เครื่องวัดระดับน้ำมันโดยอาศัยหลักการสะท้อนคลื่นอัลตร้าโซนิค

โดย

นายธนวัฒน์	ควรรติกุล	รหัสนักศึกษา	B4603436
นายบูรนิติ	มาศอมรพันธุ์	รหัสนักศึกษา	B4604747
นางสาวรัชดาพร	มานุวงศ์	รหัสนักศึกษา	B4607373

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา 427499 โครงงานศึกษาวิศวกรรมโทรคมนาคม

ประจำภาคการศึกษาที่ 3 ปีการศึกษา 2550

หลักสูตรวิศวกรรมศาสตร์บัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม

หลักสูตรปรับปรุงพ.ศ. 2546

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

โครงงาน เครื่องวัดระดับน้ำมันโดยอาศัยคลื่นหลักการสะท้อนอัลตร้าโซนิค

ผู้ดำเนินงาน 1. นายธนวัฒน์ ควรรติกุล รหัสนักศึกษา B4603436
2. นายบูรนิติ มาศอมรพันธุ์ รหัสนักศึกษา B4604747
3. นางสาวรัชดาพร มานุวงศ์ รหัสนักศึกษา B4607373

อาจารย์ที่ปรึกษา ผศ.ดร.รังสรรค์ วงศ์สรรค์
สาขาวิชา วิศวกรรมโทรคมนาคม
ภาคการศึกษาที่ 1/2549

บทคัดย่อ

โครงงานฉบับนี้เป็นการวิเคราะห์และออกแบบวงจรคำนวณระดับน้ำมันในถังน้ำมัน รวมถึงการประเมินและทดสอบถึงประสิทธิภาพของเครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิค เป็นตัวบ่งบอกถึงระดับน้ำมันในถังน้ำมัน โดยตัวที่บ่งบอกคือ แผงวงจรอิเล็คทรอนิกส์ที่ได้ผ่านการออกแบบและเขียนชุดคำสั่ง ซึ่งผลที่ได้คือความถี่ และนำความถี่ที่ได้นั้นมาประมวลและวิเคราะห์ผล เพื่อที่จะได้ทราบถึงระดับปริมาณน้ำมันในถัง ซึ่งเราสามารถนำผลที่ได้มาใช้ประโยชน์ในการวิเคราะห์ ถึงระดับน้ำมันในถังที่คงเหลืออยู่ในถังน้ำมัน เพื่อที่จะสามารถนำไปใช้ประโยชน์ได้จริง

กิตติกรรมประกาศ

ในการจัดทำโครงงานนี้จะมิอาจสำเร็จลุลวงไปได้ด้วยดี ถ้าหากมิได้รับความกรุณาจากอาจารย์ที่ปรึกษา ผศ.ดร.รังสรรค์ วงศ์สรรค์ ผู้ที่ให้แนวคิดแรกเริ่มของวงจรเชื่อมต่อคอมพิวเตอร์ช่วยสอนปฏิบัติการ ที่ให้ความช่วยเหลือในเรื่องของแนวคิด การติดตามเอาใจใส่ และชี้แนะข้อบกพร่องที่ข้าพเจ้าได้มองข้ามไปในบางส่วน

นอกจากนี้ยังต้องขอขอบพระคุณผู้ที่เกี่ยวข้อง ดังนี้

คณาจารย์ประจำสาขาวิชาวิศวกรรมโทรคมนาคมทุกท่าน ที่ได้สั่งสอนให้ความรู้

คุณประพล จาระตะคุ ผู้ที่ให้ความช่วยเหลือในเรื่องของการเบิกจ่ายงบประมาณ ตลอดจนอุปกรณ์ในการทดลองต่างๆ

คุณมณีรัตน์ ทุมพงษ์ เลขานุการประจำสาขาวิชาวิศวกรรมโทรคมนาคม ที่ให้ความช่วยเหลือในเรื่องของงานเอกสาร

นายไพโรจน์ บุญไทย ที่ช่วยให้คำแนะนำเกี่ยวกับอุปกรณ์ไมโครคอลโทรเลอร์

พี่ๆและเพื่อนๆ สาขาวิชาวิศวกรรมโทรคมนาคม ที่ให้ความช่วยเหลือในเรื่องของการเขียนโปรแกรมและการทดสอบโปรแกรมที่ใช้ในการวัดปริมาตรที่เหลือ

สุดท้ายนี้ คุณงานความดีที่เกิดจากโครงงานฉบับนี้ ขอมอบแก่บิดามารดา ผู้ที่คอยห่วงใย ให้กำลังใจ ให้โอกาส และให้การสนับสนุนทางการศึกษามาโดยตลอด

นายธนวัฒน์ ควรรติกุล
นายบูรนิติ มาศอมรพันธุ์
นางสาวรัชดาพร มานุวงศ์

สารบัญ

	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญรูป	ฉ
สารบัญตาราง	ซ
สารบัญตาราง	ฌ
บทที่ 1 บทนำ	
1.1 ความเป็นมา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของการทำงาน	1
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและหลักการ	
2.1 ไมโครคอนโทรลเลอร์	3
2.1.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์ตระกูล MCS-51	3
2.1.2 คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ตระกูล MCS-51	4
2.1.3 การจัดขาของไมโครคอนโทรลเลอร์ตระกูล MCS-51	4
2.2 คลื่นอัลตร้าโซนิค(Ultrasonic Sensor)	6
2.3 USB (Universal Serial Bus)	8
2.3.1 ระบบการจัดส่งข้อมูล USB	8
2.4 สรุปเรื่องทฤษฎีและหลักการ	9

สารบัญ (ต่อ)

หน้า

บทที่ 3 การออกแบบและสร้างวงจรสร้างและรับคลื่นอัลตร้าโซนิค ในการตรวจวัดระดับน้ำมัน

3.1 กล่าวนำ 10

3.2 เซนเซอร์วัดระยะทาง Ultrasonic Distance Detector Module รุ่นSRF 05 10

3.2.1 การต่อใช้งานของ SRF05 12

3.3 วงจรไมโครคอนโทรเลอร์ 14

3.4 วงจรเชื่อมต่อ USB 14

3.4.1 การออกแบบลายวงจรเชื่อมต่อ USB ลงบนแผ่นวงจรพิมพ์ 14

3.4.2 การเลือกใช้ไอซีในการออกแบบ 16

3.4.3 การติดตั้งอุปกรณ์อิเล็คทรอนิคส์ตามที่ได้ออกแบบ 17

3.5 จอแสดงผล (LCD) 18

3.6 ชุดปุ่มกด (KEYPAD) 18

3.7 สรุปเรื่องการออกแบบและสร้างวงจรสร้างและรับคลื่นอัลตร้าโซนิค ในการตรวจวัดระดับน้ำมันในถังน้ำมัน 20

บทที่ 4 การทดลองเครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิค

4.1 กล่าวนำ 21

4.2 การใช้โปรแกรม FLIP V2.2.4 และATMEL Microcontroller 21

4.2.1 วิธีการ Download Program ให้กับ MCU 23

4.2.2 การ Download แบบ Manual 23

4.2.3 ลำดับขั้นตอนการดาวน์โหลด HEX File แบบธรรมดา 25

4.2.4 การ Download แบบอัตโนมัติ 29

4.2.5 ลำดับขั้นตอนการ Download HEX File แบบอัตโนมัติ 29

4.2.6 ปัญหาต่างๆในขณะใช้งานโปรแกรม FLIP และแนวทางการแก้ไข 34

4.3 การเขียนชุดโปรแกรมคำสั่งควบคุมอุปกรณ์ต่างๆ 37

4.3.1 ลำดับความคิดในการเขียนชุดโประแกรมคำสั่งควบคุม อุปกรณ์ต่างๆ (Algorithms) 37

สารบัญ (ต่อ)

	หน้า
4.3.2 ชุดโปรแกรมคำสั่งควบคุมเซนเซอร์วัดระยะทาง ด้วยคลื่นอัลตร้าโซนิค SRF05	38
4.3.3 ชุดโปรแกรมคำสั่งควบคุมจอแสดงผล (LCD)	39
4.3.4 ชุดโปรแกรมคำสั่งควบคุมการรับค่าจากชุดปุ่มกด (Keypad)	40
4.3.5 ชุดโปรแกรมคำสั่งควบคุมการคำนวณหาปริมาตร	41
4.4 การทดสอบเครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิค ในถังรูปแบบต่างๆ	49
4.5 สรุปการทดลองเครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิค	55
บทที่ 5 สรุปและข้อเสนอแนะ	
5.1 กล่าวนำ	56
5.2 ปัญหาที่พบในระหว่างการทำโครงงานและวิธีการแก้ปัญหา	56
5.3 ข้อเสนอแนะ	57
5.4 แนวทางการพัฒนาต่อไป	57
5.5 บทสรุป	57
บรรณานุกรม	59
ภาคผนวก (ก)	60
Data Sheet	61
Ultrasonic Distance Detector Module-SRF05	62
FT232BL USB UART (USB Serial) I.C.	78
HY-1601-C802 LCD Module	103
ET-MINIKEY-4x4 (KEYPAD)	129
ภาคผนวก (ข)	130
ประวัติผู้จัดทำ	132

สารบัญรูป

	หน้า
2.1 บล็อกไดอะแกรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51	3
2.2 การจัดขาของ MCS-51	4
2.3 สัญลักษณ์ของ Ultrasonic	7
วงจรชุดส่ง Ultrasonic	7
วงจรชุดรับ Ultrasonic	8
2.4 การเชื่อมต่อพอร์ต USB	9
3.1 แสดงหลักการตรวจจับวัตถุโดยใช้สัญญาณความถี่เหนือเสียง	10
3.2 (ก) แสดงเซ็นเซอร์วัดระยะทาง รุ่นSRF 05	11
3.2 (ข) แสดงขาสัญญาณของ SRF05 และการกำหนดโหมดทำงาน	11
3.3 ไดอะแกรมเวลาแสดงสัญญาณที่ส่งไปยังเครื่องวัดระยะทางด้วยอัลตร้าโซนิก	13
3.4 ไดอะแกรมเวลาแสดงสัญญาณที่ส่งไปยังเครื่องตรวจจับและวัดระยะทาง	13
3.5 ชุดวงจรไมโครคอนโทรลเลอร์ AT89C51ED	14
3.6 วงจรเชื่อมต่อ USB	15
3.7 ลายวงจรที่ได้จากโปรแกรม Protel99SE บนแผ่นวงจรพิมพ์	15
3.8 USB-to-Serial (Single Channel) Controller Block Diagram	16
3.9 Top View ของ TUSB3410	17
3.10 ชุดวงจรเชื่อมต่อ USB	17
3.11 จอแสดงผล LCD รุ่น LCD 16 CHARS x1 LINE,AV	18
3.12 ชุดปุ่มกด KEYPAD-Module ET-MINIKEY	19
3.13 แสดงรูปภายในกล่อง	20
3.14 แสดงภายนอกกล่องพร้อมใช้งาน	20
4.1 (ก) ถังน้ำมันรูปทรงลูกษาศก์	21
4.1 (ข) ถังรูปทรงทรงกระบอก	21
4.2 การสั่ง Run โปรแกรม FLIP V2.4.4	25
4.3 การสั่งเลือกกำหนดเบอร์ของ MCU ในบอร์ดสำหรับการดาวน์โหลดโปรแกม	25
4.4 การกำหนด Comport ให้ตรงกับสายที่ทำการต่อไว้	26
4.5 ทำการติดต่อติดต่อสื่อสารกับ MCU ใน Monitor Mode	26

สารบัญรูป (ต่อ)

หน้า

4.6 การเปิด Hex File ที่ต้องการจะ Download 27
4.7 กำหนดค่า Operation Flow 28
4.8 ตรวจสอบค่า Device BSB และ SBV 28
4.9 การสั่ง Run โปรแกรม FLIP V2.4.4 29
4.10 การสั่งเลือกกำหนดเบอร์ของ MCU ในบอร์ดสำหรับการดาวน์โหลดโปรแกม 30
4.11 กำหนดค่าต่างๆใน Preferences 30
4.12 กำหนดค่าที่จะใช้ในการควบคุมสัญญาณ RESET และ PSEN
ของ MCU ในบอร์ด 31
4.13 การกำหนด Comport ให้ตรงกับสายที่ทำการต่อไว้ 31
4.14 เริ่มต้นทำการติดต่อสื่อสารกับ MCU ใน Monitor Mode 32
4.15 การเปิด Hex File ที่ต้องการดาวน์โหลดให้กับ MCU
เพื่อให้รอใน Buffer ของโปรแกรม 32
4.16 กำหนดค่าของ Operation Flow 33
4.17 ตรวจสอบค่า Device BSB และ SBV ให้เป็น 00 ทั้งหมด 33
4.18 Error แสดงว่าไฟล์ที่เลือกไม่ใช่ HEX File 36
4.19 แผนภาพแสดงความคิดในการเขียนชุดโประแกรมคำสั่ง
ควบคุมอุปกรณ์ต่างๆ (Flowchart) 39

สารบัญตาราง

หน้า

4.1 ตารางแสดงผลการทดสอบเครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิค ในถังทรงสี่เหลี่ยม 49

4.2 ตารางแสดงผลการทดสอบเครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิค ในถังทรงกระบอก 50

4.3 ตารางแสดงผลการทดสอบเครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิค ในถังทรงสี่เหลี่ยมโดยการใช้น้ำแทนการใช้น้ำมัน 51

4.4 ตารางแสดงผลการทดสอบเครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิค ในถังทรงกระบอกโดยการใช้น้ำแทนการใช้น้ำมัน 51

5.1 แสดงรายละเอียดของปัญหาที่พบ และวิธีแก้ปัญหาของโครงงาน 56

สารบัญกราฟ

	หน้า
4.19 กราฟระหว่างค่าความผิดพลาดกับปริมาตรที่ทำการวัดในถังทรงสี่เหลี่ยม	53
4.20 กราฟระหว่างค่าความผิดพลาดกับปริมาตรที่ทำการวัด ในถังทรงกระบอก	53
4.21 กราฟระหว่างค่าความผิดพลาดกับปริมาตรที่ทำการวัด ในถังทรงสี่เหลี่ยม โดยการใช้น้ำแทนการใช้น้ำมัน	54
4.22 กราฟระหว่างค่าความผิดพลาดกับปริมาตรที่ทำการวัด ในถังทรงกระบอก โดยการใช้น้ำแทนการใช้น้ำมัน	54

บทที่ 1
บทนำ

1.1 ความเป็นมา

เนื่องจากในยุคปัจจุบันมีการติดต่อค้าขายกันมากขึ้น ยานพาหนะจึงมีความจำเป็นสำหรับการขนส่งสินค้า ซึ่งยานพาหนะจะขับเคลื่อนได้นั้น จำเป็นต้องใช้น้ำมัน และในยานพาหนะแต่ละชนิดต่างใช้น้ำมันในการขับเคลื่อนไม่เท่ากัน ดังนั้นหากทราบระดับน้ำมันในถังน้ำมันที่แน่นอน ก็จะสามารถทำให้คาดการณ์ว่าน้ำมันที่เหลืออยู่ในถังน้ำมันเพียงพอที่จะทำให้ถึงจุดหมายปลายทางได้หรือไม่ แต่สำหรับยานพาหนะรุ่นเก่าจะมีตัวบ่งชี้ปริมาณน้ำมันที่ไม่ละเอียดเท่าที่ควร

โครงงานนี้จึงนำความรู้เรื่องเซนเซอร์แบบอัลตราโซนิค (Ultrasonic Sensor) ซึ่งอาศัยหลักการการสะท้อนของเสียงที่ปล่อยออกไปยังเป้าหมายที่ต้องการ ทำให้ทราบถึงระยะห่างระหว่างต้นทางและปลายทาง เพื่อนำมาประยุกต์ใช้เป็นส่วนประกอบในการจัดทำเครื่องวัดปริมาณน้ำมันและนำผลที่ได้มาทำการวิเคราะห์และประมวลผล ซึ่งจะทำให้ทราบถึงระดับน้ำมันในถังบรรจุน้ำมันได้อย่างถูกต้องและมีความละเอียดยิ่งขึ้น เพื่อที่จะนำไปประยุกต์ใช้งานได้จริงอย่างมีประสิทธิภาพ

1.2 วัตถุประสงค์

1.2.1 เพื่อศึกษาวิธีการหลักการและวิธีการทำงานของเซนเซอร์แบบอัลตราโซนิค (Ultrasonic Sensor)

1.2.2 เพื่อศึกษาผลลัพธ์ที่ได้จากเซนเซอร์แบบอัลตราโซนิค (Ultrasonic Sensor) มาใช้ในการหาปริมาตรน้ำมันในถัง โดยใช้ไมโครคอนโทรเลอร์เป็นตัวประมวลผล และแสดงผลออกที่หน้าจอแสดงผล

1.3 ขอบเขตของการทำงาน

1.3.1 ศึกษาค้นคว้าข้อมูลเกี่ยวกับทฤษฎีคลื่นอัลตร้าโซนิค (Ultrasonic Wave)

1.3.2 ทำการทดสอบค่าความสูงที่วัดได้จากเซนเซอร์แบบอัลตราโซนิค (Ultrasonic Sensor) ทำการออกแบบโปรแกรมประมวลผลผลลัพธ์ที่ได้เซนเซอร์แบบอัลตราโซนิค (Ultrasonic Sensor) เพื่อคำนวณหาปริมาตรคงเหลือของน้ำมันที่มีในถังทดลอง

1.3.3 ปรับแต่งโปรแกรมเพื่อให้ทำงานได้ตามคุณสมบัติที่คาดหวังไว้

1.3.4 สามารถนำไปใช้งานได้จริงในการวัดระดับน้ำมัน

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาและทำการค้นคว้าเกี่ยวกับการทำงานของเซนเซอร์อัลตร้าโซนิค (Ultrasonic Sensor) ที่มี

1.4.2 นำผลลัพธ์ที่ได้จากเซนเซอร์แบบอัลตราโซนิค (Ultrasonic Sensor) มาทำการประมวลผลโดยใช้ไมโครคอลโทลเลอร์ (MCS 51) เพื่อคำนวณหาปริมาตรน้ำมันในถัง

1.4.3 ทำการปรับแต่งให้สามารถใช้งานได้ง่ายขึ้น

1.4.4 ทำการทดลองจริงกับน้ำมัน

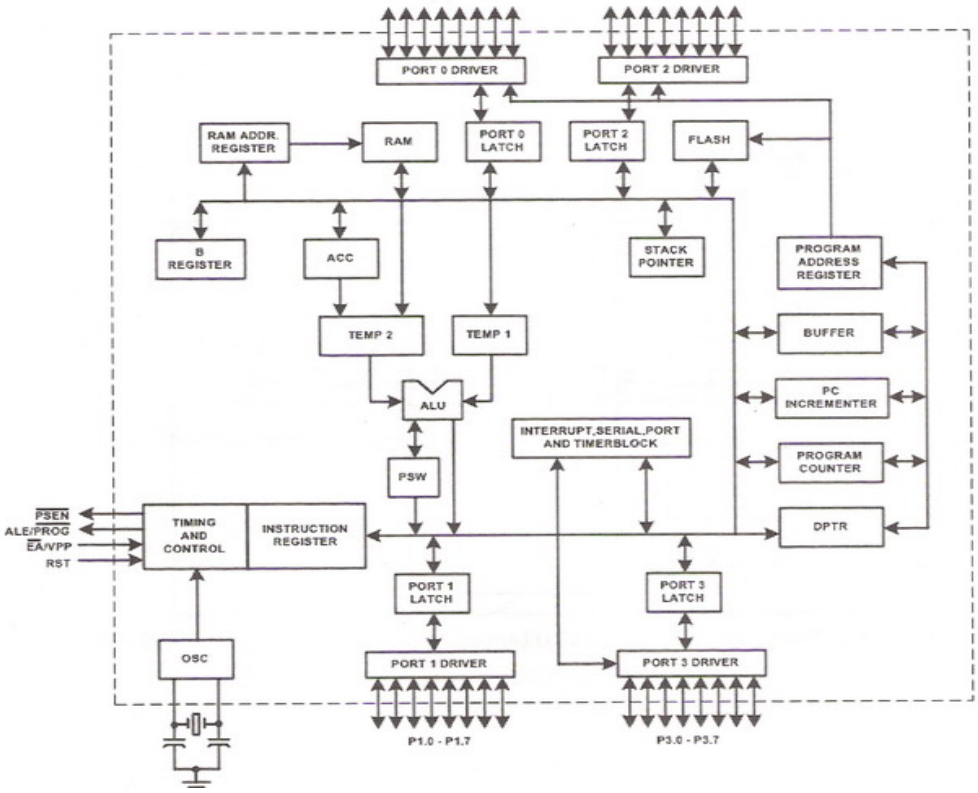
บทที่ 2
ทฤษฎีและหลักการ

2.1 ไมโครคอนโทรลเลอร์ MCS-51 [1]

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ได้จัดให้มีส่วนประกอบภายในเพื่ออำนวยความสะดวกแก่ผู้ใช้ เช่น ไทเมอร์ (Timer) เคาน์เตอร์ (Counter) พอร์ตอนุกรม (Serial port) และสำหรับไมโครคอนโทรลเลอร์เบอร์ใหญ่ ๆ ยังอาจมีส่วนอื่นเพิ่มเติมเข้ามาอีก เช่น เบอร์ 80C515, 80C535 จะมีวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล

2.1.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ใช้เทคโนโลยีการผลิตแบบ NMOS และ CMOS ซึ่งภายในได้รวมวงจรต่างไว้อย่างครบถ้วนพร้อมที่จะทำงานได้เมื่อจ่ายไฟเลี้ยงและสัญญาณนาฬิกา (Clock) ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ได้ถูกผลิตออกมามากมายหลายเบอร์โดยบริษัทต่าง ๆ ไม่ว่าจะเป็นเบอร์อะไรก็ตาม ถ้าเป็นไมโครคอนโทรลเลอร์ตระกูล MCS-51 แล้วจะมีโครงสร้างต่าง ๆ ที่คล้ายกัน จะต่างกันออกไปในส่วนของความสามารถพิเศษของแต่ละเบอร์ ยกตัวอย่างเช่นเบอร์ AT89C51 มีไทเมอร์ 2 ตัว ในขณะที่เบอร์ AT89C52 มีไทเมอร์ 3 ตัว เป็นต้น โดยที่โครงสร้างต่างๆ ภายในของไมโครคอนโทรลเลอร์ตระกูล MCS-51 ได้แสดไว้ดังรูปที่ 2.1



รูปที่ 2.1 บล็อกไดอะแกรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51

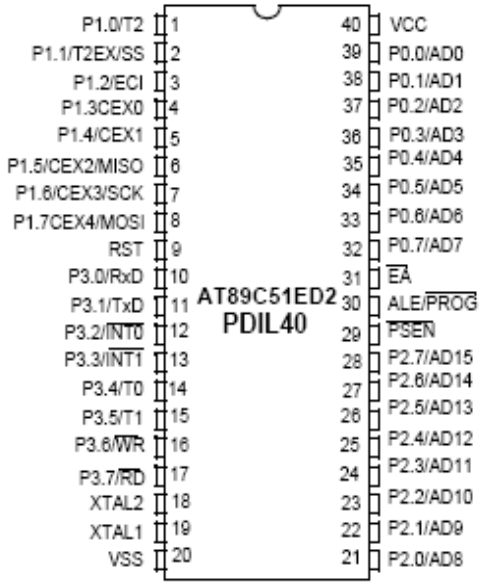
2.1.2 คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ตระกูล MCS-51

เนื่องจากคุณสมบัติของไมโครคอนโทรลเลอร์แต่ละเบอร์นั้นมีความสามารถที่แตกต่างกันออกไปในรายละเอียดปลีกย่อย ดังนั้นจะขออ้างถึงเบอร์ AT89C51 ของบริษัท Atmel ซึ่งเป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต ซึ่งมีคุณสมบัติดังนี้

- มีหน่วยความจำโปรแกรมแบบแฟลซ (Flesh Memory) ขนาด 8 กิโลไบต์
- มีหน่วยความจำแบบ EEPROM ขนาด 2 กิโลไบต์ (kbyte)
- แหล่งจ่ายไฟกระแสตรงขนาด 5 โวลต์ (ทำงานในช่วง 4-6 โวลต์)
- ทำงานได้ด้วยสัญญาณนาฬิกาตั้งแต่ 0-24 เมกกะเฮิร์ต (MHz)
- มีหน่วยความจำข้อมูล (RAM) ขนาด 256 ไบต์ (byte)
- มีพอร์ต 32 พอร์ตอิสระสามารถเข้าถึงในระดับบิตได้
- มีไทเมอร์/เคาเตอร์ขนาด 16 บิต ทั้งหมด 3 ตัว
- รองรับการอินเตอร์รัปต์ได้ 8 แหล่ง
- สามารถสื่อสารข้อมูลแบบอนุกรมได้ด้วย UART Channel

2.1.3 การจัดขาของไมโครลคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์ในตระกูล MCS- 51 จะมีโครงสร้างของการจัดเรียงขาที่คล้าย ๆ กัน ได้แสดงไว้ดังรูปที่ 2.2 และมีส่วนประกอบของขาดังต่อไปนี้



รูปที่ 2.2 การจัดขาของ MCS-51

VCC ต่อไฟเลี้ยง

GND ต่อกราวด์

Port0 (P0.0-P0.7) เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต สามารถทำงานได้ทั้งสองหน้าที่ คือ เป็นพอร์ตอินพุตและเอาท์พุตทั่วไป และใช้เป็นพอร์ตสำหรับติดต่อกับหน่วยความจำภายนอก คือรับและส่งข้อมูลพร้อมทั้งกำหนดแอดเดรส (Address) ไบต์ต่ำ

Port1 (P1.0-P1.7) เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต มีการต่อความต้านทานพูลอัพ (pull-up resister) ไว้ภายใน ทำหน้าที่เป็นพอร์ตอินพุตและเอาท์พุตทั่วไป นอกจากนี้ยังใช้งานเป็นขาอินพุตเอาท์พุตของไทเมอร์ 2

Port2 (P2.0-P2.7) เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต มีการต่อความต้านทานพูลอัพ (pull-up resister) ไว้ภายใน สามารถทำงานได้สองหน้าที่คือเป็นพอร์ตอินและพุตเอาท์พุตทั่วไป และใช้เป็นพอร์ตสำหรับติดต่อกับหน่วยความจำภายนอกคือกำหนดแอดเดรสไบต์สูง

Port3 เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต มีการต่อความต้านทานพูลอัพ (pull-up resister) ไว้ภายใน ทำหน้าที่เป็นพอร์ตอินพุตและเอาท์พุตทั่วไป นอกจากนี้ยังใช้เป็นขาสัญญาณควบคุมการติดต่อกับหน่วยความจำการอินเตอร์รัปต์ และอื่น ๆ

RST เป็นขาอินพุตที่ใช้รับสัญญาณสำหรับรีเซตซีพียู โดยซีพียู (CPU) จะถูกรีเซต (Reset) เมื่อขานี้เป็นลอจิก “1” นาน 2 แมชินไซเคิล หรือ 24 ไซเคิลของสัญญาณนาฬิกา

ALE/PROG ทำหน้าที่เป็นขาเอาท์พุตเมื่อซีพียูต้องการติดต่อกับหน่วยความจำภายนอก คือจะทำการส่งสัญญาณพัลส์ (Pulse) ออกมาที่ขานี้เพื่อทำการแลกแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก และขานี้จะเป็นอินพุตเมื่อยู่ในระหว่างโปรแกรมแฟรช

PSEN เป็นขาเอาท์พุต ใช้ในการติดต่อกับหน่วยความจำภายนอก คือเมื่อซีพียูทำการประมวลผลกับหน่วยความจำโปรแกรมภายนอกขานี้จะแอคทีฟสองครั้งในแต่ละแมชินไซเคิล

EA/VPP เป็นขาอินพุตและต้องการลอจิก “0” เพื่อยอมให้ซีพียูสามารถเข้าถึงหน่วยความจำโปรแกรมภายนอกได้ ซึ่งอยู่ที่ตำแหน่ง 0000H ถึง FFFFH นอกจากนี้แล้วขานี้ยังใช้รับไฟ 12 โวลต์เพื่อใช้ในระหว่างที่ทำการโปรแกรมแฟลช

XTAL1 เป็นขาอินพุตของวงจรออสซิลเลเตอร์แอมปลิไฟเออร์ และยังเป็นขาอินพุตของวงจรกำเนิดสัญญาณนาฬิกาภายใน

XTAL2 เป็นขาเอาท์พุตของวงจรออสซิลเลเตอร์แอมปลิไฟเออร์

2.2 คลื่นอัลตร้าโซนิค (Ultrasonic) [2]

ระบบอัลตร้าโซนิค (Ultrasonic) หมายถึง คลื่นเสียงที่มีความถี่สูงเกินกว่าที่หูมนุษย์จะได้ยิน โดยทั่วไปแล้วหูของมนุษย์โดยเฉลี่ยจะได้ยินเสียงสูงถึงเพียงแค่ประมาณ 15 KHz เท่านั้น ดังนั้นโดยปกติแล้วคำว่าอัลตร้าโซนิคหมายถึงคลื่นเสียงที่มีความถี่สูงกว่า 20 KHz ขึ้นไป สาเหตุที่มีการนำเอาคลื่นย่านอัลตร้าโซนิคมาใช้ก็เพราะว่าเป็นคลื่นที่สามารถกำหนดการส่งคลื่นไปยังเป้าหมายที่ต้องการได้โดยเจาะจง เรื่องนี้เป็นคุณสมบัติของคลื่นอย่างหนึ่ง ยิ่งคลื่นมีความถี่สูงขึ้นความยาวคลื่นก็จะยิ่งสั้นลง ถ้าความยาวคลื่นยาวกว่าช่องเปิด (ที่ให้เสียงนั้นออกมา) ของตัวกำเนิดเสียงความถี่นั้นเช่น คลื่นความถี่ 300 Hz ในอากาศจะมีความยาวถึงประมาณ 1000 กิโลเมตร ซึ่งจะยาวกว่าช่องที่ให้คลื่นเสียงออกมาจากตัวกำเนิดเสียงโดยทั่วไปมากมายคลื่นจะหักเหที่ขอบด้านนอกของตัวกำเนิดเสียงทำให้เกิดการกระจายทิศทางคลื่นแต่ถ้าความถี่สูงขึ้นมาอยู่ในย่านอุลตร้าโซนิค อย่างเช่น 40 KHz จะมีความยาวคลื่นในอากาศเพียงประมาณ 8 มม. เท่านั้นซึ่งเล็กกว่ารูเปิดของตัวที่ให้กำเนิดเสียงความถี่นี้มากคลื่นเสียงจะไม่มีการเลี้ยวเบนที่ขอบจึงพุ่งออกมาเป็นลำแคบ ๆ หรือที่เราเรียกว่ามีทิศทาง

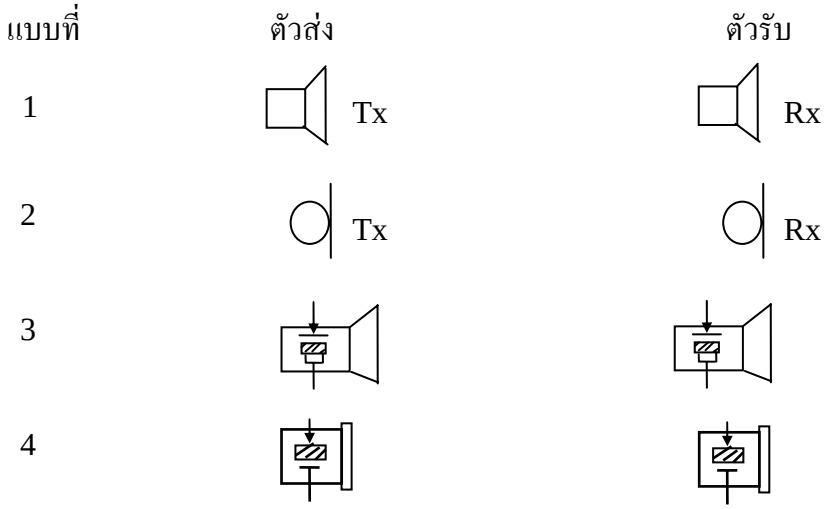
การมีทิศทางของคลื่นเสียงย่านอัลตร้าโซนิคทำให้เรานำไปใช้งานได้หลายอย่าง เช่น นำไปใช้ในเครื่องควบคุมระยะไกล (Ultrasonic remote control) โดยให้น้ำสั่นที่ความถี่สูง เครื่องวัดความหนาของวัตถุโดยสังเกตระยะเวลาที่คลื่นสะท้อนกลับมา เครื่องวัดความลึกและทำแผนที่ใต้ท้องทะเล เป็นต้น โดยความถี่ที่ใช้ขึ้นอยู่กับการใช้งาน เช่น คลื่นเสียงต้องเดินทางผ่านอากาศแล้ว ความถี่ที่ใช้ก็มักจะจำกัดอยู่เพียงไม่เกิน 50 KHz เพราะที่ความถี่สูงขึ้นกว่านี้อากาศจะดูดกลืนคลื่นเสียงเพิ่มขึ้นมาก ทำให้ระดับความแรงของคลื่นเสียงที่ระยะห่างออกไปลดลงอย่างรวดเร็ว ส่วนการใช้งานด้านการแพทย์ซึ่งต้องการรัศมีทำการสั้น ๆ ก็อาจใช้ความถี่ในช่วง 1 MHz ถึง 10 MHz ขณะที่ความถี่เป็น GHz (10^9 Hz) ก็มีใช้กันในหลายๆการใช้งานที่ตัวกลางที่คลื่นเสียงเดินทางผ่านไม่ใช่อากาศอุปกรณ์ที่สามารถแปลงพลังงานในรูปอื่นให้มาเป็นพลังงานทางกล โดยการสั่นไปมาซึ่งทำให้เกิดคลื่นเสียงย่านอุลตร้าโซนิคกระจายไปในอากาศได้หรือแปลงพลังงานทางกลให้มาเป็นพลังงานในรูปอื่นได้นั้น มีชื่อเรียกว่า อัลตร้าโซนิคทรานสดิวเซอร์ (Ultrasonic Transducer) ในปัจจุบันอุลตร้าโซนิคทรานสดิวเซอร์มีหลายแบบขึ้นอยู่กับหลักการที่ใช้ แบบที่นิยมใช้กันมากได้แก่

- แบบเปียโซอิเลคทริก (Piezo-electric Transducer) ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้าและ พลังงานทางกล โดยมีความถี่เรโซแนนซ์คงที่อยู่ค่าหนึ่ง

- แบบแมกนีโตสตริคทีฟ (Magnetostrictive Transducer) ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้าในขดลวดกับตำแหน่งความยาวของแกนเหล็กที่สวมขดลวดนั้นอยู่

- แบบอิเลคโตรสตริคทีฟ (Electrostrictive Transducer) ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้ากับพลังงานทางกล

โดยทั่วไป ลักษณะปกติของตัวส่ง Ultrasonic จะเปรียบเสมือนลำโพงส่งสัญญาณเสียง และตัวรับจะมีลักษณะเหมือน Microphone ดังนั้นสัญลักษณ์ที่ใช้จึงมีลักษณะคล้าย ๆ กัน ซึ่งมีหลายแบบ คือ



รูปที่2.3 สัญลักษณ์ของ Ultrasonic

ตัวส่ง คือ อัลตร้าโซนิคที่ถูกออกแบบเจาะจงมาให้แปลงสัญญาณไฟฟ้าที่ให้แก่ตัวมัน ให้ออกมาเป็นคลื่นเสียงย่านอัลตร้าโซนิค หน้าที่ของตัวส่งจึงคล้าย ๆ กับเป็นลำโพง

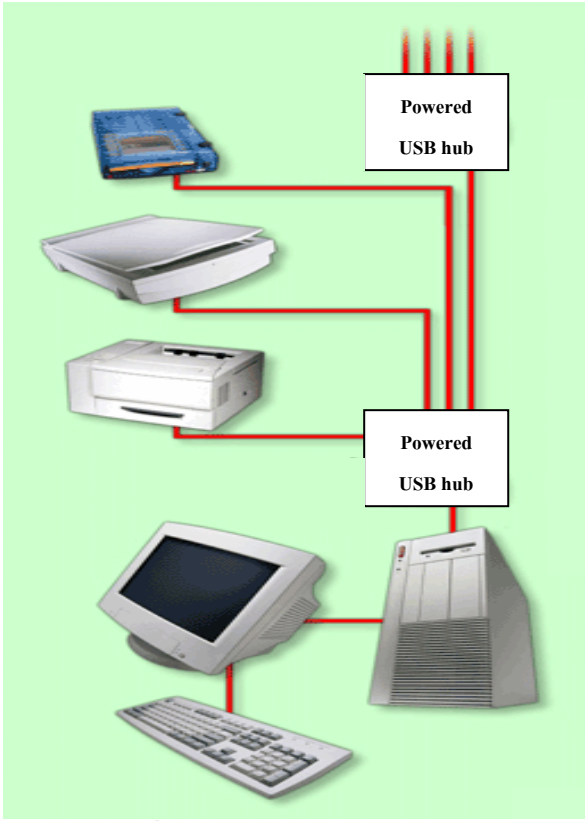
ตัวรับ คือ อัลตร้าโซนิคที่ถูกออกแบบเจาะจงมาให้แปลงคลื่นเสียงย่านอัลตร้าโซนิคที่มาตกกระทบตัวมันให้ออกมาเป็นสัญญาณไฟฟ้า หน้าที่ของตัวรับจึงคล้าย ๆ กับเป็นไมโครโฟน ด้วยเหตุนี้เวลาเขียนสัญลักษณ์ของอัลตร้าโซนิคจึงนิยมเขียนตามหน้าที่ของมันคือถ้าเป็นตัวส่งก็เขียนสัญลัษณ์เป็นลำโพง ถ้าเป็นตัวรับก็เขียนสัญลักษณ์เป็นไมโครโฟน

2.3 USB (Universal Serial Bus USB) [3]

การสื่อสารระหว่างคอมพิวเตอร์และอุปกรณ์ต่อพ่วงต่างๆจำเป็นที่จะต้องมี พอร์ตI/O เพื่อใช้ในการสื่อสาร (I/O = input , output) ในสมัยก่อน พอร์ตที่รู้จักกันดี ก็คือ พอร์ตอนุกรม (Serial Port)หรือเรียกกันว่า พอร์ตคอม (Com Port) และ พอร์ตขนาน (Parallel Port) แต่ข้อจำกัดของพอร์ต เหล่านี้นั้นมีมากพอสมควรไม่ว่าจะเป็นเรื่องของความเร็ว จำนวนอุปกรณ์จะนำมาต่อได้ และคุณสมบัติทางด้าน Plug & Play USB ก็เป็น พอร์ต I/O ที่ทำงานในลักษณะซีเรียลบัส (Serial Bus) เช่นกัน สามารถต่อได้กับอุปกรณ์ถึง 127 อุปกรณ์ ต่อ 1 พอร์ต และ สนับสนุนการทำงานแบบ Plug & Play อย่างเต็มรูปแบบ โดยจะทำการค้นหา และตั้งค่าของอุปกรณ์ที่นำมาต่อให้อย่างอัตโนมัติ ไม่จำเป็นต้องกำหนดค่า (assign) IRQ เพียงแต่ใส่แผ่นไดร์เวอร์ (Driver) ของอุปกรณ์ที่นำมาติดตั้ง และส่วนมากไม่จำเป็นต้องปิดแล้วเปิดเครื่องใหม่ (Restart) หลังจากติดตั้งไดร์เวอร์ด้วย ถ้าหากอุปกรณ์นั้นใช้กำลังไฟฟ้าไม่มากพอร์ต USB ก็จ่ายไฟเลี้ยงให้กับอุปกรณ์ต่างๆ ได้ด้วย

2.3.1 ระบบการจัดส่งข้อมูล USB

สำหรับ USB เราเรียกระบบบัสเช่นนี้ว่า การติดต่อแบบ "Host/Slave" หมายถึงตัวเครื่องคอมพิวเตอร์จะเป็นผู้จัดการ การจัดส่งข้อมูลทั้งหมด และอุปกรณ์ที่นำมาต่อพ่วงเพียงแต่พร้อมที่จะรับส่งข้อมูลเท่านั้น โดยตัว USB Host Controller หรือตัวควบคุมของ USB นั้น จะรวมอยู่ในตัวชิพเซต (Chipset) บนเมนบอร์ดแต่ถ้าหากเป็นชิพเซตรุ่นเก่าๆก็จะยังไม่มีตัวควบคุม USB ในการจัดส่งข้อมูลนั้น ภายในสาย USB จะมีสายภายในทั้งหมด 4 เส้น ในจำนวนนี้ 2 เส้นใช้เพื่อเป็นสายในการส่งข้อมูล อีก 2 เส้นใช้สำหรับจ่ายไฟเลี้ยง แรงดันบวก 5 โวลต์ และ กราวด์ ข้อมูลที่ถูกส่งผ่าน USB นั้นจะต่างไปจากพอร์ตขนาน และพอร์ตอนุกรมซึ่งทั้ง 2 ชนิดนี้นั้น จะส่งข้อมูลเป็นบิตเดี่ยวๆ แต่ USB จะส่งเป็นชุดข้อมูล อย่างเช่น ถ้าหากต้องการ เก็บข้อมูลไปยัง USB Zip drive PC จะทำการแบ่งข้อมูลออก มา 64 ไบต์ แล้วใส่ข้อมูลของแอดเดรสเข้าไปแล้วส่งไปยังพอร์ต USB จากนั้นก็จะทำเช่นนี้จนกระทั่งส่งข้อมูลครบสิ่งที่ทำให้ USB ส่งข้อมูลได้เร็วถึง 12 เมกกะบิตต่อวินาที (Mbits/s) นั้นอยู่ที่แม่นยำของการส่ง เพราะ สายที่นำมาใช้สำหรับ USB นั้น มีระดับสัญญาณรบกวน และ ความเพี้ยนของรูปสัญญาณนั้นน้อยกว่า ทั้งพอร์ตขนาน และพอร์ตอนุกรม และการส่งสัญญาณแบบ isochronous data delivery หมายถึง อุปกรณ์แต่ละชิ้นบนระบบบัสนั้น จะถูกจำกัดขนาดของแบนด์วิดธ์ (bandwidth) ให้แน่นอน อย่างไรก็ตามแบนด์วิดธ์โดยรวมแล้วก็ไม่เกิน 12 เมกกะบิตต่อวินาที นี่เป็นเหตุผลที่ทำให้ USB สามารถต่ออุปกรณ์ได้มากที่สุด 127 อุปกรณ์



รูปที่ 2.4 การเชื่อมต่อพอร์ต USB

จากรูปที่ 2.3 เป็นการต่ออุปกรณ์ต่อพ่วงออกจากเครื่องคอมพิวเตอร์ไปยังอุปกรณ์ต่าง ๆ เช่น จากพอร์ต USB แรกไปยัง Powered USB Hub แล้วต่อไปยังอุปกรณ์อื่น ๆ รวมถึงต่อไปยัง Powered USB Hub เพื่อต่ออุปกรณ์เพิ่มเข้าอีกได้เรื่อย ๆ ในลักษณะของการต่อแบบอนุกรม ส่วนพอร์ต USB ชุดที่ 2 หลังเครื่องนั้นสามารถต่อไปยังจอมอนิเตอร์ (Monitor) ได้โดยอิสระต่อกัน จะสังเกตได้ว่าถ้าหากการต่อมี USB Hub เข้ามาต่อพ่วงด้วยที่ตัว Hub จะต้องมีตัวจ่ายไฟด้วยเพราะลำพังไฟเลี้ยงจากเครื่องคอมพิวเตอร์เองคงไม่พอส่งไปเลี้ยงอุปกรณ์ต่าง ๆ

ดังนั้นเราอาจมองได้ว่าสิ่งที่จะต่อจากพอร์ต USB นั้นมี 2 ชนิด ก็คือ ต่อเข้าได้กับ Hubs หรือต่อกับ อุปกรณ์ Hubs นั้นมีหน้าที่ ทำให้ สามารถต่ออุปกรณ์ได้มากขึ้นบนระบบบัส บางครั้ง Hubs จะเข้าไป รวมกับอุปกรณ์ อย่างเช่น มอนิเตอร์ ที่มี USB Hubs ในตัว โดยจะ สามารถปรับค่าต่างๆของมอนิเตอร์ผ่านระบบปฏิบัติการ และ นอกจากนั้นมีหน้าที่เป็น Hubs ด้วย

2.4 สรุป

เครื่องวัดปริมาตรน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิค มีองค์ประกอบที่สำคัญคือ เซนเซอร์โดยอาศัยคลื่นอัลตร้าโซนิค (Ultrasonic Sensor) ไมโครลคอนโทรลเลอร์ เบอร์ MCS-51 และระบบการจัดส่งข้อมูลผ่าน USB

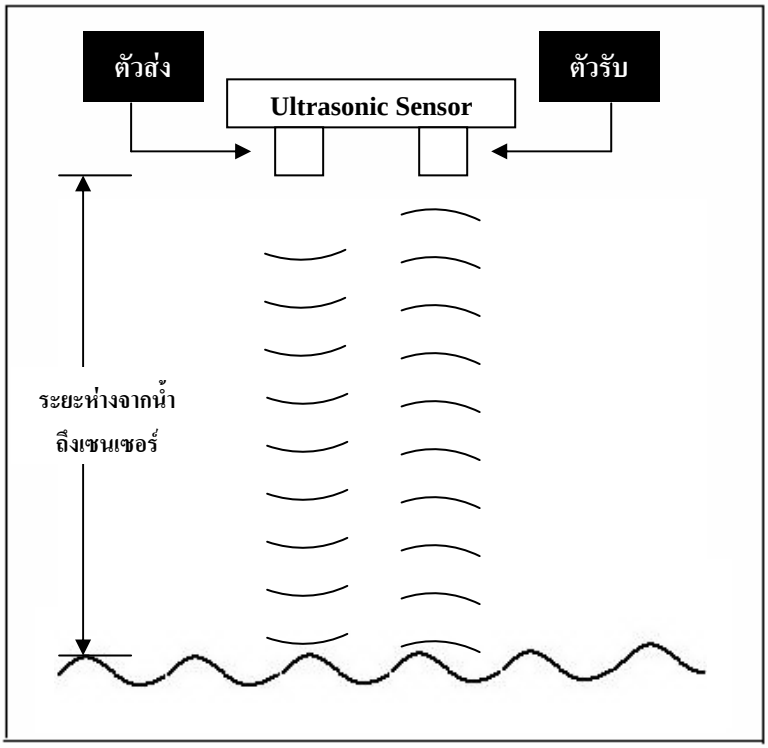
บทที่ 3
การออกแบบและสร้างวงจรสร้างและรับคลื่นอัลตร้าโซนิคในการตรวจวัดระดับน้ำมัน

3.1 กล่าวนำ

เนื้อหาในบทนี้จะกล่าวถึงการออกแบบและการสร้างวงจรสร้างและรับคลื่นอัลตร้าโซนิคในการตรวจวัดระดับน้ำมันในถัง แล้วทำการทดสอบการใช้งานได้จริงของอุปกรณ์แต่ละตัวที่นำมาประกอบเป็นเครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิค

อุปกรณ์โดยรวมระบบของเครื่องวัดระน้ำมันในถังโดยอาศัยคลื่นอัลตร้าโซนิค มีอยู่ทั้งหมด 5 ส่วนด้วยกัน คือ เซ็นเซอร์วัดระยะทาง, วงจรไมโครคอนโทรลเลอร์, วงจรเชื่อมต่อพอร์ต USB, จอแสดงผลLCD, KEYPAD

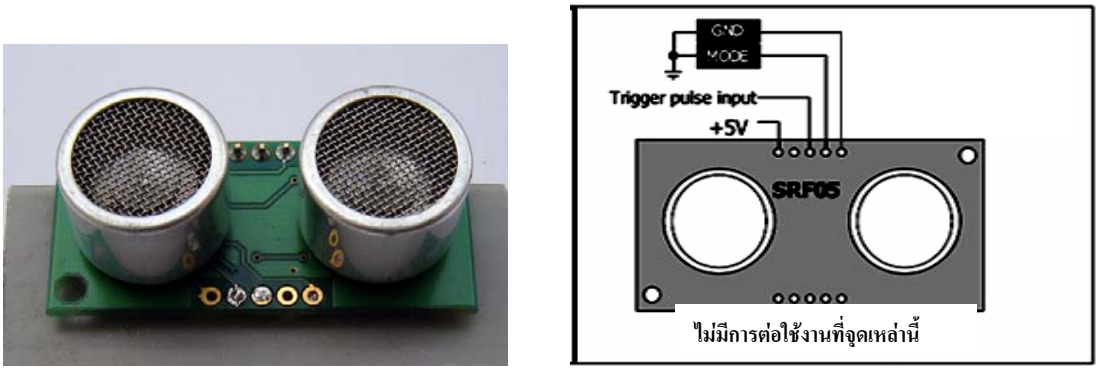
3.2 เซ็นเซอร์วัดระยะทาง Ultrasonic Distance Detector Module รุ่นSRF 05



รูปที่ 3.1 แสดงหลักการตรวจจับวัตถุโดยใช้สัญญาณความถี่เหนือเสียงหรืออัลตร้าโซนิก

เครื่องตรวจจับและวัดระยะทางด้วยอัลตร้าโซนิก (Ultrasonic Distance Detector Module) เป็นแผงวงจรวัดตรวจจับและวัดระยะทางด้วยอัลตร้าโซนิกที่มีแม่นยำและเที่ยงตรงสูงโดยสามารถวัดระยะทางได้ตั้งแต่ 1 เซนติเมตรไปจนถึง 4 เมตร โดยเครื่องตรวจจับและวัดระยะทางด้วยอัลตร้า-โซนิก (Ultrasonic Distance Detector Module) ถูกออกแบบมาให้ใช้งานกับไมโครคอนโทรลเลอร์ได้ง่ายโดยใช้ขาเชื่อมต่อเพียง 1 หรือ 2 ขา โดยจะทำการส่งสัญญาญคลื่นอัลตร้าโซนิกออกไปแล้ววัดระยะเวลาที่มีสัญญาณสะท้อนตอบกลับมา โดยเอาต์พุตที่ได้จะอยู่ในรูปของความกว้างพัลส์ซึ่งสัมพันธ์กับระยะทางของวัตถุที่ตรวจจับได้ โดยความถี่อัลตร้าโซนิกที่ถูกส่งออกไปในอากาศด้วยความเร็ว 1.125 ฟุตต่อมิลลิวินาที (ประมาณ 346 เมตรต่อวินาที) ดังนั้นเมื่อทราบความเร็วในการเคลื่อนที่ของคลื่น เวลา เริ่มส่งคลื่นและเวลาที่รับเสียงสะท้อนกลับมาจึงสามารถคำนวณหาค่าของระยะทางได้ ดังแสดงหลักการตรวจจับในรูปที่ 3.1

ระยะทางที่ได้นั้นจะต้องมีการคำนวณค่ากลับทางคณิตศาสตร์เมื่อใช้กับไมโครคอลโทรลเลอร์แล้วถือว่าเป็นเรื่องที่ยุ่งยากพอสมควร ดังนั้นเครื่องเครื่องตรวจจับและวัดระยะทางด้วยอัลตร้า-โซนิก (Ultrasonic Distance Detector Module) จึงประมวณผลค่าทางคณิตศาสตร์ต่างๆไว้เรียบร้อยแล้ว จากนั้นส่งผลลัพธ์ที่วัดได้ออกมาเป็นพัลส์ที่มีความกว้างสัมพันธ์กับระยะทางที่วัดได้ จะใช้สายสัญญาณเพียงเส้นเดียว จึงทำให้สะดวกมากในการนำมาเชื่อมต่อไมโครคอลโทรลเลอร์



(ก) **(ข)**

รูปที่ 3.2 **(ก)** แสดงเซ็นเซอร์วัดระยะทาง Ultrasonic Distance Detector Module รุ่นSRF 05
(ข) แสดงขาสัญญาณของ SRF05 และการกำหนดโหมดทำงาน

3.2.1 การต่อใช้งานของ SRF05

มีจุดต่อสำหรับใช้งานอยู่ทั้งหมด 5 จุดตามรูปที่ 2

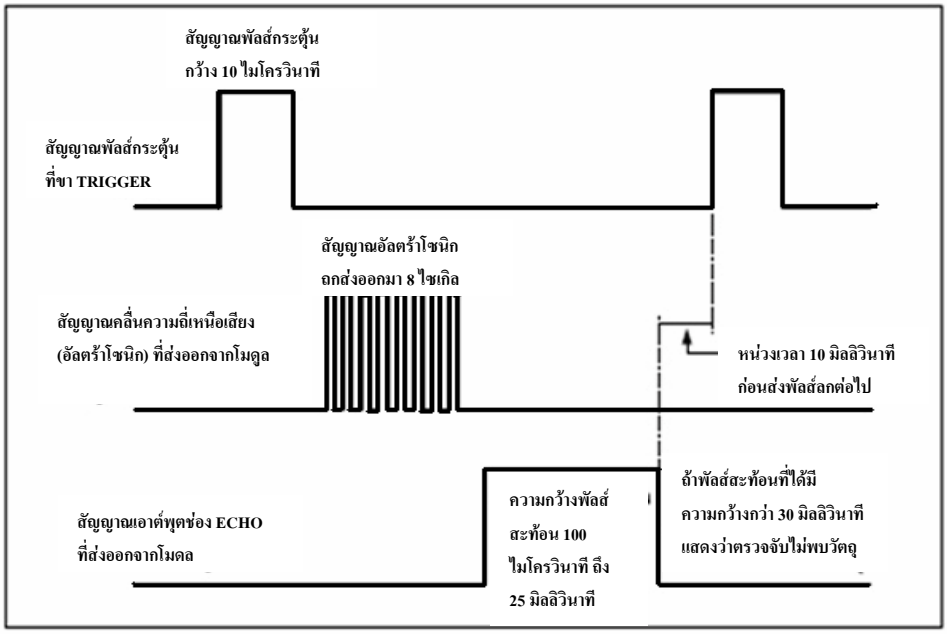
1. ขาไฟเลี้ยง (+5V) สำหรับต่อไฟเลี้ยงแรงดัน +5V

2. ขาEcho Pulse Output (ECHO) เป็นขาเอาต์พุตสำหรับส่งสัญญาณกัลส์ออกจาก SRF05 ซึ่งการใช้งานจะนำขานี้ไปต่อเข้ากับพอร์ตอินพุตของไมโครคอนโทรลเลอร์ เพื่อตรวจจับความกว้างของสัญญาณพัลส์ที่ส่งออกมาเพื่อแปลความหมายออกมาเป็นระยะทางอีกครั้งหนึ่ง

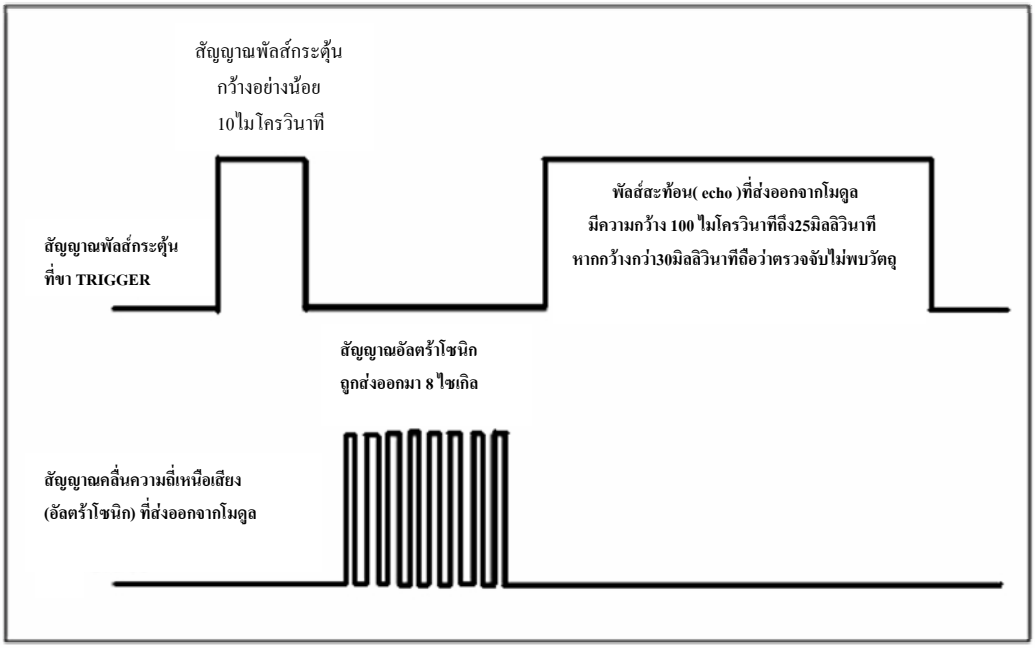
3. ขา Trigger Pulse Input (TRIGGER) เป็นขาอินพุตรับสัญญาณพัลส์ที่มีความกว้างอย่างน้อย 10 ไมโครวินาทีเพื่อกระตุ้นการสร้างคลื่นอัลตรโซนิกความถี่ 40 kHz ออกสู่อากาศจากตัวส่ง ดังนั้นเมื่อคลื่นความถี่ดังกล่าวนี้เคลื่อนที่กระทบสิ่งกีดขวางที่อยู่เบื้องหน้าก็จะเกิดการสะท้อนกลับเข้ามายังตัวรับและถูกแปลงออกมาเป็นความกว้างของสัญญาณพัลส์ที่จะส่งออกไปทางเขา Echo Pulse Output นอกจากนี้ในโหมด 1 สัญญาณ จะใช้จุดนี้เป็นจุดสื่อสารข้อมูลอนุกรมเพื่อรับส่งค่าการวัดกับไมโครคอนโทรลเลอร์

4. ขา MODE สำหรับเลือกรูปแบบการติดต่อกับ SRF05ปล่อยลอยไว้ (NC): เลือกใช้ติดต่อแบบ 2 สัญญาณ ผ่านจุดต่อถ ECHO และ TRIGGERต่อลงกราวด์: เลือกให้ติดต่อแบบ 1 สัญญาณ ผ่านจุดต่อ TRIGGER

5. ขา GND สำหรับต่อกราวด์



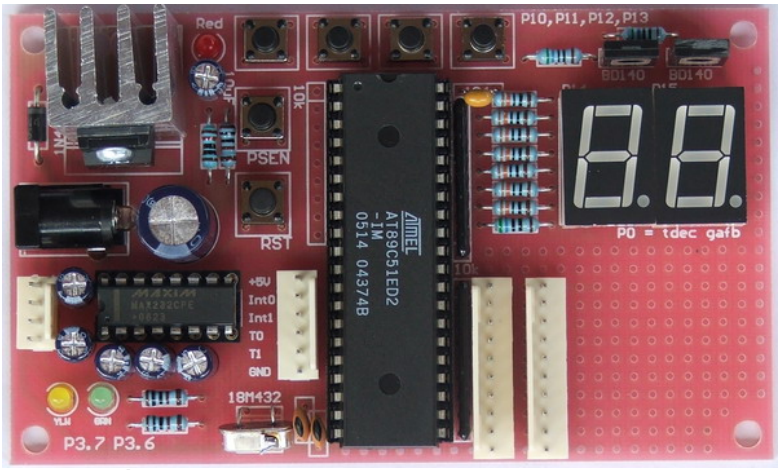
รูปที่ 3.3 ไดอะแกรมเวลาแสดงสัญญาณที่ส่งไปยังเครื่องตรวจจับและวัดระยะทางด้วยอัลตร้าโซนิก (Ultrasonic Distance Detector Module) และสัญญาณที่ตอบรับกลับมาจากเครื่องตรวจจับและวัดระยะทางด้วยอัลตร้าโซนิก (Ultrasonic Distance Detector Module) ในกรณีติดต่อแบบ 2 สัญญาณ



รูปที่ 3.4 ไดอะแกรมเวลาแสดงสัญญาณที่ส่งไปยังเครื่องตรวจจับและวัดระยะทางด้วยอัลตร้าโซนิก (Ultrasonic Distance Detector Module) และสัญญาณที่ตอบรับกลับมาจากเครื่องตรวจจับและวัดระยะทางด้วยอัลตร้าโซนิก (Ultrasonic Distance Detector Module)

3.3 วงจรไมโครคอนโทรเลอร์

ได้เลือกใช้วงจรไมโครคอนโทรเลอร์สำเร็จรูป ในตระกูล MCS-51 ที่ใช้ไอซีเบอร์ AT89C51ED2 ของ ATMEL ดังที่ได้แสดงวงจรรวมไว้ดังรูปที่ 3.5

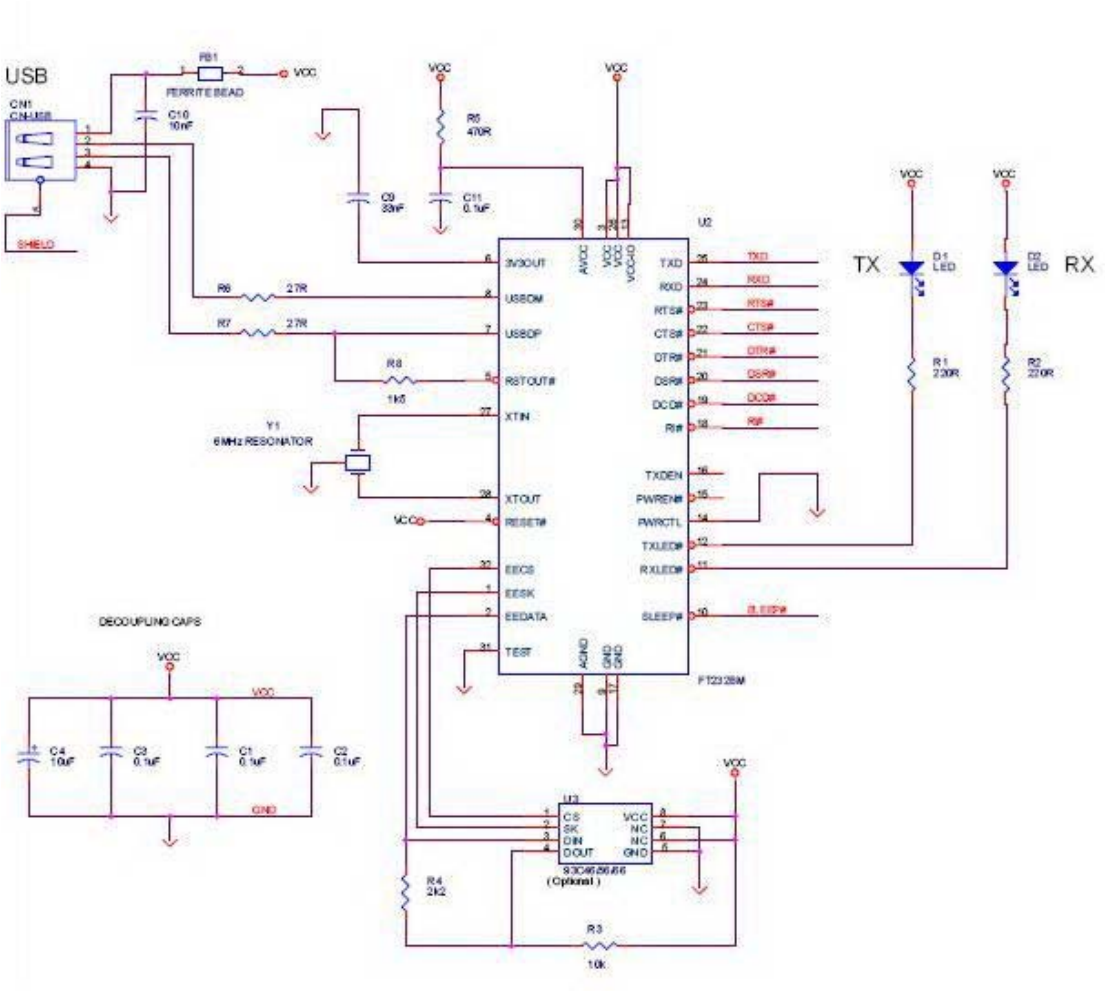


รูปที่ 3.5 ชุดวงจรไมโครคอนโทรลเลอร์ AT89C51ED2

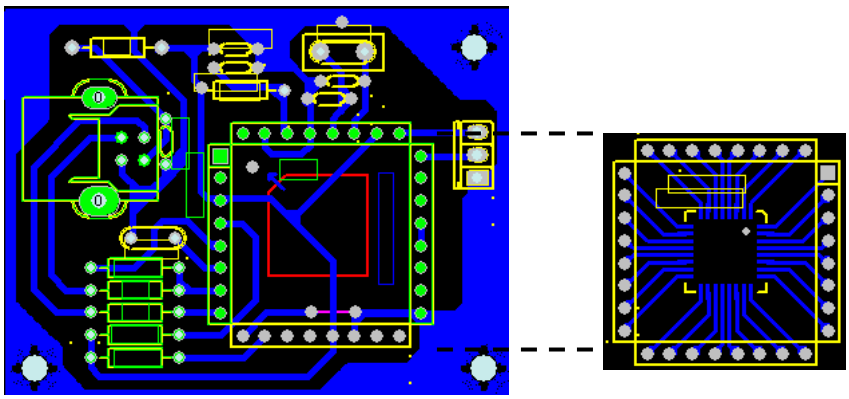
3.4 วงจรเชื่อมต่อ USB

3.4.1 การออกแบบลายวงจรเชื่อมต่อ USB ลงบนแผ่นวงจรพิมพ์

จากรูปที่ 3.6 ได้แสดงถึงวงจรเชื่อมต่อ USB ที่ได้ทำการออกแบบไว้โดยเลือกใช้ไอซีที่มีคุณสมบัติตามต้องการ จากนั้นทำการออกแบบลายวงจรด้วยโปรแกรม Protel 99SE เพื่อที่ได้ลายวงจรเป็นแผ่นลายทองแดงที่จะสามารถนำมาใช้งานได้ตามรูปที่ 3.7



รูปที่ 3.6 วงจรเชื่อมต่อ USB

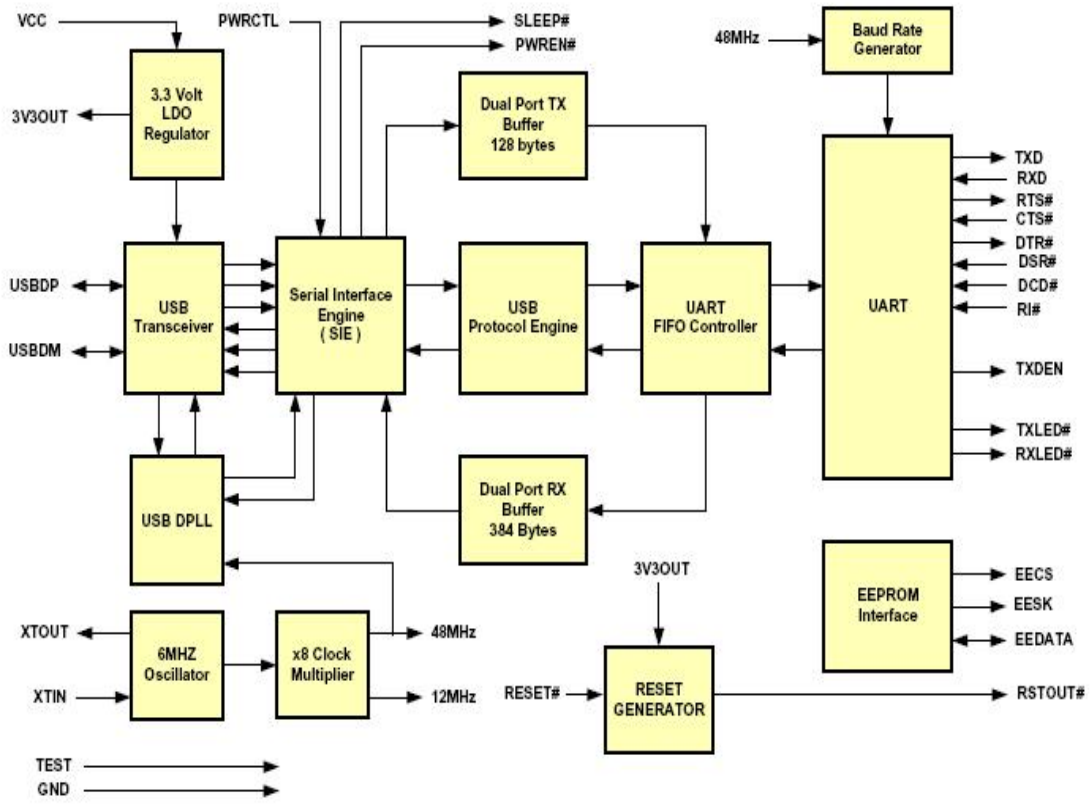


รูปที่ 3.7 ลายวงจรที่ได้จากโปรแกรม Protel99SE บนแผ่นวงจรพิมพ์

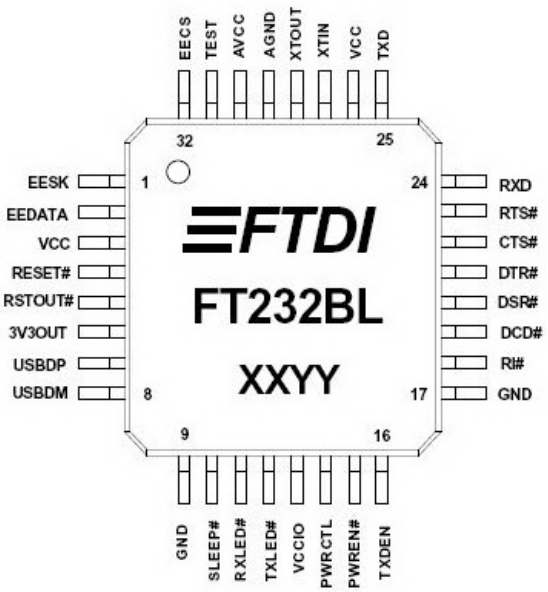
3.4.2 การเลือกใช้ไอซีในการออกแบบ

ในวงจรเชื่อมต่อ USB ได้เลือกใช้ไอซีจำนวน 1 ตัว คือ FT232BL

FT232BL มีคุณสมบัติเป็นตัวเชื่อมต่อระหว่างอุปกรณ์ที่ส่งสัญญาณแบบอนุกรม ให้สามารถเชื่อมต่อกับคอมพิวเตอร์โดยผ่านทางพอร์ต USB ได้ และทำหน้าที่แทน พอร์ต RS-232 ได้ ซึ่งมีแผนผังการทำงานดังรูปที่ 3.8 และมีการจัดเรียงขาของไอซี ดังรูปที่ 3.9



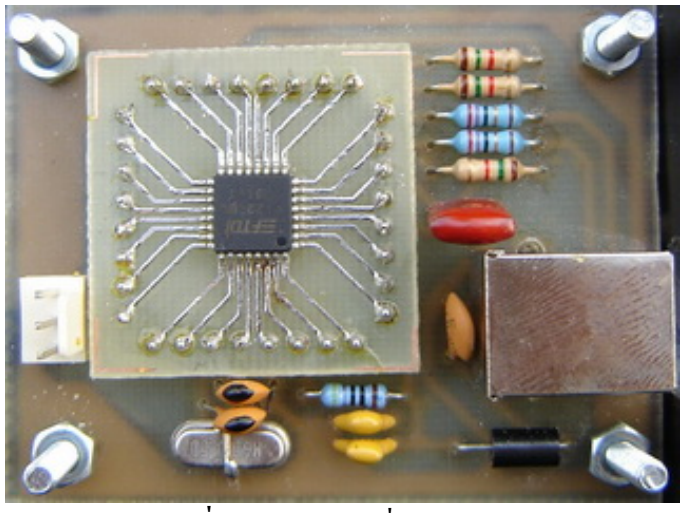
รูปที่ 3.8 USB-to-Serial (Single Channel) Controller Block Diagram



รูปที่ 3.9 Top View ของ FT232BL

3.4.3 การติดตั้งอุปกรณ์อิเล็คทรอนิคส์

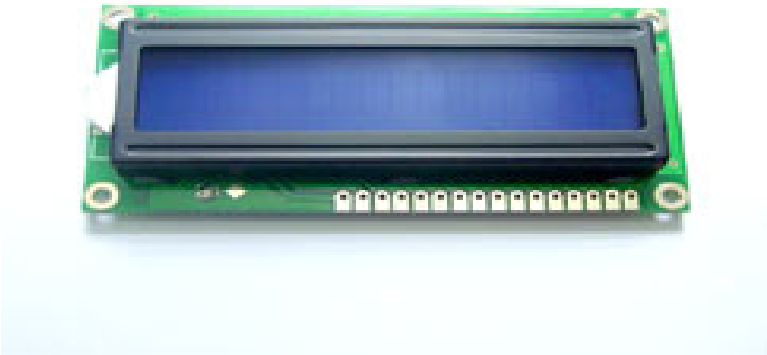
นำอุปกรณ์ที่ได้ออกแบบและจัดเตรียมมาประกอบลงบนแผ่นปริ้นท์ได้ดังรูปที่ 3.11



รูปที่ 3.10 ชุดวงจรเชื่อมต่อ USB

3.5 จอแสดงผลLCD

ได้เลือกใช้จอแสดงผล LCD รุ่น LCD 16 CHARS x1 LINE,AV DISPLAY โดยรุ่นที่นำมาใช้จะแสดงผล 1บรรทัดและ แสดง 16 ตัวอักษร ซึ่งเพียงพอต่อการนำมาใช้เป็นอุปกรณ์สำหรับแสดงผลต่างๆ แต่ละค่าในโครงการนี้



รูปที่ 3.11 จอแสดงผล LCD รุ่น LCD 16 CHARS x1 LINE,AV DISPLAY

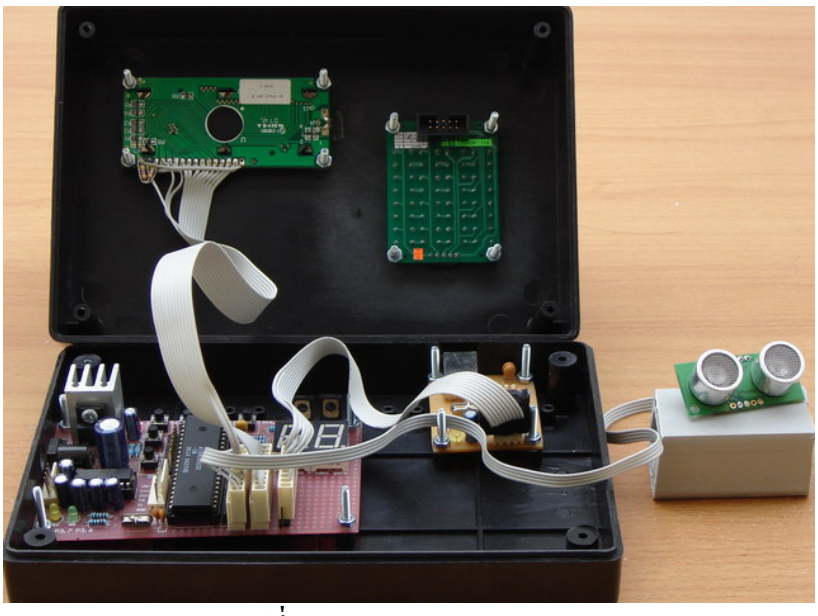
3.6 KEYPAD

กลุ่มผู้ทดลองได้เลือกใช้ KEYPAD Module ET-MINIKEY 4x4 ซึ่งเป็นชุด Matrix 4x4 โดยจะมี PIN ใช้ต่อกับไมโครคอนโทรลเลอร์ทั้งหมด 8 PIN โดย PIN1-PIN4 (Colum) จะใช้ต่อเป็น Input เพื่อให้ MCU อ่านค่ารหัสของคีย์ที่ถูกกด ส่วน PIN5-PIN8 (ROW) ใช้ต่อเป็น Output เพื่อให้ MCU ส่งลอจิก “0” มาทำการสแกนคีย์ในแต่ละแถว การทำงานนั้น MCU จะอ่านสถานะทางลอจิกของคีย์แต่ละหลักเข้ามาทาง PIN1-PIN4 ซึ่งถ้าไม่มีการกดคียจะอ่านลอจิกได้ “1” ถ้ามีการกดคีย์ลอจิกที่อ่านได้ในหลักนั้นจะเป็น “0” แต่ก่อนที่จะอ่านค่าลอจิกแต่ละหลัก MCU จะต้องให้ลอจิก “0” แก่แถวของคีย์แต่ละแถว (PIN5-PIN8) ในการอ่านลอจิกเข้ามาแต่ละครั้งเสมอ

รูปที่ 3.12 ชุดปุ่มกด KEYPAD-Module ET-MINIKEY Matrix 4x4

3.7 สรุปเรื่องการออกแบบและสร้างวงจรสร้างและรับคลื่นอัลตร้าโซนิคในการตรวจวัดระดับน้ำมันในถังน้ำมัน

เมื่อทำการประกอบอุปกรณ์แต่ละอุปกรณ์เข้าด้วยกัน จากนั้นได้ทำการบรรจุลงในกล่องดังรูปที่ 3.14 แสดงรูปภายในกล่อง และรูปที่ 3.15 แสดงภายนอกกล่องพร้อมใช้งาน เพื่อแสดงความเป็นระเบียบและสะดวกต่อการนำไปใช้งานจริง และได้ทำการทดสอบโดยรวมขององค์ประกอบย่อยของแต่ละอุปกรณ์ เพื่อที่อุปกรณ์ต่างๆจะทำงานได้อย่างสมบรูณ์



รูปที่3.13 แสดงรูปภายในกล่อง

รูปที่3.14 แสดงภายนอกกล่องพร้อมใช้งาน

บทที่ 4
การเขียนโปรแกรมและการทดลองเครื่องวัดระดับน้ำมัน
โดยอาศัยคลื่นอัลตร้าโซนิค

4.1 กล่าวนำ

ในบทนี้อุปกรณ์ที่ได้ออกแบบและสร้างขึ้นมาประกอบเข้าเป็นเครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลต้าโซนิค เพื่อทำการวัดระดับน้ำมันภายในถังที่มีรูปร่างและความจุที่แตกต่างกัน ได้แก่ ถังรูปทรงสี่เหลี่ยม ความจุ 10 ลิตร หรือถังรูปทรงกระบอกความจุ 7 ลิตร แสดงดังรูปที่ 4.1

(ก)

(ข)

รูปที่ 4.1 (ก) ถังน้ำมันรูปทรงสี่เหลี่ยม
ขนาด 140X438X194 มิลลิเมตร
(ข) ถังรูปทรงทรงกระบอก
ขนาด รัศมี80 มิลลิเมตร X สูง185 มิลลิเมตร

4.2 การใช้โปรแกรม FLIP V2.2.4 และATMEL Microcontroller

ระบบของโปรแกรม FLIP (Flexible In-system Programmer) เป็นโปรแกรมสำหรับพัฒนาไมโครคอนโทรลเลอร์ของATMELโดยสามารถใช้สนับสนุนการพัฒนาโปรแกรมไมโครคอนโทรลเลอร์ในตระกูล MCS51ในกลุ่มที่ใช้การพัฒนาแบบISPซึ่งได้แก่เบอร์ AT89C51RD2 /AT89C51ED2และAT89C51AC2 โดยโปรแกรมทำงานภายใต้ระบบปฏิบัติการของ Windows9X/Me/NT/2000 และ Windows XP โดยสนับสนุนการเชื่อมต่อกับระบบฮาร์ดแวร์ที่ใช้การเชื่อมต่อแบบ RS232 หรือ CAN หรือ USB ซึ่งวิธีการเชื่อมต่อของโปรแกรม FLIP กับระบบฮาร์ดแวร์ของไมโครคอนโทรลเลอร์นั้น จะขึ้นอยู่กับความสามารถของตัวไมโครคอนโทรลเลอร์ ที่จะนำมาทำการพัฒนาว่าสามารถใช้การติดต่อสื่อสารด้วยวิธีใดได้บ้าง แต่สำหรับชุดไมโครคอนโทรลเลอร์อัน

ได้แก่ T89C51RD2/AT89C51RD2/AT89C51ED2 และAT89C51AC2 นั้นจะสามารถใช้การเชื่อมต่อผ่านทางพอร์ตอนุกรม RS232เท่านั้น ไม่สามารถเชื่อมต่อผ่านระบบการสื่อสารของ CANหรือ USB ได้โดยโปรแกรม FLIP จะใช้สำหรับดาวน์โหลด(download) ข้อมูลให้หน่วยความจำของไมโครคอนโทรลเลอร์ที่ทำงานในมอนิเตอร์โหมด (Monitor Mode) เพื่อให้ผู้ใช้สั่งจัดการกับหน่วยความจำภายในตัว CPU ไม่ว่าจะเป็นการ สั่งลบข้อมูล (Erase) สั่งตรวจสอบข้อมูลในหน่วยความจำ (Blank Check) สั่งโปรแกรมข้อมูลให้กับหน่วยความจำโปรแกรมของCPU (Program)โดยโปรแกรมสั่งเปรียบเทียบข้อมูลจากบัพเพอร์ (Buffer) กับหน่วยความจำในตัว CPU (Verify) หรือสั่งอ่านข้อมูลจากหน่วยจำลองหน่วยจำของCPU (Read) เป็นต้น แต่เนื่องจากบอร์ดไมโครคอนโทรลเลอร์ตระกูล MCS51 ของบริษัท อีทีที จำกัด ในปัจจุบันนั้น จะมีอยู่ด้วยกันมากมายหลายรุ่น และ แต่ละรุ่นก็สามารถเลือกติดตั้งใช้งานกับ MCU ของATMELได้หลายเบอร์เนื่องจากมีโครงสร้างทางฮาร์ดแวร์และกาจัดเรียงขาสัญญาณที่เหมือนกัน ดังนั้นในขั้นตอนของการ Download Program ให้กับ MCU นั้น ผู้ใช้เองต้องเลือกกำหนดเบอร์ของMCU ให้กับโปรแกรม FLIP ให้ถูกต้องกับเบอร์ของ MCU ที่ติดตั้งใช้งานอยู่ภายในบอร์ดด้วยเพราะถ้าเลือกกำหนดเบอร์ไม่ถูกต้องจะทำให้การ Download Program เกิความผิดพลาดขึ้นได้โดยทาง ATMEL เองก็ได้มีการผลิต MCUซึ่งมีรหัสเบอร์ใกล้เคียงกันแต่มีโครงสร้างในการ Download Program ที่แตกต่างกัน ซึ่งทำให้ผู้ใช้มักเกิดความสับสนและเข้าใจผิดในการเลือกกำหนดเบอร์ของ MCU เพื่อสั่ง Download Program อยู่บ่อยๆ ตัวอย่างดังเช่น T89C51RD2 และ AT89C51RD2 ซึ่งมีรหัสเบอร์ที่ใกล้เคียงกันมากแต่เป็น MCU ที่มีความแตกต่างกันทางด้าโครงสร้างของความจำและวิธีการDownload Program ที่แตกต่างกันซึ่งถ้าหากว่าในบอร์ดได้ทำการติดตั้ง MCU เบอร์ T89C51RD2 ไว้ แต่ผู้ใช้เลือกกำหนดเบอร์ MCU ให้กับโปรแกรม FLIP เป็น AT89C51RD2 ก็จะทำให้ไม่สามารดาวน์โหลดโปรมแกรมให้กับMCU ได้อย่างถูกต้อง ดังนี้เป็นต้น ซึ่งในที่นี้จึงขออธิบายถึงจุดสังเกต ในการดูเบอร์ของ MCU ของATMEL เพื่อเป็นแนวทางให้ผู้ใช้ในการเลือกกำหนดเบอร์ของ MCU ให้กับโปรแกรมFLIPดังนี้

- T89C51RD2 เป็น CPU ของ ATMEL Wireless ซึ่ง MCU เบอร์นี้จะผลิตขึ้นในนามของ ATMEL Wireless โดยจะสามารถสังเกตเห็นลักษณะการพิมพ์เบอร์ที่ตัวถังของ MCU โดยมีรหัส ATMEL และมีรหัสเบอร์เป็น 89C51RD2-CM โดยจะไม่มีตัวหนังสือ AT นำหน้าตัวเลขเบอร์ของ 89C51RD2-CM ด้วย ซึ่งถ้าติดตั้ง MCUรุ่นนี้ให้กับบอร์ดไว้ในขั้นตอนของการเลือกเบอร์ MCU จะต้องเลือกเป็น T89C51RD2 เท่านั้นถ้าเลือกเป็น AT89C51RD2 จะไม่ดาวน์โหลดโปรแกรมได้อย่างถูกต้อง

- AT89C51RD2 เป็น CPU ของ ATMEL โดยจะสามารถสังเกตเห็นลักษณะการพิมพ์เบอร์ที่ตัวถังของ MCU โดยมีรหัส ATMEL และมีรหัสเบอร์เป็น AT89C51RD2-IMซึ่งถ้าติดตั้ง MCU รุ่นนี้

ให้กับบอร์ดไว้ ในขั้นตอนของการเลือกเบอร์ MCU จะต้องเลือกเป็นAT89C51RD2 เท่านั้น ถ้าเลือกเป็น T89C51RD2 จะไม่สามารถดาวน์โหลดโปรแกรมได้อย่างถูกต้อง

- AT89C51ED2 เป็น CPU ของ ATMEL โดยจะสามารถสังเกตเห็นลักษณะการพิมพ์เบอร์ที่ตัวถังของ MCU โดยมีรหัส ATMEL และมีรหัสเบอร์เป็น AT89C51ED2-IM ซึ่งถ้าติดตั้ง MCU รุ่นนี้ให้กับบอร์ดไว้ ในขั้นตอนของการเลือกเบอร์ MCU จะต้องเลือกเป็น AT89C51ED2 เท่านั้น ถ้าเลือกเป็นเบอร์อื่นๆ จะไม่สามารถดาวน์โหลดโปรแกรมได้อย่างถูกต้อง

4.2.1วิธีการ Download Program ให้กับ MCU

สำหรับวิธีการดาวน์โหลด MCU ของ ATMELในกลุ่มเบอร์ที่ใช้วิธีการดาวน์โหลดแบบ ISP เช่น T89C51RD2/AT89C51RD2/AT89C51ED2 และAT89C51AC2นั้นสามารถใช้โปรแกรม FLIP ของATMELในการสั่งดาวน์โหลดได้ซึ่งวิถีการดาวน์โหลดโปรแกรมสามารถทำได้ 2แนวทาง

4.2.1.1 การ ดาวน์โหลดแบบธรรมดา (Manual Download)

สามารถใช้ได้กับบอร์ดไมโครคอนโทรลเลอร์ของบริษัท อีทีที จำกัด ทุกๆรุ่นที่มีพอร์ตสื่อสารอนุกรม RS232 แบบ 4 Pin และมีสวิตช์ RESET และสวิตช์ PSEN (บางรุ่นเรียกสวิตช์ LOAD) อยู่ภายในบอร์ดด้วย

4.2.1.2 การDownloadแบบอัตโนมัติ (Auto Download)

สามารถใช้งานได้กับ บอร์ดไมโครคอนโทรลเลอร์ของบริษัท อีทีที จำกัด ทุกๆรุ่นที่ขั้วดาวน์โหลดแบบ 5 Pin ซึ่งเรียกว่า “ETDOWNLOAD” (บางรุ่นอาจเรียก LOAD-RD2) อยู่ภายในบอร์ดด้วยซึ่งบอร์ไมโครคอนโทรลเลอร์ของ บริษัท อีทีที จำกัด ซึ่งสามารถทำการติดตั้งใช้งาน กับ MCU ของ ATMEL ในกลุ่มเบอร์ที่ใช้การดาวน์โหลดแบบ ISP ได้นั้นมีอยู่ด้วยกันมากมายหลายรุ่น โดยบางรุ่นก็สามารถเลือกติดตั้ง MCU ได้มากกว่า 1 เบอร์ ซึ่งการที่จะเลือกใช้ MCU เบอร์ใดนั้นขึ้นอยู่กับความต้องการของผู้ใช้เป็นหลัก ซึ่งได้แก่

- บอร์ดไมโครคอนโทรลเลอร์รุ่น ET-BASE51
- บอร์ดไมโครคอนโทรลเลอร์รุ่น ET-BASE51 V2
- บอร์ดไมโครคอนโทรลเลอร์รุ่น CP-JR51AC2 V1.0 / EXP
- บอร์ดไมโครคอนโทรลเลอร์รุ่น CP-JR51AC2 V2.0
- บอร์ดไมโครคอนโทรลเลอร์รุ่น CP-JR51RD2 และ CP-JR51RD2 EXP
- บอร์ดไมโครคอนโทรลเลอร์รุ่น CP-AT32 PLUS V2
- บอร์ดไมโครคอนโทรลเลอร์รุ่น CP-SPI/RD2 V1 / EXP
- บอร์ดไมโครคอนโทรลเลอร์รุ่น CP-SPI/RD2 V2

- บอร์ดไมโครคอนโทรลเลอร์รุ่น CP-SPI/RD2 V3
- บอร์ดไมโครคอนโทรลเลอร์รุ่น CP-89C51 V1 PLUS
- ชุดพัฒนาโปรแกรม MCS51 รุ่น ET-SPI51 V1 และ ET-SPI51 V2
- ชุดพัฒนาเรียนรู้หุ่นยนต์รุ่น ET-ROBOT RD2
- ชุดทดลอง MCS51 รุ่น ET-LAB3A (MCS51)

การ Download แบบ Manual

ในการดาวน์โหลดแบบนี้จะใช้กับสาย RS232 แบบ 4 Pin ในการสั่งดาวน์โหลดโปรแกรม ซึ่งสามารถใช้งานได้กับโปรแกรม FLIP ทุกรุ่น แต่ในปัจจุบันโปรแกรม FLIP ได้รับการปรับปรุงเป็น “FLIP V2.4.4” แล้วซึ่งเมื่อต้องการให้โปรแกรม FLIP ติดต่อกับ CPUในมอนิเตอร์โหมด (Monitor Mode) ด้วยวิธีการดาวน์โหลดแบบธรรมดา (Manual Download) นั้นจะต้องสั่ง Resetให้ CPUเข้าทำงานในมอนิเตอร์โหมด (Monitor Mode) ก่อน จึงจะสามารถสั่งงาน CPU ผ่านทางโปรแกรม FLIP ได้ ซึ่งหลักการสำหรับ Reset ให้ CPU เข้าทำงานใน มอนิเตอร์โหมด (Monitor Mode) จะต้องกำหนดให้ขาสัญญาณ PSEN มีสภาวะเป็น “0” ในขณะที่ CPU หลุดพ้นจากสภาวะของการ Reset ซึ่งตามปรกติแล้วหลังการ Reset ทุกครั้ง CPU จะตรวจสอบสภาวะของขาสัญญาณ PSEN ว่าเป็น “0” หรือไม่ถ้าไม่ใช่ก็จะทำงานในโหมการทำงานปรกติแต่ถ้าใช่ก็จะตรวจสอบสถานะของสัญญาณอื่นๆที่เกี่ยวข้องกับการทำงานในมอนิเตอร์โหมด (Monitor Mode) ถ้าเงื่อนไขอื่นๆถูกต้องก็จะเข้าทำงานในมอนิเตอร์โหมด (Monitor Mode) ทันที สำหรับบอร์ดของบริษัท อีทีที จำกัด รุ่นที่มีสวิตช์ RESET และสวิตช์ PSEN อยู่ภายในบอร์ดนั้นการที่จะสั่ง Reset ให้ CPU ของATMEL เข้าทำงานในมอนิเตอร์โหมด (Monitor Mode) ด้วยวิธีการแบบธรรมดานั้นจะต้องทำตามลำดับขั้นตอนดังต่อไปนี้

1. กดสวิตช์ PSEN (บางรุ่นเรียกสวิตช์ LOAD) ค้างไว้เพื่อกำหนดสถานะขาสัญญาณPSEN ให้เป็น “0”

2. กดสวิตช์ RESET เพื่อส่งสัญญาณ RESET ให้กับ CPU โดยสวิตช์ PSEN ต้องกดค้างอยู่เช่นเดิม

3. ปล่อยสวิตช์ RESET เพื่อปล่อยให้ CPU พ้นจากสภาวะการ Reset (สวิตช์ PSEN กดค้าง)

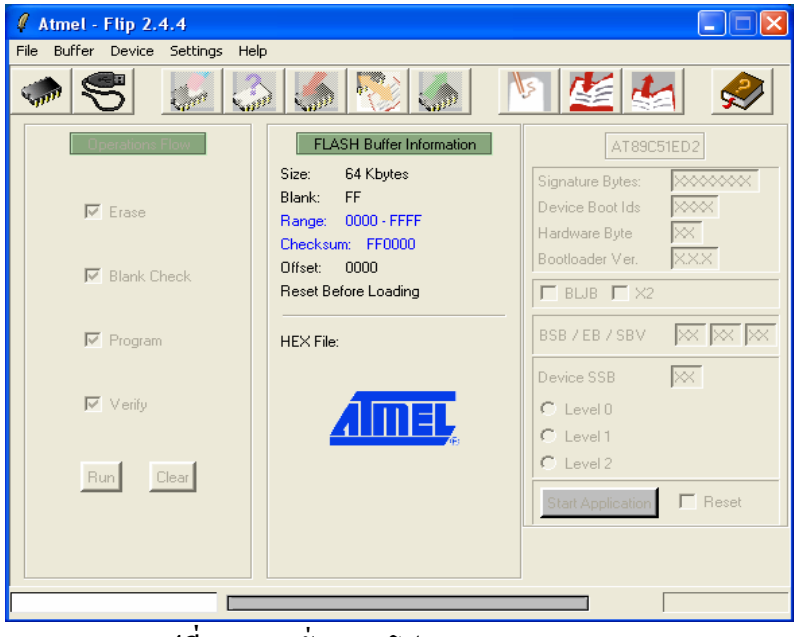
4. ปล่อยสวิตช์ PSEN เป็นลำดับสุดท้ายซึ่งผู้ใช้จะต้องทำการสั่ง Reset การทำงานของ MCU ให้เริ่มต้นในมอนิเตอร์โหมด (Monitor Mode) รอไว้ก่อนที่จะสั่ง Connect การเชื่อมต่อด้วยโปรแกรม FLIP เสมอ ไม่เช่นนั้นแล้วจะไม่สามารถสั่งดาวน์โหลด ได้อย่างถูกต้อง

ลำดับขั้นตอนการดาวน์โหลด HEX File แบบธรรมดา

1. ต่อสาย RS232 แบบ 4 PIN จาก Com Port ของเครื่องคอมพิวเตอร์ PC เข้ากับขั้ว RS232 แบบ 4 Pin ของบอร์ด

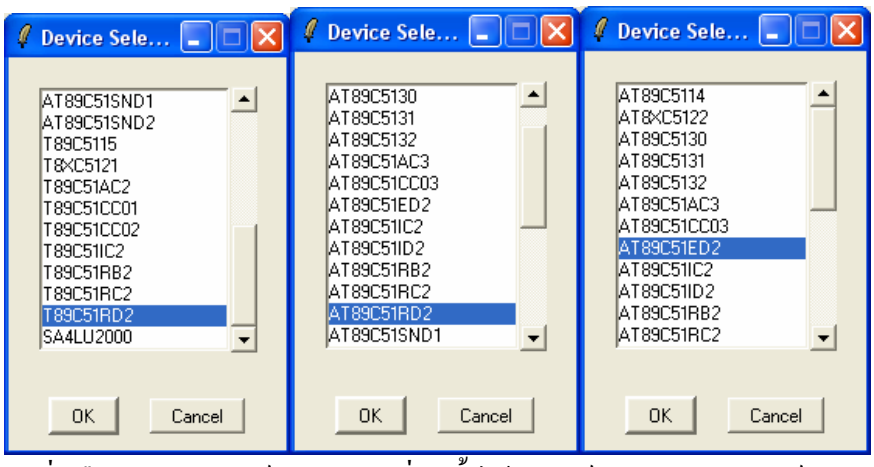
2. จ่ายไฟเลี้ยงวงจรให้บอร์ด ซึ่งจะสังเกตุเห็น LED แสดงสถานะของ PWR สีแดงติดสว่าง

3. สั่ง Run โปรแกรม FLIP V2.4.4 ซึ่งจะได้ผลดังรูปที่ 4.2



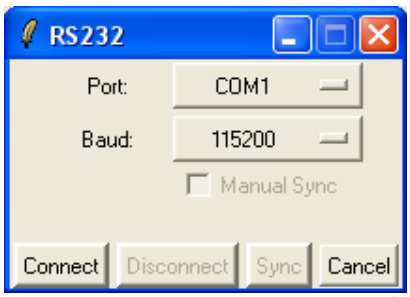
รูปที่ 4.2 การสั่ง Run โปรแกรม FLIP V2.4.4

4. สั่งเลือกกำหนดเบอร์ของ MCU ที่ติดตั้งไว้ในบอร์ด โดยเลือก Device ⟶ Select ซึ่งต้องเลือกกำหนดให้ตรงกับที่ทำการติดตั้งไว้จริงๆในบอร์ดด้วย ดังตัวอย่างในรูปที่ 4.3



รูปที่ 4.3 การสั่งเลือกกำหนดเบอร์ของ MCU ที่ติดตั้งไว้ในบอร์ดสำหรับการดาวน์โหลดโปรแกรม

5. คลิกเมาส์ที่คำสั่ง setting ⟶ Communication ⟶ RS232 จากนั้นเลือกกำหนดComport ให้ตรงกับที่ต่อสายไว้จริง ดังรูปที่ 4.4 (ในตัวอย่างใช้ COM1)



รูปที่ 4.4 การกำหนด Comport ให้ตรงกับสายที่ทำการต่อไว้

6. ทำการรีเซ็ต MCU ให้เข้าทำงานใน Monitor โดยมีลำดับขั้นตอนดังนี้

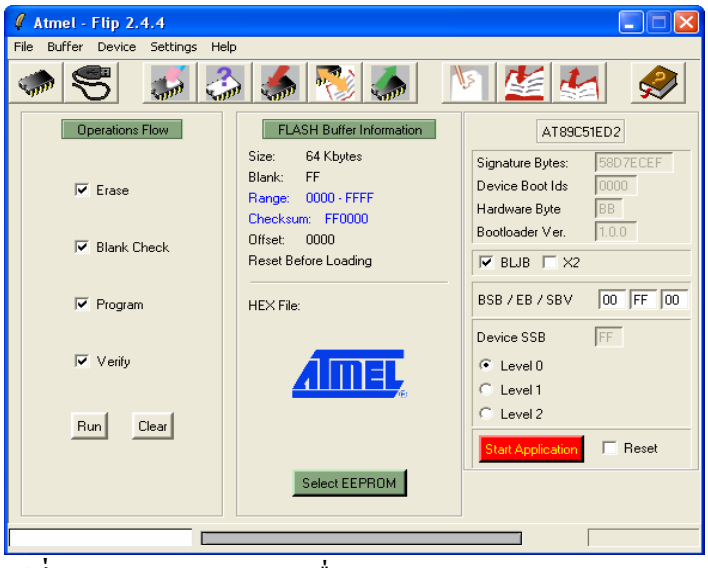
a) กดสวิตช์ PSEN ค้างไว้เพื่อกำหนดสถานะขาสัญญาณ PSEN ให้เป็น “0”

b) กดสวิตช์ RESET เพื่อส่งสัญญาณ RESET ให้กับ CPU โดยสวิตช์ PSEN ต้องกดค้างอยู่เช่นเดิม

c) ปล่อยสวิตช์ RESET เพื่อปล่อยให้ CPU พ้นจากสภาวะการ Reset (สวิตช์ PSEN ยังกดค้างอยู่)

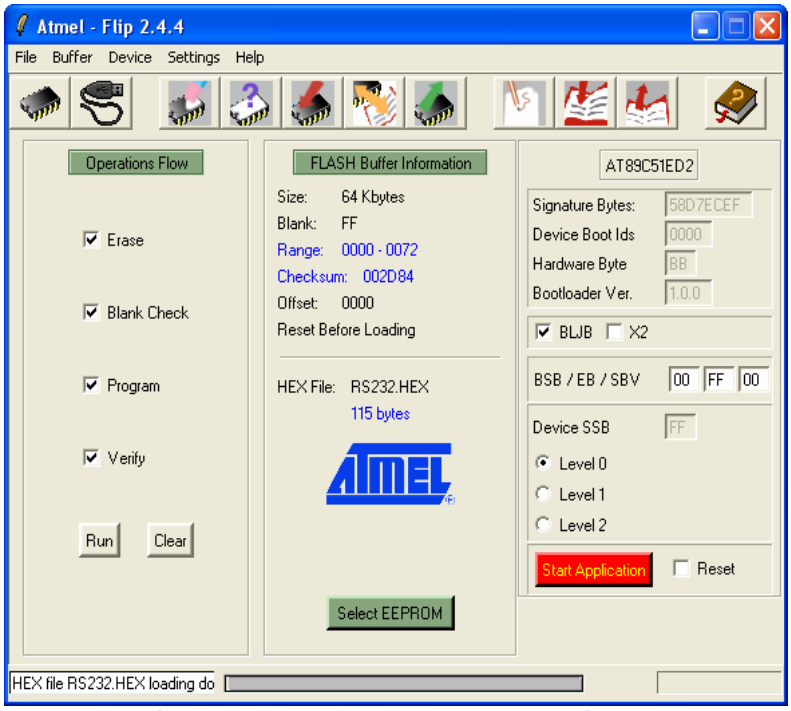
d) ปล่อยสวิตช์ PSEN เป็นลำดับสุดท้าย

7. คลิกเมาส์ที่ปุ่ม Connect เพื่อทำการติดต่อสื่อสารกับ MCU ใน Monitor Mode ซึ่งจะได้ผลดังรูปที่ 4.5



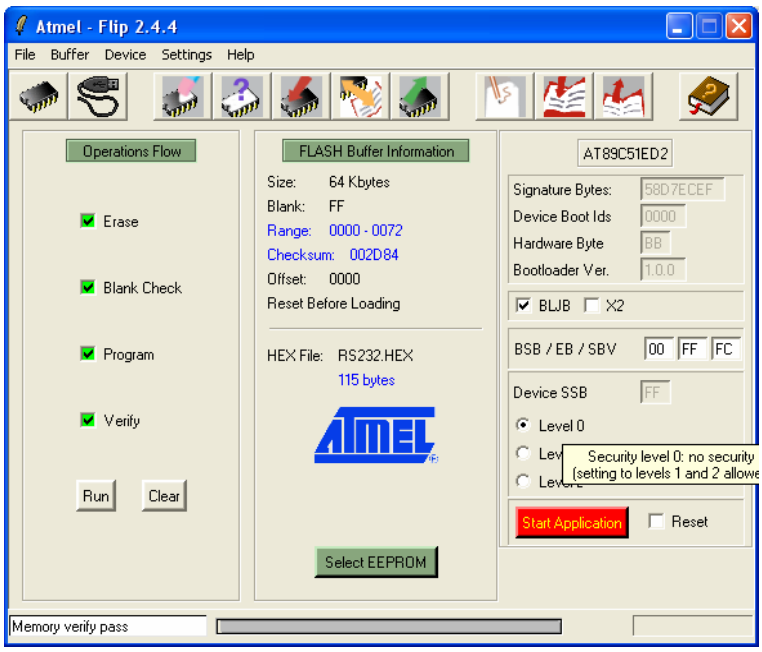
รูปที่ 4.5 ทำการติดต่อติดต่อสื่อสารกับ MCU ใน Monitor Mode

8. สั่งเปิด Hex File ที่ต้องการจะ Download ให้กับ MCU มารอไว้ใน Buffer ของโปรแกรม FLIP โดยใช้คำสั่ง File ⟶ Load Hex File… ดังรูปที่ 4.6



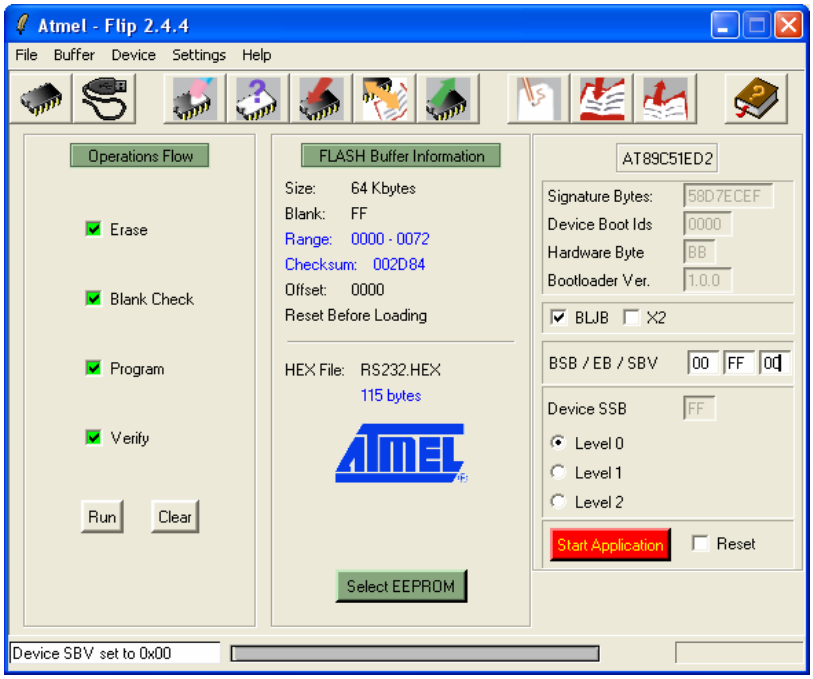
รูปที่ 4.6 การเปิด Hex File ที่ต้องการจะ Download ให้กับ MCU เพื่อให้รอใน Buffer ของโปรแกรม

9. คลิกเมาส์ที่หน้าตัวเลือกคำสั่งใน Tab ของ Operation Flow ให้ครบทุกคำสั่ง ซึ่งได้แก่ Erase, Bank Check, Program, Verify จากนั้นคลิกเมาส์ที่ปุ่มคำสั่ง Run และรอจนการทำงานของโปรแกรมเสร็จเรียบร้อยดังรูปที่ 4.7



รูปที่ 4.7 กำหนดค่า Operation Flow

10. ตรวจสอบค่า Device BSB และ SBV ว่ามีค่าเป็น 00 ทั้งหมดแล้วหรือยัง ซึ่งถ้ายังไม่เป็น 00 ให้ทำการแก้ไขค่าให้เป็น 00 โดยคลิกเมาส์ในช่องตัวเลขแล้วพิมพ์ค่า 00 แทนที่ลงไปทั้ง 2 ช่องดังรูป



รูปที่ 4.8 ตรวจสอบค่า Device BSB และ SBV

11. กดสวิตช์ Reset ให้กับบอร์ดเพื่อให้บอร์ดเริ่มต้นทำงานตามโปรแกรมที่ได้ทำการ Download ไปให้ ซึ่งถ้าไม่เกิดความผิดพลาดใดๆจะเห็น MCU เริ่มต้นทำงานทันที

การ Download แบบอัตโนมัติ

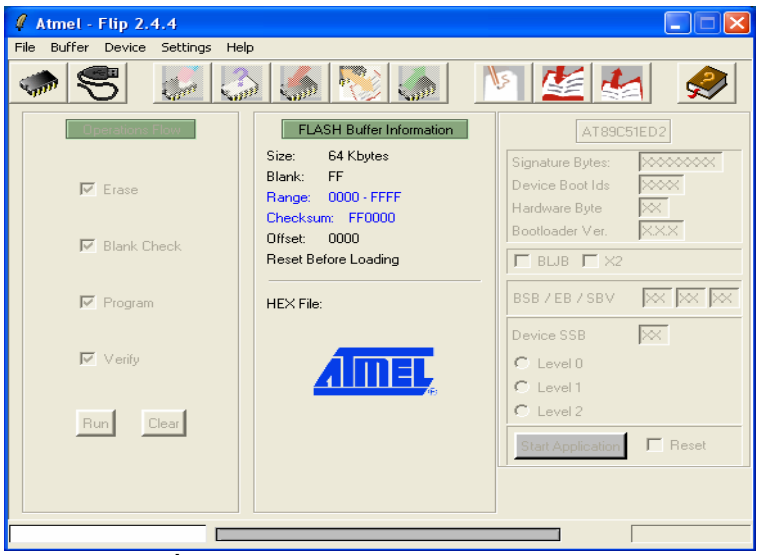
สำหรับการ Download แบบนี้จะเกิดความสะดวกในการใช้งานเป็นอย่างมาก เนื่องจากผู้ใช้ไม่ต้องเสียเวลามาคอยกดสวิตช์ PSEN และ RESET ภายในบอร์ดเพื่อบังคับให้ MCU เข้าทำงานใน Monitor ให้ยุ่งยากอีกต่อไป เนื่องจากโปรแกรม FLIP V2.4.4 ของ ATMEL ได้รับการปรับปรุงให้สามารถสั่ง Download โปรแกรมให้กับ MCU ด้วยวิธีการแบบอัตโนมัติได้ด้วย โดยจะใช้สัญญาณ RTS ในการกำหนดโลจิกให้กับสัญญาณ PSENของ MCU และใช้สัญญาณ DTRในการควบคุมการรีเซ็ตของ MCUซึ่งบอร์ไมโครคอนโทรลเลอร์ของบริษัท อีทีที จำกัดทุกรุ่นที่มีขั้ว ET-DOWNLOAD (บางรุ่นเรียก LOAD-RD2) แบบ 5PIN อยู่ภายในบอร์ดนั้น ได้จัดเตรียมวงจรสำหรับการ Download แบบอัตโนมัติไว้ด้วยแล้ว โดยให้ใช้สาย RS232 รุ่น ET-DOWNLOADแบบ 5 Pin ต่อเข้ากับบอร์ดที่ขั้ว ET-DOWNLOAD ของบอร์ดเท่านั้น

ลำดับขั้นตอนการ Download HEX File แบบอัตโนมัติ

1. ต่อสาย RS232 Download แบบ 5 PIN จาก Com Port ของเครื่องคอมพิวเตอร์ PC เข้ากับขั้ว ET-DOWNLOAD (ในบางรุ่นอาจเรียกว่า LOAD-RD2) แบบ 5 Pin ของบอร์ด

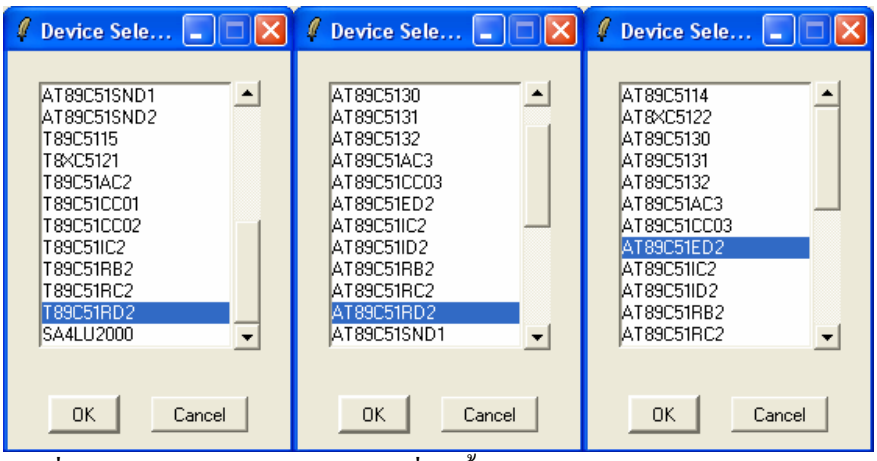
2. จ่ายไฟเลี้ยงวงจรให้บอร์ด ซึ่งจะสังเกตุเห็น LED แสดงสถานะของ PWR สีแดงสว่างอยู่

3. สั่ง Run โปรแกรม FLIP V2.4.4 ซึ่งจะได้ผลดังรูปที่ 4.9



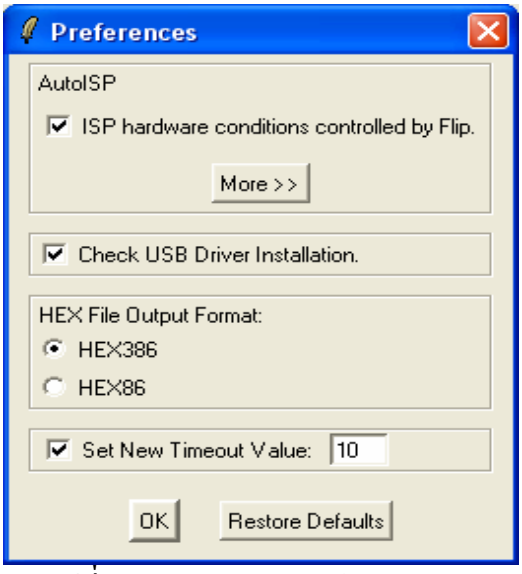
รูปที่ 4.9 การสั่ง Run โปรแกรม FLIP V2.4.4

4. สั่งเลือกกำหนดเบอร์ของ MCU ที่ติดตั้งไว้ในบอร์ด โดยเลือก Device ⟶ Select ซึ่งต้องเลือกกำหนดให้ตรงกับที่ทำการติดตั้งไว้จริงๆในบอร์ดด้วย ดังรูป 4.11



รูปที่ 4.10 การสั่งเลือกกำหนดเบอร์ของ MCU ที่ติดตั้งไว้ในบอร์ดสำหรับการดาวน์โหลดโปรแกรม

5. คลิกเมาส์ที่เมนูคำสั่ง Settings ⟶ Preferences… ในขั้นตอนเลือกกำหนดค่า Time-Out ของการสื่อสาร โดยคลิกเมาส์เติมเครื่องหมาย (√) ที่หน้าตัวเลือกในหัวข้อ **“Set New Timeout Value”** พร้อมกับกำหนดค่าในช่องรับค่าเท่ากับ10 จากนั้นให้เลือกกำหนดการสื่อสารเป็นแบบ Auto ISP โดยให้คลิกเมาส์เติมเครื่องหมาย (√) ที่หน้าตัวเลือกในหัวข้อ **“ISP Hardware conditions controlled by Flip.”** แล้วเลือกคลิกเมาส์ที่ปุ่มคำสั่ง **“More >>”** เพื่อเข้าไปกำหนดคุณสมบัติของสัญญาณควบคุม ดังรูปที่ 4.12

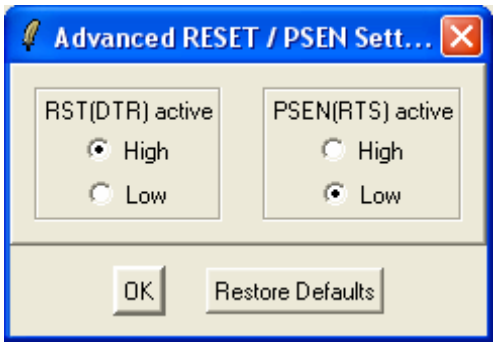


รูปที่ 4.11 กำหนดค่าต่างๆใน Preferences

ในขั้นตอนนี้จะเป็นการเลือกกำหนดคุณสมบัติของสัญญาณจาก Comport ที่จะใช้ในการควบคุมสัญญาณ RESET และ PSEN ของ MCU ในบอร์ด ให้เข้าทำงานใน Monitor Mode โดยอัตโนมัติ โดยให้เลือกกำหนดค่าตัวเลือกเป็นดังนี้

- **RST (DTR) active** จะใช้สำหรับเลือกกำหนดคุณสมบัติของสัญญาณ DTRที่จะใช้ในการควบคุมการ Reset ของ MCU โดยให้เลือกเป็น High

- **PSEN (RTS) active** จะใช้สำหรับเลือกกำหนดคุณสมบัติของสัญญาณ RTSที่จะใช้ในการกำหนดโลจิกให้กับขาสัญญาณ PSEN ของ MCU โดยให้เลือกกำหนดเป็นLowซึ่งเมื่อทำการเลือกกำหนดค่าต่างๆดังกล่าวข้างต้นเสร็จเรียบร้อยแล้วให้เลือกคลิกเมาส์ที่ปุ่มคำสั่ง Oเพื่อให้โปรแกรม FLIP ทำการบันทึกค่าตัวเลือกนี้ไว้ ซึ่งในครั้งต่อๆไป ที่เรียกใช้งานโปรแกรFLIP อีก ก็ไม่จำเป็นต้องเข้ามากำหนดค่าในส่วนนี้อีกแล้ว ในครั้งต่อไปให้ข้าขั้นตอนนี้ไปเลย

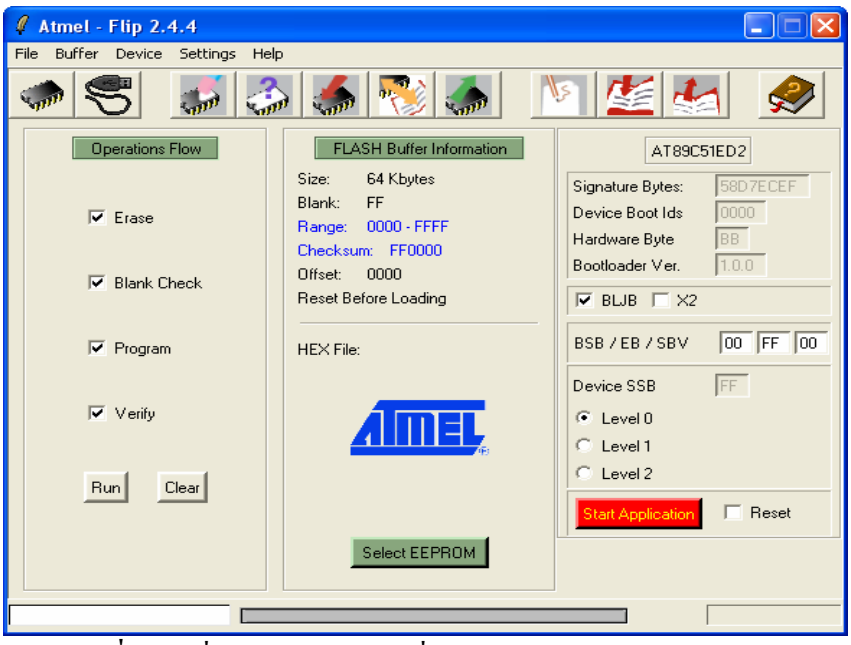


รูปที่ 4.12 กำหนดค่าที่จะใช้ในการควบคุมสัญญาณ RESET และ PSEN ของ MCU ในบอร์ด

6. คลิเมาส์ที่คำสั่ง setting ⟶ Communication ⟶ RS232 จากนั้นเลือกกำหนดComport ให้ตรงกับที่ต่อสายไว้จริง ดังรูปที่ 4.14

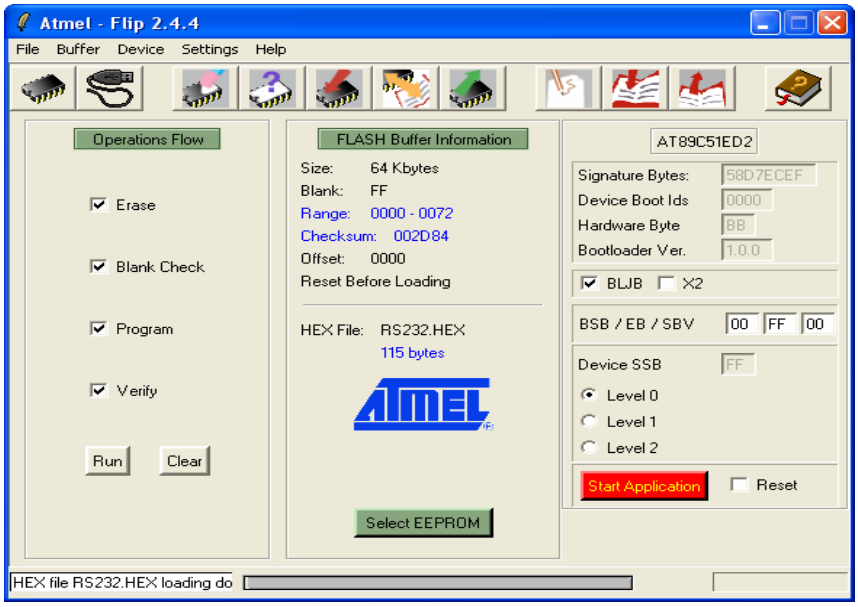
รูปที่ 4.13 การกำหนด Comport ให้ตรงกับสายที่ทำการต่อไว้

7. คลิกเมาส์ที่ปุ่ม Connect เพื่อให้โปรแกรม FLIP เริ่มต้นทำการติดต่อสื่อสารกับ MCU ใน Monitor Mode ซึ่งจะได้ผลดังรูปที่ 4.15



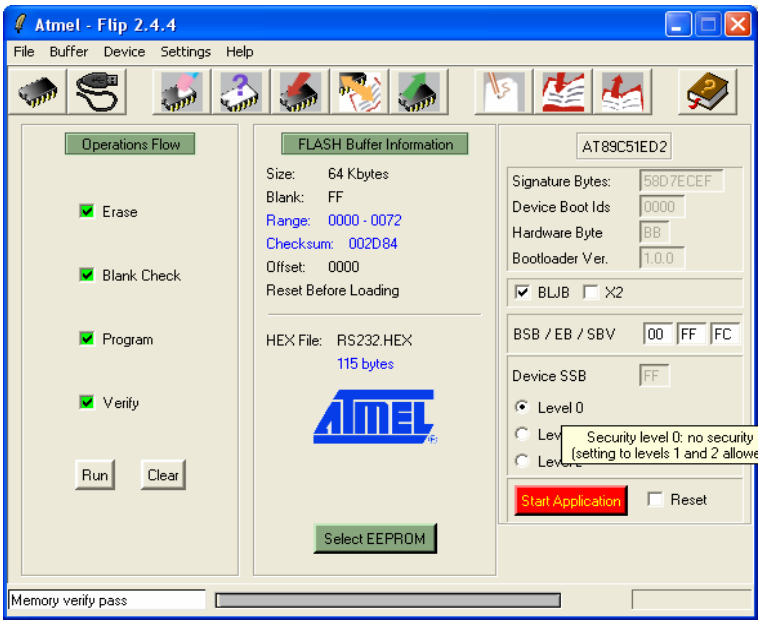
รูปที่ 4.14 เริ่มต้นทำการติดต่อสื่อสารกับ MCU ใน Monitor Mode

8. สั่งเปิด Hex File ที่ต้องการจะ Download ให้กับ MCU มารอไว้ใน Buffer ของโปรแกรม FLIP โดยใช้คำสั่ง File ⟶ Load Hex File… ดังรูปที่ 4.16



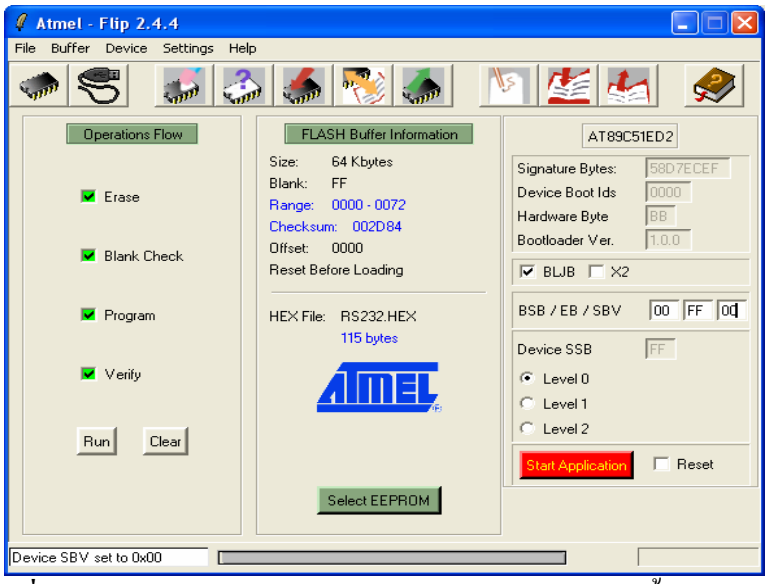
รูปที่ 4.15 การเปิด Hex File ที่ต้องการดาวน์โหลดให้กับ MCU เพื่อให้รอใน Buffer ของโปรแกรม

9. คลิกเมาส์ที่หน้าตัวเลือกคำสั่งใน Tab ของ Operation Flow ให้ครบทุกคำสั่ง ซึ่งได้แก่ Erase, Bank Check, Program, Verify จากนั้นคลิกเมาส์ที่ปุ่มคำสั่ง Run และรอจนการทำงานของโปรแกรมเสร็จเรียบร้อยดังรูป



รูปที่ 4.16 กำหนดค่าของ Operation Flow

10. ตรวจสอบค่า Device BSB และ SBV ว่ามีค่าเป็น 00 ทั้งหมดแล้วหรือยัง ซึ่งถ้ายังไม่เป็น 00 ให้ทำการแก้ไขค่าให้เป็น 00 โดยคลิกเมาส์ในช่องตัวเลขแล้วพิมพ์ค่า 00 แทนที่ลงไปทั้ง 2 ช่องดังรูป 4.18



รูปที่ 4.17 ตรวจสอบค่า Device BSB และ SBV ให้เป็น 00 ทั้งหมด

11. ทำการคลิกเมาส์ที่ปุ่ม Start Application กดสวิตช์ Reset ให้กับบอร์ดเพื่อให้บอร์ดเริ่มต้นทำงานตามโปรแกรมที่ได้ทำการ Download ไปให้ ซึ่งถ้าไม่เกิดความผิดพลาดใดๆจะเห็นMCU เริ่มต้นทำงานทันที และควรถอดสาย Download ออกจากบอร์ดด้วยเสมอเนื่องจากเมื่อจบการทางานจากโปรแกรม FLIP ไปแล้วสถานะของสัญญาณ RTS และDSR ของพอร์ตสื่อสารอนุกรม RS232 อาจมีสถานะไม่แน่นอน ซึ่งอาจทำงานค้างอยู่ อันจะส่งผลทำให้ MCU ถูกรีเซ็ตอยู่ตลอดเวลาทำให้ MCU ไม่ทำงาน

ปัญหาต่างๆในขณะใช้งานโปรแกรม FLIP และแนวทางการแก้ไข

ในบางครั้งเมื่อเรียกใช้คำสั่งต่างๆของโปรแกรม FLIP แล้ว อาจเกิดความผิดพลาดบางประการขึ้น ซึ่งอาจไม่ใช่ปัญหาที่เกิดจากความบกพร่องของระบบฮาร์ดแวร์ แต่อาจเกิดการการกำหนดพารามิเตอร์บางอย่างในโปรแกรมไม่ถูกต้องหรือข้ามขั้นตอนบางประการไป ซึ่งเมื่อโปรแกรม FLIP ไม่สามารถปฏิบัติตามคำสั่งที่ผู้ใช้งานสั่งไปได้สำเร็จจะแสดงอาการ Error ต่างๆให้ทราบ ซึ่งพอสรุปได้ดังนี้

1. **Time Out Error** เป็นความผิดพลาดที่เกิดจากการที่โปรแกรม FLIP ไม่สามารถทำการสื่อสารกับ CPU ใน Monitor Mode ได้ ซึ่งอาจเกิด หลายสาเหตุ เช่น

- ใช้สายสัญญาณในการ Download ไม่ถูกต้อง ซึ่งถ้าใช้งานบอร์ดไมโครคอนโทรลเลอร์ของบริษัท อีทีที จำกัด รุ่นที่สามารถใช้งานกับโปรแกรม Flip นั้น ถ้าใช้สายสัญญาณRS232 แบบ 4Pin จะต้องใช้การ Download แบบ Manual เท่านั้น ส่วนถ้าต้องการDownload แบบ Auto จะต้องใช้สาย ET-DOWNLOAD (5Pin) ของ บริษัท อีทีที จำกัด

- การต่อสายสัญญาณระหว่างขั้วต่อ RS232 ของบอร์ดไมโครคอนโทรลเลอร์กับขั้วต่อพอร์ตสื่อสารอนุกรม RS232 ของคอมพิวเตอร์ยังไม่เรียบร้อย หรือ ต่อไม่ตรงกับที่กำหนดตัวเลือกไว้ในโปรแกรม หรือ การกำหนดรูปแบบและตัวเลือกต่างๆในการสื่อสารไม่ถูกต้องเมื่อพบปัญหานี้ให้ลองทำการตรวจสอบค่าต่างๆในการสื่อสารใน Setting ⟶ Preferences... และ Setting ⟶ Communication ⟶ RS232

- ยังไม่ได้รีเซ็ตให้ CPU เข้าทำงานใน Monitor Mode รอไว้ก่อนที่จะสั่งงานโปรแกรมในกรณี Download แบบ Manual หรือกำหนดค่า Option สำหรับการ Download แบบAuto ที่ “**ISP Hardware conditions controlled by Flip.**” ไม่ถูกต้อง หรือ บอร์ดยังไม่พร้อมทำงาน เช่น ยังไม่ได้จ่ายไฟเลี้ยงให้บอร์ด

- กำหนดค่า Baud rate เร็วเกินไป ซึ่งในกรณีที่ใช้งานกับเครื่องคอมพิวเตอร์ที่มีความเร็วมากๆนั้น ควรกำหนดค่า Baud rate ในการสื่อสารให้ช้าลง ซึ่งอาจใช้ค่า 19200 หรือ 9600ก็พอ เพราะถ้ากำหนดให้ความเร็วมากเกินไป จะทำให้เกิดความผิดพลาดบ่อยครั้งขึ้น

2. **Software Security Bit Set. Cannot access device Data** เป็นความผิดพลาดที่เกิดจากการนำ CPU ที่มีการสั่ง Lock Bit ของ Security Bit ไว้ก่อนแล้ว จึงมาสั่ง Program หรือ Verifyหรือ Read ในภายหลังโดยยังไม่ได้สั่งลบข้อมูลเก่าออกเสียก่อน ซึ่งให้แก้ปัญหาด้วยการสั่งลบข้อมูล (Erase) เสียก่อนแล้วจึงสั่งเขียนข้อมูลใหม่อีกครั้งหนึ่ง

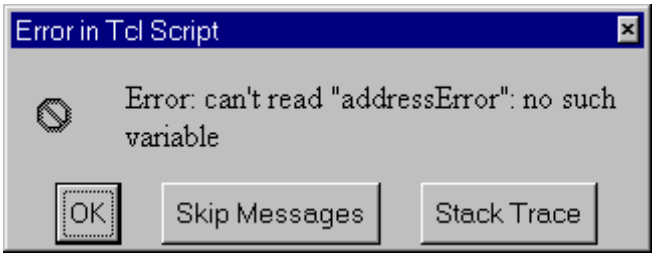
3. **The board reply is not correct** เป็นความผิดพลาดที่เกิดจากการสื่อสารข้อมูลระหว่างโปรแกรม FLIP กับ ไมโครคอนโทรลเลอร์ เกิดความผิดพลาดในลักษณะของ Framing Errorขึ้น ซึ่งปัญหาอาจเกิดจากการกำหนดค่า Baud rate ไม่ถูกต้องกับค่าความถี่ของ Crystal

4. **The RS232 port could not be opened** เป็นความผิดพลาดที่เกิดจากโปรแกรม FLIP ไม่สามารถสั่งเปิดการทำงานของพอร์ตสื่อสารอนุกรม RS232 ของเครื่องคอมพิวเตอร์ได้ ซึ่งอาจเกิดจากการกำหนดหมายเลข Comport ในโปรแกรมที่เลือกไว้ไม่มีอยู่จริง หรือมีโปรแกรมอื่นเรียกใช้งาน Comport นั้นค้างอยู่ หรือเรียกใช้งานโปรแกรม FLIP ในขณะที่กำลังมีการใช้งานComport อยู่ด้วย ซึ่งให้ลองปิดโปรแกรม FLIP แล้วสั่งเปิดโปรแกรมใหม่ดู ถ้ายังเกิดปัญหาเดิมอยู่อีกอาจลองตรวจสอบสาเหตุอื่นๆที่เกี่ยวข้องและทำการแก้ไข

5. **Check sum error** เป็นความผิดพลาดที่เกิดจากการที่ CPU รับข้อมูลที่ส่งไปจากคอมพิวเตอร์PC ไม่ครบถูกต้องทั้งหมด ซึ่งปัญหาอาจเกิดจากการกำหนดความเร็วในการสื่อสารBaud rateเร็วเกินไป หรือกำหนดไว้ไม่เหมาะสมกับค่าความถี่ Crystal ค่า 18.432MHz ให้ลองเปลี่ยนค่า Baud rate ให้ช้าลงกว่าเดิม ซึ่งค่าที่เหมาะสมได้แก่ 9600, 19200 และ 38400 แต่ถ้าคอมพิวเตอร์ไม่เร็วมากนักก็อาจกำหนดเป็น 57600 หรือ 115200 ก็ได้ แต่ถ้ากำหนดค่าสูงๆแล้วเกิด Error ควรลดค่า Baud rate ให้ช้าลงกว่าเดิม

6. การสั่ง Load HEX ไม่ได้ เป็นความผิดพลาดที่เกิดจากการที่โปรแกรม FLIP ไม่สามารถอ่านข้อมูลใน HEX File ออกมาได้ ซึ่งอาจเกิดจากไฟล์ที่สั่งโหลดนั้น ไม่ใช่ไฟล์แบบ Intel HEX เนื่องจากโปรแกรม FLIP สามารถใช้งานกับไฟล์แบบ Intel HEX เท่านั้น ส่วนไฟล์ในรูปแบบอื่นๆจะไม่สามารถนำมาใช้งานกับโปรแกรมนี้ได้ ส่วนปัญหาอีกประการหนึ่งที่มักพบอยู่บ่อยๆคือโปรแกรม FLIP ไม่สามารถอ่าน HEX File ได้ทั้งๆที่ไฟล์ที่สั่งให้อ่านนั้นเป็นไฟล์แบบ Intel HEXอยู่แล้ว ซึ่งที่พบอยู่บ่อยๆก็ได้แก่ HEX File ที่สั่งแปลโดยใช้โปรแกรม Assembler ของSXA51.EXE เนื่องจาก HEX File ที่ได้จากการแปลของโปรแกรมตัวนี้จะเกิดบันทัดว่างอยู่ในไฟล์ในส่วนเริ่มต้มบรรทัดแรกด้วย 1 บรรทัด ซึ่งตามรูปแบบของ HEX File แล้ว ในแต่ละบรรทัดของไฟล์จะต้องเริ่มต้นด้วย

เครื่องหมายโคลอน (☺ แล้วตามด้วยข้อมูลต่างๆในบรรทัดนั้น แต่เมื่อบรรทัดแรกเป็นบรรทัดว่าง โปรแกรมจึงแสดง Error ว่าไม่ใช่ HEX File โดยโปรแกFLIPจะแสดง Error .ให้ทราบดังรูปที่ 4.21



รูปที่ 4.18 Error แสดงว่าไฟล์ที่เลือกไม่ใช่ HEX File

7. **เมื่อสั่งโปรแกรมข้อมูลให้กับ CPU เรียบร้อยแล้วหลังจากรีเซ็ตบอร์ดแล้วไม่ทำงาน** ซึ่งปัญหานี้อาจเกิดจากสาเหตุความผิดพลาดหลายประการ ซึ่งพอสรุปได้ 2 กรณี คือ

- โปรแกรมที่เขียนขึ้นไม่ถูกต้องยังไม่สามารถทำงานได้เอง ซึ่งปัญหานี้ผู้ใช้ต้องหาทางตรวจสอบและแก้ไขความผิดพลาดที่เกิดขึ้นเอง

- ยังไม่ได้มีการสั่ง Load HEX เข้ามารอไว้ยัง Buffer แล้วสั่งโปรแกรม (Program Device) ซึ่งโปรแกรม FLIP จะนำข้อมูลที่อยู่ใน Buffer เขียนไปยังหน่วยความจำของโปรแกรม

- สวิตช์ PSEN อาจถูกกดค้างอยู่ จึงทำให้การรีเซ็ตบอร์ดทุกๆครั้งนั้น CPU จะเข้าไปทำงานใน Monitor Mode เสมอ ซึ่งปัญหานี้สามารถตรวจสอบได้โดยการวัดระดับลอจิกที่ขาสัญญาณ PSEN ของ CPU ซึ่งควรมีสภาวะเป็น “1” ถ้าไม่มีการกดสวิตช์ PSEN ไว้และควรมีสภาวะเป็น “0” ถ้ามีการกดสวิตช์ PSEN ไว้

- สวิตช์ RESET อาจถูกกดค้างอยู่ จึงทำให้ CPU ไม่สามารถหลุดพ้นจากสภาวะการรีเซ็ตได้ ซึ่งปัญหานี้สามารถตรวจสอบได้โดยการวัดระดับลอจิกที่ขาสัญญาณ RESET ของCPU ซึ่งควรมีสภาวะเป็น “0” ถ้าไม่มีการกดสวิตช์ RESET ไว้ และควรมีสภาวะเป็น “1” ถ้ามีการกดสวิตช์ RESET ไว้

- ในกรณีของการ Download ด้วยวิธีการ Auto Download นั้น หลังการ Download เสร็จแล้วควรถอดสาย Download ออกด้วยทุกครั้ง เนื่องจากเมื่อจบการทำงานของโปรแกรมFLIP ไปแล้วสัญญาณ RTS และ DTR ของพอร์ตสื่อสาร RS232 อาจมีสถานะไม่แน่นอนทำให้สัญญาณทั้ง 2 ควบคุมการรีเซ็ตของ MCU อยู่ตลอดเวลาได้

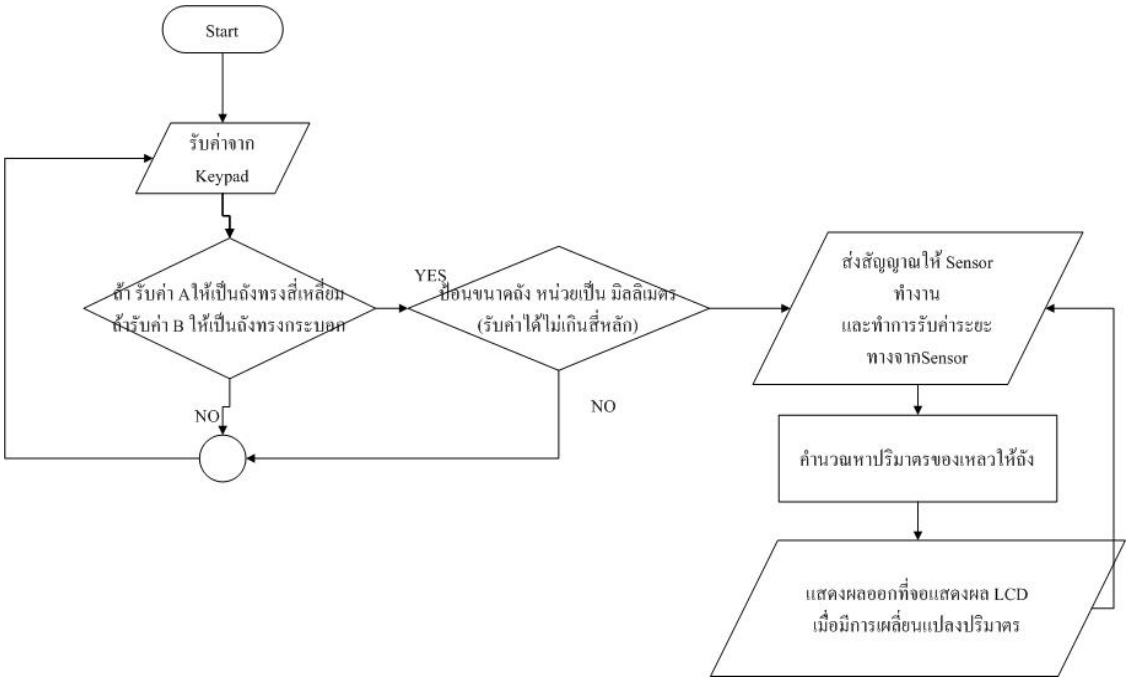
- ค่าของ Device BSB ยังไม่ได้ถูกกำหนดให้มีค่าเป็น 00H ไว้ ซึ่งจะทำให้โปรแกรมกระโดดไปทำงานยังตำแหน่งที่ชี้โดย Device SBV แทน ซึ่งถ้าค่าของ Device SBV ไม่ใช้ศูนย์ก็จเหมือนกับว่าโปรแกรมไม่ทำงาน ซึ่งการแก้ไข ปัญหานี้ หลังจากสั่งโปรแกรมข้อมูลให้กับ CPU เรียบร้อยแล้วควรกำหนดให้ค่าของ Device BSB และ Device SBV มีค่าเป็น 00H ไว้ทั้งคู่จะดีที่สุด

4.3 การเขียนชุดโปรแกรมคำสั่งควบคุมอุปกรณ์[4]

4.3.1 ลำดับความคิดในการเขียนชุดโประแกรมคำสั่งควบคุมอุปกรณ์ (Algorithms)

1. รับค่าจากชุดปุ่มกด เพื่อเลือกรูปแบบของถังน้ำมัน
2. รับค่าจากชุดปุ่มกด เพื่อกำหนดขนาดถังน้ำมันที่ได้เลือกในข้อที่ 1
3. สั่งให้ชุดเซนเซอร์ทำงาน โดยส่งสัญญาณคลื่นสี่เหลี่ยมที่มีความกว้างไม่ต่ำกว่า 10 ไมโครวินาที เพื่อที่หาความระยะห่างจากเซนเซอร์กับผิวของเหลว
4. ประมวณผลจากสูตรทางคณิตศาสตร์ โดยคำนึงถึงรูปแบบถังน้ำมันที่เลือก
5. แสดงผลที่ได้จาดการประมวลออกทางจอแสดงผล (LCD)
6. เมื่อระดับของเหลวในถังเปลี่ยนไป ทำการประมวลผลข้อที่ 4

จะสามรถเขียนเป็นแผนภาพแสดงความคิดในการเขียนชุดโประแกรมคำสั่งควบคุมอุปกรณ์ (Flowchart) ดังรูป 4.19



รูปที่ 4.19 แผนภาพแสดงความคิดในการเขียนชุดโประแกรมคำสั่งควบคุมอุปกรณ์

4.3.2 ชุดโปรแกรมคำสั่งควบคุมเซนเซอร์วัดระยะทางด้วยคลื่นอัลตร้าโซนิค SRF05

```
sbit echo = P3^6;
sbit trigger = P3^7;
void trig_pulse(void)
    { trigger = 1;
    delay_ms(1);
    trigger = 0;}
unsigned int distance()
{   unsigned int mc;
    trigger = 0;
    echo = 1;
            TMOD &= 0x0F;
            TMOD |= 0x10;
            TH1 = 0x00;
            TL1 = 0x00;
            TF1 = 0;
            TR1 = 0;
            trig_pulse();
while(!echo);
            TR1 = 1;
                while(echo);
            TR1 = 0;
            TF1 = 0;
            mc = TH1;
            mc <<=8;
            mc+=TL1;
            delay_ms(10);
            mc=mc*1.1;
return(mc/10);}
```

4.3.3 ชุดโปรแกรมคำสั่งควบคุมจอแสดงผล (LCD)

```
sbit e =P3^4 ;
sbit rs=P3^5 ;
void inti_lcd(void);
void lcd_comm(unsigned char com);
void delay(int z) ;
void lcd_display(unsigned char text);
void lcd_comm(unsigned char com)
      { rs = 0 ;
        e = 1 ;
        P2 =  com ;
        delay(5);
       e = 0 ;
       delay(5);     }
void inti_lcd( )
       { delay(100);
        lcd_comm(0x38) ;
        lcd_comm(0x0C) ;
        lcd_comm(0x01) ;  }
void delay(int z)
       { int x,y ;
         for(x=0;x<z;x++)
         {for(y=0;y<200;y++);}      }
 void lcd_display(unsigned char text)
       {  rs = 1 ;
         e = 1  ;
         P2 = text ;
         delay(5);
         e = 0 ;
         delay(5);   }
```

4.3.4 ชุดโปรแกรมคำสั่งควบคุมการรับค่าจากชุดปุ่มกด (Keypad)

```
unsigned int keypad( )
    {    unsigned chardat_scankey[]={0x77,0xb7,0xd7,0xe7,0x7b,0xbb,0xdb,0xeb
         ,0x7d,0xbd,0xdd,0xed,0x7e,0xbe,0xde,0xee};
        unsigned char i ,j , x ,y,z=1,w ;
         while(z)
         { x=0xf7;
            for(i=0;i<4;i++)
            {P1 = x ;
             for(j=0;j<16;j++)
                {y = P1 ;
                 if(y==dat_scankey[j])
                   {delay(100);
                    if(y==dat_scankey[j])
                       {z=0;
                    w=j;} } }
                      x = x>>1;
                      x =x|0xf0; }
          }return (w);   }
```

4.3.5 ชุดโปรแกรมคำสั่งควบคุมการคำนวณหาปริมาตร

```
#include <reg52.h>
#include <stdio.h>
#include <lcd.h>
#include <keypad.h>
#include <math.h>
#include <ult.h>
void show_error()
{       unsigned int i,x=0x82;
        unsigned char error[]="ERROR";
        inti_lcd();
        lcd_comm(0x80);
        for(i=0;i<5;i++)
        {lcd_display(error[i]);}
}
void show_length()
{       unsigned int i;
        unsigned char length1[]="Length";
        unsigned char num3[]="mm.";
        inti_lcd();
        lcd_comm(0x80);
        for(i=0;i<6;i++)
                {lcd_display(length1[i]);}
                 lcd_comm(0xc4);
                 for(i=0;i<3;i++)
                        {lcd_display(num3[i]);}
}
void show_width()
{       int i;
        unsigned char width1 []="Width";
```

```
        unsigned char num3[]="mm.";
        inti_lcd();
        lcd_comm(0x80);
        for(i=0;i<5;i++)
                {lcd_display(width1[i]);}
                 lcd_comm(0xc4);
                 for(i=0;i<3;i++)
                        {lcd_display(num3[i]);}
}
void show_high()
{       int i;
        unsigned char high1 []="High";
        unsigned char num3[]="mm.";
        inti_lcd();
        lcd_comm(0x80);
        for(i=0;i<4;i++)
                {lcd_display(high1[i]);}
                 lcd_comm(0xc4);
                 for(i=0;i<3;i++)
                        {lcd_display(num3[i]);}
}
void show_radiance()
{       int i;
        unsigned char radiance1[]="Radiance";
        unsigned char num3[]="mm.";
                inti_lcd();
                lcd_comm(0x80);
                for(i=0;i<8;i++)
                        {lcd_display(radiance1[i]);}
                        lcd_comm(0xc4);
```

```
                for(i=0;i<3;i++)
                        {lcd_display(num3[i]);}
}
unsigned int key_number(unsigned int n)
{       unsigned int key=0,q=0,x=0xc0,number=0;
        unsigned char number1[]="0123456789ABCDEF ";
                while(q<4)
                { key=keypad();
                  if((key>=0)&&(key<=9))
                        {lcd_comm(x);
                         lcd_display(number1[key]);
                         delay_ms(5);
                                if(q==0)
                        {number=key;}
                else {number=(number*10)+key;}
                        x++;  q++;
              }else if(key==15)
                    {   q=4;
                }else
                { show_error();
                  delay_ms(10);
                  if (n==1)
                {show_length();}
                else if (n==2)
                {show_width();}
                else if (n==3)
                {show_high();}
                else if (n==4)
                {show_radiance();}
```

```
            number=0;
            q=0;
            x=0xc0;}
            }return(number);
}
void show_volume(unsigned long value)
{           unsigned int i;
            unsigned char number[]="0123456789ABCDEF ";
            unsigned char num3[]="Lt.";
            unsigned char vol[]="VOL.";
            unsigned char a,b,c,d,e,f,g,h,k;
a=b=c=d=e=f=g=0;
                        k =(value%1000000000)/100000000;
                        h =(value%100000000)/10000000;
                        g =(value%10000000)/1000000;
                        f = (value%1000000)/100000;
                        e = (value%100000)/10000;
                        d = (value%10000)/1000;
                        c = (value%1000)/100;
                        b = (value%100)/10;
                        a = value%10;
if(k==0)
            {k=16;
if(h==0)
            {h=16;
if(g==0)
            {g=16;
if(f==0)
            {f=16;
```

```
if(e==0)
        {e=16;
if(d==0)
        {d=16;
                                                }
                                                    }
                                                     }
                                                }
                                            }
                                        }
        inti_lcd();
        lcd_comm(0x80);
        for(i=0;i<3;i++)
{lcd_display(vol[i]);}
 lcd_comm(0xC4);
 lcd_display(number[a]);
 lcd_comm(0xC3);
 lcd_display(number[b]);
 lcd_comm(0xC2);
 lcd_display(vol[3]);
 lcd_comm(0xC1);
 lcd_display(number[c]);
 lcd_comm(0xC0);
 lcd_display(number[d]);
 lcd_comm(0x87);
 lcd_display(number[e]);
 lcd_comm(0x86);
 lcd_display(number[f]);
 lcd_comm(0x85);
 lcd_display(number[g]);
```

```
 lcd_comm(0x84);
 lcd_display(number[h]);
 lcd_comm(0x83);
 lcd_display(number[k]);
 lcd_comm(0xC5);
        for(i=0;i<3;i++)
        {lcd_display(num3[i]);}
}
void cubic()
{        unsigned int length=0,width=0,high=0;
        unsigned long value,old_value;
 //*Recieve Length
                show_length();
                length= key_number(1) ;
 //*Recieve Width
                show_width();
                width =key_number(2) ;
//*recieve High
                show_high();
                high =key_number(3) ;
//*Calculate Volume
                while(1)
                        {       value=distance();
                                value=high-value;
                                value=(value*length*width)/10000;
        if(value != old_value)
        {   show_volume(value);
            delay_ms(20);
            old_value = value;
        }
```

```
        }
        }
void cylinder()
        {       float z;
                unsigned int radiance=0,high=0;
                unsigned long value,old_value;
        //Recieve Radiance
                 show_radiance();
                 radiance= key_number(4) ;
        //recieve High
                 show_high();
                 high =key_number(3) ;
        //Calculate Volume
        while(1){       value=distance();
                        value=high-value;
                        z=(radiance*radiance*value)/10000;
                                value=z*22/7;
                if(value != old_value)
          {     show_volume(value);
                delay_ms(20);
                old_value = value;
           }
                }
        }
        void main()
        {       unsigned int key,i;
                 unsigned char start[]="Enter Type";
                 while(1){
                        inti_lcd();
                        lcd_comm(0x80);
```

```
                    for(i=0;i<8;i++)
                            {lcd_display(start[i]);}
                    lcd_comm(0xc0);
                    for(i=8;i<9;i++)
                            {lcd_display(start[i]);}
key=keypad();
                    if(key==12)
                    {           while(1)
                            {cubic();}
                    }
                    else if(key==13)
                    {
                             while(1)
                            {cylinder();}
                    }
                    }
                    else
                    {show_error();
                     delay_ms(10);
                    }
            }
}
```

4.4 การทดสอบเครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิคในถังรูปแบบต่างๆ

เป็นการทดสอบเพื่อวัดปริมาตรระดับน้ำมันที่ทำการทดลองในถังทรงสี่เหลี่ยม และถังรูปทรงกระบอก เพื่อทดสอบประสิทธิภาพเครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิค มีขั้นตอนดังนี้

4.4.1 ต่อชุดอุปกรณ์ต่างๆ

โดยเตรียมชุดอุปกรณ์ให้พร้อมต่อการใช้งาน

4.4.2 ป้อนรูปแบบถังบรรจุน้ำมันและขนาดถังบรรจุน้ำมัน

โดยเลือกรูปแบบถังจากปุ่มที่กำหนด โดยที่ A คือถังรูปทรงสี่เหลี่ยม และ B คือถังรูปทรงกระบอก และขนาดของถังบรรจุน้ำมันที่ป้อนเข้าไปต้องมีหน่วยเป็น มิลลิเมตร

โดยที่ถังรูปทรงสี่เหลี่ยมต้องป้อนค่าขนาดถัง 3ด้านคือ กว้าง ยาว และสูง

และถังรูปทรงกระบอกต้องป้อน 2ค่าคือ รัศมีปากท่อ และสูง

ได้ผลการทดลองตามตารางที่ 4.1 และตารางที่ 4.2 และการทดลองกับน้ำแทนการทดลองกับน้ำมันในตารางที่ 4.3

ตารางที่ 4.1 แสดงผลการทดสอบเครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิคในถังรูปทรงสี่เหลี่ยม

ปริมาตรน้ำมันจริงที่เติมลงถังทดลอง(ลิตร)	ปริมาตรครั้งที่ 1 (ลิตร)	ปริมาตรครั้งที่ 2 (ลิตร)	ปริมาตรครั้งที่ 3 (ลิตร)	ค่าเฉลี่ย (ลิตร)	ค่าความผิดพลาด (%error)
0.50	0.48	0.49	0.48	0.48	4.00
1.00	0.99	1.00	0.98	0.99	1.00
1.50	1.47	1.49	1.47	1.47	1.47
2.00	2.00	2.00	1.98	1.99	0.50
2.50	2.46	2.48	2.46	2.47	1.20
3.00	2.96	2.96	2.96	2.96	1.30
3.50	3.47	3.43	3.47	3.46	1.10
4.00	3.97	3.97	3.97	3.97	0.75
4.50	4.43	4.47	4.47	4.46	0.80
5.00	4.96	4.96	4.96	4.96	0.80
5.50	5.43	5.47	5.45	5.45	0.91
6.00	5.91	5.97	5.95	5.94	1.00

6.50	6.43	6.43	6.43	6.45	0.76
7.00	6.95	6.95	6.95	6.95	0.71
7.50	7.48	7.45	7.47	7.47	0.47
8.00	7.97	7.98	8.00	7.98	0.25

ค่าความผิดพลาดเฉลี่ย = 1.09 %

ตารางที่ 4.2แสดงผลการทดสอบเครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิคในถังรูปทรงกระบอก

ปริมาตรน้ำมันจริงที่เติมลงถังทดลอง(ลิตร)	ปริมาตรครั้งที่ 1 (ลิตร)	ปริมาตรครั้งที่ 2 (ลิตร)	ปริมาตรครั้งที่ 3 (ลิตร)	ค่าเฉลี่ย (ลิตร)	ค่าความผิดพลาด (%error)
0.50	0.50	0.50	0.48	0.49	2.00
1.00	0.99	0.98	0.98	0.98	2.00
1.50	1.47	1.47	1.47	1.47	2.00
2.00	1.98	2.00	1.98	1.98	1.00
2.50	2.48	2.47	2.47	2.47	1.20
3.00	2.98	2.98	2.97	2.98	0.67
3.50	3.46	3.47	3.48	3.47	0.85
4.00	3.95	3.98	3.97	3.97	0.75
4.50	4.41	4.46	4.49	4.45	1.11
5.00	4.98	4.98	4.98	4.98	0.40
5.50	5.47	5.49	5.48	5.48	0.36
6.00	5.95	5.98	5.98	5.97	0.50
6.50	6.47	6.49	6.47	6.48	0.36
7.00	6.97	6.98	6.98	6.98	0.28

ค่าความผิดพลาดเฉลี่ย = 0.96 %

ตารางที่ 4.3แสดงผลการทดสอบเครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิคในถังรูปทรงสี่เหลี่ยม โดยใช้วัดน้ำแทนการวัดน้ำมัน

ปริมาตรน้ำมันจริงที่เติมลงถังทดลอง(ลิตร)	ปริมาตรครั้งที่ 1 (ลิตร)	ปริมาตรครั้งที่ 2 (ลิตร)	ปริมาตรครั้งที่ 3 (ลิตร)	ค่าเฉลี่ย (ลิตร)	ค่าความผิดพลาด (%error)
0.50	0.50	0.50	0.48	0.49	2.00
1.00	0.99	0.98	0.98	0.98	1.00
1.50	1.47	1.47	1.47	1.47	0.67
2.00	1.98	1.97	1.98	1.98	1.00
2.50	2.48	2.47	2.47	2.47	1.20
3.00	2.98	2.98	2.97	2.98	1.00
3.50	3.46	3.47	3.48	3.47	0.86
4.00	3.95	3.98	3.97	3.97	0.50
4.50	4.41	4.46	4.49	4.45	0.44
5.00	4.98	4.98	4.98	4.98	0.80
5.50	5.47	5.49	5.48	5.48	0.91
6.00	5.95	5.98	5.98	5.97	0.17
6.50	6.47	6.79	6.47	6.48	0.15
7.00	6.97	6.98	6.98	6.98	0.29
7.50	7.48	7.48	7.49	7.48	0.27
8.00	7.98	7.99	8.00	7.99	0.12
8.50	8.47	8.51	8.50	8.49	0.12
9.00	8.95	9.00	9.00	8.98	0.22
9.50	9.48	9.50	9.49	9.49	0.11
10.00	9.99	10.00	9.99	9.99	0.10

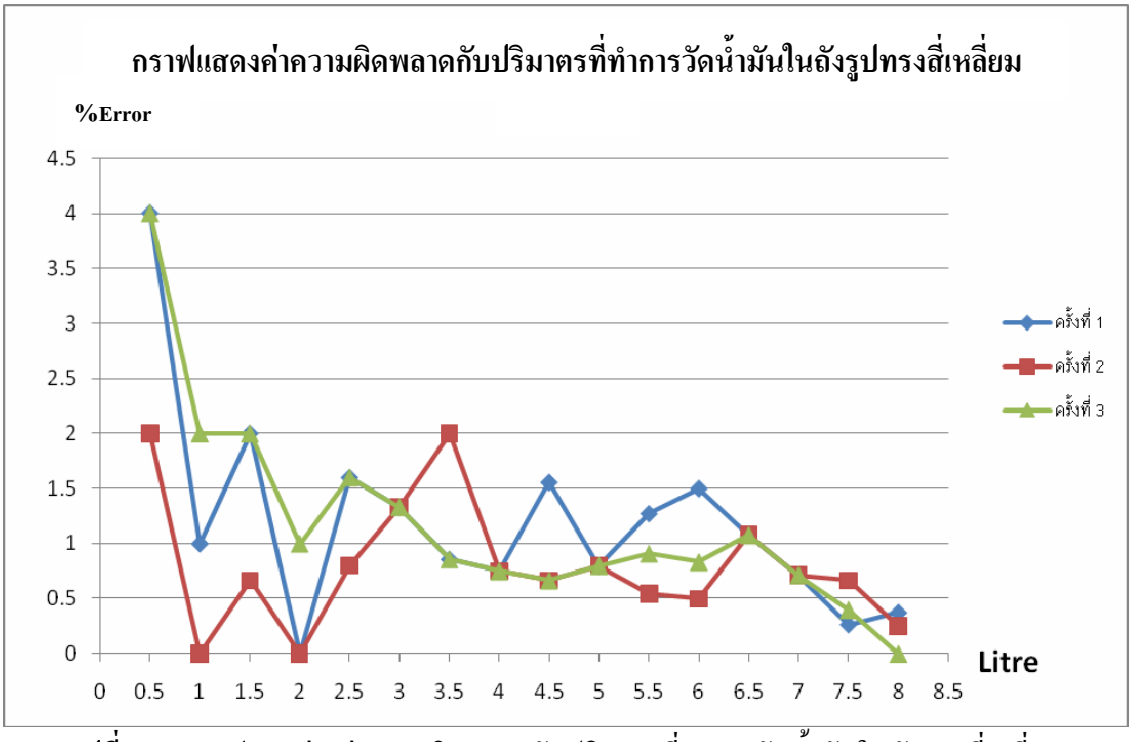
ค่าเฉลี่ยค่าความผิดพลาดรวม = 0.59 %

ตารางที่ 4.4 แสดงผลการทดสอบเครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิคในถังรูปทรงกระบอก โดยใช้วัดน้ำแทนการวัดน้ำมัน

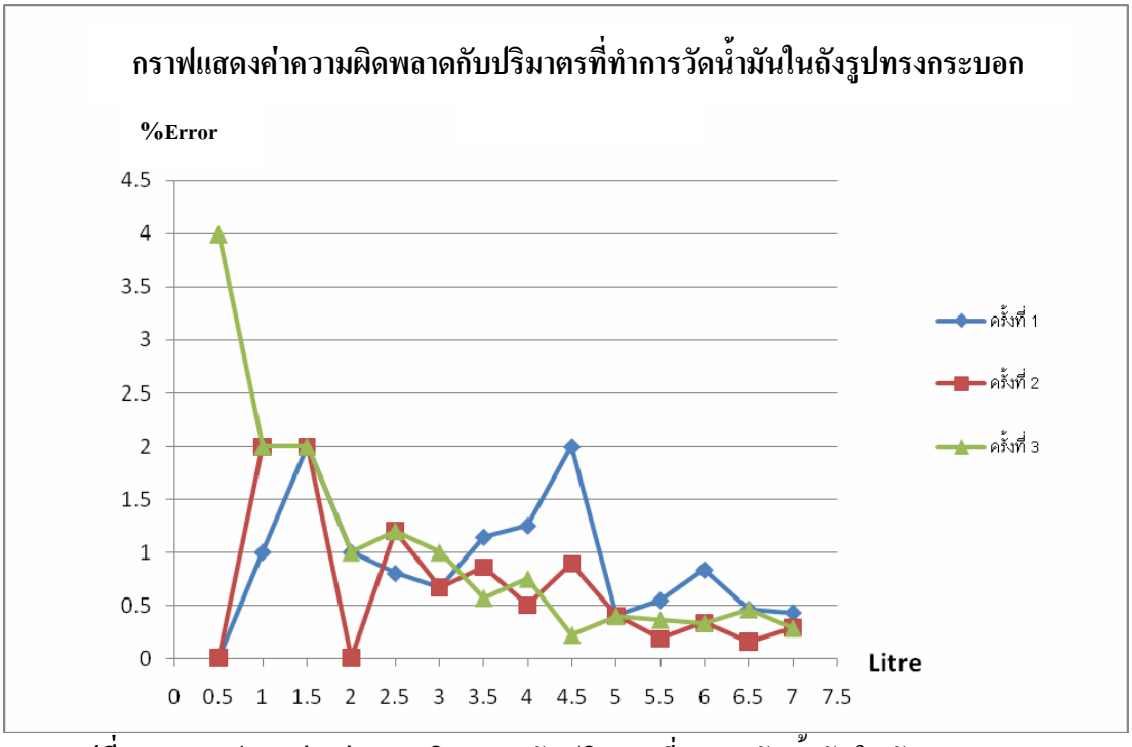
ปริมาตรน้ำมันจริงที่เติมลงถังทดลอง(ลิตร)	ปริมาตรครั้งที่ 1 (ลิตร)	ปริมาตรครั้งที่ 2 (ลิตร)	ปริมาตรครั้งที่ 3 (ลิตร)	ค่าเฉลี่ย (ลิตร)	ค่าความผิดพลาด (%error)
0.50	0.50	0.50	0.50	0.50	0.00
1.00	1.00	1.00	1.00	1.00	0.00
1.50	1.50	1.50	1.49	1.49	0.67
2.00	2.01	2.01	2.00	2.00	0.00
2.50	2.48	2.51	2.50	2.49	0.40
3.00	2.96	2.98	3.00	2.98	0.67
3.50	3.48	3.48	3.49	3.48	0.57
4.00	3.97	4.00	4.00	3.99	0.25
4.50	4.46	4.51	4.50	4.49	0.22
5.00	4.96	5.02	5.00	4.99	0.20
5.50	5.46	5.50	5.50	5.49	0.18
6.00	6.01	5.99	6.00	6.00	0.00
6.50	6.50	6.50	6.50	6.50	0.00
7.00	7.00	7.01	7.00	7.00	0.00

ค่าเฉลี่ยค่าความผิดพลาดรวม = 0.23 %

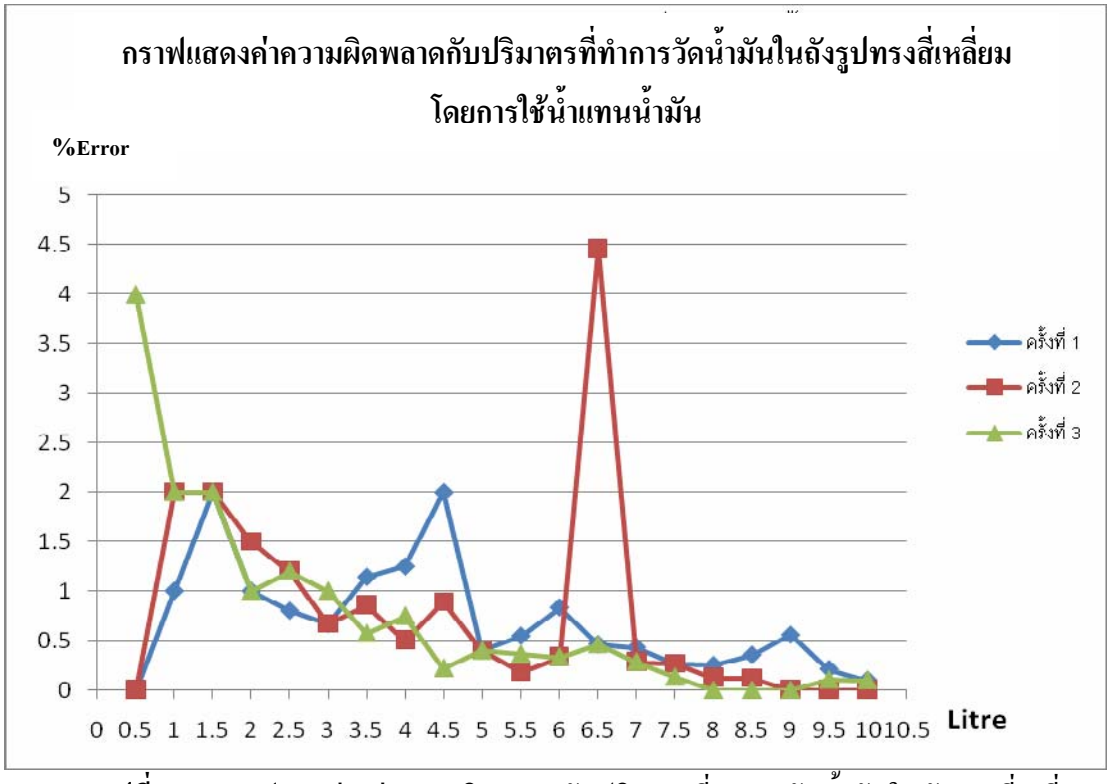
จากผลการทดลอง นำค่าที่ได้มาวาดกราฟระหว่างค่าความผิดพลาดกับปริมาตรที่ทำการวัดในถังทรงสี่เหลี่ยมและถังทรงกระบอก ดังรูปที่ 4.19และรูปที่ 4.20 ตามลำดับดังนี้



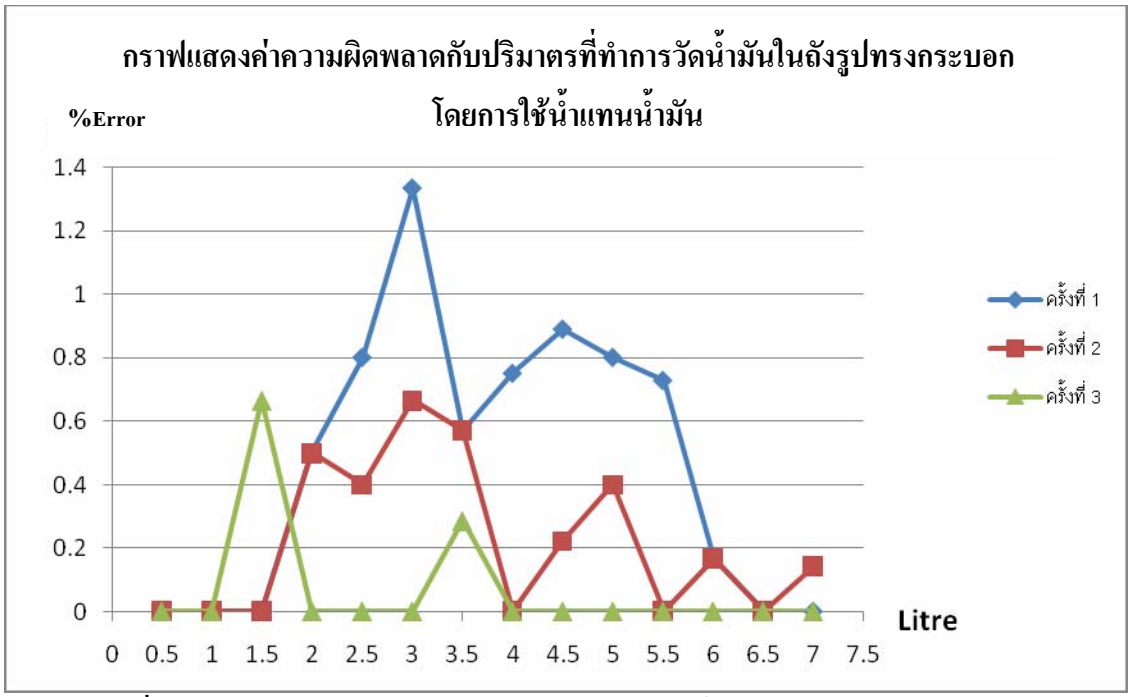
รูปที่ 4.19 กราฟระหว่างค่าความผิดพลาดกับปริมาตรที่ทำการวัดน้ำมันในถังทรงสี่เหลี่ยม



รูปที่ 4.20 กราฟระหว่างค่าความผิดพลาดกับปริมาตรที่ทำการวัดน้ำมันในถังทรงกระบอก



รูปที่ 4.21 กราฟระหว่างค่าความผิดพลาดกับปริมาตรที่ทำการวัดน้ำมันในถังทรงสี่เหลี่ยม โดยการใช้น้ำแทนน้ำมัน



รูปที่ 4.22 กราฟระหว่างค่าความผิดพลาดกับปริมาตรที่ทำการวัด ในถังทรงกระบอก โดยการใช้น้ำแทนน้ำมัน

4.5 สรุปการทดลองเครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิค

เครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิคที่ออกแบบและสร้างขึ้นมานี้ สามารถวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิคที่ส่งคลื่นกระทบผิวของน้ำมันที่คงอยู่ในถัง 2รูปแบบคือ รูปทรงสี่เหลี่ยมและรูปทรงกระบอก แล้วทำการประมวลผลทางไมโครคอนโทรเลอร์ จากการทดสอบซ้ำหลายๆครั้งในถังทั้ง 2รูปแบบ พบว่าในแต่ละครั้งที่ทดสอบจะมีผลที่ใกล้เคียงกันของแต่ละปริมาตรที่ได้ทำการทดสอบ ดังนั้นจึงพบว่าเครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิคนี้ค่อนข้างที่จะมีความเที่ยงตรงในการวัดปริมาตร สามารถนำมาใช้งานในการวัดระดับน้ำมันในถังทั้ง 2 รูปแบบคือ ถังรูปทรงสี่เหลี่ยมและถังรูปทรงกระบอกได้

บทที่ 5
บทสรุป และข้อเสนอแนะ

5.1 กล่าวนำ

ในบทนี้จะกล่าวถึงระบบรวมของเครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิค โดยได้อธิบายปัญหาที่พบในระหว่างการทำโครงงาน วิธีแก้ปัญหา ข้อเสนอแนะ แนวทางการพัฒนาต่อไป และบทสรุปของโครงงานที่จัดทำขึ้น

5.2 ปัญหาที่พบในระหว่างการทำโครงงานและวิธีแก้ปัญหา

ตารางที่ 5.1 แสดงรายละเอียดของปัญหาที่พบ และวิธีแก้ปัญหาของโครงงาน

ปัญหาที่พบ	สาเหตุและแนวทางแก้ไข
1. ปริมาตรน้ำมันที่วัดได้ไม่ตรงกับปริมาตรน้ำมันที่ตวงใส่ลงไปขณะทำการทดลอง	สาเหตุ เนื่องจากขณะที่ทำการทดลองได้มีการสูญเสียน้ำไปบางส่วนเนื่องจากการรวมตัวของน้ำแล้วไปเกาะตัวที่ผนังถ้วยตวงและผนังถังบรรจุน้ำมัน แนวทางแก้ไข พยายามเทน้ำมันลงถังที่ทำการทดลองอย่างระมัดระวัง เพื่อป้องกันน้ำมันเกาะข้างผิวผนังถังน้ำมัน
2. ลายวงจรที่กัดแผ่น Print มีขนาดเล็กมาก	สาเหตุ เนื่องจากลายวงจรที่ใช้ในการทำตัวเชื่อมต่อ USB มีความละเอียดสูง ลายของวงจรจึงมีขนาดเล็กมาก แนวทางแก้ไข ในการออกแบบลายวงจรบนแผ่นวงจรพิมพ์ จะใช้โปรแกรม Protel99SE ที่สามารถทำให้ลายวงจรมีขนาดเล็กและคมตามที่ต้องการ
3. อุปกรณ์เกิดการชำรุดและเสียบ่อย	สาเหตุ เนื่องจากอุปกรณ์อิเล็คทรอนิกส์มีความบอบบางและบางตัวต้องมีการบัดกรี และขา IC สั้นมากทำให้ยากต่อการบัดกรี แล้วการทดลองต้องถอดเข้า-ออกบ่อยจึงเกิดการชำรุดได้ง่าย แนวทางแก้ไข ถ้า IC ตัวใดมีขาที่สั้น จะใช้การเป่าลมร้อนในการบัดกรีแทนการใช้หัวแร้ง และไม่ถอดอุปกรณ์เข้า-ออกบ่อย

5.3 ข้อเสนอแนะ

5.3.1 ลายวงจรบนแผ่นวงจรพิมพ์ที่ออกแบบเสร็จแล้ว ควรที่จะใช้เครื่อง Network วัดเพื่อจะดูว่าอินพุตและเอาท์พุต Matching ที่ 50 โอห์ม หรือไม่

5.3.2 เนื่องจาก IC มีขนาดเล็ก ควรที่จะระมัดระวังในการบัดกรี เพราะเวลาบัดกรีขาของ IC อาจติดกันทำให้เกิดการ Short Circuit ได้ และควรระมัดระวังเรื่องความร้อนในการบัดกรีด้วย เช่น ไม่ควรจี้ขา IC เป็นเวลานานๆ เพราะอาจทำให้ IC พังได้

5.3.3 เนื่องจากเกิดการรวมตัวของน้ำมัน และเกาะตามผนังของถังน้ำมัน ทำให้ปริมาตรเวลาทำการวัดคลาดเคลื่อนได้ จึงควรใช้สารเคลือบผิวภาชนะเพื่อป้องกันการเกาะตัวของน้ำมันที่ผนังของถังน้ำมัน

5.4 แนวทางการพัฒนาต่อไป

5.4.1 ปรับปรุงชุดอุปกรณ์อัลตร้าโซนิคที่สามารถส่ง-รับคลื่นอัลต้าโซนิค ที่มีความสูงมากกว่า 4 เมตรและมีความละเอียดสูงได้

5.4.2 ปรับปรุงชุดกล่องอุปกรณ์เครื่องวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิค ให้สามารถใช้งานโดยไม่ใช้ Adapter เป็นตัวจ่ายแรงดันให้เครื่องทำงานได้ โดยอาจจะใช้ถ่านแบต-เตอรี่แทน

5.4.3 ปรับปรุงชุดอุปกรณ์ที่ใช้ในการส่งข้อมูลจากตัวเซนเซอร์ที่อยู่ถังน้ำมัน ไปยังจอแสดงผลระดับน้ำมันจากที่ใช้สาย เป็นไร้สาย

5.5 บทสรุป

ระบบรวมของเครื่องวัดปริมาตรน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิคประกอบไปด้วย

- การวัดปริมาตรน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิคในถังรูปทรงสี่เหลี่ยม
- การวัดปริมาตรน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิคในถังรูปทรงกระบอก

ก่อนที่จำทำการต่อชุดอุปกรณ์และวงจรต่างๆในระบบ จะทำการทดสอบชุดอุปกรณ์และวงจรข้างต้น ซึ่งได้ผลตามบทที่ 3 และบทที่ 4

จากการทดสอบเมื่อทำการต่อชุดอุปกรณ์ต่างๆในระบบรวมทั้งหมด แล้วทำการทดลองการววัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิค สามารถวัดระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซ-นิคที่ส่งคลื่นกระทบผิวสูงสุดของระดับน้ำมันที่คงอยู่ในถัง 2 รูปแบบคือ รูปทรงสี่เหลี่ยมและรูปทรงกระบอก แล้วทำการประมวลผลทางไมโครคอนโทรเลอร์ เมื่อทำการทดสอบได้ทำการทดสอบซ้ำหลายๆรอบในถังทั้ง 2 รูปแบบ พบว่าในแต่ละครั้งที่ได้ทำการทดสอบ และเมื่อผลการทดสอบออกมามีผลที่คล้ายคลึงกันในแต่ละค่าที่ได้ทำการทดสอบในแต่ละครั้ง แสดงว่าเครื่องวัด

ระดับน้ำมันโดยอาศัยคลื่นอัลตร้าโซนิคนี้มีความเที่ยงตรงและแม่นยำ สามารถนำมาใช้งานได้จริง ในการวัดระดับน้ำมันในถังทั้ง 2 รูปแบบคือ ถังทรงสี่เหลี่ยมและถังทรงกระบอกได้

บรรณานุกรม

[1] สันติ นุราชและคณะ. , **“เรียนรู้ไมโครคอนโทรลเลอร์ MCS-51 ฉบับภาษา C”**, Micro Research Technology. Ltd, Part, ปทุมธานี.

[2] http://fivedots.coe.psu.ac.th/Software.coe/LAB/Lab_y3/ULTRASO.DOC

[3] http://www.ftdichip.com/Documents/DataSheets.htm

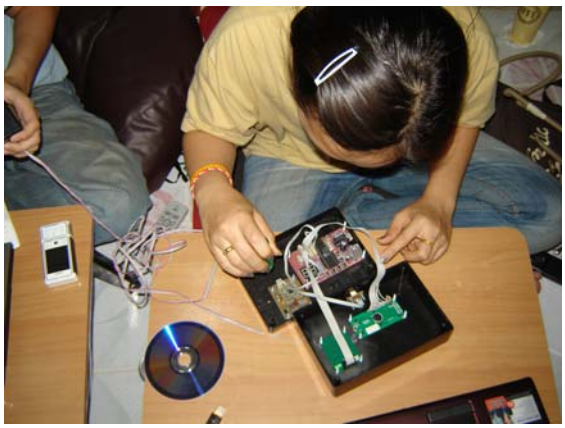
[4] ขจร อนุดิตย์ “**การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ MCS-51 ด้วยภาษาซี**.” พิมพ์ครั้งที่ 1, เพียร์สัน เอ็ดดูเคชั่น อินโดไชน่า, กรุงเทพฯ: 2547

ภาคผนวก (ก)

Datasheet

ภาคผนวก (ข)

รูปในการปฏิบัติโครงงาน



นำวงจรจ่างๆต่อลงในกล่อ

ต่อเชื่อมวงจรต่างๆในกล่องโดยการบัดกรี

ทดลองจริงกับถังรูปทรงต่างๆ

ประวัติผู้จัดทำ

นายธนวัฒน์ ควรรติกุล เกิดเมื่อวันที่ 24 ธันวาคม พ.ศ. 2528 ภูมิลำเนาเดิมอยู่บ้านเลขที่ 43 ถนนมหาชัยดำริห์ ตำบลตลาด อำเภอเมืองมหาสารคาม จังหวัดมหาสารคาม สำเร็จการศึกษาระดับชั้นมัธยมศึกษาตอนปลายจากโรงเรียนสารคามพิทยาคม อำเภอเมืองมหาสารคาม จังหวัดมหาสารคาม ปัจจุบันกำลังศึกษาอยู่ชั้นปีที่ 5 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา

นายบูรนิติ มาศอมรพันธุ์ เกิดเมื่อวันที่ 19 มกราคม พ.ศ. 2528 ภูมิลำเนาเดิมอยู่บ้านเลขที่ 264/108 หมู่ที่ 13 ถนนเทพารักษ์ ตำบลในเมือง อำเภอเมืองขอนแก่น จังหวัดขอนแก่น สำเร็จการศึกษาระดับชั้นมัธยมศึกษาตอนปลายจากโรงเรียนขอนแก่นวิทยายน อำเภอเมืองขอนแก่น จังหวัดขอนแก่น ปัจจุบันกำลังศึกษาอยู่ชั้นปีที่ 5 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา

นางสาวรัชดาพร มานุวงศ์ เกิดเมื่อวันที่ 19 มีนาคม พ.ศ. 2528 ภูมิลำเนาเดิมอยู่บ้านเลขที่ 14/26 ถนนสระหลวง ตำบลในเมือง อำเภอเมืองพิจิตร จังหวัดพิจิตร สำเร็จการศึกษาระดับชั้นมัธยมศึกษาตอนปลายจากโรงเรียนพิจิตรพิทยาคม อำเภอเมืองพิจิตร จังหวัดพิจิตร ปัจจุบันกำลังศึกษาอยู่ชั้นปีที่ 5 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา