

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมแบบกระจายส่วน

DISTRIBUTED CONTROL SYSTEMS



เลขามู.....  
เลขทะเบียน **72241**  
วัน,เดือน,ปี **12 ส.ย. 2550**

b. *117 65601*  
i. ....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

ปริญญาโทปีการศึกษา 2549

ภาควิชาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมแบบกระจายส่วน  
DISTRIBUTED CONTROL SYSTEMS

ผู้จัดทำ

นายสิทธิพล	กนกรัตน์มณี	46010831
นางสาวสุกัญญา	จันทะเลิศ	46010844
ว่าที่ร้อยตรีณัฐวุฒิ	เกิดกองทัพ	47015281
นายวิเชียร	วิมานประดิษฐ์	47015302

.....อาจารย์ที่ปรึกษา  
(ผู้ช่วยศาสตราจารย์ พรสุข รติโรจน์อนันต์)



## ระบบควบคุมแบบกระจายส่วน

โดย

นายสิทธิพล	กนกรัตน์มณี	46010831
นางสาวสุศัคนา	จันทะเลิศ	46010844
ว่าที่ร้อยตรีณัฐวุฒิ	เกิดกองทรัพย์	47015281
นายวิเชียร	วิมานประดิษฐ	47015302

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ พรสุข รติโรจน์อนันต์

ปีการศึกษา 2549

### บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการศึกษาระบบควบคุมกระจายส่วน (ดีซีเอส) เพื่อทำการควบคุมกระบวนการ 2 กระบวนการ คือกระบวนการระดับน้ำอันดับหนึ่งและกระบวนการระดับน้ำอันดับสอง โดยใช้ Microcontroller dsPIC 30F4011 เป็นตัวควบคุมกระบวนการและเชื่อมต่อกับกระบวนการทั้งสองกับคอมพิวเตอร์ส่วนบุคคล โดยใช้มาตรฐานอาร์เอส 232/485 ในการสื่อสาร เพื่อติดต่อและควบคุมกระบวนการ สำหรับการควบคุมกระบวนการน้ำอันดับหนึ่งใช้การควบคุมแบบ พีไอ และการควบคุมกระบวนการน้ำอันดับสองใช้การควบคุมแบบ พีไอดี โดยหาค่าพารามิเตอร์ในการปรับแต่งด้วยวิธี ซิกเลอร์และนิโคลส์กับวิธีดาค์ลิน และนำค่าผลตอบสนองของทั้ง 2 วิธีมาเปรียบเทียบกัน ซึ่งผลจากการทดลองพบว่าสามารถควบคุมกระบวนการทั้ง 2 ผ่านระบบดีซีเอสได้เป็นอย่างดี

# DISTRIBUTED CONTROL SYSTEMS

By

Mr. Sittiphon Kanokratmanee

Mrs. Sukatkana Jantalert

Acting Sub Lt. Nuttavut Kerdgongsup

Mr. Wichaen Wimanpradid

Advisor

Asst. Prof. Phornsuk Ratiroch-anant

Academic Year 2006

## ABSTRACT

This thesis is to study the Distributed Control System (DCS) and its application for controlling the first-order level process and second-order level process by using a microcontroller dsPIC 30F4011 as the process controller for each process. The communication between process and PC in DCS is performed by using RS232/485. The first-order process is controlled by PI mode while PID controller is used for the second-order process. The process reaction curve will be used together with Both Ziegler-Nichols Method or Dahlin Synthesis Method to obtain the PID parameters. After comparing both controlling methods from the experiment results, the Dahlin method can obtained the best control performance. This confirms the good control performance by using DCS strategy

## กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลงด้วยดี เนื่องด้วยได้รับการสนับสนุนอย่างดียิ่ง โดยคณะผู้จัดทำขอกราบขอบพระคุณบิดามารดาและบุพการีทุกท่านที่ได้อุปการะอย่างดียิ่ง ไม่ว่าจะเป็นผู้ให้ชีวิต ผู้ฟูมฟักเลี้ยงดู อบรมสั่งสอน ตลอดจนส่งเสริมด้านการศึกษา และขอบคุณสมาชิกในครอบครัวที่เป็นกำลังใจให้เสมอมา

ขอกราบขอบพระคุณอาจารย์พรสุข รัตโรจน์นันต์ อาจารย์ที่ปรึกษาโครงการเป็นอย่างสูง ที่สละเวลาอันมีค่าเพื่อมาให้คำปรึกษาและแนะนำแนวทางในการดำเนินโครงการจนทำให้โครงการประสบความสำเร็จเป็นอย่างดี และขอกราบขอบพระคุณอาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาและให้คำแนะนำกับคณะผู้จัดทำ ทำให้มีความรู้เพื่อมาใช้ในการดำเนินการ โครงการนี้ได้เป็นอย่างดีด้วยดี

ขอขอบคุณภาควิชาวิศวกรรมระบบควบคุมที่เอื้อเพื่อเครื่องมือและอุปกรณ์อย่างครบครัน รวมถึงเอื้อเพื่อสถานที่ในการทำโครงการขึ้นนี้ได้เป็นอย่างดี

ขอขอบคุณเพื่อนๆทุกคนที่คอยให้กำลังใจ และให้คำแนะนำ ตลอดจนสร้างเสียงหัวเราะที่ทำให้คณะผู้จัดทำผ่อนคลายจากความตึงเครียด ตลอดช่วงระยะเวลาการทำโครงการ

ผู้จัดทำ

นายสิทธิพล

กนกรัตน์มณี

นางสาวสุศักดา

จันทะเลิศ

ว่าที่ร้อยตรีณัฐวุฒิ

เกิดกองทัพ

นายวิเชียร

วิมานประดิษฐ์

# สารบัญ

	หน้า
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VIII
สารบัญตาราง	XIII
<b>บทที่ 1 บทนำ</b>	
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์	1
1.3 ขั้นตอนการศึกษาและจัดทำโครงการ	2
1.4 รายละเอียดของปริญญานิพนธ์	2
<b>บทที่ 2 ทฤษฎีและหลักการ</b>	4
2.1 ดีซีเอส (DCS)	4
2.1.1 ประวัติของดีซีเอส	4
2.1.2 ความหมายของดีซีเอส	7
2.1.3 ส่วนประกอบของดีซีเอส	7
2.2 อุปกรณ์ที่ใช้วัดและควบคุมกระบวนการ	13
2.2.1 อุปกรณ์ประเภทเครื่องมือวัดและควบคุม	13
2.2.2 อุปกรณ์ต่างๆที่ใช้ในการทำงานของระบบ	20
2.3 ทฤษฎีและหลักการควบคุมของพีไอดี (PID)	22
2.3.1 หน่วยควบคุมชนิดพีไอดี	22
2.3.2 แบบการควบคุมของหน่วยควบคุมชนิดพีไอดี	23
2.3.3 วัตถุประสงค์ของการติดตั้งหน่วยควบคุมชนิดพีไอดี	23
2.4 เครื่องควบคุมแบบพีไอดี	28
2.4.1 การควบคุมแบบโพรบ โพรชันแนล (Proportional)	28
2.4.2 การควบคุมแบบอินทิกรัล (Integral)	30
2.4.3 การควบคุมแบบเดริเวทีฟ (Derivative)	31

## สารบัญ(ต่อ)

	หน้า
2.4.4 การควบคุมแบบโพรบ โฟชันแนล-อินทิกรัล –เดริเวทีฟ	33
2.5 สมการดิฟเฟอเรนเชียล	34
2.5.1 การควบคุมแบบโพรบ โฟชันแนล (Proportional)	35
2.5.2 การอินทิเกรต (Integral)	35
2.5.3 การอนุพันธ์ (Derivative)	38
2.5.4 การออกแบบตัวควบคุมแบบพีไอดี	39
2.6 การวิเคราะห์คุณลักษณะของกระบวนการ	40
2.6.1 การวิเคราะห์คุณลักษณะของกระบวนการ	40
2.6.2 วิธีการหาพารามิเตอร์จากเส้นโค้งปฏิกิริยากระบวนการ (Process Reaction Curve)	42
2.7 วิธีการหาค่าพารามิเตอร์ในการปรับแต่งกระบวนการแบบต่างๆ	47
2.7.1 การควบคุมแบบอัตราส่วนการลดลงหนึ่งส่วนสี่ (Quarter Decay Ratio)	47
2.7.2 การหาพารามิเตอร์โดยวิธีอัลทิเมท (Ultimate Method)	48
2.7.3 การหาค่าพารามิเตอร์โดยวิธีเส้นโค้งปฏิกิริยากระบวนการ (Process Reaction Curve)	50
2.7.4 การสังเคราะห์ค่าพารามิเตอร์ของการควบคุมแบบพีไอดี ด้วยวิธีคาล์ดลิน	51
2.8 หลักการเขียนโปรแกรมวิซวล เบสิก 6 (Visual Basic 6)	55
2.8.1 จุดเด่นของวิซวล เบสิก	55
2.8.2 องค์ประกอบต่างๆ ของวิซวล เบสิก 6.0	55
2.8.3 รูปแบบการสร้างแอปพลิเคชันด้วยวิซวล เบสิก	57
2.8.4 ขั้นตอนการสร้างแอปพลิเคชันด้วยวิซวล เบสิก	59
2.8.5 ActiveX Control เบื้องต้น	60
2.8.6 การเชื่อมต่อแบบอนุกรม	61
2.8.7 ฟังก์ชัน ค่าคงที่ การส่งค่าพารามิเตอร์ และตัวอย่างโปรแกรมย่อยที่ใช้ใน โปรแกรมวิซวล เบสิก	75
2.9 การเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอก	81
2.9.1 การสื่อสารแบบอนุกรม	81
2.9.2 มาตรฐานพอร์ตอนุกรมแบบอาร์เอส-232	85

## สารบัญ(ต่อ)

	หน้า
2.9.3 คอนเน็กเตอร์สำหรับพอร์ตอาร์เอส-232 และการเชื่อมต่อ	86
2.9.4 UART	88
2.9.5 มาตรฐานการสื่อสารข้อมูลอนุกรมแบบอาร์เอส 485 (RS -485)	89
2.9.6 ความรู้เบื้องต้นเกี่ยวกับไมโครคอนโทรลเลอร์ dsPIC 30F4011	92
2.10 การแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอก และการแปลงสัญญาณ อนาลอกเป็นสัญญาณดิจิทัล	94
2.10.1 การแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก (Digital to Analog Converter, DAC)	94
2.10.2 คุณสมบัติของอุปกรณ์แปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก	95
2.10.3 วงจรแลคเคอร์ R-2R	97
2.10.4 แรงดันอนาล็อกเอาต์พุตของวงจร DAC (Voltage Output DAC)	100
2.10.5 การแปลงสัญญาณดิจิทัลเป็นอนาลอกด้วยไอซี ทีซี 1231 (TC1231)	100
2.10.6 การแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล	103
2.10.7 คุณสมบัติของตัวแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล (ADC)	103
2.10.8 การประมาณค่าโดยวิธีซัคเซสซีฟ (Successive Approximation )	105
<b>บทที่ 3 การออกแบบและโครงสร้าง</b>	107
3.1 การออกแบบดีซีเอสทีใช้ในการควบคุมระดับน้ำ	107
3.2 โครงสร้างการรับส่งข้อมูล ระหว่างคอมพิวเตอร์ควบคุมกระบวนการ กับ หน่วย เชื่อมต่อกระบวนการ	109
3.3 รูปแบบโปรแกรมของคอมพิวเตอร์หน่วยติดต่อกและปฏิบัติการของพนักงาน	110
3.4 แผนภูมิแสดงขบวนการปฏิบัติงาน (Flow Chart)	115
<b>บทที่ 4 การทดลองและผลการทดลอง</b>	128
4.1 กล่าวนำ	128
4.2 ผลการทดลองเครื่องมือวัดระดับน้ำของกระบวนการ	128
4.3 ผลการทดลองอุปกรณ์แปลงจากอนาลอกเป็นดิจิทัล (ADC) ในกระบวนการ อันดับหนึ่ง	130

## สารบัญ(ต่อ)

	หน้า
4.4 ผลการทดลองอุปกรณ์แปลงจากอนาลอกเป็นดิจิตอล(ADC)ในกระบวนการ 131 อันดับสอง	
4.5 ผลการทดลองอุปกรณ์แปลงจากดิจิตอลเป็นอนาลอก (DAC) ในกระบวนการ	132
4.6 ผลการทดลองระบบควบคุมระดับน้ำอันดับหนึ่ง	133
4.7 ผลการทดลองระบบควบคุมระดับน้ำอันดับสอง	136
<b>บทที่ 5 บทวิจารณ์และสรุป</b>	140
5.1 ผลการดำเนินงาน	140
5.2 การแก้ไขปัญหา	141
5.3 สรุปผลการทำโครงการ	141
5.4 แนวทางการพัฒนา	142
<b>ภาคผนวก</b>	
วงจรควบคุมกระบวนการอันดับหนึ่ง	144
วงจรควบคุมกระบวนการอันดับสอง	145
วงจรแหล่งจ่ายไฟ	146
วงจรอนุกรม RS485	147
วงจรแปลงอนุกรม RS485/RS232	148
<b>บรรณานุกรม</b>	149

# สารบัญรูป

รูปที่	หน้า
2.1 การเชื่อมโยงสายสัญญาณของระบบควบคุมแบบเดิม	5
2.2 ระบบควบคุมแบบดิจิทัล	6
2.3 การติดตั้งระบบควบคุมแบบดิจิทัล	8
2.4 โครงสร้างของเครื่องมือสำหรับควบคุมกระบวนการผลิตติดตั้งในบริเวณพื้นที่ผลิต	8
2.5 โครงสร้างของวาล์วควบคุม	14
2.6 วาล์วควบคุม	14
2.7 สมบัติการไหลเฉพาะของวาล์ว	15
2.8 เครื่องวัดความดันแบบดิฟเฟอเรนเชียล	16
2.9 ส่วนประกอบพื้นฐานของอุปกรณ์แปลงกระแสไฟฟ้าเป็นความดันลม	17
2.10 เอสแอลซีดี อินดิเคตติ้ง คอนโทรลเลอร์	18
2.11 แมกเนติก โฟลว์ มิเตอร์	19
2.12 ถังพักน้ำ	20
2.13 ถังวัดระดับน้ำ	20
2.14 ป้อนน้ำ	21
2.15 ตัวจ่ายลม (Air supply)	21
2.16 หน่วยควบคุมแบบพีไอดี	22
2.17 การควบคุมแบบป้อนกลับ	24
2.18 แนวคิดของการควบคุมแบบคาสเคด	24
2.19 การควบคุมแบบคาสเคด	26
2.20 วงควบคุมป้อนไปหน้า	26
2.21 การควบคุมแบบป้อนกลับและป้อนไปหน้า	27
2.22 แผนภาพบล็อกตัวควบคุมแบบ โพรบ โพชั่นเน็ล	28
2.23 ผลของการควบคุมแบบ โพรบ โพชั่นเน็ลต่อเอาต์พุทของตัวควบคุม	29
2.24 ผลของ โพรบ โพชั่นเน็ล แบนด์ต่อการเปิดวาล์ว	29
2.25 ค่า โพรบ โพชั่นเน็ล ออฟเซต	30
2.26 แผนภาพบล็อกการควบคุมแบบ โพรบ โพชั่นเน็ล-อินทิกรัล	30

## สารบัญรูป (ต่อ)

รูปที่	หน้า
2.27 การตอบสนองต่อค่าผิดพลาดที่เป็นขั้นบันไดของตัวควบคุมแบบโพรบโพซันแนล- อินทิกรัล	31
2.28 แผนภาพบล็อกการควบคุมแบบโพรบโพซันแนล-เดริเวทิฟ	32
2.29 เอาท์พุทของตัวควบคุมแบบโพรบโพซันแนล –เดริเวทิฟ เมื่ออินพุทเป็นฟังก์ชันเร็มพ์	32
2.30 แผนภาพบล็อกสำหรับตัวควบคุมพีไอดี	33
2.31 การตอบสนองของตัวควบคุมโหมดต่าง ๆ	34
2.32 การอินทิเกรตวิธีสี่เหลี่ยมผืนผ้า	35
2.33 การอินทิเกรตวิธีสี่เหลี่ยมคางหมู	36
2.34 การอินทิเกรตวิธีซิมสัน 1/3	37
2.35 แสดงลักษณะการอนุพันธ์	38
2.36 บล็อกไดอะแกรมของระบบควบคุมแบบป้อนกลับ	40
2.37 บล็อกไดอะแกรมของระบบควบคุมแบบป้อนกลับหลังการลดรูป	41
2.38 บล็อกไดอะแกรมสำหรับการทดสอบกระบวนการแบบวงเปิด	42
2.39 ผลตอบสนองจากการทดสอบกระบวนการแบบวงเปิด	43
2.40 ผลตอบสนองขั้นบันไดของระบบ FODPT	44
2.41 ค่าพารามิเตอร์แบบจำลอง FOPDT จากวิธีพีดี1	45
2.42 ค่าพารามิเตอร์แบบจำลอง FOPDT จากวิธีพีดี2	46
2.43 ค่าพารามิเตอร์แบบจำลอง FOPDT จากวิธีพีดี3	47
2.44 เส้นโค้งตอบสนองกระบวนการสำหรับอัตราส่วนการลดลงหนึ่งส่วนสี่	48
2.45 การตอบสนองที่แสดงค่าอัลทิเมทเคนและคาบอัลทิเมท	49
2.46 ระบบควบคุมวงปิด	51
2.47 แสดงผลตอบสนองแบบปิดที่มีการหน่วงเวลาเป็นเวลา $t_0$	52
2.48 แสดงองค์ประกอบต่างๆ ของวิซวล เบสิก	56
2.49 ไดอะล็อก New Project	57
2.50 แสดง ActiveX Control พื้นฐานที่ปรากฏใน Toolbox	60
2.51 แสดงการเพิ่ม Microsoft Comm Control 6.0 ใช้งานกับพอร์ตอนุกรม	61
2.52 การสื่อสารแบบซิงโครนัส	82
2.53 การสื่อสารข้อมูลแบบอะซิงโครนัส	82

## สารบัญรูป (ต่อ)

รูปที่	หน้า
2.54 การส่งข้อมูลขนาด 8 บิตที่ใช้ในการโอนย้ายข้อมูลแบบอนุกรม	84
2.55 การใช้บิตพาริตีเพื่อตรวจสอบความผิดพลาดในการส่งข้อมูลแบบอนุกรม	84
2.56 DTE และ DCE	85
2.57 แสดงการจัดขาคอนเน็กเตอร์พอร์ตอนุกรมและลักษณะการต่อกับอุปกรณ์ภายนอก	86
2.58 แสดงการส่งข้อมูลแบบซิมเพล็กซ์	88
2.59 แสดงการส่งข้อมูลแบบฮาล์ฟดูเพล็กซ์	89
2.60 แสดงการส่งข้อมูลแบบฟูลดูเพล็กซ์	89
2.61 แสดงระบบ RS-485 แบบ 2 สาย	91
2.62 แสดงระบบอาร์เอส-485 แบบ 4 สาย	92
2.63 โครงสร้างของขาไมโครคอนโทรลเลอร์ dsPIC 30F4011	94
2.64 ระบบการแปลงสัญญาณดิจิทัลเป็นอนาล็อก	95
2.65 ทรานส์เฟอร์ เคิร์ฟ ในอุดมคติของ DAC 3 บิต	95
2.66 คลื่นไซน์ที่สร้างจาก DAC	96
2.67 วงจรแลดเดอร์ R-2R แปลงสัญญาณดิจิทัลให้เป็นกระแสเอาต์พุต Iout	98
2.68 การเปลี่ยนแปลงกระแสเอาต์พุต Iout ของวงจรแลดเดอร์ R-2R ให้เป็นแรงดันเอาต์พุต	99
2.69 โครงสร้างขาของตัวแปลงสัญญาณดิจิทัลเป็นอนาล็อก TC 1321	100
2.70 บล็อกไดอะแกรมของ TC 1321	101
2.71 รูปแบบการประยุกต์ใช้งาน ของ TC 1321	102
2.72 วงจร ADC ขนาด 4 บิต	
(ก) สัญลักษณ์ของวงจร (ข) กราฟคุณสมบัติระหว่างอินพุตกับเอาต์พุต	104
2.73 โพลลิวาร์ตวิธีการทำงานของการประมาณค่าโดยวิธีซัคเซสซีฟ	105
3.1 แสดงรูปแบบการติดต่อที่ออกแบบ	108
3.2 แสดงการรับส่งข้อมูล	109
3.3 แสดงการกำหนดรูปแบบการสื่อสารของระบบบิตซีเอส	110
3.4 แสดงหน้าต่างหลักของโปรแกรมติดต่อของพนักงานผู้ใช้	111
3.5 แสดงหน้าต่างกลุ่มกระบวนของ โปรแกรมติดต่อของพนักงานผู้ใช้	112
3.6 แสดงหน้าต่างแบบจำลองอันดับหนึ่งของ โปรแกรมติดต่อของพนักงานผู้ใช้	113

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.7 แสดงหน้าต่างแบบจำลองอันดับสองของ โปรแกรมติดต่อของพนักงานผู้ใช้	114
3.8 แสดงหน้าต่างบันทึกสัญญาณระบบอันดับหนึ่งของ โปรแกรมติดต่อของพนักงานผู้ใช้	114
3.9 แสดงหน้าต่างบันทึกสัญญาณระบบอันดับสองของ โปรแกรมติดต่อของพนักงานผู้ใช้	115
3.10 โฟลวชาร์ตแสดงการทำงานของกระบวนการอันดับหนึ่ง	116
3.11 โฟลวชาร์ตกระบวนการอันดับหนึ่งเมื่อเกิดการอินเตอร์รัพท์ในโหมดAuto	117
3.12 โฟลวชาร์ตกระบวนการอันดับหนึ่งเมื่อเกิดการอินเตอร์รัพท์ใน โหมด Manual	118
3.13 โฟลวชาร์ตแสดงการทำงานของกระบวนการอันดับสอง	119
3.14 โฟลวชาร์ตแสดงการอินเตอร์รัพท์ในการรับค่าของกระบวนการอันดับหนึ่ง	120
3.15 โฟลวชาร์ตแสดงการอินเตอร์รัพท์ในการส่งค่าของกระบวนการอันดับหนึ่ง	121
3.16 โฟลวชาร์ตแสดงการอินเตอร์รัพท์ในการรับค่าของกระบวนการอันดับสอง	122
3.17 โฟลวชาร์ตแสดงการอินเตอร์รัพท์ในการส่งค่าของกระบวนการอันดับสอง	123
3.18 แสดงโฟลวชาร์ตการทำงานของโปรแกรมแสดงผลหน้าต่างหลัก	124
3.19 โฟลวชาร์ตแสดงการทำงานของ โปรแกรมหน้าต่างแสดงผลรวม	125
3.20 โฟลวชาร์ตแสดงการทำงานของกระบวนการอันดับหนึ่งและกระบวนการ อันดับสอง ในส่วนของหน้าจอแสดงผล	126
4.1 กราฟเปรียบเทียบสัญญาณที่ได้จากเครื่องมือวัดของกระบวนการอันดับหนึ่ง	129
4.2 กราฟเปรียบเทียบสัญญาณที่ได้จากเครื่องมือวัดของกระบวนการอันดับสอง	129
4.3 กราฟที่ได้จากการทดลองแปลงอนาลอกเป็นดิจิตอลในกระบวนการอันดับหนึ่ง	131
4.4 กราฟที่ได้จากการทดลองแปลงอนาลอกเป็นดิจิตอลในกระบวนการอันดับสอง	132
4.5 กราฟที่ได้จากการทดลองอุปกรณ์แปลงจากดิจิตอลเป็นอนาลอกในกระบวนการ อันดับหนึ่ง	133
4.6 กราฟที่ได้จากการทดลองอุปกรณ์แปลงจากดิจิตอลเป็นอนาลอกในกระบวนการ อันดับสอง	133
4.7 ผลตอบสนองของระบบควบคุมแบบพีไอในกระบวนการอันดับหนึ่ง ด้วยวิธีซีเกลอร์และนิโคลส์	134
4.8 ผลตอบสนองของระบบควบคุมแบบพีไอในกระบวนการอันดับหนึ่งด้วยวิธีคาล์ลิน	134
4.9 ผลตอบสนองของระบบควบคุมแบบพีไอในกระบวนการอันดับหนึ่ง หลังจากปรับปรุงค่าพารามิเตอร์โดยเปลี่ยนค่าร้อยละ 60 เปอร์เซ็นต์	135

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.10 ผลตอบสนองของระบบควบคุมแบบพีไอในกระบวนการอันดับหนึ่ง หลังจากปรับปรุงค่าพารามิเตอร์ โดยเปลี่ยนค่าครั้งละ 10 เปอร์เซ็นต์	135
4.11 ผลตอบสนองของระบบควบคุมแบบพีไอในกระบวนการอันดับสอง ด้วยวิธีซีเกลอร์และนิโคลส์	136
4.12 ผลตอบสนองของระบบควบคุมแบบพีไอในกระบวนการอันดับสองด้วยวิธีคาล์ลิน	137
4.13 ผลตอบสนองของระบบควบคุมแบบพีไอดีในกระบวนการอันดับสอง ด้วยวิธีซีเกลอร์และนิโคลส์	138
4.14 ผลตอบสนองของระบบควบคุมแบบพีไอดีในกระบวนการอันดับสองด้วยวิธีคาล์ลิน	138
4.15 ผลตอบสนองของระบบควบคุมแบบพีไอดีในกระบวนการอันดับสอง หลังจากปรับปรุงค่าพารามิเตอร์ โดยเปลี่ยนค่าครั้งละ 60 เปอร์เซ็นต์	139
4.16 ผลตอบสนองของระบบควบคุมแบบพีไอดีในกระบวนการอันดับสอง หลังจากปรับปรุงค่าพารามิเตอร์ โดยเปลี่ยนค่าครั้งละ 10 เปอร์เซ็นต์	139

# สารบัญตาราง

ตารางที่	หน้า
2.1 สมการของการคณิตศาสตร์ควบคุมโหมดต่างๆ	33
2.2 ค่าพารามิเตอร์โดยวิธีการอัลทิเมท	50
2.3 ค่าพารามิเตอร์โดยวิธีการเส้นโค้งปฏิบัติการกระบวนการ	51
2.4 แสดงสูตรสำหรับหาค่าพารามิเตอร์ของตัวควบคุมแบบพีไอดีตามวิธีค่าหาลิ้น	54
2.5 ตารางแสดงประเภทของแอปพลิเคชัน	57
2.6 แสดงหน้าที่ของ ActiveX Control แต่ละแบบ	60
2.7 แสดงค่าคงที่สำหรับคุณสมบัติ Handshake	73
2.8 แสดงค่าคงที่สำหรับคุณสมบัติ OnComm	74
2.9 แสดงค่าคงที่สำหรับคุณสมบัติ Error	74
2.10 แสดงค่าคงที่สำหรับคุณสมบัติ InputMode	74
2.11 ชนิดข้อมูลต่างๆที่ใช้ใน โปรแกรมมิชวล เบสิก	76
2.12 คำอธิบายและอัตราการเร็วในการส่งข้อมูล	83
2.13 หน้าที่ของขาต่างๆใน คอนเน็กเตอร์แบบ DB-25 ตัวผู้ และ DB-9 ตัวผู้	87
2.14 ข้อดี-ข้อเสียของการต่อระบบแบบ 2 สาย และ 4 สาย	92
2.15 รายละเอียดของขา TC 1321	102
4.1 ผลการทดลองเครื่องมือวัดระดับน้ำของกระบวนการ	128
4.2 ผลการทดลองอุปกรณ์แปลงจากอนาลอกเป็นดิจิตอลในกระบวนการอันดับหนึ่ง	130
4.3 ผลการทดลองอุปกรณ์แปลงจากอนาลอกเป็นดิจิตอลในกระบวนการอันดับสอง	131
4.4 ผลการทดลองอุปกรณ์แปลงจากดิจิตอลเป็นอนาลอก(DAC)ในกระบวนการ	132
4.5 ค่าพารามิเตอร์การควบคุมแบบพีไอของกระบวนการควบคุมระดับน้ำอันดับหนึ่ง	133
4.6 ค่าพารามิเตอร์การควบคุมแบบพีไอของกระบวนการควบคุมระดับน้ำอันดับสอง	136
4.7 ค่าพารามิเตอร์การควบคุมแบบพีไอดีของกระบวนการควบคุมระดับน้ำอันดับสอง	137

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญญานิพนธ์

ในปัจจุบันระบบควบคุมในโรงงานอุตสาหกรรมล้วนแต่เป็นระบบควบคุมอัตโนมัติ ซึ่งระบบควบคุมอัตโนมัติเริ่มมีการใช้โดย เจมส์ วัตต์ ในปี พ.ศ. 2331 แต่ความสามารถทางด้านเทคโนโลยีค่อนข้างจำกัด ผู้ผลิตเครื่องมือจึงมีการคิดค้นและพัฒนากระบวนการควบคุมขึ้นใหม่โดยแก้ไขจุดบกพร่องเดิม และเนื่องด้วยปัจจุบันมีเทคโนโลยีใหม่ๆที่ทำให้การควบคุมระบบเป็นไปอย่างมีประสิทธิภาพ และได้ผลผลิตที่ดีขึ้น มีผลทำให้การควบคุมโรงงานอุตสาหกรรมมีความซับซ้อนมากขึ้น และทำให้การควบคุมเป็นไปได้ยากจากการที่ต้องควบคุมจากหน้าจอที่มีจำนวนมาก คีชีเอสจึงถูกคิดค้นเพื่อตอบสนองความต้องการของผู้ใช้ทำให้คือระบบควบคุมมีความน่าเชื่อถือ , ทำให้พนักงานที่ทำงานทำได้ง่ายขึ้น รวมถึงการบำรุงรักษาและแก้ไขการกระบวนการผลิตก็ทำได้ง่าย และแก้ไขจุดบกพร่องของระบบควบคุมแบบเดิม คีชีเอสจึงเป็นเครื่องมือสำหรับตรวจสอบและควบคุมการผลิตที่เหมาะสมกับโรงงานอุตสาหกรรมในปัจจุบัน

ในปัญญานิพนธ์นี้ เป็นการศึกษากระบวนการบิตชีเอสและสร้างระบบบิตชีเอส เพื่อควบคุมกระบวนการ 2 กระบวนการคือ กระบวนการควบคุมระดับน้ำอันดับหนึ่งและกระบวนการควบคุมระดับน้ำอันดับสอง โดยกระบวนการทั้งสองอยู่ห่างออกจากกัน ผู้ใช้สามารถควบคุมและตรวจสอบการทำงานของกระบวนการทั้งสองได้จากคอมพิวเตอร์เพียงเครื่องเดียว ซึ่งคอมพิวเตอร์และกระบวนการทั้งสองเชื่อมต่อกันด้วยเครือข่ายคอมพิวเตอร์

### 1.2 วัตถุประสงค์ในการทำปัญญานิพนธ์

1. เพื่อศึกษาการทำงานของอุปกรณ์เครื่องมือวัดและควบคุมทางอุตสาหกรรมเพื่อสร้างระบบควบคุมระดับน้ำ
2. เพื่อศึกษาการรับส่งข้อมูลกับอุปกรณ์ภายนอกของคอมพิวเตอร์โดยผ่านพอร์ตอนุกรม
3. เพื่อศึกษาหลักการและวิธีการควบคุมกระบวนการด้วยระบบบิตชีเอสที่ใช้ในโรงงานอุตสาหกรรม
4. เพื่อศึกษาหลักการและแนวคิดในการเขียนโปรแกรมเพื่อใช้ในการสร้างตัวคอนโทรลเลอร์
5. เพื่อศึกษาหลักการและแนวคิดในการเขียนโปรแกรมเพื่อใช้ในการแสดงผลและควบคุมกระบวนการ

6. เพื่อศึกษาการวิเคราะห์หาค่าพารามิเตอร์ของตัวควบคุมพีไอดีที่ใช้ในการควบคุมกระบวนการ และปรับแต่งค่าพารามิเตอร์เพื่อให้ได้ค่าพารามิเตอร์ที่เหมาะสมกับกระบวนการที่ต้องการควบคุม

### 1.3 ขั้นตอนการศึกษาและจัดทำโครงการ

1. ทำการตรวจสอบกระบวนการที่ต้องการ ควบคุม
2. ทำการซ่อมแซมและปรับปรุงในส่วนที่ชำรุดเสียหายให้สามารถใช้งานได้
3. ศึกษาทฤษฎีการควบคุมแบบต่างๆ
4. ศึกษาส่วนประกอบของกระบวนการและตัวอุปกรณ์ที่ใช้ในการควบคุมและวัดสัญญาณ
5. ศึกษาและทดลองการควบคุมด้วยเอสแอลซีดี อินดิเคตติ้ง คอนโทรลเลอร์ (SLCD Indicating Controller)
6. ศึกษาวิธีการปรับแต่งค่าพารามิเตอร์ที่ใช้ในการปรับแต่งค่าพีไอดีของตัวคอนโทรลเลอร์
7. ศึกษาการใช้งานโปรแกรมวิซวลเบสิก (Visual Basic) และเขียนหน้าต่างแสดงผล
8. ศึกษาการใช้งาน dsPIC และการเขียนโปรแกรมเพื่อควบคุมไมโครคอนโทรลเลอร์ (Microcontroller)
9. ทดสอบการควบคุมและบันทึกผล
10. สรุปและเปรียบเทียบการควบคุมแบบต่างๆที่ได้ศึกษา
11. ทำรายงานและเตรียมส่งโครงการ

### 1.4 รายละเอียดของปฏิญานิพนธ์

เนื้อหาที่จะกล่าวในปฏิญานิพนธ์ฉบับนี้ประกอบด้วย

บทที่ 1 บทนำ กล่าวถึงวัตถุประสงค์ หลักการใหม่ ขั้นตอนการศึกษา และการจัดทำโครงการ พร้อมทั้งรายละเอียดของปฏิญานิพนธ์ของแต่ละบท

บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง กล่าวถึงอุปกรณ์ต่างๆที่ใช้ในกระบวนการควบคุมระดับน้ำ ทฤษฎีที่เกี่ยวข้องกับการควบคุม, วงจรต่างๆที่ใช้ในโครงการ, การเขียนโปรแกรมวิซวลเบสิก, และไมโครคอนโทรลเลอร์ รวมถึงการติดต่อข้อมูลระหว่างกระบวนการกับคอมพิวเตอร์ และนำเอาความรู้ไปประยุกต์ใช้ในการจัดทำโครงการ

บทที่ 3 หลักการออกแบบ นำเสนอการประกอบโครงสร้างของระบบ รวมถึงแนวคิดในการออกแบบระบบควบคุม

บทที่ 4 การทดลอง เป็นส่วนการทดสอบองค์ประกอบต่างๆ ในระบบ ตลอดจนการทดลองการใช้งานระบบดีซีเอสที่สร้างขึ้น หลังจากหาค่าพารามิเตอร์พีไอดี (PID) เพื่อใช้ควบคุมกระบวนการ

บทที่ 5 บทวิจารณ์และสรุป จะสรุปผลการดำเนินงาน ปัญหาที่เกิดขึ้น และแนวทางการปรับปรุงพัฒนาโครงการนี้ต่อไป



## บทที่ 2

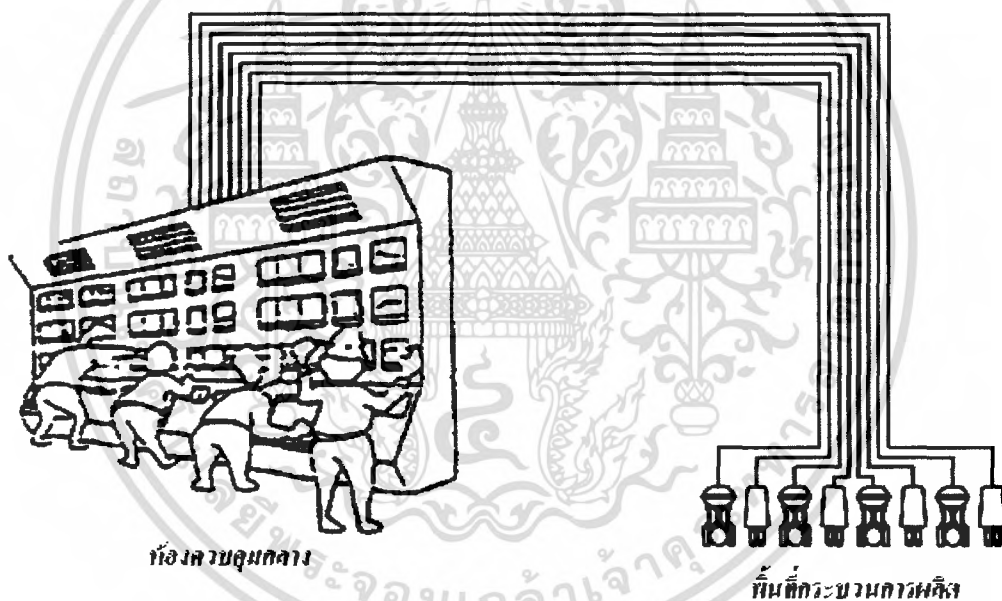
# ทฤษฎีและหลักการ

### 2.1 ดีซีเอส (DCS)

#### 2.1.1 ประวัติของดีซีเอส

ดีซีเอส (DCS) ย่อมาจาก Distributed Control System เป็นเครื่องมือสำหรับควบคุมกระบวนการผลิตในโรงงานอุตสาหกรรมซึ่งได้รับการปรับปรุงและพัฒนาให้เหมาะสมกับสภาพแวดล้อมของโรงงานอุตสาหกรรมและสถานการณ์ปัจจุบัน เดิมระบบควบคุมในโรงงานอุตสาหกรรมติดตั้งเครื่องควบคุมชนิดนิวเมติกส์ทั้งหมดไว้ในห้องควบคุมกลาง (central control room) ของโรงงานอุตสาหกรรม โดยเชื่อมต่อสัญญาณมาตรฐาน 3-15 ปอนด์/คร.นิ้ว ระหว่างเครื่องวัดและส่งสัญญาณทั้งหมดในบริเวณกระบวนการผลิตกับเครื่องควบคุมในห้องควบคุม ทำให้พนักงานสามารถตรวจสอบสภาพของกระบวนการผลิตและดูแลระบบควบคุมทั้งหมดจากภายในห้องควบคุม โดยไม่ต้องออกไปตรวจสอบในบริเวณพื้นที่การผลิตนอกห้องควบคุม การตรวจสอบและควบคุมกระบวนการผลิตจึงสามารถดำเนินไปได้อย่างมีประสิทธิภาพและรวดเร็วยิ่งขึ้น เมื่อเปรียบเทียบกับระบบควบคุมเดิมก่อนหน้านั้น ในขณะที่ยังใช้อุปกรณ์ควบคุมชนิดนิวเมติกส์ และไม่มีเครื่องวัดและส่งสัญญาณชนิดนิวเมติกส์ไม่มีการกำหนดสัญญาณมาตรฐาน และมีเครื่องควบคุมชนิดนิวเมติกส์ทำให้จำเป็นต้องติดตั้งอุปกรณ์ควบคุมไว้ในบริเวณกระบวนการผลิตเป็นจุดๆ ทั่วโรงงานอุตสาหกรรม การควบคุมกระบวนการผลิตทั้งหมดของโรงงานอุตสาหกรรมจากเครื่องควบคุมภายในห้องควบคุมกลาง เพียงแห่งเดียวเรียกว่า การควบคุมแบบศูนย์กลาง (centralized control) และการควบคุมกระบวนการผลิตโดยติดตั้งอุปกรณ์ควบคุมใกล้กับกระบวนการผลิตเป็นจุดๆ ทั่วโรงงานอุตสาหกรรม เรียกว่าการควบคุมแบบกระจาย (distributed control) แม้ว่าต่อมาจะมีการใช้เครื่องควบคุมเครื่องวัดและเครื่องส่งสัญญาณชนิดอิเล็กทรอนิกส์ แทนเครื่องมือชนิดนิวเมติกส์ทำให้ระยะทางในการติดต่อระหว่างเครื่องวัดและเครื่องส่งสัญญาณในกระบวนการผลิตกับเครื่องควบคุมในห้องสามารถมีระยะห่างได้มากขึ้น แต่ระบบควบคุมยังคงรักษาโครงสร้างของการควบคุมแบบศูนย์กลางไว้เช่นเดิม เพียงแต่การใช้อุปกรณ์และเครื่องมือชนิดอิเล็กทรอนิกส์และใช้สัญญาณไฟฟ้ามาตรฐานขนาด 4-20 มิลลิแอมแปร์ แทนการใช้สัญญาณเดิมทำให้โรงงานอุตสาหกรรมสามารถขยายขนาดให้ใหญ่ขึ้นเพื่อรองรับความต้องการขยายกำลังการผลิตของโรงงานอุตสาหกรรม

โรงงานอุตสาหกรรมขนาดใหญ่ที่ต้องการขยายกำลังการผลิตของคนตลอดเวลา ทำให้ห้องควบคุมกลางของโรงงานอุตสาหกรรมต้องมีขนาดใหญ่มากจึงจะมีพื้นที่เพียงพอสำหรับติดตั้งอุปกรณ์และเครื่องควบคุมในโรงงานอุตสาหกรรมทั้งหมด ซึ่งเป็นอุปสรรคสำหรับการปฏิบัติงานของพนักงานในสมัยนั้นที่คอยเดินตรวจสอบเครื่องมือจำนวนมากบนแผงควบคุมตลอดเวลา จึงมีความพยายามในการแก้ปัญหาดังกล่าว โดยการกระจายหน้าที่ของเครื่องมือสำหรับการควบคุมการผลิตออกเป็นส่วนๆ แต่สามารถเก็บรวบรวมข้อมูลของกระบวนการผลิตทั้งหมดแสดงให้พนักงานควบคุมทราบและตรวจสอบได้ตลอดเวลา เพื่อให้การควบคุมกระบวนการผลิตดำเนินไปอย่างถูกต้องและมีประสิทธิภาพ แต่เนื่องจากความก้าวหน้าทางด้านวิทยาศาสตร์และเทคโนโลยีที่เกี่ยวข้องในขณะนั้นยังไม่เพียงพอสำหรับการแก้ปัญหาทางการเก็บรวบรวมข้อมูลและการสื่อสารระหว่างเครื่องมือควบคุมกระบวนการผลิตแต่ละส่วนทั่วโรงงานอุตสาหกรรมจึงยังไม่มีการพัฒนาเครื่องควบคุมแบบกระจายอย่างจริงจังในขณะนั้น



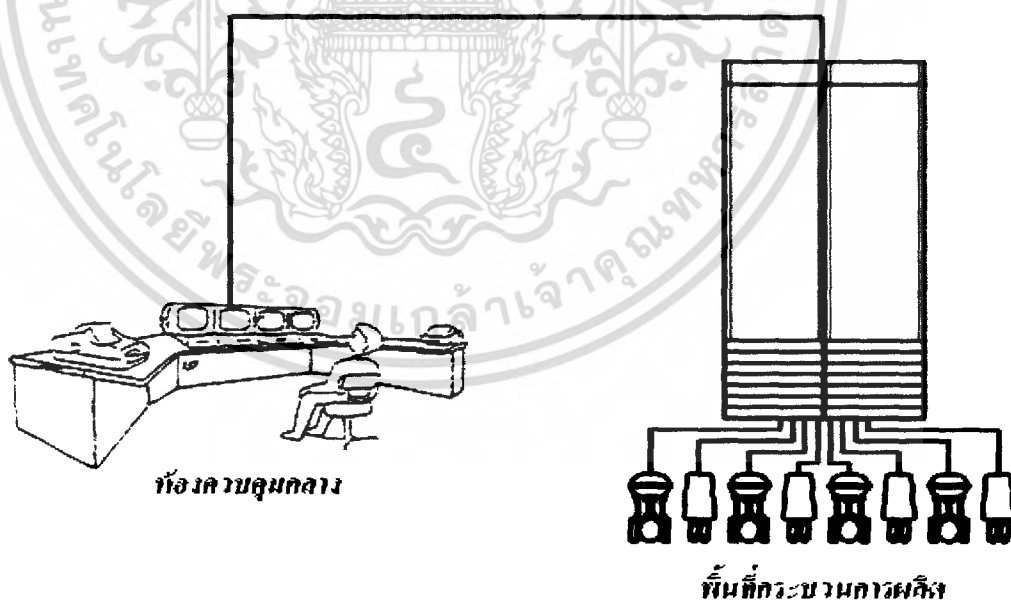
รูปที่ 2.1 การเชื่อมโยงสายสัญญาณของระบบควบคุมแบบเดิม

การติดตั้งเครื่องคอมพิวเตอร์ในโรงงานอุตสาหกรรมสำหรับตรวจสอบและควบคุมกระบวนการผลิตทำให้สามารถกำหนดโครงสร้างและเปลี่ยนแปลงคุณสมบัติของระบบควบคุมโดยวิธีเขียนและแก้ไข โปรแกรมในหน่วยความจำของเครื่องคอมพิวเตอร์ แทนวิธีเดินสายสัญญาณเชื่อมต่อระหว่างอุปกรณ์และเครื่องมือควบคุมต่างๆ ความสามารถของระบบควบคุมในการควบคุมกระบวนการผลิตจึงไม่ถูกจำกัดด้วยคุณสมบัติของอุปกรณ์และเครื่องควบคุมชนิดนิวแมติกส์และอิเล็กทรอนิกส์ เช่นเดียวกับระบบควบคุมแบบเดิม แต่พนักงานในโรงงานยังคงต้อง

ปฏิบัติหน้าที่อยู่ร่วมกันภายในห้องควบคุมกลางเช่นเดียวกับระบบควบคุมที่ติดตั้งอุปกรณ์และเครื่องควบคุมชนิดนิวเมติกส์และอิเล็กทรอนิกส์ โรงงานอุตสาหกรรมที่ติดตั้งเครื่องคอมพิวเตอร์จึงมีจอภาพและเป็นพิมพ์ของเครื่องคอมพิวเตอร์เพิ่มขึ้นภายในห้องควบคุม

ดีซีเอส รุ่นแรกถูกสร้างขึ้นเมื่อ พ.ศ. 2517 เป็นเครื่องมือสำหรับควบคุมกระบวนการผลิตแทนเครื่องควบคุมชนิดนิวเมติกส์และอิเล็กทรอนิกส์ โดยแยกอุปกรณ์สำหรับควบคุมการผลิตและอุปกรณ์สำหรับติดต่อและปฏิบัติงานของพนักงานออกจากกัน อุปกรณ์สำหรับควบคุมกระบวนการผลิตติดตั้งในบริเวณพื้นที่การผลิต ทำหน้าที่รับสัญญาณวัดของกระบวนการผลิตจากเครื่องวัดและส่งสัญญาณควบคุมสำหรับควบคุมกระบวนการผลิตไปยังวาล์วควบคุม ส่วนอุปกรณ์สำหรับติดต่อและปฏิบัติงานของพนักงานติดตั้งในบริเวณห้องควบคุมสำหรับพนักงาน ทำหน้าที่ตรวจสอบและควบคุมกระบวนการผลิต โดยระหว่างอุปกรณ์ควบคุมกระบวนการผลิตและอุปกรณ์ติดต่อและปฏิบัติงานของพนักงานเชื่อมโยงติดต่อกัน โดยเครือข่ายคอมพิวเตอร์

อุปกรณ์สำหรับควบคุมกระบวนการผลิตจะติดตั้งตามตำแหน่งต่างๆ ในบริเวณกระบวนการผลิตทั่วโรงงานอุตสาหกรรมเป็นการกระจายหน้าที่การควบคุมกระบวนการผลิตให้กับเครื่องมือสำหรับควบคุมกระบวนการผลิต และติดตั้งอุปกรณ์ติดต่อและปฏิบัติงานของพนักงานในบริเวณห้องควบคุมกลาง ทำให้พนักงานสามารถตรวจสอบข้อมูลจากกระบวนการผลิตและควบคุมกระบวนการผลิตของโรงงานทั้งหมดจากภายในห้องควบคุมเพียงแห่งเดียว



รูปที่ 2.2 ระบบควบคุมแบบดีซีเอส

ดีซีเอสทำให้วิธีการปฏิบัติงานของพนักงานเพื่อตรวจสอบและดูแลกระบวนการผลิตในโรงงานอุตสาหกรรมแตกต่างจากพนักงานในโรงงานอุตสาหกรรมที่ยังใช้อุปกรณ์และเครื่องควบคุมแบบเดิมอย่างสิ้นเชิง จากภายในห้องควบคุมของโรงงานอุตสาหกรรมแบบเดิมที่ยังต้องติดตั้งมาตรวัดและเครื่องควบคุมอุปกรณ์อื่นอีกเป็นจำนวนมากบนแผงควบคุม และพนักงานต้องเดินตรวจสอบอ่านค่าจากมาตรวัดแล้วจึงตัดสินใจสั่งงานหรือปรับเครื่องควบคุมอยู่ตลอดเวลา การติดตั้งดีซีเอสทำให้ภายในห้องควบคุมไม่จำเป็นต้องมีแผงของอุปกรณ์และเครื่องมือขนาดใหญ่ อีกต่อไป เปลี่ยนมาติดตั้งจอภาพและเป็นพิมพ์ ซึ่งเป็นอุปกรณ์สำหรับทำหน้าที่ติดต่อและปฏิบัติการของพนักงานแทน ในจอภาพจะเป็นภาพคอมพิวเตอร์กราฟฟิก แสดงภาพจำลองของแผงหน้าปัดเครื่องมือต่างๆและภาพจำลองของเครื่องจักรและอุปกรณ์ในกระบวนการผลิตเพื่อเพิ่มความสะดวกแก่พนักงานในการตรวจสอบและควบคุมกระบวนการผลิต วิธีการปฏิบัติงานของพนักงานก็เปลี่ยนมานั่งประจำอยู่หน้าจอภาพและเป็นพิมพ์ของดีซีเอส แทนการเคลื่อนที่ไปยังตำแหน่งต่างๆรอบห้องควบคุมเพื่อตรวจสอบเครื่องวัดและปรับเครื่องควบคุมในการควบคุมแบบเดิม

### 2.1.2 ความหมายของดีซีเอส

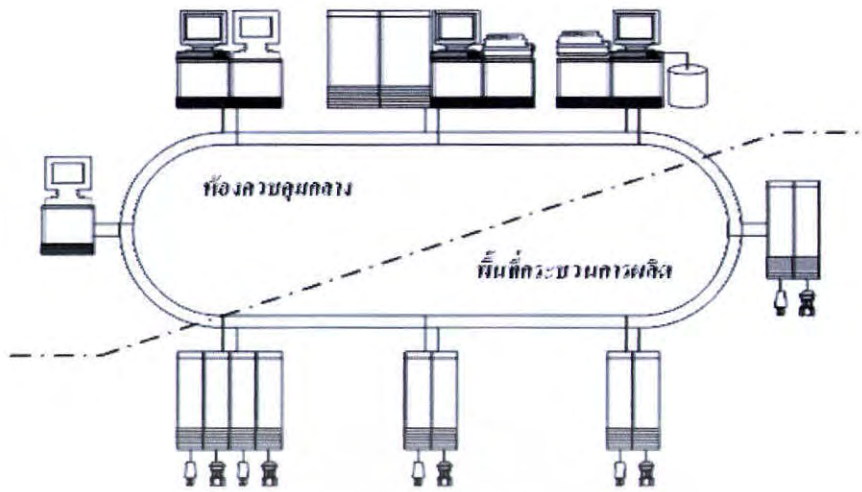
ดีซีเอส คือ เครื่องข่ายคอมพิวเตอร์สำหรับควบคุมกระบวนการผลิต โดยเชื่อมโยงสายสัญญาณเครือข่ายระหว่างเครื่องคอมพิวเตอร์กับเครื่องมือวัดและควบคุมในกระบวนการผลิต เครื่องข่ายคอมพิวเตอร์ของดีซีเอสอาจจะเป็นเครือข่ายเส้นทางเดียวหรือแบบวงแหวนที่ได้รับการปรับปรุงให้นำเชื่อถือมากขึ้น

### 2.1.3 ส่วนประกอบของดีซีเอส

ดีซีเอสแต่ละบริษัทอาจมีโครงสร้างที่ต่างกันตามการออกแบบของแต่ละบริษัท แต่ดีซีเอสของทุกบริษัทต้องแบ่งอุปกรณ์ประกอบภายในของดีซีเอสตามหน้าที่การปฏิบัติงาน เป็นหน่วยเครื่องมือเสมอ แม้ว่าอุปกรณ์ประกอบของดีซีเอสแต่ละบริษัทจะเรียกชื่อต่างกัน แต่ดีซีเอสต้องประกอบเครื่องมือที่มีลักษณะคล้ายคลึงกัน

ดีซีเอสประกอบด้วย 2 ส่วนใหญ่ๆคือ

1. เครื่องมือสำหรับควบคุมกระบวนการผลิตติดตั้งในบริเวณพื้นที่ผลิต
2. เครื่องมือสำหรับติดต่อระหว่างดีซีเอสกับพนักงานติดตั้งในบริเวณห้องควบคุมแทนแผงควบคุมเดิม



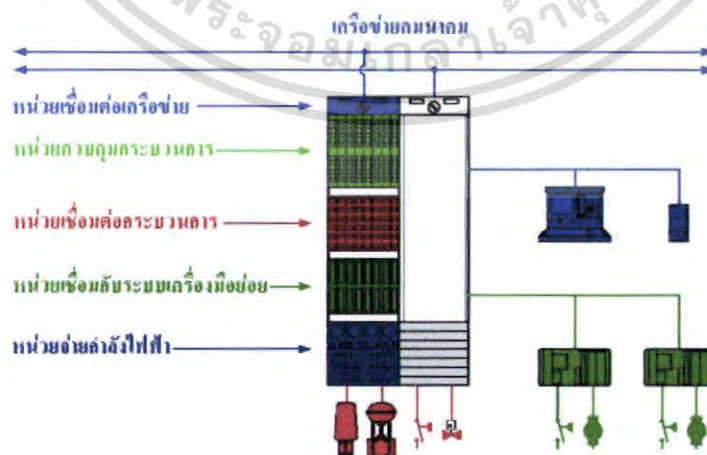
รูปที่ 2.3 การติดตั้งระบบควบคุมแบบกระจาย

### 2.1.3.1 เครื่องมือสำหรับควบคุมกระบวนการผลิตติดตั้งในบริเวณพื้นที่ผลิต

ทำหน้าที่รับสัญญาณวัดจากกระบวนการผลิต คำนวณค่าสัญญาณควบคุมและส่งสัญญาณควบคุมกลับไปยังกระบวนการผลิต ภายในตัวมันจะประกอบด้วยส่วนประกอบย่อยๆ ดังนี้

1. หน่วยเชื่อมต่อกระบวนการ
2. หน่วยควบคุมกระบวนการ
3. หน่วยเชื่อมต่อเครือข่าย
4. หน่วยจ่ายกำลังไฟฟ้า
5. หน่วยเชื่อมต่อกับระบบเครื่องมือย่อย

โดยมีรายละเอียดต่างๆ ดังนี้



รูปที่ 2.4 โครงสร้างของเครื่องมือสำหรับควบคุมกระบวนการผลิตติดตั้งในบริเวณพื้นที่ผลิต

1. หน่วยเชื่อมต่อกระบวนการ เป็นอุปกรณ์ติดต่อระหว่างคิซีเอสกับกระบวนการผลิต ทำหน้าที่รับสัญญาณวัดจากกระบวนการผลิตให้กับคิซีเอส และส่งสัญญาณควบคุมจากคิซีเอสไปยังกระบวนการผลิต ภายในหน่วยรับและส่งสัญญาณประกอบด้วยอุปกรณ์ขยายสัญญาณ อุปกรณ์แปลงสัญญาณอนาล็อกเป็นดิจิทัล และอุปกรณ์แปลงสัญญาณดิจิทัลเป็นอนาล็อก หน่วยเชื่อมต่อกระบวนการจะรับสัญญาณอนาล็อกจากเครื่องมือวัดต่างๆ เช่นรับสัญญาณกระแสไฟฟ้า 4-20 มิลลิแอมป์ หรือแรงดันไฟฟ้า 1-5 โวลต์ แล้วแปลงสัญญาณที่รับมานี้เป็นสัญญาณดิจิทัลเพื่อส่งเข้าไปประมวลผลในหน่วยควบคุมกระบวนการ แล้วรับสัญญาณควบคุมจากหน่วยควบคุมกระบวนการที่เป็นสัญญาณดิจิทัลมาแปลงกลับเป็น อนาล็อกเพื่อส่งออกไปควบคุมอุปกรณ์อื่นๆ เช่น วาล์ว

2. หน่วยควบคุมกระบวนการ เป็นส่วนประกอบที่สำคัญที่สุดของคิซีเอสสำหรับการเชื่อมต่อและควบคุมกระบวนการผลิต ทำหน้าที่รับสัญญาณวัดและรับคำสั่งของผู้ใช้ คำนวณค่าของสัญญาณควบคุม และส่งสัญญาณควบคุมป้อนกลับ ไปยังกระบวนการผลิต โดยผ่านทางหน่วยเชื่อมต่อกระบวนการ หน่วยควบคุมกระบวนการเป็นอุปกรณ์สำหรับตรวจสอบ และควบคุมการผลิตของคิซีเอส การควบคุมกระบวนการการผลิตของคิซีเอส ประกอบด้วย การควบคุมแบบดิจิทัลโดยตรง และ การควบคุมแบบติดดับ (On/Off Control) คิซีเอสส่วนมากทำหน้าที่ควบคุมการผลิต แบบดิจิทัลโดยตรงมากกว่าควบคุมแบบติดดับ

2.1 การควบคุมแบบดิจิทัลโดยตรง คือ การรับสัญญาณวัดชนิดอนาล็อกจากกระบวนการผลิตเพื่อประมวลผล โดยวิธีคำนวณผลทางคณิตศาสตร์และส่งสัญญาณควบคุมอนาล็อกกลับไปยังกระบวนการผลิตเพื่อรักษาสภาพกระบวนการผลิตให้ตรงกับความต้องการตลอดเวลา โดย ทำหน้าที่รับสัญญาณวัดและรับคำสั่งของผู้ใช้ คำนวณค่าของสัญญาณควบคุม และส่งสัญญาณควบคุมป้อนกลับไปยังกระบวนการผลิตโดยผ่านทางหน่วยเชื่อมต่อกระบวนการเป็นอุปกรณ์ติดต่อระหว่างคิซีเอสกับกระบวนการผลิต ทำหน้าที่รับสัญญาณวัดจากกระบวนการผลิตให้กับคิซีเอส และส่งสัญญาณควบคุมจากคิซีเอสไปยังกระบวนการผลิต

2.2 การควบคุมแบบติดดับ คือ การรับสัญญาณชนิดดิจิทัล ของอุปกรณ์และสวิตช์ต่างๆ จากกระบวนการผลิตเพื่อประมวลผลโดยวิธีการปฏิบัติแบบตรรก และส่งสัญญาณควบคุมแบบดิจิทัล หรือสภาวะติด-ดับ เพื่อควบคุมอุปกรณ์ในบริเวณกระบวนการผลิต ซึ่งหน่วยควบคุมกระบวนการภายในประกอบด้วย หน่วยประมวลผล หน่วยความจำ วงจรเชื่อมต่อระหว่างหน่วยควบคุมกระบวนการกับหน่วยเชื่อมต่อเครือข่ายและหน่วยเชื่อมต่อกระบวนการ

หน่วยประมวลผล ทำหน้าที่รับข้อมูลจากหน่วยเชื่อมต่อกระบวนการและหน่วยเชื่อมต่อเครือข่ายมาประมวลผล เพื่อควบคุมกระบวนการแบบดิจิทัล โดยตรงหรือควบคุมกระบวนการแบบติดดับ และส่งข้อมูลจากหน่วยควบคุมกระบวนการ ไปยังหน่วยเชื่อมต่อกระบวนการและหน่วยเชื่อมต่อเครือข่าย

หน่วยความจำของหน่วยควบคุมกระบวนการประกอบด้วย

หน่วยความจำชนิดรอม ทำหน้าที่เก็บ โปรแกรมสำหรับควบคุมการปฏิบัติงานการ ตรวจสอบและควบคุมกระบวนการผลิตของดีซีเอส หน่วยความจำชนิดแรม ทำหน้าที่เก็บข้อมูล สำหรับควบคุมการผลิตและผลลัพธ์การปฏิบัติงานตรวจสอบควบคุมกระบวนการผลิตของดีซีเอส

การเชื่อมต่อระหว่างหน่วยควบคุมกระบวนการกับหน่วยเชื่อมต่อเครือข่ายและหน่วยเชื่อมต่อกระบวนการประกอบด้วย การเชื่อมต่อแบบขนาน (parallel interface) และการเชื่อมต่อแบบอนุกรม (serial interface) การเชื่อมต่อแบบขนานเป็นวิธีการติดต่อระหว่างอุปกรณ์ต่างๆของดีซีเอส โดย การรับและส่งข้อมูลพร้อมกันจำนวนหลายบิต เช่นการเชื่อมต่อแบบมาตรฐาน ไออีอีอี-488 (IEEE-488) ประกอบด้วย สายสัญญาณสำหรับส่งข้อมูลและตรวจสอบความถูกต้องของข้อมูล จำนวน 16 เส้น ส่วนการเชื่อมต่อแบบอนุกรมเป็นวิธีการติดต่อระหว่างอุปกรณ์ต่างๆ โดยการรับส่งข้อมูลที่ละ 1 บิต ดีซีเอสมักเลือกวิธีการเชื่อมต่ออนุกรมตามแบบมาตรฐานต่างเช่น มาตรฐานการเชื่อมต่อแบบ อาร์เอส-232 (RS-232) และอาร์เอส-422 (RS-422) สำหรับรายละเอียดของวิธีการเชื่อมต่อจะกล่าวถึงภายหลัง

3 หน่วยเชื่อมต่อเครือข่าย คือ อุปกรณ์สำหรับติดต่อและรับส่งข้อมูลระหว่างเครื่องมือต่างๆที่ทำหน้าที่เชื่อมต่อและควบคุมกระบวนการผลิตของดีซีเอสกับเครือข่ายคมนาคมของระบบควบคุม ในโรงงานอุตสาหกรรม เนื่องจากดีซีเอสประกอบด้วยอุปกรณ์ต่างๆเป็นจำนวนมากติดตั้งกระจายกันอยู่ทั่ว โรงงานอุตสาหกรรม อุปกรณ์ทั้งหมดจะสามารถทำงานร่วมกัน ได้ต้องอาศัยเครือข่ายคมนาคมที่จะทำหน้าที่เชื่อมโยงข้อมูลและคำสั่งของอุปกรณ์ต่างๆเป็นข้อมูลเดียวกัน เครือข่ายคมนาคมของดีซีเอสอาจเป็นเครือข่ายเส้นทางเดียวหรือวงแหวน แต่ต้องประกอบด้วยเครือข่ายหลักหนึ่งเครือข่ายและเครือข่ายสำรองเพิ่มขึ้นจากเครือข่ายปกติอีกหนึ่งเครือข่ายเสมอ ทำให้เครือข่ายคมนาคมของดีซีเอสเป็น โครงสร้างเครือข่ายแบบเส้นทางเดียวสองชั้นหรือเครือข่ายแบบวงแหวนสองชั้น

เครือข่ายแบบเส้นทางเดียว อาจประกอบด้วยเครื่องคอมพิวเตอร์หลักเพียงเครื่องเดียวและคอมพิวเตอร์รองจำนวนหลายเครื่อง แต่ระหว่างเครื่องคอมพิวเตอร์หลักและเครื่องคอมพิวเตอร์รอง จะเชื่อมโยง โดยใช้สายสัญญาณเครือข่ายเส้นเดียวกัน เครื่องคอมพิวเตอร์หลักจึงสามารถรับข้อมูลจากเครื่องคอมพิวเตอร์รองได้ที่ละหนึ่งเครื่อง แต่สามารถส่งควบคุมไปยังเครื่องคอมพิวเตอร์รองพร้อมกันได้หลายเครื่อง เครือข่ายแบบเส้นทางเดียวใน โรงงานอุตสาหกรรมจะติดตั้งเครื่องคอมพิวเตอร์หลักไว้ในห้องควบคุมกลาง และเชื่อมโยงสายสัญญาณ ไปยังเครื่องคอมพิวเตอร์รองทั้ง โรงงาน โดยใช้สายสัญญาณเส้นเดียวกัน แต่การชำรุดหรือขาดของสายสัญญาณของเพียงจุดเดียว อาจทำให้การติดต่อระหว่างเครื่องคอมพิวเตอร์ในระบบทั้งหมดต้องหยุดชะงักลงทันที

เครือข่ายแบบวงแหวน ได้พัฒนาโดยการปรับปรุงจากเครือข่ายเส้นทางเดียวเพื่อลดเซชจุดบกพร่องของเครือข่ายเส้นทางเดียว เมื่อสายสัญญาณเครือข่ายชำรุดหรือขาด โดยเชื่อมโยงปลายสายทั้งสองด้านของสายสัญญาณเครือข่ายบรรจบกันเป็นวง ทำให้การติดต่อระหว่างเครื่องคอมพิวเตอร์ในระบบสามารถทำงานได้ตามปกติ

4. หน่วยจ่ายกำลังไฟฟ้า เป็นอุปกรณ์จ่ายกำลังไฟฟ้าให้กับอุปกรณ์ทุกส่วนของดีซีเอส อุปกรณ์จ่ายกำลังไฟฟ้า ทำหน้าที่กำจัดสัญญาณรบกวนและปรับแรงดันให้เหมาะสมกับอุปกรณ์ต่างๆของดีซีเอส หน่วยจ่ายกำลังไฟฟ้าของดีซีเอสทั่วไปสามารถจ่ายแรงดันไฟฟ้ากระแสตรง 5 โวลต์ และ 12 โวลต์ โดยใช้อุปกรณ์รักษาระดับแรงดันไฟฟ้า ทำหน้าที่รักษาแรงดันไฟฟ้าให้คงที่สำหรับเป็นแหล่งจ่ายให้เครื่องมือต่างๆของดีซีเอสทุกชุดแยกอิสระกันทั้งหมดเพื่อป้องกันการรบกวนจากการปฏิบัติงานของเครื่องมือใกล้เคียง

5. หน่วยเชื่อมต่อกับระบบเครื่องมือย่อย เป็นอุปกรณ์ที่เชื่อมต่อระหว่างดีซีเอสกับเครื่องมืออื่นในระบบควบคุมภายนอกดีซีเอส เช่น พีแอลซี (PLC) และเครื่องมืออื่นๆในระบบควบคุมที่มีหน่วยประมวลผลของตนเอง ดีซีเอสสามารถติดต่อกับเครื่องมืออื่นในระบบควบคุมได้โดยผ่านหน่วยเชื่อมต่อนี้

### 2.1.3.2 เครื่องมือสำหรับติดต่อระหว่างดีซีเอสกับพนักงานติดตั้งในบริเวณห้องควบคุม แทนแผงควบคุมเดิม

ทำหน้าที่เป็นหน่วยติดต่อและปฏิบัติงานของพนักงาน คือเครื่องมือสำหรับทำหน้าที่เป็นหน่วยติดต่อระหว่างดีซีเอสกับผู้ใช้ เพื่อให้ผู้ใช้สามารถตรวจสอบสภาพการปฏิบัติงานของระบบควบคุมและกระบวนการผลิต และป้อนคำสั่งหรือข้อมูลสำหรับให้เครื่องมือและระบบควบคุมนำไปปฏิบัติเพื่อควบคุมและดูแลกระบวนการผลิต ผู้ใช้ที่เกี่ยวข้องและปฏิบัติงานร่วมกับดีซีเอสโดยตรง คือผู้ใช้ระดับวิศวกรและผู้ใช้ระดับพนักงาน ผู้ใช้ระดับวิศวกรของดีซีเอส ทำหน้าที่ออกแบบระบบควบคุม ดูแลการติดตั้งเครื่องมือวัดและเครื่องมือควบคุมในโรงงานอุตสาหกรรม และรับผิดชอบในการดูแลรักษาเครื่องมือต่างๆให้เรียบร้อยตลอดเวลา ผู้ใช้ระดับพนักงาน คือพนักงานควบคุมและปฏิบัติการ ทำหน้าที่ตรวจสอบและควบคุมการผลิตให้เรียบร้อยตลอดเวลา

#### 2.1.3.2.1 จอภาพของดีซีเอส

จอภาพของดีซีเอสสามารถแสดงภาพจำลองกระบวนการผลิตและเครื่องมือต่างๆ ในระบบควบคุมเช่น กระดาษบันทึกสัญญาณ หน้าปัดของเครื่องวัดและเครื่องควบคุม ทำให้พนักงานสามารถตรวจสอบข้อมูลจากอุปกรณ์ต่างๆ บนจอภาพของดีซีเอส เช่นเดียวกับที่การตรวจสอบข้อมูลบนแผงควบคุมของระบบควบคุมเดิม สำหรับการควบคุมการผลิตพนักงานสามารถสั่งงานได้

โดยใช้เป็นพิมพ์เลือกจอแสดงภาพกระบวนการผลิตและเครื่องมือที่ต้องการ และสั่งงานบนจอภาพ แทนการปรับเครื่องควบคุมแบบเดิม

จอภาพของดีซีเอสอาจจะมีหลายจอภาพด้วยกันเพื่อให้พนักงานที่ใช้งาน ได้สามารถ ตรวจสอบกระบวนการผลิตและเครื่องมือต่างๆ ในระบบควบคุมจอภาพของ ดีซีเอสดังนี้

1. ภาพกระบวนการผลิตรวม คือจอภาพแสดงภาพรวมของกระบวนการผลิต จะแสดง ข้อมูลการข้อมูลการปฏิบัติงานของเครื่องควบคุมให้กับพนักงานทราบ โดยแสดงเพียง ข้อมูลความแตกต่างระหว่างเป้าหมายควบคุมกับสภาพกระบวนการ
2. กลุ่มกระบวนการ คือจอภาพแสดงข้อมูลของเครื่องมือในกระบวนการผลิตที่ใกล้เคียง กัน หรือเครื่องมือที่มีความสัมพันธ์กันทางใดทางหนึ่ง ภาพกลุ่มกระบวนการแสดง ข้อมูลในกระบวนการผลิตเช่นเดียวกับแผงหน้าปัดของเครื่องมือวัดที่ติดตั้งเรียงเป็นแถว ในห้องควบคุม
3. หน่วยเครื่องมือ คือจอภาพแสดงข้อมูลของเครื่องมือวัดหรือเครื่องมือควบคุมใน กระบวนการผลิตของดีซีเอส ข้อมูลของหน่วยเครื่องมือประกอบด้วย ชื่อและรหัสของ เครื่องมือ รายละเอียดของเครื่องมือ หน้าที่ คุณสมบัติ หน่วยทางวิศวกรรม ระดับช่วงการ วัดสูงสุดและต่ำสุดของเครื่องมือ
4. ภาพแนวโน้มกระบวนการ คือ จอภาพแสดงเส้นกราฟของข้อมูลจากตัวแปรต่างๆ ใน กระบวนการผลิต เช่นเดียวกับ กระดาษบันทึกของเครื่องบันทึกสัญญาณ เพื่อแสดง แนวโน้มของกระบวนการ
5. ภาพจำลองกระบวนการ คือ จอภาพแสดงกระบวนการผลิตบนจอภาพของดีซีเอส การ จำลองภาพกระบวนการของดีซีเอส ประกอบด้วย รูปภาพ สัญลักษณ์ ตัวอักษร และตัวเลข ภาพจำลองกระบวนการแบ่งเป็น 2 ส่วนคือ ภาพนิ่ง และภาพเคลื่อนไหว ภาพนิ่งคือภาพ จำลองเครื่องมือ เครื่องจักร ส่วนที่ไม่ต้องการให้เปลี่ยนแปลงตามสภาพของกระบวนการ ผลิต ภาพเคลื่อนไหว คือภาพที่สามารถเปลี่ยนแปลงตามสภาพของกระบวนการผลิตให้ เห็นบนจอภาพ เช่น ระดับของเหลว สีของเครื่องมือและอุปกรณ์ ตัวเลขและข้อมูลใน กระบวนการผลิต
6. สัญญาณเตือนเหตุการณ์ คือ จอภาพแสดงเหตุการณ์ผิดปกติ หรือเหตุการณ์สำคัญ เช่น การเตือนเมื่อตัวแปรในกระบวนการผลิตมีค่าสูงหรือต่ำเกินไปหรือเปลี่ยนแปลงเร็วกว่า ปกติ หรือหน้าสัมผัสของอุปกรณ์วงจรเปิด-ปิดของวงจร ไฟฟ้า สัญญาณเตือนของดีซีเอส จะประกอบด้วยสัญญาณเสียงและข้อความบนจอภาพ

การใช้ดีซีเอสใน โรงงานอุตสาหกรรม ควรมีหน้าจอแสดงผลอย่างน้อย 3 ส่วนหลักๆคือ ภาพจำลองกระบวนการ จอภาพแสดงกระบวนการผลิตรวม จอภาพแสดงสัญญาณเตือน การตั้ง

หน้าจอดีซีเอสควมรหน้าจอแสดงผลสามหน้าจอหลักนี้ แต่ถ้ากระบวนการผลิตมีขนาดใหญ่อาจมีหน้าจอที่ใหญ่กว่านี้ได้

หน่วยเก็บข้อมูลและประวัติกระบวนการ จะอยู่ในหน่วยติดต่อและปฏิบัติการของพนักงาน เป็นอุปกรณ์สำหรับเก็บข้อมูลเพื่อเป็นข้อมูลสำหรับตรวจสอบและสรุปผลของการปฏิบัติงานของพนักงานและระบบควบคุม เพื่อใช้เป็นข้อมูลสำหรับปรับปรุงแก้ไขระบบในภายหลัง

## 2.2 อุปกรณ์ที่ใช้วัดและควบคุมกระบวนการ

อุปกรณ์ที่ใช้วัดและควบคุมกระบวนการ แบ่งเป็น 2 แบบคือ

1. อุปกรณ์ประเภทเครื่องมือวัดและควบคุม
2. อุปกรณ์ต่างๆที่ใช้ในการทำงานของระบบ

### 2.2.1 อุปกรณ์ประเภทเครื่องมือวัดและควบคุม

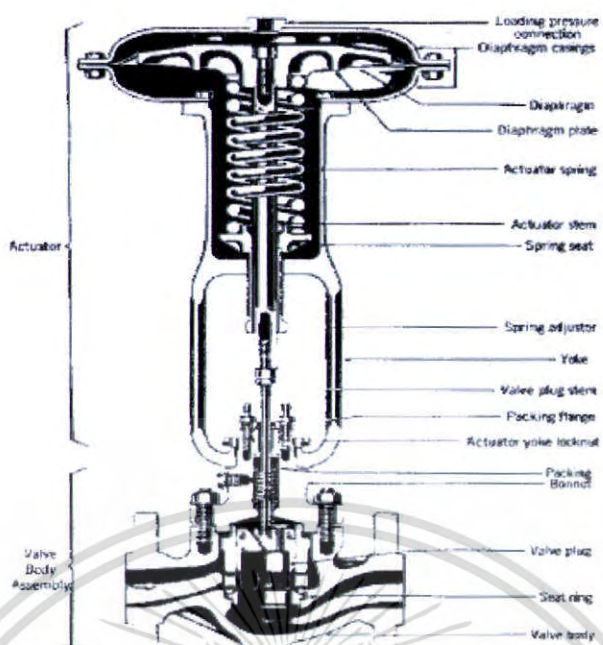
#### 2.2.1.1 วาล์วควบคุม

เป็นอุปกรณ์ควบคุมสุดท้าย (Final Control Element) ที่เชื่อมต่อระหว่างตัวควบคุม กับกระบวนการ และยังเป็นตัวชดเชยกระบวนการได้ด้วย วาล์วควบคุมประกอบด้วย 2 ส่วนคือ

1. ตัวขับเคลื่อนวาล์วควบคุม ประกอบด้วย ไดอะแฟรมเคส(Diaphragm Case) ไดอะแฟรมเพลท (Diaphragm Plate) และสปริง (spring)

2. ตัววาล์ว ประกอบด้วยซีท (Seat) และปลั๊ก (Plug) ทำหน้าที่บังคับพื้นที่ให้ของเหลวไหลมากหรือน้อยตามการเปิดของวาล์ว โดยตัวขับเคลื่อนวาล์วจะควบคุมการขึ้นลงของปลั๊กให้แน่นที่ซีทตามสัญญาณควบคุม เป็นการบังคับพื้นที่ให้ของเหลวไหลผ่านเพื่อควบคุมการไหล

ทั้งตัวขับเคลื่อนและตัววาล์วสามารถออกแบบให้ทำงานแบบทางตรง(Direct-Normally Open) และแบบกลับทาง (Reverse-Normally Open)



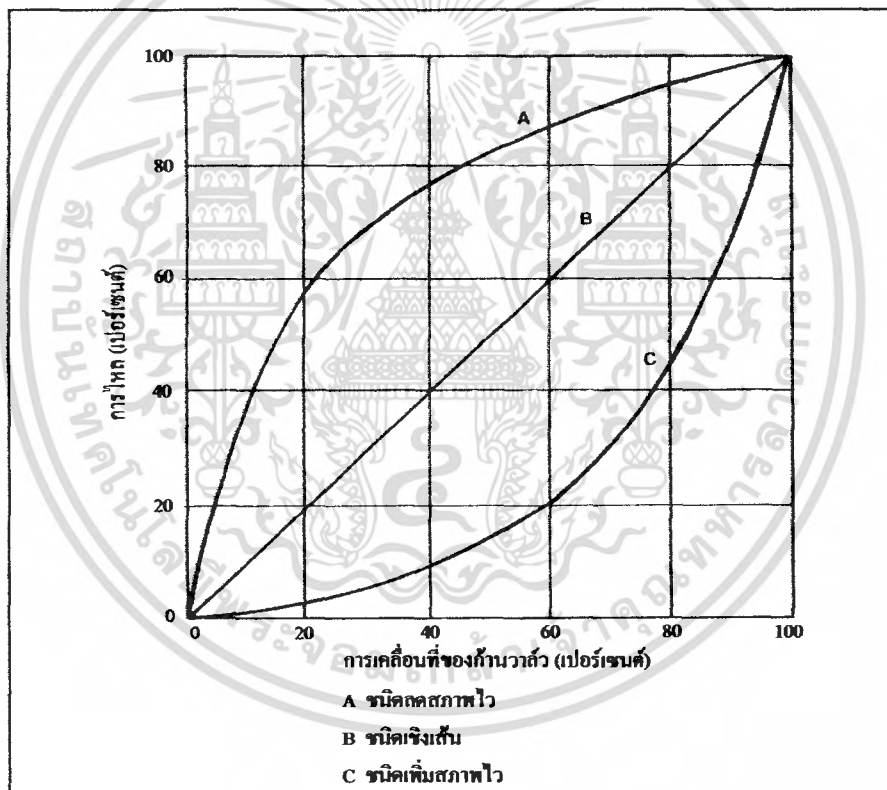
รูปที่ 2.5 โครงสร้างของวาล์วควบคุม



รูปที่ 2.6 วาล์วควบคุม

สมบัติการไหลเฉพาะ (Flow characteristics) ของวาล์วแบบก้านวาล์วเคลื่อนที่เชิงเส้นมีสามประเภท ขึ้นกับความสัมพันธ์ระหว่างตำแหน่งวาล์ว, การเคลื่อน, จังหวะ หรือการเปิดวาล์ว เทียบกับผลการไหล ประเภทของวาล์วที่แสดงในรูปที่ 2.7 มีดังนี้

1. ควิก โอเพนิง (Quick opening) วาล์วประเภทนี้มีสภาพไว (การเปลี่ยนแปลงการไหลต่อการเปลี่ยนแปลงของตำแหน่งวาล์วที่กำหนด) ลดลงเมื่อเพิ่มการไหล
2. เชิงเส้น (Linear) วาล์วประเภทนี้มีสภาพไวมากหรือน้อยคงที่ตลอดช่วงการไหล
3. อีควอลเปอร์เซ็นต์เทจ (Equal percentage) วาล์วประเภทนี้กำเนิดมาจากว่าสภาพไว ณ อัตราไหลใด ๆ ที่กำหนดจะเป็นค่าเปอร์เซ็นต์ของค่าของอัตราไหลที่กำหนดให้ หรือเมื่อจำนวนเปอร์เซ็นต์ของการเปิดวาล์วเพิ่มขึ้นเท่าๆ กันทุกครั้ง ทำให้การไหลแต่ละครั้งเพิ่มขึ้นเท่ากับ สองเท่าโดยประมาณของการไหลก่อนมีการเพิ่มการเปิดวาล์ว



รูปที่ 2.7 สมบัติการไหลเฉพาะของวาล์ว

### 2.2.1.2 เครื่องวัดความดันแบบดิฟเฟอเรนเชียล

ความดันดิฟเฟอเรนเชียลการวัดความดันโดยอาศัยหลักการความดันต่าง ที่มีประยุกต์ใช้งานอย่างกว้างขวางในอุตสาหกรรม เช่นการวัดอัตราการไหล วัดระดับของเหลว วัดความหนืดหรือวัดค่าความหนาแน่นของของเหลว ในปัจจุบันอุปกรณ์สำหรับวัดความดันดิฟเฟอเรนเชียลได้รับการ

ออกแบบให้อยู่ในแคปซูลเดียวกันมีชื่อเรียก ย่อๆว่า ดีพีทรานสมิเตอร์ ในส่วนของทรานสมิเตอร์ ส่วนใหญ่เป็นแบบอิเล็กทรอนิกส์ ให้สัญญาณเอาต์พุต เป็นกระแสหรือแรงดันไฟฟ้า ซึ่งมีข้อดีคือ สามารถส่งข้อมูลนี้ไปยังเครื่องบันทึก เครื่องควบคุม หรือเครื่องขับเคลื่อนได้หลายอย่างพร้อมกันได้ ในระยะไกล ซึ่งในปัจจุบันได้มีการพัฒนาให้เหมาะสมและสะดวกในการใช้งานมากขึ้น เช่น ทนต่อสภาพแวดล้อม, การเปลี่ยนแปลงอุณหภูมิ และความชื้นได้ดี ให้ผลการทำงานที่ดีมีความเที่ยงตรงสูง โดยมาตรฐานของสัญญาณเอาต์พุตของทรานสมิเตอร์จะถูกกำหนดให้อยู่ในมาตรฐานเดียวกันคือ

- เอาต์พุต แบบนิวเมติกส์ 3-15 ปอนด์ต่อตารางนิ้ว (psi)
- เอาต์พุต แบบกระแสไฟฟ้า 4-20 มิลลิแอมแปร์
- เอาต์พุต แบบแรงดันไฟฟ้า 1-5 โวลท์



รูปที่ 2.8 เครื่องวัดความดันแบบดิฟเฟอเรนเชียล

### 2.2.1.3 ชุดปรับปรุงคุณภาพลมอัด

เนื่องจากวาล์วควบคุมนั้นต้องใช้ลมในการควบคุมการเปิด-ปิด ลมจึงเป็นสิ่งที่สำคัญในการควบคุม เพราะฉะนั้นลมที่ใช้ต้องมีความสะอาดเพื่อจะยืดอายุการทำงานของวาล์วควบคุม โดยชุดปรับปรุงคุณภาพลมอัดประกอบด้วย 3 ส่วนคือ ฟิวเจอร์หรือตัวกรอง(Filter), อุปกรณ์ควบคุมความดันในระบบลม (regulator), อุปกรณ์ที่ช่วยในการหล่อลื่น ภายในระบบลม (lubricator)

1.ฟิวเจอร์หรือตัวกรอง ทำหน้าที่กรองเศษผง และน้ำ

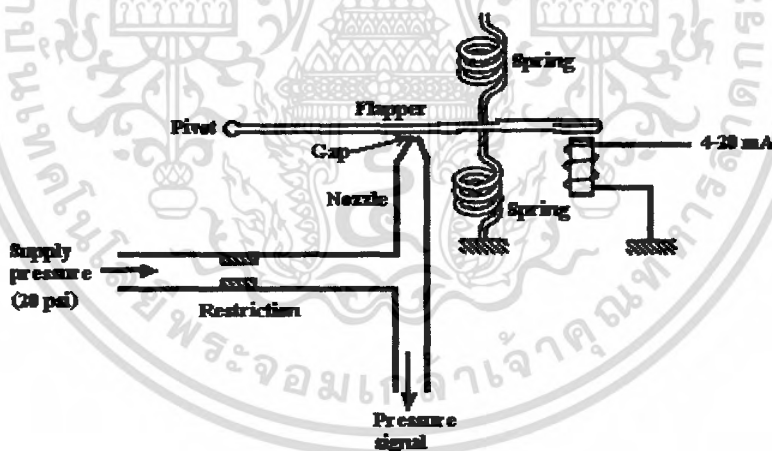
## สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2. อุปกรณ์ควบคุมความดันในระบบทำหน้าที่ปรับความดันให้ค่าของความดันเท่ากับค่าความดันที่ใช้งาน เพื่อป้องกันความผิดพลาดในการควบคุม และรักษาอายุการใช้งานของวาล์วควบคุม
3. อุปกรณ์ที่ช่วยในการหล่อลื่น ภายในระบบ ทำหน้าที่ช่วยในการป้องกันการสึกหรอโดยใช้การผสมน้ำมันหล่อลื่นเข้าไปในลมอัด

### 2.2.1.4 อุปกรณ์แปลงกระแสไฟฟ้าเป็นความดันลม (Current to Pressure Converter)

เป็นอุปกรณ์ที่ทำหน้าที่แปลงกระแสไฟฟ้าเป็นความดันลม โดยมีหลักการทำงานดังรูปที่ 2.9 เมื่อเราป้อนกระแสไฟฟ้าขนาด 4-20 มิลลิแอมแปร์ ให้แก่ขดลวดทำให้เกิดสนามแม่เหล็กดูดปลายข้างหนึ่งของแฟลปเปอร์ (Flapper) ทำให้ช่องว่าง (Gap) ระหว่างแฟลปเปอร์และนอซเซิล (Nozzle) แคบลงอากาศที่รั่วไหลออกมาทางปลายของนอซเซิลก็จะลดลง สัญญาณลมที่ได้ออกมา ก็จะลดลงไปด้วย โดยที่กระแสไฟฟ้ามีขนาด 4 มิลลิแอมแปร์จะได้สัญญาณลมขนาด 3 ปอนด์ต่อตารางนิ้ว และที่กระแสไฟฟ้ามีขนาด 20 มิลลิแอมแปร์ จะได้สัญญาณลมขนาด 15 ปอนด์ต่อตารางนิ้ว

โดยในระบบอุปกรณ์แปลงกระแสไฟฟ้าเป็นความดันลมทำหน้าที่เปลี่ยนสัญญาณไฟฟ้าจากไมโครคอนโทรลเลอร์ (Microcontroller) ให้กลายเป็นสัญญาณลมเพื่อใช้ควบคุมวาล์วควบคุม



รูปที่ 2.9 ส่วนประกอบพื้นฐานของอุปกรณ์แปลงกระแสไฟฟ้าเป็นความดันลม

### 2.2.1.5 เอสแอลซีดี อินดิเคติง คอนโทรลเลอร์ (SLCD Indicating Controller)

ตัวควบคุมทำหน้าที่เป็นอุปกรณ์ในการตัดสินใจให้กับระบบควบคุม โดยการเปรียบเทียบสัญญาณของกระบวนการที่ได้จากอุปกรณ์วัด และส่งสัญญาณซึ่งก็คือตัวแปรที่ถูกควบคุมกับค่าปรับตั้ง จากนั้นก็จะส่งสัญญาณที่เหมาะสมให้กับวาล์วควบคุมหรืออุปกรณ์ควบคุมขั้นสุดท้ายแบบอื่น เพื่อที่รักษาไว้ซึ่งค่าของตัวแปรที่ถูกควบคุมกับจุดเป้าหมาย

ในระบบนี้ใช้เอสแอลซีดี อินดิเคตติ้ง คอนโทรลเลอร์ เป็นอุปกรณ์ควบคุมซึ่งใช้ไมโครโปรเซสเซอร์เป็นตัวควบคุมการทำงานของเครื่อง โดยมีคุณสมบัติมาตรฐานดังนี้

สัญญาณเข้า/ออก (Input/Output Signal)

สัญญาณเข้าแบบอนาล็อก (Analog Input) : ระดับ 1 ถึง 5 โวลต์กระแสตรง จำนวน 4 จุด

สัญญาณออกแบบอนาล็อก (Analog Output) : ระดับ 1 ถึง 5 โวลต์กระแสตรง จำนวน 2 จุด

: ระดับ 4 ถึง 20 มิลลิแอมแปร์ จำนวน 1 จุด

สถานะแสดงสัญญาณเข้า (Status Input) : เป็นหน้าสัมผัสหรือสัญญาณแรงเคลื่อนไฟฟ้า 1 จุด

สถานะของสัญญาณออก (Status Output) : เป็น ทรานซิสเตอร์ คอนแทค (Transistor Contact)

จำนวน 5 จุด

สัญญาณผิดปกติ (Fail Output) : เป็น ทรานซิสเตอร์ คอนแทค 1 จุดเมื่อมีการผิดปกติของแหล่งจ่ายพลังงาน หรืออื่นๆ

สัญญาณควบคุมของเครื่องควบคุมมี 2 แบบคือ

- สัญญาณควบคุมแบบกลับ (Reverse action) คือ ถ้าตัวควบคุมรับสัญญาณวัด (Manipulate Signal) เข้ามามากขึ้น เป็นผลให้สัญญาณควบคุมลดลง

- สัญญาณควบคุมแบบตรง (Direct action) คือ ถ้าตัวควบคุมรับสัญญาณวัดเข้ามามากขึ้น เป็นผลให้สัญญาณควบคุมเพิ่มขึ้น



รูปที่ 2.10 เอสแอลซีดี อินดิเคตติ้ง คอนโทรลเลอร์

### 2.2.1.6 แมกเนติก โฟลว์ มิเตอร์ (Magnetic Flow Meter)

เป็นเครื่องมือวัดการไหล โดยอาศัยหลักการ หาค่าความเร็วเฉลี่ยในการไหลของของไหล โดยใช้กฎของฟาราเดย์ ซึ่งมีใจความว่า “เมื่อของเหลวที่นำไฟฟ้าไหลผ่านสนามแม่เหล็ก จะเกิดแรงเคลื่อนไฟฟ้าขึ้นในแนวตั้งฉากกับทิศทางการไหลของสนามแม่เหล็ก”

หลักการสร้างแม่เหล็ก มี 2 วิธีคือ

1. การใช้แม่เหล็กถาวร (DC Excitation)

2. ใช้ไฟกระแสสลับ (AC) 50 เฮิร์ตซ์ หรือ พัลส์ (Pulse) ที่ความถี่ต่างๆ

ข้อมูลทั่วไปของแมกเนติกโฟลว์ มิเตอร์

- ไม่เกิดแรงดันสูญเสีย เนื่องจากท่อของเครื่องมือวัดที่มีขนาดเท่ากับขนาดท่อปกติ
- สามารถวัดของไหล ที่เป็นสารกึ่งตัวนำได้
- ไม่จำเป็นต้องมีระยะท่อตรงก่อนและหลังเครื่องวัด
- สามารถทำการวัดการไหลได้สองทาง โดยกลับขั้ว อิเล็กโทรด (Electrode) เท่านั้น
- วัดได้โดยไม่ขึ้นอยู่กับค่าความหนืด ค่าความหนาแน่น อุณหภูมิ ความดัน สภาพการไหล
- ความแม่นยำ (Accuracy) จาก  $\pm 0.5 - 2$  เปอร์เซ็นต์ ของค่าเต็มสเกล
- เรนจอะบิลิตี้ (Rangeability) 20:1 แบบพิเศษอาจถึง 30:1

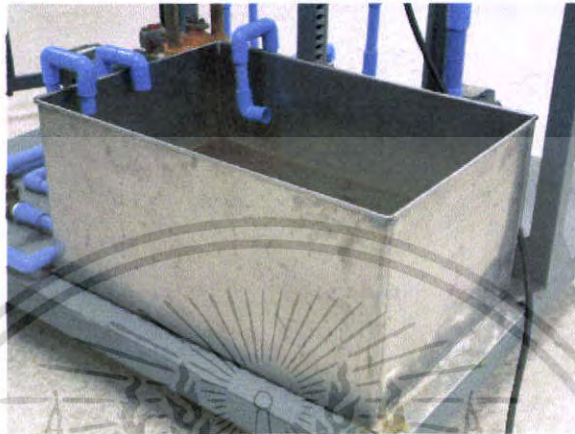


รูปที่ 2.11 แมกเนติก โฟลว์ มิเตอร์

## 2.2.2 อุปกรณ์ต่างๆที่ใช้ในการทำงานของระบบ

### 2.2.2.1 ถังพักน้ำ

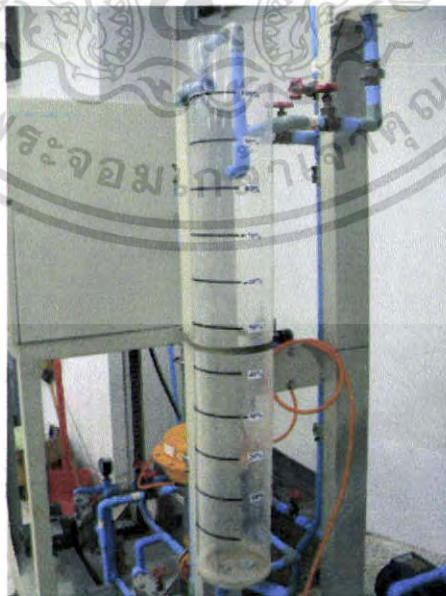
เป็นอุปกรณ์สำหรับเก็บน้ำเพื่อใช้สำรองในระบบควบคุม จะมีปริมาตรสำรองไว้เพียงพอต่อปริมาตรสูงสุดและต่ำสุดของกระบวนการ



รูปที่ 2.12 ถังพักน้ำ

### 2.2.2.2 ถังวัดระดับน้ำ

เป็นอุปกรณ์สำหรับเก็บน้ำเพื่อใช้ในระบบควบคุม จะมีมาตรวัดระดับบอกทุก 10 เซนติเมตร เพื่อใช้ในการควบคุม



รูปที่ 2.13 ถังวัดระดับน้ำ

### 2.2.2.3 ปั้มน้ำ

เป็นอุปกรณ์เพิ่มความดันให้แก่ระบบ



รูปที่ 2.14 ปั้มน้ำ

### 2.2.2.4 ตัวจ่ายลม (Air supply)

เป็นอุปกรณ์ที่ทำหน้าที่เป็นตัวผลิตลมที่มีแรงดันเพื่อใช้ในการควบคุมวาล์วควบคุม โดยจะจ่ายลมที่มีแรงดันคงที่ให้กับ ตัวแปลงกระแสไฟฟ้าเป็นสัญญาณลม เพื่อปรับความดันลมให้เหมาะสมกับสัญญาณควบคุม



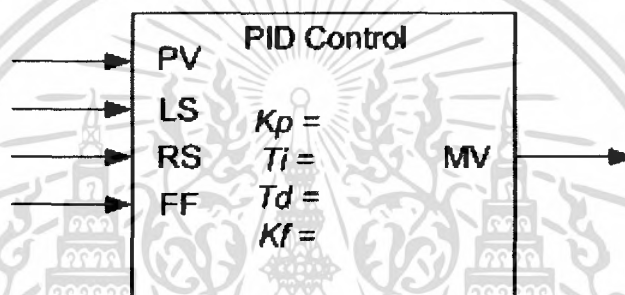
รูปที่ 2.15 ตัวจ่ายลม (Air supply)

## 2.3 ทฤษฎีและหลักการควบคุมของพีไอดี (PID)

### 2.3.1 หน่วยควบคุมชนิดพีไอดี

หน่วยควบคุมชนิดพีไอดี เป็นหน่วยเครื่องมือภายในของดีซีเอส สำหรับควบคุมกระบวนการแทนเครื่องควบคุมแบบอิเล็กทรอนิกส์หรือนิวแมติกส์ชนิดพีไอดี ความสัมพันธ์ระหว่างสัญญาณออกของหน่วยควบคุมชนิดพีไอดี สามารถแสดงโดยสมการ

$$MV = K_p \left[ (SV - PV) + \frac{1}{T_i} \int (SV - PV) \times dt + T_d \frac{d}{dt} (SV - PV) \right] + K_f \times FF$$



รูปที่ 2.16 หน่วยควบคุมแบบพีไอดี

โดยที่

MV คือ สัญญาณควบคุมของหน่วยควบคุมชนิดพีไอดี

PV คือ ตัวแปรกระบวนการของหน่วยควบคุมชนิดพีไอดี

SV คือ เป้าหมายการควบคุมของหน่วยควบคุมชนิดพีไอดี

LS คือ เป้าหมายการควบคุม (SV) ของหน่วยควบคุมชนิดพีไอดี เมื่อแบบการปฏิบัติหน้าที่ของหน่วยควบคุมชนิดพีไอดี เป็นการควบคุมแบบอัตโนมัติ เป็นเป้าหมายการควบคุมปกติ (local set point) ของหน่วยควบคุมชนิดพีไอดี

RS คือ เป้าหมายการควบคุม (SV) ของหน่วยควบคุมชนิดพีไอดี เมื่อแบบการปฏิบัติหน้าที่ของหน่วยควบคุมชนิดพีไอดี เป็นแบบการควบคุมระยะไกล หรือแบบการควบคุมแบบวงอันดับ (Cascade Mode) เป็นเป้าหมายการควบคุมระยะไกล (remote set point) ของหน่วยควบคุมชนิดพีไอดี

FF คือ สัญญาณเข้าสำหรับการควบคุมแบบป้อนไปหน้า (feed forward control)

$K_p$  คือ อัตราการควบคุมแบบโพรพอร์ชันแนล (Proportional) ของหน่วยควบคุมชนิดพีไอดี

$T_i$  คือ ช่วงเวลาอินทิเกรต ของหน่วยควบคุมชนิดพีไอดี

$T_d$  คือ ช่วงเวลาอนุพันธ์ ของหน่วยควบคุมชนิดพีไอดี

$K_f$  คือ อัตราการควบคุมแบบป้อนไปหน้า ของหน่วยควบคุมชนิดพีไอดี

### 2.3.2 แบบการควบคุมของหน่วยควบคุมชนิดพีไอดี

แบ่งเป็น 3 แบบดังนี้

MAN แบบการควบคุมด้วยมือ สัญญาควบคุม (MV) ของหน่วยควบคุมชนิด พีไอดี อนุญาตให้ผู้ใช้เป็นผู้กำหนดโดยตรงจากหน่วยติดต่อและปฏิบัติการของผู้ใช้ของดีซีเอส

AUT ควบคุมอัตโนมัติ สัญญาควบคุม (MV) ของหน่วยควบคุมชนิดพีไอดี ได้จากการประมวลผลเพื่อกำหนดสัญญาควบคุมตามสมการของเครื่องควบคุมแบบพีไอดี ภายในหน่วยควบคุมกระบวนการของดีซีเอส โดยหน่วยควบคุมชนิดพีไอดี รับเป้าหมายการควบคุม SV จากเป้าหมายการควบคุมปกติ (LS)

REM แบบการควบคุมระยะไกล (REM) หรือแบบการควบคุมแบบคาสเคด สัญญาควบคุม (MV) ของหน่วยควบคุมชนิดพีไอดี ได้จากการประมวลผลเช่นกันกับแบบการควบคุมอัตโนมัติ (AUT) โดยหน่วยควบคุมชนิดพีไอดี รับเป้าหมายการควบคุม SV จากเป้าหมายการควบคุมระยะไกล แทนเป้าหมายการควบคุมปกติ

### 2.3.3 วัตถุประสงค์ของการติดตั้งหน่วยควบคุมชนิดพีไอดี

#### 2.3.3.1 การควบคุมแบบป้อนกลับ

การควบคุมแบบป้อนกลับเป็นการควบคุมแบบเป็นพื้นฐานของการควบคุม เพราะเนื่องจากมีความง่ายในการใช้งานและหน่วยควบคุมชนิดพีไอดี ก็ทำหน้าที่รักษาเสถียรภาพของระบบควบคุม

**ข้อดี**

ง่ายต่อการใช้งานเนื่องจากใช้เพียงตัวควบคุมและตัวรับรู้เพียงอย่างละ 1 ตัวก็สามารถทำงานได้

**ข้อเสีย**

ต้องมีความผิดพลาดเกิดขึ้นก่อนระบบควบคุมแบบป้อนกลับจึงจะสามารถแก้ไขความผิดพลาดได้

ซึ่งจากรูปที่ 2.17 การควบคุมแบบป้อนกลับเป็นการกำหนดโครงสร้างของระบบควบคุมแบบพื้นฐานกระบวนการผลิตประกอบด้วยสัญญาณวัดเดียวและสัญญาณควบคุมเดียว หน่วย



ปฏิบัติการแก้ไขเกิดขึ้น เช่นเดียวกันเมื่อมีปฏิบัติการแก้ไขแล้วต้องรอคอยผลการแก้ไขด้วย การควบคุมสาเหตุให้โอกาสเหมาะในการหาค่าตัวแปรถูกควบคุมที่อยู่ระหว่างกลางวงควบคุม และทำให้ปฏิบัติการแก้ไขต่อการรบกวนได้ฉับพลันกว่า

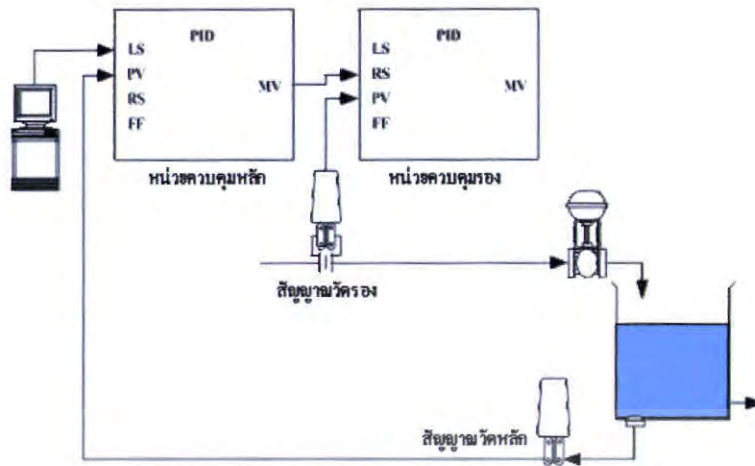
### ข้อดี

1. สิ่งรบกวน (Disturbance) ที่มีผลต่อวงควบคุมจะถูกวงควบคุมตรวจจับและแก้ไข ก่อนที่จะไปรบกวนที่วงปฐมภูมิ
2. คอนโทรลละบิลิตี (Controllability) ของวงควบคุมจะเพิ่มขึ้น เนื่องจากวงควบคุมมีความเร็วในการตอบสนอง ต่ออุปกรณ์ที่มีไดนามิกเพิ่มขึ้น ระหว่างวาล์วควบคุมและตัวแปรในทุติยภูมิ
3. ความไม่เป็นเชิงเส้นของวงควบคุมจะถูกจัดการและแก้ไขและกำจัดออกจากวงปฐมภูมิเพื่อให้วงปฐมภูมิมีความสัมพันธ์เชิงเส้น

### ข้อเสีย

มีความสิ้นเปลืองซึ่งต้องใช้ตัวควบคุมเพิ่มจากระบบควบคุมแบบป้อนกลับ

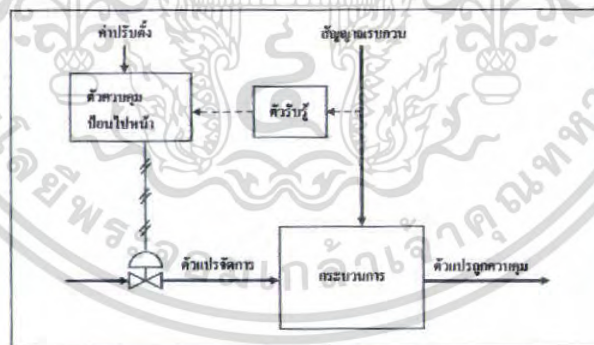
จากรูปที่ 2.19 การควบคุมแบบสาเหตุ เป็นการกำหนดโครงสร้างของระบบควบคุมสำหรับกระบวนการผลิตที่มีสัญญาณวัดหลายสัญญาณและสัญญาณควบคุมสัญญาณเดียว หน่วยควบคุมพีไอดีตัวที่ 1 เป็นหน่วยควบคุมของวงปฐมภูมิ ปฏิบัติงานควบคุมกระบวนการผลิตแบบควบคุมอัตโนมัติ (AUT) รับเป้าหมายการควบคุมปกติ (LS) และตรวจสอบสัญญาณวัดหลักของกระบวนการผลิต (PV) และส่งสัญญาณควบคุม (MV) เป้าหมายการควบคุมของหน่วยควบคุมชนิดพีไอดีที่ 2 หน่วยควบคุมชนิดพีไอดีที่ 2 เป็นหน่วยควบคุมของวงทุติยภูมิ ปฏิบัติงานควบคุมกระบวนการผลิตแบบการควบคุมระยะไกล (REM) รับเป้าหมายการควบคุมระยะไกล (RS) และตรวจสอบสัญญาณวัดรองของกระบวนการผลิต (PV) และส่งสัญญาณควบคุม (MV) ให้กับกระบวนการผลิตเพื่อรักษาเสถียรภาพของระบบควบคุม



รูปที่ 2.19 การควบคุมแบบคาสเคด

### 2.3.3.3 การควบคุมแบบป้อนไปหน้า (Feedforward Control)

การควบคุมป้อนไปหน้าจะใช้ตัวรับรู้วัดสัญญาณรบกวนที่เข้าไปในกระบวนการและส่งข้อมูลไปให้ตัวควบคุมป้อนไปหน้า ตัวควบคุมป้อนไปหน้าจะกำหนดการเปลี่ยนแปลงที่ต้องการใช้ในการสร้างตัวแปรจัดการเพื่อให้ผลของการรบกวนไปพร้อมกับผลของการเปลี่ยนตัวแปรจัดการแล้วไปทำให้ตัวแปรถูกควบคุมไม่เกิดการเปลี่ยนแปลง



รูปที่ 2.20 วงควบคุมป้อนไปหน้า

#### ข้อดี

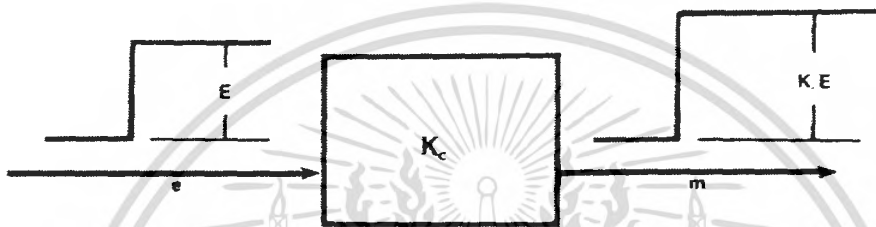
1. ไม่มีค่าผิดพลาดเนื่องจากตัวควบคุมแบบป้อนไปหน้าจะวัดสัญญาณรบกวนล่วงหน้าและแก้ไขก่อนที่จะมากระทบกับกระบวนการ



## 2.4 เครื่องควบคุมแบบพีไอดี

### 2.4.1 การควบคุมแบบโพรบโพชันแนล (Proportional)

การควบคุมแบบโพรบโพชันแนล เป็นโหมดควบคุมที่ง่ายที่สุด โดยที่เอาต์พุตของตัวควบคุมเป็นสัดส่วนทางพีชคณิตกับค่าสัญญาณผิดพลาดที่เป็นอินพุตของตัวควบคุม ดังแสดงในรูปข้างล่าง สัญญาณอินพุตเป็นค่าผิดพลาด,  $e$  ที่เป็นขั้นบันไดมีขนาดเท่ากับ  $E$  ในโหมดนี้ถ้าเอาต์พุตของตัวควบคุม  $m$  คำนวณได้จาก



รูปที่ 2.22 แผนภาพบล็อกตัวควบคุมแบบโพรบโพชันแนล

$$m(t) = \bar{m} + K_p e(t)$$

เมื่อ  $m(t)$  = ค่าเอาต์พุตของตัวควบคุม

$\bar{m}$  = ค่าไบแอส (Bias)

$K_p$  = อัตราการขยายของตัวควบคุม (Gain)

ในตัวควบคุมระดับอุตสาหกรรมการปรับค่าเกนไม่แสดงเป็นค่าเกน แต่จะแสดงเป็นค่าโพรบโพชันแนล แบนด์ (Proportional band) เนื่องจากค่าเกนแสดงเป็นจำนวนเท่าที่ไม่มีหน่วยยากแก่การคาดคะเน โพรบโพชันแนล แบนด์ แสดงเป็นเปอร์เซ็นต์ค่าผิดพลาดหรืออินพุตที่เกิดขึ้นที่สัมพันธ์กับการเปลี่ยนเอาต์พุตของตัวควบคุมไปหนึ่งร้อยเปอร์เซ็นต์ หรือเปลี่ยนไปค่าเต็มสเกล จึงมีหน่วยเป็นเปอร์เซ็นต์ โดยสามารถแสดงความสัมพันธ์กับ โพรบโพชันแนล เกน (Proportional gain) ดังนี้

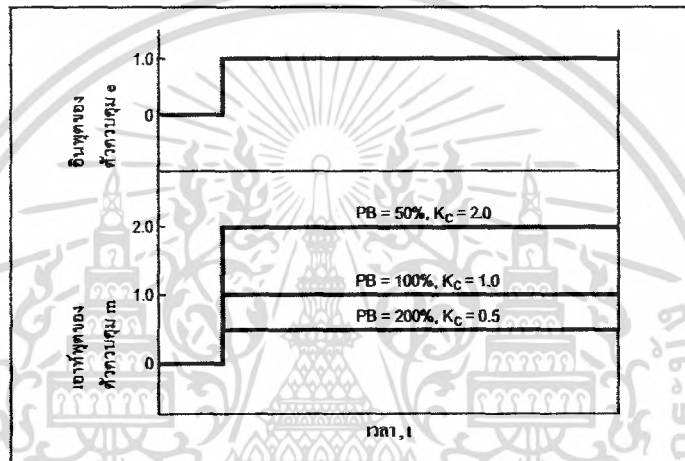
$$PB = \frac{1}{K_p} \times 100\%$$

### ข้อดี

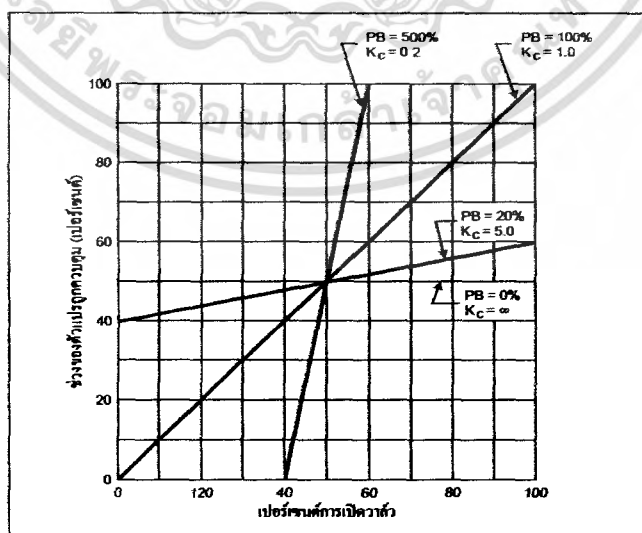
การควบคุมแบบโพรบโพซันแนล เป็นโหมดควบคุมที่ง่ายมีใช้ทั่วไปในโหมดควบคุมต่อเนื่องมาก โหมดนี้จะให้ความสัมพันธ์ระหว่างเอาต์พุตและอินพุตของตัวควบคุมเป็นเชิงเส้น (linear) และง่ายต่อการปรับแต่ง (tune) เพราะมีพารามิเตอร์ปรับค่าเพียงตัวเดียว อีกทั้งมีเสถียรภาพดี การตอบสนองเร็ว และมีเสถียรภาพสัมพัทธ์ทางไดนามิก

### ข้อเสีย

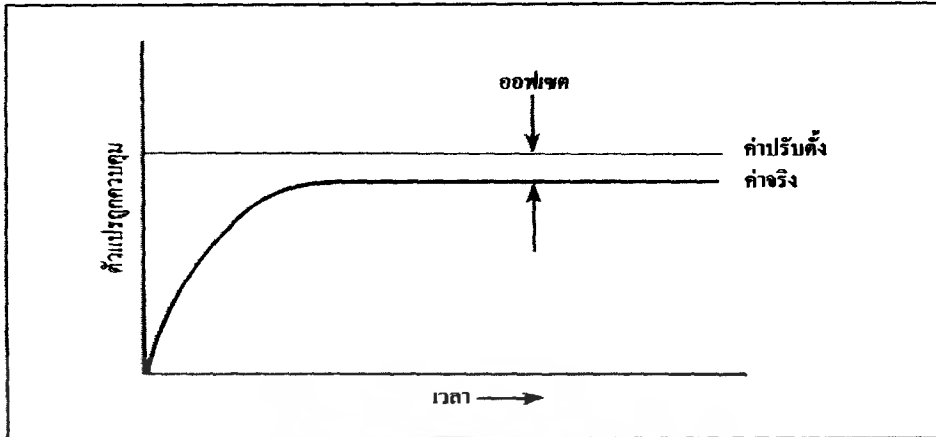
การควบคุมแบบโพรบโพซันแนล คือที่ค่าสถานะคงตัวมีออฟเซต (offset) อยู่เสมอคือที่สถานะคงตัวมีความแตกต่างระหว่างค่าที่ต้องการควบคุมหรือค่าปรับตั้งกับค่าจริงของตัวแปรถูกควบคุม



รูปที่ 2.23 ผลของการควบคุมแบบโพรบโพซันแนลต่อเอาต์พุตของตัวควบคุม



รูปที่ 2.24 ผลของโพรบโพซันแนล แบบค้ต่อการเปิดวาล์ว

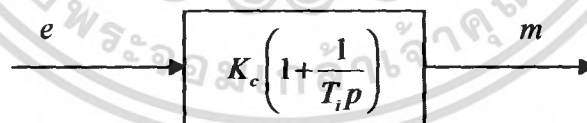


รูปที่ 2.25 ค่าโพรบโพชันแนล ออฟเซต

#### 2.4.2 การควบคุมแบบอินทิกรัล (Integral)

การควบคุมแบบอินทิกรัล คือการอินทิเกรตสัญญาณผิดพลาดของอินพุต  $e$  นั่นคือ การควบคุมแบบอินทิกรัลนั้นค่าตัวแปรจัดการจะเปลี่ยนด้วยอัตราที่เป็นสัดส่วนกับค่าผิดพลาด แต่ถ้าตัวแปรถูกควบคุมอยู่ที่ค่าปรับตั้ง (ไม่มีเบี่ยงเบน) ตัวควบคุมสุดท้ายจะนิ่งอยู่ที่ค่าเดิม นั่นหมายความว่า ที่สภาวะคงตัวหากมีการควบคุมแบบอินทิกรัลจะ ไม่มีออฟเซตเลย คือค่าผิดพลาดที่สภาวะคงตัว ต้องเป็นศูนย์

ในทางปฏิบัติการควบคุมแบบอินทิกรัลมักใช้ร่วมกับการควบคุมแบบโพรบโพชันแนล และเรียกชื่อใหม่ว่า การควบคุมแบบโพรบโพชันแนล-อินทิกรัล (proportion-integral mode, PI)



รูปที่ 2.26 แผนภาพบล็อกการควบคุมแบบ โพรบ โพชันแนล-อินทิกรัล

โดยที่

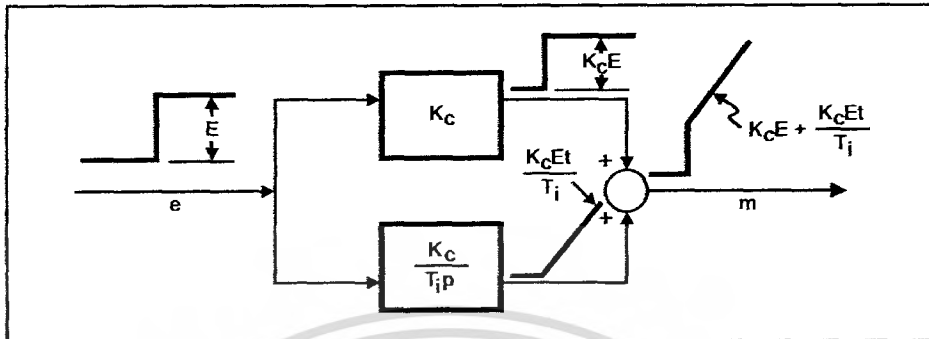
$K_c$  = โพรบโพชันแนลเกน

$T_i$  = เวลาอินทิกรัล (Integral time)

$e$  = สัญญาณผิดพลาด

$m$  = เอาท์พุทของตัวควบคุม

$p$  = ปฏิบัติการครีเวทฟเทียบกับเวลา,  $d/dt$  ดังนั้น  $1/p$  คือ การอินทิเกรตเทียบกับเวลา  $\int \dots dt$



รูปที่ 2.27 การตอบสนองต่อค่าผิดพลาดที่เป็นขั้นบันไดของตัวควบคุมแบบโพรบโพซันแนล-อินทิกรัล

#### ข้อดี

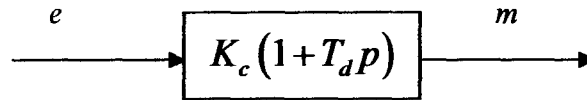
สามารถขจัดออฟเซตจากการควบคุมแบบโพรบโพซันแนลด้วยการควบคุมแบบอินทิกรัล

#### ข้อเสีย

เสถียรภาพจะลดลงเล็กน้อยจากการเพิ่มการควบคุมแบบอินทิกรัล เนื่องจากเสมือนว่าใส่วงจรถ่วง (lag) ให้แก่ระบบและการปรับแต่งมีความยุ่งยากขึ้นจากการควบคุมแบบโพรบโพซันแนลเพียงอย่างเดียว เพราะมีพารามิเตอร์ปรับค่าสองตัวคือ  $K_c$  และ  $T_i$  การควบคุมด้วยแบบอินทิกรัลเพียงอย่างเดียว เป็นวิธีที่เป็นไปได้แต่ไม่เป็นวิธีที่ใช้กันจึงไม่กล่าวในรายละเอียดต่อไป

#### 2.4.3 การควบคุมแบบครีเวทฟ (Derivative)

พื้นฐานการควบคุมแบบครีเวทฟ จะขึ้นกับอัตราการเปลี่ยนแปลงของสัญญาณผิดพลาด,  $e$  การควบคุมโดยใช้แบบครีเวทฟเพียงอย่างเดียวก็ไม่เหมาะกับการปฏิบัติจริงถึงแม้ทางทฤษฎีจะเป็นไปได้ ทั้งนี้เนื่องจากบางครั้งค่าผิดพลาดอาจมีค่ามากและคงสถานะนี้จนสัญญาณเอาต์พุทของตัวควบคุมเป็นศูนย์ได้ แต่ถ้าสัญญาณผิดพลาดเป็นฟังก์ชันขั้นบันไดสัญญาณเอาต์พุทของตัวควบคุมจะเข้าสู่สถานะอิ่มตัวการควบคุมก็จะหยุดชะงักเช่นกัน ดังนั้นจึงนิยมใช้การควบคุมแบบครีเวทฟ ร่วมกับการควบคุมแบบโพรบโพซันแนล (Proportional-Derivative mode, PD)



รูปที่ 2.28 แผนภาพบล็อกการควบคุมแบบโพรบโพซันเน็ล-เดริเวทิฟ

โดยที่

$K_c$  = โพรบโพซันเน็ลแกน

$e$  = สัญญาณผิดพลาด

$m$  = เอาท์พุทของตัวควบคุม

$T_d$  = เวลาปฏิบัติการเดริเวทิฟ (Derivative time)

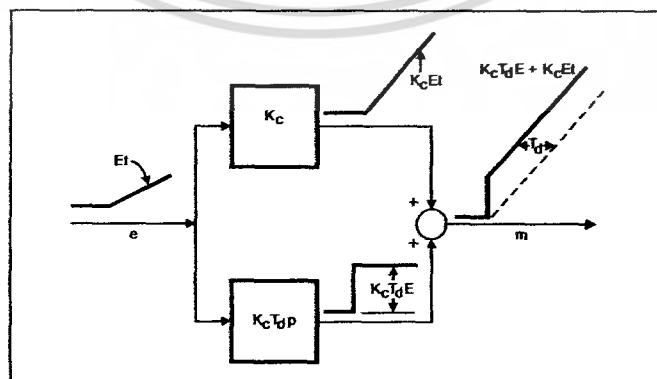
$p = \frac{d}{dt}$

ข้อดี

ประโยชน์ของการใส่การควบคุมแบบเดริเวทิฟ เหมือนในสว่างจรรยาหน้า (lead) เพิ่มเข้าไปในตัวควบคุมเพื่อชดเชยการล่าช้าในวงควบคุม ซึ่งทำให้ระบบมีเสถียรภาพ

ข้อเสีย

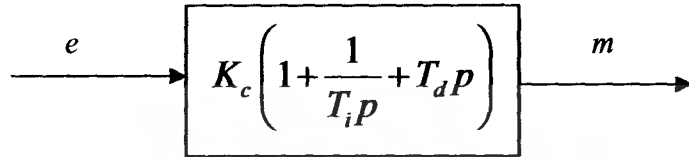
การปรับแต่งมีความยุ่งยากขึ้นจากการควบคุมแบบโพรบโพซันเน็ล เพียงอย่างเดียว เพราะว่าเพราะมีพารามิเตอร์ปรับค่าสองตัว ที่สำคัญคือถ้าวงควบคุม มีสัญญาณรบกวน การควบคุมแบบเดริเวทิฟ จะทำการคำนวณสัญญาณนั้นให้เป็นสัญญาณเอาท์พุทของตัวควบคุมด้วย ทำให้เกิดความผิดพลาดในการควบคุม จึงต้องมีการใส่ตัวกรองเข้าไปในส่วน โปรแกรมควบคุมหรือส่วนฮาร์ดแวร์



รูป 2.29 เอาท์พุทของตัวควบคุมแบบโพรบโพซันเน็ล -เดริเวทิฟ เมื่ออินพุทเป็นฟังก์ชันเร็มพ์

#### 2.4.4 การควบคุมแบบโพรบโพซันแน็ล-อินทิกรัล-เดริเวทิฟ

การควบคุมแบบโพรบโพซันแน็ล-อินทิกรัล-เดริเวทิฟ หรือเรียกสั้น ๆ ว่า การควบคุมแบบพีไอดีเป็นการควบคุมต่อเนื่องที่ทำงานสลับซับซ้อนมากที่สุดที่มีใช้ในวงควบคุมป้อนกลับ รูปที่ 2.30 แสดงแผนภาพบล็อกสำหรับตัวควบคุมพีไอดี



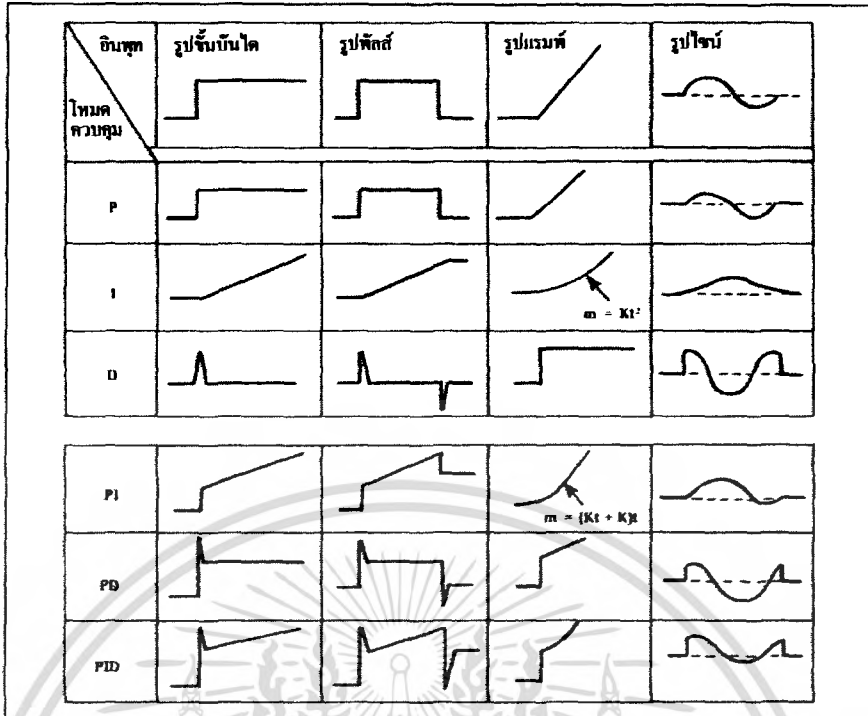
รูปที่ 2.30 แผนภาพบล็อกสำหรับตัวควบคุมพีไอดี

**ข้อดี** การควบคุมแบบสาม โหมดจะทำให้การตอบสนองของตัวแปรถูกควบคุมที่เร็วและไม่มีออฟเซต

**ข้อเสีย** การปรับแต่งค่าพารามิเตอร์มีความยุ่งยาก เพราะมีพารามิเตอร์ 3 ตัวคือ  $K_c$  และ  $T_i$  และ  $T_d$

ตารางที่ 2.1 สมการของการคณิตศาสตร์ควบคุมโหมดต่างๆ

สัญลักษณ์	ชนิด	สมการคณิตศาสตร์
P	โพรบ โพซันแน็ล	$m = K_c e$
I	อินทิกรัล	$m = \frac{1}{T_i} \int edt$
PI	โพรบ โพซันแน็ล-อินทิกรัล	$m = K_c \left[ e + \frac{1}{T_i} \int edt \right]$
PD	โพรบ โพซันแน็ล-เดริเวทิฟ	$m = K_c \left[ e + T_d \frac{de}{dt} \right]$
PID	โพรบ โพซันแน็ล-อินทิกรัล-เดริเวทิฟ	$m = K_c \left[ e + \frac{1}{T_i} \int edt + T_d \frac{de}{dt} \right]$



รูปที่ 2.31 การตอบสนองของตัวควบคุม โหมดต่าง ๆ

## 2.5 สมการดิฟเฟอเรนเชียล

เนื่องจากสัญญาณที่คอนโทรลเลอร์ นำมาคำนวณนี้เป็นสัญญาณดิฟเฟอเรนเชียล ดังนั้น สมการในการคำนวณพีไอดี เค็มที่ใช้สำหรับสัญญาณอนาล็อก จึงไม่สามารถนำมาใช้ได้โดยตรง สามารถพิจารณาได้ดังนี้

$$M(t) = M_P(t) + M_I(t) + M_D(t)$$

โดยที่  $M(t)$  คือ สัญญาณของคอนโทรลเลอร์

$M_P(t)$  คือ สัญญาณของการควบคุมแบบโพรบโพซันเน็ล

$M_I(t)$  คือ สัญญาณของการควบคุมแบบอินทิกรัล

$M_D(t)$  คือ สัญญาณของการควบคุมแบบเดริเวทีฟ

### 2.5.1 การควบคุมแบบโพรพอร์ชันแนล (Proportional)

ในส่วนของค่า  $M_p(t)$  (Proportional Controller) จะมีค่าเท่ากับ

$$M_p = K_p e(t)$$

### 2.5.2 การอินทิเกรต (Integral)

การอินทิเกรตแบ่งออกได้ 3 วิธีคือ

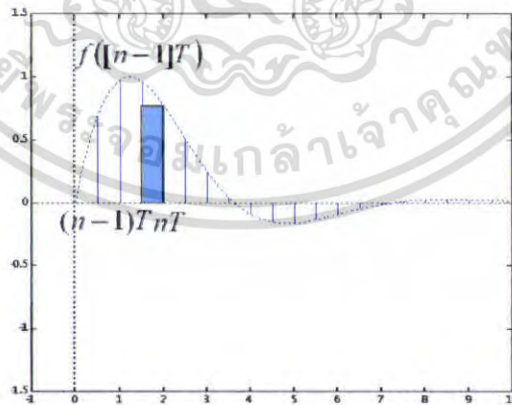
#### 1. การอินทิเกรตวิธีสี่เหลี่ยมผืนผ้า

การอินทิเกรตวิธีสี่เหลี่ยมผืนผ้า คือการหาพื้นที่ใต้กราฟ ซึ่งสามารถประมาณค่าพื้นที่ใต้กราฟเป็นรูปสี่เหลี่ยมผืนผ้าหลายๆรูปต่อกัน โดยในการหาพื้นที่ใต้กราฟนั้นจะหาได้โดยการคำนวณพื้นที่ของสี่เหลี่ยมผืนผ้าทุกรูปที่อยู่ใต้กราฟนำมาบวกกัน

จากสูตรการหาพื้นที่สี่เหลี่ยมผืนผ้า คือ กว้าง  $\times$  สูง

เมื่อกำหนด  $T$  เป็นคาบในการแซมปลิง (sampling) ที่มีค่าน้อย ๆ แล้ว จะคิดพื้นที่เป็นสี่เหลี่ยมผืนผ้าได้ ดังนี้ สูตรการอินทิเกรตวิธีสี่เหลี่ยมผืนผ้า

$$\int_{(n-1)T}^{nT} f(t) dt = T \times f(nT)$$



รูปที่ 2.32 การอินทิเกรตวิธีสี่เหลี่ยมผืนผ้า

สูตรการอินทิเกรตวิธีสี่เหลี่ยมผืนผ้า

$$\int_{(n-1)T}^{nT} f(t) dt = T * f(nT)$$

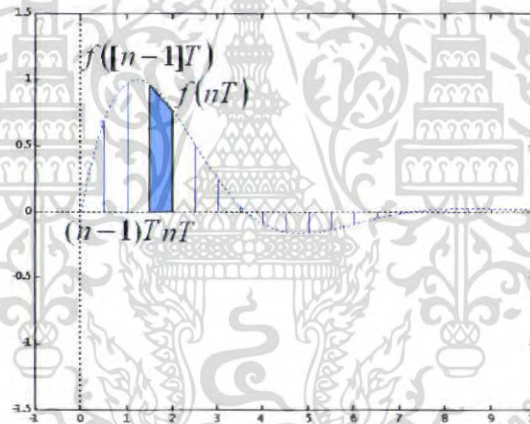
พิสูจน์การอินทิเกรตวิธีสี่เหลี่ยมผืนผ้า

$$g(t) = \int_0^t f(t) dt$$

$$g(nT) = g([n-1]T) + \int_{(n-1)T}^{nT} f(t) dt$$

$$g(nT) = g([n-1]T) + [T * f(nT)]$$

2. การอินทิเกรตวิธีสี่เหลี่ยมคางหมู



รูปที่ 2.33 การอินทิเกรตวิธีสี่เหลี่ยมคางหมู

การอินทิเกรตวิธีสี่เหลี่ยมคางหมู คือการหาพื้นที่ใต้กราฟ ซึ่งสามารถประมาณค่าพื้นที่ใต้กราฟเป็นแบบรูปสี่เหลี่ยมคางหมูหลายรูปต่อกัน ในการหาพื้นที่ใต้กราฟนั้นจะหาโดยการคำนวณพื้นที่ของสี่เหลี่ยมคางหมูทุกรูปที่อยู่ใต้กราฟนำมาบวกกัน

สูตรการหาพื้นที่สี่เหลี่ยมคางหมู คือ  $(\frac{1}{2}) \times (\text{ผลบวกของด้านคู่ขนาน})$

เมื่อกำหนด T เป็นคาบในการแซมปลิงมีค่าน้อย ๆ แล้ว จะคิดพื้นที่เป็นสี่เหลี่ยมคางหมูได้ ดังนี้

สูตรการอินทิเกรตวิธีสี่เหลี่ยมคางหมู

$$\int_{(n-1)T}^{nT} f(t) dt = \frac{T}{2} [f([n-1]T) + f(nT)]$$

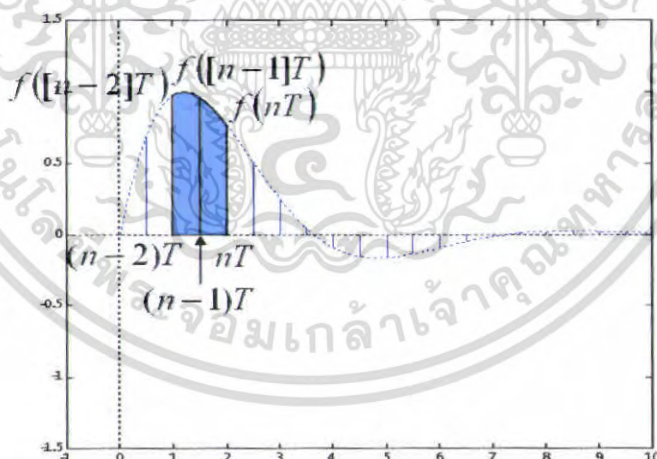
พิสูจน์การอินทิเกรตวิธีสี่เหลี่ยมคางหมู

$$g(t) = \int_0^t f(t) dt$$

$$g(nT) = g([n-1]T) + \int_{(n-1)T}^{nT} f(t) dt$$

$$g(nT) = g([n-1]T) + \frac{T}{2} [f([n-1]T) + f(nT)]$$

3. การอินทิเกรตวิธีซิมสัน 1/3



รูปที่ 2.34 การอินทิเกรตวิธีซิมสัน 1/3

การอินทิเกรตวิธีซิมสัน 1/3 คือการหาพื้นที่ใต้กราฟ ซึ่งสามารถประมาณค่าพื้นที่ใต้กราฟเป็นรูปแบบซิมสัน 1/3 หลายรูปต่อกัน ในการหาพื้นที่ใต้กราฟแบบซิมสันนั้นจะได้ผลที่ใกล้เคียงกับความเป็นจริงมาก

การอินทิเกรตวิธีซิมสัน 1/3

$$\int_{(n-2)T}^{nT} f(t)dt = \frac{T}{3} [f([n-2]T) + 4f([n-1]T) + f(nT)]$$

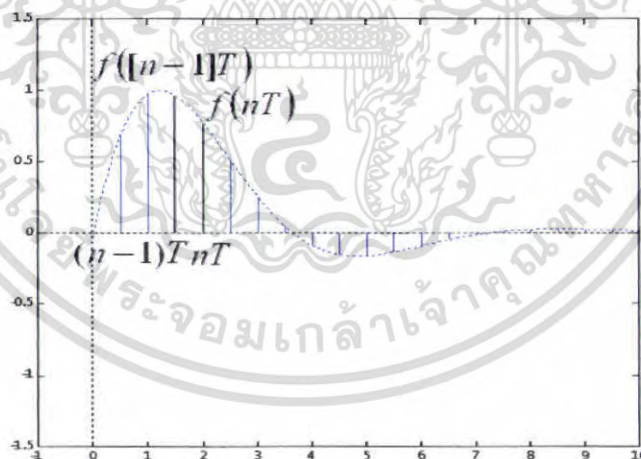
พิสูจน์การอินทิเกรตวิธีซิมสัน 1/3

$$g(t) = \int_0^t f(t)dt$$

$$g(nT) = g([n-2]T) + \int_{(n-2)T}^{nT} f(t)dt$$

$$g(nT) = g([n-2]T) + \frac{T}{3} [f([n-2]T) + 4f([n-1]T) + f(nT)]$$

### 2.5.3 การอนุพันธ์ (Derivative)



รูปที่ 2.35 แสดงลักษณะการอนุพันธ์

ค่าของตัวควบคุมแบบเดริเวทีฟ จากรูปค่าในส่วนของเดริเวทีฟก็คือความชัน (Slope) ของกราฟนั่นเอง

## พิสูจน์การอนุพันธ์

$$g(t) = \frac{d}{dt} f(t)$$

$$g(nT) = \frac{f(nT) - f([n-1]T)}{T}$$

### 2.5.4 การออกแบบตัวควบคุมแบบพีไอดี

ในการวิเคราะห์และออกแบบตัวควบคุมนั้น จำเป็นต้องศึกษาธรรมชาติและคุณลักษณะของกระบวนการ รวมถึงอุปกรณ์ต่างๆที่ใช้ในการควบคุมกระบวนการทั้งหมด เพื่อใช้ศึกษาและวิเคราะห์ผลกระทบ ที่เกิดขึ้นต่อกระบวนการเมื่อถูกรบกวนจากการเปลี่ยนแปลงของสภาพแวดล้อมภายนอกหรือได้รับการควบคุมจากเครื่องควบคุมหรืออุปกรณ์ควบคุมอื่นๆ ทั้งนี้ก็เพื่อที่จะได้เลือกการควบคุมและค่าพารามิเตอร์ของตัวควบคุมที่เหมาะสม ในอันที่จะได้รับการควบคุมที่มีประสิทธิภาพมากที่สุด โดยทั่วไป การศึกษาธรรมชาติ และคุณลักษณะ หรือคุณสมบัติต่างๆของกระบวนการที่ใช้กันอยู่ มี 2 วิธี คือ

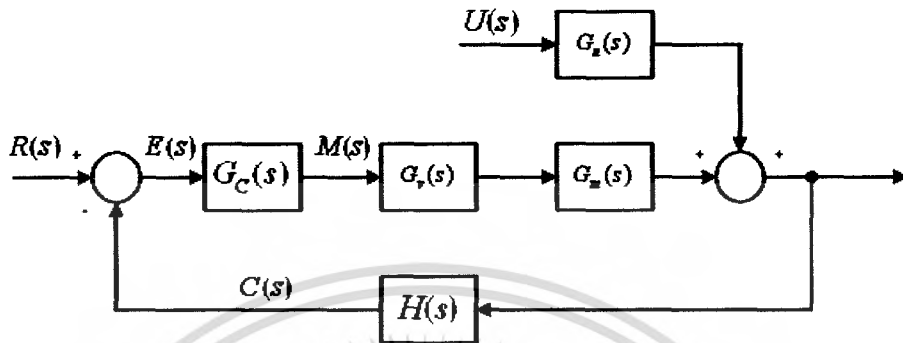
1. การศึกษาคุณลักษณะของกระบวนการ โดยใช้วิธีการทดสอบ วิธีนี้การวิเคราะห์กระบวนการสามารถทำได้โดยใช้เครื่องวัดและเครื่องควบคุมทดลองป้อนสัญญาณอินพุท และสังเกตผลการเปลี่ยนแปลงของสัญญาณเอาต์พุทนี้จะนำไปพยากรณ์ หรือประมาณค่าพารามิเตอร์ของกระบวนการ เพื่ออธิบายคุณลักษณะกระบวนการต่อไป

2. การศึกษาคุณลักษณะของกระบวนการ โดยใช้วิธีการวิเคราะห์ทางคณิตศาสตร์ วิธีนี้การวิเคราะห์คุณลักษณะของอุปกรณ์และกระบวนการจะถูกแสดงในลักษณะของสมการคณิตศาสตร์ต่างๆ และวิเคราะห์คุณสมบัติของกระบวนการจากการหาคำตอบของสมการเหล่านี้การศึกษาธรรมชาติและคุณลักษณะหรือคุณสมบัติของกระบวนการ โดยใช้วิธีการวิเคราะห์ทางคณิตศาสตร์ เป็นวิธีที่ยุ่งยาก และต้องอาศัยความรู้ทางคณิตศาสตร์ และฟิสิกส์ ซึ่งจำเป็นต้องอาศัยผู้มีความรู้ และความชำนาญ ดังนั้น ในทางปฏิบัติระบบควบคุมของกระบวนการทาง

โดยทั่วไปในอุตสาหกรรมส่วนใหญ่จะใช้วิธีการศึกษาคุณลักษณะของกระบวนการ โดยวิธีการทดลองป้อนสัญญาณขั้นบันไดให้กระบวนการ เพื่อสังเกตผลที่จะเกิดขึ้นในสภาพความเป็นจริง และนำค่าพารามิเตอร์ หรือ ค่าคุณลักษณะของกระบวนการที่ได้ไปใช้ในการสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุม เพื่อใช้ในการควบคุมกระบวนการต่อไป

## 2.6 การวิเคราะห์คุณลักษณะของกระบวนการ

### 2.6.1 การวิเคราะห์คุณลักษณะของกระบวนการ



รูปที่ 2.36 บล็อกไดอะแกรมของระบบควบคุมแบบป้อนกลับ

โครงสร้างของระบบควบคุมแบบป้อนกลับโดยทั่วไป แสดงโดยรูปที่ 2.36

เมื่อ

$R(s)$  = การแปลงลาปลาซ (Laplace Transform) ของค่าอินพุตอ้างอิงหรือค่าเป้าหมาย

$E(s)$  = การแปลงลาปลาซของค่าความคลาดเคลื่อน (Error)

$M(s)$  = การแปลงลาปลาซของสัญญาณควบคุมจากตัวควบคุม

$C(s)$  = การแปลงลาปลาซของสัญญาณเอาต์พุตของทรานสมิตเตอร์ (Transmitter)

$U(s)$  = การแปลงลาปลาซของสัญญาณรบกวน (Disturbance)

$G_c(s)$  = ฟังก์ชันถ่ายโอนของตัวควบคุม

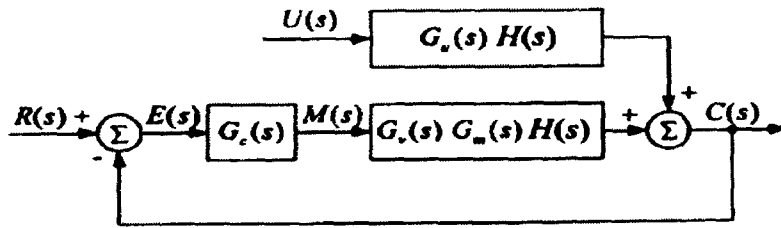
$G_v(s)$  = ฟังก์ชันถ่ายโอนของวาล์วควบคุม(หรืออุปกรณ์ควบคุมสุดท้าย)

$G_m(s)$  = ฟังก์ชันถ่ายโอนของกระบวนการซึ่งอยู่ระหว่างสัญญาณที่ได้จากการควบคุม และสัญญาณการทำงาน

$G_d(s)$  = ฟังก์ชันถ่ายโอนของกระบวนการซึ่งอยู่ระหว่างสัญญาณที่ได้จากการควบคุม และสัญญาณรบกวน

$H(s)$  = ฟังก์ชันถ่ายโอนของเซนเซอร์-ทรานสมิตเตอร์ (Sensor – Transmitter)

โครงสร้างของระบบควบคุมแบบป้อนกลับในรูป สามารถสรุปเพื่อให้เข้าใจง่ายดังรูปที่ 2.37



รูปที่ 2.37 บล็อกไดอะแกรมของระบบควบคุมแบบป้อนกลับหลังการลดรูป

กำหนดให้ การรวมกันของ  $G_v(s)G_m(s)H(s)$  เป็นฟังก์ชันถ่ายโอนของกระบวนการดังนั้น

$$G(s) = G_v(s)G_m(s)H(s)$$

ซึ่งฟังก์ชันถ่ายโอนของกระบวนการ แสดงตามสมการได้ดังนี้

กระบวนการอันดับหนึ่งที่มีเวลาไร้ผลตอบสนอง (First-Order-Plus-Dead-Time (FOPDT) Model)

$$G(s) = \frac{Ke^{-t_0s}}{\tau s + 1}$$

กระบวนการอันดับสองที่มีเวลาไร้ผลตอบสนอง (Second-Order-Plus-Dead-Time (SOPDT) Model)

$$G_1(s) = \frac{Ke^{-t_0s}}{(\tau_1s + 1)(\tau_2s + 1)}$$

หรือ

$$G_1(s) = \frac{Ke^{-t_0s}}{\tau^2 + 2\zeta\tau s + 1}$$

โดยที่  $K$  = อัตราการขยายของกระบวนการ (Process Gain)

$t_0$  = ค่าการหน่วงเวลาของกระบวนการ (Process Dead Time)

$\tau, \tau_1, \tau_2$  = ค่าคงตัวของเวลาของกระบวนการ (Process Time Constant)

$\zeta$  = อัตราการหน่วง (damping ratio)

โดยทั่วไป ฟังก์ชันถ่ายโอนของกระบวนการจะถูกประมาณการให้เป็นรูปแบบของกระบวนการอันดับหนึ่งที่มีเวลาไร้ผลตอบสนอง (First-order Plus Dead Time (FOPDT)) เนื่องจาก

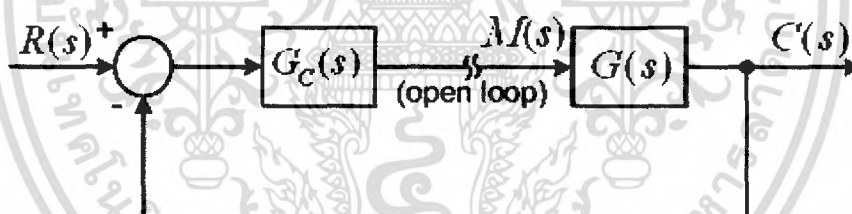
เป็นการประมาณการที่ดีที่สุดสำหรับกระบวนการทางอุตสาหกรรม โดยจากสมการคุณลักษณะของระบบการที่เราจะหาค่าพารามิเตอร์ของพีไอดี เราจะต้องรู้ค่าตัวแปร 3 ค่าด้วยกันคือ ค่าอัตราขยายของกระบวนการ ( $K$ ) ค่าการหน่วงเวลาของกระบวนการ ( $t_0$ ) และค่าคงตัวของเวลาของกระบวนการ ( $\tau$ ) จากสมการจะเห็นได้ว่าคุณลักษณะของกระบวนการอยู่ในรูปของตัวแปร 3 ตัว ซึ่งสามารถหาได้จากการทดสอบกระบวนการหรือการหาค่าจากเส้นโค้งปฏิกิริยากระบวนการ ซึ่งจะนำเสนอในหัวข้อถัดไป

## 2.6.2 วิธีการหาพารามิเตอร์จากเส้นโค้งปฏิกิริยากระบวนการ (Process Reaction Curve)

### วิธีการหาพารามิเตอร์โดยวิธีเส้นโค้งปฏิกิริยากระบวนการ

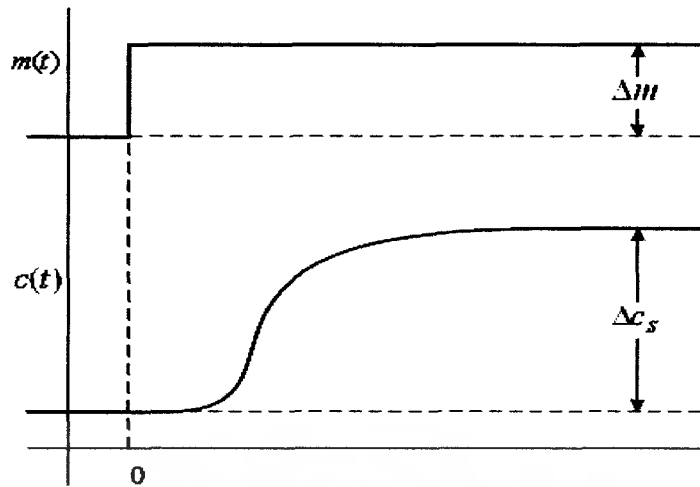
ขั้นตอนการหาเส้นโค้งปฏิกิริยากระบวนการมีขั้นตอนดังนี้

1. เลือกตัวควบคุมให้ทำงานแบบควบคุมด้วยมือ (Manual) หรือเป็นการเปิดวงของกระบวนการดังรูปที่ 2.38 แล้วทำการป้อนสัญญาณขั้นบันได  $m(t)$  เข้าไปยังระบบ โดยที่ขนาดสัญญาณขั้นบันไดที่ป้อนเข้าไปในนั้นจะต้องมีขนาดใหญ่เพียงพอที่จะทำให้เครื่องมือวัดตรวจจับและวัดค่าได้ แต่ขนาดไม่ใหญ่จนทำให้สัญญาณผิดเพี้ยนได้



รูปที่ 2.38 บล็อก ไดอะแกรมสำหรับการทดสอบกระบวนการแบบวงเปิด

2. บันทึกค่าผลตอบสนองที่ออกมาจากกระบวนการ  $C(t)$  จากอุปกรณ์บันทึกสัญญาณที่สามารถมั่นใจได้ว่าค่าที่ได้จะเหมาะสมทั้งสองค่าทั้งขนาดและระยะเวลาการทำงานของกระบวนการ ในการบันทึกค่านั้นต้องบันทึกค่าตั้งแต่แรกจนกระทั่งผลตอบสนองที่ออกมาจากกระบวนการคงที่ ซึ่งระยะเวลาในการบันทึกขึ้นอยู่กับความไวของผลตอบสนองในกระบวนการ บางกระบวนการอาจจะใช้เวลา 2-3 นาที แต่บางกระบวนการอาจใช้เวลา 2-3 ชั่วโมง ซึ่งตลอดเวลาในการวัดค่าต้องไม่ให้มีสัญญาณอื่นมารบกวนระบบ จากการทดสอบเราจะได้ผลตอบสนองกระบวนการเป็นรูปร่างเป็นตัวเอส (S) ดังรูปที่ 2.39 ซึ่งผลตอบสนองจะครอบคลุมถึงกระบวนการอันดับสองและกระบวนการอันดับสูง ที่มีหรือไม่มีเวลาไรผลตอบสนองก็ได้



รูปที่ 2.39 ผลตอบสนองจากการทดสอบกระบวนการแบบวงเปิด

จากการทดสอบกระบวนการจะได้สมการดังนี้

$$C(s) = G(s)M(s)$$

พิจารณาการเปลี่ยนแปลงค่าของสัญญาณควบคุม  $\Delta m$  และการกระบวนการตามรูปแบบของกระบวนการอันดับหนึ่งที่มีเวลาไร้ผลตอบสนอง ในรูปของลาปลาซ ได้ว่า

$$C(s) = \frac{Ke^{-t_0s}}{\tau s + 1} \frac{\Delta m}{s}$$

ทำการกระจายเศษส่วนย่อยได้สมการดังนี้

$$C(s) = K\Delta m e^{-t_0s} \left[ \frac{1}{s} - \frac{\tau}{\tau s + 1} \right]$$

แปลงลาปลาซผกผันเพื่อให้อยู่ในโดเมนของเวลา

$$C(t) = K \times \Delta m \times u(t - t_0) \left[ 1 - e^{-(t-t_0)/\tau} \right]$$

เมื่อสัญญาณขั้นบันไดในพจน์ของ  $u(t-t_0)$  กำหนดให้

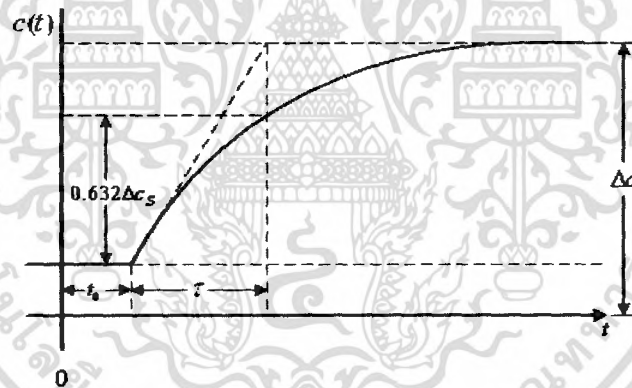
$$C(t) = t \leq t_0$$

โดย  $\Delta c(t) = c(t) - c(0)$  เป็นการเปลี่ยนแปลงของสัญญาณเอาต์พุตของกระบวนการจากค่าสถานะ (Initial Value) ดังนั้นเมื่อผลตอบสนองของกระบวนการเข้าสู่สถานะคงที่ (Steady State)  $\Delta c(t)$  จะเขียนแทนด้วย  $\Delta c_s$  ซึ่งมีค่าเท่ากับ

$$\Delta c_s = \lim_{t \rightarrow \infty} \Delta c(t) = K \Delta m$$

ดังนั้นค่าของอัตราขยายของระบบจะมีค่าเท่ากับ

$$K = \frac{\Delta c_s}{\Delta m}$$



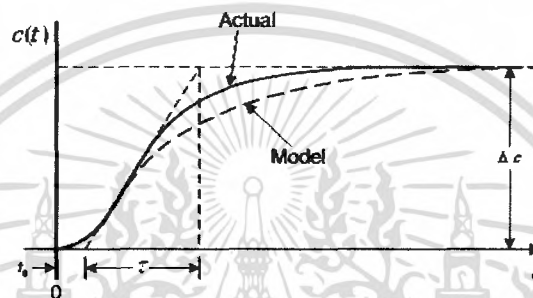
รูปที่ 2.40 ผลตอบสนองขั้นบันไดของระบบ FODPT

โดยจากรูปที่ 2.40 จะได้ว่า  $t_0$  คือเวลาไร้ผลตอบสนอง (dead time) และ  $\tau$  คือ ค่าคงตัวเวลา (time constant)

เมื่อเราได้เส้นโค้งกระบวนการแล้วเราสามารถหาค่า อัตราขยายของกระบวนการ(K), ค่าการหน่วงเวลาของกระบวนการ ( $t_0$ ), และค่าคงตัวเวลาของกระบวนการ ( $\tau$ ) เพื่อนำพารามิเตอร์ต่างๆ เพื่อใช้ในการหาค่าปรับแต่งพีไอดี ตามวิธีของ ซีกเลอร์และนิโคลส์ และวิธีของคาลินห์มีวิธีการหาได้ 3 วิธีคือ

วิธีที่ 1 ฟิต 1 (Fit 1) มีวิธีการคือ การใช้เส้นสัมผัสวงไปสัมผัสกับเส้นโค้งของผลตอบสนอง โดยลากเส้นเฉียงจากจุดที่มีการเริ่มการเปลี่ยนแปลงไปถึงจุดที่มีการเปลี่ยนแปลงมากที่สุด โคนเส้นที่ลานั้นต้องสัมผัสกับโค้งของผลตอบสนองมากที่สุดสำหรับรูปแบบของกระบวนการอันดับหนึ่งที่มีเวลาไร้ผลตอบสนอง (First-order Plus Dead Time) จะเริ่มลากเส้นเฉียงที่เวลา จากการตรวจสอบอย่างละเอียดพบว่าผลตอบสนองดังรูป 2.43 จะได้สมการที่เริ่มต้นของอัตราการเปลี่ยนแปลงคือ

$$\left. \frac{dC}{dt} \right|_{t_0} = K \Delta m \left[ \frac{1}{\tau} \right] = \frac{\Delta c_s}{\tau}$$

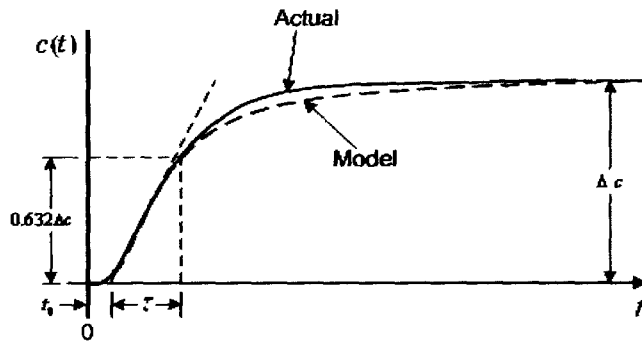


รูปที่ 2.41 ค่าพารามิเตอร์แบบจำลอง FOPDT จากวิธีฟิต 1

จากรูปที่ 2.41 เห็นว่าการลากเส้นเฉียงจะเริ่มต้นที่เวลา  $t = t_0$  และไปสิ้นสุดที่เวลา  $t_0 + \tau$  ซึ่งจากกราฟเราจะได้ค่าของ  $t_0$  และ  $\tau$  และจากรูปจะพบว่าค่าของค่าคงตัวของเวลาสูงกว่าความเป็นจริง

วิธีที่ 2 ฟิต 2 (Fit 2) ในวิธีการนี้ ค่าเวลาไร้ผลตอบสนองของกระบวนการ  $t_0$  จะเหมือนกับวิธีที่ 1 แต่ค่าคงตัวของเวลาของกระบวนการ  $\tau$  ที่ได้มาจากวิธี 1 เป็นค่าที่ได้รับอิทธิพลจากรูปแบบของผลตอบสนองที่เวลา  $t = t_0 + \tau$  จากสมการ  $C(t) = K \times \Delta m \times u(t - t_0) [1 - e^{-(t-t_0)/\tau}]$  แทนค่า  $t = t_0 + \tau$  จะได้

$$C(t) = K \times \Delta m [1 - e^{-1}] = 0.632 \Delta c_s$$



รูปที่ 2.42 ค่าพารามิเตอร์แบบจำลอง FOPDT จากวิธีฟิต 2

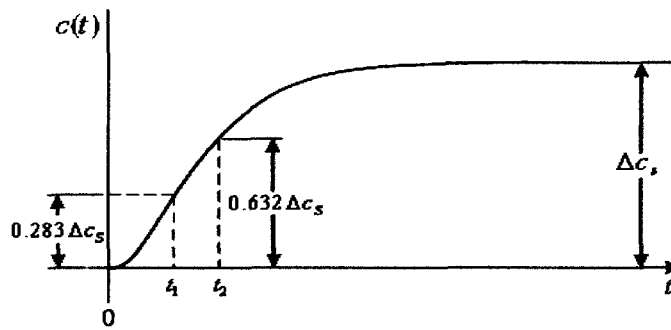
จากรูปที่ 2.42 เป็นการแสดงรูปแบบจำลองของการใช้วิธีที่ 2 จะเห็นผลตอบสนองมีค่าใกล้เคียงกับความเป็นจริงมากกว่าวิธีที่ 1 โดยค่า  $\tau$  ของวิธีที่ 2 ปกติจะมีค่าน้อยกว่าวิธี ฟิต 1 และค่า  $t_0$  มีค่าเท่ากัน

วิธีที่ 3 ฟิต 3 (Fit 3) เป็นการปรับปรุงจากวิธี ฟิต 2 วิธีแรกเพราะการกำหนดค่า  $t_0$  และ  $\tau$  ของ 2 วิธีการแรกมีความถูกต้องน้อย เพราะการลากเส้นสัมผัสกับเส้น โค้งผลตอบสนองของกระบวนการ ณ จุดที่มีการเปลี่ยนแปลงสูงสุดที่ค่า  $(t_0 + \tau)$  จะขึ้นอยู่กับผู้ที่ลากเส้น ในการประมาณค่าทั้งสองเพื่อจัดการลากเส้น โคนใช้เส้นสัมผัส ดอกเตอร์ ซีซิล แอล สมิทท์ (Dr. Cecil L. Smith) ได้นำเสนอวิธีการเลือกค่า  $t_0$  และ  $\tau$  โดยเลือกจากแบบจำลองและแบบความเป็นจริงในจุดที่มีการเปลี่ยนแปลงสูงสุด โดยได้เลือกจุดที่เวลา  $(t_0 + \tau/3)$  และ  $(t_0 + \tau)$  ตามลำดับซึ่งการหาจุดแรกได้จาก

$$C\left(t_0 + \frac{\tau}{3}\right) = K \times \Delta m [1 - e^{-1/3}] = 0.283\Delta c_s$$

และจุดสองหาได้จาก

$$C(t) = K \times \Delta m [1 - e^{-1}] = 0.632\Delta c_s$$



รูปที่ 2.43 ค่าพารามิเตอร์แบบจำลอง FOPDT จากวิธีพีต3

กำหนดให้เวลาที่

$$t_1 = \left(t_0 + \frac{\tau}{3}\right)$$

และ

$$t_2 = (t_0 + \tau)$$

ดังนั้น

$$t_0 = t_2 - \tau$$

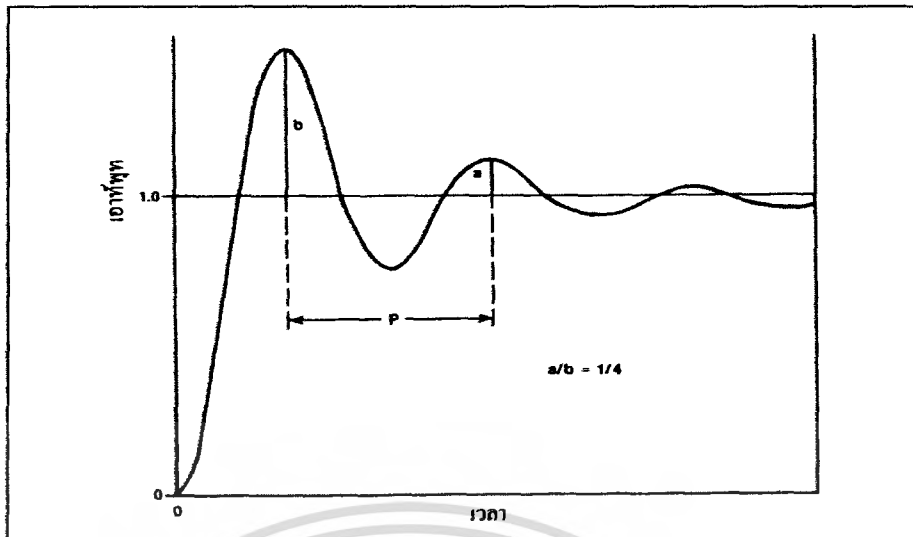
$$\tau = \frac{3}{2}(t_2 - t_1)$$

ค่าคุณลักษณะของกระบวนการทั้ง 3 ค่าคือ  $K$ ,  $\tau$  และ  $t_0$  นี้จะนำไปใช้ในการหาสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุมพีไอดี

## 2.7 วิธีการหาค่าพารามิเตอร์ในการปรับแต่งกระบวนการแบบต่างๆ

### 2.7.1 การควบคุมแบบอัตราส่วนการลดลงหนึ่งส่วนสี่ (Quarter Decay Ratio)

เนื่องจากเกิดปัญหาที่ว่าอะไรคือการควบคุมที่ดี ซึ่งในแต่ละกระบวนการอาจแตกต่างกัน ดังนั้นจึงมีเกณฑ์ปกติที่ใช้กำหนดการควบคุมที่ดี คือการปรับตัวควบคุมจนกระทั่งเส้นโค้งตอบสนองมีอัตราส่วนของแอมพลิจูด (amplitude) หรืออัตราส่วนการลดทอน (decay ratio) เท่ากับหนึ่งส่วนสี่ อัตราส่วนการลดทอนหนึ่งส่วนสี่คืออัตราส่วนยอดพุ่งเกิน (overshoot) สูงสุดยอดแรก ของเส้นโค้งตอบสนองกระบวนการต่อยอดพุ่งเกินยอดที่สองเท่ากับสี่ต่อหนึ่งดังแสดงในรูปที่ 9.1



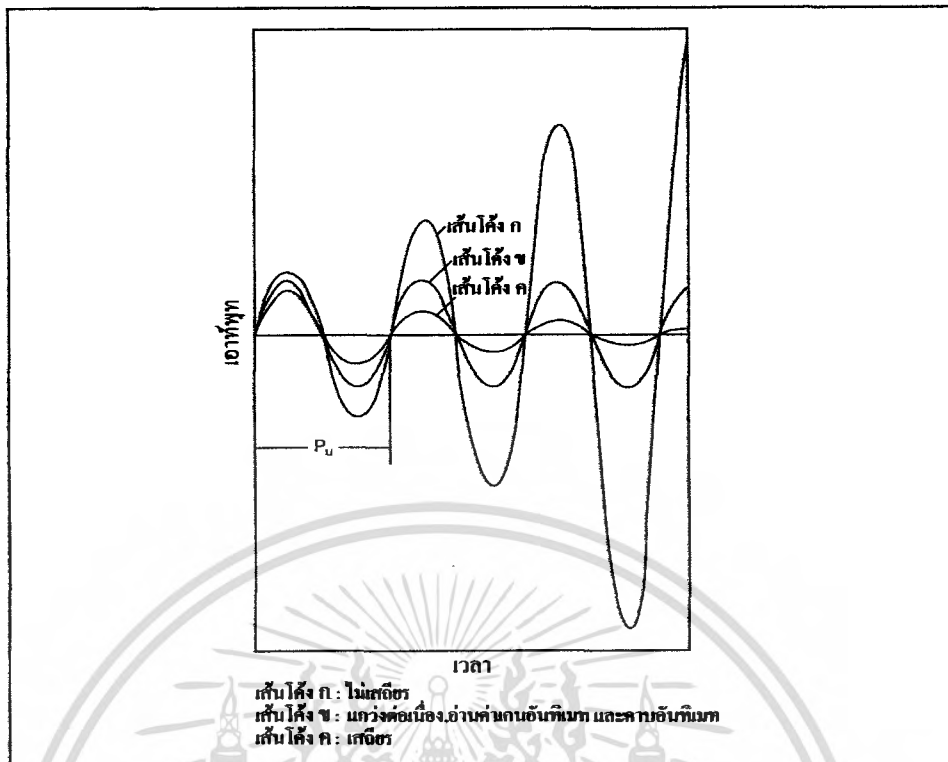
รูปที่ 2.44 เส้นโค้งตอบสนองกระบวนกรสำหรับอัตราส่วนการลดลงหนึ่งส่วนสี่

ซึ่งอัตราส่วนการลดลงเท่ากับหนึ่งส่วนสี่ แต่เป็นการแสดงค่าที่เหมาะสมที่สุดของช่วงสถานะชั่วคราว (Transient) กับช่วงสถานะคงตัว (Steady-State) ในบางกรณีมีความสำคัญที่ต้องปรับแต่งวงจรควบคุมไม่ให้มีค่าพุงเกิน ในบางกรณีอาจต้องการการตอบสนองที่เรียบ (smooth) และช้า ในบางวงจรควบคุมต้องการการตอบสนองเร็วและไม่สนใจปัญหาการแกว่ง (oscillation) และอะไรอื่น ๆ อีกปัญหาคือต้องกำหนดว่าอะไรคือการควบคุมที่ดีสำหรับแต่ละระบบ

### 2.7.2 การหาพารามิเตอร์โดยวิธีอัลทิเมท (Ultimate Method)

วิธีอัลทิเมทนำเสนอโดย ซีเกลอร์ (Ziegler) และ นิโคลส์ (Nichols) ในปี 1942 การนำคำว่าอัลทิเมทมาใช้ในวิธีนี้เนื่องจากต้องหาค่าเกนอัลทิเมท (ultimate gain) และคาบอัลทิเมท (ultimate period) ค่าเกนอัลทิเมทคือค่าเกนสูงสุดที่สามารถทำให้ระบบมีเสถียรภาพในที่ตัวควบคุมทำงานด้วยโหมดโพรบโพซันแน็ลเพียงอย่างเดียว

ในระบบควบคุมป้อนกลับใด ๆ หากมีการปิดวงควบคุม(หากตัวควบคุมถูกตั้งอยู่ที่ตำแหน่งอัตโนมัติ) ผู้ควบคุมสามารถเพิ่มเกนของตัวควบคุมไปเรื่อยๆ วงควบคุมจะเข้าสู่การแกว่งมากขึ้นๆ และต้องเพิ่มค่าเกนต่อไปอีกจะปรากฏการแกว่งต่อเนื่องที่ตัวแปรถูกควบคุม นี่ก็คือค่าเกนสูงสุดที่ระบบสามารถทำงานได้ก่อนกลายเป็นระบบไม่เสถียร ค่าเกนนี้เรียกว่าอัลทิเมทเกน และคาบ  $\theta$  เวลาที่เกิดการแกว่งต่อเนื่องคือคาบอัลทิเมท  $\theta$  จุดนี้หากมีการเพิ่มค่าเกนอีกระบบจะกลายเป็นไม่เสถียรทันที ค่าอัลทิเมทเกนและคาบอัลทิเมทแสดงในรูปที่ 2.45



รูปที่ 2.45 การตอบสนองที่แสดงค่าอัลทิเมทเคนและคาบอัลทิเมท

### 2.7.2.1 ขั้นตอนการหาค่าอัลทิเมทเคนและคาบอัลทิเมท

1. ปรับค่าเวลาอินทิกรัล (Integral time) และเวลาเดริเวทีฟ (Derivative time) ให้หยุดปฏิบัติการ ปล่อยให้ตัวควบคุมทำงาน เฉพาะโหมคโพรบโพซันแน็ลนั่นคือปรับค่า  $T_i$  เท่ากับอนันต์ และ  $T_d$  เท่ากับศูนย์ หรือเข้าใกล้ค่าดังกล่าวให้มากที่สุดเท่าที่ตัวควบคุมจะทำได้
2. รักษาตัวควบคุมให้อยู่ในตำแหน่งอัตโนมัติ นั่นคือปล่อยให้วงควบคุมปิด
3. ตั้งค่าโพรบโพซันแน็ลเกินที่ค่าต่างๆ ให้ทดลองดูว่าค่าเกินต่ำพอหรือยัง โดยการขยับค่าปรับตั้งไปมาเป็นเวลาสอง-สามวินาทีและตั้งกลับมาไว้ที่ค่าเดิม แล้วสังเกตเส้นโค้งการตอบสนอง
4. หากเส้นโค้งการตอบสนองไม่มีการขยับตัวลงดังเส้นโค้ง ก ในรูปที่ 2.45 แสดงว่าค่าเกินยังสูงเกินไปต้องลดค่าเกินลงและทำซ้ำขั้นตอนที่ 3
5. หากเส้นโค้งการตอบสนองมีการขยับตัวดังเส้นโค้ง ค ในรูปที่ 2.45 แสดงว่าค่าเกินต่ำเกินไปต้องเพิ่มค่าเกินหรือ และทำซ้ำขั้นตอนที่ 3
6. เมื่อได้เส้นโค้งการตอบสนอง ดังเส้นโค้ง ข ในรูปที่ 2.45 แสดงว่าได้ค่าอัลทิเมทเคนแล้วให้จดค่าทั้งอัลทิเมทเคนและคาบอัลทิเมทไว้ นี่คืค่าเกินที่ระบบอยู่ในสภาวะการแกว่งต่อเนื่อง ให้ค่าอัลทิเมทเคนเป็น  $K_u$  และคาบอัลทิเมทเป็น  $P_u$

ด้วยเหตุผลและการทดสอบที่คล้ายกันจะ ได้สมการที่เป็นกฎเกณฑ์ง่าย ๆ และดีสำหรับการปรับตั้งตัวควบคุมที่มีความสลับซับซ้อนมากขึ้นดังนี้

ตารางที่ 2.2 ค่าพารามิเตอร์โดยวิธีการอัลทิเมท

ชนิดของตัวควบคุม	โพรบโพ ชั้นเน็ลเกน $K'_c$	เวลา อินทิกรัล $\tau'_i$	เวลา เดริเวทิฟ $\tau'_D$
โพรบ โพชั้นเน็ล, พี	$\frac{K_{cu}}{2}$	-	-
โพรบ โพชั้นเน็ล-อินทิกรัล, พีไอ	$\frac{K_{cu}}{2.2}$	$\frac{T_{cu}}{1.2}$	-
โพรบ โพชั้นเน็ล-อินทิกรัล- เดริเวทิฟ, พีไอดี	$\frac{K_{cu}}{1.7}$	$\frac{T_{cu}}{2}$	$\frac{T_{cu}}{8}$

วิธีอัลทิเมทที่ได้นำเสนอใช้กับระบบวงปิดไม่เหมาะกับกระบวนการที่ทำงานช้า (slow process) เช่นเตาเผาอิฐปูน กระบวนการที่แยกลำดับส่วนหรือหม้อไอน้ำ ทั้งนี้เนื่องจาก มันอาจต้องใช้เวลาเป็นวันจึงจะเห็นรูปแบบการควบคุมและ ไม่เป็นการปลอดภัยและประหยัดที่จะปล่อยให้ระบบเกิดการแกว่ง

### 2.7.3 การหาค่าพารามิเตอร์โดยวิธีเส้นโค้งปฏิบัติการกระบวนการ(Process Reaction Curve)

ปัญหาหนึ่งของวิธีอัลทิเมทที่ใช้กับวงควบคุมปิดคือต้องทำให้วงควบคุมเกิดการแกว่งเพื่อหาค่าอัลทิเมทเกนและคาบอัลทิเมท ห้องควบคุมบางแห่งมีปัญหาเกี่ยวกับเรื่องนี้มาก วิธีการวงเปิดต้องการเฉพาะที่จะทำให้กระบวนการอยู่ในสภาวะควบคุมไม่ได้ (เกิดการแกว่งต่อเนื่อง) แต่วิธีเส้นโค้งปฏิบัติการกระบวนการจะให้ข้อมูลเกี่ยวกับทาง ไคเนมิกของระบบควบคุมป้อนกลับได้แน่นอนกว่า และใช้ได้กับ ทุกกระบวนการ

เส้นโค้งปฏิบัติการกระบวนการคือปฏิบัติการของกระบวนการต่อสัญญาณอินพุตแบบขั้นบันได เส้นโค้งปฏิบัติการกระบวนการคือปฏิบัติการของอุปกรณ์อื่นทั้งหมด ดังที่ได้เสนอในการพิตแบบต่างๆ โดยในที่นี้ซีเกลอร์และนิโคลส์ ได้ใช้วิธีการ พิต1 ในการหาค่าพารามิเตอร์ต่างๆ ได้ดังนี้

ตารางที่ 2.3 ค่าพารามิเตอร์โดยวิธีการเส้นโค้งปฏิบัติการกระบวนการ

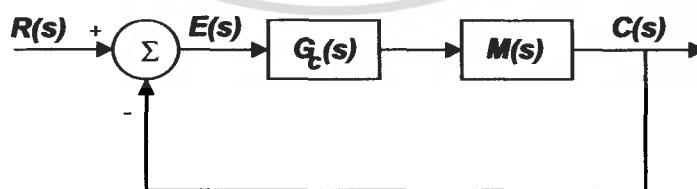
ชนิดของตัวควบคุม	โพรบโพชั่นแน็ล แกน $K'_C$	เวลา อินทิกรัล $T'_I$	เวลา เดริเวทิฟ $T'_D$
โพรบโพชั่นแน็ล, พี	$\frac{1}{K} \left( \frac{t_0}{\tau} \right)^{-1}$	-	-
โพรบโพชั่นแน็ล-อินทิกรัล, พีไอ	$\frac{0.9}{K} \left( \frac{t_0}{\tau} \right)^{-1}$	$3.33t_0$	-
โพรบโพชั่นแน็ล-อินทิกรัล- เดริเวทิฟ, พีไอดี	$\frac{1.2}{K} \left( \frac{t_0}{\tau} \right)^{-1}$	$2.0t_0$	$\frac{1}{2}t_0$

ซึ่งวิธีการนี้จะสามารถใช้กับกระบวนการที่ไม่มีตำแหน่งโพลอยู่ที่จุดกำเนิดหรือตำแหน่งของโพลเชิงซ้อน ซึ่งจะไม่มีผลตอบสนองเป็นรูปตัว S และอัตราส่วนของ  $\frac{t_0}{\tau}$  จะต้องมีค่าอยู่ในช่วง 0.1 - 0.5

#### 2.7.4 การสังเคราะห์ค่าพารามิเตอร์ของการควบคุมแบบพีไอดีด้วยวิธีคาสลิน

การหาค่าคุณลักษณะของกระบวนการโดยการทดสอบดังที่กล่าวมา จะพิจารณาในลักษณะวงเปิดซึ่งต่างจากการสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุมพีไอดี ที่ทำการพิจารณาในลักษณะการควบคุมแบบวงเปิด เนื่องจากต้องพิจารณารวมไปถึงตัวควบคุมด้วย จากฟังก์ชันถ่ายโอนแบบวงปิด(Closed loop) คือ

$$\frac{C(s)}{M(s)} = \frac{G_c(s)G(s)}{1+G_c(s)G(s)}$$



รูปที่ 2.46 ระบบควบคุมวงปิด

จัดสมการให้เป็นฟังก์ชันถ่ายโอนของตัวควบคุมได้ คือ

$$G_c(s) = \frac{1}{G(s)} \times \frac{C(s)/R(s)}{1 - [C(s)/R(s)]}$$

ฟังก์ชันถ่ายโอนนี้จะใช้สำหรับสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุมต่อไป พิจารณาผลตอบสนองแบบวงปิดเป็นผลตอบสนองแบบอันดับหนึ่งที่มีเวลาไร้ผลตอบสนองเท่ากับที่มีฟังก์ชันถ่ายโอนคือ

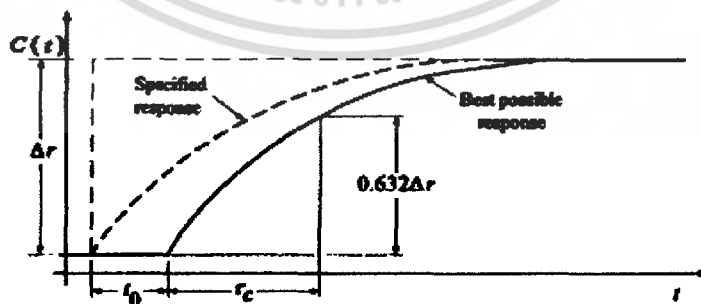
$$\frac{C(s)}{R(s)} = \frac{e^{-t_0 s}}{\tau_c s + 1}$$

ฟังก์ชันถ่ายโอนสำหรับสังเคราะห์ค่าพารามิเตอร์ตัวควบคุมตามกระบวนการแบบ FOPDT คือ

$$G_c = \frac{\tau s + 1}{K e^{-t_0 s}} \times \frac{e^{-t_0 s}}{\tau_c s + 1 - e^{-t_0 s}}$$

หรือ

$$G_c = \frac{\tau s + 1}{K} \times \frac{1}{\tau_c s + 1 - e^{-t_0 s}}$$



รูปที่ 2.47 แสดงผลตอบสนองแบบปิดที่มีการหน่วงเวลาเป็นเวลา  $t_0$

ถึงแม้ว่าตัวควบคุมชนิดนี้มีความเป็นไปได้ในทางทฤษฎี แต่อย่างไรก็ตามการทำให้เกิดผลในทางปฏิบัตินั้นเป็นไปได้ยาก สาเหตุที่สำคัญที่สุดคือตัวควบคุมพีไอดี แบบดั้งเดิมนั้นมีการทำงานในแบบอนาล็อกและพจน์  $e^{-t_0s}$  ไม่สามารถทำให้เกิดผลได้จริงในทางปฏิบัติด้วยอุปกรณ์ทางอนาล็อก แต่ตัวควบคุมพีไอดี ในปัจจุบันประกอบขึ้นด้วยไมโครโปรเซสเซอร์และดิจิทัลคอมพิวเตอร์จึงทำให้พจน์ของการหน่วงเวลาเกิดผลได้จริง ซึ่งเมื่อมีการทำงานในพจน์นี้เรียกว่า พจน์นี้จะเรียกว่า พจน์การชดเชยการหน่วงเวลา (Dead-time compensation)

เทอมเอ็กซ์โพเนนเชียล ( $e^{-t_0s}$ ) สามารถประมาณค่าได้โดย First-order Pade approximation คือ

$$e^{-t_0s} = 1 - t_0s + \frac{1}{2!}(t_0s)^2 - \frac{1}{3!}(t_0s)^3 + \dots$$

ใช้เฉพาะ 2 เทอมแรก คือ  $1 - t_0s$  (สำหรับกระบวนการอันดับหนึ่ง) แทนในสมการจะได้

$$G_c(s) = \frac{\tau s + 1}{K} \times \frac{1}{(\tau_c + t_0)s}$$

หรือ

$$G_c(s) = \frac{\tau}{(\tau_c + t_0)K} \times \left[ 1 + \frac{1}{\tau s} \right]$$

จากสมการ แสดงให้เห็นว่า จากกระบวนการที่เป็นแบบ FOPDT เมื่อนำมาหาฟังก์ชันถ่ายโอนของตัวควบคุมจะได้เป็นตัวควบคุมแบบพีไอเท่านั้น โดยมีค่าพารามิเตอร์ คือ

$$K_c = \frac{\tau}{K(\tau_o + t_0)} \quad \text{และ} \quad T_i = \tau$$

แต่จากการสังเคราะห์หาค่าพารามิเตอร์ของตัวควบคุมพีไอดี แบบวงปิดของค่าหาลิม ตามกระบวนการ FOPDT สามารถทำเป็นตัวควบคุมแบบพีไอดี ที่มีค่าพารามิเตอร์ของตัวควบคุมคือ

$$K_c = \frac{\tau}{K(\tau_o + t_0)}, T_i = \tau \quad \text{และ} \quad T_d = \frac{t_0}{2}$$

ตารางที่ 2.4 แสดงสูตรสำหรับหาค่าพารามิเตอร์ของตัวควบคุมแบบพีไอดีตามวิธีค่าหัดลิน

ชนิดของกระบวนการ	ชนิดของตัวควบคุม	ค่าพารามิเตอร์
$G(s) = K$	ไอ	$K_c = \frac{1}{K\tau_c}$
$G(s) = \frac{K}{\tau s + 1}$	พีไอ	$K_c = \frac{\tau}{K\tau_c}$ $\tau_i = \tau$
$G(s) = \frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)}, \tau_1 > \tau_2$	พีไอดี	$K'_c = \frac{\tau_1}{K\tau_c}$ $\tau'_i = \tau_1$ $\tau'_D = \tau_2$
$G(s) = \frac{Ke^{-t_0 s}}{\tau s + 1}$	พีไอดี <sup>a</sup>	$K'_c = \frac{\tau}{K(t_0 + \tau_c)}$ $\tau'_i = \tau$ $\tau'_D = \frac{\tau_0}{2}$
$G(s) = \frac{K}{s}$	พี	$K_c = \frac{1}{K\tau_c}$

<sup>a</sup> ใช้ได้ทั้งตัวควบคุมแบบพีไอดี และพีไอ ( $T_d = 0$ ) โดยแบบพีไอดี ใช้ได้เมื่อค่า  $t_0$  มากกว่า  $\tau/4$

ข้อแนะนำ สูตรสังเคราะห์หาค่าพารามิเตอร์ของตัวควบคุมของคาลินห์ ในโหมดพีไอดี ควรใช้ เมื่อค่า  $t_0$  มากกว่า  $\tau/4$  กรณีนอกเหนือจากนี้ ควรใช้การควบคุมแบบพีไอ และสำหรับความต้องการควบคุมให้ค่าพุ่งเกินมีค่าไม่เกิน 5% สำหรับการเปลี่ยนค่าเป้าหมายแนะนำให้ใช้  $\tau_c = t_0$  ดังนั้นที่ความต้องการค่าพุ่งเกิน (Overshoot) 5%

$$K_c = \frac{\tau}{K(t_0 + t_0)} = \frac{0.5}{K} \left[ \frac{\tau}{t_0} \right]$$

## 2.8 หลักการเขียนโปรแกรมวิซวล เบสิก 6 (Visual Basic 6)

วิซวล เบสิก ถือได้ว่าเป็นเครื่องมือที่ใช้เขียนโปรแกรมบน วินโดวส์ (Windows) ที่ได้รับความนิยมสูงสุด ทั้งนี้เพราะผ่านการพัฒนาอย่างต่อเนื่องจากเวอร์ชันแรกที่ทำงานบนดอส (DOS) แล้วมาได้รับความนิยมในเวอร์ชัน 3.0 ที่ทำงานบน วินโดวส์ 3.1 จนก้าวมาถึงเวอร์ชันล่าสุดคือ เวอร์ชัน 6.0 ที่ได้รับความนิยมทั้งเมืองไทยและทั่วโลก

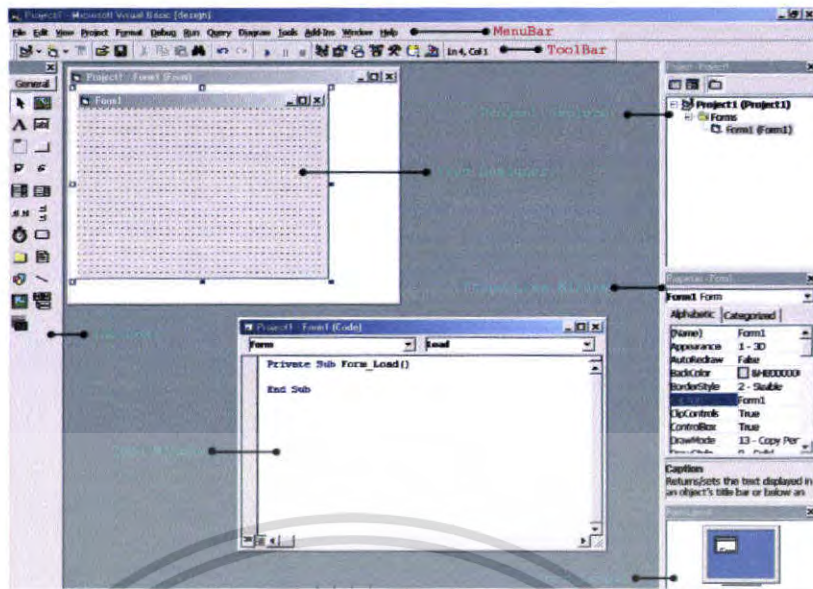
### 2.8.1 จุดเด่นของวิซวล เบสิก

สร้างแอปพลิเคชันได้ง่ายและรวดเร็ววิซวล เบสิก ได้รับการวางตัวเป็นเครื่องมือที่ช่วยให้สร้างแอปพลิเคชันได้อย่างรวดเร็วและง่าย เพื่อลดเวลาการสร้างแอปพลิเคชันให้สั้นลง ทั้งนี้เพราะมีการจัดงานที่ต้องทำซ้ำๆ ออกไป และขจัดสิ่งไม่จำเป็นต้องรู้เกี่ยวกับการควบคุมฮาร์ดแวร์ การจัดการภายในของ วินโดว์ ออกไป เหลือเฉพาะการให้ความสำคัญกับปัญหาของงานจริงๆ เท่านั้น ภาษาเขียนโปรแกรมที่ง่ายต่อการเรียนรู้วิซวล เบสิก นั้นจะใช้ภาษาเบสิก (Basic) ในการเขียนโปรแกรมซึ่งทำให้อ่านโค้ดแล้วมีความใกล้เคียงกับภาษาที่เราใช้งานปกติอีกทั้งยังสื่อความหมายเข้าใจได้ง่ายกว่าภาษาโปรแกรมอื่นๆ

รวมเครื่องมืออำนวยความสะดวกในการเขียนโปรแกรม นอกเหนือจากจะง่ายต่อการเรียนรู้แล้ววิซวล เบสิกยังมีเครื่องมือที่ช่วยให้การเขียน โปรแกรมเป็นเรื่องที่ไม่ยุ่งยากเพราะมีเครื่องมือช่วยให้ไม่ต้องจำไวยากรณ์ภาษาที่ยุ่งยาก ตรวจสอบข้อผิดพลาดว่าโปรแกรมที่เขียนนั้นถูกต้องตามหลักภาษาหรือไม่ มีการแยกแยะส่วนของโปรแกรมอย่างเป็นระเบียบ ทำให้งานของโปรแกรมเมอร์ลดลงได้มากนอกจากจะมีเครื่องมือช่วยในการเขียน โปรแกรมแล้วยังมีเครื่องมือที่ใช้ทดสอบแก้ไข โปรแกรม (Debugger) ที่เขียนขึ้นมาว่าทำงานได้ถูกต้องหรือไม่ มีระบบขอความช่วยเหลือ (Online Help) ไว้อ้างอิง และขอความช่วยเหลือในจุดที่เราสงสัยข้อข้องใจเครื่องมือทั้งหมดที่กล่าวมาถูกจัดรวมไว้ในสภาพแวดล้อมการทำงานเดียวกัน (เรียกย่อๆ ว่า ไอดีอี, IDE ซึ่งมาจาก Integrate Development Environment) ทำให้เรียกใช้งานได้สะดวกตั้งแต่เขียนโปรแกรม ทดสอบ แก้ไข สร้างชุดติดตั้งรวมทั้งระบบขอความช่วยเหลือ ซึ่งเราสามารถเพิ่มเติมเครื่องมือชนิดใหม่ๆ เข้าไปได้เรื่อยๆ

### 2.8.2 องค์ประกอบต่างๆ ของวิซวล เบสิก 6.0

เมื่อเราเปิดใช้งานวิซวล เบสิก 6.0 ก็จะพบกับลักษณะการทำงานที่เรียกว่า ไอดีอี คือ รวบรวมเครื่องมือ เครื่องมือ ข้อมูลที่ใช้งานต่างๆ ไว้ในหน้าจอเดียว ทำให้เรียกใช้งานได้ง่าย

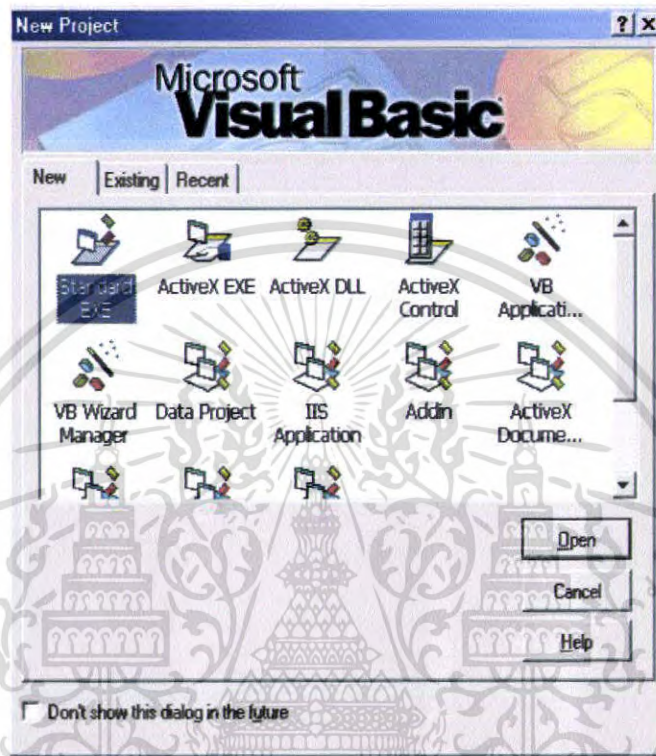


รูปที่ 2.48 แสดงองค์ประกอบต่างๆ ของวิซวล เบสิก

MenuBar	เมนูบาร์ เป็นส่วนที่รับคำสั่งในแบบเมนู เมื่อผู้ใช้ทำการสร้างแอปพลิเคชันด้วยวิซวล เบสิก เป็นเหมือนศูนย์กลางที่ควบคุมการสร้างแอปพลิเคชัน
ToolBar	ในการใช้งานเมนูบาร์สั่งงานอาจจะมีขั้นตอนที่ยุ่งยาก เพื่อลดขั้นตอนลง ผู้ใช้สามารถคลิกที่ทูลบาร์เพียงครั้งเดียว ก็สามารถสั่งงานที่เราต้องการ ได้ (เป็นเหมือนคีย์ลัดในการทำงาน)
ToolBox	ทูลบ็อกซ์ เป็นกล่องเก็บ ActiveX Control ซึ่งนำมาประกอบเป็นส่วนต่างๆ ของแอปพลิเคชัน
Project Explorer	เป็นเครื่องมือที่ใช้ควบคุมการทำงานของโปรเจกต์ (ซึ่งจะกล่าวถึงความหมายของโปรเจกต์ในหัวข้อต่อไป)
Properties Window	เป็นส่วนกำหนดพร็อพเพอร์ตี้ให้กับออบเจกต์ต่างๆ
Form Layout	ฟอร์มเลย์เอาต์ เป็นหน้าต่างราวๆ ของฟอร์มที่ได้จากการรันแอปพลิเคชันทำให้ผู้ใช้ทราบตำแหน่งของส่วนต่างๆ ที่จะปรากฏบนจอภาพเมื่อแอปพลิเคชันทำงาน
Form Designer	ฟอร์มดีไซเนอร์ เป็นส่วนที่มองเห็นในขณะออกแบบแอปพลิเคชันของวิซวล เบสิก ซึ่งผู้ใช้จะสามารถออกแบบหน้าต่างของแอปพลิเคชันผ่านฟอร์มดีไซเนอร์
Code Window	โค้ดวินโดว์ เป็นส่วนที่เราเขียนโปรแกรม (เรียกสั้นๆ ว่าเขียนโค้ด) เพื่อควบคุมการทำงานของแอปพลิเคชัน

### 2.8.3 รูปแบบการสร้างแอปพลิเคชันด้วยวิซวล เบสิก

เมื่อเปิดวิซวล เบสิก 6.0 ขึ้นมา จะพบกับไดอะล็อกบ็อกซ์ New Project ซึ่งแสดงประเภทของแอปพลิเคชันที่สร้างได้ สำหรับแอปพลิเคชันแต่ละประเภทที่สร้างได้นั้นมีความหมายดังตารางที่ 2.5



รูปที่ 2.49 ไดอะล็อก New Project

แอปพลิเคชันสำหรับใช้ในการสร้างโปรเจกใหม่

ตารางที่ 2.5 ตารางแสดงประเภทของแอปพลิเคชัน

ชนิดของแอปพลิเคชัน	คำอธิบาย
Standard EXE	เป็นแอปพลิเคชันทั่วไปที่มีการใช้ใน วินโดว์ เมื่อสร้างแล้วจะได้ไฟล์ที่มีนามสกุลเป็น .EXE
ActiveX EXE	เป็นการสร้างแอปพลิเคชันชนิดที่เรียกว่า out – of – process OLE
ActiveX DLL	เป็นการสร้างแอปพลิเคชันชนิดที่เรียกว่า in – process OLE server
ActiveX Control	เป็นการสร้าง ActiveX Control เมื่อสร้างแล้วจะได้ไฟล์นามสกุลเป็น .OCX

## ตารางที่ 2.5 (ต่อ) ตารางแสดงประเภทของแอปพลิเคชัน

ชนิดของแอปพลิเคชัน	คำอธิบาย
VB Application Wizard	เป็นการสร้างวิซาร์ดเพื่อใช้งานร่วมกับแอปพลิเคชัน ซึ่งมักจะเป็นแอปพลิเคชันที่มีความซับซ้อน จึงต้องมีวิซาร์ดช่วยลดความยุ่งยาก
VB Wizard Manager	เป็นเครื่องมือที่ใช้สร้างวิซาร์ด
Data Project	เป็นการสร้างแอปพลิเคชันเพื่อให้ทำงานร่วมกับ Data Object
DHTML Application	เป็นการสร้างแอปพลิเคชันอินเทอร์เน็ตในฝั่งไคลเอนท์
IIS Application	เป็นการสร้างแอปพลิเคชันอินเทอร์เน็ตในฝั่งเซิร์ฟเวอร์
Addin	เป็นเครื่องมือที่ช่วยในการสร้าง Add-in ซึ่งวิซวล เบสิก ให้สร้างเครื่องมือใช้เฉพาะกับงานของเราได้
ActiveX Document DLL	เป็นการสร้างแอปพลิเคชันชนิดที่เรียกว่า in- process ActiveX Document
ActiveX Document EXE	เป็นการสร้างแอปพลิเคชันชนิดที่เรียกว่า out-of- process ActiveX Document

ในไดอะล็อกบ็อกซ์ New Project ยังมีอีก 2 แท็บซึ่งมีความหมายดังนี้

แท็บ Existing แสดงโปรเจกต์ที่เคยมีการสร้างมาก่อน ซึ่งช่วยให้เราไม่ต้องเสียเวลาค้นหาไฟล์เคอร์ที่เก็บโปรเจกต์เดิมๆ

แท็บ Recent แสดงโปรเจกต์ที่เคยมีการสร้างมาก่อน และถูกเรียกมาแก้ไขล่าสุด

### ออบเจกต์ (Object) และ เมธอด (Method)

1. ออบเจกต์ คือการนำอุปกรณ์ต่างๆ ที่มีอยู่ในโปรแกรมมาประกอบกันเป็นแอปพลิเคชัน ในส่วนที่พบเห็นกันมากเป็นลักษณะ ของปุ่มกด ฟอรัม หรือปุ่มตัวเลือกต่างๆ โดยที่เรายังสามารถกำหนดพรีอเพอร์ตี้ต่างๆ ให้กับออบเจกต์ได้ที่ Properties Window

2. เมธอด คือ ความสามารถของออบเจกต์ ซึ่งจะถูกเรียกผ่านทางกรเขียน โปรแกรม

## การใช้เมธอดด้านกราฟฟิก

### 2.1 การลบกราฟฟิกด้วยเมธอด Cls

เมธอด Cls ใช้ในการลบกราฟฟิกที่วาด ใช้ได้กับ Form คอนโทรล Image หรือ คอนโทรล PictureBox โดยจะมีรูปแบบคำสั่งดังนี้

<ชื่อคอนโทรล>.Cls

เมธอดนี้มีประโยชน์ เมื่อต้องการลบกราฟฟิกที่วาดด้วยเมธอดอื่น เพื่อวาดกราฟฟิกใหม่

### 2.2 การวาดจุดกราฟฟิกด้วยเมธอด Pset

เมธอดPset ใช้ในการวาดจุดภาพ (เรียกว่า Pixel) ในฟอร์ม คอนโทรล Image หรือ คอนโทรล PictureBox โดยจะมีรูปแบบคำสั่งดังนี้

<ชื่อคอนโทรล>.Pset (x,y),[Color]

โดยที่ x,y คือ ตำแหน่งที่จะวาดจุดภายในของคอนโทรลนั้น

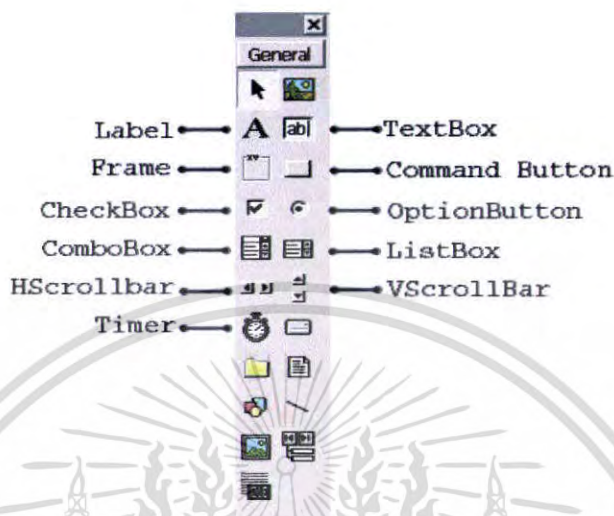
Color คือ เป็นสีของจุดที่วาด ถ้าไม่มีการเซตค่าจะใช้สีตาม คุณสมบัติ ForeColor ของ คอนโทรลนั้นแทน

### 2.8.4 ขั้นตอนการสร้างแอปพลิเคชันด้วยวิซวล เบสิก

1. ออกแบบแอปพลิเคชัน
2. ตกแต่งหน้าต่างแอปพลิเคชัน
3. เขียนโค้ดกำกับการทำงานของแอปพลิเคชัน
4. ทดสอบการทำงานของแอปพลิเคชัน
5. บันทึกเก็บไว้ในคอมพิวเตอร์
6. การสร้างไฟล์ .EXE (Make)

### 2.8.5 ActiveX Control เบื้องต้น

ActiveX Control ถูกสร้างขึ้นมาเพื่อที่ใช้ในสร้างแอปพลิเคชันให้ง่ายขึ้น โดยมี ส่วนประกอบพื้นฐานที่ควรรู้จักมีดังนี้

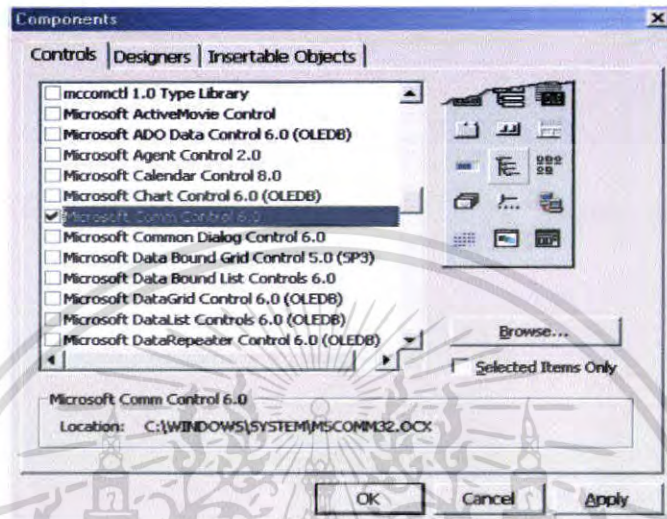


รูปที่ 2.50 แสดง ActiveX Control พื้นฐานที่ปรากฏใน Toolbox

ตารางที่ 2.6 แสดงหน้าที่ของ ActiveX Control แต่ละแบบ

ActiveX Control	คำอธิบาย
Label	เป็นแถบข้อความ มักใช้เขียนข้อความให้อ่านอย่างเดียว
TextBox	เป็นช่องให้ผู้ใช้งานกรอกข้อความ
Frame	เป็นกรอบที่จัดกลุ่ม คอนโทรล ต่างๆ ออกเป็นพวกๆ
Command Button	เป็นปุ่มกดให้ผู้ใช้งานกด <Enter> หรือคลิกที่ปุ่มนี้
CheckBox	เป็นปุ่มให้ผู้ใช้งานคลิกเลือก ซึ่งจะเลือกได้กี่ตัวก็ได้
OptionButton	เป็นปุ่มให้ผู้ใช้งานคลิกเลือก ซึ่งเลือกได้เพียง 1 ตัวเท่านั้น
ComboBox	เป็นรายการข้อมูลให้ผู้ใช้งานเลือก
ListBox	เป็นรายการข้อมูลให้ผู้ใช้งานเลือก
HScrollBar	เป็นแถบเลื่อนตามแนวนอน
VScrollBar	เป็นแถบเลื่อนตามแนวตั้ง
Timer	เป็นตัวจับเวลา

นอกเหนือจากที่กล่าวมาแล้วยังมี ActiveX Control ต่างๆ ที่ถูกซ่อน ซึ่งหนึ่งในนั้นที่มีความสำคัญมากที่สุดที่ใช้ในโครงการนี้ก็คือ Microsoft Comm Control 6.0 ซึ่งเป็น ActiveX Control ที่ใช้เกี่ยวกับการติดต่อกับพอร์ตอนุกรม โดยสามารถเพิ่มเติมเข้ามาไว้ใน Toolbox โดยเลือกเมนู Project > Components... หรือคลิกขวาที่ Toolbox แล้วเลือก Component... จะปรากฏหน้าต่างดังรูปที่ 2.51



รูปที่ 2.51 แสดงการเพิ่ม Microsoft Comm Control 6.0 ใช้งานกับพอร์ตอนุกรม

## 2.8.6 การเชื่อมต่อแบบอนุกรม

### 2.8.6.1 คอลโทรลที่ใช้ติดต่อ (คอนโทรล MSComm)

สำหรับการใช้งานวิซวล เบสิก ตั้งแต่เวอร์ชัน 2 เป็นต้นมา ในวิซวล เบสิก จะมีคัสตอมคอนโทรลสำหรับการสื่อสารอนุกรมผ่านทางพอร์ตอนุกรมของคอมพิวเตอร์มาให้ โดยในวิซวล เบสิก เวอร์ชัน 2 และเวอร์ชัน 3 จะใช้ชื่อว่า MSCOMM.VBX ส่วนเวอร์ชัน 4 ใช้ชื่อว่า MSCOMM16.OCX สำหรับการทำงานกับระบบปฏิบัติการ 16 บิต และ MSCOMM32.OCX สำหรับการทำงานกับระบบปฏิบัติการ 32 บิต สำหรับในวิซวล เบสิก เวอร์ชัน 5 จะมีเพียง MSCOMM32.OCX เท่านั้นเพราะถูกออกแบบมาให้ใช้งานกับระบบปฏิบัติการ 32 บิต

MSComm จัดเตรียมทางเลือกเอาไว้ 2 ทางเพื่อความสะดวกในการสื่อสารข้อมูล ทางแรกคือ การสื่อสารข้อมูลที่กระตุ้นด้วยเหตุการณ์ (event-driven communication) เป็นรูปแบบการใช้งานที่มีประสิทธิภาพมากสำหรับการตอบสนองแบบทันทีทันใด เช่น เมื่อตัวอักษรถูกส่งมาที่พอร์ตอนุกรมหรือเกิดการเปลี่ยนแปลงที่ Data Carrier Detect (DCD) หรือขา Request To Send (RTS) เหตุการณ์ Oncomm ของ MSComm จะสามารถตรวจจับสัญญาณนั้นได้ทันที ซึ่งจะกล่าวถึงรายละเอียดในหัวข้อคุณสมบัติ CommEvent ต่อไป ส่วนทางเลือกที่สองเป็นการคอยตรวจสอบค่า

เหตุการณ์และความผิดพลาดที่เกิดขึ้นด้วยการค่าที่เปลี่ยนแปลงภายในคุณสมบัติ CommEvent หลังจากให้โปรแกรมทำงานในฟังก์ชันต่างๆ ไปเรียบร้อยแล้ว ซึ่งวิธีนี้ใช้งานได้ดีในกรณีที่โปรแกรมมีขนาดเล็ก

คอนโทรล MSComm 1 ตัวสามารถควบคุมการทำงานของพอร์ตอนุกรมได้ 1 พอร์ต ถ้าในโปรแกรมที่ใช้งานต้องการติดต่อกับพอร์ตอนุกรมมากกว่า 1 พอร์ตจะต้องใช้คอนโทรล MSComm มากกว่า 1 ตัวเพื่อควบคุมพอร์ตอนุกรมในแต่ละพอร์ต แอแดปเตอร์ของพอร์ตอนุกรมและแอแดปเตอร์ของการเกิดอินเตอร์รัปต์สามารถเปลี่ยนแปลงได้จากการแก้ไขค่าที่ Control Panel

### 2.8.6.2 คุณสมบัติ (property) ของ MSComm

#### 1. CommPort

ใช้ในการกำหนดและอ่านค่าพอร์ตอนุกรมที่ติดต่อยู่ (COM1, COM2, COM3, COM4)

รูปแบบการใช้งาน

```
object.CommPort[ ] = value
```

โดย Value เป็นค่าของพอร์ตอนุกรม ชนิดของข้อมูลเป็น Integer ค่า Value สามารถกำหนดได้ในช่วง 1-16 (ค่าเริ่มต้นกำหนดไว้ที่ 1) เมื่อมีการกำหนดค่าแล้วทำการเปิดพอร์ตโดยใช้คุณสมบัติ PortOpen แต่ถ้าพอร์ตนั้นไม่มีอยู่ในระบบ MSComm จะสร้างสัญญาณแสดงข้อผิดพลาด error 68 ขึ้นมา ซึ่งหมายถึง อุปกรณ์ตัวนี้ไม่มีอยู่ระบบ ดังนั้นการเขียนโปรแกรมจึงจำเป็นต้องกำหนดตำแหน่งของพอร์ตอนุกรมก่อนที่ใช้คำสั่ง OpenPort

#### 2. Setting

ใช้ในการกำหนดและอ่านค่าอัตราบอด, พาริตี, จำนวนของบิตข้อมูล, จำนวนของบิตปิดท้าย

รูปแบบการใช้งาน

```
object.Settings [ ] = value
```

ค่า Value มีชนิดข้อมูลเป็นแบบ String มีรูปแบบเป็น “BBBB,P,D,S” โดย BBBB เป็นค่าอัตราบอด P เป็นค่าพาริตี D เป็นจำนวนของบิตข้อมูล และ S เป็นจำนวนของบิตปิดท้าย ปกติแล้วค่านี้ถูกกำหนดไว้เป็น “9600, N, 8, 1”

ค่าอัตราบอดมาตรฐานที่ใช้กับ MSComm มีดังนี้

110	บิตต่อวินาที
300	บิตต่อวินาที
600	บิตต่อวินาที
1,200	บิตต่อวินาที
2,400	บิตต่อวินาที
9,600	บิตต่อวินาที (ค่าปกติ)
14,400	บิตต่อวินาที
19,200	บิตต่อวินาที
28,800	บิตต่อวินาที
38,400	บิตต่อวินาที (สงวน)
56,000	บิตต่อวินาที (สงวน)
128,000	บิตต่อวินาที (สงวน)
256,000	บิตต่อวินาที (สงวน)

สำหรับค่ามาตรฐานในการกำหนดค่าพาริตีมีดังนี้

สัญลักษณ์	รายละเอียด
E	พาริตีคู่ (Even)
M	ลอจิก "1" (Mark)
N	ไม่ใช่ (ค่าปกติ)
O	พาริตีคี่ (Odd)
S	ลอจิก "0" (Space)

ค่าที่ใช้ในการกำหนดจำนวนบิตมี 5 ค่าคือ 4, 5, 6, 7 และ 8 (เป็นค่าปกติ)

ค่าที่ระบุจำนวนบิตปิดท้ายมี 3 ค่าคือ 1 (เป็นค่าปกติ), 1.5 และ 2

ตัวอย่างการใช้งานคำสั่ง Settings โดยจะเป็นการกำหนดค่าอัตราบอดเท่ากับ 9600 ไม่มีพาริตี จำนวนบิตข้อมูล 8 บิต และ บิตปิดท้าย 1 บิต สามารถเขียน โปรแกรมได้ดังนี้

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

หมายเหตุ สาเหตุที่ค่าที่กำหนดจะต้องอยู่ภายในเครื่องหมายอัญประกาศ "" เนื่องจาก ค่าที่กำหนดนี้อยู่ในรูปตัวแปร String

### 3. PortOpen

ใช้ในการกำหนดและอ่านค่าสถานะของพอร์ตอนุกรม เพื่อเปิดและปิดพอร์ตอนุกรม  
รูปแบบการใช้งาน

```
object.PortOpen [ = value ]
```

ค่า Value มีชนิดข้อมูลเป็นแบบบูลีนคือ True กับ False โดย True หมายถึงการเปิดพอร์ตอนุกรมและ False หมายถึงการปิดพอร์ตอนุกรม สำหรับการปิดพอร์ตนั้นจะมีการเคลียร์บัฟเฟอร์รับข้อมูลและบัฟเฟอร์ส่งข้อมูลด้วย คอนโทรล MSComm จะปิดพอร์ตอนุกรมโดยอัตโนมัติเมื่อออกจากโปรแกรม ก่อนที่จะใช้คุณสมบัติ PortOpen ต้องตรวจสอบให้แน่ใจก่อนว่าคุณสมบัติ CommPort นั้นได้ทำการกำหนดตำแหน่งของพอร์ตอนุกรมไว้ถูกต้องหรือไม่ มิเช่นนั้น MSComm จะแสดงข้อผิดพลาด Error 68 แจ้งแก่ผู้ใช้งาน หรือถ้าพอร์ตอนุกรมนั้นถูกเปิดเอาไว้แล้ว โปรแกรมก็จะแจ้งข้อผิดพลาดออกมาเช่นเดียวกัน

ถ้าคุณสมบัติ DTREnable หรือ RTSEnable ถูกกำหนดให้เป็น True ก่อนที่จะทำการเปิดพอร์ต ค่าคุณสมบัติของ DTREnable หรือ RTSEnable จะถูกเซตเป็น False หลังจากปิดพอร์ต แต่ถ้าเซตเป็น False หลังจากปิดโปรแกรมแล้ว ค่าที่กำหนดไว้จะเป็นค่าเดิม

ตัวอย่างการใช้คำสั่งเปิดพอร์ต เพื่อติดต่อสื่อสารกับพอร์ตอนุกรม COM1 และมีอัตราบอด 9,600 บิตต่อวินาที ไม่มีพาริตี จำนวนบิตข้อมูล 8 บิต และบิตปิดท้าย 1 บิต มีดังนี้

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

```
MSComm1.CommPort = 1
```

```
MSComm1.PortOpen = True
```

### 4. Input

อ่านค่าและลบค่าขบวนข้อมูลจากบัฟเฟอร์ภาครับ  
รูปแบบการใช้งาน

```
object.Input
```

คุณสมบัติ InputLen เป็นตัวกำหนดจำนวนของตัวอักษรที่จะอ่าน โดยคุณสมบัติ Input การกำหนดค่าให้ InputLen เท่ากับ 0 เป็นการกำหนดให้คุณสมบัติ Input ทำการอ่านค่าข้อมูลในบัฟเฟอร์รับข้อมูลทั้งหมด

คุณสมบัติ InputMode เป็นตัวกำหนดชนิดของข้อมูลที่คุณสมบัติ Input รับเข้ามา ถ้า InputMode ถูกกำหนดเป็น comInputModeText คุณสมบัติ Input จะส่งค่าข้อมูลกลับมาในรูปแบบของข้อความชนิดข้อมูลเป็นแบบ Variant ถ้า InputMode กำหนดเป็น comInputModeBinary คุณสมบัติ Input จะส่งข้อมูลกลับมาในรูปแบบของไบนารีและชนิดข้อมูลเป็นแบบ Variant

ตัวอย่าง โปรแกรมแสดงให้เห็นถึงวิธีการรับข้อมูลจากบัพเฟอร์รับข้อมูลทั้งหมด

```
Private Sub Command1_Click()
```

```
Dim InString as String
```

```
    MSComm1.InputLen = 0           ' Retrieve all available data.
```

```
    If MSComm1.InBufferCount Then ' Check for data.
```

```
        InString = MSComm1.Input   ' Read data
```

```
    End If
```

```
End Sub
```

#### 5. InBufferCount

ส่งค่าจำนวนของตัวอักษรที่อยู่ในบัพเฟอร์ภาครับ  
รูปแบบการใช้งานคำสั่ง

```
object.InBufferCount [ = value ]
```

คำสั่ง InBufferCount จะแสดงค่าจำนวนของตัวอักษร ซึ่งรับมาจากภายนอกและยังเก็บอยู่ในบัพเฟอร์ภาครับ เพื่อให้ผู้ใช้งานอ่านค่าออกไป สำหรับการเคลียร์ค่าบัพเฟอร์ภาครับทำได้โดยกำหนดให้ InBufferCount มีค่าเป็น 0

หมายเหตุ อย่าสับสนระหว่างคำสั่ง InBufferSize และ InBufferCount คำสั่ง InBufferSize นั้นใช้เพื่อกำหนดขนาดของบัพเฟอร์ภาครับ

#### 6. InBufferSize

กำหนดและคืนค่าขนาดของบัพเฟอร์ภาครับในหน่วยเป็นไบต์  
รูปแบบการใช้งานคำสั่ง

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

คำสั่ง `InBufferSize` ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ ค่าเริ่มต้นกำหนดไว้ที่ 1,024 ไบต์

หมายเหตุ การกำหนดค่าบัฟเฟอร์ภาครับขนาดใหญ่จะทำให้ หน่วยความจำที่เหลือสำหรับการใช้งานส่วนอื่นๆ จะเหลือน้อย อย่างไรก็ตามการกำหนดค่า บัฟเฟอร์ภาครับที่น้อยเกินไปจะทำให้เกิดการ โอเวอร์โฟลวหรือข้อมูลล้นบัฟเฟอร์ เว้นแต่จะมีการใช้แฮนด์เช็ก ดังนั้นค่าปานกลางที่เหมาะสมก็คือค่า 1,024 ซึ่งเป็นค่าเริ่มต้นนั่นเอง แต่ถ้าโปรแกรมมีการเกิดโอเวอร์โฟลวแล้วจึงค่อยปรับเพิ่มค่าขนาดของบัฟเฟอร์ให้มีค่ามากขึ้น

## 7. InputLen

กำหนดค่าและคืนค่าจำนวนของตัวอักษรที่อ่านจากบัฟเฟอร์ภาครับ  
รูปแบบการใช้งานคำสั่ง

```
object.InputLen [ = value ]
```

ค่าเริ่มต้นของคุณสมบัติ `InputLen` มีค่าเท่ากับ “0” การกำหนดค่าเท่ากับ “0” จะทำให้ คำสั่ง `Input` ของ `MSComm` อ่านค่าข้อมูลที่อยู่ภายในบัฟเฟอร์ภาครับทั้งหมด

ถ้าไม่มีข้อมูลอยู่ในบัฟเฟอร์ภาครับมากเท่ากับจำนวน `InputLen` คำสั่ง `Input` จะส่งค่าว่าง (“”) กลับออกมา ผู้ใช้งานสามารถตรวจสอบข้อมูลในข้อมูลในบัฟเฟอร์ภาครับได้โดยใช้คุณสมบัติ `InBufferCount` โดยกำหนดให้มีข้อมูลอยู่ในบัฟเฟอร์ภาครับก่อนแล้วจึงค่อยอ่านข้อมูลจากบัฟเฟอร์ภาครับ

คุณสมบัตินี้มักใช้กับการอ่านค่าข้อมูลจากเครื่องมือหรือเครื่องจักรที่มีการกำหนดค่าขนาดความยาวของข้อมูลเอาไว้แล้ว

ตัวอย่าง โปรแกรมการอ่านค่าตัวอักษรออกมา 10 ตัวอักษร

```
Private Command1_Click()
Dim CommData as String
MSComm1.InputLen = 10           'Specify a 10 character block of data.
CommData = MSComm1.Input       'Read data.
End Sub
```

## 8. InputMode

กำหนดค่าและคืนค่าชนิดของข้อมูลที่ได้รับ โดยคำสั่ง Input  
รูปแบบการใช้งานคำสั่ง

```
object.InputMode [ = value ]
```

คุณสมบัติ InputMode ใช้กำหนดว่าข้อมูลชนิดไหนที่รับเข้ามาผ่านคำสั่ง Input โดยข้อมูล  
จะเลือกได้ 2 ประเภทคือ

comInputModeText สำหรับข้อมูลที่อยู่ในรูปข้อความตัวอักษรตามมาตรฐาน ANSI โดย  
จะต้องกำหนดค่าเป็น "0" และค่าเริ่มต้นของการรับค่าข้อมูลก็จะเป็นค่านี้

comInputModeBinary สำหรับข้อมูลอื่นๆ ซึ่งจะเก็บในรูปแบบไบนารีรวมกันอยู่เป็น ไบต์ข้อมูล

ตัวอย่างการใช้งาน InputMode ต่อไปนี้จะทำการอ่านค่าข้อมูล 10 ไบต์จากพอร์ตอนุกรม  
และเก็บข้อมูลไว้ในตัวแปรแบบอาร์เรย์ ชนิดข้อมูลเป็นแบบ ไบต์

```
Private Sub Command1_Click()
```

```
Dim Buffer as Variant
```

```
Dim Arr() as Byte
```

```
MSComm1.CommProt = 1
```

```
MSComm1.PortOpen = True
```

```
MSComm1.InputMode = comInputModeBinary
```

```
Binary data
```

```
Do Until MSComm1.InBufferCount < 10
```

```
Input buffer
```

```
DoEvents
```

```
Loop
```

```
Buffer = MSComm1.Input
```

```
Arr =Buffer
```

```
End Sub
```

## 9. Output

ใช้ในการส่งขบวนของข้อมูลไปยังบัพเฟอร์ส่งข้อมูล

รูปแบบการใช้งาน

```
object.Output [           = value ]
```

ค่า value เป็นค่าของตัวอักษรที่เขียนไปยังบัพเฟอร์ส่งข้อมูล คุณสมบัติ Output สามารถใช้ในการส่งข้อมูลตัวอักษรหรือข้อมูลไบนารีก็ได้ โดยการส่งข้อมูลเป็นรูปแบบตัวอักษรจะต้องกำหนดข้อมูลเป็นแบบ Variant และมีข้อมูลภายในเป็นแบบ String สำหรับการส่งข้อมูลไบนารีจะต้องกำหนดชนิดของข้อมูลเป็นแบบ Variant และมีข้อมูลภายในเป็นแบบ Byte

ตัวอย่างโปรแกรมการส่งค่าที่ป้อนจากคีย์บอร์ดไปยังพอร์ตอนุกรม โดยใช้คุณสมบัติ

Output

```
Private Sub Form_KeyPress (KeyAscii As Integer)
```

```
Dim Buffer as Variant
```

```
MSComm1.CommPort = 1
```

```
MSComm1.PortOpen = True
```

```
Buffer = Chr$(KeyAscii)
```

```
MSComm1.Output = Buffer
```

```
End Sub
```

## 10. OutBufferCount

คืนค่าจำนวนของข้อมูลตัวอักษรที่เก็บอยู่ในบัพเฟอร์ภาคส่ง และสามารถใช้อำนาจนี้เพื่อเคลียร์บัพเฟอร์ภาคส่งได้ด้วย

รูปแบบการใช้งานคำสั่ง

```
object.OutBufferCount [           = value ]
```

ผู้ใช้งานสามารถเคลียร์บัพเฟอร์ภาคส่งได้โดยการกำหนดค่า OutBufferCount เท่ากับ “0” หมายถึง ระวังการสับสนระหว่างคุณสมบัติ OutBufferCount กับ OutBufferSize ซึ่ง OutBufferSize ใช้เพื่อกำหนดขนาดของบัพเฟอร์ภาคส่ง

## 11. OutBufferSize

กำหนดค่าและคืนค่าขนาดของบัฟเฟอร์ภาคส่ง ชนิดตัวแปรเป็นแบบไบต์  
รูปแบบการใช้งานคำสั่ง

```
object.OutBufferSize [ = object ]
```

คุณสมบัติ OutBufferSize ใช้สำหรับกำหนดขนาดของบัฟเฟอร์ภาคส่ง โดยค่าปกติที่ใช้งานจะมีค่าเท่ากับ 512 ไบต์

หมายเหตุ การกำหนดค่าบัฟเฟอร์ภาคส่งที่มากเกินไปจะทำให้ มีหน่วยความจำเหลือให้ใช้งานน้อย แต่อย่างไรก็ตามถ้ากำหนดค่าน้อยเกินไป จะทำให้เกิดข้อมูลล้นบัฟเฟอร์ขึ้นได้ ยกเว้นจะมีการใช้ แชนด์เซ็ค วิธีการที่ถูกต้องในการกำหนดค่าคือ ทดลองใช้ค่าเริ่มต้นคือค่า 512 ไบต์ดูก่อนถ้าโปรแกรมทำงานแล้วเกิดการล้นของข้อมูลค่อยเพิ่มค่าของ OutBufferSize ให้มากขึ้น

## 12. ParityReplace

กำหนดและคืนค่าตัวอักษรที่ไปวางแทนในตำแหน่งที่เกิดข้อผิดพลาดจากพาริตี  
รูปแบบการใช้งานคำสั่ง

```
object.ParityReplace [ = value ]
```

บิตพาริตี เป็นบิตที่ทางภาคส่งข้อมูลทำการส่งมาพร้อมกับข้อมูล เพื่อตรวจสอบข้อผิดพลาดของข้อมูล โดยเมื่อมีการใช้บิตพาริตี คอนโทรล MSCOM จะทำการบอกบิตทุกบิตที่มีค่าลอจิก “1” ในแต่ละไบต์ และทำการตรวจสอบผลลัพธ์ว่าบิตที่อ่านได้นั้นมีจำนวนลอจิก “1” เป็นเลขคู่หรือคี่ และตรงกับค่าที่กำหนดไว้แต่ต้นหรือไม่ ถ้าค่าที่นำมาบวกแล้วมีพาริตีไม่ตรงแสดงว่าการรับส่งข้อมูลผิดพลาด

การกำหนดค่า เริ่มต้นให้กับ ParityReplace นั้นกำหนดให้ใช้เครื่องหมาย (?) ไปวางไว้ที่ตำแหน่งที่เกิดพาริตีผิดพลาด ถ้ากำหนดค่า ParityReplace ให้เป็นค่าว่าง (“”) จะเป็นการยกเลิกการใช้งาน ParityReplace และไม่มีป้อนข้อมูลแทนเมื่อตรวจพบข้อผิดพลาด

ParityReplace ใช้ชนิดข้อมูลเป็นแบบสตริง แต่จะการกำหนด จะกำหนดได้เพียงไบต์เดียวเท่านั้น ซึ่งจะสามารถใช้ค่าใดๆ ก็ได้ที่เป็น โค้ด ANSI มีค่าอยู่ระหว่าง 0-255

### 13. DTREnable

ใช้ในการกำหนดสถานะลอจิกของขา Data Terminal Ready (DTR) โดยสัญญาณของขา DTR จะส่งจากคอมพิวเตอร์ไปยัง โมเด็มเพื่อแสดงว่าคอมพิวเตอร์พร้อมที่จะรับข้อมูลแล้ว ชนิดของข้อมูลเป็นแบบบูลีน

รูปแบบการใช้งาน

```
object.DTREnable [ = value ]
```

ค่า Value เป็นค่าสถานะ True หรือ False เพื่อกำหนดลอจิกของขา DTR ให้เป็น “0” หรือ “1” โดย

True หมายถึง ให้ขา DTR มีลอจิก “1”

False หมายถึง ให้ขา DTR มีลอจิก “0” (ค่าปกติ)

หมายเหตุ เมื่อขา DTR ถูกกำหนดสถานะให้เป็น True ที่ขา DTR จะมีสถานะลอจิก “1” เมื่อทำการเปิดพอร์ตและจะมีสถานะเป็น “0” เมื่อมีการปิดพอร์ต เมื่อขา DTR ถูกกำหนดสถานะเป็น False ที่ขา DTR จะมีสถานะลอจิก “0” ตลอดเวลาไม่ว่าจะใช้คำสั่งเปิดพอร์ตหรือปิดพอร์ต

สำหรับการใช้งาน โมเด็ม การทำให้ขา DTR เป็นลอจิก “0” จะเป็นการวางหูโทรศัพท์หรือยกเลิกการติดต่อ

### 14. RTSEnable

ใช้เพื่อกำหนดสถานะลอจิกให้ขา Request To Send (RTS) โดยขา RTS จะเป็นสัญญาณที่ส่งจากคอมพิวเตอร์ไปยัง โมเด็มเพื่อร้องขอส่งข้อมูล ชนิดของข้อมูลเป็นแบบ Boolean

รูปแบบการใช้งาน

```
object.RTSEnable [ = value ]
```

ค่า Value เป็นค่าสถานะ True หรือ False เพื่อกำหนดลอจิก “0” หรือ “1” ให้ขา RTS โดย True หมายถึง ให้ขา RTS มีลอจิก “1”

False หมายถึง ให้ขา RTS มีลอจิก “0” (เป็นค่าปกติ)

หมายเหตุ เมื่อขา RTSEnable ถูกกำหนดให้เป็น True ขา RTS จะมีสถานะลอจิก “1” เมื่อเปิดพอร์ตและมีสถานะลอจิก “0” เมื่อปิดพอร์ต และเมื่อมีการกลับสถานะของขา RTS ขา TxD จะมีสถานะลอจิกเป็น “1” แต่ที่คุณสมบัติ Break ยังคงเป็นค่าเดิม

### 15. EOFEnable

เป็นการกำหนดให้ MSCComm รอสัญลักษณ์แสดงส่วนท้ายสุดของไฟล์ (End of file : EOF) ระหว่างการรับอินพุตเข้ามา ถ้าพบสัญลักษณ์ EOF ภาคอินพุตจะหยุดรับข้อมูล และเหตุการณ์ OnComm จะถูกกระตุ้นให้ทำงาน คุณสมบัติ CommEvent จะมีค่าเท่ากับ 7 หรือ ComEvEOF

รูปแบบการใช้งาน

```
object.EOFEnable [ = value ]
```

โดย value เป็นค่าสถานะ True หรือ False เพื่อเ็นเปิดหรือปิดการทำงานของ เหตุการณ์ OnComm เมื่อตรวจพบสัญลักษณ์ EOF โดย

True หมายถึง เหตุการณ์ OnComm จะถูกกระตุ้นให้ทำงาน EOF

False หมายถึง เหตุการณ์ OnComm จะไม่ถูกกระตุ้นให้ทำงานด้วย EOF (เป็นค่าปกติ)

เมื่อ EOFEnable กำหนดให้เป็น False ส่วนควบคุมจะ ไม่มีการตรวจสอบสัญลักษณ์ EOF

### 16. CTS Holding

ผู้ใช้งานสามารถตรวจสอบการทำงานของขา Clear To Send (CTS) ได้ว่ามีสถานะลอจิก “0” หรือ “1” โดยค่าที่อ่านได้จะเป็นบูลีน True และ False ถ้าค่า CTS Holding เป็น True ขา CTS จะมีสถานะลอจิกเป็น “1” ถ้าค่า CTS Holding เป็น False ขา CTS จะมีสถานะลอจิกเป็น “0”

รูปแบบการใช้งาน

```
object.CTSHolding
```

เมื่อขา CTS เป็นลอจิก “0” (CTSHolding = False) และเกิดไทม์เอาต์ คอนโทรล MSCComm จะกำหนดให้คุณสมบัติ CommEvent มีค่าเป็น comEventCTSTO (Clear To Send Timeout) และกระตุ้นให้เกิดเหตุการณ์ OnComm

### 17. CDHolding

ผู้ใช้งานสามารถตรวจสอบการทำงานของขา Data Carrier Detect (DCD) ได้ว่ามีสถานะ ลอจิกเป็น “1” หรือ “0” โดยค่าที่อ่านได้จะเป็นบูลีน True และ False ถ้าค่า CDHolding เป็น True ขา DCD จะมีสถานะลอจิก “1” ถ้าค่า CDHolding เป็น False ขา DCD จะมีสถานะลอจิก “0”

รูปแบบการใช้งาน

```
object.CDHolding
```

เมื่อขา DCD มีลอจิก “1” (CDHolding = True) และเกิดไทม์เอาต์ คอนโทรล MSComm จะกำหนดให้คุณสมบัติ CommEvent มีค่าเป็น comEventCDTO (Carrier Detect Timeout Error) และกระตุ้นให้เกิดเหตุการณ์ OnComm

#### 19. DSRHolding

ผู้ใช้งานสามารถตรวจสอบการทำงานของขา DSR ได้ว่ามีสถานะลอจิก “1” หรือ “0” โดยค่าที่อ่านได้จะเป็นบูลีน True และ False ถ้าค่า DSRHolding เป็น True ขา DSR จะมีสถานะลอจิก “1” ถ้าค่า DSRHolding เป็น False ขา DSR จะมีสถานะลอจิก “0”

รูปแบบการใช้งาน

object.DSRHolding

เมื่อขา DSR เป็นลอจิก “1” (DSRHolding = True) และเกิดไทม์เอาต์ คอนโทรล MSComm จะกำหนดให้คุณสมบัติ CommEvent มีค่าเป็น comEventDSRTO (Data Set Ready Timeout) และกระตุ้นให้เกิดเหตุการณ์ OnComm

#### 20. Handshaking

กำหนดคุณสมบัติและค่านิยามรูปแบบแฮนด์เช็กทางฮาร์ดแวร์  
รูปแบบการใช้งานคำสั่ง

object.Handshaking [ = value ]

ค่าตัวแปร Value ที่ใช้กำหนดค่ากำหนดได้ 4 รูปแบบด้วยกันคือ

1. comNone ค่าที่กำหนดคือ 0 เป็นการกำหนดให้ไม่มีการแฮนด์เช็ก (เป็นค่าเริ่มต้น)
2. comXOnOff ค่าที่กำหนดคือ 1 เป็นการกำหนดให้ใช้แฮนด์เช็กแบบ XON/XOFF
3. comRTS ค่าที่กำหนดคือ 2 เป็นการกำหนดให้ใช้ขา RTS/CTS (Request To Send/Clear To Send)
4. comRTSXOnOff ค่าที่กำหนดคือ 3 เป็นการกำหนดให้ใช้ทั้งแบบ Request To Send และ XON/XOFF

คุณสมบัติ Handshaking ใช้เพื่อกำหนดรูปแบบการสื่อสารภายใน ระหว่างที่ข้อมูลถูกส่งไปยังบัพเฟอร์ภาครับ เมื่อข้อมูลตัวอักษรถูกส่งมาถึงพอร์ตอนุกรม อุปกรณ์สื่อสารข้อมูลจะทำการย้ายข้อมูลไปยังบัพเฟอร์ภาครับ เพื่อที่จะให้โปรแกรมสามารถอ่านค่าไปใช้งานได้ ถ้าไม่มีบัพเฟอร์

ภาครับ โปรแกรมที่ใช้งานจะต้องทำการอ่านค่าข้อมูลโดยตรงจากฮาร์ดแวร์ของพอร์ตอนุกรม ซึ่งผู้ใช้งานจะเกิดปัญหาข้อมูลสูญหายได้ เนื่องจากว่าการเปลี่ยนแปลงของข้อมูลที่ส่งเข้ามามีการเปลี่ยนแปลงอย่างรวดเร็ว

คุณสมบัติ handshaking ช่วยให้ผู้ใช้งานแน่ใจได้ว่าข้อมูลที่ได้รับมานั้นไม่มีการสูญหายเมื่อบัฟเฟอร์ภาครับที่รับข้อมูลนั้นเกิดข้อมูลล้นหรือ โอเวอร์โฟลว (overflow) โดยใช้วิธีการตรวจสอบความพร้อมของบัฟเฟอร์ว่าพร้อมรับข้อมูลหรือไม่ก่อนที่จะส่งข้อมูลมาให้

## 21. Break

ใช้ในการเซตและเคลียร์ค่าสัญญาณ Break ชนิดของข้อมูลเป็นแบบ Boolean  
รูปแบบการใช้งาน

```
object.Break [ = value ]
```

โดย Value เป็นค่าบูลีน ถ้า Value = True หมายถึง การส่งสัญญาณ Break ออกไป (ขา TxD เป็นลอจิก “1”) ถ้า Value = False หมายถึงการเคลียร์สัญญาณ Break (ขา TxD เป็นลอจิก “0”)

เมื่อกำหนดให้สัญญาณ Break เป็น True จะเป็นการหยุดการส่งข้อมูลชั่วคราวจนกว่าจะมีการสั่งให้สัญญาณ Break เป็น False

ตัวอย่าง เป็นวิธีการส่งสัญญาณ Break ออกไปเป็นช่วงเวลาสั้นๆ ที่ 1/10 ของวินาที

```
MSComm1.Break = True
```

```
Duration! = Timer + .1
```

```
Do Until Timer > Duration!
```

```
Dummy =DoEvents()
```

```
Loop
```

```
MSComm1.Break = False
```

### 2.8.6.3 ค่าคงที่คุณสมบัติของคอนโทรล MSComm

ตารางที่ 2.7 แสดงค่าคงที่สำหรับคุณสมบัติ Handshake

ค่าคงที่	ค่า	รายละเอียด
comNone	0	ไม่ใช้การตรวจสอบแฮนด์เชก
comXonXoff	1	ไม่ใช้การตรวจสอบแฮนด์เชกแบบ Xon/Xof
comRTS	2	ใช้ในการตรวจสอบแฮนด์เชกผ่านทางขา RTS และ CTS
comRTSXonXoff	3	กำหนดการตรวจสอบแฮนด์เชกทั้ง แบบ RTS, CTS และ Xon/Xoff

**ตารางที่ 2.8** แสดงค่าคงที่สำหรับคุณสมบัติ OnComm

ค่าคงที่	ค่า	รายละเอียด
comEvSend	1	ส่งค่าเหตุการณ์ (send event)
comEvReceive	2	รับค่าเหตุการณ์ (receive event)
comEvCTS	3	มีการเปลี่ยนแปลงที่ขา CTS
comEvDSR	4	มีการเปลี่ยนแปลงที่ขา DSR
comEvCD	5	มีการเปลี่ยนแปลงที่ขา DCD
comEvRing	6	ตรวจจับสัญญาณกระดิ่งของ โทรศัพท์
comEvEOF	7	ตรวจพบตำแหน่งท้ายสุดของไฟล์ (End of file)

**ตารางที่ 2.9** แสดงค่าคงที่สำหรับคุณสมบัติ Error

ค่าคงที่	ค่า	รายละเอียด
comEventBreak	1001	ได้รับสัญญาณ Break
comEventCTSTO	1002	ขา CTS เกิด ไทม์เอาต์
comEventDSRTO	1003	ขา DSR เกิด ไทม์เอาต์
comEventFrame	1004	เกิดข้อผิดพลาดที่เฟรมข้อมูล (Framing error)
comEventOverrun	1006	พอร์ตอนุกรมเกิด โอเวอร์รัน (Port overrun)
comEventCDTO	1007	ขา DCD เกิด ไทม์เอาต์
comEventRxOver	1008	บัฟเฟอร์รับข้อมูลเกิด โอเวอร์โฟลว์
comEventRxParit	1009	เกิดข้อผิดพลาดที่พาริตี (Parity error)
comEventTxFull	1010	บัฟเฟอร์ส่งข้อมูลเต็ม

**ตารางที่ 2.10** แสดงค่าคงที่สำหรับคุณสมบัติ InputMode

ค่าคงที่	ค่า	รายละเอียด
comInputModeText	0	ข้อมูลที่รับมีคุณสมบัติเป็นข้อความ (ค่าปกติ)
comInputModeBinary	1	ข้อมูลที่รับเข้ามาเป็นข้อมูลไบนารี

## 2.8.7 ฟังก์ชัน ค่าคงที่ การส่งค่าพารามิเตอร์ และตัวอย่างโปรแกรมย่อยที่ใช้ในโปรแกรม วิชาล เบสิก

### 2.8.7.1 ฟังก์ชัน (Function)

#### 1. ฟังก์ชันเกี่ยวกับข้อความ (String)

Instr

ฟังก์ชันหาคำแหน่งของข้อความ

รูปแบบ Instr Z(ตำแหน่งเริ่มต้นในการนับ,ข้อความ,ข้อความที่ต้องการหาคำแหน่ง)

ตัวอย่าง Instr (“ABCDEF”,”C”) เป็นการค้นหาตัวอักษร “C” ซึ่งผลลัพธ์คือตำแหน่งที่ 3 หรือถ้าต้องการกำหนดตำแหน่งเริ่มต้นในการนับได้โดยเริ่มนับจากตัวที่ 4 ฟังก์ชันก็สามารถเขียนได้เป็น Instr (4,“ABCDEF”,”C”)

Mid

ฟังก์ชันตัดข้อความโดยกำหนดค่าภายในฟังก์ชัน

รูปแบบ Mid(ข้อความ,ตำแหน่งเริ่มต้น,จำนวนตัวนับจากตำแหน่งเริ่มต้น)

ตัวอย่าง Mid(“I love SoftPress”,3,5) เป็นการตัดข้อความ โดยเริ่มต้นจากตัวที่ 3 และนับถัดไปอีก 5 ผลที่ได้คือ love

Space

ฟังก์ชันเพิ่มช่องว่างตามจำนวนที่กำหนดไว้ในฟังก์ชัน

รูปแบบ Space(จำนวนช่องว่าง)

ตัวอย่าง Space(100) ให้แสดงช่องว่าง 100 ช่อง

#### 2. ฟังก์ชันเกี่ยวกับตัวเลข

Val

แปลงค่าตัวเลข

รูปแบบ Val(ตัวเลขที่เป็นข้อความ)

ฟังก์ชันนี้มักใช้กับการแปลงค่าจากการรับข้อมูลผ่านคอนโทรลที่กดปุ่มอักษร ถึงแม้จะพิมพ์ข้อมูลเป็นตัวเลขแต่ข้อมูลดังกล่าวจะกลายเป็นข้อมูลประเภทข้อความโดยอัตโนมัติ ซึ่งถ้าจะนำข้อมูลดังกล่าวมาคำนวณจะต้องใช้ฟังก์ชัน Val

Round

ฟังก์ชันตัวเลข

รูปแบบ Round (ตัวเลข,จำนวนหลักทศนิยม)

ตัวอย่าง Round(1234.1234,3) ผลที่ได้คือ 1234.123

### 2.8.7.2 ค่าคงที่ (Constant)

#### ชนิดของข้อมูล (Data Type)

ชนิดของข้อมูลที่เรากำหนดให้กับตัวแปรนั้นมีจุดประสงค์เพื่อให้การเก็บข้อมูลของเรามีประสิทธิภาพ เพราะใช้พื้นที่ขนาดเหมาะสมในการเก็บข้อมูลโดยในวิซวล เบสิก 6 จะมีชนิดข้อมูลต่างๆดังตารางต่อไปนี้

ตารางที่ 2.11 ชนิดข้อมูลต่างๆที่ใช้ใน โปรแกรมวิซวล เบสิก

ชนิดของข้อมูล	ขนาดไบต์	คำอธิบาย
Byte	1	มีค่าตั้งแต่ 0-255
Integer	2	มีค่าตั้งแต่ -32,767 ถึง 32,768
Long	4	มีค่าตั้งแต่ -2,147,483,648 ถึง 2,147,483,647
Single	4	จะแสดงรายละเอียดต่อไป
Double	8	จะแสดงรายละเอียดต่อไป
Currency	8	มีค่าตั้งแต่ -922,337,203,685,477.5808 ถึง -922,337,203,685,477.5807
Dec		จะแสดงรายละเอียดต่อไป
Boolean	1	มีค่า True (แทนค่าด้วยค่า -1) กับ False (แทนค่าด้วยค่า 0)
String	?	ใช้ในการเก็บข้อความที่เป็นชุดของตัวอักษร
Date		เป็นข้อมูลชนิดวันที่และเวลา
Variant		เป็นข้อมูลที่สามารถใช้แทนชนิดของข้อมูลอื่นได้
Object		เป็นข้อมูลชนิดคอนโทรล

## หมายเหตุ

### Single

เป็นชนิดข้อมูลประเภททศนิยมยกกำลังที่มีความละเอียดแบบ double precision (ความละเอียดสูง) จะใช้เนื้อที่ 8 ไบต์ในการเก็บข้อมูลชนิดนี้ มีค่าดังต่อไปนี้

- 4.94065645841247E-324 ถึง 1.79769313486232E308 สำหรับเลขบวก
- -1.79769313486232E308 ถึง -4.94065645841247E-324
- 0

### Double

เป็นชนิดข้อมูลประเภททศนิยมยกกำลังที่มีความละเอียดแบบ single precision (ความละเอียดต่ำ) จะใช้เนื้อที่ 4 ไบต์ในการเก็บข้อมูลชนิดนี้ มีค่าดังต่อไปนี้

- 1.40129E-45 ถึง 3.402823E38 สำหรับเลขบวก
- -3.402823E38 ถึง -1.40129E-45 สำหรับเลขลบ
- 0

### Dec

เป็นชนิดข้อมูลประเภทตัวเลขที่สามารถเก็บค่าตัวเลขได้ใหญ่มาก ซึ่งจะเก็บตัวเลขจำนวนเต็มที่มีค่าอยู่ระหว่าง -79,228,162,514,264,337,593,543,950,335 ถึง +79,228,162,514,264,337,593,543,950,335 แต่ถ้าเราจะเก็บข้อมูลเป็นเลขทศนิยมจะมีค่าอยู่ระหว่าง -7.9228162514264337593543950335 ถึง +7.9228162514264337593543950335 ใช้เนื้อที่หน่วยความจำ 12 ไบต์ในการเก็บข้อมูลชนิดนี้

### 2.8.7.3 โปรแกรมย่อย ฟังก์ชัน และการส่งพารามิเตอร์

วิธีการเขียนโปรแกรมที่ดีคือการแยกโปรแกรมออกเป็นส่วนย่อยๆ ที่ทำงานอย่างหนึ่งจนเสร็จ ที่เราเรียกว่าโปรแกรมย่อย (Procedure) ข้อดีของการใช้โปรแกรมย่อยคือ

- ช่วยให้โปรแกรมของเราทำความเข้าใจได้ง่ายขึ้นเพราะมีการแบ่งส่วนย่อยๆ
- ช่วยให้เรานำโปรแกรมย่อยไปใช้กับโปรแกรมอื่นได้ ถ้าโปรแกรมนั้นต้องการฟังก์ชันที่เหมือนหรือใกล้เคียงกัน
- ลดความซ้ำซ้อนในการเขียนโปรแกรมโดยนำส่วนที่ต้องใช้ซ้ำๆ มาเขียนเป็นโปรแกรมย่อย ในวิชวล เบสิก 6.0 จะมีโปรแกรมย่อยที่ใช้งานทั่วไปอยู่ 2 ประเภทคือ
- ในโปรแกรมย่อย Sub เป็นโปรแกรมย่อยที่จะทำงานอย่างหนึ่ง
- โปรแกรมย่อย Function เป็นโปรแกรมย่อยที่จะคือผลลัพธ์ที่ได้จากการทำงานกลับมาด้วย

## 1. โปรแกรมย่อย Sub

เป็นโปรแกรมที่ไม่มีการส่งค่ากลับมา ที่มีรูปแบบดังนี้

[Private| Public] [Static] Sub <ชื่อของโปรแกรมย่อย> (พารามิเตอร์ที่ส่งมา)

‘ ชุดคำสั่ง

[Exit Sub]

End Sub

- คำว่า Private หรือ Static จะเป็นคำประกาศขอบเขตที่ต้องการให้โปรแกรมย่อยนี้มีขอบเขตอยู่ในเฉพาะใน โมดูลนี้ หรือทุก โมดูลใน โปรเจคที่สร้างขึ้น
- คำว่า Static เป็นการบอกว่า ให้ตัวแปรที่อยู่ในโปรแกรมย่อยนี้ เป็นแบบ Static ทั้งหมด
- คำสั่ง Exit Sub จะทำให้ออกจากโปรแกรมย่อยทันที เมื่อพบกับคำสั่งนี้
- คำว่า End Sub เป็นคำสั่งที่บอกว่าเป็นการจบการทำงานของโปรแกรมย่อยนี้

## 2. โปรแกรมย่อย Function

เป็นโปรแกรมย่อยที่มีการส่งค่ากลับมาด้วย มีรูปแบบดังนี้

[Private| Public] [Static] Function <ชื่อของโปรแกรมย่อย> (พารามิเตอร์ที่ส่งมา) As Type

‘ ชุดคำสั่ง

[Exit Function]

End Function

- คำว่า Private หรือ Static จะเป็นคำประกาศขอบเขตที่ต้องการให้โปรแกรมย่อยนี้มีขอบเขตอยู่ในเฉพาะใน โมดูลนี้ หรือทุก โมดูลใน โปรเจคของเรา
- คำว่า Static เป็นการบอกว่า ให้ตัวแปรที่อยู่ในโปรแกรมย่อยนี้ เป็นแบบ Static ทั้งหมด
- คำสั่ง Exit Function จะทำให้ออกจากโปรแกรมย่อยทันที เมื่อพบกับคำสั่งนี้
- คำว่า End Function เป็นคำสั่งที่บอกว่าเป็นการจบการทำงานของโปรแกรมย่อยนี้
- คำว่า As Type หลัง Function ใช้กำหนดชนิดข้อมูลที่ฟังก์ชันนี้ส่งกลับมาให้โปรแกรมหลัก

## 3. การส่งค่าพารามิเตอร์ให้กับโปรแกรมย่อย

ในการเรียกใช้โปรแกรมย่อยบางครั้ง เราต้องส่งข้อมูลที่โปรแกรมย่อยต้องใช้ในการทำงาน จากโปรแกรมหลัก ซึ่งเราเรียกว่าการส่งพารามิเตอร์ การส่งพารามิเตอร์แบ่งเป็น 2 ประเภทคือ

### 3.1 การส่งค่าแบบส่งค่าไปทางเดียว (Pass By Value)

เป็นการส่งค่าของตัวแปร ไปอย่างเดียว เมื่อมีการแก้ไขค่านั้นๆ ในโปรแกรมย่อยที่รับค่าเข้ามา จะไม่กระทบต่อค่าของตัวแปรเดียวกันในโปรแกรมหลัก เราจะใช้คำว่า ByVal อยู่หน้าพารามิเตอร์ที่ส่งเพื่อบอกว่าเป็นพารามิเตอร์แบบส่งค่าไปอย่างเดียวนั่นเอง

```
Sub MySub (ByVal Y As Integer)
```

```
    Y=Y*3
```

```
End Sub
```

ถ้ามีการเรียกโปรแกรมย่อย MySub ดังนี้

```
Dim A As Integer
```

```
A = 3
```

```
MySub A
```

จากคำสั่งข้างต้น หลังจากการเรียกใช้โปรแกรมย่อย MySub แล้วค่าของ A ในโปรแกรมหลักจะไม่มีเปลี่ยนแปลงไป คือ จะมีค่า 3 เหมือนเดิม ไม่ใช่ 9

### 3.2 การส่งค่าแบบส่งค่าแบบอ้างอิง (Pass By Reference)

การส่งค่าพารามิเตอร์แบบนี้ จะทำให้โปรแกรมย่อยที่ถูกเรียกใช้สามารถแก้ไขค่าของตัวแปรที่ส่งเป็นพารามิเตอร์ให้แก่โปรแกรมย่อยนั้นได้ ซึ่งการส่งพารามิเตอร์แบบนี้จะเป็นการส่งพารามิเตอร์ที่ วิซวล เบสิก 6.0 ใช้อยู่โดยทั่วไปเช่น

```
Sub MySub (ByRef Y As Integer)
```

```
    Y=Y*3
```

```
End Sub
```

ถ้ามีการเรียก โปรแกรมย่อย MySub ดังนี้

Dim A As Integer

A = 3

MySub A

จากคำสั่งข้างต้น หลังจากการเรียกใช้โปรแกรมย่อย MySub แล้วค่าของ A ในโปรแกรมหลักจะมีการเปลี่ยนแปลงเป็น 9 ไม่ใช่ 3 สำหรับคีย์เวิร์ด ByRef นั้นไม่จำเป็นต้องใส่ก็ได้ เพราะเนื่องจากเป็นค่าเบื้องต้นอยู่แล้ว

#### 4. DoEvents

ในระบบปฏิบัติการวินโดวส์ เราสามารถทำงานได้หลายอย่างพร้อมๆกัน ด้วยคุณลักษณะเฉพาะของวินโดวส์ที่เราเรียกว่า Preemptive Multitasking ด้วยคุณสมบัตินี้ทำให้โปรแกรมสามารถทำงานเบื้องหน้า (ติดต่อกับผู้ใช้ในขณะนั้น) และทำงานเบื้องหลัง (ไม่ได้ติดต่อกับผู้ใช้ แต่จะทำงานตอนที่โปรแกรมเบื้องหน้าหยุดพัก หรือรอการตัดสินใจของผู้ใช้)

สำหรับการทำงานแบบเบื้องหลังนั้น (Background Processing) โปรแกรมจะไม่ตอบสนองต่ออีเวนต์ได้เพราะระบบวินโดวส์จะส่งอีเวนต์จากผู้ใช้ไปให้โปรแกรมเบื้องหน้าเท่านั้นซึ่งปัญหานี้เราสามารถแก้ไขได้

ในวิชาลเบสิก 6.0 มีคำสั่ง DoEvents ที่ช่วยให้โปรแกรมที่ทำงานเบื้องหลังสามารถตอบสนองกับอีเวนต์อื่นได้ในระหว่างการทำงานอันยาวนาน เช่น ให้ปุ่ม Cancel ได้ยกเลิกการทำงานได้ในระหว่างการทำสำเนาหลายๆไฟล์ ที่ต้องใช้เวลาในการทำงานซึ่งคำสั่งนี้มีรูปแบบต่อไปนี้

ตัวอย่างการใช้งานคำสั่ง DoEvents

ในโปรแกรมการนับเลข 0 - 10000

No DoEvents คือ จะไม่มีการแสดงตัวเลขในเท็กซ์บ็อกซ์จนกว่าการนับเลขจะเสร็จสิ้น

DoEvents คือ เมื่อมีการนับเลขเพิ่มขึ้นหนึ่งค่าจะแสดงค่าที่นับได้ในเท็กซ์บ็อกซ์โดยมีการเปลี่ยนแปลงตลอดการนับ

## 2.9 การเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอก

การเคลื่อนย้ายข้อมูลจากคอมพิวเตอร์ไปยังอุปกรณ์ต่อพ่วงภายนอกหรือคอมพิวเตอร์ด้วยกัน มีด้วยกัน 2 รูปแบบคือ รับส่งข้อมูลแบบขนานและรับส่งข้อมูลแบบอนุกรม การรับส่งข้อมูลแบบขนานเป็นการรับหรือส่งข้อมูลคราวละ 4 หรือ 8 บิตในเวลาเดียวกัน ทำให้การรับและส่งข้อมูลมีความเร็วสูง ทว่าจำนวนของสายที่ใช้ในการถ่ายทอข้อมูลต้องมีมากเท่ากับจำนวนบิตของข้อมูลที่ทำการถ่ายทอด้วย นอกจากนี้ยังมีสายที่ใช้สำหรับควบคุมและตรวจสอบการรับส่งข้อมูลด้วย ซึ่งอาจต้องใช้สายมากเป็น 2 เท่าของจำนวนบิตข้อมูลก็ได้ ส่งผลให้ราคาของสายที่ใช้ในการเชื่อมต่อแบบขนานมักจะมีราคาแพง อีกข้อจำกัดหนึ่งของการถ่ายทอข้อมูลแบบขนานคือ ระยะทางในการถ่ายทอข้อมูล โดยปกติจะอยู่ที่ประมาณ 10-15 ฟุต ในขณะที่การรับส่งข้อมูลแบบอนุกรมจะเป็นการรับส่งข้อมูลครั้งละ 1 บิต โดยมีรูปแบบการรับส่งที่เป็นมาตรฐาน ต้องมีการตรวจสอบความพร้อมในการรับและส่งข้อมูลของตัวส่งและตัวรับ การรับส่งข้อมูลแบบอนุกรมมีข้อดีในเรื่องของจำนวนสายสัญญาณที่น้อยมากและไม่แปรผันตามจำนวนบิตของข้อมูล ระยะทางในการรับส่งข้อมูลสูงกว่าแบบขนานมาก โดยปกติถ้าเป็นพอร์ตอนุกรม อาร์เอส-232ซี (RS-232C) จะสามารถต่อสายได้ยาว 50 ฟุตโดยประมาณ

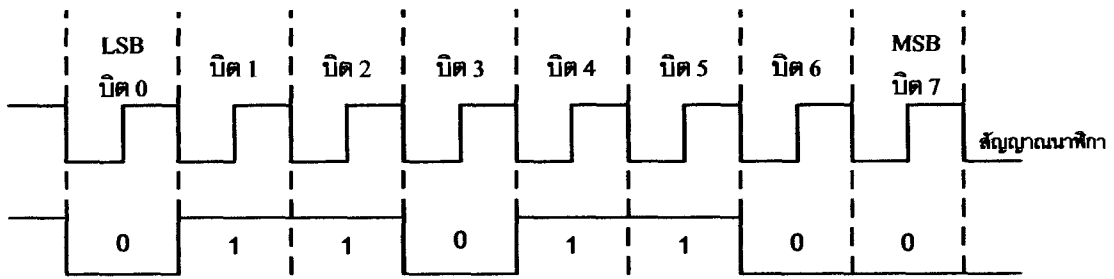
### 2.9.1 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมแบ่งได้เป็น 2 แบบคือ

1. การสื่อสารอนุกรมแบบซิงโครนัส
2. การสื่อสารอนุกรมแบบอะซิงโครนัส

#### 2.9.1.1 การสื่อสารข้อมูลแบบซิงโครนัส

1. การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิกา ร่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือคีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้นคือ สัญญาณนาฬิกา, ข้อมูลและกราวด์ รูปที่ 1 แสดงให้เห็นถึงไคอะแกรมเวลาของการสื่อสารข้อมูลแบบซิงโครนัส



รูปที่ 2.52 การสื่อสารแบบซิงโครนัส

### 2.9.1.2 การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับและส่งข้อมูลโดยไม่จำเป็นต้องมีสัญญาณนาฬิกาพร้อมด้วย แต่จะใช้การกำหนดค่าอัตราเร็วในการรับและส่งข้อมูลให้มีค่าเท่ากัน ซึ่งเรียกอัตราเร็วนี้ว่า อัตราบอด หรือบอดเรต (baud rate) มีหน่วยเป็น บิตต่อวินาที (bit per second : bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกันคือ

1. บิตเริ่มต้น (start bit) มีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรม มีขนาด 5,6,7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี (parity bit) มีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้ายหรือบิตหยุด (stop bit) มีขนาด 1, 1.5 หรือ 2 บิต



รูปที่ 2.53 การสื่อสารข้อมูลแบบอะซิงโครนัส

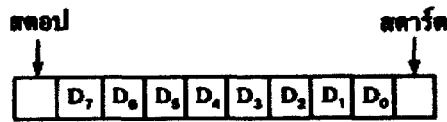
แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส เมื่อไม่มีการส่งข้อมูล ขา DATA จะมีสถานะลอจิก "1" เรียกสถานะนี้ว่า สถานะหยุดรอ (waiting stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขา DATA มีลอจิก "0" ด้วยช่วงระยะเวลา 1 บิต เรียกบิตนี้ว่า บิตเริ่มต้นจากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุดหรือบิต LSB ก่อน ซึ่งข้อมูลที่ต้องการส่งอาจมีจำนวน 5,6,7 หรือ 8 บิตก็ได้ จากนั้นตามด้วยบิตพาริตีซึ่งใช้ในการตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่จะส่งคือ บิตปิดท้ายหรือบิตหยุดโดยจะเป็นการทำให้ขา DATA มีสถานะลอจิก "1" อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต, 1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัสหรืออัตราบอดที่ใช้สำหรับพอร์ตอนุกรม อาร์เอส-232 มีด้วยกันหลายค่า ได้แก่ 110, 150, 300, 600, 1,200, 2,400, 4,800, 9,600 และ 19,200 บิตต่อวินาที โดยมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ เนื่องจากอัตราบอดคือค่าของจำนวนบิตที่สามารถส่งได้ใน 1 วินาที สมมติว่า ข้อมูลอนุกรมมีขนาด 8 บิต ไม่มีการตรวจสอบพาริตี มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูล 1 ไบต์ จะมีความยาวเท่ากับ 10 บิต ถ้าใช้อัตราบอดในการส่งข้อมูลเท่ากับ 9,600 บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที

ตารางที่ 2.12 ค่าอัตราบอดและอัตราการเร็วในการส่งข้อมูล

Baud Rate	Bytes/Second
110	10
150	15
300	30
600	60
1200	120
2400	240
4800	480
9600	960
19200	1920
38400	3840

จากตารางพบว่าในการส่งข้อมูลด้วยอัตราบอด 110 จะมีรูปแบบต่างจากอัตราอื่น ๆ คือ จะใช้บิตหยุด 2 บิต ดังนั้นจึงต้องส่งข้อมูลที่มีขนาด 11 บิต



รูปที่ 2.54 การส่งข้อมูลขนาด 8 บิตที่ใช้ในการโอนย้ายข้อมูลแบบอนุกรม

การตรวจสอบพาริตีสามารถกำหนดให้เป็นแบบคี่ (odd), แบบคู่ (even) หรือไม่มีการตรวจสอบพาริตีก็ได้ พาริตีคี่หรือพาริตีคู่แสดงถึงจำนวนลอจิก "1" ทั้งหมดภายในข้อมูลที่ส่งไป 1 ไบต์รวมพาริตีว่ามีจำนวนเป็นเลขคู่หรือเลขคี่ ยกตัวอย่าง ข้อมูลที่จะทำการส่งมีขนาด 8 บิต มีค่าเท่ากับ 99H หรือ 10011001B จะเห็นว่าข้อมูลในไบต์นี้มีจำนวนลอจิก "1" จำนวน 4 ตัวซึ่งเป็นเลขคู่ ดังนั้นถ้ากำหนดค่าพาริตีเป็นคู่ ค่าของบิตพาริตีจะต้องมีลอจิกเป็น "0" แต่ถ้ากำหนดพาริตีเป็นคี่ ค่าของบิตพาริตีจะต้องเป็น "1" เพื่อให้ข้อมูล 1 ไบต์รวมทั้งบิตพาริตีเป็นคี่ ถ้าข้อมูลมีค่าพาริตี ไม่ตรงตามที่ กำหนดบิตพาริตีในรีจิสเตอร์สถานะของ UART (Universal Asynchronous Receiver Transmitter) ก็จะถูกเซตเพื่อแสดงว่าข้อมูลที่ได้รับมาผิดพลาดและ โปรแกรมที่ทำการรับข้อมูลนั้นก็ จะขอให้มีการส่งข้อมูลมาใหม่



แสดงการใช้บิตพาริตีเพื่อตรวจสอบความผิดพลาดในการโอนย้ายข้อมูลแบบอนุกรม

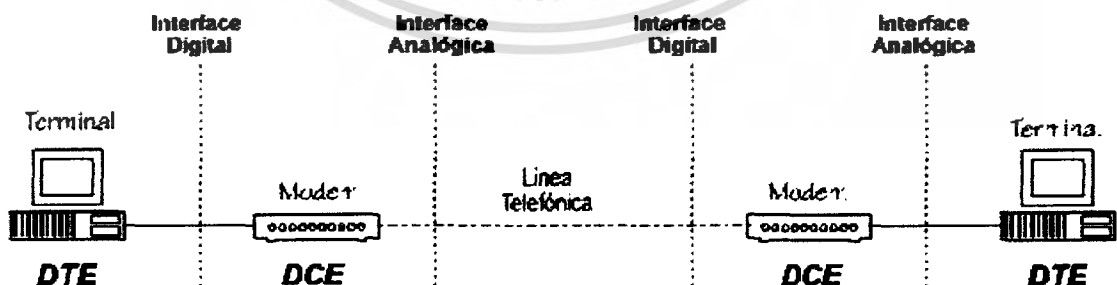
รูปที่ 2.55 การใช้บิตพาริตีเพื่อตรวจสอบความผิดพลาดในการส่งข้อมูลแบบอนุกรม

การใช้พาริตีนี้จะสามารถตรวจสอบความผิดพลาด ได้เพียง 50 เปอร์เซ็นต์ของความผิดพลาดทั้งหมดทั้งนี้เนื่องจากมันจะจับความผิดพลาด ได้เฉพาะกรณีที่ข้อมูลนั้นเกิดข้อผิดพลาด เป็นจำนวนคี่บิต ถ้าข้อมูลนั้นเกิดความผิดพลาดเป็นจำนวนคู่บิตก็จะไม่ทำให้ค่าพาริตีเปลี่ยนแปลง ความผิดพลาดก็จะไม่ถูกตรวจพบ UART ยังสามารถตรวจสอบ framing error ได้ framing error จะเกิดขึ้นเมื่อ UART ได้รับข้อมูลที่มีบิตเริ่มต้นและบิตหยุดในตำแหน่งที่ไม่ถูกต้องซึ่ง บิตพาริตีถูก

สร้างขึ้นจากภาคส่งข้อมูลของ UART เป็นอุปกรณ์ที่ใช้ในการรับและส่งข้อมูลอนุกรม ซึ่งจะกล่าวถึงในรายละเอียดภายหลัง) ซึ่งทางภาครับจะต้องกำหนดคุณสมบัติการตรวจสอบพาริตีที่ตรงกันเอาไว้ว่าจะตรวจสอบพาริตีที่หรือพาริตีคู่ จากนั้นภาครับของ UART จะทำการตรวจสอบค่าพาริตีที่เกิดขึ้นว่าเป็นคู่หรือเป็นคี่ โดยการนับจำนวนลอจิก 1 ทั้งหมดรวมทั้งบิตพาริตีด้วย ถ้ากำหนดพาริตีไว้เป็นคู่แต่อ่านค่าตัวเลขในการนับออกมาได้ตัวเลขเป็นคี่ ทางภาครับจะแสดงข้อผิดพลาดออกมาให้ผู้ใช้งานทราบ กระบวนการดังกล่าวเป็นวิธีการตรวจสอบความผิดพลาดที่เกิดขึ้นในการรับส่งข้อมูลที่ง่ายที่สุด แต่มันสามารถตรวจสอบได้เมื่อมีบิตข้อมูลที่ทำการรับส่งผิดพลาดเพียงบิตเดียวเท่านั้น

### 2.9.2 มาตรฐานพอร์ตอนุกรมแบบอาร์เอส-232 (RS-232)

มาตรฐานการเชื่อมต่อแบบอนุกรมอาร์เอส-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน อาร์เอส-232 ในอดีตนั้นถูกออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนี้ส่งผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดซึ่งอยู่ห่างไกลกัน โดยสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association : EIA ) ได้วางมาตรฐานที่มีชื่อเรียกกันว่า อีไอเอ อาร์เอส-232 (EIA RS-232 ) มาตรฐานนี้ในช่วงแรกจะใช้คอนเน็คเตอร์เป็นแบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ -3 โวลต์ จนถึง -12 โวลต์ แสดงว่ามีข้อมูล (mark) และ +3 โวลต์ ถึง +12 โวลต์ แสดงว่าเป็น ช่องว่าง (space) มาตรฐาน อาร์เอส-232 ถูกใช้ในการกำหนดรูปแบบการสื่อสารข้อมูลกันระหว่างอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment: DTE) ซึ่งเป็นอุปกรณ์สำหรับส่งข้อมูล (เอาต์พุต) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating: DCE) ซึ่งเป็นอุปกรณ์สำหรับรับข้อมูล (อินพุต)



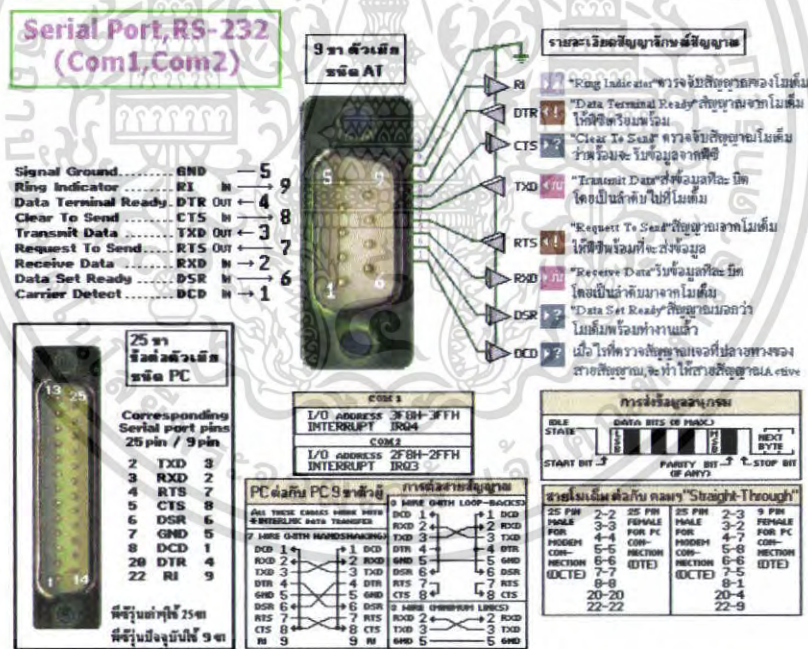
รูปที่ 2.56 DTE และ DCE

อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัวเช่น ไมโครคอนโทรลเลอร์หรือ ไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE ทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น

ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่เราเห็นได้ชัดคือ คอนเน็กเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็กเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็กเตอร์ที่อยู่ทีโมเด็มจะเป็นแบบ DCE สำหรับการใช้งานในคอมพิวเตอร์ พอร์ตอนุกรม อาร์เอส-232 ถูกใช้เพื่อเชื่อมต่อกับ โมเด็ม, เม้าส์ และเครื่องพิมพ์ที่มีสามารถติดต่อทางพอร์ตอนุกรมได้

### 2.9.3 คอนเน็กเตอร์สำหรับพอร์ตอาร์เอส-232 และการเชื่อมต่อ

มาตรฐานการเชื่อมต่อแบบอาร์เอส-232 จะใช้คอนเน็กเตอร์แบบ DB-25 ตัวผู้ หรือ DB-9 ตัวผู้ ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อใช้งาน เพียง 9 เส้นเช่นเดียวกับคอนเน็กเตอร์แบบ DB-9 เนื่องจากขาอื่นๆ ที่เคยมีการใช้งานมาในอดีตไม่ค่อยมีสำคัญมากนักจึงถูกยกเลิกไป



- 2.ขา Receive Data : RD หรือ RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยจะนำข้อมูลที่อ่านได้ไปเก็บไว้ในรีจิสเตอร์บัฟเฟอร์
- 3.ขา Transmitted Data : TD หรือ TxD ขานี้ใช้เพื่อส่งข้อมูลอนุกรมออกจากคอมพิวเตอร์ โดยการนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป
- 4.ขา Data Terminal Ready : DTR เป็นขาเอาต์พุตที่ใช้สำหรับส่งสัญญาณออกจากคอมพิวเตอร์ เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อกับอุปกรณ์ปลายทาง โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ และถ้าใช้การเชื่อมต่อแบบใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องเชื่อมต่อขา DTR และ DSR ของพอร์ตอนุกรมเข้าด้วยกัน และจะต้องต่อเชื่อมเข้ากับขา DCD ด้วยในกรณีที่ใช้โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห์
- 5.ขา Signal Ground : GND เป็นขาราวด์ของสัญญาณ
- 6.ขา Data Set Ready : DSR ขานี้จะใช้ควบคู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอก
- 7.ขา Request To Send : RTS เป็นขาเอาต์พุตสำหรับส่งสัญญาณร้องขอให้อุปกรณ์ปลายทางส่งข้อมูลมาให้คอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ซึ่งในกรณีที่มีการเชื่อมต่อแบบ 3 สาย จะต้องเชื่อมต่อกับขา RTS และ CTS เข้าด้วยกัน เพื่อให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา
- 8.ขา Clear To Send : CTS เป็นขาอินพุตทำหน้าที่รอรับสัญญาณที่ส่งเข้ามา เมื่อมีการส่งสัญญาณเข้ามาที่ขานี้ ข้อมูลที่ขา TxD จะถูกส่งออกไป ขานี้จะใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลแล้วหรือยัง
- 9.ขา Ring Indicator : RI ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มแล้วยังมีความต้องการตรวจสอบสัญญาณเรียกจากสายโทรศัพท์

ตารางที่ 2.13 หน้าที่ของขาต่างๆในคอนเน็คเตอร์แบบ DB-25 ตัวผู้ และ DB-9 ตัวผู้

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

### ตารางที่ 2.13 (ต่อ) หน้าทีของขาต่างๆในคอนเน็กเตอร์แบบ DB-25 ตัวผู้ และ DB-9 ตัวผู้

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

#### 2.9.4 UART

UART มาจากคำว่า Universal Asynchronous Receiver Transmitter ซึ่งหมายถึงอุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัสนั่นเอง สำหรับการสื่อสารอนุกรมบนคอมพิวเตอร์แล้ว UART ถือว่าเป็นหัวใจสำคัญของการสื่อสารอนุกรม

หน้าที่หลักของ UART คือแปลงข้อมูลที่อยู่ในรูปแบบขานานจากซีพียูให้อยู่ในรูปแบบอนุกรมแบบอะซิงโครนัส แล้วส่งทำการส่งออก ไป และแปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขานานก่อนที่จะส่งเข้าสู่ซีพียู ซึ่งนอกจาก UART จะส่งข้อมูลไปยังซีพียูแล้ว ยังแจ้งรายละเอียดอื่นๆ ของข้อมูล ให้คอมพิวเตอร์รับทราบด้วย อาทิ อัตราเร็วในการรับส่งข้อมูลหรืออัตราบอด, รูปแบบการส่งข้อมูล, ความผิดพลาดที่เกิดขึ้นระหว่างการส่งข้อมูล เช่น ผิดพลาดจากพาริตี, เฟรมข้อมูล, โอเวอร์รัน เป็นต้น

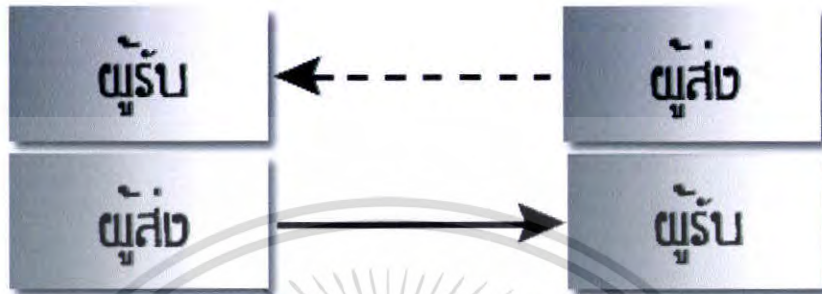
ภายใน UART จะมีวงจรสร้างอัตราบอดโปรแกรมได้ (programmable baudrate generator) โดยการกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของ UART โดยตัวหารนี้จะมีขนาด 16 บิต ดังนั้นจะสามารถกำหนดตัวหารอยู่ในช่วง 1 - 65,535 ซึ่งมี รูปแบบในการส่งสัญญาณสำหรับการสื่อสารข้อมูลระหว่างอุปกรณ์กับเครื่อง คอมพิวเตอร์ในเครือข่ายมีลักษณะของรูปแบบการส่งสัญญาณเป็น 3 รูปแบบดังนี้คือ

1. ซิมเพล็กซ์ (Simplex) การส่งข้อมูลแบบซิมเพล็กซ์มีรูปแบบคือสัญญาณข้อมูลถูกส่งในทิศทางเดียว โดยฝ่ายส่งทำหน้าที่ส่งและฝ่ายรับทำหน้าที่รับ ซึ่งทั้งสองฝ่ายจะทำหน้าที่เพียงส่ง และรับเพียงอย่างเดียวเท่านั้น ตัวอย่างของการส่งข้อมูลนี้คือการดูโทรทัศน์



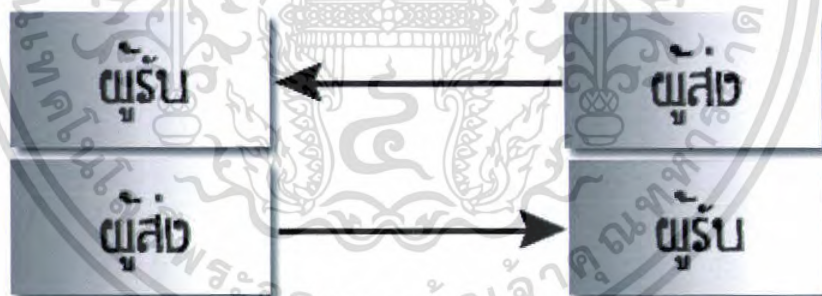
รูปที่ 2.58 แสดงการส่งข้อมูลแบบซิมเพล็กซ์

2. ฮาล์ฟดูเพล็กซ์ (Half Duplex) การส่งข้อมูลแบบฮาล์ฟดูเพล็กซ์ ซึ่งมีรูปแบบการส่งสัญญาณข้อมูลที่สวนทางกันได้โดยการสลับกัน จะส่งหรือรับสัญญาณในเวลาเดียวกันไม่ได้ เช่น การสื่อสารของระบบวิทยุมือถือ



รูปที่ 2.59 แสดงการส่งข้อมูลแบบฮาล์ฟดูเพล็กซ์

3. ฟูลดูเพล็กซ์ (Full Duplex) การส่งข้อมูลแบบฟูลดูเพล็กซ์ ซึ่งมีรูปแบบการส่งข้อมูลสองทิศทางได้ในเวลาเดียวกันหรือส่งข้อมูลได้พร้อมกันสองทาง เช่น การสื่อสารโดยใช้โทรศัพท์



รูปที่ 2.60 แสดงการส่งข้อมูลแบบฟูลดูเพล็กซ์

### 2.9.5 มาตรฐานการสื่อสารข้อมูลอนุกรมแบบอาร์เอส 485 (RS -485)

มาตรฐานอาร์เอส 485 กำหนดโดยสมาคมผู้ประกอบการอุตสาหกรรมอิเล็กทรอนิกส์หรือ EIA เป็นมาตรฐานการเชื่อมต่อสัญญาณแบบอนุกรม (Serial Communication) มีลักษณะการเชื่อมต่อเป็นแบบหลายจุด (Multi-point หรือ Multi-drop) สายสัญญาณที่ใช้มีทั้งแบบที่เป็น 2 สายและแบบที่เป็น 4 สาย การต่อแบบหลายจุดนี้ทำให้สามารถมองสายสัญญาณเป็นบัสนำสัญญาณได้ (Signal Bus) จำนวนคอมพิวเตอร์หรืออุปกรณ์ที่สามารถอยู่บนอาร์เอส-485 บัสหนึ่งถูกกำหนดไว้ที่

32 ตัว ในกรณีที่ต้องการเพิ่มจะต้องมีตัวทวนสัญญาณ (Signal Repeater) หรือใช้ตัวส่ง-รับสัญญาณที่มีอิมพีแดนซ์ (ความต้านทานเสมือน) สูงขึ้น ซึ่งเราอาจเพิ่มจำนวนจุดเชื่อมต่อขึ้นได้ถึง 128 จุด ความยาวของสายสัญญาณตามมาตรฐานอาร์เอส-485 นี้สามารถยาวได้ถึง 1.2 กิโลเมตร เช่นเดียวกับมาตรฐานอาร์เอส-422 แต่การสื่อสารจะเป็นแบบสองทางไม่พร้อมกัน (แบบฮาล์ฟดูเพล็กซ์) มีเพียงคอมพิวเตอร์หรืออุปกรณ์ตัวเดียวเท่านั้นที่สามารถส่งสัญญาณออกได้ ณ เวลาหนึ่งๆ ส่วนที่เหลือจะเป็นผู้รับสัญญาณ หรือผู้ฟัง

#### ระบบเครือข่ายอาร์เอส 485

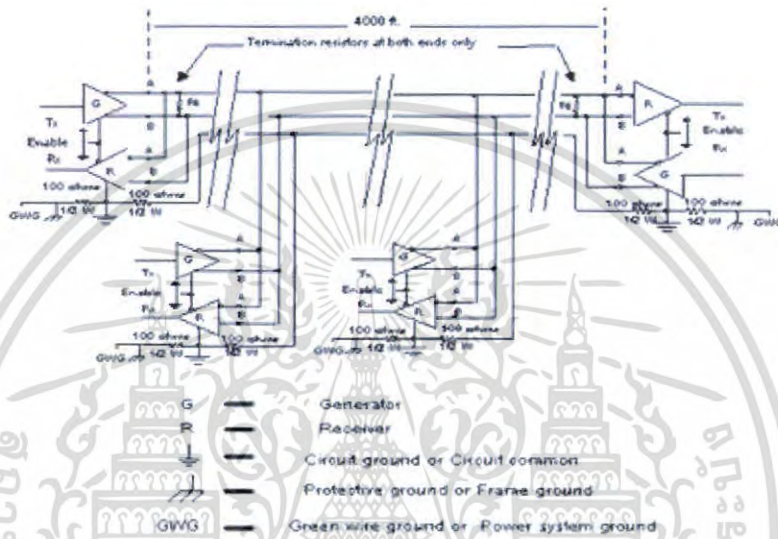
- สามารถต่ออุปกรณ์ในระบบได้ไม่เกิน 32 ตัว
- ติดต่อกันแบบฮาล์ฟดูเพล็กซ์
- ระบบสมดุล (Balanced system) ใช้สายสัญญาณ 1 คู่ และ สายกราวด์ 1 เส้น
- ติดต่อกันระยะทางไม่เกิน 1,200 เมตร หรือ 4,000 ฟุต
- ความยาวของระบบและจำนวนโหนดสามารถขยายได้โดยตัวทวนสัญญาณ (Repeater)

#### ฮาร์ดแวร์ของอาร์เอส 485

ระบบเครือข่ายอาร์เอส-485 สามารถติดต่อกันผ่านสาย 2 เส้น ( โดยทั่วไปใช้สายคู่บิดเกลียว หรือ twisted pair ) สัญญาณข้อมูลที่ส่งจะถูกวัดความแตกต่างของสัญญาณที่ตัวรับ ซึ่งมีข้อดีคือสามารถส่งข้อมูลในระยะไกลและกำจัดปัญหาสัญญาณรบกวนได้ดี ลักษณะการเชื่อมต่อของอาร์เอส-485 สามารถทำได้ใน 2 ลักษณะคือ 2 สาย (two-wire) และ 4 สาย (four-wire) ในระบบ 2 สาย ตัวส่งและตัวรับของแต่ละอุปกรณ์ต่อเข้ากับสายคู่บิดเกลียว 1 เส้น ระบบ 4 สาย ตัวส่งของพอร์ตแม่ (master port) ต่อเข้ากับตัวรับของตัวลูก (slave port) ทุกตัวในระบบผ่านสายคู่บิดเกลียวเส้นหนึ่ง และตัวส่งของตัวลูกทุกตัวจะต่อเข้ากับตัวรับของตัวแม่ผ่านสายคู่บิดเกลียวอีกเส้นหนึ่ง การติดต่อกันในระบบอุปกรณ์ทุกตัวจะต้องถูกกำหนดตำแหน่ง (address) ตัวแม่ติดต่อกับตัวลูกทุกตัวโดยตรงแต่ตัวลูกแต่ละตัวจะติดต่อกับตัวแม่ได้เท่านั้นไม่สามารถติดต่อกันเองได้ และในการติดต่อนั้นตัวลูกจะติดต่อกับตัวแม่ได้ที่ละตัว เมื่อตัวลูกตัวใดตัวหนึ่งกำลังติดต่อกับตัวแม่อยู่ ตัวลูกที่เหลือจะไม่สามารถติดต่อกับตัวแม่ได้และจะอยู่ในสถานะที่เรียกว่า ไตร-สเตท (tri-state) ซึ่งฮาร์ดแวร์บางผลิตภัณฑ์สามารถจัดการสถานะดังกล่าวได้โดยอัตโนมัติ แต่สำหรับบางกรณีจำเป็นต้องใช้ซอฟต์แวร์เป็นตัวจัดการ ผลจากการทำงานแบบไตร-สเตททำให้ต้องเสียเวลาระยะเวลาหนึ่งระหว่างการสิ้นสุดการส่งข้อมูลและการทำไตร-สเตทซึ่งช่วงเวลาดังกล่าวเรียกว่าเทิร์น-อราวด์ ดีเลย์ (turn-around delay) ในช่วงเวลานี้จะไม่มีตัวลูกตัวใดสามารถส่งข้อมูลได้ ซึ่งมีผลกับระบบแบบ 2 สาย แต่ไม่มีผลกับระบบแบบ 4 สาย ในทางอุดมคติแล้วระยะเวลาดังกล่าวควรมีค่าเท่ากับระยะของ

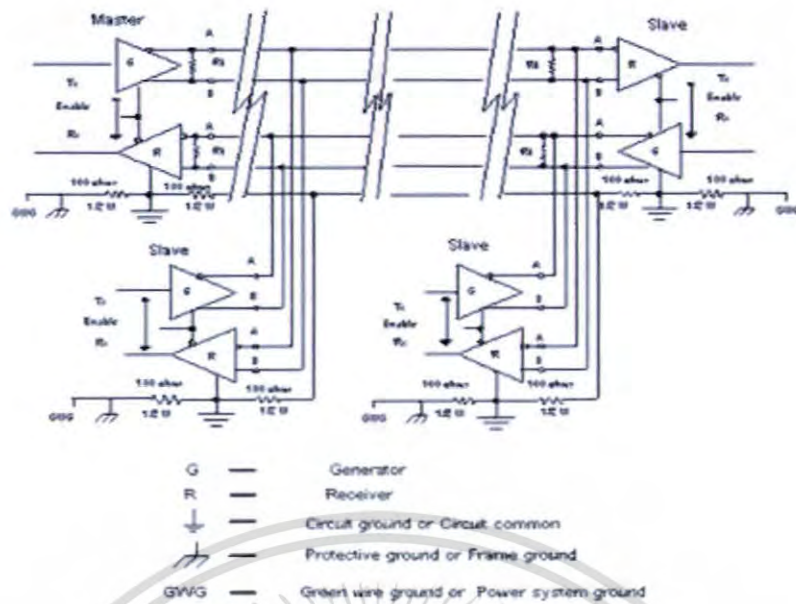
ความยาว 1 ตัวอักษรที่อัตราการบอดนั้นๆ เช่น ที่อัตราบอด 9,600 บิตต่อวินาทีควรมีค่าเทิร์น-อราวด์ ดีเลย์ เท่ากับ 1 มิลลิวินาที ซึ่งผู้ผลิตควรบอกค่าเทิร์น-อราวด์ ดีเลย์ของอุปกรณ์ไว้ด้วย

จากรูปที่ 2.61 แสดงระบบเครือข่ายที่เรียกการต่อระบบอาร์เอส-485 แบบ 2 สาย (two-wire multidrop) จากรูปสังเกตได้ว่าการต่อความต้านทานขั้วที่โหนดปลายทั้งสองด้านของสายส่งแต่ไม่มีการต่อขั้วปลายที่โหนดที่อยู่ระหว่างสายส่ง การต่อขั้วปลาย (termination) มักใช้กับระบบที่มีอัตราการส่งข้อมูลสูงและระยะทางยาว สำหรับสายกราวด์ที่มีความจำเป็นมากในระบบอาร์เอส-485



รูปที่ 2.61 แสดงระบบอาร์เอส-485 แบบ 2 สาย

แสดงระบบอาร์เอส-485 แบบ 4 สาย ประกอบด้วยสายสัญญาณ 4 เส้น และสายกราวด์ 1 เส้น มีลักษณะการติดต่อแบบ โหนดแม่ (master node) และ โหนดลูก (slave node) ตัวส่งของตัวแม่จะต่อถึงตัวรับของตัวลูกทุกตัวผ่านสายสัญญาณ(โดยมากใช้เป็นสายคู่ตีเกลียว) 1 คู่ และตัวส่งของตัวลูกทุกตัวจะต่อเข้ากับตัวรับของตัวแม่ผ่านสายสัญญาณอีก 1 คู่ โดยในการติดต่อนั้นตัวแม่จะสามารถติดต่อกับตัวลูกได้ทุกตัวแต่ตัวลูกทุกตัวจะติดต่อกับตัวแม่ได้เท่านั้น ไม่สามารถติดต่อกับตัวลูกตัวอื่นหรือติดต่อกันเองได้ และในการติดต่อจากตัวลูกไปยังตัวแม่นั้นจะทำได้ทีละ 1 ตัว เมื่อมีตัวใดตัวหนึ่งติดต่อหรือทำการส่งข้อมูลอยู่ ตัวลูกตัวอื่นที่เหลือในระบบจะไม่สามารถส่งข้อมูลได้ โดยต้องอยู่ในสถานะ disable หรือ ไตร-สเตทเนื่องจากตัวลูกจะไม่สนใจการติดต่อของตัวลูกตัวอื่นกับตัวแม่ ถือเป็นข้อดีคือทำให้สามารถต่ออุปกรณ์ที่มีโปรโตคอลต่างกันเข้าไว้ในระบบเดียวกันได้



รูปที่ 2.62 แสดงระบบอาร์เอส-485 แบบ 4 สาย

ตารางที่ 2.14 ข้อดี-ข้อเสียของการต่อระบบแบบ 2 สาย และ 4 สาย

	ระบบ 2 สาย	ระบบ 4 สาย
ข้อดี	<ul style="list-style-type: none"> <li>-ราคาในการเดินสายถูก</li> <li>-แต่ละโหนดสามารถติดต่อกันได้โดยตรง</li> </ul>	<ul style="list-style-type: none"> <li>-ต้องคำนึงถึงทึร์น-อราวด์ ดีเลย์</li> <li>-เนื่องจากการติดต่อบแบบแม่และลูก ลูกแต่ละตัวไม่สามารถติดต่อกันได้ ทำให้อุปกรณ์ที่มีโปรโตคอลต่างกันเข้าในระบบเดียวกันได้</li> </ul>
ข้อเสีย	ต้องคำนึงถึงทึร์น-อราวด์ ดีเลย์	

ข้อสังเกต ในการตั้งข้อสายเคเบิ้ล ในระบบ 2 สาย หมายถึง สายสัญญาณ 2 เส้นและสายกราวด์ 1 เส้น สำหรับระบบ 4 สาย หมายถึง สายสัญญาณ 4 เส้นและสายกราวด์ 1 เส้น

### 2.9.6 ความรู้เบื้องต้นเกี่ยวกับไมโครคอนโทรลเลอร์ dsPIC 30F4011

dsPIC คือชื่อของไมโครคอนโทรลเลอร์ 16 บิตจาก Microchip Technology Inc. ผู้ผลิตไมโครคอนโทรลเลอร์ PIC ซึ่งรู้จักกันเป็นอย่างดีในแวดวงนักพัฒนาระบบไมโครคอนโทรลเลอร์ โดย Microchip Technology ได้กำหนดชื่ออย่างเป็นทางการสำหรับไมโครคอนโทรลเลอร์อนุกรม

ใหม่นี้ว่า Digital Signal Controller หรือ DCS นั้นหมายความว่า dsPIC เป็นไมโครคอนโทรลเลอร์ที่ได้รับการออกแบบมาเป็นพิเศษเพื่องานประมวลผลสัญญาณดิจิทัลสำหรับสร้างระบบควบคุมอัตโนมัติที่มีความสามารถสูง

### คุณสมบัติเด่นโดยรวมของ dsPIC

#### คุณสมบัติของซีพียู

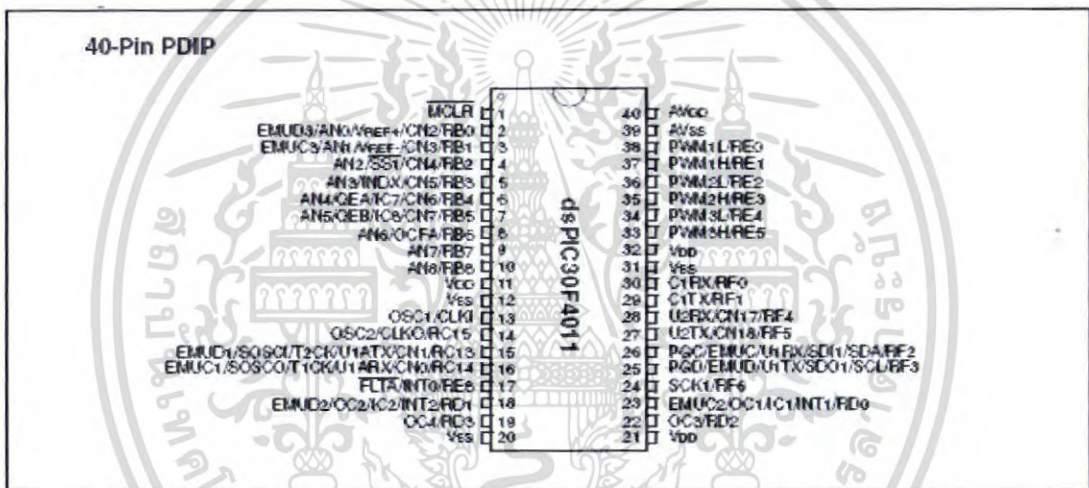
- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูแบบ RISC
- ความเร็วในการทำงานสูงถึง 30 ล้านคำสั่งต่อวินาที
- มี 84 คำสั่งภาษาแอสเซมบลีมาตรฐาน รองรับรูปแบบการอ้างแอดเดรสได้อย่างอิสระ
- ชุดคำสั่งมีขนาด 24 บิต สามารถประมวลผลข้อมูลได้ 16 บิต
- มีหน่วยความจำโปรแกรมเป็นแบบแฟลช สามารถลบและเขียนใหม่ได้ไม่น้อยกว่า 100,000 ครั้ง สามารถป้องกันการอ่านได้ และสามารถโปรแกรมตัวเอง โดยใช้กระบวนการทางซอฟต์แวร์
- มีหน่วยความจำข้อมูลอีพียูที่สามารเขียนและลบใหม่ได้ไม่น้อยกว่า 1,000,000 ครั้ง
- มีอินเตอร์รัปต์เวกเตอร์จำนวนมาก จึงรองรับการตอบสนองของสัญญาณอินเตอร์รัปต์ได้ดี
- มีวงจรตรวจจับแรงดันไฟเลี้ยงต่ำกว่ากำหนดแบบโปรแกรมได้
- มีวอตช์ด็อกไทมเมอร์แบบโปรแกรมได้
- มีวงจรตรวจสอบการทำงานของวงจรถูกกำเนิดสัญญาณนาฬิกา
- รองรับการโปรแกรมในวงจรแบบอนุกรม
- สามารถเลือกโหมดการใช้พลังงานได้

#### คุณสมบัติด้านการประมวลผลสัญญาณดิจิทัล

- มีแอดคิวมูเลเตอร์ขนาด 40 บิต 2 ตัวรองรับการประมวลผลทางคณิตศาสตร์ได้เป็นอย่างดี
- มีหน่วยประมวลผลด้านการคูณและหารเลข 17 บิตในรูปของฮาร์ดแวร์ จึงทำให้สามารถคูณและหารเลขได้อย่างรวดเร็ว
- ทำการคูณเลข 16 บิต ได้ภายในสัญญาณนาฬิกาเพียง 1 ไชเคิล
- มีตัวเลื่อนข้อมูลบาร์เรล 40 สเตจ ช่วยให้การประมวลผลข้อมูลที่มีจำนวนบิตมาก ๆ สามารถทำได้อย่างรวดเร็ว
- มีวงจรเพดซ์ข้อมูลคู่ จึงทำให้สามารถประมวลผลข้อมูลได้อย่างรวดเร็ว

### คุณสมบัติของไมโครฟังก์ชันพิเศษ

- สามารถจ่ายกระแสออกทางขาพอร์ต 25 mA ทั้งแบบกระแสซิงก์และซอร์ส
- ไทเมอร์/เคาน์เตอร์มีขนาด 16 บิต ไม่น้อยกว่า 3 ตัว ต่อใช้งานร่วมกันเป็นไทเมอร์ 32 บิตได้
- มีไมโครตรวจจับและเปรียบเทียบสัญญาณดิจิทัล
- มีส่วนเชื่อมต่ออุปกรณ์อนุกรมทั้งแบบ SPI และผ่านระบบบัส I2C
- มีไมโครสื่อสารข้อมูลอนุกรม UART พร้อมบัฟเฟอร์แบบ FIFO
- มีวงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัล ความละเอียด 10 หรือ 20 บิต
- มีไมโครสร้างสัญญาณ PWM สำหรับควบคุมมอเตอร์
- มีไมโครเชื่อมต่อตัวเข้ารหัสแบบควอดราเจอร์



รูปที่ 2.63 โครงสร้างของขาไมโครคอนโทรลเลอร์ dsPIC 30F4011

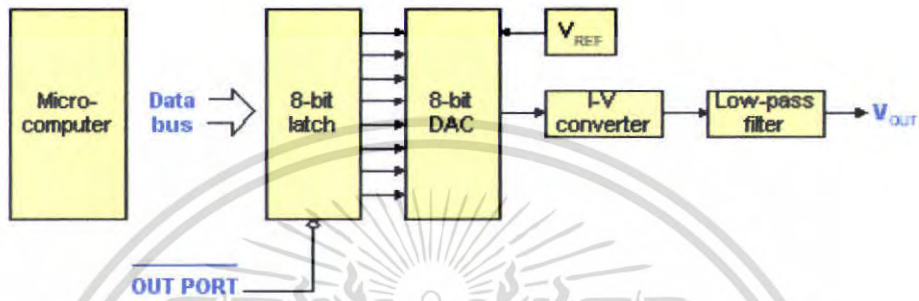
## 2.10 การแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกและการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล

### 2.10.1 การแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก (Digital to Analog Converter ,DAC)

อุปกรณ์ทางไฟฟ้า/อิเล็กทรอนิกส์ โดยทั่วไปที่เป็นอนาล็อก สามารถควบคุมการทำงานโดยการให้อินพุตเป็นระดับ แรงดันที่แตกต่างกัน ตัวอย่างเช่น มอเตอร์กระแส ตรง ซึ่งควบคุมความเร็ว โดยเปลี่ยนระดับแรงดัน (หรือกระแส) ของขดลวดสนาม

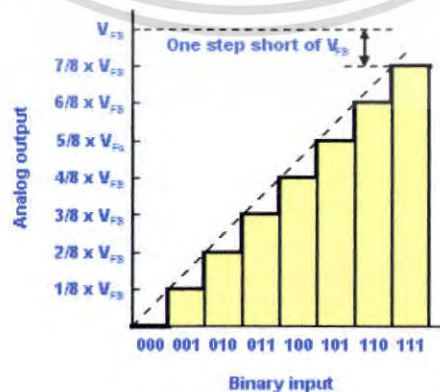
เมื่อนำระบบดิจิทัล หรือ ไมโครคอนโทรลเลอร์ มาใช้ควบคุม อุปกรณ์ทางอนาล็อกเหล่านี้ จึงต้องมีวงจรซึ่ง สามารถแปลงสัญญาณทางดิจิทัลเป็นระดับแรงดันต่อเนื่อง แบบอนาล็อก ตั้งแต่ศูนย์ โวลต์จนถึงระดับสูงสุดที่กำหนดไว้ เรียกว่าวงจร Digital to Analog Converter (DAC)

### 2.10.2 คุณสมบัติของอุปกรณ์แปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก



รูปที่ 2.64 ระบบการแปลงสัญญาณดิจิทัลเป็นอนาล็อก

ซึ่งรูปที่ 2.64 แสดงถึงส่วนประกอบหลักของระบบการแปลงสัญญาณดิจิทัลเป็นอนาล็อก โดยทั่วไป ไมโครคอมพิวเตอร์จะมีเอาต์พุตเป็นค่าไบนารี วงจรเลขรับค่าไบนารีเข้ามาเพื่อส่งไปยังอุปกรณ์แปลงสัญญาณดิจิทัลเป็นอนาล็อก ในวงจรจะใช้แหล่งกำเนิดแรงดันหรือ กระแสคงที่ เพื่ออ้างอิงในการแปลงข้อมูล ไบนารีเป็นระดับกระแส ต่อมาจะมีวงจรแปลงจากกระแสเป็นแรงดัน (current-to-voltage converter) ซึ่งปกติจะใช้โอปแอมป์ที่ขยายสัญญาณอนาล็อกที่ได้จะผ่าน วงจรกรองความถี่ต่ำ (low-pass filter) เพื่อกำจัดสัญญาณความถี่สูงที่แฝงอยู่ในสัญญาณที่ถูกสร้างขึ้นมา



รูปที่ 2.65 ทรานสเฟอร์ เคิร์ฟ ในอุดมคติของ DAC 3 บิต

รูปที่ 2.65 เป็นกราฟแสดงถึงความสัมพันธ์ระหว่าง เอาต์พุตที่เป็นอนาล็อกกับอินพุตที่เป็นดิจิตอลขนาด 3 บิตเรียกว่าทรานสเฟอร์ เคิร์ฟ (transfer curve) สังเกตว่าเมื่ออินพุตไบนารีเพิ่มขึ้น เอาต์พุตอนาล็อกจะเพิ่ม ในลักษณะขั้นบันได ขนาดของแต่ละขั้นจะ หาได้จาก

$$\text{stepsize} = V_{FS}/2^n$$

เมื่อให้  $V_{FS}$  คือระดับแรงดันเอาต์พุตสูงสุด

$n$  คือจำนวนบิตของอินพุต

เนื่องจากเอาต์พุตของ DAC จะเพิ่มเป็นขั้นๆ รูปคลื่นสัญญาณ ที่ได้จาก DAC จึงมีลักษณะไม่เรียบ ดังตัวอย่างในรูปที่ 2.66 ซึ่งแสดงถึงสัญญาณ ไชน์ ที่สร้างจาก DAC



รูปที่ 2.66 คลื่น ไชน์ ที่สร้างจาก DAC

ถ้าเพิ่มจำนวนบิต ความละเอียดของ DAC จะเพิ่มขึ้น เช่น เมื่อ ใช้ DAC 12 บิต และ  $V_{FS} = 5.0$  V ความละเอียดคือ  $5.0 \text{ V} / 4096 = 1.22 \text{ mV}$  ซึ่งจะ ละเอียดกว่า DAC 8 บิตถึง 16 เท่า ซึ่งความถูกต้องของ DAC ขึ้นอยู่กับหลายส่วน คือ

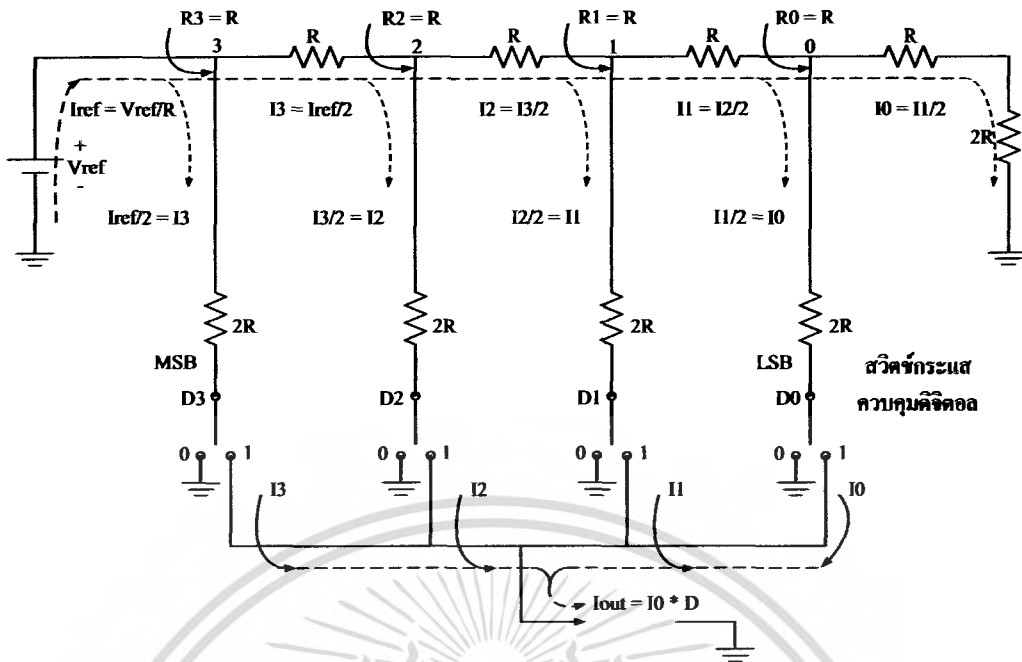
1. ค่าผิดพลาดควอนติเซชัน(Quantization error) ตัวอย่างเช่น DAC บิต  $V_{FS} = 5.0$  V เอาต์พุตจะมีความละเอียด 19.53 mV ถ้าต้องการเอาต์พุต 4.00 V DAC จะให้อาต์พุตได้ใกล้เคียง ที่สุดคือ 4.04 V (19.53 mV x 205) ผิดพลาด 4 mV โดยทั่วไปค่าผิดพลาดจะเท่ากับ +/- 0.5 LSB (least significant bit) ตัวอย่างเช่น DAC 8 บิต ความผิดพลาดจะเป็น 1 ใน 512 หรือ +/- 0.195 %

2. ความผิดพลาดของค่าออฟเซตและเกน(Offset and gain errors) ตัวอย่างเช่น เมื่ออินพุตไบนารีเท่ากับ 0 แต่เอาต์พุตของ DAC ไม่เป็น 0 เรียกว่า offset error และอาจเกิดร่วมกับ gain error ความผิดพลาดเหล่านี้จะทำให้ทรานสเฟอร์ เวิร์ฟในรูปที่ 2.65 โค้งขึ้น หรือลง ขึ้นอยู่กับความไม่สมดุลภายใน DAC อย่างไรก็ตาม ความผิดพลาดของค่าออฟเซตและเกนจะแก้ไขได้โดยใช้ความต้านทานปรับค่าได้ต่อไว้ภายนอก
3. ความไม่เป็นเชิงเส้น(Nonlinearity) คือค่าความคลาดเคลื่อนสูงสุดของทรานสเฟอร์ เวิร์ฟเทียบกับเส้นตรงจากจุดศูนย์และจุดสูงสุด ซึ่งจะขึ้นอยู่กับความผิดพลาดของส่วนประกอบภายใน DAC ใน data sheet ของ DAC จะระบุเป็นเปอร์เซ็นต์เทียบกับค่าสูงสุด หรือ ระบุเป็นเศษส่วนของ LSB (โดยทั่วไปคือ +/- 0.5 LSB)
4. เซตติง ไทม์(Settling time) คือช่วงเวลานับแต่ให้อินพุตจนกระทั่ง DAC ให้ เอาต์พุต วัตเมื่อเอาต์พุตที่ได้ผิดพลาดจากค่าจริง น้อยกว่า 0.5 LSB ค่าเวลานี้อาจน้อยกว่า 100 นาโนวินาทีสำหรับ DAC ความเร็วสูง และอาจมากกว่า 100 ไมโครวินาทีสำหรับ DAC ราคาถูก

### 2.10.3 วงจรแลตเตอร์ R-2R

วงจรแลตเตอร์ R-2R แสดงดังรูปที่ 2.67 สัญญาณดิจิตอลอินพุตที่ป้อนให้กับวงจรมีตำแหน่งควบคุมสวิตช์กระแสตรงตามตำแหน่งบิตนั้นๆ เมื่อทำการพิจารณาค่าความต้านทานเทียบเท่าที่มองจากโหนด 0 ( $R_0$ ) เข้าไปจะมีค่าเท่ากับ  $2R//2R = R$  และพิจารณาค่าความต้านทานเทียบเท่าที่มองจากโหนด 1 ( $R_1$ ) เข้าไปจะได้เท่ากับ  $2R//(R+R)$  เช่นเดียวกันค่าความต้านทานเทียบเท่าที่มองจากโหนด 2 ( $R_2$ ) และ โหนด 3 ( $R_3$ ) เข้าไปต่างก็มีค่าเท่ากับ R เช่นกัน สรุปได้ว่า  $R_0 = R_1 = R_2 = R_3 = R$  ทำให้กระแสอ้างอิง  $I_{REF}$  มีค่าประมาณ

$$I_{REF} = \frac{V_{REF}}{R_3} = \frac{V_{REF}}{R}$$



รูปที่ 2.67 วงจรแลดเดอร์ R-2R แปลงสัญญาณดิจิทัลให้เป็นกระแสเอาต์พุต Iout

ที่โหนด 3 กระแสอ้างอิง  $I_{REF}$  จะไหลแยกออกไปสองส่วนมีค่าเท่ากัน โดยมีค่าเท่ากับ

$$I_3 = \frac{I_{REF}}{2}$$

และเมื่อทำการพิจารณากระแสที่ไหลแยกออกจากโหนดแต่ละโหนด จะได้ความสัมพันธ์ของกระแสในแต่ละสาขา ดังนี้คือ

$$I_2 = \frac{I_3}{2} = \frac{I_{REF}}{4}$$

$$I_1 = \frac{I_2}{2} = \frac{I_{REF}}{8}$$

$$I_0 = \frac{I_1}{2} = \frac{I_{REF}}{16}$$

ผลรวมของกระแสเอาต์พุต Iout ที่โหนดเอาต์พุตเมื่อสวิตช์กระแสทุกบิตอยู่ในตำแหน่ง “1” มีค่าเท่ากับ

$$I_{OUT} = I_3 + I_2 + I_1 + I_0 = I_0 \times \left( 1 + \frac{I_1}{I_0} + \frac{I_2}{I_0} + \frac{I_3}{I_0} \right)$$

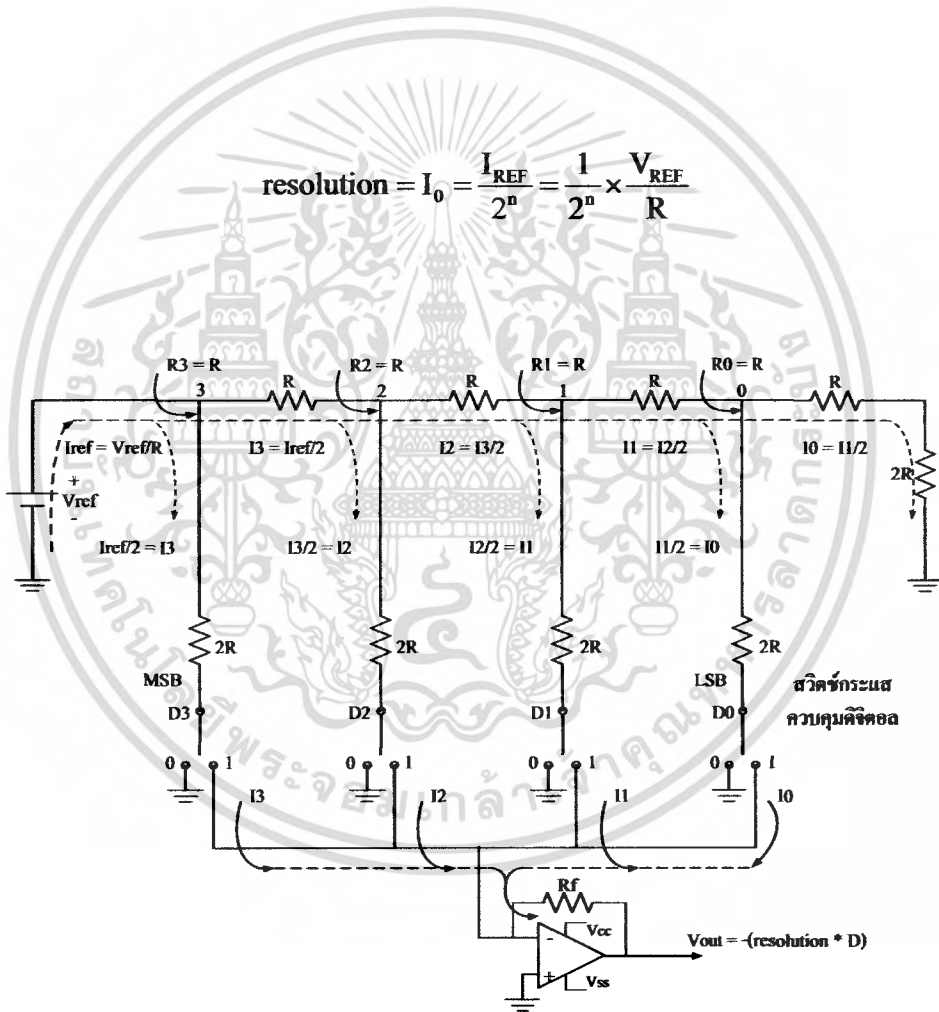
โดยที่  $D$  มีค่าเท่ากับเลขฐานสิบของสัญญาณดิจิทัลอินพุต และ  $I_0$  คือค่ากระแสต่ำสุดในวงจรแลคเคอร์ เมื่อกำหนดให้  $I_0$  คือค่ารีโซลูชันของวงจรแลคเคอร์ ดังนั้นสมการอินพุต-เอาต์พุตของวงจร

$$I_{OUT} = I_0 \times (2^0 + 2^1 + 2^2 + 2^3) = I_0 \times D$$

สามารถเขียนอธิบายได้เป็น

$$I_{OUT} = \text{resolution} \times D$$

โดยที่



รูปที่ 2.68 การเปลี่ยนแปลงกระแสเอาต์พุต  $I_{out}$  ของวงจรแลคเคอร์ R-2R ให้เป็นแรงดันเอาต์พุต

#### 2.10.4 แรงดันอนาล็อกเอาต์พุทของวงจร DAC (Voltage Output DAC)

กระแสอนาล็อกเอาต์พุท  $I_{out}$  ของวงจร DAC ในรูปที่ 2.54 สามารถเปลี่ยนให้อยู่ในรูปของแรงดันอนาล็อกเอาต์พุท  $V_{out}$  ได้โดยการใช้โอปแอมป์ต่อรวมตัวต้านทานป้อนกลับ  $R_f$  เพิ่มเข้าไปในวงจรเพื่อทำหน้าที่เปลี่ยนกระแสให้เป็นแรงดัน ดังรูปที่ 2.69

แรงดันเอาต์พุทของวงจร  $V_{out}$  มีค่าเท่ากับ

$$V_{OUT} = -I_{OUT} R_f$$

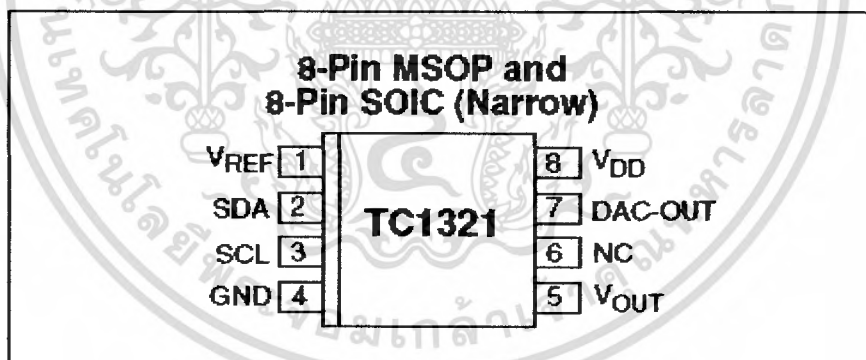
ดังนั้นจะได้

$$V_{OUT} = -(\text{current resolution} \times D) \times R_f$$

และค่ารีโซลูชันของวงจร DAC ได้ดังนี้

$$\text{resolution} = I_0 R_f$$

#### 2.10.5 การแปลงสัญญาณดิจิทัลเป็นอนาล็อกด้วยไอซี ทีซี 1321 (TC1321)



รูปที่ 2.69 โครงสร้างขาของตัวแปลงสัญญาณดิจิทัลเป็นอนาล็อก TC 1321

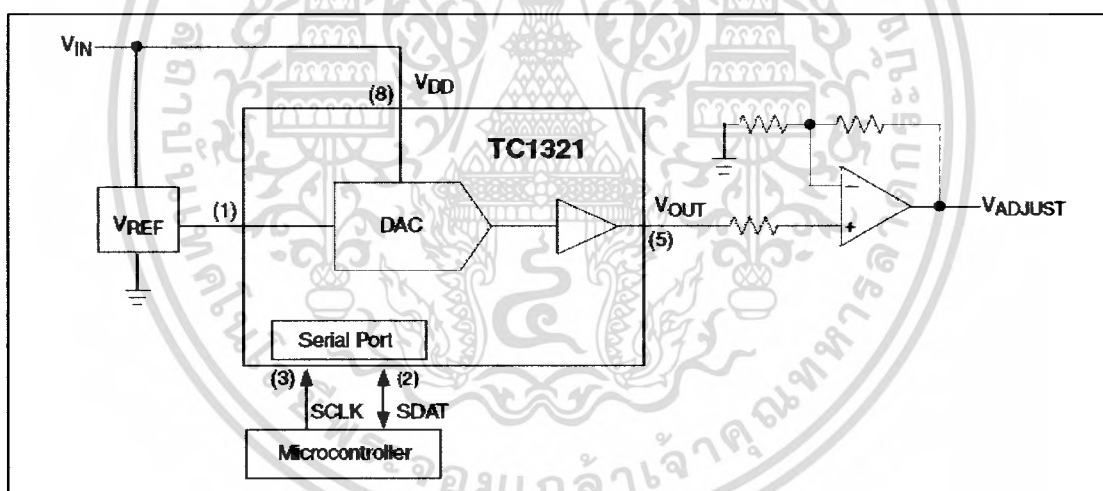
## ลักษณะทั่วไปของ TC 1321

เป็นโมดูลแปลงสัญญาณดิจิทัลเป็นอนาล็อกที่มีความละเอียด 10 บิต มีแรงดันเอาพุทอ้างอิงและทำงานในช่วงระหว่าง 2.7-5.5 โวลท์ การติดต่อสื่อสารนั้นจะติดต่อผ่านทางบัสและไอสแควร์ซี (SMBus/I<sup>2</sup>C)

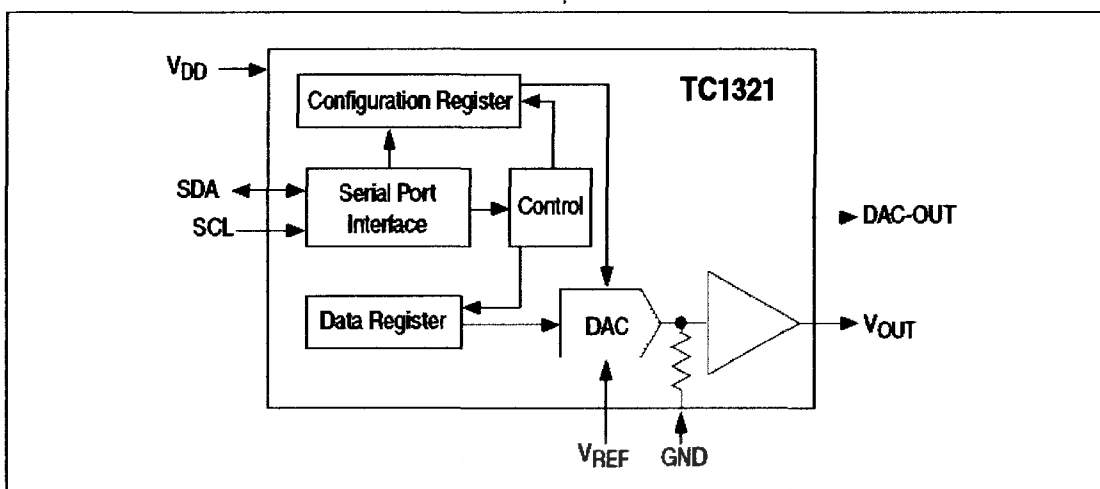
## ลักษณะโครงสร้างของ TC1321

- เป็น โมดูลแปลงสัญญาณดิจิทัลเป็นอนาล็อกที่มีความละเอียด 10 บิต
- กินกระแสก่อนข้างน้อยเพียง 350 ไมโครแอมแปร์ ( $\mu\text{A}$ )
- ทำงานในช่วงระหว่าง 2.7-5.5 โวลท์ V
- ติดต่อผ่านทาง SMBus/I<sup>2</sup>C
- มี 8 ขามีแบบ SOIC และ แบบ MSOP

## บล็อกไดอะแกรมของ TC 1321



รูปที่ 2.70 บล็อกไดอะแกรมของ TC 1321



รูปที่ 2.71 รูปแบบการประยุกต์ใช้งาน ของ TC 1321

### การประยุกต์ใช้งาน

- สามารถขยายสัญญาณและลดทอนขนาดของสัญญาณ
- สามารถโปรแกรมแหล่งจ่ายได้

### รายละเอียดของแต่ละขา

ตารางที่ 2.15 รายละเอียดของขา TC 1321

Pin Number	Symbol	Description
1	Vref	อินพุตแรงดันอ้างอิง
2	SDA	ส่งค่าข้อมูลได้ทั้งไปทั้งกลับ
3	SCL	ส่งสัญญาณนาฬิกาทั้งขาเข้าและออก
4	GND	ขากราวด์
5	Vout	ขา Output
6	NC	ขานี้ไม่ได้ใช้
7	DAC-OUT	ขา Output ของ D/A
8	Vdd	ขาต่อไฟเลี้ยง (+)

### 2.10.6 การแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล

ในปัจจุบันมีไมโครโปรเซสเซอร์ และไมโครคอนโทรลเลอร์ เข้ามาช่วยในการควบคุม อุปกรณ์ต่างๆ มากมาย ซึ่งทำให้การควบคุมนั้นทำได้ง่ายและรวดเร็วยิ่งขึ้น แต่ในการควบคุมนั้น เราจำเป็นต้องใช้ สัญญาณดิจิทัลในการติดต่อกับไมโครโปรเซสเซอร์ หรือไมโครคอนโทรลเลอร์ แต่ในความเป็นจริงนั้น เราใช้สัญญาณอนาล็อกในการควบคุม ดังนั้นเราจึงจำเป็นต้องมีการเปลี่ยนสัญญาณอนาล็อก เป็นสัญญาณดิจิทัล แล้วจึงนำสัญญาณนั้นเข้ามาสู่ไมโครโปรเซสเซอร์ หรือไมโครคอนโทรลเลอร์เพื่อใช้ควบคุมระบบต่อไป

### 2.10.7 คุณสมบัติของตัวแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล (Analog to Digital Converter , ADC)

มีส่วนที่สำคัญอยู่ 2 ประการคือ

#### 1. สมการอินพุท-เอาต์พุท

สัญลักษณ์และคุณสมบัติที่ระหว่างสัญญาณอนาล็อกอินพุทกับสัญญาณดิจิทัลเอาต์พุทของ วงจร ADC ขนาด 4 บิต แสดงดังรูปที่ 2.56 ประกอบด้วยช่องรับสัญญาณอนาล็อกอินพุท 1 ช่อง สัญญาณและช่องจ่ายสัญญาณดิจิทัลเอาต์พุทจำนวน 4 ช่องซึ่งแสดงถึงลักษณะของวงจร ADC ขนาด 4 บิต และในทำนองเดียวกันกับวงจร DAC คำรีโซลูชันของวงจร ADC สามารถกำหนดความหมายได้สองลักษณะเช่นกัน ดังนี้คือ

- คำรีโซลูชันของวงจร ADC ขนาด  $n$  บิต มีค่าขึ้นกับจำนวนบิตสูงสุดของสัญญาณดิจิทัลเอาต์พุท ซึ่งสามารถเขียนอธิบายได้ว่า

$$\text{resolution} = 2^n$$

- คำรีโซลูชัน หมายถึง อัตราส่วนการเปลี่ยนแปลงของสัญญาณอนาล็อกอินพุท  $V_{in}$  ที่ทำให้สัญญาณดิจิทัลเอาต์พุทเปลี่ยนแปลงไปหนึ่ง บิตนับสำคัญต่ำสุด ดังนั้นหากทราบค่าแรงดันอนาล็อก

อินพุทเต็มสเกล,  $V_{IFS}$  ที่ทำให้สัญญาณดิจิทัลเอาต์พุทมีสถานะลอจิกเป็น “1” ทั้งหมดทุกบิตแล้ว จะสามารถคำนวณหาค่ารีโซลูชันของวงจรได้เท่ากับ

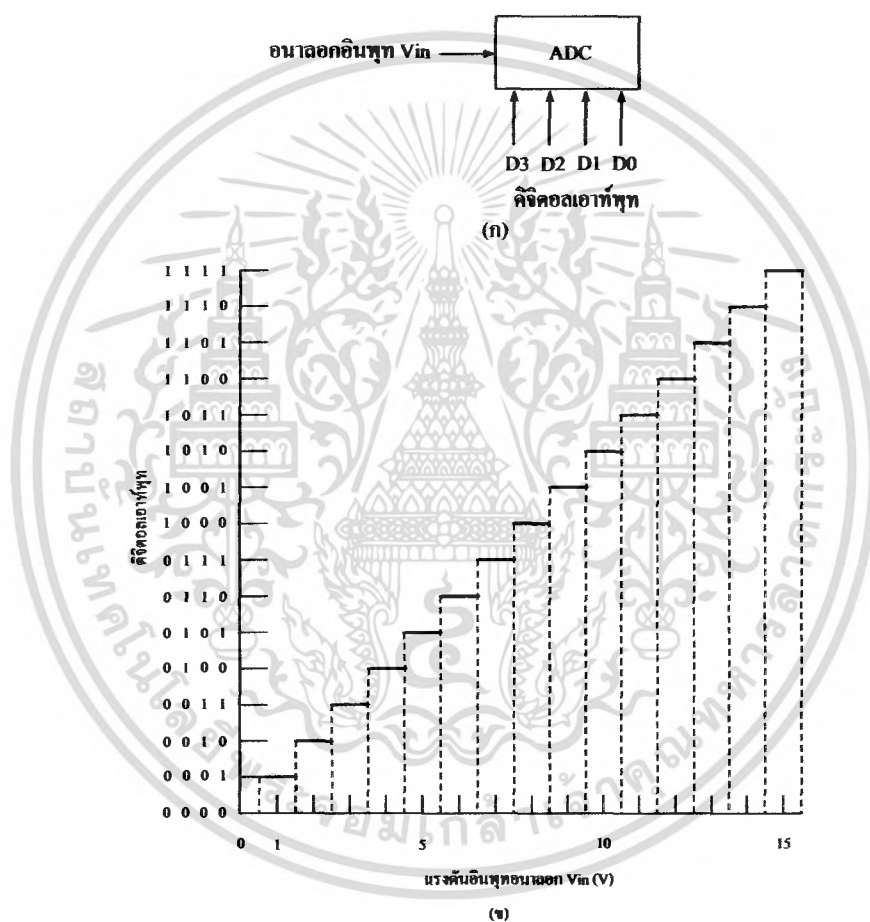
$$\text{resolution} = \frac{V_{IFS}}{2^n - 1}$$

สำหรับสมการอินพุท-เอาต์พุทของวงจร ADC สามารถหาได้จากความสัมพันธ์ ดังนี้

สัญญาณดิจิทัลเอาต์พุต = เลขฐานสองของ D

เมื่อ D มีค่าเท่ากับเลขฐานสิบของสัญญาณดิจิทัลเอาต์พุต หรือ D มีค่าเท่ากับจำนวนบิตนัยสำคัญต่ำสุดของสัญญาณดิจิทัลเอาต์พุต ซึ่ง D สามารถหาได้จาก

$$D = \frac{V_{in}}{\text{resolution}}$$



รูปที่ 2.72 วงจร ADC ขนาด 4 บิต

(ก) สัญลักษณ์ของวงจร (ข) กราฟคุณสมบัติระหว่างอินพุตกับเอาต์พุต

## 2. ความคลาดเคลื่อนควอนไทเซชัน (Quantization Error)

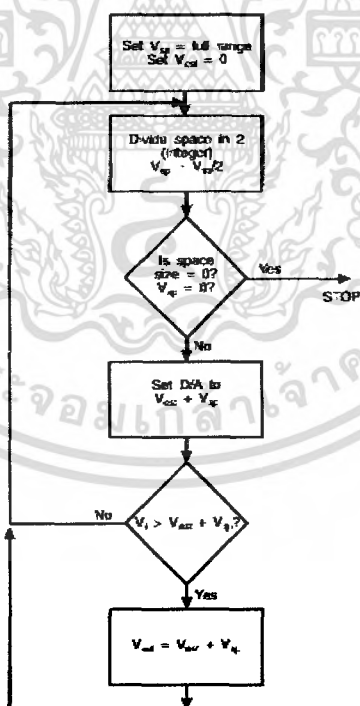
จากรูปที่ 2.56 จะสังเกตเห็นว่าสัญญาณดิจิทัลเอาต์พุตมีค่าเท่ากับ 0101 ในทุกๆ ค่าของแรงดันอนาล็อกอินพุตที่มีค่าอยู่ระหว่าง 4.5V ถึง 5.5V ซึ่งเป็นค่าความคลาดเคลื่อนที่เกิดขึ้นใน

กระบวนการแปลงสัญญาณ และจะเรียกความคลาดเคลื่อนดังกล่าวนี้ว่า “ความคลาดเคลื่อนควอนทิเซชัน” มีค่าเท่ากับ  $+1/2$  บิตนัยสำคัญต่ำสุด การลดผลความผิดพลาดเนื่องจากค่าความคลาดเคลื่อนควอนทิเซชันนี้สามารถกระทำได้โดยการเพิ่มจำนวนบิตของวงจรให้มากขึ้นเพื่อให้ค่ารีโซลูชันมีความละเอียดมากขึ้นนั่นเอง

### 2.10.8 การประมาณค่าโดยวิธีซัคเซสซีฟ (Successive Approximation)

ใช้หลักการของ "binary search" ในการหาคำตอบ โดยนำค่าผลลัพธ์มาเปรียบเทียบกับค่ากึ่งกลางของช่วง เพื่อให้ทราบว่า คำนี้นั้น มากกว่า หรือน้อยกว่า โดยจะปรับช่วงให้แคบลงมาเรื่อยๆ แล้วเปรียบเทียบผลลัพธ์กับค่ากึ่งกลางของช่วงไปเรื่อยๆ จนได้ผลลัพธ์ที่ต้องการ เช่น เลขที่เป็นคำตอบคือ 3 จากช่วงของคำตอบที่ 0-7 ครั้งแรกเอาค่า  $(0+7)/2 = 4$  มาเปรียบเทียบ ได้ผลว่า คำตอบที่ต้องการอยู่ในช่วงที่น้อยกว่า 4 ครั้งที่ 2 ก็เลือกค่า  $(0+4)/2 = 2$  มาเปรียบเทียบ ได้ผลว่าคำตอบที่ต้องการอยู่ในช่วงที่มากกว่า 2 แต่น้อยกว่า 4 ครั้งที่ 3 ก็เลือกค่า  $(2+4)/2 = 3$  มาเปรียบเทียบ ได้ผลว่าคำตอบที่ต้องการ

จากหลักการที่กล่าวมาอาจเขียนโฟลวชาร์ต(flow chart) ได้ดังนี้



Binary Search Strategy

รูปที่ 2.73 โฟลวชาร์ตวิธีการทำงานของการประมาณค่าโดยวิธีซัคเซสซีฟ

ข้อดีของวิธีนี้ คือ เวลาที่ใช้ในการหาคำตอบ  $n$  รอบแน่นอน (สำหรับ  $n$  bit converter ซึ่งอ้างอิงได้  $2^n$  ระดับ และระดับ  $V_{in}$  ที่คงที่)

แต่มีข้อเสีย คือ ถ้า  $V_m$  เปลี่ยนทันทีทันใด ขณะที่กำลังทำ binary search อยู่ นั่น คำตอบที่ได้จะผิดพลาด



## บทที่ 3

# การออกแบบและโครงสร้าง

### 3.1 การออกแบบดีซีเอส (DCS) ที่ใช้ในการควบคุมระดับน้ำ

ในการทำโปรเจกต์เป็นการนำหลักการ และ โครงสร้างของดีซีเอสตามที่ได้กล่าวมาแล้วข้างต้นเพื่อนำมาใช้ออกแบบระบบควบคุม โดยที่กระบวนการผลิตนั้นมีอยู่สองกระบวนการด้วยกัน เป็นการควบคุมระดับน้ำอันดับหนึ่ง และควบคุมระดับน้ำอันดับสอง การออกแบบระบบควบคุมโดยใช้ดีซีเอสตามโครงสร้าง ก็จะมีอยู่สองส่วนด้วยกันคือ หน่วยควบคุมกระบวนการ จะติดตั้งอยู่ที่กระบวนการทั้งสองกระบวนการ และหน่วยติดต่อและปฏิบัติการของพนักงานจะติดตั้งอยู่ในห้องควบคุม โดยที่ทั้งสองส่วนติดต่อถึงกันโดยเครือข่ายคอมพิวเตอร์ แต่เนื่องจากกระบวนการของเราเป็นกระบวนการที่เล็กมีเครื่องมือวัดและควบคุมไม่มาก หน่วยควบคุมกระบวนการจึงใช้เป็นไมโครคอนโทรลเลอร์(Microtroller) หนึ่งตัวต่อหนึ่งกระบวนการ ส่วนหน่วยติดต่อและปฏิบัติการก็ใช้เป็นคอมพิวเตอร์ในการติดต่อกับหน่วยควบคุมกระบวนการ โดยคอมพิวเตอร์หน่วยติดต่อและปฏิบัติการและหน่วยควบคุมกระบวนการติดต่อถึงกันโดยใช้มาตรฐานอาร์เอส-485 (RS-485)เพื่อรับส่งข้อมูลกัน

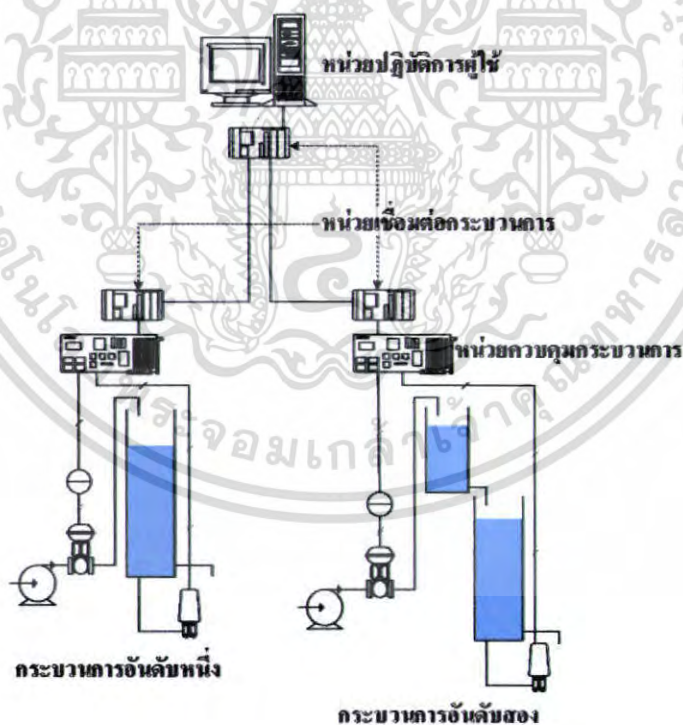
หน่วยติดต่อและปฏิบัติการของพนักงานได้ ใช้คอมพิวเตอร์หนึ่งเครื่องเพื่อ ทำหน้าที่รับค่าและแสดงผลออกมาให้ผู้ใช้งานทราบ โดยจอภาพจะแสดงภาพจำลองของกระบวนการทั้งสองกระบวนการ จะแสดงเป็นภาพเคลื่อนไหวบอกระดับของน้ำในถังจะบอกเป็นเปอร์เซ็นต์ ผู้ใช้สามารถปรับระดับน้ำในถังตามที่ต้องการได้โดยใส่ค่าของระดับน้ำที่ต้องการรักษาระดับ และผู้ใช้ก็สามารถปรับค่าพีไอดี (PID) ที่ใช้ในการควบคุมเพื่อให้ระบบควบคุมทำการควบคุมระดับน้ำได้ดีและรวดเร็วขึ้นเมื่อรับค่าต่างๆเสร็จแล้ว ก็จะส่งค่าที่ได้รับนี้ไปให้ไมโครคอนโทรลเลอร์ที่ทำหน้าที่เป็นหน่วยควบคุมกระบวนการเพื่อประมวลผลและส่งสัญญาณควบคุมไปควบคุมกระบวนการต่อไป

หน่วยควบคุมกระบวนการ ในหน่วยนี้จะมีไมโครคอนโทรลเลอร์หนึ่งตัวต่อหนึ่งกระบวนการเป็นตัวทำหน้าที่ประมวลผลและส่งสัญญาณควบคุมออกไปควบคุมกระบวนการ โดยที่ไมโครคอนโทรลเลอร์แต่ละตัวจะควบคุมกระบวนการแต่ละกระบวนการพร้อมกัน โดยสามารถส่งสัญญาณควบคุมได้พร้อมกันอีกด้วย หน่วยควบคุมกระบวนการนี้จะรับค่าต่างๆที่ใช้ในการควบคุมเช่น ค่าพีไอดี ค่าปรับตั้ง(setpoint)และรับค่าจากเครื่องมือวัดในกระบวนการ นำค่าทั้งหมดมาประมวลผล โดยใช้ตัวประมวลผลแบบ พีไอดี เมื่อประมวลผลเสร็จก็จะส่งสัญญาณที่ได้กลับไปควบคุมกระบวนการ และส่งสัญญาณนี้ไปที่หน่วยติดต่อผู้ใช้ด้วยเพื่อแสดงผลออกมาให้ผู้ใช้งานทราบ

ในหน่วยควบคุมกระบวนการนี้สามารถรับค่าต่างๆที่ใช้ในการควบคุมจากคอมพิวเตอร์ในหน่วยติดต่อผู้ใช้งาน

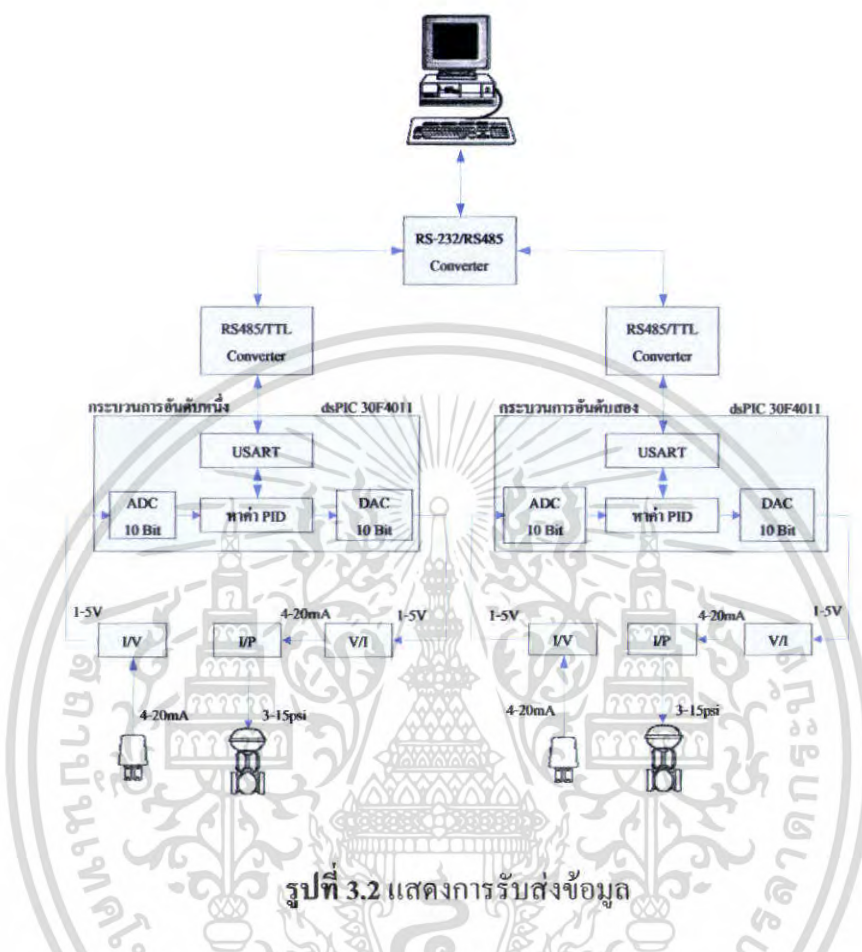
การติดต่อกันระหว่างคอมพิวเตอร์กับเครื่องมือต่างๆในกระบวนการจะติดต่อกันผ่านหน่วยเชื่อมต่อกระบวนการ

หน่วยเชื่อมต่อกระบวนการ ทำหน้าที่รับค่าจากเครื่องมือวัด (เช่น เซอร์วอเตอร์ระดับน้ำและวัดอัตราการไหลของน้ำ) เป็นกระแสไฟฟ้า 4 - 20 มิลลิแอมป์แปร จากนั้นแปลงให้อยู่ในรูปของแรงดันคือ 1 - 5 โวลท์ที่เป็นสัญญาณอนาล็อกมาแปลงค่าเป็นสัญญาณดิจิทัล เพื่อส่งให้คอมพิวเตอร์ประมวลผล และรับค่าจากคอมพิวเตอร์ที่เป็นดิจิทัลมาแปลงกลับเป็นอนาล็อกเพื่อส่งออกไปควบคุมวาล์วเปิด-ปิดน้ำ หน่วยเชื่อมต่อกระบวนการนี้จะมีด้วยกันสองชุดเพราะมีกระบวนการอยู่สองกระบวนการการแปลงสัญญาณอนาล็อกเป็นดิจิทัลใช้คุณสมบัติในไมโครคอนโทรลเลอร์ dsPIC 30F4011 ในการแปลงอนาล็อกเป็นดิจิทัลขนาด 10 บิต และแปลงดิจิทัลเป็นอนาล็อกในส่วนนี้เราได้ใช้ IC TC 1321 ที่มีขนาด 10 บิต มีรายละเอียดดังที่กล่าวไปในบทที่ 2 ออกมาเป็นในรูปแบบของแรงดัน หน่วยเชื่อมต่อกระบวนการทั้งสองตัวนี้จะติดอยู่ในกระบวนการทั้งสองกระบวนการ



รูปที่ 3.1 แสดงรูปแบบการติดต่อที่ออกแบบ

### 3.2 โครงสร้างการรับส่งข้อมูล ระหว่างคอมพิวเตอร์ควบคุมกระบวนการ กับ หน่วยเชื่อมต่อกระบวนการ



รูปที่ 3.2 แสดงการรับส่งข้อมูล

ตัวแปลงสัญญาณจากอาร์เอส-232 เป็นอาร์เอส-485 (RS-232/RS-485 Converter)

ทำหน้าที่แปลงข้อมูลไปกลับระหว่างการสื่อสารแบบ อาร์เอส-232 กับ อาร์เอส-485 เพื่อให้สามารถสื่อสารได้ในระยะทางที่ไกลขึ้น

ตัวแปลงสัญญาณจากอาร์เอส-485เป็นทีทีแอล (RS-232/TTL Converter)

ทำหน้าที่แปลงข้อมูลไปกลับระหว่างการสื่อสารแบบ อาร์เอส-485 กับ สัญญาณทีทีแอล เพื่อให้ไมโครคอนโทรลเลอร์รับรู้ค่าที่ส่งมาจากคอมพิวเตอร์หน่วยติดต่อและปฏิบัติการของพนักงาน

dsPIC 30F4011

ทำหน้าที่เป็นอุปกรณ์ UART ในการรับค่าในการควบคุมมาจากหน่วยติดต่อและปฏิบัติการของพนักงาน อีกทั้งยังเป็นตัวควบคุมกระบวนการอีกด้วย

การแปลงสัญญาณอนาล็อกเป็นดิจิทัล (Analog to digital หรือ A/D)

อุปกรณ์ การแปลงสัญญาณอนาล็อกเป็นดิจิทัล ใช้คุณสมบัติที่มีอยู่ใน dsPIC 30F4011 เป็นอุปกรณ์แปลงสัญญาณอนาล็อกเป็นดิจิทัล ขนาด 10 บิต

การแปลงสัญญาณดิจิทัลเป็นอนาล็อก (Digital to Analog หรือ D/A)

อุปกรณ์ การแปลงสัญญาณดิจิทัลไปเป็นดิจิทัล ในโครงงานนี้ใช้ไอซีเบอร์ ที่ซี 1321(TC 1321) เป็นอุปกรณ์การแปลงสัญญาณดิจิทัลไปเป็นดิจิทัล

อุปกรณ์แปลงแรงดันเป็นกระแสไฟฟ้า (V/I)

ในส่วน V/I จะเป็นอุปกรณ์ที่ใช้ในการแปลงค่าของแรงดันไฟฟ้าให้ไปเป็นกระแสไฟฟ้า จาก 1-5 โวลต์เป็น 4-20 มิลลิแอมป์แปรเพื่อที่ใช้ในการส่งสัญญาณได้ไกลขึ้น

อุปกรณ์แปลงกระแสเป็นแรงดันไฟฟ้า (I/V)

ในส่วน I/V จะเป็นอุปกรณ์ที่ใช้ในการแปลงค่าของกระแสไฟฟ้าให้เป็นแรงดันไฟฟ้าจาก 4-20 มิลลิแอมป์แปรเป็น 1-5 โวลต์

อุปกรณ์แปลงกระแสไฟฟ้าเป็นแรงดันลม (I/P)

ในส่วน I/P จะเป็นอุปกรณ์ที่ใช้ในการแปลงค่าของกระแสไฟฟ้าให้เป็นลมจาก 4-20 มิลลิแอมป์แปรเป็น 3-15 ปอนด์ต่อตารางนิ้วเพื่อใช้กับวาล์วควบคุมและตัววัดระดับน้ำ

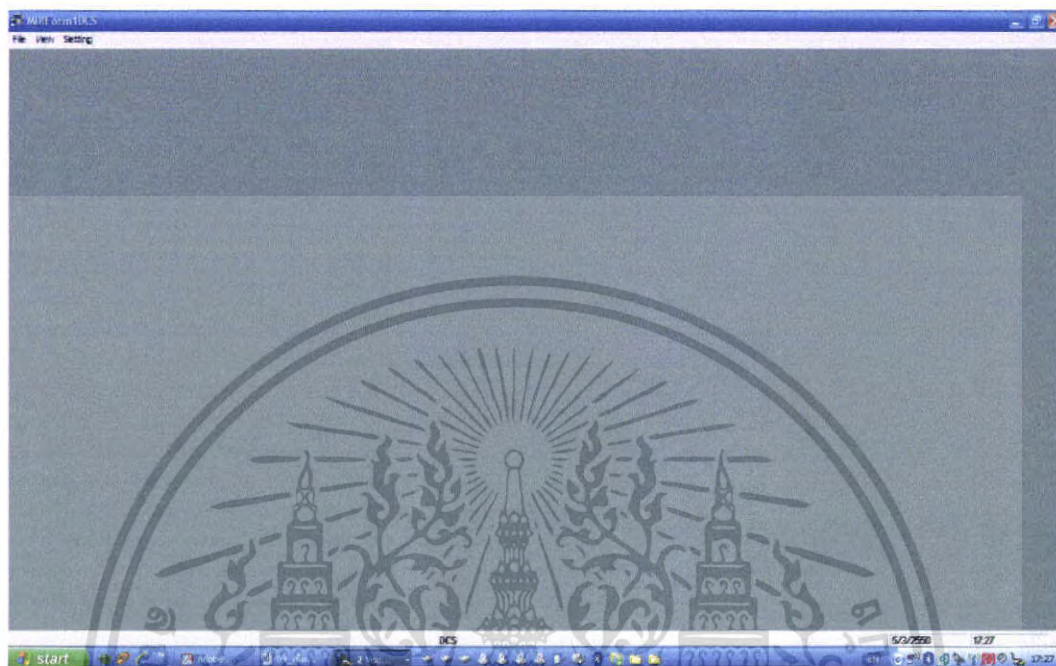
### 3.3 รูปแบบโปรแกรมของคอมพิวเตอร์หน่วยติดต่อและปฏิบัติการของพนักงาน

ส่วนนี้เป็นลักษณะการออกแบบ โปรแกรมติดต่อของพนักงานผู้ใช้กับตัวระบบดิจิทัล โดยผ่านโปรแกรม วิชาการเบสิก เมื่อต้องการควบคุมกระบวนการที่อยู่ในระบบส่วนแรกที่ต้องเจอเป็นรูปแบบการตั้งค่าของการสื่อสารมีลักษณะดังรูปที่ 3.3



รูปที่ 3.3 แสดงการกำหนดรูปแบบการสื่อสารของระบบดิจิทัล

ถ้าผู้ใช้ต้องการค่ามาตรฐานของระบบส่วนของโปรแกรมมีรูปแบบการใช้งานที่ง่ายโดยกดปุ่ม Default จากนั้นก็กดปุ่ม OK จากนั้นจะเข้าสู่หน้าต่างโปรแกรมหลักมีลักษณะดังรูปที่ 3.4



รูปที่ 3.4 แสดงหน้าต่างหลักของโปรแกรมติดต่อของพนักงานผู้ใช้

จากรูปที่ 3.4 มีส่วนประกอบอยู่ 2 ส่วนหลักๆ คือ ส่วนเมนูด้านบน และส่วนแสดงสถานะด้านล่าง ส่วนของเมนูด้านบนประกอบด้วย 4 เมนูคือ

#### 1. File

- Start
- Stop
- Exit

#### 2. View

- Group Display
- Plant Fist Order
- Plant Second Order
- Recorder Plant Fist Order
- Recorder Plant Second Order

#### 3. Settings

ส่วนแสดงสถานะด้านล่างประกอบด้วย 3 ส่วนคือ

1. สถานะคิชีเอสกำลังใช้งานอยู่
2. สถานะของวันที่
3. สถานะของเวลา

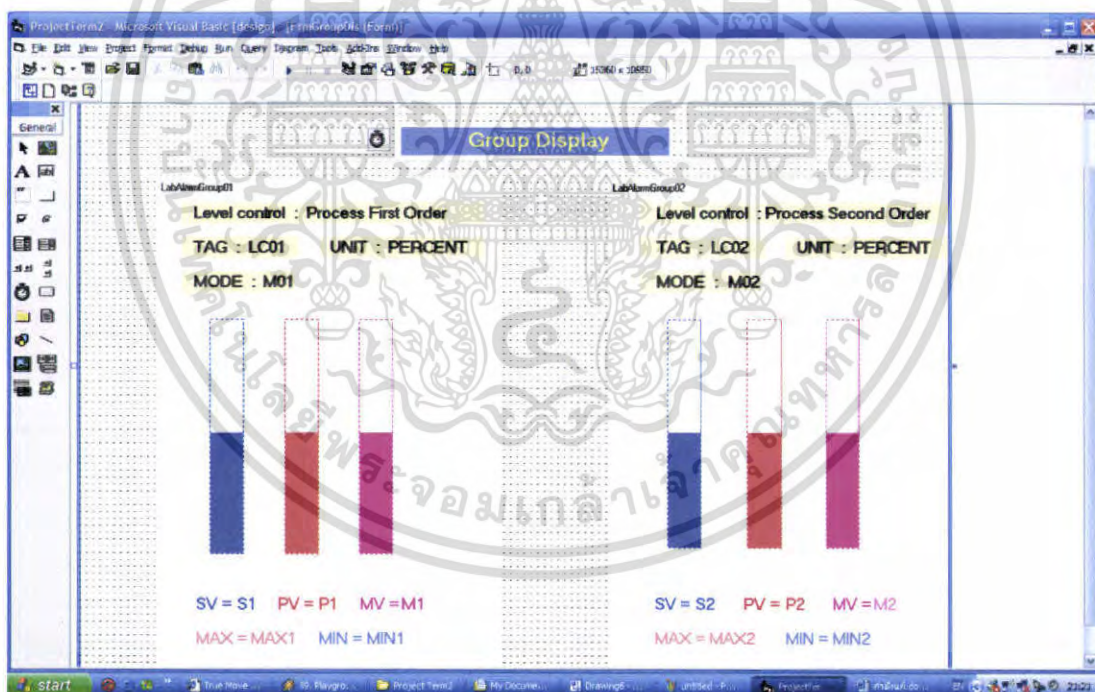
เมื่อต้องการเริ่มติดต่อกับหน่วยควบคุมกระบวนการกระทำได้โดยการเข้าที่ File > Start

เมื่อต้องการยกเลิกการติดต่อกับหน่วยควบคุมกระบวนการกระทำได้โดยการเข้าที่File > Stop

เมื่อต้องการปิดโปรแกรมเข้าที่ File > Exit

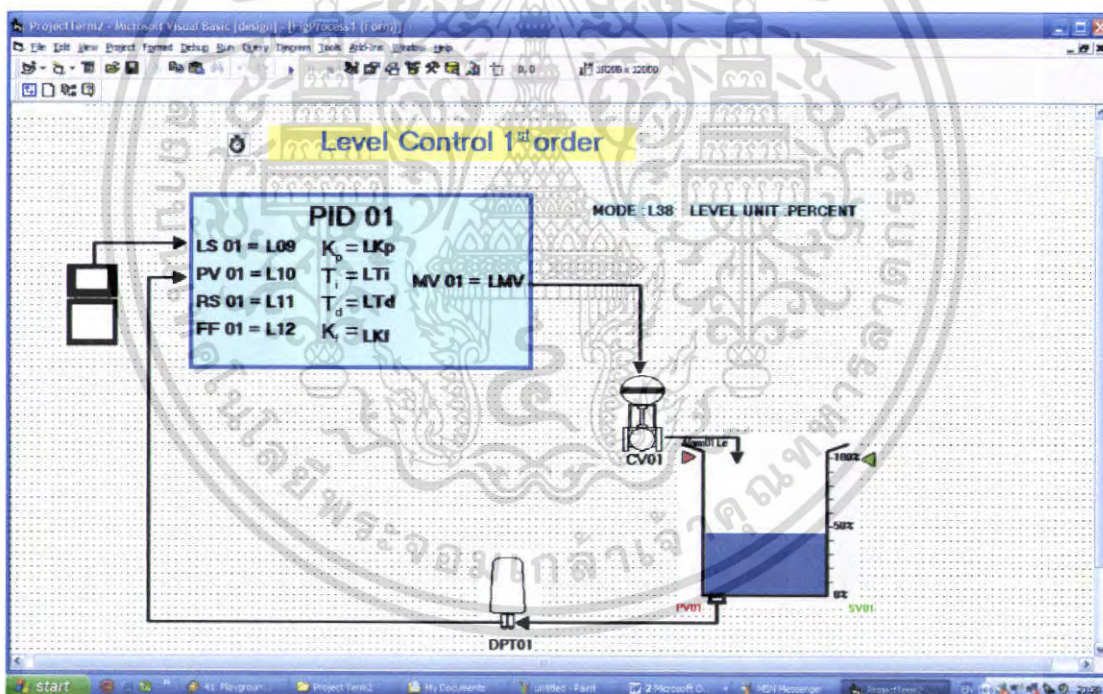
หมายเหตุ ถ้าต้องการเปลี่ยนแปลงค่าภายใน Setting ต้องทำการยกเลิกการติดต่อกับหน่วยควบคุมกระบวนการก่อน

ในส่วนของ โปรแกรมมีความสามารถอีกอย่างหนึ่งคือสามารถเลือกดูภาพของกลุ่มกระบวนการ แบบจำลองกระบวนการ และกราฟที่แสดงผลตอบสนองต่างๆ ของกระบวนการ โดยสามารถเลือกดูได้ที่ View > Form ต่างๆ ที่อยู่ภายใน โดยมีลักษณะต่างๆดังนี้



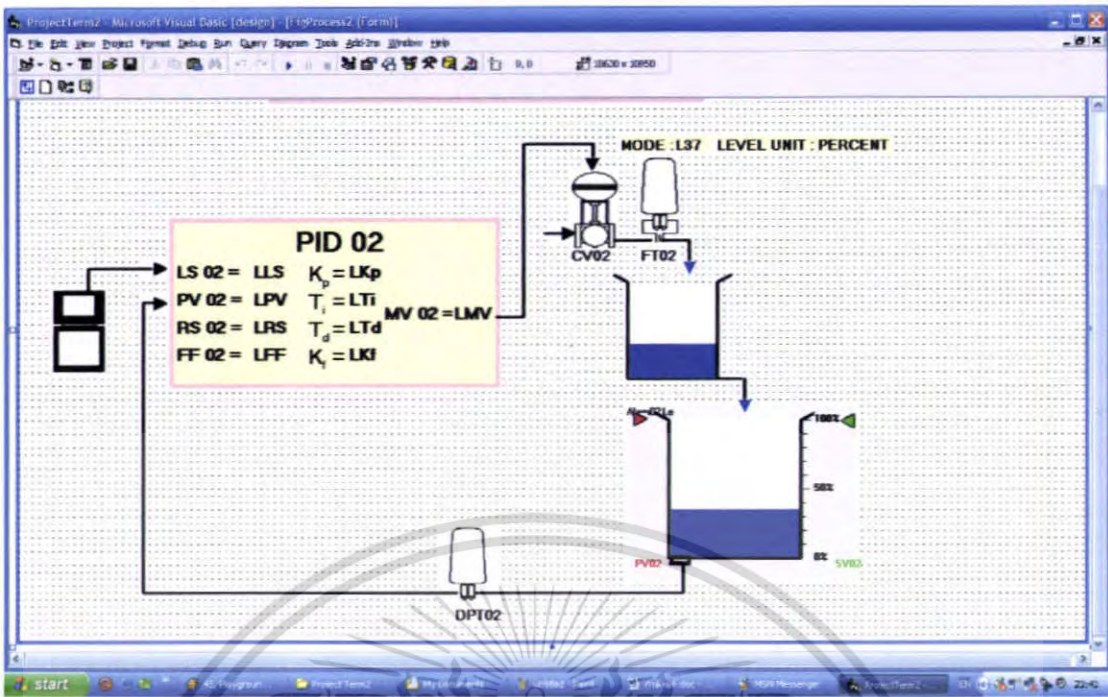
รูปที่ 3.5 แสดงหน้าต่างกลุ่มกระบวนการของโปรแกรมติดต่อของพนักงานผู้ใช้

ภายในส่วนของหน้าต่างกลุ่มกระบวนการแบ่งออกเป็น 2 ส่วนคือ กระบวนการควบคุมอันดับหนึ่ง และกระบวนการควบคุมอันดับสอง ในส่วนของรายละเอียดภายในหน้าต่างนี้บอกถึงหมายเลขของหน่วยควบคุมกระบวนการ (TAG) จุดประสงค์ของการควบคุม (DISP) โหมดของการควบคุม(MODE) ถ้าเป็น AUT เป็นการควบคุมแบบอัตโนมัติ หรือ MAN เป็นการควบคุมแบบแมน และค่าตัวแปรที่แสดงมีหน่วยเป็นแบบไหน (UNIT) ซึ่งภายในหน้าต่างนี้ช่วยให้ผู้ปฏิบัติงานสามารถดูการควบคุมหลายๆ กระบวนได้เพียงหน้าต่างเดียวซึ่งเพิ่มความปลอดภัยในการควบคุมมากขึ้นถ้าหากมีกระบวนการไหนที่เกิดการผิดพลาดขึ้นก็สามารถที่ทราบได้ในเวลาที่รวดเร็ว อีกทั้งภายในหน้าต่างนี้มีการแสดงความผิดพลาดขึ้นถ้าเกินกำหนดค่าที่ได้ตั้งเอา โดยแสดงเป็นลักษณะของสัญญาณเสียงและไฟกระพริบสีแดงจะแสดงแยกส่วนกันของแต่ละกระบวนการ และยังเพิ่มความรวดเร็วในการแก้ไขถ้าเกิดสัญญาณเตือนขึ้นเพียงแค่ผู้ปฏิบัติงานกดเมาส์บริเวณจุดที่มีการเตือน ตัวโปรแกรมจะเปลี่ยนจากหน้าต่างกลุ่มกระบวนการเข้าไปที่หน้าต่างของกระบวนการต่างๆ ที่สามารถแก้ไขต้นเหตุของการเกิดสัญญาณเตือนได้

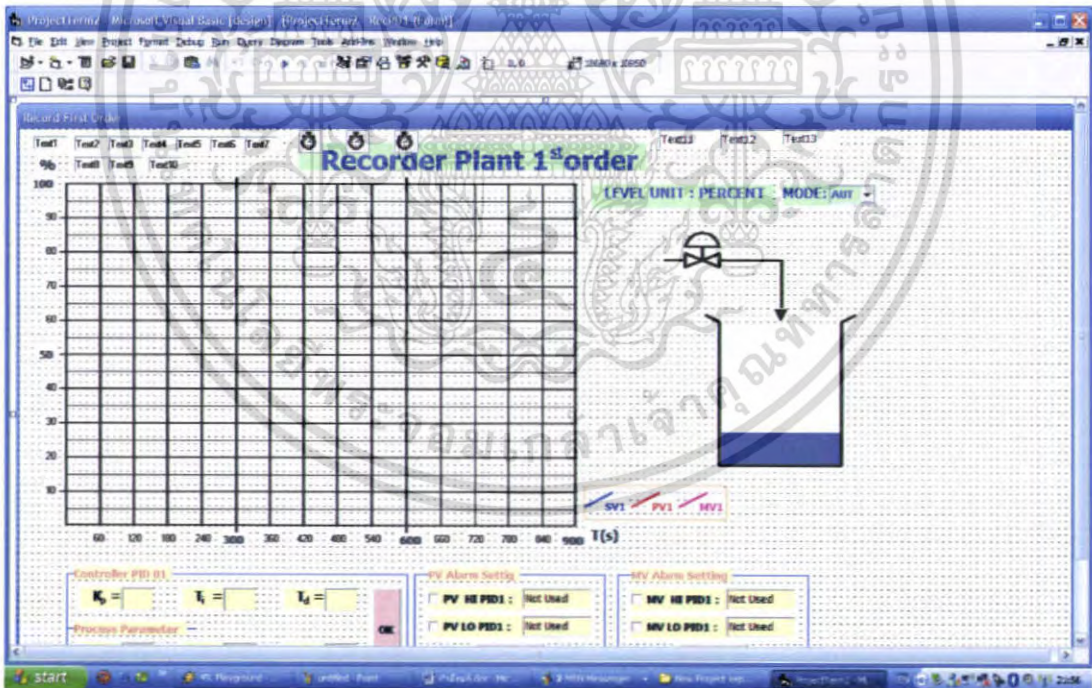


รูปที่ 3.6 แสดงหน้าต่างแบบจำลองอันดับหนึ่งของโปรแกรมติดต่อของพนักงานผู้ใช้

ภายในหน้าต่างแบบจำลองอันดับหนึ่งและอันดับสองเป็นการแสดงแบบจำลองของกระบวนการที่ถูกควบคุมแสดงเป็นกราฟิกของระดับน้ำขึ้นน้ำลงได้ แสดงค่าพารามิเตอร์ต่างๆ ที่ถูกตั้งค่าไว้อย่างละเอียด และยังมีสัญญาณเตือนและไฟกระพริบที่ตัวอักษรที่เกิดเหตุการณ์ผิดปกติ ซึ่งจะละเอียดมากกว่าหน้าต่างกลุ่มกระบวนการ

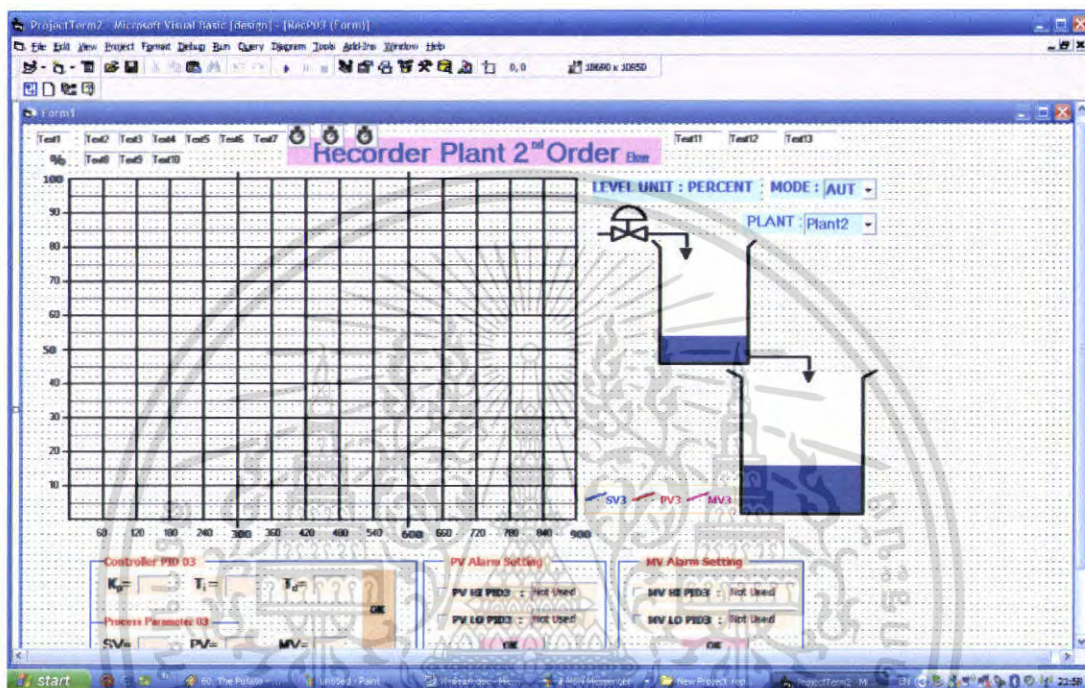


รูปที่ 3.7 แสดงหน้าต่างแบบจำลองอันดับสองของโปรแกรมติดต่อของพนักงานผู้ใช้



รูปที่ 3.8 แสดงหน้าต่างบันทึกสัญญาณระบบอันดับหนึ่งของโปรแกรมติดต่อของพนักงานผู้ใช้

ภายในหน้าต่างบันทึกสัญญาณระบบอันดับหนึ่งและอันดับสองเป็นการแสดงรูปที่ 3.8 และ 3.9 ตามลำดับผลตอบสนองของระบบซึ่งประกอบด้วย ค่าตัวแปรกระบวนการ (Process Variable, PV), ตัวแปรจัดการ (Manipulated Variable, MV), ค่าปรับตั้ง (Set point, SV) ค่าพารามิเตอร์, แสดงค่า PV และ MV, ตั้งค่า PV MV ALARM และเลือกโหมดการควบคุมแบบ AUT หรือ MAN ถ้าเป็นการปรับ MAN ผู้ปฏิบัติงานสามารถใส่ค่าตัวแปรจัดการได้เอง

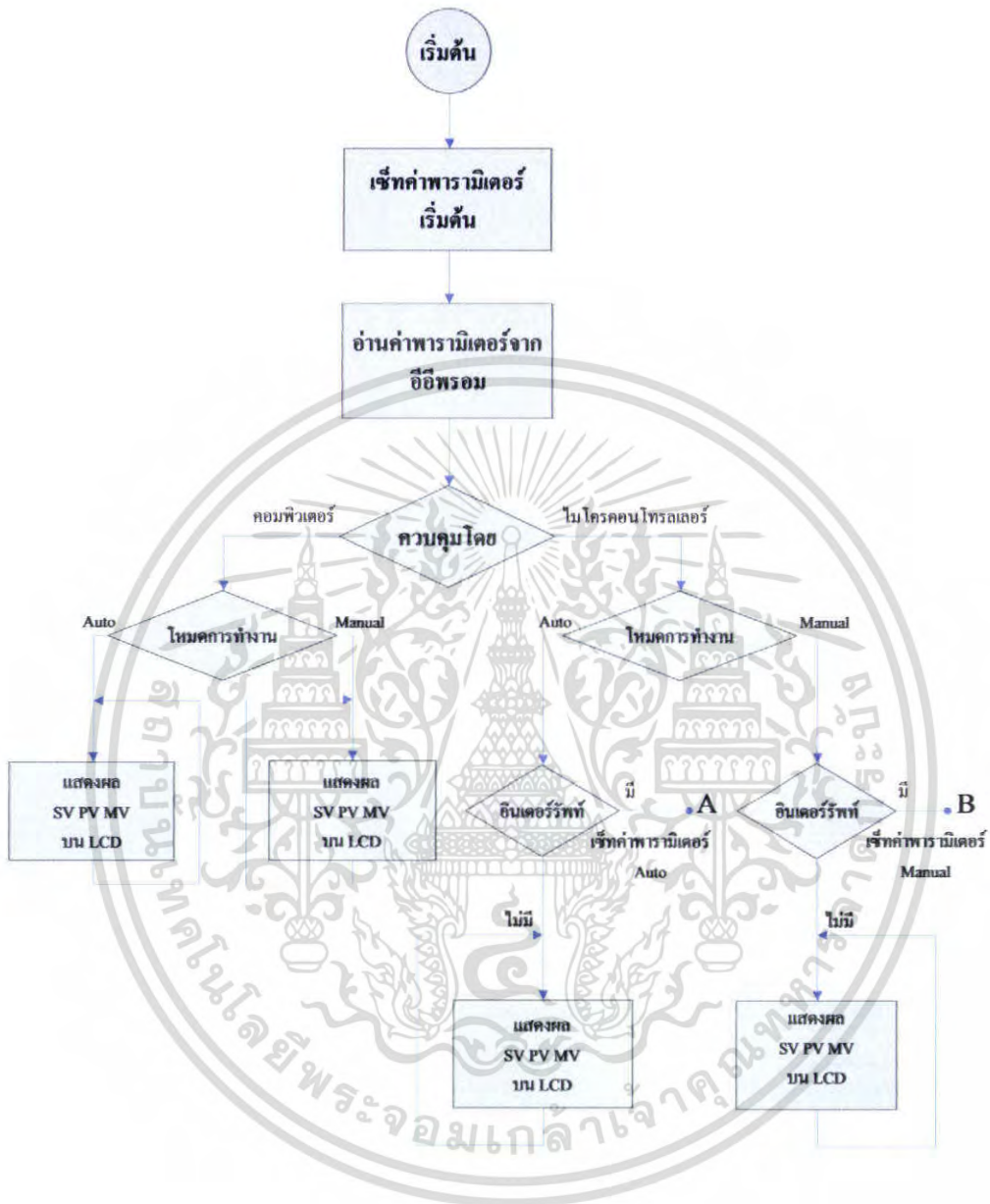


รูปที่ 3.9 แสดงหน้าต่างบันทึกสัญญาณระบบอันดับสองของ โปรแกรมติดต่อของพนักงานผู้ใช้

### 3.4 แผนภูมิแสดงขบวนการปฏิบัติงาน (Flow Chart)

ในหัวข้อนี้อธิบายถึง ขบวนการปฏิบัติงานของคอมพิวเตอร์หน่วยติดต่อและปฏิบัติการของพนักงานซึ่งใช้โปรแกรมวิซวลเบสิก รวมทั้งหน่วยควบคุมกระบวนการทั้ง 2 กระบวนซึ่งใช้ภาษาซี สำหรับไมโครคอนโทรลเลอร์ โดยมีหลักการทำงานของโปรแกรมหาดังนี้

## โฟลวชาร์ตการทำงานของไมโครคอนโทรลเลอร์ dsPIC30F4011 ของกระบวนการอันดับหนึ่ง

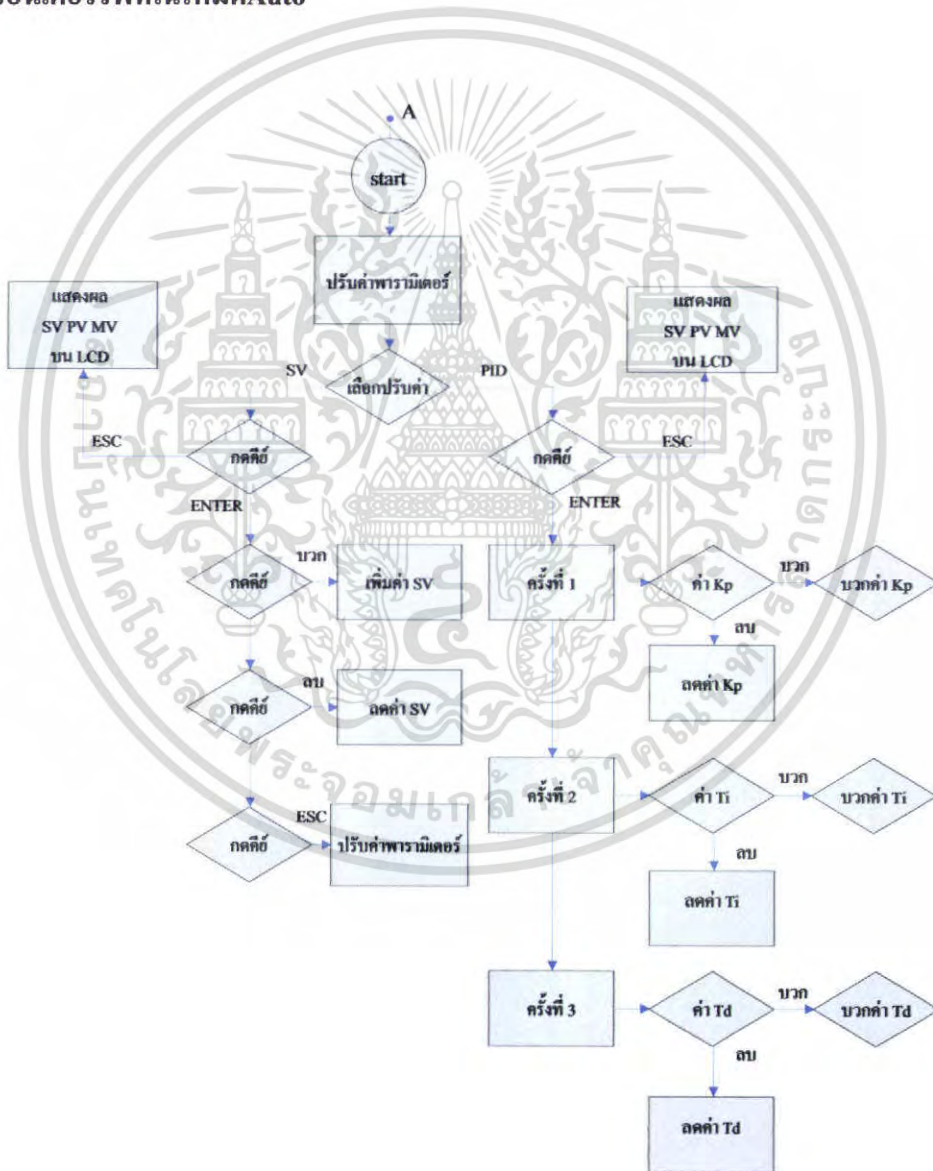


รูปที่ 3.10 โฟลวชาร์ตแสดงการทำงานของกระบวนการอันดับหนึ่ง

ตัวไมโครคอนโทรลเลอร์เริ่มการทำงานด้วยการกดปุ่ม “Start” จากนั้นจะเริ่มเซตค่าพารามิเตอร์เริ่มต้นและถูกส่งมาให้ อีอีพรอม เพื่ออ่านค่าพารามิเตอร์และจะส่งค่ามาว่าจะเลือกใช้อะไรเป็นตัวควบคุมระหว่าง “คอมพิวเตอรื” หรือ “ไมโครคอนโทรลเลอร์” ถ้าเลือกโหมดการทำงานเป็น คอมพิวเตอรื ควบคุม จะมี 2 โหมดให้เลือกการทำงาน คือ มีโหมดอัตโนมัติกับโหมดทำงานด้วยมือ การทำงานของ 2 โหมดนี้จะทำงานคล้ายคลึงกัน คือ จะรับค่าพารามิเตอร์มาจะ

โชว์ค่า SV,PV,MVบนหน้าจอแอลซีดี (LCD)แต่ถ้าเลือกโหมดการทำงานเป็น ไมโครคอนโทรลเลอร์ควบคุม จะมี 2 โหมดให้เลือกการทำงานคือโหมดอัตโนมัติและโหมดทำงานด้วยมือ ใน 2 โหมดนี้จะทำงานคล้ายคลึงกันคือ หลังจากเลือกโหมดการทำงานแล้วจะเกิดอินเทอร์รัพท์ขึ้น ผู้ใช้เองต้องไปปรับเซ็ทค่าพารามิเตอร์ซึ่งจะได้กล่าวต่อไปในโพลิวาร์ตการทำงานของกระบวนการอันดับหนึ่งเมื่อเกิดการอินเทอร์รัพท์ในโหมดอัตโนมัติและโหมดควบคุมด้วยมือ แต่ถ้าไม่เกิดอินเทอร์รัพท์เกิดขึ้น ค่า SV ,PV, MV, จะแสดงค่าต่างๆบนจอแอลซีดี

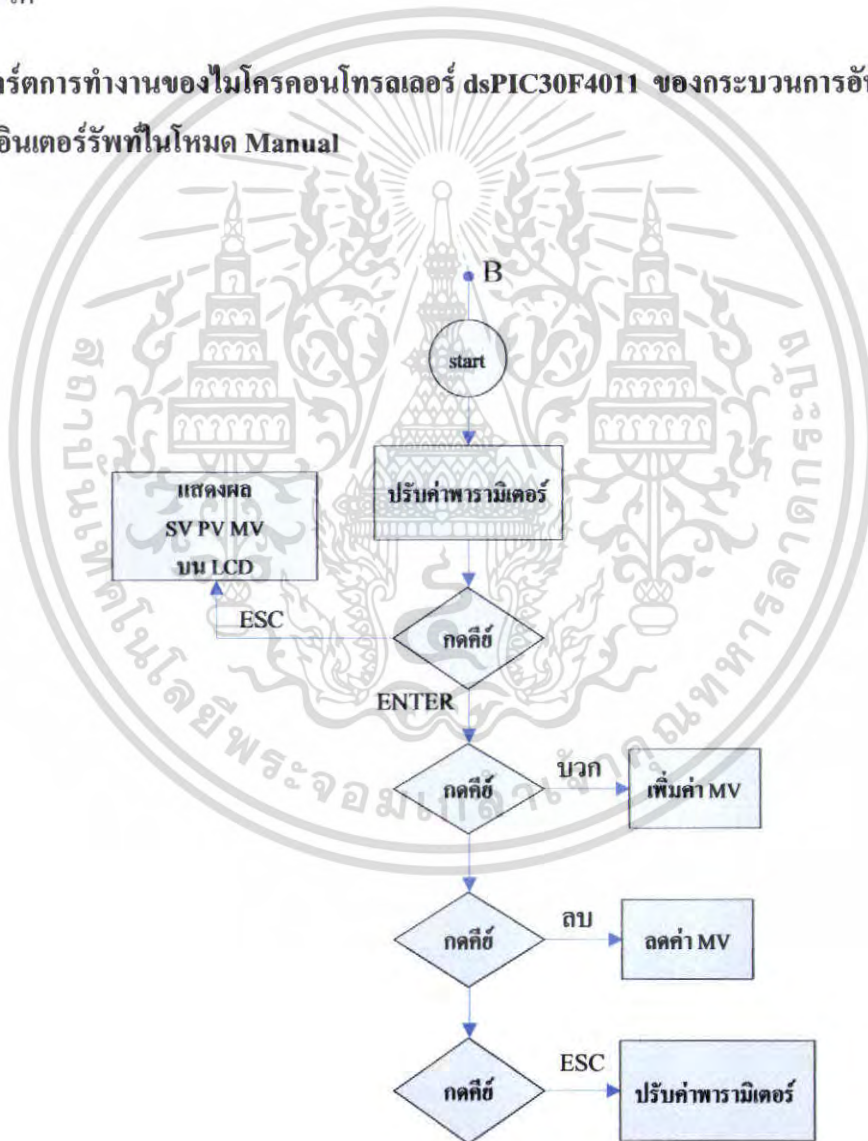
โพลิวาร์ตการทำงานของไมโครคอนโทรลเลอร์ dsPIC30F4011 ของกระบวนการอันดับหนึ่งเมื่อเกิดการอินเทอร์รัพท์ในโหมดAuto



รูปที่ 3.11 โพลิวาร์ตกระบวนการอันดับหนึ่งเมื่อเกิดการอินเทอร์รัพท์ในโหมดAuto

การทำงานต่อจากโฟลวชาร์ตที่ 3.11 เมื่อเกิดอินเทอร์รัพท์ในโหมดอัตโนมัติการทำงาน เริ่มต้นด้วยการปรับค่าพารามิเตอร์ โดยมีฟังก์ชันให้เลือกว่าเราจะปรับฟังก์ชันระหว่าง SV หรือ PID ถ้าเลือกปรับค่าของ SV กดคีย์หนึ่งครั้ง ค่าพารามิเตอร์ต่างๆ จะแสดงบนหน้าจอแอลซีดี กด Enter แล้วเราจะเข้าไปปรับค่าของ SV ให้เพิ่มขึ้น กดอีกครั้ง คราวนี้จะเข้าไปปรับค่าของ SV ให้ลดลง กดอีกครั้งจะเข้าไปปรับค่าพารามิเตอร์ แต่ถ้าเราเลือกปรับฟังก์ชัน PID กดคีย์ครั้งแรก ค่าพารามิเตอร์ต่างๆ จะแสดงบนหน้าจอแอลซีดี กดคีย์อีกครั้งจะเป็นการเข้าไปเซ็ทค่าของ Kp โดยเราสามารถเพิ่มหรือลดค่าของ Kp ได้ กดคีย์อีกครั้งจะเป็นการเซ็ทค่าของ Ti โดยที่เราสามารถเพิ่มหรือลดค่าของ Ti ได้ กดคีย์อีกครั้งจะเป็นการเข้าไปเซ็ทของ Td โดยเราสามารถเพิ่มหรือลดค่าของ Td ได้

โฟลวชาร์ตการทำงานของไมโครคอนโทรลเลอร์ dsPIC30F4011 ของกระบวนการอันดับหนึ่งเมื่อเกิดการอินเทอร์รัพท์ในโหมด Manual

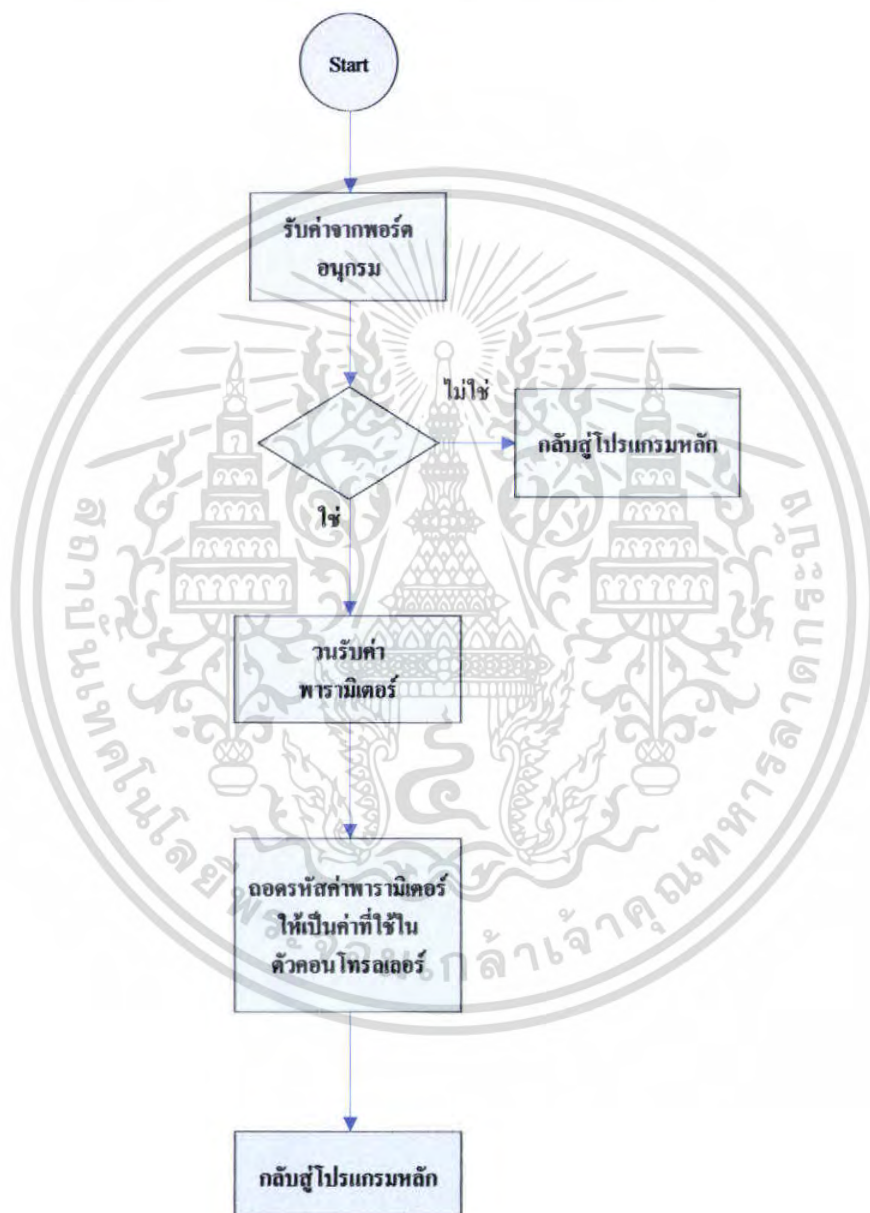


รูปที่ 3.12 โฟลวชาร์ตกระบวนการอันดับหนึ่งเมื่อเกิดการอินเทอร์รัพท์ในโหมด Manual



Remote ได้ และในส่วนของไมโครคอมพิวเตอร์จะสามารถเลือกโหมดการทำงานเป็น Remote ได้ ส่วนการอินเตอร์รัพท์ที่เกิดขึ้นเหมือนกับโฟลวชาร์ตกระบวนการอันดับหนึ่งเมื่อเกิดการอินเตอร์รัพท์ในโหมดอัตโนมัติ และโหมดควบคุมด้วยมือ คือจะมีการเซ็ทค่าต่างๆ

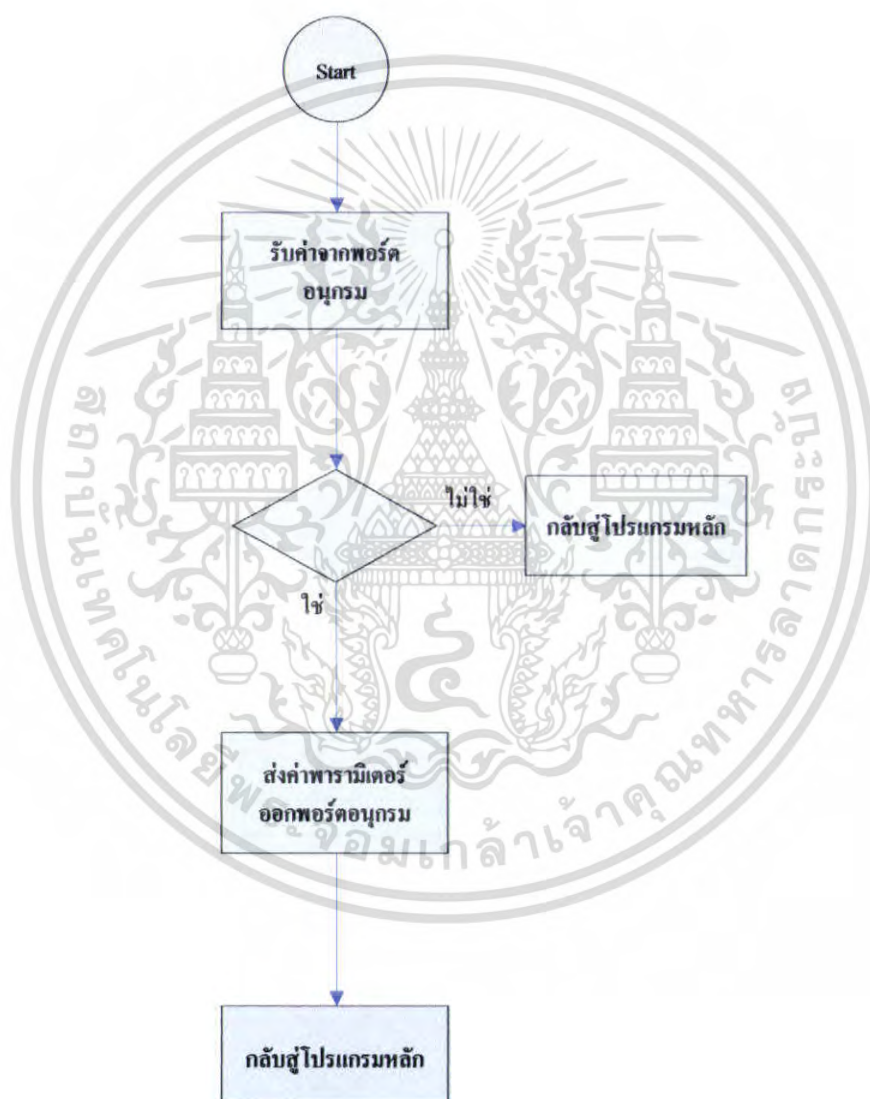
**โฟลวชาร์ตแสดงการอินเตอร์รัพท์ในการรับค่าของกระบวนการอันดับหนึ่ง**



รูปที่ 3.14 โฟลวชาร์ตแสดงการอินเตอร์รัพท์ในการรับค่าของกระบวนการอันดับหนึ่ง

การทำงานจะเริ่มขึ้นเมื่อรับค่ามาจากพอร์ตอนุกรม โปรแกรมจะทำการตัดสินใจว่าข้อมูลที่รับมาจากพอร์ตอนุกรมนั้นใช่หรือไม่ใช่ ถ้าใช่จะวนรับค่าพารามิเตอร์ของพีไอดี แล้วจะถูกส่งมาถอดรหัสค่าพารามิเตอร์ที่ใช้ในตัวคอนโทรลเลอร์ ถอดเสร็จแล้วจะกลับเข้าสู่โปรแกรมหลักอีกครั้งเพื่อทำงานต่อไป แต่ถ้าค่าที่รับมาจากพอร์ตอนุกรมไม่ใช่ค่าพารามิเตอร์ โปรแกรมจะทำการตัดสินใจกลับเข้าสู่โปรแกรมหลัก

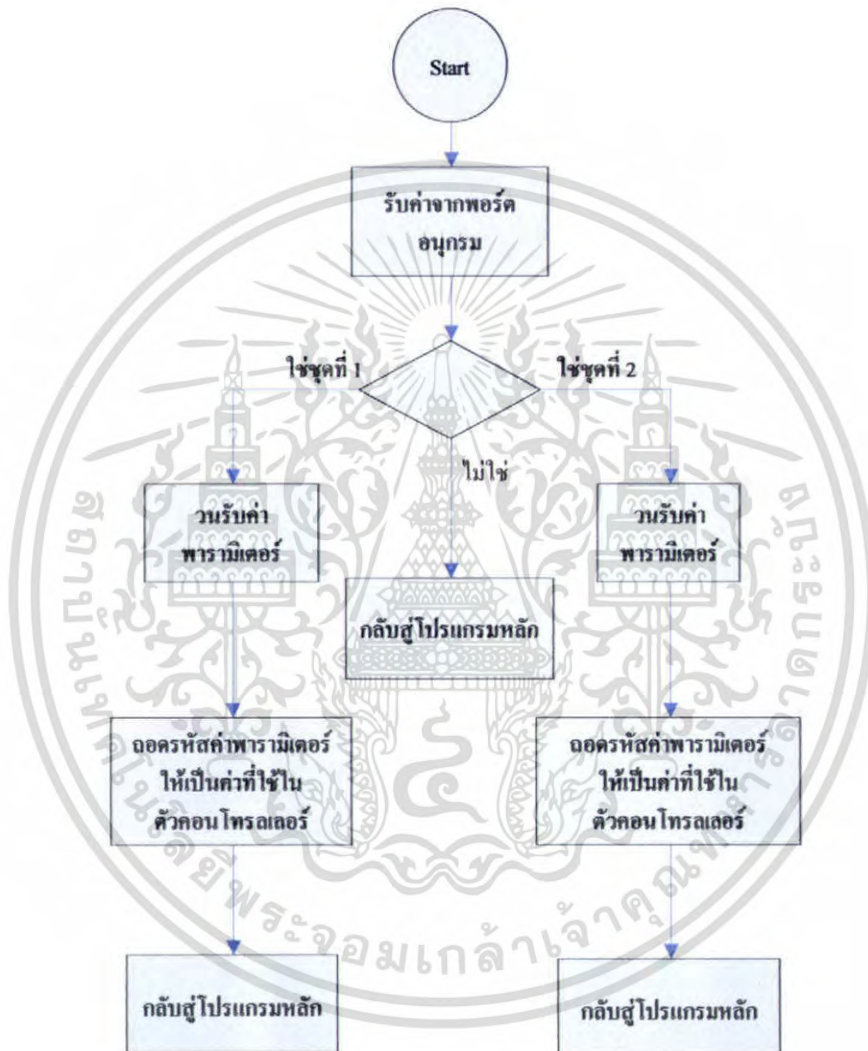
โฟลวชาร์ตแสดงการอินเทอร์รัพท์ในการส่งค่าของกระบวนการอันดับหนึ่ง



รูปที่ 3.15 โฟลวชาร์ตแสดงการอินเทอร์รัพท์ในการส่งค่าของกระบวนการอันดับหนึ่ง

การทำงานจะเริ่มขึ้นเมื่อรับค่ามาจากพอร์ตอนุกรม โปรแกรมจะตัดสินใจว่า ค่าที่รับมาใช่หรือไม่ใช่ค่าพารามิเตอร์ ถ้าใช่จะกระทำการส่งค่าพารามิเตอร์ออกจากพอร์ตอนุกรมและกลับสู่โปรแกรมหลัก แต่ถ้าค่าที่รับมาไม่ใช่ค่าพารามิเตอร์ ก็จะกลับสู่โปรแกรมหลักเหมือนกัน

### โฟลวชาร์ตแสดงการอินเทอร์รัพท์ในการรับค่าของกระบวนการอันดับสอง

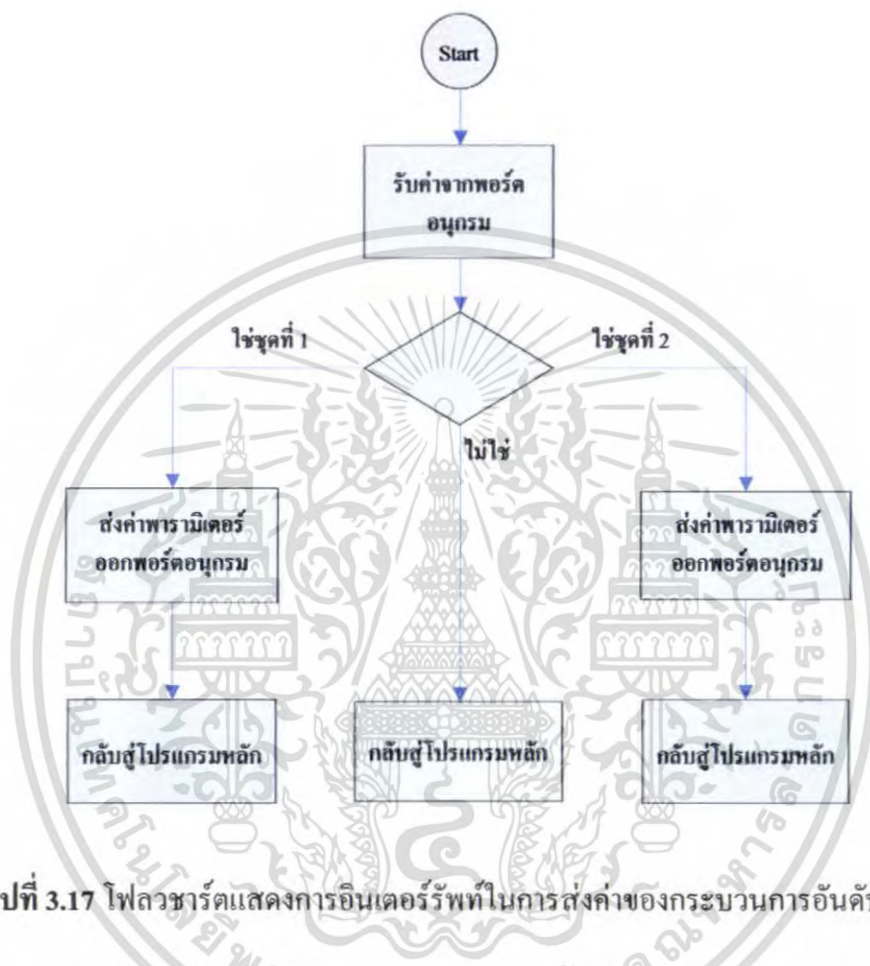


รูปที่ 3.16 โฟลวชาร์ตแสดงการอินเทอร์รัพท์ในการรับค่าของกระบวนการอันดับสอง

เนื่องจากการอินเทอร์รัพท์ในการรับและส่งค่าของกระบวนการอันดับสองนั้น จะมีค่าพีไอคืออยู่สองชุด การทำงานจะเริ่มขึ้นเมื่อรับค่ามาจากพอร์ตอนุกรม โปรแกรมจะทำการตัดสินใจว่า ข้อมูลที่รับมาจากพอร์ตอนุกรมนั้นใช่หรือไม่ใช่ ถ้าใช่ชุดที่ 1 กับชุดที่ 2 จะวนรับค่าพารามิเตอร์ของพีไอดี แล้วจะถูกส่งมาถอดรหัสค่าพารามิเตอร์ที่ใช้ในตัวคอนโทรลเลอร์ ถอดเสร็จแล้วจะกลับเข้าสู่

โปรแกรมหลักอีกครั้งเพื่อทำงานต่อไป แต่ถ้าค่าที่รับมาจากพอร์ตอนุกรมไม่ใช่ค่าพารามิเตอร์ โปรแกรมจะทำการตัดสินใจกลับเข้าสู่โปรแกรมหลัก

โฟลวชาร์ตแสดงการอินเทอร์รัพท์ในการส่งค่าของกระบวนการอันดับสอง



รูปที่ 3.17 โฟลวชาร์ตแสดงการอินเทอร์รัพท์ในการรับและส่งค่าของกระบวนการอันดับสอง

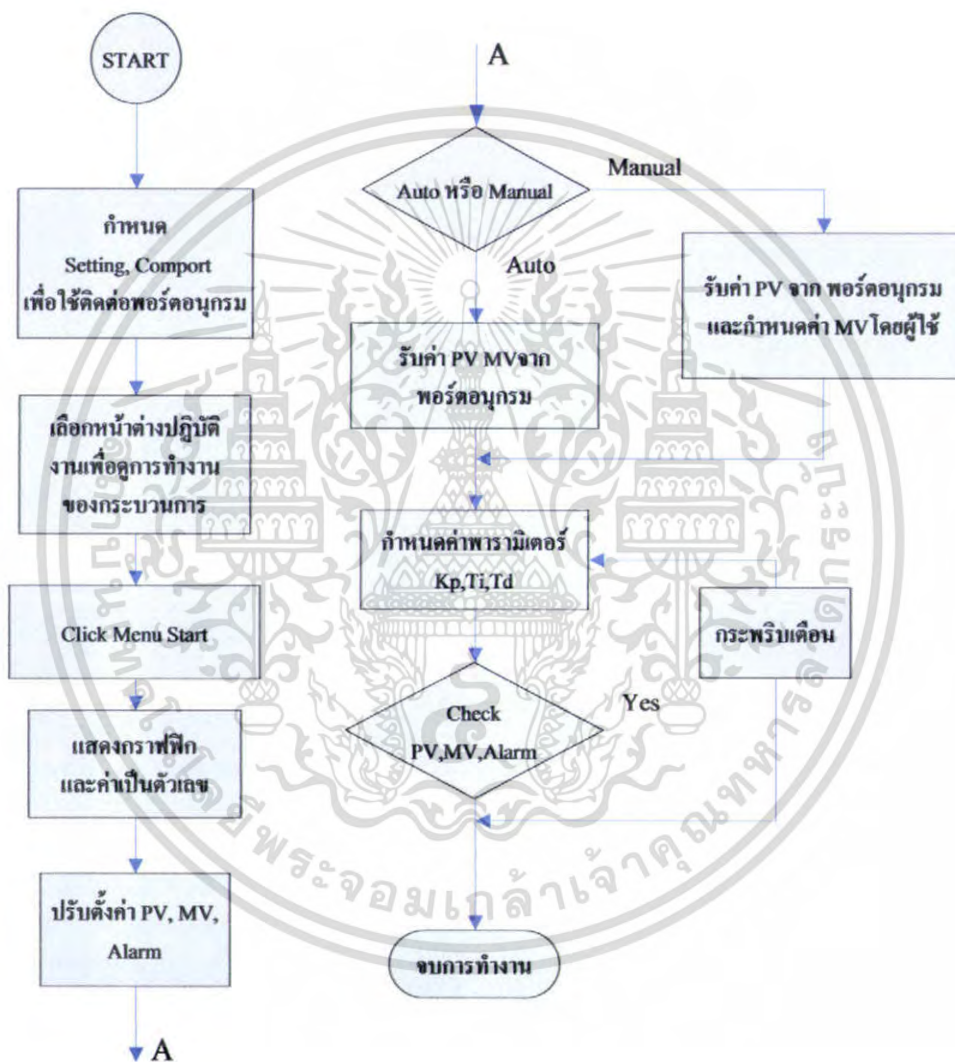
เนื่องจากการอินเทอร์รัพท์ในการรับและส่งค่าของกระบวนการอันดับสองนั้น จะมีค่าพีไอคืออยู่สองชุด การทำงานจะเริ่มขึ้นเมื่อรับค่ามาจากพอร์ตอนุกรม โปรแกรมจะทำการตัดสินใจว่าค่าที่รับมาใช่หรือไม่ใช่ค่าพารามิเตอร์ ถ้าใช่ทั้งจุดที่ 1 และจุดที่ 2 จะกระทำการส่งค่าพารามิเตอร์ออกทางพอร์ตอนุกรมและกลับสู่โปรแกรมหลัก แต่ถ้าค่าที่รับมาไม่ใช่ค่าพารามิเตอร์ ก็จะกลับสู่โปรแกรมหลัก

โฟลวชาร์ตโปรแกรมแสดงผลหลักของหน่วยติดต่อผู้ใช้งาน

ซึ่งมีโฟลวชาร์ตการทำงานหลัก 3 ส่วนดังนี้

- โฟลวชาร์ตแสดงการทำงานของโปรแกรมหลัก (Visual Basic main window)
- โฟลวชาร์ตแสดงผลรวมของกระบวนการอันดับหนึ่งและอันดับสอง (Group Display)
- โฟลวชาร์ตแสดงการทำงานของกระบวนการอันดับหนึ่งและอันดับสองในส่วยของการทำงานของกระบวนการและการบันทึกผล

### โฟลวชาร์ตแสดงการทำงานของโปรแกรมหลัก

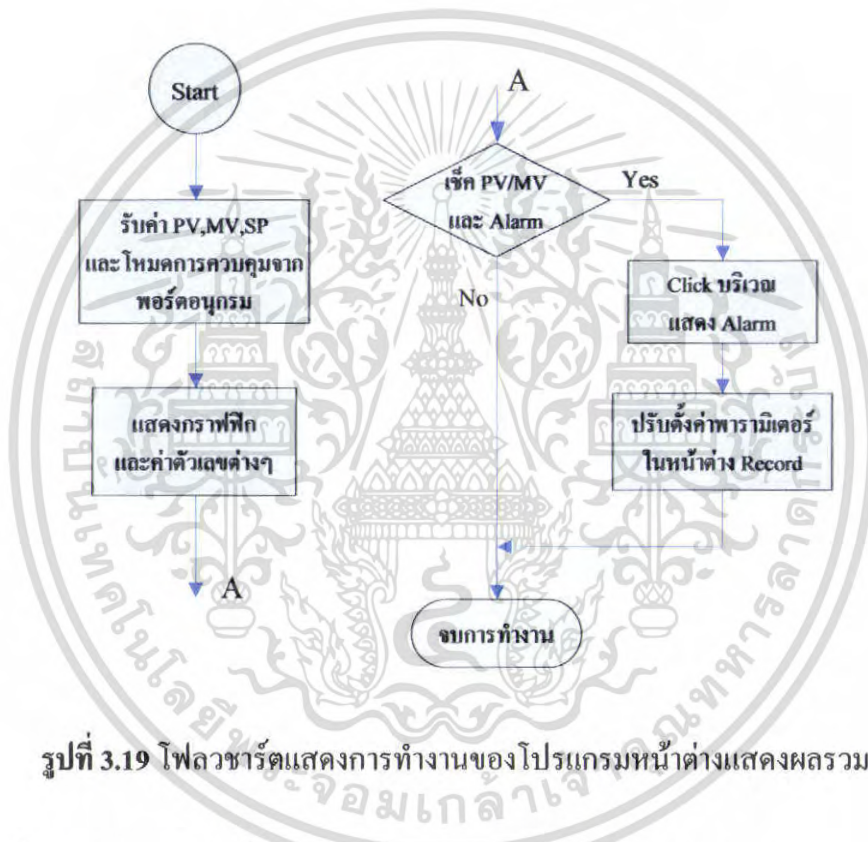


รูปที่ 3.18 แสดงโฟลวชาร์ตการทำงานของโปรแกรมแสดงผลหน้าต่างหลัก

เริ่มต้นการทำงานของโปรแกรมด้วยการกำหนดค่าSettings ให้กับการเชื่อมโยงด้วย พอร์ตอนุกรม แล้วคลิกไปที่ “Menu Strat” หน้าหน้าต่างปฏิบัติงานจะแสดงผลกราฟฟีกและตัวเลขแสดงค่าต่างๆ จากนั้นทำการเลือกโหมดการทำงานระหว่างควบคุมแบบอัตโนมัติ (“AUT”) หรือแบบ

ควบคุมด้วยมือ (“MAN”) ถ้าเลือกการควบคุมแบบอัตโนมัติโปรแกรมจะรับค่า PV และ MV มาจาก พอร์ตอนุกรม ถ้าเลือกการควบคุมแบบควบคุมด้วยมือ โปรแกรมจะรับค่า PV จากพอร์ตอนุกรม และผู้ใช้กำหนดค่า MV ให้กับระบบ โปรแกรมจะทำการตรวจสอบการเตือนของระบบ (Alarm) โดยตรวจสอบค่า PV และ MV ของระบบ หากมีสิ่งผิดปกติ จะมีการกระพริบเตือนและเมื่อผู้ใช้คลิกบริเวณที่มีการกระพริบเตือนของหน้าต่างปฏิบัติงานแล้วหน้าต่างปฏิบัติงานจะเปลี่ยนเป็นหน้าต่างเก็บข้อมูลและปรับตั้งค่า (Recorder) ของกระบวนการที่มีการเตือนนั้น

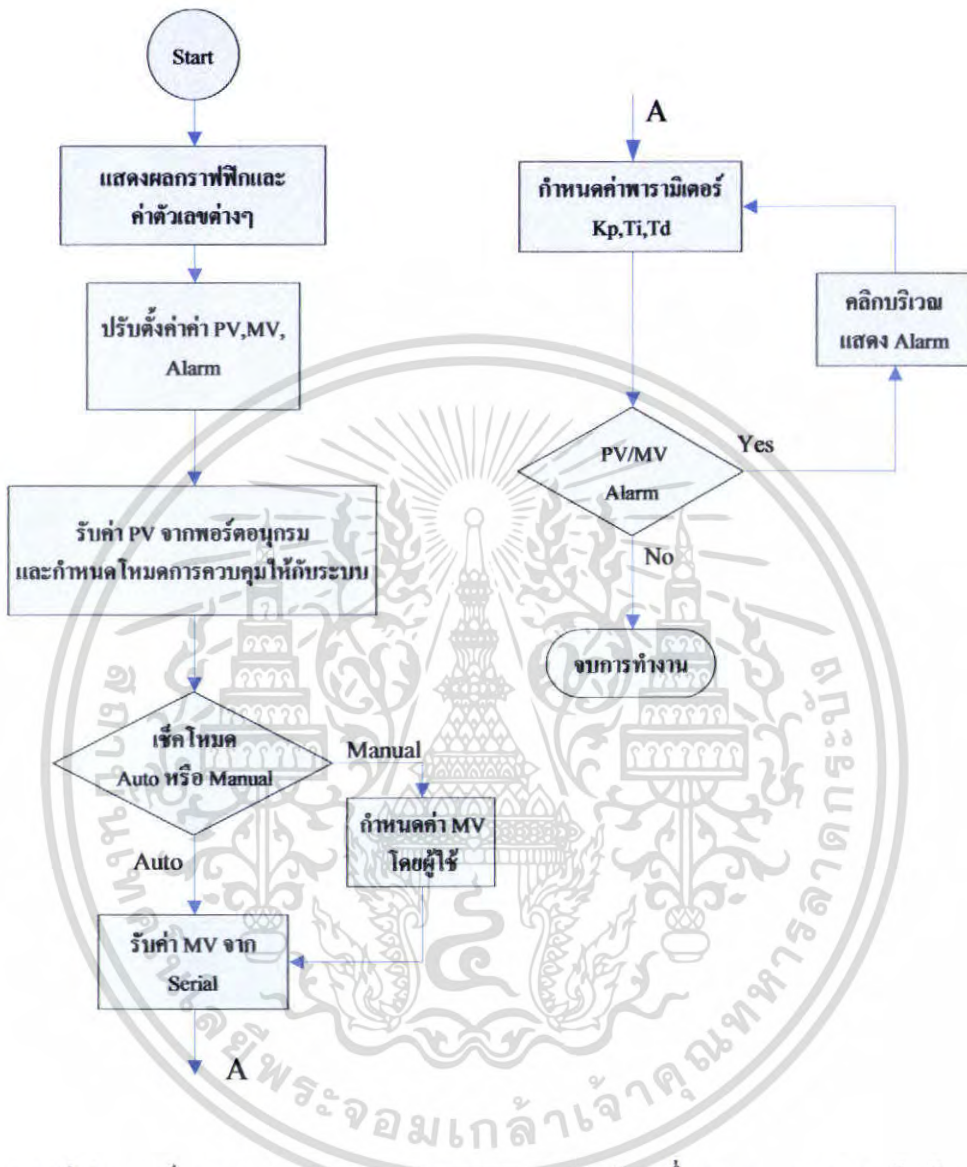
### โฟลวชาร์ตแสดงผลรวมของกระบวนการอันดับหนึ่งและอันดับสอง



รูปที่ 3.19 โฟลวชาร์ตแสดงการทำงานของโปรแกรมหน้าต่างแสดงผลรวม

เริ่มการทำงานโดยเริ่มแสดงกราฟฟิก(Graphic) ค่าตัวแปรต่างๆของระบบ ทั้งของ กระบวนการอันดับหนึ่ง และกระบวนการอันดับสอง โดยมีลักษณะเป็นกราฟแท่งสามสีคือ น้ำเงิน (SV) , แดง (PV) และ ชมพู (MV) จากนั้นโปรแกรมก็จะทำการตรวจสอบสัญญาณเตือน (Alarm) ของกระบวนการอันดับหนึ่งและกระบวนการอันดับสอง หากไม่มีสัญญาณเตือน โปรแกรมก็จะทำงานต่อไปหรือสามารถจบการทำงานของโปรแกรมได้

โฟลวชาร์ตแสดงการทำงานของกระบวนการอันดับหนึ่งและอันดับสองในส่วนของการแสดงการทำงาน  
การทำงานของกระบวนการและการบันทึกผล



รูปที่ 3.20 โฟลวชาร์ตแสดงการทำงานของกระบวนการอันดับหนึ่งและกระบวนการอันดับสองใน  
ส่วนของหน้าจอแสดงผล

แสดงการทำงานของกระบวนการอันดับหนึ่งและกระบวนการอันดับสอง เริ่มต้นการทำงาน  
ของโปรแกรม เมื่อเลือกหน้าต่างปฏิบัติงาน จากนั้นโดยเริ่มต้นแสดงกราฟิก PV และ SV ใน  
รูปของถังวัดระดับ และลูกศรชี้ตำแหน่งบนถังวัดระดับ แสดงค่าการเปลี่ยนแปลงค่าต่างๆของ  
กระบวนการค่าพารามิเตอร์พีไอดีของกระบวนการ ได้แก่ค่า SV, PV, MV, Kp, Ti และTd จากนั้น  
ทำการเลือกโหมดการทำงานระหว่างควบคุมแบบอัตโนมัติ (“AUT”) หรือแบบควบคุมด้วยมือ  
 (“MAN”) ถ้าเลือกการควบคุมแบบอัตโนมัติโปรแกรมจะรับค่า PVและ MV มาจาก พอร์ตอนุกรม

ถ้าเลือกการควบคุมแบบควบคุมด้วยมือ โปรแกรมจะรับค่า PV จากพอร์ตอนุกรมและผู้ใช้กำหนดค่า MV ให้กับระบบ โปรแกรมจะทำการตรวจสอบการเตือนของระบบ (Alarm) โดยตรวจสอบค่า PV และ MV ของระบบ หากมีสิ่งผิดปกติ จะมีการกระพริบเตือนและเมื่อผู้ใช้คลิกบริเวณที่มีการกระพริบเตือนของหน้าต่างปฏิบัติงานแล้วหน้าต่างปฏิบัติงานจะเปลี่ยนเป็นหน้าต่างเก็บข้อมูลและปรับตั้งค่า (Recorder) ของกระบวนการที่มีการเตือนนั้น หากไม่มีสัญญาณเตือนกลับไปทำงานในโปรแกรมทันที



## บทที่ 4

### การทดลองและผลการทดลอง

#### 4.1 กล่าวนำ

ในบทนี้จะเป็นการนำระบบคิซีเอส (DCS) ที่สร้างขึ้นมาทดลองเพื่อทดสอบว่าระบบคิซีเอสที่สร้างขึ้นสามารถทำงานได้อย่างที่ต้องการหรือไม่ โดยก่อนที่จะทำการทดลองระบบต้องมีการทดสอบความพร้อมของอุปกรณ์ก่อน เช่น เครื่องมือวัดระดับน้ำของกระบวนการ, อุปกรณ์แปลงจากอนาล็อกเป็นดิจิทัล เพื่อให้การสัญญาณที่ออกมาจากอุปกรณ์มีความถูกต้องไม่ทำให้การทำงานของระบบมีความผิดพลาด หลังจากนั้นจึงทำการทดลองหาค่าพารามิเตอร์โดยวิธีซีเกลอร์และนิโคลส์ และวิธีของคาร์ลลิน และผลตอบสนองของกระบวนการมาเปรียบเทียบกัน และทดลองปรับปรุงค่าพารามิเตอร์เพื่อให้ได้ผลตอบสนองของกระบวนการที่ดีและเหมาะสมที่สุด

#### 4.2 ผลการทดลองเครื่องมือวัดระดับน้ำของกระบวนการ

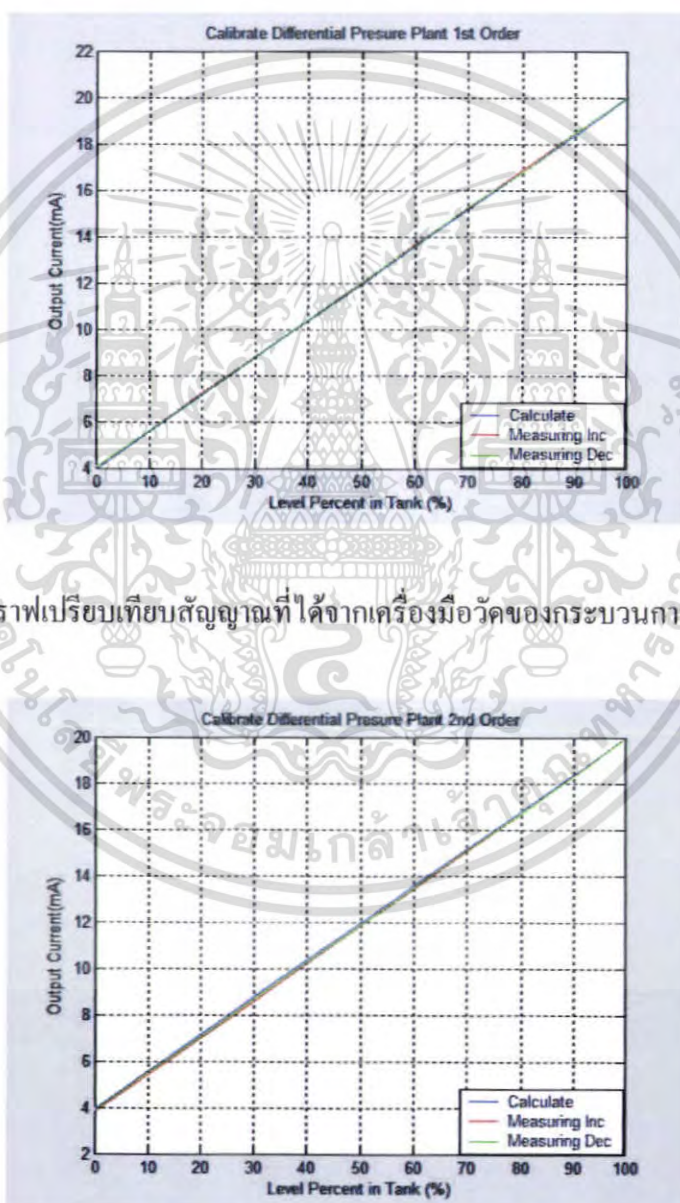
การทดลองนี้จะเป็นการทดสอบเครื่องมือวัดระดับน้ำว่าสามารถส่งสัญญาณออกมาได้ถูกต้องกับระดับน้ำภายในถังหรือไม่ โดยเครื่องมือวัดระดับน้ำจะส่งสัญญาณไฟฟ้าออกมาในย่าน 4-20 มิลลิแอมแปร์ทดลองโดยใส่น้ำลงในถังที่ตำแหน่ง 0% เครื่องมือวัดจะต้องส่งสัญญาณออกมา 4 มิลลิแอมแปร์ และเมื่อระดับน้ำในถังมากขึ้นเครื่องมือวัดก็จะส่งสัญญาณมากขึ้น และถ้าใส่น้ำในถังที่ตำแหน่ง 100 เปอร์เซ็นต์ เครื่องมือวัดจะต้องส่งสัญญาณออกมา 20 มิลลิแอมแปร์

#### ตารางที่ 4.1 ผลการทดลองเครื่องมือวัดระดับน้ำของกระบวนการ

ระดับน้ำในถัง (เปอร์เซ็นต์)	ค่าที่ได้จากการ คำนวณ (มิลลิแอมแปร์)	ค่าจากกระบวนการ อันดับหนึ่ง (มิลลิแอมแปร์)		ค่าจากกระบวนการ อันดับสอง (มิลลิแอมแปร์)	
		ขอบขาขึ้น	ขอบขาลง	ขอบขาขึ้น	ขอบขาลง
0	4	4.08	4.06	3.85	3.94
10	5.6	5.67	5.67	5.43	5.53
20	7.2	7.27	7.26	7.01	7.10
30	8.8	8.86	8.86	8.62	8.73
40	10.4	10.46	10.46	10.26	10.27
50	12	12.03	12.09	11.87	11.89

ตารางที่ 4.1(ต่อ) ผลการทดลองเครื่องมือวัดระดับน้ำของกระบวนการ

60	13.6	13.69	13.64	13.46	13.49
70	15.2	15.23	15.27	15.09	15.13
80	16.8	16.91	16.84	16.71	16.72
90	18.4	18.51	18.48	18.35	18.35
100	20	20.05	20.05	20.00	20.00



รูปที่ 4.1 กราฟเปรียบเทียบสัญญาณที่ได้จากเครื่องมือวัดของกระบวนการอันดับหนึ่ง

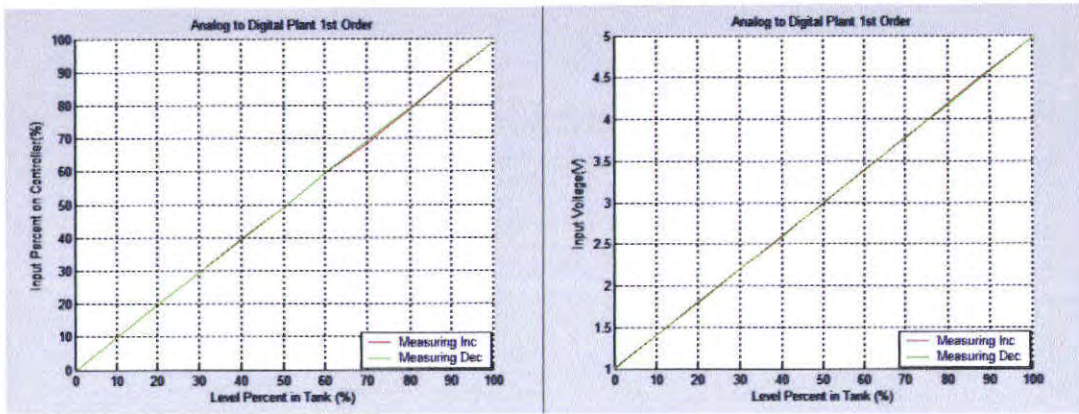
รูปที่ 4.2 กราฟเปรียบเทียบสัญญาณที่ได้จากเครื่องมือวัดของกระบวนการอันดับสอง

### 4.3 ผลการทดลองอุปกรณ์แปลงจากอนาล็อกเป็นดิจิตอล (ADC) ในกระบวนการอันดับหนึ่ง

การทดลองนี้จะเป็นการทดลองส่งสัญญาณอนาล็อกจากเครื่องมือวัดระดับน้ำที่แปลงเป็นแรงดันไฟฟ้า(1-5 โวลต์) โดยผ่านอุปกรณ์แปลงอนาล็อกเป็นดิจิตอล เพื่อส่งให้กับคอมพิวเตอร์

ตารางที่ 4.2 ผลการทดลองอุปกรณ์แปลงจากอนาล็อกเป็นดิจิตอลในกระบวนการอันดับหนึ่ง

ระดับน้ำในถัง (เปอร์เซ็นต์)	สัญญาณที่วัดได้ (โวลต์)		ค่าที่อ่านออกมาได้จากคอมพิวเตอร์ (เปอร์เซ็นต์)	
	ขอบขาขึ้น	ขอบขาลง	ขอบขาขึ้น	ขอบขาลง
0	1.021	1.011	0.0	0.0
10	1.418	1.413	10.0	10.0
20	1.812	1.807	20.1	19.9
30	2.207	2.205	29.7	29.7
40	2.605	2.601	39.8	39.7
50	2.993	3.004	49.3	49.3
60	3.402	3.388	59.8	59.7
70	3.783	3.789	68.7	69.5
80	4.198	4.178	79.2	79.4
90	4.591	4.581	89.4	89.5
100	4.967	4.967	99.0	99.0



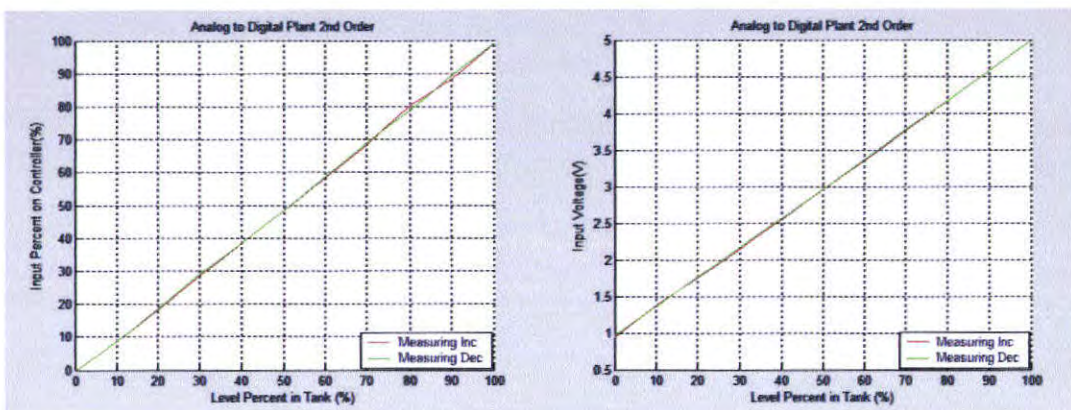
รูปที่ 4.3 กราฟที่ได้จากการทดลองแปลงอนาล็อกเป็นดิจิทัลในกระบวนการอันดับหนึ่ง

#### 4.4 ผลการทดลองอุปกรณ์แปลงจากอนาล็อกเป็นดิจิทัล (ADC) ในกระบวนการอันดับสอง

การทดลองนี้จะเป็นการทดลองส่งสัญญาณอนาล็อกจากเครื่องมือวัดระดับน้ำที่แปลงเป็นแรงดันไฟฟ้า(1-5 โวลต์) โดยผ่านอุปกรณ์แปลงอนาล็อกเป็นดิจิทัล เพื่อส่งให้กับคอมพิวเตอร์

ตารางที่ 4.3 ผลการทดลองอุปกรณ์แปลงจากอนาล็อกเป็นดิจิทัลในกระบวนการอันดับสอง

ระดับน้ำในถัง (เปอร์เซ็นต์)	สัญญาณที่วัดได้ (โวลต์)		ค่าที่อ่านออกมาได้จากคอมพิวเตอร์ (เปอร์เซ็นต์)	
	ขอบข้างขึ้น	ขอบข้างลง	ขอบข้างขึ้น	ขอบข้างลง
0	0.96	0.983	0.0	0.0
10	1.386	1.379	8.8	8.8
20	1.751	1.772	18.7	18.9
30	2.154	2.179	28.7	29.3
40	2.563	2.567	38.7	38.6
50	2.966	2.971	48.6	48.8
60	3.365	3.372	58.7	58.9
70	3.771	3.779	68.5	69.1
80	4.178	4.179	80.6	79.0
90	4.581	4.585	88.4	89.5
100	4.999	4.999	99.4	99.4

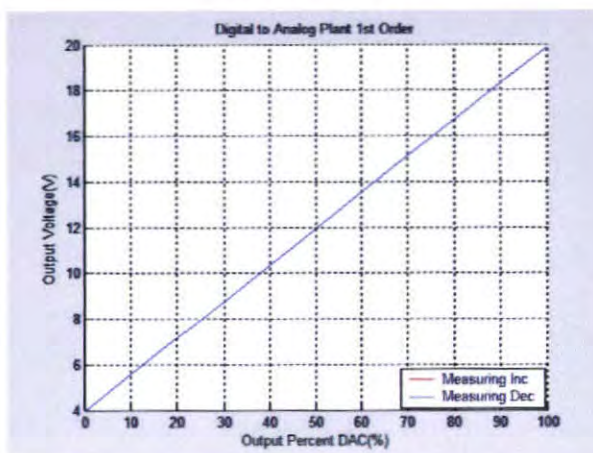


รูปที่ 4.4 กราฟที่ได้จากการทดลองแปลงอนาล็อกเป็นดิจิทัลในระบบการอันดับสอง

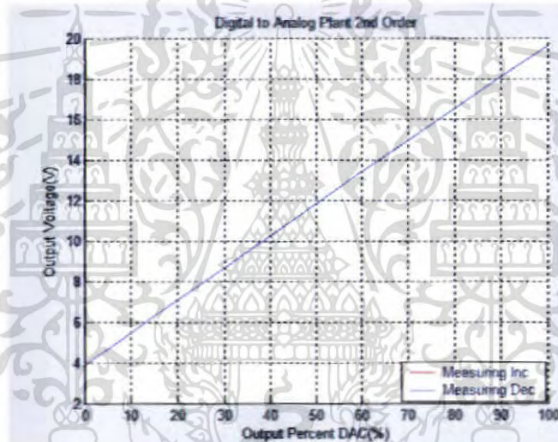
### 4.5 ผลการทดลองอุปกรณ์แปลงจากดิจิทัลเป็นอนาล็อก (DAC) ในระบบการ

ตารางที่ 4.4 ผลการทดลองอุปกรณ์แปลงจากดิจิทัลเป็นอนาล็อก (DAC) ในระบบการ

ค่าที่ต้องการวัด (เปอร์เซ็นต์)	ค่าที่ได้จากระบบการอันดับที่หนึ่ง (มิลลิแอมแปร์)		ค่าที่ได้จากระบบการอันดับที่สอง (มิลลิแอมแปร์)	
	ขอบขาขึ้น	ขอบขาลง	ขอบขาขึ้น	ขอบขาลง
0	4.00	4.00	3.93	3.93
10	5.6	5.6	5.52	5.52
20	7.19	7.19	7.11	7.11
30	8.79	8.79	8.70	8.70
40	10.39	10.39	10.27	10.27
50	11.99	11.99	11.85	11.85
60	13.58	13.58	13.43	13.43
70	15.18	15.18	15.00	15.00
80	16.76	16.76	16.57	16.57
90	18.33	18.33	18.14	18.14
100	19.88	19.88	19.68	19.68



รูปที่ 4.5 กราฟที่ได้จากการทดลองอุปกรณ์แปลงจากดิจิทัลเป็นอนาล็อก  
ในกระบวนการอันดับหนึ่ง



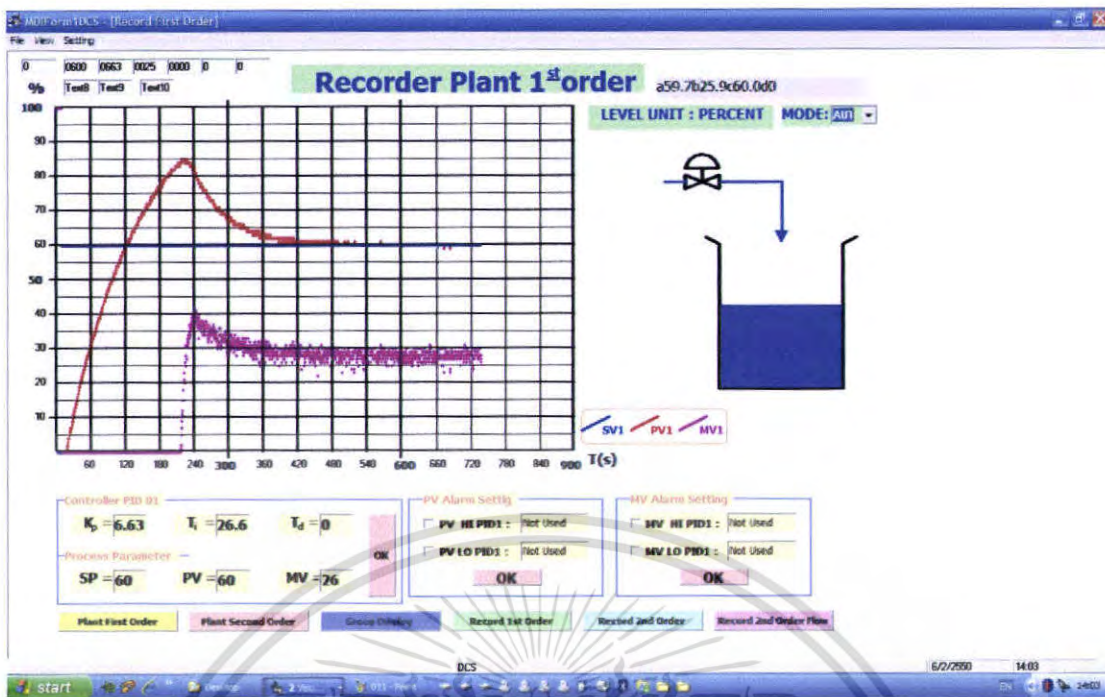
รูปที่ 4.6 กราฟที่ได้จากการทดลองอุปกรณ์แปลงจากดิจิทัลเป็นอนาล็อก  
ในกระบวนการอันดับสอง

#### 4.6 ผลการทดลองระบบควบคุมระดับน้ำอันดับหนึ่ง

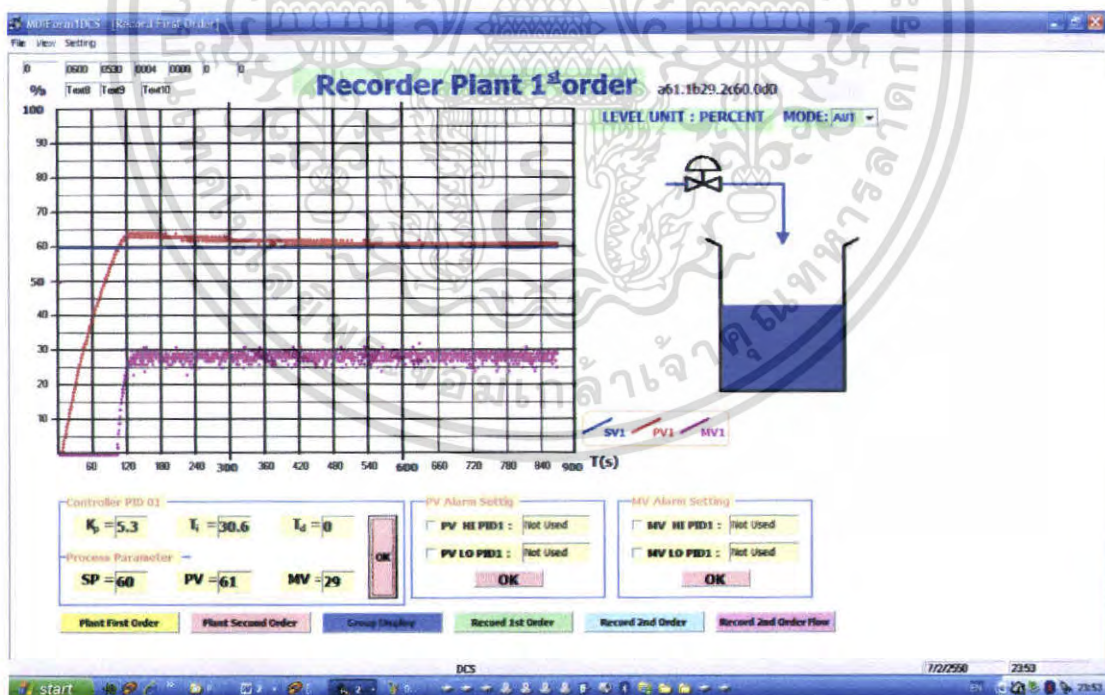
##### 4.6.1 การควบคุมแบบพีไอ

ตารางที่ 4.5 ค่าพารามิเตอร์การควบคุมแบบพีไอของกระบวนการควบคุมระดับน้ำอันดับหนึ่ง

ค่าพารามิเตอร์	ซีเกลอร์และนิโคลส์	คาร์ทลิน
Kp	6.63	5.30
Ti	26.64	130.6

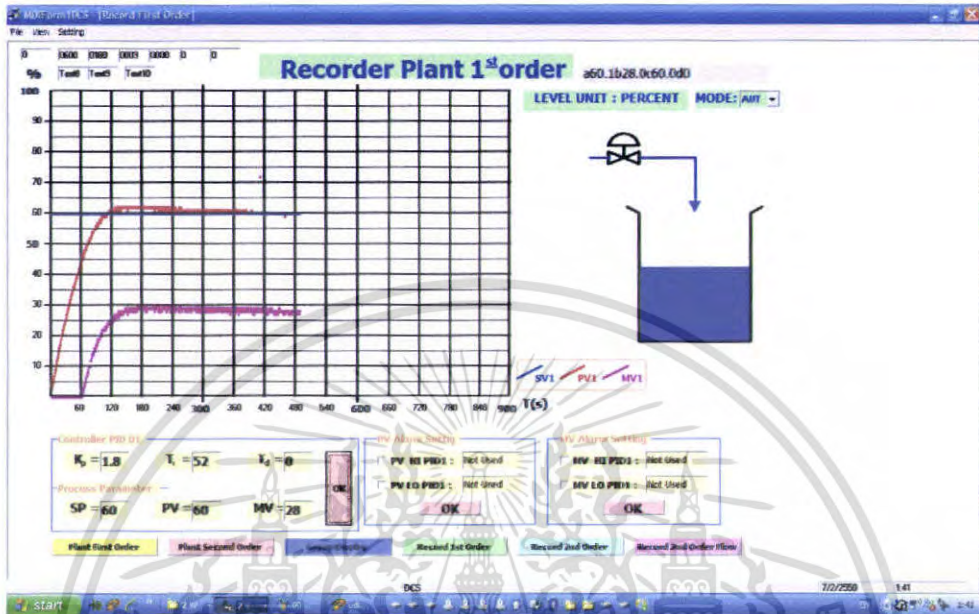


รูปที่ 4.7 ผลตอบสนองของระบบควบคุมแบบพีไอในกระบวนการอันดับหนึ่ง ด้วยวิธีเกิลอร์และนิโคลส์

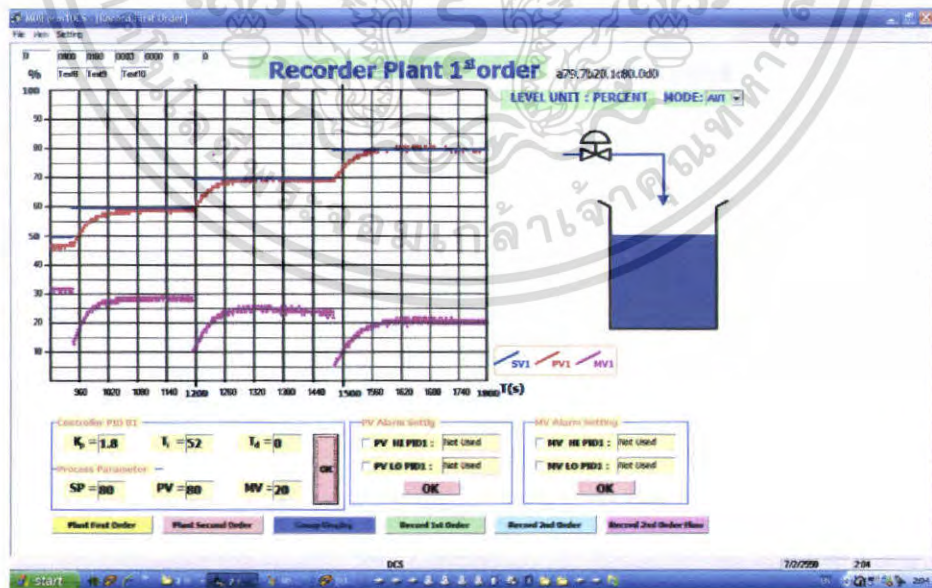


รูปที่ 4.8 ผลตอบสนองของระบบควบคุมแบบพีไอในกระบวนการอันดับหนึ่ง ด้วยวิธีคาล์ดิน

เมื่อได้พิจารณาค่าจากระบบนี้เป็นระบบอันดับหนึ่งดังนั้น การควบคุมแบบพีไอจึงมีความเหมาะสม จึงนำค่าพีไอมาปรับปรุง จึงได้ค่าพารามิเตอร์ที่ทำให้ระบบผลตอบสนองที่เร็วและมีค่าพุงเกินน้อยโดยใช้ค่า  $K_p=1.8$  ,  $T_i = 52$



รูปที่ 4.9 ผลตอบสนองของระบบควบคุมแบบพีไอในกระบวนการอันดับหนึ่งหลังจากปรับปรุงค่าพารามิเตอร์ โดยเปลี่ยนค่าครั้งละ 60 เปอร์เซ็นต์



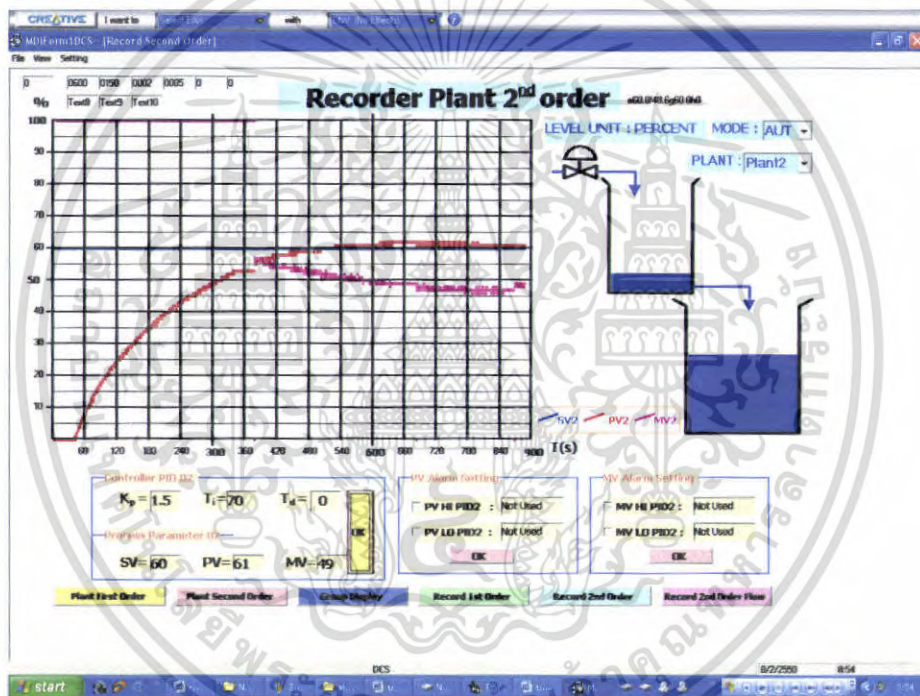
รูปที่ 4.10 ผลตอบสนองของระบบควบคุมแบบพีไอในกระบวนการอันดับหนึ่งหลังจากปรับปรุงค่าพารามิเตอร์ โดยเปลี่ยนค่าครั้งละ 10 เปอร์เซ็นต์

## 4.7 ผลการทดลองระบบควบคุมระดับน้ำอันดับสอง

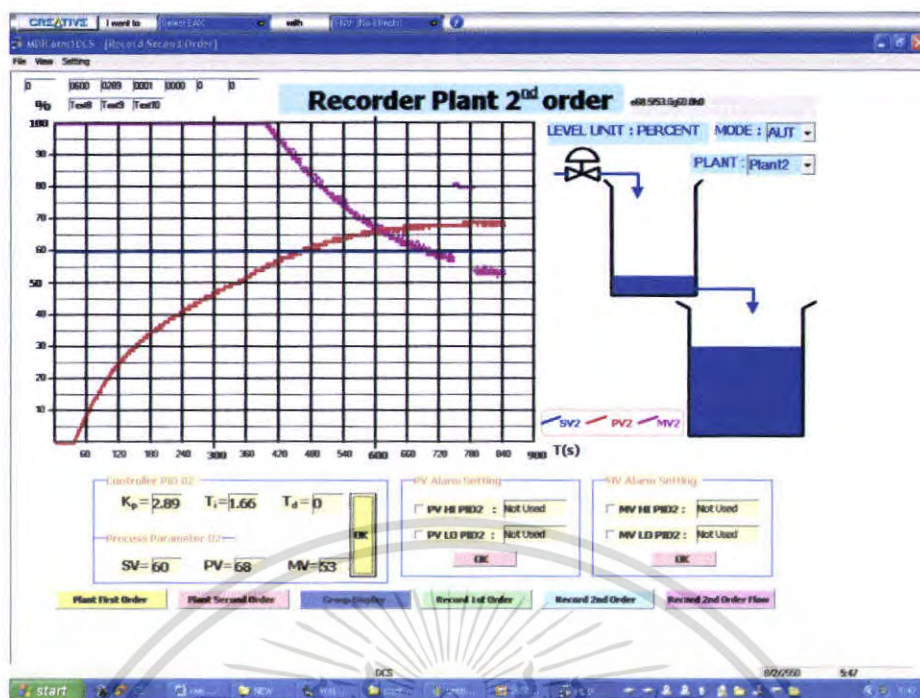
### การควบคุมแบบพีไอ

ตารางที่ 4.6 ค่าพารามิเตอร์การควบคุมแบบพีไอของกระบวนการควบคุมระดับน้ำอันดับสอง

ค่าพารามิเตอร์	ซีเกลอร์และนิโคลส์	ดาห์ลิน
Kp	1.50	2.89
Ti	70	241.66



รูปที่ 4.11 ผลตอบสนองของระบบควบคุมแบบพีไอในกระบวนการอันดับสอง  
ด้วยวิธีซีเกลอร์และนิโคลส์

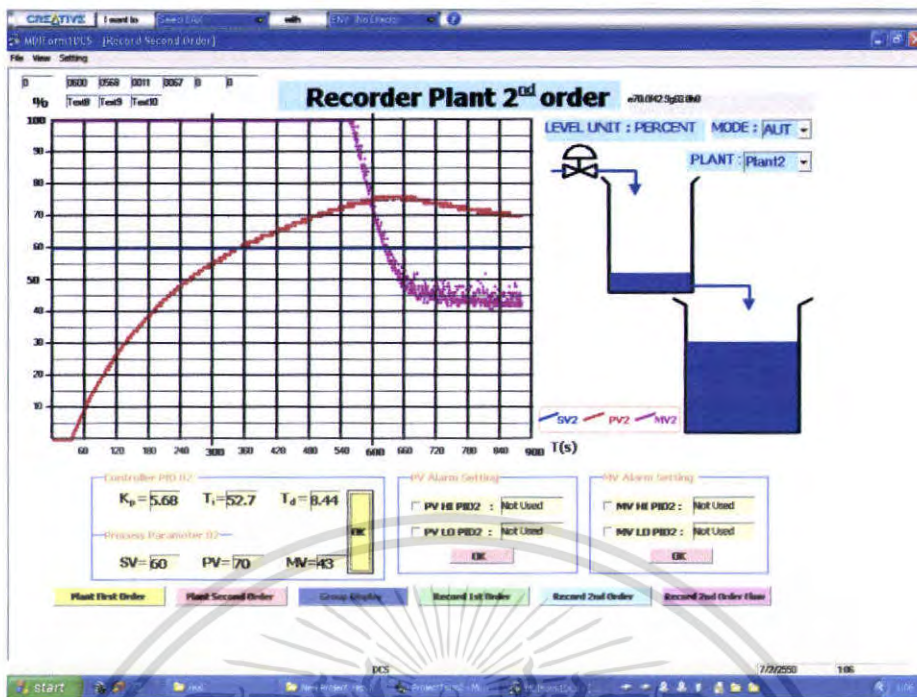


รูปที่ 4.12 ผลตอบสนองของระบบควบคุมแบบพีไอในกระบวนการอันดับสอง  
ด้วยวิธีคาล์ลิน

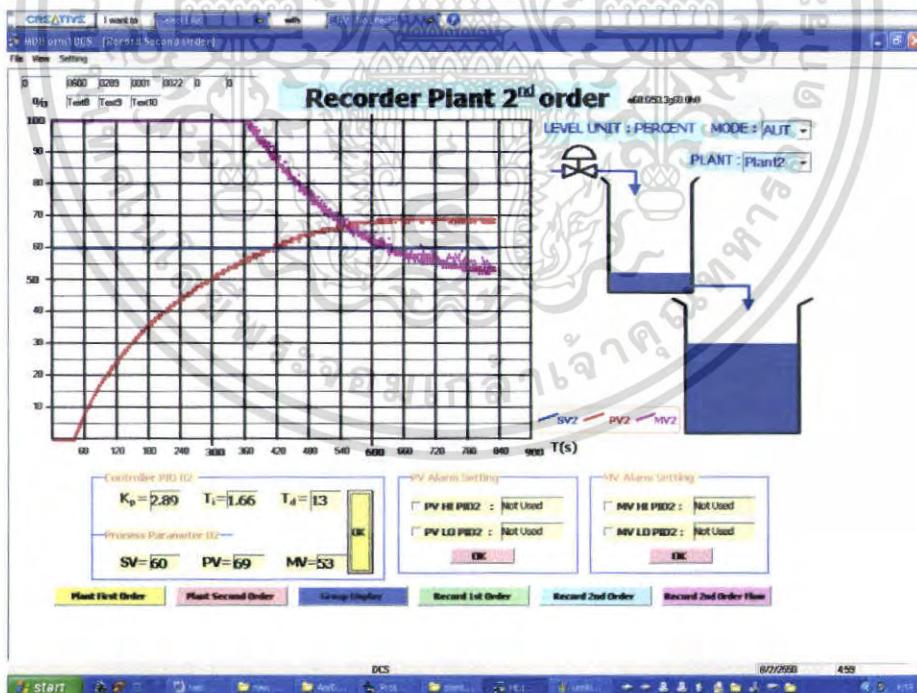
การควบคุมแบบพีไอดี

ตารางที่ 4.7 ค่าพารามิเตอร์การควบคุมแบบพีไอดีของกระบวนการควบคุมระดับน้ำอันดับสอง

ค่าพารามิเตอร์	ซีเกลอร์และนิโคลส์	คาล์ลิน
$K_p$	5.68	2.89
$T_i$	52.77	241.66
$T_d$	8.44	13

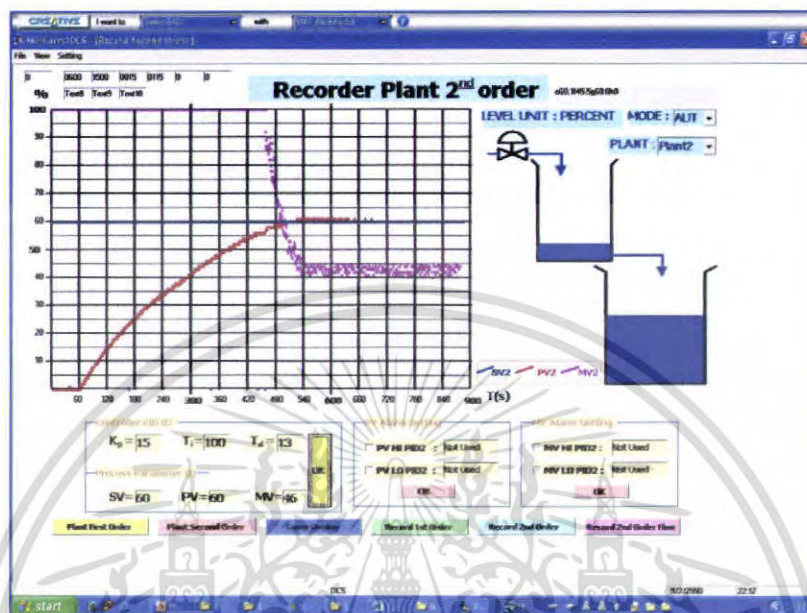


รูปที่ 4.13 ผลตอบสนองของระบบควบคุมแบบพีไอดีในกระบวนการอันดับสอง ด้วยวิธีซีเกลอร์และนิโคลส์

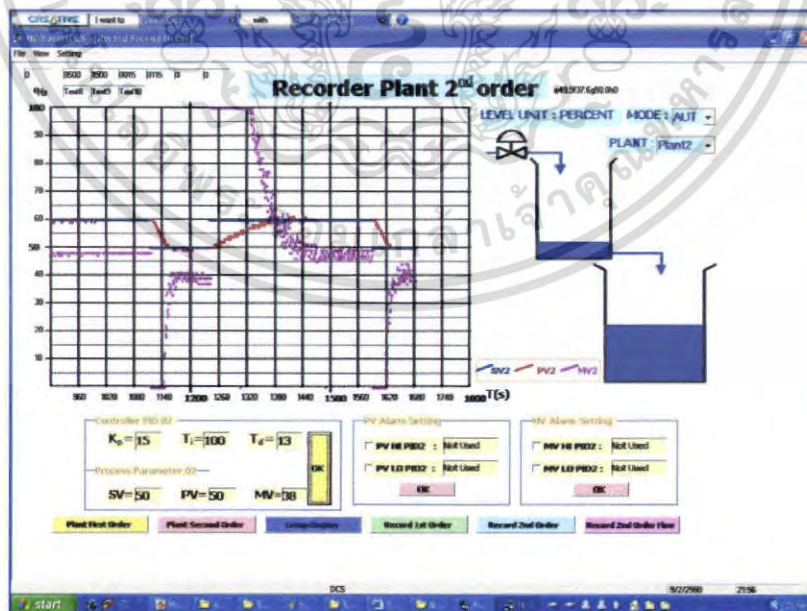


รูปที่ 4.14 ผลตอบสนองของระบบควบคุมแบบพีไอดีในกระบวนการอันดับสอง ด้วยวิธีดาห์ลิน

เมื่อได้พิจารณาค่าจากระบบนี้เป็นระบบอันดับสอง ที่มีผลตอบสนองช้า ดังนั้น การควบคุมแบบพีไอดีจึงมีความเหมาะสม จึงนำค่าพีไอดีมาปรับปรุง จึงได้ค่าพารามิเตอร์ที่ทำให้ระบบผลตอบสนองที่เร็วและไม่มีค่าฟุ้งเกิน โดยใช้ค่า  $K_p=15$ ,  $T_i=100$ ,  $T_d=13$



รูปที่ 4.15 ผลตอบสนองของระบบควบคุมแบบพีไอดีในกระบวนการอันดับสองหลังจากปรับปรุงค่าพารามิเตอร์โดยเปลี่ยนค่าครั้งละ 60 เปอร์เซ็นต์



รูปที่ 4.16 ผลตอบสนองของระบบควบคุมแบบพีไอดีในกระบวนการอันดับสองหลังจากปรับปรุงค่าพารามิเตอร์โดยเปลี่ยนค่าครั้งละ 10 เปอร์เซ็นต์

## บทที่ 5

# บทวิจารณ์และสรุป

### 5.1 ผลการดำเนินงาน

จากการออกแบบและสร้างระบบดิจิทัล เพื่อใช้ในการควบคุมกระบวนการระดับน้ำทั้งสองกระบวนการ จากการดำเนินงานพบว่าสามารถควบคุมกระบวนการทั้งสองได้อย่างที่ได้ออกแบบไว้ คือสร้างระบบควบคุมกระบวนการที่ติดตั้งอยู่ในกระบวนการในที่ไม่โครคอนโทรลเลอร์ dsPIC 30F4011 เป็นหน่วยควบคุมกระบวนการและใช้เครื่องคอมพิวเตอร์ส่วนบุคคลเป็นหน่วยปฏิบัติการของผู้ใช้ และทั้งสองส่วนนี้ติดต่อกันด้วยมาตรฐาน อาร์เอส-485 และผู้ใช้สามารถมองเห็นกระบวนการทั้งสองกระบวนการผ่านหน้าจอคอมพิวเตอร์ ซึ่งจะแสดงภาพจำลองกระบวนการผลิต และการทำงานของอุปกรณ์ต่างๆในกระบวนการ เช่น เครื่องมือวัดระดับน้ำ เป็นต้น และผู้ใช้ยังสามารถปรับแต่งระบบควบคุม เช่น การปรับค่าปรับตั้ง, การปรับค่าพารามิเตอร์ของตัวควบคุม เลือกโหมดในการควบคุมด้วยมือ หรือแบบอัตโนมัติ ซึ่งผู้ใช้สามารถควบคุมและตรวจสอบทั้งสองกระบวนการได้พร้อมกัน โดยไม่ต้องไปปรับค่าที่กระบวนการ ซึ่งเป็นไปตามจุดประสงค์ของระบบดิจิทัล

เนื่องจากกระบวนการระดับน้ำทั้งสองใช้ตัวควบคุมแบบพีไอดี ดังนั้นจึงต้องหาค่าพารามิเตอร์ที่เหมาะสมในการปรับแต่งกระบวนการ โดยในที่นี้ ได้ใช้วิธีของซีเกลอร์และนิโคลส์ กับวิธีคาล์วินเพื่อหาค่าพารามิเตอร์ แล้วนำผลตอบสนองที่มาเปรียบเทียบกับกัน และสามารถสรุปได้ดังนี้

#### 1. การทดลองของกระบวนการอันดับหนึ่ง

1.1 จากการทดลองควบคุมกระบวนการด้วยตัวควบคุมแบบพีไอ โดยใช้พารามิเตอร์ตามที่คำนวณไว้พบว่าผลตอบสนองของวิธีคาล์วิน ให้ผลตอบสนองที่ดีกว่าวิธีของซีเกลอร์และนิโคลส์ เพราะมีพุงเกินน้อยกว่าและสามารถเข้าสู่สภาวะคงตัวได้เร็วกว่า

1.2 เนื่องจากกระบวนการเป็นกระบวนการอันดับหนึ่งซึ่งมีผลตอบสนองที่เร็ว ดังนั้นการใช้ตัวควบคุมแบบพีไอจึงเพียงพอต่อการควบคุมกระบวนการ เนื่องจากมีความสะดวกในการปรับปรุงค่าพารามิเตอร์

1.3 ค่าพารามิเตอร์ที่ใช้ในการปรับแต่งพารามิเตอร์เลือกค่าพารามิเตอร์ให้ระบบมีผลตอบสนองที่เร็ว และยอมรับการเกิดค่าพุงเกินได้

## 2. การทดลองของกระบวนการอันดับสอง

2.1 จากการทดลองควบคุมกระบวนการด้วยตัวควบคุมแบบพีไอ โดยใช้พารามิเตอร์ตามที่คำนวณไว้พบว่าผลตอบสนองของวิธีคาล์ลิน ให้ผลตอบสนองที่ดีกว่าวิธีของซีเกลอร์และ นิโคลส์ เพราะมีพุงเกินน้อยกว่าและสามารถเข้าสู่สภาวะคงตัวได้เร็วกว่า

2.2 จากการทดลองควบคุมกระบวนการด้วยตัวควบคุมแบบพีไอดี โดยใช้พารามิเตอร์ตามที่คำนวณไว้พบว่าผลตอบสนองของวิธีคาล์ลิน ให้ผลตอบสนองที่ดีกว่าวิธีของซีเกลอร์และนิโคลส์ เพราะมีพุงเกินน้อยกว่าและสามารถเข้าสู่สภาวะคงตัวได้เร็วกว่า

2.3 ค่าพารามิเตอร์ที่ใช้ในการปรับแต่งพารามิเตอร์เลือกค่าพารามิเตอร์เลือกค่าพารามิเตอร์ที่ทำให้ไม่เกิด overshoot เนื่องจากระบบมีการแกว่ง เมื่อมีการเกิด overshoot

3. ค่าพารามิเตอร์ที่คำนวณได้ อาจยังไม่ใช่ค่าที่เหมาะสมที่สุดจึงต้องมีการปรับแต่งค่าพารามิเตอร์อีกครั้งเพื่อให้ได้ผลตอบสนองที่ต้องการของแต่ละระบบ

## 5.2 การแก้ไขปัญหา

1. ใช้คุณสมบัติ Auto Redraw แทนเพื่อเก็บรูปภาพและข้อมูลต่างๆ โดยใช้การ Print Screen แทน
2. ยกเลิกการใช้งาน Stack buffer( Use FIFO buffer ) โดยเข้าไปที่ Control Panel>> Hardware>>Device management เลือกพอร์ตอนุกรมเข้าไปที่ Properties>>Port Setting>>Advance แล้วยกเลิกการใช้ Use FIFO buffer
3. ทำการแปลงค่าทศนิยมเป็นเลขจำนวนเต็มแล้วส่งค่าออกไปยังตัวไมโครคอนโทรลเลอร์
4. สอบถามจากอาจารย์และคอบัฟเฟอร์เข้าที่ขาส่งสัญญาณ SDA และ SCL ของระบบบัส I2C
5. ตรวจสอบวงจรและทำการแก้ไขวงจรใหม่อีกครั้ง
6. หลังจากทดลองควบคุมกระบวนการแล้วพยายามไม่ให้น้ำแห้งจนหมด จะเหลือน้ำค้างอยู่ในระบบส่วนหนึ่ง
7. พยายามปรับให้ระบบมีค่าคงตัวเวลามีค่าน้อยเพื่อลดเวลาในการบันทึกค่า
8. พยายามให้คอมพิวเตอร์ทำงานโดยใช้โปรแกรมควบคุมเพียงอย่างเดียว

## 5.3 สรุปผลการทำโครงการ

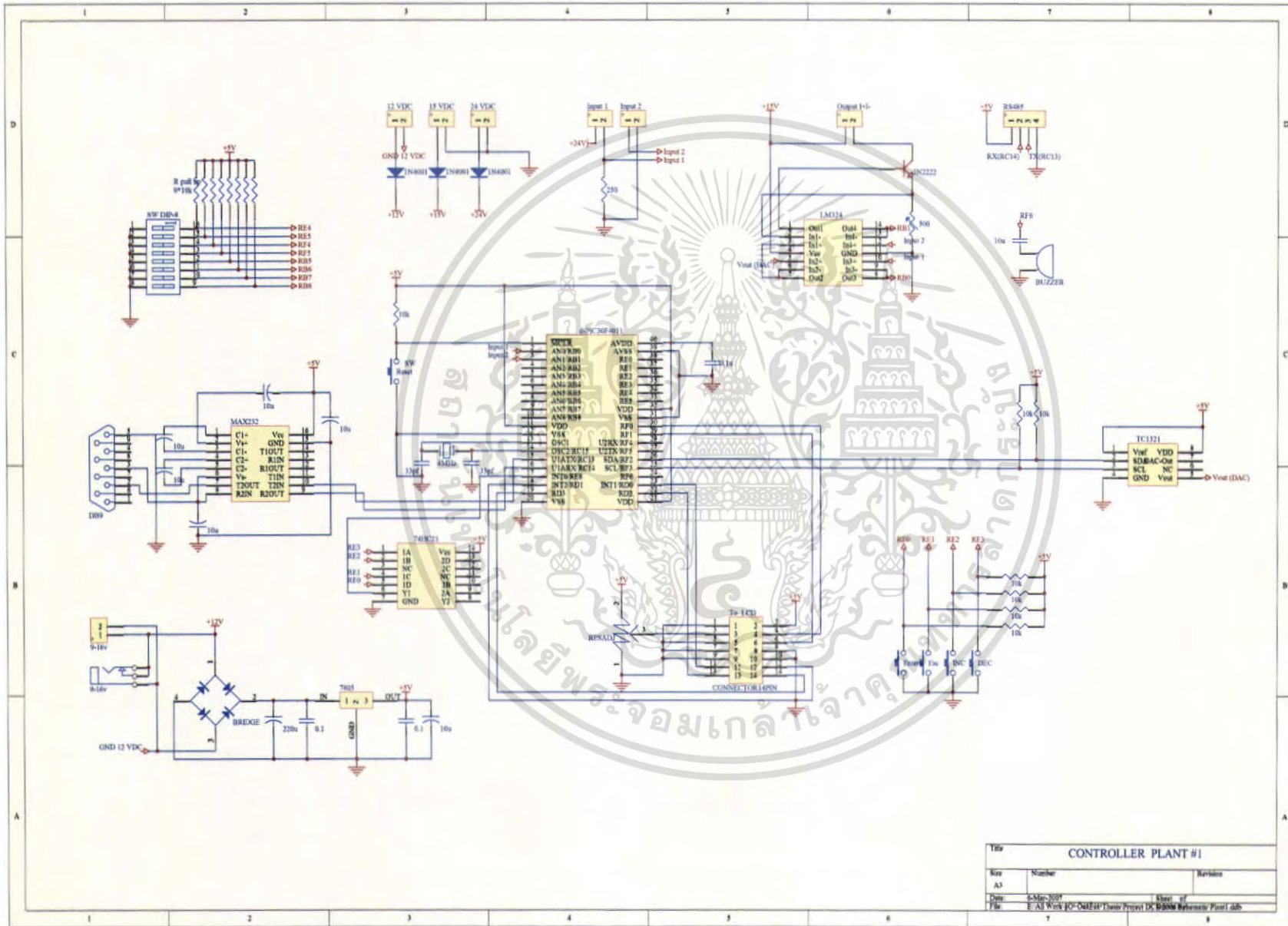
จากการสร้างระบบคิซีเอสโดยใช้ไมโครคอนโทรลเลอร์ ระบบสามารถควบคุมได้ ซึ่งตัวคอนโทรลเลอร์ที่สร้างขึ้นมาเป็นตัวควบคุมกระบวนการที่สามารถนำมาใช้งานได้จริง สามารถนำมาใช้ควบคุมเดี่ยวๆได้และยังสามารถที่ต่อเข้ากับระบบเพื่อควบคุมระยะไกลโดยผ่านตัวเชื่อมต่อ

เครือข่าย ในโครงการนี้ใช้การเชื่อมต่อด้วยระบบอาร์เอส422/485 การควบคุมจะใช้โปรแกรมที่สร้างขึ้นมาเพื่อใช้ในการควบคุมเฉพาะระบบที่สร้างขึ้นม่านั้นไม่สามารถเชื่อมต่อกับระบบควบคุมอื่น ระบบที่ต้องการควบคุมของโครงการนี้คือ ระบบควบคุมระดับน้ำซึ่งเป็นกระบวนการอันดับหนึ่งและกระบวนการอันดับสอง จากผลในการควบคุมโดยการหาค่าพารามิเตอร์ด้วยวิธีซีเกลอร์และนิโคลส์ และการหาค่าพารามิเตอร์ด้วยวิธีคาล์วิน จะเป็นค่ากลางที่จะใช้ในการปรับแต่งค่าพารามิเตอร์ของระบบควบคุมระดับน้ำอันดับหนึ่งและระบบควบคุมระดับน้ำอันดับสอง จากนั้นต้องปรับแต่งค่าพารามิเตอร์ให้เหมาะกับกระบวนการอีกครั้งเพื่อให้ได้ค่าที่ดีที่สุด ซึ่งโครงการที่ทำนี้ค่าที่ดีที่สุดสำหรับระบบควบคุมระดับน้ำกระบวนการอันดับหนึ่งคือ  $K_p = 1.8$ ,  $T_i = 52$ ,  $T_d = 0$  และค่าที่ดีที่สุดสำหรับระบบควบคุมระดับน้ำกระบวนการอันดับสองคือ  $K_p = 15$ ,  $T_i = 100$ ,  $T_d = 13$

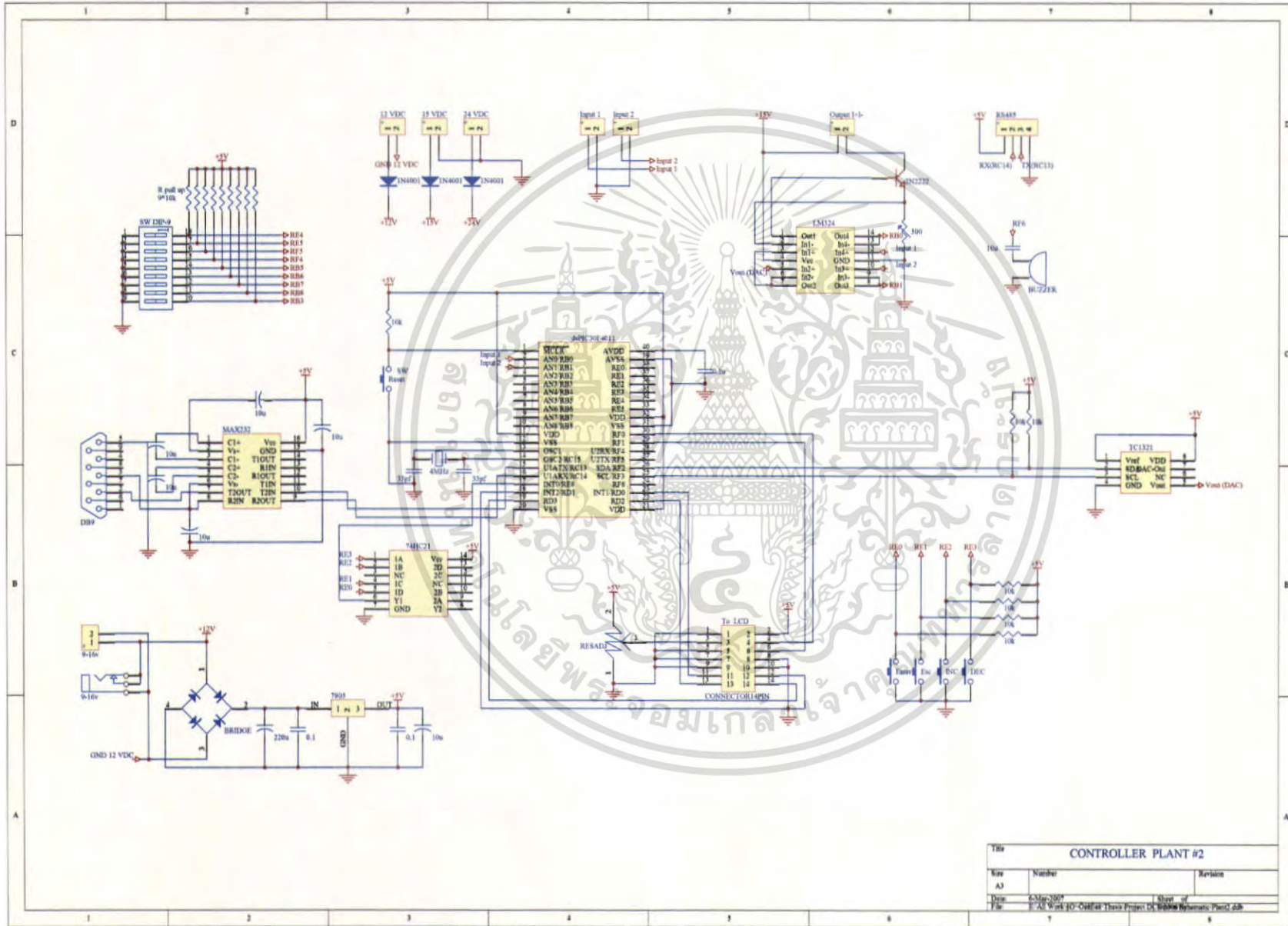
#### 5.4 แนวทางการพัฒนา

1. ทำระบบให้สามารถทดสอบผลตอบสนองโดยรวมของกระบวนการและทำระบบเป็นแบบ ปรับแต่งค่าพารามิเตอร์อัตโนมัติ
2. สร้างตัวคอนโทรลเลอร์ให้มีขนาดเล็กเหมาะแก่การใช้งานจริง
3. สร้างระบบควบคุมเป็นแบบไร้สาย



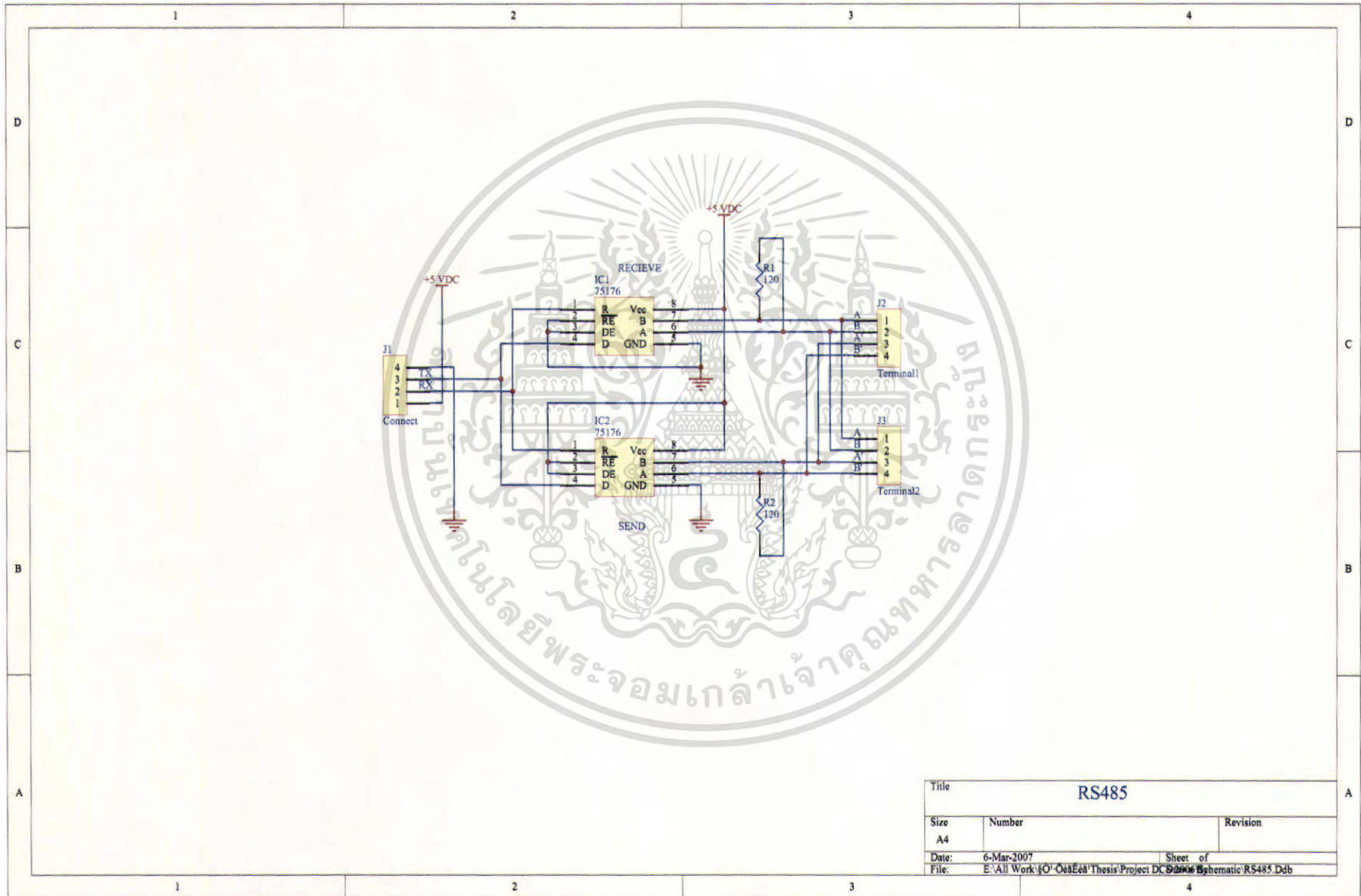


Title		
CONTROLLER PLANT #1		
Size	Number	Revision
A3		
Date	6-Mar-2007	Sheet 4 of 4
File	E:\AI Work\PCB-Cad\PCB Design Project DC-RBMN Reference Plant1.dtb	

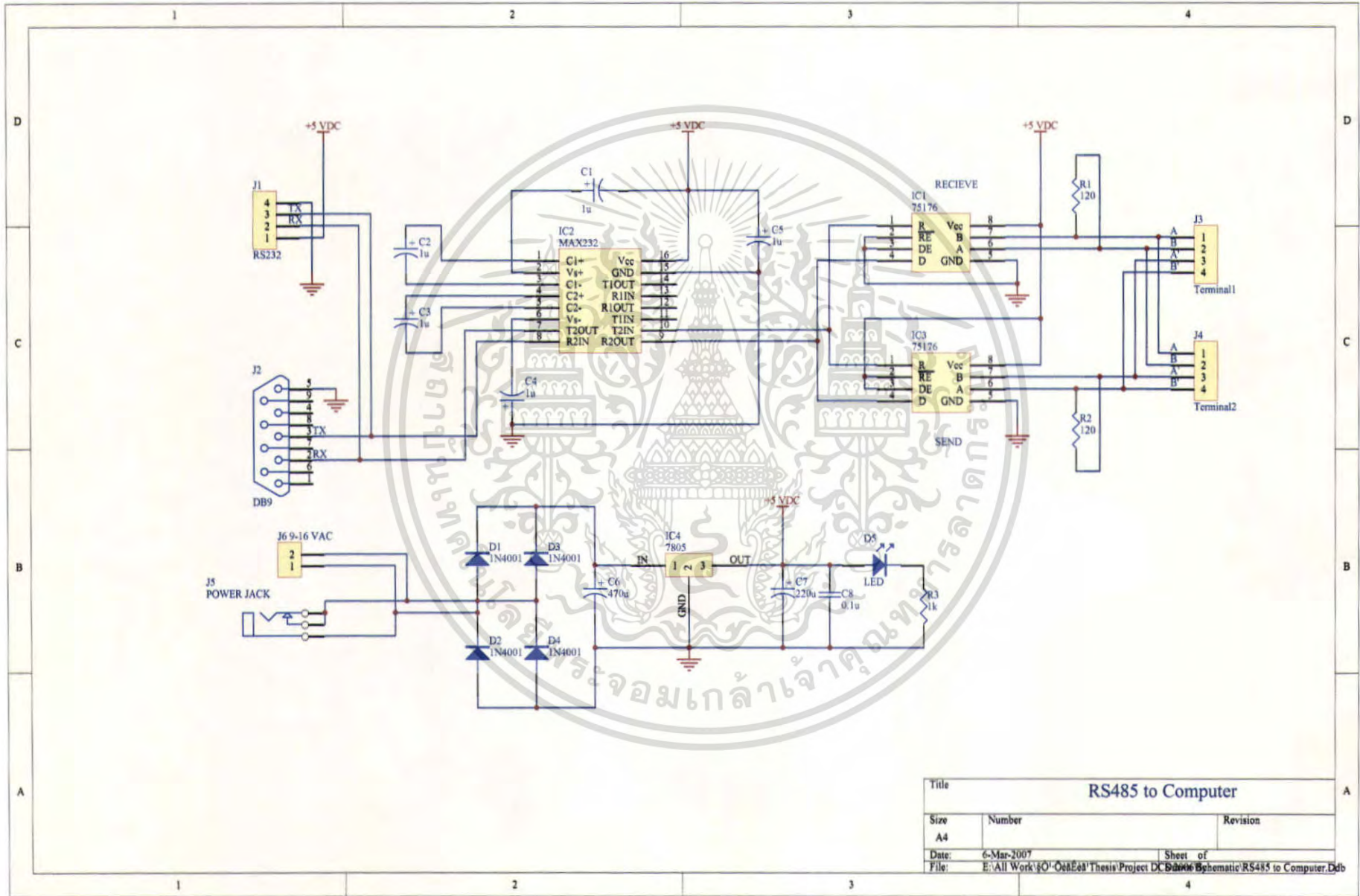


Title		
CONTROLLER PLANT #2		
Rev	Number	Revision
AJ		
Date	6/Mar/2007	Sheet of
File	E:\AI 9\ai 10-Original Thesis Project Dr. Bhanu Srinivasan Plant02.dwg	





Title		RS485
Size	Number	Revision
A4		
Date:	6-Mar-2007	Sheet of
File:	E:\All Work\O' Oat\Ed Thesis Project DC 2006\Bchematic RS485.Ddb	



Title			RS485 to Computer		
Size	Number	Revision			
A4					
Date:	6-Mar-2007	Sheet of			
File:	E:\All Work\IO\O&E\Thesis Project DC Bus\Bchematic\RS485 to Computer.Ddb				

## บรรณานุกรม

1. ผศ.พรสุข รัตโรจน์นันต์.พื้นฐานการควบคุมกระบวนการ (Fundamentals of Process Control Theory).กรุงเทพมหานคร: แผนกตำรา คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.2549.
2. รศ.สุเชียร เกียรติสุนทร.ดีซีเอส (Distributed Control System:DCS).กรุงเทพมหานคร:สมาคมส่งเสริมเทคโนโลยี(ไทย-ญี่ปุ่น).2543.
3. ชาริน สิทธิธรรมชารี.ประชา พถกษ์ประเสริฐ.Microsoft Visual Basic 6.0.กรุงเทพมหานคร: บริษัท ชัคเซส มีเดีย จำกัด.2548.
5. Carlos A. Smith, Armando B. Corripio. Principles and Practice of Automation Process Control. 2nd ED. New York : John Wiley & Sons, Inc 1997.
6. Armando B. Corripio. Tuning of Industrial Control System. 2nd ED. North Carolina : ISA – The Instrumentation, Systems, and Automation Society, Inc 2001.
7. นคร ภัคดีชาติ, ชัยวัฒน์ ลิ้มพรจิตรวิไล. คู่มือการทดลอง dsPIC Microcontroller เบื้องต้นด้วยโปรแกรมภาษา C กับ MPLAB C30. กรุงเทพมหานคร : บริษัท อินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด. 2548