



โปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์

โดย

นางสาวนัฐญา วิคุณละพัน B4701798

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา 427499 โครงการงานศึกษาวิศวกรรมโทรคมนาคม

ประจำภาคการศึกษาที่ 1 ปีการศึกษา 2550

หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม

หลักสูตรปรับปรุงพ.ศ. 2545

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

โครงการ โปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์
ผู้ดำเนินงาน นางสาวนัฐญา วิคุณละพัน B4701798
อาจารย์ที่ปรึกษา ผศ.ดร.รังสรรค์ วงศ์สรรคค์
สาขาวิชา วิศวกรรมโทรคมนาคม
ภาคการศึกษาที่ 1/2550

บทคัดย่อ

โครงการนี้ได้ออกแบบจัดทำโปรแกรมคอมพิวเตอร์สำหรับช่วยสอนปฏิบัติการอิเล็กทรอนิกส์ เป็นการรับข้อมูลแล้วส่งผ่านข้อมูลไปแสดงผลยังจอคอมพิวเตอร์ สามารถแบ่งออกได้เป็น 2 ส่วนคือวงจรเชื่อมต่อระหว่างแบบตัวอย่างของปฏิบัติการทางไฟฟ้ากับเครื่องคอมพิวเตอร์ และ โปรแกรมควบคุม โดยวงจรเชื่อมต่อนี้จะเป็นส่วนที่ทำหน้าที่ในการรับคำสั่งสัญญาณพื้นฐานที่ได้จากการปฏิบัติการซึ่งเป็นคำสั่งสัญญาณในรูปแบบอนาลอก และทำการแปลงคำสั่งสัญญาณให้อยู่ในรูปแบบดิจิทัล เพื่อที่จะสามารถใช้เชื่อมต่อกับเครื่องคอมพิวเตอร์ได้ โดยในโครงการนี้จะได้ออกแบบตัวอย่างของปฏิบัติการทางไฟฟ้า เพื่อนำมาทดสอบการใช้งานของวงจรเชื่อมต่อที่ได้ออกแบบสร้างนี้ ในส่วนของโปรแกรมควบคุม จะมีการเขียนโปรแกรมควบคุมทางคอมพิวเตอร์ เพื่อแสดงผลที่ได้จากการวัดด้วยโปรแกรม Microsoft Visual Basic

กิตติกรรมประกาศ

ในการจัดทำโครงการนี้จะมีอาจสำเร็จลุล่วงไปได้ด้วยดี ถ้าหากมิได้รับความกรุณาจากอาจารย์ที่ปรึกษา ผศ.ดร.รังสรรค์ วงศ์สรรคร์ ผู้ที่ให้แนวคิดแรกเริ่มของโปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการ ที่ให้ความช่วยเหลือในเรื่องของแนวคิด การติดตามเอาใจใส่ และชี้แนะข้อบกพร่องที่ข้าพเจ้าได้มองข้ามไปในบางส่วน

นอกจากนี้ยังต้องขอขอบพระคุณผู้ที่เกี่ยวข้อง ดังนี้

คณาจารย์ประจำสาขาวิชาวิศวกรรมโทรคมนาคมทุกท่าน ที่ได้สั่งสอนให้ความรู้

คุณประพล จาระตะคุ ผู้ที่ให้ความช่วยเหลือในเรื่องของการเบิกจ่ายงบประมาณ ตลอดจนอุปกรณ์ในการทดลองต่างๆ

คุณมณีรัตน์ ทุมพงษ์ เลขานุการประจำสาขาวิชาวิศวกรรมโทรคมนาคม ที่ให้ความช่วยเหลือในเรื่องของงานเอกสาร

พี่ๆ สาขาวิชาวิศวกรรมโทรคมนาคม ที่ให้ความช่วยเหลือในเรื่องของการเขียนโปรแกรม และการทดสอบวงจรเชื่อมต่อคอมพิวเตอร์ช่วยสอนปฏิบัติการ

สุดท้ายนี้ คุณงานความดีที่เกิดจากโครงการฉบับนี้ ขอมอบแก่บิดามารดา ผู้ที่คอยห่วงใย ให้กำลังใจ ให้โอกาส และให้การสนับสนุนทางการศึกษามาโดยตลอด

นางสาวนัฐญา วิคุณละพัน

สารบัญ

	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญรูป	ฉ
สารบัญตาราง	ซ
บทที่ 1 บทนำ	1
1.1 ความเป็นมา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของการทำงาน	1
1.4 ขั้นตอนการดำเนินงาน	1
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 ไมโครซอฟท์วิซวลเบสิก 6.0 (Microsoft Visual Basic 6.0)	3
2.1.1 ลักษณะโครงสร้างพื้นฐานการใช้งาน Microsoft Visual Basic 6.0	3
2.2 พื้นฐานการสื่อสารแบบอนุกรม	11
2.2.1 องค์ประกอบของการรับส่งข้อมูลแบบอนุกรม	13
2.2.2 การเขียนโปรแกรมติดต่อและควบคุม Serial Port กับ Visual Basic 6.0	14
2.3 วงจรแบ่งแรงดัน	17
2.4 วงจรแบ่งกระแส	19
2.5 สรุป	20

สารบัญ

	หน้า
บทที่ 3 การออกแบบและการเขียนโปรแกรมคอมพิวเตอร์ช่วยสอน	21
ปฏิบัติการอิเล็กทรอนิกส์	
3.1 ขั้นตอนการทำงานของโปรแกรมประมวลผลและแสดงผล	21
3.1.1 กำหนดและออกแบบขั้นตอนของโปรแกรม	21
3.2 การออกแบบรูปแบบของโปรแกรม	23
3.3 การเขียนโปรแกรม Microsoft Visual Basic 6.0	28
3.3.1 การเขียนโค้ดโปรแกรมเพื่อควบคุมฟอร์มของโปรแกรม	28
3.3.2 อธิบายการทำงานของโปรแกรม	33
3.4 สรุปผลการออกแบบโปรแกรม	34
บทที่ 4 การทดสอบโปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการ	35
4.1 การทดสอบโปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการ	35
4.1.1 การติดตั้งชุดอุปกรณ์เพื่อทดสอบโปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการ	35
4.1.2 การเปิดใช้งานโปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการ	35
4.1.3 การทดสอบโปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการ	36
4.2 สรุปผลการทดสอบโปรแกรม	41
บทที่ 5 สรุปและข้อเสนอแนะ	42
5.1 ขั้นตอนการทำงานของโปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการ	42
5.2 ปัญหาที่พบในขั้นตอนการทำงาน	42
5.3 ข้อจำกัดของโครงการ	42
5.4 ผลที่ได้จากโครงการ	43
5.5 แนวทางการพัฒนาต่อ	43

สารบัญ

	หน้า
บรรณานุกรม	44
ภาคผนวก (ก)	45
การโหลดโปรแกรมทดสอบอุปกรณ์เชื่อมต่อและการติดตั้งไดรเวอร์ USB	46
ภาคผนวก (ข)	53
โปรแกรมควบคุมและสั่งการให้แสดงผลผ่านทางหน้าจอคอมพิวเตอร์ของ “วงจรเชื่อมต่อภายนอก”	54
โปรแกรมควบคุมการทำงานของวงจรแปลงสัญญาณอนาลอกเป็นดิจิทัล	57
ภาคผนวก (ค)	60
วงจรมอนิเตอร์สัญญาณเชื่อมต่อภายนอก	61
วงจรแปลงสัญญาณอนาลอกเป็นดิจิทัล	61
วงจรมอนิเตอร์คอนโทรลเลอร์	63
วงจรเชื่อมต่อ USB	64
ประวัติผู้จัดทำ	68

สารบัญรูป

	หน้า
2.1 กรอบโต้ตอบเมื่อเริ่มเปิด Visual Basic	3
2.2 หน้าต่างของ Visual Basic เมื่อเริ่มโปรแกรม	5
2.3 Toolbars	6
2.4 Toolboxes	6
2.5 แสดงรายการคอนโทรล ActiveX เพิ่มเติม	7
2.6 Form Designer	8
2.7 Project Explorer แบบ โปรเจกต์เดียวและแบบหลายโปรเจกต์	8
2.8 Properties Window	9
2.9 Form Layout	10
2.10 Immediate Window	10
2.11 กรอบโต้ตอบ New Project	10
2.12 Code Editor	11
2.13 แสดงลักษณะของคอนเน็คเตอร์	12
2.14 เพิ่ม Components MSComm	15
2.15 เลือกที่รายการ MSComm	15
2.16 คอนโทรล MSComm พร้อมใช้งาน	16
2.17 (ก) วงจรแบ่งแรงดัน	18
2.17 (ข) วงจรแบ่งแรงดันซึ่งกำหนดค่ากระแส i	18
2.18 วงจรแบ่งแรงดันที่ต่อกับโหลด R_L	19
2.19 วงจรแบ่งกระแส	20
3.1 ภาพโครงสร้างโดยรวมของโปรแกรม	21
3.2 ภาพแสดงขั้นตอนการทำงานของโปรแกรม	22
3.3 ภาพแสดงรูปแบบของโปรแกรม	23
4.1 ชุดอุปกรณ์ที่ใช้ในการทดสอบ โปรแกรม	35
4.2 แสดงโปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการบนหน้าต่าง โปรแกรม Visual Basic	36
4.3 ผลการทดสอบวงจรแบ่งแรงดัน โดยใช้ค่า $R_2 = 1k\Omega$	37

สารบัญรูป

	หน้า
4.4 ผลการทดสอบวงจรแบ่งแรงดันโดยใช้ค่า $R_2 = 6.7k\Omega$	37
4.5 ผลการทดสอบวงจรแบ่งแรงดันโดยใช้ค่า $R_2 = 10k\Omega$	38
4.6 ผลการทดสอบวงจรแบ่งแรงดันกระแสโดยใช้ค่า $R_3 = 1k\Omega$	38
4.7 ผลการทดสอบวงจรแบ่งแรงดันกระแสโดยใช้ค่า $R_3 = 6.7k\Omega$	39
4.8 ผลการทดสอบวงจรแบ่งแรงดันกระแสโดยใช้ค่า $R_3 = 10k\Omega$	39

สารบัญตาราง

	หน้า
2.1 แสดงประเภทของโปรแกรมประยุกต์ที่มีให้เลือกใช้ของโปรแกรม Visual Basic	4
2.2 แสดงส่วนประกอบของโปรเจกต์	8
2.3 แสดงรายละเอียดของสายสัญญาณ	12
3.1 แสดงการกำหนดค่า Property ให้กับคอนโทรลต่างๆของโปรแกรม	23
3.2 แสดงส่วนของโค้ดโปรแกรม	29
4.1 เปรียบเทียบระหว่างผลการทดสอบและผลการคำนวณของวงจรแบ่งแรงดัน	40
4.2 เปรียบเทียบระหว่างผลการทดสอบและผลการคำนวณของวงจรแบ่งกระแส	40

บทที่ 1

บทนำ

1.1 ความเป็นมา

ในปัจจุบันการใช้งานเครื่องคอมพิวเตอร์มีบทบาทสำคัญในการใช้งานในห้องปฏิบัติการมากขึ้นกว่าในสมัยก่อน และเครื่องมือที่ใช้ในการทดลองหรือวัดค่าต่าง ๆ มีราคาที่ย่อมเยาในบางประเภท และความนิยมในการใช้งานเครื่องมือที่มีการติดต่อกับคอมพิวเตอร์ผ่านทางพอร์ต USB มีความแพร่หลาย ดังนั้นโครงการที่จัดทำขึ้นนี้ก็เพื่อเป็นการสร้างอุปกรณ์ ที่ช่วยในการเรียนรู้ในห้องปฏิบัติการให้ทำการแสดงผลที่ได้ออกมาผ่านทางหน้าจอคอมพิวเตอร์ และทำการเขียนโปรแกรมในส่วนควบคุมและแสดงผลโดยใช้โปรแกรม Microsoft Visual Basic โดยติดต่อกับอุปกรณ์ผ่านทางพอร์ต USB

1.2 วัตถุประสงค์

- 1.2.1 เพื่อศึกษาวางจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล
- 1.2.2 เพื่อศึกษาวางจรเชื่อมต่อคอมพิวเตอร์จาก RS-232 ไปเป็น USB
- 1.2.3 เพื่อพัฒนาโปรแกรมคอมพิวเตอร์ให้ทำการประมวลผล
- 1.2.4 เพื่อพัฒนาโปรแกรมแสดงผลด้วยโปรแกรม Microsoft Visual Basic

1.3 ขอบเขตของการทำงาน

- 1.3.1 ศึกษาค้นหาข้อมูลเกี่ยวกับพอร์ต USB, การแปลงสัญญาณอนาลอกเป็น สัญญาณดิจิทัล
- 1.3.2 ศึกษาการเขียน โปรแกรม Microsoft Visual Basic
- 1.3.3 เขียนโปรแกรมประมวลผลค่าการวัดเพื่อใช้แสดงผลผ่านหน้าจอคอมพิวเตอร์
- 1.3.4 นำอุปกรณ์ต้นแบบทั้งหมดมาทำการทดสอบเพื่อให้ได้ตามวัตถุประสงค์

1.4 ขั้นตอนการดำเนินงาน

- 1.4.1 ศึกษาและทำการค้นหาเกี่ยวกับการทำงานของพอร์ต USB และการแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล
- 1.4.2 ศึกษาการเขียน โปรแกรม Microsoft Visual Basic

- 1.4.3 จัดเตรียมวัสดุและอุปกรณ์ต่างๆ
- 1.4.4 เขียนโปรแกรมเพื่อให้ไมโครคอนโทรลเลอร์สั่งการให้วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลทำงานและเกิดการเชื่อมต่อข้อมูลกับคอมพิวเตอร์ผ่านทางพอร์ต USB
- 1.4.5 ทดสอบและแก้ไขการทำงานของระบบที่ได้ออกแบบ
- 1.4.6 สรุปผลการทดลองและประเมินผลและจัดทำรูปเล่มรายงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 ได้เรียนรู้วิธีหาความรู้ด้วยตัวเองเพื่อนำมาปฏิบัติและประยุกต์ใช้งานจริง
- 1.5.2 ได้เรียนรู้วิธีและกระบวนการแก้ปัญหาต่างๆที่เกิดขึ้นอย่างมีประสิทธิภาพ
- 1.5.3 สามารถพัฒนาทักษะการใช้โปรแกรมที่เกี่ยวกับโครงงานให้มีความชำนาญขึ้น
- 1.5.4 ได้รับความรู้มากขึ้นในทางปฏิบัติหลังจากทำการศึกษาทางทฤษฎีมาแล้ว
- 1.5.5 สามารถค้นคว้าหาความรู้เพิ่มเติมนอกเหนือจากหลักสูตรที่เรียนได้
- 1.5.6 ได้บรรลุเป้าหมายตามวัตถุประสงค์ที่วางไว้ทั้งหมด

บทที่ 2

ทฤษฎีและหลักการ

2.1 ไมโครซอฟท์วิซวลเบสิก 6.0 (Microsoft Visual Basic 6.0) [1]

โปรแกรมไมโครซอฟท์วิซวลเบสิก 6.0 เป็นโปรแกรมที่ใช้สำหรับสร้างหรือพัฒนาโปรแกรมใช้งานบนวินโดวส์โดยใช้ภาษา Visual Basic กับระบบปฏิบัติการ Windows ของบริษัทไมโครซอฟท์ โดยภาษา Visual Basic พัฒนามาจากภาษา BASIC (Beginner's All-purpose Symbolic Instruction Code) ซึ่งเป็นภาษาคอมพิวเตอร์ระดับสูง ใช้งานง่าย เหมาะสำหรับผู้เริ่มใช้คอมพิวเตอร์เพราะใช้คำในภาษาอังกฤษที่เข้าใจง่าย และเมื่อเป็น Visual Basic ซึ่งใช้ลักษณะของการมองเห็นได้ (Visual) ที่เป็นการติดต่อกับผู้ใช้ด้วยกราฟฟิกหรือรูปภาพ (Graphic User Interface - GUI) จึงทำให้การพัฒนาโปรแกรมใช้งานทำได้สะดวก รวดเร็ว และมีความสามารถสูง เหมาะสำหรับการพัฒนาโปรแกรมใช้งานหลายด้าน เช่นงานด้านคำนวณทั่วไป งานด้านฐานข้อมูล เกมส์ และยังสามารถทำการเชื่อมต่อกับอุปกรณ์ภายนอกหรือที่เรียกว่า ฮาร์ดแวร์อินเทอร์เน็ตเฟสได้อีกด้วย

2.1.1 ลักษณะโครงสร้างพื้นฐานการใช้งาน Microsoft Visual Basic 6.0



เมื่อเข้าสู่โปรแกรม Visual Basic จะแสดงกรอบโต้ตอบสำหรับเลือกชนิดของโปรแกรมประยุกต์ ที่ต้องการ



รูปที่ 2.1 กรอบโต้ตอบเมื่อเริ่มเปิด Visual Basic

ตารางที่ 2.1 แสดงประเภทของโปรแกรมประยุกต์ที่มีให้เลือกใช้ของโปรแกรม Visual Basic6.0

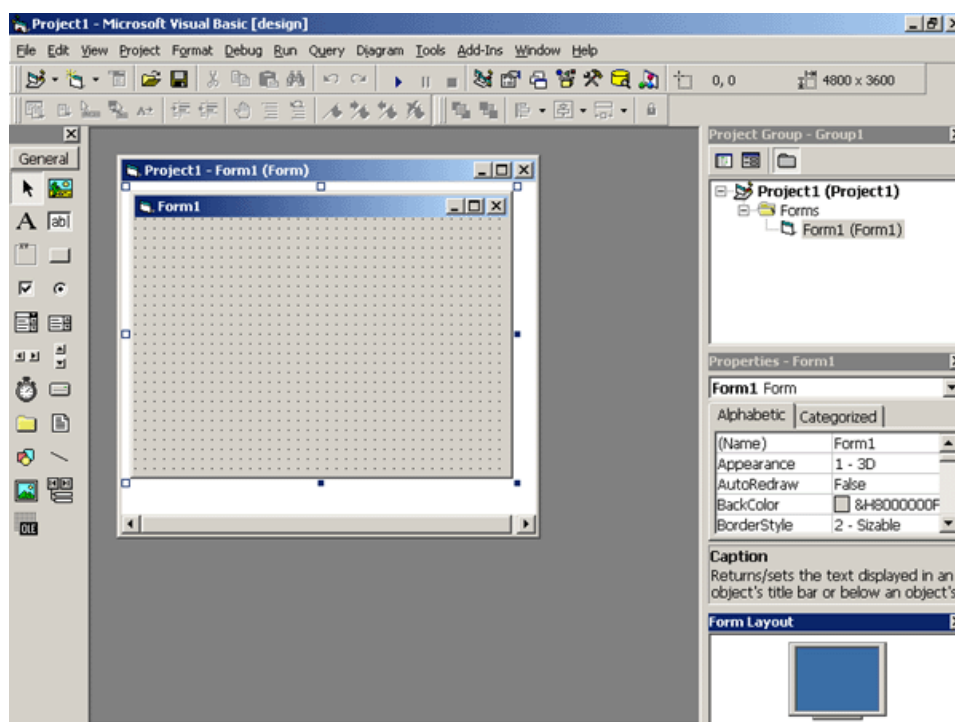
 Standard EXE	ใช้พัฒนาโปรแกรมประยุกต์ทั่ว ๆ ไป
 ActiveX EXE	ใช้พัฒนาโปรแกรมประยุกต์ที่สามารถใช้งานและเชื่อมโยงกับโปรแกรมประยุกต์อื่น ๆ ที่สนับสนุนเทคโนโลยี ActiveX
 ActiveX DLL	เป็นโปรแกรมประยุกต์ชนิดเดียวกับ ActiveX.EXE แต่จะเก็บเป็นไฟล์ไบนารีที่ไม่สามารถประมวลผลได้ด้วยตัวเอง จะต้องถูกเรียกใช้งานจากโปรแกรมประยุกต์อื่น ๆ
 ActiveX Control	ใช้สร้างคอนโทรล ActiveX ขึ้นมาใช้งานเอง
 VB Application Wizard	เป็นเครื่องมือที่ช่วยให้สามารถสร้างโปรแกรมประยุกต์ได้อย่างรวดเร็ว โดยจะสร้างองค์ประกอบเบื้องต้นหลัก ๆ ของโปรแกรมประยุกต์ จากขั้นตอนที่ได้เลือกไว้
 VB Wizard Manager	ใช้สำหรับสร้างโปรแกรมการจัดการต่าง ๆ เช่น การติดต่อกับฐานข้อมูล เป็นต้น
 Data Project	เป็นชนิดโปรเจกต์ที่เป็นแบบฟอร์ม เพื่อติดต่อกับฐานข้อมูลโดยผ่านทางคอนโทรล ADO Data Control
 IIS Application	โปรแกรมประยุกต์ชนิดที่ใช้กับ Web Server
 Addin	ใช้สำหรับเพิ่มเติม utility เข้าไปใน Visual Basic เพื่อเพิ่มประสิทธิภาพ
 ActiveX Document DLL	ใช้สร้างโปรแกรมประยุกต์ที่ประมวลผลบน Internet จะเก็บอยู่ในรูปไฟล์ .dll ไม่สามารถประมวลผลได้ด้วยตัวเอง ต้องให้โปรแกรมประยุกต์อื่น ๆ ที่สนับสนุนเทคโนโลยี ActiveX เรียกใช้งาน เช่น Internet Explorer เป็นต้น
 ActiveX Document Exe	ใช้สร้างโปรแกรมประยุกต์ชนิดที่ประมวลผลบน Internet เช่นกัน แต่จะเก็บอยู่ในรูปไฟล์ .exe สามารถประมวลผลได้ด้วยตัวเอง แต่ server จะต้องสนับสนุนเทคโนโลยี ActiveX ด้วยเช่นกัน เช่น Internet Explorer เป็นต้น

 DHTML Application	ใช้พัฒนาโปรแกรมประยุกต์รูปแบบของเอกสาร Dynamic HTML ซึ่งจะ เป็นมาตรฐานใหม่ของการแสดงผลบน web
 VB Enterprise Edition Controls	ใช้สำหรับโหลด Visual Basic ในรูปแบบที่ใช้พัฒนาโปรแกรมประยุกต์ใน ระดับ Enterprise ซึ่ง Visual Basic จะเพิ่มคอนโทรล ActiveX อีกจำนวน หนึ่งขึ้นมาโดยอัตโนมัติ

สำหรับ แท็บ Existing ใช้สำหรับเปิดโปรเจกต์ที่คุณมีอยู่แล้ว แต่ยังไม่เคยเปิดใช้

แท็บ Recent จะแสดงรายชื่อโปรเจกต์ที่เคยเรียกใช้แล้ว

เมื่อเลือกชนิดของโปรแกรมประยุกต์เป็นแบบ Standard EXE จะเข้าสู่หน้าต่างของ Visual Basic ดังรูปที่ 1- 2



รูปที่ 2.2 หน้าต่างของ Visual Basic เมื่อเริ่มโปรแกรม

ในแต่ละส่วนของ Visual Basic จะมีหน้าที่แตกต่างกันไป ซึ่งในระหว่างการพัฒนาโปรแกรมประยุกต์ จะต้องใช้ส่วนต่าง ๆ เหล่านี้ ในการพัฒนาโปรแกรมประยุกต์

- ทูลบาร์ (Toolbars)

เป็นแถบสัญลักษณ์ที่ใช้สำหรับเข้าถึงชุดคำสั่งของ Visual Basic ได้ทันที โดยจะนำคำสั่งที่ถูกใช้งานบ่อย ๆ มาแสดง



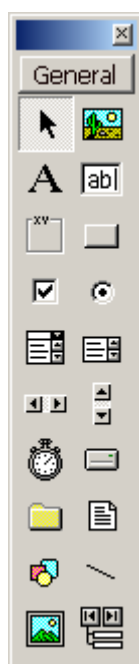
รูปที่ 2.3 Toolbars

ทูลบาร์สามารถแบ่งออกได้เป็น 4 กลุ่มใหญ่ ๆ คือ

1. Standard Toolbars เป็นทูลบาร์มาตรฐานประกอบด้วยคำสั่งที่เกี่ยวข้องกับการจัดการ Project
2. Edit Toolbars เป็นทูลบาร์ที่ประกอบไปด้วยคำสั่งที่ใช้สำหรับช่วยในการเขียนโค้ดใน code editor
3. Debug Toolbars เป็นทูลบาร์ที่ประกอบไปด้วยคำสั่งที่ใช้สำหรับตรวจสอบการทำงานของโปรแกรม
4. Form Editor Toolbars เป็นทูลบาร์ที่ประกอบไปด้วยคำสั่งที่ใช้สำหรับช่วยในการปรับขนาด, ย้าย, เปลี่ยนตำแหน่งคอนโทรลต่าง ๆ ที่อยู่บนฟอร์ม - ทูลบอกส์ (Toolboxes)

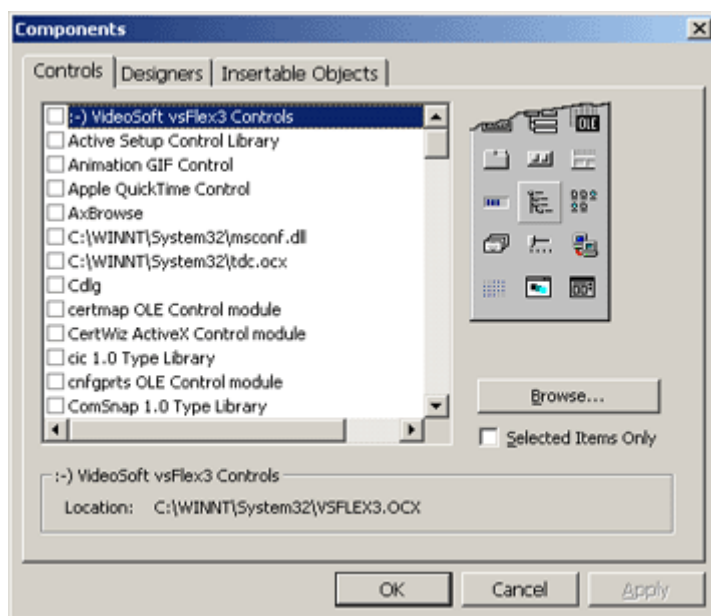
คือแถบสัญลักษณ์ Controls ต่าง ๆ ที่ใช้สำหรับพัฒนาโปรแกรมประยุกต์ แบ่ง เป็น 2 กลุ่ม คือ

1. คอนโทรลภายใน (Intrinsic controls) เป็นชุดคอนโทรลมาตรฐานของ Visual Basic ทุก ๆ ครั้งที่มีการเรียกใช้ Form เพื่อสร้างโปรแกรมประยุกต์ คอนโทรลชุดนี้จะถูกเรียกขึ้นมาอัตโนมัติ สามารถเลือกใช้งานคอนโทรลกลุ่มนี้ได้ทันที



รูปที่ 2.4 Toolboxes

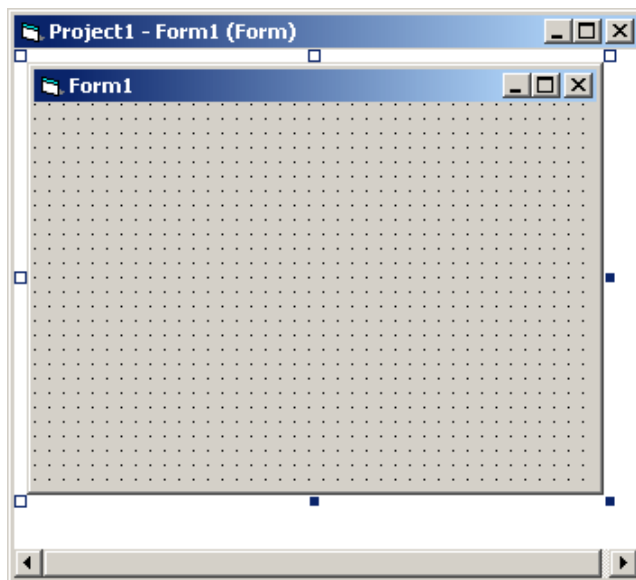
2. คอนโทรล ActiveX (ActiveX controls) เป็นชุดคอนโทรลเพิ่มเติมที่ไม่โครซอฟท์จัดเตรียมไว้ เพื่อเพิ่มขีดความสามารถในการพัฒนาโปรแกรมประยุกต์ การเพิ่มคอนโทรลกลุ่มนี้เข้ามาในทูลบ็อกซ์ทำได้โดยเลือกเมนู Project/Components (หรือคลิกขวาตรงแถบทูลบ็อกซ์เลือกคำสั่ง)



รูปที่ 2.5 แสดงรายการคอนโทรล ActiveX เพิ่มเติม

- Form Designer

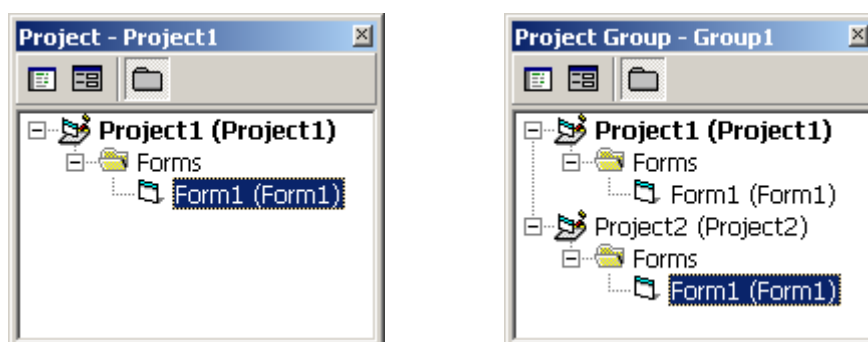
เป็นส่วนที่ใช้ออกแบบการแสดงผลส่วนที่ใช้ติดต่อกับผู้ใช้ ฟอร์มเป็นออบเจกต์แรกที่ถูกเตรียมไว้ให้ใช้งาน คอลโทรลทุกตัวที่ต้องการใช้งานจะต้องนำไปบรรจุไว้ในฟอร์ม นำคอลโทรลมาประกอบกันขึ้นเป็นโปรแกรมประยุกต์ ทุกครั้งที่เปิด Visual Basic ขึ้นมา หรือ สร้าง Project ใหม่ จะมีฟอร์มว่าง 1 ฟอร์มถูกสร้างเตรียมไว้เสมอ



รูปที่ 2.6 Form Designer

- Project Explorer

Project Explorer ใช้สำหรับบริหารและจัดการโปรเจกต์ โดยจะแสดงองค์ประกอบของแต่ละโปรเจกต์แบบโครงสร้างต้นไม้ (tree-view) ตัวโปรเจกต์จะหมายถึงโปรแกรมประยุกต์ซึ่งจะอยู่ส่วนบนสุด ถัดมา จะแสดงส่วนประกอบต่าง ๆ ของโปรเจกต์นั้น ๆ ว่าประกอบด้วยอะไรบ้าง เช่น ฟอर्मโมดูล รายงาน เป็นต้น ถ้ามี 2 โปรเจกต์ขึ้นไป ก็จะแสดงแยกออกเป็นส่วนตัวต่างหากอีกโปรเจกต์ ถ้าต้องการใช้งานส่วนใด ของโปรเจกต์ไหนก็สามารถคลิกเลือกได้ทันที



รูปที่ 2.7 Project Explorer แบบ โปรเจกต์เดียว และ แบบหลายโปรเจกต์

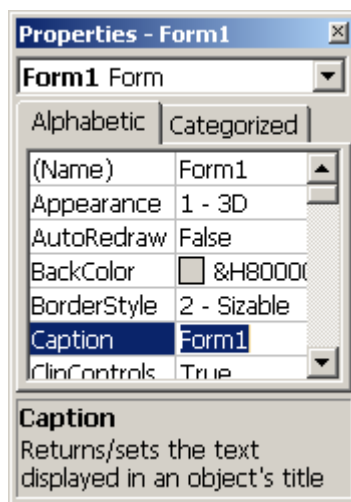
ตารางที่ 2.2 แสดงส่วนประกอบของโปรเจกต์

Project(n)	คือ โปรแกรมประยุกต์ที่พัฒนาอยู่ มีนามสกุล .vbp
Form(n)	เป็นฟอร์มที่มีอยู่ในโปรเจกต์นั้น ๆ ใน 1 โปรเจกต์อาจมีมากกว่า 1 ฟอร์มก็ได้ มีนามสกุล .frm

Modules	เป็นที่เก็บชุดคำสั่งที่คุณเขียนขึ้นมา โดยจะเก็บชุดคำสั่งที่ใช้บ่อย ๆ มีนามสกุล .bas
Class Modules	เป็นโมดูลชนิดพิเศษที่มีลักษณะเป็นอ็อบเจกต์ ที่สามารถสร้างขึ้นมาได้ จะมีนามสกุล .cls
User controls	เป็นส่วนที่เก็บคอนโทรล ActiveX ที่คุณสร้างขึ้นมา มีนามสกุล .ctl
Designers	เป็นส่วนของรายงานที่ถูกสร้างขึ้นมีนามสกุลเป็น .dsr

- Properties Window

หน้าต่างคุณสมบัติเป็นส่วนที่ใช้กำหนดคุณสมบัติของอ็อบเจกต์ที่ถูกเลือก (active) หรือได้รับความสนใจ (focus) อยู่ขณะนั้น ซึ่งสามารถที่จะปรับเปลี่ยนค่าต่าง ๆ ของคอนโทรลเพื่อให้เกิดความเหมาะสมและตรงกับความต้องการใช้งานได้ทันที



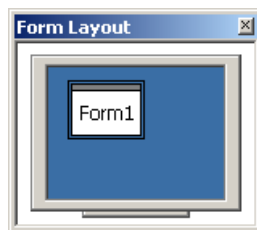
รูปที่ 2.8 Properties Window

ในหน้าต่างคุณสมบัติ จะประกอบไปด้วยแท็บ 2 แท็บ คือ

1. แท็บ Alphabetic เป็นแท็บที่แสดงรายการคุณสมบัติ เรียงตามตัวอักษรในภาษาอังกฤษ
2. แท็บ Categorized เป็นแท็บที่แสดงรายการคุณสมบัติ โดยการจัดกลุ่มของคุณสมบัติที่มีหน้าที่คล้ายกัน หรือมีความสัมพันธ์กัน

- หน้าต่าง Form Layout

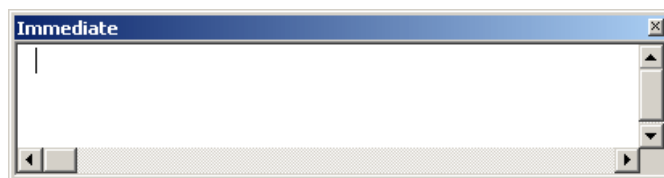
เป็นส่วนที่แสดงให้เห็นตำแหน่งของฟอร์ม และสามารถกำหนดตำแหน่งของฟอร์ม ที่ปรากฏบนจอภาพในขณะประมวลผลได้ โดยการเคลื่อนย้ายฟอร์มจำลอง ที่อยู่ในจอภาพจำลองด้วยการ drag เมาส์ ไปยังตำแหน่งที่คุณต้องการ โดยจะมีผลในขณะประมวลผลเท่านั้น



รูปที่ 2.9 Form Layout

- Immediate Window

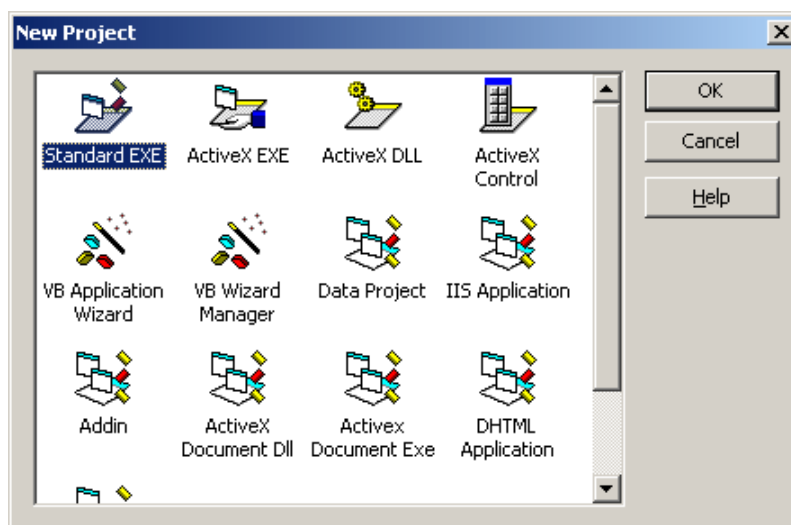
เป็นหน้าต่างที่ให้ประโยชน์ ในกรณีที่คุณต้องการทราบผล การประมวลผลโดยทันที เช่น การทดสอบโปรแกรมย่อยต่าง ๆ เป็นต้น เมื่อคุณสั่งประมวลผลโปรแกรมโปรเจกต์ หน้าต่างนี้จะปรากฏขึ้น โดยอัตโนมัติ



รูปที่ 2.10 Immediate Window

- หน้าต่าง New Project

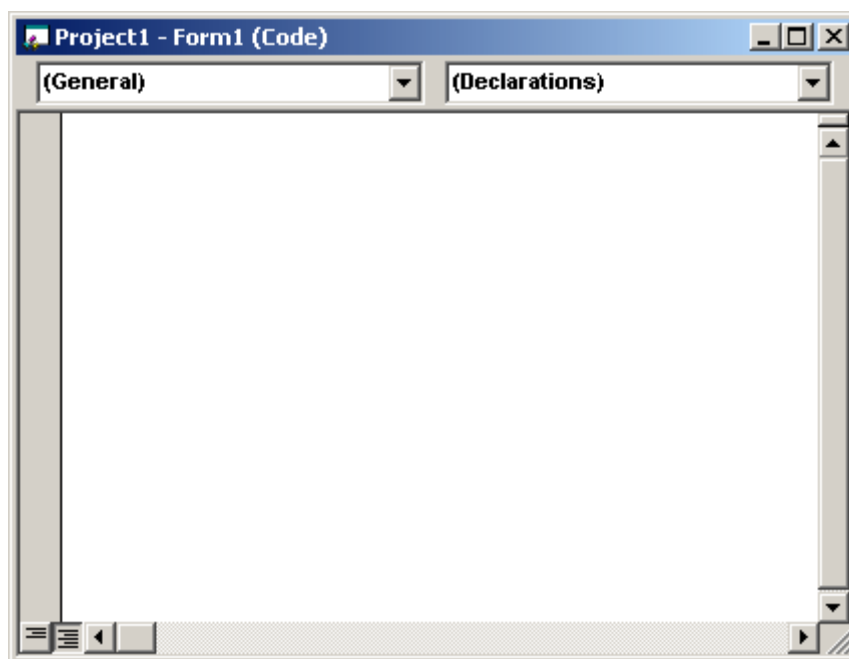
หน้าต่าง New Project จะปรากฏขึ้นมาเมื่อเลือกเมนู File/New Project กรอบโต้ตอบนี้ จะแสดงชนิดของโปรแกรมประยุกต์ ที่คุณต้องการพัฒนา ซึ่งจะคล้ายกับตอนที่เปิดโปรแกรม Visual Basic ขึ้นมาครั้งแรก



รูปที่ 2.11 กรอบโต้ตอบ New Project

- หน้าต่าง Code Editor

เป็นส่วนที่ใช้ในการเขียนชุดคำสั่งสำหรับการประมวลผล และควบคุมการทำงานของคอลโทรลต่าง ๆ



รูปที่ 2.12 Code Editor

2.2 พื้นฐานการสื่อสารแบบอนุกรม (Serial port) [2]

ถึงแม้ว่าการสื่อสารแบบอนุกรมในเครื่องคอมพิวเตอร์นั้นจะมีความเร็วในการสื่อสารช้ากว่าแบบขนาน ทั้งนี้ก็เพราะว่า การเคลื่อนย้ายข้อมูลแบบอนุกรมนั้นเป็นการส่งข้อมูลครั้งละ 1 บิต แต่พอร์ตนานนั้นสามารถส่งข้อมูลได้ครั้งละหลายๆ บิตพร้อมกัน ส่งผลให้การสื่อสารข้อมูลแบบอนุกรมมีความเร็วต่ำกว่าแบบขนาน แต่การส่งข้อมูลแบบอนุกรมนั้น สามารถส่งข้อมูลได้ไกลกว่าแบบขนาน อีกทั้งสายสัญญาณที่ใช้ยังมีน้อยกว่าการส่งข้อมูลแบบขนานอีกด้วย การสื่อสารแบบอนุกรม สามารถแบ่งออกเป็น 3 รูปแบบดังนี้

1. Simplex สามารถส่งข้อมูลได้อย่างเดียว เป็นการสื่อสารแบบทางเดียว
2. Half - Duplex สามารถส่งข้อมูล ไปยังปลายทางและสามารถรับข้อมูลจากปลายทางได้ แต่ไม่สามารถทำการส่งและรับข้อมูลได้ในเวลาเดียวกัน
3. Full - Duplex สามารถรับและส่งข้อมูลได้ในเวลาเดียวกัน

การส่งข้อมูลแบบอนุกรมในเครื่องคอมพิวเตอร์นั้นจะใช้มาตรฐาน RS - 232C

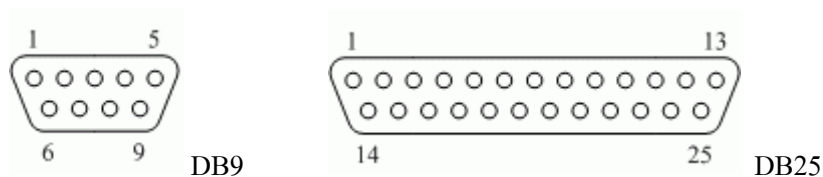
มาตรฐาน RS - 232C เป็นมาตรฐานที่ได้รับการออกแบบมาเพื่อที่จะทำให้อุปกรณ์จากผู้ผลิตต่างกันสามารถทำงานร่วมกันได้ มาตรฐานหลายชนิดได้รับการออกแบบขึ้นมา แต่มาตรฐานที่ได้รับความนิยม

นิยมและใช้กันกว้างขวางมากที่สุดคือ มาตรฐาน RS - 232C ซึ่งถูกประกาศใช้ในปี 1969 โดยสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association : EIA) มาตรฐาน RS - 232C ได้แบ่งอุปกรณ์ออกเป็น 2 ประเภท ทั้งนี้ก็เพื่อป้องกันไม่ให้เกิดการส่งข้อมูลบนสายเส้นเดียวกัน ซึ่งอุปกรณ์ทั้งสองประเภทนี้ก็คือ

อุปกรณ์ DTE (Data Terminal Equipment) เป็นอุปกรณ์สำหรับส่งข้อมูล (เอาต์พุต)
 อุปกรณ์ DCE (Data Communication Equipment) เป็นอุปกรณ์สำหรับรับข้อมูล (อินพุต)

ตามมาตรฐาน RS - 232C คอนเน็คเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็คเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งคอนเน็คเตอร์ที่นิยมใช้กันอยู่จะเป็นชนิด D-Type แบบ 9 ขา และ แบบ 25 ขา โดยจะติดตั้งอยู่หลังเครื่องคอมพิวเตอร์ ระดับแรงดันจะมีค่าระหว่าง -3 V ถึง -15 V สำหรับลอจิก High และ ลอจิก Low จะมีระดับแรงดันระหว่าง +3 V ถึง +15 V สามารถรับส่งข้อมูลได้ด้วยความยาวของสายสัญญาณสูงสุด 50 ฟุต หรือ 150 เมตรแต่ถ้าเราต้องการสื่อสารกับอุปกรณ์อื่นที่อยู่ห่างกันมากๆ เราจำเป็นต้องใช้อุปกรณ์อื่นๆ เข้าช่วย เช่นการใช้โมเด็ม เป็นต้น

ลักษณะของคอนเน็คเตอร์แบบ D-Type หัวต่อแบบ D-Type ที่ใช้ในการสื่อสารแบบอนุกรมของเครื่องคอมพิวเตอร์นั้น จะมีอยู่ 2 ลักษณะคือ แบบ 9 ขา และแบบ 25 ขา บางครั้งเราจะเรียกว่า DB9 และ DB25 ซึ่งหัวต่อทั้งสองชนิดจะมีลักษณะการทำงานของสัญญาณเหมือนกัน แต่การจัดเรียงไม่เหมือนกัน



รูปที่ 2.13 แสดงลักษณะของคอนเน็คเตอร์

ตารางที่ 2.3 แสดงรายละเอียดของสายสัญญาณ

D-Type 25 Pin	D-Type 9 Pin	สัญลักษณ์	ชื่อสัญญาณ
Pin 2	Pin 3	TD	Transmit Data
Pin 3	Pin 2	RD	Receive Data
Pin 4	Pin 7	RTS	Request To Send
Pin 5	Pin 8	CTS	Clear To Send

Pin 6	Pin 6	DSR	Data Set Ready
Pin 7	Pin 5	SG	Signal Ground
Pin 8	Pin 1	CD	Carrier Detect
Pin 20	Pin 4	DTR	Data Terminal Ready
Pin 22	Pin 9	RI	Ring Indicator

รายละเอียดของสายสัญญาณประกอบไปด้วย

- Transmit Data : TD ใช้สำหรับส่งข้อมูลอนุกรมออกจากคอมพิวเตอร์
- Receive Data : RD ใช้สำหรับรับข้อมูลอนุกรมเข้ามายังคอมพิวเตอร์
- Request To Send : RTS ใช้สำหรับส่งข้อมูลไปยังอุปกรณ์ปลายทาง เพื่อร้องขอให้อุปกรณ์ปลายทางส่งข้อมูลกลับมา
- Clear To Send : CTS ใช้สำหรับตรวจสอบว่าอุปกรณ์ที่เชื่อมต่อด้วยพร้อมที่จะรับข้อมูลหรือไม่ โดยจะคอยนับสัญญาณ RTS เมื่อทุกอย่างพร้อมก็จะทำการส่งข้อมูลออกทางขา TD
- Data Set Ready : DSR ใช้สำหรับตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง จะใช้คู่กับขา TD เป็นกราวด์ของระบบ
- Signal Ground : SG
- Carrier Detect : CD ขานี้จะ Active เมื่อมีการส่งสัญญาณ Carrier จากโมเด็ม
- Data Terminal Ready : DTR ใช้สำหรับบอกให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อด้วยโดยขา DTR นี้ต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง
- Ring Indicator : RI ขานี้จะ Active เมื่อโมเด็มได้รับสัญญาณเรียกเข้าจากสายโทรศัพท์

2.2.1 องค์ประกอบของการรับส่งข้อมูลแบบอนุกรม

การสื่อสารแบบอนุกรมที่นิยมใช้กับคอมพิวเตอร์นั้น เป็นการสื่อสารข้อมูลแบบอะซิงโครนัส นั่นคือ ต้องใช้สายสัญญาณเส้นเดียวทำหน้าที่ทั้งส่วนที่เป็นข้อมูล และส่วนที่ใช้ควบคุมการส่งข้อมูล ดังนั้นข้อมูลที่อ่านได้แต่ละบิตจากการส่งแบบอนุกรม จึงต้องถูกแยกแยะว่าใช้สำหรับวัตถุประสงค์ใด โดยสามารถแบ่งได้เป็น 4 ส่วนคือ

1. Start Bit	ขนาด 1 บิต
2. Data Character (บิตข้อมูล)	ขนาด 7 หรือ 8 บิต
3. Parity Bit	ขนาด 1 บิต
4. Stop Bit	ขนาด 1 หรือ 2 บิต

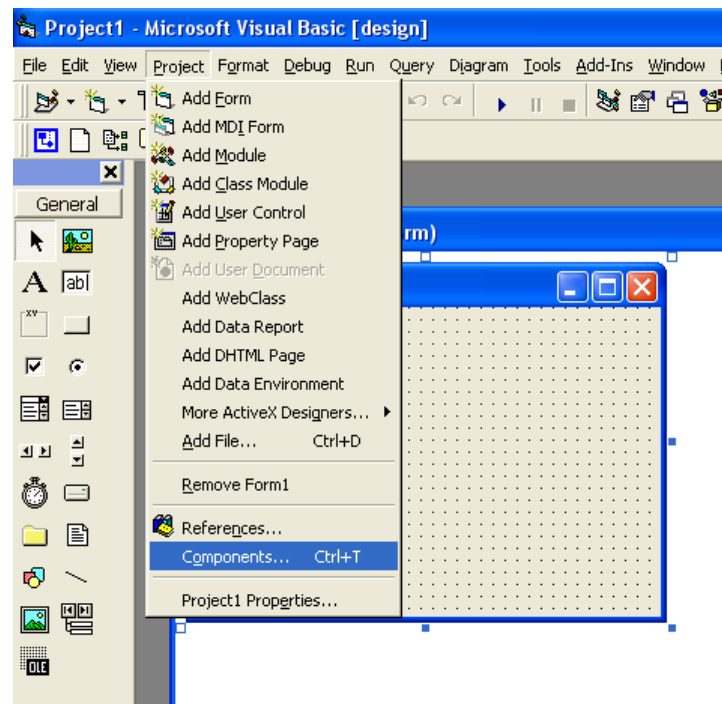
แต่ละบิตที่ถูกส่งออกไปเป็นกลุ่มจะประกอบไปด้วยบิตเริ่มต้น บิตข้อมูล บิตพาริตี (จะมีหรือไม่มีก็ได้) และบิตจบ โดยแต่ละส่วนมีหน้าที่หลักดังนี้

- Start Bit หรือบิตเริ่มต้นจะใส่ที่จุดเริ่มต้นเสมอ เพื่อเตือนอุปกรณ์ฝ่ายรับว่าข้อมูลกำลังจะมาถึง
- Data Character หรือบิตข้อมูล การส่งบิตข้อมูลจะส่งเป็นกลุ่มๆ โดยทั่วไปจะส่งเป็น 7 หรือ 8 บิต ซึ่งเพียงพอสำหรับการส่ง ASCII Word
- Parity Bit หรือบิตพาริตี ใช้ในการตรวจสอบความถูกต้องของข้อมูลที่ส่ง โดยจะใส่บิตพาริตีเข้าไป แต่ทั้งตัวรับและตัวส่งจะต้องรู้กันว่าใช้พาริตีแบบไหนในการส่งข้อมูล
- Stop Bit หรือบิตจบ เป็นบิตที่ส่งมาปิดท้ายข้อมูล

อัตราเร็วในการรับส่งข้อมูลแบบอนุกรม อุปกรณ์ทั้ง 2 อย่างจะต้องทำงานด้วยอัตราเร็วเท่ากัน ซึ่งอัตราเร็วในการสื่อสารแบบอะซิงโครนัสคือ ค่าบอดเรต (Baud Rate) มีหน่วยเป็นบิตต่อวินาที ซึ่งค่าอัตราเร็วในการสื่อสารแบบอนุกรมสำหรับมาตรฐาน RS-232C นั้นมีใช้ดังนี้ 110, 150, 300, 600, 1200, 2400, 4800, 9600 และ 19200 บิตต่อวินาที

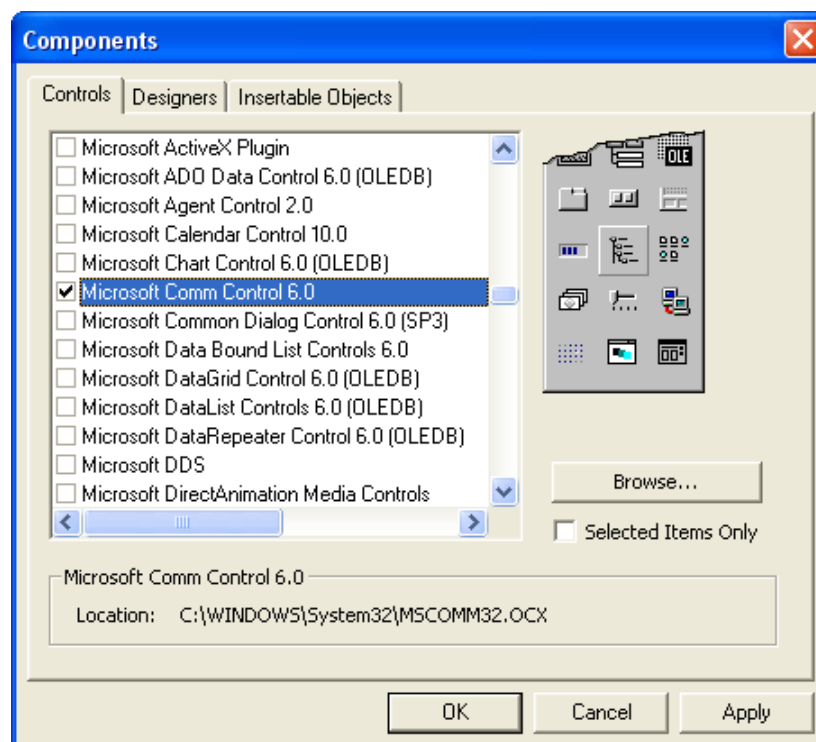
2.2.2 การเขียนโปรแกรมติดต่อ และควบคุม Serial Port กับ Visual Basic

2.2.2.1 คอนโทรลที่สำคัญที่ทำให้ Visual Basic สามารถสื่อสารผ่านพอร์ตอนุกรมได้นั้นก็คือ คอนโทรล MSComm ซึ่งไม่ใช่คอนโทรลมาตรฐาน ดังนั้นถ้าเราต้องการใช้งาน MSComm เราจะต้องทำการเพิ่มคอนโทรลนี้เข้าไปใน ToolBox ซึ่งสามารถทำได้โดยคลิกขวาที่ ToolBox แล้วเลือกเมนู Components ดังรูปที่ 2.14



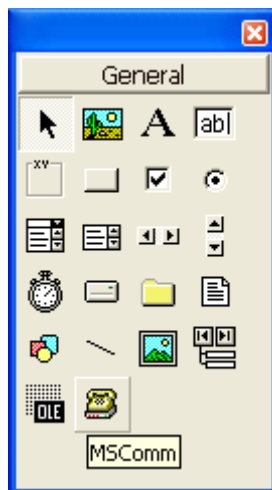
รูปที่ 2.14 เพิ่ม Components MSComm

จากนั้นจะปรากฏไดอะล็อก Components ขึ้นมา จากนั้นให้คลิกเลือกที่ Microsoft Comm Control 6.0 แล้วคลิกปุ่ม OK ดังรูปที่ 2.15



รูปที่ 2.15 เลือกที่รายการ MSComm

จากนั้นก็ปรากฏภายใน ToolBox จะมีไอคอนรูปโทรศัพท์ ซึ่งเป็นไอคอนของ MSComm ปรากฏขึ้นมาให้เราเลือกใช้งานดังรูปที่ 2.16



รูปที่ 2.16 คอนโทรล MSComm พร้อมใช้งาน

2.2.2.2 ฟร็อพเพอร์ตีที่สำคัญในการใช้งาน MSComm

- CommPort ใช้ในการกำหนดหมายเลขของพอร์ตอนุกรมที่เราต้องการติดต่อ ตัวอย่างเช่น ถ้าเรากำหนดให้การเขียนโปรแกรมติดต่อกับพอร์ต Com1 จะเขียนเป็น

```
MSComm1.CommPort = 1
```

- Settings ใช้ในการกำหนดอัตรา Baud Rate หรือความเร็วในการส่งข้อมูล มีหน่วยเป็นบิตต่อวินาที, พาริตี, จำนวนของบิตข้อมูล, จำนวนของบิตปิดท้าย ยกตัวอย่างเช่น เรากำหนดให้มีการเขียนโปรแกรมใช้งานที่ Baud Rate = 9600 บิตต่อวินาที ไม่มีพาริตี จำนวนบิตข้อมูลเท่ากับ 8 บิต และมีบิตปิดท้าย 1 บิต

```
MSComm1.Settings = "9600, N, 8, 1"
```

- PortOpen ใช้สำหรับเปิดและปิดการใช้งานพอร์ตอนุกรม ยกตัวอย่างเช่น เราจะเปิดใช้งานพอร์ตอนุกรม ให้กำหนดค่า Value เป็น True เขียนโค้ดได้ดังนี้

```
MSComm1.PortOpen = True
```

แต่ถ้าต้องการปิดพอร์ตอนุกรม ให้กำหนดค่า Value เป็น False

```
MSComm1.PortOpen = False
```

- InBufferSize เป็นการกำหนดขนาดของ Buffer ในการรับข้อมูลเข้ามา โดยมีรูปแบบการทำงานดังนี้

```
object.InBufferSize [= value]
```

- OutBufferSize เป็นการกำหนดขนาดของ Buffer ในการส่งข้อมูลออกไป โดยมี

รูปแบบการกำหนดดังนี้

`object.OutBufferSize [= value]`

- `InputLen` เป็นการกำหนดค่าของข้อมูลที่อ่านจาก Buffer ภาครับ โดยมี

รูปแบบการกำหนดดังนี้

`object.InputLen [= value]`

- `InputMode` เป็นการกำหนดค่าชนิดของข้อมูลที่ได้รับเข้ามา โดยมีรูปแบบการกำหนดดังนี้

`Object.InputMode [= value]`

โดยที่เราสามารถเลือกชนิดของข้อมูลได้ 2 ประเภท คือ

1. `comInputModeText` ข้อมูลที่รับเข้ามาเป็นข้อความปกติ เราสามารถตั้งค่าให้อยู่ในโหมดนี้ได้โดยการกำหนด `value` ให้เป็น “ 0 ”

2. `comInputModeBinary` ข้อมูลที่รับเข้ามาเป็นข้อมูลไบนารี เราสามารถตั้งค่าให้อยู่ในโหมดนี้ได้โดยการกำหนดค่า `value` ให้เป็น “ 1 ”

- `Input` ใช้ในการอ่านค่าข้อมูลจากพอร์ตอนุกรม ยกตัวอย่างเช่น ถ้าเราอ่านค่าจากบัฟเฟอร์ของพอร์ตอนุกรม แล้วนำมาเก็บไว้ในตัวแปรที่ชื่อว่า `Data` จะเขียนโค้ดได้ดังนี้

`Data = MSComm1.Input`

- `Output` ใช้ในการส่งข้อมูลออกไปจากพอร์ตอนุกรม โดยมีรูปแบบของการเขียนดังนี้

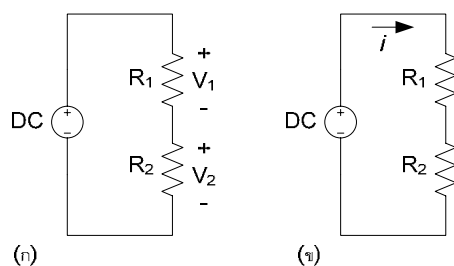
`object.Output [= value]`

- `EOFEnable` เป็นการบอกว่าสิ้นสุดของไฟล์ `End of File` มีรูปแบบการใช้งานดังนี้

`Object.EOFEnable [= value]`

2.3 วงจรแบ่งแรงดัน [3]

ในวงจรแบ่งแรงดันเราสามารถวิเคราะห์วงจรโดยใช้กฎของโอห์ม และกฎของเคอร์ชอฟฟ์เพื่อช่วยในการวิเคราะห์ เราจะได้กำหนดกระแส (i) ดังแสดงในรูปที่ 2.17 จากกฎกระแสของเคอร์ชอฟฟ์ กระแสที่ผ่าน R_1 และ R_2 มีค่าเท่ากัน จากการใช้กฎแรงดันของเคอร์ชอฟฟ์รอบทางเดินวงรอบปิด จะได้สมการ



รูปที่ 2.17 (ก) วงจรแบ่งแรงดัน

(ข) วงจรแบ่งแรงดันซึ่งกำหนดค่ากระแส i

$$v_s = iR_1 + iR_2 \quad (2.1)$$

หรือ

$$i = \frac{v_s}{R_1 + R_2} \quad (2.2)$$

เราสามารถใช้กฎของโอห์มเพื่อคำนวณหาค่า v_1 และ v_2 :

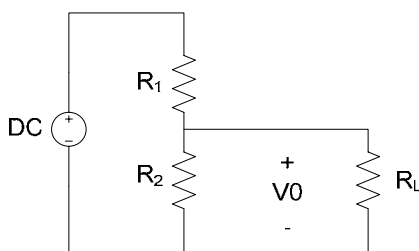
$$v_1 = iR_1 = v_s \frac{R_1}{R_1 + R_2} \quad (2.3)$$

$$v_2 = iR_2 = v_s \frac{R_2}{R_1 + R_2} \quad (2.4)$$

สมการที่ (2.3) และ (2.4) แสดงให้เห็นว่า v_1 และ v_2 คืออัตราส่วนของ v_s โดยที่อัตราส่วนแต่ละค่า คืออัตราส่วนความต้านทานที่ต้องการหาค่าแรงดันที่ตกคร่อมตัวมันอยู่ต่อผลบวกของความต้านทานทั้งสอง เนื่องจากอัตราส่วนทั้งสองค่านี้มีค่าน้อยกว่า 1.0 เสมอ ค่าแรงดัน v_1 และ v_2 จึงมีค่าน้อยกว่าค่าแรงดัน v_s จากแหล่งกำเนิด

ถ้าเราต้องการให้ v_s มีค่าๆ หนึ่งและทราบค่าของ v_s ซึ่งจะมีค่า R_1 และ R_2 จำนวนอนันต์ที่ให้ค่า v_s ที่ต้องการ ตัวอย่างเช่น สมมติว่า v_s เท่ากับ 15 โวลต์ และ v_s ที่ต้องการคือ 5 โวลต์ ดังนั้น $v_2/v_s = 1/3$ จากสมการที่ (2.4) จะเห็นว่า $R_2 = \frac{1}{2}R_1$ ปัจจัยอื่นๆ ที่ต้องคำนึงถึงในการเลือกค่า R_1 และ R_2 คือกำลังไฟฟ้าสูญเสียที่เกิดขึ้นจากการแบ่งแรงดันจากแหล่งกำเนิด และผลของการต่อวงจรแบ่งแรงดันกับอุปกรณ์อื่น ๆ ในวงจร

พิจารณาการต่อ R_L ขนานกับ R_2 ดังวงจรแสดงในรูปที่ 2.18 ตัวต้านทาน R_L ทำหน้าที่เสมือนโหลดของวงจรแบ่งแรงดัน โหลดของวงจรใดๆ สามารถประกอบด้วยอุปกรณ์ไฟฟ้าเพียงตัวเดียวหรือมากกว่าซึ่งดึงกำลังไฟฟ้าจากวงจร เมื่อตัวต้านทาน R_L ต่อในวงจรแบ่งแรงดัน



รูปที่ 2.18 วงจรแบ่งแรงดันที่ต่อกับโหลด R_L

เอาที่พูดสามารถหาได้จาก

$$v_0 = \frac{R_{eq}}{R_1 + R_{eq}} v_s \quad (2.5)$$

เมื่อ

$$R_{rq} = \frac{R_2 R_L}{R_2 + R_L} \quad (2.6)$$

แทนสมการที่ (2.6) ลงในสมการที่ (2.5) จะได้

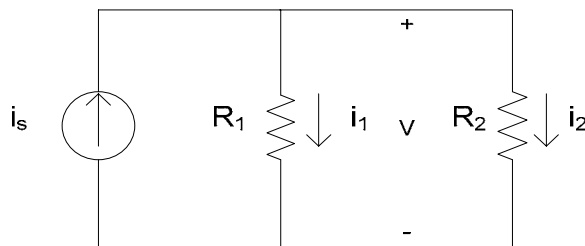
$$v_0 = \frac{R_2}{R_1 [1 + (R_2 / R_L)] + R_2} v_s \quad (2.7)$$

จะเห็นว่าสมการที่ (2.7) จะกลายเป็นสมการที่ (2.4) เมื่อ $R_L \rightarrow \infty$ สมการที่ (2.7) แสดงให้เห็นว่าถ้า $R_L \gg R_2$ อัตราส่วนแรงดัน v_0 / v_s ไม่เปลี่ยนแปลงไปมากนัก จากการเพิ่มโหลดในวงจรแบ่งแรงดัน

ลักษณะอีกอย่างของวงจรแบ่งแรงดันคือ ความไวของการแบ่งแรงดันต่อค่าความคลาดเคลื่อนที่ยอมรับได้ของตัวต้านทาน ค่าความต้านทานของตัวต้านทานที่มีขายในปัจจุบันจะมีความคลาดเคลื่อนตามจำนวนร้อยละที่ระบุ

2.4 วงจรแบ่งกระแส [3]

วงจรแบ่งกระแสดังแสดงในรูปที่ 2.19 ประกอบด้วยตัวต้านทาน 2 ตัวที่ต่อขนานกับแหล่งกำเนิดกระแส วงจรแบ่งกระแสถูกออกแบบเพื่อแบ่งกระแส i_s ไปยังตัวต้านทาน R_1 และ R_2 เราสามารถหาความสัมพันธ์ระหว่างกระแส i_s กับกระแสที่ผ่านตัวต้านทานแต่ละตัว (i_1 และ i_2) โดยใช้กฎของโอห์ม และกฎกระแสของเคอร์ชอฟฟ์ แรงดันที่คร่อมตัวต้านทานที่ต่อขนานกัน คือ



รูปที่ 2.19 วงจรแบ่งกระแส

$$V = i_1 R_1 = i_2 R_2 = \frac{R_1 R_2}{R_1 + R_2} \cdot i_s \quad (2.8)$$

จากสมการที่ (2.8) จะได้

$$i_1 = \frac{R_2}{R_1 + R_2} \cdot i_s \quad (2.9)$$

$$i_2 = \frac{R_1}{R_1 + R_2} \cdot i_s \quad (2.10)$$

สมการที่ (2.9) และ (2.10) แสดงให้เห็นว่าค่ากระแสถูกแบ่งให้ไหลไปยังตัวต้านทานที่ต่อขนาน โดยที่ค่ากระแสที่ไหลผ่านตัวต้านทานตัวหนึ่งมีค่าเท่ากับค่ากระแสที่ไหลเข้ามายังตัวต้านทานทั้งสองคูณด้วยค่าความต้านทานอีกตัวหนึ่งหารด้วยผลบวกของค่าความต้านทานทั้งสองตัว

2.5 สรุปผล

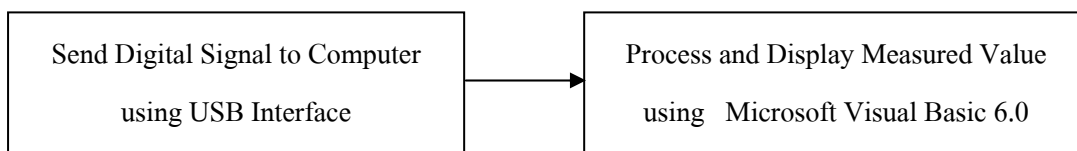
โปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการนี้ได้ถูกพัฒนาขึ้นมาโดยใช้คุณสมบัติการสื่อสารแบบอนุกรมของระหว่างคอมพิวเตอร์และอุปกรณ์เชื่อมต่อภายนอกของโปรแกรม Microsoft Visual Basic 6.0 เพื่อที่จะรับค่าสัญญาณทางไฟฟ้าจากวงจรแบ่งแรงดันและวงจรแบ่งกระแสจากอุปกรณ์ภายนอกเข้ามาประมวลผลแล้วทำการแสดงผลที่โปรแกรม

บทที่ 3

การออกแบบและการเขียนโปรแกรมคอมพิวเตอร์ช่วยสอน

ปฏิบัติการอิเล็กทรอนิกส์

กระบวนการทำงานของโปรแกรมเชื่อมต่อคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์ มีอยู่ทั้งหมด 2 ส่วนด้วยกัน คือ รับค่าสัญญาณดิจิทัลจากอุปกรณ์เชื่อมต่อภายนอกโดยผ่านทางพอร์ต USB, ประมวลผลและแสดงผลโดยโปรแกรม Microsoft Visual Basic 6.0 ดังแสดงไว้ในรูปที่ 3.1

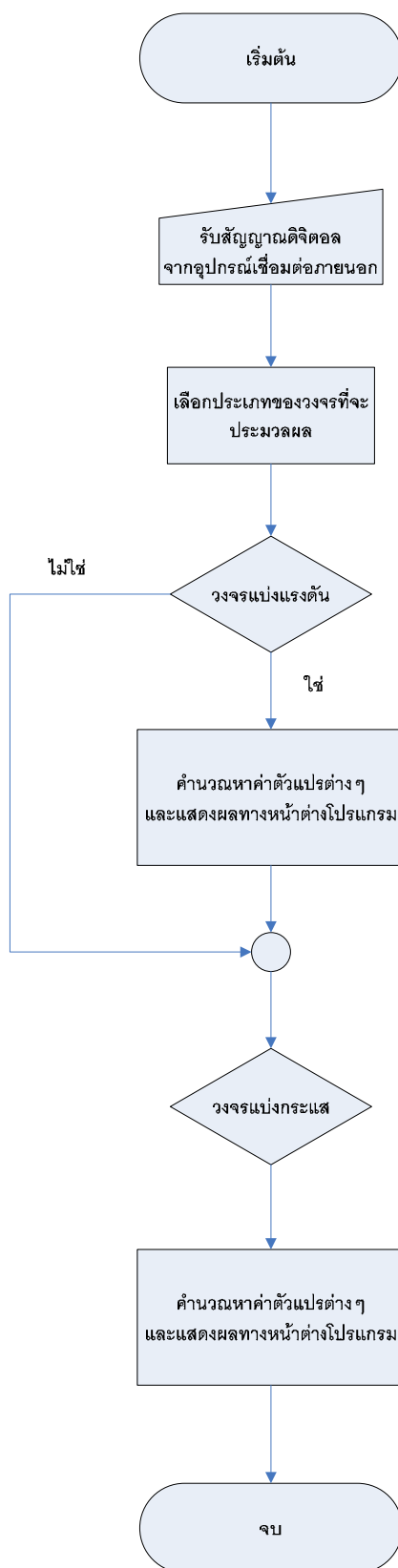


รูปที่ 3.1 ภาพโครงสร้างโดยรวมของโปรแกรม

3.1 ขั้นตอนการทำงานของโปรแกรมประมวลผลและแสดงผล (Microsoft Visual Basic 6.0)

3.1.1 กำหนดและออกแบบขั้นตอนการทำงานของโปรแกรม

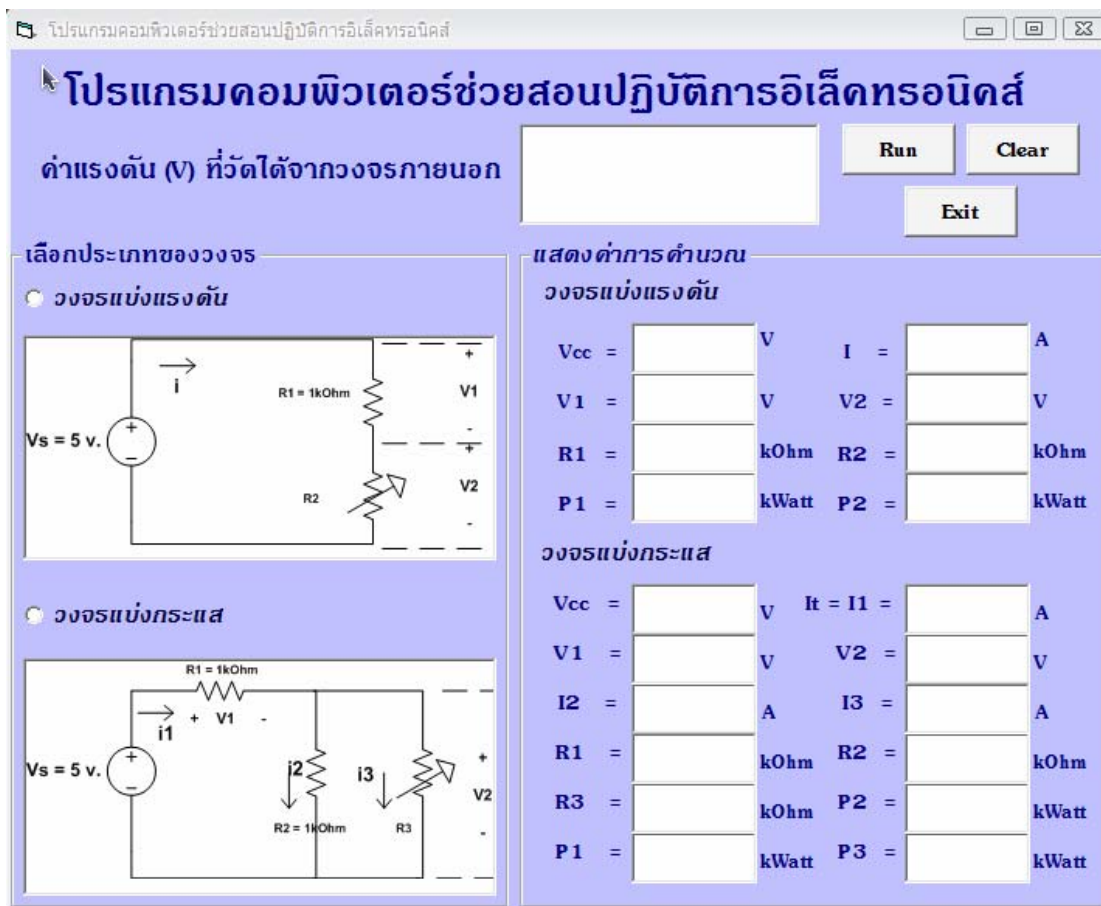
จากรูปที่ 3.2 ได้แสดงแผนผังแสดงขั้นตอนการทำงานของโปรแกรมไว้ จากนั้นทำการนำขั้นตอนดังกล่าวไปออกแบบรูปแบบของโปรแกรม



รูปที่ 3.2 ภาพแสดงขั้นตอนการทำงานของโปรแกรม

3.2 ออกแบบรูปแบบของโปรแกรม

จากรูปที่ 3.3 ได้แสดงหน้าต่างของโปรแกรมในการรับค่า การประมวลผลและการแสดงผลโดยใช้โปรแกรม Microsoft Visual Basic 6.0



รูปที่ 3.3 ภาพแสดงรูปแบบของโปรแกรม

เมื่อสร้างหน้าต่างของโปรแกรมขึ้นมาแล้วจึงสร้างตัวควบคุมต่างๆขึ้นมาดังแสดงในรูปที่ 3.3 โดยทำการกำหนดคุณสมบัติให้แก่ตัวควบคุม ดังนี้

ตารางที่ 3.1 แสดงการกำหนดคุณสมบัติให้กับตัวควบคุมต่างๆของโปรแกรม

ตัวควบคุม	คุณสมบัติ	ค่าที่กำหนด
Form	Name	Form1
	Caption	โปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์
MSComm	Name	MSComm1
	InBufferSize	1024

ตัวควบคุม	คุณสมบัติ	ค่าที่กำหนด
Timer	Name	Timer1
	Interval	1
Command Button	Name	Command1
	Caption	Run
Command Button	Name	Command2
	Caption	Clear
Command Button	Name	Command3
	Caption	Exit
Frame	Name	Frame1
	Caption	เลือกประเภทของวงจร
Frame	Name	Frame2
	Caption	แสดงค่าการคำนวณ
Option Button	Name	Option1
	Caption	วงจรแบ่งแรงดัน
Option Button	Name	Option2
	Caption	วงจรแบ่งกระแส
Picture Box	Name	Picture1
	Picture	Bitmap
Picture Box	Name	Picture2
	Picture	Bitmap
TextBox	Name	Text1
	Text	
TextBox	Name	Text2
	Text	
TextBox	Name	Text3
	Text	
TextBox	Name	Text4
	Text	
TextBox	Name	Text5
	Text	
TextBox	Name	Text6
	Text	

ตัวควบคุม	คุณสมบัติ	ค่าที่กำหนด
TextBox	Name	Text7
	Text	
TextBox	Name	Text8
	Text	
TextBox	Name	Text9
	Text	
TextBox	Name	Text10
	Text	
TextBox	Name	Text11
	Text	
TextBox	Name	Text12
	Text	
TextBox	Name	Text13
	Text	
TextBox	Name	Text14
	Text	
TextBox	Name	Text15
	Text	
TextBox	Name	Text16
	Text	
TextBox	Name	Text17
	Text	
TextBox	Name	Text18
	Text	
TextBox	Name	Text19
	Text	
TextBox	Name	Text20
	Text	
TextBox	Name	Text21
	Text	
Label	Name	Label1
	Caption	โปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์

ตัวควบคุม	คุณสมบัติ	ค่าที่กำหนด
Label	Name	Label2
	Caption	ค่าแรงดัน(V)ที่วัดได้จากวงจรภายนอก
Label	Name	Label3
	Caption	วงจรแบ่งแรงดัน
Label	Name	Label4
	Caption	วงจรแบ่งกระแส
Label	Name	Label5
	Caption	V2
Label	Name	Label6
	Caption	Vcc
Label	Name	Label7
	Caption	V1
Label	Name	Label8
	Caption	V2
Label	Name	Label9
	Caption	I
Label	Name	Label10
	Caption	R1
Label	Name	Label11
	Caption	R2
Label	Name	Label12
	Caption	P1
Label	Name	Label13
	Caption	P2
Label	Name	Label14
	Caption	Vcc
Label	Name	Label15
	Caption	V1
Label	Name	Label16
	Caption	R1
Label	Name	Label17
	Caption	P1

ตัวควบคุม	คุณสมบัติ	ค่าที่กำหนด
Label	Name	Label18
	Caption	It = I1
Label	Name	Label19
	Caption	I2
Label	Name	Label20
	Caption	I3
Label	Name	Label21
	Caption	R2
Label	Name	Label22
	Caption	R3
Label	Name	Label23
	Caption	P2
Label	Name	Label24
	Caption	P3
Label	Name	Label25
	Caption	V
Label	Name	Label26
	Caption	V
Label	Name	Label27
	Caption	V
Label	Name	Label28
	Caption	A
Label	Name	Label29
	Caption	kOhm
Label	Name	Label30
	Caption	kOhm
Label	Name	Label31
	Caption	kWatt
Label	Name	Label32
	Caption	kWatt
Label	Name	Label33
	Caption	V

ตัวควบคุม	คุณสมบัติ	ค่าที่กำหนด
Label	Name	Labe34
	Caption	V
Label	Name	Label35
	Caption	V
Label	Name	Label36
	Caption	A
Label	Name	Label37
	Caption	A
Label	Name	Label38
	Caption	A
Label	Name	Label39
	Caption	kOhm
Label	Name	Label40
	Caption	kOhm
Label	Name	Label41
	Caption	kOhm
Label	Name	Label42
	Caption	kWatt
Label	Name	Label43
	Caption	kWatt
Label	Name	Label44
	Caption	kWatt

3.3 การเขียนโปรแกรม Microsoft Visual Basic 6.0

3.3.1 การเขียนโค้ดโปรแกรมเพื่อควบคุมหน้าต่างของโปรแกรม

จากรูปแบบของโปรแกรมที่ทำการออกแบบดังรูปที่ 3.3 นำไปทำการเขียนคำสั่งเพื่อควบคุมให้โปรแกรมทำงานได้ตามที่ออกแบบขั้นตอนไว้ ดังนี้

ตารางที่ 3.2 แสดงส่วนของโค้ดโปรแกรม

Option Explicit	
Dim b As Single	} ประกาศค่าตัวแปรต่างๆให้เป็นประเภท Single
Dim c As Single	
Dim i As Single	
Dim ir1 As Single	
Dim ir2 As Single	
Dim ir3 As Single	
Dim r3 As Single	
Dim pr1 As Single	
Dim pr2 As Single	
Dim pr3 As Single	
Dim r2 As Single	
Dim rdet As Variant	ประกาศค่าตัวแปร rdet ให้เป็นประเภท Variant
Private Sub Form_Load()	
MSComm1.CommPort = 3	กำหนดให้โปรแกรมติดต่อกับพอร์ต 3
MSComm1.Settings = "9600,N,8,1"	กำหนดอัตราบอด พาริตีบิต จำนวนบิตข้อมูล และจำนวนของบิตปิดท้าย (กำหนดคุณสมบัติให้พอร์ต)
End Sub	
Private Sub MSComm1_OnComm()	
On Error Resume Next	
If MSComm1.CommEvent = comEvReceive Then	การรับจำนวนอักษรเพื่อรอคำสั่ง input
rdet = MSComm1.Input	รับค่าจากพอร์ตไปเก็บไว้ในตัวแปรชื่อ rdet
End If	
Text1.Text = CSng(rdet)	แสดงค่าที่เป็นเลขทศนิยมของ rdet ที่ Text1
in case of voltage divider	กรณีที่เป็นวงจรแบ่งแรงดัน
If Option1.Value = True Then	ถ้ามีการเลือกที่ option1
Call voltagedivider	ให้ทำการเรียกใช้โปรแกรมย่อยชื่อ voltagedivider
End If	

<pre>' in case of current divider If Option2.Value = True Then Call currentdivider End If End Sub</pre>	<p>กรณีที่เป็นวงจรแบ่งกระแส</p> <p>ถ้ามีการเลือกที่ option2 ให้ทำการเรียกใช้โปรแกรมย่อยชื่อ currentdivider</p>
<pre>Sub voltagedivider() b = CSng(rdet) c = 5 - b r2 = b * (1 / (5 - b)) i = c / 1 i1 = I i2 = b / r2 pr1 = i * c pr2 = i * b Text2.Text = 5 Text3.Text = c Text4.Text = 1 Text5.Text = pr1 Text6.Text = i Text7.Text = b Text8.Text = r2 Text9.Text = pr2 End Sub</pre>	<p>ทำการคำนวณค่าตัวแปรต่างๆของวงจรแบ่งแรงดัน</p> <p>แสดงค่าตัวแปรต่างๆที่ได้จากการคำนวณที่ TextBox</p>

```
Sub currentdivider()
```

```
    b = CSng(rdet)
```

```
    c = 5 - b
```

```
    ir1 = c / 1
```

```
    ir2 = b / 1
```

```
    ir3 = ir1 - ir2
```

```
    r3 = b / ir3
```

```
    pr1 = ir1 * c
```

```
    pr2 = ir2 * b
```

```
    pr3 = ir3 * b
```

ทำการคำนวณค่าตัวแปรต่างๆของวงจรแบ่งกระแส

```
    Text10.Text = 5
```

```
    Text11.Text = c
```

```
    Text12.Text = ir2
```

```
    Text13.Text = 1
```

```
    Text14.Text = r3
```

```
    Text15.Text = pr1
```

```
    Text16.Text = ir1
```

```
    Text17.Text = b
```

```
    Text18.Text = ir3
```

```
    Text19.Text = 1
```

```
    Text20.Text = pr2
```

```
    Text21.Text = pr3
```

แสดงค่าตัวแปรต่างๆที่ได้จากการคำนวณที่ TextBox

```
End Sub
```

```
Private Sub Option1_Click()
```

```
    Option1.Value = True
```

```
    Option1.Enabled = True
```

```
    Option2.Enabled = False
```

```
    Option1.SetFocus
```


OptionButton1 ถูกเลือก

สามารถใช้งาน OptionButton1 ได้



ไม่สามารถใช้งาน OptionButton2

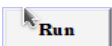
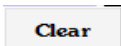
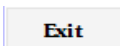
```
End Sub
```


<pre>Private Sub Option2_Click() Option2.Value = True Option1.Enabled = False Option2.Enabled = True Option2.SetFocus End Sub</pre>	<p>OptionButton2 ถูกเลือก ไม่สามารถใช้งาน OptionButton1 สามารถใช้งาน OptionButton2 ได้</p>
<pre>Private Sub Command1_Click() If MSComm1.PortOpen = False Then MSComm1.PortOpen = True MSComm1.RThreshold = 3 End If End Sub</pre>	<p>คอนโทรลของ CommandButton1 ตรวจสอบว่าพอร์ตติดต่อเปิดใช้อยู่หรือไม่ ถ้าไม่เปิดใช้ ทำการเปิดใช้งานพอร์ตทอนุกรม กำหนดค่า Threshold ที่รับเข้าเท่ากับ 3</p>
<pre>Private Sub Command2_Click() If Option1.Value = True Then Option1.Enabled = False Option2.Enabled = True End If If Option2.Value = True Then Option2.Enabled = False Option1.Enabled = True End If Text2.Text = "" Text3.Text = "" Text4.Text = "" Text5.Text = "" Text6.Text = "" Text7.Text = "" Text8.Text = "" Text9.Text = ""</pre>	<p>คอนโทรลของ CommandButton2 ถ้ามีการเลือกใช้งาน OptionButton1 อยู่ให้ ไม่สามารถใช้งาน OptionButton1 ได้ สามารถเลือกใช้งาน OptionButton2 ได้ ถ้ามีการเลือกใช้งาน OptionButton2 อยู่ให้ ไม่สามารถใช้งาน OptionButton2 ได้ สามารถเลือกใช้งาน OptionButton1 ได้</p> <p>เคลียร์ค่าใน TextBox ต่างๆให้ว่าง</p>

<pre> Text10.Text = "" Text11.Text = "" Text12.Text = "" Text13.Text = "" Text14.Text = "" Text15.Text = "" Text16.Text = "" Text17.Text = "" Text18.Text = "" Text19.Text = "" Text20.Text = "" Text21.Text = "" End Sub </pre>		<p>เคลียร์ค่าใน TextBox ต่างๆให้ว่าง</p>
<pre> Private Sub Command3_Click() End End Sub </pre>	<p>คอนโทรลของ CommandButton3</p> <p>จบการใช้งานโปรแกรม</p>	

3.3.2 อธิบายการทำงานของโปรแกรม

จากโค้ดข้างต้น การทำงานของโปรแกรมจะเริ่มจากการประกาศประเภทต่างๆ และจากคำสั่งต่างๆที่อยู่ในโพรซีเจอร์ Form_Load() ซึ่งจะกำหนดคุณสมบัติของคอนโทรล MSComm ต่อจากนั้นในส่วนของโพรซีเจอร์ MSComm1_OnComm() กำหนดการทำงานของ MSComm โดยเริ่มด้วยถ้ามีการส่งข้อมูลเข้ามาให้นำค่าที่รับเข้ามาได้ไปเก็บไว้ในตัวแปรที่ชื่อว่า rdet และแสดงค่าที่ TextBox1 หลังจากนั้นทำการตรวจสอบที่ OptionButton ต่างๆว่ามีการทำงานหรือไม่ โดยกำหนดว่าถ้ามีการเลือกใช้งานที่ OptionButton1 ให้ทำการเลือกใช้งานโปรแกรมย่อยที่ชื่อว่า voltagedivider แต่ถ้ามีการเลือกใช้งานที่ OptionButton2 ให้ทำการเรียกใช้งานโปรแกรมย่อยที่ชื่อว่า currentdivider ในส่วนการทำงานในโพรซีเจอร์ Sub voltagedivider() และ Sub currentdivider() จะเป็นการคำนวณค่าต่างๆตามหลักการของวงจรแบ่งแรงดัน วงจรแบ่งกระแส และทำการแสดงผลที่ได้ที่ช่อง TextBox ต่างๆ ในโพรซีเจอร์ถัดไปเป็นส่วนของ Sub Option1_Click() และ Sub Option2_Click() (หรือปุ่ม  วงจรแบ่งแรงดัน และปุ่ม  วงจรแบ่งกระแส) เป็นการทำงานและกำหนดการทำงานของ OptionButton ให้สามารถเลือกใช้งานได้ทีละ 1 ปุ่ม และ

สุดท้ายเป็นส่วนของโพธิ์เจอร์ของคอนโทรล CommandButton ต่างๆ โดยกำหนดให้ Sub Command1_Click() กำหนดให้ตรวจสอบสถานะของพอร์ตและทำการเปิดใช้งานพอร์ต ส่วนของ Sub Command2_Click() () กดที่คอนโทรลนี้เพื่อทำการเคลียร์ค่าที่เลือกใช้งานไว้ให้สามารถเลือกใช้งานปุ่มอื่นได้ และเคลียร์ค่าใน TextBox ต่างๆด้วย จบด้วยคอนโทรล Sub Command3_Click() () ออกจากการทำงานของโปรแกรม

3.4 สรุปผล

การออกแบบโปรแกรมเชื่อมต่อคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์ด้วยโปรแกรม Microsoft Visual Basic 6.0 เริ่มจากการกำหนดและออกแบบขั้นตอนการทำงานของโปรแกรม การสร้างหน้าต่างโปรแกรม จัดวางตัวควบคุมต่างๆของโปรแกรม กำหนดคุณสมบัติให้ตัวควบคุม แล้วจึงทำการเขียนโค้ดโปรแกรมเพื่อควบคุมการทำงานของโปรแกรมให้สามารถรับสัญญาณดิจิทัลจากอุปกรณ์ภายนอกแล้วทำการประมวลผล แสดงผลให้ถูกต้องและเป็นไปตามที่เราได้ออกแบบไว้

บทที่ 4

การทดสอบโปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์

4.1 การทดสอบโปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์

เป็นการทดสอบเพื่อตรวจสอบความถูกต้องของข้อมูลที่ส่งออกมาจากพอร์ต การรับข้อมูล การประมวลผลและการแสดงผลของโปรแกรมโดยมีขั้นตอนดังต่อไปนี้

4.1.1 การติดตั้งชุดอุปกรณ์เพื่อทดสอบโปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์

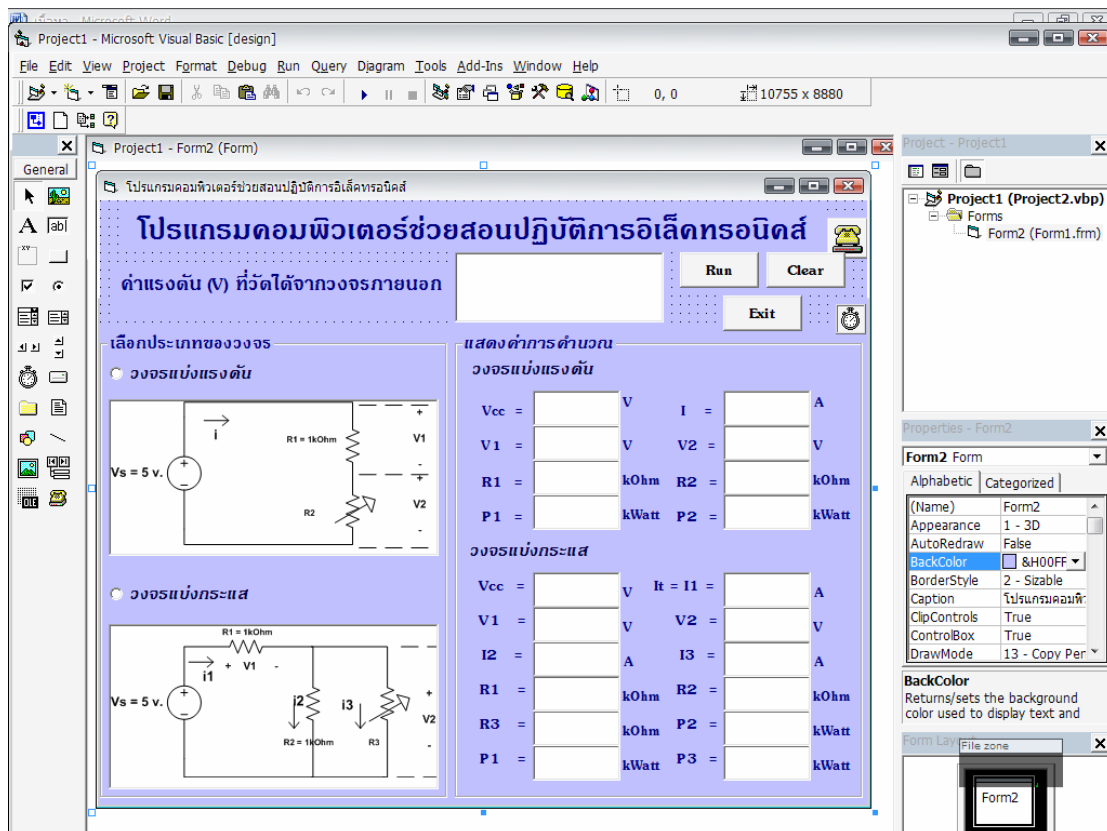
รูปที่ 4.1 แสดงถึงการต่ออุปกรณ์เพื่อทำการทดสอบโปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์



รูปที่ 4.1 ชุดอุปกรณ์ที่ใช้ในการทดสอบโปรแกรม

4.1.2 การเปิดใช้งานโปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์

ทำการเปิดโปรแกรม Microsoft Visual Basic 6.0 เพื่อใช้งานโปรแกรมที่ทำการเขียนไว้แล้ว ในการทดสอบสามารถเปิดได้โดยตรงจากไฟล์นามสกุล .EXE



รูปที่ 4.2 แสดงโปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์บนหน้าต่างโปรแกรม Visual Basic 6.0

4.1.3 การทดสอบโปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์

โดยวงจรที่นำมาทดสอบโปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์เป็นวงจรแบ่งแรงดัน โดยที่เราจะทำการวัดค่าแรงดันที่ตกคร่อม R_2 และทำการแสดงค่าแรงดันที่วัดได้นี้ไปแสดงผลและคำนวณหาค่าตัวแปรต่างๆที่โปรแกรม Microsoft Visual Basic 6.0 ได้ผลแสดงดังรูปที่ 4.3, รูปที่ 4.4, รูปที่ 4.5, รูปที่ 4.6, รูปที่ 4.7 และรูปที่ 4.8 พร้อมทั้งตารางการเปรียบเทียบระหว่างผลการทดลองและผลที่ได้จากการคำนวณ ตามตารางที่ 4.1 และตารางที่ 4.2

โปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์

โปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์

ค่าแรงดัน (V) ที่วัดได้จากวงจรภายนอก **2.5** Run Clear
Exit

เลือกประเภทของวงจร

วงจรแบ่งแรงดัน



$V_s = 5\text{ v.}$

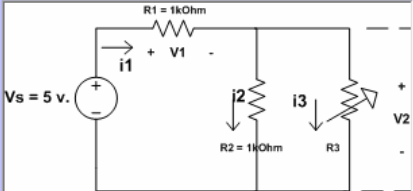
$R_1 = 1\text{ kOhm}$

R_2

V_1

V_2

วงจรแบ่งกระแส



$V_s = 5\text{ v.}$

$R_1 = 1\text{ kOhm}$

$R_2 = 1\text{ kOhm}$

R_3

V_1

V_2

แสดงค่าการคำนวณ

วงจรแบ่งแรงดัน

Vcc =	5	V	I =	2.5	A
V1 =	2.5	V	V2 =	2.5	V
R1 =	1	kOhm	R2 =	1	kOhm
P1 =	6.25	kWatt	P2 =	6.25	kWatt

วงจรแบ่งกระแส

Vcc =		V	I _t = I ₁ =		A
V1 =		V	V2 =		V
I2 =		A	I3 =		A
R1 =		kOhm	R2 =		kOhm
R3 =		kOhm	P2 =		kWatt
P1 =		kWatt	P3 =		kWatt

รูปที่ 4.3 ผลการทดสอบวงจรแบ่งแรงดันโดยใช้ค่า $R_2 = 1\text{ kOhm}$

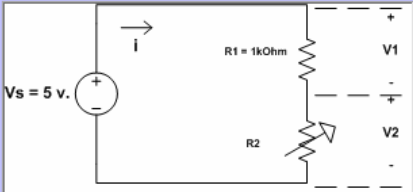
โปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์

โปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์

ค่าแรงดัน (V) ที่วัดได้จากวงจรภายนอก **4.36** Run Clear
Exit

เลือกประเภทของวงจร

วงจรแบ่งแรงดัน



$V_s = 5\text{ v.}$

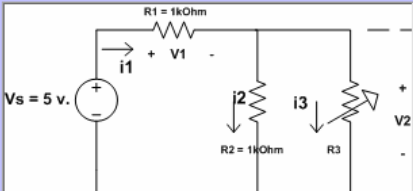
$R_1 = 1\text{ kOhm}$

R_2

V_1

V_2

วงจรแบ่งกระแส



$V_s = 5\text{ v.}$

$R_1 = 1\text{ kOhm}$

$R_2 = 1\text{ kOhm}$

R_3

V_1

V_2

แสดงค่าการคำนวณ

วงจรแบ่งแรงดัน

Vcc =	5	V	I =	.6399999	A
V1 =	.6399999	V	V2 =	4.36	V
R1 =	1	kOhm	R2 =	6.812501	kOhm
P1 =	.4095998	kWatt	P2 =	2.7904	kWatt

วงจรแบ่งกระแส

Vcc =		V	I _t = I ₁ =		A
V1 =		V	V2 =		V
I2 =		A	I3 =		A
R1 =		kOhm	R2 =		kOhm
R3 =		kOhm	P2 =		kWatt
P1 =		kWatt	P3 =		kWatt

รูปที่ 4.4 ผลการทดสอบวงจรแบ่งแรงดันโดยใช้ค่า $R_2 = 6.8\text{ kOhm}$

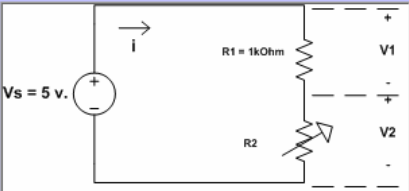
โปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์

โปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์

ค่าแรงดัน (V) ที่วัดได้จากวงจรภายนอก **4.55** Run Clear
Exit

เลือกประเภทของวงจร

- วงจรแบ่งแรงดัน



แสดงค่าการคำนวณ

วงจรแบ่งแรงดัน

Vcc =	5	V	I =	.4499998	A
V1 =	.4499998	V	V2 =	4.55	V
R1 =	1	kOhm	R2 =	10.11112	kOhm
P1 =	.2024998	kWatt	P2 =	2.047499	kWatt

วงจรแบ่งกระแส

Vcc =		V	It = I1 =		A
V1 =		V	V2 =		V
I2 =		A	I3 =		A
R1 =		kOhm	R2 =		kOhm
R3 =		kOhm	P2 =		kWatt
P1 =		kWatt	P3 =		kWatt

รูปที่ 4.5 ผลการทดสอบวงจรแบ่งแรงดันโดยใช้ค่า $R_2 = 10 \text{ kOhm}$

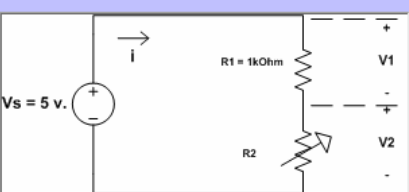
โปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์

โปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์

ค่าแรงดัน (V) ที่วัดได้จากวงจรภายนอก **1.67** Run Clear
Exit

เลือกประเภทของวงจร

- วงจรแบ่งกระแส



แสดงค่าการคำนวณ

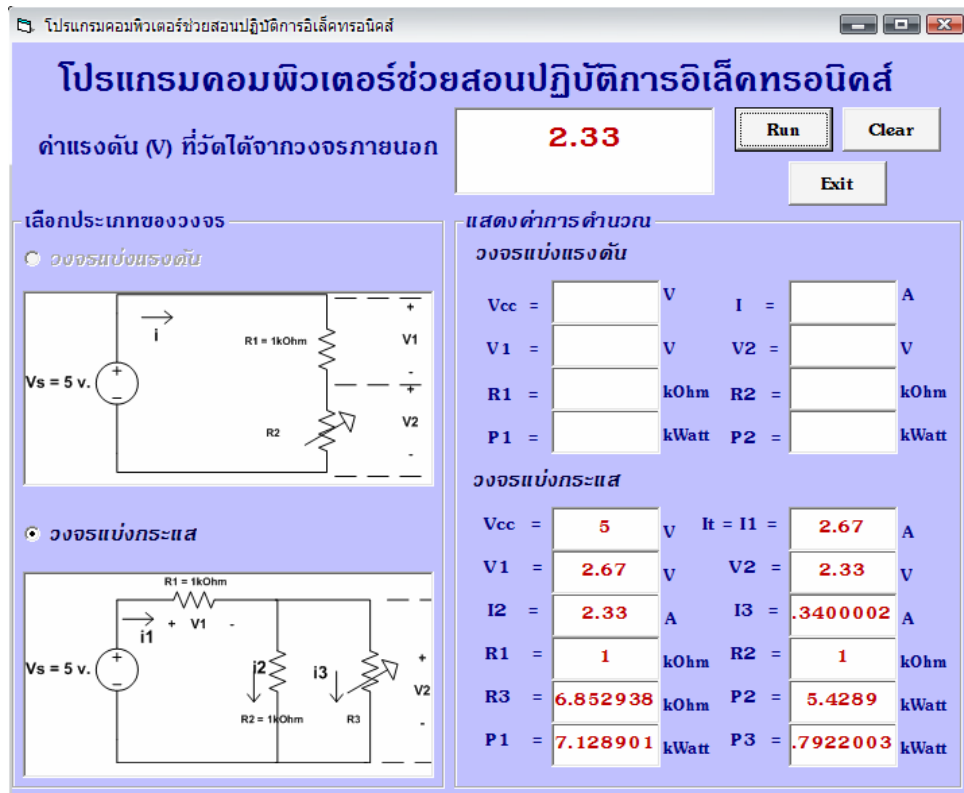
วงจรแบ่งแรงดัน

Vcc =		V	I =		A
V1 =		V	V2 =		V
R1 =		kOhm	R2 =		kOhm
P1 =		kWatt	P2 =		kWatt

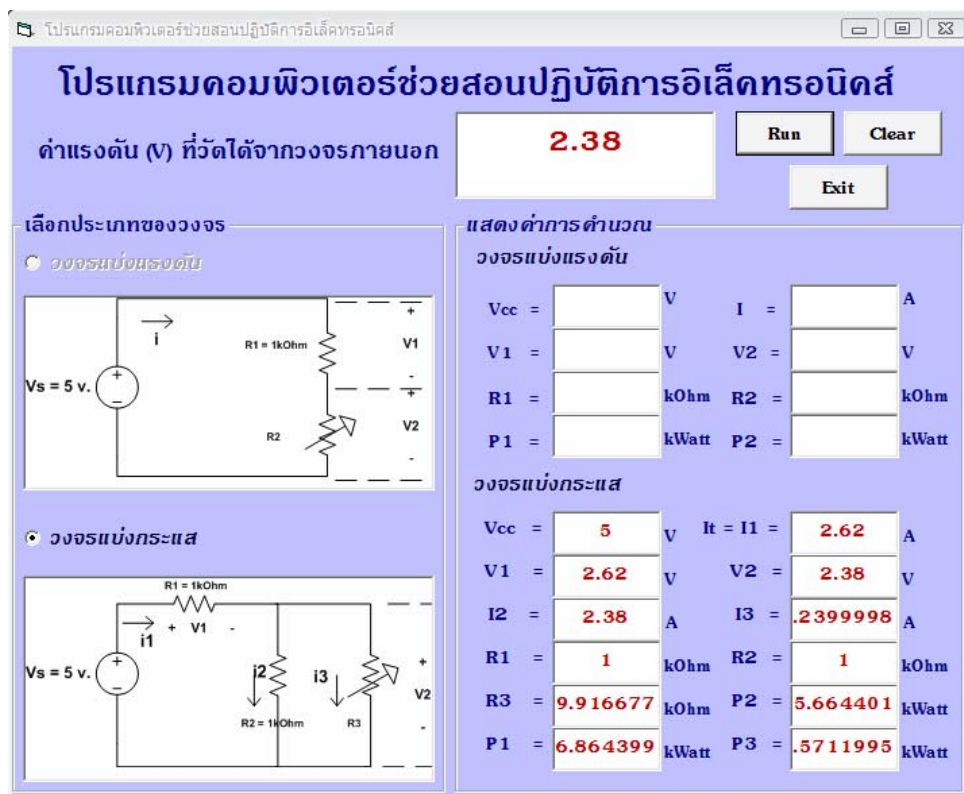
วงจรแบ่งกระแส

Vcc =	5	V	It = I1 =	3.33	A
V1 =	3.33	V	V2 =	1.67	V
I2 =	1.67	A	I3 =	1.66	A
R1 =	1	kOhm	R2 =	1	kOhm
R3 =	1.006024	kOhm	P2 =	2.7889	kWatt
P1 =	11.0889	kWatt	P3 =	2.7722	kWatt

รูปที่ 4.6 ผลการทดสอบวงจรแบ่งกระแสโดยใช้ค่า $R_3 = 1 \text{ kOhm}$



รูปที่ 4.7 ผลการทดสอบวงจรแบ่งกระแสโดยใช้ค่า $R_3 = 6.8\text{ kOhm}$



รูปที่ 4.8 ผลการทดสอบวงจรแบ่งกระแสโดยใช้ค่า $R_3 = 10\text{ kOhm}$

ตารางที่ 4.1 เปรียบเทียบระหว่างผลการทดสอบและผลการคำนวณของวงจรแบ่งแรงดัน

	ครั้งที่ 1			ครั้งที่ 2			ครั้งที่ 3		
	การวัด	การ คำนวณ	ผิด พลาด (%)	การวัด	การ คำนวณ	ผิด พลาด (%)	การวัด	การ คำนวณ	ผิด พลาด (%)
R₂ (kOhm)	1.000	1.000	0.000	6.812	6.8	0.176	10.111	10.000	1.110
I (mA)	2.500	2.500	0.000	0.639	0.641	0.312	0.449	0.454	1.101
V₁ (volt)	2.500	2.500	0.000	0.639	0.641	0.312	0.449	0.454	1.101
V₂ (volt)	2.500	2.500	0.000	4.360	4.358	0.045	4.550	4.540	0.220
P₁ (kWatt)	6.250	6.250	0.000	0.409	0.410	0.244	0.202	0.206	1.941
P₂ (kWatt)	6.250	6.250	0.000	2.790	2.793	0.107	2.047	2.060	0.631

ตารางที่ 4.2 เปรียบเทียบระหว่างผลการทดสอบและผลการคำนวณของวงจรแบ่งกระแส

	ครั้งที่ 1			ครั้งที่ 2			ครั้งที่ 3		
	การวัด	การ คำนวณ	ผิด พลาด (%)	การวัด	การ คำนวณ	ผิด พลาด (%)	การวัด	การ คำนวณ	ผิด พลาด (%)
R₃ (kOhm)	1.006	1.000	0.600	6.852	6.800	0.764	9.916	10.000	0.840
I₁ (mA)	3.330	3.330	0.000	2.670	2.671	0.037	2.620	2.619	0.038

	ครั้งที่ 1			ครั้งที่ 2			ครั้งที่ 3		
	การวัด	การ คำนวณ	ผิด พลาด (%)	การวัด	การ คำนวณ	ผิด พลาด (%)	การวัด	การ คำนวณ	ผิด พลาด (%)
I_2 (mA)	1.670	1.670	0.000	2.330	2.328	0.085	2.380	2.381	0.041
I_3 (mA)	1.660	1.660	0.000	0.340	0.342	0.584	0.239	0.2381	0.377
V_1 (volt)	3.330	3.330	0.000	2.670	2.671	0.037	2.620	2.619	0.038
V_2 (volt)	1.670	1.670	0.000	2.330	2.328	0.085	2.380	2.381	0.041
P_1 (kWatt)	11.088	11.088	0.000	7.128	7.134	0.084	6.864	6.859	0.072
P_2 (kWatt)	2.788	2.788	0.000	5.428	5.421	0.129	5.664	5.669	0.088
P_3 (kWatt)	2.772	2.772	0.000	0.792	0.797	0.627	0.571	0.566	0.880

4.2 สรุปผล

ผลการทดสอบโปรแกรมด้วยการเชื่อมต่อกับชุดอุปกรณ์โดยมีการต่อวงจรไฟฟ้า 2 แบบ คือ วงจรแบ่งแรงดันและวงจรแบ่งกระแส โดยทำการทดสอบที่ละวงจร จากผลการทดสอบโปรแกรมพบว่าผลที่ได้จากการทดสอบและผลที่ได้จากการคำนวณมีค่าแตกต่างกันไม่มากนัก มีเพียงค่าผิดพลาดที่จุดทศนิยมที่ใช้ในการคำนวณเท่านั้น สามารถเปรียบเทียบได้จากค่าเปอร์เซ็นต์ความผิดพลาดของแต่ละวงจรที่มีค่าไม่สูงมากนัก นั่นคือโปรแกรมทดสอบชุดนี้สามารถทำงานได้

บทที่ 5

สรุปและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงขั้นตอนการทำงานโดยรวมทั้งหมดของโปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์ และผลการทำงานโดยรวม รวมทั้งปัญหาที่เกิดขึ้นและข้อจำกัดในการทดลอง

5.1 ขั้นตอนการทำงานของโปรแกรมคอมพิวเตอร์ช่วยสอนปฏิบัติการอิเล็กทรอนิกส์

- 5.1.1 รับค่าแรงดันไฟฟ้าที่เป็นสัญญาณดิจิทัลจากวงจรไฟฟ้าของอุปกรณ์เชื่อมต่อภายนอกโดยผ่านทางพอร์ต USB
- 5.1.2 ทำการประมวลผลค่าที่รับเข้ามา โดยแบ่งออกเป็นวงจรแบ่งแรงดันและวงจรแบ่งกระแส
- 5.1.3 แสดงผลการคำนวณของโปรแกรม

5.2 ปัญหาที่พบในขั้นตอนการทำงาน

- 5.2.1 ตัวต้านทานที่นำมาใช้ในการทดลองนั้นเป็นตัวต้านทานที่มีขายอยู่ทั่วไปซึ่งจะมีค่าคลาดเคลื่อนภายในตัวต้านทานเองโดยส่วนใหญ่แล้วจะมีค่าคลาดเคลื่อนอยู่ที่ $+1\%$ และ $+5\%$ จึงทำให้ผลการทดลองที่ได้เกิดค่าคลาดเคลื่อนตามไปด้วย

5.3 ข้อจำกัดของโครงงาน

- 5.3.1 ค่าแรงดันอินพุตที่วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัลสามารถรับได้สูงสุดเพียง 5 โวลต์เท่านั้นจึงทำให้ไม่สามารถทำงานที่ค่าแรงดันสูงมากได้

5.4 ผลที่ได้จากโครงการ

- 5.4.1 ทำให้มีความรู้ความเข้าใจในหลักการทำงานของโปรแกรมเชื่อมต่อคอมพิวเตอร์ช่วยสอนปฏิบัติการและสามารถนำไปประยุกต์ใช้ในอนาคตได้
- 5.4.2 ได้เรียนรู้วิธีการออกแบบโปรแกรม และขั้นตอนการออกแบบโปรแกรม
- 5.4.3 ได้ชุดอุปกรณ์ต้นแบบในการเชื่อมต่อเพื่อทดสอบโปรแกรม

5.5 แนวทางการพัฒนาต่อ

- 5.5.1 ปรับปรุงชุดอุปกรณ์เชื่อมต่อภายนอกให้สามารถวัดค่าและส่งค่าได้หลายรูปแบบ เช่น สามารถวัดค่ากระแสไฟฟ้า ค่าความต้านทานและค่าความถี่ เป็นต้น
- 5.5.2 ปรับปรุงและพัฒนาโปรแกรมที่ใช้แสดงผลให้มีความหลากหลายและสวยงามด้วยโปรแกรมประยุกต์ต่างๆ เช่น LabVIEW และ MATLAB เป็นต้น

บรรณานุกรม

- [1] <http://www.thaiio.com>
<http://www.thanom.net/vb1.asp>
<http://www.Vbthailand.com>
- [2] อภิชาติ ภู่วลัฒ , “เริ่มต้นเขียนโปรแกรมติดต่อและควบคุมฮาร์ดแวร์ด้วย Visual Basic.”, พิมพ์ครั้งที่ 1 , Infopress Developer Book, นนทบุรี: 2546
- [3] สราวุฒิ สุจิตจร และกิตติ อัครกัจจมงคล (แปลและเรียบเรียง). “วงจรไฟฟ้า Electric Circuit.” พิมพ์ครั้งที่ 1, เพียร์สัน เอ็ดดูเคชั่น อินโดไชน่า, กรุงเทพฯ: 2547

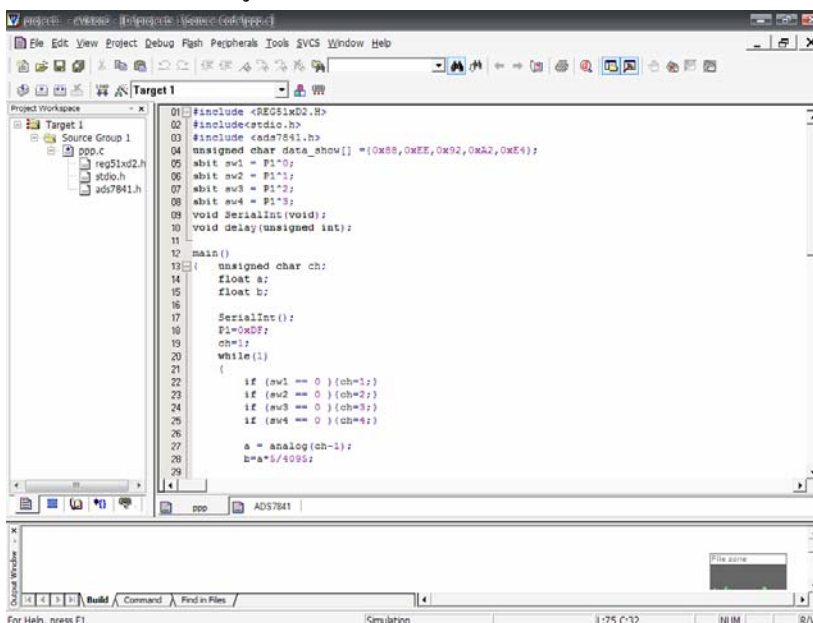
ภาคผนวก (ก)

การโหลดโปรแกรมทดสอบอุปกรณ์เชื่อมต่อและการติดตั้งไดรเวอร์ USB


การทดสอบอุปกรณ์เชื่อมต่อภายนอก

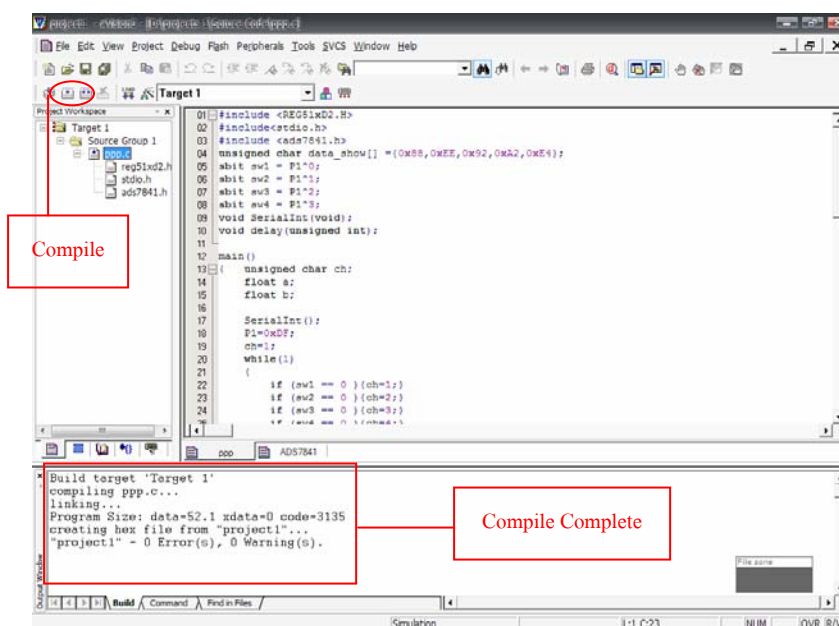
การเขียนโปรแกรมทดสอบ

เปิดโปรแกรม Keil70 ดังรูปเพื่อใช้ในการเขียนโปรแกรม



รูปแสดง โปรแกรม Keil70

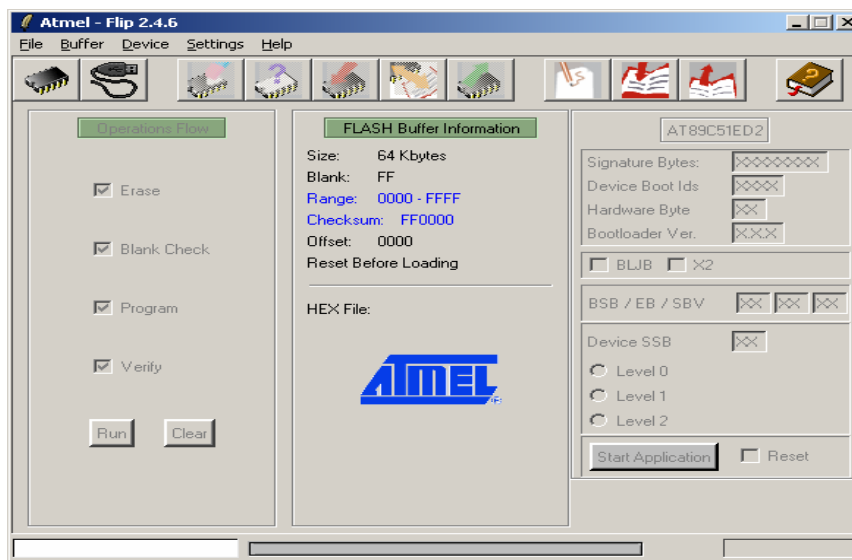
เมื่อทำการเขียนโปรแกรมเสร็จแล้วทำการคอมไพล์ (Compile) โดยกดปุ่ม  ถ้าสำเร็จ โดยไม่เกิดความผิดพลาดของโปรแกรมที่เขียนจะได้หน้าต่างดังรูป



รูปแสดงการคอมไพล์สมบูรณ์

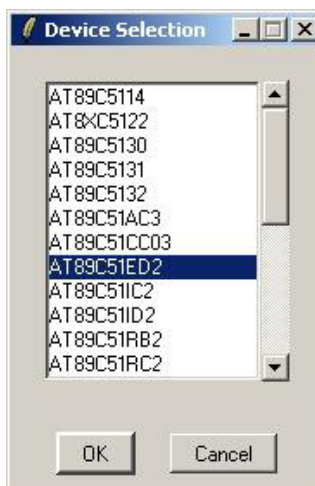
การโหลดโปรแกรมทดสอบ

เปิดโปรแกรม FLIP จะได้หน้าต่างดังรูปข้างล่าง ใช้ในการโหลดโปรแกรมลงไมโครคอนโทรลเลอร์




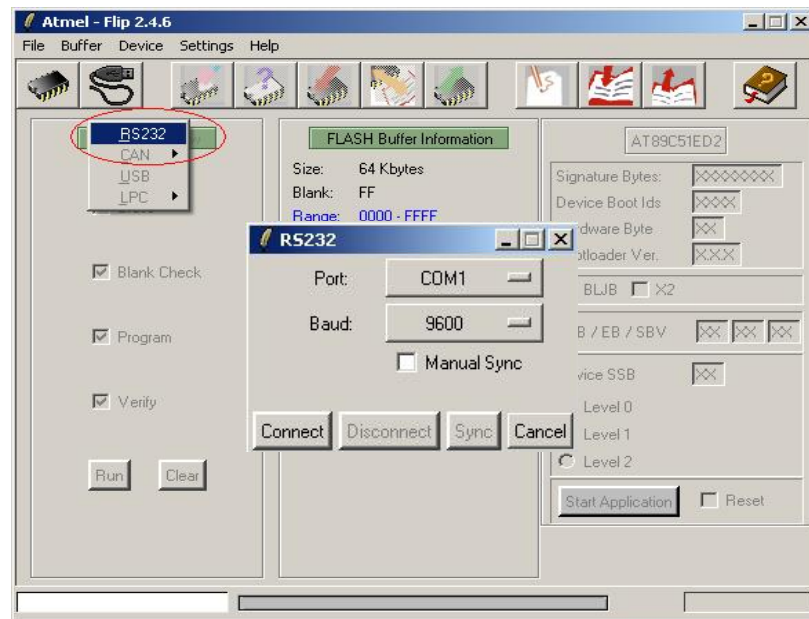
รูปหน้าต่างโปรแกรม FLIP

เลือกชนิดของไมโครคอนโทรลเลอร์ AT89C51ED2 ที่  จะได้หน้าต่างดังรูป




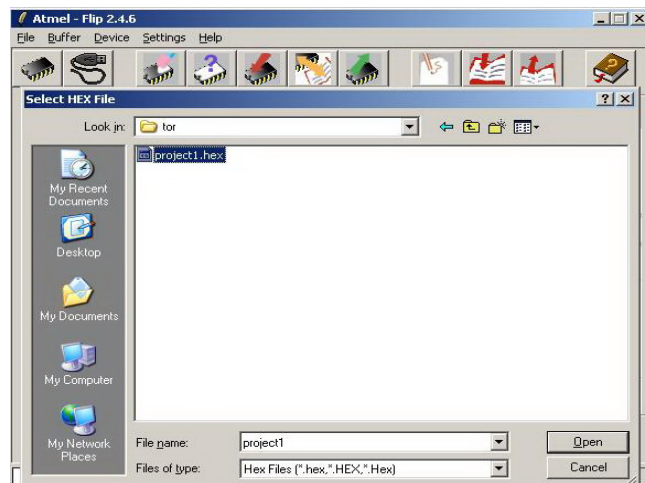
รูปหน้าต่างเลือกชนิดไมโครคอนโทรลเลอร์

ตั้งค่าการเชื่อมต่อระหว่างโปรแกรมกับไมโครคอนโทรลเลอร์ที่  แล้วทำการเลือกเป็นแบบ RS-232, เลือกพอร์ต COM และตั้งค่าอัตราการส่งข้อมูล (Baud Rate) แล้วกด Connect ดังรูป




รูปการตั้งค่าการเชื่อมต่อ

เมื่อตั้งค่าการเชื่อมต่อเรียบร้อยแล้วนั้น ทำการโหลดไฟล์ .HEX ที่ได้จากการคอมไพล์ ในโปรแกรม Keil70 โดยเลือกที่  จะปรากฏหน้าต่างดังรูป



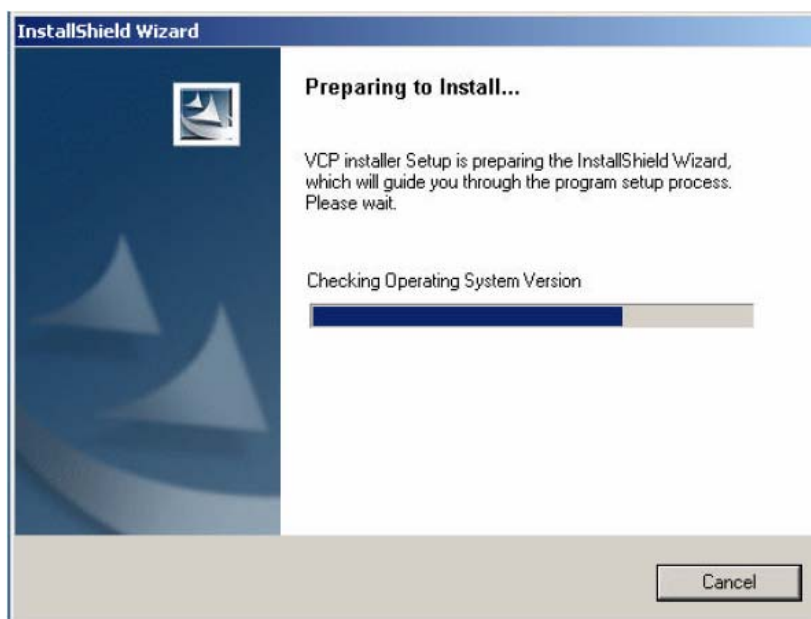
รูปหน้าต่างการโหลดไฟล์ .HEX

เมื่อเลือกไฟล์ .HEX ลงเรียบร้อยแล้วให้กดปุ่ม  โปรแกรมจะทำการ โหลดไฟล์ที่เลือกบอร์ดไมโครคอนโทรลเลอร์โดยอัตโนมัติ

การติดตั้งไดรเวอร์วงจรเชื่อมต่อ USB

การติดตั้งไดรเวอร์ของ TUSB3410 แบ่งเป็น 3 ขั้นตอนดังต่อไปนี้

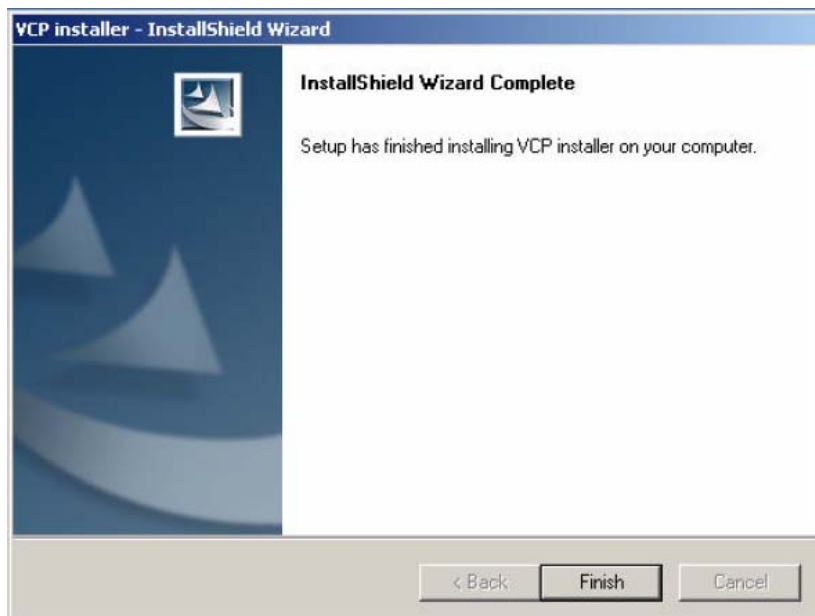
ขั้นตอนที่ 1 ทำการติดตั้งไดรเวอร์ด้วยไฟล์ setup.exe ซึ่งมีขั้นตอนดังรูปด้านล่างตามลำดับ



(ก)



(ข)



(ก)

(ก) Driver Installation Progress

(ข) License agreement

(ค) Pre-installation of drivers complete

ขั้นตอนที่ 2 ต่อวงจรเชื่อมต่อ USB เข้ากับเครื่องคอมพิวเตอร์ จะปรากฏหน้าต่างดังรูป ให้ติดตั้งฮาร์ดแวร์ (TUSB3410 device) เลือกที่ “Install the software automatically”



(ก)



(๗)

(ก) Found New Hardware (“TUSB3410 Device”)

(ข) Device installation complete

ขั้นตอนที่ 3 ติดตั้งไดรเวอร์ของ “USB-Serial Port” ดังรูป เลือกที่ “Install the software automatically”



(ก)

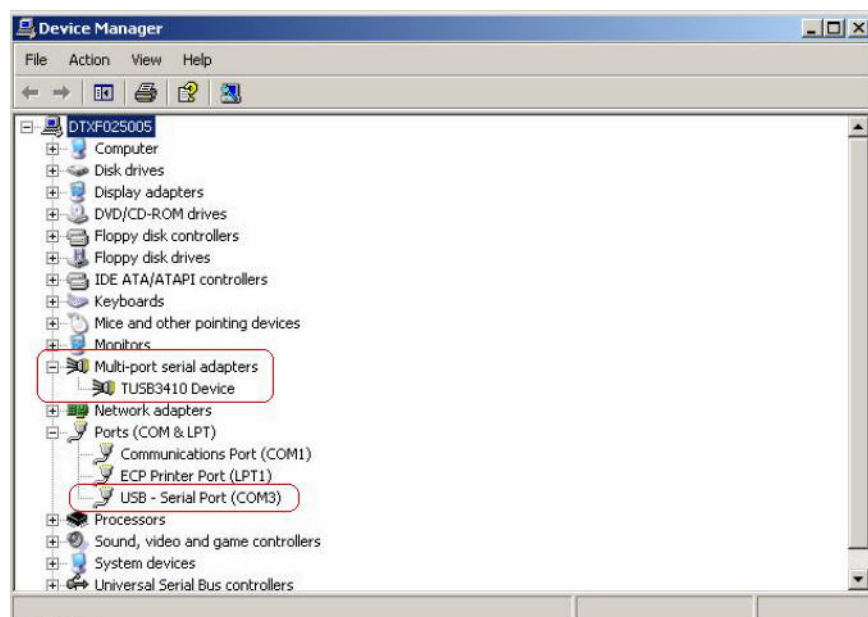


(ข)

(ก) Found New Hardware (USB – Serial Port)

(ข) USB-Serial port installation complete

เมื่อทำการติดตั้งไดรเวอร์เสร็จเรียบร้อยแล้ว ทำการตรวจสอบดูว่าอุปกรณ์พร้อมใช้งานหรือไม่ที่ My Computer-> Properties -> Hardware -> Device Manager ดังรูป



รูปแสดงหน้าต่าง Windows XP Device Manager

ภาคผนวก (ข)

โปรแกรมควบคุมและสั่งการให้แสดงผลผ่านทางหน้าจอกอมพิวเตอร์ ของ
“วงจรเชื่อมต่อภายนอก”

```
#include <REG51xD2.H>
#include<stdio.h>
#include <ads7841.h>

unsigned char data_show[]={0x88,0xEE,0x92,0xA2,0xE4};

sbit sw1 = P1^0;
sbit sw2 = P1^1;
sbit sw3 = P1^2;
sbit sw4 = P1^3;

void SerialInt(void);
void delay(unsigned int);

main()
{ unsigned char ch;
  float a;
  float b;
  SerialInt();
  P1=0xDF;
  ch=1;
  while(1)
  {
    if (sw1 == 0 ){ch=1;}
    if (sw2 == 0 ){ch=2;}
    if (sw3 == 0 ){ch=3;}
    if (sw4 == 0 ){ch=4;}

    a = analog(ch-1);
    b=a*5/4095;
```

```

        if (ch == 1 ){P0=data_show[ch]; putchar(12);
        printf("%.2f\n",b);
delay(60000);
}

        if (ch == 2 ){P0=data_show[ch]; putchar(12);
        printf("%.2f\n",b);
        delay(60000);
}

        if (ch == 3 ){P0=data_show[ch]; putchar(12);
        printf("%.2f\n",b);
        delay(60000);
}

        if (ch == 4 ){P0=data_show[ch]; putchar(12);
        printf("%.2f\n",b);
        delay(60000);
}
}

void SerialInt(void)
{
    TR1=0;
    SCON = 0x52; /* SCON0 */ /* setup serial port control */
    TMOD = 0x21;
    TCON = 0x69; /* TCON */
    TH1 = 0xFB; // 28800 at Clock 32 Mhz
    TR1=1;
}

void delay(unsigned int a)
{
    unsigned int i,b;

```



```
for(i=0;i<=a;i++)
    {      b=10000;
      while(b<=0)
          b--;
    }
return;
}
```

โปรแกรมควบคุมการทำงานของวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล

```
* Filename : ads7841.h
* Hardware : Controller -> P89V51RD2
*   XTAL    -> 18.432 MHz
*   Mode    -> 6 Clock/MC
*   I/O     : DCLK P3.3
*   CS      P3.4
*   DIN     P3.5
*   DOUT    P3.7
```

.....

```
#define DCLK  P3_3 /* DCLK pin */
#define CS    P3_4 /* CS pin */
#define DIN   P3_5 /* DIN pin */
#define DOUT  P3_7 /* DOUT pin */

int analog(char ch)
{
    unsigned char i, crt_byte;      /* Variable for counter */
    unsigned int dat;              /* Variable for storing analog value */
    switch(ch)                     /* Check selected channel for reading */
    {
        case 0 : crt_byte = 0x97; /* Select analog channel 0 */
                break;
        case 1 : crt_byte = 0xD7; /* Select analog channel 1 */
                break;
        case 2 : crt_byte = 0xA7; /* Select analog channel 2 */
                break;
        case 3 : crt_byte = 0xE7; /* Select analog channel 3 */
                break;
    }
}
```

```

        default: crt_byte = 0x97; /* Select analog channel 0 */
                break;
    }
    dat = 0;
    DCLK = 0;    /* Clear DCLK pin */
    DIN = 0;     /* Clear DIN pin */
    CS = 0;     /* Clear CS pin */

    for(i=0; i<8; i++) /* Loop to send control byte for ADS7841 */
    {
        DIN = crt_byte & 0x80; /* Send data bit to ADS7841 */
        DCLK = 0; /* Clear clock pin */
        DCLK = 1; /* Set clock pin */
        crt_byte = crt_byte<<1; /* Shift bit once next time */
    }

    DCLK = 0; /* Clear clock pin */
    DCLK = 1; /* Set clock pin */

    for(i=0; i<12; i++) /* Loop to read 12-bit digital value from ADS7841 */
    {
        DCLK = 0; /* Clear clock pin */
        DCLK = 1; /* Set clock pin */
        dat = dat<<1; /* Shift bit once to store data next bit */
        dat = dat | DOOUT; /* Read data bit from DOOUT pin */
    }

```

```
for(i=0;i<4;i++)          /* Generate rising edge clock for success frame */
{
    DCLK = 0;             /* Clear clock pin */
    DCLK = 1;             /* Set clock pin */
}
    CS = 1;               /* Set CS pin */
return dat;
}
```

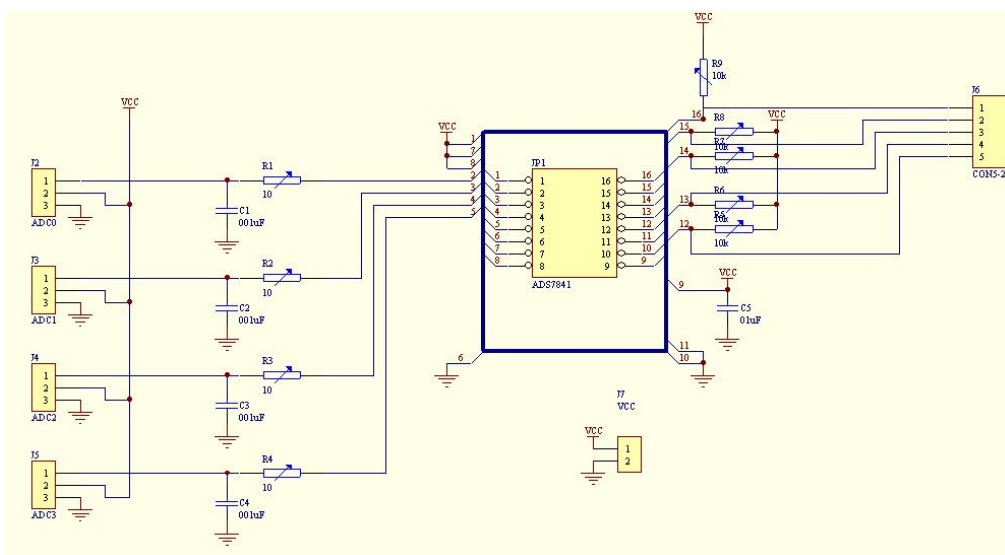
ภาคผนวก (ค)

วงจรภายในชุดอุปกรณ์เชื่อมต่อภายนอก

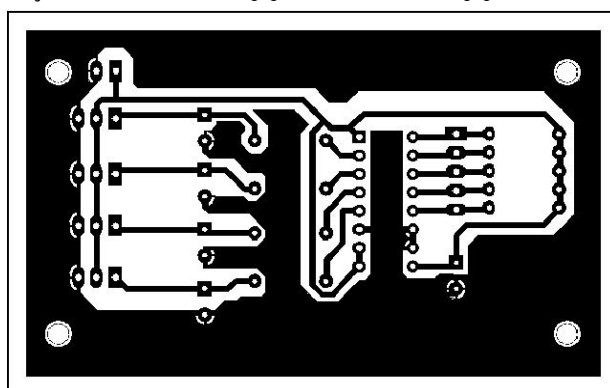
วงจรงแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล (A/D)

การออกแบบลายวงจร A/D ลงบนแผ่นปริ้นท์ (PCB)

จากรูปได้แสดงถึงวงจรที่ได้ทำการออกแบบวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัลไว้ โดยเลือกใช้ไอซีที่มีคุณสมบัติตามต้องการ จากนั้นทำการออกแบบลายวงจรด้วยโปรแกรม Protel 99SE [4] เพื่อที่ได้ลายวงจรเป็นแผ่นลายทองแดงที่จะสามารถนำมาใช้งานได้ตามรูปถัดไป



รูปแสดงวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล

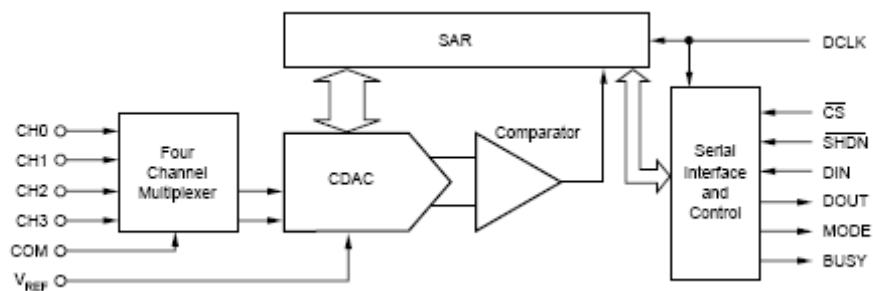


รูปแสดงลายวงจรที่ได้จากโปรแกรม Protel99SE บนแผ่นปริ้นท์

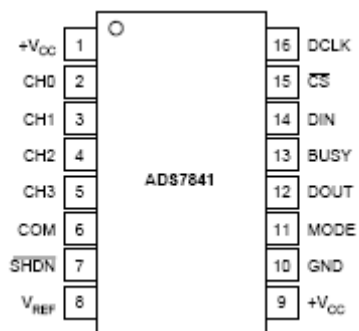
การเลือกใช้ไอซีในการออกแบบ

ในวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัลนั้นได้เลือกใช้ ไอซี เบอร์ ADS7841 ในการแปลงสัญญาณ ซึ่งมีฟังก์ชันการทำงานและการจัดเรียงขาของไอซี ดังรูป (ก) และ (ข)

ตามลำดับ โดยที่ไอซีนี้สามารถรับสัญญาณอินพุตอนาลอกได้ทั้งหมด 4 ช่อง มีอัตราในการสุ่มเลือกตัวอย่าง 200 กิโลเฮิร์ต มีความแม่นยำในการรับสัญญาณ 12 บิต



(ก)



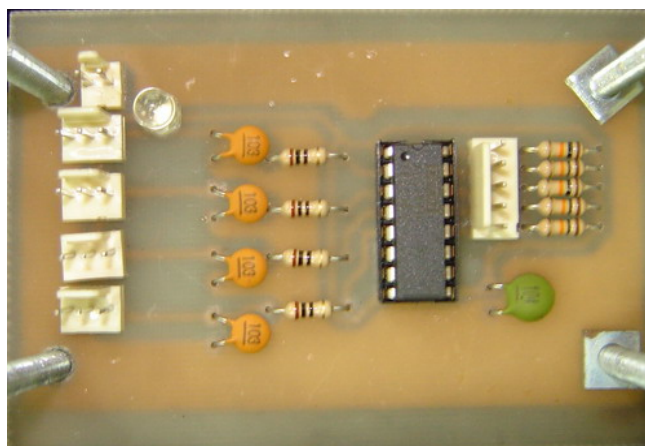
(ข)

(ก) Functional Diagram ของ ADS7841

(ข) Top View ของ ADS7841

การลงอุปกรณ์อิเล็กทรอนิกส์ตามที่ได้ออกแบบไว้

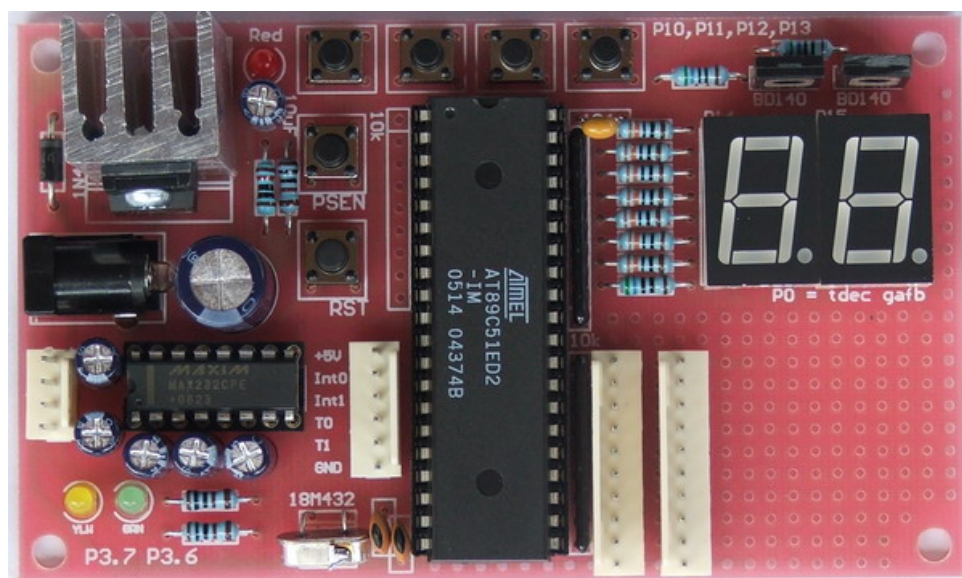
นำอุปกรณ์ที่ได้ออกแบบและจัดเตรียมมาประกอบลงบนแผ่นปริ้นท์ที่ได้ดังรูป



รูปแสดงชุดวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล

วงจรไมโครคอนโทรลเลอร์

ได้เลือกใช่วงจรไมโครคอนโทรลเลอร์สำเร็จรูป ในตระกูล MCS-51 ที่ใช้ไอซีเบอร์ AT89C51ED2 ของ ATMEL ดังที่ได้แสดงวงจรรวมไว้ดังรูป

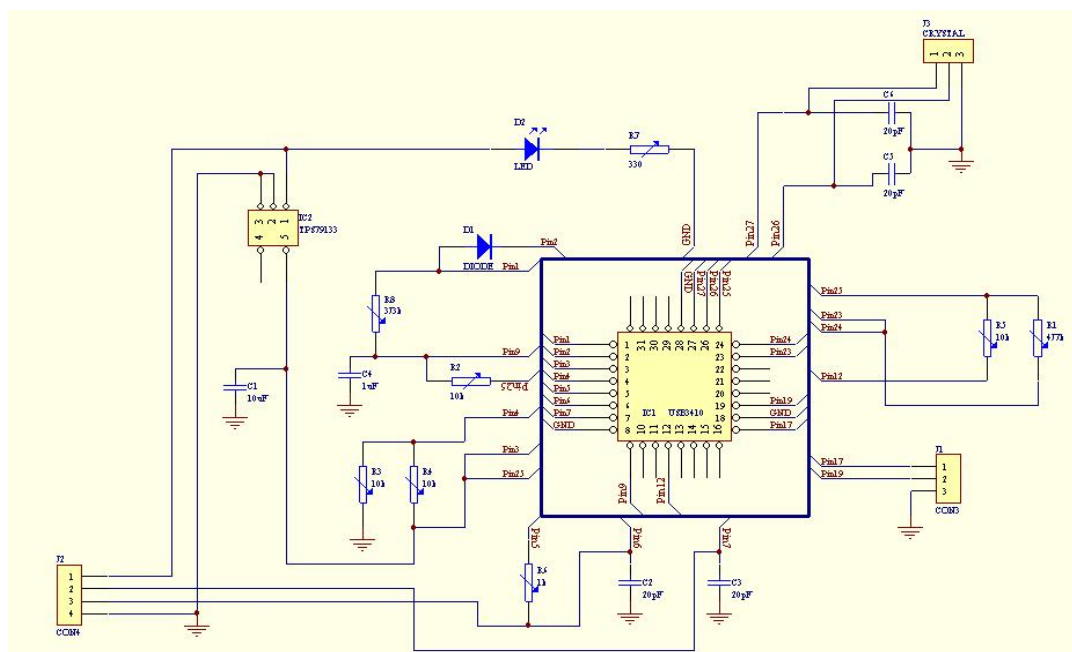


รูปแสดงชุดวงจรไมโครคอนโทรลเลอร์ AT89C51ED2

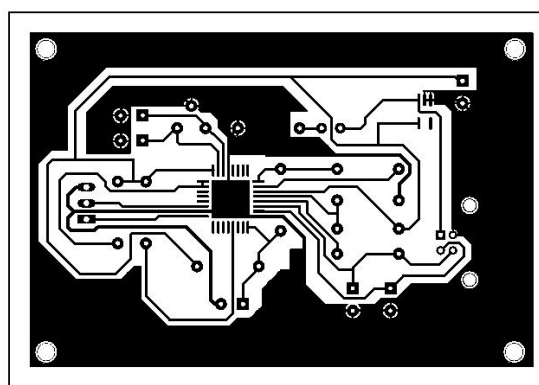
วงจรเชื่อมต่อ USB

การออกแบบลายวงจรเชื่อมต่อ USB ลงบนแผ่นปริ้นท์

จากรูปได้แสดงถึงวงจรเชื่อมต่อ USB ที่ได้ทำการออกแบบไว้โดยเลือกใช้ไอซีที่มีคุณสมบัติตามต้องการ จากนั้นทำการออกแบบลายวงจรด้วยโปรแกรม Protel 99SE เพื่อที่ได้ลายวงจรเป็นแผ่นลายทองแดงที่จะสามารถนำมาใช้งานได้ตามรูปข้างล่างตามลำดับ



รูปแสดงวงจรเชื่อมต่อ USB

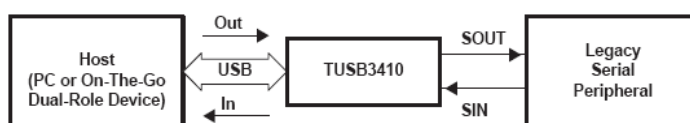


รูปแสดงลายวงจรที่ได้จากโปรแกรม Protel99SE บนแผ่นปริ้นท์

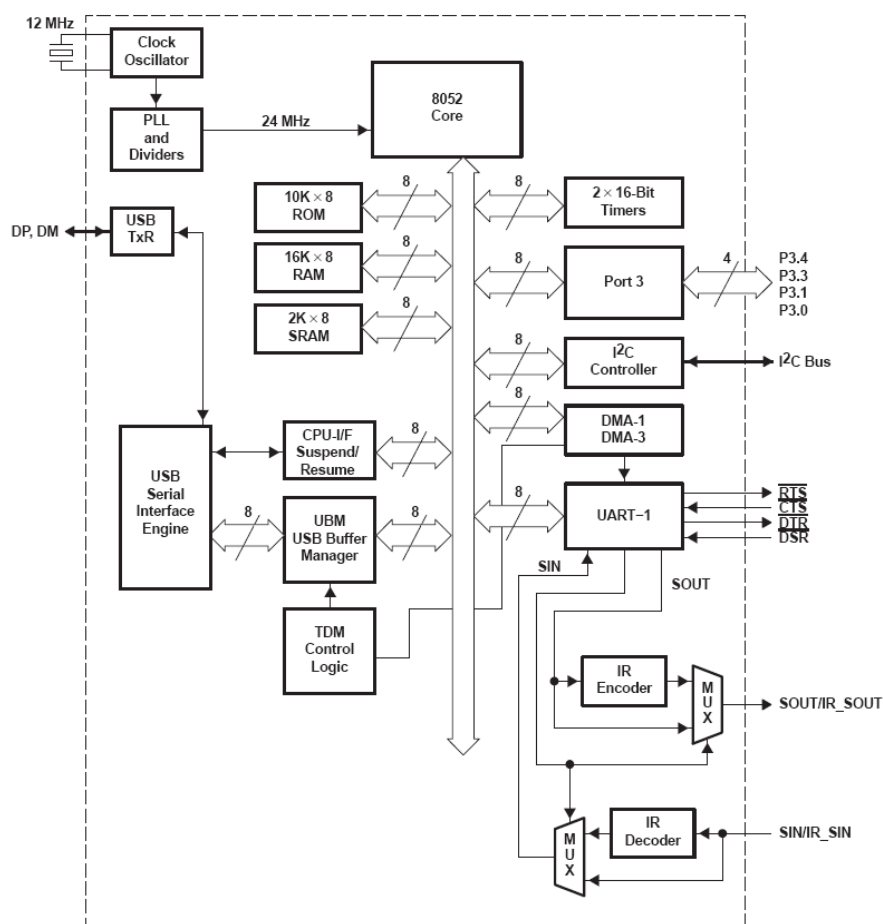
การเลือกใช้ไอซีในการออกแบบ

ในวงจรเชื่อมต่อ USB ได้เลือกใช้ไอซีจำนวน 2 ตัว คือ TUSB3410 และ TPS79133

TUSB3410 มีคุณสมบัติเป็นตัวเชื่อมต่อระหว่างอุปกรณ์ที่ส่งสัญญาณแบบอนุกรม ให้สามารถเชื่อมต่อกับคอมพิวเตอร์โดยผ่านทางพอร์ต USB ได้ และทำหน้าที่แทน พอร์ต RS-232 ได้ซึ่งมีแผนผังการทำงานดังรูป (ก) และ (ข) และมีการจัดเรียงขาของไอซี ดังรูปข้างล่างตามลำดับ



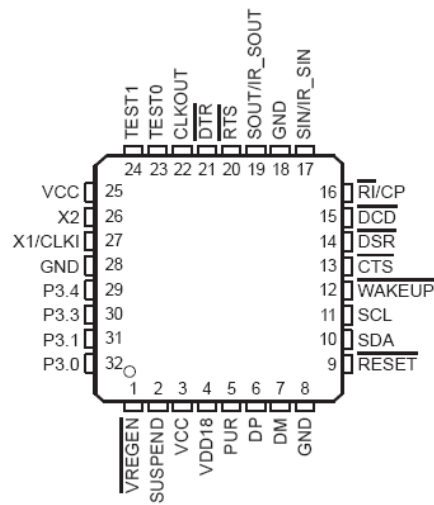
(ก)



(ข)

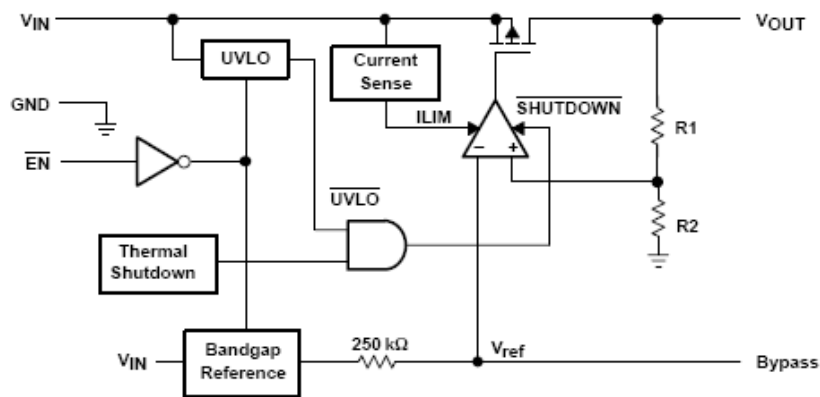
(ก) Data Flow

(ข) USB-to-Serial (Single Channel) Controller Block Diagram

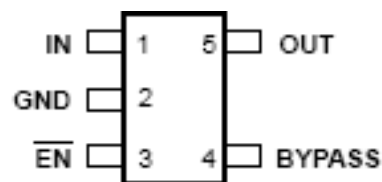


รูปแสดง Top View ของ TUSB3410

TPS79133 มีคุณสมบัติในการ แปลงระดับแรงดันที่ได้รับจากพอร์ต USB ของเครื่องคอมพิวเตอร์ จาก 5 โวลต์เป็น 3 โวลต์เพื่อจ่ายให้กับไอซี TUBS3410 ทำงาน โดยไอซีมีฟังก์การทำงานและการจัดเรียงขา ดังรูป (ก) และรูป (ข) ตามลำดับ



(ก)



(ข)

(ก) Functional Block Diagram

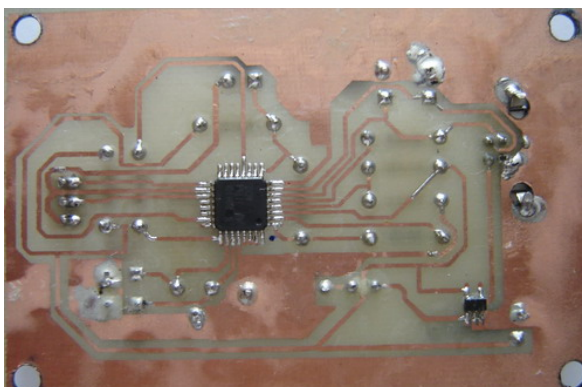
(ข) Top View ของ TPS79133

การลงอุปกรณ์อิเล็กทรอนิกส์ตามที่ได้ออกแบบ

นำอุปกรณ์ที่ได้ออกแบบและจัดเตรียมมาประกอบลงบนแผ่นปริ้นท์ที่ได้ดังรูป (ก) และรูป (ข) ตามลำดับ



(ก)



(ข)

(ก) ชุดวงจรเชื่อมต่อ USB ด้านบน

(ข) ชุดวงจรเชื่อมต่อ USB ด้านล่าง

ประวัติผู้จัดทำ



นางสาวนัฐญา วิลุนละพัน เกิดเมื่อวันที่ 2 พฤษภาคม พ.ศ. 2529 ภูมิลำเนาเดิม
อยู่บ้านเลขที่ 60 ถนนเทศบาล4 ตำบลในเมือง อำเภอเมือง จังหวัดสุรินทร์
สำเร็จการศึกษาระดับชั้นมัธยมศึกษาตอนปลายจากโรงเรียนสิรินธร อำเภอ
เมือง จังหวัดสุรินทร์ ปัจจุบันกำลังศึกษาอยู่ชั้นปีที่ 5 สาขาวิชาวิศวกรรม
โทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี
จังหวัดนครราชสีมา