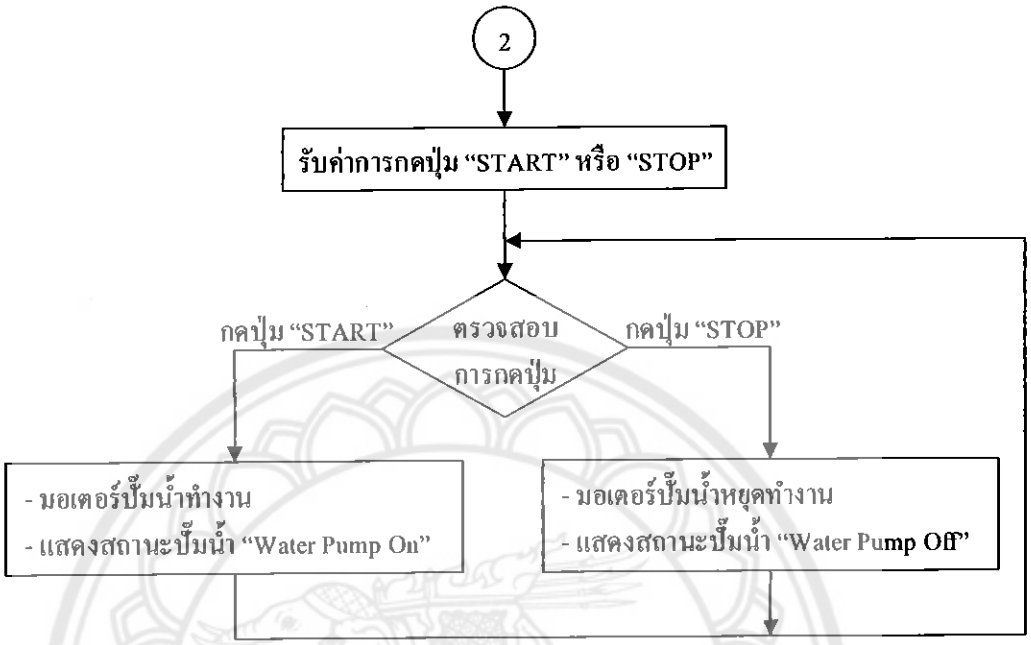
รูปที่ 3.3 ลำดับการทำงานของส่วนควบคุมแบบวิธีการควบคุมอัตโนมัติ

จากแผนผังลำดับการทำงานของส่วนควบคุมในรูปที่ 3.3 เมื่อเข้าสู่แบบวิธีการทำงานแบบอัตโนมัติ ระบบจะเริ่มการทำงานทันที โดยเริ่มจากการตรวจสอบระดับผ่านทางตัวรับรู้ความดันส่วนต่าง ซึ่งจะวัดค่าออกมาเป็นแรงดัน แล้วถูกแปลงค่าด้วยโมดูลแปลงสัญญาณแอนะล็อกเป็นดิจิตอล ส่งค่าข้อมูลเข้าราสเบอร์รี่พาย ประมวลผลออกมา ปรากฏบนส่วนต่อประสานกับผู้ใช้ พร้อมทั้งแสดงสถานะการทำงานของปั๊มน้ำและระดับน้ำจริงที่อยู่ในถัง และนำค่าระดับน้ำนี้มาตรวจสอบตาม 2 เงื่อนไขดังนี้

กรณีที่ 1 ถ้าค่าระดับน้ำในถังน้ำมีค่าสูงกว่าหรือเท่ากับค่าระดับน้ำขีดจำกัดบนที่ผู้ใช้กำหนดไว้ ข้อมูลจะถูกให้ราสเบอร์รี่พายประมวลผลเพื่อสั่งรีเลย์ให้ตัดวงจรเพื่อหยุดการทำงานของของปั๊มน้ำ และแสดงสถานะบนหน้าจอเป็น “Water Pump Off”

กรณีที่ 2 ถ้าค่าระดับน้ำที่มีอยู่ในถังน้ำมีค่าน้อยกว่าค่าระดับน้ำขีดจำกัดล่าง ที่ผู้ใช้กำหนดไว้ จะมีการส่งข้อมูลไปให้ราสเบอร์รี่พาย แล้วสั่งให้รีเลย์ควบคุมให้ปั๊มน้ำทำงาน มีการแสดงสถานะเป็น “Water Pump On” ปรากฏบนหน้าจอ และในขณะที่ปั๊มน้ำมีการทำงานหากระดับน้ำที่มีอยู่ในถังเพิ่มขึ้นจนมีปริมาณมากกว่าหรือเท่ากับระดับน้ำขีดจำกัดบนที่ผู้ใช้กำหนดไว้ จะมีการส่งข้อมูลไปให้ราสเบอร์รี่พาย แล้วสั่งให้รีเลย์ควบคุมให้ปั๊มน้ำหยุดการทำงาน มีการแสดงสถานะบนหน้าจอ เป็น “Water Pump Off” และหลังจากปั๊มน้ำหยุดการทำงานแล้วจะยังคงมีการตรวจสอบระดับน้ำในถังว่ามีปริมาณน้อยกว่าหรือเท่ากับค่าระดับน้ำต่ำสุดที่ผู้ใช้กำหนดหรือไม่ หากระดับน้ำในถังต่ำกว่าหรือเท่ากับค่าระดับน้ำขีดจำกัดล่างที่ผู้ใช้กำหนด ราสเบอร์รี่พายสั่งให้รีเลย์ต่อวงจรให้ปั๊มน้ำทำงาน และแสดงสถานะบนหน้าจอเป็น “Water Pump On” อีกครั้ง

ลำดับการทำงานของส่วนควบคุมที่กำหนดด้วยส่วนต่อประสานกับผู้ใช้แบบวิธีด้วยมือแสดงดังรูปที่ 3.4



รูปที่ 3.4 ลำดับการทำงานของส่วนควบคุมแบบวิธีการควบคุมด้วยมือ

จากแผนผังลำดับการทำงานของส่วนควบคุมในรูปที่ 3.4 เมื่อเข้าสู่แบบวิธีการทำงานด้วยมือ ระบบจะเริ่มการทำงานทันที ขึ้นอยู่กับการเลือกกดปุ่มจากผู้ใช้ แบ่งการทำงานจากการกดปุ่มออกเป็น 1 กรณี คือ

กรณีที่ 1 เมื่อผู้ใช้กดปุ่ม "START" จะมีการส่งข้อมูลไปให้ราสเบอร์รี่พาย แล้วสั่งให้รีเลย์ควบคุมให้ปั๊มน้ำทำงาน มีการแสดงสถานะเป็น "Water Pump On" ปรากฏบนหน้าจอส่วนต่อประสานกับผู้ใช้

กรณีที่ 2 เมื่อผู้ใช้กดปุ่ม "STOP" จะมีการส่งข้อมูลไปให้ราสเบอร์รี่พาย แล้วสั่งให้รีเลย์ควบคุมให้ปั๊มน้ำหยุดการทำงาน มีการแสดงสถานะเป็น "Water Pump Off" ปรากฏบนหน้าจอส่วนต่อประสานกับผู้ใช้

การกดปุ่ม "START" และ ปุ่ม "STOP" สามารถทำได้โดยทันที ขึ้นอยู่ความต้องการของผู้ควบคุมที่ต้องการสั่งการให้ปั๊มน้ำทำงานหรือหยุดการทำงาน

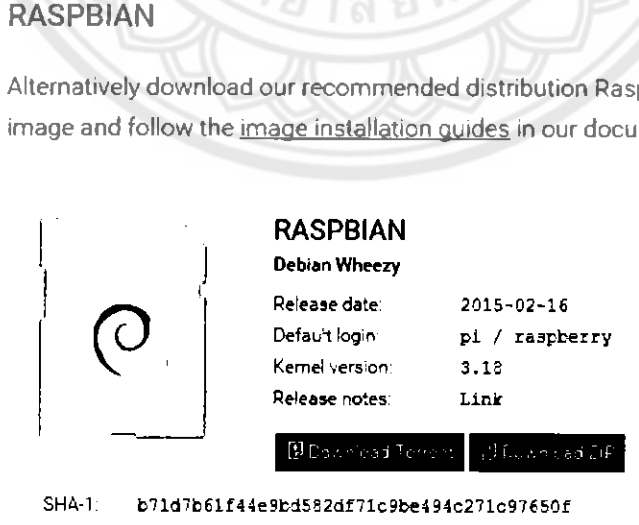
3.2 การออกแบบระบบควบคุมการทำงาน

ระบบควบคุมการเปิดและปิดปั๊มน้ำจะประกอบด้วย 2 ส่วนหลัก คือ ชุดอุปกรณ์ที่ต่อเข้ากับราสเบอร์รี่พาย และส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ต โดยในการเริ่มการทำงานบนราสเบอร์รี่พาย ต้องมีการตั้งค่าเริ่มต้นให้กับราสเบอร์รี่พาย การเขียนคำสั่งเพื่อควบคุมการทำงานและมีการสร้างส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ตให้ทำงานร่วมกับราสเบอร์รี่พาย เพื่อให้สามารถใช้งานบนระบบแลนไร้สายได้ โดยออกแบบให้ส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ตสามารถโต้ตอบกับผู้ใช้งานได้ โดยมีความเหมาะสมกับการใช้งาน มีรายละเอียดดังนี้

3.2.1 การติดตั้งระบบปฏิบัติของราสเบอร์รี่พายลงในเอสดีการ์ด

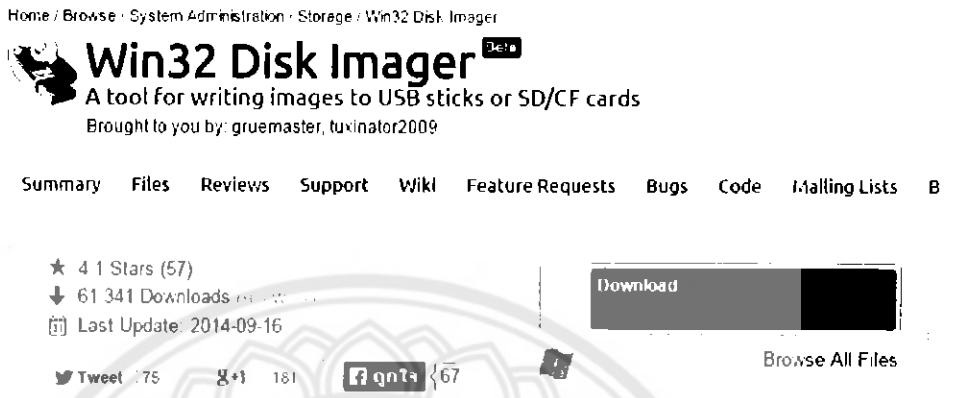
โครงสร้างของราสเบอร์รี่พายที่ถูกออกแบบให้มีช่องเสียบการ์ดความจำได้นั้นเป็นข้อดีอย่างหนึ่ง ในการเก็บข้อมูลรวมถึงการติดตั้งระบบปฏิบัติการ ให้มีความสะดวกต่อการใช้งาน ซึ่งหากต้องการเปลี่ยนระบบปฏิบัติการก็สามารถทำได้โดยการถอดเอสดีการ์ดแล้วเปลี่ยนอันใหม่เข้าไป ซึ่งการติดตั้งระบบปฏิบัติการลงในเอสดีการ์ดมีขั้นตอนดังนี้

1) ทำการดาวน์โหลดไฟล์อิมเมจของระบบปฏิบัติการจาก http://www.raspberrypi.org/downloads เลือกราสเบียน (RASBIAN) ชนิดวีซี (Wheezy) ทำการดาวน์โหลดในรูปแบบ *.zip เมื่อดาวน์โหลดเสร็จทำการแตกไฟล์ จะได้อิมเมจไฟล์ *.img ดังรูปที่ 3.5



รูปที่ 3.5 ไฟล์อิมเมจของระบบปฏิบัติการ

2) ทำการดาวน์โหลดโปรแกรมวิน 32 ดิสก์อิมเมจเจอร์ (Win32Diskimager) จาก http://sourceforge.net/projects/win32diskimager/ เพื่อใช้ในการเขียนอิมเมจไฟล์ลงในเอสดีการ์ด ดังรูปที่ 3.6



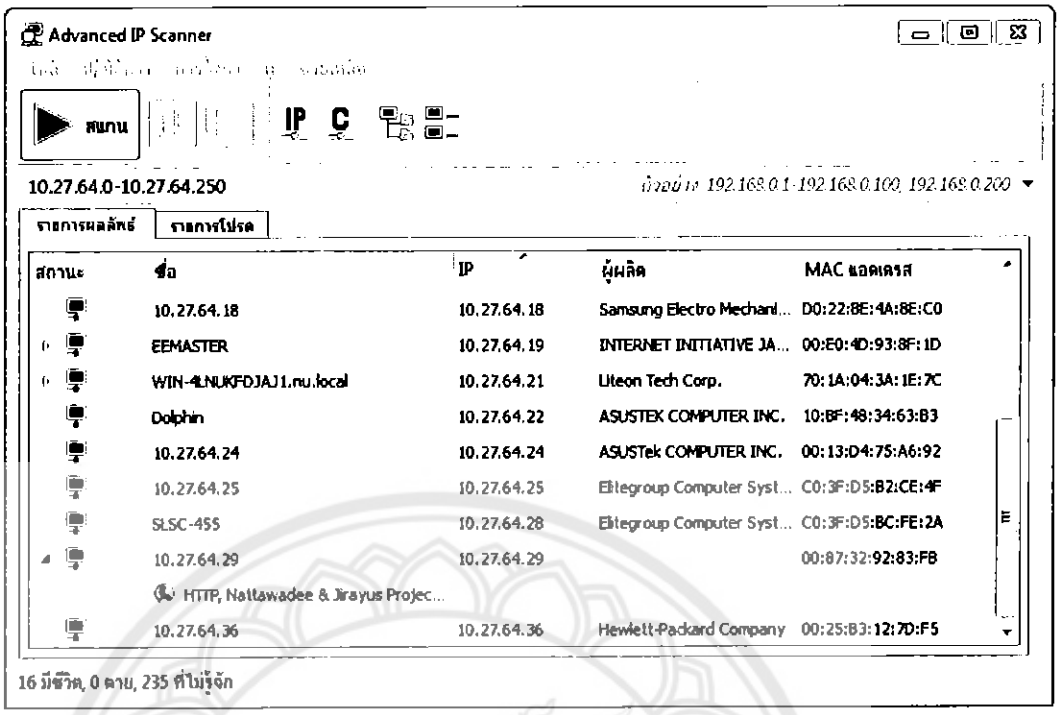
รูปที่ 3.6 โปรแกรมวิน 32 ดิสก์อิมเมจเจอร์

3) เปิดโปรแกรมวิน 32 ดิสก์อิมเมจเจอร์ เพื่อนำไฟล์อิมเมจมาเขียนลงในเอสดีการ์ด

3.2.2 การตั้งค่าราสเบอร์รี่พายให้ทำงานร่วมกับระบบแลนไร้สาย

การตั้งค่าราสเบอร์รี่พายให้สามารถเข้าไปควบคุมราสเบอร์รี่พายโดยใช้อุปกรณ์ภายนอกผ่านระบบแลนไร้สายได้ มีดังต่อไปนี้

1) ตั้งค่าราสเบอร์รี่พายให้รู้จักกับชื่อของระบบแลนไร้สายที่จะใช้งาน โดยใช้โปรแกรมดับเบิลยูพีเอ คอนฟิก (WPA Config) ที่อยู่ภายในตัวราสเบอร์รี่พาย
2) ทำการอนุญาตเอสเอสเอช (Secure Shell, SSH) เพื่อให้สามารถใช้โปรแกรมพุตตี้ (PuTTy) และวินเอสซีพี (WinSCP) โดยใช้โปแกรมเทอร์มินอล แอลเอกซ์ (Terminal LX) พิมพ์ sudo raspi-config และหาเลขที่อยู่แมค (MAC Address) ของราสเบอร์รี่พาย พิมพ์คำสั่ง ifconfig เพื่อจะทราบเลขที่อยู่แมคของราสเบอร์รี่พายนี้
3) เมื่อตั้งค่าข้างต้นเรียบร้อย สามารถเปลี่ยนจากการตั้งค่าผ่านราสเบอร์รี่พายโดยตรงผ่านทางอุปกรณ์ภายนอก เช่น แล็ปท็อปหรือคอมพิวเตอร์ได้ เพื่อง่ายต่อการตั้งค่าและเขียนคำสั่งการใช้งานอื่นๆ
4) ใช้โปรแกรมแอดวานซ์ไอพีสแกนเนอร์ (Advanced IP scanner) ในการค้นหาเลขที่อยู่ไอพี (IP Address) ของราสเบอร์รี่พาย โดยอ้างอิงจากเลขที่อยู่แมค ดังรูปที่ 3.7



รูปที่ 3.7 โปรแกรมแอดวานซ์ไอพีสแกนเนอร์

5) เข้าโปรแกรมพุตตี้ ใส่เลขที่อยู่ไอพีของราสเบอร์รี่พาย เพื่อทำการเข้าระบบ พิมพ์ "pi" ที่ login as: และใส่รหัสผ่าน "raspberry" เพื่อเข้าไปควบคุมราสเบอร์รี่พายผ่านบรรทัดคำสั่ง ดังรูปที่ 3.8

รูปที่ 3.8 โปรแกรมพุตตี้

3.2.3 การเขียนโปรแกรมติดต่ออินพุตหรือเอาท์พุตโดยใช้ไลบรารีไวริ่งพาย

ในการใช้ราสเบอร์รี่พายเพื่อควบคุมอุปกรณ์ใดๆจำเป็นต้องมีการใช้ขา (pin) ในการส่งสัญญาณเพื่อไปขับให้เกิดการทำงาน ซึ่งการอ้างอิงขาอินพุตหรือเอาท์พุตเทียบกับไวริ่งพาย (WiringPi) แสดงดังรูปที่ 3.9 ซึ่งผู้ใช้สามารถเลือกใช้งานขาที่ต้องการให้มีการส่งสัญญาณได้ด้วยตัวเอง และต้องมีการเขียนรหัสคำสั่งให้ตรงกับขาที่เลือกให้มีการส่งสัญญาณ โดยในการเขียนโปรแกรมจะใช้ตามรูปแบบของขาไวริ่งพาย ซึ่งการติดตั้งและการใช้งานไลบรารีไวริ่งพายมีขั้นตอนดังนี้

P1: The Main GPIO connector							
WiringPi Pin	BCM GPIO	Name	Header		Name	BCM GPIO	WiringPi Pin
		3.3v	1	2	5v		
8	Rv1:0 - Rv2:2	SDA	3	4	5v		
9	Rv1:1 - Rv2:3	SCL	5	6	0v		
7	4	GPIO7	7	8		14	15
		0v	9	10		15	16
0	17	GPIO0	11	12	GPIO1	18	1
2	Rv1:21 - Rv2:27	GPIO2	13	14	0v		
3	22	GPIO3	15	16	GPIO4	23	4
		3.3v	17	18	GPIO5	24	5
12	10	MOSI	19	20	0v		
13	9	MISO	21	22	GPIO6	25	6
14	11	SCLK	23	24	CE0	8	10
		0v	25	26	CE1	7	11
WiringPi Pin	BCM GPIO	Name	Header		Name	BCM GPIO	WiringPi Pin

รูปที่ 3.9 ตารางสำหรับเลือกใช้งานอินพุตหรือเอาท์พุตของราสเบอร์รี่พาย

ที่มา: http://wiringpi.com/pins

1) ทำการเข้าสู่ระบบการใช้งานราสเบอร์รี่พาย ในโหมดบรรทัดคำสั่ง จากนั้นพิมพ์คำสั่งเพื่อติดตั้ง git โดยพิมพ์คำสั่ง sudo apt-get install git-core และทำการคัดลอกไลบรารีไวริ่งพาย โดยพิมพ์คำสั่ง git clone git://git.drogon.net/wiringPi

2) จากนั้นพิมพ์คำสั่ง cd wiringPi เพื่อเข้าไปที่ไดเร็กทอรี่ WiringPic และใช้คำสั่ง git pull origin เพื่อตรวจสอบการอัพเดทของไวริ่งพาย จากนั้นเริ่มการคอมไพล์และติดตั้งโดยใช้คำสั่ง ./build

3.2.4 การจัดการไฟล์บนราสเบอร์รี่พาย

ในการจัดการไฟล์ที่อยู่บนเอสดีการ์ดของราสเบอร์รี่พาย จะทำการอัพโหลดและดาวน์โหลดผ่านทางโปรโตคอลเอสเอฟทีพี (SSH file transfer protocol, SFTP) ซึ่งในที่นี้ใช้ผ่านโปรแกรมวินเอสซีพี ซึ่งขั้นตอนดังนี้

1) ทำการดาวน์โหลดโปรแกรมวินเอสซีพี จาก http://winscp.net/eng/download.php แล้วทำการติดตั้งโปรแกรมให้เรียบร้อย หลังจากนั้นเปิดโปรแกรมขึ้นมา ทำการตั้งค่าต่างๆ ดังรูปที่ 3.10 จากนั้นเลือกที่ปุ่ม Login เพื่อเข้าสู่ระบบ

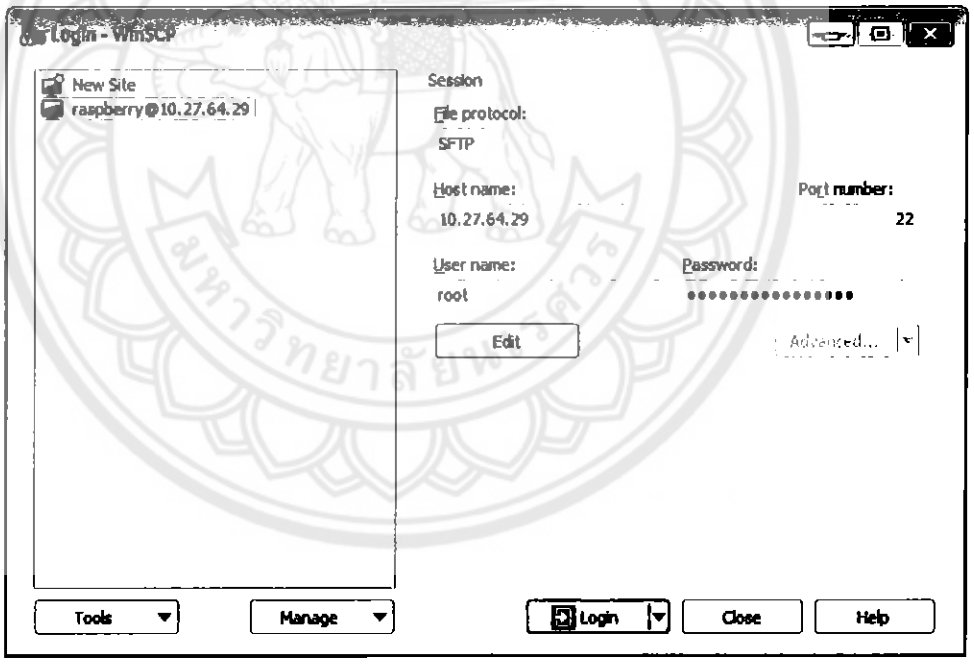
File protocol: SFTP

Host name: 10.27.64.29 (หมายเลขที่อยู่ไอพีของราสเบอร์รี่พาย)

Port number: 22

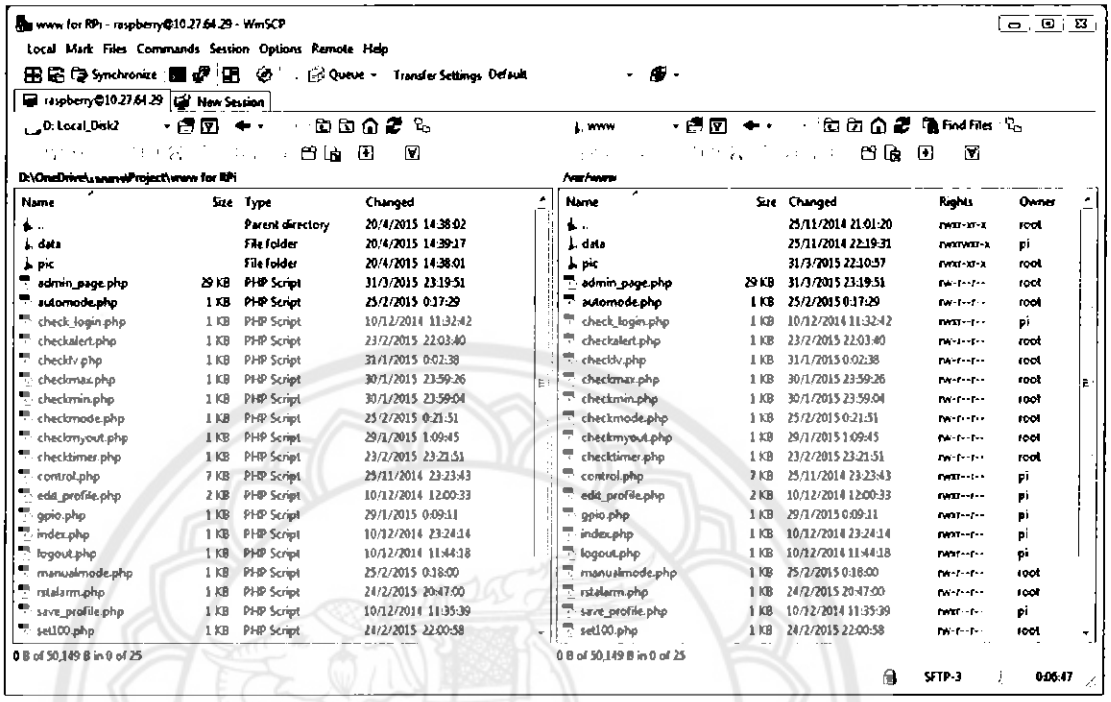
User name: root

Password: root



รูปที่ 3.10 โปรแกรมวินเอสซีพี

2) หลังจากนั้นจะสามารถเห็นไฟล์ที่อยู่ในเอสดีการ์ดของราสเบอร์รี่พาย ดังรูปที่ 3.11 โดยในส่วนนี้ผู้ใช้สามารถจัดการไฟล์ต่างๆได้ตามต้องการ



รูปที่ 3.11 หน้าต่างสำหรับการจัดการไฟล์

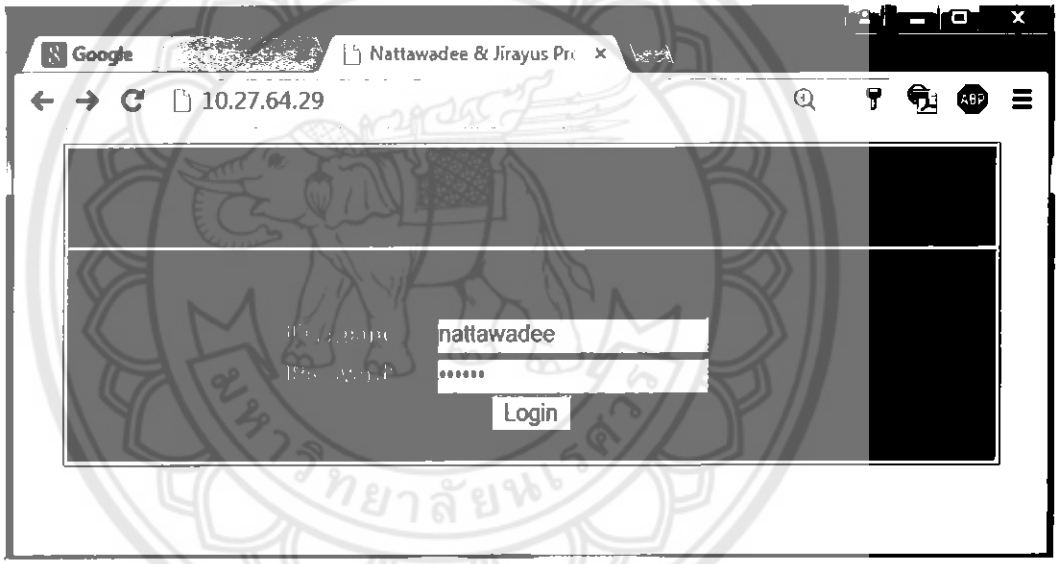
3.2.5 การตั้งค่าราสเบอร์รี่พายร่วมกับส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ต

การสร้างเว็บเซิร์ฟเวอร์บนราสเบอร์รี่พาย เพื่อให้ราสเบอร์รี่พายสามารถโต้ตอบผ่านส่วนประสานกับผู้ใช้บนอินเทอร์เน็ตได้ มีดังต่อไปนี้

1) ทำการลงไฟล์อะแพชี (Apache) ในราสเบอร์รี่พาย โดยผ่านโปรแกรมพุตตี้ ใช้คำสั่ง sudo apt-get install apache2 -y เพื่อทำให้ราสเบอร์รี่พายทำงานผ่านเว็บเซิร์ฟเวอร์ได้ และลงไฟล์พีเอชพี ใช้คำสั่ง sudo apt-get install php5 libapache2-mod-php5 –y เพื่อทำให้เว็บเซิร์ฟเวอร์รู้จักกับพีเอชพี
2) ในการทำงานจะใช้พีเอชพีในการสร้างไฟล์ จึงต้องทำการเปลี่ยนไฟล์ที่เป็นสกุล .html ให้เป็น .php โดยใช้คำสั่ง sudo mv index.html index.php เพื่อที่จะสามารถอ่านไฟล์ที่เป็นพีเอชพีได้
3) ตั้งค่าเพื่อใช้งานเอสเอฟทีพี ผ่านโปรแกรมวินเอสซีพี โดยใช้โปรแกรมพุตตี้ พิมพ์คำสั่ง sudo chown pi /var/www เพื่อให้ผู้ใช้ (pi) มีสิทธิในการจัดการไฟล์ที่อยู่ในโฟลเดอร์ /var/www ได้

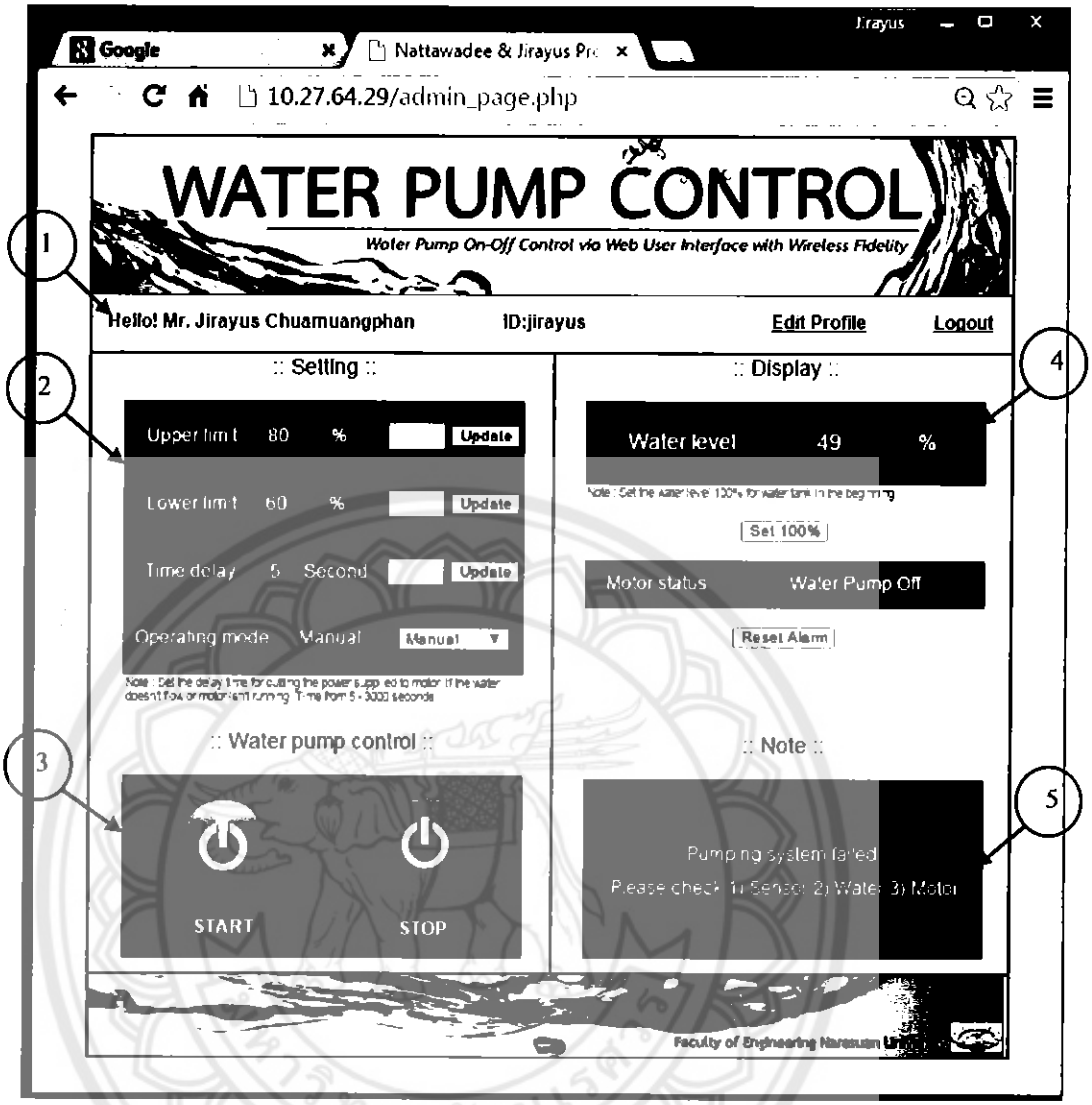
4) การเขียนโปรแกรมติดต่อกับช่องสำหรับเชื่อมต่ออินพุตเอาท์พุตบนราสเบอร์รี่พาย โดยลงชุดคำสั่งไวริ่งพายในโปรแกรมพุตตี้
5) นำไฟล์พีเอชพีที่ได้เขียนคำสั่งติดต่อกับช่องสำหรับเชื่อมต่ออินพุตหรือเอาท์พุต ที่อยู่ในราสเบอร์รี่พาย ย้ายมาปรับแต่งในโปรแกรมอะโดบี ครีมวีฟเวอร์ บนคอมพิวเตอร์ ให้ออกมาเป็นหน้าต่างส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ตตามที่ต้องการ
6) บันทึกไฟล์ส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ตเป็นไฟล์พีเอชพี ย้ายไฟล์กลับเข้าไปที่ราสเบอร์รี่พาย เพื่อใช้งาน

ในส่วนของการสั่งการมีหน้าต่างของส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ต นำไฟล์พีเอชพีของราสเบอร์รี่พายมาปรับแต่งในโปรแกรมอะโดบี ครีมวีฟเวอร์ โดยเริ่มจากส่วนที่เป็นการเข้าสู่ระบบ (Login) เพื่อระบุชื่อผู้ใช้และเป็นการรักษาความปลอดภัยให้กับผู้ใช้ดังรูปที่ 3.12



รูปที่ 3.12 หน้าต่างส่วนต่อประสานกับผู้ใช้สำหรับเข้าสู่ระบบ

เมื่อเข้าสู่ระบบแล้วจะเป็นหน้าต่างที่ให้ผู้ใช้สามารถตั้งค่าการควบคุมการทำงาน และแสดงสถานะการทำงานในปัจจุบันดังรูปที่ 3.13



รูปที่ 3.13 หน้าต่างส่วนต่อประสานกับผู้ใช้สำหรับการควบคุม

จากรูปที่ 3.13 หน้าต่างสำหรับใช้ในการควบคุม และแสดงสถานะการทำงานของปั๊มน้ำ จะประกอบด้วย 5 ส่วนหลัก ดังนี้

ส่วนที่ 1 คือ ชื่อผู้ใช้ เป็นส่วนที่จะแสดงชื่อผู้ที่เข้าสู่ระบบ และสามารถออกจากระบบได้โดยกดที่ออกจากระบบ (Logout)

ส่วนที่ 2 คือ ส่วนของตั้งค่าคุณลักษณะของการทำงานได้แก่การตั้งค่าความสูงของระดับน้ำในถังน้ำโดยการกำหนดขีดจำกัดบน และขีดจำกัดล่าง การตั้งค่าเรื่องของการหน่วงเวลา เพื่อเป็นการตรวจสอบการทำงานของปั๊มน้ำ ตัวรับรู้ รวมไปถึงอุปกรณ์อื่นๆที่เกี่ยวข้องในระบบทางอ้อมด้วยวิธีการหน่วงเวลาเพื่อดูว่าระดับน้ำมีการเปลี่ยนแปลงหรือไม่ และส่วนของแบบวิธีการทำงานในที่นี้มีให้เลือก 2 วิธีคือ แบบวิธีการควบคุมอัตโนมัติและแบบวิธีการควบคุมด้วยมือ

ส่วนที่ 3 คือ ปุ่มเปิดและปิดปั๊มน้ำ โดยกดปุ่มสีเขียว (ON) ให้ปั๊มน้ำเริ่มทำงาน และปุ่มสีแดง (OFF) ให้ปั๊มน้ำหยุดทำงาน เป็นการทำงานร่วมกับการทำงานแบบควบคุมด้วยมือ โดยผู้ใช้จะเปิดหรือปิดปั๊มน้ำ ก็สามารถควบคุมผ่านส่วนนี้ รวมไปถึงการใช้งานปุ่มสีแดง (OFF) ให้ปั๊มน้ำหยุดทำงานซึ่งสามารถทำงานร่วมกับแบบควบคุมอัตโนมัติ เพื่อเป็นการปิดการทำงานของปั๊มน้ำหากมีเหตุการณ์ฉุกเฉิน

ส่วนที่ 4 คือ ส่วนของการแสดงผล มีการแสดงค่าของระดับน้ำในถังน้ำ แสดงค่าออกมาในรูปแบบเปอร์เซ็นต์ โดยการแสดงผลแบบเปอร์เซ็นต์เป็นการอำนวยความสะดวกในการนำไปใช้งานได้กับถังน้ำที่มีขนาดที่แตกต่างกันออกไป โดยในการเริ่มต้นการใช้ที่ถังน้ำขนาดไม่เท่ากันนั้นควรมีการตั้งค่าที่ระดับน้ำเต็มถังให้มีปริมาณ 100% ด้วยปุ่ม Set 100% ที่อยู่ถัดลงมาจากส่วนแสดงระดับน้ำในถัง เพื่อความแม่นยำในการอ่านค่าระดับน้ำในถัง และมีส่วนของการแสดงสถานะการทำงานของปั๊มน้ำว่าทำงานอยู่หรือไม่ด้วย

ส่วนที่ 5 คือ ส่วนของแจ้งเตือน จากการหน่วงเวลาที่มีไว้เพื่อเป็นการตรวจสอบการทำงานของปั๊มน้ำ ตัวรับรู้ รวมไปถึงอุปกรณ์อื่นๆที่เกี่ยวข้องในระบบทางอ้อม หากครบเวลาที่ตั้งไว้แล้วระดับน้ำยังไม่มีการเปลี่ยนแปลง ในส่วนนี้จะมีการแสดงความผิดปกติที่เกิดขึ้นในระบบ เพื่อให้ผู้ใช้ตรวจสอบอุปกรณ์ที่เกี่ยวข้องว่าอุปกรณ์ใดที่มีการทำงานความผิดปกติ เมื่อผู้ใช้ทำงานตรวจสอบและทำการแก้ไขแล้ว จึงจะกลับมาสั่งการระบบนี้ได้อีกครั้ง หลังจากการกดปุ่ม Reset Alarm ที่อยู่เหนือขึ้นไปของส่วนแสดงความผิดปกตินี้

3.2.6 การประมวลผลและควบคุมรีเลย์ด้วยราสเบอร์รี่พาย

1) ราสเบอร์รี่พาย

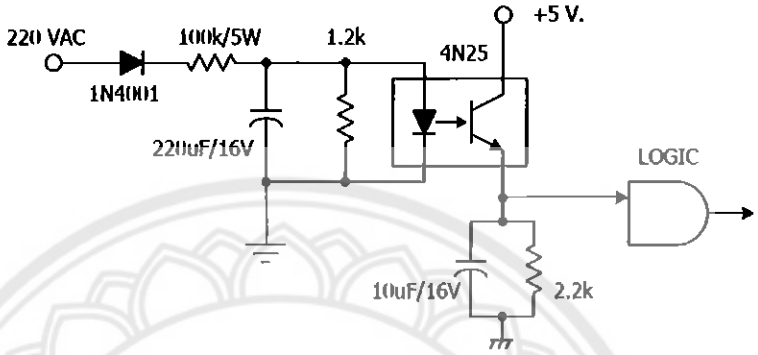
ในโครงงานนี้ใช้ราสเบอร์รี่พายในควบคุมการทำงานของรีเลย์ โดยรับค่าแรงดันไฟฟ้าจากตัวรับรู้ความดันส่วนต่างซึ่งสอดคล้องกับระดับน้ำในถัง มาประมวลผล โดยค่าที่รับมาจากตัวรับรู้ความดันส่วนต่างจะเป็นข้อมูลแอนะล็อก แต่ราสเบอร์รี่พายมีการรับข้อมูลเข้าเพื่อสั่งการเป็นแบบดิจิตอล ดังนั้นจึงต้องมีการแปลงค่าสัญญาณโดยใช้โมดูลตัวแปลงแอนะล็อกเป็นดิจิตอลให้กับราสเบอร์รี่พายเพื่อส่งค่าข้อมูลกลับไปแสดงผลทางหน้าต่างส่วนต่อประสานกับผู้ใช้ รวมถึงส่งคำสั่งไปยังรีเลย์เพื่อใช้ตัดและต่อวงจรโดยขาที่ 11 โดยใช้สัญญาณสั่งการเป็นค่าแรงดัน 3.3 V ในโครงงานนี้ได้นำอุปกรณ์มาต่อเพื่อสั่งการ 3 ส่วน ซึ่งประกอบไปด้วย ราสเบอร์รี่โมดูลตัวแปลงแอนะล็อกเป็นดิจิตอลและโมดูลรีเลย์ดังรูปที่ 3.14

รูปที่ 3.14 การเชื่อมต่ออุปกรณ์สำหรับประมวลผลและควบคุมด้วยราสเบอร์รี่พาย

จากรูปที่ 3.14 เป็นการเชื่อมต่ออุปกรณ์สำหรับประมวลผลและควบคุมผ่านขาสัญญาณอินพุตหรือเอาท์พุตของราสเบอร์รี่พาย โดยในโครงงานนี้ใช้ขาหมายเลข 2 หมายเลข 3 หมายเลข 5 และหมายเลข 9 เชื่อมต่อระหว่างราสเบอร์รี่พายกับโมดูลตัวแปลงแอนะล็อกเป็นดิจิตอล ใช้ขาหมายเลข 2 หมายเลข 4 และหมายเลข 6 เชื่อมต่อระหว่างราสเบอร์รี่พายกับโมดูลรีเลย์ และใช้ขาหมายเลข 2 และหมายเลข 7 ระหว่างราสเบอร์รี่พายกับตัวรับรู้ความดันส่วนต่าง จะเห็นได้ว่าในการใช้งานขาสัญญาณจากราสเบอร์รี่พายมีการใช้ขาร่วมกันนั่นคือขาหมายเลข 2 ซึ่งเป็นขาสำหรับไฟเลี้ยง 5 V ที่จะจ่ายไปยังอุปกรณ์ต่างๆที่เชื่อมต่อเข้ากับราสเบอร์รี่พาย

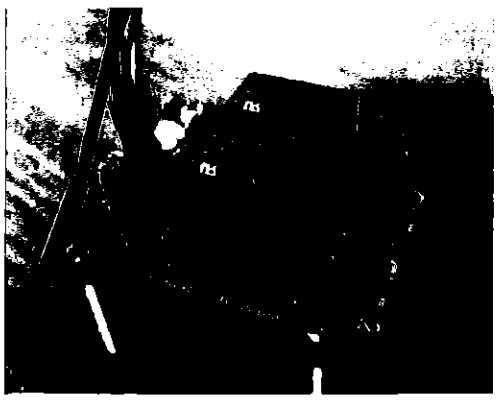
2) แผงวงจรรีเลย์และวงจรขับ

การต่อใช้งานสัญญาณสั่งงานควบคุมการทำ งานของอุปกรณ์ผ่านรีเลย์ โดยมีขนาดแรงดันสัญญาณ Vin และรีเลย์ แรงดันไฟต่ำเท่ากัน คือขนาดเท่ากับ 12 V อาศัยทรานซิสเตอร์ขับรีเลย์โดยตรง ดังรูปที่ 3.15



รูปที่ 3.15 แผนภาพแสดงวงจรขับโลจิกโดยใช้ตัวเชื่อมต่อด้วยแสงกับแรงดันไฟสูง

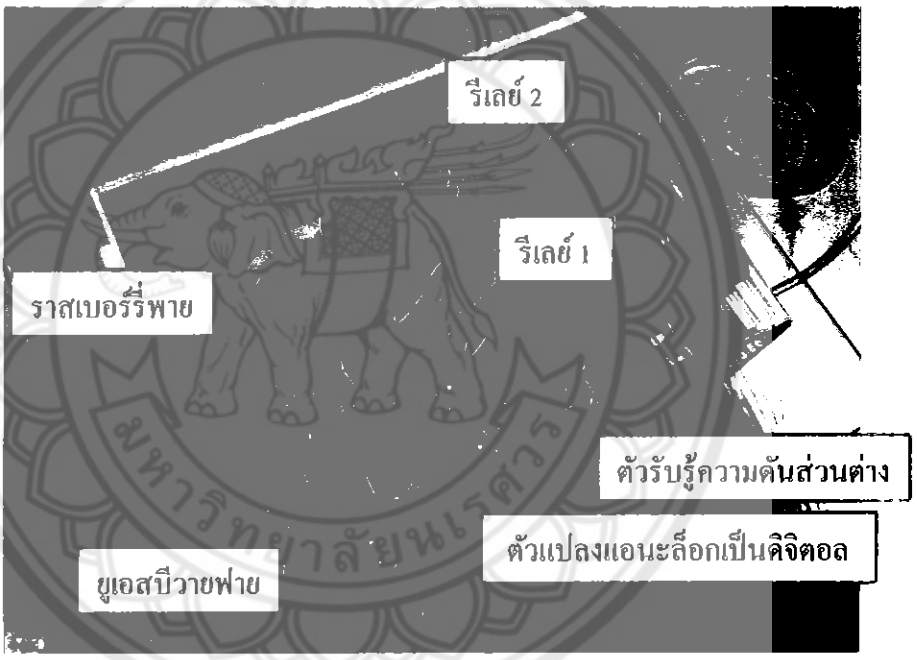
จากรูปที่ 3.15 เป็นวงจรขับโลจิกโดยใช้ตัวเชื่อมต่อด้วยแสง (Opto coupler) กับแรงดันไฟสูง โดยใช้สัญญาณจากแรงดันไฟสูงมาขับออปโต หมายเลข 4N25 มีการเรียงแรงดันเป็นไฟตรงและลดลงเป็นแรงดันไฟต่ำเพื่อขับออปโต จากคุณสมบัติของออปโตจะช่วยแยกส่วนที่เป็นแรงดันไฟสูง ออกจากส่วนที่เป็นแรงดันไฟต่ำ ซึ่งข้อดีของการทำงานแบบนี้คือ เมื่อถ้าหากส่วนใดเกิดการลัดวงจรจะไม่ทำให้เกิดความเสียหายกับวงจรที่เหลือ การต่อวงจรขับด้วยออปโตจะไม่ใช้ขับที่ขารีเลย์โดยตรง เพราะคุณสมบัติของออปโตที่มีความจำกัดของกระแสเอาท์พุต ไม่สามารถขับรีเลย์ที่ต้องการกระแสสูงกว่าได้ ในการทำงานเรากำหนดให้ราสเบอร์รี่พาย ส่งค่าลอจิก 1 มาเพื่อให้ทรานซิสเตอร์นำกระแส และถ้าต้องการให้ปั๊มน้ำหยุดทำงาน เราต้องให้ราสเบอร์รี่พายส่งค่าลอจิก 0 มาทำให้ทรานซิสเตอร์หยุดนำกระแส



รูปที่ 3.16 โมดูลรีเลย์แบบใช้ออปโต

3.3 แบบจำลองของระบบมอเตอร์ปั๊มน้ำ

ในโครงงานนี้ได้มีการออกแบบให้ชุดอุปกรณ์ที่ใช้ควบคุมและสั่งการปั๊มน้ำอยู่ในกล่องสี่เหลี่ยมผืนผ้า ขนาด 22 cm x 32 cm และสูง 15 cm ทั้งนี้เพื่อความสะดวกต่อผู้ใช้ในการเคลื่อนย้ายอุปกรณ์ไปใช้งาน โดยการเริ่มต้นใช้งานนั้นเพียงแค่ต่อเข้ากับไฟบ้าน (220 V 50 Hz) ปั๊มน้ำที่ต้องการควบคุม และต่อสายยางเข้ากับตัวรับรู้ความดันส่วนต่างที่มีอยู่ภายในกล่องจะสามารถเริ่มการควบคุมการเปิดและปิดปั๊มน้ำได้ จากรูปที่ 3.17 จะเห็นว่ามีการต่อรีเลย์เพิ่มอีก 1 ตัว (รีเลย์ 2) เพื่อคั่นการต่อระหว่างไฟบ้านกับราสเบอร์รี่พายโดยตรง ทั้งนี้เพื่อเป็นการป้องกันการเกิดการลัดวงจรจากกระแสไฟย้อนเข้าราสเบอร์รี่พาย เป็นการป้องกันความเสียหายที่อาจเกิดขึ้นกับอุปกรณ์หลักที่ใช้ในการประมวลผลในระบบ



รูปที่ 3.17 กล่องอุปกรณ์สำหรับควบคุมและสั่งการปั๊มน้ำ

1) ตัวรับรู้ความดันส่วนต่าง รุ่น MPX5010DP มีขั้วต่อทางไฟฟ้าจำนวน 6 ขา ซึ่งขาที่ 1 เป็นขาเอาท์พุต ขาที่ 2 ต่อกับกราวด์ และขาที่ 3 ต่อกับไฟเลี้ยงซึ่งต้องการใช้ 5 V ดังรูปที่ 3.18 เรานำสายยางต่อเข้ากับจุกของ MPX5010DP เพื่อใช้ผิวความดันของน้ำมาเปรียบเทียบกับความดันบรรยากาศ โดยป้องกันไม่ให้ตัวรับรู้สัมผัสกับน้ำโดยตรงเพื่อหลีกเลี่ยงอันตรายที่อาจเกิดจากไฟฟ้ารั่วได้

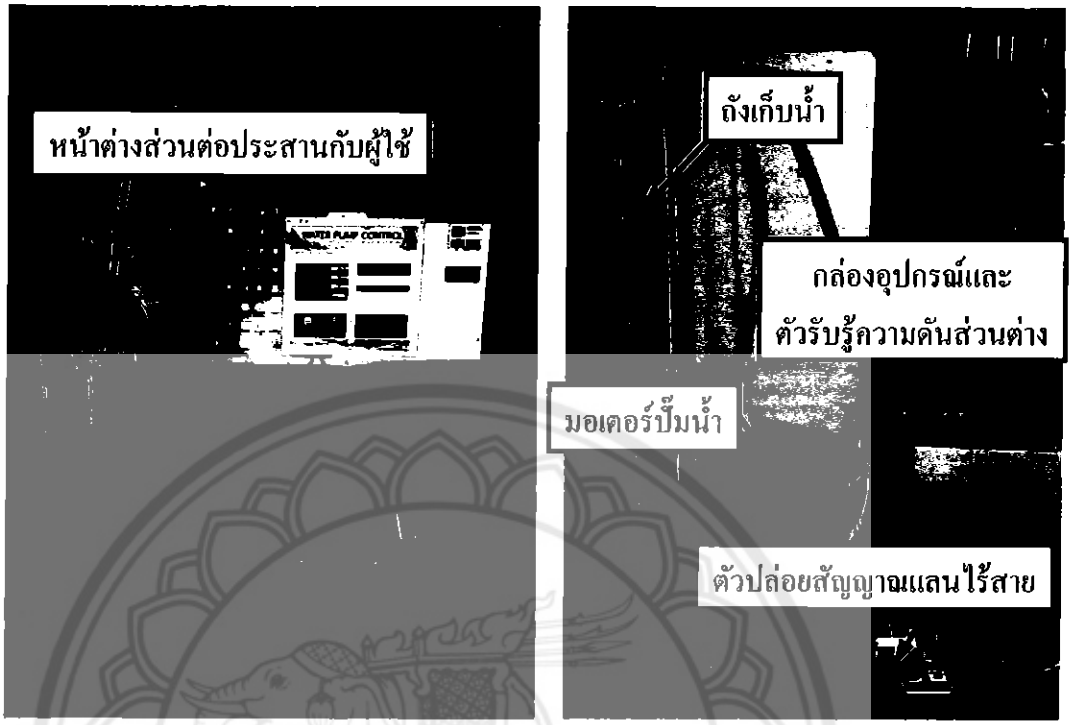
รูปที่ 3.18 ตัวรับรู้ความดันส่วนต่าง รุ่น MPX5010DP

2) มอเตอร์ปั๊มน้ำ ดังรูปที่ 3.19 ซึ่งใช้กับไฟบ้าน (220 V 50 Hz) พิกัด 2,500 W สูบน้ำได้สูงสุด 2 m ซึ่งเพียงพอที่จะใช้งานกับความสูงของถังเก็บน้ำที่สร้างขึ้นในแบบจำลอง

รูปที่ 3.19 มอเตอร์ปั๊มน้ำ รุ่น AP2500 พิกัด 2,500 W

3) ถังเก็บน้ำ (รูปที่ 3.20) ซึ่งถูกออกแบบโดยเน้นความสูงมากกว่าความจุ โดยสร้างมีขนาด ฐาน 14 cm x 14 cm และสูง 110 cm

การเชื่อมต่ออุปกรณ์ในแบบจำลองการควบคุมมอเตอร์ปั๊มน้ำแสดงได้ดังรูปที่ 3.20



(ก) อุปกรณ์ที่สามารถเชื่อมต่อกับอินเทอร์เน็ตได้ เช่น สมาร์ตโฟน คอมพิวเตอร์ เป็นต้น

(ข) กล่องอุปกรณ์ราสเบอร์รี่พายและแบบจำลองระบบปั๊มน้ำ

รูปที่ 3.20 ส่วนประกอบและการเชื่อมต่ออุปกรณ์ในแบบจำลองของระบบควบคุมปั๊มน้ำ

บทที่ 4

ผลการทดสอบและการวิเคราะห์ผล

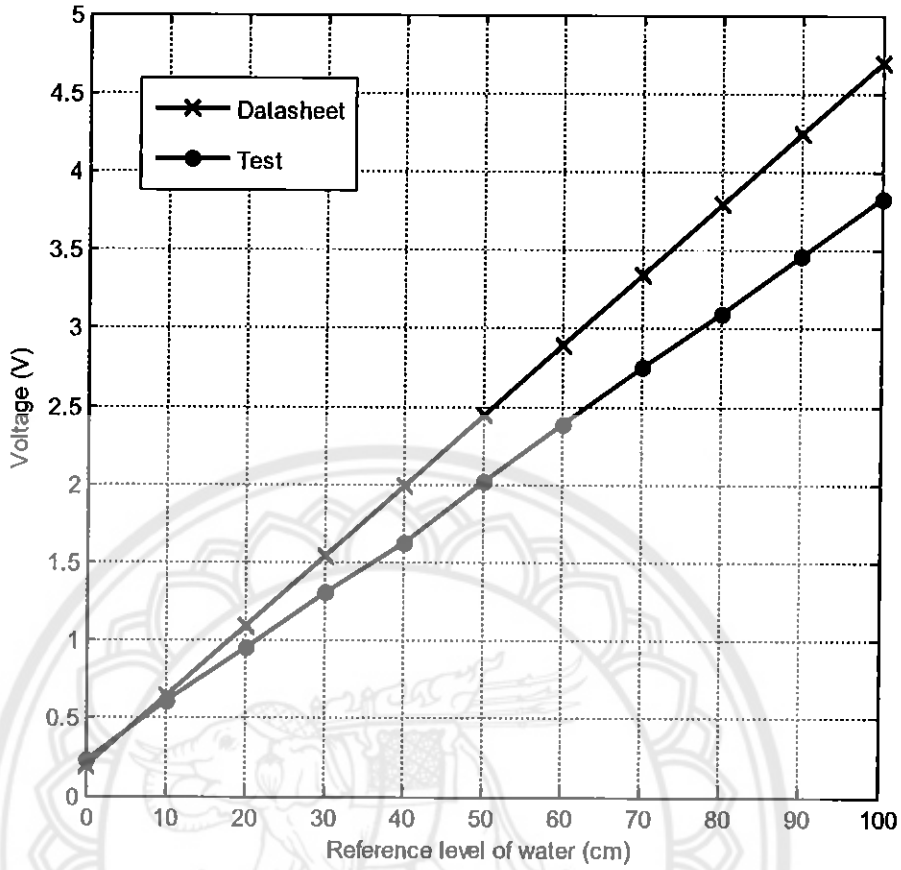
4.1 การทดสอบวัดแรงดันไฟฟ้าจากตัวรับรู้ความดันส่วนต่าง

การทดสอบนี้มีจุดประสงค์เพื่อวัดค่าแรงดันไฟฟ้าที่ได้จากตัวรับรู้ความดันส่วนต่างในแต่ละระดับน้ำทุกๆ 10 cm ตั้งแต่ 0 - 100 cm โดยทำการทดสอบ 5 ครั้งสำหรับแต่ละค่าระดับน้ำแล้วหาค่าเฉลี่ยของแรงดันที่แต่ละระดับน้ำ ผลการทดสอบแสดงดังตารางที่ 4.1 แล้วนำไปเปรียบเทียบกับข้อมูลที่ได้จากผู้ผลิตต่อไป

ตารางที่ 4.1 ค่าแรงดันไฟฟ้าจากตัวรับรู้ความดันส่วนต่าง

ระดับน้ำ (cm)	ค่าแรงดันไฟฟ้าของตัวรับรู้ความดันส่วนต่าง (V)					
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	ค่าเฉลี่ย
0	0.250	0.208	0.251	0.231	0.228	0.234
10	0.615	0.601	0.638	0.634	0.578	0.613
20	0.960	0.949	0.975	0.991	0.905	0.956
30	1.309	1.300	1.321	1.350	1.259	1.308
40	1.472	1.652	1.678	1.707	1.619	1.626
50	2.028	2.012	2.020	2.082	1.970	2.025
60	2.393	2.363	2.383	2.474	2.332	2.389
70	2.752	2.726	2.743	2.856	2.700	2.755
80	3.119	3.085	3.108	3.109	3.064	3.097
90	3.478	3.438	3.448	3.488	3.460	3.462
100	3.852	3.824	3.811	3.856	3.834	3.835

จากตารางที่ 4.1 สามารถนำค่าแรงดันไฟฟ้าที่วัดได้ มาเขียนเป็นกราฟได้ดังรูปที่ 4.1จากการทดสอบจะพบว่าเมื่อระดับน้ำเพิ่มสูงขึ้น ค่าแรงดันไฟฟ้าที่ได้จากตัวรับรู้ความดันส่วนต่างจะมีค่าเพิ่มสูงขึ้นตามกันแบบเชิงเส้น จากรูปที่ 4.1 จะเห็นได้ว่าที่ระดับน้ำสูงขึ้นค่าแรงดันที่วัดได้จากตัวรับรู้มีค่าแตกต่างจากข้อมูลที่ได้จากผู้ผลิตเพิ่มมากขึ้น โดยมีสาเหตุมาจากความผิดพลาดภายในตัวรับรู้ความดันส่วนต่าง และการติดตั้งสายยางที่ต่อมาจากหัวจุกของตัวรับรู้ รวมไปถึงน้ำในถังที่ทดสอบมีการกระเพื่อม



รูปที่ 4.1 ความสัมพันธ์ระหว่างแรงดันไฟฟ้าจากตัวรับรู้ความดันส่วนต่างกับระดับน้ำ

4.2 การทดสอบหาค่าความคลาดเคลื่อนของค่าระดับน้ำที่แสดงผล

การทดสอบนี้มีจุดประสงค์เพื่ออ่านค่าของระดับน้ำที่ปรากฏบนหน้าต่างส่วนต่อประสานกับผู้ใช้ในหน่วยเป็นเปอร์เซ็นต์ (%) และค่าระดับความสูงของน้ำจริงที่อยู่ในถังซึ่งมีการรับค่ามาจากตัวรับรู้ความดันส่วนต่างที่ต่ออยู่ โดยกำหนดให้น้ำในถังมีความสูง ตั้งแต่ 10 - 100 cm โดยทำการทดสอบ 5 ครั้ง ในแต่ละระดับน้ำทุกๆ 10 cm แล้วหาค่าเฉลี่ยของความระดับน้ำที่อ่านได้บนหน้าต่างส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ตที่แต่ละระดับความสูงของน้ำเปรียบเทียบกับค่าระดับความสูงของน้ำจริงที่อยู่ในถัง และหาค่าความคลาดเคลื่อนของระดับน้ำที่เกิดขึ้น แสดงดังตารางที่ 4.2

ตารางที่ 4.2 ค่าระดับน้ำที่แสดงบนส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ต

ระดับน้ำ	ค่าระดับน้ำที่แสดงบนส่วนต่อประสานกับผู้ใช้ (%)						ค่าความคลาดเคลื่อน	
(cm)	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	ค่าเฉลี่ย	(cm)	(%)
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	10.0	10.0	10.0	10.0	10.0	10.0	0.0	0.0
20	20.0	20.0	20.0	20.0	20.0	20.0	0.0	0.0
30	30.0	30.0	30.0	30.0	30.0	30.0	0.0	0.0
40	38.0	40.0	39.0	40.0	40.0	39.4	-0.6	-1.5
50	18.0	48.0	49.0	50.0	50.0	49.0	-1.0	-2.0
60	58.0	57.0	59.0	58.0	60.0	58.4	-1.6	-2.7
70	70.0	70.0	71.0	70.0	70.0	70.2	-0.2	-0.3
80	77.0	78.0	80.0	80.0	80.0	79.0	-1.0	-1.3
90	87.0	88.0	88.0	87.0	90.0	88.0	-2.0	-2.2
100	97.0	100.0	100.0	98.0	97.0	98.6	-1.4	-1.4

จากตารางที่ 4.2 เป็นการอ่านค่าของระดับน้ำแสดงบนหน้าจอของส่วนต่อประสานกับผู้ใช้ โดยมีการเปรียบเทียบจากระดับความสูงของน้ำในถังแล้วนำมาหาค่าเฉลี่ย พบว่าค่าที่ปรากฏบนหน้าต่างส่วนต่อประสานกับผู้ใช้ในหน่วยเปอร์เซ็นต์มีความใกล้เคียงกับค่าระดับน้ำจริงในถังที่วัดเป็นหน่วยเซนติเมตร (cm) ให้ระดับสูงสุดอยู่ที่ 100 cm โดยดูจากค่าความคลาดเคลื่อนที่ผิดไปจากค่าความสูงจริงที่วัดได้ในถัง พบว่าในช่วง 0 - 30 cm ไม่พบค่าความคลาดเคลื่อน แต่ที่ระดับน้ำสูงกว่า 30 cm เริ่มมีค่าความคลาดเคลื่อน โดยค่าความคลาดเคลื่อนมากที่สุดที่พบคือ -1.6 cm คิดเป็น -2.7 % ที่ระดับน้ำในถังสูง 60 cm

4.3 การทดสอบการทำงานในแบบวิธีการควบคุมอัตโนมัติ

การทดสอบนี้มีจุดประสงค์เพื่อทดสอบประสิทธิภาพการต่อไฟและตัดไฟให้กับวงจรที่ต้องการทำงานในแบบควบคุมอัตโนมัติ โดยเป็นการทำงานร่วมกับตัวรับรู้ความดันส่วนต่างที่วัดค่าความสูงของระดับน้ำที่อยู่ในถัง เงื่อนไขการทำงานในแบบควบคุมอัตโนมัติคือ เมื่อระดับน้ำในถังอยู่ต่ำกว่าระดับขีดจำกัดล่างจะมีการสั่งการให้ปั๊มน้ำทำงานเพื่อสูบน้ำเข้าถัง และเมื่อระดับน้ำอยู่ในถังสูงกว่าหรือเท่ากับระดับของขีดจำกัดบนปั๊มน้ำจะหยุดทำงาน และรอให้ระดับน้ำถูกใช้จนถึงระดับค่าขีดจำกัดล่างปั๊มน้ำจะเริ่มทำงานอีกครั้งเป็นการทำงานแบบวนซ้ำไปเรื่อยๆ

จากการทดสอบ การทำงานในแบบวิธีอัตโนมัติสามารถสั่งการให้รีเลย์ตัดต่อวงจรได้ตรงตามเงื่อนไขที่ออกแบบไว้ (ค่าขีดจำกัดบนและค่าขีดจำกัดล่าง) โดยมีความคลาดเคลื่อนจากระดับจริงของน้ำในถังเพียงเล็กน้อย อันเนื่องมาจากการตรวจวัดค่าของตัวรับรู้ความดันส่วนต่าง

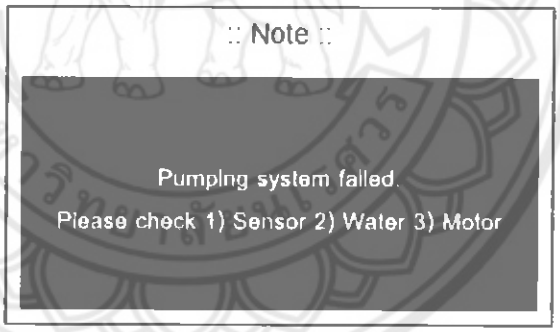
4.4 การทดสอบการทำงานในแบบวิธีการควบคุมด้วยมือ

การทดสอบนี้มีจุดประสงค์เพื่อทดสอบประสิทธิภาพการต่อไฟและตัดไฟให้กับวงจรที่ต้องการทำงานในแบบควบคุมด้วยมือ โดยเป็นการทดสอบความถูกต้องในการสั่งการต่อไฟและตัดไฟผ่านทางหน้าต่างส่วนต่อประสานกับผู้ใช้ โดยมีให้เลือกสั่งการผ่าน 2 ปุ่มคือ ปุ่ม START และปุ่ม STOP พบว่าการสั่งการผ่านทางหน้าจอสามารถเริ่มเดินเครื่องและหยุดเดินเครื่องปั๊มน้ำได้ตามที่ออกแบบไว้

4.5 การทดสอบการทำงานที่มีการตั้งเวลา

ในการทำงานแบบควบคุมอัตโนมัติ ได้มีการออกแบบให้มีการหน่วงเวลาเพื่อรอหลังจากมีคำสั่งให้เริ่มทำงาน เป็นการจับเวลาเพื่อรอให้ระดับน้ำในถังมีการเปลี่ยนแปลง ซึ่งทำหน้าที่เปรียบเสมือนการตรวจสอบการทำงานของอุปกรณ์ทางอ้อม โดยใช้เวลาเป็นเงื่อนไขตรวจสอบการทำงานว่าหากครบเวลาที่กำหนดแล้ว น้ำในถังยังไม่มีการเปลี่ยนแปลง นั่นแสดงถึงการเกิดสภาวะความผิดปกติขึ้นกับมอเตอร์ ตัวรับรู้ รวมไปถึงอุปกรณ์อื่นๆที่เกี่ยวข้องกับการปั๊มน้ำ ในการทำงานแบบควบคุมอัตโนมัตินี้จะถูกเปลี่ยนไปเป็นการทำงานแบบควบคุมด้วยมือ และจะมีข้อความแจ้งเตือนถึงสภาวะความผิดปกตินี้ ดังรูปที่ 4.2

จากการทดสอบการทำงานแบบควบคุมอัตโนมัติร่วมกับการตั้งเวลา โดยการสร้างสถานการณ์จำลองผลการทดสอบกับแบบจำลองแสดงให้เห็นว่า เฉพาะกรณีที่ปั๊มน้ำทำงานอยู่แล้วน้ำหยุดไหลกะทันหัน การเริ่มจับเวลาของระบบมีความล่าช้ากว่าที่ควรจะเป็นประมาณ 2 s เนื่องจากการกระเพื่อมของระดับน้ำภายในถังหลังจากที่หยุดไหล แต่หากเริ่มจับเวลาจากสภาวะที่น้ำหยุดนิ่ง พบว่าเวลาที่จับได้ตรงตามเวลาที่ตั้งไว้ให้กับระบบ



รูปที่ 4.2 ข้อความแจ้งเตือนถึงสภาวะความผิดปกติที่เกิดขึ้นกับระบบ

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

ในโครงงานนี้ได้พัฒนาระบบควบคุมการเปิดและปิดปั๊มน้ำโดยใช้วิธีการควบคุมทางไกลแบบไร้สายผ่านระบบแลนไร้สายร่วมกับราสเบอร์รี่พาย โดยการเขียนชุดคำสั่งโดยใช้ภาษาไพธอนเพื่อควบคุมการรับและส่งข้อมูลรวมทั้งสร้างส่วนต่อประสานกับผู้ใช้บนอินเทอร์เน็ต ในส่วนของการประมวลผลของระบบควบคุมได้นำราสเบอร์รี่พายมาใช้โดยเขียนโปรแกรมด้วยภาษาพีเอชพีเพื่อควบคุมการทำงานของรีเลย์ซึ่งเชื่อมต่อระหว่างแหล่งจ่ายไฟกระแสสลับขนาด 220 V ความถี่ 50 Hz กับปั๊มน้ำ ในโครงงานได้สร้างแบบจำลองโดยใช้ปั๊มน้ำขนาด 2,500 W เพื่อสูบเข้าไปเก็บในถังน้ำที่มีความสูง 110 cm และใช้ตัวรับรู้ความดันส่วนต่างรุ่น MPX5010DP เพื่อตรวจวัดระดับน้ำในถังซึ่งให้ค่าแรงดันเอาท์พุตที่แปรผันตรงกับค่าระดับความสูงของน้ำที่ต้องการวัด โดยมีแบบวิธีการควบคุม 2 วิธีให้ผู้ใช้งานเลือกได้คือแบบวิธีการควบคุมอัตโนมัติ และแบบวิธีการควบคุมด้วยมือ ในแบบวิธีการควบคุมอัตโนมัตินั้นราสเบอร์รี่พายรับค่าแรงดันจากตัวรับรู้ความดันส่วนต่างเข้ามาประมวลผลตลอดเวลา หลังจากที่เชื่อมต่อระบบควบคุมเรียบร้อยแล้วการเปิดและปิดปั๊มน้ำจะเป็นไปอย่างอัตโนมัติโดยขึ้นอยู่กับค่าระดับน้ำในถังซึ่งวัดได้จากตัวรับรู้เทียบกับค่าขีดจำกัดล่างและขีดจำกัดบนซึ่งผู้ใช้สามารถกำหนดค่าเองได้ตามเวลาจริง ทั้งนี้ผู้ใช้สามารถหยุดการทำงานของปั๊มน้ำได้ทุกขณะที่ต้องการ และหลังจากมีการสั่งการหยุดทำงานแล้วแบบวิธีการควบคุมจะถูกเปลี่ยนไปเป็นแบบวิธีการควบคุมด้วยมือเพื่อให้ผู้ควบคุมสามารถเปิดและปิดปั๊มน้ำด้วยตัวเองได้ทุกขณะ นอกจากนี้ในแบบวิธีการควบคุมแบบอัตโนมัติผู้ใช้ยังสามารถตั้งค่าหน่วงเวลาสำหรับตรวจจับการเปลี่ยนแปลงของระดับน้ำในถังเพื่อแจ้งให้ผู้ใช้ได้ตรวจสอบความผิดปกติของอุปกรณ์ในระบบหากระดับน้ำในถังไม่เปลี่ยนแปลงหลังจากสั่งเริ่มการทำงาน

5.2 ปัญหาและแนวทางการแก้ไข

1) ความคลาดเคลื่อนจากการติดตั้งสายยางในขณะทำการวัดและจากการวัดค่าระดับน้ำของตัวรับรู้ส่งผลให้ค่าระดับน้ำที่แสดงบนหน้าต่างส่วนต่อประสานกับผู้ใช้ไม่ตรงกับค่าจริงของระดับน้ำในถัง และการเปรียบเทียบค่ากับขีดจำกัดบนและขีดจำกัดล่างที่ผู้ใช้กำหนดมีการสั่งเปิดและปิดปั๊มน้ำช้ากว่าที่ควรจะเป็น ดังนั้นก่อนเริ่มใช้งานจึงควรติดตั้งสายยางที่ต่อกับจุกตัวรับรู้ความดันส่วนต่างให้ตรง และปรับตั้งค่าการหน่วง

เวลาในการรับและส่งสัญญาณข้อมูลให้เหมาะสม หรือเลือกใช้ตัวรับรู้ความดันส่วนต่างที่มีความแม่นยำในการวัดค่ามากขึ้น

2) การจำลองสถานการณ์ในกรณีปั๊มน้ำเสียหายหรือไม่มีน้ำให้สูบเข้าหลังจากที่ปั๊มน้ำเริ่มทำงานแล้ว พบว่ามีความคลาดเคลื่อนในการเริ่มจับเวลาประมาณ 2 s เนื่องจากถังน้ำที่ใช้ในการทดสอบซึ่งมีรูปทรงที่แคบและสูงส่งผลให้ระดับน้ำมีการเปลี่ยนแปลงรวดเร็วและมีความกระเพื่อม แนวทางการแก้ไขปัญหา คือหากเปลี่ยนขนาดถังที่มีขนาดที่กว้างขึ้นกว่าที่ใช้ในแบบจำลองจะช่วยให้ระดับน้ำในถังเปลี่ยนแปลงช้าลงและมีการกระเพื่อมน้อยลง

3) ความสูงในการติดตั้งใช้งานตัวรับรู้ความดันส่วนต่างอาจส่งผลให้ความคลาดเคลื่อนในการวัดระดับน้ำเพิ่มขึ้นเนื่องจากความดันบรรยากาศมีค่าเปลี่ยนแปลงตามระดับความสูง ดังนั้นการใช้งานตัวรับรู้ความดันส่วนต่างเพื่อวัดระดับน้ำในที่สูงจำเป็นต้องเขียนโปรแกรมเพื่อปรับค่าที่อ่านได้จากตัวรับรู้ให้สอดคล้องกับระดับความสูงที่ติดตั้งเพื่อให้ค่าระดับน้ำที่แสดงบนหน้าจอส่วนต่อประสานกับผู้ใช้มีความแม่นยำ

5.3 แนวทางในการพัฒนาต่อไป

จากหลักการและระบบควบคุมที่พัฒนาขึ้นในโครงงานนี้ เราสามารถเพิ่มจำนวนปั๊มน้ำที่ต้องการควบคุม รวมถึงการนำไปประยุกต์ใช้ควบคุมการทำงานของอุปกรณ์ไฟฟ้าอื่นๆที่ อยู่ไกลจากผู้ควบคุมห้องควบคุม ได้แก่ การเปิดและปิดอุปกรณ์ไฟฟ้าที่มีใช้งานทั่วไปในครัวเรือน รวมไปถึงอุปกรณ์เครื่องจักรกลไฟฟ้าภายในโรงงาน เพื่อเป็นการช่วยอำนวยความสะดวกแก่ผู้ใช้งาน นอกจากนี้ยังสามารถนำไปพัฒนาในเรื่องของการออนไลน์เพื่อสั่งการและดูสถานะการทำงานผ่านหน้าเว็บ เพื่อความสะดวกโดยไม่มีข้อจำกัดทางด้านโครงข่ายระแบบแลนไร้สาย โดยนำไฟล์พีเอชพีไปฝากไว้กับผู้ให้บริการขององค์กรใดๆ ซึ่งจะมีรูปแบบการทำงานแบบเปิดไว้ตลอดเวลา

เอกสารอ้างอิง

[1] Informal soft. "ทำความรู้จักกับราสเบอร์รี่พาย". สืบค้นเมื่อ 8 กันยายน 2557, จาก http://www.informalsoft.com.

[2] บริษัท อีทีที จำกัด. "เริ่มต้นการใช้งานบอร์ด Raspberry Pi". สืบค้นเมื่อ 8 กันยายน 2557, จาก http://www.etteam.com/.

[3] บริษัท อะเมซอน จำกัด. "ตัวรับส่งแบบยูเอสบีของระบบแลนไร้สาย". สืบค้นเมื่อ 11 กันยายน 2557, จาก http://www.amazon.com/.

[4] สรกฤช สิริปรีดากุล และพิมลลักษณ์ จิรกุลกนก. (2551). "การพัฒนาระบบควบคุมไฟฟ้าโดยใช้ Zigbee ประเภทหัวข้อพิเศษ Mobile application". ในโครงการวิจัย พัฒนา และวิศวกรรม (หน้า 5-7). กรุงเทพฯ: ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยธรรมศาสตร์.

[5] กิตติศักดิ์ ยังประคับ. "เทคโนโลยี Wi-Fi". สืบค้นเมื่อ 11 กันยายน 2557, จาก http://www.vcharkarn.com.

[6] บริษัท PHP Group. "ความรู้เบื้องต้นเกี่ยวกับพีเอชพี". สืบค้นเมื่อ 18 กันยายน 2557, จาก http://php.net/.

[7] Ken Lambert (2014). "Fundamentals of Python: Data Structures" สืบค้นเมื่อ 18 กันยายน 2557, จาก https://www.python.org/.

[8] ไทยกู๊ดวิวดอทคอม. "การใช้งานภาษาจาวาสคริปต์". สืบค้นเมื่อ 18 กันยายน 2557, จาก http://www.thaigoodview.com/.

[9] บริษัท อีลีเมนต์ โฟร์ทีน จำกัด. "เอกสารประกอบการใช้งานตัวรับรู้ความดันส่วนต่าง" สืบค้นเมื่อ 25 กันยายน 2557, จาก http://th.element14.com/.

[10] มณีนุช สมานหมู่ และปิยะ นากสงค์. "มือใหม่ Photoshop Dreamweaver CS6 ออกแบบสร้างเว็บไซต์" พิมพ์ครั้งที่ 2. กรุงเทพฯ: ซีเอ็ดยูเคชั่น, สนพ., 2556.

ภาคผนวก ก

รหัสต้นฉบับเพื่อสร้างชุดคำสั่งสำหรับควบคุมการทำงานของระบบ

```
//เรียกใช้ไลบรารี่อุปกรณ์และฟังก์ชันต่างๆ//
<?php
        session_start();
        if($_SESSION['UserID'] == "")
        {               echo "Please Login!";
                        exit();                                         }
        if($_SESSION['Status'] != "ADMIN")
        {               echo "This page for Admin only!";
                        exit();                                         }
mysql_connect("localhost","root","root");
mysql_select_db("mydatabase");
$strSQL = "SELECT * FROM member WHERE UserID = '".$_SESSION['UserID']."' ";
$objQuery = mysql_query($strSQL);
$objResult = mysql_fetch_array($objQuery); ?>

 //การเริ่มการเขียนคำสั่งหน้าเว็บ//
<html>
<head>
<title>Nattawadee & Jirayus Project 2557</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<style type="text/css">
        body {          background-image: url(pic/BG3.jpg);             }
        .font_note {    font-size: small;
                        color: #0046A6;                                 }
        .fontnote {     font-size: xx-small;
                        color: #0046A6;                                 }
        .note    {      font-size: x-small;
                        color: #0046A6;                                 }
        .head    {      font-size: medium;                              }
        .headd {        font-size: large;                               }
        .upperbox {     height: 25px;
                        width: 50px;                                    }
```

```
.table-front {     color: #FFFFFF;
                   background-color: #0A8AC8;                    }
.a { font-family: Gotham, Helvetica Neue, Helvetica, Arial, sans-serif;    }
```

//ส่วนของการเขียนคำสั่งให้มีการตั้งค่าต่างๆบนหน้าเว็บ//

```
</style>
</head>
<body>
<table width="800" border="1" align="center" cellspacing="0" class="a">
<tbody>
<tr bgcolor="#FFFFFF">
<td colspan="2"><img src="pic/HEADER.jpg" width="800" height="134" alt=""/></td>
</tr>
<tr bgcolor="#FFFFFF">
<td height="50" colspan="2"><table width="780" height="40" border="0"
align="center">
<tbody>
<tr align="left">
<th width="340" height="36" valign="middle" scope="col"> Hello! <?php echo
$objResult["Name"];?></th>
<th width="230" valign="middle" scope="col">ID:<?php echo
$objResult["Username"];?></th>
<th width="137" scope="col"><a href="edit_profile.php">Edit Profile</a></th>
<th width="55" scope="col"><a href="logout.php">Logout</a></th>
</tr>
</tbody>
</table></td>
</tr>
<tr bgcolor="#FFFFFF">
<td width="400" align="center" valign="top"><table width="100" border="0">
<tbody>
<tr align="center">
```

```
<td valign="top"><p class="headd">:: Setting ::</p>
<table width="350" border="1" align="center" cellspacing="0" class="table-front">
<tbody>
<tr>
<td width="225" height="58" align="center" valign="middle"><table width="200"
border="0" align="center" class="table-front">
<tbody>
<tr>
<td width="89" height="50" align="center" valign="middle">Upper limit :</td>
<td width="46" height="50" align="center" valign="middle" id="txtMax"> </td>
<td width="51" align="center" valign="middle">%</td>
</tr>
</tbody>
</table></td>
<td width="50" align="center" valign="middle"><input name="maxlv" type="number"
class="upperbox" id="maxlv" max="100" min="1" onKeyPress="return isNumeric(event)"
maxlength="3" oninput="maxLengthCheck(this)"></td>
<td width="61" align="center" valign="middle"><input type="button"
name="btnupdateUpper" id="btnupdateUpper" value="Update"></td>
</tr>
<tr>
<td height="58" align="center" valign="middle"><table width="200" border="0"
align="center" class="table-front">
<tbody>
<tr>
<td width="90" height="50" align="center" valign="middle">Lower limit :</td>
<td width="43" height="50" align="center" valign="middle" id="txtMin"> </td>
<td width="53" align="center" valign="middle">%</td>
</tr>
</tbody>
</table></td>
```

```
<td align="center" valign="middle"><input name="minlv" type="number"
class="upperbox" id="minlv" max="99" min="0" onKeyPress="return isNumeric(event)"
maxlength="2" oninput="maxLengthCheck(this)"></td>
<td align="center" valign="middle"><input type="button" name="btnupdateLower"
id="btnupdateLower" value="Update"></td>
</tr>
<tr>
<td height="58" align="center" valign="middle"><table width="200" border="0"
align="center" class="table-front">
<tbody>
<tr>
<td width="90" height="50" align="center" valign="middle">Time delay :</td>
<td width="42" height="50" align="center" valign="middle" id="txttimer"> </td>
<td width="54" align="center" valign="middle">Second</td>
</tr>
</tbody>
</table></td>
<td align="center" valign="middle"><input name="mSettimer" type="number"
class="upperbox" id="mSettimer" max="3000" min="5" onKeyPress="return isNumeric(event)"
maxlength="4" oninput="maxLengthCheck2(this)"></td>
<td align="center" valign="middle"><input type="button" align="center" name="button"
id="btnsettimer" value="Update"></td>
</tr>
<tr>
<td height="58" align="center" valign="middle"><table width="216" height="51"
border="0" align="center" class="table-front">
<tbody>
<tr>
<td width="125" height="47" align="center">Operating mode :</td>
<td width="81" align="center" valign="middle" id="txtMode"> </td>
</tr>
</tbody>
```

```
</table></td>
<td colspan="2" align="center" valign="middle"><select name="select" id="myselect"
onChange="selectmode()">
<option>Automatic</option>
<option selected="selected">Manual</option>
</select></td>
</tr>
</tbody>
</table>
<table width="350" border="0">
<tbody>
<tr>
<td><p class="note">Note : Set the delay time for cutting the power supplied to motor, if
the water doesn't flow or motor isn't running. Time from 5 - 3000 seconds.</p></td>
</tr>
</tbody>
</table></td>
</tr>
<tr>
<td height="217" align="center" valign="bottom"><p class="headd">:: Water pump
control ::</p>
<table width="350" height="159" border="1" align="center" cellspacing="0"
class="table-front">
<tbody>
<tr>
<th width="158" height="136" align="center" valign="middle" scope="col"><img
src="pic/start.png" alt="" width="90" height="90" id="button_on" onMouseDown="changeOn()"
onMouseUp="invertOn()" onMouseOut="invertOn()"
onMouseOver="JavaScript:this.style.cursor='hand'"/>
<p>START</p></th>
<th width="158"height="136" align="center" valign="middle" scope="col"> <img
src="pic/stop.png" alt="" width="9"height="90" id="button_off"
```

```
onMouseDown="changeOff()"onMouseUp="invertOff()"onMouseOut="invertOff()"
onMouseOver="JavaScript:this.style.cursor='hand'"/>
        <p>STOP</p></th>
        </tr>
        </tbody>
        </table></td>
        </tr>
        </tbody>
        </table></td>
        <td width="396" align="center" valign="top"><table width="100" height="536"
border="0">
        <tbody>
        <tr align="center">
        <td height="325" valign="top"><p class="headd">:: Display ::</p>
        <table width="350" height="73" border="1" cellspacing="0" class="table-front">
        <tbody>
        <tr>
        <td width="228"><table width="340" height="69" border="0" align="center"
class="table-front">
        <tbody>
        <tr align="center" valign="middle" class="headd">
        <td width="164" height="50">Water level :</td>
        <td width="73" height="50" id="txtlv"> </td>
        <td width="89">%</td>
        </tr>
        </tbody>
        </table></td>
        </tr>
        </tbody>
        </table>
        <table width="350" border="0">
        <tbody>
```

```
<tr>
<td width="262" class="note">Note : Set the water level 100% for water tank in the beginning.</td>
</tr>
</tbody>
</table>
<p>
<input type="button" name="st100" id="btnst100" value="Set 100%">
</p>
<table width="350" height="41" border="1" cellspacing="0" class="table-front">
<tbody>
<tr>
<td width="416"><table width="339" height="37" border="0" align="center" class="table-front">
<tbody>
<tr align="center" valign="middle">
<td width="118">Motor status :</td>
<td id="tstatus" width="211"></td>
</tr>
</tbody>
</table></td>
</tr>
</tbody>
</table>
<p>
<input type="button" name="Reset Alarm" id="btnRstAlarm" value="Reset Alarm">
</p></td>
</tr>
<tr>
<td height="205" align="center" valign="top"><p class="headd">:: Note ::</p>
<table width="350" height="157" border="1" cellspacing="0" class="table-front">
<tbody>
```

```
<tr>
<td height="151"><table width="339" height="129" border="0" align="center"
class="table-front">
<tbody>
<tr>
<td width="350" height="25" align="center" valign="bottom"
id="txtWarning"> </td>
</tr>
<tr>
<td height="25" align="center" valign="top" id="txtWarning2" > </td>
</tr>
</tbody>
</table></td>
</tr>
</tbody>
</table></td>
</tr>
</tbody>
</table></td>
</tr>
<tr bgcolor="#FFFFFF">
<td colspan="2"><img src="pic/button.jpg" width="800" height="70" alt=""/></td>
</tr>
</tbody>
</table>
<script language="javascript" type="text/javascript">
        var button_on = document.getElementById("button_on");
        var button_off = document.getElementById("button_off");
        var tstatus = document.getElementById("tstatus");
        var btnupdate = document.getElementById("btnupdate");
        var btnsettimer = document.getElementById("btnsettimer");
        var mymax = document.getElementById("maxlv");
```

```
var mymin = document.getElementById("minlv");
var textmin = document.getElementById("txtMin");
var textmax = document.getElementById("txtMax");
var textlv = document.getElementById("txtlv");
var textalert = document.getElementById("txtWarning");
var textalert2 = document.getElementById("txtWarning2");
var texttimer = document.getElementById("txttimer");
var msettimer = document.getElementById("mSettimer");
var btnrstalarm = document.getElementById("btnRstAlarm");
var btnset100 = document.getElementById("btnst100");
var textmode = document.getElementById("txtMode");
var myselect = document.getElementById("myselect");
var btnupdateUpper = document.getElementById("btnupdateUpper");
var btnupdateLower = document.getElementById("btnupdateLower");
setInterval(function () {mytimer()}, 500);
function mytimer(){
        checkmypin();
        checkmin();
        checkmax();
        checklv();
        checkalert();
        checktimer();
        checkmode();                                    }
function changeOn(){
button_on.src="pic/start2.png";                         }
function invertOn(){
button_on.src="pic/start.png";                          }
function changeOff(){
button_off.src="pic/stop2.png";                         }
function invertOff()    {button_off.src="pic/stop.png";         }
function selectmode()   {if(button_on.alt=="auto"){
var myanswer = confirm("ต้องการเปลี่ยนโหมดเป็นโหมด Manual ใช่หรือไม่");
```

```
            if(myanswer == true)
                {manualmode();                                              }}
            else if(button_on.alt==="manual"){
        var myanswer = confirm("ต้องการเปลี่ยนโหมดเป็นโหมด Automatic ใช่
หรือไม่");
            if(myanswer == true)
                {automode();                                                }}}
            function checkmode(){
        var request = new XMLHttpRequest();
        request.open( "GET" , "checkmode.php");
        request.send(null);
        request.onreadystatechange = function () {
            if (request.readyState == 4 && request.status == 200) {
            if (parseInt(request.responseText)){
            textmode.textContent = "Automatic";
            button_off.alt = "auto";
            button_on.alt = "auto";
        myselect.selectedIndex = 0;                                         }
            else{    textmode.textContent = "Manual";
            button_off.alt = "manual";
            button_on.alt = "manual";
        myselect.selectedIndex = 1;                                         }}
            else if (request.readyState == 4 && request.status == 500) {
            alert ("server error");
            return ("fail");                                                }
            else { return ("fail");                                         }}}
        function automode(){
            if (textalert.textContent !== "Pumping system failed."){
        var request = new XMLHttpRequest();
        request.open( "GET" ,"automode.php");
        request.send(null);
        request.onreadystatechange = function () {
```

```
        if (request.readyState == 4 && request.status == 200) {          }
        else if (request.readyState == 4 && request.status == 500){
        alert ("server error");
        return ("fail");                                                 }
        else { return ("fail");                                          }}}
        else{alert("กรุณา Reset Alarm ก่อนดำเนินการใดๆ");                }}
function manualmode(){
        if (textalert.textContent !== "Pumping system failed."){
var request = new XMLHttpRequest();
request.open( "GET" ,"manualmode.php");
request.send(null);
request.onreadystatechange = function () {
        if (request.readyState == 4 && request.status == 200) {          }
        else if (request.readyState == 4 && request.status == 500){
        alert ("server error");
        return ("fail");                                                 }
        else { return ("fail");                                          }}}
        else{alert("กรุณา Reset Alarm ก่อนดำเนินการใดๆ");                }}
        btnupdateUpper.addEventListener("click", function() {
        if (textalert.textContent !== "Pumping system failed.") {
        if(mymax.value.length>0){
var max = parseInt(mymax.value);
var min = parseInt(textmin.textContent);
        if(max > min){updatemaxmin (max,min);
        alert("Upper setting level = " + mymax.value);
        alert("ตั้งค่า Upper setting เสร็จเรียบร้อย");
        mymax.value = "";                                                }
        else{alert("กรุณาป้อนค่า Upper setting level ให้ถูกต้อง");       }}
        else{alert("กรุณาป้อนค่า Upper setting level");                  }}
        else{alert("กรุณา Reset Alarm ก่อนดำเนินการใดๆ");                }});
btnupdateLower.addEventListener("click", function() {
        if (textalert.textContent !== "Pumping system failed."){
```

```
            if(mymin.value.length>0){
var max = parseInt(textmax.textContent);
var min = parseInt(mymin.value);
            if(max > min){
            updatemaxmin (max,min);
            alert("Lower setting level = " + mymin.value);
            alert("ตั้งค่า Lower setting เสร็จเรียบร้อย");
            mymin.value = "";                                    }
            else{alert("กรุณาป้อนค่า Lower setting level ให้ถูกต้อง");    }}
            else{alert("กรุณาป้อนค่า Lower setting level");          }}
            else{alert("กรุณา Reset Alarm ก่อนดำเนินการใดๆ");        }});
btnsettimer.addEventListener("click", function() {
            if (textalert.textContent !== "Pumping system failed."){
            if (msettimer.value.length >= 1){
            if(parseInt(msettimer.value) >= 5){
var mytimer = parseInt(msettimer.value);
updatetimer (mytimer);
            alert("ตั้งค่า Time delay เป็นเวลา"+mytimer+" วินาที เสร็จสมบูรณ์");
            msettimer.value ="";                                 }
            else{alert("ไม่สามารถตั้งค่า Time delay ได้ เนื่องจากมีค่าน้อยกว่า 5");
            msettimer.value ="";                                 }}}
            else{alert("กรุณา Reset Alarm ก่อนดำเนินการใดๆ");        }});
function updatetimer (mtime){
var request = new XMLHttpRequest();
request.open( "GET" , "updatetimer.php?mtime=" + mtime);
request.send(null);
request.onreadystatechange = function () {
            if (request.readyState == 4 && request.status == 200) {  }
            else if (request.readyState == 4 && request.status == 500) {
            alert ("server error");
            return ("fail");                                     }
            else { return ("fail");                              }}}
```

```
function checktimer(){
var request = new XMLHttpRequest();
request.open( "GET" , "checktimer.php");
request.send(null);
request.onreadystatechange = function () {
        if (request.readyState == 4 && request.status == 200) {
        texttimer.textContent = parseInt(request.responseText);                }
        else if (request.readyState == 4 && request.status == 500) {
        alert ("server error");
        return ("fail");                                                       }
        else { return ("fail");                                                }}}
function checkalert(){
var request = new XMLHttpRequest();
request.open( "GET" , "checkalert.php");
request.send(null);
request.onreadystatechange = function () {
        if (request.readyState == 4 && request.status == 200) {
var mflag = parseInt(request.responseText);
        if (mflag){       textalert.textContent = "Pumping system failed.";
        textalert2.textContent = "Please check (1)Sensor (2)Water (3)Motor";
var new_status  change_pin ( 0, 0);
        if (new_status !== "fail") {
        return 0;                                                              }}
        else{    textalert.textContent = "";
                 textalert2.textContent = "";                                  }}
        else if (request.readyState == 4 && request.status == 500) {
        alert ("server error");
        return ("fail");                                                       }
        else { return ("fail");                                                }}}

function checklv(){
        var request = new XMLHttpRequest();
```

```
request.open( "GET" , "checklv.php");
request.send(null);
request.onreadystatechange = function () {
        if (request.readyState == 4 && request.status == 200) {
        textlv.textContent = parseInt(request.responseText);        }
        else if (request.readyState == 4 && request.status == 500) {
        alert ("server error");
        return ("fail");                                            }
        else { return ("fail");                                     }}}
function checkmin(){
var request = new XMLHttpRequest();
request.open( "GET" , "checkmin.php");
request.send(null);
request.onreadystatechange = function () {
        if (request.readyState == 4 && request.status == 200) {
        textmin.textContent = parseInt(request.responseText);       }
        else if (request.readyState == 4 && request.status == 500) {
        alert ("server error");
        return ("fail");                                            }
        else { return ("fail");                                     }}}
function checkmax(){
var request = new XMLHttpRequest();
request.open( "GET" , "checkmax.php");
request.send(null);
request.onreadystatechange = function () {
        if (request.readyState == 4 && request.status == 200) {
        textmax.textContent = parseInt(request.responseText);       }
        else if (request.readyState == 4 && request.status == 500) {
        alert ("server error");
        return ("fail");                                            }
        else { return ("fail");                                     }}}
function checkmypin (){
```

```
var request = new XMLHttpRequest();
request.open( "GET" , "checkmyout.php");
request.send(null);
request.onreadystatechange = function () {
        if (request.readyState == 4 && request.status == 200) {
        if (parseInt(request.responseText)){
        tstatus.textContent = "Water Pump On!!!";                    }
        else{tstatus.textContent = "Water Pump Off";                 }}
        else if (request.readyState == 4 && request.status == 500) {
        alert ("server error");
        return ("fail");                                             }
        else { return ("fail");                                      }}}
function maxLengthCheck(object) {
        if (object.value.length >= object.maxLength){
        object.value = object.value.slice(0, object.maxLength)
        if(object.value>100){
        object.value = 100                                           }}}
function maxLengthCheck2(object) {
        if (object.value.length >= object.maxLength){
        object.value = object.value.slice(0, object.maxLength)
        if(object.value>3000){
        object.value = 3000                                          }}}
        function isNumeric (evt) {
var theEvent = evt || window.event;
var key = theEvent.keyCode || theEvent.which;
key = String.fromCharCode (key);
var regex = /[0-9]/;
        if ( !regex.test(key) ) {
        theEvent.returnValue = false;
        if(theEvent.preventDefault) theEvent.preventDefault();       }}
function rstalarm(){
var request = new XMLHttpRequest();
```

```
request.open( "GET" ,"rstalarm.php");
request.send(null);
request.onreadystatechange = function () {
if (request.readyState == 4 && request.status == 200) {          }
else if (request.readyState == 4 && request.status == 500) {
alert ("server error");
return ("fail");                                                 }
else { return ("fail");                                          }}}
function set100(){
var request = new XMLHttpRequest();
request.open( "GET" ,"set100.php");
request.send(null);
request.onreadystatechange = function () {
if (request.readyState == 4 && request.status == 200) {          }
        else if (request.readyState == 4 && request.status == 500) {
        alert ("server error");
        return ("fail");
}
        else { return ("fail");
}}}
function updaterun(){
var request = new XMLHttpRequest();
request.open( "GET" ,"updaterun.php");
request.send(null);
request.onreadystatechange = function () {
        if (request.readyState == 4 && request.status == 200) {
}
        else if (request.readyState == 4 && request.status == 500) {
        alert ("server error");
        return ("fail");
}
```

```
                else { return ("fail");
        }}}
                function updatemaxmin (mxlv, mnlv){
                var request = new XMLHttpRequest();
                request.open( "GET" ,
"updatemaxmin.php?mxlv="+mxlv+"&mnlv="+mnlv );
                request.send(null);
                request.onreadystatechange = function () {
                        if (request.readyState == 4 && request.status == 200) {
        }
                        else if (request.readyState == 4 && request.status == 500) {
                        alert ("server error");
                        return ("fail");
        }
                        else { return ("fail");
        }}}
                function change_pin (pin, status) {
                var request = new XMLHttpRequest();
                request.open( "GET" , "gpio.php?pin=" + pin + "&status=" + status );
                request.send(null);
                request.onreadystatechange = function () {
                        if (request.readyState == 4 && request.status == 200) {
                        return (parseInt(request.responseText));
        }
                        else if (request.readyState == 4 && request.status == 500) {
                        alert ("server error");
                        return ("fail");
        }
                        else { return ("fail");
        }}}
                        button_on.addEventListener("click", function () {
                        if(textalert.textContent !== "Pumping system failed."){
```

```
                if ( button_on.alt === "manual") {
            var new_status = change_pin ( 0, 1);
                if (new_status !== "fail") {
                return 0;
}}}
                else{alert("กรุณา Reset Alarm ก่อนดำเนินการใดๆ");
            }});
                button_off.addEventListener("click", function () {
                if(textalert.textContent !== "Pumping system failed."){
                if ( button_off.alt === "manual") {
            var new_status = change_pin ( 0, 0);
                if (new_status !== "fail") {
                return 0;
}}
                else if (button_off.alt === "auto"){
            var myanswer = confirm("ต้องการหยุดระบบ Automatic ใช่หรือไม่");
                if(myanswer == true){
                manualmode();
                var new_status = change_pin ( 0, 0);
                if (new_status !== "fail") {
                return 0;
}}}
                else{alert("กรุณา Reset Alarm ก่อนดำเนินการใดๆ");
            }});
                btnrstalarm.addEventListener("click", function(){
                rstalarm();
                alert("รีเซต Alarm เรียบร้อย");
});
                btnset100.addEventListener("click", function(){
                if (textalert.textContent !== "Pumping system failed.") {
                if (parseInt(textlv.textContent)){
```

```
                    set100();
        }
                    else{alert("ไม่สามารถตั้งค่าได้เนื่องจากระดับมีค่าเป็น 0");
        }}
                    else{alert("กรุณา Reset Alarm ก่อนดำเนินการใดๆ");
        }});
                    btnupdate.addEventListener("click", function() {
                    if (textalert.textContent !== "Pumping system failed.") {
                    if(mymax.value.length>0 && mymin.value.length>0){
                var max = parseInt(mymax.value);
                var min = parseInt(mymin.value);
                    if(max > min){
                    updatemaxmin (max,min);
                    alert("Max Level="+mymax.value+" และMin
Level="+mymin.value);
                    alert("ตั้งค่า Max Level และ Min Level เสร็จเรียบร้อย");
                    mymax.value = "";
                    mymin.value = "";
        }
                    else{
                    alert("กรุณาป้อนค่า Min Level หรือ Max Level ให้ถูกต้อง");
        }}
                    else{
                    alert("กรุณาป้อนค่า Min Level และ Max Level ให้ครบถ้วน");
        }}
                    else{alert("กรุณา Reset Alarm ก่อนดำเนินการใดๆ");
        }});
            </script>
            </body>
            </html>
```

ภาคผนวก ข

รายละเอียดข้อมูลของโมดูลตัวแปลงแอนะล็อกเป็นดิจิตอล รุ่น PCF8591

8-bit A/D and D/A converter PCF8591

1 FEATURES

- Single power supply
- Operating supply voltage 2.5 V to 6 V
- Low standby current
- Serial input/output via I²C-bus
- Address by 3 hardware address pins
- Sampling rate given by I²C-bus speed
- 4 analog inputs programmable as single-ended or differential inputs
- Auto-incremented channel selection
- Analog voltage range from V_{SS} to V_{DD}
- On-chip track and hold circuit
- 8-bit successive approximation A/D conversion
- Multiplying DAC with one analog output.

2 APPLICATIONS

- Closed loop control systems
- Low power converter for remote data acquisition
- Battery operated equipment
- Acquisition of analog values in automotive, audio and TV applications.

3 GENERAL DESCRIPTION

The PCF8591 is a single-chip, single-supply low power 8-bit CMOS data acquisition device with four analog inputs, one analog output and a serial I²C-bus interface. Three address pins A0, A1 and A2 are used for programming the hardware address, allowing the use of up to eight devices connected to the I²C-bus without additional hardware. Address, control and data to and from the device are transferred serially via the two-line bidirectional I²C-bus.

The functions of the device include analog input multiplexing, on-chip track and hold function, 8-bit analog-to-digital conversion and an 8-bit digital-to-analog conversion. The maximum conversion rate is given by the maximum speed of the I²C-bus.

4 ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
PCF8591P	DIP16	plastic dual in-line package; 16 leads (300 mil)	SOT38-4
PCF8591T	SO16	plastic small outline package; 16 leads; body width 7.5 mm	SOT162-1

5 BLOCK DIAGRAM

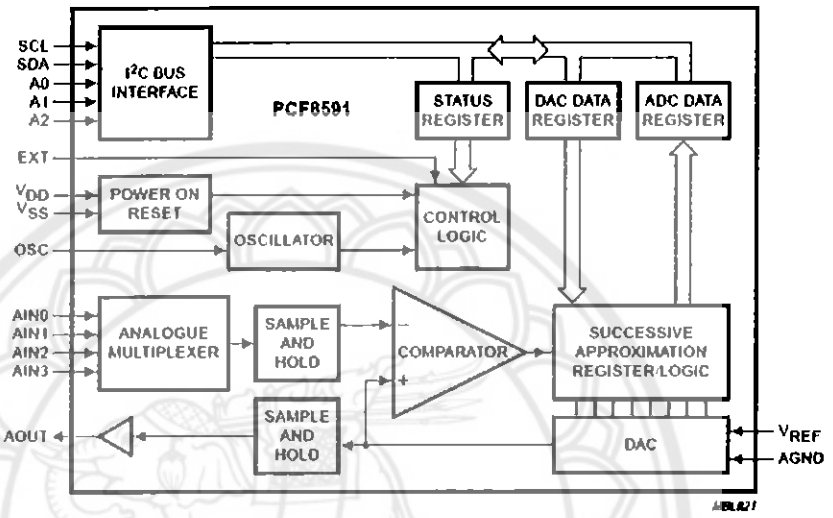


Fig.1 Block diagram.

6 PINNING

SYMBOL	PIN	DESCRIPTION
AIN0	1	analog inputs (A/D converter)
AIN1	2	
AIN2	3	
AIN3	4	
A0	5	hardware address
A1	6	
A2	7	
V_{SS}	8	negative supply voltage
SDA	9	I²C-bus data input/output
SCL	10	I²C-bus clock input
OSC	11	oscillator input/output
EXT	12	external/internal switch for oscillator input
AGND	13	analog ground
V_{REF}	14	voltage reference input
AOUT	15	analog output (D/A converter)
V_{DD}	16	positive supply voltage

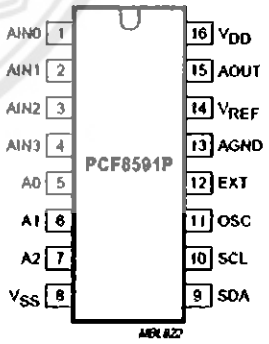


Fig.2 Pinning diagram (DIP16).

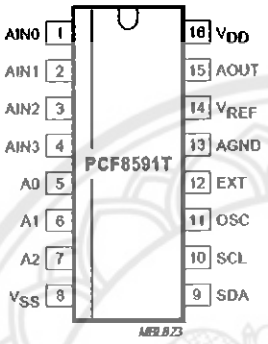


Fig 3 Pinning diagram (SO16).

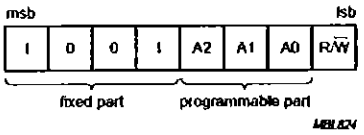


Fig.4 Address byte.

7 FUNCTIONAL DESCRIPTION

7.1 Addressing

Each PCF8591 device in an I²C-bus system is activated by sending a valid address to the device. The address consists of a fixed part and a programmable part. The programmable part must be set according to the address pins A0, A1 and A2. The address always has to be sent as the first byte after the start condition in the I²C-bus protocol. The last bit of the address byte is the read/write-bit which sets the direction of the following data transfer (see Figs 4, 16 and 17).

7.2 Control byte

The second byte sent to a PCF8591 device will be stored in its control register and is required to control the device function. The upper nibble of the control register is used for enabling the analog output, and for programming the analog inputs as single-ended or differential inputs. The lower nibble selects one of the analog input channels defined by the upper nibble (see Fig.5). If the auto-increment flag is set, the channel number is incremented automatically after each A/D conversion.

If the auto-increment mode is desired in applications where the internal oscillator is used, the analog output enable flag in the control byte (bit 6) should be set. This allows the internal oscillator to run continuously, thereby preventing conversion errors resulting from oscillator start-up delay. The analog output enable flag may be reset at other times to reduce quiescent power consumption.

The selection of a non-existing input channel results in the highest available channel number being allocated. Therefore, if the auto-increment flag is set, the next selected channel will be always channel 0. The most significant bits of both nibbles are reserved for future functions and have to be set to logic 0. After a Power-on reset condition all bits of the control register are reset to logic 0. The D/A converter and the oscillator are disabled for power saving. The analog output is switched to a high-impedance state.

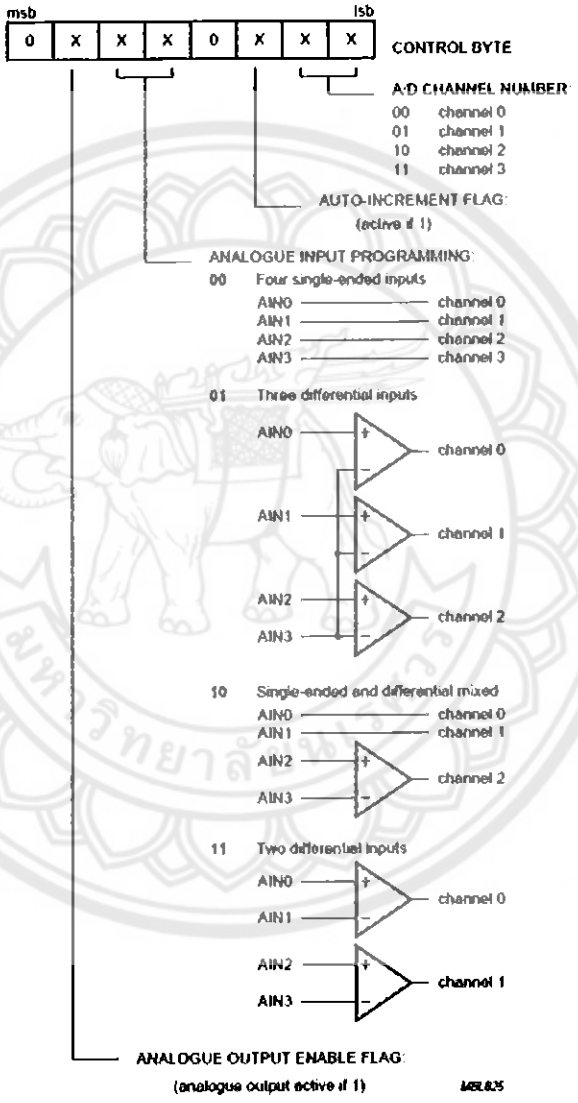


Fig.5 Control byte.

7.4 A/D conversion

The A/D converter makes use of the successive approximation conversion technique. The on-chip D/A converter and a high-gain comparator are used temporarily during an A/D conversion cycle.

An A/D conversion cycle is always started after sending a valid read mode address to a PCF8591 device. The A/D conversion cycle is triggered at the trailing edge of the acknowledge clock pulse and is executed while transmitting the result of the previous conversion (see Fig.9).

Once a conversion cycle is triggered an input voltage sample of the selected channel is stored on the chip and is converted to the corresponding 8-bit binary code. Samples picked up from differential inputs are converted to an 8-bit twos complement code (see Figs 10 and 11).

The conversion result is stored in the ADC data register and awaits transmission. If the auto-increment flag is set the next channel is selected.

The first byte transmitted in a read cycle contains the conversion result code of the previous read cycle. After a Power-on reset condition the first byte read is a hexadecimal 80. The protocol of an I²C-bus read cycle is shown in Chapter 8, Figs 16 and 17.

The maximum A/D conversion rate is given by the actual speed of the I²C-bus.

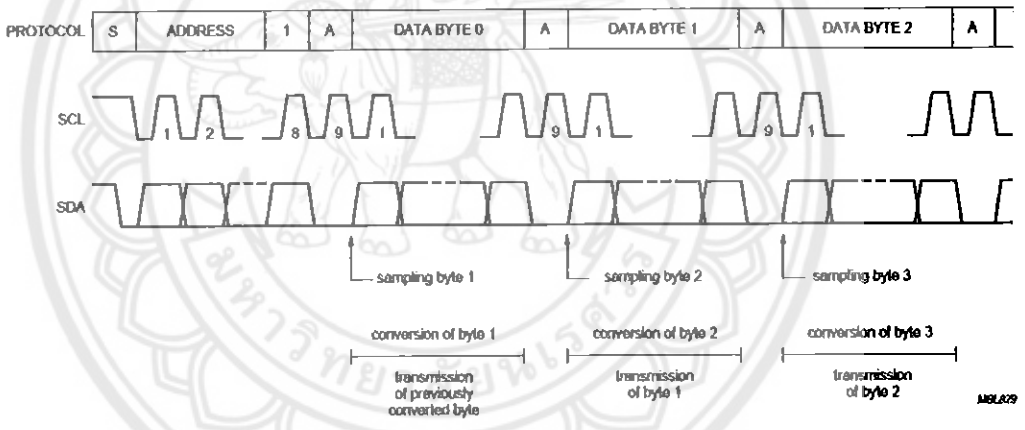


Fig.9 A/D conversion sequence.

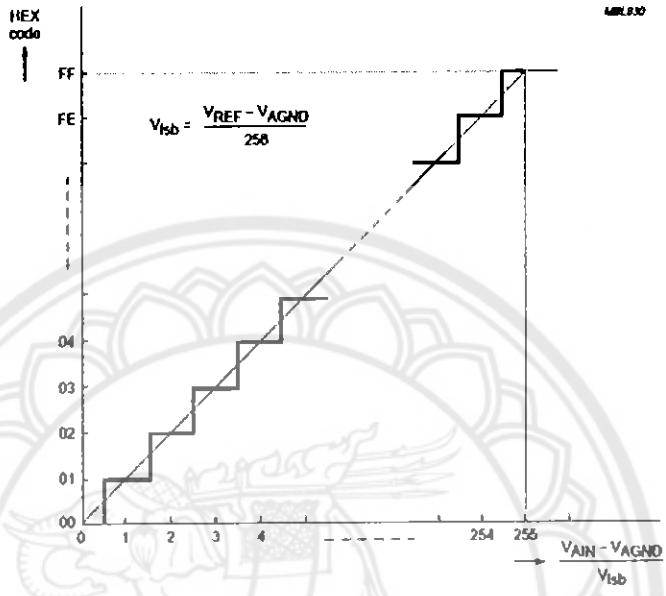


Fig.10 A/D conversion characteristics of single-ended inputs.

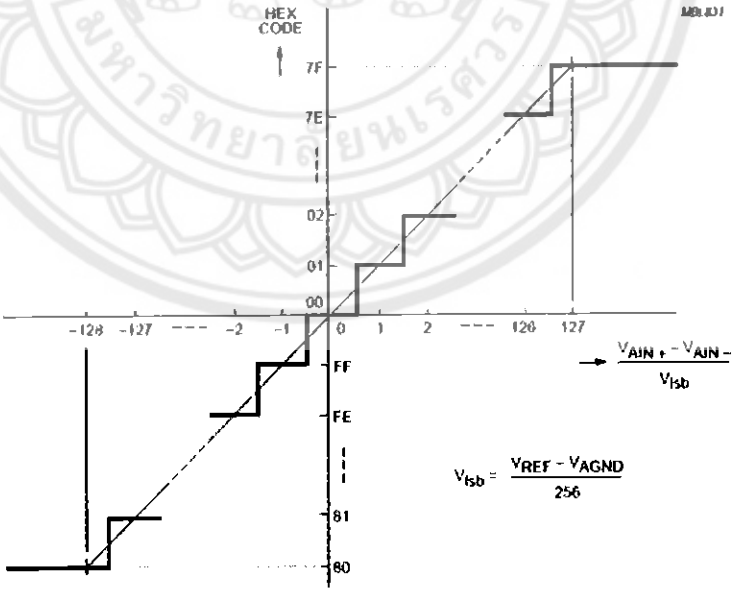


Fig.11 A/D conversion characteristics of differential inputs.

7.5 Reference voltage

For the D/A and A/D conversion either a stable external voltage reference or the supply voltage has to be applied to the resistor divider chain (pins V_{REF} and AGND). The AGND pin has to be connected to the system analog ground and may have a DC off-set with reference to V_{SS}.

A low frequency may be applied to the V_{REF} and AGND pins. This allows the use of the D/A converter as a one-quadrant multiplier; see Chapter 15 and Fig.7.

The A/D converter may also be used as a one or two quadrant analog divider. The analog input voltage is divided by the reference voltage. The result is converted to a binary code. In this application the user has to keep the reference voltage stable during the conversion cycle.

7.6 Oscillator

An on-chip oscillator generates the clock signal required for the A/D conversion cycle and for refreshing the auto-zeroed buffer amplifier. When using this oscillator the EXT pin has to be connected to V_{SS}. At the OSC pin the oscillator frequency is available.

If the EXT pin is connected to V_{DD} the oscillator output OSC is switched to a high-impedance state allowing the user to feed an external clock signal to OSC.

8 CHARACTERISTICS OF THE I²C-BUS

The I²C-bus is for bidirectional, two-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor. Data transfer may be initiated only when the bus is not busy.

8.1 Bit transfer

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as a control signal.

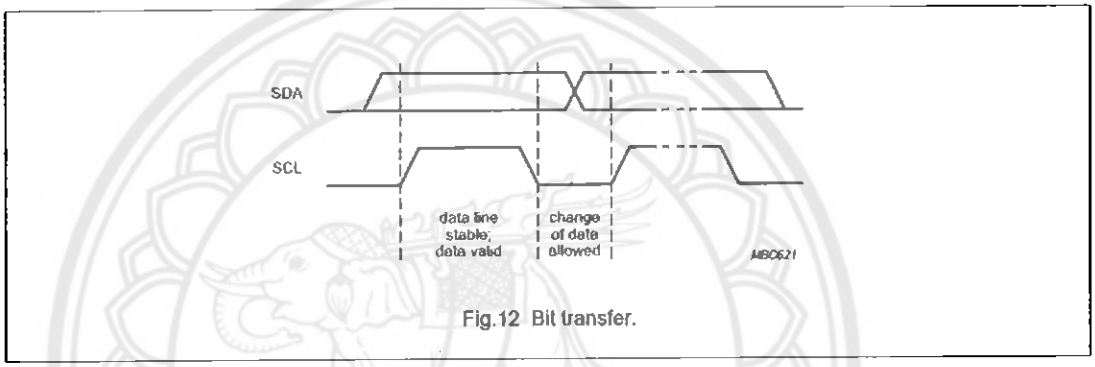


Fig.12 Bit transfer.

8.2 Start and stop conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line, while the clock is HIGH, is defined as the start condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH, is defined as the stop condition (P).

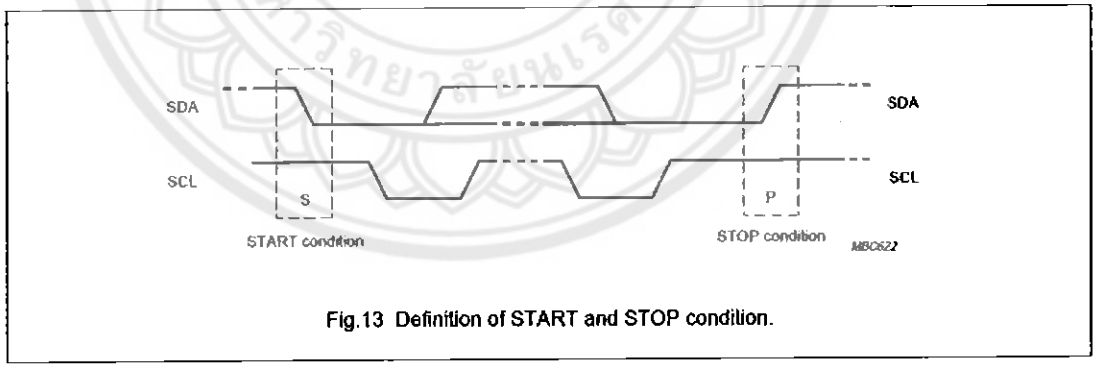


Fig.13 Definition of START and STOP condition.

8.5 I²C-bus protocol

After a start condition a valid hardware address has to be sent to a PCF8591 device. The read/write bit defines the direction of the following single or multiple byte data transfer. For the format and the timing of the start condition (S), the stop condition (P) and the acknowledge bit (A) refer to the I²C-bus characteristics. In the write mode a data transfer is terminated by sending either a stop condition or the start condition of the next data transfer.

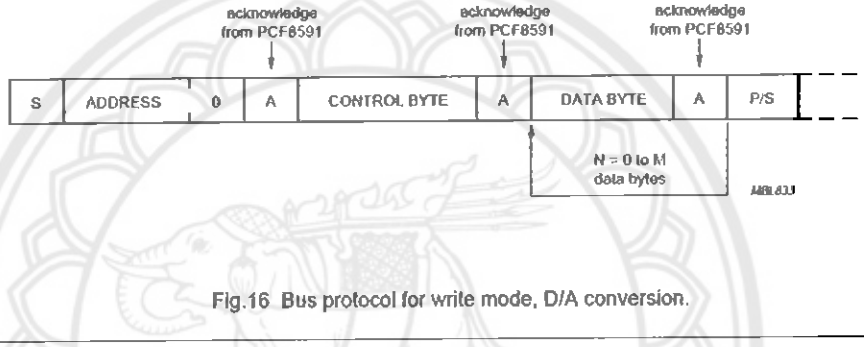


Fig.16 Bus protocol for write mode, D/A conversion.

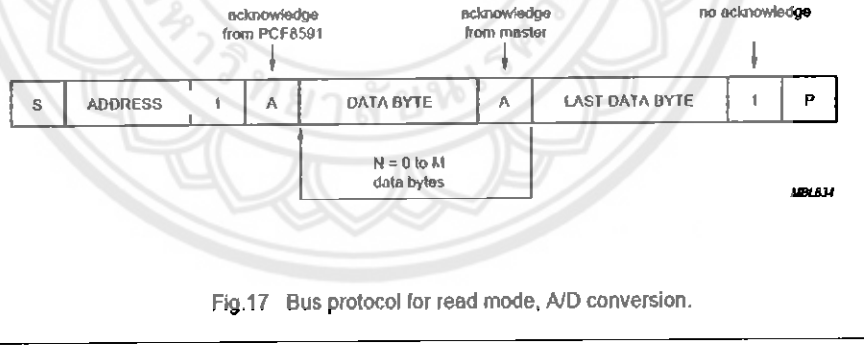


Fig.17 Bus protocol for read mode, A/D conversion.

9 LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 60134).

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
V_{DD}	supply voltage (pin 16)	−0.5	+8.0	V
V_I	input voltage (any input)	−0.5	V_{DD} + 0.5	V
I_I	DC input current	–	±10	mA
I_O	DC output current	–	±20	mA
I_{DD}, I_{SS}	V_{DD} or V_{SS} current	–	±50	mA
P_{tot}	total power dissipation per package	–	300	mW
P_O	power dissipation per output	–	100	mW
T_{amb}	operating ambient temperature	−40	+85	°C
T_{stg}	storage temperature	−65	+150	°C

10 HANDLING

Inputs and outputs are protected against electrostatic discharge in normal handling. However it is good practice to take normal precautions appropriate to handling MOS devices (see *"Handling MOS devices"*).

11 DC CHARACTERISTICS

V_{DD} = 2.5 V to 6 V; V_{SS} = 0 V; T_{amb} = −40 °C to +85 °C unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Supply						
V_{DD}	supply voltage (operating)		2.5	–	6.0	V
I_{DD}	supply current					
	standby	$V_I = V_{SS}$ or V_{DD}; no load	–	1	15	µA
	operating, AOUT off	f_{SCL} = 100 kHz	–	125	250	µA
	operating, AOUT active	f_{SCL} = 100 kHz	–	0.45	1.0	mA
V_{POR}	Power-on reset level	note 1	0.8	–	2.0	V
Digital inputs/output: SCL, SDA, A0, A1, A2						
V_{IL}	LOW level input voltage		0	–	$0.3 \times V_{DD}$	V
V_{IH}	HIGH level input voltage		$0.7 \times V_{DD}$	–	V_{DD}	V
I_L	leakage current					
	A0, A1, A2	$V_I = V_{SS}$ to V_{DD}	−250	–	+250	nA
	SCL, SDA	$V_I = V_{SS}$ to V_{DD}	−1	–	+1	µA
C_i	input capacitance		–	–	5	pF
I_{OL}	LOW level SDA output current	V_{OL} = 0.4 V	3.0	–	–	mA
Reference voltage inputs						
V_{REF}	reference voltage	$V_{REF} > V_{AGND}$; note 2	V_{SS} + 1.6	–	V_{DD}	V
V_{AGND}	analog ground voltage	$V_{REF} > V_{AGND}$; note 2	V_{SS}	–	V_{DD} − 0.8	V
I_{LI}	input leakage current		−250	–	+250	nA
R_{REF}	input resistance	pins V_{REF} and AGND	–	100	–	kΩ
Oscillator: OSC, EXT						
I_{LI}	input leakage current		–	–	250	nA
f_{OSC}	oscillator frequency		0.75	–	1.25	MHz

Notes

1. The power on reset circuit resets the I²C-bus logic when V_{DD} is less than V_{POR}.
2. A further extension of the range is possible, if the following conditions are fulfilled:

$$\frac{V_{REF} + V_{AGND}}{2} \geq 0.8V, V_{DD} - \frac{V_{REF} + V_{AGND}}{2} \geq 0.4V$$

12 D/A CHARACTERISTICS

V_{DD} = 5.0 V; V_{SS} = 0 V; V_{REF} = 5.0 V; V_{AGND} = 0 V; R_L = 10 kΩ; C_L = 100 pF; T_{amb} = −40 °C to +85 °C unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Analog output						
V_{OA}	output voltage	no resistive load	V_{SS}	–	V_{DD}	V
		R_L = 10 kΩ	V_{SS}	–	0.9 × V_{DD}	V
I_{LO}	output leakage current	AOUT disabled	–	–	250	nA
Accuracy						
OS_e	offset error	T_{amb} = 25 °C	–	–	50	mV
L_e	linearity error		–	–	±1.5	LSB
G_e	gain error	no resistive load	–	–	1	%
t_{DAC}	settling time	to ½ LSB full scale step	–	–	90	µs
f_{DAC}	conversion rate		–	–	11.1	kHz
SNRR	supply noise rejection ratio	f = 100 Hz; V_{DDN} = 0.1 × V_{PP}	–	40	–	dB

13 A/D CHARACTERISTICS

V_{DD} = 5.0 V; V_{SS} = 0 V; V_{REF} = 5.0 V; V_{AGND} = 0 V; R_S = 10 kΩ; T_{amb} = −40 °C to +85 °C unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Analog inputs						
V_{IA}	analog input voltage		V_{SS}	–	V_{DD}	V
I_{LIA}	analog input leakage current		–	–	100	nA
C_{IA}	analog input capacitance		–	10	–	pF
C_{iD}	differential input capacitance		–	10	–	pF
V_{IS}	single-ended voltage	measuring range	V_{AGND}	–	V_{REF}	V
V_{ID}	differential voltage	measuring range; $V_{FS} = V_{REF} - V_{AGND}$	$\frac{-V_{FS}}{2}$	–	$\frac{+V_{FS}}{2}$	V
Accuracy						
OS_e	offset error	T_{amb} = 25 °C	–	–	20	mV
L_e	linearity error		–	–	±1.5	LSB
G_e	gain error		–	–	1	%
GS_e	small-signal gain error	ΔV_i = 16 LSB	–	–	5	%
CMRR	common-mode rejection ratio		–	60	–	dB
SNRR	supply noise rejection ratio	f = 100 Hz; V_{DDN} = 0.1 × V_{PP}	–	40	–	dB
t_{ADC}	conversion time		–	–	90	µs
f_{ADC}	sampling/conversion rate		–	–	11.1	kHz

14 AC CHARACTERISTICS

All timing values are valid within the operating supply voltage and ambient temperature range and reference to V_{IL} and V_{IH} with an input voltage swing of V_{SS} to V_{DD}.

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT
I²C-bus timing (see Fig.20; note 1)					
f_{SCL}	SCL clock frequency	–	–	100	kHz
t_{SP}	tolerable spike width on bus	–	–	100	ns
t_{BUF}	bus free time	4.7	–	–	µs
$t_{SU;STA}$	START condition set-up time	4.7	–	–	µs
$t_{HD;STA}$	START condition hold time	4.0	–	–	µs
t_{LOW}	SCL LOW time	4.7	–	–	µs
t_{HIGH}	SCL HIGH time	4.0	–	–	µs
t_r	SCL and SDA rise time	–	–	1.0	µs
t_f	SCL and SDA fall time	–	–	0.3	µs
$t_{SU;DAT}$	data set-up time	250	–	–	ns
$t_{HD;DAT}$	data hold time	0	–	–	ns
$t_{VD;DAT}$	SCL LOW-to-data out valid	–	–	3.4	µs
$t_{SU;STO}$	STOP condition set-up time	4.0	–	–	µs

Note

1. A detailed description of the I²C-bus specification, with applications, is given in brochure *"The I²C-bus and how to use it"*. This brochure may be ordered using the code 9398 393 40011.

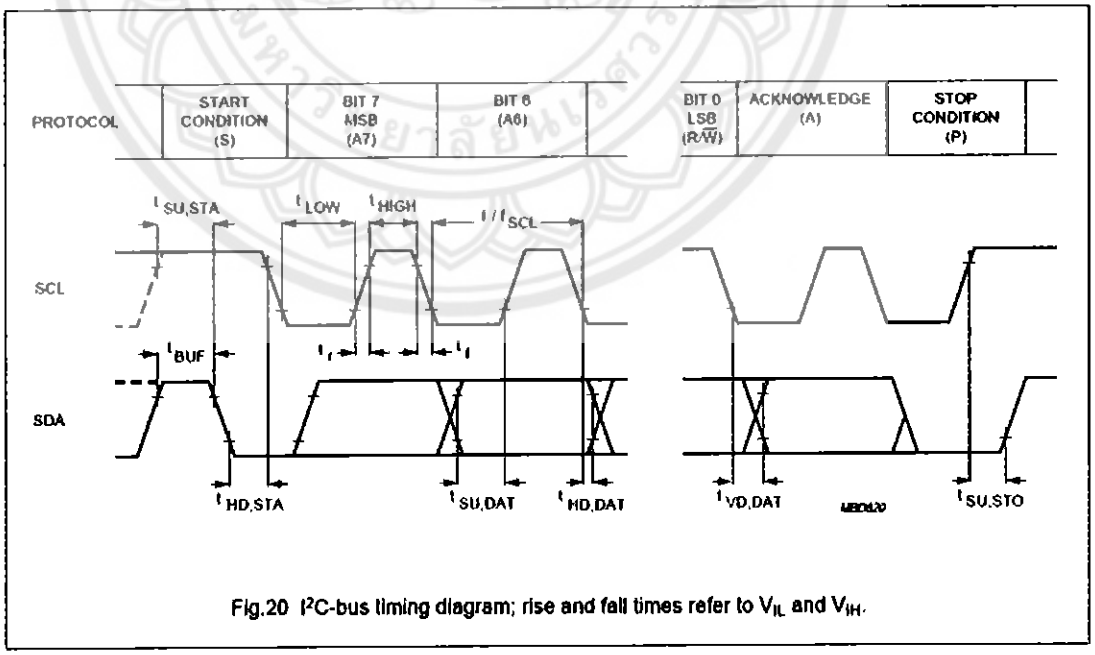


Fig.20 I²C-bus timing diagram; rise and fall times refer to V_{IL} and V_{IH}.

ภาคผนวก ค

รายละเอียดข้อมูลของตัวรับรู้ความดันส่วนต่าง รุ่น MPX5010DP

MOTOROLA
SEMICONDUCTOR TECHNICAL DATA

Order this document
by MPX5010/D

MPX5010 SERIES

INTEGRATED PRESSURE SENSOR
0 to 10 kPa (0 to 1.45 psi)
0.2 to 4.7 V OUTPUT

Integrated Silicon Pressure Sensor On-Chip Signal Conditioned, Temperature Compensated and Calibrated

The MPX5010 series piezoresistive transducer is a state–of–the–art monolithic silicon pressure sensor designed for a wide range of applications, but particularly those employing a microcontroller or microprocessor with A/D inputs. This patented, single element transducer combines advanced micromachining techniques, thin–film metallization, and bipolar processing to provide an accurate, high level analog output signal that is proportional to the applied pressure.

Features

- 5.0% Maximum Error over 0° to 85°C
- Ideally Suited for Microprocessor or Microcontroller–Based Systems
- Patented Silicon Shear Stress Strain Gauge
- Durable Epoxy Unibody Element
- Temperature Compensated over –40° to +125°C

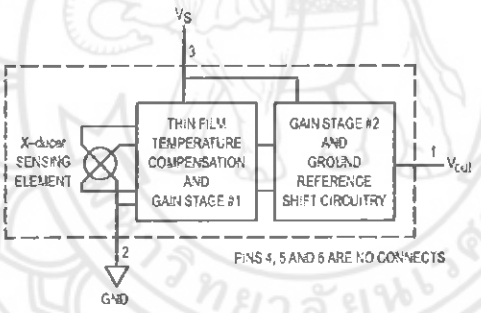
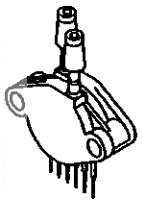


Figure 1. Fully Integrated Pressure Sensor Schematic

BASIC CHIP CARRIER ELEMENT
CASE 867–08, STYLE 1



DIFFERENTIAL PORT OPTION
CASE 867C–05, STYLE 1

PIN NUMBER			
1	V_{out}	4	N/C
2	Gnd	5	N/C
3	V_S	6	N/C

NOTE: Pins 4, 5, and 6 are internal device connections. Do not connect to external circuitry or ground. Pin 1 is noted by the notch in the Lead.

MAXIMUM RATINGS(1)

Parametrics	Symbol	Value	Unit
Overpressure(2) (P1 > P2)	P_{max}	75	kPa
Burst Pressure(2) (P1 > P2)	P_{burst}	100	kPa
Storage Temperature	T_{stg}	–40 to +125	°C
Operating Temperature	T_A	–40 to +125	°C

1. T_C = 25°C unless otherwise noted.
2. Exposure beyond the specified limits may cause permanent damage or degradation to the device.

Senseon and X–ducer are trademarks of Motorola, Inc.

MPX5010 SERIES

OPERATING CHARACTERISTICS (V_S = 5.0 Vdc, T_A = 25°C unless otherwise noted, P1 > P2)

Characteristic		Symbol	Min	Typ	Max	Unit
Pressure Range[1]		P_{OP}	0	—	10	kPa
Supply Voltage[2]		V_S	4.75	5.0	5.25	Vdc
Supply Current		I_o	—	7.0	10	mAdc
Minimum Pressure Offset[3] @ V_S = 5.0 Volts	(0 to 85°C)	V_{off}	0	0.2	0.425	Vdc
Full Scale Output[4] @ V_S = 5.0 Volts	(0 to 85°C)	V_{FSO}	4.475	4.7	4.925	Vdc
Full Scale Span[5] @ V_S = 5.0 Volts	(0 to 85°C)	V_{FSS}	—	4.5	—	Vdc
Accuracy[6]	(0 to 85°C)	—	—	—	±5.0	$\%V_{FSS}$
Sensitivity		V/P	—	450	—	mV/kPa
Response Time[7]		t_R	—	1.0	—	ms
Output Source Current at Full Scale Output		I_{O+}	—	0.1	—	mAdc
Warm-Up Time[8]		—	—	20	—	ms
Offset Stability[9]		—	—	±0.5	—	$\%V_{FSS}$

MECHANICAL CHARACTERISTICS

Characteristic	Symbol	Min	Typ	Max	Unit
Weight, Basic Element (Case 867)	—	—	4.0	—	Grams
Common Mode Line Pressure[10]	—	—	—	690	kPa

NOTES:

1. 1.0 kPa (kiloPascal) equals 0.145 psi.
2. Device is ratiometric within this specified excitation range.
3. Offset (V_{off}) is defined as the output voltage at the minimum rated pressure.
4. Full Scale Output (V_{FSO}) is defined as the output voltage at the maximum or full rated pressure.
5. Full Scale Span (V_{FSS}) is defined as the algebraic difference between the output voltage at full rated pressure and the output voltage at the minimum rated pressure.
6. Accuracy (error budget) consists of the following:
   - Linearity: Output deviation from a straight line relationship with pressure over the specified pressure range.
   - Temperature Hysteresis: Output deviation at any temperature within the operating temperature range, after the temperature is cycled to and from the minimum or maximum operating temperature points, with zero differential pressure applied.
   - Pressure Hysteresis: Output deviation at any pressure within the specified range, when this pressure is cycled to and from the minimum or maximum rated pressure, at 25°C.
   - TcSpan: Output deviation over the temperature range of 0° to 85°C, relative to 25°C.
   - TcOffset: Output deviation with minimum rated pressure applied, over the temperature range of 0° to 85°C, relative to 25°C.
   - Variation from Nominal: The variation from nominal values, for Offset or Full Scale Span, as a percent of V_{FSS}, at 25°C.
7. Response Time is defined as the time for the incremental change in the output to go from 10% to 90% of its final value when subjected to a specified step change in pressure.
8. Warm-up is defined as the time required for the product to meet the specified output voltage after the Pressure has been stabilized.
9. Offset stability is the product's output deviation when subjected to 1000 hours of Pulsed Pressure, Temperature Cycling with Bias Test.
10. Common mode pressures beyond what is specified may result in leakage at the case–to–lead interface.

ON–CHIP TEMPERATURE COMPENSATION, CALIBRATION AND SIGNAL CONDITIONING

Figure 2 illustrates the Differential/Gauge Sensing Chip in the basic chip carrier (Case 867). A fluorosilicone gel isolates the die surface and wire bonds from the environment, while allowing the pressure signal to be transmitted to the sensor diaphragm.

The MPX5010 series pressure sensor operating characteristics, and internal reliability and qualification tests are based on use of dry air as the pressure media. Media, other than dry air, may have adverse effects on sensor performance and long–term reliability. Contact the factory for information regarding media compatibility in your application.

Figure 3 shows a typical decoupling circuit for interfacing the integrated sensor to the A/D input of a microprocessor. Proper decoupling of the power supply is recommended.

Figure 4 shows the sensor output signal relative to pressure input. Typical, minimum, and maximum output curves are shown for operation over a temperature range of 0° to 85°C using the decoupling circuit below. (The output will saturate outside of the specified pressure range.)

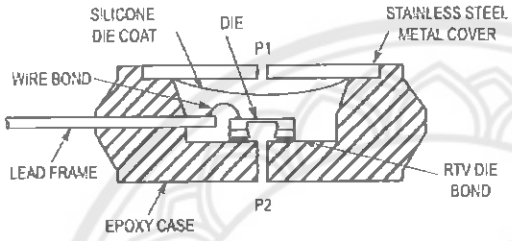


Figure 2. Cross–Sectional Diagram (Not to Scale)

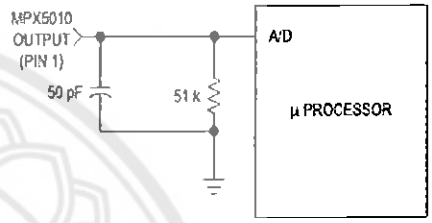


Figure 3. Typical Decoupling Filter for Sensor to Microprocessor Interface

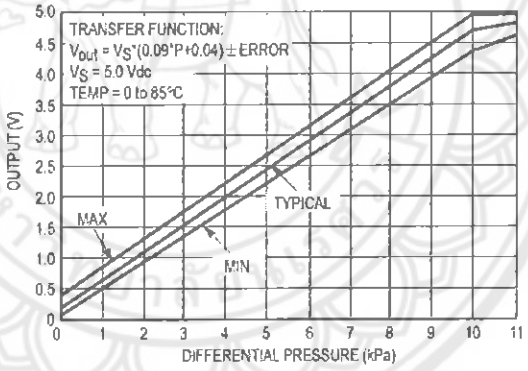


Figure 4. Output versus Pressure Differential

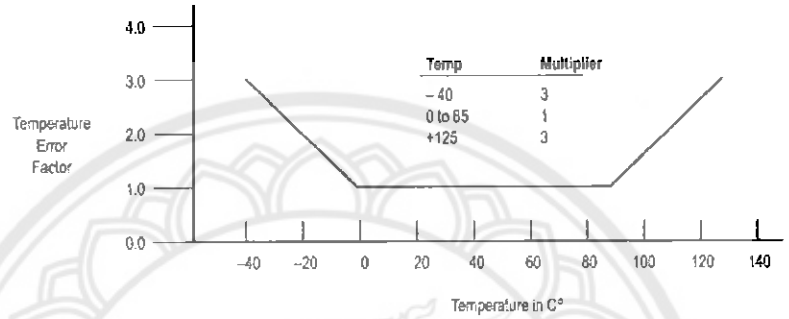
MPX5010 SERIES

Transfer Function (MPX5010D)

Nominal Transfer Value: $V_{out} = V_S \times (0.09 \times P + 0.04)$
$\pm$ (Pressure Error x Temp. Factor x 0.09 x V_S)
$V_S = 5.0\ V \pm 0.25\ Vdc$

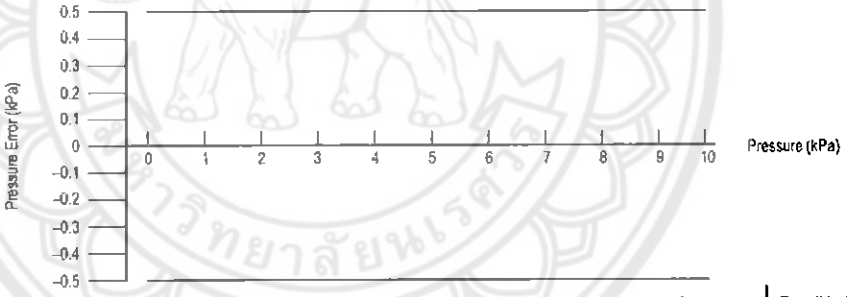
Temperature Error Band

MPX5010D Series



NOTE: The Temperature Multiplier is a linear response from 0° to –40°C and from 85° to 125°C.

Pressure Error Band



Pressure	Error (Max)
0 to 10 kPa	± 0.5 kPa

PRESSURE (P1)/VACUUM (P2) SIDE IDENTIFICATION TABLE

Motorola designates the two sides of the pressure sensor as the Pressure (P1) side and the Vacuum (P2) side. The Pressure (P1) side is the side containing fluoro silicone gel which protects the die from harsh media. The Motorola MPX pressure sensor is designed to operate with positive differential pressure applied, P1 > P2.

The Pressure (P1) side may be identified by using the table below:

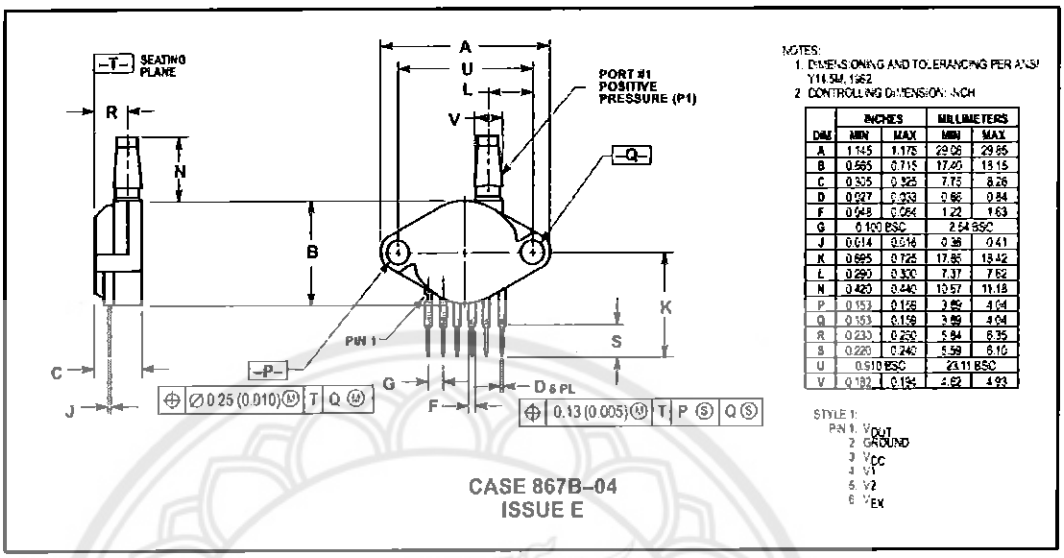
Part Number	Case Type	Pressure (P1) Side Identifier
MPX5010D	867–08	Stainless Steel Cap
MPX5010DP	867C–05	Side with Part Marking
MPX5010GP	867B–04	Side with Port Attached
MPX5010GVP	867D–04	Stainless Steel Cap
MPX5010GS	867E–03	Side with Port Attached
MPX5010GVS	867A–04	Stainless Steel Cap
MPX5010GSX	867F–03	Side with Port Attached
MPX5010GVSX	867G–03	Stainless Steel Cap

ORDERING INFORMATION

The MPX5010 pressure sensor is available in differential and gauge configurations. Devices are available in the basic element package or with pressure port fittings that provide printed circuit board mounting ease and barbed hose pressure connections.

Device Type	Options	Case Type	MPX Series	
			Order Number	Device Marking
Basic Element	Differential	867–08	MPX5010D	MPX5010D
Ported Elements	Differential Dual Ports	867C–05	MPX5010DP	MPX5010DP
	Gauge	867B–04	MPX5010GP	MPX5010GP
	Gauge Vacuum Port	867D–04	MPX5010GVP	MPX5010GVP
	Gauge, Axial	867E–03	MPX5010GS	MPX5010D
	Gauge Vacuum Axial	867A–04	MPX5010GVS	MPX5010D
	Gauge, Axial PC Mount	867F–03	MPX5010GSX	MPX5010D
	Gauge Vacuum Axial PC Mount	867G–03	MPX5010GVSX	MPX5010D

PACKAGE DIMENSIONS–CONTINUED

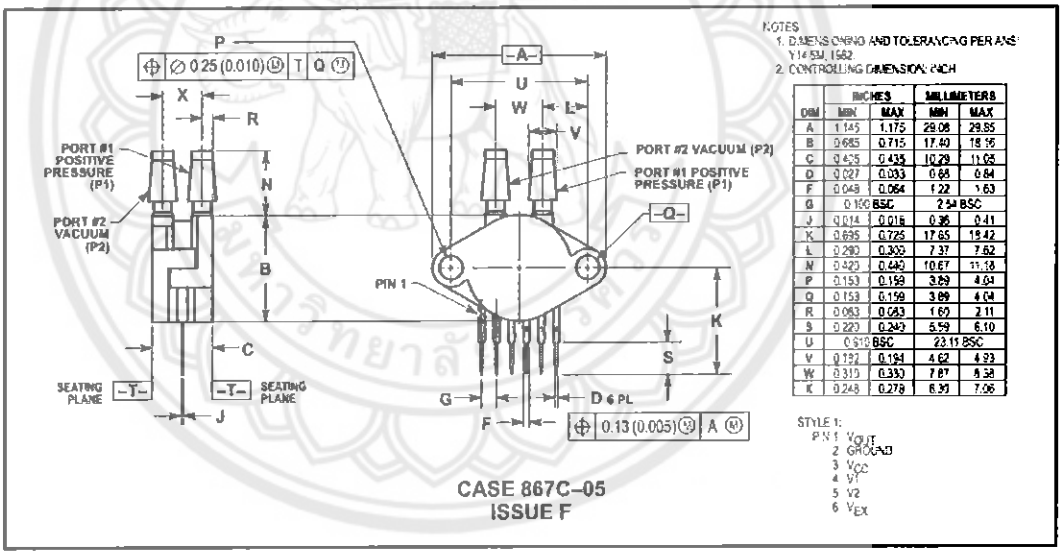


NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.

DIM	INCHES MIN	INCHES MAX	MILLIMETERS MIN	MILLIMETERS MAX
A	1.145	1.175	29.08	29.85
B	0.685	0.715	17.40	18.16
C	0.305	0.325	7.75	8.26
D	0.027	0.033	0.68	0.84
F	0.048	0.064	1.22	1.63
G	0.100 BSC		2.54 BSC	
J	0.014	0.016	0.36	0.41
K	0.695	0.725	17.65	18.42
L	0.290	0.300	7.37	7.62
N	0.420	0.440	10.67	11.18
P	0.153	0.159	3.89	4.04
Q	0.153	0.159	3.89	4.04
R	0.230	0.250	5.84	6.35
S	0.220	0.240	5.59	6.10
U	0.910 BSC		23.11 BSC	
V	0.182	0.194	4.62	4.93

STYLE 1:
PIN 1. V_{OUT}
2. GROUND
3. V_{CC}
4. V1
5. V2
6. V_{EX}

PRESSURE SIDE PORTED (GP)



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.

DIM	INCHES MIN	INCHES MAX	MILLIMETERS MIN	MILLIMETERS MAX
A	1.145	1.175	29.08	29.85
B	0.685	0.715	17.40	18.16
C	0.425	0.435	10.29	11.05
D	0.027	0.033	0.68	0.84
F	0.048	0.064	1.22	1.63
G	0.100 BSC		2.54 BSC	
J	0.014	0.016	0.36	0.41
K	0.695	0.725	17.65	18.42
L	0.290	0.300	7.37	7.62
N	0.420	0.440	10.67	11.18
P	0.153	0.159	3.89	4.04
Q	0.153	0.159	3.89	4.04
R	0.063	0.083	1.60	2.11
S	0.220	0.240	5.59	6.10
U	0.910 BSC		23.11 BSC	
V	0.182	0.194	4.62	4.93
W	0.310	0.330	7.87	8.38
X	0.248	0.278	6.30	7.06

STYLE 1:
PIN 1. V_{OUT}
2. GROUND
3. V_{CC}
4. V1
5. V2
6. V_{EX}

PRESSURE AND VACUUM SIDES PORTED (DP)