บทที่ 2
หลักการและทฤษฎี

ระบบการจัดเก็บข้อมูล (Data Acquisition System) เป็นระบบที่ใช้ซอฟท์แวร์ในการควบคุม สามารถแสดงผลเป็นกราฟฟิก และรับข้อมูลที่มีปริมาณมาก อีกทั้งยังทำการประมวลผลแบบอัตโนมัติ โดยที่มีความเร็วในการสุ่มข้อมูลสูงคือความเร็วของการสุ่มข้อมูลอยู่ในช่วง 20,000-250,000 ต่อวินาที

2.1 ส่วนประกอบที่สำคัญของดาต้าแอคควิสิชั่น

2.1.1 พีซี (PC)

2.1.2 ตัวแปลงสัญญาณ (Tranducers)

2.1.3 สภาวะของสัญญาณ (Signal Conditioning)

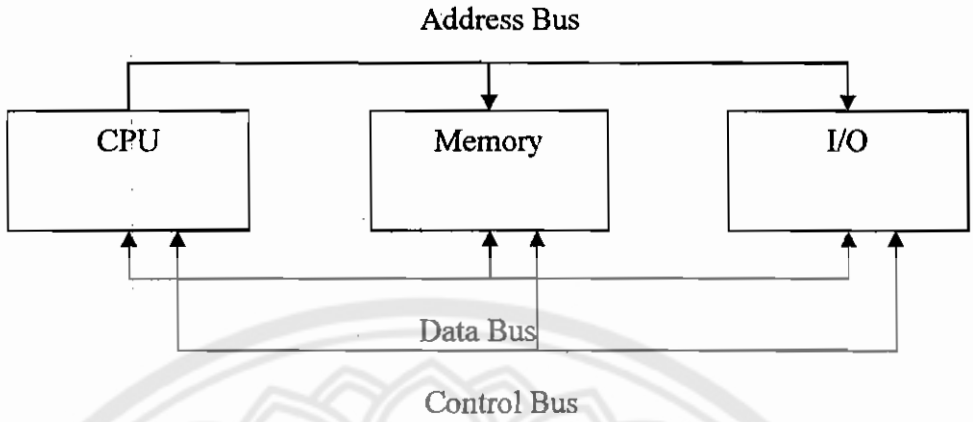
2.1.4 ฮาร์ดแวร์ (DAQ Hardware)

2.1.5 ซอฟแวร์ (Software)

2.2 การจัดระบบโครงสร้างคอมพิวเตอร์

การทำงานภายในคอมพิวเตอร์ทุกเครื่องจะประกอบด้วยส่วนประกอบสามส่วนคือ หน่วยประมวลผลกลางหรือซีพียู (Central Processing Unit : CPU) หน่วยความจำ (Memory) และอุปกรณ์อินพุตเอาต์พุต (I/O Devices) ดังแสดงในรูปที่ 2.1 การทำงานของระบบคอมพิวเตอร์นั้น ซีพียู จะต้องทำงานตามคำสั่งที่เก็บเอาไว้ในหน่วยความจำ จากนั้นจะติดต่อกับผู้ใช้หรือแสดงผลลัพธ์จากการประมวลผลทางอุปกรณ์อินพุตเอาต์พุต อุปกรณ์อินพุตเอาต์พุตที่เราพบโดยทั่วไปได้แก่ คีย์บอร์ด จอแสดงผล เครื่องพิมพ์ เป็นต้น ซีพียู จะติดต่อกับหน่วยความจำและอุปกรณ์อินพุต เอาต์พุตต่างๆ ทางระบบบัส (Bus)

2.3 ส่วนประกอบที่สำคัญของไมโครโปรเซสเซอร์ (MCS - 8051)



รูปที่ 2.1 ส่วนประกอบของคอมพิวเตอร์

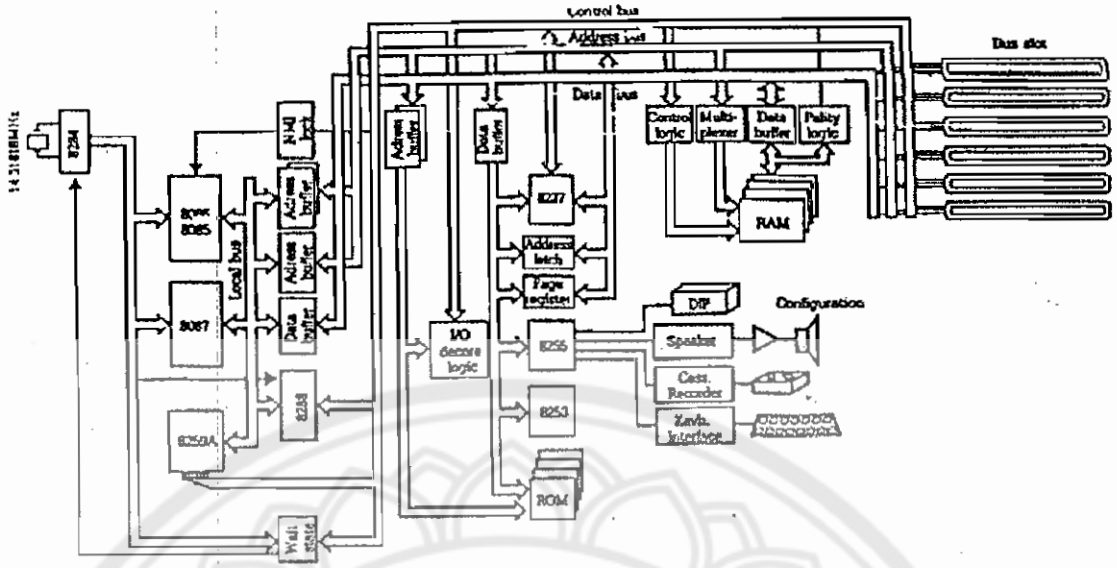
2.3.1 บัส หมายถึง กลุ่มของสัญญาณที่ใช้ในการติดต่อสื่อสารข้อมูลต่างๆ ของระบบคอมพิวเตอร์ ในคอมพิวเตอร์จะมีระบบบัสอยู่สามประเภทได้แก่ แอดเดรสบัส (address bus) บัสข้อมูล (data bus) และบัสควบคุม (control bus)

2.3.1.1 แอดเดรสบัส : จะประกอบด้วยสัญญาณแอดเดรสหลายเส้นขึ้นอยู่กับรุ่นของซีพียู แอดเดรสบัสนี้จะเป็นบัสที่ส่งข้อมูลออกจากซีพียูได้ทิศทางเดียวเราเรียกบัสนี้ว่า ยูนิไดเร็กชั่นแนลบัส (Unidirectional Bus) ซีพียู จะใช้ในการอ้างตำแหน่งของหน่วยความจำ และอ้างตำแหน่ง ของพอร์ต

2.3.1.2 บัสข้อมูล : เป็นเส้นทางที่ใช้ส่งข้อมูลเข้าออกให้กับ ซีพียู โดยจะทำงานร่วมกับแอดเดรสบัส โดยบัสประเภทนี้ข้อมูลจะวิ่งได้สองทางซึ่งเรียกบัสประเภทนี้ว่าไดเร็คชั่นแนลบัส (Directional Bus)

2.3.1.3 บัสควบคุม : เป็นกลุ่มของสัญญาณที่ใช้ควบคุมอุปกรณ์ต่างๆ ที่ต่ออยู่กับ ซีพียู สัญญาณประเภทนี้ได้แก่สัญญาณอ่านเขียนข้อมูล สัญญาณอินเทอร์รัพต์ต่างๆ นอกจากนี้อุปกรณ์ต่างๆ ก็สามารถควบคุม ซีพียู ได้โดยใช้สัญญาณในระบบบัสควบคุมเช่นกัน

การทำงานของระบบบัสนั้นบัสทั้งสามประเภทจะทำงานสัมพันธ์กัน เช่น เมื่อซีพียูต้องการอ่านข้อมูลจากหน่วยความจำ จะส่งสัญญาณแอดเดรสบัสออกไปเพื่อเลือกตำแหน่งของหน่วยความจำที่ต้องการอ่าน จากนั้น ซีพียู จะส่งสัญญาณอ่านข้อมูลซึ่งอยู่ในส่วนของบัสควบคุม ต่อมาหน่วยความจำจะส่งข้อมูลออกมาทางบัสเข้าไปให้ ซีพียู การจัดอุปกรณ์อินพุต เอาต์พุต และหน่วยความจำให้ต่อกับ ซีพียู แสดงในรูปที่ 2.2



รูปที่ 2.2 การต่อส่วนต่างๆ เข้ากับหน่วยประมวลผลกลาง

2.3.2 หน่วยคำนวณและลอจิกหรือเอแอลยู (Arithmatic LogicUnit : ALU)

ทำหน้าที่ประมวลผลข้อมูล คำนวณและปฏิบัติการทางคณิตศาสตร์ ผลลัพธ์จะเก็บไว้ในแอคคิวมูเลเตอร์รีจิสเตอร์ (Accumulator Register) หรือนำไปเก็บไว้ในรีจิสเตอร์เป็นการชั่วคราว โดยทั่วไป จะรับและส่งข้อมูลสู่แอคคิวมูเลเตอร์ (Accumulator) ความสามารถของเอแอลยูจะมีความแตกต่างกันตามรุ่นของไมโครโปร เซสเซอร์

2.3.3 รีจิสเตอร์ (Register)

รีจิสเตอร์เป็นส่วนประกอบส่วนหนึ่งของไมโครโปรเซสเซอร์ ทำหน้าที่เก็บข้อมูลชั่วคราวภาย ในซีพียู สำหรับเตรียมการประมวลผลต่างๆ โดยรีจิสเตอร์ภายในซีพียูนั้นจะมีหลายตัว สามารถแบ่งเป็นกลุ่มๆ ได้ดังนี้

2.3.3.1 รีจิสเตอร์ใช้งานทั่วไป (General Purpose Unit) : รีจิสเตอร์ใช้งานทั่วไปหรือกลุ่มรีจิเตอร์ข้อมูล ซึ่งรีจิสเตอร์เหล่านี้มีหน้าที่เก็บรักษาข้อมูลได้

2.3.3.2 รีจิสเตอร์ตัวชี้และอินเด็กซ์ (Pointer & Index Register) : รีจิสเตอร์กลุ่มนี้ใช้ในการชี้ตำแหน่งหน่วยความจำที่ต้องการติดต่อ

2.3.3.3 รีจิสเตอร์เซกเมนต์ (Segmentment Register) : กลุ่มรีจิสเตอร์เซกเมนต์ เป็นรีจิสเตอร์ในส่วนของ บีไอยู ที่ใช้ในการอ้างตำแหน่งหน่วยความจำเมื่อต้องการอ่านเขียนข้อมูล โดยจะแบ่งหน่วยความจำออกเป็นเซกเมนต์

2.3.3.4 รีจิสเตอร์คำสั่ง (Instruction Pointer) : จะทำงานร่วมกับรีจิสเตอร์เซกเมนต์ โดยจะใช้เก็บตำแหน่งคำสั่งถัดไปที่จะให้ไมโครโปรเซสเซอร์ทำงาน

2.3.3.5 รีจิสเตอร์แฟลก (Flags Register) : จะใช้เก็บสถานะต่างๆ จากการประมวลผลของไมโครโปรเซสเซอร์แล้ว

2.3.4 หน่วยควบคุมทางลอจิก (Logic Control Unit)

ตัวแปลรหัสคำสั่ง ทำหน้าที่เก็บในรีจิสเตอร์ คำสั่งต่างๆ จากนั้นจะส่งสัญญาณไปควบคุมส่วนต่างๆ ในไมโครโปรเซสเซอร์ให้ทำตามคำสั่งนั้นๆ

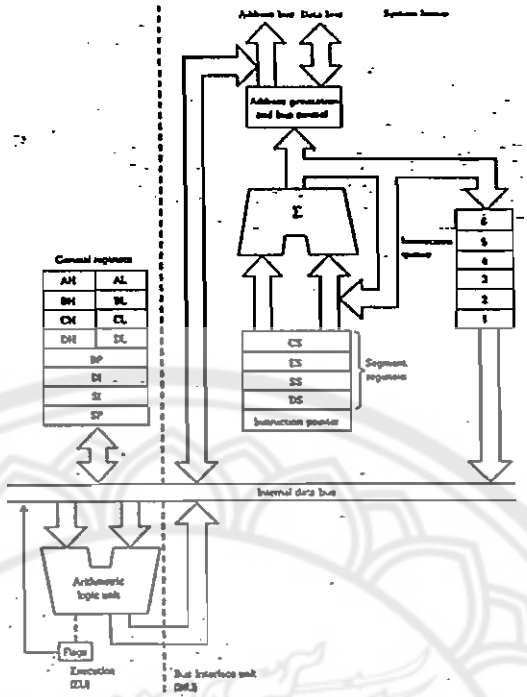
2.4 การทำงานทั้งหมดของไมโครคอมพิวเตอร์

การทำงานทั้งหมดของไมโครคอมพิวเตอร์ จะถูกควบคุมด้วยไมโครโปรเซสเซอร์ โดยทำหน้าที่เป็นหน่วยประมวลผลกลาง การทำงานของไมโครโปรเซสเซอร์จะทำหน้าที่ตามคำสั่งที่ป้อนให้ การให้คอมพิวเตอร์ทำงานคือการให้ไมโครโปรเซสเซอร์ทำงาน โดยจะเริ่มจากการอ่านข้อมูลจากหน่วยความจำโปรแกรมเข้ามาในตัวไมโครโปรเซสเซอร์ จากนั้นจะถอดรหัสคำสั่งและทำงานตามคำสั่ง เมื่อทำงานตามคำสั่งแล้วจะไปอ่านรหัสคำสั่งต่อไปออกมาเราเรียกกระบวนการนี้ว่าเฟทแอนด์เอ็กซีคิวไซเคิล (fetch and execute cycle) และการทำงานคำสั่งหลายๆ คำสั่งต่อกันเราเรียกว่าการทำโปรแกรม เมื่อไมโครโปรเซสเซอร์เริ่มทำงานกระบวนการต่างๆ จะเป็นดังต่อไปนี้

2.4.1 ค่าในรีจิสเตอร์ไอพี (Instruction Pointer,IP) จะถูกคำนวณเป็นค่าตำแหน่งหน่วยความจำ โปรแกรมและจะส่งออกมาทางเอาต์พุตของ บีไอยู (Bus Interface Unit) เพื่ออ้างไปที่ตำแหน่งหน่วยความจำโปรแกรมที่ไมโครโปรเซสเซอร์ต้องการติดต่อ จากนั้นจะอ่านรหัสโปรแกรม จากหน่วยความจำเข้ามายังบีไอยู เพื่อให้หน่วย อียู (Execute Unit) ทำการประมวลผล

2.4.2 ค่ารีจิสเตอร์ไอพี (instruction pointer) จะเพิ่มค่าขึ้นอีกเพื่ออ่านข้อมูลรหัสคำสั่งต่อไป

2.4.3 ไมโครโปรเซสเซอร์ จะทำงานตามรหัสคำสั่งที่อ่านเข้ามา และอ่านค่าคำสั่งต่อไป



รูปที่ 2.3 โครงสร้างภายในไมโครโปรเซสเซอร์

2.5 ไมโครคอนโทรลเลอร์ตระกูล MCS-8051

ไมโครคอนโทรลเลอร์ตระกูล MCS 8051 ประกอบด้วยไมโครคอนโทรลเลอร์หลายรุ่น ซึ่งมีสถาปัตกรรมพื้นฐานที่เหมือนกัน เพียงแต่มีขนาดหรือจำนวนของหน่วยทำงานภายในที่แตกต่างกันออกไป เพื่อความเหมาะสมในงานประยุกต์ต่าง ๆ ตามความต้องการ ดังแสดงในให้เห็นในตารางของรูปที่ 2.4

EMBEDDED CONTROLLERS										
Feature	8051AH	8031AH	8751H	80C51BH	80C31BH	87C51	8052AH	8032AH	8752	8044H
Program Memory (Bytes)	4K	-	4K	4K	-	4K	8K	-	8K	4K
RAM Memory (Bytes)	128	128	128	128	128	128	256	256	256	192
Program Memory Expansion (Off Chip) (Bytes)	64K	64K	64K	64K	64K	64K	64K	64K	64K	64K
Data Memory Expansion (Off Chip) (Bytes)	64K	64K	64K	64K	64K	64K	64K	64K	64K	64K
Max Clock Frequency (MHz)	12	12	12	16	16	16	16	12	12	12
Typical instruction Time (uS)	1	1	1	0.75	0.75	0.75	1	1	1	1
16-Bit Timer / Counters	2	2	2	2	2	2	3	3	3	2
Serial Communications	Synchronous Mode, Asynchronous Mode, 9 or 10 - Bit Programmable									HDLC/SDLC
No. of I/O Lines	32	16	32	32	16	32	32	16	32	32
Interrupt Sources (Two Priority Levels)	5	5	5	5	5	5	6	6	6	5
Power Requirements (ICC Max. mA)	125	250	24	24	29	175	175	175	200	
Programmable Power Modes	-	-	-	4.0 mA	4.0 mA	4.0 mA	-	-	-	-
Idle Power Down	-			50 uA	51 uA	52 uA				30 mA

รูปที่ 2.4 ไมโครคอนโทรลเลอร์ ตระกูล MCS 8051

2.5.1 คุณลักษณะพื้นฐานของ MCS-8051

จากแผนภาพในรูปที่ 2.3 แสดงให้เห็นถึงหน่วยการทำงานพื้นฐานของ ไมโครคอนโทรลเลอร์เบอร์ต่าง ๆ ที่จัดอยู่ภายในตระกูล MCS-51 นี้ประกอบด้วย

2.5.1.1 หน่วยประมวลผลกลางขนาด 8 บิต

2.5.1.2 หน่วยประมวลผลสำหรับข้อมูลแบบบิต (Boolean Processor)

2.5.1.3 ความสามารถในการอ้างตำแหน่งของหน่วยความจำ โปรแกรม 64 กิโลไบต์

2.5.1.4 ความสามารถในการอ้างตำแหน่งของหน่วยความจำข้อมูล 64 กิโลไบต์

2.5.1.5 หน่วยความจำโปรแกรมภายในในขนาด 4 กิโลไบต์แบบอีพีรอม (เบอร์ 8751) หรือแบบรอม (เบอร์ 8051)

2.5.1.6 หน่วยความจำแบบแรม ภายในจำนวน 128 ไบต์

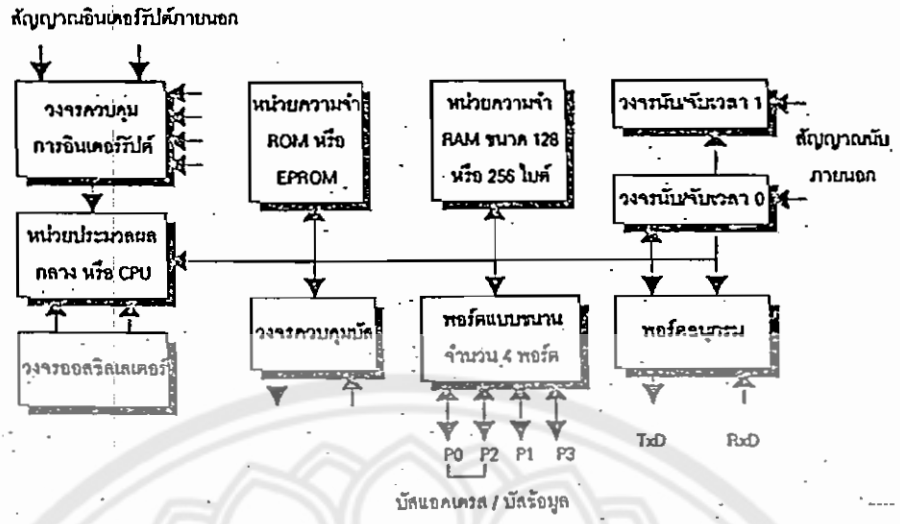
2.5.17 พอร์ตอินพุท/เอาต์พุทแบบจนสนจำนวน 32 เส้น ซึ่งสามารถแยกทำงานได้อย่างอิสระ

2.5.1.8 วงจรนับ/จับเวลาขนาด 16 บิต จำนวนสองวงจร

2.5.1.9 วงจรสื่อสารแบบอนุกรมแบบฟูลดูเพล็กซ์ (Full Duplex)

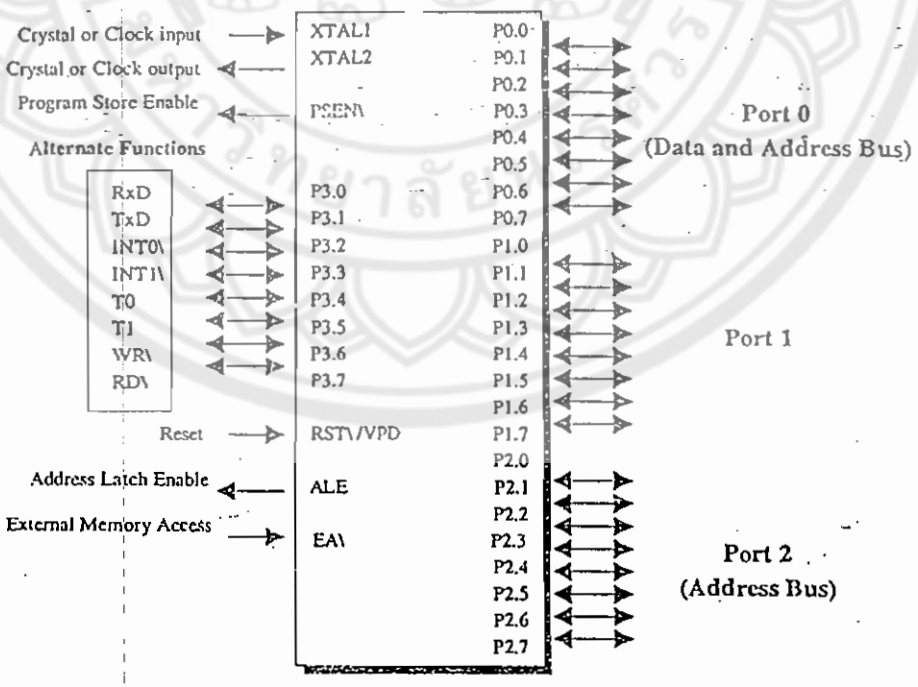
2.5.1.10 วงจรควบคุมการอินเตอร์รับต์จากแหล่งกำเนิดสัญญาณ 6 ประเภท พร้อมการกำหนด ลำดับความสำคัญได้สองระดับ

2.5.1.11 วงจรออสซิลเลเตอร์ภายใน



รูปที่ 2.5 แผนภาพบล็อกแสดงหน่วยทำงานพื้นฐานของ MCS-51

โดยมากแล้วไมโครคอนโทรลเลอร์ตระกูลนี้มักจะมีรูปร่างของไอซี เป็นแบบดีไอพี (DIP) ขนาด 40 ขาดังแผนภาพในรูปที่ 2.6 ซึ่งแต่ละขาสัญญาณจะมีหน้าที่ระบุชัดเจนตามสัญลักษณ์ชื่อย่อที่กำกับในแต่ละขา อย่างไรก็ตามจะมีบางขาสัญญาณที่อาจจะมีหน้าที่ได้มากกว่าหนึ่งอย่าง ซึ่งการกำหนดว่าจะทำงานในลักษณะใดขึ้นอยู่กับการเชื่อมต่อวงจรเท่ากับขาสัญญาณ และโปรแกรมการควบคุมของระบบนั้น

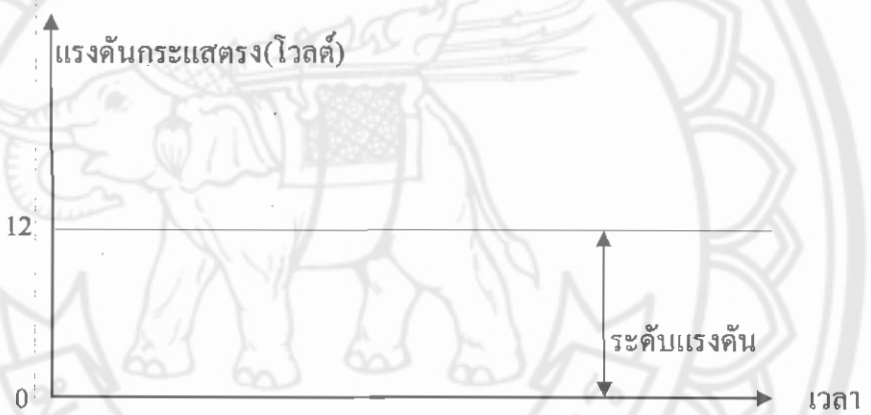


รูปที่ 2.6 การกำหนดหน้าที่ขาสัญญาณของไอซีMCS-8051

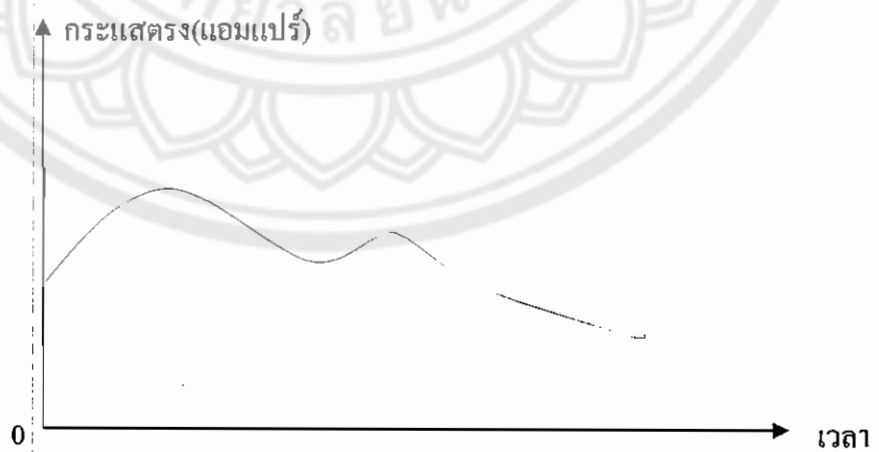
2.6 ข้อมูลอะนาลอกกับข้อมูลดิจิตอล

สัญญาณอิเล็กทรอนิกส์พื้นฐานมีอยู่ด้วยกัน 2 ชนิด คือ สัญญาณอะนาลอก และสัญญาณดิจิตอล สัญญาณเหล่านี้อาจเป็นกระแสไฟฟ้าที่ใช้ทำต่างๆ หรือเป็นแรงดันไฟฟ้าที่ใช้ทำหน้าที่เช่นเดียวกันนี้ก็ได้ การแยกประเภทของสัญญาณที่กำหนดให้จะขึ้นอยู่กับลักษณะอาการที่สัญญาณนั้นแปรหรือเปลี่ยนแปลงไป

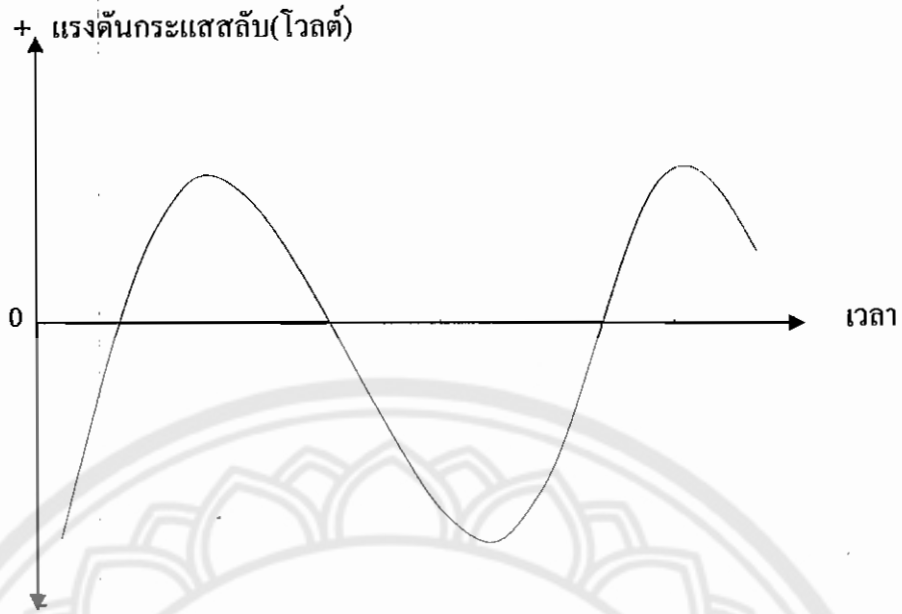
สัญญาณอะนาลอกเป็นสัญญาณที่แปรตามเวลาอย่างต่อเนื่อง ตัวอย่างของสัญญาณ อะนาลอกได้แก่แรงดันและกระแส สัญญาณอะนาลอกนี้อาจเป็นแบบกระแสตรง หรือกระแสสลับก็ได้ รูปที่ 2.7 แสดงถึงระดับแรงดันกระแสตรงที่คงตัว รูปที่ 2.8 แสดง แสดงถึงสัญญาณกระแสตรงที่แปรตามเวลาอย่างต่อเนื่อง ขนาดของกระแสในรูปนี้มีการเปลี่ยนแปลง แต่ทิศทางของกระแสยังคงเป็นอย่างเดิม สำหรับรูปที่ 2.9 นั้นแสดงถึงแรงดันกระแสสลับรูปคลื่นไซน์ (sine wave) ซึ่งมีการเปลี่ยนแปลงอย่างต่อเนื่อง



รูปที่ 2.7 แรงดันกระแสคงตัว

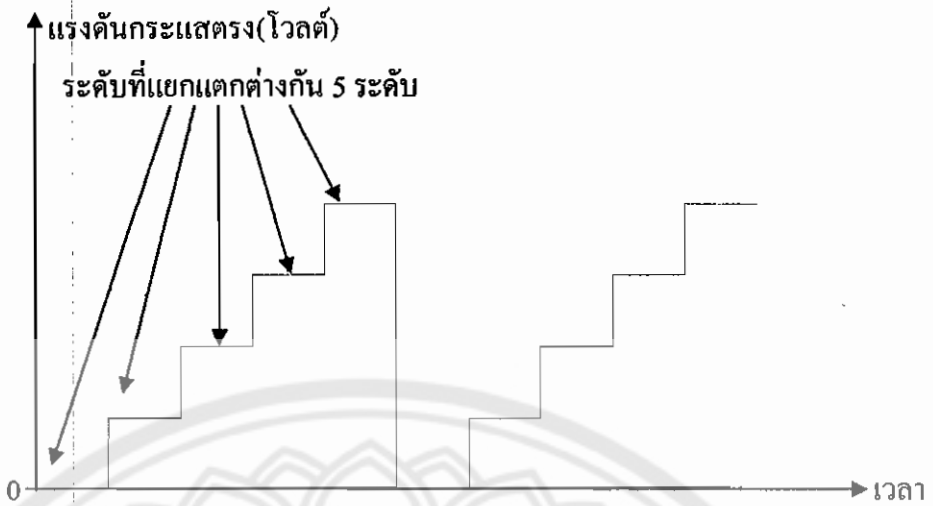


รูปที่ 2.8 กระแสตรงที่แปรตามเวลา

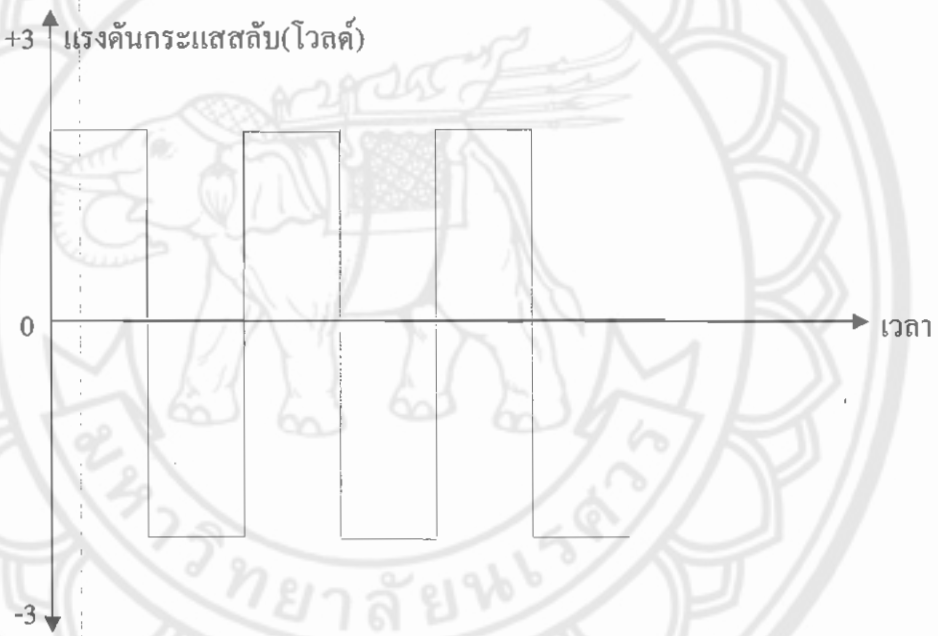


รูปที่ 2.9 ค่าของแรงดันเปลี่ยนแปลงตามเวลา ทิศทางของแรงดันเปลี่ยนกลับไปกลับมา

สัญญาณอิเล็กทรอนิกส์พื้นฐานอีกแบบหนึ่งก็คือสัญญาณดิจิตอล สัญญาณดิจิตอลเป็นสัญญาณที่แปรตามเวลาในลักษณะของการกระโดดเป็นขั้น ๆ แทนที่จะแปรตามเวลาอย่างต่อเนื่อง ดังเช่นในกรณีของสัญญาณอะนาลอก สัญญาณดิจิตอลระดับหรือสถานะที่แยกห่างจากกันตั้งแต่ 2 ระดับหรือ 2 สถานะขึ้นไป รูปด้านล่างนี้แสดงถึงสัญญาณดิจิตอลแบบต่าง ๆ สัญญาณดิจิตอลคล้ายกับสัญญาณอนาลอกในแง่ที่ว่าอาจจะอยู่ในรูปของกระแสสลับหรือกระแสตรงก็ได้ ในรูปที่ 2.10สัญญาณที่แสดงไว้เป็นแรงดันกระแสตรงที่แปรตามเวลาเป็นขั้นต่าง ๆ กัน รูปนี้แสดงระดับของแรงดันเอาไว้ 5 ระดับด้วยกัน สัญญาณนี้ไม่ได้แปรตามเวลาอย่างต่อเนื่อง แต่กระโดดจากระดับหนึ่งไปเป็นอีกระดับหนึ่งที่อยู่ถัดไปอย่างทันทีทันใด



รูปที่ 2.10 แรงดันกระแสตรงที่แปรตามเวลาเป็น 5 ขั้น



รูปที่ 2.11 สัญญาณกระแสสลับ 2 ระดับที่เปลี่ยนแปลงอยู่ระหว่าง -3 กับ +3 โวลต์

โดยทั่วไปสัญญาณดิจิตอลอาจจะมีระดับได้หลายระดับหรือมีขั้นได้หลายขั้น แต่ส่วนมากแล้วสัญญาณดิจิตอลจะมีระดับเพียง 2 ระดับเท่านั้น สัญญาณดิจิตอลหลายระดับใน รูปที่ 2.10 นั้นอาจถือว่าเป็นสัญญาณอะนาลอกที่ไม่ต่อเนื่องก็ได้ แต่ถ้าสัญญาณได้เปลี่ยนแปลงกลับไปกลับมาอยู่ระหว่างค่ากระแสคงตัวหรือค่าแรงดันคงตัวเพียงสองค่าเท่านั้นแล้ว สัญญาณนั้นจะเป็นสัญญาณดิจิตอลเสมอ รูปที่ 2.11 แสดงถึงสัญญาณดิจิตอลกระแสสลับ จากรูปสัญญาณเปลี่ยนแปลงกลับไปกลับมาอยู่ระหว่าง +3 โวลต์ กับ -3 โวลต์เท่านั้น สัญญาณลักษณะนี้เป็นสัญญาณดิจิตอลกระแสสลับ

สิ่งสำคัญที่ควรชี้ให้เห็นในที่นี้ก็คือ ถึงแม้ปริมาณหลายอย่างจะมีคุณสมบัติแบบอะนาลอกอยู่ในตัวก็ตาม แต่ปริมาณอะนาลอกแทบทุกอย่างจะสามารถแสดงได้ในรูปของปริมาณดิจิตอล มีบ่อยครั้งที่มีการแปลงปริมาณอะนาลอกไปเป็นปริมาณดิจิตอล เพื่อความสะดวกในการแสดงผล

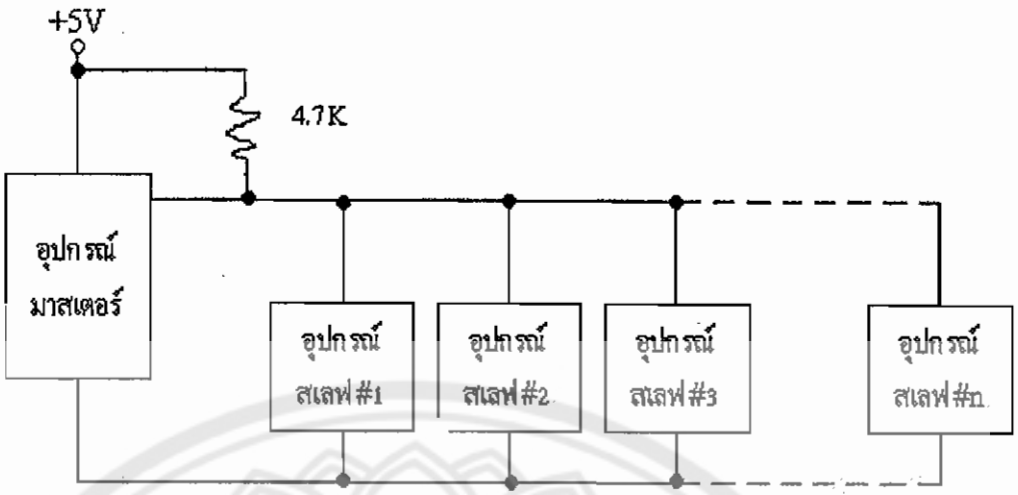
2.7 ระบบการสื่อสารข้อมูลอนุกรมแบบหนึ่งสาย (1-Wire™ Serial Bus)

ระบบการสื่อสารข้อมูลแบบนี้ผู้คิดค้นคือ ดัลลัสเซมิคอนดักเตอร์ ดังนั้นในบางครั้งจึงเรียกระบบการสื่อสารข้อมูลแบบนี้ว่า ระบบสื่อสารข้อมูลดัลลัสหนึ่งสาย (Thee Dallas 1-Wire™ Bus) ระบบสื่อสารข้อมูลแบบนี้เป็นระบบที่มีความชาญฉลาดและใช้จำนวนสายสัญญาณเพียง 1 เส้นเท่านั้นโดยไม่ต้องมีสายสัญญาณนาฬิกามาควบคุมจังหวะการถ่ายทอดข้อมูล เหมือนกับระบบสื่อสาร ข้อมูลอนุกรมในแบบอื่นๆ เนื่องจากสายข้อมูลนั้นจะทำหน้าที่เสมือนหนึ่งเป็นสายสัญญาณนาฬิกาในตัว ส่วนค่าของข้อมูลจะพิจารณาจากลักษณะของรูปสัญญาณที่ปรากฏบนสายสัญญาณในแต่ละช่วงของเวลา รูปแบบการถ่ายทอดข้อมูลจะเป็นแบบอะซิงโครนัสในระดับบิด ไม่มีการกำหนดความยาวของข้อมูลเป็นระดับไบต์ ระบบสื่อสารแบบนี้เหมาะที่จะใช้ในการสื่อสารข้อมูลระหว่างไอซีบนแผงวงจรเดียวกัน หรือสร้างเป็นโครงข่ายสื่อสารแบบทวิสต์แพร์ก็ได้

2.7.1 คุณสมบัติทางเทคนิคของระบบบัสหนึ่งสาย

สายสัญญาณบนระบบบัสแบบหนึ่งสายนี้จะเป็นสายสัญญาณแบบสองทิศทาง แต่ข้อมูลจะสามารถเดินทางได้ในทิศทางเดียวภายในช่วงเวลาหนึ่งๆ นั่นคือ มีลักษณะคล้ายกับระบบสื่อสารแบบฮาล์ฟดูเพล็กซ์ (half-duplex) ตัวอย่างที่เห็นได้ชัดคือ การใช้งานวิทยุสื่อสารหรือวิทยุสมัครเล่น อุปกรณ์บนระบบบัสต้องมีการระบุอย่างชัดเจนว่า ตัวใดเป็นอุปกรณ์มาสเตอร์ ตัวใดเป็นอุปกรณ์สเลฟ โดยส่วนใหญ่อุปกรณ์มาสเตอร์ คือ ไมโครคอนโทรลเลอร์ ส่วนอุปกรณ์สเลฟได้แก่ ไอซีตรวจจับอุณหภูมิ ไอซีหน่วยความจำแรม เป็นต้น อุปกรณ์มาสเตอร์จะเป็นตัวจัดเตรียมความพร้อมของสายสัญญาณและควบคุมการถ่ายทอดข้อมูลบนสายสัญญาณนั้น ข้อมูลทั้งหมดไม่ว่าจะเป็นข้อมูลควบคุมหรือข้อมูลใช้งานจะถูกส่งลงบนสายสัญญาณที่มีเพียงเส้นเดียวนี้ทั้งหมด ในระหว่างการทำงานอุปกรณ์มาสเตอร์และสเลฟสามารถเป็นได้ทั้งตัวส่งและ ตัวรับขึ้นอยู่กับเงื่อนไขของการทำงานในขณะนั้น ยกตัวอย่างเช่น ถ้าหากมีการเขียนข้อมูลจากอุปกรณ์มาสเตอร์ไปยังอุปกรณ์สเลฟ ตัวส่งคืออุปกรณ์มาสเตอร์ ตัวรับคืออุปกรณ์สเลฟ ในทางตรงกันข้ามหากเป็นการอ่านข้อมูลจากอุปกรณ์สเลฟ ตัวส่งจะกลายเป็นอุปกรณ์สเลฟ และตัวรับคืออุปกรณ์มาสเตอร์ ในระบบบัสหนึ่งระบบ ต้องมีอุปกรณ์มาสเตอร์เพียงตัวเดียวเท่านั้น

สายสัญญาณของระบบบัสนี้ต้องกำหนดสภาวะปกติไว้ที่ลอจิกสูง สามารถทำได้โดยการต่อตัวต้านทานค่าประมาณ 7.4 กิโลโอห์มพูลอัปกับไฟเลี้ยง +5 โวลต์ ดังนั้นอุปกรณ์ที่นำมาต่อบนระบบบัสนี้จึงต้องออกแบบให้ภาคเอาต์พุตที่ต้องต่อกับสายสัญญาณมีลักษณะเป็นคอลเล็กเตอร์เปิด



รูปที่ 2.12 การเชื่อมต่อบนระบบบัสหนึ่งสาย

2.7.2 รูปแบบของการสื่อสารข้อมูลแบบหนึ่งสาย

ในการติดต่อสื่อสารข้อมูลในระบบบัสหนึ่งสาย อุปกรณ์มาสเตอร์จะสามารถติดต่อกับอุปกรณ์สเลฟได้ครั้งละหนึ่งตัวเท่านั้น ดังนั้นอุปกรณ์สเลฟแต่ละตัวต้องมีข้อมูลกำหนดแอดเดรสเฉพาะตัว โดยจะเก็บไว้ในหน่วยความจำรอมภายในอุปกรณ์สเลฟนั้นๆ โดยปกติอุปกรณ์สเลฟในระบบบัสหนึ่งสายของดัลลัสนี้จะมีหน่วยความจำขนาด 64 บิตหรือ 8 ไบต์ สำหรับเก็บข้อมูลต่างๆ ที่สำคัญของอุปกรณ์แต่ละตัว ซึ่งประกอบด้วย

2.7.2.1 รหัสของตระกูลจำนวน 8 บิต

2.7.2.2 เลขหมายประจำตัว (Serial number) จำนวน 48 บิต

2.7.2.3 รหัสตรวจสอบความผิดพลาด (CRC : Cyclical Redundancy Check) จำนวน 8 บิต

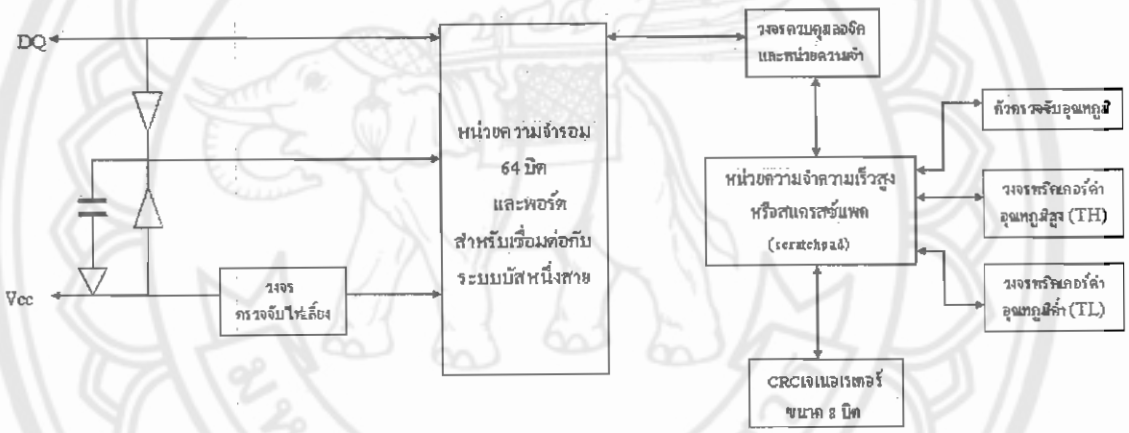
ระบบบัสหนึ่งสายนี้สามารถอ่านข้อมูลประจำตัวของอุปกรณ์สเลฟได้ ด้วยการใช้คำสั่งอ่านหน่วยความจำรอม (Read ROM) ในกรณีที่บนสายสัญญาณมีอุปกรณ์สเลฟเพียงตัวเดียว ไม่จำเป็นต้องอ้างแอดเดรสในการติดต่อ

รูปแบบการติดต่อบนระบบบัสหนึ่งสายจะเริ่มต้นขึ้นเมื่ออุปกรณ์มาสเตอร์ทำการรีเซตและกำหนดแอดเดรสของอุปกรณ์ที่ทำการติดต่อ ถ้าหากมีอุปกรณ์สเลฟเพียงตัวเดียวสามารถข้ามขั้นตอนการติดต่อกับหน่วยความจำรอมในอุปกรณ์สเลฟได้ จะเรียกวิธีการดังกล่าวว่า การไม่ติดต่อหน่วยความจำรอม หรือ สคิปรอม (Skip ROM) จากนั้นรอการตอบรับจากอุปกรณ์สเลฟ เมื่อมีการตอบรับสมบูรณ์ก็จะสามารถเริ่มต้นขั้นตอนการอ่านหรือเขียนข้อมูลได้ต่อไป

2.7.3 ไอซีตรวจจับอุณหภูมิดีเอส 1820 (DS1820)

เป็นไอซีตรวจจับอุณหภูมิที่ใช้ในการติดต่อแบบระบบบัสหนึ่งสาย มีขาต่อใช้งานเพียง 3 ขา คือดีคิว (DQ) ซึ่งเป็นขาเชื่อมต่อกับระบบบัส ขาต่อไฟเลี้ยงภายนอก และขากราวด์ ดังแสดงการจัดขาของไอซีดีเอส 1820 ในรูปที่ 2.13 และมีโครงสร้างการทำงานภายในแสดงในรูปที่ 2.14

รูปที่ 2.13 การจัดขาของไอซีดีเอส 1820



รูปที่ 2.14 โครงสร้างการทำงานภายใน

หัวใจสำคัญของไอซีดีเอส 1820 อยู่ที่ตัวตรวจจับอุณหภูมิและหน่วยความจำความเร็วสูง ที่เรียกว่า สแครตซ์แพต (Scratchpad) ซึ่งมีขนาด 9 ไบต์ มีการจัดสรรหน่วยความจำส่วนนี้แสดงในรูปที่ 2.15

	ไบต์
ข้อมูลอุณหภูมิไบต์ต่ำ (TL)	0
ข้อมูลอุณหภูมิไบต์สูง	1
ข้อมูลอุณหภูมิค่าสูง	2
ข้อมูลอุณหภูมิค่าต่ำ (TL)	3
สำรองไว้	4
สำรองไว้	5
รีจิสเตอร์เก็บค่าการนับ	6
รีจิสเตอร์เก็บค่าการนับต่อ ^{0}C	7
CRC	8

รูปที่ 2.15 การจัดสรรหน่วยความจำภายในไอซีดีเอส 1820

เมื่อวัดอุณหภูมิได้ก็จะนำค่าที่วัดได้นี้มาเก็บไว้ในสแครตซ์แพดที่ไบต์ 0 และ 1 ทั้งนี้เนื่องจากไอซีดีเอส 1820 สามารถให้ข้อมูลของอุณหภูมิได้ละเอียดถึง 16 บิต เมื่อนำมาแปลงเป็นเลขฐานสิบจึงสามารถแสดงความละเอียดได้ของค่าอุณหภูมิได้ถึง 0.5 องศาเซลเซียส หรือ 0.9 องศาฟาเรนไฮต์โดยมีย่านการวัดอุณหภูมิ -55 ถึง 125 องศาเซลเซียสหรือ -67 ถึง 257 องศาฟาเรนไฮต์ ซึ่งใช้เวลาในแปลงค่าอุณหภูมิเป็นข้อมูลดิจิตอลประมาณ 200 มิลลิวินาที สามารถกำหนดขอบเขตของอุณหภูมิที่ทำการวัดได้ และให้แจ้งเตือนเมื่อค่าของอุณหภูมิสูงขึ้นหรือลดลงถึงค่าที่กำหนด โดยค่าอุณหภูมิที่กำหนดนี้จะเก็บไว้ในสแครตซ์แพดในไบต์ 2 และ 3

2.7.4 คำสั่งเพื่อควบคุมการทำงานของไอซีดีเอส 1820

ในการติดต่อกับไอซีดีเอส 1820 จะมีคำสั่งที่ต้องส่งให้แก่ไอซีดีเอส 1820 เพื่อกำหนดรูปแบบการทำงาน คำสั่งที่ใช้มากที่สุดมีด้วยกัน 3 คำสั่งดังนี้

2.7.4.1 คำสั่งไม่ติดต่อกับหน่วยความจำรอมหรือสคริปรอม เนื่องจากในการใช้งานไอซีดีเอส 1820 โดยปกติแล้วจะมีไอซีดีเอส 1820 อยู่บนสายสัญญาณเพียงตัวเดียวจึงไม่จำเป็นต้องใช้ข้อมูลกำหนดแอดเดรส ดังนั้นจึงไม่ต้องติดต่อกับหน่วยความจำรอมเพื่ออ่านข้อมูล ข้อมูลของคำสั่งสคริปรอมที่ต้องส่งให้ไอซีดีเอส 1820 คือ 0CH

2.7.4.2 คำสั่งแปลงอุณหภูมิ (Convert T) มีค่าเท่ากับ 44H เมื่อส่งคำสั่งนี้ให้ไอซีดีเอส 1820 จะต้องทำการวนลูปอย่างน้อย 200 มิลลิวินาที เพื่อให้ไอซีดีเอส 1820 ได้ใช้เวลาในการแปลงค่าอุณหภูมิเป็นข้อมูลดิจิตอลมาเก็บไว้ในสแครตซ์แพด

2.7.4.3 คำสั่งอ่านข้อมูลจากสแครตซ์แพด (Read Scratchpad) มีค่าเท่ากับ 0BEH เมื่อส่งคำสั่งนี้ไอซีดีเอส 1820 จะทยอยส่งข้อมูลค่าอุณหภูมิออกมาทั้งหมด 9 ไบต์ ดังมีรายละเอียดในตารางที่ 2.1

ตารางที่ 2.1 ลำดับการทำงานของไอซีดีเอส 1820

ขั้นตอนที่	การทำงานของอุปกรณ์มาสเตอร์	ข้อมูลหรือสภาวะ	รายละเอียด
1	ตัวส่ง	รีเซต	สร้างสัญญาณรีเซต
2	ตัวรับ	ตอบรับ	รอการตอบรับจาก DS1820
3	ตัวส่ง	OCCH	คำสั่ง Skip Rom
4	ตัวส่ง	44H	คำสั่ง แปลงอุณหภูมิ(Convert T)
5	ตัวรับ	ข้อมูล 1 ไบต์	อ่านแฟลต Busy 8 ครั้ง
6	ตัวส่ง	รีเซต	สร้างสัญญาณรีเซต
7	ตัวรับ	ตอบรับ	รอการตอบรับจาก DS 1820
8	ตัวส่ง	OCCH	คำสั่ง Skip Rom
9	ตัวส่ง	OBEH	คำสั่งอ่านค่าจากสแครตช์แพด
10	ตัวรับ	ข้อมูล 9 ไบต์	อ่านค่าอุณหภูมิจากสแครตช์แพด
11	ตัวส่ง	รีเซต	สร้างสัญญาณรีเซต
12	ตัวรับ	ตอบรับ	รอการตอบรับจาก DS 1820
13			ทำการคำนวณค่าที่ได้จาก DS 1820 เป็นตัวเลขฐานสิบแล้วนำไปแสดงผลหรือใช้งานอื่นต่อไป

2.8 ระบบควบคุมโดยพีไอดี (PID)

พรอบพอร์ชั่นแนล อินดิกรัล ดีริเวทีฟ (Proportional-integral-derivative) หรือ ระบบควบคุมโดยพีไอดี โดยส่วนมากแล้วจะมีลักษณะฟังก์ชันของการควบคุมเหมือนกับการควบคุมเครื่องทำความร้อนให้คงที่ แต่จะมีการใส่อัลกอริทึม (algorithm) เพิ่มเติมเข้าไปเพื่อให้ค่าเอาต์พุตเป็นไปตามที่ต้องการ การใช้พีไอดี ในการควบคุมระบบใดๆ ก็ตามจะดูจากค่าของเอาต์พุตที่มีค่าผิดพลาด (error) และดูช่วงเวลาในการเกิดความผิดพลาด จากข้อผิดพลาดของระบบควบคุมดังกล่าวจะทำการแก้ได้จากการปรับค่าคงที่หรือเพิ่มค่าพีไอดี เพื่อทำการควบคุมเอาต์พุต และสมการพีไอดี (PID) จะเป็นดังนี้คือ

$$U(t) = K_P[e(t) + \frac{1}{Ti}\int e(t)dt + Td\frac{de(t)}{dt}] \qquad (2.1)$$

การปรับเปลี่ยนค่า K_P, K_I และ K_D มีผลต่อระบบดังตารางที่ 2.2 เมื่อระบบเป็นแบบเฟิร์สออเดอร์

ตารางที่ 2.2 ผลของการใส่ค่า K_P, K_I และ K_D

CL RESPONSE	RISE TIME	OVERSHOOT	SETTING TIME	S-S ERROR
K_P	Decrease	Increase	Small Change	Decrease
K_I	Decrease	Increase	Increase	Small change
K_D	Small Change	Decrease	Decrease	Small Change

การออกแบบระบบควบคุมจำเป็นที่จะต้องตรวจสอบผลตอบสนอง (Response) ของระบบต่อสัญญาณ ที่ป้อนเข้า เพื่อทดสอบคุณสมบัติของระบบและผลตอบสนอง โดยป้อนสัญญาณที่รู้รูปร่างแน่นอนเพื่อใช้เป็นสัญญาณอ้างอิงให้กับระบบ แล้วจึงพิจารณาจากผลตอบสนองของระบบในโดเมนเวลา (Time Domain) เพื่อหาผลตอบสนองของระบบชั่วขณะ (Transient Response) ความคลาดเคลื่อนในสภาวะคงที่ (Steady state error) และเสถียรภาพของระบบ (Stability)

ในการศึกษาผลตอบสนองของระบบควบคุมนิยมใช้สัญญาณ แบบหนึ่งหน่วย (Step function) แบบพัลส์ (Pulse function) แบบลาด (Ramp function) และแบบซายน์ (Sine function) สำหรับผลตอบสนองทางความถี่ ส่วนฟังก์ชันมาตรฐานอื่น ๆ ขึ้นอยู่กับความต้องการศึกษาพฤติกรรมของระบบ ต่อการเปลี่ยนแปลงของสัญญาณที่ป้อนเข้า เช่น

2.8.1 กรณีต้องการศึกษาพฤติกรรมของระบบ ต่อการเปลี่ยนแปลงของอินพุตอย่างทันทีทันใด ใช้สัญญาณที่มีลักษณะเป็นแบบหนึ่งหน่วย (Step Function)

2.8.2 กรณีที่อินพุตมีค่าเพิ่มขึ้นหรือลดลงในอัตราสม่ำเสมอใช้สัญญาณแบบลาด

2.8.3 กรณีที่อินพุตเปลี่ยนค่ากลับไปกลับมาใช้พัลส์ฟังก์ชัน (Pulse Function)

2.8.4 กรณีอินพุตเป็นช็อค(Shock function) ใช้อิมพัลส์(Impulse Function) หรือสัญญาณอื่น ๆ เช่น ซายน์ (Sine Function) หรืออาจจะนำฟังก์ชันต่าง ๆ มารวมกันเพื่อให้ได้เป็นฟังก์ชันที่เหมาะสมกับอินพุตที่ต้องการ โดยทั่วไปผลตอบสนองต่อเวลาของระบบควบคุม แบ่งการพิจารณาออกเป็น 2 ส่วน คือ ผลการตอบสนองชั่วขณะ (Transient Response) จะพิจารณาลักษณะการเปลี่ยนแปลงของระบบควบคุมทันทีหลังจากสัญญาณอินพุตเริ่มเปลี่ยนแปลงจนกระทั่งเข้าสู่จุดหยุดนิ่ง และผลการตอบสนองเชิงสถิตย์ หรือผลตอบสนองสภาวะคงที่ (Steady state Response) จะพิจารณาลักษณะการเปลี่ยนแปลงของระบบควบคุม หลังจากระบบควบคุมเข้าสู่สภาพสมดุลใหม่หรือการเปลี่ยนแปลงทั้งหมดหยุดลง

ถ้าให้ C(t) เป็นผลตอบสนองเวลา เราสามารถเขียนเป็นสมการได้ดังนี้

$$C(t) = C(t)_t + C_{ss}(t) \qquad (2.2)$$

กำหนดให้ $C(t)_t$ = ผลตอบสนองชั่วขณะ

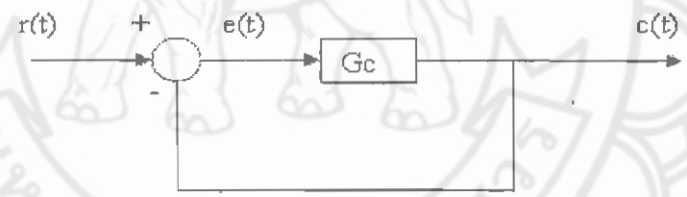
$C_{ss}(t)$ = ผลตอบสนองสภาวะคงที่

ผลตอบสนองชั่วขณะนั้นจะมีค่าเข้าสู่ศูนย์ เมื่อเวลาเข้าสู่อนันต์ ($t \to \infty$) ดังสมการ

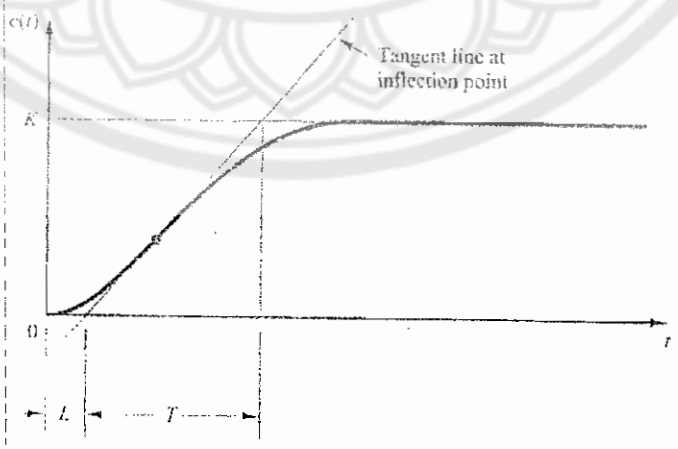
$$\lim_{t\to\infty} C(t) = 0$$

ระบบควบคุมส่วนใหญ่ จะมีความเฉื่อย และความฝืดอยู่ในระบบ ทำให้สัญญาณเอาต์พุต ไม่สามารถติดตามการเปลี่ยนแปลงของสัญญาณอินพุตได้อย่างทันทีทันใด ทำให้เกิดช่วงของผลตอบสนองชั่วขณะขึ้นในช่วงแรก ส่วนผลตอบสนองในเวลาถัดไปก็เป็นผลตอบสนองสภาวะคงที่

การชดเชยระบบควบคุมโดยใช้สมการพีไอดีจากหลักการของซีกเลอร์ นิโคล (Ziegler-Nichols) หากเป็นระบบเฟิร์สออเดอร์ (First Order) สามารถหาค่าต่าง ๆ ได้จากการคำนวณจากกราฟผลตอบสนองของระบบดังรูปที่ 2.16 เมื่อ r(t) คือสัญญาณอินพุต และ c(t) คือสัญญาณทางด้านเอาต์พุต



รูปที่ 2.16 บล็อกไดอะแกรมการป้อนกลับของระบบ

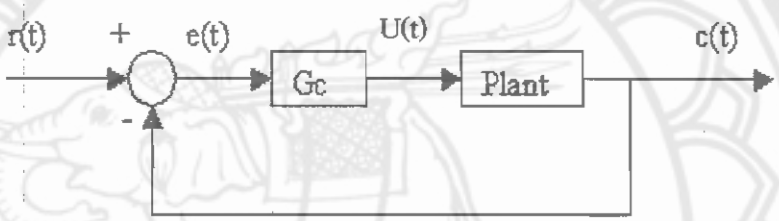


รูปที่ 2.17 กราฟการหาค่าพีไอดีของซีกเลอร์ นิโคล

และจากกราฟก็จะหาค่าของพีไอดีได้ดังความสัมพันธ์ในตารางที่ 2.3

ตารางที่ 2.3 ค่าของเคพี ทีไอและทีดี ที่ได้จากการดูรูปที่ 2.16

Type of Control	*Kp*	*Ti*	*Td*
P	$\frac{T}{L}$	∞	0
PI	$0.9\frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2\frac{T}{L}$	$2L$	$0.5L$



รูปที่ 2.18 บล็อกไดอะแกรมการป้อนกลับของระบบที่มีตัวควบคุม

ดังนั้นจากหลักการของซีกเลอร์ นิโคลจะได้สมการชดเชยระบบโดยการใส่ค่า อัตราขยาย (Gain) จาก $U(t) = K_P e(t)$ เมื่อดูจากตารางจะเห็นว่า $K_P = \frac{T}{L}$ ดังนั้นสมการของสัญญาณควบคุมในกรณีที่ตัวควบคุมเป็นแบบพีคือ

$$U(t) = \frac{T}{L} e(t)$$

$$U(t)_n \cong \frac{T}{L} e(t)_n \qquad (2.3)$$

สมการของสัญญาณควบคุมในกรณีที่ตัวควบคุมเป็นแบบพีไอคือ

$$U(t) = K_P e(t) + K_I \int e(t)dt$$

$$U(t) \cong 0.9\frac{T}{L} e(t) + \frac{0.27T \int e(t)dt}{L^2} \qquad (2.4)$$

สมการของสัญญาณควบคุมในกรณีที่ตัวควบคุมเป็นแบบพีไอดีคือ

$$U(t) = K_P e(t) + K_I \int e(t)dt + K_D K_P \frac{dc(t)}{dt}$$
$$\cong 1.2\frac{T}{L}e(t) + \frac{0.6T\int e(t)dt}{L^2} + \frac{0.72T^2}{L}\frac{de(t)}{dt} \qquad (2.5)$$

$U(t)$ = สัญญาณควบคุม

K_P = อัตราขยายเกนเคพี

K_I = อัตราขยายเกนเคไอ

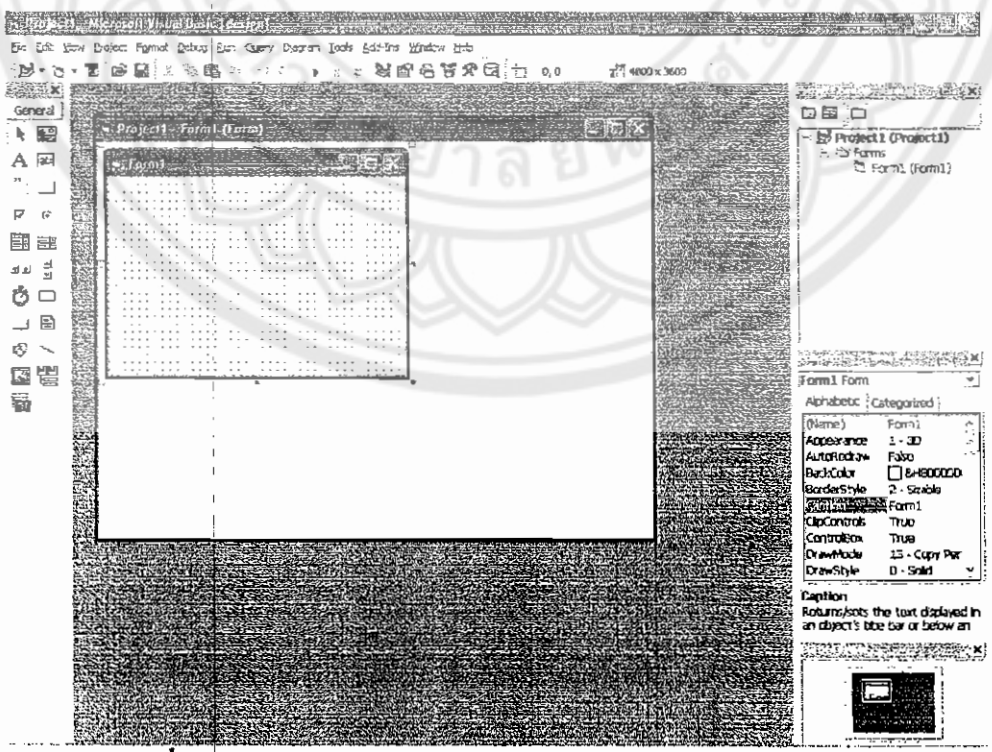
K_D = อัตราขยายเกนเคดี

$e(t)$ = ค่าความผิดพลาด

$\Delta t = t_n - t_{n-1}$

2.9 โปรแกรมวิชวลเบสิก 6.0 (Visual Basic)

โปรแกรมวิชวลเบสิกเวอร์ชั่น 6.0 เป็นภาษาระดับ อินดัสเตรียล สเตร็งค์ (Industrial Strength) คือสามารถพัฒนาซอฟต์แวร์ที่ใช้ได้จริงในธุรกิจต่าง ๆ และเป็นภาษาที่แพร่หลายที่สุดในโลก



รูปที่ 2.19 ส่วนประกอบต่างๆ ของหน้าจอหลักของโปรแกรมวิชวลเบสิก

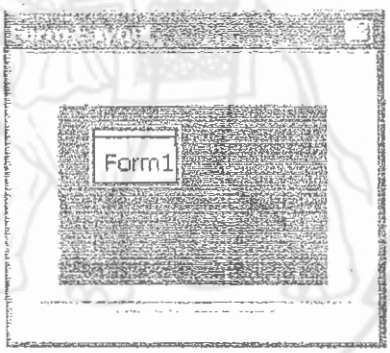
การทำงานของหน้าจอหลักมีลักษณะเหมือนกับหน้าจอวินโดว์โดยทั่วไป หน้าจอนี้เป็นสภาวะแวดล้อมในการพัฒนาวิชวลเบสิก เพราะจะแสดงเครื่องมือหลัก

เมนูบาร์ (Menu Bar) และ ทูลบาร์ (Tool Bar) จะมีคำสั่งในเมนูย่อยที่ช่วยให้ในการทำงานสะดวกรวดเร็วขึ้น

ฟอร์มวินโดว์ (Form Windows) พื้นที่ของวินโดว์ จะเป็นที่แสดงอุปกรณ์ส่วนต่าง ๆ โปรแกรม ที่ใช้ โดยที่ชิ้นส่วนต่าง ๆ จะทำงาน หรือถูกควบคุมโดยโปรแกรมที่เขียนขึ้น บนพื้นที่ของฟอร์มจะเห็นชิ้นส่วนต่าง ๆ เช่นปุ่ม (Button) ชื่อลาเบล (Label) เป็นต้น

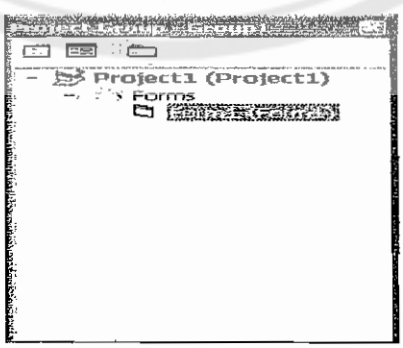
ทูลบอกซ์ (Tools Box) ในทูลบอกซ์ ประกอบไปด้วยคอนโทรลซึ่งเป็นภาพกราฟฟิกที่ใช้สร้างหน้าจอ

ฟอร์มเลเอ้าท์วินโดว์ (Form Layout Windows) ในหน้าหน้าจอเล็กๆจากรูปที่ 2.18 จะแสดงตำแหน่งของฟอร์ม เมื่อโปรแกรมได้รับการประมวลผล สามารถเลื่อนตำแหน่งของฟอร์ม ในฟอร์มเลเอ้าท์วินโดว์



รูปที่ 2.20 ฟอร์มเลเอาต์วินโดว์

โปรเจ็คเอ็กโพลเลอร์ (Project Explorer Windows) วินโดว์นี้จะแสดงโครงสร้างแบบต้นไม้ เพื่อให้เน้นถึงฟอร์มและโมดูลต่าง ๆ



รูปที่ 2.21 โปรเจ็คเอ็กโพลเลอร์

ห้องสมุดคณะวิศวกรรมศาสตร์

4800020

โคดวินโดส์ (Code Windows) สามารถเปิดโคดวินโดส์ เพื่อเปิดหน้าต่างดูรายละเอียดของเนื้อโปรแกรม

รูปที่ 2.22 โคดวินโดว์

2.9.1 การติดต่อกับอุปกรณ์ภายนอกทางพอร์ตอนุกรมด้วยโปรแกรมวิชวลเบสิก 6.0

โปรแกรมวิชวลเบสิกสามารถติดต่ออินพุต เอาท์พุต ผ่านทางพอร์ตอนุกรม (Serial Port or RS-232) ของเครื่องคอมพิวเตอร์ได้หรือที่รู้จักในชื่อ คอม 1 และคอม 2 แต่เนื่องจากระดับสัญญาณของพอร์ตอนุกรม RS-232 มีระดับแรงดันดั้ง ±3 โวลต์ ถึง ±12 โวลต์ แต่ในขณะที่ระดับสัญญาณของไมโครคอนโทรลเลอร์ MCS-8051 อยู่ในระดับทีทีแอล ดังนั้นจึงไม่สามารถเชื่อมต่อพอร์ตอนุกรมได้โดยตรง จึงต้องอาศัยการเชื่อมต่อผ่านไอซีพิเศษที่ทำหน้าที่ในการแปลงระดับสัญญาณจึงจะสามารถทำการติดต่อกับอุปกรณ์ภายนอก ผ่านพอร์ตอนุกรมของเครื่องคอมพิวเตอร์ได้ โปรแกรม วิชวลเบสิกสามารถติดต่อกับอุปกรณ์ต่างๆ โดยการติดต่อผ่านทางพอร์ตอนุกรมได้ ข้อดีของการติดต่อข้อมูลกันผ่านทางพอร์ตอนุกรม ก็คือสามารถใช้ได้ในระยะทางไกลๆระหว่างอุปกรณ์ ที่ติดต่อสื่อสารกันเนื่องจากที่โปรแกรมวิชวลเบสิก 6 จะมีตัวคอนโทรลที่ชื่อว่า MS Comm ที่ใช้ติดต่อกับพอร์ตอนุกรมไว้ให้อยู่แล้วจึงทำให้มีความสะดวกมากขึ้น ในการที่จะทำการเขียนโปรแกรมเพื่อที่จะติดต่ออุปกรณ์ทางพอร์ตอนุกรม

2.9.2 การเขียนโปรแกรมติดต่อกับพอร์ตอนุกรมด้วยวิชวลเบสิก สามารถทได้ 2 วิธีคือ

การติดต่อแบบอินเตอร์รัพต์ คือคือขบวนการอินเตอร์รัพท์ อุปกรณ์รอบข้างเกือบทุกชิ้น จะต้องปฏิบัติหรือทำงานอยู่ เพื่อที่ส่งสัญญาณไปให้แก่ซีพียูเสมอ ถ้าอุปกรณ์นั้นๆ พร้อมที่จะรับส่งข้อมูล โดยที่เมื่อข้อมูลเข้ามาก็จะทำให้มี CommEvent กับ OnComm Event การติดต่อแบบโพลลิ่ง ในระบบพีซีการติดต่อแบบโพลลิ่ง เป็นการส่งผ่านข้อมูลระหว่างเทอร์มินอลกับซีพียู กรณีข้อมูลเป็นประเภทไบต์ที่ส่งจากคีย์บอร์ด โดยวิธีการนี้จะตรวจสอบคีย์บอร์ดว่ามีข้อมูลส่งมา หรือไม่ โดยจะตรวจสอบตลอดเวลา การทำงานกับข้อมูลที่รับเข้ามา จะตรวจสอบด้วยความเร็วที่สูงกว่าอัตราความเร็วข้อมูลที่ส่งเข้ามาทางคีย์บอร์ด **การที่ซีพียูส่งสัญญาณออกไป** ตรวจสอบพบว่า มีข้อมูลที่ต้องส่งเข้ามาในโปรแกรมวิชวลเบสิกจะใช้การตรวจสอบข้อมูลที่มาจากพอร์ตอนุกรมตลอด โดยจะใช้คอนโทรลไทม์เมอร์ (Control Timer) เข้ามาช่วยในการเขียนโปรแกรมซึ่งสามารถตรวจสอบได้ถึงระดับ 1 มิลลิวินาที ในตัวคอนโทรล MSComm มี Event ที่ใช้เพียง Event เดียวเท่านั้นเอง ก็คือ OnComm Event ซึ่งจะใช้ในการติดต่อแบบอินเตอร์รัพต์ การเขียนโปรแกรมติดต่อผ่านทางพอร์ตอนุกรมแบบธรรมดาจะใช้ comEvent เพียง comEvReceive comEvSend

2.9.3 ไทม์เมอร์คอนโทรล (Timer Control)

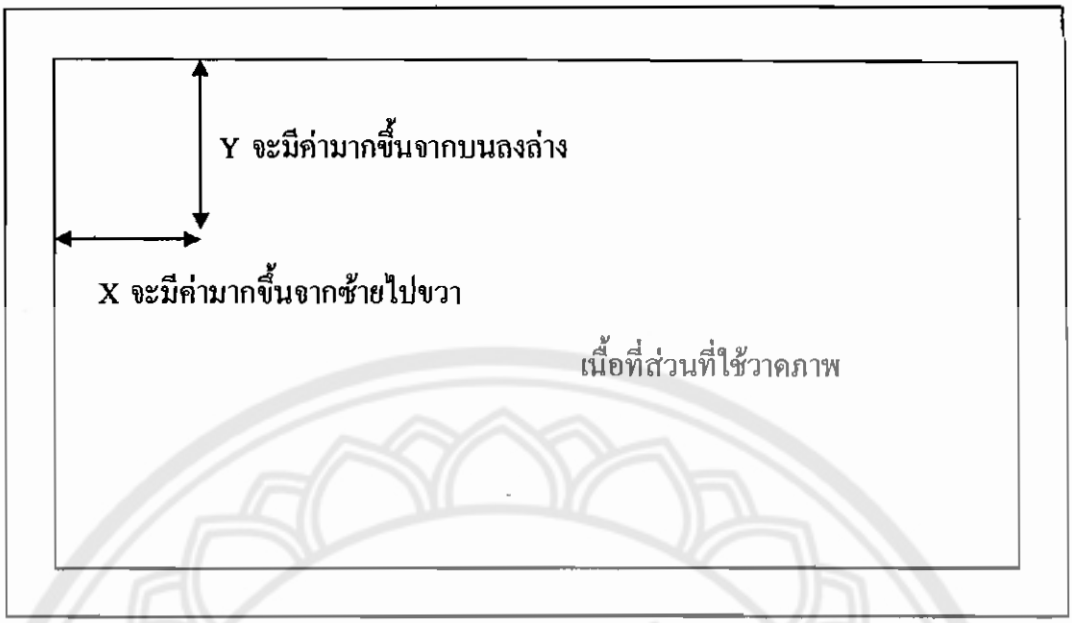
ไทม์เมอร์ (Timer) เป็นคอนโทรลที่ทำหน้าที่เหมือนตัวจับเวลาที่เราสามารถสั่งให้ทำงานเมื่อครบช่วงเวลาที่กำหนดคุณสมบัติสำคัญของไทม์เมอร์มีดังนี้

2.9.3.1 อีนาเบิล (Enable) จะเป็นพร็อพเพอร์ตี้ที่ใช้ควบคุมการทำงานของไทม์เมอร์ โดย ถ้าอีนาเบิล มีค่าเป็นจริง จะทำให้ไทม์เมอร์ทำงาน และจะหยุดไทม์เมอร์เมื่อมีค่าเป็นเท็จ

2.9.3.2 อินเทอร์วอล (Interval) จะใช้ระบุช่วงเวลาในการทำงานแต่ละครั้ง ค่าของอินเทอร์วอล อยู่ในช่วง 1 ถึง 65,535 มิลลิวินาที (65.535 วินาที) นั่นคือถ้ากำหนดเป็น 1000 จะมีการทำงานทุก 1 วินาทีและหากกำหนดให้อินเทอร์วอลเป็น 0 ก็จะเหมือนกับสั่งไม่ให้ไทเมอร์ทำงานนั่นเอง

2.9.4 กราฟิกบนวิชวลเบสิก 6.0

ระบบพิกัดของวิชวลเบสิกสามารถกำหนดขึ้นมาเองได้ ทั้งหน่วยการวัดและขอบเขตของ**พิกัดในออบเจ็คที่สามารถวาดภาพลงไปได้ดังรูปที่ 2.21**



รูปที่ 2.23 ระบบพิกัด

สเกลพิกัดปกติที่วิชวลเบสิกใช้นั้นจะเรียกว่าทวิป (Twip) ซึ่งขนาดของหน่วยทวิปนี้จะผูกติดอยู่กับการพิมพ์ผลลัพธ์ นั่นคือ 1 ทวิปจะมีขนาดเมื่อพิมพ์เป็น 1 ใน 1,440 นิ้ว ระบบพิกัดของจอภาพที่ใช้จะมีมุมบนซ้ายมือเป็นจุดเริ่มต้น (0,0) และมุมล่างขวาเป็นค่าที่มากที่สุดขึ้นกับขนาดความละเอียดของจอภาพ การอ้างตำแหน่งจะอ้างถึงค่าในแนวแกนนอน (X) และแกนตั้ง (Y) ตามลำดับ จะเห็นว่าการกำหนดสเกลแบบทวิปนี้ ให้ผลการพิมพ์ที่แน่นอนกว่าการกำหนดแบบอื่น อย่างไรก็ตามผู้ที่ใช้สามารถกำหนดสเกลที่จะใช้ได้ถึง 7 แบบด้วยกัน โดยกำหนดค่าให้กับพร็อพเพอร์ตี้สเกลโหมด

2.9.5 คำสั่งการกำหนดสีในวิชวลเบสิก

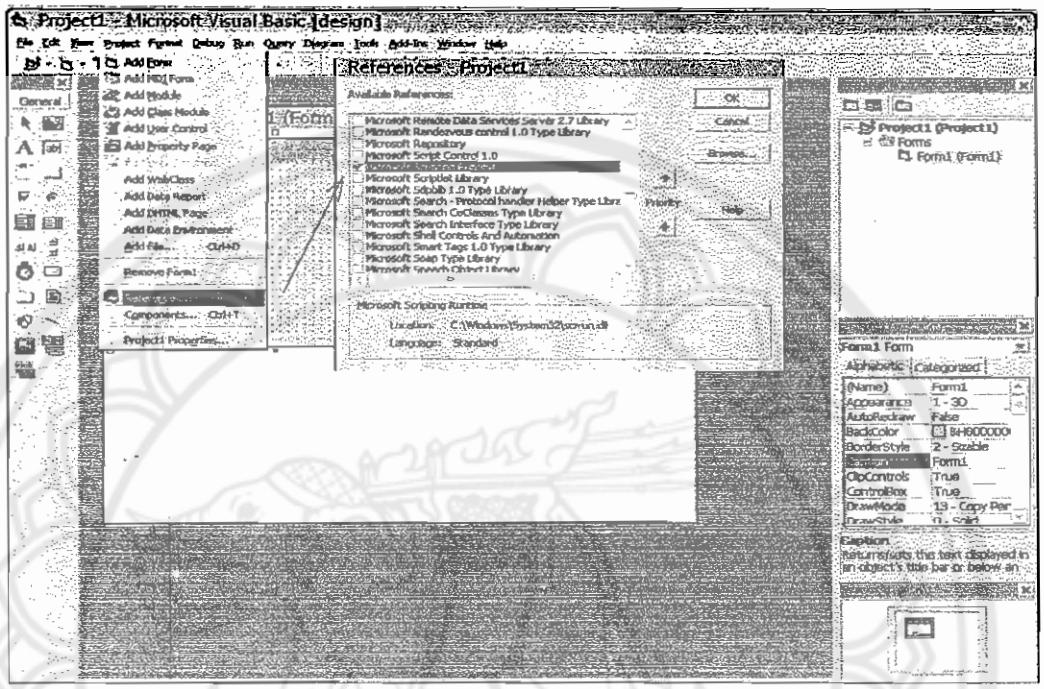
การกำหนดสีในออบเจ็ค มักจะใช้พร็อพเพอร์ตี้ต่างๆ ในการตั้งค่าสีส่วนต่างๆ ของออบเจ็คนั้นๆ เช่น สีพื้น (BackColor) หรือสีเส้น (ForeColor) เป็นต้น โดยสีต่างๆนี้เราสามารถเลือกตั้งค่าได้โดยตรง หรือจากฟังก์ชันอาร์จีบี (RGB( )) ,QBColor( ) หรือใช้ค่าคงที่ที่เป็นสีในระบบของเครื่อง (Systems Color)

2.9.5.1 ฟังก์ชันอาจีบี มีรูปแบบของคำสั่งเป็น RGB (red, green, blue) โดย red green และ blue เป็นค่าตัวเลขจำนวนเต็มตั้งแต่ 0 – 255 โดยตัวแปรทั้งสามจะเป็นค่าของแม่สีหลัก แดง เขียว น้ำเงิน ตามลำดับ ฟังก์ชันอาร์จีบีจะนำสีทั้ง 3 ซึ่งเป็นแม่สีที่มีความเข้ม-อ่อนตามค่าตัวเลขมาผสมกัน เพื่อให้ได้สีที่ต้องการ ดังนั้นจากคำสั่ง RGB เราจะสามารถเลือกสีได้ต่างกันถึง 16.7 ล้านสี (256*256*256) โดยสามารถใช้ค่ารหัสเป็นเลขฐานสิบหกได้ คือ ค่าตั้งแต่ 0 ถึง &HFFFFFF

2.9.5.2 ฟังก์ชันคิวบีคัลเลอร์ (QBColor) จะเป็นฟังก์ชันที่ใช้ในการตั้งสีพื้นฐาน 16 สี ของระบบที่ใช้การ์ดแสดงผลรุ่นเก่าที่มีสีได้จำกัดโดยค่ารหัสสีจะเป็นค่าตัวเลข 0 – 15

2.9.6 ระบบไฟล์ (File System object)

การใช้งานไฟล์ซิสเต็มอ็อบเจ็ค (FSO) นั้นจะต้องเพิ่มโปรเจ็ครีเฟอร์เร็น (Project Reference) ชื่อ ไมโครซอฟต์ สคริปติ้ง รันไทม์ (Microsoft Scripting Runtime)



รูปที่ 2.24 การเพิ่มโปรเจ็ครีเฟอร์เร็น

หลังจากการเพิ่มโปรเจ็ครีเฟอร์เรนแล้ววิชวลเบสิกก็จะมีข้อมูลคลาส (Class) ที่เป็นเสมือนแม่พิมพ์สำหรับสร้างไฟล์การสร้างออบเจ็คไฟล์ซิสเต็มออบเจ็ค จะต้องทำการเพิ่มไมโครซอฟต์สคริปติ้งรันไทม์ตามขั้นต้นก่อน ไม่เช่นนั้นจะไม่สามารถเรียกใช้ออบเจ็คของไฟล์ซิสเต็มออบเจ็คได้